

“

ARQUITETURA DE COMPUTADORES | 21010 | 2025/2026

Período de realização: 19 de Dezembro de 2025 a 9 de Janeiro de 2026

Data Limite de entrega: 9 de Janeiro de 2026 até às 23:59 hora de Portugal Continental

Temática: Organização básica do computador

Objetivos:

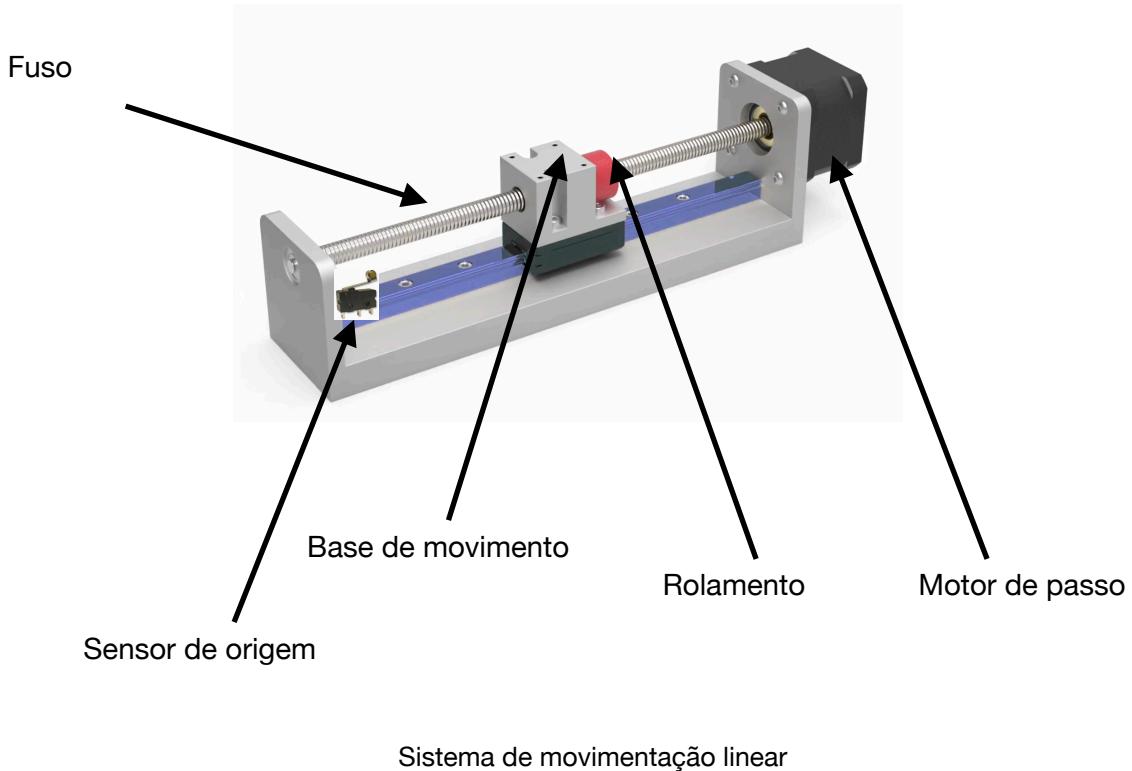
- . Arquitetura de um computador
- . Conjunto de instruções
- . Programação em linguagem Assembly P3

Trabalho a desenvolver

Implementação de um sistema de movimentação linear

Um motor de passo é um motor especial para a realização de movimentos precisos. Os motores de passo mais comuns, recebem dois sinais digitais, um para indicar a direção do movimento (clockwise ou counterclockwise) e outro que por causa impulso de 0 para 1 que recebe, faz o eixo do motor rodar 1.8°. Claro que uma volta completa, 360°, necessita de $360/1.8 = 200$ impulsos, também muitas vezes designados por passos.

Transformar o movimento rotacional do eixo do motor num movimento linear pode ser feito através de vários processos, mas vamos pensar na utilização de um fuso acoplado ao eixo do motor. Esse fuso é um varão com uma rosca em espiral e nessa rosca existe um mecanismo composto por um rolamento que está associado a uma base de movimento.



Ver vídeo explicativo em:

<https://www.youtube.com/watch?v=XZG2vp-egeg>

Vamos considerar que o fuso em causa, por cada volta completa que dá faz a base movimentar-se 10 milímetros e que o eixo tem o comprimento de 100 milímetros. A posição mais à esquerda do eixo terá a coordenada de 0 milímetros e a posição mais à direita do eixo terá a coordenada de 100 milímetros.

Na posição mais à esquerda existe um sensor que designamos por origem e devolve um sinal que se ativa quando a base de movimento atinge esse sensor.

