INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

NPTEL
ONLINE CERTIFICATION COURSE

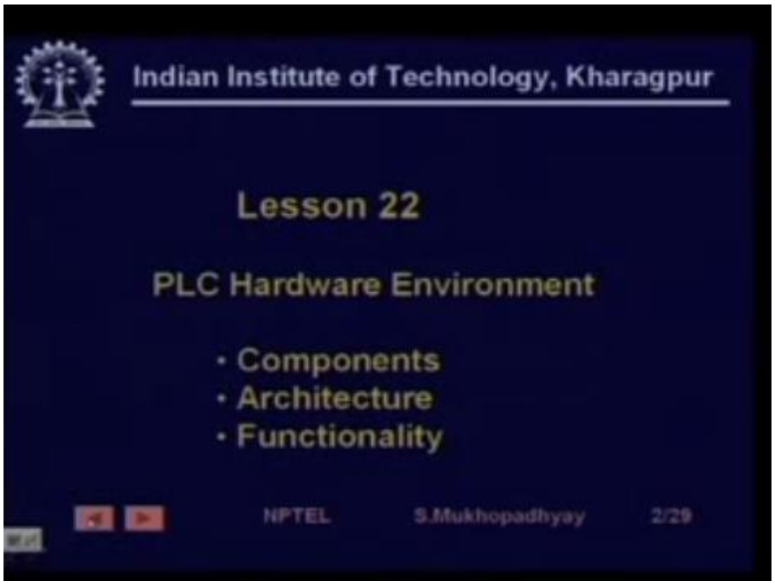
On Industrial Automation and Control

By Prof. S. Mukhopadhyay
Department of Electrical Engineering
IIT Kharagpur

Topic Lecture – 30
PLC Hardware Environment

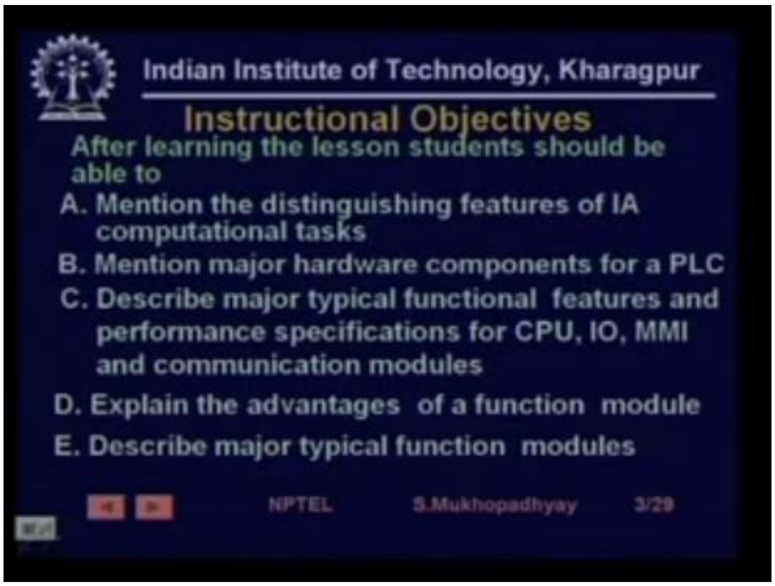
Welcome to lesson 22 so far we have learnt about the basic functioning of a PLC we have learnt how to write a program for it but all this time we have seen the PLC as an abstract device we have not seen what is inside a PLC system what actually physically makes it so in this lesson we are going to look at the hardware environment of the PLC's.

(Refer Slide Time: 00:56)



Basically what PLC systems are made of right so we are going to look at components we are going to look at components of the PLC system their architecture by which I mean that how they are organized how they are connected and their functionality that is describing what they do okay so now so let us see before we begin it is customary to see the instructional objectives.

(Refer Slide Time: 01:40)



So the instructional objectives are the following first after going through this lesson one should be able to mention the distinguishing features of industrial automation computational tasks a PLC basically is a computer so it computes computation it performs computational tasks related to industrial automation but these tasks have certain very distinguishing features compared to the other tasks which are let us say done in an office environment so we are going to you should be able to mention some of these features mention major hardware components of a PLC described major typical functional features and performance specifications for the CPU central processing unit IO are input output MMI the man-machine interface and communication modules.

