# NuMicro™ ISP Flow And Command Set

## 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 is made and released to end customer. Nuvoton provides a ISP (In-System Programming) method to update the flash code of the NuMicro™ 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, SPI, RS485, CAN 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 document describes the ISP code flow 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/

# Table of Contents:

# List of Figures:

# 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 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), we mainly take USB, UART as an example in this document. 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 https://gitee.com/OpenNuvoton/Nuvoton_Tools 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, "*NuvoISP.exe"* is provided to demonstrate ISP function through USB and UART interfaces.

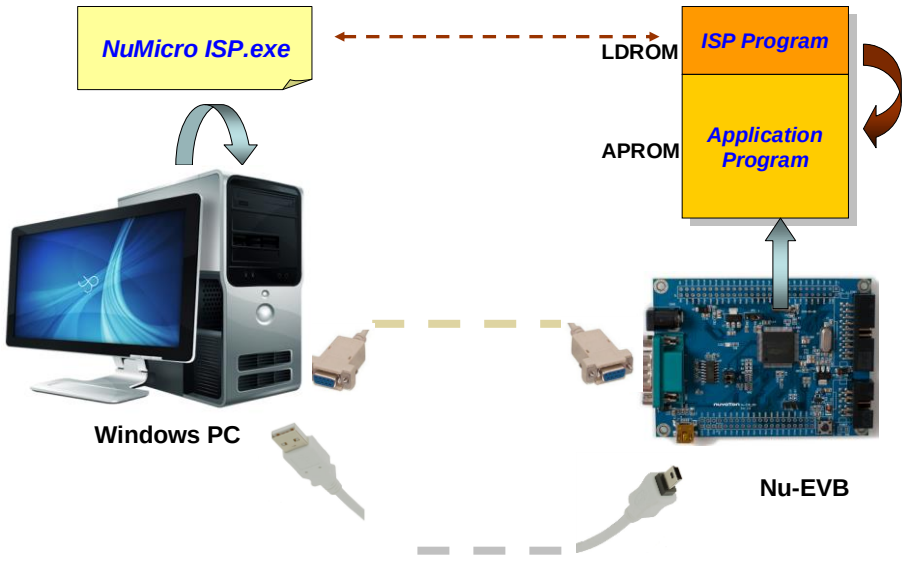


**Figure 1. ISP Through USB and UART**

## 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.

| 0 | | 63 |
|---|---|---|
| Command | Packet number | …... |

**Figure 2. Packet sent**

| 0 | | 63 |
|---|---|---|
| Checksum | Packet number | …... |

**Figure 3. Packet Received**

# 2. 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 UART interface.

On the other side, a windows PC plays the master device. A PC ISP program, *NuvoISP.exe,* is used to send command and data to slave device. The *NuISP.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 USB GPIO control pin is low for USB interface, or NuvoISP.exe is running for UART interface before testing ISP function.

## 2.1 Project and Source Files

The Keil uVision IDE tool is used to compile the programs running on the slave side. The source files of ISP code can be found at directory "*BSP/SampleCode/ISP/*" of each BSP. BSP can be found at https://github.com/OpenNuvoton or https://gitee.com/OpenNuvoton. The following diagrams show the screen snapshot after open the projects.

The source files of master ISP program, *NuvoISP.exe*, can be found at https://github.com/OpenNuvoton/ISPTool, or formal release on official website ISP Programming Tool Vx.xx.

## 2.2 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.

## 2.3 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 introduced later.

### 2.3.1 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 USB GPIO control pin is LOW for USB. Or receive packet from NuvoISP.exe for UART
- System starts according to a *SYSRESETREQ* event.

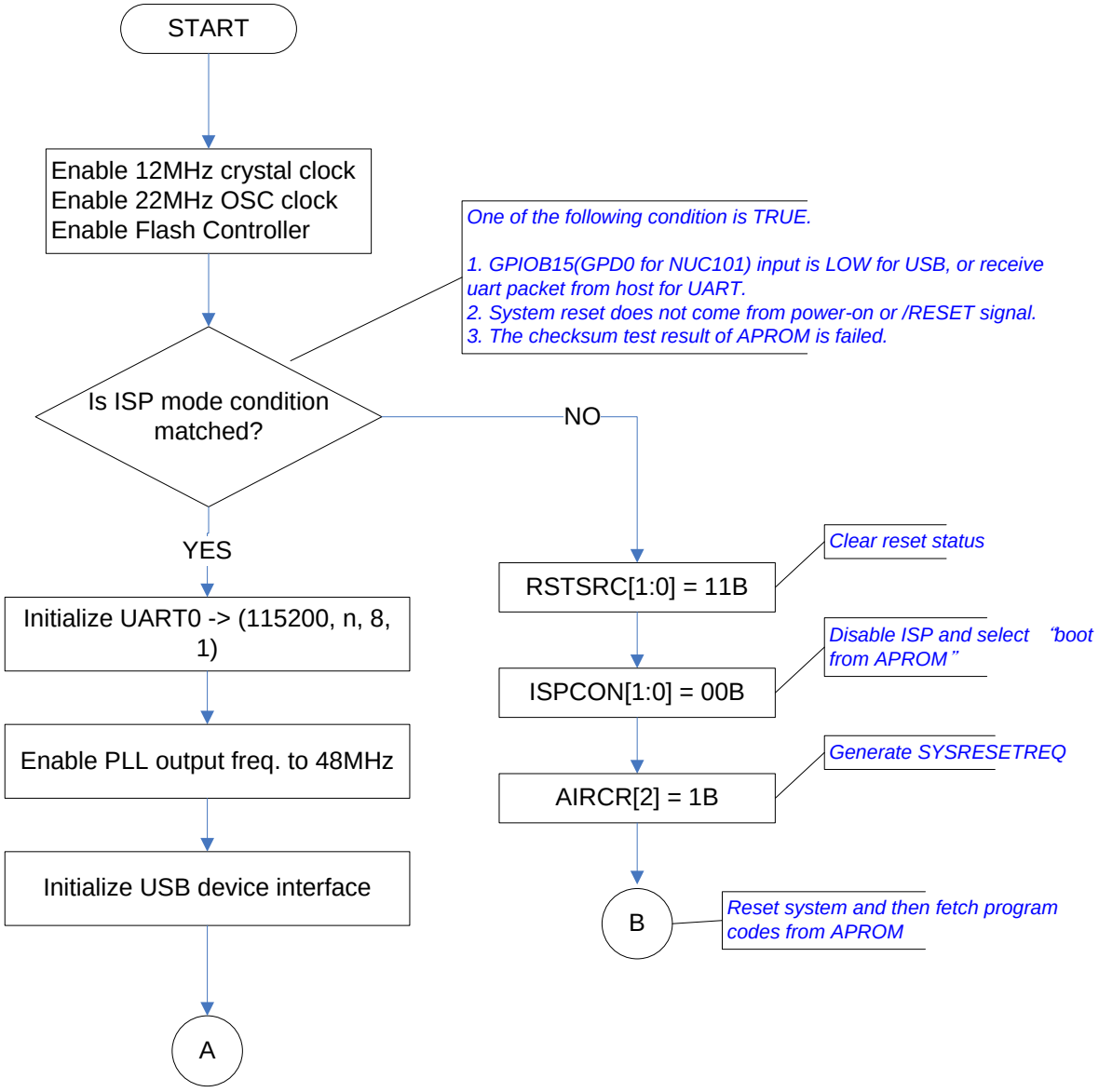


**Figure 4. USB/UART ISP Booting Flow**

### 2.3.2 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 UART 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.

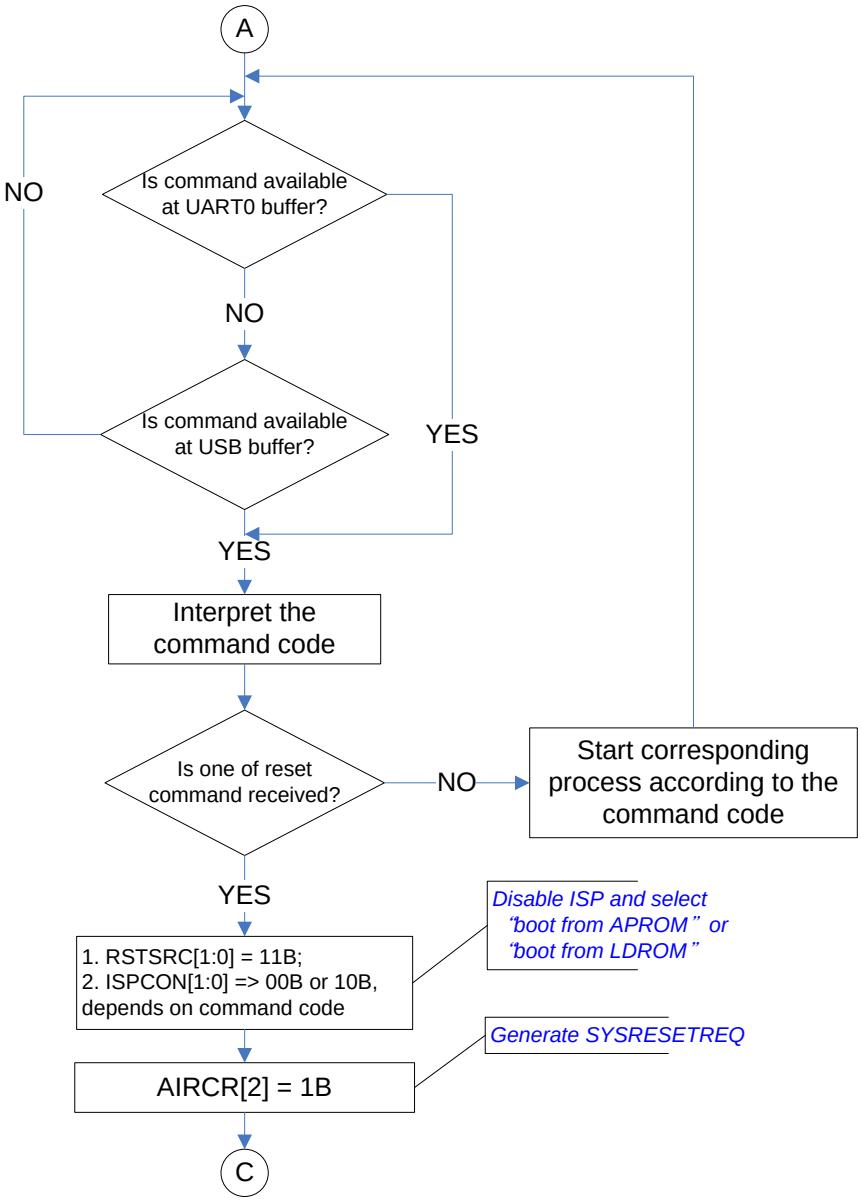


**Figure 5. USB and UART ISP Main Flow**

# 3. ISP Command Set

ISP packet is **64** byte fixed no matter command packet or ack packet. The ISP supported commands are listed as Table1 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 “NuvoISP.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.

