a

MicroConverter® Technical Note - uC004
Understanding the Serial Download Protocol

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'. This in-circuit code download feature is conducted over the device UART serial port and is thus commonly referred to as 'serial download'. Serial download capability allows developers to re-program the part while it is soldered directly onto the target system avoiding the need for an external device programmer and the requirement for having to remove 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 application of the external reset signal. For the MicroConverter, the PSEN input pin is pulled low through a resistor (1kΩ). In the case of the ADuC814 the DLOAD pin must be pulled High through a resistor (1k?). If this condition is detected by the part on power-on or during application of a hard Reset input then the part will enter serial download mode. In this mode an on-chip resident loader routine is initiated, the on-chip loader configures the device UART appropriately and, via a specific serial download protocol will communicate with any host machine to manage the download of data into its FLASH/EE memory spaces. It should be noted 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, as this is also generated on-chip. A graphical example of the serial download for the ADuC812 MicroConverter in operation is shown in Figure 1 below.

As part of Analog Devices' QuickStart development tool-suite, a windows executable program is provided (WSD.exe) which allows the user to download code from the PC (PC serial ports COM1, 2, 3 or 4) to the MicroConverter. It should however be emphasized 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 technical note.

The objective of this technical note is therefore 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.

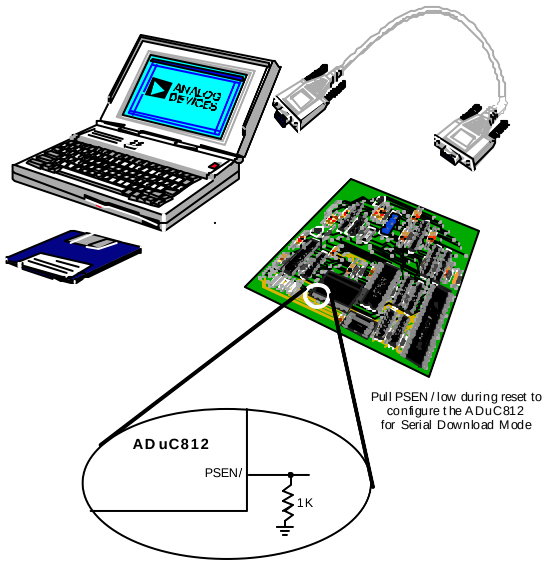

ANALOG DEVICES
Pull PSEN/ low during reset to configure the ADuC812 for Serial Download Mode
ADuC812
PSEN/
1K

Figure 1. MicroConverter in Serial Download Mode

Figure 1. MicroConverter in Serial Download Mode

The serial download protocol on any MicroConverter part is defined by the on-chip loader routine. The MicroConverter on-chip loader has undergone 2 revisions, namely :

Loader Version 1: - is on all ADuC812 silicon with branded date code < 9933 (silicon shipped prior to August 1999)

Loader Version 2: - is on all ADuC812 silicon with branded date code >= 9933 (silicon shipped after August 1999)
- is on all ADuC814 silicon
- is on all ADuC816 silicon
- is on all ADuC824 silicon
- is on all ADuC831 and ADuC832 silicon
- is on all ADuC836 and ADuC834 silicon
- is on all ADuC841, ADuC842 and ADuC843 silicon
- is on 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 straight forward, functional and is detailed in section 3.0.

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 and more, additional features facilitate both a more flexible and more secure use of the in-circuit download capability of the MicroConverter.

For the purposes of clarity in the description below, 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.

2.0 THE MICROCONVERTER VERSION 2 LOADER :

Note: The version 2 loader is on the following MicroConverters

Loader Version 2: - is on all ADuC812 silicon with branded date code >= 9933 (silicon shipped after August 1999)
- is on all ADuC814 silicon
- is on all ADuC816 silicon
- is on all ADuC824 silicon
- is on all ADuC831 and ADuC832 silicon
- is on all ADuC836 and ADuC834 silicon
- is on all ADuC841, ADuC842 and ADuC843 silicon
- is on all ADuC845, ADuC847 and ADuC848 silicon

2.1 RUNNING THE LOADER :

As mentioned previously, the loader on the MicroConverter is run by pulling the PSEN pin low (DLOAD High for ADuC814) through a resistor (typically 1kΩ pulldown) and resetting the part (toggling the 'RESET' input pin on the part itself or a power cycle will reset the part). On reset or after a power cycle the loader will immediately send the following 25 byte ID data packet.

