An overview of Computerised Numeric Control (C.N.C.) and Programmable Logic Control (P.L.C.) in machine automation

By Pradeep Chatterjee ,
Engine Division Maintenance,
TELCO,
Jamshedpur –831010
E-mail: pradeep@telco.co.in
pchats@hotmail.com

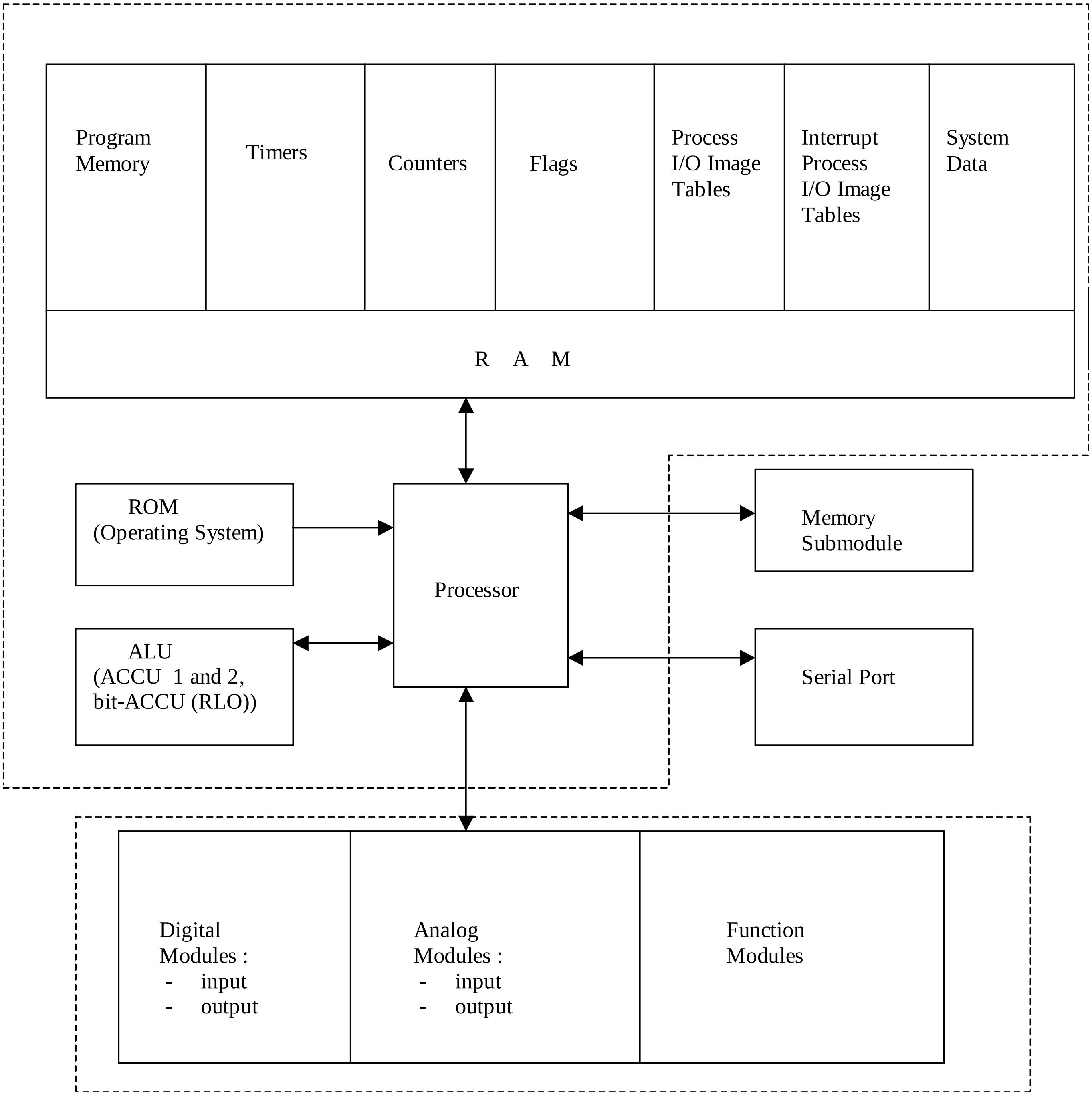
Abstract: Throughout the ages people have been working in research institutes and industries in machine automation so as to have less involvement of operator in machine operations. In the development of automation controllers the trend has been to move towards soft controllers so as to provide better control, more flexibility and more reliability with intelligent diagnostics of machine faults. So industries have gradually moved from conventional relay logic control to programmable logic control and then to computarised numeric control.

