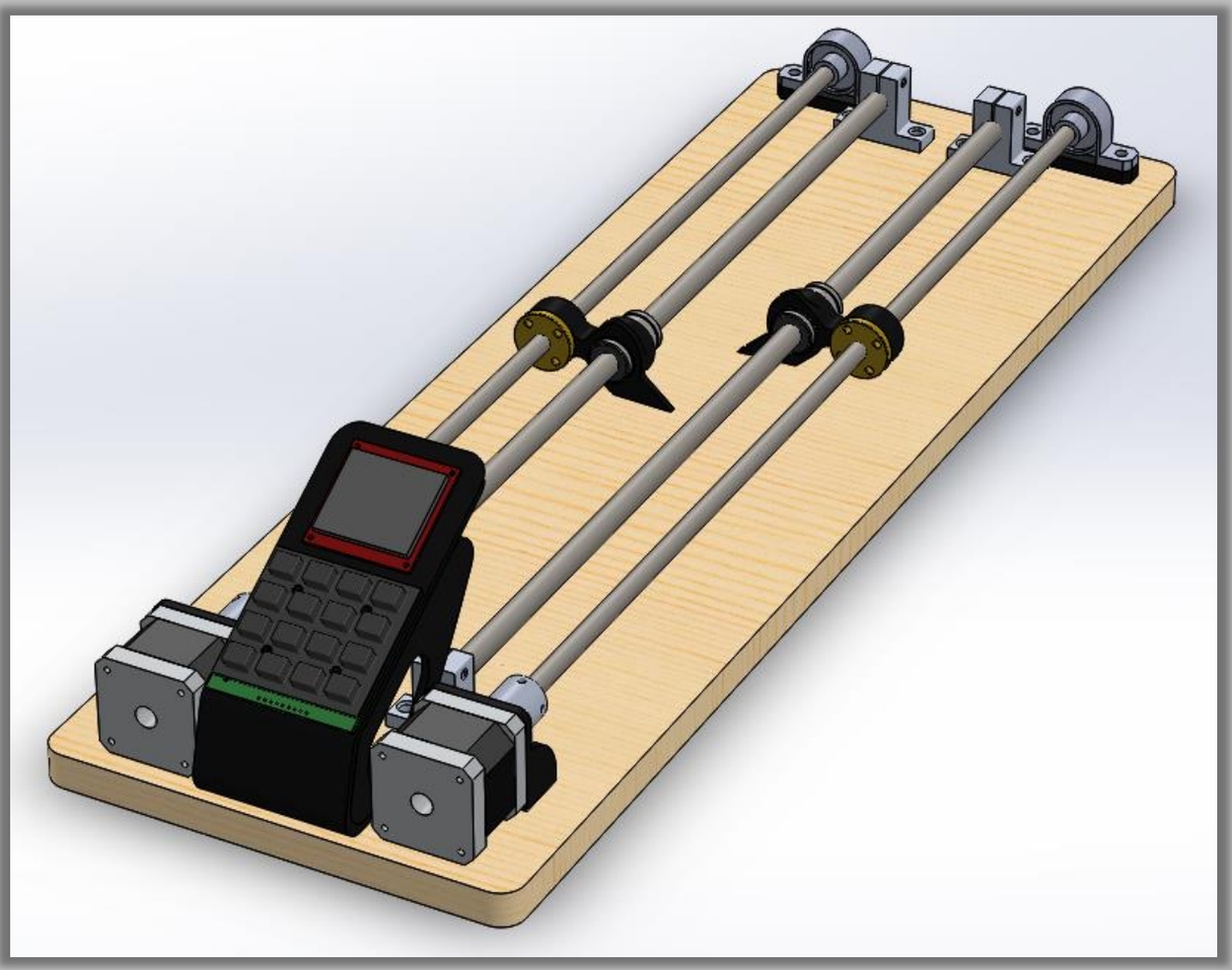
Elementos de máquinas - Fusos

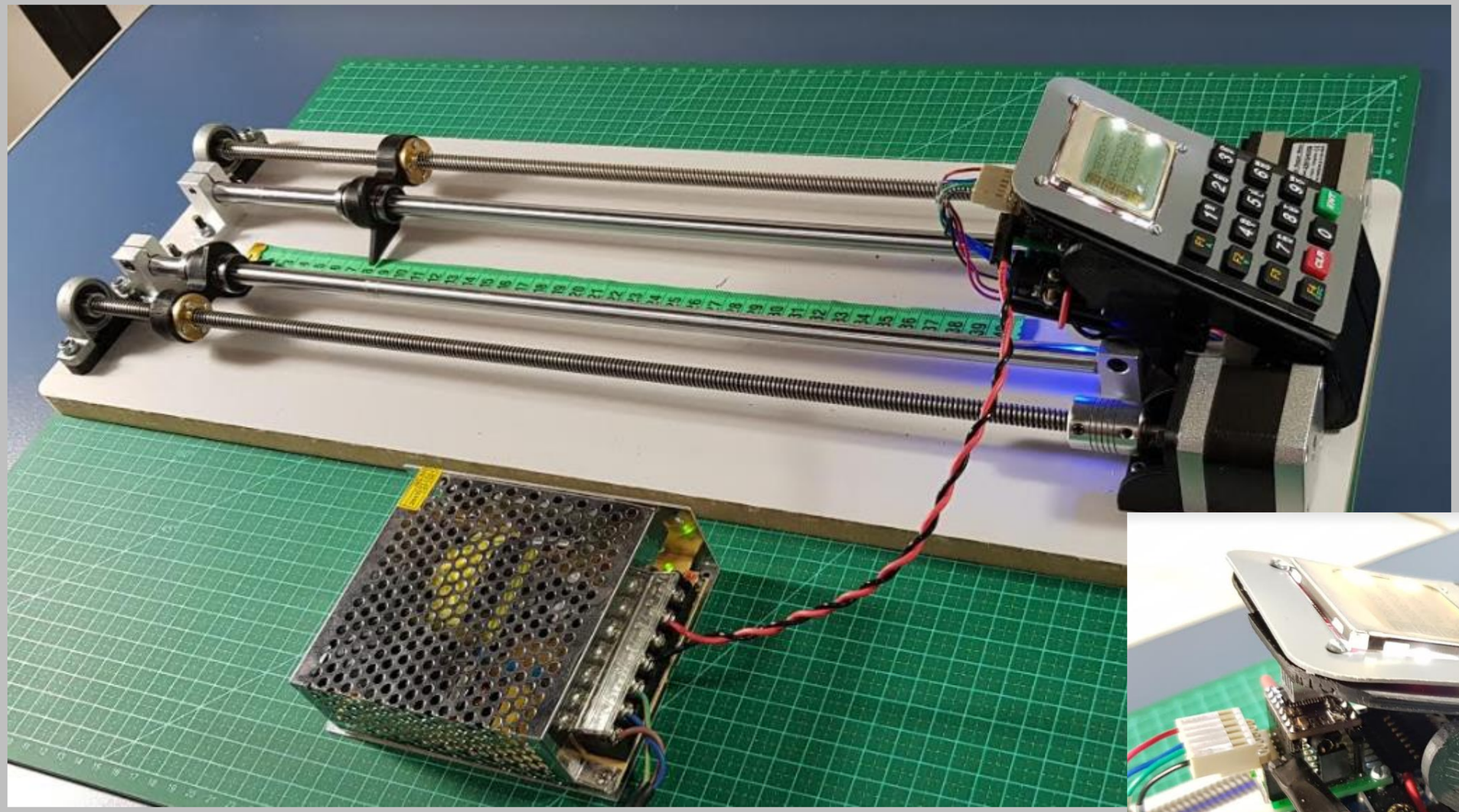


Por Fernando Koyanagi

Recursos usados

- Fuso trapezoidal de 8mm de diâmetro e 2mm de passo
- Fuso trapezoidal de 8mm de diâmetro e 8mm de passo
- Castanha flangeada para fuso 8x2
- Castanha flangeada para fuso 8x8
- Mancais para fusos de 8mm de diâmetro
- Guia linear cilíndrica 10mm de diâmetro
- Rolamentos cilíndricos para guias 10mm
- Suportes para guias cilíndricas de 10mm
- Motores NEMA 17
- Acopladores de eixo
- Arduino Uno
- Driver DRV8825
- Teclado matricial 4x4
- Display Nokia 5110
- Peças plásticas diversas
- Parafusos e porcas
- Base de madeira
- Fonte externa de alimentação 12V

