

Controladores Programáveis

AUTOMAÇÃO

PROF. GUILHERME FRÓES SILVA



ESCOLA
POLITÉCNICA

<https://guilhermepucrs.github.io/automacao>

Índice

Histórico

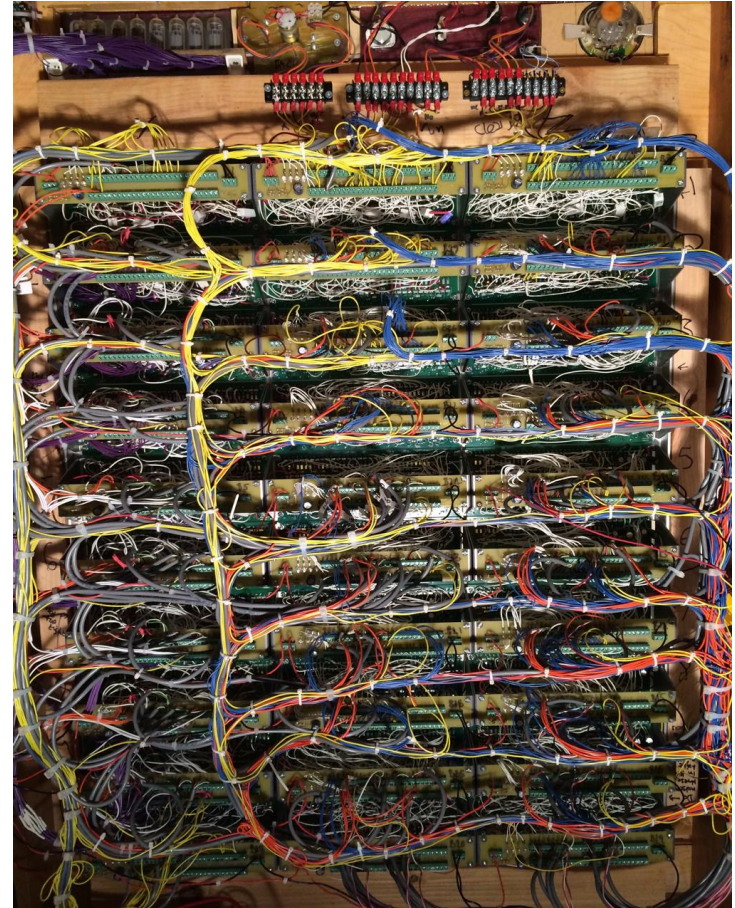
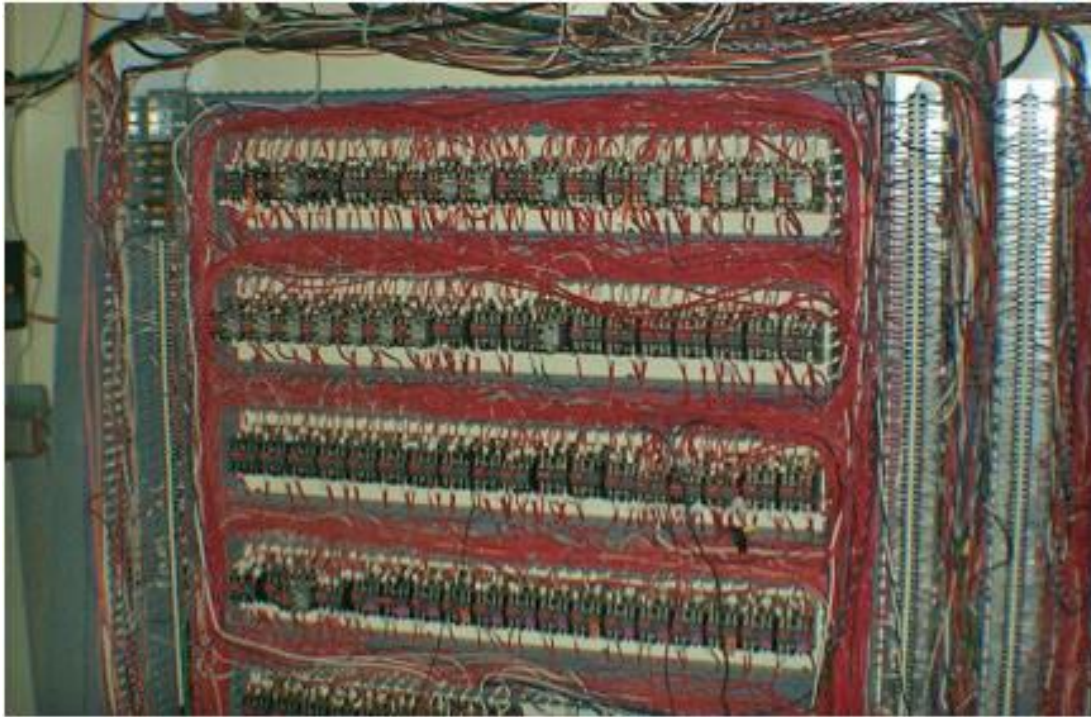
Ciclo de Execução

