analog dialogue

A forum for the exchange of circuits and systems for real-world signal processing

PROGRAMMABLE μMAC 5000

Volume 17, Number 3, 1983

ANALOG DEVICES

Editor's Notes

ON CHANGE

You may have noticed—on this issue's cover—a small change in the definition of the forum contemplated by this Journal. If you have been an avid and faithful reader of our covers, you have in fact seen it evolve over the years. For example, the Premier Issue's "A Journal for the Exchange of Operational Amplifier Technology," became Volume 3's "A Journal for the Exchange of Analog Technology."

In Volume 5, the focus shifted and expanded to "A forum for the exchange of circuit technology: Analog and Digital, Monolithic and Discrete," which graced our first pictorial cover (an artist's rendering of the TO-99 version of the AD530, the world's first complete monolithic multiplier/divider). Our subtitle remained that way for 8 years, while *Analog Dialogue* grew from 16 pages to 20 pages per issue and began to sport 4-color photographic covers.

Then, in 1979, Volume 13, No. 1—the first issue dedicated to a single product—introduced MACSYM II, and with it, "A forum for the exchange of circuits and systems for measurement and control." With the coming of the LTS-2000 computerized test system, in Volume 14, Number 3, the subtitle became "A forum for the exchange of circuits and systems for measurement, control, and test." Digital multipliers and multiplier/accumulators soon broadened "measurement, control, and test" to the more-inclusive "real-world signal pocessing." And, with the introduction of the µMACBASIC-programmable µMAC-5000 in an issue replete with both software and hardware, it now seems appropriate to expand our horizons to "A forum for the exchange of circuits, systems, *and software* for real-world signal processing."

Incidentally, we are serious about the "Dialogue" and "Forum" aspects of this publication's title. But we can't by ourselves legitimately create both sides of a dialogue or the other voices in a forum. We need your inputs.

What kinds of inputs? We'd like to consider the useful things readers accomplish with our technologies, the technical problems our products have helped to solve, and useful ideas for design (implemented and found workable) using our products. We don't need finished technical papers—competent editorial help is available to get good technical material into interesting, readable print.

SESQUIDECENNIAL

This issue marks the completion of our first 15 years as Editor of this publication. Though we're still struggling to fulfill the promise of quarterly publication, made in our first issue (its third year), we've been delighted with the increase in size, growing diversity of contents, and improvement in overall quality. Moreover, we've been happy to be a participant in the phenomenal technological growth of Analog Devices that has been evidenced in these pages, and manifested in the continual need to adjust the framework of the *Dialogue*. We thank you, our readers, for your many favorable comments, and we continue to invite your participation. We're looking forward to even more enjoyment in times to come. ▶

Dan Sheingold

THE AUTHORS

Donald W. Brockman (page 22) is Technical Communications Manager at ADI's Computer Labs Division (formerly Computer Labs Incorporated), which he joined in 1968 from Bell Telephone Laboratories. His formal education includes a B.A. degree, and he has worked in engineering, writing, recruiting, and sales functions before his present job in Marketing. His photo appeared recently in *Analog Dialogue* 16-2.

Don R. James (page 16), a Senior Associate at Systems Application Engineering, Inc., Houston, Texas, received a B.A. in Mathematics from Mississippi State University in 1966. An expert in the areas of factory automation and plant management systems, sensor-based monitoring/control systems, and computer-based process-control systems, he has designed mill-wide plant-management data bases, hierarchical communication networks, quality assurance and customer acceptance test systems, and digital fire-control systems for ship-based guided missiles. He is a Senior Member of I.S.A.

Richard Quinn (page 20) is a Senior Marketing Engineer with ADI's Measurement and Control Division. Prior to that, he worked for ADI as corporate Marketing Analyst. Rich has a BSEE from Case Western Reserve University and an MBA from the Amos Tuck School of Business Administration. Earlier, he worked at Gould as a Marketing Analyst and at Reliance Electric as a Control-Systems Engineer.

Dave Reynolds (page 3), Senior Marketing Engineer for Measurement and Control Systems Interface Products at ADI since July, 1981, earned a BSEE at the University of Southern California, and both an MSE and an MBA, from the Moore and Wharton Schools at the University of Pennsylvania. He joined ADI as a Strategy and Systems Analyst, became an Operations Systems Analyst in ADI's Systems Components Division, then joined Marketing. He enjoys team sports and racquetball.

(continued on page 26)

analog dialogue

2 Technology Way, P.O. Box 280, Norwood, Mass. 02062

Published by Analog Devices, Inc., and available at no charge to engineers and scientists who use or think about I.C. or discrete analog, conversion, signal processing and test circuits and systems. Correspondence is welcome and should be addressed to Editor, *Analog Dialogue*, P.O. Box 280, Norwood, Massachusetts, U.S.A. 02062. Analog Devices, Inc., has representatives and sales offices throughout the world. For information regarding our products and their applications, you are invited to use the enclosed Business Reply card, write to the above address, or phone 617-329-4700, TWX 710-394-6577 or 710-336-0562, Telex 924 491, or cable ANALOG NORWOODMASS.

STAND-ALONE SINGLE-BOARD MEASUREMENT AND CONTROL SYSTEM

Provides Signal Conditioning, Conversion, Computation, Serial Communication
μMAC-5000 Is Programmable in a Powerful Extended Basic Language: μMACBASIC

by Dave Reynolds

The μMAC-5000* is a new single-board programmable measurement-and-control system. Combining direct connection to sensors—via screw terminals—with complete signal conditioning, digital inputs and outputs, a 16-bit microcomputer, an extended BASIC language for measurement and control, serial communication facilities, a power supply with uninterruptible features, and ruggedized construction, it provides an instrumentality for measurement and control which is applicable in a broad range of stand-alone or distributed control systems (Figure 1). The number and kinds of analog and digital inputs and outputs is expandable via the family of μMAC expansion I/O boards, introduced in *Analog Dialogue* 15-1 (1981) and subsequent issues.

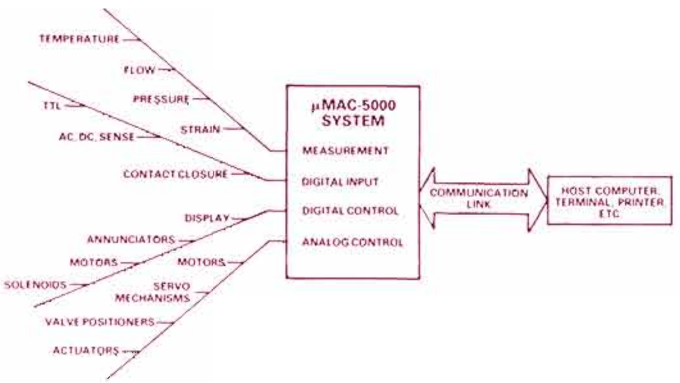
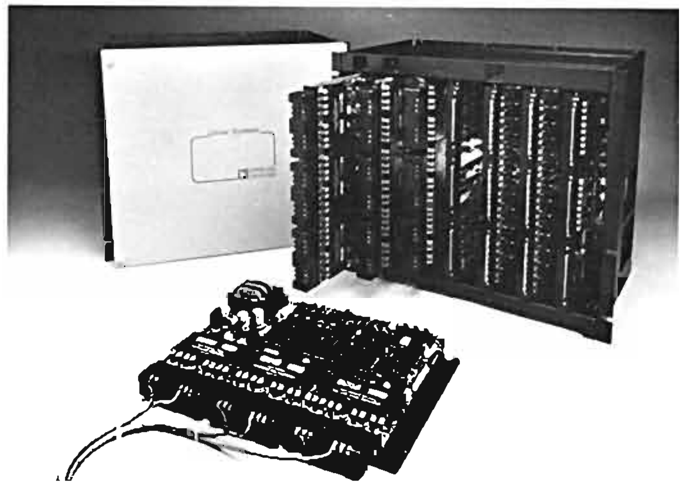


Figure 1. μMAC measurement-and-control concept.

The μMAC-5000 is designed for applications requiring distributed intelligence or local front ends in industrial process control, machine control, energy management, product test, supervisory control and data acquisition (SCADA), and pilot plants. It is also well-suited as an intelligent front end for any personal computer in R & D and laboratory applications, for measurements employing thermocouples, strain gages, RTDs, current transmitters, and low- or high-level voltages and currents, and for control involving binary switching or sensing.

Stored in read-only memory (ROM) is the μMACBASIC operating system, which allows novice or experienced users to create and store powerful measurement and control programs. On-board plug-interchangeable signal conditioning modules combine ± 1000V isolation, sensor excitation, and amplification for direct screw-terminal connection of a variety of sensors. Digital I/O channels provide direct interface to contact closures, TTL levels or high level ac and dc voltages (with the μMAC-4020). Because the μMAC-5000 is compatible with Analog Devices μMAC-4000 series of expander boards and signal-conditioning modules, users can connect additional analog and digital I/O boards to suit the specific number and kinds of input/output channels to fit the needs of the application.

*Use the reply card for technical data.



The μMAC-5000 consists of a single master board with 4, 8, or 12 analog inputs, 8 digital outputs, and 8 digital inputs, of which 2 may be used as pulse inputs. The 5-MHz, 16-bit 8088 CPU processes the data collected from the input channels, sends out control signals, and communicates with other equipment as directed by the program installed in memory.

The basic computer has 32K bytes of read-write memory (RAM)—expandable to 64K—of which 12K is required by the computer, and 80K bytes of read-only memory (ROM)—expandable to 96K. Since the system makes available at least 16K bytes

IN THIS ISSUE

Volume 17, Number 3, 1983 – 28 Pages

of both RAM and ROM, the user has the option to download programs into read-write memory from a host computer or store them in non-volatile EPROM. Two additional ROMs, used for compiling during program development, can be removed from run-time systems; consequently, 3 user ROMs can be added if desired. A battery backup supply protects the volatile read-write memory against loss of information when ac line power is removed. Figure 2 depicts the basic elements of the μMAC-5000 system.

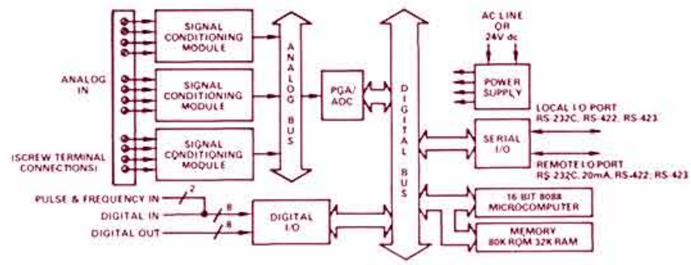


Figure 2. System block diagram.

DIGITAL COMMUNICATIONS

The μMAC-5000 communicates with terminals or host computers over a serial link in a party-line or radial configuration. Its full- or half-duplex universal synchronous/asynchronous receiver/transmitter (USART) receives and transmits data at selectable baud rates and performs parity and cyclic redundancy checking. Other security procedures, such as checksum, can be implemented in μMACBASIC. Conversion routines written in μMACBASIC can also translate the 7-bit ASCII code to additional character codes.

The μMAC-5000 has two serial ports for local or remote communications. The local port communicates via RS-232C, RS-422 or RS-423; its primary use is to support program development, a local terminal, and/or a printer. The remote port supports RS-232C, 20mA, RS-422 and RS-423. In remote applications, using 20-mA current loop or RS-422 connections, the system can be located up to 10,000 feet from the host computer. Control lines (RTS, CD, DTR and CTS) also permit connection to a modem for longer distance communications. This port communicates at from 110 baud to 19.2k baud asynchronously (Figure 3).

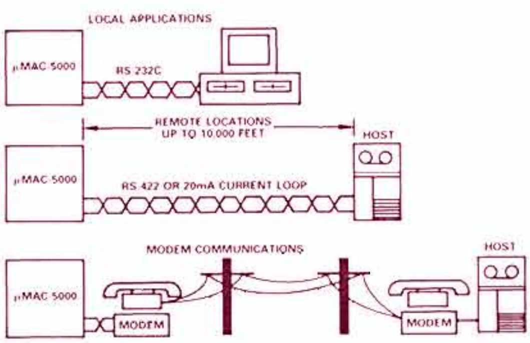


Figure 3. Modes of communication over various distances.

SIGNAL CONDITIONING

The master board on the μMAC-5000 contains all the necessary hardware to permit direct connection of sensors in harsh industrial environments. The μMAC-5000 alone can be used as a single-board system solution for applications having 28 I/O points or less. The system can also address expander boards for connection to hundreds of analog and digital I/O points, in clusters or multidrop configurations, for applications requiring additional I/O channels.

On-board signal conditioning is available for 4, 8, or 12 analog input channels, 8 digital output channels and 8 digital input channels, two of which can be used as pulse or frequency inputs. The digital channels may represent individual events or binary choices—or be combined to form digital words. An integrating converter provides up to 14 bits of resolution (13 bits plus sign). The higher conversion speeds can be obtained with a minimum resolution of 11 bits. (Figure 4's photo shows the system architecture in visual detail and Figure 5 shows the same in block form).

A choice among four different plug-in 4-channel signal conditioning modules allows the user to mix-and-match various types of sensors in groups of up to 4 of each kind. These modules have input protection, can provide isolation, filtering, and excitation, and handle the most common forms of sensors, including thermocouples, RTDs, strain gages, current transmitters and semiconductor current-output thermometers (AD590). Table 1 outlines the input types and signal ranges each QMX module handles.

Input Type/Span	Nonisolated Modules		Isolated Modules	
	High-Level Low-Level QMX01	Low Level with Excitation QMX02	Low Level QMX03	High Level QMX04
dc, ±25mV, ±50mV, ±100mV	√		√	
dc, ±1V, ±5V	√			√
dc, ±10V	√		*	
dc, 0-1mA Transmitter	√		√	
dc, 0-20mA Transmitter	√		√	
dc, 4-20mA Transmitter	√		√	
Thermocouple, Types J,K,S,T,R,E,B,W	√		√	
100Ω Platinum RTD, α = 0.00385		√		
100Ω Platinum RTD, α = 0.00392		√		
Strain Gage, ±30mV		√		
Strain Gage, ±100mV		√		
AD590, AC2626		√		

*±10V isolated input using AC1814 attenuating screw terminal connector.

Table 1. Signal-Conditioning Modules.

The QMX03 and QMX04 (Figure 6) employ electromagnetic isolation techniques to provide ± 1000V peak continuous isolation for protection and 160dB of common-mode rejection. For most industrial applications, isolation is recommended to eliminate ground loops and high common-mode-voltage problems, and to protect personnel and equipment from hazardous electrical faults.

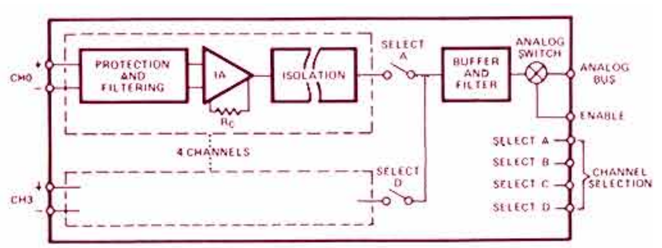


Figure 6. QMX03/QMX04 modules with 1000-volt isolation.

The QMX01 handles four channels of high-and low-level differential analog inputs, including thermocouples of types J,K,T,S,R,E,B, and W. The module provides 130-V normal-mode protection to prevent damage if line voltage is accidentally connected across the input terminals. Common-mode voltages up to ± 8.5V do not degrade measurement accuracy.

Overview of the μMAC 5000

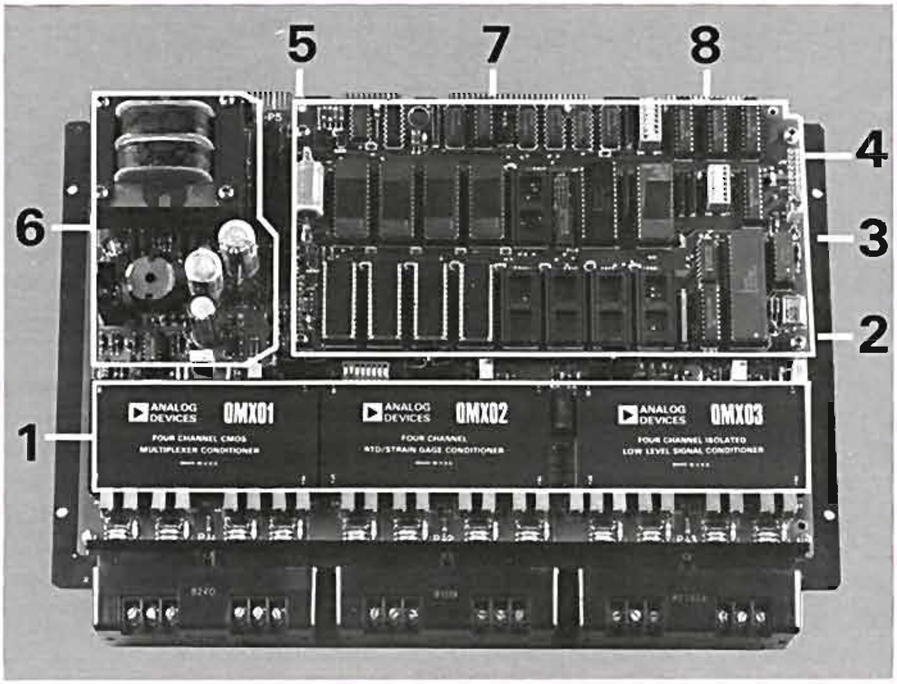


Figure 4. Photograph of the μMAC-5000, showing the system architecture.

