**Iqble\_BootLoader(Usart)**

The Bootloader code starts executing on a device Reset. If there are no conditions to enter the firmware upgrade mode, the Bootloader starts executing the user application. The Bootloader performs Flash erase/program operations while in the firmware upgrade mode.

Entering the Firmware Upgrade Mode

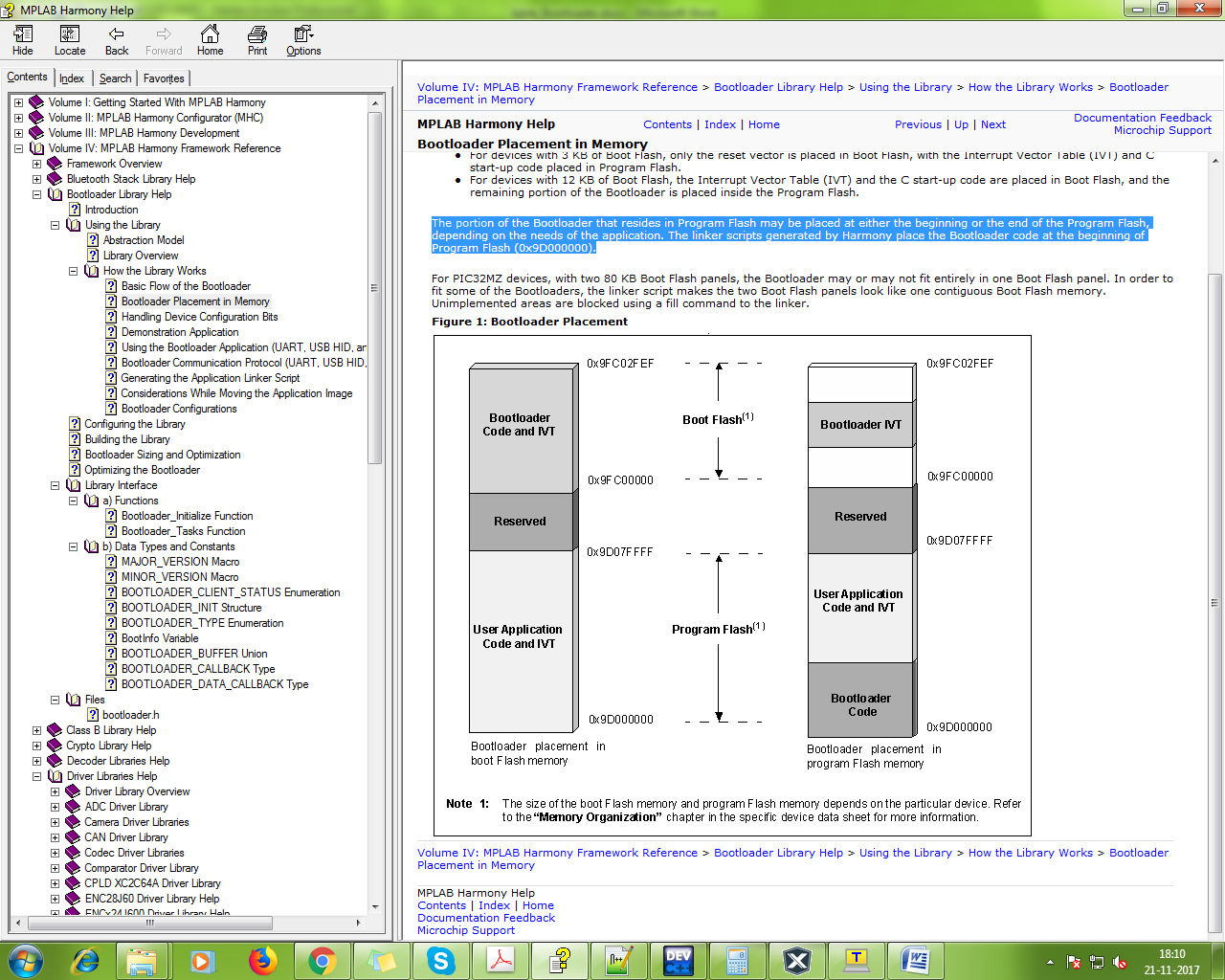
On a device Reset, the Bootloader forces itself into the firmware upgrade mode if the content of the user application’s reset vector address is erased. On PIC32, press and hold the switch, SW1(from BSP\_harmony), during power-up. While in firmware upgrade mode, the LED labeled LED1 on the PIC32 will blink.

