

# Geradores de Pulso Encoders

Na indústria, de modo geral, é muito comum a utilização de dispositivos capazes de indicar fisicamente a posição exata de um equipamento ou um certo ângulo de movimento. Para essas aplicações existe um equipamento capaz de converter movimentos lineares, rotacionais e angulares em uma série de pulsos elétricos que, se interpretados da maneira correta, podem fornecer importantes informações sobre o funcionamento de máquinas e equipamentos. Esse dispositivo é chamado de “*encoder*”, ou simplesmente um “*gerador de pulsos*”.

## 11.1. O Gerador de Pulso

O encoder é um dispositivo eletromecânico que converte movimentos rotacionais em uma série de pulsos digitais elétricos. Com o auxílio de sistemas mecânicos, vários tipos de movimentos podem ser transformados em movimentos rotacionais. Dessa forma, o encoder pode realizar medições de movimentos lineares, angulares ou simplesmente movimentos rotacionais. Os pulsos enviados podem ser usados para determinação de variáveis como velocidade [m/s]; aceleração; rotação [rpm]; inclinação [graus °], direção [CW/CCW] e posição. Existem ainda encoders lineares que operam pelo princípio de deslizamento, porém não são muito utilizados devido à sua configuração e fragilidade.

Em outras palavras, encoder é um transdutor capaz de converter movimentos mecânicos em sinais elétricos padronizados.

Dentre as aplicações em que a utilização de encoders é essencial, podemos destacar:

- Tornos, fresadoras e ferramentas NC e CNC; medição de posição linear ou rotação.
- Robôs, para medição de inclinação, posição linear, graus de liberdade.
- Motores elétricos, medição de velocidade.
- Sistemas de posicionamento, antenas, radares, telescópios, medição de elevação (inclinação), azimute (posição linear).
- Sistemas rotativos de posicionamento, medição de velocidade, posição linear, ângulo de posicionamento.

Quando estamos nos referindo a transdutores de movimento, devemos levar em consideração três aspectos básicos que determinam o tipo de transdutor e suas características, são eles: resolução, precisão e repetibilidade.

**Resolução:** é a definição da quantidade de pulsos elétricos que o dispositivo envia à saída a cada revolução de 360°, ou seja, a cada volta



completa o encoder envia uma série de pulsos definidos em sua configuração, ex.: quando utilizamos um encoder com 1000 pulsos, significa que, a cada revolução completa do eixo teremos uma transição na saída de 1000 pulsos ou analogamente 1000 passos.

Dessa forma, é possível calcular a resolução em unidades de ângulo/passo, ou seja, para um encoder com 1000 pulsos temos uma resolução de:

$$\frac{360^\circ}{\text{Número de Passos}} = \text{ângulo} / \text{passo} \quad \therefore \frac{360^\circ}{1000} = 0,36^\circ / \text{passo}$$

Então, a cada pulso na saída, o eixo se movimentou numa fração de 0,36°.

Existem ainda alguns tipos de encoders que apresentam resoluções em forma digital, ou seja, a resolução é expressa na forma binária, um encoder com 10 bits de resolução, por exemplo, disponibiliza em suas saídas um valor entre 0 e 1023 que indica proporcionalmente a posição em relação ao giro do eixo. Podemos encontrar encoders com resolução variando entre 10 bits e 16 bits, mais adiante voltaremos a abordar esse tipo de encoder com mais detalhes.

A definição de resolução é característica básica de um encoder. Não se pode dimensionar ou identificar um encoder sem esse parâmetro.

**Precisão:** expressa a diferença entre o valor real da medição com o valor indicado pelo instrumento. Em outras palavras, é a quantidade de pulsos enviados em um determinado posicionamento pela quantidade de pulsos que o dispositivo deveria ter realmente enviado para o mesmo posicionamento.

Os encoders apresentam essa variável em percentagem por unidade de ângulo, quanto maior a precisão, melhor a indicação do valor real pelo valor indicado, quanto maior o erro, mais distante é a indicação do valor real pelo valor indicado.

**Repetibilidade:** a repetibilidade determina a característica de um encoder indicar um mesmo valor de saída para um mesmo ponto de deslocamento, ou seja, ao posicionarmos seguidas vezes o eixo do instrumento em um mesmo ponto de referência, o sinal indicado pelo instrumento deve ser o mesmo na maioria das vezes.

Em algumas aplicações essa variável é de extrema importância. Assim como a precisão, a repetibilidade é dada em percentagem relacionada à unidade de ângulo, porém, em algumas aplicações, ela é considerada mais importante do que a precisão, por exemplo, imagine um telescópio ou radar, que utilize um encoder como elemento de medição de inclinação. Caso o instrumento não tenha uma boa repetibilidade, poderão ocorrer erros no posicionamento quando o usuário necessitar posicionar o equipamento repetidas vezes nas mesmas coordenadas. Já em equipamentos em que os



movimentos são repetitivos e aleatórios, o fator de repetibilidade não é tão importante quanto a sua precisão.

### 11.1.1 Princípio de Funcionamento

Os encoders têm um princípio de funcionamento muito simples, baseia-se em três elementos básicos, um elemento emissor de luz infravermelha, outro receptor, e um disco ranhurado que corta perpendicularmente o feixe de luz infravermelha, emitida pelo conjunto ótico emissor/receptor.

O disco rotativo é montado entre os dispositivos óticos, que estão diretamente alinhados. Quando a luz infravermelha é cortada pelo disco, o sinal elétrico no sistema ótico receptor é interrompido, e analogamente quando o disco permite a passagem da luz infravermelha, o sinal elétrico no sistema ótico receptor é permitido, esses sinais são colhidos por um circuito eletrônico que amplifica, trata e filtra o sinal antes de disponibilizar na saída do encoder. De modo geral, a resolução do encoder é definida pela quantidade de ranhuras que o disco possui, assim, se definirmos um encoder com resolução de 1000 pulsos/revolução, muito provavelmente o disco ranhurado deverá possuir 1000 ranhuras. Acompanhe a ilustração abaixo para visualizar o mecanismo básico do conjunto:

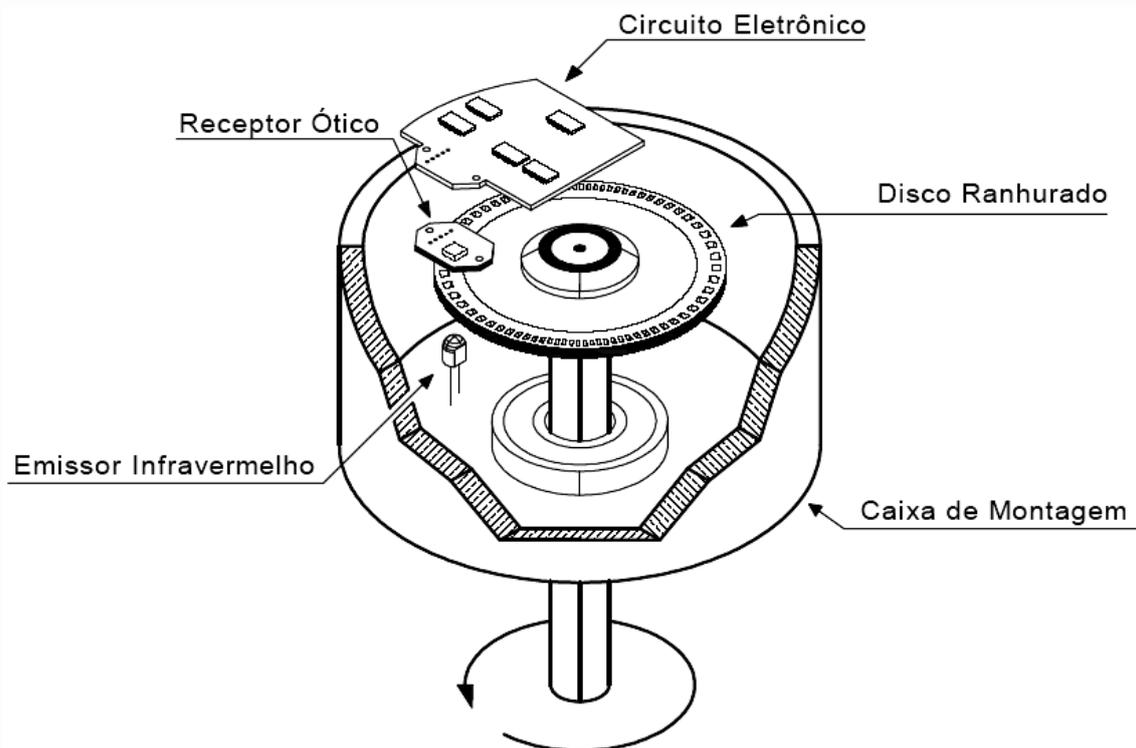


Figura 11.1

Um dos elementos mais críticos do encoder é o disco ranhurado, é ele que determina a precisão dos pulsos gerados pelo dispositivo. Por essa razão o disco é confeccionado em vidro cristal superfino e as ranhuras são gravadas no vidro de forma precisa, e cadenciada. Atualmente com as novas tecnologias



disponíveis, essas ranhuras são gravadas a laser, garantindo perfeita harmonia e cadência entre as marcações.

Um detalhe muito importante sobre equipamentos é o fato de o dispositivo ser muito sensível a choques mecânicos. Devido à fragilidade do material utilizado no disco (vidro), o dispositivo não pode receber impactos diretos ou quedas, isso pode ocasionar a quebra do disco ranhurado, e conseqüentemente a perda definitiva do instrumento. Outro detalhe que deve ser levado em consideração é a impregnação de poeira ou outro tipo de impureza no interior do equipamento, nesse caso o equipamento pode apresentar falhas ou imprecisão nas leituras. Para resolver o problema, basta o usuário desmontar o dispositivo com cautela e fazer a limpeza de todo o circuito, inclusive do disco ranhurado. Para essa aplicação, a utilização de álcool isopropílico é uma boa indicação.

## **11.2. Tipos de Encoders**

Os encoders ou geradores de pulso podem ser divididos em dois grupos, os incrementais e os absolutos. Ambos possuem a mesma função, transformar movimento rotacional ou linear em sinais elétricos padronizados, para medição de grandezas como velocidade, sentido de giro, aceleração etc. Abaixo abordamos o funcionamento dos dois modelos.

### **11.2.1 Encoder Incremental**

O encoder incremental é o mais comum, possui uma interface de sinais mais simples, além de possuir um custo mais baixo quando comparado ao encoder absoluto, por essa razão é um dos mais utilizados e preferidos pela maioria dos usuários. Esse dispositivo tem a característica de possuir no mínimo duas saídas defasadas em um ângulo de  $90^\circ$  entre si, e uma saída de referência, sendo as saídas identificadas como “A” (canal A) e “B” (canal B), e a referência definida como “Z” ou “0” (zero absoluto ou zero do encoder).

Os canais “A” e “B” fornecem pulsos quadrados quando o eixo é movimentado. A quantidade de pulsos fornecida pelos canais “A” e “B” é definida pela resolução do dispositivo, ou seja, se a resolução do encoder for de 500 passos/revolução, a cada giro completo do eixo ( $360^\circ$ ) teremos 500 pulsos no canal “A” e 500 pulsos no canal “B”, defasados em um ângulo de  $90^\circ$ . Já o canal de referência fornece um pulso quadrado com duração de no mínimo 1 pulso do canal “A”, porém somente uma única vez por revolução, ou seja, enquanto os canais “A” e “B” emitem uma quantidade de pulsos que depende de sua resolução, o canal “Z” ou “0” emite somente 1 pulso a cada revolução completa do eixo ( $360^\circ$ ), dessa forma, o usuário pode utilizá-lo como pulso de referência ou posição de zero absoluto do equipamento.



## Encoder Incremental

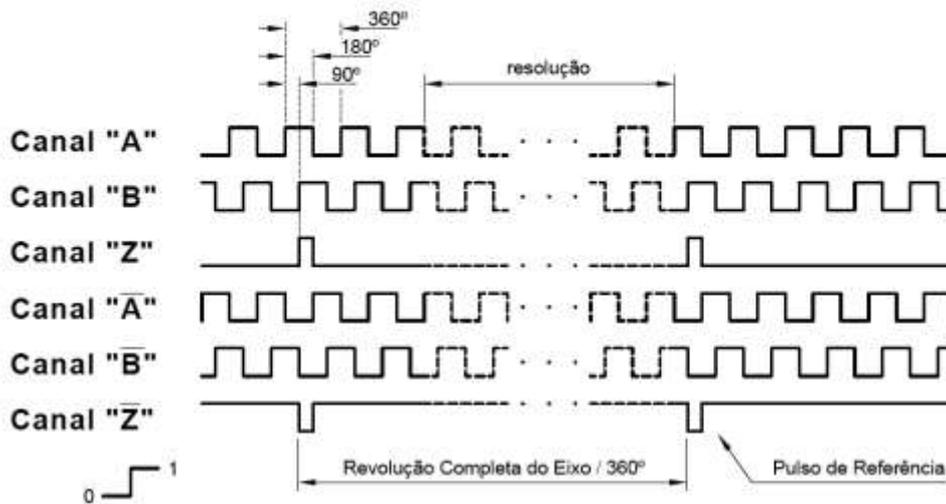


Figura 11.2

Esses dispositivos também podem ser encontrados com seis canais de saída, os canais básicos "A", "B" e "Z", e três canais inversos, "/A", "/B" e "/Z", esses canais representam a mesma seqüência de pulsos dos canais básicos, porém com sinal invertido, em outras palavras, quando o canal "A" estiver em nível alto (1) o canal "/A" estará em nível baixo (0), e assim analogamente para os canais "/B" e "/Z". Na figura 11.2, podemos visualizar a seqüência lógica desses sinais.

