

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DO CURSO DE ENGENHARIA ELETRÔNICA
ENGENHARIA ELETRÔNICA**

GUSTAVO CRESTANI RAMBO

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE PARA
UM DISTRIBUIDOR DE SEMENTES**

TRABALHO DE CONCLUSÃO DE CURSO

TOLEDO

2015

GUSTAVO CRESTANI RAMBO

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE PARA
UM DISTRIBUIDOR DE SEMENTES**

Trabalho de Conclusão de Curso de Graduação apresentado na disciplina de Trabalho de Conclusão de Curso 2 (TCC2), como requisito parcial para obtenção do título de Engenheiro Eletrônico, da Coordenação do Curso de Engenharia Eletrônica, da Universidade Tecnológica Federal do Paraná – UTFPR, Campus Toledo.

Orientador: Prof. Dr. Gerson Filippini

TOLEDO

2015



TERMO DE APROVAÇÃO

Título do Trabalho de Conclusão de Curso N^o 029

Desenvolvimento de um Sistema de Controle para um Distribuidor de Sementes

por

Gustavo Crestani Rambo

Esse Trabalho de Conclusão de Curso foi apresentado às 10:20 h do dia **26 de novembro de 2015** como requisito parcial para a obtenção do título **Bacharel em Engenharia Eletrônica**. Após deliberação da Banca Examinadora, composta pelos professores abaixo assinados, o trabalho foi considerado **APROVADO**.

Prof. Dr. Evandro Marcos Kolling
(UTFPR-TD)

Profa. M. Jaqueline Vargas
(UTFPR-TD)

Prof. Dr. Gerson Filippini
(UTFPR-TD)
Orientador

Visto da Coordenação

Prof. M. Alessandro Paulo de Oliveira
Coordenador da COELE

LISTA DE ILUSTRAÇÕES

Figura 1 - Vista Superior do Distribuidor de Sementes da Empresa.	18
Figura 2 - Vista Lateral do Distribuidor de Sementes da Renz e Cia.	18
Figura 3 - Disco Perfurado Inclinado	20
Figura 4 - Distribuidor de Sementes por Dedos Prensos 21	21
Figura 5 - Sistema Pneumático	22
Figura 6 - Sistema para funcionamento do controlador no distribuidor de sementes.....	24
Figura 7 - Microcontrolador PIC16F877	26
Figura 8 - Motor Unipolar de 8 fios	29
Figura 9 - Motor de Passo Bipolar 8 fios Ligação Meio-Passo	30
Figura 10 - Motor de Passo Bipolar 8 fios Ligação em Série	30
Figura 11 - Motor de Passo Bipolar 8 fios Ligação em Paralelo.....	31
Figura 12 - Sinal de Pulso a ser gerado para melhor controle do Motor de Passo	32
Figura 13 - Gerador de Pulso agregado junto com o <i>Driver</i> e Controlador	32
Figura 14 - Gerador de Pulso acoplado com o Controlador	33
Figura 15 - Motor de Passo Akyiama AK85H/6-1.8.....	33
Figura 16 - Características do Motor de Passo AK85H/6-1.8.....	34
Figura 17 - Driver de Motor de Passo AKDMP5-5.0A	35
Figura 18 - Transformador 12 V 1 A, 60 V 5 A.	36
Figura 19 - LED Infravermelho TIL32	38
Figura 20 - Fototransistor Infravermelho TIL78	38
Figura 21 - Resposta justificada a esquerda ou a direita	40
Figura 22 - Exemplo de Display LCD 16x2	40
Figura 23 - Teclado matricial 4x4	42
Figura 24 - Lógica do Sistema aplicado a AP.....	44
Figura 25 - Tempo de sucção dos grãos, para que A chegue em B.	47
Figura 26 - Fluxograma do Sistema Controlador do Distribuidor de Sistemas. 48	48
Figura 27 - Curva de Compensação criada para o sistema	49
Figura 28 - Placa de ensaio usada inicialmente no controlador do sistema.....	52
Figura 29 - Modelo da Placa Ilha utilizada para o protótipo do sistema	52
Figura 30 - Protótipo usado para o Controlador do Distribuidor de Sementes . 53	53
Figura 31 - Sensor emissor-receptor IR instalado.....	54
Figura 32 - Ligação do <i>Driver</i> e do Motor de Passo	55
Figura 33 - Ligações do Teclado 4x4 e do <i>Display</i> LCD	55

LISTA DE TABELAS

Tabela 1 - Descrição das Funções dos Pinos do Display LCD	41
Tabela 2 - Resultados na empresa Renz e Cia.....	57
Tabela 3 - Resultados em Laboratórios da universidade	58

LISTA DE SIGLAS, ABREVIATURAS E ACRÔNIMOS

A	<i>Ampéres</i>
A/D	<i>Analógico-digital</i>
CA	<i>Tensão alternada</i>
CC	<i>Tensão continua</i>
cm	<i>Centímetros</i>
IR	<i>Infravermelho</i>
k Ω	<i>Quilo Ohms</i>
kgf.cm	<i>Quilograma-força centímetro</i>
km/h	<i>Quilômetros por hora</i>
LCD	<i>Liquid-Crystal Display</i>
LED	<i>Light Emittion Diode</i>
MHz	<i>MegaHertz</i>
mA	<i>MiliAmpéres</i>
ms	<i>Milissegundos</i>
nm	<i>Nanômetros</i>
ρ F	<i>PicoFaraday</i>
PWM	<i>Pulse-Width Modulation</i>
RPM	<i>Rotações por minuto</i>
μ s	<i>Microsssegundos</i>
V	<i>Volts</i>

SUMÁRIO

1	INTRODUÇÃO	11
2	OBJETIVOS	13
2.1	OBJETIVO GERAL	13
2.2	OBJETIVO ESPECÍFICO	13
3	JUSTIFICATIVA	15
4	REFERENCIAL TEÓRICO	16
4.1	EMPRESA RENZ E CIA.	16
4.2	DISTRIBUIDOR DE SEMENTES DA EMPRESA RENZ E CIA.	17
4.3	TIPOS DE DISTRIBUIDORES DE SEMENTES EXISTENTES	19
4.3.1	Distribuidores de sementes do tipo Discos Perfurados	19
4.3.2	Distribuidores de sementes do tipo Dedos Prensos	20
4.3.3	Distribuidores de sementes por um Sistema Pneumático de Pressão	21
4.4	INTERFACE USUÁRIO X MÁQUINA	22
4.5	SENSORES DE RUPTURA DE FEIXE	22
4.6	MICROCONTROLADORES	23
5	MATERIAIS E MÉTODOS	24
5.1	MICROCONTROLADOR PIC16F877A	25
5.2	MOTOR DE PASSO	28
5.2.1	Motor de Passo AKYIAMA AK85H/6-1.8	33
5.2.2	Driver Motor de Passo AKYIAMA AKDMP5-5.0A	35
5.3	SENSOR EMISSOR-RECEPTOR INFRAVERMELHO	37
5.3.1	LED Infravermelho - TIL32	37
5.3.2	Fototransistor Infravermelho - TIL78	38
5.4	DISPLAY LCD	40
5.5	TECLADO MATRICIAL 4x4	42
6	METODOLOGIA	44
6.1	SOFTWARE PARA PROGRAMAÇÃO	45
6.2	LÓGICA DE PROGRAMAÇÃO	46

