

理解串行下载协议 (之前称为uC004)

简介

本应用笔记描述用于ADuC8xx系列产品的串行下载协议。

MicroConverter®产品系列的诸多功能之一就是器件能够将代码在线下载至片内Flash/EE程序存储器。该在线代码下载功能是通过器件的UART串行端口进行的，因此通常称为串行下载。

借助串行下载功能，开发人员可以直接对焊接到目标系统上的器件重新编程，而无需使用外部器件编程器或者拆除器件来使用外部编程器。串行下载还使得现场系统升级变为可能；唯一要求是通过串行端口访问MicroConverter。这就意味着，制造商可以现场升级系统固件，而无需换出该器件。

任何MicroConverter器件都可以籍由特定引脚配置在上电时或者通过施加外部复位信号而配置为串行下载模式。对于MicroConverter，需要通过电阻(1 kΩ)拉低PSEN输入引脚。而对于ADuC814，必须通过电阻(1 kΩ)拉高DLOAD引脚。如果在上电时或施加外部复位信号时器件检测到这种状态，该器件会进入串行下载模式。

在这种模式下，片内固化的加载程序例程即会启动，然后片内加载程序会相应地配置器件UART并通过特定串行下载协议与各个主机通信，以管理Flash/EE存储空间的数据下载。注意，串行下载模式在器件的标称电源电压范围内工作，并且无需特定的外部高编程电压，因为该电压也在片内生成。图1所示为实际操作中ADuC812 MicroConverter的串行下载示例。

ADI公司的QuickStart™开发工具套件提供了一个Windows®可执行程序(WSD.exe)，允许用户从PC(PC串行端口COM1、COM2、COM3或COM4)下载代码至MicroConverter。但值得注意的是，无论主机为PC还是微控制器、DSP或其它，只要主机遵守本应用笔记中详述的串行下载协议，就可以下载代码至MicroConverter。

本应用笔记旨在详细地描述MicroConverter串行下载协议，让最终用户不仅能够完全了解该协议，还可以根据需要在最终目标系统中成功实施该协议(嵌入式主机至嵌入式MicroConverter)。

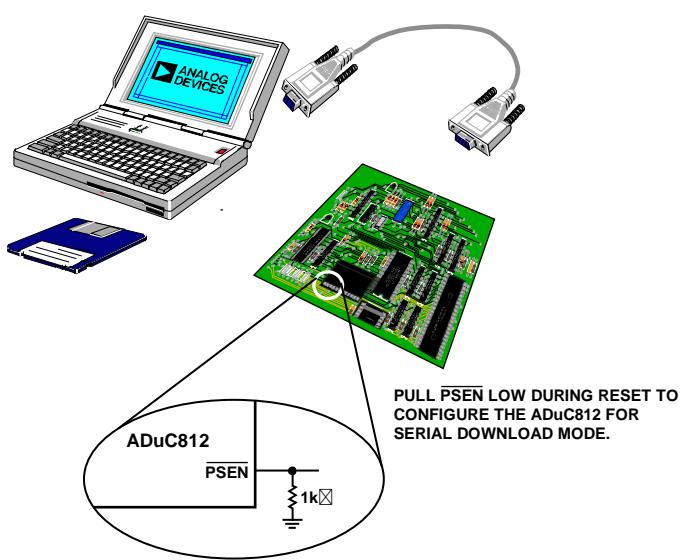


图1. 串行下载模式下的MicroConverter

目录

简介	1
背景知识	3
MicroConverter Version 2加载程序	4
运行加载程序	4
物理接口	4
查询加载程序	4
定义数据传输包格式	4
数据包命令功能	6
MicroConverter Version 1加载程序	11
运行加载程序	11
物理接口	11
查询加载程序	11
将代码下载至Version 1加载程序	11
运行代码	11
标准Intel Hex格式	12
Windows串行下载器(WSD.exe)	13
可用的下载和运行选项	13
与Windows串行下载器的命令行接口	14

背景知识

所有MicroConverter器件上的串行下载协议均由片内加载程序例程定义。MicroConverter片内加载程序具有以下两种版本：Version 1和Version 2。

标记日期代码在9933之前的所有ADuC812硅片(1999年8月之前发运的硅片)采用的是Version 1。

以下硅片上采用的是Version 2：

- 所有标记日期代码为9933或更早的ADuC812硅片(1999年之后发运的硅片)
- 所有ADuC814硅片
- 所有ADuC816硅片
- 所有ADuC824硅片
- 所有ADuC831和ADuC832硅片
- 所有ADuC836和ADuC834硅片
- 所有ADuC841、ADuC842和ADuC843硅片
- 所有ADuC845、ADuC847和ADuC848硅片

