Practical Interface Circuits for Micros

George Loveday

Practical Interface Circuits for Micros

George Loveday
CEng, MIERE
Senior Lecturer in Electronic Engineering
Bromley College of Technology

Pitman

PITMAN PUBLISHING LIMITED
128 Long Acre, London WC2E 9AN

Associated Companies
Pitman Publishing New Zealand Ltd, Wellington
Pitman Publishing Pty Ltd, Melbourne

First published in Great Britain 1984

ISBN 0 273 01998 8

Text set in 10/12 pt Linotron Times,
printed in Great Britain
at The Pitman Press, Bath

Preface

The art of interfacing a microcomputer or a microprocessor within a control system requires an understanding of both software and hardware techniques. In this book I have attempted to integrate both of these aspects to give an introduction to the more practical methods involved in interfacing. The text includes chapters on: electronic devices and circuits used in interfacing; sensors and output devices, data convertors (A to D and D to A); interface adaptors and microprocessors; and system examples My aim has been to make the text suitable for programmers who have a wish to extend their knowledge of electronic components and systems, and for electronic engineers and students who may need to broaden their knowledge of microprocessors and software techniques In view of this there may be topics which will effectively be revision for some readers.

Further information on electronic devices, components and techniques can be found in *Essential Electronics* (Loveday) and on microprocessors in *Microprocessors: Essentials, Components, Systems* (Meadows and Parsons), both published by Pitman.

Throughout the text the following symbols have been used

for immediate mode

$ for hexadecimal numbers

I have also taken to spelling Analogue as Analog.

G L June 1984

Contents

1 Systems

1.1 The Systems Model

Suppose we need to connect up a microprocessor or microcomputer as the controlling element of some process; perhaps several different devices such as input sensors, amplifiers, convertors and output drivers will be required. Having assembled the various elements and made all the correct connections we will then have a *system*, that is to say a collection of various component parts all working together as a whole. The problems encountered in setting up the system and in getting it running will be mainly concerned with

1 Achieving the correct interconnection method between the various parts.
2 Ensuring that signals from any one part of the system are compatible with those parts to which it is connected.
3 That a proper set of instructions with correctly arranged timing are programmed into the microcomputer.

The first two points are about the types of interface circuitry to be used, that is the circuits which link the various elements, while the third depends upon the quality of the software, i.e. the program.

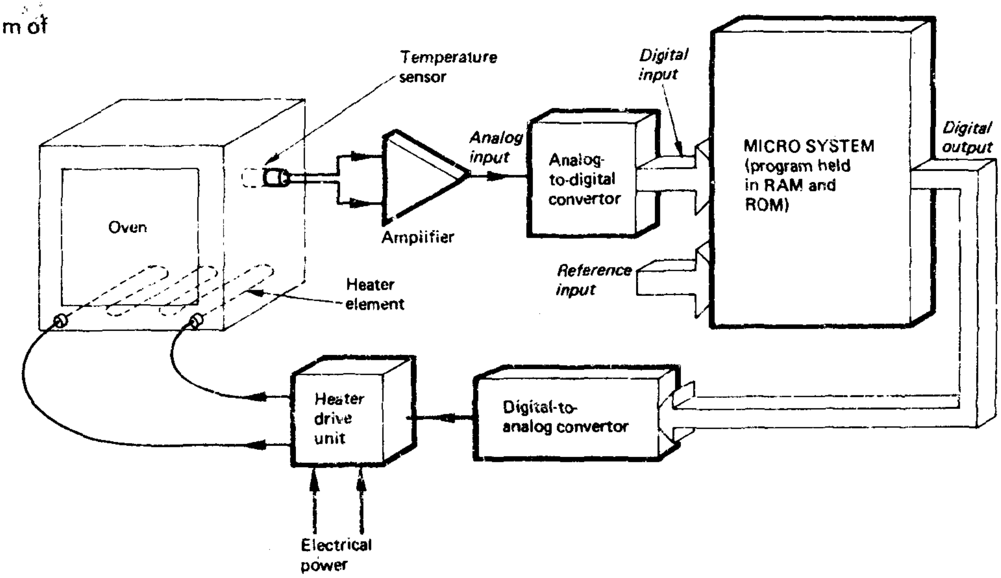
A good first approach to any problem such as this, *before* beginning to write the program, is to follow a logical design sequence:

a) Define the exact task that the system has to carry out.
b) Sketch a block diagram of the essential parts of the system.
c) Produce a flowchart of the actions required by the system.