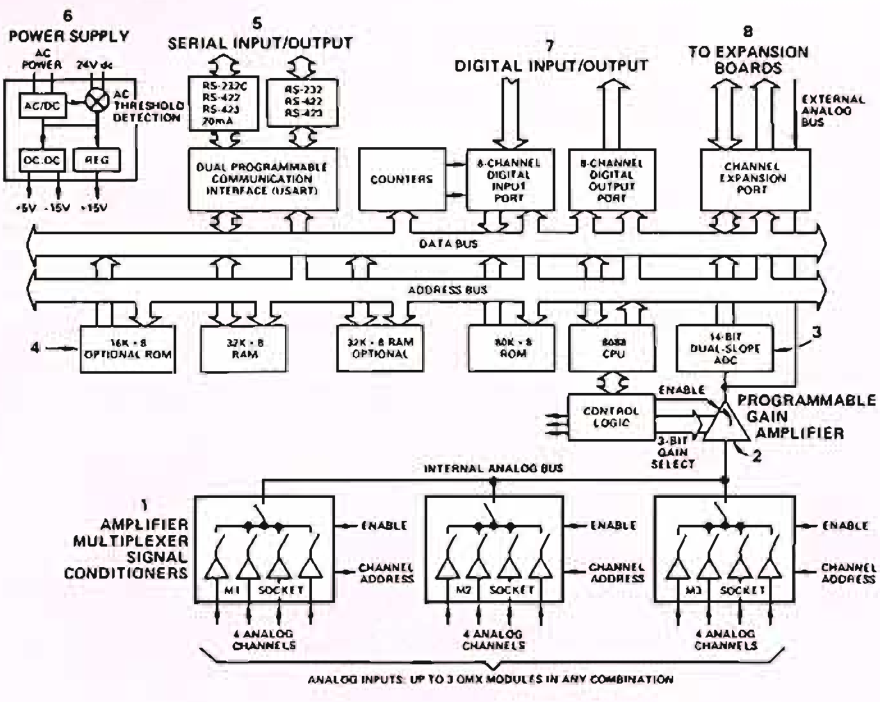


Figure 5. System functional block diagram.

LEGEND

1) **Multiplexed Signal-Conditioning Modules.** Three plug-in 4-channel signal conditioning modules provide multiplexing, preamplification, and optional transformer isolation and bridge excitation. Four mix-matchable module types are available to handle many input sources: thermocouples, RTDs, strain gages, 4-20mA current loops, and millivolt, volt, and milliampere signals.

2) **Programmable-Gain Amplifier (PGA).** Digital logic sets the gain of the PGA to amplify the preamplified input signal to the full-scale range of the a/d converter (hidden).

3) **A/D Converter.** An integrating converter provides resolutions from 14 bits (13 bits + sign) to 11 bits, depending on the desired number of conversions per second (hidden).

4) **Intelligence.** A 16-bit μC with 80K bytes of ROM and 32K bytes of battery backed-up RAM supports stand-alone applications. Plug-in sockets support an optional 16K Bytes of ROM and 32K bytes of RAM. The μMAC BASIC compiler and operating system use 80K bytes of ROM and 12K bytes of RAM. A DIP-switch option permits running programs from either ROM or RAM.

5) **Communications.** Two serial ports connect the μMAC-5000 to any host computer. Local and remote ports communicate in RS-232C, RS-422, or RS-423. The remote port also supports 20-mA current loops.

6) **Power Supply.** An ac/dc converter and dc/dc converter generate +5V and ±15V system voltages from an ac power line or +24V dc source. Using both, the system can detect brownouts or power outages and switch to dc without disturbing program execution. Factory-set line-voltage options are: 100V, 115V, 220V, or 240V.

7) **Digital I/O.** 8 digital inputs, optically isolated for 300V peak, can sense contact closures or accept TTL signals. The 8 latched-TTL outputs can sink up to 24mA. The μMAC-4020, connected to the digital I/O port, controls and monitors digital signals at line voltages. Two counters provide for pulse accumulator and frequency inputs.

8) **Expansion Port.** The expansion port provides for up to 6 mix/match expansion boards to greatly increase the number of available analog and digital I/O channels.

The QMX02 (Figure 7) provides protection, filtering, gain, and multiplexing for RTDs, strain gages and the AD590 semiconductor temperature sensor (AC2626 probe version), plus excitation for RTDs. The µP enables constant-current excitation for 3-wire RTDs (if required), selects gain and channel, and enables an analog bus for conversion of a signal from the appropriate module.

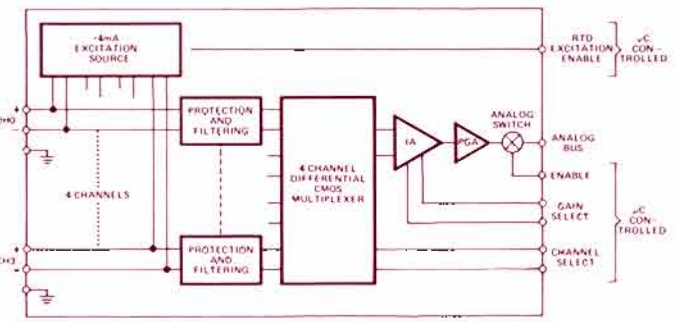


Figure 7. QMX02 low-level 4-channel input module with excitation.

Thermocouples can be connected directly to the µMAC-5000, using the appropriate modules (QMX01—nonisolated, QMX03—isolated) and the AC1800 isothermal connector. The AC1800 provides cold-junction compensation by sensing the variations in ambient temperature in the vicinity of the screw terminals. This temperature reading is measured periodically through a software command, converted by the a/d converter and stored in memory. The CPU automatically uses it to provide software correction of thermocouple measurements. The thermocouple is also linearized (digitally), using a multi-segment piecewise-linear approximation employing lookup tables stored in ROM. The temperature reading is converted into engineering units of °C.

The µMAC-5000 has 8 digital inputs and 8 digital outputs. The inputs have 300-V optical isolation and can sense either logic levels, ac or dc voltages, or contact-closure states. The outputs can sink up to 24 mA, and can be beefed up by the µMAC-4020 expander board to handle high-power ac and dc circuits. Digital input channels 6 and 7 can optionally be used for pulse accumulation or frequency inputs (from flowmeters or V/F converters). Both pulse inputs have a 32-bit counter and operate over a 0 to 20 kHz frequency range.

EXPANSION

The unit of expansion is the *cluster*. A multiple-board configuration (cluster) consists of one µMAC-5000 programmable Master Board and up to six Expander boards. Because it contains the communications capability and intelligence, a µMAC-5000 must reside in each cluster. The 9 ½" × 13" boards can be mounted in an available card cage or installed in sealed industrial cabinets (NEMA enclosures) for operation in corrosive or environmentally damaging locations. Table 2 outlines the functions available on expander boards.

A *network*, using up to 16 clusters in a multidrop—or party-line—configuration (Figure 8), allows the host computer to monitor and control up to 1,344 analog and 4,864 digital points via a single 20-mA or RS-422 communications line. For applications requiring more I/O points, e.g., refineries, a radial-line configuration provides further I/O expansion—limited only by the availability of computer ports (Figure 9).

Analog outputs, generated by d/a converters, are available on the

Model	Function	Description
µMAC-4010	•Analog input •Digital I/O	•Up to 12 channels using QMX modules •16 channels digital I/O
µMAC-4015	•Isolated analog input	•Low cost, 12 channels of same input type •1500V isolation
µMAC-4030	•8 channels analog output	•Voltage or isolated 4-20mA outputs •Readback, bumpless transfer
µMAC-4040	•Digital I/O	•32 channels isolated digital inputs •32 channels digital outputs (TTL)
µMAC-4050	•Multi-function digital I/O •8 channels mixed function	•Pulse accumulator input •Frequency input •Pulse output •Time proportional output
µMAC-4020	•Interface subsystem to high level ac and dc voltages and currents	•Solid state relay subsystem 16 channels, mix of input and output functions

Table 2. Expander Boards.

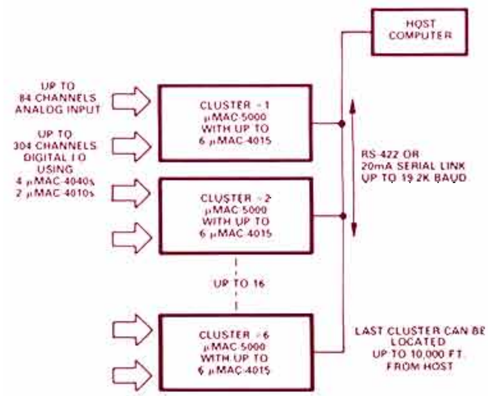


Figure 8. Expansion clusters in multidrop configuration.

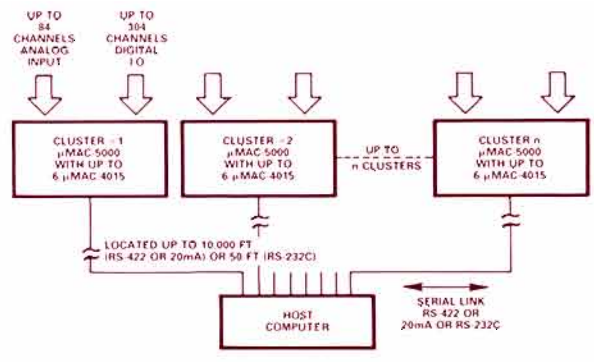


Figure 9. Expansion clusters in star configuration.

µMAC-4030 Analog Output expander. It features 12-bit resolution, voltage and current (4-20 mA/0-20 mA) outputs with ± 1000V isolation, and is capable of either automatic or manual operation with bumpless transfer and programmable slew rate.

Thirty-two digital input and thirty-two output channels are available on the µMAC-4040 with ± 300V isolation. Analog input expansion is accomplished with the µMAC-4010, or a lower-cost µMAC-4015 (see page 13). The µMAC-4010 provides twelve analog input channels, using the QMX modules, and 16 digital I/O channels. Figure 10 shows a four-board configuration with additional analog inputs, digital I/O, and analog outputs. The boards can be combined in 4 of the 7 slots of a µMAC-5000 card cage.

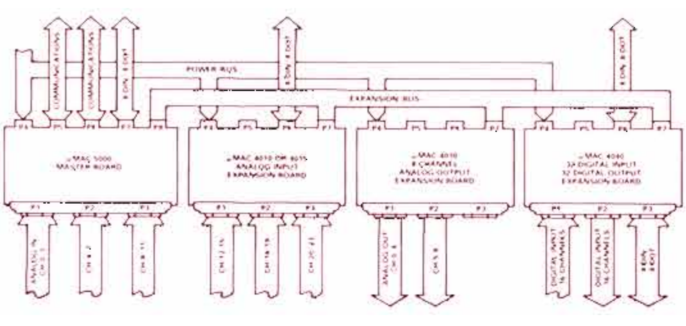


Figure 10. Expansion cluster with µMAC-5000 and 3 expander boards.

Each µMAC-5000 and each expander board can be connected to an ac line, +24V dc power, or both. When connected, +24V dc power provides a backup to an ac line; on-board circuitry detects intermittent ac power losses and switches to +24V dc without disturbing the operation of the µMAC-5000. A flag is set to inform the host computer that the battery backup mode has been activated. Primary ac voltage is factory-configured: 100V, 115V, 220V or 240V; input power is transformed and regulated to the three dc voltages necessary to power the analog and digital circuitry.

SOFTWARE

The µMAC-5000 uses a real-time measurement-and-control programming language called µMACBASIC, an extension of the standard BASIC language. The language is syntax-compatible with the industry standard MICROSOFT™ BASIC, used in many popular personal computers. Programs written in MICROSOFT can be converted to µMACBASIC with relative ease.

µMACBASIC incorporates the standard programming and arithmetic commands found in BASIC and adds to that a library of specialized real world measurement and control commands, such as AIN for analog input. This structure permits the user to collect, manipulate, and output real world data by constructing a program from a series of high-level BASIC commands. By addressing analog and digital I/O channels, the data is inputted or outputted with one simple command. A list of µMACBASIC Measurement & Control commands can be found in Table 3.

The µMAC-5000 software, for example, on the command AIN(channel), automatically finds the voltage on an analog input channel and converts it to digital; if identified as originating at a thermocouple, the data is linearized, compensated, and changed into engineering units. Programmers are relieved of writing these steps into their program. The use of an Analog input command AIN illustrates this:

```
10 TEMP1 = AIN(1)
20 PRINT "THE TEMPERATURE IS ";TEMP1;" DEGREES C"
RUN
```

The response might be:

```
THE TEMPERATURE IS 23.2 DEGREES C
```

The first command measures the analog input on channel 1 (e.g., a thermocouple), computes the correct temperature in degrees Celsius, and assigns the value to the variable TEMP1. The second command formats and prints the data on a CRT or printer.

ANALOG INPUT/OUTPUT COMMANDS	
AIN	Returns the analog input of specified channel
COLLECT	Returns multiple analog inputs of specified channel into an array
TEMP	Returns the cold junction temperature associated with specified channel
AOT	Outputs a value to specified analog output channel
RDAC	Returns the analog output value of the specified channel
DIGITAL I/O COMMANDS	
DIN	Returns the state of a digital input port
DOT	Outputs a value to specified digital output port
FREQUENCY I/O AND EVENT COUNTING	
PULSE	Generates a single pulse output from a specified µMAC-4050 channel
TPRO	Generates a time proportional output from a specified µMAC-4050 channel
EVREAD	Returns value of specified Event Counter
EVLOAD	Preloads Event Counter with a value
EVSTART	Arms a specified channel for event counting
EVSTOP	Disarms a previously designated event counting channel
FREQ	Reads the frequency of the specified channel
FREQGATE	Starts the gate for frequency counting
FREQSTAT	Returns status of gate
SUPERVISORY COMMANDS	
DSTATUS	Returns µMAC-4030 dip switch status
INTYPE	Sets up analog input channels
RTYPE	Returns analog input configuration
ZDAC	Resets µMAC-4030 to the power up condition
CALIBRATION COMMANDS	
CALON/CALOFF	Engage/Disable automatic calibration

Table 3. µMACBASIC Measurement & Control Commands.

An analog output to control a heater proportionally (in very simple-minded fashion), based on analog input data, could be expressed quite compactly:

```
120 AOT(2) = (200-AIN(1))/200
```

The AIN(1) input is the measured process temperature in degrees C, and the 200 represents the desired temperature. The denominator of 200 scales the temperature difference to a percent of the 200°C span. The statement puts out on channel 2 the largest signal (1.0) when the temperature is zero, the smallest signal when the temperature reaches the 200°C level (0), and negative signals above 200°C.

The digital input command DIN reads the logic levels on the digital input channels, either a "1" or "0".

```
10 SWITCH1 = DIN (0,0,7)
20 PRINT "THE DIGITAL INPUT ON PORT 0 CHANNELS 0-7
   IS ";SWITCH1
RUN
```

The response might be:

```
THE DIGITAL INPUT ON PORT 1 IS 255
```

The program assigns the label, 'SWITCH1' to the radix-10 value of the digital logic levels at input port 0 and bits 0 to 7 (0,0,7). For example, if this value was 255, which equals $2^8 - 1$, it would indicate that all eight digital inputs are on.

The digital output commands provide ON/OFF control. If the desired process temperature range is 140°C to 160°C, the following statements could be used to control a heater.

```
240 IF AIN(1) < 140 THEN DOT(5,3,3) = 1
250 IF AIN(2) > 160 THEN DOT(5,3,3) = 0
```

These statements turn on channel 3 (actually the channels from 3 to 3) of digital output port 5. If the temperature is less than 140, the system turns on the heater. If it greater than 160, the system turns it off. The two statements implement control around a nominal 150°C setpoint with a ± 10 °C hysteresis.

In addition to the standard analog and digital I/O commands, μMACBASIC offers further useful commands. A system clock provides timing and timekeeping functions. The statement:

```
WAIT (5.6)
```

causes the program to wait 5.6 seconds before proceeding to the next μMACBASIC statement. However, because μMACBASIC runs in a *concurrent* I/O environment, hardware operations, such as communications, pulse or frequency accumulation, and a/d conversion continue without requiring polling by the BASIC program.

Statements STIME, SDATE, GTIME and GDATE allow the user to set or get the time or date under program control. A TIMER statement provides a stop watch function with a 0.01s resolution.

ADVANCED μMACBASIC FEATURES

μMACBASIC provides the user with a structured modular programming language, borrowing concepts from languages such as PASCAL and ADA. The goal is to allow programmers to divide the overall program into modules that are easier to develop and test. The benefits of this approach are:

- Modular programs are less prone to error and easier to fix. The modules isolate errors to within each module so the user does not have to debug the entire program.
- Several programmers can work on modules independently, improving productivity
- Structured programs are easier to maintain since they are more self-documenting and can be read and understood by persons other than the original author.

The most significant of these features is the use of additional block structures (such as DO . . . WHILE, DO . . . IF . . . REPEAT, etc.) and the ability to create Functions and Procedures.

