# U S E R ' S  M A N U A L

# QUAD SERIAL RS232 M-MODULE

## MODEL M217

(FORMERLY HP E2261A)

Document Part No: 11029524

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, Texas 78681 <> www.chtech.com

## COPYRIGHT

C&H Technologies, Inc. (C&H) provides this manual "as is" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. C&H may make improvements and/or changes in the product(s) and/or program(s) described in this manual at any time and without notice.

This publication could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication.



A Reader's Comment Form is provided at the back of this publication. If this form has been removed address comments to:

C&H Technologies, Inc.
Technical Publications
445 Round Rock West Drive
Austin, Texas 78681-5012

Or visit our web site for support information at:

http://www.chtech.com.

## AMENDMENT NOTICE

C&H Technologies, Inc. makes every attempt to provide up-to-date manuals with the associated equipment. Occasionally, changes are made to the equipment wherein it is necessary to provide amendments to the manual. If any amendments are provided for this manual they are printed on colored paper and will be provided with the module and manual. Manual updates may also be found on out web site at www.chtech.com.

## NOTE

The contents of any amendment may affect operation, maintenance, or calibration of the equipment.

# INTRODUCTION

This manual describes the operation and use of the C&H Model M217 Quad Serial RS232 M-Module (Part Number 11029520). This module was formerly manufactured by HP (Agilent) as Model E2261A. C&H obtained the manufacturing rights from Agilent and now manufacturers it as C&H Model M217. This mezzanine module is designed to interface within any M/MA-Module carrier adhering to the ANSI/VITA 12-1996 M-Module specification. These carriers are available in many formats such as Ethernet, VME, VXI, PXI, cPCI, and the PC.

Contained within this manual are the physical and electrical specifications, installation and startup procedures, functional description, and configuration and programming guidelines to adequately use the product.

This manual is based on a low level register access, and is written in such a manner to provide understanding to the user based on this type of access. If a driver is provided, please refer to the driver documentation for instruction using the higher level interface provided by the driver.

TABLE OF CONTENTS

LIST OF FIGURES



LIST OF TABLES

## 1.0 GENERAL DESCRIPTION

The M217 provides four full-duplex (asynchronous) RS232 Data Terminal Equipment (DTE) ports on a single wide M-Module adhering to the ANSI/VITA 12-1996 specification for M-Modules. The M217 may be installed on any carrier board supporting the M-Module specification. Carriers are available that allow the M217 to be used in Ethernet, VXI, VME, PCI, cPCI and other system architectures.

### 1.1 PURPOSE OF EQUIPMENT

The M217 provides four full-duplex RS232 ports that can be used for communication with a variety of test instrumentation and equipment.

### 1.2 SPECIFICATIONS OF EQUIPMENT

#### 1.2.1 Key Features

- Four (4) full-duplex (asynchronous) RS232 Data Terminal Equipment (DTE) ports
- Independently programmable baud rate (13 fixed rates from 75 baud to 38.4K baud)
- Independently programmable handshake mode: RTS/CTS and/or DTR/DSR hardware handshake, XON/XOFF software handshake or NONE
- Independently programmable data format: 5 to 8 data bits plus parity; odd, even or no parity: 1, 1.5 or 2 stop bits
- Independent 2Kbyte Input and Output buffer for each port

### 1.2.2 Specifications

The M217 incorporates the standard 40-pin, 20x2 row connector interface to the carrier board for power and data/control, but does not have the 24-pin optional connector for carrying user-connections back onto the carrier board.

User input/output is provided through a standard 44-pin D-subminiature female receptacle (CONEC part number 302A10889X or equivalent). A mating connector kit can be ordered separately as AM111 (C&H Part Number 11029700-0001). The connector pin-outs are shown in Appendix A.

Table I. General Characteristics

| | |
|---|---|
| Drive Distance | 50 ft. |
| Isolation | None |

Table II. Power Requirements

| Input Supply | IPM (A) | IDM (A) |
|---|---|---|
| +5VDC | 0.10 | 0.01 |
| +12VDC | 0.01 | 0.005 |
| -12VDC | 0.01 | 0.005 |

### 1.2.3 Mechanical

The mechanical dimensions of the module are in conformance with ANSI/VITA 12-1996 for single-wide M-Module modules. The nominal dimensions are 5.687" (144.5 mm) long × 2.082" (52.9 mm) wide.

### 1.2.4 Bus Compliance

The module complies with the ANSI/VITA 12-1996 Specification for single-wide M-Modules and the MA-Module trigger signal extension. The module also supports the optional IDENT and VXI-IDENT functions.

| | |
|---|---|
| Module Type: | MA-Module |
| Addressing: | A08 |
| Data: | D16 |
| Interrupts: | supported |
| DMA: | not supported |
| Triggers: | not supported |
| Identification: | IDENT |
| Manufacturer ID: | $0FFF_{16}$ (See note below) |
| Model Number: | $067D_{16}$ |
| VXI Model Number: | $025A_{16}$ |

Note: C&H obtained the manufacturing rights from Hewlett Packard (Agilent) for this module. The ID's have been retained as Hewlett Packard to provide compatibility with existing SW drivers.

### 1.2.5 Applicable Documents

ANSI/VITA 12-1996 Standard for The Mezzanine Concept M-Module Specification, Approved May 20, 1997, American National Standards Institute and VMEbus International Trade Association, 7825 E. Gelding Dr. Suite 104, Scottsdale, AZ 85260-3415, http://www.vita.com

## 2.0 INSTALLATION

### 2.1 UNPACKING AND INSPECTION

Verify that there has been no damage to the shipping container. If damage exists then the container should be retained, as it will provide evidence of carrier caused problems. Such problems should be reported to the shipping courier immediately, as well as to C&H. If there is no damage to the shipping container, carefully remove the module from its box and anti static bag and inspect for any signs of physical damage. If damage exists, report immediately to C&H.

### 2.2 HANDLING PRECAUTIONS

The module contains components that are sensitive to electrostatic discharge. When handling the module for any reason, do so at a static-controlled workstation, whenever possible. At a minimum, avoid work areas that are potential static sources, such as carpeted areas. Avoid unnecessary contact with the components on the module.

### 2.3 INSTALLATION OF M/MA MODULES

All M-Modules must be installed into the carrier before the carrier is installed into the host system. To install a module, firmly press the connector on the M/MA-Module together with the connector on the carrier as shown in Figure 1. Secure the module through the holes in the bottom shield using the original screws.

**CAUTION: M/MA-Module connectors are NOT keyed. Use extra caution to avoid misalignment. Applying power to a misaligned module can damage the M/MA-Module and carrier.**

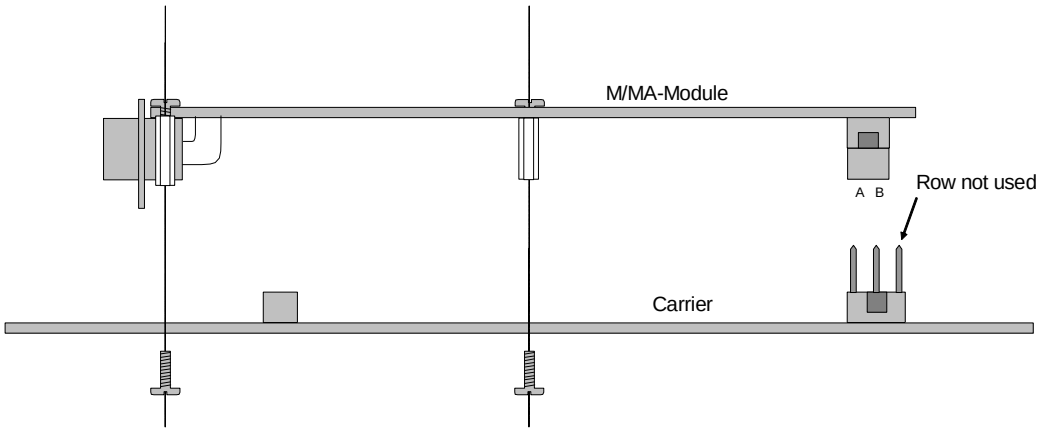



**Figure 1. M-MODULE Installation**

## 2.4 PREPARATION FOR RESHIPMENT

If the module is to be shipped separately it should be enclosed in a suitable water and vapor proof anti-static bag. Heat seal or tape the bag to insure a moisture-proof closure. When sealing the bag, keep trapped air volume to a minimum. The shipping container should be a rigid box of sufficient size and strength to protect the equipment from damage. If the module was received separately from a C&H system, then the original module shipping container and packing material may be re-used if it is still in good condition.

