

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

PROTÓTIPO DE UM ALIMENTADOR AUTOMÁTICO PARA
ANIMAIS DE ESTIMAÇÃO

NÁDIA OCHAKOWSKI

BLUMENAU
2007

2007/1-34

NÁDIA OCHAKOWSKI

**PROTÓTIPO DE UM ALIMENTADOR AUTOMÁTICO PARA
ANIMAIS DE ESTIMAÇÃO**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciências
da Computação — Bacharelado.

Prof. Miguel Alexandre Wisintainer - Orientador

**BLUMENAU
2007**

2007/1-34

PROTÓTIPO DE UM ALIMENTADOR AUTOMÁTICO PARA ANIMAIS DE ESTIMAÇÃO

Por

NÁDIA OCHAKOWSKI

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente: _____
Prof. Miguel Alexandre Wisintainer – Orientador, FURB

Membro: _____
Prof. Mauro Marcelo Mattos, Dr. – FURB

Membro: _____
Prof. Antonio Carlos Tavares – FURB

Blumenau, 09 de julho de 2007

Dedico este trabalho a todos que me ajudaram
na realização deste.

AGRADECIMENTOS

Ao marceneiro Augusto, pela paciência e por aceitar minhas coordenadas na construção da estrutura do alimentador.

A empresa Kowski Mecânica e Plásticos, principalmente a meu tio Célio, por ter confeccionado a rosca sem fim e os extensores do eixo dos motores.

As professoras Joyce e Fabiane por estarem sempre disponíveis para esclarecer minhas dúvidas.

Aos meus amigos por me suportarem nesta fase em que o único assunto é o TCC e principalmente ao Léo que sempre me deu uma mão nos momentos de dificuldade.

Ao meu orientador, Miguel, pelos incentivos e ajuda em momentos de dificuldade. E por ter acreditado na conclusão deste trabalho.

A meu grande amor Juliano, pela paciência, prestatividade, consolo e revisão deste trabalho.

E principalmente a Deus, por ter me dado força para concluir este trabalho.

Há grandes homens que fazem com que todos se sintam pequenos. Mas o verdadeiro grande homem é aquele que faz com que todos se sintam grandes.

Charles Dickens

RESUMO

Este trabalho apresenta a implementação de um protótipo de um alimentador automático para animais de estimação utilizando o microcontrolador PIC16F877, a linguagem C para desenvolver o software embarcado, o circuito integrado RTC para manter atualizada a hora, um display LCD e um teclado telefônico para a interação do usuário e dois motores de passo, um para acionar a rosca sem fim e outro para girar 90 graus o fundo falso do recipiente de comida. Através deste protótipo é possível, determinar a quantidade de refeições, agendar horários para refeição, determinar a quantidade de ração a ser liberada.

Palavras-chave: Microcontrolador. Alimentador. RTC.

ABSTRACT

This work presents the implementation of an automatic pet feeder prototype by using a PIC16F877 microcontroller, the C language to develop the embarked software, an integrated RTC circuit for the time update, a LCD display and a telephonic keyboard for user's interaction. Two step engines have also been used, one to set in motion the no ending thread and the other to turn the false bottom of the food container ninety degrees. Through this prototype, it will be possible to determine the amount of meals, set meal schedules and determine the amount of pet food to be served.

Key-words: Microcontroller, Pet Feeder, RTC.

LISTA DE ILUSTRAÇÕES

Figura 1 - Motor de passo unipolar	18
Figura 2 - Motor de passo bipolar	18
Figura 3 - Resistência entre as bobinas do motor de passo unipolar.....	19
Figura 4 - Resistência entre as bobinas do motor de passo unipolar.....	19
Figura 5 - Bobinas do motor de passo bipolar.....	20
Quadro 1 - Seqüência de passos	21
Figura 6 - Circuito ULN2003	22
Figura 7 - Circuito para amplificar a corrente de um fotodiodo.....	23
Figura 8 - Circuito elétrico do teclado.....	25
Figura 9 - Estrutura do alimentador.....	30
Figura 10 - Rosca sem fim.....	31
Figura 11 - Cano PVC	31
Figura 12 - Esquema eletrônico.....	33
Figura 13 - Diagrama de atividades da programação do alimentador.....	35
Figura 14 - Diagrama de seqüência da programação do alimentador	36
Figura 15 - Diagrama de atividades do processo.....	37
Quadro 2 - Memória EEPROM.....	39
Quadro 3 - Rotina de leitura da EEPROM para inicialização do <i>array</i>	39
Quadro 4 - Rotina que controla o momento para liberar a ração	40
Figura 16 - Tela com informações iniciais	40
Figura 17 - Tela de entrada.....	40
Figura 18 - Tela para informar a quantidade de refeições	41
Figura 19 - Tela para entrada do primeiro horário de refeição.....	41
Figura 20 - Tela com horário 2 menor que o horário 1	41
Figura 21 - Tela de aviso para horário inválido	41
Figura 22 - Tela com o segundo horário de refeição.....	41
Figura 23 - Tela para acionar a liberação da ração.....	42
Figura 24 - Tela do momento da liberação da ração	42
Figura 25 - Tela para determinar a quantidade de ração	42
Figura 26 - Tela com o horário atual e a próxima refeição agendada	42
Figura 27 - Estrutura do alimentador.....	43

LISTA DE SIGLAS

BIOS - *Basic Input and Output System*

CGRAM - *Character Generator RAM*

CGROM - *Character Generator ROM*

DC - *Direct Current*

DDRAM - *Data Display RAM*

EEPROM - *Electrically Erasable Permanent Read-Only Memory*

I2C - *Inter Integrated Communication*

LCD - *Liquid Crystal Display*

LED - *Light Emitting Diode*

LPT - *Line Print Terminal*

LSB - *Least Significant Bit*

MSB - *Most Significant Bit*

MSSP - *Master Synchronous Serial Port*

PVC - *PoliVinil Chloride*

RAM - *Random Access Memory*

RF - *Requisitos Funcionais*

RISC - *Reduced Instruction Set Computer*

RNF - *Requisitos Não Funcionais*

RTC - *Real Time Clock*

SCL - *Serial Clock Line*

SDA - *Serial Data Line*

SPI - *Serial Peripheral Interface*

ST - *Schmitt Trigger*

TTL - *Transistor-Transistor Logic*

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS DO TRABALHO	12
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 NUTRIÇÃO DE CÃES E GATOS	14
2.2 ALIMENTADORES DISPONÍVEIS NO MERCADO.....	15
2.3 MICROCONTROLADOR PIC16F877	16
2.4 MOTOR DE PASSO	17
2.5 SENSORES E ATUADORES	22
2.6 LCD	24
2.7 TECLADO	25
2.8 RTC	26
2.9 TRABALHOS CORRELATOS.....	26
3 DESENVOLVIMENTO DO TRABALHO	28
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	28
3.2 ESPECIFICAÇÃO	29
3.2.1 Especificação da estrutura.....	29
3.2.2 Especificação do hardware.....	32
3.2.3 Especificação do software.....	34
3.3 IMPLEMENTAÇÃO	37
3.3.1 Técnicas e ferramentas utilizadas.....	38
3.3.2 Operacionalidade da implementação	40
3.4 RESULTADOS E DISCUSSÃO	43
4 CONCLUSÕES.....	46
4.1 EXTENSÕES	46
REFERÊNCIAS BIBLIOGRÁFICAS	47

1 INTRODUÇÃO

Animais de estimação, especialmente cães e gatos, estão se tornando cada vez mais presentes nas famílias do mundo todo. No Brasil não é diferente. A estimativa atual gira em torno de mais de vinte e sete milhões de cães e onze milhões de gatos espalhados pelo país (ORSINI, 2004). Se antes eram utilizados principalmente para caça ou guarda, hoje busca-se no animal de estimação a companhia para diminuir os efeitos da solidão e do estresse. Com o aumento dos animais também aumenta a variedade de produtos e serviços oferecidos para atender esse nicho de mercado. As lojas de produtos agropecuários, que mantinham numa única prateleira rações para cães, deram espaço a modernas e grandiosas lojas, os chamados *pet shop*, verdadeiros *shoppings centers* caninos e felinos com diversos tipos de rações, brinquedos e acessórios. Há até SPAs¹ que investem em ofurôs sob medida para cães. Isso só reforça a potencialidade econômica do setor.

Todo esse investimento decorre justamente da convivência cada vez mais estreita e dos vínculos afetivos desenvolvidos entre homens e bichos, porém, nem sempre é possível atender todos os seus desejos. Compromissos profissionais, viagens, o ritmo cada vez mais frenético do dia-a-dia e outros contratempos às vezes obrigam os donos a passarem algum tempo longe dos seus animais de estimação.

Neste contexto desenvolveu-se um alimentador automático que pode ser programado para liberar alimento em períodos agendados pelo usuário e que possua autonomia para dar comida ao cão ou gato por vários dias. É um sistema de malha fechada, pois antes de liberar o novo alimento, são despejados em outro reservatório os restos de ração deixados pelo animal.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um protótipo de um alimentador automático para animais de estimação.

Os objetivos específicos do trabalho são:

¹ Spa do latim *sanus per aquam*, significa “em boa saúde através da água”. É um termo técnico para complexos turísticos que providenciam atividades de lazer normalmente em contato com a natureza (SPA, 2006).

- a) disponibilizar a estrutura mecânica do alimentador;
- b) disponibilizar um hardware dotado de motores, sensores, LCD, teclado, alarme e RTC, a ser acoplado ao alimentador;
- c) desenvolver um software, responsável pelo controle do alimentador;
- d) permitir o agendamento dos horários para alimentação através de um teclado telefônico.

1.2 ESTRUTURA DO TRABALHO

O capítulo dois apresenta a fundamentação teórica necessária para o desenvolvimento do presente trabalho. Estando estruturado da seguinte forma: Nutrição de cães e gatos, Alimentadores disponíveis no mercado, Microcontrolador PIC16F877, Motor de passo, Sensores e atuadores, LCD, Teclado, RTC e Trabalhos correlatos.

No capítulo três são apresentados os principais requisitos do protótipo, especificação da estrutura, do hardware e do software. Em seguida é apresentada a implementação, que abrange as técnicas e ferramentas utilizadas e operacionabilidade da implementação. Ao final deste capítulo são expostos os resultados e discussão.

Por fim, o capítulo quatro apresenta as conclusões obtidas com o desenvolvimento deste trabalho e sugestões para extensão deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados alguns conceitos teóricos relacionados ao trabalho. São abordados os seguintes tópicos: nutrição de cães e gatos, os alimentadores disponíveis no mercado, as principais características do microcontrolador PIC16F877, motor de passo, sensores e atuadores, *display* LCD, Teclado e RTC. Por fim, são apresentados os trabalhos correlatos.

2.1 NUTRIÇÃO DE CÃES E GATOS

Com o passar dos milênios o ser humano se deu à liberdade de domesticar diversos animais. Este processo modificou também o hábito alimentar dos mesmos. É bem verdade que quase a totalidade dos cães e gatos e muitas aves que convivem com o homem já não existem mais na Natureza, ou seja, são espécies criadas pelo homem (ALBANO, 2007).

A Nutrição Animal é uma área em amplo desenvolvimento. Hoje existem vários tipos de rações, como as secas ou as úmidas, para as mais diversas situações e condições. Rações *light*, *diet*, e também aquelas para animais com dificuldades absorptivas ou para animais obesos. A variedade de tipos leva o proprietário às mais diversas escolhas (ALBANO, 2007).

É necessário levar em consideração que cada animal possui um metabolismo diferente, necessidades protéicas e nutricionais das mais variadas. Portanto, há a necessidade de escolher uma alimentação adequada a cada bicho de estimação (ALBANO, 2007).

Não só a escolha do tipo de ração é importante como também a quantidade que será disponibilizada ao animal. Uma forma de estabelecer a quantidade de ração é seguir a descrição nas embalagens. Esta indicação segue a regulamentação do Ministério da Agricultura, que se utiliza de fórmulas pré-estabelecidas para determinar a necessidade diária dos animais de acordo com o peso (MOGIANA ALIMENTOS, 2005).