These Procedures and Functions are blocks of program code stored in memory and accessed by a key word from the main program. A Procedure is a block of code that is similar to a subroutine, in that it executes a particular operation and can be called by name from any section of a program. The program statement below contains examples of both a procedure and a function:

```
10 IF AVG(3,10) > 700, THEN ALARM (2)
```

A standard BASIC language contains neither an average statement, AVG, nor an ALARM statement. The function AVG performs the indicated averaging operation, listed below and calls up the procedure ALARM when the statement is true. Both the function and the procedure can be written with *local* line numbers, variables, and data lists.

```
REAL FUNCTION: AVG
10 INTEGER ARG CHAN, SAMPLE
20 FOR I = 1 TO SAMPLE
30 CUM = CUM + AIN(CHAN)
40 NEXT
50 RESULT = CUM/SAMPLE
PROCEDURE: ALARM
10 DOT (1,3,3) = 1
20 EXIT
```

While both get data from the program, the primary difference between procedure and function is that the function transfers data to and from the program and returns to the main program via a RESULT statement, while the procedure performs an action and exits. The function AVG passes the two values, channel number and the number of samples, from the main program to the function and returns the value in the RESULT statement.

These program modules can be developed by separate programmers and then installed. They can be expanded to include a complete Proportional-Integral-Derivative (P-I-D) control loop and accessed through a command such as PIDLOOP. This ability to develop custom software modules also enables a system house or OEM manufacturer to develop, install and protect proprietary algorithms through the use of the COMPRESS command (see the article on μMACBASIC software in the following pages).

PROGRAM DEVELOPMENT

The μMAC-5000 requires only a dumb terminal for program development. The μMACBASIC firmware will initialize the system and prepare the system for BASIC programming. A personal computer can also function as a development system. Workstation Operating Systems (WOS) for the Analog Devices MACSYM 150, APPLE IIe, and IBM PC allow you to program in μMACBASIC and also allow you to access the printer, disk drive and graphics of the personal computer. WOS packages can be implemented for any other computer in the appropriate high-level languages.

Additional features of the μMAC-5000's software will be discussed in the article on the following pages.

SUMMARY

The industrial signal-conditioning and programming capabilities of the μMAC-5000 open up a number of applications in automated measurement and control. These applications include:

Industrial Process Control and Monitoring: The μMAC-5000 readily performs stand-alone measurements of temperature, pressure, and flow, and can employ them in P-I-D control loops, in addition to analog and digital I/O. μMACBASIC allows the creation of proprietary algorithms that can be transparent to (or concealed from) the user.

Machine/Boiler/Furnace Control: High immunity to electrical noise, 1000V isolation, and the capability of operation at temperatures up to 60°C provide an alternative to programmable controllers where analog I/O is needed in addition to digital I/O.

Remote Terminal Units: The ability to emulate existing communications protocols and to support modem communication opens up applications in remote monitoring and control for pipelines, utilities, and oilfields.

Energy Management: Its low cost, ability to support large numbers of I/O channels, and stand-alone capabilities permit the user to monitor and control large numbers of locations with μMAC-5000s and report back to a central computer for data collection.

Industrial, Government or University R&D: Its ease of use and adaptability make μMAC-5000 the logical choice for monitoring or controlling experiments or pilot plants in a lab environment.

Personal Computers as Data Loggers: The μMAC-5000 can function as a intelligent "front end" for a personal computer to acquire, log, manipulate, and display data from a lab experiment or remote process. ▶

PROGRAM INTERRUPTS, ERROR TRAPPING, PROGRAM WORKSPACES

Advanced Programming Features of μMAC BASIC Real-Time Measurement and Control Programming Language

by John Sylvan

The μMAC-5000's high-level language, μMACBASIC (pages 7 and 8), imparts stand-alone capabilities to the single-board measurement and control system. μMACBASIC extends the standard BASIC language by adding specialized commands and structured programming for real-world monitoring and control.

High-level BASIC commands for analog inputs (AIN), analog outputs (AOT), digital inputs (DIN), and digital outputs (DOT) form the core of μMACBASIC. When combined with the standard library of BASIC commands, plus additional Measurement-And-Control BASIC commands, the μMAC-5000 can perform data logging, digital sequencing and P-I-D control.The modular structure further permits programmers—for OEMs, end users, and system houses—to develop their own high-level program commands. Once stored in memory, these commands can be accessed by the use of a single keyword.

SOFTWARE REQUIREMENTS IN THE REAL WORLD

Program interrupts and error trapping are critical in real-world applications. Since the system monitors and controls real-world events, software errors and long system response time can lead to dangerous situations. A software design which provides self-documentation, error detection, and software-interrupts minimizes this hazard.

PROGRAM INTERRUPTS

In real-time repetitive monitoring, standard BASIC programs run in a top-down fashion, beginning at the first line, executing to the last line, processing any branches and subroutines, and then looping back to the first line of the program. Sequential execution may be tolerable for data processing applications, but the processing of real-world signals for control requires the ability of the program to branch if certain process conditions occur.

A software interrupt alters the execution of the main program to test the status of external flags. After executing each line of code, the system checks the status of the software interrupts. If an external flag is set, the μP executes the procedure called up by the interrupt and then returns to the main program. If the flag is not set, the machine executes the next line of the main program.

μMACBASIC can process six software interrupts. Two are clock-based, two are communication interrupts and two are event-counting interrupts. *Clock-based* interrupts separate time-based events, such as the precise timing required for differentiation and integration with respect to time in a P-I-D loop calculation, from less timely routine data processing. A *communication interrupt* allows users to define a character that, when received at a communication port, will interrupt the execution of the program (e.g., CONTROL-C). Upon receipt of the character, a flag is set, and the system processes the interrupt. *Event-based* interrupts allow the user to specify a predetermined number of events to be counted down. When the event counter reaches zero, it sets a flag and initiates the interrupt routine.

ERROR DETECTION AND HANDLING

Two types of software errors exist—compiler errors (errors found during compiling) and run-time errors—programming errors that manifest themselves while the program is running.

The μMAC-5000 catches compiler errors after the user enters a line of code and hits the RETURN key. Compiler errors are usually things like a misspelled command or wrong syntax. The check for compiler errors eliminates some, but not all, of the errors in a program. If the program decribes what turns out to be an illegal operation, such as a concealed division by zero, a run-time error occurs. Run-time errors halt program execution and—if not trapped by an ON ERROR statement—at best, can cause frustration, and—at worst—can damage equipment or endanger personnel, if the computer is controlling large amounts of energy.

The simple heating tank application shown in Figure 1 demonstrates the dangers of generating a run-time error. In the application, after a number of program steps, the tank's temperature is

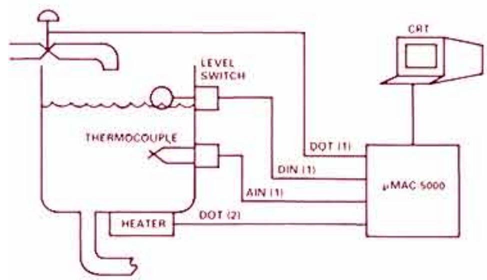


Figure 1. Temperature-profile control system.

programmed to follow a repetitive profile—9 steps of temperature, changing at twenty-minute intervals. The control system automatically maintains the temperature by checking it every two minutes and turning a heater on if the temperature is below the set point and turning it off if above the set point. It also checks the level of liquid in the tank, and adds liquid if the level is below the set point. The following BASIC program controls the process:

```
                                               REMARKS
 10 DIM T(7)                                   7 steps of temp. can
  .                                            be indexed
  .                                            (not enough).
  .
510 FOR M = 0 TO 8                             9 steps of temp. set
520   READ T(M)                                Input Mth setpoint
530   DATA 90,88,85,82,80,83,86,88,89          Setpoints (9)
540   FOR N = 1 TO 10                          10 temp. checks
550     T2 = AIN(1)                            Measure temperature
560     IF T(M) > T2 THEN DOT (1,1,1) = 1      Heater on if temp low
570     IF T(M) < = T2 THEN DOT (1,1,1) = 0    Heater off if temp hi
580     IF DIN (0,1,1) = 1 THEN DOT(1,2,2) = 1 Valve on if fluid low
590     IF DIN (0,1,1) = 0 THEN DOT(1,2,2) = 0 Valve off if fluid hi
600     WAIT 120                               Check again in 2 mins.
610   NEXT                                     Repeat temp. check
620 NEXT                                       New temp. setpoint
```

All seems to be in order, except that at line 10, the programmer has reserved space for only seven, instead of the required nine set points. The operator starts the program running, finds that it is working OK the first few times around, and—after one hour—walks away, expecting the program to cycle endlessly.

Problems arise when the count, M, reaches 7, the memory slot for the setpoint value is found to be missing, and execution stops, with an indication of an INDEX ERROR (or BAD SUBSCRIPT ERROR). If the operator is not there to respond, the program remains in a suspended state while the process continues to run. Any valve or heater that is on will remain that way until the operator corrects the error. Here's what conceivably could happen if the system remains unattended:

Since the temperature profile is in its increasing phase, the heater is probably on; it remains on, and the temperature of the liquid continues to increase. If the valve is on, the tank will continue to fill until it reaches overflow; if the valve is off, the level may drop as the liquid evaporates, boils away, or is drawn off on demand. If the level sinks too low, while the temperature continues to rise, the liquid will start to vaporize, build up pressure (if enclosed), and possibly create havoc.

μMACBASIC has a number of error-handling tools which can "trap" such errors and avoid halting the program.

An ON ERROR GOTO/GOSUB is conditional. It tells the program what line to shift to if, and only if, an error occurs. When the error does occur, the program branches to the specified place, typically a special routine for handling the error. The limitation of the command is that it *must* be executed before the error occurs. In the above example, the following might be added:

```
  20 ONERROR GOTO 1000
   .
   .
   .
1000 DOT(1,1,1) = 0          Heater off
1010 DOT(1,2,2) = 0          Fluid flow off
   .
   .
   .
```

When the run-time error occurs, the error command traps it by shifting to the segment starting at line 1000, which turns off the heater and flow of fluid, may sound an alarm, print out an error message—perhaps including the state of all variables in the system and the number of the command at which the error occurred—and might even restart the program. Without error trapping, the program (but not the process) would halt when the machine detected a request for non-dimensioned memory elements.

Logical errors are eliminated from programs by careful planning and thorough testing. Error conditions that must be anticipated are primarily external or mechanical errors, e.g., the failure of a card or sensor, or improper operator response from the keyboard.

Proper documentation also helps avoid errors. The design of μMACBASIC improves documentation through structured programming and lengthened variable names. Most BASIC languages limit unique variable names to two characters such as A=1, A2=3 and T4=55 (additional characters may be permitted, but the computer could not distinguish between variable names starting with the same two characters, e.g., SEtpoint and SEttingFactor). The user may wind up with a program looking like:

```
10 SP = 50
20 SF = 5
30 T1 = AIN(12)
40 ES = T1 - SP
50 AOT(4) = ES * SF
60 WAIT(5)
70 GOTO 30
```

Anyone other than the programmer, looking at the program, has difficulty understanding what SP, SF, T1, and ES represent. μMACBASIC permits unique variable names of up to 256 characters, starting with a letter. Thus, variable names are long enough to give the reader of a program insight as to the meaning of a variable. The underscore character is permitted, resulting in variables that appear as two or more words, separated by underscored spaces. The next program demonstrates improved documentation.

```
10 REM  TEMPERATURES ARE IN DEGREES C
20 SETPOINT = 50
30 SCALING_FACTOR = 5
40 VAT1_TEMP = AIN(12)
50 ERROR_SIGNAL = VAT1_TEMP - SETPOINT
60 AOT(4) = ERROR_SIGNAL * SCALING_FACTOR
70 WAIT(5)
80 GOTO 40
```

The program measures the temperature input on analog input channel 12, calculates an error signal from the difference between the setpoint and temperature at channel 12, and outputs a scaled analog control signal on channel 4. Each variable is identified with a descriptive name that indicates the variable it represents.

The program also illustrates a feature that simplifies BASIC programming for real world measurement and control. The temperature reading on channel 12 is in degrees centigrade. Built into the firmware on the μMAC-5000 are programs to turn raw sensor data into the proper engineering units. For thermocouples, the user programs the type of sensor input on a specific channel with an INTYPE statement. Whenever the system accesses that channel, it automatically selects the proper amplification, compensates for cold-junction temperature, and linearizes the signal, and then converts it to the proper engineering units. The number of program steps is thereby reduced to one: AIN(12).

ADVANCED μMACBASIC FEATURES

In measurement and control, many actions are repetitive in nature. The system performs similar calculations or procedures on various pieces of data as it scans input and output channels. μMACBASIC simplifies this kind of data processing by allowing the programmer to add new keywords (Procedures and Functions) to call complex operations. For example, to run a series of digital outputs, you could create a procedure with the name SEQUENCE. Subsequently, each time you entered the keyword, SEQUENCE, the system would execute that procedure.

μMACBASIC's main program and Procedures/Functions use their own areas of memory, called *workspaces*. In a workspace, the language allocates room for variables and for workspace code. These workspaces offer the benefit that a user does not have to call out a cryptic subroutine somewhere in the main program sequence, such as GOTO 1000. Since Functions and Procedures occupy their own workspaces, it is impossible to inadvertently access them as one might a BASIC subroutine.

A Function assumes a value calculated within the Function's code block, returns a value to the main program, and can be used in expressions and evaluated like a variable. An example of a function in a standard BASIC system would be a command such as SIN(X) which calculates the value of the sine of X. A typical Function in μMACBASIC would be AIN, which obtains the digital value of a specified analog input signal in appropriate engineering units.

A Procedure is a block of code similar to a subroutine. Within a Procedure is a series of commands that trigger certain actions such as sound an alarm and display a message. However, Procedures have important advantages over subroutines: with their own workspace and line numbers, Procedures cannot be executed inadvertently, since their internal code is invisible to the main program; subroutines can operate on all variables within a program (global operation), while Procedures can operate on local and/or global variables, as desired—thus, a Procedure can be prevented from changing or accessing any variable in the main program; subroutines are executed by a line-number branch (e.g., GOSUB 1000), while Procedures are executed by name (ALARM), becoming a keyword within the language, suggestive of the operation performed. Once completed, Functions and Procedures return to the points in the main program from which they were called.

The user writes Functions and Procedures in BASIC. Once this code has been written and debugged, it can be permanently added to the system by using a COMPRESS command. The compression preserves the object code and a few essentials, but eliminates the source code, reducing the size of the module by nearly 50%. Once compressed, the module cannot be edited or listed—it is inaccessible to the programmer. Users can also transfer programs to nonvolatile electrically programmable ROM for permanent storage with a PROGRAM MODULE command. This ability to Create, COMPRESS, and PROGRAM MODULE functions and procedures into ROM permits programmers for system houses and original equipment manufacturers to develop proprietary software for end users and protect their software investment.

Designed for harsh industrial environments, the μMAC-5000 protects programs from inadvertent erasure during power supply interruptions. The up-to-8 modules in RAM can only be eliminated by disconnecting both ac power and the battery backup.

The program below shows the use of a Function in Proportional, Integral and Derivative control. The PID calculation is shared among twelve independent loops for closed loop control.

```
  10 REM   12 CHANNEL PID LOOP CONTROL
  20 DIM   RESET(12), UPDATE_TIME(12), INTEGRAL(12),
           DERIVATIVE(12), PROP_BAND(12), RATE(12)
  30 FOR C      0 TO 11: REM  C     CHANNEL
  40   PROCESS_VARIABLE      AIN(C)
  50   AOT(C)      PID_LOOP(C)
  60 NEXT

REAL FUNCTION: PID_LOOP
  10 EXTERNAL RESET, INTEGRAL, DERIVATIVE,
     UPDATE_TIME, PROP_BAND, SETPOINT, RATE,
     PROCESS_VARIABLE
  20 INTEGER ARG D
  30 PROCESS_ERROR      PROCESS_VARIABLE
     SETPOINT(D)
  40 INTEGRAL(D)     RESET(D) * (INTEGRAL(D) +
     PROCESS_ERROR) * (UPDATE_TIME(D)      TIMER)
  50 DERIVATIVE(D)     RATE(D) * (PROCESS_ERROR -
     DERIVATIVE(D))/(UPDATE_TIME(D) - TIMER)
  60 CONTROL_OUTPUT   PROP_BAND(D) *
     PROCESS_ERROR + INTEGRAL(D) + DERIVATIVE(D)
  70 UPDATE_TIME(D) = TIMER
  80 RESULT = CONTROL_OUTPUT
```

The second line of the program saves space in memory for the P-I-D variables for the 12 different PID control loops. Since the user wants to maintain separate parameter and variable values for each P-I-D loop, the arrays provide this function. The proper data for each loop can be entered with an INPUT statement in the main program. A FOR/NEXT loop circulates through all twelve channels, measures an analog input parameter for each channel, and calculates the proper response with the use of the function PID_LOOP.

At each pass through the loop, C increments and each variable takes on a value corresponding to either the channel or data in the array with that location. For example, in the first pass through, PROCESS_VARIABLE = AIN(0), the analog input at channel 0, since C now equals 0. Similarly, with the variable SETPOINT, the program instructs the system to go to the first memory location in the array, the point labeled (0), and use that data for calculations.

The Function PID_LOOP performs the P-I-D calculation, producing an output proportional to the sum of the error, its integral, and its derivative, multiplied by the appropriate coefficients. The integral is generated at line 40 of the PID_LOOP, and its summation and multiplications are performed in line 60.

At line 50 of the main program, the following happens: the system reads the instruction and goes to the function, it then executes the eight lines of of the Function and then returns with the value of RESULT; this value is then converted to an analog output at the corresponding channel, to close the control loop. The use of the function PID_LOOP permits one or more programmers to develop software independently of each other. Since procedure and function modules have independent line numbers, conflicts for specific portions of memory space are minimized.

In systems where a μMAC-5000 is used for its powerful analog capabilities, it might also be used as a Programmable Controller, if digital control is necessary. A PC must operate with inputs and outputs at industrial line voltages, and it must be designed for reliable operation in harsh industrial environments, at ambient temperatures of 60°C or more. The hardware configuration of the μMAC-5000 meets these requirements.

