



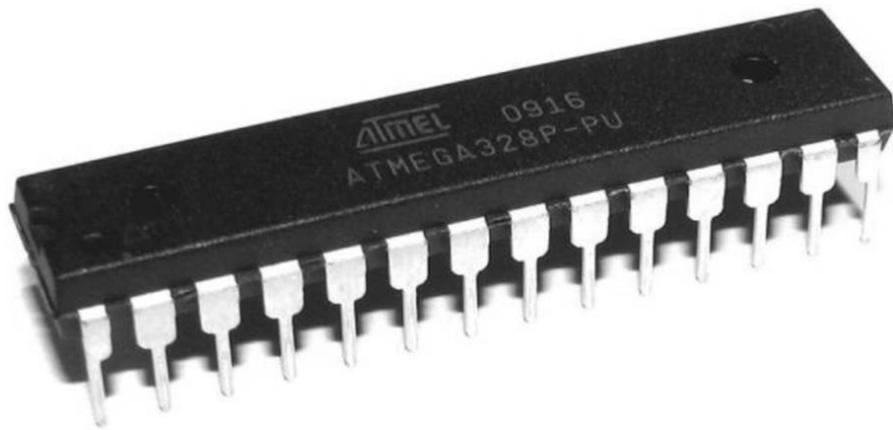
**UNIVERSIDADE DO ESTADO DE MINAS GERAIS**