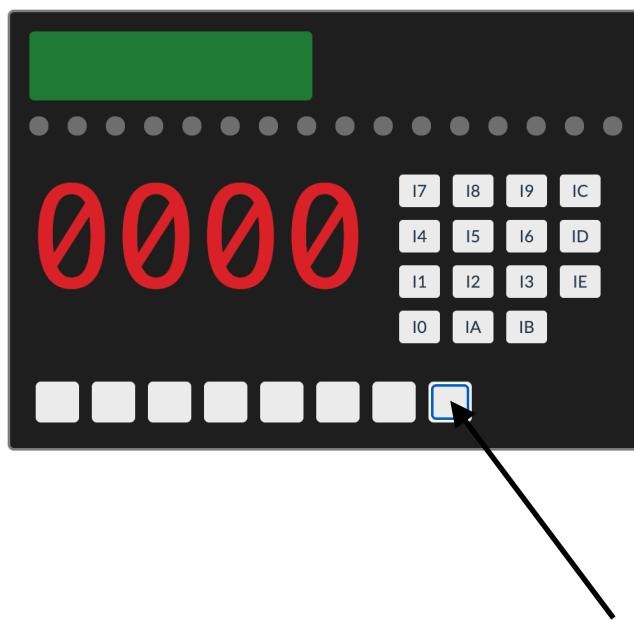
Vamos então construir um programa em Assembly P3, dividido por alíneas que permita o comando e controlo deste sistema de movimento linear constituído principalmente pelo motor, pelo fuso, pelo rolamento e pela base de movimento.

O seu programa deverá ser codificado a partir da posição 0000h de memória. No entanto as variáveis associadas ao programa deve residir a partir da posição 8000h de memória.

Apesar de não ser alvo de avaliação o sistema de entradas/saídas da placa do P3, pode usar, no simulador um botão para simular o sensor de origem e dois indicadores luminosos, um para verificar o sinal de direção do motor e outro para verificar o sinal dos impulsos. No entanto, as funções para usar o módulo de entradas/saídas são fornecidas neste enunciado e portanto se quiser, pode não realizar a simulação no módulo de entradas/saídas, mas seria melhor se o fizesse.

Entradas de informação:

Usar a rotina **LE_SENSOR** que devolve o valor 1 se o sensor de origem se ativar. Podemos simular ativando o botão indicado durante a execução do programa.



Premir este botão, ficará azul para simular a ativação do sensor de origem.

LE_SENSOR: MOV R1, M[FFF9h]
 AND R1,1
 RET

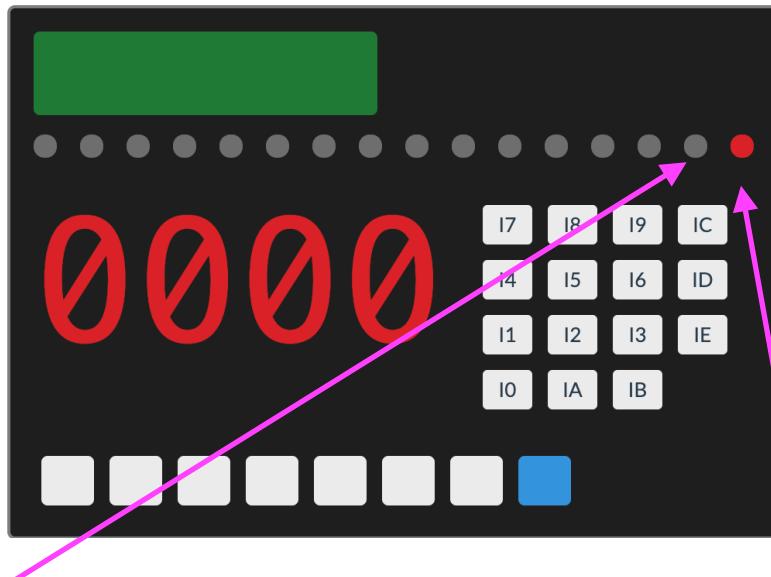
8000h: Valor da posição atual do sistema em milímetros.

8001h: Valor para onde deslocar o sistema em milímetros.

Saídas de informação:

Usar a rotina **SET_DIRECAO** que ativará o sinal de direção do motor. Esta rotina recebe em R1 o valor da direção e mostra num indicador o seu valor (ver figura em baixo). Além disso, esta rotina escreve no endereço **8002h** o valor dado para a direção para posterior consulta de outras rotinas. Como gerar o sinal de sentido de movimento para o motor? Se se pretender movimento para a esquerda, a posição 8002h tem de estar a 0, caso contrário tem de estar a 1.

SET_DIRECAO: MOV M[FFF8h],R1
 MOV M[_SENTIDO_MOVIMENTO],R1
 RET



Indicador de impulso do motor. Ligado significa 1, desligado significa 0. Quando se está a dar impulsos ao motor, este indicador deve piscar

Indicador de direção do motor. Ligado significa 1 ou movimento à direita, desligado significa 0 ou movimento à esquerda

Como gerar impulsos ou passos para a movimentação do motor? Usar a rotina **IMPULSO** que ativará a saída de impulsos durante um determinado tempo. Vamos simular no segundo indicador luminoso como se mostra na figura anterior.