Linguagens de Programação

Especificações

Nomenclaturas

Histórico



Histórico



Histórico

No fim da década de 1960, os circuitos integrados permitiram o desenvolvimento de minicomputadores que foram logo utilizados para o controle *on-line* de processos industriais.

Histórico

Em 1969:

Facilidade de programação

Facilidade de manutenção com conceito de *plug-in*

Alta confiabilidade

Dimensões menores que as dos painéis a relés, para a redução de custos

Envio de dados para processamento centralizado

Preço competitivo

Sinais de entrada de 115Vac

Sinais de saída de 115Vac

Expansão em módulos

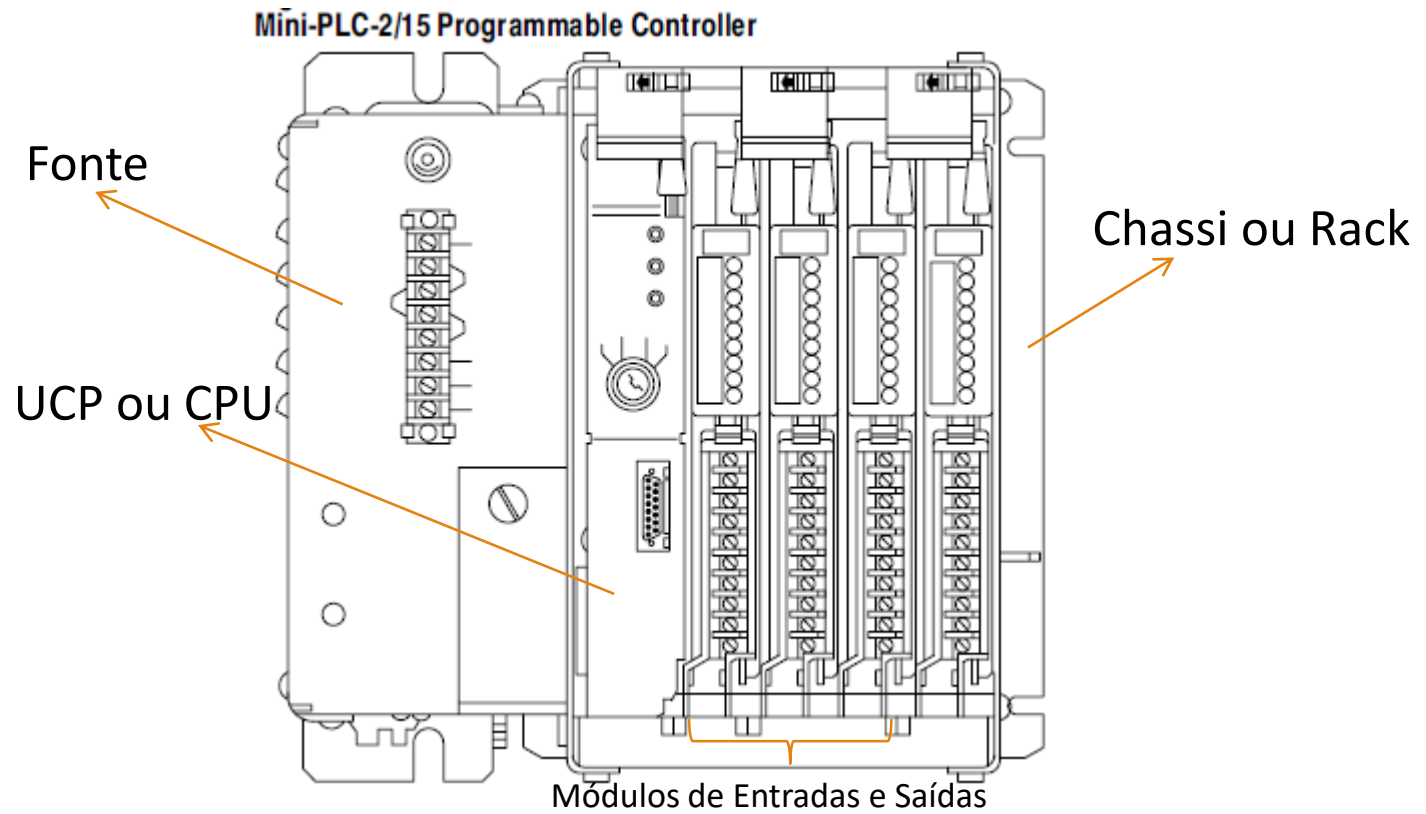
Mínimo de 4000 palavras na memória, ou 4Kword

