AutoMax®
Pulsetach Input Module

M/N 57C421B

Instruction Manual J-3680-3

RELIANCE ELECTRIC

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

DANGER

ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL AND OTHER APPLICABLE MANUALS IN THEIR ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

WARNING

REGISTERS AND BITS IN THE MODULE THAT ARE DESCRIBED AS "READ ONLY" OR FOR "SYSTEM USE ONLY" MUST NOT BE WRITTEN TO BY THE USER. WRITING TO THESE REGISTERS AND BITS MAY RESULT IN IMPROPER SYSTEM OPERATION. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

CAUTION: This 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. The plastic cover should not be removed. Failure to observe this precaution could result in damage to or destruction of the equipment.

Table of Contents

Appendices

List of Figures

1.0 INTRODUCTION

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

The Pulsetach Input module (M/N 57C421B) is used to accumulate pulses from a photo-electric pulsetach. The pulsetach can be either single- or dual-channel with 5V or 12 VDC inputs. The module contains a 24-bit counter, a 24-bit comparator, and a 16-bit internal timer. It can accept an input frequency up to 150 kHz. Digital inputs (5V to 12 VDC) are provided for a latch input, count stop input, and origin input.

The module can be programmed to interrupt on a variety of conditions: a periodic time interval, an external latch input, an external count stop input, a marker (Z) pulse and origin input, or a comparator equal condition.

In order to use interrupts on this module it must be located in a rack containing at least one Processor module. Interrupts cannot be used with Pulsetach Input modules located in remote racks.

This manual describes the functions and specifications of the module, how to install and service the module, and programming information.

1.1 Related Publications

You must be familiar with the instruction manuals which describe your system configuration. These may include, but are not limited to, the following:

- J-3675 AutoMax ENHANCED BASIC LANGUAGE INSTRUCTION MANUAL
- J-3676 AutoMax CONTROL BLOCK LANGUAGE INSTRUCTION MANUAL
- J-3650 AutoMax PROCESSOR MODULE INSTRUCTION MANUAL
- J2-3094 AutoMax ENHANCED LADDER LANGUAGE EDITOR
- Your Reliance AutoMax PROGRAMMING EXECUTIVE INSTRUCTION MANUAL.
- Your personal computer and DOS operating system manuals.
- IEEE-518 GUIDE FOR THE INSTALLATION OF ELECTRICAL EQUIPMENT TO MINIMIZE ELECTRICAL NOISE INPUTS TO CONTROLLERS FROM EXTERNAL SOURCES

The thick black bar shown on the right-hand margin of this page will be used throughout this instruction manual to signify new or revised text or figures.

1.2 Related Hardware

M/N 57C421 contains one AutoMax Pulsetach Input module. The module is used with Terminal Strip/Cable Assembly M/N 57C372, which must be ordered separately. This assembly is used to connect field signals to the faceplate of the module.

2.0 MECHANICAL/ELECTRICAL DESCRIPTION

The following is a description of the faceplate LEDs, field termination connectors, and the electrical characteristics of the module.

2.1 Mechanical Description

The Pulsetach Input module is a printed circuit board assembly that plugs into the backplane of an AutoMax? rack. It consists of the printed circuit board, a faceplate, and a protective enclosure.

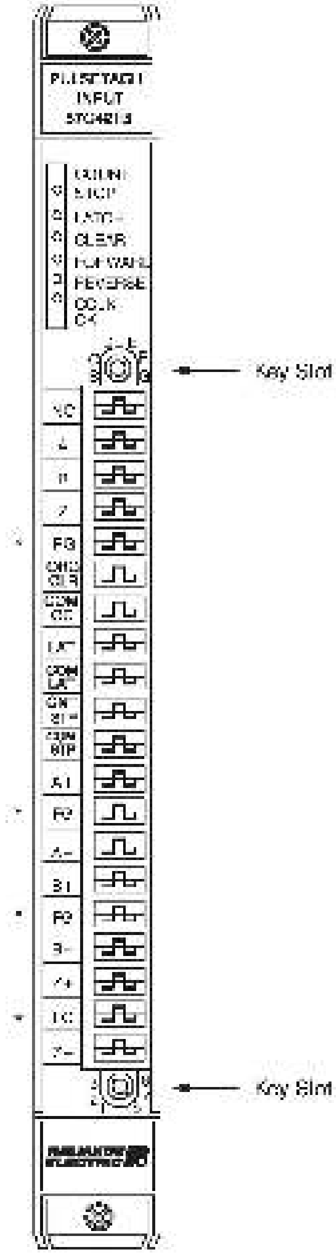
The faceplate contains tabs at the top and bottom to simplify removing the module from the rack. Module dimensions are listed in Appendix A. The back of the module contains two edge connectors that attach to the system backplane.

The faceplate of the module contains six LED module status indicators and a female connector socket. Input signals are brought into the module via a multi-conductor cable (M/N 57C372). One end of this cable has a plug that mates with the faceplate connector while the other end of the cable has stake-on connectors that are attached to a terminal strip for convenient field wiring connections. The module's faceplate connector socket and the cable's plug have provisions for keying the cable to its respective module to prevent re-insertion of the cable into the wrong module.

The six LEDs on the faceplate of the module are shown in figure 2.1. The first three LEDs are labeled "COUNT STOP", "LATCH" and "CLEAR". These LEDs correspond to the module's three external digital inputs. They turn on whenever their corresponding input is true (high) regardless of whether the input has been enabled on the module (see register 6).

The next two LEDs are labeled "FORWARD" and "REVERSE". These LEDs, when on, indicate whether the counter is counting pulses in the forward direction or in the reverse direction.

The last LED is labeled "CCLK OK". When this LED is on, it indicates that the constant clock (CCLK) signal is present on the backplane.



*FG = Frame Ground (Not Internally Connected)

Figure 2.1 - Module Faceplate

2.2 Electrical Description

The module contains a pulsetach-to-digital converter that supplies data to a 24-bit up/down counter. See figure 2.2. The counter counts up as it follows the pulses received from the pulsetach turning in the forward direction. It counts down as it follows the pulses received from the pulsetach in the reverse direction.

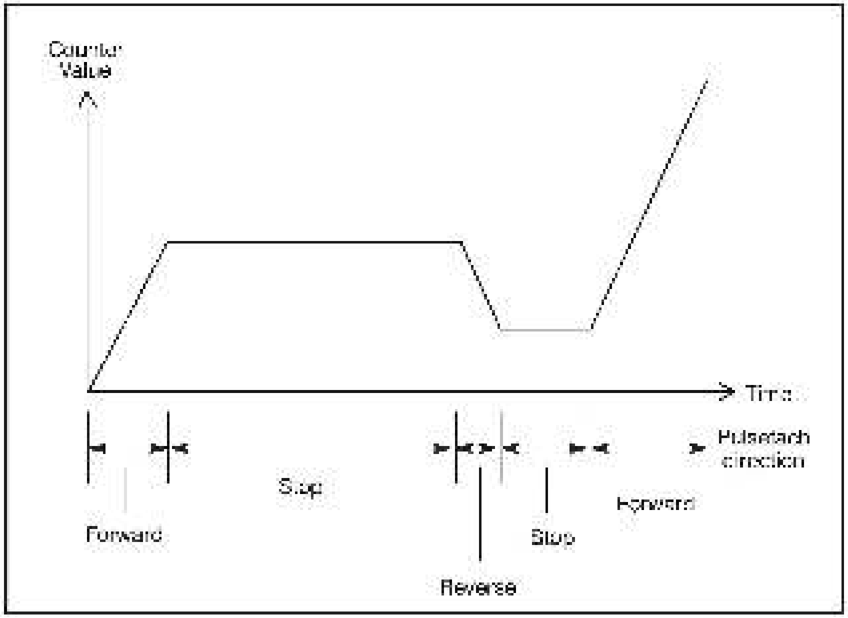


Figure 2.2 - Relationship of Pulsetach Direction to Counter Value

The counter data is latched (i.e., frozen in time) and transferred to latch registers at user-specified intervals. The application program accesses the counter data through the latch registers.

The module also contains an internal 200 kHz clock that can be used for timing operations.

The module's pulsetach input circuitry is shown in figure 2.3.

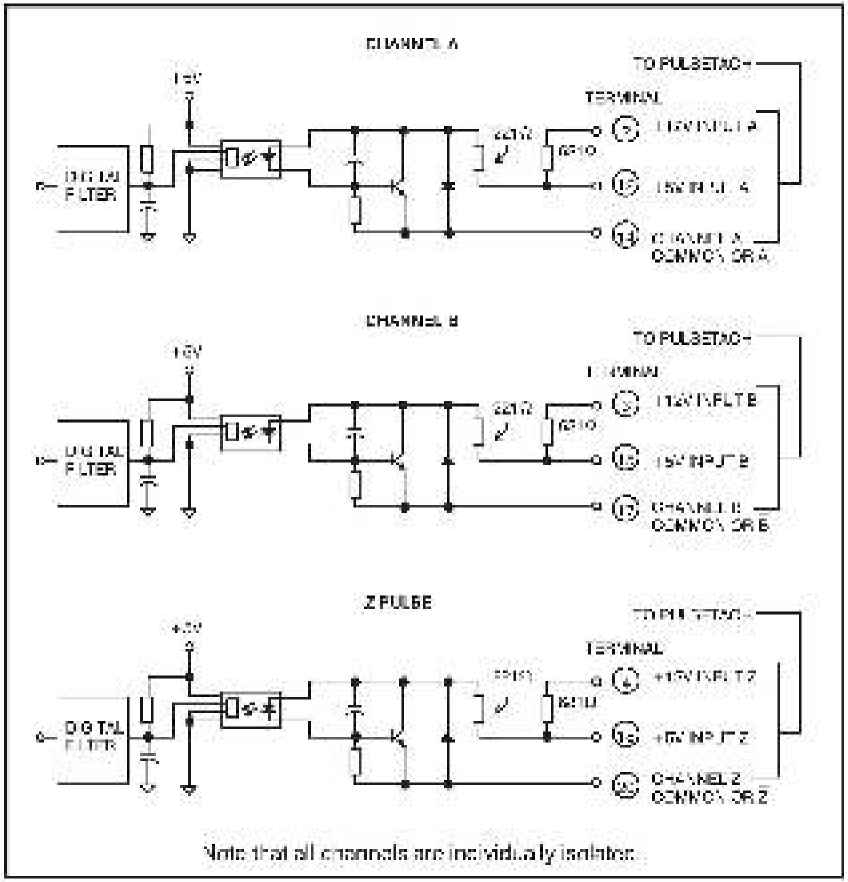


Figure 2.3 - Module Pulsetach Input Circuitry

The module provides three external digital inputs which are enabled through software (see register 6). Each input causes the module to perform a specific function at the occurrence of an external signal on that input as shown below:

- Origin Clear Input - Reset the counter
- External Latch Input - Read the counter
- External Count Stop Input - Stop the counter

The module's digital input circuitry is shown in figure 2.4.

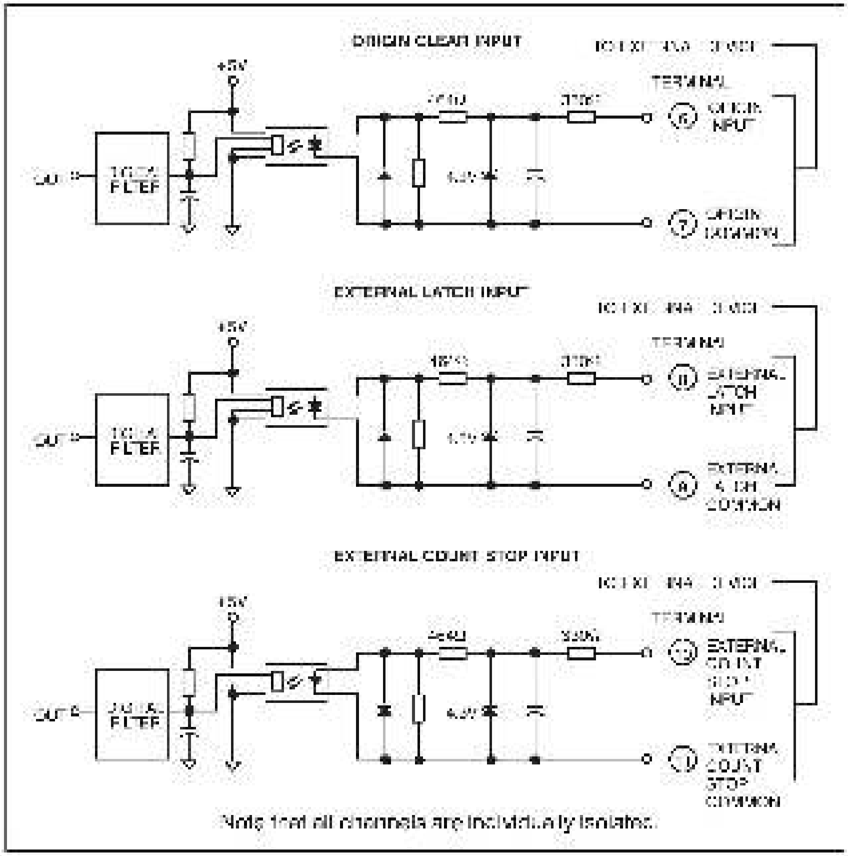


Figure 2.4 - Module Digital Input Circuitry

3.0 INSTALLATION

This section describes how to install and remove the module and its cable assembly.

CAUTION: The user is responsible for conforming with all applicable local, national, and international codes. Failure to observe this precaution could result in damage to, or destruction of, the equipment.

3.1 Wiring

The installation of wiring should conform to all applicable codes.

To reduce the possibility of electrical noise interfering with the proper operation of the control system, exercise care when installing the wiring from the system to the external devices. For detailed recommendations refer to publication IEEE 518.

3.2 Initial Installation

Use the following procedure to install the module:

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

Step 1 Remove power from the system. Power to the rack as well as all power to the wiring leading to the module should be off.

Step 2 Take the module out of its shipping container. Take the module out of the anti-static bag. Be careful not to touch the connectors on the back of the module.

Step 3 Insert the module into the desired slot in the rack. Refer to figure 3.1. Use a screwdriver to secure the module into the slot.

Step 4 Mount the terminal strip (from cable assembly M/N 57C372) on a panel. The terminal strip should be mounted to allow easy access to the screw terminals. Be sure the terminal strip is close enough to the rack so that the cable will reach between the terminal strip and the module. The cable assembly is approximately 60 inches long.

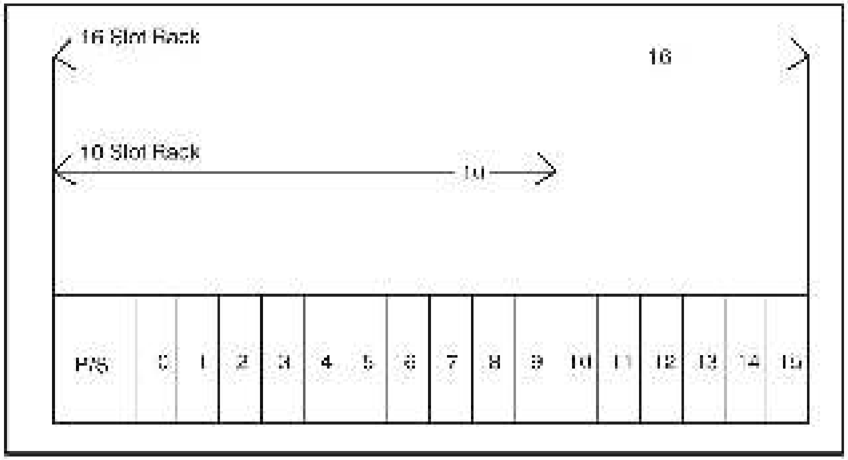


Figure 3.1 Rack Slot Numbers

Step 5. Attach the pulsetach but leave the mechanical coupling between the pulsetach and the motor or flowmeter unconnected.

