NuMicro® ISP Programming Guide

Document Information

Abstract	Describe the ISP code architecture, I/O interfaces and the usage of ISP commands.
Apply to	NuMicro® Cortex-M Series.

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

www.nuvoton.com

Table of Contents

1 Overview

Most of modern consumer products has the capability to upgrade its firmware code. With this feature, products are able to continuously support new functions after it is released to end customer. Nuvoton provides an ISP (In-System Programming) tool to do firmware update.

Internal flash memory of NuMicro® chips can be divided into three regions, LDROM, APROM and data flash. The APROM usually saves the program code developed for specific application. And LDROM saves a special program such as ISP code which provides booting control and firmware upgrade functions. Data flash can be used to save some user data. After power ON, ISP program detects a pre-defined condition. If the condition is matched, ISP will be started. Otherwise, the control privilege will be passed to the application program in APROM. The pre-defined condition depends on product requirement and available hardware resource. Nuvoton ISP code supports USB, UART, I²C, SPI, RS485, CAN download interfaces. For UART, I²C, SPI, RS485, CAN interface, ISP supports auto-detect mode only, ISP will be started if NuvoISP.exe is running when boot from LDROM. ISP defines certain GPIO control pin and input LOW as the condition to enter ISP for the USB interface, user can upgrade the firmware after enter ISP. We release all ISP source code for user study or modification.

When application code is running, there's still a way to stop current execution and restart ISP code. The application program should firstly set boot option bit to "boot from LDROM", and then set SYSRESETREQ bit to reset NuMicro® MCU. Nuvoton provides an example application code. A command parser is implemented to interpret command received from I/O interface. When a "CMD_RUN_LDROM" command is received, program will generate reset to start execution from LDROM

The ISP program supports USB, UART, I²C, SPI, RS485, CAN as input/output interfaces. All interfaces share the same ISP command set, (except CAN, it uses only a few of the command set). On the other hand, I²C, SPI, RS485, CAN needs Nu-Link2 adapter as a bridge, users can refer to https://github.com/OpenNuvoton/Nuvoton_Tools (or in https://gitee.com) and Nu-Link2 adapter user manual for more bridge information. The commands and program code data are transmitted through the I/O interface. The ISP program updates the internal APROM according to the commands sent from Host. The type of Host may be a PC or microcontroller; it depends on the product feature. A windows-based application program, "*Nuvoton NuMicro ISP Programming Tool.exe"* is provided to demonstrate ISP function through USB and UART interfaces.

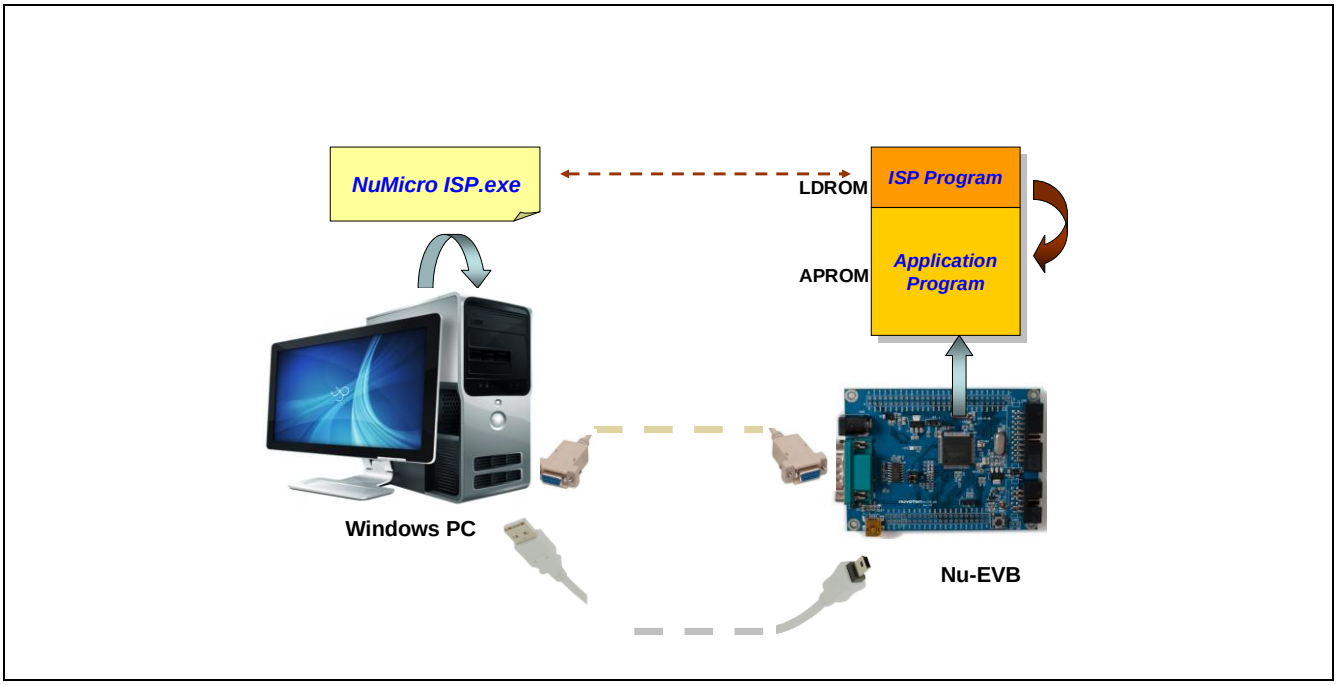

Figure 1.1 ISP Through USB and UART

The Nu-Link2-Pro can be used as an In-System Programming (ISP) programmer. It can work with *Nuvoton NuMicro ISP Programming Tool.exe*, or serve as a stand-alone (Offline) ISP programmer. It provides multi-interfaces bridge, such as UART, RS-485, USB, I²C, SPI, and CAN, to perform ISP function to NuMicro® Family microcontrollers.

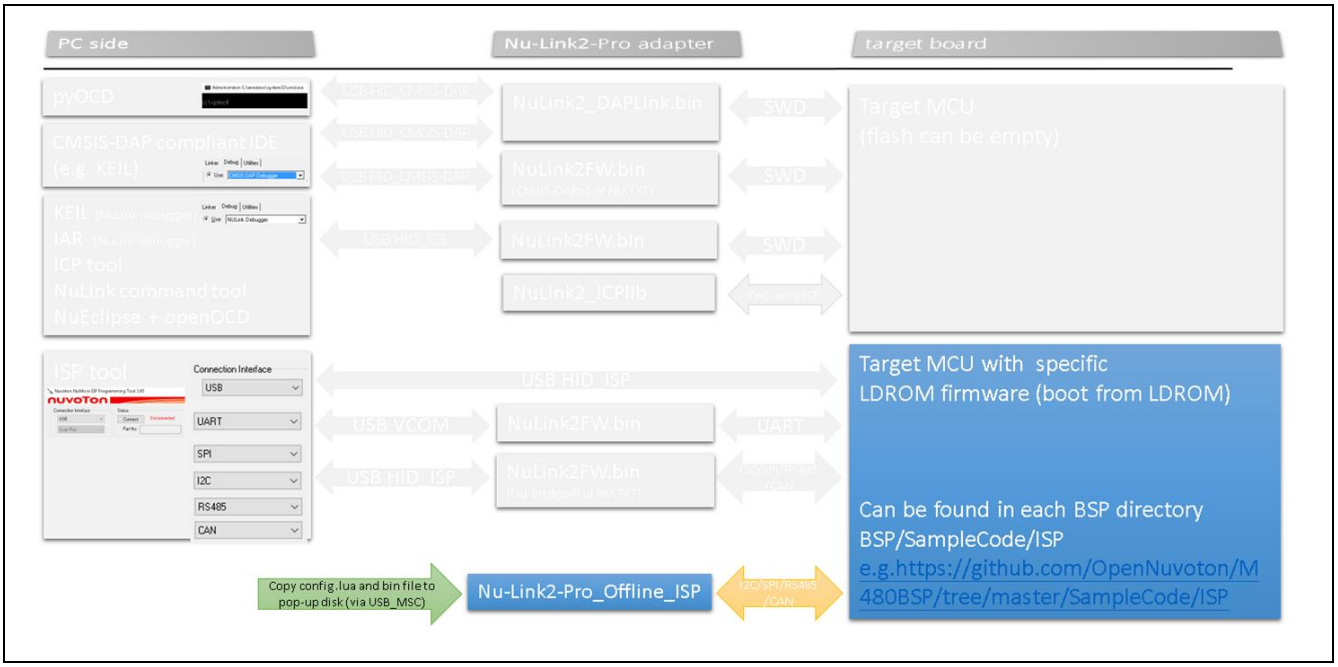

Figure 1.2 ISP Through Nu-Link2-Pro

1.1 Receive/Send data

ISP packet is 64 bytes fixed no matter ack packet received or command packet sent. If data want to send is less than 64 bytes, please pad zero.

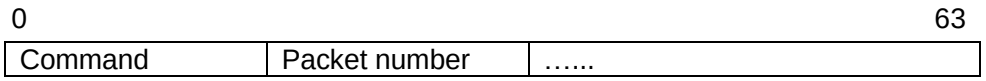

Figure 1.4 Packet sent

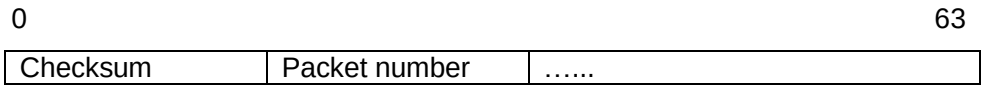

Figure 1.5 Packet Received

2 Pin Definition

The following content is the definition of the target board pin and the Nu-Link2-Pro adaptor.

2.1 Target Board Pin Definition

Download BSP sample code and find the sample code project which can be found in SampleCode\ISP. As shown in Figure 2.1, user can find which pins used for each interface by check the function SYS_Init() in main.c .The interfaces of ISP firmware sample code may be different for each NuMicro® chip series.

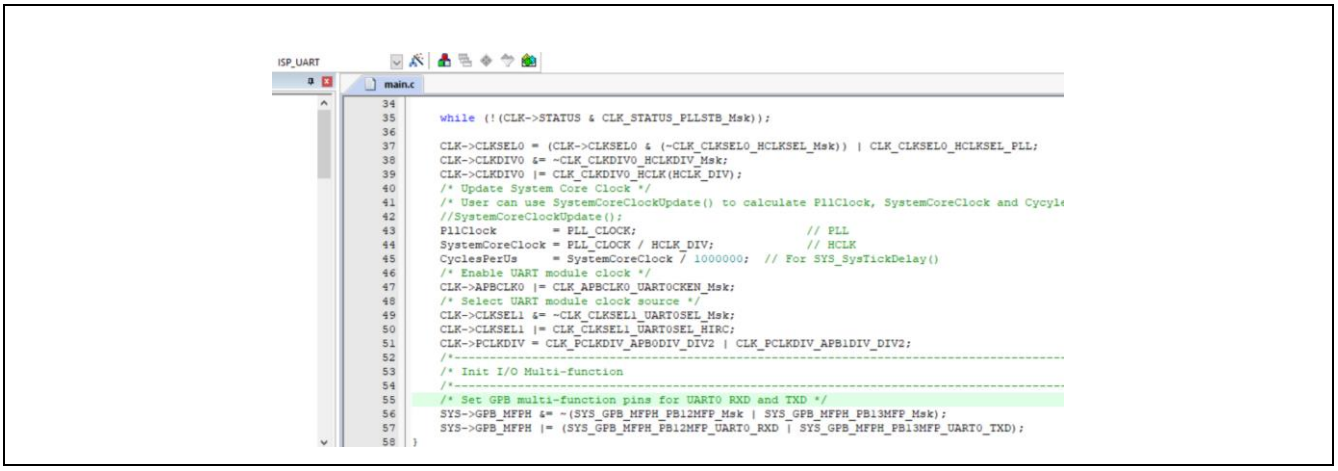

Figure 2.1 M480 BSP Sample: Pin Definition

2.2 Nu-Link2-Pro Pin Definition

The connection pin definition of Nu-Link2-Pro is show in Figure 2.2.

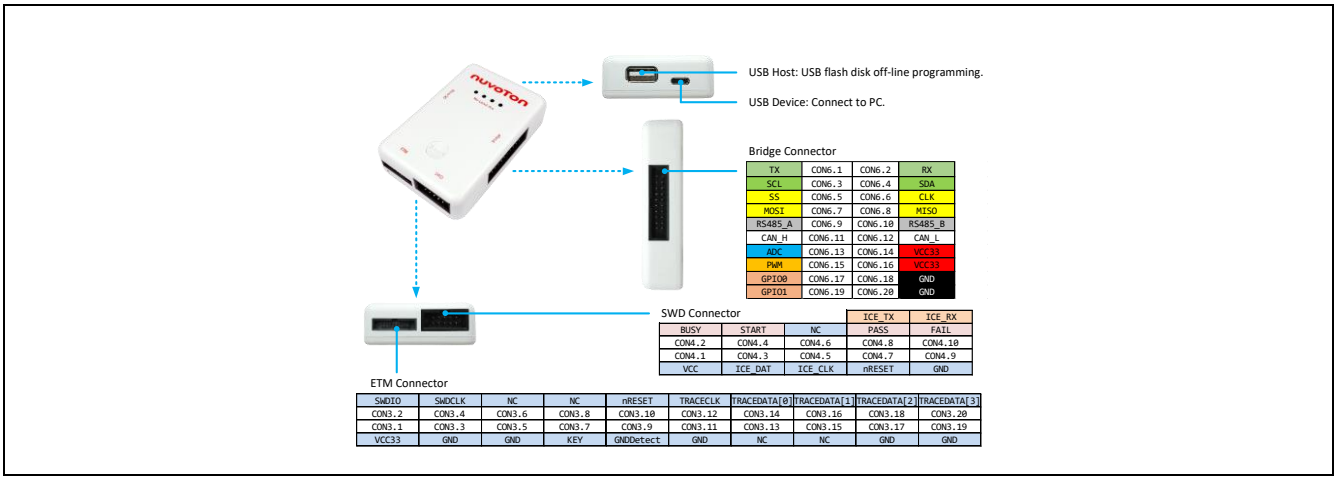

Figure 2.2 Nu-Link2-Pro Pin Definition

3 ISP Through USB and UART

User can connect the NuMicro® target chip with a Windows PC by USB. An ISP program PC application, *NuMicro ISP Programming Tool.exe,* is used to send command and data to target device. The *NuMicro ISP Programming Tool.exe* can download a new application code (in binary/hex format) to update the APROM.

Make sure the detect pin (different chips have different detect pin) is low for USB interface, and *NuMicro ISP Programming Tool.exe* is running on PC before test ISP function.

Target chip should have a ISP and an application program been saved in LDROM and APROM respectively. The command and data are transmitted via USB or UART0 interface.

3.1 Boot Loader Process

For the target chip, user can download the source files in BSP sample code and use Keil uVision IDE Tool to compile the programs which can be found in SampleCode\ISP\ISP_HID and SampleCode\ISP\ISP_UART respectively for USB and UART connection.

User can download the source files of ISP program PC application from GitHub, which can be found at directory "*NuMicro ISP Programming Tool Vx.xx*".

3.2 Communication Interface

The NuMicro® MCU ISP program imitates a HID device when using USB as the communication interface. The Interrupt IN and OUT pipes are used, and each transmitted packet size is fixed at 64 bytes.

When using UART communication interface, the ISP program initializes the UART0 interface to be in this setting (115200, n, 8, 1). The transmitted packet format through UART0 interface is the same as USB interface. Each packet size is fixed at 64 bytes, too.

3.3 The ISP Program

When ISP receives packet from UART or USB, it will parse the packet and take an action according to the transmitted command code. The commands set will be depicted in Chapter 8.

3.3.1 ISP Booting Flow

At booting phase, NuMicro® MCU is able to fetch code from either LDROM or APROM. It is controlled by config0 register. The "boot from LDROM" option in config0 register inside the NuMicro® MCU should be configured done before to run ISP programming. The ISP program will decide to enter ISP mode or pass the control privilege to APROM by checking the following conditions.

- The input state of GPIO detect pin is LOW for USB. Or receive packet from *NuMicro ISP Programming Tool.exe* for UART
- System starts according to a SYSRESETREQ event.

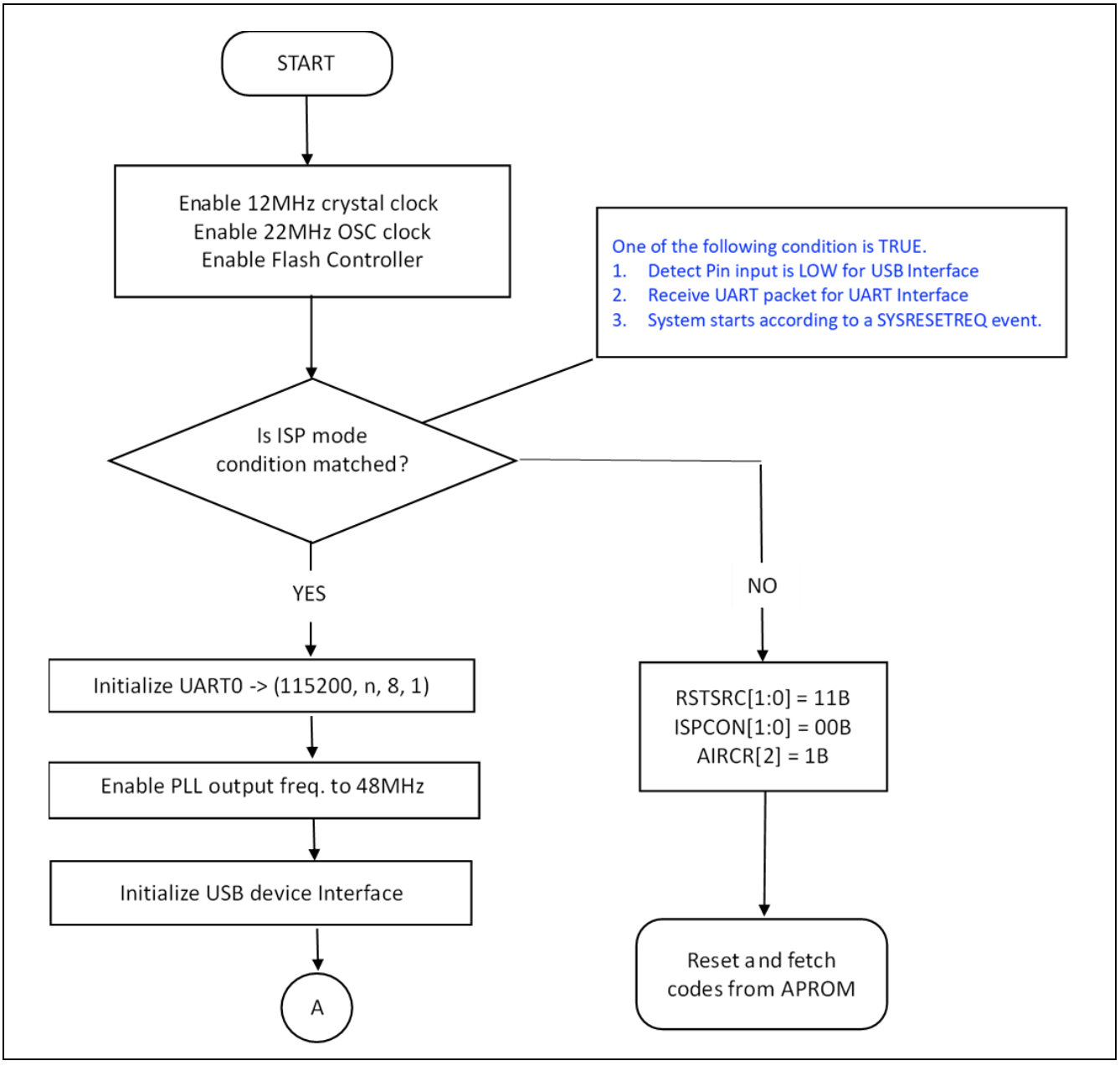

Figure 3.3 USB/UART ISP Booting Flow

3.3.2 ISP Main Flow

The ISP program goes into a command parsing loop whenever it detects that a ISP mode condition occurs. The ISP program recursively checks the commands received from UART0 and USB interfaces, and do proper operations. If one of reset commands is received, the ISP program will select the desired booting mode and reset whole system.

