



Sérgio Valle dos Reis

Painel Braille Interativo

Campinas
2013

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Sérgio Valle dos Reis

Painel Braille Interativo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Engenharia Elétrica, na área de Eletrônica, Microeletrônica e Optoeletrônica.

Orientador: Prof. Dr. José Antonio Siqueira Dias

Esse exemplar corresponde à versão final da dissertação defendida pelo aluno Sérgio Valle dos Reis, e orientada pelo Prof. Dr. José Antonio Siqueira Dias.

Campinas
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

R277p Reis, Sérgio Valle, 1960-
Painel braille interativo / Sérgio Valle dos Reis. – Campinas, SP : [s.n.], 2013.

Orientador: José Antonio Siqueira Dias.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Livros para cegos. 2. Deficientes visuais - Serviços para. 3. Cegos. 4. Deficientes visuais. I. Dias, José Antonio Siqueira Dias, 1954-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Interactive braille panel

Palavras-chave em inglês:

Braille books

People with visual disabilities - Services for

Blind

Visually impaired

Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica

Titulação: Mestre em Engenharia Elétrica

Banca examinadora:

José Antonio Siqueira Dias [Orientador]

José Antônio dos Santos Borges

Luiz César Martini

Data de defesa: 11-07-2013

Programa de Pós-Graduação: Engenharia Elétrica

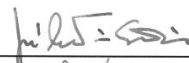
COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Sérgio Valle dos Reis

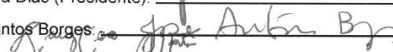
Data da Defesa: 11 de julho de 2013

Título da Tese: "Painel Braille Interativo"

Prof. Dr. José Antonio Siqueira Dias (Presidente):



Prof. Dr. José Antônio dos Santos Borges:



Prof. Dr. Luiz César Martini:

Agradecimentos

À minha companheira Teca, por ser a luz que ilumina o meu caminho e por ter sugerido que eu transformasse um projeto pessoal em projeto de mestrado.

Ao professor Dr. José Antonio Siqueira Dias, por ter acreditado na minha proposta e pelas valiosas sugestões apresentadas.

Aos colegas e amigos do departamento pelas contribuições ao meu projeto.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Departamento de Eletrônica e Microeletrônica da Faculdade de Engenharia Elétrica e Computação da UNICAMP (DEMIC), pela concessão da bolsa de estudo.

Este trabalho é dedicado à Teca (minha companheira) e aos meus filhos Lucas e Alice.

O amor é um poderoso instrumento de transformação da realidade. A verdadeira compaixão (que é o amor incondicional por todos os entes vivos) requer um elevado nível espiritual, muito além do alcance da maioria de nós. Entretanto, todos podemos exercitar a compaixão com aqueles que nos são próximos: a companheira ou companheiro, os filhos, os pais, os irmãos, os amigos, os conhecidos em geral. Se nos empenharmos em fazer isto, estaremos prestando enorme contribuição para a evolução e bem estar de toda a humanidade.

Resumo

O acesso a material de leitura didático (e mesmo para lazer) em braille é muito restrito e representa uma grande dificuldade para aquelas pessoas com deficiência de visão grave. Os lançamentos pelas editoras de títulos em braille são quase inexistentes, já que este tipo de produção é trabalhosa e cara. No Brasil, apenas algumas instituições oferecem estes produtos. Uma alternativa para sanar este problema é a utilização de dispositivos chamados "displays braille" que, acoplados a um computador, permitem a apresentação de um texto em braille. Estes dispositivos são caros (custam na faixa de 3000 dólares) e não são fabricados no Brasil.

Neste trabalho, propomos uma nova abordagem para a produção de um dispositivo dinâmico e interativo de leitura em braille. A abordagem proposta permitirá a produção de um "Painel Braille" com custo aproximado de um décimo do valor dos atuais dispositivos existentes no mercado. É apresentado o desenho de um display braille com a utilização de peças octogonais giratórias com os pontos braille, estas peças giram em torno de um eixo e são movimentadas através da utilização de dois motores de passo. Descrevemos a construção de um protótipo e os testes realizados.

Palavras-chave: Braille, display braille.

Abstract

The access to didactic and leisure reading material in braille is very restricted and it represents a huge difficulty for those with severe vision impairment. Publishers rarely launch braille versions of its books, because this kind of production is very laborious and expensive. An alternative to solve this issue is the use of braille displays, which when connected to a computer allows the presentation of text in braille dynamically. These devices are expensive (circa 3 thousand dollars) and aren't produced in Brazil.

In this work, we propose a new approach to the production of a dynamic and interactive braille display device. This suggested approach will enable the production of a cheaper device, with cost close to about ten percent of the full value of current devices on the market. This work devises a design of a braille display device using spinning octagonal pieces with braille marks, these pieces spin around a screw axis driven by two stepper motors. It reports the construction of a prototype and the tests made with it.

Keywords: Braille, Braille display.

Sumário

Lista de Figuras	xi
1 Introdução	1
1.1 Objetivo	2
2 Displays braille existentes no mercado	3
2.1 Introdução	3
2.2 Célula Braille da METEC AG	3
2.3 Displays braille disponíveis no Brasil	4
2.3.1 Braille BI 32 - fabricante: Humanware	5
2.3.2 Braille BI 40 - fabricante: Humanware	6
2.3.3 Braille B 80 - fabricante: Humanware	6
2.3.4 Linha Braille Usb 640 - fabricante: Optelec	7
2.4 Custo de displays braille comerciais	7
3 Construção de um Painel Braille	9
3.1 O Disco Braille	9
3.2 O fuso	12
3.3 O painel	13
3.4 A produção das peças de plástico	15
3.5 A eletrônica de controle do painel	18
3.5.1 Descrição da placa de acionamento	20
3.5.2 Descrição do MSP430 LaunchPad	24
4 Testes funcionais e conclusão	25
4.1 Desempenho do Painel	25
4.2 Contribuições	26
4.3 Conclusões	27
5 Trabalhos Futuros	28
Referências bibliográficas	29
A Listagem do Programa em C	30

B Listagem do Programa em G-CODE**48**

Lista de Figuras

2.1	Cécula braille da METEC: (a) foto (b) desenho esquemático [1]	4
2.2	Display braille modelo Brailiant BI 32	5
2.3	Display braille modelo Brailiant BI 40	6
2.4	Display braille modelo Brailiant B 80	6
2.5	Display braille modelo Usb 650	7
3.1	Cela braille (desenho extraído da NBR 9050)	9
3.2	Desenho do disco braille	10
3.3	Foto disco braille: frente	11
3.4	Foto disco braille: verso	12
3.5	Foto do fuso sem o painel	13
3.6	Foto do painel	14
3.7	Foto do painel montado	15
3.8	Foto da impressora 3D	16
3.9	Tela do software utilizado para desenhar as peças de plástico	17
3.10	Placa de controle do painel	18
3.11	Placa de acionamento dos motores de passo	19
3.12	Placa de controle montada sobre placa de acionamento	19
3.13	Esquemático da placa de acionamento	20
3.14	Esquemático do MSP430 LaunchPad (parte 1 de 3)	21
3.15	Esquemático do MSP430 LaunchPad (parte 2 de 3)	22
3.16	Esquemático do MSP430 LaunchPad (target socket)(parte 3 de 3)	23

Capítulo 1

Introdução

As alternativas de "leitura" para as pessoas com problemas de visão são: textos impressos em braille, "livros falados" e a utilização de computadores com o auxílio de softwares que reproduzam, com a utilização de voz artificial, o conteúdo apresentado na tela do computador.

Os "livros falados" e os textos em braille requerem uma produção especial e a disponibilidade de títulos é, por vários motivos, bem pequena. Além disso, quando do lançamento de livros novos, o intervalo de tempo até a disponibilização destes títulos na forma de livros em braille ou "livros falados" é muito grande, chegando a dois ou mais anos. Os livros em braille ainda têm o inconveniente de serem muito volumosos.

Já os "softwares de leitura de tela" permitem o acesso a qualquer título ou conteúdo que esteja disponível em formato digital. A sofisticação destas ferramentas de software tem inclusive contribuído para uma menor utilização do braille, já que a aprendizagem do braille é relativamente difícil e requer um treino de sensibilidade. Não é necessária nenhuma habilidade especial para a utilização destes softwares, o que não é verdade para o caso da utilização do braille.

Entretanto, a importância do braille permanece e pode ser facilmente percebida quando um determinado livro está disponível tanto na forma de livro impresso em braille quanto na forma de conteúdo digital. Neste caso, para leitores com experiência de leitura em braille, a leitura do livro impresso é muito mais ágil assim como é mais fácil a consulta, posterior à leitura completa do livro, de determinados trechos que se deseje rever.

Os "softwares de leitura de tela" também não resolvem a questão do acesso a títulos publicados recentemente porque na maior parte das vezes estes títulos não estão disponíveis em formato digital aberto.

Por outro lado, modernos estudos de neurologia [2] realizados com pessoas videntes sugerem que o ato de ler provoca alterações no cérebro e inclusive estabelecem uma relação direta entre o grau de "letramento" de um indivíduo e estas alterações. Por analogia, podemos supor que a aprendizagem do braille provoque mudanças similares e possa ter consequências benéficas para o desempenho intelectual em geral.

É na intersecção da utilização do braille e do computador que surgem os dispositivos capazes de exibir em braille uma parte de um texto selecionado na tela do computador. Estes dispositivos são denominados de "displays brilles" e existem em vários modelos com capacidade de apresentar 20, 40 ou 80 caracteres em braille. Estas linhas dispõem de "pinos" móveis que sobem ou descem para apresentar os pontos que constituem os caracteres em braille. Geralmente são equipamentos

periféricos, e não são úteis a não ser que estejam acoplados a um computador. Um display braille com 40 caracteres terá 240 pinos móveis que têm que ser movidos para cima e para baixo de forma independente. Isto faz com que estes dispositivos não sejam baratos: custam na faixa de 3000 dólares.

Não existem fabricantes nacionais de displays brailles e o alto custo destes equipamentos aliado às despesas de importação fazem com que estes dispositivos sejam muito pouco utilizados no Brasil.

No contexto deste trabalho adotamos a terminologia "painel braille" para nos referirmos a um display braille que possa funcionar sem estar acoplado a um computador e que funcione como um livro eletrônico em braille.

1.1 Objetivo

O objetivo deste trabalho é a proposição de uma nova forma de produção de um painel braille que permita a fabricação de um dispositivo de baixo custo e, portanto, adequado à realidade brasileira.

Espera-se que o custo de produção deste painel braille seja tão baixo que permita a utilização deste equipamento em todas as escolas e bibliotecas e, quem sabe até, o fornecimento do mesmo pelo governo a cada pessoa com deficiência visual, dentro das atuais políticas de acessibilidade desenvolvidas no âmbito governamental.

Capítulo 2

Displays braille existentes no mercado

2.1 Introdução

A maioria dos displays braille são dispositivos eletro-mecânicos que apresentam os caracteres braille pela movimentação de pinos com pontas arredondadas através de furos em uma superfície plana. [3] O mecanismo mais empregado para movimentar os pinos é a utilização de cristais piezoelétricos (que se expandem quando uma tensão é aplicada). Neste caso, é necessário a utilização de um cristal para cada ponto (ou 6 cristais por caracter). Outros displays braille utilizam um mecanismo eletromagnético de movimentação dos pinos.

Nos displays braille eletromagnéticos cada pino é rodeado por um invólucro cilíndrico que contém uma bobina. O pino é ligado a uma mola e também a uma haste de ferro que passa através do invólucro. O conjunto funciona como um solenóide em miniatura. Quando passa uma corrente pela bobina, o campo magnético gerado atrai a haste de ferro e o pino baixa. Quando não há corrente passando pela bobina, a mola mantém o pino levantado.

Nos displays piezoelétricos, cada pino é montado sobre uma peça de metal que por sua vez está ligada a um cristal piezoelétrico. Se uma tensão suficientemente grande é aplicada sobre o cristal, este torna-se ligeiramente mais curto. Isto faz com que a peça de metal se curve para cima, levantando o pino. Por outro lado, na ausência de tensão, o cristal volta a sua dimensão normal desfazendo a curvatura da peça de metal e, consequentemente, fazendo o pino baixar.

A maioria dos displays braille utiliza células braille piezoelétricas pois estas células são comercialmente disponíveis [4] além de serem relativamente leves, de tamanho reduzido e controladas pela aplicação direta de uma tensão elétrica o que simplifica bastante o circuito de controle.

2.2 Célula Braille da METEC AG

A empresa Metec AG, baseada em Stuttgart, Alemanha, desenvolve e fabrica vários tipos de células braille desde o ano de 1974. A figura 2.1 mostra a foto e o desenho esquemático do modelo P16 de célula braille fabricada pela METEC. Esta célula apresenta 8 pinos acoplados a barras de metal que por sua vez estão ligados a cristais piezoelétricos bimórficos. A aplicação de tensão ao cristal faz com que a barra se dobre baixando o pino ("ponto da célula braille"). A retirada da tensão faz a barra voltar ao normal levantando o pino. A voltagem a ser aplicada ao cristal é da ordem de

200 Volts DC. Os cristais estão montados sobre uma placa de circuito impresso. Na borda desta placa há um conector de 10 vias. O pino 1 deste conector deve ser conectado a uma tensão DC positiva de 200V e o pino 2 ao terra. Os outros oito pinos correspondem aos 8 pontos da célula. Quando a voltagem de 200 Volts é aplicada a um determinado pino, o cristal bimórfico ligado ao pino se dobra e o ponto correspondente baixa. Se a tensão de 0 Volts é aplicada ao pino, ocorre o contrário e o ponto sobe.

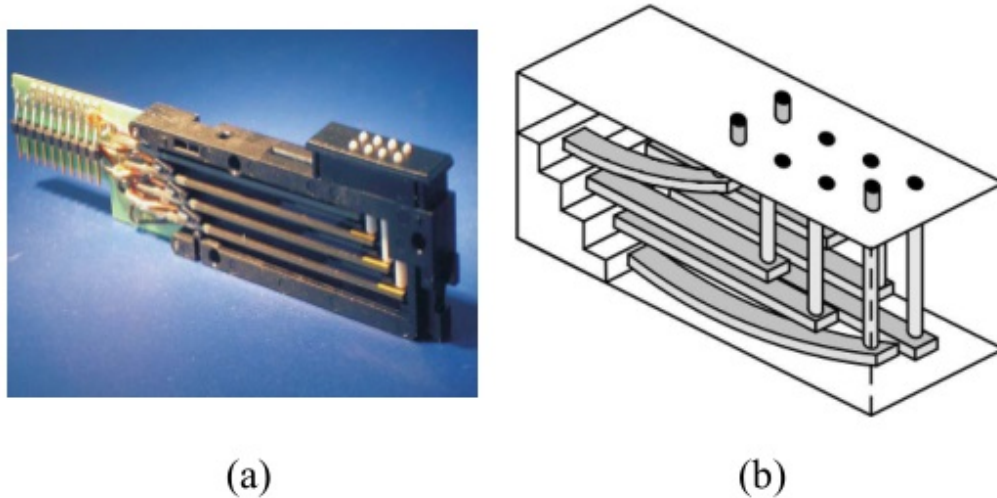


Fig. 2.1: Célula braille da METEC: (a) foto (b) desenho esquemático [1]

2.3 Displays braille disponíveis no Brasil

No Brasil, conseguimos localizar duas empresas que importam e vendem displays braille: Lararama Assoc.Bras.Assistência ao Defic.Visual (Laratec) [6] e a CIVIAM [7]. Obtivemos uma cotação dos displays braille comercializados pela LARATEC:

- Linha Braille Brailiant BI 32 - R\$ 8.500,00
- Linha Braille Brailiant BI 40 - R\$ 10.500,00
- Linha Braille Brailiant B 80 - R\$ 19.500,00

Já a CIVIAM comercializa um modelo da Optelec: Linha Braille Usb 640.

Vamos descrever estes modelos utilizando as informações fornecidas por estas empresas e complementando com algumas informações obtidas nos sites dos fabricantes.

2.3.1 Braille BI 32 - fabricante: Humanware

A linha Braille, ou display Braille, é um dispositivo de saída de computador que exibe dinamicamente em Braille a informação da tela. Consiste em um sistema eletromecânico de celas Braille ligado a uma porta de saída do computador. Cada cela tem uma superfície plana com 8 furos dispostos no formato e tamanho de uma cela Braille padrão e 8 pinos que podem ser levantados ou descidos através dos furos para formar caracteres Braille. Sob o comando do usuário do computador, um leitor de telas (software) transforma os dados exibidos na tela em sinais elétricos que são enviados à linha Braille. O sistema interpreta esses sinais e faz com que cada pino das celas seja levantado através de o furo ou fique escondido para formar assim caracteres Braille que podem ser lidos pelo tato por um deficiente visual.

Características:

- 32 celas Braille de alta qualidade com 32 cursores de toque.
- Sistema de botões e teclas que permite emular teclado QWERTY completo, permitindo contralar inteiramente tanto o computador quanto o leitor de tela diretamente do display.
- 8 teclas de entrada estilo Perkins para digitação em Braille de 6 ou 8 pontos
- 6 teclas de comando dispostas em duas colunas de três, nos extremos esquerdo e direito do display.
- Compatível com leitores de tela NVDA, Jaws e Window Eyes.
- Conexão simples ao PC ou laptop via cabo USB ou Bluetooth.
- Bateria de 20 horas de autonomia com conexão Bluetooth.



Fig. 2.2: Display braille modelo Brailiant BI 32

2.3.2 Braille BI 40 - fabricante: Humanware

Características:

- 40 celas Braille de alta qualidade com 40 cursores de toque.
- Sistema de botões e teclas que permite emular teclado QWERTY completo, permitindo contralar inteiramente tanto o computador quanto o leitor de tela diretamente do display.
- 8 teclas de teclas de entrada estilo Perkins para digitação em Braille de 6 ou 8 pontos
- 6 teclas de comando dispostas em duas colunas de três, nos extremos esquerdo e direito do display.
- Compatível com leitores de tela NVDA, Jaws e Window Eyes.
- Conexão simples ao PC ou laptop via cabo USB ou Bluetooth.
- Bateria de 20 horas de autonomia com conexão Bluetooth.



Fig. 2.3: Display braille modelo Brailiant BI 40

2.3.3 Braille B 80 - fabricante: Humanware

Características:

- 80 celas Braille de alta qualidade com 80 cursores de toque.
- 8 botões de avanço lateral e de linha.
- 6 teclas de comando dispostas em duas colunas de três, nos extremos esquerdo e direito do display.
- Compatível com leitores de tela NVDA, Jaws e Window Eyes.
- Conexão simples ao PC ou laptop via cabo USB ou Bluetooth.
- Bateria de 20 horas de autonomia com conexão Bluetooth.



Fig. 2.4: Display braille modelo Brailiant B 80

2.3.4 Linha Braille Usb 640 - fabricante: Optelec

A Linha Braille USB 640 possui design compacto e teclas fáceis de localizar. Possibilita acesso fácil aos programas garantindo o acesso a qualquer PC ou MAC. Conecte a sua USB640 ao seu PC ou MAC através da porta USB e aprecie o acesso instantâneo em conjunto com os leitores de tela mais populares de mercado. O seu design ergonômico, aumenta as possibilidades de navegação e operação do software de leitura de tela. A ergonomia é desenhada para uma eficiência ideal ao trabalhar com um computador, PDA ou Telefone Celular.

Especificações:

- Conexão através da porta USB (não necessita de pilhas e/ou baterias);
- Funciona em ambiente Windows, MAC e Linux;
- Compatível com os mais populares leitores de tela;
- 40 células braille Optelec;
- 40 cursores com dupla função;
- Cabo USB, CD com driver para leitor de ecrã e manual em português formato PDF;
- Dimensões: 33,8 x 7,5 x 1,8cm;
- Peso: 550 gramas.



Fig. 2.5: Display braille modelo Usb 650

2.4 Custo de displays braille comerciais

Os displays braille disponíveis comercialmente custam entre 1.600 e 15.000 dólares, dependendo principalmente do número de caracteres braille. O site da AFB ("American Foundation for the

Blind") [5] lista 34 modelos diferentes de displays braille. Estes modelos estão listados na tabela 2.1 abaixo:

modelo	fabricante	n. de caracteres	com.	preço (US\$)
ALVA 544 Satellite Traveller	Optelec	44	USB / RS 232	5.195
ALVA 570 Satellite	Optelec	70	USB / RS 232	8.995
ALVA 584 Satellite Pro	Optelec	84	USB / RS 232	9.995
ALVA BC640	Optelec	40	USB / Bluetooth	3.295
ALVA USB 640	Optelec	40	USB	2.995
Braille Voyager 44	Optelec	44	USB / RS 232	4.395
ALVA BC680	Optelec	80	USB / Bluetooth	7.995
Bookworm	Handy Tech EG	8	USB	1.695
Braille Star 40	Handy Tech EG	40	USB / RS 232	5.590
Braille Star 40	Handy Tech EG	40	USB / Bluetooth	5.990
Braille Wave	Handy Tech EG	40	USB / RS 232	5.290
Braille Wave	Handy Tech EG	40	USB / Bluetooth	5.690
Easy Braille	Handy Tech EG	40	USB	3.990
Easy Braille	Handy Tech EG	40	USB / Bluetooth	4.490
Modular Evolution	Handy Tech EG	64	RS 232	9.995
Modular Evolution	Handy Tech EG	88	RS 232	12.995
Braille Star 80	Handy Tech EG	80	USB / RS 232	9.990
BrailleConnect 12	HumanWare	12	Bluetooth	1.995
BrailleConnect 32	HumanWare	32	USB / Bluetooth	3.995
BrailleConnect 40	HumanWare	40	USB / Bluetooth	4.995
Brailliant BI 32	HumanWare	32	USB / Bluetooth	2.595
Brailliant BI 40	HumanWare	40	USB / Bluetooth	2.995
Brailliant B 80	HumanWare	80	USB / Bluetooth	7.985
Braillex EL 2D-80	F. H. Pappenmeier	80	USB / RS 232	14.075
PAC Mate 20 Display	Freedom Scientific	20	USB	1.395
PAC Mate 40 Display	Freedom Scientific	40	USB	2.695
Focus 40 Blue Braille Display	Freedom Scientific	40	USB / Bluetooth	2.795
Focus 80 Blue Braille Display	Freedom Scientific	80	USB / Bluetooth	7.795
Seika Braille Display	Perkins Products	40	USB	2.295
Seika Braille Display	Perkins Products	40	USB / Bluetooth	2.495
Sync Braille (20 Cell)	HIMS, Inc.	20	USB	1.995
Sync Braille (32 Cell)	HIMS, Inc.	32	USB	3.195
VarioPro 64	Baum Retec	64	USB / RS 232	—
VarioPro 80	Baum Retec	80	USB / RS 232	—

Tab. 2.1: Lista de modelos de display braille

Capítulo 3

Construção de um Painel Braille

3.1 O Disco Braille

O caracter braille é constituído por 6 pontos organizados na forma de uma matriz de 3 linhas por 2 colunas.

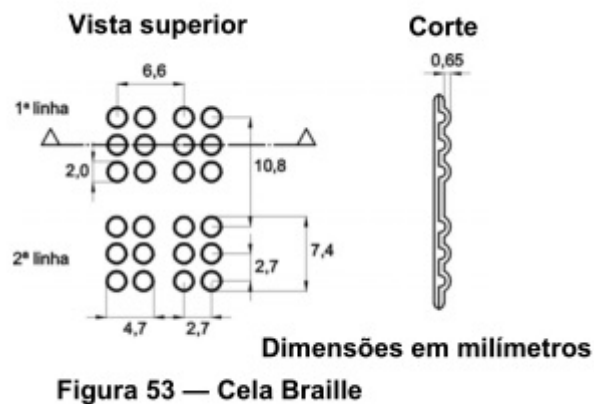


Figura 53 — Cella Braille

Fig. 3.1: Cella braille (desenho extraído da NBR 9050)

O texto abaixo extraído na NBR 9050 [8] ilustra a organização e o tamanho dos pontos de um caracter braille (cela braille):

5.6.1.3 O arranjo de seis pontos e o espaçamento entre as celas Braille, conforme figura 53, devem atender às seguintes condições:

- a) diâmetro do ponto na base: 2 mm;
- b) espaçamento vertical e horizontal entre pontos - medido a partir do centro de um ponto até o centro do próximo ponto: 2,7 mm;
- c) largura da cela Braille: 4,7 mm;
- d) altura da cela Braille: 7,4 mm;
- e) separação horizontal entre as celas Braille: 6,6 mm;

- f) separação vertical entre as celas Braille: 10,8 mm;
- g) altura do ponto: 0,65 mm.

