

Motor de passo controlado por Arduíno

Física Experimental IV - 2º semestre de 2018

1 Características

- O programa começa ligando o motor
- O programa deve comandar um motor de passos que faz um suporte girar em passos de 4° , de zero a 180° com intervalo de 0,5 s.
- Chegando ao fim do percurso, o motor deve retornar à posição inicial e aí permanecer até que novo comando seja dado. Sugere-se que ele volte à posição inicial em mais ou menos 20 s.

2 Os comandos do motor

- `#include <Stepper.h>`
 - Biblioteca do motor de passos
- `const int stepsPerRevolution = 2048;`
 - Define do número de passos por volta
- `Stepper myStepper(stepsPerRevolution,8,10,9,11);`
 - Ligação nas entradas digitais do Arduíno nano
- `myStepper.setSpeed(10);`
 - Define velocidade de rotação em rpm
- `myStepper.step(npassos);`
 - Faz com que ele gire n passos

3 Comando da velocidade da porta serial

- `Serial.begin(9600);`
 - 9600 é o número de bits por segundo transmitidos

4 Comandos do sensor

- `sensorValue0 = analogRead(A1);`
 - Enviando para a entrada analógica A1. O valor dessa leitura estará entre 0 e 1023. Se quisermos transformar isso em uma tensão que varia entre 0 e 5V, deve ser feito o cálculo pertinente

5 Botão liga/desliga

- `int button;`
 - `button` é uma variável inteira
- `button = LOW;`
 - Desligado
- `button = HIGH;`
 - Ligado
- `button = digitalRead(5);`
 - Informação conectada na entrada digital 5 do Arduino nano

6 Planilha de visualização online

- `Serial.println("CLEAR SHEET");`
 - Excel - auto explicativo
- `Serial.println("CLEAR DATA");`
 - Excel - auto explicativo
- `Serial.println("LABEL,Angle (graus),Voltage (V)");`
 - Excel - Labels de 2 colunas da tabela
- `Serial.print("DATA,");`
 - Excel - aviso da maneira como serão escritos os dados
- `Serial.print(teta);`

- `Serial.print(",");`
- `Serial.println(voltage,dig);`
 - As 3 linhas acima explicam que as 2 variáveis são separadas por vírgula e que ocorre mudança de linha após escrevê-las