IMPULSO:

```

PUSH R1
MOV R1, M[_SENTIDO_MOVIMENTO]
AND R1,1
MOV M[FFF8h],R1
OR R1,2
MOV M[FFF8h],R1
CALL DELAY
AND R1,1
MOV M[FFF8h],R1
CALL DELAY
POP R1
RET

```

DELAY:	PUSH R7
	MOV R7,0FFFh
CICLO_DELAY:	CMP R7,0
	BR.Z FIM_DELAY
	DEC R7
	BR CICLO_DELAY
FIM_DELAY:	POP R7
	RET

Pensando que o eixo tem 100 milímetros, e que o fuso dá uma volta completa de 360 graus para fazer deslocar a base de movimento 10 milímetros, o número máximo de impulsos que se podem dar ao motor será de:

1.8 graus por impulso, o que significa que para uma volta completa temos de dar 200 impulsos, visto que $360 \text{ graus} / 1.8 = 200$.

Por cada volta completa o eixo desloca-se 10 milímetros, pelo que para se deslocar 100 milímetros temos de dar 10 voltas completas, 2000 impulsos.

- a) **[1 valor]** Quando se inicia o sistema, não se sabe em que posição está o eixo. Para isso serve o sensor de origem. Faça uma rotina em Assembly P3 que primeiramente faça o sentido de rotação ser para a esquerda (use a rotina **SET_DIRECAO**) e depois, ciclicamente vá gerando impulsos (usar a rotina **IMPULSO**), e testando o sensor de origem (usar a rotina **LE_SENSOR**). Quando este se ativar, a rotina deve parar de gerar impulsos e deve atualizar a posição atual do eixo em 0 milímetros.
- b) **[1 valor]** Faça uma rotina em Assembly P3 que receba em R1 o valor da posição atual do eixo em milímetros, em R2 o valor em milímetros da posição para onde se quer deslocar o eixo, e devolva em R3 o valor 1 se o movimento for possível ou 0 se o movimento não for possível. Não esquecer os limites do eixo [0..100] milímetros.
- c) **[1 valor]** Faça uma rotina em Assembly P3 que receba em R1 o valor atual da posição do eixo em milímetros, em R2 o valor em milímetros da posição final pretendida do movimento do eixo, e se o movimento for possível de ser executado, devolver em R3 o número de impulsos necessários para o movimento. Em R4 deve devolver o valor do sentido do movimento, 0 para esquerda, 1 para a direita.
- d) **[1 valor]** Faça um programa em Assembly P3 que inicie o sistema do eixo. Depois, move o eixo para as seguintes posições absolutas:

- 30 mil
- 10 mil
- 50 mil

Nota: Antes de qualquer movimento deve usar a rotina SET_DIRECAO para definir o sentido do mesmo.

Pode verificar o vídeo de simulação do EFólio B em :

https://youtu.be/lbHy_ikZr6c

Regras a observar no efólio B

Relatório: O relatório deve de apresentar todas as decisões relevantes e todas as opções tomadas na construção das alíneas e programas. Deve também mostrar evidências do funcionamento do seu trabalho.

Formato de entrega: Um único ficheiro em formato ZIP com o nome correspondente ao número do aluno (ex: 999999EFolioB.zip). Este ficheiro deve conter os ficheiros do relatório (em formato pdf) e os ficheiros ou ficheiro com o código do seu trabalho. O código deve estar em formato que seja possível copiar para testes.

Não são aceites entregas fora da plataforma Moodle.

Avaliação:

Critérios de correção:

- . Funcionalidade testada, implementada e testada no relatório (50%):
 - Alínea a) 15%
 - Alínea b) 15%
 - Alínea c) 10%
 - Alínea d) 10%
- . Simplicidade e modularidade da solução (utilização correta de rotinas e código simples): 10%
- . Eficiência (contagem do número de instruções e ciclos de relógio relativamente a um valor de referência excepto alínea a): 10%
- . Apresentação do código (ideação e comentários): 10%
- . Relatório: 20%
 - Justificação das opções
 - Legibilidade

Penalizações:

- . Trabalhos não conforme as regras de entrega do E-Fólio B: até 10%
- . Trabalhos que não compilem corretamente: 100%
- . Código sem comentários ou apenas comentários a descrever instruções (ex: MOV R2,R1 ; mover para R2 o valor de R1): até 50%
- . Deteção de qualquer tipo de fraude: 100%

- . Trabalhos entregues após data limite dentro de um dia de atraso: 10%
- . Trabalhos entregues após data limite com mais de um dia de atraso: 100%