A definição de resolução de um encoder incremental é definida pela sigla PPR, que tem seu significado associado aos termos, passos/revolução ou pulsos/revolução. Assim, quanto maior for a resolução de um encoder incremental, maior será a quantidade de pulsos gerados a cada revolução do eixo, a resolução reflete a menor unidade de grau que o dispositivo poderá medir, por exemplo, se em um mecanismo de automação é necessário realizar medições de inclinação de uma plataforma em unidades de 1° grau, o usuário poderá optar por um dispositivo que possua uma resolução de 360 PPR, ou seja, a cada revolução completa do eixo do encoder teremos na saída 360 pulsos elétricos, nesse caso a cada pulso enviado ao sistema a plataforma se inclinou 1°. O cálculo de resolução de um encoder incremental é bastante simples, acompanhe:

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{pulsos}]} = \text{PPR} [\text{graus} \times \text{pulso}]$$

ou

$$\text{resolução} [\text{pulsos}] = \frac{360 [\text{graus}]}{\text{PPR} [\text{graus} \times \text{pulso}]}$$

Desse modo, se o usuário definir que o sistema deverá realizar medições em passos de 0,1° grau, devemos dimensionar um encoder com resolução de:

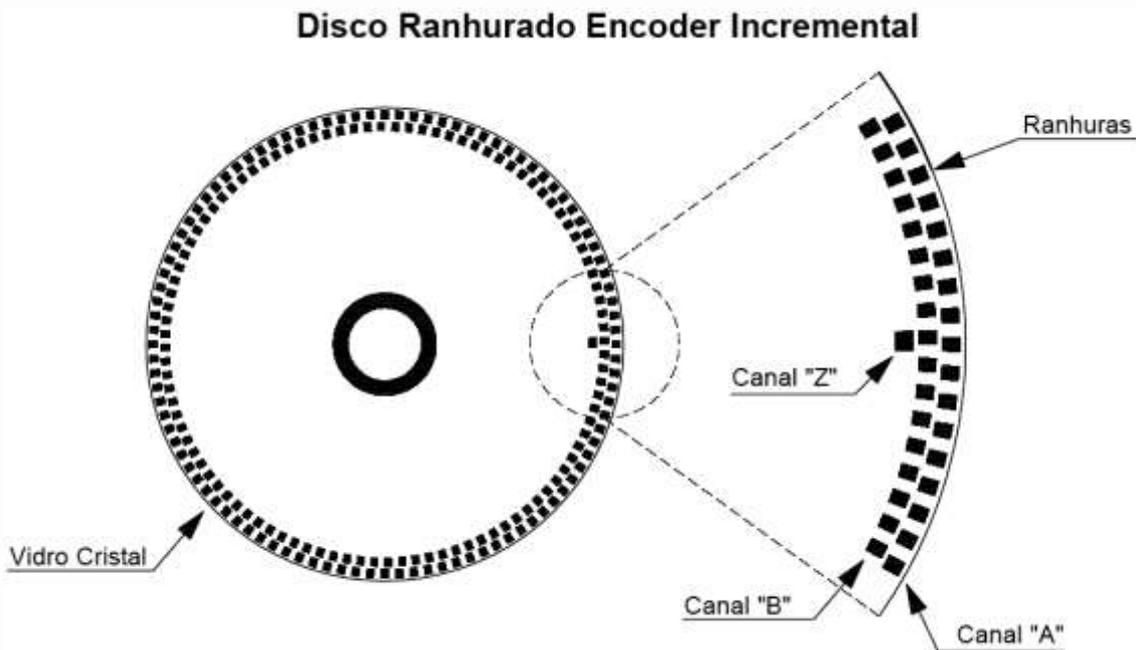


$$resolução [pulsos] = \frac{360 [graus]}{PPR [graus \times pulso]} \therefore \frac{360}{0,1} \therefore resolução = 3600 PPR$$

Nesse caso, a cada pulso enviado ao sistema, o eixo do encoder se deslocou 0,1° grau.

Em um ambiente industrial, a precisão de um encoder pode sofrer alteração, fatores externos como fadiga mecânica, transientes ou oscilações no suprimento de energia elétrica, temperatura elevada, erros introduzidos devido à instalação elétrica e a leitura dos sinais podem ser considerados as maiores causas de imprecisão, em sistemas automatizados baseados nesse dispositivo.

Como visto anteriormente, o disco ranhurado é um dos principais elementos que compõem um encoder, devido sua fragilidade e precisão, o dispositivo deve ser manuseado com muito cuidado. O disco ranhurado de um encoder incremental é exclusivo para esse tipo de dispositivo, ou seja, não pode ser utilizado para reparar encoders absolutos, por exemplo, abaixo podemos visualizar com detalhes a estrutura do disco ranhurado para encoder incremental, acompanhe:



Na figura 11.3, visualizamos em detalhes a estrutura de um disco ranhurado para um encoder do tipo incremental, as ranhuras são posicionadas entre a luz infravermelha do circuito ótico, e como a rotação do disco a luz infravermelha é cortado pelas ranhuras, caracterizando assim a seqüência de pulsos na saída do dispositivo, conforme observado na figura 11.2.

Quando utilizamos encoders incrementais as vantagens são inúmeras, interface elétrica mais simples, menor número de fiação para conexão, alta precisão, menor custo etc, porém existe uma pequena desvantagem: de modo



geral, para que o circuito se referencie, é necessário que o encoder tenha uma transição de um pulso elétrico no canal “Z”, isso ocorre somente uma vez a cada revolução do disco, então em um equipamento que utilize um encoder incremental como referência de posição deverá entrar em funcionamento e somente após obtermos uma transição no canal “Z” o circuito passa efetivamente a realizar medições na posição, por essa razão os encoders incrementais não podem ser utilizados em qualquer sistema, equipamentos onde não é possível realizar esse “primeiro ciclo” sem controle é recomendado à utilização de encoders do tipo absoluto. Esse detalhe de perda de referência também é observado quando o equipamento é desligado, sofre uma falta de energia ou o eixo do encoder é movimentado com o equipamento desligado.

### 11.2.2 Encoder Absoluto

O princípio de funcionamento de um encoder absoluto é semelhante ao encoder incremental, ambos utilizam-se do princípio de um disco rotativo ranhurado que, ao ser cortado por um feixe de luz infravermelha, disponibiliza na saída pulsos elétricos padronizados proporcionais à posição do eixo do equipamento. A principal diferença está na quantidade de saídas disponíveis e na codificação do sinal elétrico da saída. Enquanto no encoder incremental a posição inicial do eixo é dada a partir de um pulso no canal de referência, o encoder absoluto possui um código único para cada posição do eixo. Isto define ao encoder absoluto uma característica única, ele nunca perde a referência de posição real, mesmo com falhas de energia ou movimentação do eixo com o suprimento de energia desligado. Isso é possível graças ao código gravado no disco ranhurado, o código mais utilizado é o binário, pois é facilmente compreendido pela maioria dos dispositivos de interface eletrônica e não é necessário qualquer tipo de conversão para identificação da real posição do eixo do encoder.

A resolução de um encoder absoluto é dada em código binário, ao contrário da resolução de um dispositivo incremental, cuja quantidade de punhos é referido por revolução, o número de bits de um encoder absoluto determina a relação entre a posição em grau X código binário, ou seja, utilizando a fórmula básica de resolução digital podemos determinar que se um encoder absoluto, por exemplo, tiver sua resolução definida em 10 bits, equivale a dizer que em sua posição “zero” o valor obtido na saída será “0<sub>10</sub>” ou em código binário de 10 bits “0000000000<sub>2</sub>”, e em sua posição “final” (359°) o valor na saída será de “1023<sub>10</sub>” ou “1111111111<sub>2</sub>”, esses valores são obtidos avaliando o sinal elétrico na saída e convertendo-os em códigos binários, portanto, se o encoder tiver uma resolução de 10 Bits, teremos 10 saídas para codificação do sinal, se for de 14 bits teremos 14 saídas, e assim por diante, de forma geral estas saídas são identificadas como D0; D1; D2; D3; Dn, de modo que n é dado por “resolução - 1” [10 - 1 = 9], acompanhe o cálculo abaixo:

$$2^{N^{\circ} \text{ de Bits}} = \text{Resolução}$$

$$2^{10} = 1024 \text{ [ } 0_{10} \text{ - } 0000000000_2 \text{ a } 1023_{10} \text{ - } 1111111111_2 \text{ ]}$$

11



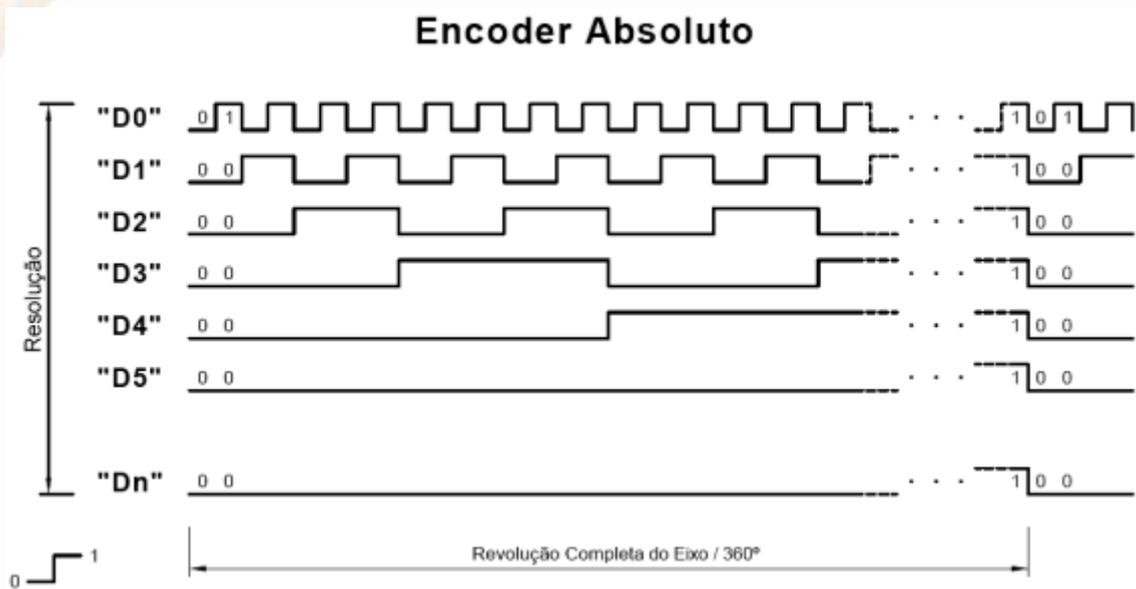


Figura 11.4

Para o cálculo de resolução em graus X código binário, podemos considerar o mesmo método utilizado para os encoders incrementais:

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = PPR [\text{graus} \times \text{Bit}'s]$$

ou

$$\text{resolução} [\text{Bit}'s] = \frac{360 [\text{graus}]}{PPR [\text{graus} \times \text{Bit}'s]}$$

Desse modo, para um encoder absoluto com resolução de 10 bits temos:

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = PPR [\text{graus} \times \text{Bit}'s] \quad \therefore \quad \frac{360}{1024} = 0,35^\circ \times \text{Bit}$$

Os encoders absolutos podem ser encontrados com resoluções que variam de 10 a 16 Bits. Considerando as operações acima, podemos determinar a resolução real de cada modelo, acompanhe:

### Encoder Absoluto 10 Bits

$$2^{\text{N}^\circ \text{ de Bits}} = \text{Resolução} \quad \therefore \quad 2^{10} = 1024$$

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = PPR [\text{graus} \times \text{Bit}'s] \quad \therefore \quad \frac{360}{1024} = 0,35^\circ \times \text{Bit}$$

*0,35° graus por transição de Bit*

### Encoder Absoluto 11 Bits

$$2^{\text{N}^\circ \text{ de Bits}} = \text{Resolução} \quad \therefore \quad 2^{11} = 2048$$

11



$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = \text{PPR} [\text{graus} \times \text{Bit}'s] \quad \therefore \frac{360}{2048} = 0,17^\circ \times \text{Bit}$$

*0,17º graus por transição de Bit*

### **Encoder Absoluto 12 Bits**

$$2^{N^\circ \text{ de Bits}} = \text{Resolução} \quad \therefore 2^{12} = 4096$$

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = \text{PPR} [\text{graus} \times \text{Bit}'s] \quad \therefore \frac{360}{4096} = 0,087^\circ \times \text{Bit}$$

*0,087º graus por transição de Bit*

### **Encoder Absoluto 13 Bits**

$$2^{N^\circ \text{ de Bits}} = \text{Resolução} \quad \therefore 2^{13} = 8192$$

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = \text{PPR} [\text{graus} \times \text{Bit}'s] \quad \therefore \frac{360}{8192} = 0,043^\circ \times \text{Bit}$$

*0,043º graus por transição de Bit*

### **Encoder Absoluto 14 Bits**

$$2^{N^\circ \text{ de Bits}} = \text{Resolução} \quad \therefore 2^{14} = 16384$$

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = \text{PPR} [\text{graus} \times \text{Bit}'s] \quad \therefore \frac{360}{16384} = 0,021^\circ \times \text{Bit}$$

*0,021º graus por transição de Bit*

### **Encoder Absoluto 15 Bits**

$$2^{N^\circ \text{ de Bits}} = \text{Resolução} \quad \therefore 2^{15} = 32768$$

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = \text{PPR} [\text{graus} \times \text{Bit}'s] \quad \therefore \frac{360}{32768} = 0,010^\circ \times \text{Bit}$$

*0,010º graus por transição de Bit*

### **Encoder Absoluto 16 Bits**

11



$$2^{N^{\circ} \text{ de Bits}} = \text{Resolução} \quad \therefore 2^{16} = 65536$$

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{Bit}'s]} = \text{PPR} [\text{graus} \times \text{Bit}'s] \quad \therefore \frac{360}{65536} = 0,005^{\circ} \times \text{Bit}$$

*0,005° graus por transição de Bit*

Essas características de codificação dos sinais das saídas são possíveis graças ao código gravado no disco ranhurado. Esse código representa fielmente a codificação binária tradicional e é muito semelhante à gravação do disco de um encoder incremental, porém, ao invés de possuir somente três conjuntos ópticos emissor/receptor para os canais “A” “B” e “Z”, no encoder absoluto existem vários conjuntos ópticos, cada grupo emissor/receptor é responsável pelo sinal de um canal, então, se o encoder possuir uma resolução de 12 bits, o conjunto óptico possuirá 12 pares, emissor/receptor, sendo um para cada canal, e assim de forma semelhante para os dispositivos com resoluções diferentes. A seguir na figura 11.5 podemos visualizar um exemplo de um disco ranhurado para encoder absoluto.

Análogo ao disco ranhurado de um encoder incremental, o encoder absoluto possui ranhuras características que definem a codificação do sinal de saída, o disco apresentado na figura 11.5 representa um disco com codificação binária tradicional, porém outras codificações são possíveis, desde que sejam equivalentes a um código digital. O código “GRAY”, por exemplo, também pode ser utilizado para definição de sinais de saída de encoders absolutos.

