Using M031 Bootloader to Upgrade APROM through BLE Module

Example Code Introduction for 32-bit NuMicro® Family

Document Information

Application	This example code uses ISP_UART bootloader program in LDROM with IAP mode function to upgrade firmware in APROM through ESP32 BLE module.
BSP Version	M031_Series_BSP_CMSIS_V3.05.000
Hardware	NuMaker-M032KI V1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. Overview

This example code uses ISP_UART bootloader program of the M031 series microcontroller (MCU) to upgrade firmware in APROM through ESP32 BLE module.

The IAP (In-Application Programming) bootloader allows the M031 to update the firmware in other Flash address block. After finishing the IAP process, the update process will remap the start address of new firmware to Vector Page Address and then user can use mobile phone's APP to call system reset function to run the new firmware.

The ESP32 BLE module can interface with other systems to provide Wi-Fi and Bluetooth functionality through its UART interface. The sample code needs to connect the ESP32 BLE module to the NuMaker-M032KI board for BLE to UART transparent transmission. The ESP32_BLE_UART.ino in the ThirdParty folder is ESP32 firmware supports ISP command, user can use ESP32 Arduino IDE to upgrade the ESP32 BLE module.

Users can download and install the ISPTooL Android v1.1.1.apk APP from https://play.google.com/store/apps/details?id=com.nuvoton.nuisptool_android&hl=zh-hk and select the BLE interface to connect with ISP_UART bootloader program to upgrade firmware in APROM.

This example code provides two test programs. One programs data to APROM at offset 0x10000 and prints the contents of the address, and the other programs data to APROM at offset address 0x50000 and prints the contents of the address. User can copy the bin file to mobile phone to test the IAP process.

1.1 Principle

For OTA (Over-The-Air) application, the memory space concept is a slightly different in terms of FMC commands and CPU. From FMC command viewpoint, the physical address concept is adopted. All of the FMC commands regard APROM memory as physical address except "remapping" command. From CPU viewpoint, the virtual address concept is adopted. APROM memory space is separated into two banks. In the runtime stage, the virtual address depends on address operation model (OP0, OP1) as shown in Figure 1-1.

