

REFERÊNCIA RÁPIDA PARA USO DO SOLVER

Linguagem Descritiva

OBJETIVOS

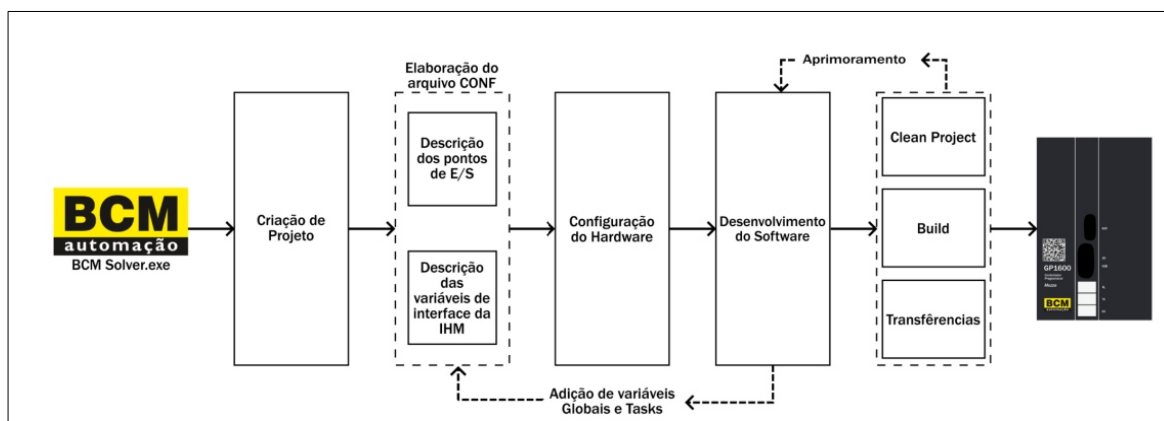
Fornecer referência rápida dos procedimentos básicos para uso do Solver no desenvolvimento de programas de automação. Ao lado de cada seção os capítulos correspondentes do manual são listados para consulta. Este documento tem enfoque no uso da linguagem descritiva, para uso de outras linguagens consulte os outros documentos da família.

VISÃO GERAL

O solver é o ambiente de programação para os CLPs da BCM Automação, com suporte para todas linguagens da normal IEC61131-3. Conta com editor moderno, com *syntax-highlighting*, *autocomplete*, sugestões de correção, link de declaração de funções e variáveis, histórico de alterações, etc. Oferece ainda um poderoso simulador dos programas, funcionalidades de depuração e monitoração dos CLPs, editor integrado de IHM gráfica, entre diversas outras.

O fluxo de desenvolvimento de uma aplicação segue as etapas gerais listadas abaixo, que serão detalhadas neste documento.

- I. Especificação da aplicação, levantamento dos pontos de E/S, descrição do processo em um diagrama de estados;
- II. Criação de um novo projeto no Solver;
- III. Declaração das variáveis de E/S, configuração do CLP;
- IV. Implementação da lógica do processo;
- V. Transferência para o controlador, depuração do programa;



PROCEDIMENTOS

1. Criando o projeto (3)

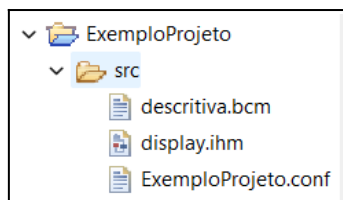
Objetivo: Criar um novo projeto com a estrutura geral já preenchida;

1.1. Crie um novo projeto em **File→New→Project**;

1.2. Clique no *Template* “**Linguagem Descritiva Default**” e em **Next**;

1.3. De um nome para o projeto e clique em **Finish**;

1.4. Será criada uma pasta no caminho padrão do Solver (*C:\BCM-Solver\win32.win32.x86_64\eclipse\workspace*), que será exibida na aba lateral esquerda do Solver com os demais projetos. Dentro da pasta criada se encontra uma pasta **src** com os arquivos-fonte do projeto, sendo o **.conf** relacionado a declaração das variáveis globais e das tasks, o **.bcm** das rotinas dos processos e o **.ihm** de configuração do display. A aba esquerda do Solver mostrará todos projetos criados. Para configurar quais projetos serão exibidos consulte o manual.