#### Disco Ranhurado Encoder Absoluto

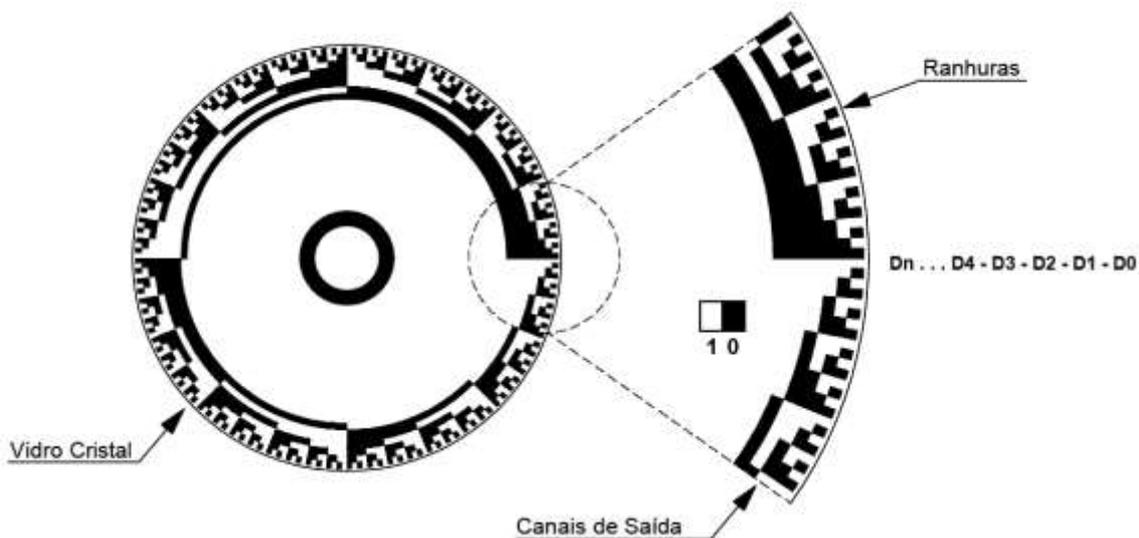


Figura 11.5

Do modo como é construído, o encoder absoluto indica instantaneamente a posição real do eixo do equipamento no momento que é energizado, portanto elimina a necessidade de um canal de referência (“Z”), assim em sistemas nos quais o encoder absoluto esteja instalado não é necessário o “primeiro ciclo” a vazio para referenciar todo o sistema.



As vantagens na instalação dos encoders absolutos são inúmeras, como, por exemplo, a manutenção da posição real, interface eletrônica pode ser ligada diretamente a circuitos digitais, alta resolução grau x bits, elaboração de lógicas de controle mais precisas. Porém, por outro lado, algumas desvantagens podem ser consideradas quando nos referimos aos encoders absolutos, como por exemplo, custos mais elevados, maior quantidade de fiação de dados, necessidade de interfaces de controle eletrônicas dedicadas, maior susceptibilidade a ruídos elétricos presentes na instalação.

### **11.3 Tipos de Saída**

A interface elétrica de um encoder é responsável pelo envio das informações através de circuitos eletrônicos padronizados, esses sinais são enviados diretamente a sistemas de controle como Clps, Robôs, Sistemas CNC ou diretamente para circuitos dedicados à base de microcontroladores. Esses sinais podem variar independentemente do tipo de encoder utilizado, incremental ou absoluto. Atualmente, com a ascensão de novas tecnologias e redes de comunicações, não é raro encontrarmos dispositivos dotados de interfaces seriais, como RS 232/485, ou em outros padrões comuns na indústria como Device Net, Interbus, ModBus, Profibus, dentre outros. A escolha do tipo de interface a ser utilizada deve ser definida no projeto do sistema, lembrando que a escolha de um padrão de comunicação pode acarretar complexidade no desenvolvimento da aplicação e demanda dispositivos especiais de conexão, como conectores específicos, adaptadores ou transdutores de sinal, por outro lado pode ser um grande aliado em sistemas de maior porte, pois a utilização de padrões de comunicação em rede diminui consideravelmente a fiação na instalação, disponibiliza maior imunidade a ruídos e interferências elétricas, possibilita um aumento de velocidade na transmissão dos dados etc.

Com exceção das interfaces seriais, os sinais elétricos mais comuns disponíveis para esses dispositivos podem ser classificados em seis grupos, são eles:

PNP - PNP Coletor Aberto – NPN - NPN Coletor Aberto - Push-Pull - Drive de Linha

#### **11.3.1 PNP**

As interfaces do tipo PNP se caracterizam por possuírem um transistor do tipo PNP com o terminal coletor ligado diretamente à saída, polarizado com um resistor na configuração pull-down, ou seja, o resistor está ligado, ao ponto de referência 0V. Assim, quando o disco ranhurado do encoder bloquear o feixe de luz infravermelho do conjunto ótico, o resistor força o sinal de 0V na saída, evitando assim flutuações de tensão na saída. Na figura 11.6 podemos identificar o circuito típico dessa configuração, acompanhe:

11



## Interface do tipo PNP

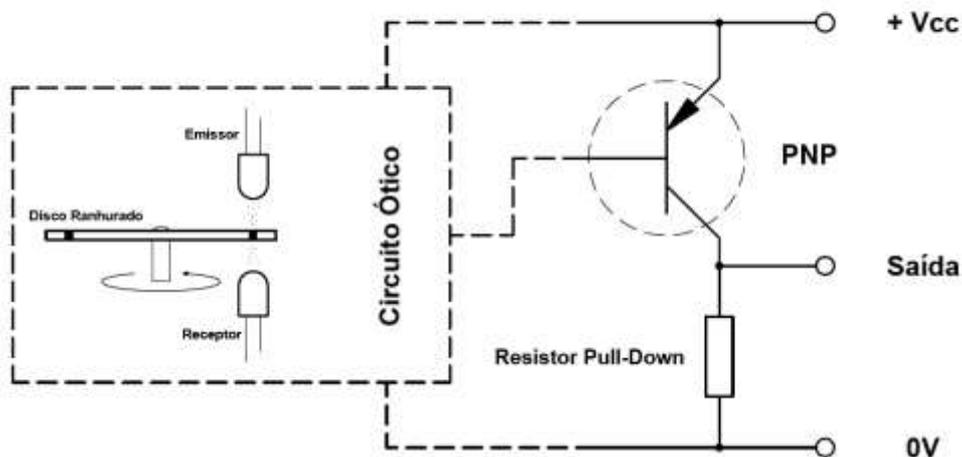


figura 11.6

O circuito típico visto na figura 11.6 se assemelha aos antigos circuitos TTL que eram utilizados para chaveamento de circuitos lógicos e que forneciam uma razoável quantidade de corrente elétrica, por essa razão o circuito é considerado compatível com a tecnologia TTL. Esse tipo de configuração pode sofrer alguns problemas quando a fiação utilizada para conexão do encoder até o sistema de controle for muito longa. Devido à alta frequência dos sinais de saída, o comprimento do cabo deverá ser levado em consideração na elaboração do projeto com interfaces do tipo PNP.

Quando a saída está definida em nível lógico alto (1) a tensão no coletor (saída) será muito próxima à tensão  $+V_{cc}$ , e de modo oposto quando a saída estiver definida em nível lógico baixo (0), a tensão no coletor será de 0V.

### 11.3.2 PNP Coletor Aberto

A interface do tipo PNP com Coletor Aberto segue as mesmas configurações observadas na interface do tipo PNP, porém com uma única diferença: não possui o resistor de polarização na configuração de Pull-Down, a polarização é feita a partir da carga, em outras palavras, a partir do circuito de controle. Com o ajuste do valor do resistor de polarização, existe a possibilidade de se obter níveis diferentes de tensões na carga.

O resistor de Pull-Down deve ser dimensionado para que o transistor opere em sua região quiescente, ou seja, na região onde obtemos o máximo de corrente de saída, quando dimensionado corretamente, em estado lógico alto (1) a tensão na saída se aproxima de  $+V_{cc}$ , e em estado lógico baixo (0) a tensão obtida é de 0V. A figura 11.7 demonstra essa configuração.

11



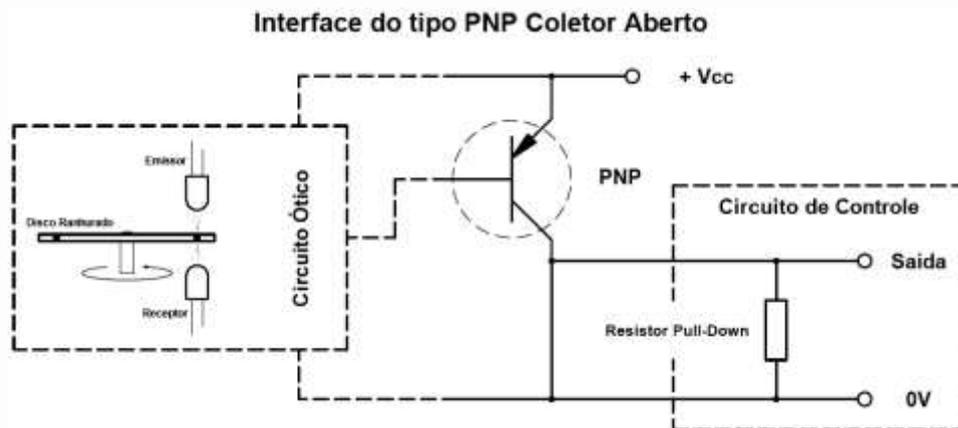


Figura 11.7

Os problemas com comprimento de fiação de conexão observados na interface do tipo PNP também são observados para a configuração de PNP Coletor Aberto.

### 11.3.3 NPN

A interface do tipo NPN pode ser considerada similar às interfaces vista nos itens 13.3.1 e 13.3.2, a diferença está no tipo de transistor utilizado, que neste caso, é do tipo NPN. Desse modo à referência de tensão 0V é ligada diretamente ao emissor do transistor e o resistor de polarização é do tipo Pull-Up, assim quando o disco ranhurado estiver bloqueando a passagem do feixe de luz infravermelho, a saída estará com nível alto (1), ou seja, +Vcc, que é forçado pelo resistor de polarização. Na outra situação, com o disco ranhurado permitindo a passagem do feixe de luz infravermelho, a saída terá um nível lógico baixo (0) 0V. Desse modo podemos afirmar que a interface do tipo NPN apresenta lógica inversa à interface PNP. Na prática, para um encoder incremental, esse fator pouco importa, pois o que interessa no sinal são as transições de “0” para “1” que representam o trem de pulso de contagem, já para os encoders absolutos deve ser levado em consideração, pois os sinais são colhidos por um circuito digital e ele deve ser configurado para a leitura de sinais com nível alto (1) ou baixo (0). Na figura 11.8 podemos visualizar o circuito típico de uma interface do tipo NPN.



### Interface do tipo NPN

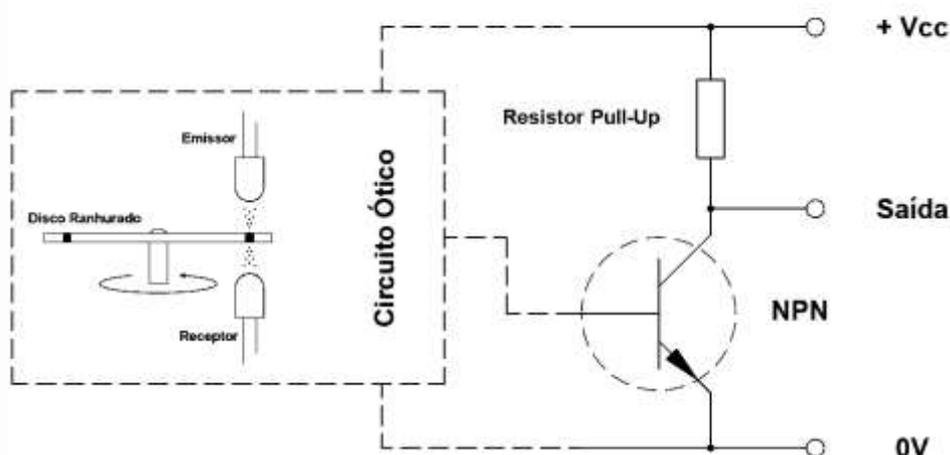


Figura 11.8

#### 11.3.4 NPN Coletor Aberto

A configuração do Tipo NPN com Coletor Aberto se caracteriza pela ausência do resistor de polarização, como foi visto anteriormente na interface PNP Coletor Aberto, porém agora o transistor é do tipo NPN e a polarização é realizada pela carga. O circuito típico que representa essa configuração pode ser visto na figura abaixo.

### Interface do tipo NPN Coletor Aberto

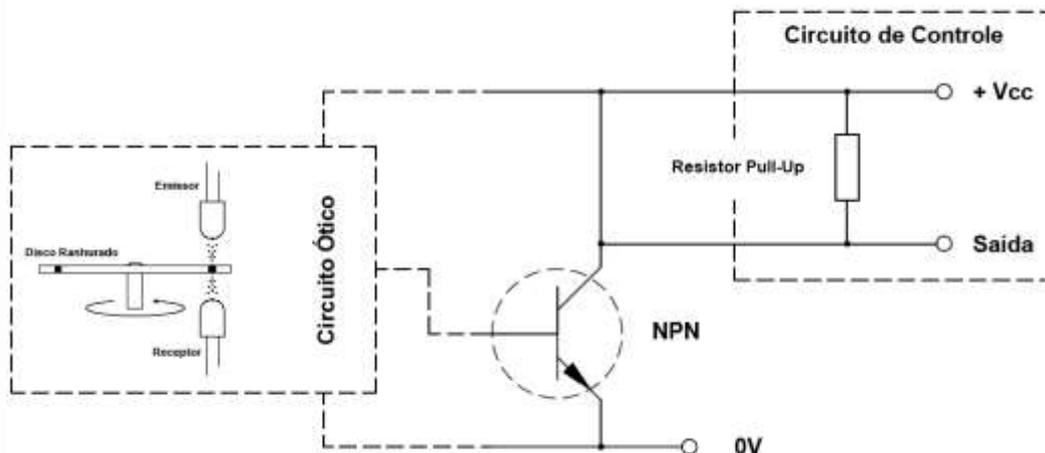


Figura 11.9

Nesse tipo de configuração também podem existir problemas quando a fiação de conexão entre o encoder e o sistema de controle for muito longa, portanto esse fator deve ser previsto na elaboração do projeto.

