Application Note

Two-level BOOT scheme based on CAN interface

Introduction

During the development of embedded products, users will develop their own two-level BOOT code (indicated by IAP later) to upgrade the application program (indicated by APP later), and there are many peripheral interfaces for upgrading, such as I2C, USART, CAN, SPI, ETH, USB, etc.

This document is mainly aimed at the above-mentioned application scenarios of Nsing Technology MCU series products, and guides users how to use IAP examples to realize the function of IAP upgrade APP through CAN peripheral interface.

This document is only suitable for Nsing Technology MCU products with CAN peripheral interface. The currently supported product series are N32G43x series, N32L43x series, and N32L40x series products.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Contents

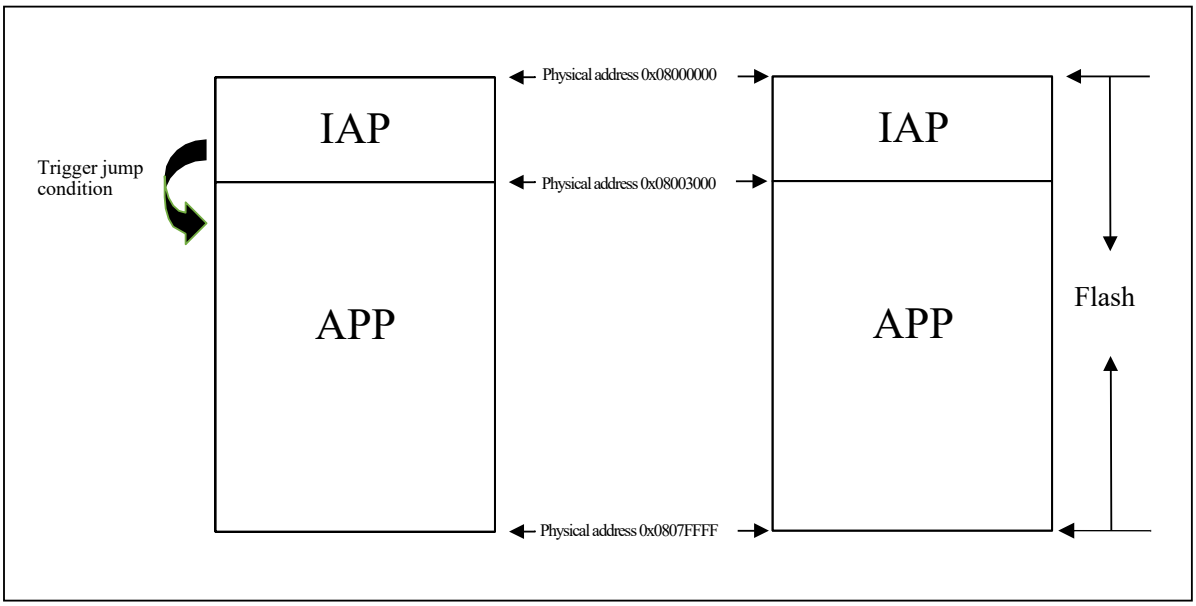
NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

1. Implementation Mechanism of APP Upgrade through IAP

When users develop some products, they need to upgrade the APP software code for the subsequent use of the products. In this case, IAP code should be pre-built in the Flash of MCU, and the APP should be downloaded and updated through the reserved peripheral interface.

IAP codes are generally stored in the first address segment of Flash in MCU, while APP codes are stored after IAP, as shown in Figure 1-1.

Figure 1-1 Store Addresses Of IAP and APP And Redirect Diagram



Chip operating mode is usually divided into two modes:

1. APP application mode: IAP code runs after the chip is powered on, and jumps to APP execution code after the jump condition is triggered.
2. IAP upgrade mode: after the chip is powered on, the IAP code is run, and the jump condition is not triggered, the IAP code is executed for APP upgrade

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

2. Function Description of IAP Routine Based On CAN Peripheral Interface

The corresponding Keil project can be opened in the SDK example directory (Nsing.n32g43x_Library.1.2.0 \projects\n32g43x_EVAL\Applications\IAP\CAN).

