DemoKey 2.4

Demonstrator Quickstart Guide

Version	**2.0**
Date	**September 6, 2018**
Status	**Approved**
Reference	**Demonstrator Quickstart Guide**

www.intrinsic-id.com

History Information

Version	Date	Change Description	Modified by:	Reviewed by:
1.0	2017-09-21	First release.	PB	OH, SG, TA, ML
2.0	20 2018-08-07	New Product Naming Update.	DA	BO

Table of Contents

1. Introduction

1.1. Purpose

The purpose of the demonstrator is to show how unclonable keys can be generated from the unique physical properties of the underlying hardware using Intrinsic ID's DemoKey. This product is based on the concept of SRAM Physical Unclonable Functions (PUFs), where the underlying SRAM is used as a silicon fingerprint from which device-unique keys can be derived. The demonstrator shows the derivation of an ECC 256 bit private key from the SRAM PUF. In addition, it proves that the same key will be reconstructed each time, even after power cycles.

The public key corresponding to the derived ECC 256 bit private key can be computed to create an ECC keypair. This keypair in turn can be used for creating device certificates or authenticating the hardware.

DemoKey is a simulation version mimicking the behavior and functionality of the Intrinsic ID's production version BroadKey. In order to allow developers to effortlessly migrate from DemoKey to BroadKey, the API's are compatible.

After going through this Quickstart Guide, before adding the DemoKey software in a user-defined project, it is highly recommended to read the *IID-DK2-4-DS.pdf* and IID-DK2-4-IG*.pdf*.

Disclaimer: DemoKey is a simulation version of the commercial BroadKey product, intended explicitly for demonstration and evaluation purposes only.

1.2. Definitions, Acronyms and Abbreviations

AC	Activation Code
API	Application Programming Interface
bk_	BroadKey (as prefix in function/variable names)
IP	Intellectual Property
NVM	Non-Volatile Memory
PUF	Physical Unclonable Function
SD	Start-up Data (of uninitialized SRAM)
SRAM	Static Random Access Memory

1.3. Overview

This document is organized as follows:

- Section 2 contains the prerequisites needed to run the demonstrator.
- Section 3 contains practical details for running and debugging the demonstrator.
- Section 4 contains a functional overview of the demonstrator's execution flow.

2. Prerequisites

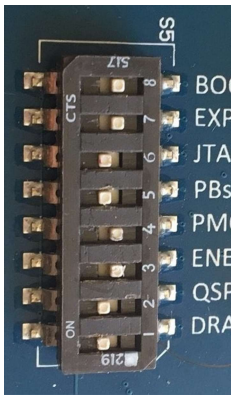
2.1. Required Software and Tools

- e² studio Integrated Solution Development Environment (ISDE)
- Synergy Software Package (SSP), Release 1.4.1
- DK-S7G2 Development Kit

Please read *Development Toolbox Setup Guide.pdf* to learn how to install the ISDE and SSP package and how to set up and connect the DK-S7G2 kit.

2.2. Dip Switch Settings

In order to be able to run the demonstrator, the *S5* DIP switches on the main board must be configured as indicated below:



DIP switch name	Setting
BOOT	Off
EXP	Off
JTAG	On
PBs	On[1]
PMOD	Off
ENET1	Off[1]
QSPI	On[1]
DRAM	On

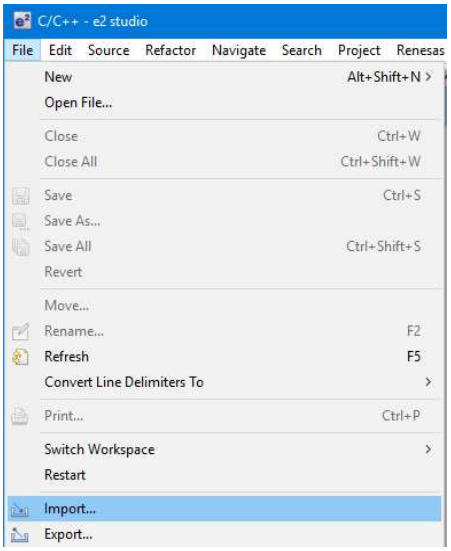
[1] Switch setting differs from the setting as seen in the Development Toolbox Setup Guide.