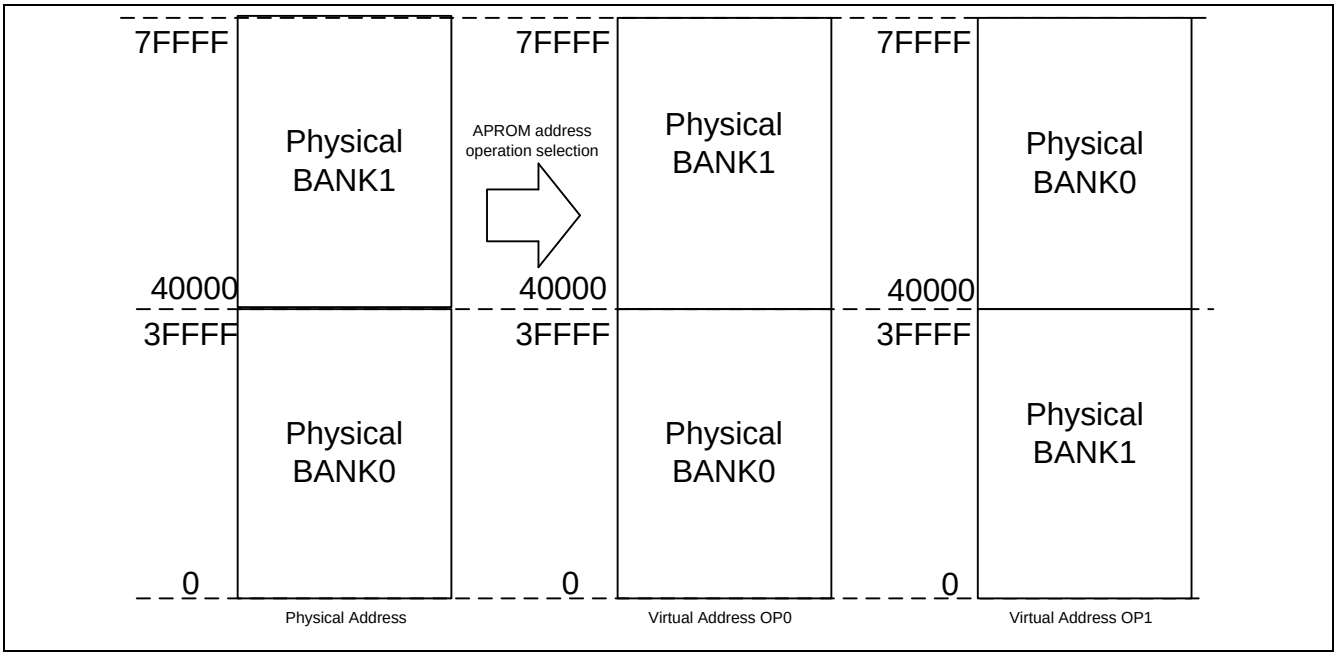


Figure 1-1 Address Operation Model

To activate APROM reboot with different address operation model, use address operation model selection command 0x2C with FMC_ISPDAT “0x5AA5_5AA5” and FMC_ISPADDR ”0x0” (Address OP0) or ”0x1” (Address OP1) to select an address operation model.

After compiling test program of this example code and generating the binary file, there is version information at the offset address 0x100 of the binary. The bootloader program will determine which APROM bank's firmware version is the latest and set it as the active bank. For example, if the firmware version of APROM bank0 is the latest, the bootloader program will set APROM bank0 as active bank. The firmware in APROM Bank0 will be executed. When user uses ISPTooL Android v1.1.1.apk to connect the bootloader program through BLE module, the bootloader program will erase APROM bank1 to upgrade new firmware. Figure 1-2 shows the bootloader program flow chart.

