



UNIVERSIDADE LUTERANA DO BRASIL
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA



VALDONIR TEIXEIRA CARVALHO

BOMBA DE INFUSÃO PERISTÁLTICA MICROPROCESSADA

Canoas, Dezembro de 2008.



VALDONIR TEIXEIRA CARVALHO

BOMBA DE INFUSÃO PERISTÁLTICA MICROPROCESSADA

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da ULBRA como um dos requisitos obrigatórios para a obtenção do grau de Engenheiro Eletricista.

Departamento:

Engenharia Elétrica

Área de Concentração

Engenharia Biomédica

Professor Orientador:

MSc. Eng. Eletr. Augusto A. D. de Mattos – CREA-RS: 88003

Canoas, Dezembro de 2008.



FOLHA DE APROVAÇÃO

Nome do Autor: Valdonir Teixeira Carvalho

Matrícula: 981103059-6

Título: Bomba de Infusão Peristáltica Microprocessada

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da ULBRA como um dos requisitos obrigatórios para a obtenção do grau de Engenheiro Eletricista

Professor Orientador:

MSc Eng. Eletr. Augusto A. D de Mattos

CREA-RS: 88003

Banca Avaliadora:

MSc.Eng. Eletr. Luis Fernando Espinosa Cocian

CREA-RS: 88.866-D

Conceito Atribuído (A-B-C-D):

Dr.Eng. Eletr. João Carlos Verneti dos Santos

CREA-RS: 45.852-D

Conceito Atribuído (A-B-C-D):

Assinaturas:

Autor
Valdonir Teixeira Carvalho

Orientador
Augusto A. D de Mattos

Avaliador
Luis Fernando Espinosa Cocian

Avaliador
João Carlos Verneti dos Santos

Relatório Aprovado em:



DEDICATÓRIA

Dedico aos meus pais, a minha esposa
Valéria e a meu filho Felipe.



AGRADECIMENTOS

A minha mãe e meu pai (in memória).

A minha esposa Valéria e meu filho Felipe pelo incentivo e paciência que tiveram comigo neste período.

A todos que colaboraram direta ou indiretamente na elaboração deste trabalho, o meu reconhecimento.

Ao Professor Augusto A. D. Mattos pelo estímulo, dedicação e esforço pessoal proporcionado.



EPIGRAFE

“O futuro chega rápido, não podemos perder tempo”.

O autor



RESUMO

CARVALHO, Valdonir. **Bomba de Infusão Peristáltica Microprocessada.** Trabalho de Conclusão de Curso em Engenharia Elétrica - Departamento de Engenharia Elétrica. Universidade Luterana do Brasil. Canoas, RS.

Este projeto mostra o desenvolvimento de um sistema de infusão peristáltica utilizando um motor de passo. O equipamento é composto por bomba peristáltica acionada pelo motor de passo e o “equipo” (conjunto reservatório, tubos e gotejador) por onde flui o líquido. Um sensor que conta gotas permite monitorar e controlar o volume infundido. A interface com o usuário é feita através de um teclado e *display*. O teclado é utilizado para a inserção das condições de ajuste, e as informações são apresentadas no *display*. Esse conjunto é controlado por um *hardware* microcontrolado baseado no microcontrolador AT89C52. A velocidade do motor de passo é alterada conforme a programação o monitoramento é feito através de um sensor.

Palavras chave: Bomba de infusão. Sistemas para infusão. Microcontrolador. Infusão. Motor de passo.



ABSTRACT

CARVALHO, Valdonir. Micro processed Peristaltic Infusion Pump. Final paper in the Electrical Engineering course – Electric Engineering Department. Universidade Luterana do Brasil – ULBRA (Brazil Lutheran University). Canoas, Rio Grande do Sul, 2008.

This Project a micro processed peristaltic system was developed with the use of a step motor. The equipment is compound by a peristaltic pump which is enacted by a step engine and an *equipo* (a set of containers, pipes and drippers) through which the liquid flows. A dripping sensor allows controlling and tracking of the inflowing liquid. The user interface is available by a keyboard and display. This set is controlled by a micro processed *hardware* based on the AT89C52 microcontroller. The speed of the step motor is altered according to programming and controlling chosen by the user through the sensor.

Keywords: Infusion pump, infusion systems, microcontroller, infusion, step motor.



LISTA DE ILUSTRAÇÕES

Figura 1: Concentração da droga x tempo	18
Figura 2: Componentes do sistema de infusão com controle de fluxo manual.....	19
Figura 3: Sistema de infusão por gravidade	21
Figura 4: Sistema de infusão por gravidade com contador automático.....	22
Figura 5: Sistema de infusão peristáltico linear	23
Figura 6: Esquema do mecanismo de infusão por seringa	24
Figura 7: Modelos comerciais de bomba de seringa	25
Figura 8: Movimento do pistão para baixo, reservatório enche.....	26
Figura 9: Movimento do pistão para cima, reservatório é esvaziado.....	26
Figura 10: Representação do mecanismo peristáltico.....	27
Figura 11: Representação do mecanismo peristáltico rotativo	28
Figura 12: Mecanismo peristáltico rotativo com batente	29
Figura 13: Mecanismo peristáltico rotativo sem batente	30
Figura 14: Motor de passo unipolar.....	32
Figura 15: Diagrama de blocos da bomba de infusão.....	34
Figura 16: Hardware do Display ligado ao KIT ULBEE51.....	37
Figura 17: Teclado 16 teclas	38
Figura 18: Câmara de gotejamento transmissor e receptor óptico.....	39
Figura 19: Circuito transmissor e receptor óptico.....	39
Figura 20: Driver de potência para acionamento do motor de passo.....	41
Figura 21: Esquema proposto para realização dos ensaios.....	43
Figura 22: Adaptador, câmara de gotejamento e respiro	44
Figura 23: Tubo plástico, tubo silicone e pinça rolete.....	45
Figura 24: Filtro e ponta do equipo.....	45
Figura 25: Cateter agulhado com agulhas 19G.....	46
Figura 26: Gráfico de passo (ms) em função da taxa de infusão (ml/h).....	47
Figura 27: Representação da função e sua linha de reta de ajuste linear de 12 ml/h à 19,2 ml/h	48
Figura 28: Representação da função e sua linha de ajuste exponencial do intervalo de 12 ml/h à 19,2 ml/h	49
Figura 29: Representação da curva de estabilização do equipo nos primeiros 60 minutos.....	52



Figura 30: Curva de percentual de erro do ensaio 1,2 e 3 do equipo 2.....	53
Figura 31: Curvas de percentual de erro do ensaio 1, 2 e 3 com sensor de gotejamento.....	54
Figura 32: Fluxograma do software.....	58
Figura 33: Mensagem de solicitação de taxa	59
Figura 34: Mostra no display taxa digitada	59
Figura 35: Configuração da taxa de infusão.....	60
Figura 36: Solicitação de volume total a infundir.....	60
Figura 37: Confirmação do valor do volume total.....	60
Figura 38: Informação do tempo da infusão total	61
Figura 39: Atualização do volume infundido.....	61
Figura 40: Alteração de parâmetros de infusão.....	61
Figura 41: Fim de infusão módulo KVO	62
Figura 42: Curva de passo do intervalo de 19,2 ml/h à 31,8 ml/h	64
Figura 43: Medida do passo do intervalo de 19,2 ml/h à 31,8 ml/h (Nível Alto).....	65
Figura 44: Medida do passo do intervalo de 19,2 ml/h à 31,8 ml/h (Nível Baixo).....	66



LISTA DE QUADROS

Quadro 1 – Atuação de um motor de passo unipolar (<i>full-step</i>)	33
Quadro 2 – Atuação de um motor de passo unipolar (<i>half-step</i>)	33
Quadro 3 – Pinagem do display	36



LISTA DE TABELAS

Tabela 1 – Valores obtidos com o primeiro ensaio de passo X volume/tempo	47
Tabela 2 – Valores de erros obtidos com as funções geradas para o intervalo de curva	49



LISTA DE ABREVIATURAS E SIGLAS

LCD: Liquid Crystal Display
VI: Intravenosa
KVO: Keep Vein Open
UTI: Unidade de Tratamento Intensivo
ECRI: Emergency Care Research Institute



SUMÁRIO

1 INTRODUÇÃO	15
2 REFERENCIAL TEÓRICO.....	17
2.1 DISPOSITIVOS OU SISTEMAS DE INFUSÃO	17
2.1.1 Equipo com Controle de Fluxo Manual	19
2.1.2 Controlador de Infusão	20
2.1.3 Bomba de infusão	22
2.1.3.1 Controle de fluxo	22
2.1.3.2 Mecanismos de infusão	23
2.2 SISTEMAS PERISTÁLTICOS DE INFUSÃO	27
2.3 MOTOR DE PASSO	30
2.3.1 Motor de passo unipolar.....	32
3 DESCRIÇÃO DO SISTEMA.....	34
3.1 DIAGRAMA DE BLOCOS	34
3.2 KIT ULBEE51	34
3.3 DISPLAY	35
3.4 TECLADO	37
3.5 SENSOR DE GOTEJAMENTO.....	38
3.6 ALARMES.....	40
3.7 DRIVER DE POTÊNCIA PARA ACIONAMENTO MOTOR DE PASSO.....	40
4 RESULTADOS DE FUNCIONAMENTO	42
4.1 RESULTADOS DOS ENSAIOS DE VALIDAÇÃO E TESTE.....	42
4.1.1 DETERMINAÇÃO DA TAXA DE INFUSÃO (TEMPO DE PASSO).....	46
4.1.2 RESULTADOS DOS ENSAIOS EM OPERAÇÃO NORMAL	51
4.1.3 DESCRIÇÃO DOS ENSAIOS COM SENSOR DE GOTEJAMENTO	53
4.2 DESCRIÇÃO DO SOFTWARE	54
4.2.1 FLUXOGRAMA DO SOFTWARE.....	57
4.2.2 OPERAÇÃO DO EQUIPAMENTO.....	59
4.2.3 ROTINAS DE DISPLAY	62
4.2.4 ROTINA DE PASSO	63



4.2.5 ROTINAS USADAS COM O SENSOR DE GOTEJAMENTO	67
5.1 CONCLUSÕES	69
REFERÊNCIAS.....	71
OBRAS CONSULTADAS	72
ANEXOS	74
ANEXO A – HARDWARE KIT ULBEE51	75
ANEXO B – HARDWARE TECLADO 16 TECLAS	77
APÊNDICE A – CURVAS E CÁLCULOS PARA PASSO MOTOR	79
APÊNDICE B – DADOS E CÁLCULOS ENSAIO PRÁTICO	100
APÊNDICE C – CÓDIGO FONTE DO SOFTWARE.....	103



1 INTRODUÇÃO

Atualmente, no tratamento de pacientes se faz necessário o uso de terapias intravenosas. Com o avanço da biomedicina, medicamentos mais potentes são desenvolvidos a fim de terem uma maior eficiência nos tratamentos. O sucesso da utilização desses medicamentos está diretamente relacionada à forma como a terapia intravenosa for administrada. O volume do medicamento administrado e a velocidade com que ele é injetado no paciente é um fator importante para o sucesso do tratamento. A infusão de soluções intravenosas contendo medicamentos é uma atividade rotineira em hospitais e ambulatórios [1].

Infusão significa introduzir um líquido qualquer que não seja o sangue em um vaso sanguíneo. Este método é muito utilizado, pois, as vias rápidas de circulação de fluido e medicamento até os órgãos do paciente, são as rotas intravenosas e intra-arterial. Para isso, utiliza-se o processo por gravidade, que consiste em frascos flexíveis suspensos por hastes onde a velocidade de infusão é controlada manualmente por pinças. Este método ocasiona inexatidão no volume infundido, ocorrendo o maior desvio quando as vazões são baixas (menores que 50 ml/h). Também devido ao volume infundido ser controlado visualmente e a pressão no equipo (maiores explicações no item 4.1) ser menor à medida que a coluna de solução diminui. Assim, as bombas de infusão surgiram para facilitar a infusão venosa e tornar consistente o uso de drogas potentes e de ação rápida [1].

O trabalho tem o objetivo de desenvolver o projeto de um equipamento de infusão capaz de controlar a vazão de um determinado medicamento através do tempo. Este equipamento é composto por um reservatório onde está contido o remédio a ser infundido no paciente, um sistema eletrônico microprocessado, responsável pelo monitoramento do sensor de gota e a emissão de alarmes, um teclado de 16 teclas, um display, assim como do acionamento do motor de passo; que é responsável por gerar a pressão na linha do equipo e conseqüentemente fazer o líquido circular.

No capítulo 2, é apresentado o referencial teórico sobre dispositivos de infusão, equipo e utilização, motor de passo e sistema peristáltico. No capítulo 3, são tratadas as diversas partes de hardware que compõem o projeto. Os elementos de



hardware incluem o teclado, display, sensor de gotejamento, alarme término de infusão e driver de potência para o acionamento do motor de passo. No capítulo 4, apresenta-se a metodologia, os resultados dos ensaios executados e a descrição de partes importantes do software de controle do sistema. No capítulo 5, são apresentadas as conclusões com relação ao objetivo proposto, assim como as sugestões de trabalhos futuros.



2 REFERENCIAL TEÓRICO

2.1 DISPOSITIVOS OU SISTEMAS DE INFUSÃO

Como a maioria dos pacientes hospitalizados recebe medicamentos por **infusão**, esta prática impulsionou o desenvolvimento de dispositivos para infundir drogas com pressão superior à pressão sangüínea, com maior controle e precisão. Estes dispositivos são utilizados para introduzir no sistema circulatório de pacientes líquidos e agentes farmacológicos por via intravenosa (IV), epidural (na medula nervosa da coluna vertebral) e, menos freqüentemente, intra-arteriais, em aplicações diversas como [6]:

- manutenção dos níveis apropriados de fluídos de um paciente durante e após cirurgias, tratamento de queimaduras e controle de desidratação em pacientes pediátricos;
- nutrição parenteral (endovenosa) total de pacientes;
- para manter a veia aberta, facilitando administração de medicamentos em uma emergência, *KVO (Keep Vein Open)* é um recurso de bombas de infusão programáveis, representado pela infusão mínima ao final da infusão, mantendo o vaso aberto;
- infusão de drogas, em quantidades efetivas e não tóxicas, de forma contínua (por exemplo, hormônio do crescimento) ou intermitente (por exemplo, antibióticos).

A **administração contínua de drogas** pode ser considerada a **aplicação mais importante dos dispositivos de infusão**, pois a concentração de uma droga em seu sítio de ação deve ser suficientemente alta para ser efetiva, mas não tão elevada a ponto de ser tóxica [6].

Os modos tradicionais de administração de drogas, através de injeções ou pílulas, resultam em flutuações ao longo do tempo na concentração da droga (figura 1), que podem fazer com que esta fique abaixo ou acima do valor ótimo, mesmo que a concentração média esteja dentro da faixa.

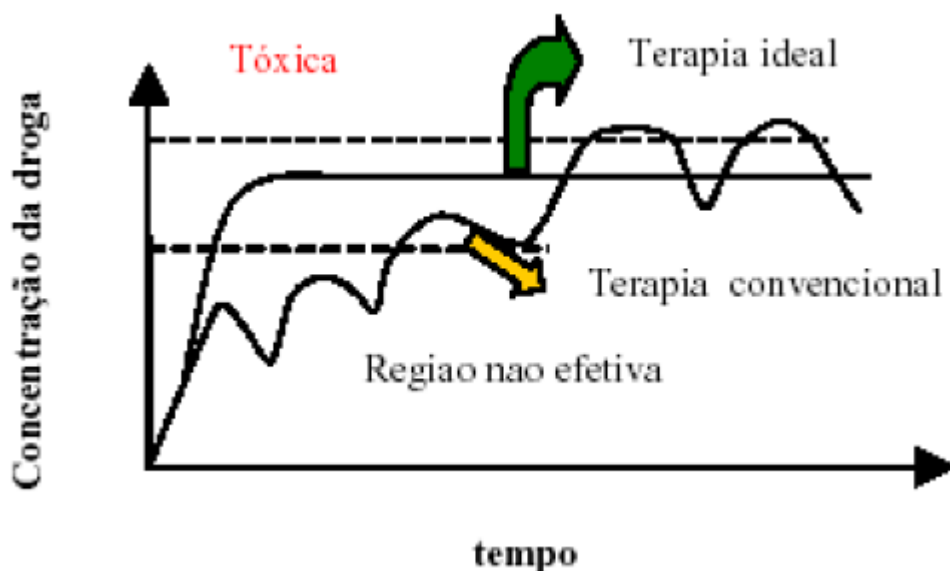


Figura 1: Concentração da droga x tempo. [1]

A infusão contínua de drogas reduz as flutuações e, se a taxa de infusão for correta, assegura uma ação terapêutica contínua.

A **infusão contínua de drogas** pode ser usada para aplicação de [6]:

- Drogas vasoativas para controlar a pressão arterial;
- Anestésicos durante cirurgias;
- Quimioterapia para tratamento de câncer;
- Agentes indutores de trabalho de parto;
- Drogas anti-arrítmicas;
- Insulina;
- Supressores de dor e de trabalho de parto;
- Hormônios; etc.

Dispositivos de infusão são utilizados em pacientes hospitalizados em ambulatórios, centros cirúrgicos, UTIs e em ambulâncias. Os pacientes que necessitam de terapia prolongada de infusão, mas que estão livres de outros cuidados hospitalares, podem ser tratados em casa [6].

Existem três sistemas de infusão: **o equipo com controle de fluxo manual**, o que utiliza um **controlador de infusão** (automático ou semi-automático) para estabelecer o fluxo determinado pelo operador; e **bomba de infusão**, que gera,

monitora e controla o fluxo [1].

2.1.1 Equipo com Controle de Fluxo Manual

É o sistema e infusão mais simples que consiste de um reservatório com um líquido a ser infundido, e um equipo, composto por um tubo, uma câmara de gotejamento e uma pinça rolete ou grampo que tem a função de comprimir o tubo do equipo e controlar o fluxo do líquido para o paciente (Figura 2). A pressão de infusão é a diferença entre a pressão hidrostática gerada pela coluna de líquido no equipo e a pressão venosa (que varia em torno de 10 mmHg) [1].

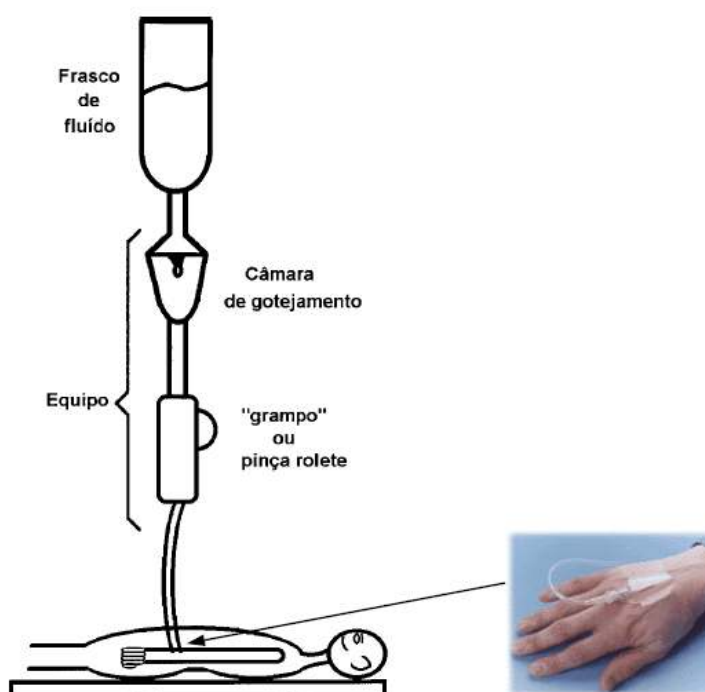


Figura 2: Componentes do sistema de infusão com controle de fluxo manual [1].

Este tipo de sistema apresenta como vantagens menor custo e simplicidade na operação e como desvantagens pode-se citar a imprecisão devido ao fluxo variar no tempo devido: ao posicionamento (altura) do reservatório, a viscosidade e a temperatura do líquido, as variações da pressão venosa do paciente, a redução da coluna de líquido ao longo da infusão, a eventual formação de coágulo na linha de infusão.



Este sistema só pode ser usado em veias periféricas, devido à baixa pressão gerada. Seu uso é impraticável em veias centrais e artérias, onde a pressão do sangue é alta.

Devido ao controle estar baseado na contagem das gotas por unidade de tempo não permite estabelecer um fluxo que resulte num número de gotas que não seja inteiro, por exemplo, 10,5 gotas/min, tendo uma grande taxa de erro, pois possui um erro que faz parte do processo que é a variação das gotas em função do tempo.

Não possui alarmes, e a monitoração do processo depende exclusivamente do profissional da saúde.

2.1.2 Controlador de Infusão

O controlador de infusão é destinado a regular a **vazão do líquido** administrado ao paciente pela pressão positiva gerada pela força da gravidade. A regulação do fluxo é controlada por uma contagem eletrônica de gotas.

Utilizando a força da gravidade temos dois sistemas de infusão:

Sistema de infusão por gravidade com controlador semi-automático de infusão, que com a ajuda do grampo, o operador ajusta o fluxo desejado. Na câmara de gotejamento existe um diodo emissor de luz e um sensor receptor (um foto-diodo). O feixe de luz é interrompido pela passagem de cada gota, enviando um sinal ao contador de gotas. Quando a contagem de gotas não esta de acordo com o valor pré-determinado, soa a alarme e o operador reajusta o grampo. (Figura 3).

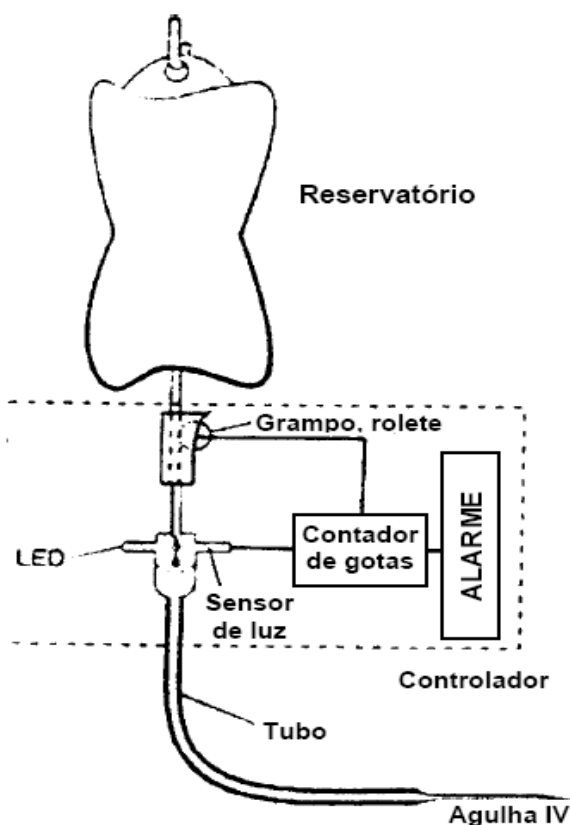


Figura 3: Sistema de infusão por gravidade [1].

Sistema de infusão por gravidade com controlador automático, estabelece e controla o fluxo de infusão e a pressão de infusão é determinada pela pressão gravitacional da coluna de líquido no reservatório e no equipo.

O contador de gotas ao invés de acionar o alarme, atua sobre o grampo, ajustando o fluxo para o valor definido pelo operador (Figura 4).

Este tipo de sistema apresenta como vantagens menor custo que as bombas de infusão e maior controle que o gotejador simples. Já como desvantagens pode-se citar a baixa pressão de infusão, que é definida pela altura da coluna de líquido, e o risco de entupimento do equipo ou na agulha.



Figura 4: Sistema de infusão por gravidade com contador automático. [1].

2.1.3 Bomba de infusão

As bombas de infusão são necessárias quando se necessita *precisão* e *segurança* no controle dos fluidos ao paciente. A infusão é por um longo período ou quando necessitamos que o fluído seja maior que os obtidos pelos equipamentos ajustados manualmente.

Em bombas de infusão a pressão de infusão é independente e maior que a pressão gravitacional. O controle do fluxo pode ser *volumétrico* ou *não volumétrico*, o mecanismo de infusão pode ser *peristáltico*, por *pistão* ou por *seringa*.

2.1.3.1 Controle de fluxo

Controle volumétrico: Controla a vazão e velocidade do líquido, independente das suas características. Tem dimensão de unidade de *volume* por unidade *tempo* (ml/h).

Controle não-volumétrico: Controla a quantidade de gotas e velocidade do

líquido, sendo o volume infundido dependente do tamanho da gota, do tipo de equipo usado, da temperatura e das características do líquido como viscosidade e densidade. Tem dimensão de unidade de *número de gotas* por unidade de *tempo* (gotas/min).

2.1.3.2 Mecanismos de infusão

Mecanismos de infusão são as partes responsáveis pelo fluxo, atuando em trechos do equipo, sendo os mais comuns os *peristálticos* (rotativos ou lineares), por *seringa* e por *pistão* ou *diafragma*.

Peristáltico rotativo: A bomba que utiliza este mecanismo, caracteriza-se pelo esmagamento do tubo (equipo) por onde passa o líquido a ser infundido. Devido a este trabalho estar direcionado para este tipo de sistema iremos abordar mais sobre este tipo de mecanismo no capítulo 2.2.

Peristáltico linear: Mecanismo composto por inúmeras placas que pressionam o tubo contra um batente, realizando um movimento ondulatório gerado por um fuso, acionado pelo motor, justaposto às placas. O fluido é impulsionado pela pressão aplicada ao tubo em posições consecutivas conforme figura 5[1].

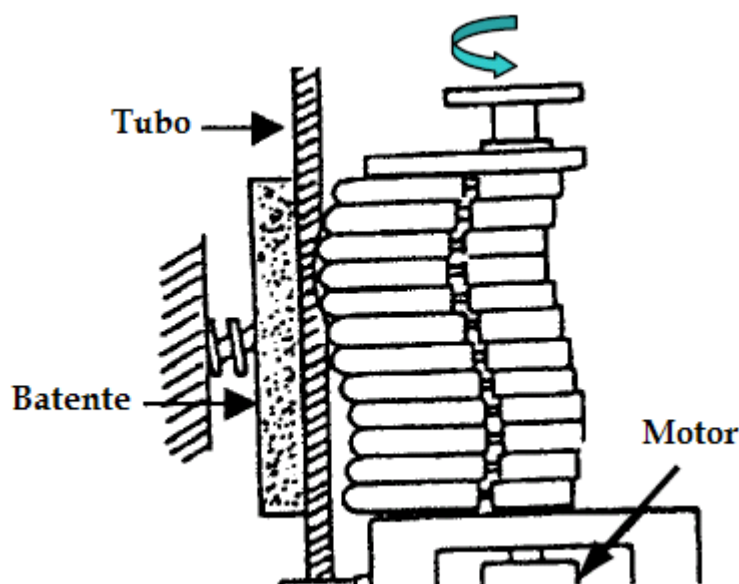


Figura 5: Sistema de infusão peristáltico linear [1].

Vantagens: Infusão contínua podem utilizar praticamente todos os tipos de equipo e são fáceis de instalar [1].

Desvantagens: Custo de aquisição e de manutenção mais alto. Geralmente são mais ruidosas que as rotativas [1].

Seringa: O mecanismo da bomba de seringa utiliza uma rosca sem fim para movimentar o êmbolo e utilizam uma seringa como reservatório do líquido, que é acionada por um motor de passo conforme a estrutura mecânica mostrada na figura 6.

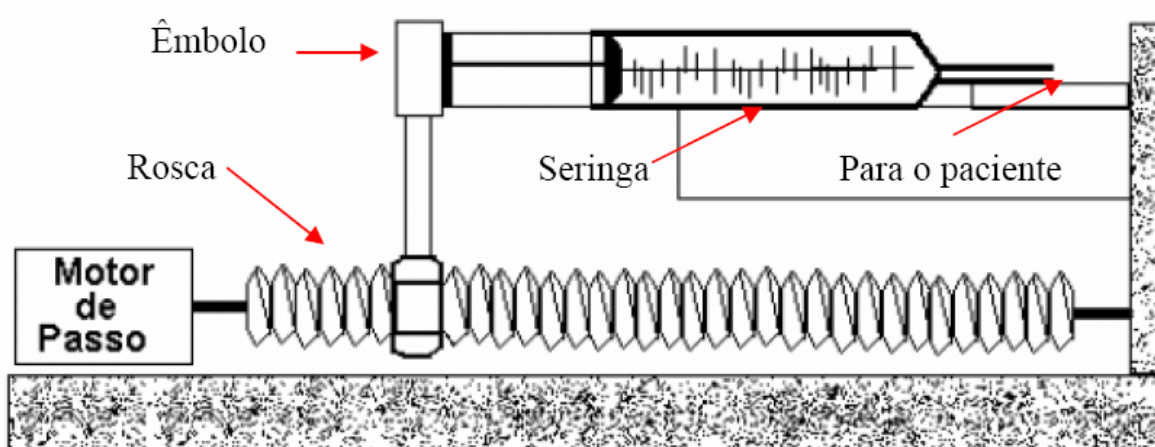


Figura 6: Esquema do mecanismo de infusão por seringa [6].

As bombas de seringa são mais precisas, oferecendo o fluxo contínuo para pequenos volumes sendo utilizadas para a infusão de medicamentos com concentração elevada por um longo período de tempo, aplicações pediátricas e terapia intensiva [6].

Vantagens: Devido a grande precisão mesmo sendo para baixos volumes, são normalmente utilizadas na infusão de medicamentos muito tóxicos ou que exijam exatidão na dose. Também são simples de usar e apresentam uma grande confiabilidade.

Desvantagens: A exatidão depende do modelo de seringa e servem para infusão de pequenos volumes.

Na figura 7 segue alguns modelos comerciais de bomba de seringa,



Figura 7: Modelos comerciais de bomba de seringa [6].

Pistão: O mecanismo utiliza um *cassete* contendo um dispositivo semelhante a um pistão, e com saídas dos dois lados. Um motor de passo aciona o pistão fazendo com que o cassete se movimente para dentro e para fora do cilindro.

O movimento para dentro bombeia o fluido do cassete para o paciente conforme representado na figura 8. Enquanto o movimento para fora retira fluido do reservatório enchendo novamente o cilindro conforme a figura 9 [1].

