

Máquina numericamente controlada para desenho de circuitos eletrônicos

Mateus Alves da Rocha

Programa de Graduação em Engenharia Eletrônica
Universidade de Brasília, UnB FGA
Gama, DF, Brasil
mateusalves1992@hotmail.com

Tatielen Rodrigues Dutra Pereira

Programa de Graduação em Engenharia Eletrônica
Universidade de Brasília, UnB FGA
Gama, DF, Brasil
tatielen.rodrigues@hotmail.com

Abstract— Este documento contém informações básicas sobre o projeto a ser desenvolvido na disciplina de Microcontroladores e microprocessadores da Universidade de Brasília, UnB, Campus Gama. Onde, será desenvolvido uma máquina CNC para auxiliar na elaboração de placas de circuito impresso.

Keywords—CNC, placa de circuito impresso, PCI, MSP430.

I. JUSTIFICATIVA

Quando se cria um projeto sempre existe a expectativa de fazê-lo em uma placa de circuito impresso (PCI). Normalmente os circuitos no começo são simples, logo as placas também, assim é possível desenhá-las usando apenas canetas próprias para isso. Conforme os conhecimentos são avançados os circuitos tendem a ficar maiores e complexos, não sendo possível desenhá-los manualmente, assim é necessário utilizar outra das possíveis formas de impressão de circuitos, algumas delas são o processo fotográfico, processo térmico, entre outros.

Ao escolher o processo de fotográfico, figura 1, é necessário imprimir um fotolito invertido e espelhado em uma transparência. Nesse passo já é possível que exista uma adversidade, pois na impressão das transparências por jato de tinta o preto pode não ficar opaco o suficiente para uma foto-impressão adequada, sendo o ponto fraco do processo [1]. Para que não ocorra esse problema, é necessário imprimir várias cópias e sobrepor-las. Depois é necessário limpar a placa adequadamente e aplicar uma emulsão foto sensível sobre a placa, após a emulsão estar seca coloca-se o fotolito sobre a placa e coloca-os em contato com a luz.

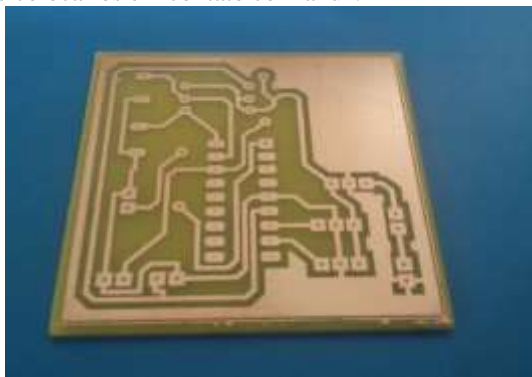


Figura 1 – Placa de circuito impresso confeccionada pelo processo fotográfico.

Ao utilizar o processo térmico, figura 2, será necessário imprimir o circuito em uma transparência igual ao processo descrito anteriormente, e depois colocar uma fonte de calor, um ferro de passar na maioria das vezes, sobre a transparência e a placa de cobre. Nesse processo é necessário atentar-se com o tempo do ferro sobre a placa, pois se ela aquecer demais o cobre pode aquecer demasiadamente e se desprender da placa estragando assim o trabalho. Caso ocorra pequenas falhas é possível corrigi-las com uma caneta de marcação permanente [2]. Após será usado uma solução de perclorato de ferro para remover materiais desejados da placa de cobre.



Figura 2 – Placa de circuito impresso confeccionada pelo processo térmico.

Na confecção de circuitos impressos pode haver vários erros na passagem dos circuitos para as placas como foi mostrado acima, com o uso da CNC será possível minimizar esses erros, pois ele irá desenhar o circuito na placa não precisando de nenhum tipo de transferência.

II. OBJETIVOS

Com o auxílio do microcontrolador msp430 será implementado um Controle Numérico Computadorizado ou Comando Numérico Computadorizado (CNC) para a impressão de circuitos impressos, ele conterá uma caneta permanente que será guiada pela placa por motores, o MSP430 receberá as coordenadas do circuito por um software.

