CTI 2541
REDUNDANT PROCESSOR MANAGER
INSTALLATION AND OPERATION GUIDE
Version 1.4

CTI Part # 062-00229-014

|||||||||||||||||||||||

DOCUMENT DISCLAIMER STATEMENT

Every effort has been made to ensure the accuracy of this document; however, errors do occasionally occur. CTI provides this document on an "as is" basis and assumes no responsibility for direct or consequential damages resulting from the use of this document. This document is provided without express or implied warranty of any kind, including but not limited to the warranties of merchantability or fitness for a particular purpose. This document and the products it references are subject to change without notice. If you have a comment or discover an error, please call us toll-free at 1-800-537-8398.

REVISION HISTORY

Version 1.0   3/9/98   Original Release.

Version 1.1   6/9/98   Documented Fail Safe switch changes on Rev C boards,
                         Added RTS COM switch,
                         Updated front panel IN CONTROL LED description,
                         Added offset 11 and 13 function descriptions.

Version 1.2   01/20/99   Documented RTS COM switch functionality on Rev D boards,
                           Added Siemens® RF I/O converter cabling information,
                           Included RTS COM switch setting requirements,
                           Added Configuration Port settings,
                           Added note that on-line edits can cause PLC switchover.

Version 1.3   04/01/99   Added V1.4 firmware enhancement information below:
                            Added Software Reset information,
                            Added Switchover Inhibit information,
                            Added Virtual Base PLC 'X' status bit information for Offsets 16-22.

Version 1.4   11/25/03   Corrected Figures 23, 24, 25 with correct RTS+ and RTS- lines.

PREFACE

This ***Installation and Operation Guide*** provides installation and operation instructions for the CTI 2541 RPM module. The information in this manual is directed to individuals who will be installing, maintaining, and troubleshooting the module.

This manual assumes that you are familiar with the installation and operation of SIMATIC® 505 programmable controllers. Please refer to the appropriate SIMATIC® user documentation for specific information on SIMATIC® 505 programmable controllers and I/O modules.

This manual describes the use of the 2541 Redundant Processor Manager module in a warm backup system.

USAGE CONVENTIONS

NOTE:

Notes alert the user to special features or procedures.

CAUTION:

Cautions alert the user to procedures that could damage equipment.

WARNING:

Warnings alert the user to procedures that could damage equipment and endanger the user.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1. OPERATING CONCEPTS

1.1. Introduction

This chapter describes the operation of the 2541 Redundant Processor Manager (RPM) module. The module, designed for use with Siemens® Simatic® 505 PLC equipment, switches the control of remote I/O between two PLCs. The RPM module can automatically transfer control based on when a PLC fault is detected. In addition, PLC logic can be used to control switching, or the operator can manually switch using module pushbuttons.

NOTE:

While the module is designed to reliably switch control with a minimum impact on the process being controlled, it is not a substitute for a "Hot" backup system in which PLC scans are completely synchronized.

The information in this chapter should help you understand how the RPM module may apply to your control situation.

1.2. Glossary of Terms

Term	Definition
Active PLC	The PLC which is actually controlling the I/O.
Standby PLC	The PLC that is ready to assume control of the I/O.
Switchover	The action associated with transferring control of the I/O from one PLC to the other. This event is also referred to as Transfer of Control.
Bumpless Transfer	Transfer of control from the Active PLC to the Standby PLC without affecting the operation of the process being controlled.
Hot-Backup	A control configuration in which the operation of redundant PLCs is completely synchronized. The processors solve each element of logic and update I/O at exactly the same time. Data is shared seamlessly between the PLCs. Control may be immediately switched between the PLCs, even in the middle of a scan, without any control or communications interruption, resulting in bumpless transfer.

Term	Definition
Warm-backup	A control configuration in which the operation of redundant PLCs is asynchronous (not synchronized). Since the scan of the processors is independent, each processor will usually be solving logic and updating I/O at different times. Although control of the I/O may be switched within an interval short enough to avoid an I/O communications failure, bumpless transfer is not automatically guaranteed. The user must determine data transfer requirements necessary to ensure bumpless transfer.
Virtual Base	An entity created within the RPM module that appears to be a remote I/O base to the attached PLCs. Virtual bases can be used by the PLC to monitor and control the RPM module and to transfer data from one PLC to the other.
Home Base	The Virtual Base that contains the discrete points used to monitor the status of the RPM and attached PLCs and to control the RPM operation. Home Base will contain 32 discrete inputs (X) and 32 discrete outputs (Y). It may contain up to 14 additional high-density word modules (32WX/32WY) used for critical data transfer.
Critical Data	Data that must be copied every I/O scan from the Active to the Standby PLC to facilitate bumpless transfer. Critical Data is transferred via Virtual Bases.
Non Critical Data	Data that may be transferred from the Active to the Standby PLC over several scans. Non-critical data may be transferred via Virtual Bases or via V memory transfer.

1.3. RPM Overview

The 2541 Redundant Processor Manager (RPM) module provides a *warm backup* solution for SIMATIC® 505 controllers. Should the *Active* PLC fail, the 2541 can automatically switch the I/O and operator interface devices over to a *Standby* PLC.

The RPM module provides switchover features that facilitate *bumpless transfer*. Since the RPM module intercepts all I/O updates received by the Active PLC and mirrors them to the Standby PLC, the Standby operates as if it were actually controlling the physical I/O. Internal PLC data, such as counter and timer values, can be transferred from the Active PLC to the Standby PLC to eliminate program divergence.

LEDs on the front bezel indicate the status of the module and PLC control conditions. Recessed front panel pushbuttons allow the user to manually transfer control between the PLCs. The module uses the standard Siemens® front-access connector to simplify terminating PLC, I/O, and operator interface connections. An RS-232 serial port can be used to configure and diagnose the module.

A complete backup system can be configured using two bases, two power supplies, two PLCs, and one or more racks of remote I/O. Only one RPM module is required to provide warm backup.
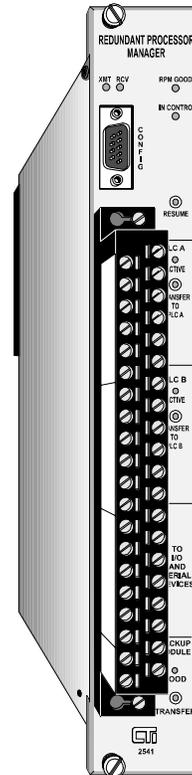
1.4. Siemens® 505 Overview

Controller and I/O Architecture

Figure 1. 2541 RPM Module

Siemens® 505 series PLC controllers are installed in a base assembly with a power supply. This local base can also contain local I/O, which communicates with the PLC via the backplane of the base. The local base is addressed as Base 0.

The PLC provides a Remote I/O port for communicating with remote I/O bases. A Remote Base Controller (RBC) in each remote base connects to the PLC via RS-485 cabling. When the RBC is polled by the PLC, it reads the I/O module data, transmits the input data to the PLC, and writes the outputs to the I/O. Up to 15 remote bases and associated I/O may be connected to the Remote I/O data port. Remote bases are addressed as 1 through 15; addresses may be assigned in any order, as long as every base has a unique address.

Remote Base PLC Communications

On initial system power up or when the PLC is initially connected to the remote I/O data network, each remote base is "logged" onto the PLC and validated against the system configuration. Once a remote base and associated I/O are accepted by the PLC, the PLC begins polling the base, writing all outputs and receiving input module data as the reply. In this fashion, each remote base is sequentially accessed by the PLC. Because of the sequential nature of polling and the time required

to communicate on the serial data link, each remote base added to the configuration increases the PLC scan time.

Should the PLC fail to update a remote I/O base within the allotted time-out period (approximately 500 ms), the remote base controller will set outputs to a predetermined state. The RBC can be configured by jumper to turn off all outputs or to hold last value in this case. If the PLC is operational and the I/O communications link is intact, the PLC will update the I/O within this period, regardless of the operating mode or scan time.

Should the PLC not receive a response from the Remote Base Controller within a certain time interval, the PLC will set the status of the remote base to "offline". When a remote base is offline, all related inputs are set to zero. The PLC will periodically attempt to re-establish communications with the remote base. Once communication is re-established, the PLC will begin polling the remote base again. Status Word 2 can be used to monitor the status of all remote bases.

The messages between the PLC and the remote base may also contain non-I/O (task code) instructions. These messages are assigned a lower priority than the I/O messages and are queued into the high-priority I/O data stream.

1.5. RPM Remote I/O Switching

The Active PLC is directly connected to the remote I/O through a relay circuit in the RPM module.
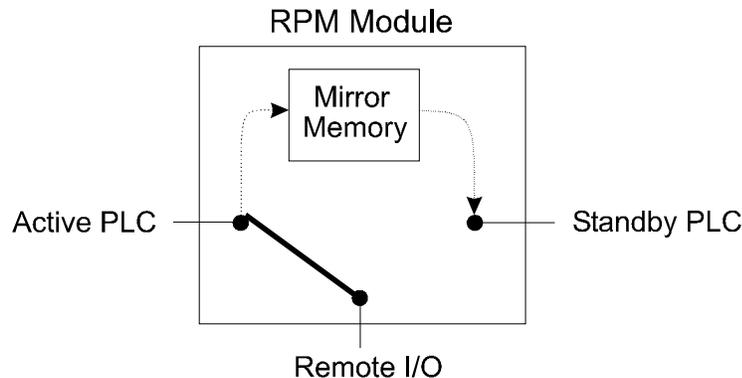
The module firmware can set this relay to connect either PLC A or PLC B to the remote I/O. The RPM module provides hardware that will connect the I/O to a specific PLC in case of power failure or module fault. A slide switch on the circuit board selects this connection option.

Since the Standby PLC is disconnected from the remote I/O by the relay circuit, it cannot obtain data directly from the I/O. To keep the Standby PLC I/O data current, the RPM module caches the input data that was obtained by the Active PLC. When the Standby PLC performs an I/O update, the RPM module provides the reply from the cache. Thus, the Standby PLC logic continues to operate as if it were controlling the I/O and receiving updates.

When one of a predetermined set of events occurs (such as a loss of PLC scan), the RPM module switches the connection of the remote I/O to the Standby PLC. The Standby PLC then becomes the Active PLC, directly controlling the I/O while the other PLC becomes standby.

In addition to the I/O, the RPM module will switch nine low-level signals via a separate set of relays. These circuits, which will handle up to 24V AC/DC and 0.5 amps with a resistive load, are switched in concert with the I/O. These circuits are typically used to switch serial Operator Interface Devices between the PLCs, although they may be used for other purposes.

4

1.6. Switchover Conditions

The RPM module continually monitors both the Active and Standby PLCs for the following events.

- **I/O update** - The loss of I/O update indicates that the PLC is not attempting to communicate with the I/O.

- **PLC "heartbeat"** - The heartbeat is a change in state of a designated Y output every scan. PLC logic is used to toggle the output. The heartbeat is used to detect when the PLC is not in Run mode; when the PLC is placed in Program mode, the heartbeat will cease.

- **PLC Status Words** - The first status word contains a bit which indicates that a fatal error has occurred. Other status words provide program checksum data used to determine whether both PLCs have identical programs.

If the RPM module detects a fault condition in the Active PLC, it will automatically transfer control to the Standby PLC, if the Standby PLC is operational. You can select which events are interpreted as a fault condition using the module configuration program. See Section *2.6. Configuring the RPM Module.* The events include:

- **Missing I/O Update** - A remote I/O update from the Active PLC is not received within a specified period. This is the primary event used to detect PLC failure.

- **Missing Heartbeat** - A "heartbeat" pulse, generated by user PLC logic, is not received from the active PLC. This event is used to detect when the PLC is placed in program mode.

- **Fatal Error** - A fatal error is detected in the active PLC. This event is detected by monitoring the PLC status word.

A manual switchover can also be requested using PLC logic or through the front panel pushbuttons.

Switchover will be inhibited if a fault condition is detected in the Standby PLC. The events that result in a Standby fault condition match those selected by the user for the Active PLC:

- **Missing I/O Update** - A remote I/O update from the Standby PLC is not received within a specified period.

- **Missing Heartbeat** - A "heartbeat" pulse, generated by user PLC logic, is not received from the active PLC.

- **Fatal Error** - A fatal error is detected in the Standby PLC. In addition, manual switchover will be prevented if the user has disabled PLC logic and pushbutton transfer when configuring the module.

In addition**,** manual switchover can also be inhibited if the programs in the PLCs do not match.

Switchover will be inhibited for all conditions if the Switchover Inhibit feature is enabled through PLC logic. See *Chapter 5. PLC Programming* for PLC Control Bit information.

1.7. Data Transfer

In addition to data obtained from the I/O, many applications will require that data values from the Active PLC be transferred to the Standby PLC to prevent divergence of the programs. The selection of the data to be transferred depends upon the application and may include timers, counters, and certain loop data. *Critical data* includes rapidly changing variables that must be transferred each I/O scan in order to ensure bumpless transfer. **Non-Critical data** includes slowly changing values that can be transferred less often.

Transferring Data via Virtual Bases

Virtual Bases may be used to transfer data between the PLCs. A Virtual Base appears to both PLCs as if it were a real remote base, although it is actually a block of memory in the RPM module. A Virtual Base may be assigned to any address (1 – 15) not being used by a real I/O base. The 2541 Configuration program is used to define the Virtual Bases.

Every Virtual base may contain up to 16 virtual modules. You must create at least one Virtual Base designated as Home Base. Home Base contains one discrete input module (32X) and one discrete output module (32Y). These discrete points can be used by the PLC to monitor the RPM status and to control the RPM operation. The remaining slots can be used for the word modules.

Each Virtual Base may contain one or more virtual word modules. These modules are configured as high-density input/output (32WX / 32WY). For each "module" a WY point is tied to a corresponding WX point, so that the value written to the WY appears in the corresponding WX. See the following diagram.
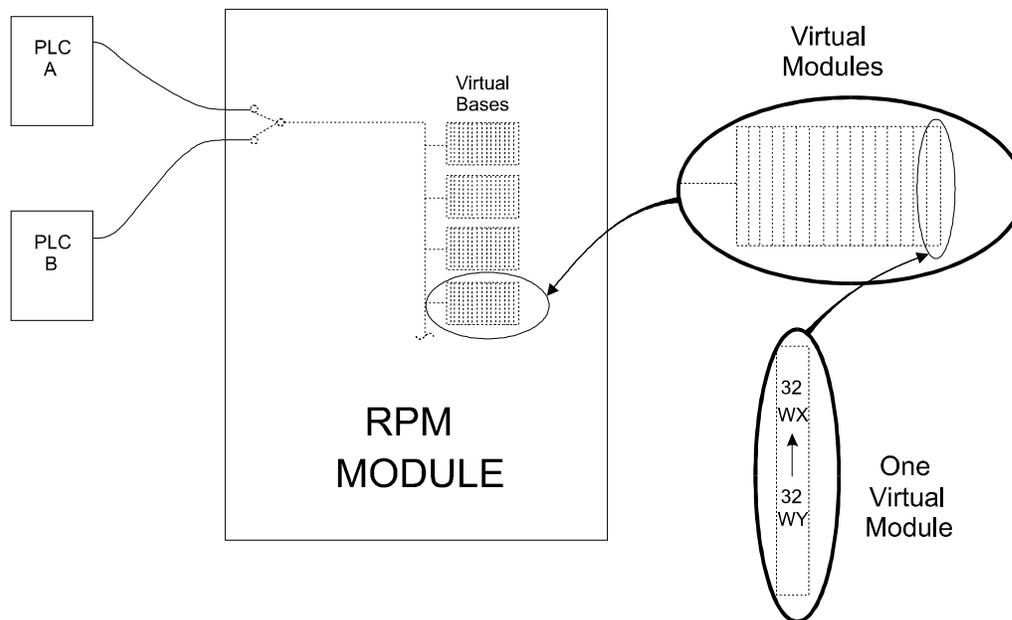
Figure 2. Virtual Base Concept

Transferring data via Virtual Bases requires user-generated logic. The logic in the Active PLC copies a critical data item to a WY word mapped to a Virtual Base. Logic in the Standby PLC copies

6

the corresponding WX word back to the appropriate PLC data element. By observing certain status bits in the Home Base, each PLC can determine whether it is the Active or Standby PLC. See *Section 5.1. Monitoring and Controlling RPM Status* for detailed information. The following figure illustrates how data is transferred from the Active PLC to the Standby PLC using Virtual Bases.
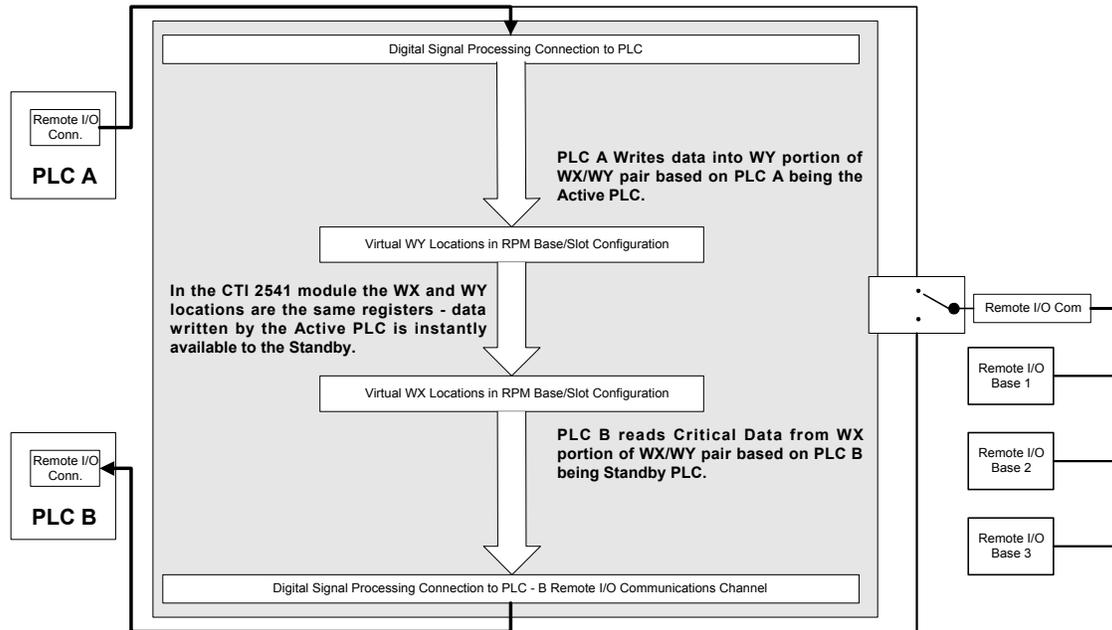
Figure 3. Virtual Base Data Transfer

Transferring Data via V Memory Transfer

The RPM module provides an alternate means of transferring data from the Active PLC to the Standby PLC. To accomplish this transfer, the RPM reads a user-designated block of V memory from the Active PLC and writes a corresponding block of V memory to the Standby PLC. No PLC programming is needed to accomplish this task, it can be implemented by the configuration program.

Since this is accomplished using task codes, which have a lower priority than I/O messages, it will usually take several scans before all of the data block can be transferred. Consequently, this method can be used for non-critical data that does not require transfer every scan.

CHAPTER 2. INSTALLATION

The installation of the Module consists of the following steps:
1. Planning the installation,
2. Unpacking the module,
3. Setting module switches,
4. Physical installation,
5. Configuring the RPM module,
6. Configuring the PLC I/O to include RPM virtual bases,
7. Checking the module operation.

2.1. Installation Planning

Rack Installation

The 2541 RPM module is a standard double-wide module Simatic® 505 module. In a typical installation, you will install the RPM module in a rack with one of the PLCs. The figure below illustrates the RPM module installed in a four-slot rack with PLC B. In case power to the rack with PLC B fails, the RPM module can be set to automatically switch control to PLC A. For additional protection, you could select bases with redundant power supplies.
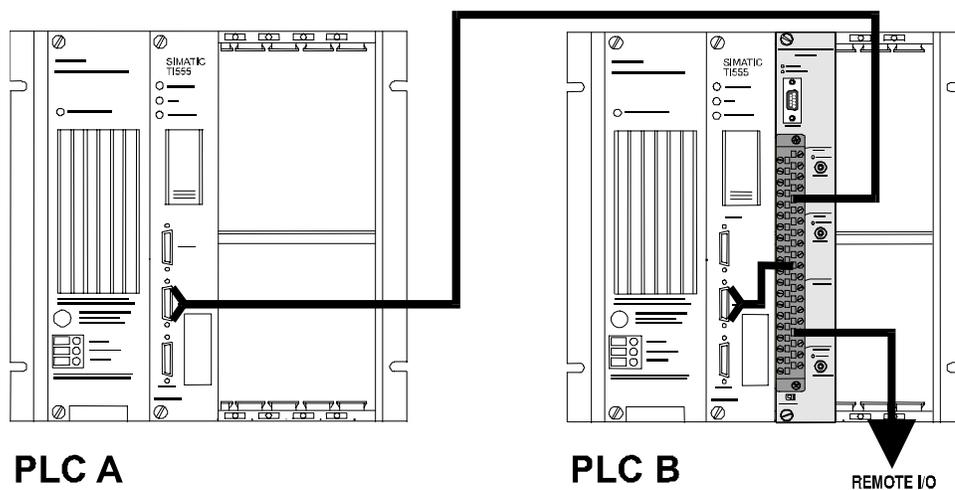
Figure 4. Typical RPM Installation

If you wish to do so, you may install the RPM module in a rack separate from both PLCs. Although the module draws power from the backplane and monitors the Early Power Fail (EPF) signal, it does not communicate with the PLC via the local base.