Montagem (foto ou vídeo da montagem)



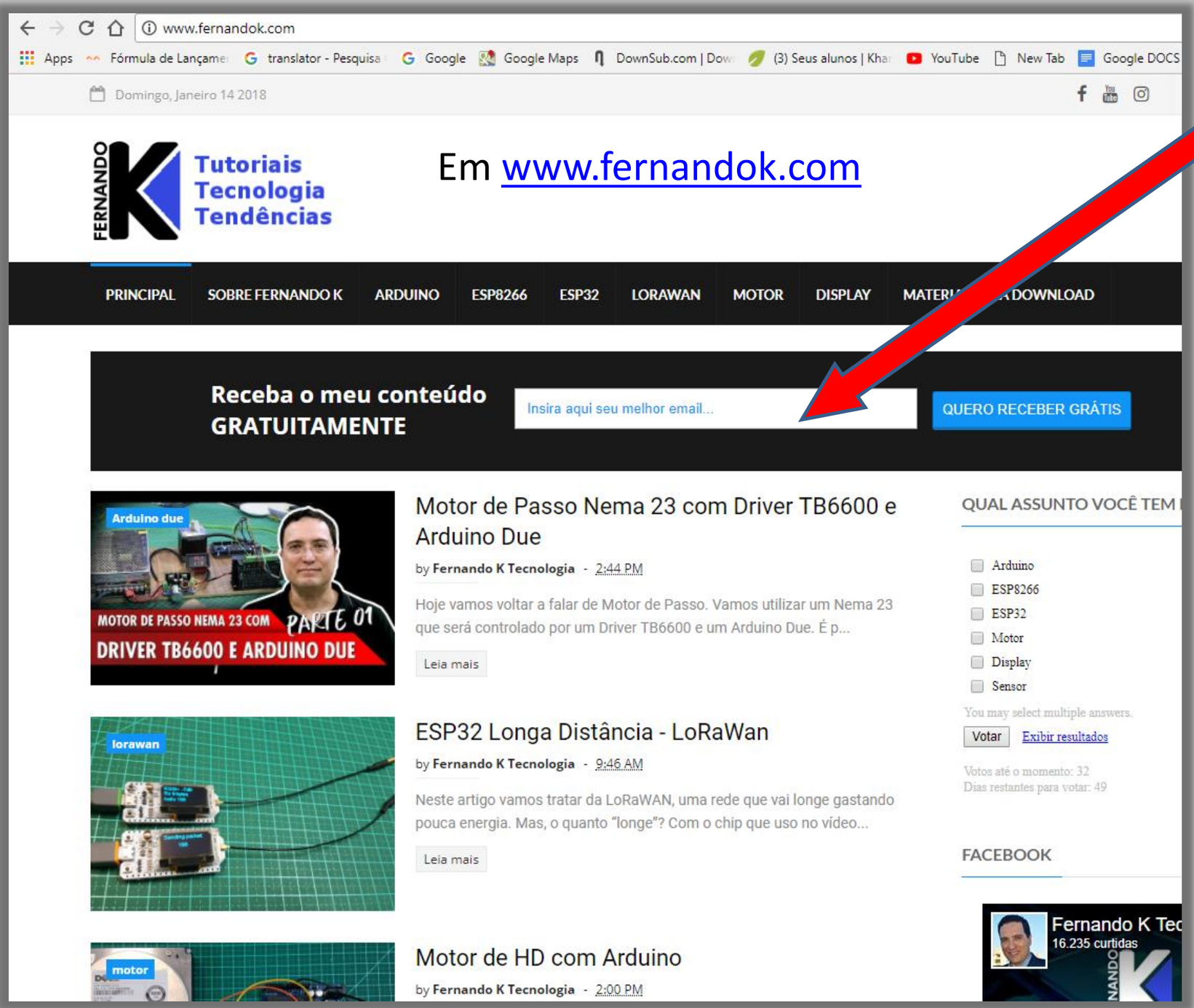
Intenção dessa aula

1. Apresentar algumas características interessantes e aplicações dos fusos.

2. Demonstrar algumas formas de calcular o movimento provocado por um fuso.
3. Apresentar um montagem de teste de fusos.

Sobre fusos – O que são?

- Os fusos são elementos de máquinas (como os parafusos).
- São barras retas formadas por roscas de passos contínuos.
- São utilizados em mecanismos que exigem movimento linear e posicionamento.
- Podem exercer altas forças de tração e compressão e transmitem torque.
- Permitem movimentação com travamento automático.
- Podem ser construídos de diversos materiais, sendo os mais comuns alumínio e aço.


www.fernandok.com
Apps
Fórmula de Lançame
translator - Pesquisa
Google
Google Maps
DownSub.com | Dow
(3) Seus alunos | Kha
YouTube
New Tab
Google DOCS
Domingo, Janeiro 14 2018
FERNANDO K
Tutoriais
Tecnologia
Tendências
Em www.fernandok.com
PRINCIPAL
SOBRE FERNANDO K
ARDUINO
ESP8266
ESP32
LORAWAN
MOTOR
DISPLAY
DOWNLOAD
Receba o meu conteúdo GRATUITAMENTE
Insira aqui seu melhor email...
QUERO RECEBER GRÁTIS
Arduino due
MOTOR DE PASSO NEMA 23 COM DRIVER TB6600 E ARDUINO DUE
PARTE 01
Motor de Passo Nema 23 com Driver TB6600 e Arduino Due
by Fernando K Tecnologia - 2:44 PM
Hoje vamos voltar a falar de Motor de Passo. Vamos utilizar um Nema 23 que será controlado por um Driver TB6600 e um Arduino Due. É p...
Leia mais
lorawan
ESP32 Longa Distância - LoRaWan
by Fernando K Tecnologia - 9:46 AM
Neste artigo vamos tratar da LoRaWAN, uma rede que vai longe gastando pouca energia. Mas, o quanto "longe"? Com o chip que uso no vídeo...
Leia mais
motor
Motor de HD com Arduino
by Fernando K Tecnologia - 2:00 PM
QUAL ASSUNTO VOCÊ TEM
Arduino
ESP8266
ESP32
Motor
Display
Sensor
You may select multiple answers.
Votar
Exibir resultados
Votos até o momento: 32
Dias restantes para votar: 49
FACEBOOK
Fernando K Tec
16.235 curtidas

