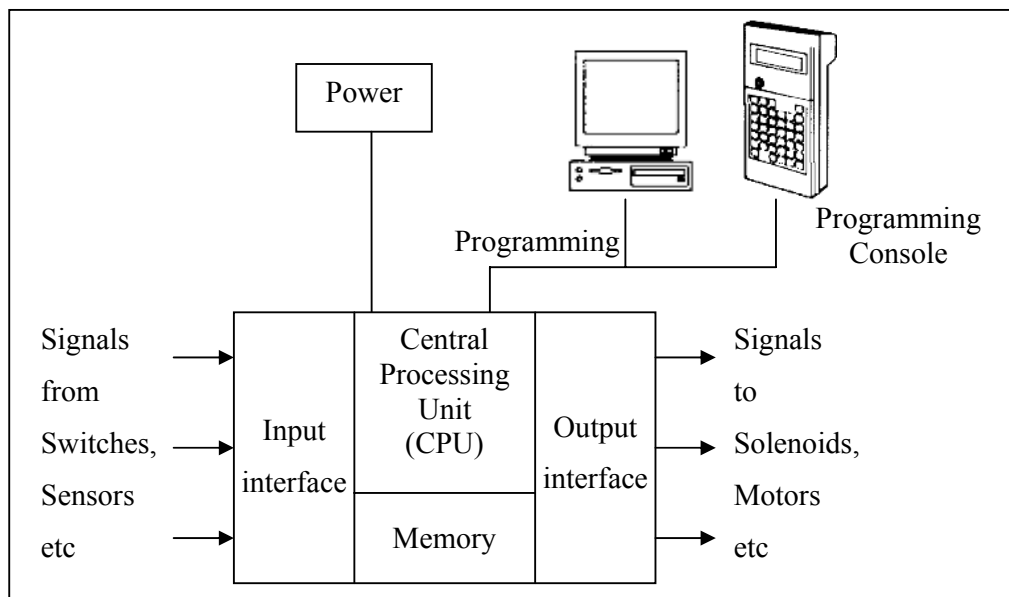# CHAPTER 2

# LITERATURE REVIEW

## 2.1     Basic of Programmable Logic Controllers

Programmable logic controller (PLC) is a digital electronic device that uses a programmable memory to store instruction and to implement specific functions such as logic, sequence, timing, counting and arithmetic to control machines and process. Figure 2.0 shows Block Diagram of PLC. [1]



**Figure 2.0**: Block Diagram of PLC.

## 2.1.1   The power supply

The power supply gives the needed voltage and current to each component in the system. If the there have no power, the PLC does not operate. This power supply basically have a special characteristic which usually have the back up unit that give the power while the electricity cut down suddenly.

### 2.1.2    Control Processing Unit (CPU)

The Central Processing Unit (CPU) is a microprocessor that co-ordinates and monitor all the activities of the PLC system. It executes the program which stored in the memory. There have one communication bus system which brings the data to or from CPU to the input or output memory units which govern by CPU. CPU have been provided with a clock which it frequency control by external quartz crystal or oscillator RC. The typical frequency was depends on the type of microprocessor and usage. Clock determines the speed of PLC operation which provides timing or synchronization for all elements in the system. [1]

### 2.1.3    Memory

There are various type of memory unit; Read-Only Memory (ROM), Random Access Memory (RAM), Erasable Programmable Read Only Memory (EPROM) and Electrical Erasable Programmable Read Only Memory (EEPROM). It is the area that holds the operation system and user memory. The operating system is actually a system software that co-ordinates the PLC. Ladder program, Timer and Counter Values are stored in the user memory.  [1]

### 2.1.4    Output and Input Interface

Input devices (e.g. mechanical contact, proximity switches) and output devices (e.g. motor, solenoids) from the machine or process to be controlled are connected to the PLC. A user enters a sequence of instructions (known as the program) into the PLC's program memory. Then, the controller monitors the state of the inputs and switches outputs according to the users program. The stored program can be modified so the new control features can be added or old ones changed without rewiring the input and output. [1]

## 2.2  Ladder Logic Diagram

Commonly, the language or command for PLC designing base on the ladder logic diagram. There have numerical of symbol that have been used to developed the ladder logic diagram which every symbols represent the different functions. The formats for designing the ladder logic diagram are:

1.  Circuits are arranged as a series of horizontal lines containing inputs (referred to as contacts) and outputs (referred to as coils).

2.  Inputs must always go before outputs and are in the form of normally open and normally closed contacts. The ladder symbol for a normally open contact is ⊣⊢. The symbol for normally closed contact is ⊣/⊢

3.  There must always be at least one output on each line. An output is for example, a PLC output relay. The ladder symbol for a PLC output is drawn either as two parentheses close together, i.e. or as a circle.

