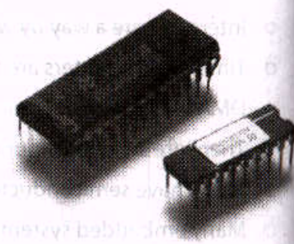
3 SENSORS, ADCs AND ACTUATORS



In this chapter, you will learn

- The working principle of a number of sensors
- The use of temperature and light sensors in practical circuits
- How an LED is used in sensing circuits
- Circuits which use photodiodes and photo transistors
- The working principle of proximity and range sensors
- How an optical encoder works
- The data and control interface of an ADC
- The use of parallel and serial ADCs
- The use of seven segment LEDs and OLEDs
- The ways of using Character and Graphical LCDs
- Usage of stepper motors, DC motors, current drivers and optocouplers
- The working principle of a relay and some practical circuits

Introduction

In our discussion on general purpose I/O (GPIO) in Chapter 2, it was mentioned that depending on the application, the GPIO pins can be used as output or input pins, and various peripheral devices can be connected to these pins. In this chapter, we discuss a few standard and widely used I/O devices, that is, sensors which are input devices and actuators which are output devices. Sensors are required for getting data from the real world and actuators are needed to get the embedded system to 'act' based on this data.

We also devote a small amount of space to 'Analog to Digital Converters' or ADCs as they are commonly referred to. All sensors have analog outputs. But for using them with MCUs, their analog voltages are to be converted to digital numbers. ADCs perform this function, which is a very critical part of the whole system. Without an ADC of good resolution and sensitivity, the whole point in having good sensors is lost. Taking this aspect into consideration, the important aspects of A to D conversion have been looked into.

Chapter-opening image: Two ADC chips.

3.1 | Sensors

Remember the block diagram (Figure 1.2) that we started with. Any embedded system needs sensors—depending on the application, it may be just one sensor or as in most cases, many sensors. A sensor converts a physical quantity into a corresponding voltage. Take an application like a home security system—there should be sensors for sensing temperature, light, motion, humidity, etc. The data obtained from these sensors decide the course of action for the actuators of the system. In this section, we take a quick look at some of the commonly used sensors.

3.1.1 | Temperature Sensors

3.1.1.1 | *Thermistor*

A thermistor is a thermally sensitive resistor, which means that its resistance is 'affected' by the temperature variations around it. Thermistors are made with semiconductor materials and there are two kinds of thermistors—those with NTC (negative temperature coefficient) and PTC (positive thermal coefficient). For the former, the resistance of the thermistor decreases with increase in temperature, and for the latter, it is just the reverse.

Figure 3.1 is a very simple circuit which uses a NTC thermistor. At normal temperatures, the transistor is OFF, because the high resistance of the thermistor prevents it from getting sufficient base current. When the temperature increases, the resistance of the thermistor decreases and at a certain value of thermistor resistance, the base current needed to turn on the transistor is obtained. This switches the transistor ON. The collector voltage goes low, and the LED lights up. Since the collector is connected to the input pin of an MCU, the port P1.1 switches from high to low at a particular 'triggering temperature'. The exact value of this temperature can be varied by varying the value of R. That is why, it is shown as a variable resistor. Figure 3.2 is the photograph of a thermistor.

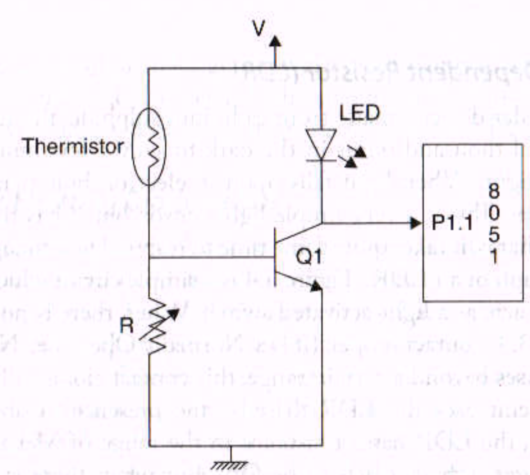


Figure 3.1 | A simple circuit using a thermistor

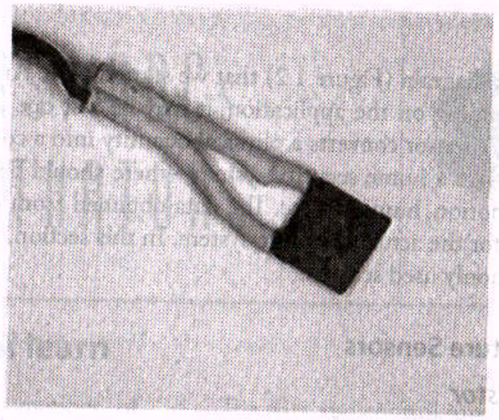


Figure 3.2 | Photograph of a thermistor

3.1.1.2 | *Thermocouple*

A thermocouple is also a sensor for measuring temperature. In this, there are two dissimilar metals, joined together at one end and they produce a voltage proportional to the temperature difference between the two ends of the pair of conductors. One junction is kept at a constant temperature and is called the reference (cold) junction, while the other is the measuring (Hot) junction. When the two junctions are at different temperatures, a voltage is developed across the junction.

Thermocouples are used in many high temperature applications like furnaces, turbines and engine temperature measurements in industries and automobiles.

3.1.2 | Light Sensors

Sensing of light, or rather the blocking of the light that is being sensed continuously is an important feature in many systems. Let's make a review of some of the popular light sensors.

3.1.2.1 | *Light Dependent Resistor (LDR)*

LDRs are very popular devices, made from cadmium sulphide, the resistance of which changes from several thousand ohms in the dark to only a few hundred ohms in the presence of bright light. When light falls upon it, electron hole pairs are created and conductivity increases. This is a very simple light sensor, but it has the disadvantage of 'sluggish' response, that is, it takes quite some time to respond to a change in illumination. Fig 3.3 is a photograph of an LDR. Figure 3.4 is a simple circuit which uses an LDR as a sensor. The circuit acts as a light activated switch. When there is no ambient lighting, the relay (Section 3.3.4) contact is open (it is a 'Normally Open', i.e., NO contact) When the light level increases beyond a certain range, this contact closes.

This simple circuit uses the LDR to sense the presence or absence of light. In the absence of light, the LDR has a resistance in the range of Mega ohms and so the transistor does not get sufficient bias to be ON. But when there is ambient lighting, the resistance of the LDR falls and the transistor gets the bias current to conduct. The

Figure 3.3 | Photograph of an LDR

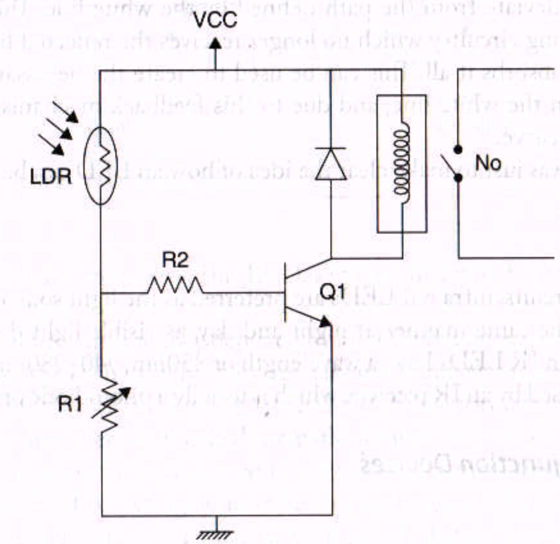


Figure 3.4 | A circuit which uses an LDR for sensing light

transistor switches ON and the relay gets energized—its contact closes. This 'closing' can be used to activate some action, as needed. The amount of illumination required to cause the 'switching' may be adjusted by using the variable resistance R1.

3.1.2.2 | *Light Emitting Diodes (LED)*

LEDs are light generating devices, and as such do not act directly as sensors. Fig 3.5 shows the photograph of an LED. But they can be made to emit light, which is detected by a photo detecting device. This 'detection' can be used as a sensor value. Let's think of some typical applications which use this technique for sensing.

In what is called a line following robot, a robot is made to move continuously on a white line (drawn on a black surface). An LED is fixed under the robotic vehicle. The light from this LED strikes the white line on the ground and reflects it back. There is a photo detecting circuitry to receive this, and the corresponding activation circuitry ensures that the vehicle moves continuously on the white line.

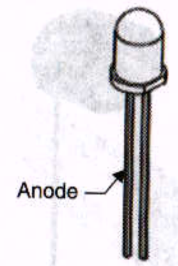


Figure 3.5 | A photograph of an LED

But when the path becomes a curve, the moving vehicle which is moving in a 'straight path' will deviate from the path defined by the white line. This will be sensed by the photodetecting circuitry which no longer receives the reflected light, because the black background absorbs it all. This can be used to create the necessary logic to bring the vehicle back on the white line, and due to this feedback mechanism, it is made to navigate along the curve.

This example was just to make clear the idea of how an LED can be part of a 'sensor' mechanism.

Infra Red LED

For many sensor circuits, infra red LEDs are preferred as the light source. This is because it can be used in the same manner at night and day, as visible light does not affect its operation. Common IR LEDs have a wavelength of 850nm, 940~980nm. The light generated by this is sensed by an IR receiver, which is usually a photodiode or phototransistor.

3.1.2.3 | *Photojunction Devices*

Photodiodes

This is a diode similar to regular semiconductor diodes except that its outer casing is either transparent or has a clear lens to focus the light onto the PN junction for exposure to light, i.e. it is packaged with a window to allow light to reach the sensitive part of the device. It is designed to operate in reverse bias, so that reverse current flows. When light energy strikes the junction, it is this current that increases.

Photodiodes are very smart light sensors that can switch from 'ON' and 'OFF' in nanoseconds. They are commonly used in very many applications from robotics to cameras, TV remote controls, scanners, fax machines, copiers, etc.

Phototransistors

A phototransistor is basically a photodiode with gain. The phototransistor light sensor has its collector-base PN-junction reverse biased and is also exposed to radiant light source. Any normal transistor can be easily converted into a phototransistor light sensor by connecting a photodiode between the collector and base. Now, let's think of an application where light sensing is used.

Intrusion Detection In this circuit (Figure 3.6), an infra red LED output which is activated by an astable oscillator (using the IC NE555, for example) is used to generate a

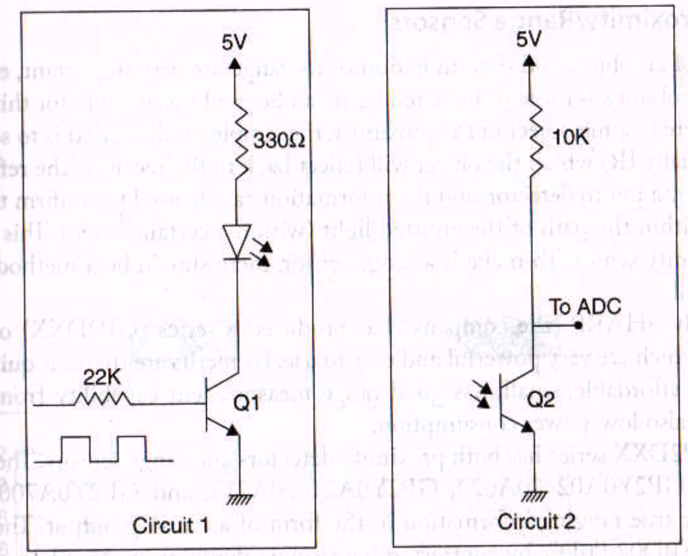


Figure 3.6 | An intrusion detection setup

pulse of some particular frequency. This LED is in Circuit 1 which is the light transmitter circuit.

This pulse train is continuously detected by an infra red detector in Circuit 2. When an intruder blocks the path of light (between circuits 1 and 2), the momentarily absence of light is sensed by the IR sensor in Circuit 2 and this information can be used as an indication that an intruder has blocked the path of light. Any action can be initiated by this, for example, a relay can be activated to trigger an alarm on sensing the intruder.

There is a standard IC acting as an infra red receiver, and it is the TSOP series. For using this, the signal transmitted by the IR LED should have a standard format (refer to its data sheet). The TSOP IC package contains a PIN photodiode, a bandpass filter and a demodulator which converts the signal to a format that a microcontroller can use, i.e. a high or low pulse. Because of the preamplifier and bandpass filter inside, the received signal is robust and free of noise. Such receivers are very popular in simple intruder detector systems, and also in remote control systems, proximity detectors, etc. Fig 3.7 is a photograph of the TSOP (SM0038) IC.

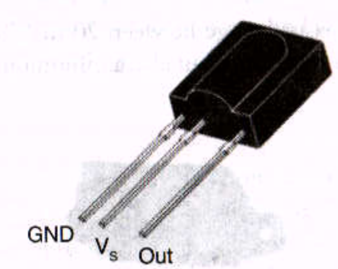


Figure 3.7 | Photograph of a TSOP IC

3.1.3 | Proximity/Range Sensors

Detection of an object and determination of its range are very important, especially in the field of robotics. Visible or infra red light can be used successfully for this. To detect whether there is some object in the proximity, the simple method used is to send a beam of light (usually IR) which the object will reflect back to the receiver. The reflected light is detected by a photo detector, and the information can be used to confirm that there is an object within the path of the emitted light (within a certain range). This is what we call a proximity sensor. To make it a range sensor, there should be a method to find its range, as well.