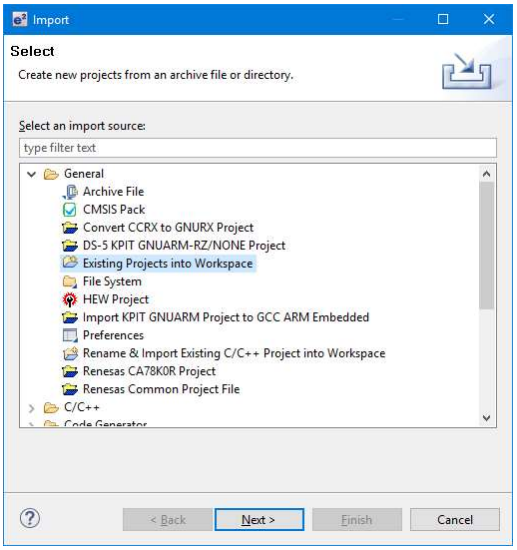
3. Running the Demonstrator

3.1. Building the Demonstrator Image

The first step is building the demonstrator project into an executable image that can be flashed and run on the board.

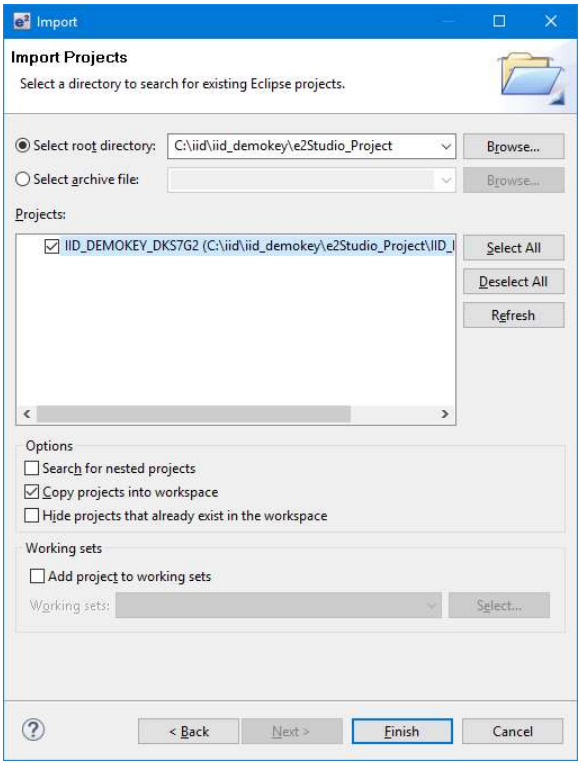
The project, named IID_DemoKey_DKS7G2, must be imported in an e² studio workspace. In the menu click on **File** and select **Import…**





In the **Import** dialog from the **General** category select **Existing Projects into Workspace**. Press **Next** to go to the **Import Projects** dialog.

Browse to the directory containing the **IID_DemoKey_DKS7G2** project and make sure that the **Copy projects into workspace** option is enabled. Check that the project is selected and press **Finish**.