Cabling

You will need to construct cables to connect the RPM to the PLCs, remote I/O, and any serially connected MMI devices. To connect the remote I/O, you will need three cables, one from each PLC I/O connector to the RPM module, and one cable from the RPM module to the first remote I/O rack. These cables must comply with the standard for Siemens® SIMATIC® 505 remote I/O installation.

If you are switching operator interface devices, you will also require a similar set of cables for each of these devices.

See *Appendix A* for details on making all cables.

Identifying Programming Requirements

Since the RPM module will mirror all remote IO data, the Standby PLC will constantly be updated with data from the remote I/O. However, in order to facilitate bumpless transfer, many applications will also require that special programming techniques be used. For example, the Standby PLC may require certain data from the Active PLC to avoid program divergence. These data items may include:

- Selected timer, counter, and drum data

- Selected Analog loop and Alarm values

- Data Items written by Operator Interface or HMI stations

You will need to determine which data items need to be transferred and whether the data items must be transferred every scan (Critical Data) or whether they can be transferred over several scans (Non-critical Data). See Chapter 4. Switchover Considerations and *Chapter 5. PLC Programming* for specific guidelines and techniques.

Communicating with Operator Interface Devices

Transferring control to the standby PLC may result in a brief interruption in serial communications because switchover may occur in the middle of a message. Although most Operator Interface devices that communicate with the PLC detect and recover from these communication interruptions, you should test your particular devices to ensure that they can tolerate this interruption.

Note:

> *Any data written to the active PLC by the Operator Interface is not automatically mirrored to the Standby PLC. You will need to explicitly transfer this data.*

A TCP/IP Ethernet network may be used rather than using serial communications. If you choose to install a single 505-CP2572 module in a remote base, data will be written and read to the Active PLC only. Switchover will follow the I/O control.

An improved configuration, as seen in Figure 5 below, employs two 505-CP2572 network modules, one in each PLC local base. Since the modules are in the local base, each is dedicated to a specific PLC. With this configuration, an MMI application could monitor and update both PLCs concurrently. In addition, you can program the standby PLC while the active PLC is in control, a function not possible when using a single network module or switched serial communications.
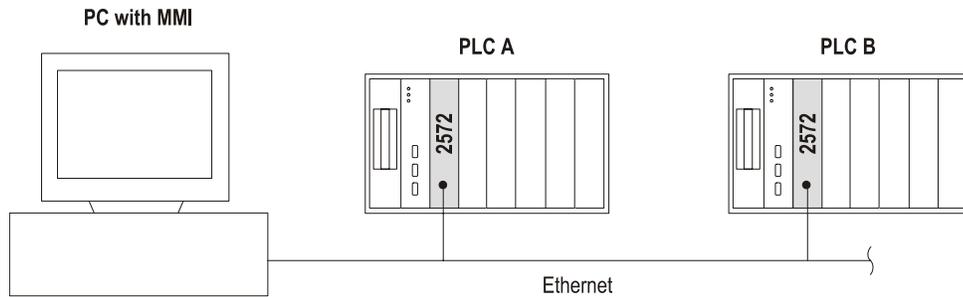
Figure 5 Dual 2572 Configuration for continuous PLC communications

2.2. Unpacking the Module

Open the shipping carton and remove the special anti-static bag that contains the module. After discharging any static build-up, remove the module from the static bag. Do not discard the static bag. Always use this bag for protection against static damage when the module is not inserted into the I/O base.

> *CAUTION:*
>
> *The components on the 2541 RPM module can be damaged by static electricity discharge. To prevent this damage, the module is shipped in a special anti-static bag. Static control precautions should be followed when removing the module from the bag and when handling the printed circuit card during configuration.*

2.3. Setting the 2541 RPM Fail Safe Switches

The RPM Fail Safe switches are used as a hard-wired backup feature. The function of these switches are to allow the user to select which PLC will be in control of the I/O when one of the following events occurs:

- The RPM module detects an Early Power Fail signal. This signal is generated by the rack power supply when power drops below a certain value (typically indicating an imminent power failure).

- Power to the local rack is lost.

- The RPM module itself fails.

During normal operation the switches are disconnected from remote I/O circuit by an energized relay, allowing the RPM software to control the connection of the remote I/O. However, if one of the previous events occurs, the disconnecting relay is de-energized and the position of the switches controls the connection of the I/O.

The Fail Safe switches can be set to the "A", "B", or "None" position. Both switches must be set to the same position for proper operation. The "A" position corresponds to PLC A and the "B" position to PLC B. Either the "A" or "B" position should be selected. For example, if the RPM module is placed in a rack with PLC B, you would set the fail-safe switch to "A". Then, in case of a power failure on the PLC B rack, the module would automatically select PLC A to control the I/O. The "None" position is reserved for future use and should not be selected.

The locations of the Fail Safe switches are shown in Figure 6. For Rev C circuit boards use figure A. Rev A and B circuit boards used a single combined switch as shown in figure B.

Rev C circuit boards   Rev A & B circuit boards
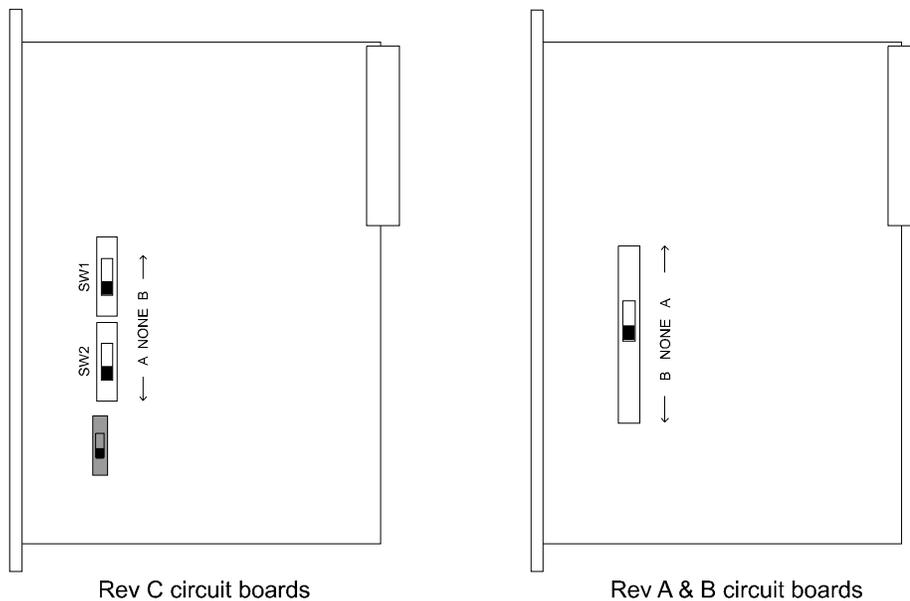
Figure 6. 2541 Failsafe Switch Locations

Note:

12

Both Fail Switches (SW1 and SW2) must be in the same position for proper hardwire switching.
There are no indications of improperly set switches until a fail safe situation occurs.

2.4. Setting the RTS COM Switch

The RTS COM switch is used to configure six of the Front Access connector terminals. For typical applications, with the switch set to COM, the function of these six terminals are for switching auxiliary devices. The RTS position is for applications where RTS signaling will be required, the switch configures these terminals to pass or send RTS signals. The layout and definition of all front panel terminals are found in *Appendix A. Connectors and Cabling.*

Note:

Set the RTS COM switch to COM for all applications not using a Siemens® 505-6860 RS485/RF I/O channel converter. The RTS mode is for functionality requiring the use of RTS signal lines. Setting this switch to RTS will affect terminals on the front access panel on Rev. D boards.
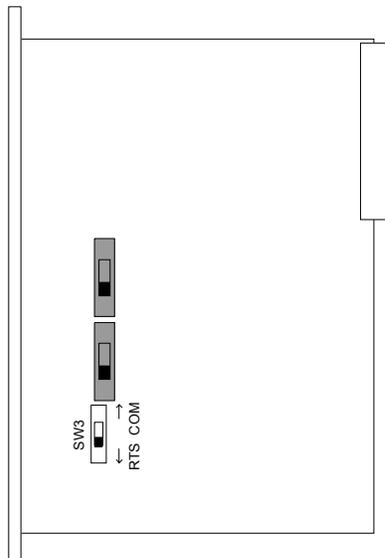
Figure 7. RTS COM Switch Location

Note:

Applications using Siemens® RF channel converters cannot use CTI's 2541-LIB Bypass Wiring Block. The 2541-LIB was designed for use with standard Remote I/O, which does not require using RTS signals.

2.5. Physical Installation

Inserting the Module into the I/O Base

Ensure that power to the base is off. Hold the top and bottom of the bezel and slide the module carefully into the slot, pushing it all the way into the base. If you have inserted the module correctly, you will feel a slight increase in resistance as the module mates with the base backplane connector. Once the module is fully seated in the slot, tighten the captive screws at the top and bottom to hold the module in place. To remove the module from the I/O base, remove power from the base, loosen the captive screws, and then remove the module. Take care not to damage the connector at the back of the module when inserting or removing the module.

Connecting Cables

Cables are connected directly to the Front Access connector. At a minimum, you must connect the cable from the I/O racks and the cables from the Remote I/O connector of each PLC. If you are switching Operator Interface devices, you will need to connect the serial cables from the devices and the PLC programming ports. Cabling requirements and procedures are described in *Appendix A. Connectors and Cabling.*

Power Up and Diagnostics

When power is applied, the module will initiate a Power-On-Self-Test (POST). During the POST routine, the RPM module will cycle all LEDs on and off as a visual LED test. The relays will remain in the de-energized position and the connection of the remote I/O will be determined by the position of the Fail-safe switch.

Once the POST routine has been successfully completed, the RPM GOOD LED will be illuminated. If the RPM GOOD indicator does not remain illuminated, the module has detected a component failure. See *Chapter 6. Troubleshooting* for help in diagnosing and correcting problems. The state of the other LEDs will be determined by status of the PLCs and the configuration settings.

2.6. Configuring the RPM Module

Figure 8. 2541 Configuration Display

The RPM module is configured using the CTI 2541 Configuration Program, a 32-bit application that runs under Windows 95 or Windows NT. Configuration data can be read from the module, modified, and downloaded to the module via the configuration port. In addition, configuration data can be saved to disk for subsequent retrieval.

Setting Control Parameters

The control parameters determine how the module will transfer control between the PLCs. The following describes the items that may be set.

Normal Startup PLC - This parameter allows the user to select which PLC (A or B) will normally be selected when the RPM module first assumes control.

Time Out Values (ms)

    I/O Update – This parameter allows you to specify the amount of time (in ms) that must elapse before the RPM will log an I/O timeout. If I/O Update is selected as a transfer option, the RPM will switch to the Standby PLC when an I/O timeout event occurs, if the Standby PLC is operational. The timeout value can range from 10 ms to 500ms. The value should be set to a value slightly greater than the maximum scan time. The maximum scan time can be read using your PLC programming software. You should set this parameter to the lowest possible value, since the delay in detecting the loss of I/O update adds to the switchover time. Setting the value too small may cause the RPM to erroneously switch back and forth between the PLCs or to inhibit switchover because the Standby PLC is detected as not operational.

> *Note:*
>
> *If the sum of the Standby scan time plus the I/O Update Timeout exceeds 450 ms, the remote I/O base could time out during switchover, bumping the outputs. For this situation, you must configure the remote I/O to hold last value in order to prevent control bumps.* See *Chapter 4. Switchover Considerations.*

    RBC Off-line Declaration – This parameter allows you to specify the amount of time (in ms) that must elapse before a remote base will be declared as off-line to the Standby PLC. This value should be set to a value slightly larger than the I/O Update Timeout.

V Memory Transfers – V Memory Transfer is used to move non-critical data from the Active PLC to the Standby PLC. The data will typically be transferred over several scans. You can specify the starting location of the V memory block and the number of words to be transferred.

> *Note:*
>
> *You should attempt to minimize the number of words to be transferred. The more words that you transfer, the more time it takes to transfer the data. The transfer rate will depend upon the PLC scan time and the I/O activity. In general, about 50 – 100 words can be transferred in one second.*

Event Log - The RPM module keeps a log of the last 64 events. This parameter allows the user to select whether the log wraps around or simply stops recording events when full. Unless you have a reason to choose differently, you should select the option to wrap. See *Chapter 6. Troubleshooting.*

Transfer Options - These options allow you to select the events that can cause a transfer of control. If one of these events occurs and the option is enabled, the RPM will switch to the Standby PLC, provided the Standby is operational. The effect of enabling or disabling any particular parameter in this group is indicated in the following table.

> *Note:*
>
> *Switchover will be inhibited for all events when the Switchover Inhibit feature is enabled. See Chapter 5. PLC Programming, PLC Control Bits section for Switchover Inhibit information.*

Parameter	Operation
PLC I/O Update This event should be enabled during normal operations. It may be disabled for testing and troubleshooting.	**Enabled:** *Active PLC* - If the RPM Module does not detect an I/O Update from the Active PLC within the timeout period, it will transfer control to the Standby PLC if the Standby is operational. *Standby PLC* - If the RPM Module does not detect an I/O Update from the Standby PLC within the timeout period, it will declare the Standby PLC as **not** operational. Transfer to the Standby PLC will be inhibited. **Disabled:** *Active PLC* – The RPM module will not automatically transfer control to the Standby PLC even if the Active PLC fails to update the I/O. *Standby PLC* - An I/O update timeout on the Standby PLC will not prevent RPM module from transferring control to the Standby PLC based on other events.
Heartbeat This event should be enabled if you wish to switch control when the Active PLC is placed in Program mode or to prevent switching to the Standby PLC that is in Program mode. Note that PLC logic is required to generate the Heartbeat.	**Enabled:** *Active PLC* – If the RPM Module does not detect a Heartbeat from the Active PLC, it will transfer to control the Standby PLC, if the Standby is operational. *Standby PLC* – If the RPM Module does not detect a Heartbeat from the Standby PLC, the RPM module will declare the Standby PLC as **not** operational. Transfer to the Standby PLC will be inhibited. **Disabled**: Active PLC - The RPM module will not automatically transfer control to the Standby PLC even if the Active PLC Heartbeat is missing. Standby PLC – A missing Standby PLC Heartbeat will not prevent RPM module from transferring control to the Standby PLC based on other events.

Parameter	Operation
PLC Fatal Error Monitor This event will cause a switchover if the active PLC reports a Fatal Error. It should normally be enabled.	**Enabled:** *Active PLC* - If Active PLC status word indicates a fatal error, the RPM module will transfer control to the Standby PLC, if the Standby is operational. *Standby PLC* - If the Standby PLC status word indicates a fatal error, then the RPM module will declare the Standby PLC as **not** operational. Transfer to the Standby PLC will be inhibited. **Disabled:** *Active PLC* - The RPM module will not transfer control to the Standby PLC when a fatal error is detected. *Standby PLC* - A fatal error reported by the Standby PLC will not prevent the RPM module from transferring control to the Standby PLC based on other events.
Checksum Match You may choose to enable this option if you wish to prevent manual switching to a Standby PLC whose program is different from the Active PLC.	**Enabled**: Inhibits manual switchover if Program (L) checksums in the Active and Standby PLCs do not match. **Disabled**: Does not inhibit manual switchover, even if the checksums do not match. Note: Checksum match can always be monitored by the PLC. The home base provides a discrete input that indicates whether the checksums match. See *Status Monitoring Inputs* on Page 35.
PLC Logic Request	**Enabled:** Allows PLC logic to request switchover. *Note: Switchover will be inhibited if the Standby PLC is not operational.* **Disabled**: Prevents PLC logic from controlling switching.
Pushbutton Request	**Enabled:** Allows front panel pushbuttons to request switchover to the standby PLC. *Note: Switchover will be inhibited if the Standby PLC is not operational* **Disabled**: Prevents pushbuttons from controlling switchover.

Configuring I/O Bases

The Configuration program allows you to specify the real I/O bases and the Virtual bases. All bases must be identified to the module. Data from the real bases is used to update the Standby PLC I/O table. The Virtual Bases are used to transfer data from one PLC to the other.

Base Configuration - Allows the user to specify the remote base configuration. Up to 15 bases can be configured. For each possible base the user may specify whether:

1. The base is not in the configuration (not present).

2. The base is a Real Base, which the RPM will mirror to the Standby PLC. You must specify all real bases that are present in the system configuration so that the Standby PLC will receive all updates.

3. The base is a Virtual Base used for monitoring and controlling RPM operation and for transferring critical data. For each virtual base, you will select the number of virtual modules in the base. The virtual word modules are used to transfer the critical data from the Active PLC to the Standby PLC.

One Virtual Base must be configured as Home Base. The Home Base will contain two virtual discrete modules, one 32 X and one 32 Y. These discrete points can be accessed by PLC logic to monitor and control the RPM module. Home Base may also contain up to 14 virtual high-density word modules (combination 32X/32Y).

Note:

Since each Virtual Base adds to the scan time (just like a real base), you should configure the minimum number required for transferring data.

If you need to transfer more data than the Home Base will support, you may configure additional Virtual bases. Each additional Virtual Base may contain up to 16 virtual high-density word modules. When creating additional Virtual Bases, ensure that you have a block of unused words in the PLC I/O table that can be assigned to the word modules.

NOTE:
After changing the Virtual base configuration, you must take the following steps before the changes will take effect:

1.   Reset the RPM module by clicking on the Module Reset icon in the toolbar of the 2541 Configuration program.

2.   Manually press the Resume pushbutton on the module front panel.

2.7. Configuring PLC I/O

Note:

You must configure all Virtual Bases in the RPM module before completely configuring the PLC I/O. Configuration is necessary so that the PLC can detect that a (virtual) base is present.

Once the Virtual Bases have been created in the RPM, you can configure the PLC I/O table. Since RPM Virtual Bases appear to the PLC as real I/O, configuration can be accomplished using the standard PLC programming software. All bases, both real and virtual, must be configured. Ensure that all virtual modules within virtual bases are configured accurately.

Note:

The PLC must be the Active PLC (communicating directly with the I/O), before you can properly configure the PLC I/O.

The PLC being configured must be communicating directly with the I/O through the RPM module; that is, it must be the Active PLC. If the PLC is the Standby PLC, you can use the RPM module pushbuttons to change its role to the Active PLC. Once you have configured one PLC, you must create an identical I/O configuration in the other. To accomplish this task, you could switch the other PLC to Active and repeat the configuration process. A more efficient method may be to simply save the configuration from the first PLC and load it to the other.

2.8. Module Checkout

Once the module has been installed and the I/O configured, you can test the operation of the module. To simplify testing, disable all transfer options except I/O Timeout and Pushbutton transfer. If any of the following tests do not produce the correct results, see

Startup Test

Cycle Power to the Module. After the module completes a Power-On-Self-Test, the MODULE GOOD LED should light. The PLC that you designated in the configuration program as Startup PLC should be selected; its ACTIVE LED should be on solid. The LED for the other PLC should be blinking slowly, indicating that it is available for transfer.

Manual Transfer Test

Use the recessed pushbuttons to transfer control to the Standby PLC. Switchover to the Standby PLC should take place without bumping the I/O.

> *Note:*
>
> *If the pushbuttons are not effective, you may need to enable pushbutton transfer using the RPM Configuration Program.*

Automatic Transfer Test

Remove the I/O cable from the Active PLC. The RPM should transfer control to the Standby PLC. Repeat the test with the other PLC.