'8xx' is the Product ID
eg - 812 for the ADuC812
- 824 for the ADuC824

10 bytes	- product identifier	**'ADI<space> 8xx<space><space><space>'**
4 bytes	- firmware version number	**'V2xx'**
2 bytes	- line feed and carriage return	
2 bytes	- hardware configuration	
6 bytes	- Reserved for future use	
1 byte	- 2's complement checksum of the first 24 bytes	

'V2xx' means a Version 2 Loader. 'xx' is the version revision number.

2.2 THE PHYSICAL INTERFACE :

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

Note: For the **ADuC812, ADuC831, ADuC841** the baud rate is derived indirectly from the master clock frequency i.e. 9600 baud rate will be set based on a master clock frequency of 11.0592MHz. If the master clock frequency is increased or decreased the baud rate will increase or decrease according to the relevant Baud rate equation given in the ppart datasheet. For the ADuC812 the equation is as follows:

Baud rate = 9600baud x Crystal Freq/11.0592MHz

e.g. if the master clock frequency is set to 1MHz then the loader will configure the UART at 868baud.
The PC program WSD.exe, discussed in section 5.0 allows the user to input the target master clock rate and thus reconfigure the PC baud rate to match that of the loader.

Note: The ADuC841 will only operate in download mode with a crystal < 16.0MHz or alternately a 20MHz crystal. Values between 16.0MHz and 20MHz 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 16.777216MHz for SAR ADC parts (ADuC814, ADuC832 and ADuC842, ADuC843) and a 12.583MHz core frequency for all ? -? parts (ADuC834, ADuC836, ADuC845, ADuC847, ADuC848). If a 32.768kHz watch crystal is present then the PLL will automatically lock to these frequency (16.777216MHz or 12.58MHz). 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 12.58MHz part):

baud rate = 9600baud x PLL lock freq/12.583MHz

2.3 INTERROGATING THE LOADER :

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

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

Interrogation Data Packet : <21h> <5Ah> <00h> <A6h>
or ‘!’ ‘Z’ <00h> <Checksum>

The MicroConverter loader responds immediately by sending its 25 byte ID data packet mentioned earlier and detailed again below.

Note: The version 1 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. See section 4.2 for more details.

25 byte loader ID Data Packet :

10 bytes	- product identifier e.g. `‘ADI 812    ’`
4 bytes	- firmware version numbere.g. `‘V201’`
2 bytes	- line feed and carriage return
2 bytes	- hardware configuration
6 bytes	- Reserved for future use
1 byte	- 2’s complement checksum of the first 24 bytes

2.4 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 below.

Table 1. Data Transport Packet Format

Packet Start ID	# DataBytes	Data 1 (Command Function)	Data *x* (x = 2 → 25)	Checksum
07h and 0Eh	1 – 25 Dec	‘C’, ‘A’, ‘W’, ‘E’, ‘S’, ‘T’, ‘B’ or‘U’	xxx	100h - # DataBytes+ Σ Data x

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 maximum number of data bytes allowed is 25 but can vary from 1 to 25 depending on the data packet being transmitted.

Command Function Field :
The 'Command Function' field describes the function of the data packet. One of 6 valid command functions are allowed. It should be noted that some functions (eg. Erase commands) require that no additional data be transmitted. The six command functions are described by one of six ASCII characters, 'C', 'A', 'W', 'E', 'S', 'B' or 'U'.

Data Byte Fields 2 – 25 :
The data bytes to be downloaded are contained in these. 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 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 25 fields (if they exist). The checksum is the two's complement value of this summation. i.e. The sum of No. of data Bytes and Data Bytes 1-25 and the Checksum should equal 100h. This can also be expressed mathematically as:

$$\textbf{Checksum = 100h -} \ (\text{No. } \textbf{Data Bytes} + \sum_{N=1}^{25} \textbf{Data Byte}_N\,)$$

The loader routine will respond with a 'NAK' (07h) as a negative response or with an 'ACK' (06h) as positive response to the previous data packet communication. A 'NAK' will be 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.
- Loader is requested 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 below.

