1 Programmable logic controllers

This chapter is an introduction to the programmable logic controller, its general function, hardware forms and internal architecture. This overview is followed up by more detailed discussion in the following chapters.

1.1 Controllers

What type of task might a control system have? It might be required to control a sequence of events or maintain some variable constant or follow some prescribed change. For example, the control system for an automatic drilling machine (Figure 1.1(a)) might be required to start lowering the drill when the workpiece is in position, start drilling when the drill reaches the workpiece, stop drilling when the drill has produced the required depth of hole, retract the drill and then switch off and wait for the next workpiece to be put in position before repeating the operation. Another control system (Figure 1.1(b)) might be used to control the number of items moving along a conveyor belt and direct them into a packing case. The inputs to such control systems might be from switches being closed or opened, e.g. the presence of the workpiece might be indicated by it moving against a switch and closing it, or other sensors such as those used for temperature or flow rates. The controller might be required to run a motor to move an object to some position, or to turn a valve, or perhaps a heater, on or off.

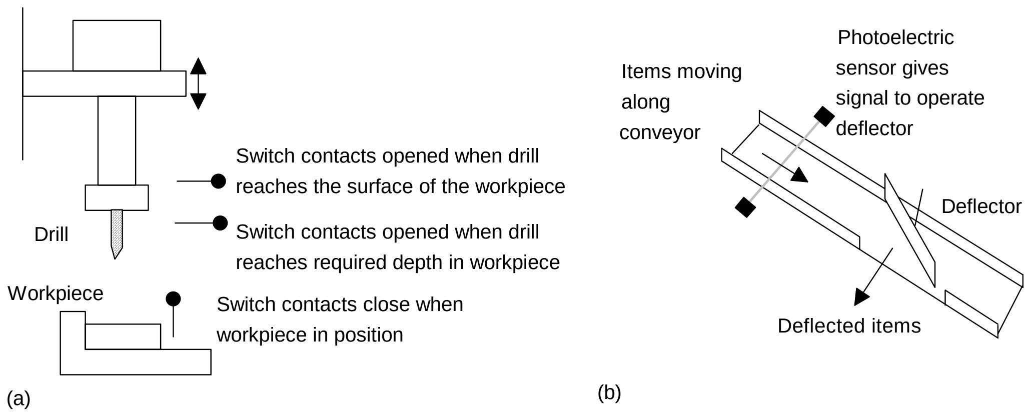

Figure 1.1 *An example of a control task and some input sensors: (a) an automatic drilling machine, (b) a packing system*

What form might a controller have? For the automatic drilling machine, we could wire up electrical circuits in which the closing or opening of switches would result in motors being switched on or valves being actuated. Thus we might have the closing of a switch activating a relay which, in turn, switches on the current to a motor and causes the drill to rotate (Figure 1.2). Another switch might be used to activate a relay and switch on the current to a pneumatic or hydraulic valve which results in pressure being switched to drive a piston in a cylinder and so results in the workpiece being pushed into the required position. Such electrical circuits would have to be specific to the automatic drilling machine. For controlling the number of items packed into a packing case we could likewise wire up electrical circuits involving sensors and motors. However, the controller circuits we devised for these two situations would be different. In the 'traditional' form of control system, the rules governing the control system and when actions are initiated are determined by the wiring. When the rules used for the control actions are changed, the wiring has to be changed.

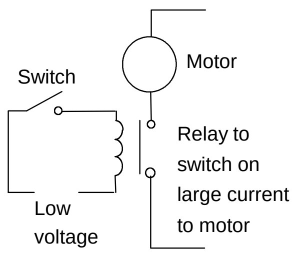

Figure 1.2 *A control circuit*

1.1.1 Microprocessor controlled system

Instead of hardwiring each control circuit for each control situation we can use the same basic system for all situations if we use a microprocessor-based system and write a program to instruct the microprocessor how to react to each input signal from, say, switches and give the required outputs to, say, motors and valves. Thus we might have a program of the form:

If switch A closes
Output to motor circuit
If switch B closes
Output to valve circuit

By changing the instructions in the program we can use the same microprocessor system to control a wide variety of situations.