Take the example of an oven controller in which the oven temperature has to be maintained at a preset value set by a reference input. For a start we shall assume that only a simple type of control is required, but obviously a micro system would be capable of very sophisticated and subtle control of the heating process.

The task as already defined is to maintain the temperature in the oven as close as possible to the desired value, and *fig. 1.1* shows the basic block diagram of the proposed system. The temperature inside the oven has to be sensed by some kind of transducer (a device for converting one form of energy into another). In this case, either a thermocouple or a thermistor with bridge arrangement would be suitable to convert the heat energy into an electrical signal. An amplifier is necessary to increase the size of the relatively weak transducer signal; and before being sampled by the microcomputer this varying signal (an analog signal) must be converted into a suitable digital form. The digital word applied to the microcomputer will then be proportional to the temperature level in the oven.

Fig. 1.1 Block diagram of a heater system



The instructions contained in the program memory of the microcomputer will cause this digital signal to be sampled and compared with the reference input (a digital word set to be proportional to the desired temperature). Depending on the result of this comparison, the microcomputer will output a digital command to adjust the amount of electric power applied to the heater. The flowchart for the various steps required in the program is shown in *fig. 1.2*

What must be noted from this initial outline of our temperature control system is that, although providing an excellent starting point for a design, it in no way gives enough detail to allow the systems to be fully implemented. Also it only shows one of several methods of carrying out the task. The type of sensor, amplifier, analog-to-digital convertor and heater drive unit are not closely specified. In each case there are several different devices, circuits and techniques to choose from. It is at this stage that the most difficulty in getting the system operational may be experienced. Some skill and knowledge is required to get the various pieces of hardware to operate at the optimum level. The following sections deal with the problems which arise in this design area, that is in the interface circuits and input/output devices.

1.2 The Interface Problem

What is the definition of the word interface? It could be taken as

a) "The boundary between two regions", as for example the beach is the interface between the land and the sea, or

b) "The area in which the interactions take place between any two connected parts in a system".

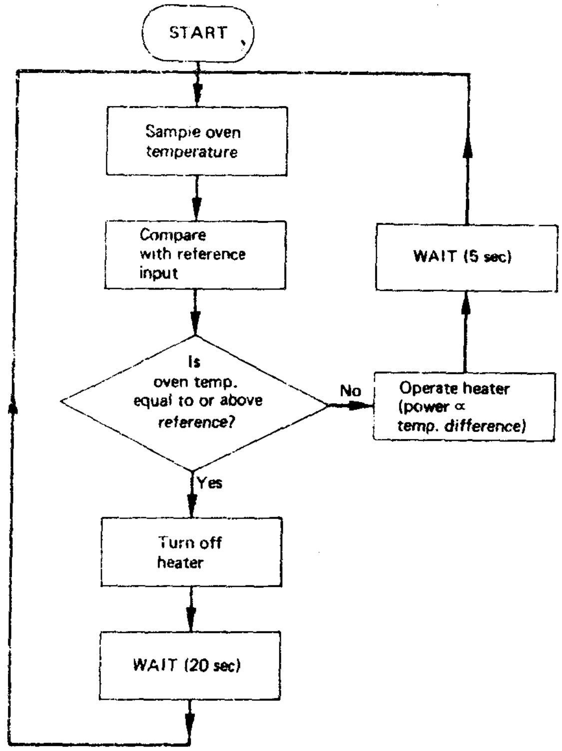
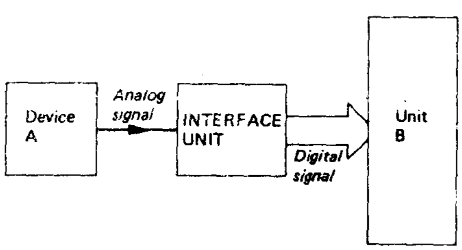


Fig. 1.2 Flowchart for heater system

For our purposes the latter is more appropriate and an interface can be taken to mean any electronic circuit that takes a signal from one part of a system and adjusts it so that the resulting signal is fully compatible with another part of the system. This is illustrated in *fig. 1.3* where the interface receives a varying signal from device A and converts it into a suitable form (in this case digital) to match the input requirements of unit B.