2. Importando Projetos

Objetivo: Importar um projeto recebido

2.1. Clique em **File→Import**

2.2. Selecione a opção **General→ Existing Project into Workspace**

2.3. Deixe selecionado “**Select root directory**” e clique no botão **Browse** ao lado;

2.4. Selecione a pasta do projeto que deseja importar;

2.5. Se a opção “**Copy projects into workspace**” for selecionada, o solver copiará automaticamente os arquivos para o caminho padrão (*C:\BCM-Solver\win32.win32.x86_64\eclipse\workspace*). Caso contrário, a edição será diretamente sobre os arquivos no caminho apontado;

2.6. Clique em **Finish** para concluir a importação;

3. Declarando pontos de E/S (4)

Objetivo: Declarar os pontos de E/S que serão utilizados no programa

- 3.1. Uma pasta com o nome do projeto que pode ser visualizada na aba esquerda. Clique sobre a seta no lado esquerdo da pasta e do lado esquerdo da pasta **src** para mostrar os arquivos;
- 3.2. Clique duas vezes sobre o arquivo **descritiva.bcm**. Ele será aberto no editor de arquivos no centro da ferramenta;
- 3.3. Os pontos de E/S devem ser listados entre as palavras **VAR_EXTERNAL** e **END_VAR**.
Os pontos digitais devem ser declarados como **BOOL** e os analógicos como **WORD**;

```
VAR_EXTERNAL  
LIGA_FORNO: BOOL;    //SD1  
BOTA0_EMER: BOOL;    //ED1  
SENS_TEMPE: WORD;    //EA1  
CMD_INVERSOR: WORD;  //SA1  
END_VAR
```

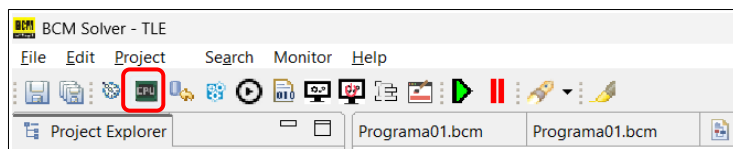
- 3.4. Sincronize as variáveis globais clicando em **Project**→**Sincroniza variáveis globais**;
- 3.5. Alternativamente à função **Project**→**Sincroniza variáveis globais**, as variáveis globais podem ser copiadas manualmente para o arquivo de extensão .conf, inseridas entre as palavras **VAR_GLOBAL** e **END_VAR**.

```
VAR_GLOBAL  
LIGA_FORNO: BOOL;    //SD1  
BOTA0_EMER: BOOL;    //ED1  
SENS_TEMPE: WORD;    //EA1  
CMD_INVERSOR: WORD;  //SA1  
END_VAR
```

4. Configurando o Controlador (6)

Objetivo: Configurar o modelo do controlador e informar quais pontos de E/S serão utilizados e com quais variáveis;

4.1. Clique no ícone com quadrado preto na barra de itens;



4.2. No lado esquerdo do Solver abrirá um retângulo de nome **Project Configurator**;

4.3. Para selecionar a CPU, clique com o botão direito do mouse no primeiro item e selecione a CPU do controlador desejado;

