



TMS28x CAN Bootloader

Functional Specification

FS-0057

About Oztek

Oztek Corp. is proven innovator of power, control, and instrumentation solutions for the most demanding industrial applications. Oztek products include variable motor drives, grid tie inverters, frequency converters, standalone inverters, DC/DC converters, and DSP based control boards for power control applications.

Trademarks

OZDSP1000, OZDSP1100, OZDSP2000, and OZDSP3000 are trademarks of Oztek Corp. Other trademarks, registered trademarks, and product names are the property of their respective owners and are used herein for identification purposes only.

Notice of Copyright

CAN Downloader User's Manual © August 2010 Oztek Corp. All rights reserved.

Exclusion for Documentation

UNLESS SPECIFICALLY AGREED TO IN WRITING, Oztek Corp. ("Oztek")

- (A) MAKES NO WARRANTY AS TO THE ACCURACY, SUFFICIENCY OR SUITABILITY OF ANY TECHNICAL OR OTHER INFORMATION PROVIDED IN ITS MANUALS OR OTHER DOCUMENTATION.

- (B) ASSUMES NO RESPONSIBILITY OR LIABILITY FOR LOSSES, DAMAGES, COSTS OR EXPENSES, WHETHER SPECIAL, DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL, WHICH MIGHT ARISE OUT OF THE USE OF SUCH INFORMATION. THE USE OF ANY SUCH INFORMATION WILL BE ENTIRELY AT THE USER'S RISK.

- (C) IF THIS MANUAL IS IN ANY LANGUAGE OTHER THAN ENGLISH, ALTHOUGH STEPS HAVE BEEN TAKEN TO MAINTAIN THE ACCURACY OF THE TRANSLATION, THE ACCURACY CANNOT BE GUARANTEED. APPROVED OZTEK CONTENT IS CONTAINED WITH THE ENGLISH LANGUAGE VERSION WHICH IS POSTED AT WWW.OZTEKCORP.COM.

Date and Revision

November 2013 Rev B

Part Number

FS-0057

Contact Information

USA

Telephone: 603-546-0090

Fax: 603-386-6366

Email techsupport@oztekc corp.com

Table of Contents

1. Introduction	3
1.1 Referenced Documents	3
1.2 Definitions	3
1.3 Bootloader Memory Configuration	4
1.4 Flash Memory Map.....	4
1.5 Configuration Registers	4
1.6 EEPROM - Bootloader.....	5
1.7 EEPROM - Data	5
2. CCP Command Subset	5
3. Description of Commands	6
3.1 Connect Command	7
3.2 Set Memory Transfer Address (MTA) Command	8
3.3 Download Command	9
3.4 Download 6 Bytes Command	9
3.5 Disconnect Command.....	10
3.6 Build Checksum Command.....	10
3.6.1 CRC Algorithm	11
3.6.2 S-Record Checksum.....	11
3.6.3 Full Image Checksum.....	11
3.7 Clear Memory Command	12
3.8 Unlock Command	12
3.9 Action Service Request Command	13
4. Example Command Sequences.....	14
4.1 Program DSP Flash Memory	14
4.2 Updating Serial EEPROMs.....	17
5. Bootloader and Application Interaction.....	19
5.1 Bootloader Version Number.....	20
5.2 Forcing a Modified Reboot	20
5.3 Code to Trip the Watchdog	21
6. CCP Over Modbus	21
6.1 Serial Bus Configuration	22
6.2 Modbus Register Map	22
6.3 Sending CRO Commands	23
6.3.1 Downloading Complete S-Records.....	24
6.4 Reading DTO Responses	26
Warranty and Product Information.....	28
Return Material Authorization Policy.....	30

Table of Tables

Table 1 - Flash Memory Map	4
Table 2 - Configuration Registers	5
Table 3 - Supported CCP Command Codes	5
Table 4 - Command Return Code Status	6
Table 5 - Categories of Errors	6
Table 6 - CCP packet format	7
Table 7 - Connect Command Format	7
Table 8 - DSP Part ID Codes	8
Table 9 - Set MTA Command Format	8
Table 10 - Bootloader Address Extensions	8
Table 11 - Download Command Format	9
Table 12 - Download-6 Command Format	9
Table 13 - Disconnect Command Format	10
Table 14 - Build Checksum Command Format	10
Table 15 - Clear Memory Command Format	12
Table 16 - Unlock Command Format	13
Table 17 - Action Service Request Command Format	13
Table 18 - Bit Rate Enumeration	14
Table 19 - Modbus Register Map	23
Table 20 - Modbus CRO Command Byte Sequence	24
Table 21 - Custom "Program S-Record" CCP Command Format	25
Table 22 - Modbus DTO Read Request Byte Sequence	26
Table 23 - Modbus DTO Read Response Byte Sequence	27

1. Introduction

This document describes the communication protocol supported by the Oztek SPI Bootloader applications for the Texas Instruments C2000 microcontroller family.

The Bootloader is a small program residing in a serial EEPROM device on the control board which is automatically loaded into the DSP RAM upon reset. For this to occur, the device's boot mode must be set to "boot from SPI". Once loaded, the program loops for a finite amount of time, waiting for a "connect" command from a host controller. Once connected, additional commands are then sent from the host that allow the DSP's internal Flash memory to be programmed. Provisions are also made for upgrading the Bootloader itself. Upon a timeout, the program then executes the application code residing in Flash.

The protocol implemented in the Bootloader is a subset of the ASAP Standard CAN Calibration Protocol (CCP) Version 2.1. In order to maintain the smallest possible memory footprint, only the commands required for programming memory (Flash or Serial EEPROM) are implemented. This document describes each of the supported CCP commands as well as examples of the command sequences required to program Flash/Serial EEPROM.

The CCP protocol has also been adapted for use over a serial communications link such as RS-232/422/485 using the Modbus protocol. This is described in more detail in Section 6 (CCP Over Modbus).

1.1 Referenced Documents

Ref.	Document	Description
[1]	UM-0015	Oztek Image Downloader Utility User's Manual
[2]	CCP Version 2.1	ASAP CAN Calibration Protocol (http://www.asam.net)
[3]	PI-MBUS-300 Rev J	Modicon Modbus Protocol Ref Guide (http://modbus.org)

1.2 Definitions

CAN	Controller Area Network
CRC	Cyclic Redundancy Check
DSP	Digital signal processor
EEPROM	Electrically Erasable Programmable Read Only Memory
GUI	Graphical User Interface
PCB	Printed Circuit Board
POR	Power On Reset
RAM	Random Access Memory
USB	Universal Serial Bus

1.3 Bootloader Memory Configuration

The Bootloader includes three separate sections of memory that can be accessed by the Host application, namely Configuration registers, Flash Memory, and Serial EEPROM memory. These three memory sections are accessed via the CCP commands using a 32-bit absolute address as well as an 8-bit extension to distinguish between the various devices/regions. See Section 3.2 (Set Memory Transfer Address (MTA) Command) for more details on addressing these memory regions.