Ao invés de usar pinos móveis para reproduzir o relevo dos pontos de uma célula braille, o que implica na utilização de dispositivo de mecânica fina para cada ponto, nosso projeto baseia-se na idéia de substituir estes pinos por um disco de plástico com oito lados sendo que em cada um dos lados há o relevo de cada uma das oito configurações possíveis de 3 pontos. Desta forma, um caracter braille será constituído por dois discos colocados um ao lado do outro e rotacionados de forma que uma das faces (que contém o padrão do caracter a ser exibido) fique voltada para a superfície do painel onde estes discos estão alojados.

A figura abaixo apresenta uma vista lateral deste disco, na forma em que ele foi originalmente concebido:

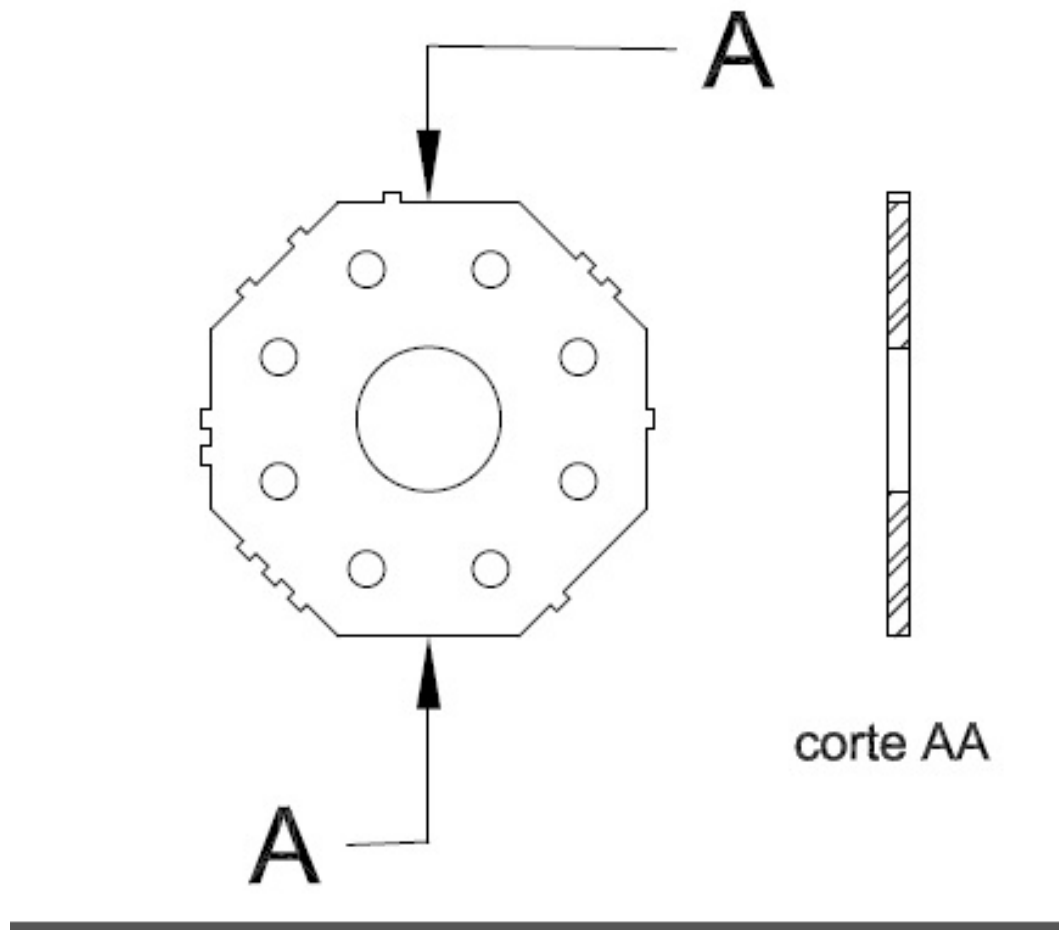


Fig. 3.2: Desenho do disco braille

Já o disco efetivamente utilizado na construção do protótipo do painel é mostrado nas fotos abaixo:

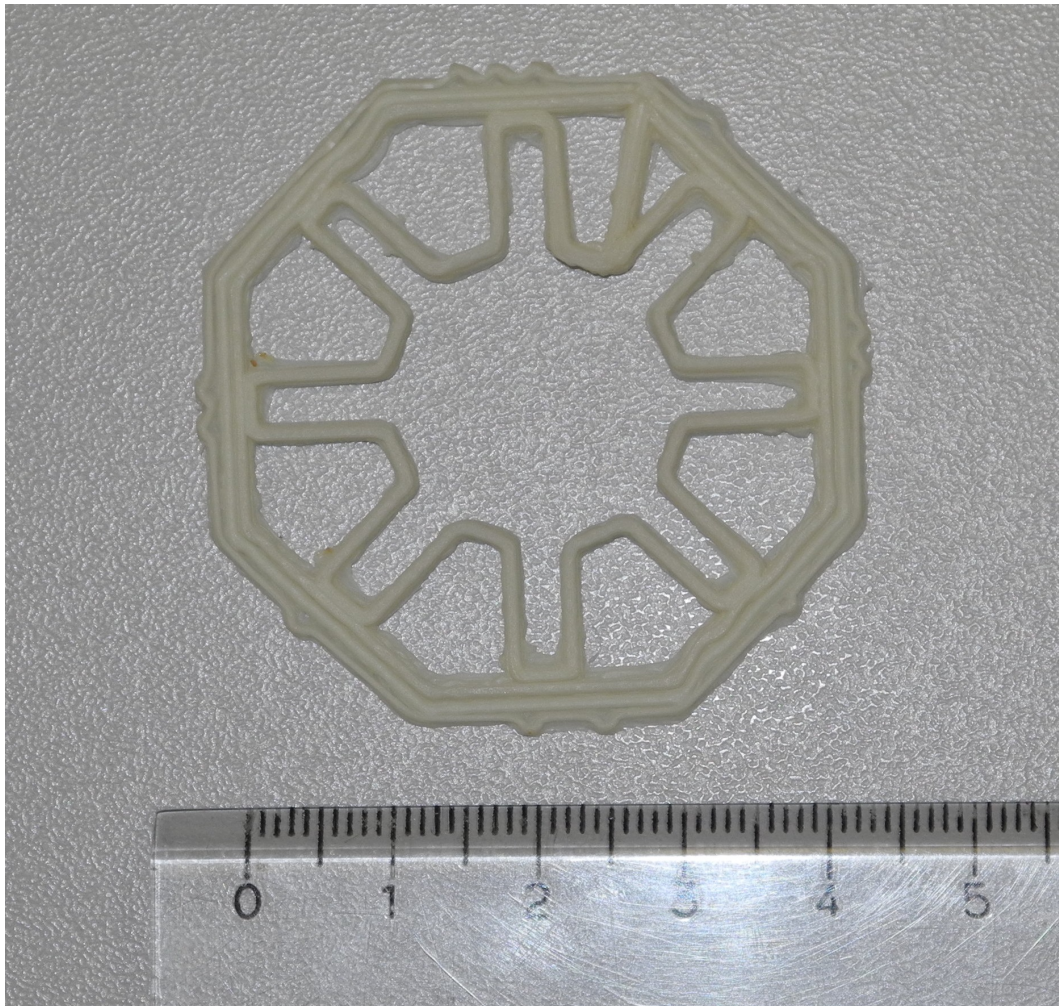


Fig. 3.3: Foto disco braille: frente



Fig. 3.4: Foto disco braille: verso

Como pode ser observado nas fotos, foram introduzidos rasgos no disco, um para cada face, de forma a permitir a navegação da haste por sob os discos (como explicado no próximo ítem).

3.2 O fuso

O protótipo construído consiste de 4 caracteres braille formados por 8 discos de plástico montados sobre um eixo. Sobre este eixo existe uma haste que está montada sobre uma porca que por sua vez está em um fuso. Um motor de passo rotaciona o fuso e a rotação deste provoca a movimentação da haste. A haste é movimentada até se posicionar sob o disco que se deseja rotacionar. Uma vez

posicionada a haste, um segundo motor de passo gira a haste e, conseqüentemente, o disco, até que o padrão correto correspondente ao caracter desejado seja posicionado na superfície do painel.

O eixo foi construído utilizando-se um cano de alumínio comum de 19 mm de diâmetro no qual foi feito um rasgo para a movimentação da haste. O fuso é uma barra rosqueada comum comprada em casas de ferragens. A haste é um parafuso rosqueado em uma rosca feita em uma porca comum.

Segue foto do fuso já com os dois motores de passo montados nas suas extremidades. Um dos motores está acoplado à barra rosqueada para provocar a rotação do fuso e desta forma movimentar a haste. O outro está acoplado ao cano de alumínio de forma a provocar a rotação da haste.



Fig. 3.5: Foto do fuso sem o painel

3.3 O painel

O painel onde os discos são fixados consiste de um retângulo de plástico com aberturas retangulares onde uma das faces do disco se encaixa. As aberturas estão separadas duas a duas por uma distância aproximada da distância que separa dois caracteres braille. A separação entre cada abertura de cada par é equivalente à distância entre os pontos das duas colunas de pontos do caracter braille (conferir a figura 3.1).

Segue abaixo uma foto do painel fabricado para o protótipo com 4 caracteres braille:

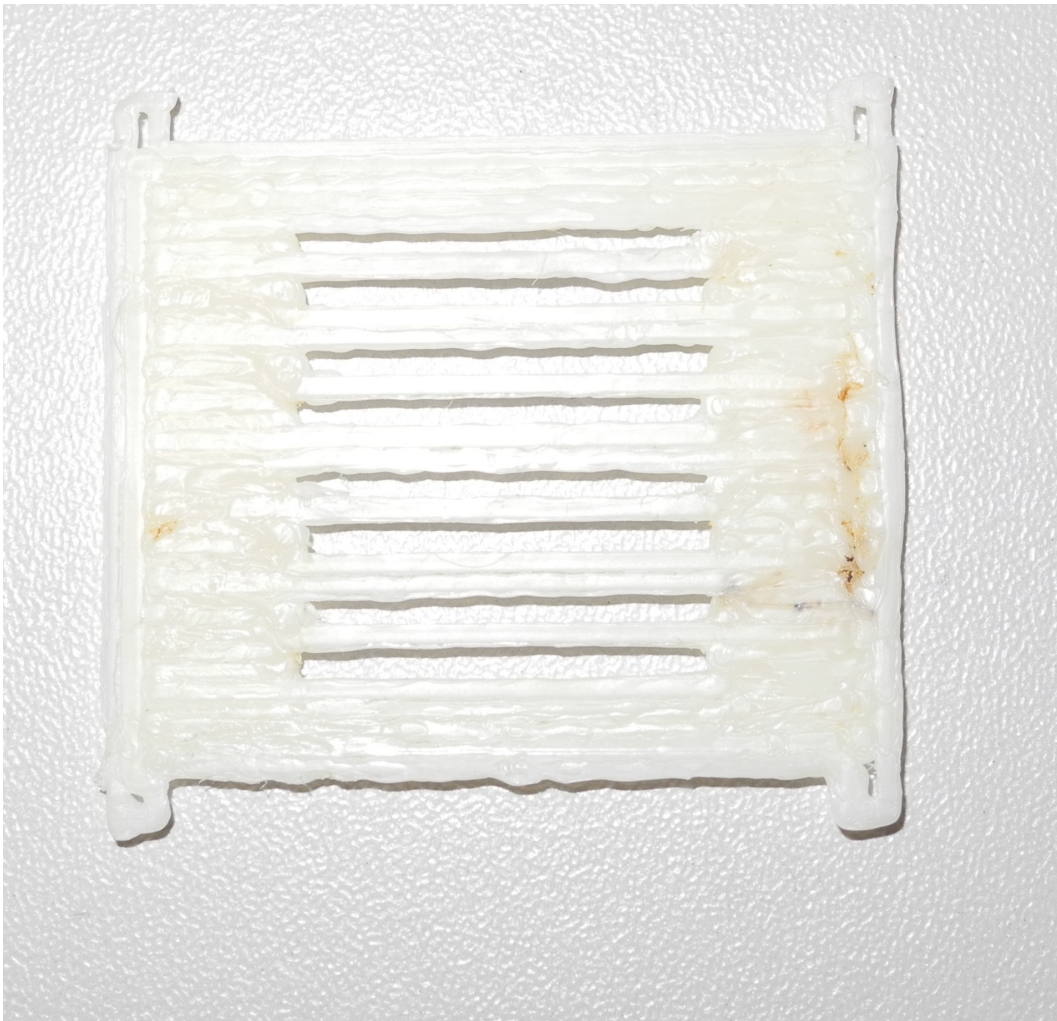


Fig. 3.6: Foto do painel

As abas presentes nas laterais do painel servem para fixar o painel em dois suportes que também são montados sobre o eixo. O mesmo painel é usado na parte de baixo para ajudar na fixação dos discos.

Segue foto do painel (já com os discos) montado sobre o eixo:

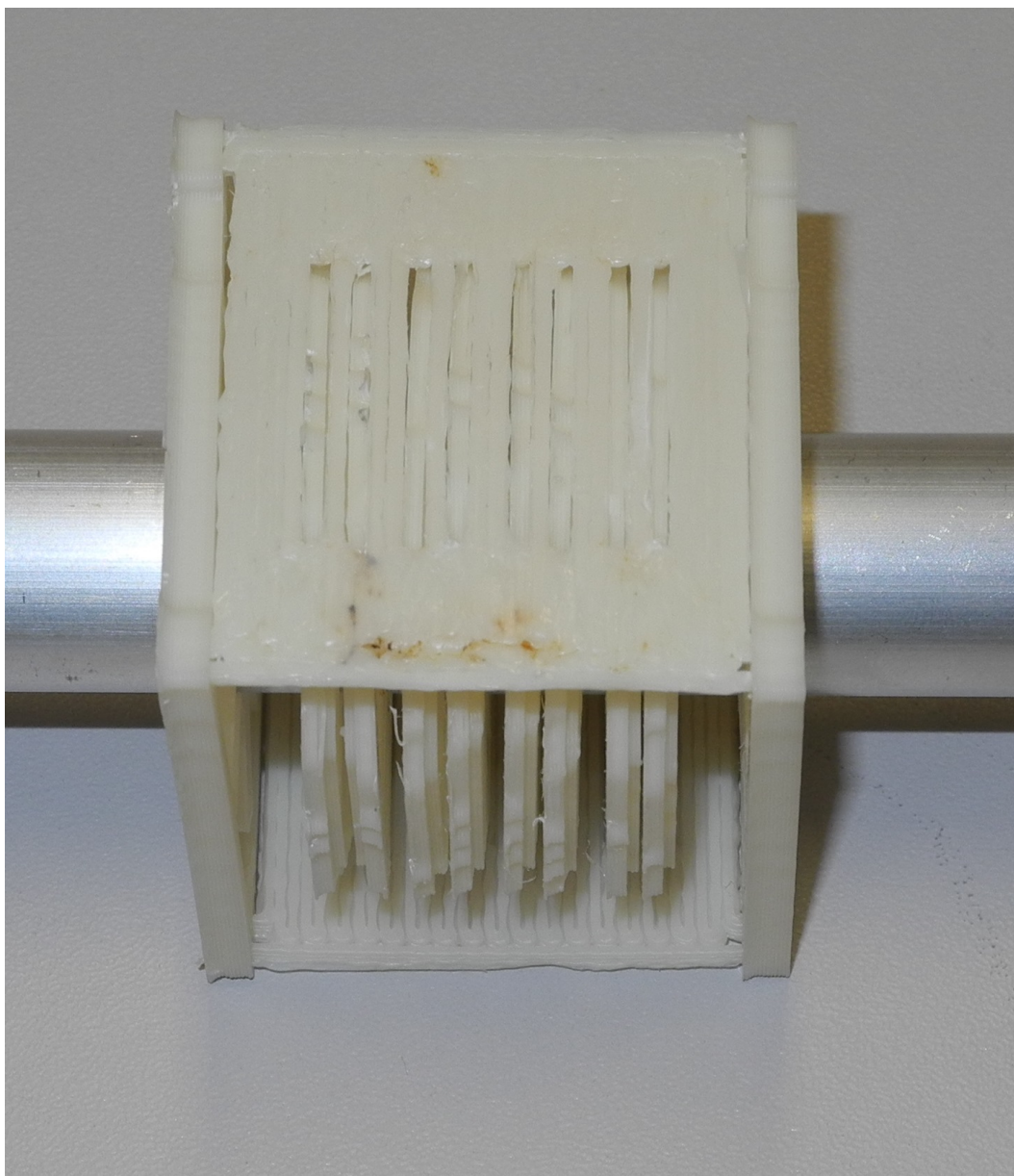


Fig. 3.7: Foto do painel montado

3.4 A produção das peças de plástico

Depois de várias tentativas, sem sucesso, de mandar fabricar o disco braille, optamos pela utilização de uma impressora 3D para produzir os discos e as demais peças do painel braille. A utilização da impressora 3D permitiu chegar ao formato atual dos discos de uma forma gradual.

A impressora 3D utilizada é mostrada na foto abaixo:

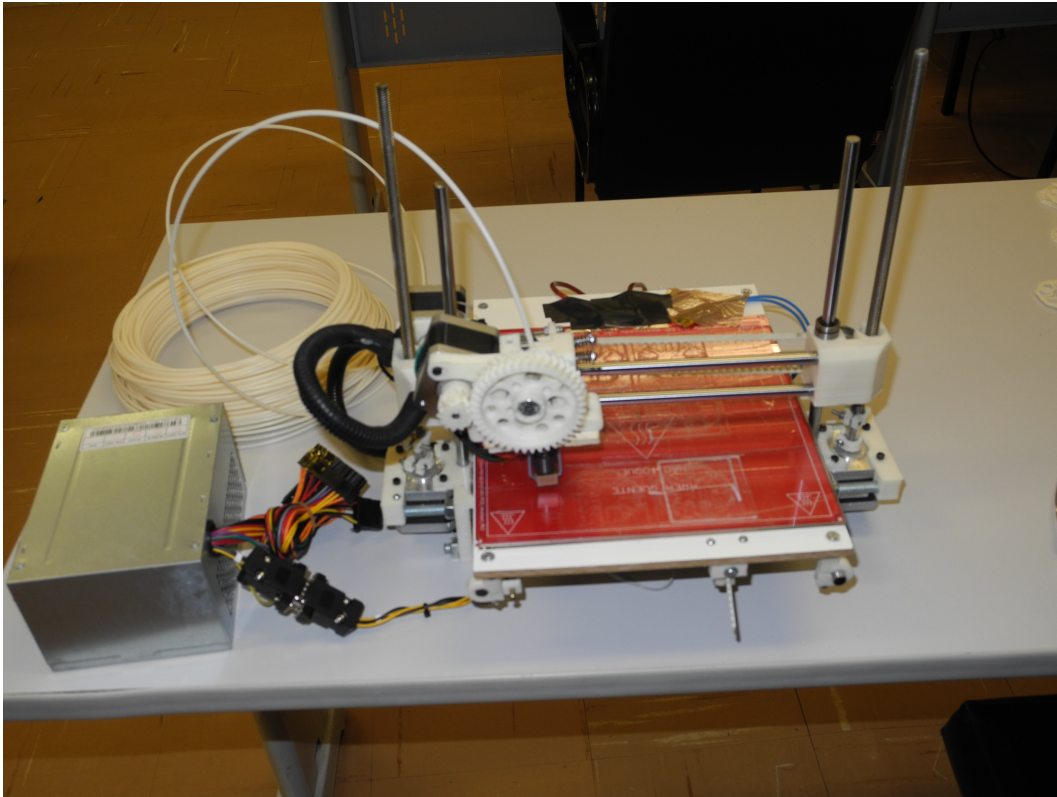


Fig. 3.8: Foto da impressora 3D

Normalmente, a produção de peças em impressoras 3D se dá em quatro etapas:

1. produção de um modelo 3D da peça que se deseja fabricar;
2. exportação do arquivo 3D da peça no formato stl;
3. utilização de um software de fatiamento que, a partir de parâmetros fornecidos da impressora e do fio de ABS utilizado, produz uma sequência de comandos numa linguagem denominada G-CODE [9], que consiste basicamente na descrição dos movimentos que a cabeça de extrusão deve fazer, com a especificação da velocidade e da quantidade de plástico extrudada em cada trecho percorrido pela cabeça de extrusão;
4. utilização de um software de comunicação com a impressora para o envio da sequência de comandos em G-CODE.

