AN-1262
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

ADuCM350 Serial Download Protocol

INTRODUCTION

A key feature of the ADuCM350 is the ability of the device to download code to the on-chip flash program memory while in circuit. An in-circuit code download, conducted over the device UART serial port, is commonly referred to as a serial download.

The serial download capability allows developers to reprogram the part while it is soldered directly onto the target system, avoiding the need for an external device programmer. The serial download feature also enables system upgrades to be performed in the field; all that is required is serial port access to the ADuCM350. As a result, system firmware can be upgraded in the field without having to swap out the device.

The ADuCM350 can be configured for serial download mode via a specific pin configuration at power on or after any reset.

In this mode, an on-chip resident loader routine is initiated. The on-chip serial downloader configures the device UART and, via a specific serial download protocol, communicates with any host machine to manage the download of data into its flash memory spaces. The format of the program data to download must be little endian.

Note that serial download mode operates within the standard supply rating of the part. Therefore, there is no requirement for a specific high programming voltage because it is generated on chip.

As part of the development tools, a Windows® program (CM3WSD.exe) is provided by Analog Devices, Inc. CM3WSD.exe allows a user to serially download Intel extended hexadecimal files as created by assembler/compilers to the ADuCM350 via the serial port.

Note, however, that any master host machine (PC, microcontroller, or DSP) can download to the ADuCM350 device once the host machine adheres to the serial download protocols detailed in this application note.

This application note details the ADuCM350 device serial download protocol, allowing end users to understand and successfully implement this protocol (embedded host to embedded ADuCM350 device) in an end-target system.

For the purposes of clarity, the term host refers to the host machine (PC, microcontroller, or DSP) attempting to download data to the ADuCM350 device. The term loader refers to the on-chip serial download firmware on the ADuCM350 device.

TABLE OF CONTENTS

REVISION HISTORY

11/13—Revision 0 Initial Version

RUNNING THE ADuCM350 LOADER

The loader on the ADuCM350 device is initiated by pulling the BOOT/kernel_gpio pin high through a resistor (typically 1 kΩ pull-down) and resetting the part.

THE PHYSICAL INTERFACE

Once triggered, the loader waits for the host to send a backspace (BS = 0x08) character to synchronize. The loader measures the timing of this character and, accordingly, configures the ADuCM350 UART serial port to transmit/receive at the host baud rate with eight data bits and no parity. The baud rate must be between 9600 bps and 115200 bps, inclusive.

Upon receiving the backspace, the loader immediately sends the following 24 byte ID data packet:

- 15 bytes = product identifier
- 3 bytes = hardware and firmware version number
- 4 bytes = reserved for future use
- 2 bytes = line feed and carriage return

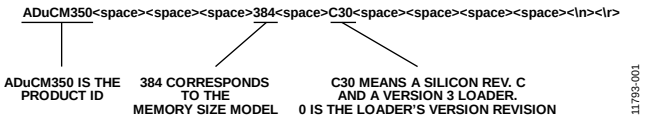

Figure 1. Example ID Data Packet

DEFINING THE DATA TRANSPORT PACKET FORMAT

Once the UART has been configured, a data transfer can begin. The general communications data transport packet format is shown in Table 1.

Start ID Field

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

Number of Bytes Field

The next field is the total number of bytes field. The minimum number of bytes is five, which corresponds to the command and value fields. The maximum number of bytes allowed is 255: a command function, a 4-byte value, and 250 bytes of data.

Command Field (Data 1)

The command field describes the function of the data packet. One of four valid command functions is allowed. The four command functions are described by one of four ASCII characters: E, W, V, or R. The list of data packet command functions is shown in Table 2.

Value Field (Data 2 to Data 5)

The value field contains a 32-bit value in big endian format.

Data Bytes Field (Data 6 to Data 255)

The data bytes field contains a maximum of 250 data bytes.

Checksum Field

The data packet checksum is written into the checksum field. The twos complement checksum is calculated from the summation of the hexadecimal values in the number of bytes field and the hexadecimal values in the Data 1 to Data 255 fields (as many as exist). The checksum is the twos complement value of this summation. Thus, the LSB of the sum of all the bytes from the number of data bytes to the checksum inclusive should be 0x00. This can also be expressed mathematically as

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

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

Acknowledge of Command

The loader routine issues a BEL (0x07) as a negative response or an ACK (0x06) as a positive response to each data packet.