III. REQUISITOS

A. Descrição do projeto

O projeto proposto nesse relatório visa atender às necessidades de projetistas de circuitos eletrônicos que demandem precisão nos desenhos dos circuitos e rapidez na produção de diversas placas simultaneamente. O projeto será de baixo custo uma vez que fará uso de peças reutilizadas de impressoras e outros dispositivos eletrônicos. E, por fim, deseja-se que seja portátil garantindo o fácil transporte do aparato.

B. Requisitos do sistema

A função desse sistema é substituir o método de transferência por meio de calor e também o método de transferência fotográfico. Para o sucesso desse projeto, a máquina deverá ser capaz de:

- Traçar com uma precisão satisfatória o esquemático de um circuito a partir das coordenadas fornecidas por um programa de processamento de imagens;
- Terminar o desenho do circuito rapidamente e com menor erro possível;
- Informar ao usuário quando o processo de desenho estiver terminado através de aviso sonoro e visual;
- Estimar o tempo de desenho e informar ao usuário através de um display localizado na própria CNC.

IV. BENEFÍCIOS

Com o uso da CNC será possível confeccionar placas de circuitos internos mais facilmente, pois ela terá mais precisão ao passar o circuito para a placa. Poderá ser feito PCB's em maior escala, pois a máquina poderá repetir o mesmo processo inúmeras vezes.

V. REVISÃO BIBLIOGRÁFICA

O estado da arte relacionado ao tipo de máquina que se deseja construir nesse projeto é bastante vasto. Foram criados dispositivos CNCs para variadas atividades. Desde construções de fresadoras CNCs, impressoras 3Ds até máquinas Plotters para mercados locais [3] [4] [5].

Inclusive, diversas soluções para o problema da produção de circuitos impressos foram feitas utilizando essas máquinas como em [6] e [7]. Entretanto, a maioria dos projetos utilizam o microcontrolador Arduino possivelmente por possuir muito projetistas que utilizam essa plataforma e por suas numerosas bibliotecas disponíveis.

Em sua maioria, os projetos utilizando o conceito de máquina numericamente controladas obtém sucesso. Embora, um dos maiores desafios é converter a imagem que deseja desenhar em coordenadas [8]. Assim como grande parte das soluções encontradas, este projeto utilizará softwares prontos para a conversão dessas imagens nos seus respectivos G-codes.

VI. MATERIAIS UTILIZADOS

O projeto deverá contar com um microcontrolador onde será programado o controle de toda aplicação e mecânico. Para a estrutura física da CNC será necessário:

- MSP430;
- 2 drives de DVD;
- 2 Motores Shield L293D;
- Servo Motor (tower pro 9G);
- 16MHz Crystal Oscillator;
- 2 capacitores de 22pF e 1 de 100nF.
- 10K resistor.
- Jumps;
- Parafusos, porcas e arruelas;
- Caneta.

Para a montagem da CNC primeiramente foram desmontados os drives de DVD para tirar os motores de passo e os trilhos, figura 3. Depois para a montagem dos eixos X e Y os motores de passo com trilhos foram colocado sobre placas de plástico para fazer a base da mesa da CNC como mostra a figura 4. O servo motor foi utilizado para fazer o eixo Z e ele juntamente com a caneta foram anexados a uma superfície plana no eixo Y, figura 5.



Figura 3 – Drives desmontados.



Figura 4 – Montagem dos eixos X e Y.



Figura 5 – Montagem do eixo Z.

Depois foi contruido o cicuito para testar os motores passo a passo (eixos X e Y). A CNC concluida na parte mecanica, figura 6.



Figura 6 – CNC concluida.

VII. VISÃO GERAL DO SISTEMA

Esta seção é dedicada a fornecer uma visão geral do sistema proposto no projeto. Para isto, serão utilizados dois esquemas gráficos comuns nas engenharias, o diagrama de blocos e o fluxograma.