### 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.

## 3.1 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.

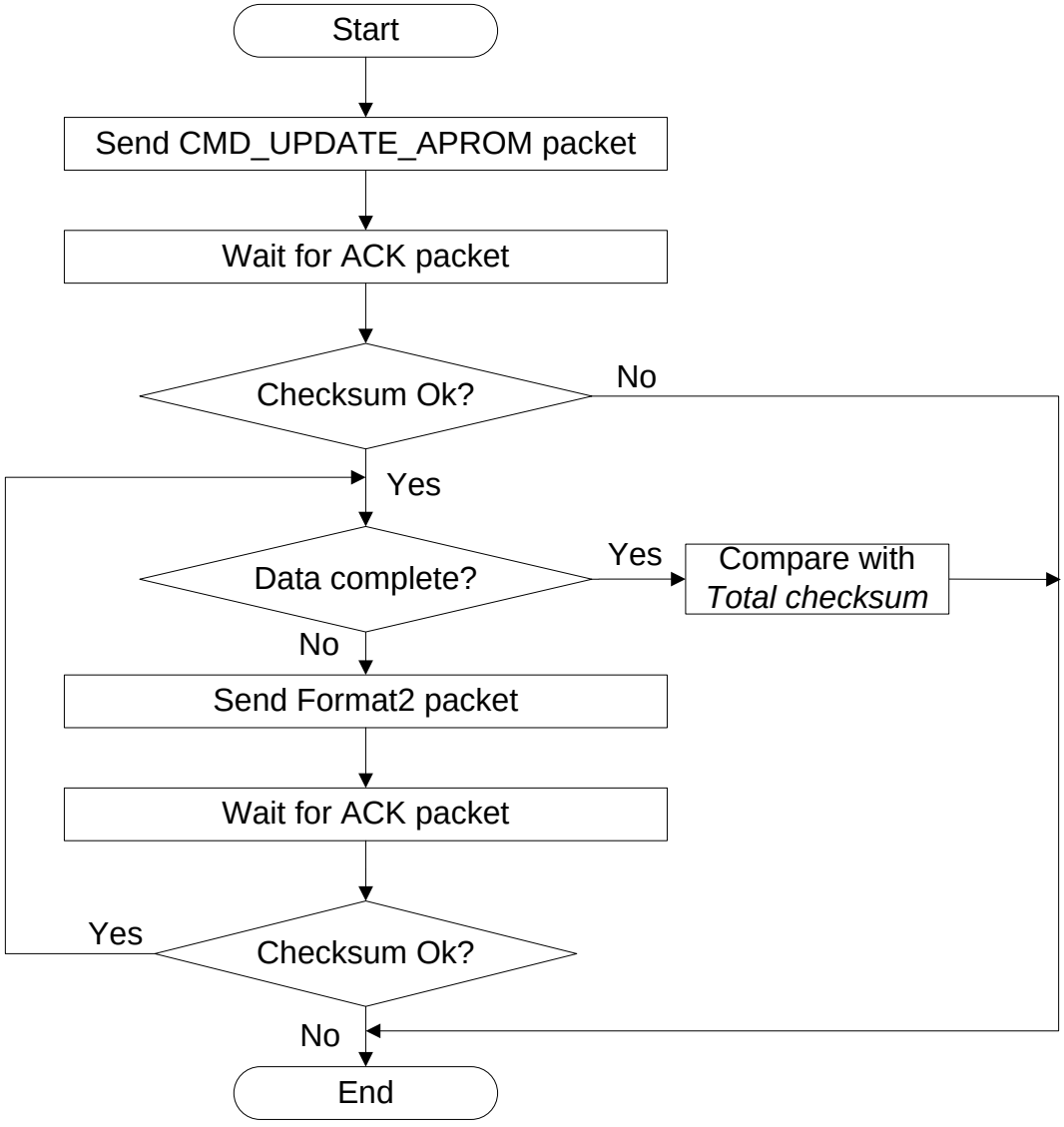


**Figure 6. Program APROM Command: Host Side**

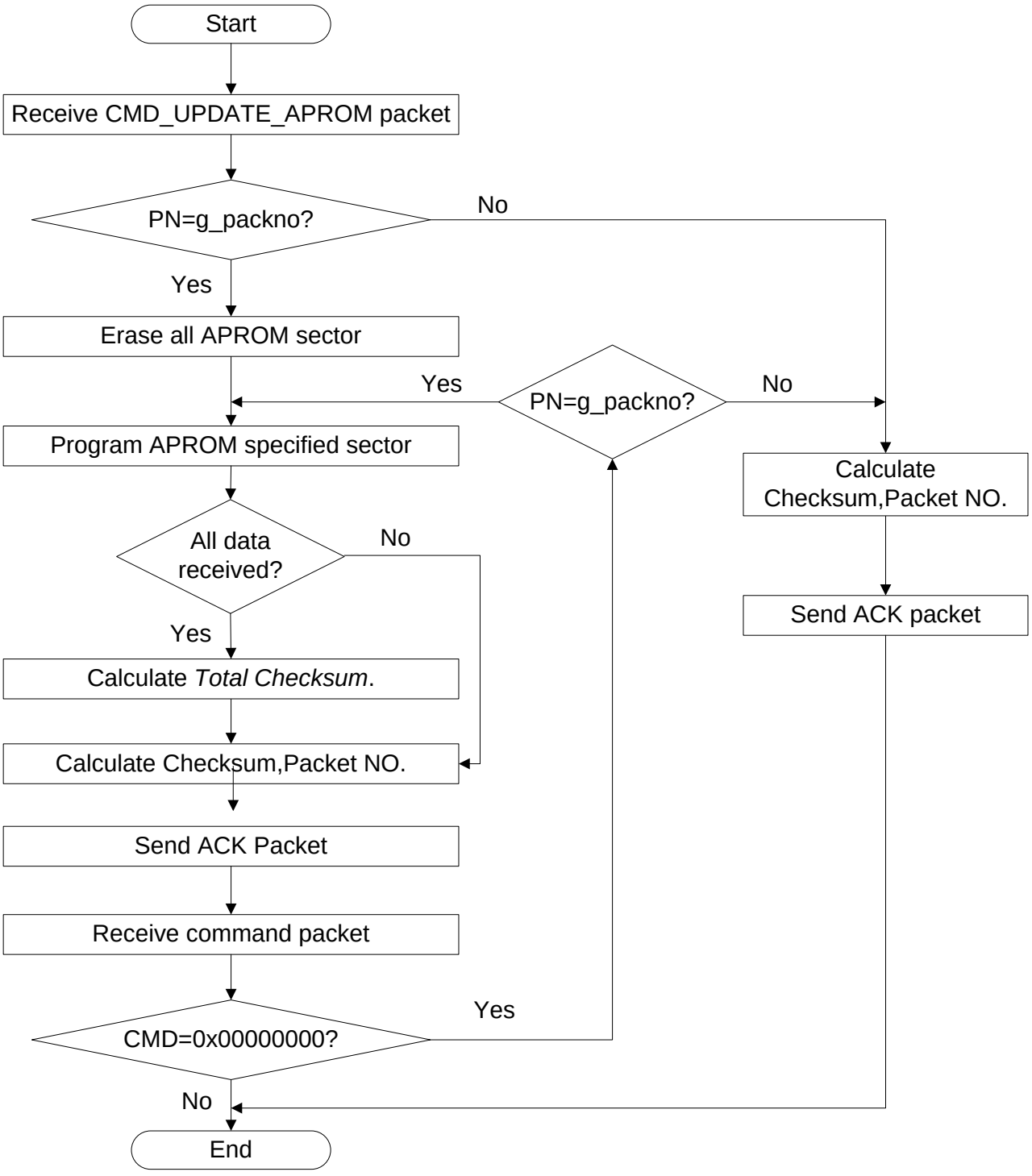


**Figure 7. Program APROM Command: Device Side**

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

## 3.2 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.

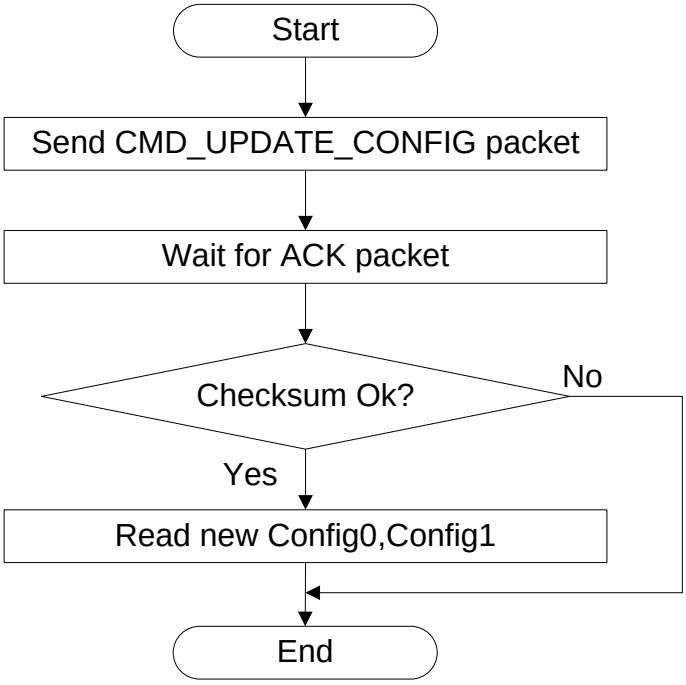


**Figure 8. Update *Config0* and *Config1* Command: Host Side**

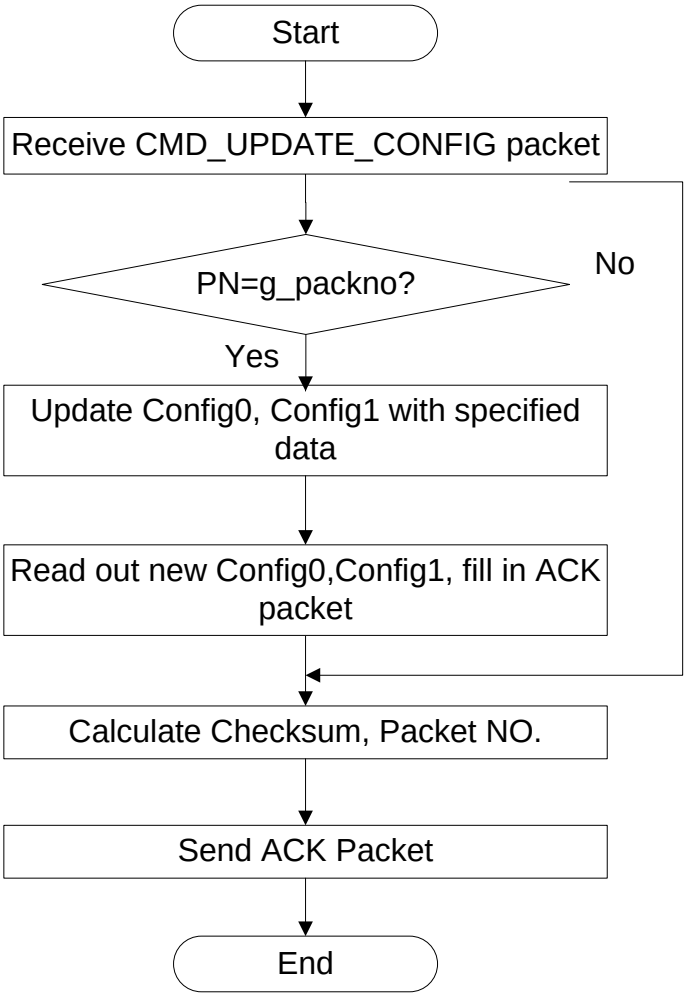


**Figure 9. Update *Config0* and *Config1* Command: Device Side**

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

## 3.3 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.

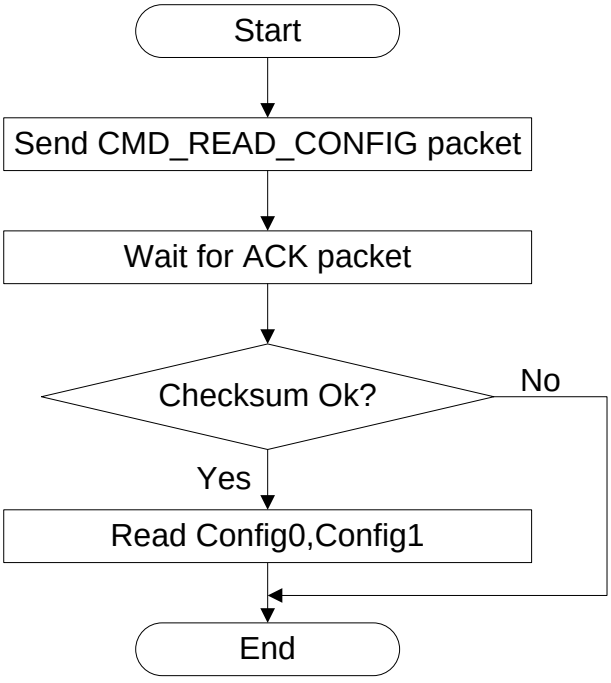