The functionality of IAP routine is divided into two main parts:

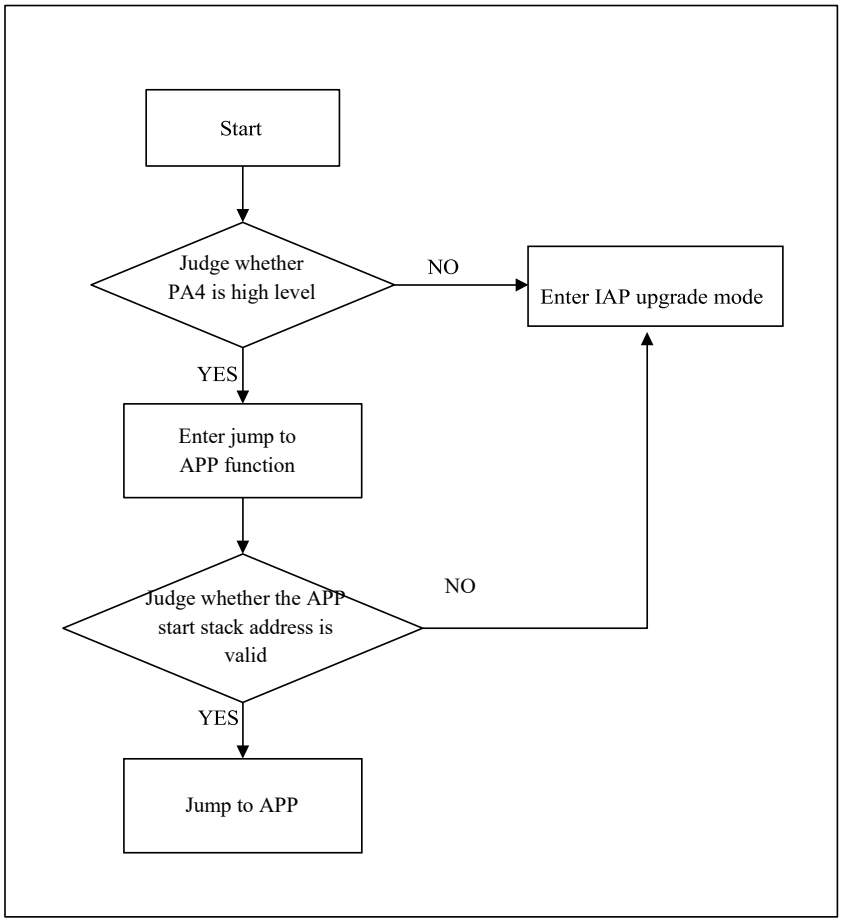
1. Judge whether the trigger condition of jumping to APP is effective, and control whether the program jumps to APP according to this;
2. Enter IAP upgrade mode, receive commandcommands from host computer (or other chips) through CAN peripheral interface and make corresponding response;

2.1 Trigger Condition Judgment

IAP routines are triggered by the level state of GPIO PA4 (this can be modified by connectting the PA4 of the N32G43x series minimum system development board N32G43XR-STB to high level or low level). Figure 2-1 below is the schematic diagram of program flow at different level states of PA4.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Figure 2-1 Schematic Diagram of the Program Flow in Different Levels of PA4