6.3	MONTAGEM DO PROTÓTIPO DO SISTEMA.....	51
7	RESULTADOS	56
7.1	CONTROLE TOTAL DO DISTRIBUIDOR DE SEMENTES	56
7.2	COMPENSAÇÃO DO ERRO NO DISTRIBUIDOR DE SEMENTES...	59
7.3	PROJETOS FUTUROS.....	60
8	CONSIDERAÇÕES FINAIS	61
9	REFERÊNCIAS BIBLIOGRÁFICAS	62
	APÊNDICE A – Simulação Proteus ISIS do Sistema de Controle	66
	APÊNDICE B – Código das Funções de Interrupção para o controle do motor de passo.....	67
	APÊNDICE C – Código das Funções para conversão AD e UART	68
	APÊNDICE D – Código de controle do Display LCD	71
	APÊNDICE E – Leitura do Teclado matricial 4x4	76
	APÊNDICE F – Código principal para o Controle do Distribuidor de Sementes	78

RESUMO

RAMBO, Gustavo. **Desenvolvimento de um sistema de controle de um distribuidor de sementes**. 2015. 82f. Trabalho de Conclusão de Curso 2 (Bacharelado em Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná. Toledo, 2015.

Este trabalho apresenta um sistema de controle para um distribuidor de sementes, aplicado a agricultura de precisão em parceria com a empresa Renz e Cia. de Cascavel – PR. Mesmo possuindo grande eficiência quanto à distribuição das sementes no solo por meio de um distribuidor de sementes com sistema a vácuo, erros oriundos da vinda de mais de uma semente ou pela falta de sementes no solo ainda acontecem e devem ser tratados. Com o arquivamento destes dados e a apresentação dos erros por meio de uma interface de exibição, este projeto buscou com o monitoramento e automação de alguns processos, compensar os erros existentes pela falta de semente e controlar a velocidade no motor de passo do distribuidor, assim como apresentar informações convenientes ao usuário.

Palavras-Chave: Agricultura de Precisão. Distribuidor de Sementes. Sistema a Vácuo.

ABSTRACT

RAMBO, Gustavo. **Development of a control system of a distributor of seeds**. 2015. 82f. Trabalho de Conclusão de Curso 2 (Bacharelado em Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná. Toledo, 2015.

This work presents the proposal to make a rewarding seed Distributor, applied to precision agriculture in partnership with Renz & Cia. of Cascavel – PR, even possessing great efficiency of grain distribution of seeds in the soil through a seed distributor with vacuum system, errors from the coming of more of a seed or seeds in the soil still happen and should be treated. With the archiving of this data and the presentation of errors through an interface, this project seeks with the monitoring and automation of some processes, compensate for the errors due to lack of seed and control your speed on the stepper motor Distributor, as well as presenting information convenient to the user.

Keywords: precision agriculture. distributor of seeds. vacuum system.

1 INTRODUÇÃO

A Agricultura de Precisão (AP) começou a receber seus primeiros relatos no ano de 1929, mas somente a partir da década de 80 que obteve seu destaque, devido ao avanço do sistema informatizado, o melhoramento de sistemas de informações geográficas e do monitoramento da colheita, (EMBRAPA, 2015).

No Brasil, este ramo teve no início dos anos 90 suas primeiras inserções no mercado, onde foram embarcados nas máquinas agrícolas os receptores GNSS (Global Navigation Satellite System), computadores de bordo e sistemas que geravam o mapeamento da produtividade. Sua principal finalidade é aumentar o retorno econômico, buscar sustentabilidade e diminuir os efeitos ao meio ambiente. Hoje, grandes empresas nacionais e internacionais instaladas no Brasil investem na agricultura de precisão, tendo em vista o grande apelo dos agricultores pela melhor utilização de sua terra, proporcionando um aumento da sua produção e conseqüentemente um ganho financeiro, (STARA, 2015).

Os Distribuidores de Sementes ao longo desses anos também sofreram diversas mudanças. O que antes era feito a mão, com semeadoras manuais muito usadas no início dos anos 70, passando pelas semeadoras por tração animal até as primeiras tracionadas por trator, hoje pode ser automatizado e controlado por sensores que trazem a informação ao usuário por um computador de bordo ou por outro mecanismo, rapidamente.

Atualmente, os distribuidores de sementes mais usados são: discos perfurados, dedos prensos e o sistema pneumático com a utilização do vácuo.

Este último tipo de distribuidor de sementes citado, é a mais nova tecnologia desse mercado fazendo com que sua eficiência fique próxima a 100%, quando falamos na uniformidade na distribuição destas sementes em relação ao solo.

Mesmo com este alto índice de efetividade há muitos obstáculos a serem superados, como o processamento dos dados de todas as sementes. Com um tratamento correto dos erros que acontecem durante o processo, como possíveis vindas de mais de uma semente ou falta de semente em uma determinada área, poderão servir como alertas para o usuário de um possível erro constante ou de pequenas avarias que podem estar ocorrendo.

Este processo de armazenamento dos erros segue ainda sendo melhorado pelos fabricantes, com desafios a serem resolvidos, sendo possível o melhoramento dos sistemas já criados, assim como a criação de outros.

Tendo como base isto, o sistema de controle que foi criado neste projeto possui um microcontrolador, da família PIC para realização desta automação no qual irá conter informações para o monitoramento de todo o processo.

Estas informações serão usadas para qualificar o distribuidor de sementes e para falhas que acontecerem no processo, onde caso nenhuma semente for succionada para colocação no disco do distribuidor de sementes, será criado um compensador que irá aumentar a velocidade de rotação do motor de passo do distribuidor de sementes, para que assim não haja discrepância entre os espaçamentos das sementes no solo. O usuário também poderá visualizar estas informações por meio de uma interface.

2 OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo deste trabalho é aperfeiçoar o processo de um distribuidor de sementes que possui um sistema á vácuo de distribuição em uma semeadora através de um sistema de controle, monitoramento e automação, usando a semente de soja, em parceria com a Empresa Renz e Cia. , de Cascavel - PR.

2.2 OBJETIVO ESPECÍFICO

O sistema de controle criado irá coletar de modo autônomo as informações, de todas as informações relevantes no processo, tais como: a quantidade de sementes a ser despejada em cada linha¹, a quantidade de vezes que nenhuma semente foi lançada ao solo, a velocidade no motor de passo do distribuidor e também indicando por meio de um sensor, a passagem de sementes no disco do distribuidor de sementes. Com isso, possuímos um monitoramento de todo o processo.

