SP 800-90B Non-Proprietary Public Use Document
Infineon 16-bit Security Controller M7893 Entropy Source
Document Version 1.1

Hardware Versions:

M7893B11

Infineon Technologies AG
Am Campeon 1-15
Neubiberg, BY 85579, Germany
September 17, 2024

Revision History

Version	Change
1.0	Initial version
1.1	Configuration Settings

Table of Contents

Description 4

Security Boundary 4

Operating Conditions 4

Configuration Settings 4

Physical Security Mechanisms 4

Conceptual Interfaces 5

Min-Entropy Rate 5

Health Tests 5

Maintenance 5

Required Testing 5

Description

The Infineon 16-bit Security Controller M7893 Entropy Source is a physical entropy source. The entropy source is a Non-IID source and was tested for the hardware version listed on the title page.

Security Boundary

The Infineon 16-bit Security Controller M7893 Entropy Source is entirely contained within a single-chip module.

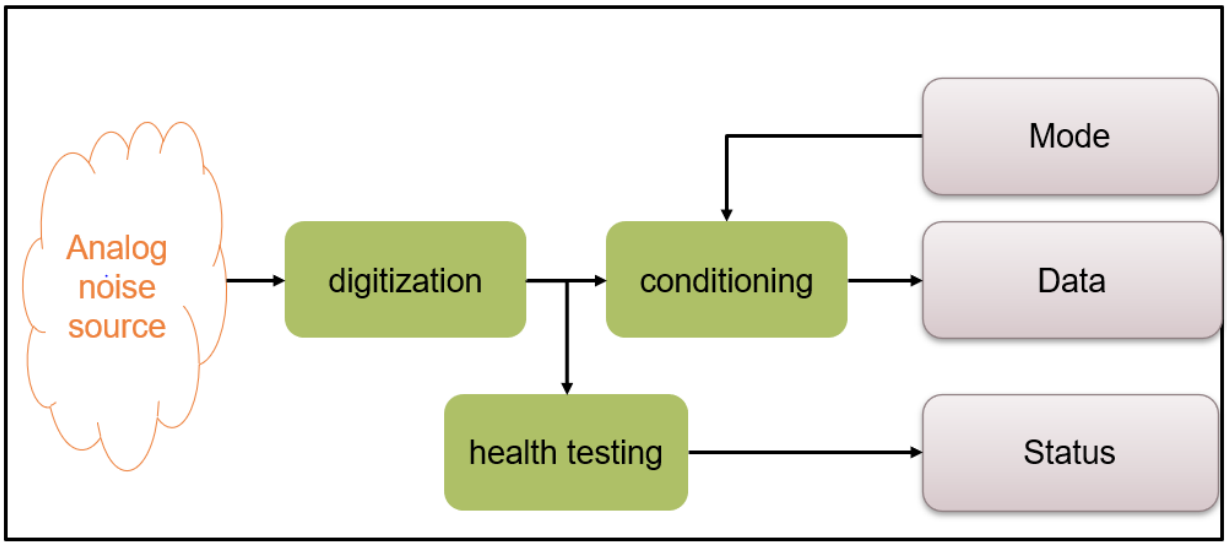


Figure 1. The Infineon 16-bit Security Controller M7893 Entropy Source

Operating Conditions

The following table lists the relevant operating conditions for which the Infineon 16-bit Security Controller M7893 Entropy Source was characterized.

Parameter	Value	Description
Temperature	-40°C to 115°C	Operating ambient temperature range
VCC	1.62 V to 5.50 V	Operating supply voltage

Configuration Settings

To select the SP 800-90B validated source, the Infineon 16-bit Security Controller M7893 Entropy Source must be set to either **BYTE MODE** or **WORD MODE** in the TRNG_CTRL register. To start the entropy source, the mode must be set from **Disabled Mode** into **Enabled Mode**. There are no further configuration settings available to the operator.

Physical Security Mechanisms

The Infineon 16-bit Security Controller M7893 Entropy Source is part of a single-chip module that meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations. The module employs standard passivation techniques.

Conceptual Interfaces

GetEntropy interface is a read operation on the ***Data*** register of the Infineon 16-bit Security Controller M7893 Entropy Source. Once the ready flag is set in the ***Status*** register, 16 bit of entropy data can be read from the ***Data*** register.

There is no dedicated *HealthTest* interface. Data is only released once the health testing is passed. In case health testing fails, a flag is set in the ***Status*** register. To restart the entropy source, the ***Mode*** must be set to **Operational Mode**.

GetNoise is an internal interface only available in a dedicated privileged mode.

Min-Entropy Rate

We claim an output (H_out) of at least 2.55928 bits per byte, or 5.11856 bits of min entropy per 16-bit block.

Health Tests

The Transition Count health test, a developer-defined health test, is applied both on startup and as a continuous test and assesses the health of the noise source using blocks of at least 64 RAW bits when generating an output byte with a False Positive Rate of $\approx 2^{-21}$. As a developer-defined health test, the Transition Count health test complies with the requirements of SP 800-90B Section 4.5.

Continuous health testing is performed for each generated output symbol during normal operation. An output symbol is only released to the ***Data*** register if continuous health testing was passed.

Start-up health testing is performed by applying the Transition Count health test to 1024 RAW bits. This is initiated by reading 16 bytes of conditioned data from the Infineon 16-bit Security Controller M7893 Entropy Source ***Data*** register. If no alarm was indicated in the ***Status*** register after 16 bytes were read, the start-up health testing result is “pass” for the 1024 RAW bits.

On demand health testing is possible by reading entropy data from the Infineon 16-bit Security Controller M7893 Entropy Source ***Data*** register and monitoring the health test error bit in the ***Status*** register.

Maintenance

There is no special maintenance necessary to operate the Infineon 16-bit Security Controller M7893 Entropy Source.

Required Testing

There is no additional testing required by the target platform. Since *GetNoise* is an internal interface only available in a dedicated privileged mode, the user must rely on the health test to ensure the entropy source is configured correctly and is working as expected.