Recently SHARP (the company) has produced a series (GP2DXX) of IR range finder ICs which are very powerful and easy to use. Its merits are that it is quite accurate, easy to use, affordable, small, has good range measurement capability from inches to metres, and also low-power consumption.

The GP2DXX series has both proximity detectors and range sensors. The GP2D12, GP2D120, GP2Y0A02 ('0A02'), GP2Y0A21 ('0A21'), and GP2Y0A700 ('0A700') sensors offer true ranging information in the form of an analog output. The GP2D15 and GP2DY0D02 ('0D'), by contrast, offer a single digital value based on whether an object is present or not. None of the detectors require an external clock or signal.

3.1.3.1 | *Range Sensing Technique*

The Sharp IR Range Finder (the photograph of which is shown in Fig 3.8) works by the process of triangulation, which is a technique in which a region is divided into a series of triangular elements based on a line of known length, so that accurate measurements of distances and directions may be made by the application of trigonometry.

A pulse of IR light is emitted and then is reflected back if it strikes an object in its path. The reflected beam returns at an angle that is dependent on the distance of the reflecting object. Triangulation works by detecting this reflected beam angle—by knowing the angle, the distance can be determined (Fig 3.9). This type of IR range finder receiver has a special precision lens that transmits the reflected light onto an enclosed linear CCD array based on the triangulation angle. The CCD array then determines the angle and causes the rangefinder to give an analog value, which corresponds to the distance of the object. Additional to this, the 'Sharp IR Range Finder' circuitry applies a modulated frequency to the emitted IR beam. This ranging method is almost (not 100% true!!) immune to interference from ambient light, and is indifferent to the colour of the detected object.

Note These sensors can measure range between 20 to 150 cms (approximate), which means that there is not only a 'maximum' but also a minimum for the range measurement.

Figure 3.8 | Photograph of a SHARP sensor

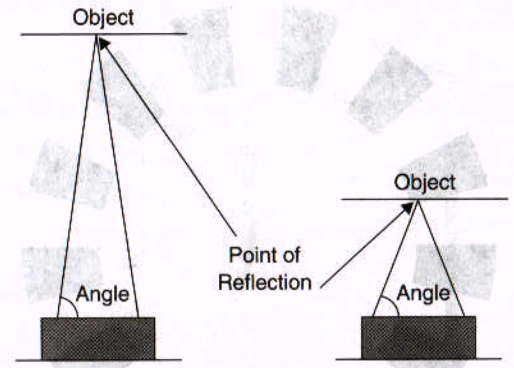


Figure 3.9 | The technique for sensing range

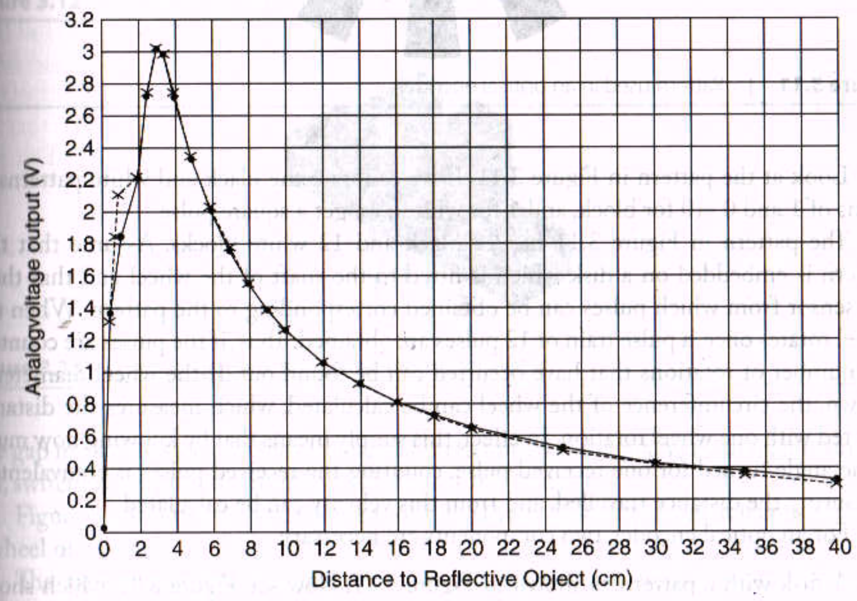


Figure 3.10 | Graph of voltage vs range for a range sensor

For example, observe the voltage output (Figure 3.10) from the GP2D120 sensor which has the range specified to be between 4 and 13 cms.

The characteristic shows that below 4 cm, the output falls and may be wrongly interpreted as a large range. In robotics, the best method would be to use more than one range finder and then to cross fire them.

3.1.4 | Encoders

There is a sensor for finding the speed and direction (and thus the position and distance travelled) of a moving vehicle. This is an 'encoder' which can be fitted to the shaft of the wheel of the vehicle. It uses optical principals and is called an 'optical encoder'. Such encoders work on the principle of counting the number of transitions across a black and white pattern.

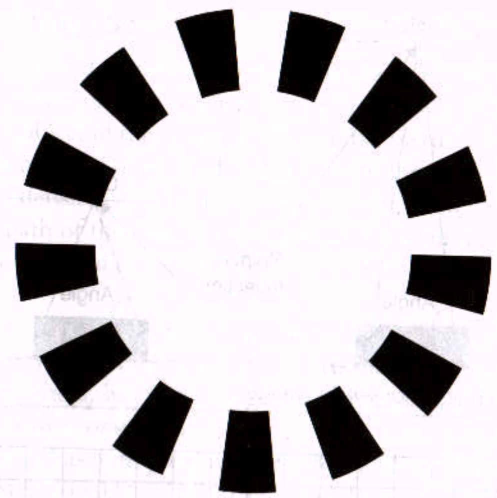


Figure 3.11 | Pattern used in an optical encoder

Look at the pattern in Figure 3.11. If we calibrate the black and white patterns in terms of 1 and 0 –(0 for block, and 1 for white), we get a square pulse.

The pattern in Figure 3.11 has 12 black and 12 white blocks. Assume that this pattern is embedded on a disk which is fitted to the shaft of the wheel and that there is a sensor from which pulses can be obtained corresponding to the pattern. When the wheel rotates once, a pulse train of 12 pulses are obtained. Thus, if the pulses are counted, the number of rotations that have occurred can be found out. If the wheel diameter is known, the circumference of the wheel can be calculated, which measures the distance covered with one wheel rotation. In effect, this simply means that by knowing how much is the angle turned for one received pulse, counting the received pulses is equivalent to measuring the distance travelled, and from this velocity can be calculated.

For an optical encoder, two components are necessary

i) A disk with a pattern as shown in Figure 3.11. Now see Figure 3.12 which shows such a disk.
ii) An LED which generates light which passes through the holes in the disk, or is blocked by the disc.
iii) An optical sensor which receives these light pulses and converts it to electrical signals.

Figure 3.12 shows a pattern disk with a few holes, which can be attached to the shaft of a moving wheel. As the wheel rotates, the light passes through the holes and this is sensed by the receiver on the PCB shown. The pulse train obtained can be used to calculate the distance travelled (from which velocity can be obtained).

IC s are available, with an IR LED and a photodetector in one package, which act as the receiver. See the optical interruptor switch in Figure 3.13 (also called the break beam switch) with a U-shaped slot using which it can be fixed to the rotating wheels.

Such an IC is the H21A1/H21A2/H21A3 series which consist of a gallium arsenide infrared emitting diode coupled with a silicon phototransistor in a plastic housing.

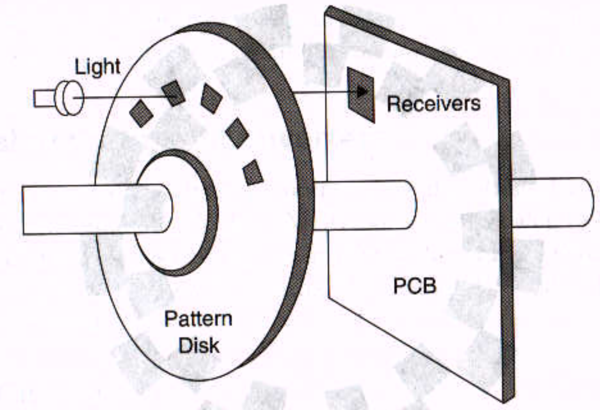


Figure 3.12 | Optical encoder transmitter and receiver

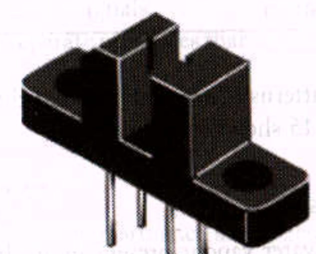


Figure 3.13 | Photograph of an optical interruptor switch

The gap in the housing provides a means of interrupting the signal with an opaque material, switching the output from an 'ON' to an 'OFF' state.

Figure 3.14 shows the arrangement of the pattern disk and the switch connected to a wheel of a robot.

The type of pattern disk that we have just discussed is non-directional because it cannot decipher the direction of movement. For directionality, a quadrature phase

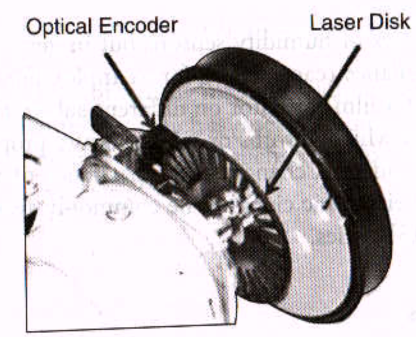


Figure 3.14 | Optical Encoder fixed to the wheel of a robot
(Reproduced with permission from Nex Robotics, Mumbai)

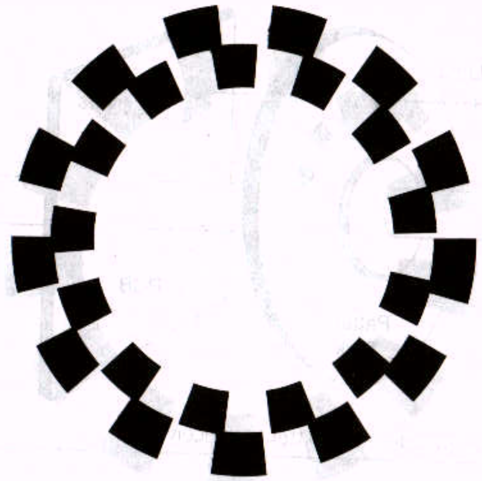


Figure 3.15 | A directional pattern

pattern with two staggered patterns is necessary, so that the system can tell which way the wheel is turning. Figure 3.15 shows such a pattern.

3.1.5 | Humidity Sensors

Humidity is the quantity of water vapour present in air. It can be expressed as being 'absolute' or 'relative'. Absolute humidity expresses the water vapour content of the air using the mass of water vapour contained in a given volume of air. Relative humidity is a ratio that compares the amount of water vapour in the air with the amount of water vapour that would be present in the air at saturation. Thus, humidity sensors measure the amount of water vapour present in air, but what are the applications for it?

In home automation systems, humidity is monitored so as to bring it to a level which makes it a comfortable environment—other applications are in the semiconductor industry where moisture levels need to be continuously monitored during wafer processing—in the household, it is used for intelligent control of living environment, microwave cooking, laundry, etc. In automobiles, this sensor information forms the basis for window de-fogging control.

There are different types of humidity sensors, but in general, sensing of humidity involves a change of impedance (capacitance, for example). For this principle, the sensor element is built out of a film capacitor on different substrates (glass, ceramic, etc.). The dielectric is a polymer which absorbs or releases water proportional to the relative environmental humidity, and thus changes the capacitance of the capacitor, which is measured by an on-board electronic circuit. One commonly available relative humidity sensor is the HS12P, HS15P series.

3.1.6 | Other Sensors

In this section, only a few types of sensors have been covered. Besides these, there are sensors for many other physical quantities—there are gas sensors, smoke sensors,

piezo-electric sensors (for sensing stress, strain etc), touch sensors and so on. As per the requirements of the applications, standard sensors for these can be easily sought out.

3.2 | Analog to Digital Converters

All sensors give an analog voltage proportional to the physical quantity sensed—to convert it to a digital number which an MCU can use, an Analog to Digital Converter (ADC) is used. Many MCUs have ADCs inside the chip (PIC, ARM, AVR, etc.), while there are MCUs like 8051 where an external ADC might be needed. In this section, we review some important aspects of ADCs.

3.2.1 | ADC Interfacing *

ADCs [Analog-to-Digital Convertors] convert analog voltages into digital codes which can be processed by embedded systems. ADCs are required for all systems that need to interface with real-world (analog) signals.

