

---

# Universidade Federal de Minas Gerais

## Escola de Engenharia

Departamento de Engenharia Eletrônica

Laboratório de Informática Industrial

### AULA 9 - PROGRAMAÇÃO SOB A NORMA IEC 61131-3 (I): INTRODUÇÃO

*Objetivos:* Familiarização com os recursos de programação sob a norma IEC 61131-3 no controlador *CompactLogix* da Rockwell Automation.

---

#### Atividades Prévias

1. Leia atentamente, em casa, o texto desta prática.
2. Desenvolva previamente, ainda em casa, os programas nas linguagens ST e FBD descritos no item “Atividades de Projeto e Programação” desta prática.
3. Elabore um pré-relatório manuscrito, entre 1 e 2 páginas, descrevendo resumidamente os passos a serem executados nesta aula prática e incluindo os programas desenvolvidos em casa.
4. Ao chegar ao laboratório, caso tenha um colega de bancada, discutam entre suas respectivas soluções e decidam, em conjunto, qual deles será o adotado na prática.

---

#### Introdução

O CLP *CompactLogix* é aderente à norma IEC 61131-3, permitindo o desenvolvimento de aplicações de controle de processos industriais nas linguagens *Ladder*, *Structured Text* (ST), *Function Block Diagram* (FBD) e *Structured Function Chart* (SFC). Para tal, o *CompactLogix* é dotado de um sistema operacional multitarefa e preemptivo, onde cada tarefa (*task*) pode ser composta de múltiplos programas, cada um deles com múltiplas rotinas e funções. As tarefas podem ser contínuas, periódicas (com período de execução configurável na faixa de 0.1 ms até 2000 s) ou por eventos.

- Tarefa Contínua (*Continuous Task*):
  - A tarefa contínua executa em *background* (ou seja, com menor prioridade). Qualquer tempo da CPU não alocado para outras operações (p. ex. comunicação de dados ou tarefas periódicas) é usado para executar os programas dentro da tarefa contínua.
  - A tarefa contínua executa o tempo todo. Quando a tarefa contínua completa um ciclo de *scan*, ela recomeça imediatamente.
  - Um projeto não requer necessariamente uma tarefa contínua. Se usada, pode haver apenas uma tarefa contínua.
- Tarefa Periódica (*Periodic Task*):
  - Uma tarefa periódica executa uma função em intervalos de tempo regulares. Sempre que o período de tempo para a tarefa periódica se renova, esta interrompe qualquer outra tarefa de prioridade mais baixa.
  - Após a execução, a tarefa periódica devolve o controle para o mesmo ponto onde a tarefa anterior foi interrompida.
- Tarefa por Evento (*Event Task*):
  - Executa apenas quando um evento específico (disparo) ocorrer.
  - Sempre que ocorrer o disparo para a tarefa de evento, esta interrompe todas as tarefas com prioridades mais baixas, executa uma vez e retorna o controle para o ponto em que a tarefa anterior parou.

Ao criar-se um projeto no *RSLogix 5000*, este automaticamente define uma tarefa (*MainTask*), um programa (*MainProgram*) e uma rotina (*MainRoutine*) para o projeto. A tarefa *MainTask* é definida inicialmente como contínua, mas pode ser alterada para periódica ou por eventos. Outras tarefas,

programas e rotinas podem ser posteriormente adicionados ao projeto, através do organizador do controlador (Fig. 1).

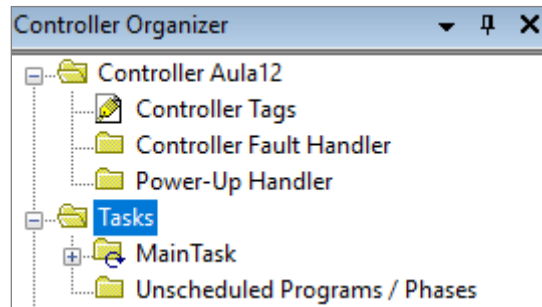


Figura 1: Pasta de “tarefas” no organizador de programas.