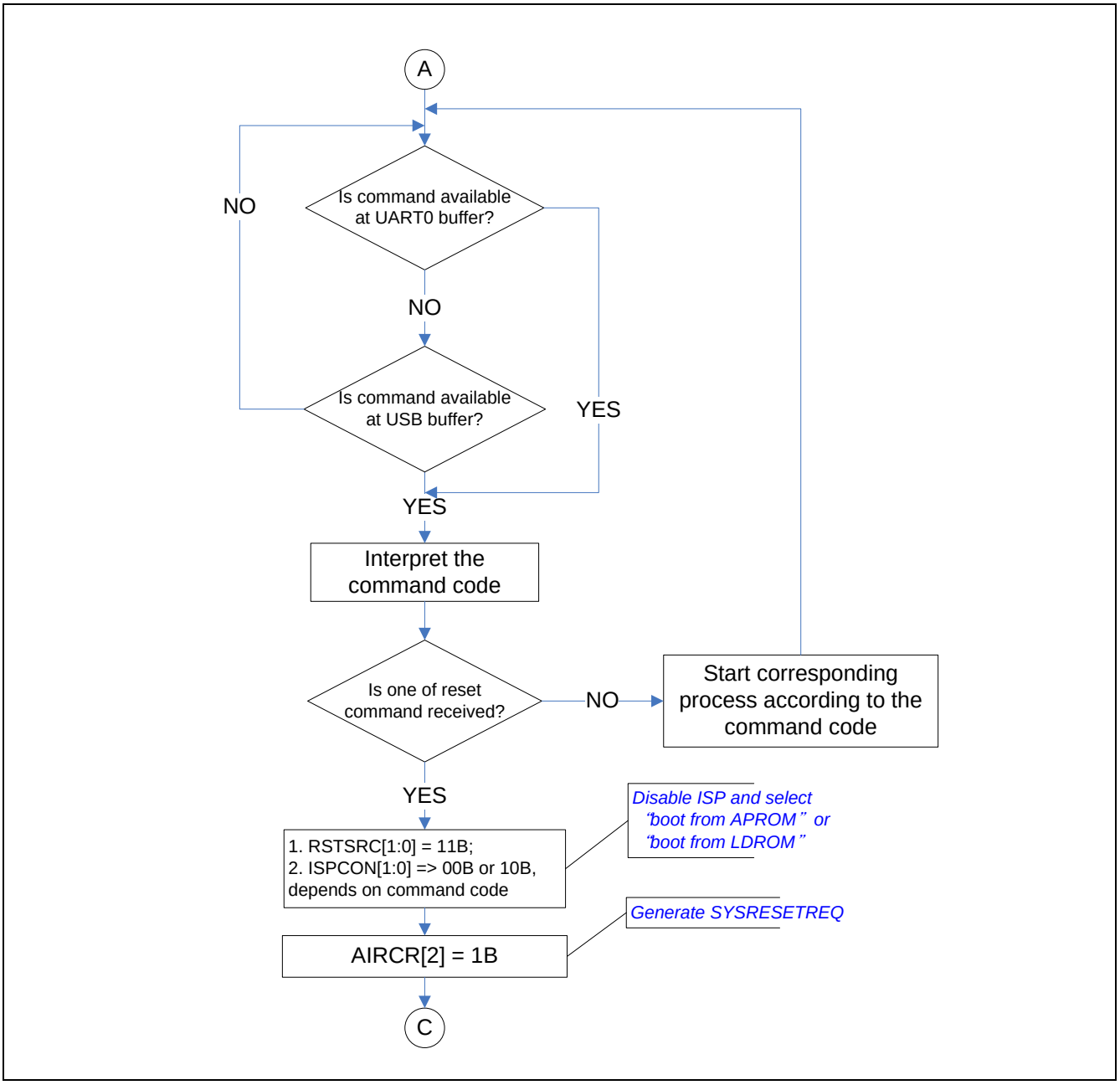

Figure 3.4 USB/UART ISP Main Flow

4 ISP Through SPI

User can connect the NuMicro® target chip with a Nu-Link2-Pro Adaptor, which is connect to a Windows PC by USB. An ISP program PC application, *NuMicro ISP Programming Tool.exe*, is used to send command and data to target device. The *NuMicro ISP Programming Tool.exe* can download a new application code (in binary/hex format) to update the APROM.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode, and *NuMicro ISP Programming Tool.exe* is running on PC before test ISP function.

Target chip should have a ISP and an application program been saved in LDROM and APROM respectively. The command and data are transmitted via SPI interface.

4.1 Project and Source Files

For the target chip, user can download the source files in BSP sample code and use Keil uVision IDE Tool to compile the programs which can be found in SampleCode\ISP\ISP_SPI.

User can download the source files of ISP program PC application from GitHub, which can be found at directory "*NuMicro ISP Programming Tool Vx.xx*".

4.2 Communication Interface

By using Nu-Link2-Pro, user can simply connect it to the target device and PC with ISP Tool to provide ISP SPI connection.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode. To set the Nu-Link2-Pro ISP-Bridge mode, connect the Nu-Link2-Pro to a Windows PC by USB. Then wait for PC to detect the NuMicro® MCU USB device. Edit the BRIDGE-MODE parameter to 2 in the file named as NU_CFG.TXT.

The connected method is shown as the following diagram.

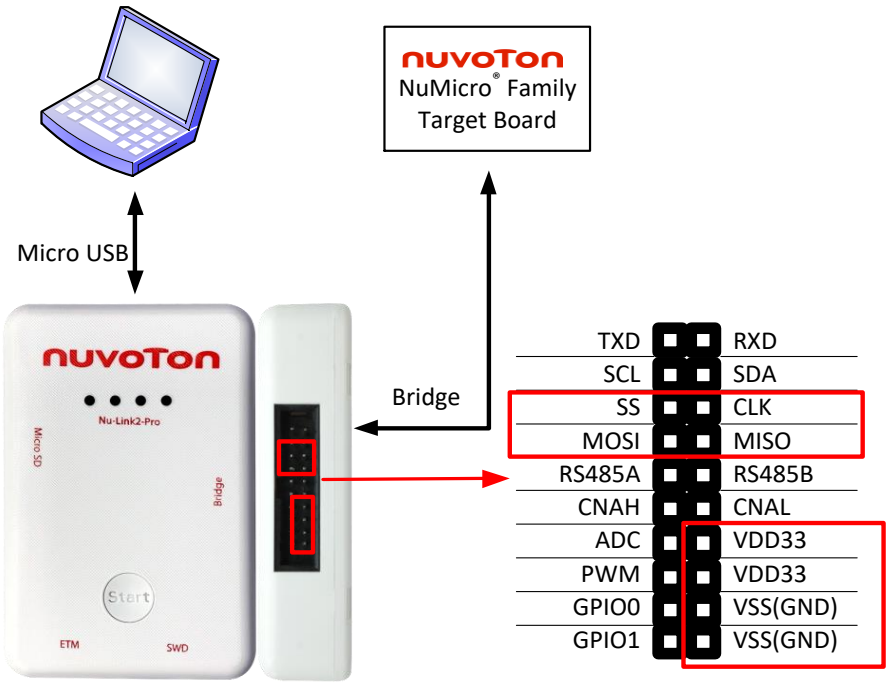

Figure 4.1 SPI Connect Block Diagram

4.3 The SPI ISP Program

When ISP receives packet from master, it will parse the packet and take an action according to the command code received. the size of packet is 64 bytes.

Different from USB/UART Connection, SPI ISP use the ISP-Bridge feature of Nu-Link2-Pro to parse the packet to ISP Tool PC Application.

4.3.1 ISP Booting Flow

The booting flow is similar to the ISP program using USB and UART. Please reference section 3.3.1 for detailed description of booting method.

ISP mode condition in SPI Interface is different from USB/UART Interface, device must receive a packet that contain “S”,” P”,” I” (receive buffer[0] & 0xFFFFFF00 = 0x53504900) to make sure that SPI Interface is successfully connect.

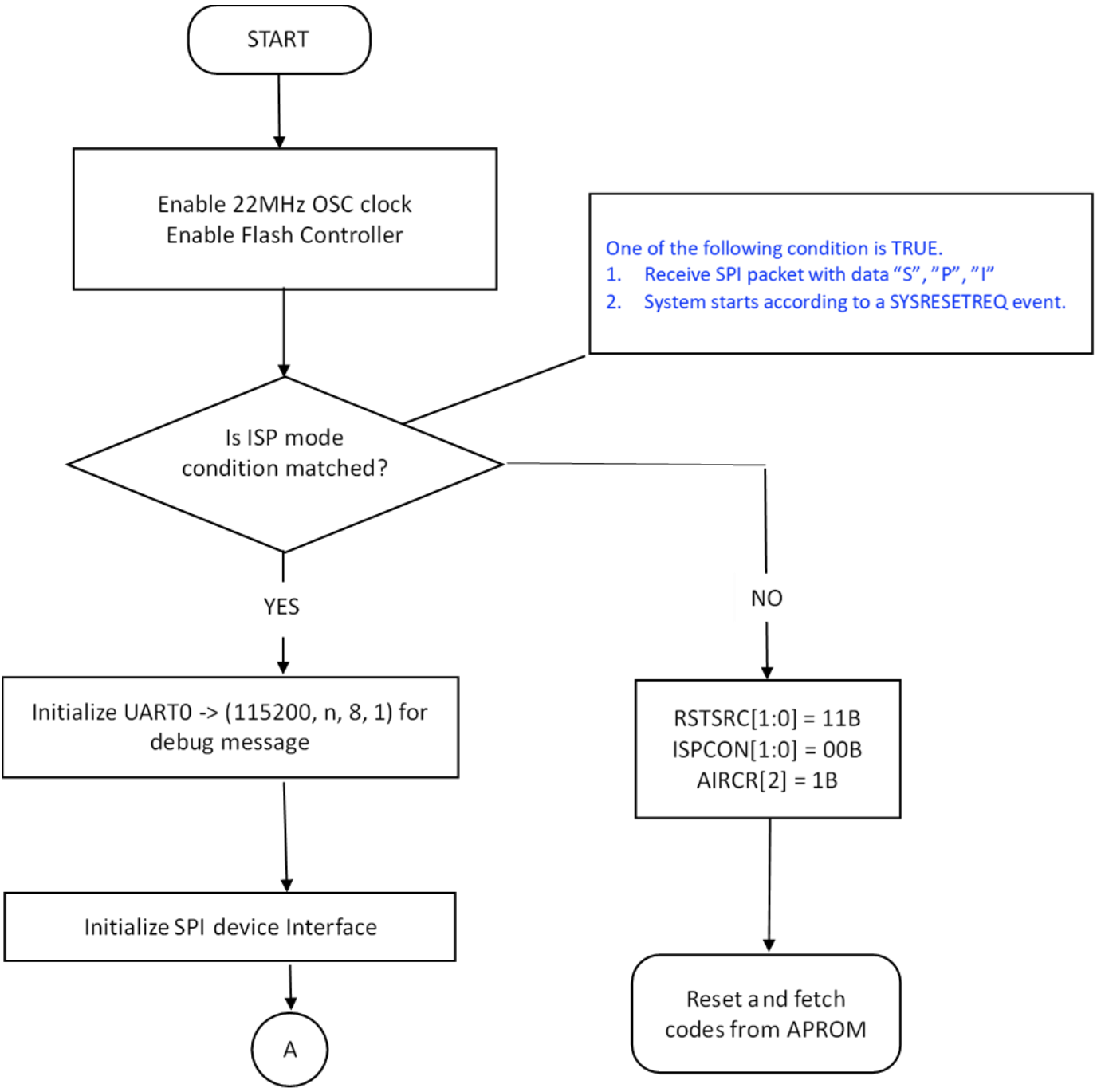

Figure 4.6 SPI ISP Booting Flow

4.3.2 ISP Main Flow

The ISP program goes into a command parsing loop whenever it detects that a ISP mode condition occurs. The ISP program recursively checks the commands received from SPI interface, and do proper operations. If one of reset commands is received, the ISP program will select the desired booting mode and reset whole system.

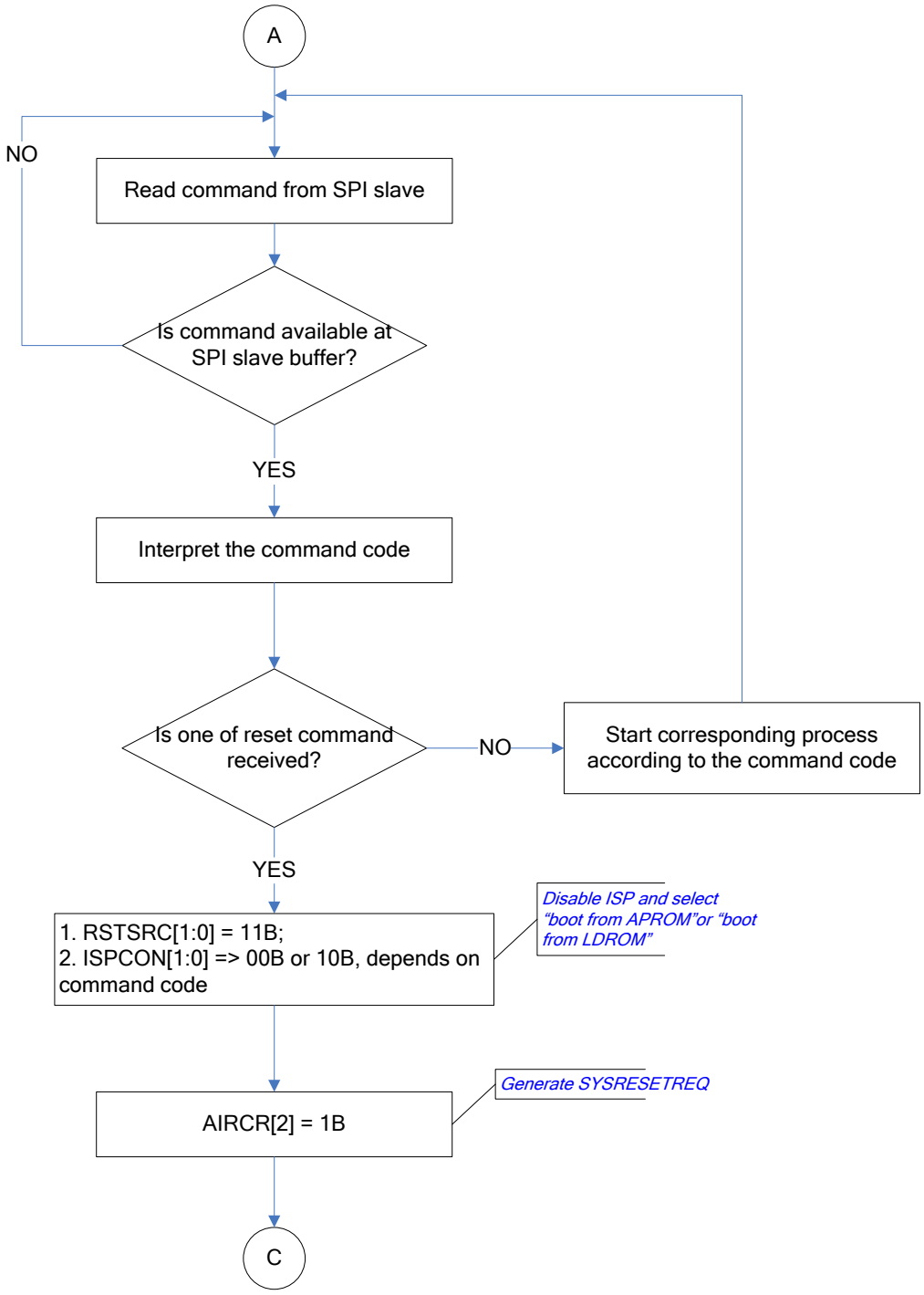

Figure 4.7 SPI ISP Main Flow

5 ISP Through I²C

User can connect the NuMicro® target chip with a Nu-Link2-Pro Adaptor, which is connect to a Windows PC by USB. An ISP program PC application, *NuMicro ISP Programming Tool.exe*, is used to send command and data to target device. *The NuMicro ISP Programming Tool.exe* can download a new application code (in binary/hex format) to update the APROM.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode, and *NuMicro ISP Programming Tool.exe* is running on PC before test ISP function.

Target chip should have a ISP and an application program been saved in LDROM and APROM respectively. The command and data are transmitted via I²C interface.

5.1 Project and Source Files

For the target chip, user can download the source files in BSP sample code and use Keil uVision IDE Tool to compile the programs which can be found in SampleCode\ISP\ISP_I2C.

User can download the source files of ISP program PC application from GitHub, which can be found at directory "*NuMicro ISP Programming Tool Vx.xx*".

5.2 Communication Interface

By using Nu-Link2-Pro, user can simply connect it to the target device and PC with ISP Tool to provide ISP I²C connection.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode. To set the Nu-Link2-Pro ISP-Bridge mode, connect the Nu-Link2-Pro to a Window PC by USB. Then wait for PC to detect the NuMicro® MCU USB device. Edit the BRIDGE-MODE parameter to 2 in the file named as NU_CFG.TXT.

The connected method is shown as the following diagram.