Fasten the field wires from the pulsetach to the cable assembly's terminal strip. Typical field connections are shown in figures 3.2 to 3.5.

Note that 5V open-collector inputs require 464 ohm, 1/2 watt pull-up resistors while 12V open-collector inputs require 1000 ohm, 1/2 watt pull-up resistors. Also note that the output open-collector transistors in the pulsetach should have more than 12 mA of current driving capability.

Use twisted pair wire, connected as shown, for the cabling between the pulsetach and the terminal strip. If you use wire with less than 2 twists per inch, it should be shielded. Note that the shield should only be connected at one end. Ground the cable shield on the module side. The recommended twisted-pair wire is Belden™ 8761 cable or equivalent.

Cable length should not exceed 600 feet. Maximum operating cable length for your installation is dependent upon the type of cable you use and the way the pulsetach is wired to the module.

Step 6. Mount the pulsetach's external power supply. The external power supply should be able to provide either 5 V at 25 mA plus the pulsetach's power requirements or 12 V at 25 mA plus the pulsetach's power requirements. Check the specifications of the pulsetach you will be using.

Fasten the wires from the power supply to the pulsetach. For best results, the power supply voltage should be adjusted to provide the specified voltage at the pulsetach.

Step 7. You may need to add termination resistors to installations where the twisted pair cable length from the pulsetach to the module exceeds 200 feet. The resistor value should be selected dynamically to provide the proper waveform.

See figure 3.2 for typical termination resistor connections.

For 5 VDC inputs, the termination resistors connect between:

- terminals 12 and 14 (input A)
- terminals 15 and 17 (input B)
- terminals 18 and 20 (input Z).

For 12 VDC inputs, the termination resistors connect between:

- terminals 2 and 14 (input A)
- terminals 3 and 17 (input B)
- terminals 4 and 20 (input Z).

When a pulsetach is wired for open collector operation, the termination resistors should be placed as shown in figure 3.3.

Step 8. Insert the cable assembly's (M/N 57C372) field terminal connector into the mating half on the module. Use a screwdriver to secure the connector to the module.

Note that both the module and the field terminal connector are equipped with "keys" as shown in figure 3.1. These keys should be used to prevent the wrong cable from being connected to a module in the event that connector needs to be removed and then re-attached later.

At the time of installation, rotate the keys on the module and the connector so that they can be connected together securely. It is recommended that, for modules so equipped, the keys on each successive module in the rack be rotated one position to the right of the keys on the preceding module.

If you use this method, the keys on a particular connector will be positioned in such a way as to fit together only with a specific module, and there will be little chance of the wrong connector being attached to a module.

Step 9. Check the wiring and be sure all connections are tight.

Step 10. With the pulsetach disconnected from the motor, apply power to the rack and the pulsetach. Use an oscilloscope to test the signal from the pulsetach. The signal at the terminal strip should be a clean square wave of 5 or 12 volts.

Step 11. Verify the installation using the Programming Executive software. Refer to the AutoMax Programming Executive manual for more information.

Select the I/O Monitor function. For local I/O, enter the slot number and register number (0-7) of the Pulsetach module. For remote I/O, enter the slot number of the master Remote I/O module, the remote I/O drop number, the slot number of the Pulsetach module, and the register number (0-7).

Monitor the counter register and rotate the pulsetach. Verify that the counter register counts in the proper direction. If the pulsetach rotates in the wrong direction, which causes the counter register to count in the wrong direction, the pulsetach input wires must be switched. In a single-ended wiring configuration, swap the A and B inputs. In a differential wiring configuration, swap the A and not A inputs.

Step 12. Turn off power to the rack and pulsetach. Connect the mechanical coupling between the motor and the pulsetach. Turn on power to the system.

3.3 Module Replacement

Use the following procedure to replace a module.

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

Step 1. Turn off power to the rack, pulsetachs, and field wiring.

Step 2. Use a screwdriver to loosen the screws holding the cable assembly's (M/N 57C372) field wiring connector to the module. Remove the cable connector from the module.

Step 3. Loosen the screws that hold the module in the rack. Remove the module from the slot in the rack.

Step 4. Place the module in an anti-static bag, being careful not to touch the connectors on the back of the module. Place the module in the cardboard shipping container.

Step 5. Take the new module out of the anti-static bag it came in. Be careful not to touch the connectors on the back of the module.

Step 6. Insert the module into the desired slot in the rack. Use a screwdriver to secure the module into the slot.

Note that if you are replacing a 57C421-1 module with a 57C421A or later module, you must add the three jumpers shown in figure 3.1 to the terminal strip of the replacement module in order for the module to operate properly. (Jumper from terminal 14 to terminal 17, and jumper from terminal 17 to terminal 20.)

If you were using a 57C421-1 module with 5 V differential inputs, you must write a zero into bit 11 of register 6. Bit 11 is no longer required to select differential inputs. Bit 11 now selects the polarity of the Z pulse.

Step 7. Attach the cable assembly's field wiring connector to the mating half of the connector on the module. Use a screwdriver to secure the connector to the module.

Step 8. Turn on power to the rack, the pulsetach, the motor, and the field wiring.

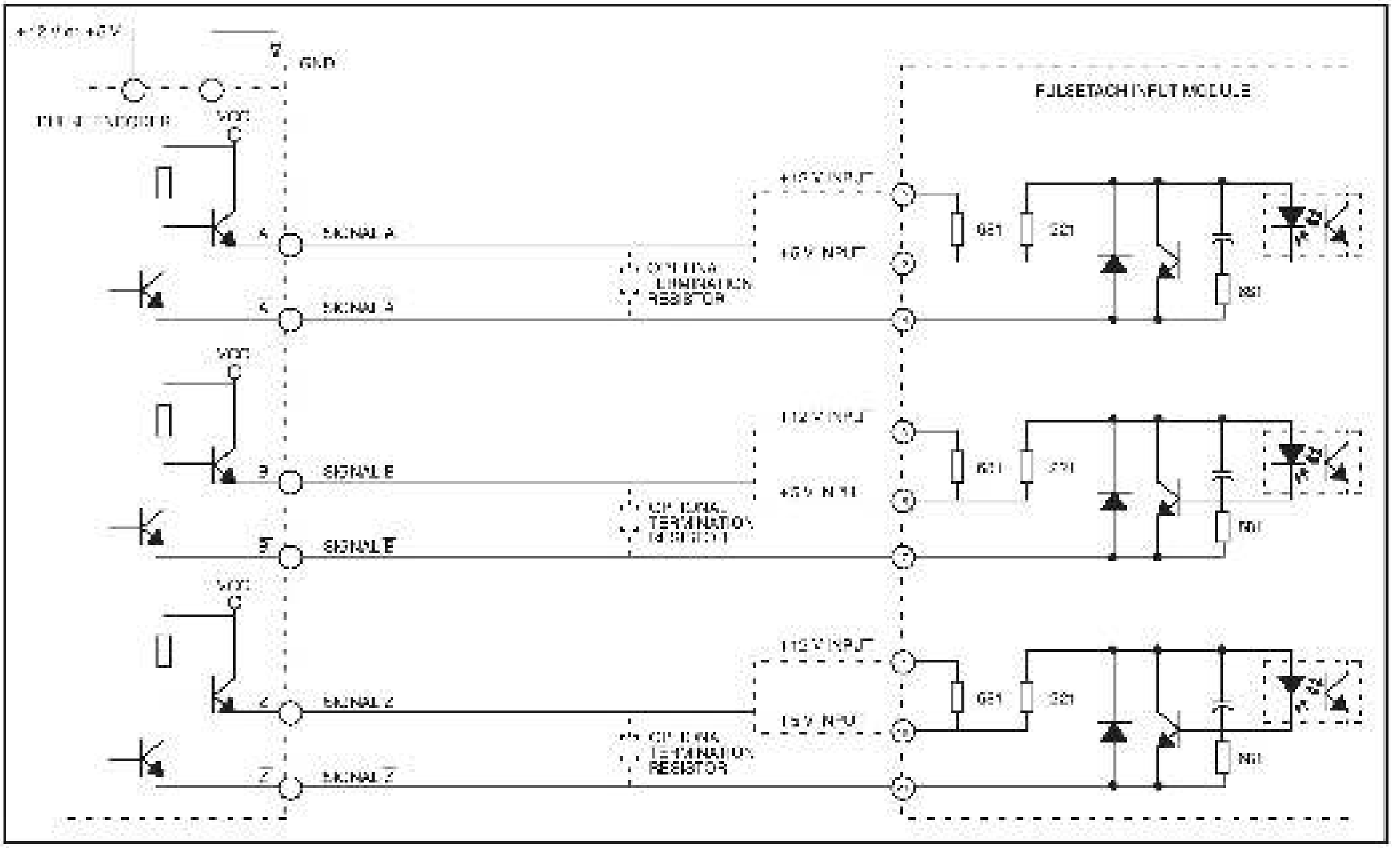


Figure 3.2 - Typical Pulsetach Connections for +5 or 12 VDC Differential Input

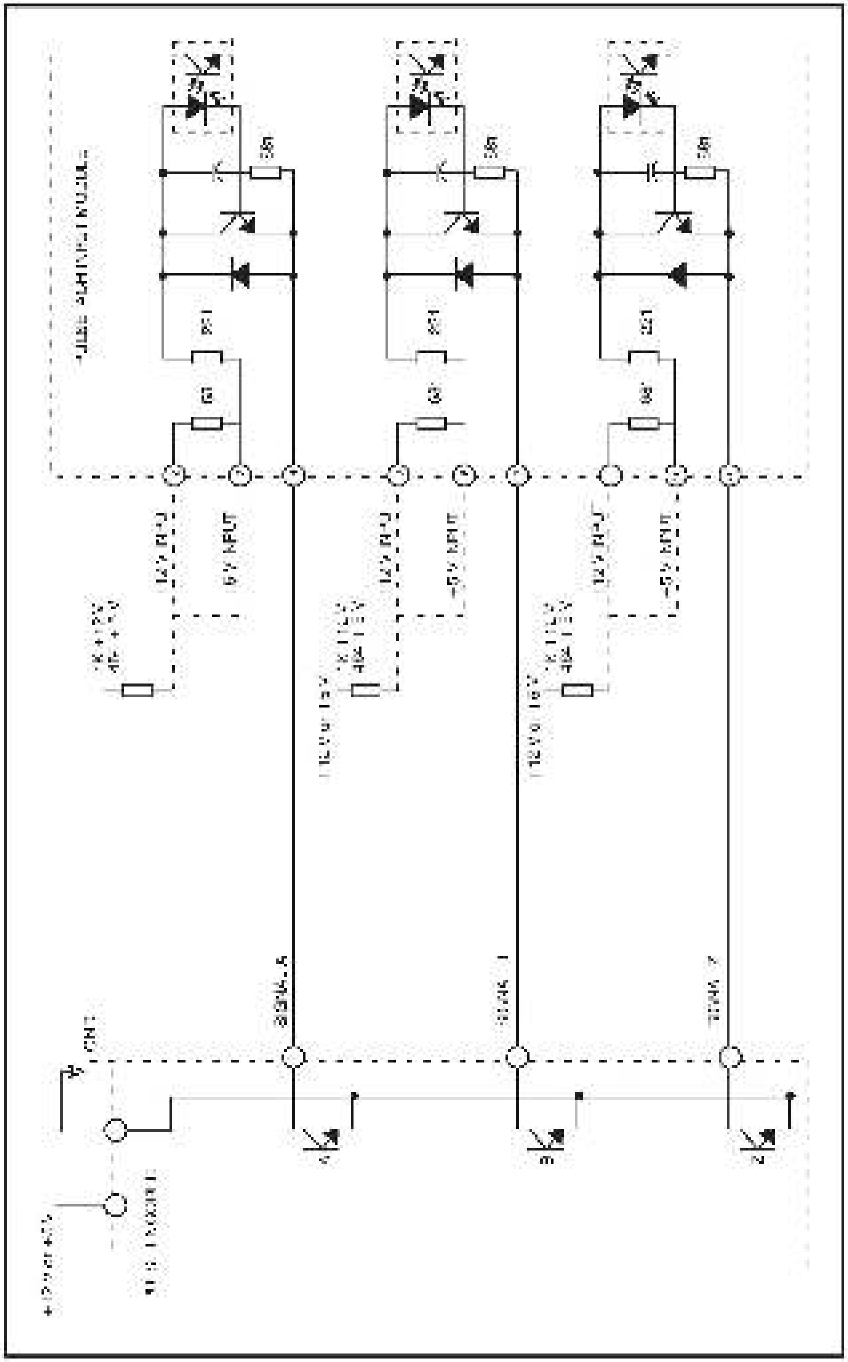


Figure 3-3 - Typical Pulsetach Connections for a Single-Ended Input with External Pull Up Resistor

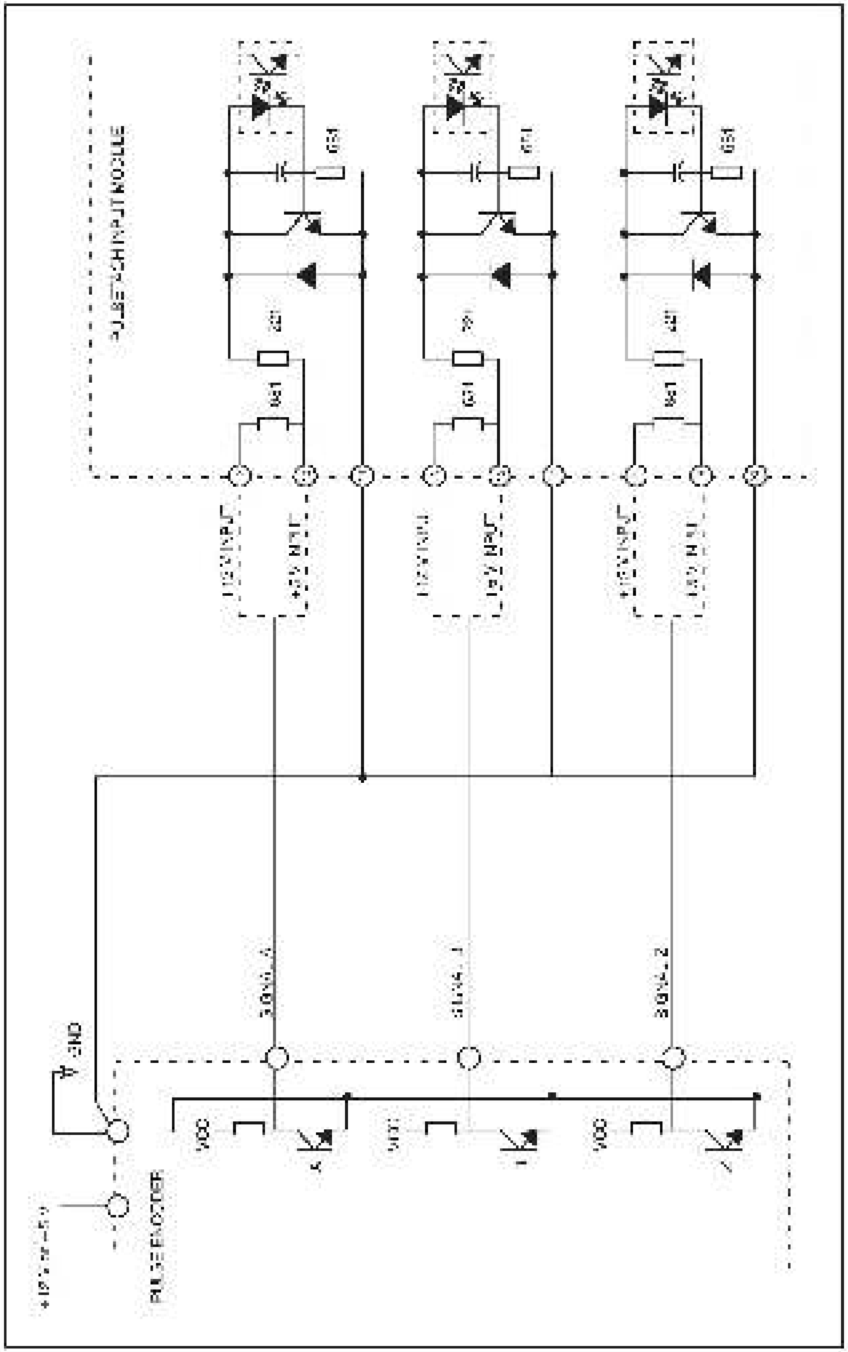


Figure 3.4 - Typical Pulsetach Connections for a Single-Ended Input with No External Pull-Up Resistor