No nosso caso, optamos por uma abordagem diferente que consistiu em fazer os desenhos das peças em 2D com a utilização de um software escrito em C para windows e na conversão destes desenhos diretamente na linguagem G-CODE. Segue abaixo a tela deste software mostrando o modelo 2D das peças utilizadas no painel:

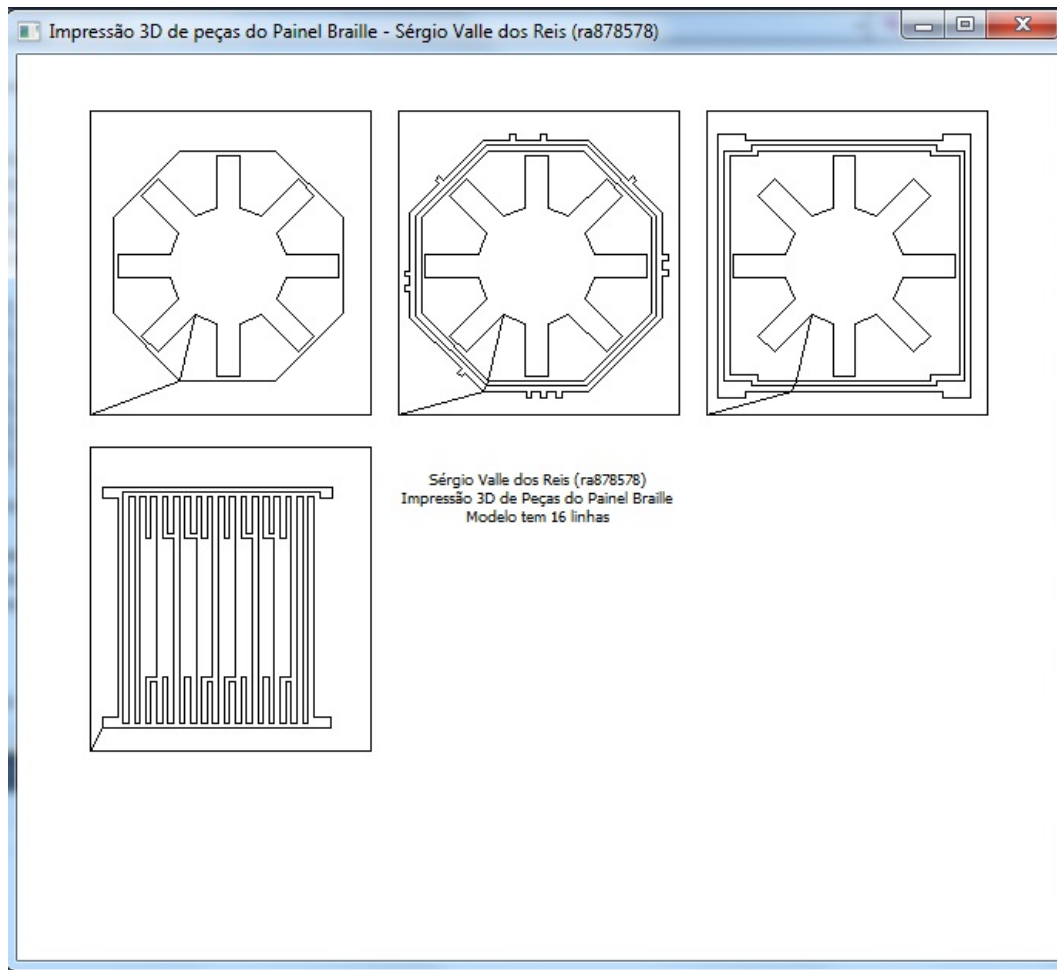


Fig. 3.9: Tela do software utilizado para desenhar as peças de plástico

3.5 A eletrônica de controle do painel

Como nosso objetivo era validar a proposta de utilização do disco braille para a produção de um painel braille de baixo custo, usamos um hardware mínimo e também de baixo custo para controlar o painel e provendo apenas as funcionalidades de controle dos motores de passo para a movimentação dos discos e uma comunicação serial (via cabo usb) com o pc para permitir o controle do painel a partir de comandos enviados pelo pc. O hardware escolhido foi um kit de desenvolvimento da Texas para a linha de microcontroladores MSP430 [10] MSP430 LaunchPad. Este kit liga-se ao pc por uma usb e provê uma comunicação usb-serial para um microcontrolador que é alojado em um soquete no kit. O microcontrolador utilizado foi o MSP430G2553 [11]. Segue uma foto do kit:

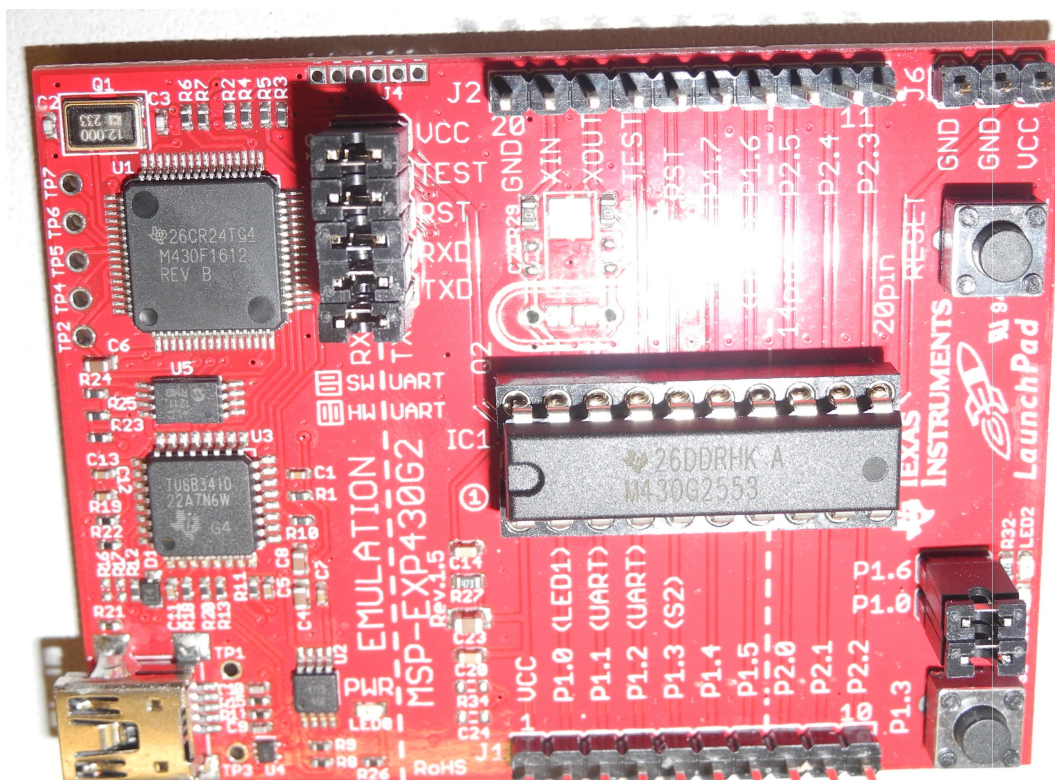


Fig. 3.10: Placa de controle do painel

Além da placa com o microcontrolador, foi utilizada uma placa com 4 CI's LMD18200 [12] que é uma ponte H completa. São usados dois destes CI'S para cada motor de passo pois os motores de passo utilizados são bipolares. Estes CI's recebem três sinais de controle que são fornecidos diretamente pelas portas de i/o do microcontrolador sem necessidade de nenhuma eletrônica adicional. Segue uma foto da placa com os drivers:

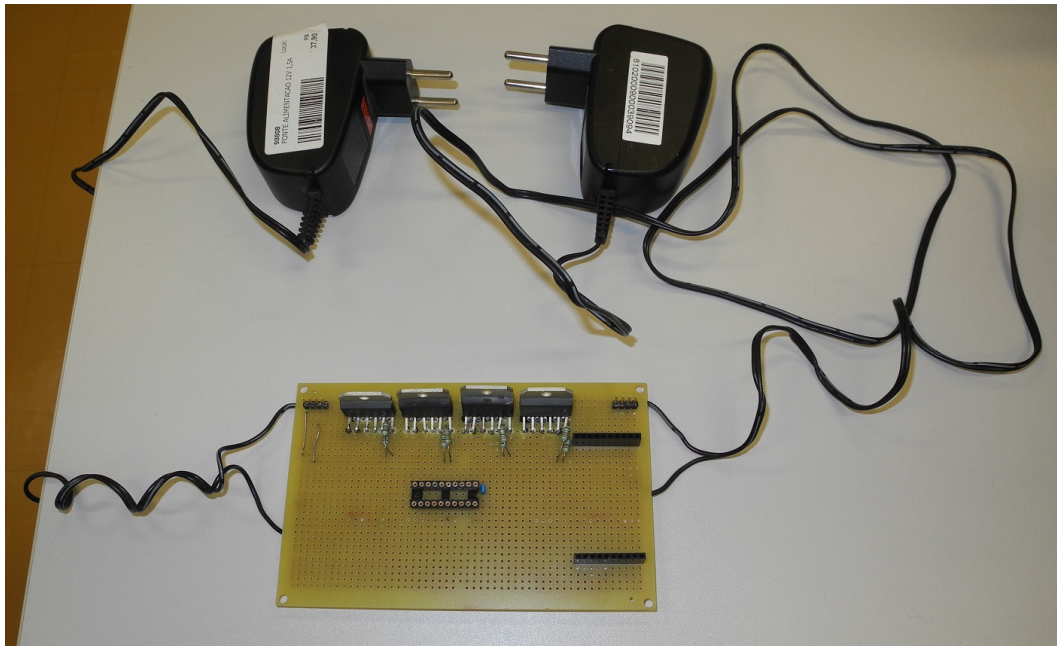


Fig. 3.11: Placa de acionamento dos motores de passo

Segue uma foto da placa microcontrolada montada sobre a placa de acionamento:

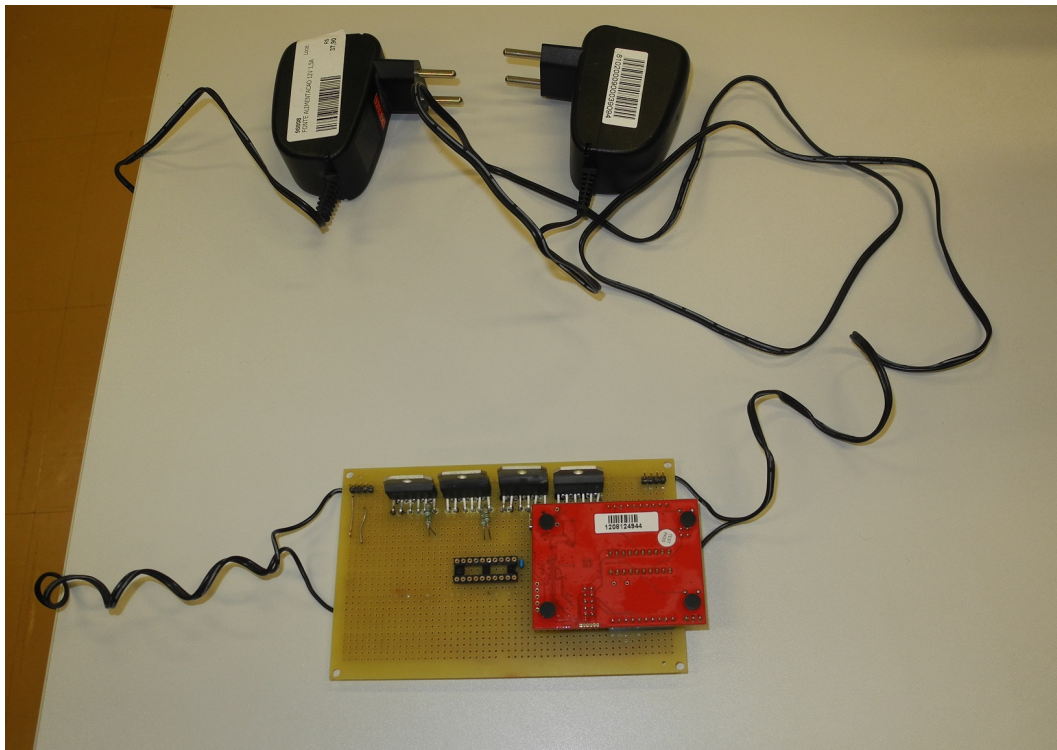


Fig. 3.12: Placa de controle montada sobre placa de acionamento

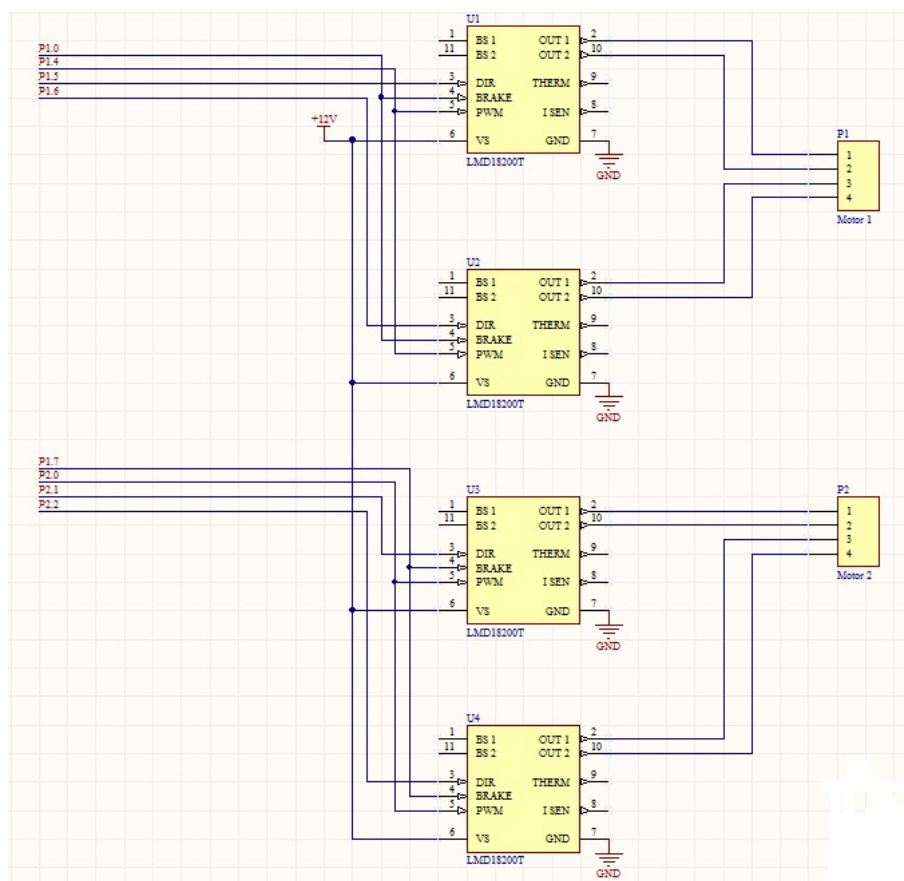


Fig. 3.13: Esquemático da placa de acionamento

3.5.1 Descrição da placa de acionamento

A figura 3.13 apresenta o esquemático da placa de acionamento.

A placa de acionamento é utilizada para movimentar dois motores de passo bipolares. Cada motor é controlado por dois pares de fio. Estes pares são ligados nos pinos 1 e 2, e 3 e 4 de P1 para o motor 1 (que controla a translação da haste) e nos pinos 1 e 2, e 3 e 4 de P2 para o motor 2 (que controla a rotação da haste).

Os sinais P1.0 (que está ligado nos sinais BRAKE de U1 e U2) e P1.4 (que está ligado nos sinais PWM de U1 e U2) controlam a energização ou não do motor 1:

P1.0 = 1 e P1.4 = 0 ==> motor 1 desenergizado

P1.0 = 0 e P1.4 = 1 ==> motor 1 energizado

O sinal P1.5 (que está ligado no sinal DIR de U1) controla a direção da corrente que passa no enrolamento do motor ligado nos pinos 1 e 2 de P1 (quando o motor está energizado):

P1.5 = 0 ==> a corrente flui do pino 2 para o pino 1

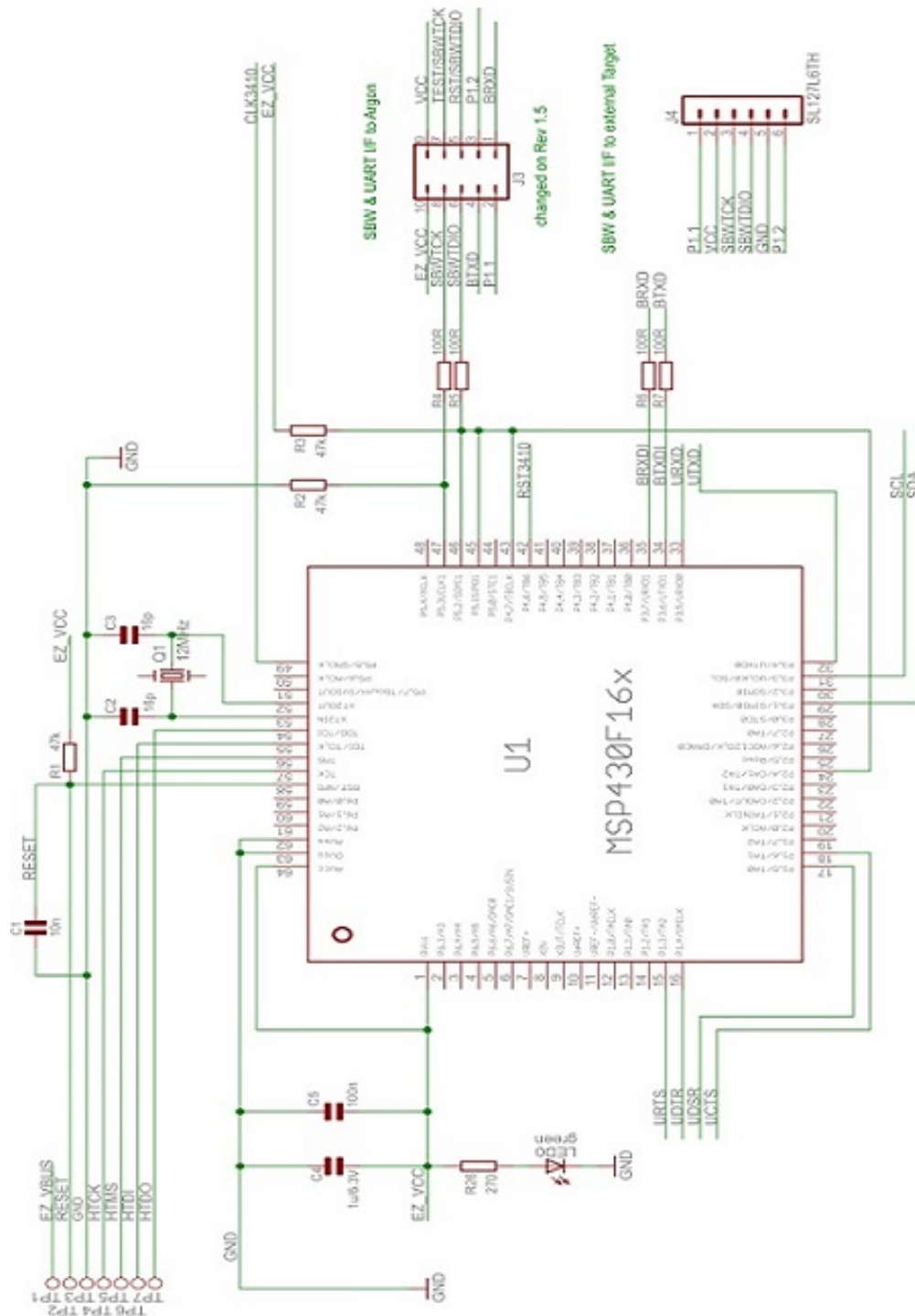


Fig. 3.14: Esquemático do MSP430 LaunchPad (parte 1 de 3)

$P_{1.5} = 1 \implies$ a corrente flui do pino 1 para o pino 2

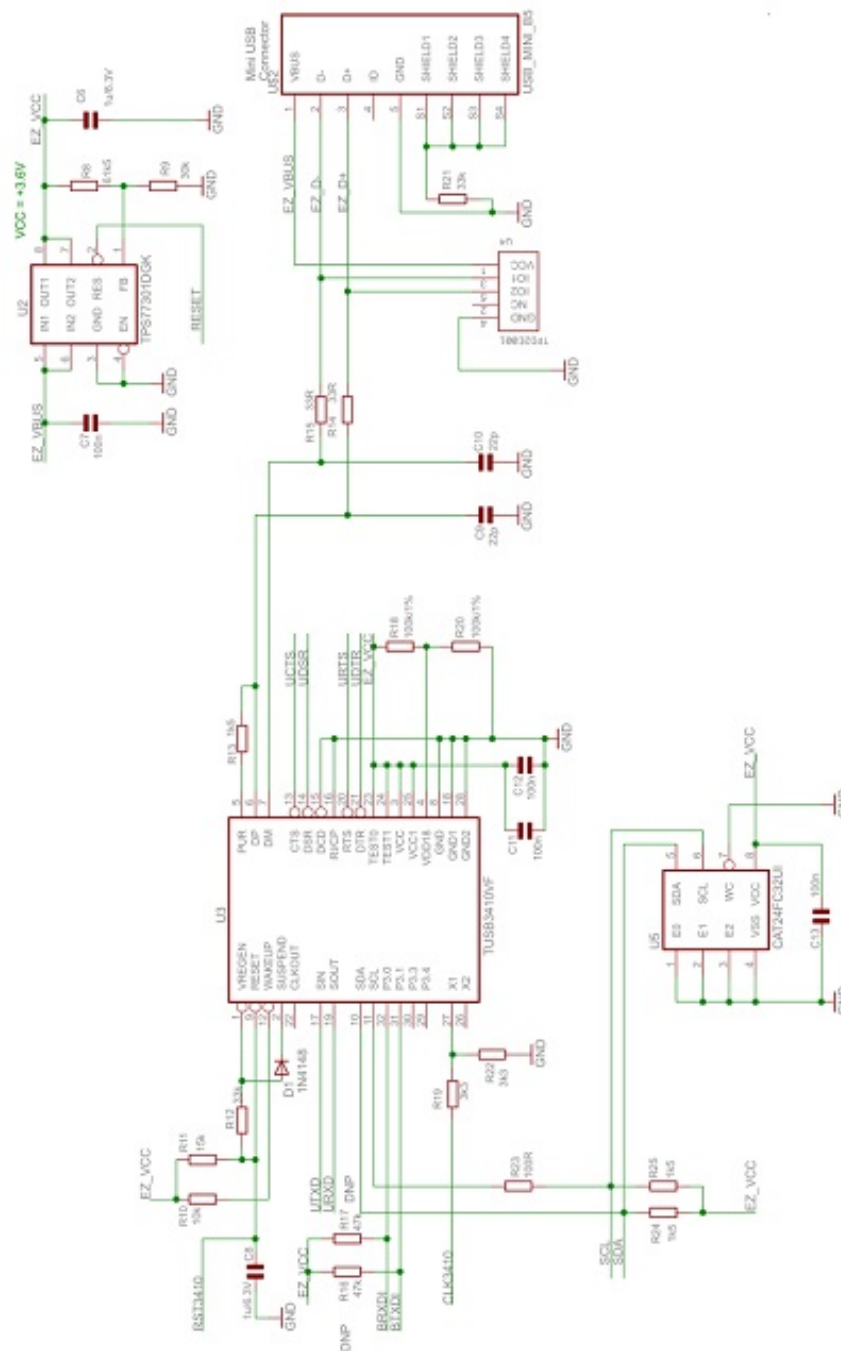


Fig. 3.15: Esquemático do MSP430 LaunchPad (parte 2 de 3)

O sinal P1.6 (que está ligado no sinal DIR de U2) controla a direção da corrente que passa no enrolamento do motor ligado nos pinos 3 e 4 de P1 (quando o motor está energizado):

$P_{1.6} = 0 \implies$ a corrente flui do pino 4 para o pino 3

P1.6 = 1 ==> a corrente flui do pino 3 para o pino 4

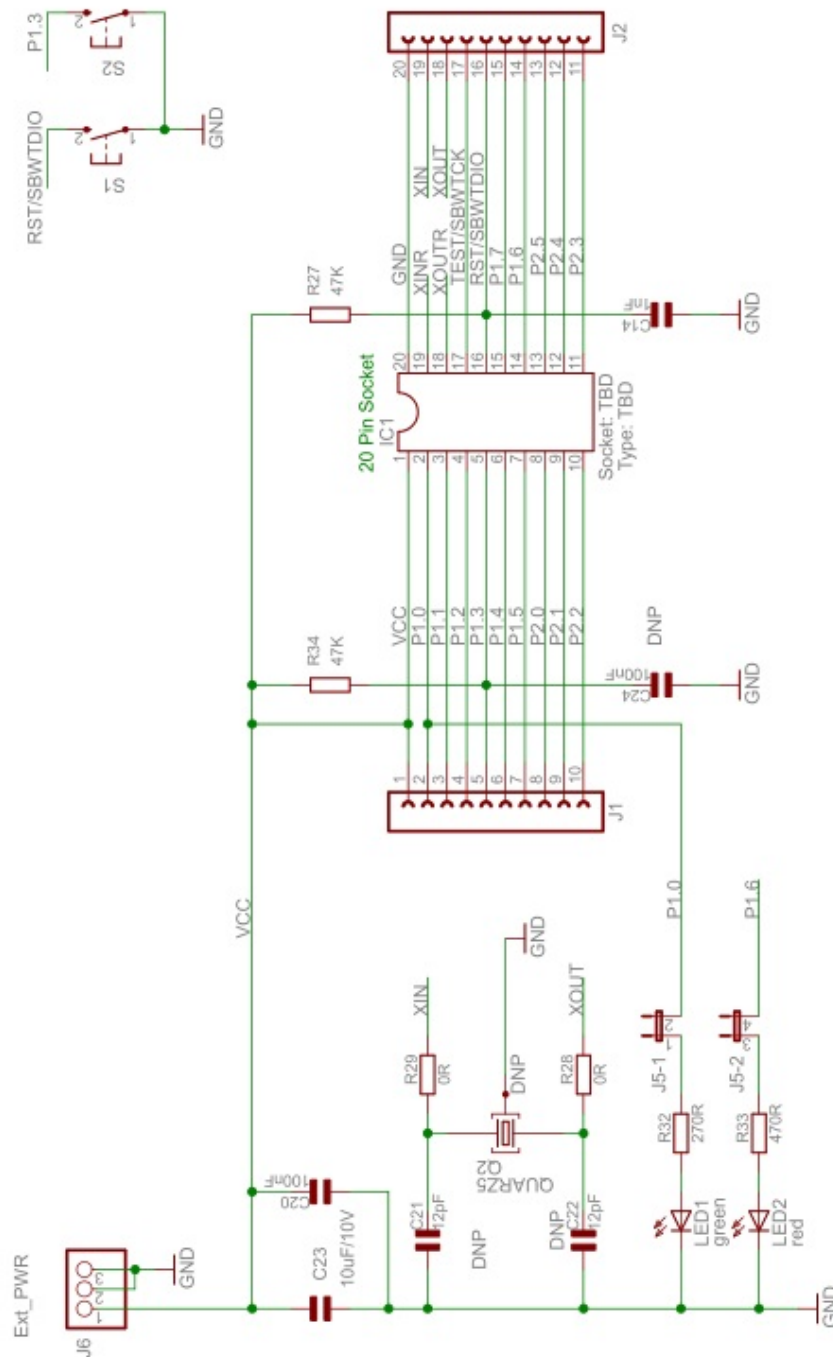


Fig. 3.16: Esquemático do MSP430 LaunchPad (target socket)(parte 3 de 3)

Os sinais P1.7 (que está ligado nos sinais BRAKE de U3 e U4) e P2.0 (que está ligado nos sinais PWM de U3 e U4) controlam a energização ou não do motor 2:

P1.7 = 1 e P2.0 = 0 ==> motor 2 desenergizado

$P1.7 = 0$ e $P2.0 = 1 \implies$ motor 2 energizado

O sinal P2.1 (que está ligado no sinal DIR de U3) controla a direção da corrente que passa no enrolamento do motor ligado nos pinos 1 e 2 de P2 (quando o motor está energizado):

$P2.1 = 0 \implies$ a corrente flui do pino 2 para o pino 1

$P2.1 = 1 \implies$ a corrente flui do pino 1 para o pino 2

O sinal P2.2 (que está ligado no sinal DIR de U4) controla a direção da corrente que passa no enrolamento do motor ligado nos pinos 3 e 4 de P2 (quando o motor está energizado):

$P2.1 = 0 \implies$ a corrente flui do pino 4 para o pino 3

$P2.1 = 1 \implies$ a corrente flui do pino 3 para o pino 4

3.5.2 Descrição do MSP430 LaunchPad

As figuras 3.14, 3.15 e 3.16 apresentam o esquemático do MSP430 LaunchPad.