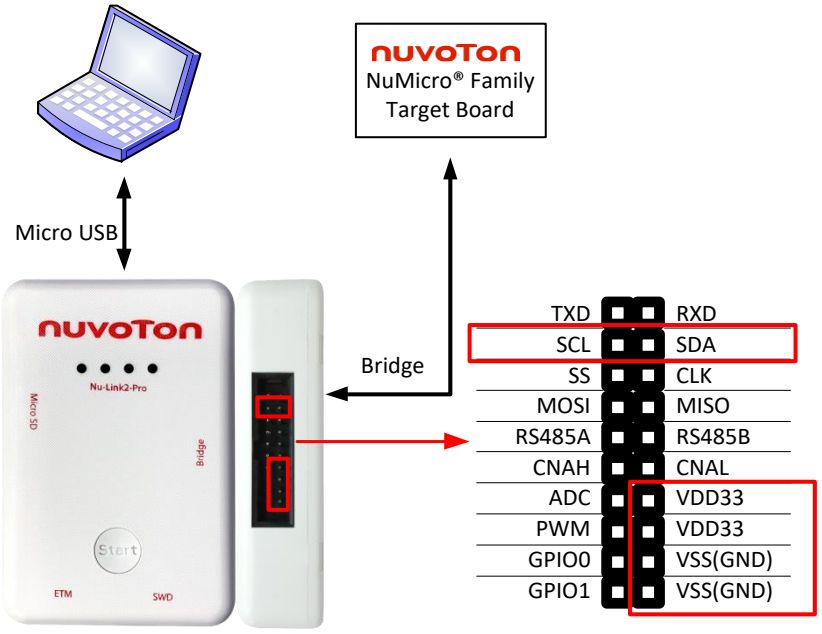

Figure 5.1 I²C Connection Block Diagram

5.3 The I²C ISP Program

When ISP receives packet from I²C master program, it will parse the packet and take an action according to the command code received. the size of packet is 64 bytes.

5.3.1 ISP Booting Flow

The booting flow is similar to the ISP program using USB, UART and SPI Interface. Please reference section 3.3.1 for detailed description of booting method.

Unlike SPI Interface, I²C Interface only check that receive data length to make sure I²C connection is working.

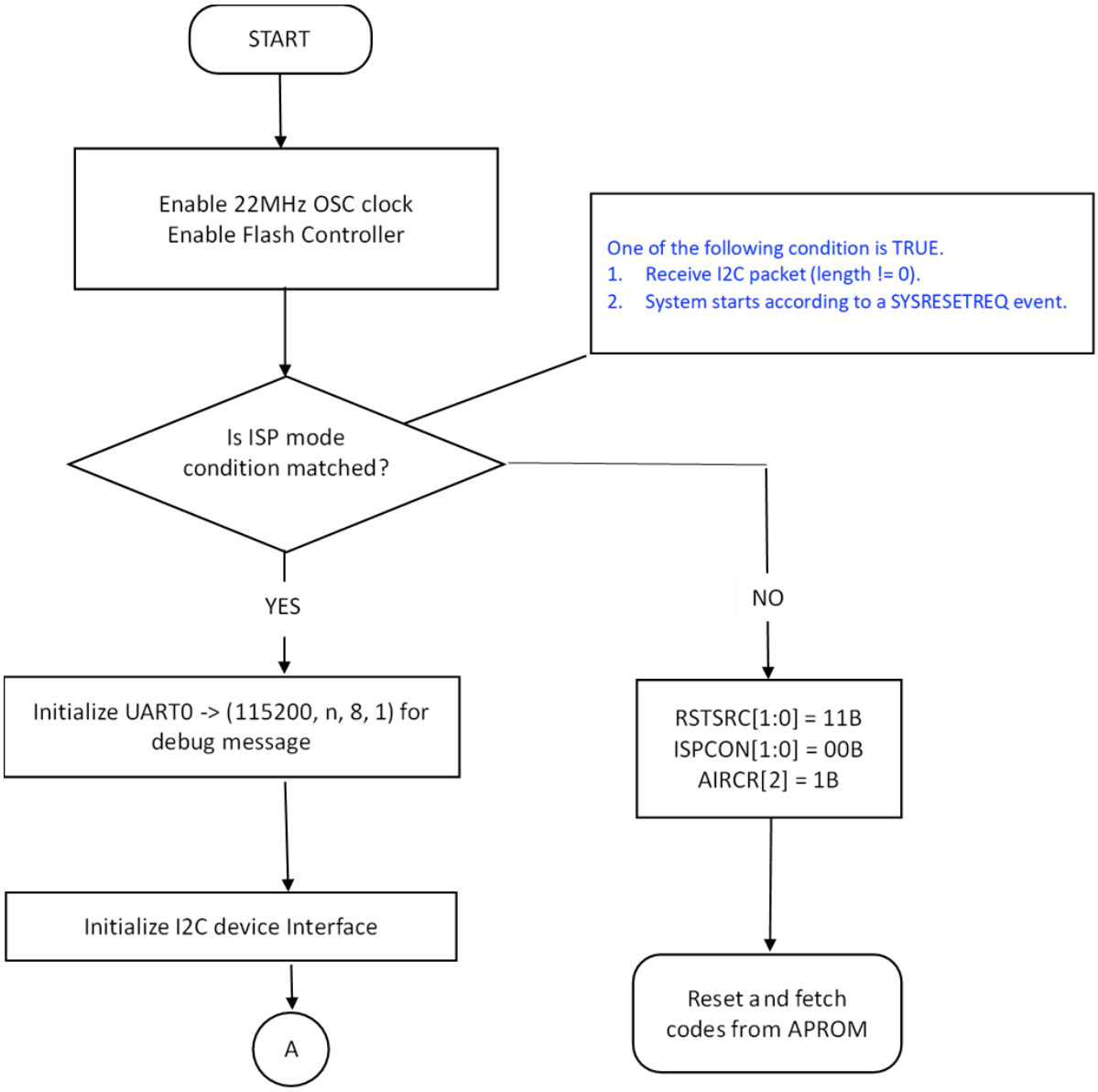

Figure 5.2 I²C ISP Booting Flow

5.3.2 ISP Main Flow

The ISP program goes into a command parsing loop whenever it detects that a ISP mode condition occurs. The ISP program recursively checks the commands received from I²C, and do proper operations. If one of reset commands is received, the ISP program will select the desired booting mode and reset whole system.

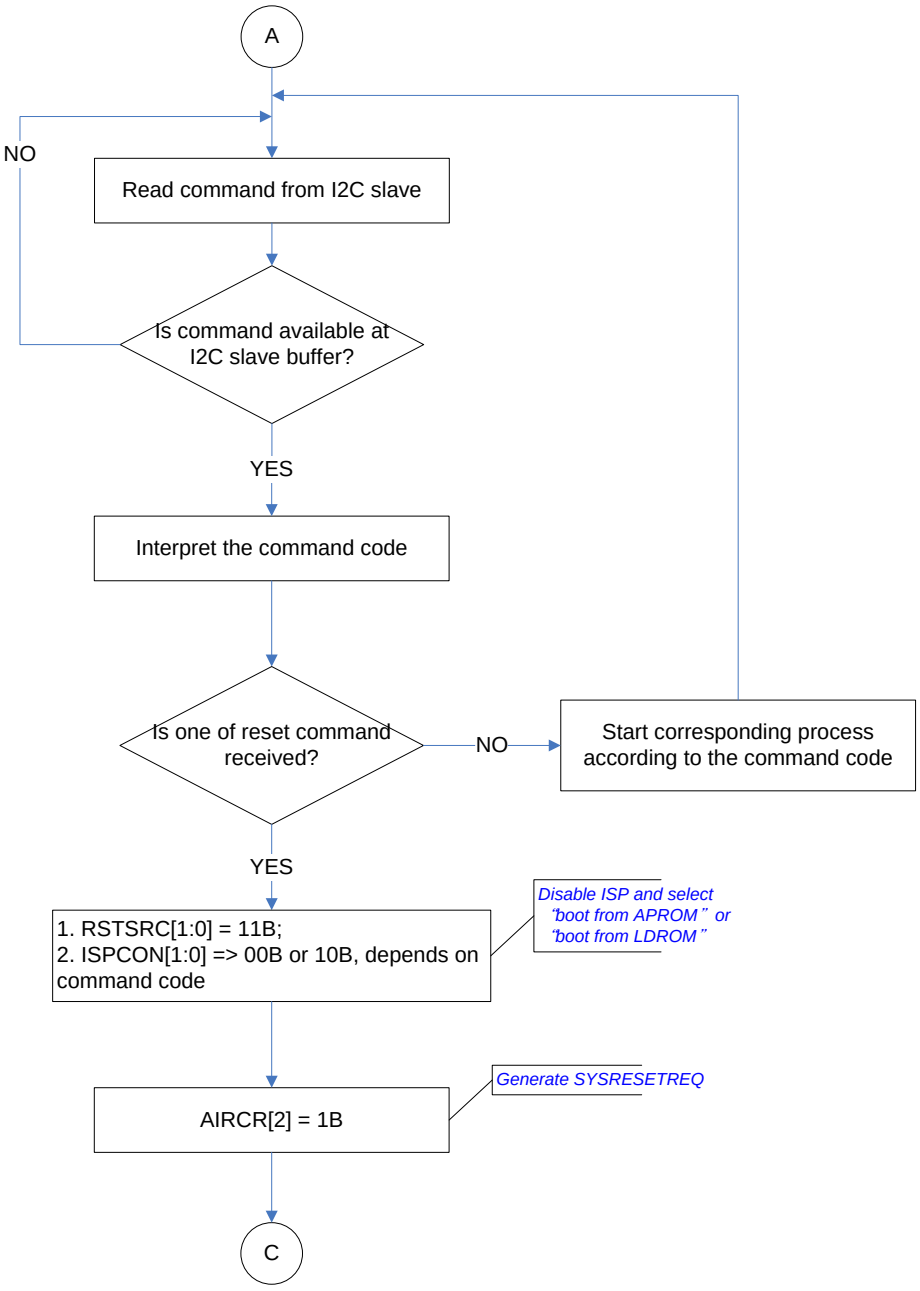

Figure 5.3 I²C ISP Main Flow

6 ISP Through RS485

User can connect the NuMicro® target chip with a Nu-Link2-Pro Adaptor, which is connect to a Windows PC by USB. An ISP program PC application, *NuMicro ISP Programming Tool.exe*, is used to send command and data to target device. The *NuMicro ISP Programming Tool.exe* can download a new application code (in binary/hex format) to update the APROM.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode, and *NuMicro ISP Programming Tool.exe* is running on PC before test ISP function.

Target chip should have a ISP and an application program been saved in LDROM and APROM respectively, and target device should have a RS485 transceiver. The command and data are transmitted via RS485 interface.

6.1 Project and Source Files

For the target chip, user can download the source files in BSP sample code and use Keil uVision IDE Tool to compile the programs which can be found in SampleCode\ISP\ISP_RS485.

User can download the source files of ISP program PC application from GitHub, which can be found at directory "*NuMicro ISP Programming Tool Vx.xx*".

6.2 Communication Interface

By using Nu-Link2-Pro, user can simply connect it to the target device and PC with ISP Tool to provide ISP RS485 connection.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode. To set the Nu-Link2-Pro ISP-Bridge mode, connect the Nu-Link2-Pro to a Window PC by USB. Then wait for PC to detect the NuMicro® MCU USB device. Edit the BRIDGE-MODE parameter to 2 in the file named as NU_CFG.TXT.

The connected method is shown as the following diagram.

For SPI interface, we suggest connecting as many GND pins as you can, which helps stability.

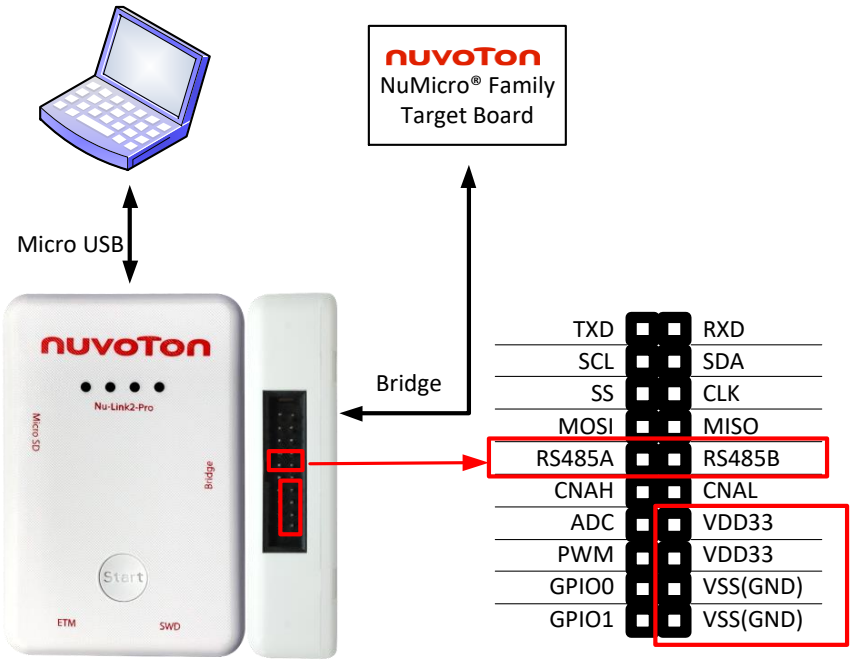

Figure 6.1 RS485 Connection Block Diagram

6.3 The RS485 ISP Program

When ISP receives packet from RS485 master program, it will parse the packet and take an action according to the command code received. the size of packet is 64 bytes.

6.3.1 ISP Booting Flow

The booting flow is similar to the ISP program using UART. Please reference section 3.3.1 for detailed description of booting method.

In Nu-Link2-Pro, an UART Interface is used to implement RS485 connection. Since UART Interface is an idle high interface, in order to correctly control the signal, a tri-state transceiver should be set on the target device.

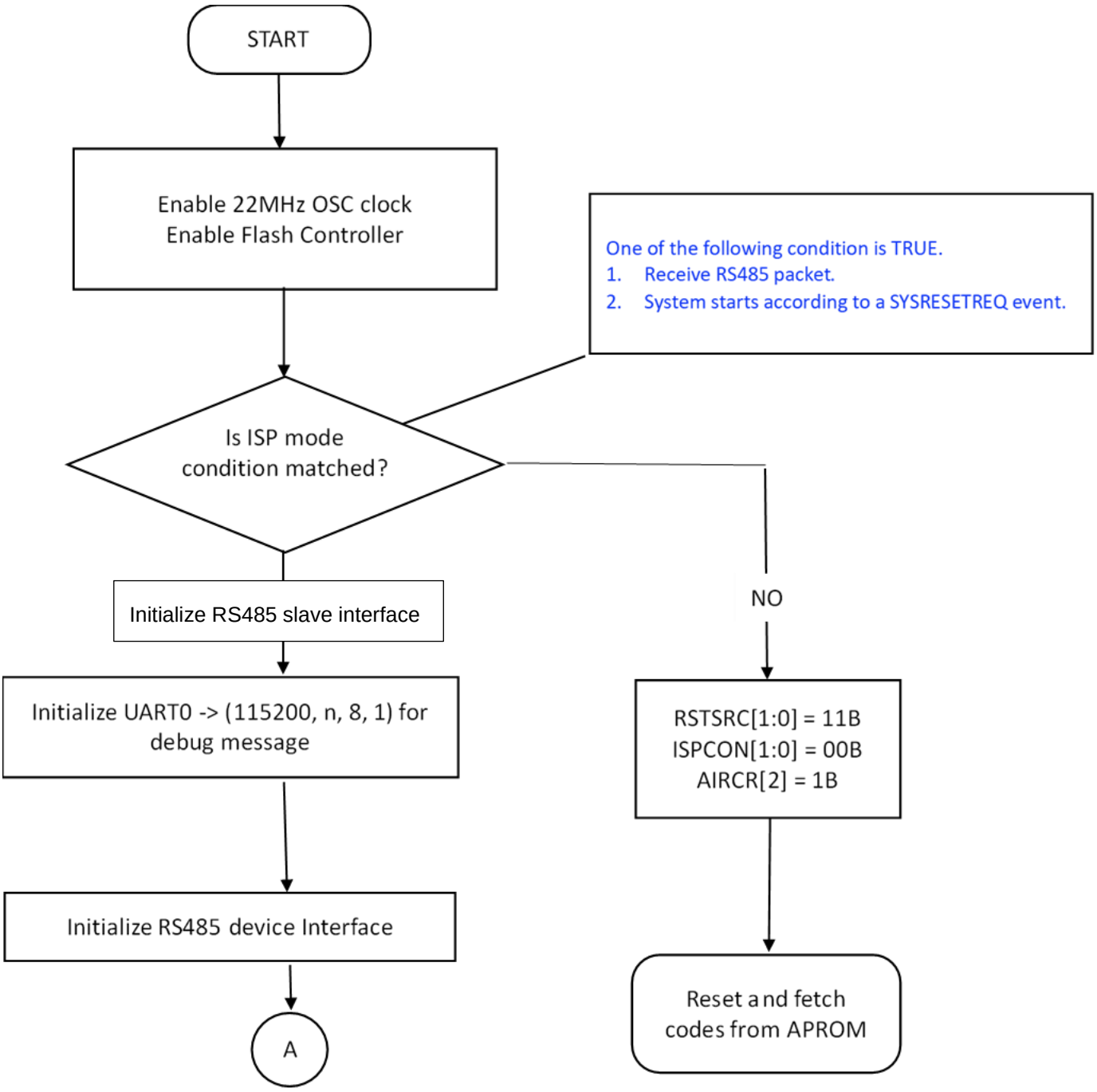

Figure 6.2 RS485 ISP Booting Flow

6.3.2 ISP Main Flow

The ISP program goes into a command parsing loop whenever it detects that a ISP mode condition occurs. The ISP program recursively checks the commands received from RS485, and do proper operations. If one of reset commands is received, the ISP program will select the desired booting mode and reset whole system.

