CHAPTER 10: DIGITAL INSTRUMENTATION PRINCIPLES

I. Why digital?

Almost all the transducers we have considered so far have had an analog output, that is, the output is a different form from the input but its magnitude is proportional to the magnitude of the input. Why should we go to the trouble and expense of turning an analog signal into a digital one in which the output quantity is converted into a series of voltage pulses representing its numerical equivalent? The main reason is precision. Once a quantity has been expressed as a number it cannot change its value like an analog signal may do while it is being processed. Also, if the signal is expressed in binary digits, the processing can be done numerically in a microprocessor or computer. An analog signal is converted into a digital one by sampling, as shown in Figure 9.12.

The conversion is done by sampling and hold amplifier as shown in Figure 9.13a. The first op-amp is a wide bandwidth amplifier which passes the analog voltage to the second op-amp while the switch is closed (sampling phase). When the switch is opened, the second op-amp maintains the voltage it has at that instant, with as little drop as possible, for the circuits which follow (holding phase). Figure 9.13b shows the process as a voltage/time graph. With the switch closed, the output tracks the input. When the hold signal is applied to the switch control there is a slight delay, the aperture time, typical 50ns before the switch opens. This can be allowed for by advancing the hold signal by the same amount. After the switching transient has died out, the output stays constant except for the droop, typically 1mV/ms. There is a similar delay on switching from hold to sample, before the output can start tracking the input again. This delay, the acquisition time, is usually one to two microseconds from the instant of switching until the switching transient has died out.

In digital systems, the voltage of the power supply to the integrated circuit chips (Vcc) is usually +5 V DC. The inputs to and outputs from the chips are the binary digits 1 or 0, '1' being indicated by any voltage from Vcc down to +2 V, and '0' by any voltage from 0V up to +0.8 V. (Somewhat different voltages may apply to some families of integrated circuit chips but the principle is the same.) These two states may be referred to as 'high' and 'low' instead of '1' and '0'.
II. The sampling theorem

“A continuous signal can be represented completely by, and reconstructed from, a set of instantaneous measurements or samples of its voltage which are made at equal spaced times. The interval between such samples must be less than one-half of the period of the highest-frequency component in the signal”

In other words, we can convert an analog signal into digital samples and convert them back into original signal provided the number of samples per second is more than twice the highest frequency in the signal. Suppose the signal contains a higher frequency than we expected, so that in fact the number of samples per second is less than twice that frequency. We then get what is known as aliasing; the samples convert back into a false lower frequency as shown in Figure 9.14
Of course, an instrumentation signal is unlikely to be the nice tidy sine wave of Figure 9.14. It is more likely to be a varying voltage which is not a regularly repeating waveform. But by Fourier analysis, any such shape can be shown to be a summation of a constant term (in this case, a DC voltage) and sine waves of various amplitudes and frequencies.

III. Analog to digital conversion

The sample voltages from the sample-and-hold amplifier are converted into binary numbers by another integrated circuit, an analogue-to-digital converter (usually abbreviated to A/D converter). The main types of A/D converter: single slope, dual slope, successive approximation and flash are described below.

1. Single-slope or single-ramp A/D converter

This is shown diagrammatically in Figure 9.15. The 'ramp' is a steadily increasing voltage; its voltage/time graph would be an inclined straight line - hence the name. The ramp generator is similar to the one shown in Figure 9.16a (i.e., an op-amp with capacitive feed-back, which makes in integrate voltage with respect to time.)
At the start of the conversion, the control circuit set both the ramp voltage and the binary counter to zero. Any positive analog voltage on the non-inverting input of the op-amp sends its output “high”. This switches on the ramp generator and at the same time enables the AND gate to pass cycles of square wave from the clock input to the binary counter. At the instant the ramp voltage exceeds the analog input voltage, the op-amp goes “low”. This switches off the clock input to the binary counter and it latches (i.e., holds constant in a memory) the binary number it has counted to. The ramp generator and counter are then reset to zero, ready to convert the next sample.

This type of A/D converter is simple and cheap but comparatively slow, a 7-bit converter taking 1ms or more to do conversion. Another disadvantage is its inaccuracy if the clock frequency or the slope of the voltage/time ramp alters due to ageing or temperature-sensitivity of components.