ADCs usually have two separate interfaces that are accessible to an embedded system.

i) Control interface
ii) Data interface

3.2.1.1 | *Control Interface*

ADCs vary widely in complexity, performance and speed. They have various modes and states which need to be managed to make them operate in the manner we need them to. For example, an ordinary SAR (Successive Approximation Register) ADC might continuously convert the input signal, generate codes and put them out on the data bus. However, the ADC might have different modes like 10-bit/12-bit/14-bit (resolution), various offset correction modes, power modes, latency modes (which determines how many clock cycles it takes for a given input to appear at the output), input clock modes (single ended/differential), etc. There might be a huge number of modes depending on the complexity of the ADC. Some might have elaborate schemes for tweaking various aspects of their operation to improve or tailor performance to our needs.

Register Control

All these modes and states are managed with the help of registers and register controlled fuses inside the ADC.

Hence, we need some way of writing into and reading from these registers. For this, we depend on the control interface which is essentially a register interface.

Various industry standards exist for such interfaces. A few among them are (Section 5.1.3)

i) SPI
ii) I2C
iii) UART

* The section 3.2.1 is written by Sabu Paul, Analog Design Engineer, Texas Instruments, Bangalore.

Pin Control

For very simple ADCs, there might not be enough number of states or modes to justify the usage of a register interface. They might simply have a few pins which control the state of the ADC. Typically, the number of such pins will be less than or equal to 4, which allows us to select between 16 different states. Some such devices might have a simple state machine inside which allows the ADC to respond to the sequence of inputs applied at the input pins. This is frequently used in ADCs used for compact biomedical applications. The ADC might be combined with an AFE (Analog Front End) and/or transceiver. The entire module can be controlled through the pin interface mentioned earlier.

3.2.1.2 | *Data Interface*

ADCs vary widely in their speed of conversion, from a few samples per second to several GSPS (Giga Samples per Second). In the case of very slow ADCs, separate data and control interfaces are not usually used. In such devices, the register interface is used to read out both the data and to write and read state information. For example, most microcontrollers contain a built-in ADC which has a register interface through which both control signals like start conversion/stop conversion as well as data can be sent and read out, respectively.

In the case of high speed ADCs, the transmission speeds of conventional register interfaces and their signalling overheads together make them unsuitable for data. Also, in most of the cases where high speed ADCs are used, control signals will have to be sent without interrupting the flow of data.

In such cases, a separate dedicated data interface is used.

Two types of interfaces used are:

i) Parallel
ii) Serial

Parallel Interface

In a parallel data interface, each data word is transmitted over several physical lines with each line carrying one data bit. There is also a clock line which is used to latch the data when it is ready. These systems are losing popularity and are used mainly in applications where the resolution and/or speed is low. In such cases, parallel interfaces make sense since they can be directly transmitted without a lot of digital manipulation by the ADC and directly read by the controller. This is unlike serial systems which require a SERDES (Serializer-Deserializer) for data transfer. A clearer picture of its pros and cons vis-à-vis the serial system will emerge during the discussion of serial systems.

Parallel interfaces can be classified based on the clock edge used for latching data.

i) **Positive edge:** The data is latched on the positive edge of the clock.
ii) **Negative edge:** The data is latched on the negative edge.
iii) **Dual edge:** The data is latched on the positive and negative edges of the clock. This is also known as a DDR [Dual Data Rate] scheme. This can reduce the number of data lines by half (at the cost of some extra hardware) or increase the speed by two.

Based on the number of lines used to transmit a single bit, we have

i) **Single ended:** A single line is used to transmit one bit and the voltage is referred to the common ground.

ii) **Differential:** A matched pair is used to transmit 1-bit. One line carries bit+V and the other bit -V. This improves noise immunity and can allow for reduced signal swing. A differential system requires twice the number of lines as a singled ended system.

Based on the voltage levels, these interfaces are classified into

i) **CMOS (1.8 -3.3V):** Modern ADCs have digital modules made using MOS technology. So, this voltage level is commonly used by a lot of ADCs for signalling.
ii) **TTL (5V).**
iii) **LVDS (700mV peak-to-peak):** Low Voltage Differential Signalling is used in differential systems. Both bit+ and bit- lines have a swing of 350mV peak-to-peak each. The noise immunity afforded by the matched pair of lines is what allows the usage of this reduced swing without compromising on BER (Bit Error Rate).

The LVDS scheme requires 2X (twice) the number of lines. To work around this problem, it normally is used in conjunction with the DDR scheme mentioned earlier. It is then called a DDR LVDS scheme. This allows us to have the exact same pin count while at the same time getting the reduced power consumption and EMI (Electro Magnetic Interference) levels of an LVDS scheme. The advantages of the LVDS scheme will become more apparent after the discussion of its use in serial systems. ICs are available which are capable of translating between voltage levels and signal modes (Differential/Single-ended).

Serial Interface

A lot of the modern day high performance devices use the serial mode of data transfer.

Here, instead of each data line carrying 1-bit of the data word, all the bits are sent serially over one single line. There are many advantages to this approach when compared to a parallel system.

Advantages

i) Fewer number of physical connections.
ii) Smaller die size made possible by reduced pin count.
iii) The size is a big advantage for some of the higher resolution (<14 bits)/multi-channel ADCs.
iv) Serial interfaces can operate differentially as the increase in the number of wires is minimal. This drastically cuts down noise pick-up and allows for lower signalling voltages.
v) The EMI generated by a balanced differential pair is much lesser than that by a large number of single ended parallel data lines. This is because the opposing currents in the pair cancel out the magnetic fields caused due to each other.
vi) The lower signal swing made possible by the common mode noise rejection in a differential system reduces power consumption.
vii) Serial interfaces allow for features like clock recovery. This makes it unnecessary to have a separate clock line. The clock is recovered from the data. This is possible because serial interfaces switch at a much higher speed than parallel buses.
viii) Quite a few high-quality serial interface standards are there which allow for easy interoperability.

Disadvantages

i) Requires a deserializer to convert the serial data to data words.
ii) Features like clock recovery require the data to be modified to make sure that there are sufficient number of transitions for the PLL (Phase Locked Loop) to lock on to and to ensure that there are an equal number of 1s and 0s in a specified number of bits to ensure that the input common mode of the receiver does not change.
iii) Parallel data has to be converted to serial data and a high speed clock generated inside the device.
iv) The speed of transmission is higher by a factor equal to the resolution of the ADC when compared to a parallel scheme. So, reliability of data transfer becomes very sensitive to clock jitter and board parasitics.

Serial interfaces are also classified into the following:

- **Single-ended:** There is only one data line and signal voltage is referred to the common ground. The zero crossings are susceptible to noise. But, the number of lines is less. Single ended interfaces come with different signalling levels. A couple of them are
  - CMOS (1.8-3.3V)
  - TTL (5V)
- **Differential:** Data is transmitted over a matched differential pair. This reduces noise and EMI generation. Signal swing can be lower because of improvement in noise. This can save power. These systems require special interfacing ICs to convert the differential signal into a single ended one for use in the embedded system. Differential systems are further classified based on voltage swing.
  - Rail-Rail Swing (e.g. USB)
  - LVDS (Low Voltage Differential Signalling 700mVpp)

Nowadays, high speed multi-channel ADCs are available which convert several input signals simultaneously into corresponding digital codes. They give out output codes serially over several output data channels. Sometimes, hybrid systems are used to get around speed limitations. These systems are mainly of the following two types:

- **Multiplexed output:** Data from several input channels is multiplexed onto one output channel to save on pin count.
- **Interleaved output:** Data from one channel can be split up into 2 or more streams which are then sent serially over multiple output data channels. This is done to support higher sampling speeds.

Conventional serial standards like USB and Firewire (Section 5.3) are not used in data converters. This is because ADCs are a special category of devices sending out one single type of data. Hence, the complications, overheads and limitations of these serial standards make them unsuitable for use in ADC interfacing. The JEDEC (*Joint Electron Devices Engineering Council)* JESD204 is the upcoming standard for serial data interfaces in high speed data convertors. This is an LVDS scheme supporting very high data transfer speeds over multiple synchronized lanes and from multiple ADCs.

The complexity of the interfacing, depends on the speed and resolution of the ADCs used. It varies from simple singled ended CMOS/TTL parallel interfaces to JEDEC SERDES interfaces. The cost also increases correspondingly.

3.2.1.3 | *Interfacing an ADC to 8051*

ADCs may be 'parallel' or 'serial', as we have just discussed. First we consider a parallel ADC.

3.2.1.4 | *An ADC with a Parallel Data Interface*

Our interest is to interface an ADC to an 8051 MCU using Port2 as the data lines, and some pins of Port 1 for the control signals needed by the ADC. When an analog voltage is given as an input to an ADC, it gets converted to a digital number which is transferred to the 8051. The digital value can be stored in the RAM of the system and may be displayed or used in further computations. See the block diagram of such a setup in Figure 3.16.

ADC 0808/0809

We choose here, the ADC0808/ADC0809 which is an 8-bit parallel ADC and is microprocessor compatible. Its functional pin diagram is shown in Figure 3.17. It is designated as an '8-Bit µP Compatible A/D Converter with 8-Channel Multiplexer'. It uses the successive approximation technique for analog to digital conversion.

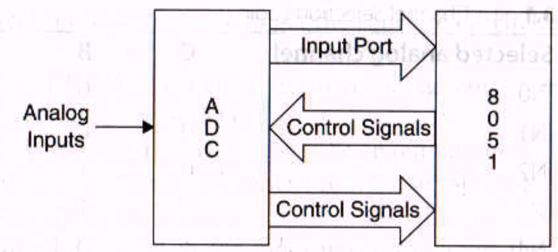


Figure 3.16 | General block diagram of the connection between an ADC and a MCU (8051)

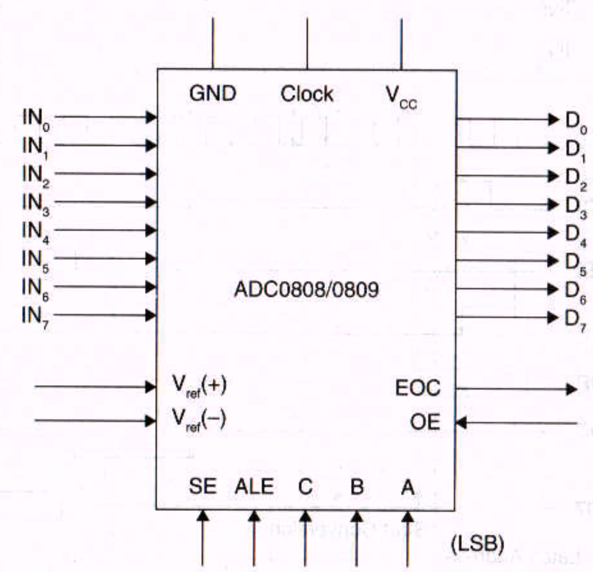


Figure 3.17 | Functional pin diagram of ADC 0808/0809

Its key specifications are given as:

1. Resolution 8 Bits
2. Total Unadjusted Error +/_ ½ LSB and +/_ 1 LSB
3. Single Supply 5 VDC
4. Low Power 15 mW
5. Conversion Time 100 μsecs

It is an 8-input multiplexed ADC, which means that it has 8 input analog signal lines, though only one of them can be operational at a time. This is selected by three address inputs A, B, C. Table 3. 1 shows the address bit configuration for selecting specific input channels. For example, if IN0 is to act as the input, the address lines C, B and A all have to be low; for IN1, the values of CBA is to be 001 and so on.

The first requirement in using the ADC is to select an input channel by giving the appropriate logic on the address pins. To latch this on to the chip, a signal called ALE (Address Latch Enable) is to be supplied on the ALE pin. ALE is to be a low to high transition (refer to Figure 3.18). After the address is latched and the analog input is

Table 3.1 | Channel Selection Logic

Selected analog channel	C	B	A
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	0	1
IN6	1	1	0
IN7	1	1	1

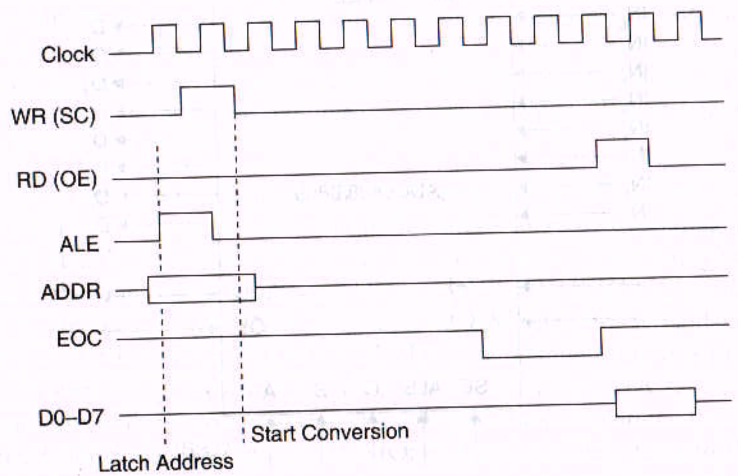


Figure 3.18 | Timing diagram for the ADC 0808/0809

available at the selected input line, the ADC must be signalled to start conversion. This (SC) is a low to high pulse of minimum specified duration (as mentioned in the data sheet). The ADC requires a clock and the speed of conversion depends on the clock rate. The maximum clock frequency is specified in the data sheet. The clock of the MCU can be divided to get the right frequency for the ADC.

