AN-1074
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

Understanding the Serial Download Protocol (Formerly uC004)

INTRODUCTION

This application note describes the serial download protocol for the ADuC8xx family of products.

One of the many features of the MicroConverter® product family is the ability of the devices to download code to their on-chip Flash/EE program memory while in-circuit. This in-circuit code download feature is conducted over the device UART serial port and is thus commonly referred to as a serial download.

Serial download capability allows developers to reprogram the part that is soldered directly onto the target system. This avoids the need for an external device programmer as well as the requirement of removing the device to use the external programmer. Serial download also opens up the possibility of system upgrades in the field; all that is required is serial port access to the MicroConverter. This means that manufacturers can upgrade system firmware in the field without having to swap out the device.

Any MicroConverter device can be configured for serial download mode via a specific pin configuration at power-on or by application of the external reset signal. For the MicroConverter, the $\overline{\text{PSEN}}$ input pin is pulled low through a resistor (1 kΩ). In the case of the ADuC814, the DLOAD pin must be pulled high through a resistor (1 kΩ). If this condition is detected by the part at power-on or during application of a hard reset input, the part enters serial download mode.

In this mode, an on-chip resident loader routine is initiated, and the on-chip loader configures the device UART appropriately, and, via a specific serial download protocol, communicates with any host machine to manage the download of data into its Flash/EE memory spaces. Note that serial download mode operates within the nominal supply rating of the part and, as such, there is no requirement for a specific external high programming voltage since this is also generated on-chip. An example of the serial download for the ADuC812 MicroConverter in operation is shown in Figure 1.

As part of the Analog Devices, Inc., QuickStart™ development tool suite, a Windows® executable program (WSD.exe) is provided which allows the user to download code from the PC (PC serial ports COM1, COM2, COM3 or COM4) to the MicroConverter. Note, however, that any master host machine (PC, microcontroller, DSP, or other) can download to the MicroConverter once the host machine adheres to the serial download protocols detailed in this application note.

The objective of this application note is to outline in detail the MicroConverter serial 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.

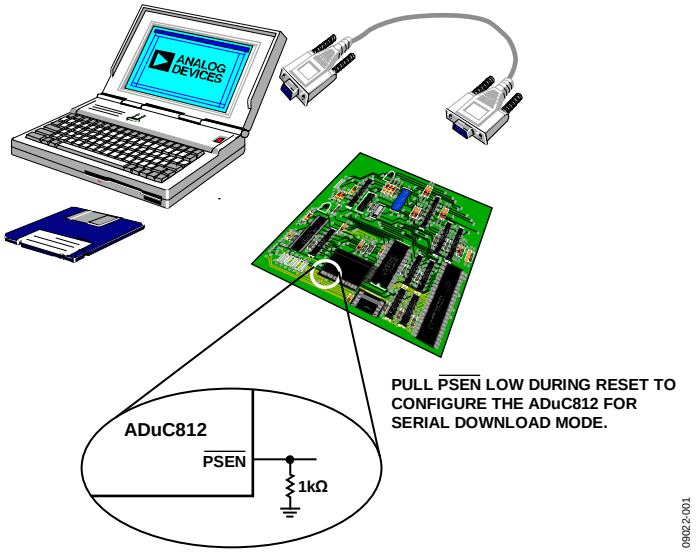

Figure 1. MicroConverter in Serial Download Mode

TABLE OF CONTENTS

BACKGROUND

The serial download protocol on any MicroConverter part is defined by the on-chip loader routine. The MicroConverter on-chip loader has undergone two revisions: Version 1 and Version 2.

Version 1 is on all ADuC812 silicon with branded date codes prior to 9933 (silicon shipped prior to August 1999).

Version 2 is on

- all ADuC812 silicon with branded date codes of 9933 or earlier (silicon shipped after August 1999)
- all ADuC814 silicon
- all ADuC816 silicon
- all ADuC824 silicon
- all ADuC831 and ADuC832 silicon
- all ADuC836 and ADuC834 silicon
- all ADuC841, ADuC842, and ADuC843 silicon
- all ADuC845, ADuC847, and ADuC848 silicon

The Version 1 loader protocol supports download of the raw Intel Hex format file directly to the MicroConverter (ADuC812 only) Flash/EE program memory space. The protocol as defined is straightforward, functional, and is detailed in the MicroConverter Version 1 Loader section.

The Version 2 loader supports a more comprehensive protocol allowing serial download to the Flash/EE program space or the Flash/EE data space. These features, as well as others, facilitate a more flexible and more secure use of the in-circuit download capability of the MicroConverter.

