Common Memory Module

(M/N 57C413B)
(M/N 57C423)

Industrial

CONTROLS

Instruction Manual J-3636-2

RELIANCE ELECTRIC

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

DANGER

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

WARNING

INSERTING OR REMOVING A MODULE MAY RESULT IN UNEXPECTED MACHINE MOTION. POWER TO THE MACHINE SHOULD BE TURNED OFF BEFORE INSERTING OR REMOVING THE MODULE. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

WARNING

THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

ReSource™ and AutoMax™ are trademarks of Reliance Electric Company or its subsidiaries.
Reliance® is a registered trademark of Reliance Electric Company or its subsidiaries.

Table of Contents

Appendices

List of Figures

1.0 INTRODUCTION

This product described in this instruction manual are manufactured or distributed by Reliance Electric Industrial Company.

The Common Memory module (57C413B and 57C423) is required in slot 0 of a DCS 6200/AutoMax rack that contains more than one Processor module. The Common Memory module stores the configuration data that must be shared among Processors, such as definitions of physical I/O. This frees Processor module memory for application tasks. The Common Memory module also arbitrates the Processors' access to the bus.

The Common Memory module may also be placed in slot 0 of a single-Processor rack. As in the case above, the configuration data which normally resides on the Processor module in a single-Processor rack is stored on the Common Memory module. When the Common Memory module is placed in any even slot other than 0, it is used for user-definable data storage only. This mode is useful if you need to control explicitly the physical allocation of memory (e.g., to define consecutive registers for shift register instructions). Note, however, that it is not possible to define arrays in this mode.

The module incorporates bus arbitration logic, a system watchdog for multiprocessing, and battery-backed RAM for data storage. When used in slot 0, M/N 57C413B has 128K bytes (64K registers) of memory; M/N 57C423 has 256K bytes (128K registers) of memory. Both versions of the module make available 128K bytes only when used in any even slot other than 0. An on-board lithium battery and a super-capacitor protect the Common Memory module from power failures. Note that the battery backup is designed to maintain the contents of RAM only. It is not a source of uninterruptible power. Should the rack lose power, the on-board battery can maintain the contents of RAM for a minimum of 600 days.

This manual describes the functions and specifications of the module. It also includes a detailed overview of installation and servicing procedures, as well as examples of programming methods. Unless specifically noted otherwise, the information in this manual describes both the M/N 57C413B and M/N 57C423 Common Memory modules.

Related publications that may be of interest:

- J-3600 ReSource AutoMax PROGRAMMING EXECUTIVE INSTRUCTION MANUAL VERSION 1.0
- J-3649 AutoMax CONFIGURATION TASK MANUAL
- J-3650 AutoMax PROCESSOR MODULE INSTRUCTION MANUAL
- J-3675 AutoMax ENHANCED BASIC LANGUAGE INSTRUCTION MANUAL
- J-3676 AutoMax CONTROL BLOCK LANGUAGE INSTRUCTION MANUAL
- J-3677 AutoMax LADDER LOGIC LANGUAGE INSTRUCTION MANUAL

- C-3681 ReSource AutoMax PROGRAMMING EXECUTIVE INSTRUCTION MANUAL VERSION 2.0
- C-3750 ReSource AutoMax PROGRAMMING EXECUTIVE INSTRUCTION MANUAL VERSION 3.0
- IEEE 518 GUIDE FOR THE INSTALLATION OF ELECTRICAL EQUIPMENT TO MINIMIZE ELECTRICAL NOISE INPUTS TO CONTROLLERS FROM EXTERNAL SOURCES

2.0 MECHANICAL/ELECTRICAL DESCRIPTION

The following is a description of the faceplate LEDs and the basic circuit functions on the module.

2.1 Mechanical Description

The Common Memory module is a printed circuit board assembly that plugs into the backplane of the DCS 5000/AutoMax rack. The module consists of a printed circuit board, a faceplate, and a protective enclosure. The faceplate contains tabs at the top and bottom to simplify removing the module from the rack. On the back of the module are two edge connectors that attach to the system backplane. Module dimensions are listed in Appendix A. See figure 2-1 for module faceplates.

The faceplate contains two green status lights. The upper status light, labeled "BAT.OK", indicates whether the on-board battery is providing sufficient voltage to retain the contents of RAM (ON) or should be replaced (OFF). See section 3.3 for directions on replacing the battery and Appendix A for battery specifications. The lower status light, labeled "SYSTEM WATCHDOG", is lit only when the module is in slot 0, the module has passed its power-up diagnostics, and the bus arbitration clock is present on the backplane. If the status light is off, it indicates the module is not operational, either because it is malfunctioning or because it is located in a slot other than 0 and is providing data storage only.

2.1.1 Checking the Status of the On-Board Battery

The status of the Common Memory module on-board battery can be checked in the following ways:

If the Common Memory module is in slot 0:

- The "BAT. OK" LED on the faceplate will be ON to indicate the battery is providing sufficient voltage to maintain the contents of RAM memory and OFF if the battery should be replaced.
- The Info/Log Processor Information Display from the ON LINE menu of the AutoMax Programming Executive will show the battery status.

If the Common Memory module is not in slot 0 (i.e., it is in any other even-numbered slot and is being used for data storage only), the battery status is indicated only by the status of the "BAT.OK" LED on the module faceplate.