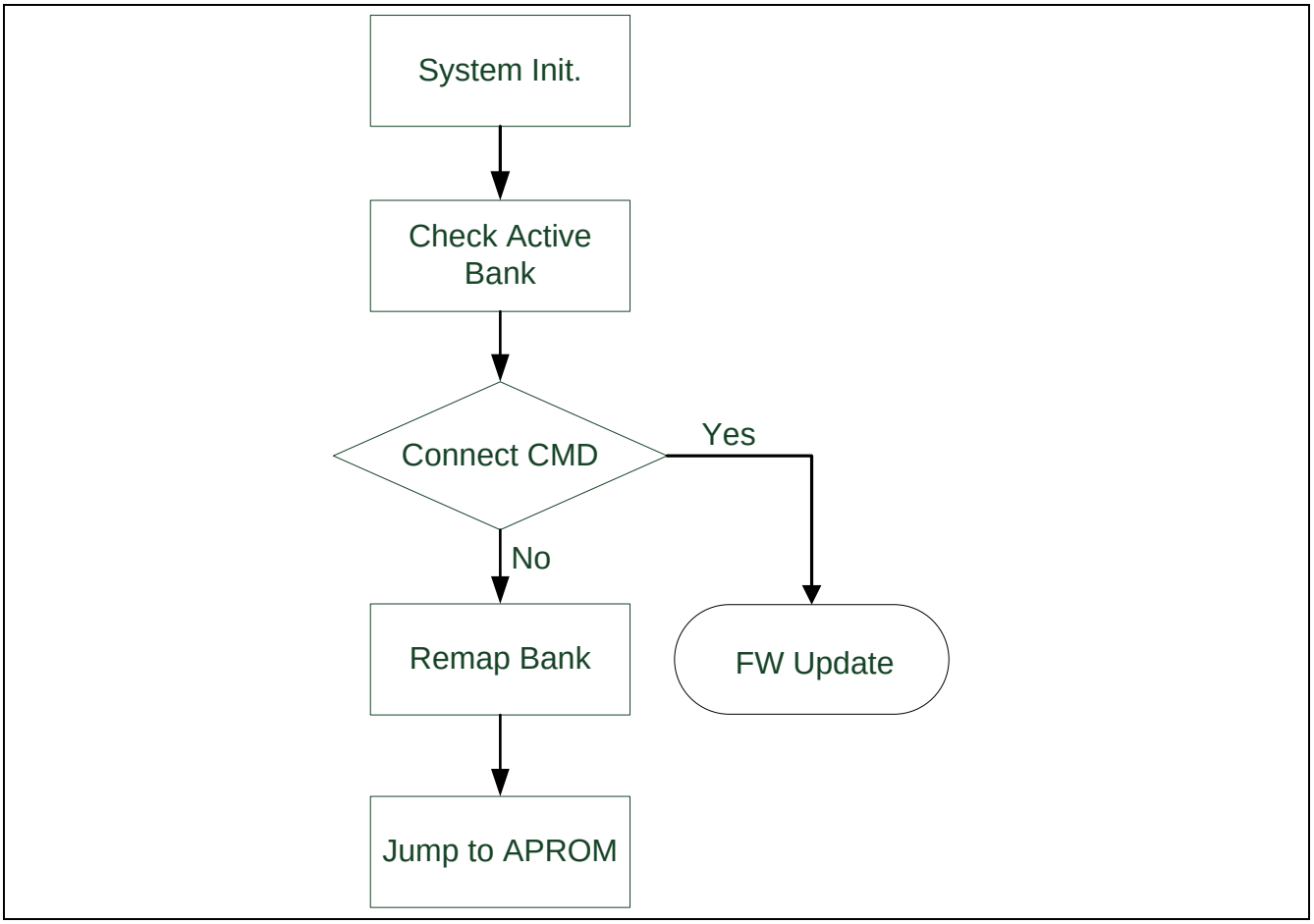


Figure 1-2 Bootloader Flow Chart

This example code needs to set chip boot from LDROM with IAP mode and program the sample code to LDROM. It is necessary to program APP0 code to APROM bank0 and bank1. To avoid UART signal interference, the VCOM function of NuMaker-M032KI board must be set to disable. First, enabling Bluetooth of mobile phone and then open ISPTooL APP, user can select BLE interface and click **FIND DEVICE** icon to scan BLE device. Select the **UART Service** device and then press the reset button of NuMaker-M032KI board to connect with ISP_UART bootloader program through ESP32 BLE module, as shown in Figure 1-3. When ISPTooL APP connect with bootloader program, user can click **OPEN** icon to set the path of firmware update to APROM. If **Reset and Run** icon is enabled, bootloader program will call system reset after the firmware in APROM is updated and then the new firmware will be switched and executed, as shown in Figure 1-4.

