Dual Bank Firmware Update User Guide

Application Note for 32-bit NuMicro®Family

Document Information

Abstract	Demonstrate how to implement Dual Bank Firmware Update under the M2L31 architecture.
Apply to	NuMicro® M2L31 Series.

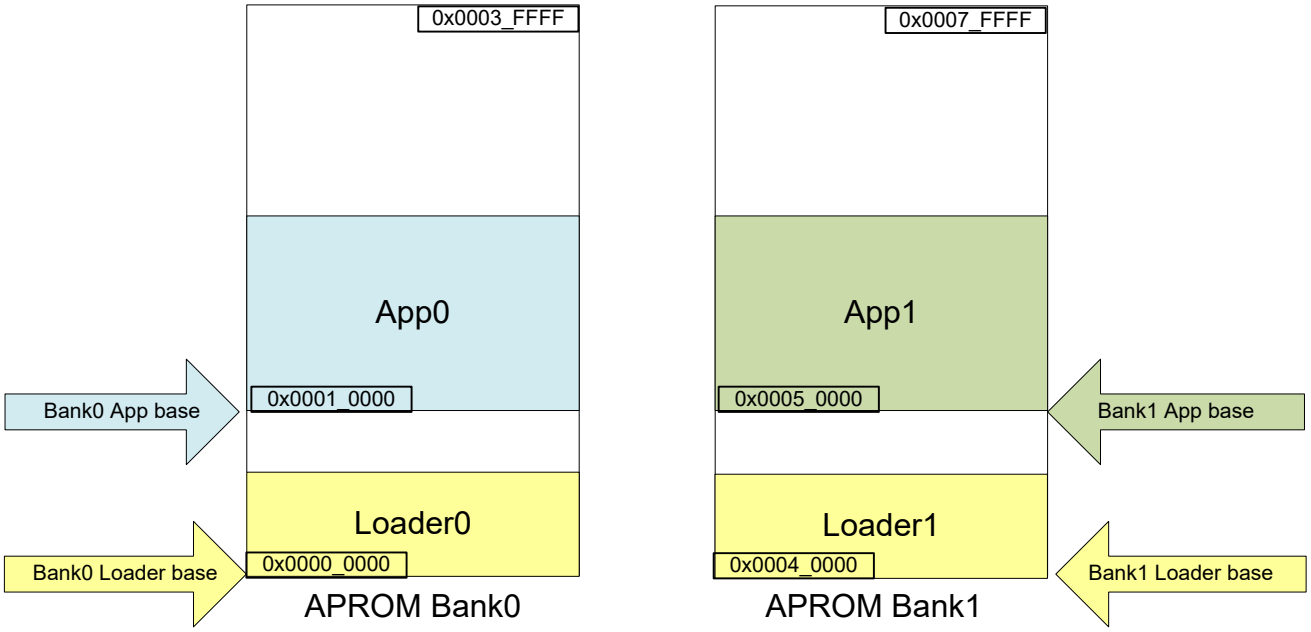
Table of Contents

1 Sample Code Introduction

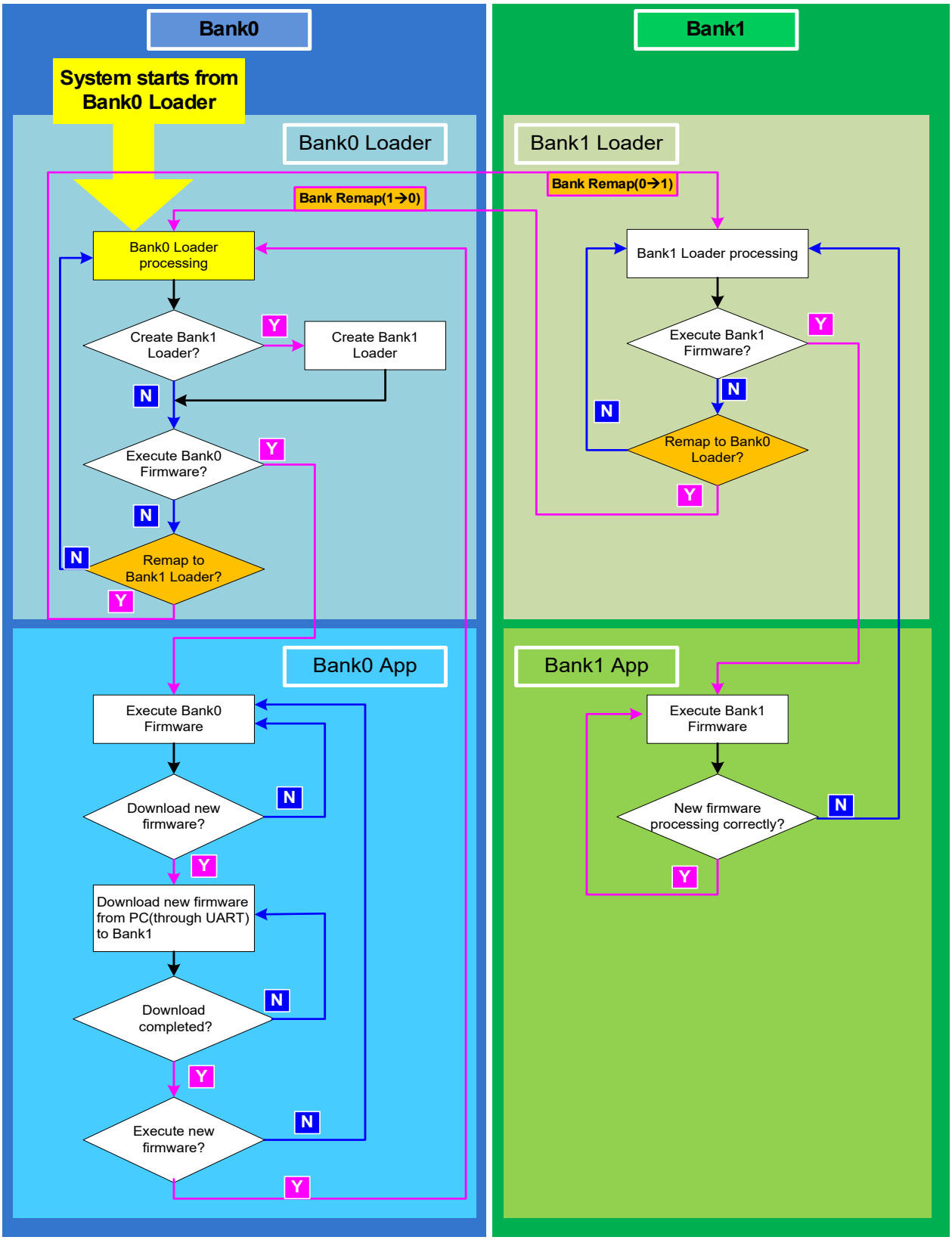
M2L31 BSP provides a Dual Bank firmware update sample program,located at:
\bsp\SampleCode\StdDriver\RMC_DualBankFwUpdate

This sample program implements a Dual Bank firmware update framework under the Dual Bank APROM architecture of M2L31. There are two main programs:

- Loader:
  Perform the control flow of system startup and firmware update, which are placedat the starting address of APROM Bank0 and Bank1 (Bank0 Loader base and Bank1 Loader base are shown in the figure below).
- App:
  Executable programs placed in the program execution areas of APROM Bank0 and Bank1(Bank0 App base and Bank1 App base are shown in the figure below). The App may be active firmware or new firmware.



The system control process is as follows:



2 The Operation of Firmware Update Sample Code

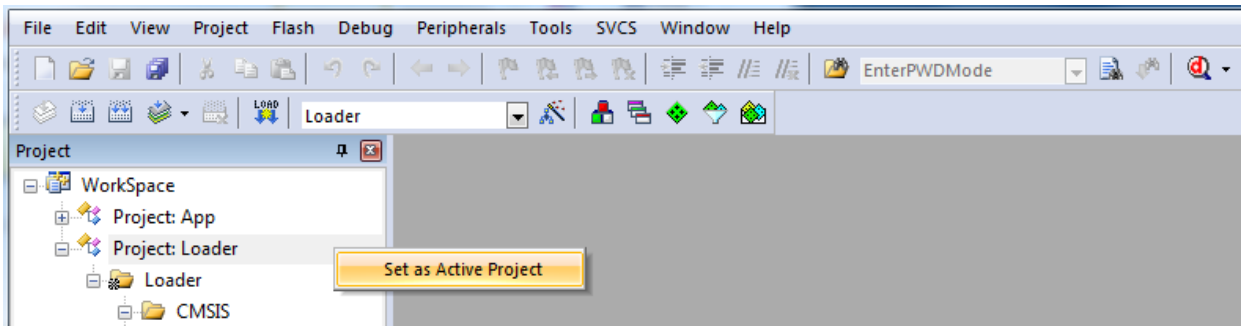
Before executing the program, first define the firmware naming used in the operation steps：