**Figure 10. Read *Config0* and *Config1* Command: Host Side**

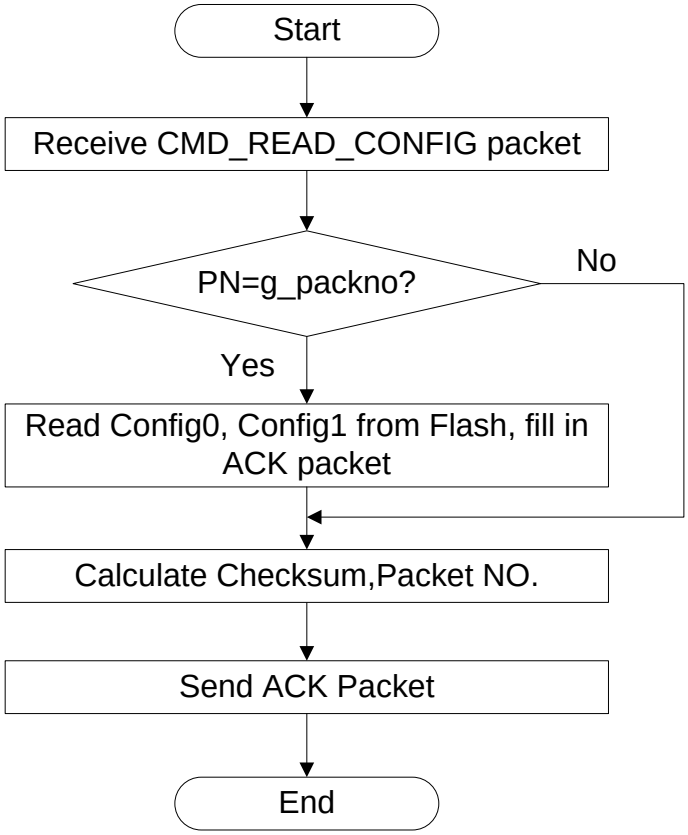


**Figure 11. Read *Config0* and *Config1* Command: Device Side**

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

## 3.4 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.

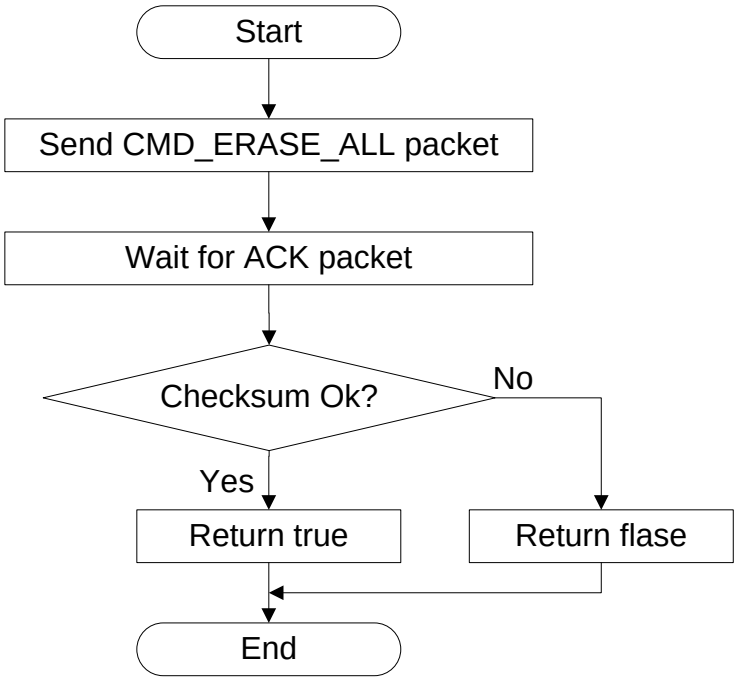


**Figure 12. Erase Flash Memory Command: Host Side**

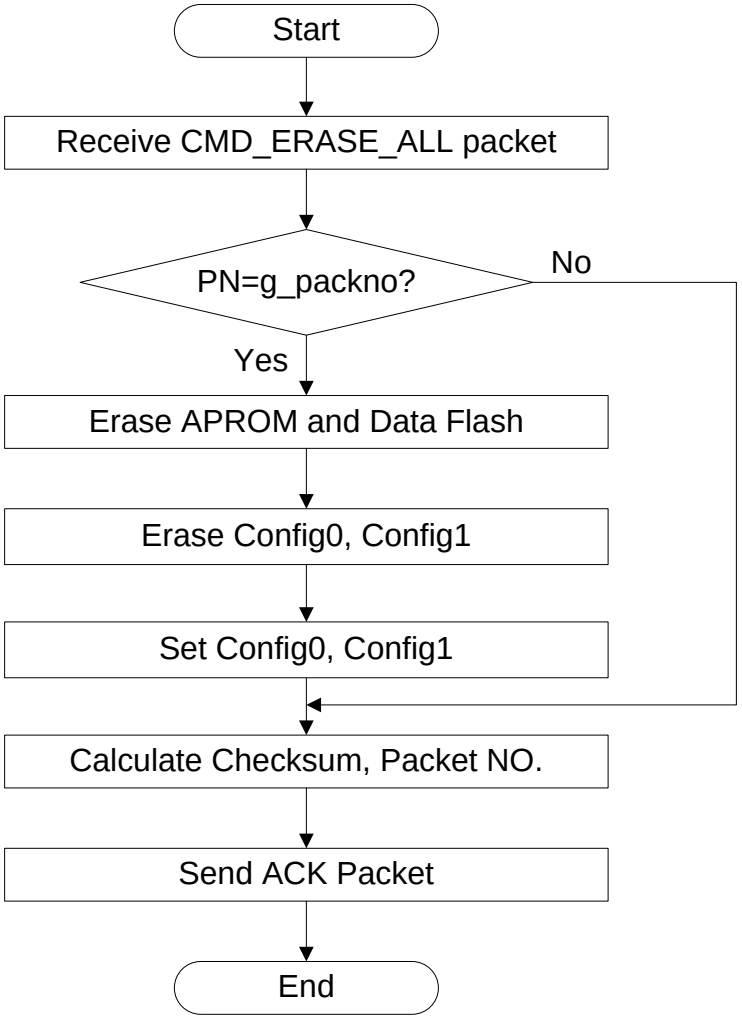


**Figure 13. Erase Flash Memory Command: Device Side**

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

## 3.5 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.

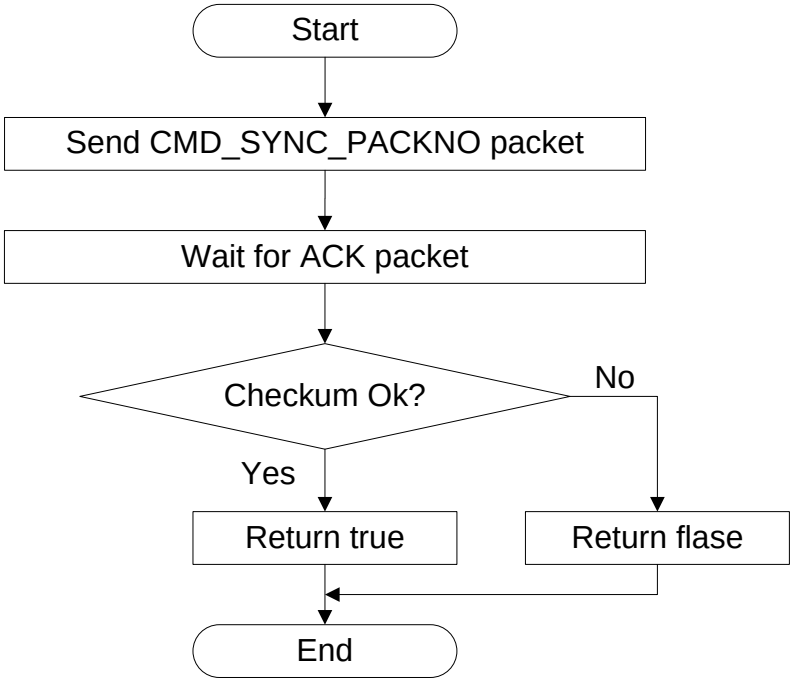


**Figure 14. Sync Packet Number Command: Host Side**

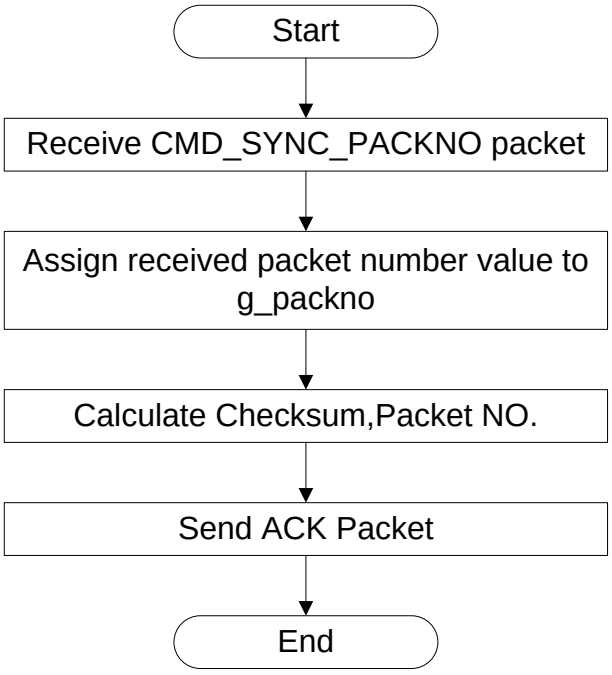