**Docente: Rildo Afonso de Almeida**

**Dispositivos Eletrônicos**



## Microcontrolador ATmega328p





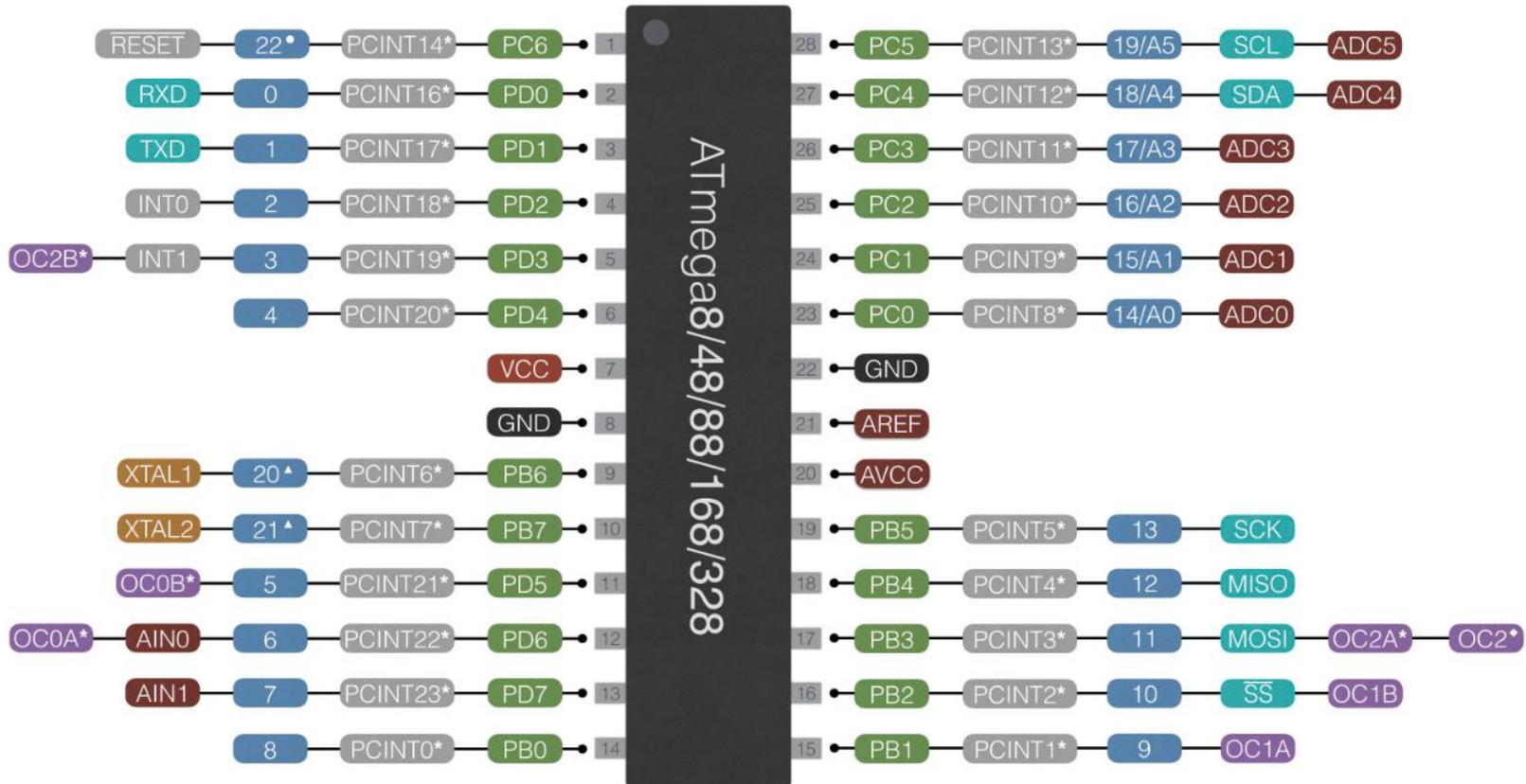
## Microcontrolador ATmega328p

MCU	ATmega328p
Family	AVR/ATmega
Vendor	Microchip (previously Atmel)
RAM	2KiB
Flash	32KiB
EEPROM	1KiB
Frequency	1MHz/8MHz (up to 20MHz with external clock)
Timers	3 (2x 8bit, 1x 16bit)
ADCs	6 analog input pins
UARTs	1
SPIs	1
I2Cs	1 (called TWI)
Vcc	2.7V - 5.5V (when clocked at 8MHz)
Datasheet	<a href="#">Official datasheet</a>