Here we shall deal how a programmable logic controller works in automation and the control of a computarised numeric controller on P.L.C. and servo drives to provide automation with accuracy.We shall further deal with the working of self diagnostics of machine faults and the flexibility it provides.

Introduction: Since there were problems related to large electrical panels with a number of electrical components and extensive wiring, people felt the need for software logic controllers. So they gave birth to Programmable Logic Controller (P.L.C) wherein the control logic is developed in ladder diagram,a software logic control, with a number of inputs taken from the environment and generating the outputs ,depending on the logic programmed, to the environment. This helped to control any machine sequence with small electrical panels, less number of electrical components and less wiring with more flexibility to change machine sequence and providing ease to reusability of the components.

These fulfilled some of their needs but the desire to obtain software controlled automation with accuracy necessitated the development of Computer Numeric Control (C.N.C). Here the controller acts as a master over the P.L.C. and connected with encoders to provide positional feedback .In order to have better control for positioning, controlled speed and quick reversal of direction of slide movement, servo motors having less inertia, along with servo drives are interfaced with the C.N.C . The user based part program decoded by the C.N.C. provides the desired profile with the control being transferred to P.L.C. and drives as and when required. Special measuring devices as locators, gauges being interfaced with the C.N.C. through independent controllers provide more accuracy.

Programmable Logic Controller[1]: The diagram below shows the functional units of a **typical P.**L.C.



Program Memory (EPROM/EEPROM): The Program Memory shown in the functional block is a specified area of the CPU RAM . This memory has the following characteristics:

- Memory contents can be changed quickly
- Memory contents are lost in case of power failures if there is no battery backup. It is always advisable to store the program in a memory submodule (EPROM/EEPROM) from where the program can be copied to CPU RAM in case the program is lost or corrupted.

Operating System(ROM): It determines how the user program is executed, how inputs and outputs are managed, how the memory is divided, and how data are managed. The operating system is fixed and cannot be changed.

Process Image Tables (PII,PIQ): The signal status of input and output modules is stored in a specified area of CPU RAM which is known as the "process image table".

Input and Output modules have the following separate image tables:

- Process image input table (PII)
- Process image output table (PIQ)

Serial Interface: Through the serial port programmers, operator panels and monitors are connected through cables and the CPU can be connected as a slave to a local area network.

Timers, Counters, Flags: There are Timers, Counters and Flags available internally in the CPU memory that are used by the control program. The program can set, delete, start and stop the timers and counters and the values are stored in reserved area of RAM memory. A separate area in the RAM memory stores some intermediate results as flags.

Input/Output Modules: These modules transfer information between CPU and process peripherals as sensors, transducers and actuators. These are of the following type:

- Digital Input and Output Module
  - These involve signal states "0" and "1" only

- Analog Input and Output Module
  - These record and generate variable quantities as voltage and current