The ADC takes a finite time to complete the conversion and then it notifies this fact by lowering the pin called EOC (End of Conversion). This should be brought to the notice of the MCU. The EOC signal can be used to interrupt the MCU, and allow the converted data to be transferred to the 8086 or this signal can be polled continuously.

To receive the digital data, the output lines of the ADC are to be activated. This is done by making high the line OE (Output Enable). Once, the output lines are activated, the converted digital data can be transferred to the 8051. The above is the sequence of actions necessary to use the ADC chip 0809 to perform analog to digital conversion and then to transfer the digital data to the microprocessor.

Let us now use the pins of 8051 for the purposes specified above. Make the connections between the 8051 and the ADC as shown in Figure 3.19. The salient points regarding the connection are as follows:

i) Port 2 is used in the input mode to get the converted digital data from the ADC to 8051.
ii) The port pins P1.7.P1.6 and P1.5 are used in the output mode as the address selection pins A, B, C of the ADC.
iii) The port pin P1.0 is used as ALE. It is to be an output pin.
iv) The port pin P1.1 is used to give the start conversion (SC) pulse to the ADC. Hence it is to be an output pin.
v) The port pin P1.2 is used in the input mode, to receive the End of Conversion (EOC) signal from the ADC.
vi) Port pin P1.3 is used as OE for the ADC. It is defined as an output pin.

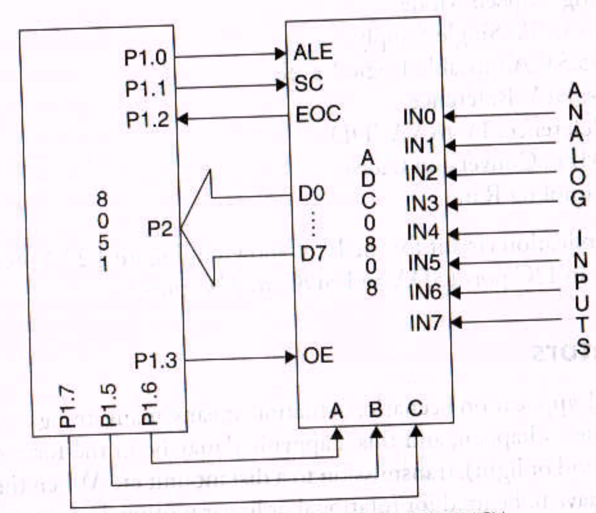


Figure 3.19 | Connections between the ADC and the 8051 MCU

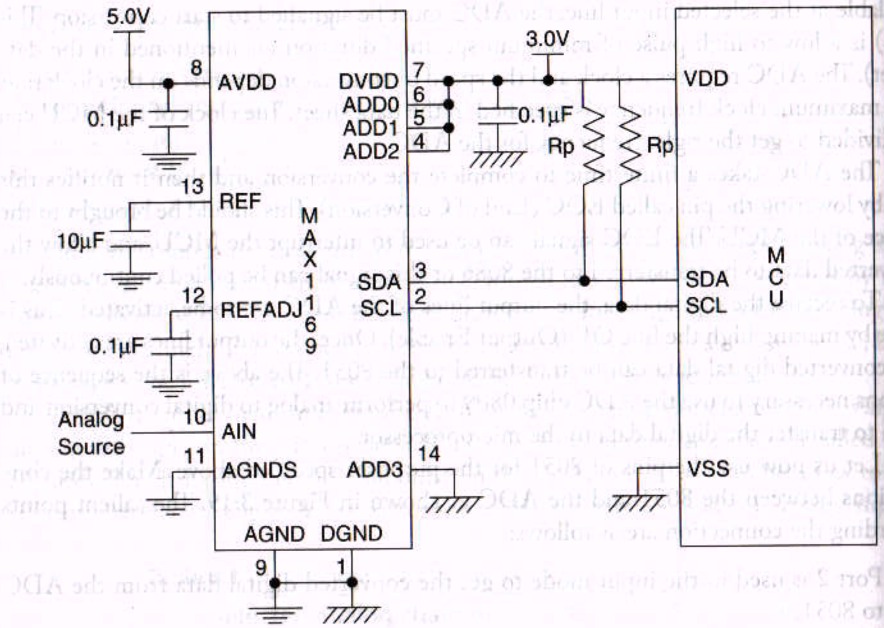


Figure 3.20 | Application circuit for the serial ADC MAX 1169 (Courtesy: Maxim Semiconductors)

3.2.1.5 | *Serial ADC*

An example for a serial ADC is the MAX 1169 which uses the I2C protocol (Section 5.2.1) for transfer. The specifications of this chip is given as

i) High-Speed I2C-Compatible Serial Interface
ii) 400kHz Fast Mode
iii) 1.7MHz High-Speed Mode
iv) +4.75V to +5.25V Single Supply
v) +2.7V to +5.5V Adjustable Logic Level
vi) Internal +4.096V Reference
vii) External Reference: 1V to VAVDD
viii) Internal 4MHz Conversion Clock
ix) 58.6ksps Sampling Rate

A typical application circuit for the IC is shown in Figure 3.20 Note that the MCU to be used needs an I2C port (SDA and SDC are I2C pins).

3.3 | Actuators

In the embedded application scenario, actuation means many things—it implies that something is made to happen, and this 'happening' may be in the form of a motion or display, alarm (sound or light), transmission to a distant unit etc. When the actuation is a 'motion', motors have to be used for rotational or linear motion. Let's start the discussion with display devices, related techniques and simple circuits where necessary.

3.3.1 | Displays

For most systems, some sort of display is necessary. Displays like LEDs, LCDs, etc. are very common. There is a lot of range and variety in the displays used in embedded systems.

Let's examine the features of some of the most popular displays.

3.3.1.1 | *Light Emitting Diodes (LED)*

A light emitting diode or LED as it is designated, works just as an ordinary semiconductor diode. It is usually made of Gallium Arsenide and is available in different colours. When it is forward biased, it conducts and also emits light which is used as a 'display unit'.

A single LED is used as an indication, of the state of a switch, the activation of any signal, reception of some data/signal, etc. It can also be made to act as an alarm by switching it on and off continuously at a certain rate. LEDs are easy to use and give a very bright and pleasing display which can be viewed equally well from any viewing angle, (unlike LCDs). The only drawback with LED displays is the high amount of current they need, unlike LCDs which need very low power. Figure 3.21 shows that a single LED can be connected to a positive power supply as shown. The value of the current limiting resistor depends on the current rating of the LED.

In case we need a number of LEDs for displays, we still use only one power source for all of them. In that case, they are connected together in either the 'common anode' or 'common cathode' LED configuration. In Figure 3.22a, the anodes of the three LEDs are connected together, and it is a 'common anode' connection. If we want to light up only the first and third LEDs, i.e., apply a '0' (i.e. ground), only at K1 and K3.

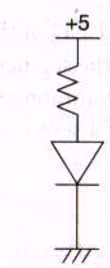


Figure 3.21 | A single LED circuit

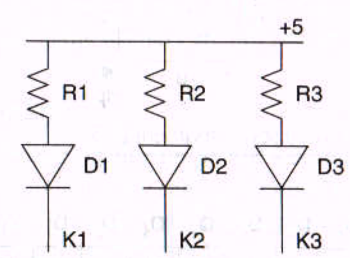


Figure 3.22a | Common anode connection

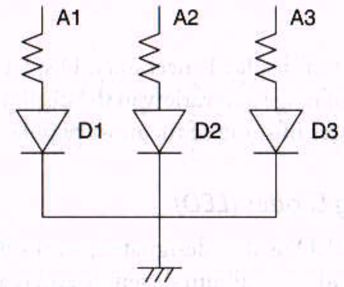


Figure 3.22b | Common cathode connection

In Figure 3.22b, which is 'common cathode', A1 and A3 alone should be given a '1' for the same result.

3.3.1.2 | *Seven Segment LED*

However, more important applications of LEDs are as alphanumeric displays. For that, seven segment LEDs are used, in which seven LEDs are arranged as the segments of a display arranged in a particular shape (see Figure 3.23a). When segments are selectively lighted up, the display of all alphanumeric characters is possible. By lighting up all the segments, we get '8' displayed. We can have one more segment in this display and it is for the decimal point. In Figure 3.23a, you can see that there are eight segments, including the segment for the decimal point. In spite of this, such displays are still designated as 'seven' segment displays. Such LED modules also can be used in either the common anode or common cathode configuration.

To light up a seven segment display LED of the common cathode type, ensure that the cathode is grounded, and give a '1' to the segments which are to be lit up. It is obvious that the opposite logic is to be used for common anode type.

Figure 3.23b shows the segments of a seven segment LED, arranged as a data byte D0 to D7.

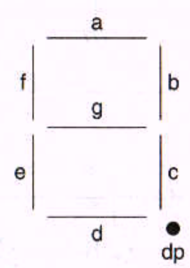


Figure 3.23a | The segments of a seven segment LED

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
dp	g	f	e	d	c	b	a

Figure 3.23b | The data byte for the seven segment LED

Example 3.1

a) Assuming a common cathode type of display, find the seven segment codes to be used for displaying
i) 8, ii) A and iii) b.

b) How will the code change of it is a common cathode display?

Solution

(a) Since it is a common cathode type, the common cathode of the LEDs of a digit is to be grounded. Then, supply the segment information to the anodes. For displaying 8, the segments to be lighted up are a, b, c, d, e, f, g. Since, we are using a common cathode type of display, the data to light up a segment is '1'. Hence, the bits from D0 to D6 is '1'.
Thus, the seven segment code for the display of 8 is 0111 1111, i.e., 7 FH.

i) Similarly for 'A' it is 0111 0111, i.e., 77 H.
ii) For 'b' it is 0111 1100, i.e., 7 CH.

(b) For common anode displays, the code for '8' is 1000 1000, for 'A' is 1000 1000 and for 'b' is 1000 0011.

3.3.1.3 | *Static Seven Segment Displays*

Now, suppose we want to use such a module to display a single character. Let us think of a scenario wherein, the number to be displayed is sent from an 8051 port to the display module. We just send the code (called seven segment code) corresponding to the segments of the LED. This code gives the information as to which of the segments are to be lighted up for the display of a specific character. In Example 3.1, if 77H is outputted through a port, the character 'A' is displayed on the segment LED connected to the port lines.

Assume we use only a one digit display module. If it is a common cathode type, we ground the common cathode and then we send the seven segment code directly to activate the required segments. This causes some of the segments to be ON and some to be OFF. Either way, as long as the display is ON, the module draws its required current from the power source. This may be in the range of 5 to 30 mA for a single segment to be lighted up. The display is ON all the time, and hence it is called a 'static' display.

Assume now that we need an eight digit display. If we use the same kind of static display, the current drawn is multiplied by 8, and this becomes quite a large amount. Multiply $7 \times 25 \times 8$ mA. This gives a value of 1.4 A, which is much too large for an electronic circuit. For this reason, static displays are not preferred for multiple digit displays.

3.3.1.4 | *Dynamic Displays*

When there is an array of digit display units, say 4 seven segment LEDs arranged in digit form in a row, a continuous display can be obtained by lighting up just one digit at a time. The next instant this digit is switched off and the next one is lighted up. This is done continuously and cyclically from digits 1 to 4 and repeated at a rapid rate. Because of the property of persistence of vision of the eyes, an illusion of continuous display is obtained. This is also called a multiplexed display.

The important points to note here are:

i) The common anode/cathode of a digit is to be activated for a digit to be active.
ii) At a time, only the segments of one digit are 'ON'.
iii) After a specified delay, this digit is switched off and the segments of the next digit are ON. The information displayed here is different from that of the previous case.
iv) Thus, for display multiplexing, consecutive digits should be switched on in a cyclic fashion, and for each digit, the segment information should be supplied.

Now, let us use this concept in a system in which an 8051 handles a dynamic display (Figure 3.24). This is an 4-digit display, of the common cathode type. The ports of 8051 are used in such a way that Port 1 supplies the digit information and Port 2 supplies the segment information. Digit information through Port 1.0 to P1.3 is to select which digit is being activated at a particular time. For segment information, the seven segment code of each digit should be sent as a byte through Port 2.