Será inserido também, para uma melhor visualização do processamento de dados, sinais de controle, no qual irão avisar ao usuário os erros que estão acontecendo durante o processo. Caso este erro se mantenha constante, o sistema avisa o usuário por meio de alarmes que serão ativados fazendo com que esta falha possa ser corrigida ou diminuída.

O uso do microcontrolador PIC 16F877A² se deu devido a suas diversas estruturas, no qual será possível para todo o sistema criado, ser configurado através de seus periféricos, tornando o projeto mais simples, ou seja, apenas informações relevantes ao controle do processo serão inseridas no

¹ Linha é o número de sementes que tem de ser descarregada em uma determinada distância pré-definida.

² PIC: família de microcontrolador criado pela *Microchip Technology* possuindo 8, 16 ou 32 bits de dados.

microcontrolador, não colocando informações e periféricos que não estarão em uso neste projeto.

3 JUSTIFICATIVA

Tendo em vista o crescente mercado agrícola e o constante crescimento da eletrônica nas últimas duas décadas, vemos que para um melhor rendimento da produção agrícola é necessário que ocorra a combinação destes dois fatores.

O uso da eletrônica no mercado agrícola aumenta a cada ano, estando ainda em um processo de consolidação no Brasil, porém com um nível de adoção ainda relativamente baixo. Grandes empresas brasileiras do ramo agrícola já aderiram a esta combinação, e hoje além de estarem fabricando produtos eletrônicos para agricultura, já estão exportando, (MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO, 2013).

A Eletrônica embarcada juntamente com os computadores de campo, é o grande exemplo disto e vem ganhando força neste cenário. Seu objetivo é aumentar o rendimento da produção agrícola e fornecer ao agricultor o maior número de informações do que está acontecendo em seu ambiente de trabalho. Dessa forma erros e quaisquer inconvenientes podem ser detectados facilmente, (SILVA, 2012).

A semeadora onde está localizado o distribuidor de sementes trabalha em parcelas, ou seja, a área para semeadura é dividida em varias partes, podendo ter diferentes tamanhos entre uma e outra, para que assim diferentes valores entre a distribuição das sementes em relação ao solo sejam feitas, para se saber qual destas áreas foi a mais produtiva. Com isso, o sistema de controle, a detecção de erros e o controle rígido do número de sementes aumentará a certeza dos dados e fornecerá ao usuário estatísticas da produtividade com maior nível de certeza.

Com o propósito de agregar à eletrônica e melhorar o processo do distribuidor de sementes, que até então era composto apenas pelo sistema mecânico, este projeto tem como finalidade diminuir os erros e falhas recorrentes no processo, com um baixo custo, contribuindo para a inserção do produto no mercado.

4 REFERENCIAL TEÓRICO

Com a finalidade de ajudar no entendimento do que será usado neste projeto, esta seção apresenta alguns dos componentes que foram usados para o controle do projeto, além de mostrar o quão efetivo já é o distribuidor de sementes da empresa Renz e Cia., assim como seu modo de funcionamento e explicar, citando alguns tipos de distribuidores de sementes, para que uma maior compreensão do sistema á vácuo usado na empresa seja feito.

4.1 EMPRESA RENZ E CIA.

A empresa Renz e Cia. Ltda. está localizada na Rua Pedro Ivo, 1339, em Cascavel - PR. Iniciou suas atividades no ano de 1978, tendo como seus principais ramos de atividade a mecânica industrial, tornearia e o comércio de peças.

A empresa surgiu composta da divisão de uma sociedade composta por dois irmãos sendo fundada em 06/09/1978. Seu objetivo era prestar serviços de manutenção para o setor agrícola, de mecânica, solda, torno, ferraria e plaina.

Atualmente, composta por 21 funcionários, possui como principais produtos de comercialização os carrinhos de carga industrial, forrageiras, polia de ferro fundido, solda, torno, base para desintegrados para adaptar em trator, entre outros e tem como seu objetivo auxiliar o consumidor em problemas mecânicos, agrícolas ou industriais.

4.2 DISTRIBUIDOR DE SEMENTES DA EMPRESA RENZ E CIA.

Os *kits* de distribuição de sementes de soja para semeadura de parcelas podem ser por volume fixo de sementes, ou por volume de semeadura pré-determinado de sementes, denominado semeadura por população, que tem como sua variável a quantidade de sementes colocada no distribuidor.

Este tipo de distribuidor pode variar tanto a quantidade das sementes na linha, como variar dentro de cada linha a quantidade de sementes para qualquer tipo de situação quanto ao nível do solo, ou também alternar a quantidade de sementes em todas as linhas.

Todos os sistemas de semeadura de parcelas de qualquer empresa possuem praticamente os mesmos itens básicos. No caso da Renz e Cia., o melhoramento de uma tecnologia primária já existente no mercado mundial, para semeadura de sementes está se transformando em um produto de alta tecnologia, para semeadura de grãos, sendo usados por grandes e pequenas empresas de pesquisa de sementes. O principal produto melhorado é o distribuidor de sementes.

A partir de um distribuidor muito simples, foram implantadas diversas melhorias, na qual tem como finalidade ajudar empresas de pesquisa, onde se necessita uma perfeita semeadura depositada em cada linha. A eficiência da quantidade semeada em relação aos produtos existentes chegou a 99,00% e uma uniformidade na distribuição no solo de 20 a 50% melhor, em relação ao existente no mercado, conforme dados retirados da própria empresa.

O sistema mecânico implantado para melhoria do distribuidor a vácuo faz com que este distribuidor possa trabalhar em terrenos bem inclinados. Não há necessidade da contagem das sementes nos pacotes a serem semeados, apenas deverão ter no mínimo a quantidade máxima em cada pacote para a semeadura, tendo como referência a eficiência deste sistema. As Figuras 1 e 2 dão uma melhor ideia de como é o funcionamento deste distribuidor.

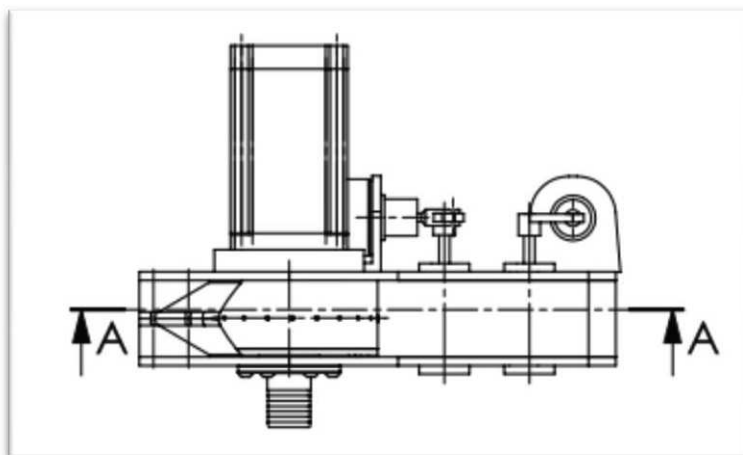


Figura 1 - Vista Superior do Distribuidor de Sementes da Empresa.

