The overall objectives of this lab is to give:

- An appreciation of the need and uses of Programmable Logic Controllers (PLC’s) in industry.
- An understanding of PLC Ladder Logic design.
- A familiarity with the PLC simulator LADSIM.
- Experience of controlling a representation of an industrial process using the Industrial Control Trainer (ICT).

Programmable Logic Controllers (PLC’s) are microprocessor devices that are used in industry to control plant operations. There are different makes and models of PLC’s but they all consist of a Power Supply, a Central Processing Unit and Input/Output modules, shown in figure 1.

Programmable Logic Controllers (PLC’s) are microprocessor devices that are used in industry to control plant operations. There are different makes and models of PLC’s but they all consist of a Power Supply, a Central Processing Unit and Input/Output modules, shown in figure 1.

### Figure 1: Basic Block Diagram of a PLC

The control of plant equipment is via the I/O modules and the CPU program. The I/O modules have external terminals that can be connected to various electrical points i.e. an output could be connected to a solenoid and an input could be a sensor or push-button.

Prior to the introduction of PLC’s, almost all plant was controlled by relay logic, known as hard wired. Any change required by the plant would mean major rewiring of the relays resulting in large amounts of production down time. With the introduction of PLC’s extremely complex changes can be made by altering the program with the minimum amount of disruption to production. PLC’s are therefore responsible for massive increases in productivity and product availability.

The Industrial Control Trainer (ICT) is used to show how a PLC can control a plant. The system has been designed to realistically demonstrate a component sorting, assembly and inspection process that can be controlled from any commonly available PLC. To start the demonstration use the following instructions.

- From Windows select MS-DOS Prompt (under the MAIN icon).
- At the C:\> prompt type CD ICT and then type GO
- Place the separate components randomly on the chain conveyor.
- Make sure the ‘Start’ indicator, next to the push-buttons on the ICT, is illuminated. Press F1 to start system.

The unit is designed to sort an aluminum ‘peg’ from a plastic ‘ring’ then assemble these two components and check for correct assembly. The components are initially randomly placed on the chain conveyor which lifts them to a higher level. When the higher level is reached the plastic components are detected by an infra red emitter/detector and ejected by a solenoid down an
assembly chute. The aluminum pegs, meanwhile, continue on the conveyor and are finally deflected down the feeder chute. From the feeder chute the aluminum components are automatically fed onto the belt conveyor.

An infra red emitter/detector is used to determine whether or not the plastic ring ‘hopper’ is empty. If it is, a rotary solenoid is used to dispense a ring from the assembly chute into the hopper. The hopper is positioned just above the belt conveyor and, when the aluminum peg passes, the peg engages with the hole in the ring and the two are assembled. Inductive, capacitive and infra red industrial sensors positioned along the belt conveyor are used to check for correct assembly. Components that have not been assembled are ejected by a solenoid into a ‘scrap’ or ‘recycle’ bin while correctly assembled components pass into a finished parts tray.

The objective of this laboratory is for you to understand PLC programming so that it may be possible to fully control this process as demonstrated.

**The report:** The lab should be written up as an informal report giving the ladder logic used for each stage and an explanation of how this ladder logic accomplished the task. If you fail to finish the whole exercise then you may write up how you would have done each uncompleted stage. Reports must be handed-in to the General Office on the 6th Floor of the Tower Building between 9.30 and 12.00, Monday to Thursday. All reports should be handed-in by 12.00 on the 7th day following the experiment. Late reports will lose 5% per day subtracted from the mark given for the report.

Now stop the system (F1) and Quit (F7). At the C:\ICT> prompt type EXIT to return to Windows.
METHODS OF PROGRAMMING

Originally PLC programming was used to directly replace the old relay logic and thus adopted the old relay control schematic diagram known as a ladder diagram which used standard electrical symbols, shown in figure 2.

![Ladder Diagram](image_url)

**Figure 2: Relay Ladder Schematic Diagram**

In the ladder diagram shown the vertical rails represent the power supply. Operation of the push-button will energize control relay 1 (CR1), providing that the normally open contact of control relay 3 (CR3) is closed (CR3 being energized elsewhere in the circuit) and the normally closed contact of control relay 4 (CR4) is in its normally closed state i.e. CR4 is not energized. When CR1 is energized, the normally open contacts of CR1 will latch CR1 and consequently energize CR2.