A comida em abundância, além de provocar obesidade, também pode estar causando outros problemas de saúde ao animal. Deixar a ração o dia todo disponível faz com que a mesma absorva umidade e acabe azedando, além de ser um possível atrativo a roedores e vetores como baratas e camundongos que podem trazer doenças bastante difíceis de serem tratadas (ALBANO, 2007).

É importante observar que cada espécie possui uma alimentação adequada. Ignorar essa regra quase sempre não traz conseqüências imediatas, mas provoca reações desagradáveis, a médio e longo prazo (ALBANO, 2007). Conforme Martins (2005) um gato obeso pode desenvolver doenças como diabete, hipertiroidismo, doenças cardíacas, circulatórias, hepáticas, articulares e doenças de pele.

Os cães devem receber a ração de maneira fracionada e em horários fixo, principalmente os destinados à guarda. Em geral os cães tendem a ter sonolência após as refeições. Se para estes são deixadas quantidades de ração a noite toda, além de uma possível obesidade, é como se estivéssemos convidando-os a tirarem algumas sonecas durante a madrugada (ALBANO, 2007).

Gatos possuem o costume de se alimentar várias vezes ao dia, portanto, é recomendado deixar ração seca sempre á disposição. Embora os gatos gostem de comer o dia todo, não se pode deixá-los comer o quanto quiserem. Uma quantidade diária de alimento tem de ser estipulada. (MARTINS, 2005).

2.2 ALIMENTADORES DISPONÍVEIS NO MERCADO

No mercado já existem vários modelos de alimentadores automáticos para animais de estimação. Estes controlam apenas o recipiente de água, esvaziando o mesmo a cada ciclo. Porém, a maioria desses equipamentos não possibilita ao cliente determinar os horários de alimentação, e sim apenas intervalos de tempo. Isso faz com que este não possa, por exemplo, dar comida apenas de manhã e ao meio dia.

O *Wing Feeder* é um alimentador programável para animais do pequeno ao grande porte, com capacidade para armazenar até quinze quilos de ração. Sua estrutura dispõe de um recipiente para água e outro para comida e ambos são abastecidos no horário programado pelo cliente. Este equipamento alimenta o animal apenas duas vezes ao dia num intervalo de doze horas. A programação do horário ocorre da seguinte forma: a pessoa deve ligar o equipamento no horário escolhido para a alimentação e a partir daquele momento a cada doze horas o equipamento irá liberar comida e água (WING FEEDER, 2006).

O *Pet Feeder* é outro modelo de alimentador automático. Este tem capacidade para armazenar dezenove quilos de ração no reservatório e dispõe de uma estrutura semelhante ao do *Wing Feeder* (TECNOTRI, 2005). O que difere ambos é o intervalo para a liberação de comida, pois o *Pet Feeder* possibilita ao cliente a escolha do intervalo de tempo para a

liberação de comida. Ou seja, através de um botão e um *display* a pessoa determina os ciclos para a liberação da ração.

Os produtos acima mencionados não têm nenhum tipo de controle sobre os recipientes de comida. Sendo assim, caso o animal tenha deixado alguma sobra de ração de um dia para outro o alimentador irá despejar nova quantidade de comida sobre a que já estava no recipiente. A falta deste controle pode gerar uma contaminação de germes e fungos e causar até alguma doença no animal de estimação.

2.3 MICROCONTROLADOR PIC16F877

Souza (2005, p. 21) define o microcontrolador como um pequeno componente eletrônico, dotado de uma inteligência programável, sendo utilizado no controle de periféricos, tais como LED, botões, LCD, motores, resistências, sensores, entre outros.

O PIC16F877 é um microcontrolador fabricado pela Microchip Technology. Sua frequência de operação (*clock*) chega até 20MHz. Pode trabalhar com uma fonte de alimentação de 2 *volts* a 5,5 *volts* (PIC16F877, 2006).

É um modelo que agrupa de uma só vez o maior número possível de recursos disponíveis. Este microcontrolador possui quarenta pinos, o que possibilita o desenvolvimento de um hardware complexo e capaz de interagir com diversos recursos ao mesmo tempo. Possui uma via de programação com 14 bits e sua arquitetura RISC se compõe de trinta e cinco instruções. É composto de uma memória de programa com 8 kbytes, memória RAM com 368 bytes e memória EEPROM interna com 256 bytes. Esta memória de programação permite a gravação rápida do programa diversas vezes no chip (SOUZA; LAVINIA, 2003, p. 21).

É constituído por três *timers*. Estes contadores possuem características próprias, como limite de contagem, tipo de incremento, geração de interrupção, periféricos associados, etc. O *timer0* e o *timer2* possuem um contador de 8 bits, já o *timer1* possui um contador de 16 bits. O que difere bastante o *timer2* dos demais *timers* é que ele não conta de zero até o limite imposto pelos 8 bits, quem impõe este limite é o valor escrito no registrador PR2. Este *timer* está relacionado somente ao *clock* interno (SOUZA; LAVINIA, 2003, p. 50).

As entradas e saídas deste microcontrolador estão agrupados por PORTs, sendo: PORTA, PORTB, PORTC, PORTD, PORTE (SOUZA; LAVINIA, 2003, p. 45).

“Internamente esses pinos são ligados de formas diferentes, principalmente pela sobrecarga de recursos aplicados a cada um deles” (SOUZA; LAVINIA, 2003, p.46).

O PIC possui dois tipos distintos de comunicação serial, denominados SPI e I2C. Estes protocolos são utilizados quando se tem necessidade de realizar a comunicação do PIC com outros microcontroladores, com memórias externas, *drives* de LCD, chip relógio/calendário, conversores, sensores e uma infinidade de outros periféricos. Estes protocolos fazem parte de um sistema comum do PIC chamado MSSP. Desta forma, os dois protocolos compartilham os mesmos recursos (pinos, hardware internos e registradores). Por este motivo, só é possível utilizar um sistema de cada vez (SOUZA; LAVINIA, 2003, p. 197).

No protocolo I2C os dispositivos envolvidos na comunicação são denominados de *Master* (Mestre) e *Slave* (Escravo). O dispositivo mestre é sempre o responsável pela geração do sinal de *clock*. Sendo assim, para implementar um barramento I2C basta configurar um dos dispositivos como mestre e os demais como escravo. Uma das principais características deste protocolo está no endereçamento, que é feito por software. Desta forma a comunicação será sempre iniciada pelo dispositivo mestre, que inicia a linha SDA com o endereço do dispositivo escravo com o qual ele irá se comunicar (ZANCO, 2006, p. 151).

Esta comunicação ocorre através de dois fios ou linhas, sendo uma linha de *clock* (SCL) e outra de dados (SDA). A ligação de diversos dispositivos nas mesmas linhas, forma um autêntico barramento de comunicação serial que suporta velocidades de até 3,4 Megabits por segundo (PERREIRA, 2006, p. 274).

2.4 MOTOR DE PASSO

O motor de passo é utilizado quando se tem a necessidade de posicionar algo de forma exata, ou de controlar os passos (voltas) do motor.

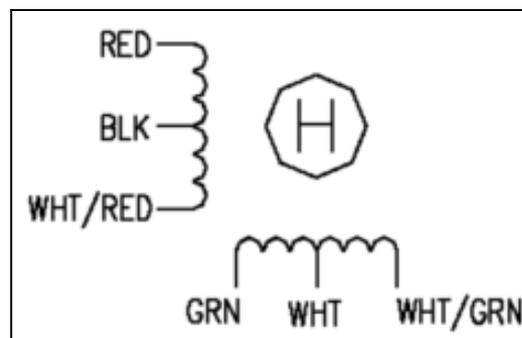
Os motores comuns, giram em velocidade constante, pois possuem apenas dois estágios de operação, ou seja, parados ou girando; enquanto os motores de passo deslocam-se por impulsos ou passos discretos e exibem três estágios: parados, ativados com rotor travado (bobinas energizadas) ou girando em etapas. Este movimento pode ser brusco ou suave, dependendo da frequência e amplitude dos passos em relação à inércia em que ele se encontra. (COSTA, 2000).

Conforme Ferraz Netto (2005), este tipo de motor possui um rotor² que é atraído

² Rotor é o eixo do motor, este eixo é formado por um ímã móvel.

seqüencialmente por um par de pólos (eletroímãs). Estes eletroímãs são ligados e desligados seguindo impulsos cuidadosamente controlados de modo que os pólos magnéticos do rotor se movam de um eletroímã para outro devidamente habilitado.

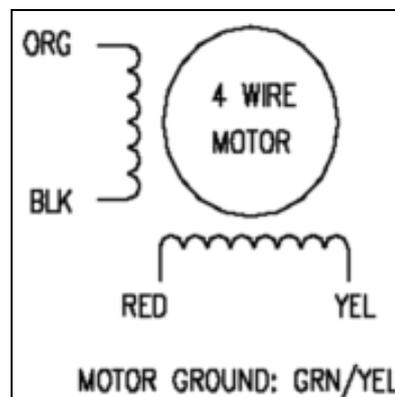
São classificados em duas categorias: unipolares e bipolares. O mais utilizado é o motor unipolar que possui duas ou quatro bobinas, controladas por cinco ou seis fios. Cada par de bobinas possui um par de fios e uma derivação central chama de fio comum, onde será ligada a alimentação. Estas bobinas são independentes, de forma que o campo magnético possa ser invertido sem a necessidade de se inverter o sentido da corrente. Conforme demonstra a Figura 1 (QUEIROZ, 2003).



Fonte: Queiroz (2003).

Figura 1 - Motor de passo unipolar

Já os Bipolares são compostos por várias bobinas em sua carcaça e por isso exigem circuitos mais complexos. Suas bobinas não possuem a derivação central, os enrolamentos são separados, sendo necessário uma polarização reversa durante a operação para o passo acontecer. Composto de quatro fios de controle, conforme Figura 2. Apesar de pequenos estes motores conseguem proporcionar um torque maior o que os torna vantajosos (QUEIROZ, 2003).



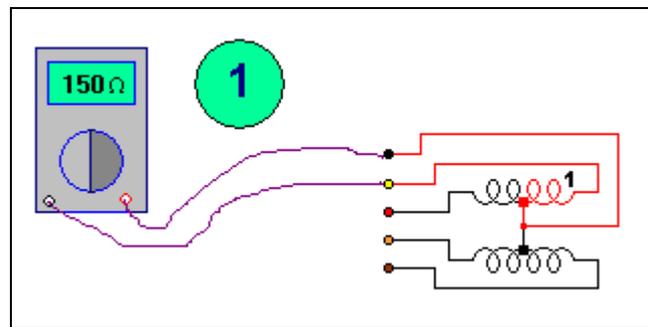
Fonte: Queiroz (2003).

Figura 2 - Motor de passo bipolar

Para controlar este deslocamento é preciso ter conhecimento da seqüência interna das bobinas, sendo necessário verificar o “*data book*” do fabricante. Porém, em motores usados,

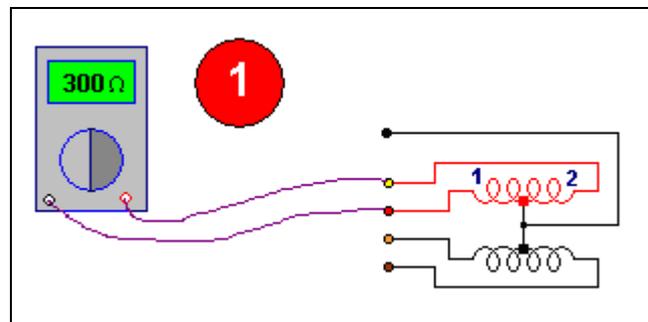
na maioria das vezes já não há mais as descrições necessárias no chassi do motor para se localizar o *data book* do mesmo. Como não existe uma padronização entre os fabricantes, que forneça através de um código de cores a identificação das ligações internas das bobinas, pode-se obter esta informação através de alguns testes utilizando um *multitest* e uma fonte (SOARES, 2004, p. 6).

