Automação Industrial

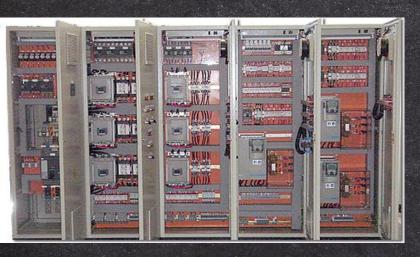
Princípio de Funcionamento de um CLP

Prof. Dr. Alexandre S. Brandão



Controlador Lógico Programável

Antigamente, o trabalho de automação era realizado por meio de relés. O funcionamento se baseava em listas de instruções e uma modificação resultava em um processo demorado cansativo e custoso.









Em um CLP, uma alteração na estratégia de controle é realizada através de um **programa fonte**, ou seja, alteração de **linhas de código**.



Configurações de um CLP

Compacto











Modular



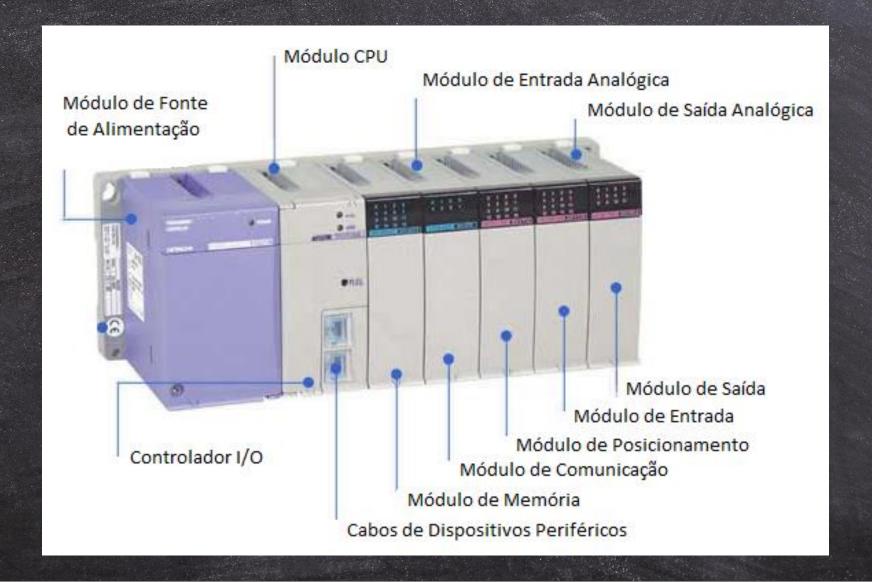






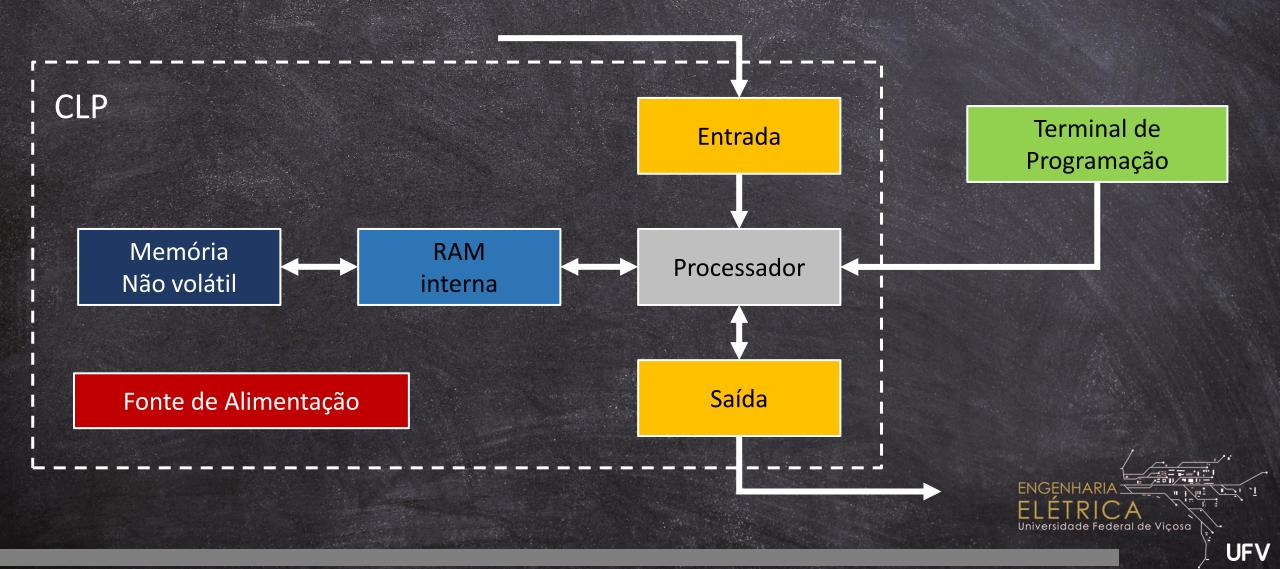


Configurações de um CLP





Constituição de um CLP



Arquitetura de um Computador

Sistema em Lote (Batch)

é caracterizado pelo sequenciamento no processamento dos pedidos de um programa

Sistema de tempo compartilhado (Time Sharing)

é um sistema onde o usuário poderá trabalhar com vários softwares simultaneamente em uma fração de tempo disponível do computador

Sistemas em tempo real (Real Time)

possuem características similares ao do tempo compartilhados, a grande diferença está na garantia do tempo de execução de cada tarefa de um programa em execução



Estrutura de Programa em CLP

O programa do usuário em um CLP é descrito segundo a linguagem de programação regulamentada pela IEC 1131

Permite escrever desde funções binárias simples até funções complexas

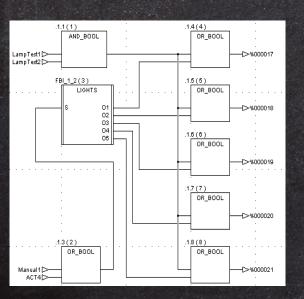
Linguagens de programação

- Textuais: Texto estruturado e Lista de instruções
- Gráficas: Diagrama de contatos (LADDER) e Blocos de funções
- Método Sequential Function Chart (Grafcet)