Heartbeat Test

Ensure that your PLC logic has been modified to generate the Heartbeat pulse and that both PLCs are in Run mode. Enable the Heartbeat option in the Transfer Options section of the Configuration program. If you place the Active PLC in program mode, the RPM should switch to the Standby PLC.

PLC Logic Transfer Test

If you wish to test whether PLC logic can be used to transfer control, enable the PLC Logic option in the Transfer Options section of the Configuration program. You should be able to set the appropriate Home Base Y output from the Active PLC to cause a switch to the Standby PLC. See *PLC Control Bits* on page 37.

CHAPTER 3. OPERATION

3.1. Front Panel

RPM Module Status and Control
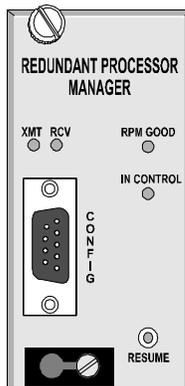
At the top of the front panel is a configuration port for the module. Using a DB9 connector, the configuration port provides a subset of the standard PC/AT RS-232 electrical interface. A computer using CTI-supplied software can be connected to the port to set the module operating parameters and to perform diagnostic operations.

The XMT and RCV LEDs light when data is transmitted or received through the configuration port. The RPM GOOD LED provides module status. The IN CONTROL LED indicates whether I/O update monitoring is enabled or disabled.

The recessed RESUME pushbutton can be used to restart the module after a diagnostic event has caused the module to go to a fail-safe condition. If the module is operating normally, this button has no function.

PLC Status and Transfer
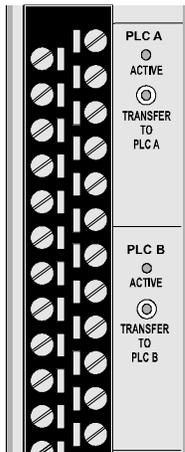
This section of the front panel displays the condition of the PLCs and allows the user to manually transfer control between them. There are two groupings, one for the PLC connected to Port A (PLC A) and the other for the PLC connected to Port B (PLC B).

The ACTIVE LED indicates the status of the PLC attached to the port. Based on the condition of the LED, you can determine whether the PLC is Active or Standby and whether it is operational.

The TRANSFER pushbutton allows you to manually transfer control of the I/O to the PLC connected to the port. For example, to request a switchover to PLC A, you would press the button in the PLC A group. Transfer will take place only if the RPM is configured to enable the front panel pushbuttons and the Standby PLC is operational.

Backup RPM Status and Control

This section is reserved for future use.

LED Indicator Functions

LED	Function (Normal Condition in Bold)
XMT	Lights when the RPM transmits a response to the PC configuration program
RCV	Lights when the RPM receives a request from the PC configuration program
RPM GOOD	**ON** – The module is operational OFF – A module fault has been detected
IN CONTROL	**ON** – I/O update monitoring is enabled. Blinking – "Switchover Inhibit" mode is enabled **OFF** – I/O update monitoring is disabled
ACTIVE	**ON SOLID** – The PLC attached to this port is the ACTIVE PLC and is Operational. BLINKING RAPIDLY – The PLC attached to this port is the ACTIVE PLC but is not Operational. *Note: This condition can occur when the Standby PLC is also not operational and thus not available to take over control of the I/O.* **BLINKING SLOWLY (1 Hz)** – The PLC attached to this port is the STANDBY PLC and is Operational. OFF – The PLC attached to this port is the STANDBY PLC but is not Operational.
BACKUP RPM	Reserved for future use.

Front-Access Termination Connector

To simplify terminating cables, the RPM module uses a standard Siemens® front-access connector. The remote I/O ports of the both PLCs and the cable from the remote I/O are connected to the appropriate terminals on this connector. In addition, nine additional terminal sets are available for switching other low-level communications signals. These may be used to switch operator interface devices between the PLC communication ports. Labels on both sides of the front access connector indicate the connection of each terminal. See *Appendix A* for wiring diagrams.

3.2. Normal Module Operation

Initial Startup

When power is applied, the module will begin a self-diagnostic test. During this test, the RPM Good LED will be off and the I/O will be connected to the PLC selected by the Fail-Safe switch. Other LEDs will be cycled on and off to allow an operator to visually confirm that the LED will light. Once the power on test is complete, the module will de-energize the Fail-Safe relay, transferring control of I/O switching to the module firmware. You may hear the relay click when this event occurs. At this time the RPM GOOD LED will be illuminated and the RPM will select the INITIAL STARTUP PLC established during module configuration as the Active PLC.

Normal Control Condition

The Active LED for one PLC should be on solid, indicating that it is the Active PLC (in control of the I/O) and it is operational. The Active LED of the other PLC should be blinking slowly, indicating that it is the Standby PLC and is operational.

Note:

The conditions used to evaluate whether a PLC is operational are established by the user using the 2541 Configuration program.

Using the PLC programming software, you should be able to observe the state of the real I/O and the Virtual Bases from both the Active and the Standby PLCs. The I/O values in both PLCs should agree. Neither PLC should be experiencing an I/O timeout.

If you are using Virtual bases to transfer critical data, the data transferred by this method should be the same in both PLCs. If you are using V memory transfer, this data may not be the same at a particular time, because it can take several scans to transfer the entire V memory block. At some time, a particular value should be the same in both PLCs. By observing the V memory transfer complete bits toggle, you can verify that V memory transfer is taking place and estimate the amount of time required to transfer the entire V memory block.

Switchover

When a fault is detected in the Active PLC, when the transfer is requested by PLC logic, or when the operator requests switchover via pushbutton, the module should transfer control to the Standby PLC. When this happens you may hear the RPM relays click. The switchover should occur without bumping the I/O or causing a remote base to time out. The LEDs should indicate the new control state (see LED section above). Note that switchover will not take place if a particular event (such as a button press) is disabled. The module logs every event whether or not it causes a switchover to occur. The last 63 events can be observed using the 2541 Configuration program. See Section *6.4. RPM Diagnostic Display* on page 50.

3.3. Operational Problems

Configuration Errors

Errors in configuring the RPM module may cause operational problems. For example, setting the I/O Update Timeout to a value less than the PLC scan time will cause the module to continuously cycle control between the PLCs. Should you observe unusual operation, you should refer to the Troubleshooting chapter in this manual.

Programming Errors and Omissions

Because the PLC processor scans are not synchronized, special programming techniques may be required to prevent I/O transitions that might disrupt the process during switchover. If you are encountering process disruptions, please refer to the Chapters on Switchover Considerations and PLC Programming.

Diagnostic Failure

If the module diagnostic routines detect an error, the RPM GOOD LED will be turned off and the watchdog timer circuit will be allowed to reset the module. The Fail-Safe relay circuit will connect the I/O to the PLC selected by the Fail-Safe switch. The hardware will hold this state until the RESUME pushbutton is pressed or power to the module is cycled. If the diagnostic failure condition persists, you should replace the RPM module.

CHAPTER 4. SWITCHOVER CONSIDERATIONS

Because the PLC scans in a system using the RPM module are not synchronized, the system designer needs to take this into account. This chapter identifies factors that should be considered in designing a backup system using the RPM module.

4.1. PLC Control Characteristics

PLC Scan

Simatic® 505 controllers operate on a scan cycle in which I/O is read, logic is solved, and outputs are written. These PLCs allow multiple types of scan operations within the single PLC scan, including Interrupt RLL, Cyclic RLL, Analog Scan, and Discrete Scan. See the *Siemens® SIMATIC® 545/555/575 Programming Reference Manual* for complete details.

Interrupt RLL may read data at any time from Interrupt I/O. Similarly, Cyclic RLL can read inputs or write outputs at any time using Immediate I/O instructions. Interrupt I/O or Immediate I/O function only with modules in the local base.

Data for the Discrete and Analog scans are obtained at one particular point in the scan cycle, Normal I/O Update. Since the RPM module switches only remote I/O and since remote bases do not support Interrupt I/O or Immediate I/O, systems using the RPM module must access all I/O access during Normal I/O Update. It is important to understand this condition when evaluating backup systems using the module.

The figure below illustrates the Main RLL / Analog scan cycle. I/O Update refers to writing discrete (Y) and word (WY) outputs and reading discrete (X) and word (WX) inputs. SF Access refers to reading and writing data using task codes or other special function protocols.

Note that I/O Update and SF (Special Function) Module Access do not occur while logic processing (Main RLL and Analog) is executing. Logic is solved based on the state of the inputs read during the prior I/O update. Outputs are not set until the I/O update following the logic processing.
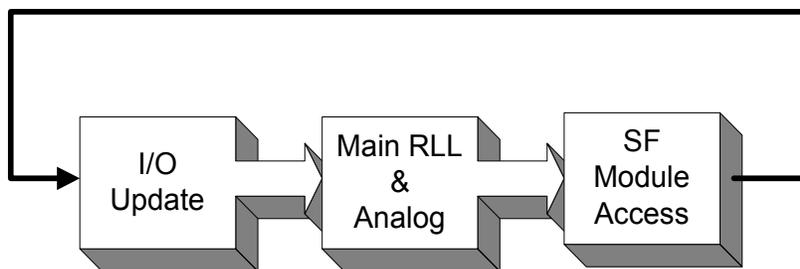
Figure 9. PLC Scan Cycle

Asynchronous Operation

Unlike systems explicitly designed for hot backup operation, the program scans of the two SIMATIC® 505 processors in a warm backup are not synchronized. The PLCs will probably start their scans at different times and may require a different amount of time to complete scans. Therefore, one PLC may be updating I/O while the other PLC is solving logic. See Figure 10 below. Note that SF Access has been omitted from the diagram for simplicity. Although SF Access does take place at the end of each scan, the RPM module does not mirror SF requests.
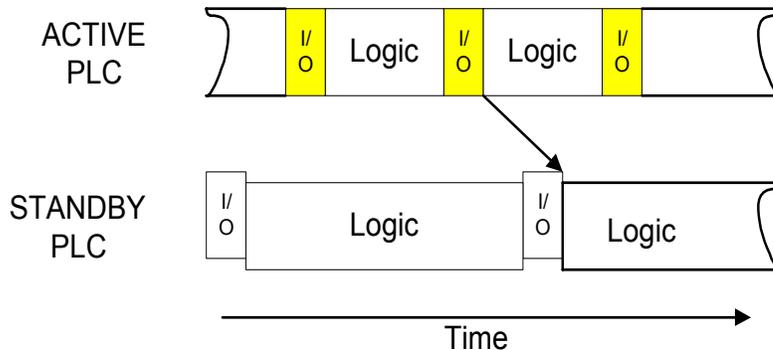
Figure 10. Asynchronous Scan Example

In a backup system using an RPM module, the Standby PLC does not read directly from the I/O. Instead it reads a cached copy of the data (Mirror I/O) which has previously been read by the Active PLC. Consequently, the Standby PLC will always be reading "old" data and will never be reading ahead of the Active PLC. This attribute simplifies the task of preventing program divergence.

Switchover Timing

Although the RPM module can switch control of the I/O in less than 50 ms, the total amount of time required for the Standby PLC to assume control includes other elements. See Figure 11 below.
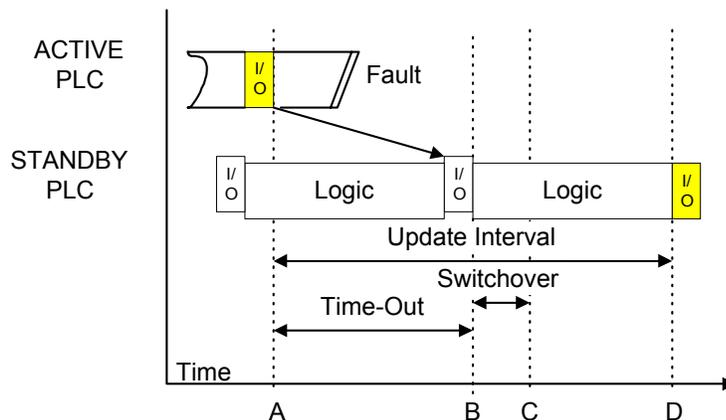
Figure 11. Switchover Timing Illustration

Figure 11 illustrates switchover based on an I/O update timeout. Prior to time A, the Active PLC is reading and writing the Remote I/O. Meanwhile the Standby PLC is not actually controlling the I/O but is reading a copy of the I/O values obtained by the Active PLC (Mirror I/O).

Some time after the Active PLC has completed an I/O Update (time A), it encounters a fault which causes it to cease updating the I/O. Concurrently, the Standby PLC continues to read the Mirror I/O, which contains the last data read by the Active PLC.

After a specified time elapses without another I/O update from the Active PLC, the RPM begins switching control of the I/O to the Standby PLC (time B). At time C, the Standby PLC is connected to the Remote I/O and at time D, this PLC updates the I/O. The outputs will be set based on the I/O data obtained by the Active PLC at time A.

The net effect is to delay setting the new outputs by a period equal to the sum of the Timeout Value plus the Switchover Time plus the remaining time before I/O update in the Standby PLC. The diagram illustrates the worst-case scenario in which switchover takes place just after the Standby PLC has completed an I/O update to the Mirror I/O. In this case, the total delay is approximately equal to the timeout period plus one additional PLC scan. As long as this delay does not cause an I/O timeout, the effect on the process will most likely be insignificant.

Since the total elapsed time cannot exceed 500 ms without incurring an I/O timeout and the timeout period must be greater than the maximum scan time, the largest scan time that can be accommodated without I/O timeout is less than 250 ms.

Note:

When the I/O timeout must be set to a value greater than 225 ms, you should consider configuring the I/O bases to hold last value upon timeout.

4.2. Preventing Process Disruptions

When implementing a warm backup system, it is important to understand that bumpless transfer is not automatically guaranteed under every condition. The system designer must make sure that the system is configured and properly programmed to avoid process excursions during switchover.

Communications Problems

Remote Base Timeout

Remote I/O bases require that an update message be received from the PLC within 500 ms. If the I/O is not updated within this period, the Remote Base Controller (RBC) will set outputs to a predetermined condition. Depending upon how the RBC is configured, the outputs will be set to zero (off) or will hold the last state. If the RBC is configured to set outputs to zero upon RBC timeout, the outputs will transition to zero and then back to the value determined by the PLC after it regains communications. For most applications, you will want to avoid this I/O "bump".

As long as the I/O Update Timeout can be set to 225ms or less, the RPM module will prevent the timeout automatically. If the I/O update must be set to a larger value, you must set the RBC to hold last value to avoid the possibility of bumping the I/O on timeout.

Standby PLC I/O Timeout

If a PLC does not receive a reply from a Remote Base Controller within a specific period, it will log out the I/O rack. When this happens, the related inputs (and derived outputs) will be set to zero while the PLC attempts to re-establish communications with the base. If the Standby PLC were to erroneously log out the remote bases, the I/O would bump when control was transferred to the Standby PLC.

To resolve this problem, the RPM module mirrors the I/O to the Standby PLC, allowing it to function as if real I/O updates were successfully completed. Thus, under normal conditions the Standby PLC will not log out the remote bases. Should an actual RBC fail to respond to the Active PLC within the timeout period, the Active PLC will log out the base.

The RPM will also stop responding to the Standby PLC after a user configurable interval (RBC Offline Declaration), which will cause the Standby PLC to log out the corresponding mirrored I/O base. In setting the RBC Offline Declaration, you should ensure that the value is greater than the I/O Update timeout period. Typical values will be 10-20% greater. When in doubt, a larger number is the safer selection.

Note:

Performing on-line edits to the Active PLC can cause the 2541 RPM module to switch to the Standby PLC. When edits or recompiles are made to the Active PLC, the PLC can extend its scan time significantly. If the scan time exceeds the I/O Update Time configured in the RPM, the RPM will switch over to the Standby PLC.

Program divergence

Divergence refers to a condition in which identical PLC programs make different decisions relative to the state of the output. Programs may diverge because:

- The PLCs read different values for the same set of inputs.

- A particular input transition is missed by the Standby PLC.

- Operator changes to values in the Active PLC are not reflected in the Standby PLC. For example, a SCADA system operator might change a setpoint or download a new recipe.

- Internal processes drift apart over time.

- Sections of the program in the Standby PLC are skipped. This programming technique may be used to ensure that the scan time of the Standby PLC is shorter than that of the Active PLC. See the following sections for reasons to implement this technique.

The effect of program divergence may or may not cause a disruption in the process being controlled when switchover occurs. If the divergence results in significant differences in critical outputs, then the I/O will "bump" (rapidly transition) when the transfer occurs.

Reading Different Input Values

The design of the RPM module mirrors I/O replies to the Standby PLC and employs a timeout delay before automatically switching control. These features minimize the possibility that, during automatic switchover, the PLCs will bump the I/O because different input values were read. However, this can still occur when the Standby PLC scan time exceeds the Active PLC scan time and control is manually transferred (pushbutton or PLC logic). See Figure 12 below.
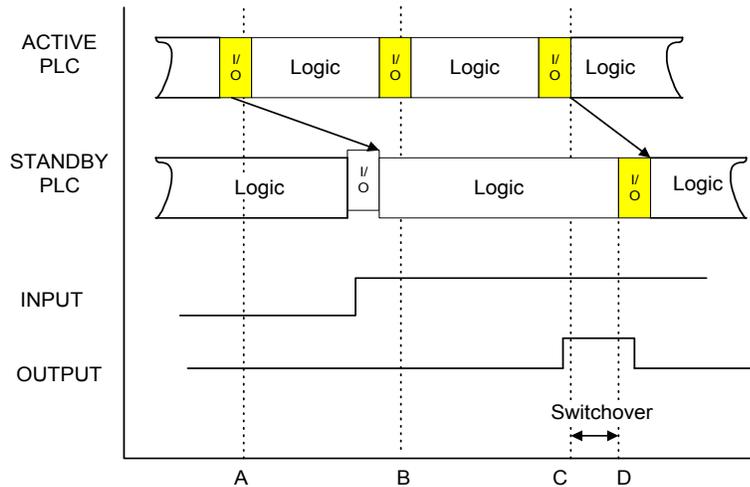
Figure 12. I/O Bump Induced by Different Inputs

Assume PLC logic sets an output on whenever a particular input is on. At time A, the Active PLC performs an I/O update, which reads the input as off. Input data is mirrored to the Standby PLC, which reads this data prior to time B. At time B the Active PLC reads the input as On and at time C sets the output high. Immediately following the I/O update, control is transferred to the Standby PLC. The Standby PLC, whose I/O update read the input as off, sets the output to off. The result is a rapid transition of the output when switchover occurs.

You may eliminate this situation by simply ensuring that the Standby PLC scan is always less than the Active PLC scan. *Chapter 5. PLC Programming* describes programming techniques that may be used to shorten the Standby PLC scan.

Missing an Input Transition

In certain cases, it is possible to miss an input transition. This can occur when the Standby scan exceeds the Active PLC scan and an input event does not persist long enough or a latching input module is used. Latching input modules hold the input until it can be read by the PLC and clear the input after the input is read.
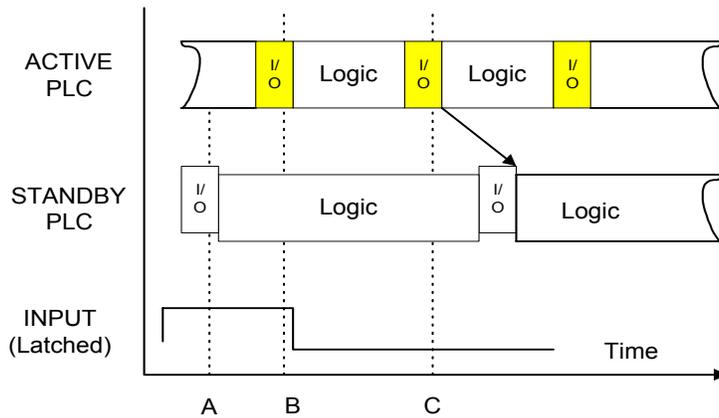
Figure 13. Missing Latched Input

In Figure 13 above, the Standby PLC performs an I/O update that reads the mirror I/O. This reflects the state of the latched input at some time in the past when the input was off. At time B the Active PLC reads the latched I/O module as high. This event causes the latched input to be cleared. At time C, before the Standby performs another I/O update, the Active PLC reads the latched input as low. When the Standby PLC performs a second I/O update from the mirror I/O, it reports the input as low. Thus, the Standby has missed the transition. Missed events of this type, especially when used to trigger events such as drum sequencing or counter increments, will cause the programs to diverge.

Again, the solution to this situation is to ensure that the Standby PLC scan time is less than the Active PLC scan time.

