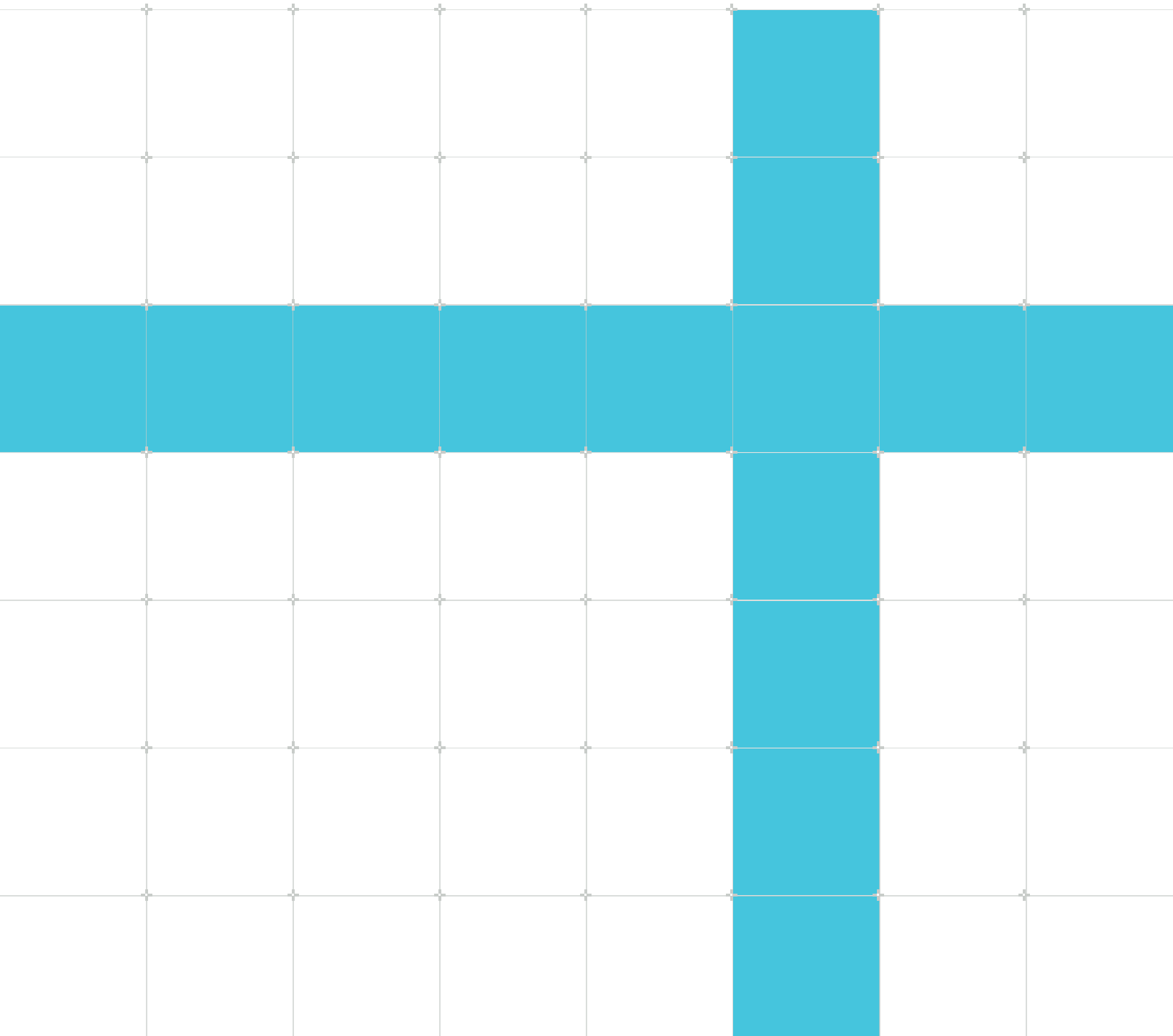
Arm Musca-S1 Getting Started Guide

Version 1.0

Issue 01
110387_0100_01_en



Arm Musca-S1 Getting Started Guide

Release information

Document history

Issue	Date	Confidentiality	Change
0100-01	10 March 2025	Non-Confidential	Initial release

Proprietary Notice

makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

1. Musca-S1 development board

The Musca-S1 test chip board architecture integrates the recommendations of Platform Security Architecture (PSA) using the same subsystem as Musca-A, but with the addition of dual eMRAM and SRAM on-chip secure memory subsystems and is PSA Level 1 and Functional API certified.