**Figure 15. Sync Packet Number Command: Device Side**

## 3.6 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.

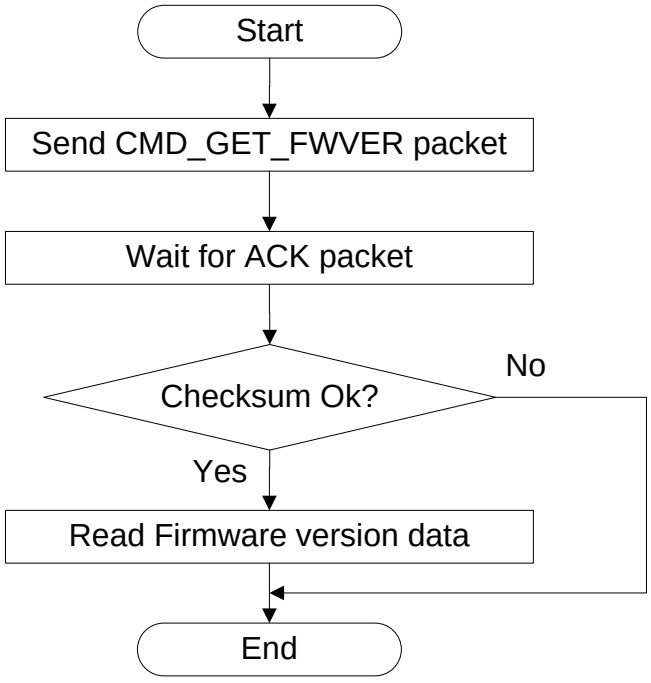


**Figure 16. Get ISP Version Command: Host Side**

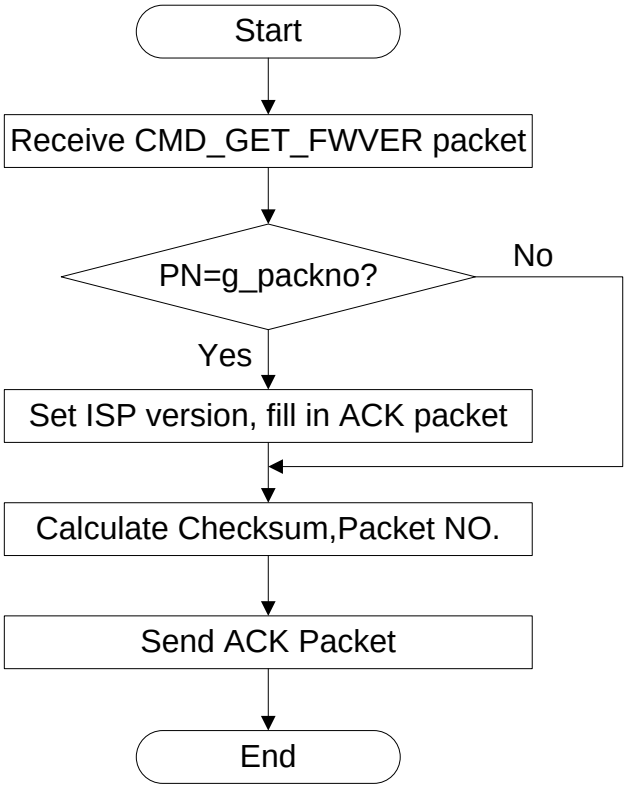


**Figure 17. Get ISP Version Command: Device Side**

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

## 3.7 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.

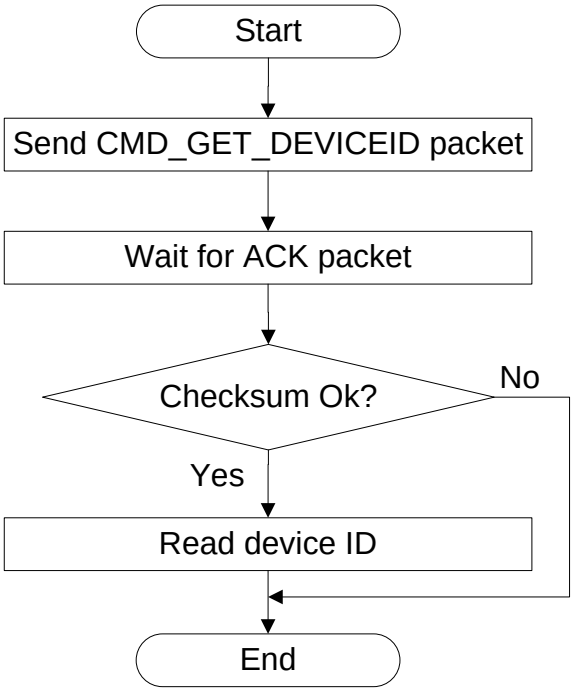


**Figure 18. Get product ID Command: Host Side**

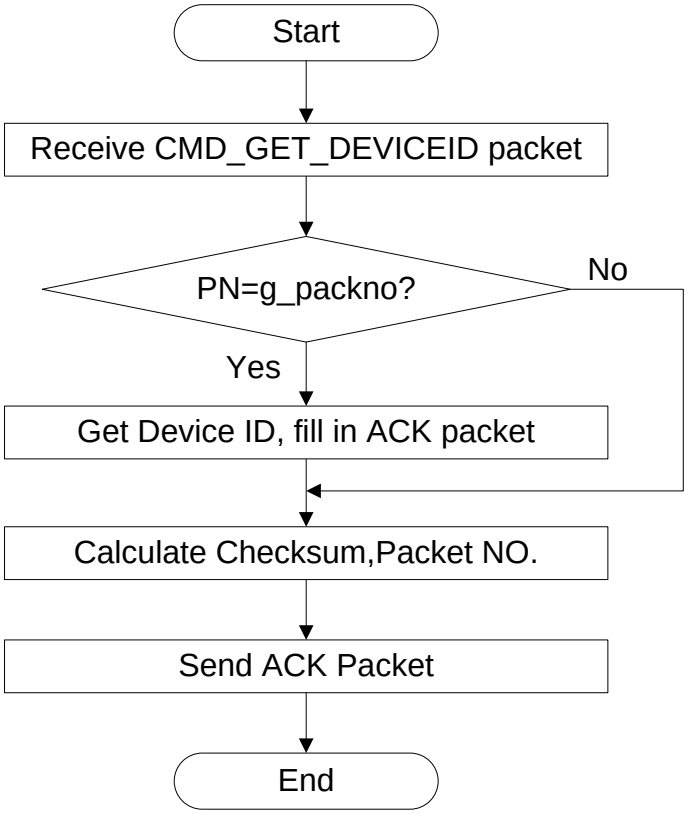


**Figure 19. Get product ID Command: Device Side**

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

## 3.8 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 *Start 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.

