Drive Interface Module

Multibus Form Factor part no. 770.91.10
Microbus Form Factor part no. 814.52.10

Industrial
CONTROLS

Instruction Manual J-3697

RELIANCE ELECTRIC

The information in this user's manual is subject to change without notice.

WARNING

ONLY QUALIFIED PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THE CONTROLLED EQUIPMENT SHOULD INSTALL, ADJUST, OPERATE, AND/OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL AND OTHER MANUALS APPLICABLE TO YOUR INSTALLATION. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

WARNING

RELIANCE STRONGLY RECOMMENDS THE USE OF AN EXTERNAL, HARDWIRED EMERGENCY STOP CIRCUIT OUTSIDE THE PROGRAMMABLE CONTROLLER CIRCUITRY. THE EMERGENCY STOP CIRCUIT MUST DISABLE THE SYSTEM IN CASE OF IMPROPER OPERATION. UNCONTROLLED MACHINE OPERATION MAY RESULT IF THIS PROCEDURE IS NOT FOLLOWED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

CAUTION

THE DRIVE INTERFACE MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE. DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Table of Contents

1.0 Introduction **1-1**
1.1 Additional Information 1-1

2.0 Mechanical/Electrical Description **2-1**
2.1 Mechanical Description 2-1
2.2 Electrical Description 2-3
2.2.1 High Speed Registers 2-5
2.2.2 Low Speed Registers 2-5

3.0 Installation **3-1**
3.1 Wiring Considerations 3-1
3.2 Initial Installation 3-1

4.0 Programming **4-1**
4.1 Register Organization 4-1
4.2 Module Status and Control Registers 4-3
4.2.1 Module ID Code Register 4-3
4.2.2 Module Configuration Control Register 4-3
4.2.3 Module Configuration Status Register 4-4
4.3 Port Status and Control Registers 4-4
4.3.1 Port Control Register 4-5
4.3.2 Port Status Register 4-6
4.4 Output Registers 4-8
4.5 Input Registers 4-8
4.6 Application Notes 4-9
4.6.1 General Application Programming 4-9
4.6.2 Powering Up/Down 4-9
4.6.3 Drive Application Notes 4-9
4.6.4 Software Handshaking 4-10
4.6.5 Resetting Faults on VCI and VGI 4-10
4.6.6 Hybrid Mode 4-10

Appendices

Appendix A
Technical Specifications (Multibus Only) A-1

Appendix B
Module Block Diagram B-1

Appendix C
Cable Connections C-1

Appendix D
Fault Codes D-1

List of Figures

Figure 2.1 – Multibus Drive Interface Module Faceplate 2-2
Figure 2.2 – Timing Example 2-4
Figure 2.3 – Typical and Worst Case Data Update 2-4

Figure 4.1 – Drive Interface Memory Map 4-3
Figure 4.2 – Module ID Code Register (register 2) 4-3
Figure 4.3 – Module Configuration Control Register (register 3) 4-4
Figure 4.4 – Module Configuration Status Register (register 4) 4-4
Figure 4.5 – Port Status and Control Register Map 4-5
Figure 4.6 – Port Control Register 4-5
Figure 4.7 – Truth Table for "TRANSMIT ENABLE," "DATA RECEIVED," and "FAULT" Bits 4-6
Figure 4.8 – Port Status Register 4-7
Figure 4.9 – Fault Detection Times 4-8

1.0 INTRODUCTION

The products described in this instruction manual are manufactured by Reliance Electric Industrial Company.

The Multibus Drive Interface module allows an AutoMax™ Processor to control Invertron™ VCI and VGI Drives. Control is accomplished via application programs that write drive parameters to the appropriate registers in the Drive Interface module's dual port RAM. Each of the four ports on the Multibus Drive Interface module can be connected to the serial port on a Microbus form-factor version of the same module mounted in a VCI or VGI Drive. The Drive Interface appears to the two hosts (AutoMax Processor and VCI or VGI Drive) as conventional dual port RAM. This mode of drive control is called the "serial" mode.

The Microbus Drive Interface module is functionally equivalent to the Multibus Drive Interface module, with the exception that the Microbus version has a single communication port instead of four. All status and control registers on the Microbus Drive Interface are controlled by the Drive microprocessor and are not accessible to the user. This instruction manual focuses on the Multibus Drive Interface module and its operation in an AutoMax system.

Data is transferred between the Drive Interface modules in the AutoMax rack and VCI/VGI Drives over two full-duplex serial communication links, one to transmit high speed data (closed loop and control), and one to transmit low speed data (configuration and monitoring). Drive Interface module dual port registers map directly to VCI and VGI Drive parameter registers. Refer to the appropriate Drive instruction manual for the specific mapping of retrieved parameters and firmware compatibility.

1.1 Additional Information

This manual describes the module's functions and specifications. It also includes a detailed overview of installation and servicing procedures. You can find additional information in the following instruction manuals depending on the equipment in your application.

Keypad Instruction Manual	49-1251
VCI Instruction Manual	49-1249
VGI Instruction Manual	49-1313
VCI Man Machine Interface Manual	49-1300
VGI Man-Machine Interface Manual	49-1301
AutoMax Programming Executive Manual V2.0	J-3884
AutoMax Programming Executive Manual V3.0	J2-3004

2.0 MECHANICAL/ELECTRICL DESCRIPTION

The following is a description of the mechanical and electricm chamcteristics of the module.

2.1 Mechanical Description

The Multibus Drive Interface module is a printed circuit board assembly that plugs into the backplane of the AutoMax rack. It consists of the printed circuit board, a faceplate, and a protective enclosure. The faceplate contains tabs at the top and bottom to simplify removing the module from the rack. On the back of the module are two edge connectors that connect to the system backplane.