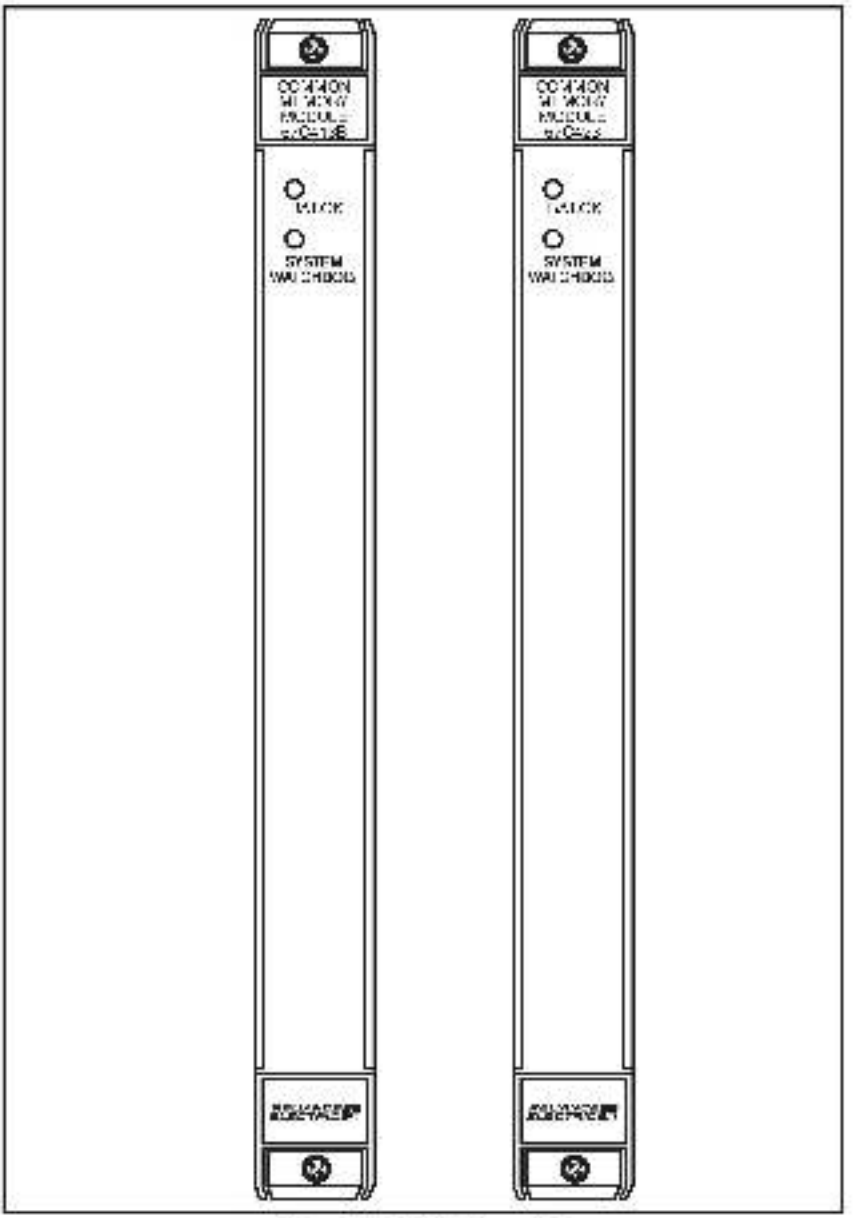


Figure 2-1 - Module Faceplate

2.2 Electrical Description

The Common Memory module incorporates the bus arbitration logic required when there are two or more Processor modules in a rack. Note that bus arbitration logic is enabled only when the module is in slot 0. The bus arbitration logic will support up to a maximum of four Processors located in slots 1-4.

The bus arbitor resolves the contention problem that arises when two or more Processors attempt to access the backplane bus at the same time. Bus arbitration logic guarantees that every Processor requesting the bus will be given a turn on the bus before any other Processor can access it a second time. If two Processors attempt to use the bus at the same time and neither one has previously accessed the bus, the Processor module located in the over-numbered slot will be permitted to access the bus first.

The super-capacitor on the Common Memory module can be charged to more than 90% of its rated capacity in approximately 5 minutes. It is typically capable of retaining the contents of RAM memory for approximately 400 minutes should the "BATLOK" light go out and power is removed from the module. (Removing or replacing the Common Memory module may affect tasks and variables in the rack. Appendix E describes the effect on tasks and variables in a DCS 5000 rack. Appendix F describes the effect on tasks and variables in an AutoMax rack.)

The module also contains a watchdog timer that is used to detect Processor failures in a multi-Processor system. If a Processor is unable to reset the watchdog timer, the timer will generate an interrupt to notify the other Processors in the rack of the failure. A "1.6" will appear on the faceplate of all Processor modules except the one that timed out. The watchdog timer is enabled only when the module is in slot 0.

Power-up diagnostics for the module are run by a Processor module. Diagnostics are performed on the RAM memory, control registers, and watchdog timers.

3.0 INSTALLATION

This section describes how to install and remove the module and the on-board battery. Note that removing or replacing the Common Memory module may affect tasks and variables in the rack. Appendix E illustrates how tasks and variables are affected in a DCS 6000 rack. Appendix F illustrates how tasks and variables are affected in an AutoMax rack.

DANGER

THE USER IS RESPONSIBLE FOR CONFORMING TO THE NATIONAL ELECTRIC CODE AND ALL OTHER APPLICABLE CODES WITH RESPECT TO WIRING, GROUNDING, DISCONNECTS, AND OVERCURRENT PROTECTION. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

3.1 Initial Installation

Use the following procedure to install the module:

Step 1. Turn off power to the rack and all connections.