Figure 3.25 shows the complete set up. Four pins of port 1 are used for 'digit driving'. These pins are connected to the bases of the four transistors Q1 to Q4. At

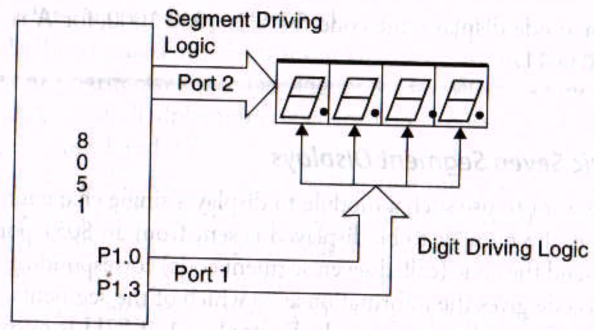


Figure 3.24 | A dynamic display for an 8051-based system

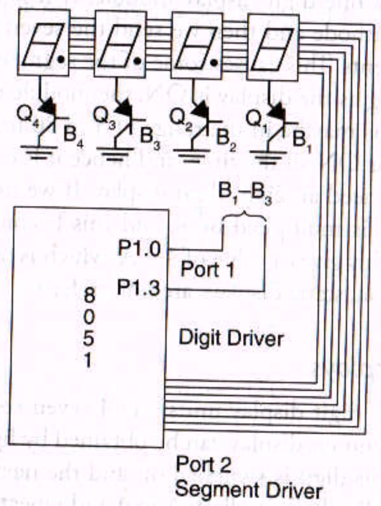


Figure 3.25 | A four digit dynamic display using the 8051.

time, only one particular transistor is to be ON. These are PNP transistors and are turned ON if a '0' is applied to the bases. This '0' goes to the emitters of the transistor which is connected to the common cathode of the segment LEDs, of a particular digit. At a time, Port 1 gives a '0' only on one of its port lines. To understand this clearly, observe Figure 3.25. The most significant digit (or the left most digit of the display) is activated by a '0'on pin P1.0. When this pin is cleared, P1.1 to Pi1.3 should be set. At the same time, the segment information for displaying the left most digit should be placed on Port 1. If Port 2 gives the data 77 H, the first digit displays 'A'.

This technique is to be repeated for all digits continuously. The steps are:

1. Select the first digit to be displayed and send a suitable logic through P1.0 to P1.3 to activate a digit.
2. Send the segment code through Port 2.
3. Call a delay of say, 3msecs.
4. Repeat this sequence for all four digits.
5. Then start again from the first step.

With 4 digits and 3 msecs delay, we can get back to the first digit every 12 msecs. This corresponds to a refresh rate of around 83 times per second, which is sufficient to fool the eye into believing that all the digits are ON at the same time [The persistence of human vision is $(1/16)^{th}$ of a second–(62.5 msec)].

3.3.1.5 | Organic LED (OLED)

This is a relatively new type of display which has gained acceptance in mobile phones, PDAs, digital media players, cameras and similar portable applications. OLED-based TVs are also making their foray into the consumer market.

The physical structure of an OLED consists of a layer of organic material (which is emissive electroluminescent) is sandwiched between two conductors (an anode and a cathode), which in turn are sandwiched between a glass top plate (seal) and a glass bottom plate (substrate). See Figure 3.26. When electric current is applied to the two conductors, a bright, electro-luminescent light is produced directly from the organic material. There are two types of OLEDs- small molecule OLED, and polymer OLED. The small molecule type is considered to have a longer lifespan. The OLED primary colour matrix is arranged in red, green and blue pixels, which are mounted directly to a printed circuit board. It expresses pure colours when an electric current stimulates the relevant pixels.

Thin organic layers serve these displays as a source of light, which offers significant advantages in relation to conventional technologies. The nature of its technology lends itself to extremely thin and lightweight designs, which makes its application domain very wide. To list out a few plus points of OLEDs:

i) Unlimited viewing angle
ii) Low power consumption
iii) Fast 'response time'
iv) Brighter and more brilliant picture

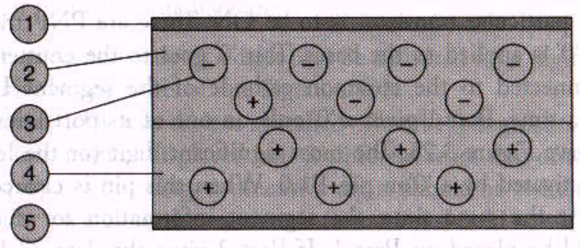


Figure 3.26 | Schematic of a bilayer OLED: 1. Cathode 2. Emissive Layer 3. Emission of radiation 4. Conductive Layer 5. Anode

When comparing it with standard display technologies, the notable points are that

i) They require no backlighting as for LCDs because they are emissive devices.
ii) The fabrication process is easy, and devices are thinner and lighter than those fabricated by cathode ray tube (CRT) display technology.

3.3.1.6 | *Liquid Crystal Displays (LCDs)*

Liquid crystal displays called LCDs are very popular, with their qualities of low power dissipation and ease of use. The only problem normally encountered is the problem of the viewing angle. The display is not equally clear at all viewing angles. They are available as character LCDs for displaying ASCII characters, and as graphical LCDs which contain display elements as dots or pixels which can be selectively illuminated, so as to display any pattern.

Character LCD

Character LCD modules of many different specifications (mostly differing in the number of lines, number of characters per line and so on) are available. An LCD module has registers, writing into which the display can be easily programmed and controlled. Here, we will discuss a 16 × 2 character LCD which looks as shown in Figure 3.27.

Pins of the LCD The LCD that we have selected, has 16 pins as shown in Table 3.2. We see that DB0 to DB7 correspond to the data pins. The others are the pins for control signals and the power supply. VEE is a pin used to adjust the contrast of the display. It is usually connected to a potentiometer, so that contrast can be adjusted. Besides that, there

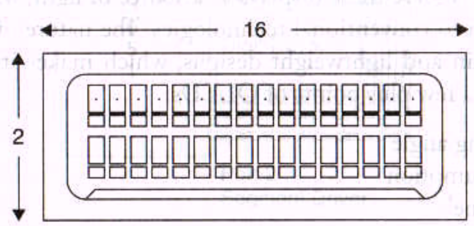


Figure 3.27 | A 16 x2 character LCD module

Table 3.2 | Pins of the 16 × 2 LCD Module

No	Symbol	Function
1	Vss	Power supply ground (0V)
2	Vcc	Power supply (5V)
3	VEE	Power supply for adjusting contrast
4	RS	Register select signal
5	R/W	Read write select signal
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	AV_{EE}	Positive voltage for back light
16	K	0V for back light

are the VCC and ground pins. There are two pins for backlight adjustment, if necessary. Backlighting means extra lighting behind the LCD panel (usually LEDs) so that the display is visible in the dark also.

- **RS-Register Select:** This pin selects between a command register and a data register. RS = 0 corresponds to the command register and RS = 1 to the data register. The data to be displayed is to be sent to the data register.
- **R/W-Read/Write:** This pin allows the user to write to or read from the display. When there is no necessity for reading the display, this pin is grounded. If both reading and writing is required, this pin is made programmable.
- **E-Enable:** The enable pin has to be given a high to low pulse, which is maintained high for at least 450 ns (may be different for other LCD modules).
- **DB0 to DB7:** These are the data pins of the LCD. Data to be written is to be sent through these pins, and data to be read will be received from the LCD through these pins. The data to be written for display are sent as ASCII characters. For writing into the command registers, there are predefined codes for the LCD. The codes for the specific LCD considered here, are given in Table 3.3.
- **Busy Flag:** It is seen that there is a minimum time required to latch one data on the LCD and get it displayed. Suppose we want to give a new data for display, the simplest way would be to introduce a small delay between sending the two display data (which can be given only one character at a time). However, another method for sending consecutive characters is to check what is called the 'Busy' flag of the LCD. For testing the busy flag, make RS = 0 first. The 'Busy' flag is DB7 and can be read

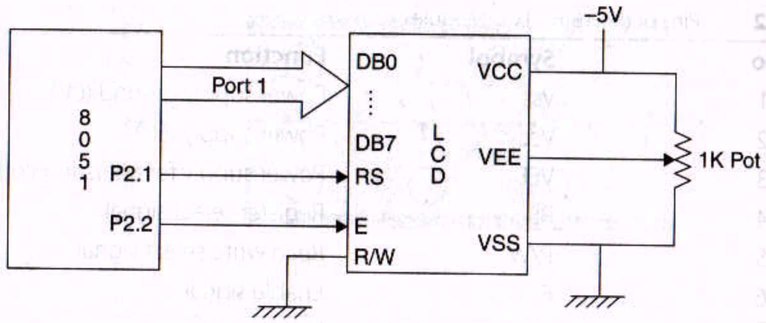


Figure 3.28 | Connecting the LCD module to the pins of 8051

when R/W = 1 and RS = 0. If DB7 is found high, it means that the LCD is busy doing its operations, and will not accept any new information. Keep checking this flag until it is low. Then, the next data can be written to it.

Note If the busy flag is to be read, the R/W pin has to be made programmable.

Now, let us do some display activities using a 16 × 2 LCD. Data and commands are sent from the ports of 8051. See Figure 3.28. Port1 is used as the data lines, and pins P2.1 and P2.2 are used for RS and E.

The connections are done as follows. Refer to Figure 3.28.

i) VSS and R/W is connected to ground.
ii) VCC is connected to 5 V supply.
iii) VEE is connected through a 10 K pot to the supply for contrast adjustment.
iv) RS is connected to P2.0 and E is connected to P2.1.
v) Pin 7-14 (DB0 to DB7) of the LCD module are connected to Port 1 of 8051.
vi) Pins 15 and 16 of the LCD are used for backlight adjustment (not shown in Figure 3.28).

Backlight There is a lamp here instead of reflected light. If backlighting is provided by LEDs as in the case of many 16 × 2 LCDs, connect pin 16 to ground, and pin 15 to Vcc through a 100 Ω resistor.

Getting a Character Displayed on the LCD To display characters on the LCD, the ASCII value of the character should be sent to the data register. But before sending the data, appropriate control signals should be activated by giving the required logic levels on the port pins. Also, first the LCD is initialized, then cleared, and then the cursor is positioned. This is done by sending command words to the LCD command register. (Refer Table 3.3).

Algorithm

i) Send command word to Port 1. The command word are 38 H (initializing LCD), 0EH (making the LCD and cursor ON), 01 (clearing the screen), 06 (shifting the cursor right) and 81 H (moving the cursor to line 1, position 1).

Table 3.3 | Command Codes of the LCD

Code (Hex)	Command
01	Clear display screen
02	Return home
04	Shift cursor left (decrement cursor)
05	Shift display right
06	Shift cursor right (increment cursor)
07	Shift display left
08	Display off, cursor off
0A	Display on, cursor on
0C	Display on, cursor off
0E	Display on, cursor blinking
0F	Display off, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80*	Force cursor to the beginning of the fi rst line
C0*	Force cursor to the beginning of the second line
38	2 lines and 5 × 7 matrix

* For a 16 × 2 line display, the addresses of the cursor positions are 80 to 8F for the fi rst line, and C0 to CF for the second line

ii) Make RS = 0 (by clearing P2. 1) for selecting the command register.
iii) Make R/W = 0 to write to LCD (if this line is grounded as in Figure 3.28, this step can be skipped).
iv) Send a H to L pulse at the E pin to complete the writing. For this, make P2.2 high for a short while and then clear it.
v) With this, the writing of commands is over. Now the required data must be written.
vi) Make RS = 1 for selecting the data register.
vii) Repeat steps 3 and 4.

In this setup (Figure 3.28), one whole 8-bit port was used up for LCD data. To save on pins, it is possible to use LCDs with just 4 data pins of a port. The data and command words are sent as in the previous case, but the method here is to send the 8 bits as two nibbles—thus only four lines of an MCU port need to be used. See Figure 3.29, which shows that only 7 port pins are needed in total, to connect an LCD module to an MCU.

Graphical LCD (GLCD)

Character LCDs have their limitations in that they can display characters only. Graphical LCDs are currently used to display customized characters and images. Graphical LCDs

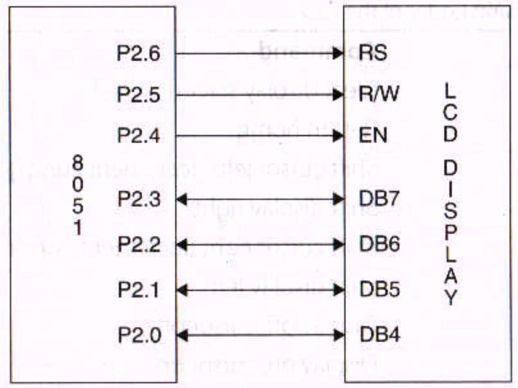


Figure 3.29 | A 4-bit LCD interface

find use in many applications; they are used in video games, mobile phones, etc. as display units. The customization is possible because the LCD has dots or pixels which may be selectively lighted up to generate the display we need. Thus, their size is specified as MxN dots or pixels.

Such LCD displays come in a variety of sizes, ranging from 32 × 80 to 240 × 320 dots/pixels. Larger displays offer more display area, cost more and take longer to refresh the entire screen with new data.

Graphical LCD modules come with inbuilt controllers which will allow us to interface the display with an MCU, for selectively lighting up the required pixels. Figure 3.30 shows the picture of a 128 × 64 dots graphical LCD manufactured by 'Vishay', with built-in controllers which are two ICs of KS0108 or equivalent. Table 3.4 gives the pin functions of this LCD display module.

Two controllers are needed because the display is split logically as half-left half and right half. It then needs two controllers with IC1 (Chipselect1) controlling the left half of the display and IC2 (Chipselect2) controlling the right half. Each controller must be addressed independently. Each half consists of 8 horizontal pages each of which is 8 bits (1 byte) high. The page addresses, 0-7, specify one of the 8 pages. That is illustrated in Figure 3.31.