For the purposes of clarity in the descriptions in this application note, the term *host* refers to the host machine (PC, 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 VERSION 2 LOADER

The Version 2 loader is on the following MicroConverters:

- ADuC812 silicon with branded date codes of 9933 or earlier (silicon shipped after August 1999)
- ADuC814 silicon
- ADuC816 silicon
- ADuC824 silicon
- ADuC831 and ADuC832 silicon
- ADuC836 and ADuC834 silicon
- ADuC841, ADuC842, and ADuC843 silicon
- ADuC845, ADuC847, and ADuC848 silicon

RUNNING THE LOADER

The loader on the MicroConverter is run by pulling the $\overline{\text{PSEN}}$ pin low (or DLOAD high for the ADuC814) through a resistor (typically 1 kΩ pull-down) and resetting the part (either toggle the RESET input pin on the part itself or a power cycle resets the part). On reset or after a power cycle, the loader immediately sends the following 25-byte ID data packet.

The 25 bytes consist of

- 10-byte product identifier
  ADI<space>8xx<space><space><space>
  where:
  8xx is the product identifier (for example, 812 for the ADuC812).
- 4-byte firmware version number (for example, V2xx for Version 2 of the loader. xx is the version revision number).
- 2-byte line feed and carriage return.
- 2-byte hardware configuration.
- 6-bytes reserved for future use.
- 1-byte twos complement checksum of the first 24 bytes.

THE PHYSICAL INTERFACE

Once triggered, the loader configures the MicroConverter UART serial port to transmit/receive at a baud rate of 9600, 8 data bits, and no parity.

For the ADuC812, ADuC831, and ADuC841, the baud rate is derived indirectly from the master clock frequency; that is, a 9600 baud rate is set based on a master clock frequency of 11.0592 MHz. If the master clock frequency is increased or decreased, the baud rate increases or decreases according to the relevant baud rate equation given in the part data sheet. For the ADuC812, the equation is as follows:

$$Baud\ Rate = 9600\ \text{Baud} \times Crystal\ Freq/11.0592\ \text{MHz}$$

For example, if the master clock frequency is set to 1 MHz, then the loader configures the UART at 868 baud.

The PC program WSD.exe, discussed in the Windows Serial Downloader (WSD.exe) section, allows the user to input the target master clock rate and, thus, reconfigure the PC baud rate to match that of the loader.

Note that the ADuC841 only operates in download mode with a crystal with a value less than 16.0 MHz or, alternately, a 20 MHz crystal. Values between 16.0 MHz and 20 MHz will not function correctly for serial download.

For the ADuC816, ADuC824, ADuC832, ADuC834, ADuC836, ADuC842, ADuC843, ADuC845, ADuC847, and ADuC848, the baud rate is configured for 9600 baud assuming a 16.777216 MHz for SAR ADC parts (ADuC814, ADuC832, ADuC842, and ADuC843) and a 12.583 MHz core frequency for all Σ-Δ parts (ADuC834, ADuC836, ADuC845, ADuC847, and ADuC848). If a 32.768 kHz watch crystal is present, then the PLL automatically locks to these frequencies (16.777216 MHz or 12.58 MHz). If the crystal is not present, then the PLL cannot be guaranteed to lock to this frequency. The WSD.exe program allows the user to change the baud rate to suit the clock generated in this circumstance. If, in the absence of a crystal, the PLL lock frequency is measured, then the actual baud rate can be derived from the following (assuming a 12.58 MHz part):

$$Baud\ Rate = 9600\ \text{Baud} \times PLL\ Lock\ Freq/12.583\ \text{MHz}$$

INTERROGATING THE LOADER

The loader should be interrogated first to verify that the loader is correctly present and that the loader firmware version number is as expected.

Version 2 of the loader can be interrogated at anytime (as long as the MicroConverter loader is running) by sending the following interrogation data packet to the MicroConverter:

<0x21> <0x5A> <0x00> <0xA6> or '!' 'Z' <0x00> <Checksum>

The MicroConverter loader responds immediately by sending its 25-byte ID data packet.

Version 1 of the loader expects to see the ! character before responding with the ID data packet. The interrogation packet should, therefore, wait after sending the ! character to see if there is any response before transmitting the Z and the remainder of the interrogation string. If there is no response from the ! character alone, then the loader is a Version 2 loader.

The 25 bytes consist of

- 10-byte product identifier, such as ADI_812_ _ _.
- 4-byte firmware version number, such as V201.
- 2-byte line feed and carriage return.
- 2-byte hardware configuration.
- 6-bytes reserved for future use.
- 1-byte twos complement checksum of the first 24 bytes.