#### 11.3.5 Push-Pull

A configuração do tipo Push-Pull traz uma melhoria significativa ao circuito de saída de um encoder quando comparado às interfaces vistas



anteriormente, PNP/PNP Coletor Aberto e NPN/NPN Coletor Aberto. Nessas configurações podemos encontrar problemas quando utilizamos fiações muito longas entre os encoders e o sistema de controle devido a altas frequências envolvidas, isso se dá ao fato que o resistor utilizado para polarização deve possuir um valor de resistência maior do que a resistência de um transistor saturado, isso causa uma limitação para carga devido à alta impedância de saída.

Para resolver esse problema, foi desenvolvido um circuito que adiciona mais um transistor a configuração, caracterizando assim a interface Push-Pull, com isso a impedância da carga pode ser menor, dessa forma o rendimento do circuito é melhorado consideravelmente e as respostas a altas frequências é garantida, possibilitando uma maior distância entre o encoder e o sistema de controle com menor perda de sinal e conseqüentemente maior precisão no sistema.

O circuito típico de uma interface do tipo Push-Pull pode ser observado na figura abaixo, acompanhe:

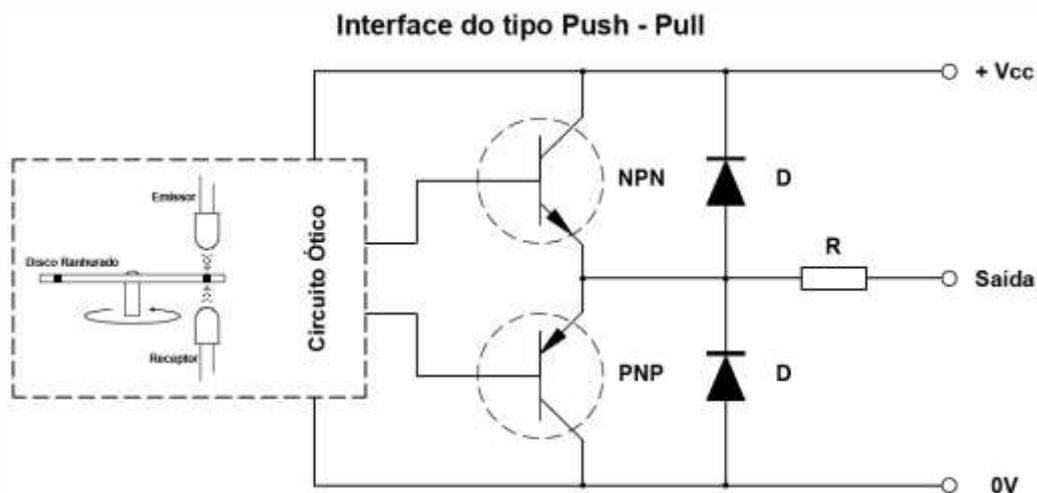


Figura 11.10

No circuito do tipo Push-Pull as transições altas (1) e baixas (0) são realizadas pelos transistores NPN e PNP, desse modo sempre haverá um dos transistores conduzindo, eliminando assim a necessidade de resistores de Pull-Up e Pull-Down. Por essa razão o circuito garante uma ótima estabilidade para altas frequências, amenizando falhas nos circuitos com altos comprimentos na fiação de interligação entre o encoder e o sistema de controle.

### 11.3.6 Drive de Linha

Na interface do tipo Drive de Linha existem algumas particularidades que influenciam o sinal enviado ao sistema de controle, ou seja, os sinais são enviados em pares, um complementar ao outro, isso significa dizer que para cada transição alta (1) teremos ao mesmo tempo uma transição baixa (0), desse modo o circuito de controle avalia a presença dos dois sinais ao mesmo tempo, se existir alguma falha em qualquer um dos sinais o pulso é ignorado e



corrigido. Esse tipo de sinal é geralmente utilizado em sistemas de extrema precisão e alta velocidade, pode ser implementado onde a distância entre o encoder e o sistema de medição for muito longa, porém a fiação de interligação deve ser blindada e aterrada para garantir assim a performance necessária para o sistema.

A interface eletrônica se dá através de circuitos integrados dedicados e específicos, o circuito típico desse tipo de interface pode ser observado na figura 11.11 abaixo.

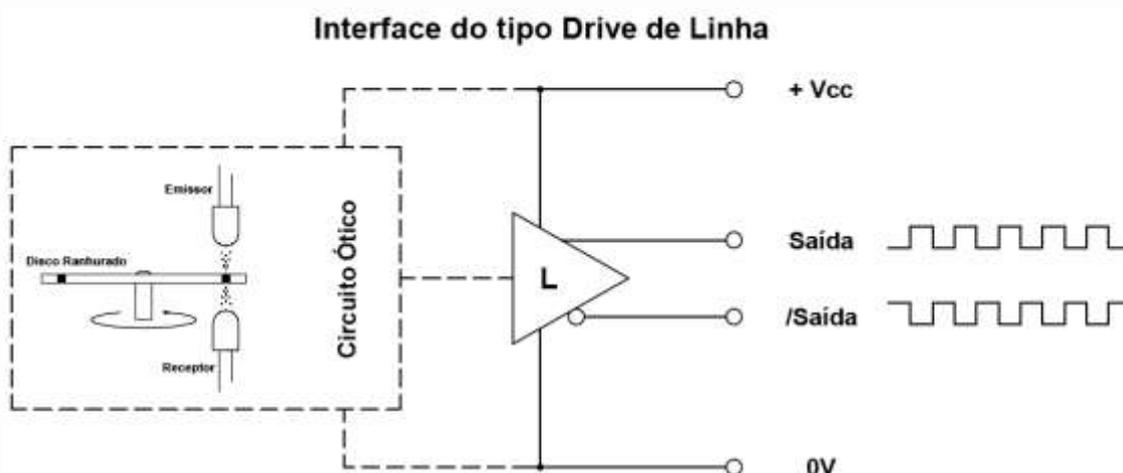


Figura 11.11

## 11.4. Interface com o CLP

Os controladores lógicos programáveis são os elementos mais utilizados na indústria que possuem a capacidade de processar os sinais elétricos provenientes dos encoders (exceto circuitos dedicados), portanto, a interface com esses dispositivos é bastante comum e utilizada, para isso o usuário deve conhecer as premissas básicas de uma lógica de controle que seja capaz de identificar e processar informações e obter o máximo de rendimento desse incrível e preciso instrumento chamado de encoder.

Para exemplificar essa interface, utilizaremos um exemplo prático, real e de simples compreensão, desse modo voltaremos a trabalhar com a ferramenta de programação para os Clps da Unitronics®, família Vision®, a mesma metodologia e princípios utilizados anteriormente em nossos exemplos, para isso idealizamos o exemplo abaixo, acompanhe:

Exemplo: *Uma indústria metalúrgica possui uma mesa giratória que posiciona peças abaixo de uma furadeira de bancada. Depois de posicionada a mesa, as peças são furadas manualmente por um operador, porém, como o mecanismo é rotacionado de forma manual, existem erros na furação causados pelo mau posicionamento da mesa, então foi solicitado ao projetista que desenvolvesse um sistema que movimentasse a mesa de forma automática com as seguintes características: o sistema deve permitir ao operador um ajuste da posição com precisão de 0,5° graus; deve possuir um raio de operação de 360° graus; o avanço da mesa será comandado por um simples botão, que, quando*

11



*pressionado, posiciona a mesa em intervalos fixos programados pelo operador, a cada toque no botão a mesa se posiciona no próximo intervalo programado. Então, de posse dessas informações, o projetista fez as seguintes considerações:*

- O sistema deverá possuir um CLP dotado de uma interface com display numérico, para que o operador possa inserir os valores de posicionamento, com resolução de 0,5° graus.*
- O sistema deverá possuir um botão para “Posicionamento” que, a cada toque do operador, a mesa se movimenta para próxima posição definida.*
- O sistema deve possuir um botão de referência, para que o operador indique ao sistema o ponto inicial da máquina. “Referência de Zero”.*
- O sistema deverá possuir um botão de posicionamento manual, para que o operador possa girar a mesa de forma contínua e assim alinhar mecânica e fisicamente a posição de zero, para início do processo.*
- O mecanismo deverá ser dotado de um motor acoplado diretamente à mesa, para que possa transmitir movimento rotacional.*
- O sistema obrigatoriamente deverá possuir um gerador de pulsos, acoplado ao eixo central da mesa para que esta transforme o movimento rotacional em pulsos elétricos padronizados, necessários para que o CLP faça o controle de posicionamento através do motor.*

A figura 11.12 abaixo expressa de forma simples os mecanismos envolvidos e utilizados para exemplificar a aplicação proposta. Assim o leitor poderá compreender com maior facilidade o exemplo proposto, acompanhe:



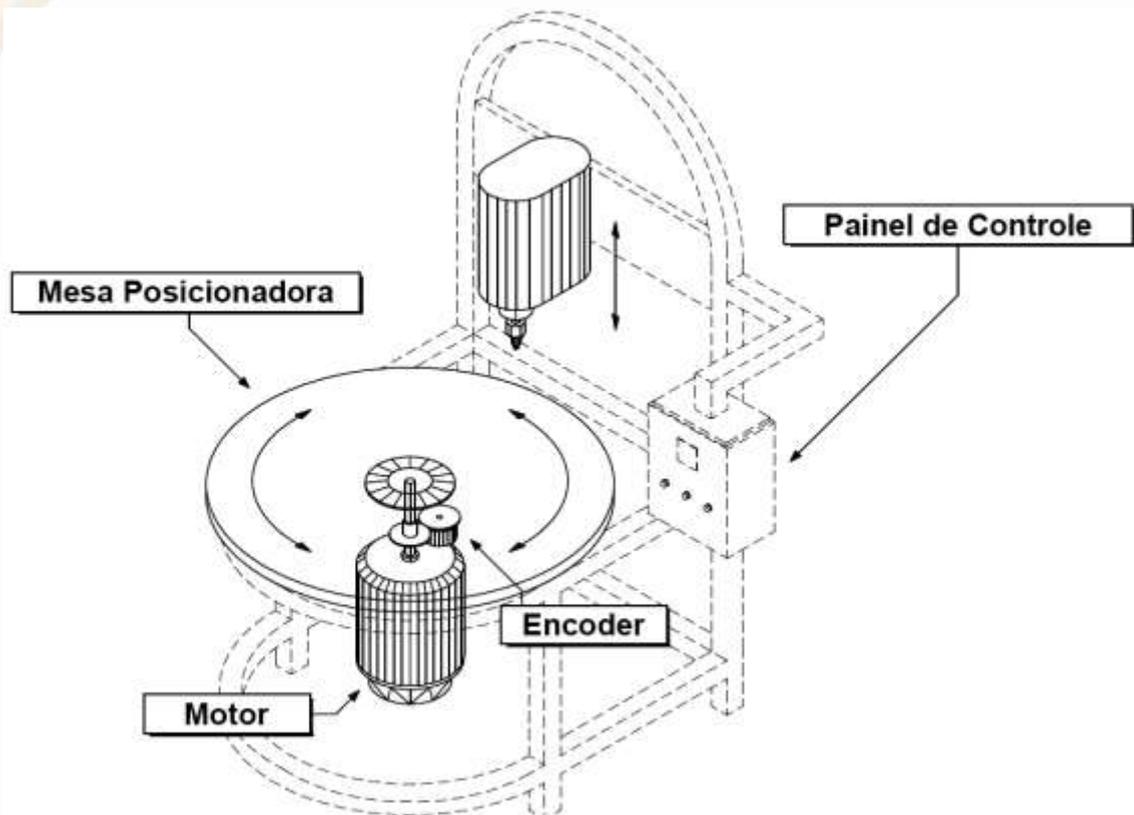


Figura 11.12

De posse dessas informações, podemos elaborar um sistema automatizado simples, que, com base nos sinais elétricos fornecidos pelo sensor de posição (encoder) o motor posicione a mesa sempre nos mesmos pontos, de forma precisa e rápida, evitando custos com peças mal furadas ou erradas devido à imprecisão operacional.

O primeiro passo na elaboração do projeto é a definição do tipo de encoder a ser utilizado. Como a aplicação é simples e os custos envolvidos são pequenos, o projetista definiu um encoder do tipo incremental, com saídas do tipo push-pull, Standard com canais de saída, "A", "B" e "Z". Para dimensionamento da resolução do dispositivo, o projetista fez o seguinte cálculo:

$$\frac{360 [\text{graus}]}{\text{resolução} [\text{pulsos}]} = PPR [\text{graus} \times \text{pulso}]$$

Como a resolução solicitada é de 0,5° graus, podemos definir que:

$$\frac{360 [\text{graus}]}{0,5 [\text{pulsos}]} = 720 PPR$$

Portanto o encoder escolhido possui resolução de 720 PPR.