Okay then explain the advantages of a function module so what are the advantages of a function module you will be able to explain and describe some major typical function module types so these are the instructional objectives.

(Refer Slide Time: 03:00)

So let us start with the industrial automation tasks and their distinguishing features, so we are going to look at the nature of industrial automation what I call IA computation that is computation related to industrial automation these kind of computations remember that are always working on physical signals right, so there are temperatures and pressures and limit switch positions starter positions etc…

Which are all very physical signals electrical in nature which are interfaced with the PLC the PLC looks at it is values and take this some of these are inputs and computes the output which it computes according to some it is software for example relay ladder logic programs okay these tasks requirements are general I mean by standards of other applications using computers let us say let us say let us a digital communication or personal computing these tasks are the computational requirement of these tasks are generally called slow are generally I mean should be slow.

Although among themselves there could be tasks which are very slow for example temperature control tasks are very slow having time constants of the order of minutes let us say flow control tasks would be faster and if you have a machine control or you know you have to have a and actuator control electromechanical control actions so in such cases the tasks are relatively faster but even those you know basically they are all made of to move to move an electromechanical system with a certain finite amount of power generally requires times of the order of milliseconds or at most a fraction of a millisecond.

Before that no perceptible motion will be produced on the other hand these speeds are much slower compared to the speed of electronics which we typically encountered in let us say communication system so in that sense they are slow although within their own category is there can be fast tasks and slowed or medium fast ask or even slow tasks right so that is what I wanted to mean when I wrote slow fast the complexity of these tasks are also not very complex some of them are some they are very simple relay ladder logic diagrams are often very simple but there are other there could be other complicated control algorithms that could be self-tuning algorithms which are which could be somewhat complicated.

So I would say that the complexity of this tasks are you know low to medium not highly complex interestingly many of these tasks are repetitive for example if you take a standard process control loop then every sampling time you have to input the controlled variable and based on your control law sends the set point and then compute the manipulated variable right so this computation of the manipulated variable and inputting the set point and the controlled variable goes on repetitively.

So these are generally repetitive tasks and they are and there real-time any computational task is called real time when there is an there is an explicit time boundary by which this task should be completed and this fact is actually taken into the design of the design of the computational system which is supposed to perform this task for example in a in a in a control loop whatever computation you are having that should be completed within a sampling time.

So there is a there is an associated deadline for completing these tasks once they started so you so you in other words even if you have a very grand control algorithm which takes a lot of time and does not complete within a sampling time is not at all suitable rather less probably a less grand control scheme which will complete within the sampling time will be preferable right.

So in that sense they are real time and for real time tasks you know one has to because you know you have to also realize that within one PLC there will be several control loops working either runs either they are discrete sequential control or their analog controls whatever it is there are a number of control laws which have to be repetitively executed right so there is a requirement of determining that which should be run at what time because some of them will have to be run let us say every 20 millisecond some of them may have to be run every 500milliseconds some of them may have to be run every two minutes right.

