



INTRINSIC ID

DemoKey 2.4 Integration Guidelines

**Renesas Synergy
S7G2 MCU**

© 2018 Intrinsic ID B.V. - all rights reserved.
The information contained herein is proprietary to Intrinsic ID B.V.
Receipt of this document does not imply any license under any intellectual property rights of Intrinsic ID.

Version 2.0
Date September 6, 2018
Status Approved
Reference IID-DK2-4-IG

www.intrinsic-id.com



Copyright in this document rests with Intrinsic ID B.V. Reproduction or publication in any medium of this document, in whole or in part, is expressly prohibited without the prior written permission of Intrinsic ID. Intrinsic ID reserves the right to make any changes to this document without prior notice. The contents of this document is provided AS-IS and without any warranties or guarantees as to accuracy or completeness. Receipt or possession of this document conveys no license under any patent or other intellectual property right of Intrinsic ID.

Intrinsic ID®, QuiddiKey®, QuiddiCard®, BroadKey™, DemoKey™, Citadel™, Spartan™, Confidentio™, Fuzzy ID™ and other designated brands included herein are trademarks of Intrinsic ID B.V. All other trademarks are the property of their respective owners.

This product contains code which is copyright 2014 Kenneth MacKay and licensed under the BSD license:

Copyright (c) 2014, Kenneth MacKay
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



1. Introduction

- 2 The purpose of this document is to illustrate the steps required for the integration of Intrinsic
3 ID's DemoKey software library in Renesas Synergy S7G2 MCU.
4 As reference, the *e² studio* toolchain and the Renesas *Synergy DK-S7G2* board are used.
5
6 The reader is expected to be familiar with the *e² studio* environment, e.g. license
7 configuration, and generic techniques of compiling, linking and building software.

2. Build Instructions

2.1. Adding DemoKey Software Library and Include Files

- 10 The first step is adding the DemoKey software library and header files to the target projects
11 workspace. Although the files can be placed anywhere, it is easiest to place them at the root
12 of your workspace next to the project that will make use of DemoKey.

- 14 The following files must be part of the include paths (specified in the compiler options):

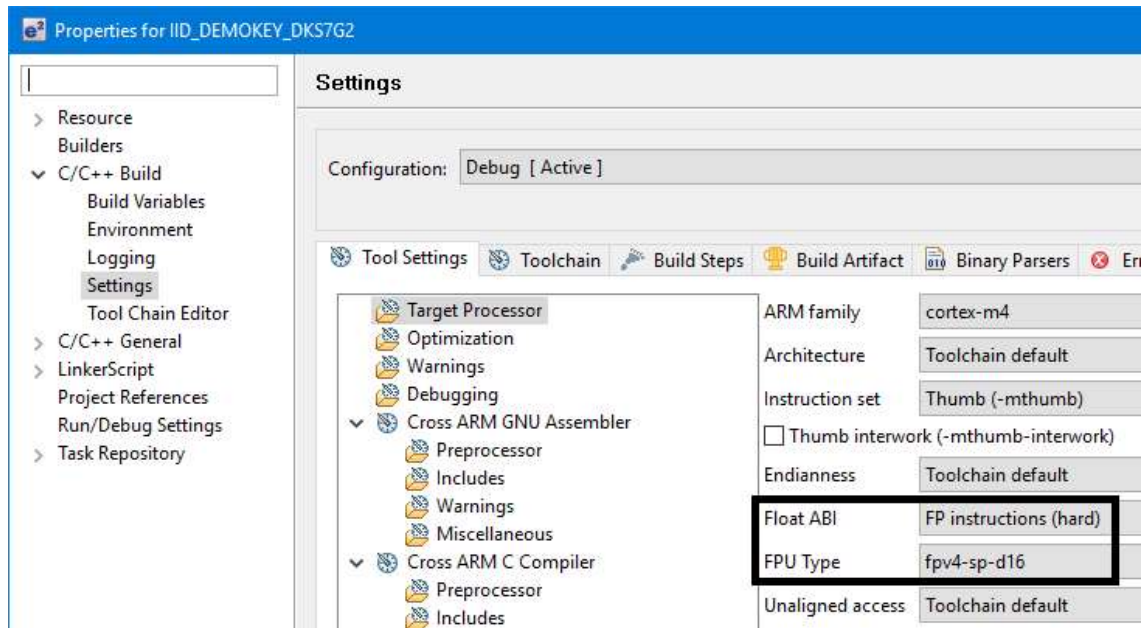
- 15 • iid_configuration.h
- 16 • iid_platform.h
- 17 • iidbroadkey.h
- 18 • iidreturn_codes.h

- 19 These files are located in the **bin/inc** folder of the package.