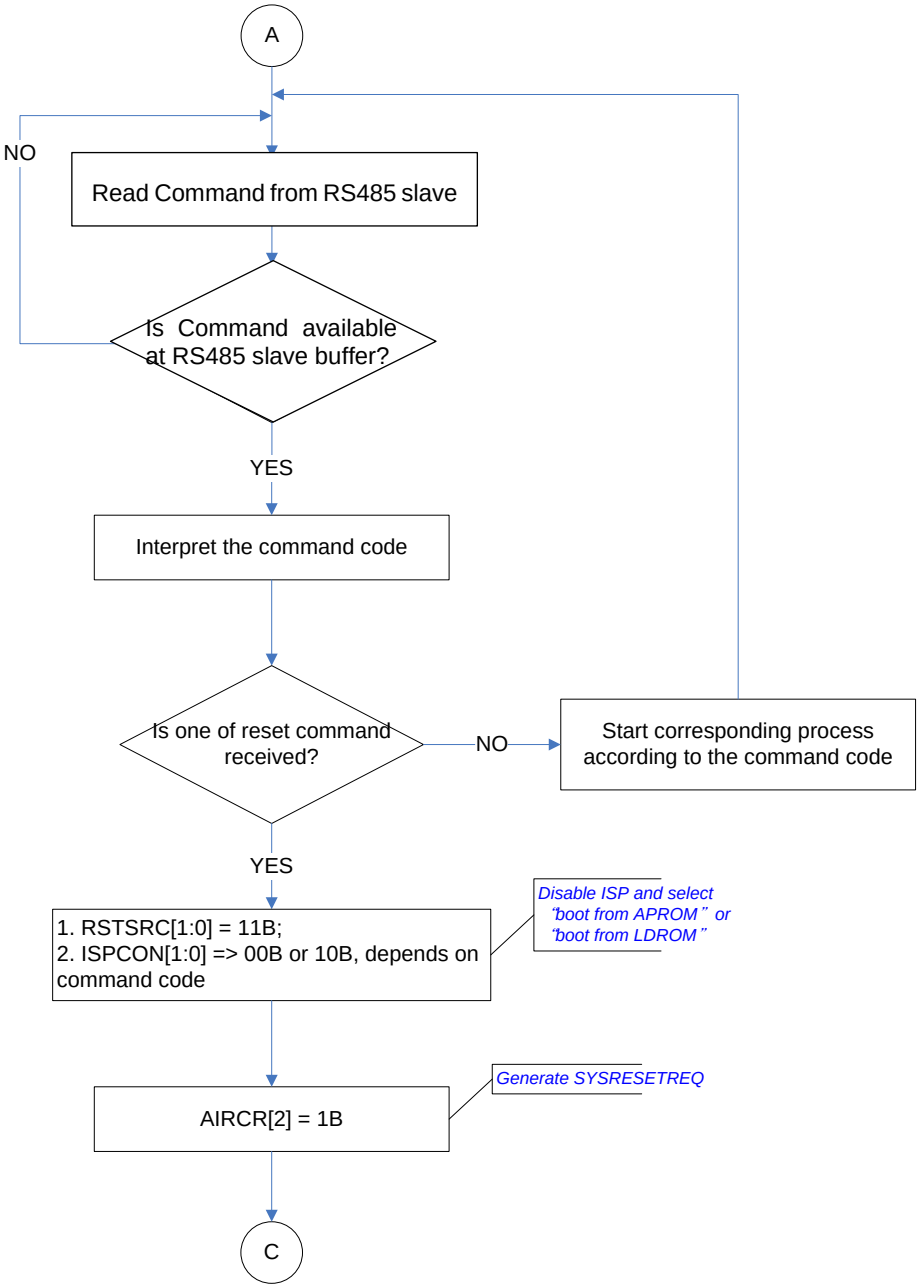

Figure 6.3 RS485 ISP Main Flow

7 ISP Through CAN

User can connect the NuMicro® target chip with a Nu-Link2-Pro Adaptor, which is connect to a Windows PC by USB. An ISP program PC application, *NuMicro ISP Programming Tool.exe*, is used to send command and data to target device. The *NuMicro ISP Programming Tool.exe* can download a new application code (in binary/hex format) to update the APROM.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode, and *NuMicro ISP Programming Tool.exe* is running on PC before test ISP function.

Target chip should have a ISP and an application program been saved in LDROM and APROM respectively. The command and data are transmitted via CAN interface.

Different from other connection type, CAN connection has lesser ISP command. To check which kind of ISP command is available in CAN connection, please check the content of Chapter 8.

7.1 Project and Source Files

For the target chip, user can download the source files in BSP sample code and use Keil uVision IDE Tool to compile the programs which can be found in SampleCode\ISP\ISP_CAN.

User can download the source files of ISP program PC application from GitHub, which can be found at directory “NuMicro ISP Programming Tool Vx.xx”.

7.2 Communication Interface

By using Nu-Link2-Pro, user can simply connect it to the target device and PC with ISP Tool to provide ISP CAN connection.

Make sure the Nu-Link2-Pro is setting as ISP-Bridge mode. To set the Nu-Link2-Pro ISP-Bridge mode, connect the Nu-Link2-Pro to a Window PC by USB. Then wait for PC to detect the NuMicro® MCU USB device. Edit the BRIDGE-MODE parameter to 2 in the file named as NU_CFG.TXT.

The connected method is shown as the following diagram.

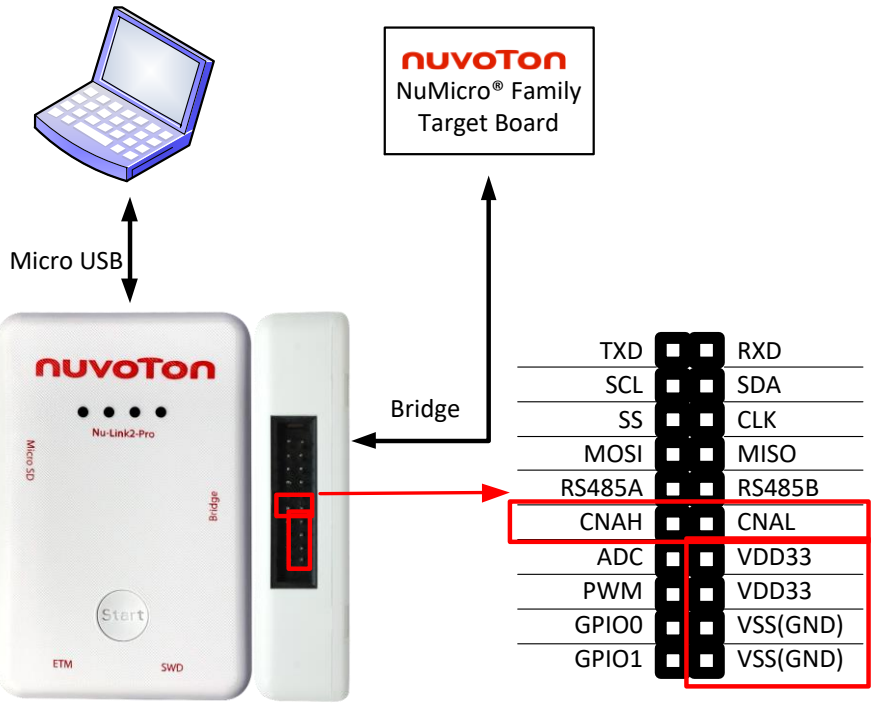

Figure 7.1 CAN Connection Block Diagram

7.3 The CAN ISP Program

When ISP receives packet from CAN master program, it will parse the packet and take an action according to the command code received. the size of packet is 64 bytes.

7.3.1 ISP Booting Flow

The booting flow is similar to the ISP program using USB and UART. Please reference section 3.3.1 for detailed description of booting method.

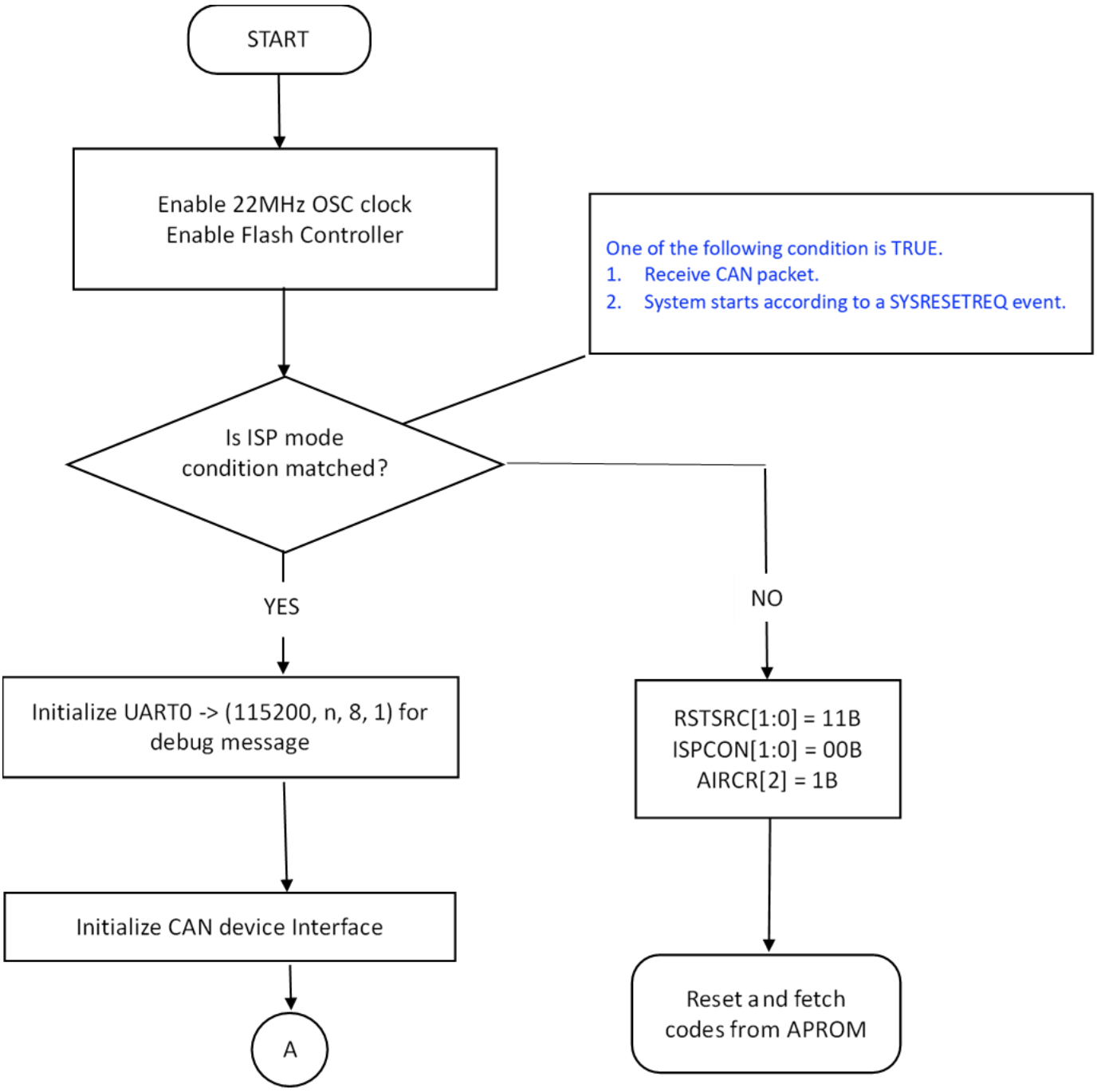

Figure 7.2 CAN ISP Booting Flow

7.3.2 ISP Main Flow

The ISP program goes into a command parsing loop whenever it detects that a ISP mode condition occurs. The ISP program recursively checks the commands received from CAN, and do proper operations. If one of reset commands is received, the ISP program will select the desired booting mode and reset whole system.

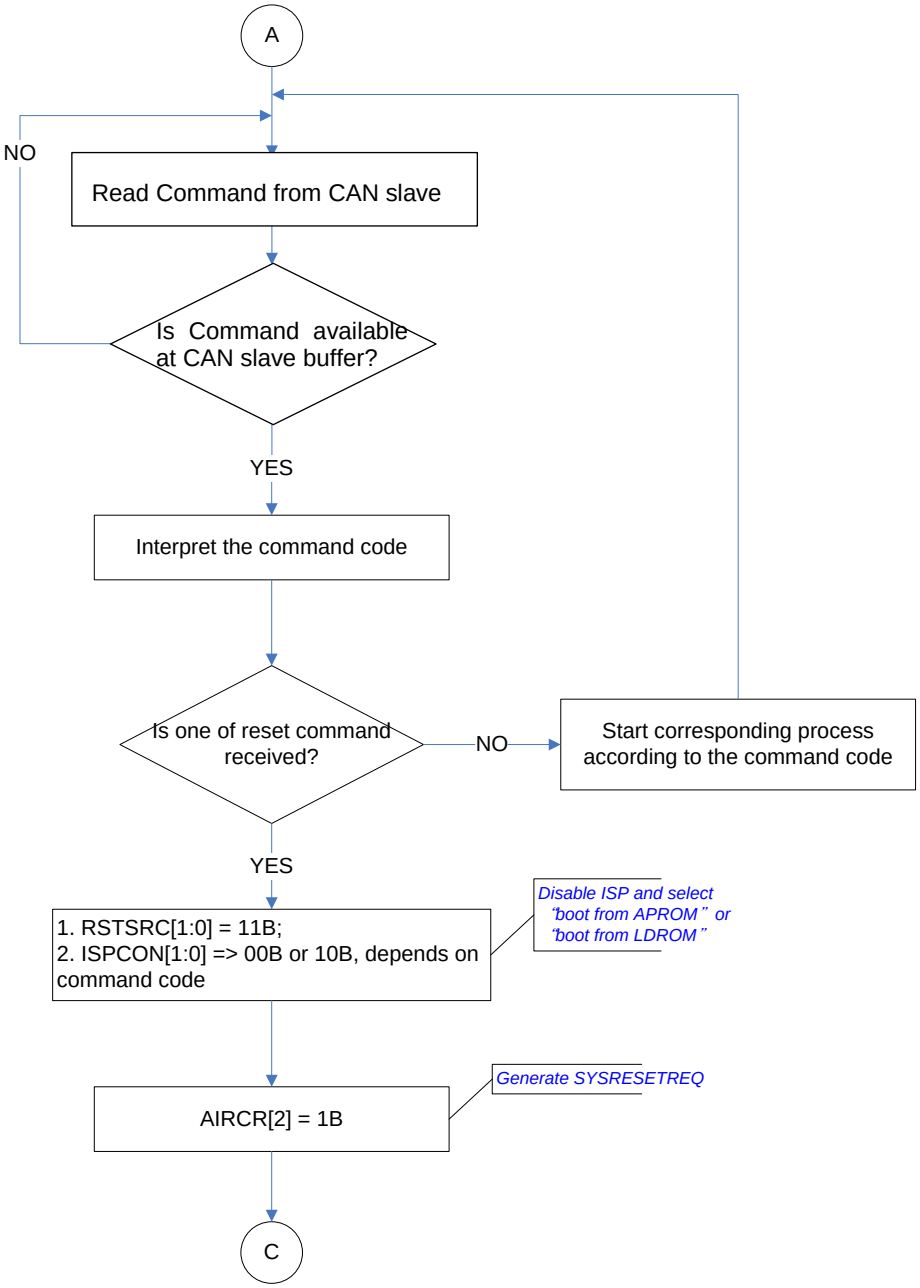

Figure 7.3 CAN ISP Main Flow

8 ISP Command Set

ISP packet is 64 byte fixed no matter command packet or ack packet. The ISP supported commands are listed as Figure 8.1 below.

Command	Command Code	ACK	Command Description
CMD_UPDATE_APROM	0x000000A0	Yes	Within the range 0~MAX_APROM_SIZE, program APROM from address user specified.
CMD_UPDATE_CONFIG	0x000000A1	Yes	Write Config0 and Config1 registers of Flash memory.
CMD_READ_CONFIG	0x000000A2	Yes	Get Config0 and Config1
CMD_ERASE_ALL	0x000000A3	Yes	Erase all APROM, including Data Flash in Flash memory and Config area. The Config registers to restored to default value.
CMD_SYNC_PACKNO	0x000000A4	Yes	Synchronize packet number with NuMicro® MCU microcontrollers before a valid command send.
CMD_GET_FWVER	0x000000A6	Yes	Get version information of ISP firmware.
CMD_GET_DEVICEID	0x000000B1	Yes	Get chip product ID.
CMD_UPDATE_DATAFLASH	0x000000C3	Yes	Program APROM from address user specified, before program, erase the corresponding sector.
CMD_RUN_APROM	0x000000AB	No	Instruct ISP to boot from APROM.
CMD_RUN_LDROM	0x000000AC	No	Instruct ISP to boot from LDROM.
CMD_RESET	0x000000AD	No	Instruct ISP to reboot.
CMD_WRITE_CHECKSUM	0x000000C9	Yes	Instruct ISP to write application length and checksum in APROM to the last 8 bytes of APROM.
CMD_GET_FLASHMODE	0x000000CA	Yes	Get boot selection.
CMD_RESEND_PACKET	0x000000FF	Yes	
CMD_CONNECT	0x000000AE	Yes	Test whether or not the ISP is active

Figure 8.1 ISP Command Set

Sync Packet Number

Before any command is sent, a sync packet number operation (*CMD_SYNC_PACKNO)* is required, the aim is used for guarantee the correctness of sequential packets, and avoid packet received by ISP and PC from duplication, if the sync operation fail, and then communication failed between PC and NuMicro® MCU.

Refer to *8.5 Command CMD_SYNC_PACKNO* for detailed.

Communication Correctness

For distinguishing older data from newer data, a sequential packet number is provided. It increases by one when sending one packet (for example, if ISP receive packet number=1 from PC, it will send packet number=2 back to PC, and wait for the package that packet number is equal to 3. and so on), PC should compare packet number received with local value, if it isn't equal, discard this packet.

Avoiding R/W data error, a checksum is added in ACK packet, and when all data received, a checksum of *Total Length* data will be returned.

CAN Interface Command Set

Please remind that the command set of the CAN interface is different than other interface connection. The command set for CAN interface is listed as Figure 8.2 below. Although the command code are different, the CAN version command do similar operation as basic ISP command.

Command	Command Code	ACK	Command Description
CMD_RUN_APROM	0xAB000000	No	Instruct ISP to boot from APROM.
CMD_GET_DEVICEID	0xB1000000	Yes	Get chip product ID.
CMD_READ_CONFIG	0xA2000000	Yes	Get Config0 and Config1
CMD_UPDATE_APROM	Write Address instead	Yes	Within the range 0~MAX_APROM_SIZE, program APROM from address user specified.

Figure 8.2 ISP Command Set for CAN Interface

8.1 Command CMD_UPDATE_APROM (0xA0)

Command Format

Host Send (named Format1, unit by byte):

4	4	4	4	-
CMD	Packet Number	Start Address	Total Length	Data……

If data can’t put into one packet, and then packet format is as following (named Format2)

0x00000000	Packet Number	Data……

Device ACK:

4	4	
Checksum	Packet Number	Total Checksum

Description

This command is used to instruct ISP to program APROM flash. The command format and data transfer format is described as in “Command Format”, user can specify start address and length to be programmed. If the program data is too big to be transferred in a packet, user can split the data into several packets by multiple transfers.

When ISP received the command packet, it erases all contents in APROM flash excluding data flash, and then reads the program data from received packet, and programs them into the specified address of flash memory. After program operation is completed, it calculates the checksum of the whole package received, then transmit ACK packet to the host.

After ACK packet is received, host will check the checksum is right or wrong, if checksum is right, it indicates that the data transfer is successful. After the operation is finished, then user can continue to run another different operation or transmit another data packet.