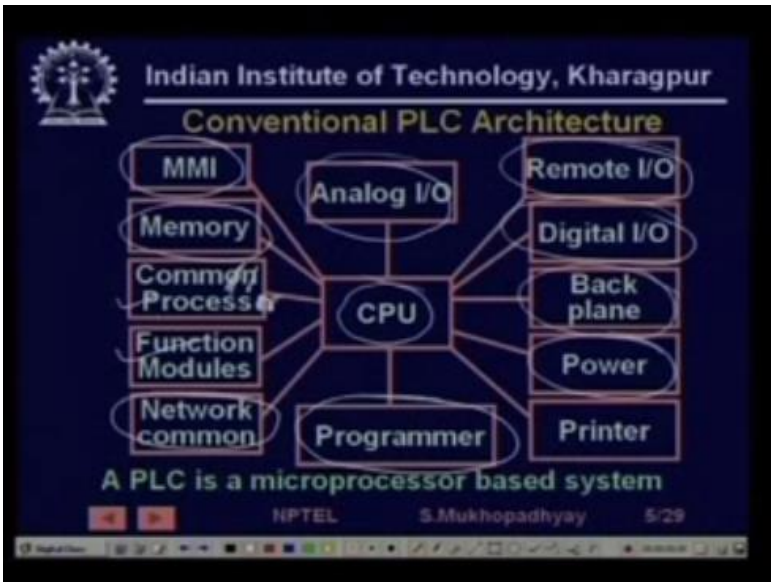
So this is called scheduling so there has to be a mechanism of scheduling tasks at the appropriate time so that computational resources are utilized and all the tasks are finished at their before their deadlines okay so this is generally done using a technology called scheduling and we will not go much into that is that is in the domain of real-time systems but generally into the case of PLC is even when there they are scheduling PLC Scheduling are generally very simple and the static in the sense that they are the strategy which is adopted is fixed it does not change according to situations generally.

Similarly they have very rudimentary kind of operating systems much less generally much less sophisticated than standard other operating systems like UNIX apart from that but one thing is very interesting that when you have a PLC I mean these computational tasks are very mission critical in nature in the sense that we do not want to have any kind of errors in them so they should have very good reliability and they should ensure always ensure worst-case performance specifications we cannot check average case performance specifications like for example you know if you have Ethernet.

Then we know that Ethernet worst case performance is generally very I mean goes down significantly compared to the average case performance so but here whatever technology we are using we should always use what we should generally always use worst case performance as the performance guarantee because the consequences of a task failure could be very great right lastly often it happens that this computational environment computational tasks have to be performed at very you know harsh physical environments having lot of dust, heat, moisture, vibrations and such things.

So the hardware has to be such that it is not affected by these one has to device you know proper ceilings one has to device proper cooling mechanisms these are also very important in the case of industrial automation computer systems to ensure that the tasks are performed reliably and as expected so having said that let us look at what makes a PLC.

(Refer Slide Time: 10:59)



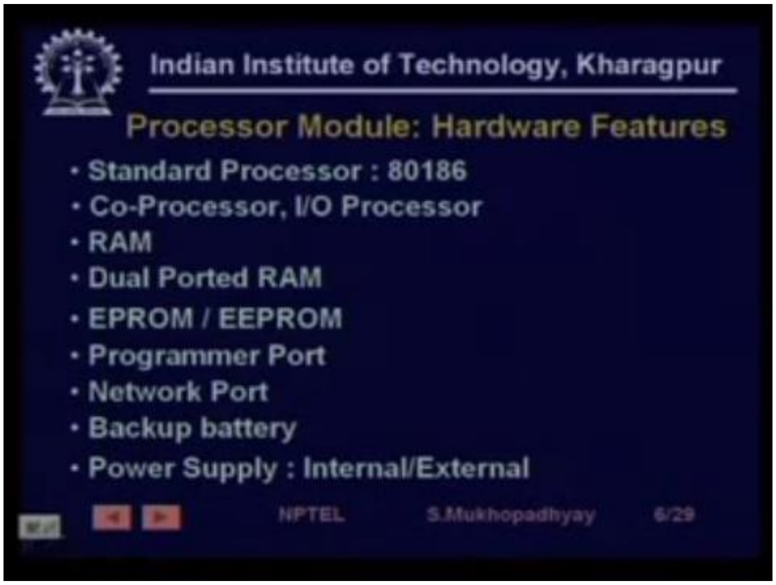
So as I said that a PLC is basically a microprocessor based system so as we know that a microprocessor based system will have the major components are number one CPU number two memory number three input/ output number for power and number five the buses right so here also you will find basically the same things so here you have the CPU this is the CPU and here is the memory as we will see memory can be of various types and you will find you can you can you can easily make out that the PLC applications are very rich in I/O because their basic job is to interact with the physical world right.

