# Analog Devices, Inc.
# ADuC703x Flash Programming via LIN Protocol 6

Date: 14th June 2007

Version: 1.1

# Contents



# Introduction

The purpose of this document is to specify a flash procedure for updating the flash memory of a slave module via LIN. The programming of the slave shall be possible without dismounting the module. The proposed procedure follows the general procedure defined by UDS [1], however, due to the limited code space available the services are limited to the minimum actually required.

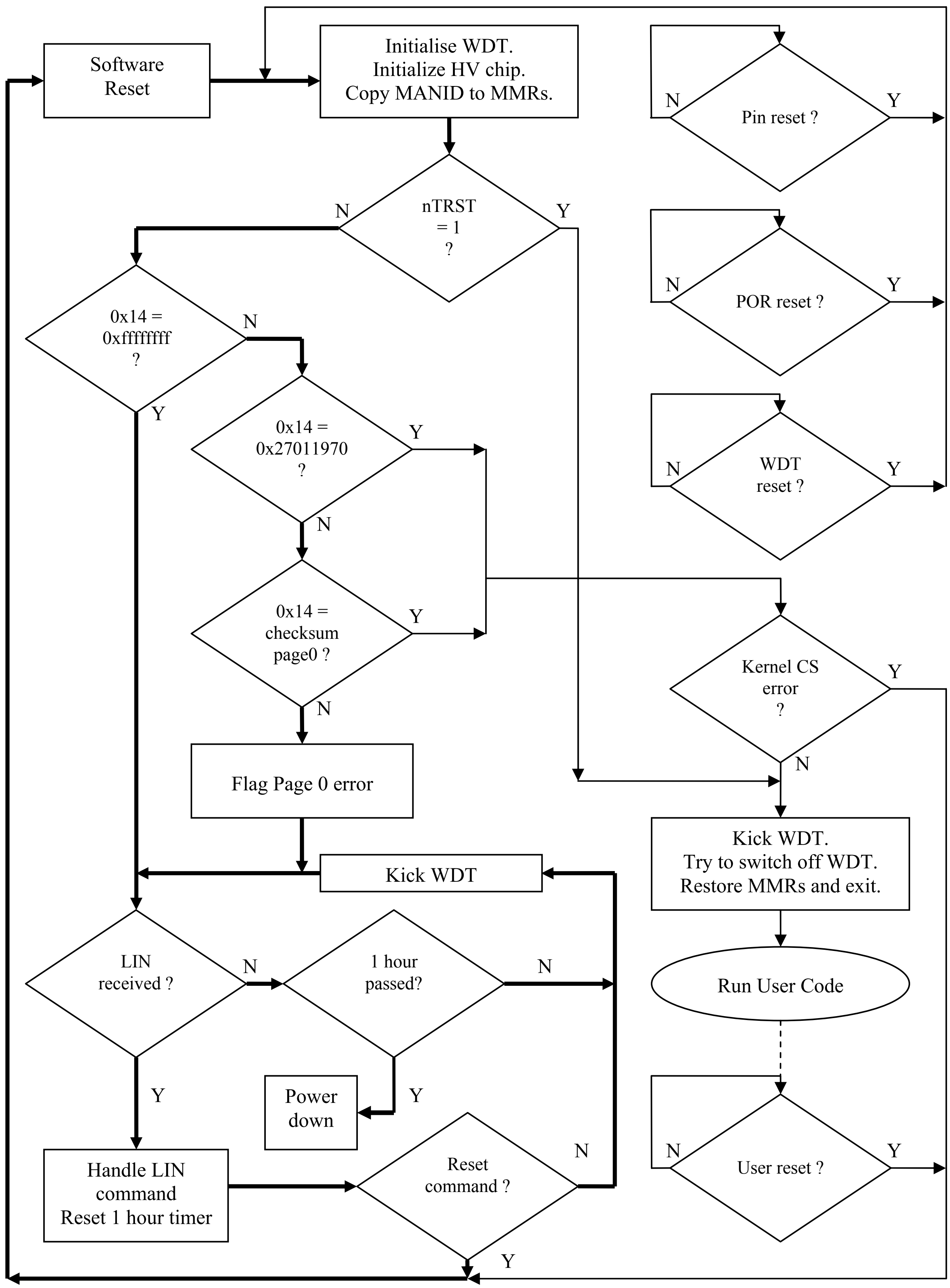
The programming sequence can be initiated and controlled by a diagnosis tester which is connected to the LIN master typically via CAN. The LIN master serves as a gateway which routes the diagnosis messages from the CAN bus to the LIN bus. To facilitate the routing of the diagnosis messages from CAN to LIN, the LIN commands for programming the module shall comply with the LIN diagnostic and configuration specification [2].