Nos motores unipolares é preciso descobrir primeiro qual dos fios é o fio comum. Para isso é preciso medir a resistência entre os fios do motor. O que apresentar uma resistência reduzida em relação aos outros será o fio comum, conforme Figura 3 e Figura 4 (SAUL; FREITAS, 2006, p. 45).



Fonte: Messias (2007).

Figura 3 - Resistência entre as bobinas do motor de passo unipolar



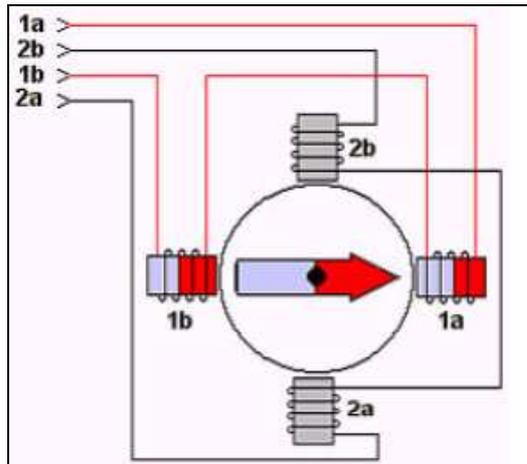
Fonte: Messias (2007).

Figura 4 - Resistência entre as bobinas do motor de passo unipolar

O segundo passo é ligar o fio comum do motor ao positivo de uma fonte e logo após fazer um contato seqüencial com o negativo da fonte em cada um dos quatro fios restantes. Este contato do fio negativo da fonte fará com que o eixo do motor desloque em uma direção. Se a seqüência realizada estiver correta, o eixo se desloca de forma regular. Caso contrário o mesmo fará passos irregulares ou inverterá o movimento subitamente (SAUL; FREITAS, 2006, p. 46).

Já nos motores bipolares também é necessário medir a resistência entre os fios, porém neste motor não existe um fio comum. O que se deve encontrar ao medir a resistência é o conjunto de fios que ligam cada bobina. Após encontrar os pares de fios que ligam cada bobina é preciso criar conjuntos de combinações até fazer com que o motor gire em uma

direção regular (Figura 5) (QUEIROZ, 2003).



Fonte: Queiroz (2003).

Figura 5 - Bobinas do motor de passo bipolar

A passagem do rotor por estas bobinas é chamada de passo que é o menor deslocamento angular. A cada passo faz-se um incremento rotativo em graus que é um pedaço de uma rotação completa. Conforme Queiroz (2003) tanto o motor de passo unipolar como o bipolar pode operar de três modos, são eles:

- a) única excitação de fase – opera com apenas uma bobina energizada a cada passo, o que proporciona menor torque, menos consumo de energia e maior velocidade;
- b) excitação dual - o motor opera com duas bobinas energizadas a cada passo, proporcionando aproximadamente 30 a 40% de mais torque, aumento do consumo de energia e velocidade em relação a excitação única;
- c) meio passo – é a combinação da excitação única e dual alternada que gera um passo com a metade do ângulo normal, este modo consome mais energia e gera uma velocidade menor que os modos anteriores. Porém é muito mais preciso e possui um torque semelhante a excitação dual.

No Quadro 1 são apresentadas as seqüências para controlar um motor de passo em cada um dos modos apresentados (QUEIROZ, 2003).

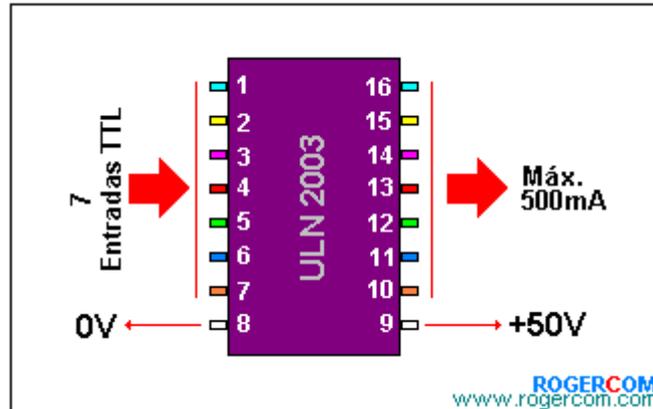
Modo de operação	Seqüência de pulsos Unipolar						Polaridade Bipolar													
Única excitação de fase	Bobinas		1	A	2	A	1	B	2	B	Bobinas		1	A	2	A	1	B	2	B
	01		1	0	0	0					01		+	-	-	-				
	02		0	1	0	0					02		-	+	-	-				
	03		0	0	1	0					03		-	-	+	-				
	04		0	0	0	1					04		-	-	-	+				
Excitação dual	Bobinas		1	A	2	A	1	B	2	B	Bobinas		1	A	2	A	1	B	2	B
	01		0	0	1	1					01		-	-	+	+				
	02		0	1	1	0					02		-	+	+	-				
	03		1	1	0	0					03		+	+	-	-				
	04		1	0	0	1					04		+	-	-	+				
Meio passo	Bobinas		1	A	2	A	1	B	2	B	Bobinas		1	A	2	A	1	B	2	B
	01		1	0	0	0					01		+	-	-	-				
	02		1	1	0	0					02		+	+	-	-				
	03		0	1	0	0					03		-	+	-	-				
	04		0	1	1	0					04		-	+	+	-				
	05		0	0	1	0					05		-	-	+	-				
	06		0	0	1	1					06		-	-	+	+				
	07		0	0	0	1					07		-	-	-	+				
	08		1	0	0	1					08		+	-	-	+				

Fonte: adaptado de Queiroz (2003).

Quadro 1 - Seqüência de passos

A velocidade do motor depende do intervalo de tempo definido entre cada passo. Desta forma quanto menor esse intervalo, maior será a velocidade em que o motor irá girar. Porém se este intervalo for menor que 10 ms, o motor perderá o torque e em vez de girar, irá vibrar (MESSIAS, 2007).

Para acionar um motor de passo é preciso de um hardware específico, chamado *driver*. É possível fazer um *driver* usando transistores de potência como os BD135, DB241 etc. Porém, a maneira mais fácil é adquirir *drivers* prontos, como o ULN 2003 ou ULN2803, que nada mais são que *arrays* de transistores Darlington que podem controlar correntes de até 500mA. Estes *drivers* estão disponíveis em formato de circuitos integrados, conforme a Figura 6 (MESSIAS, 2007).



Fonte: Messias (2007).

Figura 6 - Circuito ULN2003

A tensão de trabalho normalmente está impressa no próprio chassi do motor, desta forma a utilização da tensão correta é fundamental na obtenção do torque³ do componente. Tensões acima do estipulado pelo fabricante em seu manual costumam aumentar o torque do motor, porém, esta ação tende a diminuir a vida útil do mesmo (LEÃO et al., 2003).

2.5 SENSORES E ATUADORES

O sensor é um dispositivo que recebe e responde a estímulos de uma grandeza física, podendo fornecer diretamente ou indiretamente um sinal para corrigir desvios em sistemas (GONÇALVES, 2003). Atualmente existem vários tipos de sensores comercialmente disponíveis. (FOTODIODO, 2006).

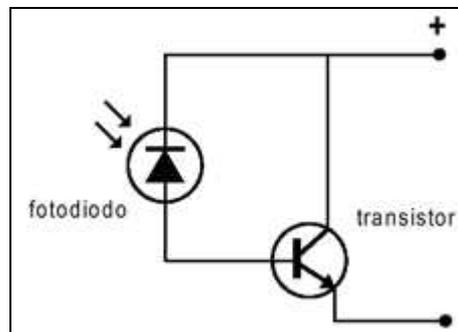
O fotodiodo é um sensor que possibilita a utilização da luz como fator determinante no controle da corrente elétrica (SOUZA; PEREIRA, 2004). Este dispositivo converte luz num sinal elétrico. Ou seja, quando a intensidade luminosa incide sobre o material semicondutor, os fótons podem fornecer aos elétrons da superfície do material energia suficiente para produzir a ruptura das ligações covalentes (RUEDA; SANTOS; RABELO, 2005).

São diodos de junção p-n e suas características elétricas dependem da quantidade de luz que incide sobre esta junção. Por este motivo precisam de uma janela transparente à luz para que possam entrar os raios luminosos e incidir na junção p-n (RUEDA; SANTOS;

³ Torque é a força exercida sobre um corpo rígido que pode girar em torno de um eixo, onde a força não passa através do ponto de apoio (eixo). O corpo tende a girar ao redor deste eixo. A variação no movimento rotacional ocorre graças ao torque, que é análogo à força, que causa variações no movimento translacional (SERWAY; JEWETT JUNIOR, 2004, p. 327).

RABELO, 2005). A corrente elétrica e o fluxo luminoso variam quase que linearmente, ou seja, um aumento na intensidade luminosa resultará em um aumento na corrente reversa. Esta corrente reversa é essencialmente nula na ausência de luz incidente (SOUZA; PEREIRA, 2004).

“As correntes obtidas com um fotodiodo são geralmente baixas, menores que $100 \mu\text{A}$ para uma incidência luminosa de $1 \text{ mW}/\text{cm}^2$ ” (AGUIAR; LAUDARES, 2001). Desta forma não se pode simplesmente substituir um botão por um fotodiodo. Com um botão pressionado a corrente obtida pelo circuito é de aproximadamente 5 mA . Para obter correntes desta magnitude é necessário amplificar o sinal do fotodiodo, ligando-o a um transistor, conforme Figura 7 (AGUIAR; LAUDARES, 2001).



Fonte: Aguiar; Laudares (2002).

Figura 7 - Circuito para amplificar a corrente de um fotodiodo

“Os foto-diodos são sensores mais sensíveis, por isso são utilizados em aplicações onde não há tanta variação de intensidade luminosa.” (VALLADÃO, 2006). É comum encontrar nestes componentes um encapsulamento com uma lente em sua porção frontal, o que faz com que o sensor seja direcional, ou seja, receba a fonte de luz somente de uma direção. Este encapsulamento faz com que os fotodiodos funcionem na região do infravermelho, possibilitando uma certa rejeição da interferência da luz natural ambiente (VALLADÃO, 2006).

Atuador é um dispositivo que realiza ações requeridas pelo sistema, que tem por objetivo reposicionar uma variável, conforme sinal gerado por um controlador. É chamado de atuador, pois atua diretamente no processo, modificando as condições de um sistema (CENTRO SENAI FUNDAÇÃO ROMI FORMAÇÃO DE FORMADORES, 2000).

2.6 LCD

O Display LCD é um elemento fundamental para facilitar a interação do usuário com o equipamento. Esta facilidade fez com que os LCDs ganhassem popularidade e hoje são amplamente utilizados numa infinidade de equipamentos eletrônicos, desde eletrodomésticos até instrumentos de medição de alta precisão (ZANCO, 2006, p. 75).

É constituído por duas placas de cristal muito finas, entre as quais há um preenchimento de cristal líquido. Este material que preenche a célula de vidro tem uma estrutura molecular cristalina que é capaz de mudar sob a influência de um campo elétrico, tornando-se transparente ou refletante conforme a direção em que se organizem as moléculas. A superfície interior das duas camadas de vidro está recoberta de uma camada condutora de metal transparente aplicado diretamente nos vidros. Uma voltagem aplicada sobre este metal cria um campo elétrico que faz as moléculas de cristal líquido mudarem de orientação (RADIOAMADORES.NET, 2003).

Para dar toda a funcionalidade encontrada em um LCD, no entanto, é necessário um circuito eletrônico complexo chamado LCD *Controler* (Controlador do LCD). Existem vários circuitos integrados LCD *Controler* no mercado. Os mais populares são o KS0066, fabricado pela Samsung Eletronics e o HD44780, fabricado pela Hitachi. O LCD *Controler* é o coração do LCD, cabendo a ele toda a tarefa pesada de fazer acender os diversos segmentos do display. (ZANCO, 2006, p. 78).