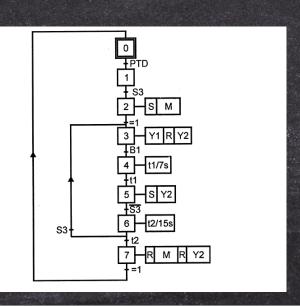


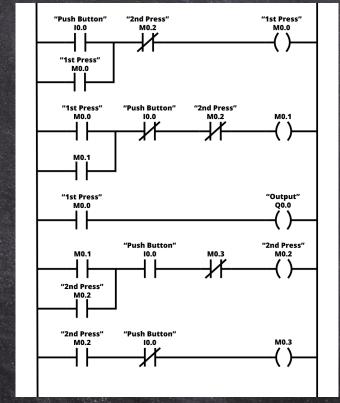
Estrutura de Programa em CLP

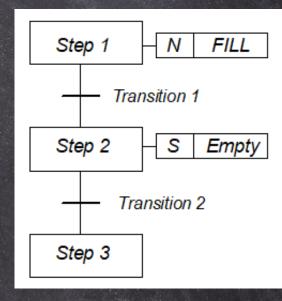
```
TIMER : TON:
TIMER(IN := NOT pulse,
      PT := t#1s); (* Blink timer *)
pulse := TIMER.Q;
 IF pulse = 1 THEN
  count := count + 1;
 (* Animate lights according to counter *)
 CASE count OF
  1. out1 .= TRUE
  2: out2 := TRUE
  3: out3
           := TRHE
   4: out4
   5: out5
   7: out7
   8: out8
 ELSE (* All lights are on, switch them off again and start
   out2 := FALSE:
   out3 := FALSE
   OUT4 := FALSE
   out5 := FALSE
   outó := FALSE
   out7 := FALSE
   count := 0;
```



```
RUN TIMER : TON; (* Blink timer *)
      (* Default for the marker *)
          run_light1
run light
  STN RUN_TIMER.IN
  CAL RUN TIMER(PT := t#15)
          RUN TIMER ET
          animatetime
          run_pulse
                       (* No pulse yet, nothing to do *)
                           (* Rotate all bits one position "up" *)
          run_light8
                                   run_light
          run light7
                                   run light8
          run_light6
                                   run_light7
          run_light5
                                   run_light6
          run_light4
                                   run light5
          run light3
                                   run_light4
          run_light2
                                   run_light3
          run_light1
                                   run_light2
          run light
                                   run_light1
          Instruction List End
```









Funcionamento Interno do CLP

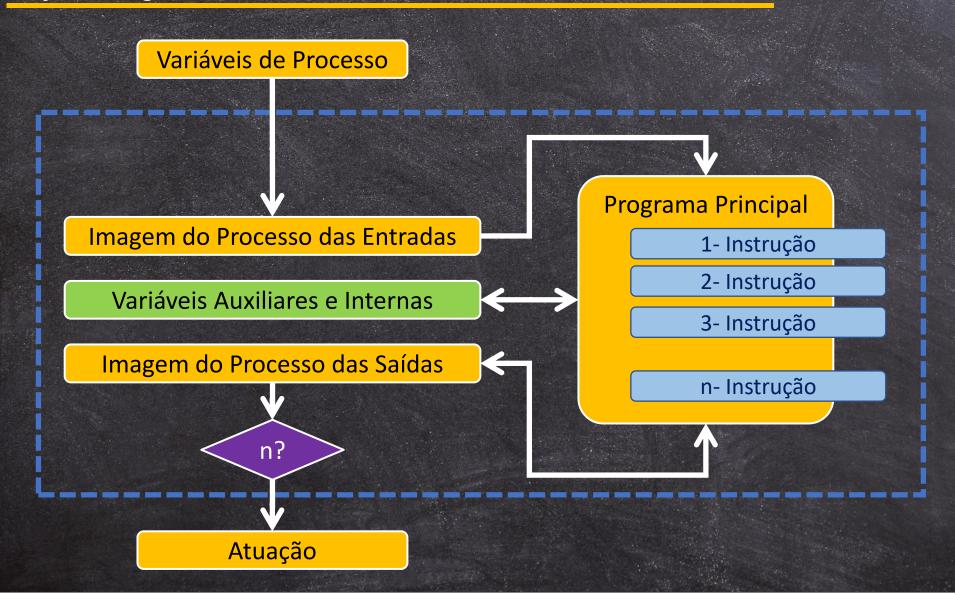
Cada instrução é introduzida em uma célula de memória da CPU do CLP e executada sequencialmente

Após o processamento da última instrução que se encontra na memória, o controle retorna à primeira instrução e o processo segue ciclicamente

Cada ciclo é executando em um tempo predeterminado, denominado ciclo de varredura (SCAN)



Operação Interna do CLP





Operação Interna do CLP

Antes de ler a instrução, faz-se a leitura das entradas do CLP

- Verificação do estado atual da planta
- Leitura armazenada em células de memória denominada Imagem do Processo das Entradas (PII)

Em função das entradas e das instruções na CPU, o controle do CLP habilita as saídas

- As informações são armazenadas em células de memória denominada Imagem do Processo de Saída (PIO)
- As saídas podem ser alteradas antes do fim do ciclo de SCAN

Da PIO, os dados são transferidos às saídas do CLP, que, por fim, atuam no processo