Fonte: Empresa Renz e Cia.

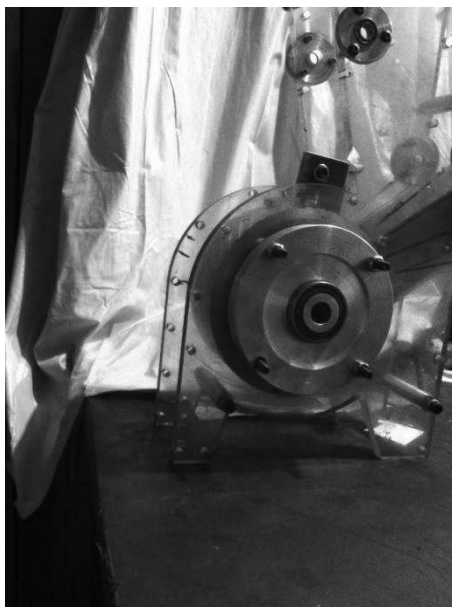


Figura 2 - Vista Lateral do Distribuidor de Sementes da Renz e Cia.

Fonte: Empresa Renz e Cia.

Também foi criado um sistema pela própria empresa de retirada das sementes duplas, com grau de eficiência de 99%, dado repassado pela empresa, assim como um novo sistema de pegar as sementes sem falhar nenhuma. Juntamente foi adicionado a este distribuidor, uma câmara de espera de sementes, e um sistema de descarte das sementes que sobram, assim

como, um sistema de limpeza de resíduos que possam estar junto com as sementes (sementes quebradas, cascas, pedrinhas).

Um motor de passo para controle de rotação é usado para aumentar o grau de confiabilidade da sementeira. Este pode ser acionado mecanicamente pela sementeira, quando utilizado em sementeira de sementes sem o controle populacional das sementes. Assim, o sistema se faz mais eficiente que os sistemas de cone e vácuo usados tradicionalmente.

O sistema de vácuo da câmara de sucção das sementes ainda pode ser aperfeiçoado quanto à sementeira de parcelas. Este sistema é composto por um disco a vácuo, que transporta a semente, pois fica apoiada em seu próprio orifício de sucção sendo assim muito mais seguro e eficiente. Dessa maneira, o sistema se difere do que existe no mercado, feito geralmente de inox, onde a semente é coletada na parte inferior do disco em uma câmara, carregando a semente e ficando suspensa até o ponto do plantio.

4.3 TIPOS DE DISTRIBUIDORES DE SEMENTES EXISTENTES

Com o passar do tempo os distribuidores de sementes foram evoluindo e com o auxílio da eletrônica, um melhor preenchimento da área a ser cultivada foi possível. Devido a melhorias nas sementeiras, diferentes mecanismos para uma correta distribuição de sementes foram feitas, buscando uma maior uniformidade na distribuição das sementes para a sementeira, maximizando a produção.

4.3.1 Distribuidores de sementes do tipo Discos Perfurados

Os distribuidores de sementes que utilizam discos perfurados são os tradicionalmente utilizados no Brasil. Podem trabalhar tanto na horizontal como em posições inclinadas, como mostra a Figura 3. Este sistema proporciona uma distribuição de semente bastante uniforme, desde que a sementeira esteja bem regulada. Porém, possui maior risco para problemas mecânicos além de exigir uma classificação rigorosa quanto a escolha das

sementes. Ainda são muito usados devido ao baixo custo, possuindo uma tecnologia simples, comparado com outros produtos equivalentes.

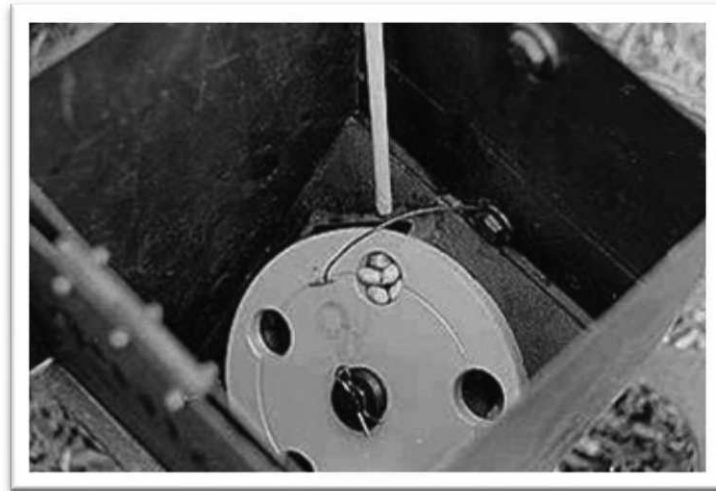


Figura 3 - Disco Perfurado Inclinado

Fonte: [PINTO, 2002]

4.3.2 Distribuidores de sementes do tipo Dedos Prensos

Este tipo de distribuidor de sementes foi desenvolvido devido a necessidade de trocar os discos das sementes cada vez que as sementes fossem trocadas e a dificuldade de se obter o disco apropriado as dimensões das sementes. Este mecanismo possui doze dedos prensos por molas que são abertos e fechados por ressaltos à medida que se gira. A semente é alimentada do depósito pra o reservatório por gravidade, (PINOTTI, 2011).

Logo após o dedo prensor se mover através do reservatório pelas sementes, ele se fecha e prende a semente entre o dedo e um prato estacionário, como mostrado na Figura 4. Devido ao giro do mecanismo do dedo prensor, a semente passa por uma abertura no prato estacionário, fazendo com que ela seja liberada para o mecanismo de posicionamento da semente.



Figura 4 - Distribuidor de Sementes por Dedos Prensorees

Fonte: [PINTO, 2002]

4.3.3 Distribuidores de sementes por um Sistema Pneumático de Pressão

Este sistema de dosagem de sementes distribui as sementes de forma homogênea, dispensando o uso de peneiras, o que possibilita o plantio de qualquer tipo de semente não havendo a necessidade de regulagem. Possui um sistema de câmara a vácuo, menor que os dosadores convencionais, bem como uma vedação de alta eficiência, o que proporciona total aproveitamento do vácuo para sucção das sementes. Isto faz com que as falhas devido às trepidações diminuam, garantindo uma melhor precisão e agilidade no plantio, (PORTELA, 1999).

É a tecnologia mais recente quando falamos em distribuidores de sementes, provendo melhores resultados quando comparadas com as demais. A Figura 5 mostra tal distribuidor.



Figura 5 - Sistema Pneumático

Fonte: [STARA, 2015]

4.4 INTERFACE USUÁRIO X MÁQUINA

Para facilidade de entendimento ao usuário do que está acontecendo, e de informações adicionais do que está ocorrendo no seu processo, é necessária muitas vezes uma forma de visualização.

Nos últimos anos, vários equipamentos foram criados para esta finalidade, como o display de cristal liquido (LCD), display *Touch Screen* e teclados que permitem esta interação entre a máquina e operador.