It can be seen, then, that a wide variety of circuits fall within the range of the description "interface" and that in fact a particular interface circuit could well be unique to one system. This extensive range of circuits and devices to choose from could be said to be the *interface problem.* The solution to this ever-expanding range of unique circuits and devices is to study techniques, for very often an interface called (x) uses the same technique in achieving a signal match or translation as a circuit (y). Although the circuits and connections may appear entirely different, the technique, i.e. the method used to get the match, is the same

Fig. 1.3 Basic interface



Having understood the various techniques it is simply a matter of adapting them to suit a particular need. Some of the commonly used interfacing techniques can be broadly listed under the following headings. A few brief explanatory notes are provided here but each technique will be covered in much more detail further on.

1 *Multiplexing and sampling*

Multiplexing is the process by which several data inputs can be switched, one at a time, to a common line (*fig. 1.4*). The switch, an electronic type, operates at a set speed or as commanded by computer control and therefore samples each input for a defined time period. During the time between samples, the input data level is held relatively constant ready for conversion. This last point is also referred to as "sample and hold".

2 *Sequencing and timing*

It follows from the previous paragraph that the sampling of input sensors and the operation of output devices has to be carefully controlled so that the required tasks are carried out in the arranged order and in correct time sequence. The microprocessor itself can be set the task of generating these timing signals via the program but quite often, to save computer time, an external timer as part of the interface is used.

3 *Conversion from analog to digital and vice versa*

To be acceptable to a microprocessor, any varying input signal must be first converted into some form of suitably coded digital word. Suppose the temperature in an oven is being monitored using a temperature transducer. After amplification, the voltage level proportional to temperature is applied to an analog-to-digital convertor (ADC), as shown in *fig. 1.5*. If a 4-bit digital word is used, a table of conversions could be drawn up as follows:

Temperature (°C)	*Analog input to ADC (V)*	*Digital output from ADC MSB LSB*
20	1	Ø Ø Ø 1
40	2	Ø Ø 1 Ø
60	3	Ø Ø 1 1
80	4	Ø 1 Ø Ø
100	5	Ø 1 Ø 1
300	15	1 1 1 1

Here, because only 4 bits exist in the digital output, each 20°C increase in temperature changes the digital output by 1 (one). The varying analog input has been split into 16 discrete levels and it would not be possible using only 4 bits to resolve temperatures closer together than 20°C. In any conversion, the greater number of bits used in the digital word, the higher will be the resolution.

In the same way, if the output from a microprocessor is required to drive some analog device, for instance a chart recorder, then a digital-to-analog convertor is required. Again, the larger the number of bits, the finer will be the resulting analog output.

