DATA TECHNOLOGY CORPORATION

DTC-10 HOST ADAPTER*

FOR THE IEEE 696.1 (S-100) BUS

SPECIFICATION

January 14, 1981

*THE DTC-10 SUPPORTS STANDARD ADDRESSING, i.e.,
8 Bit Data paths (Din and Dout)
8 Bit I/O Address (256 I/O ports)

WARRANTY DISCLAIMER:

ANY MODIFICATION OR ALTERATION TO THIS BOARD AUTOMATICALLY NULLIFIES ANY WARRANTY OFFERED BY DTC OR ITS DISTRIBUTORS.

TABLE OF CONTENTS

1.0 INTRODUCTION

2.0 DISK SUBSYSTEM

- 2.1 Theory of Operation
- 2.2 Electrical Interface

3.0 DTC-10 HOST ADAPTER

- 3.1 Interface Register Definition
- 3.2 Software Theory of Operation
- 3.3 Hardware Theory of Operation

4.0 ELECTRICAL/MECHANICAL SPECIFICATIONS

5.0 INSTALLATION

- 5.1 Inspection
- 5.2 Preparation for Use
- 5.3 Initial Checkout

6.0 REFERENCE DOCUMENTATION

- 6.1 DTC Supplied Documentation
- 6.2 Other Documentation

APPENDIX A - COMMANDS/PROGRAMMING

APPENDIX B - HOST BUS PIN ASSIGNMENT

APPENDIX C - S100 PIN ASSIGNMENT

APPENDIX D - SAMPLE 8080 PROGRAM

1.0 INTRODUCTION

The DTC-10 Host Adapter is a single board interface card for the IEEE 696.1 S-100 Bus. This Host Adpater may be utilized with any of the Data Technology Corporation fixed disk controllers (DTC-1KA series). This specification provides the programming mechanism and command block format utilized by the DTC-10 Host Adapter. The detailed specifications for the DTC controllers can be found in the respective controller documentation.

The DTC-10 Host Adapter fits into a single S-100 Bus slot and presents one unit load to the bus.

Commands are issued to the controller through the Host Adapter in the host computer. The controller accepts data from the Host Adapter and transfers the data to the correct location on the disk. In addition, the controller will detect/correct burst errors from the fixed disk drive (4 bits in length) before data is transferred to the computer.

2.0 DISK SUBSYSTEM

The DTC-10 Host Adapter and the DTC-1KA controller comprise one part of the disk storage subsystem. Each of the DTC controllers complies with the interface requirements for the particular disk drive so that installation is fairly simple. In addition, the DTC-10 Host Adapter will operate with any DTC controller/formatter with the standard DTC host interface. A list of available DTC-1KA controllers and their respective disk drives follows.

(Since new, and sometimes plug-compatible drives are consistently being introduced this list is only representative)

CONTROLLER	DISK AND CAPACITY
DTC 510	Seagate Technogy ST506 or equivalent (Olivetti, RMS, and Tandon Magnetics) 1 or 2, ST506 drives - 3 or 6 MBytes each
SA1410	Shugart Associates M600
DTC520	4 drives, ST506 type and 96 TPI mini-floppy (MPI, Shugart Associates etc.)
SA1420	Shugart Associates M600 with 96 TPI; mini-floppy
SA1401	2 Shugart Associates SA1000 (5 or 10 MBytes)
SA1403D	SA1000 with integral IBM compatible single/double density 8-inch flexible disk drive backup

SA1404	Shugart Associates SA4000 (14 to 58 MBytes)
SA1404D	SA4000 with SA800/850 integral IBM compatible single/double density flexible disk drive backup
SA1406	SA1000 with Data Electronics Streaker streaming tape backup (10 to 20 MBytes)
SA1407	SA4000 with DEI Streaker backup
DTC101	Memorex 101 (11 to 22 Mbytes) Fujitsu 2301/2 (11 to 22 Mbytes)
DTC101D	Memorex 101, Fujitsu 2301/2 and integral IBM single/double density backup
DTC900	SA1000 or Q2000 with Data Peripherals DP100 (10 Mbytes) 8 inch cartridge backup

2.1 THEORY OF OPERATION

Disk commands are issued to the DTC controller via commands stored in the main memory (the command structure is described in section 4.0 of each of the DTC controller specifications). Depending on the type of command, the controller will request up to 10 command bytes. Upon reception of the last command byte, the controller begins execution of the command.