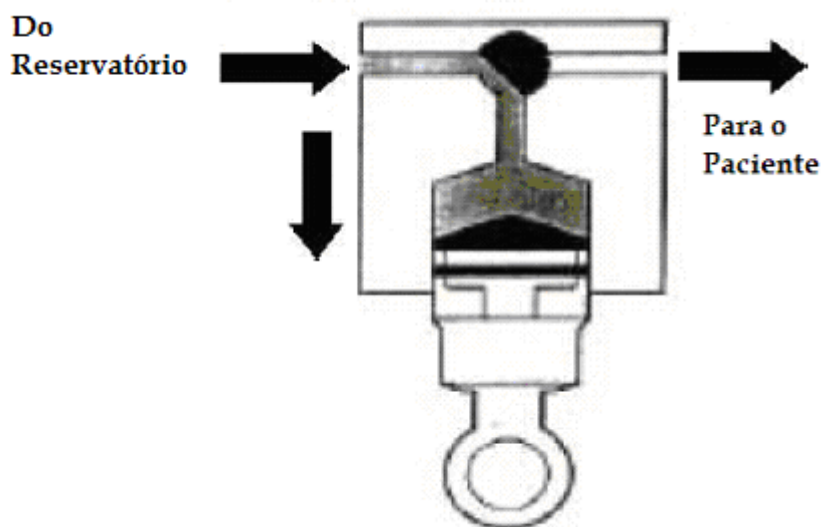


Figura 8: Movimento do pistão para baixo, reservatório enche [6].

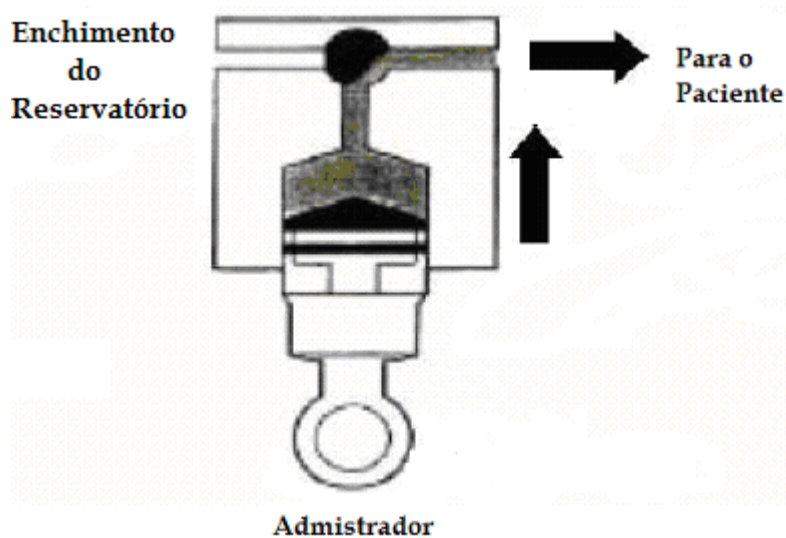


Figura 9: Movimento do pistão para cima, reservatório é esvaziado [6].

Vantagens: Grande precisão, utilizado quando um grande fluxo é necessário ser aplicado em curto espaço de tempo.

Desvantagens: Alto custo do cassete e do equipo, não pode ser utilizado a infusão lenta ou que necessitem de uma vazão constante (linear).

2.2 SISTEMAS PERISTÁLTICOS DE INFUSÃO

O sistema peristáltico é uma das mais antigas formas de dosagem, inventada pela própria natureza. Um exemplo é o esôfago ao intestino, nosso aparelho digestivo funciona pelo mesmo princípio, testado e aprovado por milhares de anos, com funcionamento intermitente e vida útil de muitas décadas.

A dosagem de produtos químicos em aplicações industriais, institucionais ou em laboratórios de pesquisa há muito tempo adotaram a peristáltica por ser um método robusto, de fácil construção e manutenção, e por não colocar partes metálicas em contato com o produto dosado, e nem apresentar dificuldades com bolhas de ar na linha de dosagem.

Para melhor compreender um funcionamento de uma bomba de infusão peristáltica, primeiramente far-se-á uma breve descrição dos princípios básicos desses sistemas.

Os sistemas peristálticos de infusão impulsionam o líquido a ser infundido através de um tubo (equip) por esmagamento em pontos sucessivos ao longo de uma determinada região, conforme indicado na figura 10.

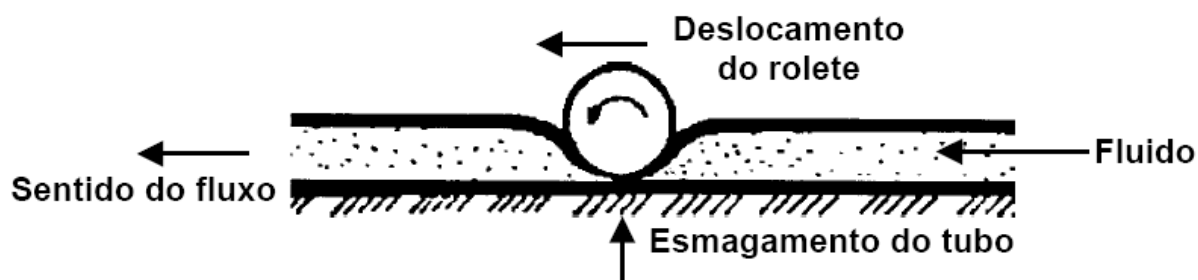


Figura 10: Representação do mecanismo peristáltico [1].

Como o esmagamento do tubo é feito em pontos sucessivos, o líquido no seu interior tende a ser empurrado no mesmo sentido em que esta se movendo. Assim, um sistema peristáltico deste tipo impele o líquido sem que as diferenças de pressão nas extremidades do tubo afete a infusão, desde que estas diferenças de pressão não sejam suficientes para deformar o próprio tubo.

Levando este princípio para o caso específico de infusões parenterais ou

venosas utilizadas na medicina moderna através das bombas de infusão, nota-se que o ato de infundir qualquer medicamento em um paciente torna-se bastante prático, pois as diferenças de pressão entre o sangue do paciente e o frasco do medicamento não afetará na infusão do mesmo. Utilizando bombas de infusão, pode-se controlar com muito mais precisão o fluxo do medicamento que está sendo administrado ao paciente, pois qualquer equipamento moderno deste tipo dispõe de circuitos de controle de fluxo, que possibilitam ajustar a velocidade de pressionamento dos diversos pontos sucessivos ao longo do tubo.

As bombas peristálticas podem usar um mecanismo **rotativo** (com e sem **batente**) ou **linear**.

Mecanismo Peristáltico Rotativo: é composto por um rotor que pressiona pequenos rolos (roletes) contra o tubo flexível (Figura 11).

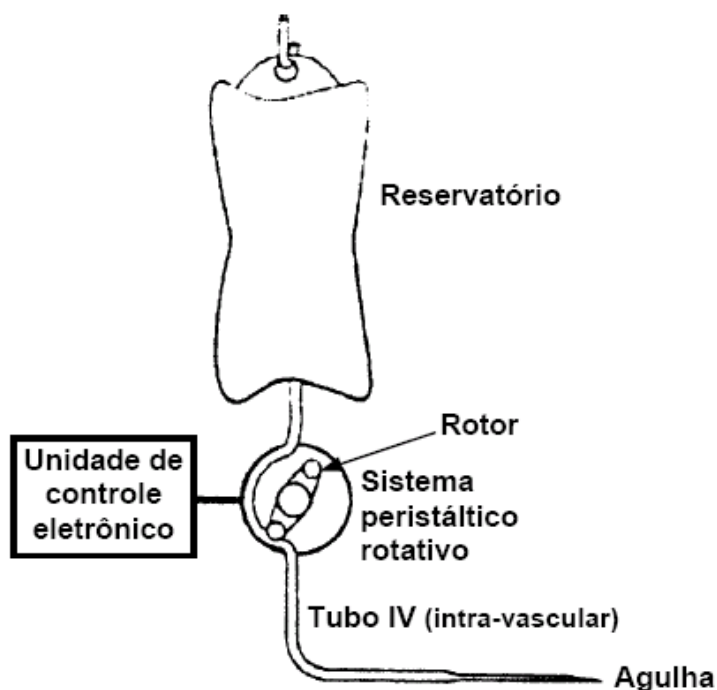


Figura 11: Representação do mecanismo peristáltico rotativo [1].

No mecanismo de infusão peristáltico rotativo **com batente**, o tubo é pressionado contra um ponto fixo (batente rígido).

Vantagem: pode ser usado **equipo comum** (com trecho de silicone), mais barato.

Desvantagem: se o batente não estiver bem ajustado para o equipo, o rotor pode **travar** (equipo muito espesso), ou pode não haver esmagamento, resultando em fluxo livre de líquido (Figura 12).

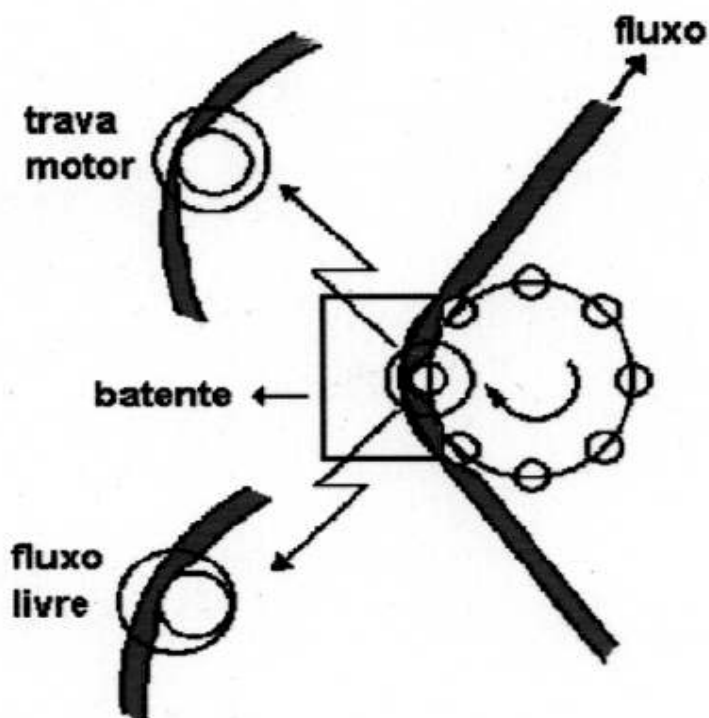


Figura 12: Mecanismo peristáltico rotativo com batente [1].

Na figura 12 são indicados dois casos extremos: fluxo livre (sistema rotativo não esmaga o tubo); e pressão excessiva (travamento do rotor).

No mecanismo de infusão peristáltico rotativo **sem batente**, o tubo não possui ponto fixo e esmagamento. Ele contém um trecho de material muito flexível (geralmente **silicone de grau médico**) que é preso sobre o rotor (Figura 13). Os roletes esticam e esmagam este trecho do tubo promovendo o fluxo.

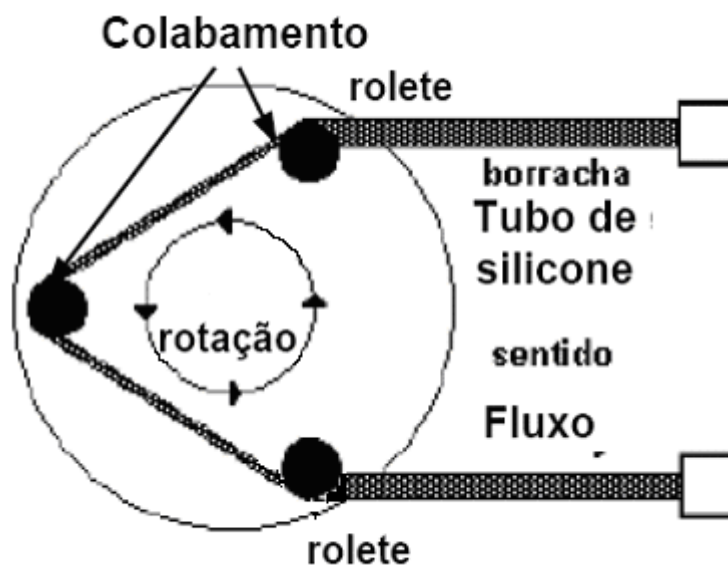


Figura 13: Mecanismo peristáltico rotativo sem batente [1].

A escolha do tubo de silicone está ligada a diversos fatores, tais como [1]:

- grau de pureza (médico, alimentício ou industrial);
- diâmetro interno do tubo: define a quantidade de fluido deslocado a cada passo do motor;
- espessura da parede do tubo: determina a força para se obter o colapso do tubo.

Vantagem: é mais fácil de ser construído: a instalação do equipamento não é tão crítica.

Desvantagem: exige equipamento especial, próprio para cada bomba (mais caro).

2.3 MOTOR DE PASSO

Os fabricantes dos equipamentos eletromédicos bombas de infusão utilizam-se do motor de passo como força de direcionamento do fluido. Os motores de passo podem ser diretamente controlados por um sistema digital, tornando-o preciso e de recomendável utilização em aplicações que venham a requerer um ajuste fino de posicionamento [2].

A velocidade de um motor elétrico convencional alimentado por corrente alternada (CA) ou por corrente contínua (CC), pode variar de acordo com sua



tensão, com o suprimento de carga mecânica ou com a frequência. Seu funcionamento é dificultado na ausência de um mecanismo de realimentação para se avaliar a exatidão da rotação do motor [2].

Normalmente, os motores de passo são projetados com enrolamento de estator polifásico, o que não foge muito dos demais motores, sendo o número de pólos determinado pelo passo angular desejado por pulsos de entrada.

O motor de passo consiste num motor DC de magnetos permanentes ou de relutância variável que apresenta as seguintes características de desempenho:

1. Rotação em ambas as direções;
2. Variações incrementais de precisão angular,
3. Repetição de movimentos bastante exatos;
4. Um torque de sustentação à velocidade zero, e
5. Possibilidade de controle digital

Um motor de passo pode mover-se em incrementos angulares bastante exatos, conhecidos como *passos*, em resposta a pulsos digitais aplicados a um *driver* a partir de um controlador digital, no nosso caso o microcontrolador AT89C52. O número de pulsos e a cadência com que estes pulsos são aplicados controlam a posição e a velocidade do motor, respectivamente. Geralmente os motores de passo podem ser fabricados com 12, 24, 72, 144 e 200 passos por revolução, que resultam em incremento de 30, 15, 5, 2.5, 2 e 1.8 graus respectivamente.

Os motores de passo podem ser **bipolares**, que requerem duas fontes de alimentação ou uma fonte de alimentação de polaridade comutável que não será descrito neste trabalho por não fazer parte do projeto, ou **unipolares**, que requerem apenas uma fonte de alimentação. Em ambos os casos as fontes utilizadas são de tensão contínua e requerem um circuito digital que produza as seqüências para produzir a rotação do motor.

Este projeto utilizou um motor de passo unipolar com as seguintes características elétricas:

Motor unipolar 04 fases, 12 Vdc, 1.8 graus /passo, 160 mA.

2.3.1 Motor de passo unipolar

Motores de passo unipolar são caracterizados por possuírem um “*center-tape*” entre o enrolamento de suas bobinas. Utiliza-se este center-tape para alimentar o motor, que é controlado aterrando-se as extremidades dos enrolamentos [2].

O número de fases é duas vezes o número de bobinas, uma vez que cada bobina se encontra dividida em duas. Na figura 14 tem-se a representação de um motor de passo unipolar de 4 fases.

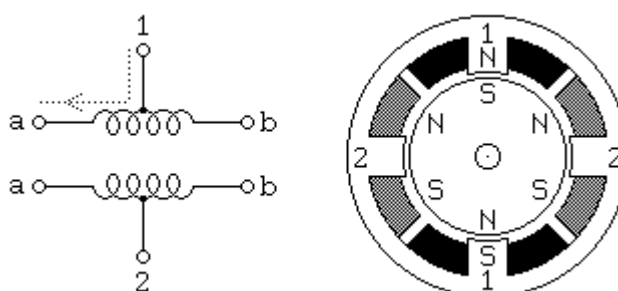


Figura 14: Motor de passo unipolar [2].

Normalmente, a derivação central dos enrolamentos está ligada ao terminal positivo da fonte de alimentação e os extremos da cada enrolamento são ligados alternadamente á terra para assim inverter a direção do campo gerado por cada um dos enrolamentos.

Na figura 14 ainda pode-se ver o corte transversal de um motor com um passo de 30 graus. O enrolamento 1 encontra-se distribuído entre o pólo superior e pólo inferior do estator do motor, enquanto que o enrolamento 2 encontra-se distribuído entre o pólo esquerdo e o pólo direito do estator. O rotor é um magneto permanente com seis pólos (3 pólos sul e 3 pólos norte), dispostos ao longo da circunferência do rotor [2].

Se a alimentação for removida do enrolamento 1 e for alimentado o enrolamento 2, o rotor irá deslocar-se 30°, ou seja um passo. Para obter uma rotação contínua do motor, deverão ser alimentados alternadamente os enrolamentos do motor. Assumindo uma lógica positiva, em que o valor lógico 1



significa fazer passar a corrente num dos enrolamentos, a seguinte seqüência, apresentada na Quadro 1, produzirá um deslocamento de oito passos ($8 \times 30 = 240^\circ$).

Quadro 1 - Atuação de um motor de passo unipolar (*full step*)- [2].

	1a	2a	1b	2b
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1

Este tipo de atuação é denominado atuação de passo completo *full step*. Outro tipo de atuação possível consiste em alimentar, alternadamente, um e dois enrolamentos, permitindo deste modo avançar meio passo de cada vez. Este tipo de atuação é denominado de meio-passo, ou "*half step*". Neste tipo de atuação, como facilmente se pode verificar, é duplicar o número de "passos" para completar uma volta. Na realidade passamos a deslocar o rotor em apenas meios passos, ou seja, seguindo ainda o caso exemplificado, o rotor desloca-se 15° por cada atuação [2].

No quadro 2, é apresentada uma atuação do tipo "*half-step*" para os mesmos 240° de deslocamento do rotor.

Quadro 2 - Atuação de um motor de passo unipolar (*half-step*) [2].

	1a	2a	1b	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0
11	0	1	0	0
12	0	1	1	0
13	0	0	1	0
14	0	0	1	1
15	0	0	0	1
16	1	0	0	1

3 DESCRIÇÃO DO SISTEMA

3.1 DIAGRAMA DE BLOCOS

Na figura 15, segue um diagrama de blocos simplificado do projeto, o qual é descrito sucintamente.

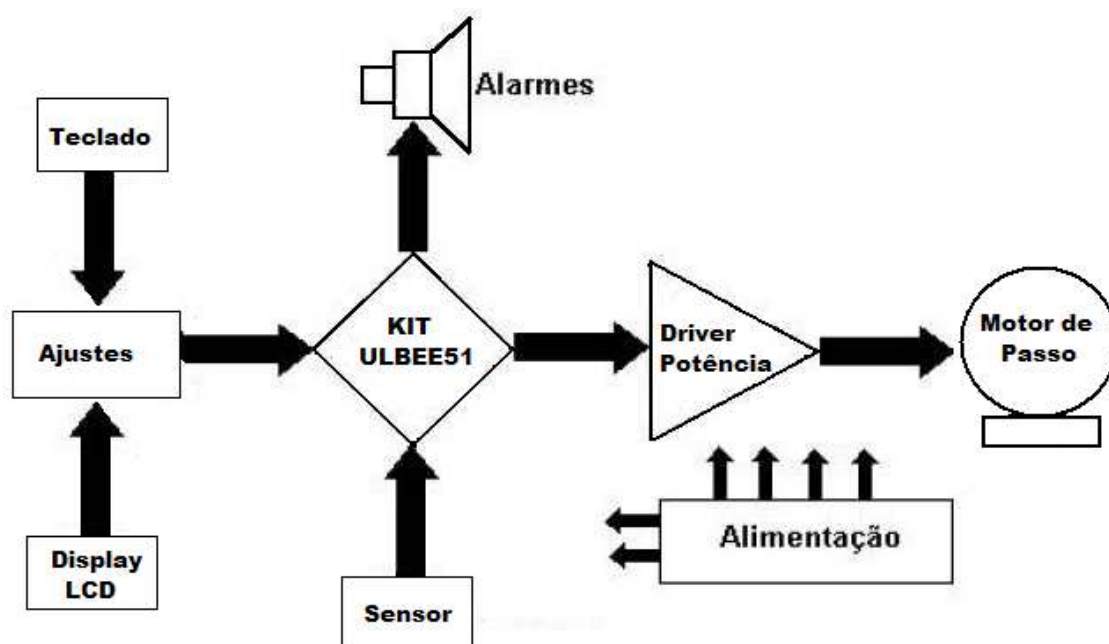


Figura 15: Diagrama de blocos da bomba de infusão.

A bomba de infusão projetada é constituída pelas seguintes partes: Teclado, Display LCD, Unidade de controle (KIT ULBEE51), alarme, sensor (contador de gotas), driver de potência para acionamento do motor de passo e motor de passo.

3.2 KIT ULBEE51

A placa Kit ULBEE51 tem a função de controlar todos os periféricos a ela acoplados como display, teclado, alarme, sensor e motor de passo.

O Kit ULBEE51 é composto pelo microcontrolador AT89C52, da família 8051, com clock de 11,0592 MHz e memória RAM externa de 64KBytes. A comunicação do computador com o kit ocorre através do conector DB9 interligado a um MAX232,



que permite a comunicação via porta serial do PC. O hardware completo do KIT ULBEE51 está no anexo A.

Na EPROM do próprio microcontrolador foi gravado o Paulmon2, que é um programa monitor que possui rotinas de comunicação serial que permitem a comunicação e o envio de programas para o microcontrolador. Além disso, o Paulmon2 possui rotinas próprias, que podem ser acessadas para executar determinadas tarefas, o que muito facilita na elaboração de um programa

O 8051 é um microcontrolador de 8 bits, com média performance, sendo aplicado em controle de sistemas. Abaixo é indicada a configuração típica para um microcontrolador da família 8051, no qual está sendo utilizado o AT89C52 :

- 64 KBytes de memória para DATA (RAM) e CODE (ROM).
- 256x8 bytes de memória RAM, dividida entre área de uso geral e registradores especiais.
- 2 *timer/counters* de 16 bits.
- 1 porta serial programável (UART).
- Interface para memória externa com capacidade de 64 KBytes de endereçamento externo para ROM e 64 KBytes de endereçamento externo para RAM.
- 4 portas de I/O.
- 6 possibilidades de interrupções com dois grupos de prioridades.

Por ser um representante da família MCS51, o microprocessador utilizado foi o AT89C52, que apresenta maior quantidade de memória RAM e mais um *timer/counter* de 16 bits que o modelo 8051 padrão. Um dos pontos que o torna compatível com o 8051 é a região em que estão alojados os registradores especiais, trabalhando assim com palavras de 8 bits.

3.3 DISPLAY

O display utilizado é o de 2 linhas nas quais pode-se escrever 16 caracteres por linha. O barramento de dados deste componente está conectado a porta P0 (P0.0 à P0.7), o pino de controle RS à porta P3.3, e o pino ENABRE ao *strobe do KIT ULBEE51*.



O quadro 3 ilustra a pinagem do display conforme o datasheet MC1602-04, onde observa-se que o display possui os pinos de dados (DB0 e DB7) e também três pinos de controle (RS, R/W, E).

Quadro 3 - Pinagem do display [3].

PIN ASSIGNMENT		
Pin no.	Symbol	Function
1	BLK	Back light cathode
2	BLA	Back light anode
3	Vss	Power Supply (GND)
4	VDD	Power supply (+)
5	VO	Contrast Adjust
6	RS	Command data input
7	R/W	Data read / write
8	E	Enable signal
9~16	DB0 ~ DB7	Data bus line

Segue uma breve descrição dos pinos de controle do display.

⇒ **E:** (Enable Signal). Este pino é usado para informar ao LCD que o usuário está enviando dados. Para enviar dados para o para o LCD, deve-se setar as outras duas linhas de controle e/ou colocar informação no barramento de dados. Quando RS e R/W estão prontas, pode-se setar E com nível lógico '0'.

⇒ **RS:** (Register Select). Quando RS esta em nível lógico '0' os dados serão tratados como um comando ou instrução especial (como limpar o display, posicionar o cursor, etc.). Quando RS possui o nível lógico '1', os dados sendo enviados representam mensagem textual que deve ser mostrada no display. Por exemplo, para mostrar o caractere "A" no display, RS deve estar em nível lógico '1'.

⇒ **RW:** (Read/Write). Quando coloca-se RW em nível lógico '0', a informação presente no barramento de dados deve ser escrita no LCD. Quando RW está em nível lógico '1', o programa está realizando uma leitura no LCD. Apenas uma instrução é um comando de leitura: "*Get LCD Status*". Todo o restante das instruções são comandos de escrita, o que se pode supor que RW irá, na maior parte do tempo, possuir o nível lógico '0'.

Na figura 16 é ilustrada a conexão do display com o KIT ULBEE51.

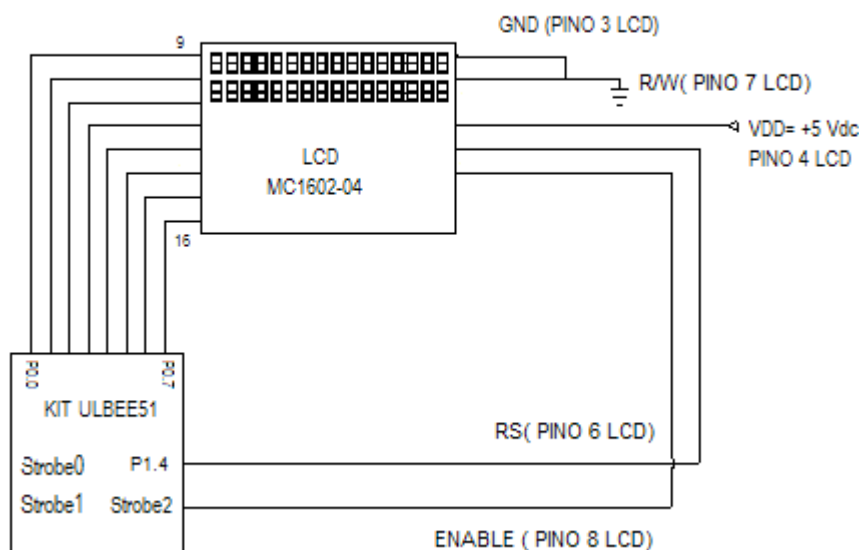


Figura 16: Hardware do Display ligado ao KIT ULBEE51.

3.4 TECLADO

O teclado implementado para este projeto é do tipo matricial, 4 linhas e 4 colunas com 16 teclas de toque numeradas de 0 até F. Este teclado utiliza um barramento de 8 bits para manipular as teclas, sendo quatro bits para receber dados e quatro bits do teclado para *controlar/varrer* qual linha de teclas será acionada, fazendo desta maneira um sistema de varredura. Este periférico se comunica com o KIT ULBEE51 através dos cabos padrão utilizando os pinos da PORT P3, conforme descritos abaixo, alimentado o próprio teclado:

Inicialmente, é feita uma varredura para se verificar se alguma tecla correspondente a uma das quatro colunas foi pressionada podendo ser coluna 1, 2, 3 ou 4 correspondente ao “*nibble*” menos significativo. Em seguida, é feita outra varredura para identificar a que linha pertence à tecla pressionada, podendo ser a linha 1, 2, 3 ou 4 e para isso carrega-se o “*nibble*” menos significativo como zero correspondente à coluna e o *nibble* mais significativo as linhas com 5V (nível lógico 1). Ou seja, a operação de leitura do teclado acontece em duas etapas, sendo a primeira que verifica a qual coluna pertence a tecla pressionada fazendo uma

varredura e em seguida a qual linha pertence a requisitada [6].

Abaixo, segue a figura 17 do teclado utilizado no projeto, e o hardware que esta no anexo B.

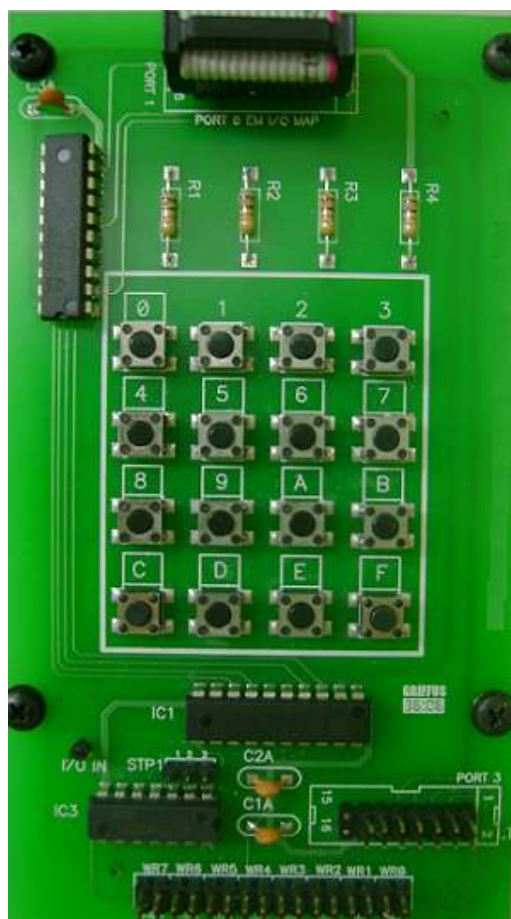


Figura 17: Teclado 16 teclas [4].

Este teclado é utilizado para a entrada de dados (programação) da infusão a ser realizada. Esses dados normalmente incluem vazão e o volume máximo a ser infundido.

3.5 SENSOR DE GOTEJAMENTO

O sensor de gotejamento faz o monitoramento da medida do fluxo entregue ao paciente através de um sensor fixado à câmara de gotejamento monitorando, cada passagem de gota. Para isso, utiliza-se um sensor fotoelétrico baseado na

transmissão e recepção de luz infravermelha, podendo ser refletida ou interrompida pelo objeto a ser detectado, ou seja, a gota do líquido que passa pela câmara de gotejamento do equipo (Figura 18).

Este tipo de sensor é composto por dois circuitos básicos conforme figura 19: Um transmissor (diodo emissor de luz infravermelha) responsável pela emissão do feixe de luz, e o fotodiodo responsável pela recepção do feixe de luz. Quando há uma luz emissora no fotodiodo receptor, o mesmo entra em condução. Com a fonte emissora de luz apagada o fotodiodo entra em corte fornecendo uma saída de 5V no anodo.

Os sensores são capazes de detectar vários tipos de objetos e através deles é possível a aplicação de implementarmos como contador de gotas, que a cada bloqueio da luz irá gerar um nível alto na saída e conseqüentemente a contagem.

