APPLICATION NOTE

IAR Example with Execution in External Flash Using MSPI XIP Mode

A-SOCAP3-ANGA05EN v1.1

Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING WITHOUT LIMITATION CONSEQUENTIAL OR INCIDENTAL DAMAGES. "TYPICAL" PARAMETERS WHICH MAY BE PROVIDED IN AMBIQ MICRO DATA SHEETS AND/OR SPECIFICATIONS CAN AND DO VARY IN DIFFERENT APPLICATIONS AND ACTUAL PERFORMANCE MAY VARY OVER TIME. ALL OPERATING PARAMETERS, INCLUDING "TYPICALS" MUST BE VALIDATED FOR EACH CUSTOMER APPLICATION BY CUSTOMER'S TECHNICAL EXPERTS. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.

Revision History

Revision	Date	Description
1.0	April 11, 2022	Initial release
1.1	January 3, 2023	Updated document part number

Reference Documents

Document ID	Description

Table of Contents

Introduction

This release of the Apollo3 SDK provides two new examples that can be use to demonstrate a work flow for installing and executing a portion of customer code from an external Flash device using the Apollo3 MSPI in XIP mode. These examples are in:

▪ /boards/apollo3_eb/examples/mspi_flash_loader

▪ /boards/apollo3_eb/examples/mspi_prime

This application note walks the reader through the steps required to produce an example with code located in both internal and external flash, then split the resulting binary into two binaries:

▪ **mspi_prime_internal** is the binary loaded into the Apollo3 internal flash

▪ **mspi_prime_external** is the binary loaded into the external flash and accessed over MSPI XIP

1.1   Assumptions

This document assumes the following:

Cygwin or equivalent with python3 is installed in user's environment

IAR Example with Execution in External Flash Using MSPI XIP Mode
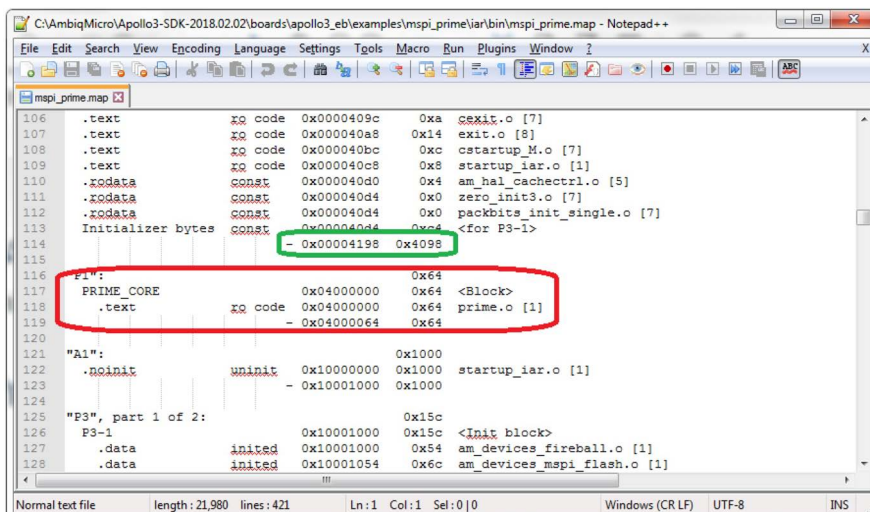
Use the following procedure:

1. Change directory to **/boards/apollo3_eb/examples/mspi_prime**.

2. Copy the **mspi_prime.icf** file their down into the **/iar** directory.

> **NOTE:** The SDK release builder populates a default **mspi_prime.icf** file. This file does not relocate the **prime.o** object into the external flash. It is instructive to compare these two files to note the differences.

3. Open IAR and rebuild the **mspi_prime** example with the new ICF.

4. Check the **/iar/bin/mspi_prime.map** file to make sure the **prime.o .text** segment is located in the external flash address range as follows.
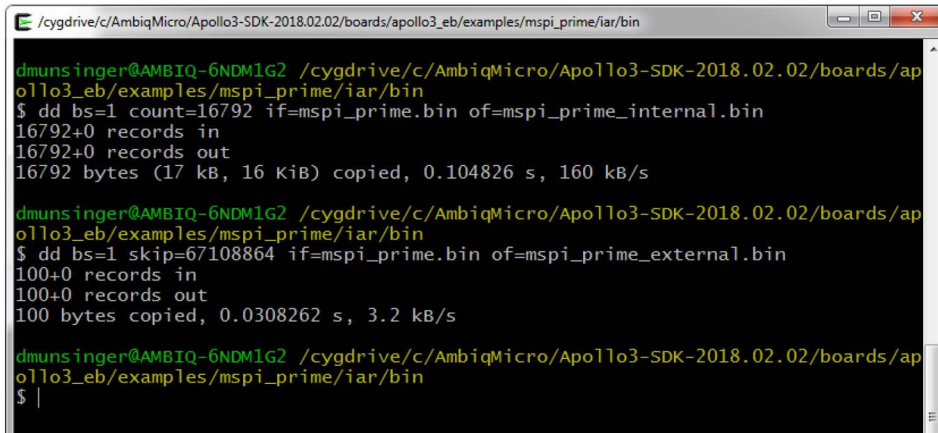
> **NOTE:** that this example only relocates the **.text** segment of the **prime.c** program. It is recommended at this time that **.rodata** be left in internal flash.

