NuMicro™ ISP Programming Guide

Introduction:

Most of modern consumer products has the capability to upgrade its firmware code running on internal microcontroller. With this feature, the product is able to continuously support new functions after it was made and released to end customer. Nuvoton provides a ISP (In-System Programming) method to update the flash code of the NuMicro™ NUC100 or NuMicro M051™ series Flash-memory-based microcontrollers.

The NuMicro™ ISP code is resident in LDROM (Loader ROM), it supports different I/O interfaces, including USB, UART, I2C, or SPI, to program or update the application code into internal APROM (Application ROM). This is a very convenient way for developer or end user to update application code of a NuMicro™ chip that was mounted on PCB (Printed Circuit Board). .

This application note describes the ISP code architecture, I/O interfaces and the usage of ISP commands.

The relative datasheet, Technical Reference Manual, BSP (Board Support Package) software, tools, and supporting information can be downloaded from Nuvoton Website - http://www.nuvoton.com/NuMicro

Table of Contents:

NuMicro™ ISP Programming Guide 1
1. Overview 5
1.1 Receive/Send data 7
2. Pin Definition 8
3. ISP Through USB and UART 9
3.1 Project and Source Files 9
3.2 Communication Interface 10
3.3 The ISP Program 10
3.3.1 ISP Booting Flow 10
3.3.2 ISP Main Flow 12
4. ISP Through SPI 14
4.1 Project and Source Files 14
4.2 Communication Interface 16
4.3 The SPI ISP Program 16
4.3.1 Data transfer on the SPI bus 16
4.3.2 Commands response delay 16
4.3.3 ISP Booting Flow 17
4.3.4 ISP Main Flow 18
4.4 The Example Application and Master ISP Program 20
5. ISP Through I2C 23
5.1 Project and Source Files 23
5.2 Communication Interface 25
5.3 The I2C ISP Program 26
5.3.1 Data transfer on the I2C bus 26
5.3.2 Commands response delay 26
5.3.3 ISP Booting Flow 27
5.3.4 ISP Main Flow 28
5.4 The Example Application and Master ISP Program 30
6. ISP Command Set 31
6.1 Command CMD_UPDATE_APROM(0xA0): 32
6.2 Command CMD_UPDATE_CONFIG (0xA1): 34
6.3 Command CMD_READ_CONFIG (0xA2): 37
6.4 Command CMD_ERASE_ALL (0xA3): 39
6.5 Command CMD_SYNC_PACKNO (0xA4): 42
6.6 Command CMD_GET_FWVER (0xA6): 44
6.7 Command CMD_GET_DEVICEID (0xB1): 46
6.8 Command CMD_UPDATE_DATAFLASH (0xC3): 48
6.9 Command CMD_RUN_APROM(0xAB): 52
6.10 Command CMD_RUN_LDROM (0xAC): 54
6.11 Command CMD_RESET(0xAD): 56
6.12 Command CMD_WRITE_CHECKSUM(0xC9): 58
6.13 Command CMD_GET_FLASHMODE(0xCA): 60
6.14 Command CMD_RESEND_PACKET(0xFF): 63
6.15 Command CMD_CONNECT(0xAE): 65
7. Commands flow 68
8. Revision History 69

List of Figures:

Figure 1. ISP Through USB and UART 6
Figure 2. ISP Through I2C and SPI 7
Figure 3. Packet sent 7
Figure 4. Packet Received 7
Figure 5. Working Window of ISP Project 9
Figure 6. Working Window of USBUartAprom Project 10
Figure 7. USBUart ISP Booting Flow 11
Figure 8. USBUart ISP Main Flow 12
Figure 9. Working Window of SPIMaster Project 14
Figure 10. Working Window of ISP Project 15
Figure 11. Working Window of SPIAprom Project 15
Figure 12. SPI Connect Block Diagram (Redraw) 16
Figure 13. SPI ISP Booting Flow 18
Figure 14. SPI ISP Main Flow 19
Figure 15. SPI/I2C Master Main Flow 22
Figure 16. Working Window of I2CMaster Project 23
Figure 17. Working Window of ISP Project 24
Figure 18. Working Window of I2CAprom Project 25
Figure 19. I2C Connection Block Diagram 25
Figure 20. Write data to Slave device by Master device 26
Figure 21. Read data from Slave device by Master device 26
Figure 22. I2C ISP Booting Flow 28
Figure 23. I2C ISP Main Flow 29
Figure 24. Program APROM Command: Host Side 33
Figure 25. Program APROM Command: Device Side 34
Figure 26. Update *Config0* and *Config1* Command: Host Side 36
Figure 27. Update *Config0* and *Config1* Command: Device Side 37
Figure 28. Read *Config0* and *Config1* Command: Host Side 38
Figure 29. Read *Config0* and *Config1* Command: Device Side 39
Figure 30. Erase Flash Memory Command: Host Side 40
Figure 31. Erase Flash Memory Command: Device Side 41
Figure 32. Sync Packet Number Command: Host Side 43
Figure 33. Sync Packet Number Command: Device Side 43
Figure 34. Get ISP Version Command: Host Side 45
Figure 35. Get ISP Version Command: Device Side 46
Figure 36. Get product ID Command: Host Side 47
Figure 37. Get product ID Command: Device Side 48
Figure 38. Program Data Flash Command (2): Host Side 50
Figure 39. Program Data Flash Command (2): Device Side 51
Figure 40. Run APROM Command: Host Side 53
Figure 41. Run APROM Command: Device Side 54
Figure 42. Run LDROM Command: Host Side 56
Figure 43. Run LDROM Command: Device Side 56
Figure 44. Reset Command: Host Side 57
Figure 45. Reset Command: Device Side 58
Figure 46. Write Checksum Command: Host Side 59
Figure 47. Write Checksum Command: Device Side 60
Figure 48. Get Flash Mode Command: Host Side 62

Figure 49. Get Flash Mode Command: Device Side 62
Figure 50. Resend Packet Command: Host Side 64
Figure 51. Resend Packet Command: Device Side 65
Figure 52. Resend Packet Command: Host Side 66
Figure 53. Resend Packet Command: Device Side 67
Figure 54. update APROM command flow 68

1. Overview

The internal flash memory 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 use 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 download interfaces. For UART interface, ISP supports autodetect mode only, ISP will be started if NuMicro ISP Programming Tool.exe is running when boot from LDROM. ISP defines the GPIO pin (GPB15 for NUC100, GPD0 for NUC101) input LOW and *SYSRESETREQ (defined at AIRCR register) and checksum error(if selected)* events as the condition to enter ISP for the other interfaces, 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 LRROM"*, 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, I2C, SPI and UART as input/output interfaces. 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. The USB and UART interfaces are friendly to communicate with a Host PC. A windows-based application program, "*NuMicro ISP.exe"* is provided to demonstrate ISP function through USB and UART interfaces.

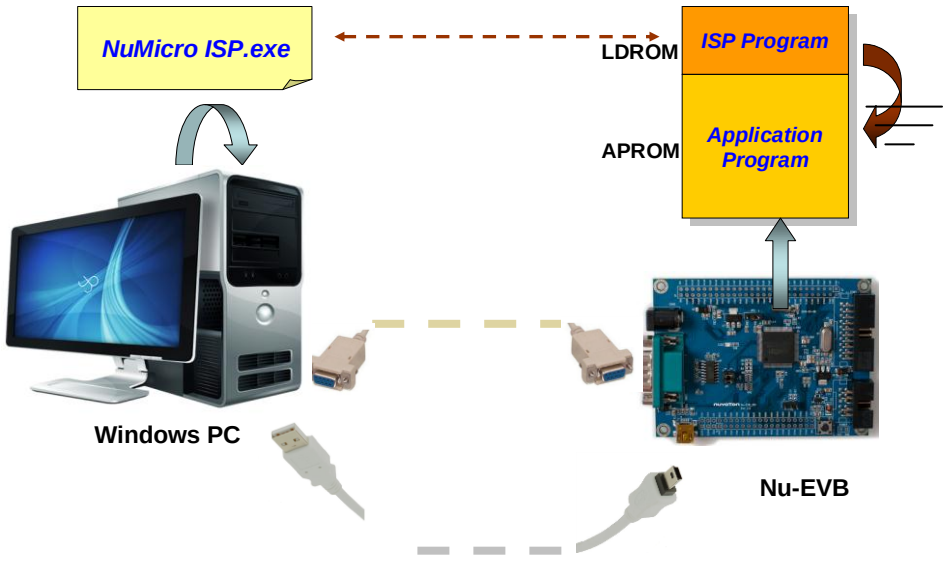


Figure 1. ISP Through USB and UART

The I2C and SPI are standard I/O interfaces of microcontroller. They are suitable to be used as a communication channel between NuMicro™ MCU and other microcontroller. Nuvoton also provides a master example code running on a NuMicro™ MCU to evaluate ISP functions through I2C and SPI interfaces. The following diagram illustrates the system environment of running ISP program.

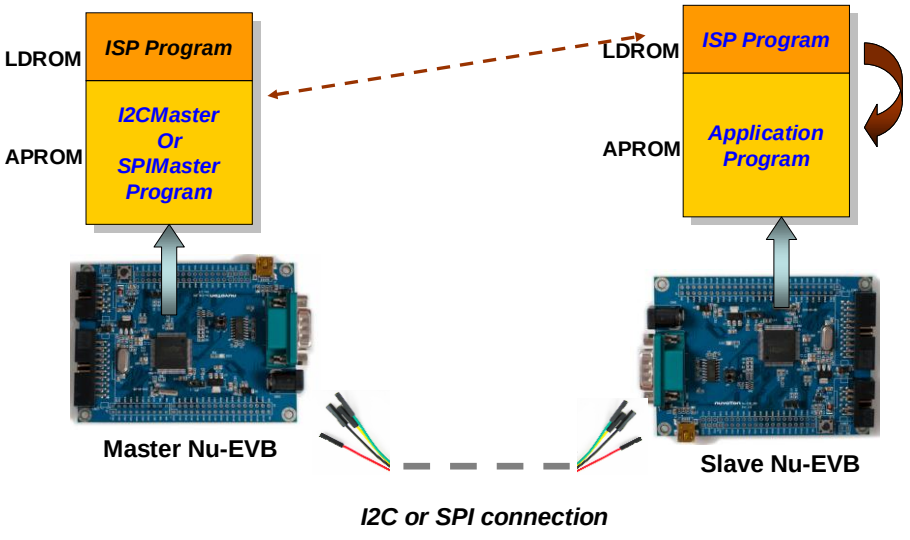


Figure 2. ISP Through I2C and SPI

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.

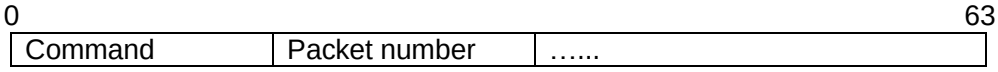


Figure 3. Packet sent

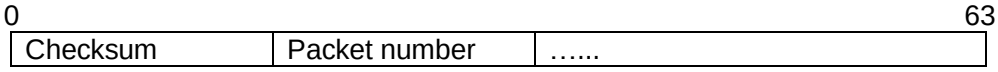


Figure 4. Packet Received

2. Pin Definition

Pins used as following for each interface

Chip	UART	SPI	I2C	USB
NUC100/NUC120/NUC130/NUC140	UART0 PB.0 = RX0 PB.1 = TX0	SPI0 PC.0=SS00, PC.1=CLK, PC.2=MISO0, PC.3=MOSI0	I2C0 PA.8=SDA PA.9=SCL	GPB15 low
NUC122	UART1 PB.4 = RX0 PB.5 = TX0			GPB14 low
M051 series	UART0 P3.0 = RXD P3.1 = TXD	SPI0 P1.4=SS0 P1.7=CLK P1.6=MISO0 P1.5=MOSI0	I2C0 P3.4 SDA P3.5 SCL	
MINI51 series	UART0 P0.1 = RX0 P0.0 = TX0	SPI0 P1.4=SS0 P1.7=CLK P1.6=MISO0 P1.5=MOSI0	I2C0 P3.4 SDA P3.5 SCL	
NANO series	UART0 PA14 = RX0 PA15 = TX0	SPI0 GPE2=CLK GPE1=SS0 GPC2=MISO GPC3=MOSI	I2C0 GPA12=SDA GPA13=SCK	GPB15 low

3. ISP Through USB and UART

A NuMicro™ MCU plays the slave device that a ISP and an application program are saved in LDROM and APROM respectively. The command and data are transmitted via USB or UART0 interface.

On the other side, a windows PC plays the master device. A PC ISP program, *NuMicro ISP Programming Tool.exe,* is used to send command and data to slave device. The *NuMicro ISP Programming Tool.exe* can download a new application code (in binary/hex format) to update the APROM. Please make sure a USB or RS232 cable is connected and GPB15(GPB15 for NUC100, GPD0 for NUC101) is low for USB interface, or NuMicro ISP Programming Tool.exe is running for UART interface before test ISP function.

3.1 Project and Source Files

The Keil uVision 4 IDE tool is used to compile the programs running on the slave side. The source files of ISP code can be found at directory "*NuvotonPlatform_Keil\Slave\UsbUART*", and a project file *UsbUARTISP.uvproj* is saved there. The demo code of application program and project file *Aprom.uvmpw* are saved at directory "*NuvotonPlatform_Keil\Slave\Aprom*". The following diagrams show the screen snapshot after open the projects.

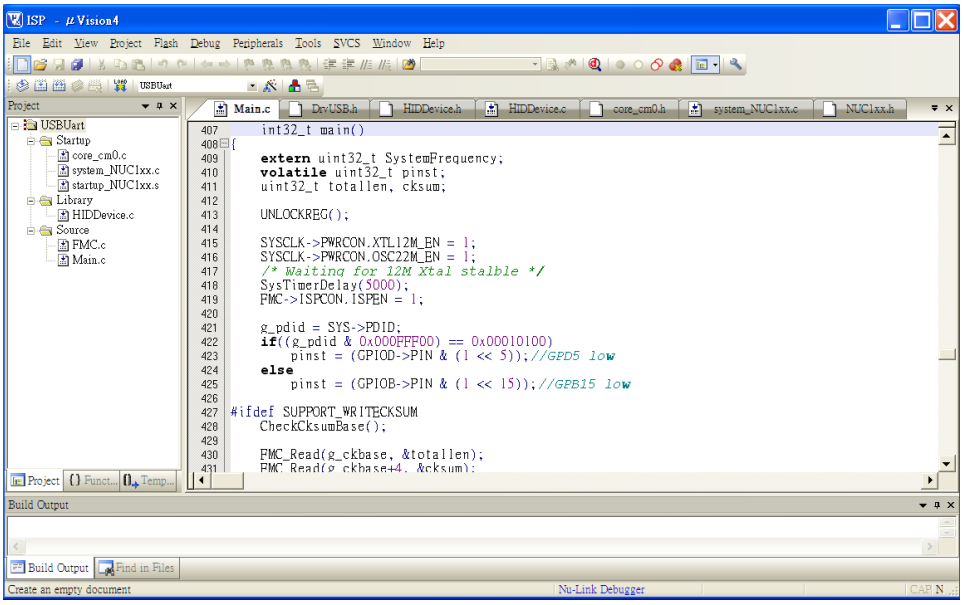


Figure 5. Working Window of ISP Project

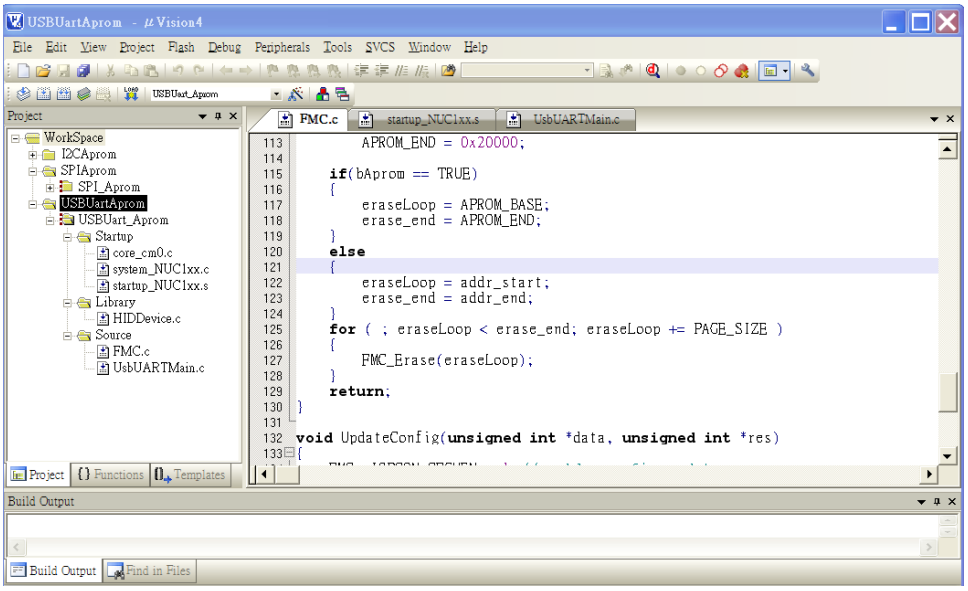


Figure 6. Working Window of USBUartAprom Project

The source files of master ISP program, *NuMicro ISP.exe*, can be found at directory "*NuMicro ISP Programming Tool Vx.xx*". A pre-built executable image named *ISP_Code_NUC100_vx.x.bin* for NUC100 *and ISP_Code_M051_vx.x.bin* for M051 is saved at the same directory for user to quickly test it.

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 5.

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 pin *GPB15(GPB15 for NUC100, GPD0 for NUC101)* is LOW for USB. Or receive packet from NuMicro ISP Programming Tool.exe for UART
- System starts according to a *SYSRESETREQ* event.