As an illustration, the modern domestic washing machine uses a microprocessor system. Inputs to it arise from the dials used to select the required wash cycle, a switch to determine that the machine door is closed, a temperature sensor to determine the temperature of the water and a switch to detect the level of the water. On the basis of these inputs the microprocessor is programmed to give outputs which switch on the drum motor and control its speed, open or close cold and hot water valves, switch on the drain pump, control the water heater and control the door lock so that the machine cannot be opened until the washing cycle is completed.

1.1.2 The programmable logic controller

A *programmable logic controller* (PLC) is a special form of microprocessor-based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes (Figure 1.3) and are designed to be operated by engineers with perhaps a limited knowledge of computers and computing languages. They are not designed so that only computer programmers can set up or change the programs. Thus, the designers of the PLC have pre-programmed it so that the control program can be entered using a simple, rather intuitive, form of language, see Chapter 4. The term *logic* is used because programming is primarily concerned with implementing logic and switching operations, e.g. if A or B occurs switch on C, if A and B occurs switch on D. Input devices, e.g. sensors such as switches, and output devices in the system being controlled, e.g. motors, valves, etc., are connected to the PLC. The operator then enters a sequence of instructions, i.e. a program, into the memory of the PLC. The controller then monitors the inputs and outputs according to this program and carries out the control rules for which it has been programmed.

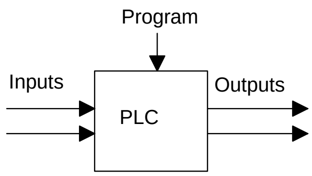

Figure 1.3 *A programmable logic controller*

PLCs have the great advantage that the same basic controller can be used with a wide range of control systems. To modify a control system and the rules that are to be used, all that is necessary is for an operator to key in a different set of instructions. There is no need to rewire. The result is a flexible, cost effective, system which can be used with control systems which vary quite widely in their nature and complexity.

PLCs are similar to computers but whereas computers are optimised for calculation and display tasks, PLCs are optimised for control tasks and the industrial environment. Thus PLCs are:

1. Rugged and designed to withstand vibrations, temperature, humidity and noise.
2. Have interfacing for inputs and outputs already inside the controller.

3 Are easily programmed and have an easily understood programming language which is primarily concerned with logic and switching operations.

The first PLC was developed in 1969. They are now widely used and extend from small self-contained units for use with perhaps 20 digital inputs/outputs to modular systems which can be used for large numbers of inputs/outputs, handle digital or analogue inputs/outputs, and also carry out proportional-integral-derivative control modes.

1.2 Hardware

Typically a PLC system has the basic functional components of processor unit, memory, power supply unit, input/output interface section, communications interface and the programming device. Figure 1.4 shows the basic arrangement.