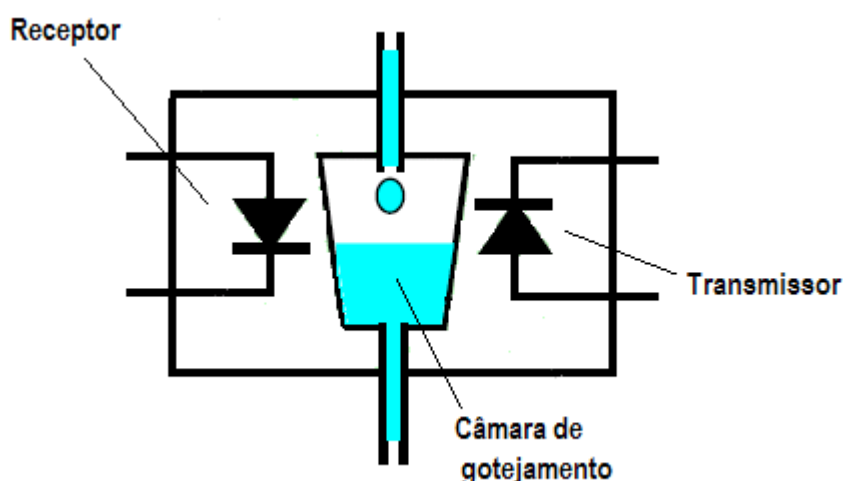


Figura 18: Câmara de gotejamento transmissor e receptor óptico

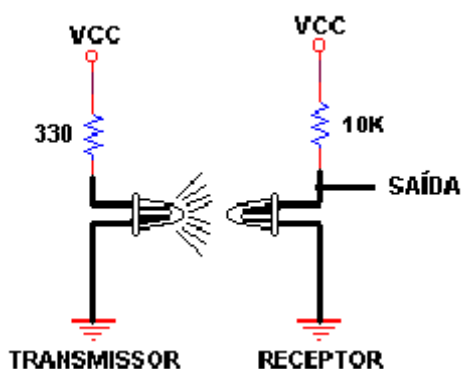


Figura 19: Circuito Transmissor e receptor óptico [5].



3.6 ALARMES

As bombas de infusão devem possuir alarmes para garantir a operação segura, dentro dos limites de tolerância definidos pelo projeto e normas, mesmo em situações imprevistas.

Por recomendação (ECRI – 1995 f) os alarmes que devem constar em um equipamento são os seguintes:

Ar na linha: Em algumas bombas de infusão este alarme é igual ao alarme de recipiente vazio, a fim de evitar bolhas de ar no equipo.

Fim de infusão: Este sensor é responsável em avisar quando do término do líquido da infusão. No projeto este sensor também é utilizado para monitorar o volume infundido, sendo posicionado junto a câmara de gotejamento do equipo onde o hardware esta montado, conforme a figura 18. Composto por um feixe de luz infravermelha (emissor) que corta a corrente do receptor, gerando um pulso. Também será usado um alarme sonoro para avisar do término da infusão.

Oclusão: Sensor responsável por monitorar a pressão do tubo no equipo.

Bateria fraca: Sensor para monitorar a bateria fraca em caso de falta de energia, mantendo a autonomia mínima na ausência de energia.

3.7 DRIVER DE POTÊNCIA PARA ACIONAMENTO MOTOR DE PASSO

Devido à tensão do motor de passo utilizado ser de 12VDC, e o valor de tensão de saída no microcontrolador ser de 5VDC, é necessário utilizar um driver transistorizado que permite um aumento na capacidade de corrente de saída. A figura 20 mostra o módulo do motor de passo para 6 fios. Os transistores usados são o TIP 122. Os resistores de 10k Ω que ligam o 8051 com o transistor servem para limitar a corrente de polarização dos transistores. Também é necessário colocar diodos (1N4007) em paralelo com as bobinas do motor, evitando a queima do transistor.

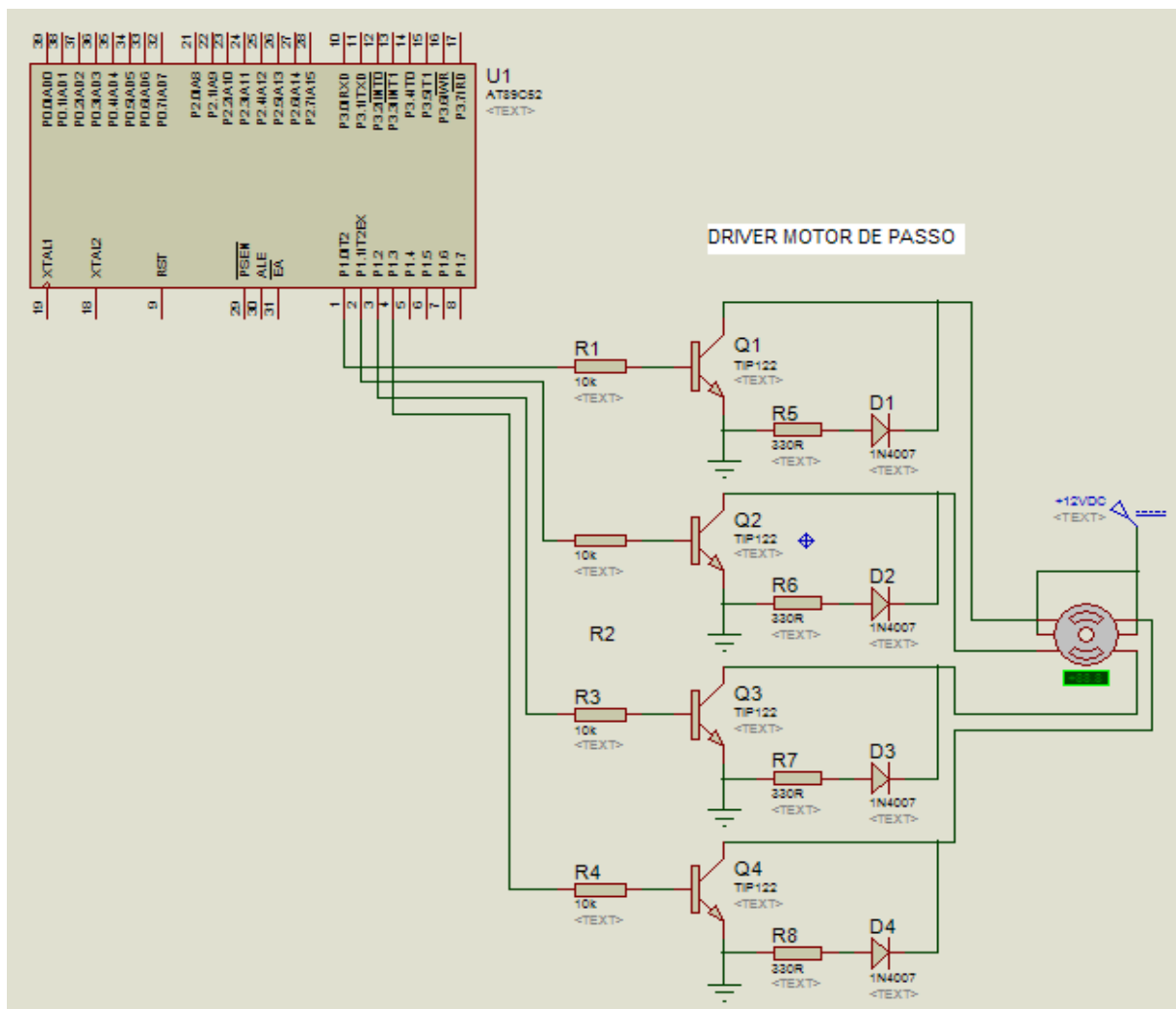


Figura 20: Driver de potência para acionamento do motor de passo.



4 RESULTADOS DE FUNCIONAMENTO

4.1 RESULTADOS DOS ENSAIOS DE VALIDAÇÃO E TESTE

Inicialmente, os métodos utilizados empregam alguns termos relacionados à administração da infusão, que também fazem parte da especificação do equipamento. Os termos são os seguintes:

Taxa de Infusão: taxa de infusão é a quantidade de tempo sobre o qual um volume específico de fluídos é infundido. Pode-se expressar esta taxa de infusão em ml/h, ml/min ou ml/seg.

Volume Infundido: é a quantidade de fluído que foi infundido. Utiliza-se este parâmetro para monitorar a quantidade de fluído administrado em um determinado período.

Volume a ser Infundido: é a quantidade de fluído que está para ser infundido.

Um passo importante é a definição dos parâmetros da bomba de infusão para verificar se as especificações definidas estão dentro do recomendado. Existe a norma NBR IEC 60601-2-24 [7], que regulamenta estes parâmetros, que exige equipamentos especiais e apropriados à sua aplicação. Em virtude da importância dos dados de saída das bombas de infusão, é importante propor um procedimento de ensaio para a verificação desses dados.

O ECRI (Emergency Care Research Institute) (998a) [7] recomenda que as bombas de infusão não devem sofrer uma variação maior que 5% nos diferentes volumes infundidos para taxas de infusão distintas.

Os ensaios quantitativos relacionados aos dados de saída da bomba de infusão são necessários e têm o principal objetivo de verificar a conformidade do valor solicitado com o valor de saída. Para a realização do procedimento de ensaio para medir os dados de saída da bomba de infusão volumétrica peristáltica, foi utilizado parte da norma NBR IEC 60601-1-24 [7] que recomenda os seguintes parâmetros para o ensaio:

Jogo de equipo, agulha 18G, 1,2mm (esta sendo usada a 19G, 1,8mm), soro fisiológico como solução a ser infundida e uma balança eletrônica com precisão de

cinco casas decimais (aqui, é usada uma proveta com escala de aproximação $\pm 0,2\text{ml}$ para medidas até 10 ml e uma com escala de aproximação $\pm 1\text{ml}$ para medidas maiores que 10 ml).

A seguir o esquema proposto para a montagem dos itens para os ensaios realizados, conforme a figura 21, onde fazem parte: o recipiente com o líquido a ser infundido, à câmara de gotejamento e sensor de gotejamento, o equipo, a unidade de controle microprocessada e, a proveta de medição.

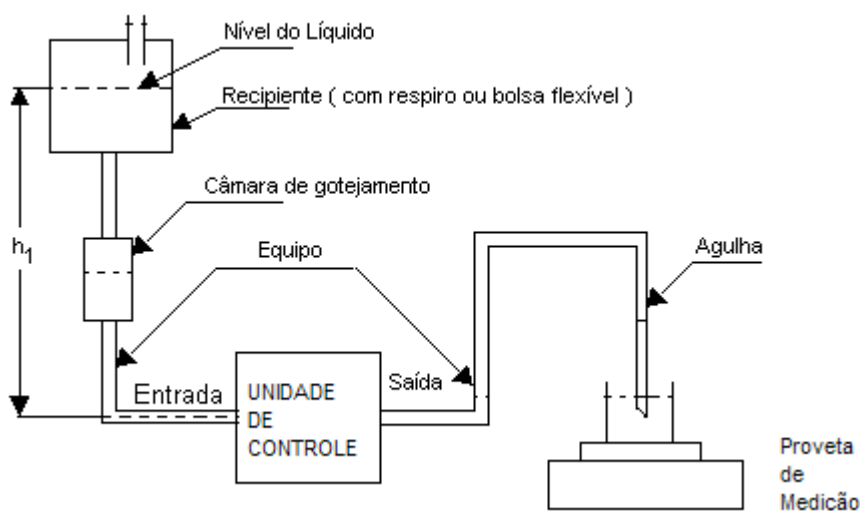


Figura 21: Esquema proposto para realização dos ensaios

No mercado, pode-se encontrar vários modelos de bombas e cada uma possui o seu equipo específico, devido a cada fabricante colocar suas próprias características em função do seu equipamento.

Para este projeto selecionou-se o equipo do Fabricante BBRAUN, que é utilizado na bomba de infusão volumétrica NUTRIMATII. Abaixo, são descritas as partes deste equipo de modo simplificado, pois não faz parte do objetivo deste projeto.

As figuras 22, 23 e 24 mostram as diferentes partes do equipo, detalhadas a seguir.



Figura 22: Adaptador, câmara de gotejamento e respiro.

Adaptador: é a extremidade conectada ao reservatório do líquido a ser infundido. É composto por um tubo plástico com uma ponta própria para ser inserida no frasco. É conectada ao respiro e a câmara de gotejamento.

Respiro: ou orifício de gotejamento. É uma abertura que determina o tamanho e a forma de gotejamento do fluido.

Câmara de gotejamento: É o compartimento transparente de plástico adaptável que contém o orifício de gotejamento, conectado ao tubo.

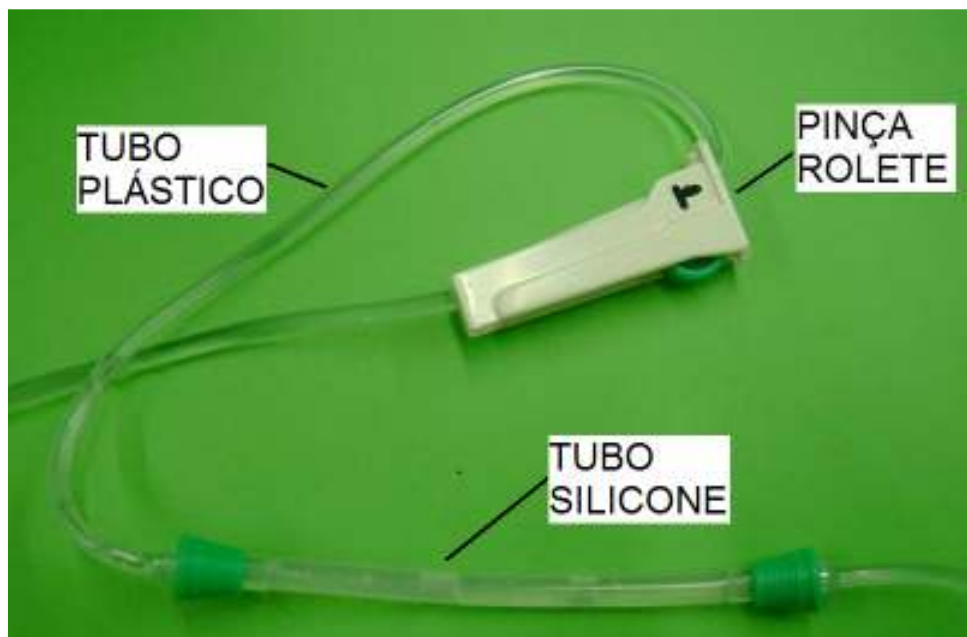


Figura 23: Tubo plástico, tubo silicone e pinça rolete.

Tubo plástico do equipo: O tubo do equipo plástico é conectado à câmara de gotejamento. O comprimento médio do equipo primário varia de 1,6 a 2,5m. O comprimento médio do tubo secundário situa-se entre 80 a 100 cm.

Tubo silicone: O tubo de silicone de grau médico é o que sofre a compressão dos roletes do motor, sendo o responsável pela infusão do líquido e o comprimento é definido de acordo com cada fabricante e seu equipamento.

Pinça: A pinça de controle de fluxo opera sobre os princípios da compressão da parede do tubo do equipo. Nos ensaios, o rolete da pinça não está comprimindo o tubo plástico.



Figura 24: Filtro e ponta do equipo.

Filtro: O filtro remove partículas estranhas que poderão estar no líquido a ser infundido.

Ponta do equipo: Onde é acoplado o cateter agulhado e faz a conexão do equipo ao paciente.

A figura 25 mostra o cateter com a agulha 19G, 1,8mm e suas principais partes, utilizado para todos os ensaios, conforme descrito no início do procedimento.

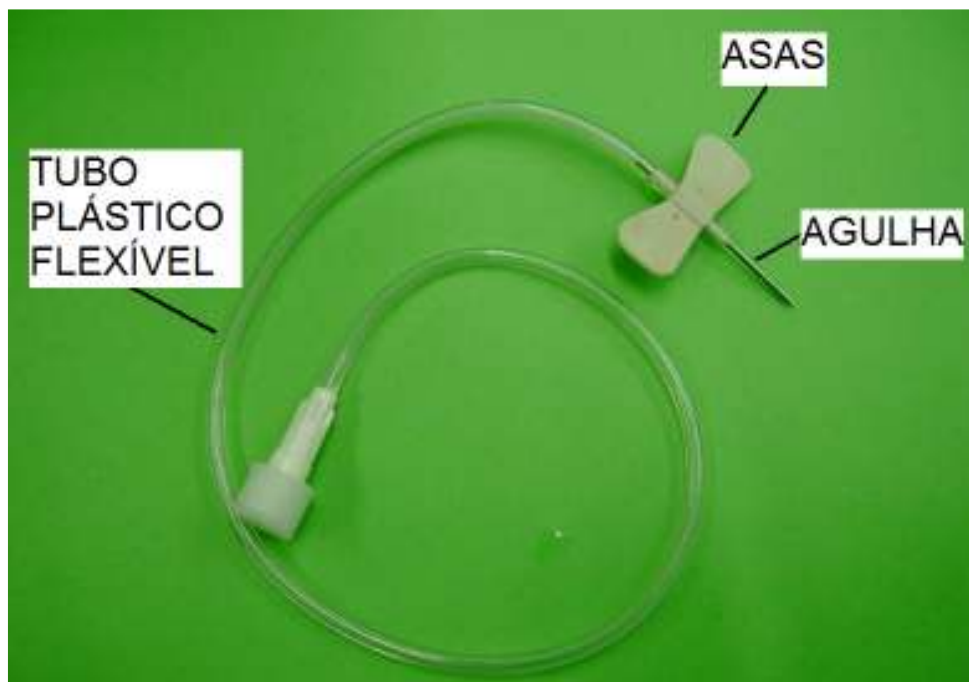


Figura 25: Cateter agulhado com agulha 19G

Cateter agulhado: É feito de aço inoxidável com comprimento de 1,25 a 3,0 cm. As asas presas à haste são feitas de borracha ou plástico flexível. Foram utilizados dois equipos de mesmo modelo para estes ensaios.

4.1.1 DETERMINAÇÃO DA TAXA DE INFUSÃO (TEMPO DE PASSO)

Primeiramente, foi realizado o ensaio para obter as curvas de **taxa de infusão em função do tempo**. Este procedimento é realizado via software, onde mediante alteração do tempo em ms, o motor de passo atinge determinada velocidade e conseqüentemente a variação de pressão no equipo, determinando assim o volume infundido na saída. Na tabela 1, seguem os valores medidos e a curva conseqüente do ensaio.



A tabela 1 apresenta os dados dos ensaios de forma parcial. Basicamente, aplica-se um tempo de valor conhecido (coluna 1, valor passo), e mede-se o tempo e o volume na saída. Os dados e cálculos completos referentes a este ensaio estão no apêndice A.

Tabela 1

Valores parciais obtidos com o primeiro ensaio de Passo x Volume/Tempo

Valor Passo	Tempo Medido (minutos)	Volume Medido (ml)	Volume Estimado (ml em 60 min)
900	00:10:00	4,40	26,40
700	00:10:00	5,30	31,80
500	00:10:00	7,20	43,20
400	00:05:00	4,70	56,40

A figura 26 apresenta a curva de valores de passo em função de volume/tempo para os valores medidos na Tabela 1. A taxa de infusão (ml/h) é a variável independente e o valor do passo, a variável dependente.

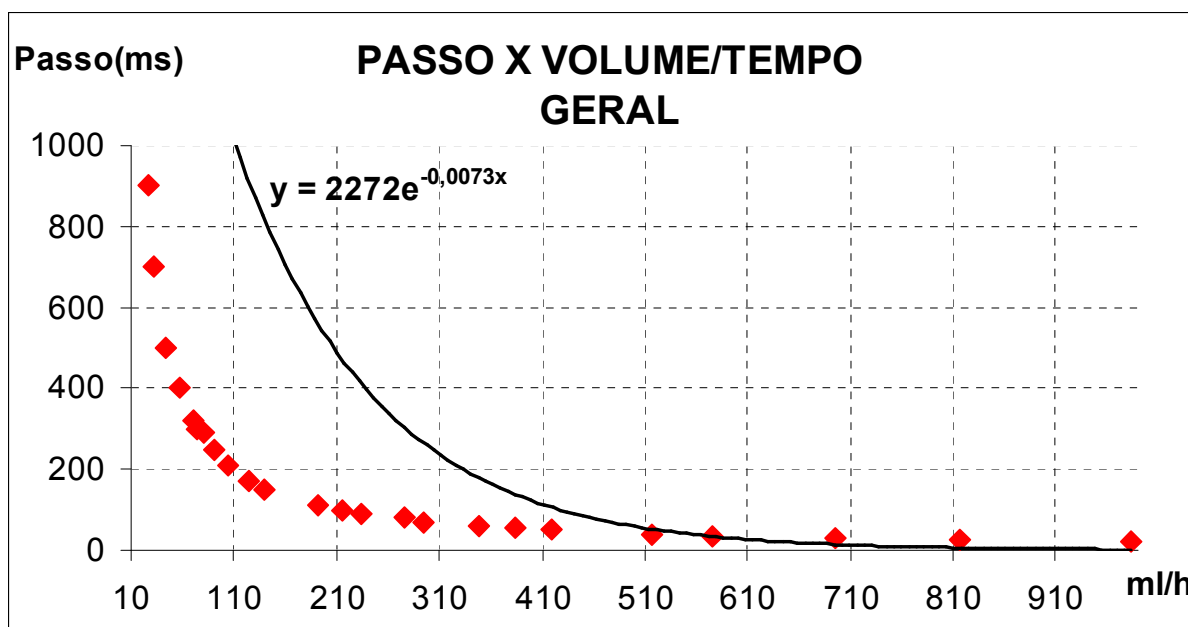


Figura 26: Gráfico de passo (ms) em função da taxa de infusão (ml/h).

Com os dados levantados, utilizou-se o software Excel da Microsoft para fitar a curva de passo do motor x volume/tempo, utilizando uma curva exponencial. Não



obteve-se um bom resultado, como mostra a curva. Então, a solução encontrada para reduzir esta taxa de erro foi dividir esta curva em várias partes menores, onde da mesma maneira traçou-se à curva de ajuste, e identificou-se a função que apresenta o menor erro. Dependendo do erro calculado, utiliza-se a que apresentar menor erro. Estes 18 fragmentos de retas e cálculos estão declarados no apêndice A.

Na figura 27 demonstra-se o fragmento da curva correspondente ao intervalo de taxa de infusão de 12ml/h à 19,2 ml/h e sua reta de ajuste.

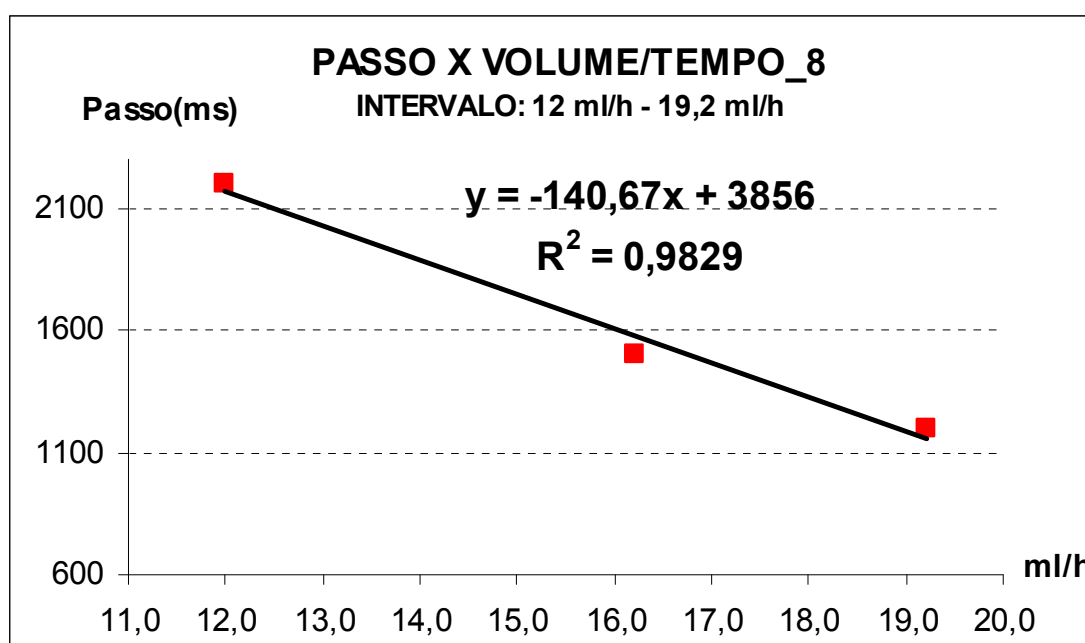


Figura 27: Representação da função e sua linha de reta de ajuste linear do intervalo de 12 ml/h à 19,2 ml/h.

A figura 28 mostra o mesmo fragmento da reta com o mesmo intervalo de taxa de infusão, porém como a sua linha de tendência exponencial e sua respectiva função.

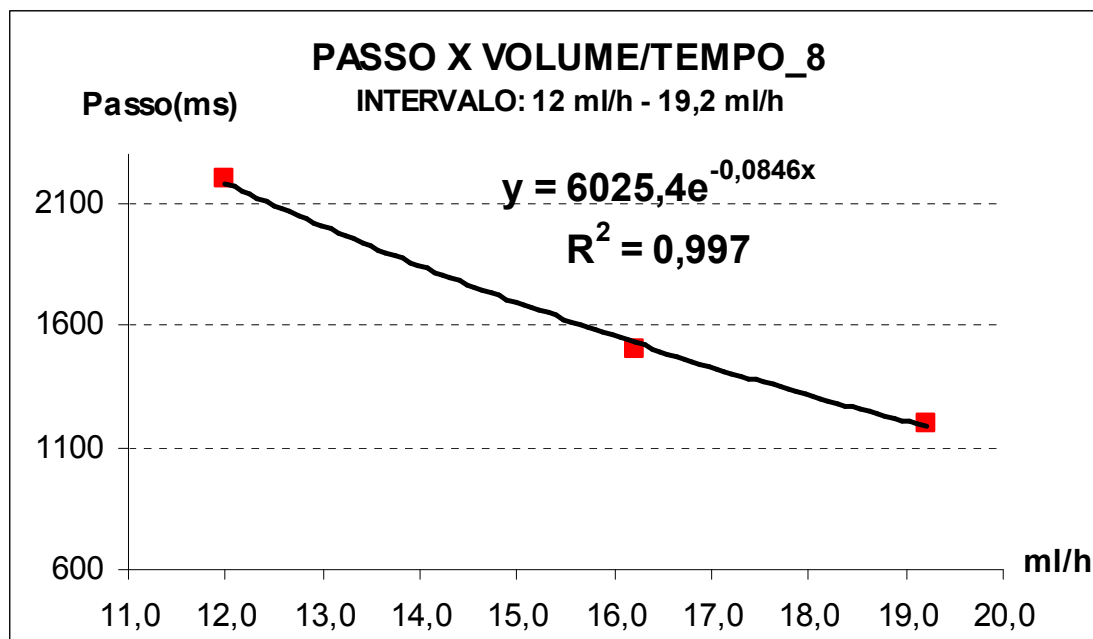


Figura 28: Representação da função e sua linha de ajuste exponencial do intervalo de 12 ml/h à 19,2 ml/h.

Com as duas funções organizou-se a tabela 2, apresentando o erro percentual do valor calculado em função do valor medido e selecionando a função com base no menor percentual de erro.

No exemplo fica claro que a função que mais se aproxima da curva real é a exponencial, pois gera o menor erro percentual do passo calculado para o valor medido.

Tabela 2

Valores de erro obtidos com as funções geradas para o intervalo de curva

PASSO x VOLUME/TEMPO_8					
Medido		Calculado Exponencial		Calculado Linear	
		$y=6025,4e(-0,0846x)$		$y=- 140,67*x + 3856$	
Valor_passo	ml/h	Valor_passo	%	ml/h	%
2200	12,00	2183,18	-0,765%	2167,96	-1,456%
1500	16,200	1530,3	2,020%	1577,146	5,143%
1200	19,200	1187,27	-1,061%	1155,136	-3,739%

Através do *coeficiente de correlação* também é possível fazer a análise para a definição da função a ser utilizada no software, visto que se tem duas variáveis (“x” e



“y”) e o estudo que envolve as equações exponencial e de reta. Onde “y” é variável dependente e “x” a independente e a relação linear entre as duas variáveis é medida pelo coeficiente de correlação (R), sendo que este valor varia de -1 a 1.

O valor 1 é a correlação perfeita positiva entre as duas variáveis e o -1 significa uma correlação negativa perfeita entre as duas variáveis, isto é, se uma aumenta, a outra sempre diminui.

Se este valor for zero significa que as duas variáveis não dependem linearmente uma da outra. No entanto, pode existir uma dependência não linear. Assim, o resultado R=0 deve ser investigado por outros meios.

Neste tipo de análise é importante que se determine o quanto a linha de regressão representa os dados. Deste modo, se faz necessário calcular o R^2 de Pearson ou coeficiente de determinação.

O coeficiente de correlação de Pearson calcula-se segundo a seguinte fórmula:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}},$$

onde: x_1, x_2, \dots, x_n e y_1, y_2, \dots, y_n são os valores medidos de ambas as variáveis e \bar{x} e \bar{y} são as médias aritméticas dessas variáveis.

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \quad \bar{y} = \frac{1}{n} \cdot \sum_{i=1}^n y_i$$

O cálculo de R é uma operação bastante simples para o software Excel, portanto não foi aprofundado o estudo com relação ao coeficiente de Pearson, apenas foi usado o resultado informado pelo software. Conforme é demonstrado nas figuras 27 e 28, onde um $R^2 = 0,997$, tem-se que 99,7% da variabilidade decorre de “x”. Inversamente, pode-se dizer que 0,3% da variância de “y” não é atribuível às diferenças em “x”. Conclui-se que o R com a função exponencial está muito mais próximo de 1 devido à forte correlação entre as variáveis.

No apêndice A encontram-se os gráficos com a informação do R calculados pelo software Excel.



4.1.2 RESULTADOS DOS ENSAIOS EM OPERAÇÃO NORMAL

Depois de traçar os 18 fragmentos da curva principal, calcula-se o erro de cada uma e define-se a função com maior precisão para utilizar no software. Esta função permite definir a velocidade do motor de passo. Os dados e cálculos para estes fragmentos do intervalo de 1ml/h á 999 ml/h referente e suas funções de ajuste estão no apêndice A. A maneira que este processo foi feito no software é detalhada no item 4.2.3.

Por exemplo, definindo que a taxa de infusão deverá ser de 10 ml/h, ao final do período de uma hora, na proveta de medição deverá ter um volume de 10 ml, com uma precisão de 5% ou seja , poderá variar de 9,5 a 10,5 ml após o mesmo período de 1 hora.

A primeira tentativa não apresentou um bom resultado, e depois de uma análise e pesquisa de equipamentos semelhantes e contato com pessoas da área de manutenção, identificou-se que todo o equipo deve ser usado durante algum tempo antes do equipamento ser testado.

Este funcionamento deve ser durante 1 hora com uma taxa constante de infusão, sendo este período necessário devido à necessidade de estabilização da elasticidade do silicone do equipo.

O ensaio de estabilização do equipo durante os primeiros 60 minutos está demonstrado na figura 29 e os dados levantados constam na tabela 3, que está no apêndice A.