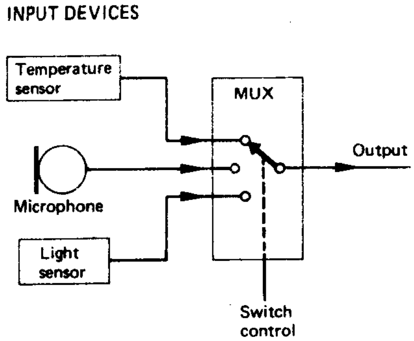


Fig. 1.4 Multiplexing

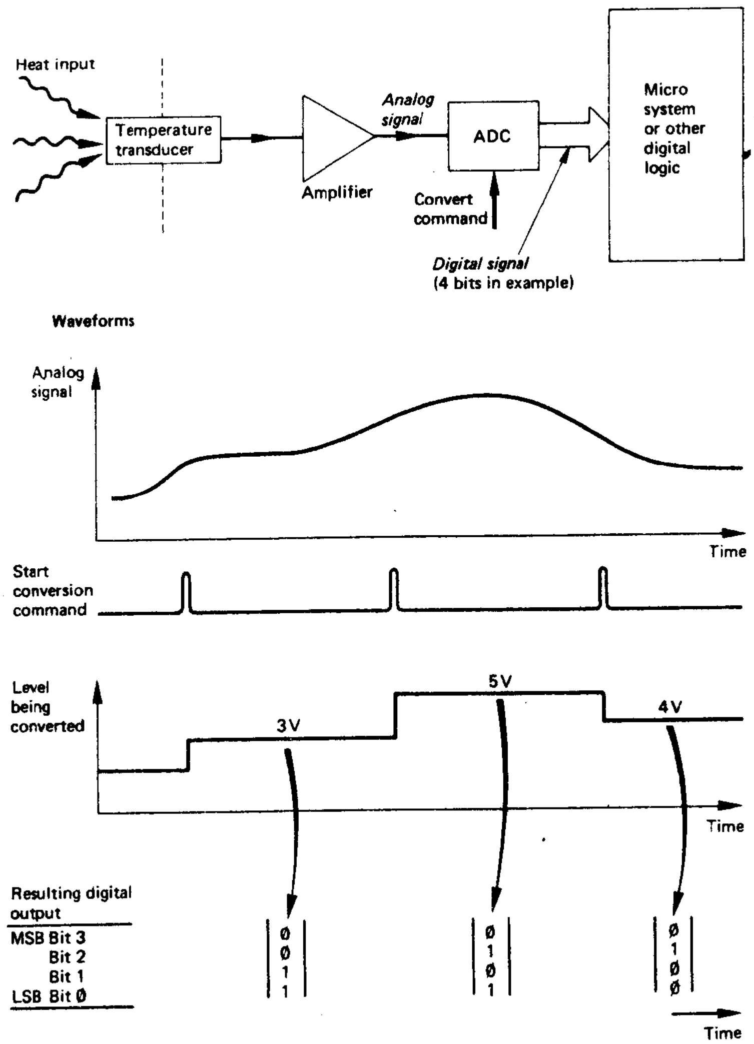


Fig. 1.5 Principle of analog-to-digital conversion

Well-established techniques exist for the task of converting from analog to digital and vice versa, and several ICs are available, most of which are provided with facilities allowing control via a microprocessor.

4 *The use of interrupts or polling*
An **interrupt** is a useful way of getting the microprocessor to service (i.e. accept data from) an input device. Basically, the device sends a signal requesting attention to the interrupt input (IRQ) of the microprocessor. The microprocessor having carried out its current program instruction will acknowledge the call and take the incoming signal. This is the basis of what is called **handshaking**, shown in *fig. 1.6*, a technique that is also used for outputting from the microprocessor to a device.

When there are several input devices to the system, an alternative method called **polling** can be used. In this method the microprocessor, via the program, checks each device in turn until it finds a device which is requesting attention. That is,

device A any data ready?
device B any data ready?
device C any data ready?
and so on

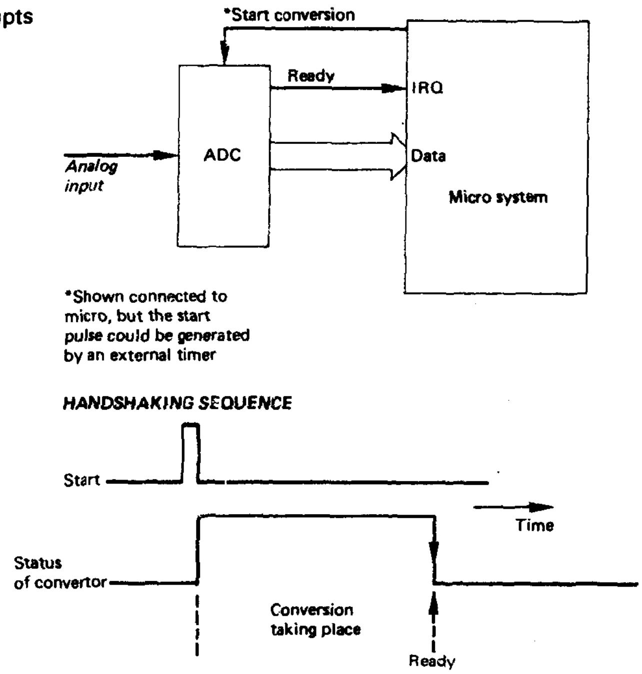
The software can arrange a *priority* for the polling so that fast devices can be checked more frequently.

5 *Controlling power at the output*
In most control systems, the object is to get some actual work achieved at the output. The power device may be controlling flow, heat, light intensity, or position, but essentially it is the tiny output signal available from the microprocessor which has to control, via the interface, the large output power. Before we consider some techniques for power control, it is important to realise that *isolation* between any high voltage supply (typically the a.c. mains at 240 volts) and the sensitive micro system is a vital feature. Any short circuit failure of the power device or interface circuit must not be allowed to connect the high voltage supply into the micro system. Apart from the safety hazard you can imagine the damage that could result.