The Microbus Drive Interface is a printed circuit board assembly that plugs into the Microbus connector in the Drive. It comes with a mounting bracket to facilitate installation. The module does not have an enclosure or faceplate. Normally, this module is provided in the form of a kit for a specific horsepower Power Module.

The faceplate of the Multibus module contains a seven-segment LED for error code display for the module. Note that the same diagnostic information is available in the Port Status register, but that the values of the respective errors given in this register are different from those that appear on the faceplate LED. See Appendix D for a listing of the diagnostic codes as they appear on the faceplate LEDs and in the Port Status register.

A green status LED, labeled "OK" is located below the seven-segment LED on the Multibus Drive Interface. This LED is on when the module has passed its power-up diagnostics. Below the "OK" LED are four ports, each equipped with a 9-pin subminiature D-shell connector. The ports are numbered 1 through 4 beginning with the top port. A cable with four twisted pairs is required to connect a port on the Drive Interface module to a Microbus Drive Interface module. Refer to figure 2-1 for the Multibus Drive Interface module faceplate and Appendix C for a pin description.

The Microbus Drive Interface contains a green "READY" LED that is turned on when the module has completed its power-up diagnostics. It also contains one 9-pin subminiature D-shell connector, as well as a Microbus connector with the corresponding flat ribbon cable already attached.

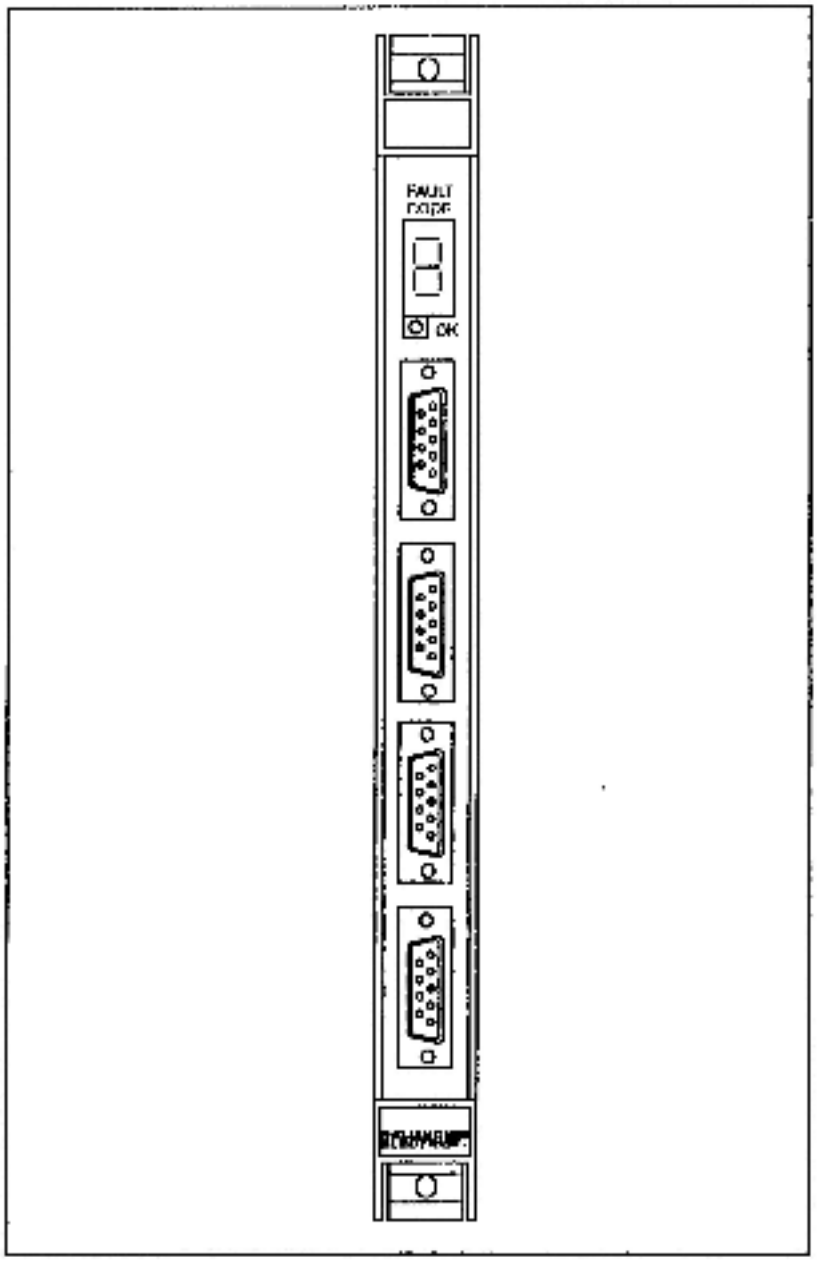


Figure 2.1 - Multibus Drive Interface Module Faceplate

2.2 Electrical Description

The Drive Interface module's memory consists of 2K X 16 dual port RAM. One port allows the host processor (AutoMax Processor, or VGI or VCI Drive) to read and write to the RAM, while the other port allows the Drive Interface module to read and write to the RAM. Illegal writes are prevented. Both the host processor and the Drive Interface module access the dual port RAM in 16-bit (word) format.

The Drive Interface module contains arbitration logic that prevents both the module and the host processor from accessing the dual port RAM at the same time. The arbitration logic allows access to the dual port RAM on a first come first served basis.

The host should not write to the RAM until the module has successfully completed its power-up testing and initialization. If the module detects a fault, a fault bit is set in the Port Status register and a fault code is written into the Status register and to the LED display. Transmission of data is inhibited.

For each port, communication takes place over two full duplex serial communication links that operate at a rate of 250 Kbaud. The communication cable (see Appendix C) is sold separately. Data is formatted into packets of seven registers each and transmitted via pulse encoding and a transformer-coupled transmission line.

One communication link is used to transmit seven registers of data as a single packet. These registers are referred to as "high speed registers." The second communication link is used to transmit 245 registers of data as 35 packets of 7 registers each. These registers are referred to as "low speed registers." The high speed registers and one packet of low speed registers are sent simultaneously. See sections 2.2.1 and 2.2.2 for more information about high speed registers and low speed registers. See figure 2.3 for typical and worst case timing on the two links.

The timing example in figure 2.2 shows the first three transmission sequences that would occur if two serial ports on the Multibus Drive Interface module were being used to control two Drives.

Sequence	Registers
1	Port 1 High speed registers Port 1 Low speed registers (1st packet of 7 registers)
2	Port 2 High speed registers Port 2 Low speed registers (1st packet of 7 registers)
3	Port 1 High speed registers Port 1 Low speed registers (2st packet of 7 registers)

Figure 2.2 – Timing Example

The communication link provides error checking on transmitted data and a data update watchdog time-out function. Bad packets are discarded, and the corresponding registers are not updated until the next update, assuming there are no errors in the next packet sequence. There is no acknowledgment of the reception of packets, and lost packets due to framing, checksum, or invalid packet number errors are not re-transmitted. The receiver simply discards a bad packet and does not update the low speed data input registers for which the packet had data. The receiver must wait until the next packet sequence to attempt to update the data.

Low speed packet updating will be re-tried 4 times, for a total of 5 attempts; high speed packet updating will be retried 2 times, for a total of 3 attempts. If valid data is not received within the allowed number of packet sequences (retries), the "DATA RECEIVED" bit in the status register of the corresponding port will be turned off. See figure 2.3.

	Typical	Worst Case*
1 port mode		
Low Speed Registers	42 ms	250 ms
High Speed Registers	1.2 ms	6 ms
4 port mode		
Low Speed Registers	200 ms	1 second
High Speed Registers	5.5 ms	20 ms

*Worst case takes into account a maximum of 5 re-try attempts on low-speed registers and 3 re-try attempts on high-speed registers. Any more re-try attempts will result in a fault.

Figure 2.3 - Typical and Worst Case Data Update

In general, if the worst-case times shown above are exceeded, while the Multibus Drive Interface has been receiving high speed data (i.e., while the Port Status "DATA RECEIVED" bit is = 1), then the Port Status register "FAULT" bit will be set equal to 1 and fault code "1" or "2" will be written to the high byte of the Port Status register. Note that the faceplate will display "C" or "c", respectively. The Port Status "DATA RECEIVED" bit will then be set to false (0).

2.2.1 High Speed Registers

The high speed communication link provides for the transmission of 14 data registers (7 output [read/write] registers, and 7 input [read only] registers) for a port between the Multibus and Microbus Drive Interface. As long as the transmission of data has been enabled on the Multibus Drive Interface, packets with this data are sent/received continuously with as little idle time as possible between packets.

2.2.2 Low Speed Registers

The low speed communication link provides for the transmission of 490 (245 output and 245 input) registers for a port between the Multibus and Microbus Drive Interface. The 245 registers of data transmitted to the remote port are sent in a sequence of 35 data packets. Each data packet contains 7 registers of data. Packets are sent sequentially, with as little idle time as possible between packets as long as the transmission line is enabled and there are no faults.

3.0 INSTALLATION

This section describes how to install the Multibus and Microbus Drive interface modules.

DANGER

THE USER IS RESPONSIBLE FOR CONFORMING WITH THE NATIONAL ELECTRICAL CODE AND ALL OTHER APPLICABLE LOCAL CODES. WIRING PRACTICES, GROUNDING, DISCONNECTS, AND OVER-PROTECTION ARE OF PARTICULAR IMPORTANCE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SERIOUS BODILY INJURY OR LOSS OF LIFE.

3.1 Wiring Considerations

Make certain that the Drive ground jumper is connected properly to ensure both safety and the reduction of electrical noise which could adversely affect the serial communication link between the Drive interface modules. Note that the Microbus Drive interface shield will be connected to chassis ground only if the bracket it is mounted on is connected to chassis ground, and if a ground strap is installed on the Drive.

The specified communication cable for the serial link does not have the shield (pin 1) tied at either end. This situation is appropriate for short cable runs typically found in a cabinet, i.e., under 10 feet, in which electrical noise is kept to a minimum through proper wire seperation and the use of conduit.

If the application requires a longer communication cable (300M, maximum), the cable should be manufactured using the directions in Appendix C. The shield must be tied at one end of the cable only, preferably at the Drive end. There should be at least two twists per inch on the twisted wire pairs to minimize capacitance.

3.2 Initial Installation

Use the following procedure to install the module:

DANGER

THIS EQUIPMENT IS AT LINE VOLTAGE WHEN A-C POWER IS CONNECTED. DISCONNECT AND LOCKOUT ALL UNGROUNDED CONDUCTORS OF THE A-C POWER LINE. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Step 1. Turn off power to the system. All power to the rack and Drive should be off.

Step 2. Take the modules out of their shipping containers. Take them out of the anti-static bags, being careful not to touch the connectors on the back of the module. Check the setting of the four switches on the Microbus Drive interface module. Switches 1, 2, and 4 should be open (in the OFF position). Switch 3 should be closed (in the ON position). The module is shipped from the factory with these settings.

Step 3. For the Multibus Drive Interface, insert the module into the desired slot in the rack. Use a screwdriver to secure the module into the slot. There are no restrictions on slot location with the exception that the module may not be installed in a slot reserved for Processor modules, i.e., slots 1-4, or slot 0.

For the Microbus Drive Interface, first connect the flat ribbon cable on the module to the the microbus 60-pin connector, using the key slot to ensure a tight fit. Attach the mounting bracket visible on the upper left hand portion of the Drive's card carrier in order to position the module in the drive. Use the smaller brackets protruding from the right and bottom edges of the MACS printed circuit board in the Drive to connect the Microbus Drive Interface to the Drive. Make certain all connections are tight to ensure good connection to chassis ground. Note that some Drives are not equipped with mounting brackets. These brackets are provided as part of each Microbus Drive Interface kit (P/N 6EC4094, 6EC4095, or 6EC4096).

When both the Multibus and Microbus Drive Interface are installed properly, connect the serial communication cable between the Microbus Drive Interface 9-pin connector and the desired 9-pin connector on the Multibus Drive Interface. Be sure to securely tighten the cable retaining screws.

Step 4. Turn on power to the rack and Drive.

DANGER

THE REMAINING STEPS ARE MADE WITH POWER ON. EXERCISE EXTREME CARE BECAUSE HAZARDOUS VOLTAGE EXISTS. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Step 5. Verify the proper installation of both the Multibus Drive Interface module and the Microbus Drive Interface module by observing the green "OK" LED and "READY" LED, respectively. If power-up diagnostics are completed successfully by the modules, these lights should be on. Use the AutoMax Programming Executive to verify the Multibus Drive Interface module can be accessed by application software. Use the on-line I/O MONITOR option to verify the module ID code can be read on the module (see section 4.2.1).

4.0 PROGRAMMING

This section describes how data is organized in the Multibus Drive Interface module. Equivalent status and control registers on the Microbus Drive Interface module are under host, i.e., Drive, control and are not accessible to the user. The following conventions will be used to describe the module:

- Registers referred to by number signify Multibus Drive Interface module status and control registers, e.g., register 2.
- Registers referred to by name only refer to all such registers in each port of the Multibus Drive Interface module, e.g. the Port Control register.
- Bits (in either module or port status and control registers) will be referred to in uppercase and in quotation marks, e.g., the "TRANSMIT ENABLE" bit.

In addition, when it is necessary to discuss specific VCI and VGI parameters, they will be referred to by name in uppercase, e.g., LOOPBK_IN.

4.1 Register Organization

Each port in the Drive Interface module contains 512 16-bit words, or registers, of memory. The registers are numbered consecutively, with register 0 being the first register on the module, and register 2047 (register 512 of port 4) being the last. Registers 0-4 are used for module status and control.

Within each port's specific register range, the first three registers are used for port status and control. There are 252 output registers (R/W) and 252 input registers (Read only) for each port. The first seven registers in the output range and in the input range are used for application-specific high speed data transmission. The 245 remaining registers in the output range and in the input range are used for application-specific low speed data transmission. See figure 4.1 for a memory map.

Beginning Register Number	Register Type		Total # of Registers
0	Module Status and Control Registers		5
5	Port Status and Control Registers		3
8	High Speed Data Output Registers	P O R T 1	7
15	Low Speed Data Output Registers		245
260	High Speed Data Input Registers		7
267	Low Speed Data Input Registers		245
512	Reserved		5
517	Port Status and Control Registers		3
520	High Speed Data Output Registers	P O R T 2	7
527	Low Speed Data Output Registers		245
772	High Speed Data Input Registers		7
779	Low Speed Data Input Registers		245
1024	Reserved		5
1029	Port Status and Control Registers		3
1032	High Speed Data Output Registers	P O R T 3	7
1039	Low Speed Data Output Registers		245
1284	High Speed Data Input Registers		7
1291	Low Speed Data Input Registers		245
1536	Reserved		5
1541	Port Status and Control Registers		3
1544	High Speed Data Output Registers	P O R T 4	7
1551	Low Speed Data Output Registers		24
1796	High Speed Data Input Registers		7
1803	Low Speed Data Input Registers		245
2047			

Where output = R/W from Multibus host and Input = Read only from the Multibus host.

Figure 4.1 - Drive Interface Memory Map

4.2 Module Status and Control Registers

The module Status and Control registers on the Multibus Drive Interface module (registers 0–4) are used to configure the serial communications port and then verify their status. These registers are used for status and control of the module itself, not for each individual port. Each port has its own individual status and control registers, which are described in section 4.3.

Register	Description
0–1	not used
2	Module ID Code
3	Module Configuration Control
4	Module Configuration Status

4.2.1 Module ID Code Register

Register 2 identifies the module type. A value of 64250 (FAFAH) identifies the module as a Multibus Drive Interface module. This register is read only. See figure 4.2.

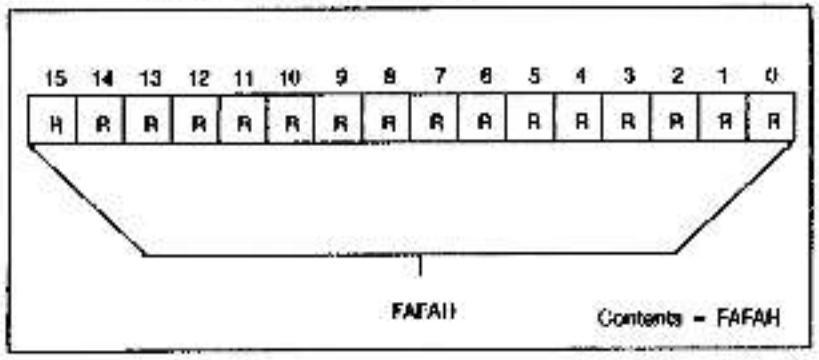


Figure 4.2 – Module ID Code Register (register 2)

4.2.2 Module Configuration Control Register

The Multibus Drive Interface module uses register 3, bit 2 ("PORT CONFIGURATION") to configure the module for single port (port 1) or multi-port communication. The status of this register is reflected in the Module Configuration Status register (see section 4.2.3) approximately 14ms after any change. See figure 4.3.

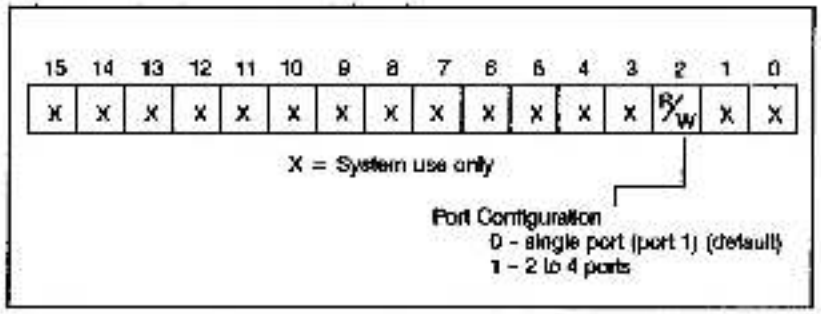


Figure 4.3 – Module Configuration Control Register (register 3)

The status of the "PORT CONFIGURATION" bit must be determined and verified in the Configuration Status register (register 4) before turning on the "TRANSMIT ENABLE" bit in any Port Control Register.

NOTE: Bit 10 of register 3 can be used for the same purpose as bit 2 because the high byte of this register (bits 8 - 15) is treated as the mirror image of the low byte (bits 0-7). All other bits in register 3 are under system control and must NOT be written to.

4.2.3 Module Configuration Status Register

The Multibus Drive Interface module uses register 4 to reflect the configuration written to register 3. Both bits 2 and 10 will be set if four-port operation was selected in register 3. This register is read only. See figure 4.4.

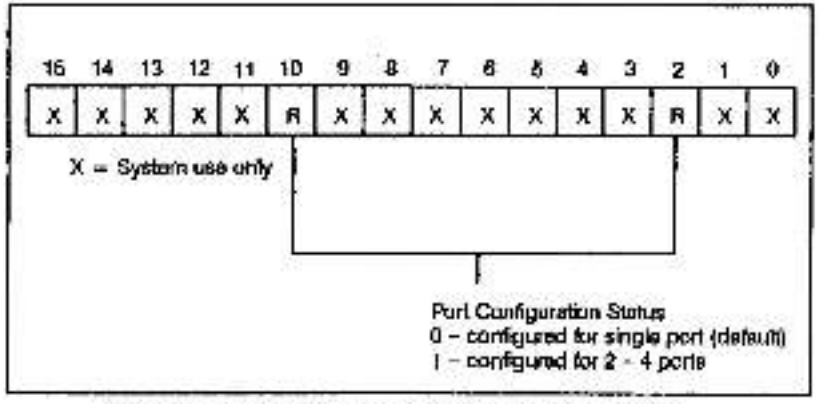


Figure 4.4 – Module Configuration Status Register (register 4).

Note that other bits in register 4 may be set. Ignore all bits except bits 2 and 10 in determining whether the configuration has been implemented correctly by the module.

4.3 Port Status and Control Registers

The Port Status and Control registers allow the host to monitor and control each port of a Multibus Drive Interface module. Figure 4.5 contains a memory map of the Port Status and Control registers.

These registers will be referred to by name, i.e., Port Control or Port Status, instead of register number to avoid confusion. See figure 4.5 for the appropriate register numbers for each port.

Port 1 Reg Number	Port 2 Reg Number	Port 3 Reg Number	Port 4 Reg Number	Description
6	517	1029	1541	Port Control
6	518	1030	1542	Port Status
7	519	1031	1543	Not Used

Figure 4.5 - Port Status and Control Register Map

4.3.1 Port Control Register

Bit 0, "TRANSMIT ENABLE," of the Port Control register on each port is used to enable and disable the transmission of the port's high and low speed registers by the Multibus Drive Interface. It may be set equal to 0 at any time to disable the transmission. Setting "TRANSMIT ENABLE" to 1 sends all registers for the port. These values will write over any values that may have been loaded into the Drive through another interface. This bit does NOT control transmission of data by the Microbus Drive Interface module, which begins transmitting data as soon as it passes its power-up diagnostics. See figure 4.6

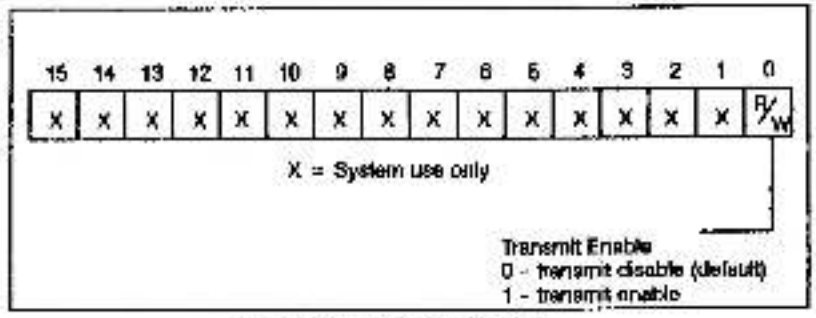


Figure 4.6 - Port Control Register

The programmer must make certain there is valid data in the output registers before enabling transmission. Note that if there is a communication fault of any kind, signaled via the Status register, transmission of data is inhibited regardless of the status of the "TRANSMIT ENABLE" bit. If the module clears the "FAULT" bit in the Port Status register and if the "TRANSMIT ENABLE" bit is set on, (assuming the fault does not re-occur) the module will begin transmitting data.

The application program should not set the "TRANSMIT ENABLE" bit equal to 1 until after the following.

- register 3 has been written to with the appropriate value by the application program
- register 4 is checked for correct configuration status
- high and low speed output registers have been initialized

– the "DATA RECEIVED" bit in the Status register has been set by the module

See figure 4.7 for a truth table showing the relationship between the "TRANSMIT ENABLE," "DATA RECEIVED," and "FAULT" bits.

"TRANSMIT ENABLE"	"DATA RECEIVED"	"FAULT"	VALID INPUT DATA	OUTPUT DATA BEING TRANSMITTED
0	0	0	No	No
0	0	1	No	No
0	1	0	Yes	No
0	1	1	Yes	No
1	0	0	No	Yes
1	0	1	No	No
1	1	0	Yes	Yes
1	1	1	Yes	No

Figure 4.7 – Truth Table for "TRANSMIT ENABLE," "DATA RECEIVED," and "FAULT" Bits

The "TRANSMIT ENABLE" bit can also be used to initiate drive fault recovery. This bit should be set to 0 for 11 ms when a fault is detected by the application program. The Drive Interface module will initiate recovery from a fault if it detects this bit going from a 0 to a 1 while it is in a faulted state (while the Port Status register "FAULT" bit is 1). The application program should set the "TRANSMIT ENABLE" bit to 1, wait at least 11 ms, and then read the Port Status register "FAULT" bit to see if the fault still exists. Because it is the rising edge of the enable bit that initiates fault recovery, the enable bit must be 0 for at least 11 ms and then set equal to 1 for at least 11 in order to ensure that the module detects a rising edge.

Note that for the VGI and VCI Drives, the condition that initiated the fault in the Drive may be deduced by reading the Drive's TRIP_CODE, PENDING_CODE, and WARNING_CODE registers for the particular port prior to executing the above-described fault recovery procedure.

4.3.2 Port Status Register

The Port Status register is used by the host to report communications faults and indicate when data has been received from the Microbus Drive Interface module. See figure 4.8.

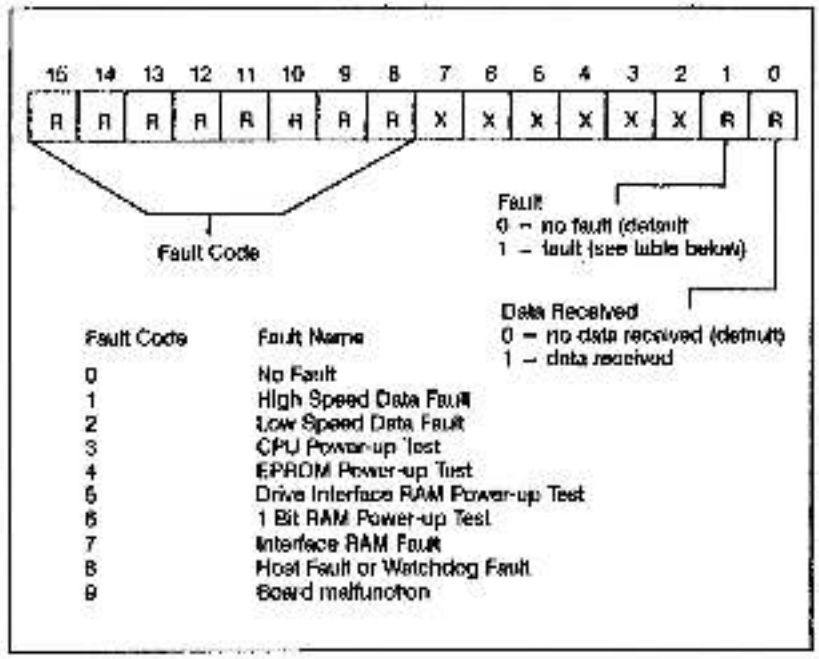


Figure 4.8 – Port Status Register

The "DATA RECEIVED" bit (bit 0) is set true (1) by the module whenever high speed data has been received within the the last X milliseconds and all low speed data has been received within the last Y milliseconds, where X and Y are defined as follows:

1 port mode X = 5, Y = 250
4 port mode X = 20, Y = 1000

Note that it is possible that if valid registers are received while the "FAULT" bit is set, the "DATA RECEIVED" bit will turn ON.

The "FAULT" bit (bit 1) is set equal to 1 by the module when a communications fault is detected by the Drive Interface module. The fault code indicating the nature of the fault is found in the high byte of this register. The status of the "FAULT" bit has no effect on the reception of data. This bit can be set and the module will continue to receive data over the serial link, assuming the link itself is operational. This bit is cleared (set equal to 0) when the module detects the "TRANSMISSION ENABLE" bit in the Port Control register change from 0 to 1 and no fault is present. See figure 4.9 for the amount of time required for fault detection by the module.

Mode	Type of Error	Time*
1-port mode	high-speed register transmission	4.76 ms
	low-speed register transmission	7.12 ms
4-port mode	high-speed register transmission	17.52 ms
	low-speed register transmission	27.9 ms

*Time between actual occurrence of error and its indication in the Port Status register.

Figure 4.9 – Fault Detection Times

The Fault Code byte (bits 8–15) identifies the type of communications fault detected by the module. The module writes a fault code into this byte upon detection of a fault if it is not already in a fault state, i.e., it records only the first fault encountered. If a fault is detected that affects more than one port (e.g., a power-up test), the corresponding code is written into the status register of each port. Note that the fault codes reflected in this register indicate the same information as the fault codes displayed on the Multibus Drive Interface faceplate LED, but that the codes displayed on the LED are different. See Appendix D for the correlation between LED codes and the codes found in this register.

4.4 Output Registers

The output registers contain application-specific data that is transmitted to the input registers of the remote port of the virtual dual port RAM. The remote port input register that will contain the output register's data is equal to the output register number plus 252. See figure 4.1 for a memory map of the registers of all ports. See the appropriate Drive instruction manual for a list of the applicable registers and their mapping in relation to the Drive Interface module.

Recall that all registers for a port will be transmitted when "TRANSMIT ENABLE" is on (1) in the Port Control register and the "FAULT" bit in the Port Status register is off (0). All registers must be initialized before the host takes control. The equivalent registers in the Drive will then be written over with the transmitted registers.

4.5 Input Registers

The input registers contain application specific data that is received from the remote port through the dual port RAM. The input registers contain data that is transmitted from the output registers of the remote port. The remote port output register that is the source of an input register's data is equal to the input register number minus 252. See figure 4.1 for a memory map of the registers on all ports. See the appropriate Drive instruction manual for a list of the applicable registers.

For the VCI and VCI Drives, drive registers outside of the memory map of the Microbus Drive Interface can be monitored via the pointer register P_SDP_MON. Registers outside of the memory map cannot be written to using the Drive Interface module. See the applicable Drive and Keypad Instruction manuals for more information about this register.

4.6 Application Notes

The following sections document common application programming issues.

4.6.1 General Application Programming

Application programming can be done most efficiently (in terms of memory utilization) using IOWRITE (type #3) statements in the application task(s) for variables that are not expected to change, i.e., that only need to be initialized. This eliminates the need to explicitly configure all registers. IOWRITE is a write command that references actual register location instead of name. If required, registers written to using IOWRITE statements can be tuned and monitored using the I/O Monitor in the AutoMax Programming Executive. Crucial variables that must be accessible to other application tasks in the AutoMax Processor should be defined through normal configuration statements. This is accomplished through IODEF statements in configuration tasks. For Programming Executive software that does not require a configuration task, i.e., V3.0 and later, the "Generic 32K Module" option in the Rack Configurator portion of the Programming Executive software.

Specifically configuring Drive Interface registers through a configuration task (or in the Rack Configurator) adds to the clarity and ease of understanding of the application program. It does so, however, at the expense of increased memory utilization.

4.6.2 Powering Up/Down

If the Drive is turned off while the Multibus Drive Interface is communicating with the Microbus Drive Interface module, the Multibus Drive Interface module will display a communication error code on the faceplate LED and write the code to the high byte of the Port Status register. This error must be cleared in the following manner.

Set the "TRANSMIT ENABLE" bit in the Multibus Drive Interface to 0 (OFF). Turn on power to the Drive. The Multibus Drive Interface module must wait for the "DATA RECEIVED" bit in the Port Status register to be set before turning on the "TRANSMIT ENABLE" bit in the Port Control register. Recall that enabling transmission will also clear the "FAULT" bit in the Port Status register if there are no other faults present. At this point, the Multibus Drive Interface is able to transmit data.

4.6.3 Drive Application Notes

All registers in VCI and VGI Drives are initialized at power-up with values stored in non-volatile RAM. As soon as the host takes control, all registers in the Drive are overwritten by the host register values. Whenever control is released, all registers in the drive are again initialized with the values stored in non-volatile RAM. Note that taking and releasing control can only be accomplished when the drive is stopped, i.e., in stand-by mode.

Note also that the Drive must be powered down if you want to replace user interface modules. The drive can determine what

interface modules are installed (Drive Interface module, Keypad, or Man-Machine Interface) ONLY on power-up.

4.6.4 Software Handshaking

The Drive parameters LOOPBK_IN and LOOPBK_OUT are provided for software handshaking between the Multibus and Microbus Drive Interface modules. The LOOPBK_IN (an output register from the Multibus Drive Interface) register must be set equal to the LOOPBK_OUT register (an input register from the Microbus Drive Interface) in the application program:

LOOPBK_IN% = LOOPBK_OUT%

LOOPBK_IN is multiplied by two (2) by the Drive microprocessor and written to LOOPBK_OUT, which is in turn transmitted to the Multibus Drive Interface. If the time between updates to LOOPBK_IN in the Drive is greater than one (1) second (3 seconds for later versions of the VGI and VCI Drive firmware) or LOOPBK_IN does not equal LOOPBK_OUT, the Drive is tripped. The software handshaking should be in a separate task that runs at a faster rate than the task that is writing data to the registers that will be transferred to the Drive. Note that timing of the software handshaking task must take into account the transmission speed of the serial link itself. Refer to section 2.2.

4.6.5 Resetting Faults on the VCI and VGI

Faults on the VCI and VGI Drives can be reset using the B_RESET parameter ONLY if there are no current or pending communication faults on the Drive Interface link and software handshaking is active. See section 4.3.1 for how to recover from communication faults. Before resetting Drive faults, try to determine the cause of the fault by examining the TRIP_CODE parameter output from the Drive. To reset the Drive fault, B_RESET must be turned on for a minimum of 50ms. It will require approximately 3 seconds for the Drive to actually clear the fault and be ready to run.

Before attempting to run the Drive again, verify that both TRIP_CODE and PENDING_CODE register outputs from the Drive are 0 to ensure there is no other error condition that will make the Drive trip again. Note that the "TRANSMIT ENABLE" bit in the Port Control register in the Multibus Drive Interface module must NOT be turned off before resetting Drive faults.

4.6.6 Hybrid Mode

In addition to Serial mode, the VCI and VGI Drives can utilize the Drive Interface module in "hybrid" mode. This mode is a variation of Local mode in which the Drive Interface module is used to provide only a speed reference to the Drive, and does not actually control it. See the appropriate Drive documentation for more information on Hybrid mode of the Drive.