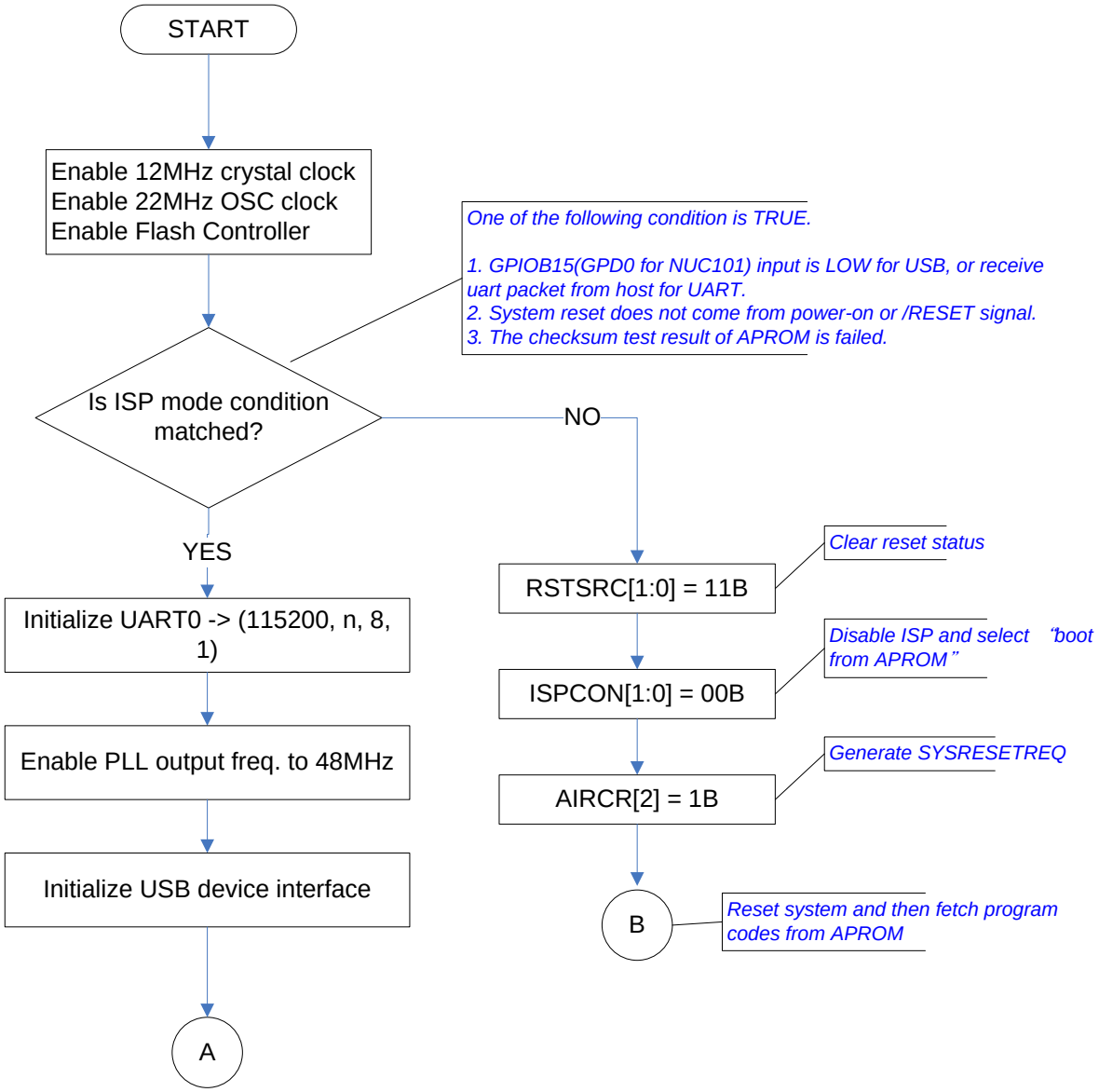


Figure 7. USBUart ISP Booting Flow

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 commands is received, the ISP program will select the desired booting mode and reset whole system.

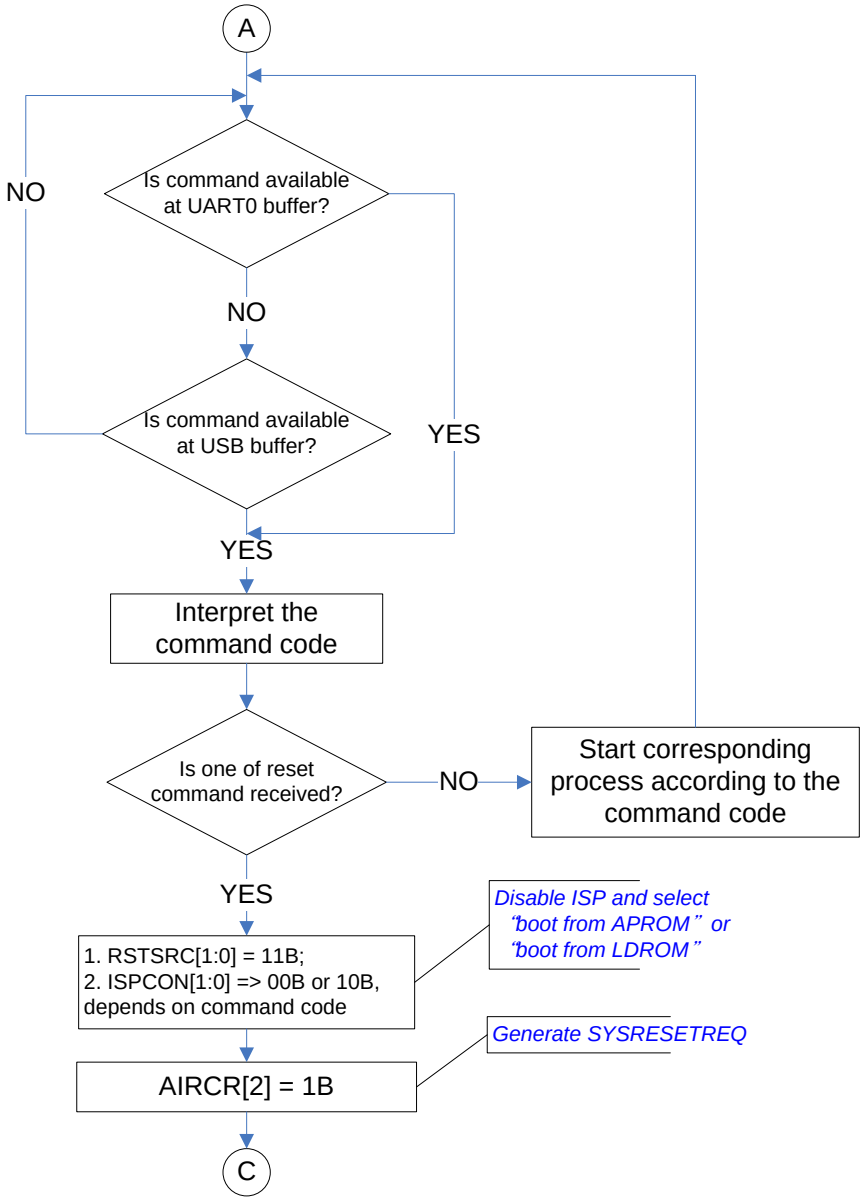


Figure 8. USBUart ISP Main Flow

4. ISP Through SPI

Two NuMicro™ MCUs are used to test the ISP function though SPI interface. One is the slave device that its APROM will be updated. The other one is the master device which sends command and data to slave device. Before starting to update application program in the slave's APROM, the master should checks the slave device whether it is running in ISP mode or not. If the application program saved at APROM of slave device is running, the master will issues command CMD_RUN_LDROM to force slave device to reset and start executing ISP program in LDROM.

4.1 Project and Source Files

All programs running on both master and slave devices are compiled by Keil uVision 4 IDE tool. The program of master device is *SPIMaster.bin*, its source code could be found at directory "*NuvotonPlatform_Keil\Master*". The project file *SPIMaster.uvproj* is saved at the same directory. The following diagram shows the screen snapshot after open it.

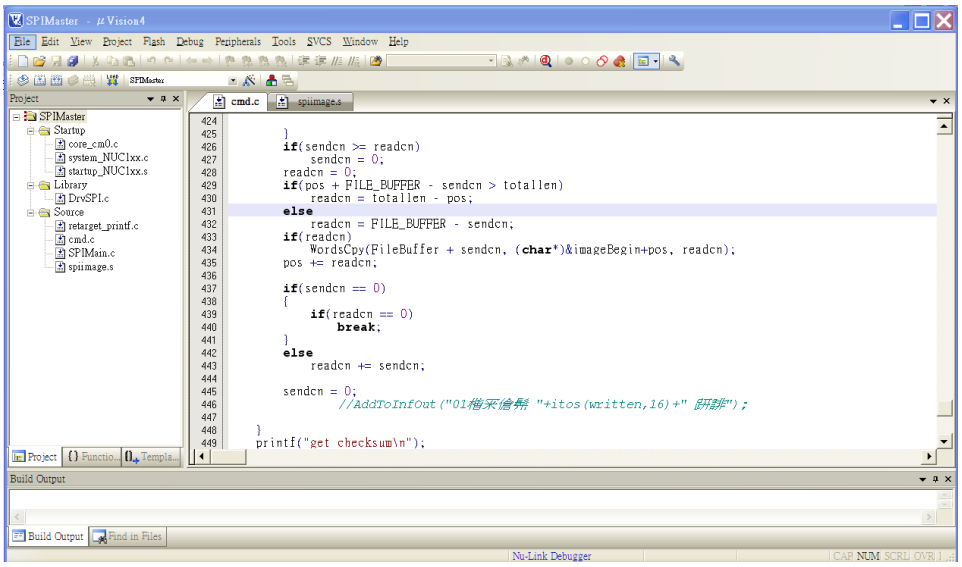


Figure 9. Working Window of SPIMaster Project

There're two programs running on slave device. The ISP and application program are saved in LDROM and APROM respectively. The source files of ISP code can be found at directory "*NuvotonPlatform_Keil\Slave\SPI*", and a *ISP.uvproj* project file is saved there. The source code of application program project file *SPIAprom.uvmpw* are saved at directory "*NuvotonPlatform_Keil\Slave\Aprom*". The following diagrams show the screen snapshot after open the projects.

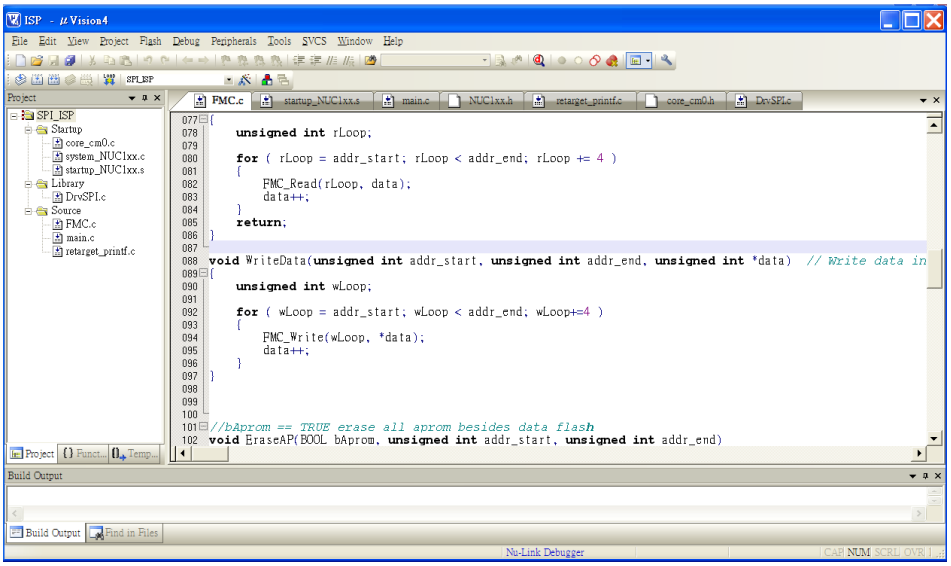


Figure 10. Working Window of ISP Project

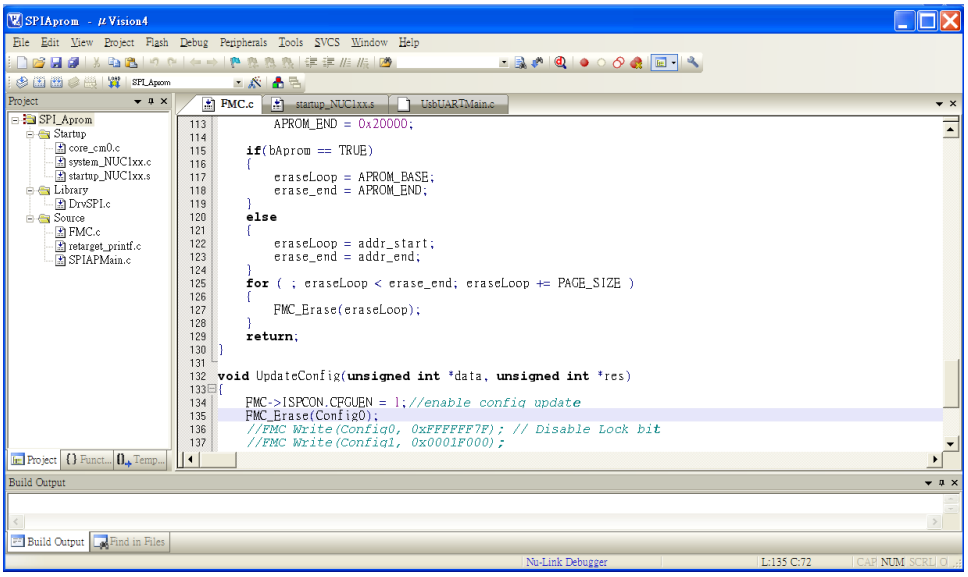


Figure 11. Working Window of SPIAprom Project

4.2 Communication Interface

The SPI I/O pins of both NuMicro™ MCUs should be connected together to test ISP function. Each transmitted packet size through SPI interface is fixed at 64 bytes. The connected method is shown as the following diagram.

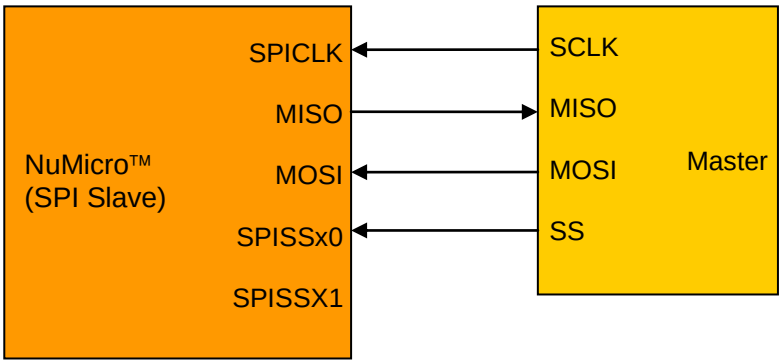


Figure 12. SPI Connect Block Diagram (Redraw)

4.3 The SPI ISP Program

When ISP receives packet from SPI master program, it will parse the packet and take an action according to the command code received. the size of packet is 64 bytes. And the format is depicted in Chapter 5.

4.3.1 Data transfer on the SPI bus

SPI feature on SPI slave device

Bit width	8 bit
Select active level	Low level
Clock idle	high
Rx edge	negative
Tx edge	negative

.

4.3.2 Commands response delay

After send SPI packet, SPI master need wait for a while to receive ack packet. The delay time as following table.

Command	Command Code	Delay
CMD_UPDATE_APROM	0x000000A0	8050ms
Format2	0x00000000	100ms
CMD_UPDATE_CONFIG	0x000000A1	50ms.
CMD_READ_CONFIG	0x000000A2	50ms
CMD_ERASE_ALL	0x000000A3	50ms
CMD_SYNC_PACKNO	0x000000A4	50ms.
CMD_GET_FWVER	0x000000A6	50ms.
CMD_GET_DEVICEID	0x000000B1	50ms.
CMD_UPDATE_DATAFLASH	0x000000C3	8050ms
Format2		100ms
CMD_RUN_APROM	0x000000AB	500ms.
CMD_RUN_LDROM	0x000000AC	500ms.
CMD_RESET	0x000000AD	500ms.
CMD_WRITE_CHECKSUM	0x000000C9	200ms.
CMD_GET_FLASHMODE	0x000000CA	50ms.
CMD_RESEND_PACKET	0x000000FF	200ms
CMD_CONNECT	0x000000AE	50ms

4.3.3 ISP Booting Flow

The booting flow is similar to the ISP program using USB and UART. Please reference section 2.3.1 for detailed description of booting method.