The Musca-S1 implementation of SSE-200 subsystem on Samsung Foundry 28FDS is ready to be used to form the core processing element of mainstream IoT devices with secure PSA Root-of-Trust (RoT). Musca-S1 can also be used to prototype secure boot, on-chip storage, execution and network device management through TM-F, Mbed OS and Pelion IoT platform integration.

The following links provide background information about the Musca-S1 development board:

- Arm Demonstrates New IoT Test Chip and Board
- Introducing Industry's-First 28FDS eMRAM IoT Test Chip and Board

,

2. Musca-S1 Frequently Asked Questions

This section provides answers to frequently asked questions about the Musca-S1 development board.

What is the difference between Musca-S1 and other Musca boards?

Musca-S1 is a port of Musca-A1 and B1 secure IoT testchips to the Samsung LNM28FDS process with the addition of a new Code MRAM memory subsystem, Body Biasing and 1.8V IO. Musca-S1 uses the same Musca-A1 Chip Package and is IO pin backward compatibility with Musca-A1.

When will Arm Musca-S1 boards be available under the Musca Loaner Program?

Getting security right in IoT requires a number of different elements, from a Root-of-Trust to a complete set of security features (e.g. isolation, encryption, authentication, secure provisioning, over-the-air update, device monitoring and many more). Designing this from scratch is time-consuming and increasingly more complex. Therefore, starting from a strong foundation makes designing security into your products much easier. So request your Musca-S1 Loaner board to learn how to make eMRAM-enabled IoT security easier and faster.

Musca-S1 boards can be requested under the Arm Musca Loaner program on the Musca-S1 Developer's page. To receive your Musca-S1 loaner board, you will be required to sign an Arm End User License Agreement (EULA)

How do I get the Blinky example supplied on the board?

The board is delivered programmed with a simple Blinky example. The source code for this example is not publicly available. However a copy of the binary file can be found in Musca-S1 firmware update and QSPI/MRAM boot recovery with instructions on how to load it onto the board.

Can I use Arm Keil MDK Microcontroller Development Kit tools with the Musca-S1 Board?

Yes. The Arm® Keil Microcontroller Development Kit (MDK) tools include C/C++ compilers, integrated development environments, RTOS, middleware, as well as debug adapters and evaluation boards for Arm Cortex®-M based devices, including the Arm Keil MDK Musca-S1 Software Pack (SP), also known as the DFP (Device Family Pack).

Keil MDK comes in four editions and can be compared and downloaded from The Arm Keil Website.

The latest version of Keil MDK is v5.30.

How do I download Arm Keil MDK Musca-S1 Board Software Pack (SP/DFP)?

A Device Family Pack (DFP) is used to extend the initial Package Description (PDSC) file to add SVD files, Flash algorithms, and device-related software components such as system and HAL files, including examples.

The Arm Keil MDK Musca-S1 SP/DFP can be downloaded from the Keil Musca-S1 page.

The latest version of Keil MDK Musca-S1 DFP is ARMMusca-S1_DFP.1.1.0.

Is Musca-S1 Supported in Trusted Firmware-M?

Yes. Trusted Firmware-M provides a reference implementation of secure world software for Armv8-M. It provides SoC developers and OEMs with a reference trusted code base complying with the relevant Arm specifications, including support for Armv8-M / Trusted Firmware-M (TF-M) and relationship with Platform Security Architecture (PSA).