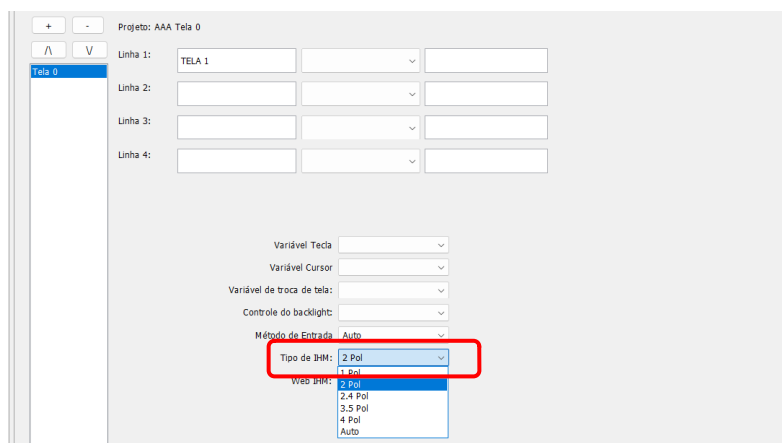
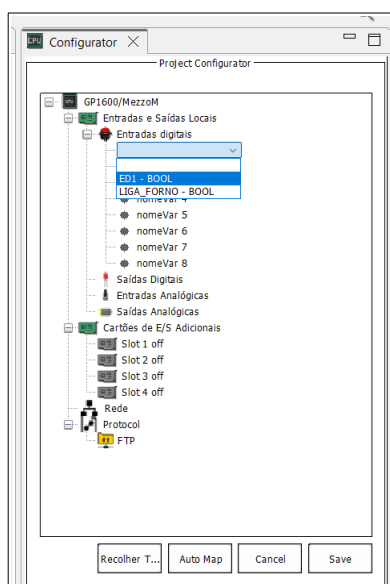
4.4. Caso o necessário, adicione os módulos pertinentes clicando com o botão direito sobre o item **Módulos de E/S**, ou sobre os Slots de E/S;

4.5. Para atribuir variáveis do programa para pontos de E/S, clique com o botão direito sobre o grupo de pontos (Entradas digitais, Saídas digitais, etc) e selecione a opção **Mapeamento a variáveis BOOL**;

4.6. Cada um dos campos abertos corresponderá ao ponto de E/S daquela numeração. Para atribuir uma variável, clique duas vezes sobre o ponto e selecione a variável desejada. Sempre que novas variáveis forem adicionadas ao projeto, deve-se sincronizar o projeto ou inseri-las manualmente na sua seção correspondente no arquivo **.conf**;

4.7. Clique em **Salvar**;

4.8. Caso o controlador possua display, ele também deve ser configurado. Abra o arquivo **display.ihm** e selecione o tipo do display no campo **Tipo de IHM**. Variáveis e mensagens também podem ser inseridas neste arquivo;



5. Programando as rotinas (4, 10, 11)

Objetivo: Implementar a lógica de controle do processo;

5.1. As lógicas de programação devem ser inseridas no arquivo **descriptiva.bcm**, abaixo do comentário `// Insira aqui as rotinas`

5.2. Pode-se colocar diversas malhas que serão executadas em paralelo. Toda Máquina deve iniciar com a palavra-chave **MAQUINA** ou **MALHA**, seguida de um identificador e ':'. Cada Estado deve iniciar com a palavra-chave **ESTADO** seguida do número do estado e ':'. Toda malha deve ter um ESTADO 0, e não pode ter dois estados com o mesmo número. Todas linhas de instrução e condições devem terminar com ponto e vírgula;

```
MAQUINA CNTR_FORNO:
ESTADO 0:
    SE BOTAO = 1 ENTAO 1;
ESTADO 1:
    LIGA FORNO;
    SE SENS_TEMPE > 200 ENTAO 2;
ESTADO 2:
    DESL FORNO;
    SE ATRASO = T#1h ENTAO 0;
```