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' (43h)	ACK (06h)	NAK (07h)
Erase Flash/EE Program & Data Memory	'A' (41h)	ACK (06h)	NAK (07h)
Program block of Flash/EE Program Memory	'W' (57h)	ACK (06h)	NAK (07h)
Program block of Flash/EE Data Memory	'E' (45h)	ACK (06h)	NAK (07h)
Set **Security** Modes(all parts except the ADuC812)	'S' (53h)	ACK (06h)	NAK (07h)
Change Baud Rate (Timer 3 enabled part only)	'B' (42h) + T3CON + T3FD	ADK (06h)	NAK (07h)
Jump to User code (Remote **RUN)** Command	'U' (55h)	ACK (06h)	NAK (07h)
Set **ETIM** Registers (ADuC812 only with V2.22 Kernel or Later)	'T'(54h)	ACK (06h)	NAK (07h)
Change Baud Rate to Fast load (ULOAD)	'B' (42h) + <T3CON> + <T3FD>	ACK (06h)	NAK (07h)
Enable BootLoad Command (ULOAD)	46h + FEh (for Bootload enabled) 46h + FFh (for Bootload disabled)	ACK (06h)	NAK (07h)

An example of a Baud Rate change is <07h><0Eh><03h><'B'><T3CON><T3FD><CS>.

An example of a Bootload enable command is : <07h><0Eh><02h><46h><FEh><BA h>

And for Bootload disable it would be : <07h><0Eh><02h><46h><FFh><B9h>

2.5 DATA PACKET COMMAND FUNCTIONS :

2.5.1 ERASE COMMAND

As shown in Table 1 above there are two ERASE commands. One of these erases both flash/EE program and data memory ('A') and other erases only the Flash/EE program memory only ('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 will respond 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 Flash/EE program and data memory is shown in table 3(a) below. An example of a data packet initiating erasure of Flash/EE program memory only is shown in table 3(b) below.

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Checksum
07h and 0Eh	01h	'A' (41h)	**BEh** 100h - # Data Bytes + Data Byte

Table 3. Erase flash/EE program and data memory command:

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Checksum
07h and 0Eh	01h	'C' (43h)	**BCh** 100h - #Data Bytes + Data Byte

Table 4. Erase flash/EE program memory only command:

2.5.2 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 preceding 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 have been written. An example of a data packet, which will program 8 locations in the Flash/EE program space, starting at address 0000H, is shown in Table 4 below.

Start ID	No. of Data Bytes	Data 1 CMD	Data2 ADR U	Data3 ADR M	Data4 ADR L	Data5 Prog 1	Data6 Prog 2	Data7 Prog 3	Data8 Prog 4	Data9 Prog 5	Data10 Prog6	Data11 Prog7	Data12 Prog8	Checksum
07h and 0Eh	0Ch (12)	'W' (45h)	00h	00h	00h	00h	0Ch	0Eh	0Ch	0Fh	0Eh	4Fh	63h	**BAh**

Table 4: Program block of Flash/EE Program Memory

2.5.3 PROGRAM BLOCK OF FLASH/EE DATA MEMORY

As can be seen from any of the MicroConverter datasheets, 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, which will program page 5 in the Flash/EE data space, is shown in Table 5 below.

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
07h and 0Eh	08h	'E' (45h)	00h	00h	05h	0Ah	0Bh	0Ch	0Dh	**80h** 100h - # Data Bytes + Σ Data Bytes

Table 5. Program block of Flash/EE Data Memory

2.5.4 SET SECURITY MODES COMMAND

Three security options exist for all parts except the ADuC812. Check out the datasheet for details of what each security mode secures. Either of the erase commands return the security mode to NO SECURITY. Hence after every download the security modes have to be set again. To set a security feature you send the appropriate data byte 2 after the command byte ('S') for the corresponding security option as shown in table 6 below.

Security Mode	Data 2 Byte
LOCK Mode	06h
SECURE Mode	05h
SECURE and LOCK mode	04h
SERIAL SAFE Mode	03h
SERIAL SAFE and LOCK mode	02h
SERIAL SAFE and SECURE mode	01h
SERIAL SAFE, SECURE and LOCK mode	00h

Table 6. Required Data 2 Byte for the different Security Modes

Note: SERIAL SAFE mode disables the serial downloader. Hence writing to this bit will prevent future serial downloads from happening. 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. Hence, 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 7 below.