2. Dual-slope A/D converter

This cancels out any inaccuracy due to variations in clock frequency or ramp slope by using two ramps in a count-up, count-down process. The principle is shown in Figure 9.16 and explained in the following sequence:

1. The binary counter is set to zero and the analogue voltage is connected to the inverting input of the first op-amp, the integrator, which generates the ramps; its output is already at zero volts.
2. The positive analogue voltage applied to the inverting input of the first op-amp causes it to start generating a steadily increasing negative voltage at a rate proportional to the input voltage (i.e. a negative ramp with slope proportional to input).
3. The output of the second op-amp (the comparator) instantly goes 'high', enabling the AND gate to admit clock pulses to the binary counter.
4. Counting proceeds until the counter is full. The overflow causes the counter to be reset to zero and start counting again, and the inverting input of the first op-amp is switched to the negative reference voltage.

5. This causes it to generate a positive ramp with slope proportional to the reference voltage. The negative output voltage of the first op-amp therefore starts to decrease; when it reaches zero the comparator output goes 'low', blocking the clock input, and the binary number in the counter is latched by the control circuit. This number is the digital equivalent of the analogue input voltage.

The cycle then recommences. The accuracy of dual-slope A/D converter is unaffected by drift in the clock frequency or in the values of the resistance or capacitance in the integrator circuit, since the upward and downward ramps are affected equally. Also, high-frequency noise disappears in the integration, and changes in the analog input signal during the integration period are averaged out. And if the duration of the integration is arranged to be a multiple of the period of a background noise such as "main hum", that noise is rejected.

The chief advantage of dual-slope converter is its slow speed. However it is used in many digital voltmeters, where its comparative slowness does not matter.

3. Successive-approximation A/D converter

This type of A/D converter, shown in Figure 9.17, is very much faster than the ramp type. Its sequence of operations is as follow:

- Starting with the most significant bit (i.e., the left-hand binary digit) set to 1 and the remaining bits set to 0, it generates the equivalent analog voltage to that number, by means of a built-in D/A. That voltage is compared with the input voltage, in the comparator. If it is less than the input voltage, the 1 is retained, otherwise it is altered to 0
- The next most significant bit is then changed from 0 to 1, the equivalent voltage is again generated, and the comparison repeated.
- The process continues until, finally, the effect of a 1 as the least significant bit (LSB) has been tested. The conversion is then complete and the number is latched to the output.
- The D/A converter output voltages are biased to increase them by the equivalent of \( \frac{1}{2} \) LSB, so that the maximum error in conversion can only be \( \frac{1}{2} \) LSB. Each comparison is operated by one clock pulse, and the complete conversion takes place in a few microseconds.
**Example:**  
Show the stages by which a 3-bit successive approximation A/D converter with an analog span of 0 to 10 volts would convert 6.5V to binary

**Solution**  
With reference to Figure 9.20, for a 3-bit D/A converter,  
10V needs (1/2 step bias)+(7 steps)+(1/2 steps) = 8 steps  
Therefore the stages of conversion are:  
[(4.5/8)x10=5.625V <6.5, therefore retain first 1]  
110  
[(6.5/8)x10=8.125V >6.5, therefore replace second 1]  
100  
[(5.5/8)x10=6.875V >6.5, therefore replace third 1]  
output is 100.

**4. Parallel, simultaneous or “flash” A/D converter**

This is the fastest converter of all. Figure 9.18 illustrates the principle, showing the circuit for a 2-bit 'flash' AID converter with 4V power supply. By means of series resistors forming a potential divider across the power supply, a number of equal voltage steps is obtained. Each of these steps is applied to the inverting input of its own op-amp. The analogue input voltage is applied to the non-inverting inputs of all those op-amps. Where the analogue input voltage is higher than the voltage on the inverting input of an op-amp, the output voltage of the op-amp is 'high' (binary 1), otherwise it is 'low' (binary 0). These two strings of 0’s and 1’s are then turned into a corresponding binary number by means of encoding gates.

The 2-bit converter requires three op-amp comparators; an n-bit converter requires (2n-1) comparators. Thus each additional bit virtually doubles the number of comparators required. All of the comparators and the encoding gates are included on the same integrated circuit chip. At present the practical limit is 11 bits; 12-bit resolution can be obtained by connecting two 11-bit chips in series. 8-bits is a more usual size; an 8-bit ‘flash’ A/D converter chip contains 255 comparators and is capable of making 20 million A/D conversions per second.