O início do período do ensaio do tempo foi simultâneo ao início de funcionamento do equipamento, garantindo, assim, a infusão dentro do intervalo definido, onde este período foi dividido em intervalos de amostragem de 5 minutos, verificando o percentual de erro e o momento que este equipo adquire a estabilização.

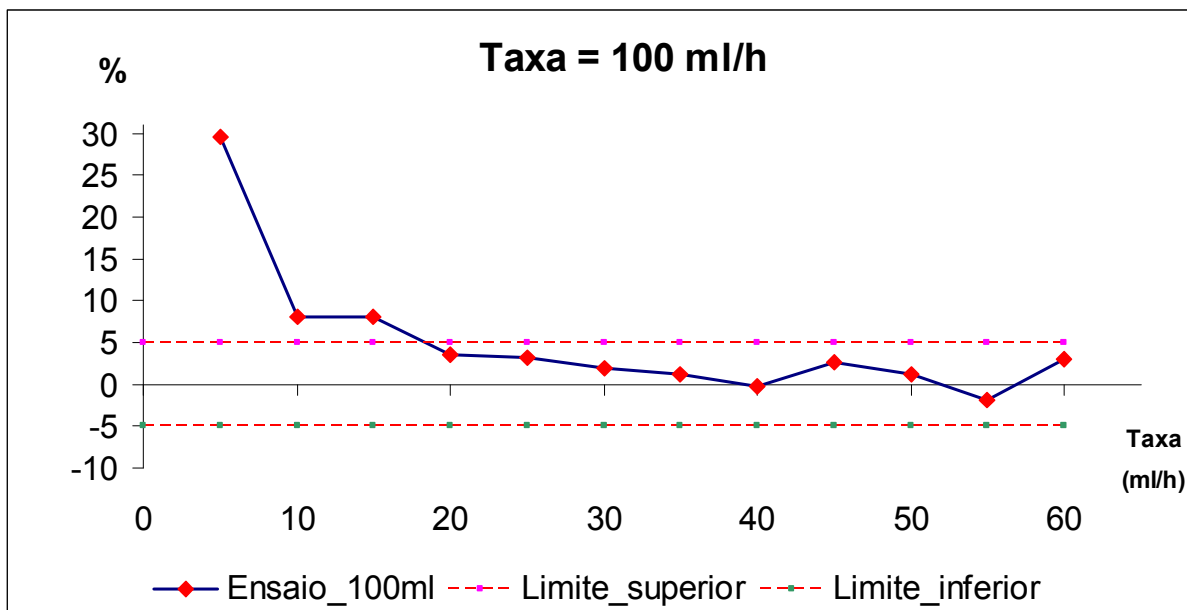


Figura 29: Representação da curva de estabilização do equipamento nos primeiros 60 minutos.

A seguir foram executados os testes com o equipamento em operação normal e com o software contemplando todas 18 funções de ajuste.

Foram executados ensaios em dois equipamentos, definidos como equipamento 1 e 2, onde foram setados valores de taxas diferentes e monitorado o volume de saída durante um determinado tempo, com o objetivo de verificar o percentual no volume infundido.

Aqui são mostrados apenas os gráficos resultantes dos ensaios, sendo as tabelas de dados e cálculos apresentados no apêndice B.

Ensaio 1 e 2: Os testes foram iniciados da menor taxa de infusão para a maior, no caso de 1 ml/h à 900 ml/h, em intervalos diferentes de tempo, monitorando o volume de saída visualmente através da proveta de medição.

Ensaio 3 : Procedimento idêntico ao ensaio 3 e 4, porém com diferentes taxas de infusão.

A figura 30 representa em percentual, o erro encontrado na taxa de infusão setado versus o volume na saída. Os dados e cálculos para a geração deste gráfico estão no apêndice B.



Verifica-se que o percentual de erro encontrado com relação à taxa de infusão definida e realizada não passou de 5%, valor este de precisão especificado pela maioria dos equipamentos com a mesma função.

Estes ensaios serviram para demonstrar que o principal objetivo do projeto foi atingido, em que se conseguiu um equipamento simples e de fácil programação e com uma precisão adequada a um tratamento ideal e com exigência de uma infusão constante e definida.

Aqui são apresentados os resultados de um único equipo, porém foram utilizados outros equipos durante a etapa de ajuste do software.

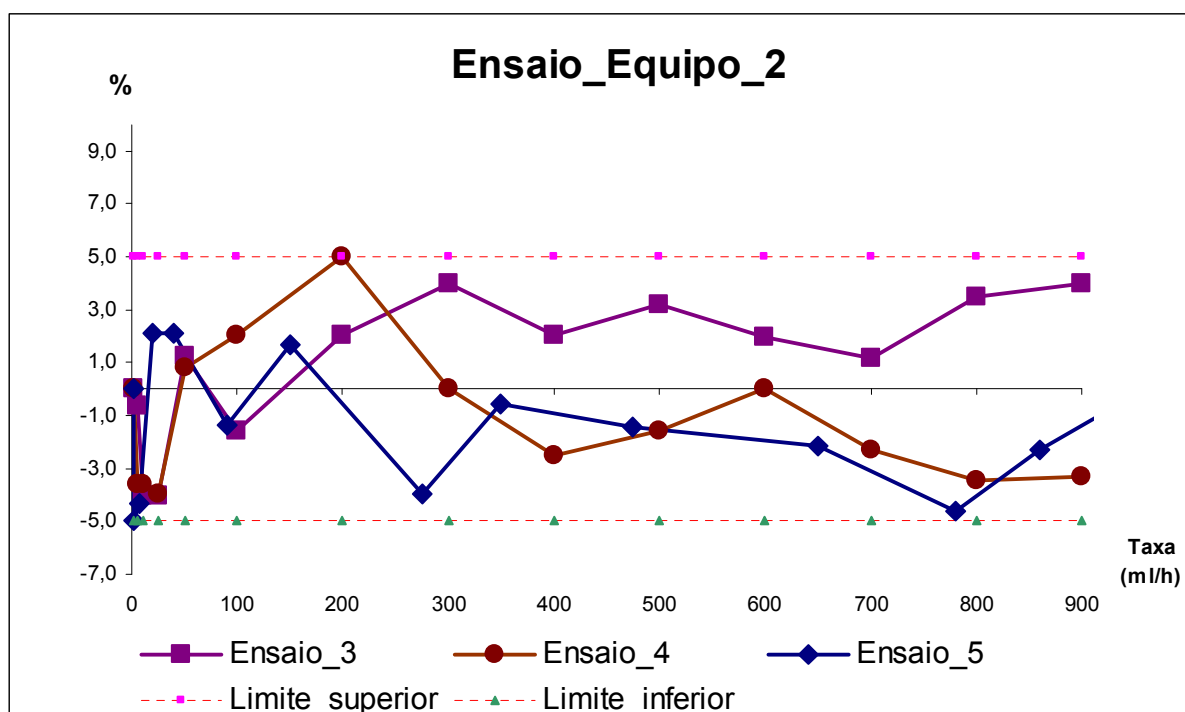


Figura 30: Curva de percentual de erro do ensaio 1,2 e 3 do equipo 2

4.1.3 DESCRIÇÃO DOS ENSAIOS COM SENSOR DE GOTEJAMENTO

Utilizando o sensor de gotejamento para determinação do volume infundido mostrou-se que devido ao percentual de erro encontrado com relação ao volume infundido, este processo não se mostra eficaz devido a variação do tamanho, forma e velocidade da gota no equipo.

A figura 31 mostra que ao utilizar o sensor de gotejamento como monitoramento do volume infundido obtém-se um percentual de erro significativo



comparando com o volume na saída. Dados e cálculos contidos no apêndice B.

Como se observa, a precisão está distante em vários pontos, principalmente nos intervalos com taxa alta de infusão.

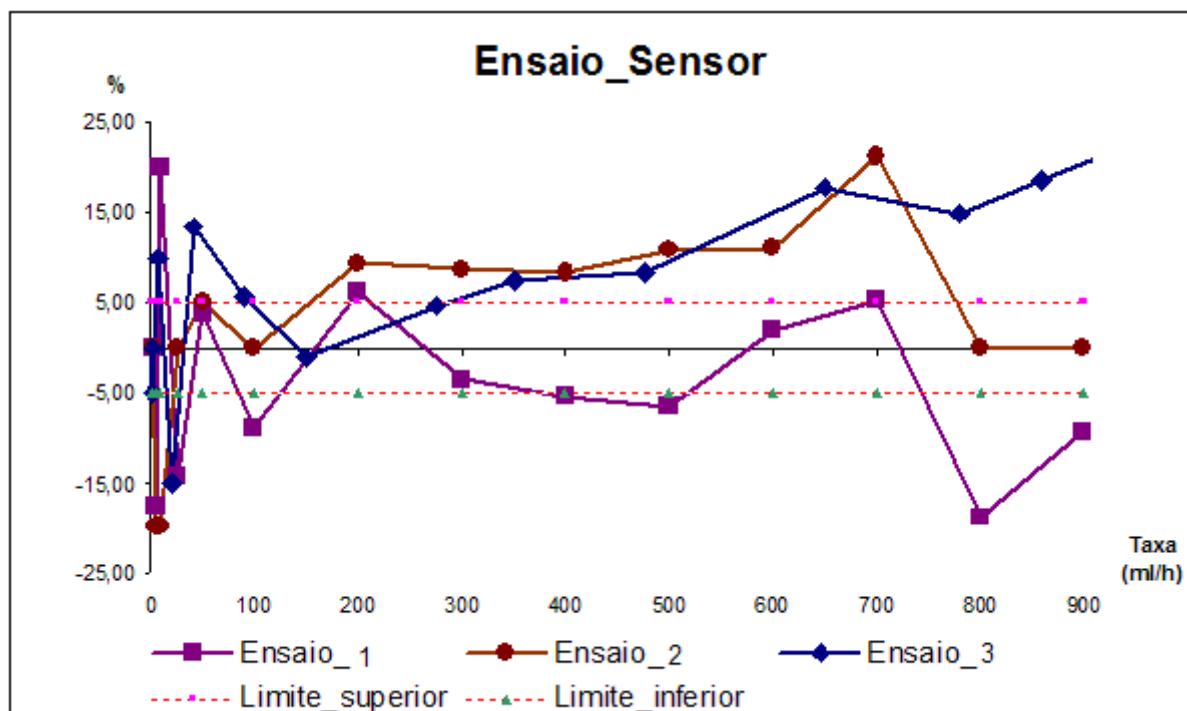


Figura 31: Curvas de percentual de erro do ensaio 1, 2 e 3 do sensor de gotejamento.

Como alternativa, o volume infundido foi monitorado a partir do número de voltas do rolete do motor de passo. Foi realizado um ensaio, e constatou-se que a cada volta completa do rolete do motor de passo, na saída da agulha foi de aproximadamente 0,5 ml.

Este volume de 0,5 ml se mantém constante para as taxas de infusão de 10, 25, 50, 100, 200, 300 e 400 ml/h. Isto demonstra que se pode fazer o monitoramento do volume através do número de voltas do motor. Mas isto não foi implementado.



4.2 DESCRIÇÃO DO SOFTWARE

Com a inicialização do programa, a mensagem que requisita a taxa de infusão é mostrada no display, que aguarda o valor digitado via teclado onde, após reconhecer este valor, analisa se é maior que 1 ou menor que 1000 ml/h. Caso o valor estiver fora desses parâmetros, apresenta mensagem no display de “taxa<1 ou taxa>1000 ml/h”, solicitando, então, que se pressione a Tecla “C” para fazer a correção do valor.

Se o valor estiver correto, a mensagem de “Tecla E para confirmar a taxa” é escrita no display e aguarda que se pressione a tecla “E” confirmando a valor da taxa em ml/h.

Na seqüência aparece a mensagem solicitando o volume total a ser infundido também em ml/h e que deve ser maior que a taxa de infusão. Sendo o valor menor, o display mostra mensagem “volume total < taxa” e solicita que se pressione a tecla “C” para correção do valor.

Caso queira alterar este valor, mesmo estando dentro do parâmetro, a tecla “C” deve ser pressionada para fazer a correção. Estando o valor do volume total correto, o display apresenta a mensagem de “tecla E para confirmar volume total” e aguarda que se pressione a tecla “E”.

Ao pressionar a tecla “E”, o display informa, em minutos, o tempo que vai ser necessário para a infusão e solicita que se pressione a tecla “E” novamente e, ao reconhecer o pressionamento da tecla, o software calcula e aciona o motor com o passo definido pela função.

No mesmo momento que o motor é acionado também inicia o processo de monitoramento através do Timer1, que está configurado como contador e que, a cada passagem da gota pelo sensor acoplado à câmara de gotejamento, faz o monitoramento do volume que está sendo infundido (maiores esclarecimentos no item 4.2.4) atualizando este valor no display. Este processo fica em “loop” até o fim do volume total infundido.

Ao final do volume infundido, definido no parâmetro inicial, o sistema aciona o alarme sonoro e entra na rotina de KVO, mostrando a mensagem no display “INFUSÃO COMPLETA INFUSÃO EM KVO” e aguarda que se pressione a tecla “F”



para que o processo se finalize, parando o alarme sonoro.

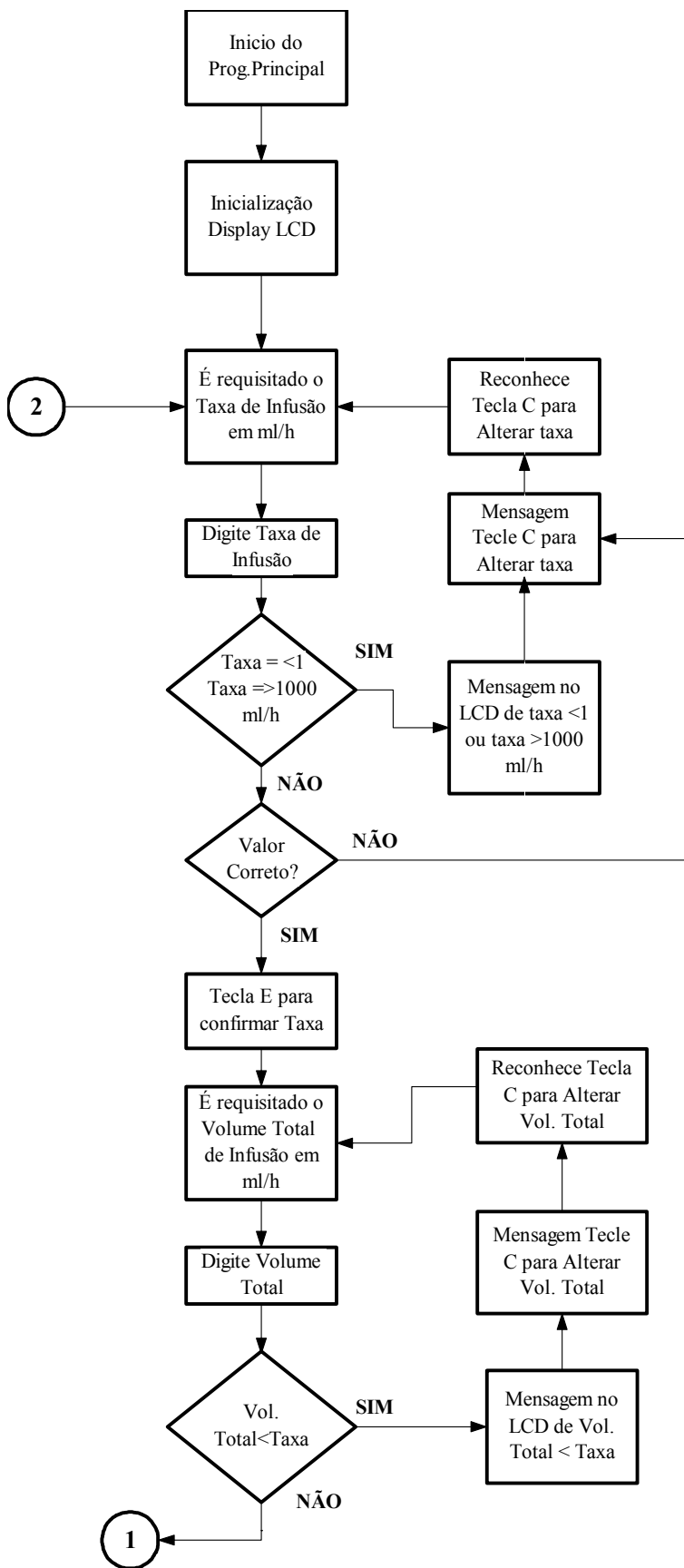
Desejando alterar os parâmetros da taxa de infusão ou o volume a infundir durante o processo, deve-se pressionar a tecla “F” para que o motor e o contador de gotas parem de funcionar. Neste momento, o display mostra a mensagem “A = ALTERA TAXA e C = LIMPA TAXA”.

Se a tecla “A” for pressionada, o valor do volume infundido é armazenado e, mesmo digitando-se outros valores dos parâmetros de taxa de infusão e volume total a ser infundido, o software continua monitorando o volume a partir do último valor de volume infundido armazenado. Caso a tecla “C” for pressionada, os parâmetros são alterados, mas o monitoramento do volume infundido reinicia-se a partir de zero.

A alteração dos parâmetros de taxa de infusão e de volume total a infundir, a qualquer momento do processo, ocorre com as mesmas mensagens descritas no início do software.



4.2.1 FLUXOGRAMA DO SOFTWARE



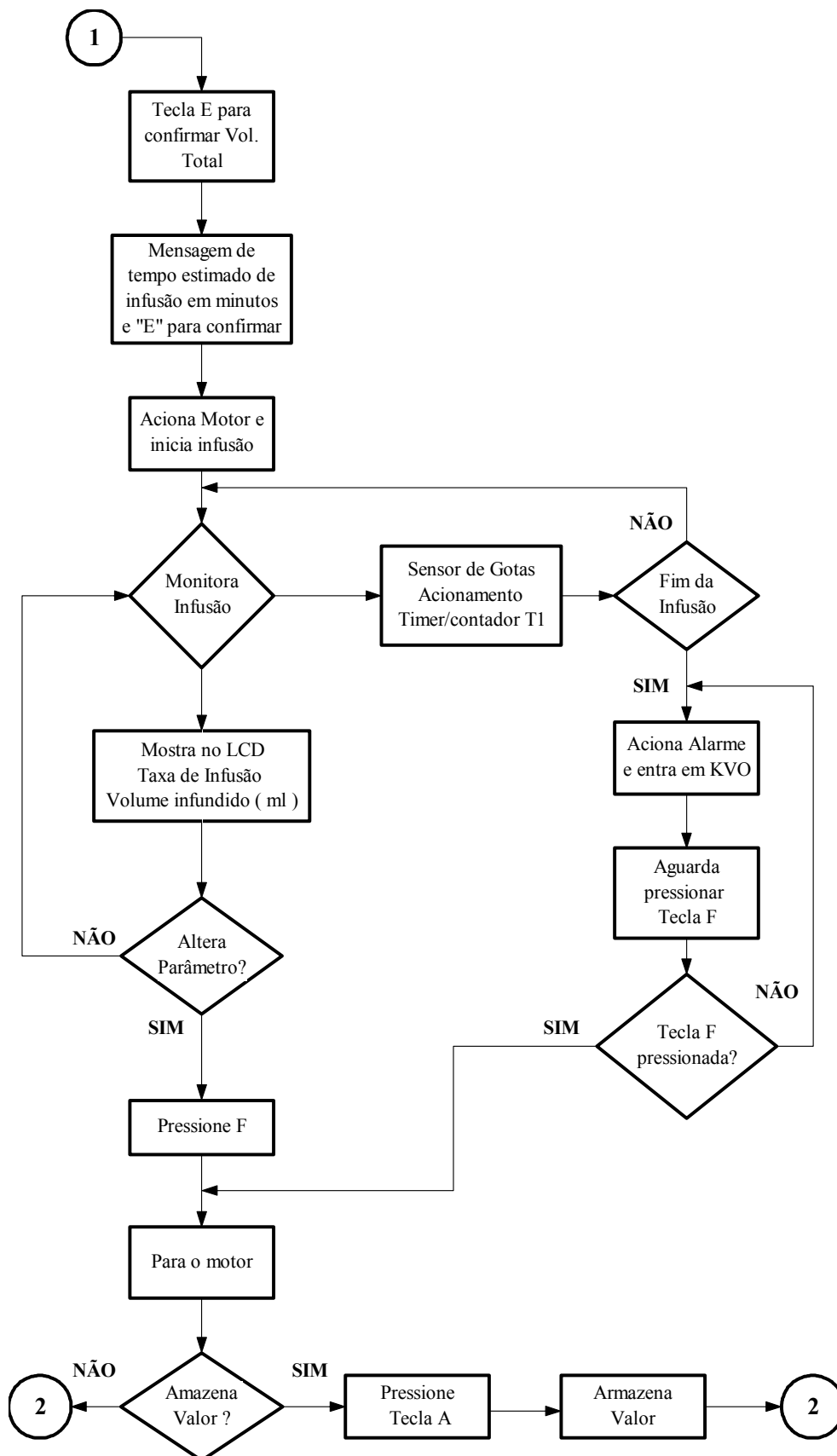


Figura 32: Fluxograma do Software



4.2.2 OPERAÇÃO DO EQUIPAMENTO

Neste capítulo, descreve-se o software desenvolvido utilizando o compilador SDCC MIDE Studio for MCS 51 para compilar o programa em linguagem C, sendo este transferido para o microcontrolador via comunicação serial utilizando o Hyper Terminal do Windows.

O software ao ser executado inicia o *display* onde aparece a primeira mensagem conforme a figura 33 solicitando a *taxa de infusão* em ml/h que deve ser digitada entre 1ml/h até 999 ml/h.



Figura 33: Mensagem de solicitação de taxa

Em seguida digita-se via teclado a seqüência de números 1, 0 e 0 que significa uma taxa de 100 ml/h, e que para melhor ilustração vamos utilizar esta taxa de infusão como exemplo conforme mostra a figura 34.



Figura 34: Mostra no display taxa digitada

Caso o valor seja menor que 1 ml/h aparece no display a mensagem “TAXA < 1 PRESSIONE TECLA C”, retornando para o menu inicial conforme figura 33.

Quando o valor da taxa estiver correto pressiona-se a tecla “E” o display LCD entra na próxima tela solicitando novamente a confirmação da taxa selecionada ou caso tenha necessidade de alteração pressiona-se no teclado letra “C” (figura 35).



Figura 35: Confirmação da taxa de infusão

A próxima solicitação do software é para digitar o volume total a infundir (figura 36)



Figura 36: Solicitação de volume total a infundir

Digita-se a seqüência de números 2, 0 e 0 que corresponde ao volume total a infundir de 200 ml, e se o valor estiver correto pressiona-se a tecla “E” para confirmar valor e se for necessário corrigir, pressiona-se tecla “C”. A seqüência é de confirmação do valor do volume total a infundir (figura 37).

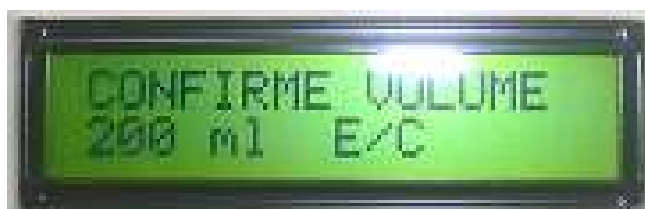


Figura 37: Confirmação do valor do volume total

Se o valor de volume total a infundir for menor que a taxa de infusão o display apresenta mensagem “VOLUME < TAXA PRESSIONE TECLA C”.

Quando taxa de infusão e volume total a infundir estão correto após confirma-se com o pressionamento da tecla “E” o software calcula e mostra no display o tempo para a infusão do volume total com a taxa correspondente (Figura 38).



Figura 38: Informação do tempo da infusão total

Com a confirmação dos dados o software aciona o motor de passo e inicia a infusão do líquido e o acionamento da rotina do Timer/contador que faz o monitoramento do volume através da contagem de gotas. Essa atualização do volume que esta sendo infundido é mostrada no display conforme figura 39.



Figura 39: Atualização do volume infundido

Caso deseja-se alterar a taxa de infusão pressionamos a tecla “F” que faz o motor parar e suspender a infusão e mostra as alternativas na tela do display.conforme a figura 40. Onde “A” (Alterar taxa) significa que pode-se alterar a taxa, mantendo armazenado o volume que estava sendo infundido e com a nova taxa alterada ele continua monitorando o volume a partir desse valor.

A alternativa “C” (Limpar Taxa) zera todo o valor armazenado do volume infundido, iniciando uma nova contagem do volume.



Figura 40: Alteração de parâmetros de infusão

Após o término do volume total infundido o software não para a infusão e para manter a veia aberta o sistema entra na infusão KVO (Keep Vein Open), que é infusão mínima ao final da infusão do volume total, mantendo o vaso sanguíneo

aberto (Figura 41).



Figura 41: Fim de infusão módulo KVO

Para sair desta parte do software pressiona-se a tecla “F” para indicar o fim da infusão e o programa retornar para o início solicitando novamente os parâmetros.

4.2.3 ROTINAS DE DISPLAY

Maiores detalhes das rotinas e sub-rotinas utilizadas como: timer, rotinas de delay nos itens 4.2.3 e 4.2.4, por serem essenciais na construção do projeto e responsável pela precisão do equipamento.

Para habilitação do display estamos usou-se pino *strobe2* do KIT ULBEE51 que corresponde ao pino 4 do conector 8 .

Para habilitar o pino RS (Register Select) que corresponde ao pino 6 do display utilizou-se o pino P1_4 do microcontrolador 8051, onde a habilitação e desabilitação do pino é feita via software .

Conforme mostra a figura 16 pode-se conectar o barramento de dados do KIT ULBEE51 ao barramento do display usando-se uma comunicação de 8 bits (8051). Neste caso os 8 bits de sinais DB0 à DB7 estão conectados aos pinos 9 à 16, onde tem-se 8 endereços distintos para comunicação entre o microcontrolador e o Display, onde pode-se efetuar a operação de leitura/escrita sem problemas.

Para a execução das instruções é necessário temporizações mínimas referente à transição do nível “0” para nível “1”, onde nestes softwares uso-se delay de 10ms. Este delay é necessário entre uma instrução e outra a fim de ajustar a velocidade de comunicação entre a CPU do 8051 e o módulo do Display.

Quem comanda a opção da comunicação ser um “dado” ou “instrução” é o



pino R/S do display. Quem faz o display receber a informação (dado ou instrução) é o pino E(Enable) do display.[9]

Quando assumi-se que um o display foi configurado para aceitar instruções de 8 bits as quais são direcionadas ao registrador de instruções quando RS=0 e R/W=0, utiliza-se a tabela 5 de instruções para facilitar a escrita de programa.[9]

Devido a não ser o objetivo principal deste projeto, não descreveu-se todas as instruções de uso para o display, apenas algumas para demonstrar o uso delas no software. As rotinas utilizadas estão declaradas no software declarado no anexo E.

Limpeza do Display: Esta instrução escreve o caractere ASCII 20 (hexadecimal) que corresponde a “branco” em todos os endereços da DD RAM “limpando” toda e qualquer mensagem escrita, com o retorno do cursor com a posição mais à esquerda na primeira linha. [9]

Retorno do cursor: Esta instrução faz retornar o cursor mais a esquerda da primeira linha e faz retornar à posição original as mensagens previamente deslocadas.

Deslocamento do cursor ou da mensagem: Esta instrução desloca o cursor ou da mensagem para a direita ou para a esquerda sem que seja necessário escrever/ler dados do display. Esta função é utilizada para correção de dados no display.

As demais rotinas não serão descritas, por não ser o principal objetivo deste trabalho, apenas demonstradas a seguir.

4.2.4 ROTINA DE PASSO

Para definição do passo do motor foi necessária uma rotina de delay de 100 microsegundos, para termos uma boa resolução. A partir dela podemos gerar tempos de, por exemplo, 5,5 ms.

A rotina que se utilizou foi uma implementada dentro de um uma diretiva ASM/ENDASM. Onde a diretiva ASM significa o início de um bloco de código-fonte-texto Assembly para ser inserido no arquivo (Assembly in-line). Esse código fonte-fonte Assembly de inclusão não é compilado, ele é anexado diretamente ao arquivo gerado pelo compilador.



A diretiva ENDASM sinaliza o fim do bloco de texto Assembly.

```
//*****ROTINA DE 100u SEGUNDOS*****//
```

```
void delay_100micros(unsigned int tempo)
{
  unsigned int c;
  for (c=0 ; c<tempo; c++)
  {
    if(le_teclado_tecla_F() > 10)
    {
      tecla_F_precionada = 70;
    }

    _asm
    mov r7,#12
    100$:
    nop
    djnz r7,100$
    _endasm;
  }
}
```

Utilizou-se esta rotina de 100 microsegundos para permitir uma maior precisão no passo do motor, devido a uma rotina que gere passo de em milisegundos (delaysms) não possibilitar a precisão necessária.

Para deixar o uso desta rotina mais claro segue na figura 42 um exemplo do valor setado em função das curvas que geram o passo do motor. Onde se escolheu uma taxa de infusão de 30 ml/h representado na curva e equação da figura 42 e onde os dados constam na Tabela 4 do apêndice B os valores calculados para a função linear.

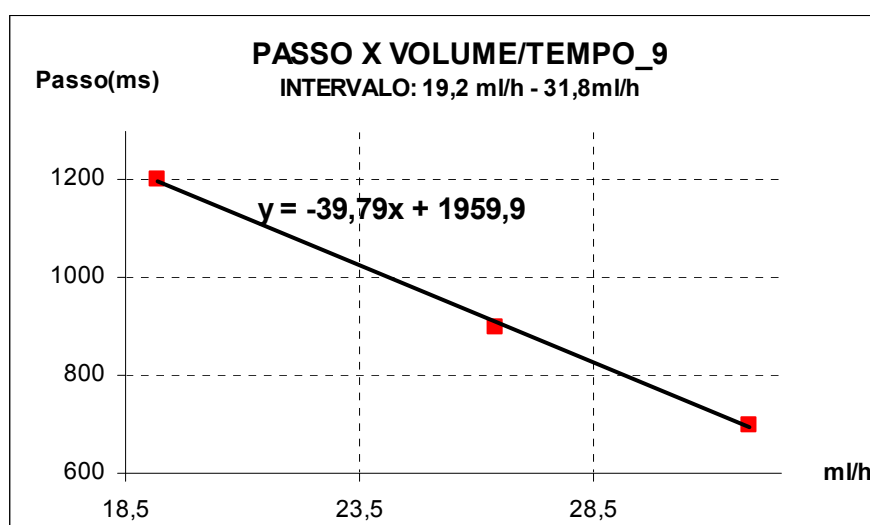


Figura 42: Curva do passo do intervalo de 19,2 ml/h à 31,8 ml/h.