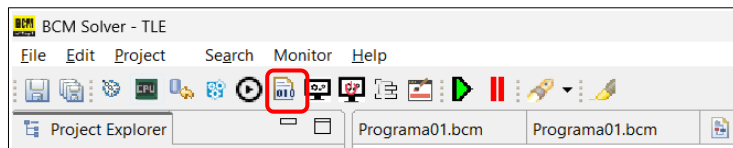
5.3. Caso utilize mais variáveis temporárias no programa elas devem ser declaradas. Abaixo são listados tipos comuns como referência. Caso sejam globais, clique em sincronizar as variáveis ou copie no arquivo **.conf**;

Função	Tipo
Valores digitais que só podem assumir 0 ou 1	BOOL
Valores inteiros para entradas analógicas	WORD
Valores inteiros pequenos (até 65535)	INT
Valores inteiros grandes	LINT
Valores com casas decimais	REAL

6. Compilando o Projeto (7)

Objetivo: Traduzir o projeto para o controlador, verificando se cometemos erros;

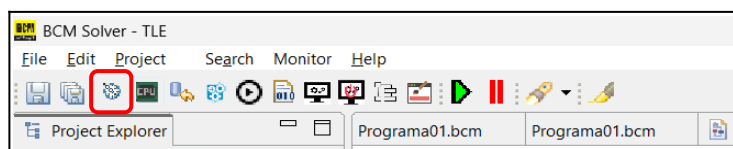
6.1. Clique no botão compilar na aba superior;



6.2. Um pop-up informará o resultado da compilação. Em caso de falha, o problema pode ser visto na aba **Problems**, na parte inferior do Solver. Clicando-se duas vezes sobre o problema ele abrirá o local do erro. Fique atento para linhas sem dois pontos, nomes errados de variáveis, erros na digitação das palavra-chave, variáveis não declaradas, etc.

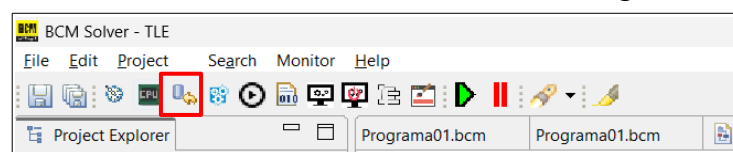
7. Transferindo o Projeto (7)

7.1. Configure a comunicação com o controlador na engrenagem na aba de comandos;

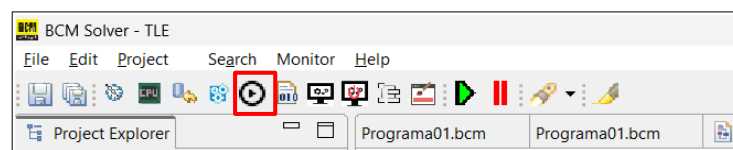


7.2. Selecione o método de transferência. No caso da serial, selecione a porta COM do controlador, e de Ethernet, insira seu endereço IP;

7.3. Clique no ícone de transferência na aba de comandos e aguarde;



7.4. No caso de transferência por Ethernet o CLP precisa ser reiniciado para carregar o novo programa. Isso pode ser feito manualmente ou através do comando abaixo;



- 8.1. Configure o canal de monitoração da mesma forma que foi feito para transferência de programa;
- 8.2. Certifique-se que o controlador está conectado, o programa está compilado e foi transferido para o controlador;
- 8.3. Clique em **Monitor→Monitoração** de variáveis para abrir uma janela de monitoração;
- 8.4. Selecione as variáveis de interesse nos campos superiores. Pode-se escrever o nome da variável ou clicar na seta para listar as variáveis existentes;
- 8.5. Clique no triângulo verde no canto inferior esquerdo para iniciar a monitoração. Os valores lidos do CLP serão exibidos na tela, empurrando os valores mais antigos para baixo;
- 8.6. Para escrever em uma variável, clique em um quadrado com valor, escreva o novo valor e pressione Enter. O quadrado deve ficar verde e a variável assumirá o novo valor no CLP;
- 8.7. Mais colunas podem ser adicionadas e a largura das colunas e o período de leitura configurado. Confira o manual para mais detalhes.
- 8.8. Feche a janela para interromper a monitoração. Não é possível transferir programas pela Serial enquanto o controlador é monitorado;