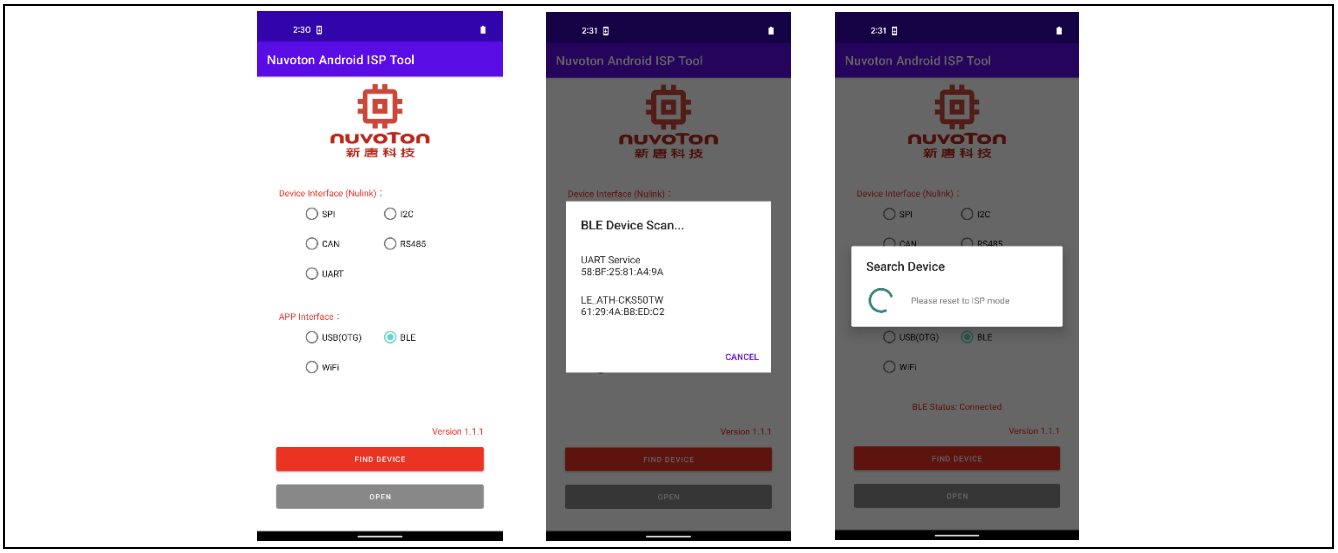


Figure 1-3 ISPTooL APP Connected to M031

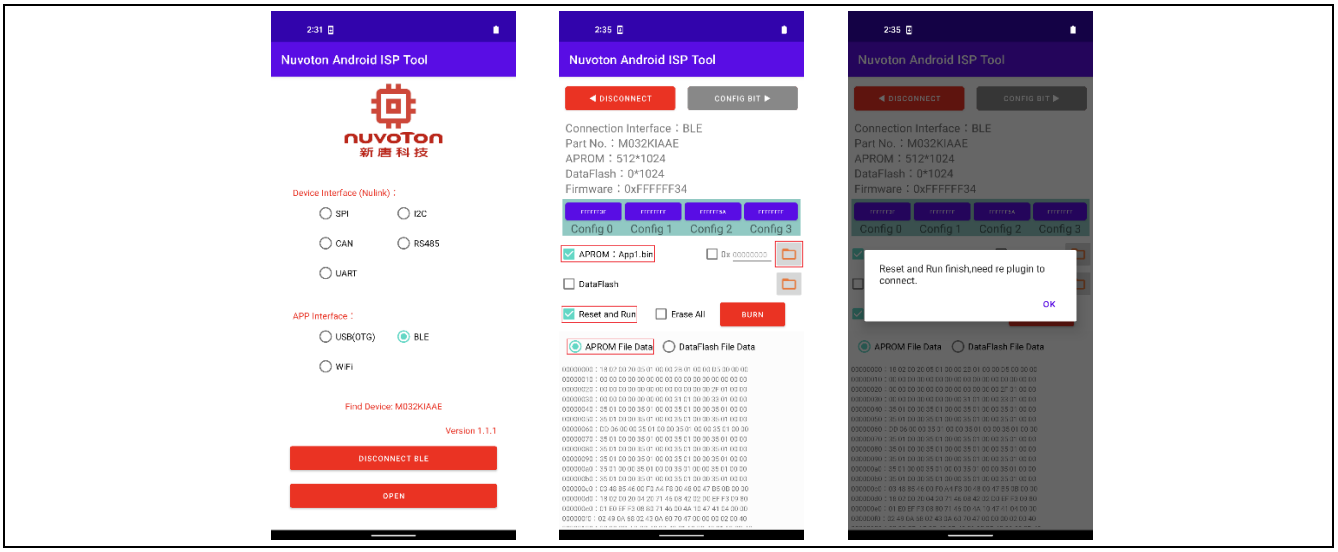
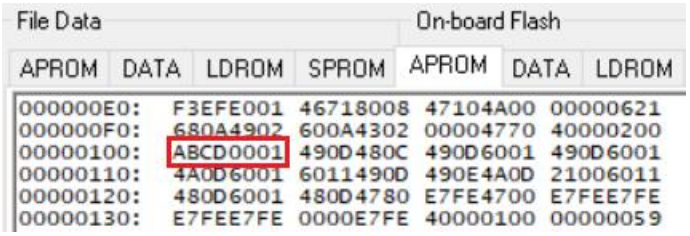


Figure 1-4 Program Code and Execute New Firmware

1.2 Demo Result

The offset at address 0x100 of firmware APP0 and APP1 includes the version information as shown in Figure 1-5.

The version of firmware APP0



The version of firmware APP1

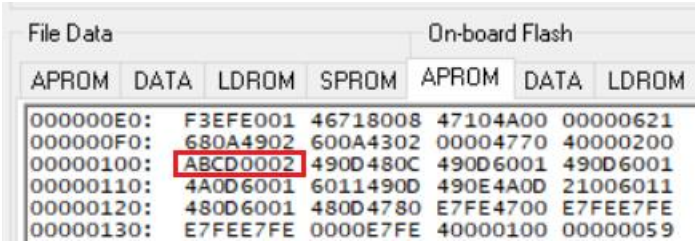


Figure 1-5 Version of Firmware

If user need not update firmware, this example code executes Bank0 firmware. The result is shown in Figure 1-6.

```
Application 0 on bank 0
    Flash Test Passed.

[CPU DUMP: 0x10000 ~ 0x10800]

0x00010000 0x00010004 0x00010008 0x0001000c 0x00010010 0x00010014 0x00010018 0x0001001c
0x00010020 0x00010024 0x00010028 0x0001002c 0x00010030 0x00010034 0x00010038 0x0001003c
0x00010040 0x00010044 0x00010048 0x0001004c 0x00010050 0x00010054 0x00010058 0x0001005c
0x00010060 0x00010064 0x00010068 0x0001006c 0x00010070 0x00010074 0x00010078 0x0001007c
0x00010080 0x00010084 0x00010088 0x0001008c 0x00010090 0x00010094 0x00010098 0x0001009c
0x000100a0 0x000100a4 0x000100a8 0x000100ac 0x000100b0 0x000100b4 0x000100b8 0x000100bc
```