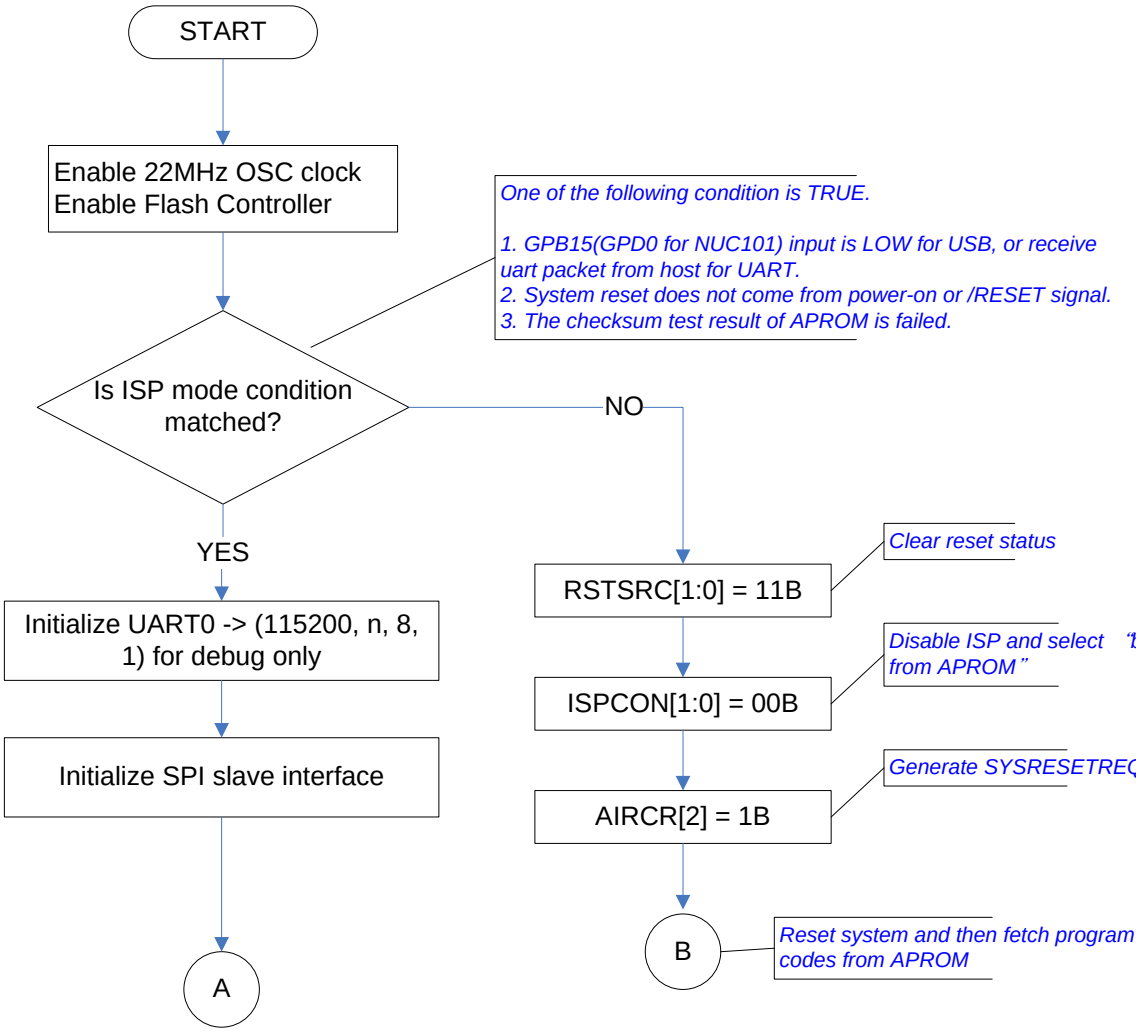


Figure 13. SPI ISP Booting Flow

4.3.4 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 commands is received, the ISP program will select the desired booting mode and reset whole system.

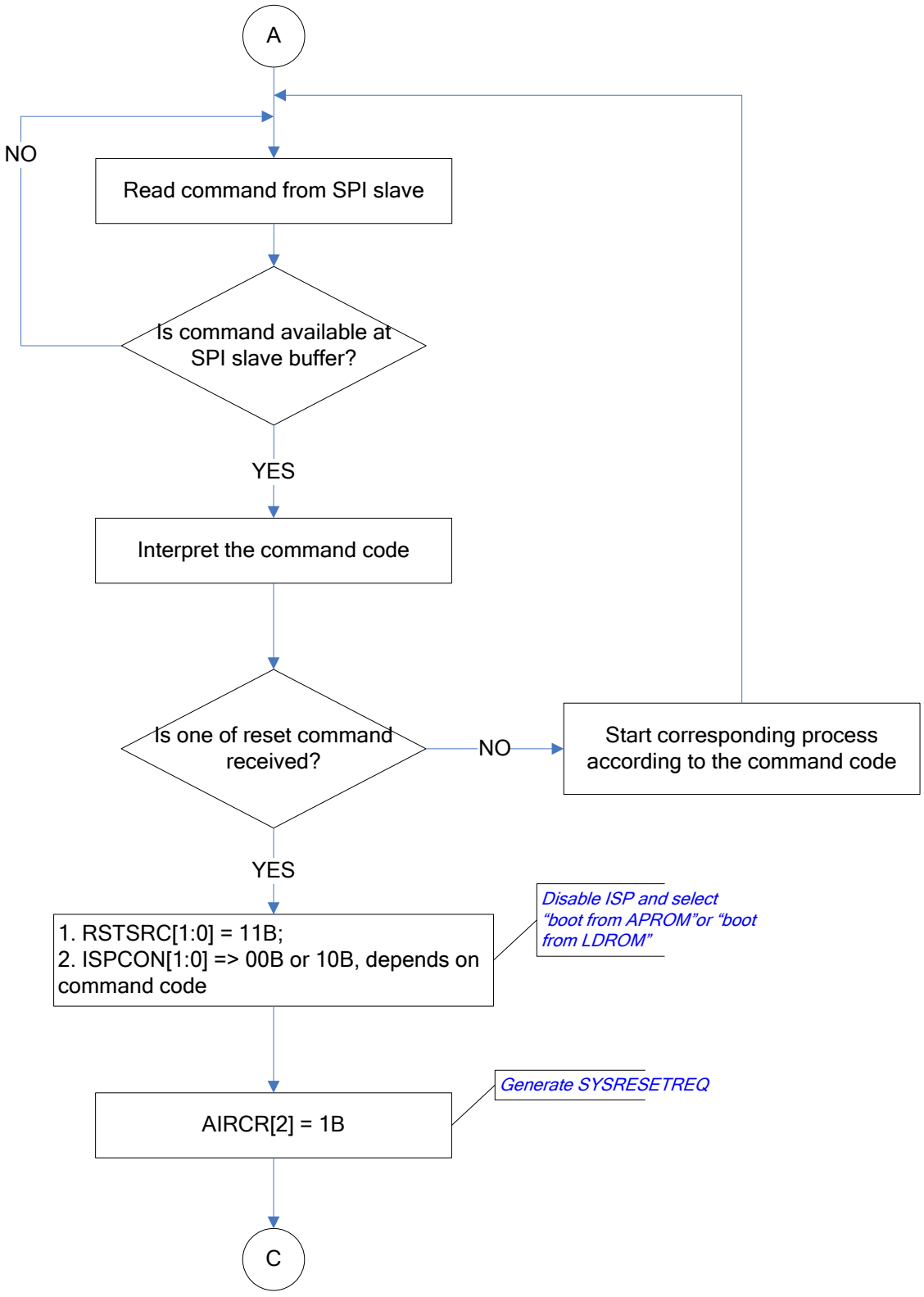


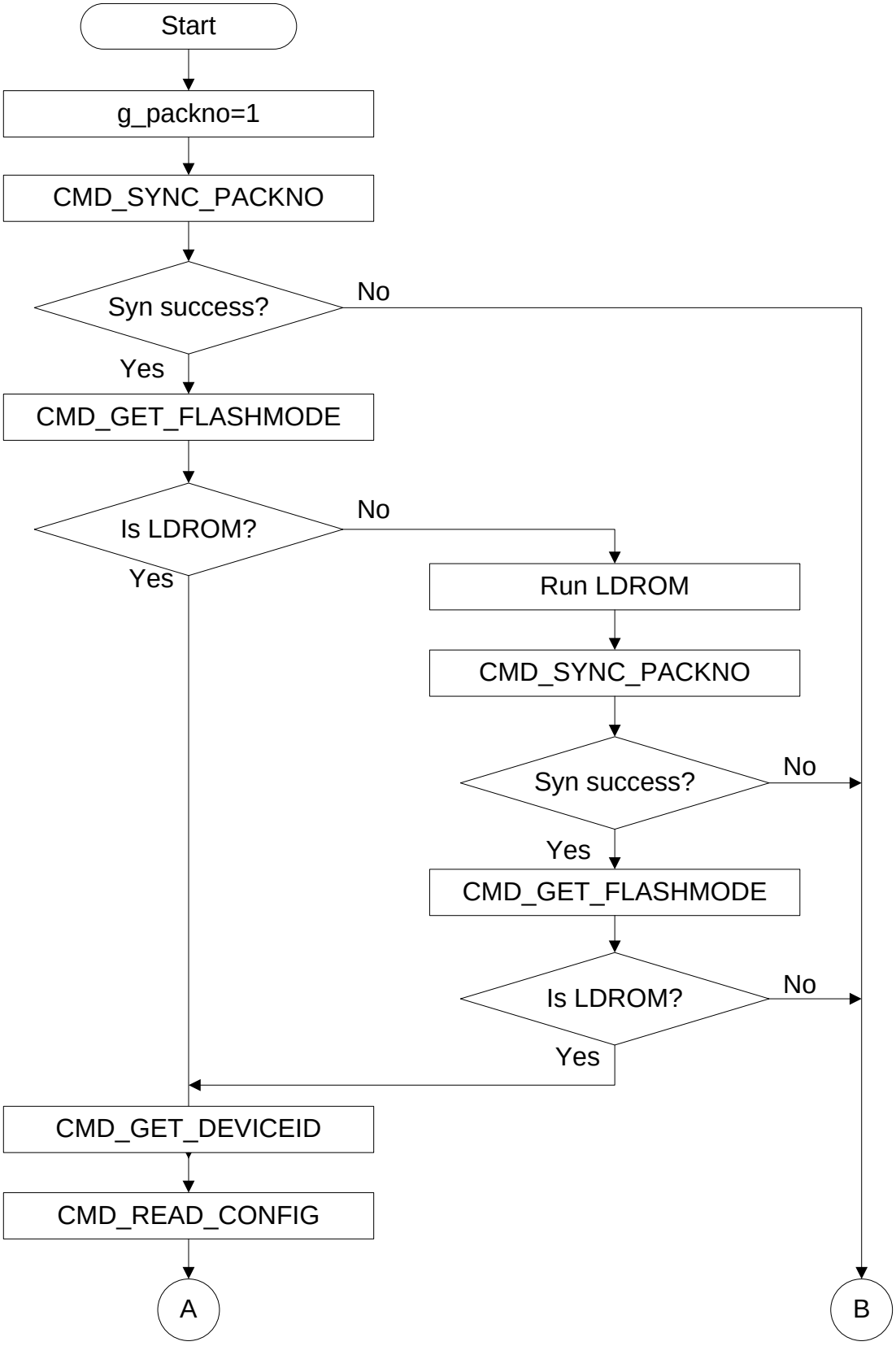
Figure 14. SPI ISP Main Flow

4.4 The Example Application and Master ISP Program

User can rebuild the programs from the SPI ISP source code. After the programs are built, please connect two NuMicro™ boards by SPI0 and do the following operations to test the ISP function.

- Burn the *SPIMaster.bin*, saved at the sub-directory *Master*, into the APROM of Master device.
- Burn the *spiaprom.bin*, saved at the sub-directory *Slave**Aprom*, into the APROM of slave device.
- Burn *ISP.bin*, saved at the sub-directory *Slave**SPI*, into LDROM of slave device.

After all programs are burned into flash, please firstly power on slave device, and then master device. The master will update *spiaprom.bin* which is stored in the sub-directory \Master into the APROM of slave device. The following diagram shows the control flow of master ISP program.



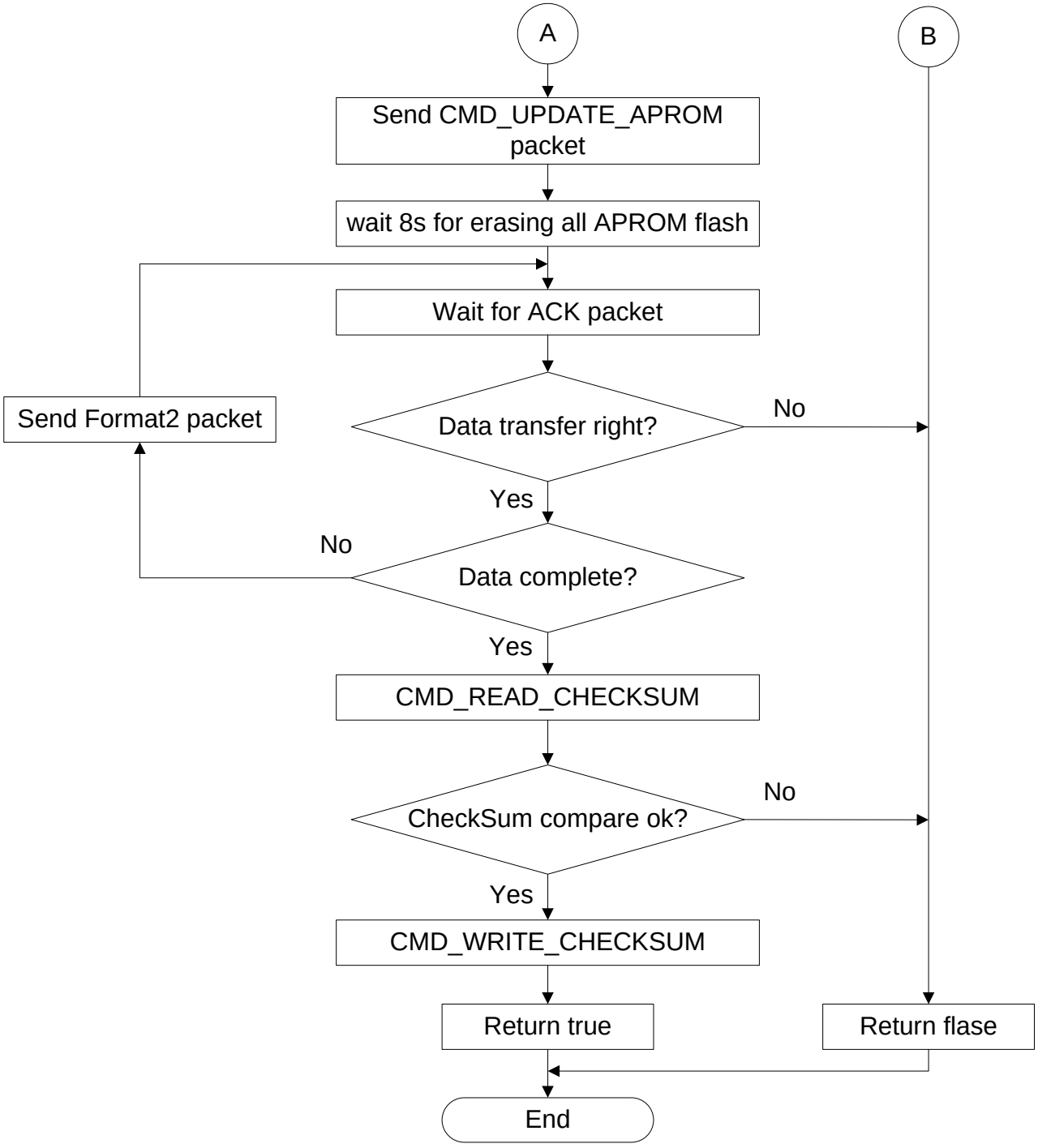


Figure 15. SPI/I2C Master Main Flow

5. ISP Through I2C

Two NuMicro™ MCUs are used to test the ISP function though I2C interface. One is the slave device that its APROM will be updated. The other one is the master device which sends command and data to slave device. Before starting to update application program in the slave's APROM, the master should check the slave device whether it is running in ISP mode or not. If application program saved at APROM of slave device is running, the master will issues command CMD_RUN_LDROM to force slave device to reset and start executing ISP program in LDROM.

5.1 Project and Source Files

All programs running on both master and slave devices are created by using Keil uVision 4 IDE tool. The program of master device is *I2CMaster.bin*, its source code could be found at directory "*NuvotonPlatform_Keil\Master*". The project file *I2CMaster.uvproj* is saved at the same place. The following diagram shows the screen snapshot after open it.

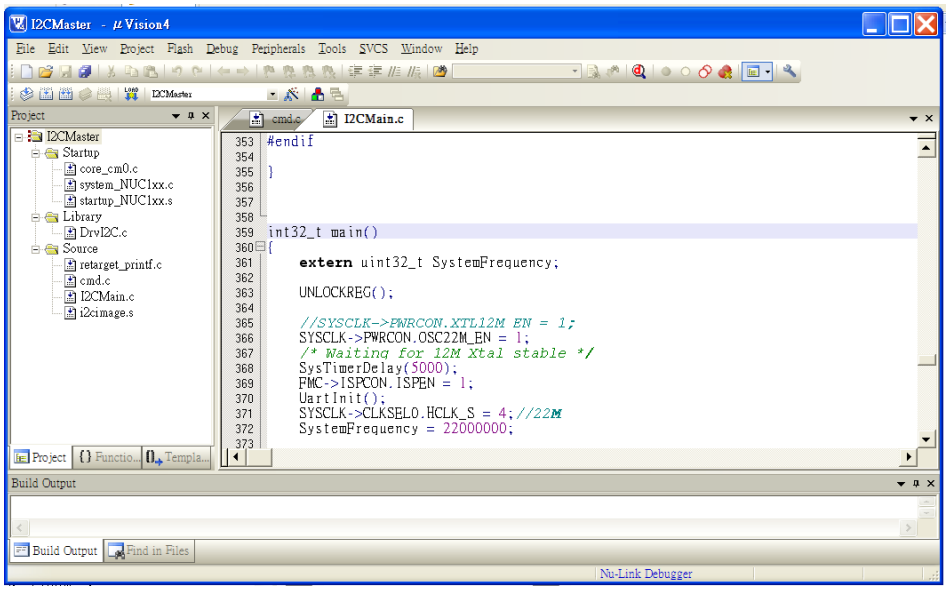


Figure 16. Working Window of I2CMaster Project

There're two programs running on slave device. The ISP and example application program are saved in LDROM and APROM respectively. The source files of ISP code can be found at directory "*NuvotonPlatform_Keil\Slave\I2Cx*", and a *ISP.uvproj* project file is saved there. The source code of example application program and project file I2C*Aprom.uvmpw* are saved at directory "*NuvotonPlatform_Keil\Slave\Aprom*". The following diagrams show the screen snapshot after open the projects.