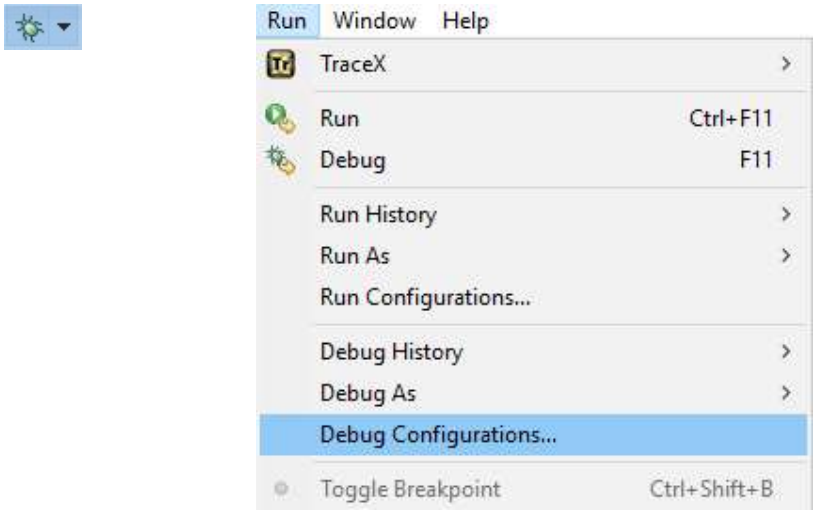
Now that the project is included in the Project Explorer, expand the project and double click on the **configuration.xml** file located in the project's root directory and wait for the file to be parsed and opened in the editor. Click on **Generate Project Content**, which will place all drivers for the board from the SSP package and place these in the project.

In order to build the imported project, right click on the project name and select **Build Project**. This will build the demonstrator firmware that can be programmed on the DK-S7G2 board.

3.2. Running and Debugging the Demonstrator

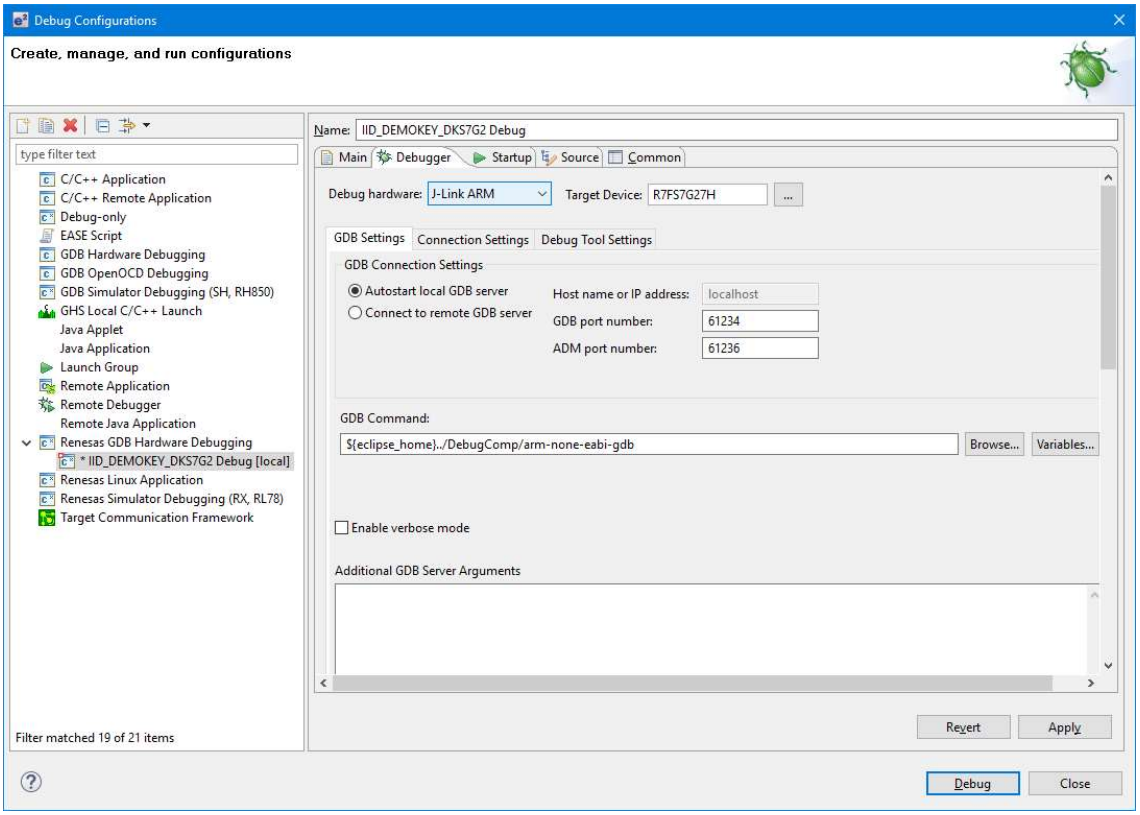
Before programming the firmware, confirm all cables are connected between the board and the PC running e^2 studio.

