

Relatório Final

Máquina de CNC para Circuitos Impressos

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

A Máquina de CNC é uma ferramenta muito útil, sendo ela usada em diversas indústrias e em diversos setores. Para a Faculdade Gama da UnB, todo semestre existem inúmeros projetos e demandas de circuitos impressos, tanto para trabalhos de matérias e projetos científicos como projetos pessoais. E essa demanda que a Máquina de CNC para Circuitos Impressos vem atender além de trazer mais comodidades em relação a outras máquinas semelhantes devido ao seu acesso remoto e automatização de processos..

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, automaticamente.

III. REQUISITOS

A Máquina de CNC deverá:

- Esculpir o circuito na PCI
- Ser capaz de se movimentar nos eixos X, Y e Z com motores de passo;
- Ler o principal tipo de arquivo para CNC (GRBL)
- Receber e ler arquivos advindos pelo Wifi(e-mail), Ethernet(e-mail) ou pelas portas USB;

Materiais utilizados:

- Raspberry Pi 3 B com o sistema embarcado desenvolvido durante a disciplina;
- Arduino Nano;
- Motores de passo NEMA17 e NEMA 23;
- Drivers de potência;
- Estabilizadores e eixos com rosca;
- Fonte de 3.3V, 5V e 12V;
- Placas de madeira como estrutura;
- Trilhos de gaveta como suporte para os eixos X, Y e Z;

IV. BENEFÍCIOS

O software desenvolvido utiliza como base o sistema operacional linux e comunicação USB com o arduino, portanto esse software pode ser utilizado em diversos dispositivos de baixo poder de processamento podendo ser utilizado por exemplo em computadores mais antigos que suportem o sistema linux e não tornando o raspberry Pi como único meio de comunicação.

Outro fator importante que foi objetivado no início do projeto foi a automatização dos processos além da opção de se utilizar um servidor de email para se enviar o código (GRBL). a ser impresso, característica fundamental do projeto tornando a impressão fácil e trazendo a comodidade ao usuário. Basta enviar um email com o código para o endereço especificado no software que automaticamente será esculpido o projeto na placa de fenolite ou outros materiais como madeiras e plásticos.

V. DESENVOLVIMENTO

Para iniciar o projeto, foram feitas e testadas as conexões de funcionalidades dos drivers de potência com os motores e

o Arduino, e do Arduino com o Raspberry Pi, para realmente verificar a viabilidade do projeto.

Primeiro foram feitas as estruturas com a madeira, trilhos e eixos a disposição. O Arduino foi montado a estrutura, juntamente com os drivers e motores.

Posteriormente o Raspberry foi conectado ao Arduino e testada a comunicação entre eles. Para esse teste foi feito um exemplo de desenho e comandado pelo Raspberry para que o Arduino o imprimisse, usando um lápis no lugar da fresadora. Também foi utilizada a interface de impressão Universal Gcode Sender para a comunicação entre o Raspberry e o Arduino, novamente apenas com a finalidade de testes e verificação de projeto.

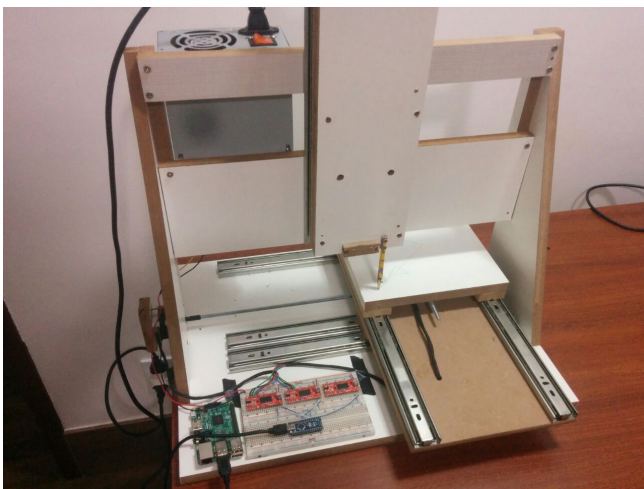
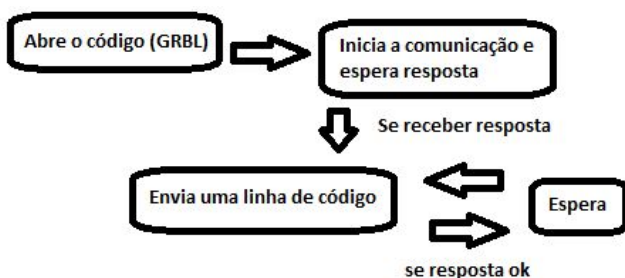