A similar situation can be created when an input does not remain in a particular state long enough to be read by both PLCs. Generally, ensuring that unlatched inputs are on for several PLC scans is considered good system design, even in single processor applications. Consequently, you may not need to make any changes to your system. Nevertheless, you should take the time to review your system to ensure that inputs are on long enough to be read during a switchover event.

Changes via Special Function Modules

Changes to PLC variables by Special Function modules or Program Port communications are not mirrored to the Standby PLC. These updates are typically made by Operator Interface Devices, programming applications, or SCADA programs using serial or network communications. For example, a supervisory control station may change a setpoint or a V memory location.

If these changes are not transferred to the Standby PLC, then the programs may diverge. Whether the changes are transferred in one scan (Critical Data) or over several scans (Normal Data) depends upon how the data value affects the process. Events, such as setpoint changes, that cause a slow change in the process can be transferred over several scans. The decision also depends on how much risk you wish to assume that the Active PLC will fail before it can complete the data transfer. Transferring the data in one scan will minimize this risk. Note that there is always a possibility, however slight, that the Active PLC will fail before it completes that particular scan.

32

Differences in Internal Processes

Certain PLC functions are not related to the PLC scan. These functions are PLC clock-time dependant. For example, timers are often used to control process activities. Over time, the timer values will tend to drift apart, since each PLC clock is slightly different. Certain special function calculations may exhibit similar characteristics. In addition, when the PLC is placed in program mode, its internal processes may diverge from the other PLC.

It is good practice to transfer data related to these processes from the Active PLC to the Standby PLC. Critical Data (data that could cause a rapid process change) should be transferred every scan. Non-Critical Data can be transferred over multiple scans. See *Chapter 5. PLC Programming* for details in implementing data transfer.

Forcing I/O

The RPM module does not automatically transfer force tables from the Active to the Standby processors. If switchover occurs while I/O is forced in the Active PLC, forces will not be carried over and the outputs will change.

Caution

To avoid potential process problems on switchover, CTI recommends that you not force I/O during normal operations. If you must force I/O for diagnostic and startup activities, you should inhibit switchover while you are performing the work, then clear forces before returning control to the RPM.

CHAPTER 5. PLC PROGRAMMING

This chapter will outline the programming techniques that may be used with a backup system using the RPM module. This chapter will present information about:

- Monitoring status inputs

- Controlling RPM switching

- Generating the Heartbeat

- Skipping sections of the PLC program

- Data Transfer

See *Chapter 4. Switchover Considerations* for reasons behind skipping sections of the program and transferring data.

5.1. Monitoring and Controlling RPM Status

Status Monitoring Inputs

The Home Base provides a group of bits which PLC logic can use to monitor the status of the RPM module and attached PLCs. The following table describes the use of these bits.

Offset	Description
0	PLC A has active control (1 = Yes, 0 = No)
1	PLC B has active control (1 = Yes, 0 = No)
2	PLC Active Flag (1 = this PLC is active, 0 = this PLC is standby)
3	PLC A Status (1 = Operational, 0 = Not Operational)
4	PLC B Status (1 = Operational, 0 = Not Operational)
5	Checksum Compare (1 = Checksums match, 0 = Checksums do not match)
6 – 8	Reserved
9	Non-Critical Data Transfer Completed - Alternates with offset 10
10	Non-Critical Data Transfer Completed - Alternates with offset 9
11	RPM has control (reflects I/O update monitoring status, 1=enabled,0=disabled)
12	Reserved for future use
13	RPM module is good (operational)
14-15	Reserved for future use
16	I/O Update Monitor Enable (1=Enabled, 0=disabled)

17	Heartbeat Monitor (1=Enabled, 0=disabled)
18	PLC Fatal Error Monitor (1=Enabled, 0=disabled)
19	PLC Logic Transfer (1=Enabled, 0=disabled)
20	Program Checksum Match (1=Enabled, 0=disabled)
21	Front Panel Pushbutton Transfer (1=Enabled, 0=disabled)
22	Switchover Enabled
23 - 31	Reserved for future use

Offset 0 - 1   *PLC Control Status* – The bits indicate which PLC is actually controlling the process (i.e. connected to the I/O). Under normal conditions, one of these bits should be on. If both bits are off, it indicates that the PLC is not communicating with the RPM module. See *Chapter 6. Troubleshooting*.

Offset 2   *PLC Active Flag* – This bit indicates whether this PLC is the Active PLC (actually controlling the process). The RPM presents this bit to the other PLC in the opposite state. For example, if the bit were 1 this PLC would be the Active PLC. The RPM would present this bit as 0 to the other PLC, indicating that it is the Standby PLC. PLC logic can use this bit to condition Critical Data transfer.

Offset 3 – 4   *PLC Operational Status* – These bits indicate whether the PLCs meet the allowable conditions for controlling the process. The allowable conditions are determined by the user during module configuration. The user may select to monitor PLC I/O Update, PLC Heartbeat, and/or PLC Fatal Error Status.

Offset 5   *Checksum Compare* – This bit indicates whether the program (L) memory checksums are the same in both PLCs. This bit is used to determine whether both PLCs contain identical programs.

Offset 6 – 8   *Reserved* – These bits are reserved for future enhancements.

Offset 9 – 10   *V Memory Data Transfer Completed* – These bits toggle every time a V memory data transfer is completed. They may be observed to verify that data transfer is taking place. PLC logic may condition logic based on the complete transfer of a block.

Offset 11   *RPM has control* – This bit indicates that the RPM has control over PLC transfer. Should always be a 1.

Offset 12   *RPM Control Status* – Reserved for future use.

Offset 13   *RPM Module is Good* – This bit indicates that the RPM module is operational.

Offset 14 – 15   *Reserved*

Offset 16   *I/O Update Monitor Enable* – This bit indicates whether the RPM module is monitoring PLC I/O Updates.

Offset 17   *Heartbeat Monitor Enable* – This bit indicates whether the RPM module monitors the active PLC for a 'Heartbeat' signal.

Offest 18   *PLC Fatal Error Monitor* – This bit indicates whether the RPM module monitors the active PLC for Fatal Errors.

36

Offest 19   *PLC Logic Transfer Request* – This bit indicates whether the RPM module is enabled to accept transfer request from the active PLC's logic .

Offest 20   *Checksum Match* – This bit indicates whether the RPM module performs a checksum match of the Active PLC and the Standby PLC logic programs.

Offest 21   *Pushbutton Transfer Request* – This bit indicates whether the RPM module is enabled to accept a PLC transfer request from the module's front panel pushbuttons.

Offest 22   *Switchover Enable* – This bit indicates whether the RPM module is enabled to transfer from the active PLC to the standby PLC based on module configuration. Note: Switchover Enable means Switchover Inhibit is NOT enabled. Switchover Inhibit overrides any PLC transfer condition set in the configuration program.

Note:

To determine the actual bit number of the item, add the offset value to the starting X value configured for the Home Base. For example, if the Home Base discrete Inputs were configured to start at X33, then the bit containing the PLC Active Flag would be X35.

PLC Control Bits

The following bits can be used by PLC logic to control the operation of the RPM module.

Offset	Description
0	Request Switchover to PLC A
1	Request switchover to PLC B
2	Heartbeat bit. PLC Logic must transition this bit every scan
3	Switchover Inhibit
4 - 31	Reserved for future use

Offset 0 - 1   *PLC Switchover Request* – PLC can use these bits to request that the RPM transfer control to a particular PLC. Although logic can request a transfer, the RPM module will perform the switchover only if the PLC is operational and the user has configured the RPM module to allow PLC logic switchover requests.

Offset 2   *Heartbeat* – This bit is used by the RPM module to monitor for a "heartbeat" generated by PLC logic. The heartbeat is used to determine whether the PLC is in Run mode. PLC logic generates the heartbeat by toggling this bit every scan.

Offset 3   *Switchover Inhibit* – This bit inhibits the RPM from transferring control to the Backup PLC while it is enabled. This feature overrides all transfer settings enabled using the 2541 Configuration program. The "In Control" LED on the module front panel blinks (1 HZ rate) when the Switchover Inhibit mode is enabled.

5.2. Generating the Heartbeat Pulse

The RPM module can be configured to use the heartbeat pulse to detect whether a PLC is in Run Mode (actively processing logic) or is in a Program mode (where logic processing is suspended). Note that the PLC logic processing state cannot be determined by simply monitoring the I/O update; the PLC continues to communicate with the I/O even in program mode. The user can select whether the heartbeat will be evaluated to determine if the PLC is operational.

If heartbeat monitoring is enabled, you must include logic to toggle the designated output every scan. The figure below illustrates the ladder logic required to implement the heartbeat. This rung must be inserted in a section of the program that is always operating (outside any SKIP/MCR area).

Figure 14. Heartbeat Generation Logic

5.3. Skipping Sections of the Program

As noted in the chapter on Switchover Considerations, many potential switchover problems can be eliminated by ensuring that the scan time of the Standby PLC is always less than the Active PLC. You can accomplish this by skipping sections of the program when the PLC is the Standby PLC. You can condition the skipping logic based on the PLC Active Flag bit. See *Section 5.1* above.

The following example illustrates ladder logic that skips a section of code. In this example, the module status words are assumed to start at X1. Note that in addition to using the Active Flag (X3) to condition the logic, you must also monitor the RPM Control Status Bits (X1 and X2). This is necessary to handle the power failure condition.). Under normal conditions, one of these bits should be on. If both bits are off, it indicates that the PLC is not communicating with the RPM module. If power to the rack containing the RPM module is lost, the module will select an Active PLC based on the position of the Fail-Safe switch. However, since the RPM module processor is not operating, the PLC will read all status bits as off (0). If you were to use only the Active Flag, the PLC would continue to skip the logic. By also using the Control Status bits in the logic, you can ensure that the logic is not skipped in this case.

```
      X1      X3
   ┤ ├──┤ ├──┤/├──────────────────────────────────────────────(SKPn)
      X2      │
   ┤ ├────────┘
.
.
.
Main Program
.
.
.
   ┌──────────────────────────────────────────────────────────(LBLn)
   │
```
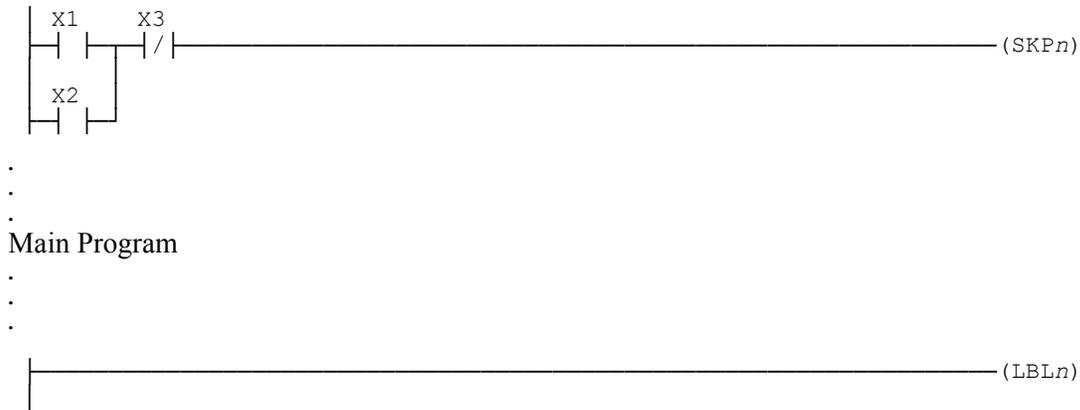
Caution:

When you skip sections of the program in the standby PLC, you must transfer all values which the section influences every scan. For example, if the section normally writes outputs, then you must transfer the corresponding outputs from the Active PLC. This is necessary to ensure that the proper output values are written to the I/O when the Standby PLC first assumes control.

5.4. Transferring Data between PLCs

Many applications will require data to be transferred from one PLC to the other to prevent program divergence. The RPM module provides two methods of transferring data between the PLCs, V-Memory Transfer and Virtual Base Transfer.

Data Transfer Direction

When transferring data using the V Memory Transfer or Virtual Base Transfer methods, the logic must detect whether the PLC is the Active PLC or the Standby PLC. As indicated in *Status Monitoring Inputs* on page 35, the RPM module provides an ACTIVE FLAG bit that indicates to the PLC whether it is the Active or Standby PLC. The bit is presented differently to each PLC. The Active PLC will read the bit as On while the Standby PLC will read the bit as Off. This bit can be used to condition PLC logic as follows:

- When the ACTIVE FLAG is On, the PLC copies data to the RPM Virtual Base output (WY) locations and, if V Memory Transfer is being used, to the user-designated V memory block.

- When the ACTIVE FLAG is Off, the PLC copies data from the RPM Virtual Base input (WX) locations and, if V Memory Transfer is being used, from the user-designated V memory block.

V Memory Transfer Method

V Memory Transfer requires minimal PLC logic but may require many scans to transfer all data. The transfer rate depends upon the scan time of the PLC and I/O activity; typical rates are 50-100 words per second. V Memory Transfer is typically used to transfer Non-Critical data that changes very slowly. The types of data transferred may include:

- Loop tuning constants,

- Alarm setpoints,

- Timer presets (assumed to be static),

- Counter presets (assumed to be static),

- Drum presets.

To use this method, the RPM module is configured to automatically transfer a specified block of V memory. See *Section 2.6. Configuring the RPM Module* on page 15.

Virtual Base Transfer Method (Critical Data)

The Virtual Base transfer method must be used to transfer Critical Data that must be transferred every I/O cycle. Critical Data may include:

- Output image registers (applicable to skipped sections of logic),

- Logic step transitions resulting from timer, drum, counter operations,

- Drum or sequencing instructions/step locations which may rapidly transition,

- Latching logic associated with logic transitions.

The Active PLC writes data to the RPM virtual base WY locations; the Standby PLC reads data from the corresponding WX locations and copies the data to the applicable data elements. This transfer method will move all critical data each time. Its primary disadvantage is the need for potentially large virtual base addressing requirements.

Note that the Active Flag bit is needed to condition the logic so that the programs can be identical in both PLCs. Depending on the state of the Active Flag, data will be written to the Virtual Base WYs (Active PLC) or read from the Virtual Base WXs (Standby PLC). Note also that the Control Status Bits (X1 and X2) are used to ensure detection of a module fault or power loss condition.

See *Critical Data Transfer* section of *Appendix B* for logic examples of transferring various data elements using the Virtual Base transfer method.

Virtual Base Transfer Method (Indexed)

The Indexed Transfer method is an alternative to the V Memory Transfer method for transferring Non-Critical data. This method is faster than V Memory transfer while consuming much less Word address space than normal Virtual Base transfer. For data items that should be updated in ten scans or less, you should use the Indexed Transfer method. These variables may include:

- Loop setpoint / bias / output values,

- Loop operational/status values (manual, auto, cascade data flags).

This method mimics the techniques used in the Siemens® Peer-link data transfer where a block of data is moved from the Active to the Standby PLC in multiple segments. The first address in each segment contains the transferred block's relative address within the complete table. One segment is transferred each scan until the entire block is moved. Once the move is completed, the process is repeated. Note that the Active Flag and Control Status bits are used to determine the transfer direction.

Monitoring Remote Base I/O Failures

For diagnostic testing, you may wish to determine the number of times a remote base failure has been detected. Status word STW02 contains bits that reflect the status of each I/O base. Simple logic can be used that will monitor this status word (looking for a non-zero value) and record the number of times an I/O base failure has occurred. The following figure illustrates logic that is more elaborate that may be employed to identify the faulty base by masking the individual bits and latching failure locations. This example tracks failures for base 1. Repeat these rungs, changing the N value in the BITP for each base to be monitored. See *Appendix B,* under *PLC Status Logic* for STW2 bit mapping. This diagnostic is useful in situations where switching is noted and may be caused by the remote I/O link failure influencing base-log on times in the Active PLC.
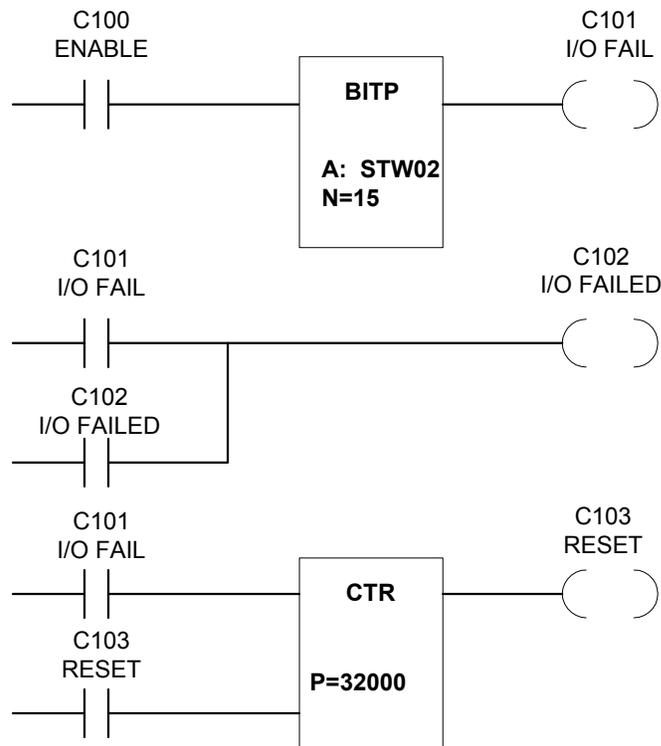
Figure 15. Remote I/O Base Failure Counter

5.5. Instructions That Could Cause Switchover Problems

Some instructions, when used with a RPM backup system, may operate unpredictably when a switchover occurs if you fail to observe certain programming considerations. For example, consider transferring the control element and data element for instructions that are also being executed in the standby processor.

Instructions that should be observed closely include:

- timer
- counter
- drum / matrix compare / sequence operations.

Timer Instructions

Timer instructions should be updated by transferring the TCP and TCC values and latching the timer outputs rather than directly using the timer output in logic thereby allowing timeout transitions to be more readily captured by the standby processor. The timer programming should transfer the counter's accumulated value from the active to the standby processor at least once after:

- starting up the backup system
- restarting a repaired system.

This allows the accumulated value in the standby processor to track the accumulated value in the active processor.

Counter Instructions

Counter instructions are similar in structure to timer instructions. As with the timer instruction, your program should transfer the counter's accumulated value from the Active to the Standby processor at least once after you:

- start up the backup system
- restart a repaired system

This allows the accumulated value in the Standby processor to track the accumulated value in the Active processor.

Drum Instructions

All drum matrix sequence dependent parameters should be transferred on each update as critical variables.

5.6. Implementing Programming Techniques

The section outlines the steps associated with developing the programming to support the RPM backup system.

Retrofitting Existing Programs

If you are retrofitting existing programs to use the RPM backup, you should perform an audit to

identify program functions and changes required. This will provide the basis for inserting required changes into the existing program.

Standby Logic Bypass

Determine the changes that must be made to the logic to ensure the Standby PLC scan is shorter than the Active PLC scan. This will involve skipping sections of the program when the PLC is Standby. Determine the related data elements that must be transferred from the Active PLC.

Critical Data Transfer

Identify all Critical Data and calculate the number of words required to transfer the data using Virtual Bases. Using the I/O map of the Virtual Bases, determine which PLC data locations will be used to transfer the data. Develop the logic to map the data to and from these locations.

Non-Critical Data Transfer

Determine the program parameters that may be classified as non-critical or normal Boolean and word locations. You may further group the non-critical data by transfer method, if V memory transfer is not fast enough for some data elements.

1. **Index Transfer Method**. Determine how you are going to block and segment the data. Reserve a section of V memory to contain the data block. Based on the segment size, determine the Virtual Base words that will be used to transfer the data. Develop the logic to transfer the data blocks by segment.

2. **V Memory Transfer Method.** Reserve a block of V memory to hold all data you wish to transfer. Develop the logic to transfer data to and from these locations. Configure the RPM module to automatically transfer the data using the 2541 Configuration Program.

Program Testing

You should test the program to ensure that the logic related to the RPM module operates properly. This can be accomplished by forcing switchover and observing the results. Although your initial tests may use PLC logic or Pushbuttons to initiate the transfer of control, you also should test automatic switchover. Automatic switchover can be forced by removing the I/O cable from the Active PLC. If you are using the Heartbeat, you can also force automatic switchover by placing the Active PLC in program mode. Finally, you should remove power from the module to observe the fail-safe operation.

CHAPTER 6. TROUBLESHOOTING

6.1. General Module Problems

When the RPM module is first powered up, the RPM GOOD LED at the top of the module should light after a few seconds. If the RPM GOOD indicator remains on, the module hardware is operational. If the indicator fails to illuminate and power has been applied to the module, the module diagnostic software has detected a hardware problem, and the module should be replaced.