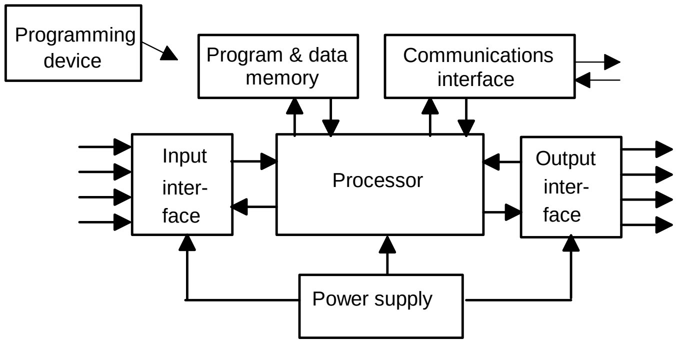

Figure 1.4 *The PLC system*

1 The *processor unit* or *central processing unit (CPU)* is the unit containing the microprocessor and this interprets the input signals and carries out the control actions, according to the program stored in its memory, communicating the decisions as action signals to the outputs.
2 The *power supply unit* is needed to convert the mains a.c. voltage to the low d.c. voltage (5 V) necessary for the processor and the circuits in the input and output interface modules.
3 The *programming device* is used to enter the required program into the memory of the processor. The program is developed in the device and then transferred to the memory unit of the PLC.
4 The *memory unit* is where the program is stored that is to be used for the control actions to be exercised by the microprocessor and data stored from the input for processing and for the output for outputting.
5 The *input and output sections* are where the processor receives information from external devices and communicates information to external devices. The inputs might thus be from switches, as illustrated in Figure 1.1(a) with the automatic drill, or other sensors such as photo-electric cells, as in the counter mechanism in Figure 1.1(b), temperature sensors, or flow sensors, etc. The outputs might be to motor starter coils, solenoid valves, etc. Input and output

interfaces are discussed in Chapter 2. Input and output devices can be classified as giving signals which are discrete, digital or analogue (Figure 1.5). Devices giving *discrete* or *digital signals* are ones where the signals are either off or on. Thus a switch is a device giving a discrete signal, either no voltage or a voltage. *Digital* devices can be considered to be essentially discrete devices which give a sequence of on–off signals. *Analogue* devices give signals whose size is proportional to the size of the variable being monitored. For example, a temperature sensor may give a voltage proportional to the temperature.

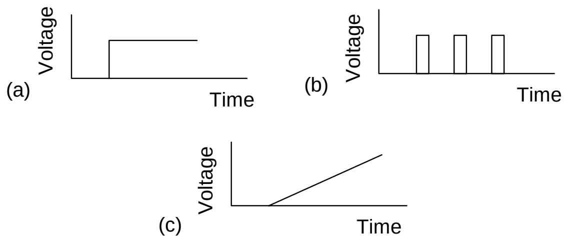

Figure 1.5 *Signals: (a) discrete, (b) digital, (c) analogue*

6 The *communications interface* is used to receive and transmit data on communication networks from or to other remote PLCs (Figure 1.6). It is concerned with such actions as device verification, data acquisition, synchronisation between user applications and connection management.

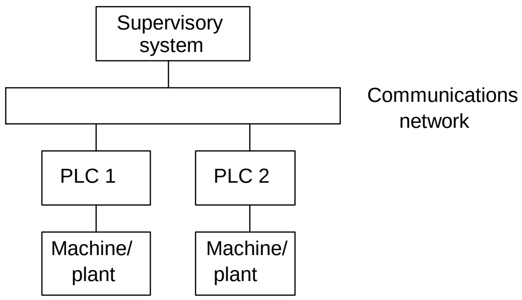

Figure 1.6 *Basic communications model*

1.3 Internal architecture

Figure 1.7 shows the basic internal architecture of a PLC. It consists of a central processing unit (CPU) containing the system microprocessor, memory, and input/output circuitry. The CPU controls and processes all the operations within the PLC. It is supplied with a clock with a frequency of typically between 1 and 8 MHz. This frequency determines the operating speed of the PLC and provides the timing and synchronisation for all elements in the system. The information within the PLC is carried by means of digital signals. The internal paths along which digital signals flow are called *buses*. In the physical sense, a bus is just a number of conductors along which electrical signals can flow. It might be tracks on a printed circuit board or wires in a ribbon cable. The CPU uses the *data bus* for sending data between the constituent elements, the *address bus* to send the addresses of locations for accessing stored data and the *control bus* for signals relating to internal control actions. The *system bus* is used for communications between the input/output ports and the input/output unit.