4.  The numerical assignment for the inputs and outputs form part of the ladder diagram.

5.  Other elements such as timers, counters and shift registers can be implemented in ladder diagrams [3]



*Example circuit*                    *Ladder diagram*
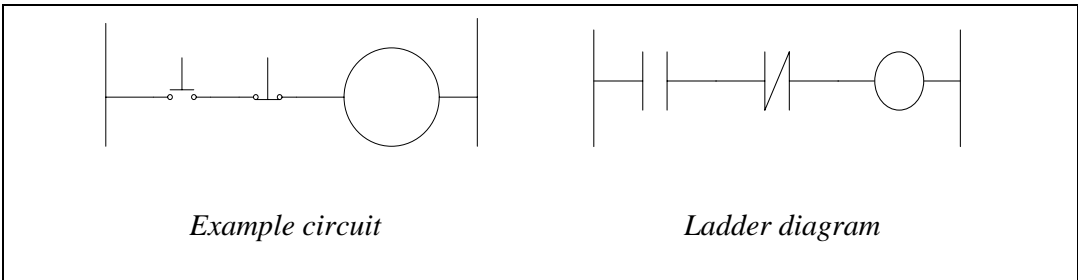
**Figure 2.1**:  The circuit and it ladder logic diagram

The term of ladder logic diagram is used because the lines of a completed diagram resemble the rungs of a ladder. Figure 2.1 shows the example of the circuit and it related ladder logic diagram. The two vertical lines are called bus lines and represent the power connections, in this case 24 V and 0V. Each horizontal line represents a program line. The output on a program line is energized turn on when the inputs contacts to it are made when the contacts connect to 24 V supply to the coil.
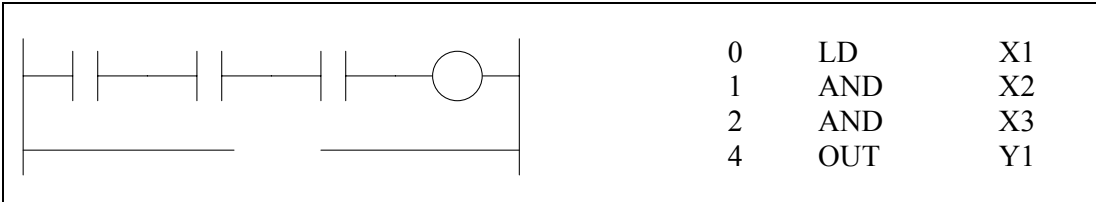
A ladder diagram can be translated into a program consisting of instruction and data. Ladder diagram are entered into memory in an address-instruction-data format. The address is a numbered, which activates a memory location. Instruction and data are entered into a sequential memory locations usually starting from address zero. [4]

## 2.3    Representative the Logic Circuit.

The basic gate can be represented with the ladder logic diagram and it was the basic in PLC programming. Then, this ladder diagram has been converting to the mnemonic instruction.  This instruction is call logic command set. This logic command set is different depending on the PLC manufacturer but it gives the same meaning regarding on its control circuit. Every command/instruction consist 2 parts; mnemonic operation or opcode and address or operand component.

| Opcode | Operand (unique address) |
|--------|--------------------------|
| OUT    | 41.02                    |

In designing the ladder logic circuit, the designer should know how to design the basic gate before designing the complex circuit. The AND gate can be represented with a serial of switch and can be design in Figure 2.2 but OR gate can be represent as parallel of switch and representative as ladder logic diagram as shown in Figure 2.3. For the NAND gate and NOR gate, there are can be represented in Figure 2.4 and 2.5 respectively, which is the NOR gate can be draw as OR gate which is normally closed. The program is beside the ladder logic diagram. [2]
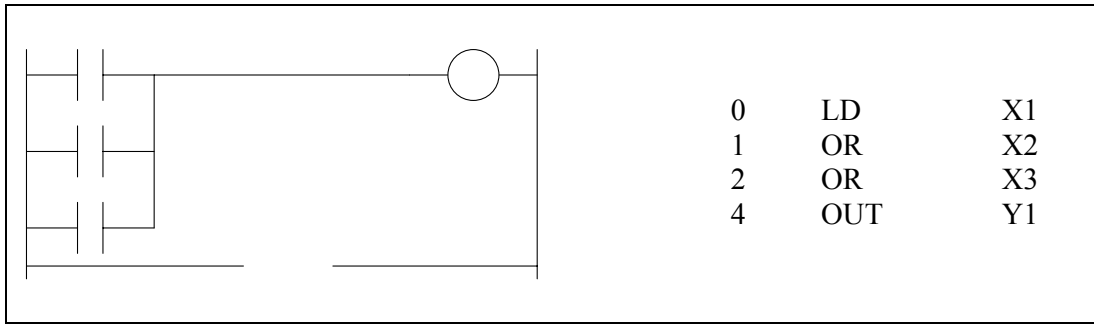