Utilizou-se a valor escolhido de taxa de infusão desejada, no caso do exemplo 30 ml/h que é digitado via teclado externo, e o software executa o cálculo para gerar o passo do motor. O cálculo executado é o descrito abaixo:

$$\text{Valor_Passo} = -39,79 \times \text{Taxa_Infusão} + 1959,9$$

$$\text{Valor_Passo} = -39,79 \times 30 + 1959,9$$

$$\text{Valor_Passo} = 766,2$$

$$\text{Passo_Motor} = 100 \mu\text{s} \times \text{Valor_Passo}$$

$$\text{Passo_Motor} = 100 \mu\text{s} \times 766,2$$

$$\text{Passo_Motor} = 76,6 \text{ms}$$

Este valor é aferido com um osciloscópio Digital TDS320 do Fabricante TEKTRONIX osciloscópio conforme a figura 43.

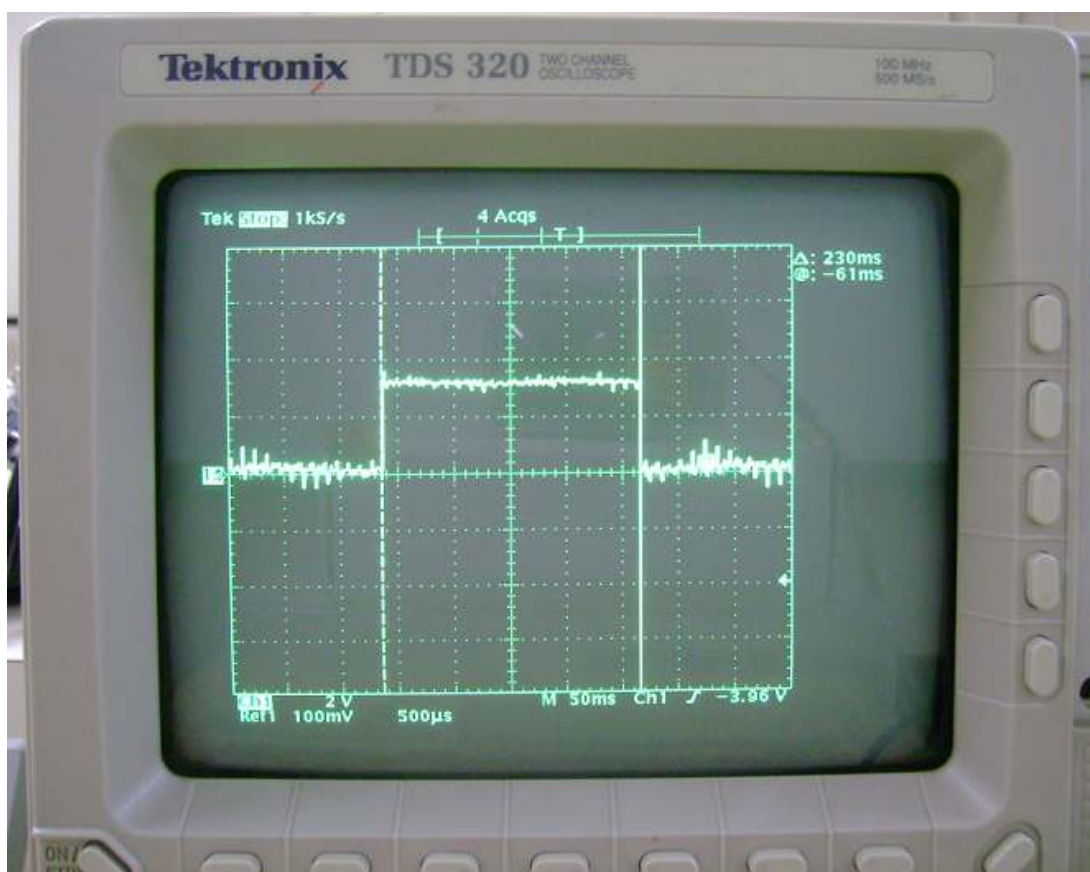


Figura 43: Medida do passo do intervalo de 19,2 ml/h à 31,8 ml/h (Nível Alto).

Esta medida é feita em um dos pinos que sai da porta P1 do microcontrolador



e dividido o valor medido por três, devido ao pino permanecer em nível alto durante três passos, conforme cálculo abaixo:

$$\text{Valor_Medido_Nível_Alto} = \frac{230ms}{3} = 76,66ms$$

Também aferiu-se o valor no momento que o pino esta no nível baixo conforme mostrado na figura 44, dividindo-se o valor por cinco, devido ao pino permanecer em nível baixo durante cinco passos, conforme cálculo abaixo.

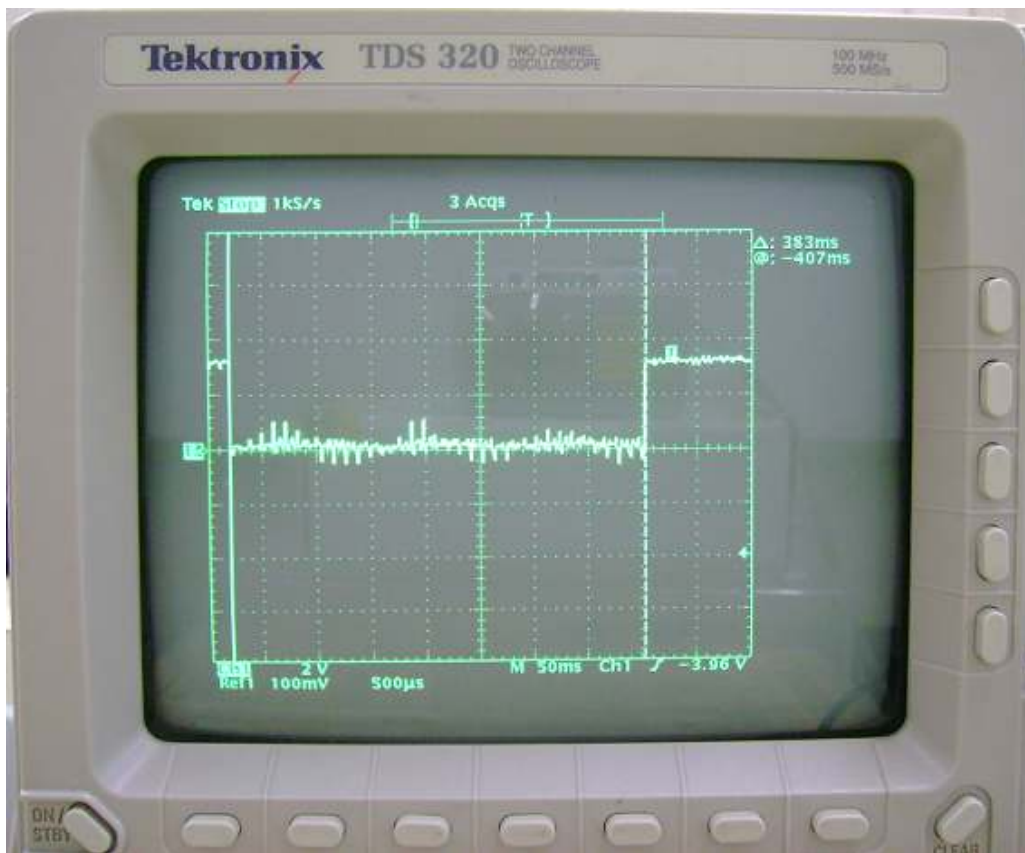


Figura 44: Medida do passo do intervalo de 19,2 ml/h à 31,8 ml/h (Nível Baixo

)

$$\text{Valor_Medido_Nível_Baixo} = \frac{383ms}{5} = 76,6ms$$



4.2.5 ROTINAS USADAS COM O SENSOR DE GOTEJAMENTO

Para fazer o monitoramento do volume utiliza-se o sensor, conforme o hardware descrito no item 3.5, e, via software, utiliza-se o acionamento do timer1 para fazer a contagem de gotas.

O AT89C52 tem dois registradores de 16 bits para timer/counter, TIMER 0 e TIMER 1 e ambos podem ser programados como timer ou counter (temporizador ou contador).

Para atender a necessidade programa-se o Timer 1 como counter (contador), onde o registrador é incrementado em resposta à transição de 1 para 0 na entrada externa correspondente T0 ou T1. No caso utiliza-se o T1, onde a saída do sensor esta ligada no pino P3_5 do AT89C52.

Nesta função a entrada é amostrada em todo o ciclo de máquina e quando uma amostragem tem nível 1 em um ciclo em nível 0 no próximo ciclo, o contador é incrementado, fazendo que um novo valor apareça no registrador no ciclo seguinte, aquele em que a transição foi detectada.

Também teremos que configurar os timers/counters no MODO 2, que configura o registrador (TLx) dos contadores como sendo contadores de 8 bits com auto-reload (autocarregamento).

Quando ocorre o overflow (estouro) de TLx, esse overflow é sinalizado em TFx e o valor contido no registrador THx é automaticamente carregado em TLx. O valor de recarga do timer é inicializado via software, e quando ocorre a recarga, o registrador THx mantém seu valor inicial, somente copiando este valor para o registrador TLx.

Segue parte do software onde habilita-se o Timer 1 como contador e valores estipulados para o THx, que no caso é FFH, onde o TLx é setado com o valor de EDH, que é valor inicial onde vai incrementando 1 a cada passada da cota pelo sensor e quando ocorrer o overflow, ele incrementa 1 ml, através do segundo bloco do software, e atualiza o display.

O incremento do volume é feito de 1 em 1 ml, e que, no caso cada ml corresponde a 18 gotas que passaram pelo sensor, sendo nesta caso a diferença de FFH e EDH.



// Bloco de habilitação do Timer 1

```
IE = 0;           // Registrador IE em 0 , desabilita todas as interrupções
ET1 = 1;         //Interrupção do TIMER1, ET1=1, Interrupção habilitada
EA = 1;         //Interrupções habilitadas
TH1 = 0xFF;     // 255d
TL1 = 0xED;     //237d
****REGISTRADOR TMOD****
TMOD = 0x50; //0101 0000, TIMER1
TR1 = 1;
```

// Rotina que incrementa a variável ml e escreve no display LCD

```
void contagem_ml (void )interrupt 3           //toda vez que o pino for de 1 a 0 vai entrar //nesta
                                              função que conta a quantidade de //gotas passadas
                                              pelo emblo.

{
    TR1 = off;
    EA = off;
    ml_parcial++; //incrementa o ml
    atualiza_display = 1;
    TH1 = 0xFF;
    TL1 = 0xED;
    TR1= on;
    EA = on;
}
```



5 CONCLUSÕES

Conforme foi visto o longo do trabalho, uma bomba de infusão peristáltica é importante para tratamentos, via intravenosa, em pacientes. Ela permite uma infusão do medicamento prescrito pelo médico de forma mais precisa, podendo ser usada, por exemplo, na aplicação de drogas para controlar a pressão arterial, de anestésicos, de hormônios, dentre outras.

Na realização desse projeto foram utilizados os conhecimentos adquiridos ao longo do curso de eletrônica, assim como a linguagem de programação C. Além disso, foi pesquisado material bibliográfico sobre as bombas de infusão e seu funcionamento.

No desenvolvimento do trabalho obteve-se um protótipo funcional de infusão volumétrica composto por bomba peristáltica acionada pelo motor de passo e o “equipo” (conjunto reservatório, tubos e gotejador), por onde flui o medicamento e que opera entre 1ml/h à 999 ml/h. O monitoramento do volume infundido é feito através do software e de sensor acoplado à câmara de gotejamento.

A interface com o usuário é realizada através de um *display*, que mostra as mensagens e valores digitados, e o teclado, que recebe as condições de ajuste do equipamento.

Conclui-se que o projeto atingiu o seu objetivo que era de fazer funcionar um equipamento de monitoramento volumétrico com a maior precisão possível. Obteve-se uma precisão de 95% na infusão do volume definido pelo usuário através da medição na proveta, isto significa que, com alguns incrementos no software e hardware, pode-se ter um equipamento bem semelhante ao comercial, que monitora e indica fim de volume.

Uma dificuldade encontrada na realização do trabalho foi o monitoramento do volume infundido através do sensor de gotas, que não apresentou ser eficaz devido à variação do tamanho, forma e velocidade da gota no equipo, ocasionando um elevado percentual de erro, conforme descrito no item 4.1.3. Como alternativa, foi realizado um ensaio monitorando o número de voltas do motor de passo e constatou-se que, a cada volta completa do rolete do motor, na saída do cateter agulhado, saiu um volume infundido de aproximadamente 0,5 ml, nas várias taxas



de infusões definidas.

Neste ensaio observou-se que ao fazer o monitoramento do volume infundido através das voltas do motor tem-se uma precisão maior, pois a cada 2 voltas completas corresponde a 1ml de volume infundido. Logo, para trabalhos futuros, sugere-se que o procedimento descrito acima seja uma alternativa válida para monitorar o volume.

Também se sugere a implementação de outras estratégias, tais como, usar o sensor de gotejamento para monitorar o fim do líquido no reservatório e o fluxo livre no equipo, assim como outros sistemas de infusão como bombas de infusão com motor linear ou tipo seringa.



REFERÊNCIAS

- [1] MUHLENA, Sérgio Santos. **Dispositivos de Infusão** –
- [2] PRATICO 1, Trabalho. Faculdade de Ciências e Tecnologia Universidade Plena, http://www.ime.eb.br/~pinho/micro/trabalhos/Mecatronica_TP1.pdf. acesso 2/12/2007
- [3] www.datasheet4U.com – Acesso em 20/04/2008
- [4] <http://www.microcontrolador.com.br> – Acesso em 01/04/2008
- [5]: SOUZA, Uilian Lucas de Souza. **Fotodiodo e Fototransistores**. <http://www.eletrica.ufpr.br/piazza/materiais/Uilian&Thiago.pdf> - Acesso em 01/05/2008.
- [6]: KAVATURO, André. **Desenvolvimento Bomba de Infusão de Seringa Microprocessada**. Universidade Tecnológica Federal do Paraná, Paraná, 2006. <http://www.daeln.ct.utfpr.edu.br/~tcc-daeln/DEZ2006/defendidos2006.pdf> - Acesso em 15/01/2008.
- [7]: ALVES, Márcio Alexandre de Castro. **Bombas de Infusão: Operação, Funcionalidade e Segurança**, Universidade Federal de Santa Catarina, Santa Catarina, 2002. <http://www.tede.ufsc.br/teses/PEEL0782.pdf> –Acesso em 20/02/2008.
- [8]: MEINCKE, Marcelino. **Display LCD: 2002**.
- [9]: NICOLOSI, Denys Emílio Campion. **Microcontrolador 8051 com Linguagem C: Prático e Didático**. 1 ed. São Paulo: Érica, 2005.
- [10]: B.BRAUN , Laboratórios. **Manual do Usuário**. Revisão 04. Rio de Janeiro.



OBRAS CONSULTADAS

NICOLOSI, D. E. C. **Laboratório de Microcontroladores Família 8051.** – 1 ed. São Paulo: Érica, 2004.

SÁ, Mauricio Cardoso de. **Programação C para Microcontroladores 8051.** 1 ed. São Paulo: Érica, 2005.

EVARISTO, Jaime. **Aprendendo a Programar C para Iniciantes.** 1 ed. Rio de Janeiro: Book Express, 2001.

NICOLOSI, Denys Emílio Campion. **Microcontrolador 8051 com Linguagem C: Prático e Didático.** 1 ed. São Paulo: Érica, 2005.

NICOLOSI, D. E. C. **Microcontrolador 8051 Detalhado.** 1 ed. São Paulo: Érica, 2000.

HIRAMA, R.T. **Método para Inspeção de Bombas Infusoras.** São José dos Campos: Universidade Estadual de Londrina, 2002.
<http://www2.uel.br/projetos/ec/Producao/bominf.PDF> - Acesso em 15/03/2008

LOPES, Daniel Fernandes. **Bombas de Infusão.** São Paulo: Centro Universitário Salesiano de São Paulo, 2006.

ALVES, Márcio Alexandre de Castro. **Bombas de Infusão: Operação Funcionalidade e Segurança.** Santa Catarina: Universidade Federal de Santa Catarina, 2002.

KAVATURO, André. **Desenvolvimento Bomba Seringa Microprocessada.** Universidade Tecnológica Federal do Paraná, 2006.
<http://www.daeln.ct.utfpr.edu.br/~tcc-daeln/DEZ2006/defendidos2006.pdf>- Acesso em 15/01/2008.

Sites consultados:

Disponível em: <http://www.provitec.com.br/> - Acesso em 31/08/2007

Disponível em: <http://www.rogercom.com> – Acesso em 04/10/2007



Disponível em: www.nmbtc.com- Acesso em 01/04/2008

Revista Latino Americana de Tecnologia Extracorpórea.
<http://perflin.com/revista/volume9/v9n4/index.html>- Acesso em 01/01/2008.

MEINCKE, Marcelino. **Display LCD**: 2002.

NICOLOSI, Denys Emílio Campion. **Microcontrolador 8051 com Linguagem C: Prático e Didático**. 1 ed. São Paulo: Érica, 2005.

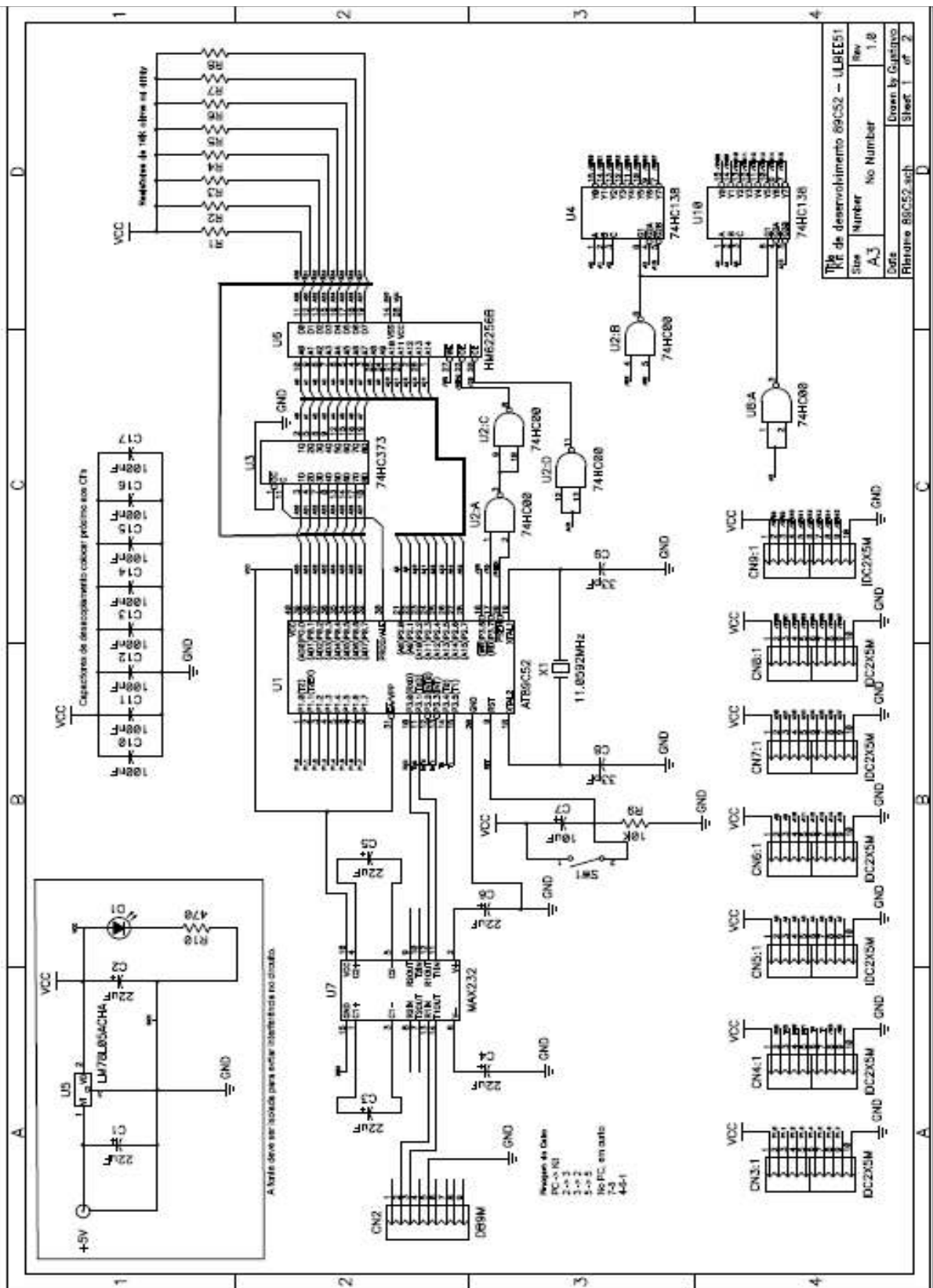
B.BRAUN , Laboratórios. **Manual do Usuário**. Revisão 04. Rio de Janeiro



ANEXOS

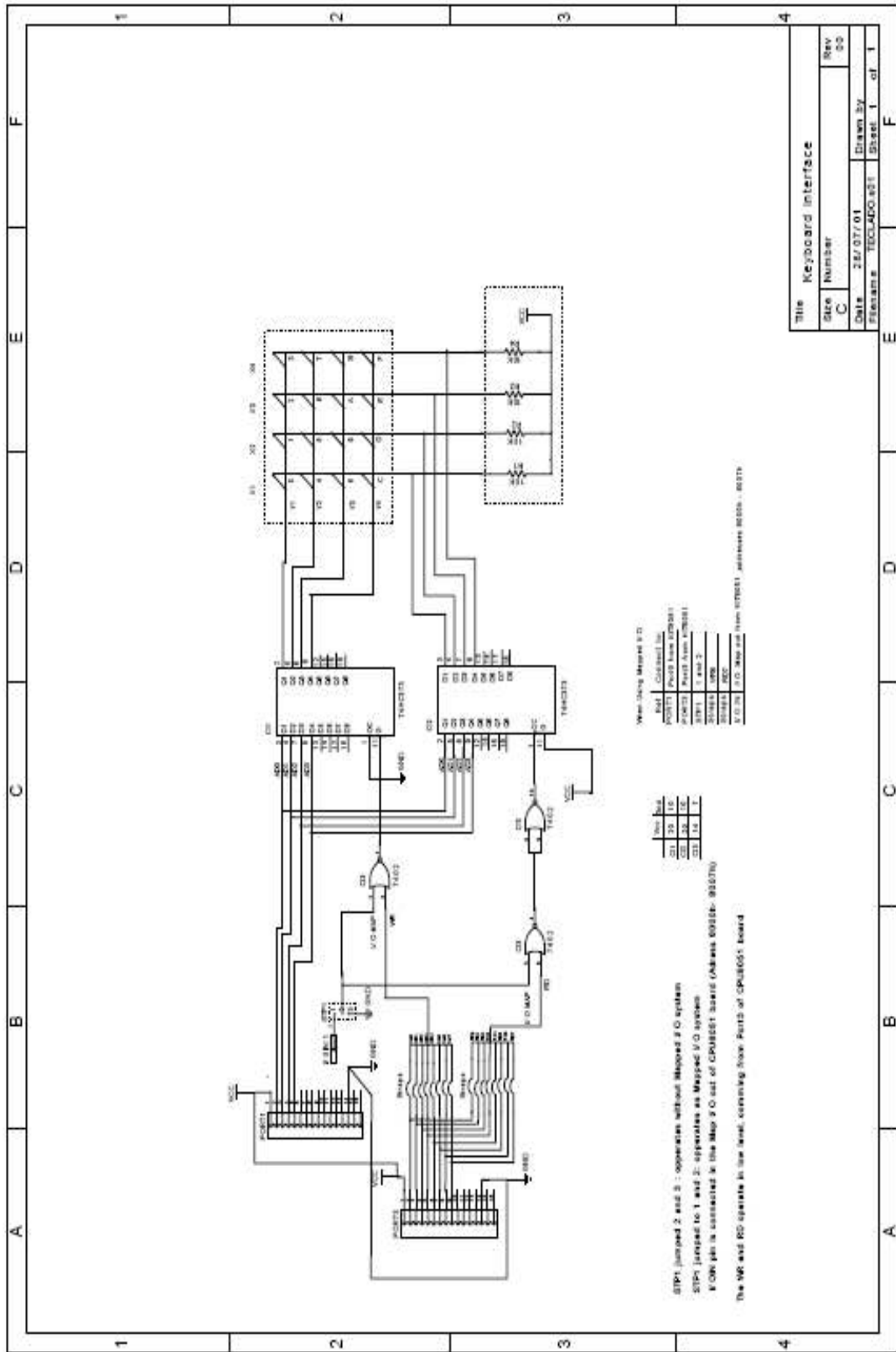


ANEXO A – HARDWARE KIT ULBEE51





ANEXO B – HARDWARE TECLADO 16 TECLAS





APÊNDICE A – CURVAS E CÁLCULOS PARA PASSO DO MOTOR

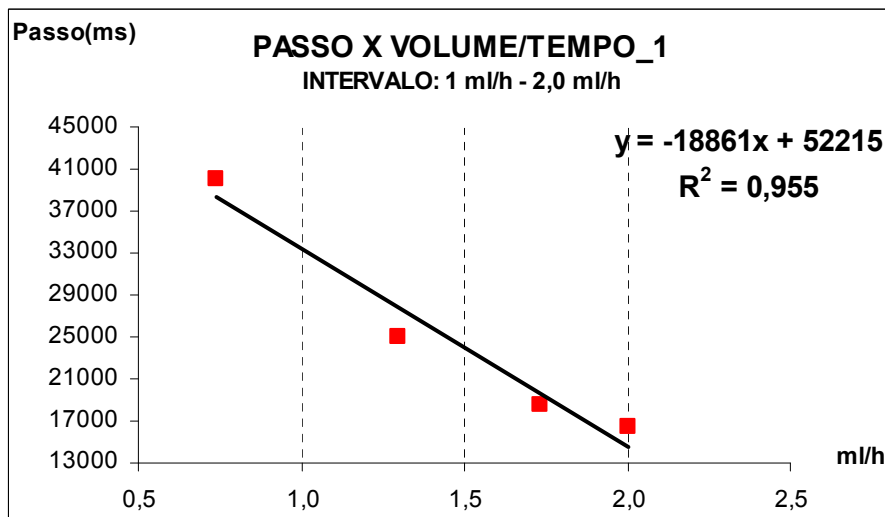
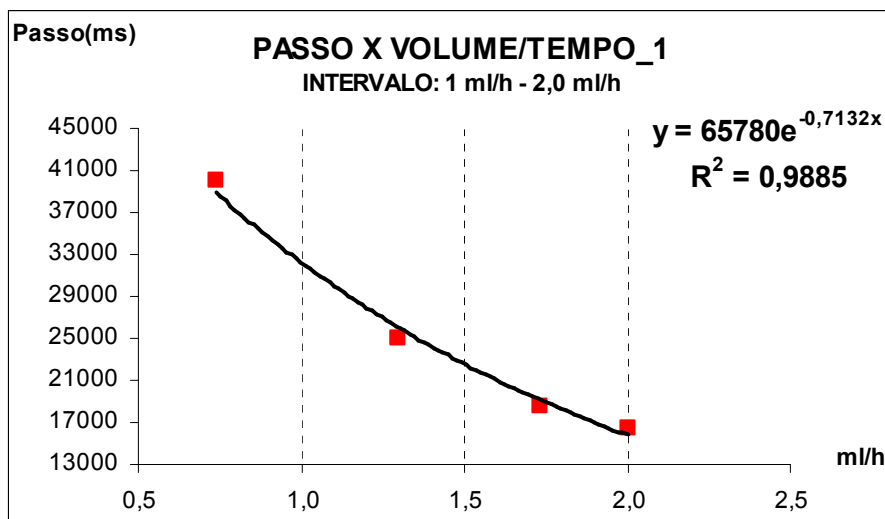
**Tabela 1**

Valores obtidos com o primeiro ensaio de Passo x Volume/Tempo refer. a figura 26

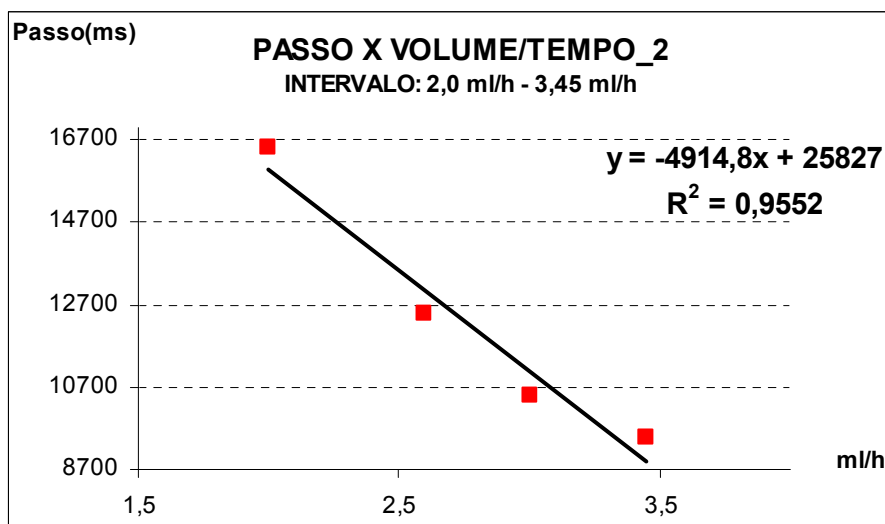
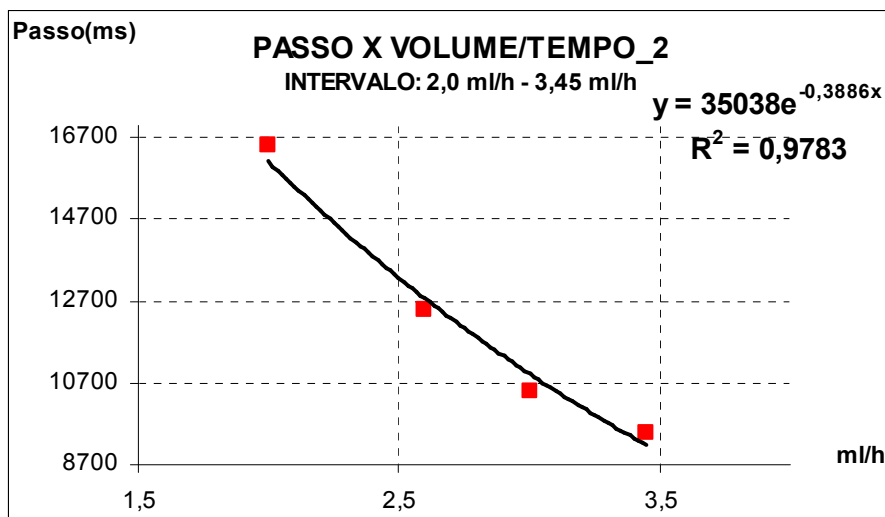
Valor Passo	Tempo Medido (minutos)	Volume Medido (ml)	Volume Estimado (ml em 60 min)
40000	00:01:05	0,80	0,74
25000	01:00:00	1,3	1,30
18500	00:45:00	1,3	1,73
16500	00:30:00	1,00	2,00
12500	00:30:00	1,30	2,60
10500	00:30:00	1,50	3,00
9500	00:20:00	1,15	3,45
8500	00:20:00	1,25	3,75
7500	00:20:00	1,30	3,90
6500	00:20:00	1,60	4,80
5500	00:15:00	1,30	5,20
4500	00:12:00	1,25	6,25
4000	00:12:00	1,40	7,00
3600	00:12:00	1,50	7,50
3200	00:10:00	1,40	8,40
2800	00:08:00	1,300	9,75
2600	00:08:00	1,400	10,50
2400	00:08:00	1,500	11,25
2200	00:08:00	1,600	12,00
1500	00:10:00	2,70	16,20
1200	00:10:00	3,2	19,20
900	00:10:00	4,40	26,40
700	00:10:00	5,30	31,80
500	00:10:00	7,20	43,20
400	00:05:00	4,70	56,40
320	00:03:00	3,50	70,00
300	00:05:00	6,10	73,20
290	00:03:00	4,00	80,00
250	00:03:00	4,50	90,00
210	00:03:00	5,20	104,00
170	00:03:00	6,20	124,00
150	00:03:00	6,95	139,00
110	00:03:00	9,60	192,00
100	00:02:00	7,20	216,00
90	00:02:00	7,80	234,00
80	00:01:00	4,60	276,00
70	00:02:00	9,80	294,00
60	00:01:00	5,80	348,00
55	00:01:00	6,40	384,00
50	00:01:00	7,00	420,00
40	00:01:00	8,60	516,00
35	00:01:00	9,60	576,00
30	00:00:30	5,80	696,00
25	00:00:30	6,80	816,00
20	00:00:30	8,20	984,00