The conventional PC is an event-driven device. That is, a certain action is taken if a given event occurs. The events in this program are digital inputs triggered by switch closures on an operator panel. Each switch selects a different sequence of actions. Figure 2 is an example of the use of μMACBASIC in a PC application, employing two Procedures. On closure of a switch, SEQUENCE1

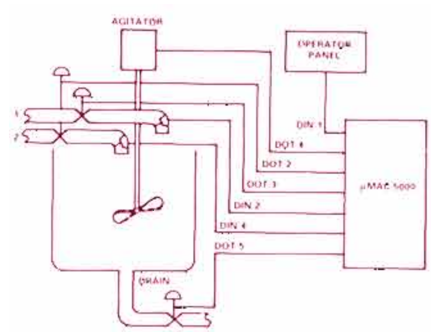


Figure 2. Mixture-control system using programmable-controller techniques.

is intended to fill a tank with two fluids, mix them, and drain the resulting mixture. If either of the two fluids does not flow into the tank, an ALARM Procedure will shut off the intake valves and provide a message for the operator.

```
  10  REM  * PROGRAMMABLE CONTROLLER *
  20  ON_CHANNEL = DIN(0,1,1)
  30  IF ON_CHANNEL = 1 THEN SEQUENCE1
  40  GO TO 20
PROCEDURE: SEQUENCE1
  10  DOT(1,2,2) = 1
  20  DOT(1,3,3) = 1
  30  IF DIN(0,1,1) = 0 OR DIN(0,2,2) = 0 THEN ALARM
  40  IF TRIGGER = 1 GOTO 150
  50  TRIGGER = 0
  60  WAIT(15)
  70  DOT(1,2,2) = 0
  80  DOT(1,3,3) = 0
  90  DOT(1,4,4) = 1
  100 WAIT(20)
  110 DOT(1,4,4) = 0
  120 DOT(1,5,5) = 1
  130 WAIT(15)
  140 DOT(1,5,5) = 0
  150 EXIT
PROCEDURE: ALARM
  10  TRIGGER = 1
  20  DOT(1,2,2) = 0
  30  DOT(1,3,3) = 0
  40  PRINT "NO FLOW IN ONE OF THE LINES, TURN OFF
SWITCH"
  50  WAIT(5)
  60  EXIT
```

The program responds to a switch closure on the operator panel. Closing the switch (at Channel 1) produces a logical "1", which calls the procedure SEQUENCE1 at Line 30 of the main program. The first two statements turn on the valves for lines 1 and 2. If there is no flow in one of the lines, the program invokes the Procedure ALARM, which shuts the valves and prints an alarm message; on execution of line 40 of SEQUENCE1, if the valves have been closed, SEQUENCE1 exits. Otherwise, the valves remain open for 15 seconds, then are closed. At that point, the agitator turns on for 20 s, turns off, and then the vessel is emptied for 15 seconds.

The program could be improved through the use of Interrupt statements. Rather than timing the agitator, each revolution of the agitator could generate a pulse output which would be counted down to zero and generate an event-based interrupt. This would save the 20 seconds the system would otherwise wait idle. Similarly, pulse-type flowmeters could count down from a preset volume of flow, then generate an interrupt, to save the 15 seconds of computer idleness during the fill time. A level-sensor in the drain could determine when the tank has been emptied, saving the 15 seconds of drain time.

Additionally, applications requiring faster execution of portions of the program can be programmed in assembly code. The processor executes the code directly, avoiding the compiler. μMACBASIC accesses assembly language routines from the main program with a CALL statement.

Another key application of computer-based equipment in industrial automation is data collection. Equipments, such as data loggers, collect and store large amounts of data for record keeping or analysis. The μMAC-5000's up-to-64K of memory and data-formatting commands can be used to scan a large number of input channels, automatically convert the information to engineering units, and then store arrays of data.

Arrays are groups of cells, each capable of storing a single value. Arrays in μMACBASIC may have a number of dimensions limited only by available memory space. Like other variables, arrays have names, but in addition, they also have subscripts that indicate which cells are being referred to. An example is:

```
LET WATER_TEMP(60,3) = 37.5
```

This statement assigns a value 37.5 to a cell in a two-dimensional array—the 60th cell in the first dimension and the third cell in the second dimension. The programmer saves space for these arrays in memory by the use of a dimension (DIM) statement. For this array, a dimension statement could be:

```
DIM WATER_TEMP(100, 10)
```

allocating enough storage space for 100 × 10 real numbers or 1000 elements.

A common way to collect data and fill arrays is with a FOR/NEXT loop. By nesting loops within one another, the programmer can fill each dimension of a multi-dimensional array. For example, the following program scans 48 analog input channels and records the hour of each scan. The main program would interrupt, i.e., halt execution once an hour and call out the subroutine.

```
10  DIM  PROCESS_TEMP(24,49)
20  GTIME(HOUR,MIN,SEC)
30  IF HOUR <> PREVIOUS_HOUR GOSUB 100
40  GOTO 20
100 PROCESS_TEMP(HOUR,48) = HOUR
110 FOR CHANNEL = 0 to 47
120   PROCESS_TEMP(HOUR, CHANNEL) = AIN(CHANNEL)
130 NEXT
140 PREVIOUS_HOUR = HOUR
150 RETURN
```

The main program repeatedly collects the time with a GTIME statement. When the hour changes, HOUR will not be equal to PREVIOUS_HOUR, and the the program will access the subroutine at line 100. The last array location is filled with the hour of the scan. The CHANNEL FOR/NEXT loop fills the first 48 cells in that array with the data from each of the 48 analog input channels. The return statement returns program execution to the (in this case very short) main program where it will check the time iteratively until the next change of the hour (if the instruction in line 50 were at the end of a much longer program, the hour checks would be less frequent).

Data in the array can be printed out at intervals chosen by the programmer. In addition, by using a Work Station Operating System (WOS) (see page 8), with one of the readily available graphics packages for personal computers, the data can be displayed graphically.

By skillful programming of the μMAC-5000, closed loop control, data logging, sequence control and other data processing operations can be combined. Users can thus replace dedicated instruments, e.g., data loggers, programmable controllers, and analog controllers, with a single flexible design. The serial communications with a host computer (or stand-alone operation) makes for truly distributed measurement and control in factory automation.

12-CHANNEL ANALOG EXPANDER BOARD FOR μMAC SYSTEMS

Less Than $100 per Channel for Low-Cost Isolated Sensor-Based Measurements
Interface Thermocouples, Current Sources, Low-or-High-Level Voltage

The μMAC-4015* is a low-cost, 12-channel, isolated input expander board for use with both the μMAC-4000* and the new stand-alone BASIC-programmable μMAC-5000* measurement-and-control systems. It accepts 12 channels of analog input, provides ± 1500 V input-to-output and channel-to-channel isolation, plus filtering, gain, and common-mode rejection.

Available in two low-level (mV) voltage versions, a current- or medium-level voltage-version and a high-level-voltage version, it accepts thermocouple signals and other voltage inputs, as well as process currents. Designed as a cost-competitive all-electronic alternative to isolated systems employing mechanical relays for harsh industrial environments, the μMAC-4015 serves applications in process control, product test, high channel-capacity R & D, multipoint recorders, and heat-profile systems.

All 12 channels on each μMAC-4015 provide both protection and performance: ± 1500-volt common-mode isolation and 130-volt normal-mode input protection—plus low-drift preamplification (± 0.003% of span per °C), filtering, open-circuited input detection, and 160 dB of common-mode rejection. Its transformer isolation and all-solid-state circuitry eliminate concerns about reliability and stability often associated with relay-isolation techniques.

All 12 channels on each of the four optional board types (see table) are configured at the factory for the specified input type and span range. The span appropriate to the actual input types to be used is set by the user, via a DIP switch. Typical accuracies are to within ± 1°C for type J thermocouples, ± 1.1°C for type K thermocouples, and ± 0.08% of full-scale for a ± 25-millivolt input signal.

Input Types Available
± 25 mV, or T, S, R, or B thermocouples
± 50 mV, or J, K, W thermocouples
± 100 mV, E thermocouples, 4-20 mA, ± 20 mA, ± 1 mA
± 5 V, ± 10 V

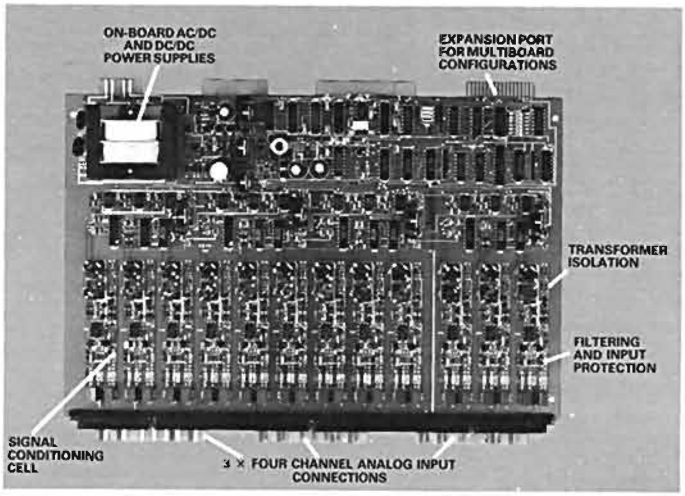
The μMAC-4015 connects to the master board via the expander port. Analog signals are preconditioned, and the analog output is fed to the a/d converter on the master board. The digital logic in the master μMAC-4000 or BASIC-programmable μMAC-5000 board determines which μMAC-4015 channel is chosen for conversion and enables the appropriate expander board's output.

The board accepts both ac (4 voltage-range options) and 24-volt dc power, deriving its required voltage levels from either source. As with other μMAC boards, if the ac supply is interrupted, power is smoothly switched over to the 24-volt source.

APPLICATIONS

The basic application of the μMAC-4015 with a μMAC Master Board is shown in Figure 1. Control signals from the Master determine which channel's high-level analog output appears on the analog output bus as an input for conversion at the Master Board.

*Use the reply card for technical data. See also the article starting on page 3.



The μMAC-4015 conditions large numbers of analog input signals with similar ranges at low cost. It plugs directly into any expansion socket in μMAC 4- or 7-slot card cages. Up to six μMAC-4015's may be used in a μMAC-5000 cluster for a total of 84 analog channels per cluster, with up to 672 channels on a single 20-mA multidrop communication line (see page 6). With thermocouples, an isothermal AC1800 screw-terminal connector for each 4 input channels provides for cold-junction compensation.

The μMAC-4015 is especially cost-effective when used for applications involving measurement of large numbers of points, e.g., from 50 to 150. Typical environments include *control rooms*, where the outputs of large numbers of thermocouples are monitored; *retrofit applications*, replacing multipoint recorders or data loggers for computer-based systems; *heat profiles* for boilers and furnaces; *monitoring and control* in industries requiring high-voltage isolation.

Since software drivers are available for personal computers (for example, Apple II and IIe, IBM, HP-85/86/87,—in addition to DEC's RT-11 and RSX operating systems—(see *Analog Dialogue* 17-2), they can easily be used as hosts for μMAC-4000 systems; the μMAC-4000 and its expansion channels become front ends to implement PC-based intelligent measurement-and-control systems involving large numbers of analog inputs at low cost. ▶

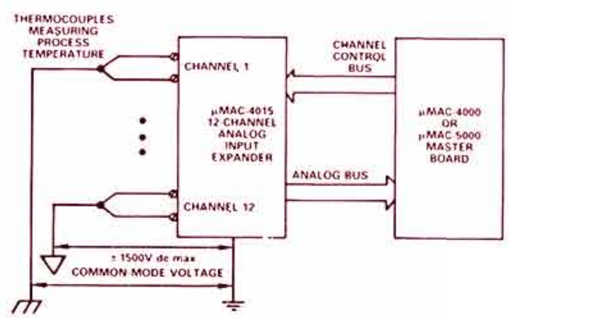


Figure 1. Typical thermocouple application (input isolation to ± 1500 V).

MUX200 INCREASES CHANNEL CAPACITY OF MACSYM 350 SYSTEMS

Form Distributed Analog/Digital Input/Output Systems with MACSYM 150 Using MACSYM 200s, ADIO Card Library and All MACBASIC ADIO Statements

by Bill Schweber

The MACSYM 350* (see *Analog Dialogue* 16-3), is a powerful measurement and control computing system, which accepts sensor-based analog and digital measurement inputs and provides analog and digital outputs for control, via a variety of analog and digital input and output cards. The system is programmable in MACBASIC (an enhanced, multitasking version of BASIC) and uses the MP/M-86 Operating System (from Digital Research, Inc.)

The MACSYM 350 extends the analog/digital capability of the basic MACSYM 150 by combining with it the accuracy, sampling speed, signal conditioning, and local data storage of the MACSYM 200 intelligent front end. The MACSYM 200 performs software-selectable 12- or 16-bit a/d conversion; its 16-bit CPU controls the Analog/Digital Input/Output (ADIO) backplane and handles local data-processing needs. ADIO cards can plug directly into the 16-slot backplane; more than 30 types are available for interfacing to virtually any real-world sensor, transducer, or actuator, via field wiring. Communication between the MACSYM 150 and the 200, via RS-422, is completely transparent to the user.

Many applications require more than a single MACSYM 200: some call for the I/O terminations for field wiring to be at physically separated locations; others simply require more additional I/O than a single MACSYM 200 can handle. With these needs in mind, the MACSYM 200 Multiplexer, the MUX200, was developed.

The MUX200 facilitates the connection of more than one MACSYM 200 to a single MACSYM 150. With a single MUX200, four MACSYM 200s (up to 64 cards) can be connected (Figure 1), and additional MUX200s make it possible for as many as 15 MACSYM 200 units (240 cards, with very large numbers of I/O points) to be connected at distances of up to 1000 ft (300 m), from node to node—and 5000 ft (1500 m) from the farthest MACSYM 200 to the MACSYM 150.

Figures 2 and 3 show radial and multidrop configurations for interfacing with up to 15 MACSYM 200s, using 5 MUX200s.

When MUX200s are used, the software transparency of the MACSYM 350 is retained. Thus, the MUX200 makes available flexible distributed systems to MACSYM users at low cost for the additional hardware, and neligible software penalty.

*For technical data use the reply card.

MUX-200 HARDWARE

The objectives fulfilled by the MUX200's hardware (Figure 4) are:

•To act as a fanout buffer for signals *from* the *MACSYM 150* to the MACSYM 200s.

•To act as an OR gate for signals from the MACSYM 200s *to* the *MACSYM 150*.

•To be flexible with respect to power, i.e., to allow the MUX200 to receive its operating power from either local ac lines or from user-furnished 5-volt supplies.

•To be resistant to electromagnetic interference (EMI), since the MUX200 and its associated cables will be in noisy environments and may possibly have long cable runs and high ground-potential differences.

•To be transparent with respect to speed, i.e., to allow data to be exchanged at the normal high rate of 307.2 kilobytes/second.

Data and timing lines between the MACSYM 150 and MACSYM 200 employ RS-422; control lines employ RS-423. The line drivers and receivers used in the MUX200 are designed to allow MUX200s to be daisy-chained to achieve longer distances and permit more than 4 MACSYM 200s to be interconnected.

The built-in power supply converts ac line power to +5 V dc, and the internal dc-to-dc converter furnishes an additional −5 V dc. Instead of relying on the ac line, the user may provide the +5V power directly from an external source.

The MUX200's shielded enclosure provides protection against EMI; for long distances between nodes, the user should provide a cable that has an extra cable shield wired to the connector pins.

Because the cost of the MUX200 is essentially that of buffers, gates, a power supply, and mechanical hardware, it is an inexpensive complement to a MACSYM 350 system, costing about the same as a sophisticated ADIO card or a long run of cable.

OPERATION

The user interconnects the MACSYM 150, MUX200s, and MACSYM 200s, and then sets a DIP switch in each MACSYM 200 to a unique number from 1 to 15. This tells each box what its number is and how it shall be identified by the user's program. Any subset of the 15 numbers may be used, and the logical assign-

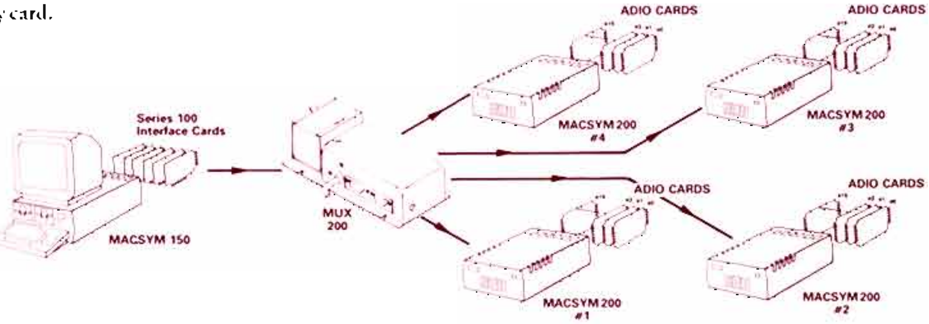


Figure 1. Four MACSYM 200s with one MUX200.