4.5 SENSORES DE RUPTURA DE FEIXE

Estes sensores são configurados com o emissor e receptor empacotados separadamente. O emissor emite uma luz através de um espaço e o receptor detecta a luz do outro lado. Quando o produto ou material, passar entre o emissor e o receptor, a luz para de bater no receptor, informando ao sensor que algo esta interrompendo o feixe. Este tipo de sensor é bastante

usado, pois possui uma alta confiabilidade para objetos opacos, (THOMAZINI, 2007).

4.6 MICROCONTROLADORES

O microcontrolador é um computador, onde executa um programa específico do usuário, tendo pleno controle de suas funções e ações sobre o qual foi especificado, (BRAIN, 2000).

Os microcontroladores oferecem uma variada combinação de projetos altamente eficientes e integrados, dotado de uma inteligência programável e muito utilizado no controle de processos lógicos. Possui aplicação diversa, podendo atuar com vários outros microcontroladores e tendo uma boa diversidade de recursos como conversor A/D, PWM e outras funções, assim como boa capacidade de processamento. Seu custo é considerado baixo, devido a sua flexibilidade de aplicações. Pelo grande numero de portas e por conter todos estes recursos, o PIC16F877A foi usado para o projeto.

5 MATERIAIS E MÉTODOS

O projeto é constituído por um sistema de controle e monitoramento, com elementos de automação que serão implementados em um distribuidor de sementes, o qual será constituído de componentes eletrônicos de sensoriamento e controle, com características para este uso. Este sistema terá o seu controle feito por microcontroladores embarcados através de um programa destinado a essa aplicação.

Um suporte em vidro que será acoplado juntamente com o distribuidor de sementes. Neste local, ficará armazenado todo o circuito de controle, juntamente com o sistema de automação, que será composto por alarmes e também irá controlar a velocidade do motor de passo caso ocorra alguma falha durante o processo. Todo o sistema será armazenado numa placa de circuito impresso onde ocorrerá todo o controle do processo. Como mostrado na Figura 6.

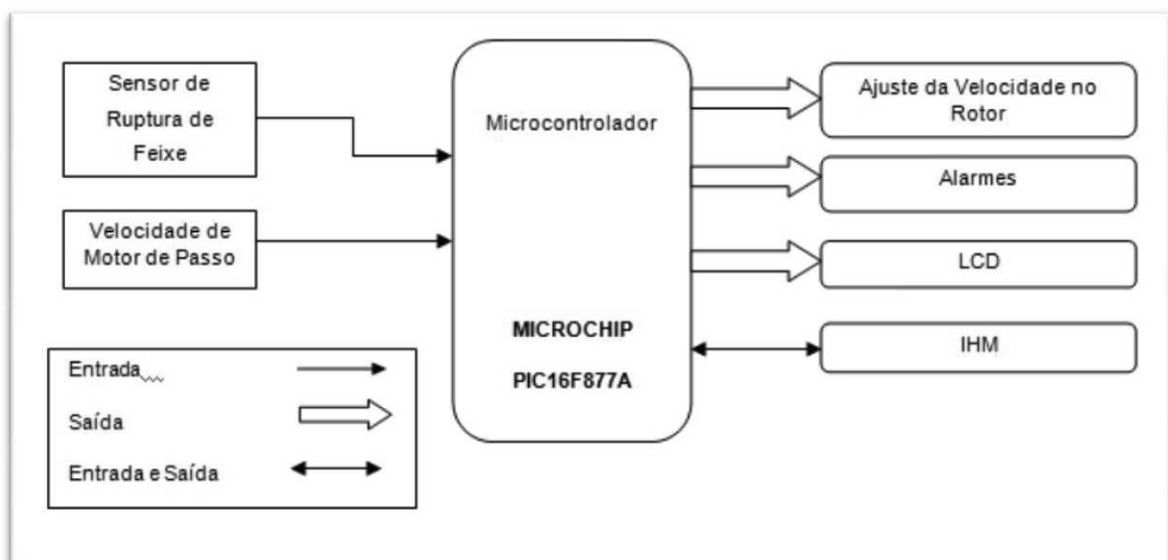


Figura 6 - Sistema para funcionamento do controlador no distribuidor de sementes

Os materiais que serão utilizados neste projeto foram escolhidos através de estudos feitos na área, por meio de fóruns e pessoas especializadas no ramo da agricultura de precisão, além de entrevistas com pessoas que

trabalham em empresas especializadas na agricultura de precisão. A principal dúvida era quanto ao tipo de sensor de ruptura de feixe e a eficiência ou não deste sensor quando acoplado ao distribuidor de sementes.

O uso de sensores, assim como o controle da velocidade do motor de passo a partir do momento que ocorra uma falha no processo, por falta de semente foi a principal variável para o aperfeiçoamento deste processo.

5.1 MICROCONTROLADOR PIC16F877A

O microcontrolador usado neste projeto será o PIC16F877A, mostrado na Figura 7. Este microcontrolador é muito usado, pois possui grande confiabilidade quanto as suas conexões periódicas. Suas características podem ser consultadas no site do fabricante.

Neste componente será feito o controle, além de toda leitura do processo. Sua escolha foi devido a ser um microcontrolador facilmente encontrado no mercado, de baixo custo, com número de portas suficientes para este sistema.

A Figura 7 ilustra todas as 33 portas que podem ser usadas como entrada e saída no sistema. As demais portas são os pontos de alimentação do microcontrolador e o oscilador externo usado para sabermos a frequência usada pelo microcontrolador.

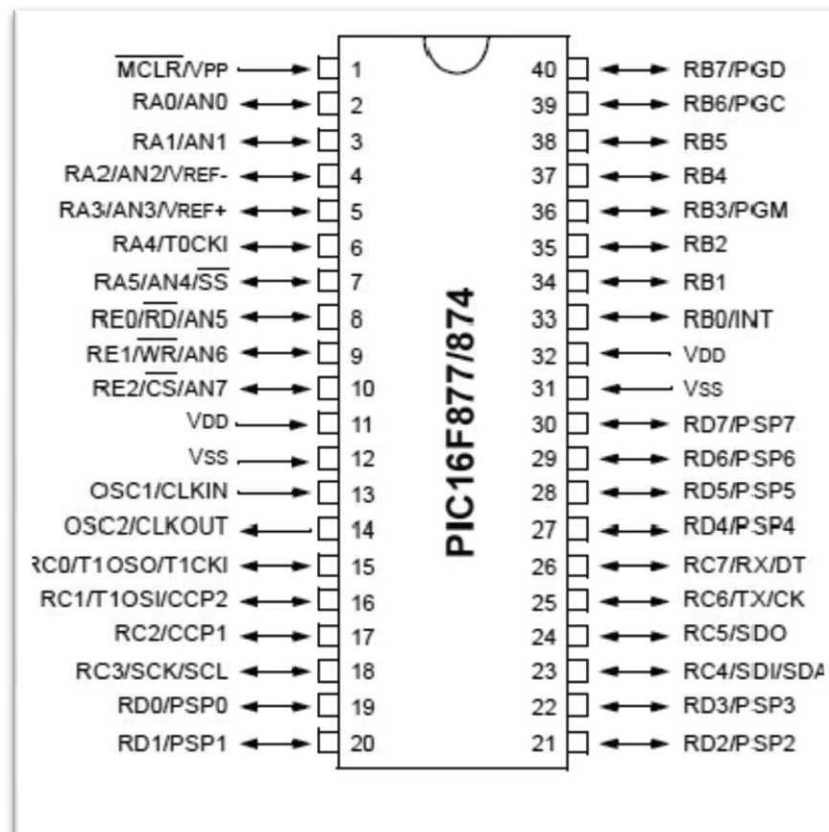


Figura 7 - Microcontrolador PIC16F877

Fonte: [MICROCHIP, 2013]

