

## I<sup>2</sup>C® Download Protocol for ADuC70xxBCPZxxI Models

by Aude Richard

### INTRODUCTION

A key feature of the ADI MicroConverter® product family is the ability of the devices to download code to on-chip Flash/EE memory while in-circuit. On the standard ADuC702x models, the in-circuit code download feature is conducted over the device UART serial port.

On models containing the letter I at the end of the product number (ADuC7020BCPZ62I, for example), the code download feature is conducted over the device I<sup>2</sup>C serial port. This application note applies only to I models ADuC7019BCPZ62I, ADuC7020BCPZ62I, and ADuC7021BCPZ62I.

A Windows® executable program (**I2CWSD.exe**) is provided that allows the user to download code from a USB port via a third-party I<sup>2</sup>C pod to the MicroConverter. Schematics and code for the I<sup>2</sup>C pod can be found at [www.analog.com](http://www.analog.com); the pod can be purchased at [www.fh-pforzheim.de/stw-svs/texte/Dongle.html](http://www.fh-pforzheim.de/stw-svs/texte/Dongle.html). Note that any master host machine with an I<sup>2</sup>C master (microcontroller, DSP, or other) can download to the MicroConverter once the host machine adheres to the I<sup>2</sup>C download protocols detailed here.

This application note outlines in detail the MicroConverter I<sup>2</sup>C download protocol, allowing end users to both understand the protocol and, if required, to implement this protocol successfully in an end-target system (with an embedded host to an embedded MicroConverter).

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.

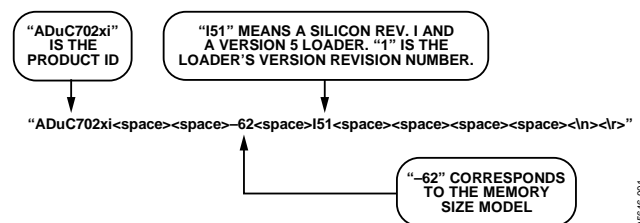


Figure 1. Product Identifier

05646-001

## RUNNING THE MICROCONVERTER LOADER

To allow an unattended download via I<sup>2</sup>C, the ADuC702x enters loader mode only if P0.0 (serial download) is low during reset and the content of Flash/EE memory at Address 0x14 is 0xFFFFFFFF. For P0.0 to determine if loader mode is entered, the user must ensure that the word at Flash Address 0x14 is 0xFFFFFFFF.

Alternatively, P0.0 can be kept low and entry to download mode can be determined by the content of Flash Address 0x14. Typically, the value at Flash Address 0x14 is not 0xFFFFFFFF; therefore, user code must have a built-in mechanism for 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 0x14 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 I<sup>2</sup>C0 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 (I<sup>2</sup>C write), and each command acknowledge request from the loader with the Byte 0x05 (I<sup>2</sup>C 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 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

## DEFINING THE DATA TRANSPORT PACKET FORMAT

Once the I<sup>2</sup>C 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 (0x07 and 0x0E). These bytes are constant and are used by the loader to detect a valid data packet start.

### 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. The list of data packet command functions is shown in Table 2.

### Address Field (Data 2 to Data 5)

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

### Data Byte Field 6 to Data Byte Field 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® HEX extended 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 by summing the hex values in the number of bytes field and the hex values in the Data 1 to Data 255 fields (if they exist). The checksum is the twos complement value of this summation: the 8-bit sum of the number of data bytes and Data Byte 1 to Data Byte 255. This can be expressed as follows:

$$CS = 0x00 - (Number\ of\ Data\ Bytes + \sum_{N=1}^{255} Data\ Byte_N)$$

Expressed differently, the 8-bit sum of all bytes excluding the start ID must be 0x00.

### Acknowledge of Command

After each command, the host must request an acknowledge from the loader for a negative response, a BEL (0x07), or a positive response, an ACK (0x06). A BEL is transmitted by the loader if it received 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 must ensure that any location where code is downloaded is erased. The recommended check for an error-free download is the Verify command.

Table 1. Data Transport Packet Format

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 to Data 5 (Address: h, u, m, l)	Data x (x = 6 to 255)	Checksum
0x07	0x0E	5 to 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 Acknowledge	Loader Negative Acknowledge
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	R (0x52)	ACK (0x06)	BEL (0x07)

Table 3. Erase Flash/EE Memory Command

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 to Data 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 to Data 5 (Address: h, u, m, l)	Data x (x = 1 to 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 to Data 5 (Address: h, u, m, l)	Data x (x = 1 to 250)	Checksum
0x07	0x0E	5 + x (6 to 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 to Data 5 (Address: h, u, m, l)	Data 6	Checksum
0x07	0x0E	0x06	P (0x50)	h, u, m, l	Type	CS

Table 7. Remote RUN Command

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 to Data 5 (Address: h, u, m, l)	Checksum
0x07	0x0E	0x05	R (0x52)	h, u, m, l = 0x1	0xA8

## DATA PACKET COMMAND FUNCTIONS

### Erase Command

The erase command is used to erase one page up to all pages of Flash/EE from a specific address determined by Data 2 to Data 5. This command also specifies the number of pages to erase. If the address is 0x00000000 and the number of pages is 0x00, the loader interprets it as a mass erase command, erasing 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 specifies the number of data bytes (which is Data 1 + Data 2 + Data 2 to Data 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 memory. 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 3-step sequence must be followed:

1. Initiate the command: type must be 0x00 and “huml” can be any value.
2. Send the address of the group of pages to protect. This step must be repeated for each group of pages. Type must be 0x0F.
3. Send the key in “huml”; type must be 0x01. FEEADR takes the value of “hu” and FEEDAT takes the value of “ml”. If no keys are required, “huml” must be 0xFFFFFFFF.

For example, to protect Page 0 to Page 7 against writing, set the read protection and use key 0x12345678. These are the commands that must be sent:

1. start sequence:  
0x07 0x0E 0x06 0x50 0xXXXXXXXX 0x00 CS
2. protection:  
0x07 0x0E 0x06 0x50 0x00000000 0x0F CS (Page 0 to Page 3)  
0x07 0x0E 0x06 0x50 0x00000200 0x0F CS (Page 4 to Page 7)  
0x07 0x0E 0x06 0x50 0x0000F800 0x0F CS (read protection)
3. key and end of sequence:  
0x07 0x0E 0x06 0x50 0x12345678 0x01 CS

Note that the protection command is only available in Revision 0 and later versions of the loader. In Revision 0, FEEADR = ml and FEEDAT = ml. In later versions, FEEADR = hu.

This protocol does not allow the Flash/EE memory to be unprotected. To remove the protection, use a mass erase command.

### 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 start executing code.

Two types of remote runs are implemented:

- A software reset, with h, u, m, l = 0x1
- A jump to user code, with h, u, m, l = 0x0

Table 7 shows an example of a remote run or reset. Executing a software reset is recommended as it resets all peripherals, however, in certain cases where P0.0 is permanently grounded and Address 0x80014 is cleared, it could be necessary to use a jump to user code. Be aware that after a jump to user code the I<sup>2</sup>C peripheral MMRs do not contain default values.

Purchase of licensed I<sup>2</sup>C components of Analog Devices or one of its sublicensed Associated Companies conveys a license for the purchaser under the Philips I<sup>2</sup>C Patent Rights to use these components in an I<sup>2</sup>C system, provided that the system conforms to the I<sup>2</sup>C Standard Specification as defined by Philips.