## Criação e edição de programas nas linguagens ST e FBD

A rotina *Main Routine* criada pelo *R.SLogix 5000* quando da criação de um projeto é definida automaticamente do tipo *Ladder*. Para substituímos a mesma por uma de outro tipo (ST, FBD ou SFC) você deve apagar a mesma e inserir outra rotina, na linguagem desejada:

1. Clique com o botão direito do mouse na rotina *Main Routine* e selecione *Delete* (Fig. 2):

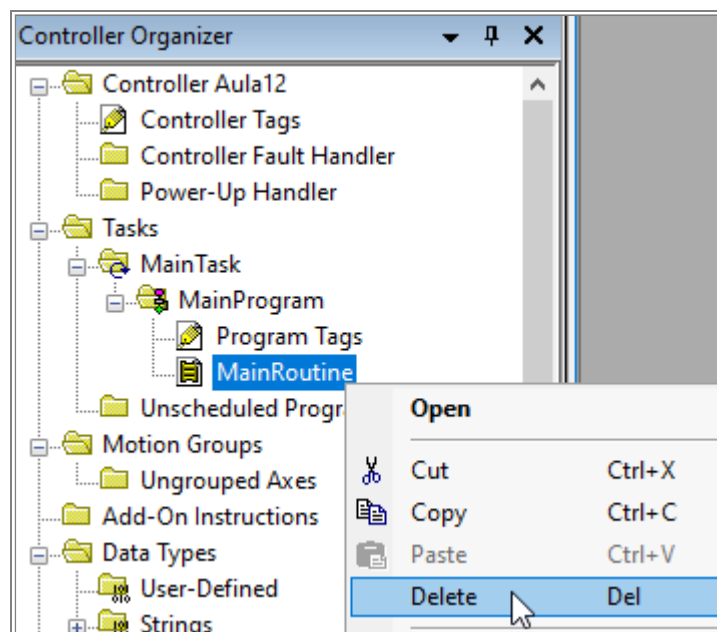


Figura 2: Cancelamento de rotinas no organizador de programas.

2. Em seguida, clique com o botão direito do mouse em *Main Program* e selecione *New Routine*: (Fig. 3):

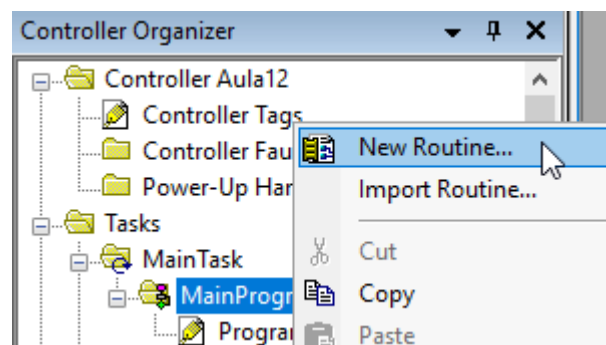


Figura 3: Criação de nova rotina, no organizador de programas.

- Na janela que se abrir (Fig. 4), defina o nome da rotina, a descrição da mesma (útil para fins de documentação) e a linguagem desejada:

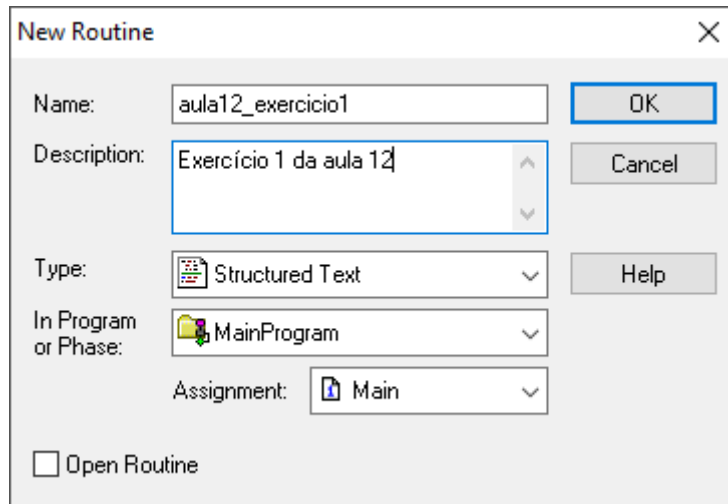


Figura 4: Janela de criação de rotina

- Após criar a nova rotina, é necessário defini-la como principal: clique no objeto *Main Program* com o botão direito do mouse, selecione *Properties* e, na janela que se abrir (Fig. 5), defina a rotina criada como *Main*, na aba *Configuration*:

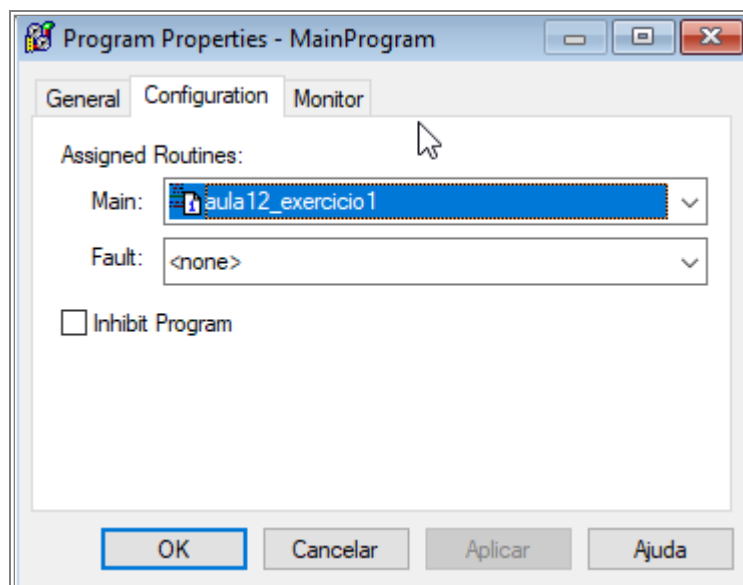
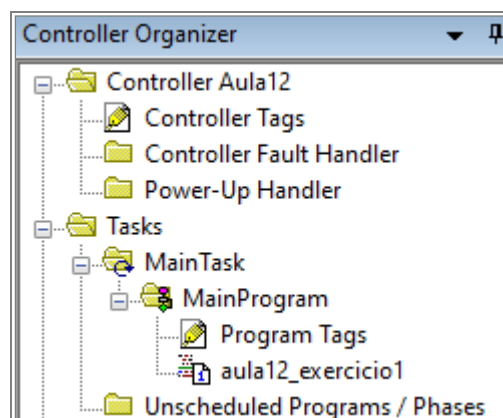


Figura 5: Janela de propriedades de um programa.

- Ao final do processo, sua área de organização do programa deverá estar configurada como exemplificado na Fig. 6:



**Figura 6:** Configuração do organizador de programas após a criação de um novo programa.

Para editar o programa, dê um duplo clique na rotina criada para que o editor ST fique ativo. Ao desenvolver seu programa, não se esqueça de definir as variáveis desejadas (internas e externas).

O mesmo procedimento acima serve para declarar e desenvolver um programa em FBD. As únicas diferenças, obviamente, são que a rotina deve ser declarada nesta linguagem, e o editor trazido pelo *RSLogix 5000* passa a ser um editor gráfico.

**NOTA:** No *RSLogix 5000*, a definição de variáveis (*tags*) é feita sempre da mesma forma em qualquer das linguagens da norma IEC61131-3, ou seja, digitando-se o nome da variável em algum elemento gráfico (ou textual), clicando-se com o botão direito do mouse neste nome e selecionando-se a opção *New tag*. Desta forma, a declaração de variáveis no corpo dos programas através das construções **VAR\_INPUT** . . . **END\_VAR**, etc., não é suportada pelo *RSLogix 5000*. Defina as variáveis da forma descrita e, no corpo do programa, inclua apenas a lógica em si.

## Atividades de projeto e programação

### Parte 1 – Projeto em texto estruturado

Desenvolva uma rotina na linguagem *Texto Estruturado* (ST) que acenda a lâmpada vermelha do MICA quando o botão “Liga” do contator C1 for acionado (em modo local), e depois de 5 segundos acenda a lâmpada verde do MICA. Ambas as lâmpadas devem apagar-se quando o botão “Desliga” for acionado.

Para esta prática use um temporizador do tipo TONR (*Timer-On Delay with Reset*), pois o temporizador TON utilizado nas aulas anteriores é disponível apenas para linguagem *Ladder*. A única diferença entre ambos é que o TONR tem uma entrada de *Reset*.

