Subsystems and Data Communications

Chapter I-4

Now that we have seen a number of basic data-acquisition and data-distribution architectures in Chapters I-2 and I-3, it may be worthwhile to pause and consider some forms of system implementation. Areas of contemporary interest include:

1. Proprietary systems, subsystems, and components for interfacing and data communication.
2. Interfacing converters with nearby destinations, such as a microprocessor data bus, using parallel and byte-serial connections.
3. Communicating sensor-based data with distant destinations using serial techniques.

In Chapter I-2, most of the configurations treat parallel data from the converter as an input to a nebulous "buffer" block. This buffer translates the converter's output into machine data format, monitors the status of conversion, initiates conversions, addresses the multiplexer, initiates sample/hold, gain-ranging, etc. This function is achieved by the purchase of proprietary interface products, ranging from IC's and modules to complete systems, and integrating them into the user's overall system.

In this chapter, we shall examine ways in which these forms of interfacing are achieved—first by outlining a number of proprietary systems and subsystems that form a hierarchy, then by a more-detailed consideration of the two basic forms of interfacing—parallel (and byte-serial) and serial.

We must be careful to limit the scope of this discussion, because any one of these topics could itself justify a volume the size of this book for a thorough in-depth treatment of all possible cases. Our method will employ the following techniques: First, this book is oriented primarily towards converters: we shall seek to maintain that focus in this chapter. Second, though we will tend to summarize in somewhat general terms, we will limit much of our discussion to ideas for which concrete embodiments can be found in the Analog Devices product line. This, in turn, permits an abbreviated treatment, with security in the knowledge that the reader who desires greater depth can find substantial amounts of detailed information relating to specific approaches in our published literature (and products available to implement them).

1. SUBSYSTEMS FOR INTERFACING CONVERTERS TO ANALOG AND DIGITAL WORLDS

Briefly summarizing some salient ideas from earlier chapters, data acqusition is the process of transforming electrical voltages or currents, usually transducer outputs, into digital information to be received at some defined destination in a system, for storage, display, processing, or further transmission (Figure 1). The data-acqusition process typically involves these forms of activity:

Analog signal-manipulation

Analog-digital conversion

Digital signal manipulation

Digital control manipulation

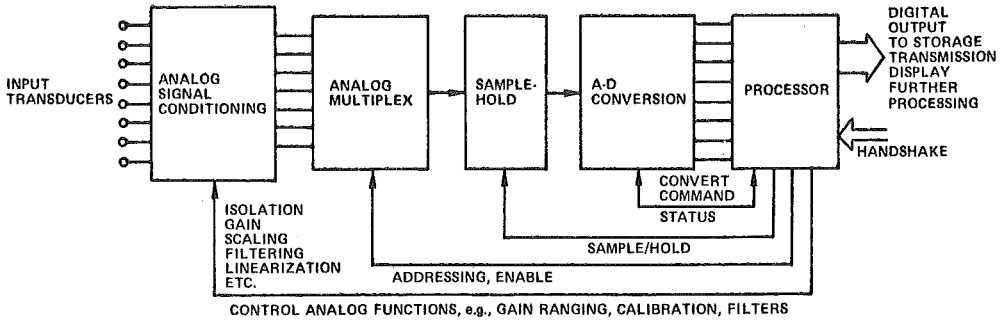


Figure 1. Data-acquisition function.

Analog signal manipulation includes such operations as isolated pre-amplification, gain adjustment, linearization, algebraic functions (perhaps involving other inputs), sample-hold, and analog multiplexing.

An *analog-to-digital converter* produces a parallel or serial digital code that represents the ratio of an analog signal to a reference voltage or current. The digital code is usually—but not always—a binary or binary-coded-decimal number proportional to the ratio.

Digital SIGNAL *manipulation* might involve multiplexing, various arithmetic and logic operations—e.g., magnitude comparisons, algebraic operations, code or format conversions—storage, transmission to a central or intermediate processor, and deriving control signals for either digital handshaking or for operations on the "real-world" portion of the system.

Digital CONTROL *manipulation* includes control of all digital operations, programming of analog functions (switching gains or circuit configurations), selection of analog channels, initiation of conversions, etc., and all of the associated software.

A *data-acquisition system* (for our purposes) is a self-contained subsystem, consisting of the conversion function (which involves at least one a/d converter) and some portion of both the analog and the digital manipulation circuitry. This definition is obviously quite flexible, since it permits a full *functional* range from a simple a/d converter (with a given analog span and digital controls) to an "intelligent" multi-channel measurement-and-control subsystem—and a full *physical* gamut from an integrated-circuit chip to a rack (or a room) full of equipment.

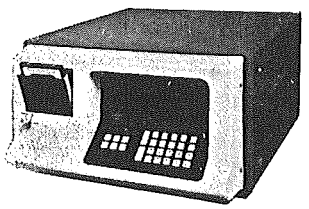
The properties that define a given subsystem are determined by the application; their choice is affected by such factors as resolution, noise levels, system size and complexity, delegation of system tasks, frequency and level of interactions, the physical environment, and (of course) cost—of hardware, of software, of wire, and of system development, prototyping, and manufacture.

When choosing the approach to take in designing a system, one system designer's component or subassembly may be another designer's turnkey system. In general, the design problem involves a classical "make-or-buy" situation. The designer will purchase available components or subsystems that reflect the level of integration that is a best compromise between out-of-pocket cost and the many costs—both overt and hidden— of expending design and manufacturing effort in technological areas that are peripheral to one's primary mission.

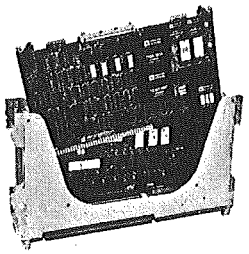
The hierarchy of systems integration is especially deep at Analog Devices, ranging as it does from integrated-circuit converter chips to complete processor-based turnkey quality-control systems for the textile industry. As examples of levels of the hierarchy that are relevant for readers of this book, Table 1 lists some specific products that were available from Analog Devices in January, 1977.*

*Since a complete discussion of their properties and applications is beyond the scope of this book, we will discuss several of these products in the context of their role in the conversion system. Complete information is available, ranging from free data sheets and brochures to complete instruction books and software manuals (at reasonable cost). Consult a nearby Analog Devices field office, or write to the address on the flyleaf of this book.

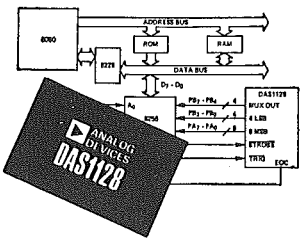
TABLE 1. REPRESENTATIVE EXAMPLES OF DATA-ACQUISITION SYSTEMS, SUBSYSTEMS AND SUBASSEMBLIES MANUFACTURED BY ANALOG DEVICES IN JANUARY, 1977.



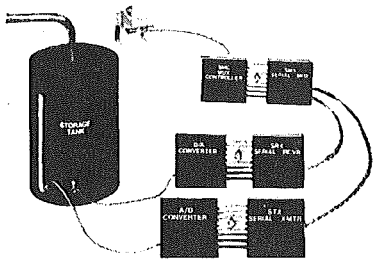
MACSYM ONE Fully Integrated Measurement And Control SYsteM
Macsym One is a processor-based Measurement and Control SYsteM used to automate the measurement, evaluation, and control of real-world events--analog and digital--while interfacing with human operators and other computers. In one compact package, it incorporates a processor, interface modules, an operator's panel, serial data ports and easy-to-use software. It can handle expandably from 8 to 4000 analog or digital input/output signals.



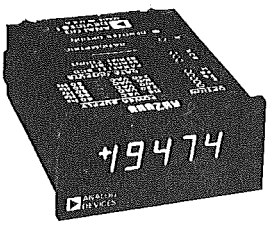
RTI-1200 Analog/Microcomputer Interface Subsystem
The RTI-1200 Real-Time Interface is a complete ready-to-use analog input/output subsystem designed for physical, electrical, and software compatibility with the SBC-80/10 Single-Board Computer--and easy interfacing with any other 8080-based microcomputer having accessible address, data, and control buses. Interfacing as a block of memory locations, it has extraordinary versatility.



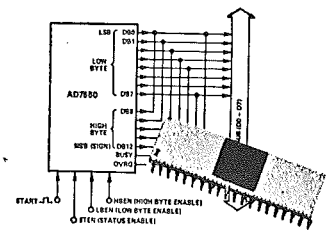
DAS1128 Data-Acquisition Subsystem
The DAS-1128 is a general-purpose 12-bit basic data-acquisition building block that can accept 8 differential or 16 single-ended analog inputs, multiplex them, sample-hold, and convert, with a maximum throughput rate of 35kHz. It can be readily interfaced with microcomputer systems.



SERDEX SERial Data-EXchange Modules
SERDEX modules (and cards) permit easy communication between the parallel-digital-data world of ADC's, DAC's, DPM's and other digital instruments, and the 2-wire ASCII world of teletypewriters and computers. *Transmitters* translate parallel data to asynchronous serial, *Receivers* translate ASCII to parallel, and *Multiplexers* permit operation of a number of modules in a system.



AD2008 Versatile Systems-Oriented Digital Panel Meter
The AD2008 is a 4½-digit line-powered DPM, with a floating opto-isolated front end, designed for making accurate measurements in adverse electrical environments. Besides an easy-to-read display (for the human interface), it has both parallel-BCD and pulse-train options available, and a variety of control facilities, for flexibility in interfacing with local or remote equipment.



AD7550 Single-Chip Microprocessor-Compatible Ratiometric ADC
The AD7550 is a 13-bit analog/digital converter on a single monolithic chip. Its three-state digital data outputs and *select* lines permit direct interfacing with an 8-bit microprocessor data bus in byte-serial format. It can be easily configured for memory-managed I/O. Its "quad-slope" conversion technique provides stabilities to within 1ppm/°C.

We will review briefly the functional repertoire of these products in relation to the conversion interface, starting with the simplest (the AD7550) and working our way up the chain.

THE AD7550 13-BIT CONVERTER ON A CHIP

The AD7550 is a 12-bit-plus-sign 2's-complement ratiometric a/d converter, built on a single monolithic CMOS chip and housed in a 40-pin DIP package. As the block diagram in Figure 2 shows, it is an integrating converter, employing the quad-slope principle (see Chapter II-2), and it requires a reference, three resistors, and a capacitor for normal fixed-reference operation. The data output is available via two sets of three-state latches*, one for the 5 more-significant bits and the other for the 8 less-significant bits. The conversion-status ($\overline{\text{BUSY}}$) and overrange outputs are also three-state; this permits them to be connected to a microprocessor's common data bus and treated as data (as we shall shortly show in Figure 12).

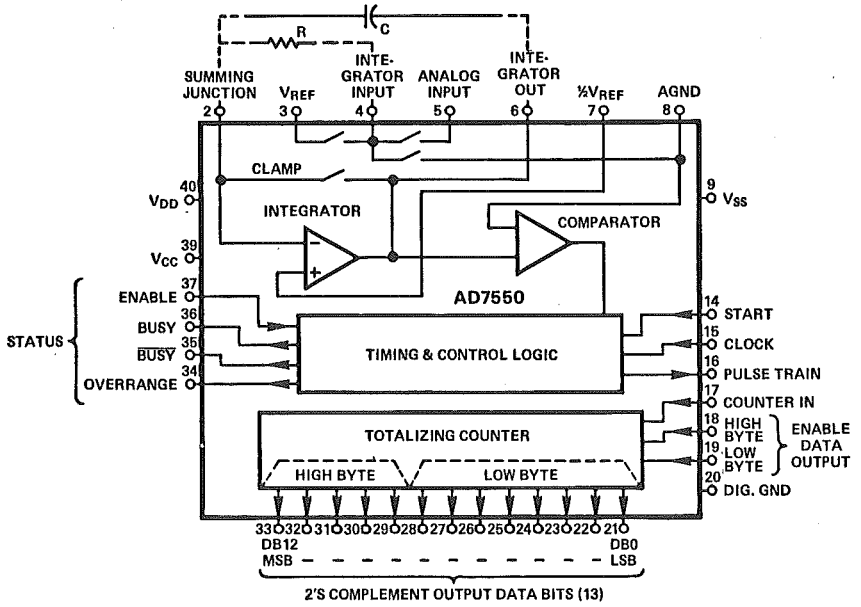


Figure 2. Block diagram of the AD7550 A/D converter.