So there so you have various kinds of I/O for example you have analog I/O you have digital I/O you have remote or sometimes called distributed I/O you also have network communication much of it is again I/O so you have various kinds of I/O you have man-machine interface which is also kind of I/O input output then you have the you have a programmer which by which you can develop a I/O then you have power you have the backplane that this is the bus which connects the various PLC elements.

Sometimes you have you must be knowing that sometimes you have you know other processors which assists the main processor so you could have a this is not common this is wrong what I meant to write is a communication process server right so please correct this by which I wanted to mean that this is a communication processor you know because it has to a PLC is has to sometimes communicate with special purpose devices.

So each that job is often given to a separate processor similarly it could have separate input , output processor it could even have a network processor in fact communication processor part of the job is to do communication may be on maybe on the network environment also sometimes you have separate these are actually I/O which are intelligent in the sense that they have their own computing capabilities so one needs to so the main CPU needs to talk to it so therefore this is also a kind of I/O so you see that a PLC is basically an basically a microprocessor based system with a lot of I/O. And we are going to see what these different types of I/O’s are so let us look at these modules at some level of depth now okay.

(Refer Slide Time: 14:28)



So you have a processor a typical processor would be I mean this is just an example it could be 80186 it could also be from the 68,000 line various choices are available sometimes these processes are you know proprietary not standard processes along with the standard processor as I said you could have coprocessors for specific purposes like I/O like I/O like handling programming input etc…

Or managing the network you obviously have RAM. RAM is required for running it you have sometimes you have dual ported RAM dual ported RAM is RAM in which simultaneously one it has it has two ports so it can be simultaneously written in two and read from okay this is important especially when you want to read the data from devices which are working without really stopping the device operation you want to read data or you want to write data so you can on the fly parameterize dumped some new parameters for a PLC while the PLC is working so the PLC continuously uses this memory but in between when during those brief instance when it is not using the memory.

If you have a dual ported RAM then using another processor you could write the same memory and you know download your parameter values in the meantime into the memory which the PLC will use as it gets them okay so this kind of you know reading and writing on the fly by another processor is possible if you have dual ported RAM you have EEPROM or EEPROM these are typically you know read only memories and they are used for storing the PLC programs you have programmer ports which are used for downloading programs you have network ports you have backup battery this is very important because by any chance if you have power failure you do not want number one your process state to be lost number two you do not want to lose your parameters.

So there is a backup battery always which will backup part of the RAM and you obviously need the power supply which is which could be internal within the processor module or it could be external have as a separate power supply module which will collect to the processor module it could be either way.

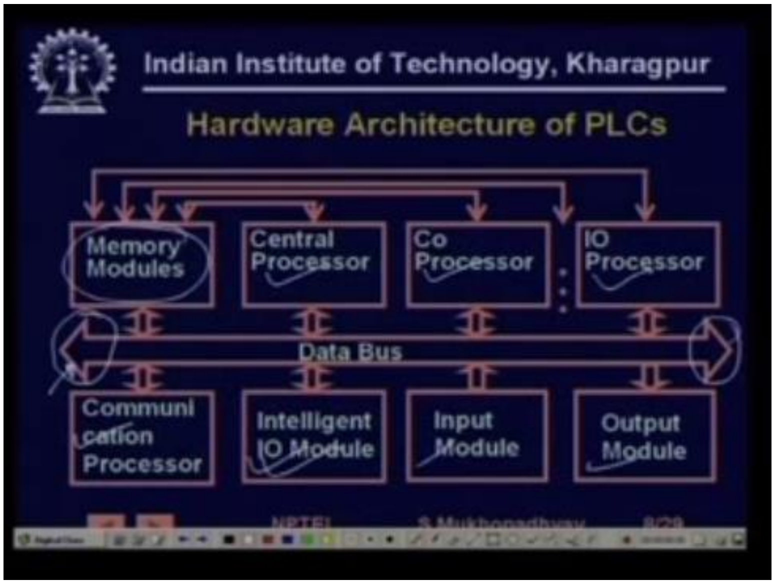
(Refer Slide Time: 17:07)

So let us look at a typical PLC system a picture which will clarify the things so here is a typical PLC system so you can see that it contains multiple CPU units actually you know these are if you look at them that these are the various CPU units okay, so these are the CPU units some of them are communication some of them could be communication processors or other coprocessors one of them will be the main processor there are a number of rack-mounted I/O modules.