Escreve valores

Inicia monitoração

Seleciona variáveis

AAA - Monitora Variáveis (1)

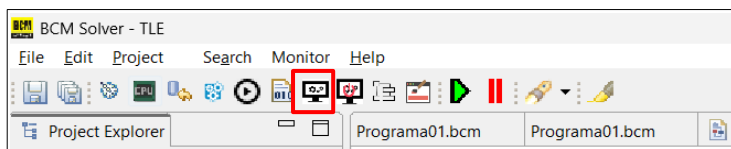
	ED1	LIGA_FORNO					
6,5	12	0					
6	12	0					
5,5	12	0					
5	0	0					
4,5	0	0					
4	0	0					
3,5	0	0					
3	0	0					
2,5	0	0					
2	12	0					
1,5	0	0					
1	0	0					
0,5	0	0					

+ - Amostragem: 0.5 Seg Tamanho Col. 100 Nome:

9. Simulando o Programa (8)

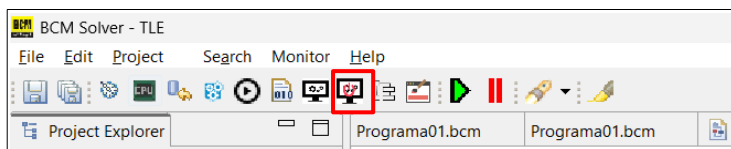
9.1.O programa desenvolvido pode ser testado sem um controlador conectado através da simulação. Para isso, configure o canal de monitoração na engrenagem na aba de comandos para **Simulador**;

9.2. Com o projeto já compilado, pressione o botão de início da simulação



9.3. Um pop-up deve surgir informando que o simulador está em execução. A partir deste momento todas operações de monitoração serão feitas sobre o controlador simulado. As variáveis podem ser escritas e lidas como descrito em 7;

9.4.Para interromper a simulação deve-se pressionar o botão de pausa da simulação. Para qualquer alteração do programa se refletir no simulador a simulação deve ser pausada, o programa compilado e a programação iniciada novamente. Mudanças feitas com o simulador em execução não se refletirão automaticamente.



10. Referência Rápida de Programação

10.1. Linguagem Descritiva

Instrução	Função
MALHA <NOME> :	Define início de uma malha.
ESTADO <NUMERO> :	Define o início de um estado Uma mesma malha não permite dois estados com o mesmo número
FACA <var> := <expr>;	Avalia expressão e move resultado para variável; Executada uma vez na entrada de um estado
FACA* <var> := <expr>;	Avalia expressão e move resultado para variável; Executada continuamente dentro de um estado.
SE exp ENTAO <estado>;	Avalia expressão e vai para estado caso verdadeira; estado pode ser uma variável ou um inteiro. Aceita combinação de expressões com operadores lógicos (E, OU);
SE exp ENTAO var:=expr;	Avalia expressão e caso verdadeira realiza a atribuição
SE ATRASO = T#<tempo> ENTAO estado;	Realiza a transição após o tempo determinado; tempo pode ser uma variável em nanossegundos. Formataadores de tempo aceitam ns,us,s,m,h (ex 2 segundos = T#2s)
VA PARA estado;	Vai para estado. Executada se nenhum SE ENTAO for atendido;
LIGA var;	Move TRUE para a variável
DESL var;	Move FALSE para a variável
INCREMENTA var;	Incrementa a variável em uma unidade
DECREMENTA var;	Decrementa a variável em uma unidade
MOSTRA (1,1, 'ESTADO 1', VAR1, 'C', 'T');	Interage com o display do controlador. Consulte manual para uso detalhado