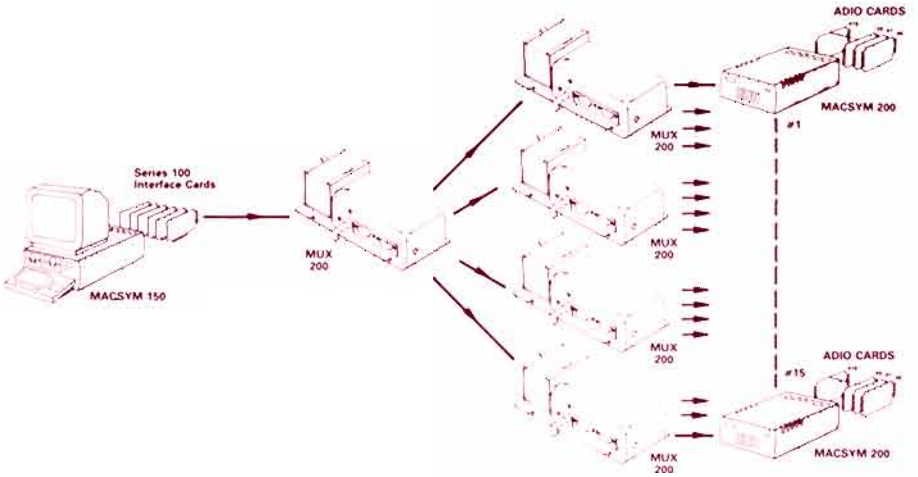


Figure 2. Up to 15 MACSYM 200s, using five MUX200s in a radial configuration.

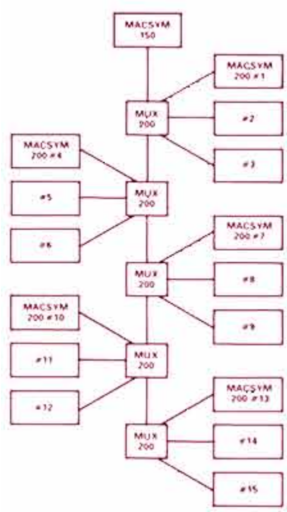


Figure 3. Up to 15 MACSYM 200s, using five MUX200s in a multi-drop configuration.

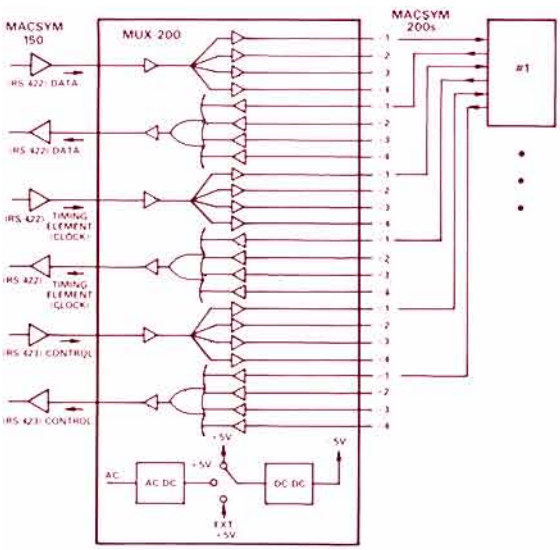


Figure 4. MUX200 block diagram.

ment of box numbers doesn't have to correspond with the MACSYM 200s' physical placement. The box numbers may be assigned arbitrarily.

The programmer then directs the ADIO statements from the MACBASIC program to particular MACSYM 200s by using the box number in the ADIO syntax. For example,

AIN:3 (slot, channel) means an analog input from the indicated slot and channel number of MACSYM 200 number 3.

DOT:12 (slot, channel) means a digital output for the indicated slot and channel number of MACSYM 200 number 12.

All ADIO statements, including the high-speed SCAN and COLLECT and the MACSYM 200 reset statement, XRESET, can be used with the box number.

Each ADIO statement from the MACSYM 150 appears at the inputs of all the MACSYM 200s, via the MUX200s. Each MACSYM 200 ignores all statements except those having a box number that agrees with the physical setting of its DIP switch. At the targeted MACSYM 200, the message is accepted, interpreted, and processed, and the appropriate response (for example, a returned analog input value) is prepared. Each response that is to be returned to the MACSYM 150 is put into proper format by the MACSYM 200, and its box number is also attached.

For such ADIO statements as AIN, AOT, DIN, and DOT, there is a single response for each statement sent by the MACSYM 150. Therefore, the MACSYM 150 expects one—and only one—data transmission from the group of MACSYM 200s. But some other ADIO statements, such as SCAN, COLLECT, and AVERAGE, involve data collected over a longer time frame; thus, the returned values are transmitted at a time that cannot be predetermined.

Rather than hamper the system user by not allowing any ADIO statements until a SCAN, COLLECT, or AVERAGE is completed, the software in the MACSYM 150 uses a polling scheme to resolve conflicts for use of the communications link to transmit data from the MACSYM 200s back to the MACSYM 150. This allows a number of MACSYM 200s to operate, executing various commands at the same time. The data packets returned from the various MACSYM 200s are handled by the polling scheme and sorted out so that all the values are eventually returned. Some transmissions will have to wait for transmissions from other MACSYM 200s to be completed. The entire process is transparent to the user's software; the communications link and polling do not have to be included in the user's program.

CONCLUSION

MACSYM 200 multiplexers offer MACSYM users a convenient, low-cost, easy-to-use way for a MACSYM 150 measurement-and control computer to communicate with up to fifteen MACSYM 200 intelligent front ends. ▣

DATA ACQUISITION INFORMATION SYSTEM:

SAE's Way of Providing an Extra Level of Software Support
Menu-Driven Monitoring System for Industrial Uses with MACSYM 150 AND 350

by Don R. James

Industrial users of measurement and control systems can choose to design, install, and write the software for their own systems—or to contract the job to an engineering firm. As control-system computers evolve away from general-purpose minicomputers and towards fully integrated modular control computers, more firms are considering installing and programming their own systems, since a dedicated control system's integrated hardware for industrial automation has the potential for lower hardware costs and simplified software development.

The cost of user-developed applications programs, however, may still be comparable to the expense of hiring an engineering firm to develop the software, since each company's problem requires a custom solution, which may not be easy to implement in software.

An intermediate approach is to purchase well-conceived basic applications software to minimize the amount of user programming effort. In the approach to be described here, Systems Application Engineering, Inc., of Houston, Texas, uses their system expertise in industrial measurement and control to develop proprietary applications software for the Analog Devices MACSYM 150. SAE's approach—offering off-the-shelf menu-based products that meet up to 90% of the software required for a broad class of applications—lowers the user's cost of automating open-loop monitoring. Typical customer groups that benefit include:

Utilities: Dedicated or distributed monitoring of electrical distribution, generation and transmission.

Process Plants: Processing sensor data for automatic control schemes, such as adaptive, analyzer-based, or conventional control.

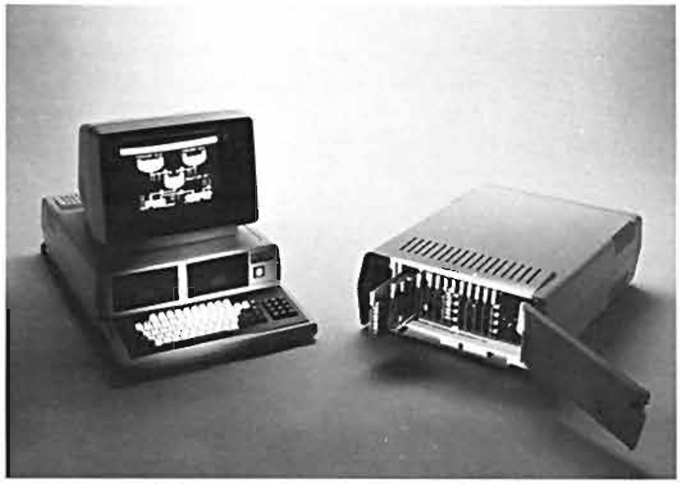
Oil and Gas: Injection control and monitoring, interrogation of wellhead equipment and implementation of pump-off control, lease automatic custody-transfer, and information for production accounting or reservoir management.

Manufacturing Automation: Applications to sequence control, sorting, searching, and selecting.

Systems Application Engineering, Inc., specializes in applying automation to the solution of real-time industrial measurement, test, control and information-reporting problems. As consultants specializing in mini- and microcomputer control systems, SAE has gained a large amount of experience in over 150 projects, leading to a deep understanding of the underlying software requirements for many applications.

At present, because software for industrial automation is not standardized or—in general—transportable, generic software must be developed for a specific computer system. The choice of a system for this purpose is a serious matter, because the hardware and the software are essentially partners in a marriage.

To function in a broad spectrum of applications, the computer system must be flexible in both the number and types of input and output channels, in processing capability, and in communications. The MACSYM 150 and MACSYM 200 subsystems provide this flexibility, employing an integrated modular design (circuits, hardware, and software) which allows the user to custom-configure the number and types of I/O channels. Plug-in signal conditioning boards connect the system to analog and digital sensors (inputs) and control (output) circuits. The number of I/O channels can be further expanded by means of communication links to remote measurement and control subsystems for monitoring and control system designs.



Figures 1 and 2 depict the hardware of SAE's small and medium DAiS (Data Acquisition information System) configurations. The small DAiS system uses a host MACSYM 150 computer, which communicates over a high-speed serial link to a single MACSYM 200 measurement and control subsystem. The host computer contains operator interfaces, such as CRT monitor, keyboard for entering data, printer, and disk storage. The MACSYM 200 intelligent remote front end has a backplane with 16 interchangeable slots. Each slot accommodates an analog or digital input/output card, selected to match the application.

The medium DAiS system provides further channel expansion with the addition of a multiplexer (see page 14) and MACSYM 200 measurement and control subsystems. One or more MACSYM 200s operate under the supervisory control of the MACSYM 150.

Both the number of I/O slots and the processing speed of the computer limit the attainable size of a control system. The system must accept real-world data, process it, and respond in a timely fashion. A high-speed CPU allows the system to process the data in real time with a minimum of execution time. The MACSYM 150 delivers a factor of two or more throughput improvement over CPUs used for many popular computers, due to its advanced microprocessor hardware (Intel's coprocessor chip-set, 8086/8087), and—in software—its multitasking MACBASIC, a compiled language, which minimizes program-execution time.

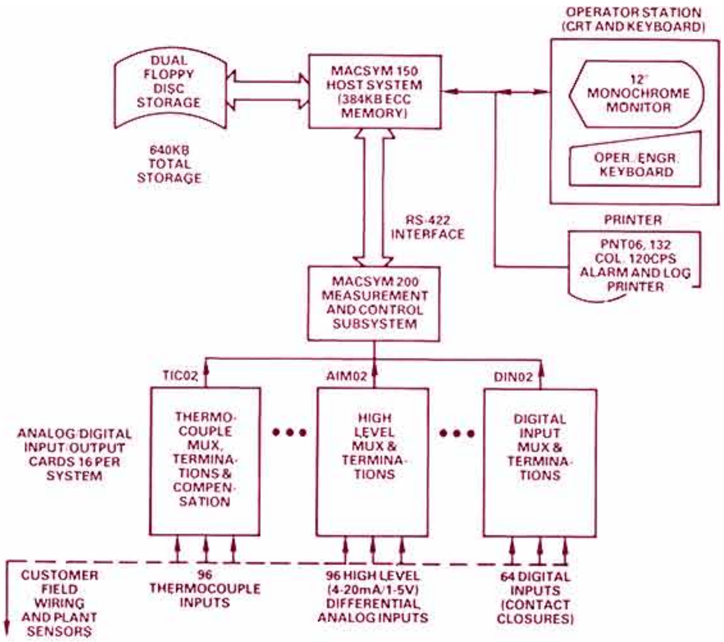


Figure 1. Small DAiS system: MACSYM 150, MACSYM 200, ADIO cards, and peripherals.

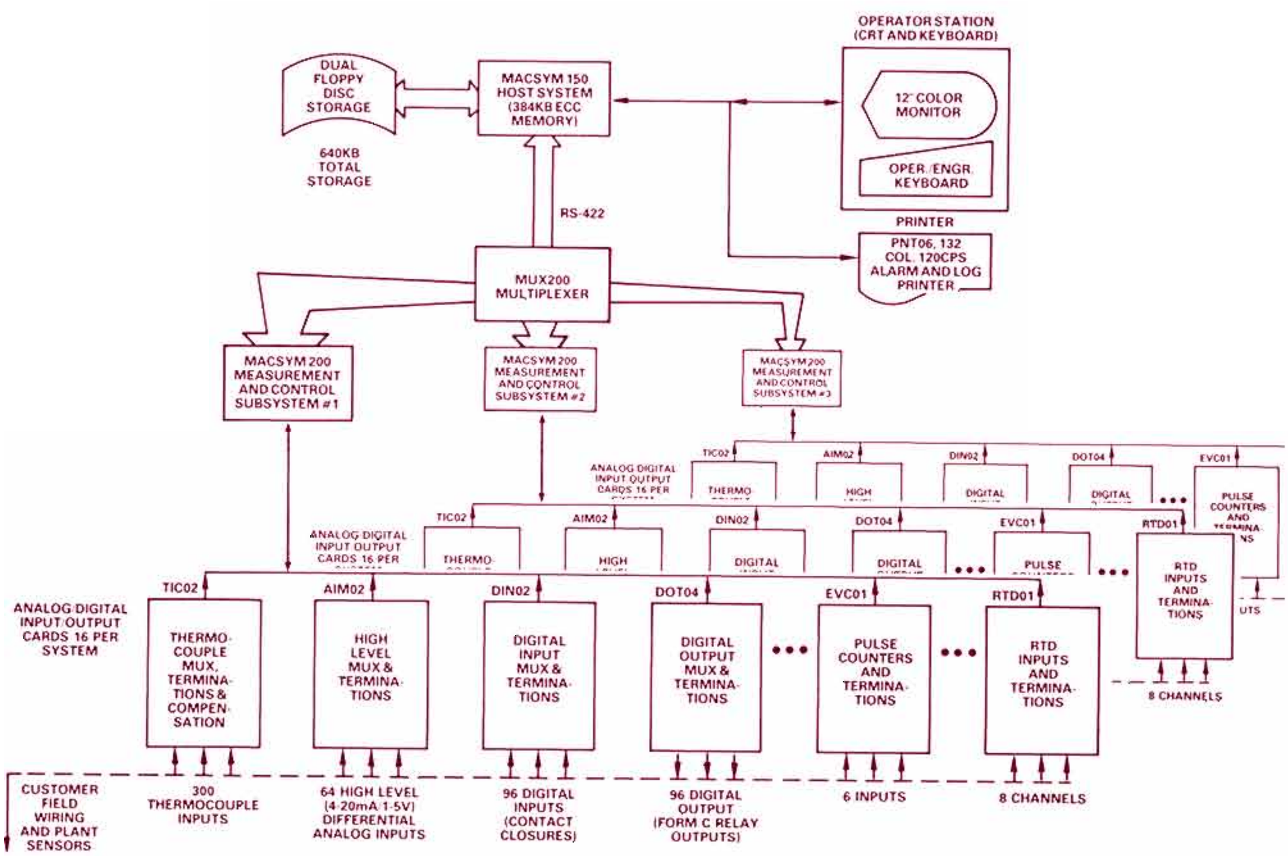


Figure 2. Large DAiS system: MACSYM 150, MACSYM 200s, multiplexers, ADIO cards, peripherals.

SOFTWARE DESIGN

Software coordinates with the hardware architecture to form an overall control structure. The hardware design includes the computer and all its interfaces to the external process and the human environment. The process environment includes: analog inputs for sensor data, analog outputs for control signals, digital inputs to detect switch closures, and digital outputs for relay switching. The human environment includes CRT displays, alarms, and data storage on disk or hard copy format from a printer.

Purchased without the DAiS software, which anticipates the kinds of programs needed for industrial applications, MACSYM requires the programmer to write instructions for collecting real world data, computing the required control signals, and formatting data for display with a sequence of program commands. MACSYM is easy to program in MACBASIC 3, even if one is not a professional programmer, but programming is still exacting, time-consuming work, and many of the resulting programming sequences turn out to be similar to those written by other users for different applications; they essentially involve "re-inventing the wheel."

Here's an example of MACBASIC: two MACBASIC 3 statements in the following program cause a digital input event (a tank level that exceeds a prescribed limit) to produce a digital output event (turning off the flow to the tank) and an operator display.

```
10 IF DIN(1,12) = 1 THEN DOT(2,12) = 0
20 IF DIN(1,12) = 1 THEN PRINT "LEVEL OF TANK A
   EXCEEDS LIMIT"
```

The meaning of line 10 is: "If the digital input from a sensor connected to slot 1, channel 12, is 1 (on, up, true, etc.), then the digital output at slot 2, channel 12, to a valve actuator, is set to 0 (off)." In a measurement and control application with hundreds of I/O points and numerous displays, the control program can exceed 1000 lines. With the estimated installed (and debugged) cost of a single line of program reaching $35, any reduction in the number of lines of new program offers substantial cost reductions.

SAE uses MACBASIC 3 to develop their proprietary *Data Acquisition and information System*. The result—from the user's point of view—is that, rather than generating an applications program by connecting a series of individual program commands, an end-user configures the software for DAiS and operates it, through a number of menu-driven software functions. DAiS software functions include:

- Specification of parameters by fill-in-the-blanks CRT displays
- Conversion of analog measurements to process engineering units
- Priority alarm/event detection and reporting
- User-specified logging—periodic and on demand
- Overview, group, and individual data displays
- Access to data by point or group name
- Access to displays via software menus
- Analog or digital real-world outputs
- Diagnostic, utility and documentation functions
- Extended history/trend function.

The user accesses each function by calling up a CRT display. No additional user-program steps are necessary; the software functions contain all the necessary programming to automatically acquire process data, convert it to engineering units, generate alarm/event reports, and provide operator/engineer interfaces.