Thus, the AD7550 can provide either full-parallel 13-bit data output (with both bytes enabled) or byte-serial data output (with both bytes wired to a common bus and enabled at different times). In addition, a pulse-stream output can be used with a remote counter for a kind of serial data transmission.

THE AD2008 SYSTEM-ORIENTED PANEL METER

The AD2008 (Figure 3) is a 4½-digit-plus-sign panel meter designed for flexible system interfacing, as well as for accurate display of input voltage or its ratio to external references. Optional BCD or pulse-train outputs permit either parallel or serial interfacing. Line-powered operation and opto-isolation permit floating measurements to be made in the presence of common-mode voltages as great as 300V rms. In addition to magnitude and sign information, digital outputs also include DATA READY and (for the parallel option) OVERLOAD; control inputs include a 4-mode external TRIGGER-HOLD (2.5 conversions/second, maximum-rate conversion, finish-this-conversion-and-hold, start conversion), and remote display controls (test segments, blank display, blank polarity).

*Three-state latches provide normal "1" or "0" data when enabled; the output floats (open switch) when not enabled. This permits a number of channels to share a common bus (enabled-one-at-a-time).

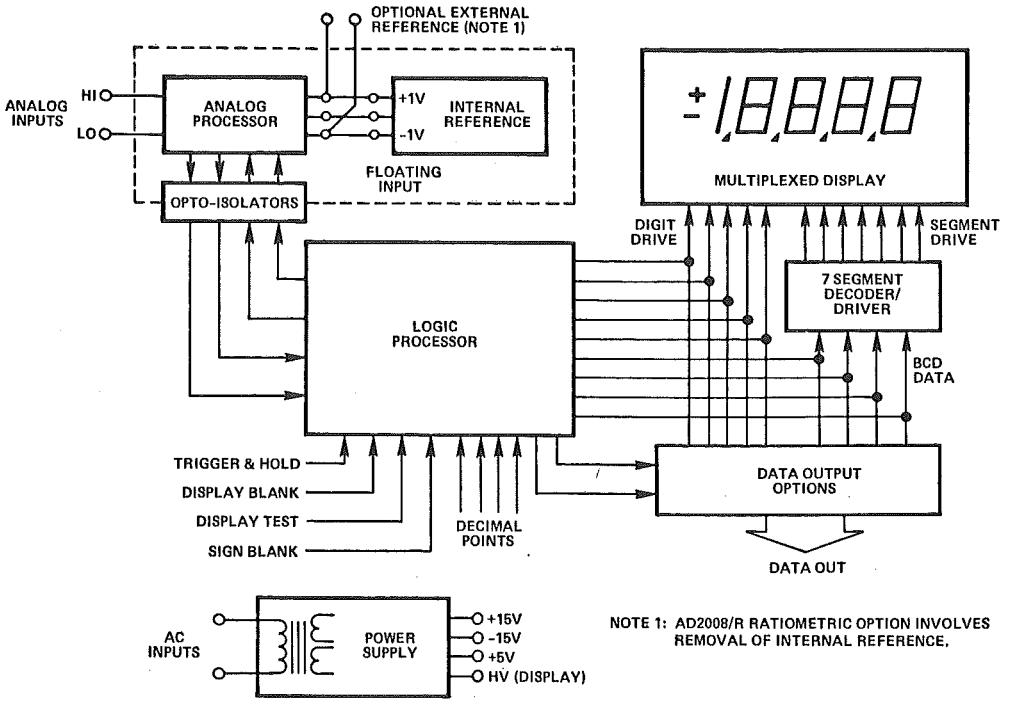


Figure 3. AD2008 block diagram

SERDEX SERIAL DATA-EXCHANGE CARDS

SERDEX (SERial Data EXchange) components are flexible building blocks of systems in which converters and other controlled devices can communicate with remote (up to 3km distant) destinations under remote control, over a single twisted wire-pair, using asynchronous serial ASCII-coded digital information. The three principal components are the STX transmitter, the SRX receiver, and the SMX multiplexer, shown in basic functional block form in Figure 4. A brief example shows how SERDEX works: The asynchronous port, a Teletype machine, transmits the ASCII code for the character "?". Upon receiving it, the STX initiates a conversion. When the conversion has been completed, the STX translates the parallel BCD output of the converter into ASCII format and sends back the numerical magnitude of the converter input for printout, digit by digit. Since the STX ignores alphabetic characters, a printout might look like this:

HOW MANY TONS OF GRAVEL ARE ON THE SCALE? 112

The STX also responds to six characters, in addition to "?", as shown; they can be used to initiate switch closures. For example, the next command might be

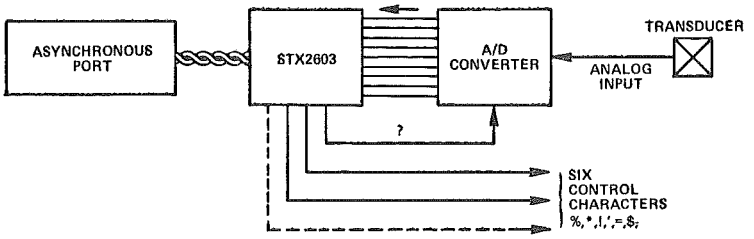
DUMP THE LOAD!

The "!" would initiate a switch closure to carry out the instruction. The SRX receiver communicates with d/a converter, to provide an analog output, in response to a transmitted ASCII number, followed by the symbol "$"; the SRX will also accept six other control characters, like the STX. For systems involving more than one transmitter and/or receiver, the SMX digital multiplexer sorts out the messages, in response to the character "#" and a a numeral. For example,

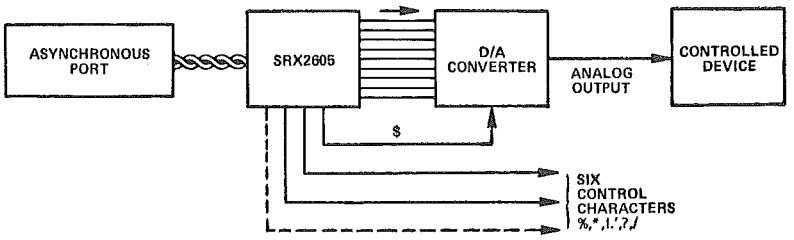
HOW MANY CC OF SOLUTION ARE IN BEAKER #2? 053

causes the multiplexer to select STX number 2; at the same time, the a/d converter associated with the level measurement performs a conversion and sends back its reading, 053.

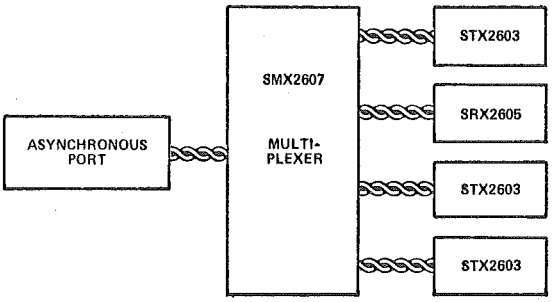
It its not hard to see that pilot measurement-and-control systems of respectable magnitude can be quickly assembled with little software development, using standard Teletype machines or computers of any scope having asynchronous serial data ports. More information on SERDEX and its applications can be found in Section 3 of this chapter.



A. Transmitter (STX2603)



B. Receiver (SRX2605)



C. Multiplexer (SMX2607)

Figure 4. The elements of SERDEX.

DAS1128 DATA–ACQUISITION SUBSYSTEM

Thus far, we have considered two forms of system-oriented converters and a family of system components that are converter-oriented. We now come to a classical data-acquisition subsystem, the DAS1128. It can be seen that its form (Figure 5), resembles the central structure in Figure 1. The DAS1128 comprises an analog-input-signal multiplexer, a sample-hold amplifier, a 12-bit a/d converter, and all of the programming, timing, and control circuitry needed to perform the data-acquisition function and provide a parallel digital output in binary, offset-binary, 1's-complement, or 2's-complement coding.

The user, by choice of external jumpers, can determine whether the device is to accept 8 channels of differential signals, 16 channels of single-ended signals, or 16-channels of "pseudo-differential" input, with a common reference. The multiplexer can be jumpered to scan continuously any number of consecutive channels, from 2 to 16, to scan sequentially on step command, or to select channels as randomly addressed.

The user can also connect the device to establish a variety of full-scale ranges, with either round-number full-scale voltage (e.g., 0 to +10V), or round-number per-bit voltage (LSB = 2.5mV, full scale = 10.24V). Where throughput rate is of prime importance, the device may be connected in an OVERLAP mode, in which the next channel may be selected while a conversion is in process; in addition, the clock frequency may be adjusted for the best conversion-time/accuracy tradeoff, and, finally, the conversion may be short-cycled to the

minimum-acceptable number of bits. The range of throughput rates is from 35,000 12-bit conversions per second for different channels to 200,000 successive 4-bit conversions per second on a single channel.

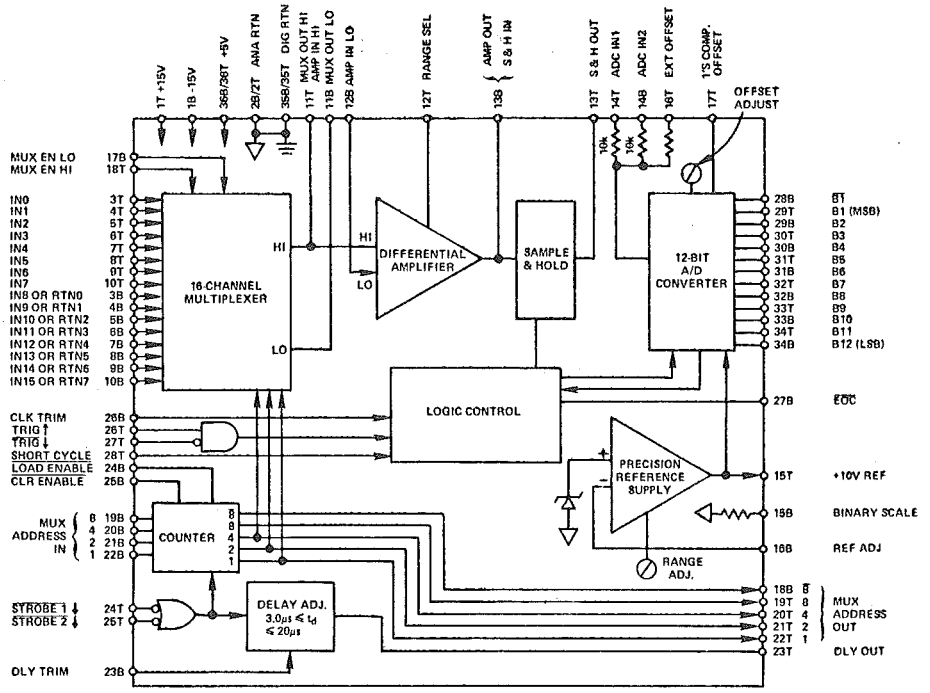


Figure 5. Block diagram of the DAS1128 data-acquisition subsystem.