## *Entering Download*

The programming procedure consists of two major phases. Phase 1 is controlled by the application software and is customer dependent. In this phase, phase 2 is enabled by corrupting address 0x14. At the following software or hardware reset phase 2 is entered by entering the kernel download mode (The above assumes nTRST = low as will be the case in a module).

The block diagram of the kernel below shows the phase 2 functionality in bold lines. Note that phase 2 is only available after a reset as described above and can not be entered directly from the application software (user mode). The dotted line at the bottom left is the exit from phase 1.

The checksum of page 0 is simply the sum of all the halfwords in page 0 excluding the two halfwords of the word at address 0x14. This sum is normally stored at address 0x14.


Software Reset
Initialise WDT. Initialize HV chip. Copy MANID to MMRs.
Pin reset ?
POR reset ?
WDT reset ?
nTRST = 1 ?
0x14 = 0xffffffff ?
0x14 = 0x27011970 ?
0x14 = checksum page0 ?
Kernel CS error ?
Flag Page 0 error
Kick WDT
Kick WDT. Try to switch off WDT. Restore MMRs and exit.
Run User Code
LIN received ?
1 hour passed?
Power down
Handle LIN command Reset 1 hour timer
Reset command ?
User reset ?
N
Y

# General requirements

## *Packet Structure*

The LIN communication with the kernel code shall comply with the following general requirements [2]:

1. The Kernel shall implement a slot for each of the two LIN diagnostic frames, i.e. master request frame and slave response frame.

2.The request from the LIN master shall comply with the following Packet Data Unit (PDU) format:

Frame Identifier **0x3C**

| | |
|---|---|
| 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 |

3. The responses shall comply with the following PDU format:

Frame Identifier **0x3D**

| | |
|---|---|
| Byte 0 | Node address (NAD) |
| Byte 1 | Protocol Control Information (PCI) |
| 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 |

4. Only the PCI type Single Frame (SF) will be used. First Frame (FF) and Consecutive Frames (CF) are not supported.

5. All frames will use the classic checksum.

6. Unrecognized commands will be ignored.

7. Any frame with an error e.g. due to a communication error will be ignored. Thus a faulty Erase Routine frame will be ignored. A Request Download frame will be ignored and therefore Transfer Data frames will not be recognized and no programming will occur. Any faulty Transfer Data frame will terminate recognition of Transfer Data frames. In fact any frame with correct NAD and PCI!=0x05 or SID!=0x36 or wrong checksum will terminate recognition of Transfer Data frames.

8. Where values are given in the ‘Byte’ allocations for the commands below these are hard coded values and not examples.

# Commands implemented in the Kernel

## *Assign NAD*

This command is required since different NADs might be used in different systems.

Request:

| | | |
|---|---|---|
| Byte 0 | Initial NAD | 0x7F |
| Byte 1 | PCI | 0x06 |
| Byte 2 | SID | 0xB0 |
| 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 | New NAD | |

This command will assign a new NAD to the slave.
The supplier ID 0x003A was assigned to Analog Devices Inc. by the LIN Consortium. The function ID is 0x0032.
Note: To guard against the loss of a slave as a result of some sort of network corruption, the slave will always recognize an Assign NAD command with an NAD of 0x7F regardless of what the actual NAD of the slave is when this Assign NAD command is transmitted. The Kernel then makes the decision as to whether the command is intended for this slave or not by checking the Supplier ID and Function ID.

Response:
The slave will not respond to this request.

## *Read By Identifier*

Request:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| 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 |

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.

Four identifiers are supported.

### • Identifier 0x0:

The Read By Identifier request with identifier 0x0 will return the LIN Product Identification information. In the case of an ADuC703X LIN product, this information comprises of the following 8-byte data frame response:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| 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 |

### • Identifiers 0x32, 0x33, 0x34

The response to these identifiers can be configured freely. The kernel code expects the contents of the data bytes to be located in the last page of the flash memory as follows.

| Address | ID | Byte | Contents |
|---|---|---|---|
| 0x977ed | 0x32 | 0 | |
| 0x977ee | 0x32 | 1 | |
| 0x977ef | 0x32 | 2 | |
| 0x977f0 | 0x32 | 3 | |
| 0x977f1 | 0x32 | 4 | |
| 0x977f2 | 0x33 | 0 | |
| 0x977f3 | 0x33 | 1 | |
| 0x977f4 | 0x33 | 2 | |
| 0x977f5 | 0x33 | 3 | |
| 0x977f6 | 0x33 | 4 | |
| 0x977f7 | 0x34 | 0 | |
| 0x977f8 | 0x34 | 1 | |
| 0x977f9 | 0x34 | 2 | |
| 0x977fa | 0x34 | 3 | |
| 0x977fb | 0x34 | 4 | |

Note: The last four bytes of this page are reserved for the checksum.