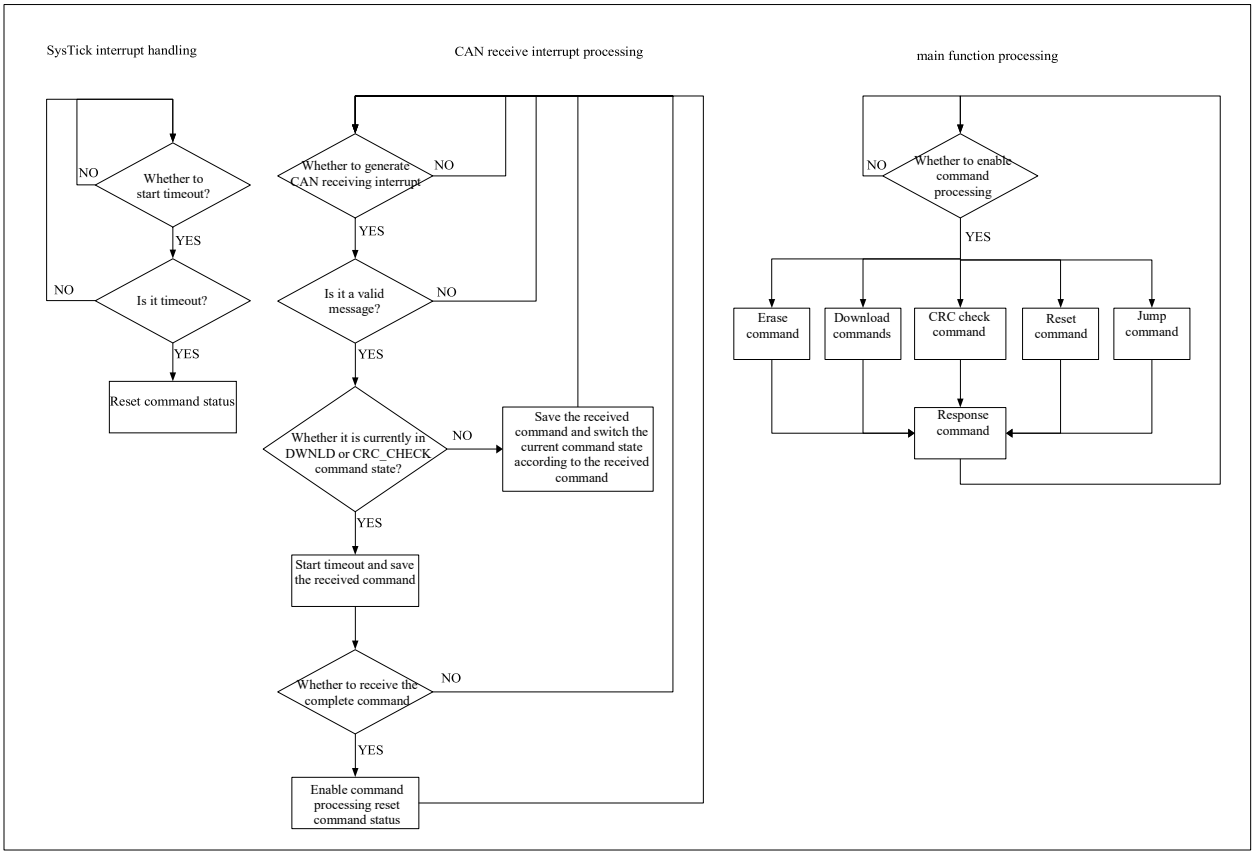
NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

2.2 IAP Upgrade Mode

After entering the IAP upgrade mode, the upgrade commandcommands and data are received through the CAN peripheral interface. For details of the upgrade commands, please refer to the upgrade commands in Chapter 3. Since one CAN message contains up to 8 bytes of valid data, some commands need to receive multiple CAN messages to form a complete command.

Figure 2-2 describes the upgrade command processing flow in IAP upgrade mode:

Figure 2-2 Schematic Diagram of Upgrade Command Processing Flow



NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

3. Upgrade Commands

All command data structures and meanings are described in detail below.

3.1 Commands and Data Structures

3.1.1 Command List

Table 3-1 Command List

Command Name	Level 1 Command Field	Level 2 Command Field	Command Description
CMD_FLASH_ERASE	0x10	0x00	Erase the specified range of Flash
CMD_FLASH_DWNLD	0x11	0x00	Download APP to Flash
CMD_DATA_CRC_CHECK	0x12	0x00	CRC check the downloaded APP
CMD_SYS_RESET	0x13	0x00	Reset system
CMD_APP_GO	0x14	0x00	Jump to APP run

3.1.2 Command Data Structure

Here are some conventions in the following description, where "<>" represents a field that must be included, and "()" represents a field included according to different commands.

