



ModusToolbox™



WICED Execute-in-Place (XIP) Application Support Guide

Associated Part Family: CYW20819
Document Number. 002-22870 Rev. **

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Contents

About This Document	3
Purpose and Audience	3
Scope	3
Acronyms and Abbreviations	3
IoT Resources and Technical Support	3
1 Introduction	4
1.1 Features Overview	4
2 Flash Layout and Compilation Command	4
Document Revision History	6

About This Document

This document explains how to use the Execute In-Place (XIP) feature on CYW20819 based platforms.

Purpose and Audience

This document is intended for application developers creating and testing designs based on Cypress WICED Studio Software Development Kit for CYW20819 based platforms.

Scope

The scope of this document is to provide information to the developers, so that they can use the XIP feature on CYW20819 based platforms.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

For a comprehensive list of acronyms and other terms used in Cypress documents, go to www.cypress.com/glossary.

IoT Resources and Technical Support

Cypress provides a wealth of data at <http://www.cypress.com/internet-things-iot> to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (<http://community.cypress.com/>).

1 Introduction

The Execute-in-Place (XIP) feature allows you to enable applications to run in-place from on-chip flash (OCF) in CYW20819.

1.1 Features Overview

The XIP feature implements support for building an application to run in-place from OCF. This feature is helpful for applications with large code size and limited SRAM constraints. By placing the application and the profile library code in flash, the application can save SRAM space. The .text section and .rodata section from the application and the profile libraries execute from flash. The remaining sections are loaded to SRAM. The patches will be executed from patch RAM.

The flash start address to place the XIP section is calculated by adding PLATFORM_APP_SPECIFIC_DS_LEN (default set in *mainapp.mk*) to ConfigDSLocation, from the platform *btp* file found in the *platform* folder.

You can modify the XIP section length (PLATFORM_APP_SPECIFIC_XIP_LEN) by updating the platform makefile if the section size requirement is more than the default. If a larger section size is required, the compilation will fail with the following error message:

```
"region `xip_section' overflowed by xxxx bytes"
```

This error message indicates by how much the default length should be increased.

Executing the code from flash will impact the speed and power. Therefore, do not place time critical functionality (such as interrupt service routines) in the XIP section. Place the part of the application code in the .text section using the section attribute as below:

```
__attribute__((section(".text"))) void foobar(void)
{
    0
}
```

2 Flash Layout and Compilation Command

FLASH layout with XIP

Unused, or reserve for DS2 (OTA download)	0x540000
XIP	ConfigDS2Location)
DS	0x501400 + PLATFORM_APP_SPECIFIC_DS_LEN
VS	0x501400 (ConfigDSLocation)
SS	0x500400 (ConfigVSLocation)
	0x500000 (ConfigSSLocation)

Figure 2-1. Flash Layout with XIP Image

In the flash layout with XIP image:

- SS = Static Section, where BD_ADDR and location of other sections are stored
- VS = Volatile Section, where Link keys, app NV data are stored
- DS/DS2 = Dynamic Section, where patches, configuration, and application code are stored. There are two sections, so that you can ping and pong for OTA upgrade
- APP XIP = XIP Section, where the application code intended to run from OCF

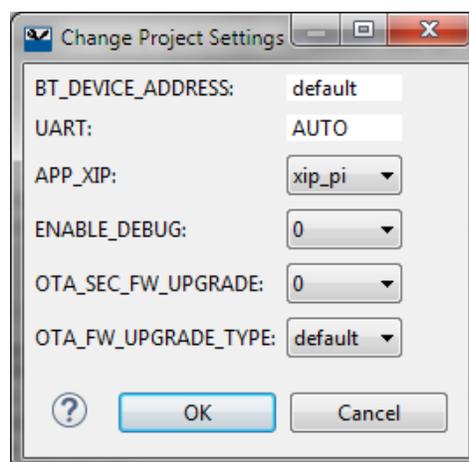
When building an application, there are two choices for XIP: xip and xip_pi, which add position independence. The trade-offs for the two settings are overall image size and compatible Over The Air (OTA) update methods. The xip_pi build results in a larger image because tables and code are added to support position-independent calls to ROM functions or accesses to RAM data. The position independence allows the load location of the image to be swapped, so fail-safe OTA firmware updates that always keep a known-good application image in On Chip Flash can be used. The drawback is that less than half of the On Chip Flash is available, limiting the overall xip_pi image size. The xip option is built to run from one load location and optimizes size. The load location cannot be swapped to support OCF-only firmware updates. The recommended OTA firmware update method for xip uses external flash to temporarily store the image during download and verification. This means the xip method can support images larger than half the OCF size, but requires external serial flash for storage.

To enable or disable position-independent XIP for the desired application, update the make command so that APP_XIP=<value> is included where <value> can be xip_pi or xip. If unspecified, the default is xip_pi.

For example:

```
make -f modus.mk APP_XIP=xip_pi program
```

To set the same option from the ModusToolbox IDE, right-click the project and select **Change Application Settings**:



Document Revision History

Document Title: WICED Execute-in-Place (XIP) Application Support Guide

Document Number: 002-22870

Revision	ECN	Issue Date	Description of Change
**	6488609	02/19/2018	Initial release

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

Cypress Developer Community

[Community](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#)
| [Components](#)

Technical Support

cypress.com/support



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.