DEFINING THE DATA TRANSPORT PACKET FORMAT

Once the host has confirmed the presence of the loader, the transfer of data can begin. The general communications data transport packet format is shown in Table 1.

Table 1. Data Transport Packet Format

Packet Start ID	No. of Data Bytes	Data 1 Command Function	Data X[1]	Checksum
0x07 and 0x 0E	1 to 25 Decimal	C, A, W, V, Q, E, S, B, U, T,F	xxx	0x100 – *No. of Data Bytes* + Σ*Data X*

[1] X = Data 2 to Data 25.

Packet Start ID Field

The first field, the Packet Start ID field, contains two start characters (0x07 and 0x0E). These bytes, which are constant, are used by the loader to detect a valid data packet.

No. of Data Bytes Field

The next field is the total number of data bytes, including Data 1 (the command function). The maximum number of data bytes allowed is 25, but can vary from 1 to 25 depending on the data packet being transmitted.

Data 1 Command Function Field

The Command Function field describes the function of the data packet. One of eleven valid command functions are allowed. Note that some functions (such as erase commands) require that no additional data be transmitted. The eleven command functions are described by one of the following ASCII characters: C, A, W, V, Q, E, S, B, U, T or F.

Data X Field

The data bytes to be downloaded are contained in these fields. For either of the erase commands, these data bytes will not exist. In the case of either of the Program to Flash/EE memory commands, this data must be stripped out of the standard Intel 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 to this field. The twos complement checksum is calculated from the summation of the hexadecimal values in the No. of Data Bytes field and the hex values in the Data 1 and Data 2 to Data X fields (if they exist). The checksum is the twos complement value of this summation. This means that the sum of the No. of Data Bytes, Data 1, Data X, and the Checksum fields should equal 0x100. This can also be expressed mathematically as

$$\text{Checksum} = 0x100 - (\textit{No. of Data Bytes} + \Sigma \textit{Data Bytes}_N)$$

The loader routine responds with a NAK (0x07) as a negative response or with an ACK (0x06) as positive response to the previous data packet communication. A NAK is transmitted by the loader under the following conditions:

- Incorrect data packet format on verification of the checksum byte.
- Loader fails to verify that data has been programmed correctly.
- Request for loader to program data to a location that has not been previously erased.

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

Table 2. Data Packet Command Functions

Command Functions	Command Byte in Data Field	Loader Positive Response	Loader Negative Response
Erase Flash/EE Program Memory Only	C (0x43)	ACK (0x06)	NAK (0x07)
Erase Flash/EE Program and Data Memory	A (0x41)	ACK (0x06)	NAK (0x07)
Program block of Flash/EE Program Memory	W (0x57)	ACK (0x06)	NAK (0x07)
Read Page of Flash/EE Program Memory	V (0x56)	0x100 data bytes + CS	NAK (0x07)
Page Download of Flash/EE Program Memory	Q (0x51)	ACK (0x06)	NAK (0x07)
Program block of Flash/EE Data Memory	E (0x45)	ACK (0x06)	NAK (0x07)
Set Security Modes (All Parts Except ADuC812)	S (0x53)	ACK (0x06)	NAK (0x07)
Change Baud Rate (Timer 3 Enabled Part Only)	B (0x42)	ACK (0x06)	NAK (0x07)
Jump to User code (Remote RUN) Command	U (0x55)	ACK (0x06)	NAK (0x07)
Set ETIM Registers (ADuC812 only with V2.22 Kernel or Later)	T (0x54)	ACK (0x06)	NAK (0x07)
Enable Bootload Command (ULOAD)	F (0x46)	ACK (0x06)	NAK (0x07)

DATA PACKET COMMAND FUNCTIONS

Erase Command

As shown in Table 2, there are two erase commands. One of these erases both Flash/EE program and data memory (A) and the other erases only the Flash/EE program memory (C).

If the host wishes to download to the Flash/EE program memory or to the Flash/EE data memory, then these memory spaces must be erased first using the C (Erase Flash/EE program memory) or A (Erase Flash/EE program and data memory) command bytes. The loader responds with a NAK if the host attempts to download to a memory space that has not been first erased.

The erase commands do not require any additional data in the data packet format. An example of a data packet initiating erasure of the Flash/EE program and data memory is shown in Table 3. An example of a data packet initiating erasure of Flash/EE program memory only is shown in Table 4.