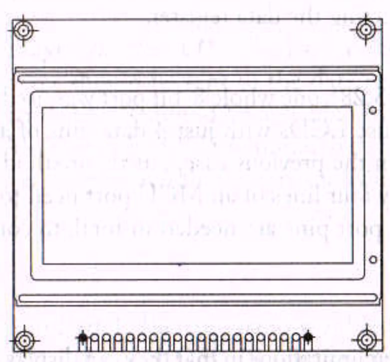


Figure 3.30 | A 128 × 64 dots (pixels) graphical LCD

Table 3.4 | Pin functions of the LCD module of Figure 3.30

Pin Number	Symbol	Function
1	Vss	GND
2	Vdd	Power Supply (+5V)
3	Vo	Contrast Adjustment
4	D/L	Data/Instruction
5	R/W	Data Read/Write
6	E	H L Enable/Signal
7	DB0	Data Bus Line
8	DB1	Data Bus Line
9	DB2	Data Bus Line
10	DB3	Data Bus Line
11	DB4	Data Bus Line
12	DB5	Data Bus Line
13	DB6	Data Bus Line
14	DB7	Data Bus Line
15	CS1	Chip Selector for IC1
16	CS2	Chip Selector for IC2
17	RST	Reset
18	Vee	Negative Voltage Output
19	A	Power Supply for LED (4.2V)
20	K	Power Supply for LED (0V)

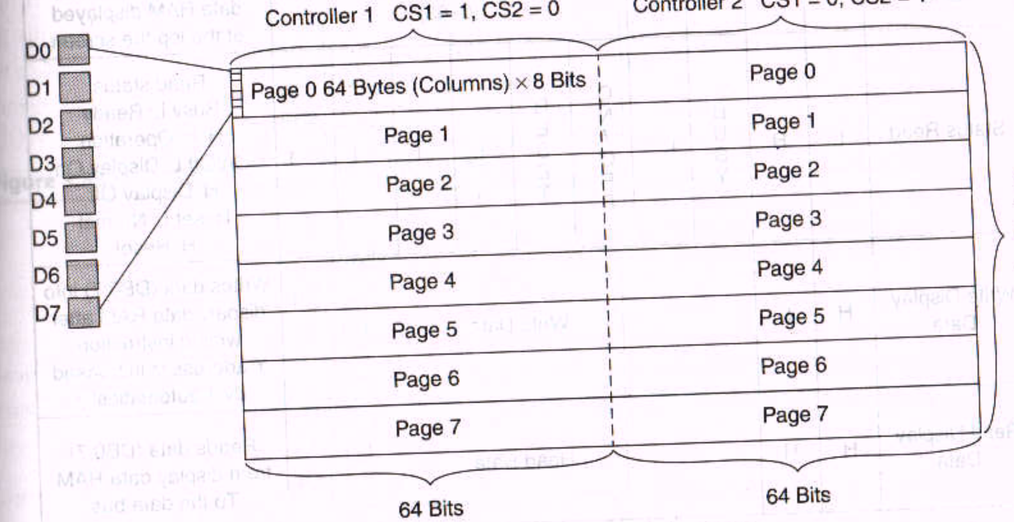


Figure 3.31 | Division into two halves, and the vertical pages of the GLCD

The Controller IC The KS0108B is a LCD driver with 64 channel output for dot matrix liquid crystal graphic display system. This device consists of the display RAM, 64-bit data latch, 64-bit drivers and decoder logic. It has the internal display RAM for storing the display data transferred from an 8-bit MCU and generates the dot matrix LCD driving signals corresponding to stored data.

Commands of the Controller The following are the KS0108 commands. See Figure 3.32.

- **Y address (0 to 63):** The Y address counter designates address of the internal RAM. An address is set by instructions and is increased by 1 automatically by read or write operations of display data. Y address 0 is the leftmost byte, and Y address 63 is the rightmost byte of a page.
- **X address (0 to 7):** This is the page address and has no count function.
- **Display line (0 to 63):** The display start line register specifies the line in RAM which corresponds to the top line of LCD panel, when displaying contents in display data RAM on the LCD panel. It is used for scrolling of the screen.

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display On/Off	L	L	L	L	H	H	H	H	H	H	Controls the display on or off internal status and display RAM data is not affected
Set Address	L	L	L	H	Y Address(0-63)						Sets the Y address in the Y addresss counter
Set Page (X Address)	L	L	H	L	H	H	H	Page (0–7)			Sets the X address at the X address register
Display Start Line	L	L	H	H	Display Start Line (0–63)						Indicates the display data RAM displayed at the lop the screen
Status Read	L	H	BUSY	L	ON/OFF	RESET	L	L	L	L	Read status Busy L: Ready H: In Operation On/Off L: Display On H: Display Off Reset L: Normal H: Reset
Write Display Data	H	L	Write Data								Writes data (DB0:7) into dispaly data RAM after writing instruction Y address is increased by 1 automatically
Read Display Data	H	H	Read Data								Reads data (DB0:7) from display data RAM To the data bus

Figure 3.32 | Commands of the controller

With this information, and an in-depth reading of the data sheets of the controller, it should now be possible for you to interface graphical LCD to an MCU like 8051, PIC, etc.

The algorithm for use is similar to that of the character LCD, i.e., send the initialization signals as commands, and then send the data.

How to light up pixels selectively?

Figure 3.33a shows the left half of the display (for one column only). There are 8 pixels in one column and there are eight such columns, one for each page. The total number of pixels in a column is numbered as P0 to P63.

Figure 3.33 b shows the eight pixels (named P0 to P7) corresponding to Page 0 of Figure 3.33a Assume we want to light up P0, P1 and P7 of this column. The data for this is to be sent as a byte. It is 1100 0001, i.e., 0xC1. Now for the next page (for pixels P8 to P15) if all the pixels except P8 and P15 are to be lighted up, the data is 0111 1110, i.e., 0x7E. This is continued for all the eight pages (one column of the display).

Thus, we write data (in bytes) for all the pixels of one column, and then go on to the next column. This is done first for the left part of the display, and then for the right half. The display will then appear to start from the left, and move to the right. We see that for any pattern, we have to generate a bit map and load it into the display RAM of the LCD controller.

Connecting a Graphical LCD to an 8051 MCU

Figure 3.34 shows the connections between a graphical LCD and 8051. Note that the connections are similar to Figure 3.28 but here two extra connections are for CS1 and CS2.

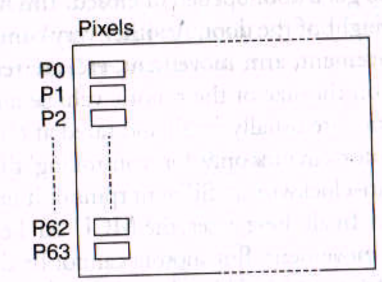


Figure 3.33a | Pixels in one column of the GLCD

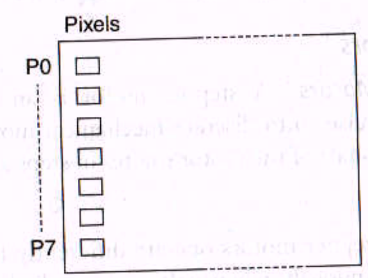


Figure 3.33b | Pixels of Page 0

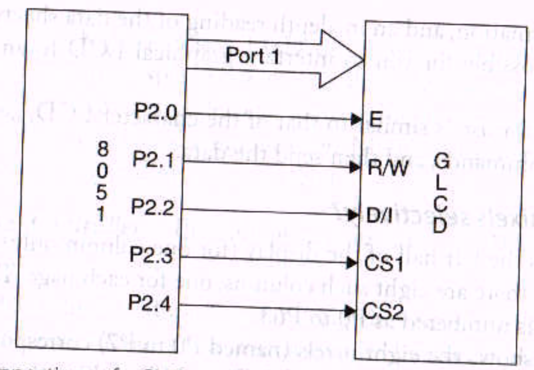


Figure 3.34 | Connections of a GLCD to 8051

D/I is the pin which selects between data and commands (similar to pin RS in Figure 3.28). Pins 19 and 20 of the LCD module are for backlighting, and are not shown in Figure 3.34.

Note Appendix H contains the program for interfacing this graphic LCD to PSoC3. The program is also given.

3.3.2 | Motors

Motors are used for rotational motion, which can be converted to linear motion when the application calls for it. In embedded systems, the rating (voltage, current, torque, etc.) of the motor to be used, depends on the application. For example, we might use a motor in a home security system to get a door opened or closed. This might require a heavy duty motor depending on the weight of the door. Another very common application is hobby robotics where vehicle movement, arm movement, etc. are required. The rating of the motor to be used depends on the size of the robotic vehicle and its activity. The motors used by hobbyists for robotics are usually small and rated at 6 to 12 V supply.

MCUs are used in motor circuits only for 'controlling' the motor. Motors may be made to rotate clockwise/anti-clockwise at different rpms or it may be that a movement by a small angle alone is needed. In all these cases, the MCU can be programmed to generate the driving logic for motor movement. But motors cannot be driven directly by a MCU because the current output from an MCU is relatively small. So there should be arrangements to get higher driving currents, and additional circuitry is usually necessary. We will examine all such aspects for two types of motors—stepper motors and DC motors.

3.3.2.1 | *Stepper Motors*

Introduction to Stepper Motors A stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical movements. When electrical pulses are applied to it, the shaft of the motor rotates in steps and this type of movement gives the motor its name.

Principle of Operation Stepper motors operate differently from normal DC motors. A DC motor rotates continuously when voltage is applied to its terminals. Stepper motors, have multiple toothed electromagnets arranged around a central gear-shaped

piece of iron (see Figure 3.35). The electromagnets are energized by an external control circuit which sends pulses to the motor.

To turn the motor shaft, one of the electromagnets is given power first, which makes the gear's teeth magnetically attracted to the electromagnet's teeth. When one tooth of the gear is thus aligned to the energized electromagnet (Electromagnet 'B' and tooth '6' are aligned in Figure 3.35), others are slightly offset from the corresponding electromagnets. When the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one, and from there the process is repeated. Each of those slight rotations is called a step, with an integral number of steps making a full rotation. In this way, the motor can be turned by a precise angle.

The rotation of the motor is related to the sequence of the input pulses:

i) The order in which a particular sequence is applied, decides the direction of rotation (clockwise or anti-clockwise).
ii) The speed of stepping depends on the frequency of the pulses applied, i.e., higher the frequency, faster the stepping motion.

We can use stepper motors for movement which needs to be finely controlled. The fine control is obtained because these motors move in steps, and the steps can be quite small in size. For example, one step can be 2 degrees and, for one complete (360 degree) rotation, 180 steps are obviously needed. For one such step, the motor needs to get a 'pulse' from a control circuit. In order to obtain a 90 degree rotation for such a motor, we must write a program to supply only 45 pulses to it.

This type of stepping motion can be used to advantage when it is needed to control aspects such as rotation angle, speed, position and synchronism. As such, they are used in applications such as printers, plotters, high end office equipment, hard disk drives, medical equipment, fax machines, and automotive and industrial applications where precise and controlled rotation is required. Robotics is another area where it is used for precise and controlled motion.

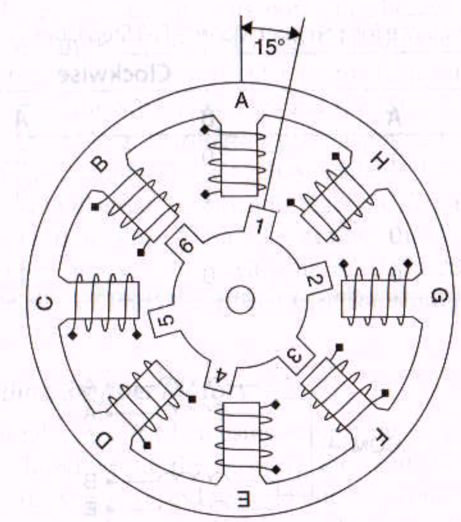


Figure 3.35 | Figure showing the operating principle of a stepper motor

Driving a Stepper Motor

Full Step Drive (two phases on) This is the usual method for full step driving of the motor. Both phases are always ON. The motor will thus have the full rated torque. This is achieved by the sequence of ones and zeros as shown in Table 3.5 which is to be repeatedly applied. (Note the presence of two '1's in each row of the table corresponding to 'two phases' being ON at any time.) Reversing the order in which the sequence is applied gives anti-clockwise rotation.

In short, for clock wise rotation, the sequence to be applied repeatedly is 09, 0CH, 06, 03. For anti-clock wise rotation, it is 03, 06, 0CH, 09. If you read about stepper motors from any other source, there is a possibility that you might see another 'sequence' being suggested. Neither that nor this is wrong. The 'sequence numbers' just depend on the way we have named the phase windings. There are four wires available at the output of any stepper motor winding and they are named A, $\overline{A}$, B and $\overline{B}$ corresponding to Figure 3.36.

Wave Drive In this drive method, only a single phase is activated at a time. It has the same number of steps as the full step drive, but the motor will have significantly less than rated torque. This sequence is 8, 4, 2, 1 for clockwise and 1, 2, 4, 8 for anti-clockwise rotation. Table 3.6 is the driving sequence for this.

Half Stepping When half stepping, the drive alternates between two phases ON and a single phase ON. This increases the angular resolution, but the motor also has less torque at the half step position (where only a single phase is on). The advantage of half stepping is that the drive electronics need not change to support it. The step-angle is half that of the previous two cases—thus the stepping resolution is increased. For anti-clockwise rotation, the order of the above sequence should be reversed. Table 3.7 is the driving sequence for this.