As entradas e saídas neste PIC são configuradas através do registrador TRIS. Quando colocado em nível alto (5 V), a porta é colocada como entrada e quando usado nível baixo (0 V) sua porta é configurada como saída.

Se configurado como saída, o pino pode ter um estado lógico imposto de acordo com o valor escrito no registrador PORT correspondente. Isso permite alimentar um dispositivo de baixo consumo ou fornecer um nível lógico de acordo com o necessário. No caso do pino ser configurado como entrada, o valor encontrado no registrador PORT é utilizado para monitorar o nível lógico em cada pino do microcontrolador.

Este microcontrolador possui 8 entradas analógicas, sendo elas de RA0 a RA5 e de RE0 a RE2. Estas portas são usadas para fazer a conversão AD. Ou seja, passar um valor digital de 0 a 5 V, para um valor entre 0 a 1023, visto que a resolução do conversor para este microcontrolador é de 10 bits.

As fontes de sinais analógicos mais encontrados são para medir luminosidade, temperatura, entre outras características. Neste estudo, o sinal analógico é relacionado conforme a variação de um sensor emissor-receptor infravermelho que irá variar conforme a presença ou não de grãos no distribuidor de sementes. Esta informação é passada ao microcontrolador pela porta RA0.

O oscilador externo usado será de 4 MHz e colocados na porta 13 e 14 do PIC. Estas portas não podem ser ocupadas como entrada ou saída do sistema, devendo exclusivamente ser ocupada quando necessário o uso de um cristal externo.

O valor encontrado para o cristal teve como base estudos feitos em fóruns e conversando com pessoas especializadas nestes tipos de microcontroladores. Devido a sua frequência elevada, dois capacitores de 22 pF foram colocados em paralelo com este cristal. Isto faz com que eventuais perdas com ruídos e interferências sejam reduzidas, (SACCO, 2014).

Também será usado um display LCD no qual irá mostrar ao usuário informações relevantes ao processo. O display LCD será colocado nas portas RB0 a RB5, onde 4 destas portas serão para recebimento dos dados e as outras 2 serão usadas para habilitar o envio dos dados. Informações como total de erros ocorridos durante o processo, total de grãos, entre outras, serão repassadas ao display.

Um teclado matricial 4x4 irá servir para que o processo seja inicializado e finalizado. As portas RC0 a RC3 e RD0 a RD3 serão usadas para que o teclado seja lido corretamente. Quatro destas portas serão usadas como entrada e as outras 4 como saídas.

Três sinais digitais irão enviar pulsos elétricos há um *driver* de motor de passo. Estes sinais serão enviados através das portas RC4 e RC5 passando informações do PASSO e DIREÇÃO, além do sinal OPTO que será colocado diretamente em nível ALTO. Estes sinais enviados ao *driver* irão acionar o motor de passo. Mais informações serão passadas posteriormente quando será explicado o uso do *driver* e do motor de passo.

O Apêndice A mostra um esquemático com o correto posicionamento de cada porta e de todo o processo que será montado no microcontrolador para o controle do distribuidor de sementes.

5.2 MOTOR DE PASSO

São dispositivos mecânicos eletromagnéticos que podem ser controlados digitalmente através de um hardware ou software específico. São utilizados principalmente por poder controlar sua velocidade, ângulo e direção, podendo girá-los num determinado ângulo com uma excelente precisão, (MESSIAS, 2006).

Os motores de passo comumente são encontrados de três tipos: os de relutância variável, magnético permanente e motores híbridos. O motor de passo magnético permanente tem como sua característica se “agarrar” quando se gira o seu eixo, mesmo estando desligado. Normalmente, possui dois rolamentos independentes e apenas quatro fios, sendo chamado de motor bipolar.

O motor por relutância variável geralmente possui um ou dois fios comum e quatro rolamentos, tendo o seu eixo “livre”. São chamados de motor unipolar, constituído por 5 ou 6 fios, (ROMERO, 2014).

Há ainda os Motores Híbridos, no qual contem 6 ou 8 fios e podem usar tanto controladoras unipolares, quanto bipolares. Este tipo de motor é montado geralmente com duas bobinas por fase e conforme as combinações dos fios podem atender aos dois tipos de controle.

Neste sistema será usado um motor de passo de 8 fios. Este tipo de motor é muito utilizado pois pode ser ligado em qualquer configuração, ou seja, unipolar ou bipolar. O motor de 8 fios em modo Unipolar é feito quando colocamos um centro de cada bobina, em um ponto comum. A Figura 8 ilustra este tipo de ligação.



Figura 8 - Motor Unipolar de 8 fios

Fonte: [TRUCCO, 2013]

O motor de passo quando utilizado no modo bipolar, com a configuração de 8 fios, pode possuir três características: aumento de velocidade, aumento do torque e também agregando velocidade e torque juntos. Estes três tipos de configuração são conhecidas como Meio-Passo, Bipolar Série e Bipolar Paralelo, respectivamente. As Figuras 9, 10 e 11 mostram como é o modo de ligação destas três configurações.



Figura 9 - Motor de Passo Bipolar 8 fios Ligação Meio-Passo

Fonte: [TRUCCO, 2013]



Figura 10 - Motor de Passo Bipolar 8 fios Ligação em Série

Fonte: [TRUCCO, 2013]



Figura 11 - Motor de Passo Bipolar 8 fios Ligação em Paralelo

Fonte: [TRUCCO, 2013]

Devido à utilização em baixas velocidades, em nosso projeto iremos usar o modo de ligação de motor Bipolar em Série. Como precisamos de um elevado torque e trabalhamos com uma velocidade no motor de passo baixa, seria inviável utilizar os outros modos de ligação, pois algumas características que o motor de passo possui seriam perdidas em relação aos outros modos.

Para ser dada uma maior precisão para o motor de passo, um posicionamento por controle digital irá facilitar essa obtenção. Isto irá ocorrer, sincronizando com precisão o sinal de saída de pulso do controlador para um *driver* de motor de passo.