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2	Checksum
07h and 0Eh	02h	'S' (53h)	05h	**A6h** (100h - # Data Bytes + Data Byte)

Table 7. Example Setting the Secure Mode Command

2.5.5 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 8 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 ADR U	Data 3 ADR M	Data 4 ADR L	Checksum
07h and 0Eh	04h	'U' (55h)	00h	00h	00h	**A7h** 100h - # Data Bytes + Σ Data Bytes

Table 8. Jump to user Code (Remote RUN) Command

RAM after a remote RUN
The ADuC812 has a clear internal RAM function called during the power-on-configuration routine. Hence after a reset (to run user code) the internal RAM comes up as all 00h. However if a remote RUN is used to run users code after a serial download then the clear RAM routine will not be called and the RAM will be preserved (i.e. the RAM will be as it was before the serial download) except for the following locations which will be corrupted by the loader:
00h→02h, 05h→0Dh, 2Ah, and 33h→47h

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 (i.e. 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 SFR table in datasheet). However using a remote RUN after serially downloading to the MicroConverter leaves the UART configured at 9600 baud. Table 9 show the SFRs which come up at non default values. All other SFRs come up at their default values after a remote RUN.

SFR corruption following remote run on ADuC812	SFR corruption following remote run on ADuC816/ADuC824	SFR corruption following remote run on all subsequent parts
TH1	TH2	T3CON
TL1	TL2	T3FD
SCON	SCON	SCON
TCON	T2CON	
TMOD	RCAP2H	
SBUF	SBUF	SBUF
	RCAP2L	

Table 9. Initial values of SFRs after a remote RUN.

2.5.6 Set ETIM Registers (Only works on ADuC812 V2.22 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 11.0592MHz it is not necessary to write to the ETIM registers at all. However, when operating at other master clock frequencies, you 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 being the high byte and ETIM1 being 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 this these registers are reset to their default and if the user is using a master clock frequency other than 11.0592MHz they should program these registers as part of their program code.
For the ADuC812 the equation is....
ETIM2, ETIM1 = 100 μs x f_{clk}

ETIM3 should always remain at its default value of 201 dec/ C9 hex.

Packet Start ID	No. of Data Bytes	Data 1 (Command)	Data 2 ETIM1	Data 3 ETIM2	Data 4 ETIM3	Checksum
07h and 0Eh	04h	‘T’ (54h)	B0h	04h	C9h	No. of Data Bytes + Σ Data Bytes1 - 4 (2’sComp)

Table 10. Programming the ETIM1 and ETIM2 registers

The above example would be for the case of a 12MHz crystal.....100us x 12MHz = 1200 (dec) = 04B0h, therefore ETIM2 = 04h and ETIM1 = B0h.