Figure 1-6 Execution Result of Bank0 Firmware

After updating new firmware, the execution result of the Bank1 firmware will be printed out by the PC UART, as shown in Figure 1-7.

```
Application 1 on bank 1
    Flash Test Passed.

[CPU DUMP: 0x50000 ~ 0x50800]

0x00050000 0x00050004 0x00050008 0x0005000c 0x00050010 0x00050014 0x00050018 0x0005001c
0x00050020 0x00050024 0x00050028 0x0005002c 0x00050030 0x00050034 0x00050038 0x0005003c
0x00050040 0x00050044 0x00050048 0x0005004c 0x00050050 0x00050054 0x00050058 0x0005005c
0x00050060 0x00050064 0x00050068 0x0005006c 0x00050070 0x00050074 0x00050078 0x0005007c
0x00050080 0x00050084 0x00050088 0x0005008c 0x00050090 0x00050094 0x00050098 0x0005009c
0x000500a0 0x000500a4 0x000500a8 0x000500ac 0x000500b0 0x000500b4 0x000500b8 0x000500bc
```

Figure 1-7 Execution Result of Bank1 Firmware

2. Code Description

The IAP bootloader main function sets system clock and UART function, and will use UART interface to communicate with the ESP32 BLE module to upgrade new firmware. It also determines which APROM bank's firmware version is the latest and sets it as the active bank. When the ISPTooL APP connects to bootloader program through BLE module, bootloader program will erase non-active bank of APROM to upgrade new firmware. If user need not update firmware, the M031 will call the RemapBank() function according to the active bank to switch the firmware in the APROM bank0 or bank1 address.

```
int32_t main(void)
{
    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();
    /* Init UART to 115200-8n1 */
    UART_Init();

    /* Checking if flash page size matches with target chip's */
    if( (GET_CHIP_SERIES_NUM == CHIP_SERIES_NUM_I) || (GET_CHIP_SERIES_NUM ==
CHIP_SERIES_NUM_G) )
    {
        if(FMC_FLASH_PAGE_SIZE != 2048)
        {
            /* FMC_FLASH_PAGE_SIZE is different from target device */
            /* Please enable the compiler option PAGE_SIZE_2048 in fmc.h */
            while(SYS->PDID);
        }
    }
    else
    {
        if(FMC_FLASH_PAGE_SIZE != 512)
        {
            /* FMC_FLASH_PAGE_SIZE is different from target device */
            /* Please disable the compiler option PAGE_SIZE_2048 in fmc.h */
            while(SYS->PDID);
        }
    }

    CLK->AHBCLK |= CLK_AHBCLK_ISPCKEN_Msk;
    FMC->ISPCTL |= (FMC_ISPCTL_ISPEN_Msk | FMC_ISPCTL_APUEN_Msk);
    g_apromSize = GetApromSize() >> 1;
    GetDataFlashInfo(&g_dataFlashAddr, &g_dataFlashSize);

    CheckActiveBank();

    SysTick->LOAD = 300000 * CyclesPerUs;
    SysTick->VAL   = (0x00);
    SysTick->CTRL = SysTick->CTRL | SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk;

    while (1)
    {
        if ((bufhead >= 4) || (bUartDataReady == TRUE))
        {
```

```
            uint32_t lcmd;
            lcmd = inpw(uart_rcvbuf);

            if (lcmd == CMD_CONNECT)
            {
                goto _ISP;
            }
            else
            {
                bUartDataReady = FALSE;
                bufhead = 0;
            }
        }

        if (SysTick->CTRL & SysTick_CTRL_COUNTFLAG_Msk)
        {
            goto _APROM;
        }
    }

_ISP:

    while (1)
    {
        if (bUartDataReady == TRUE)
        {
            bUartDataReady = FALSE;
            ParseCmd(uart_rcvbuf, 64);
            PutString();
        }
    }

_APROM:

    /* All ISP commands are relative to physical address except FMC_ISPCMD_VECMAP */
    SetVectorPageAddr(0x0000);

    if(g_u8ActiveBank == 0)
    {
        /* After bank swap command, Bank0 virtual address 0 will remap to 0 */
        RemapBank(0);
    }
    else
    {
        /* After bank swap command, Bank1 virtual address 0 will remap to 0 */
        RemapBank(1);
    }

    SYS->RSTSTS = (SYS_RSTSTS_PORF_Msk | SYS_RSTSTS_PINRF_Msk);
    FMC->ISPCTL &= ~(FMC_ISPCTL_ISPEN_Msk | FMC_ISPCTL_BS_Msk);
    SCB->AIRCR = (V6M_AIRCR_VECTKEY_DATA | V6M_AIRCR_SYSRESETREQ);

    /* Trap the CPU */
    while (1);
}
```