Seu e-mail

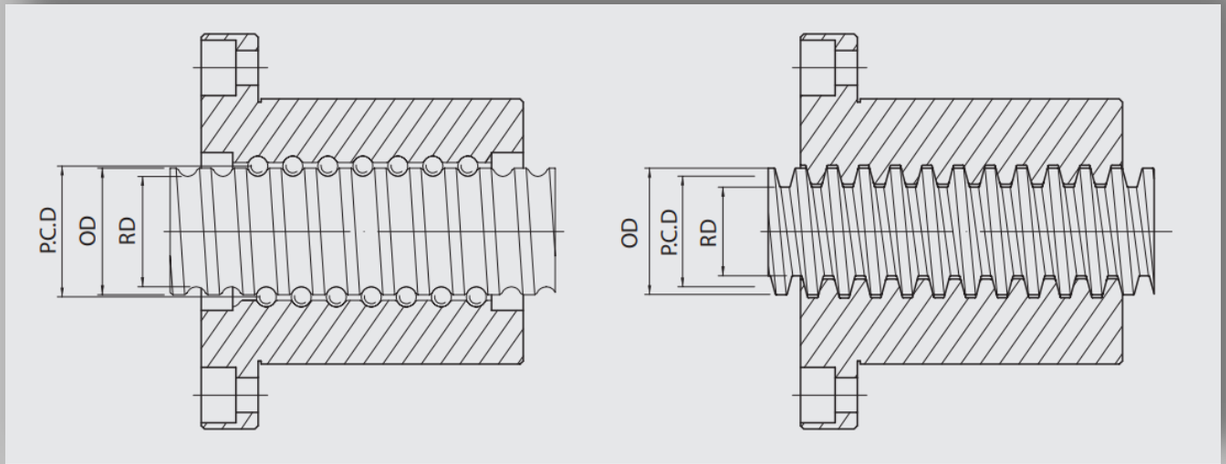
Sobre fusos – Roscas simples e de esferas

- **Os fusos de esferas:**

- **Possuem canais semicirculares onde as esferas rolam.**
- **São relativamente mais caros.**
- **Possuem baixa fricção se comparados ao fusos de rosca simples, levando a um rendimento muito superior (atrito de rolamento).**

- **Os fusos de roscas simples:**

- **possuem normalmente perfis trapezoidais, por ser esta geometria ser mais adequada a aplicação de forças no sentido axial e transmissão suave de movimento.**
- **São relativamente baratos.**
- **Possuem alta fricção se comparados aos fusos de esferas recirculantes, levando a um baixo rendimento. (atrito de escorregamento)**



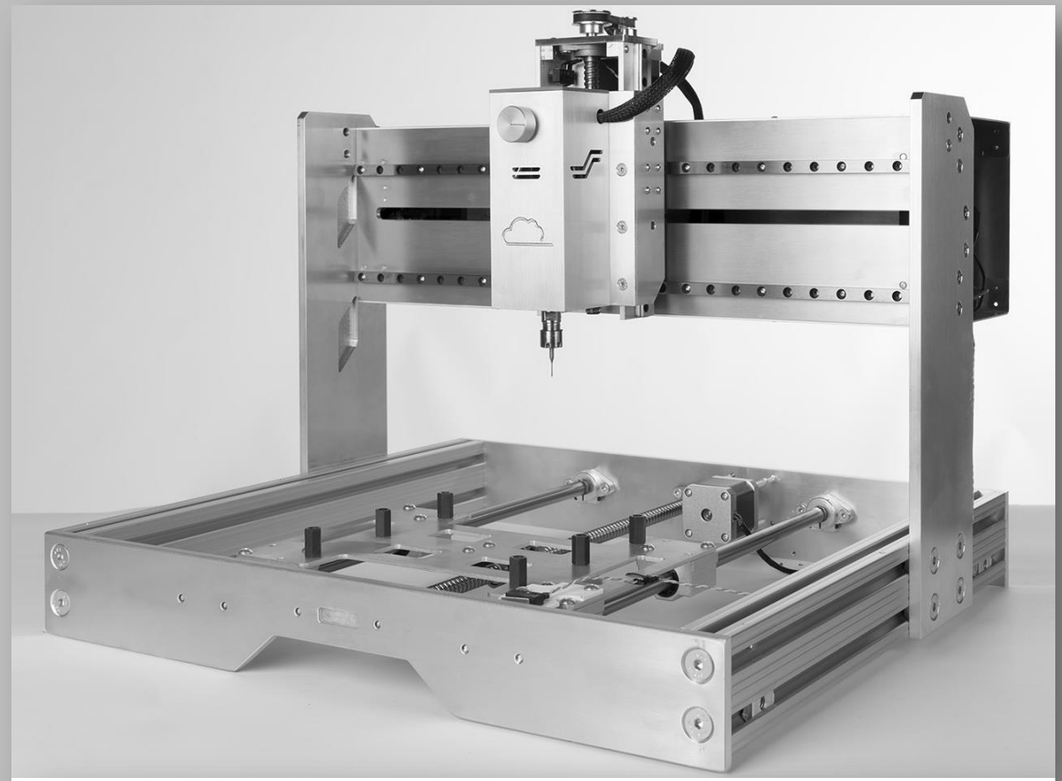
Fonte – catálogo Hiwin

Sobre fusos - Aplicações

- Os fusos podem ser aplicados em qualquer mecanismo onde haja a necessidade de movimento linear. São amplamente utilizados na indústria em maquinário e processos.

- Algumas aplicações incluem:

- Elevadores de carga
- Prensas
- Fresas e tornos
- Equipamentos CNC
- Embaladoras
- Impressoras 3D
- Equipamentos de corte e gravação a laser
- Processos industriais
- Sistemas de posicionamento e movimentação linear



Sobre fusos - Parâmetros

- Existem diversas características de um fuso que devem ser levadas em consideração ao se projetar um mecanismo.

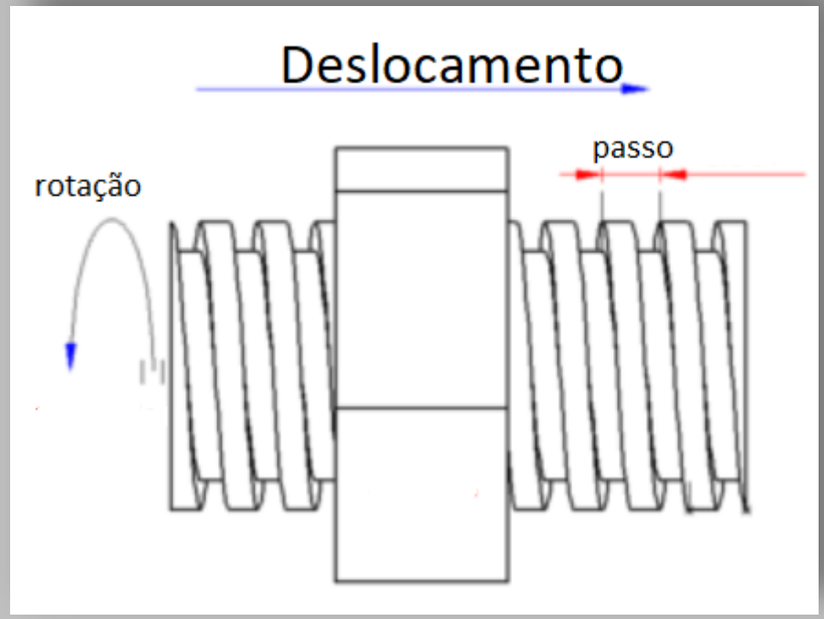
- Além do seu diâmetro e passo, é necessário reconhecer sua resistência a compressão, seu momento de inércia (resistência a alteração de seu estado de rotação), material construtivo, a velocidade de rotação a qual será submetido, direção de operação (horizontal ou vertical), a carga aplicada, entre outras.

- Mas baseando-se em mecanismos já construídos, podemos intuir vários destes parâmetros.

- Vamos reconhecer alguns bem comuns. Comecemos pelo PASSO.

Sobre fusos – Passo (deslocamento e velocidade)

- Determina o comprimento percorrido pela castanha a cada revolução.
- Dado normalmente em mm/revolução.
- Um fuso de 2mm por revolução provocará um deslocamento de 2mm a cada volta que o fuso executar.
- Influenciará na velocidade linear da castanha, uma vez que, com o aumento da velocidade de rotação, o número de revoluções por unidade de tempo aumentará e consequentemente a distância percorrida também.
- Se um fuso de 2mm por revolução girar a 60 rpm (um volta por segundo), a castanha se movimentará a 2mm por segundo.