A BEL is transmitted by the loader if it receives an incorrect checksum or an invalid address. The loader does not give a warning if data is downloaded over old (unerased) data. The PC interface must ensure that any location where code is downloaded is erased.

Table 1. Data Transport Packet Format

Start ID		No. of Bytes	Command	Value				Data Bytes	Checksum
ID0	ID1	X	Data 1	Data 2	Data 3	Data 4	Data 5	Data [x]	CS
0x07	0x0E	0x05 to 0xFF	E, W, V, or R	MSB	…	…	LSB	0x00 to 0xFF	0x00 to 0xFF

COMMANDS

The complete list of commands implemented in the on-chip loader is shown in Table 2.

Mass Erase Command

The mass erase command allows the user to mass erase flash, erasing the entire user code space.

The data packet for the mass erase command is shown in Table 3.

Write Command

The write command includes the number of data bytes (5 + x), the command, the address of the first data byte to program, and the data bytes to program. The bytes are programmed into flash as they arrive. The loader sends a BEL if the checksum is incorrect or if the address received is out of range. If the host receives a BEL from the loader, the download process should be aborted and the entire download sequence started again.

Verify Command

The loader requires two pieces of information to verify the contents of a page: the contents of the last 8 bytes of the page and the 32-bit forward signature of the page excluding the last 8 bytes.

To verify a page, a three-step sequence must be followed. Repeat this three-step sequence for each page to be verified.

1. Send the value 0x80000000 in the value field and the last word of the page (address offsets 0x7FC to 0x7FF) in the data bytes field.
2. Send the value 0x90000000 in the value field and the second last word of the page (address offsets 0x7F8 to 0x7FB) in the data bytes field.
3. Send the start page address in the value field and the forward signature of the page in the data bytes field.

After receiving these three packets, the loader computes the forward signature of the specified page using the flash controller FSIGN command and compares it to the supplied value. If it is correct and the value at Address 0x1FC of that page matches the value specified in Step 1, ACK (0x06) is returned; otherwise, BEL (0x07) is returned.

Remote Reset Command

Once the host has transmitted all data packets to the loader, the host can send a final packet instructing the loader to perform a reset. A software self-reset is implemented. The value field should always be 0x1.

The host should ensure that the BOOT/kernel_gpio pin used to initiate the serial programming is no longer asserted before issuing this command. When the part resets, reenter the kernel as normal. The loader entry check is performed once more, thus the BOOT/kernel_gpio pin must be deasserted at this time. Table 8 shows an example of a remote reset.

CRC CALCULATION

The signature is a 32-bit cyclic redundancy check (CRC) with the polynomial

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

The initial value is 0xFFFFFFFF.

The serial downloader uses the forward signature feature of the flash controller.

A convenient way to calculate the CRC is to download the free pycrc tool available online. A software application created by the pycrc tool is provided with the ADuCM350 software development kit (SDK) as part of the FlashTest example (C:\Analog Devices\ADuCM350\Eval-ADUCM350\examples\FlashTest\crc32.c).

Table 2. Data Packet Command Functions

Command Functions	Command Byte in Data 1 Field	Loader Positive Acknowledge	Loader Negative Acknowledge
Mass Erase	E (0x45)	ACK (0x06)	BEL (0x07)
Write	W (0x57)	ACK (0x06)	BEL (0x07)
Verify	V (0x56)	ACK (0x06)	BEL (0x07)
Remote Reset	R (0x52)	ACK (0x06)	BEL (0x07)

Table 3. Mass Erase Flash Memory Command

Start ID			Command	Value				Data	Checksum
ID0	ID1	No. of Bytes	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	CS
0x07	0x0E	0x06	E (0x45)	0x00	0x00	0x00	0x00	0x00	0x00 to 0xFF