![Figure 9.18 A simple example of a ‘flash’ A/D converter: (a) circuit diagram, (b) output table](image)
IV. Analog-to-Digital Conversion Considerations

Saturation Error: The most obvious limitation of an A/D converter is that it has definite upper and lower limits of voltage response. Typical full-scale ranges are 0 to 10V and -10 to +10V. If the input signal exceeds the upper or lower limits of response, the converter saturates and the recorded signal does not vary with the input. This situation can be prevented by appropriate signal conditioning, such as amplitude attenuation or DC offset removal.

Resolution and Quantization Error: As we have seen, an A/D converter responds to discrete changes in the input voltage. Thus there is a smallest increment of voltage change that can be resolved by an A/D converter. In general, the voltage resolution per bit, $\varepsilon_\text{v}$, depends on the full-scale voltage range and the number of bits of the converter:

$$\varepsilon_\text{v} = \frac{\Delta V_{\text{fs}}}{2^n}$$

$\Delta V_{\text{fs}}$ = the full-scale voltage range,
$n$ = the number of bits of the A/D converter.

Typical A/D converters have 8, 12, or 16 bits, corresponding to division of $\Delta V_{\text{fs}}$ into a total of $2^8$=256, $2^{12}$=4096, and $2^{16}$=65,536 increments. A 16-bit converter with a -10 to +10V range has a voltage resolution of 0.3mV. The voltage resolution is a known value for a given A/D converter, and it is normally needed when processing the digitized data.

The finite resolution of the A/D converter introduces error in the recorded values, since the actual analog voltage usually lies between the available bit levels. This is called quantization error (since the digital data are "quantized"), and it is entirely analogous to the reading error of a digital display. An estimate for the quantization uncertainty is $u_q=\varepsilon_\text{v}/2$ (95%). Quantization errors may be reduced by using an A/D converter with more bits.

Conversion Errors. A/D converters may also suffer from slight non-linearity, zero-offset errors, scale errors, or hysteresis. Such errors are a direct by-product of the particular method of input quantization. Normally, the manufacturer will provide specifications for the potential size of such conversion errors.

Sample Rate. The rate at which an A/D converter records successive values of a time-varying input is called the sample rate. Each A/D converter has a maximum possible sample rate, of which typical values range from about 1000Hz to more than 100MHz. Software often allows the user to specify any sample rate up to this maximum value.

Signal Conditioning for A/D Conversion. To make the best use of an A/D converter, conditioning of the analog signal is often required. The most important considerations are prevention of aliasing, minimization of quantization errors, and prevention of saturation errors. Aliasing can be prevented by using a low-pass, or antialiasing, filter to remove frequencies of $fs/2$ or more from the analog signal. Quantization error can be minimized by amplifying the signal to span as much of the full-scale range as possible. However, this approach sometimes conflicts with the need to avoid saturation errors.
V. Digital to analog conversion

After transmission, processing, etc., the digital signal may have to be converted back to analog form. This is done by another integrated circuit: a digital-to-analog converter (D/A).

The D/A converter circuit, known as an R-2R ladder, is shown in Figure 9.19. It has as many inputs as there are digits in the binary numbers which represent the voltage values.

![Figure 9.19 A binary R-2R ladder D/A converter](image)

Each input operates its own electronic switch (a logic gate) which connects that particular leg of the ladder to the reference voltage if the binary digit is a 1, or to earth if it is a 0. If all the legs but one are connected to earth, the one connected to the reference voltage produces a current which flows towards the inverting input of the op-amp and is halved by the resistance network at each junction through which it passes. Thus the current contribution of each leg is weighted to correspond to the position of that particular digit in the complete binary number; for example, in a 4-bit D/A converter the current produced by the most significant bit (MSB) will be 8 times \(2^{(4-1)}=8\) the current produced by the least significant bit (LSB). The op-amp produces an output voltage proportional to the sum of the currents.

![Figure 9.20 Output/input graph of a 3-bit D/A converter](image)
There is a wide variety of D/A converter chips available with various output voltage limits and output current limits. There are also multiplying types, in which the output voltage depends on both the binary input and a varying reference voltage, and digitally buffered types for easy interfacing with a microprocessor, the 'buffer' being a memory which stores bits until the converter can deal with them.