Version 1加载程序支持直接将原始Intel Hex格式文件下载至MicroConverter(仅限ADuC812)Flash/EE程序存储空间。该协议定义为简单的功能协议，详见“MicroConverter Version 1加载程序”部分。

Version 2加载程序支持更为全面的协议，允许串行下载至the Flash/EE程序空间或Flash/EE数据空间。借助这些功能以及其它功能，用户可以更加灵活和安全地使用MicroConverter的在线下载功能。

为了使本应用笔记中的描述更为清楚明了，“主机”一词是指尝试下载数据至MicroConverter的主机(PC、微控制器、DSP或其它机器)。“加载程序”一词特指驻留在MicroConverter上的片内串行下载固件。

MicroConverter Version 2加载程序

下列MicroConverter采用的是Version 2加载程序：

- 标记日期代码为9933或更早的ADuC812硅片(1999年之后发送的硅片)
- ADuC814硅片
- ADuC816硅片
- ADuC824硅片
- ADuC831和ADuC832硅片
- ADuC836和ADuC834硅片
- ADuC841、ADuC842和ADuC843硅片
- ADuC845、ADuC847和ADuC848硅片

运行加载程序

要运行MicroConverter上的加载程序，请通过电阻(通常为1 kΩ下拉电阻)将PSEN引脚拉低(对于ADuC814，请将DLOAD拉高)，然后复位器件(切换器件本身的RESET输入引脚或经过一个上电周期来复位器件)。复位时或经过一个上电周期后，加载程序会立即发送下列25个字节的ID数据包。

25个字节组成如下：

- 10个字节为产品标识符占

ADI<space>8xx<space><space><space>

其中：

8xx为产品标识符(例如，ADuC812时为812)。

- 4个字节为固件版本号(例如，对于Version 2加载程序，则为V2xx。xx为版本修订号)。
- 2个字节为换行符和回车符。
- 2个字节为硬件配置。
- 6个字节保留供将来使用。
- 1个字节为前24个字符的二进制补码校验和。

物理接口

触发后，加载程序将MicroConverter UART串行端口配置为以9600波特率、8个数据位和无奇偶校验的形式发送/接收数据。

对于ADuC812、ADuC831和ADuC841，波特率间接地由主时钟频率决定；也就是说，9600波特率是根据11.0592 MHz主时钟频率来设置的。如果主时钟频率升高或降低，波特率会根据器件数据手册中给出的相关波特率计算公式相应地升高或降低。对于ADuC812，计算公式如下：

$$\text{波特率} = 9600 \text{ 波特率} \times \text{晶振频率} / 11.0592 \text{ MHz}$$

例如，如果主时钟频率设置为1 MHz，那么加载程序会将UART配置为波特率868。

PC程序WSD.exe(详见“Windows串行下载器(WSD.exe)”部分)允许用户输入目标主时钟速率，进而重新配置PC波特率来与加载程序相匹配。

注意，只有在采用值小于16.0 MHz的晶振或20 MHz晶振时，ADuC841才能以下载模式工作。采用16.0 MHz和20 MHz之间的值时，串行下载无法正常工作。

对于ADuC816、ADuC824、ADuC832、ADuC834、ADuC836、ADuC842、ADuC843、ADuC845、ADuC847和ADuC848，假设SAR ADC器件的时钟频率为16.777216 MHz (ADuC814、ADuC832、ADuC842和ADuC843)且所有Σ-Δ型器件的内核频率为12.583 MHz(ADuC834、ADuC836、ADuC845、ADuC847和ADuC848)，那么波特率配置为9600。如果存在32.768 kHz时钟晶体，那么PLL会自动锁定至这些频率(16.777216 MHz或12.58 MHz)。如果不存在晶振，那么PLL无法保证可锁定至此频率。WSD.exe程序让用户可以更改波特率，以与此环境下产生的时钟相匹配。如果在缺失晶振的情况下测得PLL时钟频率，那么可以从下式中计算得出实际的波特率(假设为12.58 MHz器件)：

$$\text{波特率} = 9600 \text{ 波特率} \times \text{PLL时钟频率} / 12.583 \text{ MHz}$$

查询加载程序

首先应当查询加载程序，确保加载程序正确存在且加载程序的固件版本号与预期相符。

对于Version 2加载程序，只要MicroConverter加载程序正在运行，就可以随时对其进行查询，方法是发送以下查询数据包至MicroConverter：