10.2. Blocos de função BCM

Nome do Bloco	Função
RTC	Horário interno do controlador
PID_A	Bloco de PID do tipo A
PID_B	Bloco de PID do tipo B
PID_C	Bloco de PID do tipo C
FSM	Máquina de estados
TASK_INFO	Informações sobre as tasks
RESOURCE_INFO	Informações sobre os recursos
BUS_INFO	Informações sobre a atuação de E/S
RECIPE_CONTROL	Gerenciamento das receitas
IP_CONFIG	Configuração de IP
PDT_PASSWORD_CONFIG	Configuração da monitoração
FTP_CONFIG	Configuração do FTP
NETWORK_STATUS	Status da REDE
MODBUS_CONFIG	Configuração do ModBus
BCM210_SENSOR_CONFIG	Configuração do Sensor BCM210
BCM2_CONFIG	Configuração do BCM2
TIMER	Contagem de tempo precisa

10.3. Modos de execução de Tasks

Nome	Função
CONTINUOUS	Execução na velocidade máxima do controlador
INTERVAL	Execução periódica
SINGLE	Execução na borda de subida de variável especificada

```
TASK T1 (CONTINUOUS, PRIORITY:=1)

TASK T2 (INTERVAL:= T#10ms, PRIORITY:=1)

TASK T3 (SINGLE:=ED1, PRIORITY:=1)

PROGRAM P1 WITH T1 : <NOME DO PROGRAMA>;
```

11. Resolvendo Problemas no Solver

- 11.1. **Falhas em operações relativas à comunicação serial, seja na monitoração, transferência de programa, transferência de código fonte:** O Windows pode apresentar instabilidades quanto ao gerenciamento de portas COM. Desconecte e conecte o controlador. Se o problema persistir, feche e abra o Solver
- 11.2. **Controlador não mostra nenhuma informação no Display:** Verifique se o modelo do Display foi corretamente configurado no arquivo `.ihm.`;
- 11.3. **Projeto compila mas não transfere:** Verifique se a CPU está configurada corretamente;
- 11.4. **Janela do Solver aparenta estar sem resposta:** Verifique se nenhuma janela de pop-up do Solver está minimizada atrás da janela principal e está bloqueando a execução;
- 11.5. **Janela do configurator está toda escura ou desformatada:** Feche a janela no ícone x e abra novamente;
- 11.6. **A aba Problems ou Console foram fechadas:** Pressione Alt+Shift+Q e aguarde uma caixa de opções surgir no canto esquerdo. Selecione o item desejado;
- 11.7. **Compilação falha por variável não declarada:** Verifique se a variável em questão está declarada com o nome e tipo correto no arquivo `.conf` e no arquivo do programa.

EXEMPLOS DE USO DA LINGUAGEM DESCRITIVA

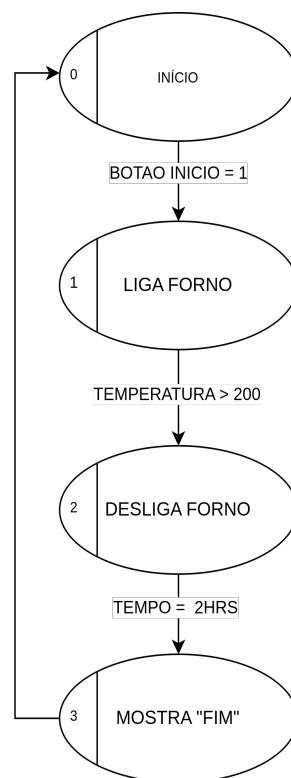
1. Forno de Secagem Simples

1.1 Descrição

Para um processo de secagem simples, um forno deve ser acionado até a temperatura interna ultrapassar 200 graus. Em sequência, deve ser resfriado naturalmente por 2 horas; O processo deve ser iniciado quando um botão for pressionado; A conversão do valor lido da entrada analógica para graus deve dar-se:

$T_{med}[\text{Graus}] = \text{ValorEA1} / 20000.0 * 500.0;$

1.2 Diagrama de Estados



1.3 Implementação

```
PROGRAM FORN01

VAR_EXTERNAL
    FORNO: BOOL;           //SD1
    BOTA0: BOOL;           //ED1
    TEMPERATURA: WORD;     //EA1
END_VAR

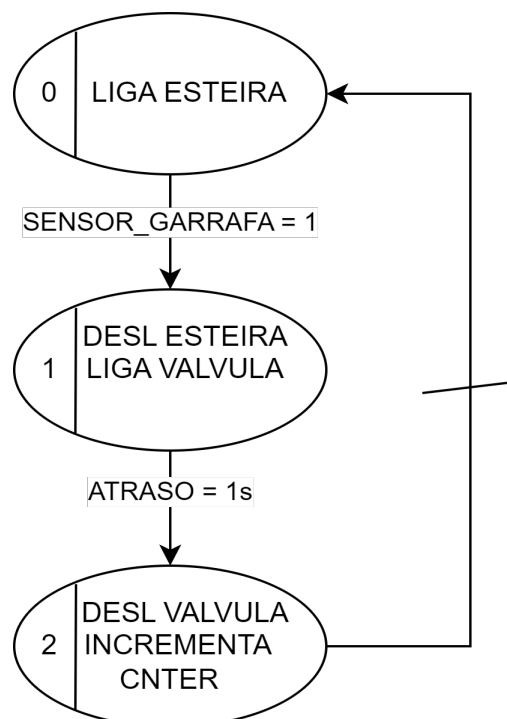
MAQUINA CNTR_FORNO:
ESTADO 0:
    SE BOTA0 = 1 ENTAO 1;
ESTADO 1:
    LIGA FORNO;
    SE TEMPERATURA/20000.0 * 500.0 > 200 ENTAO 2;
ESTADO 2:
    DESL FORNO;
    SE ATRASO = T#2h ENTAO 3;
ESTADO 3:
    MOSTRA(1,1, 'FIM', ' ', ' ');
    VA PARA 0;
END_PROGRAM
```

2. Esteira de Enchimento

2.1 Descrição

Desejamos automatizar uma linha de enchimento de garrafas. Devemos ligar a esteira até o sensor de posicionamento de garrafa ser acionado. Uma vez acionado, devemos desligar a esteira e abrir a válvula de despejo do líquido por 1 segundos. Em sequência, devemos ligar a esteira novamente e repetir o processo.

2.2 Diagrama de Estados



2.3 Implementação

```
PROGRAM ESTEIRA

VAR_EXTERNAL
    ESTEIRA      :BOOL; //SD1
    VALVULA      :BOOL; //SD2
    S_GARR       :BOOL; //ED1
    CNTER        :UDINT; //VAR INTERNA CONTAGEM
END_VAR

MAQUINA LINHA_GARRAFAS:

ESTADO 0:
    LIGA ESTEIRA;
    SE S_GARR = 1 ENTAO 1;

ESTADO 1:
    DESL ESTEIRA;
    LIGA VALVULA;
    SE ATRASO = T#1s ENTAO 2;

ESTADO 2:
    DESL VALVULA;
    INCREMENTA CNTER;
    VA PARA 0;
END_PROGRAM
```