SYMPTOM	PROBABLE CAUSE	CORRECTIVE ACTION
No LEDs, including the RPM GOOD Indicator, are lit.	No Power to Module.	Ensure that power is supplied to the rack and that the module is securely inserted.
The RPM GOOD indicator is not lit, but other LEDs are illuminated.	Module hardware failure.	Press the resume button. The module should restart and resume control. If the above action is not successful, cycle the power. If the RPM GOOD light does not remain on, replace the module.

6.2. Configuration Port Problems

The 2541 module provides two LEDs that indicate traffic on the Configuration port. The RCV LED illuminates when a signal is received at the serial port. The XMT LED illuminates when the module transmits data to the serial port. In a normal condition, the RCV LED should flash briefly when a command is sent from the PC to read or write data. Subsequently, XMT LED should flash, indicating that the RPM module is sending a reply.

If the RCV LED does not flash, you are not sending a signal from the PC. Check to ensure you are using the correct serial port on the PC and that the cable is securely plugged in. If the cable is properly installed, then you should check the cable itself for the proper wiring. See Appendix A.

Note:

A serial cable that can be successfully used to program the Siemens® Simatic® 505 PLC via the RS-232 programming port should work correctly with the RPM module.

> *Note:*
>
> *Before you begin troubleshooting switchover or data update problems you should run the RPM Configuration Program and select the Diagnostic Display. The display provides information you may need in diagnosing the problem. See Section. 6.4. RPM Diagnostic Display*

6.3. Switchover problems

SYMPTOM	PROBABLE CAUSE	CORRECTIVE ACTION
Pushbuttons will not cause switchover.	Pushbutton Transfer is not enabled in the RPM configuration.	Check the event log to determine the condition that is inhibiting the transfer. Use the RPM Configuration Program to enable pushbuttons
	The Standby PLC is not operational.	Check the event log. If a configuration error is causing the RPM to declare the PLC as not operational, correct the error. If the PLC is faulty, replace the PLC.
	You are not fully depressing the pushbuttons.	Check the event log to ensure that the pushbutton request event has been recorded. If not, find a different tool to depress the pushbutton.
PLC Logic will not cause switchover.	PLC Logic Transfer is not enabled in the RPM configuration.	Check the event log to determine the condition that is inhibiting the transfer. Use the RPM Configuration Program to enable PLC Logic transfer.
	The Standby PLC is not operational.	Check the event log. If a configuration error is causing the RPM to declare the PLC as not operational, correct the error. If the PLC is faulty, replace the PLC.

SYMPTOM	PROBABLE CAUSE	CORRECTIVE ACTION
contd - PLC Logic will not cause switchover.	Your logic is not setting the correct output points.	Check the event log to ensure that the PLC Logic request event has been recorded. If not, review your home base configuration and the PLC I/O configuration to confirm the correct y address. Check the operation by manually forcing the Y point. Correct PLC logic as required.
Automatic Switchover does not occur.	Automatic Transfer is not enabled in the RPM configuration.	See previous symptom
	The Standby PLC is not operational.	See previous symptom
Control oscillates back and forth between the PLCs	I/O Update Timeout is set too small.	Review the Event Log to confirm the cause. Increase the I/O Update Timeout to a value greater than the maximum scan time.
The outputs bump during switchover.	I/O Update Timeout is set too large, causing I/O base timeout. Settings above 225ms may cause I/O base timeouts.	Reduce the I/O Update Timeout value, if possible. If a very large I/O Update Timeout is required, configure the remote bases to hold last value.
	RBC Offline Declaration is set too small, causing the Standby PLC to erroneously log out the I/O bases.	Ensure the RBC Offline Declaration is set to a value slightly greater than the I/O Update Timeout value.
	The programs in the two PLCs are diverging causing some outputs to transition upon switchover.	See the chapter on Switchover Considerations.

Data Update Problems

SYMPTOM	PROBABLE CAUSE	CORRECTIVE ACTION
I/O is not being mirrored to the Standby PLC	The RPM I/O real bases are not configured properly.	Review the configuration to ensure that all real bases are identified to the module
	Changes to the RPM base configuration has not taken effect.	You must cycle power to the RPM module before I/O base configuration changes will be effective. In addition, you must initially transfer control of the real I/O to the Standby PLC to allow it to log in properly.
	The PLC I/O configuration is incorrect.	Ensure that the I/O is correctly configured in the Standby PLC.
Virtual Base Data is not being updated	The RPM I/O virtual bases are not configured properly.	Review the configuration to ensure that all virtual bases are identified to the module and that the number of virtual modules is entered correctly.
	Changes to the RPM base configuration have not taken effect.	You must cycle power to the RPM module before I/O base configuration changes will be effective.
	The PLC I/O configuration is incorrect.	Ensure that the I/O is correctly configured in the Standby PLC.
	The PLC logic is not operating properly.	Manually force some virtual output words and display the corresponding input words to confirm that transfer will take place. If data can be transferred manually, review and correct the PLC logic.
V Memory Transfer does not work. The data transfer bits are not toggling.	V memory Transfer is not enabled or is improperly set in the RPM configuration.	Ensure that the correct V memory Address and number of words is entered in the RPM configuration.
	The RPM Home Base is not configured. Transfer will not operate unless this is done.	Configure at least one Virtual base that will be designated as Home Base.

SYMPTOM	PROBABLE CAUSE	CORRECTIVE ACTION
V Memory Transfer operates too slowly.	The PLC Task Codes Per Scan is set too small.	Configure the PLC Task Codes per scan to the largest acceptable value. A value of 8 is preferred for maximum performance.
	Too much data is being transferred.	Reduce the size of the transfer block, if possible. Consider using the Indexed Transfer Method for certain data items. See ***Error! Not a valid result for table.***.

6.4. RPM Diagnostic Display

The 2541 RPM configuration also allows you to display information concerning the status of the RPM module and the attached PLCs. It also displays a log of the last 63 events that can be a valuable tool for diagnosing problems. This display is described on the following page.

Module Information

Module Name	2541-RPM	Board Revision	A
Product Description	Redundant Processor Manager	Manufacturing Date	07/31/97
Firmware Version	01.04	Firmware Date	09/23/97	Serial Number	0000000001

PLC A - Information

Scan Type	Variable	Operation Mode	Program	Fatal Error Present	None
Scan Time (ms)	0	Max Between Msgs (ms)	32771	Battery Low	Ok

PLC B - Information

Scan Type	Variable	Operation Mode	Program	Fatal Error Present	None
Scan Time (ms)	0	Max Delay Between Msgs	32772	Battery	Ok

Event Log Records

Event	Initial State	New State	Inhibitor
Configuration Change	PLC A	PLC A	
Scan Timeout	PLC A	PLC A	Standby Scan Timeout

note: most recent event on top

Figure 16. RPM Configuration Program Diagnostic Display

50

Module Information

This section provides general information about the module that may be useful to CTI product support.

Module Name and Description – Identifies the Module

Board Revision / Manufacturing Date – Identifies the revision of the hardware and the date the hardware was manufactured.

Serial Number – Identifies the CTI serial number assigned to the product.

Firmware Revision and Date – Identifies the firmware currently in use on the module.

PLC Information (PLC A and PLC B)

These two sections provide information about PLC characteristics that may be useful in problem diagnosis. Note that most of this information can also be obtained via the PLC programming software.

Scan Type - Indicates whether the PLC is set for a fixed scan or a variable scan.

Scan Time - Displays the peak scan time of the PLC. Note: This information is maintained by the PLC. You must use the PLC programming software to reset the peaks.

Operation Mode – Indicates whether the PLC is in Run mode or Program mode.

Max Time between Msgs – Indicates the maximum time (in milliseconds) between I/O messages from the PLC. You can use this value as an aid to setting the I/O update timeout (see Configuration). This value will survive recycling power to the module. You must use the Read/Reset option of the diagnostic program to reset it.

Note:

Powering off a PLC or removing the I/O cable will result in an I/O timeout. In any event where an I/O timeout occurs, this value will become a very large number. Once the condition which caused the timeout has been resolved, you should use the Read/Reset option to reset the value to a meaningful value.

Fatal Error Present - Indicates whether the PLC is reporting a Fatal Error code.

Battery OK - Indicates the condition of the PLC battery as reported by the PLC.

Event Log – The RPM module will store a record of the last 63 events. The most recent event is shown at the top of the list. The log record has the following fields:

Event – An event is any occurrence that could cause the RPM module to change states. For example, a pushbutton request could cause the RPM module to change states.

Initial State – The Initial State is the control condition before the event occurred. There are two possible states:

1. PLC A – PLC A is in control.

2. PLC B –PLC B is in control.

New State – The New State is the control condition after the event occurred. The possible states are the same as the Initial State.

Inhibitor – An Inhibitor is any condition which prevented the RPM from changing control states. For example, having pushbuttons disabled in the configuration will prevent a pushbutton request event from causing the RPM module to transfer control.

APPENDIX A. CONNECTORS AND CABLING

RPM Configuration Port Cabling
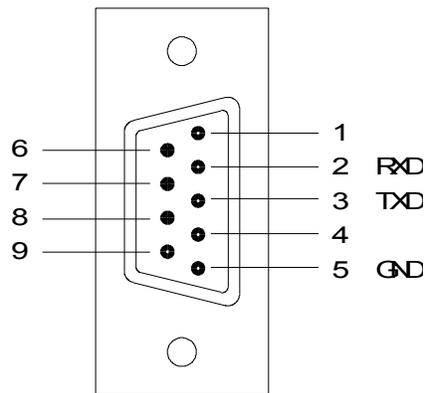
Figure 17. RPM Configuration Port Pinout

The configuration serial port on the 2541 uses a DB9 female connector to provide a standard RS-232 interface. The illustration above shows the pinout for the port. Any standard off-the-shelf null-modem (pins 2 and 3 are rolled in the cable), DB9 to DB9 serial RS-232 cable can be used. With the set baud rate of 9600, you should limit the length of RS-232 configuration cable to 50 feet or less.

Configuration Port Settings:

Baud: 9600

Data bits: 7

Parity: odd

Stop bits: 1

Terminating Cables on the 40 Pin Termination Connector
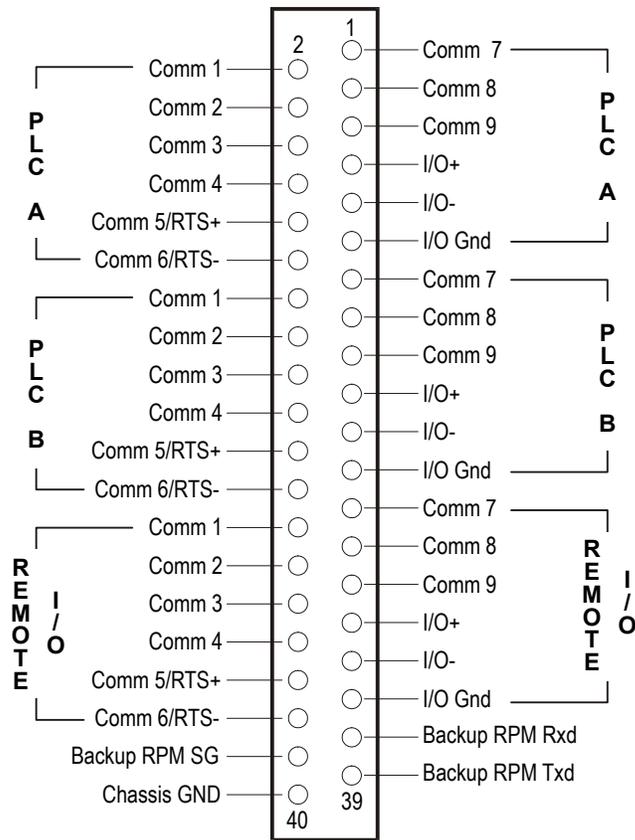
Figure 18. RPM Front Panel 40 Pin Termination Connector

All cables terminating on the RPM front panel's 40-pin termination connector must meet the following requirements:

- be a shielded twisted pair or twin axial,

- shield wire to be connected at BOTH ends,

- do NOT tin ends (cables purchased with individually pre-tinned strands are acceptable for use),

- wires are to be stranded and between 18-24 AWG.

Remote I/O Cabling

In a redundant PLC configuration, remote I/O does not connect directly to the PLCs. Remote I/O devices are connected to dedicated pins on the 40-pin front panel termination connector of the RPM module. The cable from field devices are connected to pins 31, 33 and 35 of the connector, and seen

54

above. These Common terminals are connected to the commons of the internal RPM relay. Be sure to note signal polarity when connecting all I/O connections described here.

The individual remote I/O cables from PLC A and PLC B are connected to the corresponding dedicated remote I/O pins on the termination connector. The remote I/O cable from PLC A terminates on pins 7, 9 and 11. The remote I/O cable from PLC B terminates on pins 19, 21 and 23. The pinout of the remote I/O cables for both PLCs are shown in the following figures:
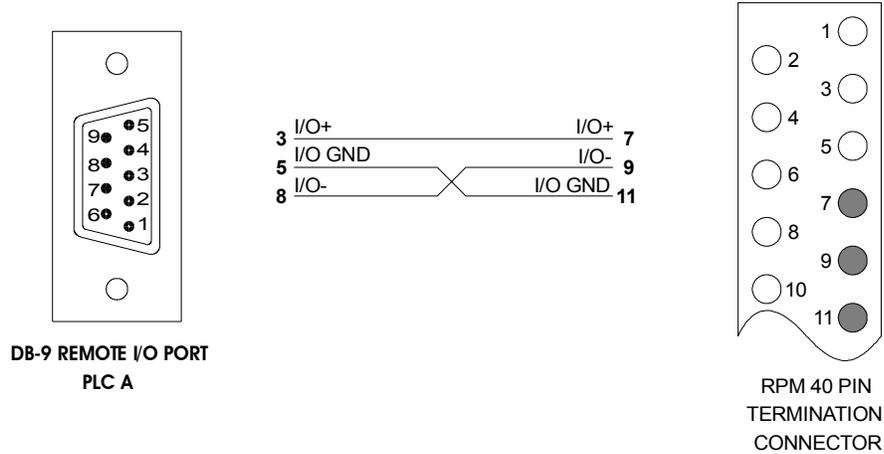
Figure 19. Typical Remote I/O Cable from PLC A to the RPM
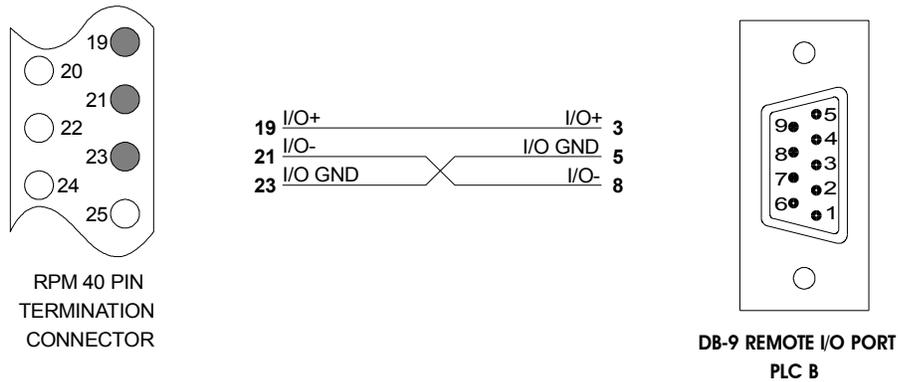
Figure 20. Typical Remote I/O Cable from PLC B to the RPM

Use only the recommended cable types, lengths, and procedures listed in the *SIMATIC® 545/555/575 System Manual*. Be careful to connect all signal wires to the correct signal polarity at both ends. Do not reverse I/O+ and I/O- connections.

System Considerations for Siemens® RS-485/RF Converters

Note:

At present, only Siemens® (model number 505-6860) RS-485/RF Channel Converters are approved for this application.

In applications using the Siemens® RS-485/RF Converter consideration must be given to the system's configuration. The location of the RF Converter relative to the PLCs and 2541 RPM module will determine capabilities and limitations to the redundancy of the system. There are two 'best scenarios' for arranging these components.

The best configuration places the RF Converter together with the 2541 RPM module in a separate rack from either of the system's PLCs. See Figure 21 below. In this configuration, the full benefit of redundant PLCs is achieved. Either PLC can be taken off line or powered down without disconnecting Remote I/O from the controlling PLC. The disadvantage is the additional cost and space requirements for an additional rack. Additionally, cycling power to the rack to reset the 2541 RPM module for changes to the RPM's configuration to take affect will also interrupt Remote I/O communications.
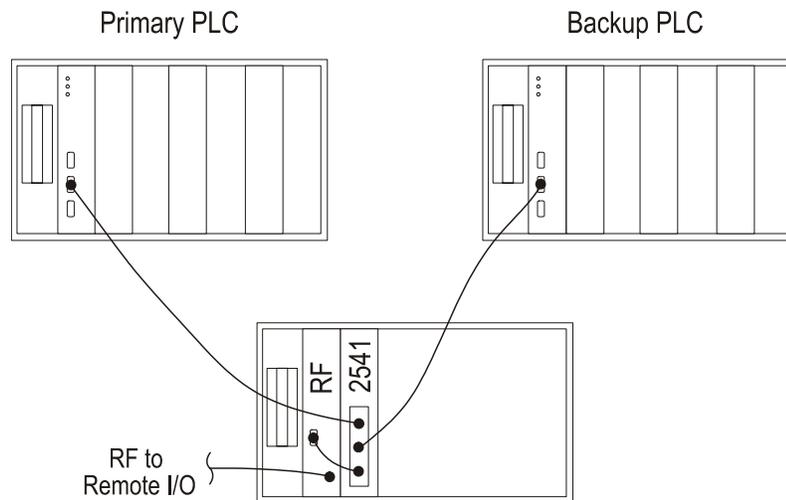
Figure 21 Three-Rack RF Converter Configuration

The next best configuration has the advantage of lower cost and smaller space, but sacrifices some of the benefit of redundant PLCs. In this configuration the RF Converter is placed in the same rack as the primary PLC, which here is defined as the PLC normally in control and is also the Fail Safe PLC (See section *2.3. Setting the 2541 RPM Fail Safe Switches* for more information). The RPM module is then located in the rack containing the backup PLC. See Figure 22 following.

56
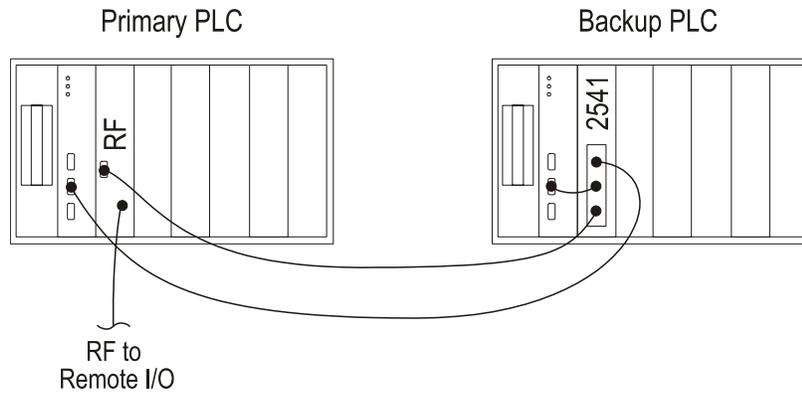
Figure 22 Two-rack RF Converter Configuration

In this configuration, should the backup PLC need to be taken off line or powered down, the system will be unaffected. Otherwise, should the primary PLC fail, the RPM module will switch control over to the backup PLC automatically (within the conditions allowed by the user). As long as power is still applied to the primary PLC rack, the system will continue to function. While powering down the rack containing the primary PLC and RF converter will disconnect Remote I/O from the backup PLC, the system can be shut down at the discretion of the user; not at the unpredictable time that the actual failure had occurred. Thus, much of the benefit of redundant PLCs is retained.

Remote I/O Cabling for Siemens® RS-485/RF Converters

Note:

In applications using the Siemens® RS-485/RF Converter, keep the cables from either PLC to the RPM and from the RPM to the RF Converter as short as possible. There is a maximum length of two feet for the cable from the RF converter to the RPM module and two feet for the cable from the RPM module to either PLC.

As in all RPM applications, remote I/O devices are connected to dedicated pins on the 40-pin front panel termination connector of the RPM module. For applications using RPM modules that have rev. "D" boards or above and use Siemens® RF converters, the cable from a RF converter is connected to pins 31, 33, 34, 35 and 36 of the connector, and seen below.