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

2.5.7 SUMMARY DOWNLOAD SEQUENCE

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

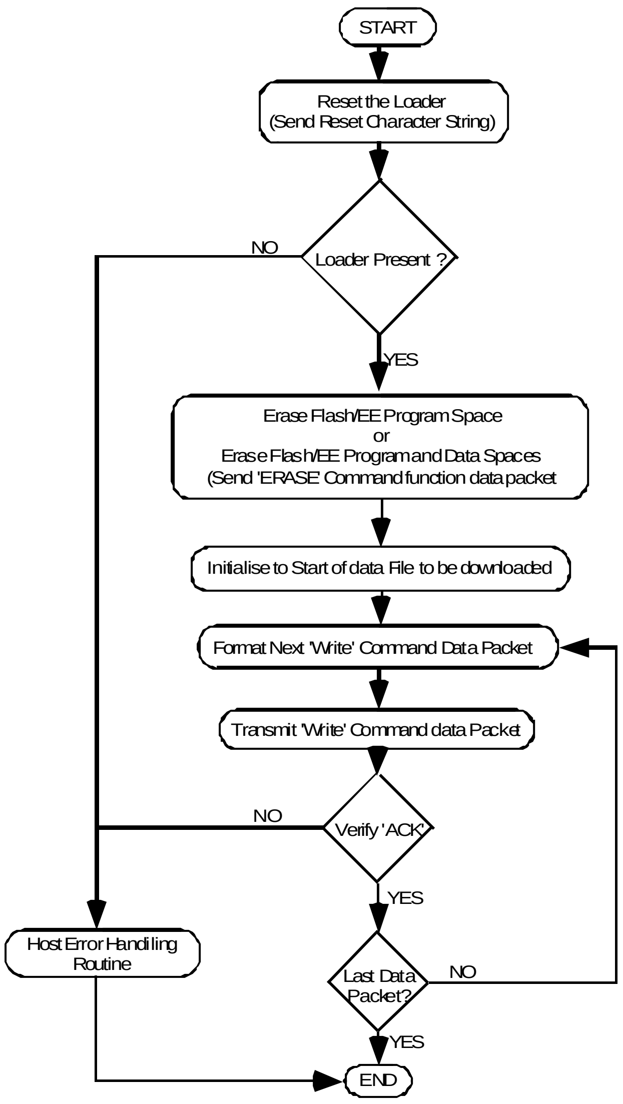

START
Reset the Loader
(Send Reset Character String)
Loader Present ?
NO
YES
Erase Flash/EE Program Space
or
Erase Flash/EE Program and Data Spaces
(Send 'ERASE' Command function data packet
Initialise to Start of data File to be downloaded
Format Next 'Write' Command Data Packet
Transmit 'Write' Command data Packet
Verify 'ACK'
NO
YES
Host Error Handling Routine
Last Data Packet?
NO
YES
END

Figure 2. Flowchart for a Typical Data Download Sequence

3.0 THE MICROCONVERTER VERSION 1 LOADER :

Note: The version 1 loader is available on some early ADuC812 MicroConverters only. The protocol is shown below:
Loader Version 1: **-** is on all ADuC812 silicon with branded date code < 9933 (silicon shipped prior to August 1999)

3.1 RUNNING THE LOADER :

As with the version 2 loader (section 2.0), the version 1 loader on the ADuC812 is initiated by pulling the PSEN pin low through a resistor (typically 1KΩ pulldown) and resetting the part (toggling the 'RESET' input pin on the part itself or a power cycle will reset the part). On reset or after a power cycle the loader will first erase the Flash/EE program and data memory and immediately send the following 11-byte ID data character string.

11 byte loader ID Data :

8 bytes	- product identifier	**"ADuC812<space>"**
3 bytes	- firmware version number	**"krl"**

3.2 THE PHYSICAL INTERFACE :

Once triggered, the loader configures the ADuC812 UART serial port to transmit/receive in the following mode, 9600 baud rate, 8 data bits, no parity.

Note: For the **ADuC812** the baud rate is derived indirectly from the master clock frequency i.e. 9600 baud rate will be set based on a master clock frequency of 11.0592MHz. If the master clock frequency is increased or decreased the baud rate will increase or decrease as follows:

Baud rate = 9600baud x Crystal Freq/11.0592MHz

e.g. if the master clock frequency is set to 1MHz then the loader will configure the UART at 868baud.
The PC program WSD.exe, discussed in section 5.0 later allows the user to input the target master clock rate and thus reconfigure the PC baud rate to match that of the loader.

3.3 INTERROGATING THE LOADER :

The loader should be first interrogated to verify both that the loader is correctly present and that the loader firmware version number is as expected.
The version 1 loader can be interrogated at 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 its 11-byte ID data character string mentioned earlier and detailed again below.

11 byte loader ID Data :

8 bytes	- product identifier	**"ADuC812<space>"**
3 bytes	- firmware version number	**"krl"**

3.4 DOWNLOADING CODE TO THE VERSION 1 LOADER :

Unlike the version 2 loader described previously (section 2.0), the version 1 loader supports direct download to Flash/EE program space only. The host must transmit an un-edited 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 (see section 3.1 above). It should also be noted that the loader does not support data download to the Flash/EE data memory space.

Once the loader has been interrogated as described in section 3.3 above, the loader will then wait for the transmission of an Intel Standard 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 following communications protocol. Note the format of the standard Intel Hex file is described in section 3.6 below.

The loader recognizes a record by the start character ':' and it is important to note that it will use the various bytes that precede the record such as address, number of bytes in record etc. as well as the checksum at the end of the standard record file format. Therefore it is important to that the host transmits the Intel hex file 'as is'.

At the end of each record, the MicroConverter will transmit an ACK (everything OK) or a NACK (everything not OK). The ACK is 06H and NACK is 07H. So the user host should interpret these appropriately i.e on ACK move onto transmit the next record on NACK report an error message to the host. On an error the micro waits for the next ':' so the host can decide to stop on an error or try to re-transmit the record.

3.5 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 ';' character and follow it by a 4-byte start address. It is important to point out here that in a real application if the host did want to download and follow this by a run from start address sequence, then this start address should be FF00hex . (The start address FF00hex ensures that a resident power-on-configuration routine runs to correctly calibrate the ADC, Internal Voltage Reference before normal user code runs from address 0000Hex.)