Table 3. Erase Flash/EE Program and Data Memory Command

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Checksum
0x07 and 0x0E	0x01	A (0x41)	0xBE 0x100 – *No. of Data Bytes* + Σ*Data Byte*

Table 4. Erase Flash/EE Program Memory Only Command

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Checksum
0x07 and 0x0E	0x01	C (0x43)	0xBC 0x100 – *No. of Data Bytes* + Σ*Data Byte*

Program Block of Flash/EE Program Memory

All of the download data packet commands require a start address. The address is contained in three bytes immediately following the command function data byte. The download data packets also contain the raw data to be downloaded.

The first data byte is written by the loader to the address specified in the data packet format. The loader then increments the address and writes the next byte, and so on, until all data bytes in the packet are written.

An example of a data packet, which programs eight locations in the Flash/EE program space, starting at Address 0x0000, is shown in Table 5.

Table 5. Program Block of Flash/EE Program Memory

Start ID	No. of Data Bytes	Data 1 CMD	Data 2 ADR U	Data 3 ADR M	Data 4 ADR L	Data 5 Prog 1	Data 6 Prog 2	Data 7 Prog 3	Data 8 Prog 4	Data 9 Prog 5	Data 10 Prog 6	Data 11 Prog 7	Data 12 Prog 8	Checksum
0x07and 0x0E	0x0C (12)	W (0x45)	0x00	0x00	0x00	0x00	0x0C	0x0E	0x0C	0x0F	0x0E	0x4F	0x63	0xBA

Read Page of Flash/EE Program Memory

The Read Page of Flash/EE program memory allows downloaded program memory to be read back and verified.

The Read Page of Flash/EE program memory command requires a page number. Each page consists of 0x100 bytes.

The loader responds with 0x100 bytes of data which correspond to the current contents of the specified page, followed by a checksum of that data.

A NACK response is returned if an erase command is not issued before this command in the download session. Therefore, it can only be used to read back code that has been downloaded in the same session and cannot be used to read back code from a previous download session.

An example of a data packet, which reads back 0x100 locations in the Flash/EE program space, starting at address 0x100, is shown in Table 6.

Table 6. Read Page of Flash/EE Program Memory Command

Start ID	No. of Data Bytes	Data 1 CMD	Data 2 ADR U	Checksum
0x07 and 0x0E	0x02 (2)	V (0x56)	0x01	0xA7

Table 7. Read Page of Flash/EE Program Memory Response

Data 1	Data 2	…	…	Data 256	Checksum
DATA	DATA	…	…	DATA	CS

PAGEDOWNLOAD Command

The PAGEDOWNLOAD command downloads 0x100 bytes of data to a specified page. This command is referred to in other documentation as the QuickDownload command.

The PAGEDOWNLOAD command requires a page number. Each page consists of 0x100 bytes.

The PAGEDOWNLOAD data packets contain the 0x100 bytes of data to be downloaded to the specified page. The Prog 1 byte is written by the loader to the first address of the page specified in the data packet format. The loader then increments the address and writes the next byte, and so on, until all data bytes in the packet are written

A NACK response is returned if the PAGEDOWNLOAD command is not supported. The Program block of Flash/EE program memory can be used if the PAGEDOWNLOAD command is not supported.

An example of a data packet, which will write 0x100 locations in the Flash/EE program space, starting at address 0x100, is shown in Table 8.

Program Block of Flash/EE Data Memory

As can be seen from any of the MicroConverter data sheets, the 640 byte Flash/EE data space must be programmed in 4-byte pages (this 640 byte space is configured as 160 pages). An example of a data packet that programs Page 5 in the Flash/EE data space is shown in Table 9.

Table 8. Page Download Command

Start ID	Data 1 CMD	Data 2 ADR U	Data 3 Prog 1	Data 4 Prog 2	Data 5 Prog 3	Data 6 Prog 4	…	Data 258 Prog 256	Checksum
0x07	Q (0x51)	0x01					…		CS

Table 9. Program Block of Flash/EE Data Memory

Start ID	No. of Data Bytes	Data 1 CMD	Data 2 ADR U	Data 3 ADR M	Data 4 ADR L	Data 5 Prog 1	Data 6 Prog 2	Data 7 Prog 3	Data 8 Prog 4	Checksum
0x07and 0x0E	0x08	E (0x45)	0x00	0x00	0x05	0x0A	0x0B	0x0C	0x0D	= 0x80 in this example =0x100 – *No. of Data Bytes* + *ΣData Bytes*

Set Security Modes Command

Three security options exist for all parts except the ADuC812. Refer to the device data sheet for details of what each security mode secures. Either of the erase commands returns the security mode to no security. Thus, after every download, the security modes have to be set again. To set a security feature, send the appropriate Data Byte 2 after the command byte (S) for the corresponding security option as shown in Table 10.