Note:

When using Siemens® RF converters a set of three 120 ohm terminating resistors are needed. These resistors MUST be placed, one each, where the remote I/O cables terminate at the DB-9 connectors of the RF converter, PLC A and PLC B.
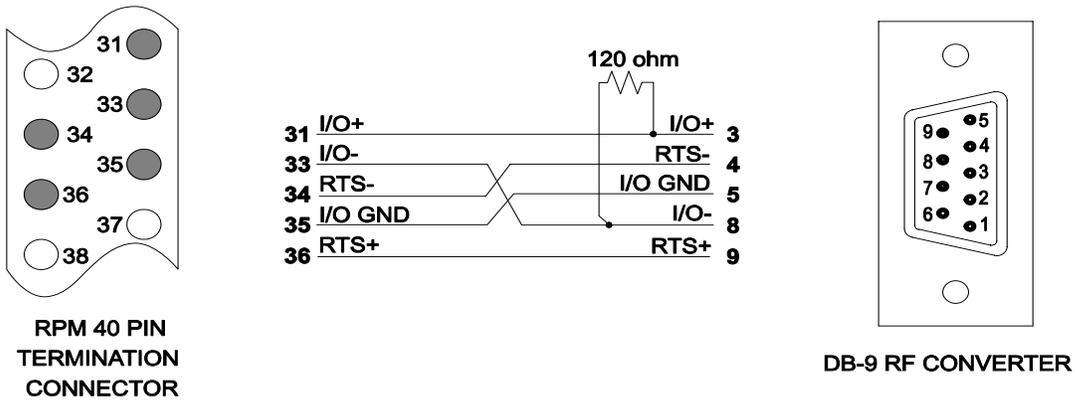
Figure 23. Typical Remote I/O Cable from RF Converter to the RPM

The individual remote I/O cables from PLC A and PLC B are connected to the corresponding dedicated remote I/O pins on the termination connector. The remote I/O cable from PLC A terminates on pins 7, 9, 10, 11 and 12. The remote I/O cable from PLC B terminates on pins 19, 21, 22, 23 and 24. The pinout of the remote I/O cables for both PLCs are shown in the following figures:
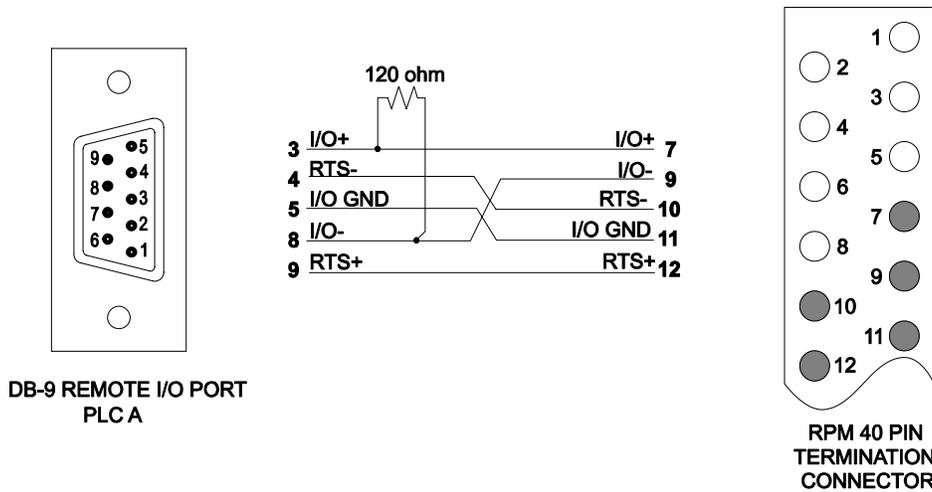
Figure 24. Typical RF Converter Remote I/O Cable from PLC A to the RPM

58
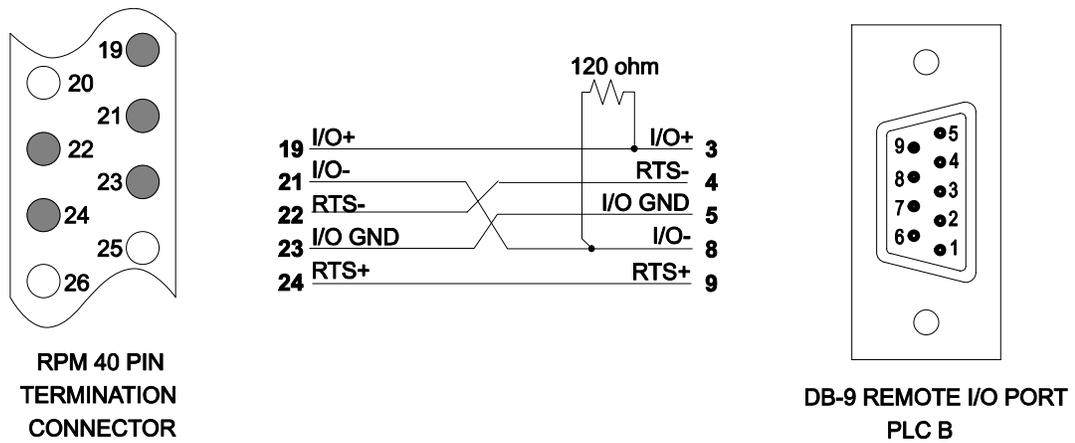
Figure 25. Typical RF Converter Remote I/O Cable from PLC B to the RPM

Specifications for Remote I/O Cables When Using RF Converters

When using Siemens® 505-6860 RF converters, special remote I/O cabling is required. This application requires the use of Beldon 9729 or 8162 cabling or equivalent. The cable must have two twisted pairs of wire individually shielded. For proper signal transmission, three 120-ohm resistors **must** be connected at the DB-9 connections at the RF converter and at both PLCs, not at the 40-pin termination connector. The following figure provides a typical wiring example for the three cables:
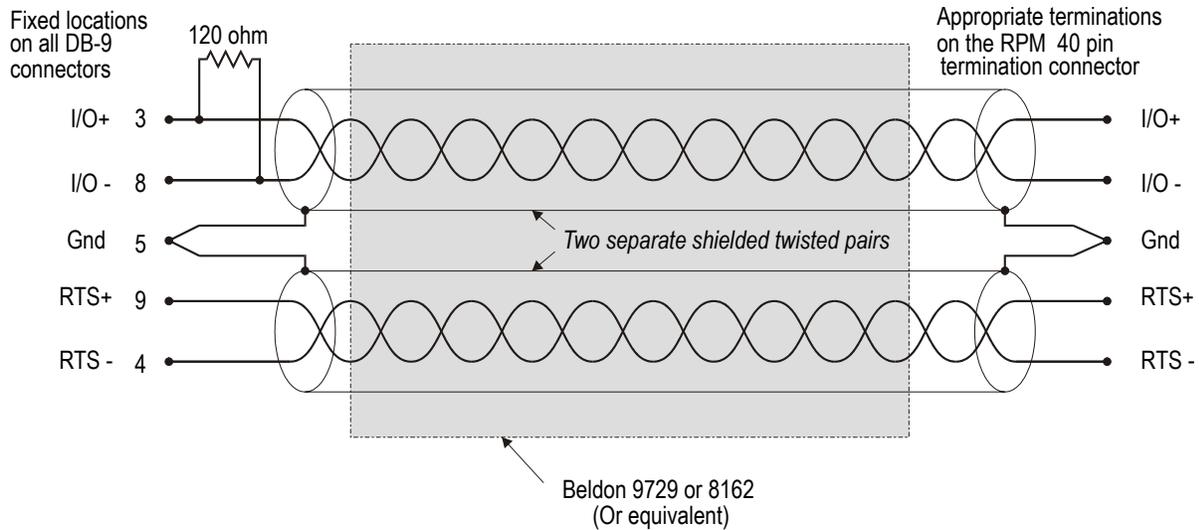
Figure 26. Typical Wiring and Specifications for Remote I/O Cables When Using RF Converters

Cabling Optional Serially Attached Operator Interfaces

Up to two serial devices can be connected to the RPM module. These devices are connected to the Comm terminals of the RPM front panel termination connector. Since each of the sets of Comm 1-9 pins on the termination connector are simple relay contacts, wiring cables to these pins requires only ensuring that connections are made correctly to the corresponding Comm numbers in the PLC A, PLC B and Remote I/O sections of the termination connector.

In the example following, a 5250-TI5 access module is connected to a typical RPM configuration. Like all other devices, the 5250 connects directly to the RPM module and from the RPM to the individual PLCs. The cable connections from the 5250 are terminated onto the section of the RPM front panel connector labeled "remote I/O". In this example, the first three non-dedicated terminals are used (Comm 1/pin 26, Comm 2/pin 28 and Comm 3/pin 30). The RS-232 cables from the two PLCs go to their respective Comm 1, Comm 2, and Comm 3 locations on the RPM connector.

Note that proper 5250 communications requires that the TXD from the 5250 is connected to the RXD of a PLC and the RXD of the 5250 is connected to the TXD of a PLC. Notice that this requirement is met by connecting the TXD of the 5250 to Comm 1 (pin 26) and the RXD to Comm 2 (pin 28), while connecting the RXD of the two PLCs to the corresponding Comm 1 terminals (pins 2 and 14) and their TXD signals to the Comm 2 terminals (pins 4 and 16).

Note:

The function of Comm terminals 5 and 6 are conditional, for the use with RF converters. These terminals are programmed, by setting the RTS COM switch, to carry RTS signals when RF converters are used. See the subheadings earlier in this appendix and Chapter 2. Installation for more information.
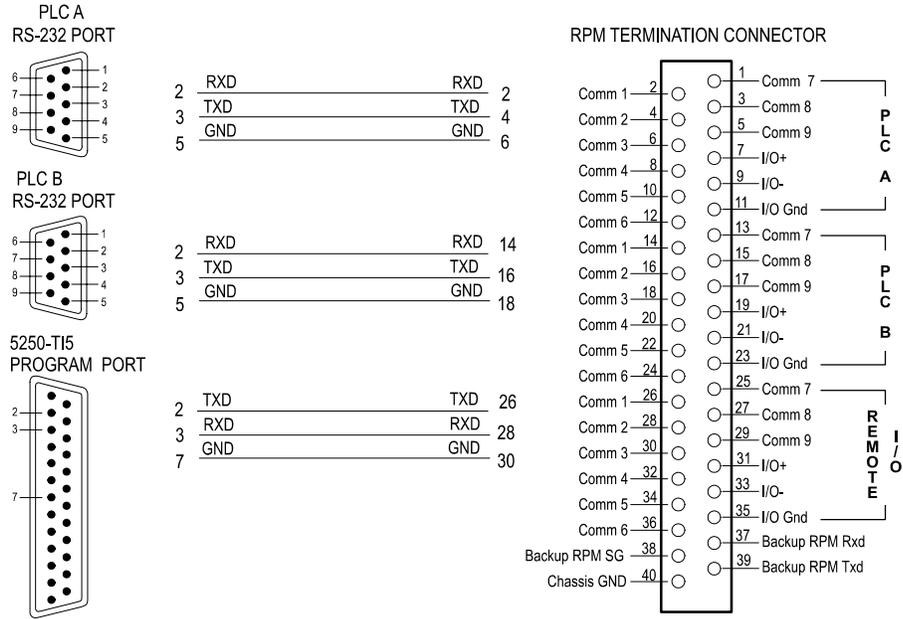
Figure 27.   Serial Device Cabling Example

APPENDIX B. PLC PROGRAMMING TECHNIQUES

Overview

In general, when configuring a RPM in a system there are two main considerations: RPM module configuration and PLC programming. This section is concerned primarily with the PLC programming aspects of integrating a RPM. When instructions are given on how to perform an action, it will be assumed that TiSoft is running and that the application is open.

The RPM is used to provide redundancy for an application. To accomplish this it copies data from the active PLC to the standby PLC. This is referred to as "data flow" and is described as data flow from the active PLC to the standby PLC. The programmer must ensure that all the necessary data is copied from the active PLC to the standby PLC in a timely manner so as to ensure a "bumpless" transfer should the active PLC fail. To do this the programmer "maps" the data to virtual addresses.

"Mapping" refers to assigning an input address, output address, mnemonic or any other piece of data to a virtual address. These virtual addresses are 32WX/32WY "pairs" that are configured in the RPM using CTI configuration software. For the RPM, the programmer must configure at least one virtual base, called the homebase, which facilitates mapping of the data. Other virtual bases can be configured based on the data requirements of the application. When configuring the homebase for the RPM the programmer must configure its I/O just as for any physical base. To configure the I/O for the virtual homebase:

1) Press **Spacebar.**

2) Press **CONFIO-F3.**

3) Use the arrow keys to highlight base 1. Press **CONFIG-F2**.

4) Follow the instructions for configuring base I/O given in the TiSoft manual. For the virtual homebase the slots can be configured as follows:

SLOT	I/O ADDRESS	X	Y	WX	WY
1	0001	32			
2	0033		32		
3	0001			32	32
4	0065			32	32
5	0129			32	32
6	0193			32	32
7	0257			32	32
8	0321			32	32
9	0385			32	32
10	0449			32	32
11	0513			32	32
12	0577			32	32
13	0641			32	32
14	0705			32	32
15	0769			32	32
16	0833			32	32

Table 1

Note that slots 1 and 2 are reserved for RPM use only. Do not assign any data to these addresses. Begin all WX/WY addressing at slot 3 in the virtual homebase. The RPM will automatically assign the WX/WY pairs. Therefore, the above table shows the following WX/WY pairs:

WX1/WY33	WX321/WY353	WX641/WY673
WX65/WY97	WX385/WY417	WX705/WY737
WX129/WY161	WX449/WY481	WX769/WY801
WX193/WY225	WX513/WY545	WX833/WY865
WX257/WY289	WX577/WY609	

Data stored in WX1 is automatically copied into WY33 by the RPM thereby forming the WX1/WY33 pair. When the programmer maps data, data is mapped from the WX address to the

64

element parameter when the PLC is the standby PLC. Data is mapped from the element parameter to the WY address when the PLC is the active PLC. The WX addresses should be contiguous and the WY addresses should be contiguous.

For example, let LSP1 be the data of interest. The programmer copies the contents of WX1 to LSP1 if the PLC is designated as the standby PLC as indicated by the X3 bit in the RPM virtual homebase. Likewise, the programmer copies the contents of LSP1 to WY33 if the PLC is designated as the active PLC as indicated by the X3 bit in the RPM virtual homebase. The flow of data can be described as the PLC reading in the value of LSP1 when in standby mode and the PLC writing out the value of LSP1 when in active mode.

By controlling the flow of data, the programmer can keep the standby PLC current with the active PLC thus ensuring "bumpless" transfer. Data flow is controlled using three methods:

1) critical data transfer

2) indexed data transfer

3) normal data transfer

Critical Data Transfer

Critical data transfer requires mapping of data that occurs every scan. As explained in *Chapter 5. PLC Programming,* this is done in the ladder logic of the program. The following format is recommended when programming for an application:

1) Establish status logic

2) Map the user-designated virtual WX addresses to discrete I/O (including all X and Y addresses) using the MIRFT command.

3) Map loop parameters using the Special Function feature.

4) Map parameters for timers, drums, counters, etc. using the MOVW command.

5) Enter all required design logic including ladder logic, special functions, loops, analog alarms for the application.

6) Map parameters for timers, drums, counters, etc. using the MOVW command.

7) Map loop parameters using the Special Function feature.

8) Map discrete I/O to the user-designated virtual WY addresses (including all X and Y addresses) using the MIRTT command.

The above format is not strictly required with the exception that all virtual WX addresses must be mapped before the ladder is executed and that all virtual WY addresses must be mapped after the ladder logic is executed.

For loops use a Special Function to map any variables designated for critical transfer. This requires at least two Special Functions - one for assigning WX address values to loop variables and one for assigning loop variables to WY addresses. It is possible to map all loops in both Special Functions. This results in the first Special Function containing all the variables of all the loops designated for

critical transfer being mapped to WX addresses and in the second Special Function containing all the same variables of all the loops being mapped to WY addresses.

PLC Status Logic

Use the BITP command to look at bit 15 of STW2 in order to monitor the status of the RPM virtual homebase (base 1 here) as shown below. This allows the PLC to know when a fault has occurred and enables the mapping. An example that tests for fault is shown below.

```
         C1                                                          C1
1       ─┤/├────────────────────────────────────────────────────────(   )
         C1
        ─┤ ├─
         C1    ┌BITP1────┐                                           C2
5       ─┤ ├───┤         ├───────────────────────────────────────────(   )
               │  A:STW2 │
               │  N=15   │
               └─────────┘
```
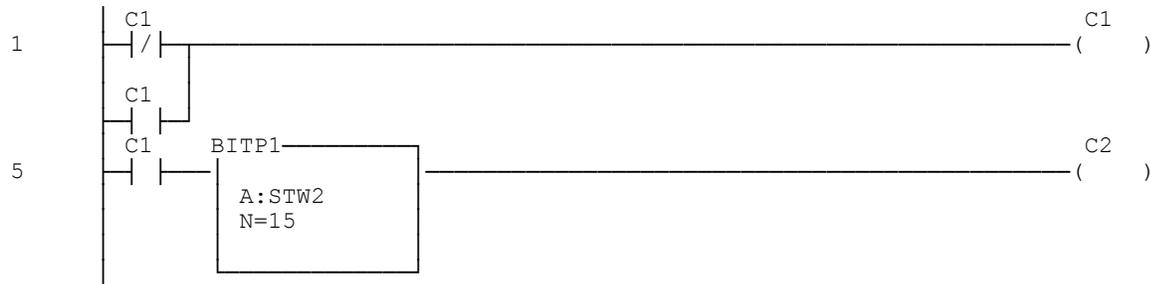
The diagram below shows the relationship between STW2 bits and the remote I/O bases they represent. Use the STW2 bit number indicated for the N value above for a homebase location other than at base 1.

STW2 Bit #		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Remote Base #   15 14 13 12 11 10 9   8   7   6   5   4   3   2   1   0

Mapping Discrete I/O for Standby Mode

To map discrete I/O use the MIRFT command. Since data is only to be copied from a WX address to an element parameter when the PLC is in standby mode, the MIRFT command must be conditional on the X3 bit in the RPM virtual homebase being a normally closed contact. Typically, map the inputs first, then map the outputs. For example, assume that WX/WY pairs are assigned to I/O addresses as shown below.

WX address	WY address	I/O address
1	033	X65
3	035	Y97
131	163	C1

Table 2

In this case, only 32 contiguous bits for X-addresses (inputs), 32 contiguous bits for Y-addresses (outputs) and 32 contiguous bits for C-addresses are being mapped. Based on the assignments above an example of the ladder logic implementation of the mapping is shown below.

