

UNIVERSIDADE FEDERAL DE VIÇOSA  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

WILLIAN RAMOS DO CARMO

**CIRCUITO DE MONITORAMENTO DA VELOCIDADE DE UM  
MOTOR USANDO ENCODER INCREMENTAL**

VIÇOSA  
2015

WILLIAN RAMOS DO CARMO

**CIRCUITO DE MONITORAMENTO DA VELOCIDADE DE UM  
MOTOR USANDO ENCODER INCREMENTAL**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Kétia Soares Moreira

VIÇOSA  
2015



**WILLIAN RAMOS DO CARMO**

**CIRCUITO DE MONITORAMENTO DA VELOCIDADE DE UM  
MOTOR USANDO ENCODER INCREMENTAL**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 19 de junho de 2015.

**COMISSÃO EXAMINADORA**

---

**Prof.<sup>a</sup> Dr.<sup>a</sup> Kétia Soares Moreira - Orientadora**  
**Universidade Federal de Viçosa**

---

**Prof.<sup>o</sup> Dr. André Gomes Tôrres A - Membro**  
**Universidade Federal de Viçosa**

---

**Eng.<sup>o</sup> Vinícius Leonardo Gadioli da Silva - Membro**  
**Universidade Federal de Viçosa**

*“Obstáculos são aqueles perigos que você vê quando tira os olhos do seu objetivo.”*

*Henry Ford*

*Primeiramente dedico essa vitória à Deus, pois sem ele nada seria possível e é através dele que todos os nossos sonhos são realizados. Dedico essa conquista especialmente à minha família, minha mãe Amélia, pois sem seu incentivo e amor eu nada seria. Dedico ao meu pai Venâncio grande exemplo de honestidade e perseverança e as minhas irmãs Fran e Lara por todo o apoio que me deram. Dedico também à todos os meus amigos de Viçosa e da Engenharia Elétrica por estarem do meu lado todo esse tempo. Finalmente eu consegui.*

## *Agradecimentos*

À minha orientadora Kétia Soares Moreira, sem o seu apoio este trabalho não seria possível. Agradecimentos também ao Departamento de Engenharia Elétrica da Universidade Federal de Viçosa, por todo o conhecimento transmitido a mim e por me tornar Engenheiro Eletricista.

## *Resumo*

O controle preciso das variáveis que compõem um processo é imprescindível para a qualidade do produto final. Com essa necessidade em foco os avanços da tecnologia nos campos da instrumentação e eletrônica foram crescentes nos últimos anos. Hoje uma infinidade de sensores transmite informações sobre os parâmetros físicos que acompanham um processo industrial. Temperatura, umidade, velocidade, luminosidade, pressão ou até mesmo a concentração de gases nas linhas de produção são fatores físicos que requerem controle para produzir produtos de qualidade e que atendam às expectativas do mercado [1].

Atualmente a automação utiliza sinais oriundos dos sensores para a tomada de decisões dentro dos processos de modo a maximizá-los [2]. Além disso, algumas funções repetitivas são retiradas das obrigações dos operários das linhas de produção melhorando suas condições de trabalho e reduzindo os riscos de acidentes [3].

Também conhecidos como transdutores, alguns sensores tem a capacidade de converter em um sinal elétrico mensurável os estímulos físicos ou químicos do ambiente a ser monitorado [4] [5]. De posse do sinal dos sensores, a aquisição e monitoramento das variáveis desejadas de um sistema podem ser feitas com o auxílio de componentes eletrônicos conhecidos como microcontroladores capazes de promover o processamento dos sinais dos sensores em informação útil [6] [7] [8]. A atuação do sistema sobre o processo dependerá da informação que é repassada aos atuadores pelo elemento de processamento [9].

A velocidade de motores é um exemplo de variável de processo onde a instrumentação e a eletrônica são aplicadas com o intuito de promover maior precisão e desempenho para o sistema. Em aplicações industriais como robôs, esteiras e máquinas CNCs são utilizados sensores conhecidos como *encoders* capazes de transformar o movimento rotativo em um trem de impulsos elétricos que auxiliam na determinação do deslocamento angular dos motores [10]. Basicamente existem dois tipos de *encoders*: incremental e absoluto. Os dois tipos são capazes de fornecer informações sobre a velocidade e sentido de rotação, porém apenas o encoder absoluto pode fornecer informações sobre posicionamento [11].

Este trabalho terá como objetivo a confecção de um circuito de aquisição e monitoramento da velocidade de um motor usando um encoder incremental como elemento sensor. O sinal do *encoder* será condicionado e enviado a um microcontrolador fabricado pela



MICROCHIP chamado PIC da família 16F877A que processará os dados e os converterá no valor da velocidade do motor disponível ao usuário através de um display LCD para acompanhamento em tempo real. Para essa tarefa será necessários conhecimentos em linguagem C de programação e sobre as funções avançadas de funcionamento dos microcontroladores PIC, tais como interrupção externa, *timers* e uso da biblioteca LCD [12].

Posteriormente esse sistema de monitoramento de velocidade (tacômetro digital) será utilizado para a confecção de um controle de velocidade de um motor do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa. O sistema também poderá ser utilizado para futuros ensaios nos laboratórios de máquinas elétricas como ferramenta didática para aulas práticas ou para estudos do comportamento de máquinas rotativas.

Conforme dito anteriormente a qualidade dos dados recebidos é fator determinante para a tomada de decisão e o produto final de quaisquer processos sejam eles industriais ou não. Analisando a folha de dados do *encoder* incremental foi possível constatar que a precisão e durabilidade do mesmo dependem de alguns fatores ambientais tais como temperatura e umidade. Então é importante que o usuário tenha o conhecimento de que as condições em que o *encoder* deverá operar são as condições ideais sugeridas pelo seu fabricante. Com essa preocupação em mente será sugerido acrescentar ao sistema de monitoramento um sistema de medição da temperatura do *encoder* que imprime no display a temperatura e se o este valor ultrapassou limite do equipamento.

## *Abstract*

The precise control of the variables that made a process is essential for the quality of the final product. Focusing in this need, the advances in technology in the instrumental and electronics fields have been growing in the last years. Today a lot of sensors transmit information about the physical parameters that accompanying na industrial process. Temperature, humidity, speed, light, pressure, or even the concentration of gases in the production lines are physical factors that require control to produce quality products that meet market expectations [1].

Currently automation uses signals from the sensors to decision-making within the mode processes to maximize them [2]. In addition, some repetitive functions are removed from the obligations of the workers of the production lines improving their working conditions and reducing the risk of accidents [3].

Also known as transducers, some sensors have the ability to convert into a measurable electrical signal physical or chemical stimuli from the environment to be monitored [4] [5]. Of the sensor signal possession, acquisition and monitoring of the desired variables of a system can be made with the aid of electronic components known as microcontrollers capable of promoting the processing of signals from the sensors into useful information [6] [7] [8] . The system performance of the process will depend on the information that is passed on to the actuators by processing element [9].

The speed of the motors is an example of process variable where the instrumentation and electronics are applied in order to promote greater accuracy and performance for the system. In industrial applications such as robots, CNC machines and treadmills are used sensors known as encoders able to turn the rotary motion into a electrical pulse train output that help in determining the angular displacement of the motors [10]. There are basically two types of encoders: incremental and absolute. Both types are able to provide information on the speed and direction of rotation, but only the absolute encoder can provide information about positioning [11]. This work will aim at making an acquisition circuit and monitoring the speed of a motor using an incremental encoder as the sensor element. The encoder signal is conditioned and sent to a microcontroller manufactured by MICROCHIP called PIC of the family 16F877A that will process the data and convert the motor speed value available to the user via an LCD display for real-time monitoring. For this task will be necessary knowledge in

C programming language and about the advanced features of operation of PIC microcontrollers, such as external interrupt, timers and use of LCD library [12].

Afterwards this speed monitoring system (digital tachometer) will be used to make a speed control of a motor Electrical Engineering Department of the Universidade Federal de Viçosa. The system can also be used for future trials in electrical machines laboratory as a teaching tool for practical classes or behavioral studies of rotating machinery.

As previously said the quality of the data received is a determining factor for the decision making and the end product of any process whether industrial or not. Analyzing the data sheet of incremental encoder has been established that the accuracy and durability of it depends on some environmental factors such as temperature and humidity. So it is important that the user has the knowledge of the conditions in which the encoder will operate are the ideal conditions suggested by the manufacturer. With this in mind it will be suggested to add to the monitoring system an encoder temperature measurement system that prints on display the temperature and if this amount exceeded the limit equipment.

## *Sumário*

1	Introdução.....	15
1.1	Tacômetros .....	17
1.1.1	Introdução.....	17
1.1.2	Tacômetros Centrífugos .....	17
1.1.3	Tacômetro Eletromagnético .....	18
1.1.4	Tacômetro Óptico.....	19
1.2	Encoders .....	19
1.2.1	Introdução.....	19
1.2.2	Funcionamento .....	21
1.3	Microcontrolador PIC.....	22
1.3.1	Introdução.....	22
1.3.2	PIC16F877A.....	23
1.4	Display LCD.....	24
2	Materiais e Métodos .....	26
3	Resultados e Discussões.....	27
3.1	Aquisição dos dados via encoder.....	27
3.2	Microprocessamento dos dados.....	28
3.2.1	Funções pertinentes do PIC .....	28
3.2.2	Fluxograma.....	29
3.2.3	Compilador do algoritmo em Linguagem C.....	30
3.2.4	Configuração do <i>Timer 0</i> : interrupção interna .....	31
3.2.5	Configuração do <i>Timer 1</i> : interrupção externa.....	33
3.3	Simulação do circuito .....	35
3.4	Desenvolvimento do circuito.....	37
3.5	Projeto do circuito impresso .....	40
3.6	Sugestões de melhorias.....	41
4	Conclusões.....	42
	Referências Bibliográficas .....	42
	Apêndice A – Folha de Dados do Encoder Óptico Incremental HTR – 1024 – 3 – 5LV.....	44
A.1	Características Elétricas.....	44
A.2	Características Mecânicas.....	44
A.3	Características Ambientais .....	44

## *Lista de Figuras*

Figura 1 – Princípio de Funcionamento do Tacômetro Centrífugo [19].	18
Figura 2 – Princípio de Funcionamento do Tacômetro Eletromagnético [18].	18
Figura 3 - Princípio de funcionamento do tacômetro óptico [18].	19
Figura 4 - Composição básica de um <i>encoder óptico</i> [17].	21
Figura 5 - Funcionamento do <i>encoder</i> : pulsos produzidos [17].	22
Figura 6 - Microcontrolador PIC16F877A [21].	23
Figura 7 - Pinagem do Microcontrolador PIC16F877A [23].	24
Figura 8 - Módulo LCD [25].	25
Figura 9 - <i>Encoder óptico incremental</i> HTR-W2-1024-3-5LV [26].	28
Figura 10 - Fluxograma do algoritmo.	30
Figura 11 - Simulação em <i>software</i> do circuito de monitoramento.	36
Figura 12 - Circuito de monitoramento da velocidade do motor.	38
Figura 13 - a) <i>Encoder</i> acoplado ao rotor do motor. b) Detalhes do acoplamento mecânico.	39
Figura 14 - Esquema da PCI com os componentes do tacômetro.	40
Figura 15 - Esquema da PCI sem os componentes do tacômetro.	40

## *Lista de Tabelas*

Tabela 1 - Características básicas do microcontrolador PIC16F877A [12]. .....	23
Tabela 2 - Síntese dos resultados obtidos (60 RPM).....	39
Tabela 3 - Especificações elétricas do <i>encoder óptico incremental</i> [26]. .....	44
Tabela 4 - Características mecânicas do <i>encoder óptico incremental</i> [26]. .....	44
Tabela 5 - Características ambientais do <i>encoder óptico incremental</i> [26]. .....	45

# *1 Introdução*

Desde o início da industrialização com o surgimento das primeiras máquinas a vapor na Inglaterra no século XVIII já existia a necessidade de desenvolver técnicas de medição de parâmetros tal como, por exemplo, determinar a pressão das caldeiras de modo a reduzir o número de acidentes em virtude das frequentes explosões. Assim nasceram os primeiros indicadores de pressão. Já nos anos 50, com o advento dos semicondutores e os primeiros passos da Eletrônica, surgem os instrumentos eletrônicos e analógicos [13]. Atualmente a importância de ter conhecimento das variáveis dos processos é indiscutível quando se trata da eficiência. Pressão, temperatura, velocidade e umidade são alguns dos vários parâmetros que compõem um processo e que precisam ser monitorados e controlados para que a eficiência do mesmo seja a máxima possível.

Com os crescentes avanços da ciência e a descoberta de novos materiais e suas propriedades também ocorreu um salto no número de parâmetros da natureza que podiam ser monitorados. Certos materiais tinham comportamentos que variavam de acordo com a mudança de um determinado fator do ambiente (temperatura, umidade ou luminosidade, por exemplo). A partir desses materiais surgiram os primeiros sensores. Normalmente o sensor é composto de algum material que muda algumas de suas características quando ocorre a mudança de uma condição específica do ambiente. Para converter essa mudança do ambiente em informação útil tal como um sinal elétrico é preciso um circuito elétrico, afinal o sinal de saída dos sensores não costuma ser de natureza elétrica. Transdutores por sua vez, recebem informações na forma de uma ou mais quantidades físicas, modifica se necessário, estas informações e fornece um sinal de saída resultante. Esse sinal por sua vez é condicionado e interpretado para ser convertido em informação útil [1].

Tendo o conhecimento sobre o comportamento das variáveis que giram em torno de um processo produtivo a próxima preocupação era o grau de confiabilidade de uma medida. A precisão e a exatidão de certas leituras e a necessidade de manter algumas variáveis constantes impulsionaram o crescimento da eficiência dos sensores e transdutores e a busca por novos materiais cada vez mais sensíveis à certos fatores do ambiente. Circuitos digitais, robótica e automação são ferramentas largamente utilizadas nos processos industriais e dependem de boas leituras dos parâmetros inerentes ao bom funcionamento da linha de produção [3]. Hoje um

operador de um sistema supervisorio pode confrontar os diversos parâmetros medidos e colocados à sua disposição ou ainda verificar o histórico do comportamento daquela variável ao longo do tempo. A vantagem é a melhoria da qualidade de vida dos operadores, uma vez que através de alguns monitores dentro de uma sala de operação o indivíduo tem conhecimento do comportamento de toda uma linha produtiva. Além disso qualquer desvio de operação que acarretaria prejuízo ou mesmo perda da qualidade do produto final são amenizados [14].

Todo o trabalho citado até agora se apoia sobre dois ramos primordiais da Engenharia Elétrica: Instrumentação e Eletrônica. A Instrumentação é um ramo da ciência particularmente interessado na otimização e eficiência dos processos industriais. Basicamente ela desenvolve e aprimora técnicas de medição, indicação, registro e controle de processos [15]. Já a Eletrônica é mais ampla, tratando de estudar os dispositivos eletrônicos e suas aplicações em circuitos [16]. O monitoramento da velocidade de um motor usando o dispositivo *encoder* é um exemplo prático dentro da instrumentação uma vez que o monitoramento e registro de variáveis no meio industrial é uma função inerente desse ramo. Por outro lado, o *encoder* necessita de um circuito que possa converter os dados que o mesmo retira do processo em informação útil, tarefa que só pode ser alcançada com o auxílio da eletrônica.

Uma variável cujo controle é de grande importância em uma vasta gama de processos é a velocidade angular, em especial o número de rotações por minuto (RPM) de peças girantes, tais como os eixos de motores e engrenagens. A unidade RPM embora não seja a unidade do Sistema Internacional (SI) para velocidade angular é uma unidade prática muito utilizada para máquinas rotativas diversas. No Sistema Internacional a unidade de velocidade angular é radiano por segundo (rd/s). Para a tarefa de leitura da velocidade de giro de peças são utilizados instrumentos chamados tacômetros. Também conhecidos como conta-giros esses instrumentos são responsáveis por fornecer o valor do número de rotações por minuto (RPM) de uma peça girante. No funcionamento interno dos tacômetros podem ser utilizados os *encoders* mencionados anteriormente. As aplicabilidades dos tacômetros são essenciais, e vão desde monitorar a velocidade de carros, motos, motores e geradores até o controle de velocidade de correias transportadoras [17].



## 1.1 Tacômetros

### 1.1.1 Introdução

Tacômetros são instrumentos de medição de velocidade angular de peças girantes tais como eixos de motor e engrenagens, também conhecidos como conta-giros [18].

A velocidade angular ( $\omega$ ) popularmente conhecida como rotação é a razão entre o deslocamento angular ( $\Delta\theta$ ) de uma peça girante e o tempo ( $\Delta t$ ) decorrido durante esse deslocamento, como mostra a Equação 1 [18].

$$\omega = \frac{\Delta\theta}{\Delta t} \quad (1)$$

Tacômetros podem ser classificados em inúmeros tipos dependendo do seu princípio de funcionamento. São facilmente encontradas aplicações com tacômetros dos tipos:

- Mecânicos;
- Eletromagnéticos;
- Ópticos.

### 1.1.2 Tacômetros Centrífugos

Dentre os tacômetros mecânicos com utilização em eixos verticais, o modelo mais comum e eficiente é o tacômetro centrífugo mostrado na Figura 1. Seu funcionamento se baseia na força centrífuga exercida sobre dois pesos rotativos articulados a um eixo rotatório. A medida que a velocidade angular aumenta o raio entre os pesos também aumenta comprimindo uma mola. A medida de compressão da mola pode ser lida e convertida em velocidade angular do eixo girante [19].

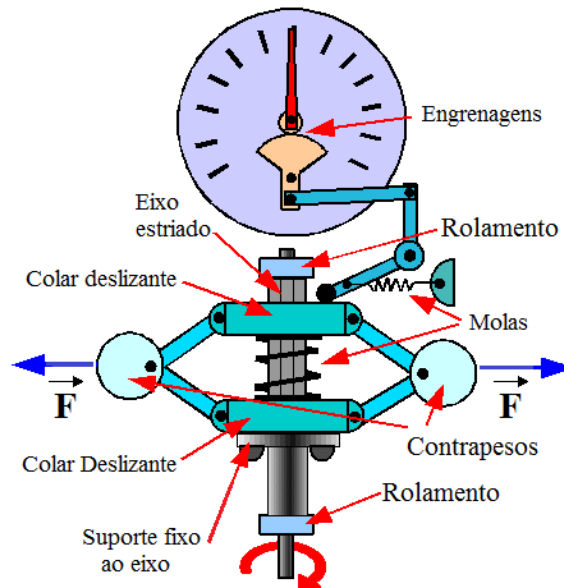


Figura 1 – Princípio de Funcionamento do Tacômetro Centrífugo [19].

### 1.1.3 Tacômetro Eletromagnético

Também conhecidos como tacogeradores os tacômetros eletromagnéticos são minigeradores de energia, se baseando no movimento rotatório de uma bobina dentro de um campo magnético formado por dois ímãs permanentes. A Figura 2 mostra o conceito de funcionamento desse tipo de tacômetro. O resultado da movimentação da bobina em meio ao campo gerado pelos ímãs é a produção de uma força eletromotriz induzida nas extremidades da bobina (Lei de Faraday), cujo valor tem relação com a velocidade angular da peça girante [18].

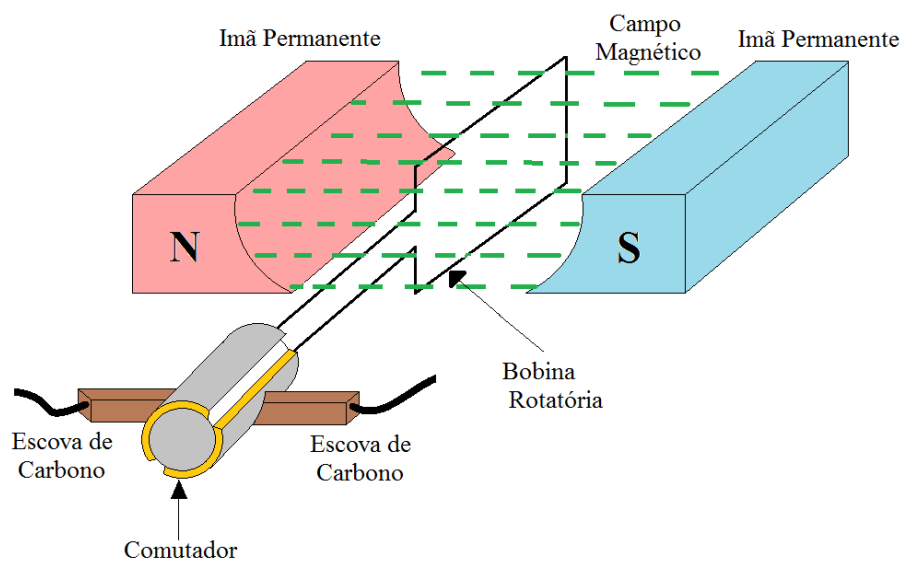


Figura 2 – Princípio de Funcionamento do Tacômetro Eletromagnético [18].

### 1.1.4 Tacômetro Óptico

Tacômetros ópticos são os mais utilizados por abranger uma larga faixa de níveis de velocidade (em RPM). Seu princípio de funcionamento se dá através da rotação de um disco composto por faixas claras e escuras e um feixe de luz infravermelha entre um par emissor-receptor óptico. O disco é posicionado entre o par emissor-receptor de tal maneira que a passagem do feixe só é permitida quando ele incide sobre a faixa clara do disco. Quando o sinal infravermelho atinge a faixa escura do disco não ocorre a recepção do sinal infravermelho pelo receptor óptico. Alguns modelos utilizam discos opacos com perfurações ou até mesmo rodas dentadas' como mostra a Figura 3. O resultado desse arranjo óptico é um sinal composto por uma série de pulsos semelhante à uma onda quadrada. A informação contida na frequência desses pulsos tem relação com a velocidade angular da peça girante monitorada. A vantagem do uso desse tipo de tacômetro está no fato do seu arranjo não oferecer praticamente nenhuma carga extra ao eixo da peça girante monitorada. A única carga desse sistema seria apenas a inércia do disco que pode ser desprezada [18].

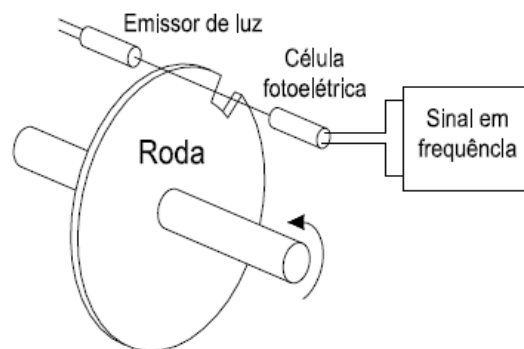


Figura 3 - Princípio de funcionamento do tacômetro óptico [18].

## 1.2 Encoders

### 1.2.1 Introdução

Também conhecidos como transdutores de posição angular os *encoders* são dispositivos eletromecânicos capazes de converter movimentos lineares ou angulares em pulsos elétricos digitais que, por sua vez, podem ser interpretados e convertidos em informações úteis tais como a mensuração de velocidade ou posição de eixos girantes ou ainda o número de voltas por minuto (RPM) [18].

Os *encoders* são de grande importância na área da automação principalmente nos processos que dependem de dados sobre a velocidade, posição de uma peça que gira ou o

sentido de rotação da mesma. Esse tipo de informação é de vital importância para o controle de certos equipamentos automatizados inclusive, em certas situações, o próprio funcionamento do *encoder* é monitorado. São inúmeras as aplicações onde o conhecimento da posição de peças móveis influencia na eficiência do controle de um equipamento e conseqüentemente na qualidade do produto final de um determinado processo [18].

Como vantagens desse transdutor podem ser citadas a sua grande durabilidade devido ao pouco desgaste de suas partes, sua excelente precisão e a necessidade de um circuito relativamente simples para o condicionamento do seu sinal de saída. Entretanto eles requerem um bom acoplamento mecânico às partes móveis do equipamento que se deseja monitorar. Além disso, são necessárias boas condições do ambiente para o seu bom funcionamento, tais como ausência de poeira e/ou outros resíduos. Temperatura e umidade controladas também são determinantes para o bom funcionamento do *encoder* [20].

A maioria dos *encoders* são do tipo *ópticos*, ou seja, utilizam um par transmissor-receptor óptico para a produção dos pulsos digitais como saída (detecção fotoelétrica) [17]. Para o trabalho em questão será tratado apenas sobre os *encoders ópticos*.

Quanto à diversidade desses equipamentos pode-se classificá-los em *encoder regular*, *encoder regular defasado* (ou *incremental*) e os *encoders absolutos*. Os *encoders regulares* são os modelos mais simples, enviando à saída um trem de pulsos digitais toda vez que o eixo do *encoder* gira. A partir da contagem de pulsos do sinal de saída pode-se determinar a velocidade do eixo ou ainda a posição do mesmo a partir de um referencial. Infelizmente esse tipo de *encoder* detecta apenas a mudança de posição do eixo e não a posição real que ele se encontra. Já os *encoders incrementais* possuem 2 faixas de ‘claros e escuros’ defasadas de modo que os sinais de saída gerados também são defasados. Sua vantagem é que, além de ser possível determinar a velocidade de um eixo, também é possível determinar o sentido de rotação. O último tipo são os *encoders absolutos* capazes de fornecer informações sobre velocidade, sentido de rotação e posição real do eixo girante. Eles são dotados de um circuito de codificação em binário onde todas as possíveis posições de parada do *encoder* tem um código único. Dessa forma é possível saber o ponto exato onde o eixo está posicionado bem como o curso final do mesmo [17] [20]. Como foco deste trabalho será abordado apenas os *encoders ópticos incrementais*.

### 1.2.2 Funcionamento

Basicamente os *encoders ópticos* são constituídos por um disco codificado acoplado à um eixo, um ou mais transmissores (LEDs infravermelhos) e um ou mais receptores (fotodiodos ou fototransistores) como mostra a Figura 4. O disco codificado é feito de plástico transparente onde estão gravadas tiras escuras que correspondem à codificação digital de cada posição. A leitura do movimento do disco codificado é feita por um par transmissor-receptor acoplado de forma que o disco fique entre esse conjunto. Assim as partes claras e escuras passam diante do par transmissor-receptor, ora impedindo que o seu sinal chegue ao receptor ora permitindo a passagem desse sinal. Esse fenômeno é conhecido como detecção fotoelétrica [17].

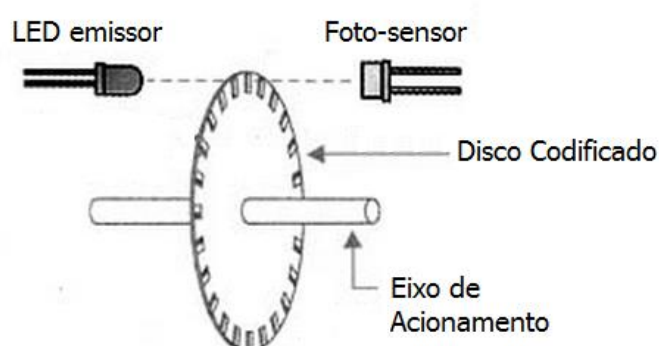


Figura 4 - Composição básica de um *encoder óptico* [17].

O eixo do *encoder* é acoplado à alguma peça móvel do equipamento que se deseja monitorar seu movimento (tal como um eixo de motor, por exemplo). A movimentação do eixo é convertida em um trem de pulsos elétricos através da detecção fotoelétrica, onde uma série de pulsos são gerados pela passagem de luz ou sinal infravermelho pelas ‘partes claras’ do disco codificado. O receptor detecta o sinal enviado pelo emissor e também a ausência desse sinal gerando assim pulsos digitais (0 e 1) como mostra a Figura 5. Os pulsos gerados se assemelham à uma onda quadrada e normalmente sua amplitude varia entre 0 e 5V, na maioria dos modelos de *encoders* disponíveis no mercado [17] [20].

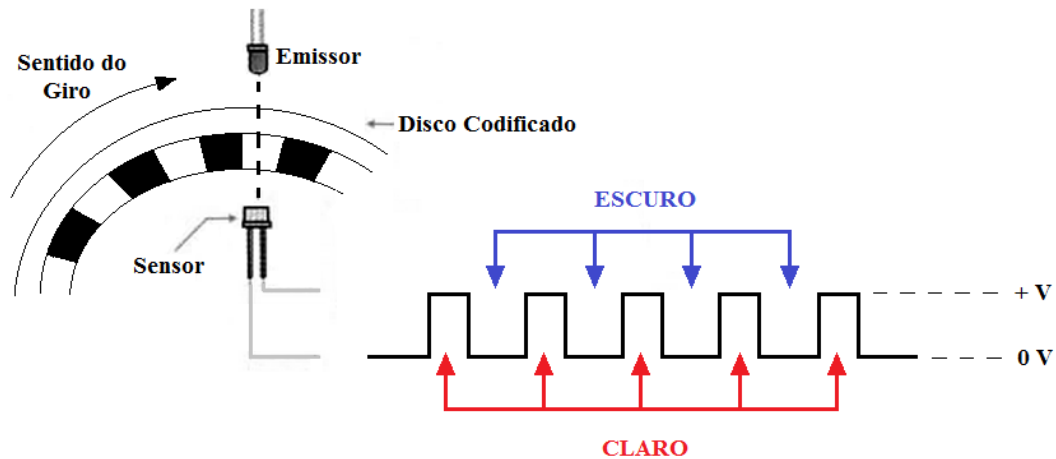


Figura 5 - Funcionamento do *encoder*: pulsos produzidos [17].

Conforme dito anteriormente, o sinal de saída dos *encoders* ópticos incrementais deve ser interpretado e convertido em informação útil para o usuário. Para realizar essa tarefa de converter o sinal do transdutor em uma posição de um eixo ou o número de rotações por minuto de um motor é necessário um circuito para aquisição e processamento dos dados e disponibilização dos mesmos ao usuário. O objetivo desse trabalho é construir esse circuito de monitoramento.

### 1.3 Microcontrolador PIC

#### 1.3.1 Introdução

Os microcontroladores PIC (do inglês *Programmable Interface Controller*) são uma extensa família de microcontroladores fabricados pela Microchip Technology Inc. Como o próprio nome diz, eles são Controladores de Interface Progrável com uma arquitetura Harvard, alta velocidade de processamento de dados e uma vasta variedade de periféricos internos. Os microcontroladores são praticamente um sistema operacional completo, dotados de CPU, memória RAM, *flash*, etc. Os PICs da família 16 e 18 processam dados de 8 bits sendo os da família 18 mais rápidos em termos de processamento. Já os PICs da família 24 são de 16 bits e são indicados para aplicações que exigem maior poder de processamento da CPU. Podendo ser programados em *Linguagem C* ou *Assembler* seus pinos possuem funções diversas: entradas/saídas digitais ou analógicas, comunicação serial padrão RS232, comunicação USB, geradores de pulsos PWM, contadores de pulsos externos (interrupção externa), *timers* e conversores analógicos/digitais dentre outras funcionalidades e periféricos. A quantidade e os tipos de funções e periféricos disponíveis, bem como o número de pinos depende do modelo do

PIC. A Figura 6 mostra um microcontrolador PIC modelo 16F877A de encapsulamento PDIP [21]. Esse microcontrolador será utilizado no projeto em questão.

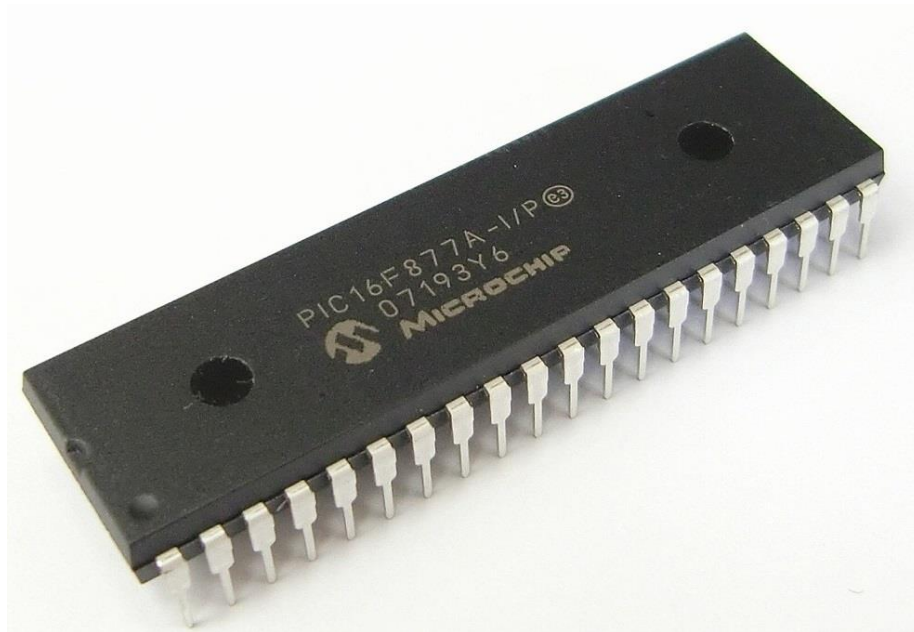


Figura 6 - Microcontrolador PIC16F877A [21].

### 1.3.2 PIC16F877A

Com 40 pinos o PIC16F877A possui processamento de dados de 8 bits podendo trabalhar em uma frequência de até 20MHz. Possui 7 pinos de alimentação e controle e 33 *I/Os* (portas que podem ser configuradas como entradas ou saídas). Esses 33 pinos de *I/Os* são agrupados em *PORTs* com no máximo 8 pinos cada [22]. A Tabela 1 mostra quais e quantos são os periféricos disponíveis dentro do PIC16F877A, bem como algumas de suas características de operação.

Tabela 1 - Características básicas do microcontrolador PIC16F877A [12].

Frequência de Operação	DC – 20MHz
Resets (and Delays)	POR BOR (PWRT, OST)
Interrupções	15
<i>I/Os</i>	Port A, B, C, D e E (total de 33 pinos)
<i>Timers</i>	3 ( <i>timer 0</i> , <i>timer 1</i> e <i>timer 2</i> )
Captura/Comparação/módulo PWM	2 ( <i>ccp1</i> e <i>ccp2</i> )
Comunicação Serial	MSSP, USART

Comunicação Paralela	PSP
Módulo Analógico Digital (10 bits)	8 canais de entrada
Comparadores Analógicos	2

Dentre as inúmeras vantagens do PIC16F877A, está o fato dele ser regravável, ou seja, reprogramável. Dessa forma o usuário sempre pode modificar o código inserido e reprogramá-lo conforme a sua necessidade. A Figura 7 mostra como são divididas as portas do PIC16F877A. Entretanto vale ressaltar que diversas portas do mesmo possuem até mais de uma função cabendo ao usuário decidir como utilizá-las através de configuração por linha de código.

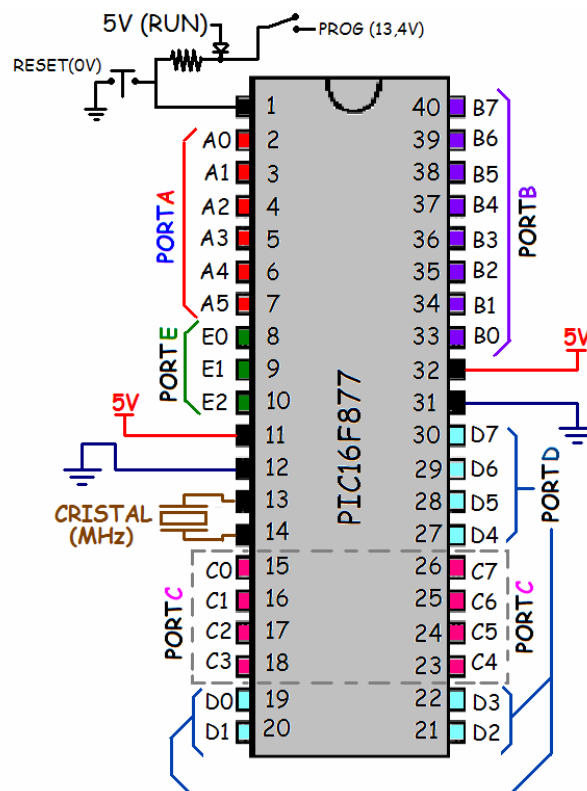


Figura 7 - Pinagem do Microcontrolador PIC16F877A [23].

## 1.4 Display LCD

Em uma infinidade de situações é interessante que o indivíduo tenha conhecimento do valor de parâmetros importantes de um sistema: temperatura, pressão, velocidade, tempo, altitude, vazão, produção diária, dentre outras. Sendo assim deve-se procurar uma boa forma de apresentar os dados, pois a partir do conhecimento desses dados o usuário daquele sistema pode tomar decisões, seja modificando os parâmetros caso eles estejam em desacordo com o



funcionamento normal do processo, ou apenas constatando que tudo está funcionando bem e que nada deve ser modificado [24].

Da mesma maneira que um computador necessita de um monitor para exibir as informações processadas aos seus usuários, o PIC também necessita de uma interface de saída para disponibilizar informações quando é requisitado. Atualmente uma maneira muito utilizada tanto por profissionais quanto por hobbystas em programação orientada à microcontroladores PIC são os displays LCD (do inglês *Liquid Crystal Display*). Em substituição aos antigos e limitados displays de sete segmentos, os displays LCD podem representar uma vasta gama de caracteres, inclusive alguns modelos têm a capacidade de representar elementos gráficos. Os modelos mais comuns de módulo LCD são os do tipo *caracter* onde é especificado o número de linhas por colunas. A Figura 5 mostra um módulo LCD do tipo *caracter* [24].



Figura 8 - Módulo LCD [25].

No mercado podem ser encontrados módulos LCD providos de uma iluminação de fundo conhecida como *LED backlight* o que facilita a visualização dos dados durante à noite. Também pode ser encontrado um pino de ajuste da intensidade de luz do display. Normalmente os displays LCD possuem um controlador próprio que permite sua interligação com outros circuitos e componentes através dos seus pinos (conexão com o microcontrolador, por exemplo). Além disso, há a necessidade de um protocolo de comunicação entre o módulo LCD e o circuito conectado a ele, de modo que possibilite que o LCD possa receber bytes de instrução e dados pelo sistema do usuário [24].

## 2 *Materiais e Métodos*

Visando a confecção do circuito de monitoramento da velocidade de um motor usando um *encoder óptico incremental* o primeiro passo foi definir uma sequência de tarefas para atacar o problema. Como primeira tarefa, o objetivo era obter alguma maneira de associar a velocidade de giro de um eixo com alguma grandeza física mensurável que posteriormente poderia ser convertida em um sinal elétrico para ser tratada computacionalmente. Para essa tarefa está à disposição um transdutor conhecido como *encoder óptico incremental* que conforme dito anteriormente gera uma resposta em frequência como saída que tem relação com a velocidade do eixo acoplado a ele.

O segundo passo é tratar o sinal oriundo do *encoder* e condicioná-lo, ou seja, de alguma forma fazer com que esse sinal seja lido por um circuito microcontrolado. Com o propósito de microprocessar os dados, utilizou-se um microcontrolador PIC16F877A. A implementação do algoritmo do circuito de monitoramento foi desenvolvida na Linguagem C de programação, por ser de mais simples entendimento. Para dar continuidade ao estudo das funções contidas dentro do PIC, em especial o uso dos *Timers* e as interrupções foi necessário, confeccionar um fluxograma do problema com o intuito de simplificar a tarefa da implementação. A tarefa de temporização usando a interrupção do *timer 0* associada ao *clock* e a tarefa de contagem de pulsos usando a leitura de clock externo assíncrono do *timer 1* que terá destaque especial e um maior detalhamento dentro do algoritmo por serem a base do funcionamento do circuito.

A terceira tarefa é compilar o código em um compilador de algoritmo em linguagem C e em seguida simular o comportamento do circuito microcontrolado. Durante a simulação do circuito completo será avaliado o funcionamento do microcontrolador PIC com o algoritmo já inserido dentro da sua memória.

Dado que a simulação em computador obteve resultado positivo a quarta e última etapa será o teste do circuito em *protoboard* e confecção da placa de circuito impresso (PCI). A simulação em *protoboard* é de vital importância para se corrigir erros no circuito e ajustar o código antes da produção do circuito definitivo.

Durante todas as etapas descritas as informações pertinentes sobre o funcionamento de cada elemento ou função utilizada será apresentada da forma mais simples e coerente possível.

## 3 *Resultados e Discussões*

### 3.1 *Aquisição dos dados via encoder*

A primeira tarefa é de algum modo converter o movimento rotatório de uma peça em uma informação elétrica para ser processada e condicionada em informação útil pelo circuito de monitoramento. Para essa tarefa foi proposto a utilização de um *encoder óptico incremental* que, conforme explicado anteriormente, gera como saída um trem de pulsos digitais que fornecerá informação sobre velocidade rotatória da peça (em RPM) de acordo com a Equação 2.

$$rotação_{RPM} = \frac{60 * (n_{PULSOS})}{resolução} \quad (2)$$

A equação acima tem como variáveis o número de pulsos gerados pelo *encoder* ( $n_{pulsos}$ ) e a resolução ou PPR (pulsos por revolução) desse transdutor que é na verdade o número de ‘faixas claras’ do seu disco codificado. Dessa forma quando o disco executa 1 giro completo o número de pulsos do sinal de saída é igual ao número de faixas claras que, por sua vez, é igual a resolução desse transdutor. A contagem de pulsos deverá ocorrer num intervalo de tempo de 1 segundo por um motivo simples: Nesse intervalo de tempo a razão entre o número de pulsos e a resolução do *encoder* será igual a velocidade angular, em radianos por segundo, da peça girante monitorada. Como 1 minuto tem exatamente 60 segundos basta apenas multiplicar o valor de rotação encontrado (em rd/s) por 60 para obter a rotação da peça em RPM (rotações por minuto). É claro que a base de tempo não precisaria ser exatamente 1 segundo, entretanto, a vantagem de usá-la é a possibilidade de encontrar também o valor da velocidade angular em radianos por segundo, sem maiores complicações.

O modelo de *encoder óptico incremental* escolhido é o HTR (chave de código HTR-W2-1024-3-L-5V) conforme mostra a Figura 9. Este transdutor é produzido pela Hontko Co. e revendido no Brasil pela Metaltex® [26]. As informações mais pertinentes da sua ‘folha de dados’ podem ser encontradas no Apêndice A.



porta RB0/int. Estas são ferramentas poderosas que permitem ao microcontrolador ‘ler’ um sinal de entrada em uma determinada porta, bem como a identificação de bordas de subida ou descida de pulsos digitais. Essa função permitirá a identificação do número de pulsos que chega na porta do PIC dentro de um intervalo de tempo. O valor predeterminado foi o de 1 segundo, logo essa base de tempo também deverá ser configurada de modo que o microcontrolador execute a contagem de tempo desses intervalos. Essa tarefa pode ser realizada com a configuração dos *Timers* para trabalho com a interrupção interna associada ao cristal de *clock* como será tratado posteriormente com mais detalhes na programação do PIC.

O uso das interrupções dos *timers* é uma boa prática de programação, pois o funcionamento desse recurso independe do processamento da CPU, podendo ambos funcionar simultaneamente. Dessa forma é interessante que no início da compilação do programa tanto a contagem de tempo quanto a contagem dos pulsos digitais iniciem praticamente ao mesmo tempo. Passado o tempo de 1 segundo, o número de pulsos é armazenado, faz-se o cálculo da rotação em RPM e o valor é disponibilizado na interface de saída (display LCD do tipo *character* 16x2). Todo o processo se repete sempre que o *timer* temporiza um intervalo de 1 segundo após o evento anterior. O display LCD foi configurado de modo que represente até 2 casas decimais depois da vírgula. Desse modo até frações de centésimos de RPM podem ser representados no LCD.

### 3.2.2 Fluxograma

Para implementar o algoritmo do tacômetro em Linguagem C é interessante que se determine um esboço do funcionamento do algoritmo através de um diagrama lógico conhecido como fluxograma. Nesse fluxograma o objetivo é tentar retratar a lógica do programa afim de tornar mais fácil a sua implementação. A Figura 10 mostra os aspectos mais pertinentes do funcionamento do tacômetro:

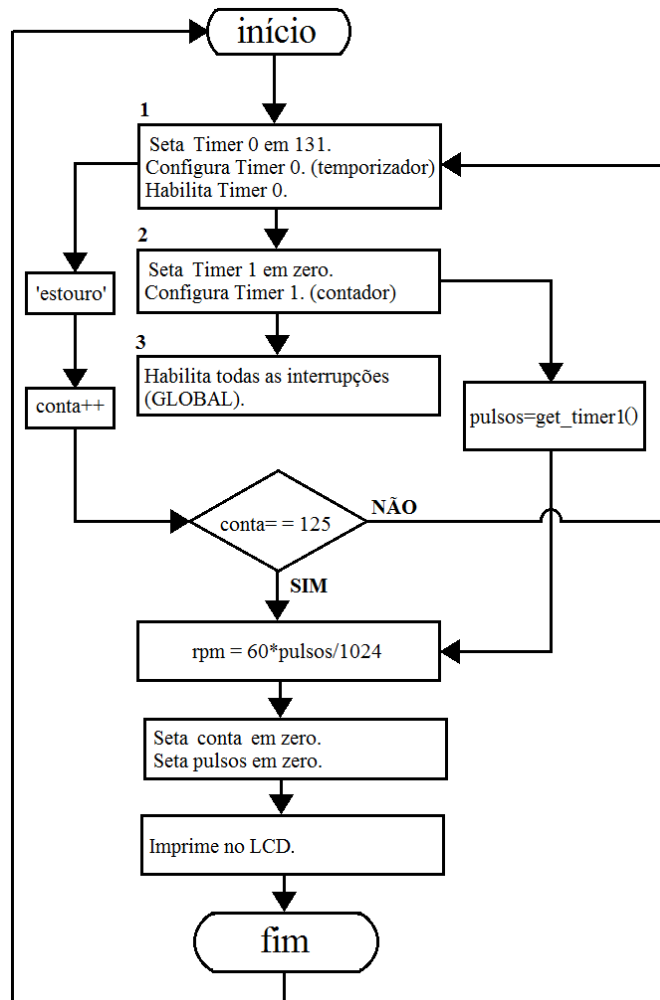


Figura 10 - Fluxograma do algoritmo.

### 3.3.3 Compilador do algoritmo em Linguagem C

O ambiente de simulação usado para compilação do algoritmo em Linguagem C em um código de máquina (sequência lógica de endereços de memória e instruções internas próprias do microcontrolador) foi o compilador C da CCS (Custom Computer Services Inc.) versão *demo* de 30 dias. Essa versão possui algumas restrições tais como o tamanho máximo de 2K de código de programa e poucos modelos de PIC para trabalho (suporta o PIC16F877 na versão *demo*) [29]. O propósito da compilação é verificar o correto funcionamento do código antes da gravação no microcontrolador. É importante salientar que o compilador acusa erros de sintaxe dentro do código (erros de digitação ou uso incorreto de comandos), mas não identifica erros de semântica. Os erros de semântica são difíceis de identificar e dependem de conhecimentos profundos da linguagem de programação, do funcionamento dos periféricos internos do PIC e da lógica do problema definida no fluxograma.

### 3.2.4 Configuração do *Timer 0*: interrupção interna

O trabalho com os *timers* do PIC é uma das tarefas mais complexas dentro do estudo dos microcontroladores, pois requer relacionar o estado de um temporizador interno do componente com uma dimensão real (tempo). Programá-lo requer conhecimentos avançados em linguagem de programação e o funcionamento dos seus periféricos internos [27].

O *timer 0* é um contador de 8 bits (0 a 255) cujo incremento ocorre a cada ciclo de máquina e depende da configuração do *prescaler* e da frequência do *clock* (dada pelo cristal oscilador). Toda vez que o *timer 0* chega ao valor de 255 ele reinicia a contagem de pulsos do zero. Na passagem do registro 255 para o registro 0 ocorre a interrupção do *timer 0* conhecida como ‘estouro’. Como primeira tarefa será configurado o *timer 0* do PIC16F877A para que ocorra um evento predeterminado a cada 1 segundo. Segundo [27] um temporizador com estouro a cada um segundo pode ser implementado da seguinte maneira:

Suponha que o cristal de oscilação acoplado ao microcontrolador tenha frequência de oscilação ( $f_{osc}$ ) igual à 4MHz e que, nas configurações do *timer 0* no PIC será utilizado um *prescaler* de 64.

Dessa maneira os incrementos (pulsos) do *timer 0* ocorrerão numa frequência de 15625Hz de acordo com a Equação 3:

$$f_{entrada} = \frac{f_{oscilação}}{4 * (prescaler)} = \frac{4.000.000}{4 * 64} = 15625Hz \quad (3)$$

E o tempo entre os incrementos ( $T_{incremento}$ ) do *timer 0* será:

$$T_{incremento} = \frac{1}{f_{entrada}} = \frac{1}{15625} = 64\mu s \quad (4)$$

Ou seja, um incremento surge 64 microsegundos após o incremento anterior. Após 255 incrementos ocorre o estouro do *timer 0*, logo o tempo entre cada estouro ( $T_{estouro}$ ) é:

$$T_{estouro} = 255 * T_{incremento} = 255 * 64\mu s = 16,320ms \quad (5)$$

Como se deseja um tempo de 1 segundo, o número de estouros que ocorrerá dentro de um intervalo de tempo de 1 segundo é:

$$n_{estouros} = \frac{1s}{16,320ms} = 61,274 \quad (6)$$

O número de estouros calculados dentro do intervalo de 1 segundo não é um valor inteiro o que é um problema. Uma possível solução seria iniciar o registro do *timer 0* em 131. Dessa forma o número de incrementos entre cada estouro seria exatamente 125 (256 - 131), ou seja, o registrador iniciaria a contagem em 131, ocorreria o estouro em 255 e o novo ciclo se iniciaria em 131[27]. Com 125 incrementos entre cada estouro o intervalo de tempo entre um estouro e outro ( $T_{novo}$ ) é:

$$T_{novo} = 125 * 64\mu s = 8ms \quad (7)$$

Assim o novo número de estouros dentro do intervalo de tempo igual à 1 segundo será:

$$n_{estouros\_novo} = \frac{1s}{8ms} = 125 \quad (8)$$

Logo ocorrerá 125 estouros num intervalo de tempo de 1 segundo. Essa informação é de vital importância para a programação do microcontrolador durante o projeto em questão. Conforme dito anteriormente a estratégia adotada para a programação do tacômetro será utilizar a leitura do número de pulsos dentro de um intervalo de 1 segundo para determinar a frequência desses pulsos e daí obter o número de rotações por segundo da peça (eixo) acoplada ao *encoder óptico*.

Em termos de programação será habilitado a interrupção interna do *timer 0* na função principal do algoritmo do tacômetro, adotando um cristal de *clock* de 4MHz e *prescaler* igual à 64 configurados. Após a habilitação da interrupção do *timer 0* e configuração para trabalho com o *clock* interno de 4MHz e *prescaler* de 64 é necessário ‘setar’ o valor do registrador do *timer 0* em 131. Dessa forma, toda vez que o código é compilado a contagem se inicializará do 131. A próxima etapa será chamar a função de interrupção do *timer 0* (função *void* fora da função principal). Essa função tem um funcionamento muito simples: inicializa-se em zero uma variável local do tipo inteira chamada ‘*conta*’. Essa variável define o número de estouros do *timer 0* e será incrementada toda vez que ocorrer um estouro. Conforme dito anteriormente, o registrador do *timer 0* inicializa no valor de 131, e após 125 interrupções tem-se a certeza que a variável *conta* tem valor igual a 125 e se passaram exatamente 1 segundo. Nesse momento é importante resetar o valor da variável ‘*conta*’ para o próximo evento (passagem de 1 segundo), resetar também a variável pulsos e então executar a ação desejada, que neste caso, é usar o valor



da interrupção externa do *timer 1* (configurado para receber *clock* externo assíncrono), calcular o valor de rotações do *encoder* em RPM dentro daquele intervalo de 1 segundo que passou e imprimir o resultado na tela do display LCD.

Dentro da função principal a interrupção interna do *timer 0* é habilitada, o *prescaler* é configurado para 64 divisões e o valor do registrador do timer 0 é inicializado em 131. Ainda dentro da função principal é colocado uma função do tipo *while* cuja condição é sempre verdadeira. O objetivo dessa função *while* é permanecer o código ‘rodando’ enquanto o microcontrolador estiver ligado (*looping infinito*). Neste *while* ocorre a impressão do valor da variável ‘rpm’ no módulo display LCD.

Segundo [28] é possível aumentar a precisão nos *timers* do PIC utilizando uma técnica conhecida como *acúmulo de erros*. Basicamente a técnica consiste em implementar no algoritmo um contador de software. Esse contador será decrementado a cada ciclo de interrupções do *timer*, e uma vez que seu valor atinja um certo limite predeterminado, será subtraído o valor da frequência de entrada do valor do contador de modo que seja mantido o resto dessa operação dentro do registrador. Essa técnica é eficaz, porém não será utilizada, pois o intervalo de tempo entre os eventos não é constante, ou seja, a passagem de 1 segundo entre os eventos estaria comprometida.

### 3.2.5 Configuração do *Timer 1*: interrupção externa

Uma tarefa muito importante dentro no projeto do algoritmo do circuito de monitoramento da velocidade do motor é a leitura do sinal do *encoder óptico*. Conforme dito anteriormente o sinal do *encoder* é composto por pulsos digitais e se assemelha à uma onda quadrada. O objetivo é identificar o número de pulsos dentro de um intervalo de 1 segundo, mas para isso é necessário ler um sinal externo ao microcontrolador PIC. Para identificar uma série de pulsos na porta do microcontrolador será utilizado a interrupção externa do *timer*, onde o registrador desse *timer* terá a importante tarefa de contar o número de pulsos.

O *timer 1* do PIC16F877A é um contador de 16 bits. Três das preocupações a serem consideradas no projeto são:

1. Qual é a frequência máxima de pulsos que o registrador do *timer 1* conseguirá contar?
2. Qual é a frequência máxima que o *encoder óptico* pode operar?
3. Qual é o número máximo de rotações por minuto do motor?

Respondendo à primeira questão, como o contador é de 16 bits o *timer 1* pode contar de 0 até 65535 pulsos antes que ocorra o estouro (interrupção). O problema é a frequência de leitura desses pulsos. Tomando a base de tempo o *clock* igual à 4MHz utilizado no *timer 0* para a confecção do temporizador de 1 segundo, o número de incrementos dentro desse intervalo de tempo dependerá da frequência do oscilador (*clock*) e da configuração do *prescaler*. Será adotado um *prescaler* dividido por 1 para que a frequência de leitura do sinal externo seja a maior possível. Assim a frequência de entrada (leitura) é dada pela equação:

$$f_{entrada} = \frac{f_{oscilação}}{4 * (prescaler)} = \frac{4.000.000}{4 * 1} = 1\text{MHz} \quad (9)$$

Esse resultado diz que a frequência de leitura será 1 milhão. Entretanto esse valor seria o correto se o processamento do PIC estivesse em condições normais e executasse apenas 1 instrução do código (o microcontrolador leva 1 microsegundo para ‘ler’ cada linha de instrução do código). Como o registrador do timer 1 tem 65536 posições pode-se estimar que no intervalo de 1 segundo a frequência máxima de trabalho desse *timer* é:

$$f_{máx} = (intervalo) * (n_{posições}) = 1s * 65536 \cong 0,065\text{MHz} \quad (10)$$

Dessa forma a frequência máxima do *timer 1* é cerca de 15 vezes menor que a frequência de entrada. De posse de  $f_{máx}$  é possível obter uma estimativa do valor máximo de rotações por minuto que o tacômetro é capaz de ler:

$$\omega_{máx} = \frac{60 * n_{posições}}{1024} = \frac{60 * 65536}{1024} = 3829,45\text{RPM} \quad (12)$$

Outra questão ainda não avaliada, mas a ser considerada na definição do limite de trabalho do tacômetro é o limite de operação do *encoder*. Observando a Tabela 1 sobre as características elétricas disponíveis na folha de dados do equipamento nota-se que a máxima resposta do *encoder óptico* utilizado varia numa faixa entre 10kHz e 100kHz dependendo do valor da sua resolução. O valor da frequência máxima do sinal de entrada levando em consideração a limitação de trabalho do microcontrolador é de aproximadamente 65,5kHz. Valor aceitável, compreendido dentro da faixa de resposta máxima do transdutor especificado.

Em termos computacionais, o primeiro passo é habilitar o *timer 1* na função principal para trabalhar com interrupção externa (*clock* externo assíncrono) e o *prescaler off* (divisão por

1). O *timer* setado em zero através da função *set\_timer\_1()*. Essa rotina e a inicialização do módulo LCD encontram-se dentro da função *main()*.

Em seguida, a função *void* da interrupção é habilitada de modo que toda vez que um pulso é lido na porta do PIC uma variável '*pulsos*' é incrementada em 1.

Essa contagem é feita até que ocorra o estouro do *timer 0* (temporizador de 1 segundo). Daí o valor da variável '*pulsos*' dentro daquele intervalo é lido e faz-se o cálculo do número de rotações por minuto como explicado anteriormente. O módulo LCD é inicializado na função principal, mas as configurações de escrita, posição do cursor, definição do número de casas a ser escrito estão todas dentro do laço *while (true)*.

É importante salientar que na descrição do funcionamento do algoritmo foram ignoradas informações importantes a respeito do código tais como a introdução do cabeçalho do PIC16F877A (*#include<16F877A.h>*), as funções de configuração do mesmo (*#fuses XT, PUT, NOWDT, NODEBUG, BROWNOUT, NOLVP, NOCPD, NOPROTECT*), a frequência de clock (*#use delay (clock=4000000)*), a configuração do LCD e principalmente, a definição das variáveis globais utilizadas. Essas pré-configurações são de vital importância e podem ser encontradas no código completo em Apêndice B.

### 3.3 Simulação do circuito

A terceira etapa seria simular o código e o circuito microprocessado em um *software* de simulação. Essa etapa é de grande importância no sentido que a simulação permite testar o circuito e atestar seu bom funcionamento mesmo antes dele ser construído. Caso a simulação dê resultados positivos restará o teste em *protoboard*. Concluindo a etapa de simulações e testes do circuito de monitoramento de velocidade, a quarta e última etapa será a confecção da placa de circuito impresso.

Em um ambiente de programação dentro do *software* Proteus/ISIS fabricado pela Labcenter Eletronics Inc. (versão *trial*) [30] foi possível simular o comportamento do circuito eletrônico composto pelo microcontrolador PIC16F877A, o display LCD (*character 16x2*), o circuito de *clock* (4MHz) e um sinal de oscilação na porta responsável pela interrupção externa do *timer 1* como mostra a Figura 11. Este ambiente de simulação permite que seja inserido o código (arquivo do tipo *.hex*) dentro do PIC. A Figura 10 mostra o circuito do tacômetro sendo simulado para um sinal externo com frequência de 1024 pulsos por segundo. De acordo com a Equação 2 era de se esperar que a velocidade em RPM indicada no display seria igual a 60, mas

é mostrado 60,29. Este erro de aproximadamente 0,48% era esperado devido aos valores da frequência de trabalho da CPU do microcontrolador. Até mesmo os relógios digitais mais precisos possuem uma margem de erro devido à imprecisão do cristal utilizado. A precisão pode ser melhorada trabalhando com um cristal de *clock* de frequência superior, mas devido à margem de erro ser pequena (inferior a 0,5%) esse procedimento não será necessário.

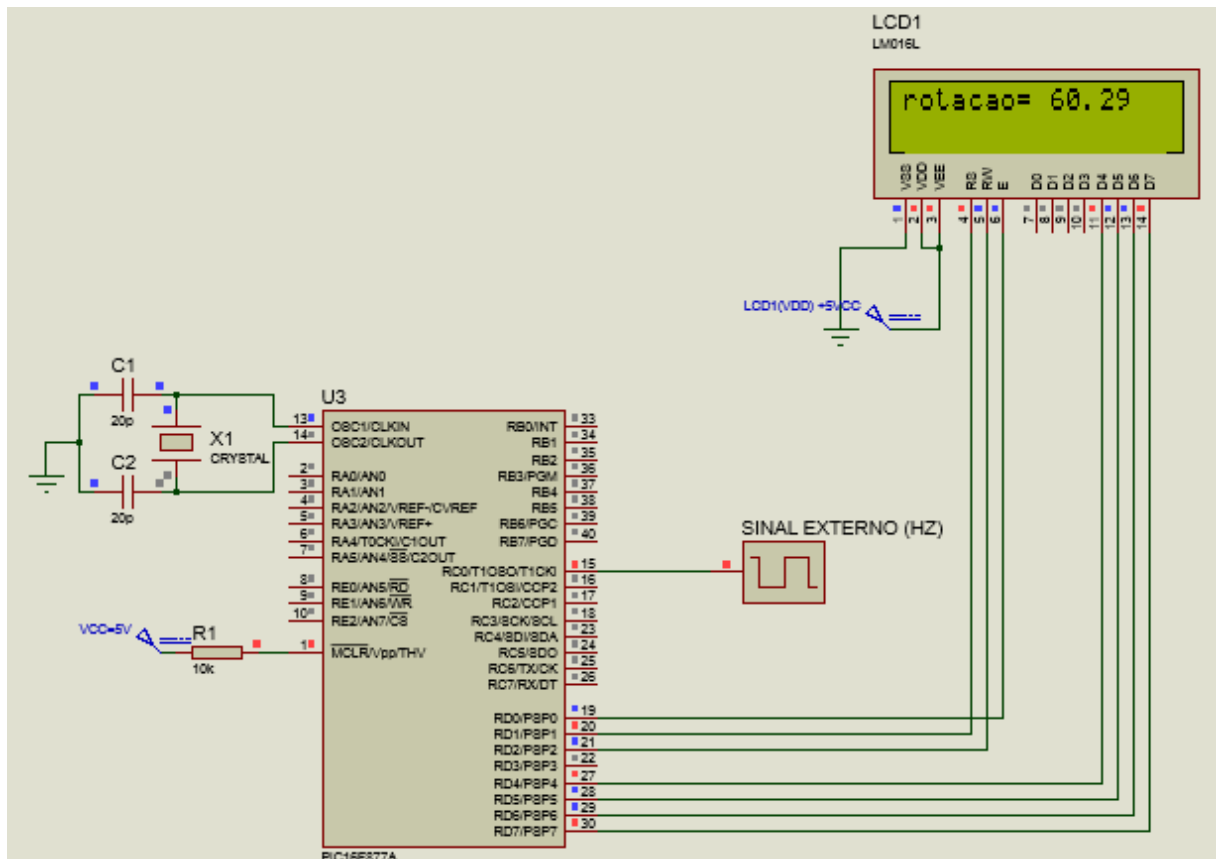


Figura 11 - Simulação em *software* do circuito de monitoramento.

A grande vantagem da utilização das interrupções do PIC é que elas não param o processamento da CPU para serem executadas. Por exemplo, na contagem de tempo: caso fosse utilizado a função *delay* (*polling*) ao invés da interrupção do *timer 0* associada ao cristal de *clock*, o processamento do código pela CPU do PIC ficaria estagnado naquele ponto onde está o *delay* durante todo o tempo estabelecido por essa função que, para este problema em questão, seria igual à 1 segundo. Durante a função *delay* as interrupções não param, mas o processamento das linhas de código do algoritmo em questão sim. Dessa forma todo intervalo de tempo igual à 1 segundo o microcontrolador suspenderia o processamento do algoritmo do tacômetro apenas para contar tempo o que em certas aplicações é ruim. Utilizar o *timer 0* para temporização de tempo ao invés da função *delay* retira essa ‘parada’ no processamento da CPU do PIC que por

sua vez pode utilizar aquele intervalo de tempo para executar outras tarefas, tais como o cálculo da velocidade de rotação em RPM e a escrita do valor na LCD. Logo a implementação de uma ‘nova tarefa’ dentro deste algoritmo não acarretaria travamento do PIC, tornando possível explorar mais funções do microcontrolador.

### 3.4 Desenvolvimento do circuito

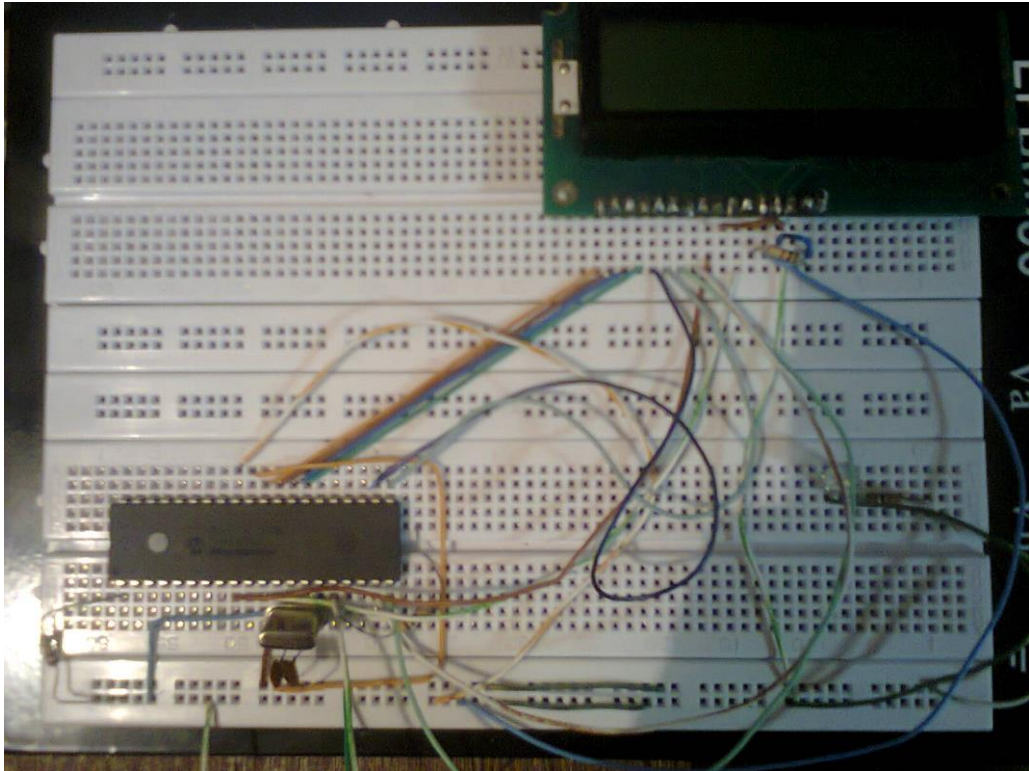
O circuito foi testado em uma matriz de contatos (*protoboard*) antes de ser desenvolvido o esquema da placa de circuito impresso. Esse ponto é de extrema importância quando se trata de circuitos experimentais.

O pino 1 do PIC16F877A é o *Master Clear* e ele requer 5V de alimentação para executar o programa. Segundo a folha de dados do microcontrolador é necessário um resistor de  $10K\Omega$  entre o pino do *Master Clear* e a alimentação [12].

O PIC16F877A requer um circuito oscilador externo gerador de *clock*. Foi escolhido um cristal oscilador de 4MHz, afinal toda a base de tempo do temporizador do *timer 0* na programação foi calculada para esse valor de cristal de oscilação. Ainda analisando a folha de dados do microcontrolador foi atestado que para a frequência de clock do cristal utilizado seria necessário um par de capacitores cuja capacitância esteja entre 15pF e 68pF [12]. Foi utilizado um capacitor cuja capacitância é 22pF. O modelo do circuito pode ser visto na Figura 10.

Foi estabelecido uma comunicação de 8 bits entre o PIC e o módulo display LCD. Dessa forma foram utilizados todos os 8 pinos de dados do display LCD e toda a *Port D* do PIC (pinos D0 à D7) conforme o código de configuração e inicialização do LCD inserido dentro do programa do tacômetro [12]. A comunicação entre LCD e PIC foi a maior alteração quando se compara a simulação em *software* e o teste do circuito em *protoboard*. Durante a simulação em *software* a comunicação era de 4 bits e os pinos de controle do LCD estavam configurados em portas do PIC diferentes das utilizadas na montagem final.

O sinal do *encoder* é transmitido para o pino 15 do PIC: Entrada para clock externo assíncrono [12]. O circuito de monitoramento da velocidade de um motor usando *encoder incremental* pode ser visto na Figura 12.



**Figura 12 - Circuito de monitoramento da velocidade do motor.**

Com o circuito montado o próximo passo foi testá-lo em laboratório. Para essa tarefa primeiramente foi testado como sinal de entrada do circuito um sinal quadrado variando entre 0 e 5V oriundo de um gerador de sinais. O uso do gerador de sinais possibilita que o usuário possa avaliar o funcionamento do circuito quando alguns dos parâmetros do sinal de entrada são modificados, tais como a frequência e a rampa de crescimento e decrescimento do sinal. Durante essa etapa foi constatado que o circuito é ideal para medir velocidades de até 64 RPM, ou seja, sinais cuja frequência seja próxima dos 1024Hz. O desvio padrão entre o valor esperado e o valor medido, quando a frequência do sinal de entrada era 1024Hz foi de aproximadamente 1,67%.

O mesmo circuito foi testado com um sinal de entrada oriundo do *encoder óptico incremental* HTR-W2-1024-3-L-5V. E novamente o valor de leitura foi satisfatório desde que a velocidade do motor permanecesse inferior à 64 RPM. O desvio padrão entre o valor 'setado' no inversor de frequência (exatos 60 RPM) e o valor medido pelo circuito (62,5 RPM no display) foi de 4,1%. A Tabela 2 mostra uma síntese dos resultados obtidos para uma velocidade de giro igual à 60 RPM.

Tabela 2 - Síntese dos resultados obtidos (60 RPM).

	Calculado	Simulado	Medido
Velocidade (RPM)	60	60,29	62,5
Erro (%)	-	0,48	4,1%

Para a realização do experimento foi utilizado um motor assíncrono, 4 pólos, ‘gaiola de esquilo’ conectado à um inversor de frequência onde era possível modificar o valor da velocidade do motor (em RPM). Ao eixo do motor foi conectado o *encoder*, alimentado com 5V de uma fonte contínua. Foi feita a aquisição do sinal da ‘fase A’ dele e inserida no circuito teste. A Figura 13 mostra o acoplamento do *encoder* ao rotor do motor.



Figura 13 - a) *Encoder* acoplado ao rotor do motor. b) Detalhes do acoplamento mecânico.

A limitação do projeto com relação a velocidade máxima de aproximadamente 64 RPM se deve ao fato de que no intervalo de 1 segundo o contador registra 61440 pulsos o que representa o produto do número de voltas durante o período (60 voltas) pela resolução do *encoder* (1024 pulsos por revolução). A medida que a velocidade de giro ultrapassa a marca de 64 RPM o número de pulsos registrados pelo contador se torna maior que o tamanho do *timer 1* que é igual a 65536 (16 bits). Nesse momento ocorre o estouro do *timer 1* responsável pela contagem de pulsos e a variável pulsos é inicializada em zero. A limitação do projeto está justamente no fato de que a Equação 2 não contabilizar corretamente o número de pulsos caso o *timer* contador de pulsos estoure. A solução para esse fato é apresentada no item 3.6 onde é sugerido melhorias para o projeto.

### 3.5 Projeto do circuito impresso

Com o auxílio do software versão *Freeware* EAGLE PCB Software Design® criando pela empresa CadSoft [30] foi possível o projeto do circuito impresso a partir do circuito desenvolvido até agora. O projeto da PCI com os componentes é mostrado na Figura 14. Por ser uma versão gratuita *software* EAGLE® possui uma limitação quanto ao valor da área da PCI limitando-a a uma região de no máximo 10 centímetros de comprimento por 8 centímetros de largura.

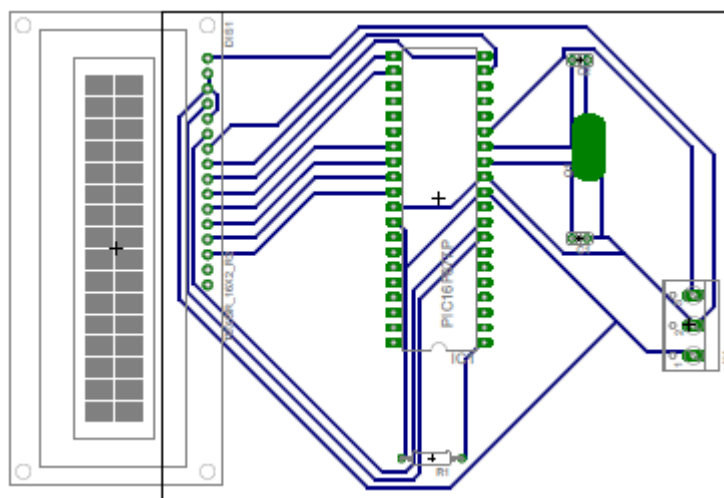


Figura 14 - Esquema da PCI com os componentes do tacômetro.

A Figura 15 mostra o esquema da PCI sem os componentes, ou seja, a 'trilha' do circuito de monitoramento de velocidade.

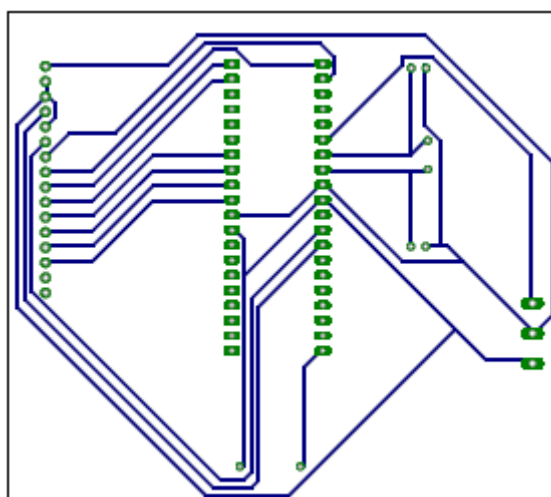


Figura 15 - Esquema da PCI sem os componentes do tacômetro.



### 3.6 Sugestões de melhorias

Para garantir a durabilidade do *encoder*, bem como o seu bom funcionamento e precisão das suas leituras é importante respeitar algumas condições de funcionamento sugeridas na folha de dados do equipamento. Ao analisar as ‘características ambientais’ do *encoder* é notado a preocupação quanto à temperatura de funcionamento do equipamento. A Tabela 3 do Apêndice A3 revela que em funcionamento, o *encoder* pode operar em uma temperatura de até no máximo 65°C. Dentre as inúmeras funcionalidades e periféricos, o microcontrolador PIC16F877A possui conversores A/D (Analógico Digital) e pinos de I/Os capazes de receber sinais analógicos. O apêndice C tem a sugestão de um algoritmo de um termômetro utilizando o LM35 (sensor de temperatura). Dentro da lógica do circuito de monitoramento da velocidade do eixo de um motor poderia ser acrescentado uma lógica que interrompa o funcionamento do *encoder* ou avise ao usuário que a temperatura limite do transdutor foi atingida.

Outra sugestão é utilizar uma nova estratégia de interpretação do sinal proveniente do *encoder* que possibilite o tacômetro medir velocidades acima dos 60 RPM. Um meio de aumentar a faixa de velocidades que o circuito é capaz de trabalhar seria adicionar mais um contador com o intuito de contar quantas vezes o *timer 1* (usado para contar pulsos) estourou. Para a implementação dessa estratégia seria necessário modificar a equação do cálculo de RPM e o algoritmo do tacômetro de modo que o produto do número de estouros ( $n_{ESTOUROS}$ ) pelo tamanho do *timer 1* seja acrescido ao valor da variável ‘pulsos’. A Equação 13 está ajustada para um algoritmo que tenha o funcionamento descrito anteriormente.

$$rotação_{RPM} = \frac{60 * (n_{PULSOS} + 65536 * n_{ESTOUROS})}{resolução} \quad (13)$$

Outra maneira de se aumentar a faixa de velocidades que o circuito de monitoramento é capaz de atuar seria o uso do módulo de captura, comparação e PWM do PIC conhecido como CCP. Com esse módulo seria possível medir o período de um sinal e obter o valor da frequência calculando o inverso do período.

Para atestar o bom funcionamento do tacômetro experimental é necessário aferí-lo, comparando seu desempenho com um produto já oferecido no mercado. Uma proposta de validar o bom funcionamento do circuito é fazer a leitura da velocidade de giro do motor usando um tacômetro óptico comercial e comparar os resultados com o circuito de monitoramento construído com o microcontrolador e o *encoder*.

Outra sugestão de implementação no projeto seria determinar o sentido de giro no motor baseado na comparação das saídas do *encoder*.

## 4 Conclusões

O projeto obteve resultado satisfatório pois o tacômetro se mostrou eficiente para leituras de frequências próximas à 1000Hz com uma margem de erro inferior à 5% tanto no processo de simulação em computador quanto em teste com o *proto-board*. O desenvolvimento do circuito utilizando as interrupções do microcontrolador PIC16F877A é um bom exemplo de prática de programação e exige um estudo apurado do funcionamento interno do PIC. O circuito de monitoramento de velocidade usando *encoder óptico incremental* poderá ser utilizado pelos alunos do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa para futuros estudos sobre controle de velocidade de motores e em aulas práticas da disciplina de Máquinas Elétricas.

É interessante ressaltar que o desenvolvimento do projeto é importante, pois utiliza equipamentos pouco utilizados dentro do laboratório dando a eles uma finalidade prática. A restrição quanto à velocidade de funcionamento foi de certa forma positiva, pois incentiva os próximos responsáveis por manusear o tacômetro desenvolvido a aprimorá-lo e torná-lo mais versátil.

Além da experiência em projetar circuitos microcontrolados o projeto foi uma boa experiência para estudo e aplicação de linguagem de programação C e também proporcionou contato com as áreas de Instrumentação e Eletrônica e *softwares* da área de Engenharia Elétrica.

## Referências Bibliográficas

- [01] Pereira, H. A. (2006). Desenvolvimento de um Sistema de Aquisição de Dados e Monitoramento Utilizando Microcontroladores. Monografia. Universidade Federal de Viçosa – 2006. 8
- [02] Pinheiro, P. B. (2011). Desenvolvimento de um Sistema de Instrumentação para Controle de Temperatura e Umidade via CLP. Monografia. Universidade Federal de Viçosa – 2011. 8
- [03] John, L. J. (2007). Monitoração Remota Via IP de Sensores Industriais. Trabalho de Conclusão de Curso. Universidade de São Paulo – 2007. 8
- [04] Dias J. E. A. (2010). Eletrônica, instrumentação e telemetria do veículo UFVbaja. Monografia. Universidade Federal de Viçosa – 2010. 8

- [05] Gomes, B. M. (2009). Aplicação de Sensores Descartáveis em uma Rede de Sensores Sem Fio para o Monitoramento do Período de Molhamento Foliar, Temperatura e Umidade. Trabalho de Conclusão de Curso. Universidade de São Paulo – 2009. 8
- [06] Carvalho, B. E. B. (2010) Instrumentação Eletrônica para a Captação e Transmissão de Dados Referentes a Avaliações do Desempenho de um Trator Agrícola de Pneus. Monografia. Universidade Federal de Viçosa – 2010. 8
- [07] Santos, P. V. F. (2011). Desenvolvimento de uma Bancada de testes para Motores de Aeromodelos. Monografia. Universidade Federal de Viçosa – 2011. 8
- [08] Coelho, T. C. (2006). Simulador de processos em tempo real. Monografia. Universidade Federal de Viçosa – 2006. 8
- [09] Pedroso, D. A. F. (2007). Desenvolvimento de um sistema de controle de temperatura. Monografia. Universidade Federal de Viçosa – 2007. 8
- [10] Página acessada no dia 23 de abril de 2015: <http://ballast.com.br/encoders-uso-e-aplicacoes/> 8
- [11] Página acessada no dia 16 de abril de 2015: <https://sites.google.com/site/aldofbaum/9-artigos/encoder> 8
- [12] Página acessada no dia 16 de junho de 2015: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf> 9,37
- [13] Página acessada no dia 16 de junho de 2015: <http://www.instrumentacao.com/historico.html> 14
- [14] Natale, F. (2000). Automação Industrial. Ed Érica - 2000. 15
- [15] Gonçalves, M. G. (2003). Monitoramento e Controle de Processos – Tomo 2. Ed Érica - 2003. 15
- [16] Boylestad, R. L., Nashelsky, L. (2013). Dispositivos Eletrônicos e Teoria de Circuitos. Ed Pearson / Prentice Hall – 2013. 15
- [17] Página acessada no dia 10 de junho de 2015: <http://www.newtoncbraga.com.br/index.php/como-funciona/5454-mec128> 16,18,19,20
- [18] Página acessada no dia 10 de junho de 2015: <https://pt.scribd.com/doc/204573017/Cap-5-Rotacao-Torque-Potencia> 17,18,19
- [19] Página acessada em 25 de maio de 2015: <http://www.sabelotodo.org/aparatos/tacentrifugo.html> 17,16
- [20] Página acessada em 16 de junho de 2015: <http://www.ebah.com.br/content/ABAAAAaVUAB/manual-encoder> 16,19
- [21] Adriando, J. D. (2008). Kit de Microcontroladores PIC - Apostila de Microcontroladores PIC16. Ed Exsto Tecnologia Ltda – 2008. 21,22
- [22] Página acessada em 15 de junho de 2015: <http://www.mecatronica.org.br/disciplinas/programacao/002/pic.pdf> 22
- [23] Página acessada em 23 de maio de 2015: <http://www.ebah.com.br/content/ABAAABDLIAK/conhecendo-pic16f877-8-bits> 23
- [24] Barbacena, I. L., Fleury, C. A. (1996). Display LCD. Manual Display Hitachi Dot Matrix LCD Module – 40/2 caracteres - 1996 23,24
- [25] Página acessada em 18 de maio de 2015: [http://mlb-s2-p.mlstatic.com/arduino-display-lcd-16x2-i2c-iic-serial-20547-MLB20193708592\\_112014-O.jpg](http://mlb-s2-p.mlstatic.com/arduino-display-lcd-16x2-i2c-iic-serial-20547-MLB20193708592_112014-O.jpg) 24
- [26] Página acessada em 18 de abril de 2015: <http://www.rtechequipamentos.com.br/pdf/7c6c963efa753e5585972f1faa428c9703092012095915000000.pdf> 26,27,43,44
- [27] Antonio, M. (2006). Apostila de: Programação de Microcontroladores PIC usando Linguagem C. Centro Federal de Educação Tecnológica do Espírito Santo – 2006. 30,31
- [28] Pereira, F. (2007). Microcontroladores PIC: Programação em C. 7ª Edição. Ed Érica – 2007 32
- [29] Página acessada em 14 de junho de 2015: <http://www.ccsinfo.com/ccsfreedemo.php> 29
- [30] Página acessada em 14 de junho de 2015: <https://db-gratuito.toggle.com/free-download/k1226065/Proteus.htm> 34
- [30] Página acessada em 14 de junho de 2015: <http://www.cadsoftusa.com/download-eagle/> 30,34

## ***Apêndice A – Folha de Dados do Encoder Óptico***

### ***Incremental HTR – 1024 – 3 – 5LV***

#### **A.1 Características Elétricas**

As principais características elétricas do *encoder óptico incremental* podem ser encontradas na Tabela 3.

**Tabela 3 - Especificações elétricas do *encoder óptico incremental* [26].**

Forma de onda do sinal de saída	Quadrada
Forma de onda ascensão e queda	2 $\mu$ s ou menos
Máxima resposta	10kHz ~ 100kHz
Corrente de Consumo	<60mA
Capacidade de Saída	20mA
Alimentação	5V
Pulsos por revolução (PPR)	1024

#### **A.2 Características Mecânicas**

As principais características mecânicas do *encoder óptico incremental* podem ser encontradas na Tabela 4.

**Tabela 4 - Características mecânicas do *encoder óptico incremental* [26].**

Diâmetro do eixo	6mm
Carga no eixo	(10~400 PPR) Axial: 1Kg, Radial: 1Kg (mais 400 PPR) Axial: 0,5Kg, Radial: 1Kg
Torque inicial (à 25°C)	30gf-cm ou menos
Velocidade máxima	6000rpm
Vibração	10g (10 $\pm$ 1500Hz)
Polaridade	Não tem proteção contra polaridade
Choque	20g 11ms
Cabo	(4,5) 50cm de comprimento
Peso	<201g

#### **A.3 Características Ambientais**

As principais características ambientais do *encoder óptico incremental* podem ser encontradas na Tabela 5.

**Tabela 5 - Características ambientais do *encoder óptico incremental* [26].**

Temperatura	0~60°C (Operação) -20~80°C (Armazenamento)
Umidade	35~90% (não condensável)
Proteção	IP50 (à prova de poeira)