One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

Flash/EE Memory Programming via LIN—Protocol 6

by Aude Richard

INTRODUCTION

A key feature of the ADuC703x family of devices is the ability of the devices to download code to on-chip Flash/EE memory while in-circuit. This in-circuit code download is performed via the LIN communications bus.

This application note describes the download protocol implemented on the ADuC703x devices using Protocol 6 to enable users to develop their own LIN programming tools either for series production programming or for application update.

In this application note, host refers to the host machine (microcontroller, DSP, or other machine) attempting to download data to the ADuC703x. Loader refers specifically to serial download firmware resident on the ADuC703x.

Note that this application note describes Protocol 6 only. Protocol 6 follows the general procedure defined by UDS (ISO/DIS 14229-1.2, Road Vehicles Unified Diagnostic Services). However, due to the limited code space available, the services are limited to the minimum actually required.

Protocol 4 is described in Application Note AN-881, Programming Flash/EE Memory via LIN—Protocol 4, available at www.analog.com.

The protocol is shown on the part branding, on Line 3. A60 refers to a released version of Protocol 6 whereas A40 refers to Protocol 4.

Table 1. Branding Example

Line	LFCSP
Line 1	ADuC7033
Line 2	BCPZ 8L
Line 3	A60 #Date Code
Line 4	Assembly Lot Number

The programming sequence can be initiated and controlled by a diagnosis tester, which is connected to the LIN master typically via the controller area network (CAN). The LIN master serves as a gateway, which routes the diagnosis messages from the CAN bus to the LIN bus. To facilitate the routing of the diagnosis messages from CAN to LIN, the LIN commands for programming the module should comply with the *LIN Diagnostic and Configuration Specification* (Revision 2.0, September 23, 2003).

RUNNING THE ADuC703x LOADER

To enable an unattended download via LIN, the ADuC703x enters loader mode only if NTRST is low during reset and the content of Flash/EE memory at Address 0x80014 is neither 0x27011970 nor the Page 0 checksum, as shown in Figure 1.

Normally, NTRST is kept low and entry to download mode is determined by the content of Flash Address 0x80014. Typically, the value at Flash Address 0x80014 is not 0xFFFFFFFF; therefore, user code must have a built-in mechanism for corrupting location 0x80014 or erasing Page 0 (Flash Address 0x0 to Flash Address 0x200) and for resetting the part. This mechanism allows entry to download mode to reprogram the part. Ideally, the value at Flash Address 0x80014 should be programmed last to enable reentry to download mode even in the event that the power fails or another error occurs during programming of the bulk of the program.

The checksum of Page 0 is simply the sum of all the half words in Page 0, excluding the two half words of the word at Address 0x80014. This sum must be stored at Address 0x80014.