## 3.0 FUNCTIONAL DESCRIPTION

### 3.1 OVERVIEW

Before considering the individual registers on the M217, you need to understand how the module's receiver and transmitter operate. This section defines several terms used in this chapter. In this discussion, *receiver* is used to describe the M217 receiving data from an external source and sending it to the host computer (VXI system Controller); *transmitter* describes the M217 receiving data from the host computer and transmitting it to the external device. Figure 2 shows a simplified block diagram of the module.

At power-on or reset, the M217 sets the default baud rate and data format for each port; both transmitter and receiver for each port are disabled. The host computer must START/STOP specific port(s) to receive or transmit. The host computer can CLOSE a port without using the disable command. The difference between CLOSE and STOP is that CLOSE clears the port's buffer.

A simplified functional block diagram is shown in Figure 2.

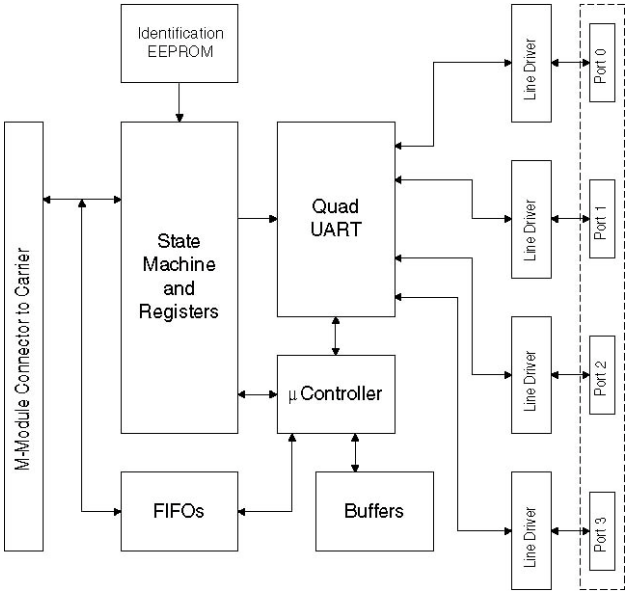



Figure 2. Functional Block Diagram

## 3.2 IDENTIFICATION AND CONFIGURATION REGISTERS

### 3.2.1 I/O Registers

There are a variety of registers used to configure and control the M217 module. These registers are located in the IOSpace. The address map of the registers is shown in Table III. Details of the registers are provided in Figure 3.

**Table III. I/O Address Map/Command Summary**

| IO REG. (HEX) | REGISTER DESCRIPTION | REGISTER TYPE |
|---|---|---|
| 00 | Status | Read Only |
| 02 | Control | Read/Write |
| 04 | Interrupt Vector | Read Only |
| 06 – 1E | Reserved | NA |
| 20 | Command Response | Read/Write |
| 22 | Parameter Register 0 | Read/Write |
| 24 | Parameter Register 1 | Read/Write |
| 26 | Command Status | Read Only |
| 28 | Interrupt Generator Serial Port 1 | Read/Write |
| 2A | Interrupt Generator Serial Port 2 | Read/Write |
| 2C | Interrupt Generator Serial Port 3 | Read/Write |
| 2E | Interrupt Generator Serial Port 4 | Read/Write |
| 30 – 34 | Reserved | NA |
| 36 | FIFO Status | Read Only |
| 38 | Interrupt Status/Control Serial Port 1 | Read/Write |
| 3A | Interrupt Status/Control Serial Port 2 | Read/Write |
| 3C | Interrupt Status/Control Serial Port 3 | Read/Write |
| 3E | Interrupt Status/Control Serial Port 4 | Read/Write |
| 40 | Transmit/Receiver Serial Port 1 | Read/Write |
| 42 | Transmit/Receiver Serial Port 2 | Read/Write |
| 44 | Transmit/Receiver Serial Port 3 | Read/Write |
| 46 | Transmit/Receiver Serial Port 4 | Read/Write |

Reg. 00

**Status**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Read-only | | | | | | | | | | | | | | | |
| Read | Not Used | | | | | | | | | | | IRQ3 | IRQ2 | IRQ1 | IRQ0 | CRDY |

CRDY ⇨ Command/Parameter Ready (1 = Command and Parameter registers are empty and available for the next command, 0 = indicates that the DONE bit in the Command Status Register (address $2E_{16}$) may not be valid.

IRQn ⇨ Interrupt Request Status Bit (1 = indicates the respective port (1 through 4) has an interrupt request pending (see note 2), 0 = interrupt request is not being asserted)

Notes:

1. At power-on or reset, all bits are "0" except CRDY which has a value of "1."
2. There are seven interrupt sources for each port. Refer to the Port Interrupt Status/Control Register for details. If Port*n* and the global system interrupt is enabled, a "1" in an any of these bits will generate an interrupt to the host controller.

Reg. 02

**Control**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Not Used | | | | | | | | | | IEN3 | IEN2 | IEN1 | IEN0 | IENA | SRST |
| Read | Not Used | | | | | | | | | | IEN3 | IEN2 | IEN1 | IEN0 | IENA | SRST |

SRST ⇨ Soft Reset (1 followed by 0 = causes soft reset)

IENA ⇨ Global Interrupt Request Enable (1 = allows interrupts to occur)

IENn ⇨ Enable Interrupt of Port n (1 = allows port to generate an interrupt to host controller)

Reg. 04

**Interrupt Vector**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Read-only | | | | | | | | | | | | | | | |
| Read | Not Used | | | | | | | | | | | | ICH3 | ICH2 | ICH1 | ICH0 |

ICHn ⇨ Interrupt Channel (1 = channel is a source interrupt of the interrupt)

Notes:

1. The value of this register is placed on the data bus (bits 7-0) during an interrupt acknowledge cycle and then the register is cleared.
2. The Port Interrupt Status/Control Register should be used to determine the actual cause of the interrupt.

**Figure 3. M217 I/O Registers**

Reg. 20

## Command/Response

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Not Used | | | | | | | | Command | | | | | | | |
| Read | Not Used | | | | | | | | Command Response | | | | | | | |

Notes:

1. This register is used with the Parameter Registers (addresses $22_{16}$ and $24_{16}$). The Parameter Registers must be set up before sending a command value to the Command / Response Register. In the following command descriptions, they are called PARM0 and PARM1.
2. The Command Status Register (address $26_{16}$) acts as a handshake signal for the Command/Response Register. Writing a value to the Command/Response register clears the CPRDY bit (bit 0) in the Command Status Register indicating that the Command / Parameter registers are not ready for the next data. Usually, a "0" on CPRDY interrupts the M-Module microcontroller. After the microcontroller executes the command, it writes a response to the Response Register and Parameter Registers. After sending the response, the microcontroller sets CPRDY indicating it is ready for the next command.

Reg. 26

## Command Status

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Not Used | | | | | | | | | | | | | | | |
| Read | Not Used | | | | | | | | DONE | CERR | - | URDY | UPAS | - | RRDY | CRDY |

DONE ⇨ Done (1 = microcontroller has completed operation and all Command Status Register bits are valid)
CERR ⇨ Command/Parameter Error (1 = error occurred while communicating with module)
URDY ⇨ Microcontroller Ready (1 = ready to interact with the host)
UPAS ⇨ Microcontroller Pass (1 = microcontroller built-in self test passed)
RRDY ⇨ Response Ready (1 = the Response Register is valid)
CRDY ⇨ Command/Parameter Register Ready (1 = Command and Parameter Registers are empty and available for the next command to be written to the microcontroller, 0 = DONE bit is not valid. This bit is set by the microcontroller, and cleared in hardware by writing to the Command/Response Register ($20_{16}$).)