One popular method for achieving effective isolation is to use a device called an **opto-coupler** or **opto-isolator**. This small fully-enclosed IC consists of a light-emitting diode optically coupled to, but electrically insulated from, a light-sensitive device. *Fig. 1.7* shows one arrangement. The light-emitting diode (LED) is connected to the micro side of the interface and converts electrical input signals into light pulses. These are transmitted via the light pipe to the receiving device which then controls the input to the high voltage part of the circuit. There is no electrical connection between the two parts of the circuit and, because the light pipe is an excellent insulator, the voltage isolation between the output and input can be in excess of 4 kV. Alternative more traditional methods of isolation include the use of transformers or relays.

Having dealt with the isolation problem we can now look at some techniques for power control.

Fig. 1.6 Use of interrupts



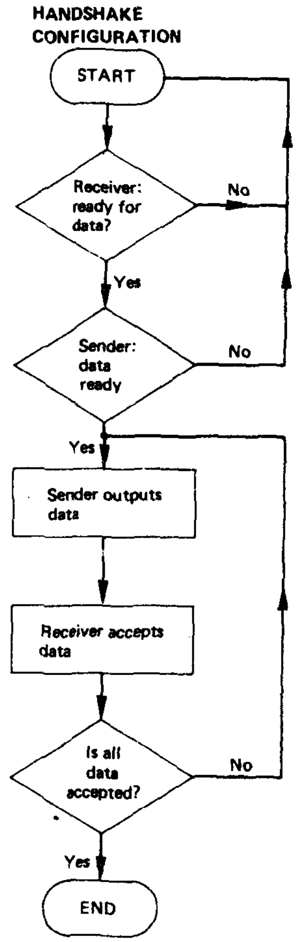
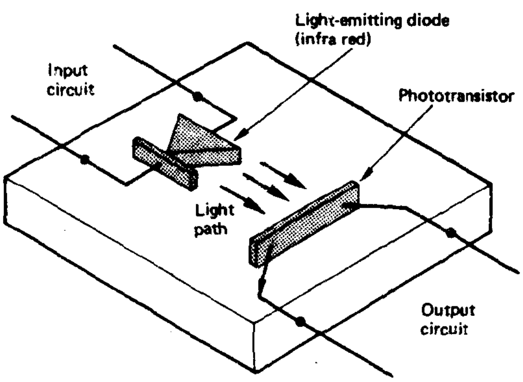
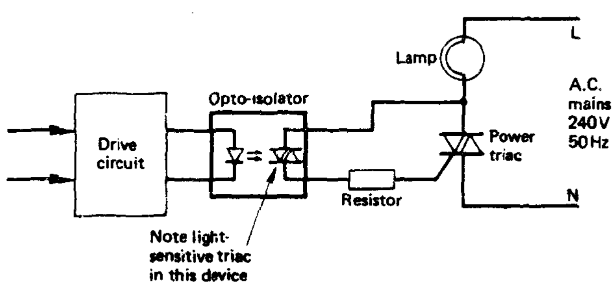


Fig. 1.7 Use of opto-coupler for isolation





1 Control in an ON/OFF switching situation is relatively straightforward. The interface will consist of an electronic power switching device such as a Darlington transistor or VMOS power FET. This will be driven fully ON when the micro outputs a logic 1 signal and it will be OFF when the micro output is logic Ø. The main points are to ensure that the device can pass the required current in the ON state and can withstand the full voltage in the OFF state. This is discussed fully later.

2 A more complicated arrangement has to be set up when the power in the load needs to be controlled over a wide range, and the techniques involved depend upon the type of supply being used. For d.c. situations it is best to use a pulse width modulation system (PWM). Suppose the speed of a d.c. motor is to be controlled over the range 60 rev/min to 3000 rev/min (*fig. 1.8*). This can be achieved by simply varying the d.c. voltage applied to the motor; but a PWM method does it by switching the motor on for a variable time period. For low speed, the motor would be switched to the d.c. supply for a relatively short time in one cycle, whereas for high speed the motor would be switched on for nearly the whole time period in one cycle. The width of the switching waveform and the frequency could be controlled via the program in the microprocessor.

