AN-806
APPLICATION NOTE

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

I²C Download Protocol for ADuC70xxBCPZxxI Models

by Aude Richard

INTRODUCTION

One of the many features of ADI's MicroConverter® product family is the ability of the device to download code to its on-chip Flash/EE memory while "in-circuit." On the standard ADuC702x models, this in-circuit code download feature is conducted over the device's UART serial port.

On models containing the letter I at the end of the product number (ADuC7020BCPZ62I, for example,), this code download feature is conducted over the device's I²C® serial port. This application note applies only to I models ADuC7019BCPZ62I, ADuC7020BCPZ62I, and ADuC7021BCPZ62I.

A windows executable program (I2CWSD.exe) is provided which allows the user to download code from a USB port via a third party I²C pod to the MicroConverter. Schematics and code for this I²C pod can also be found on the analog website at www.analog.com and the pod can be purchased at www.fh-pforzheim.de/stw-svs/texte/Dongle.html. It should however be emphasized that any master host machine with I²C master (microcontroller, DSP, or other) can download to the MicroConverter once the host machine adheres to the I²C download protocols detailed here.

This application note outlines in detail the MicroConverter I²C download protocol, allowing end-users to both fully understand the protocol and, if required, to implement this protocol (embedded host to embedded MicroConverter) successfully in an end target system.

For the purposes of clarity, the term Host refers to the host machine (microcontroller, DSP, or other machine) attempting to download data to the MicroConverter. The term Loader refers specifically to the on-chip serial download firmware resident on the MicroConverter.

THE MICROCONVERTER LOADER

Running the Loader

To allow unattended download via I²C, the ADuC702x enters loader mode only if P0.0 (serial download) is low during reset and the contents of flash at address 0x80014 is 0xFFFFFFFF. To allow P0.0 to determine if loader mode is entered, the user must ensure in the code of the part that the word at address 0x80014 is 0xFFFFFFFF.

Alternatively, P0.0 can be kept low and entry to download mode is determined by the contents of address 0x80014. Typically the value at address 0x80014 is not 0xFFFFFFFF, therefore user code should have a mechanism built in for erasing page 0 (address 0x80000 to 0x80200) and resetting the part. This mechanism allows entering download mode to reprogram the part.

When reprogramming the part, ideally the value at 0x80014 should be programmed last to allow re-entry to download mode in case power fails or another error occurs during reprogramming of the bulk of the program.

The Physical Interface

Once triggered, the loader configures the I²C0 port on P1.0 and P1.1 as a slave device. Its slave address is 0x04, therefore the host must start each transmission of data to the loader with the byte 0x04 (I²C write), and each request for data from the loader with the byte 0x05 (I²C read). The data of the first packet sent by the loader must be backspace (BS = 0x08) to start the protocol.

After receiving the backspace character, the loader waits for the host to ask for its reset string. The loader then sends the following 24-byte ID data packet:

15 bytes = product identifier
3 bytes = hardware and firmware version number
4 bytes = reserved for future use
2 bytes = line feed and carriage return

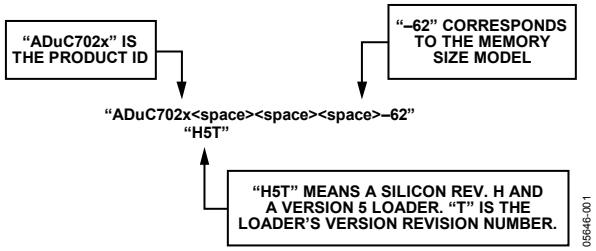

Figure 1.

DEFINING THE DATA TRANSPORT PACKET FORMAT

Once the I2C has been configured, the transfer of data can begin. The general communications data transport packet format is shown in Table 1.

Packet Start ID Field

The first field is the packet start ID field which contains two start characters (07h and 0Eh). These bytes are constant and are used by the loader to detect a valid data packet.

Number of Data Bytes Field

The next field is the total number of data bytes, including Data 1 (Command Function). The minimum number of data bytes is 5, which corresponds to the command function and the address. The maximum number of data bytes allowed is 255; command function, 4-byte address, and 250 bytes of data.

Command Function Field (Data 1)

The command function field describes the function of the data packet. One of five valid command functions is allowed. The five command functions are described by one of five ASCII characters: 'E,' 'W', 'V,' 'P' or 'R.'

Address Field (Data 2 → 5)

The address field contains a 32-bit address h, u, m, l, with MSB in h and LSB in l.

Data Byte Fields 6 – 255

User code is downloaded/verified by bytes. The data byte field contains a maximum of 250 data bytes.

The data must be stripped out of the Intel® Extended HEX 16-byte record format and reassembled by the host as part of the above data form before transmission to the loader.

Checksum Field

The data packet checksum is written into this field. The twos complement checksum is calculated from the summation of the hex values in the number of bytes field and the hex values in the Data 1 to 255 fields (if they exist). The checksum is the twos complement value of this summation: the 8-bit sum of number of data bytes and data bytes 1–255 and the checksum should equal 00h. This can also be expressed mathematically as follows:

$$CS = 0x00 - (\text{No. Data Bytes} + \sum_{N=1}^{255} \text{Data Byte}_N)$$

Acknowledge of Command

The host must interrogate the loader to have its response after each command a BEL (07h) as a negative response, or an ACK (06h) as a positive response. A BEL will be transmitted by the loader if it receives an incorrect data packet format on verification of the checksum byte. The loader does not give a warning if data is downloaded over old (unerased) data or to an invalid address. The PC interface will have to ensure that any location where code will be downloaded is erased. The list of data packet command functions is shown in Table 2.