Esses controladores permitem uma comunicação simples com sistemas microprocessados ou microcontrolados, com largura de barramento de dados selecionável para quatro ou oito bits (PEREIRA, 2006, p. 310). No modo oito bits, é utilizado todo o barramento de dados do LCD, onde um byte é enviado ao LCD através dos pinos D0 a D7. No modo quatro bits, utiliza-se apenas as vias D4 a D7, e envia-se o byte MSB em quatro bits, e posteriormente o byte LSB com quatro bits (SILVA, 2006, p. 141). Este barramento possui também uma linha de controle composta por mais três pinos: EN (Enable) pino seis, R/W (leitura/escrita) pino cinco e RS (entrada de dados ou instrução) pino quatro.

Há também a comunicação serial onde todo este controle de pinos se resume em apenas dois pinos de comunicação. Suas principais vantagens em relação a comunicação paralela são, redução do número de pinos do microcontrolador e tamanho do programa final reduzido. (SOARES, 2006, p. 34). Alguns displays possuem mais que quatorze pinos, sendo que os pinos quinze e dezesseis são utilizados no controle do “*Back Light*” (SOARES, 2004, p. 4).

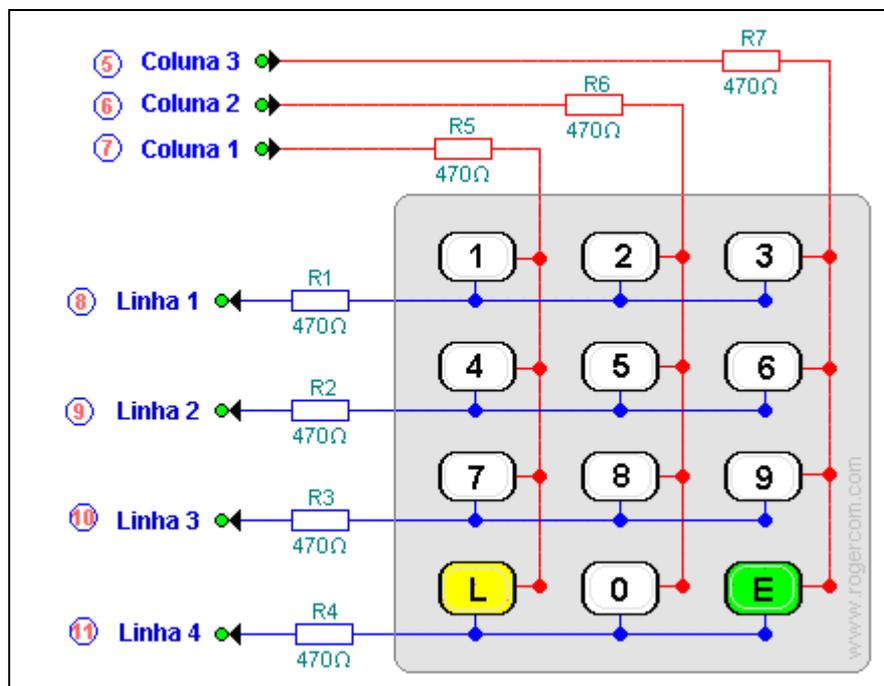
Conforme Pereira (2006, p. 310), esses controladores são compostos de 80 bytes de

memória RAM chamada de DDRAM, 64 bytes de RAM para o gerador de caracteres do usuário chamada de CGRAM e 9920 bits de memória ROM do Gerador de Caracteres (CGROM).

“Os LCDs caracteres normalmente são compatíveis com o código ASCII, e podem gerar letras, números e caracteres especiais, além de caracteres katakana, europeus e gregos.” (ZANCO, 2006, p. 75). Sendo especificado em número de linhas por colunas, determinando assim seu tamanho e quantidade de caracteres possíveis de serem apresentados.

2.7 TECLADO

Os teclados matriciais são muito utilizados sempre que se precisa coletar informações, em campo, no chão de fábrica, ou mesmo passar informações a um robô. Na Figura 8 é apresentado um teclado matriz. Suas teclas estão dispostas em linhas e colunas (SOARES, 2003, p. 56).



Fonte: Messias (2007).

Figura 8 - Circuito elétrico do teclado

A disposição das teclas num formato matriz pode reduzir o número de pinos do microcontrolador, já que desta forma não é necessário ligar um botão em cada pino. Sendo assim, o número de pinos necessários para rastrear o teclado é igual à soma do número de colunas com o número de linhas (ZANCO, 2006, p. 22).

Para descobrir qual tecla da matriz foi pressionada é necessário ler a linha e coluna referentes à tecla. Este resultado pode ser obtido através de uma técnica conhecida como varredura. Esta técnica consiste em forçar os valores lógicos das linhas em nível “1” e inserir “0” lógico na coluna que está sendo testada, mantendo todas as outras em nível “1”. Ao inserir o nível baixo em uma das colunas as linhas referentes serão lidas. Desta forma se alguma linha que estiver com o valor alterado significa que a respectiva tecla foi pressionada (SOARES, 2003, p. 56).

“Uma análise simplificada dos problemas envolvidos na leitura de teclado leva à conclusão de que basta observar o nível lógico associado a uma tecla: caso ela esteja pressionada, teremos um determinado nível; caso esteja solta, teremos outro nível.” (PEREIRA, 2006, p. 300).

2.8 RTC

RTC é um chip contendo o circuito de relógio de tempo real que fornece segundos, minutos, hora, dia, mês e ano, mesmo quando o equipamento está desligado. É um processador dedicado ao controle de horários que fornece estas informações ao BIOS, ao ambiente operacional e a outros programas que requisitam tal informação (JORNAL ELETRÔNICO NOVO MILÊNIO, 2001).

É possível manter estas informações com o sistema desligado, pois este circuito disponibiliza um pino ou porta onde é ligada a uma bateria (pilha). É um circuito programável que possui 56 bytes de RAM e consome menos do que 500nA, o que proporciona um pequeno consumo. Um exemplo de RTC é o DS1307 fabricado pela Dallas. Sua comunicação com o PIC ocorre através do barramento I2C. Este RTC necessita de duas linhas de sinal, SDA e SCL, para se comunicar com o PIC (MICROCONTROLLER PROS CORPORATION, 2003).

2.9 TRABALHOS CORRELATOS

Bernardes (2004, p. 12) apresenta uma máquina eletrônica para venda de Trident,

baseada em uma espiral, um motor DC e um *switch* de posicionamento de volta final. Para liberar o produto, a placa controladora aciona o motor DC, que irá girar a espiral em uma volta. Como os Tridents estão em cada volta da espiral, isso fará com que o produto saia da espiral e caia para o usuário pegar. O microcontrolador utilizado na placa controladora foi o PIC16F628, da mesma família que se deseja utilizar neste protótipo.

Ferreira, Purger Neto e Bueno Junior (2005) demonstram o projeto Câmera Pan-Tilt, que tem como objetivo movimentar uma câmera de vigilância. Para isso foi usado o microcontrolador PIC16F877 e dois motores de passos. A utilização de dois motores deve-se ao fato de que a câmera deverá girar no sentido horizontal e vertical. Para movimentar a câmera utilizou-se um controle de quatro botões onde foi possível obter os comandos de cima, baixo, direita e esquerda. Foi utilizada a linguagem C na programação do PIC16F877.

Manteufel (2001) apresenta um dosador de matéria-prima para pequenas empresas de produtos pré-fabricados de concreto. Neste projeto foi desenvolvido um protótipo que faz a dosagem dos materiais necessários para a produção do concreto, com quantidades previamente definidas e variáveis, de acordo com a peça que se deseja produzir. Foram utilizadas esteiras com a função de levar a matéria-prima da parte inferior dos silos até a caixa coletora ou até a balança para pesagem do cimento. Para movimentar as esteiras foram utilizados motores de corrente contínua. As esteiras são responsáveis pela dosagem dos áridos (areia e brita). A quantidade desta matéria prima depositada na caixa coletora é feita em função do tempo em que a esteira esteja ligada. O cimento também é transportado por intermédio de uma esteira que despeja o cimento sobre a bandeja de uma balança que irá medir o peso do cimento. Ao atingir o peso estipulado o motor da esteira é desligado e o cimento é despejado na caixa coletora. Neste protótipo foram utilizados dois circuitos digitais, um para fazer a leitura dos valores da balança e outro que liga e desliga os motores do protótipo. Ambos estão conectados a um computador via LPT1.

3 DESENVOLVIMENTO DO TRABALHO

Este capítulo está dividido da seguinte forma:

- a) definição dos requisitos: detalha os requisitos funcionais e não funcionais do software e do hardware;
- b) especificação: nesta seção são apresentadas a especificação da estrutura do alimentador, do hardware e do software;
- c) implementação: esta seção aborda as técnicas e ferramentas utilizadas no desenvolvimento deste protótipo, assim como apresenta trechos do código fonte desenvolvido. Também é demonstrado a operacionabilidade da implementação, através de um estudo de caso;
- d) resultados e discussões: por fim são apresentadas as dificuldades e conquista através do desenvolvimentos deste projeto.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Abaixo são detalhados os requisitos RF e RNF do protótipo.

Os RF e RNF do hardware são:

- a) possibilitar a entrada de dados através de um teclado (RF);
- b) possibilitar a visualização da entrada através de um *display* LCD (RF);
- c) armazenar a entrada na memória EEPROM de dados do microcontrolador (RNF);
- d) utilizar o microcontrolador PIC16F877 (RNF);
- e) utilizar o circuito integrado RTC, para o controle da hora (RNF);
- f) utilizar um sensor foto-diodo no reservatório da comida (RNF);
- g) utilizar motor de passo para acionar a rosca sem fim e rotacionar o fundo falso (RNF).

Os RF e RNF do software são:

- a) interpretar os valores lidos do teclado (RF);
- b) monitorar a quantidade de voltas (giros) da rosca sem fim (RF);
- c) monitorar a rotação do fundo falso do recipiente de comida (RF);
- d) monitorar a liberação do alimento conforme horários definidos (RF);

- e) monitorar o sensor do reservatório de comida (RF);
- f) ser implementado utilizando a linguagem C (RNF).

3.2 ESPECIFICAÇÃO

O sistema solicitará que se informe por meio de um teclado quantas vezes por dia se deseja liberar comida para o animal e os respectivos horários. Estas solicitações serão visualizadas através de um LCD. Diferentemente de outros produtos, não há a obrigatoriedade que se alimente o animal de oito em oito horas, por exemplo.

Antes que o novo alimento seja liberado no recipiente de comida, serão retirados eventuais restos de ração deixados pelo animal. O recipiente de comida possuirá um fundo falso que irá girar 90 graus, acionado por um motor de passo. Outro motor acionará a rosca sem fim para que a nova porção de comida seja despejada no recipiente.

O hardware foi especificado através da ferramenta Proteus Professional. Para especificar o sistema foi utilizada a ferramenta Enterprise Architect, onde foi modelado o diagrama de atividades da UML.

3.2.1 Especificação da estrutura

A Figura 9 apresenta o esquema do projeto para a criação da estrutura. Nela estão fixados todos os componentes necessários para o controle do protótipo do alimentador.