<0x21><0x5A><0x00><0xA6>或'!''Z'<0x00><校验和>

MicroConverter加载程序会立即做出响应并发送其25个字节的ID数据包。

Version 1加载程序只有在看到!字符时才会做出响应并发送ID数据包。因此，应当在发送!字符之后再发送查询数据包，即先看看是否有任何响应，然后再发送Z和查询字符串的余下部分。如果仅发送!字符，之后没有任何响应，那么加载程序为Version 2。

25个字节组成如下：

- 10个字节为产品标识符，如ADI_812_--。
- 4个字节为固件版本号，如V201。
- 2个字节为换行符和回车符。
- 2个字节为硬件配置。
- 6个字节保留供将来使用。
- 1个字节为前24个字符的二进制补码校验和。

定义数据传输包格式

一旦主机确认存在加载程序，数据传输即会开始。表1给出了通用通信数据传输包格式。

表1. 数据传输包格式

包起始ID	数据字节数	数据1命令功能	数据X ¹	校验和
0x07和0x0E	1至25 十进制	C, A, W, V, Q, E, S, B, U, T, F	xxx	至25 十进制

¹ X = 数据2至数据25。

包起始ID域

第一个域，即包起始ID域，包含两个字符(0x07和0x0E)。加载程序使用这些字节(恒定不变)来检测有效的数据包。

数据字节数域

接下来的一个域为数据字节总数，包括数据1(命令功能)。允许的最大字节数为25，但视所发送的数据包而定，从1到25不等。

数据1命令功能域

命令功能域描述数据包的功能。允许使用十一个有效命令功能之一。注意，有些功能(如擦除命令)无需发送任何额外数据。十一个命令功能由下列ASCII字符之一表示：C、A、W、V、Q、E、S、B、U、T或F。

数据X域

这些域包含要下载的数据字节。对于两个擦除命令，这些数据字节不存在。对于两个编入Flash/EE存储器命令，必须将此数据去掉标准的Intel Hex 16位记录格式，并由主机重新汇编成上述数据格式，然后再发送至加载程序。

校验和域

数据包校验和会被写入此域。二进制补码校验和是通过将数据字节数域的十六进制数值和数据1及数据2至数据X域(如果存在的话)的十六进制数值相加而计算得出的。校验和是该总和的二进制补码值。这意味着，数据字节数、数据1、数据X和校验和等域的总和应该等于0x100。这还可以通过数学方式表示为：

加载程序例程以NAK(0x07)做出对之前数据包通信的否定响应，或以ACK(0x06)做出肯定响应。在下列条件下，加载程序会发送NAK：

- 校验和字节验证发现数据包格式不正确。
- 加载程序无法验证数据编程是否正确。
- 请求加载程序将数据编入尚未擦除的位置。

表2给出了数据包命令功能的完整列表。

表2. 数据包命令功能

命令功能	数据域中的命令字节	加载程序肯定响应	加载程序否定响应
仅擦除Flash/EE程序存储器	C (0x43)	ACK (0x06)	NAK (0x07)
擦除Flash/EE程序和数据存储器	A (0x41)	ACK (0x06)	NAK (0x07)
对Flash/EE程序存储器模块编程	W (0x57)	ACK (0x06)	NAK (0x07)
读取Flash/EE程序存储器页面	V (0x56)	0x100 data bytes + CS	NAK (0x07)
下载Flash/EE程序存储器页面	Q (0x51)	ACK (0x06)	NAK (0x07)
对Flash/EE数据存储器模块编程	E (0x45)	ACK (0x06)	NAK (0x07)
设置安全模式(除ADuC812以外的所有器件)	S (0x53)	ACK (0x06)	NAK (0x07)
更改波特率(仅限使能定时器3的器件)	B (0x42)	ACK (0x06)	NAK (0x07)
跳至用户代码(远程RUN)命令	U (0x55)	ACK (0x06)	NAK (0x07)
设置ETIM寄存器(仅限具有V2.22或更高版本内核的ADuC812)	T (0x54)	ACK (0x06)	NAK (0x07)
使能引导命令(UPLOAD)	F (0x46)	ACK (0x06)	NAK (0x07)

数据包命令功能

擦除命令

如表2所示，共有两个擦除命令。其中一个命令擦除Flash/EE程序和数据存储器(A)，而另一个命令仅擦除Flash/EE程序存储器(C)。

如果主机希望下载数据至Flash/EE程序存储器或Flash/EE数据存储器，那么首先必须使用C(擦除Flash/EE程序存储器)或A(擦除Flash/EE程序和数据存储器)命令字节来擦除这些存储空间。如果主机尝试下载数据至未事先擦除的存储空间，那么加载程序会做出NAK响应。