Table 10. Required Data 2 Byte for Different Security Modes

Security Mode	Data 2 Byte
Lock Mode	0x06
Secure Mode	0x05
Secure and Lock Mode	0x04
Serial Safe Mode	0x03
Serial Safe and Lock Mode	0x02
Serial Safe and Secure Mode	0x01
Serial Safe, Secure, and Lock Mode	0x00

Serial safe mode disables the serial downloader. Thus, writing to this bit prevents future serial downloads from occurring. The only way to disable this security option is by initiating a code erase command in parallel programming mode.

Secure mode actually contains all the security of lock mode. Thus, in terms of security, secure and lock mode is actually the same as secure mode.

To set a security mode, the host sends a data packet as shown in Table 11.

Table 11. Example of Setting the Secure Mode Command

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2	Checksum
0x07 and 0x0E	0x02	S (0x53)	0x05	0xA6 0x100 – *No. of Data Bytes* +Σ*Data Byte*

Change Baud Rate Command

A dedicated baud rate timer (Timer 3) is available to some of the ADuC8xx models to generate highly accurate baud rates for different crystal values. It is not available on the ADuC812, ADuC814, ADuC816, or ADuC824. The different values of T3CON and T3FD for the required baud rate are available in the relevant data sheets.

Table 12. Example of Setting the Baud Rate Change Command

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2	Data 3	Checksum
0x07 and 0x0E	0x03	B (0x42)	T3CON	T3FD	0x100 – *No. of Data Bytes* + Σ*Data Byte*

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. This executes the code that has just been downloaded. Table 14 shows an example of a Remote RUN or a jump to user code from Address 0x00.

RAM After a Remote RUN

The ADuC812 has a clear internal RAM function called during the power-on configuration routine. Thus, after a reset (to run user code), the internal RAM appears as all 0x00. However, if a remote RUN is used to run user code after a serial download, then the clear RAM routine is called and the RAM is preserved (that is, the RAM remains as it was before the serial download) except for the following locations, which are corrupted by the loader:

0x00 to 0x02, 0x05 to 0x0D, 0x2A, and 0x33 to 0x47

Hardware resetting all subsequent parts to run user code preserves the internal RAM. However, after these parts have been reset into the Version 2 loader (that is, WSD or MS-DOS downloader) some bytes of internal RAM are corrupted.

SFRs After a Remote Run

Hardware resetting the MicroConverter to run user code loads all the SFRs to their default values (see the SFR table in the appropriate data sheet). However, using a remote run after serially downloading to the MicroConverter leaves the UART configured at 9600 baud. Table 13 show the SFRs that come up at nondefault values. All other SFRs come up at their default values after a remote run.

Table 13. Initial Values of SFRs After a Remote Run

SFR Corruption Following Remote Run on:		
ADuC812	**ADuC816/ADuC824**	**Other ADuC8x Parts**
TH1	TH2	T3CON
TL1	TL2	T3FD
SCON	SCON	SCON
TCON	T2CON	SBUF
TMOD	RCAP2H	
SBUF	SBUF	
	RCAP2L	

Table 14. Jump to User Code (Remote RUN) Command

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2 ADR U	Data 3 ADR M	Data 4 ADR L	Checksum
0x07 and 0x0E	0x04	U (0x55)	0x00	0x00	0x00	= 0xA7 in this example = 0x100 – *No. of Data Bytes* + Σ*Data Bytes*

Set ETIM Registers

Setting ETIM registers only works on theADuC812 Version 2.22 kernel or later.

On the ADuC812, Flash/EE erase and program timing is derived from the master clock. When using a master clock frequency of 11.0592 MHz, it is not necessary to write to the ETIM registers. However, when operating at other master clock frequencies, one must change the values of ETIM1 and ETIM2 to avoid degrading data Flash/EE endurance and retention. ETIM1 and ETIM2 form a 16-bit word. ETIM2 is the high byte and ETIM1 is the low byte. The value of this 16-bit word must be set to ensure optimum data Flash/EE endurance and retention. This command only sets the ETIM registers for the download duration, after which point these registers are reset to their defaults. When using a master clock frequency other than 11.0592 MHz, program these registers as part of the program code.

For the ADuC812 the equation is

$$ETIM2, ETIM1 = 100\ \mu s \times f_{clk}$$

ETIM3 should always remain at its default value of 201 decimal/C9 hexadecimal.

Table 17 shows an example with respect to a 12 MHz crystal where 100 µs × 12 MHz = 1200 decimal = 0x04B, therefore ETIM2 = 0x04 and ETIM1 = 0xB0.