Fig. 1 - Estrutura da Máquina de CNC

Fig. 2 - Interface do Universal Gcode Sender no Raspberry

Com isto foi possível passar para o próximo passo. O terceiro eixo foi incorporado à estrutura como o motor NEMA 23 e o código de comunicação entre arduino e raspberry foi iniciada utilizando-se as portas USB das placas (para dar uma maior gama de dispositivos caso o usuário queira utilizar outra plataforma que não o raspberry) e código em C# para um melhor ajuste ao projeto.

A comunicação entre arduino e Raspberry apresenta esquemático como ilustrado abaixo:



```

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

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

    char c='D';
    char inicio='0';
    int controle =0;
    int teste = 0;

    int contagem = 0;
    int caractere;
    char url[]="teste.txt";

    // FILE* arquivo = fopen(url,"r");
    FILE* arquivo = fopen("/home/pi/CNC/temp/teste.txt","r");

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

    tcsetattr(STDOUT_FILENO,old_stdio);

    // printf("Please start with ts /dev/ttyS1 (for example)\n",argv[0]);
    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); // make the reads non-blocking

    memset(&tio,0,sizeof(tio));
    tio.c_iflag=0;
    tio.c_oflag=0;
    tio.c_cflag=CS8|CREAD|CLOCAL; // 8n1, see termios.h for more information
    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); // 115200 baud
    cfsetispeed(&tio,B9600); // 115200 baud
    tcsetattr(tty_fd,TCSANOW,&tio);

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

            if (c == 'I'){
                controle =1;
            }
        }
        while (controle == 1)
        {
            if (read(tty_fd,&c,1)>0) write(STDOUT_FILENO,&c,1);
            if (read(STDIN_FILENO,&c,1)>0) write(tty_fd,&c,1);
            else{
                caractere = fgetc(arquivo);
                write(tty_fd,&caractere,1);
                write(STDOUT_FILENO,&caractere,1);
                if (caractere > 0){
                    printf("%c", caractere);
                    teste = 0;
                }
                while (caractere == '\n' && teste ==0) {
                    if (read(tty_fd,&c,1)>0) write(STDOUT_FILENO,&c,1);
                    if (read(STDIN_FILENO,&c,1)>0) write(tty_fd,&c,1);
                    if (c == 'k'){
                        teste=1;
                    }
                }
                while (caractere == '&' && teste ==0) {
                    fclose(arquivo);
                    close(tty_fd);
                    tcsetattr(STDOUT_FILENO,TCSANOW,&old_stdio);
                    return EXIT_SUCCESS;
                    return 0;
                }
            }
        }
    }

    while(caractere != EOF){
        printf("EXIT");
    }

    fclose(arquivo);
    close(tty_fd);
    tcsetattr(STDOUT_FILENO,TCSANOW,&old_stdio);
    return EXIT_SUCCESS;
}

```

Fig. 3 - Código em C do Raspberry para comandar a impressão

Para o funcionamento da CNC foi utilizado o sistema de *threads* em que 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.

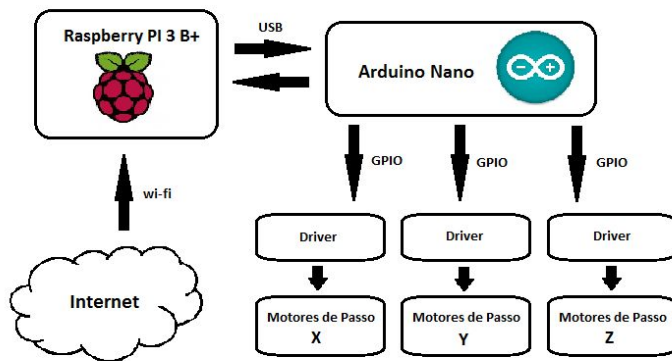


Fig. 1 - Diagrama de Blocos CNC

Para a aquisição dos arquivos a serem impressos pelo e-mail foi utilizado o *fetchmail* e configurado para averiguar a cada minuto se existem novos e-mails com pedidos de impressão, com a adição da linha “*/I * * * * /usr/bin/fetchmail > /dev/null 2>&I” no arquivo de *crontab* -e.

```

set postmaster "pi"
poll pop.gmail.com with protocol POP3 and options no dns
user 'viniisseals@gmail.com' there with password 'sembarcados'
is pi here options ssl
mda /sbin/fetchmailhandler
  