A menu structure allows users to upgrade their existing approaches to monitoring and control by means of an integrated and automated system. Since operators monitor and control process parameters through the various system displays, and no computer programming is involved, it is unnecessary to train operators as computer programmers. This is similar to the approach popular in office environments for automated order-entry systems, where the user accesses data and enters information through menus on the computer screen. An example of this type of software is an airline reservation system where seating assignments can be checked by entering the passenger's name and having the system respond with the passenger's seating assignment. Some recent personal computers also take advantage of menu-type programming so that operators can get useful results with a minimal amount of programming effort.

Before the system can acquire and convert process data, however, the user must initialize the system by providing information on the parameters of each input channel. The first display is the "menu" function, which provides access to all other functions. Figure 3 simulates the choices given on the screen for a typical main menu.

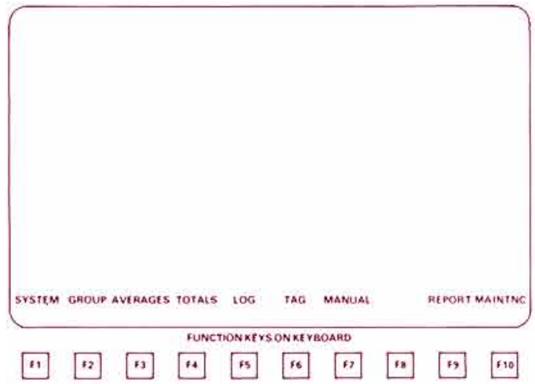


Figure 3. Typical main menu.

From the main menu, the user is routed through a series of menus until a digital- or analog-variable detail screen has been reached (Figure 4). In each case, the operator selects the desired display by pressing the function key below the appropriate displayed choice. Each one of these menus is a subset of the function selected from the main menu (Figure 5), which leads the operators into a final level of passive displays for information-only, and active displays for engineering maintenance (parameter adjustment) or data entry.

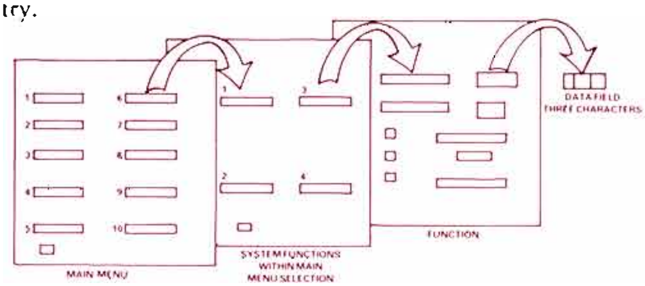


Figure 4. Progression of menus.

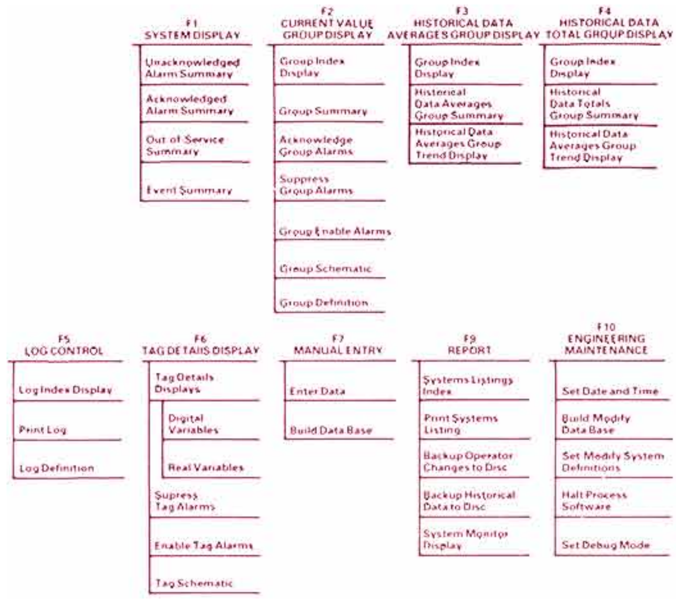


Figure 5. Typical subheadings of second-stage menus.

Data entry displays allow programmers to enter a data field, where they can enter initial data or alter, delete, or add to the information in that data field. As Figure 4 shows, this can be done by entering two consecutive display choices on the menu screens and inserting information into the data field on the final screen. In addition to the standard functions, SAE can develop custom displays for the user.

Figure 6 details a Digital Variable Detail Display (via F6). As Figure 5 shows, a similar display allows entry of real variables (i.e., analog measurements. Once a user has established the parameters for an input channel, (s)he accesses this screen and inputs the required data. For example, the user goes to the data field on the screen for the card address and enters the Location, Slot Number and Channel of the digital input or output. As shown, engineers and operators can also enter the attributes of alarm conditions including: normal state, alarm state, digital-variable-to-set ("Set Tag") on alarm state, alarm priority, suppress alarm-checking, normal and abnormal state descriptors, log number to trigger on alarm state, and alarm groups.

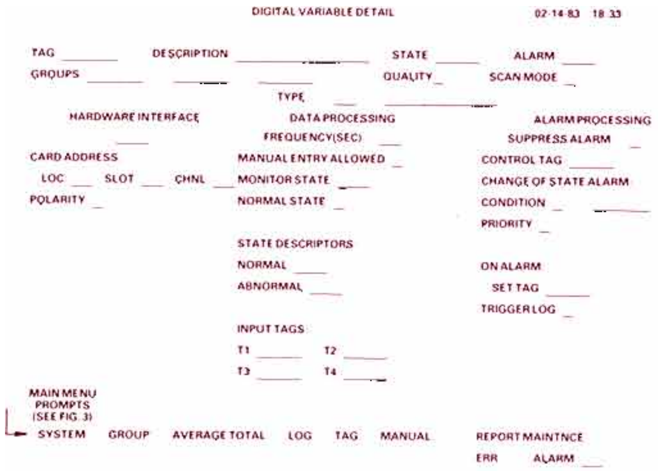


Figure 6. Digital variables display (via F6, Tag Details display, and Digital Variables).

Other data fields allow users to assign a *tag* (point identifier) to the channel, link together a number of tags as a group, and specify alarm conditions.

Data regarding a channel can be called up in various ways. For example, the user selects the second function (F2) to call up Group Summary, then selects the Group Summary menu and types in the desired Group. A user can also enter the tag designation to call up information regarding a specific channel. Figure 7 shows a display format that allows operators to check the status of a group. The group display provides current information (tag, description,

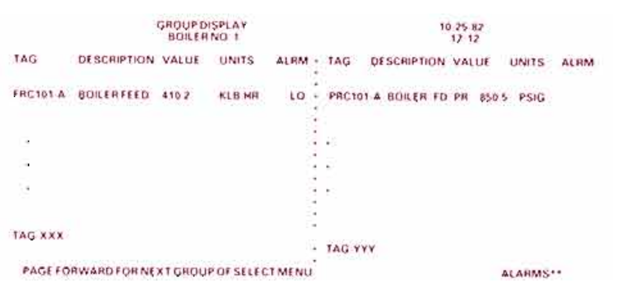


Figure 7. Simplified example of current-value group display.

value of reading, units, and alarm status) for each input assigned to that group.

A single group might, for instance, be the combination of inputs monitoring a single piece of equipment, such as a turbine. The two tags shown here as part of a group display are an input channel monitoring a boiler feed in klb/hr and boiler feed pressure in psig. The first tag has an alarm condition indicating that the feed value is below a limit set for that input channel. The operator can respond to this condition by going to another screen and altering the limit or generating a digital or analog output to alter the boiler feed.

When it scans the input channels, the system scans both analog and digital inputs. Scanning of process data is based on the input type. The system scans all digital inputs at a user-defined rate of 1, 2, 5, or 10 seconds; it scans individual analog points based on the assignment to a specific class. For analog inputs, a number of algorithms, standard with DAiS or custom-developed, convert analog data to engineering units.

Discrete inputs are processed on change of state and at system initialization. Each discrete input is specified by the user to have both a normal and abnormal state and changes can trigger an alarm or event. Analog inputs also trigger alarms or events when the signal exceeds user-defined high or low limit or rate-of-change alarm limit.

When the system detects an alarm or event, a message is logged and displayed. Alarms are initially presented as "unacknowledged." Figure 8 shows the Unacknowledged Alarm Summary screen. Operators call up this screen to see a historical display of past alarm conditions. Calling up and viewing these alarms causes their status to be changed to "acknowledged".

The display details the time of the alarm. The tag and description informs the operator as to what and where the alarm is. The Q is the quality of the the channel and relates to the last time it was scanned or whether there is an error such as an open circuit. Also listed are the value (analog, in engineering units) or status (open, close), engineering units for values, value of limit violated and limit violated code. (e.g. HIGH, HIHI, LOW, LOLO, ROC, etc). This compact display of all key information on a single screen allows operators to track and respond to alarm conditions. ▶

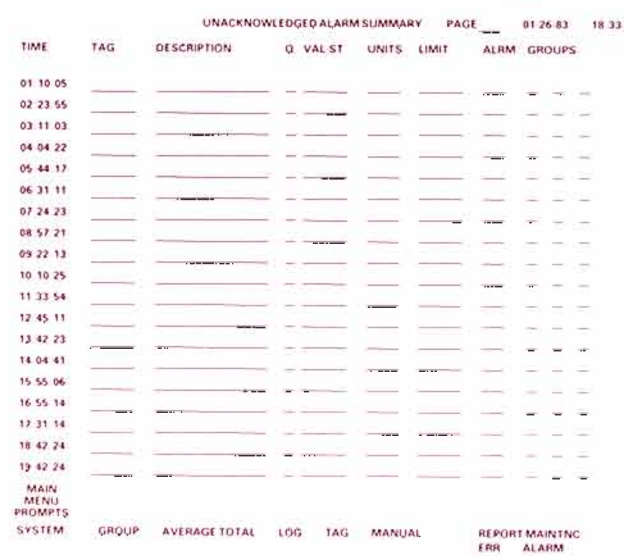


Figure 8. Unacknowledged Alarm Summary format showing times. Other details omitted for clarity.

BOOLEANS, 8-COLOR PLOTTER, 10MB WINCHESTER, BUFFERS

MACBASIC 3 Enhancements for MACSYM Add Logical Operations, Color Hardcopy, 10-Megabyte Mass Storage, and Multiple Buffers for >64K Program Memory

by Richard Quinn

MACBASIC 3 continues to evolve as the most advanced measurement and control language available. The newest version, Revision 2.0, adds Boolean operators for logic-intensive control applications, plotter support for hard-copy color graphics, 10-megabyte Winchester support, and more-flexible memory management for large memory-intensive programs.*

BOOLEAN OPERATORS

Boolean operators, which perform logical operations on numeric values, are usually used to connect two or more binary (two-valued) relations together and return a TRUE or FALSE decision to be used in making a choice between two alternatives. Included are NOT (complement), AND (conjunction), OR (disjunction), XOR (exclusive OR), IMP (implication), and EQV (equivalence). Table 1 lists MACBASIC's Boolean operators and their definitions. Boolean operators can greatly simplify program structure, making it easier to read while taking up less code space than multiple IF . . . THEN statements.

X	0011
Y	0101
NOT X	1100
X AND Y	0001
X OR Y	0111
X XOR Y	0110
X EQV Y	1001
X IMP Y	0010

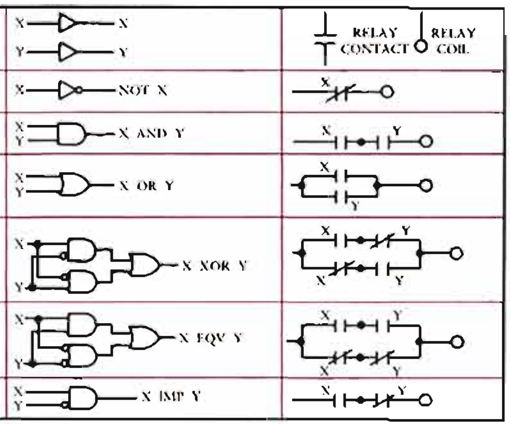


Table 1. MACBASIC Boolean Operators.

Here are two simple examples of how Boolean operators can be used in MACBASIC (−1 is TRUE, ON, CLOSED; 0 is FALSE, OFF, OPEN); the suffix, ('), indicates an integer operand):

```
IF VALVE1' = -1 ANDVALVE2' = -1 AND
   VALVE3' = -1 THEN 100

IF TEMP1 <LOWERLIMIT OR TEMP2>
   UPPERLIMIT THEN ALARM1' = -1
```

In these cases, the operands represent true or false values. However, they actually consist of 16-bit integers; logical comparisons are performed on the corresponding bits of integer operands. For example, the base-10 numbers, 63 and 16, represented by binary 111111 and 010000 may be AND-ed, by AND-ing the individual bits, to give 010000 (16). This would be stated as

A' = 63 AND 16.

with the result that A' = 16.

In the statement,

B' = 21 OR 12

21 and 12 in binary are 10101 and 01100. When or-ed the result is 11101, therefore B' = 29.

Applications

Besides such purely computational chores as implementation of more-compact programs, and making decisions, Boolean functions have direct application to the simulation, analysis, and synthesis of programmable-logic controllers. A simple example is the ladder diagram of Figure 1, in which a set of relay coils, CR4, CR5, and CR6, are energized when the series circuit is completed. For example, CR4 is energized when CR3 and either CR1 or CR2 are closed.

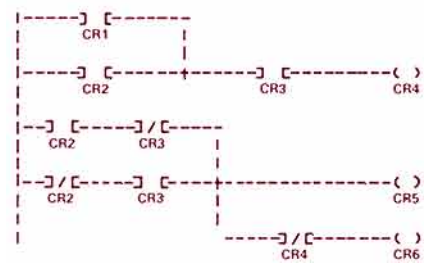


Figure 1. Switching ladder diagram.

In such an application, the user would read-in the state of contacts CR1 through CR4, using MACBASIC's DIN (digital input) command. The state of these contacts would be assigned to integer variables CR1', CR4'. 0 would mean an open contact, −1 a closed contact. The following three lines of MACBASIC 3 code would simulate the action of the relays in Figure 1:

```
10 CR4' = (CR1' OR CR2') AND CR3'
20 CR5' = CR2' XOR CR3'
30 CR6' = (CR2' XOR CR3') AND NOT (CR4')
```

The state of the relay output coils, CR4' through CR6', would be set by the DOT (digital output) statement.

8-COLOR PLOTTER FOR GRAPHICS HARD COPY

The PLT 03 8-color plotter complements the high-resolution graphic capabilities of the MACSYM 150/350 Measurement and Control System, making it possible to obtain hard copies of sophisticated color graphics displays

The PLT 03 Plotter features an 8-pen stall, a 17" × 11" plotter bed, and a pen-writing speed of 15 inches per second. The plotter soft-

* For complete information on these enhancements, get in touch with the nearest Analog Devices Sales Office or phone our direct line, (617) 326-6666.

ware driver has been integrated into MACBASIC 3 so that the programmer does not need to learn a new graphics language in order to use the plotter. The graphics commands for the plotter are identical in function to those used with the CRT screen. The commands are summarized in Table 2.

BOX	Draws a rectangle, solid or outline
CIRCLE	Draws a circle (or arc) solid or outline
COLOR	Change the pen color
FONT	Changes character size
GRAPHICS	Assigns the plotter to an RS-232 port
HAXIS	Draws a horizontal axis with tick marks
HAXISP	Draws a horizontal positive axis with tick marks
HOME	Moves pens to upper left corner of current view
HPRINT	Prints a character string horizontally
JOY	Inputs digitized position information from plotter
LABEL	Prints a character string at a specified angle
LTYPE	Makes the current line style dotted, dashed, solid, etc.
MOVE	Moves the pen but does not draw a line
PAGE	Changes the current plotter/CRT page
PLOT	Draws a line
VAXIS	Draws a vertical axis with tick marks
VAXISP	Draws a vertical positive axis with tick marks
VIEW	Establishes current plotting area
VPRINT	Prints a character string vertically
WINDOW	Defines world coordinate system for graphics view

Table 2. Plotter graphics commands.

The user directs the graphics commands to either the CRT screen or the plotter by using the PAGE command. Pages 0 through 4 are reserved for MACSYM 150/350 text and graphics displays, while pages 5 and 6 are reserved for plotter output. As an example, the following program samples an analog output and displays the output on the CRT screen.

```
10 PAGE 1                      Write all subsequent statements to
                               page 1
20 DISPLAY 1                   Display page 1
30 VIEW 9                      Full-screen (9) view
40 WINDOW 0,100,0,100          Sets up limits of coordinate system
50 MOVE 0,0                    Place cursor at 0,0
60 FOR READING = 1 TO 100      100-point loop
70 VOLTAGE = AIN(1,0)          Define "VOLTAGE" as input Slot 1,
                               Channel 0
80 PLOT READING,VOLTAGE        Plot reading number and measured
                               voltage
90 NEXT READING                Continue loop
```

With a few minor changes, the same program provides inputs to the plotter:

```
5 GRAPHICS 1,5                 Plotter connected to RS-232 port
                               # 1 can be identified as
                               PAGE 5
10 PAGE 5                      Write all subsequent statements
                               to page 5
```

The rest of the statements are identical to those for display.

```
30 VIEW 9
40 WINDOW 0,100,0,100
50 MOVE 0,0
60 FOR READING = 1 TO 100
70 VOLTAGE = AIN(1,0)
80 PLOT READING,VOLTAGE
90 NEXT READING
```

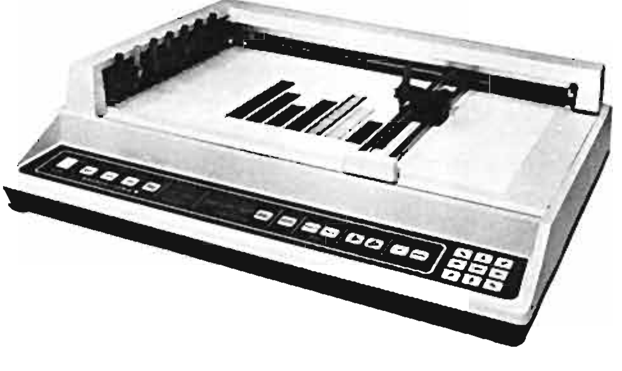


Figure 2. PLT 03 8-Color Plotter.