A. Diagrama de Blocos

O diagrama de blocos tem como objetivo apresentar o princípio de funcionamento do objetivo em questão, neste caso, a máquina numericamente. A figura 7 apresenta o diagrama de blocos do sistema proposto. Os blocos apresentados são:

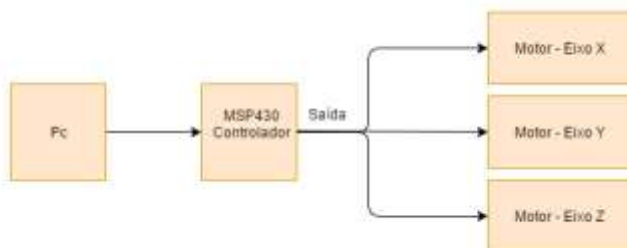


Figura 7 – Diagrama de Blocos da CNC.

- **Pc** é onde será desenhado a imagem com o auxílio do software desejado pelo usuário, por exemplo Proteus, depois é usado o *Inkscape* que é um software de gráficos vetoriais. Nele poderá configurar o tamanho da placa para o desejado, deixar a imagem com o fundo transparente, tirar detalhes indesejados e “traçar o mapa de bits”, as coordenadas. O arquivo trabalhado será um .gcode, para exportar imagens desse tipo de arquivo será necessário o *Add-on*.
- **Controlador** ele recebe as coordenadas e controla o que cada motor tem que executar, nos eixos X, Y e Z.
- **Motores** movimentam a caneta e a mesa onde a placa está fixada, como mostrada na figura 8.



Figura 8 – Motores CNC.

B. Fluxograma

O fluxograma é uma representação esquemática de um processo ou um algoritmo. Seu principal objetivo é apresentar uma forma mais simples o fluxo de informações e elementos, além da sequência operacional que caracteriza o trabalho que está sendo executado [9]. O fluxograma da máquina numericamente controlada é apresentado na figura 9.



Figura 9 – Fluxograma da CNC.

Pode-se observar que as atividades nos retângulos roxos são realizadas pelos usuários, isto é, a pessoa terá que escolher o circuito que deseja desenhado na placa, com o auxílio do software adquirir as coordenadas, já os retângulos azuis são realizados pelo sistema, neste caso a própria CNC.

O fluxograma inicia com a configuração e organização dos componentes e, uma vez que estes estejam prontos, pode-se ligar o sistema. A partir disso, o sistema irá definir, automaticamente, para onde mover a caneta desenhando assim somente nos locais desejados.

Após concluir todas as coordenadas recebidas o desenho está pronto, chegando ao fim do fluxograma e apresentando o desenho concluído. Assim ele está pronto para repetir o processo novamente, será necessário o usuário trocar a placa na qual será efetuado o novo desenho.

VIII. CÓDIGO

O código usado no MSP430 para ler as coordenadas previamente adquiridas e mover os motores está na figura 10, que será mais detalhado ao longo do texto.



Figura 10 – Código Completo.

Na figura 11, temos a parte 1 do código primeiro foi desligado o Watchdog timer, foi feito um ajuste no clock no MSP para ele funcionar mais lentamente, pode não seria possível que os motores se movesse de acordo com o clock padrão da placa.

```
1#include <msp430.h>
2#include "stepper_3AX.h"
3#include "main.h"
4/*
5 * main.c
6 */
7
8#define BAUD115200 //Será padrão para 9600
9
10int main(void) {
11    WDCTL = WDTPW | WDTHOLD; // Desligando watchdog timer
12
13    BCSCTL2 = SELM_0 + DIVM_0 + DIVS_0;
14    if (CALBC1_16MHZ != 0xFF) {
15        __delay_cycles(100000);
16
17        DCOCTL = 0x00;
18        BCSCTL1 = CALBC1_16MHZ; /* Define DCO para 16MHz */
19        DCOCTL = CALDCO_16MHZ;
20    }
21    BCSCTL1 |= XT2OFF + DIVA_0;
22    BCSCTL3 = XT2S_0 + LFX1S_2 + XCAP_1;
23    initUart();
24    PSEL = BIT1 + BIT2; // P1.1 = RXD, P1.2=TXD
25    PSEL2 = BIT1 + BIT2; // P1.1 = RXD, P1.2=TXD
26
27    IFG2 &= ~(UCA0TXIFG + UCA0RXIFG);
28    IE2 |= UCA0TXIE + UCA0RXIE;
29}
```

Figura 11 – Parte 1 do código.