| 0 | LD  | X1 |
|---|-----|----|
| 1 | AND | X2 |
| 2 | AND | X3 |
| 4 | OUT | Y1 |

**Figure 2.2**: AND Gate

8

**Figure 2.3**: OR Gate

```
0    LD        X1
1    OR        X2
2    OR        X3
4    OUT       Y1
```



**Figure 2.4**: NOR Gate

```
0    LD        X1
1    ANDNOT    X2
2    ANDNOT    X3
4    OUT       Y1
```



**Figure 2.5**: NAND Gate

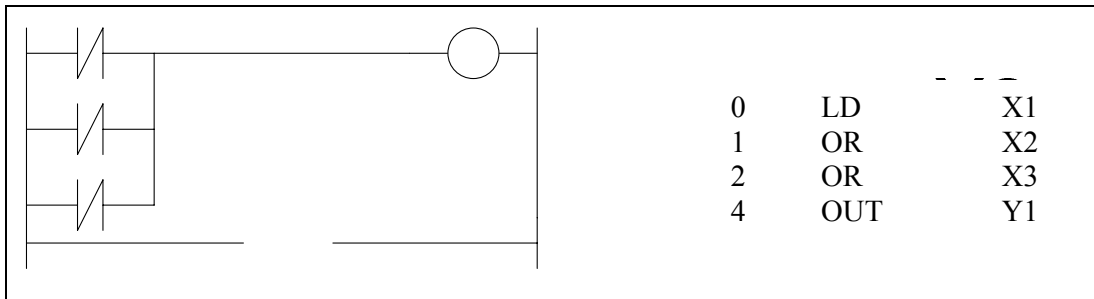```
0    LD        X1
1    OR        X2
2    OR        X3
4    OUT       Y1
```
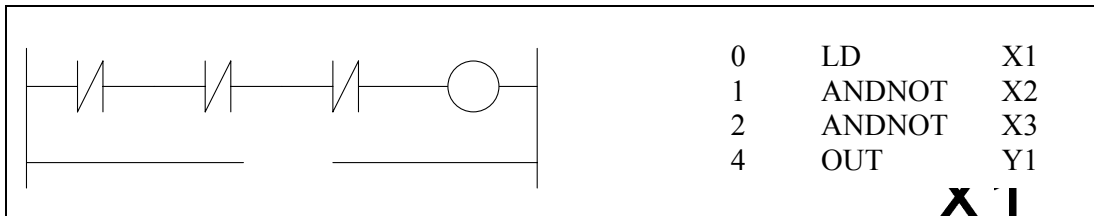
EX-OR Gate can be design by using OR Gate which is the output give 1 when one of the inputs is 1 and the output is 0 when all the input is the same. The Figure 2.6 shows the ladder logic diagram and it related program beside it. [2]
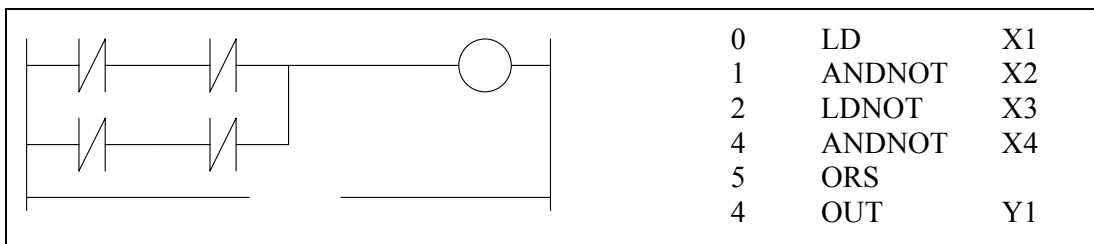


**Figure 2.6**: Gate EX-OR