66

```
      |  C2     X3     MIRFT1                                                C4
9     |  ┤/├────┤/├───┌──────────┐──────────────────────────────────────────(   )
      |             │TS:WX1    │
      |             │IR:X65    │
      |             │ N=2      │
      |             └──────────┘

      |  C2     X3     MIRFT2                                                C5
16    |  ┤/├────┤/├───┌──────────┐──────────────────────────────────────────(   )
      |             │TS:WX3    │
      |             │IR:Y97    │
      |             │ N=2      │
      |             └──────────┘

      |  C2     X3     MIRFT3                                                C6
23    |  ┤/├────┤/├───┌──────────┐──────────────────────────────────────────(   )
      |             │TS:WX131  │
      |             │IR:C1     │
      |             │ N=2      │
      |             └──────────┘
```
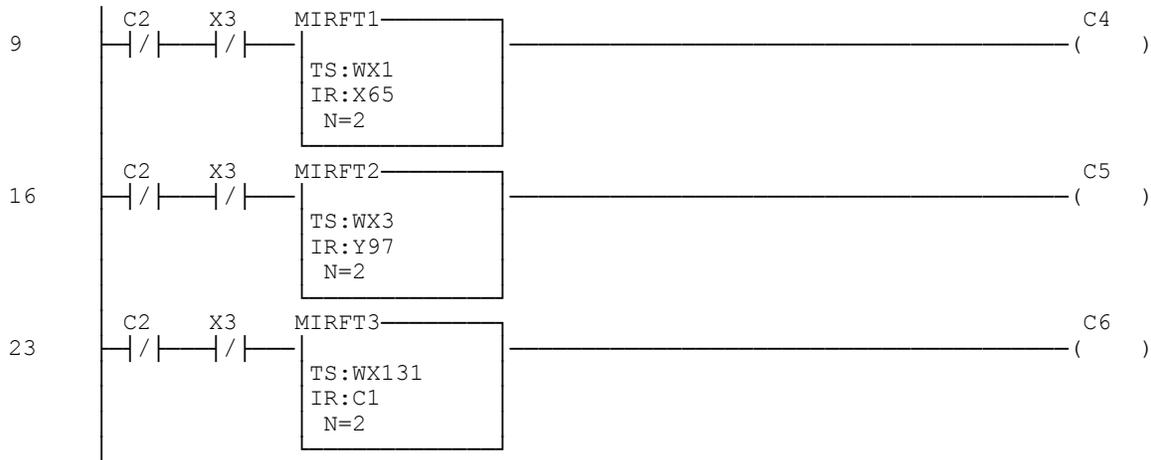
Mapping Loop Parameters for Standby Mode

To map loop parameters use the Special Function feature in ladder logic. The SFPGM box is to be conditional on the X3 bit being a normally closed contact as shown below.

```
      |  C2     X3     SFPGM1                                                C7
30    |  ┤/├────┤/├───┌──────────┐──────────────────────────────────────────(   )
      |             │          │
      |             │          │
      |             │          │
      |             └──────────┘
```
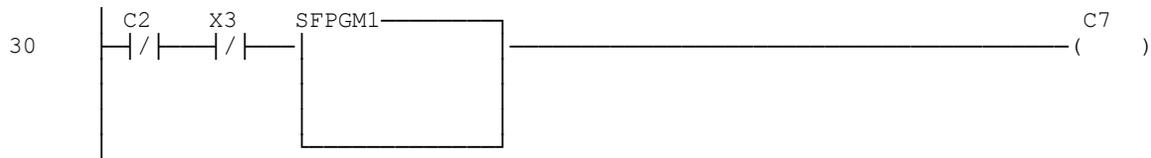
The Special Function program itself will simply consist of a series of assignments from the WX addresses to the variables of interest. This is referred to as 'mapping'. Below is an example of mapping that reads in the values from the WX addresses and assigns values to variables.

```
00002 MATH    LSP1 := WX5

00003 MATH    LSP2 := WX6

00004 MATH    V110 := WX7

00005 MATH    LACK1 := WX8

00006 MATH    LADB1 := WX9

00007 MATH    LCFH1 := WX10
```

The programmer arbitrarily assigns the WX addresses to the variables based on concerns related to the specific application. In the above example, the first three lines assign WX addresses to the setpoints of three loops. The first two loops operate in automatic mode and therefore store the contents of the WX address directly to the loop setpoints. The third line shows an example of a cascade loop that uses V-memory (in this case V110) to reference its setpoint. Here the contents of WX7 are copied to V110 which is then referenced by Loop 3 for its setpoint. In the above example the setpoints for the loops are shown mapped contiguously. Typically, a programmer would assign

contiguous addresses for all Loop 1 variables, then assign contiguous addresses for all Loop 2 variables, etc.

For loops there are eight real variables as shown below.

```
00059 MATH    LKC1. := V111.
00060 MATH    LKD1. := V113.
00061 MATH    LPVH1. := V115.
00062 MATH    LPVL1. := V117.
00063 MATH    LRCA1. := V119.
00064 MATH    LTD1. := V121.
00065 MATH    LTI1. := V123.
00066 MATH    LTS1. := V125.
```

Use V-memory to map these variables. The above example shows how to use V-memory to map the eight real loop variables. Ensure that the V-memory designated for normal memory transfer and the V-memory designated for loop variable critical transfer **DO NOT OVERLAP**. For example, if V-memory locations V1-V100 are designated for normal transfer in the RPM module configuration then start V-memory mapping for loop variables designated for critical transfer at some V-location above V100. Remember to skip an address for every assignment for real variables as shown above.

Mapping Parameters for Timers, Drums, Counters for Standby Mode

To map parameters for timers, drums and counters, use the MOVW command. The MOVW command is to be conditional on the X3 bit being a normally closed contact.

In a typical backup application, two like processors are storing and running identical programs. If these programs contain drums, timers and/or counters then the states of the programs will diverge over time since the states of the individual elements will diverge over time if the appropriate variables are not mapped correctly. This can result in hesitation or skipping of the output when a switchover occurs. To avoid hesitation or skipping the general approach is to read in the values to the appropriate variables if in standby mode and to write out the values to the appropriate variables in both modes.

In our example program, there is one drum and four counters. The states of all five elements must be kept current. The variables are mapped as follows:

WX address	WY address	Variable
094	126	DSC1
095	127	TCC1
096	128	TCC2
129	161	TCC20
130	162	TCC21

Table 3

Given the above assignments then the ladder logic implementation is given below. Note that each rung is conditional on the PLC being in standby mode via the X3 bit and the processor faulting via the C2 bit.

```
           C2     X3    MOVW2─────────────                                            C9
  34      ─┤/├───┤/├──┤                 │──────────────────────────────────────────( )
                        │   A:WX94       │
                        │   B:DSC1       │
                        │   N=1          │
                        └────────────────┘


           C2     X3    MOVW3─────────────                                            C10
  41      ─┤/├───┤/├──┤                 │──────────────────────────────────────────( )
                        │   A:WX95       │
                        │   B:TCC1       │
                        │   N=1          │
                        └────────────────┘

           C2     X3    MOVW1─────────────                                            C50
  48      ─┤/├───┤/├──┤                 │──────────────────────────────────────────( )
                        │   A:WX96       │
                        │   B:TCC2       │
                        │   N=1          │
                        └────────────────┘

           C2     X3    MOVW4─────────────                                            C51
  55      ─┤/├───┤/├──┤                 │──────────────────────────────────────────( )
                        │   A:WX129      │
                        │   B:TCC20      │
                        │   N=1          │
                        └────────────────┘

           C2     X3    MOVW5─────────────                                            C52
  62      ─┤/├───┤/├──┤                 │──────────────────────────────────────────( )
                        │   A:WX130      │
                        │   B:TCC21      │
                        │   N=1          │
                        └────────────────┘
```
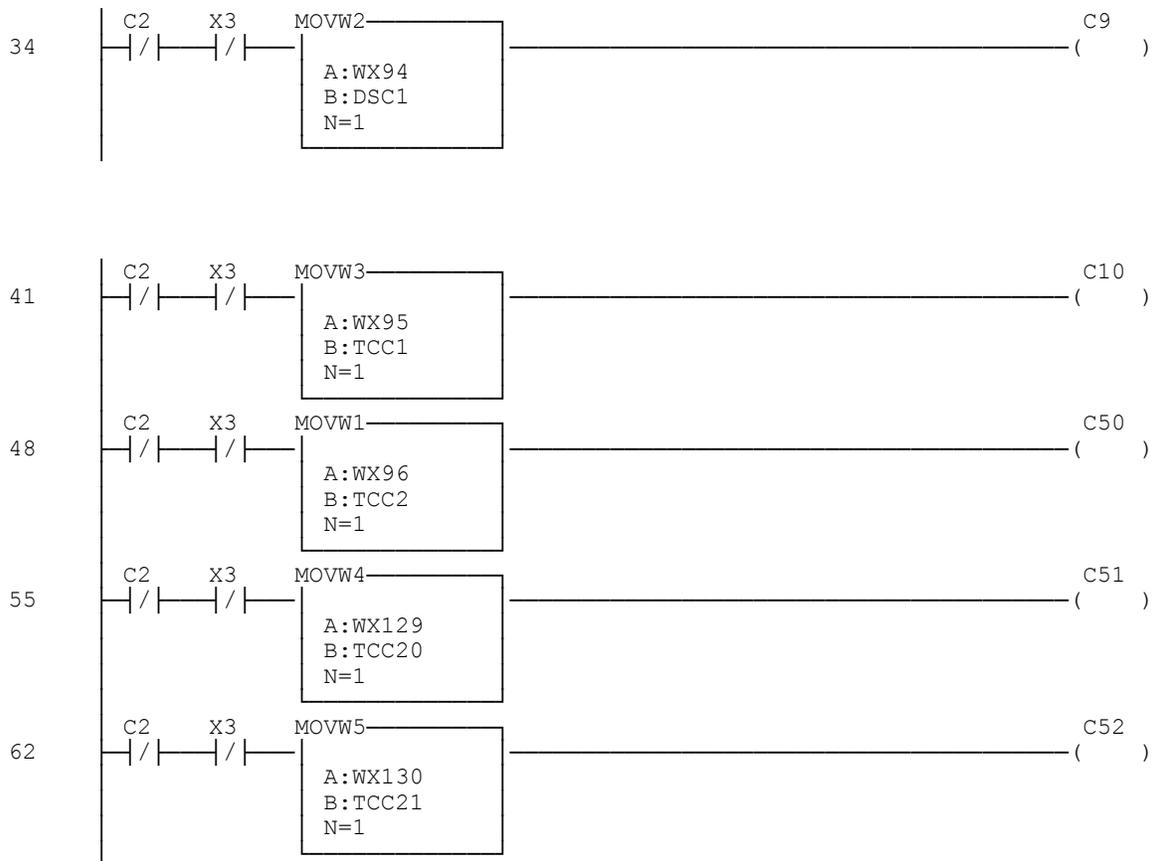
Mapping Parameters for Timers, Drums, Counters for Active Mode

After programming the main section of the ladder logic for the application the programmer must map those variables designated for critical transfer to their assigned WY addresses. To map parameters for timers, drums and counters, use the MOVW command. When the X3 bit in the RPM virtual homebase is ON the processor is in active mode. When the X3 bit is OFF then the processor is in standby mode. Recall that when the processor is in active mode it writes values to WY addresses but does not read values from WX addresses. When the processor is in standby mode it reads values from WX addresses and writes values to WY addresses. Therefore, when programming ladder logic, use normally closed contacts for X3 and the fault status bit as the conditionals for reading values from WX addresses (to indicate standby mode). Conversely, since both processors are always writing values (for drums, timers and counters only) to WY addresses, simply use the fault status bit as the conditional for writing values to the WY addresses.

Recall from table 3 the WX/WY address pairs assigned to the element variables. Below is an example of mapping the drum, timer and counter elements to their corresponding WY addresses.

```
        C2      MOVW8                                                  C25
198    ─┤/├──┌─────────┐─────────────────────────────────────────────( )
              │ A:DSC1  │
              │ B:WY126 │
              │ N=1     │
              └─────────┘

        C2      MOVW9                                                  C26
204    ─┤/├──┌─────────┐─────────────────────────────────────────────( )
              │ A:TCC1  │
              │ B:WY127 │
              │ N=1     │
              └─────────┘

        C2      MOVW10                                                 C27
210    ─┤/├──┌─────────┐─────────────────────────────────────────────( )
              │ A:TCC2  │
              │ B:WY128 │
              │ N=1     │
              └─────────┘


        C2      MOVW11                                                 C28
216    ─┤/├──┌─────────┐─────────────────────────────────────────────( )
              │ A:TCC20 │
              │ B:WY161 │
              │ N=1     │
              └─────────┘

        C2      MOVW12                                                 C29
222    ─┤/├──┌─────────┐─────────────────────────────────────────────( )
              │ A:TCC21 │
              │ B:WY162 │
              │ N=1     │
              └─────────┘
```
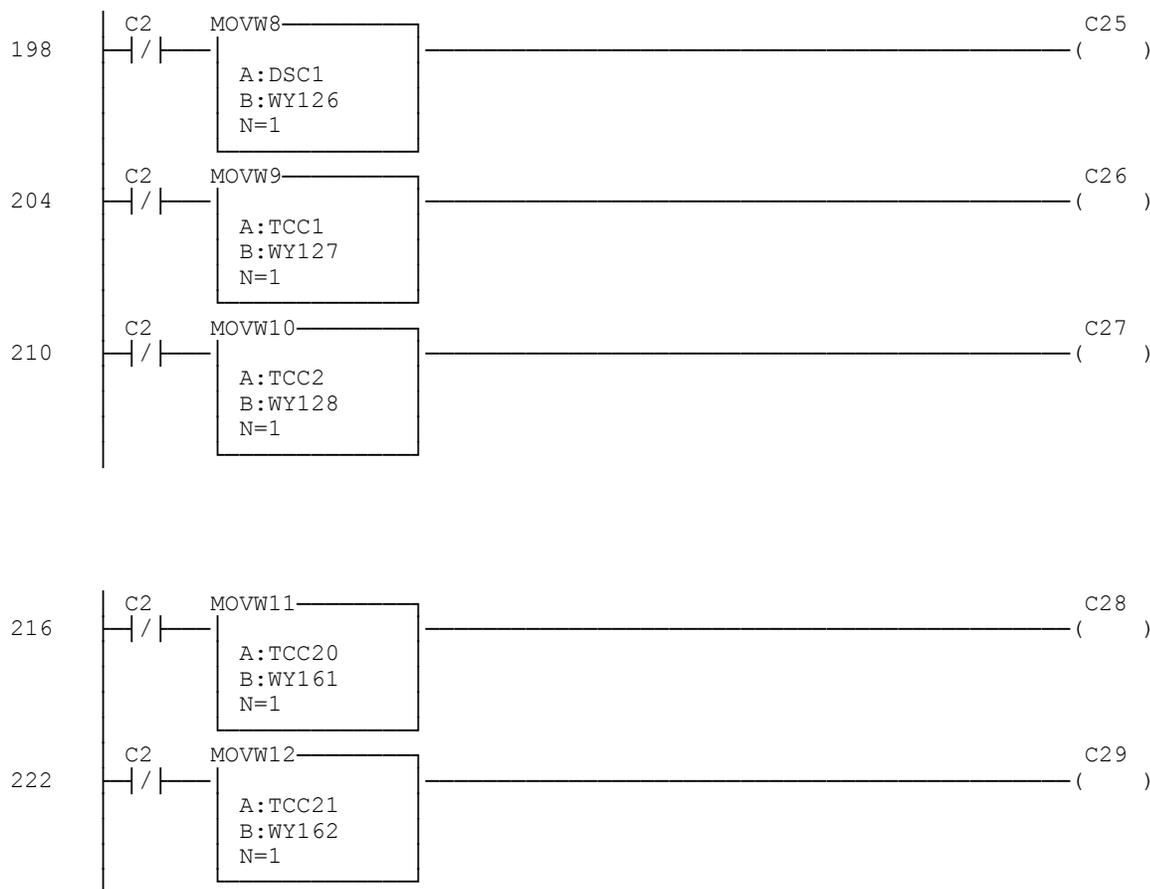
Mapping Loop Parameters for Active Mode

A second Special Function box is used for mapping loop variables to their assigned WY addresses after the main ladder logic programming is completed. Unlike drums, timers or counters, loop

parameters should be updated only when in active mode. Therefore, the SFPGM box is to be conditional on the X3 bit being a normally open contact as shown below.

```
           C2      X3      SFPGM2                                            C24
194       ┤ / ├───┤ ├─────┌──────────┐──────────────────────────────────────(   )
                         │          │
                         └──────────┘
```
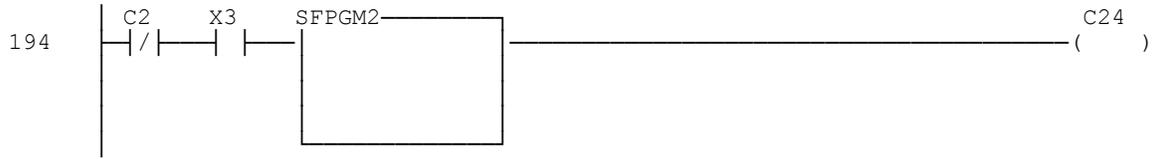
Below is an example of coding for mapping loop variables to their assigned WY addresses.

```
00002 MATH     WY37 := LSP1
00003 MATH     WY38 := LSP2
00004 MATH     WY39 := V110
00005 MATH     WY40 := LACK1
00006 MATH     WY41 := LADB1
00007 MATH     WY42 := LCFH1
```

The section of V-memory used for transferring variables designated for critical transfer can be thought of as a 'scratchpad'. Ensure that the scratchpad does not overlap the normal V-memory configured in the RPM for normal memory transfer. Below is an example of the real loop variables mapped to their assigned V-memory locations.

```
00059 MATH     V111. := LKC1.
00060 MATH     V113. := LKD1.
00061 MATH     V115. := LPVH1.
00062 MATH     V117. := LPVL1.
00063 MATH     V119. := LRCA1.
00064 MATH     V121. := LTD1.
00065 MATH     V123. := LTI1.
00066 MATH     V125. := LTS1.
```

Mapping Discrete I/O for Active Mode

Like the Special Function box, discrete I/O should only be written to their assigned WY addresses when in active mode. To map discrete I/O use the MIRTT command. The MIRTT command is to be conditional on the X3 bit being a normally open contact. Typically, map the inputs first, then map the outputs. Recall the WX/WY assignments for the discrete I/O addresses from table 1. Below is an example of the WY mapping for discrete I/O.

```
          C2     X3      MIRTT1                                                    C21
174      ─┤/├───┤ ├────┌────────────┐──────────────────────────────────────────( )
                       │ IR:X65     │
                       │ TD:WY33    │
                       │  N=2       │
                       └────────────┘

          C2     X3      MIRTT2                                                    C22
181      ─┤/├───┤ ├────┌────────────┐──────────────────────────────────────────( )
                       │ IR:Y97     │
                       │ TD:WY35    │
                       │  N=2       │
                       └────────────┘

          C2             MIRTT3                                                    C23
188      ─┤/├──────────┌────────────┐──────────────────────────────────────────( )
                       │ IR:C1      │
                       │ TD:WY163   │
                       │  N=2       │
                       └────────────┘
```
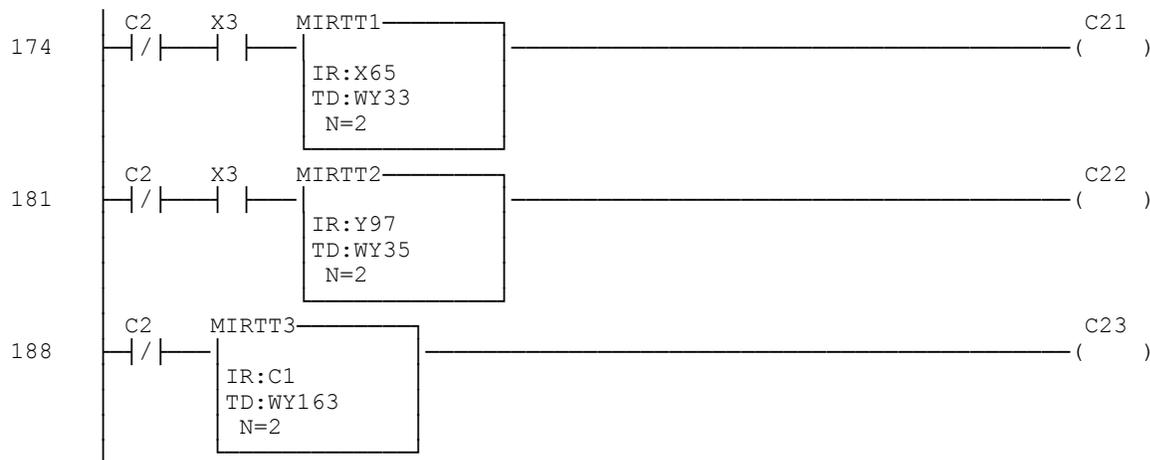
Note that mapping C-memory is not conditional on the X3 bit in the RPM virtual homebase. Some of the C-bits are used with drums, timers and counters in our example. Because of this, divergence becomes an issue with regard to those specific C-bits. Therefore, like the variables for the drums, timers and counters, these C-bits are mapped when the PLC is in either active or standby mode. This minimizes divergence and prevents skips or hesitations when a transfer occurs. Recall from table 2 the WX/WY pairs assigned to the discrete I/O in our example. The ladder logic implementation is shown above.

Ensure that mapping for both Standby Mode and Active Mode is done in the same program, since either PLC can be the active or standby PLC. Also, the same program must be loaded into both PLCs. This will ensure that bumpless transfer occurs between the two PLCs.

Indexed Data Transfer

Indexed Data transfer provides a means of transferring data that is faster than normal data transfer and slower than critical data transfer. Instead of transferring all data with every scan as is the case with **Critical data transfer**, data is divided into segments that get transferred over each scan. The number of scans required to transfer all the data designated as indexed data is determined by the total amount of data divided by the amount of data per data segment. This reduces the overall amount of data being transferred with each PLC scan improving efficiency. The programmer arbitrarily decides which data needs to be designated as indexed and how large the data blocks should be for each individual application.