TABLE OF CONTENTS

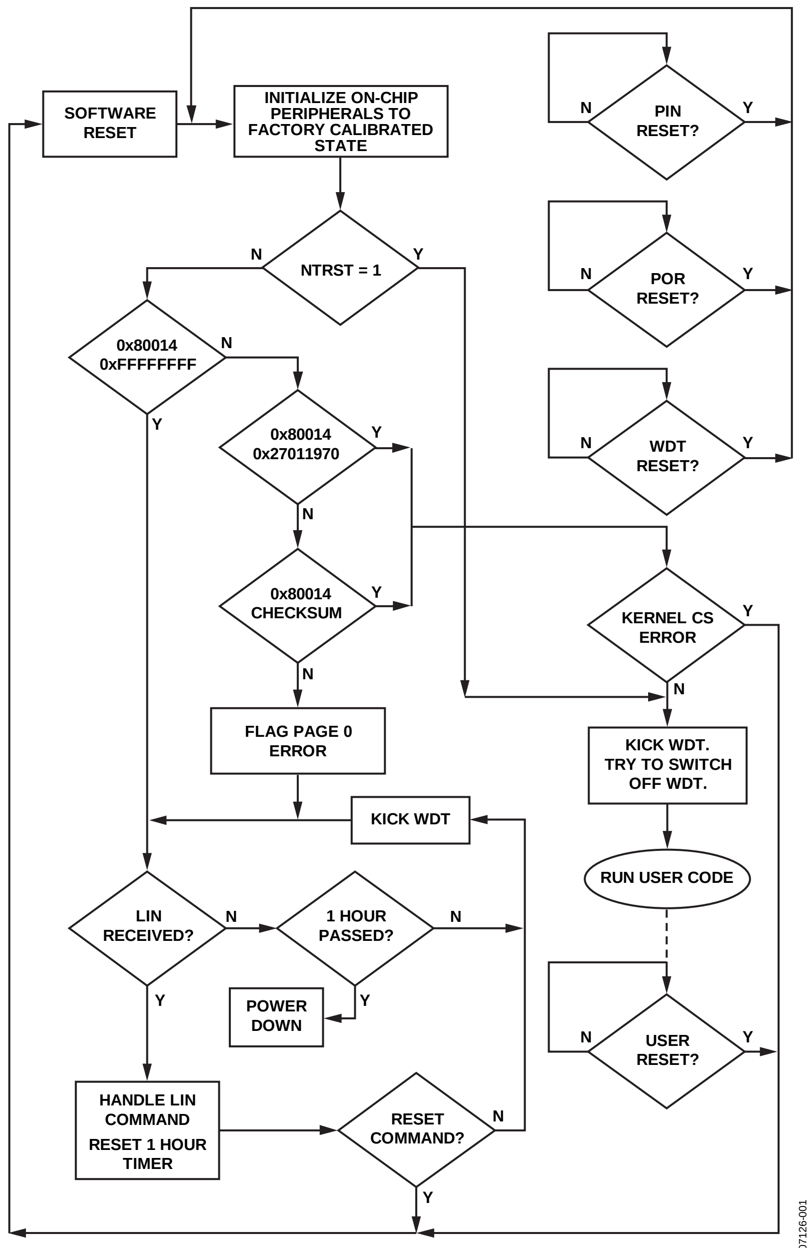

Figure 1. Entering Download Mode

PACKET STRUCTURE

The LIN communication with the loader must comply with the following general requirements as per the *LIN Diagnostic and Configuration Specification*, Revision 2.0, September 23, 2003:

- The kernel must implement a slot for each of the two LIN diagnostic frames: master request frame and slave response frame.
- The request from the LIN master must comply with the packet data unit (PDU) format shown in Table 2.
- The responses must comply with the PDU format shown in Table 3.
- Only the PCI-type single frame (SF) can be used. First frame (FF) and consecutive frames (CF) are not supported.
- All frames use the classic checksum.
- Unrecognized commands are ignored.
- Any frame with an error, such as a communication error, is ignored. Thus, a faulty erase routine frame is ignored. A faulty request download frame is ignored and therefore subsequent transfer data frames are not recognized and no programming occurs. Any faulty transfer data frame terminates recognition of transfer data frames. In fact, any frame with correct NAD and PCI ≠ 0x05 or SID ≠ 0x36 or a wrong checksum terminates recognition of transfer data frames.
- The addresses shown in the Values column in Table 5 through Table 7 and Table 9 through Table 15 in the Commands Implemented in the On-Chip Loader section are hard-coded values and are not examples.

Table 2. Frame Identifier 0x3C

Byte	Description
Byte 0	Node address (NAD)
Byte 1	Protocol control information (PCI)
Byte 2	Service identifier (SID)
Byte 3	Data 1
Byte 4	Data 2
Byte 5	Data 3
Byte 6	Data 4
Byte 7	Data 5

Table 3. Frame Identifier 0x3D

Byte	Description
Byte 0	Node address
Byte 1	Protocol control information
Byte 2	Response identifier (RSID = SID + 0x40)
Byte 3	Data 1
Byte 4	Data 2
Byte 5	Data 3
Byte 6	Data 4
Byte 7	Data 5

COMMANDS IMPLEMENTED IN THE ON-CHIP LOADER

Seven commands, described in this section, are implemented in Protocol 6.

- Assign NAD
- Read-by-Identifier
- Erase Routine
- Request Download
- Transfer Data
- Check Routine
- ECU Reset

ASSIGN NAD

This command, which assigns a new NAD to the slave, is required because different network systems need to use different NADs for their respective logical slave nodes.

Request

The supplier ID 0x003A was assigned to Analog Devices, Inc., by the LIN Consortium. The function ID for Protocol 6 is listed in Table 4.

Table 4. Function ID

Device	MSB	LSB
ADuC7032	0x00	0x32
ADuC7033	0x00	0x32
ADuC7036DCPZ	0x00	0x36
ADuC7039	0x00	0x39

Note that to guard against the loss of a slave as a result of network corruption, the slave always recognizes an Assign NAD command with the broadcast NAD 0x7F. This recognition occurs regardless of what the actual NAD of the slave is when this Assign NAD command is transmitted. The kernel then makes the decision whether or not the command is intended for this slave by checking the supplier ID and function ID. Function ID 0x32 is used as an example within this document.