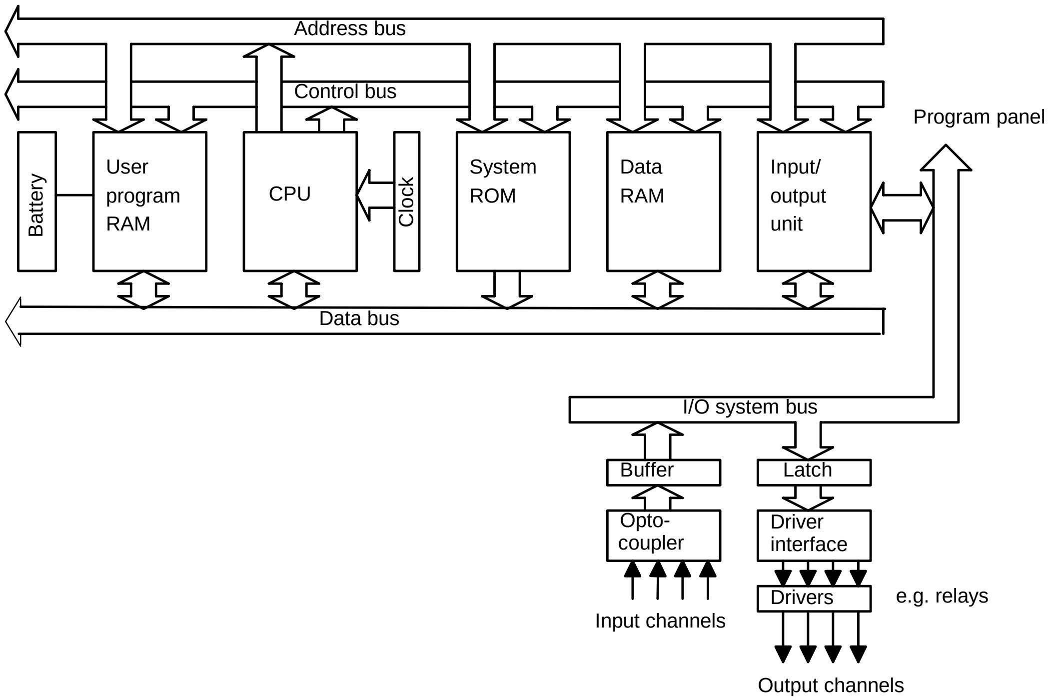

Figure 1.7 *Architecture of a PLC*

1.3.1 The CPU

The internal structure of the CPU depends on the microprocessor concerned. In general they have:

1 An *arithmetic and logic unit* (ALU) which is responsible for data manipulation and carrying out arithmetic operations of addition and subtraction and logic operations of AND, OR, NOT and EXCLUSIVE-OR.
2 Memory, termed *registers*, located within the microprocessor and used to store information involved in program execution.
3 A *control unit* which is used to control the timing of operations.

1.3.2 The buses

The buses are the paths used for communication within the PLC. The information is transmitted in binary form, i.e. as a group of *bits* with a bit being a binary digit of 1 or 0, i.e. on/off states. The term *word* is used for the group of bits constituting some information. Thus an 8-bit word might be the binary number 00100110. Each of the bits is communicated simultaneously along its own parallel wire. The system has four buses:

1 The *data bus* carries the data used in the processing carried out by the CPU. A microprocessor termed as being 8-bit has an internal data bus which can handle 8-bit numbers. It can thus perform operations between 8-bit numbers and deliver results as 8-bit values.
2 The *address bus* is used to carry the addresses of memory locations. So that each word can be located in the memory, every memory location is given a unique *address*. Just like houses in a town are each given a distinct address so that they can be located, so each word location is given an address so that data stored at a particular location can be accessed by the CPU either to read data located there or put, i.e. write, data there. It is the address bus which carries the information indicating which address is to be accessed. If the address bus consists of 8 lines, the number of 8-bit words, and hence number of distinct addresses, is $2^8 = 256$. With 16 address lines, 65 536 addresses are possible.
3 The *control bus* carries the signals used by the CPU for control, e.g. to inform memory devices whether they are to receive data from an input or output data and to carry timing signals used to synchronise actions.
4 The *system bus* is used for communications between the input/output ports and the input/output unit.

1.3.3 Memory

There are several memory elements in a PLC system:

1 System *read-only-memory* (*ROM)* to give permanent storage for the operating system and fixed data used by the CPU.
2 *Random-access memory (RAM)* for the user’s program.
3 *Random-access memory (RAM)* for data. This is where information is stored on the status of input and output devices and the values of timers and counters and other internal devices. The data RAM is sometimes referred to as a *data table* or *register table*. Part of this memory, i.e. a block of addresses, will be set aside for input and output addresses and the states of those inputs and outputs. Part will be set aside for preset data and part for storing counter values, timer values, etc.
4 Possibly, as a bolt-on extra module, *erasable and programmable read-only-memory (EPROM)* for ROMs that can be programmed and then the program made permanent.