Verify the IID_DemoKey_DKS7G2 demonstrator project is selected in the **Project Explorer**, otherwise select it now. Click on the **Debug** button in the menu, select **Debug** from the **Run** menu or press **F11**.



The demonstrator firmware will now be flashed onto the board and the debug process will start.

If by any chance e^2 studio flashes the wrong program, it is possible to force the debugger to use the demonstrator by selecting **Debug Configurations** from the **Run** menu. This will open the **Debug Configurations** dialog. In this dialog under the section **Renesas GDB Hardware Debugging**, select the demonstrator project and press **Debug**.



The debugger first halts the program at the entry point of the **Reset_Handler** function. Press the **Resume** button from the menu or press **F8** to continue. The debugger now halts at the **main** function. Again, press the **Resume** button or **F8**.

The demonstrator is now started and the user can interact with the demo.

Note: Since at this point the board is already flashed with the application, the demonstrator can be run without e² studio or the debugger.

4. Functional Overview

4.1. Demonstrator Flow

The software demonstrator includes both **Enrollment** (initial key derivation – S1 below) and **Reconstruction** (upon restart, re-deriving the same key – S2 below).

Steps of both procedures are shown on the display included in the Renesas *Synergy DK-S7G2* kit. The following flow chart depicts the general flow of the demo.