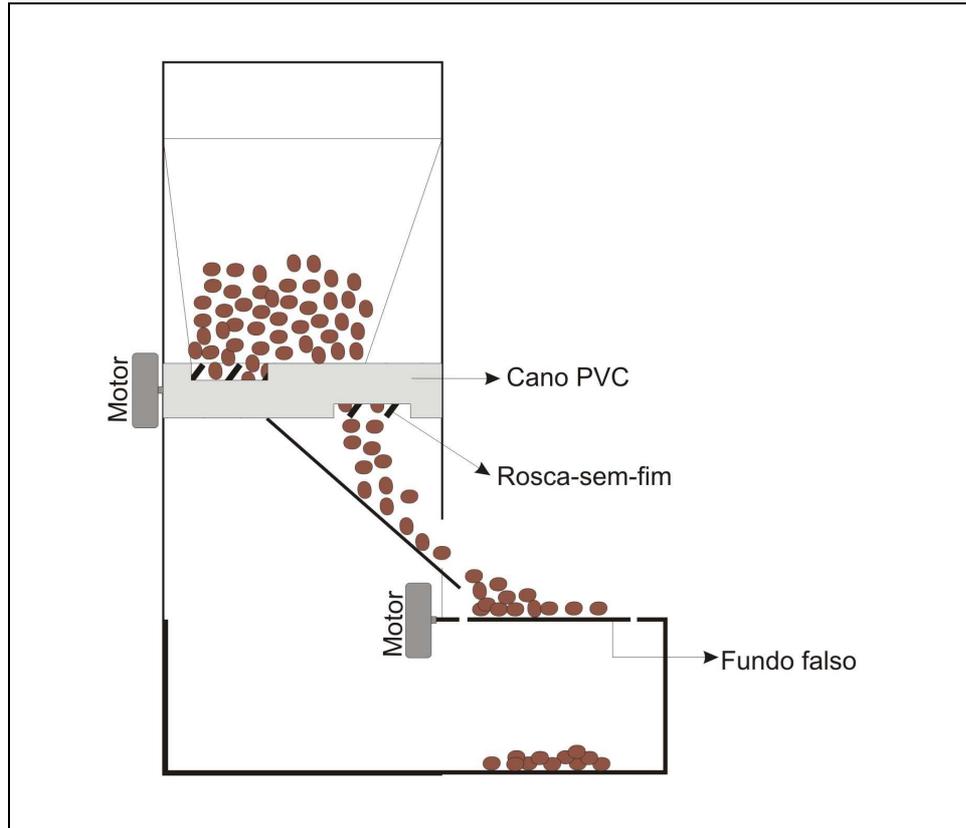


Figura 9 - Estrutura do alimentador

O material utilizado na construção da estrutura do alimentador foi madeira. A escolha por este material se deu pelo fato da facilidade do manuseio e moldagem. Esta estrutura consiste em uma caixa composta internamente por um funil também de madeira. Este funil tem a simples função de armazenar e não deixar trancar a ração, fazendo com que sempre haja ração sobre a rosca sem fim.

A rosca foi feita de nylon pois é considerado um material leve. Esta preocupação se dá pelo fato de evitar uma sobrecarga para o motor de passo que este é responsável pelo rotacionamento da mesma. Sua função é levar a ração que se encontra na parte inferior do funil para a rampa do alimentador. A Figura 10 apresenta o projeto para a criação da rosca sem fim.

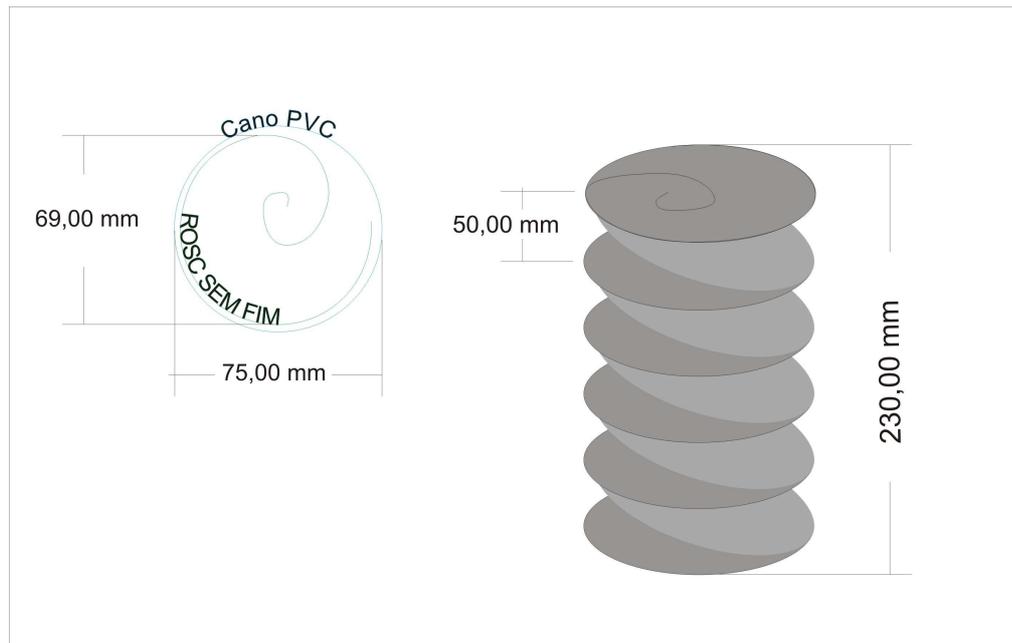


Figura 10 - Rosca sem fim

Esta rosca está dentro de um cano de PVC, o qual possui duas aberturas, que podem ser vista na Figura 11. A primeira abertura serve como entrada da ração dentre os elos da rosca e a segunda abertura como saída da ração até a rampa do alimentador.

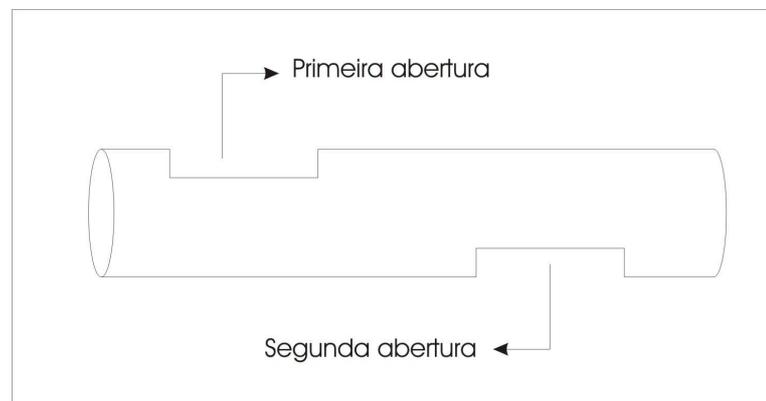


Figura 11 - Cano PVC

A rampa está posicionada logo abaixo da segunda abertura do cano PVC e faz parte da estrutura do alimentador. Sua função é direcionar a ração liberada até o prato de comida do animal.

O prato de comida é composto de um fundo falso. Este fundo falso foi criado com a intenção de não deixar que o novo alimento se misture com os restos de ração deixados pelo animal na refeição anterior. Desta forma antes de liberar o novo alimento o fundo falso do prato de comida será rotacionado noventa graus e despejará os restos de ração num segundo reservatório e retornará a sua posição original para o recebimento do novo alimento. Para esta rotação é utilizado um segundo motor de passo que possui seu eixo encaixado no fundo falso.

O reservatório para os restos de ração está localizado na parte inferior do alimentador. O mesmo funciona como uma gaveta que pode ser retirada do alimentador no momento em que o usuário desejar limpá-lo.

Na próxima seção será detalhada a construção do hardware que irá controlar os dispositivos do motor.

3.2.2 Especificação do hardware

O circuito eletrônico que simula o hardware foi especificado através da ferramenta Proteus Professional e é apresentado na Figura 12. Nesta figura foram criadas algumas marcações com a finalidade de facilitar o entendimento do mesmo.

Na marcação A da Figura 12 encontra-se o RTC DS1307, que é o responsável por informar a hora para o microcontrolador PIC16F877. O RTC está ligado a uma bateria que manterá a hora mesmo quando o sistema estiver desligado. Desta forma caso o sistema fique sem energia ou seja desligado por um tempo, quando o mesmo for novamente ligado não haverá a necessidade de ajustar o relógio.

Na marcação B está o circuito controlador do teclado. Neste circuito são utilizados quatro diodos, que fazem uma ligação entre cada linha do teclado e o pino 33 do microcontrolador. A função destes diodos é receber um sinal em uma das linhas do teclado e jogar este sinal na porta 33, a qual está configurada como interrupção externa.

Na marcação C é apresentado o esquema elétrico do motor de passo responsável por rotacionar a rosca sem fim. O motor utilizado é do tipo unipolar de seis fios. Os transistores utilizados são do tipo NPN com correntes na ordem de 2A.

Já a marcação D apresenta o mesmo circuito demonstrado na marcação C. Porém a marcação D possui o esquema responsável por rotacionar o fundo falso do prato de comida.

A marcação E apresenta o circuito do fotodiodo emissor e receptor. Estes fotodiodos funcionam na região do infravermelho que será responsável por verificar se ainda há ração no reservatório de comida. Caso o emissor e receptor consigam se comunicar será acionado o alarme que também se encontra nesta marcação.

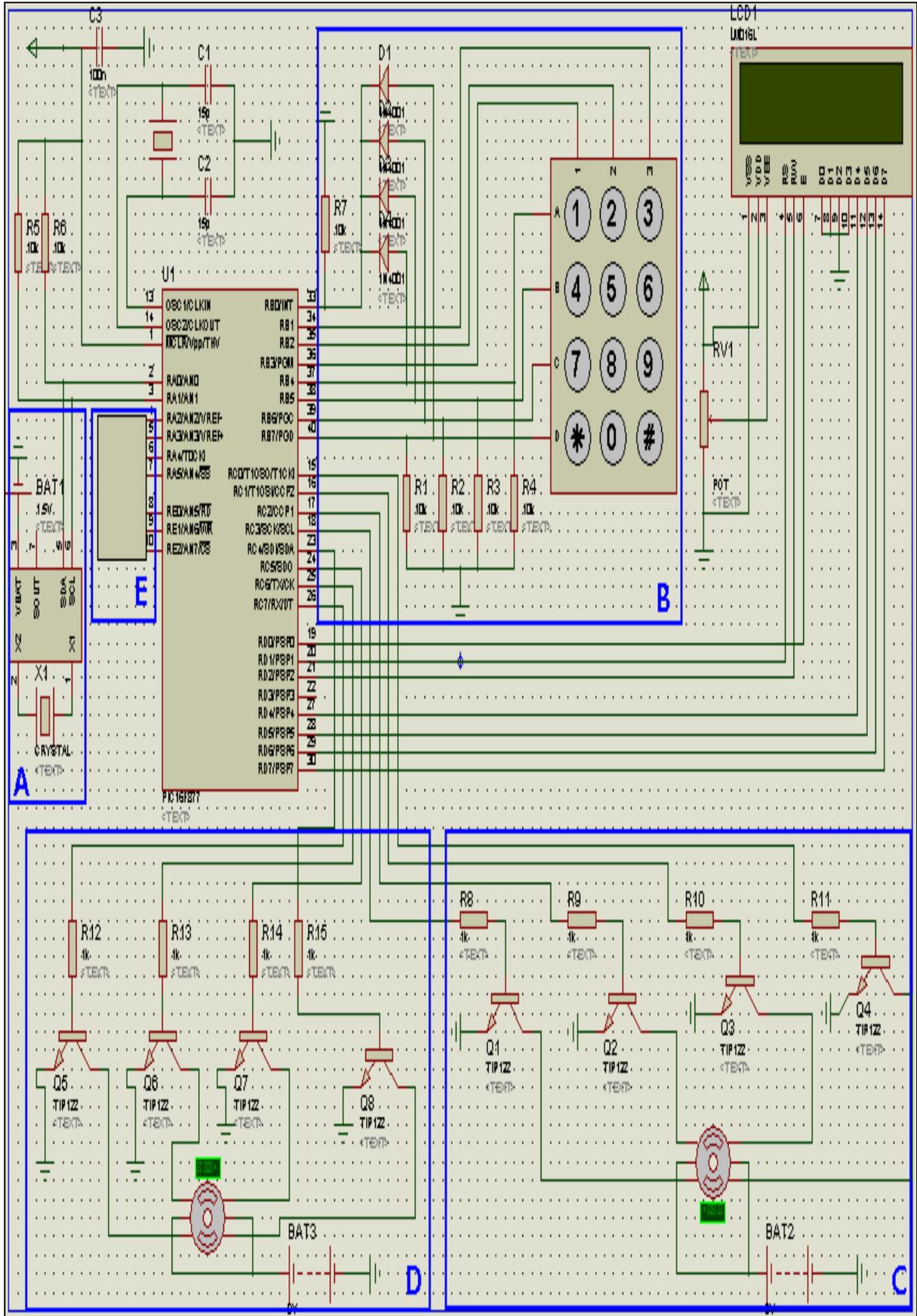


Figura 12 - Esquema eletrônico

3.2.3 Especificação do software

A especificação do software é apresentada na Figura 13, Figura 14 e Figura 15, através dos diagramas de atividades e de seqüência da UML. A ferramenta utilizada para a especificação foi o Enterprise Architect.

