arm

Running software on your target

Version 1.0

Running software on your target

Release information

Document history

Issue	Date	Confidentiality	Change
0100-01	3 April 2019	Non-Confidential	First release

Proprietary Notice

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 the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at https://www.arm.com/company/policies/trademarks.

Copyright © 2019 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

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. Overview

Once you have developed your software, you will probably want to run it on some kind of hardware device.

There are a number of different mechanisms for transferring your executable image to a target device, depending on the type of hardware you are using.

- If you are targeting a development board with a built-in USB programming interface, you can transfer your executable image by simply dragging and dropping from your host to the target.

- For devices with Joint Test Action Group (JTAG) or similar interfaces, a ULINK debug adapter lets you download programs to your target hardware and program flash memory, and debug the operation of your code.

- If you are using an Integrated Development Environment (IDE) like Arm DS-5 or Keil MDK, you can program and debug your device seamlessly using the same environment you use to write your code.

2. Transfer your code to a hardware target

There are a number of different mechanisms for transferring your executable image to a target device, depending on the type of hardware that you are using.

- The Getting your program on your board tutorial provides information for Mbed users.
- The Blinky Project with MDK-ARM video tutorial shows how to use MDK with an STM32F4 Discovery board to get your code running on a hardware device.
- Getting started with DS-MDK includes examples of developing software for heterogeneous systems containing both Cortex-A and ARM Cortex-M processors, using ULINKpro. The examples use flash memory to transfer code to the Cortex-A processor, and U-boot to transfer code to the Cortex-M processor.
- The Arm DS-5 Debugger Linux kernel debug video tutorial guides you through debugging a sample Linux kernel using the DS-5 Debugger, a DSTREAM unit and a BeagleBoard.

3. Programming flash memory

Flash is a common type of non-volatile memory that is used to store code and data. Arm Development Studio includes platform entries for common development boards, which include a flash definition section. This section defines one or more areas of flash, each with its own flash method and configuration parameters.

- About file-based flash programming explains the basics of flash programming with Arm Development Studio.
- Arm Development Studio supports flash programming for CMSIS pack connections, including the ability to flash multiple images on to your target.
- If necessary, you can create new flash methods using Jython.
- If you're using the Juno development board, the Juno ARM Development Platform Getting Started Guide provides information about flash memory programming.
- Arm Development Studio provides a number of commands relating to flash memory programming.