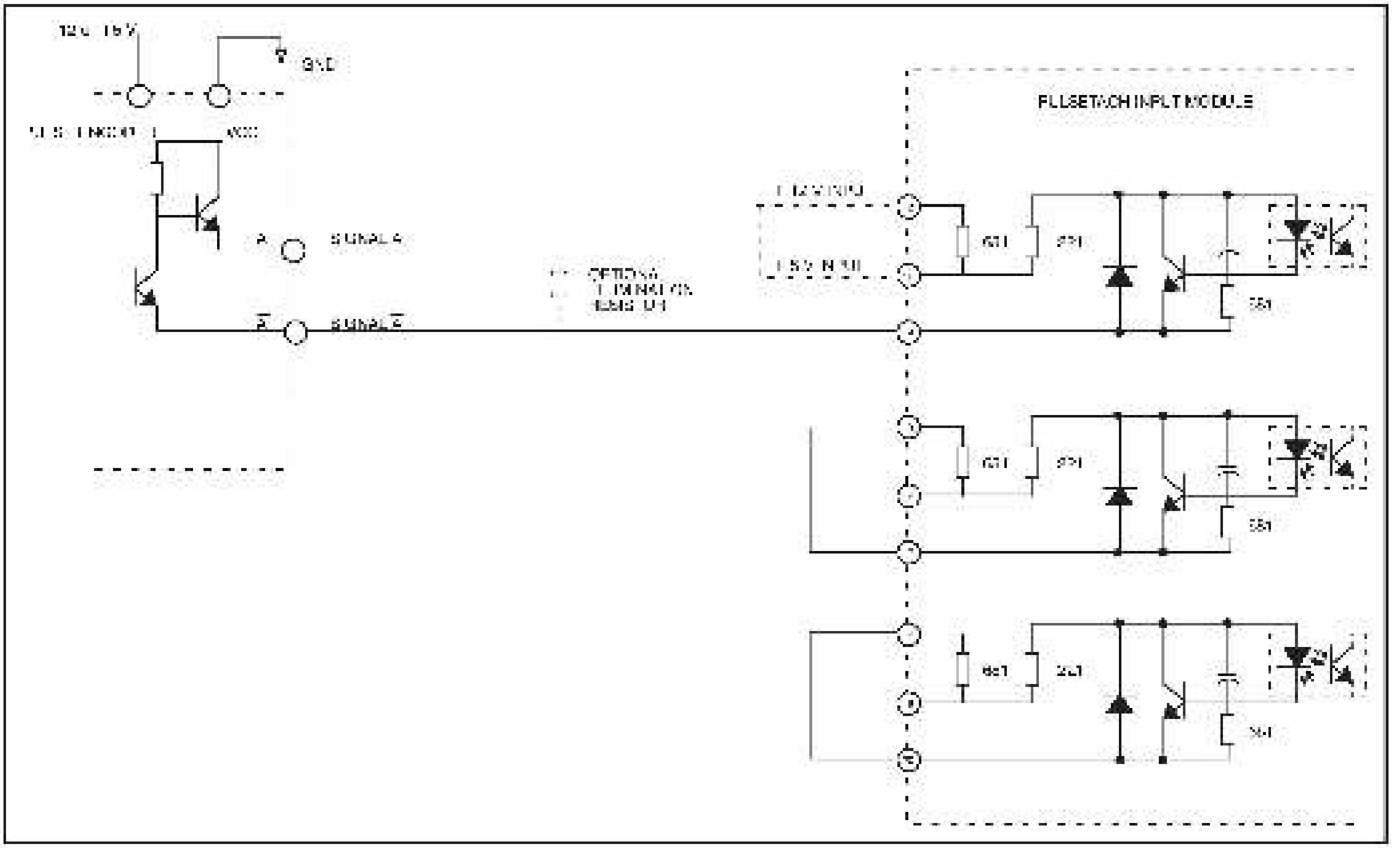


Figure 3.3 - Typical Pulsetach Connections for a Single Channel Input

4.0 PROGRAMMING

This section describes how the data is organized in the module and provides examples of how the module can be accessed by the application program.

4.1 Modes of Operation

The module's counter data can be utilized in one of four ways:

- positioning mode
- speed detection mode
- external latch mode
- timer mode

Each mode of operation is described in the following sections. Note that the speed detection, external latch, and timer modes require the use of hardware interrupts for proper timing. To use the module in these modes, the module must be located in a rack containing a Processor module. Interrupts are not supported in remote racks. See section 4.5 for more information about using interrupts.

4.1.1 Positioning Mode

Positioning mode is the module's default mode of operation at power up. In this mode, the counter value is read and transferred to the latch registers whenever the application program requests counter data. The latch registers hold this value until the next data request is received. The counter is not automatically reset when it is read. See figure 4.1.

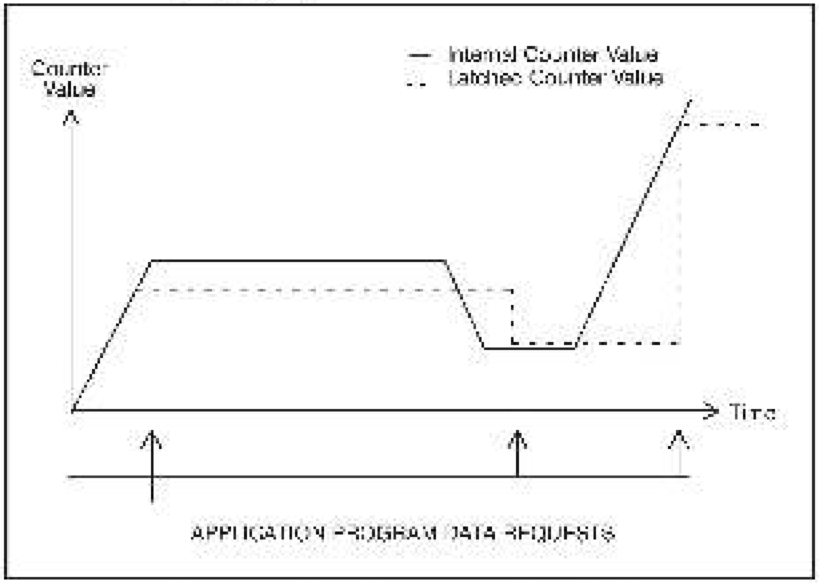


Figure 4.1 - Counter Status During Positioning Mode

4.1.2 Speed Detection Mode

Speed detection mode is enabled by setting the Timer Interrupt Enable bit (register 5, bit 5) to 1. In this mode, the counter value is read and transferred to the latch registers each time the time period defined in the Update Register (register 2) expires. Each time the counter is read, the counter is reset to zero and an interrupt is generated. The latch registers hold the latched counter value until the counter is read again. Refer to figure 4.2.

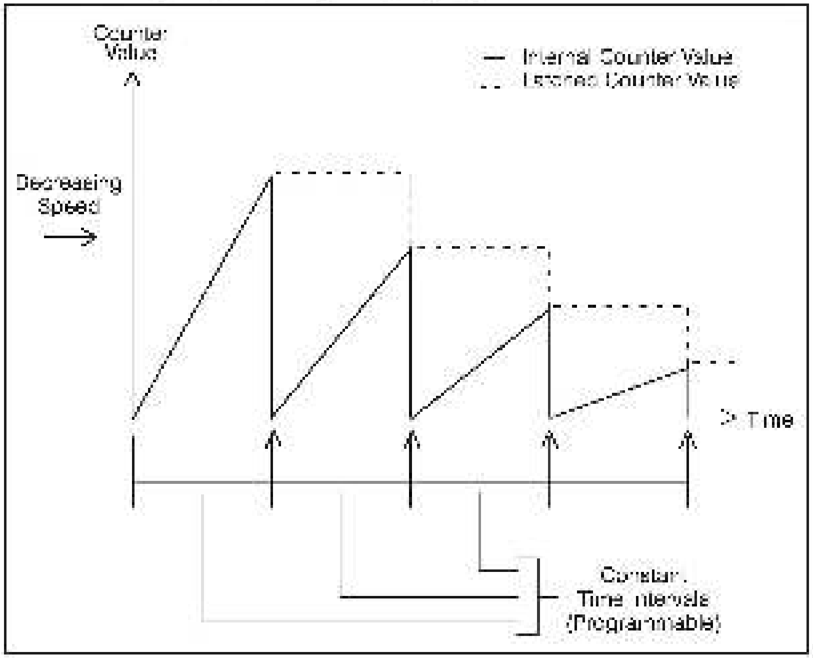


Figure 4.2 - Counter Status During Speed Detection Mode

4.1.3 External Latch Mode

External latch mode is enabled by setting the Enable External Latch Input bit (register 6, bit 5) to one. In this mode, the counter value is read and transferred to the latch registers at the occurrence of an external signal on the input connected to terminal 8. This signal can be from a push button, photo-sensor, or a similar device. Refer to figure 4.3.

The latch registers can be programmed to be either leading edge-triggered or trailing edge-triggered. The status of register 6, bit 14 (External Latch Input Select) defines when the external latch input is considered to be true. The counter is not automatically reset when it is read. To generate an interrupt when the counter is read, the External Latch Interrupt Enable bit (register 5, bit 6) must be set to one.

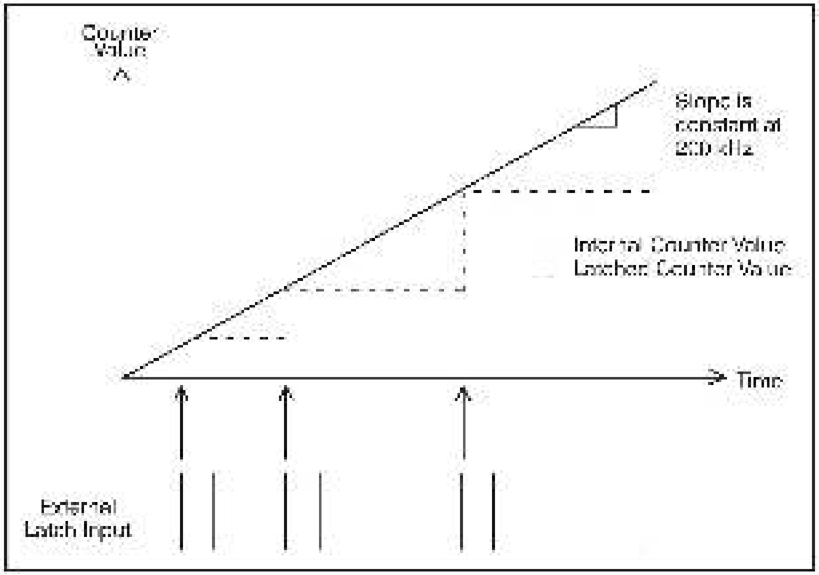


Figure 4.3 - Counter Status During External Latch Mode

4.1.4 Timer Mode

Timer mode is enabled by setting the Timer/Counter Select bit (register 5, bit 13) to one. In this mode, the module's 200 kHz clock serves as a pulse generator which provides constant and uniform pulses to the counter's input. When the input pulses are used in conjunction with the external latch signals, the time interval between two events can be measured. No external wiring is needed to use the 200 kHz clock as a counter input. See figure 4.4.

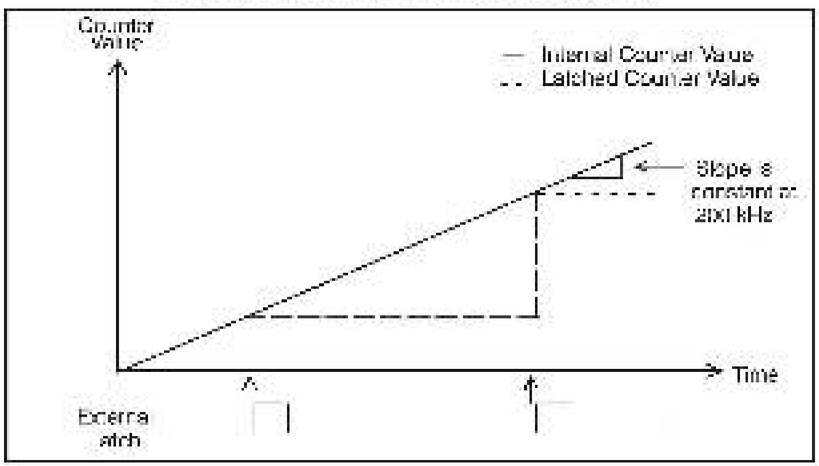


Figure 4.4 - Counter Status During Timer Mode

4.2 Register Organization

The module contains registers for the pulse counter, the comparator, the timer, module status, and module control. The register organization is shown in figure 4.5. The following sections describe each register in detail. A detailed memory map can be found in Appendix C. Note that at power up, all registers are cleared (reset to zero).

Register	Description
0	Counter Data Latch Register
1	Counter Data Latch Register
2	Counter Update Register
3	Comparator Register
4	Comparator Register
5	Interrupt Status and Control Register
6	Mode Definition Register
7	Module Status Register

Figure 4.5 - Pulsetach Module Register Organization

4.2.1 Counter Data Latch Registers (Registers 0-1)

Registers 0 and 1 contain a latched copy of the contents of the module's 24-bit signed counter. Refer to figure 4.6.

The largest value that the counter can hold is +/− 8,388,607. This information can be accessed by referencing registers 0 and 1 as a long integer or as an integer by referencing register 1. Bit 7 of register 0 is the sign bit. Bits 8 to 15 are always set to the state of bit 7. Reference the counter as an integer (register 1) if the counter value will not exceed 32767 between readings. If the counter value will exceed 32767, reference the counter as a long integer. These registers are read only.

Bits

	15 – 8	7 – 0
Register 0	extended sign	MS 8 bits of counter
Register 1	LS 16 bits of counter	

Figure 4.6 - Counter Data Latch Registers (Registers 0-1)

4.2.2 Counter Update Register (Register 2)

Register 2 contains the update period for reading the counter and updating the latch registers. Refer to figure 4.7. The update period is equal to the value in register 2 plus one. Each count in this register is equivalent to 500 microseconds. For example, if you want data latches every 22 msec., assign register 2 a value of 43 ([22 msec/.5 msec] − 1 = 43). The update period may range from 500 microseconds to 32.768 seconds. This register is read/write and is enabled whenever bit 5 of register 5 (Timer Interrupt Enable) is set.

Bits

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register 2	update period															

Figure 4.7 - Counter Update Register (Register 2)

4.2.3 Comparator Registers (Registers 3-4)

Registers 3 and 4 contain a 24-bit signed comparator. Refer to figure 4.8. Bit 7 of register 3 is the sign bit. Bits 8 to 15 are always set to the state of bit 7. The largest value that can be stored in the comparator is $2^{23} - 1 = 8,388,607$.

This information can be accessed by referencing registers 3 and 4 as a long integer or as a simple integer by referencing register 4. If the comparator is referenced as a simple integer, it can contain only positive numbers less than or equal to 32767. This register is read/write.