Um sinal de pulso é um sinal elétrico no qual o nível de tensão é continuamente alterado entre LIGADO e DESLIGADO, onde cada ciclo entre LIGAR/DESLIGAR é contabilizado como um pulso. Este comando por meio do pulso, faz com que o eixo de saída do motor gire um passo. Estes níveis de sinais são geralmente chamados de “High” e “Low”, como mostra a Figura 12. A quantidade de giros do motor de passo é proporcional ao número de sinais de pulso dados ao driver.

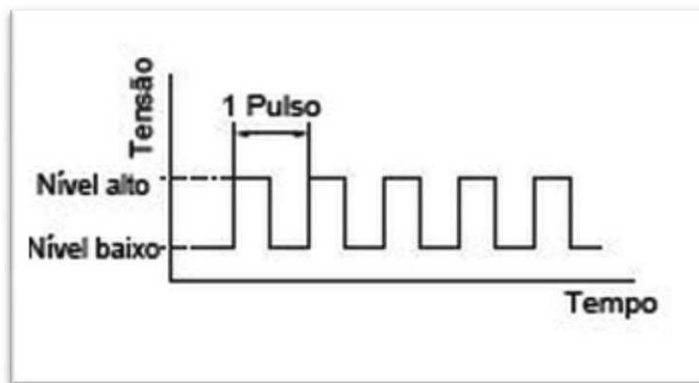


Figura 12 - Sinal de Pulso a ser gerado para melhor controle do Motor de Passo

Fonte: [ORIENTAL MOTOR DO BRASIL, 2014]

Cada motor de passo possui certo driver dedicado para seu uso. Este driver pode tanto operar no modo de entrada de pulso como, mostrado na Figura 13, como no modo de controlador acoplado, visualizado na Figura 14.

O modo de entrada de pulso para geração de pulsos, um gerador de pulsos externo é inserido ao lado do controlador para que a geração de pulsos possa ser gerenciada pelos dados oriundos do controlador programável, para só depois ser enviado estes dados ao motor de passo.

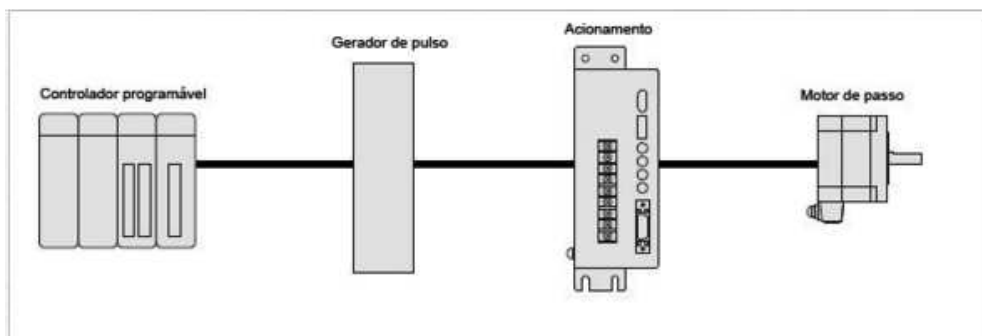


Figura 13 - Gerador de Pulso agregado junto com o *Driver* e Controlador

Fonte: [ORIENTAL MOTOR DO BRASIL, 2014]

No nosso caso iremos utilizar a função de gerador de pulso acoplado, no qual permite que o motor seja impulsionado por meio de computador ou um controlador programável, um PIC. Este tipo de gerador facilita a ligação, devido a não precisar nenhum tipo de gerador de pulso separado.

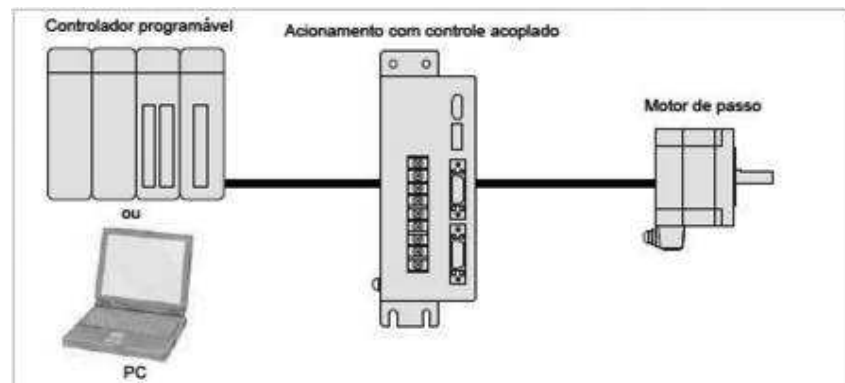


Figura 14 - Gerador de Pulso acoplado com o Controlador

Fonte: [ORIENTAL MOTOR DO BRASIL, 2014]

5.2.1 Motor de Passo AKYIAMA AK85H/6-1.8

Este motor de passo possui uma grande vantagem em relação aos demais. Seu torque de 100kgf.cm proporciona uma alta eficiência para o tipo de trabalho que será feito. Devido ao seu tamanho, será acoplado ao distribuidor de sementes externamente. O torque deve ser elevado, pois este distribuidor irá atuar nas mais difíceis condições, enfrentando lugares íngremes e também diferentes tipos de solo.



Figura 15 - Motor de Passo Akiyama AK85H/6-1.8

Fonte: [AKIYAMA, 2015]

Algumas características como grau de passo igual a 1.8° , podendo ser usado como micropasso através de *driver* específico para tal motor, e principalmente por usar o modo Bipolar em Série no qual damos prioridade maior ao torque e a baixa velocidade, fez com que escolhêssemos este motor de passo.

Este sistema criado será acoplado há um trator no qual irá trabalhar a uma velocidade de 1,5 km/h, proporcionando a colocação das sementes a 10 cm uma da outra, em relação ao solo a partir de uma determinada velocidade do motor de passo.

Algumas características deste motor podem ser observadas na Figura 16:

NEMA	Conexão		Holding torque (kgf.cm)	Corrente (A/fase)	Tensão (V/fase)	Resistência (Ω /fase)	Indutância (mH/fase)
34	Bipolar	Série	100.0	2.1	11.2	5.34	55.2
		Paralela		4.2	5.6	1.36	13.8
	Unipolar		70.0	3.0	8.0	2.67	13.8

Item	Especificação
Ângulo do passo	1.8°
Nº de fases	2
Nº de passos	200
Enrolamento	Espiras bifilares
Temperatura máx, de operação	80 °C
Temperatura ambiente	$-10^\circ\text{C} \sim 50^\circ\text{C}$
Classe de isolamento	B
Quantidade de fios	8
Peso	5 Kg
Sugestão de driver compatível	AKDMPD8/220-8.2A

Figura 16 - Características do Motor de Passo AK85H/6-1.8

Fonte: [AKIYAMA, 2015]

5.2.2 Driver Motor de Passo AKYIAMA AKDMP5-5.0A

Este é um *driver* Bipolar chooper (Driver de Corrente) no qual usa o sistema PWM (modulação por largura de pulso) para o controle, produzindo maior torque e estabilidade ao motor. Pode acionar motores até 5 A, fazendo com que todas as configurações do motor de passo AK85H/6-1.8 possam ser usadas. Tem uma proteção contra ruídos pela sua entrada digital OPTO.