Table 5. Assign NAD Request

Byte	Description	Values
Byte 0	Initial NAD	0x7F
Byte 1	PCI	0x06
Byte 2	SID	0xB0
Byte 3	Analog Devices supplier ID LSB	0x3A
Byte 4	Analog Devices supplier ID MSB	0x00
Byte 5	Function ID LSB	0x32
Byte 6	Function ID MSB	0x00
Byte 7	New NAD	User value

Response

The slave does not respond to this request.

READ-BY-IDENTIFIER

After an abort of the programming sequence, the diagnosis tester requests an identification of the LIN slave using the Read-by-Identifier request before a second programming attempt is initiated.

Request

Four identifiers (Identifier 0x0, Identifier 0x32, Identifier 0x33, and Identifier 0x34) are supported.

Table 6. Read-By-Identifier Request

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x06
Byte 2	SID	0xB2
Byte 3	Identifier	0x0/0x32/0x33/0x34
Byte 4	AD supplier ID LSB	0x3A
Byte 5	AD supplier ID MSB	0x00
Byte 6	Function ID LSB	0x32
Byte 7	Function ID MSB	0x00

Identifier 0x0

The read-by-identifier request with Identifier 0x0 returns the LIN product identification information. In the case of an ADuC703x LIN product, this information consists of the 8-byte data frame response shown in Table 7.

Table 7. Identifier 0x0 Data Frame Response

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x06
Byte 2	RSID	0xF2
Byte 3	AD Supplier ID LSB	0x3A
Byte 4	AD Supplier ID MSB	0x00
Byte 5	Function ID LSB	0x32
Byte 6	Function ID MSB	0x00
Byte 7	Variant	0x00

Identifier 0x32, Identifier 0x33, and Identifier 0x34

The response to these identifiers return user configured data from device memory. The loader expects the contents of the data bytes to be located in the last page of the Flash/EE memory, as shown in Table 8.

Table 8. Flash Locations Allocated to Other Identifiers

Address	ID	Byte	Contents
0x977ED	0x32	Data 1	User value
0x977EE	0x32	Data 2	User value
0x977EF	0x32	Data 3	User value
0x977F0	0x32	Data 4	User value
0x977F1	0x32	Data 5	User value
0x977F2	0x33	Data 1	User value
0x977F3	0x33	Data 2	User value
0x977F4	0x33	Data 3	User value
0x977F5	0x33	Data 4	User value
0x977F6	0x33	Data 5	User value
0x977F7	0x34	Data 1	User value
0x977F8	0x34	Data 2	User value
0x977F9	0x34	Data 3	User value
0x977FA	0x34	Data 4	User value
0x977FB	0x34	Data 5	User value

Note that the last four bytes of the last page of the Flash/EE memory are reserved for the checksum.

Response

Positive response of the LIN slave is shown in Table 9.

Table 9. LIN Slave Response

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x06
Byte 2	RSID	0xF2
Byte 3	Data 1	User value
Byte 4	Data 2	User value
Byte 5	Data 3	User value
Byte 6	Data 4	User value
Byte 7	Data 5	User value

The slave does not give a negative response.

ERASE ROUTINE

Overview

It is possible to erase several pages at once and to request the download and transfer of update data for several subsequent pages. Which update strategy is chosen depends only on the diagnosis tester. However, because 1 in 1000 LIN frames can be expected to show a transmission error, repeating the erase, programming, and verification cycle independently for each single page is recommended. The following three constraints have to be taken into account:

- It is not possible to program a memory area smaller than one Flash page of the slave, that is, smaller than 512 bytes.
- Special consideration must be given to Page 0 programming. It must initially be programmed with Location 0x80014 = 0xFFFFFFFF.
- In addition to the verification of the single Flash/EE memory pages, a verification of the checksum of the entire user Flash/EE memory area is recommended before the ECU reset is performed.
- After the last page is verified, Page 0 must be reprogrammed (without erasing) except 0x80014, which is assigned the checksum or another desired value.

Request

The erase routine erases the contents of N number of Flash pages, starting with Page P. Each Flash page consists of 512 bytes. The value N = 0 is reserved for future use.