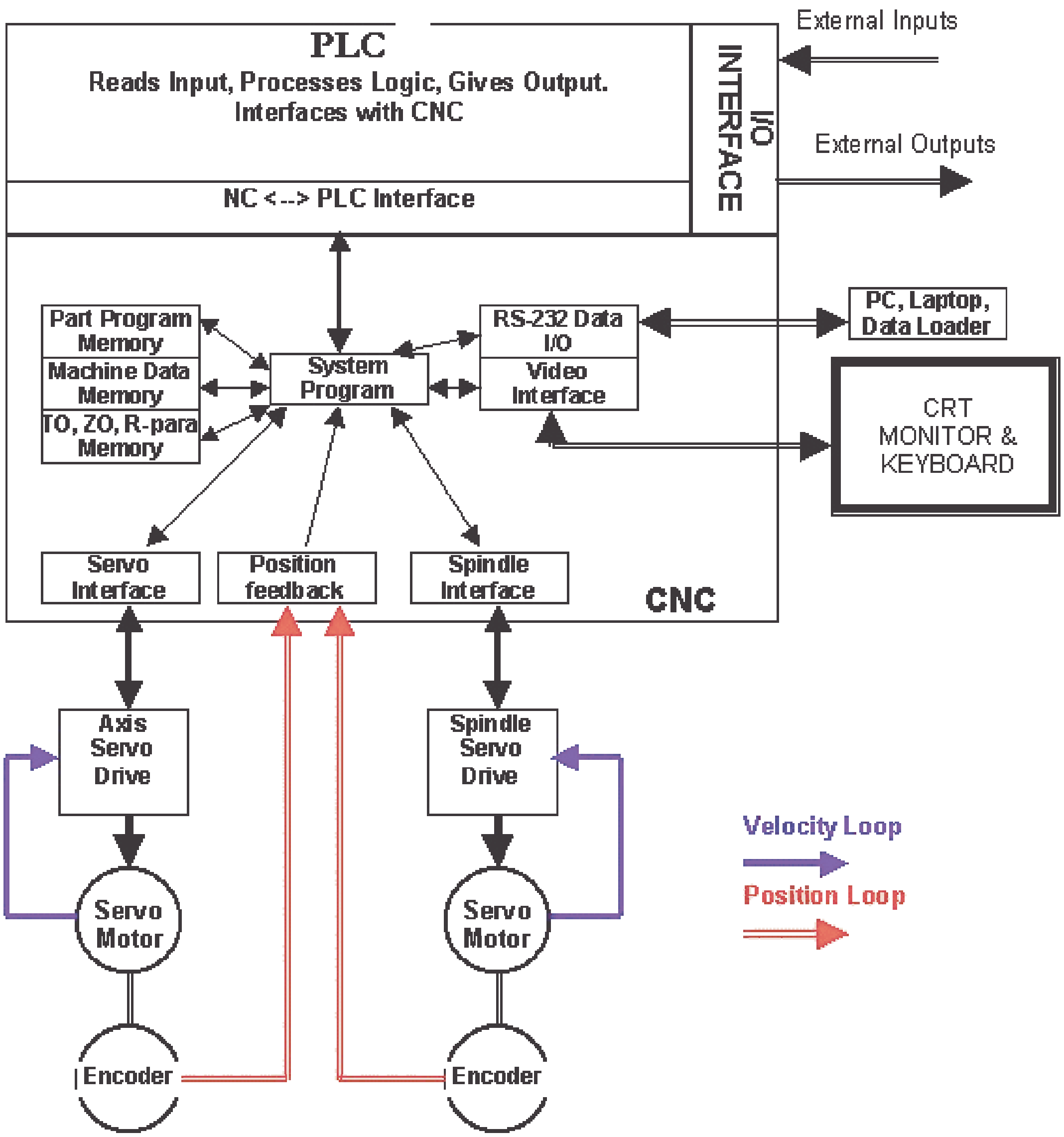
- High-speed Counter and Position detection Module
  - These are used to record counters pulses of 25/500 kHz(typical value for Siemens S5-100U PLC). It can be used for position detection tasks.

Processor: Depending on the control program the processor calls the statements from the program memory in sequence, executes them, processes information from PII and sets PIQ.

Computer Numeric Control (CNC): The major components associated with a CNC system are power supply, PLC, servo drives, CRT monitor and keyboard and some special controllers might be interfaced for specific accuracy requirements such as precise positioning of job, accurate sizing of any machining operations. The power unit supplies constant 24V D.C. to the system. PLC interfaced with the CNC is programmed for the machine sequence. The part program of the CNC determines the operational sequence of the machine. To clarify the machine sequence and operational sequence CNC-PLC interface has been dealt later in this paper. Servo drives are used in conjunction with servo motors for better variable speed control with less response time as the inertia of the rotor is less. The diagram[2] shown below shows the CNC block diagram with the necessary interfaces for a single axis machine.
There are two types of servo drives :
(1) Constant torque-speed drives used for axis motors
(2) Constant power drives used for spindle motors
The external inputs and outputs from the machine are connected to the PLC through the I/O interface of PLC. The system program of the CNC executes the part program, reads the machine data ,perform the calculations. CNC reads in the signal from the axis encoder from the measuring card which is placed in one of the slots of the controller. Thus having the machine data as pitch of the ball-screw, number of encoder pulses generated and resolution of the encoder mounted on the slide the distance moved by the slide can be calculated as

Distance = $(p/r)*e_p$ (here backlash compensation is not taken into account)

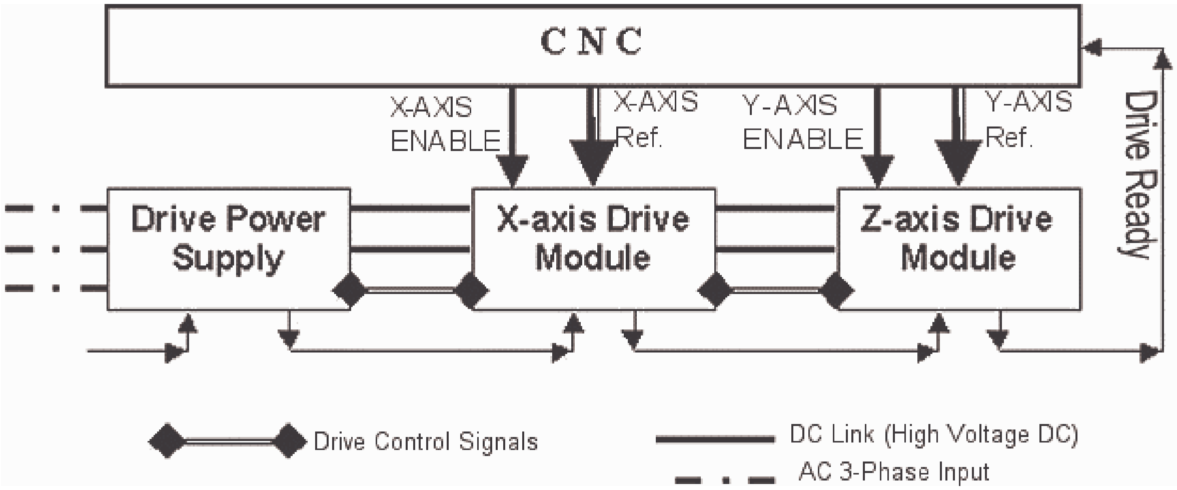
Where p = pitch of ball-screw (mm)
r = resolution of encoder in (ppr)
e_p = number of encoder pulses generated



CNC block diagram

This provides the positional feedback to the system. The velocity feedback is directly fed to the drive from the encoder or tachogenerator of the servo motor.

The diagram[2] below shows the interface of the CNC with the servo drives. CNC sends a drive enable signal to a drive and the drive tends to retain it’s position unless some command for movement is send from the CNC. To give feed command CNC along with the drive enable signal sends a reference voltage to the drive in the range of +/-10V proportional to the desired speed. The +/- sign denotes the direction of movement.



Drive-CNC interface diagram

The reference voltage is calculated as:

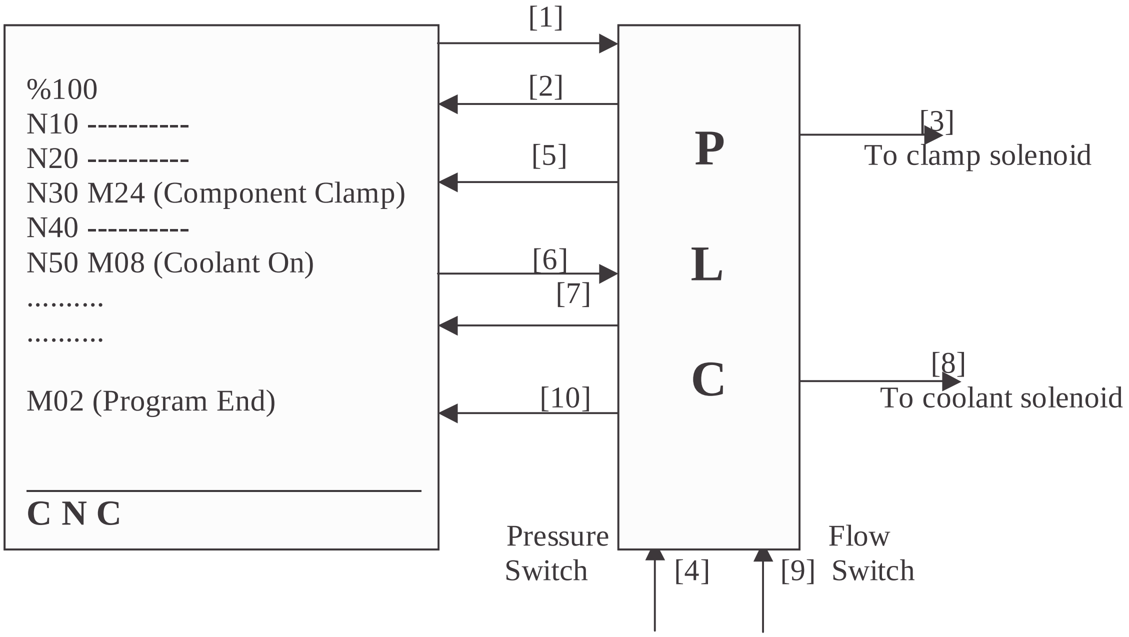
Reference voltage = (10/n)*S

Where n = maximum speed i.e. rated speed of motor (r.p.m)
S = desired speed (r.p.m)

The drive calculates the commanded speed from the reference voltage and calculates the actual motor speed from the motor encoder feedback to the drive. Thus the error is known and by changing the frequency and phase voltage drive tries to minimise the error to maintain the desired speed. If the drives and motors are in healthy condition then a drive ready signal is send to the PLC. If this signal is missing then fault appears and axes movements or spindle operations are not possible.

The diagram[2] below shows a CNC-PLC interface of a CNC system. The part program determines the operational sequence of the machine. It executes the part program as follows:

N10 ....
N20 ....
N30 CNC sets a predetermined flag in PLC for M-code M24(signal 1)
PLC resets “read in enable” (signal 2) in CNC to stop furthur execution of part Program. PLC sends an output (signal 3) to clamp solenoid from the ladder