Bits

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register 3	extended sign								MS 8 bits of comparator							
Register 4	LS 16 bits of comparator															

Figure 4.8 - Comparator Registers (Registers 3-4)

4.2.4 Interrupt Status Control Register (Register 5)

The bits in register 5 are used to enable interrupts and define other module characteristics. This register is read/write. Refer to figure 4.9.

WARNING
BITS 0, 1, 2, 7, AND 15 ARE CONTROLLED BY THE OPERATING SYSTEM AND MUST NOT BE WRITTEN TO BY THE USER. WRITING TO THESE BITS MAY RESULT IN ALL OUTPUTS BEING TURNED OFF AND ALL TASKS IN THE RACK BEING STOPPED. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

Bit: 0
Description: System use only.

Bit: 1
Description: System use only.

Bit: 2
Description: System use only.

Bits: 3 and 4
Description: Counter Clear Control
These bits are used to define the conditions under which the counter is reset to zero.

Bit 3	Bit 4	Condition
0	0	Never clear (1)
0	1	External latch (2)
1	0	Counter equal comparator
1	1	After counter is read (3)

(1) Bit 14 of this register must also be set to one.

(2) The external latch input must also be enabled (register 6, bit 0)

(3) This feature is not available in external latch mode (i.e., register 6, bit 0 = 1).

NOTE. *If the module is set up to clear the counter after the counter is read (bits 3 and 4 = zero), using the Variable Monitor or the I/O monitor function to monitor register 0 and/or 1 will cause the counter register to reset.*

Bit: 5
Description: Timer Interrupt Enable

When this bit is equal to one (i.e., speed detection mode), the counter data is latched, an interrupt is generated, and the counter is reset each time the time period specified in register 2 (Counter Update Register) expires. (Note that if bit 14 is also set to one, the counter will not be cleared after an interrupt.)

If the status of bit 5 is changed to zero after the module has been operating in the speed detection mode, the counter data will be latched when bit 5 makes the transition from one to zero and the counter will not be reset.

Bit: 6
Description: Generate CCLK

When this bit is set to one, the module will provide the CCLK signal to the rack backplane. The CCLK signal can be generated by this module, an Analog Input module (M/N 57C409), a Resolver Input module (M/N 57C411), or a Universal Drive Controller module (B/M O-57552 or O-57652). Only one module per rack may provide the CCLK signal.

If the Pulsetach Input module does not detect the CCLK signal on the backplane, it will use its own internal clock. (Under this condition, the CCLK OK LED on the module faceplate will be off.) Note that if the rack contains more than one module that can generate the CCLK signal, the backplane CCLK signal must be turned on by one of the modules in order to synchronize the modules.

Bit: 7
Description: System use only.

Bit: 8
Description: External Latch Interrupt Enable

When this bit is set to one, an interrupt is generated when the transition specified in register 6, bit 14 (External Latch Input Select) occurs. When an external latch interrupt occurs, you must reset the interrupt by writing a zero to register 7, bit 13 (External Latch Status Reset).

Bit: 9
Description: External Count Stop Interrupt Enable

When this bit is set to one, an interrupt is generated when the condition specified in register 6, bit 12 (Count Stop Input Select) occurs. When an external count stop interrupt occurs, you must reset the interrupt by writing a zero to register 7, bit 14 (External Count Stop Status Reset). Note that the Inhibit Counter bit (register 6, bit 9) is also set internally by the module when an external count stop interrupt occurs and must be reset after each interrupt to enable the module to count again.

Bit: 10
Description: Z Pulse and Origin Interrupt Enable

When this bit is set to one, an interrupt is generated whenever the Z Pulse and origin clear input signals are activated. Note that the Origin/Clear Status bit (register 6, bit 10) must be set to 0. When a Z pulse and origin interrupt occurs, you must reset the interrupt by writing a zero to register 7, bit 15 (External Origin/Clear Status Reset). For additional information, refer to register 6, bit 10.

Bit: 11
Description: Comparator Equal Interrupt Enable

When this bit is set to one, an interrupt is generated when the counter value equals the comparator value as indicated in register 7, bit 4 (Counter Equals Comparator Status). When a comparator equal interrupt occurs, you must reset the interrupt by writing a zero to register 7, bit 12.

You must set the comparator value before you enable the comparator equal interrupt (register 5, bit 11).

Note that if you do not set the comparator value before you enable the interrupt at power up (when all internal registers are equal to zero), a comparator equal interrupt will be issued and error "1b" will be displayed on the faceplate of the Processor.

Bit: 12
Description: Pulse Multiplier

This bit specifies how the incoming pulses from a quadrature pulse tach are multiplied. If the bit is set to one, the incoming frequency is multiplied by four. If the bit is set to zero, the incoming pulses are multiplied by two.

If a single-channel pulsetach is connected to the module, this bit should be set to zero. Incoming pulses from a single-channel pulsetach are not multiplied.

Bit: 13
Description: Timer/Counter Select

When this bit is set to one, the module functions as a timer using its internal 200 kHz clock (no external cabling from the pulsetach is required). If the bit is set to zero, the module functions as counter based on pulsetach inputs.

Bit: 14
Description: Counter Clear Inhibit

When this bit is set to one, the counter will not be cleared after an interrupt while the module is operating in space detection mode. Note that bits 3 and 4 of this register must both be set to zero to activate this feature. When this bit is set to zero, the module will operate in space detection mode as described in section 4.1.2, i.e. the counter will be cleared after each interrupt.

Bit: 15
Description: System use only.

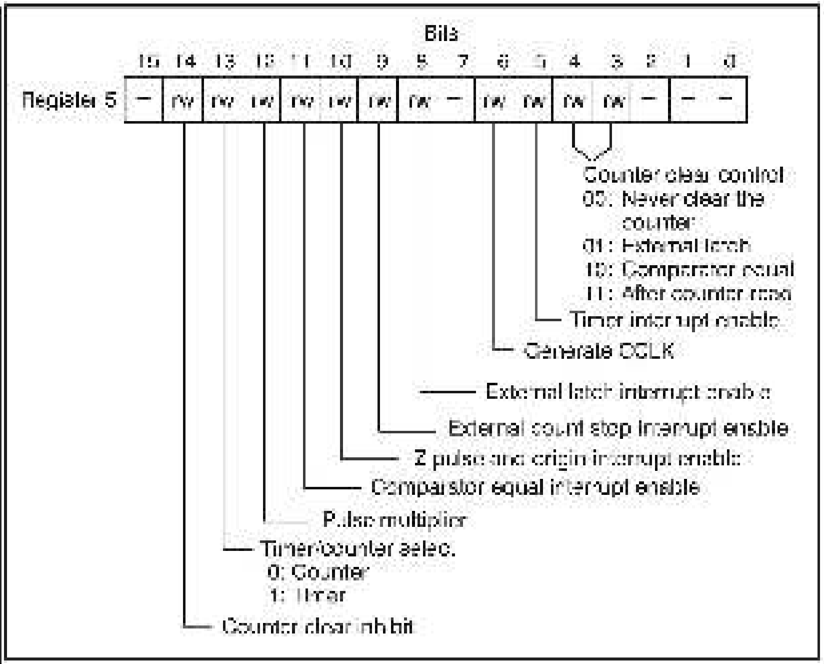


Figure 4.8 - Interrupt Status Control Register (Register 5)

4.2.5 Mode Definition Register (Register 6)

Register 6 is a control register used to define the module's operating mode. Refer to figure 4.10. This register is read/write.

WARNING

BITS 4, 5, 6, AND 7 ARE CONTROLLED BY THE OPERATING SYSTEM AND MUST NOT BE WRITTEN TO BY THE USER. WRITING TO THESE BITS MAY RESULT IN ALL OUT PUTS BEING TURNED OFF AND ALL TASKS IN THE RACK BEING STOPPED. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

Bit: 0
Description: External Latch Enable

Bit 0 is used to enable the external latch input. When this bit is set to one and the external latch input makes the transition specified by register 6, bit 14, the value in the counter at that time will be stored in registers 0 and 1.

Bit: 1
Description: External Count Stop Enable

Bit 1 is used to enable the external count stop input. When this bit is set to one and the external count stop input is equal to the condition specified by register 6, bit 12, the counter will stop counting.

Bits: 2 and 3
Description: Count Reverse and Count Forward

Bits 2 and 3 are used to define the counter direction when the clear/origin input is used to initialize the absolute position of an external device. When the clear/origin input is used for this purpose, you must define whether the counter should be counting forward or backward when the marker pulse resets the counter. Refer to register 6, bit 10 for more information.

Set bit 2 to one if the counter should be counting in the reverse direction. Set bit 3 to one if the counter should be counting in the forward direction.

Bit: 4
Description: System use only.

Bit: 5
Description: System use only.

Bit: 6
Description: System use only.

Bit: 7
Description: System use only.

Bit: 8
Description: Type of Pulsetach

Bit 8 defines the type of pulsetach connected to the module. This bit should be set to zero if a quadrature (A and B) pulsetach is connected. A quadrature pulsetach is required to count forward and reverse pulses.

This bit should be set to one if a single-input pulsetach is connected. A single-input pulsetach may be connected to either the A or B inputs. Note that with a single-input pulsetach, the counter will always count up; however, the FORWARD and REVERSE LEDs will flicker.

Bit: 9
Description: Inhibit Counter

Bit 9 is used to stop the counter from counting. When this bit is set to one, the counter will not count incoming pulses. Note that this bit is also set internally by the module when an external count stop interrupt occurs (see register 5, bit 13). This bit must be reset after an external count interrupt is generated to enable the module to count again.

Bit: 10
Description: Origin/Clear Select

Bit 10 is used to specify the action that occurs when the origin/clear input is true. If bit 10 is equal to one, the origin/clear input will reset the counter whenever it is in the same state as the value specified by register 6, bit 13. If bit 10 is equal to zero, the counter will be reset when the origin/clear input is in the same state as the value specified by register 6, bit 13, the counter is counting in the direction specified by register 6, bits 2 or 3, and the marker (Z) pulse occurs. The latter is typically used to initialize the absolute position of a machine.

Bit: 11
Description: Z Pulse Polarity

Bit 11 is used to specify the polarity of the Z pulse. If bit 11 is zero (default), the Z pulse's logic is positive. If bit 11 is one, the pulse's logic is negative.

Bit: 12
Description: Count Stop Input Select

Bit 12 is used to specify when the count stop input is considered to be true. If this bit is zero, a high input signal (+V) will be considered to be true. If this bit is one, a low input signal (0V) will be considered to be true.

Bit: 13
Description: Origin/Clear Input Select

Bit 13 is used to specify when the origin/clear input is considered to be true. If this bit is zero, a high input signal (+V) will be considered to be true. If this bit is one, a low input signal (0V) will be considered to be true.

Bit: 14
Description: External Latch Input Select

Bit 11 is used to specify when the external latch input is considered to be true. If this bit is zero, a high input signal (+V) will be considered to be true. If this bit is one, a low input signal (0V) will be considered to be true.

Bit: 15
Description: Reset Counter

Bit 15 is used to reset the 24 bit counter under software control. The counter is reset to zero whenever this bit is set.

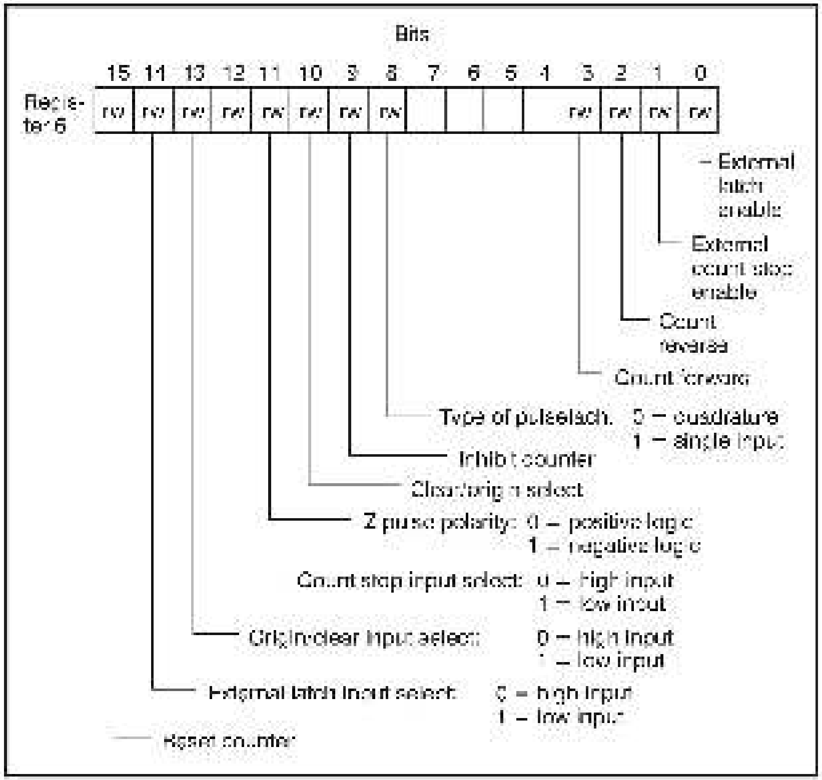


Figure 4.10 - Mode Definition Register (Register 6)

4.2.6 Module Status Register (Register 7)

Register 7 contains module status and interrupt reset control bits. Refer to figure 4.11.

Bit: 0
Description: Carry Status

This bit is set whenever a carry occurs from bit 7 of register 0 (i.e., the value of the counter has rolled over to zero in the positive direction). This bit is reset by writing a zero to register 7, bit 10.

Bit: 1
Description: Borrow Status

This bit is set whenever a borrow occurs from bit 7 of register 0 (i.e., the value of the counter has rolled over to zero in the negative direction). This bit is reset by writing a zero to register 7, bit 11.

Bit: 2
Description: Counter Greater Than Comparator

Bit 2 is set whenever the counter value (registers 0 and 1) is greater than the comparator value (registers 3 and 4).

Bit: 3
Description: Counter Less Than Comparator

Bit 3 is set whenever the counter value (registers 0 and 1) is less than the comparator value (registers 3 and 4).

Bit: 4
Description: Counter Equals Comparator

Bit 4 is set whenever the counter value (registers 0 and 1) is equal to the comparator value (registers 3 and 4). This bit can be reset by writing a zero to register 7, bit 12.

Bit: 5
Description: External Latch Input Status

Bit 5 contains the status of the external latch. This bit is set and latched whenever the external latch makes the transition specified by register 6, bit 14. Note that this bit will contain status data only if the External Latch Enable bit (register 6, bit 0) is set. This bit is reset by writing a zero to register 7, bit 13.

Bit: 6
Description: External Count Stop Internal Status

Bit 6 is set and latched whenever the external count stop input is equal to one. Note that this bit will contain status information only if the External Count Stop Enable bit (register 6, bit 1) is set. This bit is reset by writing a zero to register 7, bit 14.

Bit: 7
Description: Origin/Clear Input Status

Bit 7 contains the status of the external origin/clear input. This bit is set whenever the external origin/clear input is true. This bit can be reset by writing a zero to register 7, bit 15.

Bit: 8
Description: CCLK Off

Bit 8 indicates that the CCLK signal on the backplane is off. This signal can be generated by this module (register 5, bit 6), an Analog Input module (M/N 57C409), a Resolver Input module (M/N 57C411) or a Universal Drive Controller module (B/M O-57552 or O-57652). Only one module per rack may control the CCLK signal.

If the module does not detect the CCLK signal on the backplane, it will use its own internal clock. (Under this condition, the CCLK OK LED on the module faceplate will be off.) However, if the rack contains more than one module that can generate the CCLK signal, the backplane CCLK signal must be turned on in order to synchronize the modules.

Bit: 9
Description: Pulse Input Direction

Bit 9 contains the direction of the last count read in by the counter. The counter's direction can be either forward (0) or reverse (1).

Bit: 10
Description: Carry Status Reset

Bit 10 has a default value of one. Writing a zero to this bit will reset the Carry Status bit (register 7, bit 0), but subsequent reads will return a value of one.