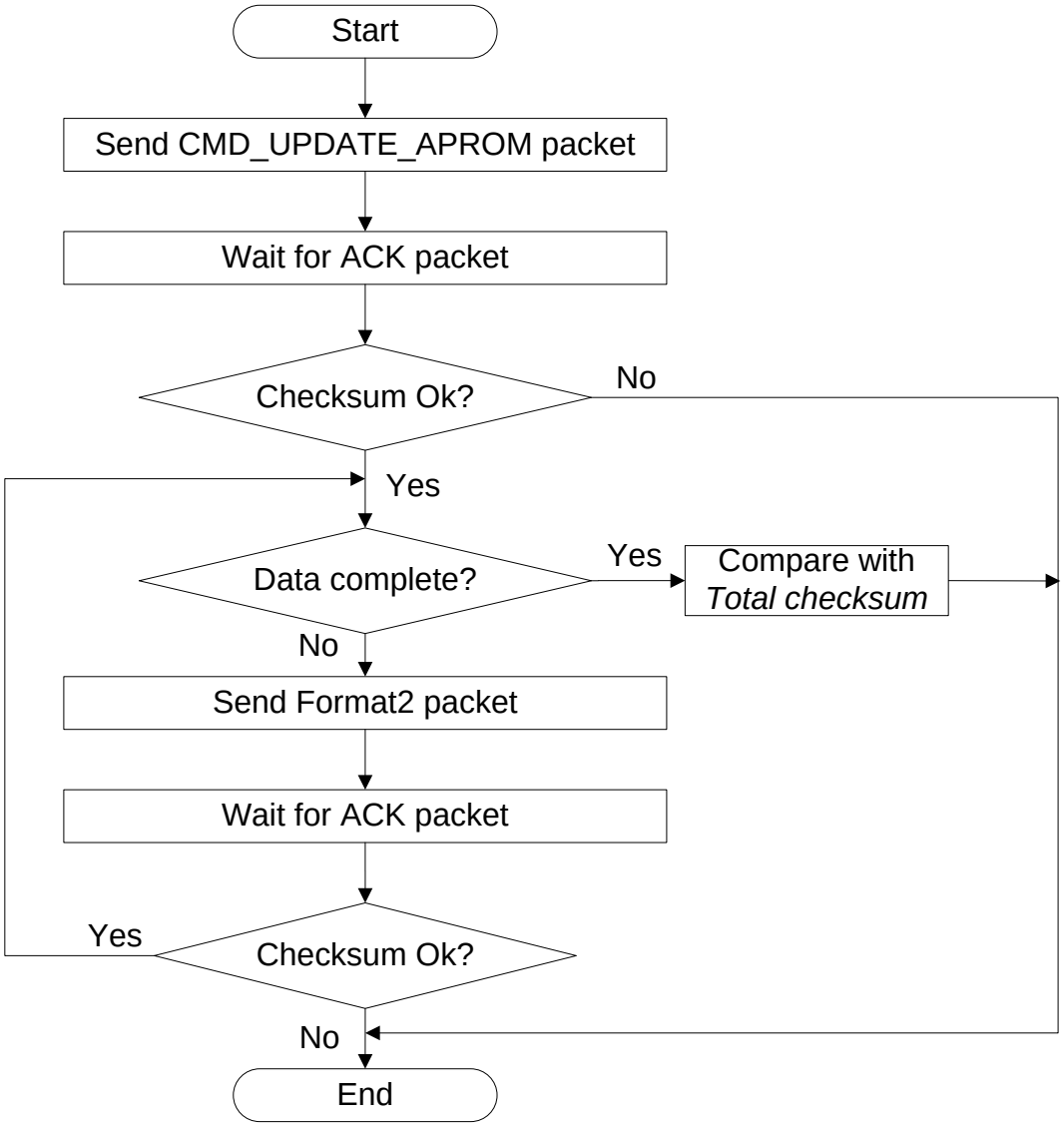

Figure 8.3 Program APROM Command: Host Side

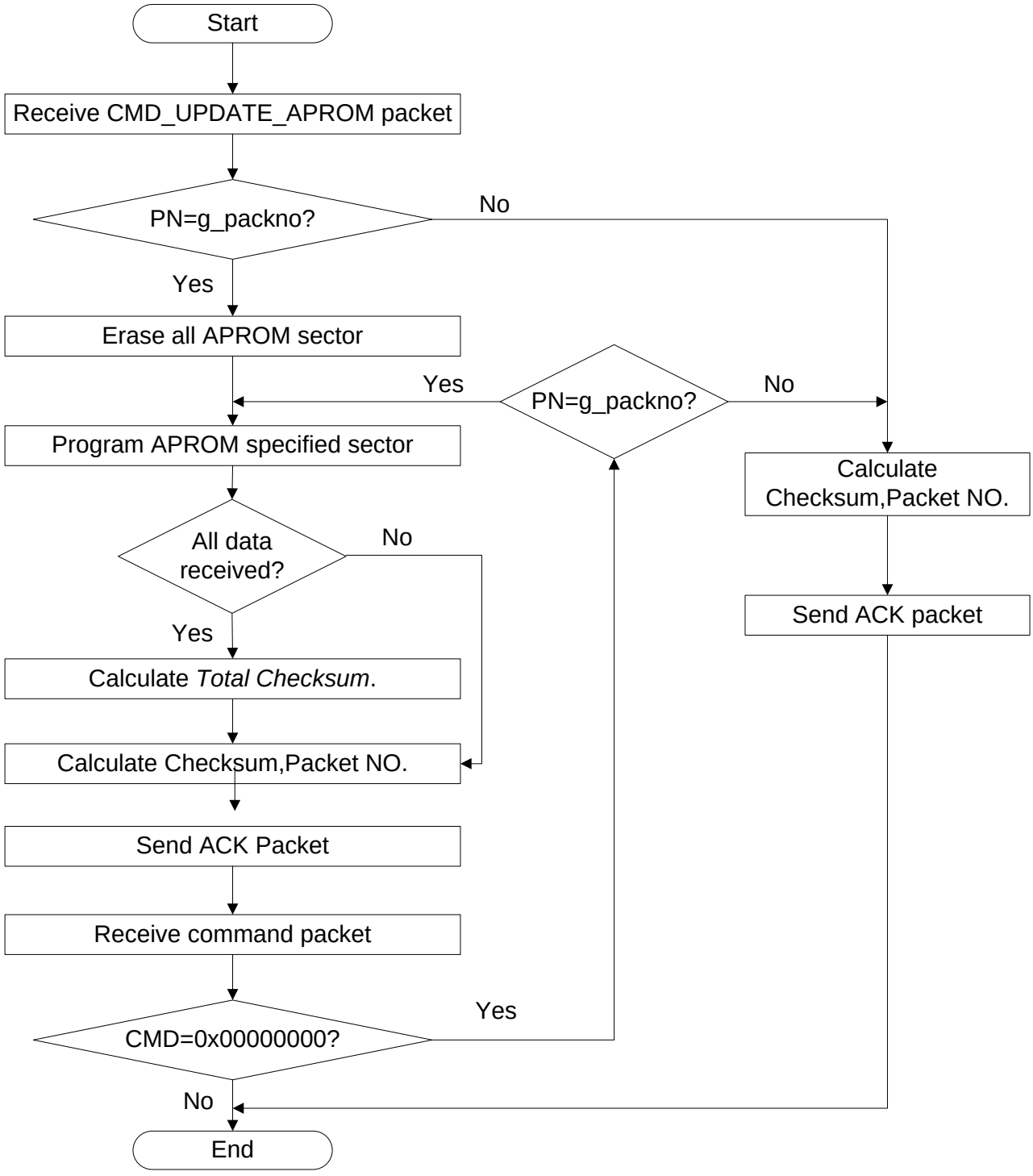

Figure 8.4 Program APROM Command: Device Side

8.2 Command CMD_UPDATE_CONFIG (0xA1)

Command Format

Host Send:

4	4	4	4
CMD	Packet Number	Config0	Config1

Device ACK:

4	4	4	4
Checksum	Packet Number	New Config0	New Config1

Description

This command is used to instruct ISP to update *Config0* and *Config1*. The command format and data transfer format is described as preceding section.

When ISP received the command packet, it erases all APROM flash excluding data flash and then reads the *Config0* and *Config1* data from the packet received and programs them into config area. Then it reads them out from config area and fills them in the ACK packet. After this, ISP calculates the checksum of the whole packet received, then transmits ACK packet to the host.

After ACK packet is received, host will check the checksum is right or wrong. If the checksum is right, it indicates that the data transfer is successful. After the operation is finished, then user can continue to run another different operation.

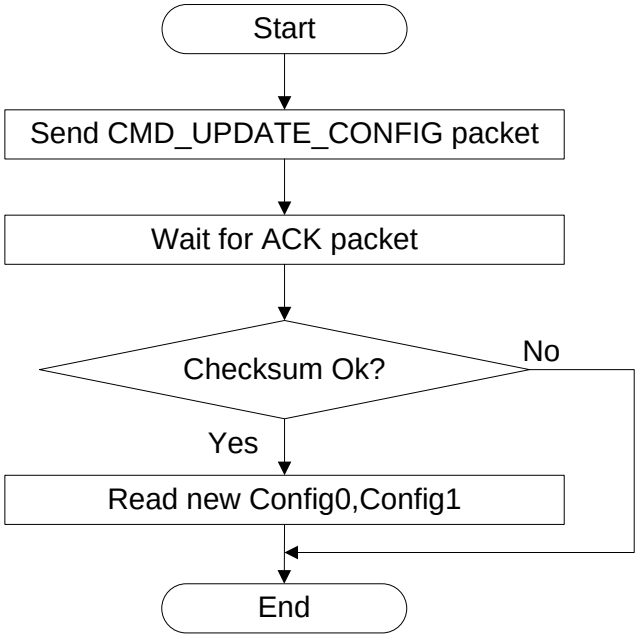

Figure 8.5 Update Config0 and Config1 Command: Host Side

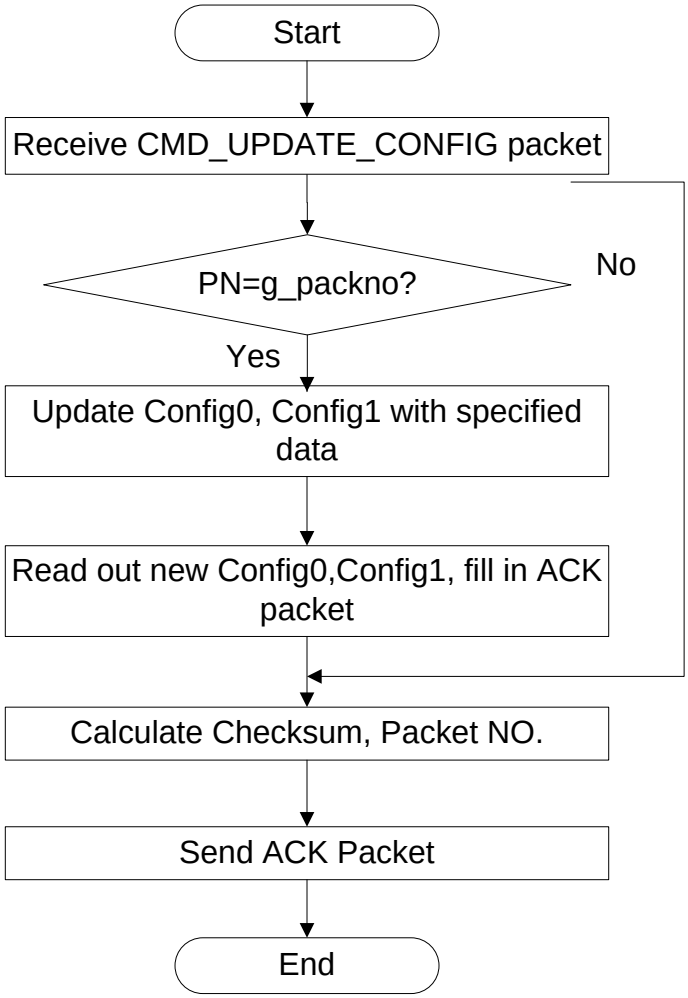

Figure 8.6 Update Config0 and Config1 Command: Device Side

8.3 Command CMD_READ_CONFIG (0xA2)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4	4	4
Checksum	Packet Number	Config0	Config1

Description

This command is used to instruct ISP to read *Config0* and *Config1* information of flash memory, and transmit them to host. The command format and data transfer format is described as preceding section.

When ISP received the command packet, it reads out the *Config0* and *Config1* from config area and fills them in the ACK packet. After this, ISP calculate the checksum of the whole packet received, then transmit ACK packet to the host.

After ACK packet is received, host will check the checksum is right or wrong. If the checksum is right, it indicates that the data transfer is successful, then user can read out the value of *Config0* and *Config1* from received packet. After this operation is finished, then user can send another command packet to start another operation.

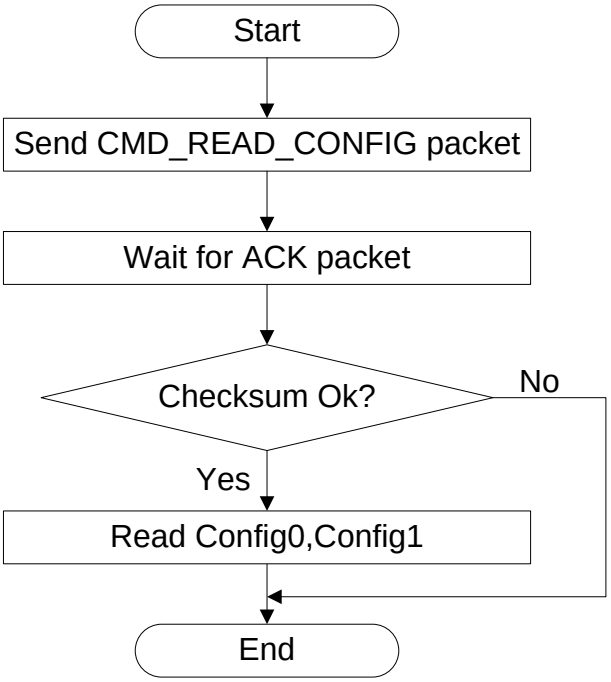

Figure 8.7 Read Config0 and Config1 Command: Host Side

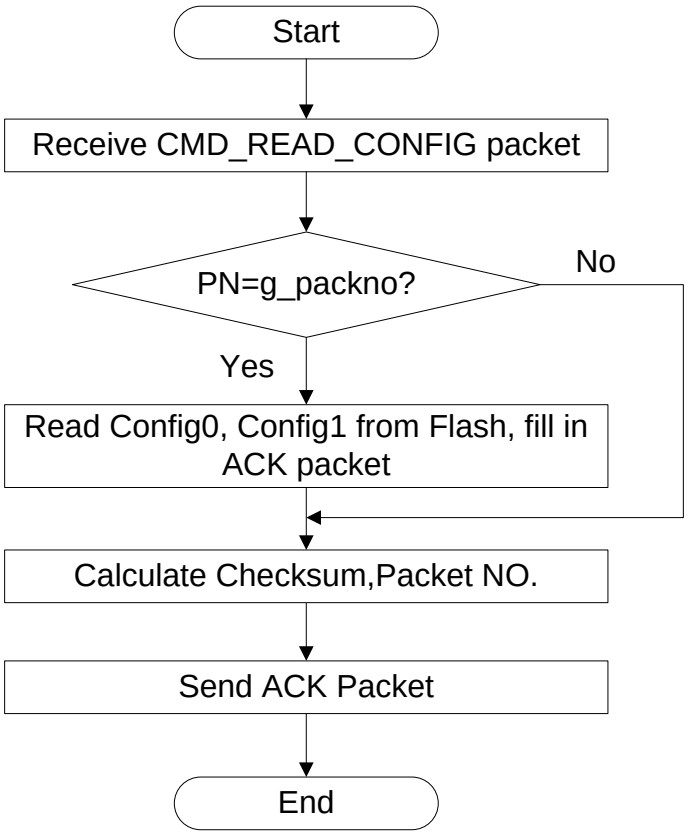

Figure 8.8 Read Config0 and Config1 Command: Device Side

8.4 Command CMD_ERASE_ALL (0xA3)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4
Checksum	Packet Number

Description

This command is used to instruct ISP to erase APROM, Data Flash, and set *Config0* and *Config1* registers to 0xFFFFFF7F and 0x0001F000. The command format and data transfer

format is described as preceding section.

When ISP received the command packet, it erases APROM, Data Flash, *Config0* and *Config1* registers. After this, ISP calculates the checksum of received data and packet number, then transmits ACK packet to the host.

After ACK packet is received, host will check the checksum is right or wrong. If the checksum is right, it indicates that the command has been implemented successfully. After this operation is finished, then user can send another command packet to start another operation.

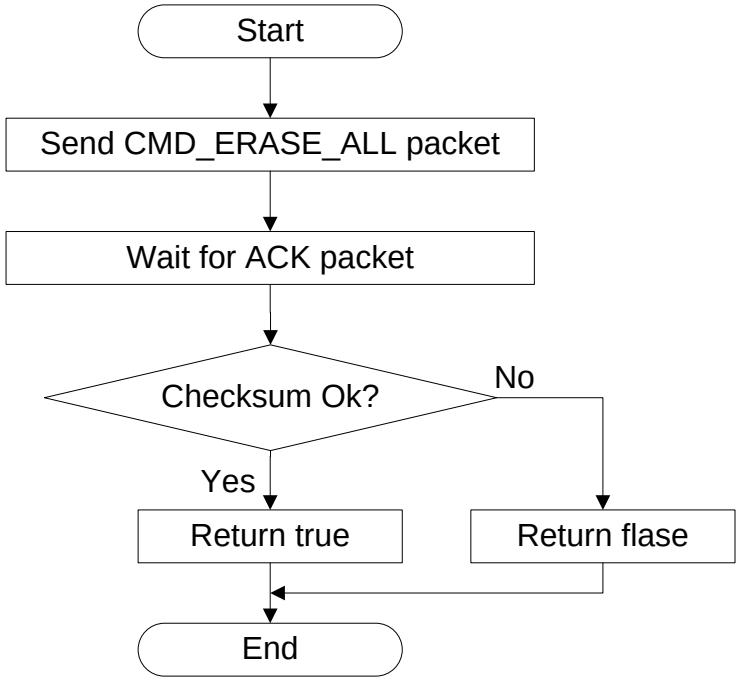

Figure 8.9 Erase Flash Memory Command: Host Side

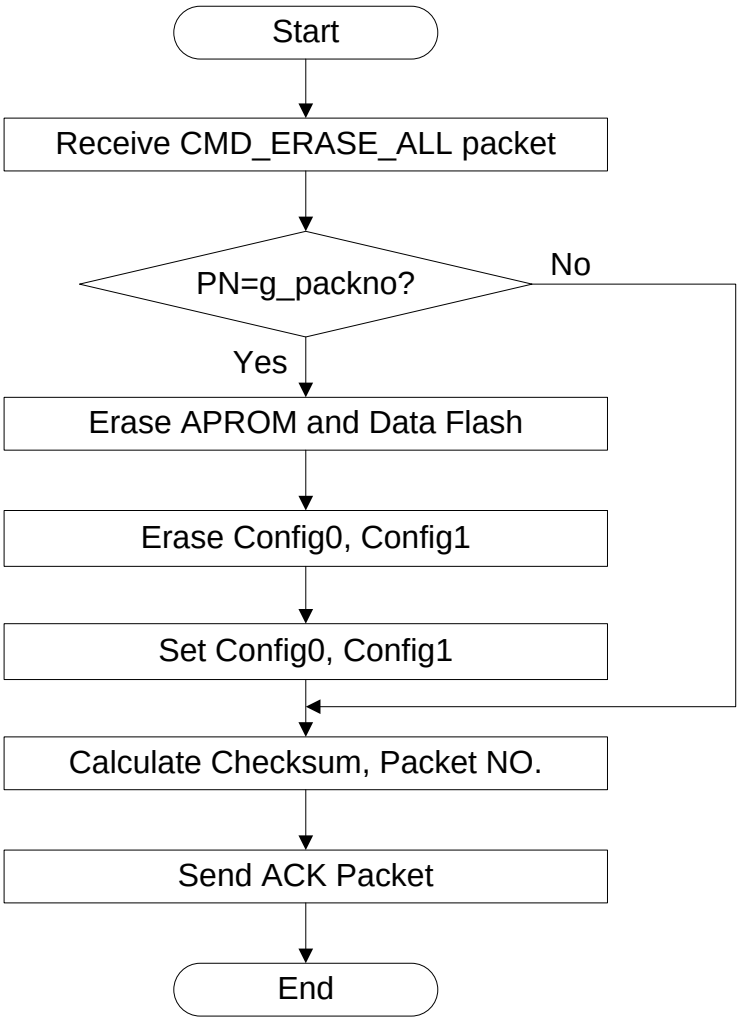

Figure 8.10 Erase Flash Memory Command: Device Side

8.5 Command CMD_SYNC_PACKNO (0xA4)

Command Format

Host Send:

4	4	4
CMD	PN	RN

PN is Packet number

RN is equal to PN

Device ACK:

4	4
Checksum	Packet Number

Description

This command is used to synchronize packet number with ISP. Before sending any command, master/host need send the command to synchronize packet number with ISP. The command format and data transfer format is described as preceding section.