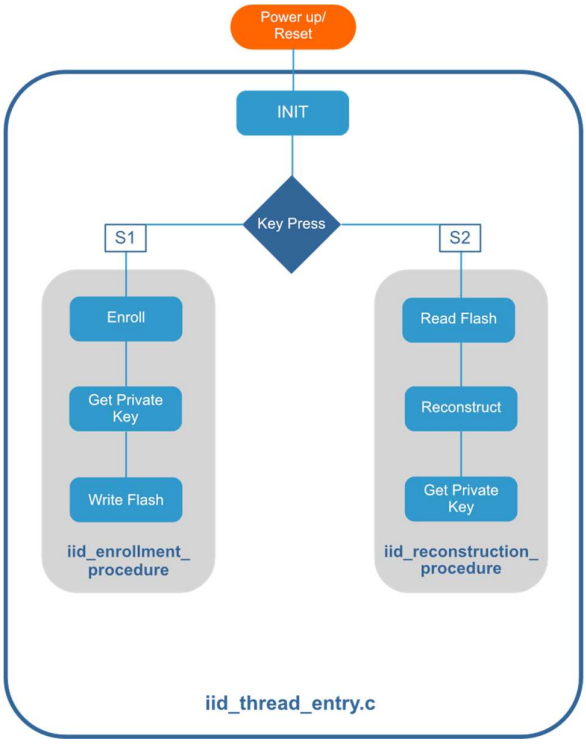


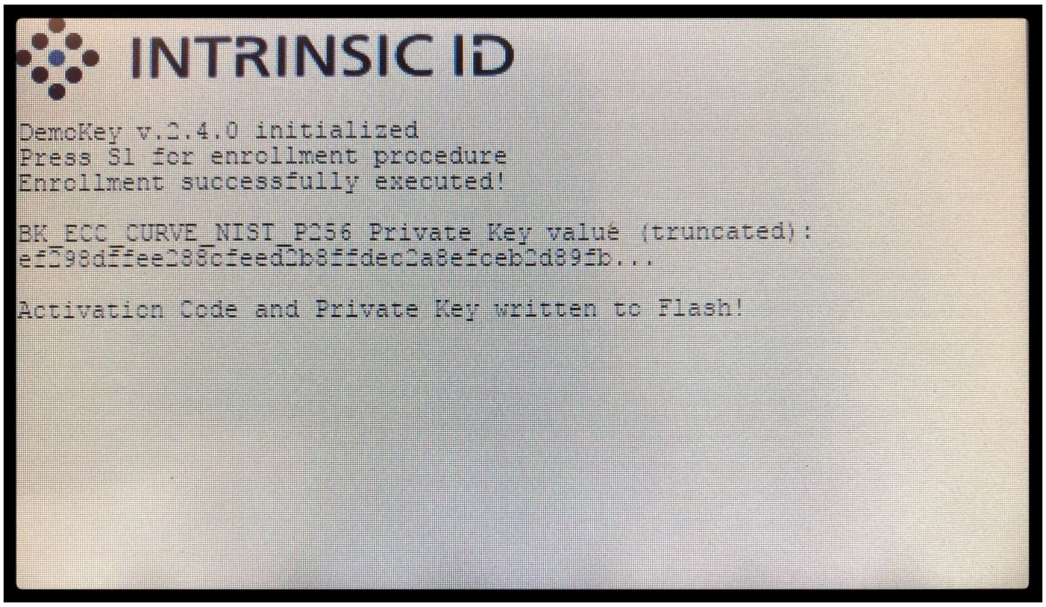
Figure 1 Demo execution flow

Note: The private key is written to flash as part of the **Enrollment** (S1) only for the purpose of comparing the key with the one derived in the **Reconstruction** (S2) step. In normal operation, the private (secret) key should never be stored in flash, since it will be reconstructed by DemoKey as part of the reconstruction (S2) process.

4.1.1. Enrollment Procedure

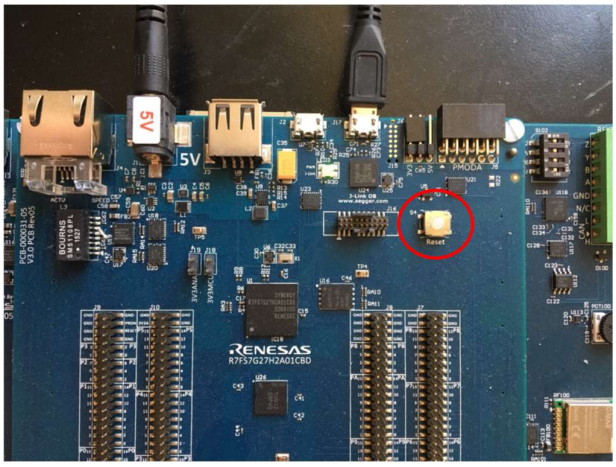
The first time the demo is started after (re)flashing the firmware, the user can only select enrollment. This is done by pressing the **S1** button on the board.

The iid_enrollment_procedure, as depicted in Figure 1, is executed. The result of the enrollment procdured can be seen on the display of the board, depicted in the figure below. The activation code created during enrollment is also used in the reconstruction and is therefore written to flash, this is described in more detail in the *Product Specification*.



4.1.2. Reconstruction Procedure

In order to test and view the reconstruction procedure (or run the enrollment procedure again), reset the board by pressing the **Reset** button on the board or by unplugging and re-inserting the power to the board.



The reconstruction procedure is only selectable when the enrollment procedure has been executed at least once. The reconstruction procedure can be started by pressing the **S2** button on the board.

The iid_reconstruction_procedure, as depicted in Figure 1, is executed. The result of the enrollment procdure can be seen on the display of the board, depicted in the figure below.