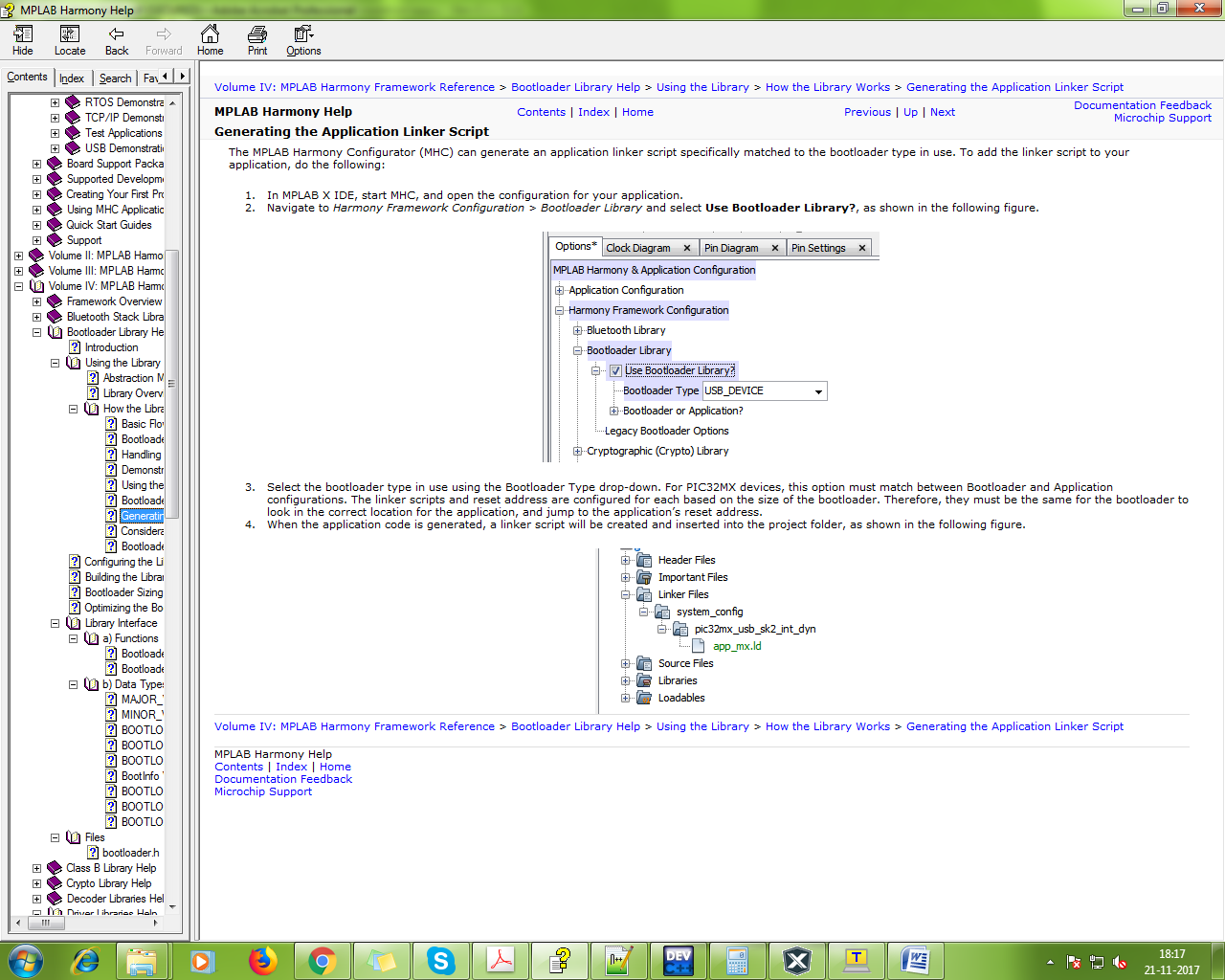
Figure 1 illustrates two schemes for the Bootloader placement based on the size of the Bootloader. Devices with a large enough Boot Flash memory can place all of the Bootloader within Boot Flash. Fitting the Bootloader within the Boot Flash memory provides the complete program Flash memory for the user application.