All other parts (ADuC814, ADuC831, ADuC832, ADuC834, ADuC836, ADuC841, ADuC842, ADuC843, ADuC845, ADuC847, and ADuC848) automatically set the required ETIM values depending on the core frequency in use. The user does not need to edit these registers.

Bootload Command

A bootload enable option exists in the serial downloader to always run from 0xE000 after reset. If using a bootloader, this option is recommended to ensure that the bootloader always executes correct code after reset. Programming the Flash/EE program memory via ULOAD mode is described in the product data sheet and in the uC007 documentation available online.

Table 15. Example of Setting the Enable Bootload Command

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2	Checksum
0x07 and 0x0E	0x02	F (0x46)	0XFE	0x100 – *No. of Data Bytes* + Σ*Data Byte*

Table 16. Example of Setting the Disable Bootload Command

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2	Checksum
0x07 and 0x0E	0x02	F (0x46)	0XFF	0x100 – *No. of Data Bytes* + Σ*Data Byte*

Summary Download Sequence

A program can now be downloaded from the host to the MicroConverter using the Version 2 loader. Figure 2 shows a flowchart for a typical download sequence.

Table 17. Programming the ETIM1 and ETIM2 Registers

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2 ETIM1	Data 3 ETIM2	Data 4 ETIM3	Checksum
0x07 and 0x0E	0x04	T (0x54)	0xB0	0x04	0xC9	= 0x100 – *No. of Data Bytes* + Σ*Data Bytes*

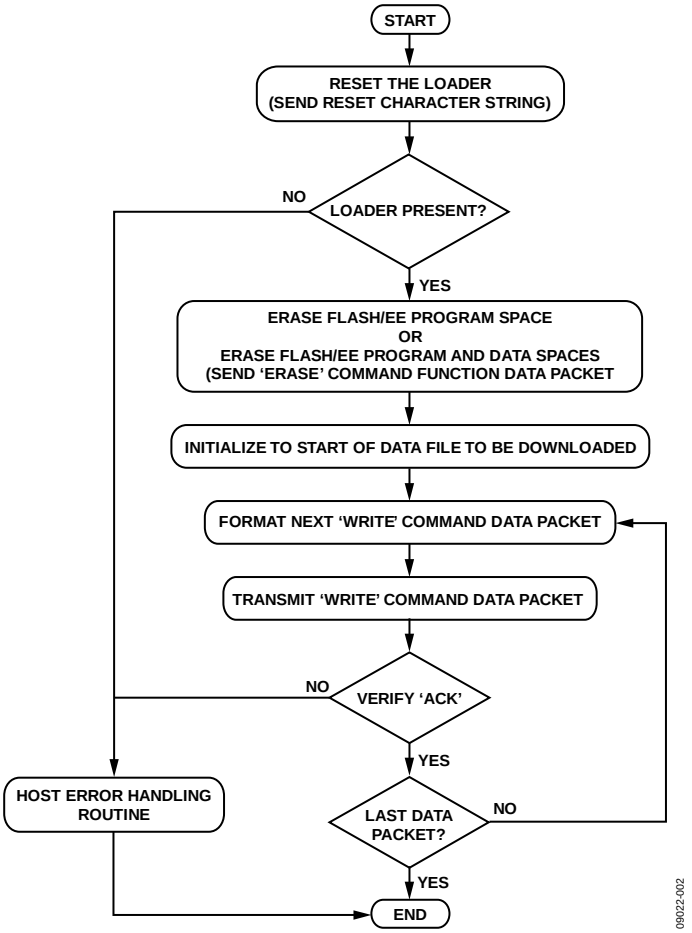

Figure 2. Flowchart for a Typical Data Download Sequence

MICROCONVERTER VERSION 1 LOADER

The Version 1 loader is available on some early ADuC812 MicroConverters only.

This loader Version is provided on all ADuC812 silicon with a branded date code of less than 9933 (silicon shipped prior to August 1999).

RUNNING THE LOADER

As with the Version 2 loader, the Version 1 loader on the ADuC812 is initiated by pulling the $\overline{\text{PSEN}}$ pin low through a resistor (typically a 1 kΩ pull-down resistor) and resetting the part. To reset the part, toggle the RESET input pin; a power cycle also resets the part. On reset or after a power cycle, the loader erases the Flash/EE program and data memory and then immediately sends the following 11-byte ID data character string:

- 8-byte product identifier: ADuC812<space>
- 3-byte firmware version number: krl

THE PHYSICAL INTERFACE

Once triggered, the loader configures the ADuC812 UART serial port to transmit/receive in the mode.