Histórico

Na década de 1990, houve um aperfeiçoamento das funções de comunicação dos CLPs, sendo então utilizados em rede. Atualmente, as principais características dos Controladores Programáveis são as seguintes:

Linguagem de programação de alto nível, caracterizando um ambiente de programação intuitivo com o usuário. Depois de concluído e depurado, o programa pode ser transferido para os CLPs, garantindo confiabilidade na sua utilização.

Histórico



Histórico

Simplificação nos quadros de painéis elétricos. Como consequência, qualquer alteração necessária torna-se mais rápida e barata.

Confiabilidade operacional: Alterações podem ser realizadas pelo programa do Controlador necessitando pouca ou nenhuma alteração na fiação elétrica e as possibilidades de haver erro nestas alterações são minimizadas.

Funções avançadas: Há grande variedade de tarefas de controle através de funções matemáticas, controle da qualidade e informações para os relatórios.

Comunicação em rede: Através de interfaces de operação, controladores e computadores em rede permitem a coleta de dados e uma enorme troca de dados em relação aos níveis da pirâmide da automação.

Histórico





MicroLogix

Maximum I/O

10 – 32 Discrete I/O

Memory Size

1K Words

Communication

DeviceNet
DH-485
RS-232-C



SLC

84 - 4096 Discrete I/O

1K - 64K Words

Ethernet
DH+, DeviceNet
DH-485
RS-232-C



PLC-5

512 - 3072 Discrete I/O

6K - 100K Words

Ethernet, ControlNet
DH+, DeviceNet
RS-422-A, RS-423-A
RS-232-C



Open Solutions

896 - 4096 Discrete I/O

32K - 100K Words
upto 64Mb RAM

Ethernet, ControlNet
DH+, DeviceNet, VMEbus
RS-422-A, RS-423-A
RS-232-C



ControlLogix

128,000 Digital
and 3800 Analogue

160K – 2M bytes

Ethernet, ControlNet
DH+, DeviceNet
RS-422-A, RS-423-A
RS-232-C



Simatic C7

Maximum I/O

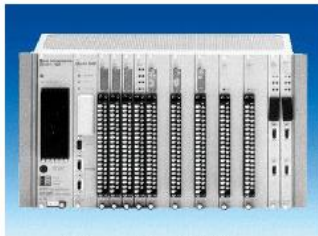
768 Digital
192 Analogue

Memory Size

32 Kb – 128 Kb

Communication

Industrial Ethernet &
PROFIBUS-DP & FMS



Simatic 505

2,048 Digital
1024 Analogue

96 Kb - 192 Kb

Industrial Ethernet &
TCP/IP TIWAY,
MODUS,
PROFIBUS-DP & FMS



Simatic S5

2,048 Digital
384 Analogue

4K – 1,664 Kb

Industrial Ethernet &
TCP/IP TIWAY,
MODUS,
PROFIBUS-DP & FMS



Simatic S7

264,144 Digital
16,384 Analogue

4 Kb – 64 Mb

Industrial Ethernet &
TCP/IP TIWAY,
MODUS,
PROFIBUS-DP & FMS



Nano

Maximum I/O

10-24 digital

Memory Size

?