O segundo passo para o desenvolvimento da aplicação é a elaboração do diagrama elétrico da instalação para posterior endereçamento das entradas



e saídas do CLP e definição dos estados lógicos de cada entrada e saída utilizada. A seqüência abaixo ilustra essas etapas. Diagrama elétrico proposto:

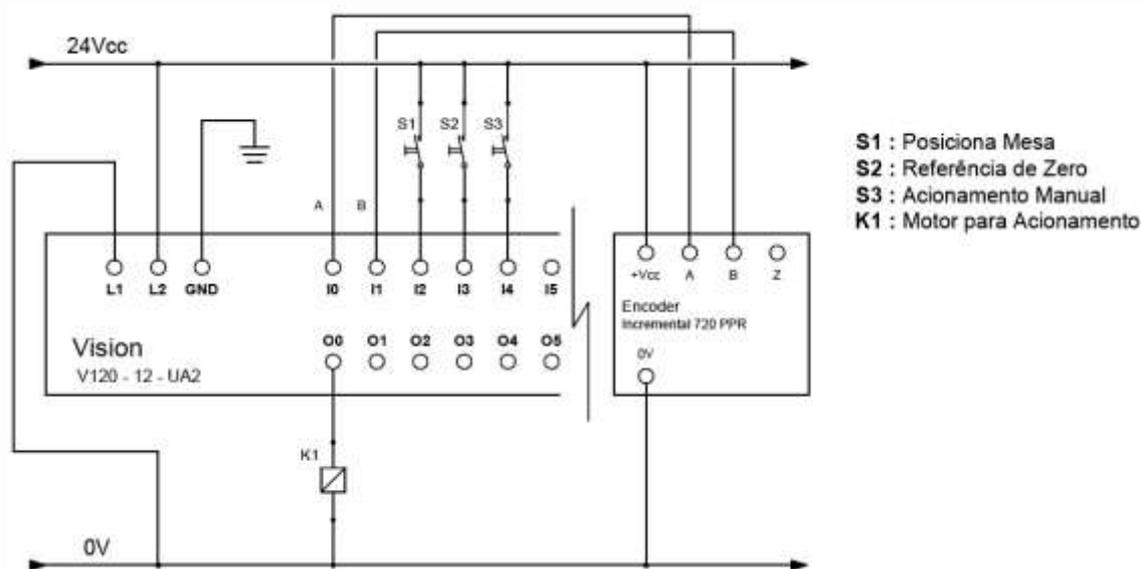


Figura 11.13

*OBS: A figura 11.12 ilustra o diagrama elétrico básico de conexão dos dispositivos externos com o CLP, um diagrama completo necessita de esquemas multifilares de força, comando e documentação técnica dos dispositivos utilizados.*

Endereçamento de Entradas e Saídas:

Dispositivos de Controle e Dispositivos Controlados	
<b>S1</b>	Botão de comando NA - "Posiciona Mesa"
<b>S2</b>	Botão de Comando NA - "Referência de Zero"
<b>S3</b>	Botão de Comando NA - "Posiciona Manual"
<b>K1</b>	Contactora de acionamento Motor de posicionamento da Mesa

Com o endereçamento das entradas e saídas, a próxima etapa do desenvolvimento da aplicação é a elaboração de uma tabela em que o projetista pode visualizar as condições lógicas das entradas e saídas, que o auxiliarão no desenvolvimento da aplicação:

Entradas Digitais	Nível " 0 " - "Falso"	Nível " 1 " - "Verdadeiro"
Entrada I0	Entrada de Sinal Canal "A" Encoder Incremental	
Entrada I1	Entrada de Sinal Canal "B" Encoder Incremental	
Entrada I2	Mesa parada	Mesa gira para próxima posição pré-definida
Entrada I3	Sem Ação	Referencia registradores internos com a posição "Zero" da mesa
Entrada I4	Mesa parada	Motor de gira de forma contínua



Saídas Digitais	Nível " 0 " - "Falso"	Nível " 1 " - "Verdadeiro"
Saída O0	Motor Parado	Motor girando

De posse dessas informações, o projetista pode iniciar a seqüência de programação do CLP. Abaixo identificamos a primeira etapa do programa, a identificação dos sinais do encoder e a geração de um valor numérico nos registradores internos referente à posição angular real da mesa do equipamento.

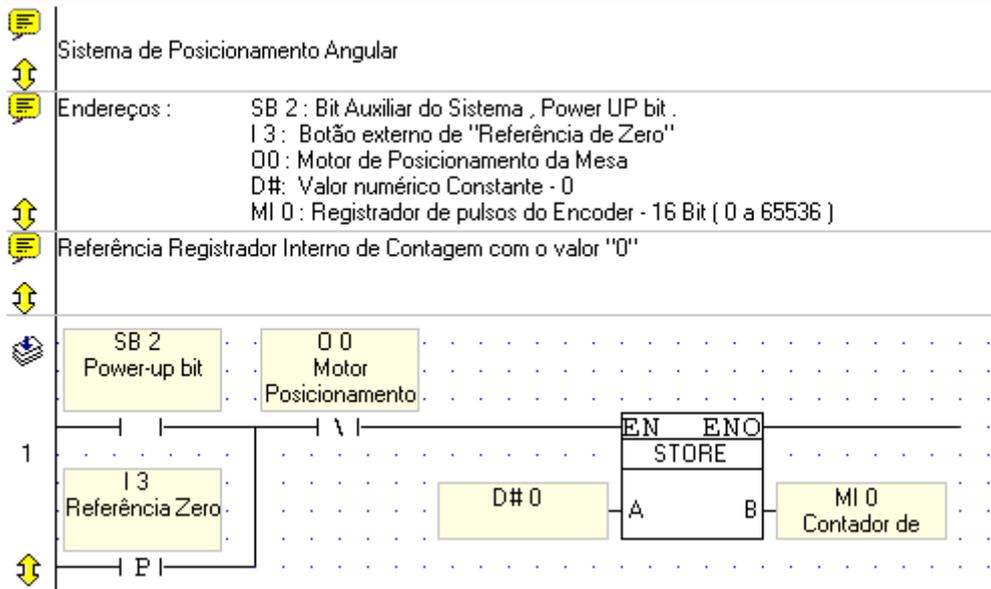


Figura 11.14

Na linha 1 do programa, representado pela figura 11.14, podemos notar o primeiro procedimento do programa, foi utilizado um bit auxiliar do sistema endereçado por "SB 2", esse bit intrínseco ao CLP tem a função de POWER UP, ou seja, após a energização do CLP, esse bit permanece ligado durante um ciclo de Scan, porém uma única vez após a energização do equipamento ou um evento de RESET do dispositivo, esse bit só se tornará ativo novamente se uma das ocorrências acima voltar a ocorrer. Em paralelo ao bit auxiliar "SB 2", temos um contato do tipo NA endereçado à entrada digital externa "I 3", nessa entrada foi conectado o botão "S2" de "Referência de Zero", portanto, quando "SB 2" ou "S2" forem verdadeiros, a condutividade lógica da linha é verdadeira até o bit endereçado à saída "O0" motor de posicionamento, esse bit garante que o sistema só se referencia quando o motor estiver parado, ou seja, com "O0" em estado lógico falso, se estas condições forem satisfeitas a função "Store" grava o valor numérico de uma constante (D# 0) no registrador interno "MI 0". O registrador "MI 0" foi endereçado para acumular os pulsos elétricos provenientes do encoder. Na próxima linha do programa verificaremos como os pulsos do encoder são armazenados nesse registrador.

*Deste modo definimos que a primeira linha do programa tem a função de "zerar" o registrador de pulsos do encoder, através do botão "S2". Zerando esse registrador atribuímos a posição "zero grau" para a mesa de posicionamento do equipamento. E, analogamente, quando o operador*



desligar e ligar o sistema, o programa atribui a referência “zero grau” para a posição atual da mesa de posicionamento, através de “SB 2”.

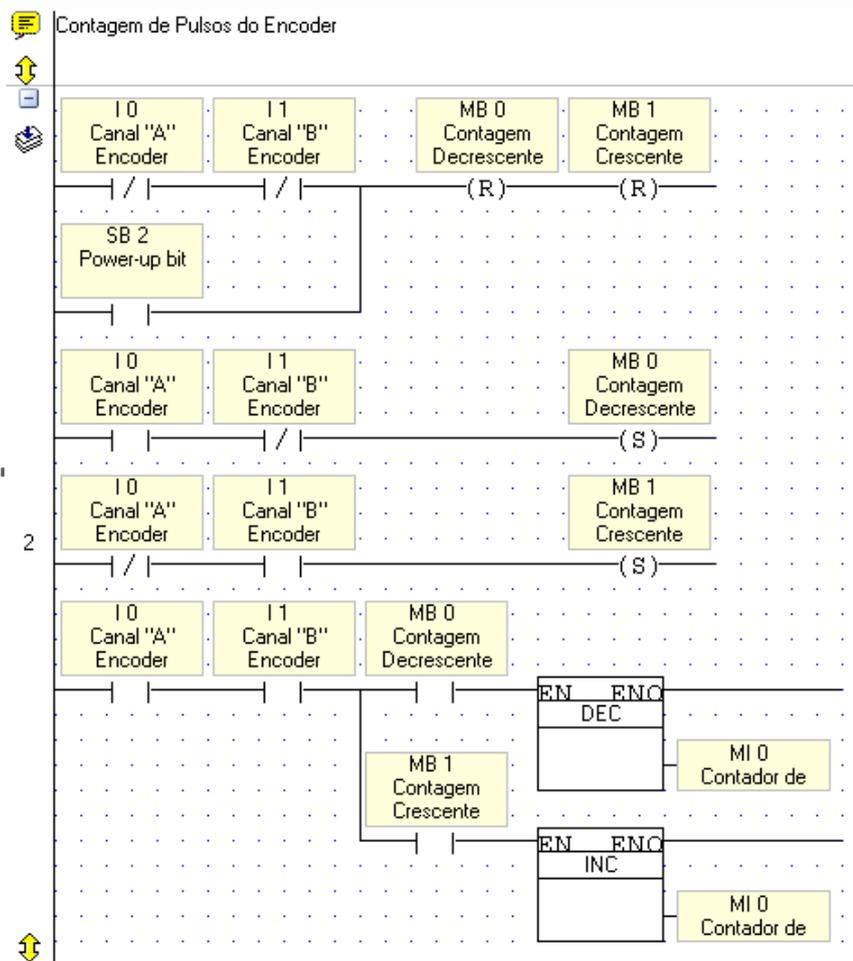


Figura 11.15

A próxima linha do programa demonstra como os pulsos do Encoder são armazenados em um registro interno da memória. O registrador de inteiros “MI 0” foi endereçado para armazenar os pulsos de contagem.

A lógica descrita nas três primeiras linhas através dos contatos de “I 0” e “I 1” (atribuídos a entradas do encoder) formam um circuito de intertravamento que, dependendo da seqüência da entrada dos pulsos dos canais “A” e “B”, podem acionar os bits “MB 0” e “MB 1” e assim determinar se a contagem é crescente ou decrescente, em outras palavras, se o encoder girar para o sentido horário (CW) a seqüência de pulsos nos canais “A” e “B” é diferente da seqüência de pulsos enviada se o sentido de giro do encoder for anti-horário (CCW). Essa identificação só é possível graças à defasagem dos sinais elétricos dos canais “A” e “B”. Para compreender como essa defasagem pode indicar o sentido de giro do equipamento, basta acompanhar o gráfico abaixo:



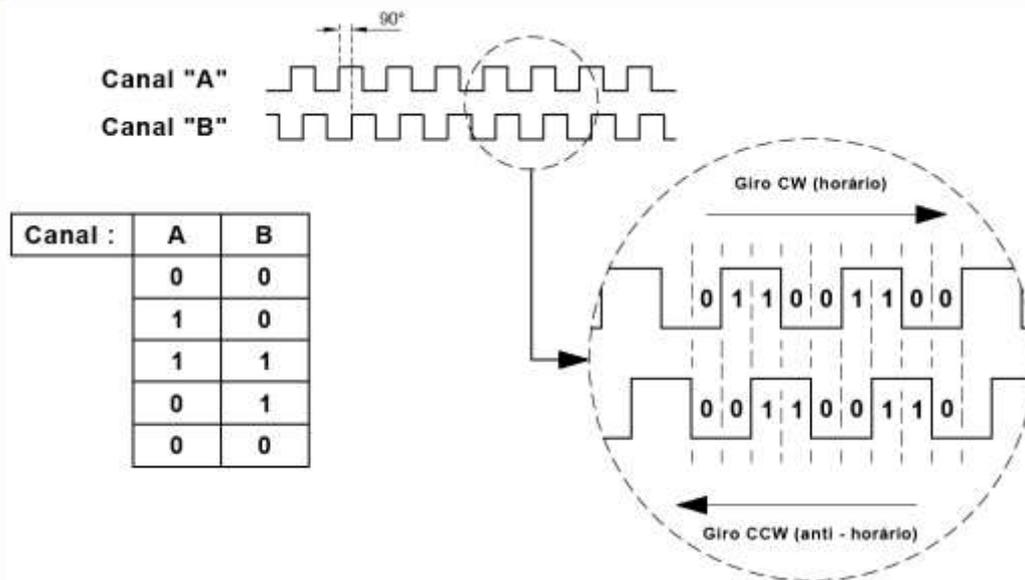


Figura 11.16

Observando a seqüência de pulsos enviados pelo encoder, podemos utilizar a defasagem entre os canais "A" e "B" para identificação do sentido de giro do encoder e, conseqüentemente, da mesa posicionadora. Então se observarmos passo a passo os níveis lógicos dos sinais, podemos identificar a seguinte informação:

- Canal "A" = 0 e Canal "B" = 0

Início do ciclo de pulsos, todas as vezes que os canais A e B estiverem simultaneamente com nível 0, teremos o início de um ciclo equivalente a um passo do encoder. Nessa situação o sistema de controle deve referenciar os bits de controle para avaliação do sentido de giro do eixo do encoder. Considerada como posição inicial da contagem.

- Canal "A" = 1 e Canal "B" = 0

A combinação observada pela presença de nível lógico 1 no canal "A" e nível lógico 0 para o canal "B", indica o deslocamento do eixo do encoder no sentido horário, nessa situação o sistema de controle deve armazenar a informação do sentido de rotação horária.

- Canal "A" = 1 e Canal "B" = 1

A combinação observada pela presença de nível lógico 1 nos canais "A" e "B", indica que o eixo do encoder continuou girando para o sentido horário, nessa situação o controle deve incrementar uma unidade de contagem no registrador de pulsos.

- Canal "A" = 0 e Canal "B" = 1



Esta combinação não terá efeito para o sistema de controle até que os canais “A” e “B” passem simultaneamente para o nível lógico 0, referenciando novamente os bits de controle (próximo passo). A combinação onde “A” = 0 e “B” = 1, só tem efeito se ocorrer após a seqüência “A” = 0 e “B” = 0. Nessa ocorrência o controle indica o deslocamento do eixo do encoder no sentido anti-horário, desse modo o sistema de controle deve armazenar a informação do sentido de rotação anti-horária.

- Canal “A” = 0 e Canal “B”= 0

Início do ciclo de pulsos, indica que o eixo do encoder completou um passo de sua resolução. Assim o sistema de controle deve se preparar para uma nova avaliação do sentido de giro e contagem de pulsos.