Table 1. Data Transport Packet Format

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data x (x = 6 → 255)	Checksum
0x07	0x0E	5 → 255	E, W, V, P or R	h, u, m, l	xx	CS

Table 2. Data Packet Command Functions

Command Functions	Command Byte in Data 1 Field	Loader Positive Response	Loader Negative Response
Erase Page	E (0x45)	ACK (0x06)	BEL (0x07)
Write	W (0x57)	ACK (0x06)	BEL (0x07)
Verify	V (0x56)	ACK (0x06)	BEL (0x07)
Protect	P (0x50)	ACK (0x06)	BEL (0x07)
Run (Jump to User Code)	R (0x52)	ACK (0x06)	BEL (0x07)

Table 3. Erase Flash/EE Memory Command

Start ID		No of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data 6 (pages)	Checksum
0x07	0x0E	6	E (0x45)	h, u, m, l	x pages (1 to 124)	CS

Table 4. Program Flash/EE Memory Command

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data x (x = 1 → 250)	Checksum
0x07	0x0E	5 + x (6 → 255)	W (0x57)	h, u, m, l	Data Bytes	CS

Table 5. Verify Flash/EE Memory Command

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data x (x = 1 → 250)	Checksum
0x07	0x0E	5 + x (6 → 255)	V (0x56)	h, u, m, l	Complemented Data Bytes	CS

Table 6. Flash/EE Memory Protection Command

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data 6	Checksum
0x07	0x0E	0x06	P (0x50)	h, u, m, l	Type	CS

Table 7. Jump to User Code (Remote RUN) Command

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Checksum
0x07	0x0E	0x05	R (0x52)	h, u, m, l = 0x80000	0xA1

DATA PACKET COMMAND FUNCTIONS

Erase Command

The erase command allows the erasing of one page up to all pages of Flash/EE from a specific address determined by data 2 → 5. This command also requires the number of pages to erase. If the address is 00000000h and the number of pages is 00h, the loader will interpret it as a mass erase command which will erase the entire user code space and the Flash/EE protection.

The data packet for the erase command is shown in Table 3.

Write Command

The write command requires the number of the data byte (which is data 1 + data 2 + data 2 → 5 + data x), the command, the address of the first data byte to program, and the data bytes to program. As the bytes arrive, they are programmed into Flash/EE. The loader sends a BEL if the checksum is incorrect or if the address received is out of range. If the host receives a BEL, the download process should be aborted and the entire download sequence started again.

Verify Command

The verify command is almost identical to the write command, as shown in Table 5. The command field is V (0x56), but to improve the chance of detecting errors the data bytes are modified: the low 5 bits are shifted to the high 5 bits, and the high 3 bits are shifted to the low 3 bits.

Table 8. Verify Command, Bit Modifications

Original Bits	Transmitted Bits	Restored Bits
7	4	7
6	3	6
5	2	5
4	1	4
3	0	3
2	7	2
1	6	1
0	5	0

The loader restores the correct bit sequence and compares it to the flash contents. If it is correct and the checksum is correct, ACK (0x06) is returned; otherwise BEL (0x07) is returned.

Flash/EE Memory Protection Command

To use this command a three-step sequence must be followed:

1. Initiation of the command: type must be 0x00 and "huml" can be any value.
2. Send the address of the group of page to protect. This step must be repeated for each group of page to protect and type must be 0x0F
3. Send the key in "huml", type must be 0x01. FEEADR will take the value of "hu" and FEEDAT will take the value of "ml". If no keys are required, "huml" must be 0xFFFFFFFF.

For example, to protect page 0 to 7 against writing, set the read protection and use key 0x12345678, the following commands must be sent.

1. start sequence:

0x07 0x0E 0x06 0x50 0xXXXXXXXX 0x00 CS

2. protection:

0x07 0x0E 0x06 0x50 0x00000000 0x0F CS (pages 0 to 3)

0x07 0x0E 0x06 0x50 0x00000200 0x0F CS (pages 4 to 7)

0x07 0x0E 0x06 0x50 0x0008F800 0x0F CS (read protection)

3. key and end of sequence :

0x07 0x0E 0x06 0x50 0x12345678 0x01 CS

Note:
The protection command is only available rev O and later revision of the loader. On rev O, FEEADR = 'ml' and FEEDAT = 'ml'. On later revision of the loader FEEADR = 'hu'.

This protocol doesn't allow to unprotect the Flash/EE memory. To remove the protection, a mass erase command can be used.

Jump to User Code (Remote Run) Command

Once the host has transmitted all data packets to the loader, the host can send a final packet instructing the loader to force the MicroConverter program counter to a given address, and thus begin executing the code that has just been downloaded. Table 7 shows an example of a Remote RUN or "jump to user code" from address 0x00.

Only run from start of the Flash/EE (h, u, m, l = 0x80000) is supported at present.

AN05646-0-1/06(A)