The DAS1128 can be made to interface with a microcomputer via peripheral interface chips, as you will see in Section 2. However, the burden of interfacing is on the designer of the system. For a subsystem that relieves the system designer of hardware problems relating to the analog-digital-processor interface (and even many of the software considerations), one can turn to the RTI-1200, which is physically, electrically, and software-compatible with a popular single-board microcomputer, the Intel SBC-80/10.

RTI-1200 REAL-TIME INTERFACE

Figure 6 shows the salient electrical features of the RTI-1200 "Real-Time Interface". First of all, the upper central portion of the diagram shows the conventional multiplexer-sample-hold-a/d converter configuration, that is characteristic of data-acquisition systems, and a pair of optional d/a converters, for analog data distribution. At the left are the direct connections to the microcomputer buses: data, control, and address. Other elements of the RTI-1200's optional (*) and inherent hardware features are overvoltage protection at the analog inputs, a programmable-gain amplifier for up to 3 bits of additional dynamic range, 4-20mA current inputs and outputs*, logic drivers for controlling external devices, a dc/dc power converter for deriving all operating voltages from the +5V computer power bus*, a pacer-clock system for relating the computer to real time and for producing repetitive events within the RTI-1200, and a socket for user-installed 1k programmable read-only memory (PROM), which may be used either in relation to tasks involving the RTI-1200 (such as data-linearizing) or simply to add memory capacity within the microcomputer system.

It should be readily apparent that, in hardware alone, the RTI-1200 marshals on one board a fairly complete data-acquisition function. Its software features make that function even more accessible. First, it is designed to appear to the computer as a block of memory locations. This means, for example, that with the appropriate address and a MEMORY WRITE command, an a/d conversion can be initiated or data strobed into one of the d/a converters. Some of the operations that become easy software exercises include:† self-checking, calibration, use of multiple RTI-1200's (with *card select*), improving throughput rates, acquiring

†Techniques for these operations, and more, can be found in the *RTI-1200 User's Guide*, $5, Analog Devices, Inc., P.O. Box 796, Norwood MA 02062.

data on sequentially numbered channels, automatic gain ranging, end-of-conversion interrupts, pacer-triggered interrupts, etc.

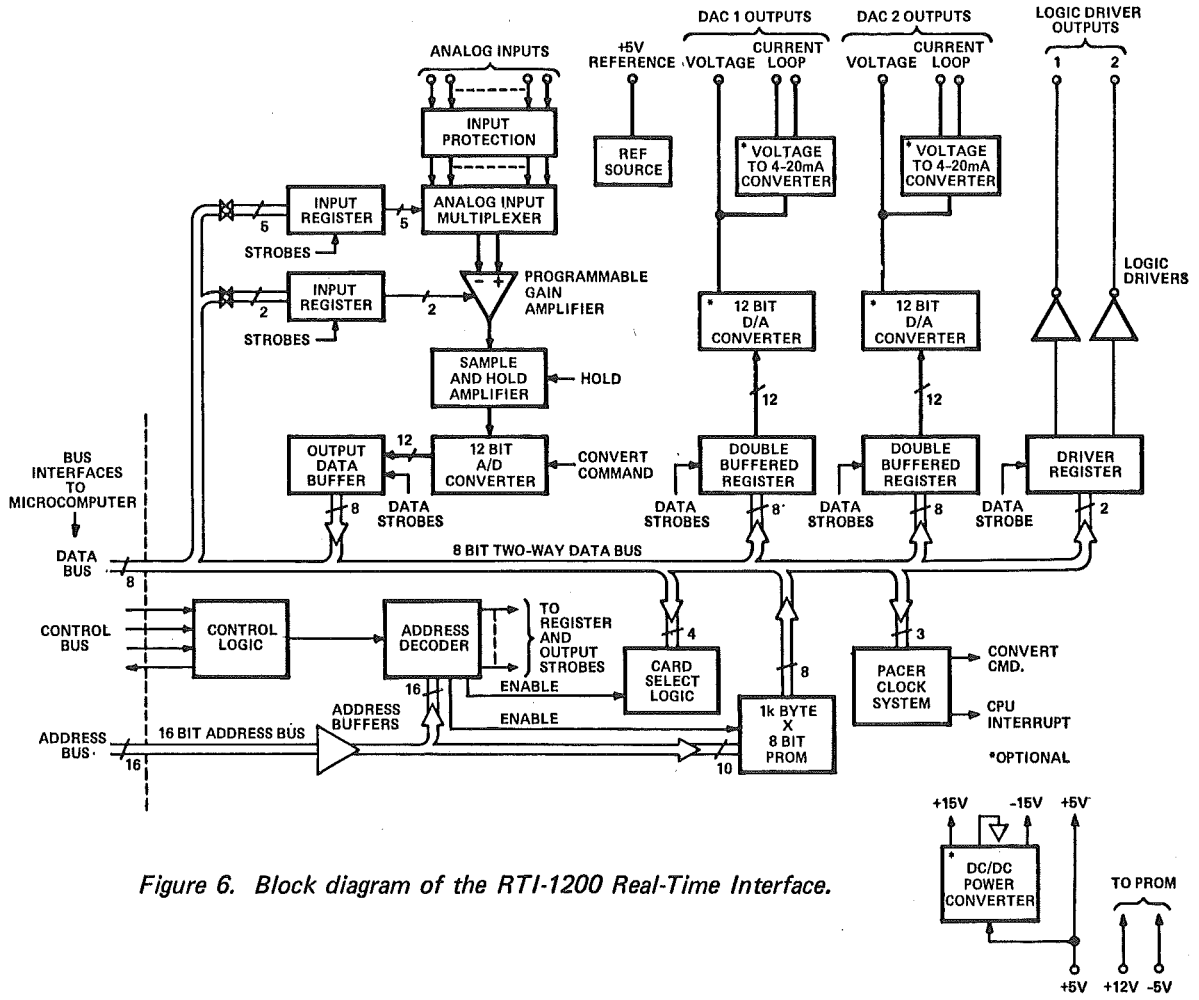


Figure 6. Block diagram of the RTI-1200 Real-Time Interface.

Beyond the RTI-1200, and the limitations of its host computer, the SBC-80/10, one would have to consider a general-purpose integrated Measurement And Control SYsteM, processor-based, that could automate the measurement, evaluation, and control of analog and digital real-world events, while interfacing with human operators, other computers, and peripherals. Such a system is Macsym One. As the block diagram in Figure 7 shows, it incorporates a processor, interface modules, an operator's panel, serial data ports. It also includes easy-to-use software involving a sophisticated operating system.

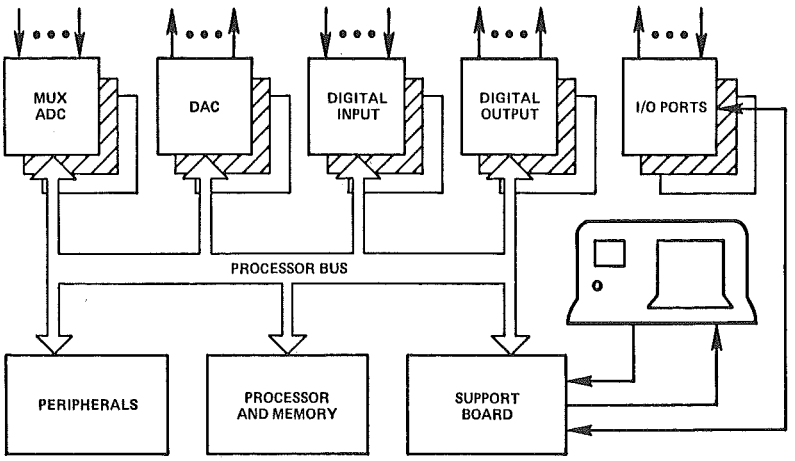


Figure 7. Block diagram of Macsym One system hardware.

MACSYM ONE–A COMPLEAT SYSTEM

Macsym One is a fully integrated hardware/software system, easily programmable in a high-level measurement and control language, designed to simplify the application of computing power to "real-world, real-time" measurement and control problems, particularly when computer knowhow, time, money, or staff is in short supply. It can be used as a programmable instrument, programmable controller, programmable data-logger, programmable calculator, and remote network processor.

Its *hardware* consists of a chassis containing a single-board processor with memory, a system-support board, an operator's control panel, and a backplane capable of accepting 16 additional option boards. Systems are configured by plugging standard off-the-shelf boards (as required for a particular application) into the backplane, which serves as a communications and power bus. Expansion chassis are also available. Input/output (I/O) boards provide functional interfaces to analog and digital real-world process signals. A unique "master-slave" expansion concept permits each family of I/O boards to be expanded 16-fold. The system supports a number of standard peripherals, including floppy disk and cassette tape for program and data storage, Tektronix 4006 storage-tube graphic display, and RS232 ports for remote communication and user interfacing via RS232-compatible peripherals.

The supporting *software* greatly simplifies the task of creating and running programs that match the system to the application. The support packages include a high-level language for program development using simple English/algebraic expressions (MacBASIC), an executive program (XMAC) that facilitates the development and manages the running of user programs or subroutines written in high-level or assembly language, and an assembly-language system–for creating assembly-language object code–that may be linked to and called from the high-level language.

This completes the summary of a gamut of proprietary systems, subsystems, and components for interfacing and data communication. It has been intentionally wide-ranging, and if the reader has found it incomplete and somewhat breathtaking, these are perceptions that were by no means lost to the writer. Let us now consider in somewhat greater detail a common application of a number of the devices described, the interfacing of parallel data.

2. INTERFACING CONVERTERS WITH MICROPROCESSORS, USING PARALLEL CONNECTIONS

Microprocessors, because of their low cost and ready availability, have established what will be the prime area for converter interfacing in the future. It therefore makes sense to indicate how the general principles discussed in earlier chapters can be applied in fairly specific ways to microprocessors.

There is plethora of detailed microprocessor architectures and software systems, differentiated by manufacturer, by "generation," and by degree of integration. It would be futile and well beyond the scope of this chapter to explore even a few of them in detail. Instead, we shall seek to show the elements that most microprocessors have in common and to indicate how some of the conversion and data-acquisition devices mentioned earlier can be interfaced to μP's. Many of the underlying ideas can be found, with considerably more flesh on them, in volumes of "The Bugbook" series*, which contain a wealth of basic information and sets of microcomputer-interfacing experiments, using 8080-based systems. EDN's Microcomputer Systems Reference Issue is also a useful source.†

A *microcomputer* is a full, operational computer system based on a microprocessor chip. A *microprocessor chip* (packaged integrated circuit) is something less than a microcomputer, and the difference between the two is simply a measure of a continually shrinking technological gap. Figure 8 shows a functional diagram of the connections to an 8080 microprocessor. They include a 16-bit unidirectional latched address bus, which is used to address one eight-bit byte out of a possible 65,536 bytes of (external) memory; an 8-bit bidirectional data bus

* A vintage example is THE BUGBOOK III, by David G. Larsen, Peter R. Rony, and Jonathan A. Titus, 1975, published by E & L Instruments, Inc., 61 First Street, Derby, Connecticut 06418.
†At hand at the time of writing was the 1976 issue (Nov. 20, No. 21), EDN Magazine, Cahners Publishing Co., Inc., Boston.

for transferring data to or from the processor; a set of power-supply terminals; a pair of clock terminals; and a set of incoming and outgoing control lines. The processor itself contains an accumulator, a set of registers, and the operational capability of carrying out up to 256 different instructions, coded in 8-bit words. The instruction groups include data-transfer; arithmetic operations; logic operations; branching operations; and stack, I/O, and machine-control operations.

