



**UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO - CECAU**



KLEUBER NASCIMENTO CAMPOS ARAÚJO

**CONTROLE EM MALHA ABERTA
DE POSICIONAMENTO
DE CILINDRO PNEUMÁTICO**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA
DE CONTROLE E AUTOMAÇÃO**

Ouro Preto, 2016

KLEUBER NASCIMENTO CAMPOS ARAÚJO

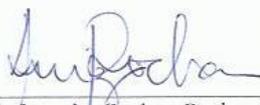
**CONTROLE EM MALHA ABERTA
DE POSICIONAMENTO
DE CILINDRO PNEUMÁTICO**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Luiz Joaquim Cardoso Rocha

Ouro Preto
Escola de Minas – UFOP
Março, 2016

Monografia defendida e aprovada, em 08 de março de 2016, pela comissão avaliadora constituída pelos professores:



Prof. Dr. Luiz Joaquim Cardoso Rocha - Orientador



Prof. M. Sc. Danny Augusto Vieira Tonidandel – Professor Convidado



Profa. Dra. Karla Boaventura Pimenta Palmieri – Professora Convidada

A659c Araújo, Kleuber Nascimento Campos.
 Controle em malha aberta de posicionamento cilindro
 pneumático
 [manuscrito] / Kleuber Nascimento Campos Araújo. – 2016.
 41f. : il., color., graf., tab.

 Orientador: Prof. Dr. Dr. Luiz Joaquim Cardoso Rocha.

 Monografia (Graduação) – Universidade Federal de Ouro
 Preto. Escola de Minas. Colegiado do Curso de Engenharia de
 Controle e Automação e Técnicas fundamentais.
 Área de concentração: Engenharia de Controle e

Fonte de catalogação: bibem@sisbin.ufop.br

SUMÁRIO

Sumário.....	1
Lista de Figuras.....	8
Lista de tabelas	9
Resumo.....	10
Abstract.....	11
1. Introdução.....	12
1.1. Objetivo	12
1.2. Justificativa.....	13
2. Elementos Pneumáticos	13
2.1. Cilindro de dupla ação.....	13
2.2. Válvula Eletropneumática Proporcional.....	15
2.3. Compressor.....	19
3. Desenvolvimento	19
3.1. Placa de Controle.....	19
3.1.1. Microcontrolador	19
3.1.2. Amplificador	21
3.1.3. <i>Layout</i>	22
3.1.4. <i>Firmware</i>	25
3.2. <i>Software</i> supervisorio.....	26
3.3. Comunicação	30
4. Controle e Resultados	33
4.1. Controle.....	33
4.1.1. Controlador <i>ON/OFF</i>	33
4.1.2. Controlador <i>Fuzzy</i>	34
4.1.3. Acionamento em Trajetória.....	35
4.2. Resultados	36
4.2.1. Controlador <i>ON/OFF</i>	36
4.2.2. Controlador <i>Fuzzy</i>	37
4.2.3. Acionamento em Trajetória.....	38

5. Trabalhos Futuros.....	39
6. Anexos.....	40
Referências	44

LISTA DE FIGURAS

Figura 2.1 - Foto bancada	13
Figura 2.2 - <i>Kit</i> Pneumática Proporcional.....	14
Figura 2.3 – Circuito acionamento cilindro dupla ação.....	14
Figura 2.4 - Equação da continuidade.....	15
Figura 2.5 - Válvula Proporcional de Vazão	16
Figura 2.6 - Acionamento da Válvula Proporcional para a direita.....	17
Figura 2.7 - Acionamento da Válvula Proporcional para a esquerda	17
Figura 2.8 - Gráfico Q x V - Válvula Proporcional de Vazão	18
Figura 2.9 – Acionamento de um cilindro de dupla ação.....	18
Figura 3.1 - PIC 18F2550 pinout	20
Figura 3.2 - Ilustração PIC 18F2550	21
Figura 3.3 - Circuito amplificador.....	21
Figura 3.4 - Circuito cristal.....	22
Figura 3.5 – Circuito reset	23
Figura 3.6 – Conector USB	23
Figura 3.7 – Circuito de gravação ICSP	24
Figura 3.8 – <i>Layout</i>	24
Figura 3.9 - Foto da placa	25
Figura 3.10 – Aba terminal	27
Figura 3.11 - Aba <i>Position Control</i>	28
Figura 3.12 - Aba <i>Trajectory</i>	29
Figura 3.13 – Aba Settings.....	30
Figura 4.1 – Curva experimental de trajetória	36
Figura 4.2 – Acionamento em Trajetória controle <i>Fuzzy</i>	38
Figura 6.1 – Fluxograma <i>firmware</i>	40
Figura 6.2 - Diagrama acionamento de posição.....	41
Figura 6.3 – Comunicação PIC Computador.....	42
Figura 6.4 - Comunicação Computador PIC.....	43

LISTA DE TABELAS

Tabela 4.1 – Controlador <i>Fuzzy</i>	35
Tabela 4.2 – Dados controle <i>ON/OFF</i> velocidade mínima	36
Tabela 4.3 – Dados controle <i>ON/OFF</i> velocidade máxima atraso 30mm	37
Tabela 4.4 – Dados controle <i>Fuzzy</i>	37

RESUMO

Este trabalho propõe a utilização de uma placa de controle em conjunto a um computador para controlar a movimentação de um atuador pneumático de dupla ação. A placa de controle é uma alternativa mais interessante que a utilização de um robusto controlador lógico programável, visto que seu custo é alto comparado a placa, e que um circuito externo ao CLP para a conexão entre o mesmo e válvula de controle pode se fazer necessário. O acionamento da válvula eletropneumática proporcional de vazão, que realiza a movimentação do cilindro pneumático, é realizado mediante comandos oriundos de um computador conectado via interface USB a placa de controle. O objetivo é realizar o controle de posicionamento do cilindro de dupla ação via um *software* supervisor executado no computador.

ABSTRACT

This work presents the construction of a control board to control an electro pneumatic valve witch will move a double action cylinder. The board uses a microcontroller as the control unit and communicates with a microcomputer through a USB interface. The control board is a more interesting way of controlling the valve than a Programmable logic controller since it costs a lot more compared to the board, and a circuit board might be necessary to connect the valve to the PLC anyway. The main goal is to control the double action cylinder's position with a software on the computer.

1. INTRODUÇÃO

A pneumática é sem dúvida o elemento mais simples, de maior rendimento e de menor custo que pode ser utilizado na solução de problemas de automação, devido a uma série de características próprias de seu fluido de utilização, o ar. Ela é um meio de deslocamento que permite altas velocidades, podendo atingir 10m/s em cilindros especiais e 500.000 rpm em turbinas pneumáticas (FIALHO, 2004).

Atuadores pneumáticos são componentes mecânicos que transformam energia pneumática em energia mecânica para movimentos retilíneos, rotativos e angulares. Como o fluido de operação é o ar, vazamentos em cilindros não causam contaminações, por isso seu uso é preferido.

Diferentemente dos sistemas somente eletroeletrônicos ou mecânicos, os elementos pneumáticos podem ser solicitados, em carga, até parar, sem sofrer danos, voltando ao seu funcionamento normal tão logo cesse a resistência (FIALHO, 2004).

Robótica é um campo relativamente novo da tecnologia moderna que atravessa as fronteiras tradicionais de engenharia. Compreender a complexidade dos robôs e suas aplicações requer o conhecimento de engenharia elétrica, engenharia mecânica, sistemas e engenharia industrial, informática, economia e matemática (SPONG, HUTCHINSON, VIDYASAGAR, 2005).

O trabalho de conclusão de curso consiste em elaborar uma interface que realiza o controle de posicionamento de um cilindro pneumático de dupla ação. O cilindro pode ser encarado como um manipulador robótico com um grau de liberdade.

1.1. OBJETIVO

Acionar uma válvula pneumática proporcional para controle de posição de um cilindro pneumático via um *software* supervisor. Para tanto, propõem-se a construção de uma placa de controle.

1.2. JUSTIFICATIVA

Com o crescente desenvolvimento da robótica se fazem necessários controles precisos para acionamento de atuadores em processos industriais, sejam em posicionamento e/ou movimentação em trajetória. Em processos como solda e pintura por exemplo um controle de posição rigoroso é imprescindível na obtenção de bons resultados.

2. ELEMENTOS PNEUMÁTICOS

2.1. CILINDRO DE DUPLA AÇÃO

O trabalho de conclusão de curso foi realizado utilizando o *Kit* Pneumática Proporcional da Festo *Didactic* Brasil que conta com uma válvula proporcional de vazão e um cilindro pneumático de dupla ação acoplado a uma régua potenciométrica. A Figura 2.1 é uma foto da bancada completa e o sistema pode ser observado pela Figura 2.2.



Figura 2.1 - Foto bancada

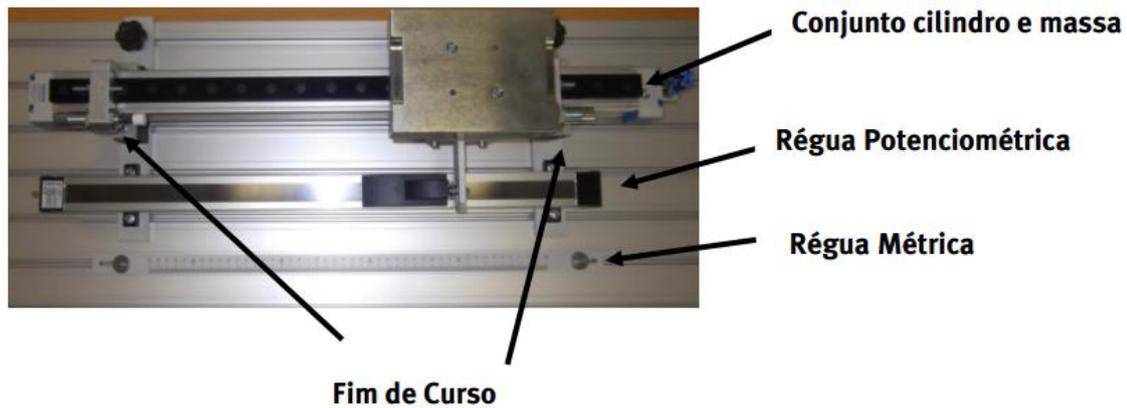


Figura 2.2 - *Kit Pneumática Proporcional (FESTO DIDACTIC BRASIL, 2012)*

O ar comprimido pode ser utilizado para mover, pressionar ou vibrar algum tipo de dispositivo de saída em atuadores pneumáticos. Cilindros pneumáticos são os atuadores mais comumente encontrados (BARBER, 1997).

Atuadores lineares, como o nome indica, são usados para mover um objeto ou aplicar uma força em uma linha reta. O atuador linear básico é o cilindro (PARR, 1998).

Cilindros podem ser de dupla ação, onde a pressão do fluido é usada para estender e retraindo o pistão, ou simples ação quando é acionado somente em um sentido. A Figura 2.3 ilustra o funcionamento de um cilindro de dupla ação.

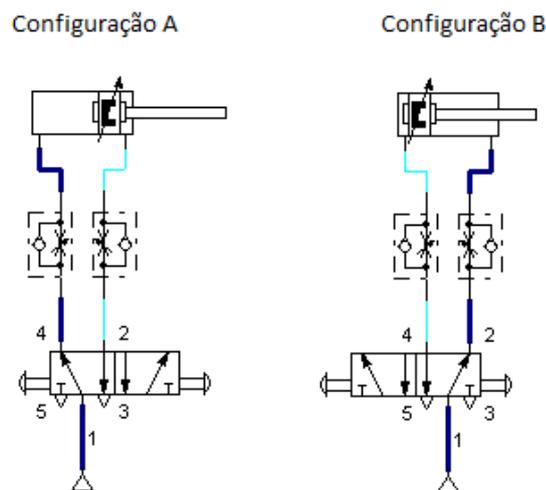


Figura 2.3 – Circuito acionamento cilindro dupla ação

Pela configuração A o ar passa pelo cilindro acionando-o para movimentar-se para a direita, já pela configuração B o ar passa pelo cilindro movimentando-o para a esquerda.

Acoplado ao atuador existe uma régua potenciométrica que permite a leitura da posição do mesmo. A régua potenciométrica, como o próprio nome já diz, funciona como um potenciômetro.

Um potenciômetro é simplesmente uma resistência ajustável com um intervalo a partir de zero a um valor definido. Um potenciômetro tem três pinos. Ao conectar apenas dois, o potenciômetro se torna um resistor variável. Utilizando todos, o potenciômetro torna-se um divisor de tensão (MCROBERTS, 2010).

No projeto utiliza-se a régua potenciométrica como um divisor de tensão.

2.2. VÁLVULA ELETROPNEUMÁTICA PROPORCIONAL

As válvulas de solenóides funcionam como um interruptor elétrico, isto é, elas podem estar ativadas ou desativadas. Em muitas aplicações é necessário controlar a velocidade, a pressão ou força de um atuador através de um sinal elétrico. Esta função é fornecida por válvulas proporcionais, nelas a posição pode ser colocada em qualquer lugar entre 0% e 100% do curso por variação da corrente do solenoide (PARR, 1998).

Considerando que o sistema é alimentado constantemente com ar comprimido provido por um compressor, pode-se utilizar para um volume de controle fixo a equação da continuidade disposta na Figura 2.4.

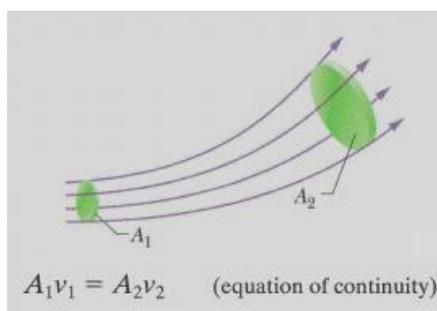


Figura 2.4 - Equação da continuidade (HALLIDAY, RESNICK, WALKER, 2008)

A relação entre a velocidade e a área da secção transversal é a chamada equação da continuidade para o fluxo de um fluido ideal. Ela nos diz que a velocidade aumenta quando a área da secção transversal diminui através do fluxo (HALLIDAY, RESNICK, WALKER, 2008).

Portanto, controlando-se a posição da válvula proporcional controla-se a abertura de passagem do fluido dentro da válvula, ou seja, a área de passagem de ar. Controlando a área, consequentemente controlam-se a velocidade do fluido e a velocidade de movimentação do atuador.

A Figura 2.5 é uma ilustração simbólica de uma válvula proporcional.

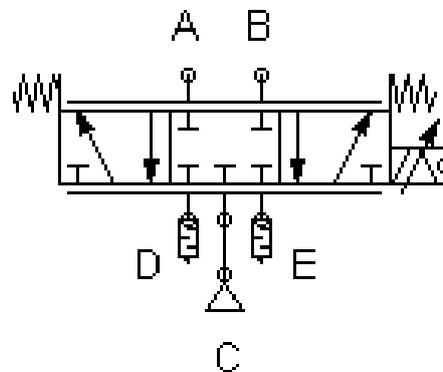


Figura 2.5 - Válvula Proporcional de Vazão

A válvula é acionada eletricamente de acordo com a tensão nela aplicada, a vazão de saída é controlada da seguinte maneira:

- Para o repouso, a tensão aplicada deve ser 5V (Figura 2.5);
- Aplicando-se uma tensão entre 0V e 5V ela permite a passagem do ar de C para B, passando pelo elemento a ser acionado e posteriormente saindo de A para D, com vazão inversamente proporcional a tensão aplicada (Figura 2.6);

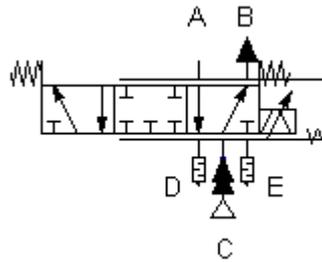


Figura 2.6 - Acionamento da Válvula Proporcional para a direita

- Aplicando-se uma entre 5V e 10V ela permite a passagem do ar de C para A, passando pelo elemento a ser acionado e posteriormente saindo de B para E, com vazão proporcional a tensão aplicada menos 5V (Figura 2.7);

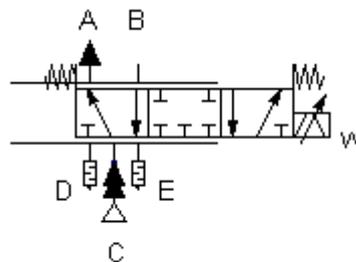


Figura 2.7 - Acionamento da Válvula Proporcional para a esquerda

Observando a Figura 2.8 o funcionamento da válvula fica mais claro. Pode-se perceber também que a válvula possui uma zona morta (um intervalo de um domínio de sinal ou faixa onde não ocorre nenhuma ação) perto de 5V.

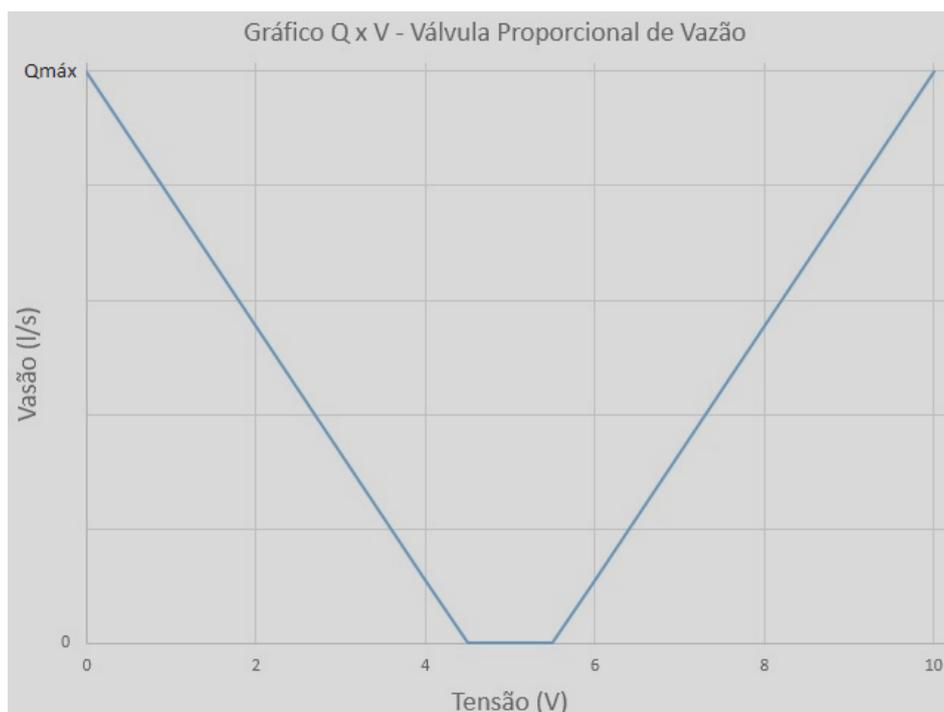


Figura 2.8 - Gráfico Q x V - Válvula Proporcional de Vazão

A válvula é alimentada por uma fonte de corrente contínua de 24V. A Figura 2.9 representa o acionamento de um cilindro de dupla ação por uma válvula proporcional de vazão.

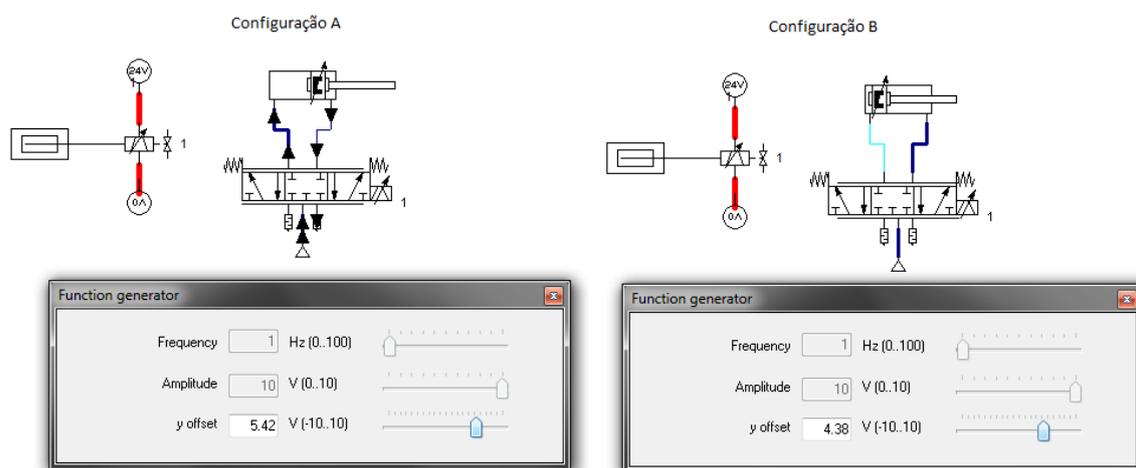


Figura 2.9 – Acionamento de um cilindro de dupla ação por uma válvula proporcional de vazão

Na figura, observa-se a esquerda, tanto na configuração A quanto na B um gerador de sinais alimentado por uma fonte de 24V. Esse gerador de sinais

realiza o acionamento da válvula proporcional. Na configuração A o gerador de sinais aplica na válvula 5,42V, movimentando o atuador para a direita. Já na configuração B o gerador de sinais aplica na válvula 4,38V, movimentando o atuador para a esquerda.

2.3. COMPRESSOR

A grande maioria dos sistemas pneumáticos usa ar comprimido como fluido de operação, um número pequeno de sistemas utiliza nitrogênio obtido comercialmente a partir de fornecedores de gás líquido (PARR, 1998).

Compressor é o equipamento que executa o processo de transformar o ar atmosférico em ar comprimido para ser empregado nos sistemas pneumáticos.

No sistema pneumático utilizado, o ar é comprimido por um compressor e armazenado em um reservatório, não sendo assim necessário que o compressor trabalhe continuamente. O compressor atua somente quando a pressão cair ao valor mínimo ajustado no pressostato.

3. DESENVOLVIMENTO

3.1. PLACA DE CONTROLE

3.1.1. MICROCOTROLADOR

Com os avanços tecnológicos os circuitos de controle diminuíram de tamanho e ganharam em desempenho. Os primeiros computadores que tinham dimensões de uma sala de aula, são facilmente ultrapassados em desempenho quando comparados aos *smartphones* atuais. Todos esses dispositivos possuem uma unidade de controle.

Microcontroladores e microprocessadores são pequenas unidades de controle, com diferentes tipos de circuitos integrados dentro e com algumas interfaces para comunicação externa (QURESHI, 2013).

Diferentemente dos microprocessadores, os microcontroladores são mais simples e possuem memórias RAM e ROM internas, portanto eles necessitam de menos circuitos integrados externos para funcionar (PEREIRA, 2002).

Processadores são circuitos integrados que possuem como objetivo processar vários tipos de dados diferentes, ou seja, eles são utilizado para tarefas gerais. Já microcontroladores são utilizados para tarefas específicas.

Para a placa desenvolvida escolheu-se o microcontrolador PIC 18F2550 da *Microchip*, como unidade de controle. Ele dispõe de entradas analógicas (para leitura da régua potenciométrica), saídas PWM (para o acionamento da válvula), interface para comunicação USB e outras interfaces que não são utilizadas no projeto.

Abaixo a Figura 3.1 mostra a disposição dos pinos do PIC e a é Figura 3.2 uma ilustração do mesmo.

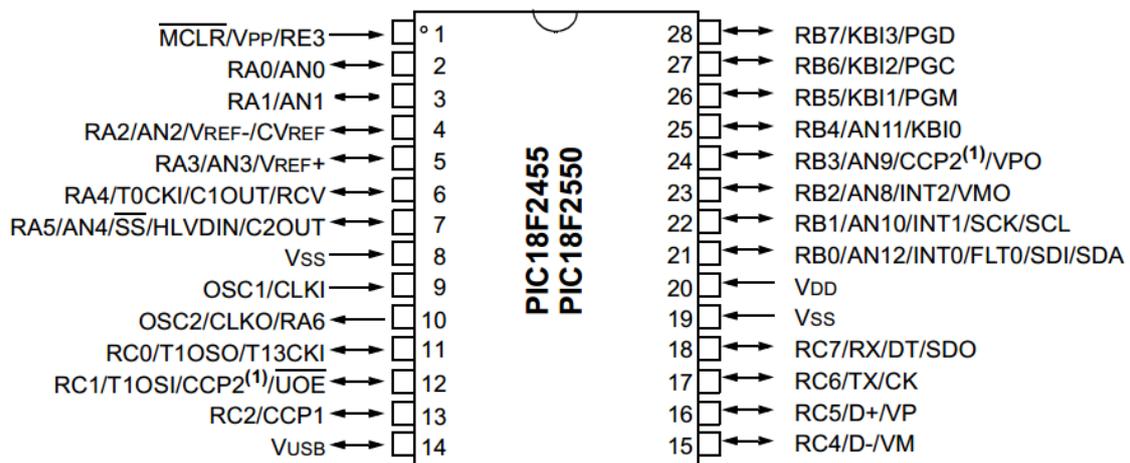


Figura 3.1 - PIC 18F2550 pinout (MICROCHIP, 2004)



Figura 3.2 - Ilustração PIC 18F2550

3.1.2. APLIFICADOR

O cilindro é acionado pela passagem de ar permitida pela válvula proporcional, que por sua vez é acionada por uma saída analógica na faixa (0 a 10)V. Como microcontrolador trabalha na faixa de tensão de (0 a 5)V, para ajustar a saída PWM do PIC para a faixa de operação da válvula utiliza-se o amplificador operacional LM741. A Figura 3.3 mostra o circuito utilizado.

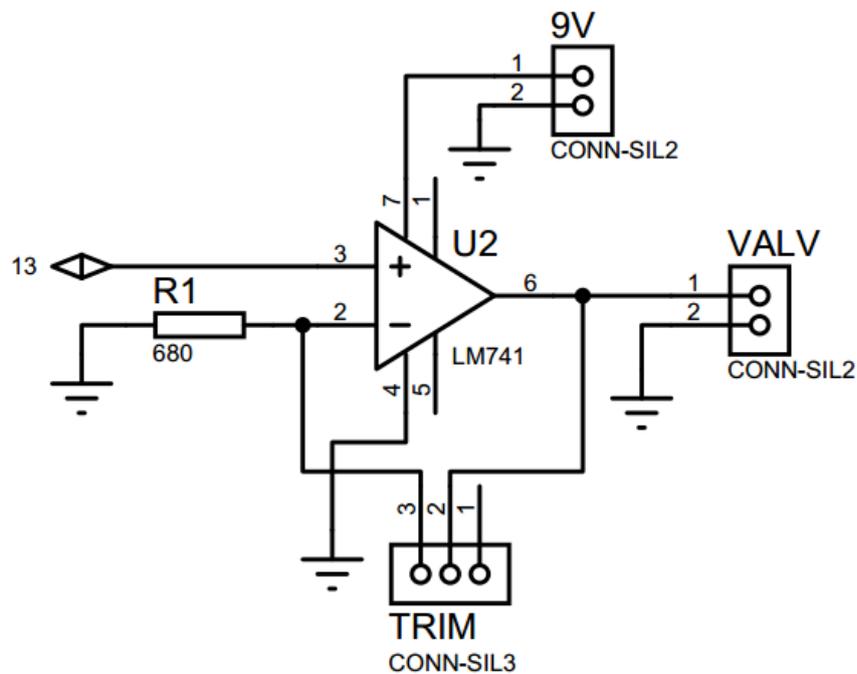


Figura 3.3 - Circuito amplificador

Um amplificador operacional é um amplificador diferencial de ganho muito alto, com alta impedância de entrada e baixa impedância de saída. Os usos típicos do amplificador operacional são proporcionar mudanças de amplitude de tensão (amplitude e polaridade), osciladores, circuitos de filtro, e diversos tipos de circuitos de instrumentação (BOYLESTAD e NASHELSKY, 1998).

O amplificador é alimentado por uma fonte assimétrica de 9V, o pino 3 do amplificador está ligado ao pino 13 do PIC que é uma saída PWM. O pino 2 está ligado a um resistor de 680kohm ligado ao terra e a um *trimpot* de 1kohm (para um ajuste fino da tensão de saída) fazendo a realimentação com o pino de acionamento da válvula.

3.1.3. LAYOUT

Utiliza-se para o microcontrolador um circuito com um cristal de 16MHz juntamente com dois capacitores cerâmicos de 22pF conectados aos pinos 9 e 10 (OSC1 e OSC2) conforme as Figura 3.4 e Figura 3.1.

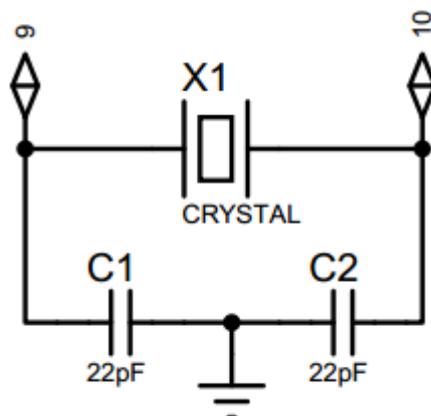


Figura 3.4 - Circuito cristal

No pino 1 do PIC (MCLR conforme a Figura 3.1), monta-se um circuito de *reset* com uma resistência, um diodo e um *push button*, de acordo com a Figura 3.5. Segundo o circuito, o pino 1 do PIC se encontra sempre em estado alto (5V), somente quando o *push button* é pressionado seu estado torna-se baixo (GND), reiniciando assim o microcontrolador.

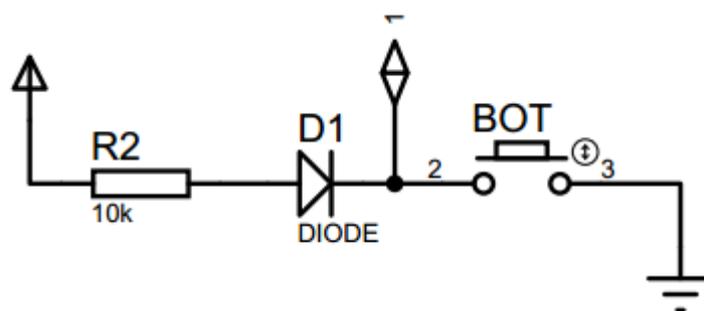


Figura 3.5 – Circuito reset

Adiciona-se um conector USB B ligado aos pinos D+ (pino 16) e D- (pino 15) do PIC, opta-se por realizar a alimentação da placa pelo conector. Um capacitor de 1nF foi adicionado entre o VCC e GND do conector. A Figura 3.6 mostra o circuito do conector USB.

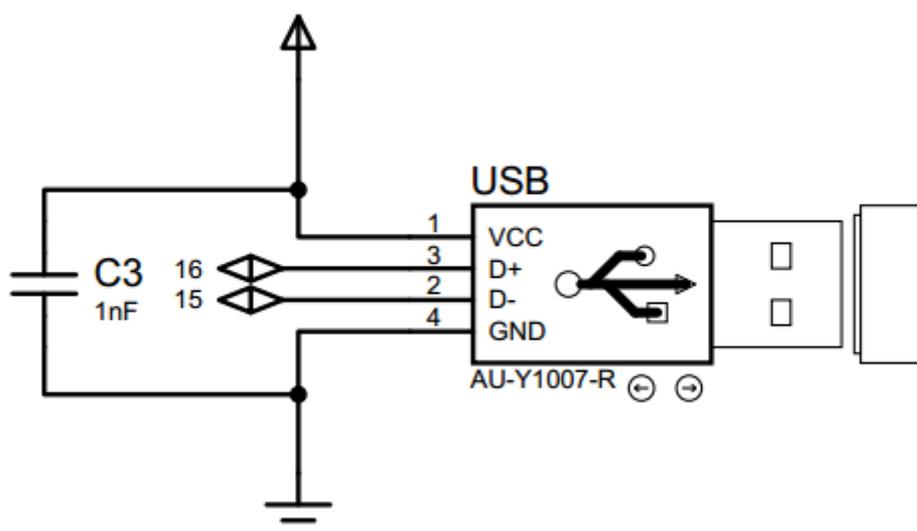


Figura 3.6 – Conector USB

Para evitar desgastes no microcontrolador e facilitar no desenvolvimento e testes do *firmware* da placa, adiciona-se um circuito de gravação ICSP. O circuito consiste apenas de um conector de 5 pinos ligado aos pinos do microcontrolador conforme Figura 3.7. Os pinos utilizados no circuito gravação são os pinos 1 (MCLR), 27 (PGC) e 28 (PGD).

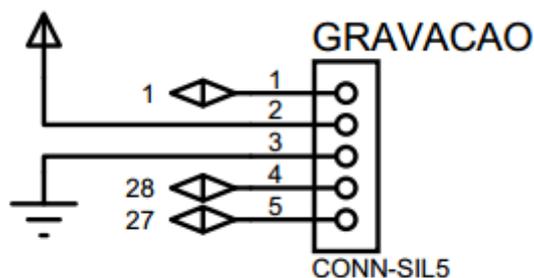


Figura 3.7 – Circuito de gravação ICSP

Foram colocados na placa três conectores, um para a régua potenciométrica, outro para a válvula e o terceiro para uma fonte externa. O circuito do amplificador operacional LM741 juntamente com a válvula e a fonte externa pode ser observado na secção 3.1.2. O conector da régua potenciométrica tem 3 pinos, um conecta-se ao VCC, outro ao GND e o último a entrada analógica AN3 (pino 5) do PIC. Também foram adicionados, mas não são utilizados, um conector para um sensor de distância e outro para um adaptador *Bluetooth*.

A Figura 3.8 mostra o layout da placa e a Figura 3.9 é uma foto da mesma.

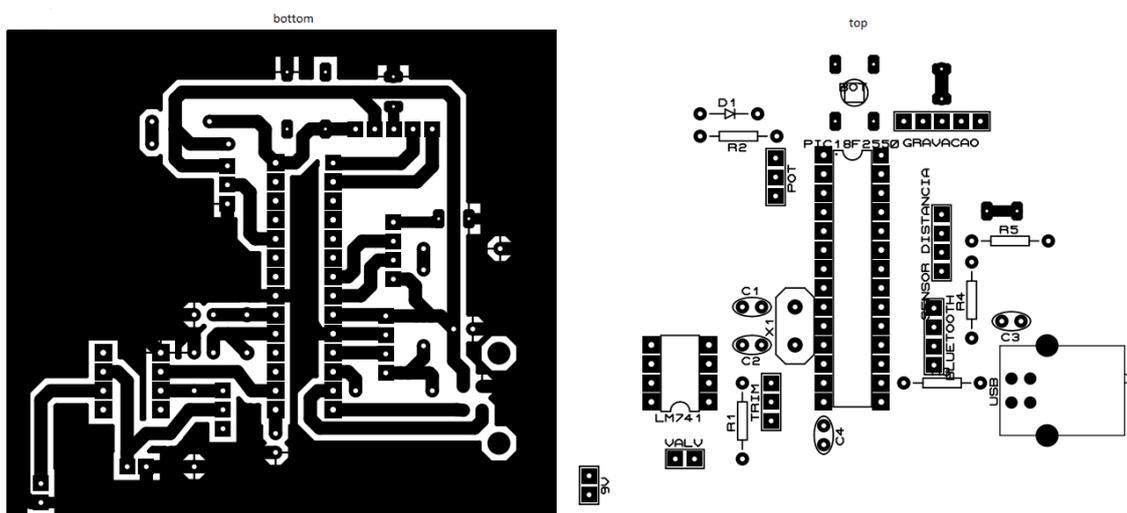


Figura 3.8 – Layout

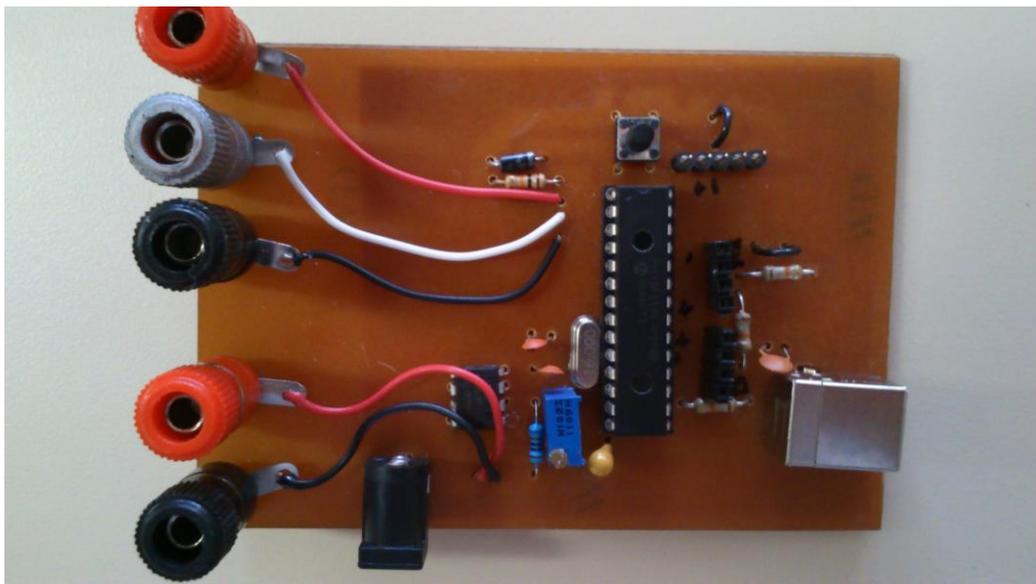


Figura 3.9 - Foto da placa

Como os cabos para o acionamento da válvula e leitura da régua potenciométrica são do tipo banana, fez-se necessária a adição de conectores do tipo banana na placa de controle. A ligação entre a placa e os conectores do tipo banana é realizada através de fios.

3.1.4. FIRMWARE

O Microcontrolador PIC 18F2550 é um dispositivo programável, ele executa ações de acordo com um arquivo hexadecimal gravado em sua memória. Portanto, para que o mesmo se comportasse da maneira desejada, desenvolve-se uma rotina de controle para seu funcionamento.

A rotina de programação do microcontrolador é um *loop* infinito, que realiza ações de acordo com os dados recebidos pela interface USB do *software* supervisor.

A movimentação para a posição desejada pode ser realizada de dois modos distintos, através de um controle *ON/OFF* ou *Fuzzy*, que serão discutidos detalhadamente na secção 4 (Controle).

No programa de controle do microcontrolador ainda existe uma rotina de interrupção que estoura a cada 500ms. Essa rotina envia periodicamente a posição do cilindro para o *software* supervisor. Porém, durante os

acionamentos de posição as interrupções são desabilitadas, portanto não é possível observar a variação da posição do cilindro durante sua movimentação, somente após o término.

Outra interrupção realiza a movimentação em trajetória, periodicamente o cilindro é enviado para a próxima posição dentro de um vetor de posições, a movimentação em trajetória é melhor explicada na secção 4 (Controle).

A Figura 6.1, nos anexos, representa um fluxograma do funcionamento do *firmware*.

3.2. SOFTWARE SUPERVISÓRIO

Programas ou aplicativos supervisórios são *softwares* capazes de realizar o controle de *hardwares* de maneira facilitada e intuitiva para o usuário. No projeto, implementa-se um aplicativo supervisório na linguagem C#, que permite o controle da posição do cilindro, através de um microcomputador ligado por uma porta USB a placa de controle.

C# foi criada na *Microsoft* no final de 1990, e seu arquiteto-chefe era Anders Hejlsberg. Como uma linguagem orientada a objeto, C# suporta encapsulamento de dados, herança, polimorfismo, e método de substituição. É uma linguagem poderosa e simples para a construção de aplicações robustas. C# utiliza a biblioteca de classes .NET e é automaticamente portátil para todos os ambientes .NET (HARWANI, 2015).

O .NET é um framework de desenvolvimento de *software* da *Microsoft*. Ele é uma coleção de ferramentas, tecnologias e linguagens que fornece um ambiente para construir e implementar aplicativos corporativos. O .NET *Framework* pode ser executado nos sistemas operacionais *Windows* (HARWANI, 2015).

Utiliza-se a IDE (*integrated development environment*, ou em português, ambiente de desenvolvimento integrado) *Visual Studio* da empresa *Microsoft* pra a construção do aplicativo supervisório.

A aplicação tem quatro abas, um menu superior e um rodapé. No menu superior o usuário tem as opções de conectar com a placa de controle ou sair

da aplicação. A opção *About* dispõe informações sobre o desenvolvedor e rodapé exibe o status da conexão da aplicação com a placa.

Na tela da aplicação existem quatro abas Terminal, *Position Control*, *Trajectory* e *Settings* conforme as Figura 3.10 a Figura 3.13.

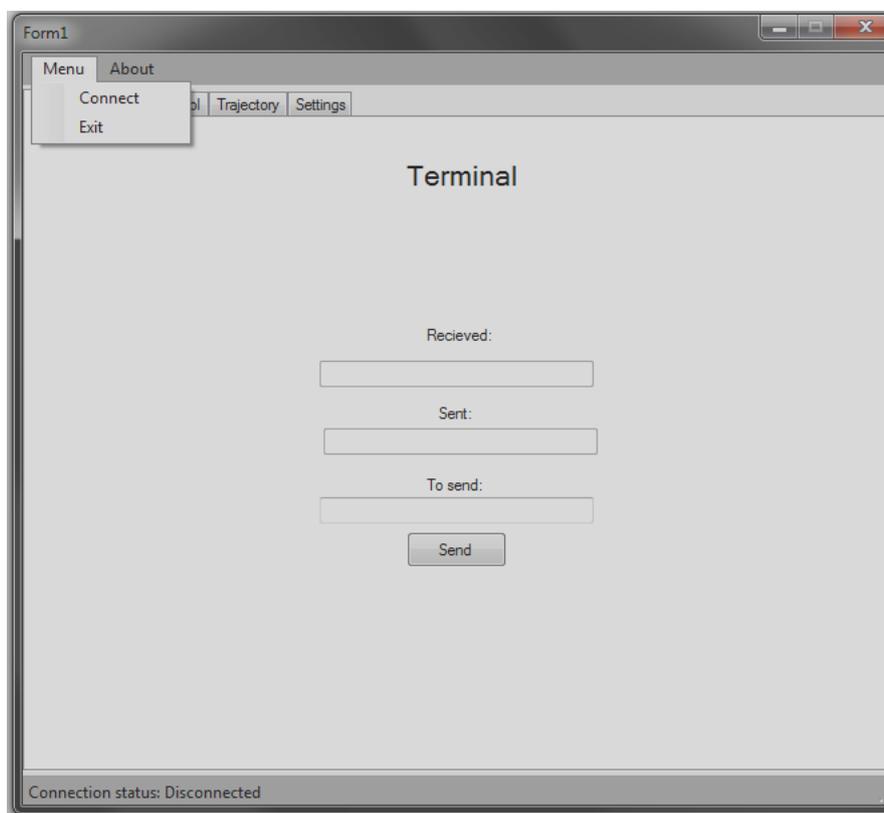


Figura 3.10 – Aba terminal

A aba Terminal executa apenas a comunicação direta entre o microcontrolador e o *software* supervisorío, nela é possível observar como os dados são enviados pelo PIC, bem como enviar dados ao mesmo. Ela é de grande ajuda no processo de confecção do *software*, pois proporciona uma melhor depuração na realização da comunicação.

A aba Terminal possui três *Text Boxes*, no *Text Box Received* o usuário recebe informações provenientes do microcontrolador, no *Text Box To send* o usuário pode enviar comandos ao mesmo e no *Text Box Sent* o usuário pode visualizar as informações envidas anteriormente. Quando o usuário pressiona o botão *Send*, o conteúdo do *Text Box* acima do botão é enviado a microcontrolador.

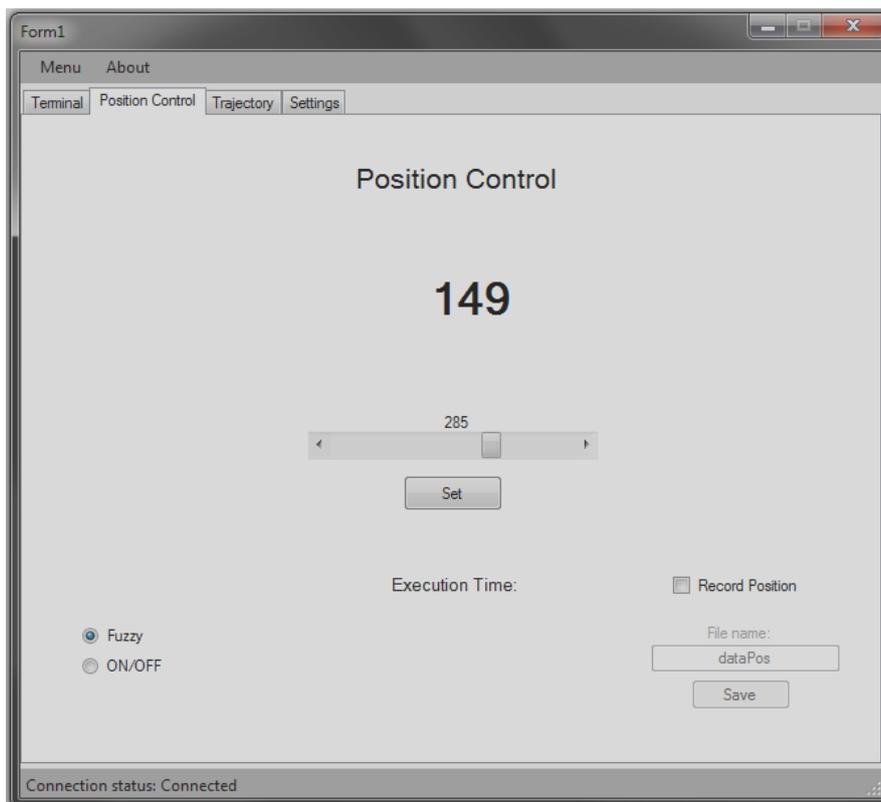


Figura 3.11 - Aba *Position Control*

A aba *Position Control* é a aba onde o controle de posição é realizado. Nela existe um *Label* onde a posição do cilindro é exibida, um *hScrollBar* para ajustar a posição para qual o cilindro será enviado, um botão *Set* que inicia a movimentação, dois *Radio Buttons* para a escolha do tipo de deslocamento e o *Label Execution Time* que dispõe o tempo gasto na movimentação.

Após escolher o tipo de movimentação *Fuzzy* ou *ON/OFF*, o usuário ajusta na barra horizontal a nova posição desejada para o cilindro, clica no botão e o cilindro movimenta-se para a posição ajustada na barra. Depois que a movimentação é finalizada, pode-se observar o tempo gasto no deslocamento pelo *Label Execution Time* e comparar para qual posição o cilindro realmente desloca-se pelo valor exibido pelo *Label* de posição.

Ainda existe na aba *Position Control* um *Check Box*, que quando está marcado o programa supervisor guarda informações sobre os deslocamentos realizados pelo cilindro. Essas informações podem ser salvas clicando no botão *Save*. O nome do arquivo a ser salvo pode ser escolhido no *Label* acima do botão. Um exemplo de como as informações são salvas é a Tabela 4.2.

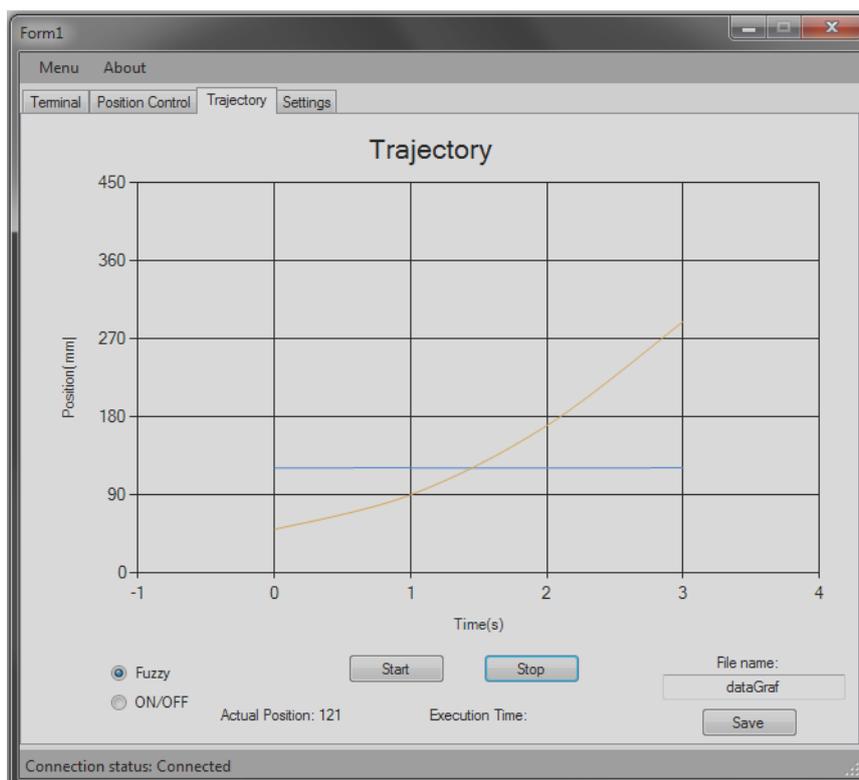


Figura 3.12 - Aba Trajectory

A aba *Trajectory* exibe em tempo real a posição do cilindro de duplação versus o tempo. A aba foi criada para a realização de trajetórias pelo cilindro.

Na aba existem dois botões *Start* e *Stop*, um gráfico, o *Label Actual Position*, dois *Radio Buttons* para a escolha do tipo de deslocamento e o *Label Execution Time* que dispõe o tempo gasto na movimentação de uma posição para outra. Quando o usuário pressiona o botão *Start* desenha-se o gráfico e o efetuator começa a movimentação, quando pressiona o botão *Stop*, o desenho é interrompido e o cilindro para de movimentar-se.

Após pressionar o botão *Stop* o usuário tem a opção de salvar os dados do gráfico em um arquivo através do botão *Save*. O nome do arquivo a ser salvo pode ser escolhido no *Label* acima do botão.

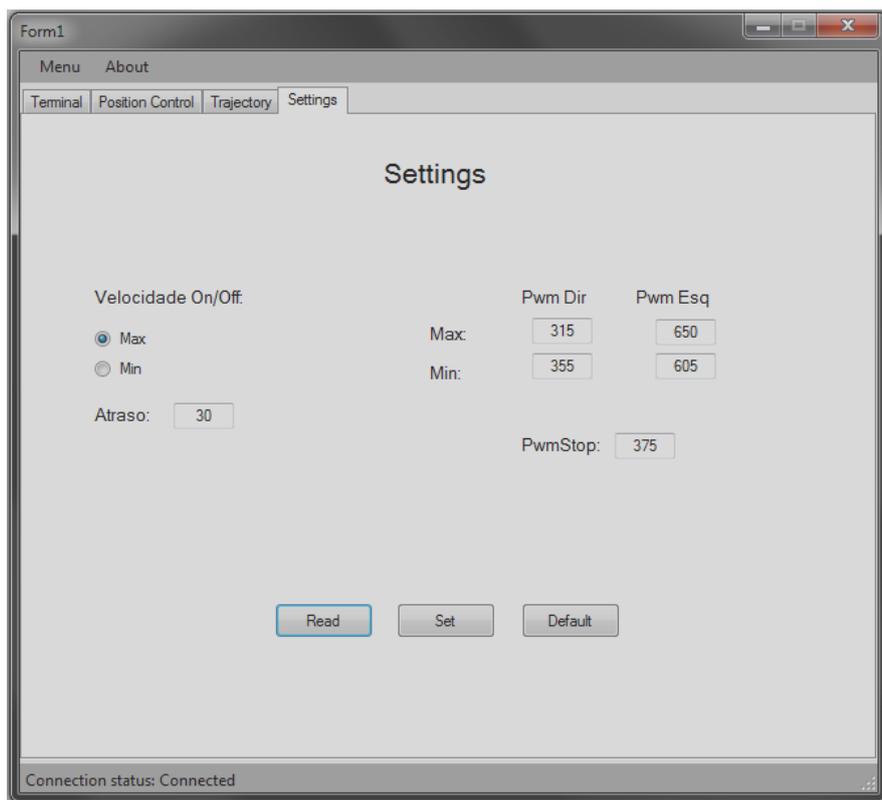


Figura 3.13 – Aba Settings

Na aba *Settings* existem dois *Radio Buttons* e o *Label Atraso* para as configurações da movimentação *ON/OFF*. Existem ainda, cinco outros *Labels* para calibração das tensões a serem aplicadas na válvula de controle, ou seja, as velocidades e a tensão para a válvula parada. O funcionamento desta aba pode ficar mais claro após a leitura da secção 4 (Controle).

Ao clicar no botão *Read* o usuário faz a leitura das configurações atuais da placa. O botão *Set* envia para a placa as configurações ajustadas e o botão *Default* ajusta a placa para as configurações padrão.

3.3. COMUNICAÇÃO

A comunicação entre o microcontrolador e o computador é realizada através de uma interface USB HID, estabelece-se que em toda troca de informação, seja do PIC para o microcomputador ou do microcomputador para o PIC, sempre serão tocados 5 *bytes*, mas nem sempre todos são utilizados. As Figura 6.3 e Figura 6.4, nos anexos, ilustram a comunicação.

O PIC pode enviar ao o programa supervisorio os seguintes caracteres:

- 'y', inicia no supervisorio uma contagem de tempo para determinar quanto tempo um deslocamento do cilindro ir  durar, todas as vezes que o cilindro inicia uma movimentac o a placa de controle envia esse caractere;
- 'x', interrompe a contagem de tempo e exibe nos *Labels Execution Time* das abas *Trajectory* e *Position Control* o tempo de movimentac o decorrido;
- 'e', a placa de controle envia o caractere para indicar algum erro em seu funcionamento;
- 'z', ajusta a velocidade de movimentac o *ON/OFF* para m xima;
- 'v', ajusta a velocidade de movimentac o *ON/OFF* para m nima;
- 'p', o caractere   enviado ao *software* supervisorio para informar a posic o atual do efetuador, como a comunicac o   realizada em *bytes* (0 a 255), e a posic o do cilindro varia de 50mm a 450mm, o caractere   seguido por 2 outros *bytes* que somados s o a posic o do cilindro;
- '1','2','3','4','5','6', seguido de outros quatro *bytes* de dados. Esses caracteres s o utilizados para enviar ao supervisorio os dados da aba *Settings*. V rios dados dessa aba s o 10 *bits*, portanto s o necess rios 4 *bytes* para envi -los.

O *software* pode enviar para a placa de controle os seguintes caracteres:

- 's', caractere enviado ao microcontrolador para ajuste de posic o, como citado anteriormente, a comunicac o   realizada em *bytes* (0 a 255),

portanto o caractere é seguido por 2 outros *bytes* que somados são a posição do cilindro a ser ajustada;

- ‘i’, habilita no PIC a interrupção que envia ao supervisor a posição do cilindro;
- ‘d’, desabilita no PIC a interrupção que envia ao supervisor a posição do cilindro;
- ‘t’, habilita no PIC a interrupção para a execução de trajetória;
- ‘e’, desabilita no PIC a interrupção para a execução de trajetória;
- ‘f’, ajusta o tipo de movimentação para *Fuzzy*;
- ‘o’, ajusta o tipo de movimentação para *ON/OFF*;
- ‘r’, faz a requisição do envio dos dados de configurações;
- ‘g’, ajusta a velocidade de movimentação *ON/OFF* para máxima;
- ‘h’, ajusta a velocidade de movimentação *ON/OFF* para mínima;
- ‘w’, ‘l’, ‘k’, ‘j’, ‘a’, ‘q’, seguido de outros quatro *bytes* de dados. Esses caracteres são utilizados para enviar ao PIC os dados da aba *Settings*. Vários dados dessa aba são 10 *bits*, portanto são necessários 4 *bytes* para enviá-los.

4. CONTROLE E RESULTADOS

4.1. CONTROLE

O acionamento de posição do cilindro pode ser representado pelo diagrama mostrado na Figura 6.2, nos anexos. O controlador recebe do supervisor uma ordem para movimentação para uma posição arbitrária [p].

Se [p] é maior que a posição atual o controlador informa ao supervisor o início da movimentação, abre a válvula para a direita, e a mantém aberta enquanto a posição [p] for maior que a posição atual, quando [p] torna-se menor ou igual a posição atual a válvula é fechada e informa-se ao supervisor o fim da movimentação.

Se [p] é menor que a posição atual o controlador informa ao supervisor o início de movimentação, abre a válvula para a esquerda, e a mantém aberta enquanto a posição [p] for menor que a posição atual, quando [p] torna-se maior ou igual a posição atual a válvula é fechada e informa-se ao supervisor o fim da movimentação.

4.1.1. CONTROLADOR *ON/OFF*

O controlador *ON/OFF* funciona como descrito na Figura 6.2, nos anexos, ele é acionado de acordo com as opções ajustadas na aba *Settings* do *software* supervisor.

Pode-se ajustar o funcionamento do controlador *ON/OFF* para o cilindro movimentar-se com velocidades máxima ou mínima. Dependendo da velocidade ajustada pode ser necessário a utilização de um atraso para que o cilindro movimente-se de forma mais precisa para a posição desejada.

O atraso não é nada além que uma adequação para diminuir a distância entre a posição ajustada e a posição obtida no controle de posição. Por exemplo, se a posição ajustada é 300mm e a posição obtida for 315mm, o atraso será de 15mm. Nesse caso, o controlador fecharia a válvula de controle

quando a posição fosse 285mm, assim o cilindro é ajustado para a posição 300mm.

Definiu-se que o maior atraso seria de 30mm, pois atrasos implicam a limitação da menor distância de movimentação do cilindro. Por exemplo, se o atraso for ajustado para 25mm, o cilindro não conseguirá mover distâncias menores que 25mm.

Quando atrasos maiores que 30mm são necessários, ou seja, a diferença da posição ajustada e a posição obtida no controle de posição é maior que 30mm, diminui-se a velocidade máxima de movimentação.

A velocidade mínima representa a máxima velocidade de movimentação do cilindro sem o uso do atraso. Em velocidades baixas de movimentação não utiliza-se atraso, portanto atribuir-se ao mesmo valor zero na aba *Settings*.

4.1.2. CONTROLADOR FUZZY

Para o controlador *Fuzzy* além das velocidades mínimas e máximas calibradas para o controlador *ON/OFF*, foram ajustadas outras velocidades intermediárias que variam de acordo com a distância do efetuator ao ponto de movimentação. As velocidades intermediárias são obtidas automaticamente calibrando-se as velocidades máximas e mínimas na aba *Settings*.

Sendo assim para o controlador *Fuzzy* existem no total oito velocidades diferentes, quatro para a esquerda e quatro para a direita. Classificando a velocidade *ON/OFF* máxima como normal (esquerda/direita), pode-se classificar as outras como baixa (esquerda/direita), muito baixa (esquerda/direita) e baixíssima (esquerda/direita).

Para as distâncias considera-se que uma distância maior que 150mm é considerada grande, distâncias entre 150mm e 100mm são médias, distâncias entre 100mm e 50mm pequenas e distâncias menores que 50mm muito pequenas.

Então, para o controlador *Fuzzy* tem-se a seguinte Tabela 4.1:

Distância	Velocidade
Grande	Normal
Média	Baixa
Pequena	Muito baixa
Muito pequena	Baixíssima

Tabela 4.1 – Controlador *Fuzzy*

Portanto, quando selecionado no supervisório a opção de movimentação *Fuzzy*, o atuador movimenta-se para a posição ajustada com velocidade variável, de acordo com a diferença da distância entre a posição atual do efetuador e a posição ajustada.

4.1.3. ACIONAMENTO EM TRAJETÓRIA

Para o acionamento em trajetória gera-se um vetor de posições com seis pontos. O acionamento é feito ponto a ponto, ou seja, o atuador pneumático é enviado periodicamente para os pontos gerados na curva.

A curva utilizada é sempre a mesma, portanto o cilindro pode realizar a movimentação em trajetória sem mesmo estar conectado ao *software* supervisório, desde que o controlador seja previamente programado para isso.

O supervisório a cada estouro de tempo plota a posição real do atuador e a posição experimental, podendo-se assim comparar-se a curva experimental com a real.

A Figura 4.1 é a curva experimental que deseja-se que seja repedita pelo atuador.

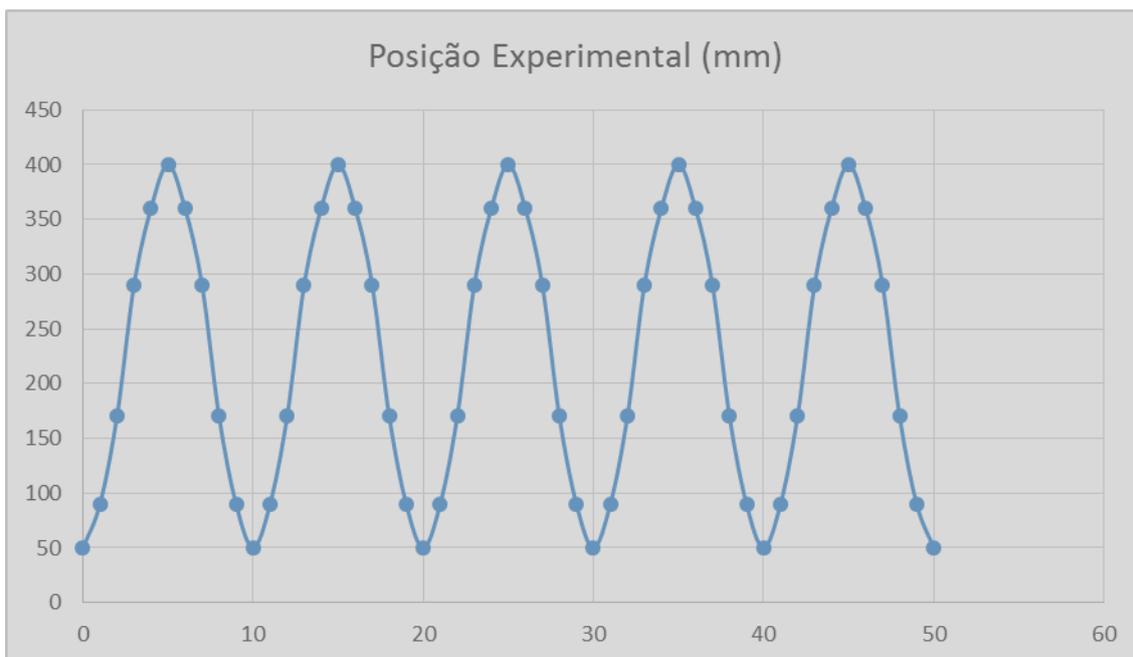


Figura 4.1 – Curva experimental de trajetória

4.2. RESULTADOS

4.2.1. CONTROLADOR ON/OFF

O comportamento do controlador *ON/OFF* pode ser observado pelas Tabela 4.2 e Tabela 4.3.

Direção	Tipo de Controle	Tempo de execução (s)	Posição Inicial Experimental (mm)	Posição Obtida Experimental (mm)	Deslocamento Experimental (mm)	Posição Inicial Real (mm)	Posição Obtida Real (mm)	Deslocamento Real (mm)
Direita	ON/OFF	5,8	100	340	240	103	336	233
Esquerda	ON/OFF	6,7	340	100	240	335	116	219
Direita	ON/OFF	5,8	100	340	240	110	338	228
Esquerda	ON/OFF	7,1	340	100	240	336	101	235
Direita	ON/OFF	6,4	100	340	240	101	332	231
Esquerda	ON/OFF	7,5	340	100	240	337	110	227
Direita	ON/OFF	6,2	100	340	240	108	334	226

Tabela 4.2 – Dados controle *ON/OFF* velocidade mínima

Pelos dados obtidos na Tabela 4.2, podem-se obter o tempo médio de movimentação igual a 6,50s e o deslocamento médio é igual a 228,43mm, com ambos, obtém-se o tempo médio de movimentação por milímetro igual 28,46ms/mm. O erro médio do atuador é de 6,71mm.

Direção	Tipo de Controle	Tempo de execução (s)	Posição Inicial Experimental (mm)	Posição Obtida Experimental (mm)	Deslocamento Experimental (mm)	Posição Inicial Real (mm)	Posição Obtida Real (mm)	Deslocamento Real (mm)
Esquerda	ON/OFF	1	340	100	240	341	133	208
Direita	ON/OFF	0,8	100	340	240	101	301	200
Esquerda	ON/OFF	0,9	340	100	240	336	131	205
Direita	ON/OFF	0,8	100	340	240	94	304	210
Esquerda	ON/OFF	0,9	340	100	240	336	129	207
Direita	ON/OFF	0,8	100	340	240	99	306	207
Esquerda	ON/OFF	0,9	340	100	240	338	132	206

Tabela 4.3 – Dados controle ON/OFF velocidade máxima atraso 30mm

Segundo a Tabela 4.3, o tempo médio de movimentação é igual a 0,871s e o deslocamento médio é igual a 206,14mm, esse deslocamento somado ao atraso de 30mm é igual a 236,14mm, portanto o tempo médio de movimentação por milímetro é igual a 3,688ms/mm. O erro médio do atuador descontando-se o atraso de 30mm é de 3,71mm.

Conclui-se então, pelos dados obtidos, que controlador ON/OFF com velocidade máxima e atraso de 30mm mostra-se mais rápido e apresenta menor erro entre a posição ajustada e a obtida.

4.2.2. CONTROLADOR FUZZY

Pela Tabela 4.4 pode-se observar o comportamento do controlador de posição Fuzzy.

Direção	Tipo de Controle	Tempo de execução (s)	Posição Inicial Experimental (mm)	Posição Obtida Experimental (mm)	Deslocamento Experimental (mm)	Posição Inicial Real (mm)	Posição Obtida Real (mm)	Deslocamento Real (mm)
Esquerda	Fuzzy	1,7	340	100	240	338	104	234
Direita	Fuzzy	1,9	100	340	240	99	334	235
Esquerda	Fuzzy	1,6	340	100	240	337	108	229
Direita	Fuzzy	1,9	100	340	240	105	333	228
Esquerda	Fuzzy	1,6	340	100	240	337	110	227
Direita	Fuzzy	1,7	100	340	240	105	331	226
Esquerda	Fuzzy	1,8	340	100	240	334	99	235

Tabela 4.4 – Dados controle Fuzzy

De acordo com os dados da Tabela 4.4, obtém-se o tempo médio de movimentação igual a 1,74s e o deslocamento médio igual a 230,57mm, então o tempo médio de movimentação por milímetro é igual a 7,54ms/mm. O erro médio do efetuador é de 6,43mm.

Segundo os dados obtidos, observa-se que o controlador *Fuzzy* obteve tempo de movimentação entre o controlador *ON/OFF* de máxima velocidade e o de mínima, o que já era de se esperar pois ele utiliza velocidades intermediárias a deles.

O controlador *ON/OFF* com velocidade máxima e atraso de 30mm mostrou-se o melhor entre todos de acordo com os dados obtidos, porém ele possui a limitação de não conseguir movimentar o cilindro distâncias menores que 30mm.

O controlador *Fuzzy* apresentou erro médio no posicionamento do efetuator próximo ao controlador *ON/OFF* com velocidade mínima, mas conseguiu reduzir o tempo de movimentação em 73,5% sem utilizar atraso, possibilitando a movimentação em pequenas distâncias.

4.2.3. ACIONAMENTO EM TRAJETÓRIA

A Figura 4.2 representa o gráfico do acionamento do cilindro em trajetória.

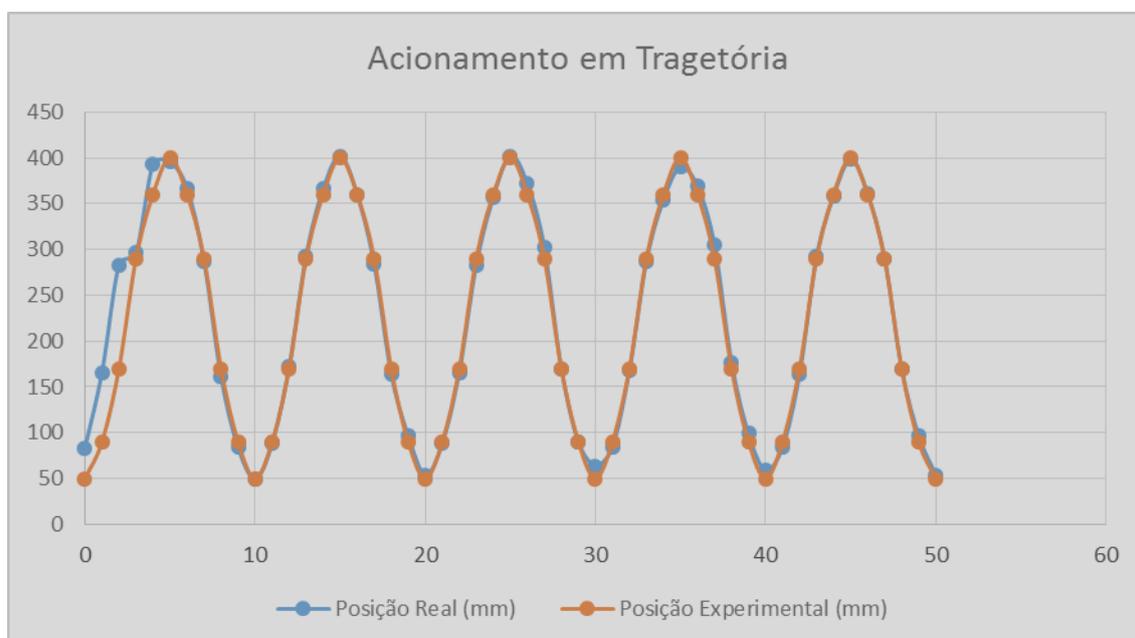


Figura 4.2 – Acionamento em Trajetória controle *Fuzzy*

Pela Figura 4.2 pode-se comparar os pontos experimentais com os pontos obtidos no acionamento prático do atuador. No gráfico observa-se que os pontos obtidos na prática estão próximos aos experimentais e que o gráfico real coincide em sua maior parte com o experimental.

Utiliza-se somente a movimentação de posicionamento com controle *Fuzzy* para o acionamento em trajetória, os outros controles de posição *ON/OFF* com velocidade mínima ou máxima não se mostraram satisfatórios.

5. TRABALHOS FUTUROS

A placa desenvolvida ainda está em sua primeira versão, alterações em seu *layout* bem como seus componentes podem ser feitas para o desenvolvimento de uma versão final.

A primeira alteração seria a mudança em seu protocolo de comunicação. Ao invés da comunicação USB construída para esta versão da placa de controle, pode-se utilizar a comunicação *Bluetooth*.

A adição de um módulo adaptador *Bluetooth* dispensaria a necessidade do uso de um microcontrolador PIC da família 18F, que possui comunicação USB. Um modelo de microcontrolador mais simples representa uma diminuição no custo final da placa, mas aquisição de um módulo adaptador *Bluetooth* representa um custo a mais.

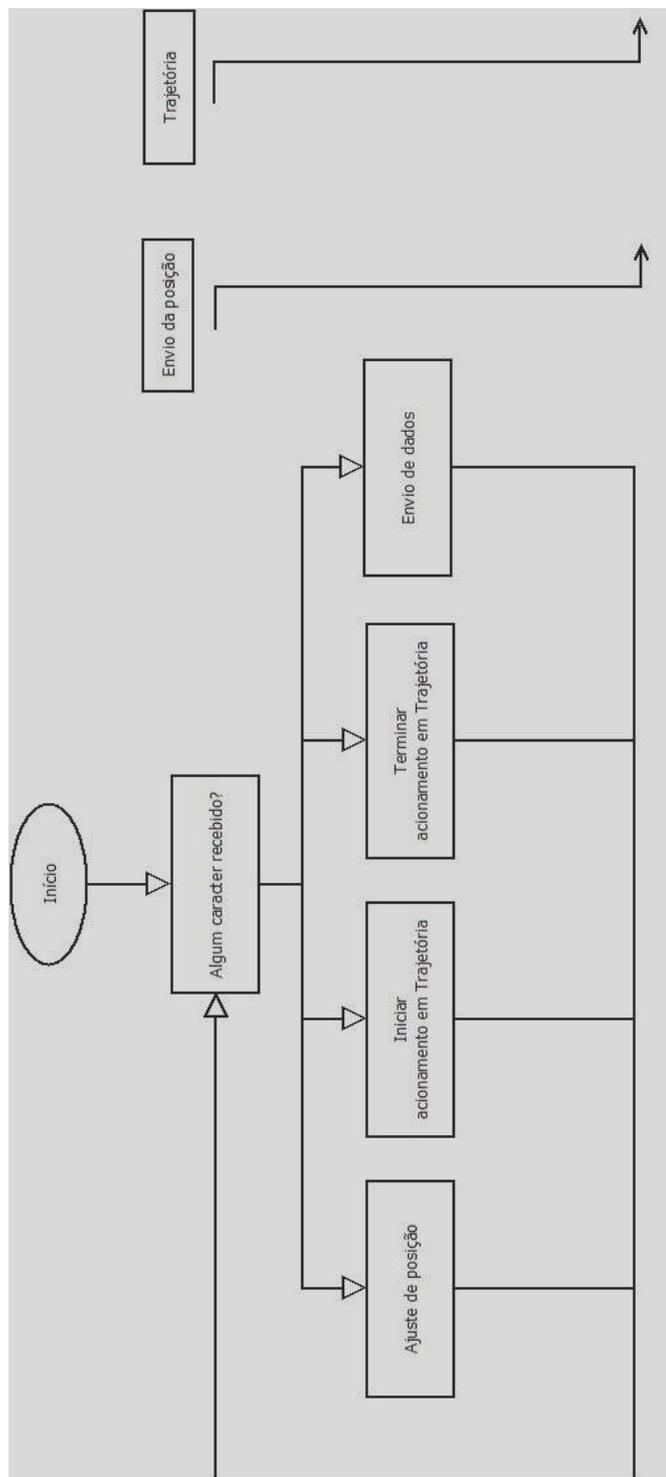
Com o conector USB, que realiza a alimentação da placa, sendo removido, a alimentação da placa de controle poderia ser realizada pela fonte de 9V junto a um regulador de tensão LM7805.

Outros pontos favoráveis a comunicação *Bluetooth*, são que com ela evita-se a necessidade da conexão de fios entre o microcomputador e a placa de controle, e permite-se a comunicação com outros dispositivos além do computador, como por exemplo *smartphones* e *tablets*.

Uma desvantagem é que computadores mais antigos, que não possuam uma interface *Bluetooth*, precisariam de um adaptador para comunicar-se com a placa.

Ainda são tópicos para trabalhos futuros a implementação da realização de trajetórias retangulares e triangulares para o efetuador pneumático.