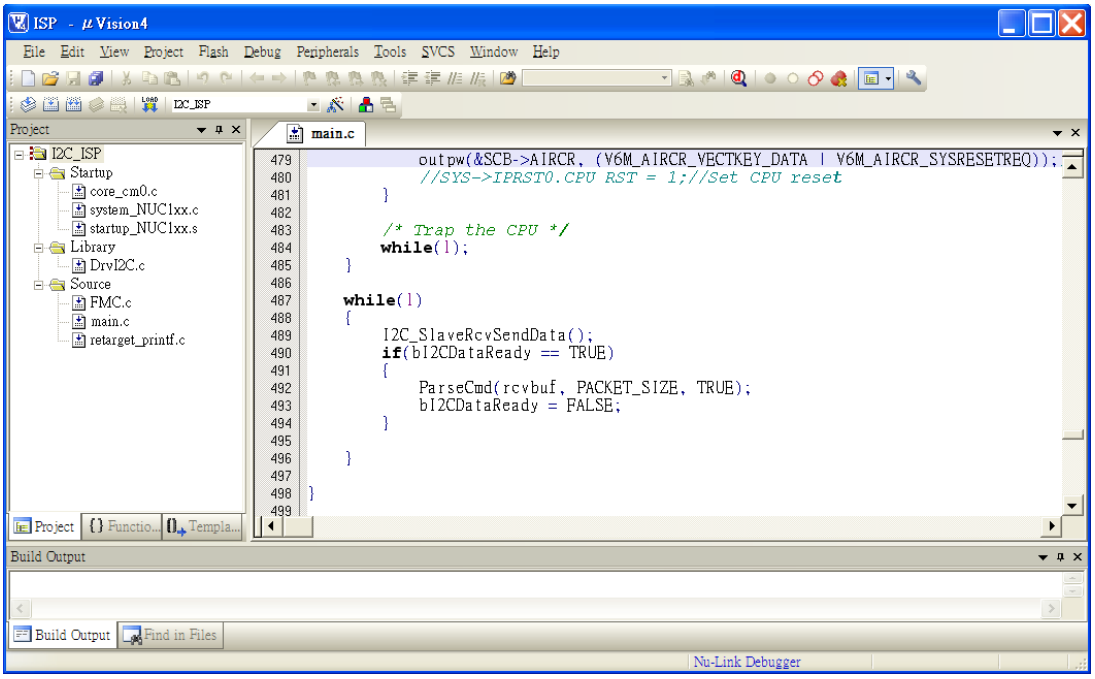


Figure 17. Working Window of ISP Project

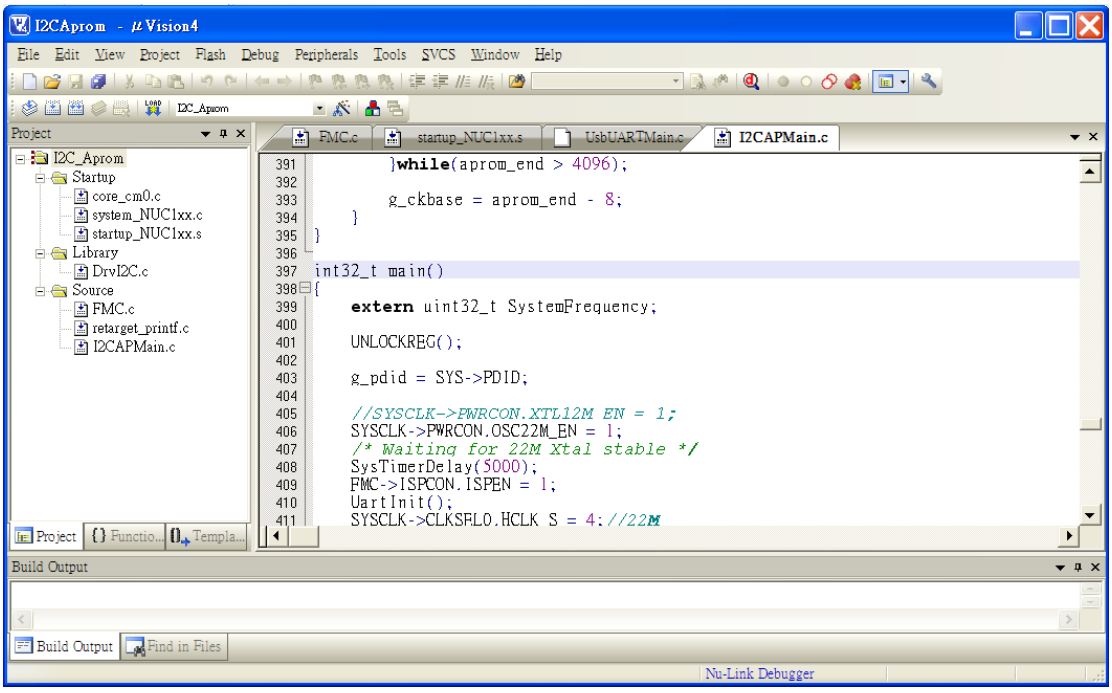


Figure 18. Working Window of I2CAprom Project

5.2 Communication Interface

The I2C I/O pins of both NuMicro™ MCUs should be connected together to test ISP function. Each transmitted packet size through I2C interface is fixed at 64 bytes. The connected method is shown as the following diagram.

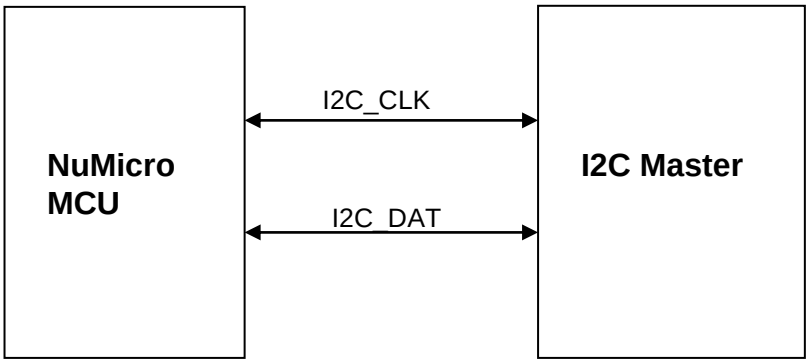


Figure 19. I2C Connection Block Diagram

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. And the format is depicted in Chapter 5.

5.3.1 Data transfer on the I2C bus

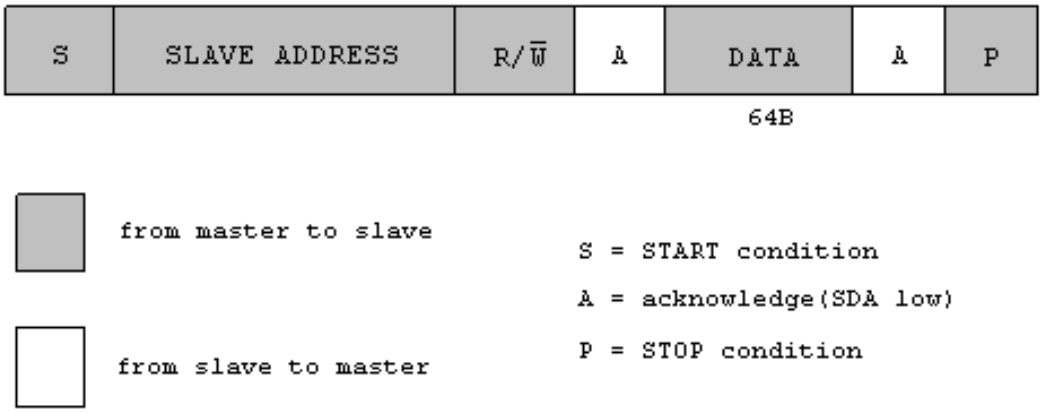


Figure 20. Write data to Slave device by Master device

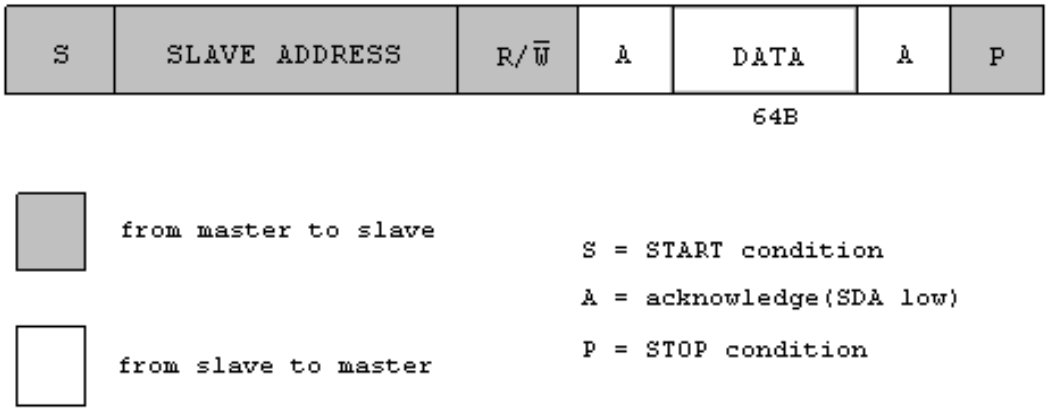


Figure 21. Read data from Slave device by Master device

5.3.2 Commands response delay

After send SPI packet, SPI master need wait for a while to receive ack packet. The delay time as following table.

Command	Command Code	Delay
CMD_UPDATE_APROM	0x000000A0	8050ms
Format2	0x00000000	100ms
CMD_UPDATE_CONFIG	0x000000A1	50ms.
CMD_READ_CONFIG	0x000000A2	50ms
CMD_ERASE_ALL	0x000000A3	50ms
CMD_SYNC_PACKNO	0x000000A4	50ms.
CMD_GET_FWVER	0x000000A6	50ms.
CMD_GET_DEVICEID	0x000000B1	50ms.
CMD_UPDATE_DATAFLASH	0x000000C3	8050ms.
Format2		100ms
CMD_RUN_APROM	0x000000AB	500ms.
CMD_RUN_LDROM	0x000000AC	500ms.
CMD_RESET	0x000000AD	500ms.
CMD_WRITE_CHECKSUM	0x000000C9	200ms.
CMD_GET_FLASHMODE	0x000000CA	50ms.
CMD_RESEND_PACKET	0x000000FF	200ms
CMD_CONNECT	0x000000AE	50ms

5.3.3 ISP Booting Flow

The booting flow is similar to the ISP program using USB and UART. Please reference section 2.3.1 for detailed description of booting method.

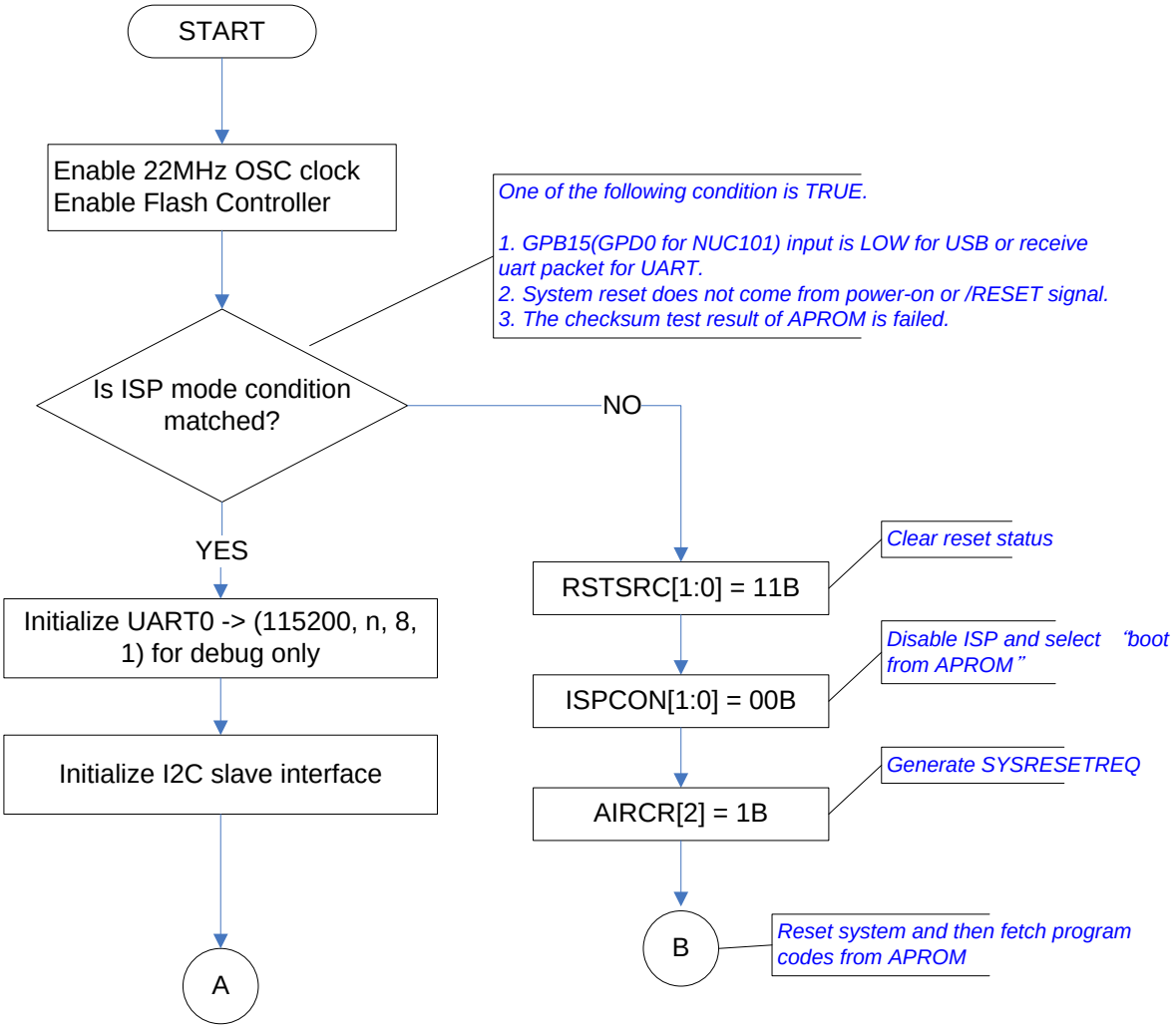


Figure 22. I2C ISP Booting Flow

5.3.4 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 commands is received, the ISP program will select the desired booting mode and reset whole system.

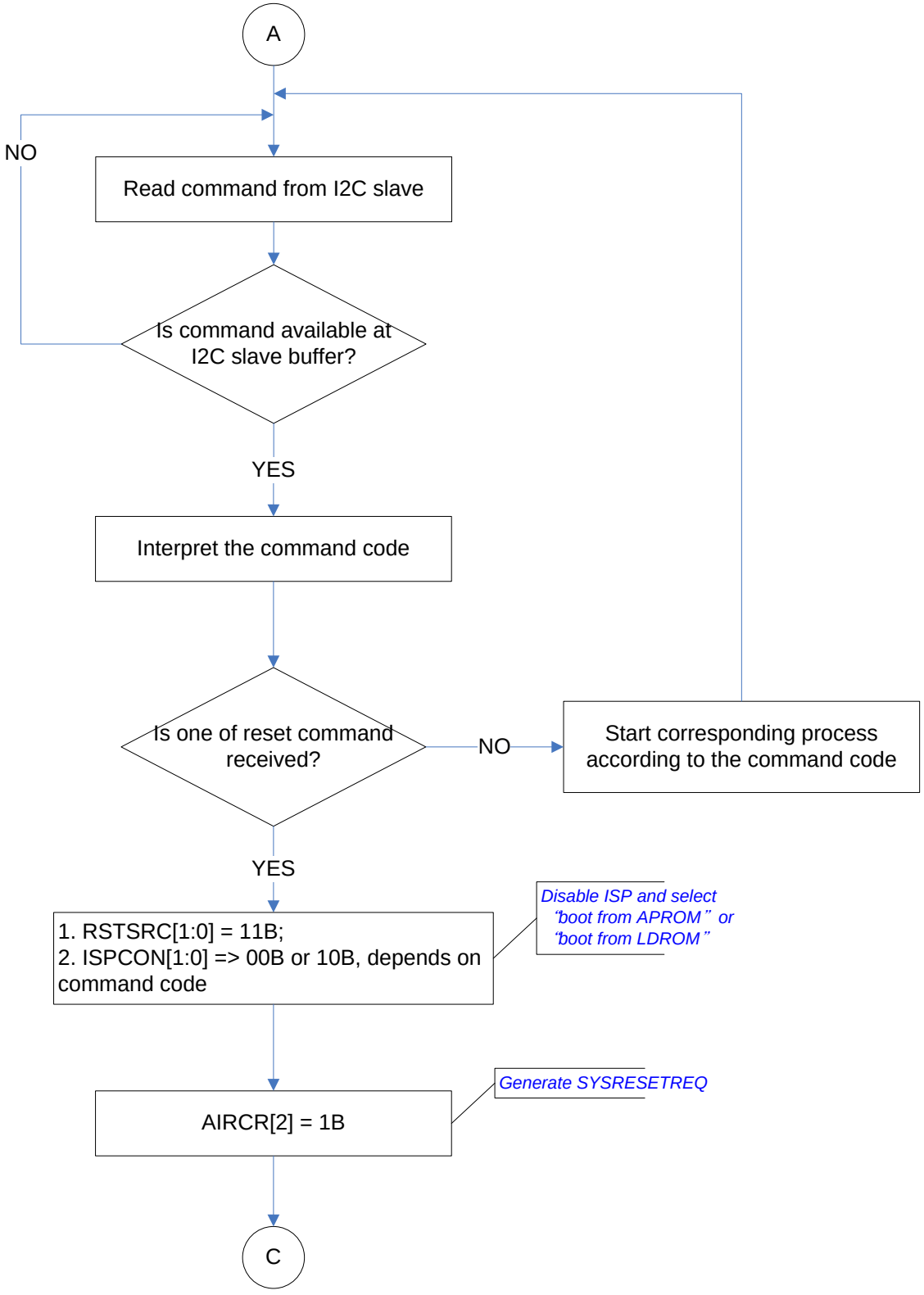


Figure 23. I2C ISP Main Flow

5.4 The Example Application and Master ISP Program

User can rebuild the programs from the I2C ISP source code. After the programs are built, please do the following operations to test the ISP function.