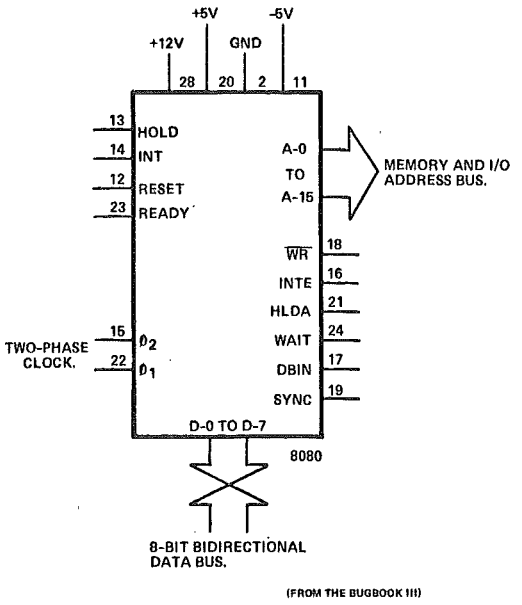


Figure 8. Functional diagram of connections to the 8080 microprocessor.

"Double-precision" operations are inherent: a number of instructions string two 8-bit data bytes together as a 16-bit word. It will be seen that this is a useful feature in dealing with converters.

MICROPROCESSOR INTERFACING, I/O VS. MEMORY

To interface a converter or a data-acquisition system to a microprocessor, a number of requirements must be fulfilled:

> It must be possible to address the converter subsystem, and, if a MUX with random addressing is used, it must be possible to address specific analog channels.
>
> The output of the converter must be transformed to a compatible format and to circuitry compatible with three-state bussing.
>
> Suitable software and control signals must be provided to initiate conversion, determine when conversion is complete, and transfer the data appropriately.

There are two common ways of interfacing peripheral devices with microcomputers (Figure 9). One utilizes an area of memory space, usually restricted in extent, set aside for I/O devices; for example, 256 inputs and 256 outputs may be used with the 8080. These inputs/outputs interface with the accumulator via peripheral interface chips, which handle the addressing, the conversion of (say) 12-bit parallel to two three-state $\leqslant$8-bit bytes, and the controls for I/O read/write.

The other approach, memory-managed interfacing, treats the external I/O devices (and their peripheral chips, if necessary) as memory, using the controls for memory read/write. This approach makes available a much larger number of possible addresses and greater freedom over their disposition, as well as the large number of instructions involving memory, including—in the case of the 8080—the double-precision 16-bit memory load-and-store instruction, which becomes directly available for input/output control.

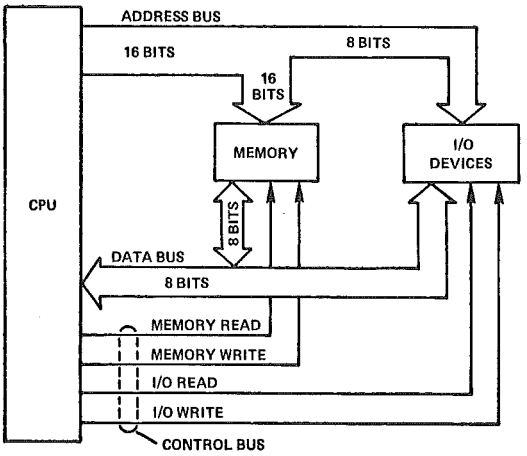


Figure 9. Accumulator (or isolated) I/O vs. memory mapped I/O.

There are some other advantages to memory-managed I/O. Figure 10 shows an interpretation of a microprocessor's principal role (that it has in common with all stored-program digital computers), that of memory controller. Its primary function is to fetch instructions from memory. Time-wasting dislocations caused by the intrusion of the outside world into this satisfying internal game are minimized if the outside world can be made to look like memory. For example, the processor may think it is writing into memory when it is really updating the output voltage of a DAC. Since different microprocessors communicate with memory in essentially the same way, it becomes simpler (for both user and vendor) for a company like Analog Devices to furnish memory-managed I/O devices (especially basic IC chips, such as the AD7550 a/d converter) that will communicate, in concept, with a large majority of available microprocessors, whatever the details of their hardware or software.

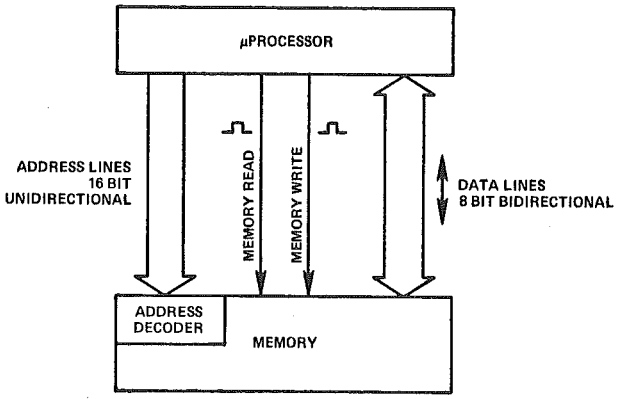


Figure 10. A microprocessor is a memory-controller.

Finally, the use of memory ports for I/O permits the use of "horizontal" memory-managed I/O ("card select" in the RTI-1200). This technique allows the design of highly structured operating-system architectures, which can sidestep the difficulties posed by lack of relative- and index-addressing modes in the 8080.

Concrete examples of some of these concepts can be given in terms of the products mentioned earlier in this chapter.

INTERFACING DAS1128 AND AN 8080

Figure 11 shows how a DAS1128 data-acquisition subsystem might be interfaced with an 8080, using an 8255 peripheral interface chip. A typical sequence of events, slightly simplified, is this:

1. A "setup" byte, addressed to this channel (address decoded), is latched (written) into the 8255. It configures the 8255 as a set of two input ports (8 and 4 bits), which will receive the data from the converter, and one 4-bit output port, which will address

the appropriate MUX channel. (The DAS1128 will have been configured for random addressing.)

2. A MUX-address byte, addressed to this channel (address decoded), appears on the data bus and is latched (written) into the MUX-address input of the DAS1128, causing the multiplexer to switch to the appropriate channel.
3. A conversion command, addressed to this channel (address decoded), is written into the DAS1128's $\overline{\text{STROBE}}$ input and initiates a conversion cycle, starting with sample-hold.
4. At some later time, when the conversion can be expected to have been completed, successive READ pulses, addressed to this channel (addresses decoded), cause data in the 8-bit and 4-bit input bytes of the 8255 to be transferred to the microprocessor. It is also possible (but not shown for the sake of simplicity) for the DAS1128's STATUS (BUSY) line to have triggered an *interrupt* cycle when the converter's output became valid, in order to get fast handling without tying up the processor during the conversion. Interfacing could be either I/O or memory-managed, trading off 8-bit (I/O) vs. 16-bit (memory) addressing for the sake of simpler software (2-byte instructions) and faster machine handling.

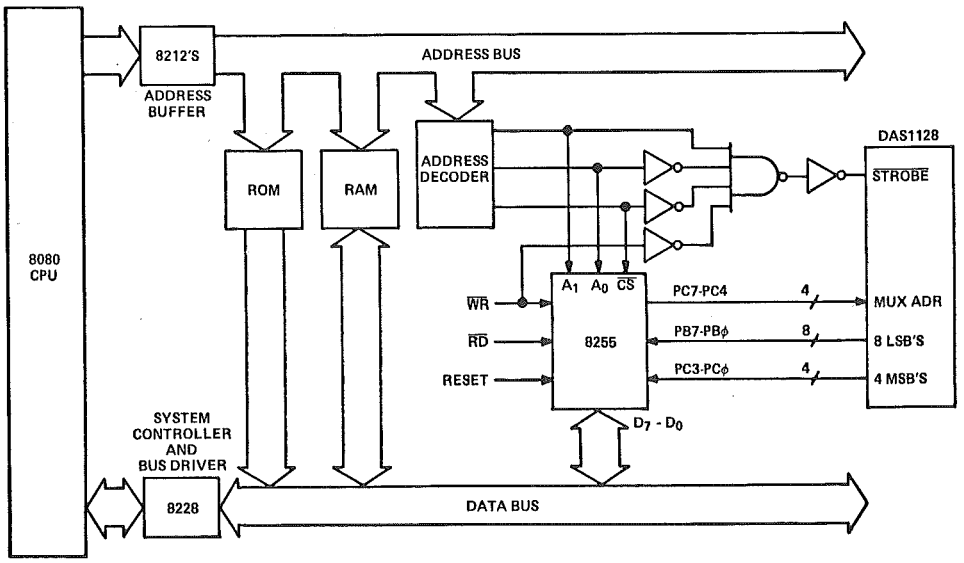


Figure 11. Interfacing a DAS1128 to an 8080 microprocessor.

It is easy to see that essentially the same technique could be used to interface a simple 12-parallel-bit converter, such as the AD572, without the MUX, and with or without a sample/hold. If a separate sample-hold (such as the AD582) were used, an appropriate time delay must be used between the initiating HOLD command and the start of conversion.

INTERFACING AD7550's WITH MICROCOMPUTERS

If the converter has three-state outputs with two separate $\leqslant$8-bit bytes that can be individually enabled, the peripheral-interface chip can be eliminated. The AD7550 is a good example of such a converter. Figure 12 shows how the AD7550 may be connected to an 8-bit data bus. The outputs of the AD7550 are arranged in three groups, each having 3-state outputs and independent *select (enable)* lines. The three groups are:

1. Low 8 bits of the data word.
2. High 5 bits of the data word.
3. The "status" bit (busy signal) and the overrange bit.

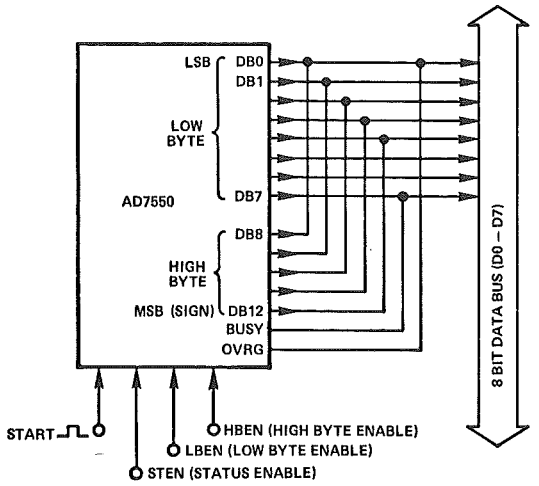


Figure 12. Interfacing the AD7550 to an 8 bit data bus. Coding of AD7550 is 13-bit 2's complement.

When wired as shown, each of the three groups can be selected and enabled (500ns *max* access time) for transmission on the data bus. A fourth signal, START, is required to initiate conversion.

A NOTE ON BIT-LABELLING

Note that the labelling of the bits differs from the normal bit-labelling used in conversion technology. Elsewhere in this text, the most-significant bit is labelled "Bit 1", and the least-significant bit is "Bit n", corresponding to the *fractional* binary weights, 2^{-j}; that is, bit 1 has the weight $2^{-1} = ½$, and the LSB has weight $2^{-n} = 1/2^n$. This corresponds naturally to the way converters work (full-scale digital output generally corresponds to a unity ratio of analog signal-to-reference). Perhaps unfortunately, microprocessor manufacturers have chosen to use *integer*-binary bit notation, in which the most-significant bit is labelled "Bit n-1" and the least-significant bit is "Bit 0", corresponding to the integer weights, 2^j; that is, bit (n-1), the MSB, has the relative weight 2^{n-1}, and bit 0, the LSB, has the weight $2^0 = 1$.