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

Step 3. Activate the on-board battery. When viewing the Common Memory module from the front, you can access the battery through the opening in the right wall portion of the protective enclosure. Activate the battery by taking it out of its holder and removing the tape that covers it. Replace the battery in its holder. Make certain that the battery is facing in the proper direction, i.e., the end marked "+" on the battery is facing the end marked "+" on the battery holder. For maximum battery life, you should not remove the tape from the battery unless you intend to turn power on to the module immediately.

Step 4. Insert the module into the desired slot in the rack. Use a screwdriver to secure the module into the slot. To enable bus arbitration logic, the Common Memory module must be placed in slot 0. For M/N 57C413H, the slot to the right of the Common Memory module must contain either a Processor module or be empty because the Common Memory module takes up two slots of logical address space. For M/N 57C423, the 3 slots to the right of the module must contain either a Processor module or be empty because the module takes up four slots of logical address space when in slot 0. Refer to figure 3.1.

To serve as data storage, the module must be placed in an even numbered slot (2, 4, 6, 8, 10, 12, 14). The slot to the right of the Common Memory module must be empty because the module (both 57C413B and 57C423) takes up two slots of logical address space when used in this way. Refer to figure 3.1

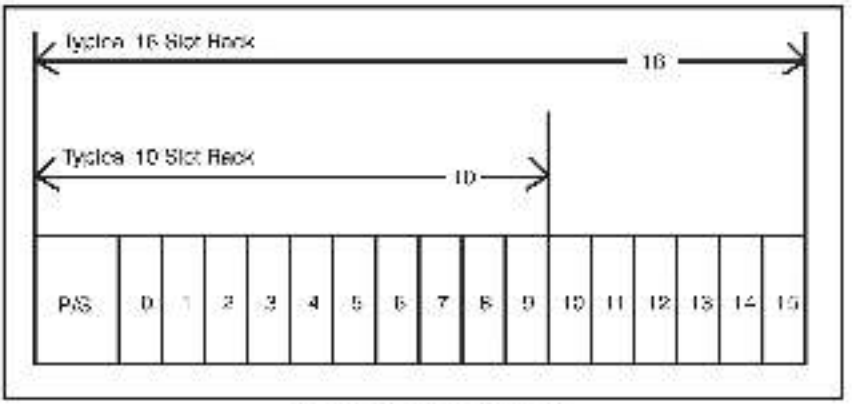


Figure 3.1 - Rack Slot Numbers

Step 5. Turn on power to the rack.

Step 6. Verify the installation. If the Common Memory module was placed in slot 0, the powerup diagnostics performed automatically by a Processor module should verify that the module is operational.

If the Common Memory module is placed in any even slot other than 0, it must be manually tested like a standard I/O module. Connect the personal computer to the system and run the Programming Executive Software.

Stop all programs that may be running.

Use the (I) MONITOR function and enter the module slot number and any valid register number (0-32767). Also enter the slot number +1 and any valid register number (0-32767) to test the upper 32K register of memory. Verify that data can be read from and written to the registers. To ensure your application task does not access old or incorrect data, you may want to create a RAM C task to write zeroes to each register location before you run any other application tasks. This is only required if you do not initialize values in your application tasks.

3.2 Module Replacement

Removing or replacing the Common Memory module may affect tasks and variables in the rack. Before beginning the procedure below, refer to Appendix E for GE 90-30 racks or Appendix F for AutoMax racks.

Use the following procedure to replace a module:

Step 1. Stop any application tasks that may be running.

Step 2. Turn off power to the rack and all connections.

Step 3. Use a screwdriver to loosen the screws that hold the module in the rack. Remove the module from the slot in the rack.

Step 4. Place the module in the anti-static bag it came in, being careful not to touch the connectors on the back of the

module. Place the module in the cardboard shipping container.

Step 5. Take the new module out of the anti-static bag, being careful not to touch the connectors on the back of the module.

Step 6. Activate the battery by taking it out of its holder and removing the tape that covers it. Replace the battery in its holder. Make certain that the battery is facing in the proper direction, i.e., the end marked "–" on the battery is facing the end marked "+" on the battery holder.

Step 7. Insert the module into the desired slot in the rack. Use a screwdriver to secure the module into the slot.

Step 8. If you are replacing the M/N 57C415 (Common Memory module that required an external battery backup with M/N 57C415-4 or 57C423, you may now remove the battery backup from the power supply. The battery backup is not required when the (Common Memory module is placed in an AutoMax rack containing Processor modules 57C430, 57C431, 57C435 or later. **DO NOT REMOVE THE BATTERY BACKUP FROM RACKS CONTAINING DCS PROCESSOR M/N 57C407.** The external battery is required to back up this Processor.

Step 9. Turn on power to the rack.

WARNING

THE BATTERY USED WITH THIS DEVICE MAY PRESENT A HAZARD IF MISTREATED. DO NOT RECHARGE, DISASSEMBLE, HEAT ABOVE 100°C (212°F), INCINERATE, OR SWALLOW. REPLACE BATTERY WITH RELIANCE ELECTRIC M/N 57C365 ONLY. DISPOSE OF USED BATTERY PROMPTLY. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

3.3 On-Board Battery Replacement

See section 3.2 for a list of the possible reasons that the "BAT. OK" light on the faceplate can shut off. If you need to replace the battery, the super-capacitor will typically provide 400 minutes of backup power between the time the "BAT.OK" light goes off and power is removed from the rack, and the time you insert and activate the new battery.