# Microcontrolador ATmega328p

- POWER
- GROUND
- PORT PIN
- ARDUINO PIN
- ANALOG
- SERIAL INTERFACE
- TIMER / PWM PIN
- INTERRUPT



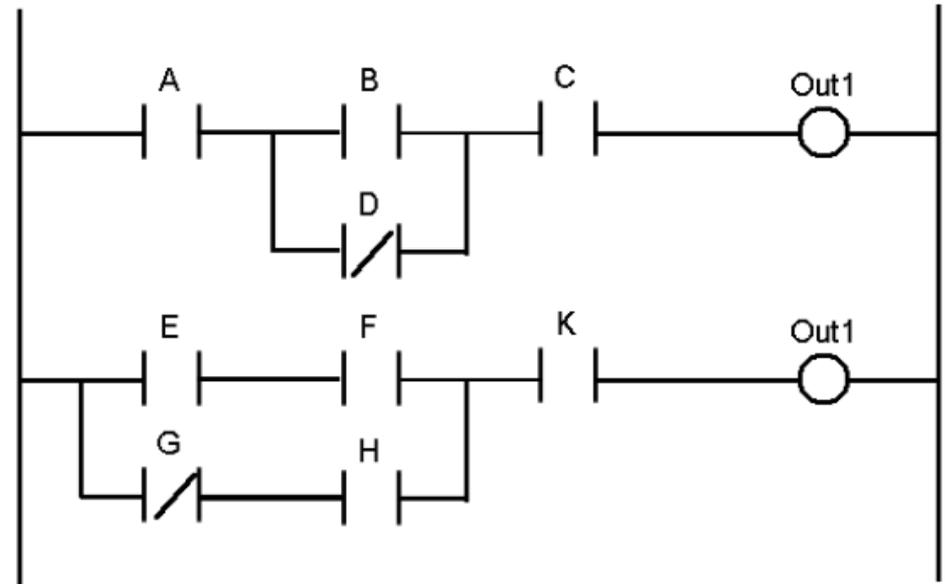


# **Programando Microcontroladores Linguagem Ladder**

## Lógica Ladder para microcontroladores

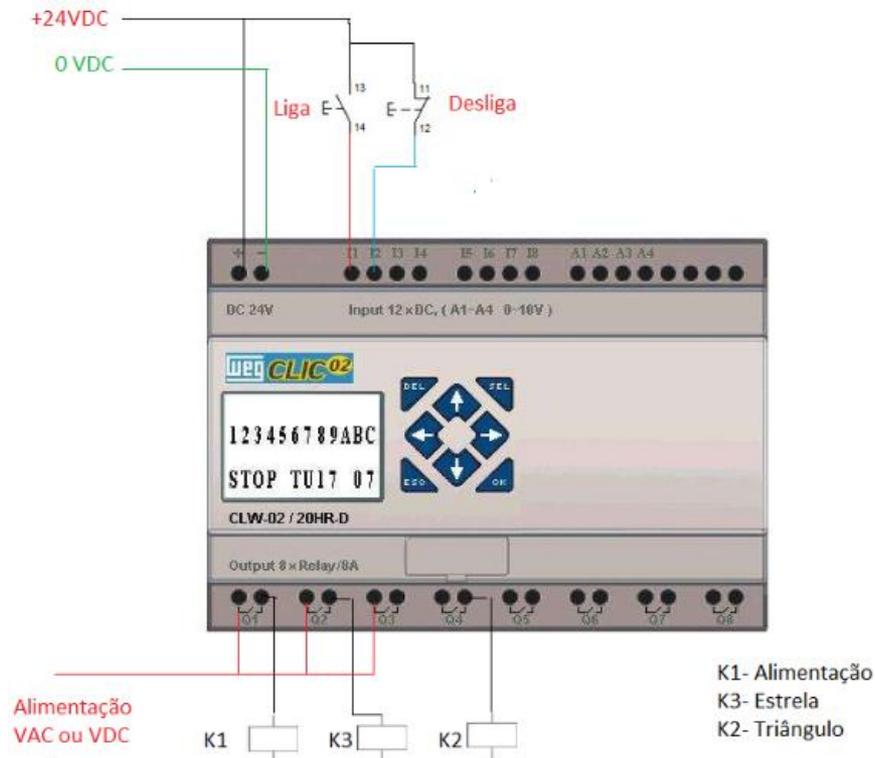
### Linguagem LADDER

Linguagem gráfica utilizada para programação de CLP's (Controlador Lógico Programável).  
O elemento básico é o relê.



# Lógica Ladder para Microcontroladores

## Linguagem LADDER





# Lógica Ladder para microcontroladores

## Entradas e saídas digitais

**Input Relays** -(Contatos) Conectados ao mundo externo.

Existem fisicamente e recebem sinais de **chaves**, **sensores**,

**etc...**



## Lógica Ladder para microcontroladores

### Entradas e saídas digitais

A instrução (XIC) é um contato normalmente aberto. Também chamada de: Examine se “ligado”.



- Quando a entrada física está “on” a instrução é verdadeira.
- Um sinal de entrada precisa estar presente para que o símbolo seja ativado.
- Pode ser usado para: Entradas Internas Entradas Externas Saídas Externas.



# Lógica Ladder para microcontroladores

## Entradas e saídas digitais

A instrução (XIO) é um contato normalmente fechado. Também chamada de: Examine se “desligado”.



- Um sinal de entrada não deve estar presente para que o símbolo seja ativado.
- Quando a entrada física está “off” a instrução é verdadeira.
- Um sinal de entrada não deve estar presente para que o símbolo seja ativado.
- Pode ser usado para: Entradas Internas Entradas Externas Saídas Externas (em alguns casos)



## Lógica Ladder para microcontroladores

### Entradas e saídas digitais

**Internal Relays** - (Contatos) Não existem fisicamente. Não recebem sinais externos; são simulados e permitem aos sistemas eliminarem relês externos.





# Lógica Ladder para microcontroladores

## Entradas e saídas digitais

**Output Relays** - (bobinas) São conectados ao mundo externo.



Existem fisicamente e enviam sinais on/off para solenóides, luzes, motores, etc... (Através de transistores, relês ou triacs)



## Lógica Ladder para microcontroladores

### Entradas e saídas digitais

Quando existe um caminho de “instruções verdadeiras” que precedem esta instrução no degrau da escada, ela também será verdadeira.

Quando a instrução é verdadeira ela está fisicamente “on”.

- Pode ser considerada como uma saída normalmente aberta.
- Pode ser usada para bobinas internas e saídas externas.

Bobina

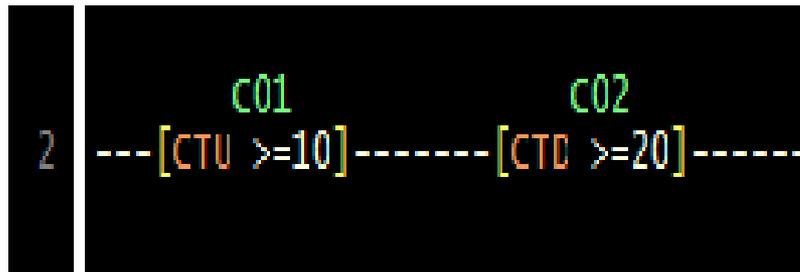
<b>Tipo</b>	<b>Fonte</b>	<input type="button" value="OK"/>
<input checked="" type="radio"/> ( ) Normal	<input type="radio"/> Rele Interno	<input type="button" value="Cancelar"/>
<input type="radio"/> (/) Negado	<input checked="" type="radio"/> Pino no Micro	
<input type="radio"/> (S) Ativar	<b>Nome:</b> <input type="text" value="new"/>	
<input type="radio"/> (R) Desativar		



## Lógica Ladder para microcontroladores

### Contadores e Temporizadores

Counters - Não existem fisicamente. São contadores simulados e podem ser programados para contar pulsos “up” ou “down”.





## Lógica Ladder para microcontroladores

### Contadores e Temporizadores

Timers - Não existem fisicamente. Incrementos variam de 1 ms a 1s.

```
2      T01          T02          T03
  -[TON 100.0 ms]---[TOF 100.0 ms]---[RTO 100.0 ms]--
```



## Lógica Ladder para microcontroladores

### Detector de Bordas de subida e de descida





## Lógica Ladder para microcontroladores

### Execução do Programa

**Check Input Status** - O sistema verifica cada entrada se está “on” ou “off”.

Armazena os dados na memória para serem usados no próximo passo.

O sistema executa o programa armazenado, baseado nas entradas lidas.

- Armazena o resultado para ser usado no próximo passo.

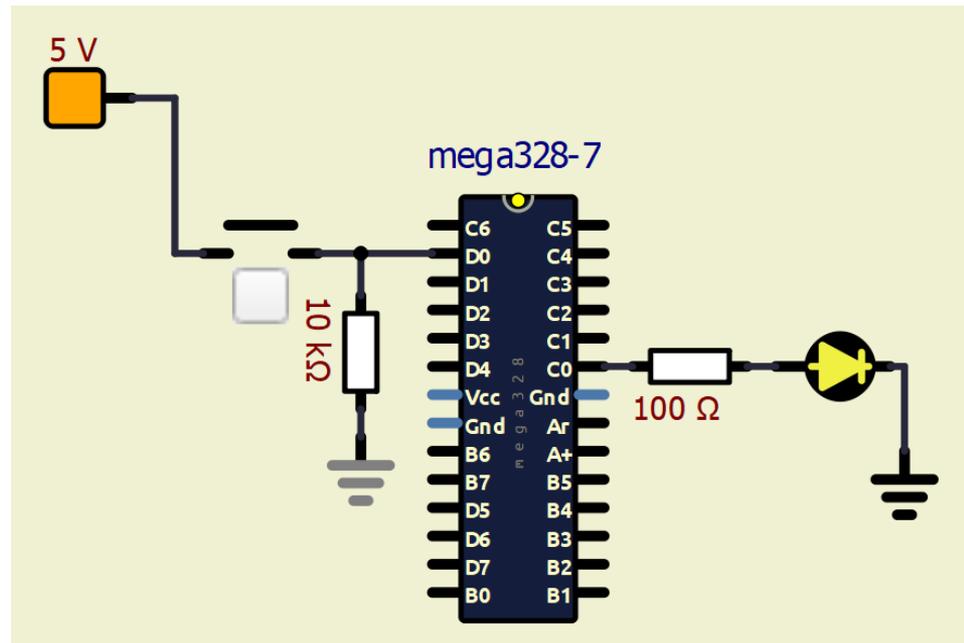
**Atualização das saídas.**

- Finalmente o sistema atualiza suas saídas baseado nos passos anteriores.

## Lógica Ladder para microcontroladores

### Exemplo 1

- 1) Elaborar um diagrama Ladder para um dispositivo liga/desliga (partida direta) em um único botão.

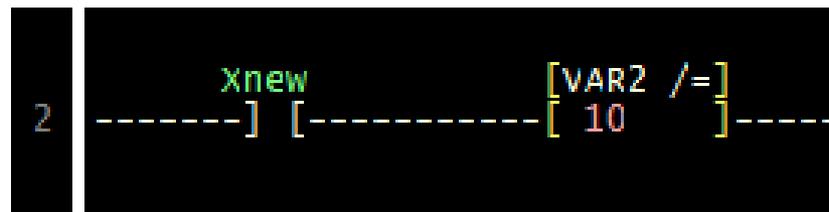


## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE COMPARAÇÕES

Instrução **IGUAL** - Compara se a variável **VAR1** é igual a um valor. Se for, torna a linha verdadeira.

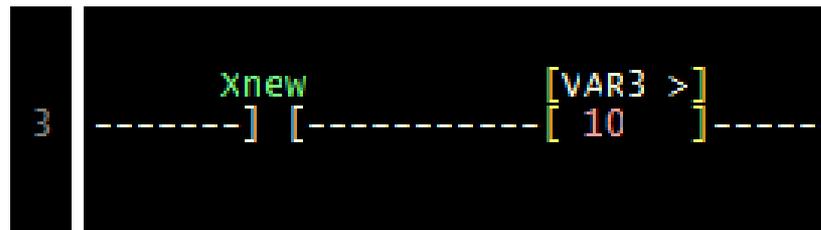


Instrução **DIFERENTE** - Compara se a variável **VAR2** é diferente a um valor. Se for, torna a linha verdadeira.

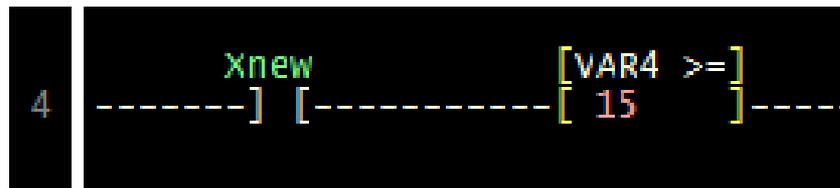


## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE COMPARAÇÕES

Instrução **MAIOR QUE** - Compara se a variável **VAR3** é maior que a um valor. Se for, torna a linha verdadeira.

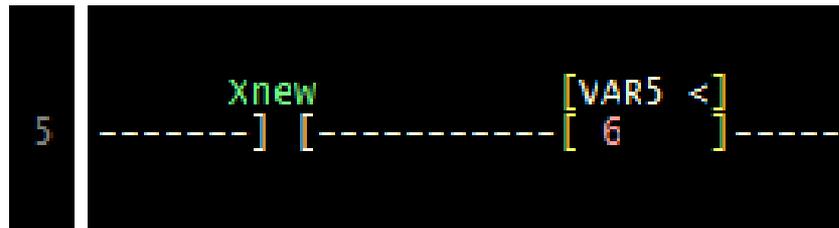


Instrução **MAIOR ou IGUAL** - Compara se a variável **VAR2** é maior ou igual a um valor. Se for, torna a linha verdadeira.



## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE COMPARAÇÕES

Instrução **MENOR QUE** - Compara se a variável **VAR5** é menor que a um valor. Se for, torna a linha verdadeira.



Instrução **MENOR QUE** ou **IGUAL** - Compara se a variável **VAR6** é menor que ou igual a um valor. Se for, torna a linha verdadeira.



## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE MOVIMENTAÇÃO

Instrução **MOVER** - Move o valor de uma variável origem para uma variável destino sempre que a linha for verdadeira.

```
-----{ pwm1 := }-----  
-----{ Ccont MOV }-----
```

## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE MATEMÁTICA

Instrução **SOMAR** - Adiciona um valor à uma variável e o resultado é apresentado em uma variável destino sempre que a linha for verdadeira.

```
-----{ADD dest :=}  
-----{ pw1 + 20 }-----
```



## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE MATEMÁTICA

Instrução **SUBTRAIR** - Subtrai um valor de uma variável e o resultado é apresentado em uma variável destino sempre que a linha for verdadeira.

```
-----{SUB dest :=}  
-----{ var1 - 10 }-----
```

## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE MATEMÁTICA

Instrução **MULTIPLICAR** - Multiplica uma variável por um valor e o resultado é apresentado em uma variável destino sempre que a linha for verdadeira.

```
{MUL dest :=}  
---{ var * 5 }---
```