The difference in format, for two-byte words, corresponds to the difference between "left-justified" words (integer point at left), e.g., (12 bits) [1 0 1 1 0 1 1 1][1 1 0 1 X X X X] and "right-justified" words (integer point at right), e.g.,[X X X X 1 0 1 1][0 1 1 1 1 1 0 1]. (For better or worse,) the "first-generation" of μP-compatible devices discussed here are formatted for ease-of-use with popular microprocessors, rather than for either historical continuity or highest analog accuracy in the single-byte 8-bit mode. Hence, in the AD7550, (and the AD7570 successive-approximations ADC, and the AD7522 multiplying DAC), the low byte has the 8 less-significant bits, the high byte has the (five) more-significant bits, and the bits are labeled from DB_{n-1} (the MSB) to DB_0 (LSB).

Figure 13 shows how two AD7550's, each connected as in Figure 12, could interface to the idealized microprocessor of Figure 10. Since the AD7550 was designed as a compromise for both parallel and byte-serial operation, the external address decoding logic is necessary, as shown. The resulting interface is nevertheless quite simple. The 1:8 decoder identifies 8 different instructions, all implemented as MEMORY READ, including the START CONVERSION, which is implemented with a dummy READ command. Following the START command, the microprocessor can examine the status of conversion at will by reading data from the *status* (BUSY) addresses. When this datum indicates that the conversion process has been completed, a single 16-bit MEMORY READ instruction will provide the 13-bit digital value.

If it is desired that the end-of-conversion (EOC) produce an INTERRUPT request, the *status* line could be wired to an interrupt request line and enabled at the start of conversion. This scheme can be extended to embrace any number of ADC's, or indeed any number of combinations of ADC's and DAC's for an analog in/out subsystem.

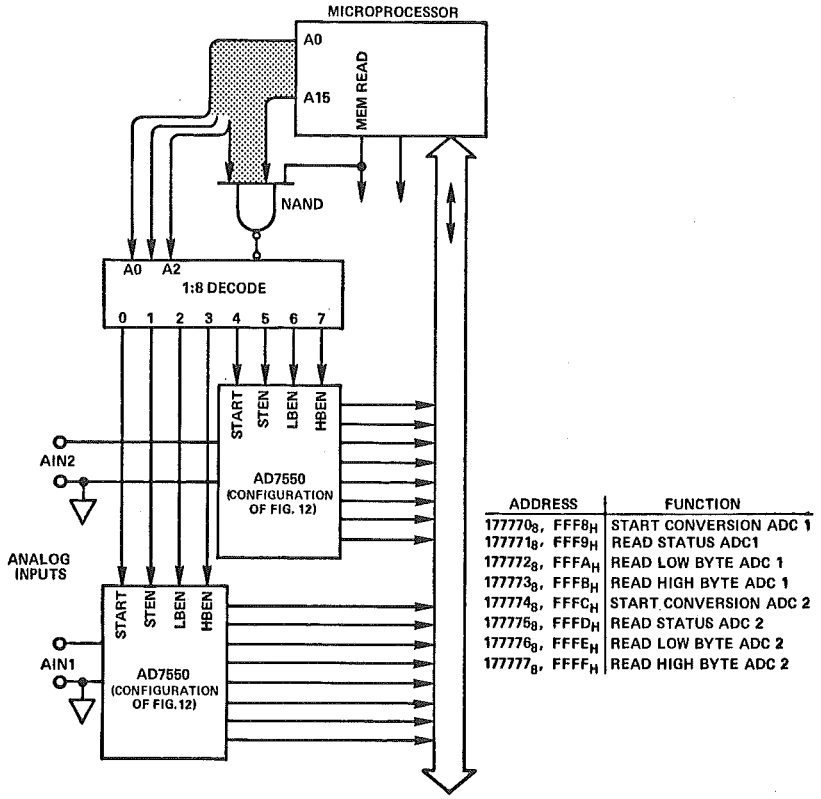


Figure 13. Interfacing multiple AD7550's to a microprocessor.

SOPHISTICATED MEMORY-MANAGED INTERFACING VIA THE RTI-1200

One of the RTI-1200's most important features is the way it interfaces to a microcomputer. It appears as a block of memory locations, using the memory-managed I/O technique; all of the 8080's memory-reference instructions can be used in communicating with the RTI-1200.

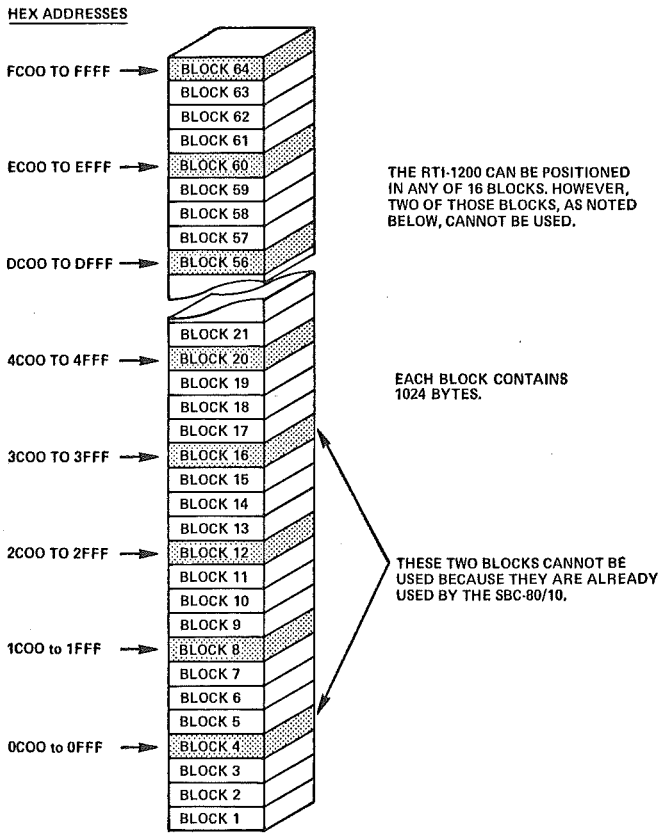


Figure 14. Positioning the RTI-1200 in the SBC-80/10's memory address space.

For example, one of the memory locations used by the RTI-1200 contains the address of the analog-input channel selected by the multiplexer. Stepping from one channel to the next can be accomplished with a single instruction that increments or decrements the contents of a memory location.

A diagram of the RTI-1200's location in memory space is shown in Figure 14. The RTI-1200 occupies a 1k (1024-byte) block, located at any of 14 positions selected by the user. The functions relating to the operation of the RTI-1200 reside in the top 16 locations. the remaining 1008 locations are reserved for use by a 1k programmable read-only memory (PROM) that a user may furnish (and plug in on-board). Such a PROM can be used simply to expand the computer's memory space independently of the RTI-1200's functions. More frequently, it can be used to store subroutines relating to the RIT-1200 (transducer linearizing, for example). This can be especially advantageous if more than one RTI-1200 is used with a single microcomputer.

Figure 15 shows a more-detailed functional view of the memory space utilized by the RTI-1200, and Figure 16, on the next page, is a bit-by-bit memory map. There are a few features of especial interest:

XFF0, the SETUP byte, is instructed as to whether or not the end of conversion will trigger an interrupt. It is also instructed as to which pacer clock will be used, and whether it will be used to trigger conversions or interrupts.

Bytes XFF5 and XFF4, and XFF7 and XFF6 write data into the two (optional) on-board d/a converters, in a "right-justified" 12-bit 2-byte format.

XFF8 provides a/d converter output, with 12-bit accuracy, but truncated to eight bits of resolution.

XFFE and XFFD provide the full 12-bit a/d converter output, in the right-justified 2-byte format.

Two bytes, XFF9 (Gain Select) and XFFA (MUX Address) can be either read or written into. This allows their condition to be determined when gain-ranging or automatically cycled MUX addressing is used.

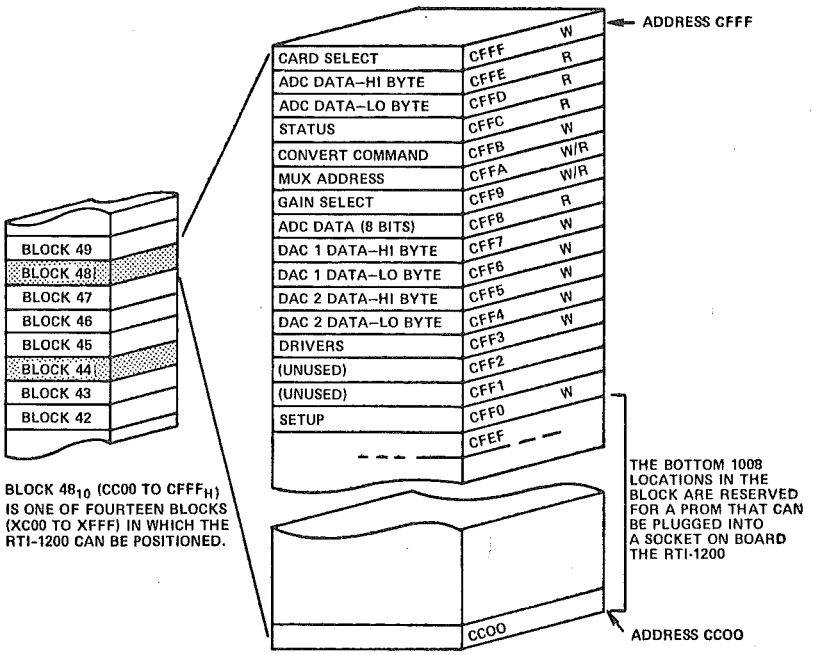


Figure 15. An expanded view of a memory block occupied by the RTI-1200.

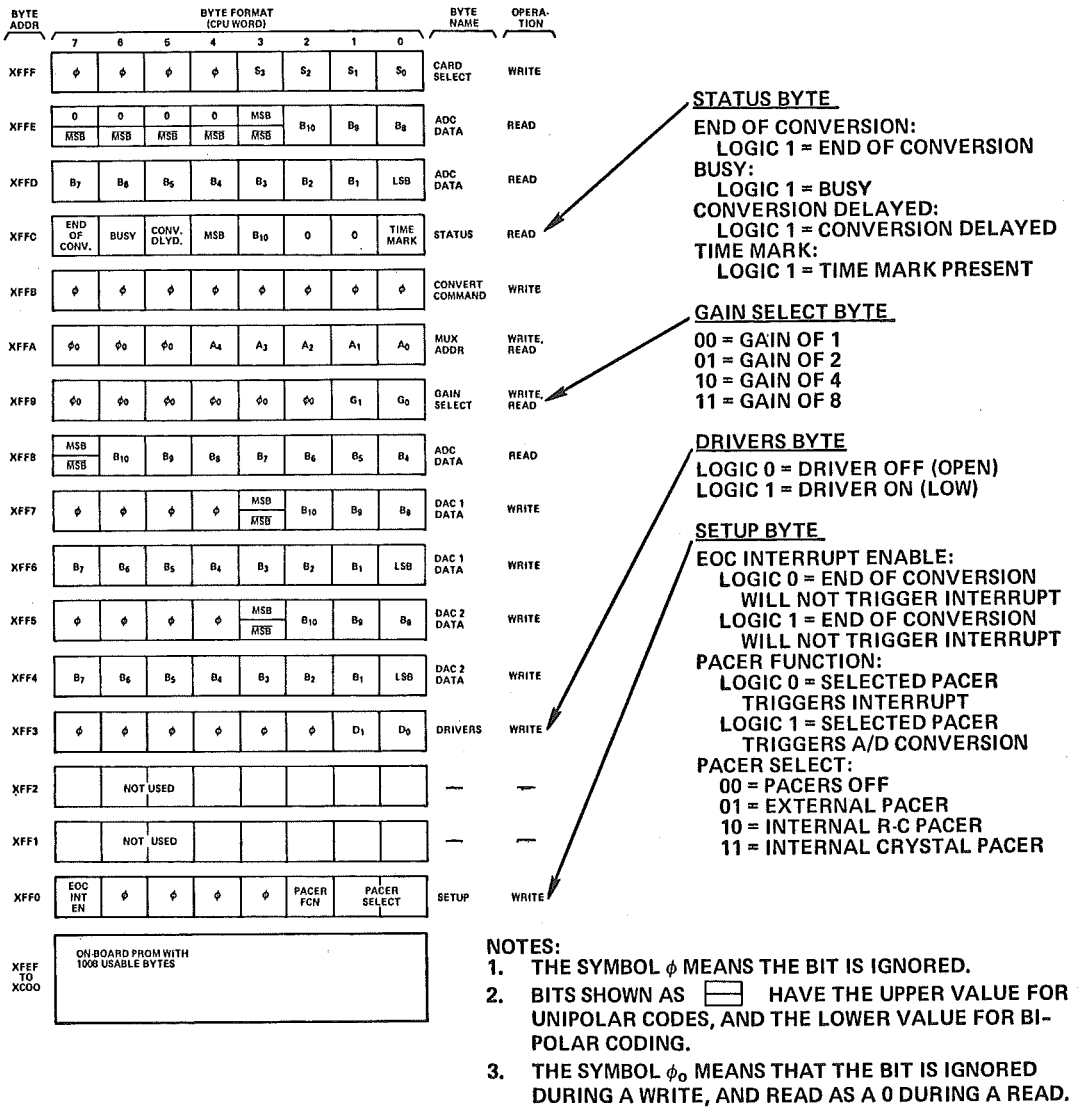


Figure 16. RTI-1200 memory map.