3 A.C. power control is achieved using either what is called phase control or burst firing. In **phase control**, a trigger pulse is generated with its time position variable (set by the micro) with respect to the start of each mains half-cycle. This trigger pulse is used to "fire" an electronic latching switch such as a triac which then connects power to the load. As the mains waveform goes through zero, the latching device switches off and then conducts again at a point in the negative half-cycle when the next trigger pulse is generated. This is more easily seen from a waveform diagram as in *fig. 1.9*. This shows that the greater the delay between the start of each mains half cycle and the trigger pulse, the smaller will be the power supplied to the load. The delay time can be controlled at the interface by a digital command signal from the microprocessor. The method is particularly useful in lamp dimmers and a.c. motor speed control.

Where a load has a much slower response, heaters being an obvious example, the **burst firing** technique can be used. In this method, the load is connected to the mains supply by the switching device for a set number of whole cycles. (Each cycle in the 50 Hz mains takes 20 msec.) Thus for low power dissipation in the load, the switch may be on for only a few cycles in every hundred (10 in a 100 gives 10%), whereas fifty cycles in every hundred would give half power. One of the advantages of this method compared with phase control is that very little interference (electrical "noise") is generated because the actual time of switching takes place when the mains voltage has just passed through zero. This is shown in *fig. 1.10*.

Fig. 1.8 Principle of pulse width modulation control

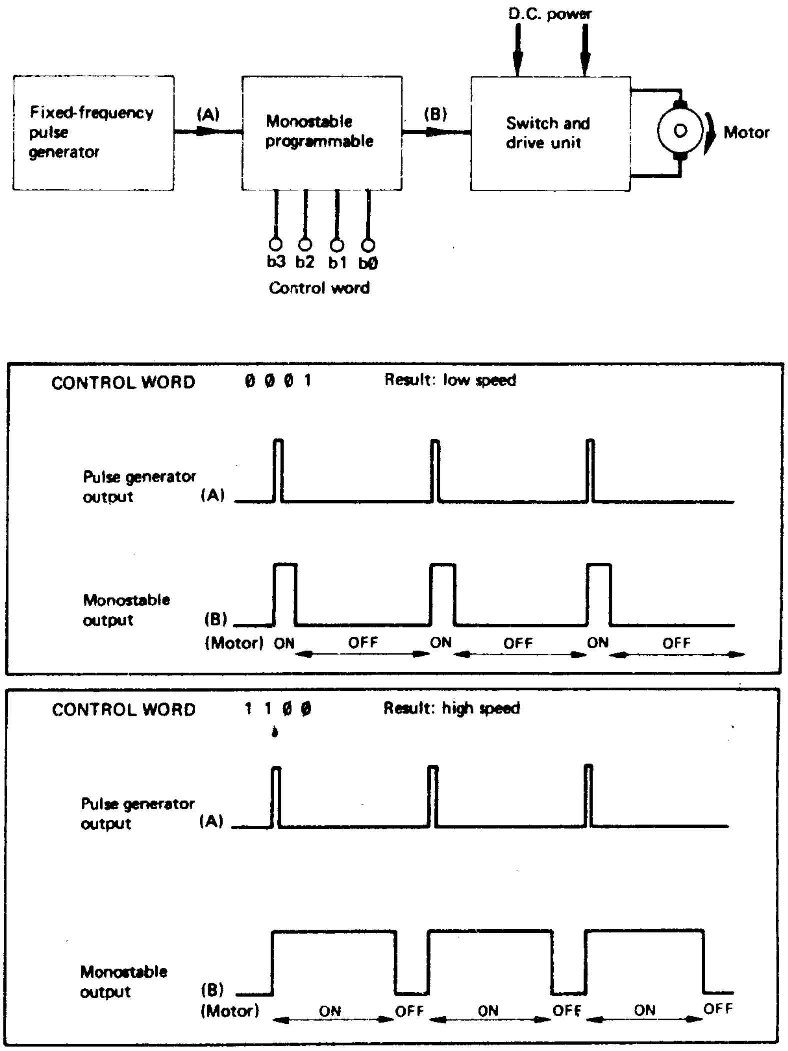
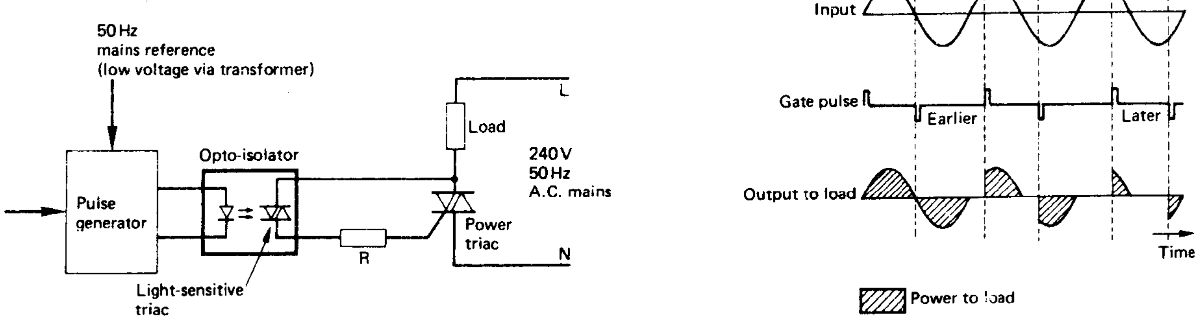