Na parte 2, figura 12, começa a comunicação serial e define a baudrate. Na figura 13, recebe a coordenada e é usado uma variável auxiliar para ter certeza que os motores iram receber cada bit da coordenada.

```
31 StepperInit();
32 StepperEnable();
33
34 StepperGoTo(0,0,zHold);
35
36
37 //Testando
38 //Desativar Stepper();
39
40 __bis_SR_register(GIE);
41 return 0;
42}
43void initUart()
44{
45    UCA0CTL1 |= UCSWRST;
46    UCA0CTL1 = UCSSEL_2 + UCSWRST;
47#ifdef BAUD115200
48    UCA0MCTL = UCBRF_0 + UCBRS_7;
49    UCA0BR0 = 138;
50    UCA0BR1 = 0;
51#else
52    UCA0MCTL = UCBRF_0 + UCBRS_6;
53    UCA0BR0 = 130;
54    UCA0BR1 = 6;
55#endif
56
57
58    UCA0CTL1 &= ~UCSWRST;
59}
```

Figura 12 – Parte 2 do código.


```

61 int ConverteString(char começaIndex, char stopIndex)
62 {
63     int returnVal = 0;
64     char i;
65     int mult = 1;
66     for(i = paraIndex; i >= começaIndex; i--)
67     {
68         returnVal += ((RXBuffer[i] - 48) * mult);
69         mult *= 10;
70     }
71     return returnVal;
72 }
73

```

Figura 13 – Parte 3 do código.

Nas partes 4 e 5, figuras 14 e 15, está sendo recebendo os valores referentes das coordenadas nos eixos X, Y e Z e alternando qual a vez que cada motor vai funcionar. Nas figuras 16 e 17, o código configura a entrada serial da CNC.

```

74 //struct Point ProcessBuffer()
75 {
76     struct Point returnVal;
77     int x,y,z, temp;
78     x = -1;
79     y = -1;
80     z = -1;
81     char começaIndex, paraIndex;
82     char mode = 0; //0:esperando, 1:x, 2:y, 3:z
83
84     if(bufIndex > 2)
85     {
86         //Comando de movimento
87         if(RXBuffer[0] == 'G' && RXBuffer[1] == '0' && (RXBuffer[2] == '0' | RXBuffer[2] == '1'))
88         {
89             char i;
90             for(i = 3; i < bufIndex; i++)
91             {
92                 if(mode == 0) //Esperando por alguma entrada
93                 {
94                     if (RXBuffer[i] == 'X')
95                     {
96                         mode = 1;
97                         começaIndex = 0;
98                         paraIndex = 0;
99                     }
100                     else if (RXBuffer[i] == 'Y')
101                     {
102                         mode = 2;
103                         começaIndex = 0;
104                         paraIndex = 0;
105                     }
106                     else if (RXBuffer[i] == 'Z')
107                     {
108                         mode = 3;
109                         começaIndex = 0;
110                         paraIndex = 0;
111                     }
112                 }
113             }
114         }
115     }
116 }

```

Figura 14 – Parte 4 do código.

```

114
115     else if(mode > 0 && mode < 4)
116     {
117         if(começaIndex > 0 && (RXBuffer[i] == '.' || RXBuffer[i] == ','))
118         {
119             temp = ConverteString(começaIndex, paraIndex);
120             if(mode == 1)
121                 x = temp;
122             else if (mode == 2)
123                 y = temp;
124             else if (mode == 3)
125                 z = temp;
126             mode = 0; //Aguardar o próximo comando
127         }
128         else if (RXBuffer[i] >= 48 && RXBuffer[i] <= 57) //Número
129         {
130             if(começaIndex == 0)
131             {
132                 começaIndex = 1;
133                 paraIndex = 1;
134             }
135             else
136                 paraIndex = i;
137         }
138     }
139     _stop(); //Para breakpoint
140 }
141
142 returnVal.x = x;
143 returnVal.y = y;
144 returnVal.z = z;
145 StepperGoToCode(returnVal);
146 EnviaSerial("$r\n", 3);
147 return returnVal;
148 }

