ARM® TrustZone® True Random Number Generator

Revision: r0p0

Software Integrator's Manual

ARM

ARM 101049_0000_00_en

ARM® TrustZone® True Random Number Generator

Software Integrator's Manual

Release information

Document History

Issue	Date	Confidentiality	Change
0000-00	27 June 2017	Non-Confidential	First release for r0p0

Proprietary Notice

Web Address

http://www.arm.com

Contents

ARM® TrustZone® True Random Number Generator Software Integrator's Manual

Preface

This preface introduces the *ARM® TrustZone® True Random Number Generator Software Integrator's Manual*.

It contains the following:

About this book

This book is for the ARM® TrustZone® TRNG True Random Number Generator. This document describes the true-random-number-generator driver that is provided by ARM. It provides the required information to integrate and use the ARM TrustZone TRNG library.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction

This chapter provides an overview of the ARM TrustZone® TRNG.

Chapter 2 TRNG driver algorithm

This chapter describes the TRNG driver algorithm.

Chapter 3 The ARM® TrustZone® TRNG API

This chapter describes the ARM TrustZone TRNG API.

Chapter 4 Product deliverables

This chapter lists ARM TrustZone TRNG product deliverables and describes its build environment.

Chapter 5 Supported TRNG driver modes

This chapter describes the TRNG driver modes supported by ARM TrustZone TRNG.

Chapter 6 Integrating ARM® TrustZone® TRNG

This chapter describes the steps that are required to integrate ARM TrustZone TRNG software into a SoC.

Glossary

The ARM® Glossary is a list of terms used in ARM documentation, together with definitions for those terms. The ARM Glossary does not contain terms that are industry standard unless the ARM meaning differs from the generally accepted meaning.

See the *ARM® Glossary* for more information.

Typographic conventions

italic
: Introduces special terminology, denotes cross-references, and citations.

bold
: Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

`monospace`
: Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

`mono`space
: Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`
: Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`
: Denotes language keywords when used outside example code.

`<and>`
: Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example:

```
MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>
```

SMALL CAPITALS
: Used in body text for a few terms that have specific technical meanings, that are defined in the *ARM® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Feedback

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to *errata@arm.com*. Give:

- The title *ARM TrustZone True Random Number Generator Software Integrator's Manual*.
- The number ARM 101049_0000_00_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

ARM tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

Other information

- *ARM Developer*.
- *ARM Information Center*.
- *ARM Technical Support Knowledge Articles*.
- *Support and Maintenance*.
- *ARM Glossary*.

Chapter 1
Introduction

This chapter provides an overview of the ARM TrustZone® TRNG.

It contains the following sections:

1.1 Overview

The ARM TrustZone TRNG enables generation and collection of a truly random bit stream from a digital logic. The TRNG is designed for simple SoC integration. The typical usage of a TRNG is key generation or for seeding approved deterministic random numbers.

Note

ARM TrustZone TRNG supports 32-bit systems.

1.2 Compliance

ARM TrustZone TRNG complies with the following specifications:

- AIS-31: *Functionality Classes and Evaluation Methodology for True Random Number Generators*, version 3.1, Version 3.1, September 2001, compliant in an implementation using FETRNG driver.
- SP 800-90B Second DRAFT: *Recommendation for the Entropy Sources Used for Random Bit Generation*, January 2016, compliant with section 4.4 Approved Continuous Health Tests.

Chapter 2
TRNG driver algorithm

This chapter describes the TRNG driver algorithm.

It contains the following section:

2.1 TRNG driver algorithm workflow

ARM TrustZone TRNG supports both FE and 800-90B TRNG driver modes.

The TRNG driver implements the following algorithm for random-bit collection:

1. Sets `SAMPLE_CNT` according to the value that is received by the ARM characterization process.
2. Sets `RND_SRC_SEL` to the fastest inverter-chain (ROSC).
3. Collects N bytes from the hardware.
4. TRNG driver samples the following errors through RNG_ISR:
   - The 800-90B driver aborts after 12 CRNGT errors.
   - The FE driver aborts in the following situations:
     — After 12 CRNGT errors.
     — Autocorrelation error.
     — VN error.
5. If random-bit collection did not complete due to one of the error conditions that are listed in *4*:
   a. The TRNG driver selects the next-fastest enabled inverter-chain (ROSC).
   b. Returns to step *3*.If no inverter-chains remain, the driver returns an error indication.

Note

If the TRNG driver returns an error, do not attempt to use ARM TrustZone TRNG to collect more bits until you reset the system.

Chapter 3
The ARM® TrustZone® TRNG API

This chapter describes the ARM TrustZone TRNG API.

It contains the following section:

3.1 API parameters

The API lets you define the number of random bits to be collected and returns the number of actual bytes.