6. ANEXOS

Figura 6.1 – Fluxograma *firmware*

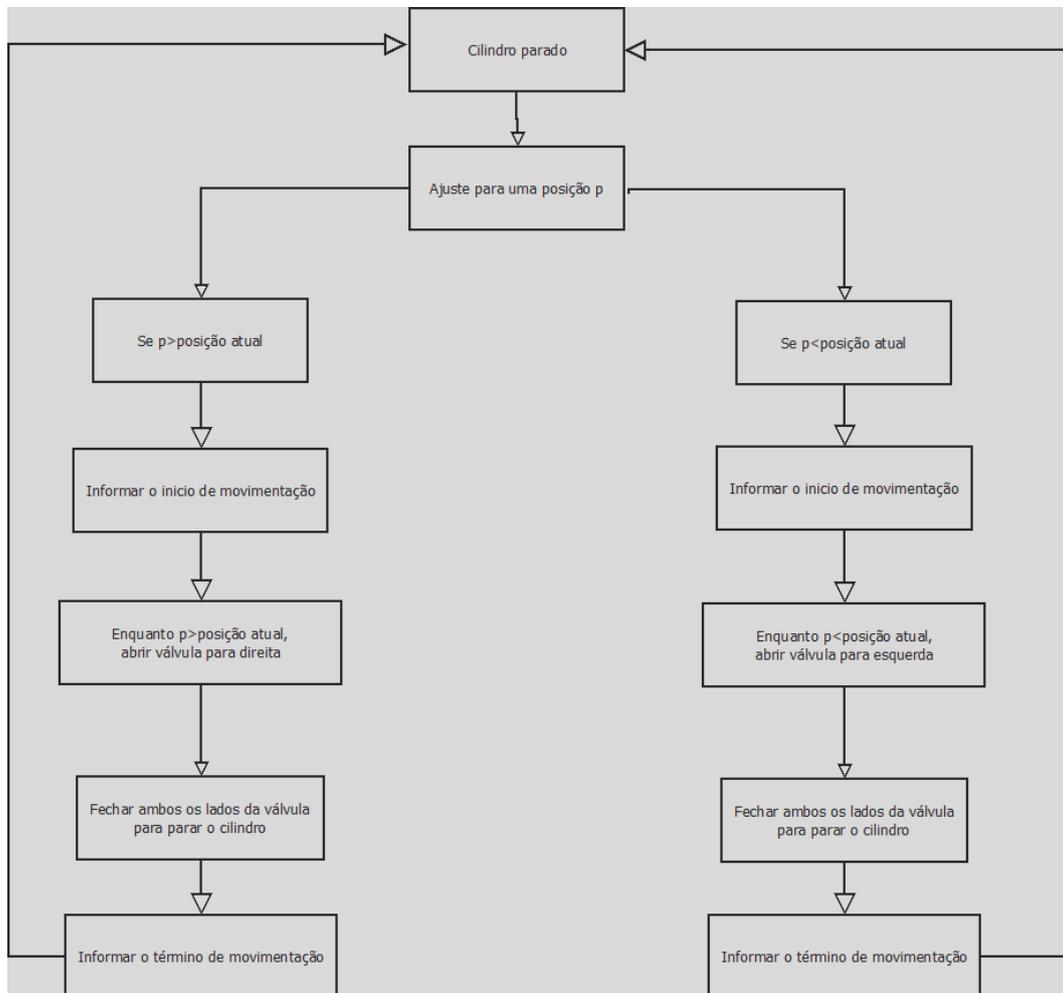


Figura 6.2 - Diagrama acionamento de posição

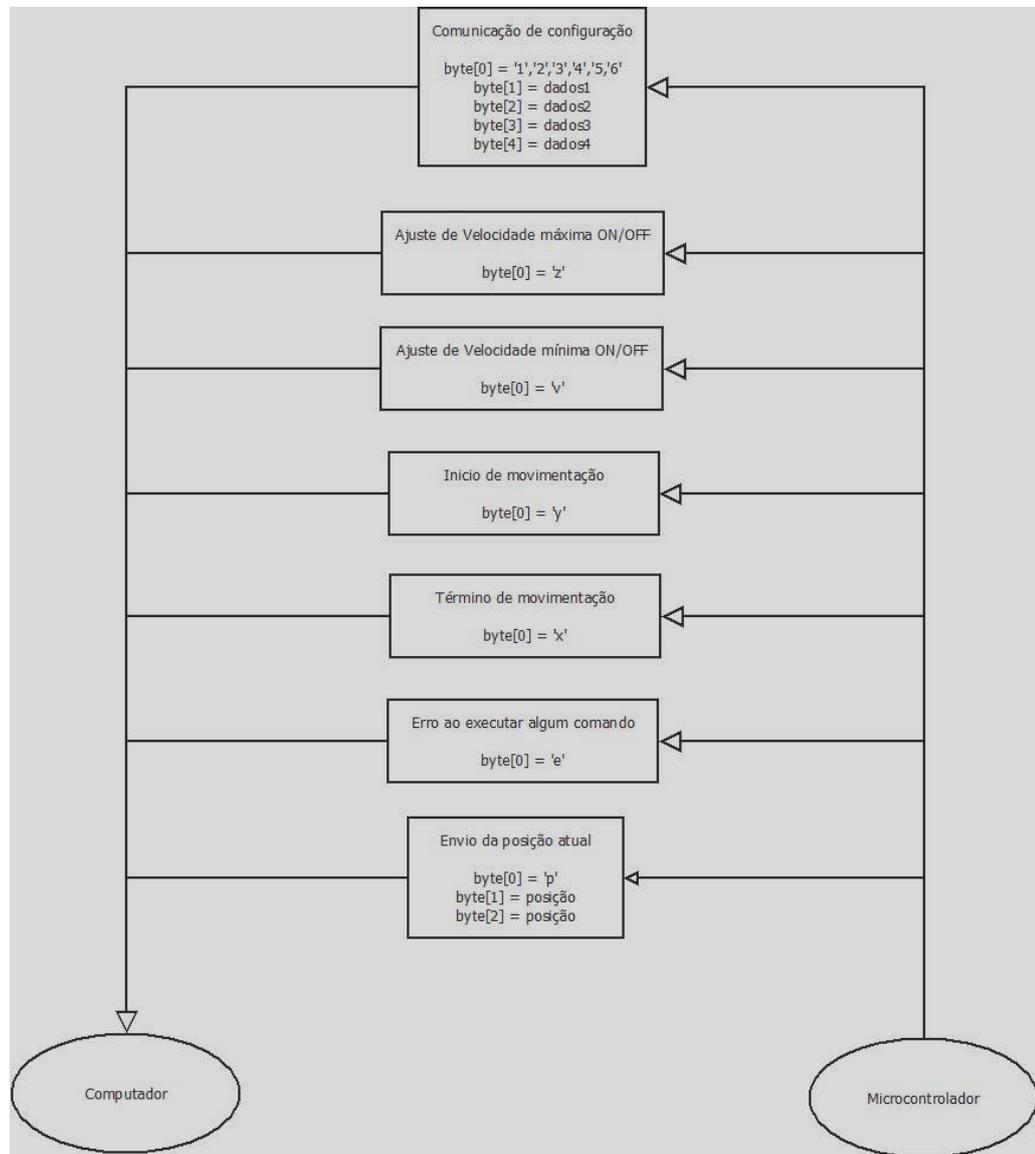


Figura 6.3 – Comunicação PIC Computador

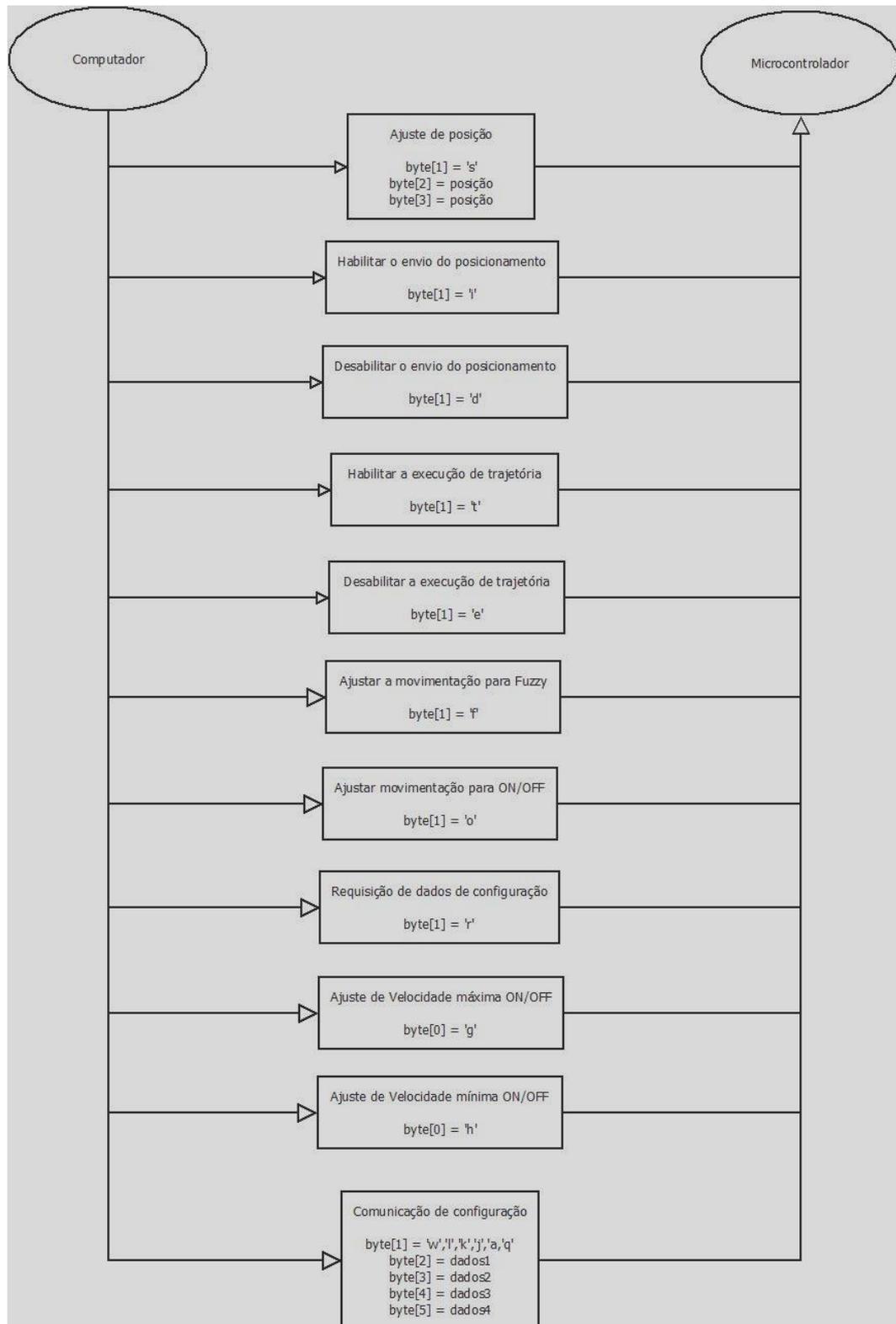


Figura 6.4 - Comunicação Computador PIC

REFERÊNCIAS

BARBER, A. **Pneumatic Handbook**. 8th. ed. Oxford: Elsevier Advanced Technology, 1997.

BOYLESTAD, R.; NASHELSKY, L. **Eletronic Devices and Circuit Theory**. 7th. ed: Prentice Hall, 1998.

FESTO DIDACTIC BRASIL. **Kit Pneumática Proporcional**. 1ª. ed. São Paulo: Festo Brasil Ltda, 2012.

FIALHO, A. B. **Automação Pneumática Projetos, Dimencinamento e Análise de Circuitos**. 2ª. ed. São Paulo: Érica Ltda, 2004.

HALLIDAY D.; RESNICK R.; WALKER J. **Fundamentals of physics**. 8th. ed.: John Wiley & Sons, Inc, 2008.

HARWANI, B. M. **Learning Object-Oriented Programming in C# 5.0**. Boston: Cengage Learning, 2015.

SPONG W. M.; HUTCHINSON S.; VIDYASAGAR M. **Robot Modeling and Control**: JOHN WILEY & SONS, INC., 2005.

MCROBERTS, M. **Beginning Arduino**. 1st ed. New York: Apress, 2010.

MICROCHIP. **PIC18F2455/2550/4455/4550 DataSheet**. 2004.

PARR, A. **Hydraulics and pneumatics: a technician's and engineer's guide**. 2nd ed. Oxford: Elsevier, 1998.

PEREIRA, F. **Microcontroladores PIC**. São Paulo: Érica Ltda, 2002.

QURESHI, A. I. **Teach yourself PIC Microcontrollers For Absolute Beginners**: Microtronics Pakistan, 2013.