- Burn the *I2CMaster.bin*, saved at the sub-directory *Master*, into the APROM of Master device.
- Burn the *i2caprom.bin*, saved at the sub-directory *Slave**Aprom*, into the APROM of slave device.
- Burn *ISP.bin*, saved at the sub-directory *Slave**I2C*, into LDROM of slave device.

After all programs are burned into flash, please firstly power on slave device, and then master device. The master will update *i2caprom.bin* which is stored in the sub-directory \Master into the APROM of slave device. The master ISP program is the same for both I2C and SPI modes. Please refer to section 3.4 Figure 13 for detailed control flow of master ISP program..

6. ISP Command Set

ISP packet is **64** byte fixed no matter command packet or ack packet. The ISP supported commands are listed as *Table 1* below. Each command is described detailed in this section and each command provides host and ISP flow chart, the "HOST Side" flow chart is the flow chart of the "NuMicro ISP Programming Tool.exe". if user want to write your own host application, you can refer to it

Table 1. ISP commands

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

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 *5.5 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.

6.1 Command CMD_UPDATE_APROM(0xA0):

Command Format

Host Send (named Format1, unit by byte):

4	4	4	4	n
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.

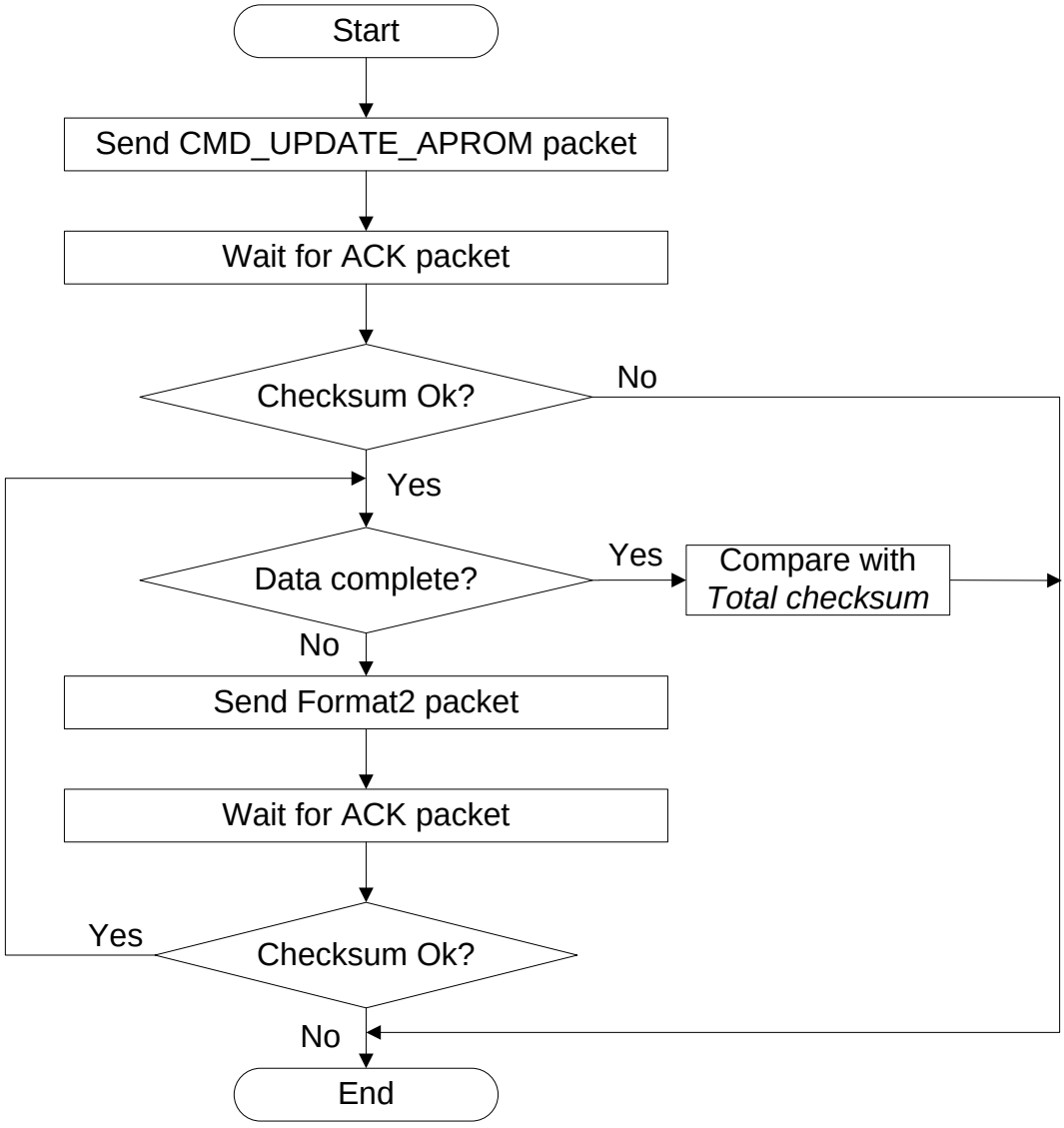


Figure 24. Program APROM Command: Host Side

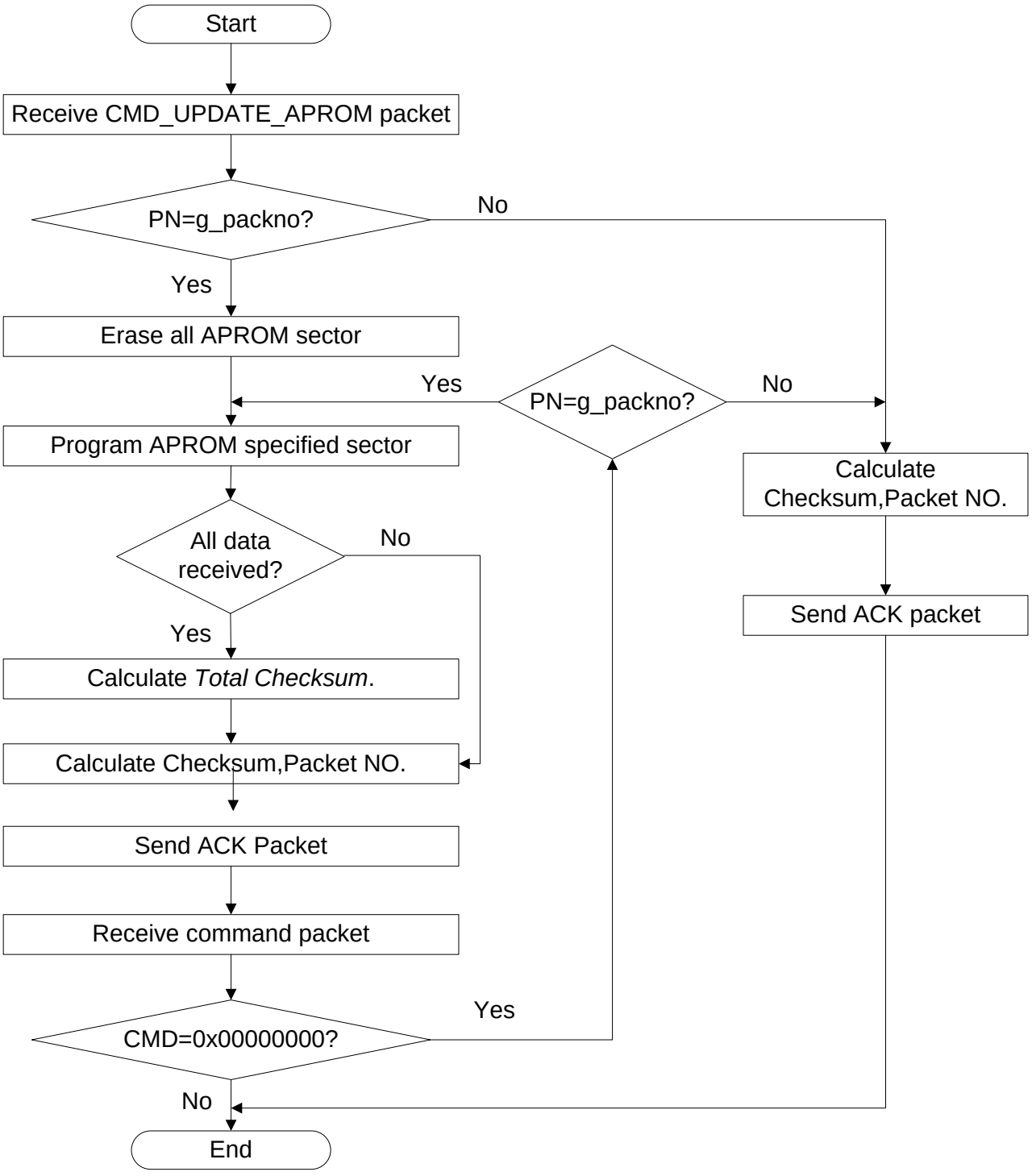


Figure 25. Program APROM Command: Device Side

Note: PN is packet number host sent; g_packno is ISP local packet number.

6.2 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.

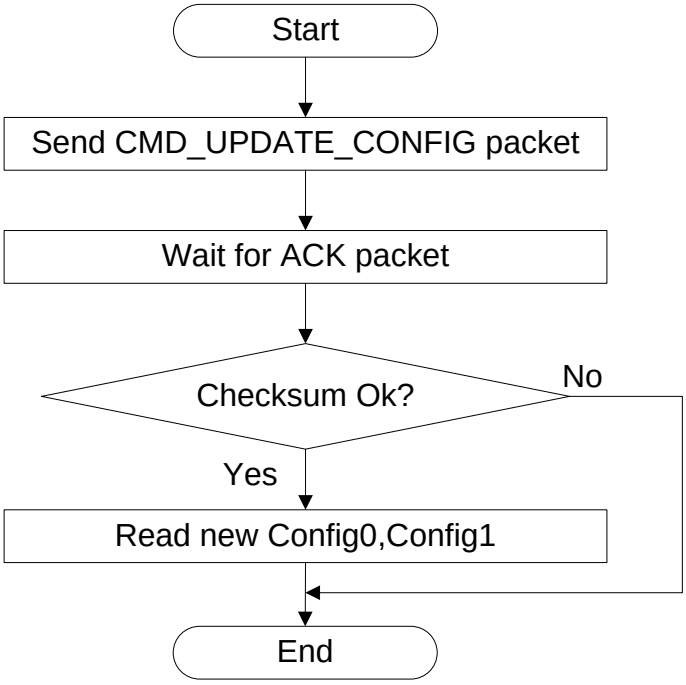


Figure 26. Update *Config0* and *Config1* Command: Host Side

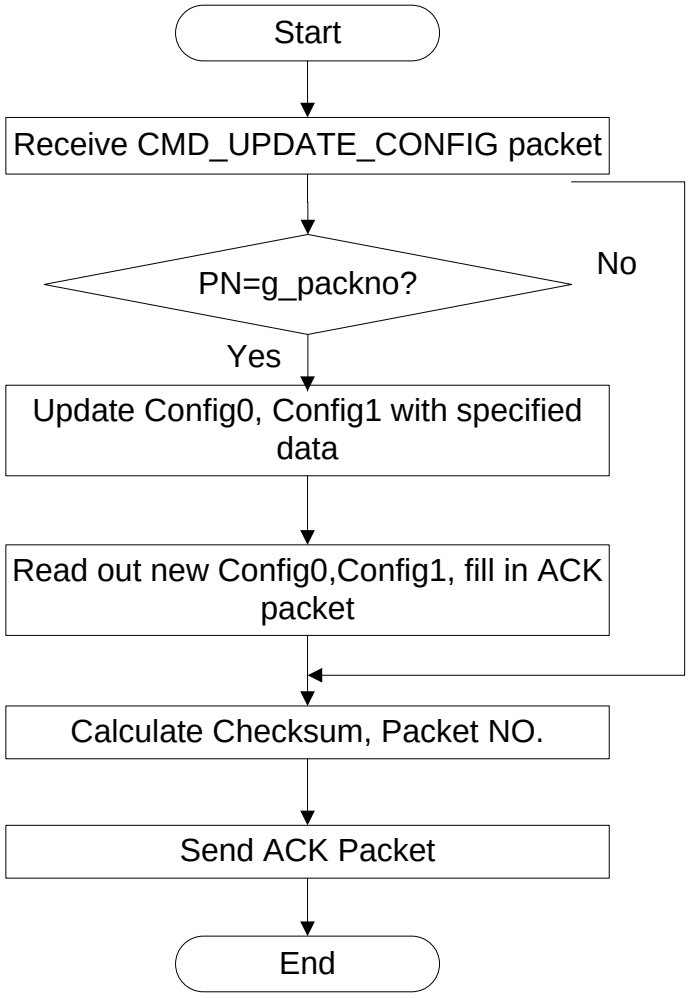


Figure 27. Update *Config0* and *Config1* Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.3 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 checks 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.

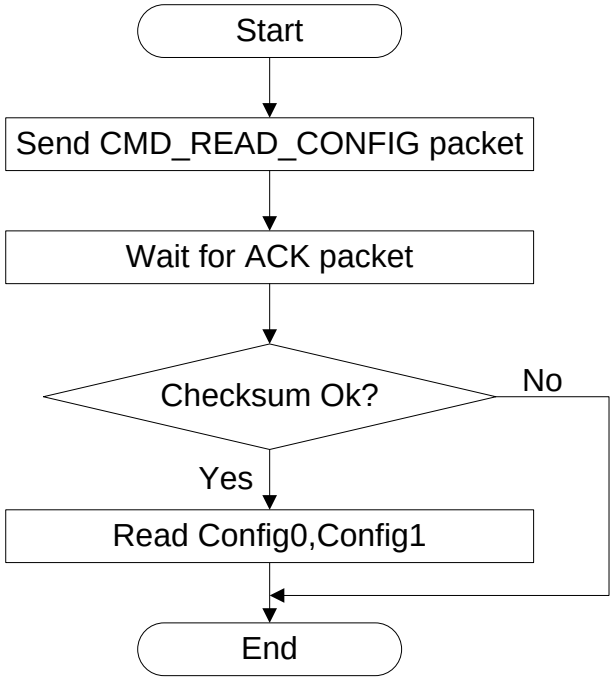


Figure 28. Read *Config0* and *Config1* Command: Host Side

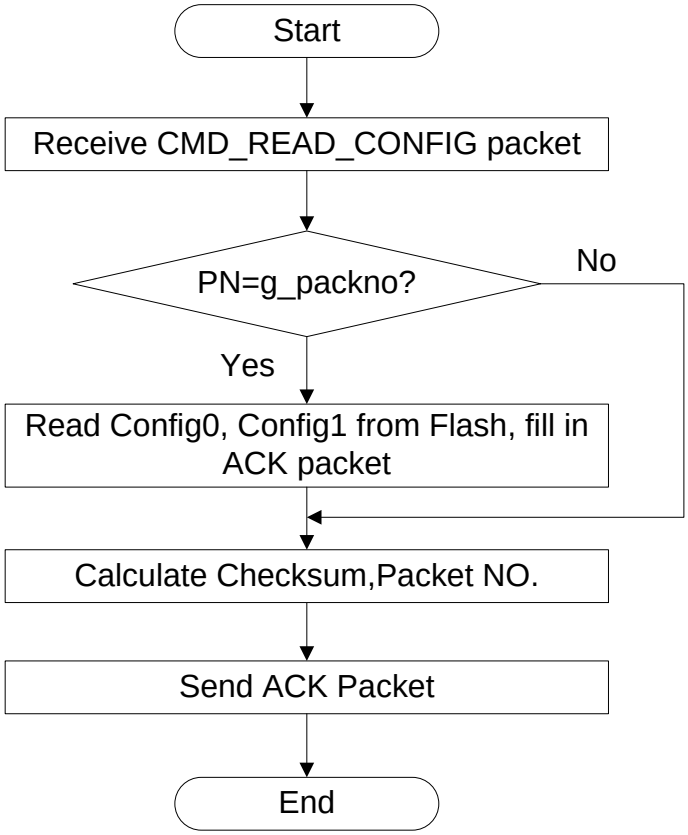


Figure 29. Read *Config0* and *Config1* Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.4 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.