3 Furadeira

3.1 Descrição

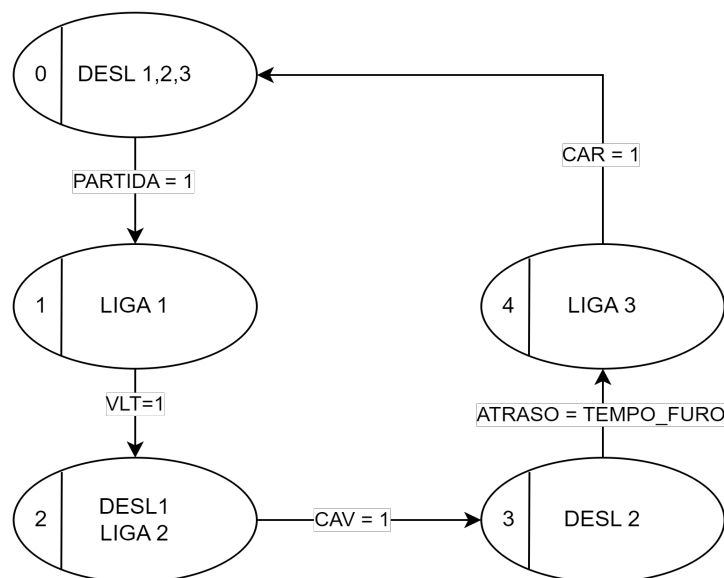
Ao ser pressionada a partida, o cabeçote deve avançar na velocidade rápida até acionar o sensor VLT;

Após acionar o sensor VLT, o cabeçote deve avançar na velocidade lenta até acionar o sensor CAV;

Após acionar o sensor CAV, o cabeçote deve permanecer na posição por um tempo configurável TEMPO_FURO;

Passado o TEMPO_FURO, o cabeçote deve recuar até atingir o sensor CAR e esperar por um novo comando;

3.2 Diagrama de Estados



3.3 Implementação

```
PROGRAM FURADEIRA_AUTOMATICA

VAR_EXTERNAL
    SD1 : BOOL ; //BOBINA AVANCO RAPIDO
    SD2 : BOOL ; //BOBINA AVANCO LENTO
    SD3 : BOOL ; //BOBINA AVANCO RECUO

    PARTIDA : BOOL; //ED1 BOTAO DE PARTIDA
    VLT : BOOL;      //ED2 SENSOR TROCA VEL
    CAV : BOOL;      //ED3 SENSOR FIM CURSO
    CAR : BOOL;      //ED4 SENSOR REPOUSO

    TEMPO_FURO : UDINT ; //TEMPO DE FURO EM MS
END_VAR

MALHA FURADEIRA1:

ESTADO 0:
    DESL SD1, SD2, SD3;
    SE PARTIDA = 1 ENTAO 1;
ESTADO 1:
    LIGA SD1;
    SE VLT = 1 ENTAO 2;
ESTADO 2:
    DESL SD1;
    LIGA SD2;
    SE CAV = 1 ENTAO 3;
ESTADO 3:
    DESL SD2;
    SE ATRASO = T#1ms * TEMPO_FURO ENTAO 4;
ESTADO 4:
    LIGA SD3;
    SE CAR = 1 ENTAO 0;

END_PROGRAM
```

AVISO LEGAL

As informações, diagramas, exemplos de software, programas, procedimentos e demais materiais apresentados neste documento são fornecidos apenas para fins informativos e educacionais.

Todo o conteúdo é disponibilizado livremente, porém sem qualquer garantia, expressa ou implícita, incluindo, mas não se limitando, às garantias de precisão, confiabilidade, adequação a uma finalidade específica ou ausência de erros.

A BCM Automação não assume qualquer responsabilidade por danos diretos, indiretos, incidentais ou consequentes decorrentes do uso das informações, softwares, diagramas ou procedimentos descritos neste documento.

O uso de qualquer material aqui apresentado é de inteira responsabilidade do usuário, que deve realizar suas próprias análises, validações e testes antes de aplicar tais informações em sistemas reais ou aplicações críticas.

A BCM Automação não assume obrigações legais decorrentes do uso das informações ou softwares disponibilizados neste documento.

Este material pode ser utilizado e distribuído livremente, desde que mantida a referência à BCM Automação como fonte original.

A BCM Automação reserva-se o direito de modificar, atualizar ou remover quaisquer informações, softwares ou materiais descritos neste documento a qualquer momento, sem aviso prévio.