Fig. 1.9 Phase control in a.c. power circuits



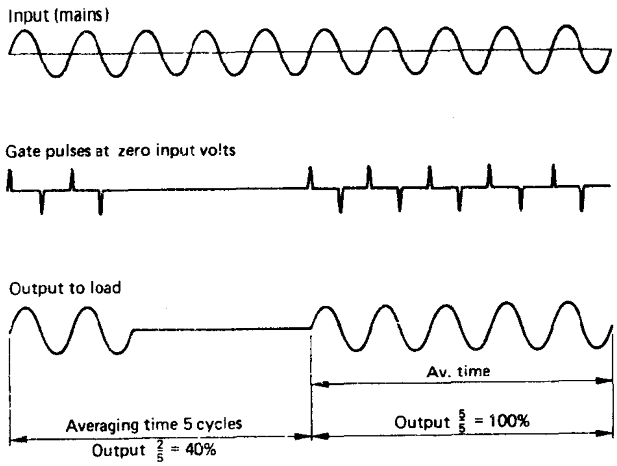


Fig. 1.10 Burst firing

1.3 Types of System

Up to now we have considered systems mainly from the digital point of view, but a further look at the various types of system and the way in which they can be implemented can be instructive. It is useful to make a comparison between the analog and digital forms since this will assist in making the decision about the advisability of interfacing a microcomputer or microprocessor in the first instance. In some situations an analog system may in fact prove simpler and cheaper to implement. Using a micro system for some trivial task could be wasting its enormous potential.

Before we compare the analog versus digital approaches, we must distinguish between the two main types of control. These are

OPEN LOOP and CLOSED LOOP

In an **open loop system**, the required output is set by a reference level applied at the input. The output is completely unaffected by the result it produces. An example is shown in *fig. 1.11a* where the input voltage level from the potentiometer sets, via the controller, the output valve to a desired position. It is assumed that, with the valve set to this position, a certain fixed amount of liquid flow will take place; but this will only be the case if conditions at the output do not vary. Should, for any reason, the viscosity of the liquid in the pipe increase, the rate of flow would fall, and the open loop system would not be able to adjust the valve's position in order to compensate.

The system can be converted to **closed loop** by using a flow-sensing transducer that feeds back a signal proportional to the rate of flow, for comparison with the input reference level. This is shown in *fig. 1.11b*. It is the **feedback** path which closes the loop and enables the system to respond to changing conditions at the output. If the flow rate falls, the feedback signal decreases, causing a net increase in the error signal to the controller. This causes the valve to be opened further to adjust the rate of flow back to nearly its original value.

Fig. 1.11a Open loop system

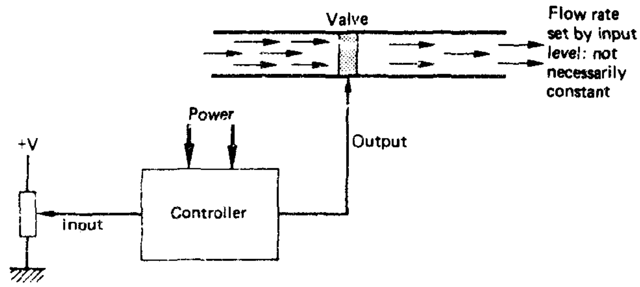
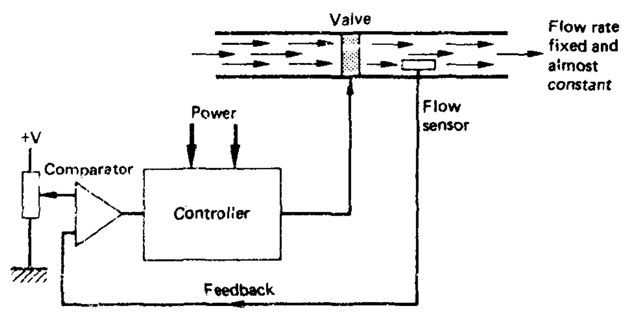