Communication

UNI-TELWAY
Modbus slave



Micro

112 – 248 digital
16 – 65 Analogue

14K – 64K

UNI-TELWAY,
X-WAY , AS-i
Modbus/Jbus
FIPIO bus

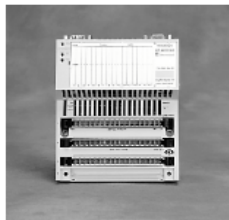


Premium

512 – 2048 Digital
32– 256 Analogue

32K – 256K

Ethernet,
UNI-TELWAY ,
X-WAY , AS-i
Modbus/Jbus
FIPIO, FIPWAY,
Modbus-plus



Momentum

16,384 discrete I/O

64K – 256K

?



Quantum

64,000 discrete I/O

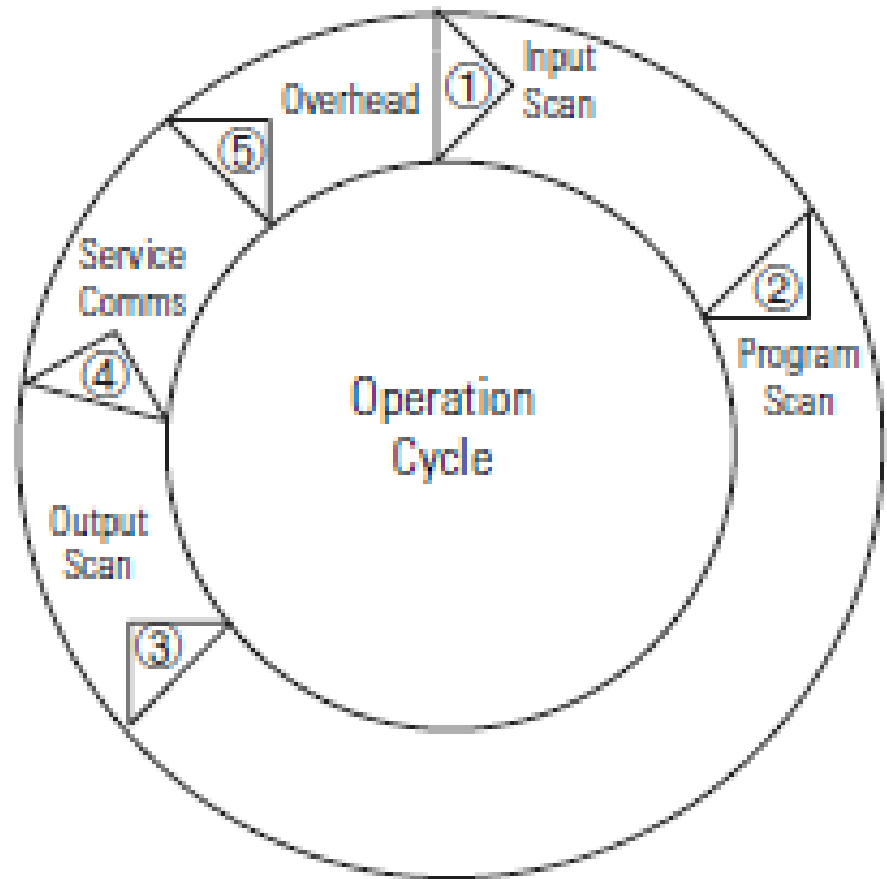
256K – 4M

?

Ciclo de Execução nos CLPs

Ciclo de Execução (scan) em Operação Normal – Modo RUN nos PLCs

1. Input scan
2. Program scan
3. Output scan
4. Service communications
5. Housekeeping and overhead



Input scan

The time required for the controller to scan and read all input data; typically accomplished within a few milliseconds.

Program scan

The time required for the processor to execute the instruction in the program. The program scan time varies depending on the instruction used and each instruction's status during the scan time.

Output scan

The time required for the controller to scan and write all output data; typically accomplished within a few milliseconds.