Na Figura 13 é apresentada a interação do usuário com o sistema. Este diagrama apresenta o momento da programação ou agendamento dos horários e definição da quantidade de ração que será liberada nos horários definidos, sendo esta uma representação a nível de usuário.

Já na Figura 14 é apresentado o diagrama de seqüência que também representa a interação do usuário no momento da programação, porém este faz uma representação a nível de software, onde são apresentadas as principais funções do sistema e as bibliotecas utilizadas.

A biblioteca `LCD.c` tem a função de apresentar as mensagens geradas pelo programa principal no LCD. A mesma possui as rotinas de posicionamento de colunas e linhas. Já a biblioteca `KBD.c` faz o tratamento do teclado onde são feitas as leituras de coordenadas. A partir disso é devolvido ao programa principal a tecla a qual foi pressionada. Esta biblioteca é iniciada após a interrupção do teclado, onde todas as interrupções estão desabilitadas para o uso desta. Após o processo de interação do usuário, no processo de agendamento, esta biblioteca é finalizada e as interrupções são novamente habilitadas. Por fim a biblioteca `DS137.c` foi utilizada para fazer as inicializações a comunicação entre o circuito DS1307 e o microcontrolador PIC16F877.

Todas as bibliotecas utilizadas foram alteradas de forma a atender as necessidades e ligações dos componentes no microcontrolador.

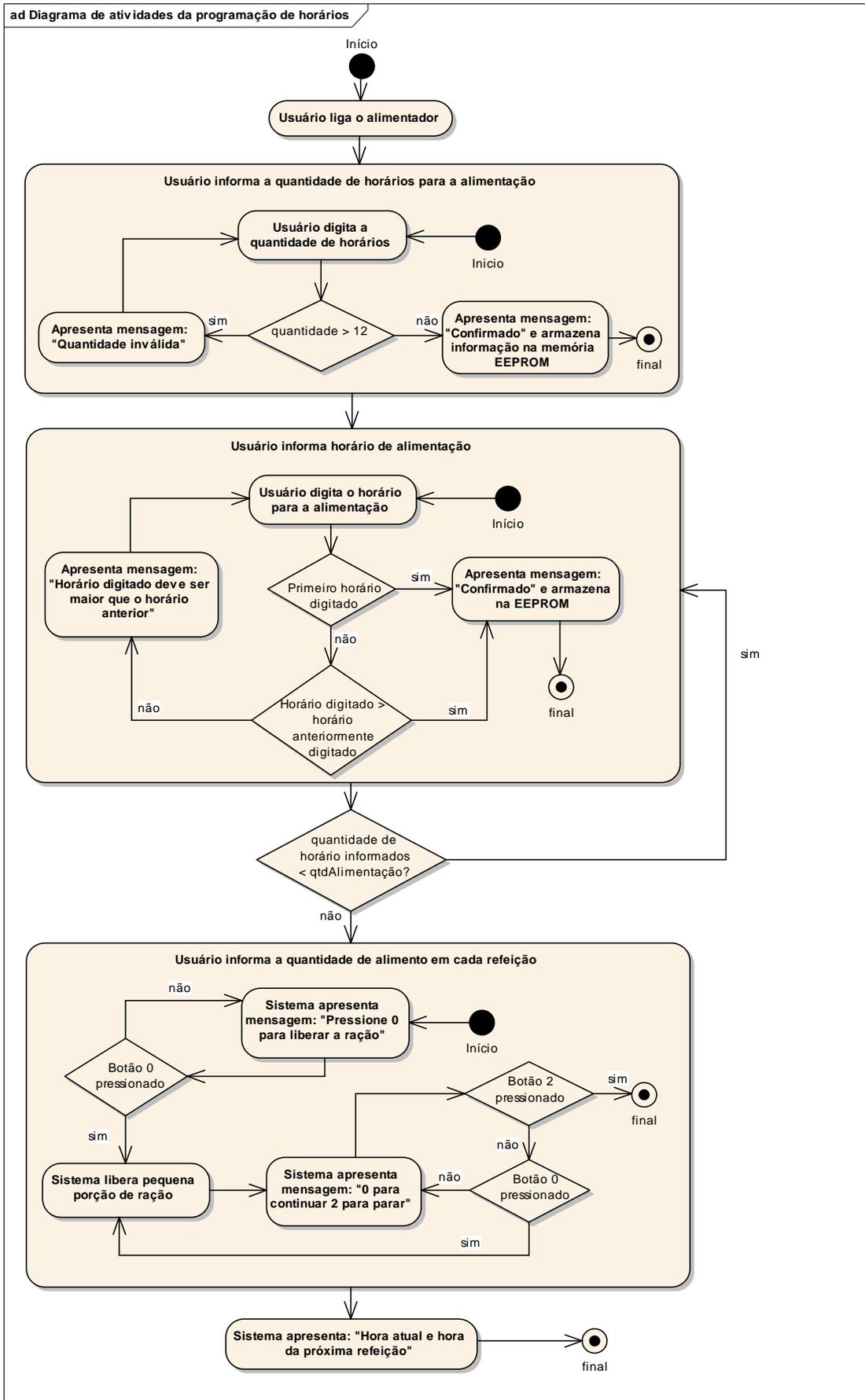


Figura 13 - Diagrama de atividades da programação do alimentador

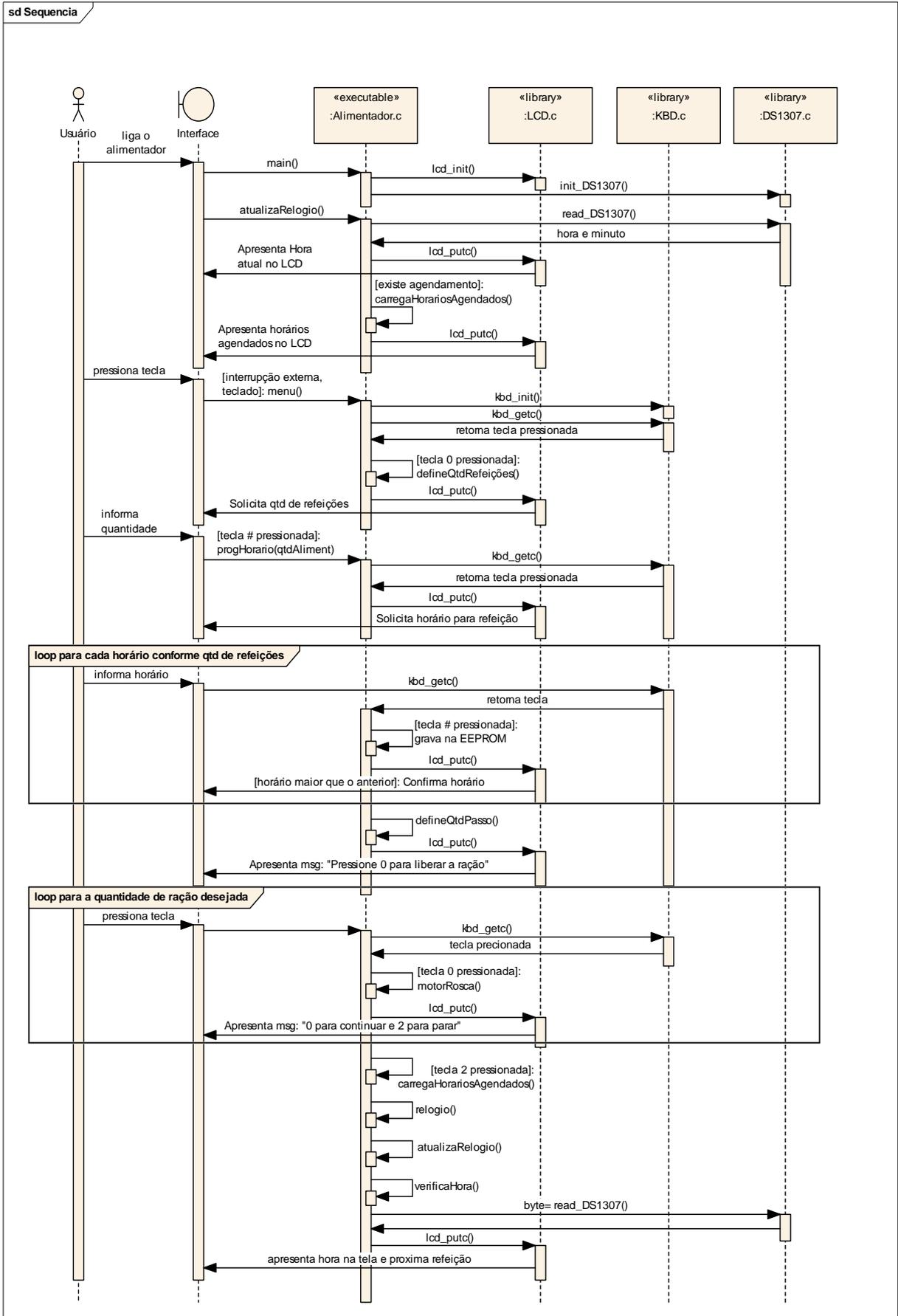


Figura 14 - Diagrama de seqüência da programação do alimentador

Na Figura 15 são apresentadas as ações do sistema após a programação dos horários e definição da quantidade de ração. Neste diagrama foi utilizada uma ação temporal para demonstrar que o processo inicia apenas no horário agendado.

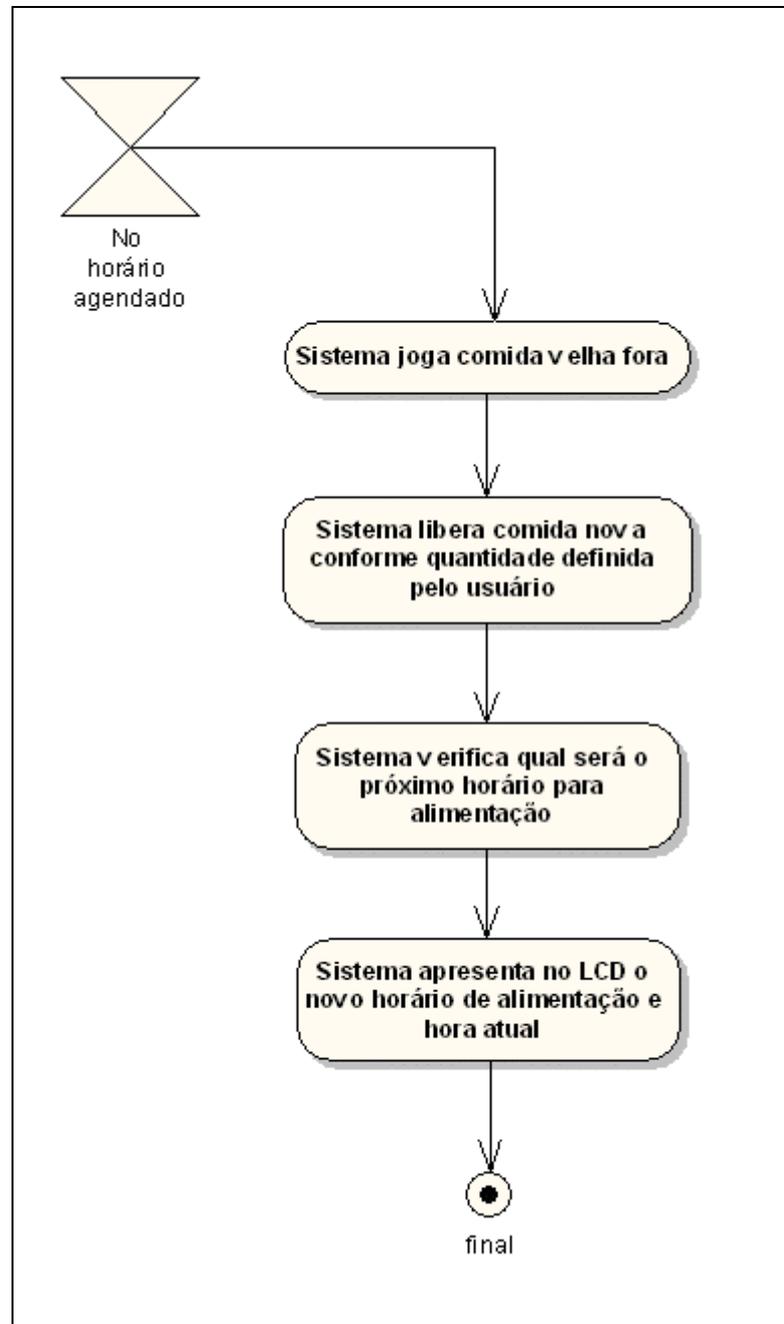


Figura 15 - Diagrama de atividades do processo

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas no desenvolvimento deste

protótipo, serão apresentados trechos do código fonte desenvolvido. Também é demonstrada a operacionabilidade da implementação, através de um estudo de caso.

