

2.0 INTRODUCTION TO THE AutoMax DISTRIBUTED CONTROL SYSTEM

AutoMax is a programmable, microprocessor-based control system capable of performing real-time control with millisecond response time. Because AutoMax is modular, you can custom-configure the system to meet the specific requirements of your application. In addition, the Processor module operating system is included in the AutoMax Executive software and can be loaded from the same device used for writing application programs.

The available hardware modules include the following: computing, communications, digital input, digital output, analog input, analog output, motion control, and motor and drive control. In addition, the system can be configured to include many other commercially available Multibus-compatible modules.

Typically, an AutoMax system consists of a number of chassis, or racks, containing various modules. Up to 43 racks, each containing at least one Processor module, can be connected together as part of a control network using Network Communication modules. For drive control applications, each AutoMax rack can control up to 20 drives (10 UDC modules maximum) by using Universal Drive Controller (UDC) modules in a rack that also contains at least one AutoMax Processor module. Using Remote Communication modules, any rack containing at least one Processor module can control up to 7 remote I/O subsystems, which do not require Processor modules.

The system is designed to support the sharing of data among Processors in a single rack (up to four), as well as among Processors on different racks in a network. Applications can be distributed among multiple Processors and racks. Common data, such as I/O values, is accessible to all Processors simply by referencing the appropriate variable name in a task.

Application programs are created in an MS-DOS™ or MS-Windows environment using an 80486-compatible personal computer. AutoMax supports three different programming languages: Ladder Language, Control Block Language, and Enhanced BASIC Language. Each of these languages is suited to a different type of task commonly found in the industrial and process control environment. AutoMax Control Block, BASIC, and PC tasks can be executed on an AutoMax Processor. Only UDC Control Block tasks can be executed on a Universal Drive Controller (UDC) module.

Ladder Language, also known as PC (programmable control) Language, is used for sequential logic operations. Ladder programming is very much like programming for a conventional programmable controller, incorporating standard ladder diagrams to establish the sequence and types of operations performed. AutoMax Ladder language is described in instruction manual J2-3093 and J2-3094.

Control Block language is used for programming control loops. The Control Block language consists of BASIC statements that contain special function calls, including amplifiers, integrators, function generators, PID controllers, etc. Control Block language is described in J-3676.

The Enhanced BASIC language is used for general programming, keyboard and CRT-based operator interfaces, and numeric processing. Enhanced BASIC language is described in J-3675.

Using the AutoMax Executive software, application programs, or tasks, created on the personal computer using Control Block or BASIC are then compiled and downloaded to one or more AutoMax Processor modules or UDC modules in a rack. Application tasks created using Ladder language do not need to be compiled before being downloaded. All application tasks can be run, monitored, modified, and stopped from the personal computer using the AutoMax Executive software.

See J-3675, J-3676, J2-3093, and J2-3094 for more detailed information about programming for AutoMax systems.

2.1 ReSource AutoMax Programming Executive Overview

The remainder of this chapter describes the AutoMax Programming Executive software, which will be referred to as the Executive software from here on. See Appendix N for the **features** that are new in this release of the Executive software.

AutoMax Executive V4.x uses Microsoft Windows 95 (or later) to provide a graphic environment for all the software functions necessary to configure the hardware in your application and to create, organize, document, and troubleshoot application tasks on a personal computer. No special commands or syntax are used.

AutoMax Executive V4.x offline functions are contained within a set of five Windows applications: **System Configurator**, **Rack Configurator**, **Variable Configurator**, **Task Manager**, and **Ladder Editor**.

The **System Configurator** is used to organize the application, or system, into sections. Each section is a functional group of racks. For example, all of the racks for a physical section of a machine could be grouped into a section. See the heading "Application Organization" in this section for more information.

The **Rack Configurator** is used to configure the modules in a rack; the racks, heads, and rails connected to a remote I/O network; and the two-point modules in digital rails. The Rack Configurator uses a diagram of the AutoMax rack for configuring modules; a diagram of the remote I/O network for configuring remote racks, heads, and rails; and a diagram of the digital rail for configuring two-point modules. The Rack Configurator can be used to add the UDC module, specify the PMI and rail hardware, and enter drive parameters.

The **Variable Configurator** is used to map variables to I/O or memory points. It uses forms specific to the card on which the I/O or memory points reside. Mapping variables to physical locations on individual modules and to common memory locations allows the programmer to create application tasks referencing variable names instead of actual physical locations. This application essentially replaces the text configuration task used in earlier versions of the executive.