```

Fig. 4 - arquivo para controlar o fetchmail

Juntamente com isto, foi criado um programa para verificar a pasta onde o email deve chegar frequentemente, e executar o início do processo de impressão.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int busca = 0;
int main()
{
    while ( busca==0){
        FILE *fp;
        fp = fopen ("/home/pi/CNC/mail/email.txt", "r");
        if (fp == NULL) {
            printf ("Procurando...\n");
            return 1;
        }
        else{
            printf ("Arquivo aberto.\n");
            system("./filtro");
            //
            system("./serial5 /dev/ttyUSB0");
            fclose (fp);
            busca=1;
            return 0;
        }
    }
    sleep(8);
}
  
```

Fig. 5 - Código de busca

Como não foi possível o download do anexo, foi feito um *script* para filtrar (em 3 etapas: a parte superior, a inferior e um marcador colocado no início) o que deveria ser impresso e o que pode ser descartado, e assim enviar o arquivo correto para o arduino.

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int valida =0;
    char c;
    FILE *f1, *f2;
    f1 = fopen("/home/pi/CNC/mailf1/teste.txt", "rt");
    if (f1 == NULL)
    {
        printf("Nao foi possivel abrir 1");
        exit(EXIT_FAILURE);
    }
    //
    f2 = fopen("/home/pi/CNC/mailf2/teste.txt","wt");
    f2 = fopen("/home/pi/CNC/temp/teste.txt","wt");
    if (f2 == NULL)
    {
        printf("Nao foi possivel abrir 2");
        exit(EXIT_FAILURE);
    }
    printf("arquivos abertos\n");
    c = fgetc(f1);
    while (!feof(f1))
    {
        if (c=='G'){
            valida=1;
        }
        if( valida== 1){
            fputc(c, f2);
        }
        c = fgetc(f1);
    }
    printf("transferencia terminada\n");
    //
    system("./filtro2");
    system("./serial5 /dev/ttyUSB0");
    fclose (f1);
    fclose (f2);
    exit(EXIT_SUCCESS);
}
  
```

Fig. 6 - Filtro para a parte o marcador

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int valida =0;
    char c;
    FILE *f1, *f2;
    f1 = fopen("/home/pi/CNC/mailf2/teste.txt", "rt");
    if (f1 == NULL)
    {
        printf("Nao foi possivel abrir 1");
        exit(EXIT_FAILURE);
    }
    f2 = fopen("/home/pi/CNC/temp/teste.txt","wt");
    if (f2 == NULL)
    {
        printf("Nao foi possivel abrir 2");
        exit(EXIT_FAILURE);
    }
    printf("arquivos abertos\n");
    c = fgetc(f1);
    while (!feof(f1))
    {
        if( valida== 0){
            fputc(c, f2);
        }
        if (c=='s'){
            valida=1;
        }
        c = fgetc(f1);
    }
    printf("transferencia terminada\n");
    //system("./filtro2");
    system("./serial5 /dev/ttyUSB0");
    fclose (f1);
    fclose (f2);
    exit(EXIT_SUCCESS);
}
  
```

Fig. 7 - Filtro para a parte inferior

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int valida = 0;
    char c;
    FILE *f1, *f2;
    f1 = fopen("/home/pi/CNC/mail/email.txt", "rt");
    if (f1 == NULL)
    {
        printf("Nao foi possivel abrir 1");
        exit(EXIT_FAILURE);
    }
    f2 = fopen("/home/pi/CNC/mailf1/teste.txt", "wt");
    if (f2 == NULL)
    {
        printf("Nao foi possivel abrir 2");
        exit(EXIT_FAILURE);
    }
    printf("arquivos abertos\n");
    c = fgetc(f1);
    while (!feof(f1))
    {
        if (c == '#'){
            valida = 1;
        }
        if( valida== 1){
            fputc(c, f2);
        }
        c = fgetc(f1);
    }
    printf("transferencia terminada e chama filtro1\n");
    system("./filtro1");
    fclose(f1);
    fclose(f2);
    exit(EXIT_SUCCESS);
}
```

Fig. 8 - Filtro para a parte superior

VI. RESULTADOS

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.