1.4 Flash Memory Map

The table below shows the internal Flash memory map of the Texas Instruments C2000 devices supported by the various Oztek Bootloader applications. When clearing Flash using the CCP “Clear Memory” command, the *Memory Size* command parameter is used to select the desired Flash sectors using a right justified, bit defined mask. For example, if all sectors are to be erased the mask for a 28335 is 0x00FF. If only sectors A, C, and D are to be erased the mask is 0x000D.

Table 1 - Flash Memory Map

Memory Sector	Starting Address							
	2801	2802/6	2808	2809	28332	28334	28335	28066/7/8/9
SECTOR A	0x3F7000	0x3F6000	0x3F4000	0x3F4000	0x33C000	0x33C000	0x338000	0x3F4000
SECTOR B	0x3F6000	0x3F4000	0x3F0000	0x3F0000	0x338000	0x338000	0x330000	0x3F0000
SECTOR C	0x3F5000	0x3F2000	0x3EC000	0x3EC000	0x334000	0x334000	0x328000	0x3EC000
SECTOR D	0x3F4000	0x3F0000	0x3E8000	0x3E8000	0x330000	0x330000	0x320000	0x3E8000
SECTOR E	n/a	n/a	n/a	0x3E4000	n/a	0x32C000	0x318000	0x3E4000
SECTOR F	n/a	n/a	n/a	0x3E0000	n/a	0x328000	0x310000	0x3E0000
SECTOR G	n/a	n/a	n/a	0x3DC000	n/a	0x324000	0x308000	0x3DC000
SECTOR H	n/a	n/a	n/a	0x3D8000	n/a	0x320000	0x300000	0x3D8000

1.5 Configuration Registers

The CCP configuration registers allow the host application to exchange information with the Bootloader firmware. The only configuration register implemented by the Bootloader application is the “Password” register. This register resides at configuration register address 0x00000000, and it is used to unlock access to the various programmable memories. The table below shows the password values required to unlock each of the available memory types. See section 3.8 (Unlock Command) for further details on setting the password register.

Table 2 - Configuration Registers

<i>Address</i>	<i>Register</i>	<i>Value</i>
0x00000000	Password	DSP Flash = 0x1234
		EEPROM (Bootloader) = 0xABCD
		EEPROM (Data) = 0x9876

1.6 EEPROM - Bootloader

The Bootloader EEPROM contains the Bootloader program's executable image. As such, it should only be programmed with a valid Bootloader image. When clearing the EEPROM, it must be completely cleared to remove any non-volatile parameters stored in the upper EEPROM addresses. For this reason, the *Memory Size* parameter of the CCP "Clear Memory" command is a *Don't Care* as the Bootloader will always erase the complete EEPROM. Due to the Bootloader code size, the EEPROM device on the board must be at least 128 Kb in size (16K x 8 bits).

1.7 EEPROM - Data

The Data EEPROM is an optional memory that is available on some Oztek control boards for use by the DSP runtime application. The Bootloader allows data to be written to this EEPROM device. The Downloader GUI supports reprogramming of this device by downloading a data file in the Motorola S-record hex file format. Clearing this EEPROM is similar to the Bootloader EEPROM in that the *Memory Size* parameter of the CCP "Clear Memory" command is a *Don't Care* as the Bootloader will always erase the complete EEPROM.

2. CCP Command Subset

The CCP protocol uses a "Command Receive Object" message for sending various commands to the embedded slave device. This section describes the CCP CRO commands supported by the Oztek Bootloader applications. These are summarized in the table below, along with the maximum timeout values that should be used by the host when waiting for a response from the Bootloader.

Table 3 - Supported CCP Command Codes

<i>Command</i>	<i>Code</i>	<i>Max Timeout (ms)</i>
CONNECT	0x01	1000
SET_MTA	0x02	1000
DOWNLOAD	0x03	1000
DISCONNECT	0x07	1000
BUILD_CHECKSUM	0x0E	1000
CLEAR_MEMORY	0x10	30,000
UNLOCK	0x13	1000
ACTION_SERVICE	0x21	1000
DOWNLOAD_6	0x23	1000

For every CRO command message received, the Bootloader will respond with the “Data Transmission Object” response message as described in the CCP protocol. The table below shows the list of DTO return codes supported by the Bootloader.

Table 4 - Command Return Code Status

Code	Description	Error Category	State Transition to
0x00	Acknowledge / No error	-	-
0x10	Processor Busy	C1	NONE (wait)
0x30	Unknown command	C3	FAULT
0x31	Command syntax	C3	FAULT
0x32	Parameter(s) out of range	C3	FAULT
0x33	Access denied	C3	FAULT
0x35	Access locked	C3	FAULT
0x36	Resource/Function unavailable	C3	FAULT
0x7F	Operational Failure	C3	FAULT

The table below is provided for reference only – it summarizes the CCP error code categories according to their severity and the intended resulting action.

Table 5 - Categories of Errors

Category	Description	Action	Retries
timeout	No handshake message	retry	2
C0	Warning	-	-
C1	Spurious (comm. Error, busy, ..)	Wait (ACK or timeout)	2
C2	Resolvable (temp power loss,...)	reinitialize	1
C3	Unresolvable (setup, overload, ..)	terminate	-

3. Description of Commands

For a detailed description of the CCP commands supported by the Bootloader, please refer to the CAN Calibration protocol Version 2.1. The protocol is available at <http://www.asam.net>. The following paragraphs describe any extensions made to the CCP standard.

All commands sent to the Bootloader should use the 29-bit extended CAN Message ID format. The Bootloader will only respond to IDs that match this pattern: 0x0B081880. The typical message format for a CCP command is shown below:

Table 6 - CCP packet format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	Command Code	Packet ID (always 0xFF)
1	Command Counter	Command Return Code
2	Data Payload	Command Counter
3		Data Payload
4		
5		
6		
7		

When sending a DTO response to the host, the Bootloader will always respond with a Packet ID of 0xFF and the Command Counter value returned will be the same value that was received in the corresponding CRO command from the host.

3.1 Connect Command

The connect command is sent by the host to the Bootloader when first attempting to establish connection. If this message is received within the initial timeout period, the Bootloader will remain connected to the host rather than automatically jumping to the application stored in Flash. None of the CRO data payload bytes are used by the Bootloader and are treated as “don’t cares”. For the DTO response, data bytes 3 and 4 are used to provide revision information for the Bootloader image currently executing on the DSP board. The DSP part ID is returned in bytes 5 and 6.

Table 7 - Connect Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x01	0xFF
1	CNT	Return Code
2	don't care	CNT
3	don't care	Major Bootloader Revision
4	don't care	Minor Bootloader Revision
5	don't care	Major DSP Part Id
6	don't care	Minor DSP Part Id
7	don't care	0x00

The DSP part ID is defined by TI and is stored at a fixed read-only memory location within the device (typically 0x00380090). The ID is returned to provide error checking before loading images into Flash. A summary of various part IDs as defined by Texas Instruments are provided in the following table:

Table 8 - DSP Part ID Codes

<i>Value</i>	<i>DSP Type</i>
0x002C	2801
0x0024	2802
0x0034	2806
0x003C	2808
0x00FE	2809
0x00E6	28232
0x00E7	28234
0x00E8	28235
0x00ED	28332
0x00EE	28334
0x00EF	28335
0x009E	28069PZP
0x009F	28069UPZ
0x009C	28069PPF
0x009D	28069UPFP

3.2 Set Memory Transfer Address (MTA) Command

The Set MTA command is sent by the host to the Bootloader to indicate the starting address for any subsequent memory transfers.