Note that the super capacitor on the module is charged by the power supply. Therefore, be sure power has been turned on to the module for at least 5 minutes to ensure the super capacitor is sufficiently charged to retain the contents of RAM memory when the battery is removed.

Use the following procedure to replace the battery on the Common Memory module.

Step 1. Stop any application tasks that may be running.

Step 2. Turn off power to the system.

Step 3. Loosen the screws that hold the module in the rack. Remove the module from the slot in the rack, being careful not to touch the connectors on the back of the module.

Step 4. Take the old battery out of the holder. Remove the tape from the new battery and insert the battery in the holder. Make certain that the battery is facing the proper direction, i.e. the end marked "+" on the battery is facing the end marked "+" on the battery holder.

Step 5. Re-insert the module into the correct slot in the rack. Use a screwdriver to secure the module into the rack.

Step 6. Turn on power to the rack. The "BAT. OK" LED should be lit.

4.0 PROGRAMMING

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

The Common Memory module has two distinct modes of operation depending on the slot this is in. If the module is located in slot 0, the bus arbitration logic and watchdog timer are enabled and the module is controlled entirely by the Intermost Processor module. If it is in any other even slot, the bus arbitration logic and watchdog timer are disabled and the module provides user configurable data storage only. This mode is useful if you need to control explicitly the physical location of memory, e.g., to define consecutive registers for shift register instructions. Note, however, that it is not possible to define arrays in this mode.

4.1 Configuration

Before creating AutoMax application tasks, the user must configure the hardware in the system. Configuration is the process of assigning variable names to I/O and memory locations in modules located in the rack. The configuration process makes it possible to create application tasks that reference variable names instead of fixed locations. This information must be loaded onto the Processor modules in the rack before application tasks can run.

This section outlines some basic system parameters that determine how to configure the Common Memory module and reference memory stored on the module. Unless specifically noted otherwise, the information below applies to both M/N 57C413D and M/N 57C423 Common Memory modules as they operate in both DCS 5000 and AutoMax systems. Recall that when other modules are in an even slot other than 0, only 64K x 16 words of memory (two logical slots in the rack) is available for configuration by the user.

4.1.1 Variable Control and Access

To understand the effect of a Common Memory module in a rack, it is important to understand variable control in AutoMax systems. There are two types of variable control in AutoMax systems: common and local. "Control" refers to whether the variable will be accessed exclusively by one application task (local), or whether it must be accessible for reads or writes by more than one application task in the rack (common). Defining variable control is accomplished as follows:

local variable:	a)	default control type; not defined in configuration
common variable:	a)	defined in configuration; there are two types, memory and I/O. If memory, can be volatile or non-volatile

Variable access in an application task is enabled by declaring the variable common or local in application tasks that reference the variable. See the appropriate programming language reference manuals for specific information on declaring variables local or common.

4.1.2 Variable Storage

Local variables are stored on the Processor module in which the application task that declares them will run. Common variables are stored on the Processor module, unless there is a Common Memory module in slot 0 of the rack, in which case the common variables are stored on the Common Memory module instead. This frees up memory on the Process module for application tasks. Because common variables can be stored on either the Processor module or a Common Memory module (if it is located in slot 0), a variable defined in the configuration and declared common in an application task is not necessarily one that is stored on the Common Memory module. In other words, when used to describe a variable in Autoview, "common" refers strictly to variable control, not to variable storage.

A Common Memory module located in a non-0 slot is treated by the system as two "general" I/O modules with 32K registers each (32K in the actual slot location and 32K in the slot to the immediate right). Because all I/O variables are considered common by the system, variables defined on this module are treated like any other common variables for storage purposes: they are stored on the Processor module in a single-Processor rack or on the Common Memory module in slot 0 (if it is installed).

When the Common Memory module is in slot 0, the allocation of its memory is under system control. During configuration, the user specifies only the name and type (e.g., boolean) of each common variable. These variables are accessible to application tasks and Programming Executive functions (e.g., monitoring) by name only.

If the Common Memory module is in any other even slot, memory on the module is allocated by register/bit address exactly as specified by the user during configuration. These variables are accessible to application tasks and Programming Executive functions by name or register/bit address.

4.1.3 Configuration Method

The method used to configure the Common Memory module depends both upon the Programming Executive software version being used, and on the slot number (0 or any other even slot) of the module. For Programming Executive software V3.0 and later, the module is configured like all other modules, i.e., using the Rack Configuration menu option. See GFK-3750, the Programming Executive V3.0 instruction manual for information on configuring the module in both slot 0 and other even slots.

For Programming Executive software V2.1 and earlier, the Common Memory module is configured in a special type of application task called the configuration task, one of which is required per rack. See Appendix C for detailed instructions on configuring the module if you are using V2.1 and earlier of the Programming Executive software.

4.2 Restrictions

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

4.2.1 Rack Slot Restrictions

A Common Memory module is required in slot 0 in a rack that contains more than one Processor module. The slot to the right of the Common Memory module must either be empty or contain a Processor module for M/N 57C416. For M/N 57C424, the 4 slots to the right of the module must either be empty or contain a Processor module.

This module may also be plugged into even slots (2,4,6,8,10,12, or 14). Since the 128K bytes (64K registers) of memory available for each module in a non-0 slot require two slots of address space, the slot to the right of the Common Memory module must be empty.