Appendix A

Technical Specifications (Multibus Only)

Ambient Conditions

- Storage temperature: -20° to 85° C
- Operating temperature: 0° to 60° C
- Humidity: 5 to 95% non-condensing

Dimensions

- Height: 11.75 inches
- Width: 1.25 inches
- Depth: 7.375 inches

System Power Requirements

- +5 Volts: 1200 mA
- +12 Volts: 40 mA

Maximum Module Power Dissipation

- 8 Watts

Isolation Voltage

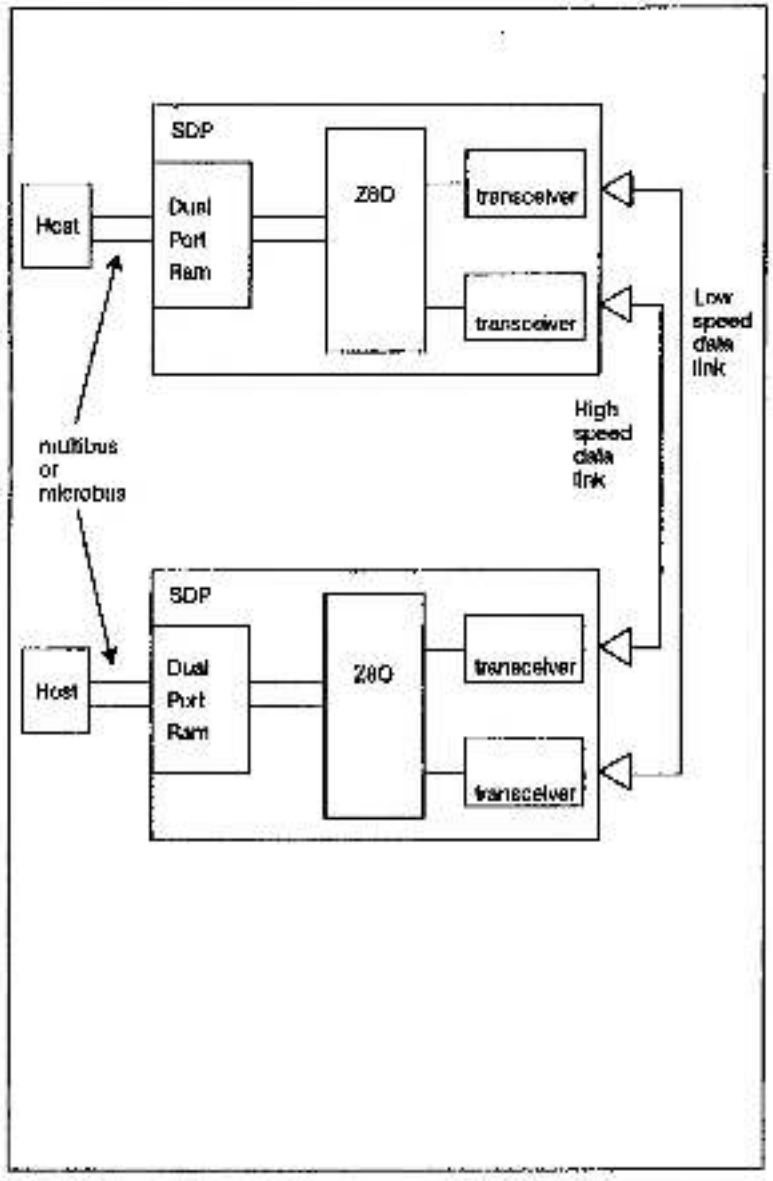
- 220 VAC at 50Hz

Maximum wire length between ports

- 300M

Appendix B

Module Block Diagram



Appendix C

Cable Connections

(Europe) Cable P/N 569.01.00 is sold separately.

(USA) Cable P/N 613201-xR, where x = length in inches, is sold separately.

where x = twisted pair wiring, 2 twists per inch

*Shield pin connects through 100 Ohm 1/4 Watt series resistor if connected. The shield pin is connected (at the Drive end) only on cable runs of over approximately 10 feet/3 meters

Appendix D

Fault Codes

Value in Port Status Register	LED Fault Code	Fault Name
0	(blank)	No Fault
1	C	High Speed Data Fault
2	c	Low Speed Data Fault
3	0	CPU Power-up Test
4	1	EPROM Power-up Test
5	2	Drive Interface RAM Power-up Test
7	6	Interface RAM Power-up Test
6	7	1 Bit RAM Power-up Test
9	(*)	Board Malfunction
8	d	Host Fault or Watchdog Fault

*When this fault occurs, the LED keeps its most recent value, which may be blank or any other fault code listed above.

Error Description

High Speed Data Fault – Indicates that, after receiving a set of valid high speed input register data, another set of of high speed input register data was not received within the maximum allowable time. The module will detect a high speed data fault whenever the Drive to which it is connected is turned off.

Low Speed Data Fault – Indicates that, after receiving every block of low speed input register data within the maximum allowable time, one or more blocks of low speed input register data was not received within the maximum allowable time.

All other errors indicate a malfunctioning module that is not user-serviceable.

RELIANCE CONTROLS
DOCUMENTATION IMPROVEMENT FORM

Document Number: _____

Page Number(s): _____

Comments (Please give chapter, page numbers or specific paragraphs that the change will affect. Include revisions from the comments or attach additional pages if necessary.)

_____

_____

_____

_____

_____

_____

_____

_____

_____

What will this improvement / suggestion provide?

_____

Originator: _____ Org: _____ Slate: _____ Zip: _____

Company: Phone: ( )

Address: _____ Date: _____

Reviewed / Wrong Address / Other	Follow-Up Action:
Author: _____	Date: _____

Thank you for your comments . . .

RELIANCE ELECTRIC

For additional information
1 Allen-Bradley Drive
Mayfield Heights, Ohio 44124 USA
Tel: (800) 241-2886 or (440) 646-3599
http://www.reliance.com/automax

Publication 1397-1.0 - July 1997