In addition to deciding which of these types he needs, the system designer must also consider the speed at which he wants the chip to work, and the resolution he requires. The speed is specified as the *settling time*; the time it takes for a full scale input to be converted to within half of the least significant bit (1/2 LSB). Settling times range from a few microseconds to a few nanoseconds, depending on cost and the number of bits in the binary input. The number of bits also determines the resolution. Figure 9.20 shows the resolution (1/2 LSB) which would be obtainable from a 3-bit D/A converter. Each additional bit in the binary number doubles the number of steps in the graph, and so halves the (theoretical) maximum error in the output voltage. The actual maximum error depends also on the stability of the reference voltage applied to the chip and the stability of the resistors in the switching circuit.

### VI. A complete digital system

We have now considered the essential components of a digital instrumentation system: a sample-and-hold amplifier, an A/D converter and a D/A converter, such as might be found in a digital tape recorder for instance. And yet if the system consisted only of those components we should be very disappointed in the result, because the output of D/A converter fed with a sequence of digital voltage values will be a series of voltage level steps.

Figure 9.21 shows another low-pass filter before the sample-and-hold amplifier, to filter out high-frequency noise in the signal and prevent aliasing, and a digital filter between the A/D and D/A converters. A digital filter is a microprocessor or logic circuit fulfilling the same function as an analogue filter but doing it much more precisely than the analogue version could, by performing addition, multiplication or delay operations on the stream of binary values. Even with all the filters in place, we shall never get absolutely perfect reproduction of the input signal at the output because the process of sampling changes a smooth analogue curve into an approximation composed of steps made up of horizontal and vertical lines. If we superimposed the approximation onto the original curve we should see that the divergence between them was a maximum at the sharp corners of the steps. It is as if by working from sample voltages of the signal we had introduced noise into the system. This noise is called *quantization noise*. To reduce quantization noise we must reduce the height of the steps by increasing the number of voltage levels at which the LSB changes by 1, and this will require more digits in the digital conversion of the voltage samples.

![Block diagram of a complete digital system](image-url)
**Example:**

A digital system is to transmit an analogue voltage in the range 0 to 10 volts. Calculate the number of discrete voltage levels it can recognize and the minimum height of each step in the analogue output if it is (a) a 7-bit system. (b) an 8-bit system. (Neglect 1/2 LSB biasing.)

**Solution**

a) 7 bits gives a maximum value of

\[ 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127. \]

Adding in the value zero (binary 0000000), the system can recognize 128 levels of voltage, (i.e. \( 2^7 \) levels). The minimum height of each step in the analogue output is:

\[ (10/127) = 0.079 \text{V}. \]

b) The corresponding values for an 8-bit system are 256 levels and 0.039 V.

**VII. Data-acquisition systems**

A *data-acquisition system* is the portion of a measurement system that quantifies and stores data. There are many ways to do this. An engineer who reads a transducer dial, associates a number with the dial position, and records the information in a log book performs all of the tasks germane to a data-acquisition system. In this section, we focus on microprocessor-based data-acquisition systems, which are used to perform data quantification and storage.

Figure 9.22 shows how a data-acquisition system (DAS) might fit into the general measurement scheme between the actual measurement and the subsequent data reduction. A typical signal flow scheme is shown in Figure 9.23 for multiple input signals to a single microprocessor-based/controller DAS.

Dedicated microprocessor systems can continuously perform their programming instructions to measure, store, interpret, and provide process control without any intervention. Such microprocessors have I/O ports to interface with other devices to measure and to output instructions. Programming allows for operations such as, which sensors to measure, and when and how often, and for data reduction. Programming can allow for decision-making and feedback to control process variables.
Personal computer-based data-acquisition systems are hybrid systems combining a data-acquisition package with both the microprocessor and human interface capability of a personal computer (PC). The interface between external instruments and the PC is done by using I/O plug-in boards, which mate either to an expansion slot in the computer or through an external board connected to the computer's bus. This provides direct access to the computer's bus, the main path used for all computer operations. Because the interior of a computer is electrically noisy, the external board is the preferred choice for low-level measurements. Cables connect between remote measuring devices and the plug-in board I/O ports. Software programs called "drivers" are required to drive the interface communication and are readily available.

Another approach is using a dedicated microprocessor. It is well suited to handling repetitive tasks involving measurement, recording, and control and portable. Direct serial or parallel communication with a host computer is possible. In contrast, availability and flexibility are the main attractions of a PC-based system. Not only can data be recorded, but the computer can be used to interface with control equipment, to make programmed decisions to control the measured process, to send data over a network line, to reduce the data into results, and to write the final report.