Sobre fusos – Fusos e motores de passo

- Para demonstrar o efeito do passo de um fuso, utilizaremos uma montagem com dois fusos (um de 2mm/rev e um de 8mm/rev).

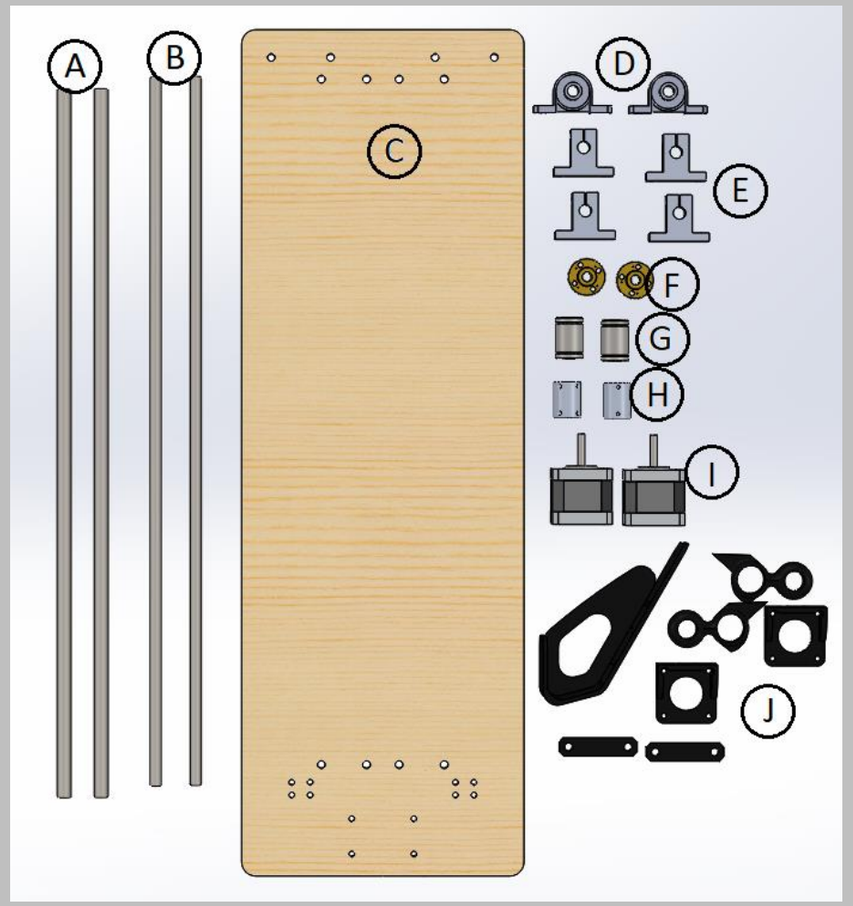
- Para ter um controle mais preciso do movimento e demonstrar um aplicação bastante comum, acoplaremos estes fusos a dois motores de passo idênticos, ligados em paralelo ao mesmo drive (logo, recebendo sinais iguais).

- Os motores são de 200 passos por volta e serão acionados em full-step. Logo, a cada 200 pulsos os motores executarão uma revolução.

- Assim fica bastante claro perceber que:
  - Para 200 pulsos aplicados, o fuso de 2mm/rev moverá sua castanha por 2mm.
  - Para 200 pulsos aplicados, o fuso de 8mm/rev moverá sua castanha por 8mm.

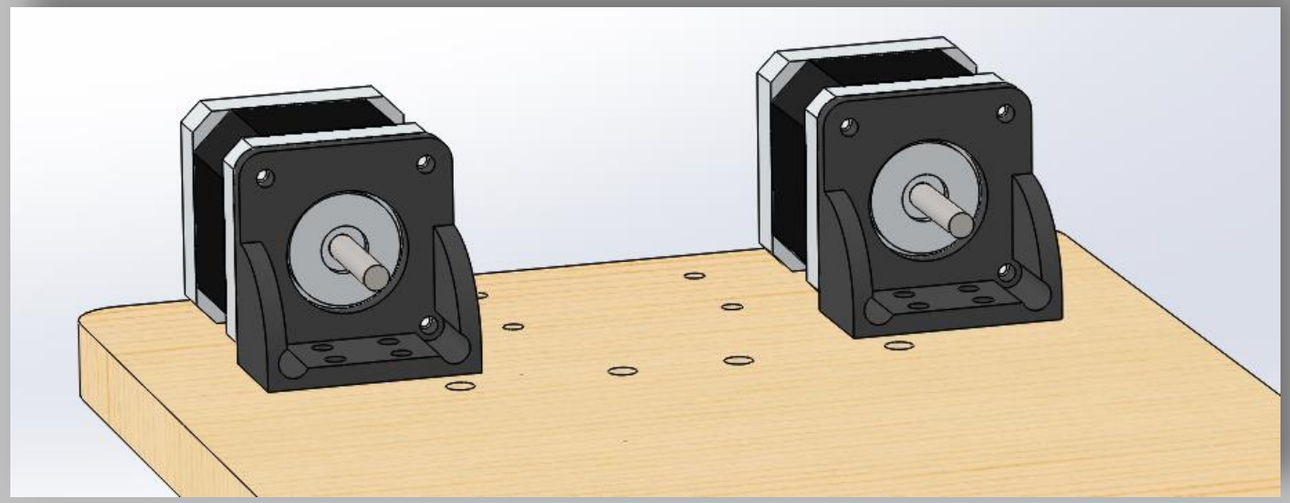
  Para dois mil pulsos aplicados, os movimentos serão respectivamente, 20mm e 80mm.

Montagem – materiais



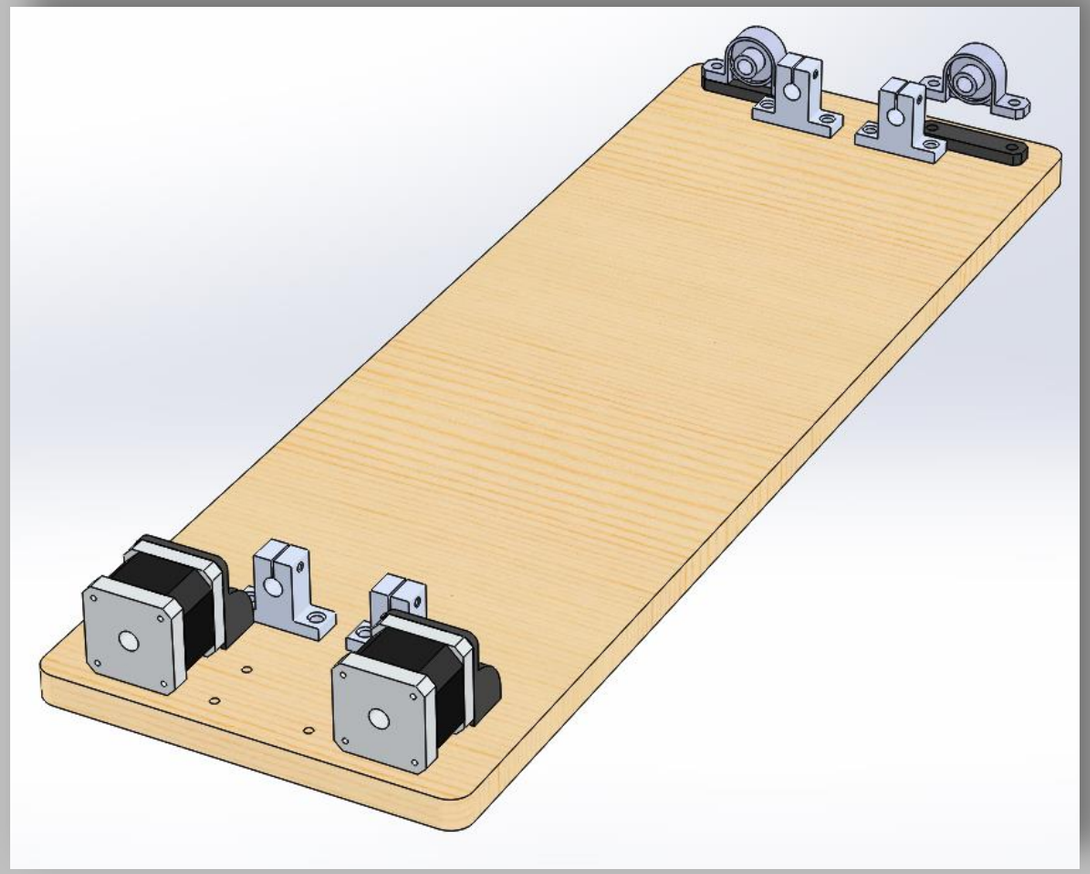
- Guias lineares de 10mm (A)
- Fusos trapezoidais de passos 2 e 8mm (B)
- Base com furação (C)
- Mancais para os fusos (D)
- Suportes das guias (E)
- Castanhas dos fusos (F)
- Rolamentos(G)
- Acopladores (H)
- Motores (I)
- Peças de plásticos diversas (cursores, suportes dos motores, calços, suporte de teclado e display (J)