Bit: 11
Description: Borrow Status Reset

Bit 11 has a default value of one. Writing a zero to this bit will reset the Borrow Status bit (register 7, bit 1), but subsequent reads will return a value of one.

Bit: 12
Description: Counter Equals Comparator Status Reset

Bit 12 has a default value of one. Writing a zero to this bit will reset the Counter Equals Comparator Status bit (register 7, bit 4) and the comparator equal interrupt (see register 5, bit 11), but subsequent reads will return a value of one.

Bit: 13
Description: External Latch Status Reset

Bit 13 has a default value of one. Writing a zero to this bit will reset the External Latch Status bit (register 7, bit 5) and the external latch interrupt (see register 5, bit 8), but subsequent reads will return a value of one.

Bit: 14
Description: External Count Stop Status Reset

Bit 14 has a default value of one. Writing a zero to this bit will reset the External Count Stop Status bit (register 7, bit 6) and the external count stop interrupt (see register 5, bit 9), but subsequent reads will return a value of one. Note that the Inhibit Counter bit (register 6, bit 9) also must be reset after an external count stop interrupt is generated.

Bit: 15
Description: External Origin/Clear Status Reset

Bit 15 has a default value of one. Writing a zero to this bit will reset the External Origin/Clear Status bit (register 7, bit 7) and the Z pulse and origin interrupt (see register 5, bit 10), but subsequent reads will return a value of one.

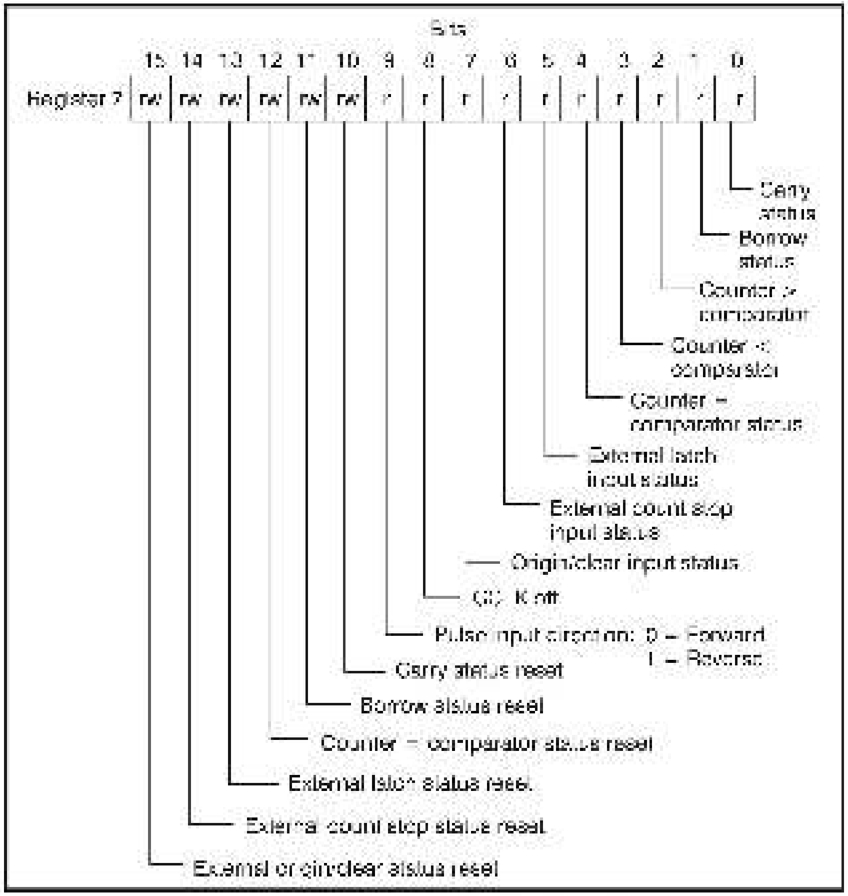


Figure 4.11 – Module Status Register (Register 7)

4.3 Variable Configuration

Before an application task can be written, you need to configure, or define, system-wide data such as the registers on the Pulsetach Input module as variables. These are variables that must be globally accessible to more than one task in the rack.

For DCS 5000 and AutoMax Version 2.1 and earlier, you define system-wide variables by writing a Configuration task. For AutoMax Version 3.0 and later, you define system-wide variables using the AutoMax Programming Executive configuration forms. After the variables are defined, you can generate the configuration file automatically.

If you are using AutoMax Version 2.1 or earlier, refer to Appendix F for examples that show how to define variables in the configuration task. If you are using AutoMax Version 3.0 or later, see the AutoMax Programming Executive for information about configuring variables.

4.4 Applying the Module

In order for hardware to be referenced by application software, it is first necessary to assign symbolic names to the hardware. This is accomplished in the configuration, as described in the section 4.3.

Each task that wishes to reference the symbolic names assigned to the Pulsetach Input module may do so by declaring those names COMMON. Once this has been done, any reference to those symbolic names within the task will reference the bit or register defined in the configurations.

The frequency with which tasks read input variables and write output variables depends on the programming language being used. Ladder Logic and Control Block tasks read all their inputs once at the beginning of each scan, regardless of how often the inputs are referenced in the task and write all output variables at the end of the scan. BASIC statements (even within Control Block tasks) read an input each time it is referenced and write an output each time it is referenced.

4.4.1 Speed Mode Example

The following is an example of a Control Block task that handles interrupts from the module. All variables declared as COMMON are assumed to be previously been defined during configuration.

In the example below, the module is continuously accumulating pulses from a pulsetach. The module is set up to capture the value of the counter at a periodic rate and then generate an interrupt. This task could be used to read in counts and then accumulate them in software. It could also be used to generate a velocity signal (dx/dt).

```
1       !
2       ! SPEED MODE EXAMPLE
3       ! TASK NAME: P2 SPEED
4       ! PRIORITY: [illegible]
5       !
1000    COMMON TIME1%                \Counter read time
1005    COMMON ISCR%                 \Interrupt and status register
1010    COMMON COUNT%                \Counter data
1015    COMMON MULT@                 \Pulse multiplier
1020    COMMON CCLK_EN@              \CCLK enable
1025    COMMON INT_EN@               \Timer interrupt enable
1900    LOCAL COUNTER_VALUE%         \Counter value [illegible]
2100    MULT@ = ON                   \Pulse multiplier used
2200    TIME1% = 50                  \Read every 50 milliseconds
2201    !
2202    ! [illegible]
2203    ! [illegible]
2204    !
3000    !
3001    ! The following statement connects the name COUNTER_EVENT
3002    ! to the [illegible] defined in ISCR%. The event name should
3003    ! be as close [illegible] as possible. The [illegible] timeout has
3004    ! been set to 12 Processor clock ticks [illegible]. If the time
3005    ! between interrupts exceeds this value, [illegible]
3006    ! will be declared and the system will be stopped. [illegible]
3007    !
3008    !
3009    !
3010    EVENT NAME=COUNTER_EVENT, INTERRUPT_STATUS=ISCR%,
        TIMEOUT=12
4000    !
4001    ! The following statement disables the 'constant clock' from
4002    ! this module. If there is more than one [illegible] task in this
4003    ! chassis, the task that enables the constant clock should
4004    ! always be the lowest priority task.
4005    !
4100    CCLK_EN@=FALSE               \Counter will be read on interrupt.
```

Speed Mode Example (Continued)

```
4105    INT_BSY = TRUE                ! Enable timer interrupt
4110    DSCX_LNGS = TRUE              ! Enable [illegible] per task
5000    !
5001    ! Place additional initialization statements here.
5002    !
6000    !
6001    ! The next statement synchronizes the task to the external
6002    ! event via the interrupt. Task execution will be suspended
6003    ! until the interrupt occurs. When the interrupt occurs, if
6004    ! this task is the highest priority task waiting to execute
6005    ! it will become active. If not, the highest priority
6006    ! task will become suspended until all higher priority
6007    ! tasks have finished executing at which point it will then
6008    ! become active.
6010    CALL START_CRIT( [illegible] = COUNTER_EVENT )
7000    !
7001    !
7002    !
7003    ! This example assumes that 32767 or fewer counts will be
7004    ! received in the 50 msec. scan because statement 10000 only
7005    ! references the least significant register (1) on the
7006    ! register.
10000   CALL PULSE_INPUT( INPUT_COUNTS%, MULTIPLIER = [illegible],     &
        OUTPUT = COUNTER_VALUES )
32767   END
```

4.4.2 Positioning Mode Example

The following is an example of a BASIC task that handles interrupts from the module. All variables declared as COMMON are assumed to have previously been defined during configuration.

In the example below, the module generates an interrupt every time it accumulates the number of pulses indicated by the compare register. Additional tasks or additional code in this task must be written to take specific action.

```
1       !
2       ! POSITIONING MODE EXAMPLE
3       ! TASK NAME: PS COMP
4       ! PRIORITY: 10
5       !
1000    COMMON COUNT%                 ! Counter data
1005    COMMON [illegible]_COMP%      ! Comparator data
1010    COMMON ISO_F%                 ! Interrupt status and control
1015    COMMON RESET%                 ! Counter reset
1020    COMMON CNTR_EQ_RST%           ! Comparator equal counter reset
1025    COMMON EQL_INT%               ! Comparator equal int. enable
1030    COMMON CLR_MOD1%              ! Counter clear condition
1035    COMMON CLR_MOD2%              ! Counter clear condition
1040    COMMON CNTR_EN%               !
1200    LOCAL LIMIT_SW_OFFSET()       ! Table of limit switch positions
1205    LOCAL INDEX%                  ! Index of limit switch interrupt
1210    LOCAL I%                      ! Index of next limit switch interrupt
1215    LOCAL INT_SERVICE0%           ! Counter for interrupt service 0
1220    LOCAL INT_SERVICE1%           ! Counter for interrupt service 1
1225    LOCAL INT_SERVICE2%           ! Counter for interrupt service 2
1230    LOCAL INT_SERVICE3%           ! Counter for interrupt service 3
3000    !
3001    ! The following statement connects the name COUNTER_EVENT
3002    ! to the interrupt defined in ISO_F%. The event name should
3003    ! be as descriptive as possible. The watchdog timeout has
3004    ! been disabled because the event is not periodic.
3005    !
3006    !
3007    !
3010    EVENT NAME=COUNTER_EVENT, INTERRUPT_STATUS=ISO_F%,     &
        TIMEOUT=DISABLED
4000    !
4001    ! The following statements initialize the counter and set
4002    ! up the interrupt control. "Constant clock" is enabled on
4003    ! this module. If there is more than one interrupt task in
4004    ! a chassis, the task that enables "constant clock" should
```

Positioning Mode Example (Continued)

```
4007    ! always be the lowest priority task.
4008    !
4010    FOR S% = 0 TO 3 \ READ LIMIT_SWITCH%(S%) \ NEXT S%
4015    RESETC1% = TRUE \ RESETC1% = FALSE       ! Zero counter
4020    S% = 0                                   ! Initialize limit switch index
4025    LOW_COMP% = LIMIT_SWITCH%(S%)            ! Set comparator to 1st value
4030    CNTR_EQ_RST% = FALSE                     ! Initialize Comp. = status
4035    COU_INT% = ON                            ! Enable comp. = interrupt
4040    OUT_MODE1% = OFF                         ! Reset on equal
4045    OUT_MODE2% = ON                          ! Reset on equal
4050    COUNTEN% = ON                            ! Enable counter clock
5985    !
5986    ! Place additional initialization software here.
5990    !
5991    !
5992    ! The next statement synchronizes the task to the external
5993    ! event via the interrupt. Task execution will be suspended
5994    ! until the interrupt occurs. When the interrupt occurs, if
5995    ! this task is the highest priority task waiting to
5996    ! execute, it will become active. If it is not the highest
5997    ! priority task, it will remain suspended until all higher
5998    ! priority tasks have executed at which point it will then
5999    ! become active.
6000    !
6010    WAIT ON COUNTER_EVENT
6997    !
6998    ! The following statements perform the interrupt service
6999    ! routine.
7000    !
7010    INDEX% = S%                                         ! Save this index value
7015    S% = S% + 1 \ IF S% > 3 THEN S% = 0                 ! Step to next point
7020    LOW_COMP% = LIMIT_SWITCH%(S%)                       ! Set up next value
7025    CNTR_EQ_RST% = FALSE                                ! Reset interrupt
7030    ON INDEX% GOSUB 8000, 8200, 8400, 8600 ! Execute routine
7035    GOTO 6010
7997    !
7998    ! Interrupt service routine for limit switch value 0
7999    !
8000    INT_SERVICE0% = INT_SERVICE0% + 1
8100    RETURN
8197    !
8198    ! Interrupt service routine for limit switch value 1
8199    !
8200    INT_SERVICE1% = INT_SERVICE1% + 1
8300    RETURN
8397    !
8398    ! Interrupt service routine for limit switch value 2
8399    !
8400    INT_SERVICE2% = INT_SERVICE2% + 1
8500    RETURN
8597    !
8598    ! Interrupt service routine for limit switch value 3
8599    !
8600    INT_SERVICE3% = INT_SERVICE3% + 1
8700    RETURN
9000    DATA 1000                       ! Limit switch position 0
9005    DATA 500                        ! Limit switch position 1
9010    DATA 2700                       ! Limit switch position 2
9020    DATA 3000                       ! Limit switch position 3
32767   END
```

4.4.3 Timer and Latch Mode Example

The following is an example of a BASIC task that handles interrupts from the module. All variables declared as COMMON are assumed to have previously been defined during configuration.

In the example below, the task counts pulses generated by the internal 200 kHz clock. Each time an external latch input signal is received, the task latches the counter data and then clears the counter.

```
1       !
2       ! TIMER AND EXTERNAL LATCH MODE EXAMPLE
3       ! TASK NAME: TIM_LATCH
4       ! PRIORITY: 10
5       !
1000    COMMON COUNT%                 ! Counter data
1005    COMMON ISCR%                  ! Interrupt status & control
1010    COMMON RESET@                 ! Counter reset
1015    COMMON EXT_LATCH_EN@          ! External latch enable
1020    COMMON LATCH_POLARITY@        ! External latch polarity
1025    COMMON LATCH_RESET@           ! External latch interrupt reset
1030    COMMON TIM_MODE@
1035    COMMON LAT_INT@
1040    COMMON CCLK_EN@
1045    COMMON CLR_MD1@
1050    COMMON CLR_MD2@
1060    LOCAL DELTA%                  ! Time between external events
2000    !
2001    ! The following statement defines the name COUNTER_EVENT
2002    ! to the interrupt defined in ISCR%. The event name should
2003    ! be as descriptive as possible. The watchdog timeout has
2004    ! been disabled because the event is not periodic.
2005    !
2006    !
2007    !
2010    EVENT NAME=COUNTER_EVENT, INTERRUPT STATUS=ISCR%,         &
        TIMEOUT=DISABLED
4000    !
4001    ! The following statements initialize the counter and set
4002    ! up the interrupt control. Constant clock is enabled on
4003    ! this module. If there is more than one interrupt task in
4004    ! a chassis, the task that enables "constant clock" should
4005    ! always be the lowest priority task.
4006    !
4010    RESET@ = TRUE \ RESET@ = FALSE ! Zero counter
4015    EXT_LATCH_EN@ = TRUE          ! Enable the external latch
4020    LATCH_POLARITY@ = FALSE       ! Latch input is high true
4025    TIM_MODE@ = ON                ! Timer mode
4030    LAT_INT@ = ON                 ! External latch int. enable
4035    CLR_MD1@ = OFF                ! Clear counter on latch
4040    CLR_MD2@ = ON                 ! External latch
4045    CCLK_EN@ = ON                 ! Enable CCLK
5098    !
5099    ! Place additional initialization code here
5000    !
5001    !
5002    ! The next statement synchronizes the task to the occurrence
5003    ! of the external latch via the interrupt. Task execution
5004    ! will be suspended until the interrupt occurs. When the
5005    ! interrupt occurs, if this task is the highest priority
5006    ! task waiting to execute, it will become active. If it is
5007    ! not the highest priority task, it will remain suspended
5008    ! until all other higher priority tasks have been executed,
5009    ! at which point it will then become active.
5010    WAIT ON COUNTER_EVENT
6000    !
6005    ! The following statements perform the interrupt service
6006    ! routine.
7000    DELTA% = COUNT%               ! Read time between external events
7005    LATCH_RESET@ = FALSE          ! Reset the interrupt
7010    GOTO 5000                     ! Wait for the next event
32767   END
```

4.5 Using Interrupts in Application Tasks

The input module can be programmed to generate interrupts on the basis of a time interval, an external latch input, an external count stop input, a marker pulse and origin input, or a comparator equal condition. Time interval-based interrupts cannot be used with any other interrupts.