Service communications

The part of the operating cycle in which communication takes place with other devices, such as an HHT or a Personal Computer (PC)

Housekeeping and overhead

Time spent on memory

Scan da CPU do PAC

Additional Considerations

As you estimate the execution interrupts for a task, consider:

Consideration;	Description:
motion planner	<p>The motion planner interrupts all other tasks, regardless of their priority.</p> <ul style="list-style-type: none"> • The number of axes and coarse update period for the motion group effect how long and how often the motion planner executes. • If the motion planner is executing when a task is triggered, the task waits until the motion planner is done. • If the coarse update period occurs while a task is executing, the task pauses to let the motion planner execute.
I/O task	<p>CompactLogix, FlexLogix, and DriveLogix controllers use a dedicated periodic task to process I/O data. This I/O task:</p> <ul style="list-style-type: none"> • Does <i>not</i> show up in the Tasks folder of the controller. • Does <i>not</i> count toward the task limits for the controller. • Operates at priority 7. • Executes at the fastest RPI you have scheduled for the system. • Executes for as long as it takes to scan the configured I/O modules.

As you assign priorities to your tasks, consider the I/O task:

If you want a task to:	Then assign one of these priorities:
interrupt or delay I/O processing	1 to 6
share controller time with I/O processing	7
let I/O processing interrupt or delay the task	8 to 15

Scan da CPU do PAC

system overhead

System overhead is the time that the controller spends on unscheduled communication.

- Unscheduled communication is any communication that you do not configure through the I/O configuration folder of the project, such as Message (MSG) instructions and communication with HMIs or workstations.
- System overhead interrupts only the continuous task.
- The system overhead time slice specifies the percentage of time (excluding the time for periodic or event tasks) that the controller devotes to unscheduled communication.
- The controller performs unscheduled communication for up to 1 ms at a time and then resumes the continuous task.

continuous task

You do not assign a priority to the continuous task. It always runs at the lowest priority. All other tasks interrupt the continuous task.

