1.0 INTRODUCTION :

One of the many features of the MicroConverter product family is the ability of the device to download code to its on-chip FLASH/EE program memory while 'in-circuit'. On the standard ADuC702x models, this in-circuit code download feature is conducted over the device UART serial port.
On the models containing the letter I at the end of the name (for example ADuC7020BCPZ62I), this in-circuit code download feature is conducted over the device I²C serial port. This application note is intended only for the use of the 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 web site at www.analog.com and the pod can be purchased from 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 in this technical note.

The objective of this technical note is therefore to outline 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 in the description below, 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.

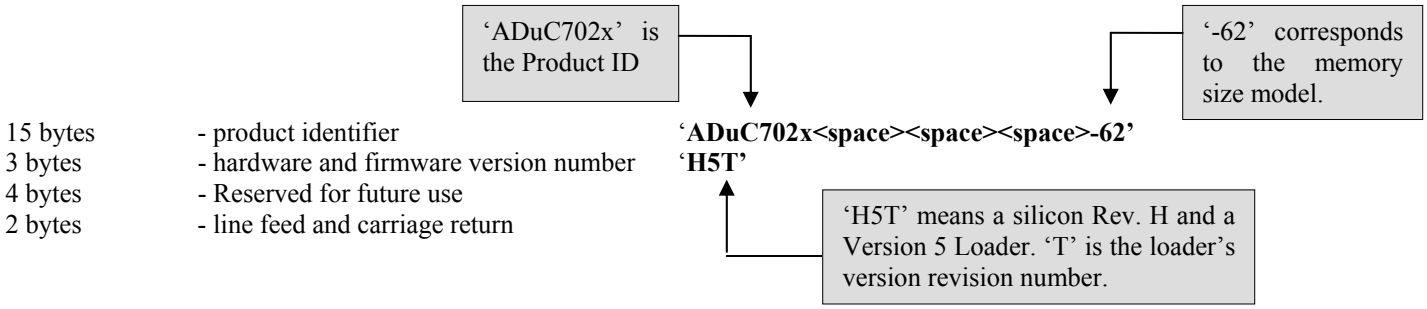
2.0 THE MICROCONVERTER LOADER :

2.1 RUNNING THE LOADER :

To allow unattended download via I²C the ADuC702x will only enter loader mode if P0.0 (serial download) is low during reset and the contents of Flash at address 0x80014 is 0xFFFFFFFF. So 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 will be determined by the contents of address 0x80014. Typically the value at address 0x80014 will not be 0xFFFFFFFF therefore user code should have a mechanism inbuilt 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 reentry to download mode, should power fail or another error occur during reprogramming of the bulk of the program.

2.2 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.



2.3 DEFINING THE DATA TRANSPORT PACKET FORMAT :

Once the reset string has been sent, the transfer of data can begin. The general communications data transport packet format is shown in table 1 below.

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data *x* (x = 6 → 255)	Checksum
07h	0Eh	5 → 255	'E', 'W', 'V' or 'R'	h, u, m, l	XX	No of DataBytes + Data 1 + Data 2 → 5 + Σ Data x (2's Comp)

Table 1: Data Transport Packet Format

Packet Start ID Field :
The first field is the 'Packet Start ID' field and 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 4 valid command functions are allowed. The four command functions are described by one of four ASCII characters, 'E', 'W', 'V' 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 re-assembled 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 two's complement checksum is calculated from the summation of the hex values in the No. of Bytes field and the hex values in the Data 1 to 255 fields (if they exist). The checksum is the two's complement value of this summation. i.e. the 8-bit sum of No. of data Bytes and Data Bytes 1-255 and the Checksum should equal 00h. This can also be expressed mathematically as:

$$\textbf{Checksum} = 00h - (\text{No. Data Bytes} + \sum_{N=1}^{255} \text{Data Byte}_N)$$

2. 4 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 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 full list of data packet command functions is shown in table 2.

Command Functions	Command Byte In Data 1 Field	Loader Positive Response	Loader Negative Response
Erase page	'E' (45h)	ACK (06h)	BEL (07h)
Write	'W' (57h)	ACK (06h)	BEL (07h)
Verify	'V' (56h)	ACK (06h)	BEL (07h)
Run (Jump to User code)	'R' (52h)	ACK (06h)	BEL (07h)

Table 2: Data Packet Command Functions

2.5 DATA PACKET COMMAND FUNCTIONS :

2.5.1 ERASE COMMAND

The erase command allows to erase from 1 page up to all pages of Flash/EE from a specific address determined by data 2 → 5. this command also requires the number of page 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.
The data packet for the erase command is shown in table 3.

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data 6 (pages)	Checksum
07h	0Eh	6	'E' (45h)	h, u, m, l	x pages	No. of DataBytes + Σ Data (2's Comp)

Table 3: Erase Flash/EE memory command

2.5.2 WRITE COMMAND

The write command requires the number of 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 will be programmed into Flash/EE. The loader will send 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.

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data x (x = 1 → 250)	Checksum
07h	0Eh	5 + No. of Data x (6 → 255)	'W' (57h)	h, u, m, l	…	No. of Data Bytes + Σ Data Bytes 1 - 515

Table 4: Program Flash/EE memory command

2.5.3 VERIFY COMMAND

The verify command is almost identical to the write command as shown in table 5. The command field is 'V' (56h) 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 as in the table below:

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

The loader restores the correct bit sequence and compares it to the flash contents. If it is correct and the checksum was correct, ACK (06h) is returned else BEL (07h) is returned.

Start ID		No. of Data Bytes	Data 1 CMD	Data 2 → 5 (Address: h, u, m, l)	Data x (x = 1 → 250)	Checksum
07h	0Eh	5 + No. of Data x (6 → 255)	'V' (56h)	h, u, m, l	Complemented data bytes	No. of Data Bytes + Σ Data Bytes 1 - 515

Table 5: Program Flash/EE memory command

2.5.4 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 5 below shows an example of a Remote RUN or 'jump to user code' from address 00h.

Packet Start ID		No. of Data Bytes	Data 1 (Command)	Data 2 → 5 (Address: h, u, m, l)	Checksum
07h	0Eh	05h	'R' (52h)	h, u, m, l	**A9h**

Table 5: Jump to user Code (Remote RUN) Command

Only run from start of the Flash/EE (h, u, m, l = 80000h) is supported at present.