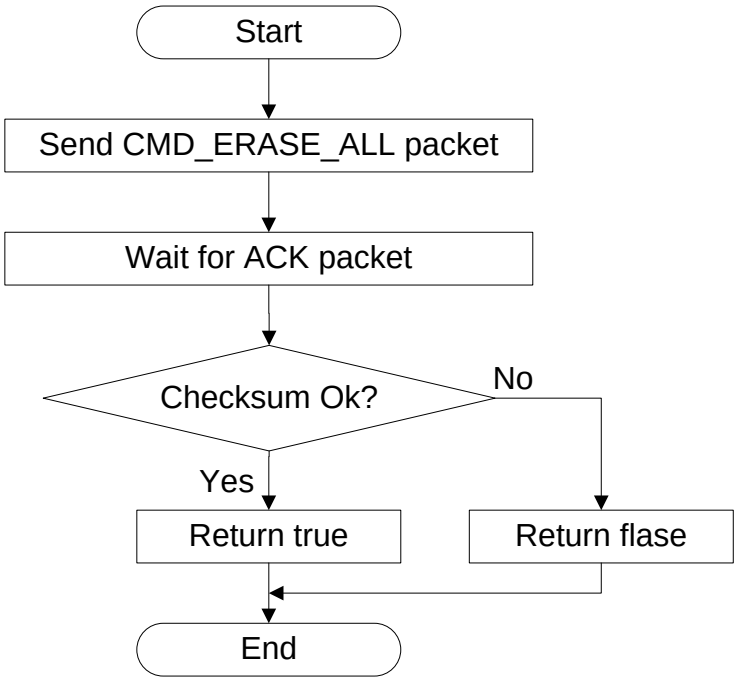


Figure 30. Erase Flash Memory Command: Host Side

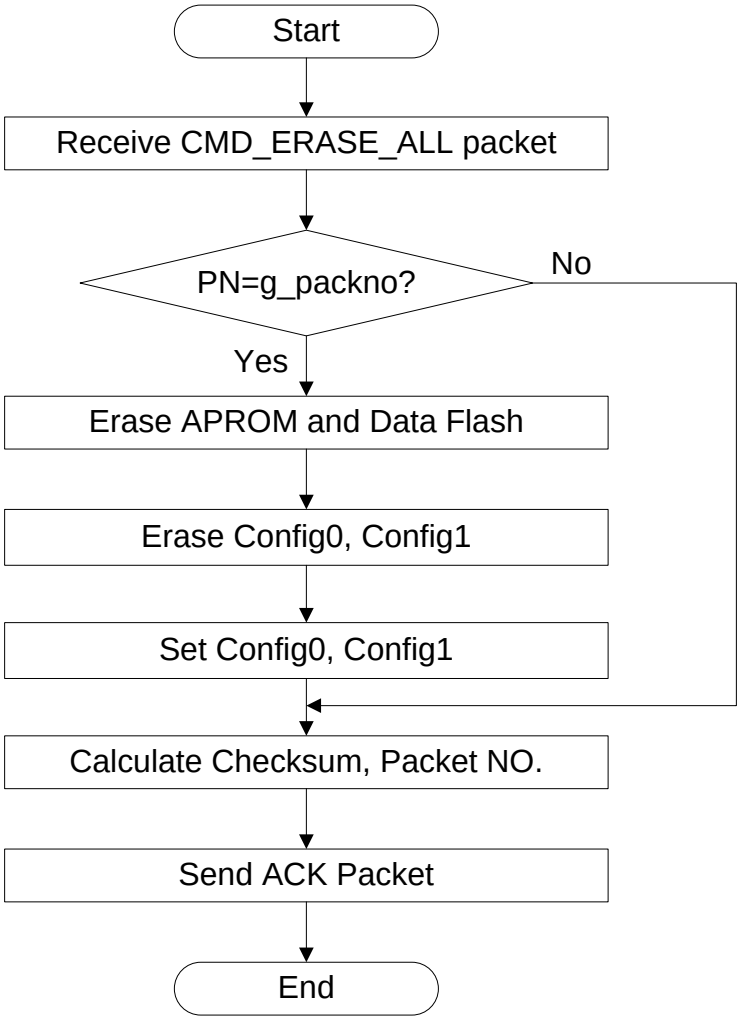


Figure 31. Erase Flash Memory Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.5 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.

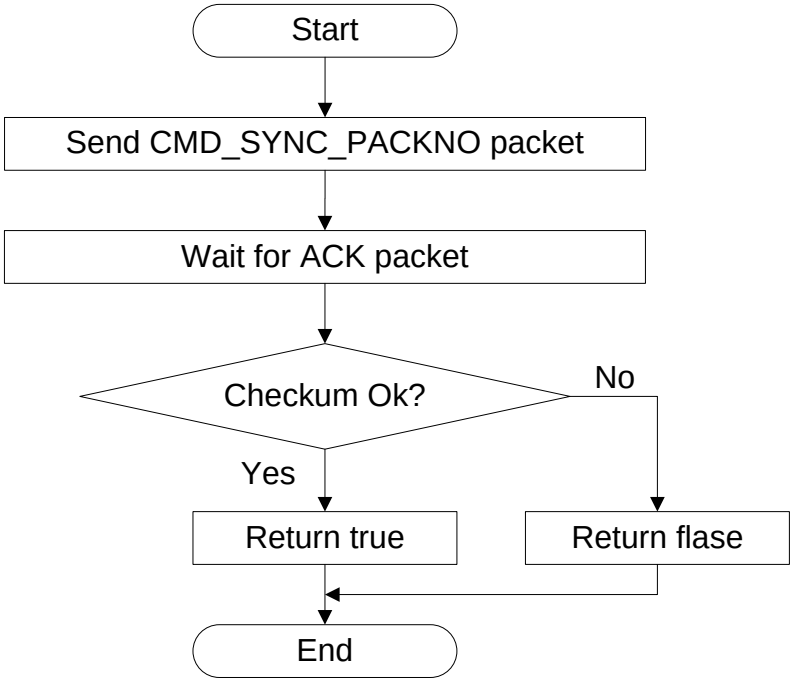


Figure 32. Sync Packet Number Command: Host Side

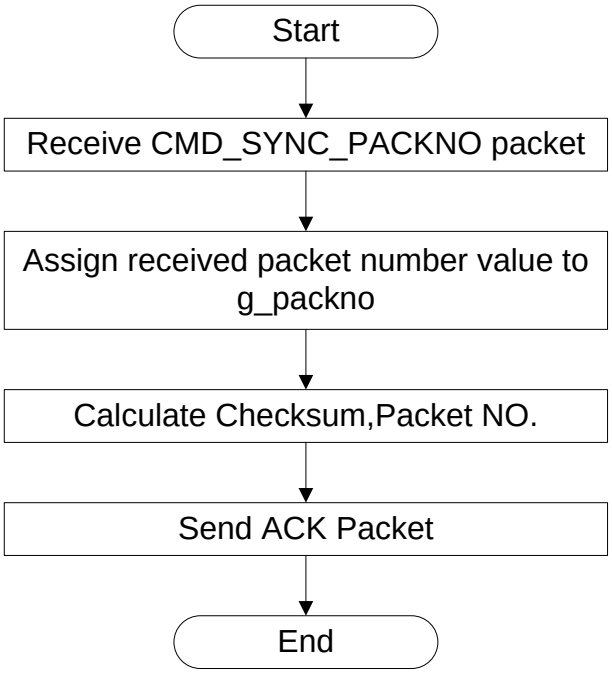


Figure 33. Sync Packet Number Command: Device Side

6.6 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 checks 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.

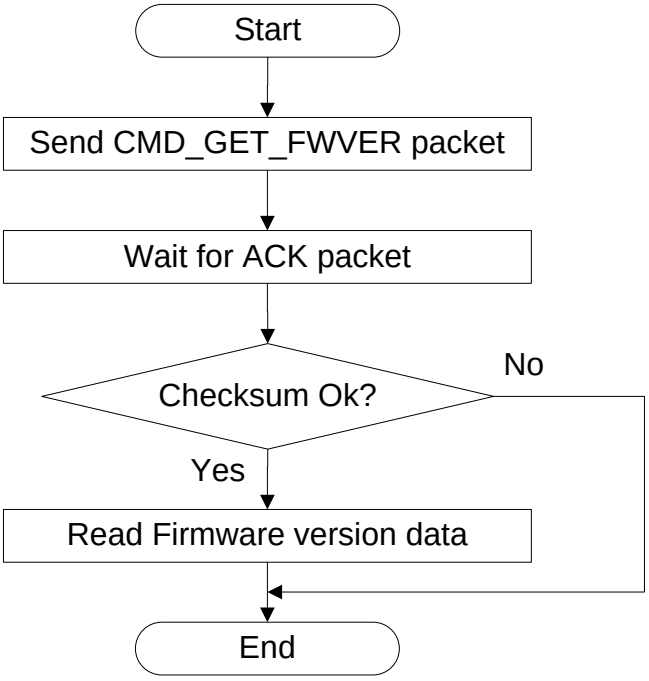


Figure 34. Get ISP Version Command: Host Side

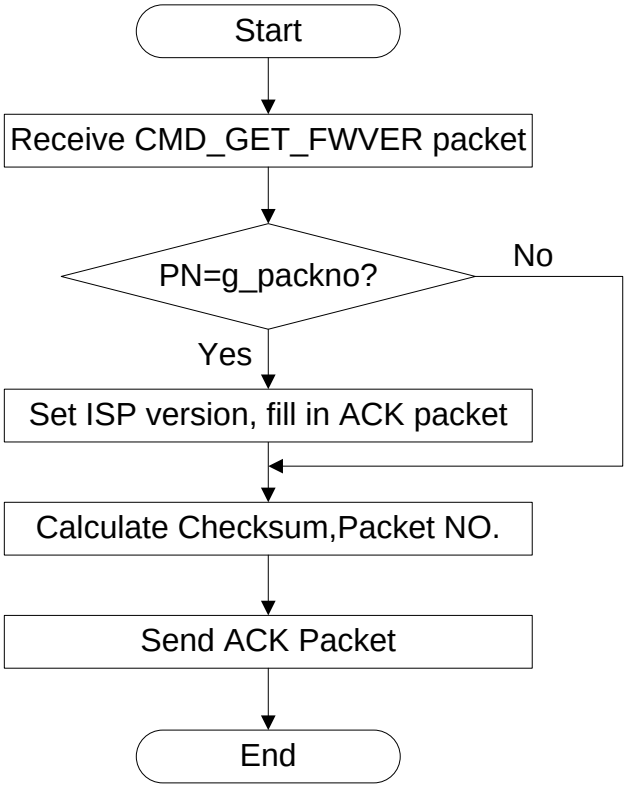


Figure 35. Get ISP Version Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.7 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.

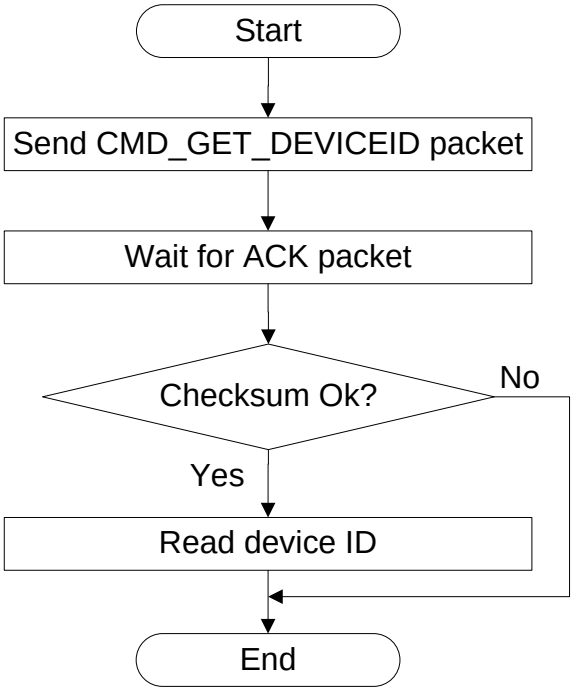


Figure 36. Get product ID Command: Host Side

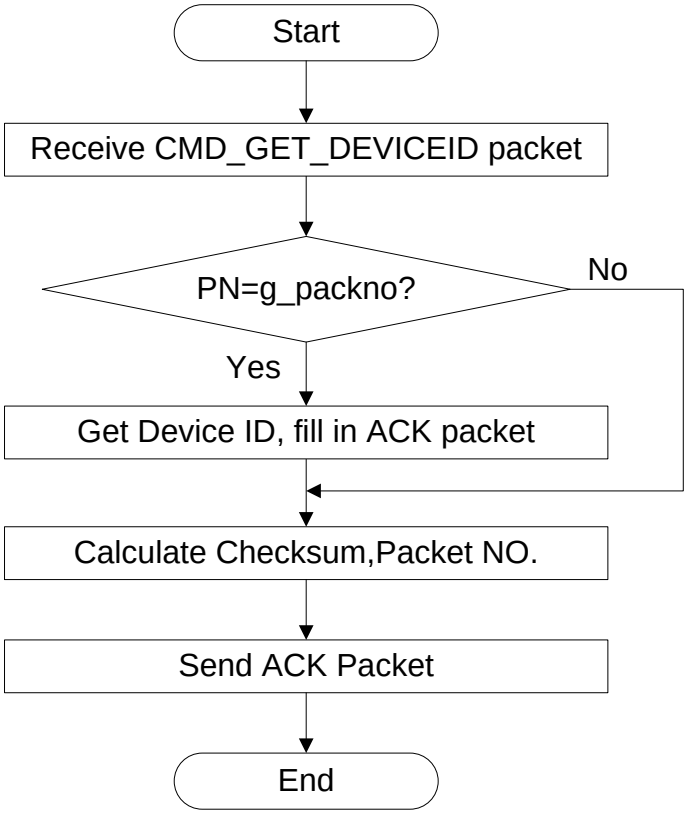


Figure 37. Get product ID Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.8 Command CMD_UPDATE_DATAFLASH (0xC3):

Command Format

Host Send (named Format1):

4	4	4	4	n
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.