*OBS: A seqüência observada será sempre a mesma, diferenciando-se somente nos passos após a transição “A” = 0 e “B” = 0, após este passo o sistema avalia o sentido de giro, horário ou anti-horário, somente após esta avaliação o controle incrementa ou decrementa valores no registrador de posição, quando os canais “A” e “B” estiverem em nível lógico 1.*

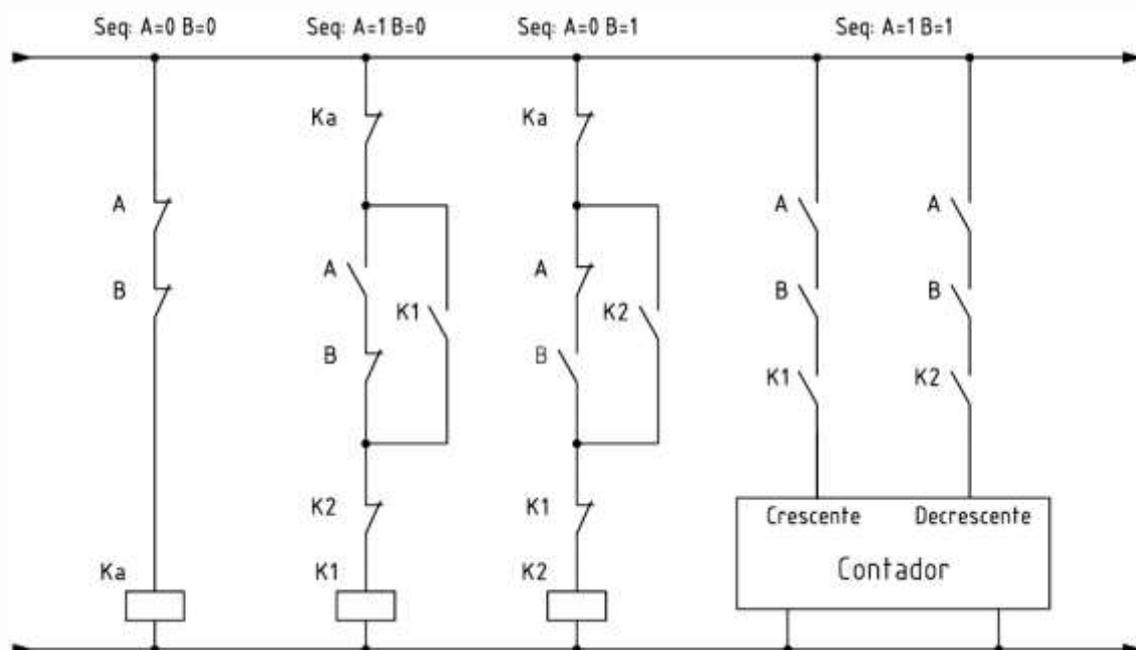


Figura 11.17

Avaliando de forma lógica as combinações, é possível montar um circuito que diferencie eletricamente os estados lógicos da seqüência de saída do encoder. Na figura 11.17 esse circuito está especificado em diagrama de contatos, desse modo à visualização da seqüência se torna mais simples.

Portanto, de acordo com o esquema acima, se o relé Ka estiver ligado, o circuito dos contactores K1 e K2 estará desligado, nessa condição, o comando assume a posição inicial da avaliação do sentido de giro do eixo do encoder. O circuito de acionamento do contactor Ka equivale a primeira linha do diagrama Ladder da fig 11.15 e ao primeiro passo lógico das entradas I0 e I1, canal “A”= 0 e canal “B”= 0.



Os circuitos de acionamento de K1 e K2, dependem exclusivamente do sentido de giro do eixo do encoder. Se o sentido for horário, canal "A" = 1 e canal "B" = 0, o contactor K1 entra em operação, mantendo-se acionado por meio de seu contato auxiliar. Por outro lado, se o sentido de giro do eixo do encoder for anti-horário, canal "A" = 0 e canal "B" = 1, entra em operação o contactor K2, que analogamente se mantém acionado por meio de seu contato auxiliar. Note que, uma vez que o contactor K1 entra em operação, ele causa um bloqueio em K2, e vice versa, se K2 entra em operação causa um bloqueio em K1. Esta configuração se mantém até que o contato de Ka entre em operação, desligando o circuito que estiver em operação.

Após o circuito de seleção do sentido de rotação ser acionado (K1 ou K2), a próxima seqüência lógica do encoder é dada pela combinação: canal "A" = 1 e canal "B" = 1, assim, quando esta seqüência for verdadeira podemos incrementar ou decrementar a contagem de um pulso no registro do contador de posição (MI 0). Depois de completada esta etapa, o encoder terá finalizado um ciclo completo de 360°, ou seja, um passo de sua resolução. Lembrando que o registrador de inteiro "MI 0" está com valor zero, valor este atribuído na energização do equipamento, ou com a ação do botão "S2", linha um do programa (figura 11.14).

Terminada esta etapa, a seqüência lógica é iniciada novamente, com os canais "A" = 0 e "B" = 0, proporcionando um ciclo contínuo, enquanto houver movimento no eixo do encoder.

O valor máximo que o registro "MI 0" pode assumir é 720. Este valor é determinado pela resolução do encoder utilizado, que fornece 720 pulsos a cada revolução de 360°, portanto, se a mesa girar no sentido horário (pulsos no bit "MB 1") o valor de "MI 0" será incrementado de 0 a 720. Por outro lado, se o operador durante o processo movimentar a mesa no sentido anti-horário (pulsos no bit "MB 0") o valor de "MI 0" será decrementado de 720 a 0. Esta lógica garante total mobilidade de movimentos para a mesa, dessa forma o sistema desenvolvido é "bastante inteligente" para descobrir se a mesa está girando para o sentido horário ou anti-horário.

Como não estamos utilizando o canal "Z" do Encoder para informar ao sistema que o eixo do sensor completou uma revolução de 360°, devemos criar uma lógica que force os valores máximo e mínimo no registro "MI 0". Se isso não for elaborado quando a mesa completar uma revolução de 360° a lógica desenvolvida para contagem continuará a incrementar ou decrementar o valor de "MI 0", dessa forma nosso sistema apresentará erro, pois os valores poderão ultrapassar o range de 720 pulsos ou assumir valores menores que zero, a solução é simples, basta desenvolver uma lógica com comparadores e no momento certo gravar no registro "MI 0" o valor desejado, acompanhe a terceira linha do programa para visualizar essas novas condições.



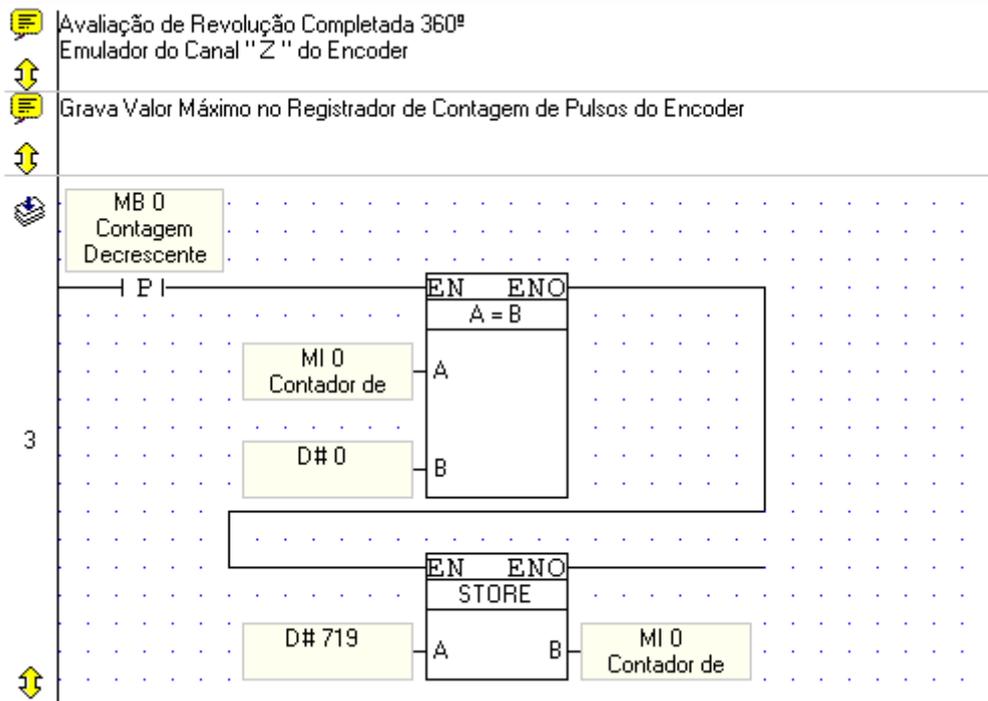


Figura 11.18

Na linha 3 do programa, utilizamos o bloco de função, de comparação de igualdade para avaliar se a contagem está próxima do fim ou início, dependendo do sentido da contagem.

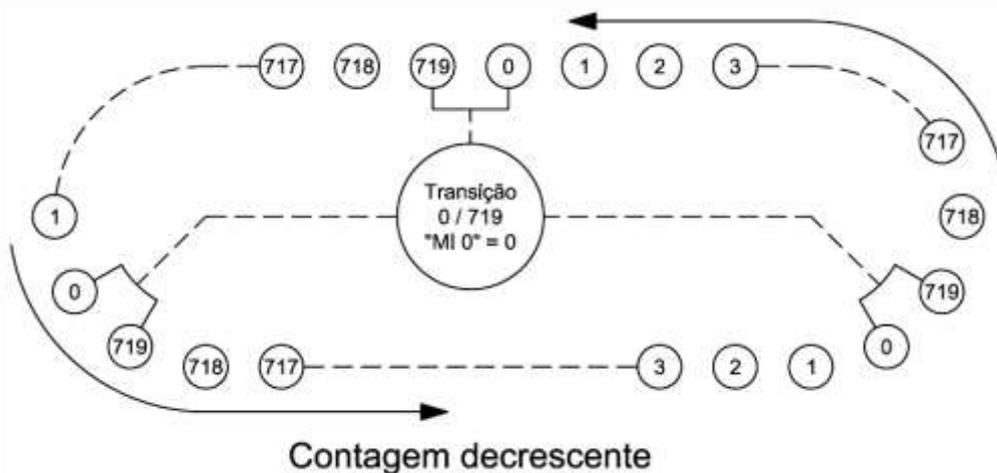


Figura 11.19

Isso é relativamente simples, pois os bits auxiliares “MB 0” e “MB 1” já disponibilizam essa informação, ou seja, se a contagem estiver sendo realizada através do bit “MB 0” sabemos que o encoder estará girando no sentido anti-horário, portanto o valor presente em “MI 0” estará sendo decrementado, por exemplo, se o registro “MI 0” estiver com o valor 300, a cada pulso de “MB 0” o valor diminui, 299 - 298 - 297 - ... - 0, desse modo, quando o valor de contagem chegar ao limite de contagem (0), “MI 0” deverá assumir o valor de 719, essa situação indica que a revolução da mesa atingiu o valor mínimo, ou seja, está na posição zero ou 0º, desse modo à lógica elaborada utiliza a transição positiva do bit “MB 0” para gravar no registrador de contagem o valor máximo



(719) no momento exato em que a contagem atingir o valor “0”, portanto se a contagem do encoder estiver sendo realizada através de “MB 0” os valores em “MI 0” assumirão a seqüência ilustrada pela figura 11.19.

Do mesmo modo, a situação inversa também é avaliada de forma similar a esse procedimento. Se a contagem for executada pelo bit “MB 1” o encoder estará girando para o sentido horário, então os valores em “MI 0” estarão sendo incrementados, por exemplo, se o registrador estiver com um valor de 300, a cada pulso de “MB 1” o valor aumenta, 300 - 301 - 302 - ... - 719, assim quando o valor de contagem atingir o valor máximo (719), “MI 0” deverá assumir o valor “0”, indicando que a movimentação da mesa atingiu o valor máximo ou 360°, com essa ação o registrador assume o valor inicial e a contagem se inicia novamente, a linha 4 do programa define esta lógica de controle:

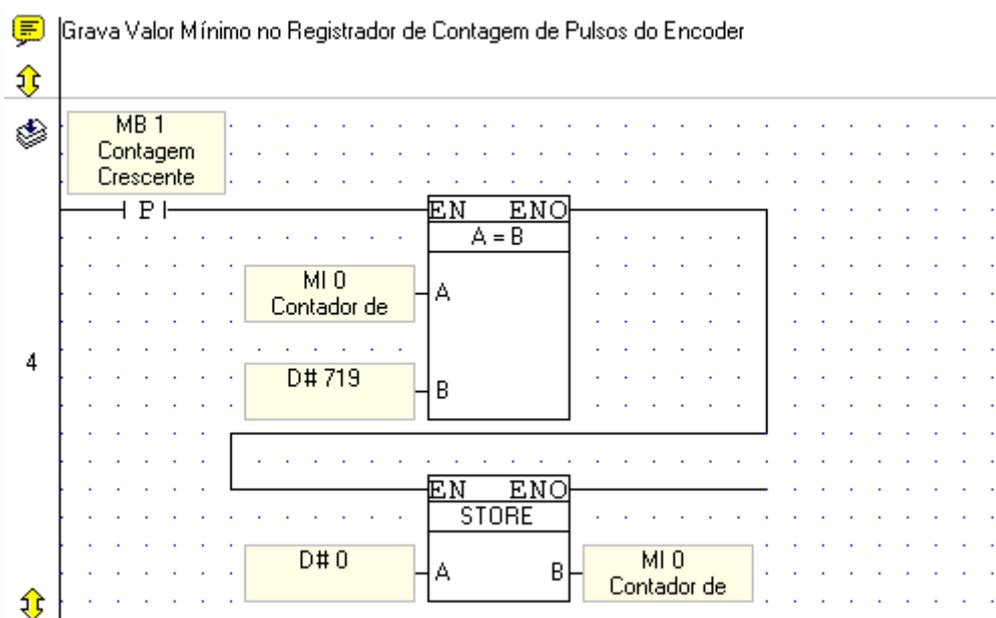


Figura 11.20

Na lógica observada na figura 11.20, o bit “MB 1” configurado para acionamento em sua transição positiva grava o valor mínimo em “MI 0” no exato momento em que o valor do registrador de contagem do encoder for igual ao valor máximo permitido (719) , nesse instante o valor “0” é gravado através do bloco de função “STORE” no registrador de pulsos do encoder, desse modo à contagem é iniciada novamente a partir do zero . A seqüência de contagem nessa situação pode ser observada na figura 11.21 a seguir.