5. Use the linux **data duplicator** or **dd** command to separate the **/iar/bin/mspi_prime.bin** into the internal and external segments (below). The first **dd** command separates the internal flash segment of the **mspi_prime** example. The second **dd** command separates the external flash segment of the **mspi_prime** example.

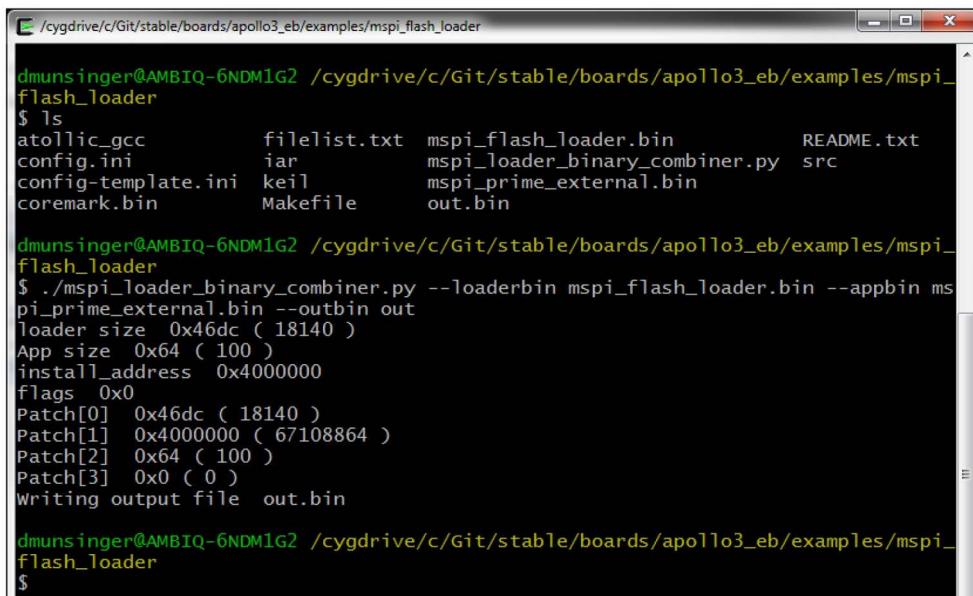> **NOTE:** The value 16792 is equivalent to the 0x4198 (see above) and is the ending location of the **mspi_prime** internal flash segment and 67108864 is 0x04000000.

```
/cygdrive/c/AmbiqMicro/Apollo3-SDK-2018.02.02/boards/apollo3_eb/examples/mspi_prime/iar/bin

dmunsinger@AMBIQ-6NDM1G2 /cygdrive/c/AmbiqMicro/Apollo3-SDK-2018.02.02/boards/ap
ollo3_eb/examples/mspi_prime/iar/bin
$ dd bs=1 count=16792 if=mspi_prime.bin of=mspi_prime_internal.bin
16792+0 records in
16792+0 records out
16792 bytes (17 kB, 16 KiB) copied, 0.104826 s, 160 kB/s

dmunsinger@AMBIQ-6NDM1G2 /cygdrive/c/AmbiqMicro/Apollo3-SDK-2018.02.02/boards/ap
ollo3_eb/examples/mspi_prime/iar/bin
$ dd bs=1 skip=67108864 if=mspi_prime.bin of=mspi_prime_external.bin
100+0 records in
100+0 records out
100 bytes copied, 0.0308262 s, 3.2 kB/s

dmunsinger@AMBIQ-6NDM1G2 /cygdrive/c/AmbiqMicro/Apollo3-SDK-2018.02.02/boards/ap
ollo3_eb/examples/mspi_prime/iar/bin
$ 
```

6. Copy the **mspi_prime_external.bin** to the **/boards/apollo3_eb/examples/mspi_flash_loader** directory.

7. Copy the IAR binary for the **mspi_flash_loader** example from the **/iar/bin** directory to the parent directory.

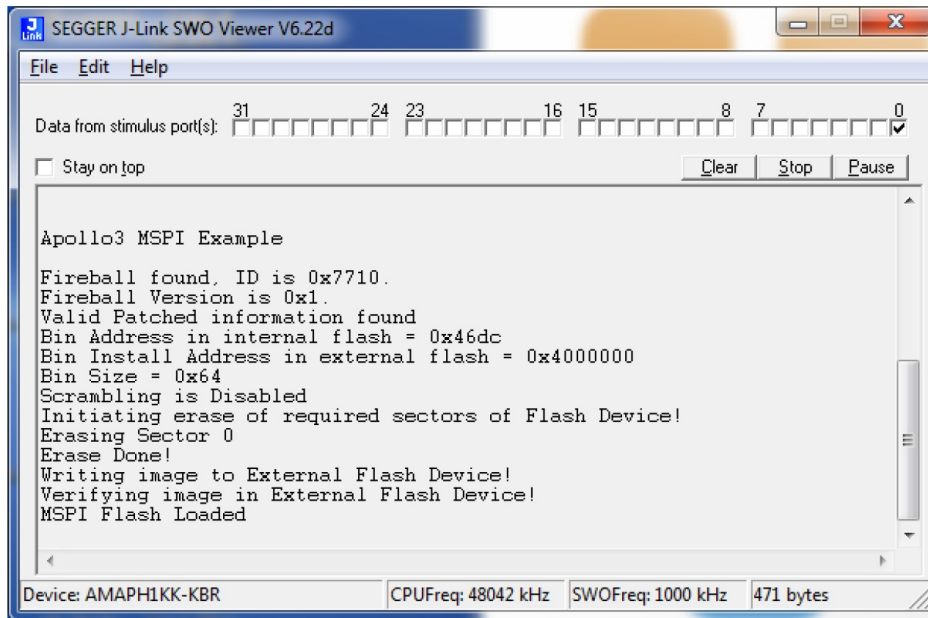8. Create the loader program using the python script (**mspi_loader_binary_combiner.py**) as follows:

```
/cygdrive/c/Git/stable/boards/apollo3_eb/examples/mspi_flash_loader

dmunsinger@AMBIQ-6NDM1G2 /cygdrive/c/Git/stable/boards/apollo3_eb/examples/mspi_
flash_loader
$ ls
atollic_gcc          filelist.txt   mspi_flash_loader.bin          README.txt
config.ini           iar            mspi_loader_binary_combiner.py  src
config-template.ini  keil           mspi_prime_external.bin
coremark.bin         Makefile       out.bin

dmunsinger@AMBIQ-6NDM1G2 /cygdrive/c/Git/stable/boards/apollo3_eb/examples/mspi_
flash_loader
$ ./mspi_loader_binary_combiner.py --loaderbin mspi_flash_loader.bin --appbin ms
pi_prime_external.bin --outbin out
loader size  0x46dc ( 18140 )
App size  0x64 ( 100 )
install_address  0x4000000
flags  0x0
Patch[0]  0x46dc ( 18140 )
Patch[1]  0x4000000 ( 67108864 )
Patch[2]  0x64 ( 100 )
Patch[3]  0x0 ( 0 )
Writing output file  out.bin

dmunsinger@AMBIQ-6NDM1G2 /cygdrive/c/Git/stable/boards/apollo3_eb/examples/mspi_
flash_loader
$
```
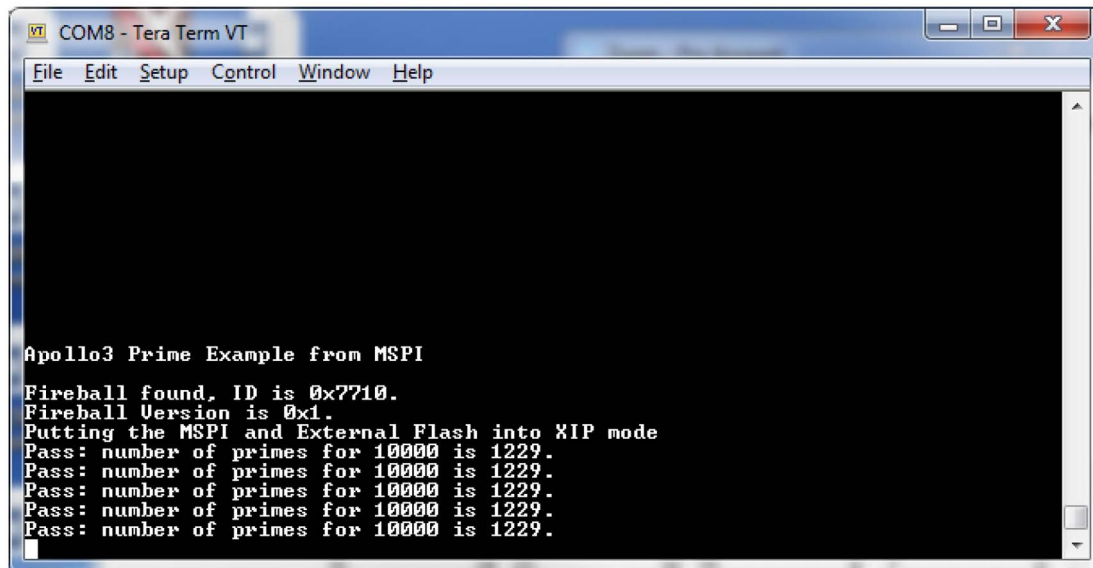
9. Use the J-Link tools or IAR to run the **out.bin** binary. This will load the **mspi_prime** external flash segment as follows:

10. Use the J-Link tools to load and run the **/boards/apollo3_eb/examples/mspi_prime/ mspi_prime_internal.bin** on the target. The output to the UART0 (115200bps) should appear as follows:

ambiq

6500 River Place Boulevard, Building 7, Suite 200, Austin, TX 78730
www.ambiq.com
sales@ambiq.com
+1 (512) 879-2850

A-SOCAP3-ANGA05EN v1.1
January 2023