```
0    LD        X1
1    ANDNOT    X2
2    LDNOT     X3
4    ANDNOT    X4
5    ORS
4    OUT       Y1
```
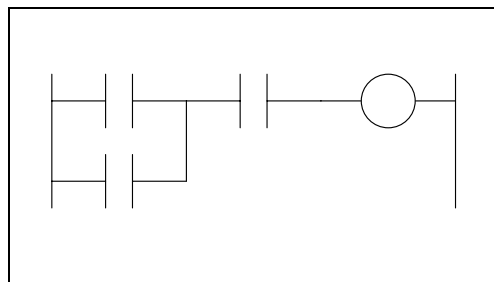
## 2.4 Programming

The PLC scanning process is such that conduction through contacts is from left to right. Contacts are always horizontal branches as the scanning process does not allow for vertical branches. Ladder circuit comprising only contacts and coils are limited to switching operations.

To allow PLC to handle more advanced control tasks manufacturers incorporate a set of special functions. Functions sets vary from PLC to PLC but invariably include counters, timers and data-handling routine. Thus, ladder diagrams comprise contacts, coils and special functional coils or blocks. [3]

### 2.4.1 Latching

A latch circuit is used to hold a coil energized even if the input contact which energized it subsequently turns off. In the Figure 2.7, when contact IN1 is momentarily turned on, control relay CR1 is energized. The logic state of CR1 is fed back as an input to keep itself energized when IN1 turns off. The latch circuit is broken (reset) by momentarily turning on IN2 so that it opens



**Figure 2.7**: Latch

The ability to latch on to an input is possible because ladder programming allows an output (coil) to be fed back as an input (a contact). It is often necessary to retain an output latch during power failure. Battery-backed auxiliary relays can be used for this purpose. Thus, when power is stored, the coils still remain latched. Some PLCs

use functions for latching and unlatching coils so that the programmer does not have to draw a lathing circuit. [3]

### 2.4.2    Timer

Timers are output instructions that are internal to the programmable controller. They are capable of providing timed control of devices that they activate or deactivate. Timing operations are used in many industrial applications. PLC timers perform various functions such as delaying an action, causing an operation to run a predetermined period of time, or recording the total accumulated time of continuous or intermittent events. They can also operate as stable or one-shot multivibrators. [7]