PSA provides a common security foundation for the whole IoT ecosystem. It includes many elements, including architecture specifications and threat models. An important part of PSA is open-source firmware. This is available in the form of Trusted Firmware-M for Arm Cortex-M23 and Arm Cortex-M33 processors, which use Arm TrustZone technology.

The Trusted Firmware-M codebase for Musca-S1 is available as a reference implementation of PSA at the Trusted Firmware Website.

3. Musca-S1 links and resources

This page provide links to resources available for Musca-S1 IoT development platform.

- Technical Reference Manual (TRM)

  Available on Arm Developer
- Technical Overview

  Available on Arm Developer

4. Musca-S1 firmware update and QSPI/MRAM boot recovery

The latest DAPLink firmware for the Musca-S1 board can be found in the Downloads section of this guide, accessible from the Arm Developer page toolbar.


Note

Please ensure the DAPLink board firmware has been updated to the latest version before attempting Drag-and-Drop updates to the boards software. DAPLink firmware S1 DUAL v1.9 supports both QSPI and MRAM boot on Musca-S1 boards.

To update the DAPLink firmware please follow these steps (as described in the board TRM):

1. Press and hold the ISP button while powering up the board using the USB lead.
2. Delete file firmware.bin that appears in the CRP DISABLD USB drive.
3. Copy DAPLink_S1_DUAL_v1_9.bin, or a later version, to the CRP DISABLD drive.
   - From a Windows System you can simply Drag and Drop the file
   - On Linux/Mac OS please use the following command:

     ```
     dd if={new_firmware.bin} of=/Volumes/CRP\ DISABLD/firmware.bin conv=notrunc
     ```
4. Power cycle the board using the USB lead. Do not press the ISP button during the power cycle.

The `musca-s1_blinky.zip` file in the Downloads section contains blinky application files for running from QSPI and MRAM. To update the application image, perform the following steps:

1. Ensure that the boot switch is set to QSPI or eMRAM as required.
2. Power up the board by connecting the USB lead and pressing the PBON button.
3. Copy the required software image onto the MBED drive.
4. Power cycle the board or press the nSRST button to reset the system and boot from the new QSPI or eMRAM software image.

The following files are available in the Downloads section of this guide, accessible from the Arm Developer page toolbar:

- `DAPLink_S1_DUAL_v1.9.zip`
- `Musca_s1_blinky.zip`

5. Board automatic run on power up

By default, power-on of the Musca-S1 board requires the user to press the PBON Button.

This can be a problem for remote power recyling of a Musca-S1 within a Board Farm.

The Downloads section of this guide contains zipped configuration files that can set or override automatic power recycling for Musca-S1 boards.

Once unzipped, the two files are fairly straightforward. They allow the board to switch on as soon as the USB power is applied, alternatively when the user presses the PBON button:

- `auto_pwr.cfg` - turn power ON when board is plugged in
- `hard_pwr.cfg` - turn power ON when push button pressed (default mode)

In order to implement either configuration, just drag and drop the appropriate unzipped configuration file onto the board Mass Storage Device.

The following files are available in the Downloads section of this guide, accessible from the Arm Developer page toolbar:

- `auto_pwr.zip`
- `hard_pwr.zip`

6. Keil Support for Musca-S1

The Musca-S1 Keil pack contains a device family package, the Musca-S1 Keil Device Family Pack.

Version 1.2.0

- Added TF-M platform components.
- Refactored TrustZone setup.
- Added MPC LUT Init debug sequence.
- Updated orderForm tag in board element
- Updated supportContact

Version 1.1.0

- Fixed partition header file with missing FPU settings.
- Added support contact and board documentation information.
- Updated BM_low_power example
- CMSIS-Driver included:
  - Flash eMRAM 1.1.0
- UART driver updated:
  - Added support for FIFO handling and ARM_USART_EVENT_RX_TIMEOUT signal event
- CMSIS-Driver included:
  - I2C: initial version 1.0.0.
- Example projects cleaned.

