I²C and SPI ISP Bridge in M031/M032

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example code demonstrates how to run I²C/ SPI interface ISP Bridge in the M031/M032 series microcontroller.
BSP Version	M031 Series CMSIS BSP V3.01.001
Hardware	NuMaker-M032SE Ver1.3

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

The Nuvoton 8-bit and 32-bit microcontrollers all provide two internal program memory blocks, APROM and LDROM. APROM usually stores user code, and LDROM generally stores boot code of In-System-Programming (ISP). A general operating procedure is that LDROM program receives user code and then updates user code into APROM through software register control to achieve the In-System Programming (ISP) function.

For the convenience of programming user code into APROM, the microcontroller is pre-programmed with the standard ISP code in LDROM before shipping from factory. User can directly program APROM through ISP in a friendly way. The ISP code supports the I^2C/SPI interface, but the personal computer does not support this interface. This example code demonstrates how to run the I^2C/SPI interface ISP Bridge in the M031/M032 series microcontroller.

1.1 Demo Result

The procedure is as follows:

1. Download and install Nuvoton NuMicro ISP Programming Tool.
2. Open the ISP Programming Tool, and select one connection interface to connect as shown in Figure 1-1.

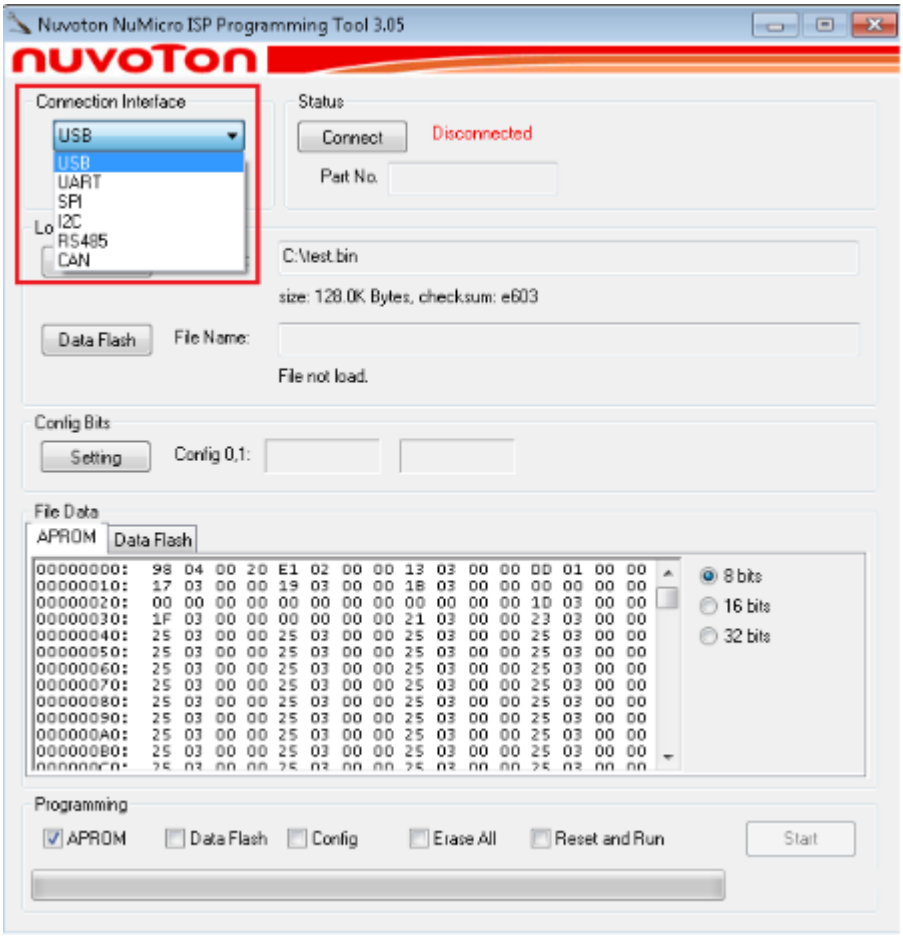

Figure 1-1 Startup Screen of ISP Tool

3. In the ISP Tool window, the connection status is shown as “Waiting for device connection” since the ISP tool has not been connected with the target chip, as shown in Figure 1-2.