## PROGRAMAÇÃO LADDER - INSTRUÇÕES DE MATEMÁTICA

Instrução **DIVIDIR** - Divide uma variável por um valor e o resultado é apresentado em uma variável destino sempre que a linha for verdadeira.

```
-----{DIV dest :=}  
-----{ var / 2 }-----
```



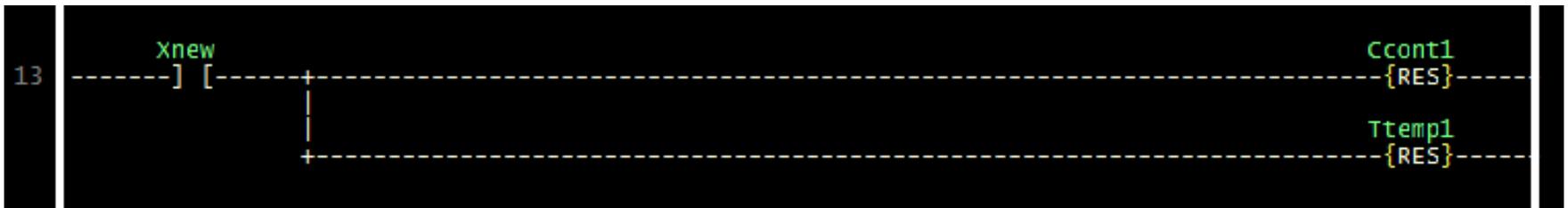
## PROGRAMAÇÃO LADDER - INSTRUÇÕES ESPECIAIS

Instrução **RELÉ DE CONTROLE MASTER** - Habilita / desabilita as linhas subsequentes a ele.

```
1  -----] [-----{MASTER RLY}---
```

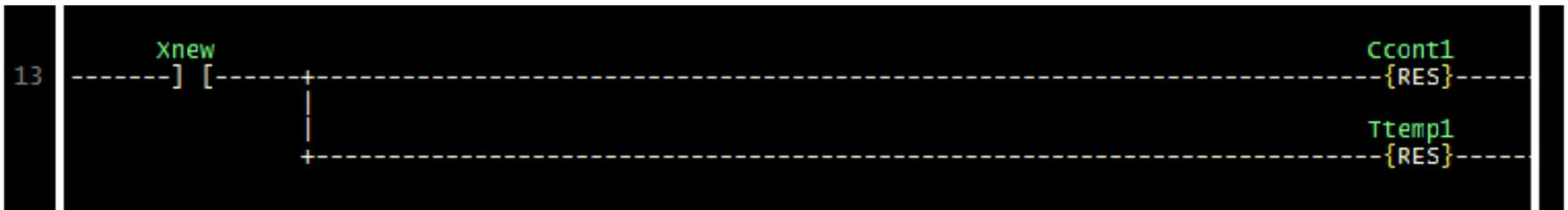
## PROGRAMAÇÃO LADDER - INSTRUÇÕES ESPECIAIS

Instrução **RES** (CONTADOR e TEMPORIZADOR) - Reseta contadores e temporizadores.



## PROGRAMAÇÃO LADDER - INSTRUÇÕES ESPECIAIS

Instrução **RES** (CONTADOR e TEMPORIZADOR) - Reseta contadores e temporizadores.