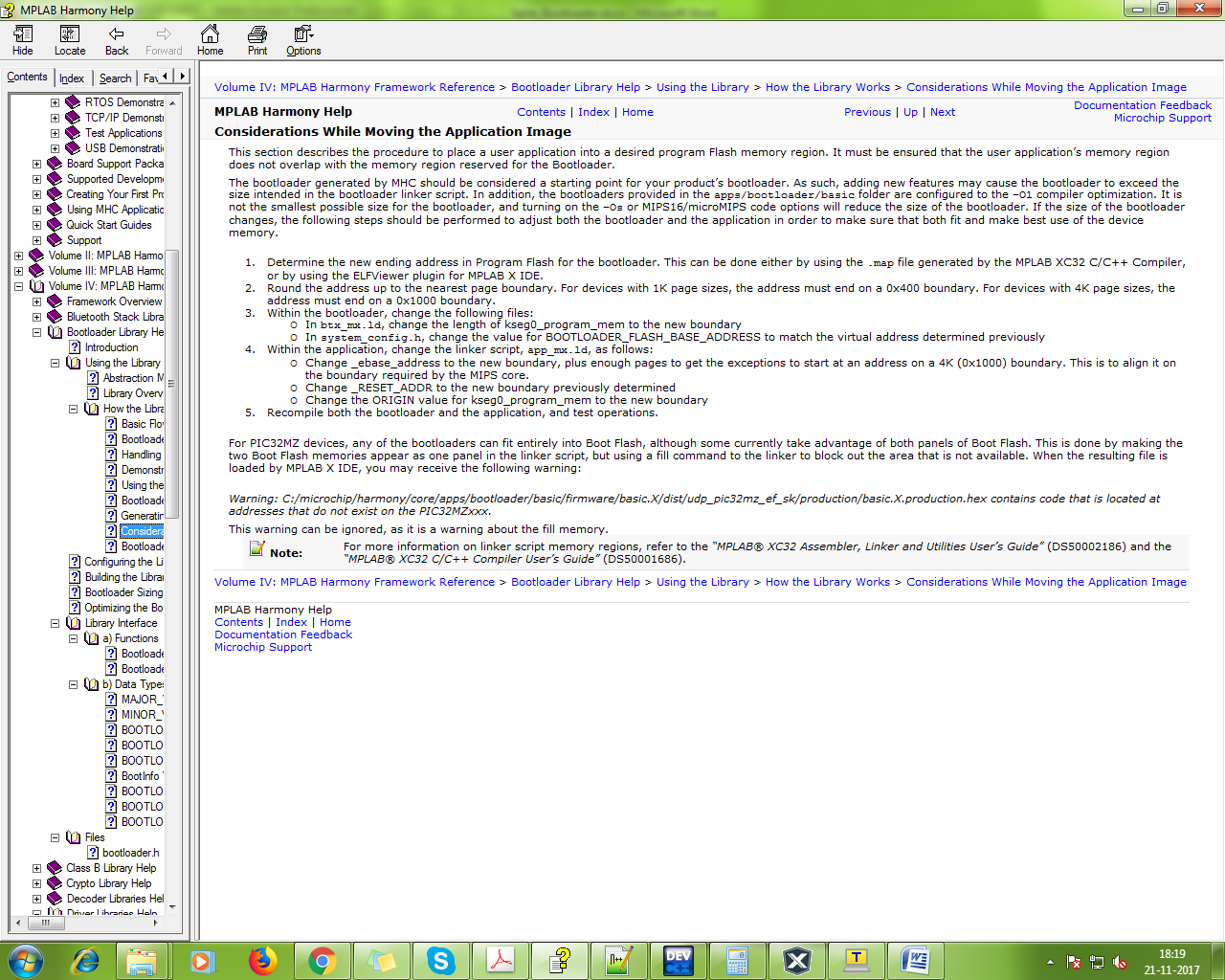
In the case of Bootloaders that exceed the size of PIC32 boot Flash, the Bootloader is split into two parts:

* For devices with 3 KB of Boot Flash, only the reset vector is placed in Boot Flash, with the Interrupt Vector Table (IVT) and C start-up code placed in Program Flash.
* For devices with 12 KB of Boot Flash, the Interrupt Vector Table (IVT) and the C start-up code are placed in Boot Flash, and the remaining portion of the Bootloader is placed inside the Program Flash.

The portion of the Bootloader that resides in Program Flash may be placed at either the beginning or the end of the Program Flash, depending on the needs of the application. The linker scripts generated by Harmony place the Bootloader code at the beginning of Program Flash (0x9D000000).