擦除命令无需任何数据包格式的额外数据。表3所示的数据包示例可执行Flash/EE程序和数据存储器擦除。表4所示的数据包示例仅执行Flash/EE程序存储器擦除。

表3. 擦除Flash/EE程序和数据存储器命令

包起始ID	数据字节数	数据1(命令)	校验和
0x07 and 0x0E	0x01	A (0x41)	0xBE 0x100 – 数据字节数 + Σ 数据字节

表5. 对Flash/EE程序存储器模块编程

起始ID	数据字节数	数据1 CMD	数据2 ADR U	数据3 ADR M	数据4 ADR L	数据5 Prog 1	数据6 Prog 2	数据7 Prog 3	数据8 Prog 4	数据9 Prog 5	数据10 Prog 6	数据11 Prog 7	数据12 Prog 8	校验和
0x07 and 0x0E	0x0C (12)	W (0x45)	0x00	0x00	0x00	0x00	0x0C	0x0E	0x0C	0x0F	0x0E	0x4F	0x63	0xBA

表4. 仅擦除Flash/EE程序存储器命令

包起始ID	数据字节数	数据1(命令)	校验和
0x07 and 0x0E	0x01	C (0x43)	0xBC 0x100 – 数据字节数 + Σ 数据字节

对Flash/EE程序存储器模块编程

所有下载数据包命令都需要起始地址。该地址包含在紧跟命令功能数据字节之后的三个字节中。下载数据包还包含要下载的原始数据。

加载程序将第一个数据字节写入数据包格式中指定的地址。然后，加载程序递增地址并写入下一个字节，依此类推，直到数据包中的所有数据字节均完成写入。

表5所示的数据包示例会从地址0x0000开始对Flash/EE程序空间的八个空间位置编程。

读取Flash/EE程序存储器页面

“读取Flash/EE程序存储器页面”允许回读和验证已下载的程序存储。

“读取Flash/EE程序存储器页面”命令需要页码。每页包含0x100个字节。

加载程序以0x100个字节的数据(对应于指定页面的当前内容)回应，紧跟其后的是该数据的校验和。

如果下载会话中在此命令之前并未发出擦除命令，则返回NACK响应。因此，该命令只可用于回读相同段中下载的代码，而无法用户回读之前下载段中的代码。

表6所示的数据包示例会从地址0x100开始回读Flash/EE程序空间的0x100个位置。

表6. 读取Flash/EE程序存储器页面命令

Start ID	Data Bytes	CMD	ADR U	Checksum
0x07 and 0x0E	0x02 (2)	V (0x56)	0x01	0xA7

表7. 读取Flash/EE程序存储器页面响应

数据1	数据2	数据256	校验和
DATA	DATA	DATA	CS

表8. 页面下载命令

起始ID	数据1 CMD	数据2 ADR U	数据3 Prog 1	数据4 Prog 2	数据5 Prog 3	数据6 Prog 4	...	数据258 Prog 256	校验和
0x07	Q (0x51)	0x01					...		CS

表9. 对Flash/EE数据存储器模块编程

起始ID	数据字节数	数据1 CMD	数据2 ADR U	数据3 ADR M	数据4 ADR L	数据5 Prog 1	数据6 Prog 2	数据7 Prog 3	数据8 Prog 4	校验和
0x07and 0x0E	0x08	E (0x45)	0x00	0x00	0x05	0x0A	0x0B	0x0C	0x0D	= 0x80(此例中) =0x100 - 数据字节数 + Σ 数据字节

PAGEDOWNLOAD命令

PAGEDOWNLOAD命令将0x100个字节的数据下载至指定页面。此命令在其它文档称为QuickDownload命令。

PAGEDOWNLOAD命令需要页码。每页包含0x100个字节。

PAGEDOWNLOAD数据包包含要下载至指定页面的0x100个数据。加载程序将Prog 1字节写入数据包格式中所指定页面的第一个地址。然后，加载程序递增地址并写入下一个字节，依此类推，直到数据包中的所有数据字节均完成写入。

如果PAGEDOWNLOAD命令不受支持，则返回NACK响应。如果PAGEDOWNLOAD命令不受支持，可使用“对Flash/EE程序存储器模块编程”。

表8所示的数据包示例会从地址0x100开始写入Flash/EE程序空间的0x100个位置。

对Flash/EE数据存储器模块编程

从任何MicroConverter数据手册均可看出，必须以4字节页面形式编入640字节的Flash/EE数据空间(即640字节空间配置为160个页面)。表9所示的数据包示例会对Flash/EE数据空间中的页面5编程。