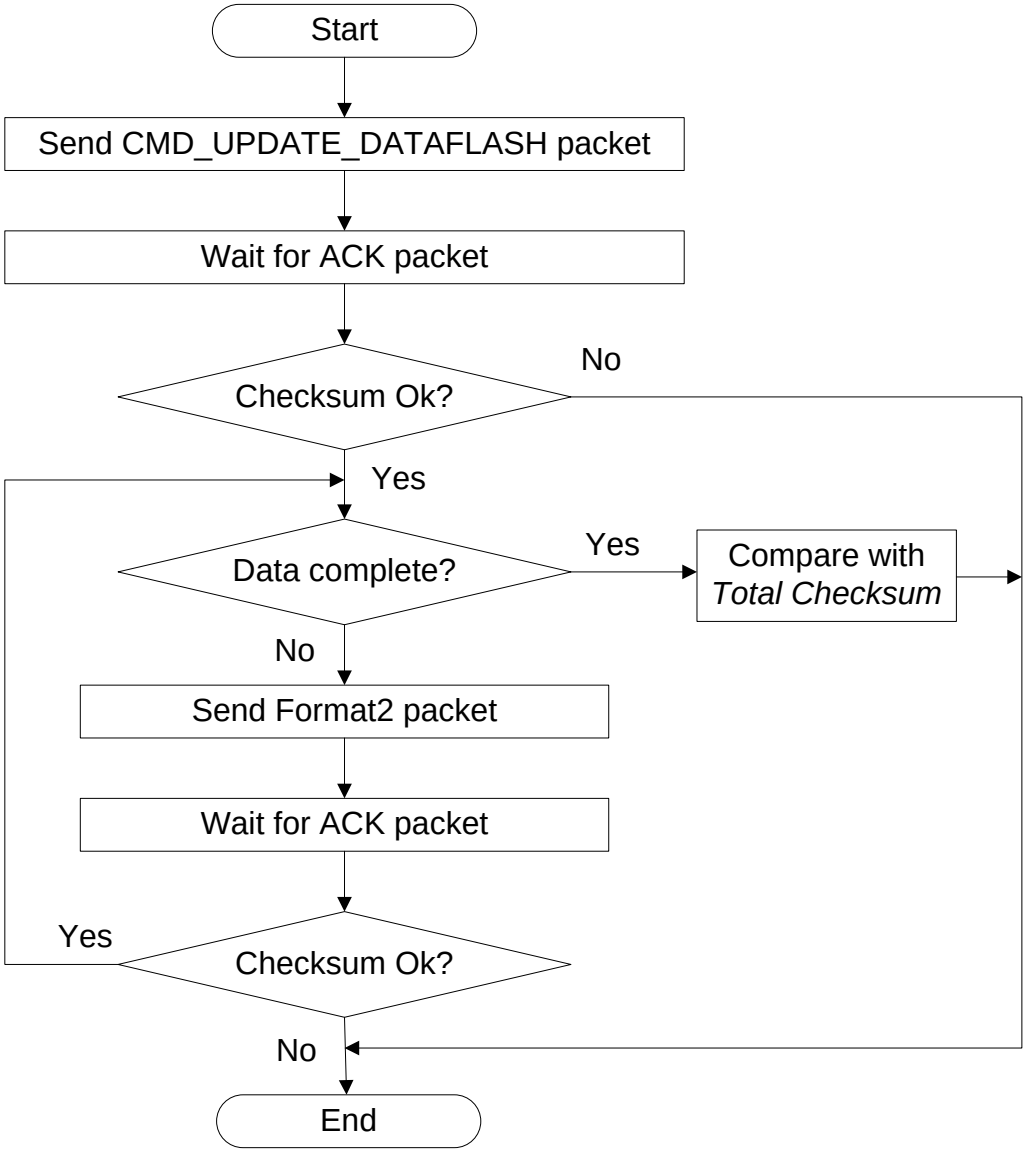


**Figure 20. Program Data Flash Command (2): Host Side**

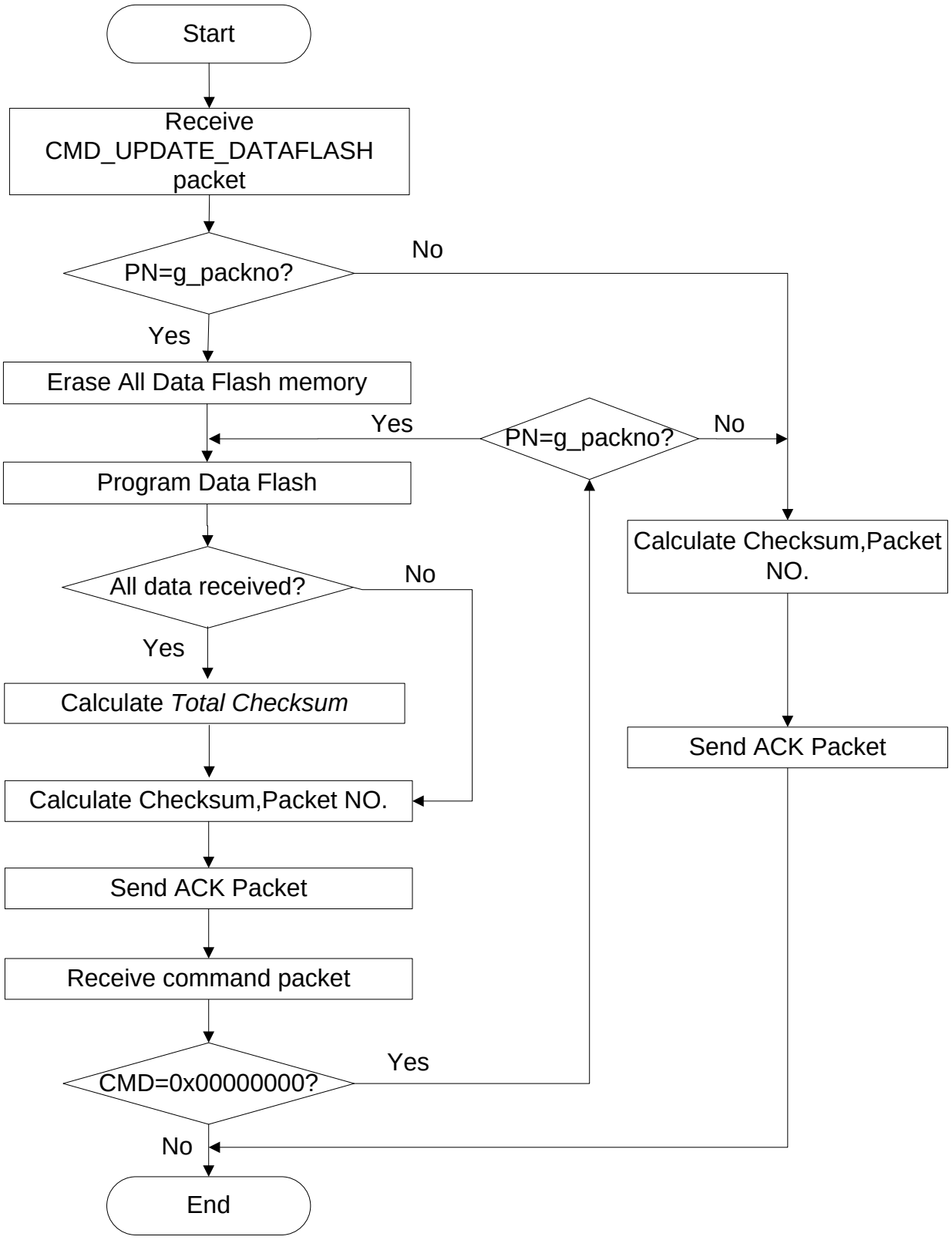


**Figure 21. Program Data Flash Command (2): Device Side**

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

## 3.9 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.

.