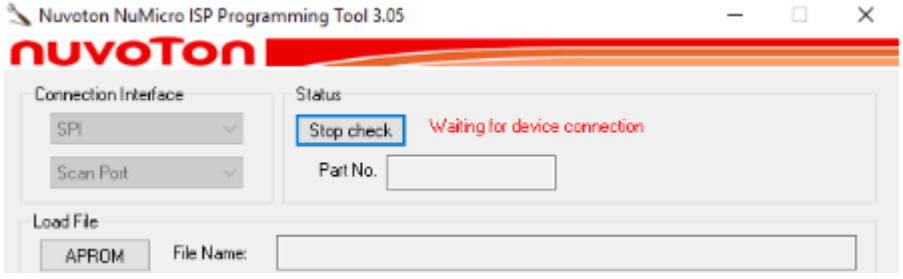

Figure 1-2 ISP is Not Connected to Any Device of ISP Tool

4. Refer to Figure 3-2 SPI Pin Connect to connect pins of ISP Bridge to the target chip by SPI interface according to the connection interface in Step 2.

5. Download BSP sample code and open ISP firmware of Keil sample code project that can be found in *SampleCode\ISP*. The interfaces of ISP firmware sample code may be

different for each NuMicro® chip series. The M480 BSP sample code download link: https://github.com/OpenNuvoton/M480BSP

6. Invoke “ISP_SPI” through **Project → Options for Target → Utilities**, select Nuvoton Nu-Link Debugger when the **Use Target Driver for Flash Programming** option is enabled, and select the **Update Target before Debugging** option, as shown in Figure 1-3.

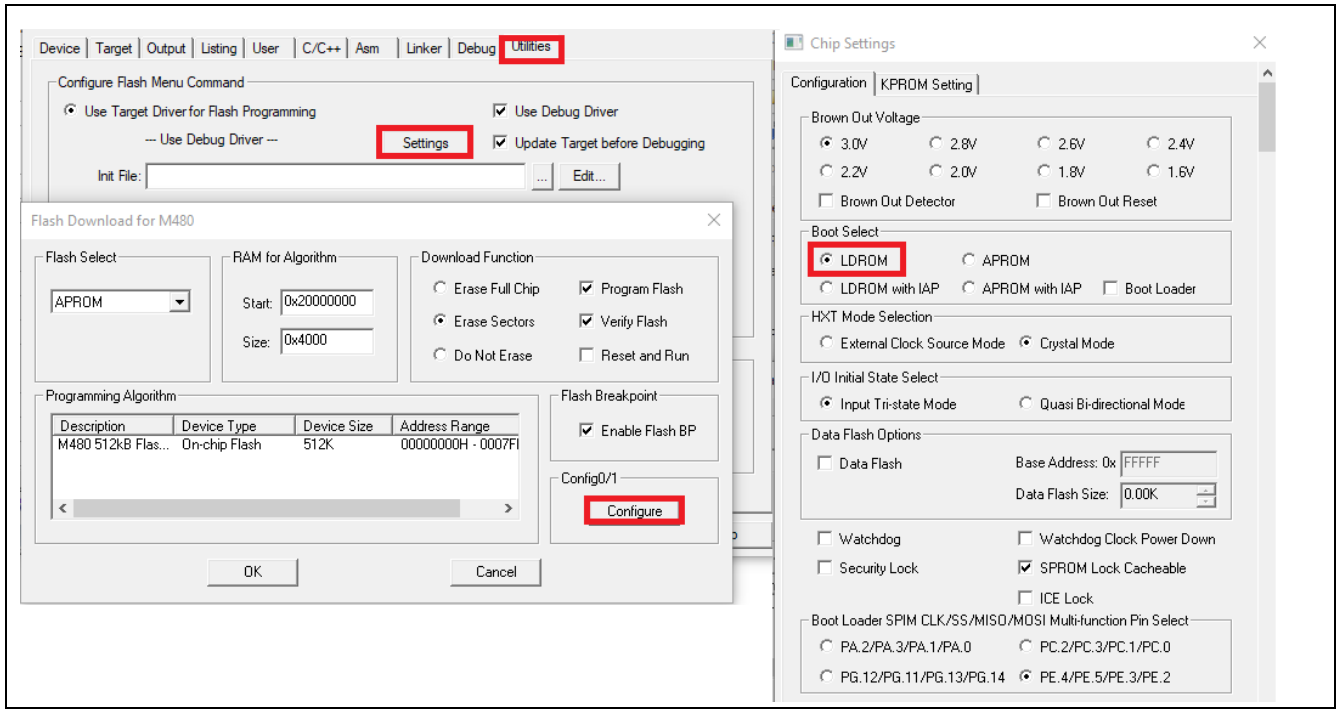

Figure 1-3 Boot from LDROM Setting in Keil ISP Firmware Project

7. Click the **Settings** button to open the Flash Download form, as shown in Figure 1-3.

8. Click the **Configure** button to open the User Configuration form and set **Boot Select** option to LDROM and click **OK** as shown in Figure 1-3.