4.2.2 Remote Racks

This module cannot be used in a remote I/O rack.

5.0 DIAGNOSTICS AND TROUBLESHOOTING

This section explains how to troubleshoot the Common Memory module.

5.1 The "SYSTEM WATCHDOG" LED is Off

Problem: The green "SYSTEM WATCHDOG" LED on a Common Memory module located in slot 0 is off, and a Processor module in the rack displays codes 4.0 through 4.6. These error codes mean that the Common Memory module has failed one of its power-up diagnostics.

Systematically swap out the Common Memory module and the Processor module(s). Make certain power is off before removing any module from the rack. After each swap, if the problem is not corrected, replace the original item before going on to the next item. If the problem persists, take all of the modules except the Common Memory module and one Processor module out of the rack. If the problem is now corrected, another module in the rack is causing the problem. Replace the remaining modules one at a time until the problem reappears. If none of these tests reveals the problem, try replacing the backplane.

5.2 The "BAT.OK" LED is Off

Problem: The "BAT. OK" LED on the Common Memory module faceplate is off. The possible causes of this problem are the following:

- The tape covering the battery has not been removed
- The battery is not facing in the proper direction
- The battery is missing
- The battery is malfunctioning
- The power supply is malfunctioning

To correct the problem, first, turn off power to the rack. Refer to steps 1-3 in section 3.3 for instructions on taking the Common Memory module out of the rack to inspect the battery. If the tape is still covering the battery, remove it. If the battery is missing or not facing in the proper direction, insert the battery with the "–" end facing the "–" marking on the battery holder. If none of the above actions correct the problem, replace the battery.

5.3 Incorrect Data

Problem: The output is either always off, always on, or different than expected. The possible causes of this error are a module in the wrong slot, a malfunctioning module, or a programming error. Use the following procedure to isolate the problem:

Step 1. Verify that the module is in the correct slot.

Refer to figure 3.1. Verify that the correct number of slots to the right of the Common Memory module are either

empty or contain a Processor module. If this is not the case, multiple modules will respond to references.

If the module is not in slot 0, i.e., it is serving as data storage, verify that the definition statements in the configuration are associated with the correct slots.

Step 2. Verify that the variables are being referenced correctly.

Verify that all tasks that reference variables on this module declare them COMMON. Only one task should be writing to any one variable.

Step 3. Verify that the module can be accessed.

Connect the personal computer to the system and run the Programming Executive Software.

Stop all programs that may be running.

If the module is in slot 0, use the VARIABLE MONITOR to read and write to the variable by name. If the variable cannot be accessed correctly, proceed to step 4.

If the module is in any even slot other than 0, use the I/O MONITOR to read and write to the desired register. If the register cannot be accessed correctly, proceed to step 4.

Step 4. Verify that the hardware is working correctly.

Systematically swap out the Common Memory module and the Processor module(s). Make certain power is off before removing any module from the rack. After each swap, if the problem is not corrected, replace the original item before going on to the next item. If the problem persists, take all of the modules except one Processor module and the Common Memory module out of the backplane. If the problem is now corrected, one of the other modules in the rack is malfunctioning. Reinsert the other modules one at a time until the problem reappears. If none of these tests reveals the problem, replace the rack/backplane.

5.4 Bus Error

Problem: A "31" or "51" through "58" appears on the Processor module's LED. This error message indicates that there was a bus error when the system attempted to access the module. The possible causes of this error are a missing module, a module in the wrong slot, or a malfunctioning module. Use the following procedure to isolate a bus error.

Step 1. Verify that the Common Memory module is in the correct slot.

If the module is in slot 0, proceed to step 4. If the module is in any other even slot, proceed to step 2.

Step 2. Verify that the module can be accessed.

Connect the personal computer to the system and run the Programming Executive Software.

Stop all programs that may be running.

The cause of a "31" error message is most likely an error in an address calculation performed by an application

program. Verify that any IOREAD functions and IOWRITE statements are using valid addresses.

If IOREAD functions and IOWRITE statements are using valid addresses or if the error message in question was "51" through "58", use the I/O MONITOR function to display the register on the memory module. Remember to specify the correct logical slot.

If you can monitor the inputs, the problem lies in the application software (proceed to step 3). If you cannot monitor the inputs, the problem lies in the hardware (proceed to step 4).

Step 3. Verify that the variable names are consistent with what is used in the application task.

If you are using the AutoMax Programming Executive version 3.1 or earlier, verify that the variables defined by IODEF and MEMDEF statements in the configuration task match the variable names used in the application task.

In AutoMax Version 3.0 or later, verify the information on the Variable Configurator "form" (screen) is consistent with the variable names used in the application task.

Step 4. Verify that the hardware is working correctly.

Systematically swap out the Common Memory module, the Processor module(s) and the backplane. Make certain power is off before removing any module from the rack. After each swap, if the problem has not been corrected, replace the original item before swapping out the next item.