Interrupts are used to synchronize software tasks to the occurrence of a hardware event. This module provides the ability to synchronize events beginning at 1.2 msec and increasing in increments of 500 μsec depending on the priority level of the task receiving the interrupt.

In order to use interrupts on the Pulsetach Input module, the module must be in a rack containing a Processor module. **Interrupts cannot be used with Pulsetach Input modules located in remote racks.**

You must first assign symbolic names to the interrupt control registers on the module during configuration. Only one task may act as the receiver for the interrupts generated by a Pulsetach Input module. That task should declare the symbolic names assigned to the interrupt control registers as COMMON. The interrupt Status and Control register (register 5) must be referenced in the hardware EVENT statement in the task receiving the interrupt. The examples in sections 4.4.1, 4.4.2, and 4.4.3 illustrate various uses of the interrupt feature. Note that the receiving task uses either the SCAN_LOOP (Control Block) statement or the WAIT ON (BASIC) statement to actually receive the signal. (Refer to the Control Block and BASIC Language instruction manuals.)

All interrupts are internally double-buffered. This helps to eliminate spurious interrupts, which could cause system errors when the module is operated in an electrically noisy environment. The application task must provide a tightly-coupled software handshake with the external asynchronous interrupt inputs coming into the module. Register 7, the Module Status register, is used in this handshaking. The status bits in register 7 are set to one when an interrupt input is received and remain set until the application program clears them. As long as the status bits in the register are equal to one, the module will not recognize additional interrupt inputs as they occur.

4.6 Restrictions

This section describes limitations and restrictions on the use of the Pulsetach Input module.

4.6.1 Writing Data to Registers

The module's counter registers (registers 0 and 1) are read-only. Attempts to write to them will cause a Bus Error (STOP ALL system error indicated by a "31" on the faceplate of the Processor).

Some examples of programs that write to the module and must not be used are:

- Referencing an input on the left side of an equal sign in a Control Block or BASIC task.
- Referencing an input as an output in a Control Block function.

4.6.2 Use In Remote I/O Racks

32-bit register references should be used with caution when this module is placed in a remote rack. The remote I/O system does not always transfer registers greater than 16 bits as a unit. As a result, it is possible for an application task to read the least significant 16 bits of a new value and the most significant 16 bits of the previous value.

Interrupts cannot be used with Pulsetach Input modules in remote racks.

WARNING
IF YOU USE DOUBLE INTEGER VARIABLES IN THIS INSTANCE, YOU MUST IMPLEMENT A SOFTWARE HANDSHAKE BETWEEN THE TRANSMITTER AND THE RECEIVER TO ENSURE THAT BOTH THE LEAST SIGNIFICANT AND MOST SIGNIFICANT 16 BITS HAVE BEEN TRANSMITTED BEFORE THEY ARE READ BY THE RECEIVING APPLICATION PROGRAM. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY OR DAMAGE TO EQUIPMENT.

4.6.3 Pulsetach Feedback Precautions

WARNING
LOSS OF, OR AN OTHERWISE IMPROPER, PULSETACH SIGNAL CAN RESULT IN UNCONTROLLED MOTOR SPEED. PROVIDE AN INDEPENDENT METHOD OF SHUTTING DOWN THE EQUIPMENT IF THIS SHOULD OCCUR. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY AND IN DAMAGE TO, OR DESTRUCTION OF, THE EQUIPMENT.

When this module is used with a pulsetach in a drive control system, you must incorporate an independent method of determining that this module is actually reading proper motor RPM. It is necessary to determine this because the Pulsetach Input module is not capable of detecting a loss of feedback in all situations, such as, for example, when a coupling breaks between the motor and the pulsetach.

WARNING
THE USER IS RESPONSIBLE FOR ENSURING THAT DRIVEN MACHINERY, ALL DRIVE TRAIN MECHANISMS, AND THE WORKPIECE IN THE MACHINE ARE CAPABLE OF SAFE OPERATION AT MAXIMUM SPEEDS. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY AND IN DAMAGE TO, OR DESTRUCTION OF, THE EQUIPMENT.

You must also determine the maximum safe operating speed for the motor, connected machinery, and material being processed. Then either verify that the system is incapable of reaching that speed, or else incorporate the necessary hardware/software to ensure that this limit will never be exceeded.

5.0 DIAGNOSTICS AND TROUBLESHOOTING

This section explains how to troubleshoot the module and field connections. If the problem cannot be corrected by following the instructions below, the module is not user-serviceable.

DANGER

ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL AND OTHER APPLICABLE MANUALS IN THEIR ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

5.1 Incorrect Data

The data is either always off, always on, or different than expected. When this happens the module is malfunctioning, is in the wrong slot, or there is a programming error. It is also possible that the input wiring is incorrectly connected.

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

Use the following procedure to isolate the error:

Step 1 Verify that the Pulsetach input module is in the correct slot as defined in the configuration.

Step 2 Verify that the pulsetach inputs are wired correctly.

Confirm that all terminal strip connections are tight.

Step 3 Verify that the pulse input circuitry on the module is working correctly. Be sure the CCLK signal is present on the rack's backplane if the rack contains more than one module that can generate the CCLK signal. Check the CCLK LED on the module's faceplate. The LED is on when the CCLK signal is present on the backplane.

Remove power from the system. Disconnect the mechanical coupling between the motor and the pulsetach. Apply power to the rack and the pulsetach.

Use an oscilloscope on the terminal strip to verify that the voltages coming from the pulsetach are at the proper level.

Rotate the pulsetach in the forward direction. The FORWARD LED on the module's faceplate should turn on. Rotate the pulsetach in the reverse direction. The REVERSE LED should turn on. If the LEDs do not turn on, the module's pulse input circuitry is not working correctly.

If the pulsetach rotates in the wrong direction, the pulsetach input wires must be switched. In a single-ended wiring configuration, swap the A and B inputs. In a differential wiring configuration, swap the A and not A inputs.

Remove power from the rack and the pulsetach. Reconnect the coupling between the motor and the pulsetach. Reapply power to the system.

Step 4. If external inputs are used, verify that the external input circuitry is working correctly.

Toggle the external input device. Verify that the LED associated with that particular bit (LATCH, COUNT STOP or CLEAR) is also toggling. If it is not, the external input circuitry on the module is not working.

If the input triggers on the incorrect level (as specified in register 6, bits 12, 13, or 14), it may indicate a problem with switch bounce. Care must be taken to eliminate any switch bounce before the input. If switch bounce persists, a proximity switch or photoelectric sensor with hysteresis is recommended.

Step 5. Verify that the module can be accessed.

Connect an IBM-compatible personal computer to the system and load the Programming Executive software. Refer to the AutoMax Programming Executive instruction manual for more information.

Using the MONITOR I/O selection, monitor the external input device and determine whether the bit is changing state when toggled.

If you are able to read the input, the problem is in the application task. Go to step 6. If the programming device cannot read the inputs, the problem is in the hardware. Go to step 6.

Step 6. Verify that the I/O definitions are correct.

For modules in a local rack, the slot number must agree with the slot that the module is actually in. Verify that the register and bit number are correct.

For modules in a remote rack, a master Remote I/O module (M/N 57C416) must be located in the master rack and connected via a coaxial cable to a slave Remote I/O module located in the drop that contains the Pulsetach Input module.

Verify that the master Remote I/O module is in the correct slot. Next, verify that the drop number on the faceplate of the slave Remote I/O module agrees with the drop number being referenced in the task.

The slot number must agree with the slot that the module is actually in. Verify that the register and bit number are correct.

Step 7. Verify that the application task is correct.

Verify that the application task that references I/O on the module has defined the corresponding variable names as COMMON.

Step 8. Verify that the hardware is working correctly.

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

For local I/O, the problem may be in one of four areas. Check these, one at a time, and determine whether the problem has been corrected before moving to the next area. When replacing modules, if the problem is not corrected, replace the original module before moving to the next step.

First, replace the Pulsetach Input module and then replace the Processor module(s).

If the problem still exists, remove all of the modules from the backplane except one Processor module and the Pulsetach Input module. If the problem is now corrected, one of the other modules in the rack is not working. Re-connect the other modules one at a time until the problem returns. If none of these tests reveals the problem, replace the backplane.

For remote I/O, first verify that the master Remote I/O module is communicating with the drop that contains the input module. Next determine whether the input module is the only module that is not working. If it is not, the problem is most likely in the remote I/O system. Refer to J-3606 (Remote I/O Communications Module Instruction manual) for additional information. Otherwise, the problem is most likely in the remote rack.

To troubleshoot the remote rack, first replace the input module and then replace the slave Remote I/O module. If the problem still exists, remove all of the modules from the remote backplane except the slave Remote I/O module and the input module. If the problem is now corrected, one of the other modules in the rack is not operating properly. Re-connect the other modules one at a time until the problem returns. If none of these tests reveals the problem, replace the remote backplane.

5.2 Bus Error

A bus error is reported on the faceplate of a Processor module as an error code display. A bus error occurs when the system attempts to access the Pulsetach Input module and the module is missing, is in the wrong slot, is not operating properly, or you are attempting to write to the wrong registers on the module.

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

Use the following procedure to isolate a bus error.

Step 1. Verify that the Pulsetach Input module is in the correct slot as defined in the configuration.

Step 2. Verify that the module can be accessed.

Connect an IBM-compatible personal computer to the system and load the Programming Executive software. Refer to the AutoMax Programming Executive instruction manual for more information.

Using the Monitor I/O selection, monitor the eight registers used by the Pulsetach Input module.

If you are able to monitor the inputs, the problem is in the application task. Go to step 3. If the programming device cannot monitor the inputs, the problem is in the hardware. Go to step 5.

Step 3. Verify that the I/O definitions are correct.

For modules in the local rack, the slot number must agree with the slot the module is actually in. For the Pulsetach Input module, the register number must be from 0 to 7.

For modules in a remote rack, a master Remote I/O module (M/N 57C416) must be located in the master rack and connected via a coaxial cable to a slave Remote I/O module located in the drop that contains the Pulsetach Input module.

Verify that the master Remote I/O module is in the correct slot. Next, verify that the drop number on the faceplate of the slave Remote I/O module agrees with the drop number being referenced in the task.

Step 4. Verify that the application task is correct.

Registers 0 and 1 of the Pulsetach Input module cannot be written to.

If a BASIC task caused the bus error, the error log will contain the statement number in the task where the error occurred. If a Ladder Logic/PC or Control Block task caused the error, you will need to search the task for any instances where you used an input as a ladder logic coil or wrote to it in a Control Block task.

Step 5 Verify that the hardware is working correctly.

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

For local I/O, the problem may be in one of three areas. Replace these items, one at a time, and determine whether the problem has been corrected before attempting to replace the next item. First, replace the Pulsetach Input module, then the Processor modules, and finally the backplane.

For remote I/O, determine whether the Pulsetach Input module is the only module that is not working. If it is not, the problem is most likely in the remote I/O system. Refer to J-3606 (Remote I/O Communications Module instruction manual) for additional information. Otherwise, the problem is most likely in the remote rack. Replace the Pulsetach Input module, now the slave Remote I/O module, and finally the backplane. Replace these items one at a time and determine whether the problem has been corrected before attempting to swap out the next item.

5.3 Interrupt Problems

Problems with interrupts fall into two categories: either no interrupts at all or too many (unexpected) interrupts. Because interrupts affect task execution, many of these problems result in error codes being displayed on the faceplate of the Processor. Examples of tasks that use interrupts are shown in chapter 4.

Perform the following checks before you begin troubleshooting the particular symptom:

WARNING

INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

Step 1 Verify that the Pulsetach Input module is in the correct slot as defined in the configuration.

Step 2 Verify that the I/O definitions are correct.

Verify that the configuration correctly defines the registers on the module.

Step 3 Verify that the application task is correct.

Verify that the application task that uses the symbolic names defined in the configuration has defined those names as COMMON.

5.3.1 No Interrupts

If interrupts are never received and the timeout parameter in the hardware EVENT statement in the task is disabled, the task will never execute and there will be no error codes displayed on the Processor module.

Step 1. Verify that the application task is correct.

Verify that your interrupt response task is checking the proper interrupt acknowledge bit to determine which bit caused the interrupt. Confirm that when an interrupt is located the interrupt acknowledge bit is being reset. If this is not done, the interrupt will occur once and will not occur again.

Step 2. Verify that the inputs are wired correctly.

Confirm that all terminal strip connections are tight. Use an oscilloscope and check the pulses coming from the pulsetach. The duration of the pulse should be at least 3 microseconds. The square wave signals should be free of electrical noise. If electrical noise is present, check the ground connections and increase the signal's shielding as necessary.

If external inputs are used, connect a voltmeter to the proper points on the terminal strip and toggle each external device. The voltmeter should alternate between 0 and maximum voltage. If this does not happen, there is a problem with either the device or the wiring to the terminal strip.

Step 3. Verify that the input circuit on the module is working correctly.

Remove power from the system. Disconnect the mechanical coupling between the motor and the pulsetach. Apply power to the rack and the pulsetach.

Move the pulsetach in both directions. The FORWARD LED on the faceplate should turn on when the pulsetach is rotated in the forward direction. The REVERSE LED should turn on when the pulsetach is rotated in the opposite direction.

If the pulsetach rotates in the wrong direction, the pulsetach input wires must be switched. In a single-ended wiring configuration, swap the A and B inputs. In a differential wiring configuration, swap the A and not A inputs.

Remove power from the rack and the pulsetach. Re-connect the coupling between the motor and the pulsetach. Re-apply power to the system.

If external inputs are used, toggle the input device and verify that the appropriate LEDs on the module's faceplate are also toggling. If they are not, the input circuit on the module is not working properly.

Step 4. Verify that the module can be accessed.

Connect an IBM-compatible personal computer to the system and load the Programming Executive software. Refer to the AutoMax Programming Executive instruction manual for more information.

Use the MONITOR I/O selection to display registers 6 and 7. Continue to toggle the input device and determine if the proper bits are changing state. If the bits are not changing state, the input circuit on the module is not working.

Step 5 Verify that the hardware is working correctly.

WARNING
INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

The problem may be in one of three areas. Replace these items one at a time and determine whether the problem has been corrected before attempting to replace the next item. First replace the Pulsetach Input module, then the Processor modules, and finally the backplane.

5.3.2 Hardware Event Time-out

A hardware event time-out results in error code "12" appearing on the faceplate of the Processor. It means that the interrupt has either never occurred or is occurring at a slower frequency than the value specified in the TIMEOUT parameter in the EVENT definition. When this time-out occurs all tasks in the rack will stop.

Step 1 Verify that the TIMEOUT value is set correctly.

Check the value specified in the TIMEOUT parameter in the event definition. The number is in ticks. The tick value defaults to 5.5 msec. The time out value should be at least 2 ticks greater than the interrupt frequency. It is usually 1.5 times the interrupt frequency.