Esta placa funciona como um sistema de desenvolvimento, permitindo a carga, execução e depuração de programas em microcontroladores da linha MSP430 que são instalados no sockete IC1 (fig. 3.16).

Além de funcionar como sistema de desenvolvimento, esta placa provê para o processador no sockete IC1 a alimentação de 3.3V via o regulador U2 (fig. 3.15) que por sua vez é alimentado pelo 5V da conexão USB. Também provê o sinal de reset.

Ela também provê comunicação serial (nos pinos P1.1 e P1.2, ver fig. 3.13). O CI U3 (fig. 3.15) provê a conversão de USB para serial. Desta forma, o microcontrolador instalado no sockete IC1 pode se comunicar serialmente com um PC conectado via um cabo USB no MSP430 LaunchPad.

O microcontrolador U1 (fig. 3.14) implementa as funções de debug e carga de programa no microcontrolador instalado no sockete IC1. Em tempo de execução do aplicativo deste microcontrolador, o microcontrolador U1 não interfere no seu funcionamento.

Capítulo 4

Testes funcionais e conclusão

4.1 Desempenho do Painei

O conjunto de placa de controle/placa de acionamento (ver figura 3.12) foi conectado aos dois motores de passo e ao PC através de um cabo USB.

Para testar o protótipo do painel, foi feito um software para o microcontrolador MSP430G2553 em C que:

- fica aguardando a recepção de um comando pela serial constituído de 2 ou 3 bytes:

t n
+ n p ou
- n p ou

O primeiro caracter (+ - ou **t**) é o comando que pode ser:

- + movimentação do motor de passo no sentido horário
- movimentação do motor de passo no sentido anti-horário
- t** programação do intervalo de tempo entre dois passos (em múltiplos de 10 milissegundos)

No caso dos comandos + ou -, o segundo byte indica o motor a ser movimentado

- =00 -> motor que gira o fuso provocando a translação da haste ou
- =01 -> motor que gira a haste

e o terceiro byte indica o número de passos a serem executados (de 1(=00) a 256(=ff))

No caso do comando **t** o segundo byte representa o intervalo de tempo entre dois passos, ou seja, se o segundo byte for **n**, o intervalo de tempo entre dois passos será **n*10** milissegundos.

Foi criado um programa em C para Windows que através do envio pela serial do computador destes três comandos fornecia as seguintes funcionalidades:

- movimentação dos dois motores no sentido horário e anti-horário de um determinado número de passos;
- estabelecimento de um ponto de partida ("uma origem") para a movimentação de cada motor;

- estabelecimento das posições de cada um dos discos em relação à translação da haste;
- estabelecimento de que face de cada disco está voltada para o painel de leitura no início do funcionamento do painel;
- movimentação dos motores de forma a posicionar a haste na origem (comando "home").

As funcionalidades acima permitiram a "parametrização" do software em C para permitir que a partir da digitação de 4 letras, o software enviasse a sequência de comandos correta para movimentar os discos de forma a apresentar os caracteres braille correspondentes.

Nos testes realizados, cada sequência de 4 caracteres braille enviados pelo computador foi corretamente exibida pelo painel. A exibição de cada sequência durou, em média, cerca de 2 segundos.

4.2 Contribuições

A principal contribuição deste trabalho é propor uma forma de construir um display braille significativamente mais barato que os existentes no mercado. Além disso, como pode ser observado na tabela 2.1 (seção 2.4), o preço dos displays braille existentes é diretamente proporcional ao número de caracteres braille do display. O mesmo não acontece com o display braille construído com a utilização dos "discos braille" em que o custo dos caracteres vai representar menos de 5% do total do custo do equipamento. Isto permitirá a construção, por exemplo, de painéis braille com várias linhas de 80 caracteres que ainda serão muito mais baratos que os modelos mais baratos de display braille comerciais e que são aqueles com menos caracteres.

A principal desvantagem da utilização dos "discos braille" é que o tempo que é necessário para exibir uma linha de caracteres é entre 10 e 20 vezes maior que no caso dos displays comerciais. Entretanto, se considerarmos a utilização de várias linhas, esta desvantagem desaparece para as aplicações mais importantes tal como a leitura de um livro ou de informações coletadas da internet, por exemplo. Nestes casos, o tempo total que será gasto na leitura de uma página será na verdade menor que no caso dos displays comerciais porque até o usuário do painel terminar de "ler" a primeira linha, as demais já terão sido montadas.

Uma outra desvantagem do método proposto em relação aos displays comerciais é que o painel braille construído com a utilização dos "discos braille" será bem mais volumoso que os displays atuais e de maneira alguma poderá ser considerado como sendo um equipamento portátil. Entretanto, não é a intenção deste trabalho propor a criação de displays braille portáteis mas sim equipamentos de mesa para serem usados em casa, no trabalho e em escolas.

É nossa crença que a construção de painéis braille de várias linhas e a um custo acessível permitirá a popularização no Brasil da utilização de displays brilles e tem potencial para alterar os paradigmas atualmente existentes em relação à apresentação dinâmica de informações em braille e que tem como ponto de partida a utilização de displays com poucos caracteres.

Uma contribuição secundária deste trabalho foi o desenvolvimento de uma metodologia de fabricação de peças em impressoras 3D a partir de um desenho em duas dimensões. Esta metodologia permite maior controle sobre a geometria das peças produzidas na impressora além de prescindir da elaboração de modelos 3D das peças a serem impressas.

4.3 Conclusões

A metodologia proposta para a fabricação de um painel braille de baixo custo pode ser considerada aprovada. O tempo previsto para a exibição de uma linha completa de 40 caracteres é de cerca de 20 segundos. Este tempo, embora grande, não nos parece ser impeditivo para a utilização do painel como uma linha braille.

A proposta mais ambiciosa de utilizar esta metodologia para produzir um livro eletrônico em braille requererá alguma forma de diminuir este tempo. Uma forma de viabilizar a utilização do painel braille como livro eletrônico seria a utilização de por exemplo 4 linhas de 40 caracteres e mostrar 4 linhas de texto ao mesmo tempo. Usando algum mecanismo de detecção do fato de uma linha já ter sido lida, esta poderia ser configurada com o conteúdo da próxima linha enquanto o leitor lê as outras linhas. Desta forma, após a apresentação das primeiras 4 linhas, a leitura poderia prosseguir sem que o leitor tenha que esperar pela apresentação de cada linha.

Capítulo 5

Trabalhos Futuros

Pretendemos continuar o desenvolvimento do painel braille e, para tanto, prevemos a realização dos seguintes trabalhos:

- montagem de uma linha completa de 40 caracteres;
- verificar a possibilidade de produzir o fuso na impressora 3D, e, conseqüentemente, usar motores de passo de menor potência que serão menores e mais baratos;
- usar um fuso com um passo de rosca maior para aumentar a velocidade de translação da haste;
- fazer a interface da linha completa de 40 caracteres com o DOSVOX de forma a testar a utilização do painel como uma linha braille;
- fazer testes com indivíduos que leiam em braille para validar a utilização do painel.

Durante os testes com o painel de 40 caracteres esperamos determinar se os discos braille fabricados na impressora 3D vão ser aprovados pelos leitores em braille.

Devido às limitações de precisão da impressora 3D talvez seja necessário mudar o método de fabricação destes discos e eles tenham que ser, por exemplo, usinados.

De qualquer forma, se o painel braille chegar a ser fabricado em maior escala, a idéia é fabricar os discos de plástico injetado.

Referências Bibliográficas

- [1] S. M. Saad, F. Razaly, M. Z. M. Zain, M. Hussein, M. S. Yaacob, A. R. Musa, and M. Y. Abdullah, “Development of piezoelectric braille cell control system using microcontroller unit (mcu),” *WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS*, 2010.
- [2] M. T. Schotten, L. Cohen, E. Amemiya, L. W. Braga, and S. Dehaene, “Learning to read improves the structure of the arcuate fasciculus.” *Cerebral Cortex*, 2012.
- [3] Wikipedia, “Refreshable braille display,” 2013. [Online]. Available: http://en.wikipedia.org/wiki/Refreshable_braille_display
- [4] M. AG, “Braille cells,” 2013. [Online]. Available: <http://web.metec-ag.de/braille%20cells.html>
- [5] A. F. for the Blind, “Refreshable braille displays,” 2013. [Online]. Available: <http://www.afb.org/ProdBrowseCatResults.asp?CatID=43>
- [6] LARATEC, “Título desconhecido,” 2013. [Online]. Available: <http://laramara.org.br/unidades-de-negocio/laratec>
- [7] CIVIAM, “Equipamentos para cegos,” 2013. [Online]. Available: <http://www.civiam.com.br/civiam/index.php/necessidadesespeciais/cegos/equipamentos.html>
- [8] ABNT NBR 9050: *Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos*, 2004.
- [9] “G-code.” [Online]. Available: <http://reprap.org/wiki/G-code>
- [10] “Get started with the launchpad evaluation platform from texas instruments.” [Online]. Available: http://www.ti.com/ww/en/launchpad/msp430_head.html
- [11] “Msp430g2x53 mixed signal microcontroller.” [Online]. Available: <http://www.ti.com/lit/ds/symlink/msp430g2553.pdf>
- [12] “Lmd18200, 3a, 55v, h-bridge.” [Online]. Available: <http://www.ti.com/lit/ds/symlink/lmd18200.pdf>

Anexo A

Listagem do Programa em C

Segue a listagem do programa utilizado para desenhar o disco braille:

```
1  #include <windows.h>
2  #include <iio.h>
3  #include <font1.h>
4  #include <stdio.h>
5  #include <dos.h>
6  #include <conio.h>
7
8  #define _USE_MATH_DEFINES
9
10 // Disable warning messages 4996.
11 #pragma warning( disable : 4996)
12
13 // acentos
14 // Å = \300 Ê = \310 Æ = \340 è = \350 fi = \370
15 // Ä = \301 Ë = \311 Ñ = \321 Û = \331 á = \341 é = \351 ã = \361 ù = \371
16 // Å = \302 Ê = \312 Ö = \322 Û = \332 ä = \342 ê = \352 ò = \362 ú = \372
17 // Å = \303 Ë = \313 Ó = \323 Û = \333 ã = \343 ë = \353 ó = \363 û = \373
18 // Å = \304 Ì = \314 Ô = \324 Û = \334 ä = \344 ì = \354 ô = \364 ü = \374
19 // Ì = \315 Õ = \325 Ì = \355 Õ = \365 ý = \375
20 // Ì = \316 Õ = \326 Ì = \356 Õ = \366
21 // Ç = \307 Ì = \317 beta = \337 ç = \347 ì = \357 sinal de dividir = \367 ÿ = \377
22
23 #define S_IREAD 0000400 /* read permission, owner */
24 #define S_IWRITE 0000200 /* write permission, owner */
25
26 #include <math.h>
27
28 double calculaDistancia (int, int, int, int);
29
30 // paginas de help consultadas
31 // http://msdn.microsoft.com/en-us/library/windows/desktop/dd743680%28v=vs.85%29.aspx (PlaySound, requer Winmm.lib)
32 //
33
34 HFONT hf;
35
36 // disco braille tem oitos lados e oito vértices
37 int vet_x[8];
38 int vet_y[8];
39 int vet_x2[8];
40 int vet_y2[8];
41
42 struct COORD
43 {
44     int x;
45     int y;
46 };
47
48 struct REGIAO
49 {
50     int x1;
51     int y1;
52     int x2;
53     int y2;
54     int x3;
55     int y3;
56     int x4;
57     int y4;
58 };
```

```

59
60 struct MOVIMENTO
61 {
62     int nLinha;
63     double x;
64     double y;
65     double z;
66     double e;
67     double f;
68     BOOL comExtrusao;
69     double razao;
70 };
71
72 #define RAO_DO_FIO 15
73
74 #define N_MAX_MOVIMENTOS 40000
75
76 struct MOVIMENTO impressao[N_MAX_MOVIMENTOS];
77
78 int indiceImpressao;
79 int indiceInicialDesenho;
80 int indiceFase2Desenho;
81 int indiceFinalDesenho;
82 int origemXImpressora3D;
83 int origemYImpressora3D;
84 int origemXDesenho;
85 int origemYDesenho;
86 double feedRateSemExtrusao;
87 double feedRateComExtrusao;
88 double feedRateComExtrusaoDeSuperficies;
89 double feedRateComExtrusaoInfill;
90 double razaoExtrusao;
91
92 struct REGIAO regioes[8];
93
94 #define N_MAX_RAZOES 10
95
96 int n_razoes_utilizadas;
97 int linha_nova_razao[N_MAX_RAZOES];
98 double razao_de_extrusao[N_MAX_RAZOES];
99 double distancia_percorrida[N_MAX_RAZOES];
100 double comprimento_extrudado[N_MAX_RAZOES];
101 double camada_z[N_MAX_RAZOES];
102 double feed_rate[N_MAX_RAZOES];
103
104 double total_razao_vezes_distancia;
105 double total_distancia_percorrida;
106
107 double menorRazao, maiorRazao;
108
109 int nLinhasDoModelo;
110
111 #define TAM_MAX_MENSAGEM_CENTRAL 4000
112 char mensagem_central[TAM_MAX_MENSAGEM_CENTRAL];
113
114 #define TAM_MAX_INICIO_DESENHO 4000
115 #define TAM_MAX_FINAL_DESENHO 4000
116
117 char inicio_desenho[TAM_MAX_INICIO_DESENHO];
118 char final_desenho[TAM_MAX_FINAL_DESENHO];
119
120 #define TAMANHO_DO_PONTO 100
121
122 double deltaCoordP;
123
124
125 BOOL jaGravouDesenhos=FALSE;
126 BOOL ehSuporte=FALSE;
127
128 #define LARGURA_SUORTE 200
129 #define COMPRIMENTO_SUORTE 500
130
131 int largura_suporte;
132 int comprimento_suporte;
133
134 LRESULT CALLBACK WndProc (HWND, UINT, WPARAM, LPARAM) ;
135
136 BOOL temCoordenada (char *lin, char id, double *valor)
137 {
138     while (*lin)
139     {
140         if (*lin==id)
141         {
142             sscanf (lin+1, "%lf", valor);
143             return (TRUE);
144         }
145         ++lin;
146     }
147     return (FALSE);
148 }
149

```

```

150
151 void progImpressao (double x, double y, double z, double e, double f, BOOL comExtrusao)
152 {
153     if (indiceImpressao==N_MAX_MOVIMENTOS) return;
154
155     impressao[indiceImpressao].nLinha=0;
156     impressao[indiceImpressao].comExtrusao=comExtrusao;
157     impressao[indiceImpressao].razao=razaoExtrusao;
158
159     if (x!=-1)
160         impressao[indiceImpressao].x=x;
161     else
162         impressao[indiceImpressao].x=impressao[indiceImpressao-1].x;
163
164     if (y!=-1)
165         impressao[indiceImpressao].y=y;
166     else
167         impressao[indiceImpressao].y=impressao[indiceImpressao-1].y;
168
169     if (z!=-1)
170         impressao[indiceImpressao].z=z;
171     else
172         impressao[indiceImpressao].z=impressao[indiceImpressao-1].z;
173
174     if (e!=-1)
175         impressao[indiceImpressao].e=e;
176     else
177         impressao[indiceImpressao].e=impressao[indiceImpressao-1].e;
178
179     if (f!=-1)
180         impressao[indiceImpressao].f=f;
181     else
182         impressao[indiceImpressao].f=impressao[indiceImpressao-1].f;
183
184     ++indiceImpressao;
185 }
186
187 BOOL interpretaGCode (char *lin, int nLin, int *indiceM)
188 {
189     int g_code, param;
190     BOOL algumaCoordenada;
191     double novoValor;
192     char valor_por_extenso[20];
193
194     if (*lin=='\0') return (FALSE);
195     if (strlen(lin)==0) return (FALSE);
196
197     if (*lin=='G')
198     {
199         sscanf (lin+1, "%d", &g_code);
200         switch (g_code)
201         {
202             case 21:
203             case 90: return (FALSE);
204
205             case 28: impressao[*indiceM].nLinha=nLin;
206                     algumaCoordenada=FALSE;
207                     if (*indiceM)
208                     {
209                         impressao[*indiceM].x=impressao[*indiceM-1].x;
210                         impressao[*indiceM].y=impressao[*indiceM-1].y;
211                         impressao[*indiceM].z=impressao[*indiceM-1].z;
212                         impressao[*indiceM].e=impressao[*indiceM-1].e;
213                         impressao[*indiceM].f=impressao[*indiceM-1].f;
214                     }
215                     else
216                     {
217                         impressao[*indiceM].x=0;
218                         impressao[*indiceM].y=0;
219                         impressao[*indiceM].z=0;
220                         impressao[*indiceM].e=0;
221                         impressao[*indiceM].f=0;
222                     }
223                     if (temCoordenada (lin, 'X', &novoValor))
224                     {
225                         impressao[*indiceM].x=novoValor;
226                         algumaCoordenada=TRUE;
227                     }
228                     if (temCoordenada (lin, 'Y', &novoValor))
229                     {
230                         impressao[*indiceM].y=novoValor;
231                         algumaCoordenada=TRUE;
232                     }
233                     if (temCoordenada (lin, 'Z', &novoValor))
234                     {
235                         impressao[*indiceM].z=novoValor;
236                         algumaCoordenada=TRUE;
237                     }
238                     if (!algumaCoordenada)
239                     {
240                         impressao[*indiceM].x=0;

```



```

241     impressao[*indiceM].y=0;
242     impressao[*indiceM].z=0;
243     impressao[*indiceM].e=0;
244     impressao[*indiceM].f=0;
245 }
246 ++(*indiceM);
247 return (TRUE);
248
249 case 92:
250 case 1: impressao[*indiceM].nLinha=nLin;
251         algumaCoordenada=FALSE;
252         if (*indiceM)
253         {
254             impressao[*indiceM].x=impressao[*indiceM-1].x;
255             impressao[*indiceM].y=impressao[*indiceM-1].y;
256             impressao[*indiceM].z=impressao[*indiceM-1].z;
257             impressao[*indiceM].e=impressao[*indiceM-1].e;
258             impressao[*indiceM].f=impressao[*indiceM-1].f;
259         }
260         else
261         {
262             impressao[*indiceM].x=0;
263             impressao[*indiceM].y=0;
264             impressao[*indiceM].z=0;
265             impressao[*indiceM].e=0;
266             impressao[*indiceM].f=0;
267         }
268         if (temCoordenada (lin, 'X', &novoValor))
269         {
270             impressao[*indiceM].x=novoValor;
271             sprintf (valor_por_extenso, "%.3lf", novoValor);
272             algumaCoordenada=TRUE;
273         }
274         if (temCoordenada (lin, 'Y', &novoValor))
275         {
276             impressao[*indiceM].y=novoValor;
277             sprintf (valor_por_extenso, "%.3lf", novoValor);
278             algumaCoordenada=TRUE;
279         }
280         if (temCoordenada (lin, 'Z', &novoValor))
281         {
282             impressao[*indiceM].z=novoValor;
283             sprintf (valor_por_extenso, "%.3lf", novoValor);
284             algumaCoordenada=TRUE;
285         }
286         if (temCoordenada (lin, 'E', &novoValor))
287         {
288             impressao[*indiceM].e=novoValor;
289             sprintf (valor_por_extenso, "%.3lf", novoValor);
290             algumaCoordenada=TRUE;
291         }
292         if (temCoordenada (lin, 'F', &novoValor))
293         {
294             impressao[*indiceM].f=novoValor;
295             sprintf (valor_por_extenso, "%.3lf", novoValor);
296             algumaCoordenada=TRUE;
297         }
298         if (!algumaCoordenada && (g_code==92))
299         {
300             impressao[*indiceM].x=0;
301             impressao[*indiceM].y=0;
302             impressao[*indiceM].z=0;
303             impressao[*indiceM].e=0;
304             impressao[*indiceM].f=0;
305         }
306         ++(*indiceM);
307         return (TRUE);
308     };
309 }
310 }
311
312 if (*lin=='M')
313 {
314     sscanf (lin+1, "%d", &param);
315     switch (param)
316     {
317         case 82:
318         case 84:
319         case 104:
320         case 109:
321         case 140: return (FALSE);
322     };
323 }
324
325 return (TRUE);
326 }
327
328 double diffPercentual (double v1, double v2)
329 {
330     if (v1>v2)
331         return (((v1-v2)/v2)*100);