Montagem – passo 01



- Seguindo a furação da base (C), montamos os dois motores (I). Para prendê-los, usamos suportes feitos na impressora 3D (J). Não apertamos nenhum dos parafusos nesta etapa de posicionamento. Isso permitirá os ajustes necessários na etapa de alinhamento.

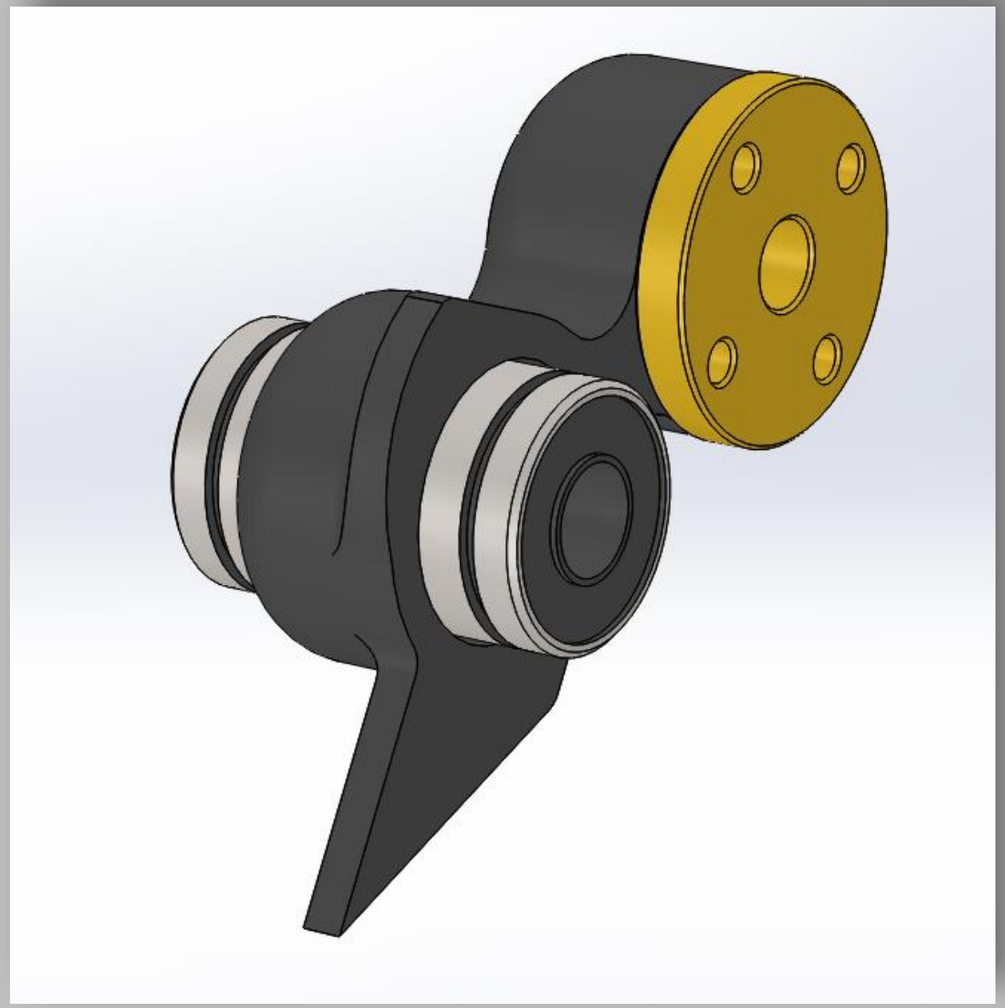
Montagem – passo 02



- Ainda seguindo a furação da base (C), posicionamos os suportes das guias (E) e os mancais (D).

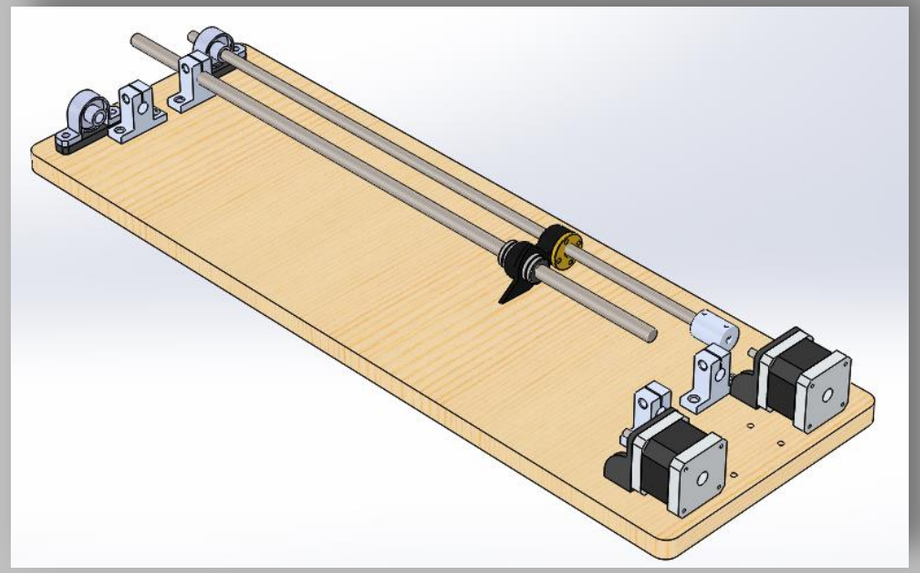
- Detalhe para o calço plástico (J) utilizado para ajustar as alturas dos mancais.

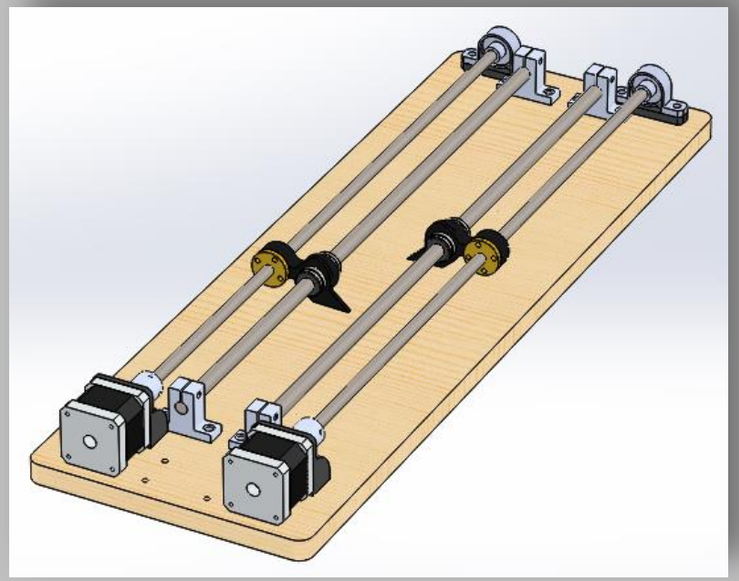
Montagem – passo 03



- Criamos um cursor usando um peça impressa para conectar o rolamento (G) à castanha (F). Usamos dois cursores, um direito outro esquerdo. Sua função é indicar a posição em uma escala sempre que quisermos determinar o deslocamento causado pelo fuso.