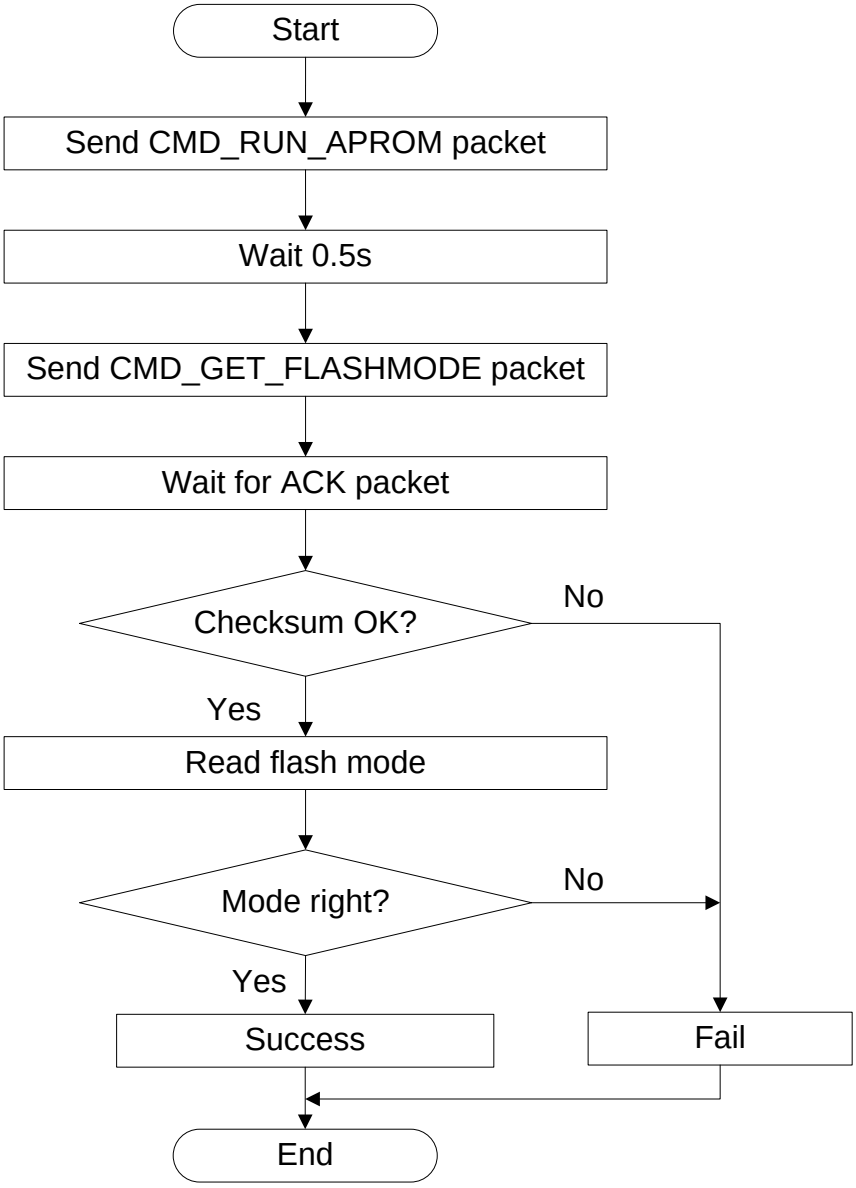


**Figure 22. Run APROM Command: Host Side**

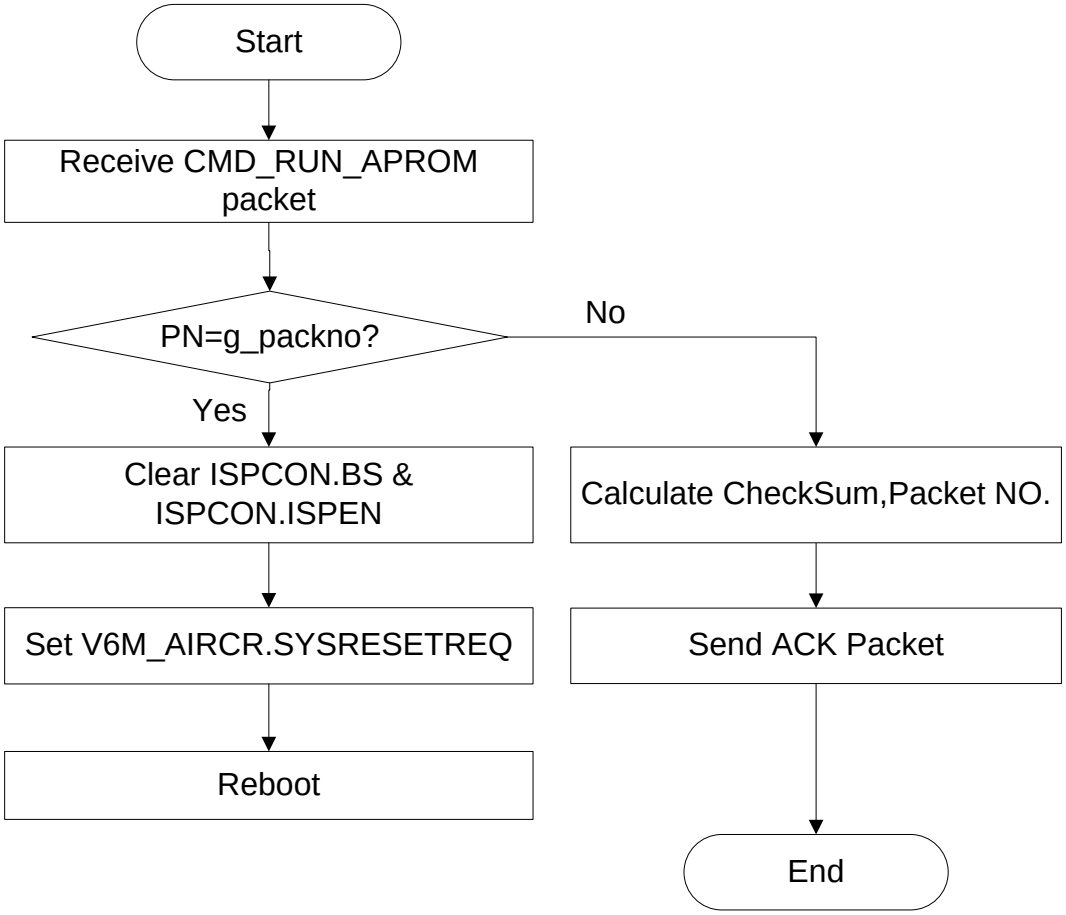


**Figure 23. Run APROM Command: Device Side**

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

## 3.10 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.

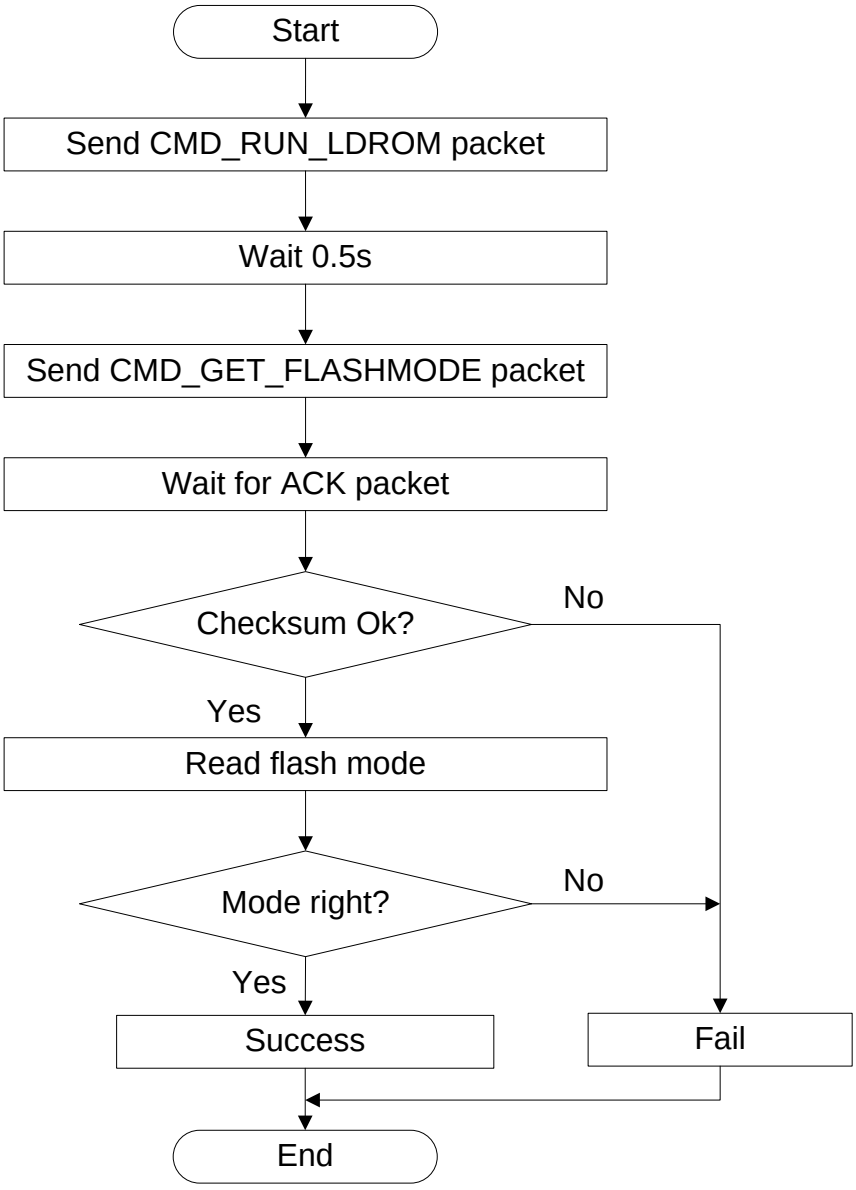


**Figure 24. Run LDROM Command: Host Side**

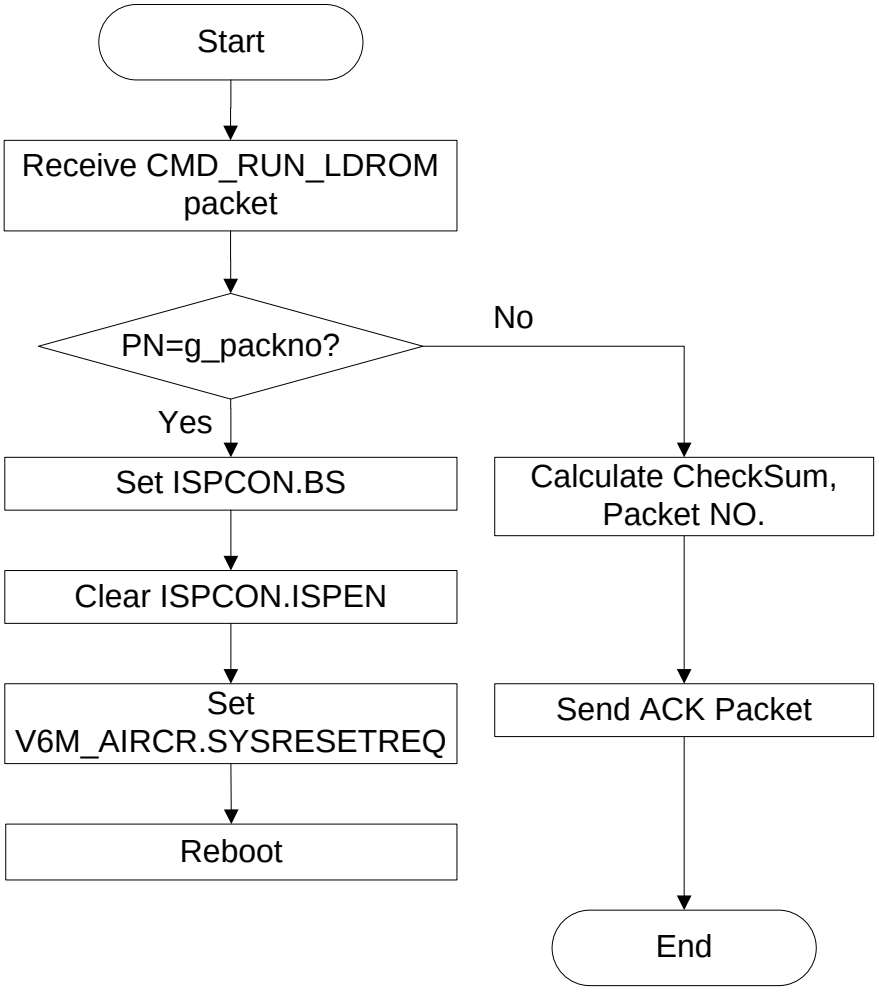


**Figure 25. Run LDROM Command: Device Side**

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

## 3.11 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.

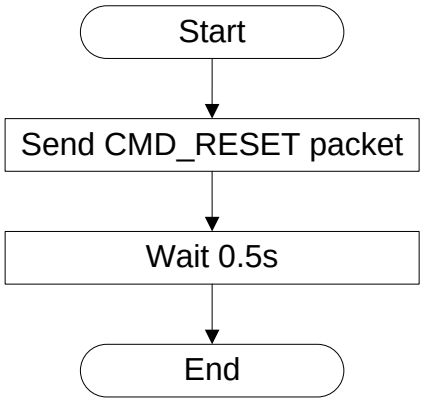


**Figure 26. Reset Command: Host Side**

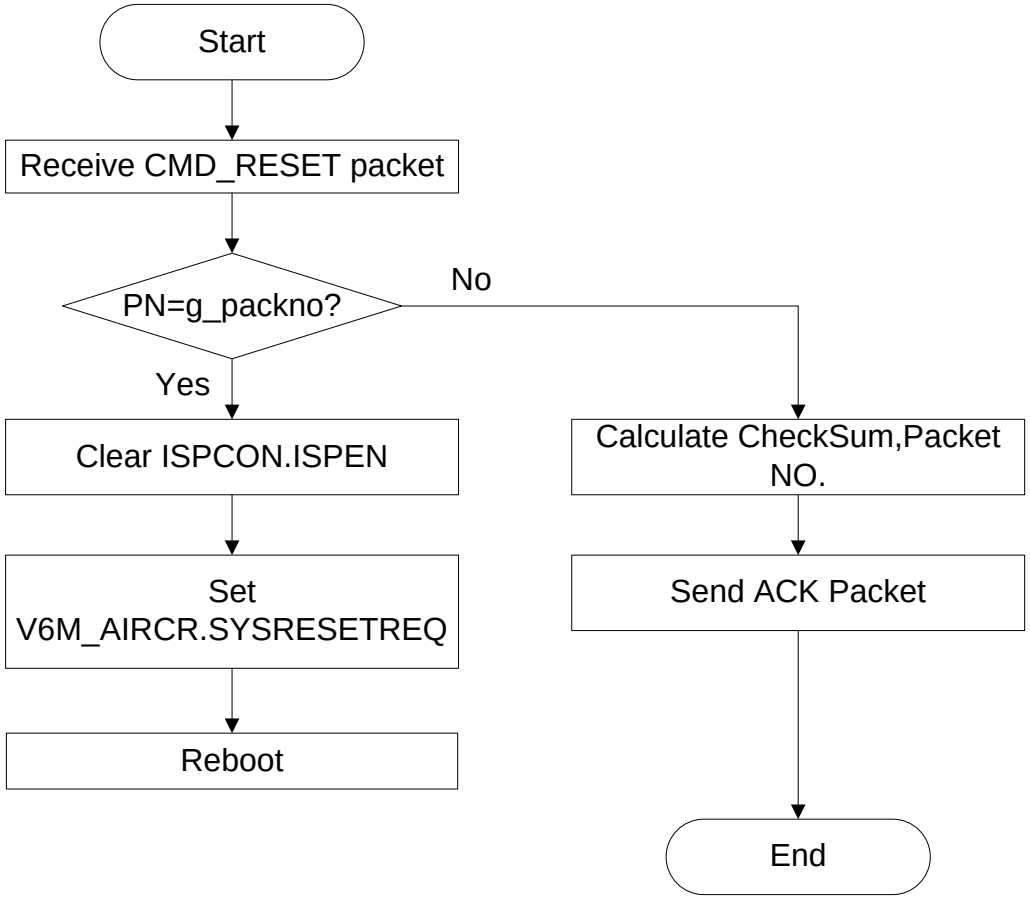


**Figure 27. Reset Command: Device Side**

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