1. Receiving command data structure:

<CMD_H + CMD_L + LEN + Par[0] + Par[1] + Par[2] + Par[3]> + (DAT).

CMD_H represents the level 1 command field, CMD_L represents the level 2 command field; LEN represents the received data length (that is, the byte length of DAT); Par[0]~ Par[3] represents 4 bytes of command parameters; DAT represents the specific received data;

2. Response command data structure:

<CMD_H + CMD_L + LEN + Par[0] + Par[1] + Par[2] + Par[3]> + (DAT).

CMD_H represents the level 1 command field, CMD_L represents the level 2 command field, and the response level 1

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

and level 2 command fields are the same as the corresponding received command; LEN represents the length of the sent data (that is, the byte length of DAT); DAT represents the specific data of the response command; The two bytes of Par[0]~ Par[1] represent the execution result of the command returned to the upper layer, and the two bytes of Par[2]~ Par[3] remain as reserved 0; if the received level 1 command, and level 2 command field does not belong to any command in the command list, respond Par[0] =0xBB, Par[1] = 0xCC.

3.2 Commands Description

3.2.1 CMD_FLASH_ERASE

The function of this function is erasing Flash in units of pages. The page address number and page number to be erased are provided by the user. The Flash space to be erased cannot exceed the entire Flash space, and at least one page is to be erased.

Receiving command:

Table 3-2 CMD_FLASH_ERASE Receiving Command

Byte / Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field:0x10							
1(CMD_L)	Level 2 command field: reserved							
2~3(LEN)	Send data length: reserved							
4~7(Par)	Page address number (2 bytes): 0~255 Number of pages (2 bytes): 1~256							
(DAT)	Nothing							

- LEN Send data length: 0x00(LEN [0]), 0x00(LEN [1]), LEN = LEN [0] + (LEN [1] << 8).
- The erase address and range consist of four bytes in the Par field:

  Par [0~1]: Page Address number (2 bytes) (0~255): Page address number = Par [0] + Par [1] << 8;

  Par [2~3]: page number (2 bytes) (1~256): Page number = Par [2] + Par [3] << 8;

  The first address of page 0 is 0x0800_3000. For each subsequent page, the page address increments by 1

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

and the first address in the page increments by0x800. For example:

The address at the beginning of page 1 is 0x0800_3000 + 1*0x800 = 0x0800_3800

The address at the top of page 2 is 0x0800_3000 + 2*0x800 = 0x0800_4000

Calculation of the entire address range to be erased:

For example, the page address is 0x01 and the number of pages is 0x02

Erasing address range:

(0x0800_3000 + 1*0x800) ~ (0x0800_3000 + 1*0x800 + 2*0x800)

That is, (first address of the page number) to (First address of the page number + number of pages x page size)

- Reserved value: 0x00;

Response command:

Table 3-3 CMD_FLASH_ERASE Response Command

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x10							
1(CMD_L)	Level 2 command field: Reserved							
2~3(LEN)	Length of sent data: Reserved							
4~7(Par)	Par[0] : status byte 1 Par[1] : status byte 2 Par[2] ~ Par[3] : reserved							
(DAT)	Nothing							

- LEN - Send data length: 0x00(LEN [0]), 0x00(LEN [1]), LEN = LEN [0] + (LEN [1] << 8).
- Status bytes (Par [0], Par [1]) are divided into the following cases according to command execution situation:

1. Return success: status flag bits (0xA0, 0xB0).
2. Return failure: status flag bits (Par [0], Par [1]).

   (1) (0xE0、0x10): Failed to erase the Flash.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

(2) (0xE0、 0x11): Erases a Flash address that exceeds the entire Flash size.