Reg. 28
Reg. 2A
Reg. 2C
Reg. 2E

## Interrupt Generation Port 1
## Interrupt Generation Port 2
## Interrupt Generation Port 3
## Interrupt Generation Port 4

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Not Used | | | | | | | | | | | | | | | Int |
| Read | Not Used | | | | | | | | | | | | | | | Int |

INT ⇨ Generate Interrupt (writing a 1 to this bit generates an interrupt)

Note: After the host computer writes to this register, a self-generated interrupt occurs. Refer to the Port Interrupt Status/Control Register, Bit 7.

**Figure 3. M217 I/O Registers (continued)**

Reg. 36

## FIFO Status

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Not Used | | | | | | | | | | | | | | | |
| Read | Not Used | | | | | | | | Port 4 RCV | Port 4 XMIT | Port 3 RCV | Port 3 XMIT | Port 2 RCV | Port 2 XMIT | Port 1 RCV | Port 1 XMIT |

Port n RCV ⇨ Port n RCV (0 = Port n receive FIFO buffer is empty)
Port n XMIT ⇨ Port n XMIT (1 = Port n transmit buffer is half full, 0 = transmit buffer is not half full, the host controller can write up to 1kbyte in the FIFO)

Note: The host controller can read data from a port only when the Port n VCV bit is a 1 indicating data is in the FIFO.

Reg. 38
Reg. 3A
Reg. 3C
Reg. 3E

## Port 1 Interrupt Status/Control
## Port 2 Interrupt Status/Control
## Port 3 Interrupt Status/Control
## Port 4 Interrupt Status/Control

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Not Used | | | | | | | | EX1 | - | NU | ERR | ETE | ERT | ERF | ETF |
| Read | Not Used | | | | | | | | EX1 | - | END | ERR | TE | RTO | RF | HF |

Read Bits:

END ⇨ End (1 = termination character has been received)
ERR ⇨ Error Source (1 = error has occurred with receiving data on the port)
EX1 ⇨ Self-Generation Interrupt (1 = host computer has generated a self-generation interrupt for the port. This interrupt is used to initialize a data transmit or receive by the driver with interrupt method. Refer to the Interrupt Generation Registers.)
HF ⇨ Transmit Half Full (1 = the transmit FIFO buffer has decreased to below half full. In this condition, the host controller can send up to 1kbytes to the transmit buffer.)
RF ⇨ Receive Filled Block (1 = the receive buffer for this channel is filled with a pre-defined amount of data. The host controller can read the data from the FIFO buffer)
RTO ⇨ Receive Time Out (1 = the microcontroller has not received data for a pre-defined time after the last BLOCK of data was sent to the FIFO. After receiving this interrupt request, the host controller should verify the actual data length or read the data via the FIFO Status Register)
TE ⇨ Transmit Error (1 = the port's transmit FIFO has become empty. This means the host controller can send at least 2048 bytes of data to the transmit FIFO without the FIFO becoming full.)

Write Bits:

ERR ⇨ Error Interrupt (1 = enables the Error Interrupt for the port)
ERF ⇨ FIFO Filled (1 = enables the Receive FIFO Filled Interrupt for the port)
ERT ⇨ Time-out (1 = enables the Receive Time-out Interrupt for the port)
ETE ⇨ Transmit FIFO Empty (1 = enables the Transmit FIFO Empty Interrupt for the port)
ETF ⇨ Transmit FIFO Half Full (1 = enables the Transmit FIFO Half Full Interrupt for the port)
EX1 ⇨ Self-Generation Interrupt (1 = enables the Self-Generation interrupt for the port)

Note: Always read the Interrupt Vector Register (at address $04_{16}$) to clear all sources of interrupts. After one interrupt source is disabled, a later interrupt request may be pending. That is, if the interrupt source is later enabled again, the interrupt request may immediately generate an interrupt unless Interrupt Status Register is read first.

**Figure 3. M217 I/O Registers (continued)**

Reg. 40 **Port 1 Transmit Receive**
Reg. 42 **Port 2 Transmit Receive**
Reg. 44 **Port 3 Transmit Receive**
Reg. 46 **Port 4 Transmit Receive**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Not Used | | | | | | | | Transmit Data Byte | | | | | | | |
| Read | Not Used | | | | | | | | Receive Data Byte | | | | | | | |

Notes:
1. This register can be read only when the BHE*n* bit in the FIFO Status Register (at address $36_{16}$) is a "1" *OR* the RF or RT0 or END bits of the Port Interrupt/Status Register for the port (addresses $38_{16}$ for Port 1, $3A_{16}$ for Port 2, $3C_{16}$ for Port 3, or $3E_{16}$ for Port 4) is set to a "1". If the host controller is set for interrupts, it should read all of the data in the Receive FIFO of the port when an interrupt is generated by RF or RTO.
2. Up to 1kBytes of data can be written to this register when BHF*n* bit in the FIFO Status Register (at address $36_{16}$) is a "0" or up to 2kBytes can be written if the TE bit of the Port Interrupt/Status Register for the port (addresses $38_{16}$ for Port 1, $3A_{16}$ for Port 2, $3C_{16}$ for Port 3, or $3E_{16}$ for Port 4) is set to a "1".

Reg. 22 **Parameter Register 0**
Reg. 24 **Parameter Register 1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Reserved | | | | | | | | Command Parameter | | | | | | | |
| Read | Reserved | | | | | | | | Result of Command | | | | | | | |

Notes:
1. The Parameter Registers must be set up before writing a command to the Command / Response Register (0x20).
2. When an 8-bit parameter is required, use only Parameter Register 0. When a 16-bit parameter is required, Parameter Register 0 is used as bits 7 - 0; Parameter Register 1 is used as bits 15 - 8.
3. Refer to the Command / Response Register for a list of commands and the appropriate parameters.
4. At power-on / reset, all bits are set to 0.

Reg. 80-EF **ID EEPROM**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write | Reserved | | | | | | | | | | | | | | | |
| Read | Reserved | | | | | | | | | | | | | ECS | ESK | EDIO |

EDIO ⇨ EEPROM Serial Data
ESK ⇨ EEPROM Shift Clock
ECS ⇨ EEPROM Chip Select

Notes:
1. The ID EEPROM register allows you to access the contents of the ID EEPROM which contains sixty-four 16-bit works of M-Module ID and VXI M-Module data.
2. CAUTION: Do not attempt to write to Bit00 of the ID EEPROM register. You could overwrite the contents of the EEPROM.

**Figure 3. M217 I/O Registers (continued)**

### 3.2.2 Command/Response Values

The host computer communicates with the on-board microcontroller through the Command/Response Register (0x20), the Parameter 0 Register (0x22) and the Parameter 1 Register (0x24). The command must be logically OR'd with the values below to specify the port number.

| Port | Binary | Hex |
|---|---|---|
| 1 | 0000 0000 | 00 |
| 2 | 0100 0000 | 40 |
| 3 | 1000 0000 | 80 |
| 4 | 1100 0000 | C0 |

For example, to set Port 2's transmitter baud rate (command 0x21) use the command 0x61 (0x21 + 0x40).

#### 3.2.2.1 Query Test Value Command (0x00)

This command is used for test and debugging. Set test values in PARM0 and PARM1 with the Set Command Test Value Command (0x20) and then read them back with this command. The test values have no special meaning and are any values you want to use.