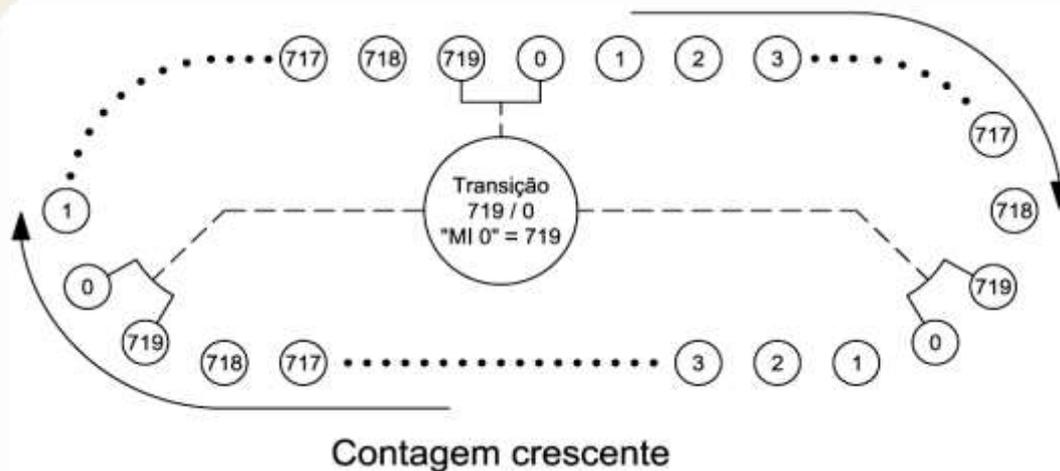


Figura 11.21

Com esses procedimentos, finalizamos a lógica de controle que avalia os sinais provenientes do encoder e identificamos as informações necessárias para que o sistema funcione conforme idealizado, porém ainda não temos os valores escalonados na unidade desejada, ou seja, em unidades de medida de ângulo (graus), pois o valor disponível no registro "MI 0" é somente uma referência de contagem dos pulsos gerados pelo encoder, em que o valor "0" equivale à posição "0°" e o valor "719" a uma posição de "360°", com auxílio do bloco de função de linearização, visto anteriormente em outros exemplos, podemos resolver esse problema de forma simples, basta inserir os dados corretos de linearização, que teremos em um registro interno os valores já correspondentes na unidade desejada. A linha 5 do programa ilustra essa seqüência, acompanhe:

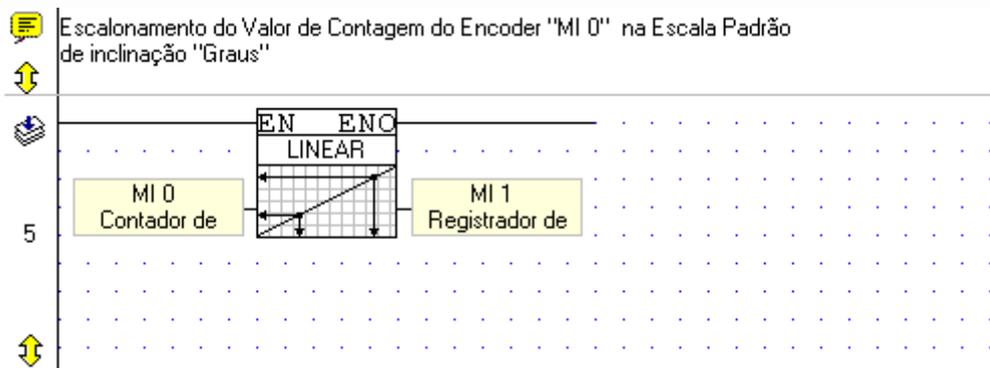


Figura 11.22

Linearization					
Params	Func	Operand	Address	Format	Description
IN	X1	D#	0	DEC	Linear conversion: X1 Value
	Y1	D#	0	DEC	Linear conversion: Y1 Value
	X2	D#	719	DEC	Linear conversion: X2 Value
	Y2	D#	3600	DEC	Linear conversion: Y2 Value
OUT	X	MI	0	DEC	Contador de Pulsos
	Y	MI	1	DEC	Registrador de Posição



Figura 11.22A

Como podemos visualizar na figura 11.22, a utilização do bloco de função de linearização é muito simples, basta inserir os dados de configuração nos parâmetros de entrada corretamente (fig 11.22A), onde identificamos os parâmetros:

- X1 : Valor mínimo do eixo "X" = 0
- Y1 : Valor mínimo do eixo "Y" = 0
- X2 : Valor máximo do eixo "X" = 719
- Y2 : Valor máximo do eixo "Y" = 3600
- X : Valor de entrada de sinal sem escalonamento ("MI 0")
- Y : Valor de saída de sinal linearizado ("MI 1")

Note que o valor máximo de linearização da saída está configurado para 3600 e não para 360. Isso é necessário pois o sistema trabalha com uma precisão de 0,5° grau, e para não utilizar registradores internos específicos para números com ponto flutuante o projetista adotou esse recurso. Assim, quando o operador necessitar de que o sistema opere com um valor de 100,5° graus, por exemplo, os registros internos apontarão para o valor 1005. Essa prática é muito comum quando trabalhamos com sistemas que operam somente com números inteiros, esse recurso facilita qualquer lógica que opere com instruções matemáticas, já que os operadores básicos matemáticos e operadores de comparação são especificados para operação com resolução de 16 bits, ou seja, podem assumir valores que variam de 0 a 65535, na maioria dos casos.



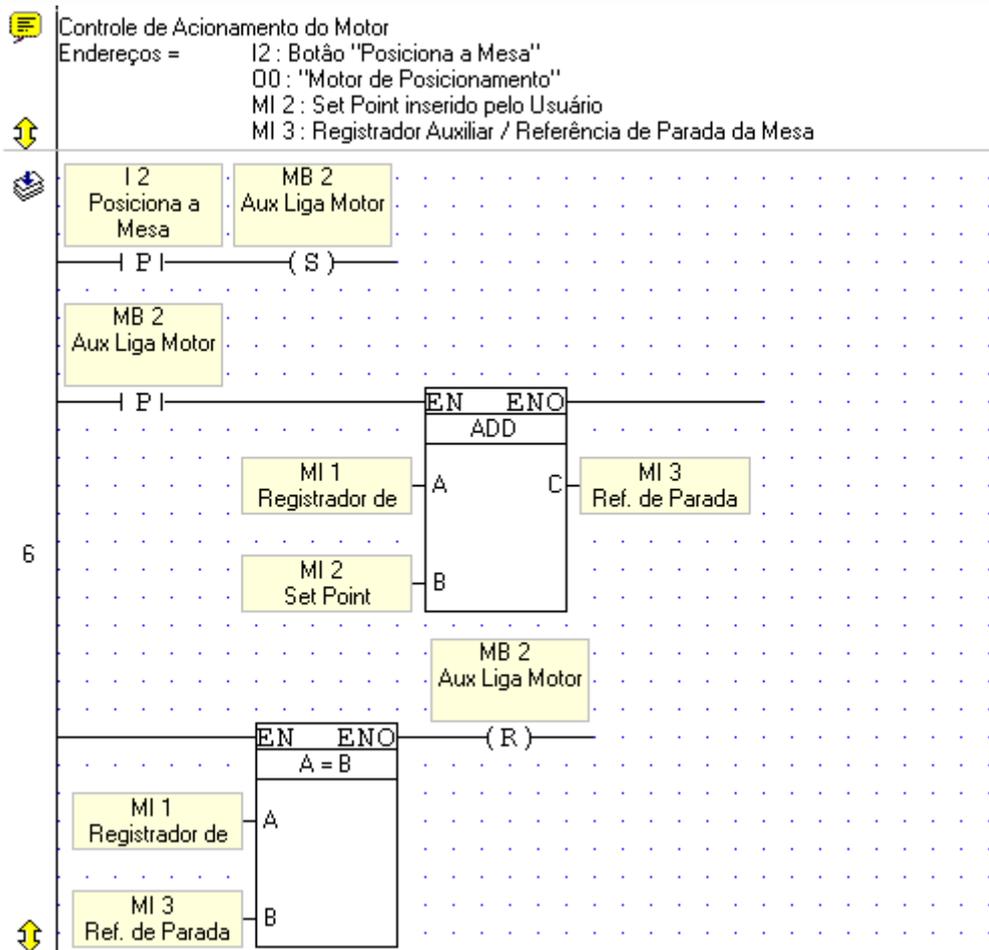


Figura 11.23

Agora temos um registrador interno endereçado por “MI 1”, que possui o valor da posição real, linearizada na escala desejada 0° a 360°, portanto, desse ponto em diante todos os raciocínios do programa podem ser elaborados com os parâmetros reais do processo.

Na seqüência do programa, devemos elaborar a lógica responsável pelo acionamento do motor, em passos definidos pelo operador, essa seqüência está descrita no escopo do projeto e define que ao toque do botão “S1” endereçado para entrada “I2”, o motor entre em funcionamento e movimente a mesa da posição atual para próxima posição programada pelo operador. Essa operação é bastante simples, pois já possuímos a informação real de posição através do registrador “MI 1” diretamente em graus, portanto, com apenas alguns comandos simples, essa ação pode ser obtida, acompanhe a próxima linha do programa ilustrada pela figura 11.23.

Na linha 6 do programa, identificamos o circuito responsável pelo acionamento do motor de posicionamento da mesa, através da transição positiva do contato aberto atribuído a “I2” (botão S1) o bit auxiliar “MB 2” é setado, quando um bit é acionado pelo operador “Set” (S) ele permanecerá acionado até que em outro ponto do programa receba o comando “Reset” (R), então, com essa ação atribuímos que se esse bit estiver ligado, o motor estará girando, como comentário foi atribuído o texto “Auxiliar Liga Motor”. Desse



modo, o projetista pode identificar que esse bit aciona diretamente a saída correspondente ao motor, ou seja, "O 0".

A partir do momento em que o bit "MB 2" for verdadeiro, ou seja, estiver setado, o motor estará girando, e deverá receber o comando de Reset somente quando a mesa atingir a posição programada pelo operador, portanto o projetista deverá elaborar uma lógica que analise o valor de Set Point inserido pelo operador e o compare com a posição real da mesa, quando essa operação for verdadeira, ou seja,

$$< \text{Posição Real} = \text{Posição de Set Point} >$$

O circuito deverá desligar ou ressetar o bit "MB 2". Essa ação irá parar o motor exatamente no ponto desejado. Para elaborar essa lógica, o projetista utilizou dois registradores de inteiros de 16 bits, o primeiro "MI 2" é responsável pelo armazenamento do valor de Set Point, inserido pelo usuário em qualquer ponto do programa via IHM, e o outro registrador atribuído a "MI 3" que simplesmente armazena o valor proveniente de uma operação matemática de adição. Após esse procedimento, o valor contido em "MI 3" é comparado com a posição real da mesa "MI 1", se os resultados forem iguais o bit auxiliar "MB 2" que comanda o motor é desligado, ou seja, ressetado, parando o motor instantaneamente. A operação matemática inserida na lógica do programa é necessária, pois a cada toque do botão "S1", o ponto de parada se desloca a um incremento definido pelo valor de Set Point; assim, a cada passo completado, o valor utilizado no comparador de igualdade muda automaticamente, dando ao sistema total liberdade de movimentação. Acompanhe a metodologia de cálculo nos diagramas abaixo:

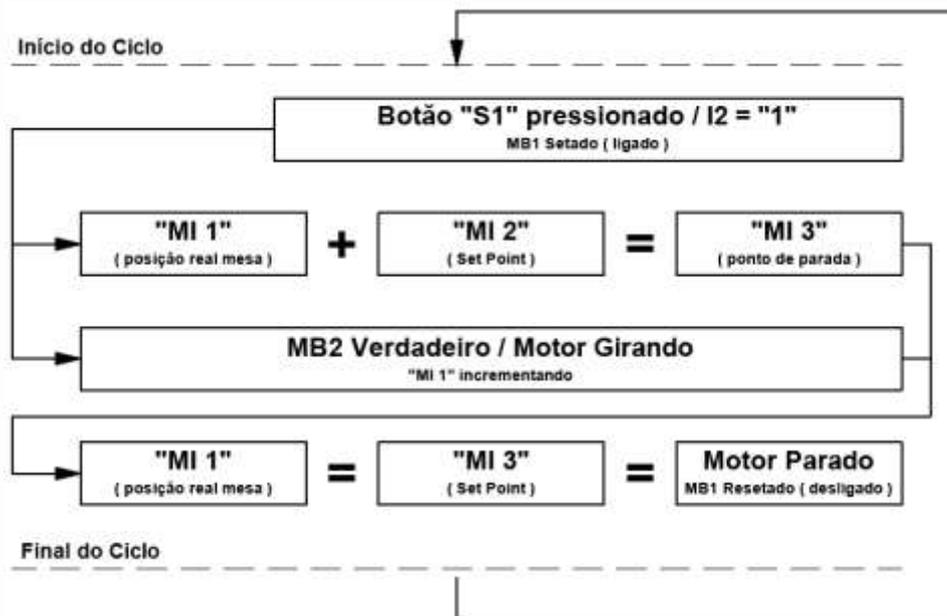


Figura 11.24



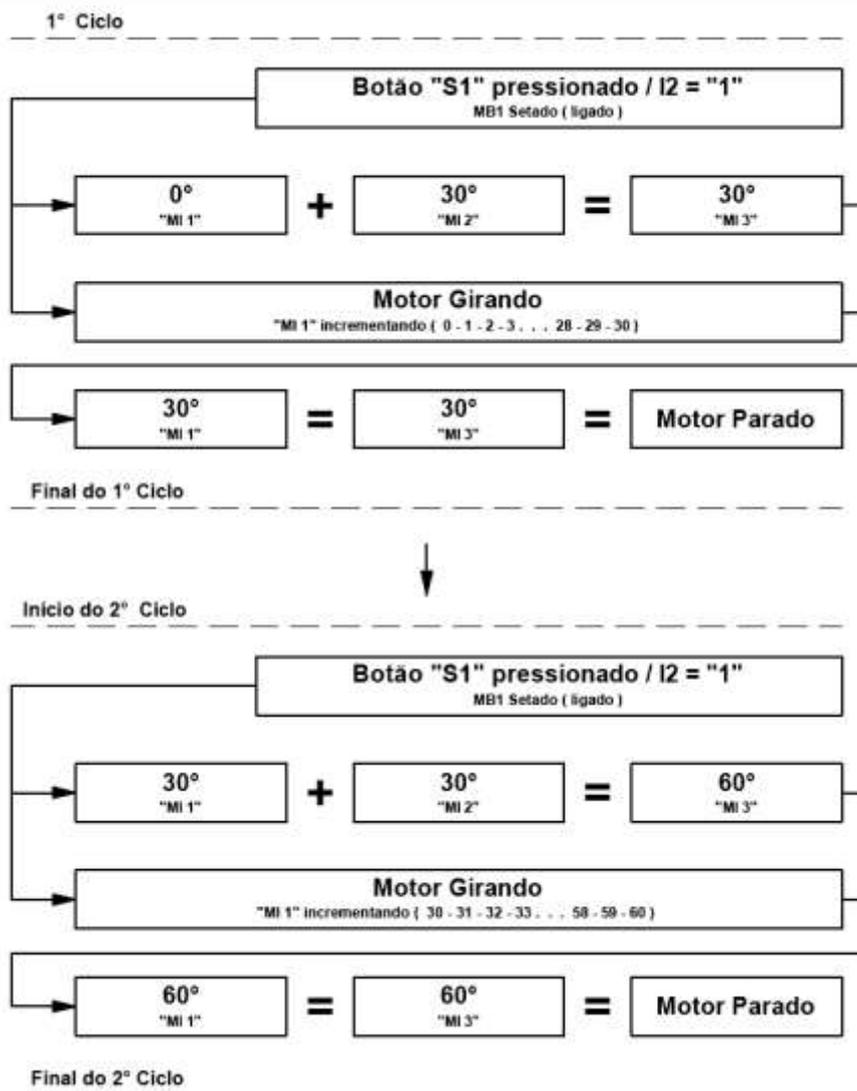


Figura 11.25

Segundo essa metodologia, a cada ciclo iniciado pelo botão "S1" o valor de parada é definido somando-se o valor da posição real/atual da mesa com o valor de Set Point programado. No diagrama da figura 11.25 segue ilustrada a metodologia com valores reais, assumindo, por exemplo, um valor de Set Point de 30°, e a mesa partindo do ponto Zero (0°).