So these are the rack mounted I/0 modules I/O racks each one of them will plug in one particular I/O card which will have various which will interface a number of I/O signals and you have the power supply here right so this is how it typically looks you can see a number of you know you can see a number of ports so for example you have a number of ports here right, so these are probably the communication processors we cannot because they are having so many you know RS-232 ports serial ports look like there are there are there are some there are some parallel ports looks like printer ports.

So this is how a PLC actually looks like there are actually this is not the only type PLC systems could be very large they could be very small everything comes into one box that is also possible this is a medium-range kind of application so having looked at that.

(Refer Slide Time: 19:20)



Let us see how these various things are schematically connected we just saw a diagram so you see that this is the scheme in which these various processors and the modules are connected they are all connected by a bus this is and obviously as you have seen that various I/O modules are mounted in racks so there will be a there will be a ribbon cable like thing that be a flat cable which will act as this bus you know it will connect the various modules and sometimes these modules can even may not be situated at the on the on the same row sometimes they will be connected one after the other sometimes they may be connected in a cabinet which is a little distance away.

So this bus really has to extend sometimes using cables also so you have central processor various kind of coprocessors I/ O processor you know nowadays we are we are having the age of you know embedded processors so some of these processes are use even DSP’s so there could be numeric coprocessors or I/O processors generally there are communication processes for specific communication to you know kinds of devices like operator stations or for communication on the network there are various kinds of I/O module some of them have processes on them so they are called intelligent I/O some of them are you know ordinary I/O which just takes signals and feeds it to the central processor.

Plus there are obviously memories so this is the basic hardware architecture which is very common for microprocessor based system they are generally organized around a bus oops and if you see inside one processor that is also pretty standard in the sense that you know you have as you all know that all microprocessor based systems are have two kinds of buses.

(Refer Slide Time: 21:24)

One kind of bus is called a data bus another kinds of bus is called an address bus so using address buses we can talk to specific devices and then we can exchange data over the databases so you know this is the these are the databases connects to various devices and these are the address buses by which the devices can be selected you have the main processor you have various kinds of memory you have application memory where the computational tasks run even for running tasks you need a for managing the resources of the PLC you need some you know programs which are typically called systems programs.

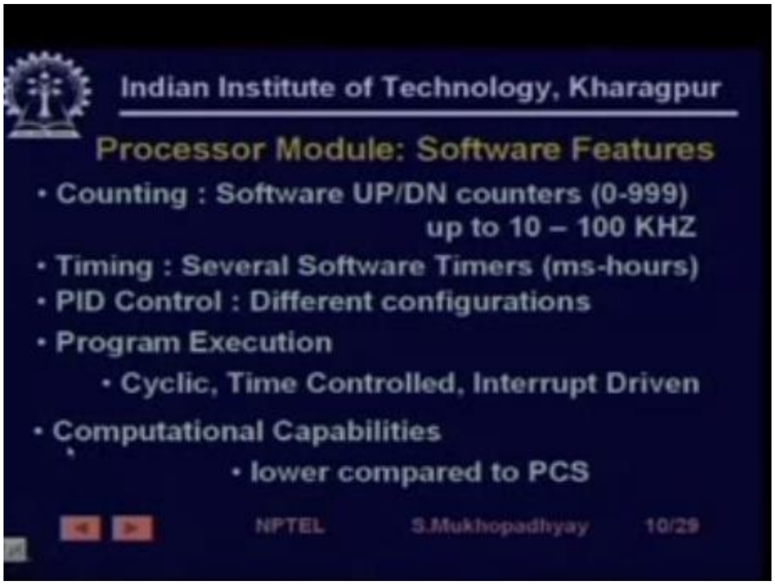
So for that you need systems program memory you need systems data memory you have you know special kinds of PLC systems are you know just like your give us have experienced that sometimes your PC hangs it just sits quite doing nothing somehow it gets mired in to some you know infinite kind of loop so this is prohibited in PLC systems if due to from reasons like wrong programming to memory corruption if the processor gets is stops doing useful work then there has to be explicit mechanisms by which the processor execution must be brought out of it and taken into a safe speed.