XFFF is a "card-select" (horizontal memory-mapped) byte. This feature permits a number of RTI-1200's to occupy the *same* block of memory interchangeably when addressed (Figure 17).

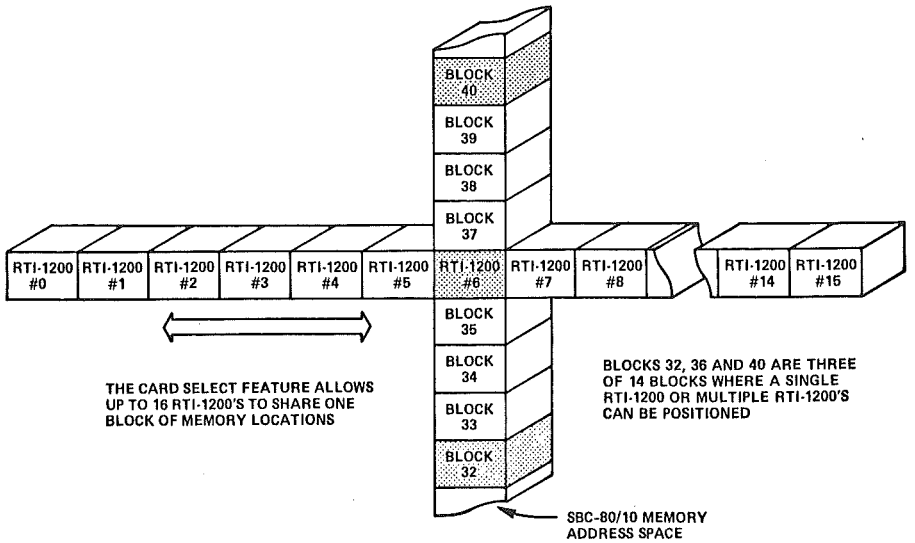


Figure 17. Vertical and horizontal memory-managed I/O as applied to the RTI-1200.

As an illustration of the simplifications afforded by memory-managed interfacing, combined with the advantages of on-board firmware, in the conception and writing of system software, here is the entire (8080 assembly language) program for acquiring a single reading from a single channel:

```
        MVI     A,Ø7H
        STA     MUXADR      ; SELECT CHANNEL Ø7H
        STA     CNVCMD      ; ISSUE CONVERT COMMAND
LOOP:   LDA     STATUS
        RLC
        JNC     LOOP        ; CHECK EOC BIT
        LHLD    ADCLO       ; READ DATA
        HLT
```

This program selects the desired analog channel (07), issues a CONVERT COMMAND, determines when the data is ready, and reads the data. At its conclusion, 12-bit data from the converter will be present in a two-byte register-pair. This program requires only that the desired gain has previously been selected (i.e., the desired data stored in the GAINSEL byte) and that the proper setup data had been written into the SETUP byte. In this case, the correct setup byte would be 00H (00, hexadecimal), which means that the pacers are turned off, and end-of-conversion will not trigger an interrupt.

3. SERIAL INTERFACING

We've shown some of the ways analog information can be handled in its translation to digital and in the interfacing of the digital information with a processor bus. This approach makes the most sense if the source of the analog information is electrically and physically *near* the processor. If, on the other hand, the data must be carried through an electrically noisy environment, and/or over distances greater than a few meters (a task which would become quite expensive for parallel data transmission, because of the high cost of wire and wiring, and the need for more driver power to deal with increased capacitance), immediate conversion to some form of digital or pulse transmission in two-wire serial form is strongly desirable.

V/F CONVERTERS

Perhaps the most-obvious approach is the use of a voltage-to-frequency converter, a device that produces a (usually asynchronous) output train of pulses or square-waves at a frequency proportional to the input voltage or current. V/f converters offer high resolution at low cost, in common with other integrating methods. A V/f converter can continously track the input signal without the need for clock pulses, convert-command signals, or any form of external logic. The direct count of its output pulses, over a time period (Figure 18), can produce a binary or BCD digital number, which represents the average value of the input during the counting period.

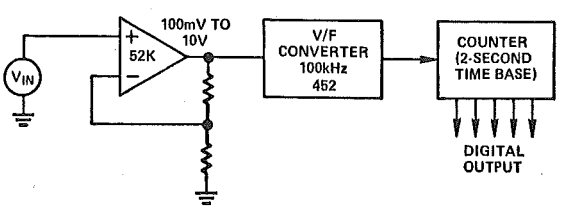


Figure 18. V/F converter used as a nearly 18-bit binary (5½BCD) A/D converter. Resolution is 1 pulse in 200,000, or 0.05% of smallest input signal (or 5ppm of full scale).

The VFC pulses require but a single wire-pair for transmission, unlike parallel converters, which—for n bits—require at least n+1 wires, or synchronous serial converters, which require a form of clock signal. The V/f converter may share a local power source with a transducer and may be optically coupled for high common-mode isolation (Figure 19).

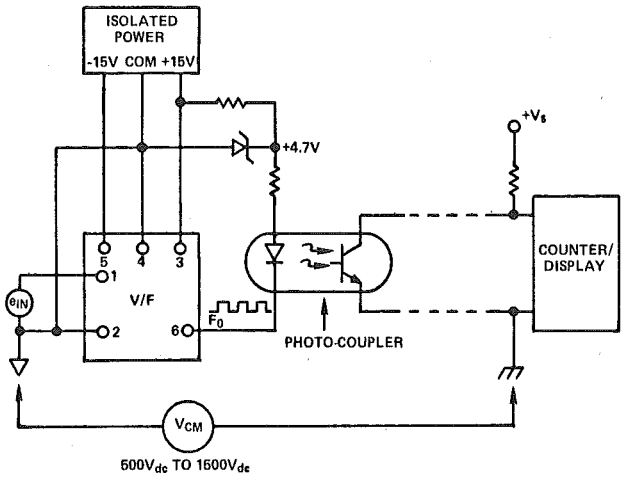
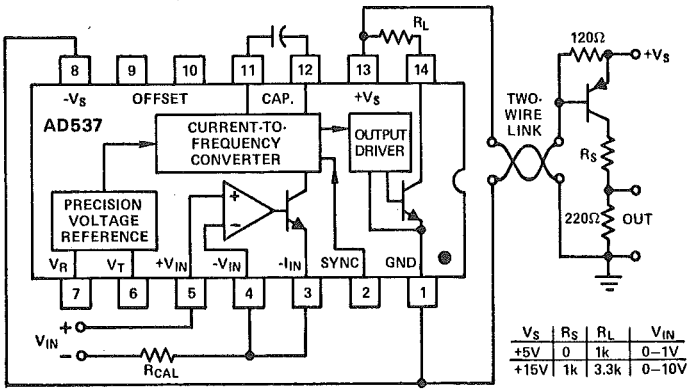


Figure 19. Optically-isolated A-D conversion.

Or a low-power-drain integrated-circuit V/f converter, such as the AD537, can use the two-wire link both to obtain its excitation voltage and to furnish an output-current pulse train, as Figure 20 shows. This avoids the need for local excitation. The current signal is converted to a DTL/TTL or CMOS-compatible signal by the single-transistor termination circuit shown. The excellent supply rejection, high output-drive capability, and square-wave output from the AD537, are all advantageous in this application.

The pulse trains from a number of AD537 VFC's may be multiplexed onto the same counter in random order by connecting their collectors together (sharing a single pullup resistor), and their emitters to the open collectors of a 1:N address decoder. Opening all gates but the one selected will cause its output pulse train to appear at the common collector terminal.

A scheme for building a *synchronous* V/f converter can be found in pages 83-85 of the NONLINEAR CIRCUITS HANDBOOK.* Synchronous VFC's produce pulses only when clocked, with a maximum rate related to the clock frequency. By a suitable interleaving of clock phases in relation to a master clock, synchronous VFC's can be multiplexed onto a single line.



THE AD537 CAN BE USED FOR TRUE TWO-WIRE OPERATION, AS SHOWN HERE. THE FREQUENCY INFORMATION IS TRANSMITTED AS A CURRENT SIGNAL ON THE SUPPLY LINE TO THE DEVICE. THE SIGNAL IS CONVERTED TO A DTL/TTL OR CMOS-COMPATIBLE SIGNAL BY THE SINGLE-TRANSISTOR-TERMINATION CIRCUIT SHOWN. THE EXCELLENT SUPPLY REJECTION, HIGH OUTPUT-DRIVE CAPABILITY AND SQUARE-WAVE OUTPUT FROM THE AD537 ARE ALL ADVANTAGEOUS IN THE APPLICATION.

Figure 20. VFC two-wire operation.

*NONLINEAR CIRCUITS HANDBOOK, Edited by D. H. Sheingold, Analog Devices, Inc., 1974 & 1976, P.O. Box 796 Norwood MA 02062, $5.95.

Although VFC's are low in cost and simple to apply in uncontrolled operation, they have shortcomings that may be serious in data-acquisition systems using 2-wire transmission. Principal among these is the absence of "handshaking," that is, they are not readily controlled, and their format is not very suitable for the interchange of information. Furthermore, the time required for a complete conversion cannot easily be shared for transmitting other information over the line in either direction.

Much more desirable would be a means of transmitting measurements and control signals at will bidirectionally over the two-wire pair and interfacing with Teletypewriters or other human-operated data terminals, as well as minicomputers, microcomputers, etc., as shown in Figure 21. This can be accomplished by the use of a standard coding and serial asynchronous word format (ASCII†), series connection in 20mA current loops, and the ability of any devices in the loop to communicate by stopping and starting the flow of current.