The timers count seconds or fractions of seconds by suing the internal CPU clock. They permit delay of the time of certain control operations. The present value of a timer is the delay period required and is typically set in the range 0.1 – 999 seconds in steps of 0.1 seconds. Figure 2.8 shows the symbol of the timer that has been used by PLC Omron.
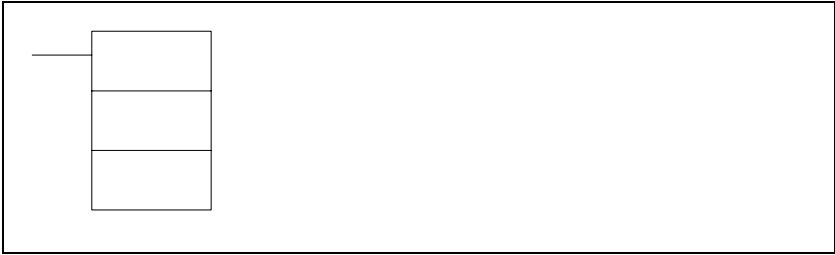


**Figure 2.8**: Symbol of the timer

A timer is activated when its execution condition goes ON and is reset (to set value) when the execution condition goes OFF. Once it activated TIM decrements the Present Value in units of 0.1 second from the set value. If the execution condition remains ON long enough for TIM to time down to zero, the Completion Flag for the TC number used will turn ON and will remain ON until TIM is reset (that is, until its execution condition goes OFF)

Figure 2.9 shows the example of timing circuit. Initially, CR1 is de-energized and CR2 is energized. When the contact labeled IN1 is turned on the timer starts measuring time. When 5 seconds have elapsed the timer turns on and its contacts energized CR1 and de-energized CR2. The timer is reset to initial state by turning of IN1. Timers maybe cascaded or link together to give large delay times. [3]
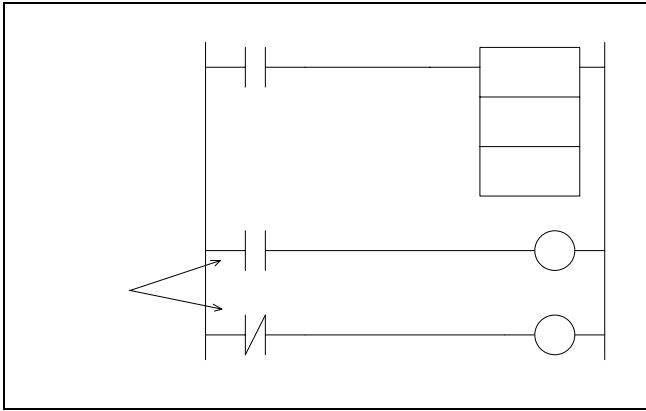


**Figure 2.9**: Timer circuit

### 2.4.3   Counters

Counters are used for counting a specified number of contacting operation. The ladder diagram representation of a counter involves two coils, one to count input pulses and one to reset the counter. All counters have an associated constant which represents the count required. This constant is called the preset value. Figure 2.10 below shows the example of counter operation. [3]
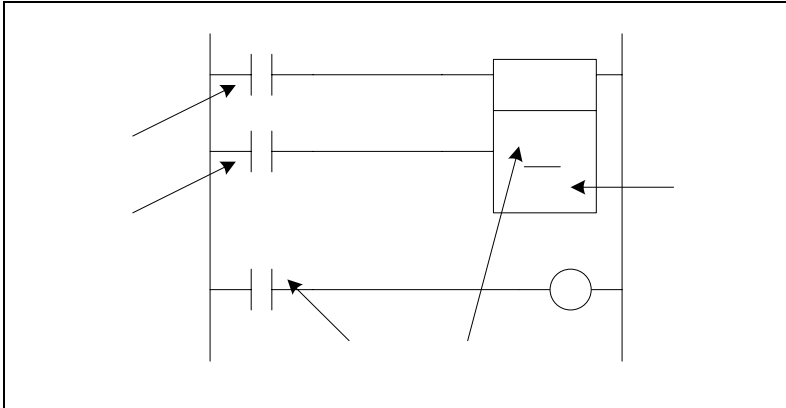


**Figure 2.10**: The example of counter operation

12