So this is typically done by watchdog timers which are hardware devices and sometimes they are also I mean yeah they are actually hardware devices and if these timers find no proper activity over certain periods of time then they by force take the processor to some safe state okay so you could have you know dual ported RAM’s as I said apart from the normal RAM’s obviously the these have there has to be various kinds of interfaces sometimes you know here they have shown here we are seeing a particular another additional processor.

See this is the main processor but we are seeing an additional processor and the job of this processor is to interface with the programmer so that you can while this system is working you know a PLC system is typically interface with the Machine and industrial machines often run continuously so you must provide mechanisms excuse me so that you are able to load new parameters read status you are able to work with the system while the system is working with the machine.

So without stopping it so for that since not only you need a dual ported RAM you need another processor which will talk to the programmer while the processor is well the main processor is actually busy controlling the machine right so this is what is done.

(Refer Slide Time: 24:29)



So typically this processor modules though this is mainly a discussion on hardware we are we would like to mention that this typically this processor modules come with some pre-programmed software minimal P program software more than that you one has to develop or one has to take from additional libraries so you have standard things that some standard things are supported like for example basic counting timing functions some software up down counters quite a bit quite a few of them actually from 0 to 999 typically typical value and several timers which can time of which the time durations of the order of milliseconds two order two of the order of several hours or even sometimes days are available.

You have various you know standard control algorithms already pre-coded so that the users do not have to can easily configure their systems the program execution in these systems is very interesting there are there are generally 3,4 different types of program execution mode specified for example a mode could be cyclic by cyclic we mean that you have a number of computational tasks begin just like RLL program execution so begin here come to the end start all over again so this just goes on typically cycle time remains more or less constant but it could vary a little bit depending on the program logic.

For example if you have a if you have some program control statements if then else kind of statements then whether some block will be executed or not will actually depend on the data so program execution time is not always constant it actually depends on data but roughly it will be constant and in fact it is preferable that it is constant so that you are not surprised that that for some value of data suddenly you are suddenly your program execution takes a very longtime and your deadlines are missed.

So it is preferred that real-time programs have predictable and not too much varying time requirements so for many classes of applications cyclic is good enough so that is the simplest apart from that you could have time controlled executions where certain operations will start after certain times either absolute times or relative times that is start the pump after 30 seconds of opening the valve this kind of controls can be executed.

So exactly 30 seconds after so this execution is timer controlled or time controlled or it could be interrupt driven there are some tasks which can suddenly occur like you know like your failures kind of tasks which once they occur they are not taken into the consideration of the regular computing flow but once they occur they are required to have very high probability very high priority I am sorry and therefore they have to immediately get the attention of the processor.

So such tasks are sometimes coded in the interrupt driven execution mode so whatever the processor was doing it will be interrupted and this requirement will be get into. So but all said and done for a given I mean typically one must understand that typically a PLC has much less computational capability compared to a PC although it is generally much more expensive or that is if you take PLC's.

And PC’s of the same kind of price then you would find that the PLC is much less computationally capable compared to a I mean the normal PC which sells in the market both in terms of memory and in terms of computing speeds that is because you know price of things get actually decided by the sales volume plus the PLC has certain other feature there they are actually very rugged and they are a different kind of packaging.

So all these things make this and the second and the third reason is that a PLC is not a is not a general-purpose computer so I mean no one is going to download nice files from the internet and store it there and then your emails keep filing up such things do not happen so therefore memory requirements are generally well understood and much less than what is required so therefore they are one keeps things only which is required because one wants to make things reliable and putting too many things into machine makes it unreliable right.

So but one has to understand that these are although they are much like PC’s in fact there are the in the market the competitors of PLC's are actually called industrial PC’s which are really PC’s run on I mean no similar operating systems windows some versions of it but they are actually they have generally much lower computational capabilities.