The plotter/driver requires a DSI-100 RS-232 port for communication between the MACSYM 150/350 and the plotter. The driver is written so that it uses a subset of the Houston Instruments HI-PLOT graphics command language. Other HI-PLOT-compatible plotters should work with the driver, but the user may have to make minor MACBASIC 3 program changes in order to take into account different sizes of plotter beds.

10-MBYTE WINCHESTER MASS STORAGE

The WDS-100 10-Megabyte Winchester provides the MACSYM 150/350 with high-density mass storage for programs requiring storage of large amounts of data. It comprises two major assemblies, the Winchester interface card, which occupies one slot in the Series 100 I/O bus, and a separate subsystem—which houses the Winchester power supply, controller board, and platter. Once formatted, the Winchester platter can store up to 10 megabytes. The controller board will detect "burst" errors up to 22 bits in length. Automatic error correction takes place for burst errors up to 11 bits in length.

The Winchester software has been written transparently so that the Winchester looks to the programmer like a MACSYM 5 ¼" floppy-disk drive, but with a capacity of 10 megabytes instead of 320 kilobytes.

Figure 3. WDS-100 Winchester Mass Storage System

MULTIPLE-PROGRAM BUFFERS

Before Revision 2.0 of MACBASIC 3, a user's program was limited to 64 kilobytes of source code (actual MACBASIC statements) and 64K for object code (compiled code for the 8086 processor). While this capacity was sufficient for many user programs, it fell short for others. Now this limitation has been eliminated. You can now create and utilize additional 64KB object- and source-code buffers. There is no degradation in system performance when multiple buffers are used, and all of the standard multitasking commands apply to them.

GROUND RULES FOR HIGH-SPEED CIRCUITS

Layout and Wiring Are Critical in Video-Converter Circuits How to Keep Interference to a Minimum

by Don Brockman and Arnold Williams

In recent issues, Analog Dialogue *has dealt extensively with topics in shielding and grounding,*[1,2] *emphasizing the techniques needed to protect the integrity and precision of analog signals in the dc and audio-frequency domain from interfering signals, whether at line frequency or at much higher frequencies. To complement those articles, we suggest here the elements of good practice for high-resolution "video-speed" converters, i.e., converters of 10-bit or greater resolution, operating at word rates above 1 MHz.*

Electronics may be frustrating for designers who cross the threshold from low-resolution-low-speed to high-resolution-high-speed designs, or from digital to analog-signal-conditioning circuits. For them, it often seems the "ground rules" have changed.

Experienced designers can readily attest to the difficulty of obtaining consistent grounds. They can relate stories about the ground that wasn't where they thought it was, or the ground that wasn't there at all, despite a conviction that "it has to be." On printed-circuit (p-c) boards, wires and/or runs that seemed to be perfectly good grounds are transformed into inductors or worse in high-speed or high-frequency circuits.

At ADI's Computer Labs Division, where high-speed circuits are its bread-and-butter, applications engineers have found that grounding is the focus of a large percentage of questions from designers making their initial foray into high-speed circuits. In most cases, the designers encountered difficulties as the result of being unaware of—or ignoring—certain basic ground rules.

BASIC PC-CARD RULES

Knowledgeable high-speed circuit designers have learned that *every* square inch of a printed-circuit board which doesn't contain circuits or conducting runs should be ground plane. Violating that simple rule invites disaster. But sometimes, strict adherence to the rule is still no guarantee of success if circuit density is too high; then one must reduce the density and create more "real estate" for the ground plane.

Our applications engineers strongly recommend that all breadboard designs be done on double-sided copper-clad boards. Although this is not a sure cure for ground problems, it improves the designer's chances.

Another basic rule for working with high-speed and/or high-frequency printed-circuit-board designs is to connect analog ground and digital ground together within the PC board. This technique is used, for example in Analog Devices card-level high-speed a/d converters (e.g., MOD-1005, MOD-1020, MOD-1205, CAV-0920, and CAV-1210*). Connecting the two grounds enhances the performance of the converters when they are operated either by themselves or as tightly knit subsystems. However, it can raise some system-level problems, to be discussed below.

Another rule for printed-circuit-board designs containing analog and digital circuitry is to use every available spare pin for making ground connections, and to use those pins to separate the analog and digital signals entering or leaving the board.

Avoid using purely insulating (e.g., "Vector") breadboards and small-diameter hookup wire (e.g., #24) for connections, including supply voltages and grounds. The approach will create ground and noise problems if the circuit is intended to operate at 1 MHz or more (it will probably lead to problems at even slower speeds).

To summarize: Use double-sided copper-clad boards with maximum ground area and heavy, well-located power-supply and ground-return leads. Tie grounds together locally.

GENERAL CIRCUIT PRACTICE

Any subsystem or circuit layout operating at high speeds with both analog and digital signals needs to have those signals physically separated as much as possible to prevent possible crosstalk between the two. Digital signals leaving or entering the layout should use runs that have minimum length. The shorter the digital runs, the lower the likelihood of coupling to the analog circuits.

Analog signals should be routed as far from digital signals as size constraints allow; and the two, ideally should never closely parallel one another's paths. If they must cross, they should do so at right angles to minimize interference. Coaxial cables may be necessary for analog inputs or outputs—a demanding condition mechanically, but sometimes the only solution electrically.

When combining track-and-hold and a/d-converter hybrids or modules on the same board, keep them as close together as is practical. All grounds need to be connected to the single, low-impedance ground plane; and the connections should be made right at the units themselves (another argument for having large amounts of good, solid ground plane available all over the p-c board).

A suggested practical approach for accomplishing this is illustrated in Figure 1, which shows a flow-chart layout, as the preferred method for combining high-speed analog and digital circuits on a p-c board.

If one assumes a 10-volt input range on the 12-bit a/d converter, the least-significant bit (LSB) of the ADC will have a value of 2.5 mV (10 V/4,096). Assume that a single pin of the p-c connector, which is used for ground, has a resistance of 0.05 ohm—and that the p-c card draws a total of 1.5 amperes.

The voltage drop at the ground pin could be 75 millivolts in these circumstances. If only digital logic were used, this voltage drop would be minuscule, hardly worth considering. However, the hypothetical real-world situation being considered here is a mixture of both analog and digital circuits, and the 75 mV can have a significant impact on the subsystem's performance.

[1] Alan Rich, "Understanding Interference-Type Noise," *Analog Dialogue* 16-3, 1982, pages 16-19.
[2] Alan Rich, "Shielding and Guarding," *Analog Dialogue* 17-1, 1983, pages 8-13.
*For technical data on any of these, use the reply card.

In this example, the digital circuits are TTL. Since TTL is a saturated logic, ground currents vary widely, and varying current flowing through the ground often produces noise signals which modulate the ground plane. This noise, created by digital switching, can couple into the analog portion of the circuit and have an important effect on performance, even at low digital levels. For example, if only 10% of the 75-millivolt I-R drop cited here couples into the analog signal, that would represent 3 LSBs.

The result? The circuit intended for operation as a 12-bit system is now reduced to a system of 10 to 11 bits, because of noise masking the 2.5-millivolt level of the desired 12-bit LSB. The recommended solution? Assign multiple pins for ground connections, to reduce the total contact resistance. As Figure 1 shows, those pins are also used to separate the analog and digital signals.

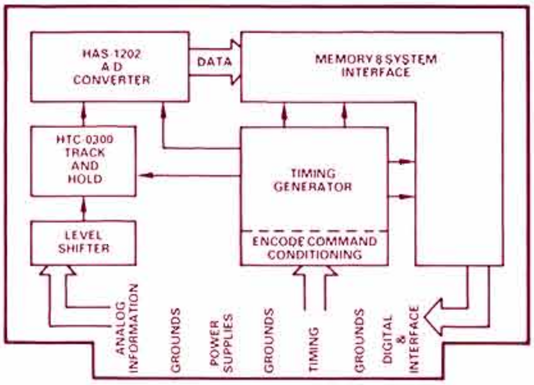


Figure 1. "Flow-Chart" layout for logical separation of functions.

This design approach may seem unnecessarily rigorous and time-consuming but can prove rewarding when the p-c board is installed in its final system location.

Locate the timing circuits near the center of the board (Figure 1) because the timing is at the heart of the circuit, being connected to all of the major circuit components of the board. A central location helps assure minimum paths for the digital signals.

Variations of this theme may not use the exact same components or functions, but the same basic techniques should be applied in any design containing analog and digital circuits. For cards with all connections at one end, avoid configurations which have analog circuits near the p-c connector, and digital signals at the opposite end of the card—or vice versa; either situation will cause analog and digital paths to pass in close proximity to one another.

SYSTEM GROUNDING

Although local ties for analog and digital grounds help the performance of a card, they can cause problems for the system designer using ADCs and DACs. In systems, data converters should be considered as *analog* (not digital) components; the system design must be assigned to experienced and capable *analog* engineers, who are used to defending millivolt signals against interference.

Place ADCs and DACs (like other analog devices) near other parts of the analog section, because: (1) reflections make it hard to transmit analog signals more than a short distance without loss of bandwidth and amplitude; and (2) noise generated by the digital section can couple into the analog through the ground plane or power supplies, or radiate to nearby analog components.

Each card in the system should be returned directly to the power supply common, using heavy wire. Where it is mandatory that a card's analog and digital grounds be separated, each should be separately returned to the power supply; don't connect the two grounds and return a single ground line to the power supply.

POWER SUPPLIES

Besides ground rules, designers of high-speed circuits must also consider the rules about power supplies to obtain best results.

Every power-supply line leading into a high-speed p-c card or data-acquisition circuit must be carefully bypassed to its ground return to prevent noise from entering the card. Ceramic capacitors, ranging in value from 0.01 to 0.1 µF, should be used generously in the layout, mounted as closely as possible to the device or circuit being bypassed; and at least one good-quality tantalum capacitor of 3 to 20 µF should be assigned to each power-supply voltage, mounted as near as possible to the incoming power pins to keep potentially high levels of low-frequency ripple off the card.

To some extent, the p-c's power-supply connector pins can introduce noise problems. If their contact resistance is sufficiently high, and a varying current is flowing, the varying IR drop which results is noise and can be coupled into parts of the card. This caution applies especially to +5-volt supplies used to power TTL systems, but the problem can be alleviated with a variation of the rule about multiple pins for making ground connections. Parallel the I-R drops by also using multiple pins for power connections.

Low-noise, low-ripple temperature-stable linear power supplies are the preferred choices for high-speed circuits. Switching power supplies often seem to meet those criteria, including ripple specifications. *But ripple specs are generally expressed in terms of rms*—and the spikes generated in switchers may often produce hard-to-filter, uncontrollable noise peaks with amplitudes of several-hundred millivolts. Their high-frequency components may be extremely difficult to keep out of the ground system.

If switchers cannot be avoided for high-speed designs, they should be carefully shielded and located as far away from the "action" as possible, and their outputs should be filtered heavily.

ABOUT IC DESIGNS

There is often a difference in implementing designs using high-precision IC circuits vis-à-vis p-c card designs using modules or hybrids. Some ICs are specifically designed to keep analog and digital grounds separated within the device, because they would be unable to perform their functions properly without the separation.

Recognizing this, IC manufacturers are generally very careful in detailing how to obtain optimum performance from their devices. Those details of the application notes frequently instruct the user to connect analog and digital grounds for the device together externally; when they do, the connection needs to be made as closely as possible to the device. In other, much rarer, instances, the characteristics of an individual device—or system—may require some remote connection of the grounds.

The best approach for getting optimum performance from any device is to follow diligently the recommendations of the manufacturer. If the recommendations are missing or vague, ask for them.

Logical signal flow generates logical treatment of ground paths and ground connections—a logical way to prevent potential problems. ▶

New Product Briefs

RMS-DC CONVERTER

Highest-Accuracy, Widest-BW True-RMS Monolithic IC

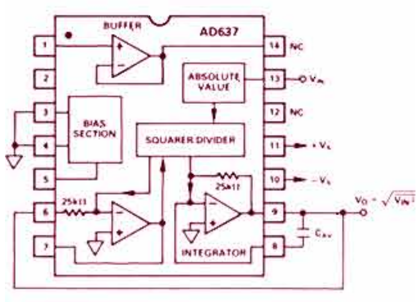
The AD637* is a complete true-rms-to-dc converter on a single monolithic chip. It computes the rms value of complex waveforms for signals up to 8 MHz (−3 dB, 2V rms) and has guaranteed maximum nonlinearity of 0.02% of full scale (10 mV to full-scale signals).

It is laser-wafer-trimmed to a maximum total unadjusted error of ±0.5mV offset and ±0.2% of reading. A single external capacitor sets the low corner-frequency, determining the ripple level, response speed, and settling time. Crest factors of 3 (10% duty cycles) add minuscule (0.1%) error, and crest factors of 10 (1% duty cycles) add only 1% error.

For power-critical applications, such as battery-powered hand-held meters, the AD637—which will operate with supplies from ±3 V to ±18 V—has a 3-state Chip Select, which reduces power-supply drain from 3 mA max to 450 μA max when invoked. The Chip Select also permits the outputs of several AD637s to be wired together to form an active analog multiplexer; the desired channel is selected by pulling its Chip Select high.

The combination of high accuracy, wide bandwidth, and low power drain open up new applications for low-cost rms-to-dc conversion in telecommunications, high-resolution instrumentation, and high-speed testing.

Other useful features include a dB output, a denominator input (for computing vector sums, mean square, and absolute value), and an on-chip buffer amplifier. Prices (hermetic DIP) start at $13.00 in 100s. ▶



*Use the reply card for technical data.

UNPARALLELED INSTRUMENTATION AMPLIFIER

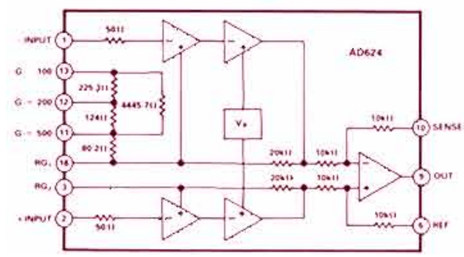
AD624: Lowest Offset Drift and Noise, 16-Bit Linearity IC Has Pin-Programmable Gains, <10ppm/°C Total Gain Tempco

The AD624* Instrumentation Amplifier is a complete high-performance amplifier fabricated on a single monolithic chip. It includes a set of laser-wafer-trimmed tracking resistors to form precise pin- programmable gains of 1, 100, 200, 500, and 1000, as well as 10 other intermediate values, without the use of any external components.

The AD624C's unprecedented performance combines *gain stability* to better than 10 ppm/°C max (G = 100), *offset drifts* of 0.25 μV/°C (input) and 10 μV/°C (output), *peak-to-peak noise* (0.1 to 10 Hz) of 0.2 μV (input) and 10 μV (output), and *nonlinearity* of 10 ppm max (16-bit) for gains from 1 to 200.

Designed for low-level transducer interface applications that demand the ultimate in precision, the AD624's typical applications include measurements of thermocouples and load cells; with its low errors and drift, it meets the stringent requirements of weighing scales.

A member of the same family as the general-purpose AD524*, introduced in *Analog Dialogue* 16-3 (page 3), the AD624 features improved low-level signal handling performance, a doubled range of pin-programmable gains, and a compatible 14-pin DIP pinout (except for gain values). The input protection circuitry of the AD624 will protect the device from differential overvoltages of ±2V under fault conditions. Prices of the AD624AD/BD/CD in 100s are $11.90/$15.55/$23.35. ▶



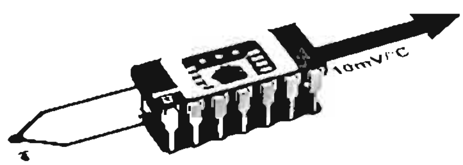
THERMOCOUPLE AMPLIFIER/COMPENSATOR

Cold-Junction-Compensated Amplifier for K Thermocouples Monolithic AD595: Pre-Trimmed for 1°C Calibration Accuracy

Model AD595* is a complete thermocouple signal conditioner on a monolithic chip in a 14-pin hermetic DIP package. Combining an ice-point reference with a precalibrated differential amplifier, it produces a high-level (10 mV/°C) output directly in response to a thermocouple signal. Proper compensation is provided for ambient cold-junction temperatures between 0 and +50°C.

In the open-loop (comparator) mode, it can function as a set-point controller; and—when used without a thermocouple—it will function as an ambient temperature transducer with a low-impedance output voltage directly proportional to degrees Celsius (10 mV/°C).