In this example, the counter that has been used was down type. The set value is 2. When the contact 000 is turn on, the counter starts to decrement the set value. At this moment, when the contact 001 is triggered, the counter reset the set value to 2. When the set value is equal to zero, the contact CNT020 trigger and energized the coil/output 1000. At this moment, when the contacts 001 is trigger, it will reset the counter and the coil 1000 is stop energized because the CNT020 is turn off. [2]

### 2.4.4 Jumps

The jump instruction is often necessary to jump over part of a ladder circuit when a contact changes its logic state. To skip ladder rungs a jump function is used. The Figure 2.11 shows the jump function. When contact 000 is turned on the scan does not execute the part of the program between the jump and jump end function coils. [5]
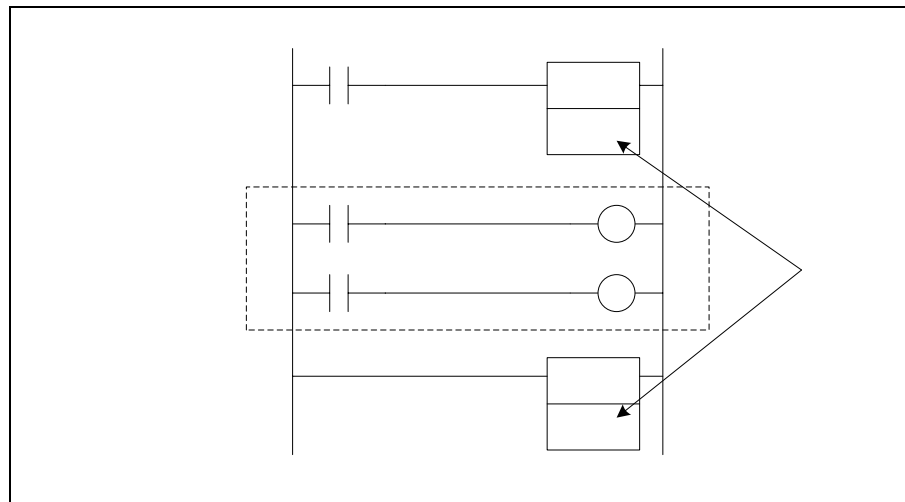


**Figure 2.11:** Jump function

**2.4.5  Keep**

Keep instruction is used to turns ON the output and to maintain its condition. The output is start ON when a set trigger contact trigger and its keep maintain the condition, and until the reset trigger is trigger, the condition is reset. Figure below shows the example of Keep function and its explanation.
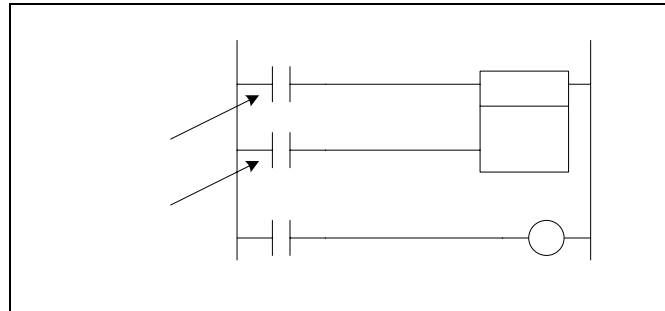


**Figure 2.12**: Keep function

When the set trigger turns ON, output of the specified relay goes ON and maintains its condition. In this moment the contact 0.02 energized the coil 10.00. Output relay goes OFF when the reset trigger turns ON and in this moment, the contact 0.02 are disconnecting and coil 10.00 stop energizing.  The output relay is ON state is maintained until a reset trigger turns ON regardless of the ON or OFF states of the set trigger. If the set trigger and reset trigger turns ON simultaneously, the reset trigger has priority. [2]

**2.5  CPM2A Communication Function**

In this section describe how to use CPM2A communication functions. The following types of communication are being described: Host Link, No-Protocol, NT Link or one-to-one link communications.