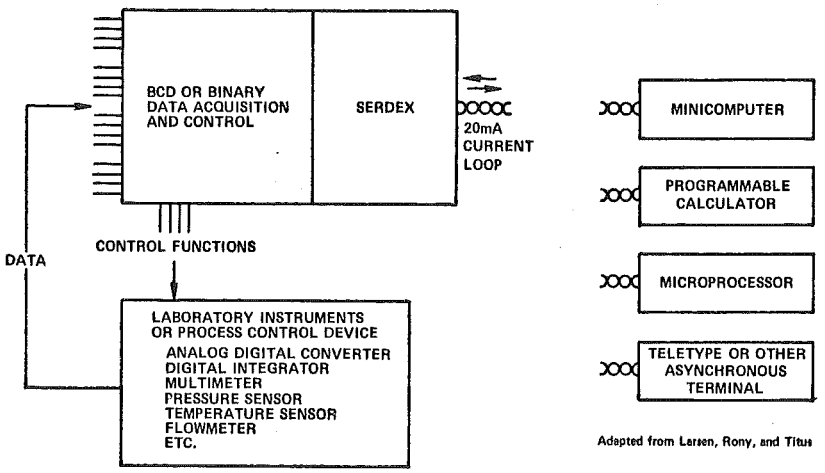


Figure 21. Interfacing processes or measurements and Asynchronous Serial ports.

MORE ON SERDEX

One such approach, as we've shown earlier, is via a system composed of SERDEX transmitters, receivers, and multiplexers (as needed), mounted on cards that include their associated clock circuitry – connected locally to a/d and d/a converters and other local input/output devices. and remotely (via the twisted pair) to an asynchronous serial port.

Figure 22 shows the basic anatomy of a system. A regulated 20mA current source provides

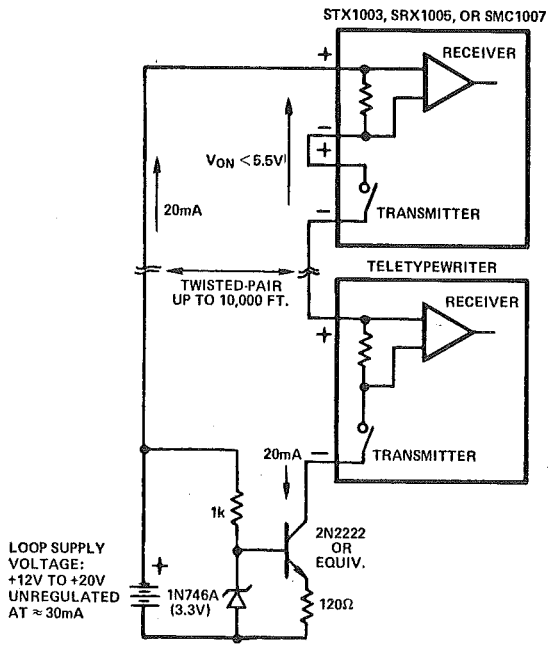


Figure 22. 20mA current loop data transmission.

†American (National) Standards (Institute) Code for Information Interchange.

current for the loop. The current flows through receivers, transmitters, teletype machines, etc., so long as the switches are closed. When any of the devices on the line causes a switch to open, the current stops flowing, and all receivers on the line detect the level change. If a switch opens and closes a number of times, it will transmit a coded message to all enabled receivers; and the code, when decoded, will provide whatever information it represents and will cause whatever subsequent actions are appropriate.

A character in the ASCII format used by SERDEX (Figure 23) consists of a START bit, followed by eight information bits (LSB first) in non-return-to-zero (NRZ) format, and 2 STOP bits. A START signal is given by turning the current off (0). When a START signal is received, the incoming bit stream is sampled at the center of each bit interval, until the character is completed. Since a maximum of 11 bits are involved in a character code, and synchronization is established by the START bit, extremely precise bit-timing is unnecessary. The eighth bit may be used as a parity bit. The STOP bits turn the current on, and it remains on until the next character arrives. Figure 23 shows some examples of ASCII codes used by SERDEX. Since SERDEX transmitters ignore the codes for alphabetic characters, they may be used on printouts to convey background information about the commands or measurements, including units of measurement. All SERDEX cards are pre-programmed (via changeable jumpers) for teletypewriter operation at 110 baud (1.76kHz clock rate). Other data-transfer rates up to 19.2k baud are available by jumpering.

EXAMPLES OF ASCII CHARACTER CODES:

CHARACTER	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1 (LSB)
0	1	0	1	1	0	0	0	0
1	1	0	1	1	0	0	0	1
2	1	0	1	1	0	0	1	0
3	1	0	1	1	0	0	1	1
4	1	0	1	1	0	1	0	0
5	1	0	1	1	0	1	0	1
6	1	0	1	1	0	1	1	0
7	1	0	1	1	0	1	1	1
8	1	0	1	1	1	0	0	0
9	1	0	1	1	1	0	0	1
!	1	0	1	0	0	0	0	1
$	1	0	1	0	0	1	0	0
%	1	0	1	0	0	1	0	1
'	1	0	1	0	0	1	1	1
/	1	0	1	0	1	1	1	1
=	1	0	1	1	1	1	0	1
?	1	0	1	1	1	1	1	1
*	1	0	1	0	1	0	1	0
#	1	0	1	0	0	0	1	1

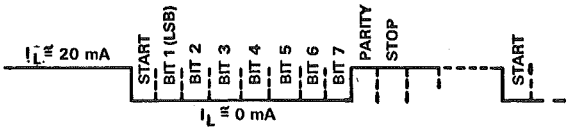


Figure 23. Typical ASCII coding and format.

When an STX serial transmitter receives a control character, an action or a series of actions is initiated. For example, when "%" is received, a pulse is initiated at the appropriate terminal. If a question mark is received (?), an a/d conversion is initiated. When the converter indicates that conversion has ended, the change of state of the *status* line causes the resulting digital data to be transmitted serially back to the control center. As the block diagram in Figure 24 indicates, up to 8 BCD digits can be sent, or up to 24 binary bits (if they are arranged in groups of three for transmission as octal digits). By the use of external shift registers, the STX2603's word output can be expanded to transmit a string of characters of any desired length, including the outputs of additional a/d converters or such ASCII characters as plus- and minus-signs, decimal point, space, carriage return, and alphabetic characters.

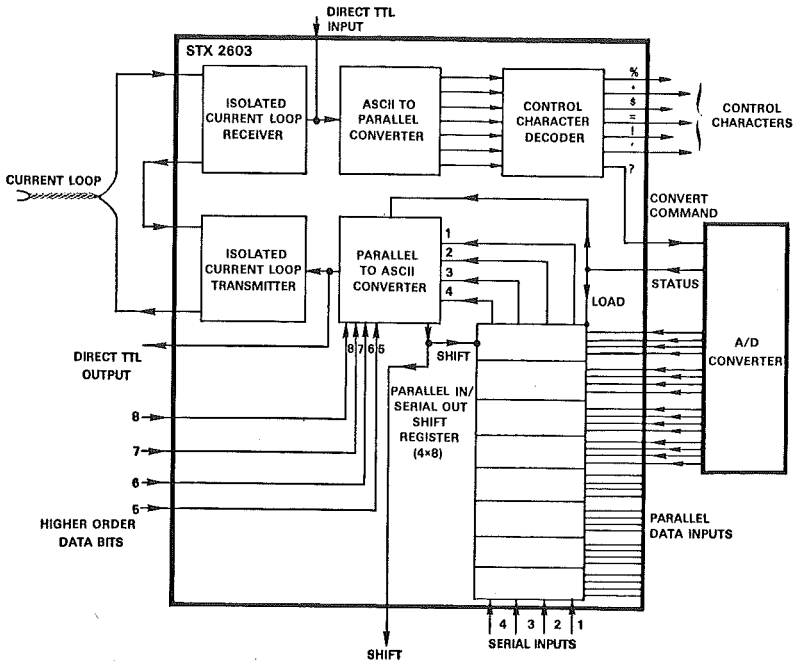


Figure 24. Block diagram of the STX2603 serial transmitter.

The serial output of the STX's parallel-loaded shift register (and any external extensions) is fed to the parallel-to-ASCII converter, where it is transmitted one-character-at-a-time to the teletypewriter of computer.

Figure 25 illustrates a possible application of the STX serial transmitter to control the liquid level in a tank from a teletypewriter. To determine the level in the tank, the operator might type

WHAT IS THE LEVEL IN THE TANK IN CENTIMETERS?

The STX2603 ignores all the alphabetic characters. When it receives the question mark (?), it initiates an a/d conversion and subsequently transmits the data back to the teletypewriter. The STX2603 can recognize and respond independently to 6 other control characters (%, *, $, !, ', and =) that may be useful for controlling system components, e.g., activating a pump, then turning it off later. Thus, a check-and-fill dialogue might proceed as follows:

WHAT IS THE LEVEL IN THE TANK IN CENTIMETERS? 287
TURN THE FILL PUMP ON%
WHAT IS THE LEVEL IN THE TANK? 496
?575
?681
?760
TURN THE FILL PUMP OFF!

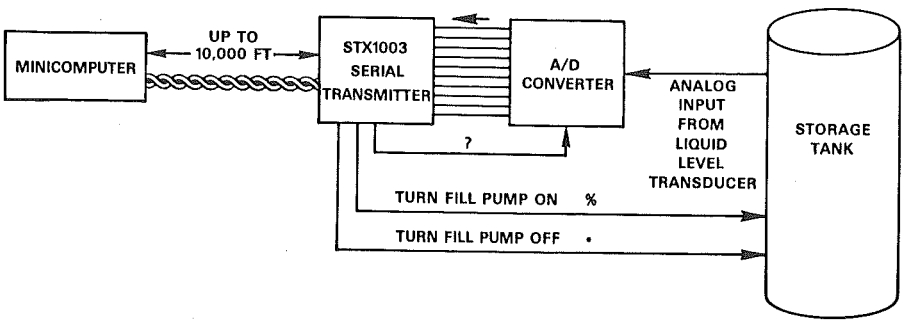


Figure 25. Application of the STX1003 Serial Transmitter.

In this manual system, the operator decides when the tank is sufficiently full and types the turn-off command. Clearly, no significant difficulty is encountered in replacing the teletypewriter with a computer having an asynchronous serial port and a high-level language capability, such as BASIC.

The SRX2605 Serial Receiver (Fig. 26) has a slightly different set of six control characters (%, ?, !, ', /, *) to which it responds by emitting a pulse at the proper pin. If it receives an equal sign (=), the internal register and *ready* flip-flop are cleared, and the data characters that follow enter the shift register. When a terminating dollar-sign is received ($), the *ready* flip-flop is set, data is strobed into the d/a converter (and updates its analog output), and acceptance of additional data is inhibited until the next = arrives. A typical instruction might read

APPLY A NUMBER OF MILLIVOLTS = 3206$

The alphabetic text is ignored. The equal sign (=) designates that data will follow; 3206 is the BCD number that programs the BCD-coded 4-digit d/a converter. and the $, indicating end-of-data, strobes the data into the d/a input register.

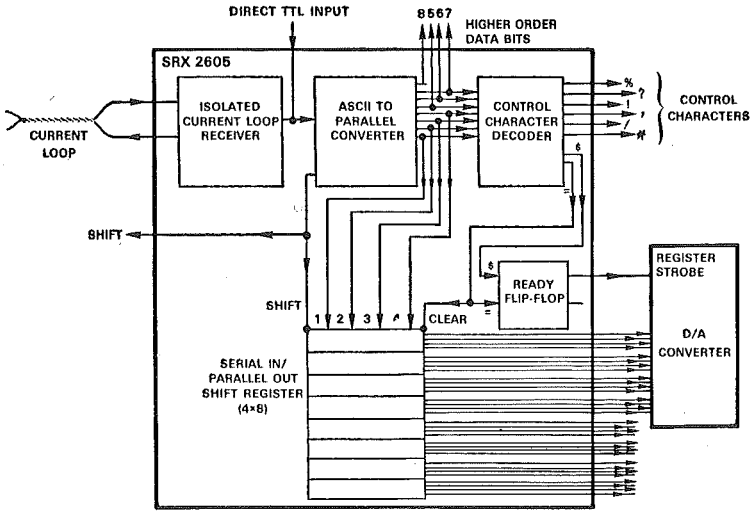


Figure 26. Block diagram of the SRX2605 serial receiver.