9. Build and download code to LDROM of the target chip, as shown in Figure 1-4.

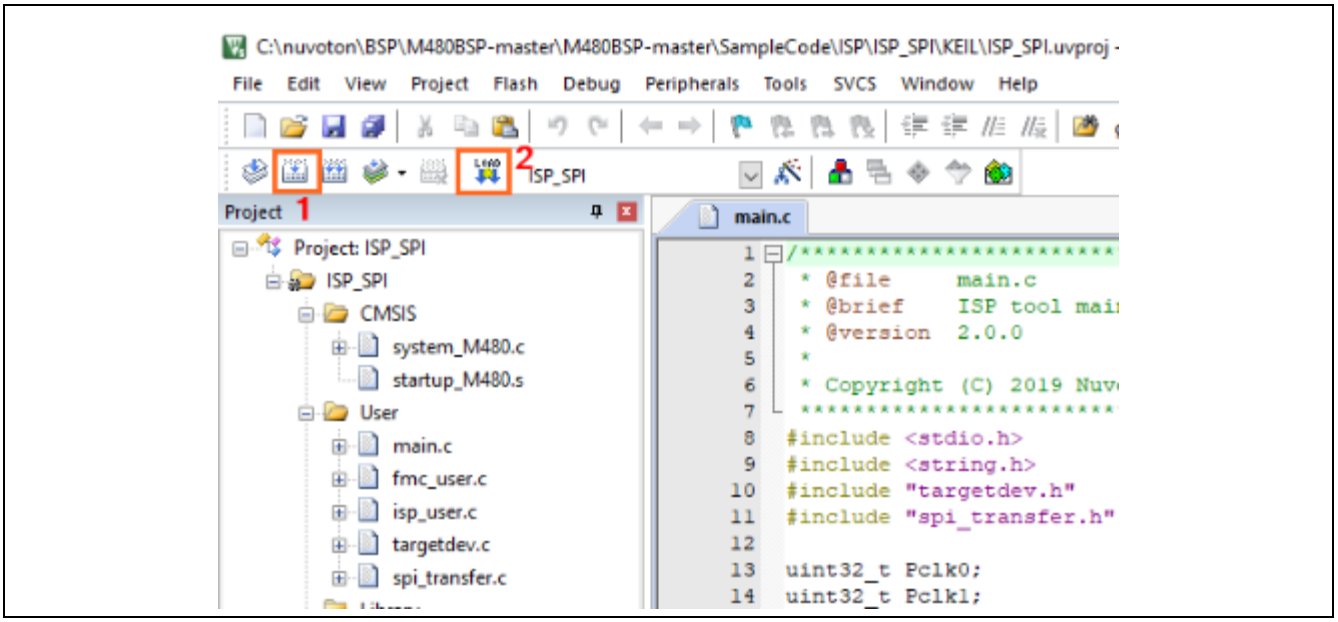

Figure 1-4 Build and Download Code

10. Open the ISP Programming Tool, click the **Connect** button, and reset the target chip to run ISP code. The ISP Programming Tool will connect to the target chip, for example, SPI connection interfaces, as shown in Figure 1-5.

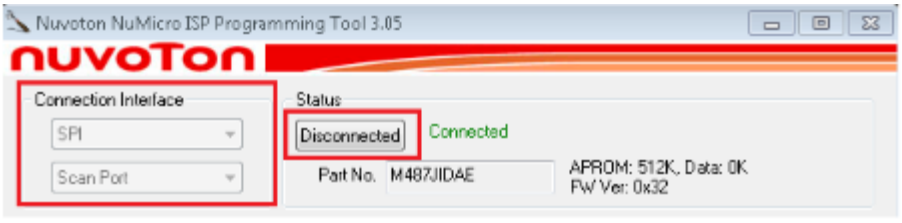

Figure 1-5 Connect to Target Chip with SPI Interface

11. Load the programming file, check the programming option and click the **Start** button. The ISP Programming Tool starts to program the file to the target chip by user selected connection interface, as shown in Figure 1-6.