Table 9 - Set MTA Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x02	0xFF
1	CNT	Return Code
2	0x00	CNT
3	ADDR EXT	0x00
4	ADDR [31:24]	0x00
5	ADDR [23:16]	0x00
6	ADDR [15:8]	0x00
7	ADDR [7:0]	0x00

Byte 2 (aka “MTA Number”) is expected to be 0x00 for the memory transfer commands that are supported by the Bootloader. Byte 3 (aka “Address Extension”) is used to select which memory device/region is being accessed, according to the following table:

Table 10 - Bootloader Address Extensions

<i>Address Extension</i>	<i>Memory Space</i>
0x00	Flash Memory
0x01	Configuration Registers
0x02	EEPROM Memory (Bootloader)
0x03	EEPROM Memory (Data)

The 32-bit address is then provided in bytes 4-7 of the CRO. The DTO response consists simply of the return code and the command count; the payload data is not used and is set to all zeros.

3.3 Download Command

This command is sent by the host to the Bootloader to transfer anywhere from one to five bytes of data to the selected memory. Byte 2 of the CRO command indicates how many data bytes are included in the message to be downloaded to the selected device. Bytes 3 through 7 contain the bytes to be written (actual number of valid bytes indicated by the size specified in Byte 2. If the device selected by the Set MTA command is unlocked and the memory address is valid, the download bytes will be stored in the Bootloader's S-Record data buffer. Note that the memory address as sent by the Set MTA command will be auto-incremented for each valid byte received.

Table 11 - Download Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x03	0xFF
1	CNT	Return Code
2	SIZE	CNT
3	BYTE 0	ADDR EXT
4	BYTE 1	ADDR [31:24]
5	BYTE 2	ADDR [23:16]
6	BYTE 3	ADDR [15:8]
7	BYTE 4	ADDR [7:0]

The DTO response contains the command return code, command count, as well as the memory address extension for the presently selected device and the next available memory address.

3.4 Download 6 Bytes Command

This command is similar to the download command described above except that it is always used to transfer exactly 6 bytes of data. Note that the memory will be auto-incremented by 6 following a valid transfer from this command.

Table 12 - Download-6 Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x23	0xFF
1	CNT	Return Code
2	BYTE 0	CNT
3	BYTE 1	ADDR EXT
4	BYTE 2	ADDR [31:24]
5	BYTE 3	ADDR [23:16]
6	BYTE 4	ADDR [15:8]
7	BYTE 5	ADDR [7:0]

3.5 Disconnect Command

When the Bootloader receives this command, it will check for a valid Flash image. If one is found, it will return an acknowledge message indicating the message was received. It will then immediately load and run the program that is stored in Flash. If no valid flash image is found, an error message will be returned, and the boot loader will remain connected, waiting for new CCP messages to be received. The DTO response consists simply of the return code and the command count; the payload data is not used and is set to all zeros.

Table 13 - Disconnect Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x07	0xFF
1	CNT	Return Code
2	don't care	CNT
3	don't care	0x00
4	don't care	0x00
5	don't care	0x00
6	don't care	0x00
7	don't care	0x00

3.6 Build Checksum Command

This command is used to verify the Host Controller's Cyclic Redundancy Check (CRC) with the Boot loader's for both a single S-record as well as for the whole image.

Table 14 - Build Checksum Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x0E	0xFF
1	CNT	Return Code
2	SIZE [31:24]	CNT
3	SIZE [23:16]	ADDR EXT
4	SIZE [15:8]	CRC [15:0]
5	SIZE [7:0]	CRC [7:0]
6	CRC [15:0]	0x00
7	CRC [7:0]	0x00

Bytes 2 through 5 of the CRO payload are used by the host to indicate the size of the checksum operation. A size of zero indicates that the CRC for the complete image should be checked. Any non-zero size indicates that the CRC for the most recently received S-Record should be checked. In both cases, the host sends the CRC value that it has calculated in CRO bytes 6 and 7. These

are then read by the Bootloader and compared to the CRC value that it has calculated as it received data from the host. If the two values match, the Bootloader will send a DTO response with a “no error” response code, along with the memory address extension for the selected memory in byte 3 and a copy of its local CRC value in bytes 4 and 5.

3.6.1 CRC Algorithm

The CRC value is calculated as a 16-bit CRC. The algorithm is given here:

```
crc = 0xFFFF;
crc = crc ^ (byte & 0x00FF);

for ( i = 0; i < 8; i++ )
{
    if ((crc & 0x0001) != 0)
        crc = (crc >> 1) ^ 0xA001;
    else
        crc = (crc >> 1);
}
```

3.6.2 S-Record Checksum

The S-Record checksum should be sent after an entire S-Record has been sent to the Bootloader via the DOWNLOAD and DOWNLOAD_6 commands. The size field must be greater than 0 (the actual value is not used by the Bootloader). If the calculated CRCs of both the Host and the Boot loader match, the S-Record will be programmed into the selected memory. If they do not match, an error will be returned to the host.

3.6.3 Full Image Checksum

After the entire image is programmed, the host should send the Build Checksum command with the size field set to zero. The Bootloader will then compare its full image CRC to that of the Host. If they do not match, an error will be returned.

If the CRCs do match, and a Flash image was programmed, then a special CRC value will be calculated. The CRC is calculated using every 128th memory location in Flash, starting from the first Flash memory location and continuing to the end of the Flash memory region. This CRC is stored in the Bootloader EEPROM at the 2 highest byte addresses of the EEPROM itself.

This CRC is used by the Bootloader to determine if a valid image exists in Flash. Before attempting to jump to Flash, the Bootloader will recalculate the CRC across the Flash memory and compare the calculated value to the value stored in the EEPROM – if the two match, the Flash image is considered valid and the Bootloader will allow jumping to the Flash application. If the two values do not match (i.e. Flash was never programmed or has since been corrupted),

the Bootloader will not attempt to run the image in Flash, and will instead remain connected and waiting for additional CCP commands.

3.7 Clear Memory Command

This command is sent to the Bootloader from the Host to initiate a clear operation of the presently selected memory device. If the selected memory device is one of the EEPROMs, the SIZE field (bytes 2-5) are ignored, and instead the Bootloader will erase the entire contents of the EEPROM. If the selected memory device is Flash, the Flash device can only be erased by sectors. For this reason, the SIZE field is interpreted as a sector mask. If bit 0 is set, then Sector A will be erased; if bit 1 is set, then Sector B will be erased, etc. The DTO response consists simply of the return code and the command count; the payload data is not used and is set to all zeros.

Table 15 - Clear Memory Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x10	0xFF
1	CNT	Return Code
2	SIZE [31:24]	CNT
3	SIZE [23:16]	0x00
4	SIZE [15:8]	0x00
5	SIZE [7:0]	0x00
6	don't care	0x00
7	don't care	0x00

3.8 Unlock Command

This command is sent to the Bootloader by the host to attempt to unlock access to one of the programmable images. This command is used to set the password value in the “Password Configuration Register”. For this reason, a Set MTA command must first be sent to select the Configuration Register memory address extension (ADDR EXT = 1) and a configuration register address of zero (ADDR [31:0] = 0).

Byte 2 in the CRO command indicates the number of data bytes to write, and this value should be set to 2 to indicate a 2-byte password value is being written. The password is then provided in bytes 3 and 4. The password value is then compared against the hard-coded passwords for each memory type (see Table 2 for password values). If a match is found, access to the corresponding memory is allowed and a “no error” DTO response will be sent, along with the present memory address extension.

Table 16 - Unlock Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x13	0xFF
1	CNT	Return Code
2	0x02	CNT
3	PW [15:8]	ADDR EXT
4	PW [7:0]	0x00
5	don't care	0x00
6	don't care	0x00
7	don't care	0x02

3.9 Action Service Request Command

The Bootloader uses the Action Service request command to allow the Host to change the bit rate of the communications interface (i.e. CAN bus or Serial Port). Bytes 2 and 3 of the CRO command define the “Action Type” – this value must be set to 0x81 to indicate a request to change the download bit rate. All other Action Types are ignored.

Table 17 - Action Service Request Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x21	n/a
1	CNT	n/a
2	ACT NUM [15:8]	n/a
3	ACT NUM [7:0]	n/a
4	DATA [15:8]	n/a
5	DATA [7:0]	n/a
6	don't care	n/a
7	don't care	n/a

CRO bytes 4 and 5 contain the action data – in this case the enumerated value for the desired bit rate setting. Valid bit rates are shown in the table below, with one column showing the standard values for the CAN-based Bootloaders and a second column showing the standard values for the Serial RS-232/422/485-based Bootloaders.

Table 18 - Bit Rate Enumeration

<i>Value</i>	<i>CAN Bit Rate</i>	<i>Serial Baud Rate</i>
0	1 Mbps	2400
1	500 Kbps	4800
2	250 Kpbs	19200
3	125 Kpbs	38400
4	100 Kbps	57600
5	50 Kbps	115200

Note that upon power-up, the Bootloader automatically resets the serial bit rate to a known value. For CAN-based Bootloaders, the default CAN bit rate is 250 kbps. For Serial-based Bootloaders, the default baud rate is 19200. Once the bit rate has been changed, the Bootloader does not send a DTO response – instead, it reconfigures the CAN or serial interface with the newly selected bit rate and re-initializes the Bootloader application. At this point, the Host should not expect a DTO response, but should rather go back and attempt to reconnect to the Bootloader. When the Bootloader application resets, the connect timeout is extended to 30 seconds, so the host has up to 30 seconds to reconnect to the Bootloader or else the Bootloader will attempt to jump to Flash.

Once the bit rate has been changed, the Bootloader will continue to communicate at the new rate for all subsequent CCP operations. Note that the Bootloader will remain at this bit rate until either a) changed by the user using a new Action Service Request command or b) by cycling the power on the control board. Note that when cycling power, the Bootloader will revert back to the default bit rates and it will be necessary for the Host to change back to the original default bit rates if it wants to reconnect.

4. Example Command Sequences

This section provides example command sequences for the common Bootloader operations in order to illustrate how to use the CCP commands and what to expect in response.

4.1 Program DSP Flash Memory

Programming the DSP's Flash memory involves connecting to the Bootloader in the DSP, setting up the Bootloader's configuration registers, erasing the Flash, and finally programming the Flash. The process is described below for a 28335 DSP.

1. Host Connects to DSP Bootloader Application:

Connect Command:

DSP Station Address = 0x0001

Host Command (CRO):	0x01	CTR	0x01	0x00	--	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	Maj Rev	Min Rev	Maj Part ID	Min Part ID	--

2. **Host writes the password for Flash access by writing to the password configuration register. First, set the pointer (MTA) to the correct register address:**

Set MTA Command:

MTA# = 0x00, Address Ext = 0x01(Cfg Reg Space), Address = 0x00000000
(Password addr)

Host Command (CRO):	0x02	CTR	0x00	0x01	0x00	0x00	0x00	0x00
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

3. **Then write the password to the configuration register:**

Unlock Command:

Data size(bytes) = 0x02, PASSWORD = 0x1234

Host Command (CRO):	0x13	CTR	0x02	0x12	0x34	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x01	0x00	0x00	0x00	0x02

4. **Set MTA to Flash Space for Clear Command:**

Set MTA Command:

MTA# = 0x00, Address Ext = 0x00 (Flash Space)

Host Command (CRO):	0x02	CTR	0x00	0x00	--	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

5. **Erase the Flash by giving the Clear Memory command including the Sector Mask:**

Clear Memory Command:

Flash Sector Selection Mask = 0x000000FF (erase all 8 sectors)

Host Command (CRO):	0x10	CTR	0x00	0x00	0x00	0xFF	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

6. **Program Flash by repeatedly issuing Set MTA commands to set the Address followed by Download 6 Bytes or Download commands until all desired memory locations are programmed. Since the MTA is auto-incremented by the DSP it only needs to be set by the Host when the data is not sequential (i.e. at the start of each S-Record). The Build Check Sum command should be issued to actually load the data into Flash (i.e. after each S-Record has been downloaded).**

Set MTA Command:*MTA# = 0x00, Address Ext = 0x00, Address = 0x00338000*

Host Command (CRO):	0x02	CTR	0x00	0x00	0x00	0x33	0x80	0x00
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

Download 6 Command:*Data = 0x01, 0x02, 0x03, 0x04, 0x05, 0x06*

Host Command (CRO):	0x23	CTR	0x01	0x02	0x03	0x04	0x05	0x06
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x00	0x00	0x33	0x80	0x03

Download 6 Command:*Data = 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C*

Host Command (CRO):	0x23	CTR	0x07	0x08	0x09	0x0A	0x0B	0x0C
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x00	0x00	0x33	0x80	0x06

Build Check Sum Command:*Size = 12 (previous 2 Download 6 commands), CRC = 0x5432*

Host Command (CRO):	0x0E	CTR	0x00	0x00	0x00	0x0C	0x54	0x32
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x00	0x54	0x32	--	--

Set MTA Command:*MTA# = 0x00, Address Ext = 0x00, Address = 0x003E9000*

Host Command (CRO):	0x02	CTR	0x00	0x00	0x00	0x3E	0x90	0x00
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

Download 6 Command:*Data = 0x01, 0x02, 0x03, 0x04, 0x05, 0x06*

Host Command (CRO):	0x22	CTR	0x01	0x02	0x03	0x04	0x05	0x06
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x00	0x00	0x3E	0x90	0x03

Download Command:*Byte Count = 2, Data = 0x01, 0x02*

Host Command (CRO):	0x03	CTR	0x02	0x01	0x02	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x00	0x00	0x3E	0x90	0x04

Build Check Sum Command:*Size = 8 (previous Download 6 & Download commands), CRC = 0x6833*

Host Command (CRO):	0x0E	CTR	0x00	0x00	0x00	0x08	0x68	0x33
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x00	0x68	0x33	--	--

Build Image Check Sum Command:

Size = 0, CRC = 0x3579

Host Command (CRO):	0x0E	CTR	0x00	0x00	0x00	0x00	0x35	0x79
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x00	0x35	0x79	--	--

7. Disconnect from the DSP causing it to jump to the application:Disconnect Command:

EndOfSession = 0x01, DSP Station Address = 0x0001

Host Command (CRO):	0x07	CTR	0x01	--	0x01	0x00	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

4.2 Updating Serial EEPROMs

The process for reprogramming both the Bootloader and Data EEPROM devices is the same. Upgrading the EEPROM involves connecting to the DSP, setting up the Bootloader's configuration registers, erasing the Serial EEPROM, and finally programming the Serial EEPROM. This example is for the Bootloader EEPROM.

1. Host Connects to DSP Bootloader Application:Connect Command:

DSP Station Address = 0x0001

Host Command (CRO):	0x01	CTR	0x01	0x00	--	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

2. Host needs to write the password for the operation by writing to the configuration register. First, set the pointer (MTA) to the correct register address:Set MTA Command:

MTA# = 0x00, Address Ext = 0x01(Config Reg Space), Address = 0x00000000 (Password addr)

Host Command (CRO):	0x02	CTR	0x00	0x01	0x00	0x00	0x00	0x00
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

3. Then write the password to the configuration register:Unlock Command:

Data size(bytes) = 2, PASSWORD = 0xABCD

Host Command (CRO):	0x13	CTR	0x02	0xCD	0xAB	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x01	0x00	0x00	0x00	0x01

4. Set MTA to Serial EEPROM Space for Clear Command:

Set MTA Command:

MTA# = 0, Address Ext = 2 (Bootloader EEPROM)

Host Command (CRO):	0x02	CTR	0x00	0x02	--	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

5. Erase the Serial EEPROM by giving the Clear Memory:

Clear Memory Command:

EEPROM Sector Selection Mask = 0x00000000 (don't care)

Host Command (CRO):	0x10	CTR	0x00	0x00	0x00	0x00	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

6. Program the Serial EEPROM by repeatedly issuing Set MTA commands to set the Address followed by Download 6 Bytes or Download commands until all desired memory locations are programmed. Since the MTA is auto-incremented by the DSP it only needs to be set by the Host when the data is not sequential (i.e. at the start of each S-Record). The Build Check Sum command should be issued to actually load the data into EEPROM (i.e. after each S-Record has been downloaded).

Set MTA Command:

MTA# = 0, Address Ext = 0x02, Address = 0x00000000

Host Command (CRO):	0x02	CTR	0x00	0x02	0x00	0x00	0x00	0x00
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

Download 6 Command:

Data = 0x01, 0x02, 0x03, 0x04, 0x05, 0x06

Host Command (CRO):	0x23	CTR	0x01	0x02	0x03	0x04	0x05	0x06
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x02	0x00	0x00	0x00	0x06

Download 6 Command:

Data = 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C

Host Command (CRO):	0x23	CTR	0x07	0x08	0x09	0x0A	0x0B	0x0C
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x02	0x00	0x00	0x00	0x0C

Build Check Sum Command:

Size = 12 (previous 2 Download 6 commands), CRC = 0x5432

Host Command (CRO):	0x0E	CTR	0x00	0x00	0x00	0x0C	0x54	0x32
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x02	0x54	0x32	--	--

Set MTA Command:*MTA# = 0, Address Ext = 0x02, Address = 0x00000200*

Host Command (CRO):	0x02	CTR	0x00	0x02	0x00	0x00	0x02	0x00
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

Download 6 Command:*Data = 0x01, 0x02, 0x03, 0x04, 0x05, 0x06*

Host Command (CRO):	0x23	CTR	0x01	0x02	0x03	0x04	0x05	0x06
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x02	0x02	0x00	0x00	0x06

Download Command:*Byte Count = 2, Data = 0x01, 0x02*

Host Command (CRO):	0x03	CTR	0x02	0x01	0x02	--	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x02	0x02	0x00	0x00	0x08

Build Check Sum Command:*Size = 8 (previous Download 6 & Download commands), CRC = 0x6833*

Host Command (CRO):	0x0E	CTR	0x00	0x00	0x00	0x08	0x68	0x33
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x02	0x68	0x33	--	--

Build Image Check Sum Command:*Size = 0, CRC = 0x2468*

Host Command (CRO):	0x0E	CTR	0x00	0x00	0x00	0x00	0x24	0x68
Bootloader Reply (DTO):	0xFF	STATUS	CTR	0x02	0x24	0x68	--	--

7. Disconnect from the DSP causing it to jump to (execute) the application:Disconnect Command:*EndOfSession = 0x01, DSP Station Address = 0x0001*

Host Command (CRO):	0x07	CTR	0x01	--	0x01	0x00	--	--
Bootloader Reply (DTO):	0xFF	STATUS	CTR	--	--	--	--	--

5. Bootloader and Application Interaction

The Bootloader provides two points of interaction with the running Flash application through two sets of RAM based variables. These variables are at “known” memory locations. Use of this interaction mechanism is voluntary on the part of the Flash application.

5.1 Bootloader Version Number

The first interaction mechanism is a way in which the Bootloader can inform the Flash application which version of Bootloader is resident in the serial EEPROM. A reserved memory section of two 16-bit variables reside at physical addresses 0x07F3 and 0x07F4. Keep in mind, that memory as addressed, accesses 16-bit quantities not 8-bit. The 16-bit quantity at location 0x07F3 is the **MAJOR** version of the Bootloader and the 16-bit quantity at location 0x07F4 is the **MINOR** version of the Bootloader. These values are placed there by the Bootloader just prior to loading and executing the Flash application. If the Flash application has no need for this information, those memory locations are free to be re-used. They are in fact, free to be re-used even if the Flash application wants the Bootloader version numbers. In the latter case, the application must simply extract the values prior to re-using the locations.

5.2 Forcing a Modified Reboot

This second point of interaction is a means by which the application can affect a reset of the DSP and instruct the Bootloader to modify its boot procedure slightly. The Bootloader is simple in its logic at bootstrap. After some preliminary system initialization, it waits two seconds for the receipt of a valid CCP message. If a message is received, it then assumes communication with a CCP v2.1 aware host application for download instructions. If after two seconds, no message is received, then the Bootloader assumes that there is no host available and that the Flash application should be executed.

However, sometimes this two second window of time may not be sufficient for the host to initiate communication with the Bootloader. For example, the Flash application in the normal course of its communications with the host, may decide that it wants an upgraded version of either the Bootloader or itself to be downloaded. The trick is then to re-invoke the Bootloader and ensure that it waits long enough for the host to gather its resources to effect communication with the host. If the mechanism described below is used, the initial wait by the Bootloader is extended from two seconds to 16 seconds for this single reboot.

A reserved memory section of three 16-bit variables reside at the physical address location 0x007F0. The 16-bit quantities at locations 0x007F0 and 0x007F1 are simply numbers, any numbers the application wishes. Then, the 16-bit value at location 0x007F2 is the **CRC** of those two previous numbers. Just prior to the Bootloader looking for a CCP message from the host, it checks these two values and their **CRC**. If the **CRC** is valid, then the normal two second wait is extended to 16 seconds for this single boot. To affect this, the application writes two random numbers into the first two locations, calculates a **CRC** for them and puts this in the third location, then allows the *Watchdog* timer to expire to affect a DSP reset. That's all there is to this procedure. If the Flash application has no need for this, those memory locations are free to be re-used. They are in fact, free to be re-used at any time except prior to a reboot procedure.

The following code snippet shows an example of setting the reboot key/CRC and then tripping the watchdog.

5.3 Code to Trip the Watchdog

The following code places two unimportant values in the first two locations and then calculates a valid CRC for them, placing that in the third location. Then the *Watchdog* counter register is set to a very small number and then enabled. The endless loop at the end of the code is just waiting for the hard reset to occur.

```
//-----
// Variables

Uint16 *pBootKeyVal1 = ( Uint16 * ) 0x07F0;
Uint16 *pBootKeyVal2 = ( Uint16 * ) 0x07F1;
Uint16 *pBootKeyCrc  = ( Uint16 * ) 0x07F2;
Uint16 tmpCrc = 0xFFFF;

//-----
// Define any 'bootkey sequence' with a valid CRC ...

*pBootKeyVal1 = 0x1234;
*pBootKeyVal2 = 0x5678;
b1Crc16Run ( ( char * ) pBootKeyVal1 >> 8, &tmpCrc );
b1Crc16Run ( ( char * ) pBootKeyVal1 & 0x00FF, &tmpCrc );
b1Crc16Run ( ( char * ) pBootKeyVal2 >> 8, &tmpCrc );
b1Crc16Run ( ( char * ) pBootKeyVal2 & 0xFF, &tmpCrc );
*pBootKeyCrc = tmpCrc;

//-----
// Fire-up the WATCHDOG and let it trip ...

EALLOW;                               // standard MACROS and
SysCtrlRegs.WDCNTR = 0x0010;          // register definitions that
SysCtrlRegs.WDCR   = 0x0028;          // are provided with TI's
EDIS;                                   // Code Composer

//-----
// Now just wait for a reboot ...

while ( 1 );
```

6. CCP Over Modbus

By its very definition, the CCP protocol is meant to be used on a CAN-based system. However, many of Oztek's standard controller's also provide serial RS-232/422/485 interfaces. For end user systems which use a serial interface to the controller and do not implement a CAN interface, Oztek has developed a serial Bootloader application that implements the CCP CRO/DTO messaging protocol as described in the previous sections of this manual. The serial communications protocol uses the Modbus protocol to access a set of 16-bit registers that are used to emulate the CRO command message from the host and the DTO response message from the Bootloader.

A Modbus reference guide can be downloaded at: http://modbus.org/docs/PI_MBUS_300.pdf.

6.1 Serial Bus Configuration

The serial Bootloader defaults to using the following serial port properties:

- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None
- Baud Rate: 19200 bits per second
- Modbus Slave Address 2

The Baud Rate may be temporarily changed to a different rate as described earlier in section 3.9 (Action Service Request Command).

The Modbus Slave Address defaults to a value of 2, as this is the default address for most of Oztek's Modbus-based controllers. The user has the option of changing the default Modbus Slave Address for their target controller by changing a data byte stored in the 4th-to-last memory location in the Bootloader EEPROM. For a 128kbit EEPROM device (which is typical on most Oztek controllers), this would be EEPROM address 0x3FFE. At power-up, the serial Bootloader reads the value stored at this location. If it is a valid and unique Modbus slave address (1 to 247), then the Bootloader will use this as the slave address. Otherwise, if the value is 0 or 248 and above, the Bootloader will assume the user has not set this value and will default to a slave address of 2.

It is up to the user's control application to provide a means for setting or changing this memory location within the Bootloader EEPROM. Care must be taken to not change any of the other EEPROM memory locations as this could lead to corrupting the Bootloader application.

6.2 Modbus Register Map

As previously mentioned, the CRO and DTO message objects described in the CCP protocol have been mapped to multiple 16-bit Modbus registers in the serial Bootloader application. The register map, including the most significant and least significant byte mappings to the actual CRO/DTO messages is as follows:

Table 19 - Modbus Register Map

Modbus Address	Register Name:	MSB Data (Bits 15:8)	LSB Data (Bits 7:0)
0x0001	DTO_WRD0	Return Code	Packet ID
0x0002	DTO_WRD1	Data Byte 0	CNT
0x0003	DTO_WRD2	Data Byte 2	Data Byte 1
0x0004	DTO_WRD3	Data Byte 4	Data Byte 3
0x0005	CRO_WRD0	CNT	Command
0x0006	CRO_WRD1	Data Byte 1	Data Byte 0
0x0007	CRO_WRD2	Data Byte 3	Data Byte 2
0x0008	CRO_WRD3	Data Byte 5	Data Byte 4
0x0009	CRO_WRD4	Data Byte 7	Data Byte 6
0x000A	CRO_WRD5	Data Byte 9	Data Byte 8
...
0x0081	CRO_WRD124	Data Byte 247	Data Byte 246
0x0082	CRO_WRD125	Data Byte 249	Data Byte 248
0x0083	CRO_WRD126	Data Byte 251	Data Byte 250

6.3 Sending CRO Commands

As the register map above shows, there are 4 consecutive 16-bit Modbus registers (addresses 0x0005 – 0x0008) that are used to define the standard 8-byte CRO command message. With the exception of the DOWNLOAD and DOWNLOAD_6 commands, the remaining commands described in the previous sections of this document are sent to the Bootloader simply by writing to the first four CRO command registers in order, starting with word 0 (CRO_WRD0, address 0x0005), and ending with word 3 (CRO_WRD3, address 0x0008). For these standard commands, upon writing to the CRO_WRD3 register, the Bootloader will then use the contents of all four registers to build the standard 8-byte CRO command message. This is then passed on to the CCP command parser that looks for a valid CCP message and acts accordingly. This is the same CCP command parser that resides in Oztek’s standard CAN-based Bootloader applications.

The CRO registers are all implemented as write-only registers. Attempts to read from these register addresses will result in an invalid error response in the Modbus response.

In order to fully utilize the serial bus throughput, to maintain coherency between all CRO message bytes (i.e. send the data all at once), and to utilize the Modbus transfer CRC error handling, the host controller should implement the writes to the CRO command registers as a single Modbus transfer using the “**Preset Multiple Registers**” function code (16 decimal, 0x10 hex) rather than using the “**Preset Single Register**” function code (06) to write to each 16-bit register individually. Using function code 16, a typical CRO command byte sequence as sent across Modbus to the Bootloader would consist of the following:

Table 20 - Modbus CRO Command Byte Sequence

Byte Order	Byte Name	Value
0	Slave Addr	2 (default)
1	Function	0x10
2	Start Addr Hi	0x00
3	Start Addr Lo	0x05
4	Reg Count Hi	0x00
5	Reg Count Lo	0x04
6	Reg 5 Data Hi	CRO CNT
7	Reg 5 Data Lo	CRO CMD
8	Reg 6 Data Hi	CRO Byte 1
9	Reg 6 Data Lo	CRO Byte 0
10	Reg 7 Data Hi	CRO Byte 3
11	Reg 7 Data Lo	CRO Byte 2
12	Reg 8 Data Hi	CRO Byte 5
13	Reg 8 Data Lo	CRO Byte 4
14	CRC	--

6.3.1 Downloading Complete S-Records

Unlike CAN messages, whose payloads are limited to a maximum of 8 bytes, Modbus messages can transfer up to 255 bytes of data. Given this, with the serial Bootloader interface, it is not necessary to segment an S-Record into multiple DOWNLOAD_6 or DOWNLOAD commands. Instead, a complete S-Record can be transferred in one Modbus transfer, which dramatically cuts down on the serial transfer time when downloading a complete image. For this reason, a special command has been created specifically for the Oztek serial Bootloader called PROGRAM_SRECORD. This custom command uses CRO command code 0x30, which is unused by the CCP protocol. This command uses the format shown in the following table:

Table 21 - Custom “Program S-Record” CCP Command Format

<i>Byte</i>	<i>Host Command (CRO)</i>	<i>Bootloader Reply (DTO)</i>
0	0x30	0xFF
1	CNT	Return Code
2	SIZE	CNT
3	ADDR_EXT	ADDR EXT
4	ADDR [31:24]	CRC [15:8]
5	ADDR [23:16]	CRC [7:0]
6	ADDR [15:8]	0x00
7	ADDR [7:0]	0x00
8	CRC [15:8]	
9	CRC [7:0]	
10	BYTE 0	
11	BYTE 1	
12	BYTE 2	
...	...	
252	BYTE 242	
253	BYTE 243	

This custom command combines the functions of the SET_MTA, DOWNLOAD/DOWNLOAD_6, and BUILD_CHECKSUM – the CRO command provides the memory address and extension for where the S-Record data should be loaded, a count of how many bytes are being loaded, and a host-calculated checksum for the complete contents of the S-Record for the Bootloader to check once all S-Record data has been read from the command.

The SIZE field in the CRO command indicates the number of S-Record data bytes being downloaded. As the table above shows, the maximum number of data bytes (and hence max value for the SIZE field) that can be transmitted in this command is 244 bytes. This is slightly less than the theoretical maximum S-Record data size (255 bytes minus the address and checksum bytes in the S-Record); however most file converters that generate S-Record data generally create much smaller S-Records, so this size restriction should not cause compatibility issues.

Once all of the data bytes in this CRO command have been received by the Bootloader, it will then compare its locally calculated CRC against the value provided by the host. If they match, the Bootloader will then attempt to program the data into the target memory. Upon success, the Bootloader will send a DTO response with a “no error” return code, and the selected memory address extension in byte 3 and the locally calculated CRC in bytes 4 and 5. Otherwise, any error in the receipt of this command or in the attempt to program the data will result in a DTO error response.

Because the serial Bootloader uses this new PROGRAM_SRECORD command for downloading data, it does not implement or support the DOWNLOAD and DOWNLOAD_6 commands. Also, because the S-Record CRC checking is performed by this command, the BUILD_CHECKSUM command is only used for checking image-wide CRCs at the end of downloading all data (i.e. SIZE=0 in the BUILD_CHECKSUM command). The per-S-Record BUILD_CHECKSUM (i.e. SIZE is not zero) is not supported by the serial Bootloader.

When sending a PROGRAM_SRECORD command, the additional CRO registers are used (CRO_WRD3, address 0x0009 through CRO_WRD126, address 0x0083) as necessary. Again, all CRO registers should be transferred at once using the “**Preset Multiple Registers**” Modbus function code as described above.

6.4 Reading DTO Responses

Unlike the CAN Bootloader, which automatically sends the DTO response message after the CRO command is received, the serial Modbus-based Bootloader does not automatically send the DTO response back to the user because it is just a slave on the Modbus interface. Instead, the host must read the DTO response directly using the DTO response Modbus registers at addresses 0x0001 through 0x0004. When reading these registers, the host should read all four registers at once using the “**Read Holding Registers**” function code (03). Using function code 03, the byte sequence as sent across Modbus to the Bootloader to read the DTO response would consist of the following:

Table 22 - Modbus DTO Read Request Byte Sequence

Byte Order	Byte Name	Value
0	Slave Addr	2 (default)
1	Function	0x03
2	Start Addr Hi	0x00
3	Start Addr Lo	0x01
4	Reg Count Hi	0x00
5	Reg Count Lo	0x04
6	CRC	--

The Modbus response from the serial Bootloader would then be as follows:

Table 23 - Modbus DTO Read Response Byte Sequence

Byte Order	Byte Name	Value
0	Slave Addr	2 (default)
1	Function	0x03
2	Byte Count	0x08
3	Reg 1 Data Hi	Return Code
4	Reg 1 Data Lo	PID = 0xFF
5	Reg 2 Data Hi	DTO Byte 0
6	Reg 2 Data Lo	DTO CNT
7	Reg 3 Data Hi	DTO Byte 2
8	Reg 3 Data Lo	DTO Byte 1
9	Reg 4 Data Hi	DTO Byte 4
10	Reg 4 Data Lo	DTO Byte 3
11	CRC	--

After sending a CRO command to the Bootloader, the host should poll the DTO response registers. If the Bootloader is busy performing the requested operation, the Return Code in the DTO response will be set to 0x10 to indicate that the processor is busy. In this case, the host should continue polling the DTO response registers until it gets a non-busy return code (0x00 for no error, or 0x30-0x36 for errors).

Warranty and Product Information

Limited Warranty

What does this warranty cover and how long does it last? This Limited Warranty is provided by Oztek Corp. ("Oztek") and covers defects in workmanship and materials in your OZSCR1000 controller. This Warranty Period lasts for 18 months from the date of purchase at the point of sale to you, the original end user customer, unless otherwise agreed in writing. You will be required to demonstrate proof of purchase to make warranty claims. This Limited Warranty is transferable to subsequent owners but only for the unexpired portion of the Warranty Period. Subsequent owners also require original proof of purchase as described in "What proof of purchase is required?"

What will Oztek do? During the Warranty Period Oztek will, at its option, repair the product (if economically feasible) or replace the defective product free of charge, provided that you notify Oztek of the product defect within the Warranty Period, and provided that through inspection Oztek establishes the existence of such a defect and that it is covered by this Limited Warranty.

Oztek will, at its option, use new and/or reconditioned parts in performing warranty repair and building replacement products. Oztek reserves the right to use parts or products of original or improved design in the repair or replacement. If Oztek repairs or replaces a product, its warranty continues for the remaining portion of the original Warranty Period or 90 days from the date of the return shipment to the customer, whichever is greater. All replaced products and all parts removed from repaired products become the property of Oztek.

Oztek covers both parts and labor necessary to repair the product, and return shipment to the customer via an Oztek-selected non-expedited surface freight within the contiguous United States and Canada. Alaska, Hawaii and locations outside of the United States and Canada are excluded. Contact Oztek Customer Service for details on freight policy for return shipments from excluded areas.

How do you get service? If your product requires troubleshooting or warranty service, contact your merchant. If you are unable to contact your merchant, or the merchant is unable to provide service, contact Oztek directly at:

USA
Telephone: 603-546-0090
Fax: 603-386-6366
Email techsupport@oztekcorp.com

Direct returns may be performed according to the Oztek Return Material Authorization Policy described in your product manual.

