

Ponto de Controle 4

Máquina de CNC para Circuitos Impresos

Tales Maurício Presa Raulino

Engenharia Eletrônica - FGA

Universidade de Brasília

Brasília, Brasil

talesmauricioraulino@gmail.com

Vinícius Lisboa do Nascimento

Engenharia Eletrônica - FGA

Universidade de Brasília

Brasília, Brasil

lisboanascimento@gmail.com

Resumo—Esse trabalho se propõe a apresentar uma solução para o desenvolvimento de placas de circuito impresso em projetos no campus da FGA. Com a CNC que será criada utilizando o Raspberry Pi 3, projetos poderão ser enviados e impressos em placas de fenolite a distância.

Palavras-chave—*Raspberry Pi 3 B, Sistemas Embarcados, Máquina de controle numérico computadorizado.*

I. INTRODUÇÃO

Para o ponto de controle 4, houveram mudanças no programa que controla a cnc para atender às demandas especificadas do projeto. Houveram mais melhorias na parte do software embarcado na CNC.

II. OBJETIVO

Com o objetivo de fabricar placas de circuito impresso para a Faculdade Gama, a máquina de CNC deve dispor de um sistema embarcado que consiga ler e imprimir os principais tipos de arquivos de circuitos, e receber esses arquivos para impressão em sequência.

III. REQUISITOS

A Máquina de CNC deverá:

- Desenhar o circuito na PCI
- Ser capaz de se movimentar nos eixos X e Y com motores de passo;
- Regular a altura (eixo Z) com servo motor.
- Ler os principais tipos de arquivos para circuitos impressos;
- Ler arquivos advindos pelo Wifi, Bluetooth ou pelas portas USB;

Materiais utilizados:

- Raspberry Pi 3 B com o sistema embarcado desenvolvido durante a disciplina;
- Arduino Nano;
- Motores de passo NEMA17;
- Servo Motor 9G;
- Drivers de potência;
- Estabilizadores e eixos com rosca;
- Fonte de 5V, 9V e 12V;
- Placas de madeira como estrutura;
- Trilhos de gaveta como suporte para os eixos X, Y e Z;
- Universal Gcode Sender.

IV. BENEFÍCIOS

Quando finalizado, o projeto contará com uma estrutura simples e hábil à impressão em placas de circuito impresso, que poderão ser enviadas remotamente para esta máquina de CNC. Algo inovador e muito útil para uma demanda razoável de impressões e muito cômodo para quem necessita de um circuito impresso, mesmo estando longe deste maquinário.

V. DESENVOLVIMENTO

A partir do ponto de controle 3, o software da cnc, que antes era em python, foi alterado e convertido para a linguagem C, como o ensinado durante as aulas.

Esse programa foi desenvolvido para manter as mesmas funcionalidades do programa em python. Ele recebe o arquivo para a impressão e dá os comandos para o Arduino execute o layout. Nesse modelo o Raspberry é o *Master* e o Arduino é o *slave*.

O Mestre requisita comunicação com o escravo, e o escravo responde, dando o “ok” de recebido. A partir daí o Mestre envia os comando um de cada vez, enquanto o

Escravo executa e envia de novo o “ok” ao final de cada execução para que o Mestre envie mais um comando.

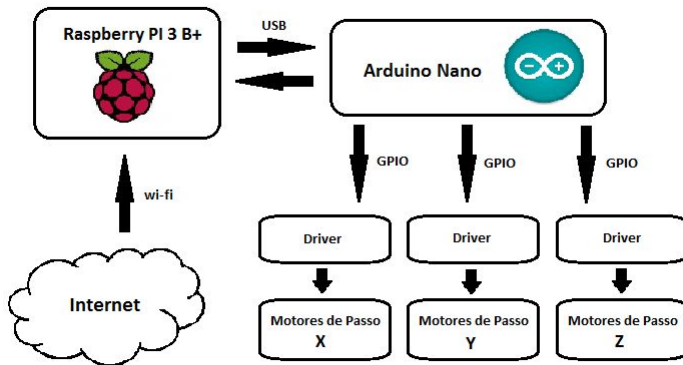


Fig. 1 - Diagrama de Blocos CNC

O código em C, é menor que o código em Python, é mais organizado e se consegue uma melhor execução da comunicação entre o Raspberry e o Arduino. As respostas do Arduino são mais claras e vem exatamente quando a tarefa acaba de ser executada por ele. Abaixo está a comparação entre os dois programas, mostrando essas diferenças.