3.3.1 Técnicas e ferramentas utilizadas

Na implementação do software foi utilizada a linguagem de programação C através do compilador PCW C Compiler IDE (CUSTOM COMPUTER SERVICES, 2006). A gravação do software no microcontrolador foi realizada utilizando a ferramenta IC-PROG (IC-PROG, 2000) em conjunto com o gravador de PIC (MAT'S ELECTRONIC CORNER, 2006) que foi desenvolvido durante este projeto. Desta forma a ferramenta IC-PROG envia o arquivo no formato HEX, devidamente convertido para binário, gerado pelo compilador, através da serial do computador para o microcontrolador PIC16F877.

No desenvolvimento do software foram utilizados os seguintes recursos do microcontrolador:

- a) interrupção interna do `timer0`;
- b) interrupção externa no `PORTB0`;
- c) comunicação paralela com o LCD, teclado e motor de passo;
- d) comunicação I2C com o RTC DS1307;
- e) gravação de dados na memória EEPROM.

Todos os dados informados pelo usuário são gravados na memória de dados interna, EEPROM. E estão organizados da seguinte forma. A quantidade de refeições sempre será armazenada na primeira posição da memória. Já a segunda posição está reservada para a quantidade de ração a ser liberada em cada refeição. Em seguida serão armazenados os horários, sendo que cada horário ocupará dois endereços de memória. Desta forma o primeiro horário a ser armazenado estará distribuído na posição de memória três e quatro. No endereço três tem-se a hora e no próximo endereço estará o minuto e assim sucessivamente até a quantidade de horário determinado pelo usuário.

ordenados de forma crescente, em consequência o *array* também estará ordenado. Desta forma uma vez encontrado o primeiro índice da próxima refeição, pela função `carregaHorariosAgendados()`, será necessário apenas incrementar o índice e dividi-lo pelo número de refeições. O resto desta divisão será o índice da próxima refeição.

```
void verificaHora(){
    if((horaAtual == refeicoes[proximaRefeicao][0]) &&
        (minutoAtual == refeicoes[proximaRefeicao][1])){
        motorFundoFalso();
        motorRosca();
        proximaRefeicao = (proximaRefeicao+1) % nroRefeicoes;
    }
    printf(lcd_putc, "\nProx. ref. %02d:%02d\r", refeicoes[proximaRefeicao][0],
        refeicoes[proximaRefeicao][1]);
}
```

Quadro 4 - Rotina que controla o momento para liberar a ração

3.3.2 Operacionalidade da implementação

Esta seção descreve o funcionamento do protótipo através de um estudo de caso. Para utilizar o equipamento é necessário ligá-lo a uma fonte de energia. Ao ligar o alimentador será apresentado no *display* LCD a seguinte seqüência de informações: quantidade de refeições programadas, o horário de cada refeição e por fim a hora atual e o horário da próxima refeição (Figura 16).

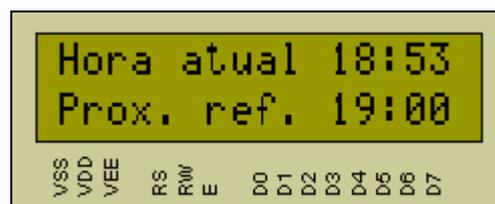


Figura 16 - Tela com informações iniciais

Para realizar a configuração da quantidade de refeições e seus respectivos horários o usuário deverá pressionar qualquer tecla do teclado. Após esta ação será apresentada uma mensagem no display LCD (Figura 17).

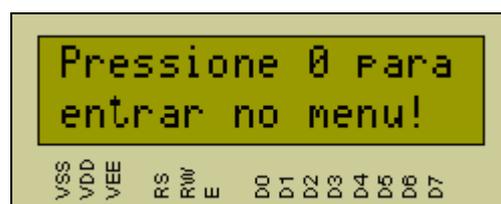


Figura 17 - Tela de entrada

Caso o usuário não pressione a tecla “0” o sistema voltará a apresentar as informações da Figura 16. Porém na situação em que a tecla “0” tenha sido pressionada o sistema solicitará

ao usuário que informe a quantidade de refeições a ser disponibilizada ao animal. Esta tela pode ser vista na Figura 18. O usuário terá sempre a opção de corrigir os dados digitados pressionando a tecla “*” para apagar. Esta ação é válida enquanto o usuário não tenha pressionado a tecla “#”, tecla responsável por gravar a informação na memória EEPROM do microcontrolador.

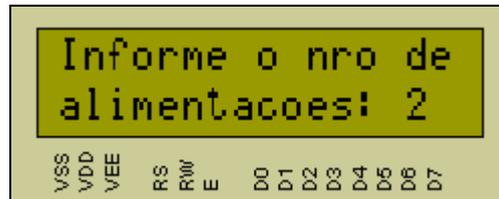


Figura 18 - Tela para informar a quantidade de refeições

Após ter informado a quantidade de refeições o sistema solicitará a respectiva quantidade em horários para a alimentação, conforme apresentado nas Figura 19 e Figura 22. Neste momento o usuário deverá informar horários de forma crescente, ou seja, não poderá informar o horário da seguinte forma: 12 horas e depois 9 horas (Figura 20). Caso o usuário tente fazer isso o sistema irá apresentar a mensagem conforme Figura 21.

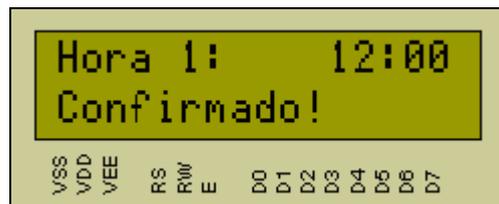


Figura 19 - Tela para entrada do primeiro horário de refeição

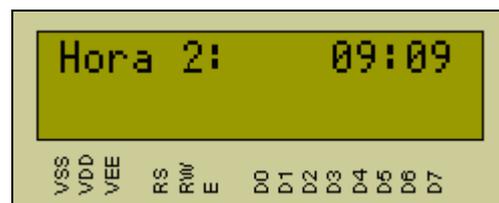


Figura 20 - Tela com horário 2 menor que o horário 1



Figura 21 - Tela de aviso para horário inválido

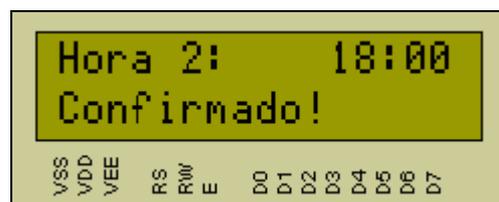


Figura 22 - Tela com o segundo horário de refeição

Ao término da digitação dos horários chega o momento de definir a quantidade de ração que deverá ser liberada a cada refeição. Esta quantidade será a mesma em todos os horários. Para iniciar este processo o sistema solicitará que a tecla “0” seja pressionada para que a ração seja liberada (Figura 23).

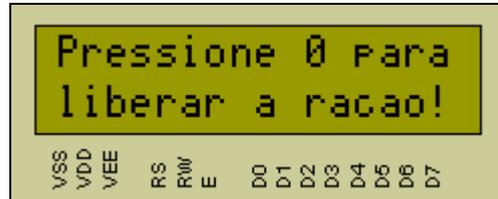


Figura 23 - Tela para acionar a liberação da ração

Ao pressionar a tecla “0” será liberada uma pequena quantidade de ração, durante este processo o display apresentará a mensagem conforme a Figura 24.

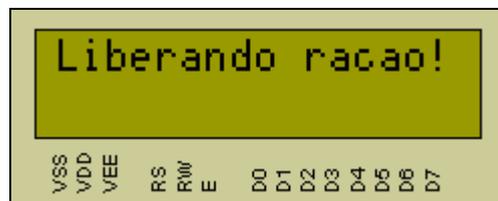


Figura 24 - Tela do momento da liberação da ração

Após a liberação desta pequena quantidade o usuário terá a opção de liberar mais ração ou parar, caso esteja satisfeito. Este processo de liberação de ração poderá se repetir várias vezes (Figura 25). Ou seja, ao final deste processo o sistema saberá quantas porções deverão ser liberadas em cada refeição.

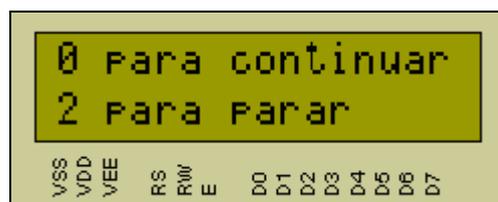


Figura 25 - Tela para determinar a quantidade de ração

Ao fim do agendamento o sistema voltará a apresentar a hora atual e a próxima refeição. Conforme dados informados nesse processo (Figura 26).

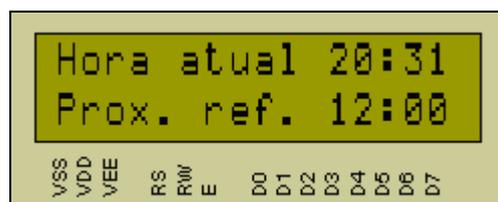


Figura 26 - Tela com o horário atual e a próxima refeição agendada

3.4 RESULTADOS E DISCUSSÃO

Em um primeiro momento tentou-se utilizar a estrutura de um alimentador por gravidade disponível no mercado. Esta reutilização não foi possível devido a falta de espaço para acoplar a rosca sem fim e os outros componentes utilizados pelo alimentador. Outro fator que dificultou a reestruturação foi o fato da maioria dos alimentadores serem de plástico, e o plástico não é um material moldável. Por estes motivos decidiu-se criar uma nova estrutura. Esta foi desenvolvida de forma a suprir praticamente todas as necessidades para um bom desempenho dos mecanismos acoplados no protótipo (Figura 27).



Figura 27 - Estrutura do alimentador

O microcontrolador se demonstrou eficiente no que diz respeito ao controle dos periféricos e interrupções. O inconveniente apresentado por este microcontrolador foi a lentidão no processo de leitura e escrita dos dados na memória EEPROM. Por este motivo,

optou-se por armazenar os dados lidos da EEPROM em um *array*, para que o sistema interagisse apenas com este *array*, tornando-o mais rápido. Porém, esta solução acabou limitando a quantidade de refeições devido ao pouco espaço de memória RAM disponível no PIC16F877.