设置安全模式命令

除ADuC812以外的所有器件均存在这三个安全选项。有关各安全模式保护的具体内容,请参见器件数据手册。两个擦除命令均可将安全模式恢复至非安全模式。因此,每次下载后,必须再次设置安全模式。要设置安全功能,请在命令类型(S)后发送相应的数据字节2来选择对应的安全选项,具体如表10所示。

表10. 不同安全模式所需的数据2字节

安全模式	数据2字节
锁定模式	0x06
保险模式	0x05
保险和锁定模式	0x04
串行安全模式	0x03
串行安全和锁定模式	0x02
串行安全和保险模式	0x01
串行安全、保险和锁定模式	0x00

串行安全模式会禁用串行下载器。因此,通过写入此位,可以防止日后发生串行下载。禁用此安全选项的唯一方式是在并行编程模式下启动代码擦除命令。

安全模式实际上包括锁定模式的所有安全功能。因此,从安全角度来看,安全模式和锁定模式实际上是相同的。

要设置安全模式,主机需要发送表11所示的数据包。

表11. 保险模式命令设置示例

包起始ID	数据字节数	数据1(命令)	数据2	校验和
0x07 and 0x0E	0x02	S (0x53)	0x05	0xA6 0x100 - 数据字节数 +Σ数据字节

更改波特率命令

有些ADuC8xx型号配有专门波特率定时器(定时器3),用于产生针对不同晶振值的高精度波特率。ADuC812、ADuC814、ADuC816或ADuC824上没有该定时器。有关所需波特率对应的不同T3CON和T3FD值,请参阅相关的数据手册。

表12. 波特率更改命令设置示例

包起始ID	数据字节数	数据1(命令)	数据2	数据3	校验和
0x07 and 0x0E	0x03	B (0x42)	T3CON	T3FD	0x100 - No. of Data Bytes + Σ Data Byte

跳至用户代码(远程RUN)命令

在将所有数据包发送至加载程序之后,主机可以发送最后一个数据包来指示加载程序将MicroConverter程序计数器强制设为给定地址。这会执行刚刚下载的代码。表14显示了远程RUN或从地址0x00跳至用户代码的示例。

执行远程RUN后的RAM

ADuC812会在上电配置例程期间调用内部RAM清除功能。因此,复位(来运行用户代码)之后,内部RAM显示为全0x00。不过,如果串行下载后使用远程RUN来运行用户代码,则会调用清除RAM例程,且RAM保留不变(也就是说, RAM仍旧保持串行下载之前那样不变),但以下位置除外,加载程序会破坏这些位置:

0x00至0x02、0x05至0x0D、0x2A和0x33至0x47

对所有后续器件执行硬件复位来运行用户代码时,内部RAM会保留不变。不过,当这些器件复位至Version 2加载程序(也即WSD或MS-DOS下载器)后,内部RAM的部分字节会被破坏。

执行远程RUN后的SFR

对MicroConverter执行硬件复位来运行用户代码时,会将所有SFR加载至其默认值(请参见相关数据手册中的SFR表)。不过,如果在以串行方式下载至MicroConverter后使用远程RUN功能,那么UART会保留为9600波特率配置。表13给出了变为非默认值的SFR。所有其它SFR在执行远程RUN后均变为其默认值。

表13. 执行远程RUN后SFR的初始值

以下器件上执行远程RUN后的SFR破坏情况:		
ADuC812	ADuC816/ADuC824	其它ADuC8x器件
TH1	TH2	T3CON
TL1	TL2	T3FD
SCON	SCON	SCON
TCON	T2CON	SBUF
TMOD	RCAP2H	
SBUF	SBUF	
	RCAP2L	

表14. 跳至用户代码(远程RUN)命令

包起始ID	数据字节数	数据1(命令)	数据2 ADR U	数据3 ADR M	数据4 ADR L	校验和
0x07 and 0x0E	0x04	U (0x55)	0x00	0x00	0x00	=0xA7(此例中) =0x100 - 数据字节数 +Σ数据字节

设置ETIM寄存器

只有在具有2.22或更高版本内核的ADuC812上才可以设置ETIM寄存器。

在ADuC812上, Flash/EE擦除和编程时序是由主时钟产生的。当主时钟频率为11.0592 MHz时, 无需写入ETIM寄存器。不过, 当在其它主时钟频率下工作时, 必须更改ETIM1和ETIM2的值, 以免数据Flash/EE耐久性和保持力下降。ETIM1和ETIM2构成一个16位字。ETIM2为高字节, 而ETIM1为低字节。该16位字的值必须进行适当设置, 以确保获得最佳数据Flash/EE耐久性和保持力。此命令只会在下载持续期间设置ETIM寄存器, 下载结束后这些寄存器即会复位至其默认值。当主时钟频率不等于11.0592 MHz时, 请将这些寄存器编程为编程代码的一部分。