Version 1.0.0

- Initial public release of this pack.
- CMSIS Startup: for Arm Compiler 6.
- CMSIS-Driver included:
  - SPI: initial version 1.0.0.
  - USART: initial version 1.0.0.
- CMSIS-Zone resource file: Device/Zone/Musca-S1.rzone.

The Musca-S1 Board Support Pack is available here.

Further Example projects that run on Musca-S1, including examples to connect to AWS, Google and Microsoft Azure, are available from the Keil Website.

7. Musca-S1 Software Examples

Three software example projects are provided with the Keil Musca-S1 Board Support Pack version 1.1.0.

Further information about these examples can be found on the following pages:

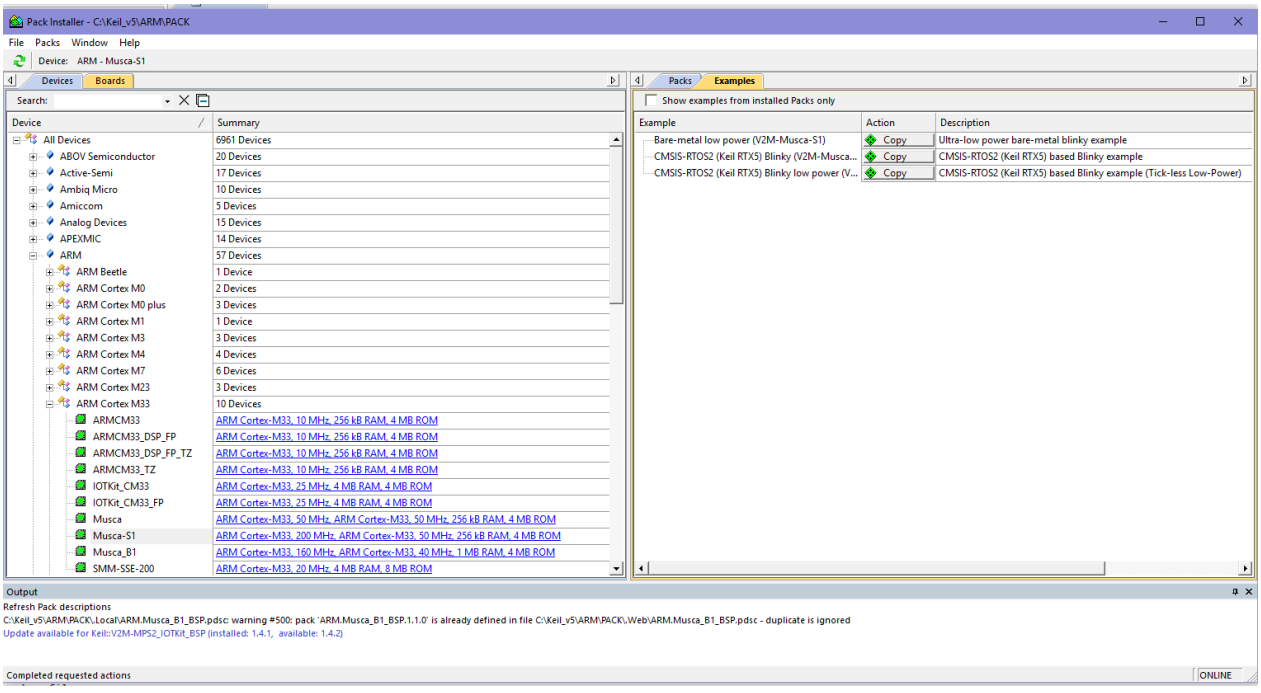
- Bare-metal low power example
- CMSIS-RTOS2 Blinky example
- CMSIS-RTOS2 Blinky low power example

Accessing the Examples

When you have Musca-S1 packs installed in Keil MDK, you can make copies of the example projects using MDK Pack Installer. From the Pack Installer under the Devices tab in the left hand pane select ARM -> ARM Cortex-M33 -> Musca_S1.