Montagem – passo 04



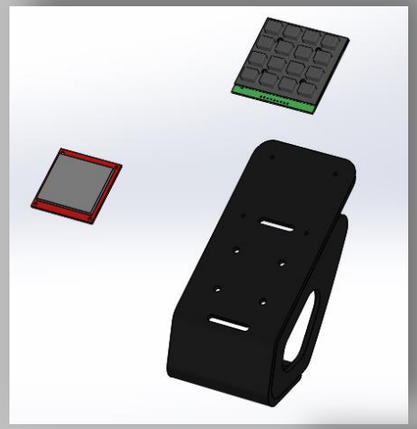


- Inserimos a guia (A) e o fuso (B) nos seus respectivos mancal (D) e suporte (E), pelo lado oposto ao motor, em seguida, inserimos a guia e o fuso no rolamento (G) e castanha (F) e na ponta do fuso inserimos também o acoplador (H). Levamos os dois até atingirem seus pontos finais (suporte oposto e motor).

- Apertamos levemente os parafusos para permitir um posterior ajuste.

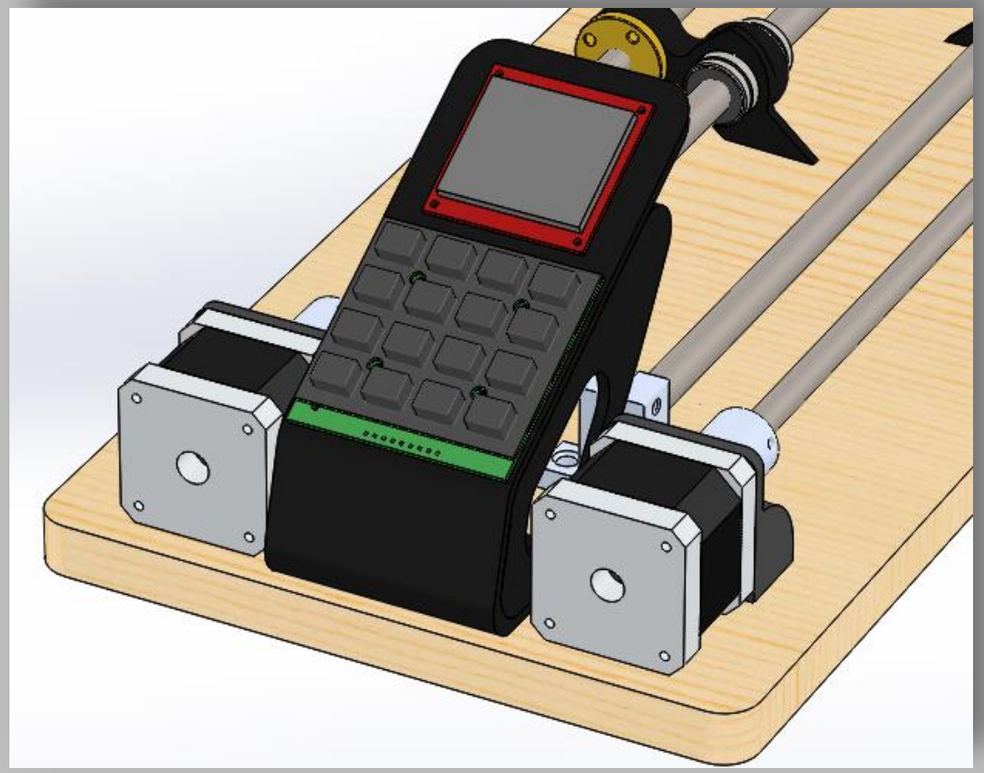
- Repetimos o procedimento usando a guia e fuso restantes.
- Com todos os componentes posicionados, efetuamos o alinhamento das partes, finalizando a etapa de montagem mecânica.

Montagem – eletrônica



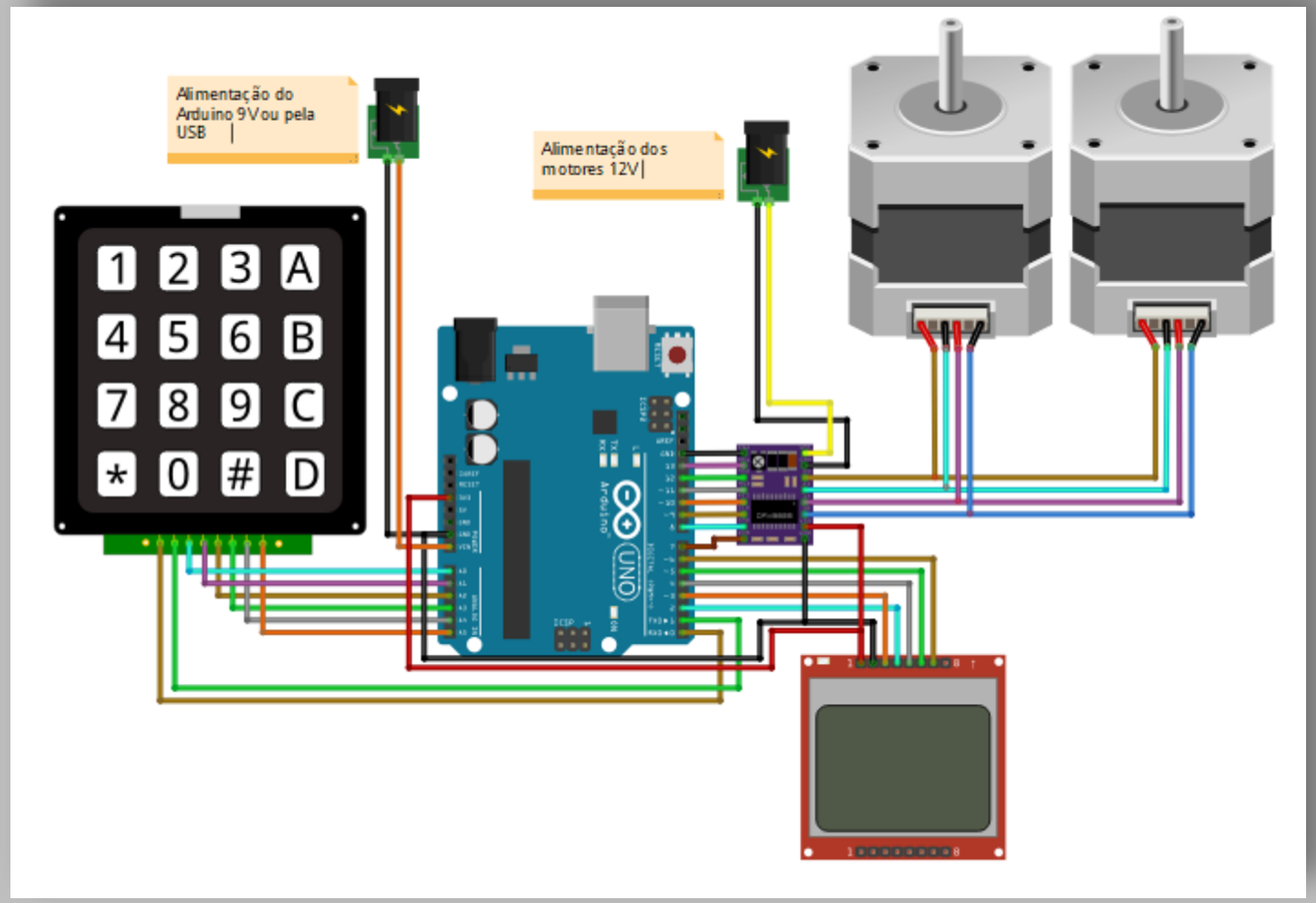
- Usando um suporte plástico impresso, fixamos o display Nokia 5110 e um teclado matricial 4x4.
- No espaço inferior do suporte residirá o Arduino Uno, o driver drv8825.