Assuming that the control relays are connected to various parts of the plant equipment, the plant will operate according to the conditions of the ladder diagram.

The ladder programming in PLC’s replaces the contacts and coils of the relays and also uses timers, counters, arithmetic functions and many more. The main disadvantage of the ladder format is that requires a fairly significant amount of memory.
LADDER LOGIC PROGRAMMING

The LADSIM package is a ladder logic editor and PLC Simulator. As the program has a built in PLC simulator it allows the user to program off-line and test the written programs. To start up LADSIM from Windows, double-click on the BYTRONIC icon and again on LADSIM. For programming the following items are available.

- 12 inputs IP0 to IP11
- 12 outputs OP0 to OP11
- 16 flags F0 to F15 (Auxiliary Relays)
- 6 Timers T1 to T6
- 6 counters C1 to C6
- 12 reset functions for the timers and counters
- 4 16-Bit Shift Registers

PROGRAMMING WITH LADSIM

1. In the Ladder Logic Editing Suite the Controls boxes can be seen.
2. Move the mouse to the normally open contact (NOC) and hold down the left mouse button, the pointer changes to a NOC. Drag the NOC to the left hand side of the ladder rung and release the mouse button. The NOC should now be placed on the rung and the Inputs selection box will be displayed.
3. Select IP0 and then OK. The ladder rung will now have a numbered input contact.
4. Now place another NOC next to the first NOC and call this IP1.
5. Finally an output is required. Drag an output , -( )-, from the controls box and place on the right side of the rung. The Output selection box is displayed and select OP0.
6. This now forms the ladder equivalent of an AND function.

Note:

- To change numbering mistakes double-click on the input or output to return to the selection box or use the Edit Control button.

Simulating the program

(Before simulating always save your program)

With the ladder diagram displayed select the Simulate button and the Debugging Simulator box will be displayed with the ladder diagram compiled. The simulator box allows you to operate the various control elements by clicking on them with the mouse. The ladder diagram will also be shown.

- To simulate the ladder diagram firstly operate IP0 by clicking on the IP0 switch box. The switch should now close and the associated contact on the ladder diagram should turn GREEN, indicating ‘ON’.
- Now close the next input contact, IP1, in the same way. IP1 on the ladder diagram will turn GREEN as well as the output OP0. On the Simulator box OP0 will turn RED indicating it has energized.

Press the STOP button to end the simulation.

7. Now delete IP1 using the Delete Control button. Insert a branch around IP0 using the Add Branch button and then place a NOC on that branch. This is a now an OR function. Simulate as before to verify operation.
Branching is versatile and can be around more than one input or individual inputs. To branch outputs, drag another output from the Controls box to the right hand side. Delete branches using the Delete Branch button.

**FURTHER FUNCTIONS**

The following functions are found in the Controls box. It is advised that you draw and simulate the ladder circuits given with each function to become familiar with their operation.

**Latch (L) And Unlatch (U) Functions**

The Latch and Unlatch functions are matched in pairs ie. if an output is latched there must be a corresponding unlatch function for that output. This is very useful as it only requires the input to be energized momentarily for the output to be latched ‘ON’. The Latched output (real or auxiliary) will remain ‘ON’ until the corresponding unlatch function is energized.

```
Rung 0

\[ \text{I} \rightarrow \text{0} \rightarrow \text{L} \rightarrow \text{O} \]

Rung 1

\[ \text{I} \rightarrow \text{1} \rightarrow \text{L} \rightarrow \text{O} \]

Rung 2

\[ \text{I} \rightarrow \text{2} \rightarrow \text{L} \rightarrow \text{O} \]

Rung 3

\[ \text{I} \rightarrow \text{3} \rightarrow \text{L} \rightarrow \text{O} \]
```

*Figure 17: Use of the Latch and Unlatch Facility*

**Timer Function**

The use of a timer allows precise control over events. LADSIM has 6 timers which when activated will increment in 1 second intervals from 1 to 32,768 seconds. In practice, PLC’s are capable of incrementing in milliseconds.

If the timer function is energized, the timer will start to increment until it reaches the preset value and the timers Done Bit (DN) will be set. The Done Bit can be used as an input contact to energize an output.