For the data transfer commands, a check is performed on the disk address, and status is flagged if it exceeds the drive limits. The data is stored in a sector buffer on the controller before it is transferred to the host or disk drive. This buffer eliminates any possibility of data overruns between the host and the disk.

Upon completion of the command, the controller will output the completion status to the data register in the Host Adapter. (Further delineation of the completion status may be requested by issuing the appropriate sense commands.)

2.2 ELECTRICAL INTERFACE

The electrical interface to the Winchester disk drives and flexible drives will conform to the requirements described in the fixed disk and flexible disk interface specifications.

The electrical interface to the S-100 Bus will conform to the particular requirements of the S-100 backplane.

3.0 DTC-10 HOST ADAPTER

The DTC-10 Host Adapter is designed to operate in most S-100 systems based upon IEEE.

3.1 INTERFACE REGISTER DEFINITION

The interface registers for the DTC-10 Host Adapter are listed below. Where b represents the 6 most significant bits of the I/O address.

HEX Address	Register	
b00	Data in/out Register	DAR
b01	Control Register (write only)	CNR
b02	Status Register (read only)	BSTAT

The bit definition for each register is described in the following sections.

3.1.1 BIT DEFINITION FOR EACH REGISTER

CONTROL REGISTER (CNR) OUTPUT ADDRESS MN1

Bit 7	Not used
Bit 6	Assert select and Data bit 0 used to access a controller
Bit 5	Not used
Bit 4	Not used
Bit 3	Not used
Bit 2	Not used
Bit 1	Enable data, after the selection process
Bit 0	Not used

3.1.2 BUS STATUS - processor can read status of host bus

BUS STATUS (BSTAT) INPUT ADDRESS MN3

Bit 7	REQ - indicates the controller either requests data or has data for the host adapter.
Bit 6	IN/OUT* (reference to controller) - low indicates data to host adapter, high indicates data to controller.
Bit 5	MSG - indicates last byte in data or command string.
Bit 4	COM/DTA* - a command to the controller will have a high, data will be low.
Bit 3	BUSY - indicates the status of the busy signal. High means controller is busy.
Bit 2	Not used (low)
Bit 1	Not used (high)
Bit 0	Not used (high)

3.2 SOFTWARE THEORY OF OPERATION

The method by which a command is executed is as follows:

1 - Device driver builds a Command Descriptor Block (CDB) in system memory (see section 4.0 of the appropriate DTC controller specification).

2 - The driver then writes the address of the first byte of the CDB into the Command I/O Pointer Block (CIOPB) of the command driver routine.

3 - The DATA ADDRESS (DAD) is also set up if a data transfer is required. Commands requiring data transfers are READ, WRITE, READ ID, REQUEST SENSE, REQUEST SYNDROME, and WRITE ECC.

4 - The driver now performs a GETCON routine which determines if the controller is busy. When it is not busy, the GETCON routine will assert the SELECT line until the controller responds with a BUSY.

5 - When the controller responds to the host adapter by asserting BUSY, the driver shifts to the OUTCON routine. In response to the REQuest bit in the BSTAT, the driver passes the command a byte at a time to the controller.

6 - The controller verifies that the command is correct and begins the command execution phase. At this time the data is transferred to the host adapter and into the S-100 memory.

7 - After the data transfer is completed, the controller enters the command completion phase. The controller sends a one-byte status to the host adapter indicating whether or not an error occurred during command execution. This is handled by the CMPSTAT routine. Finally, the controller sends the message byte (of zeroes), and the operation is complete.

8 - At this time the controller enters the idle (non BUSY) mode awaiting another command. If an error was encountered by the controller, the CMSTAT routine will return with it in the C register. It is the responsibility of the device driver to issue a REQUEST SENSE command to request any detailed information about the error.

3.3 HARDWARE THEORY OF OPERATION

The Host Adapter serves as a data channel for the controller. Commands and data are fetched/stored to the system memory as a function of REQ. The Host Adapter consists of Command and Status Registers, a DMA channel, and an interrupt latch. The registers are addressed as I/O ports. Commands and data are passed through these registers as a function of the I/O driver routine and the controller status lines. The host adapter will return an ACK after each DATA or COMMAND cycle has been completed.

Each memory cycle is initiated when the controller asserts REQ. The driver will respond by reading/writing the data register.

When data is transferred to the host adapter, the data on the host bus is held until the memory write is completed. When data is transferred to the controller, the data is latched into a holding register, then sent to the controller.

3.3.1 I/O LOGIC OPERATION (Bus Slave)

The Host Adapter responds to commands from the CPU processor to either read a particular register or write to a register. The address selection switches are set with the dipswitch at location 5D. The dipswitch selects a block of four I/O addresses. A read is selected when lines DBIN, pR/W*, and sINP are asserted with the appropriate I/O address. A write is performed when sOUT is high and pR/W* is low along with the I/O address. Because low power Schottky logic is used, the I/O logic will perform at the highest speed clocks now currently in use.

4.0 ELECTRICAL/MECHANICAL SPECIFICATIONS

HOST ADAPTER PHYSICAL PARAMETERS

(The DTC-10 Host Adapter fits into a single S-100 slot).

Width	10.0	inches
Length	5.125	inches
Height	0.75	inch
Weight	0.7	lbs.

ENVIRONMENTAL PARAMETERS

	Operating:	Storage:
Temperature (degrees F/C)	32/0 to 131/55	-40/-10 to 167/75
Relative Humidity (@ 40 degrees F, wet bulb temp, no condensation)	10% to 95%	10% to 95%
Altitude	sea level to 10K feet	sea level to 15K feet

POWER REQUIREMENTS

Voltage @ current(host adapter)	+8 VDC @ 1.5A(max)

Note: For the physical parameters of the controller, refer to its DTC controller specification.

5.0 INSTALLATION

5.1 INSPECTION

Inspect all shipping containers for damage. If a container is damaged, the contents should be checked and the Host Adapter verified electrically. If the Host Adapter is damaged, call Data Technology Corporation, Customer Service, for Material Return Authorization number. Please retain all shipping labels and documentation.

5.2 PREPARATION FOR USE

Before the Host Adapter can be used, initial setup may be required. Be sure the power requirements for the Host Adapter are met (section 4.0). The Host Adapter is installed in a vacant slot in the S-100 backplane.

A 50-pin, mass-terminated cable connects the Host Adapter to location J6 on the DTC controller board (pin 1 is marked on the Host Adapter connector (as a triagle or dot) and on the controller silkscreen). Refer to the interconnection diagram in the appropriate controller specification for connection of the controller to the disk drives. Note that all cables, including drive cables, are of the mass-terminated type, so no inadvertant signal swapping can occur.

Be sure the controller has adequate DC power (refer to the controller specification; the controller maintains the same power connector pinouts as the disk drive). To set up the controller, refer to the switch setting instructions found in the controller specification.

The following sections describe in detail the proper jumper settings on the Host Adapter.

5.2.1 ADDRESS SWITCHES

The address switch is located in position 5D

Note: If the switch is on, the logic compares for zero (0V to 0.8V) on the S-100 bus.
Bit assignment is as follows:

POSITION	ADDRESS
1	A7
2	A6
3	A5
4	A4
5	A3
6	A2
7	NC
8	NC

5.2.2 PARITY

The DTC-10 does not generate parity, so the parity option on the controller should be disabled.

5.3 INITIAL CHECKOUT

The initial verification of the disk subsystem can be done via an appropriate monitor PROM or through a debugging utility such as DDT under CP/M*.

First, verify that all the interface registers are accessible through the correct addresses and that the registers can be read/ written with the expected results. Install driver routines from reading the appendix A or from the DTC S-100 Driver BIOS Diskette. Next, attempt to issue a few commands to the disk subsystem again via the console.

A recommended approach is to first issue a RECALIBRATE command. After verifying that it executed correctly, issue a SEEK command to verify that the Logical Address calculation has been performed correctly. Then, issue a FORMAT DRIVE command; the recommended interleave for the S-100 system running at 2MHz is 4. Finally, the data transfer command should be issued to verify the data. All commands can be issued via the console programmer's interface.

6.0 REFERENCE DOCUMENTATION

This section provides information regarding the documentation available for using the DTC-10 Host Adapter.

6.1 DTC SUPPLIED DOCUMENTATION

6.1.1 DTC CONTROLLER SPECIFICAITONS

Each controller that is manufactured by DTC is described by its own specification. Refer to the appropriate controller document when attempting to program the disk subsystem.

6.1.2 DTC SOFTWARE MANUAL

This manual explains how to install CP/M onto your system using 1403D controller and the DTC-10. Also available is a DTCBIOS Diskette.

6.2 OTHER DOCUMENTATION

6.2.1 IEEE S-100

a. IEEE 696.1 Standard Specifications for S-100 Bus Interface Devices

b. S-100 CPU/ System Manual - use version appropriate for your system.

6.2.2 Disk Drive Documentation

Use the appropriate drive manufacturer's manual for your disk drive.

.PPENDIX A COMMANDS/PROGRAMMING

An I/O request to the DTC controller is performed by passing a command descriptor block (CDB) to the controller. The first byte of a CDB is the command class and opcode. The remaining bytes specify the drive logical unit number (LUN), block address, control bytes, and number of blocks to transfer. The controller performs an implied seek and verify when commanded to access a block.

Due to the different types of commands each controller recognizes, the command format for the Host Adapter will only indicate the skeletal representation of the command. The reader is directed to section 4.0 of the appropriate DTC controller specification for more detailed command information.

A.1 Command Format

A.1.1 Commands Requiring 6 Bytes

```
 ------------------------
| 7  6  5  4  3  2  1  0 |
|------------------------|
|    Command Byte 0      |   XXXX
|------------------------|
|    Command Byte 1      |   XXXX + 1
|------------------------|
|    Command Byte 2      |   XXXX + 2
|------------------------|
|    Command Byte 3      |   XXXX + 3
|------------------------|
|    Command Byte 4      |   XXXX + 4
|------------------------|
|    Command Byte 5      |   XXXX + 5
|------------------------|
| 7  6  5  4  3  2  1  0 |
 ------------------------
```

XXXX is the HEX address that is loaded into the CIOPB location

A.1.2 Commmands Requiring 10 Bytes

```
 -------------------------
| 7  6  5  4  3  2  1  0 |
|-------------------------|
|     Command Byte 0      |   XXXX
|-------------------------|
|     Command Byte 1      |   XXXX + 1
|-------------------------|
|     Command Byte 2      |   XXXX + 2
|-------------------------|
|     Command Byte 3      |   XXXX + 3
|-------------------------|
|     Command Byte 4      |   XXXX + 4
|-------------------------|
|     Command Byte 5      |   XXXX + 5
|-------------------------|
|     Command Byte 6      |   XXXX + 6
|-------------------------|
|     Command Byte 7      |   XXXX + 7
|-------------------------|
|     Command Byte 8      |   XXXX + 8
|-------------------------|
|     Command Byte 9      |   XXXX + 9
|-------------------------|
| 7  6  5  4  3  2  1  0 |
 -------------------------
```

XXXX is the HEX address that is loaded into the CIOPB location

A.2 Request Syndrome Command

The REQUEST SYNDROME Command returns 2 bytes of information. The data returned for the REQUEST SYNDROME Command is listed as follows:

```
 -------------------------
| 7  6  5  4  3  2  1  0 |
|-------------------------|
|       Data  Byte 0      |   XXXX
|-------------------------|
|       Data  Byte 1      |   XXXX + 1
 -------------------------
```

XXXX is the HEX address that is loaded into the DMA location

A.3 Drive and Controller Sense Information

Upon execution of the REQUEST SENSE command, the controller returns 4 bytes of information in the following format. Refer to Drive and Controller Sense in section 4.0 of the respective DTC controller specifications for a detailed interpretation of these bytes.

```
 -------------------------
| 7  6  5  4  3  2  1  0 |
|------------------------|
|      Data  Byte 0      |   XXXX
|------------------------|
|      Data  Byte 1      |   XXXX + 1
 -------------------------
|      Data  Byte 2      |   XXXX + 2
 -------------------------
|      Data  Byte 3      |   XXXX + 3
 -------------------------
```

XXXX is the HEX address that is loaded into the DMA location

Note: Data that is received from the controller as well as data that is sent to the controller will be transferred in the above order.

APPENDIX B HOST BUS PIN ASSIGNMENT

The host I/O bus uses a 50-pin connector (AMP 2-87227-5 or equivalent). The unused pins are spares for future use. The pin assigments are as follows:

Signal	Pin Number	
DATA0	2	
DATA1	4	
DATA2	6	
DATA3	8	
DATA4	10	
DATA5	12	
DATA6	14	
DATA7	16	
PARITY	18	
--	20	Future Usage
--	22	
--	24	
--	26	
--	28	
--	30	
--	32	
--	34	
BUSY	36	
ACK	38	
RST	40	
TDN	42	
SEL	44	
C/D	46	
REQ	48	
I/O	50	

Note: All signals are negative true and all odd pins are connected to ground. The signal lines are terminated with 220 ohms to 5V and 330 ohms to ground.

APPENDIX C IEEE S-100 SYSTEM BUS - Signal Definition

IEEE S-100 Bus
Component Side Pins

Pin	Signal	Description
1	+8 volts	Logic power - unregulated, max < 11.5v
2	+16 volts	Aux power - unregulated, max < 21.5v
3	XRDY	Act H, one of two bus ready signals
4	VI1*	Vectored interrupt line 0, active low, open collector; used with a vectored interrupt circuit to speed interrupt handling.
5	VI1*	See pin 4
6	VI2*	" " "
7	VI3*	" " "
8	VI4*	" " "
9	VI5*	" " "
10	VI6*	" " "
11	VI7*	" " "
12	NMI*	Non-maskable interrupt; active low, open collector.
13	PWRFAIL*	Power failure signal, active low
14	DMA3*	DMA request; active low, open collector
15	A18	Extended address bit 18
16	A16	Extended address bit 16
17	A17	Extended address bit 17
18	SDSB*	Disable the 8 status signals; active low, open collector
19	CDSB*	Disable the 5 control output signals; active low, open collector
20	GND	Extra ground
21	NDEF	Not defined (is used as optional DMA Request; open collector, low)
22	ADSB*	Disable the address lines (first 16); active low, open collector
23	DODSB*	Disable data output lines; active low, open collector
24	Phi Clk	Phase 1 master timing for the bus
25	PSTVAL	Status valid strobe; active low, at PSYNC time indicates that stable address and status are on the bus.
26	PHLDA	Hold acknowledge signal, active high
27	RFU	Reserved for future use
28	RFU	" " " "
29	A5	Address bit 5
30	A4	Address bit 4
31	A3	Address bit 3
32	A15	Address bit 15
33	A12	Address bit 12
34	A9	Address bit 9
35	DO1	Data out bit 1, bidirectional data 1
36	DO0	Data out bit 0, bidirectional data 0
37	A10	Address bit n10

38	DO4	Data out bit 4, bidirectional data 4
39	DO5	Data out bit 5, bidirectional data 5
40	DO6	Data out bit 6, bidirectional data 6
41	DI2	Data in bit 2, bidirectional data 10
42	DI3	Data in bit 3, bidirectional data 11
43	DI7	Data in bit 7, bidirectional data 15
44	SM1	Status indicating machine code fetch
45	SOUT	Status indicating I/O output cycle
46	SIMP	Status indicating I/O input cycle
47	SMEMR	Status indicating memory read - not an interrupt instruction fetch
48	SHLTA	Status indicating halt instruction is being acknowledged
49	CLOCK	A 2MHz clock - not required to be synchronous with other events
50	GND	Main ground

S-100 Circuit Side Pins

51	+8 volts	See pin 1
52	-16 volts	Negative aux power, unregulated, max <21.5v
53	GND	Extra ground
54	Slave CLR*	Resets bus slaves, is active with POC
55	DMA0*	DMA request line, active low, open collector
56	DMA1*	same as DMA0*
57	DMA2*	same as DMA0*
58	SXTRQ*	Status signal which requests that 16-bit slaves assert SIXTN*
59	A19	Extended address bit 19
60	SIXTN*	An active low signal asserted by 16-bit bus slaves in response to SXTRQ*
61	A20	Extended address bit 20
62	A21	Extended address bit 21
63	A22	Extended address bit 22
64	A23	Extended address bit 23
65	NDEF	Not defined (can be used as optional DMA Request, open collector, low true)
66	NDEF	same as above
67	PHANTOM*	
68	MWRT	Status indicated memory write
69	RFU	
70	GND	Extra ground
71	RFU	
72	RDY	Ready, indicates memory or I/O is ready; active high, open collector
73	INT*	The primary interrupt request signal is low true, open collector.
74	HOLD*	Request processor stop for DMA purposes; active low, open collector

75	RESET*	Master reset signal; active low, open collector
76	PSYNC	Control signal indicating beginning of new bus cycle
77	PR/W*	Read high, write low with data from CPU valid during low phase.
78	PDBIN	Control signal requesting input data
79	A0	Address bit 0
80	A1	Address bit 1
81	A2	Address bit 2
82	A6	Address bit 6
83	A7	Address bit 7
84	A8	Address bit 8
85	A13	Address bit 13
86	A14	Address bit 14
87	A11	Address bit 11
88	DO2	Data out bit 2, bidirectional data 2
89	DO3	Data out bit 3, bidirectional data 3
90	DO7	Data out bit 7, bidirectional data 7
91	DI4	Data in bit 4, bidirectional data 12
92	DI5	Data in bit 5, bidirectional data 13
93	DI6	Data in bit 6, bidirectional data 14
94	DI1	Data in bit 1, bidirectional data 9
95	DI0	Data in bit 0, bidirectional data 8
96	SINTA	Status indicating fetch of interrupt instruction
97	SWO*	Status indicating transfer of data from bus master to bus slave
98	ERROR*	Status indicating error condition during the present bus cycle
99	POC*	Power on clear, must remain low for 10ms
100	GND	Main ground

PPENDIX D   SAMPLE PROGRAM FOR DTC-10 HOST ADAPTER

The DTC-10 Host Adapter uses programmed I/O, taking advantage of the fact that the DTC controllers have a built-in sector buffer. The control lines of the host bus are available to the CPU through the Bus Status Register. Data and commmands are transmitted through the host bus by a simple handshake procedure as outlined in the DTC controller specifications. The types of commands available to the user are as follows:

STATUS Sends drive status to host adapter

TEST DRIVE READY
REQUEST SENSE
CHECK TRACK FORMAT
REQUEST SYNDROME

MOTION CONTROL Moves heads without R/W operation

SEEK
RECALIBRATE

R/W Read Write Operations

READ
WRITE
COPY

FORMAT Formats drive or tracks with specified standard format

FORMAT TRACK
FORMAT BAD TRACK
FORMAT DRIVE

DIAGNOSTICS Runs controller microdiagnostics

RAM DIAGNOSTIC
WRITE ECC
READ ID
DRIVE DIAGNOSTIC

Flow Diagrams

Status commands:

```
GET CONTROLLER
SEND COMMANDS to controller
READ STATUS DATA
COMPLETION STATUS
```

Motion Control:

```
GET CONTROLLER
SEND COMMANDS to controller
COMPLETION STATUS
```