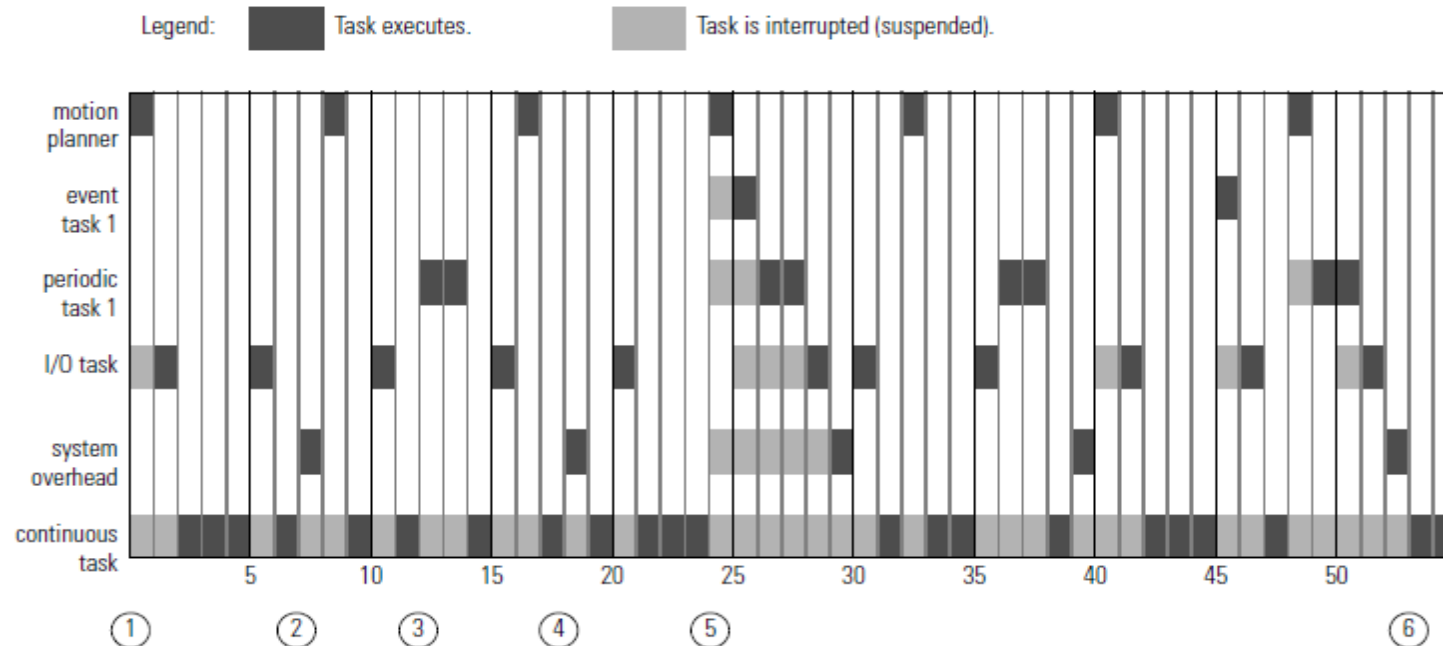
Scan da CPU do PAC

EXAMPLE

This example depicts the execution of a project with three user tasks.

Task:	Priority:	Period:	Execution time:	Duration:
motion planner	n/a	8 ms (course update rate)	1 ms	1 ms
event task 1	1	n/a	1 ms	1 to 2 ms
periodic task 1	2	12 ms	2 ms	2 to 4 ms
I/O task—n/a to ControlLogix and SoftLogix controllers. See page 3-5.	7	5 ms (fastest RPI)	1 ms	1 to 5 ms
system overhead	n/a	time slice = 20%	1 ms	1 to 6 ms
continuous task	n/a	n/a	20 ms	48 ms

Arquitetura



Description:

①	Initially, the controller executes the motion planner and the I/O task (if one exists).
②	After executing the continuous task for 4 ms, the controller triggers the system overhead.
③	The period for periodic task 1 expires (12 ms), so the task interrupts the continuous task.
④	After executing the continuous task again for 4 ms, the controller triggers the system overhead.
⑤	The triggers occurs for event task 1. Event task 1 waits until the motion planner is done. Lower priority tasks experience longer delays.
⑥	The continuous task automatically restarts.

Linguagens de Programação

São várias as linguagens de programação utilizadas nos controladores programáveis. O International Electrotechnical Committee – IEC é o responsável pela padronização dessas linguagens de programação sendo a norma IEC 1131-3 Programming Languages a recomendada para o assunto em questão.

Linguagens de Programação

Classificação das linguagens de programação:

- Structured Text - ST
- Ladder Diagram - LD ou Diagrama de Relés
- Function Block Diagram - FB
- Sequential Flow Chart – SFC
- Instruction List - IL

Ladder

The screenshot displays the RSLogix 5000 - Linguagens [Emulator]* software interface. The main workspace is divided into several panes:

- Left Pane:** A tree view showing the project structure, including Controller Languages, Tasks, MainProgram, and various data types.
- Top Pane:** A menu bar (File, Edit, View, Search, Logic, Communications, Tools, Window, Help) and a toolbar with icons for file operations, logic editing, and simulation.
- Bottom Pane:** A status bar showing the current rung (Rung 5 of 7) and application version (APP VER).

The central workspace is divided into four main views:

- MainProgram - Ladder:1:** A ladder logic diagram showing a sequence of steps (0, 1) with various logic elements like timers (TON), relays (K1, K2, K3), and outputs (Rel1).
- MainProgram - SFC:2:** A Sequential Function Chart (SFC) diagram showing a sequence of steps (Step_000, Step_001) connected by transitions.
- MainProgram - FBD:1:** A Function Block Diagram (FBD) showing a timer block (TONR_01) connected to a relay (Rel1).
- MainProgram - ST:** A Structured Text (ST) editor showing a ladder logic equivalent in text format:

```
If K1 then lampada := 1;
end_if;
If K3 then operacao := 1;
end_if;
```

The ST editor also displays the word "Texto" in a large font.