If you do not follow the download of the Intel Hex file with a ';' then the only way to force a start of code execution would be to remove the PSEN pulldown and force a reset or a new power cycle.

3.6 STANDARD INTEL HEX FORMAT :

This section describes the Intel Hex standard file format expected by the revision 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 your MicroConverter (8051 compatible source code) program. An 8051 compatible Assembler (Metalink 2-pass assembler) is available as part of the QuickStart development System or as a free download executable from the MicroConverter web-site at www.analog.com/microconverter.

An Intel hex file is a series of lines or 'hex records' containing the following fields.

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 – 16	data
Checksum	2	Sum of all bytes in record + checksum = 0

Table 10: Summary of the fields of an Intel Standard HEX files

These fields are shown more explicitly in Figure 3 below which illustrates a typical example print-out from an Intel hex file.

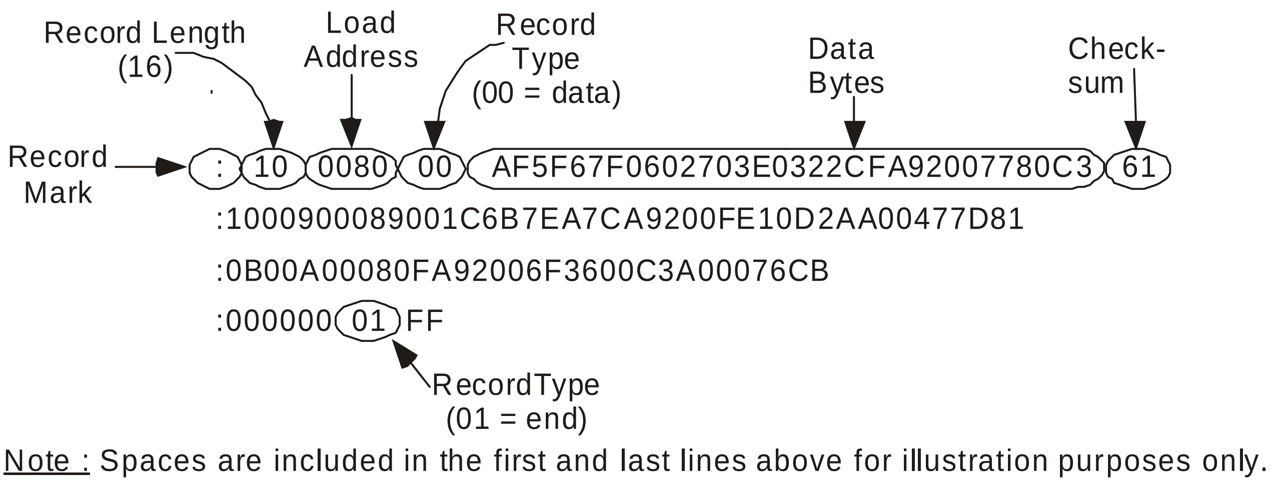

Record Length
(16)
Load
Address
Record
Type
(00 = data)
Data
Bytes
Check-
sum
Record
Mark
: 10 0080 00 AF5F67F0602703E0322CFA92007780C3 61
:1000900089001C6B7EA7CA9200FE10D2AA00477D81
:0B00A00080FA92006F3600C3A00076CB
:00000001FF
RecordType
(01 = end)
Note : Spaces are included in the first and last lines above for illustration purposes only.