- Active firmware：

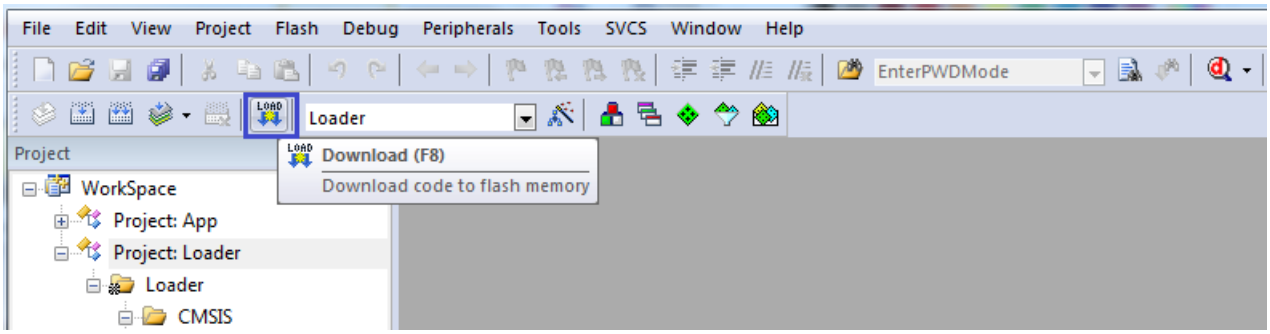
  The firmware that is initially placed in the Bank0 App area is also the firmware that the system executes under normal condition.
- New firmware：

  The new firmware loaded in firmware update process.

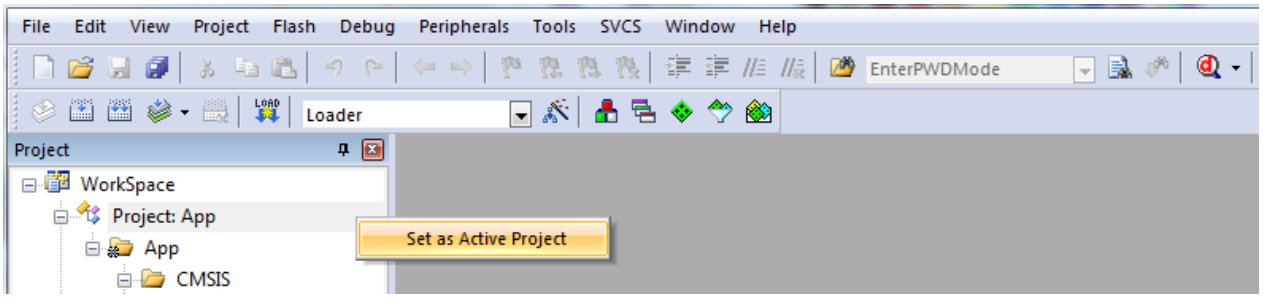
In \bsp\SampleCode\StdDriver\RMC_DualBankFwUpdate，open the project RMC_DualBankFwUpdate.uvmpw. This project has two targets: Loader and App. First select the Loader target. As shown in the figure below:



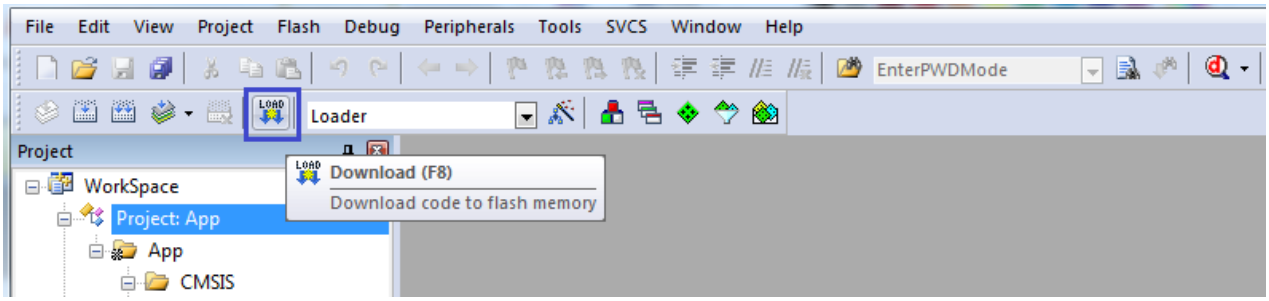
After compiling done, click the “LOAD” button to load the Loader into the Bank0 Loader execution area. As shown in the figure below:



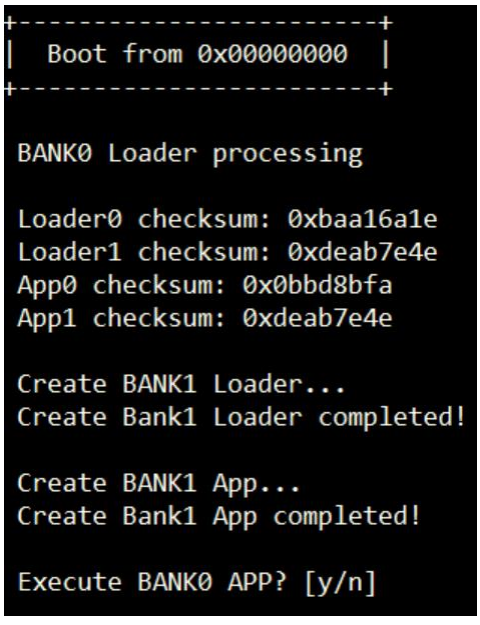
Then select the App target. As shown in the figure below:



After compiling done, click the “LOAD” button to load the App into the Bank0 App execution area. As shown in the figure below:

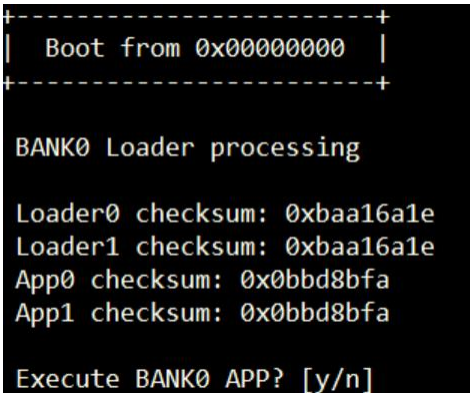


After the two programs being downloaded, press the reset button on the M2L31 to start the system. When the system is started for the first time, it will create Bank1 loader and Bank1 App by copying Bank0 Loader and Bank0 App. As shown in the figure below:

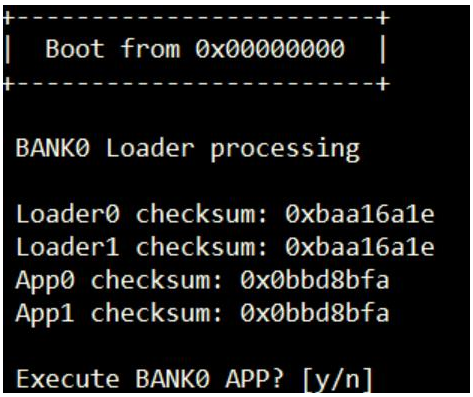


After restarting the system, Bank0 and Bank1 will have the same loader and app. Bank0's Loader is defined as Loader0 while Bank1's Loader is defined as Loader1; Bank0's App is defined as App0 while Bank1's App is defined as App1. Loader0 and Loader1 have the same

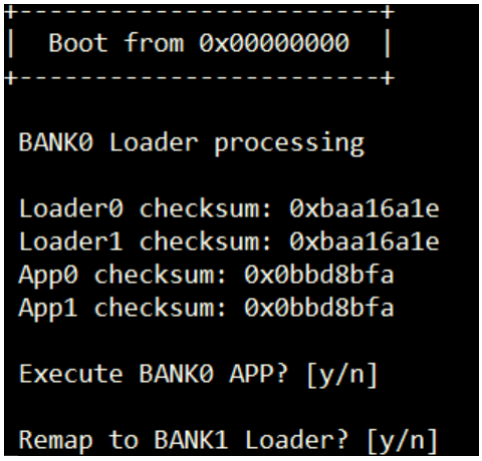
checksum value while App0 and App1 have the same checksum value. As shown in the following figure:



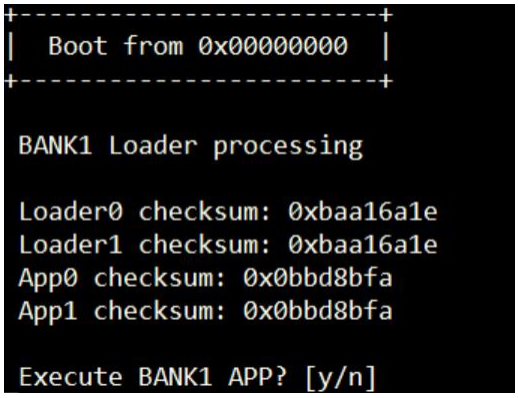
Then introduce how the Bank Remap function works under such framework. When the system is started, a dialogue message will appear to let the user decide whether to execute Bank0's firmware (Active firmware). As shown below:



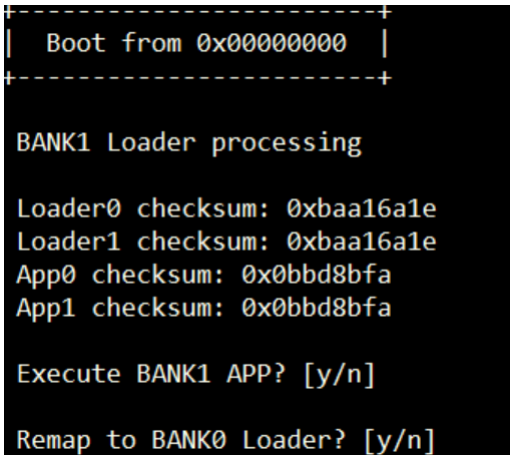
When user selects (n) to not execute Bank0's firmware, a dialog for executing Bank Remap will appear. As shown below:



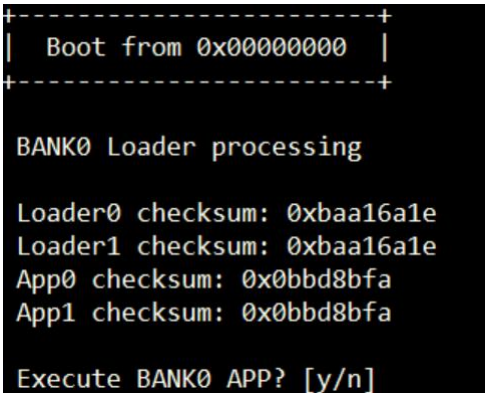
When user selects (y) to execute Bank Remap, CPU will immediately switch to Bank1 and execute Bank1 Loader. As shown below:



At this time, the flow executed in Bank1 Loader is similar to the execution of Bank0 Loader, and it can be used to select whether to execute the Bank1's App. When user selects(n) to not execute Bank1's firmware, a dialog for executing Bank Remap will appear. As shown below:



When user selects (y) to execute Bank Remap, the CPU will immediately switch back to Bank0 and execute Bank0 Loader. As shown below:



Repeat the above steps, user can switch between Bank0 and Bank1, to verify whether the Bank Remap function is normal.

Next, it will introduce how to execute firmware update process. First, the system starts to execute Bank0 loader, as shown in the following figure:

```
+------------------------+
|  Boot from 0x00000000  |
+------------------------+

 BANK0 Loader processing

 Loader0 checksum: 0xbaa16a1e
 Loader1 checksum: 0xbaa16a1e
 App0 checksum: 0x0bbd8bfa
 App1 checksum: 0x0bbd8bfa

 Execute BANK0 APP? [y/n]
```