In the Right hand pane of the Pack Installer select the Examples tab. There you will be presented with the examples available for Musca-S1. You can press the “Copy” button for the examples to create a copy of the example in a folder of your choosing.

Figure 7-1: Accessing the Musca-S1 examples



8. Bare-metal low power example

This 'Blinky' project is a simple ultra low power bare-metal example for the Arm 'Musca-S1 CM33 #0' device on the Arm 'V2M-Musca-S1' evaluation board. The example is compliant to CMSIS.

Example functionality:

- Clock Settings:
  - XTAL = 50.00 MHZ
  - SYSCLK = 50.00 MHz
- one LED is flashing in three colors during different program phases:
  - Red - code execution in SRAM
  - Green - code execution in eMRAM
  - Yellow - Ultra-Low-Power (ULP) mode

The Blinky program is available in different targets

- Debug (eMRAM) - main code and data are in eMRAM
- Debug (SRAM) - main code and data are in SRAM

In both targets the code for ultra-low power operation is kept in SRAM area that is not being turned off during ULP.

The application executes simple counting in SRAM and eMRAM. Timing and current consumption can be compared in SystemAnalyzer using ULINKplus.

Debug (eMRAM)	Debug (SRAM)
code: eMRAM	code: SRAM
ulp code: SRAM	ulp code: SRAM
data: eMRAM	data: SRAM

How to use it:

- Configure the board's BOOT switch to "MRAM".
- Build the Debug (SRAM) target.
- Go to debug to load the application into SRAM
- Run debug. LED should show three colors interchangeably.
- Exit debug without stopping.
- Enter "Energy measurement without debug" mode
- See power readings in the System Analyzer window
- Repeat the previous steps with the taget Debug (MRAM) in this case code and data are kept in MRAM (except of SRAM count and ULP parts)

9. CMSIS-RTOS2 Blinky example

This 'Blinky' project is a simple CMSIS-RTOS2 Keil RTX5 based example for ARM 'Musca-S1 CM33 #0' device using ARM 'V2M-Musca-S1' Evaluation Board.

Compliant to Cortex Microcontroller Software Interface Standard (CMSIS).

Example functionality:

- Clock Settings:
  - XTAL = 50.00 MHZ
  - SYSCLK = 50.00 MHz
- one LED is blinking with a fixed speed. No sleep used.

The Blinky program is available in different targets:

- Debug:
  - Compiler optimization Level 1
  - Keil RTX5 variant 'Source'

Debug (eMRAM)	Debug (SRAM)
code: eMRAM	code: SRAM
data: SRAM	data: SRAM

Configure the board's BOOT switch to "MRAM".

10. CMSIS-RTOS2 Blinky low power example

This 'Blinky' project is a simple CMSIS-RTOS2 Keil RTX5 based example for ARM 'Musca-S1 CM33 #0' device using ARM 'V2M-Musca-S1' Evaluation Board. Compliant to Cortex Microcontroller Software Interface Standard (CMSIS).

Example functionality:

- Clock Settings:
  - XTAL = 50.00 MHZ
  - SYSCLK = 50.00 MHz
- one LED is flashing with a fixed speed.

The Blinky program is available in different targets:

- Debug:
  - Compiler optimization Level 1
  - Keil RTX5 variant 'Source'

```
Debug (eMRAM)
--------------
code: eMRAM
data: eMRAM
idle: eMRAM off
        SRAM off
        + sleepdeep
```

Configure the board's BOOT switch to "MRAM".

How it works:

- SRAM3 is used as 'ULP RAM' which means that it is active during power down phase.
- Code and Data for ulp_* files are placed in ULP RAM because code is used to enter and leave power down phase.
- Interrupt vecors are copied to ULP RAM because S32K Timer interrupt is used for wake up.
- Main stack is placed in ULP RAM.
- Process stack is placed in ULP RAM during pwer down phase.
- OS Idle Thread checks if system can enter power down phase.

Appendix A Revisions

The revisions tables describe the technical changes between released issues of this document.

Table A-1: Issue 01

Change	Location
Initial release	-