3.2.2 CMD_FLASH_DWNLD

This command allows the user to download code into the specified Flash. The length of the received data must be 4 bytes aligned.

Receiving command:

Table 3-4 CMD_FLASH_DWNLD Receiving Command

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x11							
1(CMD_L)	Level 2 command field: Reserved							
2~3(LEN)	Length of received data: N							
4~7(Par)	Start address for downloading the Flash							
8~8+(N-1)(DAT)	DAT [0 to N-1] : specific data to be downloaded							

- LEN - receive data length : 0xXX (LEN [0]), 0xXX(LEN [1]), LEN = LEN [0] + (LEN [1] << 8)
- Par [0 ~3]: download the starting Address of the Flash. Synthetic rules: Address = Par [0] | (Par [1] < < 8) | (Par [2] < < 16) | (Par [3] < < 24).
- DAT [0 to N-1]: Indicates the specific data to be downloaded. The maximum value is 256 bytes, 4<=N<=256，the value of N must be a multiple of 4.
- Reserved value:0x00;

Response command:

Table 3-5 CMD_FLASH_DWNLD Responce Command

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x11							
1(CMD_L)	Level 2 command field: Reserved							
2~3(LEN)	Length of sent data: Reserved							

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

4~7(Par)	Par[0] : status byte 1 Par[1] : status bytes 2 Par[2]~ Par[3] : reserved
(DAT)	Nothing

- LEN - Send data length: 0x00(LEN[0]), 0x00(LEN[1]), LEN = LEN[0] + (LEN[1]<<8)
- Status bytes (Par[0], Par[1]) are divided into the following cases according to command execution situation:

  1. Download success: status flag bits (0xA0, 0xB0).
  2. Download failed: status flag bits (Par[0], Par[1])

  (1) (0xE0, 0x10): Failed to download the Flash.

  (2) (0xE0, 0x11): The downloaded Flash address exceeds the size of the entire Flash.

  (3) (0xE0, 0x12): The start address of the downloaded Flash is not 4-byte aligned;

  (4) (0xE0, 0x13): The downloaded Flash data length is not a multiple of 4 or greater than 256 bytes.

3.2.3 CMD_DATA_CRC_CHECK

This command is used to check whether the downloaded data is correct. After the downloaded data is complete, the CRC32 check is performed. The receiving command must contain the CRC for downloaded data value and checksum start address and checksum length.

Receiving command:

Table 3-6 CMD_DATA_CRC_CHECK Receiving Command

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x12							
1(CMD_L)	Level 2 command field: Reserved							
2~3(LEN)	Length of sent data: 8							
4~7(Par)	Check start address							
8~15(DAT)	DAT[0:3] : indicates the 32-bit CRC check value DAT[4:7] : Parity length (in bytes, must be a multiple of 4)							

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

- LEN - Send data length: 0x08(LEN[0]), 0x00(LEN[1]), LEN = LEN[0] + (LEN[1]<<8);
- Par [0 ~ 3]: check the starting address. Synthetic rules: Address = Par [0] | (Par [1] < < 8) | (Par [2] < < 16) | (Par [3] < < 24). The Address is only within the scope of the Flash;
- DAT [0 ~ 3]: 32 bit CRC checksum value, synthetic rules for CRC32 = DAT [0] | (DAT [1] < < 8) | (DAT [2] < < 16) | (DAT [3] < < 24);
- DAT (4 ~ 7): check length. Its synthesis rules is CRC_LEN = DAT [4] | (DAT [5] < < 8) | (DAT [6] < < 16) | (DAT [7] < < 24), in which CRC_LEN can only be in the valid range and must be a multiple of 4.
- Reserved value: 0x00;

Response command:

Table 3-7 CMD_DATA_CRC_CHECK Response Command

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x12							
1(CMD_L)	Level 2 command field: Reserved							
2~3(LEN)	Length of sent data: Reserved							
4~7(Par)	Par[0] : status byte 1 Par[1] : status bytes 2 Par[2]~ Par[3] : reserved							
(DAT)	nothing							

- LEN - Send data length: 0x00(LEN[0]), 0x00(LEN[1]), LEN = LEN[0] + (LEN[1]<<8);
- Status bytes (Par[0], Par[1]) are divided into the following cases according to command execution situation:

  1. Verification succeeded: status flag bits (0xA0, 0xB0);

  2. Verification failed: status flag bits (Par[0], Par[1])

     (1) (0xE0, 0x10): CRC verification failed.