The programs and data in RAM can be changed by the user. All PLCs will have some amount of RAM to store programs that have been developed by the user and program data. However, to prevent the loss of programs when the power supply is switched off, a battery is used in the PLC to maintain the RAM contents for a period of time. After a program has been developed in RAM it may be loaded into an EPROM memory chip, often a bolt-on module to the PLC, and so made permanent. In addition there are temporary *buffer* stores for the input/output channels.

The storage capacity of a memory unit is determined by the number of binary words that it can store. Thus, if a memory size is 256 words then it can store $256 \times 8 = 2048$ bits if 8-bit words are used and $256 \times 16 = 4096$ bits if 16-bit words are used. Memory sizes are often specified in terms of the number of storage locations available with 1K representing the number 2^{10}, i.e. 1024. Manufacturers supply memory chips with the storage locations grouped in groups of 1, 4 and 8 bits. A 4K × 1 memory has $4 \times 1 \times 1024$ bit locations. A 4K × 8 memory has $4 \times 8 \times 1024$ bit locations. The term *byte* is used for a word of length 8 bits. Thus the 4K × 8 memory can store 4096 bytes. With a 16-bit address bus we can have 2^{16} different addresses and so, with 8-bit words stored at each address, we can have $2^{16} \times 8$ storage locations and so use a memory of size $2^{16} \times 8/2^{10}$ = 64K × 8 which we might be as four 16K × 8 bit memory chips.

1.3.4 Input/output unit

The input/output unit provides the interface between the system and the outside world, allowing for connections to be made through input/output channels to input devices such as sensors and output devices such as motors and solenoids. It is also through the input/output unit that programs are entered from a program panel. Every input/output point has a unique address which can be used by the CPU. It is like a row of houses along a road, number 10 might be the 'house' to be used for an input from a particular sensor while number '45' might be the 'house' to be used for the output to a particular motor.

The input/output channels provide isolation and signal conditioning functions so that sensors and actuators can often be directly connected to them without the need for other circuitry. Electrical isolation from the external world is usually by means of *optoisolators* (the term *optocoupler* is also often used). Figure 1.8 shows the principle of an optoisolator. When a digital pulse passes through the light-emitting diode, a pulse of infrared radiation is produced. This pulse is detected by the phototransistor and gives rise to a voltage in that circuit. The gap between the light-emitting diode and the phototransistor gives electrical isolation but the arrangement still allows for a digital pulse in one circuit to give rise to a digital pulse in another circuit.

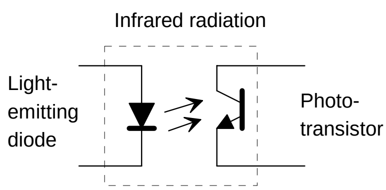

Figure 1.8 *Optoisolator*

The digital signal that is generally compatible with the microprocessor in the PLC is 5 V d.c. However, signal conditioning in the input channel, with isolation, enables a wide range of input signals to be supplied to it (see Chapter 3 for more details). A range of inputs might be available with a larger PLC, e.g. 5 V, 24 V, 110 V and 240 V digital/discrete, i.e. on−off, signals (Figure 1.9). A small PLC is likely to have just one form of input, e.g. 24 V.

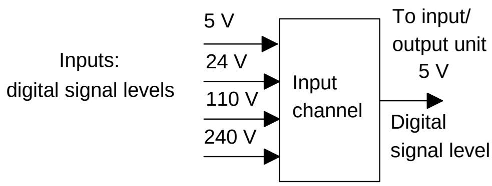

Figure 1.9 *Input levels*

The output from the input/output unit will be digital with a level of 5 V. However, after signal conditioning with relays, transistors or triacs, the output from the output channel might be a 24 V, 100 mA switching signal, a d.c. voltage of 110 V, 1 A or perhaps 240 V, 1 A a.c., or 240 V, 2 A a.c., from a triac output channel (Figure 1.10). With a small PLC, all the outputs might be of one type, e.g. 240 V a.c., 1 A. With modular PLCs, however, a range of outputs can be accommodated by selection of the modules to be used.