VIII. Analog Input-Output Communications

Data-Acquisition Boards

Analog interfacing with a computer is most often affected by using a DAS I/O board. Typical units are available in the form of an expansion plug-in board or a PCMCIA (Personal Computer Memory Card International Association) card, so the discussion narrows on these devices. A layout of a typical board is given in Figure 9.24. These boards use an expansion slot on the computer to interface with the computer bus. Field wiring from transducers or other analog equipment is usually made to a screw terminal board with subsequent connection directly to the rear of the I/O board. A typical board allows for data transfer both to and from the computer memory by using A/D conversion, D/A conversion, digital I/O, and counter-timer ports.
The signal flow for plug-in boards is nearly universal. The input lines pass through a multiplexer, which connects through an amplifiers and a sample and hold circuit (SHC) to an A/D converter. Amplifier gain is set either manually, by a resistor jumper or by a software-controlled switch.

Figure 9.25 shows resistance bridge modules allow direct interfacing of strain gauges or other resistance sensors through an on-board Wheatstone bridge.

Transfer of the data through the analog interface board can take place by software or program control, interrupt mode, or direct memory access (DMA).
In software-program control, a control program instructs the computer when to acquire data. This is also the mode commonly used when the operator controls data acquisition directly from the keyboard. In the interrupt mode, the analog board sends a request for service along the interface bus to the CPU. This causes the CPU to interrupt its current operations and to implement a service subroutine to access and acquire the information. The CPU controls the data transfer. The interrupt mode is used when data are not being acquired continuously. As an example, computer printers use the interrupt mode to transfer data out of memory onto paper.

In direct memory access the CPU sets up a DMA controller to allow information to flow directly and continuously from the A/D converter into allocated memory. DMA data transfer provides the highest possible sample rates for data acquisition.

IX. Digital Input-Output Communications

Certain standards exist for the manner in which digital information is communicated between digital devices. Serial communication methods transmit data bit by bit. Parallel communication methods transmit data in simultaneous groups of bits, for example, byte by byte. Both methods use a handshake, an interface procedure, which controls the data transfer between devices and TTL-level signals. Most lab equipment can be equipped to communicate by at least one of these means, and standards have been defined to codify communications between devices of different manufacturers.

Serial Communications RS-232C

The RS-232C protocol was initially set up to translate signals between telephone lines and computers via a modem (modulator-demodulator). The original protocol has been adapted as an interface for communication between a computer and a non-modem device, such as a printer, a digital measuring instrument, or another computer. Basically, the protocol allows two-way communication by using two single-ended signal (+) wires, noted as TRANSMIT and RECEIVE, between data-communications equipment (DCE), such as the modem, and data-terminal equipment (DTE), such as the computer. These two signals are analogous to a telephone's mouthpiece and earpiece signals. A signal GROUND wire allows signal return (-) paths. The remaining wires in the original standard are used to access the state of the telephone lines, if needed.

Most PC computers have an RS-232C compatible I/O port. The popularity of this interface is due to the wide range of equipment that can utilize it. Either a 9-pin or 25-pin connector can be used. The full connection protocol is shown in Figure 9.26. Devices may be separated by distances up to 15m by using a well-shielded cable. Communications can be half-duplex or full duplex. Half-duplex allows one device to transmit while the other receives. Full-duplex allows for both devices to transmit simultaneously.

The RS-232C is a rather loose standard, and there are no strict guidelines on how to implement the handshake. As such, compatibility problems will sometimes arise. The minimum number of wires required between DTE and DCE equipment is the three-wire connection shown in Figure 9.26. This connects only the TRANSMIT, RECEIVE, and GROUND lines while bypassing the handshake lines. The handshaking lines are jumpered to fool either device into handshaking with itself.
Figure 9.26: Standard RS-232C assignments to a 25-pin connector

Communication between similar equipment, DTE to DTE or DCE to DCE, requires only the nine lines connected as shown in Figure 9.27, so a 9-pin connector can be used. The 9-pin connector wiring scheme is shown below.