     (2) (0xE0, 0x11): The CRC check address exceeds the size of the entire Flash.

     (3) (0xE0, 0x13): The CRC check length is not a multiple of 4.

3.2.4 CMD_SYS_RESET

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

This command is used by software to reset IAP programs.

Receiving command:

Table 3-8 CMD_SYS_RESET Receiving Command

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x13							
1(CMD_L)	Level 2 command field: Reserved							
2~3(LEN)	Length of sent data: Reserved							
4~7(Par)	keep							
(DAT)	Nothing							

- Reserved value: 0x00.

Response command:

Table 3-9 CMD_SYS_RESET Response Command

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x13							
1(CMD_L)	Level command field: Reserved							
2~3(LEN)	Length of sent data: Reserved							
4~7(Par)	Par[0] : status byte 1 Par[1] : status byte 2							
(DAT)	Nothing							

- Status bytes (Par[0], Par[1]) are divided into the following cases according to command execution situation:

  1. Return success: status flag bits (0xA0, 0xB0)

  2. Return failed: status flag bit (0xE0, 0x10).

3.2.5 CMD_APP_GO

After downloading the application to the Flash, jump to the APP program for execution.

Receiving command:

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Table 3-10 CMD_APP_GO Receiving Command

Byte / Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x14							
1(CMD_L)	Level 2command field: Reserved							
2~3(LEN)	Length of sent data: Reserved							
4~7(Par)	Reserved							
(DAT)	Nothing							

- Reserved value: 0x00;

Response command:

Table 3-11 CMD_APP_GO Response Command

Byte / Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	Level 1 command field: 0x14							
1(CMD_L)	Level 2 command field: Reserved							
2~3 (LEN)	Length of sent data: Reserved							
4~7(Par)	Par[0] : status byte 1 Par[1] : status byte 2							
(DAT)	Nothing							

- Status bytes (Par[0], Par[1]) are divided into the following cases according to command execution situation:

  1. Return success: status flag bits (0xA0, 0xB0);

  2. Return failed: status flag bit (0xE0, 0x10).

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

4. Demo IAP Project

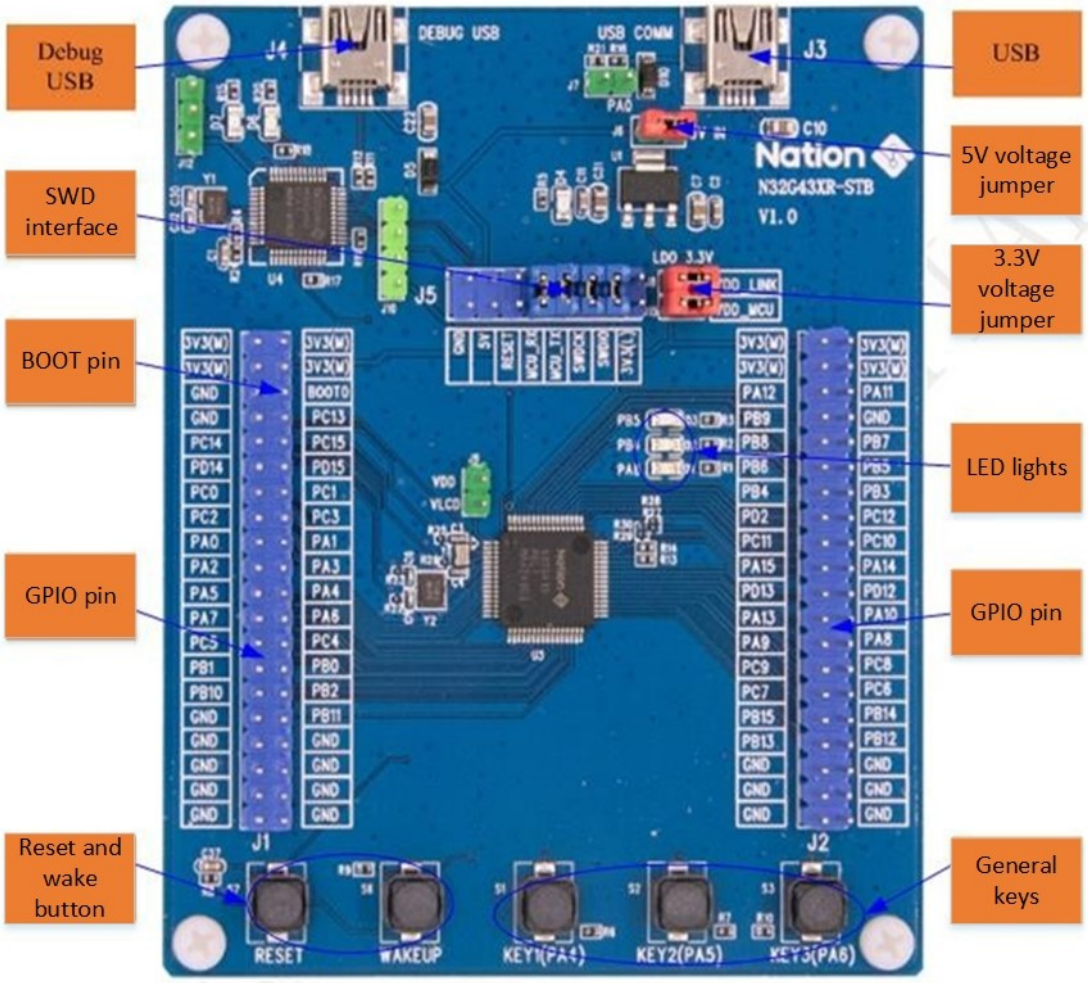
The following sections focus on how to use IAP project.

4.1 Hardware Connection

4.1.1 Development Board

Choose N32G43x series minimum system development board N32G43XR-STB V1.0, as shown in Figure 4-1. Please refer to the document "UG_N32G43XR-STB Development Board Hardware Usage Guide V1.0" for the usage guide of the development board.

Figure 4-1 N32G43XR-STB V1.0 Minimum System Development Board



CAN peripheral interfaces are PD0 and PD1 corresponding to CAN1, and the baud rate is 500Kbps.

SN65HVD230 is selected as the CAN bus tranceiver, as shown in the development board of CAN bus transceiver in

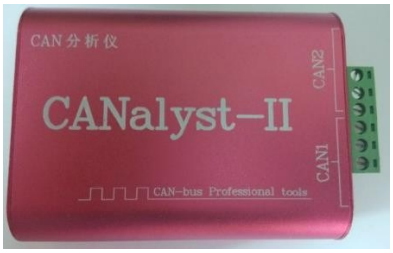
NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Figure 4-2 below.

Figure 4-2 CAN Bus Transmitter and Receiver Development Board

CAN commands are sent using CAN protocol analyzer CANNalyst-II, as shown in Figure 4-3 CAN protocol analyzer.