Step 2 Check for a "no interrupt condition". See section 5.3.1.

The TIMEOUT parameter in the EVENT statement should be disabled if interrupts are not time-based. See section 4.4.3 for an example.

5.3.3 Hardware Event Count Limit Exceeded

This condition results in error code "1b" appearing on the faceplate of the Processor module. It means that a hardware interrupt has occurred and no task is waiting. When this error occurs all tasks in the rack will stop.

Step 1 Verify that the application task is correct.

Verify that your interrupt response task contains either a "WAIT ON" event or "CALL SCAN_LOOP" statement that will be executed. Check carefully to determine whether a higher priority task is preventing the interrupt response task from running.

Step 2. Verify that the hardware is working correctly.

WARNING
INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

The problem may be in one of three areas. Replace these items one at a time and determine whether the problem has been corrected before attempting to replace the next item. First replace the Pulsetach Input module, then the Processor module(s), and finally the backplane.

Step 3. If you are using the comparator equal function, you must set the comparator value before you enable the comparator equal interrupt (bit 11 of register a).

The TIMEOUT parameter in the EVENT statement should be disabled if interrupts are not time-based. See section 4.1.3 for an example.

At power up, all internal registers are reset to zero. If you do not set the comparator value before you enable the interrupt at power up, when all internal registers are equal to zero, a comparator equal interrupt will be issued (error code "1b").

5.3.4 Illegal Interrupt Detected

This condition results in error code "1E" appearing on the faceplate of the Processor module. It means that a hardware interrupt has occurred and no event has been defined using the (BASIC) EVENT statement. When this error occurs, all tasks in the rack will be stopped.

Step 1. Verify that the application task is correct.

Verify that your interrupt response task contains an "EVENT" statement that will be executed. Check carefully to determine whether a higher priority task is preventing the interrupt response task from running.

Step 2. Verify that the hardware is working correctly.

WARNING
INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES MAY RESULT IN UNEXPECTED MACHINE MOTION. TURN OFF POWER TO THE RACK BEFORE INSERTING OR REMOVING A MODULE OR ITS CONNECTING CABLES. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

The problem may be in one of three areas. Replace these items one at a time and determine whether the problem has been corrected before attempting to swap out the next item. First replace the Pulsetach Input module, then the Processor module(s), and finally the backplane.

Appendix A

Technical Specifications

Ambient Conditions

- Storage Temperature: −40° to 185°F (−40° to 85°C)
- Operating Temperature
  (at the module): 32° to 140°F (0° to 60°C)
- Humidity: 5-90% non-condensing

Dimensions

- Height: 11.75 inches (29.8 cm)
- Width: 1.25 inches (3.2 cm)
- Depth: 7.37 inches (18.7 cm)
- Weight: 1 pound, 13 ounces (0.815 kilograms)

Maximum Module Power Dissipation

- 5.4 watts

Module Power Requirements

- Backplane: 5V x 900 mA = 4.5 watts
- Pulsetach Power Supply: 5V at 25 mA + pulsetach requirements
  12V at 25 mA + pulsetach requirements

Pulse Input Specifications

- Input Channel: 1
- Type of Input: Differential, emitter-follower,
  open-collector, or single ended
- Voltage Level: TTL to maximum 13.2 VDC
- Minimum Input Current: 5 mA
- Rated Input Current: 10 mA @ 4.0 VDC
- Maximum Input frequency: 150 kHz
- Isolation: 2500Vrms
- Digital and Analog Filter Combined: Cut-off frequency 250 kHz
- Maximum Wire Length: 600 feet (180 meters)

Appendix A

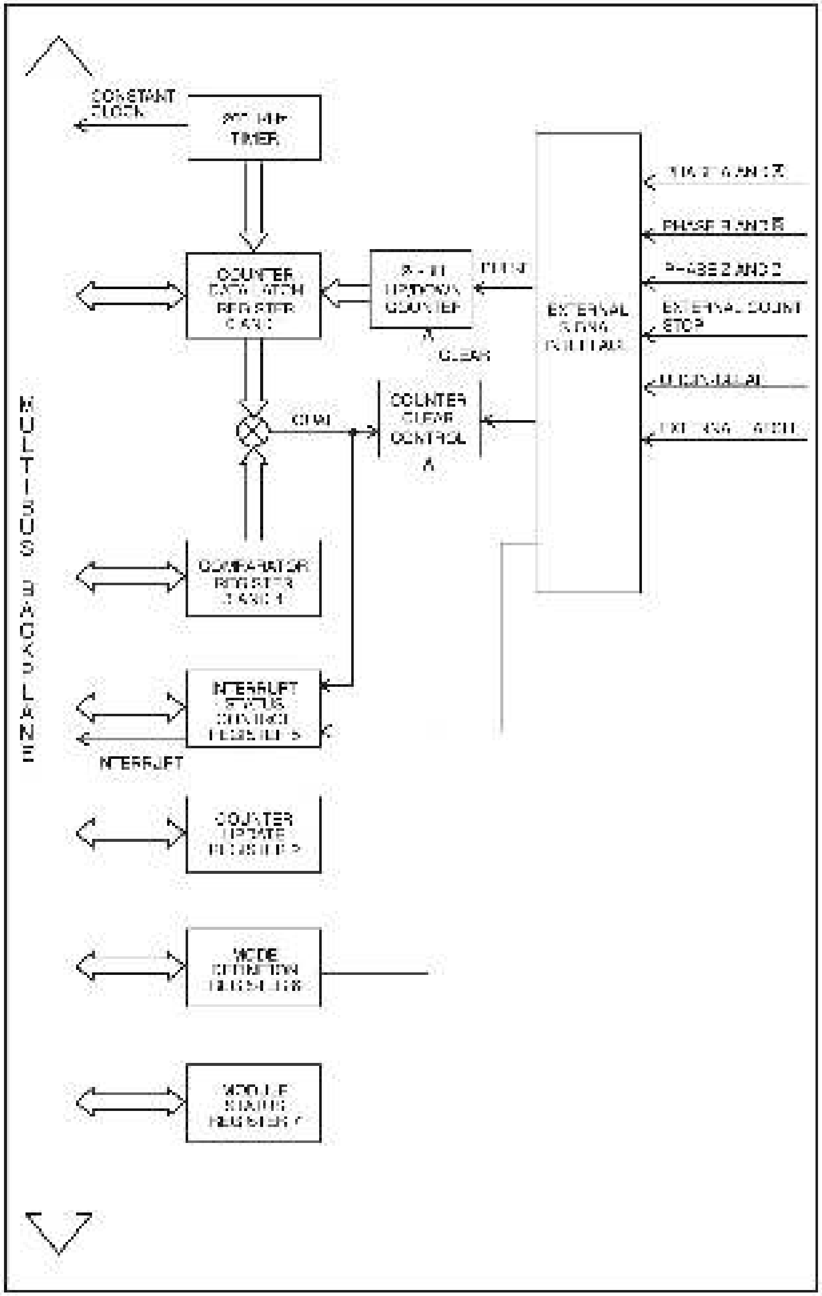
(Continued)

Control Signal Specifications

- Type of Input: Current sink or source
- Voltage Level: 5 VDC or 12 VDC
- Nominal On State Current: 10 mA
- Maximum Input frequency: 100 kHz
- Isolation: 2500Vrms
- Digital and Analog Filter Combined: Cutoff frequency 250 kHz
- Maximum Wire Length: 500 feet (150 meters)

Appendix B

Module Block Diagram



Appendix C

Field Connections

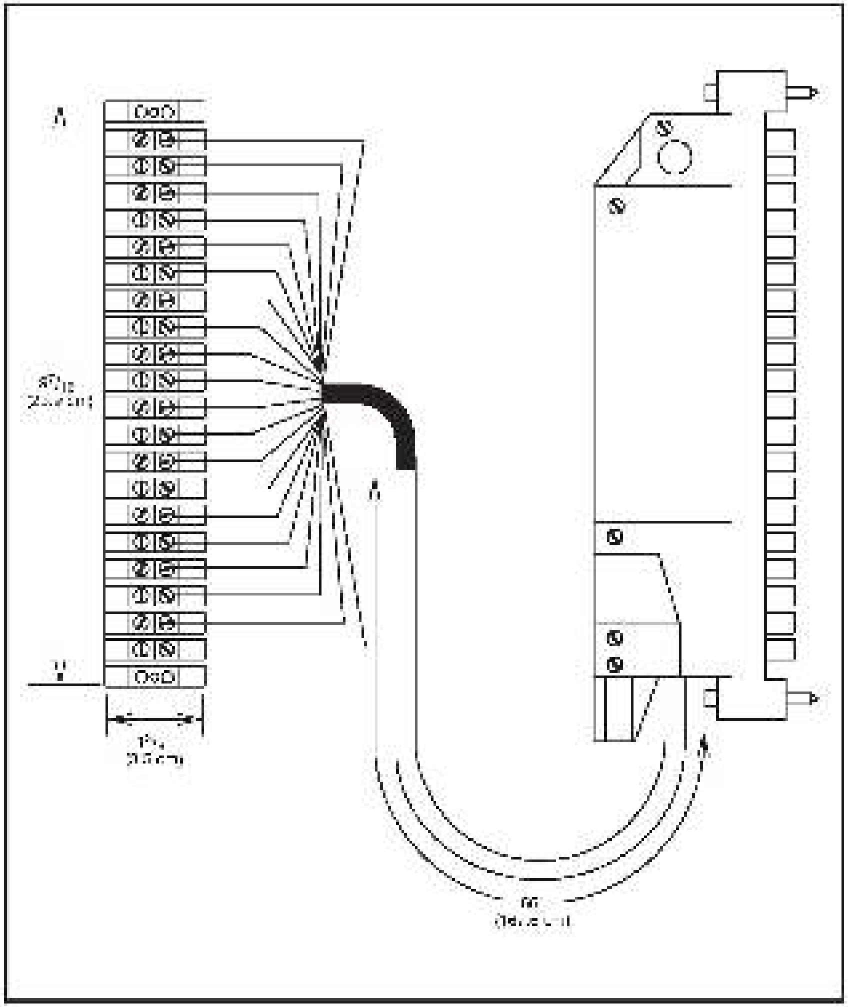
Terminal Pin No.	Wire Color Code	Function
1	Brown	No Connection
2	White/Brown	Pulsetach Phase A (+12V)
3	Red	Pulsetach Phase B (+12V)
4	Orange	Pulsetach Z Pulse (+12V)
5	White/Orange	No Connection
6	Yellow	Origin/Clear Input
7	White/Yellow	Origin/Clear Common
8	Green	Latch Input
9	White/Green	Latch Common
10	Blue	Count Stop Input
11	White/Blue	Count Stop Common
12	Violet	Pulsetach Phase A+ (+5 VDC)
13	N/A	No Connection
14	White/Violet	Pulsetach Phase A– or A Common
15	Gray	Pulsetach Phase B+ (+5 VDC)
16	Tan	No Connection
17	White/Gray	Pulsetach Phase B– or B Common
18	Pink	Pulsetach Z Pulse+ (+5 VDC)
19	White/Tan	No Connection
20	White/Pink	Pulsetach Z Pulse– or Z Common

Appendix D

Related Components

57C372 - Terminal Strip/Cable Assembly.

This assembly consists of a terminal strip, cable and mating connector. It is used to connect field signals to the faceplate of the input module. This assembly must be ordered separately from the input module.



Appendix E

Module 57421-1, 57C421A, and 57C421B Compatibility

Module 57C421A differs from module 57421-1 as follows:

- Channel A (terminals 12 to 14), channel B (terminals 15 to 17), and the Z pulse (terminals 18 to 20) differential inputs are isolated. Input voltages can range from 5 to 24V. An external series resistor must be used for voltages in excess of 12V. The minimum voltage for a 5V input is TTL level.
- The input voltage range for a single-ended input is 5 to 24V. An external series resistor must be used for voltages in excess of 12V. The minimum voltage for a 5V input is TTL level.
- The maximum input pulse frequency is 100 kHz.
- Terminal 1 is not connected to the module's internal circuitry. Module 57421-1 used this terminal as the external 12V power source for single-ended inputs.
- Terminal 5 is not connected to the module's internal circuitry. Module 57421-1 used this terminal as the common ground for the external 12V power supply used with the single-ended inputs.
- The upper byte of register 3 is the sign extension byte. Register 3 can be directly assigned through the IODEF statement in the configuration task. Register 3 and register 4 together form a 24-bit comparator.
- A 2 kHz timer mode is provided.
- The 24-bit counter can be programmed with a position feedback value with programmable scan-based interrupts.
- Register 5, bit 13 - In the 57421-1 module, this bit was not used.

  In the 57C421A module, this bit determines whether the module is to used as a counter with pulsetach inputs or as a timer using its internal 250 kHz clock. Refer to section 4.2.4 for additional information.
- Register 5, bit 14 - In the 57421-1 module, this bit was not used.

  In the 57C421A module, this bit determines whether or not the counter register is cleared when an interrupt is received. Refer to section 4.2.4 for additional information.
- Register 6, bit 11 - In the 57421-1 module, this bit if equal to zero, selected an opto-isolated input. If equal to one, it selected a line receiver input.

  In the 57C421A module, this bit establishes the polarity of the Z pulse input. If equal to zero, the Z pulse polarity is positive. If equal to one, the Z pulse polarity is negative.

Appendix E

(Continued)

In addition to these programming changes, input impedances have changed from the 57421-1 module to the 57C421A module. Be sure to check the external 5V pulse input circuitry.

- Note that if you are replacing a 57421-1 module with a 57C421A module, you must add the three jumpers shown in figure 3.4 to the terminal strip of the 57C421A module in order for the module to operate properly if the original module was used with 12 V opto-isolated inputs.

Module 57C421B differs from module 57C421A as follows:

- The 2 kHz timer has been replaced by a 200 kHz timer in 57C421B.

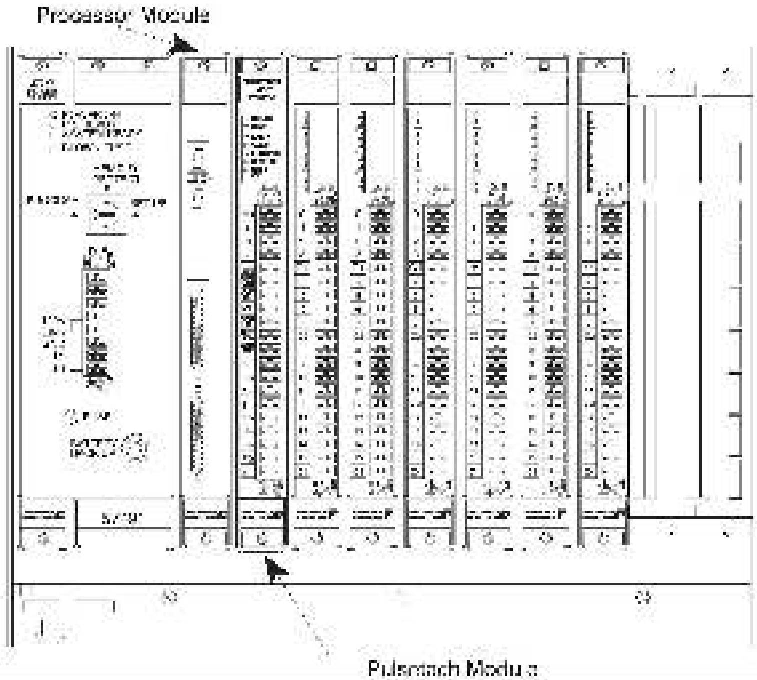
Appendix F

Defining Variables in the Configuration Task

This section describes how to configure the module when it is being used in a system with the V2.1 or earlier Programming Executive software. See instruction manual J-3619 for more information on the configuration task. For later versions of the Programming Executive software, you need to use the software forms in the Programming Executive Variable Configurator Screens.