Appendix A

Technical Specifications

Ambient Conditions

- Storage temperature: −40°C - 85°C
- Operating temperature: 0°C - 60°C
- Humidity: ≤ 90% non condensing

Maximum Module Power Dissipation

- 5.3 Watts

Dimensions

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

System Power Requirements

- +5 volts: 1050 mA

Battery Specifications

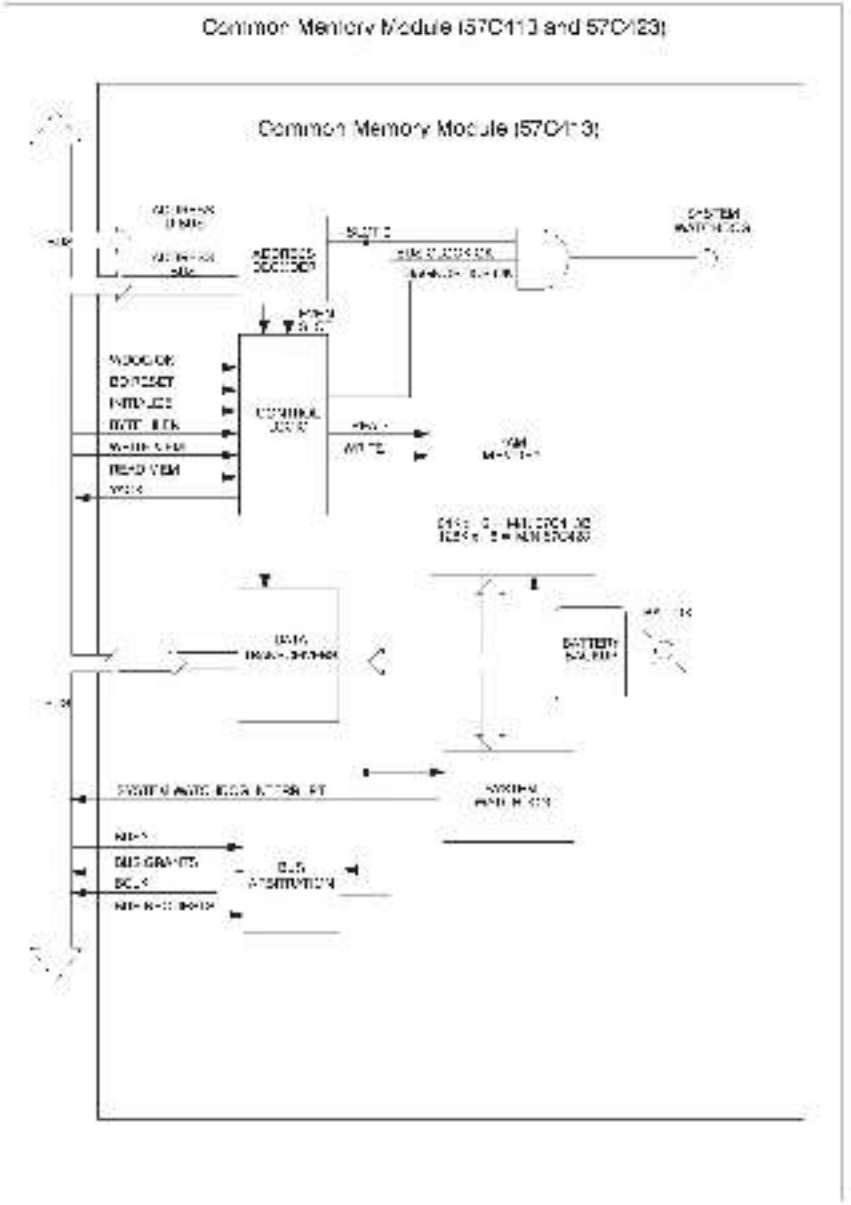
- Type: Lithium
- Size: AA
- Voltage: 3.6 Volts
- Amp Hrs.: 2.0

Memory Retention

- Minimum hold-up with battery: 600 days
- Typical hold-up with battery: 2000 days
- Minimum hold-up without battery: 130 minutes
- Typical hold-up without battery: 400 minutes
- Maximum charge-up time: 6 minutes

Appendix B

Module Block Diagram



Appendix C

Configuring the Common Memory Module in DCS 5000 or AutoMax V 2.1 or Earlier Systems

In DCS 5000 or AutoMax Version 2.1 or earlier systems, a configuration task must be created and loaded into the Processor(s) in the rack before any application task can be executed. The configuration task defines all common variables, i.e., variables that are accessible to more than one application task in the rack (physical input/output and memory variables). The location of the Common Memory module in the rack determines the type of statement required to define these variables. This appendix describes the configuration task statements required to configure the Common Memory module in DCS 5000 or AutoMax 2.1 Systems.

Common Memory Module Located in Slot 0

When the Common Memory module is in slot 0, the allocation of data storage on the module is automatic and completely under the control of the operating system. This means you can reserve storage space for the common variable, but not specify its actual address. You define common variables in the configuration task by using either the MEMDEF or NVMEMDEF statement. These statements are described in detail below.

MEMDEF Statement

The MEMDEF statement is used to define common variables that will not retain their current values in the event of a power loss. On power up, these volatile variables are set to zero if there is a battery backup. However, if there is no battery backup, all common data (variables) stored on the Common Memory module is lost. This statement may be used to define any valid data type, i.e., real, integer, double integer, boolean, and string. Common variables can then be accessed by any task that declares their variable names COMMON.

The following example defines a single precision integer (16 bits) with the name 'WINDOW%', and a boolean (1 bit) with the name 'STOPPS@'. Note the terminating characters are used to specify data type. See J-3649 for more information on data types.

```
1000 MEMDEF WINDOW%, STOPPS@
```

The MEMDEF statement also allows you to define array variables. The following example defines an array of 20 (0–19) single-precision integers with the name 'SIZES%'.

```
1650 MEMDEF SIZES%(19)
```

NVMEMDEF Statement

The NVMEMDEF statement is used to define common variables that will retain their current value in the event of a power loss (i.e., non-volatile variables. However, if the battery fails, all common data (variables) stored on the Common Memory module is lost. This statement may be used to define variables of any valid data type which can be accessed by any task that declares them COMMON.