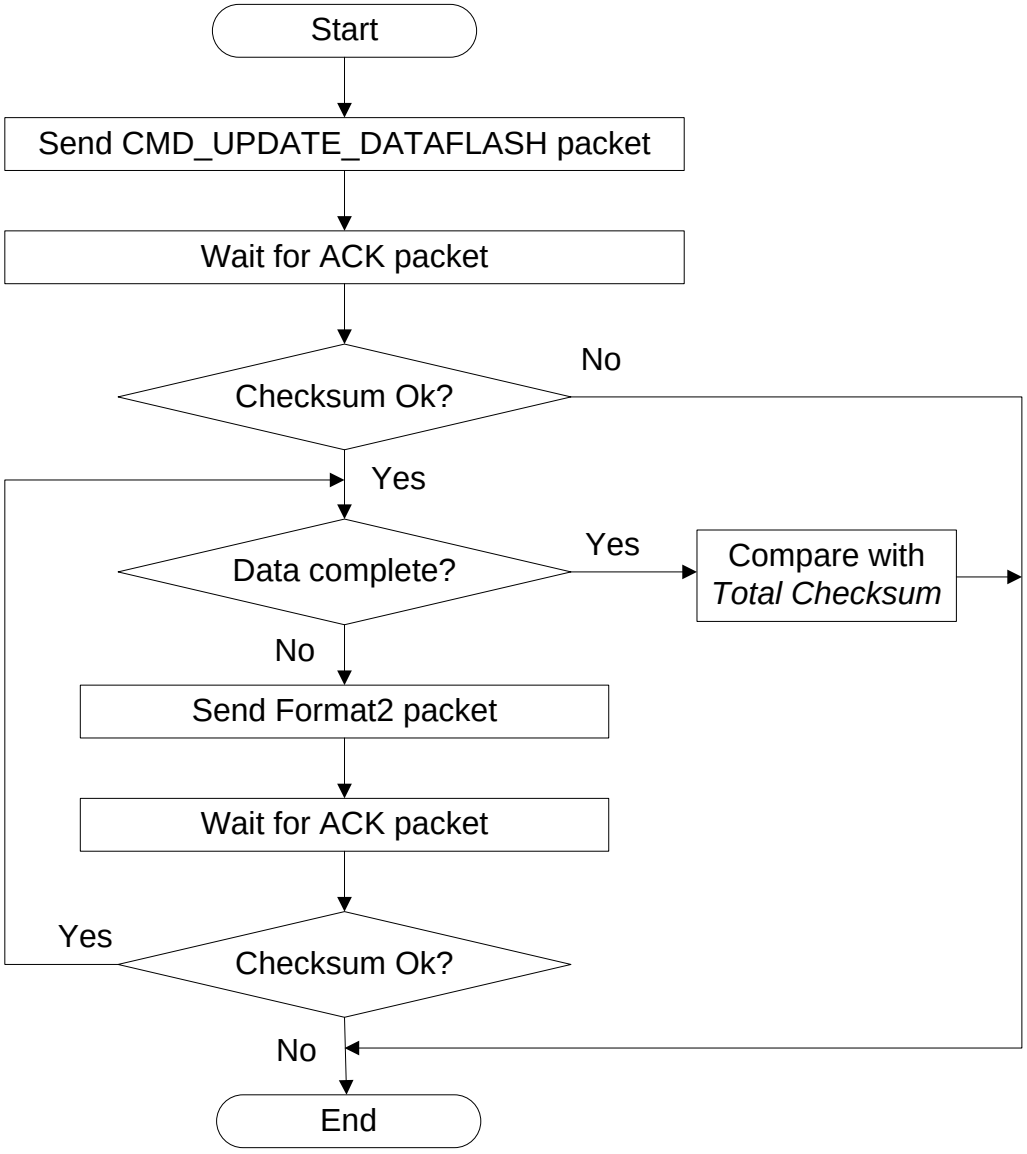


Figure 38. Program Data Flash Command (2): Host Side

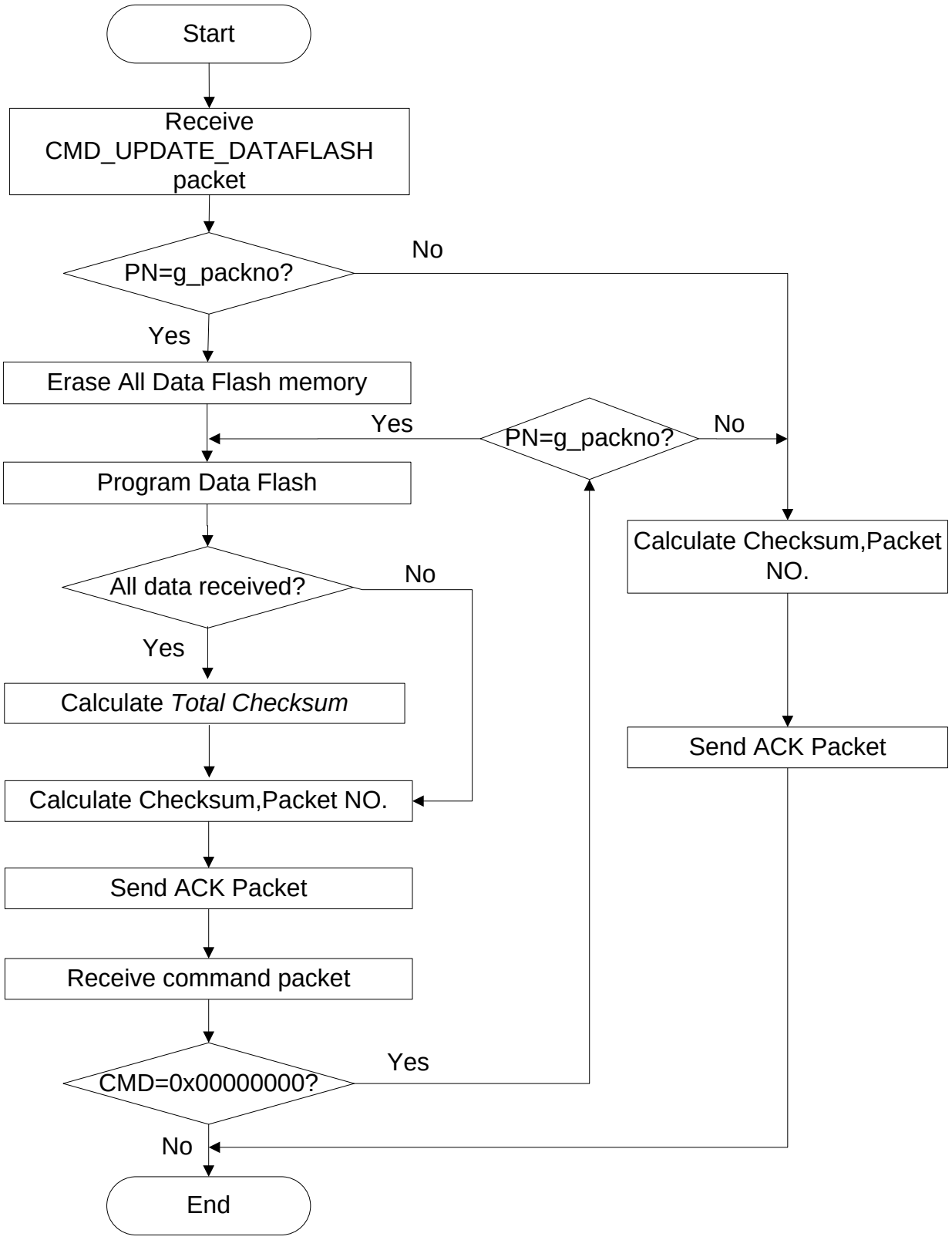


Figure 39. Program Data Flash Command (2): Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.9 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.

.

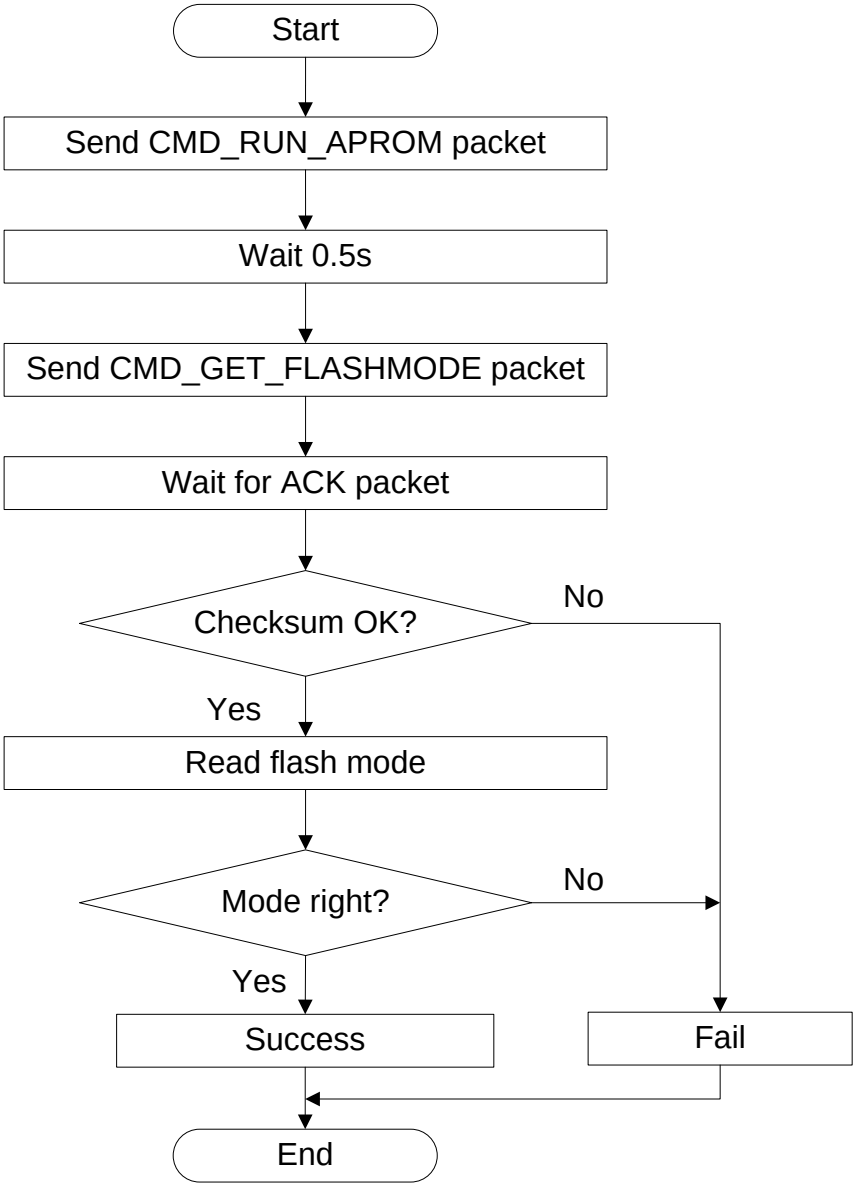


Figure 40. Run APROM Command: Host Side

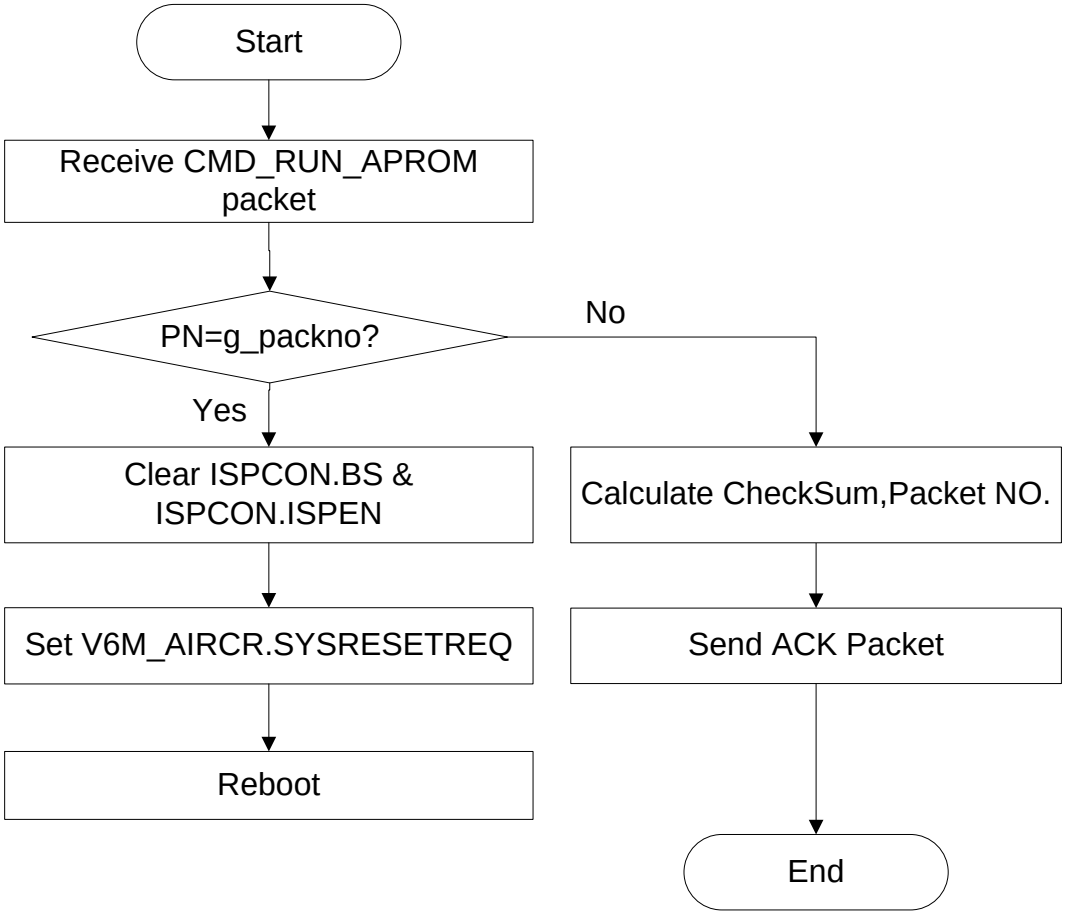


Figure 41. Run APROM Command: Device Side

Note: PN is packet number host send, g_packno is ISP local packet number.

6.10 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.

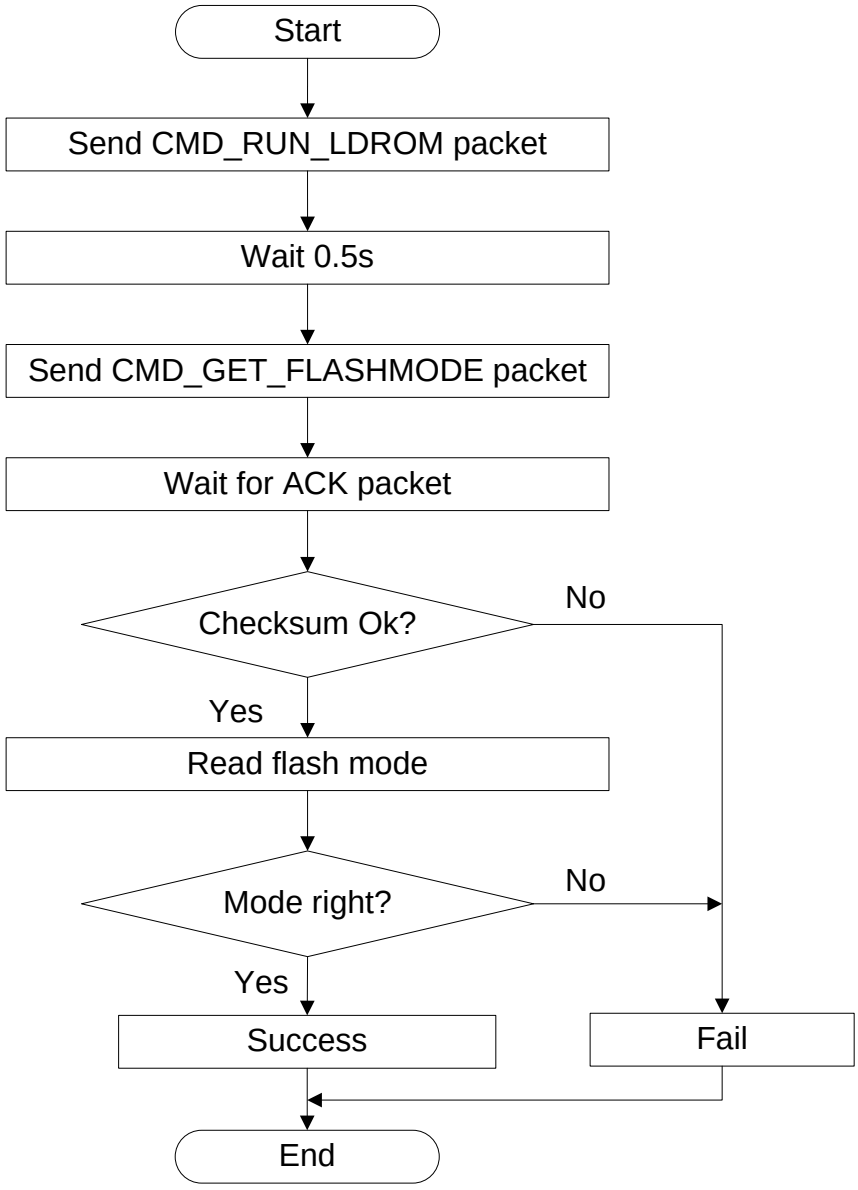


Figure 42. Run LDROM Command: Host Side

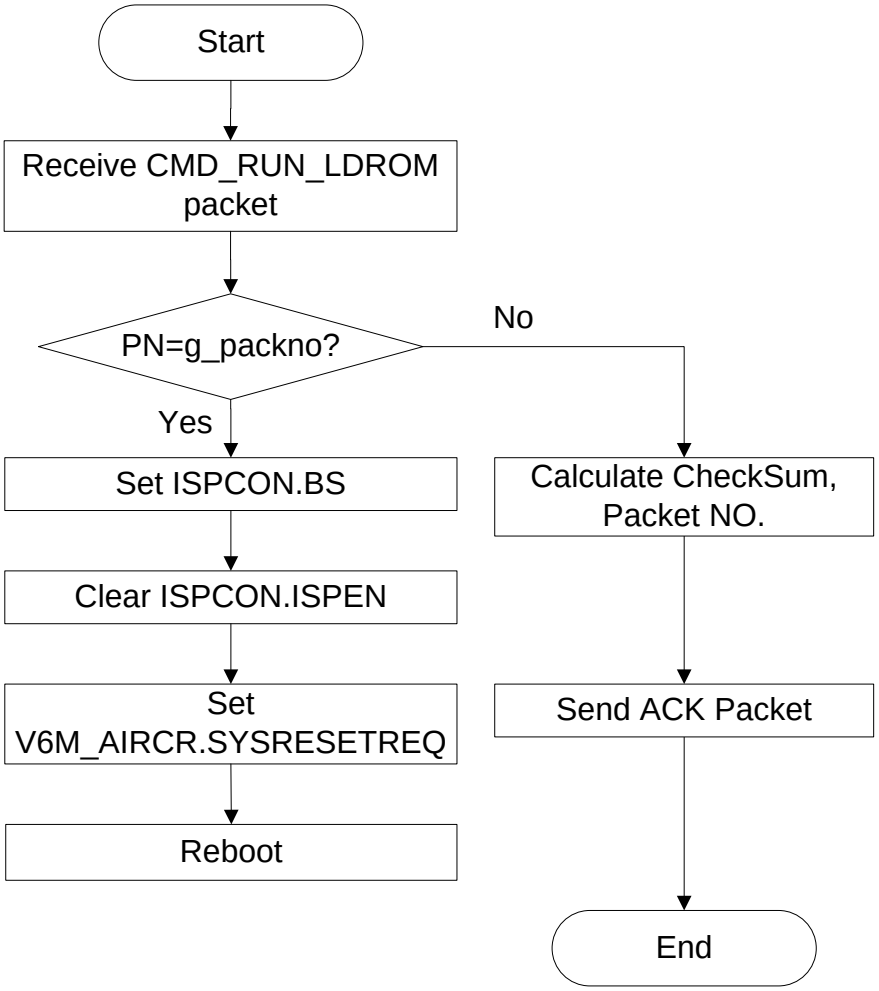


Figure 43. Run LDROM Command: Device Side

Note: PN is packet number host send, g_packno is ISP local packet number.

6.11 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 of NUC100 or M051 series. After this, the system will reboot.

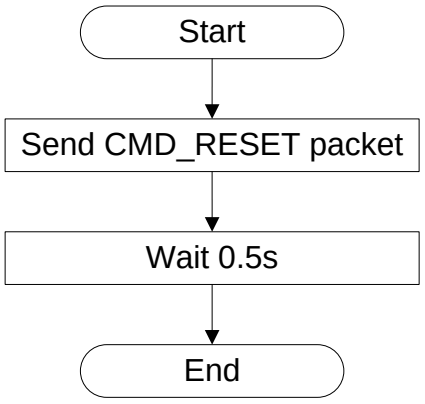


Figure 44. Reset Command: Host Side

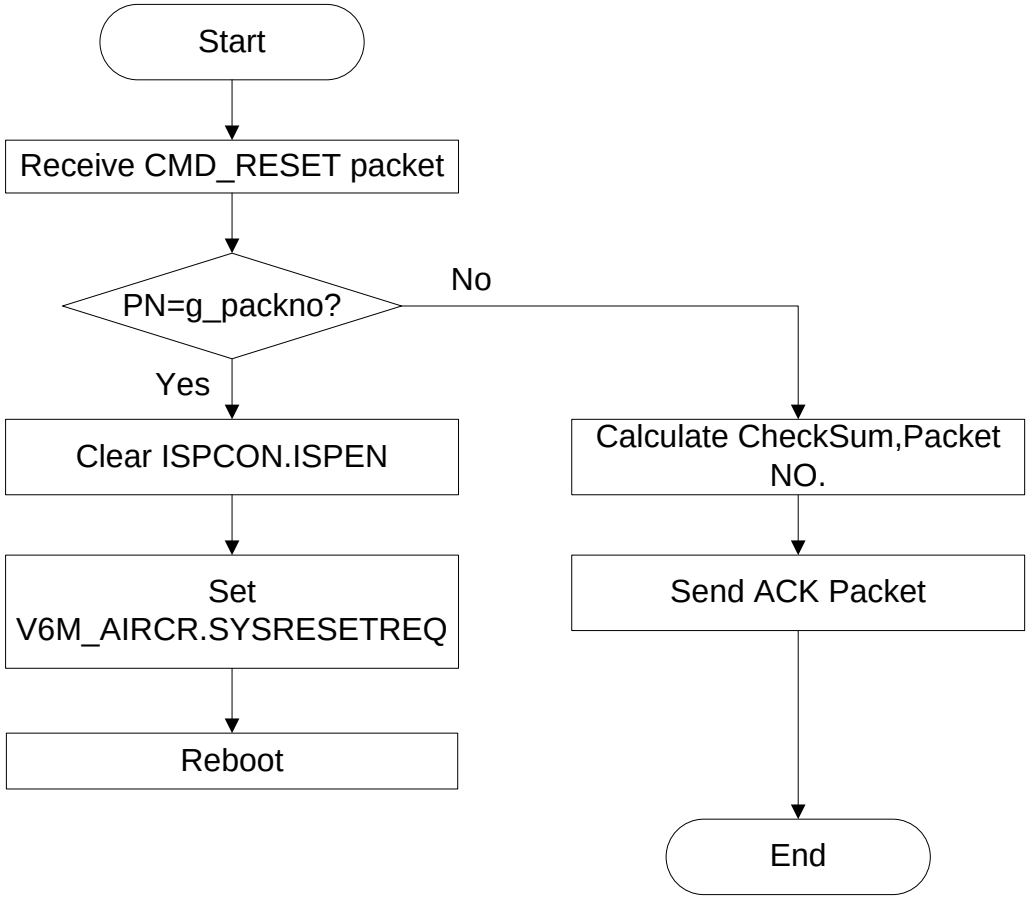


Figure 45. Reset Command: Device Side

Note: PN is packet number host send, g_packno is ISP local packet number.