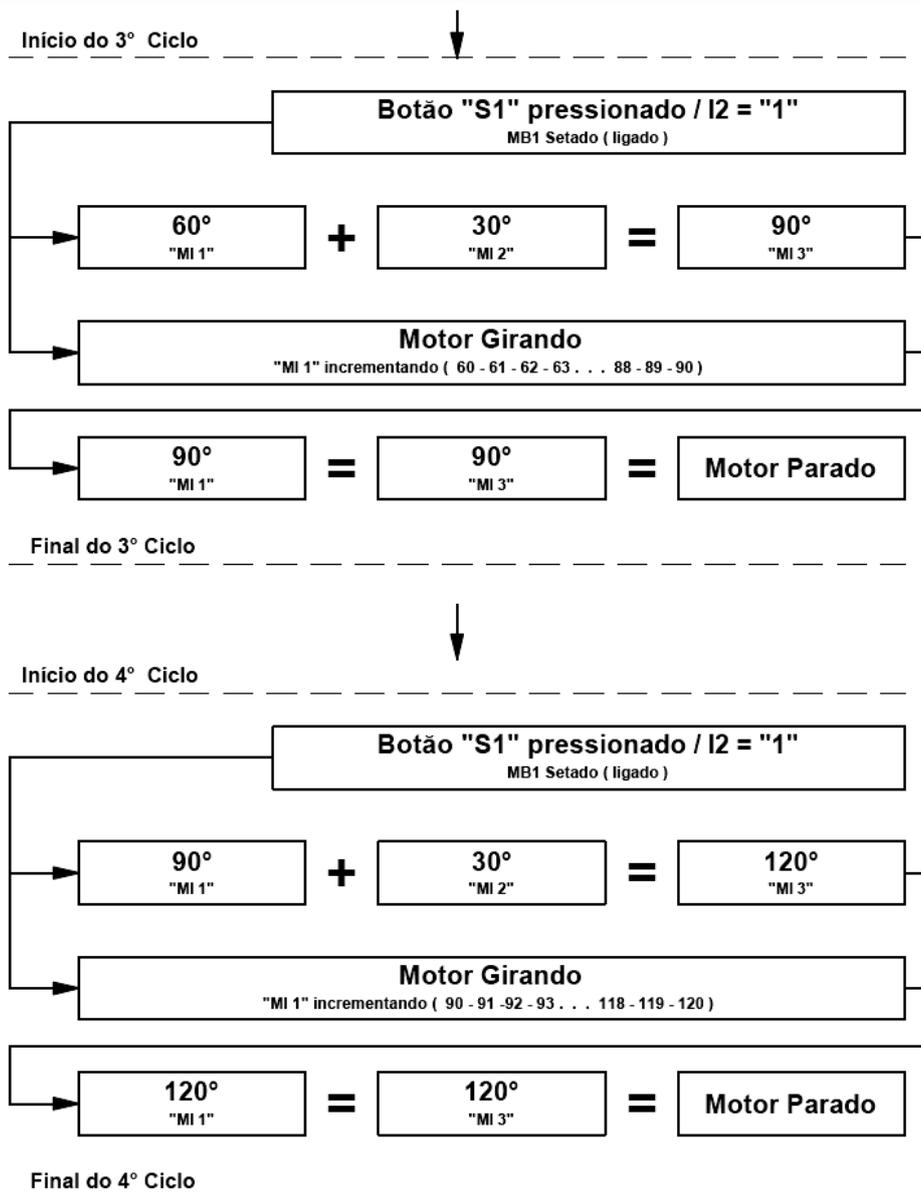


Figura 11.26

Simples, não? Com essa lógica pouco complexa, podemos comandar nossa mesa posicionadora de forma contínua em intervalos programados pelo operador e de acordo com a necessidade do processo. Fique atento porque, se em qualquer momento o operador acionar o botão "S3" (I3) referência de zero, o sistema assume a posição atual da mesa como referência zero, dando liberdade para que o operador possa interromper ou atualizar a seqüência operacional. Porém existe um detalhe não observado anteriormente na primeira linha do programa (figura 11.14), onde tratamos da inicialização do processo e definimos que o sistema assume a posição zero quando energizado, através se "SB 2" ou pela ação de "I3" (S3/referência zero), quando executarmos essa ação, o registrador de inteiro "MI 3" também deve receber o valor zero, caso contrário à posição de parada assume um valor errôneo, pois assumirá o valor adquirido no ciclo anterior ao referenciamento. Portanto devemos elaborar



também uma lógica idêntica à linha 1 (figura 11.14) para gravar o valor zero no registrador auxiliar “MI 3”, para que, a cada inicialização, ambos possam partir com os mesmos valores. Outro fator interessante é a utilização do bit auxiliar do motor em série com a lógica de referência, para que a ação de referência zero só seja válida se a mesa estiver parada, em outras palavras, o operador é impedido de referenciar o sistema com a mesa em movimento. Acompanhe a linha 7 do programa para identificar essas situações:

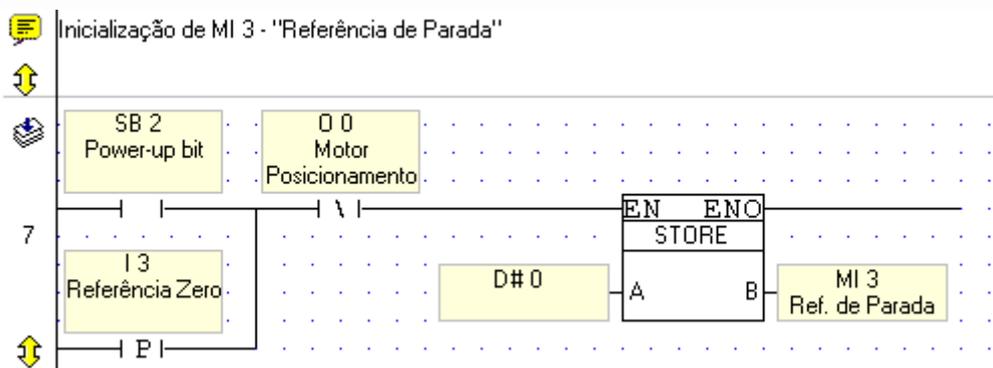


Figura 11.27

Na linha 7 do programa, podemos identificar que, se o bit auxiliar do sistema “SB 2” ou a entrada “I3” através do botão S3 forem verdadeiros, e o bit endereçado ao motor de posicionamento for falso (motor parado), a linha se torna verdadeira, atribuindo através do bloco de função “STORE” o valor da constante numérica zero (0) para o registrador “MI 3”. Essa condição é análoga à condição observada no início da aplicação (linha 1 do programa, figura 11.14).

Com todas as possibilidades avaliadas, só resta para o projetista elaborar a linha referente ao acionamento físico do motor. Como visto na figura 11.23, o bit auxiliar “MB 3” é responsável pelo acionamento da saída correspondente ao motor de posicionamento da mesa, atribuído ao endereço “O 0” e também a elaboração da lógica de acionamento manual do motor, através de “I4” botão S3, posicionamento manual. Porém devemos nos atentar para a seguinte situação: quando o operador acionar o modo de posicionamento manual, o motor entrará em movimento e poderá parar em qualquer posição, inclusive fora das posições pré-programadas pelo operador no registro do Set Point (MI 2), portanto, se pensarmos de forma “lógica”, poderemos identificar que, se houver necessidade de o operador movimentar a mesa de forma manual, é conveniente referenciar a mesa de posicionamento no momento exato em que o motor cessar o movimento, dessa forma, sempre que o operador movimentar a mesa posicionadora manualmente, o sistema estará sempre atribuindo a referência de posição zero assim que o operador soltar o botão de “posicionamento manual” (S3). Essa operação é muito simples, basta atribuir um contato normalmente fechado (NF) no modo de transição negativa (N) ao endereço correspondente à entrada “I4”, e toda vez que o operador pressionar o botão “S3” e ocorrer uma transição do estado verdadeiro, para o estado falso, os registradores “MI 1” e “MI 3” estarão recebendo o valor da constante zero, idêntico às lógicas observadas nas



figuras 11.14 e 11.27. Acompanhe esse procedimento na linha 8 do programa do usuário logo abaixo:

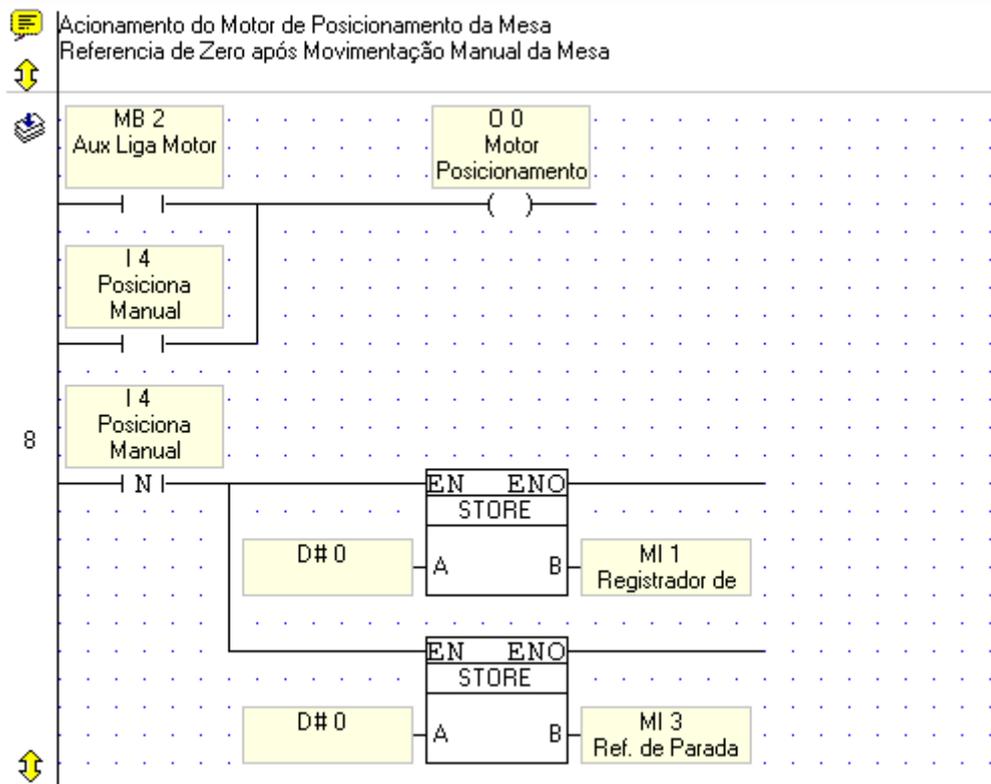


Figura 11.28

Com esse procedimento, encerramos a elaboração do programa LADDER de usuário para nosso sistema automatizado, porém ainda devemos elaborar uma interface gráfica para que o usuário possa inserir e monitorar as variáveis do processo, ou seja, a monitoração do valor instantâneo do registrador “MI 1” que representa a posição angular da mesa, e o valor do registrador “MI 2”, onde é inserido o valor de Set Point.

A ferramenta de programação *Visilogic*® para CLPs da família Vision proporciona essa funcionalidade, ou seja, existe um módulo no próprio programa em que o usuário pode desenvolver as telas de controle ou monitoramento conforme as necessidades do projeto elaborado. Nas telas gráficas, além de o usuário indicar textos e gráficos, também é possível a inserção de campos numéricos referenciados diretamente a qualquer registrador interno, em nosso exemplo “MI 1” e “MI 2”. A figura 11.29 logo abaixo ilustra a tela elaborada para esse projeto.



### Tela de IHM desenvolvida para Aplicação



Figura 11.29

Para melhor compreensão dos recursos de programação gráfica da ferramenta *Visilogic®*, consulte o guia de programação fornecido com o equipamento ou os documentos de apoio indicados para este material.

Dessa forma, concluímos o exemplo proposto para compreensão da interface elétrica utilizando um gerador de pulsos/encoder e um CLP. Vale lembrar que o exemplo foi elaborado com base em uma aplicação real e o desenvolvimento do projeto foi embasado no dispositivo proposto. Porém, para aplicações semelhantes, o usuário deve verificar a documentação técnica do CLP escolhido, bem como a compreensão da ferramenta de programação oferecida pelo fabricante.