O teclado, apesar de ser um periférico simples, acabou apresentando várias dificuldades em seu controle. Ao pressionar qualquer tecla o sistema tem que parar todo o seu controle e atender ao teclado. Este acionamento acabou sendo realizado através de uma interrupção externa no pino RB0. Porém, após esta interrupção, se o usuário estiver fazendo um reagendamento ele ainda estará interagindo com o teclado e este não poderá mais gerar interrupção. Desta forma, após gerar esta interrupção foi necessário desabilitar as interrupção e tratar o teclado de forma normal, onde utilizou-se a biblioteca kbd para o controle do mesmo.

Os motores se mostraram eficientes para o controle dos passos, porém até se obter os movimentos desejados foram enfrentados vários obstáculos, por exemplo, ocorreram perdas de circuito e uma grande dificuldade em conseguir a seqüência dos fios para o controle dos motores bipolares. Isso se deve ao fato de terem sido usados motores reciclados e também da falta de documentação, já que não existe um padrão de cores para os fios. Para o controle da rosca foi necessário o uso de motores potentes, pois o peso da ração exigiu um melhor desempenho do mesmo. Já a rosca sem fim se mostrou adequada para os objetivos propostos.

O DS1307 também demandou um certo tempo para que se conseguisse controlá-lo de forma eficiente. Para um bom desempenho do sistema foi necessário utilizar a interrupção do timer0 a cada trinta segundos para as chamadas da função que atualiza a hora no LCD.

Apesar do Proteus não ter obtido um desempenho real durante as simulações, o mesmo foi essencial para o desenvolvimento deste trabalho, sendo possível através dele fazer todos os testes necessários para a conclusão deste projeto.

Manteufel (2001) apresenta um dosador de matéria-prima onde foram utilizados motores comuns para o acionamento das esteiras. A utilização de motores DC é justificável nesta situação uma vez que a matéria-prima utilizada é pesada, havendo a necessidade de motores realmente potentes e fortes.

Bernardes (2004, p. 12) apresenta uma máquina eletrônica para venda de Trident, baseada em uma espiral, um motor DC e um *switch* de posicionamento de volta final. Neste projeto Bernardes poderia ter utilizado apenas um motor de passo ao invés do motor DC e do *switch* de posicionamento. O motor de passo funcionaria perfeitamente pois como os Trident estão entre cada elo do espiral basta que o mesmo dê um giro completo e este controle é

facilmente obtido por um motor de passo.

4 CONCLUSÕES

Este trabalho apresentou um protótipo de um alimentador para auxiliar as pessoas que possuem animais de estimação na nutrição de seus bichanos. Como foi visto no presente trabalho os animais de estimação também possuem seus horários regrados para refeições. Tanto o objetivo principal como os objetivos específicos propostos por este trabalho foram plenamente atingidos. Em quase sua totalidade, as ferramentas utilizadas foram adequadas para o êxito do mesmo. A exceção ficou por conta do *Proteus* que, não conseguindo fazer uma simulação real, deixou todo o processo lento.

A principal vantagem deste projeto em relação aos alimentadores existentes no mercado é o fato de que o mesmo não está limitado a liberar comida apenas de tempos em tempos. O alimentador proporciona a liberdade ao usuário de escolher a quantidade de refeições e os respectivos horários de liberação de alimento, não tendo a necessidade de seguir intervalos lógicos ou mesmo respeitar horários inteiros, dando assim total comodidade ao usuário. Uma outra grande qualidade é o fato do alimentador possuir um mecanismo que retira do prato a eventual sobra de ração da última refeição antes que uma nova porção seja liberada.

A desvantagem observada está na limitação da memória RAM do microcontrolador. Este fato limitou a quantidade de refeições em doze por dia.

4.1 EXTENSÕES

Como sugestão para futuros trabalhos:

- a) utilizar um LCD gráfico para melhorar a disposição das informações e facilitar a interação homem-máquina;
- b) utilizar a linguagem Java na implementação do fonte;
- c) estender o presente trabalho para o controle de água;
- d) possibilitar todo o processo de programação dos agendamentos de horários através da internet.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGUIAR, C. E.; LAUDARES, F. Aquisição de dados usando Logo e a porta de jogos do PC. **Revista Brasileira de Ensino de Física**, São Paulo, v. 23, n. 4, 2001. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-47442001000400003>. Acesso em: 18 abr. 2007.
- ALBANO, L. L. M. **Saúde animal**: aspectos importantes da nutrição canina. São Carlos, [2007]. Disponível em: <http://www.saudeanimal.com.br/artigo_luigi_nutricao010.htm>. Acesso em: 05 mar. 2007.
- BERNARDES, L. H. C. Construindo uma “vending machine” com gerenciamento remoto via GSM. **Saber Eletônica**, São Paulo, n. 381, p. 11-16, out. 2004.
- CENTRO SENAI FUNDAÇÃO ROMI FORMAÇÃO DE FORMADORES. **Fundamento de automação industrial**: tutorial. [Sta. Bárbara d’Oeste], [2000?]. Disponível em: <http://www.senaiformadores.com.br/Cursos/01/unidade/uni3_aut4.htm>. Acesso em: 07 nov. 2006.
- COSTA, A. **Robótica**: motor de passo. [S.l.], 2000. Disponível em: <http://www.mrshp.hpg.ig.com.br/rob/m_passo.htm>. Acesso em: 22 ago. 2006.
- CUSTOM COMPUTER SERVICES, inc. **C Compilers**: compilers for microchip PIC MCUs. [S.l.], 2006. Disponível em: <<http://www.ccsinfo.com/content.php?page=compilers>>. Acesso em: 07 nov. 2006.
- DAMASCENO, M. G. Teste microcontrolado de motores TTM. **Mecatrônica Fácil**, São Paulo, n. 27, p. 28-33, mar./abr. 2006.
- FERRAZ NETTO, L. **Motores elétricos**. [Barretos], [2005]. Disponível em: <http://www.feiradeciencias.com.br/sala22/motor_teorial.asp>. Acesso em: 20 ago. 2006.
- FERREIRA, G. F.; PURGER NETO, H.; BUENO JUNIOR, M. S. L. **Câmera Pan-Tilt**. [Curitiba], [2005]. Disponível em: <<http://www.ppgia.pucpr.br/~santin/ee/2005/1s/5/>>. Acesso em: 21 ago. 2006.
- FOTODIODO. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <<http://pt.wikipedia.org/wiki/Fotodiodo>>. Acesso em: 09 nov. 2006.
- GONÇALVES, L. M. G. **Sensores**. Natal, 2003. Disponível em: <<http://www.dca.ufrn.br/~lmarcos/courses/robotica/notes/sensores.doc>>. Acesso em: 09 nov. 2006.

IC-PROG. **Prototype programmer**. [S.l.], [2000?]. Disponível em: <<http://www.ic-prog.com>>. Acesso em: 17 maio 2007.

JORNAL ELETRÔNICO NOVO MILÊNIO. **RTC**. Santos, 2001. Disponível em: <<http://www.novomilenio.inf.br/glossar/eglossr.htm>>. Acesso em: 17 set. 2006.

LEÃO, F. L. et al. **Estudo do motor de passo e seu controle digital**. [Rio de Janeiro], [2003]. Disponível em: <http://www.gta.ufrj.br/grad/01_1/motor/>. Acesso em: 20 ago. 2006.

MANTEUFEL, S. F. **Protótipo de um dosador de matéria-prima para pequenas empresas de produtos pré-fabricados de concreto**. 2001. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MARTINS, L. F. Humm, come tudo...: a alimentação ideal para o seu pet estar sempre bem. **Almanaque Gatos & Raças 2005**, São Paulo, n. 3, p. 22-25, 2005.

MAT'S ELECTRONIC CORNER. **Pic programmer**. [S.l.], [2006?]. Disponível em: <<http://matmart0.tripod.com/picpic.htm>>. Acesso em: 10 mar. 2007.

MESSIAS, A. R. **Controle de motor de passo através da porta paralela**. São Paulo, [2007?]. Disponível em: <<http://www.rogercom.com/>>. Acesso em: 08 abr. 2007.

MICROCONTROLLER PROS CORPORATION. **DS1307 real-time clock**. [S.l.], 2003. Disponível em: <http://microcontrollershop.com/product_info.php?products_id=1046>. Acesso em: 17 set. 2006.

MOGIANA ALIMENTOS. **Dicas para o dia-a-dia com seu pet**. [S.l.], [2005?]. Disponível em: <<http://www.guabi.com.br/pet/caes/dicas.asp>>. Acesso em: 05 mar. 2007.

ORSINI, S. **Mercado aposta em animais de estimação**. [S.l.], 2004. Disponível em: <<http://financas.cidadeinternet.com.br/article.asp?878~196264>>. Acesso em: 10 ago. 2006.

PEREIRA, F. **PIC: programação em C**. São Paulo: Érica, 2006.

PIC16F877. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <<http://pt.wikipedia.org/wiki/PIC16F877>>. Acesso em: 20 ago. 2006.

QUEIROZ, R. A. A. **Motores de passo**. [2003?]. Relatório integrante do projeto de pesquisa AxeBot, Departamento de Ciência da Computação, Universidade Federal da Bahia, Salvador. Não paginado. Disponível em: <<http://www.axebot.ufba.br/download.htm>>. Acesso em: 12 abr. 2007.

RADIOAMADORES.NET. **Display**. [S.l.], 2003. Disponível em: <<http://www.radioamadores.net/displays.htm>>. Acesso em: 21 ago. 2006.

RUEDA, T. C.; SANTOS, L. M.; RABELO, G. F. Controle automático do nível de baseado em microcontrolador. **INFOCOMP Journal of Computer Science**, Lavras, v. 5, n. 1, Mar. 2006. Não paginado. Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v5.1/art08.pdf>>. Acesso em: 18 abr. 2007.

SAUL, C. K.; FREITAS, F. C. Controle de motor de passo pela porta paralela do PC. **Mecatrônica Fácil**, São Paulo, n. 30, p. 44-47, set./out. 2006.

SERWAY, R. A.; JEWETT JUNIOR, J. W. **Princípios de física: mecânica clássica**. São Paulo: Thomson, 2004.

SILVA, R. A. **Programando microcontroladores**. São Paulo: Ensino Profissional, 2006.

SOARES, M. J. Microcontroladores PIC: varredura de teclados. **Mecatrônica Fácil**, São Paulo, n. 11, p. 56-58, jul./ago. 2003.

_____. Display serial: para uso com microcontroladores e dispositivos com comunicação RS-232. **Mecatrônica Fácil**, São Paulo, n. 29, p. 34-41, jul./ago. 2006.

_____. Três circuitos práticos com PIC16F62A. **Eletrônica Total**, São Paulo, n. 101, p. 2-7, jul./ago. 2004.

SOUZA, D. J. **Desbravando o PIC**: ampliado e atualizado para PIC16F628A. 9. ed. São Paulo: Érica, 2005.

SOUZA, D. J.; LAVINIA, N. C. **Conectando o PIC 16F877A**: recursos avançados. São Paulo: Érica, 2003.

SOUZA, U. L.; PEREIRA, T. R. **Fotodiodos e fototransistores**. [Curitiba], 2004. Disponível em: <<http://www.eletrica.ufpr.br/piazza/materiais/Uilian&Thiago.pdf>>. Acesso em: 18 abr. 2007.

SPA. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <<http://pt.wikipedia.org/wiki/Spa>>. Acesso em: 17 set. 2006.

TECNOTRI. **Pet Feeder**: alimentador automático para animais. Marau, [2005?]. Disponível em: <<http://www.petfeeder.com.br/estrutura2.htm>>. Acesso em: 28 out. 2006.

VALLADÃO, C. **Optoeletrônica**. Rio de Janeiro, [2006]. Disponível em: <http://www.teletronica.kit.net/downloads/apgb_elo2_cap04_optoeletronica.doc>. Acesso em: 15 abr. 2007.

WING FEEDER: alimentador automático para animais. [S.l.], [2006?]. Disponível em: <<http://www.wingfeeder.com.br/?act=1>>. Acesso em: 28 out. 2006.

ZANCO, W. S. **Microcontroladores PIC**: técnicas de software e hardware para projetos de circuitos eletrônicos com base no PIC16F877A. São Paulo: Érica, 2006.