Table 3.5 | Driving Sequence for a Stepper Motor—Full Step Drive

	Clockwise			
Step No.	**A**	**B**	**$\overline{A}$**	**$\overline{B}$**
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

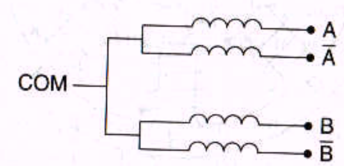


Figure 3.36 | Stepper motor windings

Table 3.6 | Driving Sequence for a Stepper Motor—Wave Drive

	Clockwise			
Step No.	**A**	**B**	**$\overline{A}$**	**$\overline{B}$**
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Table 3.7 | Driving Sequence for a Stepper Motor—Half Stepping

	Clockwise			
Step No.	**A**	**B**	**$\overline{A}$**	**$\overline{B}$**
1	1	0	0	1
2	1	0	0	0
3	1	1	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	1	1
8	0	0	0	1

Using an 8051 to Interface a Stepper Motor

We can run a motor using a sequence generated by an MCU, say 8051. The motor, however, cannot be driven directly from its port pins, because the motor requires a current much more than can be supplied by the MCU. (The exact current requirement depends on the specifications of the particular motor being used.) As such, current drivers are needed between the 8051 port lines and the leads of the motor. Transistors with high current capability (e.g. Darlington pair or power transistors) can be used. Besides this, there are special motor driving ICs available. One such IC is the ULN 2003 driving IC whose pin diagram is shown in Figure 3.37. This IC contains an array of seven Darlington pair transistors. Figure 3.38 shows the 8051 MCU generating a sequence for energizing a stepper motor, with the IC ULN 2003 being used to raise the current level. Four pins of Port1 have been used for sending the driving sequence to the motor.

Other Issues Regarding Stepper Motors

An important issue to take care of when using stepper motors is that there is a chance of back emf being produced during the de-energization of the coils. This can damage the circuits producing the sequence and hence diodes are connected which block these spikes. Such diodes are called by various names as flywheel, fly back, free wheeling or snubber diodes (Section 3.3.4.2) When discrete Darlington pair transistors are used for

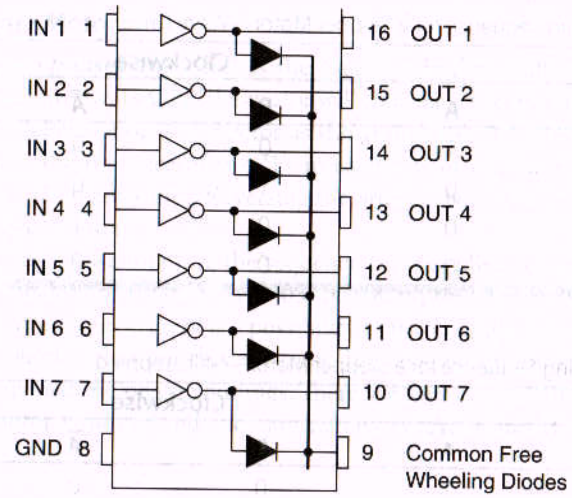


Figure 3.37 | Functional pin diagram of the driver, IC ULN2003

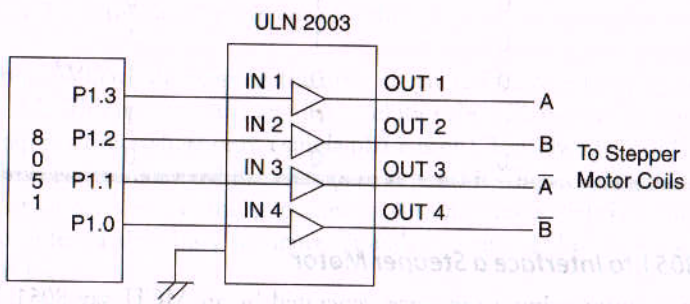


Figure 3.38 | Connections between 8051 and the stepper motor through a current driver

producing the high current required to drive the motors, diodes are connected to block this back emf. Such a diode is also in-built within the motor driving IC ULN 2003. Note the diode connected at pin 9, which is the freewheeling diode, corresponding to all the seven Darlington transistors inside the chip.

3.3.2.2 | *DC Motors*

This is a type of motor which operates on direct current and is very commonly used in embedded systems, when continuous movement is needed. The movement may be made 'controlled', in the sense that the speed and direction can be changed as per the requirements of the application. Robotics is an area where DC motors are widely used, but this is not the only application. Any type of movement is possible to be achieved with dc motors.

The DC motor has two basic parts: the rotating part that is called the armature and the stationary part called the stator that includes coils of wire called the field coils.

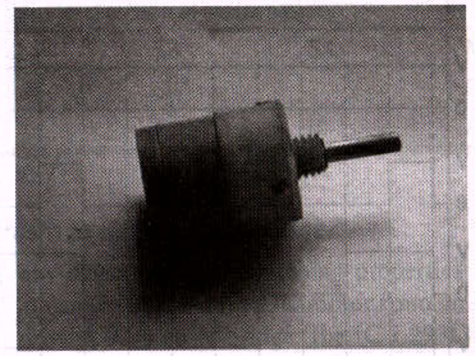


Figure 3.39 | Photograph of a small DC motor

The armature is made of coils of wire wrapped around the core, and the core has an extended shaft that rotates on bearings. The ends of each coil of wire on the armature are terminated at one end of the armature. The termination points are called the commutator, and this is where the brushes make electrical contact to bring electrical current from the stationary part to the rotating part of the machine. Figure 3.39 is a photograph of a light weight DC motor.

Characteristics of DC Motors

DC motors are non-polarized; this means that its power supply voltage can be reversed. The characteristics of a DC motor that we use in applications are as follows:

Speed Varies with Applied Voltage This feature is important for running a motor at different speeds. This can be done by increasing or decreasing the power supply voltage. But when we use electronic control, PWM (pulse width modulation) is the method for varying motor speed.

The method is to apply a pulse train to the power terminals of the motor. The average voltage obtained at the terminals is then proportional to the duty cycle of the pulse train, which is proportional to the speed of rotation (rpm) of the motor. Thus, as the duty cycle is increased, the motor rpm increases and vice versa. When the power supply is constant, it runs at 100% of its power rating (at no load). As the duty cycle reduces, the speed and the power reduce. Figure 3.40 shows pulse trains of various duty cycles.

When it is necessary to do speed control of DC motors for embedded applications, an MCU can be made to generate the PWM waveform based on some criterion, or depending on sensor output values. Many MCUs have a PWM unit as an integrated peripheral—the user just needs to use a few registers to specify the pulse repetition time (T) and the duty cycle. The 8051 does not have PWM unit—but such a waveform can be generated easily by a simple program.

Torque Varies with Current The torque of a motor is the rotary force produced on its output shaft. Torque increases with increased current, which means that it increases with increase in power supply voltage.

Reversal of Polarity of the Supply Voltage Causes Reversal of Direction of Rotation This aspect is very important in many applications, especially in robotics, when the motor

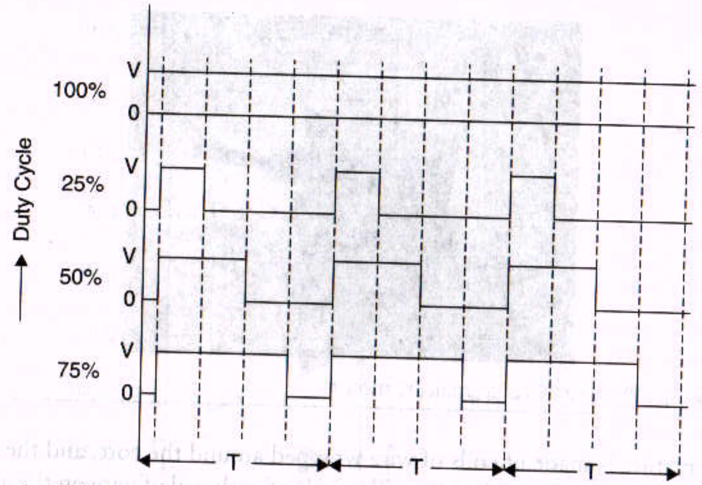


Figure 3.40 | PWM waveforms at various duty cycles

needs to reverse its direction of rotation. For example, a robotic vehicle will have to change from forward motion to reverse motion when an obstacle comes in its path. To do this dynamically, some sort of controlling switch is necessary, and this is available in the form of the H bridge.

3.3.2.3 | *H Bridge*

The H bridge is so named because it has four switching elements at the limbs of the H and the motor forms the cross bar. Figure 3.41 shows the 'idea' of the H bridge. There are two switches at the top (left and right) and two more switches at the bottom. They are named S1, S2, S3 and S4.

When the motor is not expected to rotate, all the switches are to be kept open. When switches S1 and S4 alone are closed, the motor rotates in the clockwise direction, with switches S2 and S3 closed, the rotation is anti-clockwise. In the positions when the top two switches and/or the bottom two switches are closed, the motor gets short circuited and such a situation should not be allowed.

The valid states of the switch are shown in Table 3.8, assuming that activation by a '1' corresponds to a switch closure.

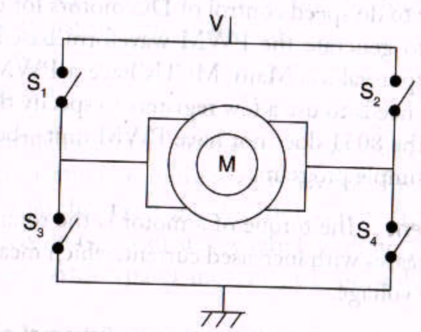


Figure 3.41 | The principle of operation of an H bridge

Table 3.8 | Switch status for direction of motor rotation

S1	S2	S3	S4	Motor Rotation
1	0	0	1	Clockwise
0	1	1	0	Anti-clockwise

What is the mechanism to realize an H bridge?

It can be done using any device that has switching properties like relays, transistors, MOSFET, etc. But if you are trying to run a DC motor from an MCU output, the best bet would be a motor driving IC with H bridge. The IC L293D is a dual H bridge IC which also provides sufficient current to drive a small motor.

The L293D IC whose pin configuration (shown in Figure 3.42) is a dual H Bridge motor driver. With one such IC, two DC motors can be driven which can be controlled in both clockwise and counter clockwise directions.

For applications that don't need reversal of direction, the four output pins can be used for driving four separate motors. This IC is rated for an output current of 600 mA and peak output current of 1.2A per channel. Moreover for protection of the circuit against back EMF, snubber/flywheel diodes (Section 3.3.4.2) are included within the IC. A simple schematic for interfacing a DC motor using L293D is shown in Figure 3.43. Refer Table 3.9 for the status of A and B.

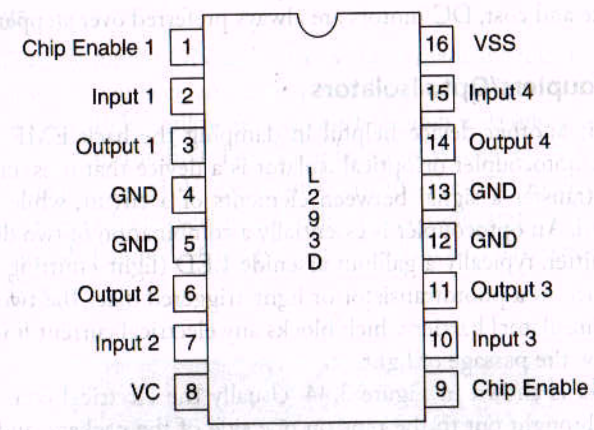


Figure 3.42 | Pins of the L293D motor driver

Table 3.9 | Action Performed for the Four Combinations of A and B

A	B	Action Performed
0	0	Motor is in the stop/brake condition
0	1	Motor rotates anti-clockwise
1	0	Motor rotates clockwise
1	1	Motor is in the stop/brake condition

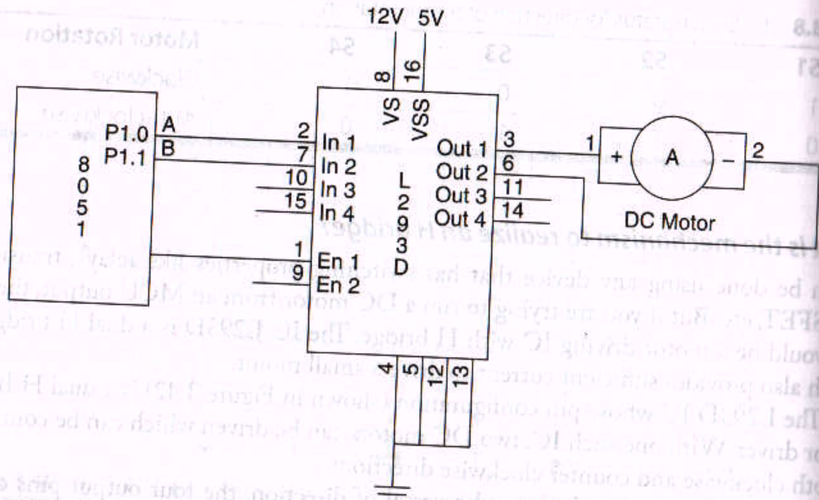


Figure 3.43 | Connections between an 8051, an H bridge and a DC motor

Three pins of the chip are needed as inputs from the MCU. The enable pin has to be set, and the pins A and B are to be controlled by the port lines P1.0 and P1.1, which generates the necessary logic to get the motor to rotate as required.