Positive response of the LIN slave:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| Byte 1 | PCI | 0x06 |
| Byte 2 | RSID | 0xF2 |
| Byte 3 | D1 | |
| Byte 4 | D2 | |
| Byte 5 | D3 | |
| Byte 6 | D4 | |
| Byte 7 | D5 | |

The slave will not give a negative response.

## Erase Routine

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, since 1 in 1000 LIN frames can be expected to show a transmission error it is recommended, that the erase, programming and verify cycle is repeated independently for each single page. Three constraints have to be taken into account:

1. It is not possible to program a memory area smaller than one flash page of the slave e.g. smaller than 512 bytes.
2. Special consideration must be given to page 0 programming. It must initially be programmed with location 0x80014 = 0xffffffff. After the last page is verified page 0 must be reprogrammed (without erasing) with all words 0xffffffff except 0x80014 = Checksum or other desired value.
3. It is recommended that in addition to the checks of the single flash pages also the checksum of the whole flash area is checked before the ECU Reset is performed.

Request:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| 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* | |
| Byte 6 | Index of start page, MSB *P* | |
| Byte 7 | Number of pages to be erased *No* | |

The Erase Routine shall erase the contents of *No* flash pages starting with page *P*. Each flash page consists of 512 bytes. The value *No* = 0 is reserved for future use.

Response:

The slave will not respond to this request.

## *Request Download*

See introduction to Erase Routine above.

Request:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| Byte 1 | PCI | 0x04 |
| Byte 2 | SID | 0x34 |
| Byte 3 | Index of start page, LSB *P* | |
| Byte 4 | Index of start page, MSB *P* | |
| Byte 5 | Number of pages to be programmed *No* | |
| Byte 6 | Unused | 0xFF |
| Byte 7 | Unused | 0xFF |

The Request Download command defines the memory area to be flashed. The subsequent data, transmitted via the Transfer Data command will be written to No pages starting with page P.

Response:
The slave will not respond to this request.

## *Transfer Data*

These requests are only acted on if following a Request Download request.
Request:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| Byte 1 | PCI | 0x05 |
| Byte 2 | SID | 0x36 |
| Byte 3 | Data packet 0 | |
| Byte 4 | Data packet 1 | |
| Byte 5 | Data packet 2 | |
| Byte 6 | Data packet 3 | |
| Byte 7 | unused | 0xFF |

The Transfer Data command is used to transmit the flash data. The slave expects *No**512 bytes of data, where *No* is the number of pages as defined with the Request Download command. Only full 4 byte words are allowed. With a LIN baud rate of 19.2 kbps it will take approximately 512/4*10ms = 1.28s to flash a single page.

Response:
The slave will not respond to this request.

## *Check Routine*

Request:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| 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* | |
| Byte 6 | Index of start page, MSB *P* | |
| Byte 7 | Number of pages *No* | |

The Check Routine command calculates the checksum for the memory area starting with page P and ending with page P+*No*-1. The response with *No* = 0 is undefined. This command should be run for each single page but also after all programming is done since 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+*No*-1. Checksum = ( Σ 16 bit words ) mod 32. For a single page 500us are required after the Check Routine request is received by the slave to calculate the checksum. We do not use a cyclic redundancy check (CRC) algorithm here for the following reasons:

1. The calculation of the CRC checksum takes about 8 times longer than the calculation of the simple checksum used here.
2. We assume an error model in which not all the halfwords or bits in the checked area are programmed as required. Such a page will always show less zeros and will give a higher checksum. Alternately programming an unerased page will consistently give more zeros and a lower checksum. The third possibility is single incorrect halfwords or bits. The probability for detecting such errors is the same whether one uses a CRC or the simple checksum.

Response:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| Byte 1 | PCI | 0x05 |
| Byte 2 | RSID | 0x71 |
| Byte 3 | Checksum LSB | |
| Byte 4 | Checksum, 2nd Byte | |
| Byte 5 | Checksum, 3nd Byte | |
| Byte 6 | Checksum, MSB | |
| Byte 7 | Unused | 0xFF |

## *ECU Reset*

Request:

| | | |
|---|---|---|
| Byte 0 | NAD | |
| 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 will perform a reset of the slave. This will start the application software if the value at address 0x14 matches the checksum of page 0 or is 0x27011970.

Response:
The slave will not respond to this request.

# References

[1] ISO/DIS 14229-1.2 Road vehicles unified diagnostic services (UDS)
[2] LIN Diagnostic and Configuration Specification, Revision 2.0, September 23, 2003

# Change history

| Revision | By | Changes |
|---|---|---|
| 1.0 | Aude Richard | Initial official release |
| 1.1 | Eckart Hartmann | Page 5, *Assign NAD* – Function ID is fixed at 0x32 for all ADuC703x parts with this protocol. |