9600 baud rate, 8 data bits, no parity

For the ADuC812, the baud rate is derived indirectly from the master clock frequency; that is, a 9600 baud rate is set based on a master clock frequency of 11.0592 MHz. If the master clock frequency is increased or decreased, the baud rate increases or decreases as follows:

Baud Rate = 9600 Baud × *Crystal Freq.*/11.0592 MHz

For example, if the master clock frequency is set to 1 MHz, then the loader configures the UART at 868 baud.

The PC program WSD.exe allows the user to input the target master clock rate and thus reconfigure the PC baud rate to match that of the loader.

INTERROGATING THE LOADER

The loader should be interrogated to verify that the loader is correctly present and that the loader firmware version number is as expected.

The Version 1 loader can be interrogated anytime (as long as the MicroConverter loader is running) by sending the following data packet to the MicroConverter:

Interrogation Character <21 hex> (ASCII '!')

The MicroConverter loader responds immediately by sending an 11-byte ID data character string.

- 8-byte product identifier: ADuC812<space>
- 3-byte firmware version number: krl

DOWNLOADING CODE TO THE VERSION 1 LOADER

Unlike Version 2, the Version 1 loader supports direct download to Flash/EE program space only. The host must transmit an unedited version of the standard Intel Hex format program file; the loader expects to see this transmitted directly as part of the download sequence.

The Version 1 loader automatically erases Flash/EE program and data spaces as soon as the loader is enabled and before transmitting the 11-byte ID data character string. Note that the loader does not support data download to the Flash/EE data memory space.

Once the loader is interrogated, the loader waits for the transmission of an Intel Hex file, which it will program into the Flash/EE program memory space. It receives this file on a per record basis, expecting the host to adhere to the standard communications protocol. Note the format of the standard Intel Hex file is described in the Standard Intel Hex Format section.

The loader recognizes a record by the start character (:). Note that it uses the various bytes that precede the record, such as address and number of bytes in the record, as well as the checksum at the end of the standard record file format. Therefore, it is important that the host transmits the Intel Hex file as is.

At the end of each record, the MicroConverter transmits an ACK (everything OK) or a NACK (everything not OK). The ACK is 0x06 and NACK is 0x07. The user host should interpret these appropriately. For example, on ACK, move to transmit the next record and, on NACK, report an error message to the host. On an error, the MicroConverter waits for the next start character (:) so that the host can decide to stop on an error or try to retransmit the record.

RUNNING THE CODE

At the end of the file, the user host can force the code to execute by sending a start address command character string. A start address can be sent by first sending the semicolon character followed by a 4-byte start address. Note that in a real application if the host wants to download and follow the download by a run from the start address sequence, then this start address should be 0xFF00. (The start address 0xFF00 ensures that a resident power-on configuration routine runs to correctly calibrate the ADC internal voltage reference before normal user code runs from Address 0x0000.)

If you do not follow the download of the Intel Hex file with a semicolon, then the only way to force a start of code execution would be to remove the $\overline{\text{PSEN}}$ pull-down and force a reset or a new power cycle.

STANDARD INTEL HEX FORMAT

This section describes the Intel Hex standard file format expected by the Version 2 loader protocol. Intel Hexadecimal format, or Intel Hex format, is a standard for storing machine language in displayable or printable format.

A standard Intel Hex file is generated by assembling a MicroConverter (8051-compatible source code) program. An 8051-compatible assembler, the MetaLink two-pass assembler, is available as part of the QuickStart development system or as a free download executable from the MicroConverter website.

An Intel Hex file is a series of lines or hex records containing the fields shown in Table 18. These fields are shown more explicitly in Figure 3, which illustrates a typical example print-out from an Intel Hex file.

Table 18. Summary of the fields of Intel Standard Hex Files

Field	Bytes	Description
Record Mark	1	: indicates the start of record.
Record Length	2	Number of data bytes in record.
Load Address	4	Starting address for data bytes.
Record Type	2	00 = data record, 01 = end record.
Data Bytes	0 to 16	Data.
Checksum	2	Sum of all bytes in record + checksum = 0.

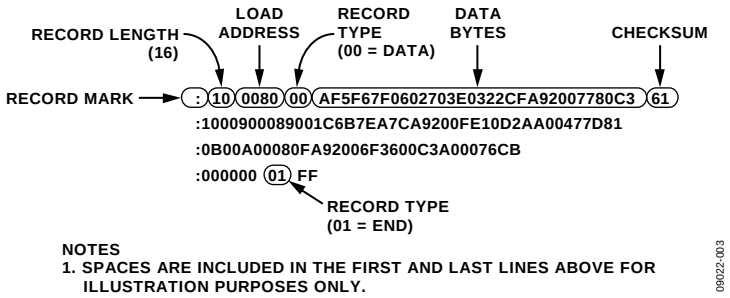

Figure 3. Intel Hex Format

WINDOWS SERIAL DOWNLOADER (WSD.EXE)

The Windows serial downloader (WSD) is a Windows software program that allows a user to serially download Intel standard Hex files as created by the ASM51 assembler. The WSD works with a Version 1 or a Version 2 loader automatically. The Intel standard Hex file is downloaded into the on-chip program Flash memory via any of the serial ports (COM1 to COM32) on a standard PC.