Código em Python:

```
#-----
#      trilha 1 sem furos
#      aprofundamento de 1mm
#-----

import serial
import time;

#--
COD1 = """
G91 G0 X5.000

G91 G0 Z-5
G91 G0 Y4.000
G00 X10.000 Y10.000
G91 G0 Y3.000
G91 G0 X3.000
G91 G0 Y-4.000
G00 X-10.000 Y-10.000
G91 G0 Y-3.000
G91 G0 X-3.000
G91 G0 Z5
"""
COD2 = """
G91 G0 X5.000

G91 G0 Z-5
G91 G0 Y4.000
G00 X10.000 Y10.000
G91 G0 Y3.000
G91 G0 X3.000
G91 G0 Y-4.000
G00 X-10.000 Y-10.000
```

```
G91 G0 Y-3.000
G91 G0 X-3.000

G91 G0 Z5

"""
COD3 = """
G91 G0 X5.000

G91 G0 Z-5
G91 G0 Y4.000
G00 X10.000 Y10.000
G91 G0 Y3.000
G91 G0 X3.000
G91 G0 Y-4.000
G00 X-10.000 Y-10.000
G91 G0 Y-3.000
G91 G0 X-3.000
G91 G0 Z5

"""
COD4 = """
G00 X-20.000 Y-10.000
G91 G0 Z-4
```

```
"""
COD5 = """
COD6 = """
COD7 = """
COD8 = """
"""
```

```
s = serial.Serial("/dev/ttyUSB0", 9600)
time.sleep(1);

s.write(COD1);
time.sleep(1);

# x = s.read()          # read one byte
y = s.read(10)          # read up to ten bytes (timeout)
#line = s.readline()    # read a '\n' terminated line

#print(line)
print(y)
y = s.read(10)          # read up to ten bytes (timeout)
#line = s.readline()    # read a '\n' terminated line

#print(line)
print(y)
y = s.read(10)          # read up to ten bytes (timeout)
#line = s.readline()    # read a '\n' terminated line

#print(line)
print(y)
y = s.read(10)          # read up to ten bytes (timeout)
#line = s.readline()    # read a '\n' terminated line

s.write(COD2);
time.sleep(5);
s.write(COD3);
time.sleep(5);
s.write(COD4);
time.sleep(5);
s.write(COD5);
time.sleep(5);
s.write(COD6);
time.sleep(2);
s.write(COD7);
time.sleep(2);
s.write(COD8);
time.sleep(2);

#-- fim
s.close()
```

Código em C:

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <termios.h>

int main(int argc, char** argv)
{
    struct termios tio;
    struct termios stdio;
    struct termios old_stdio;
    int tty_fd;

    char c='D';
    tcgetattr(STDOUT_FILENO, &old_stdio);

    printf("Please start with %s /dev/ttyS1 (for\n", argv[0]);
    example) \n";
    memset(&stdio, 0, sizeof(stdio));
    stdio.c_iflag=0;
    stdio.c_oflag=0;
    stdio.c_cflag=0;
    stdio.c_lflag=0;
    stdio.c_cc[VMIN]=1;
    stdio.c_cc[VTIME]=0;
    tcsetattr(STDOUT_FILENO, TCSANOW, &stdio);
    tcsetattr(STDOUT_FILENO, TCSAFLUSH, &stdio);
    fcntl(STDIN_FILENO, F_SETFL, O_NONBLOCK);

    memset(&tio, 0, sizeof(tio));
    tio.c_iflag=0;
    tio.c_oflag=0;
    tio.c_cflag=CS8|CREAD|CLOCAL;
    tio.c_lflag=0;
    tio.c_cc[VMIN]=1;
    tio.c_cc[VTIME]=5;

    tty_fd=open(argv[1], O_RDWR | O_NONBLOCK);
    cfsetospeed(&tio, B9600);
    cfsetispeed(&tio, B9600);

    tcsetattr(tty_fd, TCSANOW, &tio);

    while (c!='q')
    {
        if (read(tty_fd, &c, 1)>0)
        write(STDOUT_FILENO, &c, 1);

        char url[]="teste.txt";

        FILE* arquivo = fopen(url, "r");
        if(arquivo == NULL) {
            fprintf(stderr, "Erro ao abrir o arquivo.txt.");
            return 1;
        }

        int contagem = 0;
        int caractere;
        do {
            caractere = fgetc(arquivo);

            write(tty_fd, &caractere, 1);

        } while(caractere != EOF);

        fclose(arquivo);

    }

    close(tty_fd);
    tcsetattr(STDOUT_FILENO, TCSANOW, &old_stdio);

    return EXIT_SUCCESS;
}
```

Com esse código mais reduzido e claro, além das mudanças já feitas nos pontos de controle anteriores, a execução dos arquivos a serem impressos na placa de fenolite é bem mais rápida e mais confiável.

VI. CONCLUSÃO

Desde o primeiro ponto de controle o projeto evoluiu bastante. Até esse ponto de controle, a estrutura e os principais aspectos estruturais da CNC. O Raspberry usado foi de grande importância para o projeto devido as funcionalidades que ele proporciona. Já é possível realizar a impressão em placas de fenolite com o auxílio da comunicação com o Arduino.

As mudanças feitas para esse ponto de controle atendem as demandas de software exigidas durante a apresentação do ponto de controle 3, para que a linguagem fosse em C em mais clara. Ainda conseguiu-se reduzir o tamanho do código e assim ganhar agilidade em sua execução.

A CNC já é um produto funcional que atende a impressão de placas para circuitos impressos de uma camada.

REVISÃO BIBLIOGRÁFICA

- [1] Professor D. Vernon, Final Year Project Handbook, revision 2.0, Etisalat University College, Jan. 2007. p. 1-8.
- [2] <https://www.raspberrypi.org/blog/home-made-cnc-milling-machine/> 1 de Abril de 2017
- [3] <http://www.lirtex.com/robotics/div-cnc-machine> 1 de Abril de 2017
- [4] <https://www.youtube.com/watch?v=EVfNYN4Z0cM> 1 de Abril de 2