Figure 4-3 CAN Protocol Analyzer



4.1.2 Software Testing

Download IAP program to N32G43XR-STB V1.0 minimum system development board, reset and run. Open the host computer software of the CAN protocol analyzer CANNalyst-II and configure the baud rate to 500Kbps.Send the command and wait for a response, as shown in Figure 4-4.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Figure 4-4 CAN Bus Transmit Receiver Development Board

	Index	System Time	Time Stamp	Channel	Direction	Frame ID	Type	Format	DLC	Data
Unsupported instruction	00000	10:34:15.168	-	ch1	Send	0x0000	Data	Standard	0x08	x| 00 00 00 00 00 00 00 00
	00001	10:34:15.173	0x1230093	ch1	Receive	0x0400	Data	Standard	0x08	x| 00 00 08 00 BB CC 00 00
Erase instruction	00002	10:34:32.924	-	ch1	Send	0x0400	Data	Standard	0x08	x| 10 00 00 00 00 00 00 FA 00
	00003	10:34:32.994	0x125B71E	ch1	Receive	0x0400	Data	Standard	0x08	x| 10 00 08 00 A0 B0 00 00
Download instructions	00004	10:35:19.632	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 00 00 01 00 30 00 08
	00005	10:35:31.554	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00006	10:35:31.554	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00007	10:35:31.565	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00008	10:35:31.575	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00009	10:35:31.585	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00010	10:35:31.596	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00011	10:35:31.607	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00012	10:35:31.616	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00013	10:35:31.625	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00014	10:35:31.635	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00015	10:35:31.645	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00016	10:35:31.655	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00017	10:35:31.665	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00018	10:35:31.675	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00019	10:35:31.686	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00020	10:35:31.695	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00021	10:35:31.706	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00022	10:35:31.717	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00023	10:35:31.725	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00024	10:35:31.734	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00025	10:35:31.745	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00026	10:35:31.754	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00027	10:35:31.765	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00028	10:35:31.777	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00029	10:35:31.787	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00030	10:35:31.795	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00031	10:35:31.805	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00032	10:35:31.817	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00033	10:35:31.826	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00034	10:35:31.835	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00035	10:35:31.845	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00036	10:35:31.856	-	ch1	Send	0x0400	Data	Standard	0x08	x| 11 22 33 44 55 66 77 88
	00037	10:35:31.884	0x12EB0B7	ch1	Receive	0x0400	Data	Standard	0x08	x| 11 00 08 00 A0 B0 00 00
CRC check instruction	00038	10:35:49.412	-	ch1	Send	0x0400	Data	Standard	0x08	x| 12 00 08 00 00 30 00 08
	00039	10:35:58.636	-	ch1	Send	0x0400	Data	Standard	0x08	x| 14 01 8D 18 00 01 00 00
	00040	10:35:58.644	0x132C543	ch1	Receive	0x0400	Data	Standard	0x08	x| 12 00 08 00 A0 B0 00 00
Reset command	00041	10:36:11.377	-	ch1	Send	0x0400	Data	Standard	0x08	x| 13 00 00 00 00 00 00 00
	00042	10:36:11.394	0x134B658	ch1	Receive	0x0400	Data	Standard	0x08	x| 13 00 08 00 A0 B0 00 00
Jump instruction	00043	10:36:25.513	-	ch1	Send	0x0400	Data	Standard	0x08	x| 14 00 00 00 00 00 00 00
	00044	10:36:25.524	0x136DDE1	ch1	Receive	0x0400	Data	Standard	0x08	x| 14 00 08 00 A0 B0 00 00

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

5. Conclusion

The IAP routine implements a simple upgrade of the APP's two-level BOOT based on the CAN peripheral interface. Users can add commands or data encryption and decryption functions on this basis.

In addition to the CAN peripheral-based interface, there are IAP sample codes based on USART, USB, I2C, SPI and other peripheral-based interfaces, using the same command set.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

6. Version History

Version	Date	Changes
V1.0	2020.10.10	Initial release

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

7. Disclaimer

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD.(Hereinafter referred to as NSING). This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to Nations Technologies Inc. and Nations Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections. enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders. Although NATIONS has attempted to provide accurate and reliable information, NATIONS assumes no responsibility for the accuracy and reliability of this document. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NATIONS be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product.

NATIONS 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, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NATIONS and hold NATIONS harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NATIONS, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road, Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg