

UCD30xx

Boot ROM Reference Manual

Literature Number: xxxxxx

Date

1 Introduction

The UCD3000 Boot ROM provides peak and poke functionality to internal memory map space, the ability to perform flash specific tasks, the ability to download application code into the Program Flash and various checksum functionality. The Boot ROM communicates through the PMBus Interface, which is configured as a slave to listen for commands from an external USB/PMBus adapter. A series of device-specific PMBus messages have been defined to enable the various functions with the Boot ROM. This document will highlight these PMBus messages and their associated Boot ROM functions.

2 Boot ROM Function Overview

User-defined PMBus command bytes have been defined for UCD3000 Boot ROM functionality. When the Boot ROM encounters these command bytes, the Boot ROM code will process the function specified for the custom command byte. For example, a command byte of F7 indicates a Write Byte function, in which a single byte is programmed into the memory location specified in the PMBus Message.

For the UCD3000 design, the Boot ROM defaults the PMBus Interface to a device address of 0xB. The design will interpret any PMBus messages with a device address set to 0xB and one of the user-defined command bytes.

The following table introduces the supported PMBus command bytes and their associated function. These functions will be described in detail in later sections.

Boot ROM Function	Command Byte
Configure Read Address	0xFD
Read Byte	0xFC
Read Half Word	0xFB
Read Word	0xFA
Read Block	0xF9
Read Next Block	0xF8
Write Byte	0xF7
Write Half Word	0xF6
Write Word	0xF5
Write Block	0xF4
Write Next Block	0xF3
Mass Erase Flash	0xF2
Page Erase Flash	0xF1
Execute from Program Flash	0xF0
Calculate Checksum	0xEF
Read Checksum	0xEE
Read Version	0xEC

3 Peek and Poke Capability

Peek and Poke Capability allows for reading and writing of internal registers and memory within the UCD3000 through the PMBus. Reading and writing utilize different PMBus message formats when processing data. The Boot ROM allows for transfer of data in quantities as small as a single byte or as large as a block of 16 bytes. Although the PMBus protocol allows for larger block sizes, the USB/PMBus adapter used to generate the PMBus messages for communication with the UCD3000 is limited to a block size of 16 bytes maximum. Reading data from the UCD3000 through the Boot ROM requires a minimum of 2 PMBus messages in most cases. One message is used to set the read address and the second message is utilized to read out the data from the memory

location specified by the read address. The only exception to this protocol is the Read Next Block message, which auto-increments the read address originally set up prior in the initial Read Block command.

3.1 Read Functionality (Peek)

A total of 6 PMBus command bytes have been assigned for read functionality in the UCD3000 Boot ROM. Each of these read-based PMBus messages utilizes a read format, in which the PMBus Master (USB/PMBus adapter) provides an address and command byte to the PMBus Slave and expects read data returned by the slave.

3.1.1 Configure Read Address

The address where the data is read from depends on the configuration of the read address. The read address is set up through a PMBus Write Block message with a command byte of 0xFD. Starting with the device address of 0xB (with R/W bit set low), the Configure Read Address message contains a command byte of 0xFD, followed by 5 data bytes and a PEC byte. The first data byte represents the block size for a PMBus Block Write message, which will always contain a value of 0x4 for this message type. The remaining bytes represent the read address, starting with the most significant byte in the second data byte and the least significant byte in the fifth data byte. The PEC byte completes the message. The PMBus Master provides all 8 bytes of the message.

Start	Device Address & R/W (0x16)	Command Byte (0xFD)	Block Size (0x04)	Read Address[31:24]
-------	--------------------------------	------------------------	----------------------	------------------------

Read Address[24:16]	Read Address[15:8]	Read Address[7:0]	PEC	Stop
------------------------	-----------------------	----------------------	-----	------

3.1.2 Read Byte

Following the programming of the Read Address using the Configure Read Address message, any of the remaining 5 read messages can be used to read data from the internal registers. The Read Byte message reads a single byte from the address configured as the read address. The USB/PMBus adapter initiates the Read Byte message by sending a device address and a command byte of 0xFC. The Boot ROM will interpret the command byte, read a byte from the read address and complete the Read Byte Message by sending the data and a PEC.

Start	Device Address & R/W (0x16)	Command Byte (0xFC)
-------	--------------------------------	---------------------

Repeated Start	Device Address & R/W (0x17)	Data[7:0]	PEC	Stop
----------------	--------------------------------	-----------	-----	------

3.1.3 Read Half Word

The Read Half Word message reads a half word (2 bytes) from the address configured as the read address. The USB/PMBus adapter initiates the Read Half Word message by sending a device address and a command byte of 0xFB. The Boot ROM will interpret the command byte, read a half word from the read address and complete the Read Half Word Message by sending the 2 data bytes and a PEC.

Start	Device Address & R/W (0x16)	Command Byte (0xFB)
-------	--------------------------------	---------------------

Repeated Start	Device Address & R/W (0x17)	Data[15:8]	Data[7:0]	PEC	Stop
----------------	--------------------------------	------------	-----------	-----	------

3.1.4 Read Word

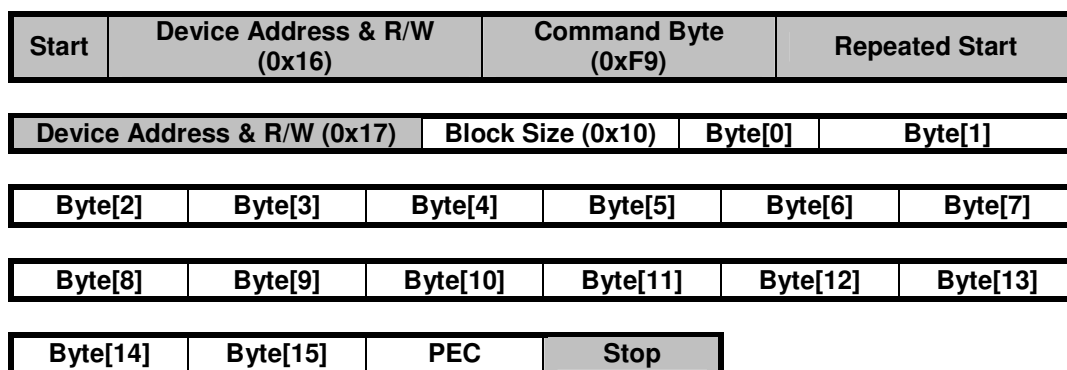
The Read Word message reads a word (4 bytes) from the address configured as the read address. The USB/PMBus adapter initiates the Read Word message by sending a device address and a command byte of 0xFA. The Boot ROM will interpret the command byte, read a word from the read address and complete the Read Word Message by sending the 4 data bytes and a PEC.

Start	Device Address & R/W (0x16)	Command Byte (0xFA)	Repeated Start	Device Address & R/W (0x17)
-------	--------------------------------	---------------------	----------------	--------------------------------

Data[31:24]	Data[23:16]	Data[15:8]	Data[7:0]	PEC	Stop
-------------	-------------	------------	-----------	-----	------

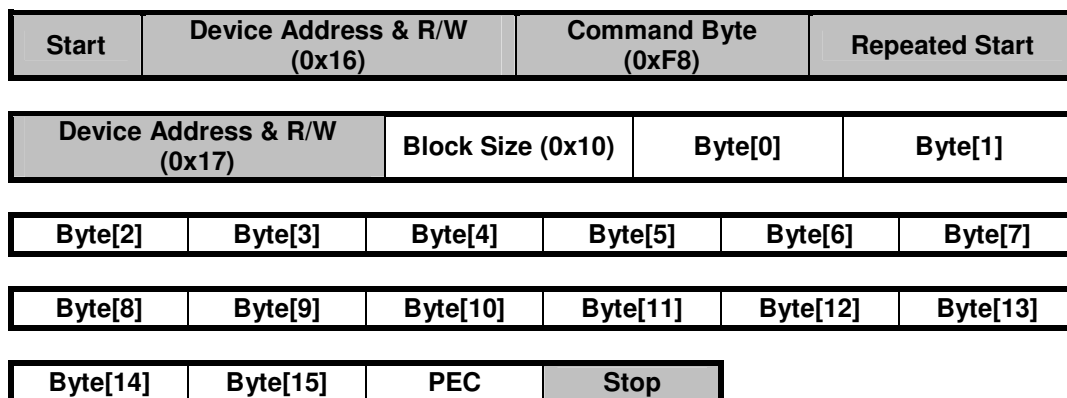
3.1.5 Read Block

The Read Block message reads a block of data (16 bytes) from the address configured as the read address. The USB/PMBus adapter initiates the Read Block message by sending a device address and a command byte of 0xF9. The Boot ROM will interpret the command byte, read 16 bytes starting at the read address and complete the Read Block Message by sending the 16 data bytes and a calculated PEC byte.



3.1.6 Read Next Block

The Read Next Block message reads a block of data (16 bytes) from the previous read address. A Read Next Block message must be preceded by either a Read Block Message or a Read Next Block Message. The read address is initially configured prior to the Read Block message that starts the data transfer. The USB/PMBus adapter initiates the Read Next Block message by sending a device address and a command byte of 0xF8. The Boot ROM will interpret the command byte, read 16 bytes starting at the new read address and complete the Read Next Block Message by sending the 16 data bytes and a calculated PEC byte. After the Read Next Block Message, the read address stored within the ROM is automatically incremented by 16.



3.1.7 Read Version

The Read Version message reads the current ROM version from the Boot ROM. The USB/PMBus adapter initiates the Read Version message by sending a device address and a command byte of 0xEC. The Boot ROM will interpret the command byte and return the block size of 4 bytes, the current version number and the PEC byte for the message. The current version for the ALR version of UCD3000 should be 0x00020002.

Start	Device Address & R/W (0x16)	Command Byte (0xEC)	Repeated Start
-------	--------------------------------	------------------------	----------------

Device Address & R/W (0x17)	Block Size (0x04)	Version[31:24]	Version[23:16]
--------------------------------	-------------------	----------------	----------------

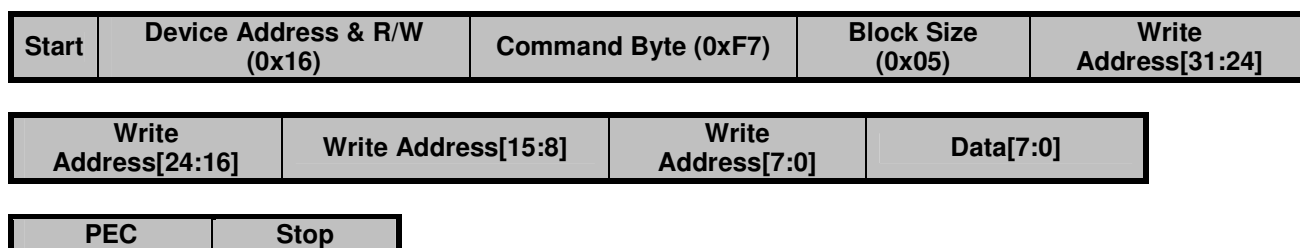
Version[15:8]	Version[7:0]	PEC	Stop
---------------	--------------	-----	------

3.2 Write Functionality (Poke)

A total of 6 PMBus command bytes have been assigned for write functionality in the UCD3000 Boot ROM. Each of these write-based PMBus messages utilizes a write format, in which the PMBus Master (USB/PMBus adapter) provides an address, the command byte and data bytes to the PMBus Slave (UCD3000).

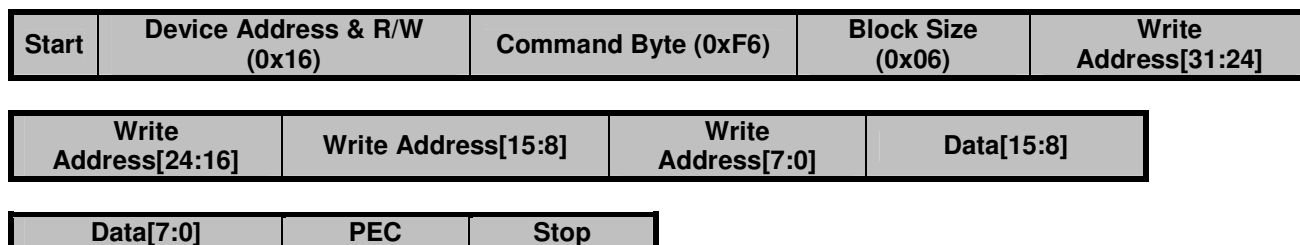
3.2.1 Write Byte

The Write Byte message programs a single byte into a specified address. The Write Byte message utilizes the PMBus write block message format. The PMBus Master (USB/PMBus adapter) initiates the message by sending the device address, a command byte of 0xF7, a block length of 0x5, the 32-bit write address location, the data byte and a PEC byte. The Boot ROM interprets the command byte and stores the single data byte at the write address specified by the incoming message.



3.2.2 Write Half Word

The Write Half Word message programs a half word into a specified address. The Write Half Word message utilizes the PMBus write block message format. The PMBus Master (USB/PMBus adapter) initiates the message by sending the device address, a command byte of 0xF6, a block length of 0x6, the 32-bit write address location, two data bytes and a PEC byte. The Boot ROM interprets the command byte and stores the half word at the write address specified by the incoming message.



3.2.3 Write Word

The Write Word message programs a 32-bit word into a specified address. The Write Word message utilizes the PMBus write block message format. The PMBus Master (USB/PMBus adapter) initiates the message by sending the device address, a command byte of 0xF5, a block length of 0x8, the 32-bit write address location, four data bytes and a PEC byte. The Boot ROM interprets the command byte and stores the word at the write address specified by the incoming message.

Start	Device Address & R/W (0x16)	Command Byte (0xF5)	Block Size (0x08)	Write Address[31:24]
-------	--------------------------------	---------------------	----------------------	-------------------------

Write Address[24:16]	Write Address[15:8]	Write Address[7:0]	Data[31:24]
----------------------	---------------------	--------------------	-------------

Data[23:16]	Data[15:8]	Data[7:0]	PEC	Stop
-------------	------------	-----------	-----	------

3.2.4 Write Block

The Write Block message programs a block of 16 data bytes starting at a specified address. The Write Block message utilizes the PMBus write block message format. The PMBus Master (USB/PMBus adapter) initiates the message by sending the device address, a command byte of 0xF4, a block length of 0x14, the 32-bit write address location, sixteen data bytes and a PEC byte. The Boot ROM interprets the command byte and stores the data bytes starting at the write address specified by the incoming message.

Start	Device Address & R/W (0x16)	Command Byte (0xF4)	Block Size (0x14)	Write Address[31:24]
-------	--------------------------------	---------------------	----------------------	-------------------------

Write Address[24:16]	Write Address[15:8]	Write Address[7:0]	Data Byte[0]
----------------------	---------------------	--------------------	--------------

Data Byte[1]	Data Byte[2]	Data Byte[3]	Data Byte[4]
--------------	--------------	--------------	--------------

Data Byte[5]	Data Byte[6]	Data Byte[7]	Data Byte[8]
--------------	--------------	--------------	--------------

Data Byte[9]	Data Byte[10]	Data Byte[11]	Data Byte[12]
--------------	---------------	---------------	---------------

Data Byte[13]	Data Byte[14]	Data Byte[15]	PEC	Stop
---------------	---------------	---------------	-----	------

3.2.5 Write Next Block

The Write Next Block message programs a block of 16 data bytes starting at the last programmed write address incremented by 16. The Write Next Block message utilizes the PMBus write block message format. Use of the

Write Next Block message assumes the write address was set previously with a Write Block message. The PMBus Master (USB/PMBus adapter) initiates the message by sending the device address, a command byte of 0xF3, a block length of 0x10, sixteen data bytes and a PEC byte. The Boot ROM interprets the command byte and stores the data bytes starting at the write address, calculated from the previously set address incremented by 16.

Start	Device Address & R/W (0x16)	Command Byte (0xF3)	Block Size (0x10)	Data Byte[0]
Data Byte[1]	Data Byte[2]	Data Byte[3]	Data Byte[4]	
Data Byte[5]	Data Byte[6]	Data Byte[7]	Data Byte[8]	
Data Byte[9]	Data Byte[10]	Data Byte[11]	Data Byte[12]	
Data Byte[13]	Data Byte[14]	Data Byte[15]	PEC	Stop

4 Flash Functions

In addition to the ability to read and write internal memory map locations, the UCD3000 Boot ROM supports various flash functions through user-defined PMBus command bytes. These functions allow the user to perform a mass erase or page erase of either the Program Flash or Data Flash. The Boot ROM also supports a PMBus command byte that initiates execution of code from the Program Flash, starting at address 0.

4.1 Mass Erase

The Boot ROM supports the initiation of a mass erase of either the Program or Data Flash through a PMBus message. The PMBus Master (USB/PMBus adapter) initiates a PMBus write byte message to the UCD3000. The master initiates the message by sending the device address, a command byte of 0xF2, a single data byte and the PEC byte. The data byte identifies which flash memory will be mass erased. A value of 0x0 indicates a mass erase of the Data Flash, while a value of 0x1 indicates a mass erase of the Program Flash. Upon receipt of the mass erase message, the Boot ROM sets the appropriate control bit to initiate a mass erase of the flash memory. The flash control logic generates the sequencing of control signals to perform a mass erase operation on the memory.

Start	Device Address & R/W (0x16)	Command Byte (0xF2)	Data Byte (0/1)	PEC	Stop
-------	-----------------------------	---------------------	-----------------	-----	------

4.2 Page Erase

The Boot ROM supports the initiation of a page erase of either the Program or Data Flash through a PMBus message. The PMBus Master (USB/PMBus adapter) initiates a PMBus write block message to the UCD3000. The master initiates the message by sending the device address, a command byte of 0xF1, a block length of 0x4, four data bytes and a PEC byte. The first data byte selects which flash memory in which the page erase will be performed. The remaining 3 data bytes select the page to be erased. Upon receipt of the page erase message, the Boot ROM sets the appropriate control bits to initiate a page erase on the selected flash memory. The flash control logic generates the sequencing of control signals to perform a page erase operation.

Start	Device Address & R/W (0x16)	Command Byte (0xF1)	Block Length (0x4)
-------	-----------------------------	---------------------	--------------------

Flash Select (0/1)	Page Select[23:16]	Page Select[15:8]	Page Select[7:0]	PEC	Stop
--------------------	--------------------	-------------------	------------------	-----	------

4.3 Execute Flash

The Boot ROM supports a function to exit the ROM code and start execution of code from the Program Flash through a PMBus message. The USB/PMBus adapter initiates a PMBus Send Byte message to the PMBus slave on the UCD3000. The master starts the message by sending the device address of 0xB, a command byte of 0xF0 and a PEC byte. Upon receipt of the user-defined command byte of 0xF0, the Boot ROM configures the memory selects for program flash operation and resets the program counter to 0. Following reconfiguration of the memory selects, the Program Flash now resides at address location 0 and code is now read from the flash instead of Boot ROM code.

Start	Device Address & R/W (0x16)	Command Byte (0xF0)	PEC	Stop
-------	--------------------------------	---------------------	-----	------

Upon power-up, the Boot ROM code does check for a checksum found at the last address location in the Program Flash. If this checksum stored in the Program Flash matches the calculated checksum by the ROM, the Boot ROM will automatically execute from the Program Flash without waiting for a PMBus message. The checksum should be stored upon programming of the Program Flash with application code.

4.4 Flash Loading

Through the use of user-defined PMBus command bytes, the Boot ROM supports loading of code into the Program and Data Flash memories through the PMBus interface. To start the process of loading code into the flash, a mass erase of the flash memory is necessary to ensure the program cycle is successful. The mass erase of the Program Flash is initiated through a PMBus message, as described in Section 4.1. The data byte in this message is set 0x01 for the Program Flash. The mass erase of the Program Flash requires 20ms to complete following completion of the PMBus message.

After performing the mass erase of the Program Flash, the application code is loaded into the flash through a series of Write Block and Write Next Block messages (as described in Sections 3.2.4 and Section 3.2.5).

The flash loading is started using a Write Block message, with the write address set to 0x10000. Due to limitations of the USB/PMBus adapter, block size for the Write Block and Write Next Block messages are limited to 16 bytes. Therefore, 4 words with the Program Flash are programmed with each message. A total of 2048 messages are required to program the entire flash.

Following the initial Write Block message, the remaining messages will be Write Next Block messages, consisting of 16 bytes each. The code compiler output file, such as a Tektronix file, will need to be preprocessed and block in the 16-byte blocks to allow for programming through the PMBus interface. In addition to the application code, the final location in the Program Flash must be programmed with the checksum for the entire Program Flash. Therefore, after the next reset cycle, the Boot ROM code will turn control over to the Program Flash and run application code.

5 Checksum Functions

The UCD3000 Boot ROM supports checksum generation, used primarily to verify programmed flash memories. A PMBus command byte of 0xEF has been assigned for calculation of a checksum, while a command byte of 0xEE is utilized for reading the previously calculated checksum.

5.1 Calculation of Checksum

The Boot ROM supports a function to calculate a checksum over a portion of the address space. The USB/PMBus adapter initiates a PMBus write block message to initiate a checksum calculation within the UCD3000. The PMBus master sends the device address, a command byte of 0xEF, the starting address of the checksum calculation, the number of bytes to process and a PEC byte. Upon detection of the command byte 0xEF, the Boot ROM reads the block of memory space and calculates a checksum.

Start	Device Address & R/W (0x16)	Command Byte (0xEF)	Block Size (0x08)	Start Address[31:24]
Start Address[24:16]	Start Address[15:8]	Start Address[7:0]	Byte Count[31:24]	
Byte Count[23:16]	Byte Count[15:8]	Byte Count[7:0]	PEC	Stop

5.2 Reading Checksum

The Boot ROM supports a function to read a previously calculated checksum over a portion of the address space. The USB/PMBus adapter initiates a PMBus read block message to read a checksum calculation within the UCD3000. The PMBus master sends the device address, a command byte of 0xEE. The UCD3000 returns the four checksum bytes and a PEC byte. Upon detection of the command byte 0xEE, the Boot ROM reads the calculated checksum and returns to the PMBus master.

Start	Device Address & R/W (0x16)	Command Byte (0xEE)	Block Size (0x04)	Checksum[31:24]
Checksum[23:16]	Checksum[15:8]	Checksum[7:0]	PEC	Stop

6 System Initialization/Information

The UCD3000 Boot ROM also supports basic system initialization. After power-up, the Boot ROM will automatically perform basic system initialization. The Boot ROM will set up the clock generation logic, configuring the ICLK (peripheral clock) to run at half of MCLK (ARM System Clock). For example, if MCLK is configured to run at 32MHz, ICLK will run at 16MHz.

After clock configuration, the memory map locations for the memories are set as follows:

0x00000 – 0x00FFF => Boot ROM
0x10000 – 0x17FFF => Program Flash
0x18800 – 0x188FF => Data Flash
0x19000 – 0x19FFF => Data RAM

After configuration of the memory map space for the memories in the system, the Boot ROM turns off the peripheral reset. Following the de-assertion of reset to the peripherals, the Boot ROM will check the checksum of the Program Flash, located in the last 32-bit word of the Program Flash. If the calculated checksum of the Program Flash performed by the Boot ROM matches the value programmed into the last word of the Program Flash (0x17FFC), the Boot ROM automatically cedes control over to the Program Flash. If not valid, the Boot ROM enters a loop, waiting for further direction from the PMBus Interface.

The ROM version of the Boot ROM can be obtained through a PMBus message, using command byte 0xEC. The PMBus Master uses a PMBus Read Block message to obtain the version code from the ROM. The message starts with the device address and the command byte of 0xEC. The Boot ROM returns a block length of 0x04, 4 bytes representing the ROM version and a PEC byte.

Start	Device Address (0x16)	Command Byte(0xEC)	Block Length (0x4)	ROM Version[31:24]
ROM Version[23:16]	ROM Version[15:8]	ROM Version[7:0]	PEC	Stop