The SMX2607 multiplexer (Figure 27) is a bidirectional 8-channel multiplexer for implementing more-elaborate system. For up to 16 channels, an additional multiplexer is used. The first 8 channels (0 to 7) are addressed from the teletypewriter or computer by transmitting the number symbol, #, followed by a digit from 0 (channel 0) through 7; for the second set of 8 channels, # is followed by a letter from P through W, i.e., 8 through 15.

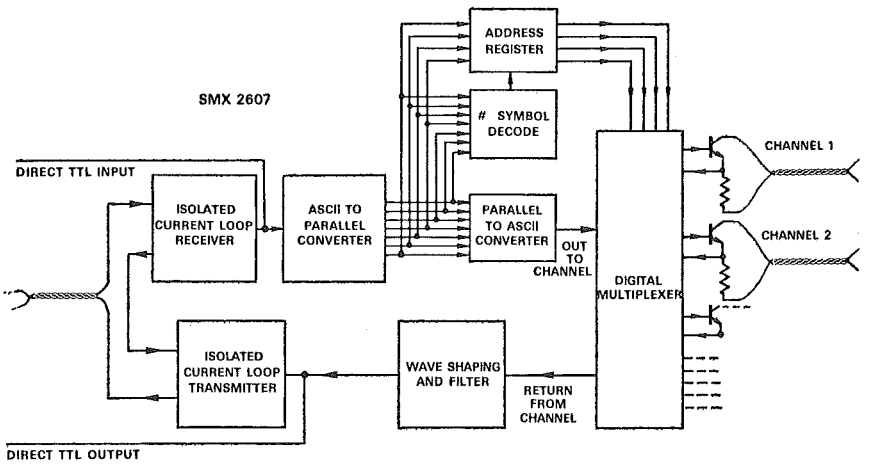


Figure 27. Block diagram of a SMX2607 multiplexer.

Multiple ranking of multiplexers permits communication with an unlimited number of channels. For example, in a dual-rank configuration (8 x 8 = 64 channels), channel 3 of the first rank and channel 6 at the second rank would be addressed if the multiplexers received the characters #36.

Serial data received from the teletypewriter or computer is converted to parallel form, decoded to extract the number symbol and channel address, then converted back to serial data for transmission to the selected channel. Data returning from the remote channel is merely relayed back to the central control station.

Figures 28 through 32 show some additional system applications of SERDEX. Further applications in laboratory instrumentation can be found in AMERICAN LABORATORY, May, 1975. The use of SERDEX with a FIFO (first-in-first-out) serial buffer memory for collecting data in bursts and playing it back at a more-leisurely rate can be found in ANALOG DIALOGUE 8-2. The seminal article describing SERDEX appeared in ANALOG DIALOGUE 7-2. Finally, Figure 33, shows how data converted (and readable) at a remote location by an AD2008/B digital panel meter can be interfaced to a teletypewriter.

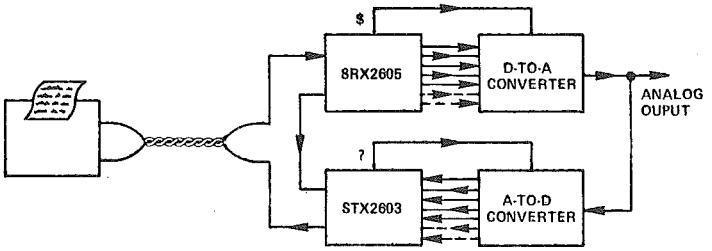


Figure 28. Verification of programmed analog output. An SRX2605 and an STX2603, operating on the same current loop, form a highly-reliable system. When a ? is typed, the D/A converter output is converted to digital and transmitted back to the teletypewriter.

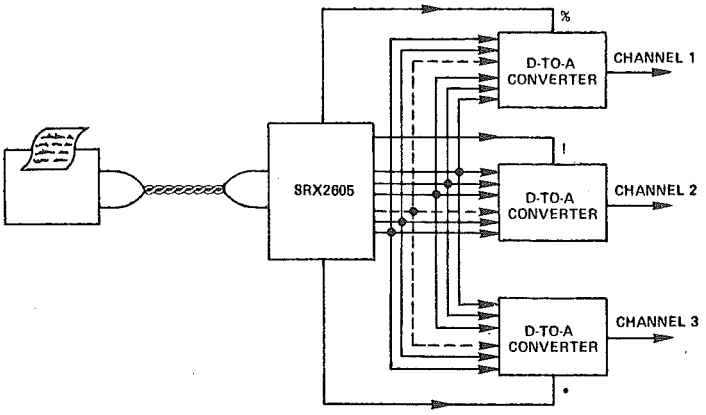


*Figure 29. Self-multiplexed analog output. Several D/A converters are connected in parallel to one SRX2605. The %, !, or * strobes the input register of the desired D/A converter.*

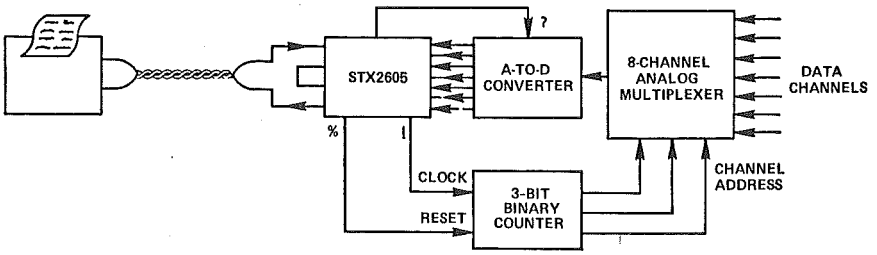


Figure 30. Teletypewriter or computer-controlled n-channel remote analog sequential channel addressing, using binary counter. ! Advances the counter, % resets to first channel, and ? initiates data transmission back to the teletypewriter.

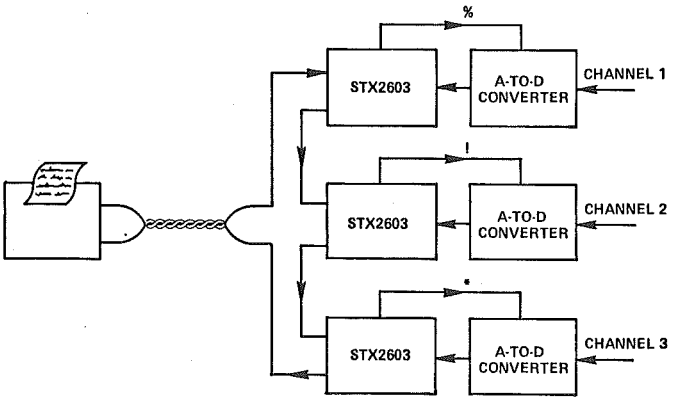


*Figure 31. Many transmitters self-multiplexed on one series current loop. %, !, or * interrogates the respective channels and transmits the data to the teletypewriter.*

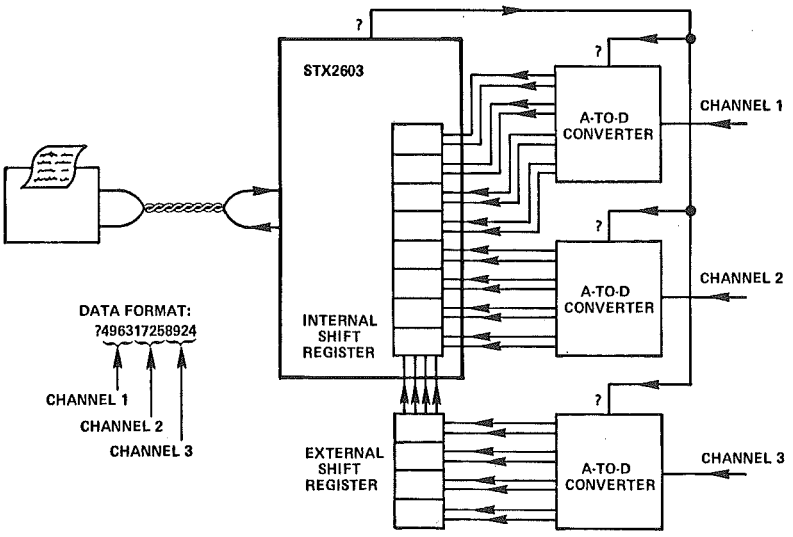


Figure 32. Using external shift register for simultaneous transmission of many channels with only one STX2603.

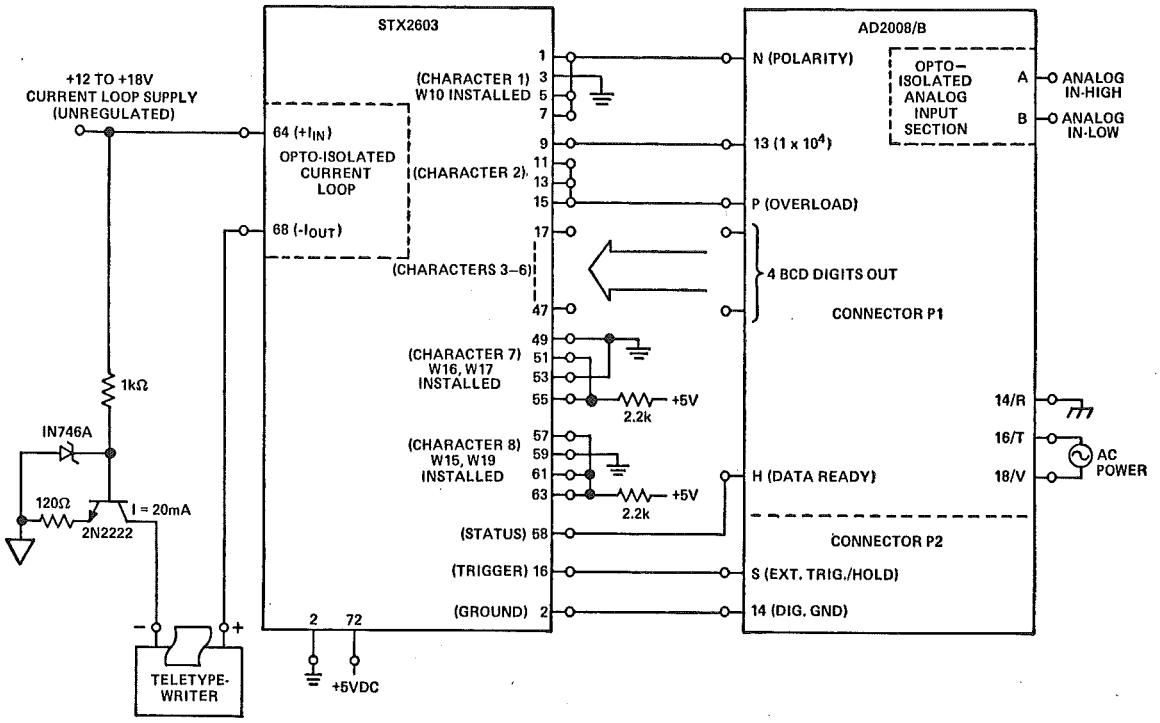


Figure 33. A digital panel meter AD2008/B is interfaced to a teletypewriter through an STX2603 Transmitter Card. Using this hookup, the DPM will convert upon receipt of a "?" typed on the teletypewriter, and transmit the data back after a conversion has been completed. The information will include polarity, overrange, overload, 4 BCD digits, line feed, and carriage return.

CONCLUSION

We have summarized in this chapter a variety of means of implementing converter-interface functions in terms of standard readily available products, at the various levels of system involvement that are likely to concern our readers. More information on any aspect of this discussion is no farther away than the nearest Analog Devices sales office.