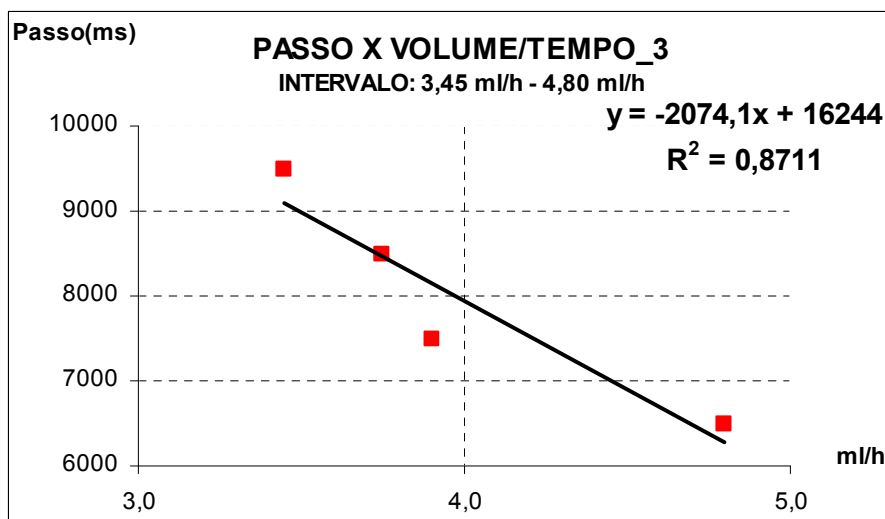
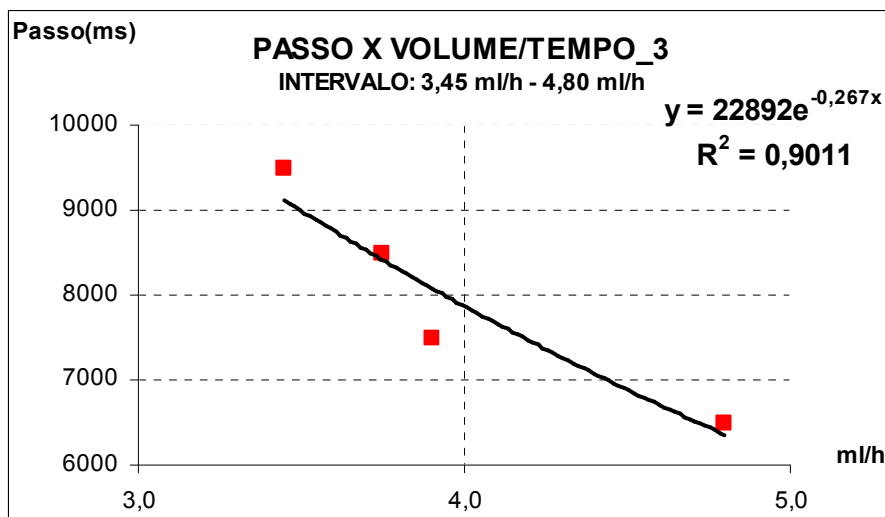
Fragmentos de retas e linhas de ajuste para a curva referente a figura 26.



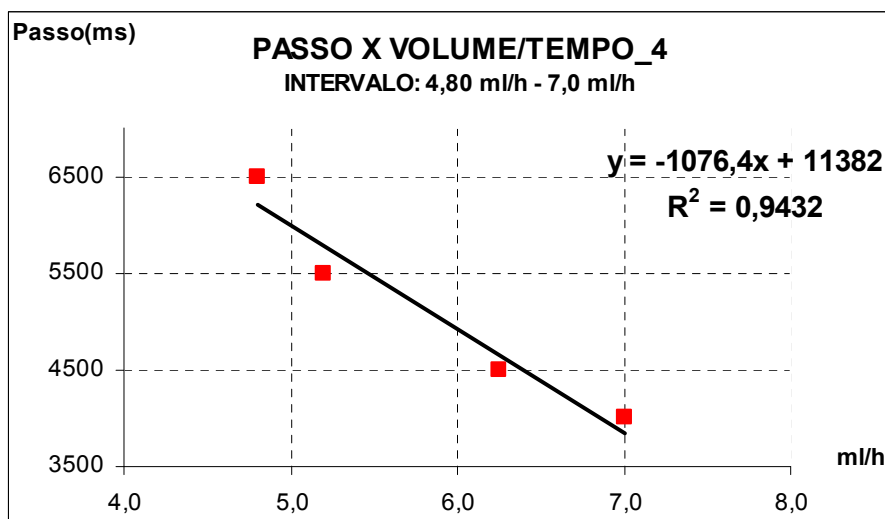
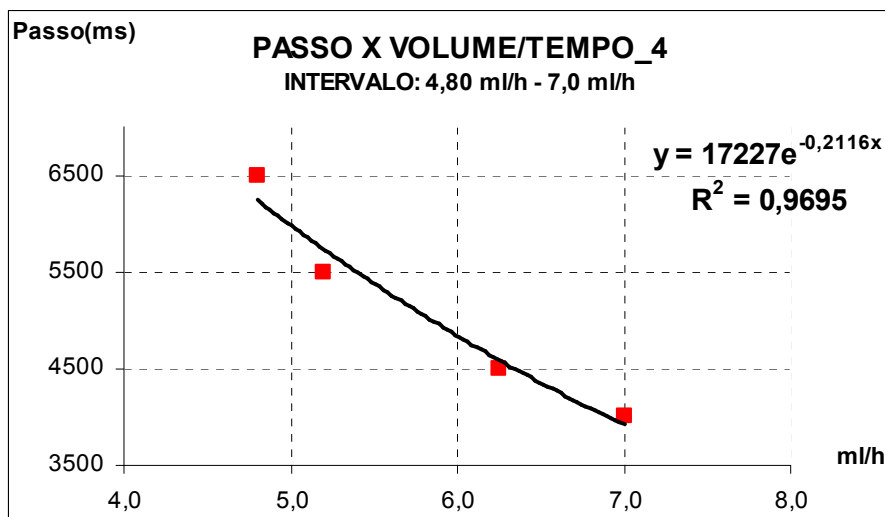
PASSO x VOLUME/TEMPO_1					
Medido		Calculado Exponencial		Calculado Linear	
		$y=65780e(-0,7132x)$		$y=-18861*x + 52215$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
40000	0,74	39927,92	-0,18%	38288,04	-4,28%
25000	1,30	26027,56	4,11%	27695,70	10,78%
18500	1,73	19567,61	5,77%	19523,23	5,53%
16500	2,00	15798,51	-4,25%	14493,00	-12,16%



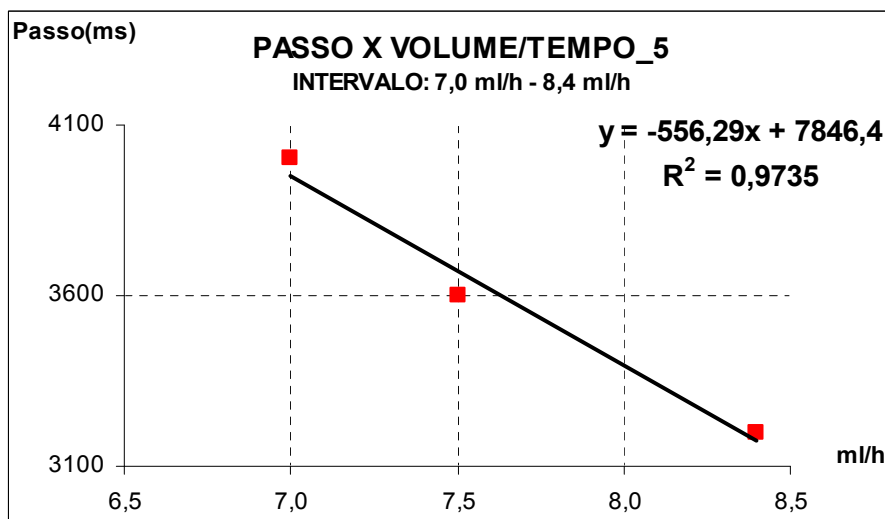
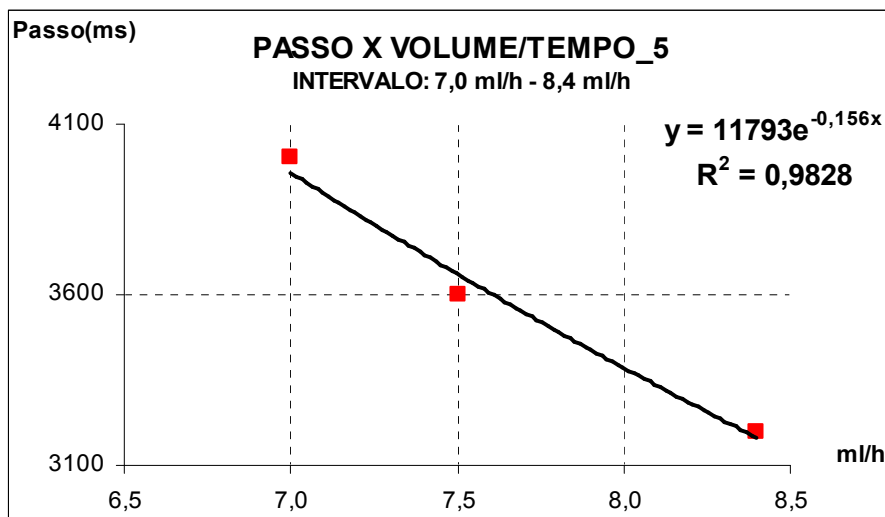
PASSO x VOLUME/TEMPO_2					
Medido		Calculado Exponencial		Calculado Linear	
		$y=35038e(-0,3886x)$		$y=- 4914,8*x + 25827$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
16500	2,00	16106,66	-2,38%	15997,40	-3,05%
12500	2,60	12756,91	2,06%	13048,52	4,39%
10500	3,00	10920,40	4,00%	11082,60	5,55%
9500	3,45	9168,40	-3,49%	8870,94	-6,62%



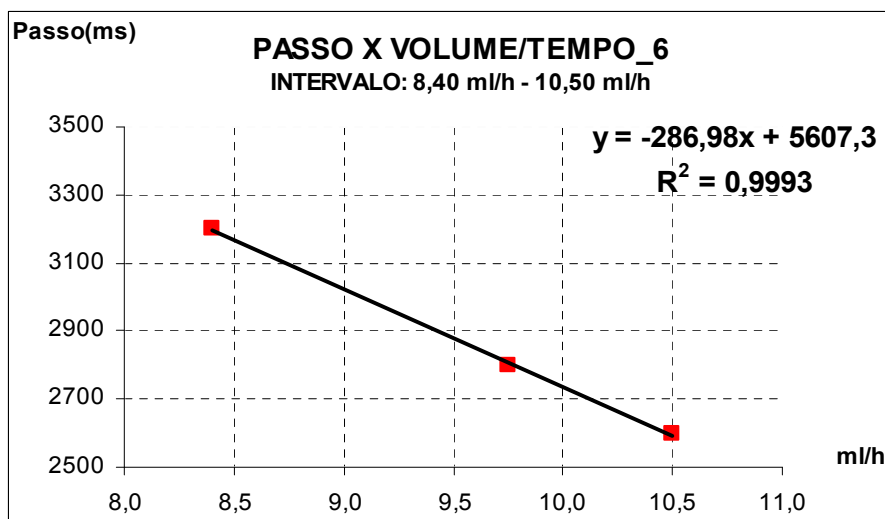
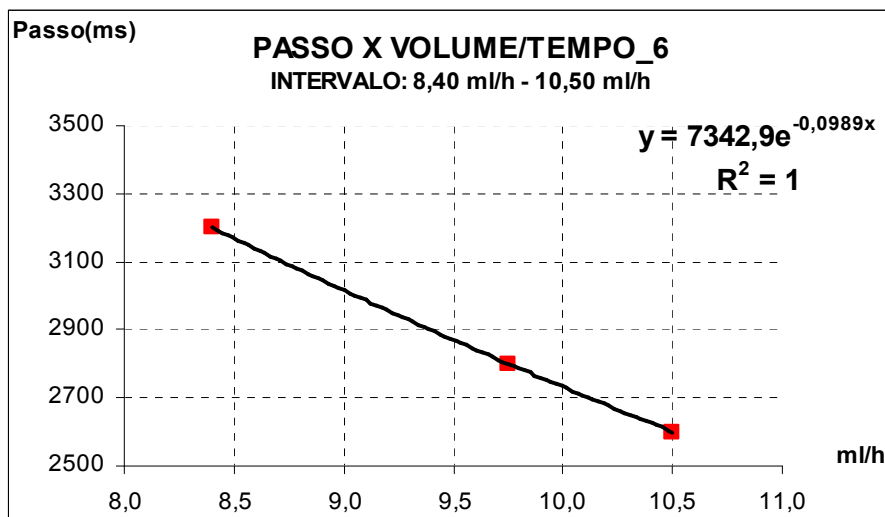
PASSO x VOLUME/TEMPO_3					
Medido		Calculado Exponencial		Calculado Linear	
		$y=22892e(-0,267x)$		$y=- 2074*x + 16244$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
9500	3,45	9112,41	-4,080%	9088,355	-4,333%
8500	3,75	8410,97	-1,047%	8466,125	-0,399%
7500	3,90	8080,77	7,744%	8155,01	8,733%
6500	4,80	6354,65	-2,236%	6288,32	-3,257%



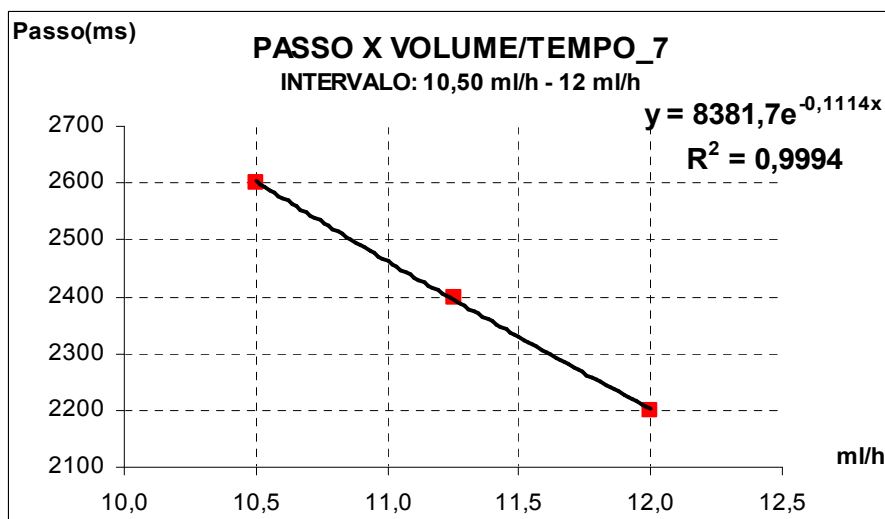
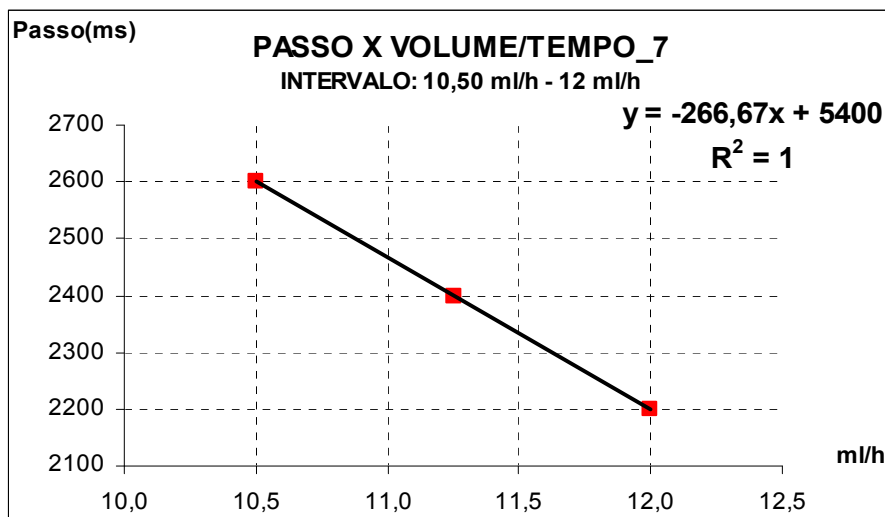
PASSO x VOLUME/TEMPO_4					
Medido		Calculado Exponencial		Calculado Linear	
		$y=17227e(-0,2116x)$		$y=- 1076,4*x + 11382$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
6500	4,80	6238,86	-4,02%	6215,28	-4,38%
5500	5,20	5732,54	4,23%	5784,72	5,18%
4500	6,25	4590,45	2,01%	4654,50	3,43%
4000	7,00	3916,81	-2,08%	3847,20	-3,82%



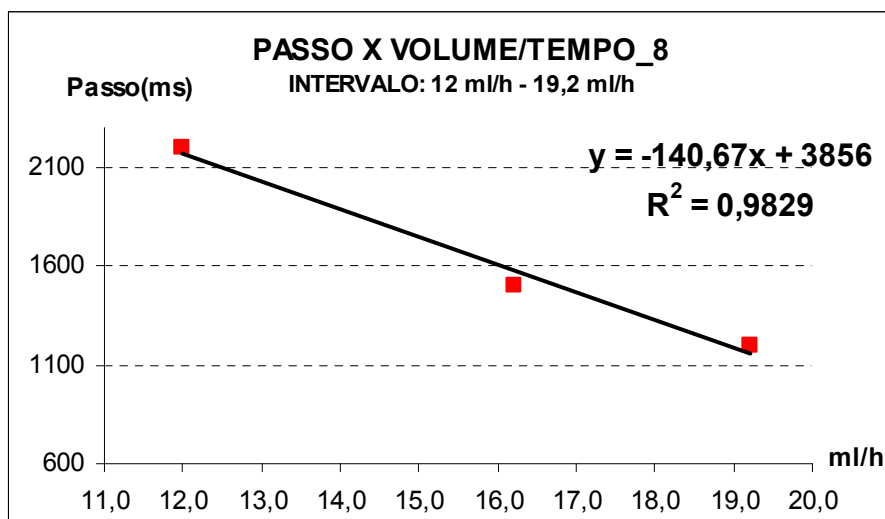
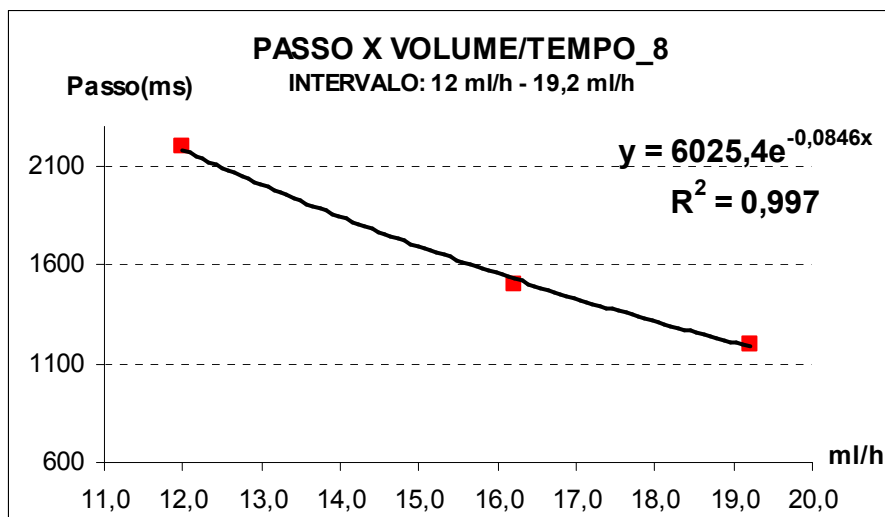
PASSO x VOLUME/TEMPO_5					
Medido		Calculado Exponencial		Calculado Linear	
		y=11793e(-0,156x)		y=- 556,29*x + 7846,4	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
4000	7,00	3957,07	-1,07%	3952,37	-1,19%
3600	7,50	3660,15	1,67%	3674,23	2,06%
3200	8,40	3180,71	-0,60%	3173,56	-0,83%



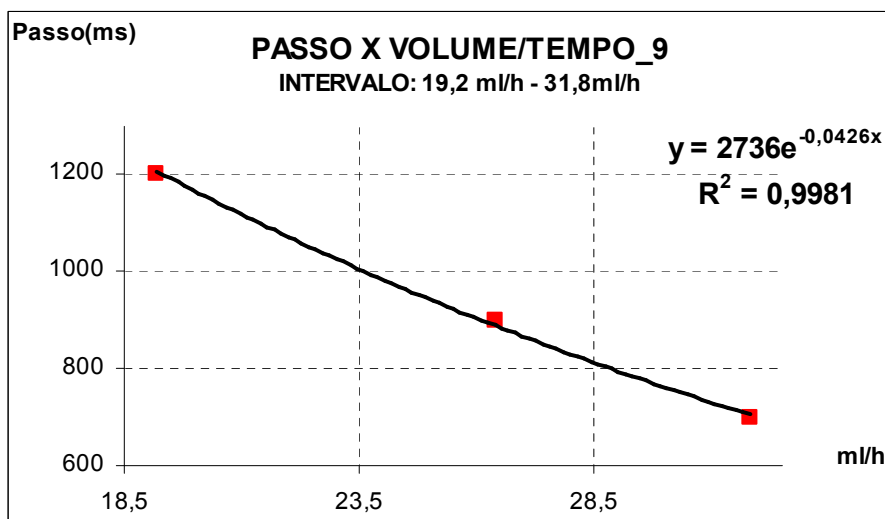
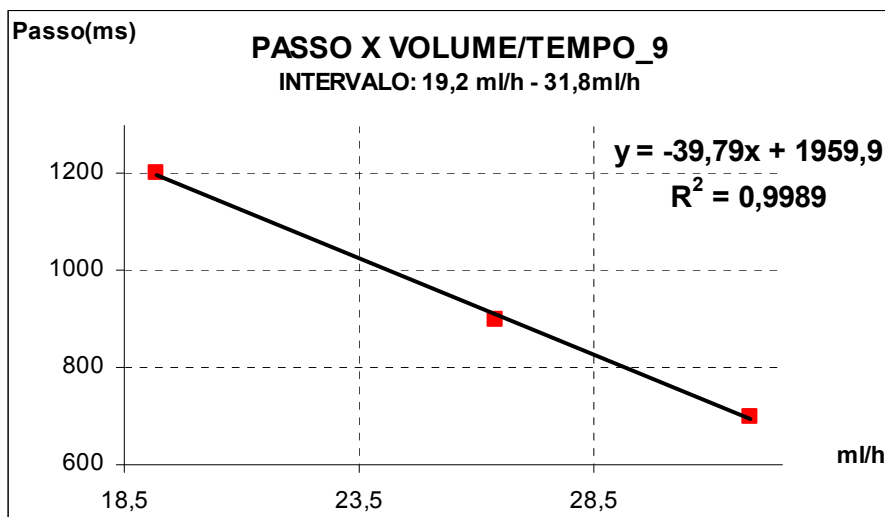
PASSO x VOLUME/TEMPO_6					
Medido		Calculado Exponencial		Calculado Linear	
		$y=7342,9e^{-0,0989x}$		$y=- 286,98*x + 5607,3$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
3200	8,40	3199,43	-0,02%	3196,668	-0,10%
2800	9,75	2799,55	-0,02%	2809,25	0,33%
2600	10,50	2599,40	-0,02%	2594,01	-0,23%



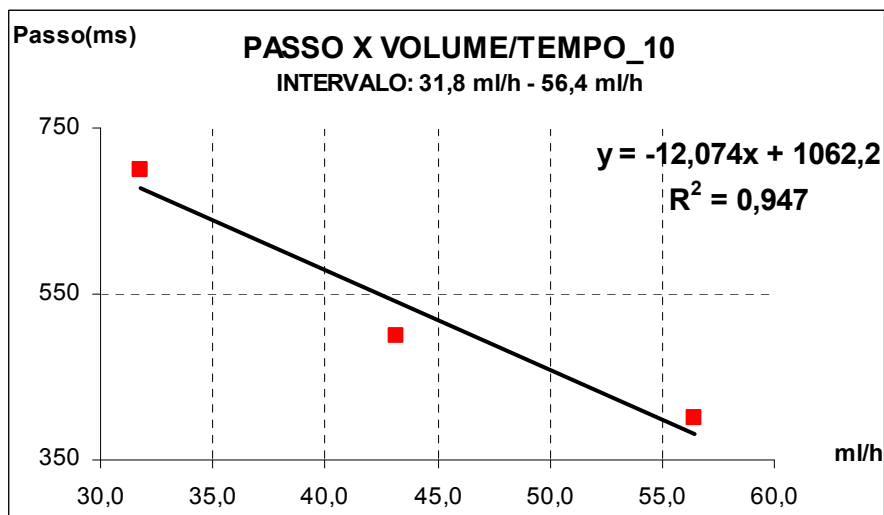
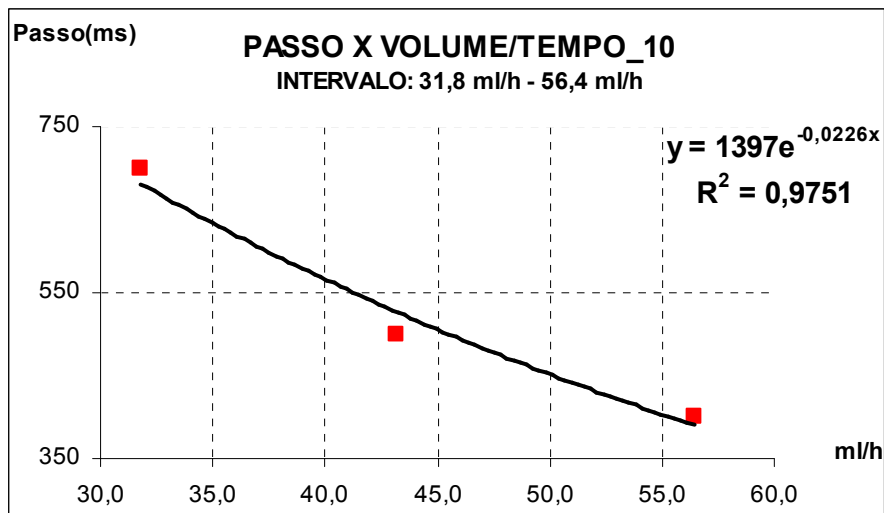
PASSO x VOLUME/TEMPO_7					
Medido		Calculado Exponencial		Calculado Linear	
		$y=8381,7e(-0,1114x)$		$y=- 266,67*x + 5400$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
2600,00	10,50	2602,18	0,08%	2599,97	0,00%
2400,00	11,25	2393,60	-0,27%	2399,96	0,00%
2200,00	12,00	2201,74	0,08%	2199,96	0,00%



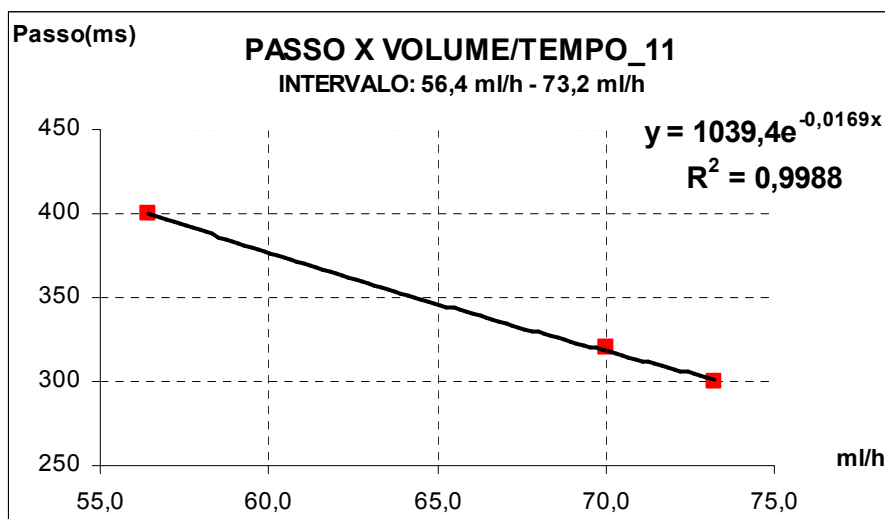
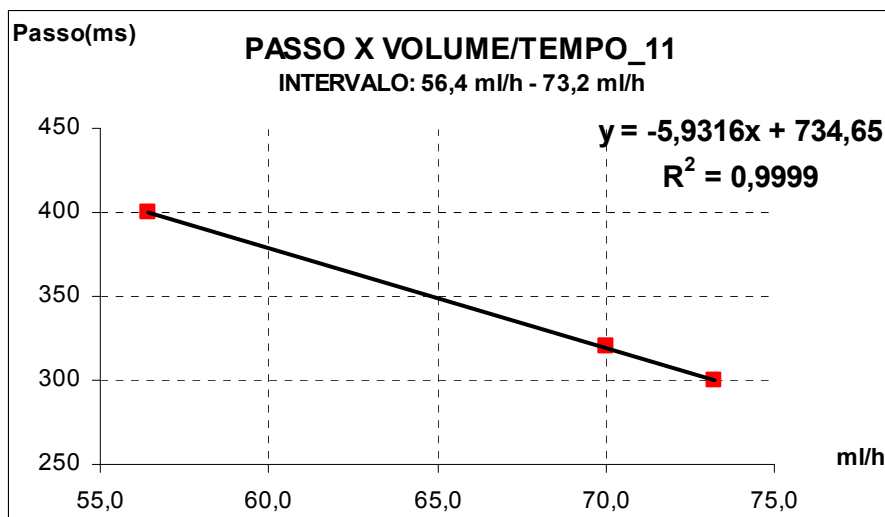
PASSO x VOLUME/TEMPO_8					
Medido		Calculado Exponencial		Calculado Linear	
		$y=6025,4e(-0,0846x)$		$y=- 140,67*x + 3856$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
2200	12,00	2183,18	-0,76%	2167,96	-1,46%
1500	16,20	1530,30	2,02%	1577,15	5,14%
1200	19,20	1187,27	-1,06%	1155,14	-3,74%



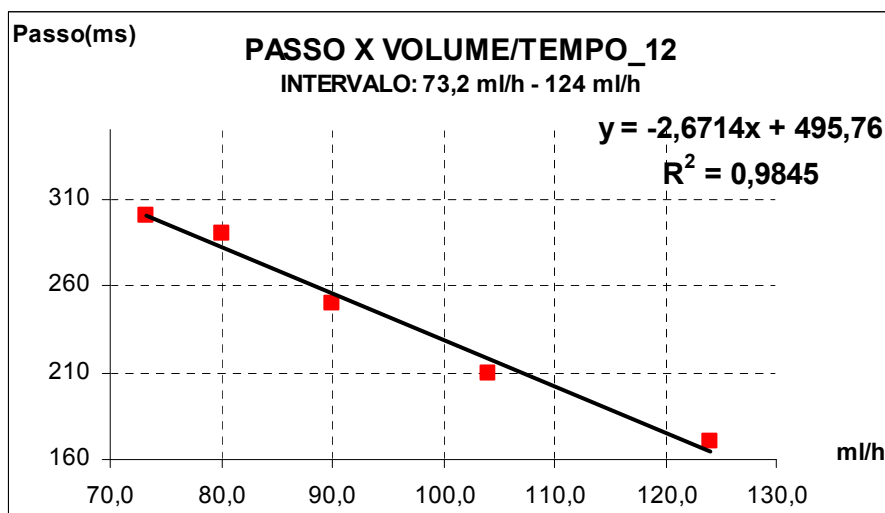
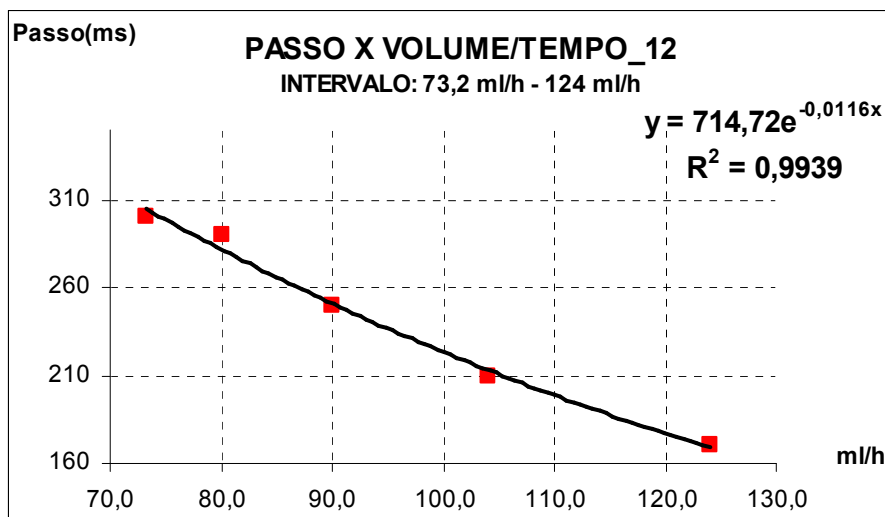
PASSO x VOLUME/TEMPO_9					
Medido		Calculado Exponencial		Calculado Linear	
		$y=2736e(-0,0426x)$		$y=- 39,79*x + 1959,9$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
1200	19,20	1207,53	0,63%	1195,93	-0,34%
900	26,40	888,56	-1,27%	909,44	1,05%
700	31,80	705,96	0,85%	694,58	-0,77%



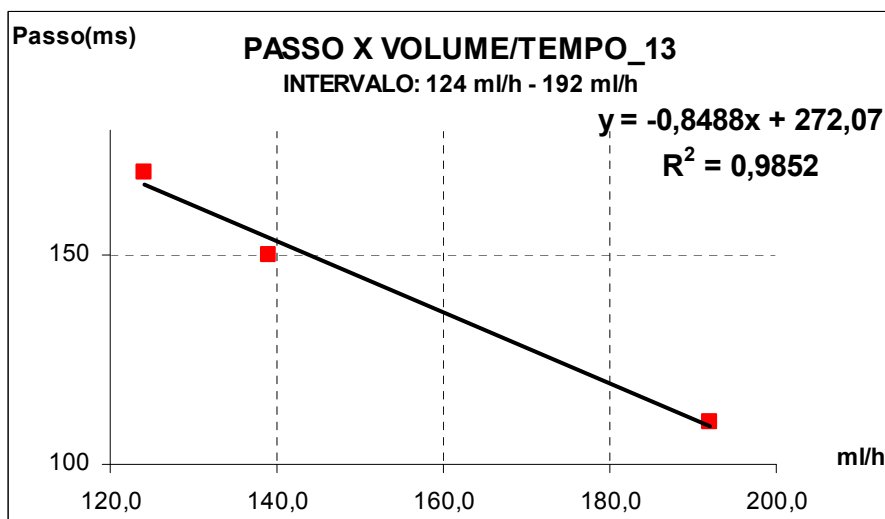
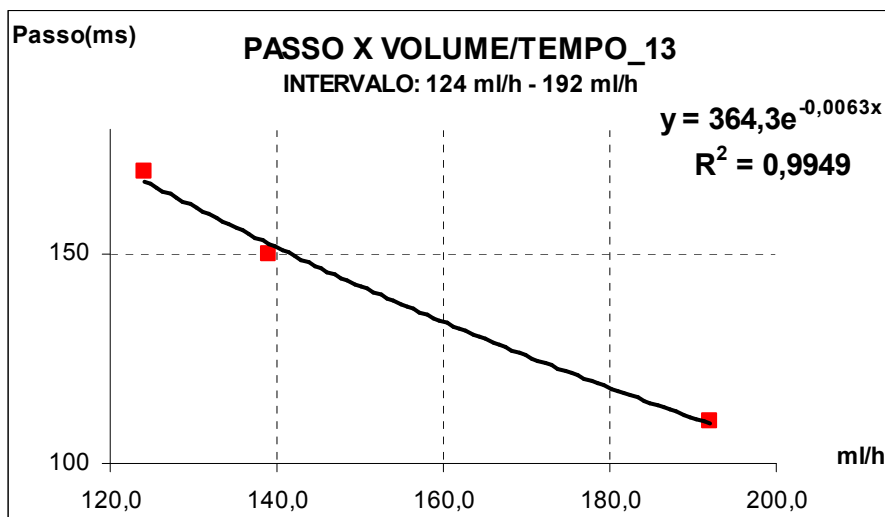
PASSO x VOLUME/TEMPO_10					
Medido		Calculado Exponencial		Calculado Linear	
		$y=1397e(-0,0226x)$		$y=- 12,074*x + 1062,2$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
700	31,80	680,89	-2,73%	678,25	-3,11%
500	43,20	526,24	5,25%	540,60	8,12%
400	56,40	390,50	-2,38%	381,23	-4,69%



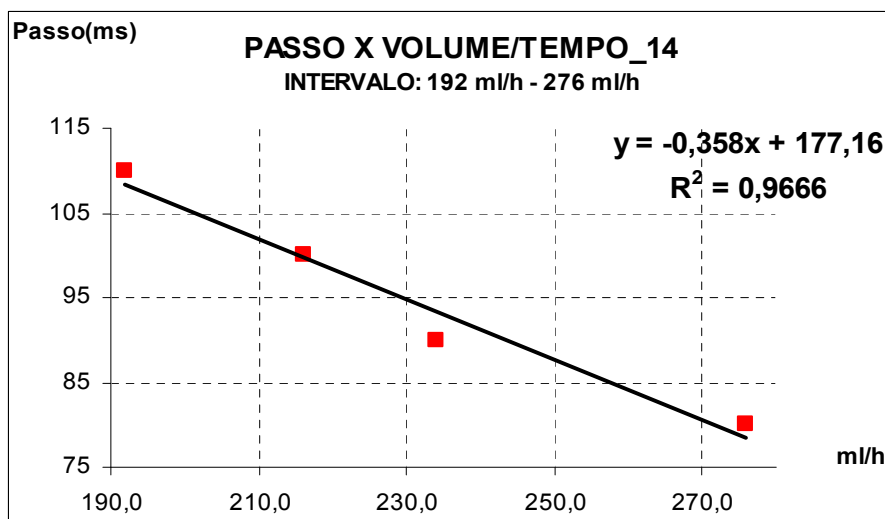
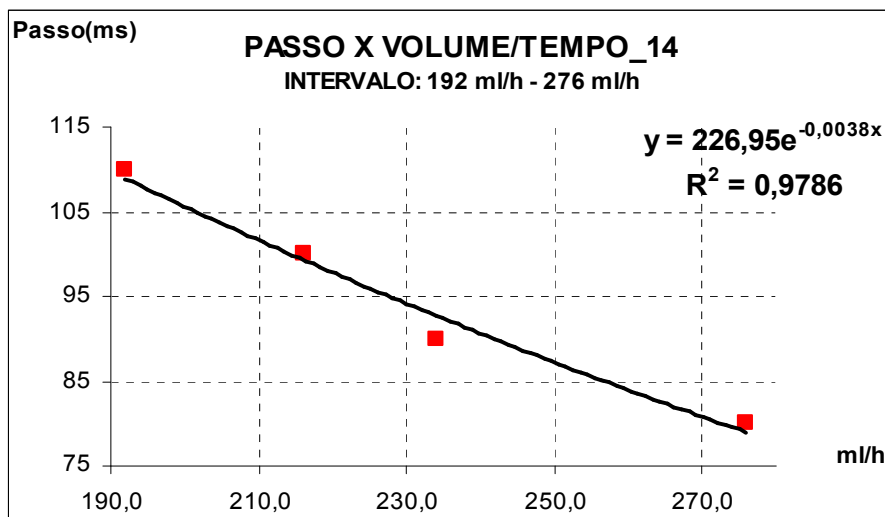
PASSO x VOLUME/TEMPO_11					
Medido		Calculado Exponencial		Calculado Linear	
		$y=1039,4e^{-0,0169x}$		$y=- 5,9316*x + 734,65$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
400	56,40	400,71	0,18%	400,11	0,03%
320	70,00	318,42	-0,49%	319,44	-0,18%
300	73,20	301,66	0,55%	300,46	0,15%



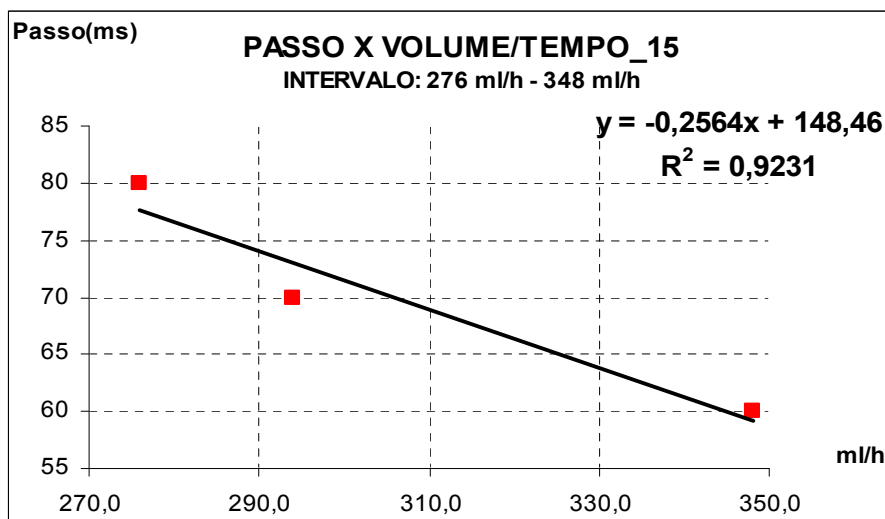
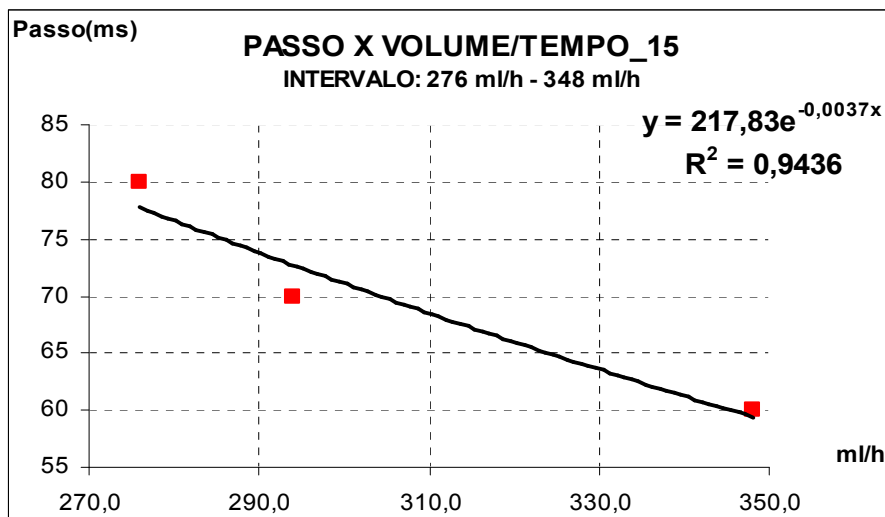
PASSO x VOLUME/TEMPO_12					
Medido		Calculado Exponencial		Calculado Linear	
		$y=714,72e(-0,0116x)$		$y=- 2,6714*x + 495,76$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
300	73,20	303,74	1,247%	300,21	0,071%
290	80,00	280,53	-3,265%	282,05	-2,742%
250	90,00	249,58	-0,167%	255,33	2,134%
210	104,00	211,90	0,906%	217,93	3,778%
170	124,00	167,72	-1,341%	164,51	-3,232%



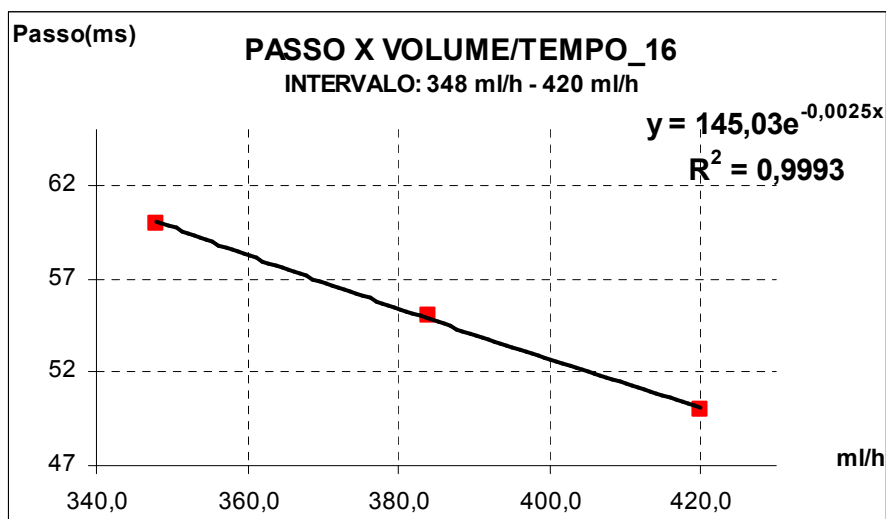
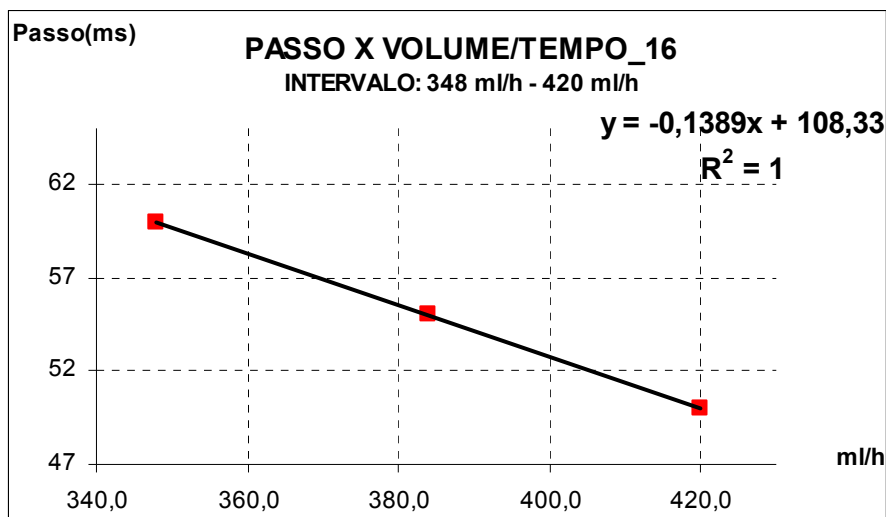
PASSO x VOLUME/TEMPO_13					
Medido		Calculado Exponencial		Calculado Linear	
		$y=364,3e(-0,0063x)$		$y=- 0,8488*x + 272,07$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
170	124,00	166,79	-1,89%	166,82	-1,87%
150	139,00	151,75	1,17%	154,09	2,72%
110	192,00	108,67	-1,21%	109,10	-0,82%



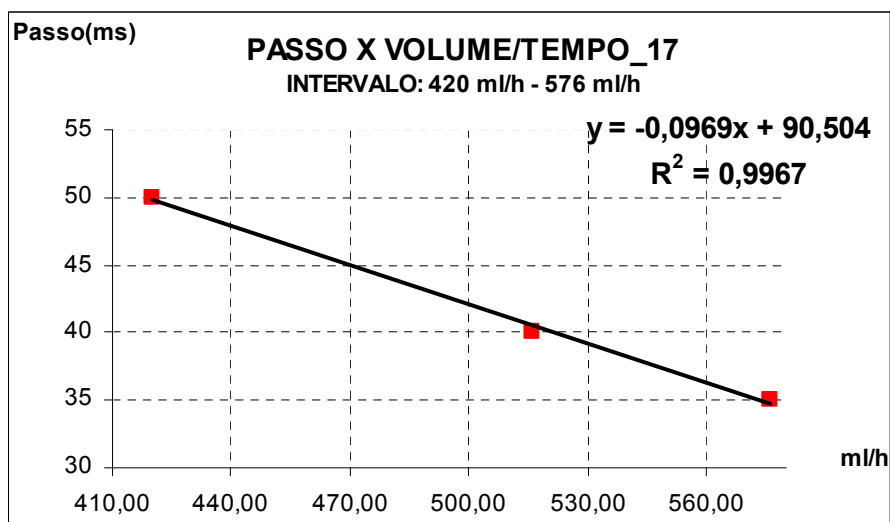
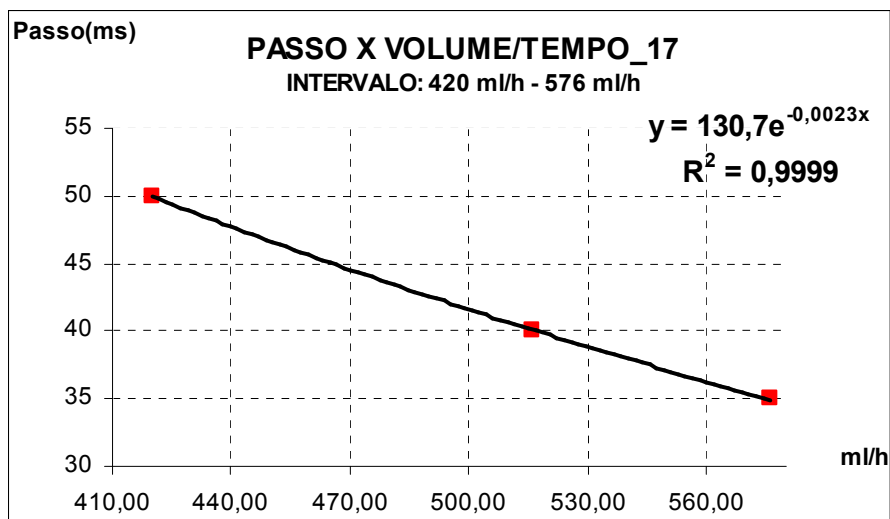
PASSO x VOLUME/TEMPO_14					
Medido		Calculado Exponencial		Calculado Linear	
		$y=226,95e^{(-0,0038x)}$		$y=- 0,358*x + 177,16$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
110	192,00	109,41	-0,54%	108,42	-1,43%
100	216,00	99,87	-0,13%	99,83	-0,17%
90	234,00	93,27	3,63%	93,39	3,76%
80	276,00	79,51	-0,61%	78,35	-2,06%



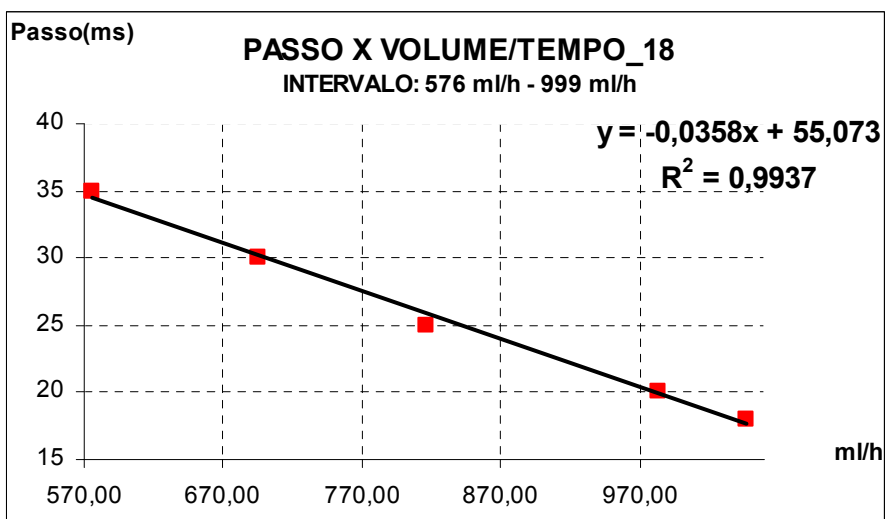
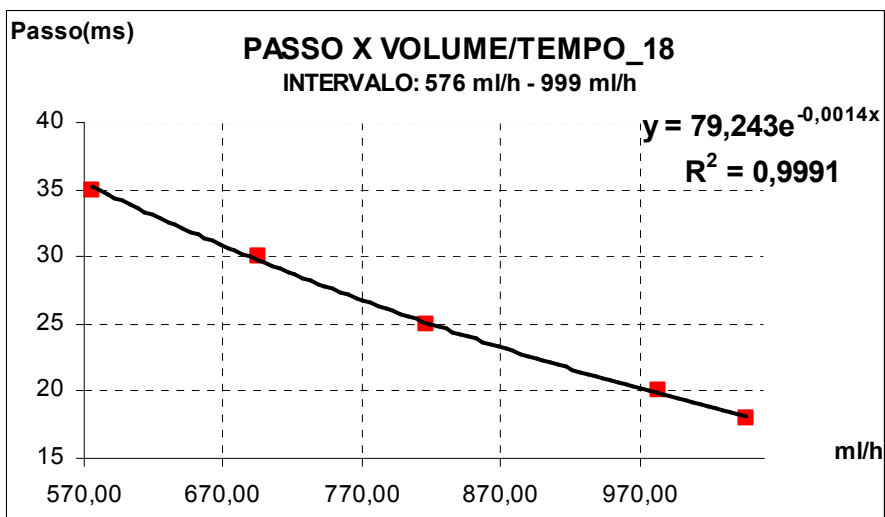
PASSO x VOLUME/TEMPO_15					
Medido		Calculado Exponencial		Calculado Linear	
		$y=217,83(-0,0037x)$		$y=- 0,2564*x + 148,46$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
80	276,00	78,45	-1,94%	77,69	-2,88%
70	294,00	73,39	4,84%	73,08	4,40%
60	348,00	60,10	0,17%	59,23	-1,28%



PASSO x VOLUME/TEMPO_16					
Medido		Calculado Exponencial		Calculado Linear	
		$y=145,03e(-0,0025x)$		$y=- 0,1389*x + 108,33$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
60	348,00	60,76	1,27%	59,68	-0,53%
55	384,00	55,53	0,96%	54,99	-0,01%
50	420,00	50,75	1,50%	49,99	-0,02%



PASSO x VOLUME/TEMPO_17					
Medido		Calculado Exponencial		Calculado Linear	
		$y=130,7e(-0,0023x)$		$y=- 0,0969*x + 90,504$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
50	420,00	49,74	-0,52%	49,81	-0,39%
40	516,00	39,88	-0,30%	40,50	1,26%
35	576,00	34,74	-0,74%	34,69	-0,89%



PASSO x VOLUME/TEMPO_18					
Medido		Calculado Exponencial		Calculado Linear	
		$y=79,243e^{(-0,0014x)}$		$y=- 0,0358*x + 55,073$	
valor_passo	ml/h	valor_passo	% erro	ml/h	% erro
35	576,00	35,37	1,06%	34,45	-1,57%
30	696,00	29,90	-0,33%	30,16	0,52%
25	816,00	25,28	1,12%	25,86	3,44%
20	984,00	19,98	-0,10%	19,85	-0,77%
18	1046,50	18,32	1,78%	17,61	-2,18%

**Tabela 3**

Valores de erro obtido com ensaio do equipo dos primeiros 60 minutos referente a figura 29.

ENSAIO_1:**Equipo_2****Inicio Ensaio: 100 ml/h****Término Ensaio: 100 ml/h**

Taxa ml/h	Tempo Medido (min)	qtd gotas (ml)	Volume Calculado (ml)	Volume Medido (ml)	% erro
100,00	0,00				
100,00	5,00		7,72	10,00	29,53
100,00	10,00	18,00	16,66	18,00	8,04
100,00	15,00	25,00	25,00	27,00	8,00
100,00	20,00	35,00	33,33	34,50	3,51
100,00	25,00	43,00	41,67	43,00	3,20
100,00	30,00	52,00	50,00	51,00	2,00
100,00	35,00	60,00	58,33	59,00	1,14
100,00	40,00	68,00	66,67	66,50	-0,25
100,00	45,00	77,00	75,00	77,00	2,67
100,00	50,00	83,00	83,33	84,40	1,28
100,00	55,00	102,00	91,67	90,00	-1,82
100,00	60,00	105,00	100,00	103,00	3,00



APÊNDICE B – DADOS E CÁLCULOS ENSAIOS PRÁTICOS



Dados e cálculos de erro percentual nos ensaios 1, 2 e 3 referente a figura 30.

ENSAIO_1:

Equipo_2

Início Ensaio: 1 ml/h

Término Ensaio: 900 ml/h

Taxa (ml/h)	Tempo Medido (min)	Volume Calculado (ml)	Volume Medido (ml)	% Erro
1	01:00:00	1	1	0,0
5	00:20:00	1,66	1,65	-0,6
10	00:15:00	2,5	2,4	-4,0
25	00:15:00	6,25	6	-4,0
50	00:10:00	8,2	8,3	1,2
100	00:05:00	8,33	8,2	-1,6
200	00:05:00	16,7	17	2,0
300	00:05:00	25	26	4,0
400	00:05:00	33,33	34	2,0
500	00:05:00	41,66	43	3,2
600	00:05:00	50	51	2,0
700	00:05:00	58,33	59	1,1
800	00:05:00	66,66	69	3,5
900	00:05:00	75	78	4,0

ENSAIO_2:

Equipo_2

Início Ensaio: 999 ml/h

Término Ensaio: 1 ml/h

Taxa (ml/h)	Tempo Medido (min)	Volume Calculado (ml)	Volume Medido (ml)	% Erro
1	01:00:00	1	1	0,0
5	00:20:00	1,66	1,6	-3,6
10	00:10:00	1,66	1,6	-3,6
25	00:10:00	4,17	4,0	-4,0
50	00:05:00	4,166	4,2	0,8
100	00:10:00	16,66	17	2,0
200	00:05:00	16,7	17,5	5,0
300	00:05:00	25	25	0,0
400	00:05:00	33,33	32,5	-2,5
500	00:05:00	41,66	41	-1,6
600	00:05:00	50	50	0,0
700	00:05:00	58,33	57	-2,3
800	00:04:00	53,333	51,5	-3,4
900	00:05:00	75	72,5	-3,3



ENSAIO_3:
 Equipo_2
 Início Ensaio: 1 ml/h
 Término Ensaio: 980 ml/h

Taxa (ml/h)	Tempo Medido (min)	Volume Calculado (ml)	Volume Medido (ml)	% Erro
1	01:00:00	1	1	0,0
2	01:00:00	2	1,9	-5,0
7	00:20:00	2,3	2,2	-4,3
20	00:10:00	3,33	3,4	2,1
40	00:05:00	3,33	3,4	2,1
90	00:05:00	7,5	7,4	-1,3
150	00:03:30	8,75	8,9	1,7
275	00:05:00	22,92	22	-4,0
350	00:05:00	29,16	29	-0,5
475	00:05:00	39,58	39	-1,5
650	00:05:00	54,16	53	-2,1
780	00:05:00	65	62	-4,6
860	00:05:00	71,66	70	-2,3
980	00:05:00	81,66	82	0,4

Tabela 4

Calculo dos valores da curva do intervalo de 19,2 ml/h à 31,8 ml/h referente a figura 42.