When ISP received the command packet, it will check received PN and RN is equal or not, if equal, ISP calculates the checksum of the whole packet received and uses the packet number received increasing by one as the packet number of ACK packet. Then transmits the ACK packet to the host. If PN is not equal to RN, ISP will discard this package and send an ACK package with packet number is zero.

After ACK packet is received, host will checks the checksum is right or wrong. If the checksum is right, it indicates that the connection with ISP has established successfully' After this operation is finished, then user can send command packet to start corresponding operation. Otherwise, if checksum is wrong, it indicates that the connection with ISP is failed, and host will not send any other command packet.

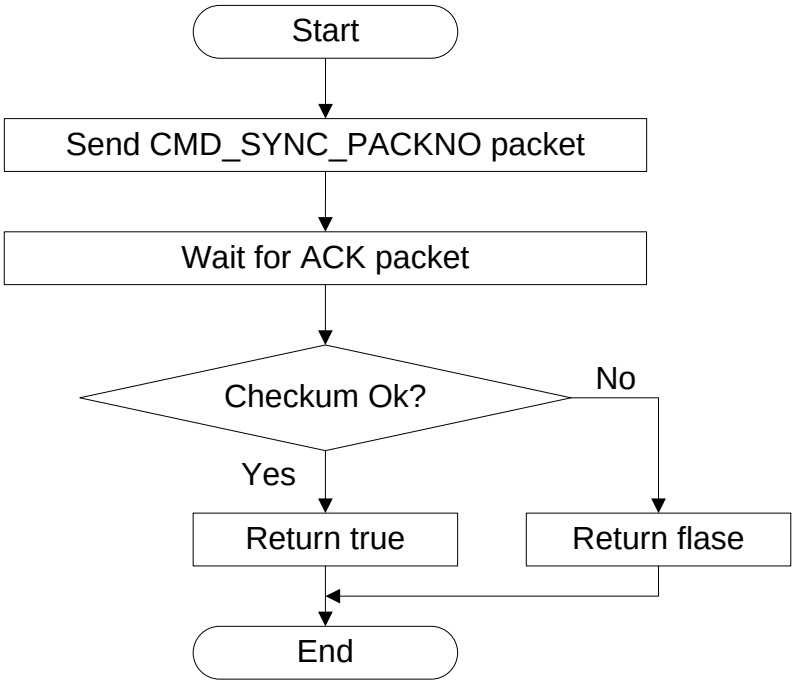

Figure 8.11 Sync Packet Number Command: Host Side

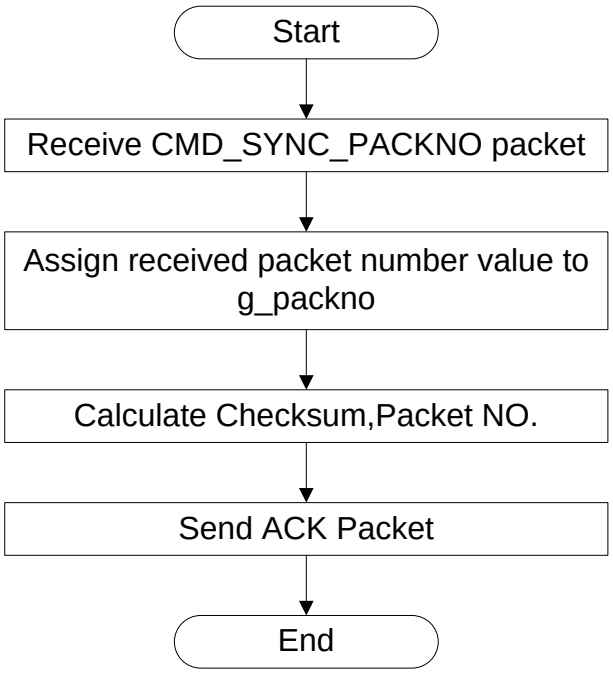

Figure 8.12 Sync Packet Number Command: Device Side

8.6 Command CMD_GET_FWVER (0xA6)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4	4
Checksum	Packet Number	version

Description

This command is used to get version of ISP. The command format and data transfer format is described as preceding section.

When ISP received the command packet, it gets version information and filling it in the ACK packet. After this, ISP calculates the checksum of the whole packet received, then transmits ACK packet to the host.

After ACK packet is received, host will check the checksum is right or wrong. If the checksum is right, it indicates that the data transfer is successful, then user can read out the version of

ISP from received packet. After this operation is finished, then user can send another command packet to start another operation.

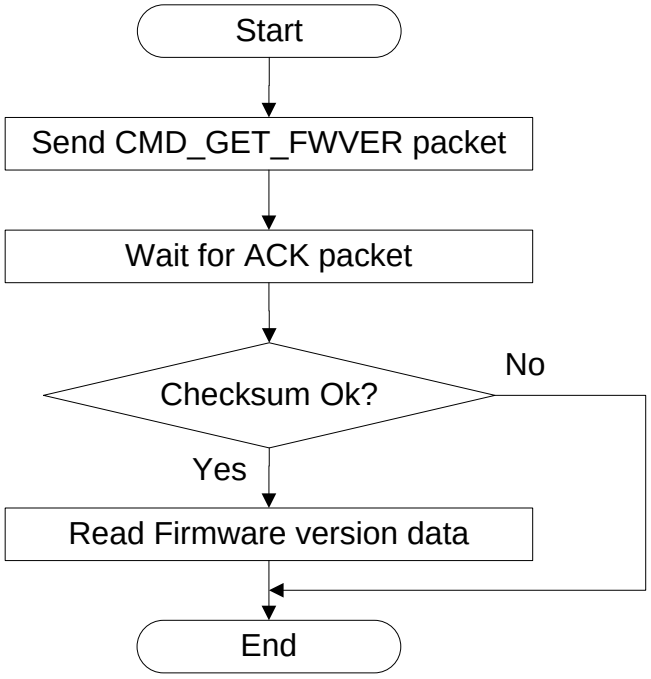

Figure 8.13 Get ISP Version Command: Host Side

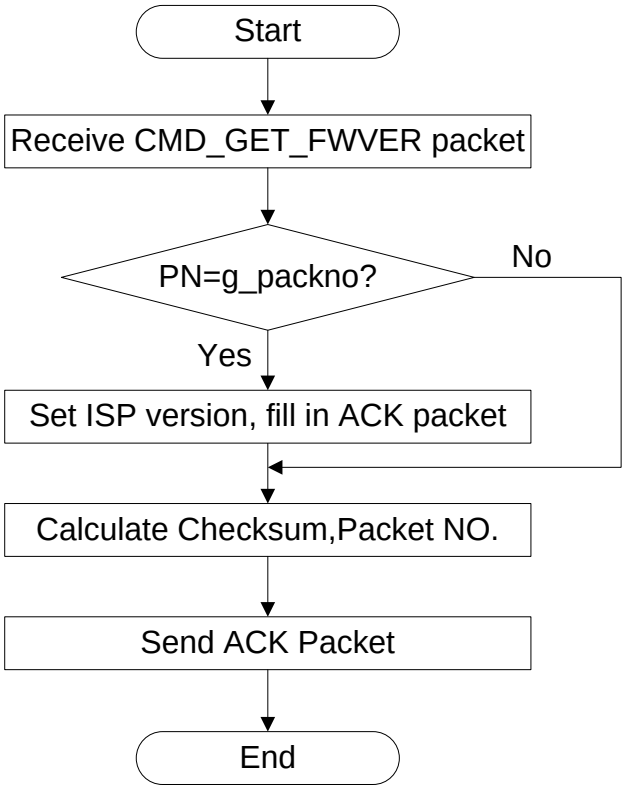

Figure 8.14 Get ISP Version Command: Device Side

8.7 Command CMD_GET_DEVICEID (0xB1)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4	4
Checksum	Packet Number	Device ID

Description

This command is used to get product ID. PC needs this ID to inquire size of APROM size and inform ISP. The command format and data transfer format is described as preceding section.

When ISP received the command packet, it gets product ID and fills it in the ACK packet. After this, ISP calculates the checksum of the whole packet received, then transmits ACK packet to the host.

After ACK packet is received, host will check the checksum is right or wrong. If the checksum is right, it indicates that the data transfer is successful, then user can read out the product ID from received packet. After this operation is finished, then user can send another command packet to start another operation.

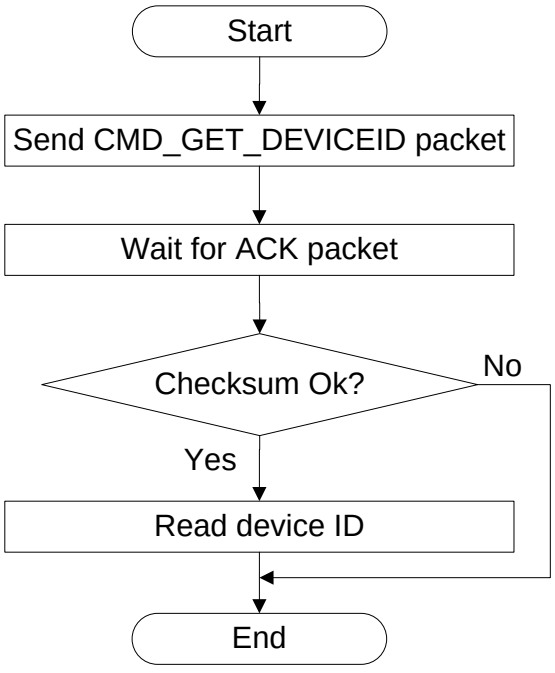

Figure 8.15 Get product ID Command: Host Side

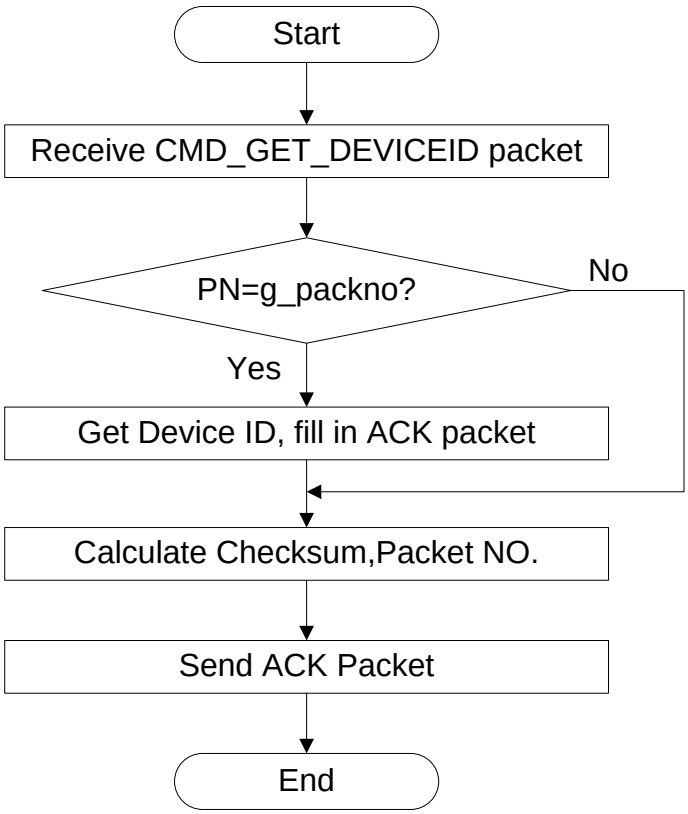

Figure 8.16 Get product ID Command: Device Side

8.8 Command CMD_UPDATE_DATAFLASH (0xC3)

Command Format

Host Send (named Format1):

4	4	4	4	
CMD	Packet Number	Start Address	Total Length	Data……

If data can't put into one packet, and then packet format is as following (named Format2)

0x00000000	Packet Number	Data……

Device ACK:

4	4	4
Checksum	Packet Number	Total Checksum

Description

This command is used to instruct ISP to program dataflash memory with data, firstly it ***erase the whole Data flash***, and then programs dataflash from the first beginning. The command format and data transfer format is described as preceding section; the parameter of S*tart Address* is ignored by ISP. If program data is too big to transfer in a packet, user can split data into several packets and transfer by multiple times.

When ISP received the command packet, it reads the program data from received packet, then programs data into the dataflash memory from the first beginning of dataflash. After program operation is completed, it calculates the checksum of the whole packet received, then transmits ACK packet to the host. If all data of *Total Length* are transmitted, the *Total Checksum* of data of *Total Length* will be return in offset 8 of ACK packet

After ACK packet is received, host will check the checksum is right or wrong. If the checksum is right, it indicates that the data transfer is successful. After this operation is finished, user can continue to run another different operation or transmit another data packet.