The API takes the following format:

```
uint32_t CC_TrngGetSource(unsigned long rngRegBase, uint8_t *outAddr, size_t *outLen, size_t
reqBits)
```

Table 3-1 cc_TrngGetSource Parameters

Name	I/O	Description
`rngRegBase`	I	ARM TrustZone TRNG logical base register address.
`outAddr`	O	Result buffer.
`outLen`	O	Number of collected bytes.
`reqBits`	I	Number of random bits to collect.

Chapter 4
Product deliverables

This chapter lists ARM TrustZone TRNG product deliverables and describes its build environment.

It contains the following sections:

4.1 Deliverable components

The following components are supplied with the software release package:

Software Package
: The ARM TrustZone TRNG driver.

Integration test
: The ARM TrustZone TRNG integration test package verifies that the ARM TrustZone TRNG software library is integrated correctly.

TRNG characterization
: The ARM TrustZone TRNG characterization core code, for you to adapt for the H/W characterization process.

4.2 Project tree

Each product package contains one or more components as applicable.

This table lists the possible components (directories).

Table 4-1 Project tree structure

Path	Description
`host/src/tztrng_lib`	Host-library domain source code, which is the primary directory for building the release.
`shared/hw/include`	Header files of inter-domain APIs and Source code that can be used on different domains simultaneously.

The Host source code also contains the integration tests directory:`host/src/tests/tztrng_test`.

4.3 Host code build environment

ARM TrustZone TRNG software supports multiple compile-time configurations.

These configurations are controlled using the `CC_CONFIG_TRNG_MODE` configuration flag, which resides in the `proj.ext.cfg` file in the root folder. This flag sets the mode of execution (800-90B or FE). For more information, see *6.2 TRNG characterization* on page 6-23.

The following makefile operations are available:

Build host components.
Run `make` from `host/src/tztrng_lib` to recursively build all host elements, and to copy them into the output directories: `host/lib`, `host/include`, and `host/bin`.

Clean old builds.
Run `make clean` from `host/src/tztrng_lib`.

Note

If the element depends on a specific library, build and export the library to `host/lib`.

The build environment depends on the following environment variables, effective in the current shell session:

Table 4-2 Build environment variables

Environment variable	Supported values	Description
`ARCH`	`arm`	The target Host architecture of the processor.
`OS`	• `linux` • `FreeRTOS`	The target OS. The partner must define and implement the relevant OS. **Note** Linux can be used only on the ARM platform. Only `FreeRTOS` must be used on the platform of a partner in a production environment.
`CROSS_COMPILE`	• `arm-ds5` • `arm-xilinx-linux-gnueabi-` • `arm-none-eabi-`	The cross-compiler toolchain prefix, same as the prefix used to compile the Linux kernel.

Chapter 5
Supported TRNG driver modes

This chapter describes the TRNG driver modes supported by ARM TrustZone TRNG.

It contains the following section:

5.1 Overview of modes

ARM TrustZone TRNG supports the following TRNG driver modes:

800-90B TRNG driver
: Supports the SP 800-90B Second DRAFT: *Recommendation for the Entropy Sources Used for Random Bit Generation* standard.

FE TRNG driver (default value)
: Supports the AIS-31: *Functionality Classes and Evaluation Methodology for True Random Number Generators* standard.

The ARM TrustZone TRNG can be configured to:

- Collect random bits according to the current 800-90B publication.
- Collect random bits in FE mode according to the AIS-31 standard.

For more information, see *4.3 Host code build environment* on page 4-18.

Chapter 6
Integrating ARM® TrustZone® TRNG

This chapter describes the steps that are required to integrate ARM TrustZone TRNG software into a SoC.

It contains the following sections:

6.1 PAL and HAL layers

You are expected to set up the Platform Abstraction Layer (PAL) and Hardware Abstraction Layer (HAL).

These layers enable access to the ARM TrustZone TRNG. After you set the layers, you can extract the driver code into the SoC software tree and compile the code.

6.2 TRNG characterization

The characterization process determines the sample count for each inverter-chain (ROSC).

Before you can use the ARM TrustZone TRNG, you must perform the characterization process.

Setting the TRNG Type

Set the TRNG type in `proj.ext.cfg`:

- For FE TRNG driver, set `CC_CONFIG_TRNG_MODE = 0`.
- For 800-90B TRNG driver, set `CC_CONFIG_TRNG_MODE = 1`.

Customizing the 800-90B TRNG driver

Configure the `host/src/tztrng_lib/include/config_trng90b.h` file for the target platform, according to the characterization process.

Customizing the FE TRNG driver

Configure the `host/src/tztrng_lib/include/config_fetrng.h` file for the target platform, according to the characterization process.