Fig. 1.11b Closed loop system



A closed loop system is inherently more accurate than an open loop type but is more complex and has more of a tendency to instability. This instability can result if too much gain is set into the error amplifier. Fast response would be achieved but the valve's position might move backwards and forwards for a few cycles before settling down following an input of a new desired position. Of course this would depend upon the inertia of the output arrangement and there are relatively simple techniques for ensuring that a closed loop system remains stable.

The system block diagrams of *fig. 1.11* are completely analog (linear) but both the open loop and the closed loop forms can also be created using microprocessor digital controllers. What then are the differences between analog and digital systems? We start by looking at signals. A true **analog signal** is one that is of a continuous nature; in other words, it is a quantity that can take any value within defined limits. The majority of inputs to systems—the electrical signals from transducers—are therefore analog. This is because most of the energy forms which have to be measured for control purposes, such as heat, force, velocity, pressure, and light intensity, are all continuously variable and not discontinuous. Take the output from a microphone; it is an electrical voltage which follows (i.e. is analogous to) the input sound wave (*fig. 1.12*).

In contrast, a **digital signal** is discontinuous and consists of either a coded serial pattern or a parallel group of discrete voltage levels. Commonly, only two levels are used:

On/Off Low/High Logic Ø/Logic 1

Fig. 1.12 The difference between analog and digital signals

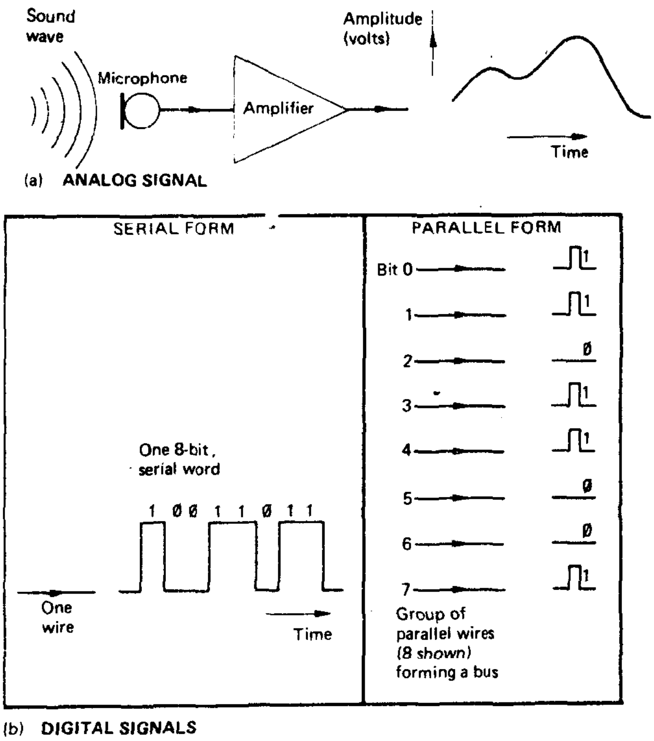


Figure 1.12 illustrates the difference between the two kinds of signal, showing how a small time section of an analog waveform could be digitised. Obviously, the conversion requires a certain amount of hardware (ADC in this case), making the digital system more complex than the analog, and some information detail contained in the analog signal will be lost in the conversion process. This loss of resolution can be minimised by using more bits in the digital word, but a finite loss always exists. To counteract these points, the digital system has several distinct advantages over the purely analog form:

a) The digital signal is based on High/Low levels which can be well defined and which will be much less affected by noise and interference than analog.
b) Many two-state electronic switches exist, making it easy for manufacturers to implement complex digital circuits. With VLSI (very-large-scale integration), more than 100 000 transistor switches can be arranged in one IC chip.
c) A digital signal, being a group of 1s and ∅s, is easy to store, process and manipulate.

This ability to store, process and manipulate data gives the **digital system** great flexibility over its actions. The system can therefore be very versatile, limited only by the ingenuity of the program. An example comparing an analog type with a digital motor speed control system will illustrate this point.