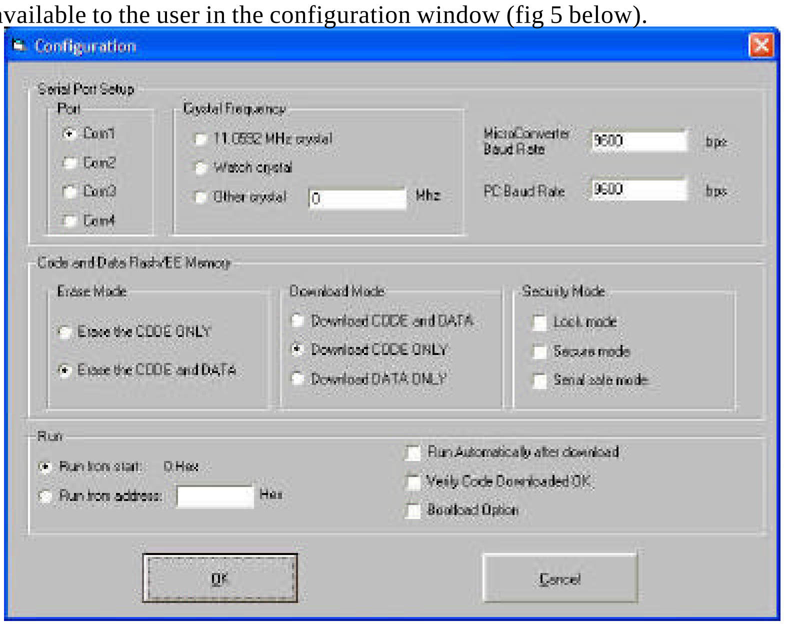
Figure 3. Intel Hex Format

4.0 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→ COM4) on a standard PC.

The configuration window allows the user to configure the DOWNLOAD and RUN options. The download should be considered as the following sequence of events....an Erase -> a Program -> and a setting of a security mode (if selected). Once the DOWNLOAD button is pressed on the WSD then the download options as set in the configuration window are sent to the MicroConverter.

The following Download and RUN options are available to the user in the configuration window (fig 5 below).



Configuration

Serial Port Setup
- Port: [✓ Com1] Com2 Com3 Com4
- Crystal Frequency: 11.0592 MHz crystal / Watch crystal / Other crystal 0 Mhz
- MicroConverter Baud Rate: 9600 bps
- PC Baud Rate: 9600 bps

Code and Data Flash/EE Memory
- Erase Mode: Erase the CODE ONLY [✓ Erase the CODE and DATA]
- Download Mode: Download CODE and DATA [✓ Download CODE ONLY] Download DATA ONLY
- Security Mode: ☐ Lock mode ☐ Secure mode ☐ Serial safe mode

Run
- [✓ Run from start: 0 Hex] Run from address: Hex
- ☐ Run Automatically after download
- ☐ Verify Code Downloaded OK
- ☐ Bootload Option

<u>Port:</u> The user can select the required PC COM port.

<u>Crystal Frequency:</u>
The user selects the device operating frequency.

Erase Mode:
The user can choose to:
(a) erase the flash/EE program memory (CODE), or,
(b) erase both the flash/EE program and data memory (CODE and DATA).

<u>Download Mode:</u>
The user can choose to:
(a) program the Flash/EE program and Data memory (CODE and DATA).
(b) program the Flash/EE data memory or
(c) program the data memory only.

<u>Security Mode:</u>
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 the security modes.

<u>RUN:</u>
The user can choose to:
(a) RUN from start : 0 Hex
(b) RUN from a particular address as entered by the user in hexadecimal format.
(c) RUN automatically after a download.
(d) Run from the Bootload option (i.e. from address E000h). See uc007 for more information on ULOAD mode.
(e) Verify the downloaded code.

Note: In the case of (a) and (b) above the code will not RUN until the user clicks on the RUN button on the WSD. In the case of (c) the RUN command is automatically sent down to the MicroConverter after the download is complete.

4.1 COMMAND LINE INTERFACE TO THE WINDOWS SERIAL DOWNLOADER:

The WSD software can be driven from a command line interface also. To download a file simply type Download.exe <file.hex>

This would download the program called 'file.hex'.

ADDITIONAL DOWNLOADER OPTIONS

To make use of the additional downloader options, type any of the following switches immediately following the ,file.hex> name (making sure to precede each with a space):

/C:n	- select the COM port required (default is 1)
/D	- download the to Dlash/EE program memory without erasing the Flash/EE data memory
/R	- run the program from the beginning, i.e. 0000hex
/R:xxxx	- run the program from the specified (xxxx) hex start address
/V	- verify code download (only works with ADuC83x and ADuC84x parts)
/X:freq	- Set the crystal frequency value being used (only for ADuC812, ADuC831 and ADuC841)
/B	- Bootload Option, runs from E000h
/M:n	- Download Mode
	- 0 = Download Code and Data (Data file must be second file name specified)
	- 1 = Download Code Only
	- 2 = Download Data Only (Data file must be second file name specified. Program hex file will not be downloaded)
/S:n	-Security Mode (All parts except the ADuC812)
	- 0 = Lock Mode
	- 1 = Secure Mode
	- 2 = Serial Safe Mode
	- 3 = All Security bits set
/T:n	- Time program remains open for. N is an integer between 0 – 9 seconds.