Below is an example of how to transfer V-memory using the **indexed data transfer** method. For this example a series of loops were used to demonstrate the method. The actual data that will be

72

transferred using the indexed data transfer method will vary depending on the application. The V-memory locations were assigned values using special function program, SFPGM3, to map the values of the loop variables to each V-memory location.

```
         C1     X3     SFPGM3─────────                                              C55
 74     ─┤/├──┤ ├──┤            │──────────────────────────────────────────────(    )
                      │            │
                      └────────────┘
```
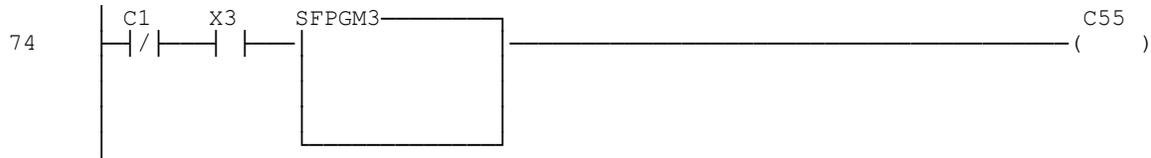
Below is the code for assigning the loop 4 variables to V-memory. Notice that reals and integers can be mixed arbitrarily. The code is similar for loops 5-20.

```
00001 MATH     V103 := LACK4
00002 MATH     V104 := LADB4
00003 MATH     V105 := LCFH4
00004 MATH     V106 := LCFL4
00005 MATH     V107 := LHA4
00006 MATH     V108 := LHHA4
00007 MATH     V109 := LLA4
00008 MATH     V110 := LLLA4
00009 MATH     V111 := LMX4
00010 MATH     V112 := LODA4
00011 MATH     V113 := LRSN4
00012 MATH     V114 := LRSF4
00013 MATH     V115 := LSPH4
00014 MATH     V116 := LSPL4
00015 MATH     V117 := LVF4
00016 MATH     V118 := LYDA4
00017 MATH     V119 := LSP4
00018 MATH     V120. := LKC4.
00019 MATH     V122. := LKD4.
00020 MATH     V124. := LPVH4.
00021 MATH     V126. := LPVL4.
00022 MATH     V128. := LRCA4.
00023 MATH     V130. := LTD4.
00024 MATH     V132. := LTI4.
00025 MATH     V134. := LTS4.
```

Once the V-memory has been assigned, the programmer must decide how many scans they need to transfer the data. For this example, the data to be transferred is at V101-V695. The MOVE instruction is chosen and the WORD designation is used. Therefore, the total number of locations to be transferred is 595 (696 -101 = 595). The maximum number of words that can be transferred in each data segment is 256. The data is arbitrarily divided into 32-WORD segments for our example.

The number of segments to be transferred is rounded up to 19 (595/32 = 18.05). Always transfer whole segments to ensure that all the data is transferred. Since there are 19 segments of V-memory to be transferred, it will take 19 scans to transfer it all. This is deemed acceptable for the purposes of this example. Normally, for each application the programmer determines the maximum number of scans they can tolerate in order to transfer the data over from the active PLC to the standby PLC. The amount of toleration is determined by how fast the data is changing and how much error in the data is acceptable for the particular application. The time required for all the data to be transferred is determined by the size of the application, the amount of data designated for **Indexed data transfer** and the size of data segments chosen.

Indexed Data Transfer - Standby Mode

Below is the code used to transfer the V-memory. First is shown the code required for the standby PLC to read in the values from the active PLC. It assigns values to V-memory from the WX addresses in the RPM. The location V950 holds the value of the offsets for the V-memory locations to be assigned. V952 holds the offsets for the WX addresses.

```
     C1    C63   MOVE55─────      MOVE62─────                            C56
78   ┤/├───┤ ├─   WORD             WORD        ────────────────────────( )

     C1    C70   TS:+0            TS:+0
     ┤/├───┤ ├─  SI:              SI:
                 TD:V950          TD:V952
                 DI:              DI:
                  N=+1             N=+1
```
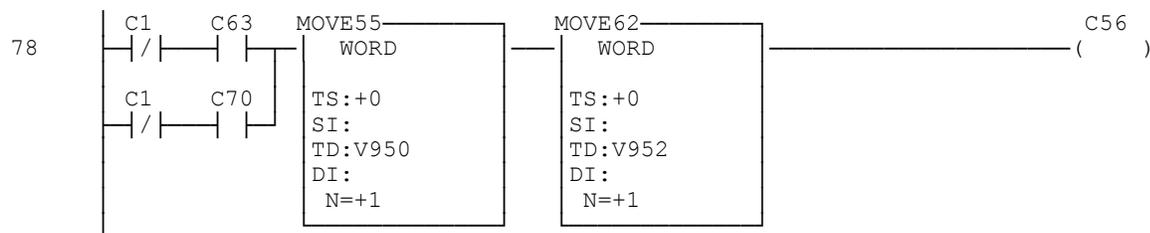
When the PLC is in standby mode, the values of the WX addresses are assigned to their respective V-memory locations.

```
     C1    X3    C61   MOVE60─────                                       C60
94   ┤/├───┤/├───┤/├─   WORD       ────────────────────────────────────( )

                       TS:WX193
                       SI:V952
                       TD:V1001
                       DI:V950
                        N=+32
```
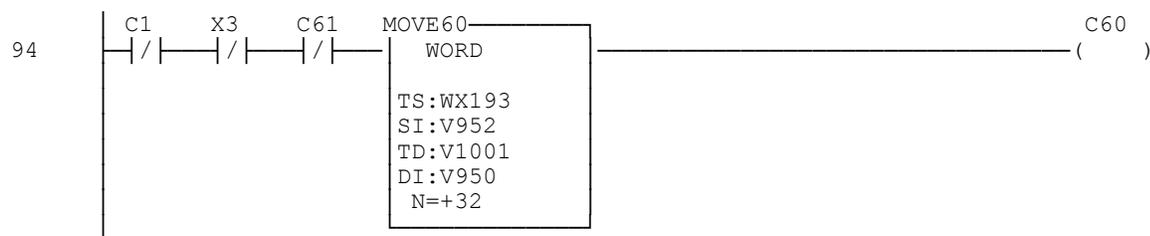
The offsets are incremented after every segment is transferred.

```
       C60    1    ADD1-----------    ADD3-----------                          C61
105    ┤ ├─────:O:──                                                          (    )
                     │           │    │           │
                     │  A:V950   │    │  A:V952   │
                     │  B:+32    │    │  B:+64    │
                     │  C:V950   │    │  C:V952   │
                     └───────────┘    └───────────┘
```
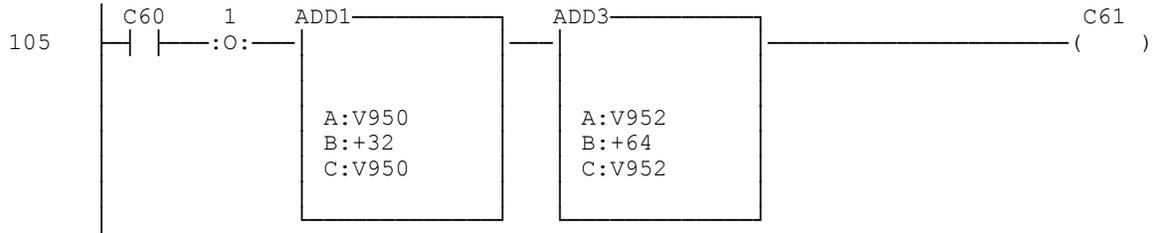
The counter contains the number of data segments to be transferred. It is reset when the transfer is complete.
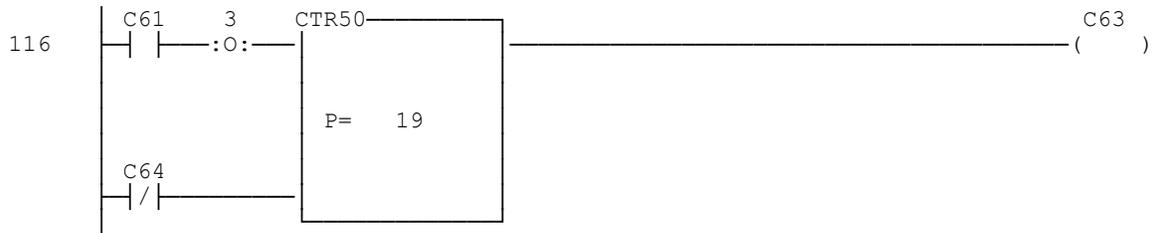
```
       C61    3    CTR50----------                                            C63
116    ┤ ├─────:O:──                                                         (    )
                     │           │
                     │  P=   19  │
       C64           │           │
       ┤/├───────────┤           │
                     └───────────┘
```

Below are momentary and latched outputs indicating that the transfer of data is complete that can be used in other parts of the logic.

```
       C63    4                                                              C64
122    ┤ ├─────:O:─────────────────────────────────────────────────────────(    )

       C63    5                                                              C65
125    ┤ ├─────:O:──┐                                                       (    )
                    │
       C65          │
       ┤ ├──────────┘
```
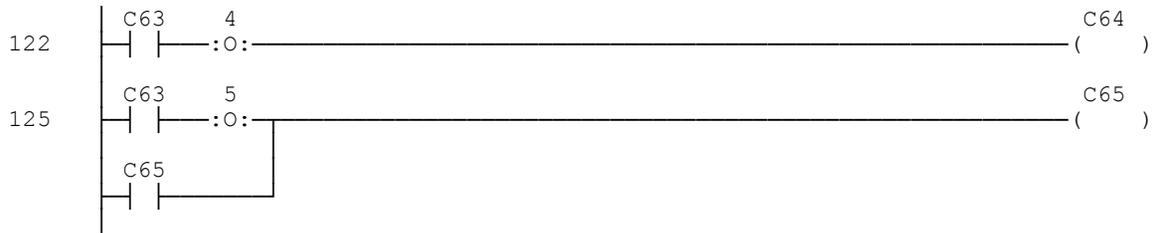
Indexed Data Transfer - Active mode

When the PLC is in active mode the PLC writes the values of the variables to the V-memory.

```
        C1      C83    MOVE56─────────      MOVE56─────────                       C80
129    ─┤/├────┤ ├──┐    WORD        │──┐    WORD        │──────────────────────( )
        C1      C70 │  TS:+0         │  │  TS:+0         │
       ─┤/├────┤ ├──┘  SI:           │  │  SI:           │
                       TD:V951       │  │  TD:V953       │
                       DI:           │  │  DI:           │
                        N=+1         │  │   N=+1         │
                       ───────────────  ─────────────────
```
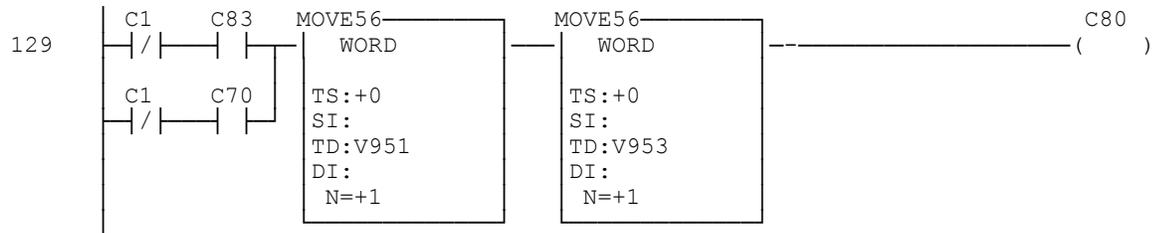
When the PLC is in active mode, the V-memory locations are assigned to their respective WY addresses in the RPM. Note that V951 contains the offset for the V-memory locations to be transferred, V953 contains the offset for the respective WY addresses.

```
        C1      X3     C82    MOVE61─────────                                     C81
145    ─┤/├────┤ ├────┤/├───    WORD        │─────────────────────────────────( )
                              TS:V1001      │
                              SI:V951       │
                              TD:WY225      │
                              DI:V953       │
                               N=+32        │
                              ───────────────
```
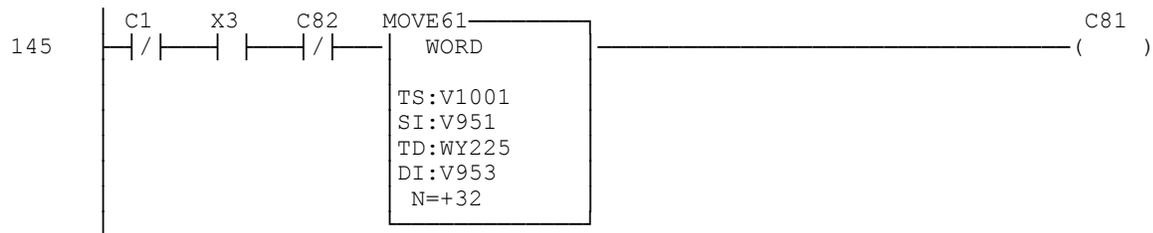
The offset is incremented after each segment is transferred.

```
        C81     2     ADD2─────────      ADD4─────────                           C82
156    ─┤ ├──── :O:─    │            │──    │            │──────────────────────( )
                        │            │      │            │
                        A:V951       │      A:V953       │
                        B:+32        │      B:+64        │
                        C:V951       │      C:V953       │
                        ───────────────    ───────────────
```
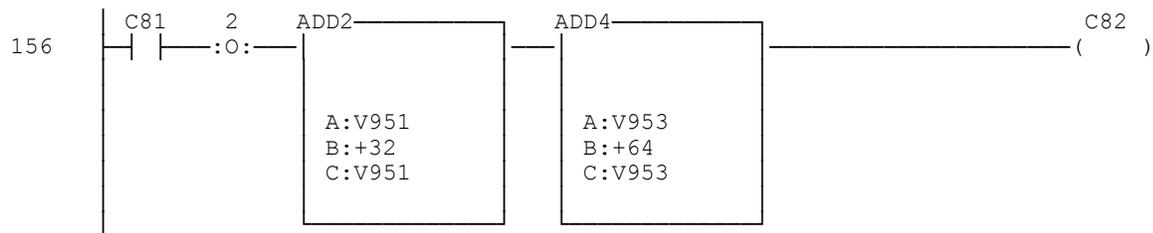
The counter contains the number of data segments to be transferred. This number should be identical
to the number in the counter used to increment data transfer in standby mode.
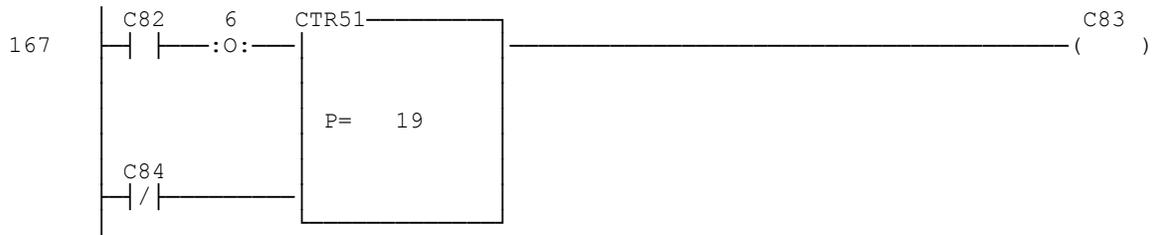
```
       |  C82    6    CTR51────────────────┐                                       C83
 167   ├──┤ ├────:O:───┤                   │────────────────────────────────────────(    )
       |                │                   │
       |                │  P=     19        │
       |                │                   │
       |  C84           │                   │
       ├──┤/├───────────┘                   │
       |                └───────────────────┘
```

Below are momentary and latched outputs indicating that the transfer of data is complete that can be
used in other parts of the logic.

```
       |  C83    7                                                                  C84
 173   ├──┤ ├────:O:──────────────────────────────────────────────────────────────(    )
       |
       |  C83    8                                                                  C85
 176   ├──┤ ├────:O:───┐                                                           (    )
       |               │
       |  C85          │
       ├──┤ ├──────────┘
```
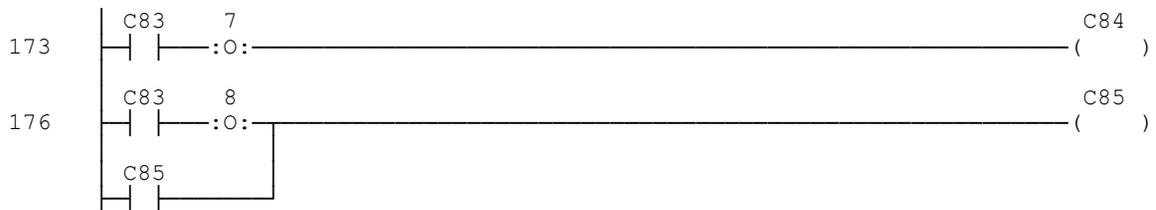
HARDWARE SPECIFICATIONS

Environmental

     Operating Temperature: 0º to 60º C (32º to 185º F)
     Storage Temperature: -40º to 85º C (-40º to 185º F)
     Relative Humidity: 5% to 95%, non-condensing

Backplane Power Consumption:

     6.0 watts @ 5 VDC

Module Size:

     Double Wide 505 I/O

Configuration Port:

     Male DB9, RS-232
     1500 VDC Isolation - channel to PLC
     9600 baud
     Odd Parity, 1 stop bit, 7 data bits

Switched I/O Ports

     Siemens® Front Access Connector

Switched Communications/Auxiliary Ports:

     Siemens® Front Access Connector
     Maximum Voltage: 24 VAC/DC
     Maximum Current: 0.5 amp
     Resistive load only

Switchover Characteristics:

     PLC Timeout Value: 25 - 500ms (user defined)
     RPM Switchover Time: <50ms
     Maximum RBC Switchover Re-connect Time: <50ms

LIMITED PRODUCT WARRANTY

CTI warrants that this CTI Industrial Product shall be free from defects in material and workmanship for a period of one (1) year after purchase from CTI or from an authorized CTI Industrial Distributor. This CTI Industrial Product will be newly manufactured from new and/or serviceable used parts which are equal to new in the Product.

Should this CTI Industrial Product fail to be free from defects in material and workmanship at any time during this (1) year warranty period, CTI will repair or replace (at its option) parts or Products found to be defective and shipped prepaid by the customer to a designated CTI service location along with proof of purchase date and associated serial number. Repair parts and replacement Product furnished under this warranty will be on an exchange basis and will be either reconditioned or new. All exchanged parts or Products become the property of CTI. Should any Product or part returned to CTI hereunder be found by CTI to be without defect, CTI will return such Product or part to the customer.

This warranty does not include repair of damage to a part or Product resulting from: failure to provide a suitable environment as specified in applicable Product specifications, or damage caused by an accident, disaster, acts of God, neglect, abuse, misuse, transportation, alterations, attachments, accessories, supplies, non-CTI parts, non-CTI repairs or activities, or to any damage whose proximate cause was utilities or utility like services, or faulty installation or maintenance done by someone other than CTI.

Control Technology Inc. reserves the right to make changes to the Product in order to improve reliability, function, or design in the pursuit of providing the best possible Product. CTI assumes no responsibility for indirect or consequential damages resulting from the use or application of this equipment.

THE WARRANTY SET FORTH ABOVE IN THIS ARTICLE IS THE ONLY WARRANTY CTI GRANTS AND IT IS IN LIEU OF ANY OTHER IMPLIED OR EXPRESSED GUARANTY OR WARRANTY ON CTI PRODUCTS, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE AND IS IN LIEU OF ALL OBLIGATIONS OR LIABILITY OF CTI FOR DAMAGES IN CONNECTION WITH LOSS, DELIVERY, USE OR PERFORMANCE OF CTI PRODUCTS OR INTERRUPTION OF BUSINESS, LOSS OF USE, REVENUE OR PROFIT. IN NO EVENT WILL CTI BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH MAY VARY FROM STATE TO STATE.

REPAIR POLICY

In the event that the Product should fail during or after the warranty period, a Return Material Authorization (RMA) number can be requested orally or in writing from CTI main offices. Whether this equipment is in or out of warranty, a Purchase Order number provided to CTI when requesting the RMA number will aid in expediting the repair process. The RMA number that is issued and your Purchase Order number should be referenced on the returning equipment's shipping documentation. Additionally, if the product is under warranty, proof of purchase date and serial number must accompany the returned equipment. The current repair and/or exchange rates can be obtained by contacting CTI's main office at 1-800-537-8398.

When returning any module to CTI, follow proper static control precautions. Keep the module away from polyethylene products, polystyrene products and all other static producing materials. Packing the module in its original conductive bag is the preferred way to control static problems during shipment. Failure to observe static control precautions may void the warranty. For additional information on static control precautions, contact CTI's main office at 1-800-537-8398.