

WARNING: Code Read Protection (CRP) in LPC Devices

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

Memory Usage

Memory Type	Address Range	Used	Percentage
SRAM1	10000000-10007fff	5.8 KB of 32.0 KB	18.2%
FLASHA0	1a000000-1a0002fb	276 bytes of 0.7 KB	36.1%
vectors	1a000000-1a000113	276 bytes of 0.7 KB	36.1%
CRP1	1a0002fc-1a0002ff	4 bytes of 0.0 KB	100.0%
.crp1	1a0002fc-1a0002ff	4 bytes of 0.0 KB	100.0%
FLASHA	1a000300-1a07ffff	27.8 KB of 511.2 KB	5.4%
.init	1a000300-1a000537	568 bytes of 511.2 KB	0.1%
.rodata	1a000538-1a006aeb	25.4 KB of 511.2 KB	4.9%
.data (load image)	1a006aec-1a007247	1.8 KB of 511.2 KB	0.3%
FLASHB0	1b000000-1b0002fb	764 bytes of 0.7 KB	100.0%
.fillFlashB0	1b000000-1b0002fb	764 bytes of 0.7 KB	100.0%
CRP2	1b0002fc-1b0002ff	4 bytes of 0.0 KB	100.0%
.crp2	1b0002fc-1b0002ff	4 bytes of 0.0 KB	100.0%
FLASHB	1b000300-1b07efff	143.3 KB of 507.2 KB	28.2%
.text	1b000300-1b024079	143.3 KB of 507.2 KB	28.2%
SDRAM	28000000-28ffffff	5.4 MB of 16 MB	34.3%

Config_Build.h uEZGUI_4357_70WVN.c

```

166  /*
167  * Memory placement section:
168  */
169  //Allocate general purpose frames memory
170  #if (MAX_NUM_FRAMES > 0)
171  uEZ_PUT_SECTION(".frames", static TUInt8 _framesMemory [LCD_FRAMES_SIZE]);
172  TUInt8 *_framesMemoryptr = _framesMemory;
173  #else
174  uEZ_PUT_SECTION(".frames", static TUInt8 _framesMemory [4]);
175  TUInt8 *_framesMemoryptr = _framesMemory;
176  #endif
177
178  // See Users Manual UM10503 Section 6.6 Code Read Protection. If you set these
179  // In uEZ 2.13 all demo applications will set CRP1 bits high. (no protection)
180  // However, bootloader projects will also set CRP1 bits high. (no protection)
181  // If you change this here then rebuild both bootloader and application will
182  // If you modified CRP1 in the application, then flash older bootloader on top
183
184  // We recommend to set the number in the bootloader only, then use the below
185  // So then application won't put anything in the start of flash, but will set
186  // If you did try to set the number in the application, it may not be able to
187  // So is very important to avoid including the CRP1 section in both a bootla
188
189  #if (DO_NOT_INCLUDE_LPC43XX_CODE_READ_PROTECTION_1 == 1)
190  // Create this define set to 1 in application code to allow for only setti
191  #else
192  uEZ_PUT_SECTION(".crp1", static const TUInt32 G_LPC43XX_CRP1 = 0xffffffff);
193  TUInt32 *_crp1ptr = (TUInt32 * const)&G_LPC43XX_CRP1;
194  #endif
195
196  #ifndef BBL_BASE_ADDRESS // Don't include anything in the second flash section
197  // We want to fill or put some kind of data in bank b at the very start of it
198  uEZ_PUT_SECTION(".fillFlashB0", static const TUInt32 G_LPC43XX_CRP2_FILL = 0x0);
199  //TUInt32 *_crp2fillptr = (TUInt32 * const)&G_LPC43XX_CRP2_FILL;
200  uEZ_PUT_SECTION(".crp2", static const TUInt32 G_LPC43XX_CRP2 = 0xffffffff);
201  TUInt32 *_crp2ptr = (TUInt32 * const)&G_LPC43XX_CRP2;
202  #endif
203
204  /*
205  * Macros:
206  */

```

Output

Show: Transcript Tasks

Building 'uEZProjectMaker' from solution 'uEZProjectMaker' in configuration 'THUMB Flash Release' Completed

Build up to date Completed

Memory Type	Used
SRAM1	5.8 KB of ...
FLASHA0	276 bytes ...
FLASHA	27.8 KB of...
FLASHB0	764 bytes ...
CRP2	4 bytes of ...
FLASHB	143.4 KB ...
SDRAM	5.4 MB of ...

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.