The following example defines a single precision integer (16 bits) with the name 'WINDOW%', and a boolean (1 bit) with the name 'STOPPS@'.

Appendix C

(Continued)

1000 NVMEMDEF WINDOW%, STOPPB@

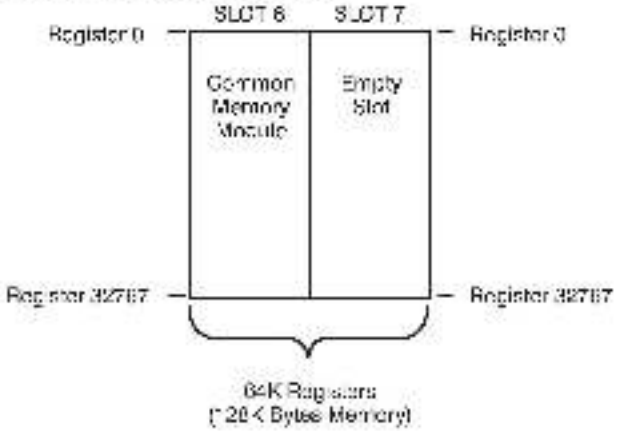
The NVMEMDEF statement also allows you to define array variables. The following example defines an array of 20 (0-19) single-precision integers with the name "SIZES%".

1650 NVMEMDEF SIZES%(19)

Common Memory Module Located in Any Even Slot Other Than 0

This section describes how to configure the Common Memory module when it is located in any even slot other than 0. In this situation the Common Memory module provides user configurable data storage only. In other words, you can specify the exact address at which to store data.

The slot to the right of the Common Memory module must be empty because the module contains 64K user-configurable registers, or the equivalent of the address space of two slots. Therefore, if the Common Memory module is placed in slot 6, the first 32K registers are accessed as slot 6. The second 32K registers are addressed as slot 7. Refer to the figure below.



32 Bit Register Reference Format

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

nnnnn IODET SYMBOLIC_NAME[SLOT=s, REGISTER=r]

When referenced as a double precision integer of 32 bits, register "r" is the high order 16 bits and register "r+1" is the low order 16 bits.

Appendix C

(Continued)

16 Bit Register Reference Format

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

nnnnn ODET SYMBOLIC_NAME%[SLOT=s, REGISTER=r]

Bit Reference Format

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

nnnnn ODET SYMBOLIC_NAME@[SLOT=s, REGISTER=r BIT=b]

Where:

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

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

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

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

SLOT = Slot number that the module is plugged into. This number may range from 2-14. If you are referencing registers on the upper half of the card remember to use the slot number of the empty slot to the right of the module in the rack.

REGISTER = Specifies the register that is being referenced. This number may range from 0-32767 for both slots.

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

Examples of Configuration Statements

The following statement assigns the symbolic name WINDOW% to register 2000 and 2001 on the memory module located in slot 10:

1000 ODEF WINDOW% [SLOT=10, REGISTER=2000]

The following statement assigns the symbolic name POSITION% to register 250 of the input module located in slot 4:

1020 ODEF POSITION%[SLOT=4, REGISTER=250]

The following statement assigns the symbolic name LIGHT@ to bit 9 of register 10 on the input module located in slot 6:

2050 ODEF LIGHT@[SLOT=6, REGISTER=10 BIT=9]

Appendix D

Summary of Common Memory Module Features

M/N 57C413

This module has 128K bytes of memory and requires external battery backup (M/N 57C492) to maintain the contents of memory in the event of a power failure.

M/N 57C413B

This module has 128K bytes of memory and an on-board lithium battery to maintain the contents of memory in the event of a power failure.

M/N 57C423

This module has 256K bytes of memory available when plugged into slot 0, and 128K bytes available when plugged into any other non-zero slot. An on-board lithium battery maintains the contents of memory in the event of a power failure. This module is supported only by the following Program Executive version: 2.1E and 3.1C and later.

Appendix E

How Removing/Replacing The Common Memory Module Affects Tasks and Variables in a DCS 5000 Rack

The following tables illustrate how removing or replacing the Common Memory module affects tasks and variables in a DCS 5000 rack. (Refer to Appendix F for information regarding the effect on tasks and variables in an AutoMax rack.) In each case, assume a BASIC task has written to variables in both the Processor (local) and the Common Memory module (common). Also assume the task was stopped before power was removed from the rack.

The following information applies only if an external battery backup (57C462) is connected to the power supply. The external battery backup is required for DCS 5000 Processor memory backup. If the battery backup is not used, all tasks are deleted when power is removed from the rack.

Common Memory Module Located In Slot 0

For the following, assume the Common Memory module is in slot 0 and a Processor module is in slot 1.

Action	App. Task Deleted?	App. Task Status	Status of Data		
			Common		Local
			Name	Val	
Replace 57C413A with [illegible] 57C413A. At power up	Yes*	N/A	N/A	N/A	N/A
Replace 57C413A with New 57C413A or 57C421. At power up	NO	UNINSTALLED*	N/A	N/A	N/A
Replace 57C413A or 57C421A with new 57C413A or 57C421. (module may contain data from another application) At power up	Error Code 13 appears on Processor LED**				
Replace battery in 57C413A or 57C421A and remove module from rack. At power up***	NO	STOP	Data 04	Zero	Data 04

* You must re-load the configuration and application tasks into the rack. This procedure is described in the Programming Executive instruction manual (I-E96300).