```

Figura 15 – Parte 5 do código.

```

150 void EnviaSerial(char * text, char len)
151 {
152     char index;
153     for(index = 0; index < len; index++)
154     {
155         char toDisp = *text++;
156         TXBuffer[index] = toDisp;
157     }
158     txIndex = 0;
159     txLen = len;
160     EnviaSerialNext(); //Começa a transmitir primeiro
161 }
162 void EnviaSerialNext()
163 {
164     if(txIndex < txLen)
165     {
166         IE2 |= UCA0TXIE;
167         //while (!(IFG2 & UCA0TXIFG));
168         UCA0TXBUF = TXBuffer[txIndex];
169         txIndex++;
170     }
171     else
172     {
173         IE2 &= ~UCA0TXIE;
174         txLen = 0;
175         txIndex = 0;
176     }
177 }
178

```

Figura 16 – Parte 6 do código.

IX. REFERÊNCIAS

```
179 //interrupção UART RX
180 #pragma vector=USCIAB0RX_VECTOR
181 __interrupt void USCI0RX_ISR_HOOK(void)
182 {
183     if(bufIndex >= 50)
184         bufIndex = 0; //no caso da string ser muito longa.
185
186     RXBuffer[bufIndex] = UCA0RXBUF;
187     bufIndex++;
188
189     if(UCA0RXBUF == '\n')
190     {
191         ProcessBuffer();
192         bufIndex = 0;
193     }
194 }
195 }
196
197 //interrupção UART TX
198 #pragma vector=USCIAB0TX_VECTOR
199 __interrupt void USCI0TX_ISR_HOOK(void)
200 {
201     EnviaSerialNext();
202 }
```

Figura 17 – Parte 7 do código.

- [1] NETO, Leocádio Benez. “Circuito impresso de qualidade com baixo custo”. Disponível em <<http://www.inape.org.br/eletronica/circuito-impresso-de-qualidade-com-baixo-custo>>. Acesso em: 04 de abril de 2017.
- [2] MARCIEL, Marcelo. “Placa de Circuito Impresso - PCI - Método de Transferência Térmica”. Disponível em <<http://www.marcelomaciel.com/2013/01/pci-metodo-termico.html#more>>. Acesso em: 04 de abril de 2017.
- [3] OCANHA, Denis. PROJETO E CONSTRUÇÃO DE UMA FRESADORA NUMÉRICAMENTE CONTROLADA. Universidade São Francisco, Itatiba – São Paulo, Brazil. Dezembro 2009.
- [4] ARRUDA, Caio Carlos et al. CONVERSÃO DE FRESADORA CNC PARA IMPRESSORA 3D. Universidade de São Paulo – USP.
- [5] VEIGA, Daniel Thadeu Torres Fernandes; DA SILVA, Fausto Amancio; TRÜEB, Sarah. CONFECÇÃO DE PLOTTER PCB PARA O MERCADO LOCAL. Universidade Salvador, Salvador – Bahia. VI congresso Nacional de Engenharia Mecânica, 18 a 21 de Agosto de 2010
- [6] ALCÂNTARA, Yuri Marinho; JUNIOR, Jander Pereira; SANTANTA, Lucas. Desenvolvimento de uma Mesa Fresadora CNC para Placas de Circuito Impresso para Telecomunicações. Seminário Estudantil de Produção Acadêmica, v. 14, 2015.
- [7] LUCCA, Jardel. Plotter de baixo custo para prototipação de placas de circuito impresso. 2013. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.
- [8] CASSIMIRO, ANDRÉ GABRIEL; DE OLIVEIRA, LUISA BONIN. Plotter Vertical. UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Curitiba, Paraná – 2014.
- [9] CITISYSTEMS, “Fluxograma de Processo – O que é, como elaborar benefícios”, 2013. Disponível em: <http://www.citisystems.com.br/fluxograma/#disqus_thread>. Acesso em: 10 de maio de 2017.