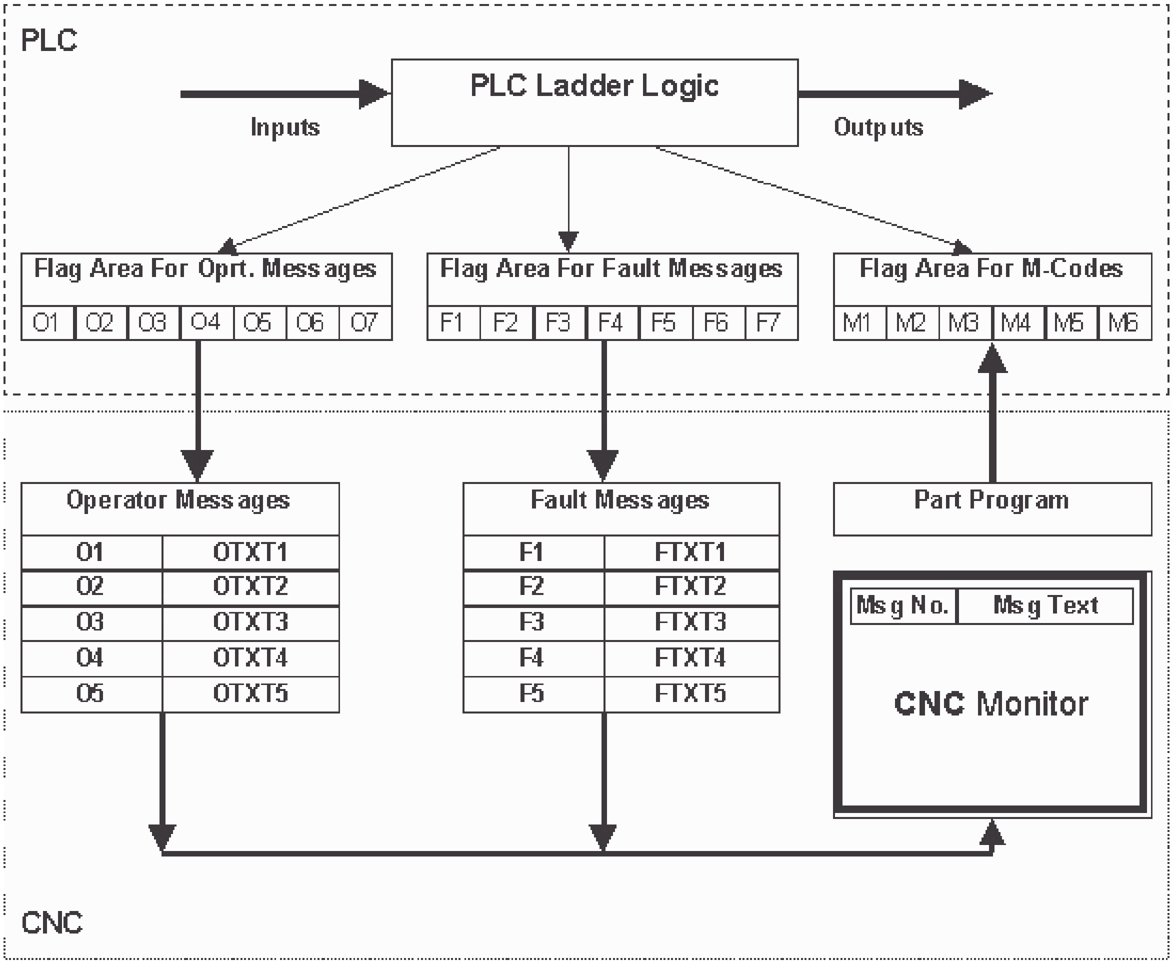
PLC-CNC interface diagram

logic. Pressure switch signal comes as input to PLC(signal 4) once clamping operation is completed. PLC sets the “read in enable” (signal 5) in CNC and execution of part program is resumed.

N40 ....

N50 Similarly M08 sets a PLC flag (signal 6).PLC resets “read in enable”in CNC (signal 7).PLC sends output to coolant solenoid (signal 8).Once coolant flow starts flow switch sends input to PLC (signal 9). PLC sets “read in enable” (signal 10) in CNC and part program execution is resumed till M02.

Intelligent Fault Diagonistics : A CNC system is capable of diagonising some of the machine faults and prompt the operator or maintenance personnel.It is mainly monitored by the PLC logic. There are preset flags for operator messages, fault messages and M-codes as shown in the diagram[2] below. When PLC generates an output and the corresponding feedback signal is not fed back as input to the PLC within a certain time the corresponding fault flag is set. During the time duration the corresponding operator flag is set which will help to prompt the operator that a certain function is going on. To make the system user friendly certain portion of the CNC system memory is reserved which contain the operator and fault messages in text format. So when a particular flag is set the system displays the corresponding text.



Conclusion : The paper gives a broad overview of PLC and CNC controllers in machine automation. Certain other factors are also involved but it could not be presented in this paper. However the topics dealt with will give a brief idea for the basic functioning of PLC and CNC controllers.

References :

[1] Siemens Simatic S5 system manual pp 2-2 to 2-4
[2] A Course of CNC Maintenence By K.V Ram (http://kv_ram.tripod.com)
[3] Microprocessors and Interfacing (Programming and Hardware)
By Douglas V. Hall , Tata McGraw-Hill