Figure 9.27: Standard serial connections between DTE and DTE or DCE to DCE equipment

Serial communication implies that data are sent in successive streams of information, 1 bit at a time. The value of each bit is represented by an analog voltage pulse with a 1 and 0 distinguished by two equal voltages of opposite polarity in the range of 3-25V. Communication rates are measured in baud, which refers to the number of signal pulses per second. A typical transmission is 10 serial bits composed of a start bit followed by a 7-or 8-bit data stream, either 1 or no parity bit, and terminated by 1 or 2 stop bits. The start and stop bits form the serial "handshake" at the beginning and end of each data byte. Asynchronous transmission means that information may be sent at random intervals. Thus the start and stop bits are signals used to initiate and to end the transmission of each byte of data transmitted. The start bit allows for synchronization of the clocks of the two communicating devices. The parity bit allows for limited error checking. Parity involves counting the number of 1's in a byte. In one byte of data, there will be an even or odd number of bits with a value of 1. An additional bit added to each byte to make the number of 1 bits a predetermined even or odd number is called a parity bit. The receiving device will count the number of transmitted bits checking for the predetermined even (even parity) or odd (odd parity) number.

Devices that use data buffers will use a software handshaking protocol such as XON/XOFF. With this, the receiving device will transmit an XOFF signal to halt transmission as the buffer nears full and an XON signal when it has emptied and is again ready to receive.

RS-422A/423A/449/485

RS-422A/423A provides recommendations for the electrical interface and RS-449 specifies functional and mechanical characteristics of a serial communication standard. Equipment designed
for RS-232C can be used with the new standards with slight modification. Using a 37-pin connector and differential-ended connections to reduce noise, they allow for communication at rates up to 2-M baud and distances up to 1000 m. The standard uses a +5V TTL pulse signal to distinguish between a 1 bit (+2 to +5.5V) and a 0 bit (-0.6 to +0.8V).

The RS-485 protocol allows for multidrop (allows up to 32 to 255 devices on one line) operation that is well suited to local area networks (LANs). It communicates in half-duplex along a two-wire bus, which makes it slower than RS-422.

**Universal serial bus**

The universal serial bus (USB) is specified to become the industry-standard extension to PC architecture. The intention of this new standard is to permit peripheral expansion for up to 128 devices and to support low to medium transfer rates from 1.5Mbs up to 12Mbs. It supports a "hot swap" feature that allows the user to plug in an USB-compatible device and be able to use it without reboot. This bus connects USB devices to a single computer host through a USB root hub.

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data lines</td>
<td>Digital I/O data line 1</td>
</tr>
<tr>
<td>2</td>
<td>Digital I/O data line 2</td>
</tr>
<tr>
<td>3</td>
<td>Digital I/O data line 3</td>
</tr>
<tr>
<td>4</td>
<td>Digital I/O data line 4</td>
</tr>
<tr>
<td>13</td>
<td>Digital I/O data line 5</td>
</tr>
<tr>
<td>14</td>
<td>Digital I/O data line 6</td>
</tr>
<tr>
<td>15</td>
<td>Digital I/O data line 7</td>
</tr>
<tr>
<td>16</td>
<td>Digital I/O data line 8</td>
</tr>
<tr>
<td>Handshake lines</td>
<td>Data valid (DAV)</td>
</tr>
<tr>
<td>Bus management lines</td>
<td>End or identify (EOI)</td>
</tr>
<tr>
<td>Ground lines</td>
<td>Shield</td>
</tr>
<tr>
<td>18-24</td>
<td>Ground</td>
</tr>
</tbody>
</table>

**Figure 9.28: Cable configuration for the USB**

The USB physical interconnect is a tiered star topology (Figure 9.28). In this setup, the root hub permits one to four attachments, which can be a combination of USB peripheral devices and additional USB hubs. Each successive hub can in turn support up to four devices or hubs. The cable length between a device and hub or between two hubs is limited to 5 m. The connecting cable (Figure 9.28) is a four-line wire consisting of a hub power line, two signal lines (+ and -), and a ground (GRD) line.

**Parallel Communications GPIB (IEEE-488)**

The general purpose interface bus (GPIB) is a high-speed parallel interface originally developed by Hewlett-Packard. The GPIB is usually operated under the IEEE-488 communication standard. The bus allows for the control of other devices through a central controller (e.g., PC computer), and it allows devices to receive or transmit information from or to the controller. This standard is well defined and widely used to interface communication between computers and printers and scientific instrumentation.