The following code determines which APROM bank's firmware version is the latest and sets it as the active bank.

```
void CheckActiveBank(void)
{
    int32_t i32Version;

    ReadData(BANK0_VER_BASE, BANK0_VER_BASE + 4, (uint32_t *)g_u32VerAddr0);
    ReadData(BANK1_VER_BASE, BANK1_VER_BASE + 4, (uint32_t *)g_u32VerAddr1);
    i32Version = g_u32VerAddr0[0] - g_u32VerAddr1[0];

    if(i32Version >= 0)
        g_u8ActiveBank = 0;
    else
        g_u8ActiveBank = 1;
}
```

3. Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M031_Series_BSP_CMSIS_V3.05.000
- IDE version
  - Keil uVersion 5.28
- ISP version
  - ISPTool Android v1.1.1.apk

3.2 Hardware Requirements

- Circuit components
  - NuMaker-M032KI V1.0
- Pin Connect
  - Connect the UART0 TX (PB.13) pin to the ESP32 RX pin and the UART0 RX (PB.12) pin to the ESP32 TX pin.

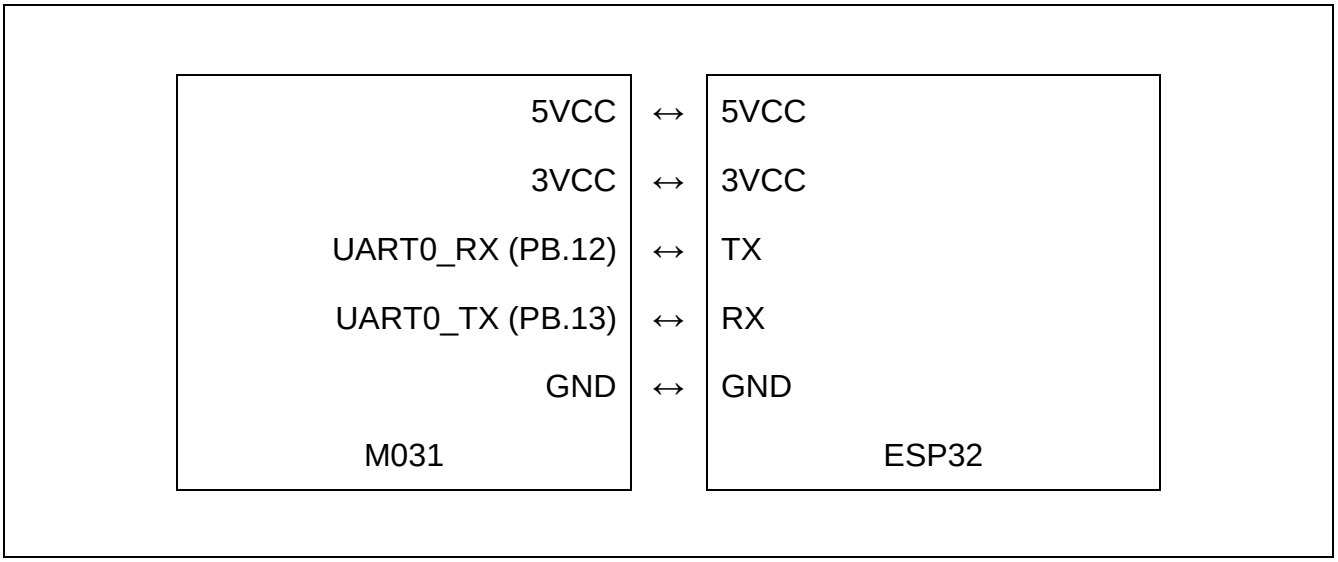


Figure 3-1 Pin Connect

4. Directory Information

The directory structure is shown below.

EC_M031_Bootloader_to_Upgrade_APROM_through_BLE_Module_V1.00	
Library	Sample code header and source files
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	Source file of example code
ThirdParty	
ESP32Code	ESP32 firmware supports ISP command

Figure 4-1 Directory Structure

5. Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *M031_Bootloader_to_Upgrade_APROM_through_BLE_Module.uvproj*.
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Enter debug mode.
   - Run

6. Revision History

Date	Revision	Description
2022.12.08	1.00	Initial version.

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.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.