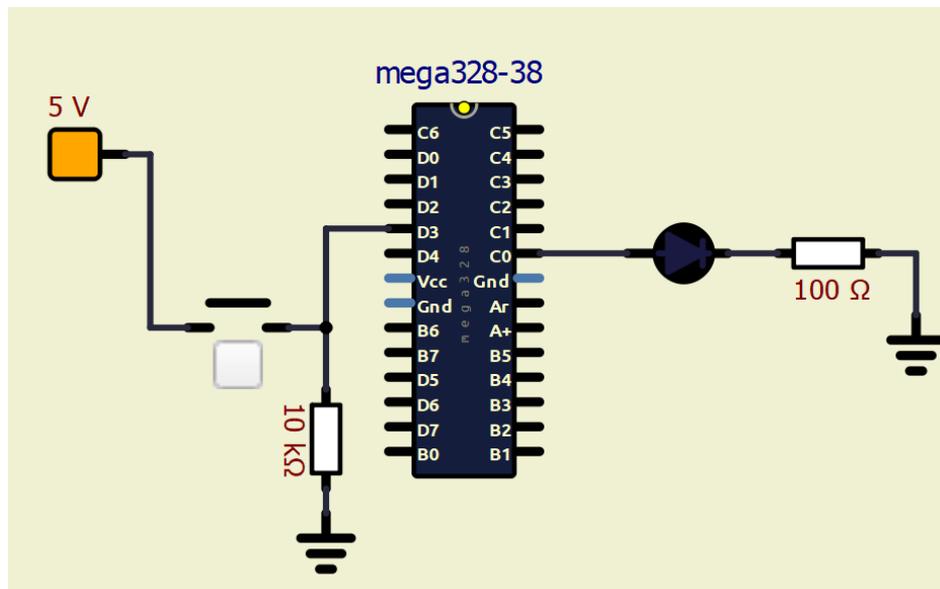


## PROGRAMAÇÃO LADDER - EXERCÍCIOS

### Exemplo 2

Baseado no esquema abaixo, desenvolver um código com as instruções vista até o momento na disciplina.

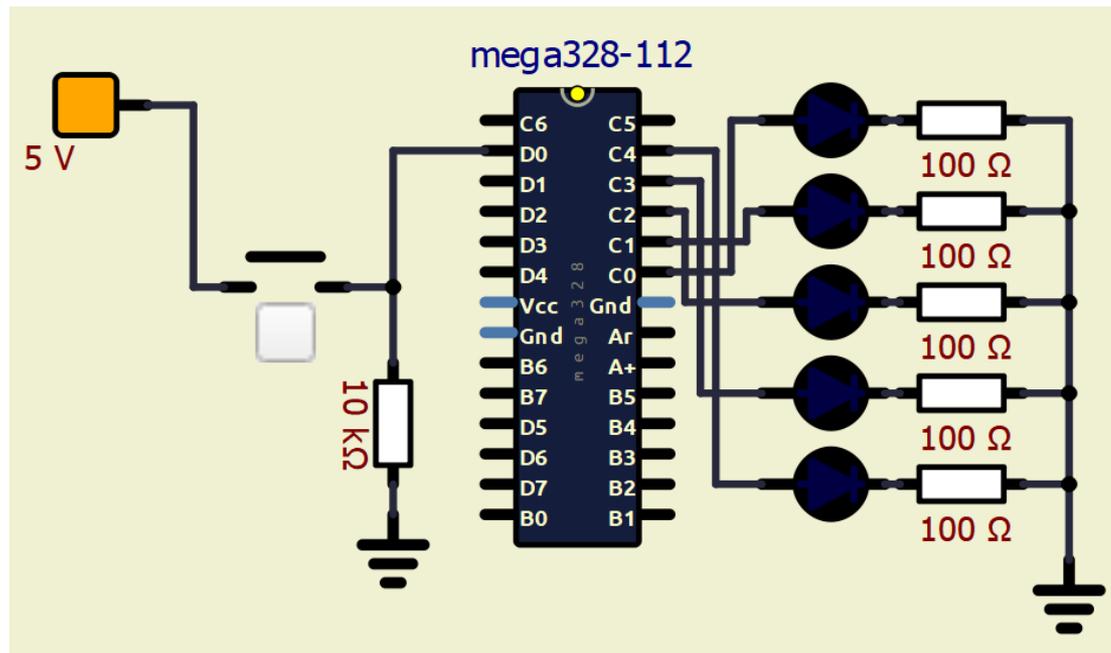
Detalhe: Ao acionar S3, o LED D2 deverá piscar em intervalos de 3 segundos.



## PROGRAMAÇÃO LADDER - EXERCÍCIOS

### Exemplo 3

Ao acionar a chave S1 deverá ocorrer o acendimento dos LEDs na sequência do LED1 para o LED5, com intervalos entre leds de 2 segundos. O ciclo deverá se repetir indefinidamente enquanto a chave S1 estiver acionada.





## Bibliografia Básica

OLIVEIRA, A.S & ANDRADE, F. S. Sistemas embarcados – Hardware e Firmware na prática. Érica, 2006.

YAGHMORET et al. Construindo Sistemas Linux Embarcados: Conceitos, técnicas, truques e dicas. 2<sup>a</sup> ed. Altabooks, 2009.



# Bibliografia Complementar

Revistas especializadas.

Artigos científicos.