6.12 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 search 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.

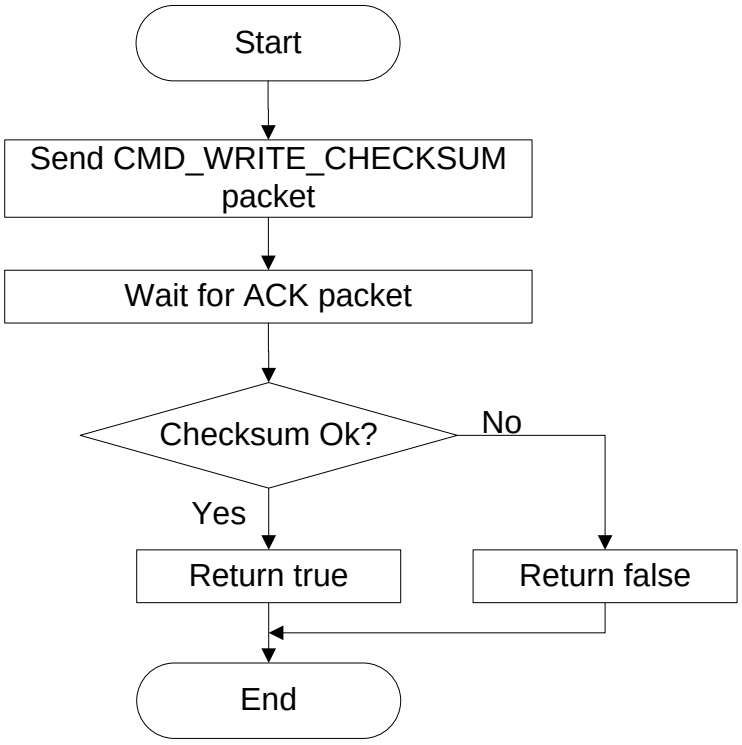


Figure 46. Write Checksum Command: Host Side

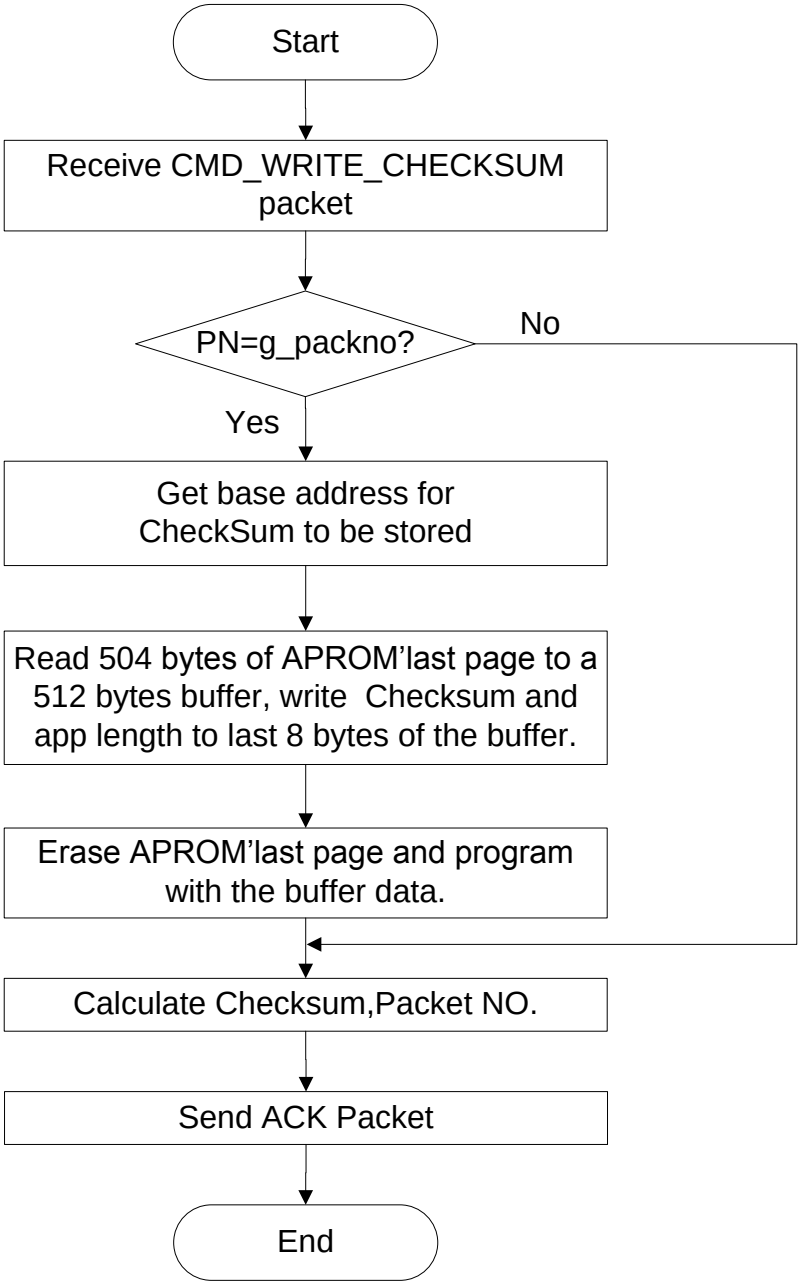


Figure 47. Write Checksum Command: Device Side

Note: PN is packet number host send, g_packno is ISP local packet number.

6.13 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. This command is mainly for I2C and SPI ISP. 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.

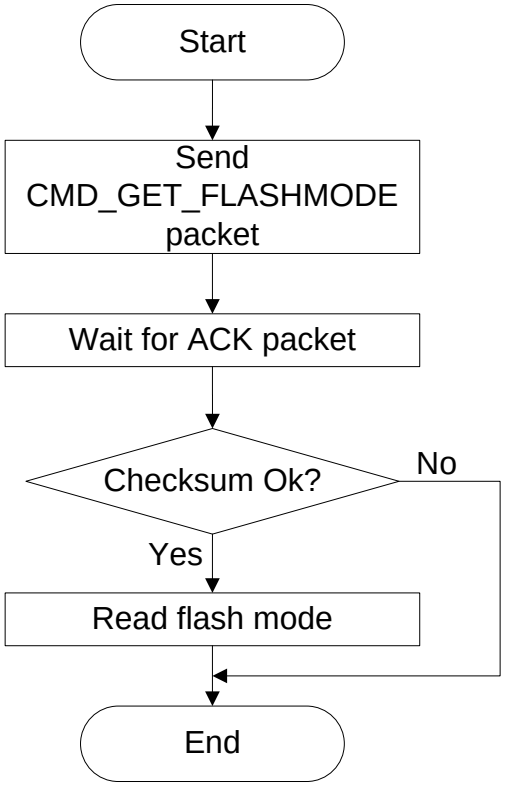


Figure 48. Get Flash Mode Command: Host Side

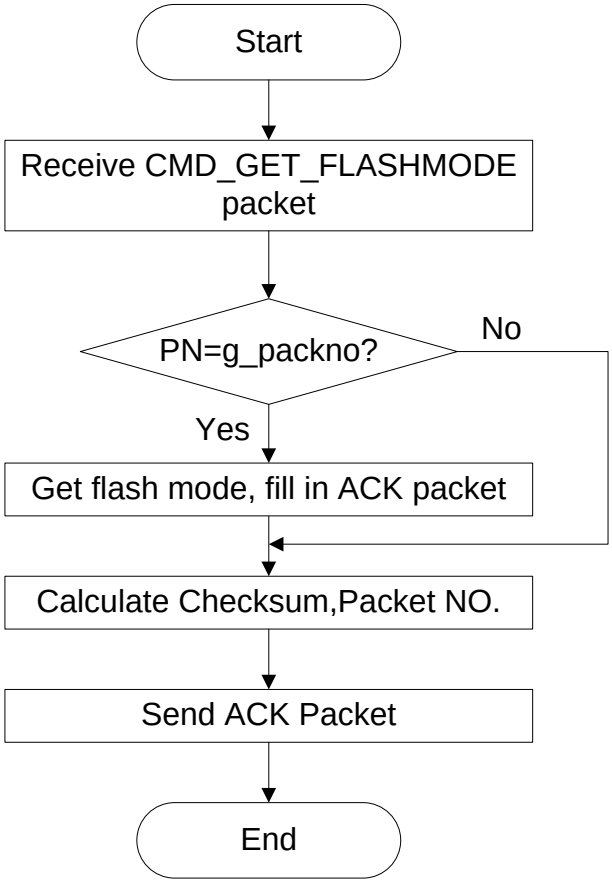


Figure 49. Get Flash Mode Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.14 Command CMD_RESEND_PACKET(0xFF):

Command Format

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4
Checksum	Packet Number

Description

The command is used for these commands of CMD_UPDATE_APROM, CMD_PROGRAM_WOERASE and CMD_PROGRAM_WERASE when error happen

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.

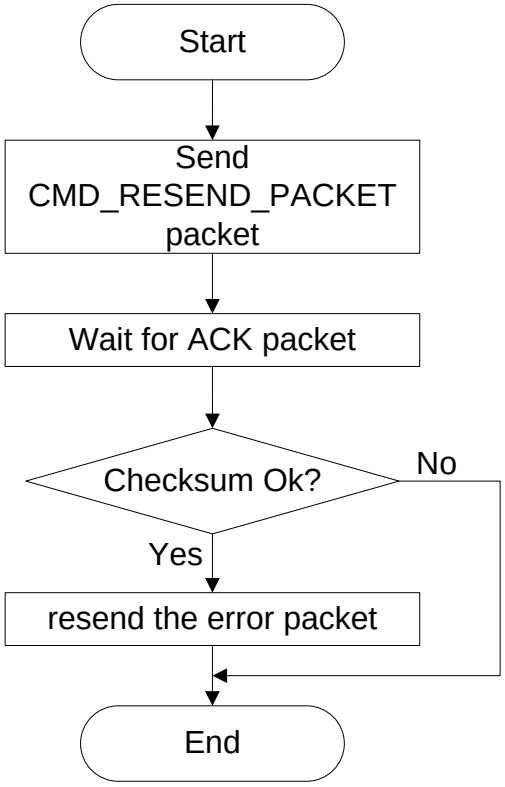


Figure 50. Resend Packet Command: Host Side

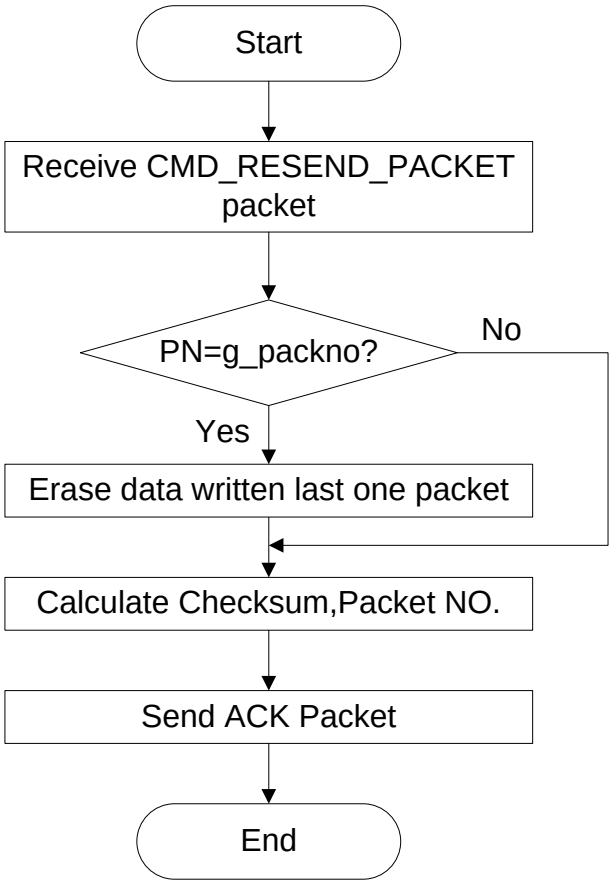


Figure 51. Resend Packet Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

6.15 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.

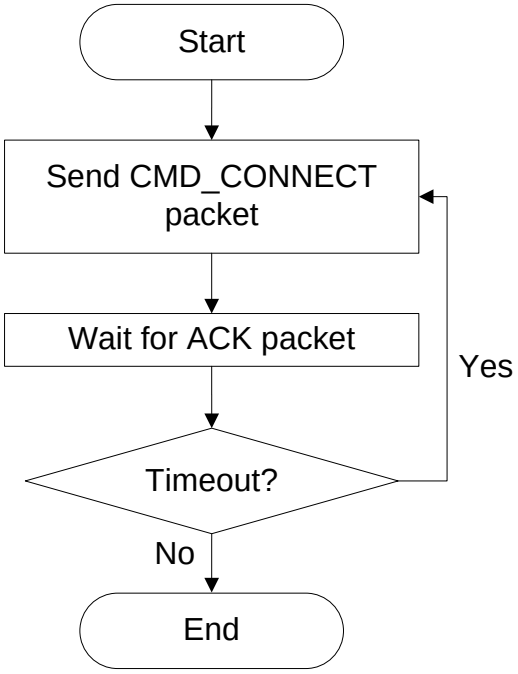


Figure 52. Resend Packet Command: Host Side

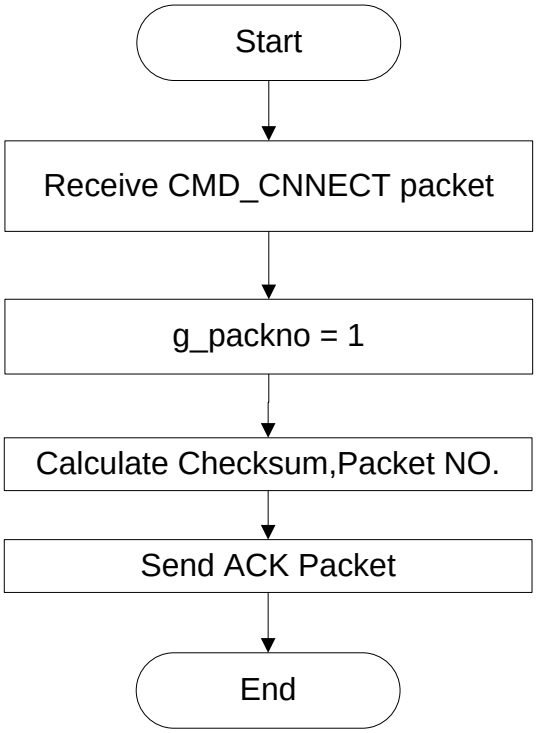


Figure 53. Resend Packet Command: Device Side

Note: PN is packet number host send; g_packno is ISP local packet number.

7. Commands flow

This flow is for command only. If user need update APROM, the following commands need issue

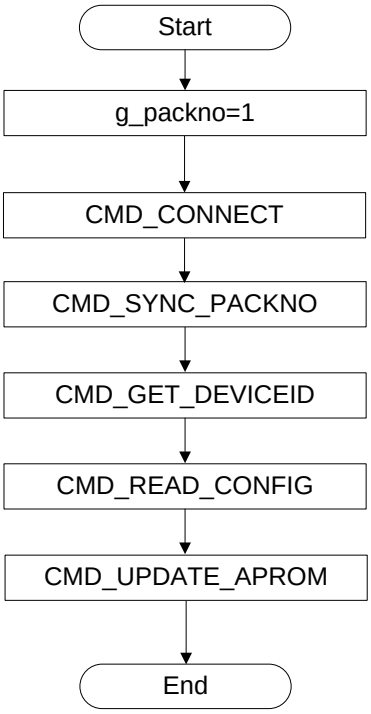


Figure 54. update APROM command flow

8. Revision History

VERSION	DATE	DESCRIPTION
V1.0	Aug 13, 2010	Preliminary version initial issued
V1.2	Sep 7, 2010	Modify all flow and overview
V1.3	Oct 8, 2010	Add I2C bus signal and SPI/I2C depict
V1.4	Mar 3,2011	Remove some command and Modify CMD_UPDATE_APROM flow
V1.5	May.16.2012	Add Pin definition for each interface