对于ADuC812, 计算公式为

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

ETIM3应该始终保持为其默认值: 201(十进制)/C9(十六进制)。

表17给出了晶振为12 MHz时的示例, 其中 $100 \mu s \times 12 \text{ MHz} = 1200$ (十进制) = 0x04B; 因此, ETIM2 = 0x04且ETIM1 = 0xB0。

所有其它器件(ADuC814、ADuC831、ADuC832、ADuC834、ADuC836、ADuC841、ADuC842、ADuC843、

表17. 对ETIM1和ETIM2寄存器编程

包起始ID	数据字节数	数据1(命令)	数据2ETIM1	数据3ETIM2	数据4ETIM3	校验和
0x07和0x0E	0x04	T(0x54)	0xB0	0x04	0xC9	=0x100 - 数据字节数 + Σ数据字节

ADuC845、ADuC847和ADuC848)均会自动根据所用内核频率设置所需的ETIM值。用户无需编辑这些寄存器。

引导命令

串行下载器中存在一个引导使能选项, 可确保在复位后始终从0xE000运行。如果使用了引导程序, 建议使用此选项, 以确保该引导程序在复位后始终执行正确的代码。欲了解通过ULOAD模式对Flash/EE程序存储器编程, 请参见产品数据手册和uC007在线文档。

表15. 使能引导命令设置示例

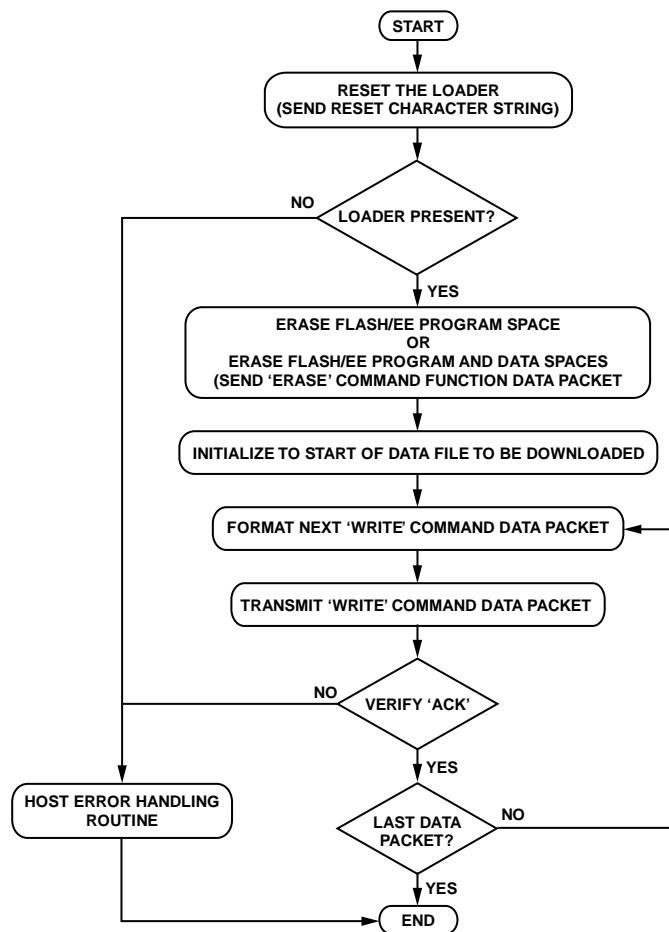
包起始ID	数据字节数	数据1(命令)	数据2	校验和
0x07和0x0E	0x02	F (0x46)	0xFF	0x100 - 数据字节数 + Σ数据字节

表16. 禁用引导命令设置示例

包起始ID	数据字节数	数据1(命令)	数据2	校验和
0x07和0x0E	0x02	F (0x46)	0xFF	0x100 - 数据字节数 + Σ数据字节

下载序列概要

现在便可以使用Version 2加载程序从主机下载程序至MicroConverter。图2所示为典型下载序列的流程图。



090222-002

图2. 典型数据下载序列的流程图

MicroConverter Version 1加载程序

只有一些早期ADuC812 MicroConverter上带有Version 1加载程序。

标识日期代码小于9933的所有ADuC812硅片(1999年8月之前发运的硅片)上均提供有此版本的加载程序。

运行加载程序

与Version 2加载程序一样, 要启动ADuC812上的Version 1加载程序, 请通过电阻(通常为1 kΩ下拉电阻)将PSEN引脚拉低并复位器件。要复位器件, 请切换RESET输入引脚; 此外, 经过一个上电周期也可复位器件。复位时或经过一个上电周期后, 加载程序会擦除Flash/EE程序和数据存储器, 然后立即发送余下的11字节ID数据字符串:

- 8个字节为产品标识符: ADuC812<space>
- 3个字节为固件版本号: krl

物理接口

触发后, 加载程序将ADuC812 UART串行端口配置为以9600波特率、8个数据位和无奇偶校验模式发送/接收数据。

对于ADuC812, 波特率间接地由主时钟频率决定; 也就是说, 9600波特率是根据11.0592 MHz主时钟频率来设置的。如果主时钟频率升高或降低, 波特率也会相应地升高或降低, 具体如下所示:

$$\text{波特率} = 9600 \text{ 波特率} \times \text{晶振频率} / 11.0592 \text{ MHz}$$

例如, 如果主时钟频率设置为1 MHz, 那么加载程序会将UART配置为波特率868。

PC程序WSD.exe允许用户输入目标主时钟速率, 进而重新配置PC波特率来与加载程序相匹配。

查询加载程序

应当查询加载程序, 确保加载程序正确存在且加载程序的固件版本号与预期相符。

对于Version 1加载程序, 只要MicroConverter加载程序正在运行, 就可以随时对其进行查询, 方法是发送以下数据包至MicroConverter:

查询字符<21 hex> (ASCII '?')

MicroConverter加载程序会立即做出响应并发送其11个字节的ID数据字符串。

- 8个字节为产品标识符: ADuC812<space>
- 3个字节为固件版本号: krl

将代码下载至Version 1加载程序

与Version 2不同, Version 1加载程序仅支持直接下载至Flash/EE程序空间。主机必须发送未编辑版的标准Intel Hex格式程序文件; 加载程序要求将此作为下载序列的一部分直接发送。

使能后, Version 1加载程序会立刻自动擦除Flash/EE程序和数据空间, 然后发送11字节ID数据字符串。注意, 该加载程序不支持将数据下载至Flash/EE数据存储空间。

完成查询后, 该加载程序等待发送Intel Hex文件, 该文件将被编入Flash/EE程序存储空间。它以记录为基础接收此文件, 并要求主机遵守标准通信协议。注意, 标准Intel Hex文件的格式将在“标准Intel Hex格式”部分中介绍。

该加载程序通过起始字符(:)来识别记录。注意, 它在记录之前设有多个字节(如地址和记录中的字节数)并在标准记录文件格式末尾设有校验和。因此, 主机必须按原样发送Intel Hex文件, 这点非常重要。

在每条记录末尾, MicroConverter发送一个ACK(一切正常)或NACK(不正常)。ACK为0x06, 而NACK为0x07。用户主机应该适当地解读这些字节。例如, 在收到ACK时, 继续发送下一条记录, 而在收到NACK时, 则向主机报告错误消息。出现错误时, MicroConverter等待下一个起始字符(:), 以便主机可以决定是继续停留在错误处, 还是尝试重新发送该记录。

运行代码

在文件传送结束时, 用户主机通过发送起始地址命令字符串来强制执行代码。发送起始地址时可以先发送冒号字符, 然后发送4字节起始地址。注意, 在实际应用中, 如果主机希望进行下载并在下载后从起始地址序列运行, 那么该起始地址应为0xFF00。(起始地址为0xFF00时, 可确保驻留的上电配置例程正确运行来校准ADC内部基准电压源, 然后可从地址0x0000正常运行用户代码。)

如果并不是在冒号之后下载Intel Hex文件, 那么强制开始代码执行的唯一方式是移除PSEN下拉电阻并强制复位或进入一个新的上电周期。

标准Intel Hex格式

此部分介绍Version 2加载程序协议所要求的Intel Hex标准文件格式。Intel十六进制格式(也称为“Intel Hex格式”)是一项以可显示或可打印格式存储机器语言的标准。

通过对MicroConverter(8051兼容源代码)程序进行汇编即可生成标准Intel Hex文件。QuickStart开发系统中提供了一种8051兼容汇编器,即MetaLink二次扫描汇编器;另外,也可从MicroConverter网站上免费下载该汇编器的可执行文件。

Intel Hex文件是一系列线路或十六进制记录,这些包含在表18所示的域中。图3所示为一个Intel Hex文件的典型样本打印输出,图中更为清楚地显示了这些域。

