FDI

WARNING: Code Read Protection (CRP) in LPC Devices

FDI

NXP LPC MCUs can use CRP to protect your Intellectual Property (IP) when it is in the MCUs memory. If you wish to use CRP, please refer to NXP's application notes, User's Manuals, or the community forums.

CRP is controlled by values written into specific memory locations of the MCU. When CRP is not used the CRP memory locations are all 0xFF's (not protected). All FDI projects are written utilizing no CRP.

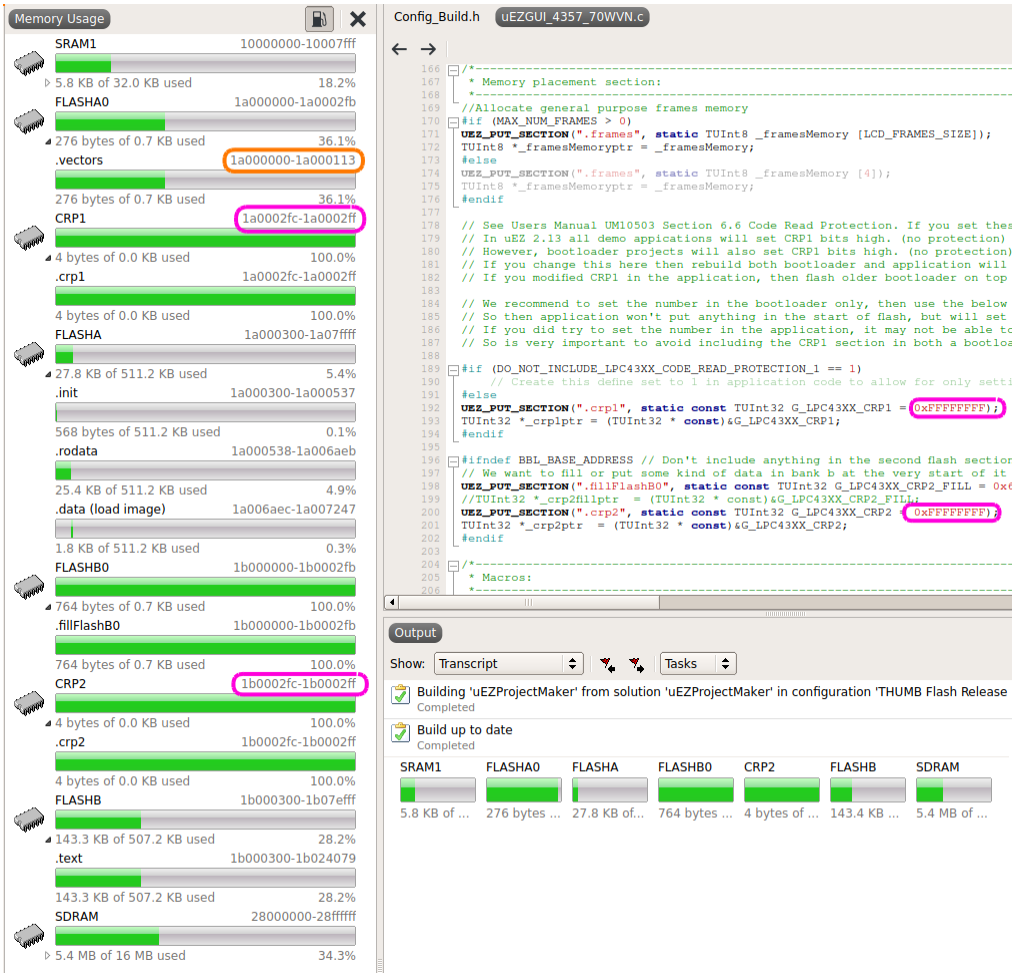
Memory section .crp1 and when needed .crp2 are each setup with 4 bytes, initialized to 0xFFFFFFFF (no CRP). We do not recommend changing the memory sections in our product demos as they are set up for the available memory of the uEZGUI. As an example, please review the Memory Usage Map and Memory placement section in uEZGUI-4357-70WVN.c.

Important Note – You should take particular care when modifying the value placed in the CRP word, as some CRP settings can disable some or all means of access to your MCU (including debug). Before making use of CRP, you are strongly advised to read NXP's documentation on this functionality. You can access the documentation for your MCU at Microcontrollers and Processors |NXP

See below for two examples of a "safe" flash mapping for the LPC4357.

- This example shows a typical project that uses all available flash with no bootloader usage.
  - Note in orange that vectors start at address 0x1A00 0000. If vectors are moved out of this region the unit won't boot up by itself.
  - Note in purple that the next region is the CRP1 region. This corresponds to the const G_LPC43XX_CRP1. This region should always be present in the normal project. If the circled number in the uEZGUI-4357-xxWVN.c platform file is changed, it will change the number in flash. Unlike other MCUs, SEGGER J-Link has no protection mechanism for these LPC families. It will write any value to flash without warning the user first.
  - After the CRP1 region, the remaining 511.2KB of Flash Bank A is available for software and can be populated in any order with read only code or data.
  - Following Flash Bank A is 3 more regions for Flash Bank B. These include the small beginning area, the second CRP (called CRP2 here), and the general purpose FlashB. These follow the same rules as in Flash Bank A. Const G_LPC43XX_CRP2 controls the second CRP2 value.
    - If Flash Bank B is not needed in the application, then all 3 parts can be excluded safely. Then no JTAG programming will occur in that flash bank. By default, FDI puts the .text section in the Flash Bank B, so the structure is always preserved and a large application can be developed before needing to have to put executable code into external QSPI flash.

WARNING: Code Read Protection (CRP) in LPC Devices



WARNING: Code Read Protection (CRP) in LPC Devices

In the **BELOW** example, this typical project excludes the first 64KB of flash, reserved for the uEZ Bootloader. It is up to the customer to adjust this mapping for use with an IAR project, if IAR is to be used instead of Crossworks.

- In this example the starting address must be 0x1A010100. (The image header, not shown, will occupy 0x1A010000 to 0x1A010100)
- The project's #defines and Crossworks variables are set up in such a way that the first CRP region is now excluded. If this region was present, the debugger would erase your bootloader since it would fall into the first 64KB.
- When using the uEZ SplitHexToBins application it is very important to handle Flash Bank B correctly. The very beginning (0x1B000000) of Flash Bank B must be included in the .hex file output or the resulting split files will not be at the correct starting address. This can cause random .text data to get programmed into CRP2, possibly locking you out from debugging or programming the MCU. For this reason, a .fillFlashB0 region was added to always fill FlashB0. It is safe to put code into FlashB0, reducing or eliminating the need for the fill. For the Rowley Crossworks fill command to work correctly, a dummy piece of read-only data needs to be included in this section.
- In the FDI Platform files this is the G_LPC43XX_CRP2_FILL global variable.