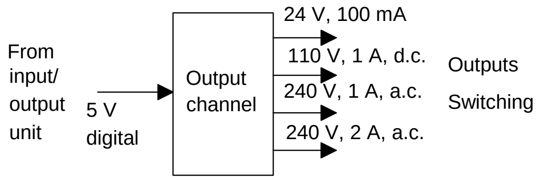

Figure 1.10 *Output levels*

Outputs are specified as being of relay type, transistor type or triac type (see Chapter 3 for more details):

1. With the *relay type*, the signal from the PLC output is used to operate a relay and is able to switch currents of the order of a few amperes in an external circuit. The relay not only allows small currents to switch much larger currents but also isolates the PLC from the external circuit. Relays are, however, relatively slow to operate. Relay outputs are suitable for a.c. and d.c. switching. They can withstand high surge currents and voltage transients.
2. The *transistor type* of output uses a transistor to switch current through the external circuit. This gives a considerably faster switching action. It is, however, strictly for d.c. switching and is destroyed by overcurrent and high reverse voltage. As a protection, either a fuse or built-in electronic protection are used. Optoisolators are used to provide isolation.

3. *Triac* outputs, with optoisolators for isolation, can be used to control external loads which are connected to the a.c. power supply. It is strictly for a.c. operation and is very easily destroyed by overcurrent. Fuses are virtually always included to protect such outputs.

1.3.5 Sourcing and sinking

The terms *sourcing* and *sinking* are used to describe the way in which d.c. devices are connected to a PLC. With sourcing, using the conventional current flow direction as from positive to negative, an input device receives current from the input module, i.e. the input module is the source of the current (Figure 1.11(a)). If the current flows from the output module to an output load then the output module is referred to as sourcing (Figure 1.11(b)). With sinking, using the conventional current flow direction as from positive to negative, an input device supplies current to the input module, i.e. the input module is the sink for the current (Figure 1.12(a)). If the current flows to the output module from an output load then the output module is referred to as sinking (Figure 1.12(b)).

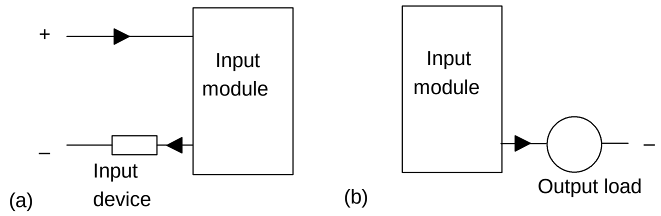

Figure 1.11 *Sourcing*

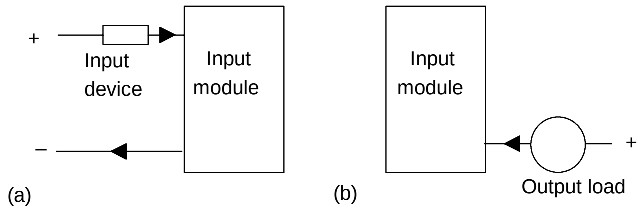

Figure 1.12 *Sinking*

1.4 PLC systems

There are two common types of mechanical design for PLC systems; a *single box*, and the *modular/rack types*. The single box type (or, as sometimes termed, brick) is commonly used for small programmable controllers and is supplied as an integral compact package complete with power supply, processor, memory, and input/output units. Typically such a PLC might have 6, 8, 12 or 24 inputs and 4, 8 or 16 outputs and a memory which can store some 300 to 1000 instructions. Figure 1.13 shows the Mitsubishi MELSEC FX3U compact, i.e. brick, PLC and Table 1.1 gives details of models in that Mitsubishi range.