The **Task Manager** is used to develop application tasks. In the offline mode, the Task Manager is used to add, edit, compile, print, and verify tasks. In the online mode, it is used to load, save, run,

stop, and delete tasks on the AutoMax Processor or UDC module. Task error logs are viewed and cleared in the Task Manager.

The AutoMax **Ladder Editor** is a Windows-based editor used to develop, edit, or view ladder (.PC) programs. The Ladder Editor allows you to have multiple editing/viewing windows into the graphical language program, and you can edit multiple programs in the same session. The AutoMax Ladder Editor is also used to monitor Ladder programs online. Note that older versions of AutoMax Ladder programs (DOS-based ladder programs) can be converted into the new Windows-based format. Instruction manual J2-3093 describes how to use the Ladder Editor.

NOTE: *You cannot monitor a Ladder program and use the Online Task Manager at the same time because they both use the same communication channel to rack.*

A menu map for AutoMax Executive V4.x is shown in figure 2.1.

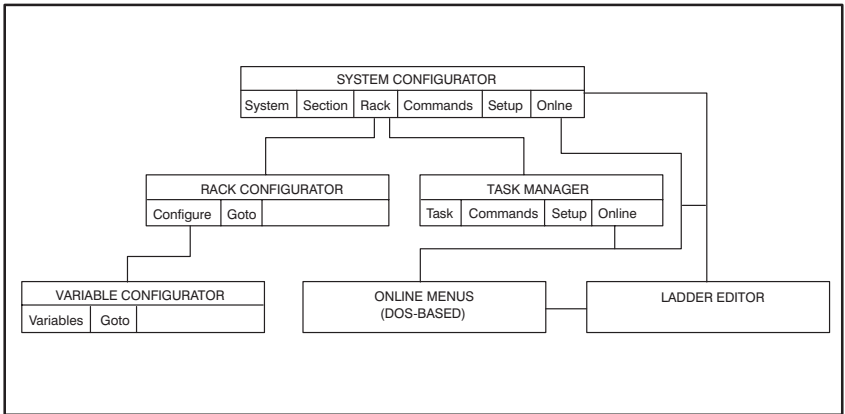


Figure 2.1 - AutoMax Menu Map

Application Organization

It is important to become familiar with the options available in each Windows application and with the hierarchy of information about your specific software application. You must first group all the software for a given application in a system. You then divide the system into sections. Sections are simply collections of racks (any number) and help organize the system in any manner appropriate to the application, e.g., related function or location.

The most basic level of application organization is the rack. The rack consists of the application tasks for the rack and all the information about the rack, including the modules and variables defined in it. Racks correspond directly to the AutoMax racks which are installed for the application.

To help you document the way in which your application is organized, documentation files can be created for a system, section, rack, or task. These text files can be used to explain design specifications, notes about troubleshooting, or any other information.

For users with numerous specific applications, the AutoMax Executive also allows you to group systems into libraries. Libraries are used to store similar systems, e.g., plating systems. The default library is called AMXLIB and will be sufficient for most users.

The library/system/section/rack hierarchy allows you to structure each application so that it can be easily understood and its parts accessed quickly. It also allows you to carry out operations, e.g., copying, on an entire system, section, or rack.

2.2 Introduction to Configuring Racks and Variables

Once you create the system/section/rack structure for your application, you can begin to configure the hardware in your installation. Hardware configuration consists of adding modules to each rack (including network and remote racks, UDC modules, and PMI rail hardware) to reflect the actual installation. The AutoMax Executive checks that modules are added properly. For example, a second AutoMax Processor module can be added only after a Common Memory Module (M/N 57C413 or 57C423) has been added. If your application includes modules that are not supported in V4.x, you can use “generic” modules. There are three generic modules: one with 32,768 registers (GEN32K), one with 8000H registers in hex (GEN32KH), and one with 32 registers (GEN32) for use in a remote I/O rack (or in a main rack when 32 registers are sufficient). Note that GEN32K and GEN32KH modules can be used only in a rack that contains an AutoMax Processor module.

After you have added a module, you can map variables to I/O points or common memory on that module using “forms” tailored as required for the specific register organization on each module. This variable configuration “map,” along with information about the application tasks for the rack, is stored in databases, and must be compiled and downloaded to the rack along with application tasks before application tasks can run in the rack. This configuration information is sometimes called the configuration task.

After you have completed the physical configuration of your system, you can generate a bill of material which lists all of the racks, heads, rails, and modules used in the system along with the needed batteries, cables, and optional hardware and software. See Appendix L for additional information.

