



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
RESOLUÇÃO DE PROBLEMAS EM ENGENHARIA
PROFESSOR AFONSO FERREIRA MIGUEL

STOCK SMART

Curitiba
Junho de 2009

LEANDRO MALICHESKI DE MACEDO

TAISA DAIANA DA COSTA

STOCK SMART

Projeto integrado apresentado como requisito parcial para o programa de aprendizagem das disciplinas de Resolução de Problemas em Engenharia e Física III, do Curso de Engenharia de Computação da Pontifícia Universidade Católica do Paraná. Orientadores: Afonso ferreira Miguel e Gil Marcos Jess.

Curitiba
Junho de 2009

RESUMO

Atualmente os grandes estabelecimentos comerciais necessitam de mais rapidez para vencer a demanda de pedidos, principalmente em dias próximos a datas comemorativas, pois nestas datas a procura dos produtos pelos consumidores aumenta. E muitas vezes o processo de entrega do pedido demora, acarretando na perda da qualidade do atendimento.

Pensando nisso o Stock Smart foi criado. Trata-se de um estoque automatizado em pequena escala, no qual os objetos armazenados nele são retirados por mecanismos controlados por computador. Este projeto é uma base para a criação de um futuro estoque automatizado em escala real.

Palavras - chave: Agilidade, Estoque, Produtos.

SUMÁRIO

1.INTRODUÇÃO	5
2.OBJETIVOS	6
2.1 Geral.....	6
2.2 Especifico.....	6
3.MATERIAIS UTILIZADOS	7
3.1 Hardware.....	7
3.2 Software.....	7
3.3 Estrutura.....	8
3.4 Equipamentos.....	8
4.DESCRICÃO DO PROJETO	9
4.1 Descrição Geral.....	9
4.2 Descrição Detalhada.....	10
4.2.1 Circuitos Elétricos.....	11
4.2.1.1 Módulo principal da placa M-4 e do Motor de Passo.....	11
4.2.1.2 Placa que comanda o motor DC.....	11
4.2.2 Software.....	11
4.2.2.1 C++.....	11
4.2.2.2 Parte Implementada.....	13
4.2.2.3 Criação dos Botões.....	25
5.PROBLEMAS ENCONTRADOS	26
6.CONCLUSÃO	27
7.ANEXOS	28

1. INTRODUÇÃO

No mercado atualmente há uma grande demanda de pedidos e entregas de mercadorias. Mas isto acontece em um ritmo que poderia ser acelerado e conseqüentemente melhorado. Em uma loja de calçados, por exemplo, do momento da realização do pedido até a hora da prova do calçado há um grande período de espera, pois o vendedor precisa procurar o calçado no estoque, achar o número do pé do cliente, a cor de preferência, e muitas vezes volta sem o calçado ou com outro diferente, e isto acontece porque ele não sabe se o calçado que possui todas as características exigidas pelo cliente já foi vendido.

Mas qual a vantagem de acelerar o processo de movimentação dos produtos? Simplesmente pelo fato de que o consumidor procura agilidade e qualidade no atendimento. O que torna o fato de os produtos chegarem rapidamente nas mãos de quem os solicita uma característica atrativa do estabelecimento, portanto o número de clientes aumenta e há um acréscimo no lucro.

No caso das lojas de calçados a vantagem estaria na rapidez com que as caixas de sapato seriam transportadas, facilitando até mesmo a vida dos vendedores e proporcionando o bem estar comum.

O Stock Smart trabalha uma forma de agilizar as entregas destes e de outros tipos de produtos. Inovando o atendimento a pronta entrega e melhorando a circulação das mercadorias. Composto por mecanismos que em conjunto realizam a busca de objetos numa prateleira, ele se torna um projeto base para o estoque automatizado de um grande estabelecimento.

2. OBJETIVOS

2.2 Geral

O Stock Smart é um estoque que consiste em uma pequena prateleira e alguns mecanismos de busca controlados por um software, que terão a função de transportar produtos que estão guardados nesta prateleira, sem a necessidade de intervenção humana.

2.3 Específicos

1. Trabalhar com motores de passo, a fim de utilizá-los na estrutura responsável pela movimentação dos objetos;
2. Trabalhar com o motor DC;
3. Utilização da placa de aquisição e controle M-4;
4. Utilização da Classe C++ **Cserial**, para o software de controle dos mecanismos da estrutura;
5. Cd de dados contendo vídeos, fotos do projeto, código-fonte e documentação.

3. MATERIAIS UTILIZADOS

3.1 Hardware

- 1 microcontrolador PIC 16F876-4 (módulo principal);
- 1 cristal de 4MHZ (módulo principal);
- 3 resistores de 1K ohm (módulo principal e motor DC);
- 1 resistor de 330 ohms (módulo principal);
- 1 interface serial MAX-232 (módulo de comunicação serial);
- 4 capacitores de 10 micro farad / 25 volts (módulo de comunicação serial);
- 12 resistores de 470 ohms (driver do motor de passo);
- 12 transistores TIP 122 (driver do motor de passo);
- 14 diodos 1N4007 (driver do motor de passo e motor DC);
- 2 relés 6V e 15A/120VAC (motor DC);
- 2 transistores BC548 (motor DC);
- Conectores;
- Leds;
- Placa de fenolite;
- Dissipadores de calor;
- Cabos flat.

3.2 Software

- Software Eagle;
- Microsoft Visual Studio C++ 6.0;
- Hyper terminal do Windows XP;
- MPLAB.

3.3 Estrutura

- Madeira;
- 3 motores de passo;
- 1 motor DC (inserido em um drive de cd);
- Eixos de impressora;

- Correias, roldanas;

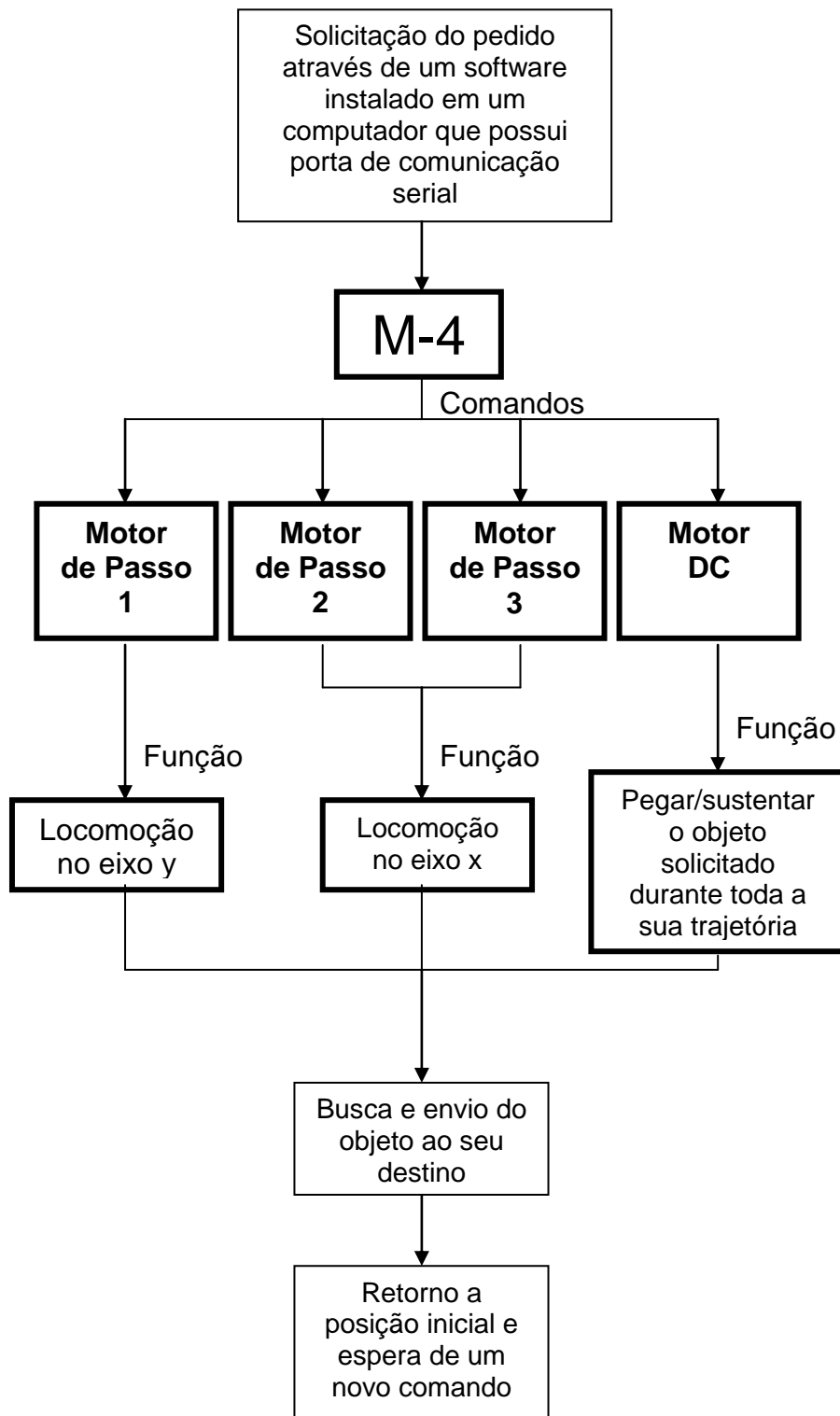
3.4 Equipamentos

- 2 fontes de tensão (5-12 volts);
- Computador com porta de comunicação serial;
- Gravador PICStart Plus;
- Protoboard;
- Alicates de bico e de corte.

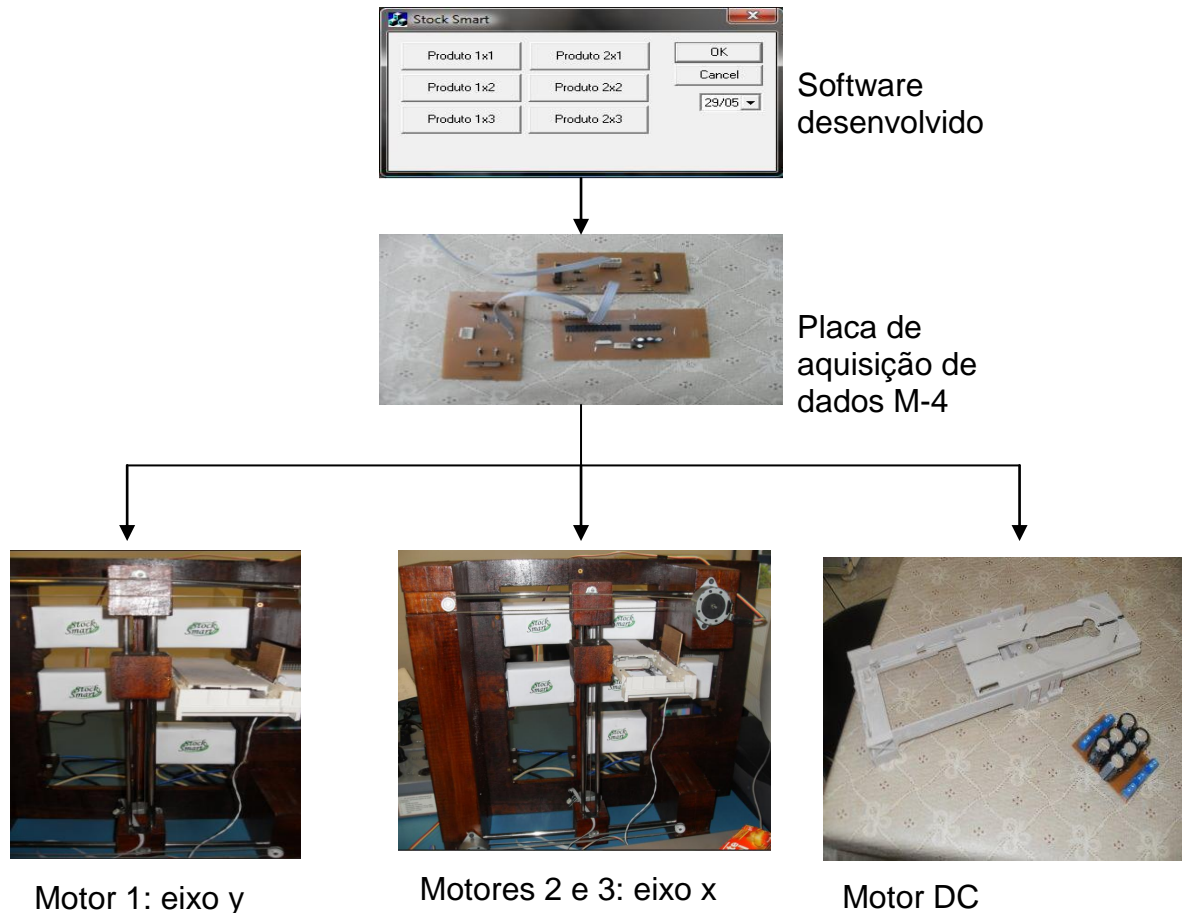
4. DESCRIÇÃO DO PROJETO

4.1 Geral

O funcionamento do projeto se dá pelo seguinte diagrama:



4.2 Detalhada



O software foi desenvolvido na linguagem de programação C++, utilizando-se como base a classe CSerial. Ele comanda através da placa de aquisição M-4, todos os motores correspondentes a estrutura como um todo.

Para isto, foi preciso a utilização de um computador com porta de comunicação serial, pois a placa M-4 recebe e envia dados através da porta serial do computador.

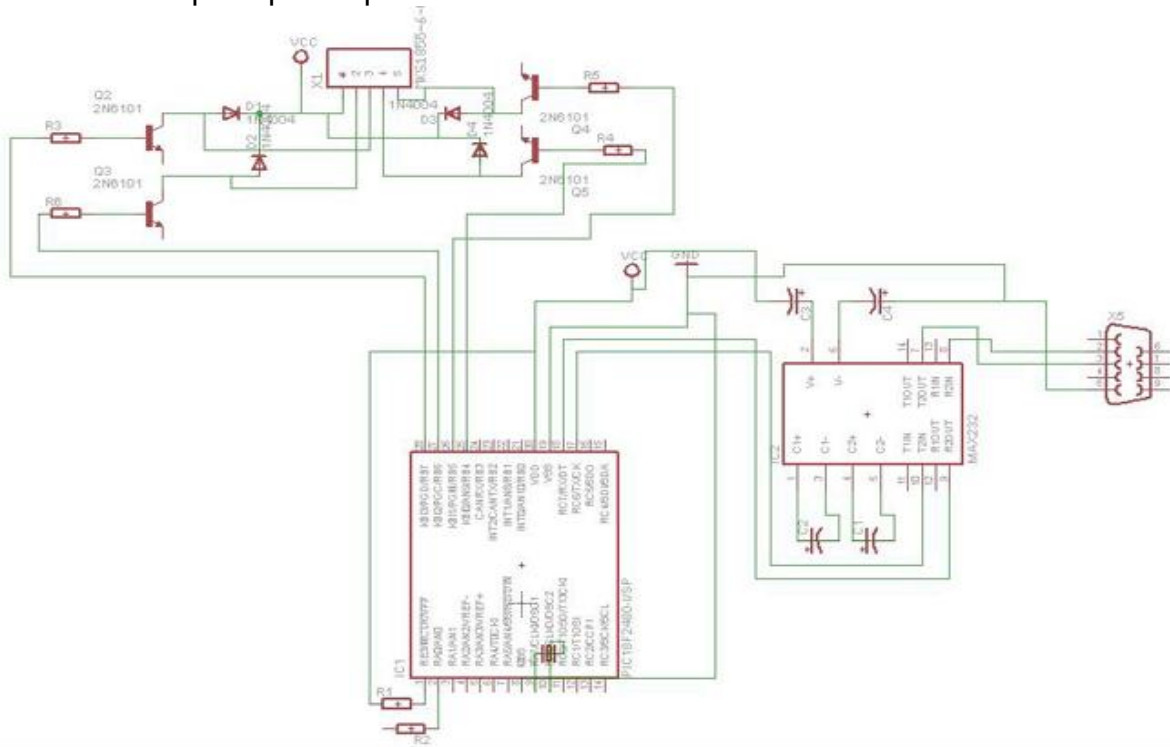
Os comandos aos motores foram, primeiramente, enviados pelo software Hyper terminal do Windows XP, a fim de realizar testes primordiais para a elaboração da programação.

A estrutura foi desenvolvida a partir de madeiras e eixos de impressora (conforme diagrama acima).

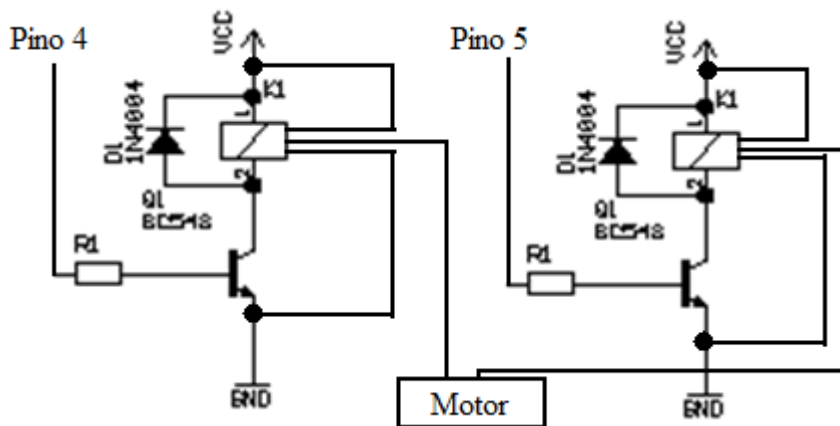
O drive de cd foi utilizado como a estrutura que “pega” e dá apoio ao objeto durante a movimentação.

4.2.1 Circuitos elétricos

4.2.1.1 Módulo principal da placa M-4 e do Motor de Passo



4.2.1.2 Placa que comanda o motor DC



4.2.2 Software

4.2.2.1 C++

```
// Serial.h: interface for the CSerial class.
//
////////////////////////////////////

#ifdef AFX_SERIAL_H__73F2A101_3DC8_11D2_ACEE_00600816809F__INCLUDED_
#define AFX_SERIAL_H__73F2A101_3DC8_11D2_ACEE_00600816809F__INCLUDED_
```

```

#if _MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000

#define READ_TIMEOUT      2500           // milliseconds
#define WRITE_TIMEOUT     2000           // milliseconds
#define READ_TIMEOUT_ABORT 1000         // milliseconds
#define WRITE_TIMEOUT_ABORT 1000        // milliseconds
#define CLEAR_BUFFER_SIZE 1
#define BLOCK_BYTES       64             // block size
#define COMM_TENTATIVAS 3

typedef enum {Nothing, PauseRxTx, Tx, Rx, CloseRXTX} SerialActions;

class CSmdStatusSerial
{
protected:
    BOOL Abort;
public:
    void SetAbortStatus(BOOL status);
    BOOL GetAbortStatus();
    BOOL NewFile;

    CSmdStatusSerial()
    {
        Abort = FALSE;
        NewFile=FALSE;
    };

    ~CSmdStatusSerial()
    {
    };
};

extern CSmdStatusSerial SmdStatusSerial;

class CSerial : public CObject
{
    DCB pDcb;
public:
    int ReadData(void *ponteiro, int numeroBytesALer);
    DWORD SendData(void* ponteiro, WORD numeroBytes);
    COMMTIMEOUTS OldTimeOuts;
    DWORD dwError;
    HANDLE hCom;

    CSerial();
    virtual ~CSerial();
    static WORD CalculaChksum(BYTE* pBytesPrograma, UINT nQtde);
    virtual void FillDcb(DCB *dcb);
    virtual void FillTimeout(COMMTIMEOUTS *to);
    virtual BOOL ChangeBaudRate(DWORD br);
    virtual void AbortNotify();
    virtual BOOL AbortTest();
    virtual void Idle(SerialActions pAction, DWORD pTotalBytes, DWORD
pCurrentBytes);
    virtual BOOL WriteArray(BYTE* pData, DWORD pNBytes, DWORD&
pNBytesWritten);
    virtual BOOL ReadArray(BYTE* pData, DWORD pNBytes, DWORD&
pNBytesReaded);
    virtual BOOL ClearReadBuffer();
};

```

```

virtual BOOL PulseDTR();
virtual BOOL TurnOffDTR();
virtual BOOL TurnOnDTR();
virtual BOOL TurnOffRTS();
virtual BOOL TurnOnRTS();
virtual BOOL Close();
virtual BOOL Open(LPCTSTR port,WORD Baudrate);

private:
    virtual BOOL WriteShortArray(BYTE* pData,DWORD pNBytes, DWORD&
pNBytesWritten);
    virtual BOOL ReadShortArray(BYTE* pData,DWORD pNBytes, DWORD&
pNBytesReaded);
};

#endif //
!defined(AFX_SERIAL_H__73F2A101_3DC8_11D2_ACEE_00600816809F__INCLUDED_)

```

4.2.2.2 Parte implementada

```

void CStockSmartDlg::OnButton1()
{
    CSerial Botaol;

    //Comandos Enviados Via Serial
    char *Velocidade = "\rSMVC\r";
    //Comandos do Motor do Eixo Y
    char *MotorY1 = "\rSM2RR\r";
    char *MotorY2 = "\rSM20150\r";
    char *MotorY3 = "\rSM20020\r";
    char *MotorY4 = "\rSM2RL\r";
    char *MotorY5 = "\rSM200A0\r";
    char *MotorY6 = "\rSM20030\r";
    char *MotorY7 = "\rSM2FREE\r";
    //Comandos dos Motores do Eixo x
    char *MotorX1 = "\rSM1RR\r";
    char *MotorX3 = "\rSM101A7\r";
    char *MotorX4 = "\rSM1RL\r";
    char *MotorX5= "\rSM1FREE\r";
    //Comandos do Motor DC
    char *MotorDC1 = "\rWR04\r";
    char *MotorDC2 = "\rWR08\r";
    char *MotorDC3 = "\rWR00\r";

    //Trava Os Demais Botões
    GetDlgItem(IDC_BUTTON2)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON3)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON4)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON5)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON6)->EnableWindow(FALSE);
    GetDlgItem(IDOK)->EnableWindow(FALSE);
    GetDlgItem(IDCANCEL)->EnableWindow(FALSE);

    //Teste Para Que A Porta COM1 Seja Aberta
    if(Botaol.Open("COM1",9600)){
        //Acelera Os Motores
        Botaol.SendData(Velocidade,strlen(Velocidade));
        Sleep(10000);
        //Envia Comomandos
        Botaol.SendData(MotorX5,strlen(MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep(10000);
    }
}

```

```

        Botaol.SendData (MotorY1, strlen (MotorY4)); //Motor Y Gira
Sentido Horario
        Sleep (10000);

        Botaol.SendData (MotorY2, strlen (MotorY2)); //Sobe até o
produto 1x1
        Sleep (13000);
        Botaol.SendData (MotorDC2, strlen (MotorDC1)); //Motor DC
Vai Para Frente
        Sleep (14000);
        Botaol.SendData (MotorY3, strlen (MotorY3)); //Motor Y Sobe
um pouco
        Sleep (10000);
        Botaol.SendData (MotorDC1, strlen (MotorDC2)); //Motor DC
volta
        Sleep (10000);
        Botaol.SendData (MotorDC3, strlen (MotorDC3)); //Desliga
Temporariamente o Motor DC
        Sleep (5000);
        Botaol.SendData (MotorY4, strlen (MotorY4)); //Gira Sentido
Anti-Horario
        Sleep (10000);
        Botaol.SendData (MotorY5, strlen (MotorY5)); //Motor Y Desce
um pouco
        Sleep (10000);
        Botaol.SendData (MotorY7, strlen (MotorY7)); //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botaol.SendData (MotorX1, strlen (MotorX1)); //Gira Sentido
Horario
        Sleep (10000);
        Botaol.SendData (MotorX3, strlen (MotorX3)); //Vai até a
entrega
        Sleep (14000);
        Botaol.SendData (MotorX5, strlen (MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep (10000);
        Botaol.SendData (MotorDC2, strlen (MotorDC1)); //Motor DC
Vai Para Frente
        Sleep (10000);
        Botaol.SendData (MotorY4, strlen (MotorY4)); //Motor Y Gira
Sentido Anti-Horario
        Sleep (14000);
        Botaol.SendData (MotorY6, strlen (MotorY6)); //Motor Y Desce
um pouco
        Sleep (14000);
        Botaol.SendData (MotorY7, strlen (MotorY7)); //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botaol.SendData (MotorDC1, strlen (MotorDC2)); //Motor DC
Volta
        Sleep (10000);
        Botaol.SendData (MotorDC3, strlen (MotorDC3)); //Desliga
Temporariamente o Motor DC
        Sleep (10000);
        Botaol.SendData (MotorX4, strlen (MotorX4)); //Gira Sentido
Anti-Horario
        Sleep (14000);
        Botaol.SendData (MotorX3, strlen (MotorX3)); //Motor X
Retorna ao Inicio
        Sleep (16000);

```

```

        Botaol.SendData (MotorX5, strlen (MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep (10000);
        Botaol.SendData (MotorY4, strlen (MotorY4)); //Motor Y Gira
Sentido Horário
        Sleep (14000);
        Botaol.SendData (MotorY5, strlen (MotorY5)); //Motor Y
Retorna ao Inicio
        Sleep (14000);
        Botaol.SendData (MotorY7, strlen (MotorY7)); //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botaol.Close ();
    }
    //Abilita Os Demais Botões
    GetDlgItem (IDC_BUTTON2) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON3) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON4) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON5) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON6) ->EnableWindow (TRUE);
    GetDlgItem (IDOK) ->EnableWindow (TRUE);
    GetDlgItem (IDCANCEL) ->EnableWindow (TRUE);
}

void CStockSmartDlg::OnButton2 ()
{
    CSerial Botaol2;
    //Comandos Enviados Via Serial
    char *Velocidade = "\rSMVC\r";
    //Comandos do Motor do Eixo Y
    char *MotorY1 = "\rSM2RR\r";
    char *MotorY2 = "\rSM200A0\r";
    char *MotorY3 = "\rSM20020\r";
    char *MotorY4 = "\rSM2RL\r";
    char *MotorY6 = "\rSM2FREE\r";
    //Comandos dos Motores do Eixo x
    char *MotorX1 = "\rSM1RR\r";
    char *MotorX3 = "\rSM101A7\r";
    char *MotorX4 = "\rSM1RL\r";
    char *MotorX5 = "\rSM1FREE\r";
    //Comandos do Motor DC
    char *MotorDC1 = "\rWR04\r";
    char *MotorDC2 = "\rWR08\r";
    char *MotorDC3 = "\rWR00\r";

    GetDlgItem (IDC_BUTTON1) ->EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON3) ->EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON4) ->EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON5) ->EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON6) ->EnableWindow (FALSE);
    GetDlgItem (IDOK) ->EnableWindow (FALSE);
    GetDlgItem (IDCANCEL) ->EnableWindow (FALSE);

    if (Botaol2.Open ("COM1", 9600)) {
        //Acelera Os Motores
        Botaol2.SendData (Velocidade, strlen (Velocidade));
        Sleep (5000);
        //Envia Comandos
        Botaol2.SendData (MotorX5, strlen (MotorX5)); //Desliga
Temporariamente o Motor X
    }
}

```

```

                Sleep(10000);
Horario          Botao2.SendData(MotorY1, strlen(MotorY1)); //Gira Sentido
                Sleep(10000);
produto 1x2      Botao2.SendData(MotorY2, strlen(MotorY2)); //Sobe até o
                Sleep(14000);
Vai Para Frente Botao2.SendData(MotorDC2, strlen(MotorDC2)); //Motor DC
                Sleep(10000);
um pouco         Botao2.SendData(MotorY3, strlen(MotorY3)); //Motor Y Sobe
                Sleep(10000);
volta           Botao2.SendData(MotorDC1, strlen(MotorDC1)); //Motor DC
                Sleep(10000);
Temporariamente o Motor DC Botao2.SendData(MotorDC3, strlen(MotorDC3)); //Desliga
                Sleep(10000);
Temporariamente o Motor Y Botao2.SendData(MotorY6, strlen(MotorY6)); //Desliga
                Sleep(10000);
entrega         Botao2.SendData(MotorX3, strlen(MotorX3)); //Vai até a
                Sleep(16000);
Temporariamente o Motor X Botao2.SendData(MotorX5, strlen(MotorX5)); //Desliga
                Sleep(10000);
Vai Para Frente Botao2.SendData(MotorDC2, strlen(MotorDC2)); //Motor DC
                Sleep(10000);
Anti-Horario    Botao2.SendData(MotorY4, strlen(MotorY4)); //Gira Sentido
                Sleep(10000);
um pouco         Botao2.SendData(MotorY3, strlen(MotorY3)); //Motor Y Desce
                Sleep(10000);
Temporariamente o Motor Y Botao2.SendData(MotorY6, strlen(MotorY6)); //Desliga
                Sleep(10000);
Volta           Botao2.SendData(MotorDC1, strlen(MotorDC1)); //Motor DC
                Sleep(10000);
Temporariamente o Motor DC Botao2.SendData(MotorDC3, strlen(MotorDC3)); //Desliga
                Sleep(10000);
Anti-Horario    Botao2.SendData(MotorX4, strlen(MotorX4)); //Gira Sentido
                Sleep(16000);
Retorna ao Inicio Botao2.SendData(MotorX3, strlen(MotorX3)); //Motor X
                Sleep(16000);
Temporariamente o Motor X Botao2.SendData(MotorX5, strlen(MotorX5)); //Desliga
                Sleep(10000);
Anti-Horario    Botao2.SendData(MotorY4, strlen(MotorY4)); //Gira Sentido
                Sleep(10000);
Retorna ao Inicio Botao2.SendData(MotorY2, strlen(MotorY2)); //Motor Y

```



```

        Sleep(10000);
        Botao2.SendData(MotorY6, strlen(MotorY6)); //Desliga
Temporariamente o Motor Y
        Sleep(10000);
        Botao2.Close();
    }

    GetDlgItem(IDC_BUTTON1)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON3)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON4)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON5)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON6)->EnableWindow(TRUE);
    GetDlgItem(IDOK)->EnableWindow(TRUE);
    GetDlgItem(IDCANCEL)->EnableWindow(TRUE);
}

void CStockSmartDlg::OnButton3()
{
    CSerial Botao3;
    //Comandos Enviados Via Serial
    char *Velocidade = "\rSMVC\r";
    //Comandos do Motor do Eixo Y
    char *MotorY1 = "\rSM2RR\r";
    char *MotorY3 = "\rSM20020\r";
    char *MotorY4 = "\rSM2RL\r";
    char *MotorY5 = "\rSM200A0\r";
    char *MotorY6 = "\rSM2FREE\r";
    //Comandos dos Motores do Eixo x
    char *MotorX1 = "\rSM1RR\r";
    char *MotorX2 = "\rSM10010\r";
    char *MotorX3 = "\rSM10197\r";
    char *MotorX4 = "\rSM1RL\r";
    char *MotorX5 = "\rSM1FREE\r";
    //Comandos do Motor DC
    char *MotorDC1 = "\rWR04\r";
    char *MotorDC2 = "\rWR08\r";
    char *MotorDC3 = "\rWR00\r";

    GetDlgItem(IDC_BUTTON1)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON2)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON4)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON5)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON6)->EnableWindow(FALSE);
    GetDlgItem(IDOK)->EnableWindow(FALSE);
    GetDlgItem(IDCANCEL)->EnableWindow(FALSE);

    if(Botao3.Open("COM1", 9600)) {
        //Acelera Os Motores
        Botao3.SendData(Velocidade, strlen(Velocidade));
        Sleep(5000);
    //Envia Comomandos
        Botao3.SendData(MotorX1, strlen(MotorX1)); //Gira Sentido
Horario
        Sleep(10000);
        //"Dorme" Durante 1000 Milisegundos
        Botao3.SendData(MotorX2, strlen(MotorX2)); //Anda um pouco
        Sleep(10000);
        Botao3.SendData(MotorX5, strlen(MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep(10000);
    }
}

```

```

        Botao3.SendData (MotorDC2, strlen (MotorDC2));           //Motor DC
Vai Para Frente
        Sleep (14000);
        Botao3.SendData (MotorY1, strlen (MotorY1));           //Gira Sentido
Horario
        Sleep (16000);
        Botao3.SendData (MotorY3, strlen (MotorY3));           //Motor Y Sobe
um pouco
        Sleep (16000);
        Botao3.SendData (MotorDC1, strlen (MotorDC1));           //Motor DC
volta
        Sleep (12000);
        Botao3.SendData (MotorDC3, strlen (MotorDC3));           //Desliga
Temporariamente o Motor DC
        Sleep (10000);
        Botao3.SendData (MotorY5, strlen (MotorY5));           //Motor Y Sobe
um pouco
        Sleep (10000);
        Botao3.SendData (MotorY6, strlen (MotorY6));           //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botao3.SendData (MotorX3, strlen (MotorX3));           //Vai até a
entrega
        Sleep (14000);
        Botao3.SendData (MotorX5, strlen (MotorX5));           //Desliga
Temporariamente o Motor X
        Sleep (10000);
        Botao3.SendData (MotorDC2, strlen (MotorDC2));           //Motor DC
Vai Para Frente
        Sleep (10000);
        Botao3.SendData (MotorY4, strlen (MotorY4));           //Gira Sentido
Anti-Horario
        Sleep (10000);
        Botao3.SendData (MotorY3, strlen (MotorY3));           //Motor Y Desce
um pouco
        Sleep (10000);
        Botao3.SendData (MotorY6, strlen (MotorY6));           //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botao3.SendData (MotorDC1, strlen (MotorDC1));           //Motor DC
Volta
        Sleep (10000);
        Botao3.SendData (MotorDC3, strlen (MotorDC3));           //Desliga
Temporariamente o Motor DC
        Sleep (10000);
        Botao3.SendData (MotorX4, strlen (MotorX4));           //Gira Sentido
Anti-Horario
        Sleep (16000);
        Botao3.SendData (MotorX3, strlen (MotorX3));           //Motor X
Retorna ao Inicio
        Sleep (16000);
        Botao3.SendData (MotorX5, strlen (MotorX5));           //Desliga
Temporariamente o Motor X
        Sleep (10000);
        Botao3.SendData (MotorY4, strlen (MotorY4));           //Motor Gira
Sentido Horario
        Sleep (10000);
        Botao3.SendData (MotorY5, strlen (MotorY5));           //Motor Y
Retorna ao Inicio
        Sleep (10000);

```

```

        Botao3.SendData (MotorY6, strlen (MotorY6));           //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botao3.Close ();
    }

    GetDlgItem (IDC_BUTTON1) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON2) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON4) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON5) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON6) -> EnableWindow (TRUE);
    GetDlgItem (IDOK) -> EnableWindow (TRUE);
    GetDlgItem (IDCANCEL) -> EnableWindow (TRUE);
}

void CStockSmartDlg::OnButton4 ()
{
    CSerial Botao4;

    //Comandos Enviados Via Serial
    char *Velocidade = "\rSMVC\r";
    //Comandos do Motor do Eixo Y
    char *MotorY1 = "\rSM2RR\r";
    char *MotorY2 = "\rSM20150\r";
    char *MotorY3 = "\rSM20020\r";
    char *MotorY4 = "\rSM2RL\r";
    char *MotorY5 = "\rSM200A0\r";
    char *MotorY7 = "\rSM2FREE\r";
    char *MotorY8 = "\rSM20030\r";
    //Comandos dos Motores do Eixo x
    char *MotorX1 = "\rSM1RR\r";
    char *MotorX2 = "\rSM100F0\r";
    char *MotorX3 = "\rSM100B7\r";
    char *MotorX4 = "\rSM1RL\r";
    char *MotorX5 = "\rSM1FREE\r";
    char *MotorX6 = "\rSM101A7\r";
    //Comandos do Motor DC
    char *MotorDC1 = "\rWR04\r";
    char *MotorDC2 = "\rWR08\r";
    char *MotorDC3 = "\rWR00\r";

    GetDlgItem (IDC_BUTTON1) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON2) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON3) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON5) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON6) -> EnableWindow (FALSE);
    GetDlgItem (IDOK) -> EnableWindow (FALSE);
    GetDlgItem (IDCANCEL) -> EnableWindow (FALSE);

    if (Botao4.Open ("COM1", 9600)) {
        //Acelera Os Motores
        Botao4.SendData (Velocidade, strlen (Velocidade));
        Sleep (10000);
        //Envia Comomandos
        Botao4.SendData (MotorX1, strlen (MotorX1));           //Gira Sentido
Horario
        Sleep (10000);
        //"Dorme" Durante 1000 Milisegundos
        Botao4.SendData (MotorX2, strlen (MotorX2));           //Anda um pouco
        Sleep (14000);
    }
}

```

```

        Botao4.SendData (MotorX5, strlen (MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep (16000);
        Botao4.SendData (MotorY1, strlen (MotorY1)); //Motor Y Gira
Sentido Horario
        Sleep (14000);

        Botao4.SendData (MotorY2, strlen (MotorY2)); //Sobe até o
produto 2x1
        Sleep (16000);
        Botao4.SendData (MotorDC2, strlen (MotorDC2)); //Motor DC
Vai Para Frente
        Sleep (14000);
        Botao4.SendData (MotorY3, strlen (MotorY3)); //Motor Y Sobe
um pouco
        Sleep (14000);
        Botao4.SendData (MotorDC1, strlen (MotorDC1)); //Motor DC
volta
        Sleep (10000);
        Botao4.SendData (MotorDC3, strlen (MotorDC3)); //Desliga
Temporariamente o Motor DC
        Sleep (14000);
        Botao4.SendData (MotorY4, strlen (MotorY4)); //Gira Sentido
Anti-Horario
        Sleep (14000);
        Botao4.SendData (MotorY5, strlen (MotorY5)); //Motor Y Desce
um pouco
        Sleep (14000);
        Botao4.SendData (MotorY7, strlen (MotorY7)); //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botao4.SendData (MotorX1, strlen (MotorX1)); //Gira Sentido
Horario
        Sleep (14000);
        Botao4.SendData (MotorX3, strlen (MotorX3)); //Vai até a
entrega
        Sleep (14000);
        Botao4.SendData (MotorX5, strlen (MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep (10000);
        Botao4.SendData (MotorDC2, strlen (MotorDC2)); //Motor DC
Vai Para Frente
        Sleep (10000);
        Botao4.SendData (MotorY4, strlen (MotorY4)); //Motor Y Gira
Sentido Anti-Horario
        Sleep (10000);
        Botao4.SendData (MotorY8, strlen (MotorY8)); //Motor Y Desce
um pouco
        Sleep (10000);
        Botao4.SendData (MotorY7, strlen (MotorY7)); //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botao4.SendData (MotorDC1, strlen (MotorDC1)); //Motor DC
Volta
        Sleep (10000);
        Botao4.SendData (MotorDC3, strlen (MotorDC3)); //Desliga
Temporariamente o Motor DC
        Sleep (10000);
        Botao4.SendData (MotorX4, strlen (MotorX4)); //Gira Sentido
Anti-Horario
        Sleep (14000);

```

```

        Botao4.SendData (MotorX6, strlen (MotorX6)); //Motor X
Retorna ao Inicio
        Sleep (16000);
        Botao4.SendData (MotorX5, strlen (MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep (10000);
        Botao4.SendData (MotorY4, strlen (MotorY4)); //Motor Y Gira
Sentido Anti-Horario
        Sleep (10000);
        Botao4.SendData (MotorY5, strlen (MotorY5)); //Motor Y
Retorna ao Inicio
        Sleep (10000);
        Botao4.SendData (MotorY7, strlen (MotorY7)); //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botao4.Close ();
    }

    GetDlgItem (IDC_BUTTON1) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON2) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON3) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON5) -> EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON6) -> EnableWindow (TRUE);
    GetDlgItem (IDOK) -> EnableWindow (TRUE);
    GetDlgItem (IDCANCEL) -> EnableWindow (TRUE);
}

void CStockSmartDlg::OnButton5 ()
{
    CSerial Botao5;
    //Comandos Enviados Via Serial
    char *Velocidade = "\rSMVC\r";
    //Comandos do Motor do Eixo Y
    char *MotorY1 = "\rSM2RR\r";
    char *MotorY2 = "\rSM200A0\r";
    char *MotorY3 = "\rSM20020\r";
    char *MotorY4 = "\rSM2RL\r";
    char *MotorY5 = "\rSM200FB\r";
    char *MotorY6 = "\rSM2FREE\r";
    char *MotorY7 = "\rSM20030\r";
    //Comandos dos Motores do Eixo x
    char *MotorX1 = "\rSM1RR\r";
    char *MotorX2 = "\rSM100F0\r";
    char *MotorX3 = "\rSM100B7\r";
    char *MotorX4 = "\rSM1RL\r";
    char *MotorX5 = "\rSM1FREE\r";
    char *MotorX6 = "\rSM101A7\r";
    //Comandos do Motor DC
    char *MotorDC1 = "\rWR04\r";
    char *MotorDC2 = "\rWR08\r";
    char *MotorDC3 = "\rWR00\r";

    GetDlgItem (IDC_BUTTON1) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON2) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON3) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON4) -> EnableWindow (FALSE);
    GetDlgItem (IDC_BUTTON6) -> EnableWindow (FALSE);
    GetDlgItem (IDOK) -> EnableWindow (FALSE);
    GetDlgItem (IDCANCEL) -> EnableWindow (FALSE);

    if (Botao5.Open ("COM1", 9600)) {

```

```

        //Acelera Os Motores
        Botao5.SendData (Velocidade, strlen (Velocidade));
        Sleep (5000);
        //Envia Comandos
        Botao5.SendData (MotorX1, strlen (MotorX1));           //Gira Sentido
Horario
        Sleep (14000);
        //"Dorme" Durante 1000 Milisegundos
        Botao5.SendData (MotorX2, strlen (MotorX2));           //Anda um pouco
        Sleep (14000);
        Botao5.SendData (MotorX5, strlen (MotorX5));           //Desliga
Temporariamente o Motor X
        Sleep (14000);
        Botao5.SendData (MotorY1, strlen (MotorY1));           //Gira Sentido
Horario
        Sleep (14000);

        Botao5.SendData (MotorY2, strlen (MotorY2));           //Sobe até o
produto 2x2
        Sleep (14000);
        Botao5.SendData (MotorDC2, strlen (MotorDC2));           //Motor DC
Vai Para Frente
        Sleep (14000);
        Botao5.SendData (MotorY3, strlen (MotorY3));           //Motor Y Sobe
um pouco
        Sleep (14000);
        Botao5.SendData (MotorDC1, strlen (MotorDC1));           //Motor DC
volta
        Sleep (14000);
        Botao5.SendData (MotorDC3, strlen (MotorDC3));           //Desliga
Temporariamente o Motor DC
        Sleep (10000);
        Botao5.SendData (MotorY6, strlen (MotorY6));           //Desliga
Temporariamente o Motor Y
        Sleep (14000);
        Botao5.SendData (MotorX1, strlen (MotorX1));           //Gira Sentido
Horario
        Sleep (14000);
        Botao5.SendData (MotorX3, strlen (MotorX3));           //Vai até a
entrega
        Sleep (14000);
        Botao5.SendData (MotorX5, strlen (MotorX5));           //Desliga
Temporariamente o Motor X
        Sleep (14000);
        Botao5.SendData (MotorDC2, strlen (MotorDC2));           //Motor DC
Vai Para Frente
        Sleep (14000);
        Botao5.SendData (MotorY4, strlen (MotorY4));           //Gira Sentido
Anti-Horario
        Sleep (14000);
        Botao5.SendData (MotorY3, strlen (MotorY3));           //Motor Y Desce
um pouco
        Sleep (14000);
        Botao5.SendData (MotorY6, strlen (MotorY6));           //Desliga
Temporariamente o Motor Y
        Sleep (14000);
        Botao5.SendData (MotorDC1, strlen (MotorDC1));           //Motor DC
Volta
        Sleep (10000);
        Botao5.SendData (MotorDC3, strlen (MotorDC3));           //Desliga
Temporariamente o Motor DC

```

```

        Sleep(14000);
        Botao5.SendData(MotorX4, strlen(MotorX4)); //Gira Sentido
Anti-Horario
        Sleep(16000);
        Botao5.SendData(MotorX6, strlen(MotorX6)); //Motor X
Retorna ao Inicio
        Sleep(16000);
        Botao5.SendData(MotorX5, strlen(MotorX5)); //Desliga
Temporariamente o Motor X
        Sleep(10000);
        Botao5.SendData(MotorY4, strlen(MotorY4)); //Gira Sentido
Anti-Horario
        Sleep(16000);
        Botao5.SendData(MotorY2, strlen(MotorY2)); //Motor Y
Retorna ao Inicio
        Sleep(16000);
        Botao5.SendData(MotorY6, strlen(MotorY6)); //Desliga
Temporariamente o Motor Y
        Sleep(16000);
        Botao5.Close();
    }

    GetDlgItem(IDC_BUTTON1)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON2)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON3)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON4)->EnableWindow(TRUE);
    GetDlgItem(IDC_BUTTON6)->EnableWindow(TRUE);
    GetDlgItem(IDOK)->EnableWindow(TRUE);
    GetDlgItem(IDCANCEL)->EnableWindow(TRUE);

}

void CStockSmartDlg::OnButton6()
{
    CSerial Botao6;
    //Comandos Enviados Via Serial
    char *Velocidade = "\rSMVC\r";
    //Comandos do Motor do Eixo Y
    char *MotorY1 = "\rSM2RR\r";
    char *MotorY3 = "\rSM20020\r";
    char *MotorY4 = "\rSM2RL\r";
    char *MotorY5 = "\rSM200A0\r";
    char *MotorY6 = "\rSM2FREE\r";
    char *MotorY7 = "\rSM20030\r";
    //Comandos dos Motores do Eixo x
    char *MotorX1 = "\rSM1RR\r";
    char *MotorX2 = "\rSM100F0\r";
    char *MotorX3 = "\rSM100B7\r";
    char *MotorX4 = "\rSM1RL\r";
    char *MotorX5 = "\rSM1FREE\r";
    char *MotorX6 = "\rSM101A7\r";
    //Comandos do Motor DC
    char *MotorDC1 = "\rWR04\r";
    char *MotorDC2 = "\rWR08\r";
    char *MotorDC3 = "\rWR00\r";

    if(Botao6.Open("COM1", 9600)) {

        GetDlgItem(IDC_BUTTON1)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON2)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON3)->EnableWindow(FALSE);
    }
}

```

```

GetDlgItem(IDC_BUTTON4)->EnableWindow(FALSE);
GetDlgItem(IDC_BUTTON5)->EnableWindow(FALSE);
GetDlgItem(IDOK)->EnableWindow(FALSE);
GetDlgItem(IDCANCEL)->EnableWindow(FALSE);
//Acelera Os Motores
    Botao6.SendData (Velocidade, strlen (Velocidade));
        Sleep (10000);
//Envia Comandos
    Botao6.SendData (MotorX1, strlen (MotorX1));           //Gira Sentido
Horario
        Sleep (10000);
//"Dorme" Durante 1000 Milisegundos
    Botao6.SendData (MotorX2, strlen (MotorX2));           //Anda um pouco
        Sleep (10000);
    Botao6.SendData (MotorX5, strlen (MotorX5));           //Desliga
Temporariamente o Motor X
        Sleep (10000);

    Botao6.SendData (MotorDC2, strlen (MotorDC2));           //Motor DC
Vai Para Frente
        Sleep (10000);
    Botao6.SendData (MotorY1, strlen (MotorY1));           //Gira Sentido
Anti-Horario
        Sleep (10000);
    Botao6.SendData (MotorY3, strlen (MotorY3));           //Motor Y Sobe
um pouco
        Sleep (10000);
    Botao6.SendData (MotorDC1, strlen (MotorDC1));           //Motor DC
volta
        Sleep (10000);
    Botao6.SendData (MotorDC3, strlen (MotorDC3));           //Desliga
Temporariamente o Motor DC
        Sleep (10000);
    Botao6.SendData (MotorY5, strlen (MotorY5));           //Motor Y Sobe
um pouco
        Sleep (10000);
    Botao6.SendData (MotorY6, strlen (MotorY6));           //Desliga
Temporariamente o Motor Y
        Sleep (10000);
    Botao6.SendData (MotorX1, strlen (MotorX1));           //Gira Sentido
Horario
        Sleep (14000);
    Botao6.SendData (MotorX3, strlen (MotorX3));           //Vai até a
entrega
        Sleep (14000);
    Botao6.SendData (MotorX5, strlen (MotorX5));           //Desliga
Temporariamente o Motor X
        Sleep (10000);
    Botao6.SendData (MotorDC2, strlen (MotorDC2));           //Motor DC
Vai Para Frente
        Sleep (10000);
    Botao6.SendData (MotorY4, strlen (MotorY4));           //Gira Sentido
Anti-Horario
        Sleep (10000);
    Botao6.SendData (MotorY3, strlen (MotorY3));           //Motor Y Desce
um pouco
        Sleep (10000);
    Botao6.SendData (MotorY6, strlen (MotorY6));           //Desliga
Temporariamente o Motor Y
        Sleep (10000);

```



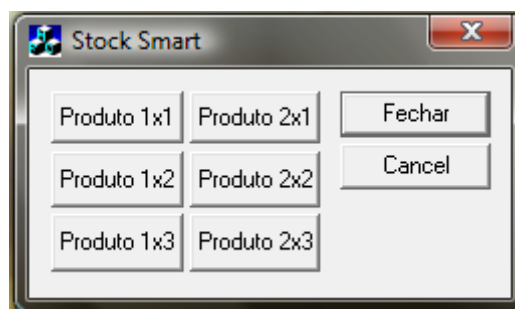
```

        Botao6.SendData (MotorDC1, strlen (MotorDC1));           //Motor DC
Volta
        Sleep (10000);
        Botao6.SendData (MotorDC3, strlen (MotorDC3));           //Desliga
Temporariamente o Motor DC
        Sleep (14000);
        Botao6.SendData (MotorX4, strlen (MotorX4));             //Gira Sentido
Anti-Horario
        Sleep (16000);
        Botao6.SendData (MotorX6, strlen (MotorX6));             //Motor X
Retorna ao Inicio
        Sleep (16000);
        Botao6.SendData (MotorX5, strlen (MotorX5));             //Desliga
Temporariamente o Motor X
        Sleep (16000);
        Botao6.SendData (MotorY4, strlen (MotorY4));             //Motor Gira
Sentido Anti-Horario
        Sleep (10000);
        Botao6.SendData (MotorY5, strlen (MotorY5));             //Motor Y
Retorna ao Inicio
        Sleep (14000);
        Botao6.SendData (MotorY6, strlen (MotorY6));             //Desliga
Temporariamente o Motor Y
        Sleep (10000);
        Botao6.Close ();
    }

    GetDlgItem (IDC_BUTTON1) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON2) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON3) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON4) ->EnableWindow (TRUE);
    GetDlgItem (IDC_BUTTON5) ->EnableWindow (TRUE);
    GetDlgItem (IDOK) ->EnableWindow (TRUE);
    GetDlgItem (IDCANCEL) ->EnableWindow (TRUE);
}

```

4.2.2.3 Criação dos botões



5. PROBLEMAS ENCONTRADOS

Os principais problemas encontrados foram relacionados com a inexperiência da equipe com projetos.

A estrutura como um todo foi complicada de ser construída, devido ao encaixe dos eixos da impressora nos pedaços de madeira que foram confeccionados. Por que se um erro milimétrico fosse cometido a estrutura se comprometeria e o trabalho seria perdido.

Após a montagem com os eixos constatamos que a movimentação da estrutura ficou muito rígida, devido a alguns fatores. Porém este problema foi solucionado com o aumento dos buracos nas madeiras que seguram os eixos.

Outro problema se deu pelo direcionamento dos motores de passo, pelo fato de não possuímos as sequências dos motores. Este foi um dos principais problemas, pois os motores logo que eram ligados nas fontes de tensão ficavam “bobos” e assim era impossível trabalhar. Esse problema também foi solucionado com a ligação de todos os motores ao mesmo GND.

6. CONCLUSÃO

O aprendizado na realização desse projeto se deu através da experiência adquirida com a construção de placas de circuito impresso utilizando softwares específicos, das técnicas de soldagem, da programação, da resolução de erros, na modelagem em madeira, da noção de como é o mercado de trabalho, do trabalho em equipe, do controle de motores de passo e DC e através de Circuitos integrados. Aprendemos a como nos relacionar e firmar compromissos.

7. ANEXOS



Figura1: construção da estrutura

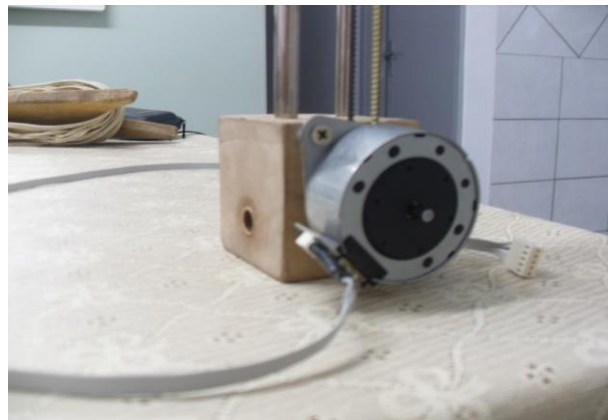


Figura 2: motor de passo

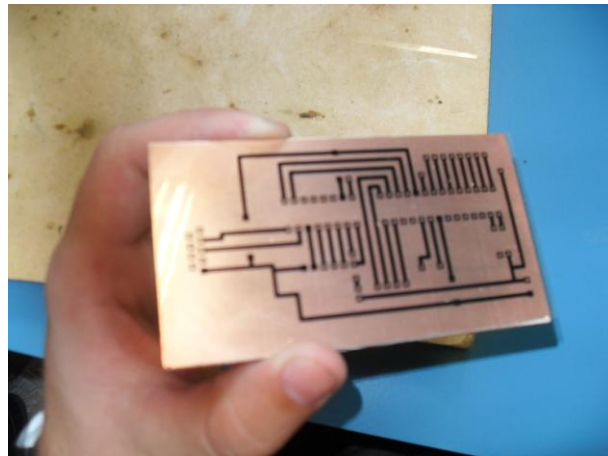


Figura 3: placa M-4

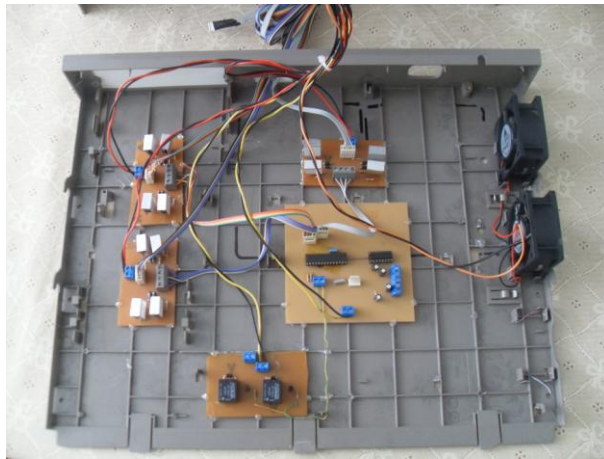


Figura 4: todas as placas



Figura 5: caixa de armazenamento dos circuitos



Figura 6: Stock Smart