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.

Já para a aquisição do arquivo por e-mail, o uso de *fetchmail* foi útil, mas como não foi possível a aquisição do anexo, o arquivo da impressão foi colocado no corpo do e-mail. a imagem abaixo mostra como o e-mail chega na pasta.

```
X-Gm-Message-State: AIVw113GdOCuzy2rAB9vmDj7Tvjg2FFK2tY2ySBJG6rk4u3aijwBE3Ce
lgtxUP+tk+2MKrL6Qzj3ucqp/WOL28k3
X-Received: by 10.107.31.20 with SMTP id f20mr6956155iof.116.1499206952268;
Tue, 04 Jul 2017 15:22:32 -0700 (PDT)
MIME-Version: 1.0
Received: by 10.79.127.193 with HTTP; Tue, 4 Jul 2017 15:22:31 -0700 (PDT)
From: "UTF-8?Tales_Maur=C3=ADcio?" <talesmauricioraulino@gmail.com>
Date: Tue, 4 Jul 2017 19:22:31 -0300
Message-ID: <CAQgwZhaXuk86eDz=UWAaltj1e0Lwzen47RVnHj8A9Xxf7XpdA@mail.gmail.com>
Subject: Vai pchha
To: viniisseals@gmail.com
Content-Type: multipart/alternative; boundary="001a114032c2d5bc3d05538551e3"

--001a114032c2d5bc3d05538551e3
Content-Type: text/plain; charset="UTF-8"

#
G90 G21 G49 G17

G00 Z20.000

X32.690 Y30.050

G00 Z0.500

G01 Z-0.200 F80

G03 X33.1882 Y29.5376 R2.75 F40

X33.8026 Y29.1716 R2.75

X34.4907 Y28.9772 R2.75

X34.9023 Y28.9715 R2.75

X35.2057 Y28.9674 R2.75

X35.8989 Y29.143 R2.75

X36.523 Y29.4921 R2.75

X37.0355 Y29.9908 R2.75

X37.4014 Y30.6051 R2.75

X37.5959 Y31.2932 R2.75

X34.8682 Y34.4041 R2.75

X34.5648 Y34.4082 R2.75

X33.8716 Y34.2326 R2.75

X33.2475 Y33.8835 R2.75

X32.7351 Y33.3848 R2.75

X32.3691 Y32.7705 R2.75

X32.1746 Y32.0824 R2.75

X32.169 Y31.6708 R2.75

X32.1649 Y31.3674 R2.75

X32.3405 Y30.6742 R2.75

X32.6895 Y30.0501 R2.75
#

--001a114032c2d5bc3d05538551e3
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr"><div style=3D"font-size:12.8px">#</div><div style=3D"font-size:12.8px">G90 G21 G49 =C2=A0G17</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">G00 Z20.000</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">=C2=A0X32.690 Y30.050</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">G00 Z0.500</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">G01 Z-0.200 F80 =C2=A0</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">G03 X33.1882 Y29.5376 R2.75 F40 =C2=A0</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X33.8026 Y29.1716 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X34.4907 Y28.9772 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X34.9023 Y28.9715 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X35.2057 Y28.9674 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X35.8989 Y29.143 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X36.523 Y29.4921 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X37.0355 Y29.9908 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X37.4014 Y30.6051 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X37.5959 Y31.2932 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X34.8682 Y34.4041 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X34.5648 Y34.4082 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X33.8716 Y34.2326 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X33.2475 Y33.8835 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X32.7351 Y33.3848 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X32.3691 Y32.7705 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X32.1746 Y32.0824 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X32.169 Y31.6708 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X32.1649 Y31.3674 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X32.3405 Y30.6742 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">X32.6895 Y30.0501 R2.75</div><div style=3D"font-size:12.8px"><br></div><div style=3D"font-size:12.8px">#</div></div>
```

