Bootstrap Mode User's Guide

Applies to the following dataTaker models:

- **DT5xx (Series 3)**

UM-0077-A0

Head Office

Australia – Melbourne

DataTaker Pty Ltd
7 Seismic Court
Rowville Victoria 3178
Tel: 03 9764 8600 +613 9764 8600
Fax: 03 9764 8997 +613 9764 8997
Email: sales@dataTaker.com.au

Offices

United Kingdom

DataTaker Ltd
Shepreth
Cambridgeshire
SG8 6GB
Tel: +44 (0) 1763 264780
Fax: +44 (0) 1763 262410
Emaill: sales@dataTaker.co.uk

United States of America

DataTaker Inc,
22961 Triton Way Suite E
Laguna Hills CA 92653-1230
Tel: 1-800-9-LOGGER
Tel: 949 452 0750 +1 949 452 0750
Fax: 949 452 1170 +1 949 452 1170
Email: sales@dataTaker.com

Worldwide Dealer Network

www.dataTaker.com

Table of Contents

List of Tables

1 Introduction

The bootstrap mode for dataTaker 5xx Series 3 data loggers allows the flash memory within the logger to be modified. The primary reason to do this is to allow for initial factory programming of the firmware and later user upgrade of the firmware.

The bootstrap mode also has many other utility functions for inspecting and modifying the internal SRAM memory and PCMCIA card memory as well as the flash memory.

2 Memory types

Bootstrap mode allows you to inspect and modify different memory devices within the dataTaker. The devices that can be inspected/modified are the flash memory (FLASH), internal SRAM memory (RAM) and the PCMCIA SRAM card memory (CARD). Internally the memory devices are accessed in a banked manner. The bootstrap mode presents all of the memory devices as linear memory devices so that the banked access requirements are not visible.

You can also inspect and modify anything in the physical address space (PHYS). Accessing memory in this way does not modify the current banking configuration. Therefore the actual memory accessed at any location depends on the banking hardware configuration at the time. The banking configuration is controlled by hardware registers accessed as memory mapped I/O – see *A1 Physical Memory Maps* on page 18 for a description of the physical address memory assignment.

3 Flash memory organization

The flash memory device has a total size of 512Kbytes which is organized as 8 by 64KByte sectors. To change a memory location it must be erased first. To erase any location the whole sector that the location belongs to must be erased. The flash device's memory is divided into several areas as follows:

Address (Hex)	Size	Description
7FFFF ↕ 70000	64KBytes	Spare (1 sector)
6FFFF ↕ 60000	64KBytes	User program (1 sector) (first 4KBytes only)
5FFFF ↕ 40000 (DT_START)	128KBytes (DT_SIZE)	dataTaker firmware (2 sectors)
3FFFF ↕ 10000	192KBytes	Spare (3 sectors)
0FFFF ↕ 00000 (BS_START)	64KBytes (BS_SIZE)	Bootstrap loader (1 sector)

Table 1 - Flash memory organization

4 Getting into bootstrap mode

The dataTaker will normally run the standard dataTaker firmware after a power-up or reset. To get into bootstrap mode the user must use one of the following methods:

1. If the standard dataTaker firmware is running you can send the special command ^ZBOOTSTRAP to the dataTaker. (This can be sent as \026BOOTSTRAP from within DeTransfer command window). The baud rate will stay at whatever rate was set via the DIP switches.
2. Install a jumper on the connector J7 (located on the bottom board near the lithium battery) and reset or power-up the dataTaker. You will be required to partially dismantle the case to access the jumper. This method is the only method available if the flash memory that holds the dataTaker firmware has been corrupted, erased or not programmed. The baud rate will be forced to 9600 baud.
3. Install jumpers on the connectors J7 & J8 (located on the bottom board near the lithium battery) and insert a pre-programmed SRAM PCMCIA card with it's write protect switch set to on. See *Creating a PCMCIA card to boot from* on page 14. You will be required to partially dismantle the case to access the jumpers. This is the only method available if the bootstrap loader in flash memory has been corrupted, erased or not programmed. The baud rate will be forced to 9600 baud when starting that dataTaker in this way.

On entering or starting in bootstrap mode the dataTaker will respond on the serial port as follows:

```
Bootstrap mode V1.00 (FLASH)
Flash Manuf ID = 01, Device ID = A4
```

If starting bootstrap mode from a PCMCIA card then the dataTaker will respond as follows:

```
Bootstrap mode V1.00 (CARD)
Flash Manuf ID = 01, Device ID = A4
```

5 Bootstrap commands

A {300|1200|2400|4800|9600|19200}	set baud rate
B{C|F|R} <start addr>,<length>	blank check memory (Flash, Card, Ram)
C{C|F|R}{C|F|R} <src addr>,<dest addr>,<length>	copy from one memory to another
E{C|F|R} <start addr>,<length>	erase memory (Flash, Card, Ram)
E{B|D}	erase section (Bootstrap, Datataker)
H[A|B|C|E|I|J|K|M|O|R|S|V|W|U]	display help for specific command
I <addr>	read input port on CPU
J	jump to logger application
K{C|F|R} <start addr>,<length>	return checksum for area of memory
M{C|F|R}	set/get memory type (Flash, Card, Ram)
O <addr>,<data>	write data to output port on CPU
R{C|F|R|P} [<start addr>][,<length>]	read memory (Flash, Card, Ram, Phys)
S{0-9}...	write S record to current memory
V{C|F|R}{C|F|R} <src addr>,<dest addr>,<length>	compare one memory to another
W{C|F|R|P} <start addr>,<data>[,<data>...]	write memory (Flash, Card, Ram, Phys)
U{B|D}	upgrade Bootstrap or dataTaker code

Table 2 - Bootstrap mode command summary

Notes:

1. All address, data and length fields are expected in hexadecimal.
2. All letters can be upper or lower case.
3. The following words can be used to substitute for specific addresses:
   DT_START - start address of the dataTaker application
   BS_START - start address of the bootstrap application
4. The following words can be used to substitute for specific lengths:
   DT_SIZE - the size of the dataTaker application
   BS_SIZE - the size of the bootstrap application

"A" command - Set baud rate

Command Format: A {300|1200|2400|4800|9600|19200}

Description: Set a new baud rate for serial port communications

"B" command - Blank check memory

Command Format: B{C|F|R} <start addr>,<length>

Description: Blank check <length> bytes starting from <start addr> in the specified memory type. The {C|F|R} letter specifies the memory type to check. If the memory type is not specified then it defaults to the current memory type. The start address field can be specified as DT_START or BS_START or as a specific address in hexadecimal. The length field can be specified as DT_SIZE or BS_SIZE or as a specific length in hexadecimal.

"C" command - Copy memory

Command Format: C{C|F|R}{C|F|R} <src addr>,<dest addr>,<length>

Description: Copy from <src addr> in specified memory to <dest addr> in specified memory. The first {C|F|R} letter specifies the source memory type, the second {C|F|R} letter specifies the destination memory type. If source memory type is not specified then it defaults to CARD. If the destination memory type is not specified then it defaults to FLASH. Address fields can be specified as DT_START or BS_START or as a specific address in hexadecimal. The length field can be specified as DT_SIZE or BS_SIZE or as a specific size in hexadecimal.

"E" command – Erase memory

Command formats: E{C|F|R} <start addr>,<length>
E{B|D}

Description: Erase <length> bytes starting from <start addr> in the specified memory type. The {C|F|R} letter specifies the memory type to erase. If the memory type is not specified then it defaults to the current memory type. The start address field can be specified as DT_START or BS_START or as a specific address in hexadecimal. The length field can be specified as DT_SIZE or BS_SIZE or as a specific length in hexadecimal.

The {B|D} letter specifies to erase the Bootstrap or dataTaker code in flash memory without having to specify a start address or size.

Note that for FLASH memory the minimum erase size is a complete 64Kbyte flash sector. This erase function will erase any sectors that are partially or wholly within the erase range.

Note that the sector containing the bootstrap loader (the first sector) cannot be erased unless the bootstrap mode is running from a PCMCIA card.

"I" command – Read I/O port

Command format: I <addr>

Description: Read a CPU I/O port at address <addr> and return its current value. Valid I/O port addresses are 0x00 to 0xFF. Consult Z180 data book for function of each port.

"J" command – Jump to dataTaker application

Command format: J

Description: Jump to (and execute) the dataTaker logger application. Note that this will only work if the bootstrap program is running from FLASH.

"K" command – Checksum memory

Command format: K{C|F|R} <start addr>,<length>

Description: Return check sum of <length> bytes starting from <start addr> in the specified memory type. The {C|F|R} letter specifies the memory type to check. If the memory type is not specified then it defaults to the current memory type. The start address field can be specified as DT_START or BS_START or as a specific address in hexadecimal. The length field can be specified as DT_SIZE or BS_SIZE or as a specific length in hexadecimal.

"M" command – Get/Set current memory type

Command Format: M[{C|F|R}]

Description: Set or get the current memory type. If the {C|F|R} letter is specified it sets the current memory type, otherwise if it is not specified then the current memory type is returned.

"O" command – Write I/O port

Command format: O <addr>,<data>

Description: Write 8 bit <data> to a CPU I/O port at address <addr>. Valid I/O port addresses are 0x00 to 0xFF. Consult Z180 data book for function of each port.

"R" command – Read memory

Command format: R{C|F|R|P} [<start addr>][,<length>]

Description: Read <length> bytes from <start addr> in the specified memory type. The {C|F|R|P} letter specifies the memory type to read. If the memory type is not specified then it defaults to the current memory type. If the start address is not specified then it defaults to zero. If the length is not specified then it defaults to 128. The start address fields can be specified as DT_START or BS_START or as a specific address in hexadecimal. The length field can be specified as DT_SIZE or BS_SIZE or as a specific address in hexadecimal. You can simply specify 'R' to continue displaying memory from last output.

"S" command – Write S record to memory

Command format: S{0-9}...

Description: Write a Motorola S record to current memory. S0 records are decoded to determine the name of the file being downloaded and to reset the record count. S5 records are decoded to determine if the record count matches. S1-3 records are decoded and written to current memory. S4,6-9 records are ignored.
Note that the sector containing the bootstrap loader (the first sector) cannot be written to unless the bootstrap mode is running from a PCMCIA card.

"V" command – Compare memory

Command format: V{C|F|R}{C|F|R} <src addr>,<dest addr>,<length>

Description: Compare (verify) <length> bytes in the specified memory starting at <src addr> against <dest addr> in the other specified memory. The first {C|F|R} letter specifies the source memory type, the second {C|F|R} letter specifies the destination memory type. If the source memory type is not specified then it defaults to CARD. If the destination memory type is not specified then it defaults to FLASH. Address fields can be specified as DT_START or BS_START or as a specific address in hexadecimal. The length field can be specified as DT_SIZE or BS_SIZE or as a specific size in hexadecimal.

"W" command – Write memory

Command Format: W{C|F|R|P} <start addr>,<data>[,<data>...]

Description: Write <data> to <start addr> in the specified memory type. The {C|F|R|P} letter specifies the memory type to write. If the memory type is not specified then it defaults to the current memory type. If more than one data fields are specified they will be written to consecutive locations. The <data> field(s) can be either 1,2 or 4 bytes. The start address fields can be specified as DT_START or BS_START or as a specific address in hexadecimal.

Note that the sector containing the bootstrap loader (the first sector) cannot be written to unless the bootstrap mode is running from a PCMCIA card.

"U" command – Upgrade firmware

Command format: U{B|D}

Description: Upgrade Bootstrap or dataTaker code from card to flash memory. This function will erase the relevant sectors and then copy the code from the card to the flash memory.

6 Updating dataTaker firmware via serial port

The dataTaker firmware is distributed as a Motorola S-Record image of the firmware. This format includes all address and data information to be able to write the correct data to the flash memory directly.

The bootstrap mode of the dataTaker will accept Motorola S-Records directly. Therefore you simply need to send the file to the dataTaker. You do not need and should not use any file transfer protocols. The Motorola S-Record format includes checksums and a record count that ensures the data is transmitted correctly.

You should ensure that flow control has been setup properly as this is required to ensure that the file is transferred correctly. For the DT5xx Series 3 data loggers this must be software (XON/XOFF) flow control as they do not support hardware flow control.

If you are using 9600 baud then the update of the dataTaker firmware will take approximately 5 minutes.

IMPORTANT NOTE: If the update procedure is interrupted or abandoned then you may be forced to use the second option of getting into bootstrap mode, using a Jumper, when retrying this procedure as the first option of using the ^ZBOOTSTRAP command only works if there is a working version of the dataTaker firmware in flash memory.

To update the dataTaker firmware held in the flash memory of your DT5xx Series 3 data logger you will need to do the following:

1. Obtain the firmware image file (e.g. dt5xxS3.sre). The latest version of dataTaker firmware is available on our web site (www.datataker.com). There is also a recent version of the firmware on our software suite CD that is shipped with your dataTaker.
2. Put the dataTaker being updated into bootstrap mode (see *4 Getting into bootstrap mode* on page 5)
3. Using a terminal program, such as DeTransfer[1] or HyperTerminal[2], issue the following command to the dataTaker to erase the dataTaker firmware:
   ```
   ED
   ```
   the dataTaker will respond as follows if the erase is successful
   ```
   Erasing Flash Memory
   ..
   Erase successful
   ```
4. Send the firmware image file (as an ASCII text file) to the dataTaker. E.g. In DeTransfer open the file as a command file and use the send all command to send the entire file to the dataTaker. Or in HyperTerminal use the “send text file...” command to send the image file.

[1] DeTransfer is copyright dataTaker and is supplied on the dataTaker software suite CD as a host application.
[2] HyperTerminal is copyright Hilgraeve Inc. and comes bundled with most Microsoft Windows operating systems

5. Wait for approximately 5 minutes (if transmitting at 9600 baud). During this time the dataTaker will respond as follows if the download is successful:

```
Writing S record file "dt5xxS3.sre" to FLASH
.....................................................
.....................................................
.....................................................
.....................................
File write successful (00001D00h records)
```

6. Confirm that the file has been transmitted correctly. If the file was not successfully written then check the following:
    a. that your communications cable is firmly attached and in good condition.
    b. if sending the file over a remote connection using modems then ensure that an error correction protocol is enabled.
    c. that the image file you are downloading is not corrupt – if obtained from our web site, try downloading it again.
    d. that flow control is enabled correctly.
7. Issue the “J” command or reset the dataTaker to start the new firmware.

Note: It has been found on some slower PC's that using DeTransfer to transmit the firmware image file does not always work. In these cases HyperTerminal was successful.

7 Placing dataTaker commands and programs into flash memory

You can place dataTaker commands/programs into the flash memory in a location reserved for them. The dataTaker will run the commands whenever it resets or powers up.

The dataTaker commands must not exceed 4Kbytes. A whole 64K sector is reserved for the dataTaker commands, however only the first 4KBytes can be used. This allows you to erase the program without affecting anything else stored in the flash memory device.

To place commands into flash memory you will need to do the following:

1. Save the commands into a text file.
2. Run the ROMCVT[1] program to convert the text file to a binary image file suitable to program into flash. See *A2 ROMCVT utility* on page 20 for specific information on running the ROMCVT utility. For example if your user program is in a file called “program.txt” then the following command will create a binary image of the program called program.bin:
   ```
   romcvt program.txt program.bin
   ```
3. Convert the binary image to Motorola S-Records using the OBJCOPY[2] utility. Ensure that the starting address of the S-Records is 60000h. For example
   ```
   objcopy --change-addresses=0x60000 -I binary -O srec program.bin program.sre
   ```
4. Fix up the Motorola S record file using the SRECFIX[3] utility program so that it is ready for downloading to the dataTaker. For example
   ```
   srecfix program.sre
   ```
5. Put the dataTaker being updated into bootstrap mode (see *4 Getting into bootstrap mode* on page 5)
6. Using a terminal program, such as DeTransfer[4] or HyperTerminal[5], issue the following command to the dataTaker to erase the sector that contains any dataTaker commands to run on reset or power up:
   ```
   EF 60000,1
   ```
   the dataTaker will respond with the following if the erase is successful
   ```
   Erasing Flash Memory
   .
   Erase successful
   ```

[1] ROMCVT is copyright dataTaker and is supplied on the dataTaker software suite CD as a utility program
[2] OBJCOPY is a GNU utility for converting between various object formats.
[3] SRECFIX is copyright dataTaker and is supplied on the dataTaker software suite CD as a utility program
[4] DeTransfer is copyright dataTaker and is supplied on the dataTaker software suite CD as a host application.
[5] HyperTerminal is copyright Hilgraeve Inc. and comes bundled with most Microsoft Windows operating systems.

7. Send the S-Record file that contains the commands (as an ASCII text file) to the dataTaker. E.g. In DeTransfer open the file as a command file and use the send all command to send the entire file to the dataTaker. Or in HyperTerminal use the “send text file...” command to send the firmware image file.
8. Confirm that the file has been transmitted correctly. The dataTaker will respond as follows if the file has been received correctly:

```
Writing S record file "program.sre" to FLASH
........
File write successful (00000073h records)
```

9. Reset the dataTaker and check that the commands have been executed.

8 Booting from an SRAM PCMCIA card

Booting from an SRAM PCMCIA card allows you to perform some more advanced functions such as updating the bootstrap loader in the flash memory.

Creating a PCMCIA card to boot from

If you wish to start a dataTaker from a PCMCIA card you must create a card with a copy of the bootstrap loader on it. The bootstrap loader can be copied from the flash memory of any dataTaker that has already been programmed or a new version of the bootstrap loader may be downloaded directly to the card. Note that any data stored on the card will be lost when the bootstrap loader is copied to the card.

The PCMCIA card must be an SRAM card with a write protect switch on it. The write protect switch is required as the dataTaker will to write to the card during power-up when booting from the card. This write is not intentional, it is a small glitch in the power-up procedure. If the card is not write protected then the stored code will be corrupted. This glitch only occurs if booting from the card, it does not happen if the card is used for data storage during normal operation.

The procedure to copy the bootstrap loader from flash to a card is as follows:

1. Put the dataTaker into bootstrap mode (see *4 Getting into bootstrap mode* on page 5)
2. Insert an SRAM PCMCIA card into the dataTaker. Ensure that the write-protect switch is OFF.
3. Using a terminal program, such as DeTransfer or HyperTerminal, issue the following command to the dataTaker to copy the bootstrap loader from the flash memory to the card:
   ```
   CFC BS_START,BS_START,BS_SIZE
   ```
   the dataTaker will respond with the following if the copy is successful
   ```
   Copying from FLASH to CARD
   ..........................................................
   Copy successful
   ```
4. If you wish to use the card to update the dataTaker firmware then you can copy the dataTaker firmware to the card as well. To do this, issue the command:
   ```
   CFC DT_START,DT_START,DT_SIZE
   ```
   the dataTaker will respond with the following if the copy is successful
   ```
   Copying from FLASH to CARD
   ..........................................................
   Copy successful
   ```
5. Set the write protect switch on the PCMCIA card to ON.
6. The card is now ready to be used to boot from.

The procedure to download a new bootstrap loader via the serial port to a card is as follows:

1. Put the dataTaker into bootstrap mode (see *4 Getting into bootstrap mode* on page 5)

2. Insert an SRAM PCMCIA card into the dataTaker. Ensure that the write-protect switch is OFF.
3. Using a terminal program, such as DeTransfer or HyperTerminal, issue the following command to the dataTaker to set the current memory to the PCMCIA card:
   ```
   MC
   ```
   the dataTaker will respond with
   ```
   Current memory type is set to CARD
   ```
4. Send the bootstrap loader image file (as ASCII text file) to the dataTaker. E.g. In DeTransfer open the file as a command file and use the send all command to send the entire file to the dataTaker. Or in HyperTerminal use the “send text file...” command to send the image file.
5. Wait for approximately 1 minute (if transmitting at 9600 baud). During this time the dataTaker will respond with the following:
   ```
   Writing S record file "bootload.sre" to FLASH
   ..............................................
   File write successful (000005D8h records)
   ```
6. Confirm that the file has been transmitted correctly.
7. If you wish to use the card to update the dataTaker firmware then you can download a copy of the dataTaker firmware to the card as well. To do this send the dataTaker firmware image file to the dataTaker (in the same way that you sent the bootstrap loader image file)
8. Set the write protect switch on the PCMCIA card to ON.
9. The card is now ready to be used to boot from.

Updating the bootstrap loader

To update the bootstrap loader on a dataTaker you will require a specially prepared SRAM PCMCIA card that contains a copy of the bootstrap loader. See *Creating a PCMCIA card to boot from* on page 14 for instructions on how to create such a card.

Starting the dataTaker from a card allows you to change the bootstrap loader programmed into the flash memory. You can also program dataTaker firmware if that is contained on the card as well.

Use the following procedure to update the bootstrap loader in flash memory:

1. Insert the specially prepared SRAM PCMCIA card and ensure that it's write protect switch is set to on.
2. Install jumpers on J7 and J8.
3. Connect a serial port cable to the dataTaker and to a PC.
4. Start a terminal package, such as DeTransfer or HyperTerminal, and set the baud rate to 9600.
5. Power on the dataTaker to start the dataTaker from the card. The bootstrap sign on message will be returned as follows
   ```
   Bootstrap mode V1.00 (CARD)
   Flash Manuf ID = 01, Device ID = A4
   ```
6. Issue the following command to the dataTaker to update the bootstrap loader
   ```
   UB
   ```
   the dataTaker will respond with the following if successful
   ```
   Updating Bootstrap in Flash from Card
   Erasing Flash Memory
   .
   Erase successful
   Copying from Card to Flash Memory
   ................
   Update successful
   ```
7. Issue the following command to the dataTaker to update the dataTaker firmware
   ```
   UD
   ```
   The dataTaker will respond with the following if successful
   ```
   Updating Datataker application in Flash from Card
   Erasing Flash Memory
   ..
   Erase successful
   Copying from Card to Flash Memory
   ................
   Update successful
   ```

Factory programming of the flash memory

When manufactured the dataTaker flash memory is completely erased. Therefore the only way to start the dataTaker is to boot from a PCMCIA card that contains a bootstrap loader image on it.

If the dataTaker is booted from a PCMCIA card and it finds that the flash memory bootstrap loader section is erased then it will automatically program the flash memory for the bootstrap loader and the dataTaker firmware with the images stored on the card. This allows for simple factory programming by simply inserting a specially prepared card, installing jumpers on J7 and J8 and then powering on the dataTaker.

Note that the actual test that determines if the boot sector is erased just checks the first 100h bytes of the boot sector to see if they are erased.

The dataTaker will respond on the serial port as follows if successful programming of the flash memory occurs:

```
Bootstrap mode V1.00 (CARD)
Flash Manuf ID = 01, Device ID = A4
*** Commencing factory program of flash memory from card ***
Updating Bootstrap in Flash from Card
Erasing Flash Memory
.
Erase successful
Copying from Card to Flash Memory
.................
Update successful
Updating Datataker application in Flash from Card
Erasing Flash Memory
..
Erase successful
Copying from Card to Flash Memory
..............................
Update successful
*** SUCCESSFUL factory program of flash memory ***
```

If the programming is not successful then the errors will be reported at the stage that failed and the last line will be returned as

```
*** FAILED factory program of flash memory ***
```

A1 Physical Memory Maps

The physical memory is organized in two different ways depending upon whether the logger was booted from flash memory of a PCMCIA card. The following tables show the different organizations:

Address (hex)	Size	Description
FFFFF ↕ 80000	512KBytes	Internal RAM – variables/stack/data store
7FFFF ↕ 50000	192KBytes	Memory mapped I/O – hardware registers
4FFFF ↕ 40000	64KBytes	PCMCIA card access window
3FFFF ↕ 00000	256KBytes	Flash code space – code and constants

Table 3 - DT5xx Series 3 physical address space when booting from flash

Address (hex)	Size	Description
FFFFF ↕ 80000	512KBytes	Internal RAM – variables/stack/data store
7FFFF ↕ 40000	256KBytes	Flash code space – code and constants
3FFFF ↕ 10000	192KBytes	Memory mapped I/O – hardware registers
0FFFF ↕ 00000	64KBytes	PCMCIA card access window

Table 4 - DT5xx Series 3 physical address space when booting from PCMCIA card

The memory mapped I/O address space is divided up as shown in the following table:

Address (hex) Normal/Card	Size	Description
78000/38000	32KBytes	82C54 – RTC and ADC X/Y counters
70000/30000	32KBytes	82C54 – 3 x High Speed Counters
68000/28000	32Kbytes	Buffer (Read Only) GPBuf1
60000/20000	32Kbytes	Latch (Write Only) GPReg3
58000/18000	32Kbytes	Latch (Write Only) GPReg2
50000/10000	32Kbytes	Latch (Write Only) GPReg1

Table 5 – DT5xx Series 3 memory mapped I/O address space

The general purpose registers and buffers are 8 bit registers. The following tables show the usage of the bits within the registers:

General Purpose Register 1 (GPReg1)

7	6	5	4	3	2	1	0
DEBNCE	AD_START	IN_STB	OUT_STB	CA19	CA18	CA17	CA16

General Purpose Register 2 (GPReg2)

7	6	5	4	3	2	1	0
CRST	CREG	CA21	CA20	HSC1_A1	HSC1_A0	-	CMERR_CL

General Purpose Register 3 (GPReg3)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	BOOT_PGE

General Purpose Buffer 1 (GPBuf1)

7	6	5	4	3	2	1	0
CARD_IN	-	-	CM_ERR	CNTR_B3	CNTR_B2	CNTR_B1	CNTR_B0

A2 ROMCVT utility

Introduction

This utility program, ROMCVT.EXE, converts dataTaker command files to binary memory image files for writing into a dataTaker EPROM or flash memory. It is supplied on the dataTaker software suite CD and is also available from our web site (www.datataker.com).

Converting Command files

The command line syntax for the converter is

ROMCVT [cmdfile.ext] [outfile.ext]

If outfile.ext is not given on the command line then you will be prompted for it. If neither file is given then both will be prompted for.

The command file being converted must not contain any deTransfer (or determinal) commands. If any are found in the command file then an error is reported and the conversion is aborted. Similarly there must be no card commands in the command file (i.e. a ';' specified at the start of a line). If any card commands are found then an error is reported and the conversion is aborted. Any comments found in the command file will be ignored and not placed in the output file. If the resulting output file is larger than 4091 characters long then an error message will be returned and the conversion will be aborted.

Creating a pre-programmed dataTaker ROM

For DT5xx Series 1 or Series 2:

The standard dataTaker ROM (should be version 3.34 or higher) needs to be loaded into the memory of the EPROM programmer. Locations 1F000 to 1FFFD are available to hold the user program. Locations 1FFFE and 1FFFF should both be zero. The first three locations, 1F000-1FFF2 hold the special signature and size of the user program. These are normally set to FF if no program is set.

Version 3.34 or higher of the dataTaker firmware should be used. The ROMable program feature was introduced in version 3.10 but had a bug that only allowed up to 256 characters of program to be specified. This was fixed in version 3.34.

For DT5xx Series 3 :

All versions of the firmware support this function. Locations 60000 to 60FFD are available to hold the user program. Locations 60FFE and 60FFF should both be zero. The first three locations, 60000-60FF2 hold the special signature and size of the user program. These are normally set to FF if no program is set. The output file created by the converter should be programmed into the FLASH memory at address 60000.