表18. 标准Intel Hex文件的域概要

域	字节	描述
记录标记	1	:指示记录开头。
记录长度	2	记录中的数据字节数。
加载地址	4	数据字节的起始地址。
记录类型	2	00 = 数据记录, 01 = 结尾记录。
数据字节	0 to 16	数据。
校验和	2	记录中所有字节之和 + 校验和 = 0。

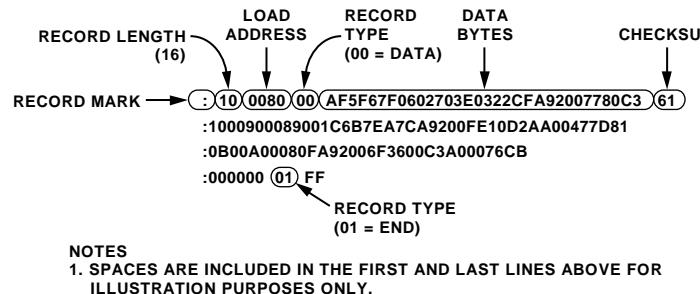


图3. Intel Hex格式

Windows串行下载器(WSD.exe)

Windows串行下载器(WSD)是一款Windows软件程序，可允许用户以串行方式下载ASM51汇编器创建的标准Intel Hex文件。WSD自动与Version 1或Version 2加载程序搭配工作。标准Intel Hex文件通过标准PC上的任意串行端口(COM1至COM32)下载到片内程序闪存上。

配置窗口允许用户配置下载和运行选项。按下WSD上的“下载”按钮之后，在配置窗口中设定的下载选项即被发送至MicroConverter。

可用的下载和运行选项

用户可以在配置窗口(见图4)中设定下列下载和运行选项：



图4. WSD配置

端口。用户可选择所需的PC COM端口。

晶振频率。用户可选择器件工作频率。

擦除模式。用户可选择擦除Flash/EE程序存储器(仅擦除CODE)或擦除Flash/EE程序和数据存储器(CODE和DATA)。

下载模式。用户可选择对Flash/EE程序和数据存储器编程(下载CODE和DATA)、对Flash/EE数据存储器编程(仅下载CODE)或仅对数据存储器编程(仅下载DATA)。

安全模式。对于ADuC816、ADuC824、ADuC834、ADuC836、ADuC831、ADuC832、ADuC841、ADuC842、ADuC843、ADuC845、ADuC847和ADuC848，三种安全模式均可用。用户可选择设置这些安全模式的任意组合。

运行。用户可选择执行以下操作：

- 从起始(0 hex)运行。
- 从以十六进制格式输入的特定地址运行。
- 下载完成后自动运行。
- 从引导选项(即从地址0xE000)运行。有关ULOAD模式的更多信息，请参见ADI公司网站上的“MicroConverter技术笔记(uC007)用户下载(ULOAD)模式”文档。

对于前两项运行条目，在用户单击WSD上的运行按钮之前，代码并不会开始运行。对于第三项条目，下载完成后会自动发送运行命令至Micro-Converter。

验证。用户可验证已下载的代码。

与Windows串行下载器的命令行接口

WSD软件也可从命令行界面驱动。要下载文件, 请输入
WSD <file.hex>。此命令会下载称为“file.hex”的程序。

其它下载器选项

要使用其它下载器选项, 请在file.hex>名称后紧接着输入
以下任意转换参数。每个转换参数后, 请留出一个空格。

- /C:n用于选择所需的COM端口(默认值为1)。
- /D用于下载至Flash/EE程序存储器, 而不擦除Flash/EE
数据存储器。
- /R用于从开头(即0x00)运行程序。
- /R:xxxx用于从指定(xxxx)十六进制起始位置运行程序。
- /V用于验证代码下载(此参数仅适用于ADuC83x和
ADuC84x器件)。
- /X:freq用于设置晶振频率值(仅适用于ADuC812、
ADuC831和ADuC841)。
- /B用于指定引导选项(此参数从0xE000运行)。
- /M:n用于指定下载模式, 其中:

0=下载代码和数据(数据文件必须是指定的第二个文件
名)。

1=仅下载代码。

2=仅下载数据。(数据文件必须是指定的第二个文件
名。将不会下载十六进制程序文件。)

- /S:n用于指定安全模式。(除ADuC812以外的所有器
件。)

0=锁定模式。

1=保险模式。

2=串行安全模式。

3=设定所有安全位。

- /T:n用于指定程序保持打开状态的时间长度, 其中n为
整数且介于0至9秒之间。

注释

注释