### 2.5.1    Link Communications

Host Link communications are a conversational-type communications protocol, in which the PC sends responses to commands issued from a host computer and can be used to read or write data in the PC's data areas and control some PC operations. There is no need for a communications program in the PC. Host Link communications can be used through the peripheral port or the CPM2A's RS-232C port. Figure 2.13 shows the host link communication and figure 2.14 shows the one-to-one communications. [1]
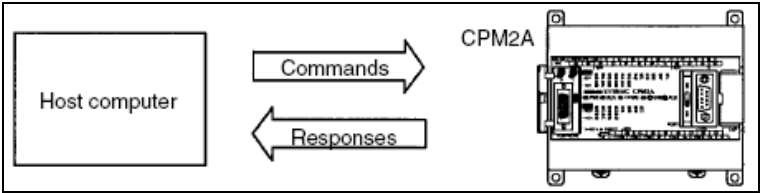

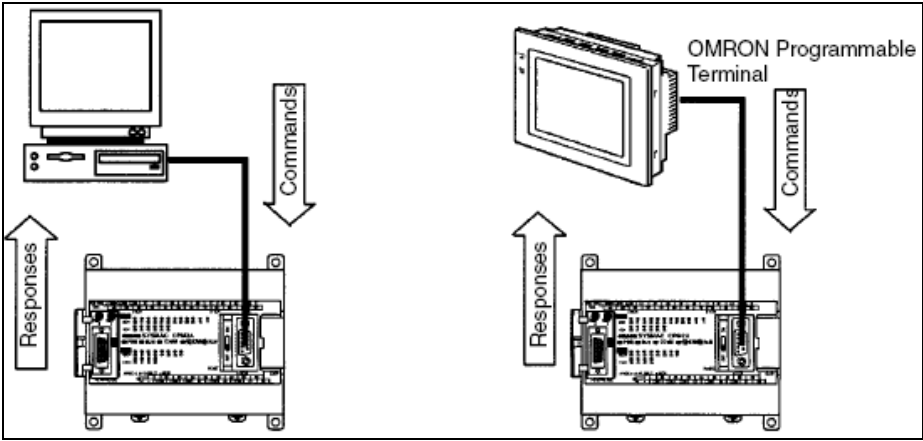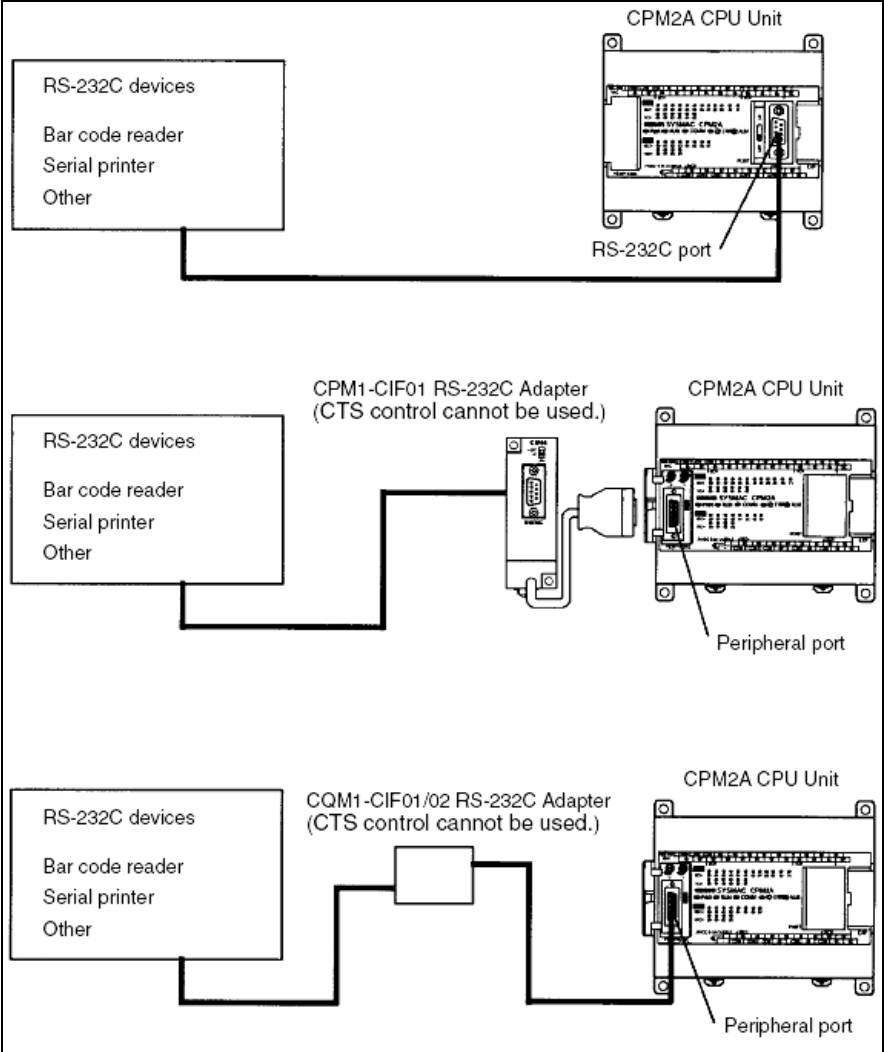
**Figure 2.13**: Host link communication.



**Figure 2.14:** One-to-one communications

### 2.5.2 No-Protocol Communication (RS232 Communication)

When no-protocol communications are used, data can be exchanged with serial devices such as bar code readers and serial printers using TXD (48) and RXD (47). No-protocol communications can be used with either an RS-232C port or peripheral port. Figure 2.1.5 shows the connection of No-Protocol Communication. [1]



**Figure 2.1.5**:  No-Protocol Communication
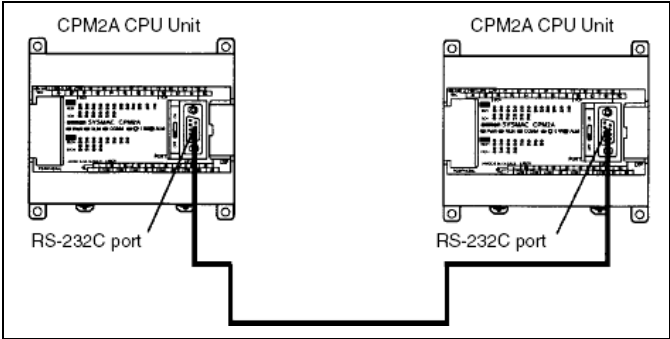
### 2.5.3 NT Link Communication

The NT Link allows a CPM2A PLC to be connected directly to an OMRON Programmable Terminal. There is no need for a communications program on the PC. The NT Link can be used with an RS-232C port. Figure 2.1.6 shows the NT Link Communication. [1]
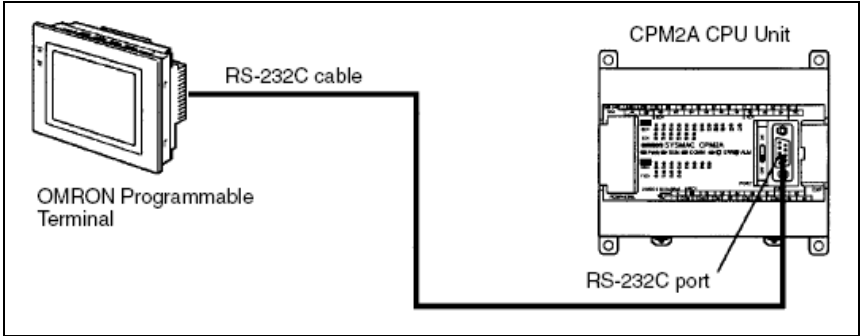


**Figure 2.16:** NT Link Communication

### 2.5.4 One-to-one Link Communication

A One-to-one data link of up to 256 bits (LR0000 to LR1515) can be created with the data area of another CPM2A/CPM2C, CQM1, CPM1, CPM1A, SRM1(-V2), or a C200HX/HG/HE PC, where one serves as the Master, the other as a Slave. There is no need for a communications program on the PC. The one-to-one link can be used with an RS-232C port. Figure 2.17 shows the connection of one-to-one link. [1]



**Figure 2.17:** One-to one Link communication.
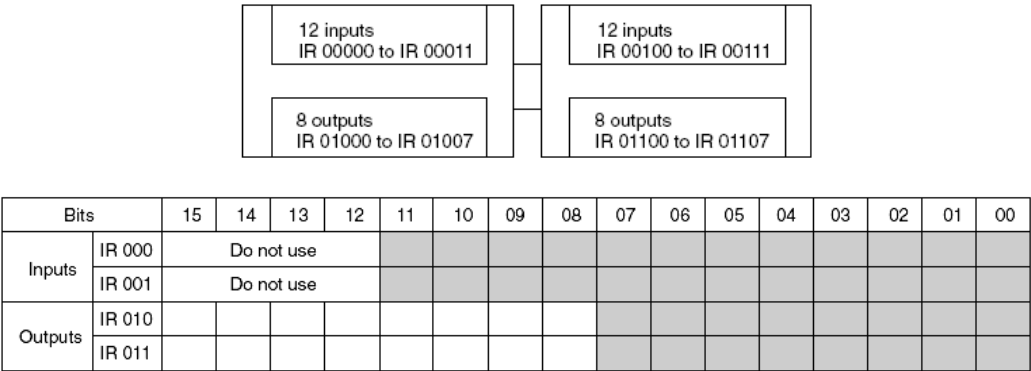
## 2.6   I/O Allocation for CPM2A 20CDR

Figure 2.18 below shows the Input/Output allocation address for CPM2A 20CDR. This information is important to know before designing the program. Different type of PLC has different I/O allocation address. [6]

12 inputs