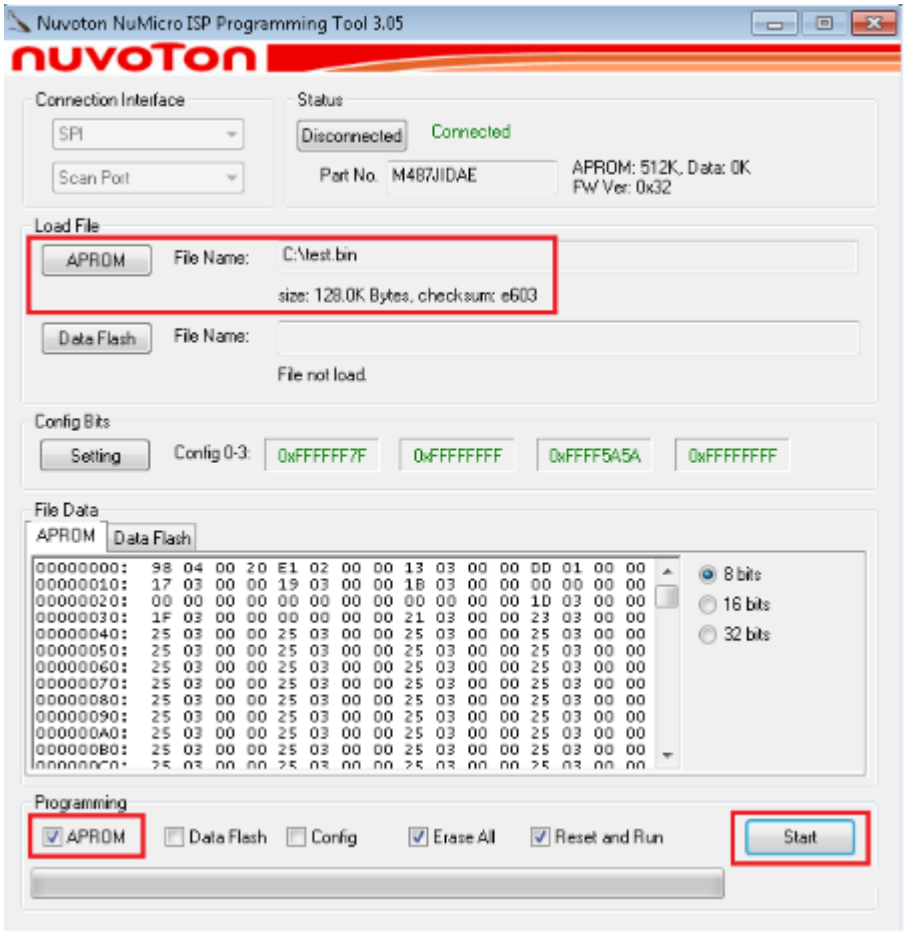

Figure 1-6 Program Data to Target Chip with SPI Interface

2 Code Description

2.1 ISP protocol

The ISP Receive/Send data packet is 64 bytes no matter ACK packet is received or command packet is sent. The data packet is sent and received via the I²C/SPI interface.