Embedded applications may use either stepper or DC motors. But when it comes to speed, weight, size and cost, DC motors are always preferred over stepper motors.

3.3.3 | Optocouplers/Opto Isolators

An optocoupler is another device helpful in damping the back EMF produced in a motor circuit. An optocoupler or optical isolator is a device that uses an optical transmission path to transfer a signal between elements of a circuit, while keeping them electrically isolated. An optocoupler is essentially a combination of two distinct devices: an optical transmitter, typically a gallium arsenide LED (light-emitting diode) and an optical receiver such as a phototransistor or light-triggered diac. The two are separated by a transparent [insulator] barrier which blocks any electrical current flow between the two, but does allow the passage of light.

The basic idea is shown in Figure 3.44. Usually the electrical connections to the LED section are brought out to the pins on one side of the package and those for the

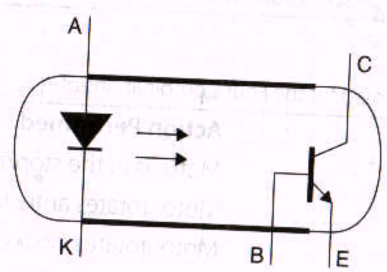


Figure 3.44 | Operating principle of an optocoupler

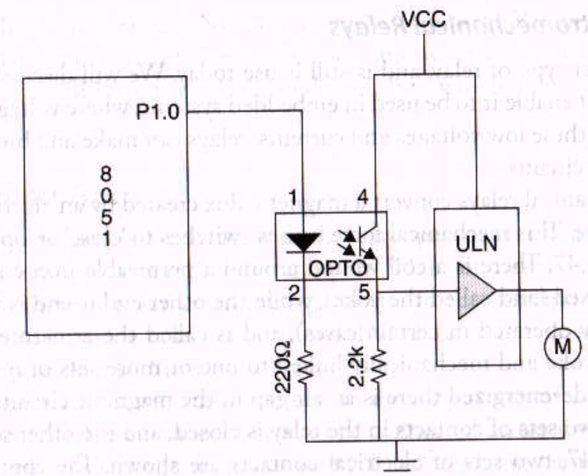


Figure 3.45 | A DC motor connected to 8051 through a current driver and an optocoupler

phototransistor or diac to the other side, to physically separate them as much as possible. Figure 3.45 shows the K817. optocoupler used in a stepper DC motor interfacing circuit, along with a driving IC ULN 2003.

3.3.4 | Relays

Switches which can be turned ON and OFF without manual control constitute a 'relay'. Relays can be used to connect and disconnect between points in a circuit, by using electrical control logic. Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example, a low voltage circuit can use a relay to switch a high voltage (say, 230V AC mains) circuit. There is no electrical connection inside the relay between the two circuits, the link is magnetic and mechanical. Such relays are 'electromechanical'. Figure 3.46 is a photograph of such a relay. Newer relays are of the semiconductor type.

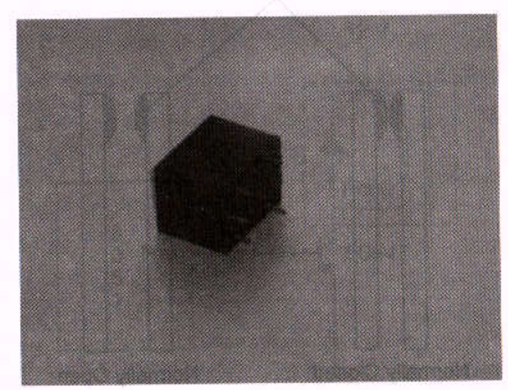


Figure 3.46 | Photograph of a relay

3.3.4.1 | *Electromechanical Relays*

This is the earliest type of relay, and is still is use today. We will discuss the types wi specifications that enable it to be used in embedded systems, where voltages and curren are low. By using these low voltages and currents, relays can make and break connectio in higher voltage circuits.

Electromechanical relays convert a magnetic flux created by an electrical signal int a mechanical force. This mechanical force causes switches to 'close' or 'open'

See Figure 3.47. There is a coil wound around a permeable iron core. One end of the iron core is fixed (and called the yoke), while the other end is end is free and sprin loaded (or gravity operated in certain cases), and is called the armature. The armatur is hinged to the yoke and mechanically linked to one or more sets of moving contact When the coil is de-energized there is an air gap in the magnetic circuit. In this condi tion, one of the two sets of contacts in the relay is closed, and the other set is open.

In Figure 3.47, two sets of electrical contacts are shown. The contacts which ar open, when the coil is de-energized are called 'Normally open (NO)' contacts–simila to this, there are 'Normally closed (NC)' contacts as well. This is shown in Figure 3.48.

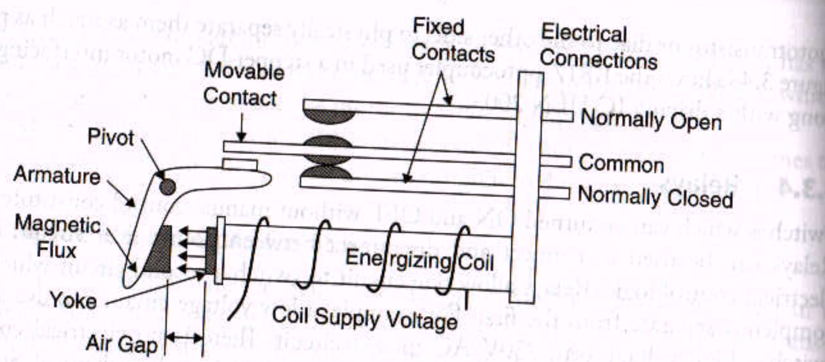


Figure 3.47 | The principle of operation of a relay

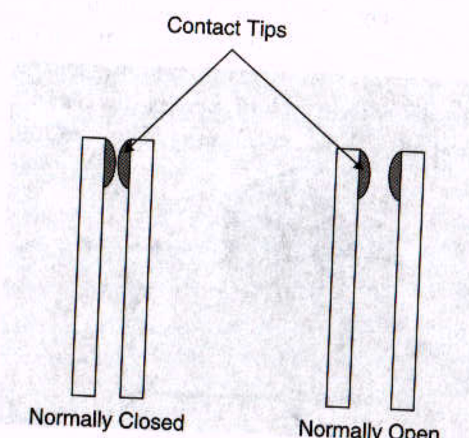


Figure 3.48 | NC and NO contacts

In a relay, there can be many NC and many NO contacts. When a contact closes, the contact is like a short circuit (zero resistance) and when open, it is an open circuit (infinite resistance). This is the ideal condition, but which may not hold true in practice.

3.3.4.2 | *Contact Types*

The energization and de-energization of a relay can open or close one or more switch contacts. Each 'contact' may be referred to as a 'pole'. Many of these contacts or poles can be connected or **'thrown'** together and this gives rise to the description of the contact types as being SPST (single pole single throw), DPST (double pole single throw) and DPDT (double pole double throw). See Figure 3.49.

For many applications, we connect a relay to the output side of a BJT or FET circuit, as in Figure 3.50. Here a diode is seen to be connected across the relay. This is the 'flywheel diode' which saves the transistor or FET from getting damaged, when there is a back emf from the coil. This is produced when the current in the coil is turned off. The magnetic flux collapses within the coil and results in a back emf which may be very high in comparison with the switching voltages used for the active device in the circuit. The diode is connected with such a polarity that the back emf makes it conduct and dissipates the story energy in it, thus preventing damage to the BJT/FET. The diode now is called the flywheel/freewheeling/snubber diode. Motors are another type of inductive load which require such a flywheel diode.

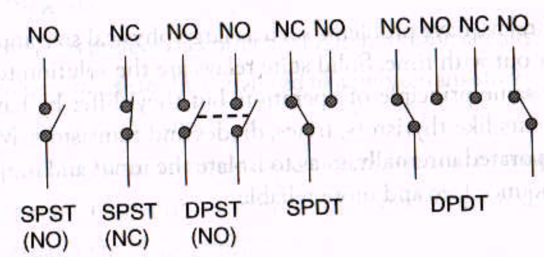


Figure 3.49 | Contact types

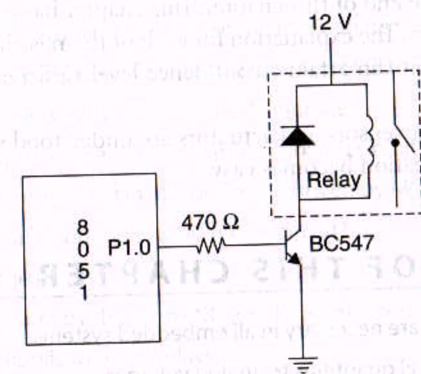


Figure 3.50 | A simple relay circuit

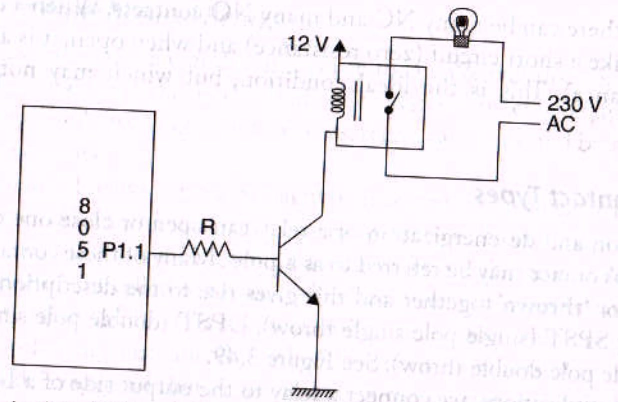


Figure 3.51 | A relay circuit switching a high voltage circuit

Relays in embedded systems are connected to output pins of microcontrollers. In many cases, it is used to control high power circuits, that is, high voltage/current, by using the lower voltage levels in the MCU. See Figure 3.51 which shows a relay used to control the switching of a bulb rated for 230V. In this case, the contacts of the relay should be able to withstand the high current passing through it.

3.3.4.3 | *Solid State Relays*

Electromechanical relays have problems such as large physical size and mechanical parts which tend to wear out with time. Solid state relays are the solution to these problems.

They have the same principle of operation, but they differ by having semiconductor switching elements like thyristors, triacs, diodes and transistors. Most of them have optoisolators incorporated internally, so as to isolate the input and output. Thus, they are smaller, noiseless, bounce free and more reliable.

Conclusion

With this, we come to the end of this chapter. This chapter has covered many commonly used sensors and actuators. The explanation for each of them is aimed at a level of knowledge with which a student can attain a confidence level sufficient to let him endeavour to do practical projects.

Once a few particular sensors and actuators are understood well, choosing the right one for a particular application becomes easy.

KEY POINTS OF THIS CHAPTER

- Sensors and actuators are necessary in all embedded systems.
- Sensors convert physical quantities to analog voltages.
- Some popular temperature sensors are thermistors and thermo couples.
- Commonly used light sensors are LDRs and photo junction devices.
- Sharp (the Company) has developed a good set of proximity and range sensors.
- Optical encoders are fixed to the wheels of moving vehicles to measure velocity.
- Humidity sensors are used in homes, factories and automobiles.
- A to D converters have data and control interfaces
- The data interface of ADCs may be parallel or serial.
- Displays are a necessity in many embedded systems.
- LEDs may be used singly or as seven segment ones.
- Seven segment LEDs may operate in the static or dynamic modes.
- OLEDs are a new kind of display devices
- LCDs are very popular and are available as Character LCD or Graphical LCD modules.
- Motors which are used in embedded systems are stepper motors or DC motors.
- Relays are either electromagnetic or solid state types.

QUESTIONS

1. What is the role of sensors in an embedded system? Name the sensors used in two popular embedded systems.
2. How does an LDR work?
3. Why are infra red LEDs preferred to ordinary LEDs in sensing circuits?
4. How can an ordinary transistor be converted to a photo transistor?
5. How does a proximity sensor work?
6. Explain the principle by which range is calculated by the Sharp range sensor?
7. What is the operating mechanism of an optical encoder? Where is such an encoder used?
8. List two applications of humidity sensors.
9. Distinguish between the terms 'control interface' and 'data interface' for an ADC.
10. Why are serial ADCs becoming more popular than the parallel ones?
11. How can a static seven segment LED system be made 'dynamic'?
12. In what ways is a Graphical LCD superior to a Character LCD?
13. What is the role of drivers in the Graphical LCD module?
14. What role does an optocoupler have in a motor driving circuit?
15. Why are current drivers needed in motor circuits driven by MCUs?
16. What is the need for an H bridge in a DC motor circuit?
17. How does an H bridge work?
18. What do you understand by the terms NO and NC contacts of a relay?
19. Discuss two simple applications of relays.
20. What are the merits of a semiconductor relay over an electromagnetic relay?