Fig. 9 - E-mail advindo pelo fetchmail

Este e-mail passando pelo processo de filtragem, retorna o arquivo pronto para ser enviado pelo Raspberry para impressão do Arduino. Pode-se notar que o código já não tem mais os marcadores colocados e nem o resquício do protocolo do *fetchmail* utilizado.


```

G90 G21 G49 G17

G00 Z20.000

X32.690 Y30.050

G00 Z0.500

G01 Z-0.200 F80

G03 X33.1882 Y29.5376 R2.75 F40

X33.8026 Y29.1716 R2.75

X34.4907 Y28.9772 R2.75

X34.9023 Y28.9715 R2.75

X35.2057 Y28.9674 R2.75

X35.8989 Y29.143 R2.75

X36.523 Y29.4921 R2.75

X37.0355 Y29.9908 R2.75

X37.4014 Y30.6051 R2.75

X37.5959 Y31.2932 R2.75

X37.6015 Y31.7049 R2.75

X37.6057 Y32.0083 R2.75

X37.43 Y32.7014 R2.75

X37.0810 Y33.3255 R2.75

X36.5823 Y33.838 R2.75

X35.968 Y34.204 R2.75

X35.2798 Y34.3985 R2.75

X34.8682 Y34.4041 R2.75

X34.5648 Y34.4082 R2.75

X33.8716 Y34.2326 R2.75

X33.2475 Y33.8835 R2.75

X32.7351 Y33.3848 R2.75

```

Fig. 10 - E-mail após a filtragem pronto para impressão

Com todo esse aparato, foi possível um exemplo de impressão da Máquina de CNC com a utilização de um lápis para simular o caminho que a fresa faria.

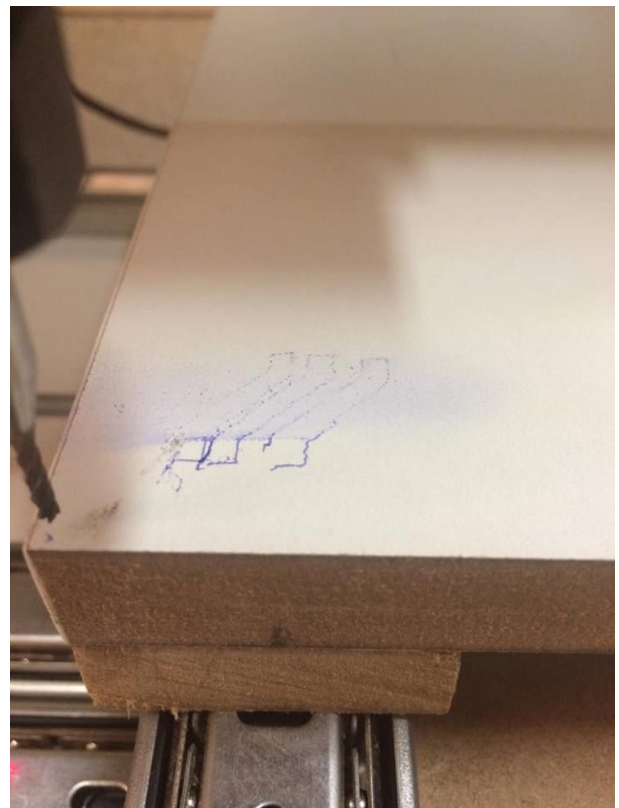


Fig. 11 - Exemplo de impressão

VII. CONCLUSÃO

Desde o primeiro ponto de controle o projeto evoluiu bastante. Até esse ponto de controle 4, 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 o ponto de controle 4 atenderam 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 adição do elemento, que baixa periodicamente os e-mails enviados a uma conta gmail, criada para a Máquina de CNC, se provou bastante útil, possibilitando a impressão de circuitos a distância e automaticamente. Sem o Raspberry, essa funcionalidade jamais seria atendida, utilizando somente o Arduino.

A CNC já é um produto funcional que atende a impressão de placas para circuitos impressos de uma camada, e projetos enviados para o e-mail da máquina de qualquer lugar.

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 2017
- [5] <https://gist.github.com/cristianp6/c063e36826a6b1623f3d> 25 de junho de 2017