Note:

- The incrementing can be observed in the simulation mode in the timers Accumulator.
- If the timer is de-energized then it will continue to increment until the preset value is reached.
- Once the Done Bit has been set it can only be switched ‘OFF’ by using the reset (RES) function. The reset is an output function that can be energized by an input contact and has priority over the timers incrementing.
- When the timer has completed its incrementing an x will appear in the DN box of the Simulator box.
**Counter Function**

The counter function uses a preset value, like the timers, but counts according to its input changing from off ‘OFF’ to ‘ON’ (positive transition). When the number of these transitions are equal to the preset value a Bit is set, further transitions will be ignored. If the counter is an UP counter it sets the UP bit, if it was a DOWN counter it sets the DN bit.

![Diagram of Timer Functions with Corresponding RES Function](image)

**Figure 18: Timer Functions with Corresponding RES Function**

![Diagram of Counter Ladder Format](image)

**Figure 19: Counter Ladder Format**
Note:

- The counting can be observed in the simulation mode in the counters Accumulators.
- The same counter can be used to count both up and down.
- Like the timers, once the Bit has been set it can only be reset by the (RES) function and this has priority over the counting.

**Shift Registers**

The shift Register uses a 16 Bit Word. The Word is just a collection of single bits grouped together and used to provide information to be used by the processor. The Shift Registers used are the Bit Shift Right (BSR) and the Bit Shift Left (BSL). As the name suggests the bits are shifted to the right and left within the register. Draw and simulate the following example.

To test the operation of the ladder diagram, operate IP0 momentarily. This causes a bit, ‘1’, to enter the shift register from the left and move one place to the right at 1 second intervals (determined by the timer) until it drops out of the right hand side of the register. Leaving IP0 closed will fill the register with ‘1’ s at one per second. Similarly, the (BSL) can be used.

![Ladder Diagram](image)

**Figure 20: Bit Shift Right Example**
INDUSTRIAL CONTROL TRAINER

The next task is to work through the stages of controlling the ICT so that a fully operational system is achieved. Figure 21 shows a schematic of the ICT with the position and labels of the inputs and outputs.

To simulate at each stage either use the Debugger or select Real-Time Control under the Control menu.

Once each stage is complete sketch the ladder diagram and briefly describe the sequence in your report.

![Schematic Diagram of the Industrial Control Trainer](image)

**Figure 21: Schematic Diagram of the Industrial Control Trainer**

**Stage 1. Sorting Routine**

Component detection and sorting according to component type is a common requirement throughout industrial control.

**Objectives:**
- Detection of plastic component traveling along the chain conveyor and ejection into assembly chute.
- Detection of metal component on chain conveyor and allowing this to pass the sorting area, be deflected off the chain conveyor and down the feeder chute.

**Procedure:**
- Activate the chain conveyor (OP3).
- In the sort area there are two infra red (IR) proximity sensors.
  - IP1, to detect either component
  - IP4, a reflective sensor to sense the metal peg.
- Eject plastic rings and metal pegs
**Stage 2. Queue Counting**

Component counting is an obvious necessity in almost all areas of plant and process control.

Objectives:
- To enhance the previous stage by including a queue counter for the assembly chute. This is essential because the assembly chute can only queue a maximum of 5 rings. Once the queue is filled, there are then two options.
  a. Stop the chain conveyor.
  b. Ignore the plastic rings at the sorting area allowing them to pass to the excess bin.

Procedure:
- Activate the chain conveyor.
- Sort the components as before.
- Increment a counter each time a plastic ring is ejected down the assembly chute. When the count reaches 5 take either action (a) or (b).

**Stage 3. Operation Timing**

Objectives:
- To combine the Sorting Routine and the Queue Counting together with Assembly Area Loading using the rotary indexing unit.
- Generate a queue count decrement as the plastic rings are taken from the queue into the assembly area.
- Introduce essential time delays to take account of system lags.

Procedure:
- Operate system as in Stage 2.
- In the Assembly Area an IR reflective sensor (IP0) detects the presence of a plastic ring in the assembly hopper. If this sensor indicates the hopper is empty then the rotary solenoid (OP1) can be indexed, loading the hopper with a plastic ring. The cue count can now be decremented, enabling the sorting routine. This sequence is shown in figure 22.