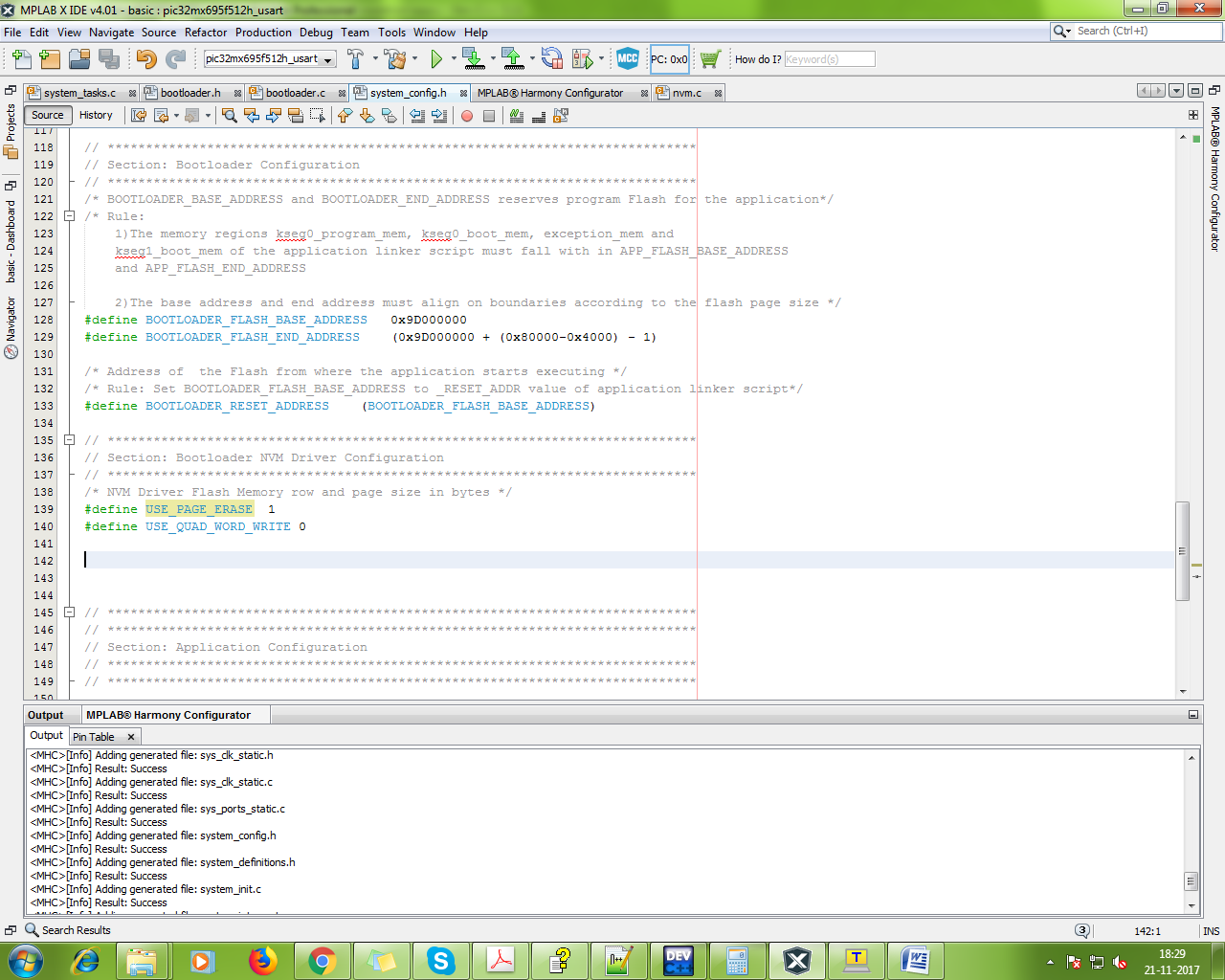




The configuration of the Bootloader Library is based on the file **system\_config.h**.

This header file ‘system\_config.h’ contains the configuration selection for the Bootloader Library. Based on the selections made, the Bootloader Library will or will not support selected features. These configuration settings will apply to all instances of the Bootloader Library.

In Bootloader Library, ‘system\_config.h’ contains the configuration macro,



**Note:**

BOOTLOADER\_FLASH\_BASE\_ADDRESS 0x9D000000 (we need to change this if Main application start form different address otherwise leave same as it is.)

BOOTLOADER\_FLASH\_END\_ADDRESS (0x9D000000 + 0x80000 - 1) (we need to change this if Main application use NVM memory, otherwise Bootloader erase all program flash memory from BOOTLOADER\_FLASH\_BASE\_ADDRESS)

For stop erasing NVM memory we need subtract “Used NVM Memory Page SIZE” from the total size of Program Flash memory and added with BOOTLOADER\_FLASH\_BASE\_ADDRESS. Ex. 0x9D000000 + (0x80000-(used\_NVM\_memory\_page\_size) – 1).

If Main Application used NVM memory then

USE\_PAGE\_ERASE 1 (set as 1 from 0, to prevent NVM memory erase on firmware update by Bootloader)

In main Application Side, make an application as bootloader application script from MHC. After configuration complete, a linker script (discuss before **Generating the Application Linker Script**) create by MHC. In this file we see all ‘\_RESET\_ADDR’, ‘kseg0\_program\_mem’ & ‘\_ebase\_address’, from there start the main application. If we need to change application ‘\_RESET\_ADDR’ then we need to change all other address too. See at **Optimizing the bootloader section**.

**Note:** download ELV viewer from plugings.