Para usar qualquer temporizador em ST, você deve primeiro definir um *tag* do tipo *base* e declarar o mesmo com o tipo de dados do temporizador desejado. No caso do TONR, o tipo de dados deve ser o FBD\_TIMER. Uma vez definido este *tag*, você deve inicializar os campos respectivos de habilitação do temporizador (`nome_do_tag.TimerEnable`) e de valor de *preset* (`nome_do_tag.PRE`). O campo `nome_do_tag.DN` vai para o nível lógico 1 quando o valor acumulado de tempo superar o tempo de *preset*. Veja o exemplo a seguir (atenção – trata-se de um exemplo; não o copie diretamente para sua lógica!):

```
meu_timer.TimerEnable := 1;    // Habilita Temporizador
meu_timer.PRE := 2000;        // Preset = 2 segundos
...
TONR (meu_timer);            // Executa a temporização

IF (meu_timer.DN) THEN      // Se TRUE, tempo alcançado
...
END_IF;
```

**DICA:** Lembre-se que que quaisquer blocos de função na norma IEC-61131, entre eles a instrução TONR, devem ser executados continuamente para que operem corretamente. Um erro comum é considerar que basta uma única chamada ao bloco TONR para deixá-lo ativo em toda a execução do programa; portanto, verifique cuidadosamente sua lógica para certificar-se que este bloco esteja sendo sempre executado, em cada varredura do CLP.

Para este projeto, empregue os endereços de E/S apresentados na tabela 1.

**Tabela 1: Endereços de entrada e saída – parte 1**

Tipo	Modo	Endereço	Elemento no MICA	ID no MICA
BOOL	Input	Local:1:I.Data.0	Botão Liga/Desliga do contator C1	RB01/00
BOOL	Output	Local:3:O.Data.0	Lâmpada Vermelha	RB03/00

Execute a montagem no MICA e teste seu programa antes de prosseguir para a parte 2.

### Instruções de montagem no MICA


1. Com o MICA energizado, certifique-se que o botão de emergência esteja pressionado.
2. Conecte os bornes de alimentação do contator C1 (situado à direita dos sensores capacitivo e fotoelétrico) à régua RB24 (fonte 24V) e à régua RB00 (terra).
3. Conecte o borne lateral L1 à régua RB24 (fonte 24V) e o borne lateral T1 ao ponto de entrada da variável “Liga/Desliga Local” na tabela 1, de forma a sinalizar ao CLP a operação de liga/desliga efetuada pelo operador. Os bornes L1 e T1 estão situados na lateral direita do MICA.
4. Alimente a régua RB03 e conecte a saída indicada na tabela 1 ao sinalizador luminoso (lâmpada).
5. Verifique atentamente se todos os elementos da montagem estão devidamente aterrados.
6. Confira com atenção a sua montagem antes de rearmar o botão de emergência.

### **Parte 2 – Projeto em Diagrama de Blocos de Função (FBD)**

Adicione ao seu *MainProgram* uma rotina na linguagem *Function Block Diagram* (FBD) que leia os sinais dos sensores capacitivo e fotoelétrico do MICA, execute a função lógica *ou-exclusivo* (XOR) entre os mesmos empregando apenas portas lógicas dos tipos AND e OR, e acenda a lâmpada amarela em função do resultado. Empregue os endereços de E/S indicados na tabela 2. Não se esqueça de declarar a rotina em FBD como principal, em seu *MainProgram*.

**Tabela 2: Endereços de entrada e saída – parte 2**

Tipo	Modo	Endereço	Elemento no MICA	ID no MICA
BOOL	Input	Local:1:I.Data.1	Sensor Capacitivo	RB01/01
BOOL	Input	Local:1:I.Data.2	Sensor Fotoelétrico	RB01/02
BOOL	Output	Local:3:O.Data.2	Lâmpada Verde	RB03/02
BOOL	Output	Local:3:O.Data.3	Lâmpada Amarela	RB03/03

**NOTA:** Use os blocos BAND (*Boolean AND*), BOR (*Boolean OR*) e BNOT (*Boolean NOT*), que estão presentes na aba do *RSLogix* referente às instruções *Move/Logical*. Estes blocos têm várias entradas, sendo necessária a eliminação das entradas desnecessárias. Para tal, clique no ícone  presente no bloco, e desmarque as mesmas.

### Instruções de montagem no MICA

1. Com o MICA energizado, certifique-se que o botão de emergência esteja pressionado.
2. Alimente os sensores e conecte suas saídas às respectivas entradas na tabela 2.
3. Conecte as saídas indicadas na tabela 2 aos sinalizadores luminosos (lâmpadas).
4. Verifique atentamente se todos os elementos da montagem estão devidamente aterrados.
5. Confira com atenção a sua montagem antes de rearmar o botão de emergência.

### **Referências**

Rockwell Automation, Guia de procedimentos, ControlLogix usando RSLogix5000.