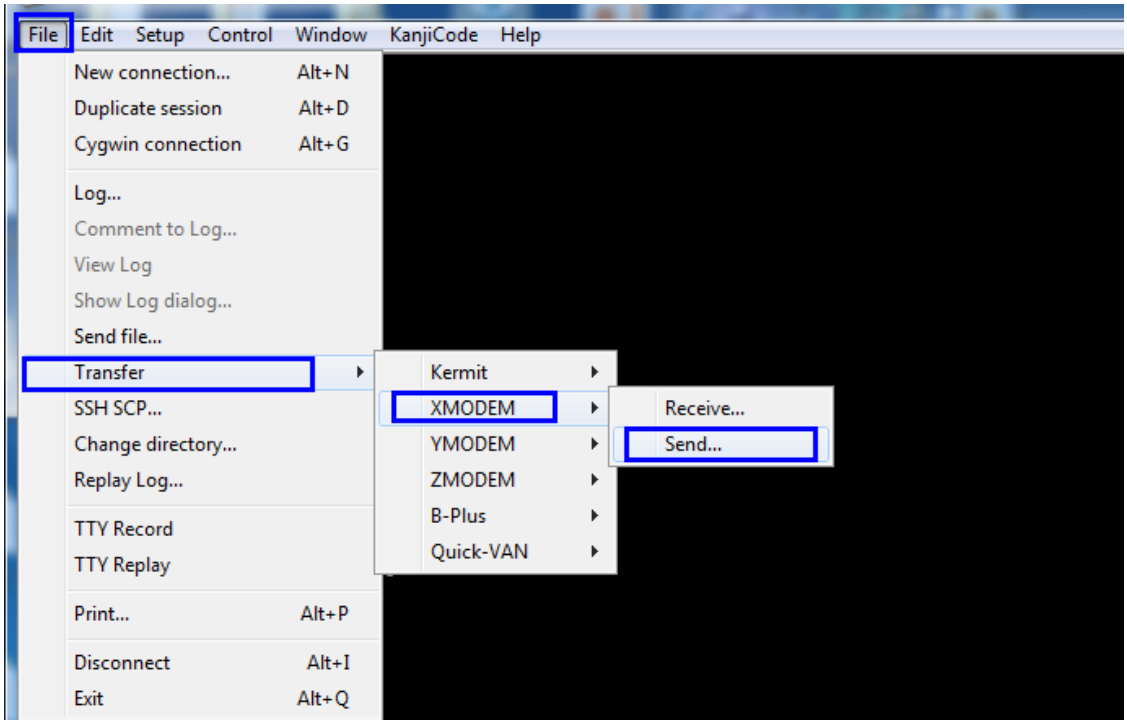
At this time, user can select (y) to execute Bank0 firmware. As shown below:

```
+------------------------+
|  Boot from 0x00010000  |
+------------------------+

 BANK0 APP processing

 Download new FW?[y/n]
```

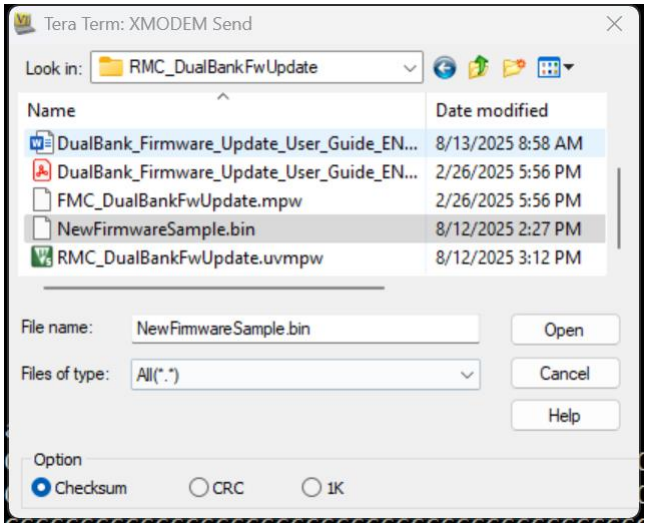
Then user can see the message "Boot from "0x10000" and "BANK0 APP processing". Then a selection dialog will appear, letting the user decide whether to do firmware update process. When user selects (y) to execute the firmware update, the Xmodem transmission start character ‘C’ will appear, as shown below:

```
+------------------------+
|  Boot from 0x00010000  |
+------------------------+

 BANK0 APP processing

 Download new FW?[y/n]

 Bank0 processing, download data to Bank1.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

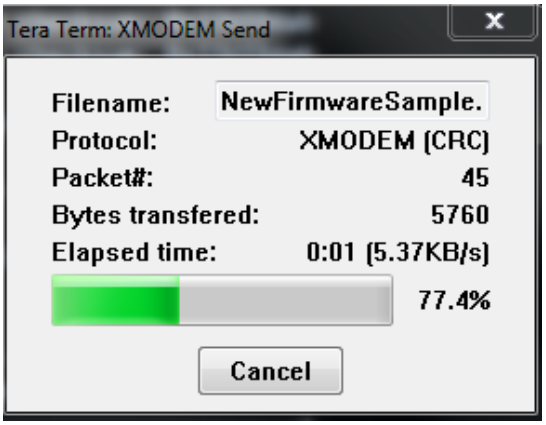
Then select”File→Transfer→XMODOM→Send” in the UART window to start Xmodem transmission, as shown below:



Then a window of XMODEM Send will appear, select NewFirmwareSample.bin (located in \bsp\SampleCode\StdDriver\RMC_DualBankFwUpdate) provided in Sample code in this window, as shown below:



After double-clicking the bin file, the file starts to transfer, as shown in the following figure:



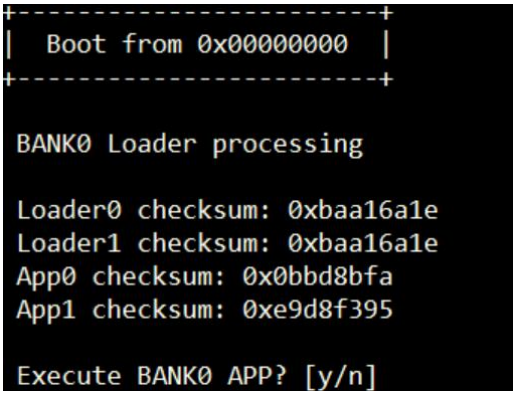
After the transfer being completed, return to the original debug window, and the message "Firmware download completed!!" will appear, indicating that the firmware update is complete. As shown below:

```
+------------------------+
|  Boot from 0x00010000  |
+------------------------+

 BANK0 APP processing

 Download new FW?[y/n]

 Bank0 processing, download data to Bank1.
CCCCCCCCCCCCCCCCXomdem transfer done!
Total trnasfer size is 10624

 Firmware download completed!!
```

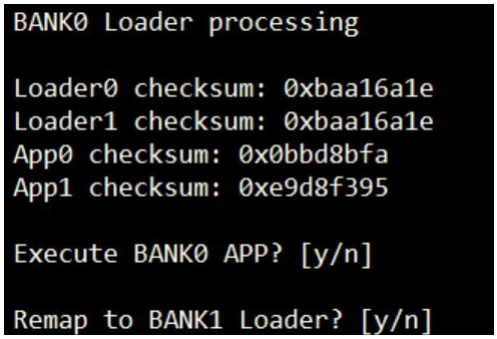
Then it will return to the screen of Bank0 firmware execution, as shown in the following figure:

```
+------------------------+
|  Boot from 0x00010000  |
+------------------------+

 BANK0 APP processing

 Download new FW?[y/n]
```

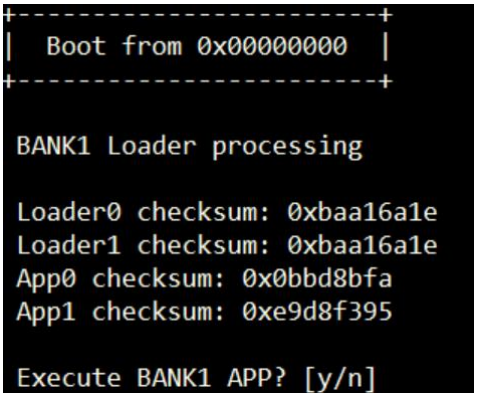
At this time, the new version of firmware is placed in Bank1 App base, so if user wants to execute the new firmware, the (n) should be selected that the system will return to the Bank0 Loader for execution. As shown below:



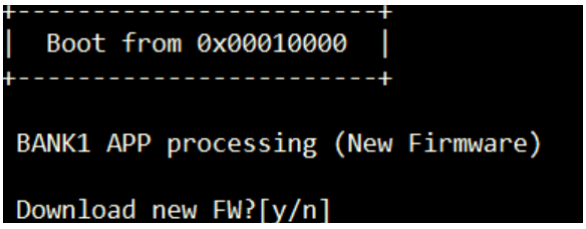
Then select (n) tonot execute Bank0 firmware, as shown in the following figure:



Then select (y) to switch the system to Bank1 Loader, as shown below:



Finally, select (y) to execute the program in Bank1 App base, which is the new version of firmware. as shown in the following figure:



Revision History

Date	Revision	Description
2025.08.15	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.