Local I/O Definition

The statements below are used to configure modules in a local rack in the configuration task. See the figure below for an example.



Sample Module in a Local Rack

32-Bit Register Reference

Use the following method to reference 32 bits as a single register value. The counter and the comparator may be referenced this way. One statement is required in the configuration task for each variable. The symbolic name of the register should be as descriptive as possible:

```
nnnnn          GREF SYMBOLIC_NAME [SLOT = s, REGISTER = r]
```

Appendix F

(Continued)

When referenced as a double integer of 32 bits, register "r" contains the high-order 16 bits and register "r−1" contains the low-order 16 bits.

16-Bit Register Reference

Use the following method to reference 16 bits as a single integer value. The counter, comparator, timer, and SCR should be referenced this way. One statement is required in the configuration task for each variable. The symbolic name of the register should be as descriptive as possible:

```
nnnnn        CDEF SYMBOLIC_NAME%(SLOT=s, REGISTER=r)
```

Bit Reference

Use the following method to reference individual bits in a register. Counter status and control bits are typically referenced this way. One statement is required in the configuration task for each bit. The symbolic name of the bit should be as descriptive as possible.

```
nnnnn        CDEF SYMBOLIC_NAME@(SLOT=s, REGISTER=r, BIT=b)
```

where:

nnnnn –	BASIC statement number. This number may range from 1-32767.
SYMBOLIC_NAME! –	A symbolic name chosen by the user and ending with (!). This indicates a double integer data type and all references will access registers "r" and "r−1".
SYMBOLIC_NAME% –	A symbolic name chosen by the user and ending with (%). This indicates an integer data type and all references will access register "r".
SYMBOLIC_NAME@ –	A symbolic name chosen by the user and ending with (@). This indicates a boolean data type and all references will access bit number "b" in register "r".
s –	Slot number that the module is plugged into. This number may range from 0-15.
r –	Specifies the register that is being referenced. For this module, using long integers, this number must be either 0 or 3. For all other references, this number may range from 0-7.
b –	Used with boolean data types only. Specifies the bit in the register that is being referenced. This number may range from 0-15.

Appendix F

(Continued)

Examples of Local I/O Definitions

The following statement assigns symbolic name COUNTS! to register 6 on the input module located in slot 11:

```
[illegible]      CDEF COUNTS!(SLOT=11, REGISTER=6)
```

The following statement assigns symbolic name UPDATE_TIME% to register 2 of the input module located in slot 4:

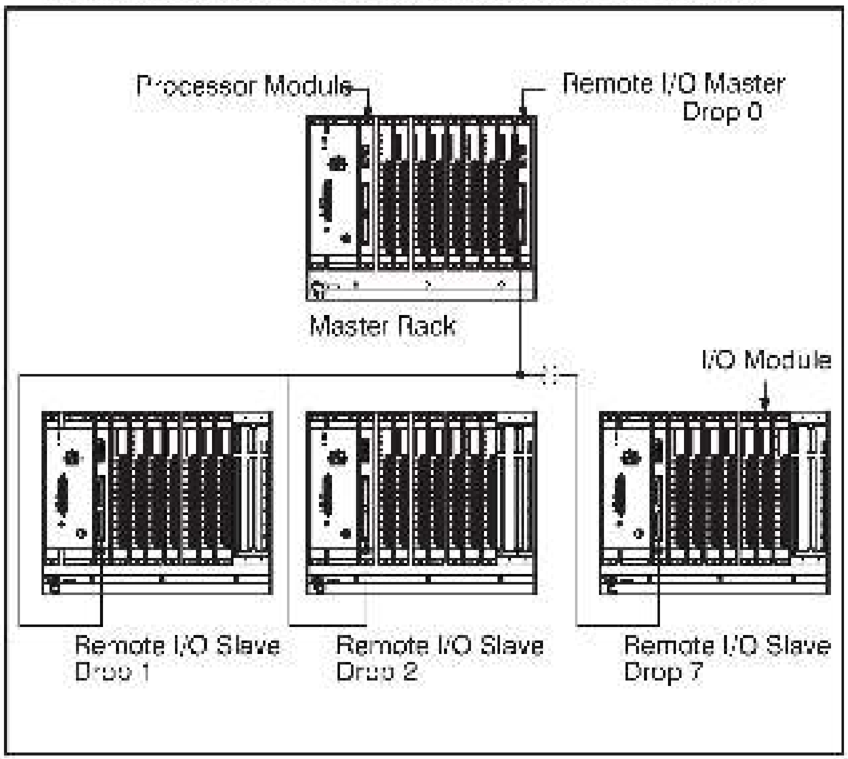
```
[illegible]      CDEF UPDATE_TIME%(SLOT=4, REGISTER=2)
```

The following statement assigns symbolic name INPUT_PULSE_TYPE@ to bit 8 of register 6 on the input module located in slot 7:

```
[illegible]      CDEF INPUT_PULSE_TYPE@(SLOT=7, REGISTER=6, BIT=8)
```

Remote I/O Definition

This section describes how to configure the module when it is located in a rack that is remote from the Processor that is referencing it. Refer to the following figure. Note that interrupts cannot be used with the module in a remote rack.



Module in a Remote Rack

Appendix F

(Continued)

32-Bit Register Reference

Use the following method to reference 32 bits as a single register value. One statement is required in the configuration task for each variable. The symbolic name of the double integer should be as descriptive as possible:

```
nnnnn     RIODEF SYMBOLIC_NAME![MSTR_SLOT=m, DROP=d, SLOT=s, REG=r]
```

When referenced as a double register of 32 bits, register r contains the high order 16 bits.

A 32 bit register reference over remote I/O should be used with care since the remote I/O system cannot guarantee that the entire 32-bit value will move as a single operation. For more information refer to the Remote I/O Communications Module Instruction Manual (J-3606).

WARNING
IF YOU USE DOUBLE INTEGER VARIABLES IN THIS INSTANCE, YOU MUST IMPLEMENT A SOFTWARE HANDSHAKE BETWEEN THE TRANSMITTER AND THE RECEIVER TO ENSURE THAT BOTH THE LEAST SIGNIFICANT AND MOST SIGNIFICANT 16 BITS HAVE BEEN TRANSMITTED BEFORE THEY ARE READ BY THE RECEIVING APPLICATION PROGRAM. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY OR DAMAGE TO EQUIPMENT.

16-Bit Register Reference

Use the following method to reference 16 bits as a single register value. One statement is required in the configuration task for each variable. The symbolic name of the register should be as descriptive as possible:

```
nnnnn     RIODEF SYMBOLIC_NAME%[MSTR_SLOT=m, DROP=d, SLOT=s, REG=r]
```

Bit Reference

Use the following method to reference individual bits in a register. One statement is required in the configuration task for each variable. The symbolic name of the register should be as descriptive as possible:

```
nnnnn     RIODEF SYMBOLIC_NAME@[MASTER_SLOT=m, DROP=d, SLOT=s, REG=r, BIT=b]
```

where:

nnnnn = BASIC statement number. This number may range from 1-32767.

SYMBOLIC_NAME! = A symbolic name chosen by the user and ending with (!). This indicates a long integer data type and all references will access registers r and r+1.

SYMBOLIC_NAME% = A symbolic name chosen by the user and ending with (%). This indicates an integer data type and all references will access register "r".

Appendix F

(Continued)

SYMBOLIC_NAME@ = A symbolic name chosen by the user and ending with (@). This indicates a boolean data type and all references will access bit number "b" in register "r".

m = Slot number that the master remote I/O module is plugged into. This number may range from 0-15.

d = Drop number for the slave remote I/O module that is in the same rack as the input module. This number may range from 1-7.

s = Slot number that the module is plugged into. This number may range from 0-15.

r = Specifies the register that is being referenced. For long integers this number must be either 0 or 3. For all other references this number may range from 0-7.

b = Used with boolean data only. Specifies the bit in the register that is being referenced. This number may range from 0-15.

Examples of Remote I/O Definitions

The following statement assigns the symbolic name UPPER_LIMIT! to the module located in slot 10 of remote I/O drop 7. This remote drop is connected to the remote I/O system whose master is located in slot 9 in the master rack:

```
[illegible]    RIODEF UPPER_LIMIT!(MASTER_SLOT=[illegible], DROP=7, SLOT=[illegible])
```

WARNING
IF YOU USE DOUBLE INTEGER VARIABLES IN THIS INSTANCE, YOU MUST IMPLEMENT A SOFTWARE HANDSHAKE BETWEEN THE TRANSMITTER AND THE RECEIVER TO ENSURE THAT BOTH THE LEAST SIGNIFICANT AND MOST SIGNIFICANT 16 BITS HAVE BEEN TRANSMITTED BEFORE THEY ARE READ BY THE RECEIVING APPLICATION PROGRAM. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY OR DAMAGE TO EQUIPMENT.

The following statement assigns the symbolic name COUNTS% to register 1 on the module located in slot 4 of remote I/O drop 3. This remote drop is connected to the remote I/O system whose master is located in slot 15 in the master rack.

```
1100    RIODEF COUNTS%(MASTER_SLOT=15, DROP=3, SLOT=4, REGISTER=1)
[illegible]    RIODEF [illegible](MASTER_SLOT=5, DROP=2, SLOT=7,    &
        [illegible]=5, BIT=12)
```

Appendix F

(Continued)

Sample Configuration Task

The following is a sample configuration for an input module located in slot 5:

```
1000    TASK PG_SPEED[TYPE=CONTROL, PRIORITY=10, SLOT=4]
1010    TASK PG_LATCH[TYPE=BASIC, PRIORITY=10, SLOT=4]
1020    TASK PG_COMP[TYPE=BASIC, PRIORITY=10, SLOT=4]
2000    ODEF COUNT_VAL%[SLOT=4, REGISTER=0]
2010    ODEF COUNT%[SLOT=4, REGISTER=1]
2020    ODEF [illegible]%[SLOT=4, REGISTER=2]
2030    ODEF COMP_VAL%[SLOT=4, REGISTER=3]
2040    ODEF LOW_COMP%[SLOT=4, REGISTER=4]
2050    ODEF ISCR%[SLOT=4, REGISTER=5]
2060    ODEF CREG1%[SLOT=4, REGISTER=6]
2070    ODEF CREG2%[SLOT=4, REGISTER=7]
2080
2090      ! REGISTER #5 BITS
2100
3000    ODEF DIS_CLAMP@[SLOT=4, REGISTER=5, BIT=14]
3010    ODEF TIM_MODE@[SLOT=4, REGISTER=5, BIT=13]
3020    ODEF MULT%[SLOT=4, REGISTER=5, BIT=12]
3030    ODEF POLL_INT@[SLOT=4, REGISTER=5, BIT=11]
3040    ODEF ORG_NEW@[SLOT=4, REGISTER=5, BIT=10]
3050    ODEF ST_IN@[SLOT=4, REGISTER=5, BIT=9]
3060    ODEF LAT_IN@[SLOT=4, REGISTER=5, BIT=8]
3070    ODEF CCLK_EN@[SLOT=4, REGISTER=5, BIT=6]
3080    ODEF INT_RX@[SLOT=4, REGISTER=5, BIT=5]
3090    ODEF CIR_MODE1@[SLOT=4, REGISTER=5, BIT=[illegible]]
3100    ODEF CIR_MODE2@[SLOT=4, REGISTER=5, BIT=[illegible]]
3110
3120      ! REGISTER #6
3130
4000    ODEF RESET@[SLOT=4, REGISTER=6, BIT=15]
4010    ODEF LATCH_POLARITY@[SLOT=4, REGISTER=6, BIT=14]
4020    ODEF ORIGIN_POLARITY@[SLOT=4, REGISTER=6, BIT=13]
4030    ODEF STOP_POLARITY@[SLOT=4, REGISTER=6, BIT=12]
4040    ODEF SINGLE_INPUT@[SLOT=4, REGISTER=6, BIT=11]
4050    ODEF [illegible]@[SLOT=4, REGISTER=6, BIT=10]
4060    ODEF INHIBIT_CNT@[SLOT=4, REGISTER=6, BIT=9]
4070    ODEF NUMBER_CHANNELS@[SLOT=4, REGISTER=6, BIT=8]
4080    ODEF Z_PULSE_REV@[SLOT=4, REGISTER=6, BIT=3]
4090    ODEF Z_PULSE_RATIO@[SLOT=4, REGISTER=6, BIT=2]
4100    ODEF EXT_STOP_EN@[SLOT=4, REGISTER=6, BIT=1]
4110    ODEF EXT_ORG_EN@[SLOT=4, REGISTER=6, BIT=0]
4120
4130      ! REGISTER #7
4140
5000    ODEF ORIGIN_RESET@[SLOT=4, REGISTER=7, BIT=15]
5010    ODEF STOP_RESET@[SLOT=4, REGISTER=7, BIT=14]
5020    ODEF LATCH_RESET@[SLOT=4, REGISTER=7, BIT=13]
5030    ODEF CNTR_TO_ZERO@[SLOT=4, REGISTER=7, BIT=12]
5040    ODEF [illegible]@[SLOT=4, REGISTER=7, BIT=11]
5050    ODEF CARRY_REAL@[SLOT=4, REGISTER=7, BIT=10]
5060    ODEF COUNT_[illegible]@[SLOT=4, REGISTER=7, BIT=9]
5070    ODEF CLOCK_ERROR@[SLOT=4, REGISTER=7, BIT=8]
5080    ODEF EXT_ORG_STATUS@[SLOT=4, REGISTER=7, BIT=7]
5090    ODEF EXT_STOP_STATUS@[SLOT=4, REGISTER=7, BIT=6]
5100    ODEF EXT_[illegible]_STATUS@[SLOT=4, REGISTER=7, BIT=[illegible]]
5110    ODEF COUNTER_[illegible]@[SLOT=4, REGISTER=7, BIT=4]
5120    ODEF COUNTER_[illegible]@[SLOT=4, REGISTER=7, BIT=3]
5130    ODEF COUNTER_[illegible]@[SLOT=4, REGISTER=7, BIT=2]
5140    ODEF BORROW_STATUS@[SLOT=4, REGISTER=7, BIT=1]
5150    ODEF CARRY_STATUS@[SLOT=4, REGISTER=7, BIT=0]
32767   END
```

Appendix G

Memory Map

Register	Bit	Description
0		Counter Data Latch Register
1		Counter Data Latch Register
2		Counter Data Update Preset Register
3		Comparator Register
4		Comparator Register
5		Interrupt Status and Control Register
	0	System use only
	1	System use only
	2	System use only
	3	Counter clear control
	4	Counter clear control
	5	Timer interrupt enable
	6	Generate OCLK
	7	System use only
	8	External latch interrupt enable
	9	External count stop interrupt enable
	10	Z pulse and latch interrupt enable
	11	Comparator equal interrupt enable
	12	Pulse multiplier
	13	Time base select
	14	Counter clear inhibit
	15	System use
6		Mode Definition Register
	0	External latch enable
	1	External count stop enable
	2	Count reverse
	3	Count forward
	4	System use only
	5	System use only
	6	System use only
	7	System use only
	8	Type of count select
	9	Inhibit count
	10	Origin/clear select
	11	Z pulse polarity
	12	Count stop input select
	13	Origin/clear input select
	14	External latch input select
	15	Reset counter
7		Module Status Register
	0	Carry status
	1	Borrow status
	2	Counter greater than comparator
	3	Counter less than comparator
	4	Counter equals comparator
	5	External latch input status
	6	External count stop input status
	7	Origin/clear input status
	8	OCLK off
	9	Pulse input direction
	10	Carry status reset
	11	Borrow status reset
	12	Counter equals comparator status reset
	13	External latch status reset
	14	External count stop status reset
	15	External origin/clear status reset

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

www.rockwellautomation.com

Corporate Headquarters
[illegible]

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
[illegible]

Headquarters for Dodge and Reliance Electric Products
[illegible]

Publication [illegible]