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: testvalue1 | PARM1: testvalue0 |
| Default/Reset Value: | PARM0: 0x55 | PARM1: 0xAA |

3.2.2.2 Query Transmitter Baud Rate Command (0x01)

This command returns a code representing the current port's transmitter baud rate. The baud rate is returned in the Result Parameters register. Set the transmitter baud rate with the Set Transmitter Baud Rate Command (0x21).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: baud rate code | PARM1: none |
| Default/Reset Value: | PARM0: 0x0B | PARM1: 0x00 |

| Baud Rate Code | Exact Baud Rate |
|---|---|
| 00 | 75 |
| 01 | 110 |
| 02 | 38400 |
| 03 | 150 |
| 04 | 300 |
| 05 | 600 |
| 06 | 1200 |
| 07 | 2000 |
| 08 | 2400 |
| 09 | 4800 |
| 0A | 1800 |
| 0B | 9600 |
| 0C | 19200 |

3.2.2.3 Query Receiver Baud Rate Command (0x02)

This command returns a code representing the current port's receiver baud rate. The baud rate is returned in the Result Parameters register. The table above lists the baud rate code and the exact baud rate. Set the receiver baud rate with the Set Receiver Baud Rate Command (0x22).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: baud code | PARM1: none |
| Default/Reset Value: | PARM0: 0x0B | PARM1: 0x00 |

3.2.2.4 Query Data Transfer Parity Mode Command (0x03)

This command returns a code representing the data transfer parity mode. To set the parity mode, use the Set Data Transfer Parity Mode Command (0x23).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: parity mode code | PARM1: none |
| Default/Reset Value: | PARM0: 0x04 | PARM1: 0x00 |

| Parity Mode Code | Parity Type |
|---|---|
| 00 | Even |
| 01 | Odd |
| 02 | Force 0 |
| 03 | Force 1 |
| 04 | None |

3.2.2.5 Query Character Length Command (0x04)

This command returns a code representing the character data length. To set the character data length, use the Set Character Length Command (0x24).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: character length code | PARM1: none |
| Default/Reset Value: | PARM0: 0x03 | PARM1: 0x00 |

| Character Length Code | Character Length |
|---|---|
| 00 | 5 |
| 01 | 6 |
| 02 | 7 |
| 03 | 8 |

3.2.2.6 Query Stop Bit Length Command (0x05)

This command returns a code representing the stop bit length. To set the Stop Bit length, use the Set Stop Bit Length Command (0x25).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: stop bit length code | PARM1: none |
| Default/Reset Value: | PARM0: 0x07 | PARM1: 0x00 |

| Stop Bit Length Code | Stop Bit Length | | Stop Bit Length Code | Stop Bit Length |
|---|---|---|---|---|
| 00 | 0.563 | | 08 | 1.563 |
| 01 | 0.625 | | 09 | 1.625 |
| 02 | 0.688 | | 0A | 1.688 |
| 03 | 0.750 | | 0B | 1.750 |
| 04 | 0.813 | | 0C | 1.813 |
| 05 | 0.875 | | 0D | 1.875 |
| 06 | 0.938 | | 0E | 1.838 |
| 07 | 1.000 | | 0F | 2.000 |

3.2.2.7 Query RTS/CTS Mode Command (0x06)

This command returns a code representing the RTS/CTS Mode. To set the RTS/CTS mode, use the Set RTS/CTS Mode Command (0x26).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: RTS/CTS Mode code | PARM1: CTS Monitor |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| PARM0 Code | RTS/CTS Mode | | PARM1 Code | CTS Monitor |
|---|---|---|---|---|
| 00 | Idle | | 00 | CTS Monitor OFF |
| 01 | On | | 01 | CTS Monitor ON |
| 02 | Off | | | |
| 03 | Standard | | | |
| 04 | IBfull | | | |

3.2.2.8 Query DTR/DSR Mode Command (0x07)

This command returns a code representing the DTR/DSR Mode. To set the DTR/DSR Mode, use the Set DTR/DSR Mode Command (0x27).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none; | PARM1: none |
| Result Parameters: | PARM0: DTR/DSR Mode code | PARM1: DSR Monitor |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| PARM0 Code | DTR/DSR Mode | | PARM1 Code | DSR Monitor |
|---|---|---|---|---|
| 00 | Idle | | 00 | DSR Monitor OFF |
| 01 | On | | 01 | DSR Monitor ON |
| 02 | Off | | | |
| 03 | Standard | | | |
| 04 | IBfull | | | |

3.2.2.9 Query Pace Mode Command (0x08)

This command returns a code representing the Pace (Xon/Xoff) Mode. To set the Xon/Xoff mode, use the Set Pace Mode Command (0x28).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none; | PARM1: none Result Parameters: |
| | PARM0: Xon/Xoff Mode code | PARM1: none Default/Reset |
| Value: | PARM0: 0x00 | PARM1: 0x00 |

| Code | Pace Mode |
|---|---|
| 00 | Tx & Rx Pace Off |
| 01 | Tx Pace On |
| 02 | RX Pace On |
| 03 | Tx & Rx Pace On |

3.2.2.10 Query BLOCK Size Command (0x09)

This command returns a code representing the BLOCK size. The PARM0 and PARM1 registers are both 16-bit registers. The lower 12 bits of each register express the BLOCK length from 1 to 2048 bytes. The upper four bits are not used. To set the BLOCK size, use the Set BLOCK Size Command (0x29).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters | PARM0: BLOCK low code | PARM1: BLOCK high code. |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x08 (2048 bytes) |

3.2.2.11 Query Port Mode Command (0x0A)

This command returns a code representing the port mode. To set the port mode, use the Set Port Mode Command (0x2A).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: Port Mode code | PARM1: none. |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| Code | RTS/CTS Mode |
|---|---|
| 00 | Normal |
| 01 | Auto-Echo |
| 02 | Local Loop |
| 03 | Remote Loop |

3.2.2.12 Query Line Status Command (0x0B)

This command returns a code representing the status of the individual RS-232 handshake lines and pacing controller information.

| | | |
|---|---|---|
| Set Parameters: | PARM0: none; | PARM1: none |
| Result Parameters: | PARM0: Line Status Code; | PARM1: none. |
| Default/Reset Value: | PARM0: 001100xxb | PARM1: 0x00 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Signal Line | TOFF | - | DTR | RTS | ROFF | - | DSR | CTS |

TOFF = 0 means XOFF has not been sent (default), TOFF = 1 means that XOFF has been sent.
DTR = 0 means DTR is ON, DTR = 1 means DTR is OFF (default).
RTS = 0 means RTS is ON, RTS = 1 means RTS is OFF (default).
ROFF = 0 means that XON has been received IF XOFF has been received previously, or it means that XOFF has not yet been received (def), ROFF = 1 means that XOFF has been received.
DSR = 0 means DSR is ON, DSR = 0 means DSR is OFF (default).
CTS = 0 means CTS is ON, CTS = 1 means CTS is OFF (default).

3.2.2.13 Query Filled Number in Receiver FIFO Command (0x0C)

This command returns an unsigned integer representing the amount of data in the receiver FIFO when the host received an interrupt.

| | | |
|---|---|---|
| Set Parameters: | PARM0: none; | PARM1: none |
| Result Parameters: | PARM0: FIFO Low Byte | PARM1: FIFO High Byte (lower 3 bits) |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

3.2.2.14 Query Error Code Command (0x0D)

This command returns a code representing one or more errors that have occurred. Reading this register clears the register.

| | | |
|---|---|---|
| Set Parameters: | PARM0: none; | PARM1: none |
| Result Parameters: | PARM0: Error Code | PARM1: none |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Signal Line | Not Used | Frame Error occurred during receive | Parity Error detected | Overflow Error detected | Not Used | Receiver Buffer is full | Not Used | Not Used |

3.2.2.15 Query Filled Number in Receiver Buffer Command (0x0E)

This command returns an unsigned integer indicating the number of bytes that has received by the port. The value does not include data in the receive FIFO if the receive FIFO is empty. If the receive FIFO is not empty, then the number will include the BLOCK data which has been moved into the receive FIFO (that generated an interrupt).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: Low Byte | PARM1: High Byte |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

3.2.2.16 Query Error Mode Command (0x13)

This command returns a value representing the established error mode. For actual error code refer to the Query Error Code Command (0x0D) command. To set the error mode, refer to the Set Error Mode Command (0x33) command.