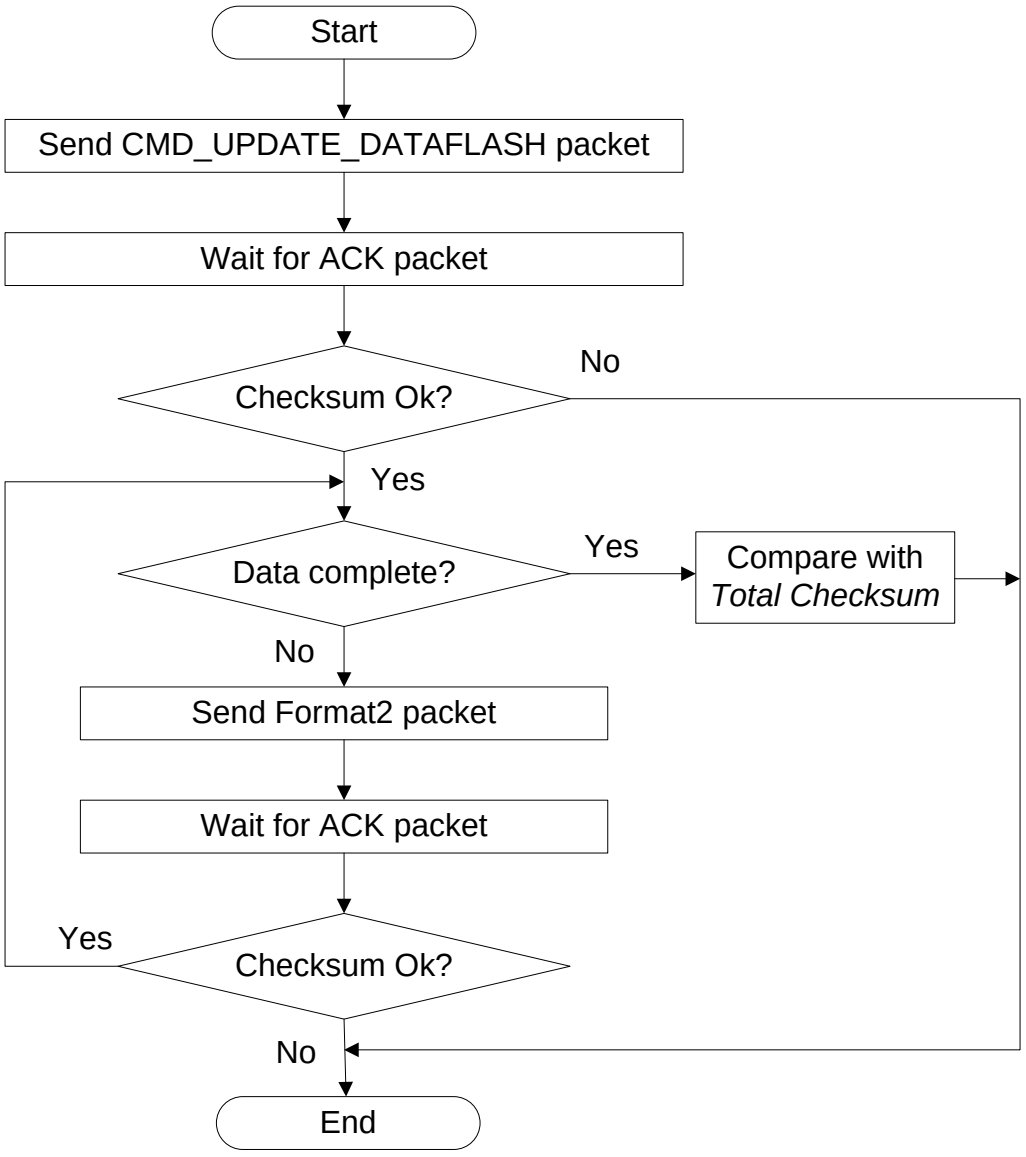

Figure 8.17 Program Data Flash Command (2): Host Side

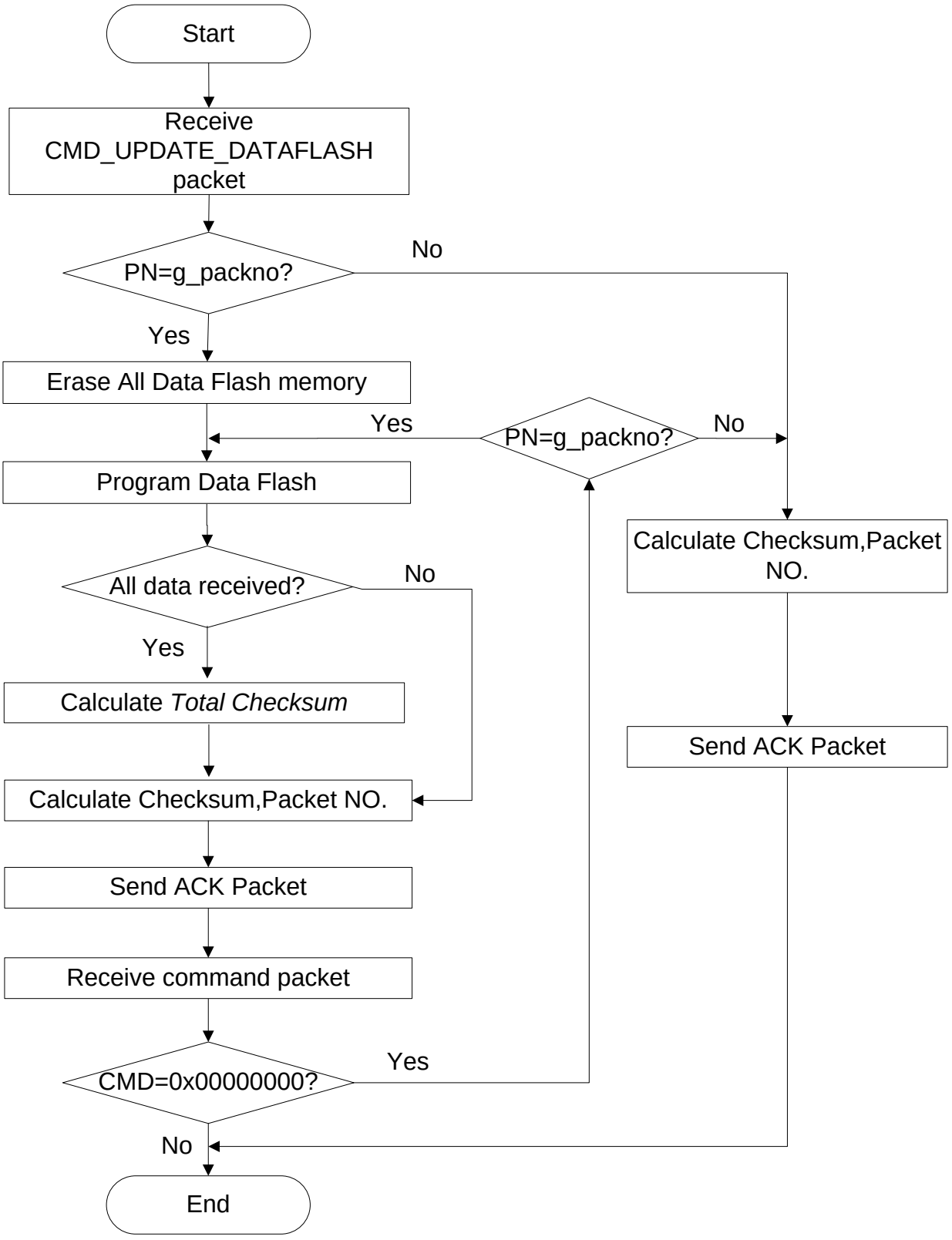

Figure 8.18 Program Data Flash Command (2): Device Side

8.9 Command CMD_RUN_APROM (0xAB)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

No ACK.

User should wait about 0.5s, the system will restart from APROM

Description

This command is used to boot from APROM. If user want to run application in APROM, you can issue this command to ISP.

When ISP received the command packet, it will modify BS bit to 0 in ISPCON register, and then issue SYSRESETREQ to AIRCR register of NuMicro® family. After this, the system will be rebooted from APROM.

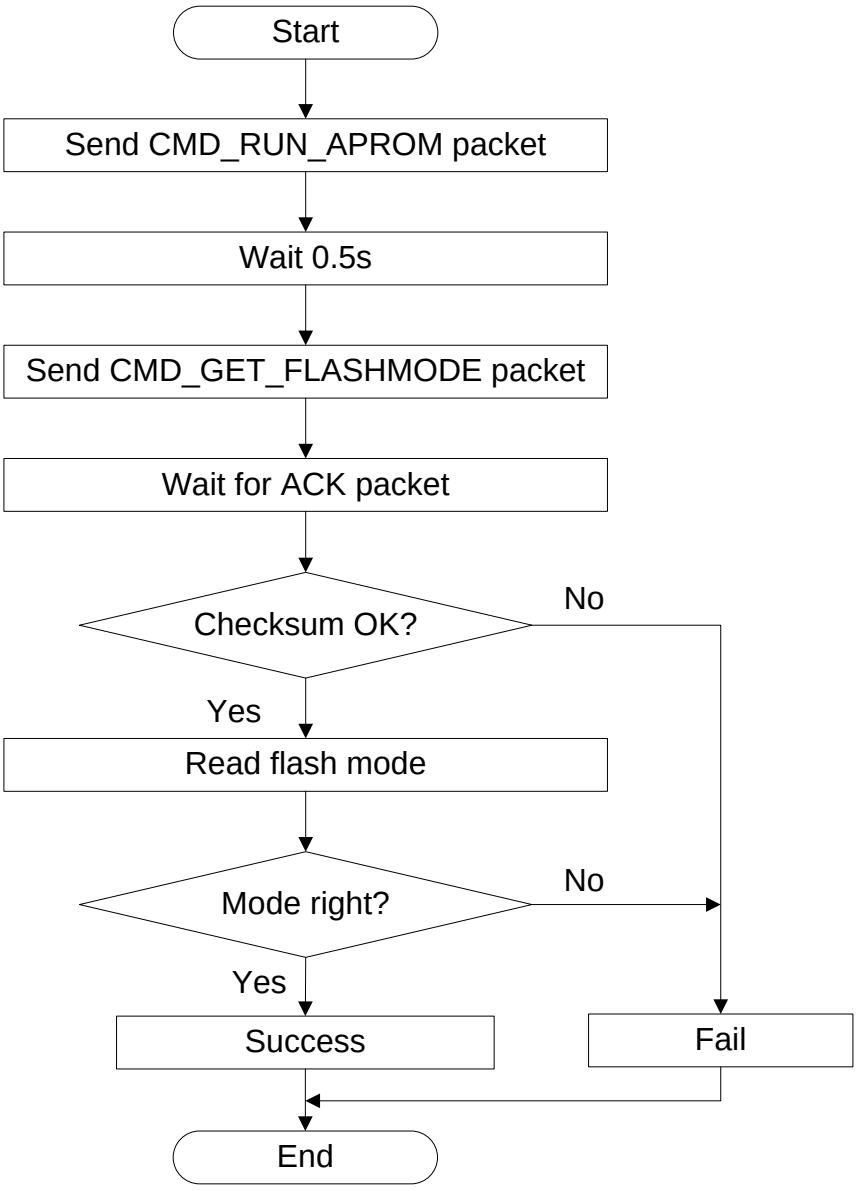

Figure 8.19 Run APROM Command: Host Side

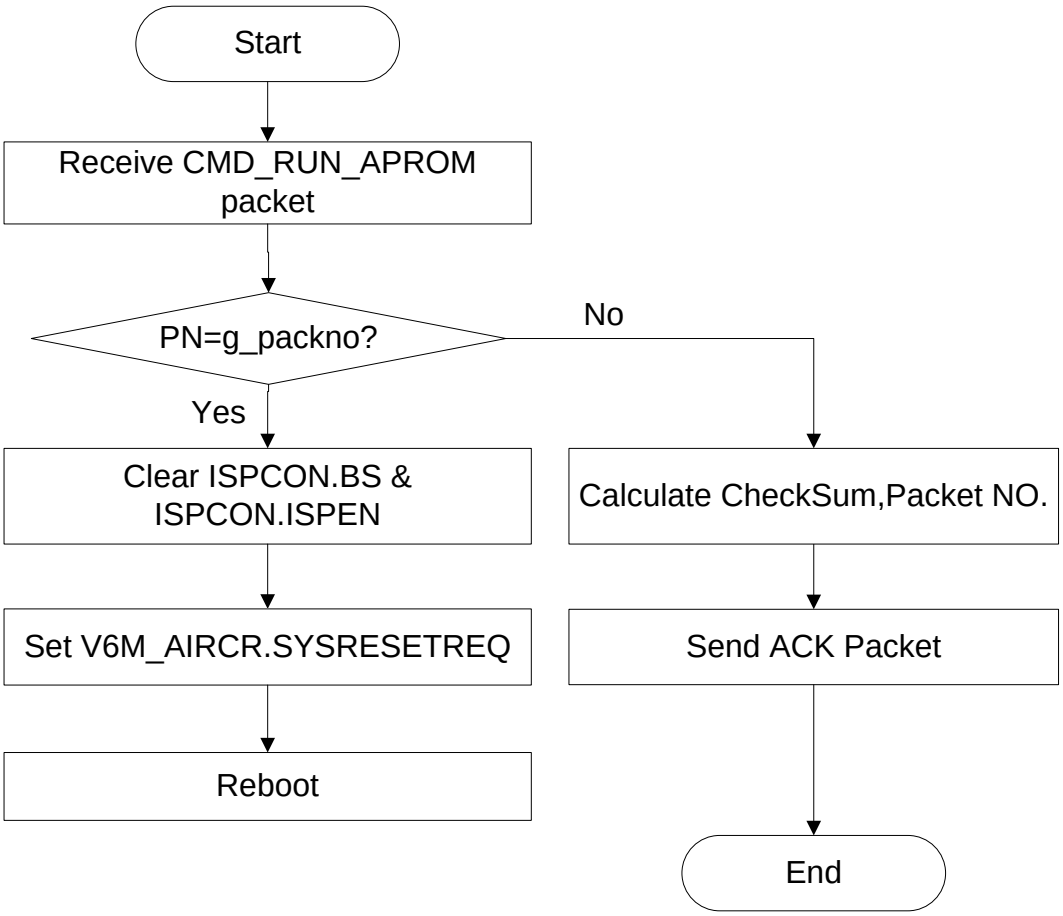

Figure 8.20 Run APROM Command: Device Side

8.10 Command CMD_RUN_LDROM (0xAC)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

No ACK.

User should wait about 0.5s, the system will restart from LDROM

Description

This command is used to boot from LDROM. If user want to reboot ISP, you can issue this command to ISP.

When ISP received the command packet, it will modify BS bit to 1 in ISPCON register, and then issue SYSRESETREQ to AIRCR register of NUC100 or M051 series. After this, the system will be rebooted from LROM.

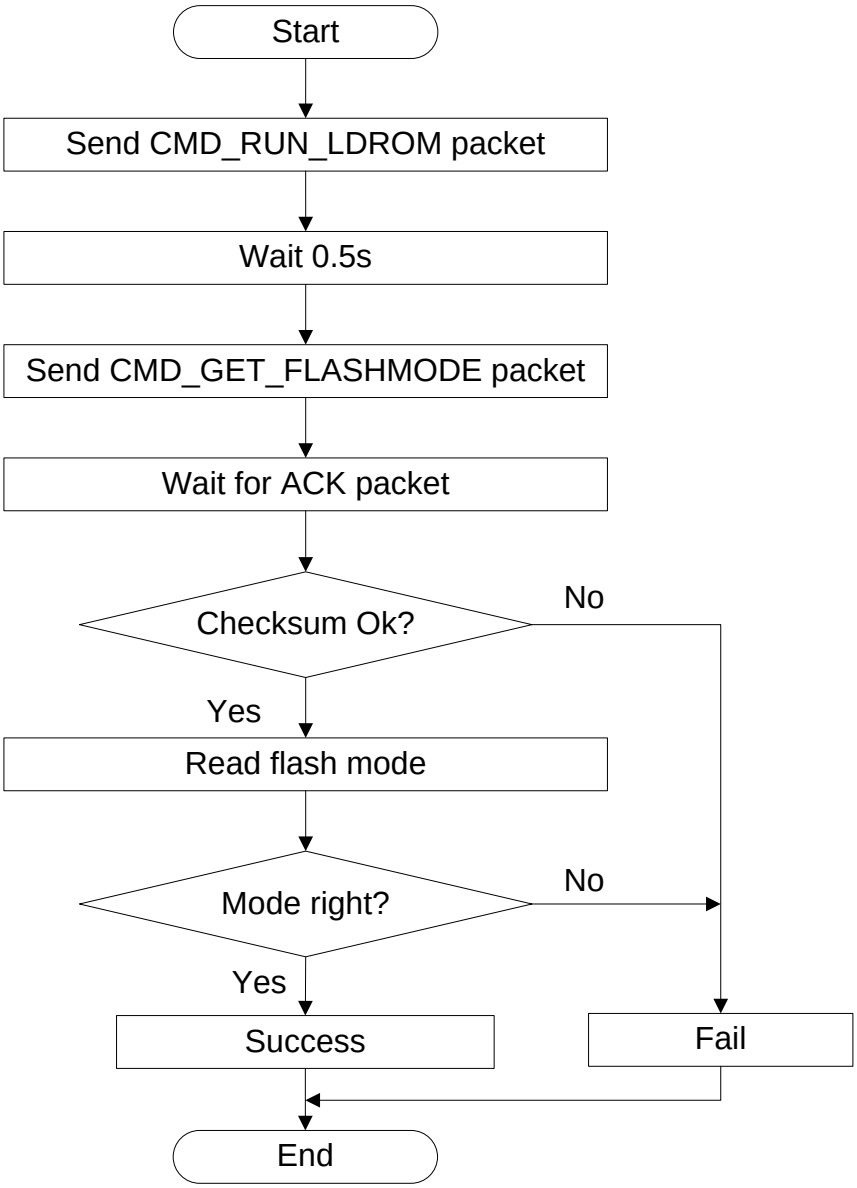

Figure 8.21 Run LDROM Command: Host Side

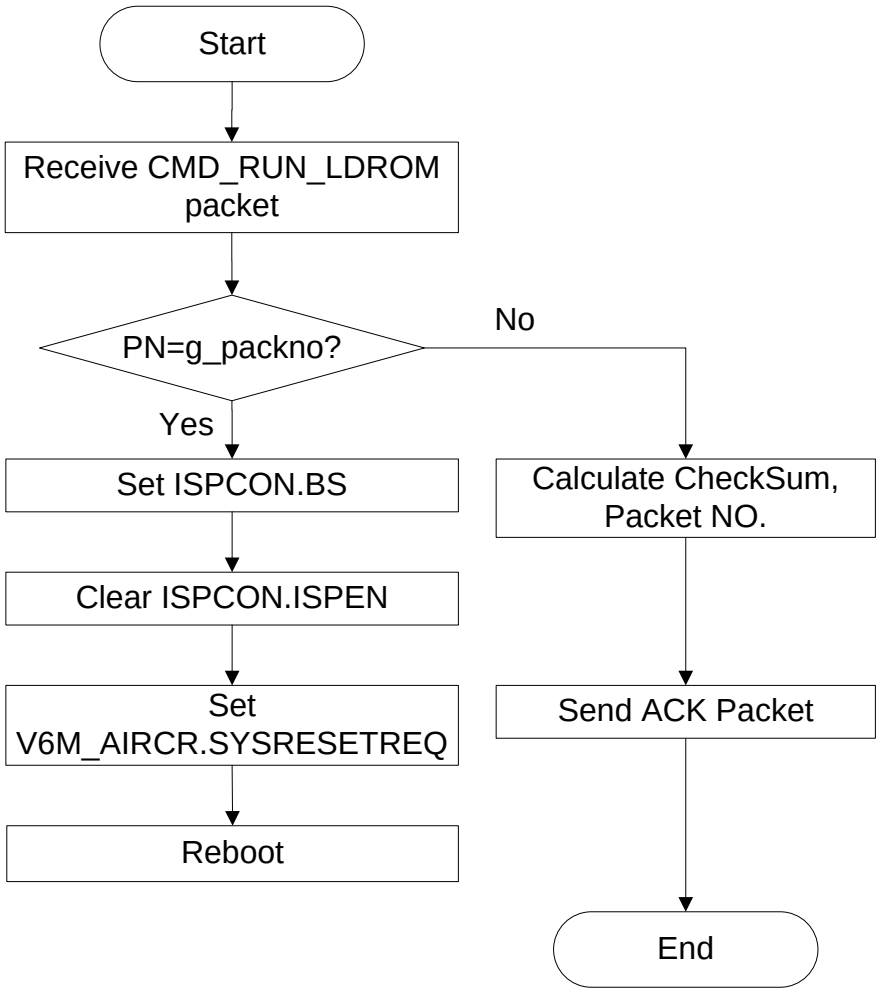

Figure 8.22 Run LDROM Command: Device Side

8.11 Command CMD_RESET (0xAD)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

No ACK.

User should wait about 0.5s, the system will restart

Description

This command is used to reboot system. If user want to reboot ISP, you can issue this command to ISP.

When ISP received the command packet, it will issue SYSRESETREQ to AIRCR register. After this, the system will reboot.

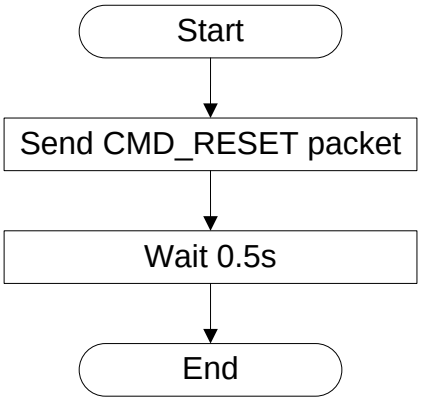

Figure 8.23 Reset Command: Host Side

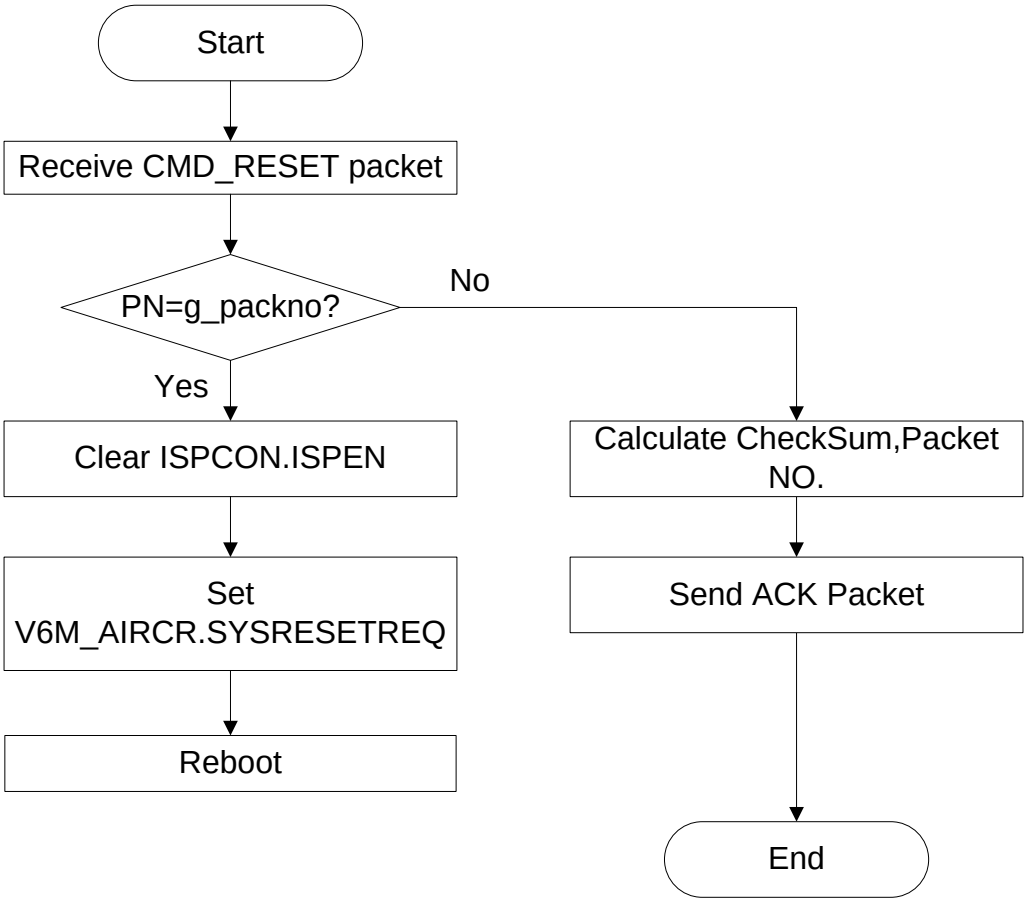

Figure 8.24 Reset Command: Device Side

8.12 Command CMD_WRITE_CHECKSUM (0xC9)

Command Format

Host Send:

4	4	4	4
CMD	Packet Number	TotalLen	Checksum

Device ACK:

4	4
Checksum	Packet Number

Description

This command is used to instruct ISP to write the program length of application and checksum into the last 8 bytes of APROM. After update APROM finished by ISP, user can issue this command to fill total length and checksum of application to APROM. ISP will write the total length and checksum to the last 8 bytes of APROM.