Table 4. Write Flash Memory Command

Start ID			Command	Value				Data Bytes	Checksum
ID0	ID1	No. of Bytes	Data 1	Data 2	Data 3	Data 4	Data 5	Data [x]	CS
0x07	0x0E	5 + x (0x06 to 0xFF)	W (0x57)	ADR [31:24]	ADR [23:16]	ADR [15:8]	ADR [7:0]	0x00 to 0xFF	0x00 to 0xFF

Table 5. Verify Flash Memory Command, Step 1

Start ID		No. of	Command	Value				Data Bytes				Checksum
ID0	ID1	Bytes	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	CS
0x07	0x0E	0x09	V (0x56)	0x80	0x00	0x00	0x00	Data at 0x7FC	Data at 0x7FD	Data at 0x7FE	Data at 0x7FF	0x00 to 0xFF

Table 6. Verify Flash Memory Command, Step 2

Start ID		No. of	Command	Value				Data Bytes				Checksum
ID0	ID1	Bytes	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	CS
0x07	0x0E	0x09	V (0x56)	0x90	0x00	0x00	0x00	Data at 0x7F8	Data at 0x7F9	Data at 0x7FA	Data at 0x7FB	0x00 to 0xFF

Table 7. Verify Flash Memory Command, Step 3

Start ID		No. of	Command	Value				Data Bytes				Checksum
ID0	ID1	Bytes	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	CS
0x07	0x0E	0x09	V (0x56)	ADR [31:24]	ADR [23:16]	ADR [15:8]	ADR [7:0]	CRC [7:0]	CRC [15:8]	CRC [23:16]	CRC [31:24]	0x00 to 0xFF

Table 8. Remote Reset Command

Start ID		No. of	Command	Value				Checksum
ID0	ID1	Bytes	Data 1	Data 2	Data 3	Data 4	Data 5	CS
0x07	0x0E	0x05	R (0x52)	0x00	0x00	0x00	0x01	0xA8

DEVELOPMENT TOOLS

The Windows serial downloader for a Cortex™-M3 based part (CM3WSD) is a Windows software program that allows a user to serially download Intel extended hex files as created by assembler/compilers to the ADuCM350 via the serial port. The Intel extended hex file is downloaded into the on-chip flash program memory via a selected PC serial port.

SOFTWARE INSTALLATION

Copy the supplied software to the PC hard drive.

All necessary software should be installed before connecting to the device.

All subsequent steps assume that the folder has been copied directly onto the C drive.

CM3WSD.exe

The folder **Analog Devices\ADuCM350\tools\SerialDownloader\CM3WSD** provides an executable called **CM3WSD.exe**. This software accepts a hex file and allows it to be downloaded to the ADuCM350 device.

PREPARING FOR DOWNLOADING

Prepare the system for downloading by configuring the board as follows:

1. Connect the ADuCM350 evaluation board as shown in Figure 2.

Figure 2. ADuCM350 Evaluation Board Connections

The USB-to-serial communication is done via the FTDI chip (U4 on the board).

2. Plug in the USB to automatically install the board on the PC as a USB serial port. A message appears indicating that the hardware was successfully installed. The COM port assigned depends on the devices previously installed on the PC.

3. To determine the assigned COM port, open the Windows Device Manager and identify the USB serial port COM number.

   In this example, COM 10 is the allocated COM port.

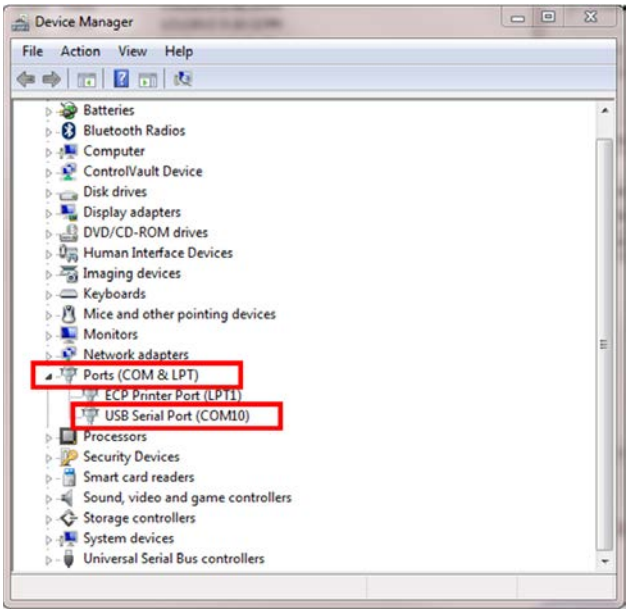

Figure 3. Device Manager Settings

4. Place the ADuCM350 into serial download mode using the following sequence:
   a. Hold down the BOOT switch on the evaluation board.
   b. Press and release the RESET switch on the evaluation board.
   c. Release the SERIAL DOWNLOAD (BOOT) switch on the evaluation board.