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: Error Mode | PARM1: none |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| Code | Error Mode |
|---|---|
| 00 | Ignore Errors |
| 01 | Acknowledge Errors & Stop |

3.2.2.17 Query Start Threshold Command (0x14)

This command returns the start threshold value which is used to control the buffer space where DTE requests DCE restart line transfer activity. To set the Start Threshold anywhere in the 16KB buffer, use the Set Start Threshold Command (0x34).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: Low Code | PARM1: High Code |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x20 (8k buffer is full) |

3.2.2.18 Query Stop Threshold Command (0x15)

This command returns the stop threshold value which is used to control the buffer space where DTE requests DCE stop line transfer activity. To set the Stop Threshold anywhere in the 16kbyte buffer, use the Set Stop Threshold Command (0x35).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: Low Code | PARM1: High Code |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x28 (10k buffer full) |

3.2.2.19 Query Parity Check Mode Command (0x1A)

This command returns the current Parity Check Mode for the serial port. To set the parity check mode, use the Set Parity Check Mode Command (0x3A).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: Parity Check Mode | PARM1: none |
| Default/Reset Value: | PARM0: 0x01 | PARM1: 0x00 |

| Code | Parity Check Mode |
|---|---|
| 00 | Off |
| 01 | On (default) |

3.2.2.20 Set Command Test Value Command (0x20)

This command is used for test and debugging. Set test values in PARM0 and PARM1 with this command and then read them back with the Query Command Test Value Command (0x00). The test values have no special meaning and are any values you want to use.

| | | |
|---|---|---|
| Set Parameters: | PARM0: testvalue0 | PARM1: testvalue1 |
| Result: | PARM0: testvalue1 | PARM0: testvalue0 |

3.2.2.21 Set Transmitter Baud Rate Command (0x21)

This command sets the current port's transmitter baud rate. To read the current transmitter baud rate value, use the Query Transmitter Baud Rate Command (0x01).

| | | |
|---|---|---|
| Set Parameters: | PARM0: baud rate code | PARM1: none |
| Default/Reset Value: | PARM0: 0x0B | PARM1: 0x00 |

| **Baud Rate Code** | **Exact Baud Rate** |
|---|---|
| 00 | 75 |
| 01 | 110 |
| 02 | 38400 |
| 03 | 150 |
| 04 | 300 |
| 05 | 600 |
| 06 | 1200 |
| 07 | 2000 |
| 08 | 2400 |
| 09 | 4800 |
| 0A | 1800 |
| 0B | 9600 (default) |
| 0C | 19200 |

3.2.2.22 Set Receiver Baud Rate Command (0x22)

This command sets the current port's receiver baud rate. To read the current baud rate value, use the Query Receiver Baud Rate Command (0x02). See the Set Transmitter Baud Rate Command (0x21) for a list of baud rates and codes.

| | | |
|---|---|---|
| Set Parameters: | PARM0: baud rate code | PARM1: none |
| Default/Reset Value: | PARM0: 0x0B | PARM1: 0x00 |

3.2.2.23 Set Data Transfer Parity Mode Command (0x23)

This command sets the data transfer parity mode. To read the current parity mode, use the Query Data Transfer Parity Mode Command (0x03).

| | | |
|---|---|---|
| Set Parameters: | PARM0: parity mode code | PARM1: none |
| Default/Reset Value: | PARM0: 0x04 (none) | PARM1: 0x00 |

| Parity Mode Code | Parity Type |
|---|---|
| 00 | Even |
| 01 | Odd |
| 02 | Force 0 |
| 03 | Force 1 |
| 04 | None |

3.2.2.24 Set Character Length Command (0x24)

This command sets the character data length. To read the current character data length, use the Query Character Length Command (0x04).

| | | |
|---|---|---|
| Set Parameters: | PARM0: character length code | PARM1: none |
| Default/Reset Value: | PARM0: 0x03 | PARM1: 0x00 |

| Character Length Code | Character Length |
|---|---|
| 00 | 5 |
| 01 | 6 |
| 02 | 7 |
| 03 | 8 |

3.2.2.25 Set Stop Bit Length Command (0x25)

This command sets the stop bit length. To read the current stop bit length, use the Query Stop Bit Length Command (0x05).

| | | |
|---|---|---|
| Set Parameters: | PARM0: stop bit length code | PARM1: none |
| Default/Reset Value: | PARM0: 0x07 | PARM1: 0x00 |

| Stop Bit Length Code | Stop Bit Length | | Stop Bit Length Code | Stop Bit Length |
|---|---|---|---|---|
| 00 | 0.563 | | 08 | 1.563 |
| 01 | 0.625 | | 09 | 1.625 |
| 02 | 0.688 | | 0A | 1.688 |
| 03 | 0.750 | | 0B | 1.750 |
| 04 | 0.813 | | 0C | 1.813 |
| 05 | 0.875 | | 0D | 1.875 |
| 06 | 0.938 | | 0E | 1.838 |
| 07 | 1.000 | | 0F | 2.000 |

Note: Upper four bits are reserved and must be set to 00.

3.2.2.26 Set RTS/CTS Mode Command (0x26)

This command sets the RTS/CTS Mode. To read the current RTS/CTS mode, use the Query RTS/CTS Mode Command (0x06).

| | | |
|---|---|---|
| Set Parameters: | PARM0: RTS/CTS Mode code | PARM1: CTS Monitor |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| PARM0 Code | RTS/CTS Mode | | PARM1 Code | CTS Monitor |
|---|---|---|---|---|
| 00 | No Change | | 00 | CTS Monitor OFF |
| 01 | On | | 01 | CTS Monitor ON |
| 02 | Off | | | |
| 03 | Standard | | | |
| 04 | IBfull | | | |

Note: PARM1 is only used when PARM0 is 00, 01, or 02.

3.2.2.27 Set DTR/DSR Mode Command (0x27)

This command sets the DTR/DSR Mode. To read the current DTR/DSR mode, use the Query DTR/DSR Mode Command (0x07).

Set Parameters: PARM0: DTR/DSR Mode code PARM1: none
Default/Reset Value: PARM0: 0x00 PARM1: 0x00

| PARM0 Code | DTR/DSR Mode | | PARM1 Code | DSR Monitor |
|---|---|---|---|---|
| 00 | No Change | | 00 | DSR Monitor OFF |
| 01 | On | | 01 | DSR Monitor ON |
| 02 | Off | | | |
| 03 | Standard | | | |
| 04 | IBfull | | | |

Note: PARM1 is only used when PARM0 is less than 03.

3.2.2.28 Set Pace Mode Command (0x28)

This command sets the Xon/Xoff Mode. To read the current Xon/Xoff mode, use the Query Xon/Xoff Mode Command (0x08).

Set Parameters: PARM0: Pace Mode code PARM1: none
Default/Reset Value: PARM0: 0x00 PARM1: 0x00

| Code | Pace Mode |
|---|---|
| 00 | Tx & Rx Pace Off |
| 01 | Tx Pace On |
| 02 | RX Pace On |
| 03 | Tx & Rx Pace On |

3.2.2.29 Set BLOCK Size Command (0x29)

This command sets the BLOCK size. The PARM0 and PARM1 registers are both 16 bit registers. The lower 12 bits of each register express the BLOCK length from 1 to 2048 bytes. Block Size should have a value greater than 1. The upper four bits are not used. To read the current BLOCK size, use the Query BLOCK Size Command (0x09).

Set Parameters PARM0: BLOCK low code PARM1: BLOCK high code.
Default/Reset Value: PARM0: 0x00 PARM1: 0x08 (2kBytes)

3.2.2.30 Set Port Mode Command (0x2A)

This command sets the port mode. To read the current Port mode, use the Query Port Mode Command (0A).

Set Parameters: PARM0: Port Mode code PARM1: watchdog timer 0 = off, 1 = on.
Default/Reset Value: PARM0: 0x00 PARM1: 0x01

| Code | Port Mode |
|---|---|
| 00 | Normal |
| 01 | Auto-Echo |
| 02 | Local Loop |
| 03 | Remote Loop |

Normal — In this mode, the M217 has the transmitter and receiver operating independently.

Auto-Echo — This mode sets the port to retransmit data automatically. The following conditions are true:
- Received data is clocked and retransmitted on the TX output.
- Receiver clock is used for the transmitter.
- Receiver must be enabled, but the transmitter does not need to be enabled.
- Received parity is checked but not regenerated for transmission.

Local Loop — In this diagnostic mode, the following are true:
- The transmitter output is internally connected to the receiver input.
- The transmitter clock is used for the receiver.
- The TxD output is held high.
- The RxD input is ignored.
- Both transmitter and receiver must be enabled.
- The host computer to the M217 continues normally.

Remote Loop — In this mode, the following are true:
- Received data is reclocked and transmitted on the TxD output.
- The receiver clock is used for the transmitter.
- The received data is not transferred to the host and error status conditions are inactive.
- The received parity is not checked and is not regenerated for transmission.
- The receiver must be enabled, but the transmitter does not need to be enabled.
- Character framing is not checked and the stop bits are retransmitted as received.

Note: Be careful when switching between modes. The selected mode is activated or deactivated immediately upon mode selection, even if this occurs during a received or transmitted character. However, when switching out of Auto-Echo or Loop Back modes; if the mode change occurs immediately after a stop bit and the transmitter is enabled, the transmitter will remain in Auto-Echo mode until the entire stop bit is retransmitted.

3.2.2.31 Start Receiver Command (0x2B)

This command causes the port to start receiving data.

Set Parameters: PARM0: 0 PARM1: none

3.2.2.32 Stop Receiver Command (0x2C)

This command stops the receiver immediately. Characters in the receiver FIFO and receiver buffer are available for transfer to the host computer.

Set Parameters: PARM0: 0 PARM1: none

3.2.2.33 Start Transmitter Command (0x2D)

This command causes the transmitter to start transmitting data from its buffer.

Set Parameters: PARM0: 0 PARM1: none

3.2.2.34 Stop Transmitter Command (0x2E)

This command causes the transmitter to stop transmitting.

Set Parameters: PARM0: 0 PARM1: none

3.2.2.35 Clear Receiver Buffer Command (0x2F)

This command clears the receiver buffer but not the receiver FIFO memory.

Set Parameters: PARM0: 0 PARM1: none

3.2.2.36 Clear Transmitter FIFO Command (0x30)

This command clears the transmitter FIFO memory. The host computer should stop sending data to the transmitter FIFO.

Set Parameters: PARM0: 0 PARM1: none

3.2.2.37 Open Port Command (0x31)

This command sets the active to its default conditions. To close the current port, use the Close Port Command (0x32).

| | | |
|---|---|---|
| Set Parameters: | PARM0: Open Code | PARM1: none |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| Open Code | Meaning |
|---|---|
| 00 | Open single port |
| 01 | Open all four ports |

The default conditions are:

| Transmitter Characteristics | Setting | | Receiver Characteristics | Setting |
|---|---|---|---|---|
| Transmitter | Disabled | | Receiver | Disabled |
| Transmitter Baud Rate | 9600 | | Receiver Baud Rate | 9600 |
| Character Length | 8 | | Block Timeout | Enabled |
| Stop Bit Length | 1 | | Error Mode | Ignore |
| Parity Mode | None | | Start Threshold | 8kBytes |
| Handshake | Off | | Stop Threshold | 10kBytes |
| RTS/CTS | Off | | Parity Check | On |
| DTR/DSR | Off | | Start Threshold | 20h |
| BLOCK Size | 2048 | | Stop Threshold | 28h |
| Channel Mode | Normal | | | |

3.2.2.38 Close Port Command (0x32)

This command closes the port immediately. After the command executes, the selected port(s) are set to disabled transmitter, disabled receiver, receiver buffer, and transmitter FIFO are cleared. To open a port and set it to its default conditions, use the Open Port Command (31).

| | | |
|---|---|---|
| Set Parameters: | PARM0: Close Code; | PARM1: none |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| Open Code | Meaning |
|---|---|
| 00 | Close single port |
| 01 | Close all four ports |

3.2.2.39 Set Error Mode Command (0x33)

This command sets the error mode. For actual error code refer to the Query Error Code Command (0x0D) command. To read the current error mode, use the Query Error Mode Command (0x13). In the Ignore mode, the M217 continues activity when it detects an error; it will generate an interrupt to the host computer. In the STOP mode, an interrupt is generated and the M217 stops receiver activity.

| | | |
|---|---|---|
| Set Parameters: | PARM0: Error Mode; | PARM1: none |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x00 |

| **Error Mode Code** | **Error Mode** |
|---|---|
| 00 | Ignore |
| 01 | Acknowledge Errors & Stop |

3.2.2.40 Set Start Threshold Command (0x34)

This command sets Start Threshold value which controls the buffer space. When the number of bytes in the receiver buffer drops to or below the Start Threshold, and some form of pacing has been set, the M217 indicates it is ready to receive. The Start threshold must be less than the Stop Threshold. To read the current Start Threshold, use the Query Start Threshold Command (0x14).

| | | |
|---|---|---|
| Set Parameters: | PARM0: Start Low Code | PARM1: Start High Code |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x20 (8KB space avail.) |

3.2.2.41 Set Stop Threshold Command (0x35)

This command sets the Stop threshold value which controls the buffer space. When the number of characters in the receiver buffer goes above the Stop Threshold value, and some form of pacing has been set, the M217 indicates it is not ready to receive data. You can set Stop Threshold anywhere in the 16kbyte buffer space, but it must never be equal to 0. The Start threshold must be less than the Stop Threshold. To read the current Stop Threshold, use the Query Stop Threshold Command (0x15).

| | | |
|---|---|---|
| Set Parameters: | PARM0: Stop Low Code | PARM1: Stop High Code |
| Default/Reset Value: | PARM0: 0x00 | PARM1: 0x28 (10KB space avail.) |

3.2.2.42 Set Parity Check Mode Command (0x3A)

This command sets the parity check mode. To read the current parity check mode, use the Query Parity Check Mode Command (0x1A).

| | | |
|---|---|---|
| Set Parameters: | PARM0: Parity Check Mode; | PARM1: none |
| Default/Reset Value: | PARM0: 0x01 | PARM1: 0x00 |

| **Parity Check Code** | **Parity Check Mode** |
|---|---|
| 00 | Off |
| 01 | On |

3.2.2.43 Query FIFO Depth Command (0x40)

This command returns the FIFO size used in the transmitter/receiver port.

| | | |
|---|---|---|
| Set Parameters: | PARM0: none; | PARM1: none |
| Result Parameters: | PARM0: FIFO depth code; | PARM1: none |
| Default/Reset Value: | PARM0: 0x22 | PARM1: 0x00 |

| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| Signal Line | Transmitter FIFO (in KBytes) | | | | Receiver FIFO (in KBytes) | | | |

3.2.2.44 Query Firmware Version Command (0x80)

This command returns the current firmware version number.

| | | |
|---|---|---|
| Set Parameters: | PARM0: none; | PARM1: none |
| Result Parameters: | PARM0: version number; | PARM1: none |

### 3.2.2.45 Query Self Test Result Command (0xC0)

This command returns an unsigned integer with the results of the self test.
To initiate a self test, use the Start Self Test Command (0xE0).

| | | |
|---|---|---|
| Set Parameters: | PARM0: none | PARM1: none |
| Result Parameters: | PARM0: self test code | PARM1: none |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Error | Port 4 | Port 3 | Port 2 | Port 1 | RAM 4 | RAM 3 | RAM2 | RAM1 |

PORT n: 0 = port's receiver and transmitter work in auto-echo mode, 1 = failure.
RAM n : 0 = port's buffer is ok, 1 = failure

### 3.2.2.46 Start Self Test Command (0xE0)

This command initiates the internal self test. The test mode is set by the Parameter Register ( a "1" means to test that port, a "0" means do not test that port). To read the results of the self test, use the Query Self Test Result Command (0xC0).

| | | |
|---|---|---|
| Set Parameters: | PARM0: self test code | PARM1: none |

| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| Error | - | - | - | - | Port 4 | Port 3 | Port 2 | Port 1 |

### 3.2.3 Module Identification

The M217 supports the identification function called IDENT. This IDENT function provides information about the module and is stored in a sixteen-word deep (32 byte) serial PROM. Access is accomplished with read/write operations on the last address in IOSpace (hex FE) and the data is read one bit at a time. The PROM is compatible with a standard IC 9603 type PROM. For specific timing information refer to the 9603 or compatible PROM data sheet. Data should not be written to the PROM.

The module also supports the VXI-IDENT function. This function is not part of the approved ANSI/VITA 12-1996 standard. This extension to the M-module IDENT function increases the size of the PROM to 64 words and includes VXI compatible ID and Device Type Registers. Details are shown in Table IV.

**Table IV. M/MA Module PROM IDENT Words**

| Word | Description | Value (hex) |
|---|---|---|
| 0 | Sync Code | 5346 |
| 1 | Module Number | 067D |
| 2 | Revision Number [1] | 0001 |
| 3 | Module Characteristics [2] | 1868 |
| 4-7 | Reserved | 0000 |
| 8-15 | M-Module Specific | 0000 |
| 16 | VXI Sync Code | ACBA |
| 17 | VXI ID | 0FFF [3] |
| 18 | VXI Device Type [4] | F25A (M217) |
| 19-31 | Reserved | 0000 |
| 32-63 | M-Module Specific | 0000 |

Notes:

1) A Revision Number greater than 1 indicates that the module was manufactured by C&H Technologies.
2) The Module Characteristics bit definitions are:

| Bit(s) | Description |
|---|---|
| 15 | 0 = no burst access |
| 14/13 | unused |
| 12 | 0 = does not need ±12V |
| 11 | 1 = needs +5V |
| 10 | 0 = no trigger outputs |
| 9 | 0 = no trigger inputs |
| 8/7 | 00 = no DMA requestor |
| 6/5 | 11 = interrupt type |
| 4/3 | 01 = 16-bit data |
| 2/1 | 00 = 8-bit address |
| 0 | 0 = no memory access |

3) The VXI ID of 0xFFF is the identification value for Hewlett-Packard. C&H has left the ID equal to this value to allow operation with existing E2290A software drivers. The revision number (see note 1) can be used to identify the module as manufactured by C&H.
4) The VXI Device Type word contains the following information:

| Bit(s) | Description |
|---|---|
| 15-12 | $F_{16}$ = 256 bytes of required memory |
| 11-0 | $260_{16}$ = C&H specified VXI model code for M217 |

## 4.0 OPERATION

### 4.1 REGISTER PROGRAMMING

The M217 is a register-based instrument that is controlled through a series of I/O registers described in Section 3.2.1. The exact method of accessing and addressing the I/O registers is dependent on the M-Module carrier used to interface the module to your data acquisition or test system. Refer to the carrier's documentation for information on the address mapping of an M-Module's I/O registers and to your system software documentation for details on data access.

Typically a high level driver is available to aid in control of the module. Refer to the software driver documentation for instructions on using the driver.

### 4.2 MICROCONTROLLER COMMANDS

The host computer sends commands to and receives data from the microcontroller on the M217 M-Module. Microcontroller register commands are described later in this chapter. The proper sequence to send a command to the M217 microcontroller is:

1. Read the value of the CPRDY bit (bit 0) in the Command Status Register (0x26) to verify the microcontroller is ready to receive a new command (bit value = 1). If bit value = 0 then the microcontroller is busy, do not send the command.
2. Write the parameters to the Parameter Registers 0 and 1 (0x22 and 0x24).
3. Write the command (an integer, listed in this chapter as a hexadecimal number) to the Command / Response Register (0x20). The command specifies the port. Writing to the Command Register signals the M-Module microcontroller to accept the command and parameter value.
4. Read the value of the CPRDY bit (bit 0) and the DONE bit (bit 7) in the Command Status Register (0x26) to verify that the microcontroller has completed the command. Both bit values should be 1.
5. Read the CERR bit (bit 6) of the Command Status Register (0x26) to ensure an error did not occur. A bit value of 0 indicates an error did not occur.
6. Read the resulting value (if any) from the parameter registers.

### 4.3 RECEIVER OPERATION

The following discussion describes general receiver operation. If you are not using a high-level driver, then you should follow these general guidelines. Refer to Figure 4.

1. The host computer must open the port and enable the port to receive data. The host can send the Start Receiver Command (0x2B). If necessary, set the port's characteristics (baud rate, stop bits, handshake mode, etc.). After the receiver is enabled, the port begins waiting for a start bit from the device.
2. As data is received, the UART on the M-Module stores the data in the buffer and continues receiving data. Each port has a 16 KByte buffer.

3. 3. When the amount of data in the buffer is greater than the defined BLOCK size, or if the amount of data in the buffer is less than a full BLOCK and the BLOCK timeout occurs, the M-Module microcontroller moves the data from the buffer to the Receiver FIFO memory, if the FIFO is empty. The microcontroller then sets the Receiver Ready Bit (bit 1) in the Port's Interrupt Status/Control Register (0x38, 0x3A, 0x3C, 0x3E).
4. If the appropriate registers are set the microcontroller interrupts the host controller indicating that data is in the receiver FIFO memory. Note that each port has the same interrupt priority. If multiple ports are receiving data the host must determine which port interrupted by polling the individual Port Interrupt Status/Control Registers (0x38, 0x3A, 0x3C, 0x3E).
5. When the host receives the interrupt it should read all of the data from the receiver FIFO. Read the data through the Port Transmit/Receive Register (0x40, 0x42, 0x44, or 0x46). The host can determine the FIFO status in two ways; either read the exact number of bytes in the BLOCK or by monitoring the FIFO Status Register (0x36).
6. After the data in the FIFO has been read and the FIFO is empty, the host computer must wait until another receiver interrupt occurs.
7. When the host computer is finished receiving all the data, the host disables the port by either sending the Close Port Command (0x32) or the Stop Receiver Command (0x2C). These commands terminate receiver operation immediately. Note: the Close command clears the receiver buffer; characters in the FIFO are still available.

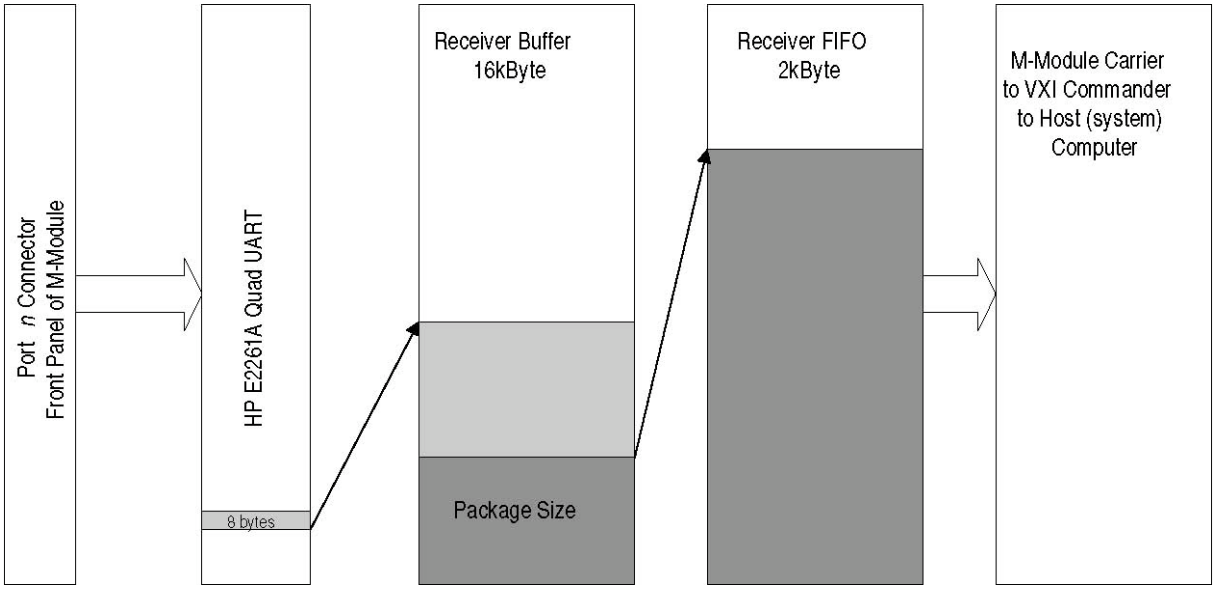



**Figure 4. Receiver Data Flow**

## 4.4 TRANSMITTER OPERATION

The following discussion describes general transmitter operation. If you are not using a high-level driver, then you should follow these general guidelines. Refer to Figure 5.

1. The host computer must open and enable the port to transmit data by sending the Open Port Command (0x31) and the Start Transmitter Command (0x2D).
2. The host then sends data to be transmitted to the Port Transmit/Receive Register (0x40, 0x42, 0x44, or 0x46). This automatically sends the data to the Transmit FIFO. If the Transmit FIFO changes from more-than-half-full to half-full, an interrupt occurs, the host (system) computer can then send up to 1KBytes of data. If the transmitter FIFO changes from non-empty to empty, an interrupt occurs, and the host can send up to 2KBytes of data.
3. The host computer can also monitor the FIFO Status Register to determine how full the Transmit FIFO is.
4. The host computer disables the transmitter by sending the Stop Transmitter Command (0x2E) or the Close Port Command (0x32). If the Close Port Command is executed, all of the data stored in the FIFO is lost. New data cannot be loaded in the FIFO if the port is disabled.

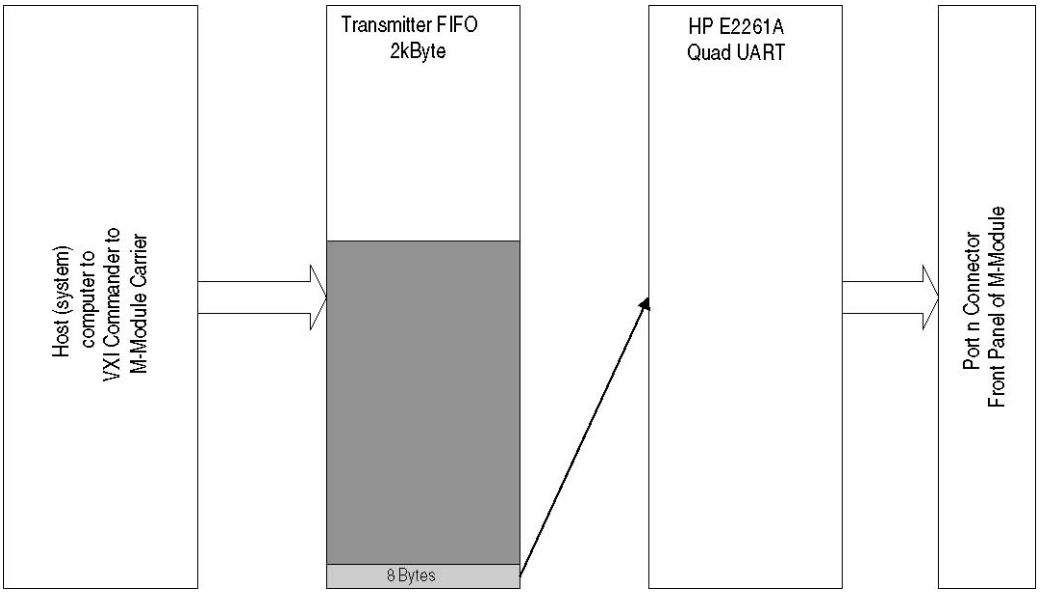
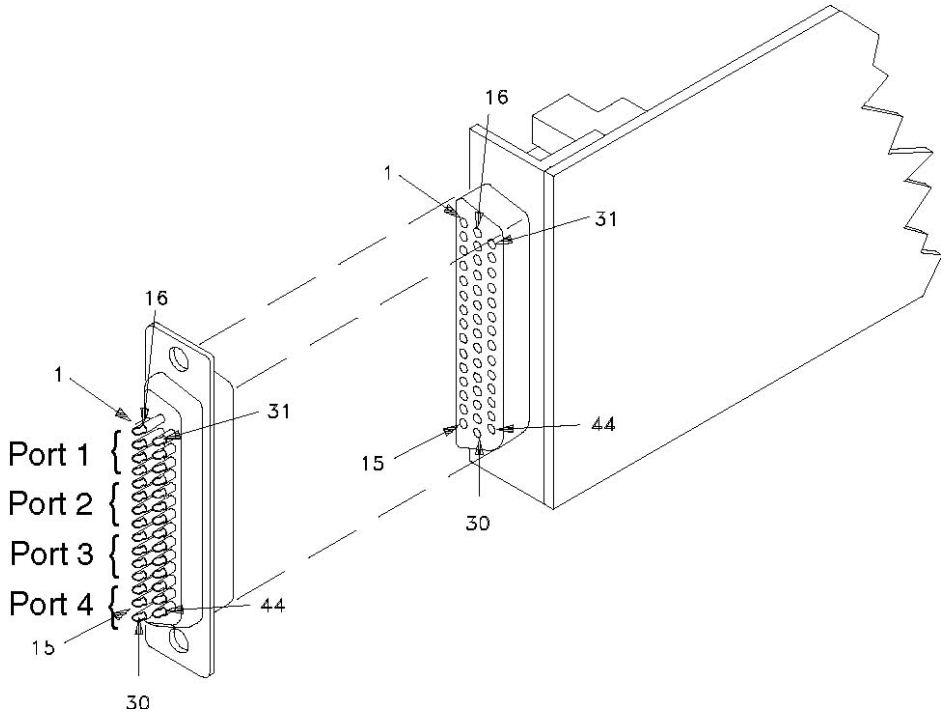



**Figure 5. Transmitter Data Flow**

# APPENDIX A: CONNECTORS




| M217 Quad RS-232 M-Module Pin Name and Function | | Port 1 Pin #s | Port 2 Pin #s | Port 3 Pin #s | Port 4 Pin #s | | 9-Pin (Male) Connector Pin # |
|---|---|---|---|---|---|---|---|
| DTR | Data Terminal Ready | 1 | 5 | 9 | 13 | | 4 |
| TD | Transmitted Data | 2 | 6 | 10 | 14 | | 3 |
| RD | Received Data | 3 | 7 | 11 | 15 | | 2 |
| RTS | Request to Send | 16 | 20 | 24 | 28 | | 7 |
| CTS | Clear to Send | 17 | 21 | 25 | 29 | | 8 |
| DSR | Data Set Ready | 18 | 22 | 26 | 30 | | 6 |
| SG | Signal Ground | 4, 8, 12, 19, 23, 27 | | | | | 5 |
| | Chassis Connection | 31, 32, 33, . . . 43, 44 | | | | | - |

If you use a shielded cable, connect the shield to a chassis connector pin on the 44-pin connector. Do not connect the other end of the shield.

**Figure A-1. Front Panel I/O Signals**

**N O T E S:**

## READER'S COMMENT FORM

Your comments assist us in improving the usefulness of C&H's publications; they are an important part of the inputs used for revision.

C&H Technologies, Inc. may use and distribute any of the information that you supply in any way that it believes to be appropriate without incurring any obligation whatsoever. You may, of course, continue to use the information, which you supply.

Please refrain from using this form for technical questions or for requests for additional publications; this will only delay the response. Instead, please direct your technical questions to your authorized C&H representative.

COMMENTS: _____

_____

_____

_____

_____

_____

_____

_____

Thank you for helping C&H to deliver the best possible product. Your support is appreciated.

Sincerely,

F. R. Harrison
President and CEO

**INSTRUCTIONS**

In its continuing effort to improve documentation, C&H Technologies, Inc. provides this form for use in submitting any comments or suggestions that the user may have. This form may be detached, folded along the lines indicated, taped along the loose edge (DO NOT STAPLE), and mailed. Please try to be as specific as possible and reference applicable sections of the manual or drawings if appropriate. Also, indicate if you would like an acknowledgment mailed to you stating whether or not your comments were being incorporated.

NOTE: This form may not be used to request copies of documents or to request waivers, deviations, or clarification of specification requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

——————————— (Fold along this line) ———————————

——————————— (Fold along this line) ———————————

_____
_____
_____

Place Stamp Here

C&H Technologies, Inc.
Technical Publications
445 Round Rock West Drive
Round Rock, Texas 78681-5012