## 3.12 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.

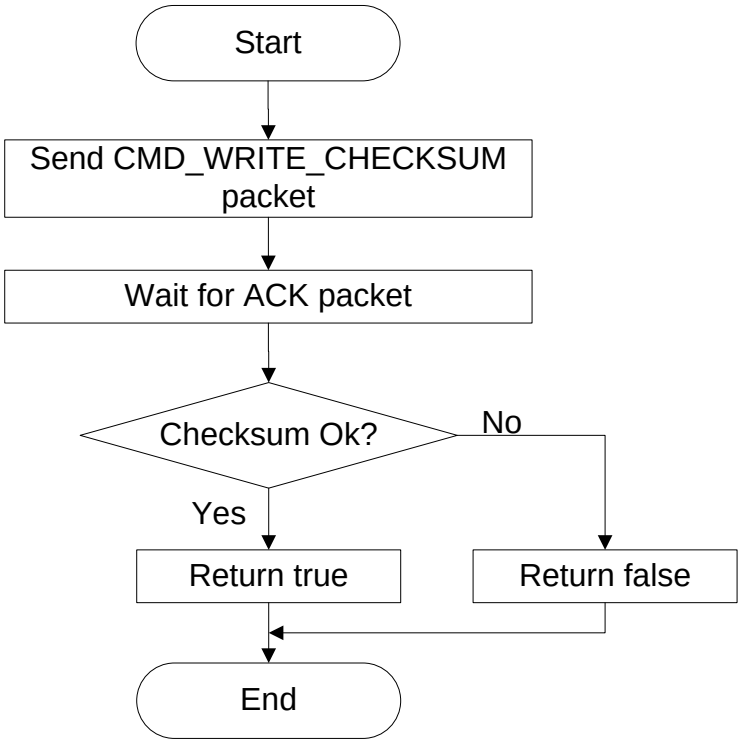


**Figure 28. Write Checksum Command: Host Side**

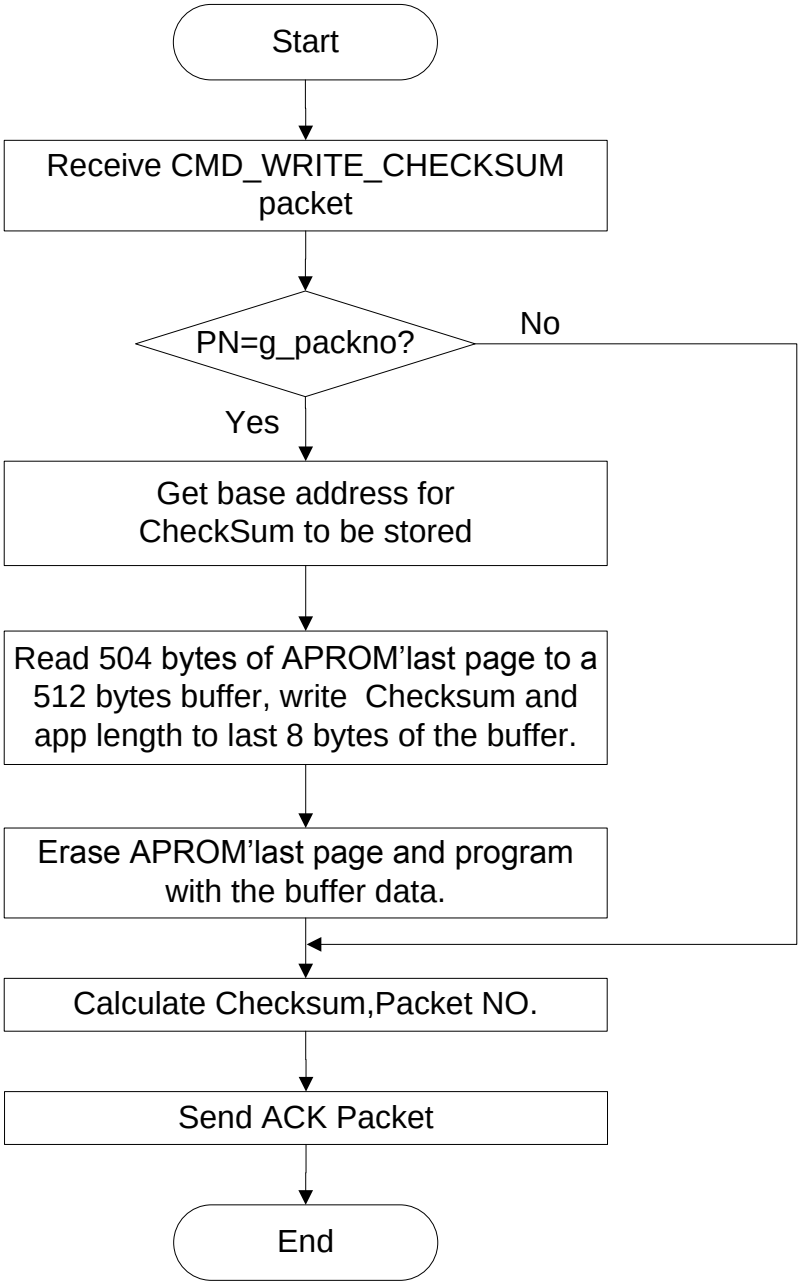


**Figure 29. Write Checksum Command: Device Side**

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

## 3.13 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.

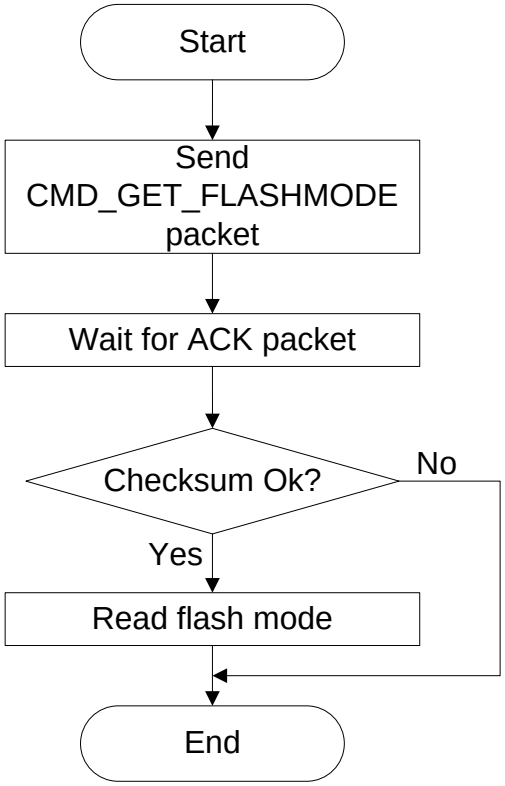


**Figure 30. Get Flash Mode Command: Host Side**

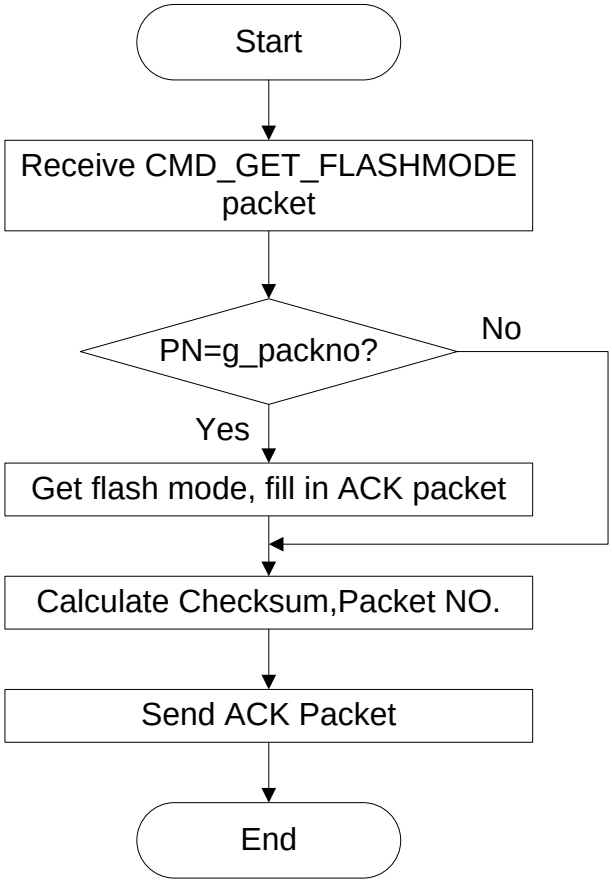


**Figure 31. Get Flash Mode Command: Device Side**

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

## 3.14 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.

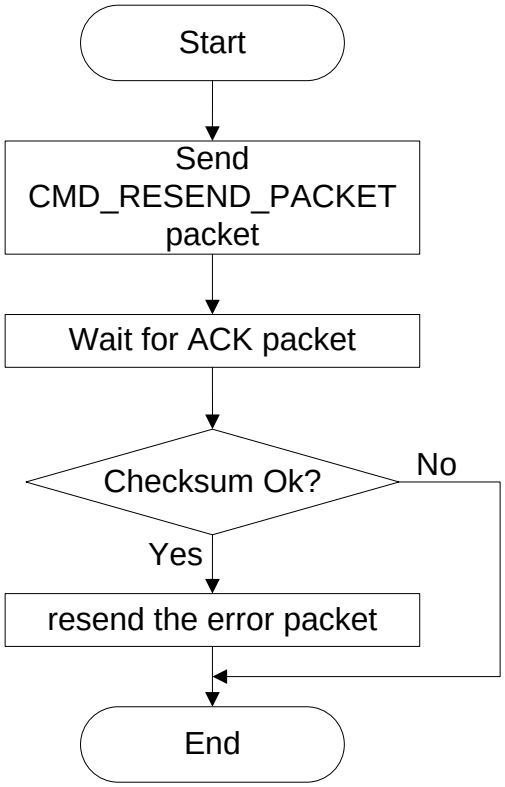


**Figure 32. Resend Packet Command: Host Side**

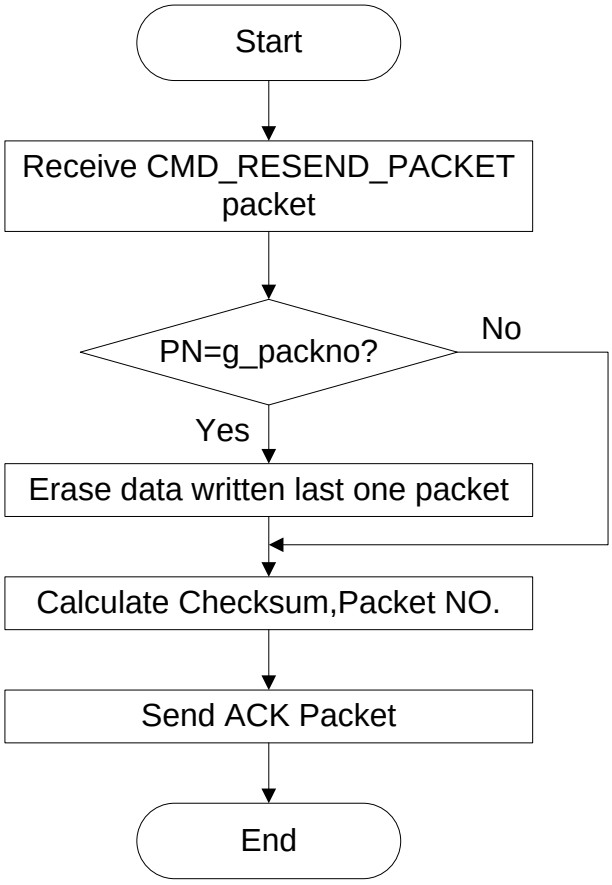