Montagem – eletrônica



- Usando a furação disponível na base, prendemos o conjunto.

Esquema elétrico



Código-fonte – Inclusão de bibliotecas e criação de objetos

```
1  //Biblioteca responsável por capturar a tecla que foi pressionada no teclado
2  #include <Keypad.h>
3  //Biblioteca responsável pelos graficos do display
4  #include <Adafruit_GFX.h>
5  //Biblioteca responsável pela comunicacao do display
6  #include <Adafruit_PCD8544.h>
7
8  //Configuracao de pinos do Display
9  // pin 6 - Serial clock out (SCLK)
10 // pin 5 - Serial data out (DIN)
11 // pin 4 - Data/Command select (D/C)
12 // pin 3 - LCD chip select (CS/CE)
13 // pin 2 - LCD reset (RST)
14 Adafruit_PCD8544 display = Adafruit_PCD8544(6, 5, 4, 3, 2);
15
16 //Biblioteca de motor de passo
17 #include <StepDriver.h>
18
19 //Instancia o driver DRV8825
20 DRV8825 d1;
```

Código-fonte – Constantes e variáveis globais

```
22  const byte LINHAS = 4; //número de linhas do teclado
23  const byte COLUNAS = 4; //número de colunas do teclado
24
25  //define uma matriz com os símbolos que deseja ser lido do teclado
26  char SIMBOLOS[LINHAS][COLUNAS] = {
27    {'A', '1', '2', '3'},
28    {'B', '4', '5', '6'},
29    {'C', '7', '8', '9'},
30    {'D', 'c', '0', 'e'}
31  };
32
33  byte PINOS_LINHA[LINHAS] = {A2, A3, A4, A5}; //pinos que indicam as linhas do teclado
34  byte PINOS_COLUNA[COLUNAS] = {0, 1, A0, A1}; //pinos que indicam as colunas do teclado
35
36
37  //instancia de Keypad, responsável por capturar a tecla pressionada
38  Keypad customKeypad = Keypad( makeKeymap(SIMBOLOS), PINOS_LINHA, PINOS_COLUNA, LINHAS, COLUNAS);
39
40  //variáveis resposnsáveis por armazenar o valor digitado
41  char customKey;
42  unsigned long distancia = 0;
43  unsigned long velocidade = 2000;
44
```

Código-fonte – Função de leitura do teclado

```
45  //Funcao responsavel por ler o valor do usuario pelo teclado------
46  unsigned long lerValor() {
47    //Escreve o submenu que coleta os valores no display
48    display.clearDisplay();
49    display.fillRect(0, 0, 84, 11, 2);
50    display.setCursor(27, 2);
51    display.setTextColor(WHITE);
52    display.print("VALOR");
53    display.setTextColor(BLACK);
54
55    display.fillRect(0, 24, 21, 11, 2);
56    display.setCursor(2, 26);
57    display.setTextColor(WHITE);
58    display.print("CLR");
59    display.setTextColor(BLACK);
60    display.setCursor(23, 26);
61    display.print("LIMPAR");
62
63    display.fillRect(0, 36, 21, 11, 2);
64    display.setCursor(5, 38);
65    display.setTextColor(WHITE);
66    display.print("F4");
67    display.setTextColor(BLACK);
68    display.setCursor(23, 38);
69    display.print("VOLTAR");
70    display.setCursor(2, 14);
71    display.display();
72    String valor = "";
73    char tecla = false;
```

Código-fonte – looping aguardando tecla pressionada

```
  //Loop infinito enquanto nao chamar o return
  while (1) {
    tecla = customKeypad.getKey();
    if (tecla) {
      switch (tecla) {
        //Se teclas de 0 a 9 forem pressionadas
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
        case '0':
          valor += tecla;
          display.print(tecla);
          display.display();
          break;
        //Se tecla CLR foi pressionada
        case 'c':
          //Limpa a string valor
          valor = "";
          //Apaga o valor do display
          display.fillRect(2, 14, 84, 8, 0);
          display.setCursor(2, 14);
          display.display();
          break;

        //Se tecla ENT foi pressionada
        case 'e':
          //Retorna o valor
          return valor.toInt();
          break;

        //Se tecla F4 (ESC) foi pressionada
        case 'D':
          return -1;
        default:
          break;
      }
    }
    //Limpa o char tecla
    tecla = false;
  }
}

```

Código-fonte – Função de movimentação do motor

```
123  //Funcao responsavel por mover o motor------------
124  void mover(unsigned long pulsos, bool direcao) {
125    for (unsigned long i = 0; i < pulsos; i++) {
126      d1.motorMove(direcao);
127    }
128  }
```

Código-fonte – setup()

```
130 void setup() {
131   //Configuracao do display --------------------
132   display.begin();
133   display.setContrast(50);
134   display.clearDisplay();
135   display.setTextSize(1);
136   display.setTextColor(BLACK);
137 
138   //Configuração do Driver DRV8825 -------------
139   // pin GND - Enable (ENA)
140   // pin 13 - M0
141   // pin 12 - M1
142   // pin 11 - M2
143   // pin 10 - Reset (RST)
144   // pin 9 - Sleep (SLP)
145   // pin 8 - Step (STP)
146   // pin 7 - Direction (DIR)
147   d1.pinConfig(99, 13, 12, 11, 10, 9, 8, 7);
148   d1.sleep(LOW);
149   d1.reset();
150   d1.stepPerMm(100);
151   d1.stepPerRound(200);
152   d1.stepConfig(1);
153   d1.motionConfig(50, velocidade, 5000);
154 
155 }
```

Código-fonte – loop() – 1ª parte – Desenhando menu

```
157 void loop() {
158
159   //Escreve o Menu do Programa no display ---
160   display.clearDisplay();
161   display.fillRect(0, 0, 15, 11, 2);
162   display.setCursor(2, 2);
163   display.setTextColor(WHITE);
164   display.print("F1");
165   display.setTextColor(BLACK);
166   display.setCursor(17, 2);
167   display.print("CRESCENTE");
168
169   display.fillRect(0, 12, 15, 11, 2);
170   display.setCursor(2, 14);
171   display.setTextColor(WHITE);
172   display.print("F2");
173   display.setTextColor(BLACK);
174   display.setCursor(17, 14);
175   display.print("DECRESCENTE");
176
177   display.fillRect(0, 24, 15, 11, 2);
178   display.setCursor(2, 26);
179   display.setTextColor(WHITE);
180   display.print("F3");
181   display.setTextColor(BLACK);
182   display.setCursor(17, 26);
183   display.print("VELOCIDADE");
184
```

Código-fonte – loop() – 2ª parte – Desenhando menu

```
184
185     display.fillRect(0, 36, 15, 11, 2);
186     display.setCursor(2, 38);
187     display.setTextColor(WHITE);
188     display.print("F4");
189     display.setTextColor(BLACK);
190     display.setCursor(17, 38);
191     display.print("ESC");
192
193     display.display();
194
195     bool esc = false;
```