PASSO x VOLUME/TEMPO_9					
Medido		Calculado Exponencial		Calculado Linear	
		$y=2736e(-0,0426x)$		$y=- 39,79*x + 1959,9$	
valor_passo	ml/h	Valor_passo	%	ml/h	%
1200	19,20	1207,53	0,628%	1195,932	-0,339%
900	26,400	888,56	-1,271%	909,444	1,049%
700	31,800	705,96	0,851%	694,578	-0,775%



APÊNDICE C – CÓDIGO FONTE DO SOFTWARE



```
#include <at89x52.h>
#include <paulmon2.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <malloc.h>

#define on 1
#define off 0
#define strobe2          0x0002          // Habilitação do LCD
#define strobe1          0x0001          // Habilitação do TECLADO
#define buzzer          P3_3              // Definição do bit para acionar o buzzer.
#define linhacoluna4 0x07 // 0111 Define os valores a serem escritos nas linhas...para ler as colunas
#define linhacoluna3 0x0b // 1011
#define linhacoluna2 0x0d // 1101
#define linhacoluna1 0x0e // 1110

typedef unsigned char BYTE;

volatile xdata at strobe2 unsigned char data_lcd;
volatile xdata at strobe1 unsigned char data_teclado;

//*****DECLARAÇÃO DOS TEXTOS*****//
code char confirma_taxa[] = "CONFIRME TAXA";
code char e_c[] = " E/C";
code char e[] = " PRES. E";
code char digite_taxa_de[] = "DIGITE TAXA DE";
code char infusao[] = "INFUSAO = ";
code char taxa5[] = "TAXA < 1";
code char precione_tecla_c[] = "PRECIONE TECLA C";
code char mlh[] = " ml/h";
code char taxa_de_infusao[] = "TAXA DE INFUSAO =";
code char a_altera [] = "A = ALTERAR TAXA";
code char c_clear [] = "C = LIMPAR TAXA";
code char digite_volume[] = "DIG. VOL. TOTAL";
code char total_a_infundir[] = "INFUNDIR = ";
code char confirme_volume[] = "CONFIRME VOLUME";
code char ml[] = " ml";
code char volume_inferior_taxa[] = "VOLUME < TAXA";
code char infusao_completa_kvo[] = "INFUSAO COMPLETA";
code char infundindo_kvo[] = "INFUSAO EM KVO";
code char tempo_a_infundir[] = "TEMPO P/ TERMINO";
code char da_infusao[] = "DA INFUSAO = ";
code char minutos[] = " min";
code char taxa_baixa_volume[] = "TAXA BAIXA VOL.";
int valor_total [5]; // Vetor de 3 posições

//*****DECLARAÇÃO DAS FUNÇÕES*****//
void inic_lcd (void);
void clr_lcd (void);
void desloc_dir (void);
void desloc_esq (void);
void seg_linha (void);
void imp_lcd (char *teste);
void inat_cursor (void);
void escreve_lcd3num(int val);
void escreve_lcdnum(int val);
void retorno_cursor (void);
void espera_tecla(char z);
char le_teclado(void);
char le_teclado_tecla_F(void);
int val_lido(void);
void delayms(char c);
void escreve_motor(void);
void digite_taxa_infusao(void);
void digite_volume_a_infundir(void);
void print_ml(void);
void printml(void);
void zera_variaveis(void);
void infusao_completa(void);
void tempo_infundir(void);
```



```
void contagem_ml (void ) interrupt 3;// 3 é o número para interpretação do compilador para TIMER1void
delay_100micros( unsigned int tempo);

/*****DECLARAÇÃO DAS FUNÇÕES GLOBAIS*****/
float valor_total_somatorio = 0; // Variavel onde sera armazenado o valor total do somatorio
da taxa de infusão
float valor_total_somatorio_volume = 0; // Variavel onde sera armazenado o valor total do
somatorio do volume
int valor_lido_teclado = 0; // Variavel onde sera armazenado o valor lido do teclado
int i = 0; // Controle para i indice do vetor
int tecla_F_precionada = 0; // Controle para tela F precionada
int ml_parcial = -3; // Variavel contagem das gotas, contado via int1

unsigned int tempo_passo_motor = 0;
char atualiza_display;
int tela = 0;
unsigned int tempo_infusao = 0;

int num = 0;
int mil = 0;
int cent = 0;
int dez = 0;
int uni = 0;

int teclaa = 1;
void main (void)
{
    int tecla_precionada = 1; // Variavel de controle que verifica se a tecla esta
pressionada
/****REGISTRADOR IE****/
    IE = 0; // Registrador IE em 0 , desabilita todas as interrupções
    ET1 = 1;//Interrupção do TIMER1, ET1=1, Interrupção habilitada
    EA = 1; //Interrupções habilitadas
    TH1 = 0xFF;
    TL1 = 0xED;

/****REGISTRADOR TMOD****/
    TMOD = 0x50; //0101 0000, TIMER1
    TR1 = 1;

    inic_lcd(); // Inicializa os parametros do LCD
    digite_taxa_infusao();// Imprime o cabeçario
    buzzer = 1;

    for(;;)
    {

/*****LÊ O TECLADO E IMPRIME NO DISPLAY*****/
        do
        {

            valor_lido_teclado = le_teclado(); // Lê o teclado

            if (valor_lido_teclado == 0 || valor_lido_teclado > 64) // Verifico se o valor lido é 0
(nenhuma tecla precionada) //
            {
                ou maior que 64 (que são as letras) //
                switch(valor_lido_teclado) // Verifica qual tecla foi
                {
                    case 0: // Caso nenhuma tecla
                    precionada retorna sem fazer nada
                        tecla_precionada=0;
                        break;
                    case 65:
                        goto CLEAR;
                        break;
                    case 67: // Caso tecla C precionada CLEAR
                        CLEAR: // Retono ao Label CLEAR
                        if(tecla_precionada==1) // verifica se a tecla esta mantida precionada,
                        imprimindo apenas 1 vez o valor
                }
            }
        }
    }
}
```



```

        }
        } else // Caso a tecla C
tenha sido precionada em qualquer parte do programa
        {
            clr_lcd();
            digite_taxa_infusao();
            tecla_precionada = 1;
            break;
        }
    case 69: // Caso teclaa E
precionada ENTER
        ENTER: // Retorno ao
Label ENTER, // Retorno ao
        if(tecla_precionada==1) // Verifica se a tecla
esta mantida precionada, imprimindo apenas 1 vez o valor
        {
            }
        else
        {
            valor_total_somatorio = valor_total[0] * 100 + valor_total[1] *
10 + valor_total[2] * 1; // Soma o valor das teclas precionadas, juntando em um unico valor a centena + dezena +
unidade
            if(valor_total_somatorio < 1) // Verifica se
o valor_total_somatorio é menor que 5, responsavel pela taxa minima de infusão
            {
                // Se for menor envia para display a mensagem
                retorno_cursor (); //
Este valor de 5 pode ser alterado para ajuste
                clr_lcd();
                imp_lcd(taxa5);
                seg_linha();
                imp_lcd(precione_tecla_c);
                valor_total_somatorio = 0;
                do
                // aguarda até que seja apertado a tecla C
                {
                    valor_lido_teclado = le_teclado();
                }while (valor_lido_teclado != 67);
            }
        else
        // caso o valor_total_somatorio seja maior que 5
        {
            retorno_cursor (); //
imprime no display a taxa selecionada
            clr_lcd();
            imp_lcd(confirma_taxa);
            seg_linha();
            escreve_lcdnum (valor_total[0]);
            escreve_lcdnum (valor_total[1]);
            escreve_lcdnum (valor_total[2]);
            imp_lcd(mlh);
            imp_lcd(e_c);
        }
    do // Aguarda até que seja
precionada a tecla E ou C
    {
        valor_lido_teclado = le_teclado();
        // Le o teclado
    }while((valor_lido_teclado != 67) && (valor_lido_teclado !=
69)); // Loop aguardando uma das teclas E ou C serem precionadas
        i=0;
        // pois são as duas possibilidades fornecidas ao usuario
        tecla_precionada = 0;
// Zera o indice do vetor

```



```

                                if (valor_lido_teclado == 67)
                                // Verifica qual tecla foi precionada anteriormente
                                {
tecla C precionada
                                }
                                valor_total_somatorio = 0;
                                goto CLEAR;
                                // Caso tecla C salta para o label CLEAR
                                }
                                }

// insira a logica soliitando o volume total.

                                break;
                                }
                                case 70:
                                // caso tecla F precionada FIM
                                if(tecla_precionada==1) // verifica se a tecla esta mantida
                                {
                                }
                                else
                                // Caso a tecla F tenha
                                // sido precionada entra nesta rotina
                                {
                                tecla_precionada = 1;
                                break;
                                }
                                }
                                else
                                // Se alguma tecla referente a numeros de 0 a 9 foi precionada
                                {
                                valor_lido_teclado = valor_lido_teclado - 48; // Diminue do valor_lido_teclado
                                // 48 em decimal e obtem o valor em decimal referente a tecla precionada em decimal
                                if(tecla_precionada==1) // verifica se a tecla
                                // esta mantida precionada, imprimindo apenas 1 vez o valor
                                {
                                }
                                else
                                {
                                escreve_lcdnum (valor_lido_teclado); // Escreve no display o
                                // valor da tecla precionada
                                valor_total [i] = valor_lido_teclado ; // Salva o valor lido da
                                // tecla no vetor que começa em 0 até 2
                                i++;
                                tecla_precionada = 1; // Seta
                                // em a variavel "tecla_precionada" para controle se a tecla esta mantida precio
                                }
                                }
                                if(i==3) //
                                // Entra nesta função se já foram adicionadas 3 valores no teclado.
                                {
                                do
                                {
                                teclado
                                // valor_lido_teclado = le_teclado(); // Le o
                                // uma das teclas E ou C serem precionadas
                                }while((valor_lido_teclado != 67) && (valor_lido_teclado != 69)); // Loop aguardando
                                // pois são as duas possibilidades fornecidas ao usuario //
                                //i=3; //
                                tecla_precionada = 0;

                                // Após digitado a taxa de infusão....

                                if (valor_lido_teclado == 67) // Verifica
                                // qual tecla foi apertada
                                {
                                i=0;
                                goto CLEAR; //
                                // Caso tecla C salta para o label CLEAR
                                }
                                else

```



```

    {
        goto ENTER; //
Caso tecla E salta para o label ENTER

    }
    }else{}
}while (valor_total_somatorio < 1);

i=3;
retorno_cursor (); // Este valor de 5 pode ser alterado para ajuste
clr_lcd();
imp_lcd(digite_volume);
seg_linha();
imp_lcd(total_a_infundir);

//////////LEITURA DO VOLUME TOTAL A INFUNDIR.

do
{
    valor_lido_teclado = le_teclado(); // Lê o teclado

    if (valor_lido_teclado == 0 || valor_lido_teclado > 64) // Verifico se o valor lido é 0
(nenhuma tecla precionada)
    { //
ou maior que 64 (que são as letras)
    switch(valor_lido_teclado) // Verifica qual tecla foi
precionada
    {
        case 0: // Caso nenhuma tecla
precionada retorna sem fazer nada
            tecla_precionada=0;
            break;
        case 65:
            goto CLEAR_1;
            break;
        case 67: // Caso tecla C precionada CLEAR
            CLEAR_1: // Retono ao Label
CLEAR
            if(tecla_precionada==1) // verifica se a tecla esta mantida precionada,
imprimindo apenas 1 vez o valor
            {
            }
            else // Caso a tecla C
tenha sido precionada em qualquer parte do programa
            {
                clr_lcd();

                digite_volume_a_infundir();
                tecla_precionada = 1;
                break;
            }
        case 69: // Caso teclaa E
precionada ENTER
            ENTER_1: // Retorno ao
Label ENTER,
            if(tecla_precionada==1) // Verifica se a tecla
esta mantida precionada, imprimindo apenas 1 vez o valor
            {
            }
            else
            {

                valor_total_somatorio_volume = valor_total[3] * 100 +
valor_total[4] * 10 + valor_total[5] * 1; // Soma o valor das teclas precionadas, juntando em um unico valor a
centena + dezena + unidade

                if(valor_total_somatorio >= valor_total_somatorio_volume)
// Verifica se o valor_total_somatorio é menor que 5, responsavel pela taxa minima de infusão

```



```
        {
// Se for menor envia para display a mensagem
returno_cursor ();
//
Este valor de 5 pode ser alterado para ajuste
clr_lcd();
imp_lcd(volume_inferior_taxa);
seg_linha();
imp_lcd(precione_tecla_c);
valor_total_somatorio_volume = 0;
do
// aguarda até que seja apertado a tecla C
{
valor_lido_teclado = le_teclado();
}while (valor_lido_teclado != 67);
}
else
// caso o valor_total_somatorio seja maior que 5
{
returno_cursor ();
//
imprime no display a taxa selecionada
clr_lcd();
imp_lcd(confirme_volume);
seg_linha();
escreve_lcdnum (valor_total[3]);
escreve_lcdnum (valor_total[4]);
escreve_lcdnum (valor_total[5]);
imp_lcd(ml);
imp_lcd(e_c);
do
// Aguarda até que seja
precionada a tecla E ou C
{
valor_lido_teclado = le_teclado();
// Le o teclado
}while((valor_lido_teclado != 67) && (valor_lido_teclado !=
69)); // Loop aguardando uma das teclas E ou C serem precionadas
i=3;
// pois são as duas possibilidades fornecidas ao usuario
tecla_precionada = 0;
// Zera o indice do vetor
if (valor_lido_teclado == 67)
// Verifica qual tecla foi precionada anteriormente
{
// Caso
tecla C precionada
valor_total_somatorio_volume = 0;
valor_total[3] = 0;
valor_total[4] = 0;
valor_total[5] = 0;
goto CLEAR_1;
// Caso tecla C salta para o label CLEAR
}
break;
}
case 70:
// caso tecla F precionada FIM
if(tecla_precionada==1) // verifica se a tecla esta mantida
{
}
else
// Caso a tecla F tenha
// Caso a tecla F tenha
tecla_precionada = 1;
break;
}
}
}
}
```



```
else // Se alguma tecla referente a numeros de 0 a 9 foi precionada
{
    valor_lido_teclado = valor_lido_teclado - 48; // Diminue do valor_lido_teclado
48 em decimal e obtem o valor em decimal referente a tecla precionada em decimal
    if(tecla_precionada==1) // verifica se a tecla
esta mantida precionada, imprimindo apenas 1 vez o valor
    {
    }
    else
    {
        escreve_lcdnum (valor_lido_teclado); // Escreve no display o
valor da tecla precionada
        valor_total [i] = valor_lido_teclado ; // Salva o valor lido da
tecla no vetor que começa em 0 até 2
        i++;
        tecla_precionada = 1; // Seta
em a variavel "tecla_precionada" para controle se a tecla esta mantida precio
    }
}
if(i==6) //
Entra nesta função se já foram adicionadas 3 valores no teclado.
{
    do
    {
        valor_lido_teclado = le_teclado(); // Le o
teclado
    }while((valor_lido_teclado != 67) && (valor_lido_teclado != 69)); // Loop aguardando
uma das teclas E ou C serem precionadas //
pois são as duas possibilidades fornecidas ao usuario
    tecla_precionada = 0;

// Após digitado a taxa de infusão....

    if (valor_lido_teclado == 67) // Verifica
qual tecla foi apertada
    {
        i=3;
        goto CLEAR_1; //
Caso tecla C salta para o label CLEAR
    }
    else
    {
        goto ENTER_1; //
Caso tecla E salta para o label ENTER
    }
}
}else{}
// verefica a taxa o tempo para o total da infusão
tempo_infusao = ((valor_total_somatorio_volume /valor_total_somatorio) * 60);
if(tempo_infusao > 999)
// caso tempo maior que 999
{
// exhibe as mensagens de TAXA BAIXA PARA O VOLUME
clr_lcd();
imp_lcd(taxa_baixa_volume);
seg_linha();
imp_lcd(precione_tecla_c);

do
// aguarda até que seja precionado a tecla E
{
    valor_lido_teclado = le_teclado(); // Le
o teclado //
}while(valor_lido_teclado != 67); //
Loop aguardando a tecla E ser precionada
tecla_precionada = 0;
i=0;
valor_total_somatorio_volume = 0;
valor_total_somatorio = 0;
goto CLEAR;
```



```
    }
    }while (valor_total_somatorio_volume <= valor_total_somatorio);
// informação para tempo decorrido.....

tempo_infundir();
do
{
    valor_lido_teclado = le_teclado(); // Le o teclado
}while(valor_lido_teclado != 69); // Loop aguardando uma das teclas E ou C serem precionadas

//tempo_passo_motor = (int)(valor_total_somatorio) ;

//*****FUNÇÕES PARA CADA CÁLCULO PARA CADA INTERVALO DE ml/h*****//
if(valor_total_somatorio >=1.0 && valor_total_somatorio < 2.0)
{
    valor_total_somatorio = (0.7132 * valor_total_somatorio);
    tempo_passo_motor = (int)(65780 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=2.00 && valor_total_somatorio < 3.45)
{
    valor_total_somatorio = (0.3886 * valor_total_somatorio);
    tempo_passo_motor = (int)(35038 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=3.45 && valor_total_somatorio < 4.80)
{
    valor_total_somatorio = (0.267 * valor_total_somatorio);
    tempo_passo_motor = (int)(22892 *expf(-valor_total_somatorio));
}

if(valor_total_somatorio >=4.80 && valor_total_somatorio < 7.00)
{
    valor_total_somatorio = (0.2116 * valor_total_somatorio);
    tempo_passo_motor = (int)(17227 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=7.0 && valor_total_somatorio < 8.40)
{
    valor_total_somatorio = (0.156 * valor_total_somatorio);
    tempo_passo_motor = (int)(11793 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=8.40 && valor_total_somatorio < 10.50)
{
    valor_total_somatorio = (0.0989 * valor_total_somatorio);
    tempo_passo_motor = (int)(7342.9 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=10.50 && valor_total_somatorio < 12)
{
    tempo_passo_motor = ( 5400 - (266.67 * valor_total_somatorio));
    //valor_total_somatorio = (0.1114 * valor_total_somatorio);
    //tempo_passo_motor = (int)(8381.7 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=12 && valor_total_somatorio < 19.2)
{
    valor_total_somatorio = (0.0846 * valor_total_somatorio);
    tempo_passo_motor = (int)(6025.4 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=19.2 && valor_total_somatorio < 31.8)
{
    tempo_passo_motor = ( 1959.4 - (39.79 * valor_total_somatorio));
    //valor_total_somatorio = (0.0196 * valor_total_somatorio);
    //tempo_passo_motor = (int)(1237 *expf(-valor_total_somatorio));
}

if(valor_total_somatorio >=31.8 && valor_total_somatorio < 56.4)
{
    valor_total_somatorio = (0.0226 * valor_total_somatorio);
    tempo_passo_motor = (int)(1397 *expf(-valor_total_somatorio));
}
if(valor_total_somatorio >=56.4 && valor_total_somatorio < 73.2)
```




```
        {
            tempo_passo_motor = ( 734.65 - (5.9316 * valor_total_somatorio));
            //valor_total_somatorio = (0.0076 * valor_total_somatorio);
            //tempo_passo_motor = (int)(433.19 *expf(-valor_total_somatorio));
        }
    if(valor_total_somatorio >=73.2 && valor_total_somatorio < 124)
        {
            valor_total_somatorio = (0.0116 * valor_total_somatorio);
            tempo_passo_motor = (int)(714.72 *expf(-valor_total_somatorio));
        }
    if(valor_total_somatorio >=124 && valor_total_somatorio < 192)
        {
            valor_total_somatorio = (0.0063 * valor_total_somatorio);
            tempo_passo_motor = (int)(364.3 *expf(-valor_total_somatorio));
        }
    if(valor_total_somatorio >=192 && valor_total_somatorio < 276)
        {
            tempo_passo_motor = ( 177.16 - (0.358 * valor_total_somatorio));
            //valor_total_somatorio = (0.0023 * valor_total_somatorio);
            //tempo_passo_motor = (int)(118.47 *expf(-valor_total_somatorio));
        }
    if(valor_total_somatorio >=276 && valor_total_somatorio < 348)
        {
            tempo_passo_motor = ( 148.46 - (0.2564 * valor_total_somatorio));
            //valor_total_somatorio = (0.0037 * valor_total_somatorio);
            //tempo_passo_motor = (int)(194.72 *expf(-valor_total_somatorio));
        }
    if(valor_total_somatorio >=348 && valor_total_somatorio < 420)
        {
            valor_total_somatorio = (0.0025 * valor_total_somatorio);
            tempo_passo_motor = (int)(145.03 *expf(-valor_total_somatorio));
        }
    if(valor_total_somatorio >=420 && valor_total_somatorio < 576)
        {
            valor_total_somatorio = (0.0023 * valor_total_somatorio);
            tempo_passo_motor = (int)(130.7 *expf(-valor_total_somatorio));
        }
    if(valor_total_somatorio >=576 && valor_total_somatorio < 984)
        {
            valor_total_somatorio = (0.0014 * valor_total_somatorio);
            tempo_passo_motor = (int)(77.904 *expf(-valor_total_somatorio));
        }
    escreve_motor(); // Função de escrita no motor
}

}
//*****DECLARAÇÃO DAS ROTINAS*****//
//*****INTERRUPÇÃO , INTERRUPÇÃO VIA INT1*****//

void contagem_ml (void) interrupt 3 //toda vez que o INT1 for a 0 vai entrar nesta função.
{
    // que conta a quantidade de gotas passadas pelo emblo.
    TR1 = off;
    EA = off;
    ml_parcial++;
    atualiza_display = 1;
    TH1 = 0xFF;
    TL1 = 0xED;
    TR1 = on;
    EA = on; // ml váriavel que armazena a contagem.
}

//***** IMPRIME NO DISPLAY O VOLUME*****//
void print_ml(void) //imprime no display o volume decorrido da infusão
{
    imp_lcd("Taxa = ");
    escreve_lcdnum (valor_total[0]);
    escreve_lcdnum (valor_total[1]);
    escreve_lcdnum (valor_total[2]);
    imp_lcd (mlh);
    seg_linha ();
}
```



```
    imp_lcd("Volume = ");
    printml();
    imp_lcd(" ml ");
    retorno_cursor ();
}
//*****IMPRIME A VARIÁVEL ML*****//
void printml(void)          //fraciona a variável ml e imprime a
{
    num = ml_parcial;

    cent = num / 100;
    num = num % 100;
    dez = num / 10;
    uni = num % 10;
    escreve_lcdnum (cent);
    escreve_lcdnum (dez);
    escreve_lcdnum (uni);
}
//*****DELAY*****//
void delayms(char c)      // Rotina de Delay com tempo de mínimo
de 1ms
{
    c;
    _asm
    0100$:   mov r0,dpl
            mov R2,#8
    0200$:   mov R1,#55
    0300$:   djnz R1,0300$
            djnz R2,0200$
            djnz R0,0100$
            ret
    _endasm;
}
//*****LÊ O TECLADO*****//
char le_teclado(void)
{
    unsigned char val_lido;
    char tecla=0;
    data_teclado = linhacoluna1;
    val_lido = data_teclado;
    val_lido = val_lido & 0x0f; //val_lido receberá o valor 1111//mascarando val_lido que é valor lido
    switch(val_lido)          //Testa todos os casos, se for verdadeiro executa
    {
        case linhacoluna1:   //caso linhacoluna1
            return('0');
            break;           //Permite sair ( quebrar)
        case linhacoluna2:
            return('1');
            break;
        case linhacoluna3:
            return('2');
            break;
        case linhacoluna4:
            return('3');
            break;
    }
    data_teclado = linhacoluna2;
    val_lido = data_teclado;
    val_lido = val_lido & 0x0f;
    switch(val_lido)
    {
        case linhacoluna1:
            return('4');
            break;
        case linhacoluna2:
            return('5');
            break;
    }
}
```



```
        case linhacoluna3:
            return('6');
            break;
        case linhacoluna4:
            return ('7');
            break;
    }
    data_teclado = linhacoluna3;
    val_lido = data_teclado;
    val_lido = val_lido & 0x0f;
    switch(val_lido)
    {
        case linhacoluna1:
            return ('8');
            break;
        case linhacoluna2:
            return('9');
            break;
        case linhacoluna3:
            return('A');
            break;
        case linhacoluna4:
            return('B');
    }
    data_teclado = linhacoluna4;
    val_lido = data_teclado;
    val_lido = val_lido & 0x0f;
    switch(val_lido)
    {
        case linhacoluna1:
            return('C');
            break;
        case linhacoluna2:
            return('D');
            break;
        case linhacoluna3:
            return('E');
            break;
        case linhacoluna4:
            return('F');
            break;
    }
    return (tecla);
}
//***** LE APENAS A TECLA F *****/
char le_teclado_tecla_F(void)
{
    unsigned char val_lido;
    char tecla=0;
    data_teclado = linhacoluna4;
    val_lido = data_teclado;
    val_lido = val_lido & 0x0f;
    switch(val_lido)
    {
        case linhacoluna4:
            return('F');
            break;
    }
    return (tecla);
}
//*****INICIALIZAÇÃO DO DISPLAY*****/
void inic_lcd (void)
{
    BYTE c;
    for(c=0; c<2; c++)
    {
        P1_4 = off;
        data_lcd = 0x38;
        delayms(5);
        P1_4 = on;
    }
}
```



```
    P1_4 = off;
    data_lcd = 0x0E;
    delayms(5);
    P1_4 = on;
    P1_4 = off;
    data_lcd = 0x01;
    delayms(5);
    P1_4 = on;
    P1_4 = off;
    data_lcd = 0x0C;
    delayms(5);
    P1_4 = on;
}
//*****LIMPA DISPLAY*****//
void clr_lcd (void)
{
    P1_4 = off;                //Desabilita pino RS do display LCD
    data_lcd = 0x01; //Display recebe instrução e escreve na DD RAM
    delayms(5); //Delay de 10 ms para ajustar a comunicação entre CPU e display
    P1_4 = on;                //Habilita pino RS do display
}
//*****DESLOCA O CURSOR PARA DIREITA*****//
void desloc_dir (void)
{
    P1_4 = off;
    data_lcd = 0x14;
    delayms(5);
    P1_4 = on;
}
//*****DESLOCA O CURSOR PARA ESQUERDA*****//
void desloc_esq (void)
{
    P1_4 = off;
    data_lcd = 0x13;
    delayms(5);
    P1_4 = on;
}
//*****RETORNA O CURSOR PARA A POSIÇÃO 0,0 (LINHA 0 COLUNAO*****//
void retorno_cursor (void)
{
    P1_4 = off;
    data_lcd = 0x02;
    seg_linha();
    delayms(5);
    P1_4 = on;
}
//*****DESLOCA A LINHA*****//
void seg_linha (void)
{
    P1_4 = off;
    data_lcd = 0xC0;
    delayms(5);
    P1_4 = on;
}

void inat_cursor (void)
{
    P1_4 = off;
    data_lcd = 0x0C;
    delayms(5);
    P1_4 = on;
}
//*****ESCREVE CARACTERES*****//
void imp_lcd (char *teste)
{
    BYTE val;
    val = *teste;
    P1_4 = on;

    while (val != 0)
    {
```



```
        P1_4 = on;
        data_lcd = val;
        delayms(5);
        P1_4 = off;
        teste ++;
        val = *teste;
    }
    P1_4 = on;
}
//*****ESCREVE NUMEROS*****//
void escreve_lcdnum(int val)
{
    if(val<0 || val>9)
    {
        return;                //caso valor fora de 0 a 9 retorna para main
    }
    else                        // caso valor entre 0 a 9
    {
        data_lcd= val + 0x30;    // soma val com 30h
        delayms(1);            // Necessário delay para instabilidade do numero quando escreve.
    }
}

//*****ROTINA DE 100u SEGUNDOS*****//
void delay_100micros(unsigned int tempo)
{
    unsigned int c;
    for (c=0 ; c<tempo; c++)
    {
        if(le_teclado_tecla_F() > 10)
        {
            tecla_F_precionada = 70;
        }
        _asm
        mov r7,#12
100$:
        nop
        djnz r7,100$
        _endasm;
    }
}

//*****SETA O MOTOR DE PASSO*****//
void escreve_motor (void)
{
    clr_lcd();
    print_ml();

    do
    {
        P1=0x0C;                //1100B
        delay_100micros(tempo_passo_motor);

        P1=0x04;                //0100B
        delay_100micros(tempo_passo_motor);

        P1=0x06;                //0110b
        delay_100micros(tempo_passo_motor);

        P1=0x02;                //0010B
        delay_100micros(tempo_passo_motor);

        P1=0x03;                //0011b
        delay_100micros(tempo_passo_motor);

        P1=0x01;                //0001B
    }
}
```



```
    delay_100micros(tempo_passo_motor);

    P1=0x09;          //1001B
    delay_100micros(tempo_passo_motor);

    P1=0x08;          // 1000b
    delay_100micros(tempo_passo_motor);

    if(atualiza_display == 1)
    {
        print_ml();
        atualiza_display = 0;
    }

    if(ml_parcial == valor_total_somatorio_volume)
    {
        if (tela == 0)
        {
            infusao_completa();
            buzzer = 0;          // caso esteja em KVO o uzzzer comoeca a acionar.
            tela = 1;
        }
        tempo_passo_motor = 83985;
    }else{}

}while(tecla_F_precionada != 70);

buzzer = 1;
clr_lcd();
imp_lcd(a_altera);
seg_linha();
imp_lcd(c_clear);

valor_total_somatorio = 0;
tecla_F_precionada = 0;

do
{
    valor_lido_teclado = le_teclado();
}while((valor_lido_teclado != 65) && (valor_lido_teclado != 67));
if (valor_lido_teclado == 65)          // Verifica qual tecla
foi apertada
    {
        teclaa = 1;
        zera_variaveis();
        // Volta para o programa inicial...
    }
    else
    {
        teclaa = 0;
        ml_parcial = 0;
        zera_variaveis();
    }
}

//*****IMPRIME CABEÇARIO *****//
void digite_taxa_infusao(void)      // imprime cabeçario
{
    imp_lcd (digite_taxa_de);
    seg_linha();
    imp_lcd(infusao);
}

void digite_volume_a_infundir(void)  // imprime cabeçario
{
    imp_lcd (digite_volume);
    seg_linha();
    imp_lcd(total_a_infundir);
}
```



```
void zera_variaveis(void)    // zera as variaveis
{
    valor_total[0] = 0;
    valor_total[1] = 0;
    valor_total[2] = 0;
    valor_total[3] = 0;
    valor_total[4] = 0;
    valor_total[5] = 0;
    valor_total_somatorio_volume = 0;
    valor_total_somatorio = 0;
    i = 0;
    tela = 0;
}
void infusao_completa(void)
{
    clr_lcd();
    imp_lcd (infusao_completa_kvo);
    seg_linha();
    imp_lcd (infundindo_kvo);
}
void tempo_infundir(void)
{
    clr_lcd();
    imp_lcd (tempo_a_infundir);
    seg_linha();
    cent = tempo_infusao / 100;                /// o valor de tempo_infusão já foi calculado.

    tempo_infusao = tempo_infusao % 100;
    dez = tempo_infusao / 10;
    uni = tempo_infusao % 10;
    escreve_lcdnum (cent);
    escreve_lcdnum (dez);
    escreve_lcdnum (uni);
    imp_lcd (minutos);
    imp_lcd (e);
}
```