```

```

332     else
333         return (((v2-v1)/v2)*100);
334     }
335
336     void getModeloGcode (void)
337     {
338         int fh;
339         int tamArquivo;
340         char *arquivo, *ini_linha, *fim_linha;
341         int nMovimento;
342         double razao, distancia;
343
344         fh=_open("modelo.gcode", O_BINARY|O_RDONLY);
345         if (fh==-1) return;
346
347         tamArquivo=_filelength (fh);
348         arquivo=(char *)malloc (tamArquivo+1);
349         _read (fh, arquivo, tamArquivo);
350         _close (fh);
351
352         nLinhasDoModelo=nMovimento=0;
353         ini_linha=arquivo;
354         total_razao_vezes_distancia=total_distancia_percorrida=0;
355
356         while (tamArquivo>0)
357         {
358             fim_linha=ini_linha;
359             while ((*fim_linha)!='\n')
360             {
361                 ++fim_linha;
362                 if (--tamArquivo==0) return;
363             }
364             *fim_linha='\0';
365
366             if (interpretaGCode (ini_linha, nLinhasDoModelo+1, &nMovimento))
367             {
368                 if (nMovimento>1)
369                 {
370                     if ((impressao[nMovimento-1].e-impressao[nMovimento-2].e)>0.01)
371                     {
372                         if ((impressao[nMovimento-1].x!=impressao[nMovimento-2].x) ||
373                             (impressao[nMovimento-1].y!=impressao[nMovimento-2].y))
374                         {
375                             distancia=calculaDistancia((int) (impressao[nMovimento-1].x*10000),
376                                                         (int) (impressao[nMovimento-1].y*10000),
377                                                         (int) (impressao[nMovimento-2].x*10000),
378                                                         (int) (impressao[nMovimento-2].y*10000));
379                             razao=distancia/((impressao[nMovimento-1].e-impressao[nMovimento-2].e)*10000);
380
381                             total_razao_vezes_distancia+=razao*distancia;
382                             total_distancia_percorrida+=distancia;
383
384                             if (razao<menorRazao) menorRazao=razao;
385                             if (razao>maiorRazao) maiorRazao=razao;
386                             if (n_razoes_utilizadas<N_MAX_RAZOES)
387                             {
388                                 if (n_razoes_utilizadas)
389                                 {
390                                     if (diffPercentual (razao, razao_de_extrusao[n_razoes_utilizadas-1])>2)
391                                     {
392                                         linha_nova_razao[n_razoes_utilizadas]=nLinhasDoModelo+1;
393                                         razao_de_extrusao[n_razoes_utilizadas]=razao;
394                                         distancia_percorrida[n_razoes_utilizadas]=distancia/10000;
395                                         camada_z[n_razoes_utilizadas]=impressao[nMovimento-1].z;
396                                         feed_rate[n_razoes_utilizadas]=impressao[nMovimento-1].f;
397                                         comprimento_extrudado[n_razoes_utilizadas++]+=
398                                             impressao[nMovimento-1].e-impressao[nMovimento-2].e;
399                                     }
400                                 }
401                                 else
402                                 {
403                                     linha_nova_razao[n_razoes_utilizadas]=nLinhasDoModelo+1;
404                                     razao_de_extrusao[n_razoes_utilizadas]=razao;
405                                     distancia_percorrida[n_razoes_utilizadas]=distancia/10000;
406                                     camada_z[n_razoes_utilizadas]=impressao[nMovimento-1].z;
407                                     feed_rate[n_razoes_utilizadas]=impressao[nMovimento-1].f;
408                                     comprimento_extrudado[n_razoes_utilizadas++]+=
409                                         impressao[nMovimento-1].e-impressao[nMovimento-2].e;
410                                     menorRazao=maiorRazao=razao;
411                                 }
412                             }
413                         }
414                     }
415                 }
416             }
417
418             if (nLinhasDoModelo)
419             {
420                 if (nLinhasDoModelo<12)
421                     sprintf (inicio_desenho+strlen(inicio_desenho), "%s\015\012", ini_linha);
422                 else

```

```

423         if (nLinhasDoModelo==12)
424             sprintf (final_desenho, "%s\015\012", ini_linha);
425         else
426             if (nLinhasDoModelo>12)
427                 sprintf (final_desenho+strlen(final_desenho), "%s\015\012", ini_linha);
428             }
429         else
430             sprintf (inicio_desenho, "%s\015\012", ini_linha);
431         ini_linha+=strlen(ini_linha)+1;
432         --tamArquivo;
433         if (*ini_linha==0x0a)
434             {
435                 ++ini_linha;
436                 --tamArquivo;
437             }
438         ++nLinhasDoModelo;
439     }
440
441     free(arquivo);
442 }
443
444
445 int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
446                    PSTR szCmdLine, int iCmdShow)
447 {
448     static char szAppName[] = "HSDimp3D" ;
449     HWND        hwnd;
450     MSG         msg;
451     WNDCLASSEX wndclass;
452     int         i;
453
454     wndclass.cbSize      = sizeof (wndclass) ;
455     wndclass.style       = CS_HREDRAW | CS_VREDRAW ;
456     wndclass.lpfnWndProc = WndProc ;
457     wndclass.cbClsExtra  = 0 ;
458     wndclass.cbWndExtra  = 0 ;
459     wndclass.hInstance  = hInstance ;
460     wndclass.hIcon       = LoadIcon (NULL, IDI_APPLICATION) ;
461     wndclass.hCursor     = LoadCursor (NULL, IDC_ARROW) ;
462     wndclass.hbrBackground = (HBRUSH) GetStockObject (WHITE_BRUSH) ;
463     wndclass.lpszMenuName = NULL ;
464     wndclass.lpszClassName = szAppName ;
465     wndclass.hIconSm     = LoadIcon (NULL, IDI_APPLICATION) ;
466
467     RegisterClassEx (&wndclass) ;
468
469     hwnd = CreateWindow (szAppName,           // window class name
470                         "Impress\343o 3D de pe\347as do Painel Braille - S\351rgio Valle dos Reis (ra878578)", // window caption
471                         WS_OVERLAPPEDWINDOW, // window style
472                         CW_USEDEFAULT,       // initial x position
473                         CW_USEDEFAULT,       // initial y position
474                         CW_USEDEFAULT,       // initial x size
475                         CW_USEDEFAULT,       // initial y size
476                         NULL,                // parent window handle
477                         NULL,                // window menu handle
478                         hInstance,           // program instance handle
479                         NULL) ;              // creation parameters
480
481
482     hf=CreateFont(
483         12,                // nHeight
484         0,                 // nWidth
485         0,                 // nEscapement
486         0,                 // nOrientation
487         FW_NORMAL,         // nWeight
488         FALSE,             // bItalic
489         FALSE,             // bUnderline
490         0,                 // cStrikeOut
491         ANSI_CHARSET,      // nCharSet
492         OUT_DEFAULT_PRECIS, // nOutPrecision
493         CLIP_DEFAULT_PRECIS, // nClipPrecision
494         DEFAULT_QUALITY,   // nQuality
495         DEFAULT_PITCH | FF_SWISS, // nPitchAndFamily
496         "Tahoma"
497     ); // lpszFacename
498
499     SendMessage (hwnd, WM_SETFONT, (LPARAM)hf, TRUE);
500
501     strcpy (mensagem_central,
502            "S\351rgio Valle dos Reis (ra878578)\rImpress\343o 3D de Pe\347as do Painel Braille\r");
503
504     deltaCoordF=TAMANHO_DO_PONTO*cos(M_PI/4);
505
506     n_razoes_utilizadas=0;
507     getModeloGcode ();
508
509     sprintf (mensagem_central+strlen(mensagem_central), "Modelo tem %d linhas\r", nLinhasDoModelo);
510
511     if (total_distancia_percorrida!=0.0)
512     {
513         sprintf (mensagem_central+strlen(mensagem_central), "Raz\343o varia de %.6lf a %.6lf\r",

```



```

605     (*p).x=(int)(x1+dist);
606     else
607     (*p).x=(int)(x1-dist);
608     return;
609 }
610
611 distTotal=calculaDistancia (x1, y1, x2, y2);
612
613 dx1=x1;
614 dx2=x2;
615 dy1=y1;
616 dy2=y2;
617
618 cb_r_eq=(dx1*dx1)+(dy1*dy1)-(dx2*dx2)-(dy2*dy2)-(dist*dist)+((distTotal-dist)*(distTotal-dist));
619 cb_r_eq/=(2*(dy1-dy2));
620
621 ca_r_eq=(dx2-dx1)/(dy1-dy2);
622
623 cb=(dx1*dy2)-(dy1*dx2);
624 cb/=(dx1-dx2);
625
626 ca=(dy1-dy2)/(dx1-dx2);
627
628 (*p).x=(int)((cb_r_eq-cb)/(ca-ca_r_eq));
629 (*p).y=(int)((ca*cb_r_eq)-(ca_r_eq*cb))/(ca-ca_r_eq);
630
631 }
632
633 void subtraiRaoDoFio (int x1, int y1, int x2, int y2, struct COORD *p1, struct COORD *p2)
634 {
635     double dist;
636
637     dist=calculaDistancia (x1, y1, x2, y2);
638     calculaPonto (x1, y1, x2, y2, RAO_DO_FIO, p1);
639     calculaPonto (x1, y1, x2, y2, dist-RAIO_DO_FIO, p2);
640 }
641
642 void progMovimentoSemExtrusao (HDC hdc, int x, int y)
643 {
644     double dx, dy;
645     double dist, d1, d2;
646
647     if (hdc!=NULL) MoveToEx (hdc, corrEscala(x), corrEscala(y), NULL);
648
649     dx=x-origemXDesenho;
650     dy=y-origemYDesenho;
651
652     dx/=100;
653     dy/=100;
654
655     dx+=origemXImpressora3D;
656     dy+=origemYImpressora3D;
657
658     d1=dx-impessao[indiceImpressao-1].x;
659     d2=dy-impessao[indiceImpressao-1].y;
660
661     dist=sqrt((d1*d1)+(d2*d2));
662
663     progImpressao (dx, dy, -1, impressao[indiceImpressao-1].e-(dist/razaoExtrusao), feedRateSemExtrusao, FALSE);
664
665 }
666
667 void progMovimentoSemExtrusaoCI (double dx, double dy, double razao, double feedRate)
668 {
669     double dist, d1, d2;
670
671     d1=dx-impessao[indiceImpressao-1].x;
672     d2=dy-impessao[indiceImpressao-1].y;
673
674     dist=sqrt((d1*d1)+(d2*d2));
675
676     razaoExtrusao=razao;
677
678     progImpressao (dx, dy, -1, impressao[indiceImpressao-1].e-(dist/razaoExtrusao), feedRate, FALSE);
679
680 }
681
682 void progMovimentoComExtrusao (HDC hdc, int x, int y)
683 {
684     double dx, dy;
685     double dist, d1, d2;
686
687     if (hdc!=NULL) LineTo (hdc, corrEscala(x), corrEscala(y));
688
689     dx=x-origemXDesenho;
690     dy=y-origemYDesenho;
691
692     dx/=100;
693     dy/=100;
694
695     dx+=origemXImpressora3D;

```

```

696 dy+=origemYImpressora3D;
697
698 d1=dx-impessao[indiceImpressao-1].x;
699 d2=dy-impessao[indiceImpressao-1].y;
700
701 dist=sqrt((d1*d1)+(d2*d2));
702
703 progImpressao (dx, dy, -1, impressao[indiceImpressao-1].e+(dist/razaoExtrusao), feedRateComExtrusao, TRUE);
704
705 }
706
707 void progMovimentoComExtrusaoCI (double dx, double dy, double razao, double feedRate)
708 {
709     double dist, d1, d2;
710
711     d1=dx-impessao[indiceImpressao-1].x;
712     d2=dy-impessao[indiceImpressao-1].y;
713
714     dist=sqrt((d1*d1)+(d2*d2));
715
716     razaoExtrusao=razao;
717
718     progImpressao (dx, dy, -1, impressao[indiceImpressao-1].e+(dist/razaoExtrusao), feedRate, TRUE);
719
720 }
721
722 void aproximaCirculo (HDC hdc, int origem_x, int origem_y, int largura_quadrado, int raio_externo, BOOL comRaios)
723 {
724     double delta_x[N_SEG_APROX_CIRCULO];
725     double delta_y[N_SEG_APROX_CIRCULO];
726     double rad;
727     double dc, d1;
728     struct COORD p1, p2, p3, p4;
729     int i, j, k;
730     int pos_atual_x, pos_atual_y;
731     int next_pos_x, next_pos_y;
732     int n_regiao;
733
734     rad=ANGULO_DE_APROXIMACAO_DO_CIRCULO;
735     rad=(M_PI/(180/rad));
736     for (i=0; i<N_SEG_APROX_CIRCULO; ++i)
737     {
738         delta_x[(i+6)*N_SEG_APROX_CIRCULO]=raio_externo*cos(rad*(i+1)+(rad/2));
739         delta_y[(i+6)*N_SEG_APROX_CIRCULO]=raio_externo*sin(rad*(i+1)+(rad/2));
740     }
741
742     pos_atual_x=origem_x+(largura_quadrado/2)+(int)delta_x[N_SEG_APROX_CIRCULO-1];
743     pos_atual_y=origem_y+(largura_quadrado/2)+(int)delta_y[N_SEG_APROX_CIRCULO-1];
744
745     progMovimentoComExtrusao (hdc, pos_atual_x, pos_atual_y);
746
747     n_regiao=0;
748
749     if (comRaios)
750         for (i=0; i<N_SEG_APROX_CIRCULO; ++i)
751         {
752             if ((i%2)==1)
753             {
754                 next_pos_x=origem_x+(largura_quadrado/2)+(int)delta_x[i];
755                 next_pos_y=origem_y+(largura_quadrado/2)+(int)delta_y[i];
756                 dc=calculaDistancia (pos_atual_x, pos_atual_y, next_pos_x, next_pos_y);
757                 j=((i/2)*8);
758                 k=((i/2)+1)*8;
759                 d1=calculaDistancia (vet_x[j], vet_y[j], vet_x[k], vet_y[k]);
760                 calculaPonto (vet_x[j], vet_y[j], vet_x[k], vet_y[k], (d1-dc)/2, &p1);
761                 calculaPonto (vet_x[j], vet_y[j], vet_x[k], vet_y[k], (d1+dc)/2, &p2);
762                 calculaPonto (pos_atual_x, pos_atual_y, p1.x, p1.y, COMPRIMENTO_DO_RAIO, &p3);
763                 calculaPonto (next_pos_x, next_pos_y, p2.x, p2.y, COMPRIMENTO_DO_RAIO, &p4);
764
765                 regioes[(n_regiao+1)*8].x1=next_pos_x;
766                 regioes[(n_regiao+1)*8].y1=next_pos_y;
767                 regioes[(n_regiao+1)*8].x2=p4.x;
768                 regioes[(n_regiao+1)*8].y2=p4.y;
769                 regioes[n_regiao].x3=pos_atual_x;
770                 regioes[n_regiao].y3=pos_atual_y;
771                 regioes[n_regiao].x4=p3.x;
772                 regioes[n_regiao+1].y4=p3.y;
773
774                 progMovimentoComExtrusao (hdc, p3.x, p3.y);
775                 //progMovimentoComExtrusao (hdc, vet_x2[j], vet_y2[j]);
776                 //progMovimentoSemExtrusao (hdc, p3.x, p3.y);
777                 progMovimentoComExtrusao (hdc, p4.x, p4.y);
778                 //progMovimentoComExtrusao (hdc, vet_x2[k], vet_y2[k]);
779                 //progMovimentoSemExtrusao (hdc, p4.x, p4.y);
780                 progMovimentoComExtrusao (hdc, next_pos_x, next_pos_y);
781
782                 pos_atual_x=next_pos_x;
783                 pos_atual_y=next_pos_y;
784             }
785             else
786             {

```

```

787         pos_atual_x=origem_x+(largura_quadrado/2)+(int)delta_x[i];
788         pos_atual_y=origem_y+(largura_quadrado/2)+(int)delta_y[i];
789
790         progMovimentoComExtrusao (hdc, pos_atual_x, pos_atual_y);
791     }
792 }
793 else
794     for (i=0; i<N_SEG_APROX_CIRCULO; ++i)
795     {
796         pos_atual_x=origem_x+(largura_quadrado/2)+(int)delta_x[i];
797         pos_atual_y=origem_y+(largura_quadrado/2)+(int)delta_y[i];
798
799         progMovimentoComExtrusao(hdc, pos_atual_x, pos_atual_y);
800     }
801 }
802
803 #define DISTANCIA_ENTRE_PONTOS 70
804 #define MARGEM_LATERAL_PONTO 50
805 #define ALTURA_CARACTER_BRILLE 600+(2*DISTANCIA_ENTRE_PONTOS)
806
807 #define DX1 MARGEM_LATERAL_PONTO
808 #define DX2 200-MARGEM_LATERAL_PONTO
809 #define DX3 MARGEM_LATERAL_PONTO+200+DISTANCIA_ENTRE_PONTOS
810 #define DX4 400+DISTANCIA_ENTRE_PONTOS-MARGEM_LATERAL_PONTO
811 #define DX5 MARGEM_LATERAL_PONTO+400+(2*DISTANCIA_ENTRE_PONTOS)
812 #define DX6 600+(2*DISTANCIA_ENTRE_PONTOS)-MARGEM_LATERAL_PONTO
813
814 void desenhaLadosDoDisco (HDC hdc, int origem_x, int origem_y, int largura_quadrado, BOOL comRelevo,
815                          BOOL apenasCalculaVertices)
816 {
817     int ladoDisco;
818     int i;
819     struct COORD p1, p2, p3, p4, p5, p6;
820
821     ladoDisco=calculaLadoDisco (largura_quadrado);
822
823     vet_x[0]=origem_x+(largura_quadrado-ladoDisco)/2;
824     vet_y[0]=origem_y;
825
826     vet_x[1]=origem_x+(largura_quadrado+ladoDisco)/2;
827     vet_y[1]=origem_y;
828
829     vet_x[2]=origem_x+largura_quadrado;
830     vet_y[2]=origem_y+(largura_quadrado-ladoDisco)/2;
831
832     vet_x[3]=origem_x+largura_quadrado;
833     vet_y[3]=origem_y+(largura_quadrado+ladoDisco)/2;
834
835     vet_x[4]=origem_x+(largura_quadrado+ladoDisco)/2;
836     vet_y[4]=origem_y+largura_quadrado;
837
838     vet_x[5]=origem_x+(largura_quadrado-ladoDisco)/2;
839     vet_y[5]=origem_y+largura_quadrado;
840
841     vet_x[6]=origem_x;
842     vet_y[6]=origem_y+(largura_quadrado+ladoDisco)/2;
843
844     vet_x[7]=origem_x;
845     vet_y[7]=origem_y+(largura_quadrado-ladoDisco)/2;
846
847     vet_x[8]=vet_x[0];
848     vet_y[8]=vet_y[0];
849
850     if (apenasCalculaVertices) return;
851
852     if (comRelevo)
853         feedRateComExtrusao=feedRateComExtrusaoDeSuperficies;
854     else
855         feedRateComExtrusao=feedRateComExtrusaoInfill;
856
857     progMovimentoComExtrusao (hdc, vet_x[0], vet_y[0]);
858
859     if (comRelevo)
860     {
861         for (i=1; i<9; ++i)
862             switch (i)
863             {
864                 case 1: calculaPonto (vet_x[0], vet_y[0], vet_x[1], vet_y[1], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
865                                     +DX1, &p1);
866                 calculaPonto (vet_x[0], vet_y[0], vet_x[1], vet_y[1], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
867                                     +DX2, &p2);
868                 calculaPonto (vet_x[0], vet_y[0], vet_x[1], vet_y[1], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
869                                     +DX3, &p3);
870                 calculaPonto (vet_x[0], vet_y[0], vet_x[1], vet_y[1], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
871                                     +DX4, &p4);
872                 calculaPonto (vet_x[0], vet_y[0], vet_x[1], vet_y[1], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
873                                     +DX5, &p5);
874                 calculaPonto (vet_x[0], vet_y[0], vet_x[1], vet_y[1], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
875                                     +DX6, &p6);
876
877                 progMovimentoComExtrusao (hdc, p1.x, p1.y);
878                 progMovimentoComExtrusao (hdc, p1.x, p1.y-TAMANHO_DO_PONTO);

```



```

878 progMovimentoComExtrusao (hdc, p2.x, p2.y-TAMANHO_DO_PONTO);
879 progMovimentoComExtrusao (hdc, p2.x, p2.y);
880 progMovimentoComExtrusao (hdc, p3.x, p3.y);
881 progMovimentoComExtrusao (hdc, p3.x, p3.y-TAMANHO_DO_PONTO);
882 progMovimentoComExtrusao (hdc, p4.x, p4.y-TAMANHO_DO_PONTO);
883 progMovimentoComExtrusao (hdc, p4.x, p4.y);
884 progMovimentoComExtrusao (hdc, p5.x, p5.y);
885 progMovimentoComExtrusao (hdc, p5.x, p5.y-TAMANHO_DO_PONTO);
886 progMovimentoComExtrusao (hdc, p6.x, p6.y-TAMANHO_DO_PONTO);
887 progMovimentoComExtrusao (hdc, p6.x, p6.y);
888 progMovimentoComExtrusao (hdc, vet_x[1], vet_y[1]);
889 break;
890
891 case 2: progMovimentoComExtrusao (hdc, vet_x[2], vet_y[2]);
892 break;
893
894 case 3: calculaPonto (vet_x[2], vet_y[2], vet_x[3], vet_y[3], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
895 +DX1, &p1);
896 calculaPonto (vet_x[2], vet_y[2], vet_x[3], vet_y[3], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
897 +DX2, &p2);
898 calculaPonto (vet_x[2], vet_y[2], vet_x[3], vet_y[3], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
899 +DX3, &p3);
900 calculaPonto (vet_x[2], vet_y[2], vet_x[3], vet_y[3], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
901 +DX4, &p4);
902
903 progMovimentoComExtrusao (hdc, p1.x, p1.y);
904 progMovimentoComExtrusao (hdc, p1.x+TAMANHO_DO_PONTO, p1.y);
905 progMovimentoComExtrusao (hdc, p2.x+TAMANHO_DO_PONTO, p2.y);
906 progMovimentoComExtrusao (hdc, p2.x, p2.y);
907 progMovimentoComExtrusao (hdc, p3.x, p3.y);
908 progMovimentoComExtrusao (hdc, p3.x+TAMANHO_DO_PONTO, p3.y);
909 progMovimentoComExtrusao (hdc, p4.x+TAMANHO_DO_PONTO, p4.y);
910 progMovimentoComExtrusao (hdc, p4.x, p4.y);
911 progMovimentoComExtrusao (hdc, vet_x[3], vet_y[3]);
912 break;
913
914 case 4: calculaPonto (vet_x[3], vet_y[3], vet_x[4], vet_y[4], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
915 +DX1, &p1);
916 calculaPonto (vet_x[3], vet_y[3], vet_x[4], vet_y[4], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
917 +DX2, &p2);
918 progMovimentoComExtrusao (hdc, p1.x, p1.y);
919 progMovimentoComExtrusao (hdc, p1.x+(int)deltaCoordP, p1.y+(int)deltaCoordP);
920 progMovimentoComExtrusao (hdc, p2.x+(int)deltaCoordP, p2.y+(int)deltaCoordP);
921 progMovimentoComExtrusao (hdc, p2.x, p2.y);
922 progMovimentoComExtrusao (hdc, vet_x[4], vet_y[4]);
923 break;
924
925 case 5: calculaPonto (vet_x[4], vet_y[4], vet_x[5], vet_y[5], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
926 +DX1, &p1);
927 calculaPonto (vet_x[4], vet_y[4], vet_x[5], vet_y[5], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
928 +DX2, &p2);
929 calculaPonto (vet_x[4], vet_y[4], vet_x[5], vet_y[5], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
930 +DX5, &p3);
931 calculaPonto (vet_x[4], vet_y[4], vet_x[5], vet_y[5], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
932 +DX6, &p4);
933
934 progMovimentoComExtrusao (hdc, p1.x, p1.y);
935 progMovimentoComExtrusao (hdc, p1.x+TAMANHO_DO_PONTO);
936 progMovimentoComExtrusao (hdc, p2.x, p2.y+TAMANHO_DO_PONTO);
937 progMovimentoComExtrusao (hdc, p2.x, p2.y);
938 progMovimentoComExtrusao (hdc, p3.x, p3.y+TAMANHO_DO_PONTO);
939 progMovimentoComExtrusao (hdc, p4.x, p4.y+TAMANHO_DO_PONTO);
940 progMovimentoComExtrusao (hdc, p4.x, p4.y);
941 progMovimentoComExtrusao (hdc, vet_x[5], vet_y[5]);
942 break;
943
944 case 6: calculaPonto (vet_x[5], vet_y[5], vet_x[6], vet_y[6], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
945 +DX3, &p1);
946 calculaPonto (vet_x[5], vet_y[5], vet_x[6], vet_y[6], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
947 +DX4, &p2);
948 progMovimentoComExtrusao (hdc, p1.x, p1.y);
949 progMovimentoComExtrusao (hdc, p1.x-(int)deltaCoordP, p1.y+(int)deltaCoordP);
950 progMovimentoComExtrusao (hdc, p2.x-(int)deltaCoordP, p2.y+(int)deltaCoordP);
951 progMovimentoComExtrusao (hdc, p2.x, p2.y);
952 progMovimentoComExtrusao (hdc, vet_x[6], vet_y[6]);
953 break;
954
955 case 7: calculaPonto (vet_x[6], vet_y[6], vet_x[7], vet_y[7], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
956 +DX3, &p1);
957 calculaPonto (vet_x[6], vet_y[6], vet_x[7], vet_y[7], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
958 +DX4, &p2);
959 calculaPonto (vet_x[6], vet_y[6], vet_x[7], vet_y[7], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
960 +DX5, &p3);
961 calculaPonto (vet_x[6], vet_y[6], vet_x[7], vet_y[7], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
962 +DX6, &p4);
963
964 progMovimentoComExtrusao (hdc, p1.x, p1.y);
965 progMovimentoComExtrusao (hdc, p1.x-TAMANHO_DO_PONTO, p1.y);
966 progMovimentoComExtrusao (hdc, p2.x-TAMANHO_DO_PONTO, p2.y);
967 progMovimentoComExtrusao (hdc, p2.x, p2.y);
968 progMovimentoComExtrusao (hdc, p3.x, p3.y);
969 progMovimentoComExtrusao (hdc, p3.x-TAMANHO_DO_PONTO, p3.y);
970 progMovimentoComExtrusao (hdc, p4.x-TAMANHO_DO_PONTO, p4.y);

```



```

969         progMovimentoComExtrusao (hdc, p4.x, p4.y);
970         progMovimentoComExtrusao (hdc, vet_x[7], vet_y[7]);
971         break;
972
973     case 8: calculaPonto (vet_x[7], vet_y[7], vet_x[8], vet_y[8], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
974             +DX5, &p1);
975             calculaPonto (vet_x[7], vet_y[7], vet_x[8], vet_y[8], ((ladoDisco-ALTURA_CARACTER_BRILLE)/2)
976                     +DX6, &p2);
977             progMovimentoComExtrusao (hdc, p1.x, p1.y);
978             progMovimentoComExtrusao (hdc, p1.x-(int)deltaCoordP, p1.y-(int)deltaCoordP);
979             progMovimentoComExtrusao (hdc, p2.x-(int)deltaCoordP, p2.y-(int)deltaCoordP);
980             progMovimentoComExtrusao (hdc, p2.x, p2.y);
981             progMovimentoComExtrusao (hdc, vet_x[8], vet_y[8]);
982             break;
983
984     default: progMovimentoComExtrusao (hdc, vet_x[i], vet_y[i]);
985             break;
986
987     };
988 }
989 else
990     for (i=1; i<9; ++i)
991     {
992         if (ehSuporte&&((i==4)|| (i==5)))
993         {
994             if (i==4)
995             {
996                 progMovimentoComExtrusao (hdc, origem_x+largura_quadrado,
997                         origem_y+largura_quadrado+largura_suporte);
998                 progMovimentoComExtrusao (hdc, origem_x+largura_quadrado-comprimento_suporte,
999                         origem_y+largura_quadrado+largura_suporte);
1000                 progMovimentoComExtrusao (hdc, origem_x+largura_quadrado-comprimento_suporte,
1001                         origem_y+largura_quadrado);
1002             }
1003             else
1004             {
1005                 progMovimentoComExtrusao (hdc, origem_x+comprimento_suporte, origem_y+largura_quadrado);
1006                 progMovimentoComExtrusao (hdc, origem_x+comprimento_suporte,
1007                         origem_y+largura_quadrado+largura_suporte);
1008                 progMovimentoComExtrusao (hdc, origem_x, origem_y+largura_quadrado+largura_suporte);
1009             }
1010         }
1011         else
1012         if (ehSuporte&&((i==2)|| (i==7)))
1013         {
1014             if (i==2)
1015             {
1016                 progMovimentoComExtrusao (hdc, origem_x+largura_quadrado-comprimento_suporte, origem_y);
1017                 progMovimentoComExtrusao (hdc, origem_x+largura_quadrado-comprimento_suporte,
1018                         origem_y-largura_suporte);
1019                 progMovimentoComExtrusao (hdc, origem_x+largura_quadrado, origem_y-largura_suporte);
1020             }
1021             else
1022             {
1023                 progMovimentoComExtrusao (hdc, origem_x, origem_y-largura_suporte);
1024                 progMovimentoComExtrusao (hdc, origem_x+comprimento_suporte, origem_y-largura_suporte);
1025                 progMovimentoComExtrusao (hdc, origem_x+comprimento_suporte, origem_y);
1026             }
1027         }
1028         else
1029             progMovimentoComExtrusao (hdc, vet_x[i], vet_y[i]);
1030     }
1031 }
1032
1033 #define N_MAX_PONTOS 20
1034 #define DIST_PREENCHIMENTO 150
1035
1036 void preencheRegiao (HDC hdc, struct REGIAO *regiao)
1037 {
1038     double dist;
1039     int n_pontos;
1040     struct COORD pt[N_MAX_PONTOS*2];
1041     int i, j;
1042
1043     dist=calculaDistancia((*regiao).x1, (*regiao).y1, (*regiao).x2, (*regiao).y2);
1044     n_pontos=(int) (dist/DIST_PREENCHIMENTO);
1045     if (n_pontos>N_MAX_PONTOS) n_pontos=N_MAX_PONTOS;
1046     j=0;
1047     for (i=0; i<n_pontos; ++i)
1048     {
1049         calculaPonto ((*regiao).x1, (*regiao).y1, (*regiao).x2, (*regiao).y2, (i+1)*DIST_PREENCHIMENTO, &(pt[j++]));
1050         calculaPonto ((*regiao).x3, (*regiao).y3, (*regiao).x4, (*regiao).y4, (i+1)*DIST_PREENCHIMENTO, &(pt[j++]));
1051     }
1052     progMovimentoSemExtrusao (hdc, pt[0].x, pt[0].y);
1053     for (i=1; i<(n_pontos*2); ++i)
1054         progMovimentoComExtrusao (hdc, pt[i].x, pt[i].y);

```

```

1060 }
1061
1062 void desenhaDiscoBraille (HDC hdc, int origem_x, int origem_y, int largura_quadrado, int diametro_furo_central,
1063                          BOOL comRelevo)
1064 {
1065     int raio_externo;
1066     int i;
1067     // struct COORD p1, p2;
1068
1069     if (comRelevo)
1070         razaoExtrusao=56.0;
1071
1072     if (ehSuporte)
1073     {
1074         largura_suporte=LARGURA_SUPOORTE/2;
1075         comprimento_suporte=COMPRIMENTO_SUPOORTE;
1076     }
1077
1078     desenhaLadosDoDisco (hdc, origem_x, origem_y, largura_quadrado, comRelevo, (!comRelevo)&&(!ehSuporte));
1079
1080     if (ehSuporte) largura_suporte=-(LARGURA_SUPOORTE/2);
1081
1082     razaoExtrusao=28.0;
1083
1084     for (i=0; i<9; ++i)
1085     {
1086         vet_x2[i]=vet_x[i];
1087         vet_y2[i]=vet_y[i];
1088     }
1089
1090     desenhaLadosDoDisco (hdc, origem_x+100, origem_y+100, largura_quadrado-200, FALSE, (!comRelevo)&&(!ehSuporte));
1091
1092     indiceFase2Desenho=indiceImpressao;
1093     desenhaLadosDoDisco (hdc, origem_x+200, origem_y+200, largura_quadrado-400, FALSE, FALSE);
1094
1095     for (i=0; i<9; ++i)
1096     {
1097         vet_x[i]=vet_x2[i];
1098         vet_y[i]=vet_y2[i];
1099     }
1100
1101     feedRateComExtrusao=feedRateComExtrusaoInfill;
1102
1103     // for (i=0; i<8; ++i)
1104     // {
1105     //     subtraiRaioDoFio (vet_x[i], vet_y[i], vet_x2[i], vet_y2[i], &p1, &p2);
1106     //     progMovimentoSemExtrusao (hdc, p1.x, p1.y);
1107     //     progMovimentoComExtrusao (hdc, p2.x, p2.y);
1108     // }
1109
1110     feedRateComExtrusao=feedRateComExtrusaoDeSuperficies;
1111
1112     raio_externo=calculaRaioExterno (diametro_furo_central);
1113     aproximaCirculo (hdc, origem_x, origem_y, largura_quadrado, raio_externo, TRUE);
1114
1115     feedRateComExtrusao=feedRateComExtrusaoInfill;
1116
1117     // for (i=0; i<8; ++i)
1118     //     preencheRegiao (hdc, &regioes[i]);
1119 }
1120
1121 #define PASSO_PAINEL 100
1122 #define COMPRIMENTO_ENCAIXE (COMPRIMENTO_SUPOORTE-300)
1123 #define N_CELULAS 2
1124
1125 void desenhaPainel (HDC hdc, int origem_x, int origem_y, int largura_quadrado)
1126 {
1127     int i, j, a, b, y1, y2, y3, y4;
1128     int x[14];
1129     int ladoDisco;
1130
1131     ladoDisco=calculaLadoDisco (largura_quadrado);
1132     ladoDisco=((largura_quadrado-200)-ladoDisco)/2-300;
1133
1134     progMovimentoComExtrusao (hdc, origem_x, origem_y);
1135     progMovimentoComExtrusao (hdc, origem_x, origem_y+COMPRIMENTO_ENCAIXE);
1136     progMovimentoComExtrusao (hdc, origem_x+300, origem_y+COMPRIMENTO_ENCAIXE);
1137     progMovimentoComExtrusao (hdc, origem_x+300, origem_y+largura_quadrado-200-COMPRIMENTO_ENCAIXE);
1138     progMovimentoComExtrusao (hdc, origem_x, origem_y+largura_quadrado-200-COMPRIMENTO_ENCAIXE);
1139     progMovimentoComExtrusao (hdc, origem_x, origem_y+largura_quadrado-200);
1140     progMovimentoComExtrusao (hdc, origem_x+(10+14*N_CELULAS)*PASSO_PAINEL+300, origem_y+largura_quadrado-200);
1141     progMovimentoComExtrusao (hdc, origem_x+(10+14*N_CELULAS)*PASSO_PAINEL+300,
1142                               origem_y+largura_quadrado-200-COMPRIMENTO_ENCAIXE);
1143     progMovimentoComExtrusao (hdc, origem_x+(10+14*N_CELULAS)*PASSO_PAINEL+80,
1144                               origem_y+largura_quadrado-200-COMPRIMENTO_ENCAIXE);
1145     progMovimentoComExtrusao (hdc, origem_x+(10+14*N_CELULAS)*PASSO_PAINEL+80, origem_y+largura_quadrado-280);
1146     progMovimentoComExtrusao (hdc, origem_x+180+(PASSO_PAINEL*2), origem_y+largura_quadrado-280);
1147
1148     origem_x+=180;
1149
1150     for (i=1; i<3; ++i)

```

```

1151 {
1152     progMovimentoComExtrusao (hdc, origem_x+(PASSO_PAINEL*(i*2)), origem_y+largura_quadrado-360);
1153     progMovimentoComExtrusao (hdc, origem_x+(PASSO_PAINEL*(i*2)), origem_y+80);
1154     progMovimentoComExtrusao (hdc, origem_x+(PASSO_PAINEL*((i*2)+1)), origem_y+80);
1155     progMovimentoComExtrusao (hdc, origem_x+(PASSO_PAINEL*((i*2)+1)), origem_y+largura_quadrado-360);
1156 }
1157
1158 for (j=0; j<14; ++j)
1159     x[j]=origem_x+(PASSO_PAINEL*(i*2))+(PASSO_PAINEL*j);
1160
1161 a=origem_y+largura_quadrado-360;
1162 b=origem_y+80;
1163 y1=(a-ladoDisco)+160;
1164 y2=origem_y+ladoDisco;
1165 y3=y2-80;
1166 y4=y1+80;
1167
1168 for (j=0; j<N_CELULAS; ++j)
1169 {
1170     progMovimentoComExtrusao (hdc, x[0], a);
1171     progMovimentoComExtrusao (hdc, x[0], y1);
1172     progMovimentoComExtrusao (hdc, x[1], y1);
1173     progMovimentoComExtrusao (hdc, x[1], a);
1174     progMovimentoComExtrusao (hdc, x[2], a);
1175     progMovimentoComExtrusao (hdc, x[2], y2);
1176     progMovimentoComExtrusao (hdc, x[0], y2);
1177     progMovimentoComExtrusao (hdc, x[0], b);
1178     progMovimentoComExtrusao (hdc, x[1], b);
1179     progMovimentoComExtrusao (hdc, x[1], y3);
1180     progMovimentoComExtrusao (hdc, x[2], y3);
1181     progMovimentoComExtrusao (hdc, x[2], b);
1182     progMovimentoComExtrusao (hdc, x[3], b);
1183     progMovimentoComExtrusao (hdc, x[3], y2);
1184     progMovimentoComExtrusao (hdc, x[4], y2);
1185     progMovimentoComExtrusao (hdc, x[4], b);
1186     progMovimentoComExtrusao (hdc, x[5], b);
1187     progMovimentoComExtrusao (hdc, x[5], y1);
1188     progMovimentoComExtrusao (hdc, x[3], y1);
1189     progMovimentoComExtrusao (hdc, x[3], a);
1190     progMovimentoComExtrusao (hdc, x[4], a);
1191     progMovimentoComExtrusao (hdc, x[4], y4);
1192     progMovimentoComExtrusao (hdc, x[5], y4);
1193     progMovimentoComExtrusao (hdc, x[5], a);
1194     progMovimentoComExtrusao (hdc, x[6], a);
1195     progMovimentoComExtrusao (hdc, x[6], b);
1196     progMovimentoComExtrusao (hdc, x[7], b);
1197     progMovimentoComExtrusao (hdc, x[7], y2);
1198     progMovimentoComExtrusao (hdc, x[8], y2);
1199     progMovimentoComExtrusao (hdc, x[8], b);
1200     progMovimentoComExtrusao (hdc, x[9], b);
1201     progMovimentoComExtrusao (hdc, x[9], y1);
1202     progMovimentoComExtrusao (hdc, x[7], y1);
1203     progMovimentoComExtrusao (hdc, x[7], a);
1204     progMovimentoComExtrusao (hdc, x[8], a);
1205     progMovimentoComExtrusao (hdc, x[8], y4);
1206     progMovimentoComExtrusao (hdc, x[9], y4);
1207     progMovimentoComExtrusao (hdc, x[9], a);
1208     progMovimentoComExtrusao (hdc, x[10], a);
1209     progMovimentoComExtrusao (hdc, x[10], y1);
1210     progMovimentoComExtrusao (hdc, x[11], y1);
1211     progMovimentoComExtrusao (hdc, x[11], a);
1212     progMovimentoComExtrusao (hdc, x[12], a);
1213     progMovimentoComExtrusao (hdc, x[12], y2);
1214     progMovimentoComExtrusao (hdc, x[10], y2);
1215     progMovimentoComExtrusao (hdc, x[10], b);
1216     progMovimentoComExtrusao (hdc, x[11], b);
1217     progMovimentoComExtrusao (hdc, x[11], y3);
1218     progMovimentoComExtrusao (hdc, x[12], y3);
1219     progMovimentoComExtrusao (hdc, x[12], b);
1220     progMovimentoComExtrusao (hdc, x[13], b);
1221     progMovimentoComExtrusao (hdc, x[13], a);
1222
1223     for (i=0; i<14; ++i) x[i]+=(PASSO_PAINEL*14);
1224 }
1225
1226 for (i=0; i<1; ++i)
1227 {
1228     progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*(i*2)), origem_y+largura_quadrado-360);
1229     progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*(i*2)), origem_y+80);
1230     progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2)+1)), origem_y+80);
1231     progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2)+1)), origem_y+largura_quadrado-360);
1232 }
1233
1234 progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2)))), origem_y+largura_quadrado-360);
1235 //progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2))))+200,
1236 //    origem_y+largura_quadrado-200-COMPIMENTO_ENCAIXE);
1237 //progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2)))),
1238 //    origem_y+largura_quadrado-200-COMPIMENTO_ENCAIXE);
1239 progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2)))), origem_y+COMPIMENTO_ENCAIXE);
1240 progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2))))+300, origem_y+COMPIMENTO_ENCAIXE);
1241 progMovimentoComExtrusao (hdc, x[0]+(PASSO_PAINEL*((i*2))))+300, origem_y);

```



```

1242     progMovimentoComExtrusao (hdc, origem_x-180, origem_y);
1243 }
1244
1245 char cmd_mudanca_de_layer[] = "G92 E0\015\012";
1246
1247 void gravaArquivoDoDesenho (char *nomeArquivo)
1248 {
1249     int fh;
1250     int i;
1251     char movimento[80];
1252
1253     fh=open (nomeArquivo, O_APPEND|O_CREAT|O_TRUNC|O_RDWR|O_BINARY,S_IWRITE|S_IREAD);
1254
1255     write (fh, inicio_desenho, strlen(inicio_desenho));
1256
1257     for (i=1; i<indiceImpressao; ++i)
1258     {
1259         strcpy (movimento, "G1");
1260         if (impressao[i].x!=impressao[i-1].x)
1261             sprintf (movimento+strlen(movimento), " X%.3lf", impressao[i].x);
1262         if (impressao[i].y!=impressao[i-1].y)
1263             sprintf (movimento+strlen(movimento), " Y%.3lf", impressao[i].y);
1264         if (impressao[i].z!=impressao[i-1].z)
1265         {
1266             write (fh, cmd_mudanca_de_layer, strlen(cmd_mudanca_de_layer));
1267             sprintf (movimento+strlen(movimento), " Z%.3lf", impressao[i].z);
1268         }
1269         if (impressao[i].e!=impressao[i-1].e)
1270             sprintf (movimento+strlen(movimento), " E%.5lf", impressao[i].e);
1271         if (impressao[i].f!=impressao[i-1].f)
1272             sprintf (movimento+strlen(movimento), " F%.3lf", impressao[i].f);
1273         strcpy (movimento+strlen(movimento), "\015\012");
1274
1275         write (fh, movimento, strlen(movimento));
1276     }
1277
1278     write (fh, final_desenho, strlen(final_desenho));
1279
1280     close (fh);
1281 }
1282
1283 void gravaDesenhoDiscoSemRelevo (void)
1284 {
1285     int i, j;
1286
1287     for (j=1; j<5; ++j)
1288     {
1289         progImpressao (-1, -1, impressao[indiceImpressao-1].z+0.300, 0, feedRateSemExtrusao, FALSE);
1290         for (i=indiceInicialDesenho; i<indiceFinalDesenho; ++i)
1291             if (impressao[i].comExtrusao)
1292                 progMovimentoComExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1293             else
1294                 progMovimentoSemExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1295     }
1296
1297     gravaArquivoDoDesenho ("disco sem relevo.gcode");
1298 }
1299
1300 void gravaDesenhoDiscoComRelevo (void)
1301 {
1302     int i, j;
1303
1304     for (j=1; j<7; ++j)
1305     {
1306         progImpressao (-1, -1, impressao[indiceImpressao-1].z+0.300, 0, feedRateSemExtrusao, FALSE);
1307         for (i=indiceInicialDesenho; i<indiceFinalDesenho; ++i)
1308             if (impressao[i].comExtrusao)
1309                 progMovimentoComExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1310             else
1311                 progMovimentoSemExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1312     }
1313
1314     for (j=0; j<3; ++j)
1315     {
1316         progImpressao (-1, -1, impressao[indiceImpressao-1].z+0.300, 0, feedRateSemExtrusao, FALSE);
1317         for (i=indiceFase2Desenho; i<indiceFinalDesenho; ++i)
1318             if (impressao[i].comExtrusao)
1319                 progMovimentoComExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1320             else
1321                 progMovimentoSemExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1322     }
1323
1324     gravaArquivoDoDesenho ("disco com relevo.gcode");
1325 }
1326
1327 void gravaDesenhoSuporte (void)
1328 {
1329     int i, j;

```

```

1333
1334     for (j=1; j<14; ++j)
1335     {
1336         progImpressao (-1, -1, impressao[indiceImpressao-1].z+0.300, 0, feedRateSemExtrusao, FALSE);
1337         for (i=indiceInicialDesenho; i<indiceFinalDesenho; ++i)
1338             if (impressao[i].comExtrusao)
1339                 progMovimentoComExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1340             else
1341                 progMovimentoSemExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1342     }
1343
1344     gravaArquivoDoDesenho ("suporte.gcode");
1345 }
1346
1347 void gravaDesenhoPainel (void)
1348 {
1349     int i, j;
1350
1351     for (j=1; j<8; ++j)
1352     {
1353         progImpressao (-1, -1, impressao[indiceImpressao-1].z+0.300, 0, feedRateSemExtrusao, FALSE);
1354         for (i=indiceInicialDesenho; i<indiceFinalDesenho; ++i)
1355             if (impressao[i].comExtrusao)
1356                 progMovimentoComExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1357             else
1358                 progMovimentoSemExtrusaoCI (impressao[i].x, impressao[i].y, impressao[i].razao, impressao[i].f);
1359     }
1360
1361     gravaArquivoDoDesenho ("painel.gcode");
1362 }
1363
1364 void preparaGravacaoGCode (void)
1365 {
1366     feedRateSemExtrusao=12000.0;
1367     feedRateComExtrusao=1200.0;
1368     feedRateComExtrusaoDeSuperficies=540.0;
1369     feedRateComExtrusaoInfill=1800.0;
1370     razaoExtrusao=28.0;
1371
1372     impressao[0].x=0;
1373     impressao[0].y=0;
1374     impressao[0].z=0;
1375     impressao[0].e=0;
1376     impressao[0].f=0;
1377     impressao[0].nLinha=0;
1378     impressao[0].comExtrusao=FALSE;
1379
1380     indiceImpressao=1;
1381
1382     progImpressao (-1, -1, 0, 0, feedRateSemExtrusao, FALSE);
1383     origemXImpressora3D=65;
1384     origemYImpressora3D=10;
1385 }
1386
1387
1388
1389
1390 LRESULT CALLBACK WndProc (HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
1391 {
1392     HDC         hdc;
1393     PAINTSTRUCT ps;
1394     RECT        rect_w, rect_t;
1395     HFONT        hfOld;
1396
1397     switch (iMsg)
1398     {
1399     case WM_CREATE :
1400         // PlaySound ("hellowin.wav", NULL, SND_FILENAME | SND_ASYNC) ;
1401         return 0 ;
1402
1403     case WM_PAINT :
1404         hdc= BeginPaint (hwnd, &ps) ;
1405
1406         GetClientRect (hwnd, &rect_w) ;
1407
1408         rect_t.bottom = rect_w.bottom;
1409         rect_t.left = rect_w.left;
1410         rect_t.right = rect_w.right;
1411         rect_t.top = rect_w.top;
1412
1413         hfOld=(HFONT)SelectObject (hdc, hf) ;
1414
1415         DrawText (hdc, mensagem_central, -1, &rect_t, DT_CALCRECT | DT_CENTER) ;
1416
1417         rect_t.top = ((rect_w.bottom - rect_w.top)-rect_t.bottom)/2;
1418         if (rect_t.top<0) rect_t.top=0;
1419         rect_t.bottom+=rect_t.top;
1420         rect_t.left = rect_w.left;
1421         rect_t.right = rect_w.right;
1422
1423         DrawText (hdc, mensagem_central, -1, &rect_t, DT_CENTER) ;

```

```

1424
1425 SetMapMode (hdc, MM_HIMETRIC);
1426
1427 preparaGravacaoGCode ();
1428
1429 origemXDesenho=1500;
1430 origemYDesenho=-6000;
1431
1432 progMovimentoSemExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1433 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho+5000);
1434 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho+5000);
1435 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho-400);
1436 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1437
1438 indiceInicialDesenho=indiceImpressao;
1439 ehSuporte=FALSE;
1440
1441 desenhaDiscoBraille (hdc, origemXDesenho, origemYDesenho, 4500, 1950, FALSE);
1442
1443 if (!jaGravouDesenhos)
1444 {
1445     indiceFinalDesenho=indiceImpressao;
1446     gravaDesenhoDiscoSemRelevo ();
1447 }
1448
1449 preparaGravacaoGCode ();
1450
1451 origemXDesenho=7000;
1452 origemYDesenho=-6000;
1453
1454 progMovimentoSemExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1455 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho+5000);
1456 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho+5000);
1457 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho-400);
1458 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1459
1460 indiceInicialDesenho=indiceImpressao;
1461
1462 desenhaDiscoBraille (hdc, origemXDesenho, origemYDesenho, 4500, 1950, TRUE);
1463
1464 if (!jaGravouDesenhos)
1465 {
1466     indiceFinalDesenho=indiceImpressao;
1467     gravaDesenhoDiscoComRelevo ();
1468 }
1469
1470 preparaGravacaoGCode ();
1471
1472 origemXDesenho=12500;
1473 origemYDesenho=-6000;
1474
1475 progMovimentoSemExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1476 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho+5000);
1477 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho+5000);
1478 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho-400);
1479 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1480
1481 indiceInicialDesenho=indiceImpressao;
1482 ehSuporte=TRUE;
1483
1484 desenhaDiscoBraille (hdc, origemXDesenho, origemYDesenho, 4500, 1950, FALSE);
1485
1486 if (!jaGravouDesenhos)
1487 {
1488     indiceFinalDesenho=indiceImpressao;
1489     gravaDesenhoSuporte ();
1490 }
1491
1492 preparaGravacaoGCode ();
1493
1494 origemXDesenho=1500;
1495 origemYDesenho=-12000;
1496
1497 progMovimentoSemExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1498 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho+5000);
1499 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho+5000);
1500 progMovimentoComExtrusao (hdc, origemXDesenho+4800, origemYDesenho-400);
1501 progMovimentoComExtrusao (hdc, origemXDesenho-200, origemYDesenho-400);
1502
1503 indiceInicialDesenho=indiceImpressao;
1504 ehSuporte=TRUE;
1505
1506 desenhaPainel (hdc, origemXDesenho, origemYDesenho, 4500);
1507
1508 if (!jaGravouDesenhos)
1509 {
1510     indiceFinalDesenho=indiceImpressao;
1511     gravaDesenhoPainel ();
1512 }
1513
1514

```

```
1515         jaGravouDesenhos=TRUE;
1516
1517         SelectObject (hdc, hfOld);
1518
1519         EndPaint (hwnd, &ps) ;
1520         return 0 ;
1521
1522     case WM_DESTROY :
1523         PostQuitMessage (0) ;
1524         return 0 ;
1525     }
1526
1527     return DefWindowProc (hwnd, iMsg, wParam, lParam) ;
1528 }
1529
```

Anexo B

Listagem do Programa em G-CODE

Segue a listagem do programa em G-CODE gerado pelo programa em C com as instruções para fazer o disco braille:

```
1 ; arquivo gerado por Sergio Valle dos Reis
2 ; ra 878578 - DEMIC - FEEC - UNICAMP
3
4 G21 ; set units to millimeters
5 M104 S215 ; set temperature
6 G28 ; home all axes
7 M109 S215 ; wait for temperature to be reached
8 G90 ; use absolute coordinates
9 G92 E0
10 M82 ; use absolute distances for extrusion
11 M92 Y64.00
12 M92 X64.00
13 G1 F12000.000
14 G1 X63.000 Y6.000 E-2.26018
15 G1 Y60.000 E-0.33161 F1200.000
16 G1 X113.000 E1.45410
17 G1 Y6.000 E3.38268
18 G1 X63.000 E5.16839
19 G1 X78.180 Y10.000 E5.44871 F540.000
20 G1 X85.690 E5.58282
21 G1 Y9.000 E5.60068
22 G1 X86.690 E5.61854
23 G1 Y10.000 E5.63639
24 G1 X88.390 E5.66675
25 G1 Y9.000 E5.68461
26 G1 X89.390 E5.70246
27 G1 Y10.000 E5.72032
28 G1 X91.090 E5.75068
29 G1 Y9.000 E5.76854
30 G1 X92.090 E5.78639
31 G1 Y10.000 E5.80425
32 G1 X96.810 E5.88854
33 G1 X110.000 Y23.180 E6.22151
34 G1 Y30.690 E6.35561
35 G1 X111.000 E6.37347
36 G1 Y31.690 E6.39133
37 G1 X110.000 E6.40919
38 G1 Y33.390 E6.43954
39 G1 X111.000 E6.45740
40 G1 Y34.390 E6.47526
41 G1 X110.000 E6.49311
42 G1 Y41.810 E6.62561
43 G1 X104.680 Y47.130 E6.75997
44 G1 X105.380 Y47.830 E6.77764
45 C1 X104.680 Y48.530 E6.79832
46 G1 X103.980 Y47.830 E6.81300
47 G1 X96.810 Y55.000 E6.99407
48 G1 X89.300 E7.12818
49 G1 Y56.000 E7.14603
50 G1 X88.300 E7.16389
51 G1 Y55.000 E7.18175
52 G1 X83.900 E7.26032
53 G1 Y56.000 E7.27818
54 C1 X82.000 E7.29603
55 G1 Y55.000 E7.31389
56 G1 X78.180 E7.39818
57 G1 X70.960 Y47.780 E7.58051
58 G1 X70.260 Y48.480 E7.59819
```


59	G1	X69.550	Y47.780	E7.61599
60	G1	X70.250	Y47.080	E7.63367
61	G1	X65.000	Y41.810	E7.76650
62	G1	Y31.600	E7.94882	
63	G1	X64.000	E7.96668	
64	G1	Y30.600	E7.98454	
65	G1	X65.000	E8.00240	
66	G1	Y28.900	E8.03275	
67	G1	X64.000	E8.05061	
68	G1	Y27.900	E8.06847	
69	G1	X65.000	E8.08632	
70	G1	Y23.180	E8.17061	
71	G1	X74.120	Y14.060	E8.40092
72	G1	X73.420	Y13.360	E8.41860
73	G1	X74.130	Y12.650	E8.43653
74	G1	X74.830	Y13.350	E8.45421
75	G1	X78.180	Y10.000	E8.53881
76	G1	X78.590	Y11.000	E8.57741 F1800.000
77	G1	X96.400	E9.21348	
78	G1	X109.000	Y23.590	E9.84963
79	G1	Y41.400	E10.48570	
80	G1	X96.400	Y54.000	E11.12209
81	G1	X78.590	E11.75816	
82	G1	X66.000	Y41.400	E12.39431
83	G1	Y23.590	E13.03038	
84	G1	X78.590	Y11.000	E13.66627
85	G1	X79.010	Y12.000	E13.70501
86	G1	X95.990	E14.31144	
87	G1	X108.000	Y24.010	E14.91803
88	G1	Y40.990	E15.52446	
89	G1	X95.990	Y53.000	E16.13106
90	G1	X79.010	E16.73749	
91	G1	X67.000	Y40.990	E17.34408
92	G1	Y24.010	E17.95051	
93	G1	X79.010	Y12.000	E18.55711
94	G1	X81.640	Y23.730	E18.98644 F540.000
95	G1	X85.450	Y22.160	E19.13361
96	G1	X85.440	Y12.770	E19.46897
97	G1	X89.540	E19.61539	
98	G1	X89.550	Y22.160	E19.95075
99	G1	X93.360	Y23.730	E20.09792
100	G1	X100.000	Y17.080	E20.43355
101	G1	X102.910	Y20.000	E20.58078
102	G1	X96.270	Y26.640	E20.91615
103	G1	X97.840	Y30.450	E21.06332
104	G1	X107.240	E21.39903	
105	G1	Y34.550	E21.54546	
106	G1	X97.840	E21.88118	
107	G1	X96.270	Y38.360	E22.02835
108	G1	X102.910	Y45.010	E22.36397
109	G1	X100.000	Y47.920	E22.51095
110	G1	X93.360	Y41.270	E22.84657
111	G1	X89.550	Y42.840	E22.99374
112	G1	X89.540	Y52.240	E23.32946
113	G1	X85.440	E23.47588	
114	G1	X85.450	Y42.840	E23.81160
115	G1	X81.640	Y41.270	E23.95877
116	G1	X74.990	Y47.920	E24.29465
117	G1	X72.070	Y45.010	E24.44188
118	G1	X78.730	Y38.360	E24.77800
119	G1	X77.160	Y34.550	E24.92518
120	G1	X67.760	E25.26089	
121	G1	Y30.450	E25.40732	
122	G1	X77.160	E25.74303	
123	G1	X78.730	Y26.640	E25.89020
124	G1	X72.070	Y20.000	E26.22608
125	G1	X74.980	Y17.090	E26.37306
126	G1	X81.640	Y23.730	E26.70893
127	G92	E0		
128	G1	Z0.300	E0.00000	F12000.000
129	G1	X78.180	Y10.000	E0.25284 F540.000
130	G1	X85.690	E0.38695	
131	G1	Y9.000	E0.40481	
132	G1	X86.690	E0.42267	
133	G1	Y10.000	E0.44052	
134	G1	X88.390	E0.47088	
135	G1	Y9.000	E0.48874	
136	G1	X89.390	E0.50659	
137	G1	Y10.000	E0.52445	
138	G1	X91.090	E0.55481	
139	G1	Y9.000	E0.57267	
140	G1	X92.090	E0.59052	
141	G1	Y10.000	E0.60838	
142	G1	X96.810	E0.69267	
143	G1	X110.000	Y23.180	E1.02564
144	G1	Y30.690	E1.15974	
145	G1	X111.000	E1.17760	
146	G1	Y31.690	E1.19546	
147	G1	X110.000	E1.21332	
148	G1	Y33.390	E1.24367	
149	G1	X111.000	E1.26153	

150	G1	Y34.390	E1.27939	
151	G1	X110.000	E1.29724	
152	G1	Y41.810	E1.42974	
153	G1	X104.680	Y47.130	E1.56409
154	G1	X105.380	Y47.830	E1.58177
155	G1	X104.680	Y48.530	E1.59945
156	G1	X103.980	Y47.830	E1.61713
157	G1	X96.810	Y55.000	E1.79820
158	G1	X89.300	E1.93230	
159	G1	Y56.000	E1.95016	
160	G1	X88.300	E1.96802	
161	G1	Y55.000	E1.98588	
162	G1	X83.900	E2.06445	
163	G1	Y56.000	E2.08230	
164	G1	X82.900	E2.10016	
165	G1	Y55.000	E2.11802	
166	G1	X78.180	E2.20230	
167	G1	X70.960	Y47.780	E2.38464
168	G1	X70.260	Y48.480	E2.40231
169	G1	X69.550	Y47.780	E2.42012
170	G1	X70.250	Y47.080	E2.43780
171	G1	X65.000	Y41.810	E2.57063
172	G1	Y31.600	E2.75295	
173	G1	X64.000	E2.77081	
174	G1	Y30.600	E2.78867	
175	G1	X65.000	E2.80652	
176	G1	Y28.900	E2.83688	
177	G1	X64.000	E2.85474	
178	G1	Y27.900	E2.87260	
179	G1	X65.000	E2.89045	
180	G1	Y23.180	E2.97474	
181	G1	X74.120	Y14.060	E3.20505
182	G1	X73.420	Y13.360	E3.22273
183	G1	X74.130	Y12.650	E3.24066
184	G1	X74.830	Y13.350	E3.25834
185	G1	X78.180	Y10.000	E3.34294
186	G1	X78.590	Y11.000	E3.38154 F1800.000
187	G1	X96.400	E4.01761	
188	G1	X109.000	Y23.590	E4.65375
189	G1	Y41.400	E5.28983	
190	G1	X96.400	Y54.000	E5.92622
191	G1	X78.590	E6.56229	
192	G1	X66.000	Y41.400	E7.19844
193	G1	Y23.590	E7.83451	
194	G1	X78.590	Y11.000	E8.47040
195	G1	X79.010	Y12.000	E8.50914
196	G1	X95.990	E9.11556	
197	G1	X108.000	Y24.010	E9.72216
198	G1	Y40.990	E10.32859	
199	G1	X95.990	Y53.000	E10.93519
200	G1	X79.010	E11.54161	
201	G1	X67.000	Y40.990	E12.14821
202	G1	Y24.010	E12.75464	
203	G1	X79.010	Y12.000	E13.36124
204	G1	X81.640	Y23.730	E13.79057 F540.000
205	G1	X85.450	Y22.160	E13.93774
206	G1	X85.440	Y12.770	E14.27309
207	G1	X89.540	E14.41952	
208	G1	X89.550	Y22.160	E14.75488
209	G1	X93.360	Y23.730	E14.90205
210	G1	X100.000	Y17.080	E15.23768
211	G1	X102.910	Y20.000	E15.38491
212	G1	X96.270	Y26.640	E15.72028
213	G1	X97.840	Y30.450	E15.86745
214	G1	X107.240	E16.20316	
215	G1	Y34.550	E16.34959	
216	G1	X97.840	E16.68530	
217	G1	X96.270	Y38.360	E16.83248
218	G1	X102.910	Y45.010	E17.16810
219	G1	X100.000	Y47.920	E17.31508
220	G1	X93.360	Y41.270	E17.65070
221	G1	X89.550	Y42.840	E17.79787
222	G1	X89.540	Y52.240	E18.13359
223	G1	X85.440	E18.28001	
224	G1	X85.450	Y42.840	E18.61573
225	G1	X81.640	Y41.270	E18.76290
226	G1	X74.990	Y47.920	E19.09878
227	G1	X72.070	Y45.010	E19.24601
228	G1	X78.730	Y38.360	E19.58213
229	G1	X77.160	Y34.550	E19.72931
230	G1	X67.760	E20.06502	
231	G1	Y30.450	E20.21145	
232	G1	X77.160	E20.54716	
233	G1	X78.730	Y26.640	E20.69433
234	G1	X72.070	Y20.000	E21.03021
235	G1	X74.980	Y17.090	E21.17719
236	G1	X81.640	Y23.730	E21.51306
237	G92	E0		
238	G1	20.600	E0.00000	F12000.000
239	G1	X78.180	Y10.000	E0.25284 F540.000
240	G1	X85.690	E0.38695	

241	G1	Y9.000	E0.40481	
242	G1	X86.690	E0.42267	
243	G1	Y10.000	E0.44052	
244	G1	X88.390	E0.47088	
245	G1	Y9.000	E0.48874	
246	G1	X89.390	E0.50659	
247	G1	Y10.000	E0.52445	
248	G1	X91.090	E0.55481	
249	G1	Y9.000	E0.57267	
250	G1	X92.090	E0.59052	
251	G1	Y10.000	E0.60838	
252	G1	X96.810	E0.69267	
253	G1	X110.000	Y23.180	E1.02564
254	G1	Y30.690	E1.15974	
255	G1	X111.000	E1.17760	
256	G1	Y31.690	E1.19546	
257	G1	X110.000	E1.21332	
258	G1	Y33.390	E1.24367	
259	G1	X111.000	E1.26153	
260	G1	Y34.390	E1.27939	
261	G1	X110.000	E1.29724	
262	G1	Y41.810	E1.42974	
263	G1	X104.680	Y47.130	E1.56409
264	G1	X105.380	Y47.830	E1.58177
265	G1	X104.680	Y48.530	E1.59945
266	G1	X103.980	Y47.830	E1.61713
267	G1	X96.810	Y55.000	E1.79820
268	G1	X89.300	E1.93230	
269	G1	Y56.000	E1.95016	
270	G1	X88.300	E1.96802	
271	G1	Y55.000	E1.98588	
272	G1	X83.900	E2.06445	
273	G1	Y56.000	E2.08230	
274	G1	X82.900	E2.10016	
275	G1	Y55.000	E2.11802	
276	G1	X78.180	E2.20230	
277	G1	X70.960	Y47.780	E2.38464
278	G1	X70.260	Y48.480	E2.40231
279	G1	X69.550	Y47.780	E2.42012
280	G1	X70.250	Y47.080	E2.43780
281	G1	X65.000	Y41.810	E2.57063
282	G1	Y31.600	E2.75295	
283	G1	X64.000	E2.77081	
284	G1	Y30.600	E2.78867	
285	G1	X65.000	E2.80652	
286	G1	Y28.900	E2.83688	
287	G1	X64.000	E2.85474	
288	G1	Y27.900	E2.87260	
289	G1	X65.000	E2.89045	
290	G1	Y23.180	E2.97474	
291	G1	X74.120	Y14.060	E3.20505
292	G1	X73.420	Y13.360	E3.22273
293	G1	X74.130	Y12.650	E3.24066
294	G1	X74.830	Y13.350	E3.25834
295	G1	X78.180	Y10.000	E3.34294
296	G1	X78.590	Y11.000	E3.38154 F1800.000
297	G1	X96.400	E4.01761	
298	G1	X109.000	Y23.590	E4.65375
299	G1	Y41.400	E5.28983	
300	G1	X96.400	Y54.000	E5.92622
301	G1	X78.590	E6.56229	
302	G1	X66.000	Y41.400	E7.19844
303	G1	Y23.590	E7.83451	
304	G1	X78.590	Y11.000	E8.47040
305	G1	X79.010	Y12.000	E8.50914
306	G1	X95.990	E9.11556	
307	G1	X108.000	Y24.010	E9.72216
308	G1	Y40.990	E10.32859	
309	G1	X95.990	Y53.000	E10.93519
310	G1	X79.010	E11.54161	
311	G1	X67.000	Y40.990	E12.14821
312	G1	Y24.010	E12.75464	
313	G1	X79.010	Y12.000	E13.36124
314	G1	X81.640	Y23.730	E13.79057 F540.000
315	G1	X85.450	Y22.160	E13.93774
316	G1	X85.440	Y12.770	E14.27309
317	G1	X89.540	E14.41952	
318	G1	X89.550	Y22.160	E14.75488
319	G1	X93.360	Y23.730	E14.90205
320	G1	X100.000	Y17.080	E15.23768
321	G1	X102.910	Y20.000	E15.38491
322	G1	X96.270	Y26.640	E15.72028
323	G1	X97.840	Y30.450	E15.86745
324	G1	X107.240	E16.20316	
325	G1	Y34.550	E16.34959	
326	G1	X97.840	E16.68530	
327	G1	X96.270	Y38.360	E16.83248
328	G1	X102.910	Y45.010	E17.16810
329	G1	X100.000	Y47.920	E17.31508
330	G1	X93.360	Y41.270	E17.65070
331	G1	X89.550	Y42.840	E17.79787

332	G1	X89.540	Y52.240	E18.13359
333	G1	X85.440	E18.28001	
334	G1	X85.450	Y42.840	E18.61573
335	G1	X81.640	Y41.270	E18.76290
336	G1	X74.990	Y47.920	E19.09878
337	G1	X72.070	Y45.010	E19.24601
338	G1	X78.730	Y38.360	E19.58213
339	G1	X77.160	Y34.550	E19.72931
340	G1	X67.760	E20.06502	
341	G1	Y30.450	E20.21145	
342	G1	X77.160	E20.54716	
343	G1	X78.730	Y26.640	E20.69433
344	G1	X72.070	Y20.000	E21.03021
345	G1	X74.980	Y17.090	E21.17719
346	G1	X81.640	Y23.730	E21.51306
347	G92	E0		
348	G1	Z0.900	E0.00000	F12000.000
349	G1	X78.180	Y10.000	E0.25284 F540.000
350	G1	X85.690	E0.38695	
351	G1	Y9.000	E0.40481	
352	G1	X86.690	E0.42267	
353	G1	Y10.000	E0.44052	
354	G1	X88.390	E0.47088	
355	G1	Y9.000	E0.48874	
356	G1	X89.390	E0.50659	
357	G1	Y10.000	E0.52445	
358	G1	X91.090	E0.55481	
359	G1	Y9.000	E0.57267	
360	G1	X92.090	E0.59052	
361	G1	Y10.000	E0.60838	
362	G1	X96.810	E0.69267	
363	G1	X110.000	Y23.180	E1.02564
364	G1	Y30.690	E1.15974	
365	G1	X111.000	E1.17760	
366	G1	Y31.690	E1.19546	
367	G1	X110.000	E1.21332	
368	G1	Y33.390	E1.24367	
369	G1	X111.000	E1.26153	
370	G1	Y34.390	E1.27939	
371	G1	X110.000	E1.29724	
372	G1	Y41.810	E1.42974	
373	G1	X104.680	Y47.130	E1.56409
374	G1	X105.380	Y47.830	E1.58177
375	G1	X104.680	Y48.530	E1.59945
376	G1	X103.980	Y47.830	E1.61713
377	G1	X96.810	Y55.000	E1.79820
378	G1	X89.300	E1.93230	
379	G1	Y56.000	E1.95016	
380	G1	X88.300	E1.96802	
381	G1	Y55.000	E1.98588	
382	G1	X83.900	E2.06445	
383	G1	Y56.000	E2.08230	
384	G1	X82.900	E2.10016	
385	G1	Y55.000	E2.11802	
386	G1	X78.180	E2.20230	
387	G1	X70.960	Y47.780	E2.38464
388	G1	X70.260	Y48.480	E2.40231
389	G1	X69.550	Y47.780	E2.42012
390	G1	X70.250	Y47.080	E2.43780
391	G1	X65.000	Y41.810	E2.57063
392	G1	Y31.600	E2.75295	
393	G1	X64.000	E2.77081	
394	G1	Y30.600	E2.78867	
395	G1	X65.000	E2.80652	
396	G1	Y28.900	E2.83688	
397	G1	X64.000	E2.85474	
398	G1	Y27.900	E2.87260	
399	G1	X65.000	E2.89045	
400	G1	Y23.180	E2.97474	
401	G1	X74.120	Y14.060	E3.20505
402	G1	X73.420	Y13.360	E3.22273
403	G1	X74.130	Y12.650	E3.24066
404	G1	X74.830	Y13.350	E3.25834
405	G1	X78.180	Y10.000	E3.34294
406	G1	X78.590	Y11.000	E3.38154 F1800.000
407	G1	X96.400	E4.01761	
408	G1	X109.000	Y23.590	E4.65375
409	G1	Y41.400	E5.28983	
410	G1	X96.400	Y54.000	E5.92622
411	G1	X78.590	E6.56229	
412	G1	X66.000	Y41.400	E7.19844
413	G1	Y23.590	E7.83451	
414	G1	X78.590	Y11.000	E8.47040
415	G1	X79.010	Y12.000	E8.50914
416	G1	X95.990	E9.11556	
417	G1	X108.000	Y24.010	E9.72216
418	G1	Y40.990	E10.32859	
419	G1	X95.990	Y53.000	E10.93519
420	G1	X79.010	E11.54161	
421	G1	X67.000	Y40.990	E12.14821
422	G1	Y24.010	E12.75464	

423	G1	X79.010	Y12.000	E13.36124	
424	G1	X81.640	Y23.730	E13.79057	F540.000
425	G1	X85.450	Y22.160	E13.93774	
426	G1	X85.440	Y12.770	E14.27309	
427	G1	X89.540	E14.41952		
428	G1	X89.550	Y22.160	E14.75488	
429	G1	X93.360	Y23.730	E14.90205	
430	G1	X100.000	Y17.080	E15.23768	
431	G1	X102.910	Y20.000	E15.38491	
432	G1	X96.270	Y26.640	E15.72028	
433	G1	X97.840	Y30.450	E15.86745	
434	G1	X107.240	E16.20316		
435	G1	Y34.550	E16.34959		
436	G1	X97.840	E16.68530		
437	G1	X96.270	Y38.360	E16.83248	
438	G1	X102.910	Y45.010	E17.16810	
439	G1	X100.000	Y47.920	E17.31508	
440	G1	X93.360	Y41.270	E17.65070	
441	G1	X89.550	Y42.840	E17.79787	
442	G1	X89.540	Y52.240	E18.13359	
443	G1	X85.440	E18.28001		
444	G1	X85.450	Y42.840	E18.61573	
445	G1	X81.640	Y41.270	E18.76290	
446	G1	X74.990	Y47.920	E19.09878	
447	G1	X72.070	Y45.010	E19.24601	
448	G1	X78.730	Y38.360	E19.58213	
449	G1	X77.160	Y34.550	E19.72931	
450	G1	X67.760	E20.06502		
451	G1	Y30.450	E20.21145		
452	G1	X77.160	E20.54716		
453	G1	X78.730	Y26.640	E20.69433	
454	G1	X72.070	Y20.000	E21.03021	
455	G1	X74.980	Y17.090	E21.17719	
456	G1	X81.640	Y23.730	E21.51306	
457	G92	E0			
458	G1	Z1.200	E0.00000	F12000.000	
459	G1	X78.180	Y10.000	E0.25284	F540.000
460	G1	X85.690	E0.38695		
461	G1	Y9.000	E0.40481		
462	G1	X86.690	E0.42267		
463	G1	Y10.000	E0.44052		
464	G1	X88.390	E0.47088		
465	G1	Y9.000	E0.48874		
466	G1	X89.390	E0.50659		
467	G1	Y10.000	E0.52445		
468	G1	X91.090	E0.55481		
469	G1	Y9.000	E0.57267		
470	G1	X92.090	E0.59052		
471	G1	Y10.000	E0.60838		
472	G1	X96.810	E0.69267		
473	G1	X110.000	Y23.180	E1.02564	
474	G1	Y30.690	E1.15974		
475	G1	X111.000	E1.17760		
476	G1	Y31.690	E1.19546		
477	G1	X110.000	E1.21332		
478	G1	Y33.390	E1.24367		
479	G1	X111.000	E1.26153		
480	G1	Y34.390	E1.27939		
481	G1	X110.000	E1.29724		
482	G1	Y41.810	E1.42974		
483	G1	X104.680	Y47.130	E1.56409	
484	G1	X105.380	Y47.830	E1.58177	
485	G1	X104.680	Y48.530	E1.59945	
486	G1	X103.980	Y47.830	E1.61713	
487	G1	X96.810	Y55.000	E1.79820	
488	G1	X89.300	E1.93230		
489	G1	Y56.000	E1.95016		
490	G1	X88.300	E1.96802		
491	G1	Y55.000	E1.98588		
492	G1	X83.900	E2.06445		
493	G1	Y56.000	E2.08230		
494	G1	X82.900	E2.10016		
495	G1	Y55.000	E2.11802		
496	G1	X78.180	E2.20230		
497	G1	X70.960	Y47.780	E2.38464	
498	G1	X70.260	Y48.480	E2.40231	
499	G1	X69.550	Y47.780	E2.42012	
500	G1	X70.250	Y47.080	E2.43780	
501	G1	X65.000	Y41.810	E2.57063	
502	G1	Y31.600	E2.75295		
503	G1	X64.000	E2.77081		
504	G1	Y30.600	E2.78867		
505	G1	X65.000	E2.80652		
506	G1	Y28.900	E2.83688		
507	G1	X64.000	E2.85474		
508	G1	Y27.900	E2.87260		
509	G1	X65.000	E2.89045		
510	G1	Y23.180	E2.97474		
511	G1	X74.120	Y14.060	E3.20505	
512	G1	X73.420	Y13.360	E3.22273	
513	G1	X74.130	Y12.650	E3.24066	

514	G1	X74.830	Y13.350	E3.25834	
515	G1	X78.180	Y10.000	E3.34294	
516	G1	X78.590	Y11.000	E3.38154	F1800.000
517	G1	X96.400	E4.01761		
518	G1	X109.000	Y23.590	E4.65375	
519	G1	Y41.400	E5.28983		
520	G1	X96.400	Y54.000	E5.92622	
521	G1	X78.590	E6.56229		
522	G1	X66.000	Y41.400	E7.19844	
523	G1	Y23.590	E7.83451		
524	G1	X78.590	Y11.000	E8.47040	
525	G1	X79.010	Y12.000	E8.50914	
526	G1	X95.990	E9.11556		
527	G1	X108.000	Y24.010	E9.72216	
528	G1	Y40.990	E10.32859		
529	G1	X95.990	Y53.000	E10.93519	
530	G1	X79.010	E11.54161		
531	G1	X67.000	Y40.990	E12.14821	
532	G1	Y24.010	E12.75464		
533	G1	X79.010	Y12.000	E13.36124	
534	G1	X81.640	Y23.730	E13.79057	F540.000
535	G1	X85.450	Y22.160	E13.93774	
536	G1	X85.440	Y12.770	E14.27309	
537	G1	X89.540	E14.41952		
538	G1	X89.550	Y22.160	E14.75488	
539	G1	X93.360	Y23.730	E14.90205	
540	G1	X100.000	Y17.080	E15.23768	
541	G1	X102.910	Y20.000	E15.38491	
542	G1	X96.270	Y26.640	E15.72028	
543	G1	X97.840	Y30.450	E15.86745	
544	G1	X107.240	E16.20316		
545	G1	Y34.550	E16.34959		
546	G1	X97.840	E16.68530		
547	G1	X96.270	Y38.360	E16.83248	
548	G1	X102.910	Y45.010	E17.16810	
549	G1	X100.000	Y47.920	E17.31508	
550	G1	X93.360	Y41.270	E17.65070	
551	G1	X89.550	Y42.840	E17.79787	
552	G1	X89.540	Y52.240	E18.13359	
553	G1	X85.440	E18.28001		
554	G1	X85.450	Y42.840	E18.61573	
555	G1	X81.640	Y41.270	E18.76290	
556	G1	X74.990	Y47.920	E19.09878	
557	G1	X72.070	Y45.010	E19.24601	
558	G1	X78.730	Y38.360	E19.58213	
559	G1	X77.160	Y34.550	E19.72931	
560	G1	X67.760	E20.06502		
561	G1	Y30.450	E20.21145		
562	G1	X77.160	E20.54716		
563	G1	X78.730	Y26.640	E20.69433	
564	G1	X72.070	Y20.000	E21.03021	
565	G1	X74.980	Y17.090	E21.17719	
566	G1	X81.640	Y23.730	E21.51306	
567	G92	E0			
568	G1	Z1.500	E0.00000	F12000.000	
569	G1	X78.180	Y10.000	E0.25284	F540.000
570	G1	X85.690	E0.38695		
571	G1	Y9.000	E0.40481		
572	G1	X86.690	E0.42267		
573	G1	Y10.000	E0.44052		
574	G1	X88.390	E0.47088		
575	G1	Y9.000	E0.48874		
576	G1	X89.390	E0.50659		
577	G1	Y10.000	E0.52445		
578	G1	X91.090	E0.55481		
579	G1	Y9.000	E0.57267		
580	G1	X92.090	E0.59052		
581	G1	Y10.000	E0.60838		
582	G1	X96.810	E0.69267		
583	G1	X110.000	Y23.180	E1.02564	
584	G1	Y30.690	E1.15974		
585	G1	X111.000	E1.17760		
586	G1	Y31.690	E1.19546		
587	G1	X110.000	E1.21392		
588	G1	Y33.390	E1.24367		
589	G1	X111.000	E1.26153		
590	G1	Y34.390	E1.27939		
591	G1	X110.000	E1.29724		
592	G1	Y41.810	E1.42974		
593	G1	X104.680	Y47.130	E1.56409	
594	G1	X105.380	Y47.830	E1.58177	
595	G1	X104.680	Y48.530	E1.59945	
596	G1	X103.980	Y47.830	E1.61713	
597	G1	X96.810	Y55.000	E1.79820	
598	G1	X89.300	E1.93230		
599	G1	Y56.000	E1.95016		
600	G1	X88.300	E1.96802		
601	G1	Y55.000	E1.98588		
602	G1	X83.900	E2.06445		
603	G1	Y56.000	E2.08230		
604	G1	X82.900	E2.10016		

605	G1	Y55.000	E2.11802	
606	G1	X78.180	E2.20230	
607	G1	X70.960	Y47.780	E2.38464
608	G1	X70.260	Y48.480	E2.40231
609	G1	X69.550	Y47.780	E2.42012
610	G1	X70.250	Y47.080	E2.43780
611	G1	X65.000	Y41.810	E2.57063
612	G1	Y31.600	E2.75295	
613	G1	X64.000	E2.77081	
614	G1	Y30.600	E2.78867	
615	G1	X65.000	E2.80652	
616	G1	Y28.900	E2.83688	
617	G1	X64.000	E2.85474	
618	G1	Y27.900	E2.87260	
619	G1	X65.000	E2.89045	
620	G1	Y23.180	E2.97474	
621	G1	X74.120	Y14.060	E3.20505
622	G1	X73.420	Y13.360	E3.22273
623	G1	X74.130	Y12.650	E3.24066
624	G1	X74.830	Y13.350	E3.25834
625	G1	X78.180	Y10.000	E3.34294
626	G1	X78.590	Y11.000	E3.38154 F1800.000
627	G1	X96.400	E4.01761	
628	G1	X109.000	Y23.590	E4.65375
629	G1	Y41.400	E5.28983	
630	G1	X96.400	Y54.000	E5.92622
631	G1	X78.590	E6.56229	
632	G1	X66.000	Y41.400	E7.19844
633	G1	Y23.590	E7.83451	
634	G1	X78.590	Y11.000	E8.47040
635	G1	X79.010	Y12.000	E8.50914
636	G1	X95.990	E9.11556	
637	G1	X108.000	Y24.010	E9.72216
638	G1	Y40.990	E10.32859	
639	G1	X95.990	Y53.000	E10.93519
640	G1	X79.010	E11.54161	
641	G1	X67.000	Y40.990	E12.14821
642	G1	Y24.010	E12.75464	
643	G1	X79.010	Y12.000	E13.36124
644	G1	X81.640	Y23.730	E13.79057 F540.000
645	G1	X85.450	Y22.160	E13.93774
646	G1	X85.440	Y12.770	E14.27309
647	G1	X89.540	E14.41952	
648	G1	X89.550	Y22.160	E14.75488
649	G1	X93.360	Y23.730	E14.90205
650	G1	X100.000	Y17.080	E15.23768
651	G1	X102.910	Y20.000	E15.38491
652	G1	X96.270	Y26.640	E15.72028
653	G1	X97.840	Y30.450	E15.86745
654	G1	X107.240	E16.20316	
655	G1	Y34.550	E16.34959	
656	G1	X97.840	E16.68530	
657	G1	X96.270	Y38.360	E16.83248
658	G1	X102.910	Y45.010	E17.16810
659	G1	X100.000	Y47.920	E17.31508
660	G1	X93.360	Y41.270	E17.65070
661	G1	X89.550	Y42.840	E17.79787
662	G1	X89.540	Y52.240	E18.13359
663	G1	X85.440	E18.28001	
664	G1	X85.450	Y42.840	E18.61573
665	G1	X81.640	Y41.270	E18.76290
666	G1	X74.990	Y47.920	E19.09878
667	G1	X72.070	Y45.010	E19.24601
668	G1	X78.730	Y38.360	E19.58213
669	G1	X77.160	Y34.550	E19.72931
670	G1	X67.760	E20.06502	
671	G1	Y30.450	E20.21145	
672	G1	X77.160	E20.54716	
673	G1	X78.730	Y26.640	E20.69433
674	G1	X72.070	Y20.000	E21.03021
675	G1	X74.980	Y17.090	E21.17719
676	G1	X81.640	Y23.730	E21.51306
677	G92	E0		
678	G1	Z1.800	E0.00000	F12000.000
679	G1	X78.180	Y10.000	E0.25284 F540.000
680	G1	X85.690	E0.38695	
681	G1	Y9.000	E0.40481	
682	G1	X86.690	E0.42267	
683	G1	Y10.000	E0.44052	
684	G1	X88.390	E0.47088	
685	G1	Y9.000	E0.48874	
686	G1	X89.390	E0.50659	
687	G1	Y10.000	E0.52445	
688	G1	X91.090	E0.55481	
689	G1	Y9.000	E0.57267	
690	G1	X92.090	E0.59052	
691	G1	Y10.000	E0.60838	
692	G1	X96.810	E0.69267	
693	G1	X110.000	Y23.180	E1.02564
694	G1	Y30.690	E1.15974	
695	G1	X111.000	E1.17760	

696	G1	Y31.690	E1.19546	
697	G1	X110.000	E1.21332	
698	G1	Y33.390	E1.24367	
699	G1	X111.000	E1.26153	
700	G1	Y34.390	E1.27939	
701	G1	X110.000	E1.29724	
702	G1	Y41.810	E1.42974	
703	G1	X104.680	Y47.130	E1.56409
704	G1	X105.380	Y47.830	E1.58177
705	G1	X104.680	Y48.530	E1.59945
706	G1	X103.980	Y47.830	E1.61713
707	G1	X96.810	Y55.000	E1.79820
708	G1	X89.300	E1.93230	
709	G1	Y56.000	E1.95016	
710	G1	X88.300	E1.96802	
711	G1	Y55.000	E1.98588	
712	G1	X83.900	E2.06445	
713	G1	Y56.000	E2.08230	
714	G1	X82.900	E2.10016	
715	G1	Y55.000	E2.11802	
716	G1	X78.180	E2.20230	
717	G1	X70.960	Y47.780	E2.38464
718	G1	X70.260	Y48.480	E2.40231
719	G1	X69.550	Y47.780	E2.42012
720	G1	X70.250	Y47.080	E2.43780
721	G1	X65.000	Y41.810	E2.57063
722	G1	Y31.600	E2.75295	
723	G1	X64.000	E2.77081	
724	G1	Y30.600	E2.78867	
725	G1	X65.000	E2.80652	
726	G1	Y28.900	E2.83688	
727	G1	X64.000	E2.85474	
728	G1	Y27.900	E2.87260	
729	G1	X65.000	E2.89045	
730	G1	Y23.180	E2.97474	
731	G1	X74.120	Y14.060	E3.20505
732	G1	X73.420	Y13.360	E3.22273
733	G1	X74.130	Y12.650	E3.24066
734	G1	X74.830	Y13.350	E3.25834
735	G1	X78.180	Y10.000	E3.34294
736	G1	X78.590	Y11.000	E3.38154 F1800.000
737	G1	X96.400	E4.01761	
738	G1	X109.000	Y23.590	E4.65375
739	G1	Y41.400	E5.28983	
740	G1	X96.400	Y54.000	E5.92622
741	G1	X78.590	E6.56229	
742	G1	X66.000	Y41.400	E7.19844
743	G1	Y23.590	E7.83451	
744	G1	X78.590	Y11.000	E8.47040
745	G1	X79.010	Y12.000	E8.50914
746	G1	X95.990	E9.11556	
747	G1	X108.000	Y24.010	E9.72216
748	G1	Y40.990	E10.32859	
749	G1	X95.990	Y53.000	E10.93519
750	G1	X79.010	E11.54161	
751	G1	X67.000	Y40.990	E12.14821
752	G1	Y24.010	E12.75464	
753	G1	X79.010	Y12.000	E13.36124
754	G1	X81.640	Y23.730	E13.79057 F540.000
755	G1	X85.450	Y22.160	E13.93774
756	G1	X85.440	Y12.770	E14.27309
757	G1	X89.540	E14.41952	
758	G1	X89.550	Y22.160	E14.75488
759	G1	X93.360	Y23.730	E14.90205
760	G1	X100.000	Y17.080	E15.23768
761	G1	X102.910	Y20.000	E15.38491
762	G1	X96.270	Y26.640	E15.72028
763	G1	X97.840	Y30.450	E15.86745
764	G1	X107.240	E16.20316	
765	G1	Y34.550	E16.34959	
766	G1	X97.840	E16.68530	
767	G1	X96.270	Y38.360	E16.83248
768	G1	X102.910	Y45.010	E17.16810
769	G1	X100.000	Y47.920	E17.31508
770	G1	X93.360	Y41.270	E17.65070
771	G1	X89.550	Y42.840	E17.79787
772	G1	X89.540	Y52.240	E18.13359
773	G1	X85.440	E18.28001	
774	G1	X85.450	Y42.840	E18.61573
775	G1	X81.640	Y41.270	E18.76290
776	G1	X74.990	Y47.920	E19.09878
777	G1	X72.070	Y45.010	E19.24601
778	G1	X78.730	Y38.360	E19.58213
779	G1	X77.160	Y34.550	E19.72931
780	G1	X67.760	E20.06502	
781	G1	Y30.450	E20.21145	
782	G1	X77.160	E20.54716	
783	G1	X78.730	Y26.640	E20.69433
784	G1	X72.070	Y20.000	E21.03021
785	G1	X74.980	Y17.090	E21.17719
786	G1	X81.640	Y23.730	E21.51306

787	G92 E0
788	G1 Z2.100 E0.00000 F12000.000
789	G1 X79.010 Y12.000 E0.42933 F1800.000
790	G1 X95.990 E1.03576
791	G1 X108.000 Y24.010 E1.64235
792	G1 Y40.990 E2.24878
793	G1 X95.990 Y53.000 E2.85538
794	G1 X79.010 E3.46181
795	G1 X67.000 Y40.990 E4.06840
796	G1 Y24.010 E4.67483
797	G1 X79.010 Y12.000 E5.28143
798	G1 X81.640 Y23.730 E5.71076 F540.000
799	G1 X85.450 Y22.160 E5.85793
800	G1 X85.440 Y12.770 E6.19329
801	G1 X89.540 E6.33972
802	G1 X89.550 Y22.160 E6.67507
803	G1 X93.360 Y23.730 E6.82225
804	G1 X100.000 Y17.080 E7.15787
805	G1 X102.910 Y20.000 E7.30510
806	G1 X96.270 Y26.640 E7.64047
807	G1 X97.840 Y30.450 E7.78764
808	G1 X107.240 E8.12336
809	G1 Y34.550 E8.26978
810	G1 X97.840 E8.60550
811	G1 X96.270 Y38.360 E8.75267
812	G1 X102.910 Y45.010 E9.08829
813	G1 X100.000 Y47.920 E9.23527
814	G1 X93.360 Y41.270 E9.57089
815	G1 X89.550 Y42.840 E9.71806
816	G1 X89.540 Y52.240 E10.05378
817	G1 X85.440 E10.20021
818	G1 X85.450 Y42.840 E10.53592
819	G1 X81.640 Y41.270 E10.68309
820	G1 X74.990 Y47.920 E11.01897
821	G1 X72.070 Y45.010 E11.16620
822	G1 X78.730 Y38.360 E11.50233
823	G1 X77.160 Y34.550 E11.64950
824	G1 X67.760 E11.98521
825	G1 Y30.450 E12.13164
826	G1 X77.160 E12.46736
827	G1 X78.730 Y26.640 E12.61453
828	G1 X72.070 Y20.000 E12.95040
829	G1 X74.980 Y17.090 E13.09738
830	G1 X81.640 Y23.730 E13.43326
831	G92 E0
832	G1 Z2.400 E0.00000 F12000.000
833	G1 X79.010 Y12.000 E0.42933 F1800.000
834	G1 X95.990 E1.03576
835	G1 X108.000 Y24.010 E1.64235
836	G1 Y40.990 E2.24878
837	G1 X95.990 Y53.000 E2.85538
838	G1 X79.010 E3.46181
839	G1 X67.000 Y40.990 E4.06840
840	G1 Y24.010 E4.67483
841	G1 X79.010 Y12.000 E5.28143
842	G1 X81.640 Y23.730 E5.71076 F540.000
843	G1 X85.450 Y22.160 E5.85793
844	G1 X85.440 Y12.770 E6.19329
845	G1 X89.540 E6.33972
846	G1 X89.550 Y22.160 E6.67507
847	G1 X93.360 Y23.730 E6.82225
848	G1 X100.000 Y17.080 E7.15787
849	G1 X102.910 Y20.000 E7.30510
850	G1 X96.270 Y26.640 E7.64047
851	G1 X97.840 Y30.450 E7.78764
852	G1 X107.240 E8.12336
853	G1 Y34.550 E8.26978
854	G1 X97.840 E8.60550
855	G1 X96.270 Y38.360 E8.75267
856	G1 X102.910 Y45.010 E9.08829
857	G1 X100.000 Y47.920 E9.23527
858	G1 X93.360 Y41.270 E9.57089
859	G1 X89.550 Y42.840 E9.71806
860	G1 X89.540 Y52.240 E10.05378
861	G1 X85.440 E10.20021
862	G1 X85.450 Y42.840 E10.53592
863	G1 X81.640 Y41.270 E10.68309
864	G1 X74.990 Y47.920 E11.01897
865	G1 X72.070 Y45.010 E11.16620
866	G1 X78.730 Y38.360 E11.50233
867	G1 X77.160 Y34.550 E11.64950
868	G1 X67.760 E11.98521
869	G1 Y30.450 E12.13164
870	G1 X77.160 E12.46736
871	G1 X78.730 Y26.640 E12.61453
872	G1 X72.070 Y20.000 E12.95040
873	G1 X74.980 Y17.090 E13.09738
874	G1 X81.640 Y23.730 E13.43326
875	G92 E0
876	G1 Z2.700 E0.00000 F12000.000
877	G1 X79.010 Y12.000 E0.42933 F1800.000

```
878 G1 X95.990 E1.03576
879 G1 X108.000 Y24.010 E1.64235
880 G1 Y40.990 E2.24878
881 G1 X95.990 Y53.000 E2.85538
882 G1 X79.010 E3.46181
883 G1 X67.000 Y40.990 E4.06840
884 G1 Y24.010 E4.67483
885 G1 X79.010 Y12.000 E5.28143
886 G1 X81.640 Y23.730 E5.71076 F540.000
887 G1 X85.450 Y22.160 E5.85793
888 G1 X85.440 Y12.770 E6.19329
889 G1 X89.540 E6.33972
890 G1 X89.550 Y22.160 E6.67507
891 G1 X93.360 Y23.730 E6.82225
892 G1 X100.000 Y17.080 E7.15787
893 G1 X102.910 Y20.000 E7.30510
894 G1 X96.270 Y26.640 E7.64047
895 G1 X97.840 Y30.450 E7.78764
896 G1 X107.240 E8.12336
897 G1 Y34.550 E8.26978
898 G1 X97.840 E8.60550
899 G1 X96.270 Y38.360 E8.75267
900 G1 X102.910 Y45.010 E9.08829
901 G1 X100.000 Y47.920 E9.23527
902 G1 X93.360 Y41.270 E9.57089
903 G1 X89.550 Y42.840 E9.71806
904 G1 X89.540 Y52.240 E10.05378
905 G1 X85.440 E10.20021
906 G1 X85.450 Y42.840 E10.53592
907 G1 X81.640 Y41.270 E10.68309
908 G1 X74.990 Y47.920 E11.01897
909 G1 X72.070 Y45.010 E11.16620
910 G1 X78.730 Y38.360 E11.50233
911 G1 X77.160 Y34.550 E11.64950
912 G1 X67.760 E11.98521
913 G1 Y30.450 E12.13164
914 G1 X77.160 E12.46736
915 G1 X78.730 Y26.640 E12.61453
916 G1 X72.070 Y20.000 E12.95040
917 G1 X74.980 Y17.090 E13.09738
918 G1 X81.640 Y23.730 E13.43326
919 M104 S0 ; turn off temperature
920 G28 X0 ; home X axis
921 M84 ; disable motors
922 M140 S0 ; turn off bed
923
```