What proof of purchase is required? In any warranty claim, dated proof of purchase must accompany the product and the product must not have been disassembled or modified without prior written authorization by Oztek. Proof of purchase may be in any one of the following forms:

- The dated purchase receipt from the original purchase of the product at point of sale to the end user
- The dated dealer invoice or purchase receipt showing original equipment manufacturer (OEM) status
- The dated invoice or purchase receipt showing the product exchanged under warranty

What does this warranty not cover? Claims are limited to repair and replacement, or if in Oztek's discretion that is not possible, reimbursement up to the purchase price paid for the product. Oztek will be liable to you only for direct damages suffered by you and only up to a maximum amount equal to the purchase price of the product. This Limited Warranty does not warrant uninterrupted or error-free operation of the product or cover normal wear and tear of the product or costs related to the removal, installation, or troubleshooting of the customer's electrical systems. This warranty does not apply to and Oztek will not be responsible for any defect in or damage to:

- a) The product if it has been misused, neglected, improperly installed, physically damaged or altered, either internally or externally, or damaged from improper use or use in an unsuitable environment
- b) The product if it has been subjected to fire, water, generalized corrosion, biological infestations, or input voltage that creates operating conditions beyond the maximum or minimum limits listed in the Oztek product specifications including high input voltage from generators and lightning strikes
- c) The product if repairs have been done to it other than by Oztek or its authorized service centers (hereafter "ASCs")
- d) The product if it is used as a component part of a product expressly warranted by another manufacturer
- e) The product if its original identification (trade-mark, serial number) markings have been defaced, altered, or removed
- f) The product if it is located outside of the country where it was purchased
- g) Any consequential losses that are attributable to the product losing power whether by product malfunction, installation error or misuse.

Disclaimer

Product

THIS LIMITED WARRANTY IS THE SOLE AND EXCLUSIVE WARRANTY PROVIDED BY OZTEK IN CONNECTION WITH YOUR OZTEK PRODUCT AND IS, WHERE PERMITTED BY LAW, IN LIEU OF ALL OTHER WARRANTIES, CONDITIONS, GUARANTEES, REPRESENTATIONS, OBLIGATIONS AND LIABILITIES, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE IN CONNECTION WITH THE PRODUCT, HOWEVER ARISING (WHETHER BY CONTRACT, TORT, NEGLIGENCE, PRINCIPLES OF MANUFACTURER'S LIABILITY, OPERATION OF LAW, CONDUCT, STATEMENT OR OTHERWISE), INCLUDING WITHOUT RESTRICTION ANY IMPLIED WARRANTY OR CONDITION OF QUALITY, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE TO THE EXTENT REQUIRED UNDER APPLICABLE LAW TO APPLY TO THE PRODUCT SHALL BE LIMITED IN DURATION TO THE PERIOD STIPULATED UNDER THIS LIMITED WARRANTY. IN NO EVENT WILL OZTEK BE LIABLE FOR: (a) ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, LOST REVENUES, FAILURE TO REALIZE EXPECTED SAVINGS, OR OTHER COMMERCIAL OR ECONOMIC LOSSES OF ANY KIND, EVEN IF OZTEK HAS BEEN ADVISED, OR HAD REASON TO KNOW, OF THE POSSIBILITY OF SUCH DAMAGE, (b) ANY LIABILITY ARISING IN TORT, WHETHER OR NOT ARISING OUT OF OZTEK'S NEGLIGENCE, AND ALL LOSSES OR DAMAGES TO ANY PROPERTY OR FOR ANY PERSONAL INJURY OR ECONOMIC LOSS OR DAMAGE CAUSED BY THE CONNECTION OF A PRODUCT TO ANY OTHER DEVICE OR SYSTEM, AND (c) ANY DAMAGE OR INJURY ARISING FROM OR AS A RESULT OF MISUSE OR ABUSE, OR THE INCORRECT INSTALLATION, INTEGRATION OR OPERATION OF THE PRODUCT. IF YOU ARE A CONSUMER (RATHER THAN A PURCHASER OF THE PRODUCT IN THE COURSE OF A BUSINESS) AND PURCHASED THE PRODUCT IN A MEMBER STATE OF THE EUROPEAN UNION, THIS LIMITED WARRANTY SHALL BE SUBJECT TO YOUR STATUTORY RIGHTS AS A CONSUMER UNDER THE EUROPEAN UNION PRODUCT WARRANTY DIRECTIVE 1999/44/EC AND AS SUCH DIRECTIVE HAS BEEN IMPLEMENTED IN THE EUROPEAN UNION MEMBER STATE WHERE YOU PURCHASED THE PRODUCT. FURTHER, WHILE THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, YOU MAY HAVE OTHER RIGHTS WHICH MAY VARY FROM EU MEMBER STATE TO EU MEMBER STATE OR, IF YOU DID NOT PURCHASE THE PRODUCT IN AN EU MEMBER STATE, IN THE COUNTRY YOU PURCHASED THE PRODUCT WHICH MAY VARY FROM COUNTRY TO COUNTRY AND JURISDICTION TO JURISDICTION.

Return Material Authorization Policy

Before returning a product directly to Oztek you must obtain a Return Material Authorization (RMA) number and the correct factory "Ship To" address. Products must also be shipped prepaid. Product shipments will be refused and returned at your expense if they are unauthorized, returned without an RMA number clearly marked on the outside of the shipping box, if they are shipped collect, or if they are shipped to the wrong location.

When you contact Oztek to obtain service, please have your instruction manual ready for reference and be prepared to supply:

- The serial number of your product
- Information about the installation and use of the unit
- Information about the failure and/or reason for the return
- A copy of your dated proof of purchase

Return Procedure

Package the unit safely, preferably using the original box and packing materials. Please ensure that your product is shipped fully insured in the original packaging or equivalent. This warranty will not apply where the product is damaged due to improper packaging. Include the following:

- The RMA number supplied by Oztek clearly marked on the outside of the box.
- A return address where the unit can be shipped. Post office boxes are not acceptable.
- A contact telephone number where you can be reached during work hours.
- A brief description of the problem.

Ship the unit prepaid to the address provided by your Oztek customer service representative.

If you are returning a product from outside of the USA or Canada - In addition to the above, you **MUST** include return freight funds and you are fully responsible for all documents, duties, tariffs, and deposits.

Out of Warranty Service

If the warranty period for your product has expired, if the unit was damaged by misuse or incorrect installation, if other conditions of the warranty have not been met, or if no dated proof of purchase is available, your unit may be serviced or replaced for a flat fee. If a unit cannot be serviced due to damage beyond salvation or because the repair is not economically feasible, a labor fee may still be incurred for the time spent making this determination.

To return your product for out of warranty service, contact Oztek Customer Service for a Return Material Authorization (RMA) number and follow the other steps outlined in "Return Procedure".

Payment options such as credit card or money order will be explained by the Customer Service Representative. In cases where the minimum flat fee does not apply, as with incomplete units or units with excessive damage, an additional fee will be charged. If applicable, you will be contacted by Customer Service once your unit has been received.