** Select DOWNLOAD from the ON LINE menu to download the configuration and application tasks. Select ERROR CLEAR to clear the LED fault code on the Processor module. Refer to I-E96300 for more information about the ON LINE menu.

*** Assumes super-capacitor has maintained memory on the module during the time required to replace the battery.

Appendix E

(Continued)

The Common Memory Module Is In Any Even Slot Other Than 0

In the following, assume the Common Memory module is in slot 2 and the single Processor module in the rack is in slot 1. Recall that in this configuration, common variables, as well as local variables, are being stored on the Processor module.

Action	App. Task Deleted?	App. Task Status	Status of Data				
			Common				Local
			NonVol	Vol	I/O		
Replace 57C413A with Same or new 57C413A At power up	NO	STOP	Data OK	Zero	Lost (Random)	Data OK
Replace 57C413A with New 57C413B or 57C423 At power up	NO	STOP	Data OK	Zero	Zero*	Data OK
Replace 57C413A with programmed 57C413B or 57C423 (module may contain data from another application) At power up	NO	STOP	Data OK	Zero	Value from programmed module at same address*	Data OK
Replace 57C413B or 57C423 with New 57C413B or 57C423 At power up	NO	STOP	Data OK	Zero	Zero*	Data OK
Replace 57C413B or 57C423 with programmed 57C413B or 57C423 (module may contain data from another application) At power up	Error Code [illegible] appears on Processor LEDs**					
Replace battery on 57C413B or 57C423 and return same module to rack At power up***	NO	STOP	Data OK	Zero	Data OK	Data OK

* To ensure your application task does not access old or incorrect data, create a BASIC task to write zeroes to each register location before you run any other application tasks.

** Select DOWNLOAD from the ON LINE menu to download the configuration and application tasks. Select ERROR CLEAR to clear the LED fault code on the Processor module. Refer to J-3630 for more information about the ON LINE menu.

*** Assumes super-capacitor has maintained memory on the module during the time required to replace the battery.

Appendix F

How Removing/Replacing The Common Memory Module Affects Tasks and Variables in an AutoMax Rack

The following tables illustrate how removing or replacing the Common Memory module affects tasks and variables in an AutoMax rack. (Refer to Appendix E for information regarding the effect on tasks and variables in a DCS 5000 rack.) In each case, assume a BASIC task has written to variables in both the Processor (local) and the Common Memory module (common). Also assume the task was stopped before power was removed from the rack.

Common Memory Module Located In Slot 0

For the following, assume the Common Memory module is in slot 0 and a Processor module is in slot 1.

Action	App. Task Restart?	App. Task Status	State of Data		
			Common		
			Non-Vol	Vol	Local
Replace 57C413A with Same or new 57C413A. All power up	YES*	N/A	N/A	N/A	N/A
Replace 57C413A with New 57C413B or 57C423 All power up	YES*	N/A	N/A	N/A	N/A
Replace 57C413B or 57C423 with new or used 57C413 or 57C423 (may be used in another application) All power up	YES*	N/A	N/A	N/A	N/A
Replace battery on 57C413B or 57C423 and main power is off All power up**	NO	STOP	Data OK	Zero	Data OK

* You must re-load the configuration and application tasks onto the rack. This procedure is described in the AutoMax Programming Executive instruction manual. Refer J-3750 if you are using AutoMax Version 3.0 or later. Refer to J-3567 if you are using AutoMax Version 2.1 or earlier.

** Assumes super-capacitor has maintained memory on the module during the time required to replace the battery.

Appendix F

(Continued)

The Common Memory Module Is In Any Even Slot Other Than 0

In the following, assume the Common Memory module is in slot 2 and the single Processor module in the rack is in slot 1. Recall that in this configuration, common variables, as well as local variables, are being stored on the Processor module.

Action	App. Task Deleted?	App. Task Status	Status of Data			
			Common			Local
			Non-Vol	Vol	I/O	
Replace 57C413A with Same or new 57C413A At power up	NO	STOP	Data OK	Zero	Lost (Random)	Data OK
Replace 57C413A with New 57C413B or 57C423 At power up	NO	STOP	Data OK	Zero	Zero*	Data OK
Replace 57C413A with used 57C413B or 57C423 (module may contain data from another application) At power up	NO	STOP	Data OK	Zero	Value from programmed module at same address*	Data OK
Replace 57C413B or 57C423 with New 57C413B At power up	NO	STOP	Data OK	Zero	Zero*	Data OK
Replace 57C413B or 57C423 with used 57C413B or 57C423 (module may contain data from another application) At power up	NO	STOP	Data OK	Zero	Value from used module at same address*	Data OK
Replace battery on 57C413B or 57C423 and return same module to use. At power up**	NO	STOP	Data OK	Zero	Data OK	Data OK

* To ensure your application task does not access old or incorrect data, create a BASIC task to write zeroes to each register location before you run any other application tasks.

** Assumes super-capacitor has maintained memory on the module during the time required to replace the battery.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

[illegible]

[illegible]

[illegible]

Publication [illegible]

Copyright © [illegible] Rockwell Automation, Inc. All rights reserved. Printed in U.S.A.