![Figure 22: Timing Diagram for the Assembly Area](image-url)
These are the time delays that need to built into the sequences.

a) ‘Hold on’ time of rotary solenoid. (approx. 1 sec.)

b) IF: No plastic rings in the queue AND assembly hopper is empty

THEN: A delay between the solenoid pushing the plastic rings down the chute and the rotary solenoid opening to let a plastic ring into the hopper is required. (approx. 1 sec.)

\[
\text{hopper full} \quad \text{hopper empty}
\]

c) On detecting a falling edge on IP0 a delay is required before the rotary solenoid allows another plastic ring into the hopper. This allows the metal peg and plastic ring to assemble correctly and clear the assembly area before the process is repeated. (approx. 2 sec.)

**Stage 4. Plastic Ring Detection and Sorting**

Objective:

- To use the belt conveyor and standard sensor arrangement to detect the presence of a **single** plastic ring and to eject this into the reject bin. The sensors mounted on the belt conveyor can detect the presence and material of the component. These sensors are:
  - (IP7) Inductive metal sensor.
  - (IP5) Infra red (IR) through beam proximity sensor used to detect the presence of any object.
  - (IP6) Capacitive sensor which detects almost any material. It is set up to detect a plastic ring on a metal peg.
  - (IP3) IR reflective sensor next to IP6 to detect the presence of any component.
  - (IP2) IR reflective sensor next to the reject solenoid to detect the presence of any object passing. This is used in conjunction with the reject solenoid to eject items into the reject bin.

\[
\begin{align*}
\text{IP7 Inductive Sensor} \\
\text{IP5 Through Beam} \\
\text{IP6 Capacitive Sensor} \\
\text{IP3 IR at Capacitive Area} \\
\text{IP2 IR at Reject Area} \\
\text{OP2 Reject Solenoid}
\end{align*}
\]

**Figure 23: Sensor Sequence for Plastic Ring**
Procedure:
- Activate the belt conveyor (OP4).
- Logically gate the results from one pass of these sensors to detect a plastic ring and eject it. The timing diagram for a plastic ring passing through the sensors is shown in figure 23. The signal sequence can also be observed by the LED’s.

![Timing Diagram for Plastic Ring](image)

**Figure 24: Sensor Sequence for a Metal Component**

![Timing Diagram for Assembled Product](image)

**Figure 25: Signal Sequence for an Assembled Product**
Stage 5. Metal Component Detection and Sorting

Objective:
- To use the belt conveyor and sensor arrangement to detect the presence of a single metal component and eject this into the reject bin.

Procedure:
- Activate the belt conveyor.
- Use the signals from the sensors (discussed in Stage 4) to detect the presence of a metal peg and then eject it. The timing diagram for a metal peg passing through the sensors is shown in figure 24.

Stage 6. Component / Assembly Detection and Sorting

Objectives:
- Detection of a plastic ring traveling along the belt conveyor and ejecting it into the reject bin.
- Detection of a metal component traveling along the belt conveyor and ejecting it into the reject bin.
- Detection of an assembled product traveling along the belt conveyor and accepting it.

Procedure:
- Activate the belt conveyor.
- Combine the sequences for an unassembled metal peg and plastic ring (discussed in Stages 4 and 5) with the signal sequence for an assembled product, shown in figure 25.

Stage 7. Component Queue Handling

The previous method will only work with one component traveling along the conveyor belt at any one time, in reality this is an unlikely situation.

Objective:
- Enhance the detection and sorting sequences developed in Stage 6 to deal with a queue of components traveling along the belt conveyor.

Procedure:
In order to obtain data on a single product amongst a queue of products one of the following approaches is required.
- The decision on the component type may be made at a single point along the length of the conveyor. This can be achieved by putting the capacitive sensor directly opposite the inductive probe. This data must be stored in an array so that a decision on ejecting can be made.
- With the standard sensor arrangement, used in the previous stages, as a component passes a sensing area the data may be stored against the component number in an array.

Stage 8. Complete System Control

Objective:
- To sort the randomly loaded components on the chain conveyor into the assembly and feeder chutes. Loading the assembly hopper with a plastic ring when empty, as developed in Stage 3.
- Check the assembly of the metal and plastic components as they pass in a queue along the belt conveyor. Accepting the assembled components but rejecting either a single plastic ring or a metal peg, as developed in Labwork 7.

Procedure:
- Merge Solutions from Stages 3 and 7 and debug this into the final control program.