Select	When a section of your code represents
Relay Ladder Logic (RLL)	<ul style="list-style-type: none">•Continuous or paralel execution of multiple operations (not sequenced)•Boolean or bit based operations•Complex logical operations•Message and comunication processing•Machine interlocking•Operations that service or maintenance personnal may have to interpret in order to troubleshoot the machine of process•Servo motion control
Function Block Diagram (FBD)	<ul style="list-style-type: none">•Continuous process and drive control•Loop control•Calculations in circuit flow
Sequential Function Chart (SFC)	<ul style="list-style-type: none">•High-level management of multiple operations•Repetitive sequences of operations•Batch process•Motion control sequencing•State machine operations
Structured Text (ST)	Complex mathematical operations structured text Specialized array or table loop processing ASCII string handing or protocol processing

Das Especificações aos Controladores

O objetivo principal da Engenharia da Automação é projetar os programas em um computador (controlador industrial) que, interagindo com um dado processo, modificam o seu desempenho e procuram obter:

- Evolução “automática”, isto é, com grande independência de operadores;
- Interface ergonômica com os operadores;
- Alguma correção automática perante defeitos ou falhas.

Das Especificações aos Controladores

Na literatura há uma grande confusão de nomes para designar as diferentes partes dos programas na automação; por isso, segue abaixo algumas nomenclaturas para esclarecimento:

- **Controle dinâmico:** age sobre os processos dinâmicos (movidos pelo tempo) existentes na planta industrial, buscando sua regularidade e sua rapidez de resposta a comandos;
- **Controle de Eventos Discretos:** comanda as atividades gerais da planta industrial (vista como sistema a eventos), levando-a ao comportamento desejado, independentemente de operadores;

Das Especificações aos Controladores

- **Controlador de falhas:** (para eventos dinâmicos e discretos): entra em ação quando são detectados defeitos em máquinas ou peças, ou eventos externos anormais. Também é chamado de controlador de contingências ou recuperador de falhas (error recovery controller);
- **Sistema Supervisório:** permite aos operadores acompanhar e interagir/operar com o sistema de produção, quando necessário.

Das Especificações aos Controladores

- **Controlador Multidisciplinar:** é um controlador industrial que tem a capacidade de controlar mais de uma disciplina no processo produtivo, como:
 - Controle de Intertravamento;
 - Controle de Batelada – Batch;
 - Controle e Gerenciamento de Inversores de Frequência – Drives;
 - Controle e Gerenciamento de Servo-acionamento – Motion Control - Robos;
 - Controle de Processo – Malha fechada (PID, Lead, Lag);
 - Controle de Segurança de Máquinas – Safety Control;
 - Controle de acionamentos do tipo CNC, SDCD ou DCS, etc.

Das Especificações aos Controladores



Das Especificações aos Controladores

Controladores Lógicos versus Controladores de Automação

Estrutura fixa, modular e fixa+modular

Escalabilidade: Pequeno, Médio e Grande porte.

- Memória da CPU;
- Quantidade de I/Os digitais e analógicos;
- Quantidade de dispositivos em rede de comunicação que o controlador está gerenciando;

O que é necessário saber para especificar um Controlador Industrial?

Número de I/O's locais

Tipos de sinais dos I/Os locais e remotos (Digitais: 24Vdc? 127Vac? 220Vac? Analógicos: 0-10Vdc, 4-20mA, IR e/ou IT)

Número de I/O's remotos e sua distribuição

Quais as conexões de rede de comunicação necessárias

Quais as disciplinas de controle necessárias

Tamanho de memória de CPU necessária

O que é necessário saber para especificar um Controlador Industrial?

Tipos e quantidade de tarefas de execução necessária

Tipo de máquina ou processo que será automatizado

Quais dispositivos serão conectados na rede de comunicação

Arquitetura de controle simples ou redundante

Tipos de linguagem de programação necessária

Topologia da rede de comunicação

Nível de segurança do sistema (security)

Nomenclaturas

CLP: Controladores Lógicos Programáveis

PLC: Programmable Logic Controller

CAP: Controladores de Automação Programáveis

PAC: Programmable Automation Controller

SDCD: Sistema Digital de Controle Distribuído

DCS: Distributed Control System

CNC: Computer Numerical Control - Controle Numérico por Computador

Próxima Aula

SENSORES E ATUADORES INDUSTRIAIS

Obrigado 😊

ATÉ A PRÓXIMA AULA