2.3 Introduction to Creating Application Tasks

After you have configured I/O points and common memory using variable names, you can create application tasks (programs) using these variable names instead of locations. You can create BASIC tasks, PC/Ladder Logic tasks, and Control Block tasks. BASIC and Control Block tasks are stored in text files which are edited using a text editor; PC/Ladder Logic tasks are stored in binary files and edited using a custom editor. A Ladder Editor is included with the AutoMax Programming Executive software. In addition, you can create “include” files which are added to a BASIC or Block task when it is compiled (by using the BASIC language INCLUDE statement).

Once the variables have been configured and the application tasks created, you can compile the tasks to create object code that will run on the AutoMax Processor or UDC module(s) in the rack. PC/Ladder Logic tasks are stored as object code and do not need to be compiled.

Drive Control

Tasks that control Distributed Power Drives (DPS) are stored and executed on UDC (Universal Drive Controller) modules in the rack. UDC tasks are written in control block language. They are referred to as "UDC tasks." See the appropriate DPS programming and configuration instruction manual for more information about UDC tasks.

Multi-Tasking

AutoMax Processors are capable of multi-tasking. Each AutoMax Processor can execute multiple application tasks based on a priority basis, sharing common data between tasks. Application task execution can also be coordinated using BASIC language statements and functions in both BASIC and Control Block tasks.

Each UDC module can execute two independent Control Block tasks to control two separate drives of any type, usually referred to as Drive A and Drive B. UDC tasks A and B execute sequentially (task A executes, then B) based on a user-defined interval of 500 μ sec CCLK signal. If only one task is required, i.e., the UDC module will control only one drive, the drive can be connected through either channel A or B.

The UDC tasks on one module share the data in the dual port memory. UDC tasks cannot, however, share data with tasks on other UDC modules, or with tasks on AutoMax Processors, unless an AutoMax task reads from or writes to the UDC's dual port memory. Only AutoMax tasks can move data between two or more UDC tasks on different UDCs, and between UDC tasks and AutoMax tasks.

The following table illustrates the differences between UDC modules and AutoMax Processors.

UDC Module and AutoMax Processor Comparison

	Maximum # in a Rack	Maximum # of Tasks**	Programming Languages
AutoMax Processor	4	32	BASIC, Control Block* or PC/Ladder Logic
UDC Module	10	2 (1 each for drives A/B)	Control Block*

* A subset of BASIC statements and functions is permitted. The two types of Control Block tasks (UDC and AutoMax) support slightly different subsets. These are listed in the Control Block instruction manual (J-3676) and in Appendix C of this manual.)

** Each rack can have a maximum of 32 tasks loaded into it. This total does not include BASIC "include" tasks.

2.4 Introduction to Online Operations

Before you can go online to any rack in the system, the operating system, or runbase, for the AutoMax Processor and UDC module(s) must be loaded to the rack. In general, you will only be concerned with the file types listed below for purposes of loading to the rack.

1. Operating system (OS) file
Each Processor module and UDC module must have an operating system file. The operating system for each Processor module in a rack is usually the same. The Processor operating system is included in the Executive software. UDC OS files are included in the separately-purchased DPS software options. Note that UDC operating systems contain within them operating systems for the Power Module Interface (PMI) hardware they are connected to. See section 5.4.3 for more information.
2. Configuration file
One configuration file is required for each rack. This file is stored on the Processor module. If there is more than one Processor module in the rack, the file is stored on the Common Memory module. Configuration files are identified by the extension “.CNF” in the file name. After they have been prepared for loading to the rack, the configuration files will have extension .OBJ. See section 5.4.3 for more information.
3. Application task files
Application tasks are stored and executed on Processor modules and Universal Drive Controller modules in the rack. Each Processor module may store/execute more than one task. Each Universal Drive Controller module may store/execute one or two tasks. Application tasks are identified by the one of the following extensions in the file name: .BLK , .PC, or .BAS. After the files have been prepared for loading to the rack, however, they will have extensions .OBJ, .PC, and .OBJ, respectively. See chapter 8 for more information.
4. Parameter object file
Each Universal Drive Controller module must have one parameter object file. Parameter object files are identified by the extension “.POB” in the file name. See chapter 8 for more information.

2.5 AutoMax Help Screens

The AutoMax offline “help” feature uses the same menu structure and options as Windows Help. For a description of the Windows Help feature, refer to the *Microsoft Windows User's Guide*.

Help is available for each of the four applications in AutoMax Executive V4.x (System Configurator, Rack Configurator, Variable Configurator, and Task Manager). Help is also available for most dialog boxes. In addition, pop-up help is available for key terms. Simply press F1 (or click on Help) from any screen or dialog box listing that option. F1 is reserved exclusively for accessing the Help feature. Note that help for online commands, also accessed using F1, consists of text files and does not use the Windows Help menu structure.