**Figure 33. Resend Packet Command: Device Side**

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

## 3.15 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. E.g. 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.

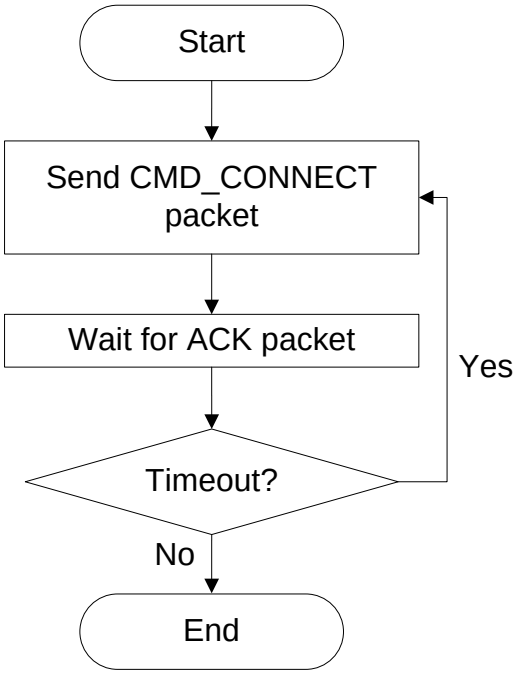


**Figure 34. Resend Packet Command: Host Side**

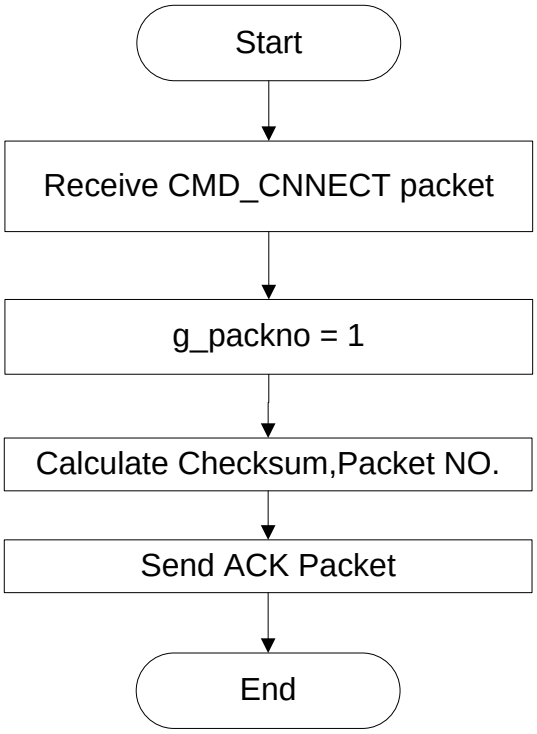


**Figure 35. Resend Packet Command: Device Side**

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

# 4. Commands flow

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

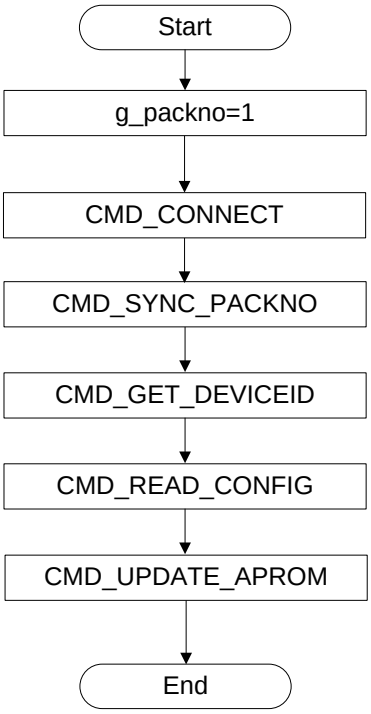


**Figure 36. update APROM command flow**

# 5. Revision History

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| V1.0 | Dec.31.2019 | Modified from ISP Application Note v1-5. Rename the document and remove some obsolete sections. |
| | | |