IR 00000 to IR 00011

8 outputs
IR 01000 to IR 01007

| Bits | | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inputs | IR 000 | Do not use | | | | | | | | | | | | | | | |
| Outputs | IR 010 | | | | | | | | | | | | | | | | |
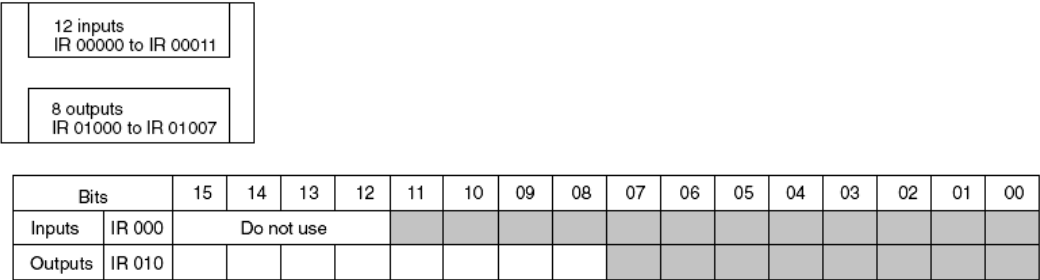
**Figure 2.18**: I/O Allocation Address for CPM2A 20CDR

## 2.7   Expansion I/O

The definition of expansion I/O is to expend or add the numbers of input or output. When a PLC is having a limitation of I/O, expansion I/O are required. The example in Figure 2.19 shows the I/O allocation when using the expansion I/O with 20 I/O pints to the CPM2A with 20 I/O points. [6]

12 inputs
IR 00000 to IR 00011

12 inputs
IR 00100 to IR 00111

8 outputs
IR 01000 to IR 01007

8 outputs
IR 01100 to IR 01107

| Bits | | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inputs | IR 000 | Do not use | | | | | | | | | | | | | | | |
| | IR 001 | Do not use | | | | | | | | | | | | | | | |
| Outputs | IR 010 | | | | | | | | | | | | | | | | |
| | IR 011 | | | | | | | | | | | | | | | | |

**Figure 2.19:** CPM 2A with 20 I/O points + Expansion I/O Unit with 20 I/O points

## 2.8 IR Transceiver Circuit

Figure 2.20 shows the example of IR (infrared transceiver) circuit. L1 is IR transmitter and T1 is infrared receiver. The combination of L1 and T1 should be mounting like the diagram as shown in figure 2.21. The enough space for the widest car should be leave. The IR transceiver should be aligning. When a car blocked between IR transmitter and IR receiver, the red LED will be light up. [8]
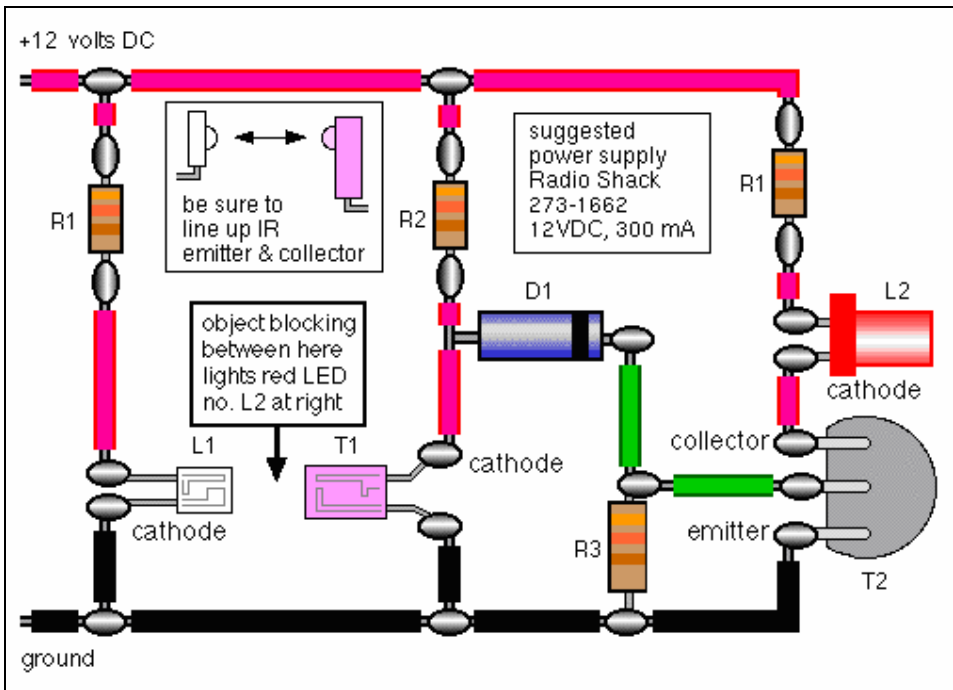


**Figure 2.20:** The example of IR transceiver circuit.
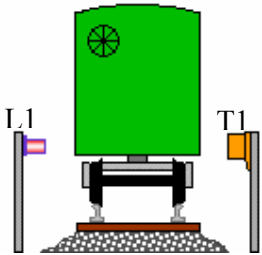


**Figure 2.21: M**ounting the IR transmitter and receiver.

Table 2.0 shows the components list that has been used in IR transceiver circuit as shown in Figure 2.20.

| | |
|---|---|
| R1 | 470 Ω 1/4W resistor |
| R2 | 10 kΩ ¼ W resistor |
| R3 | 4.7 kΩ ¼ W resistor |
| D1 | 1N4148 diode |
| L1/T1 | Infrared LED and detector set |
| L2 | Red LED |
| T2 | MPS 3904 transistor |

**Table 2.0**: Components List