The configuration window allows the user to configure the download and run options. Once the Download button is pressed on the WSD, then the download options as set in the configuration window are sent to the MicroConverter.

DOWNLOAD AND RUN OPTIONS AVAILABLE

The following download and run options are available to the user in the configuration window (see Figure 4):

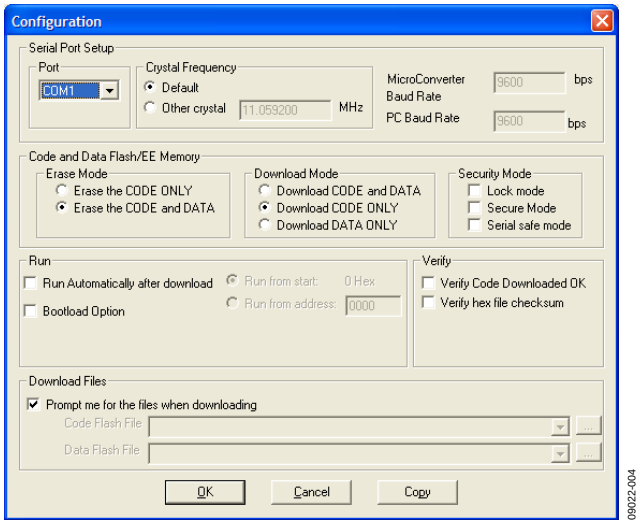

Figure 4. WSD Configuration

Port. The user can select the required PC COM port.

Crystal Frequency. The user selects the device operating frequency.

Erase Mode. The user can choose to erase the Flash/EE program memory (**Erase the CODE ONLY**) or erase both the Flash/EE program and data memory (**CODE and DATA**).

Download Mode. The user can choose to program the Flash/EE program and Data memory (**Download CODE and DATA**), program the Flash/EE data memory **(Download CODE ONLY),** or program the data memory only **(Download DATA ONLY).**

Security Mode. For the ADuC816, ADuC824, ADuC834, ADuC836, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843, ADuC845, ADuC847, and ADuC848 three security modes are available. The user can choose to set any combination of these security modes.

Run. The user can choose to

- Run from the start (0 hex).
- Run from a particular address entered in hexadecimal format.
- Run automatically after a download.
- Run from the bootload option (that is, from Address 0xE000). Refer to the MicroConverter Technical Note (uC007) User Download (ULOAD) Mode documentation on the Analog Devices website for more information on ULOAD mode.

With the first two Run entries, the code does not run until the user clicks the **Run** button on the WSD. For the third entry, the run command is automatically sent down to the Micro-Converter after the download is complete.

Verify. The user can verify the downloaded code.

COMMAND LINE INTERFACE TO THE WINDOWS SERIAL DOWNLOADER

The WSD software can be driven from a command line interface also. To download a file, enter **WSD <file.hex>.** This downloads the program called file.hex.

Additional Downloader Options

To make use of the additional downloader options, enter any of the following switches immediately following the file.hex> name. Precede each switch with a space.

- **/C:n** to select the COM port required (default is 1).
- **/D** to download to Flash/EE program memory without erasing the Flash/EE data memory.
- **/R** to run the program from the beginning (that is, 0x00).
- **/R:xxxx** to run the program from the specified (xxxx) hex start address.
- **/V** to verify code download (this only works with ADuC83x and ADuC84x parts).
- **/X:freq** to set the crystal frequency value (only for ADuC812, ADuC831, and ADuC841).
- **/B** for the bootload option (this runs from 0xE000).
- **/M:n** for download mode where:
  0 = download code and data (Data file must be second file name specified).
  1 = download code only.
  2 = download data only. (The data file must be the second file name specified. The program hex file will not be downloaded.)
- **/S:n** for security mode. (All parts except the ADuC812.)
  0 = lock mode.
  1 = secure mode.
  2 = serial safe mode.
  3 = all security bits set.
- **/T:n** for the time the program remains open where n is an integer from 0 to 9 seconds.

NOTES

NOTES

AN09022-0-7/10(0)