It is similar to the AD594 compensator for Type J thermocouples, described in *Analog Dialogue* 16-3, page 3, but is precalibrated to work directly with *Type K* thermocouples. Like the AD594, it can be re-



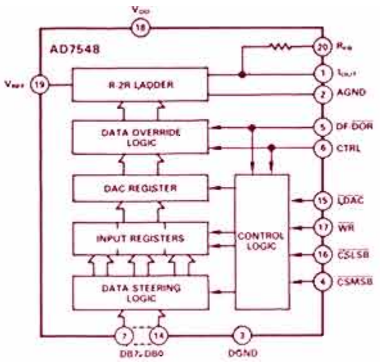
calibrated for other thermocouple types by the addition of 2 or 3 resistors. In addition, both types provide a flexible-format alarm that indicates when either lead of the thermocouple is open-circuited. The alarm can be interfaced to TTL, CMOS, or direct indicator and control functions.

Like the AD594, the AD595 will work over a wide range of power supplies, from a single supply at +5V, to a dual ±15-volt supply for wider temperature ranges, including negative values. The low-power device draws quiescent current of typically 160μA. Prices (100s) start at $9.50. ▶

μP-COMPATIBLE 12-BIT MULTIPLYING DAC

CMOS AD7548 Interfaces to an 8-Bit Bus

Double-Buffered, Left- or Right-Justified Data Format



The AD7548*, a 12-bit CMOS Multiplying d/a converter, can interface directly with 8-bit microprocessors. Its primary application is in high-accuracy analog output systems employing 8-bit data buses.

The AD7548 is a member of a family of 12-bit CMOS multiplying DACs, capable of interfacing with a comprehensive variety of data buses:

- AD7543* Serial input ("1-bit" bus)
- AD7542* 3 4-bit nybbles (4-bit bus)
- AD7548* 4 + 8 bits (8-bit bus)
- AD7545* 12 bits (12 or 16-bit bus)

It is specified for operation on +5V, +12V or +15V supplies, is monotonic over temperature in all grades, and is capable of full 4-quadrant multiplication; all grades have a low 5 ppm/°C max gain tempco.

Double-buffered loading is essential in DACs where the data bus is smaller than the DAC's word length. The feature is also useful in multi-DAC systems; it allows all the DACs to be updated simultaneously—or at will—after they have been loaded. Data may be loaded in either left- or right-justified format, as determined by a control input. Data may be overridden by all-0's or all 1's for system calibration or troubleshooting.

Housed in a compact, 0.3" plastic (JN/KN), Cerdip (AQ/BQ), or hermetic (SD/TD) 20-pin package, the AD7548 saves time, space, and money. Prices (100s) start at $9.48.

LOW-COST μP-COMPATIBLE IC 8-BIT ADC

AD673 Includes Reference, Clock, Comparator, Output Buffers

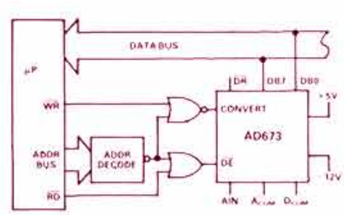
Converts in 20 μs, Has No Missing Codes over Temperature

The AD673* complete monolithic successive-approximation 8-bit a/d converter consists of a DAC, voltage reference, clock, comparator, SAR, and 3-state buffers. No external components are needed for a full-accuracy 8-bit conversion in 20 μs (30 max).

It interfaces to many popular μPs without external buffers or peripheral interface adapters. Output buffer access time is 250 ns max.

Operating on supplies of +5 V and −12 V to −15 V, the AD673 accepts analog inputs of 0 to +10V (unipolar) or −5 V to +5 V (bipolar), as determined by hard-wiring the state of a logic pin. A positive pulse on the CONVERT line initiates the 20 μs conversion cycle. $\overline{\text{DATA READY}}$ indicates completion of the conversion. $\overline{\text{DATA ENABLE}}$ controls the 8-bit 3-state output buffers.

*Use the reply card for technical data.



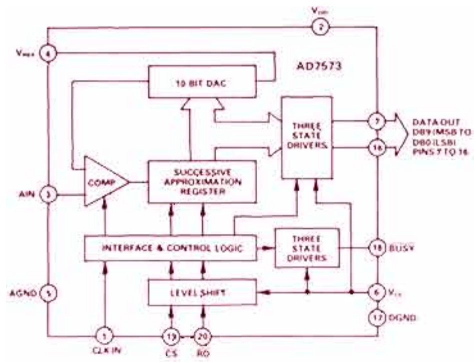
The AD673JN and AD673JD are specified for the 0°C to +70°C temperature range. The AD673SD, which guarantees 8-bit accuracy and no missing codes from −55°C to +125°C, is also available in a MIL-STD-883 version.

The AD673SD is furnished in a 20-pin hermetically sealed ceramic DIP. The AD673J is available in a choice of plastic or ceramic packaging. Prices start at $7.90 (100s, AD673JN - plastic).

10-BIT IC ADC

Low-Cost, μP-Compatible AD7573

No missing codes, T_{min} to T_{max}



The AD7573* is a monolithic CMOS A/D Converter capable of performing 10-bit conversions in 80 microseconds. A successive-approximation device, with an internal clock oscillator, it requires only a single supply, plus a voltage reference.

It is designed for easy microprocessor interfacing using two standard control signals, $\overline{\text{CS}}$ (decoded device address) and $\overline{\text{RD}}$ ($\overline{\text{READ}}$/WRITE control) for Conversion Start and Data Read operations. Conversion results are available via a 10-bit-wide three-state output bus.

When the AD7573 is operated with a single +15-V supply, all logic inputs and outputs are CMOS-compatible. With an added +5V supply, its logic levels are compatible with TTL/CMOS. A ratiometric device, it requires a 5-V reference (e.g., AD584*) for 10-V input levels.

The AD7573 interfaces to μPs quite flexibly. Depending on the connection, it can interface as ROM—reading the results of the last conversion and initiating a new one to be read later (simplest); static RAM—starting a conversion on WRITE and getting the conversion data later on READ (provides best-defined conversion timing); and slow memory—initiating a conversion, and putting the computer in WAIT until the conversion is completed (provides the freshest data).

Available in a variety of performance levels and small 20-pin DIP packages, the AD7573 provides versatility and excellent performance at low cost. Prices in 100s start at $18.50 (AD7573JN).

Worth Reading

BOOK REVIEW

Electrostatic Discharge Control – Successful Methods for Microelectronics Design and Manufacturing, by Tarak N. Bhar and Edward J. McMahon, Hayden Book Company, Inc., Rochelle Park NJ, 1983 (194 pp., hard cover) – *Not* available from Analog Devices.

One of the major barriers to the successful and reliable application of microelectronic devices is an insidious natural phenomenon known to the ancients and immortalized in American legend by Benjamin Franklin: ESD – electrostatic discharge.

As the vacuum-tube and its high-voltage power supplies were increasingly displaced by transistors and integrated circuits, and relegated to specialty areas of electronics, designers breathed a deep sigh of relief; for many years thereafter, they enjoyed the palpable pleasure of working with circuits employing low-voltage power supplies, lethal to neither humans nor semiconductors.

But as CMOS circuits came into increasing use, as feature size of integrated circuits shrunk towards small numbers of microns, and as air conditioning—with its low relative humidity—became an everyday feature, if not a *sine qua non*, of the electronic assembly plant, as well as for the laboratory, sophisticated electronic components were increasingly visited with troubles, from sudden death to permanent degradation. A large share of the problems were traced to ESD, in its many manifestations.

Here now is a book that deals with the subject in a practical way, ranging from the nature and origins of ESD to ways of controlling and dealing with it. The chapter titles are:

- Background
- Mathematics of Static Electricity
- Static Electricity
- Failure Modes and Mechanisms
- Sensitivity of Electronic Parts
- MOS Integrated Circuits
- Effects on Special Devices
- Effects on Bipolar Devices
- Sensitivity of Thick-Film Resistors
- ESD Control Program
- Protective Materials
- Protective Equipment
- Protective Packaging
- Additional Design Considerations
- ESD Susceptibility Testing.

The authors present case histories of ESD damage to microelectronic components to alert readers to the severity of damage from ESD phenomena. They detail the manufacturing processes that generate static electricity, and discuss failure modes and mechanisms in depth—and illustrate them with examples.

Among the useful items of information presented in the book are tables listing typical capacitances of personnel in work situations (192 pF for a seated person), prime sources of static electricity in an electronics work area, typical measured electrostatic voltages (e.g., 18,000 V for a work chair padded with urethane foam in 10-20% RH), constituents of parts susceptible to ESD (constituents, part types, failure mechanisms, and failure indicators), part failure analysis procedures, ESD training curriculum, ESD-protective work-station options, etc. The book has an index, and each chapter is followed by a list of useful references.

Copiously illustrated, with line drawings, circuits, and photographs, this book (or one like it—but we are aware of very few with its brevity, practicality, and completeness) should be an absolute necessity for anyone designing, manufacturing, testing, or working with microelectronic components and circuits—or who manages an organization that does.

NEW PUBLICATION FROM ANALOG DEVICES

Interfacing the Real World to Your Computer is a free 16-page brochure that describes a wide range of methods to interface sensors, transducers, output actuators, and digital I/O to computers. Including key selection criteria, it describes such products as signal conditioners (with and without isolation), two-wire transmitters, remote and local signal-conditioning subsystems, hi-lo alarm-limit subsystems, μC-compatible analog I/O subsystems, and remote, intelligent I/O subsystems. ▶

MORE AUTHORS *(continued from page 2)*

Bill Schweber (page 14) is a Systems Marketing Engineer with ADI's Measurement and Control Division. Before this, he designed microprocessor-based controls for materials-testing equipment. Bill has a BSEE from Columbia University, an MSEE from the University of Massachusetts, and is a Registered Professional Engineer. His hobbies include bicycling, photography, model railroading, and "just taking care of things around the house."

John Sylvan (page 9), Technical Publicity Associate, has been with Analog Devices since his graduation from Colby College in 1980. Working in the Technical Communications Department for the last two years, John has written numerous articles on Analog Devices' computer- and integrated-circuit technologies.

Arnold Williams (page 22) is a Senior Applications Engineer at ADI's Computer Labs Division (formerly Computer Labs, Incorporated), where he worked in Engineering from 1970 to 1975. He then worked as a Contract Engineer and in Component Sales until 1980, when he rejoined CLD as the Division's first Applications Engineer.

Potpourri

An Eclectic Collection of Miscellaneous Items of Timely and Topical Interest. Further Information on Products Mentioned Here May Be Obtained Via the Reply Card.

IN THE LAST ISSUE . . . (Volume 17, Number 2, 1983 -- 28 pages):

NEW-PRODUCTS: DIGITAL MULTIPLIERS & MULTIPLIER/ACCUMULATORS . . . ADI's Digital Signal Processing Division announces the cool-running ADSP-1012 12-Bit X 12-Bit Multiplier and the ADSP-1009 12-Bit X 12-Bit Multiplier/Accumulator, which draw 95% less power than the devices bearing the same number currently popular in the industry, which they replace, pin-for-pin and function-for-function. Cycle times are compatible with 5-MHz clock rates. Because they dissipate only 150 mW at a 6-MHz clock rate, the ADSP-1012 and AD2P-1009 (S and T grades)can be guaranteed to operate over the full military temperature range, -55°C to +125°C ambient - not just case - temperature. Typical prices (100s) are $115 (ADSP-1012K) and $135 (ADSP-1009K).

DATA SHEETS . . . A newly revised and updated uMAC-4000 data sheet is available upon request. It includes information on the family of extender boards, including the new uMAC-4015; and it also includes enhancements to software drivers for HP-86/87 and DEC RT-11 compatibility. Please note this erratum: Correct accuracy figures for R and B type thermocouples (page 10 of earlier data sheets) should be ±3.0°C (±5.4°F) . . . An Ordering Guide and Price List for the uMAC-4000 Series is available upon request.

RELIABILITY NOTE . . . A report is available from your local Component Sales office summarizing the comprehensive qualification testing of ADI's monolithic AD DAC80 low-cost 12-bit DAC. In sum: "The AD DAC80 has exhibited excellent performance and reliability characteristics, in both hermetic and plastic-encapsulated DIP packages, as demonstrated under various accelerated test conditions." The procedure consisted of an Operating Life Test - per MIL-STD-883 Method 1015 - for 1000 hours at +125°C, utilizing both plastic and hermetic parts, and 1000 hours of temperature-humidity-bias operating life test at 85°C/85% R.H. on plastic devices.

AWARD-WINNING SOFTWARE MANUAL . . . The MACBASIC 3 CONCEPTS Manual has won a 1983 Achievement Award in the annual publications competition of the Boston Chapter of the Society for Technical Communications. For further information, get in touch with our nearest MACSYM sales office.

PATENT . . . U. S. Patent 4,395,647 has been granted to William H. Morong, III, for "Half-Wave Signal Isolator with Means for Controlling Flux in the Coupling Transformer."

ANALOG DEVICES

MACSYM 150 hitches directly to sensors and actuators— no add-on hassles or costs.

Here's the one measurement and control computer that goes direct to the source. Our 16-bit MACSYM 150 has it all—including six built-in I/O slots for direct connection to thermocouples and other sensors and actuators at the turn of a screw. So you'll never again have to cope with the built-in headaches of competitive models, which require expensive add-on I/O units before they can even start interfacing with sensors and actuators.

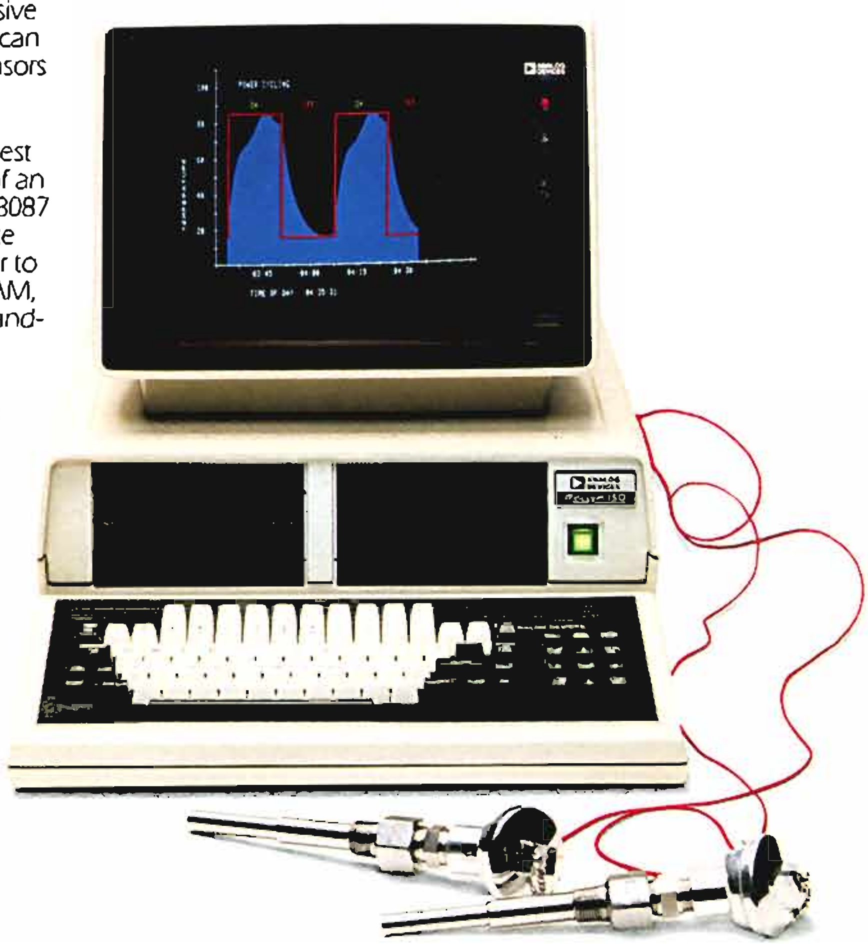
MACSYM 150's everything you need today in a product test computer. The combination of an Intel* 8086 processor plus an 8087 math coprocessor and 1 Mbyte of RAM space give you power to spare. Systems with 128 KB RAM, single floppy disk, and black-and-white graphics start at $6540.

And taking advantage of all these advanced features is as easy as programming in MACBASIC, which is as easy as you can get. With MP/M-86,* MACSYM 150 can run CP/M-86* software, going from measurement and control to personal computing to make your other jobs a breeze.

And when you're ready to expand to even more I/O points, MACSYM 150 readily transforms into a distributed measurement and control system. Just upgrade it with our remote measurement front-end, and the MACSYM 150 becomes the MACSYM 350, with the capacity to interface to over 700 channels. Your MACSYM 350 will handle virtually every input—including thermocouples, strain gages, limit switches, and tachometers—as well as motor and contactor outputs.

For more on the computer that takes the hitches out of product test, write Analog Devices, Three Technology Way, Norwood, MA 02062.

*MP/M-86 and *CP/M-86 are registered trademarks of Digital Research Inc.
*Intel is a registered trademark of Intel Corporation



What makes it the ideal product test computer? There's only one hitch.

Headquarters (617) 329-4700. California (714) 842-1717, (408) 947-0633, Colorado (303) 320-6076, Illinois (312) 653-5000, Maryland (301) 992-1994, Michigan (313) 644-1866; New Jersey (201) 967-1126, New York (716) 634-7970, Ohio (614) 764-8870, (216) 531-9455, Pennsylvania (215) 643-7790, Texas (713) 664-5866, Belgium (32 3) 237 48 03, Denmark (45 2) 845800, France (33 1) 687-34-11, Holland (31 1) 620 51080, Israel (972 052) 28995, Italy (39 2) 6898045, (39 2) 6894924, Japan (81 3) 263-6826, Sweden (46 8) 282740, Switzerland (41 22) 31 57 60, United Kingdom (44 1) 9410466, West Germany (49 89) 570050

Use the reply card for Technical Data.