Table 10. Erase Routine Request

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x06
Byte 2	SID	0x31
Byte 3	Subfunction ID, 1st byte	0xFF
Byte 4	Subfunction ID, 2nd byte	0x00
Byte 5	Index of start page, LSB P	User value
Byte 6	Index of start page, MSB P	User value
Byte 7	Number of pages to be erased, N	User value

Index, in Byte 5 and Byte 6, is the page start address nine bits right-shifted. For example, the start address of Page 2 in Flash/EE memory is 0x80400; right-shifted by nine bits is 0x0402. This index is represented as Byte 5 = 0x02 and Byte 6 = 0x04.

Response

The slave does not respond to this request.

REQUEST DOWNLOAD

Refer to the Erase Routine Overview section.

Request

Table 11. Download Request

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x04
Byte 2	SID	0x34
Byte 3	Index of start page, LSB P	User value
Byte 4	Index of start page, MSB P	User value
Byte 5	Number of pages to be programmed, N	User value
Byte 6	Unused	0xFF
Byte 7	Unused	0xFF

The request download command defines the memory area to be programmed. The subsequent data, transmitted via the transfer data command, is written to N number of pages, starting with Page P.

Response

The slave does not respond to this request.

TRANSFER DATA

These requests are acted on only when following a download request.

Request

The transfer data command transmits Flash data. The slave expects N × 512 bytes of data, where N is the number of pages as defined by the request download command. Only full 4-byte words are allowed. With a LIN baud rate of 19.2 kbps, it takes approximately 512/4 × 10 ms = 1.28 sec to flash a single page.

Table 12. Data Transfer Request

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x05
Byte 2	SID	0x36
Byte 3	Data 1	User value
Byte 4	Data 2	User value
Byte 5	Data 3	User value
Byte 6	Data 4	User value
Byte 7	Unused	0xFF

Response

The slave does not respond to this request.

CHECK ROUTINE

Request

The check routine command calculates the checksum for the memory area starting with Page P and ending with Page P + N − 1. The response, with N = 0, is undefined. This command should be run for each single page, but also after all programming is done because errors in the erase or download commands could affect pages other than the intended pages. The diagnosis tester compares the checksum received from the LIN slave with a reference checksum provided in the Flash data container. If the checksums differ, the programming procedure is repeated. The checksum is the sum over all 16-bit values from the first 16-bit word of Page P to the last 16-bit word of Page P + N − 1. Checksum = (Σ 16-bit words) modulo 32. For a single page, 500 μs are required after the check routine request is received by the slave to calculate the checksum. A cyclic redundancy check (CRC) algorithm is not used here for the following reasons:

- The calculation of the CRC checksum takes about 8 times longer than the calculation of the simple checksum used here.
- Analog Devices assumes an error model in which not all the half words or bits in the checked area are programmed as required. Such a page always shows less zeros and gives a higher checksum. Alternatively, programming an unerased page consistently gives more zeros and a lower checksum. The third possibility is single incorrect half words or bits. The probability for detecting such errors is the same whether one uses a CRC or the simple checksum.

Table 13. Check Routine Request

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x06
Byte 2	SID	0x31
Byte 3	Subfunction ID, 1st byte	0xFF
Byte 4	Subfunction ID, 2nd byte	0x01
Byte 5	Index of start page, LSB P	User value
Byte 6	Index of start page, MSB P	User value
Byte 7	Number of pages, N	User value

Response

Table 14. Check Routine Response

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x05
Byte 2	RSID	0x71
Byte 3	Checksum LSB	User value
Byte 4	Checksum, 2nd byte	User value
Byte 5	Checksum, 3nd byte	User value
Byte 6	Checksum, MSB	User value
Byte 7	Unused	0xFF

ECU RESET

Request

Table 15. ECU Reset Request

Byte	Description	Values
Byte 0	NAD	User value
Byte 1	PCI	0x02
Byte 2	SID	0x11
Byte 3	Subfunction ID	0x01
Byte 4	Unused	0xFF
Byte 5	Unused	0xFF
Byte 6	Unused	0xFF
Byte 7	Unused	0xFF

The ECU reset command performs a reset of the slave. The ADuC703x device restarts as shown in Figure 1. If the value at Address 0x80014 matches the checksum of Page 0 or is 0x27011970, this results in the execution of the application software.

Response

The slave does not respond to this request.

SAMPLE LIN PROGRAMMING UTILITIES

Analog Devices provides LINBWSD, demonstration software that requires a USB-to-LIN adaptor. A Flash/EE programming demonstration utility, developed with the CANoe.LIN™ tool from Vector Informatik GmbH, is also available.

For more information, contact the Microcontroller Development Group at Analog Devices.