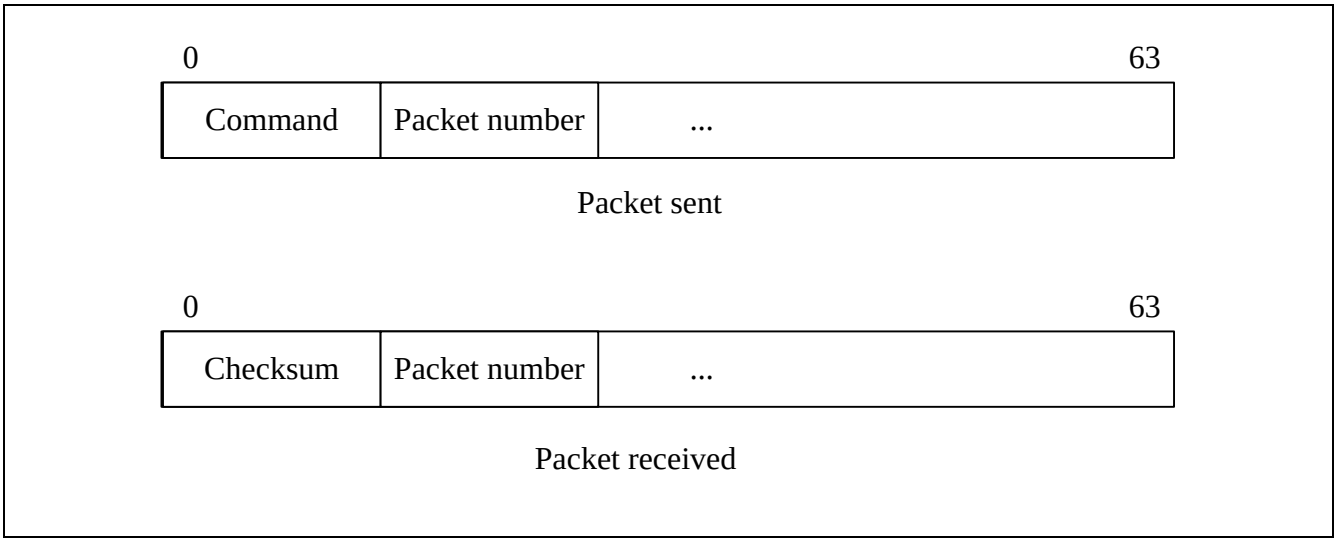

Figure 2-1 ISP Packet Sent and Received

After the USB host generates out token interrupt, the USB devices interrupt out callback function gets ISP IN packet buffer.

```
void EP3_Handler(void)  /* Interrupt OUT handler */
{
    uint8_t i;
    uint8_t *ptr;
    /* Interrupt OUT */
    ptr = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP3));

    for (i = 0; i < EP3_MAX_PKT_SIZE; i++)
        g_lib_CmdBuf[i] = ptr[i];

    g_lib_CmdFromTool = TRUE;
    USBD_SET_PAYLOAD_LEN(EP3, EP3_MAX_PKT_SIZE);
}
```

In the main loop routine, the ISP_Bridge_Main function will send and receive ISP packet via the I²C/SPI interface (command select).

```
void ISP_Bridge_Main(void)
{
```

```
    static uint32_t cks;
    static uint8_t cmd;

    // forward isp command to target device (I2C/SPI)
    if (g_lib_CmdFromTool == TRUE)
    {
    //Please refer the source code
    ...
    }

    // polling response for SPI/I2C interface
    if (g_lib_CmdToTarget && (g_lib_CmdFromTool == FALSE))
    {
    //Please refer the source code
    ...
    }
}
```