The IEEE-488 standard for the GPIB operates from a 16-wire bus with a 24-wire connector. A 25-pin connector is standard. The bus is formed by eight data lines plus eight lines for bus management and handshaking (two-way control communication). The additional eight lines are used
for grounds and shield. Bit parallel, byte serial communication at data rates up to 1 Mbyte/s are possible, with 1 to 10 kbytes/s most common. Connector lines are limited to a length of roughly 4m.

The standard requires a controller, a function usually served by the laboratory computer, and permits multidrop operation, allowing up to 15 devices to be attached to the bus at any time. Each device has its own bus address (addresses 1-14). The bus controller (address 0) controls all the bus activities and sequences all communications to and between devices, such as which bus device transmits or receives and when. This is done along the bus management and handshaking lines. The communication along the bus is bi-directional. Normally, ground true ITL logic (≤0.8V HIGH, ≥2V LOW) is used.

The IEEE-488 standard specifies the following.

*Data bus:* This is an 8-bit parallel bus formed by the eight digital I/O data lines (lines 1-4 and 13-16). Data are transmitted as one 8-bit byte at a time. The handshake and bus management lines communicate through the transmission of a HIGH or LOW signal along the line. A HIGH signal asserts a predetermined situation.

*Handshake bus:* A three-wire handshake is specified. The Data Valid (DAV) line (line 6) asserts that data is available on the data bus and is valid. The Not Ready for Data (NRFD) line (line 7) is asserted by a device until it is ready to receive data or instructions. The Not Data Accepted (NDAC) line (line 8) is asserted by a device until it has accepted a set of data. To illustrate the handshake concept, consider the handshake between a controller (computer) and a measuring instrument. The controller unasserts NRFD, indicating that it is ready to accept data. When ready, the measuring instrument asserts DAV, indicating to the controller that valid data is being sent over the data bus. When the controller has read and accepted the data, it unasserts NDAC.

*Bus management:* There are five lines reserved to manage the flow of information on the data bus. The Interface Clear (IFC) line (line 9) is used by the controller to clear devices, such as during the initial boot. The Service Request (SRQ) line (line 10) is used by a device to assert that it is ready to be serviced by the controller, such as when it is ready to transmit data. The Attention (ATN) line (line 11) asserts that the data on the data bus are commands from the controller; devices are not to transmit. A low signal allows device messages onto the data line, such as the device data readings. The End or Identify (EOI) line (line 5) is used by the transmitting device to indicate the end of a data transmission. With the ATN line asserted, it is used by the controller to poll its devices to determine which device asserted its SRQ. The Remote Enable (REN) line (line 17) is used by the controller to place a device in its remote mode. When asserted, the front panels of the device are deactivated and the controller has command over device programming.

The interplay between the controller and the devices on the bus are controlled by software programs. Most scientific devices rely on software drivers; many of these are built around menu-driven programs, which are designed to provide some flexibility in the specific setup or operating parameters of the device. A parallel interface is faster and more efficient at data transfer than a serial interface.

**EXAMPLE**

A strain transducer has a static sensitivity of 2.5 V/unit strain (2.5 mV/µε) and requires 5Vdc of power. It is to be connected to a DAS having a ±5 V, 12-bit A/D converter and its signal measured at 1000Hz. The transducer signal is to be amplified and filtered. For an expected measurement range of 1-500 µε, specify appropriate values for amplifier gain, filter type, and cutoff frequency, and show a signal flow diagram for the connections.
SOLUTION

The signal flow diagram is shown in Figure 9.29. Power is drawn off the data-acquisition board and routed to the transducer. Transducer signal wires are shielded and routed through the amplifier and filter, and they are connected to the data-acquisition board connector block by using twisted pairs to channel 0, as shown. The differential-ended connection at the board will reduce noise.

The amplifier gain is determined by considering the minimum and maximum signal magnitudes expected, and the quantization error of the DAS. The nominal signal magnitude will range from 2.5μV to 1.25 mV. An amplifier gain of $G = 1000$ will boost this from 2.5mV to 1.25V. This lower value is of the order of the quantization error of the 12-bit converter. A gain of $G = 3000$ will raise the low end of the signal out of quantization noise while keeping the high end out of saturation.

With a sample rate of $f_s = 1000$ Hz, a general purpose anti-alias, low-pass Butterworth filter with 40 dB/decade roll-off and a cutoff frequency set at $f_0= 500$ Hz would meet the task.

![Figure 9.29: Line connection for the example](image-url)