- 21 The library file, **libS7_DemoKey_256_hardfp_abi.a**, located in the **bin/lib** folder of the
22 package, must be added to the statically linked libraries. The library has been compiled using
23 *Hard Floating Point ABI* support and *fpv4-sp-d16* as *FPU type*, hence the compiler must be
24 set accordingly.

- 26 This can be achieved by the following actions:

- 27 • Right click on the project and select **Properties**.
- 28 • In the Properties dialog, expand **C/C++ Build** and select **Settings**.
- 29 • In the **Tool Settings** tab, make sure **Target Processor** is selected.
- 30 • Set **Float ABI** to **FP instructions(hard)** and **FBU Type** to **fpv4-sp-d16**.



1
2

3 2.2. Adding and Using Designated SRAM Area

4 To make sure that the SRAM used by DemoKey (as SRAM PUF) is not used by any part of
5 the program, it must be excluded from the rest of the available memory. This can be
6 accomplished by placing the SRAM in an additionally created linker segment.

7
8 Listing A.1 shows an example of a linker script file generated by an e² studio project for the
9 Renesas Synergy DK-S7G2 board¹.

10 A.1. Linker Script File for Renesas Synergy DK-S7G2 Board

```

/*
    Linker File for S7G2 MCU
*/

/* Linker script to configure memory regions. */
MEMORY
{
    FLASH (rx)      : ORIGIN = 0x00000000, LENGTH = 0x04000000 /* 4M */
    RAM (rwx)       : ORIGIN = 0x1FFE0000, LENGTH = 0x00A00000 /* 640K */
    DATA_FLASH (rx) : ORIGIN = 0x40100000, LENGTH = 0x00100000 /* 64K */
    QSPI_FLASH (rx) : ORIGIN = 0x60000000, LENGTH = 0x40000000 /* 64M, Change
in QSPI section below also */
    SDRAM (rwx)     : ORIGIN = 0x90000000, LENGTH = 0x20000000 /* 32M */
}

```

¹ When an e² studio project is created, the linker script file **S7G2.ld** is placed under the **script** directory of the project workspace.



[...]

1

2 Any section of uninitialized SRAM can be used as SRAM PUF. For simplicity, here is
3 described the procedure to assign the last 1024 bytes of SRAM to the newly created SRAM
4 PUF segment:

- 5 • Reduce the size of the already existing SRAM segment, e.g. “RAM”, by 1024 bytes.
- 6 • Create a new memory segment, e.g. “PUF”, of 1024 bytes and place it just after the
7 “RAM” segment. (I.e. Assigning it to what was previously the last 1024 bytes of the
8 “RAM” segment)
- 9 • Change the starting Stack Pointer (SP) address to: original SP address – 1024 [*this
10 step is NOT required if the starting SP address is automatically determined by the
11 linker, i.e. for an e² studio-based project*].
- 12 • Define and export a symbol pointing to the “PUF” segment, e.g. “__puf”, which can
13 be referred to in the source code [*this step is not required, as the absolute PUF
14 address can directly be used in the source code, without any symbol*].

15 Listing A.2 shows the outcome of the above described procedure applied to Listing A.1.

16 A.2. Modified Linker Script File for Renesas Synergy DK-S7G2 17 Board

```

/*
    Linker File for S7G2 MCU
*/

/* Linker script to configure memory regions. */
MEMORY
{
    FLASH (rx)      : ORIGIN = 0x00000000, LENGTH = 0x04000000 /* 4M */
    RAM (rwx)       : ORIGIN = 0x1FFE0000, LENGTH = 0x009FC00 /* 639K */
    PUF (rw)        : ORIGIN = 0x2007FC00, LENGTH = 0x0000400 /* 1K */
    DATA_FLASH (rx) : ORIGIN = 0x40100000, LENGTH = 0x00100000 /* 64K */
    QSPI_FLASH (rx) : ORIGIN = 0x60000000, LENGTH = 0x4000000 /* 64M, Change
in QSPI section below also */
    SDRAM (rwx)     : ORIGIN = 0x90000000, LENGTH = 0x2000000 /* 32M */
}

__puf = ORIGIN(PUF);

```

[...]

18

19

20

21

22



- 1 Listing A.3 shows an example of how to use the newly defined “*__puf*” symbol with the
- 2 BROADKEY library.

3 A.3. Example of DemoKey Initialization

```
[...]  
  
extern int __puf;  
  
iid_return_t return_value = bk_init((uint8_t *)&__puf, 1024);  
  
if (IID_SUCCESS != return_value) {  
    /* ... handle error ... */  
}  
  
[...]
```

4