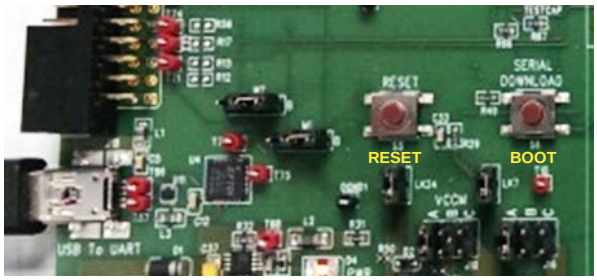

Figure 4. Boot and Reset Switches

DOWNLOADING

To begin a download,

1. Launch the Windows serial downloader by double clicking **CM3WSD.exe** in the following directory: **C:\Analog Devices\ADuCM350\tools\SerialDownloader\CM3WSD**.

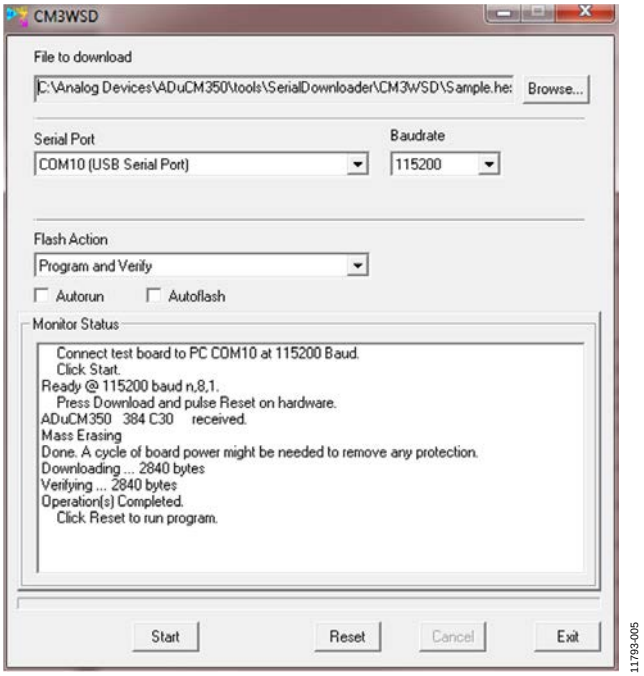

Figure 5. Downloading Using CM3WSD

2. Select the following file: **C:\Analog Devices\ADuCM350\tools\SerialDownloader\CM3WSD\Sample.hex**. The sample.hex file is a simple program that causes the LEDs marked as DISPLAY and DISPLAY1 on the board to continuously blink, indicating the successful download of code.
3. Select the correct USB serial port from the **Serial Port** box.
4. Select a baud rate of **115200** from the **Baudrate** box.
5. Click **Start**. The CM3WSD sends a reset command to the ADuCM350.
   a. If the ADuCM350 is in serial download mode and the COM port between the PC and the evaluation board is set up correctly, the CM3WSD starts downloading the sample.hex file and displays a progress bar.
   b. After the file is successfully downloaded, the **Monitor Status** box displays the message **Operation(s) Complete**.
6. Click **Reset** to run the program.
   a. The LEDs begin blinking on the evaluation board indicating that the sample.hex file has been downloaded and is executing.
   b. The **Monitor Status** box displays the message **Running**.

Notes

- If **Autorun** is selected, CM3WSD issues a reset command after flashing.
- If **Autoflash** is selected, CM3WSD repeats the current selected action until cancelled. This is useful if the same action (for example, mass erase) needs to be run on a number of parts.

NOTES

AN11793-0-11/13(0)