When ISP received the command packet, it searches APROM size and then write the total length and checksum to the last 8 bytes of APROM and fills them in the ACK packet, then transmits ACK packet to the host.

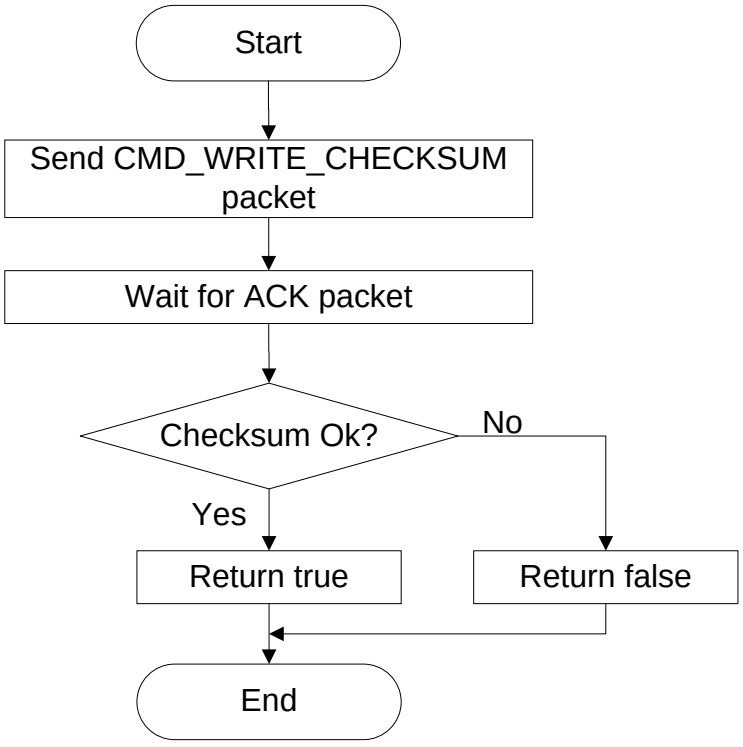

Figure 8.25 Write Checksum Command: Host Side

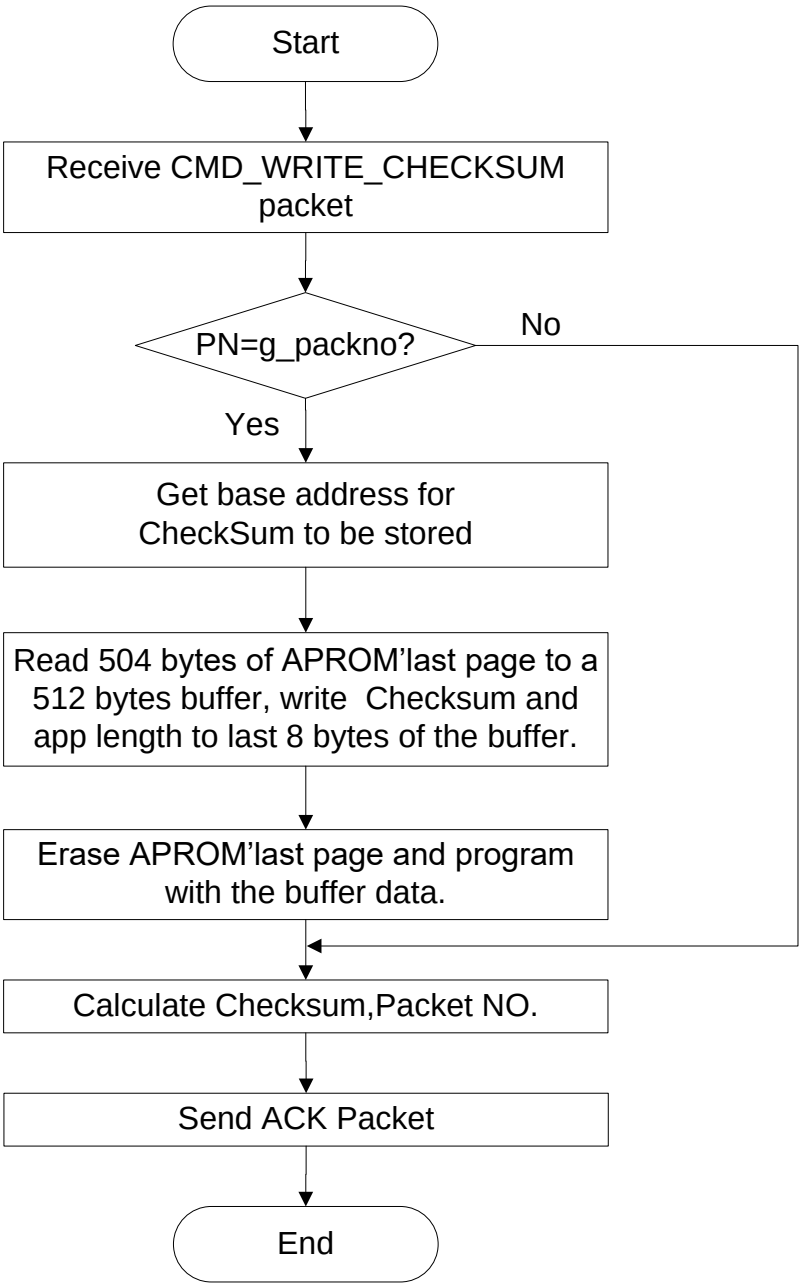

Figure 8.26 Write Checksum Command: Device Side

8.13 Command CMD_GET_FLASHMODE (0xCA)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4	4
Checksum	Packet Number	Mode

Mode = 1, Run in APROM

Mode = 2, Run in LDROM

Description

This command is used to get boot selection (BS) bit. If boot selection is APROM, the mode of returned is equal to 1. Otherwise, if boot selection is LDROM, the mode of returned is equal to 2. If boot selection is APROM, user should issue CMD_RUN_LDROM to reboot from LDROM. The command format and data transfer format is described as preceding section.

When ISP received the command packet, it reads ISPCON register to get BS bit, and fills it in the ACK packet base on BS bit, then transmits ACK packet to the host.

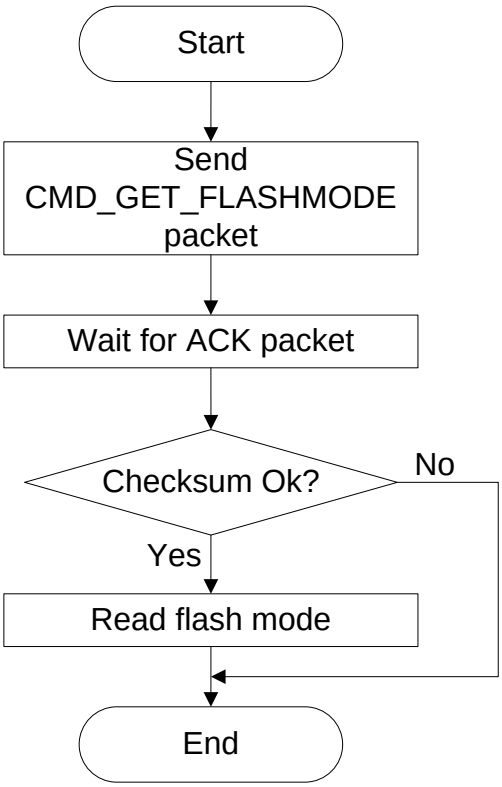

Figure 8.27 Get Flash Mode Command: Host Side

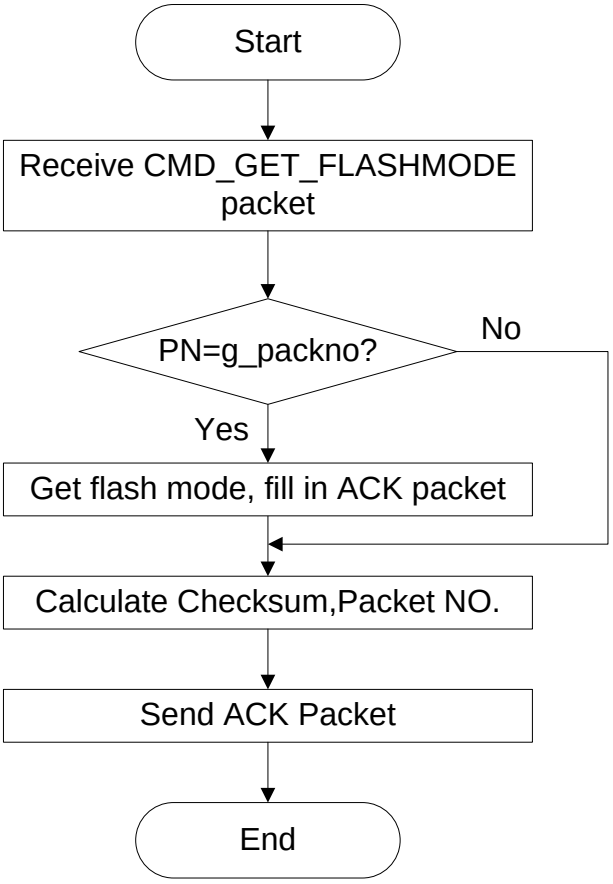

Figure 8.28 Get Flash Mode Command: Device Side

8.14 Command CMD_RESEND_PACKET (0xFF)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4
Checksum	Packet Number

Description

This command is used to inform ISP that the next packet is a duplicate packet of the previous packet. The command is used to resend a packet when error happen. For example: if error happen when host(PC) send or receive a packet, host can send CMD_RESEND_PACKET command to inform ISP, the next packet is a duplicate packet of the error packet, after the command CMD_RESEND_PACKET, host will resend the error packet

When ISP received the command packet, it will erase data that is written in last error packet.

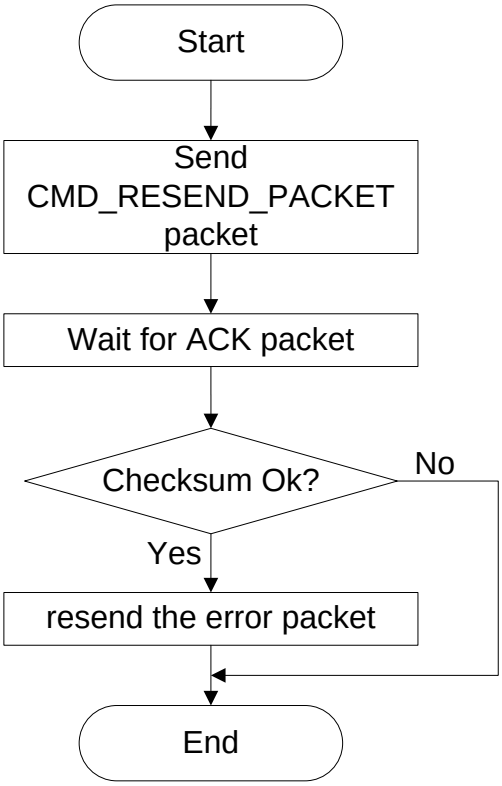

Figure 8.29 Resend Packet Command: Host Side

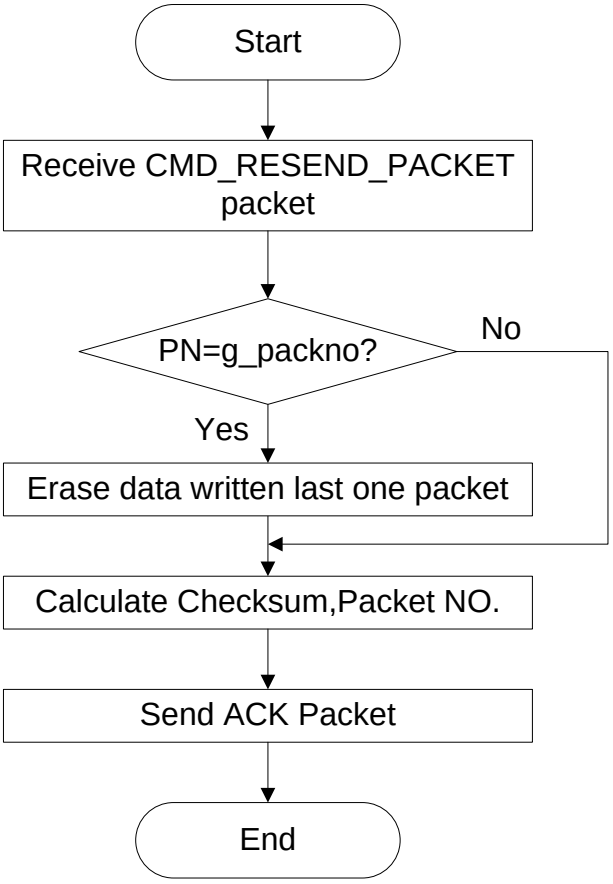

Figure 8.30 Resend Packet Command: Device Side

8.15 Command CMD_CONNECT (0xAE)

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4
Checksum	Packet Number

Description

The command is used to check whether or not ISP is running. If ISP is running, ISP will response 64 bytes ack packet. Eg: user can use it to support auto detect function, Host may send the command repeatedly until ISP response

When ISP received the command packet, it will restore g_packno to 1 and return ack to host.

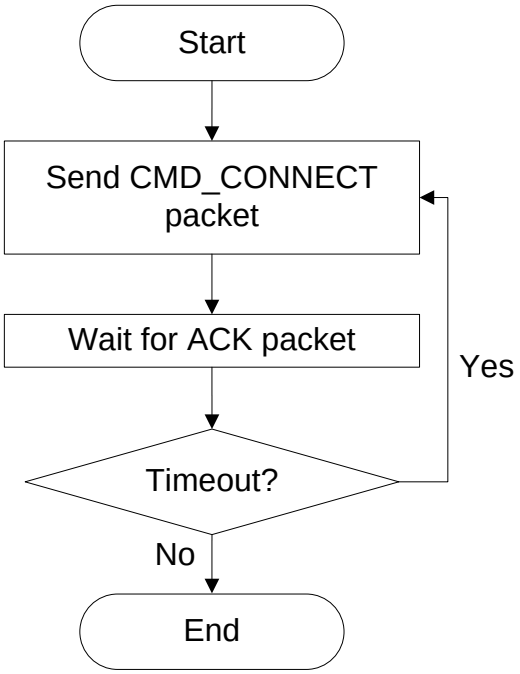

Figure 8.31 Connect Command: Host Side

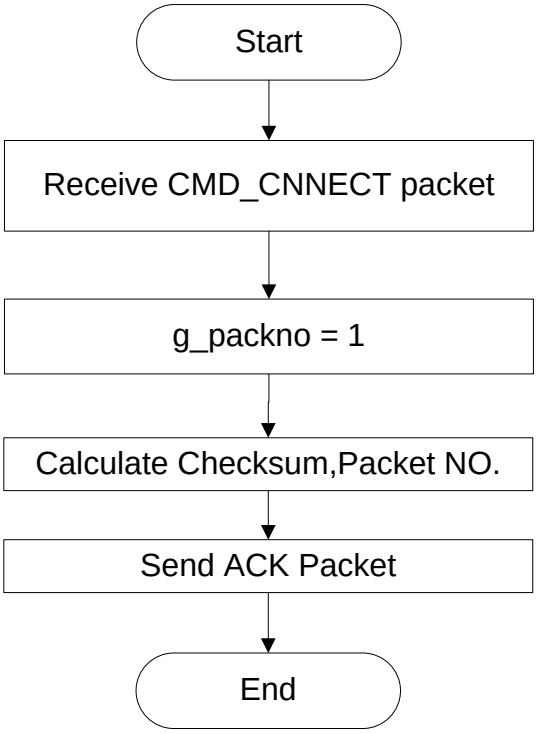

Figure 8.32 Connect Command: Device Side

9 Commands flow

If user need update APROM, the following commands need issue.

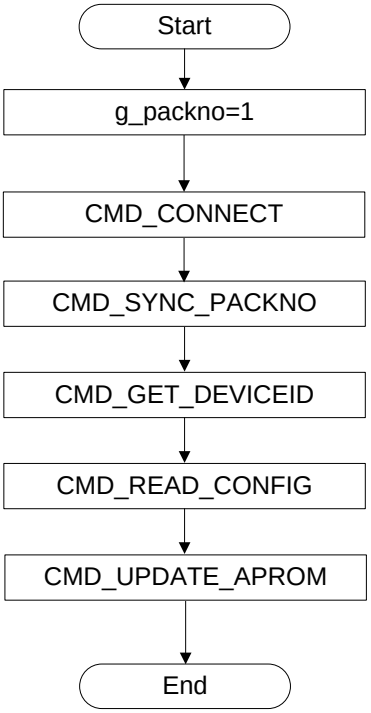

Figure 9.1 Update APROM command flow

Revision History

Date	Revision	Description
2010.08.13	1.0	Initially issued.
2010.09.07	1.2	Modify all flow and overview
2010.10.08	1.3	Add I^2C bus signal and SPI/I^2C depict
2011.03.03	1.4	Remove some command and Modify CMD_UPDATE_APROM flow
2012.05.16	1.5	Add Pin definition for each interface
2023.01.30	2.0	Add new interface introduction

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.