After the ISP received buffer moves to the USB IN buffer, the host will get it in USB devices interrupt in callback function.

```
void EP2_Handler(void)  /* Interrupt IN handler */
{
    uint8_t *ptr;
    /* Prepare the data for next HID IN transfer */
    ptr = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP2));
    USBD_MemCopy(ptr, (void *)&g_lib_AckBuf[0], EP2_MAX_PKT_SIZE);
    USBD_SET_PAYLOAD_LEN(EP2, EP2_MAX_PKT_SIZE);
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M031 Series CMSIS BSP V3.01.001
- IDE version
  - Keil uVersion 5.26

3.2 Hardware Requirements

- Circuit components
  - NuMaker-M032SE Ver1.3.
- Pin Connect

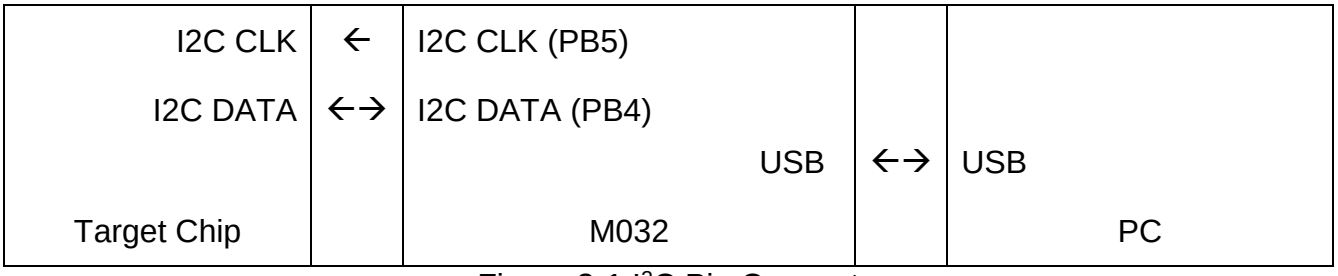

Figure 3-1 I²C Pin Connect

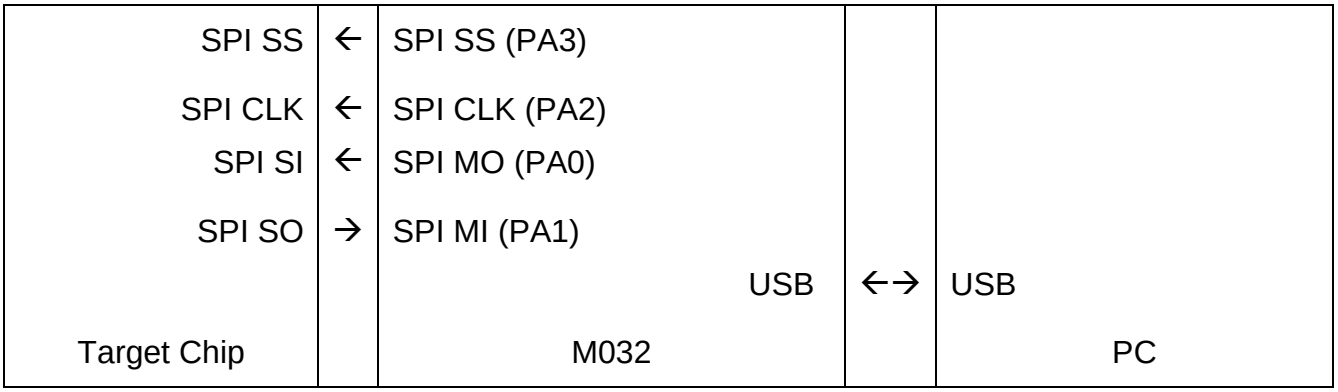

Figure 3-2 SPI Pin Connect

4 Directory Information

The directory structure is shown below.

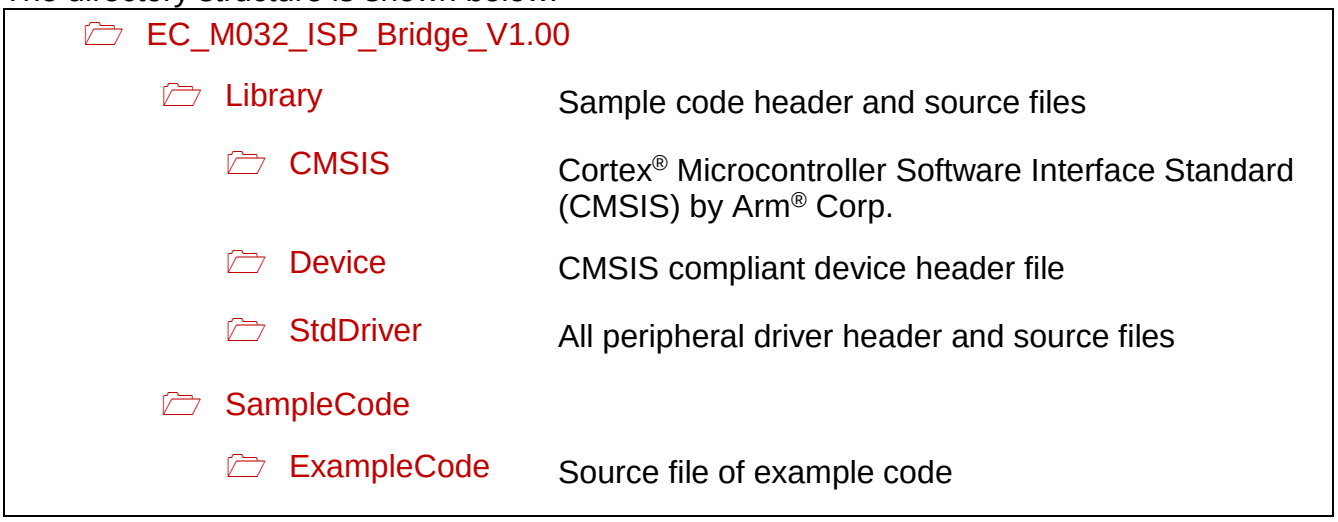

Figure 4-1 Directory Structure

5 Example Code Execution

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

6 Revision History

Date	Revision	Description
2021.12.10	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.

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.