Código-fonte – loop() – 3ª parte – Executando

```
197   //Loop enquanto a tecla F4 (ESC) nao for pressionada
198   while (!esc) {
199
200     //captura a tecla pressionada do teclado
201     customKey = customKeypad.getKey();
202
203     //caso alguma tecla foi pressionada
204     if (customKey) {
205
206       //Trata a tecla apertada
207       switch (customKey)
208       {
209         //Se tecla F1 foi pressionada
210         case 'A':
211           distancia = lerValor();
212
213           //Se tecla ESC foi pressionada
214           if (distancia == -1) {
215             esc = true;
216           } else {
217             //Escreve a tela "Movendo" no display
218             display.clearDisplay();
219             display.fillRect(0, 0, 84, 11, 2);
220             display.setCursor(21, 2);
221             display.setTextColor(WHITE);
222             display.print("MOVENDO");
223             display.setTextColor(BLACK);
224             display.setCursor(2, 14);
225             display.print(distancia);
226             display.print(" Passos");
227             display.display();
```

Código-fonte – loop() – 4ª parte – Executando

```
229             //Move o motor
230             mover(distancia, HIGH);
231
232             //Volta ao menu
233             esc = true;
234           }
235           break;
236
237         //Se tecla F2 foi pressionada
238         case 'B':
239           distancia = lerValor();
240
241           //Se tecla ESC foi pressionada
242           if (distancia == -1) {
243             esc = true;
244           } else {
245
246             //Escreve a tela "Movendo" no display
247             display.clearDisplay();
248             display.fillRect(0, 0, 84, 11, 2);
249             display.setCursor(21, 2);
250             display.setTextColor(WHITE);
251             display.print("MOVENDO");
252             display.setTextColor(BLACK);
253             display.setCursor(2, 14);
254             display.print(distancia);
255             display.print(" Passos");
256             display.display();
```

Código-fonte – loop() – 5ª parte – Executando

```
258             //Move o motor
259             mover(distancia, LOW);
260
261             //Volta ao menu
262             esc = true;
263           }
264           break;
265
266         //Se tecla F3 foi pressionada
267         case 'C':
268           velocidade = lerValor();
269           if (velocidade == -1) {
270             esc = true;
271           } else {
272             //Escreve a tela "Velocidade" no display
273             display.clearDisplay();
274             display.fillRect(0, 0, 84, 11, 2);
275             display.setCursor(12, 2);
276             display.setTextColor(WHITE);
277             display.print("VELOCIDADE");
278             display.setTextColor(BLACK);
279             display.setCursor(2, 14);
280             display.print(velocidade);
281             display.print(char(229));
282             display.print("s");
283
```

Código-fonte – loop() – 6ª parte – Executando

```
283
284                 display.fillRect(31, 24, 21, 11, 2);
285                 display.setCursor(33, 26);
286                 display.setTextColor(WHITE);
287                 display.println("OK!");
288                 display.setTextColor(BLACK);
289                 display.display();
290
291                 //Configura nova velocidade ao motor
292                 d1.motionConfig(50, velocidade, 5000);
293                 delay(2000);
294
295                 //Volta ao menu
296                 esc = true;
297               }
298               break;
299             //Se tecla F4 (ESC) foi pressionada
300             case 'D':
301             //Se tecla CLR foi pressionada
302             case 'c':
303             //Se tecla ENT foi pressionada
304             case 'e':
305             //Volta ao menu
306               esc = true;
307             default:
308               break;
309           }
310
311       }
312        //Limpa o char customKey
313       customKey = false;
314     }
315 }
316
```

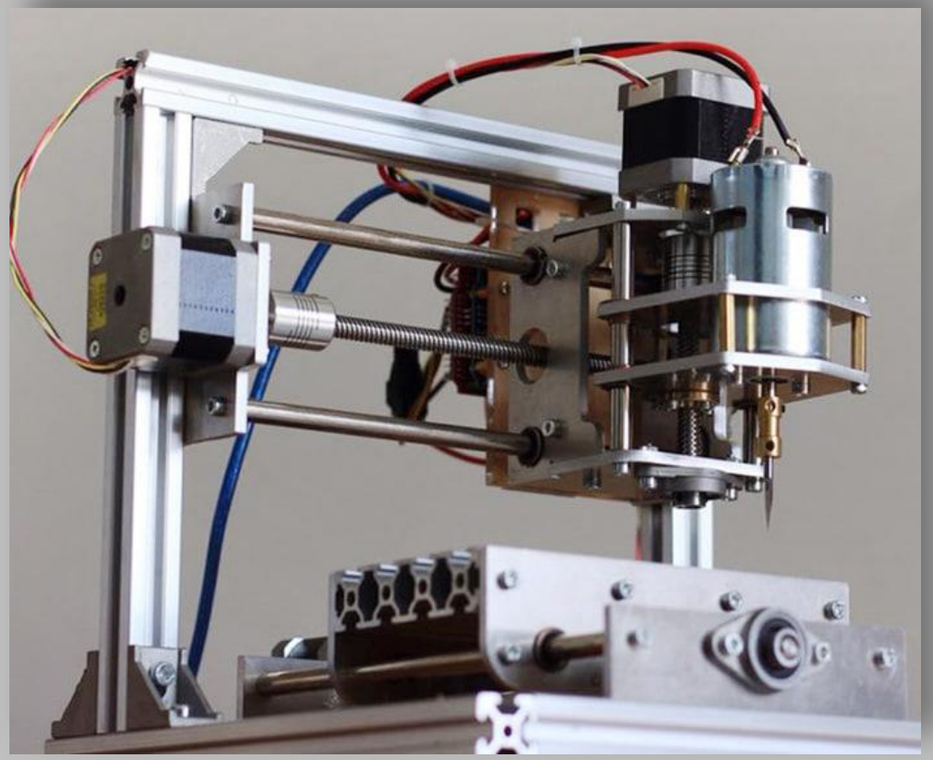
Sobre fusos – Configurações em máquinas

- Em máquinas CNC como impressoras 3D e routers por exemplo, o programa responsável pelo controle do posicionamento precisa saber como os movimentos ocorrerão em função do número de pulsos dados ao motor de passo.

- Se o driver do motor de passo permitir a aplicação de micro-passos, essa configuração deve ser levada em consideração no cálculo do deslocamento produzido.

- Por exemplo: se um motor de 200 passos por revolução, estiver ligado a um driver configurado para 1/16, então serão necessários 16 x 200 pulsos para uma única revolução do fuso, ou seja, 3200 pulsos para cada revolução. Se este fuso tiver passo de 2mm por revolução, serão necessários 3200 pulsos no driver para que a castanha se mova 2mm.

- De fato, os softwares controladores costumam usar um razão para especificar esta relação, o “número de pulsos por milímetro” ou “steps/mm”



Sobre fusos – Configurações em máquinas

- No Marlin, por exemplo, vemos na seção @section motion:

- /**
- * Default Axis Steps Per Unit (steps/mm)
- * Override with M92
- * X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
- */
- #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 3200, 100}

- Neste exemplo, podemos concluir que os eixos X e Y precisão de 80 pulsos para se deslocar 1mm, enquanto que o Z precisa de 3200 pulsos e o extrusor E0 precisa de 100.

```
/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *                                      X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT   { 80, 80, 3200, 100}
```

Sobre fusos – Configurações em máquinas

$100=250.000	X steps/mm
$101=250.000	Y steps/mm
$102=250.000	Z steps/mm

- Acima, vemos os comandos de configuração do GRBL. Com o comando $100, podemos ajustar o número de pulsos necessários para provocar um deslocamento de um milímetro no eixo X.
- No exemplo acima podemos ver que o valor atual é de 250 pulsos por mm.
- Os eixos Y e Z podem ser configurados respectivamente o $101 e $102

Em www.fernandok.com

Download arquivos PDF e INO do código fonte