Write Sector(s):

```
GET CONTROLLER
SEND COMMANDS
LOAD DATA
COMPLETION STATUS
```

Read Sector(s):

```
GET CONTROLLER
SEND COMMANDS
WAIT FOR REQ
READ DATA
COMPLETION STATUS
```

Copy:

```
GET CONTROLLER
SEND COMMANDS
COMPLETION STATUS
```

Diagnostics:

```
GET CONTROLLER
SEND COMMANDS
COMPLETION STATUS
```

ROGRAMMMING:

```
BASE equals Base I/O Address
DATAIN equals BASE
DATAOUT equals BASE
BCON equals BASE+1 ;Buss Control
BSTAT equals BASE+2 ; Bus Status
DMAOUT equals Base+3 ; DMA control bytes
DMAIN equals BASE+3  ; DMA status information
CIOPB ; Command Address
DMA   ; Data Address
```

Sample program to GET CONTROLLER:

```
GETCON: IN BSTAT                ;input from status port
        ANI 08H                 ;select bit 3 (busy)
        JNZ GETCON              ;if busy wait in getcon loop
        MVI A,40H               ;get ready to assert SEL and DATA0
        OUT BCON                ;to get attention of controller
CBUSY:  IN BSTAT                ;input from bus status
        ANI 08H                 ;again look at BUSY
        JZ CBUSY                ;we have controller attention else loop
        MVI A,02H               ;get ready to allow data enable
        OUT BCON                ;done
        RET                     ;return from get controller routine
```

Sample program to OUTPUT COMMANDS:

```
OUTCOM: LHLD CIOPB              ;load pointer to command queue
COMREQ: IN BSTAT                ;input from bus status
        MOV C,A                 ;store in C
        ORA A                   ;set flags
        JP COMREQ               ;wait for REQ
        ANI 10H                 ;check for command/ data
        RZ                      ;return when data is requested
        MOV A,C                 ;also see if controller switched direction
        ANI 40H
        RZ                      ;if it wants to send data, return
        MOV A,M                 ;move commands from queue to accumulator
        OUT DATAOUT             ;write comands to controller
        INX H                   ;increment pointer
        JMP COMREQ              ;loop as long as commands are requested
                                 from controller
```

ample program to SEND DATA TO CONTROLLER (a WRITE operation):

```
        LHLD DMA                ;load pointer to data
DAREQ:  IN BSTAT                ;input fron bus status
        MOV C,A                 ;store
        ANI 80H                 ;set flags
        JZ DAREQ                ;wait for REQ
        ANI 10H                 ;check for COM
        JNZ CMPSTAT             ;on receipt of command completion status is
                                 present
        MOV A,M                 ;move data into accumulator
        OUT DATAOUT             ;output to controller
        INX H                   ;increment pointer
        JMP DAREQ               ;go back for another byte
CMPSTAT:IN DATAIN               ;input completion status
        MOV C,A                 ;place in C for futher use
LREQ:   IN BSTAT                ;looking for last REQ
        MOV B,A                 ;save for checking
        ANI 80H                 ;check for REQ
        JZ LREQ                 ;loop untill found
        IN DATAIN               ;input last byte
        ORA A                   ;see if last byte is non-zero
        JNZ BADBYTE             ;if last byte is non zero
        MOV A,C                 ;now check completion status
        ORA A                   ;to see if it is zero
        JNZ BADSTAT             ;if not zero
        MOV A,B                 ;Now check last bus status
        ANI 01H                 ;for parity error
        JNZ BADPAR              ;high is bad parity
        XRA A                   ;zero accumulator
        RET                     ;GREAT! everything is OK
```

For information on how to decode errors generated, refer to the appropriate DTC controller specification.

Sample program to READ DATA FROM CONTROLLER:

```
READ:   LHLD DMA                ;load data pointer
RDREQ:  IN BSTAT                ;input bus status
        MOV C,A                 ;store for further checking
        ANI 80H                 ;look for REQ
        JZ RDREQ                ;else loop
        MOV A,C
        ANI 10H                 ;check for COM
        JNZ CMPSTAT             ;if COM present must be completion status
        IN DATAIN               ;input data from controller
        MOV M,A                 ;move data to pointer
        INX H                   ;increment pointer
        JMP RDREQ
```