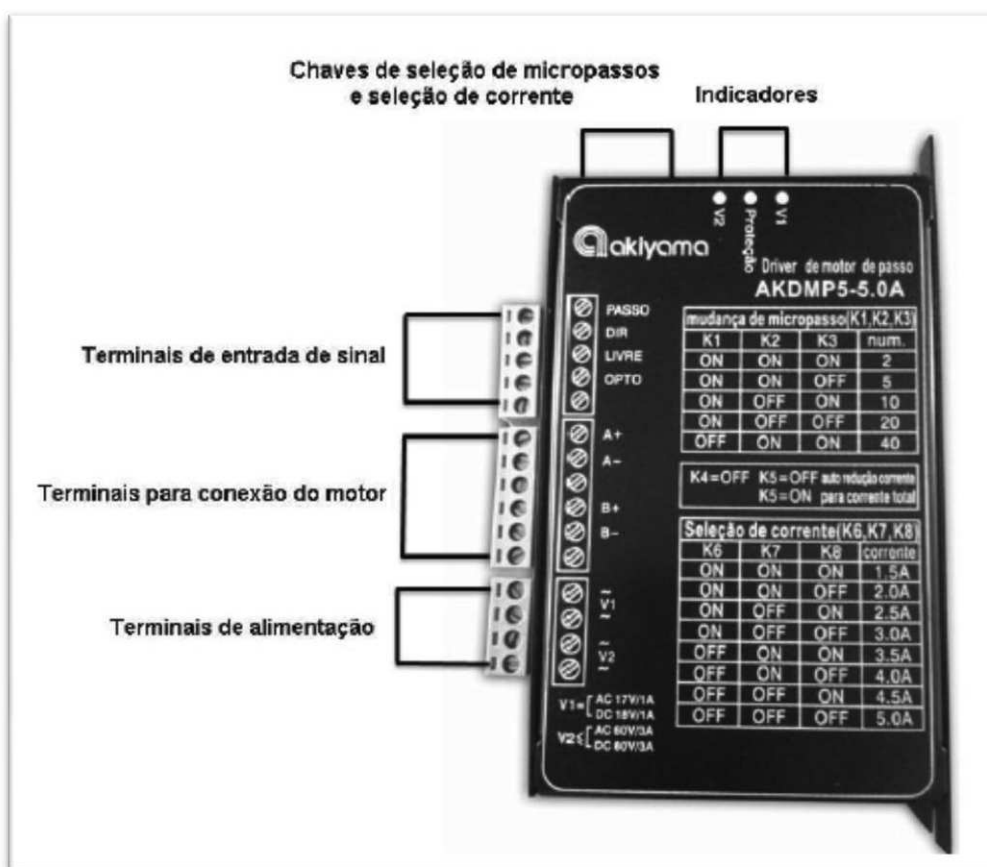


Figura 17 - Driver de Motor de Passo AKDMP5-5.0A

Fonte: [AKIYAMA, 2015]

Além disso, seu sinal de controle é dado pela forma NPN, controlando-os através dos sinais de PASSO e DIREÇÃO. O sinal de PASSO é usado para controlar a posição e a velocidade do motor de passo. Um trem de pulsos irá acionar o motor de passo através de uma velocidade proporcional a este trem

de pulsos. Um controle total deste motor pode ser feito quando isto for utilizado.

Este sinal deve estar entre 8-15 mA. O sinal de DIREÇÃO deve acionar o motor em sentido horário, quando colocado em nível alto (5 V), e no sentido anti-horário quando em nível baixo (0 V). Os pinos RC4 e RC5 serão usados respectivamente como DIREÇÃO e PASSO.

A entrada digital OPTO deve receber uma fonte de alimentação de 5 V e é usada como uma entrada de alimentação auxiliar dos circuitos de comando, além de servir como proteção para ruídos outras interferências.

Os terminais de conexão para o motor devem estar colocados no modo Bipolar Série, como explicado anteriormente, proporcionando uma melhor eficiência quanto ao torque.

Os terminais de alimentação do *driver* podem ser acionados de duas maneiras. Por fonte de tensão CC ou fonte de tensão CA. No *datasheet* do *driver* é encontrado maiores informações sobre como devem ser estas fontes. Neste trabalho será usado um transformador com dupla saída de tensão. Uma fonte com 12 V, 1 A e a outra com 60 V, 5 A, ambas com tensão alternada. Este transformador foi feito para tal uso, devido ao valor de 60 V, 5 A, não ser facilmente encontrado no mercado.

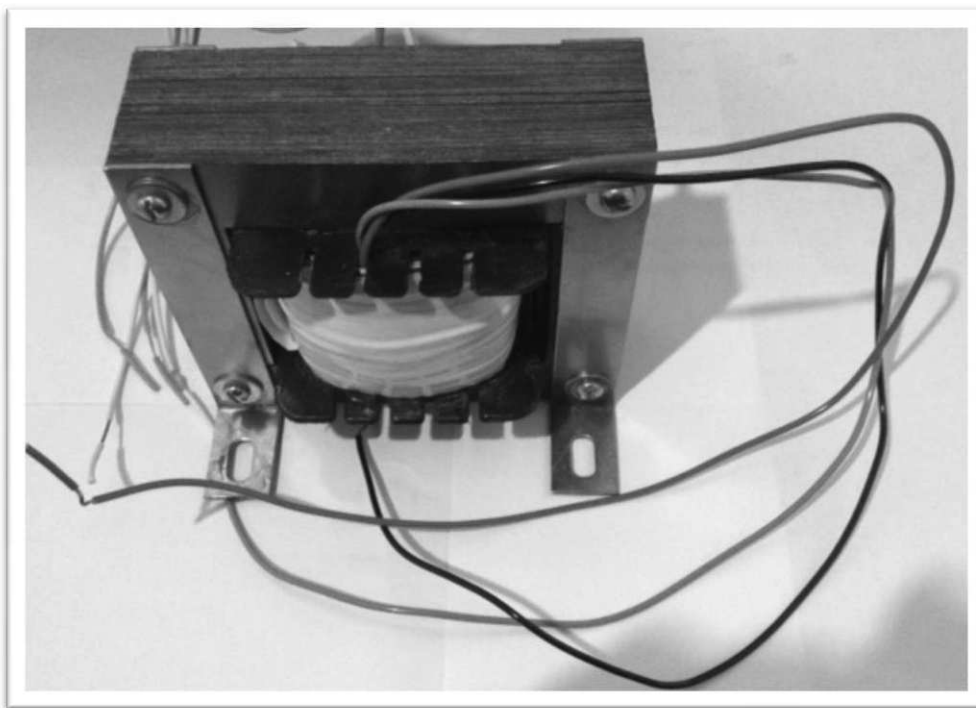


Figura 18 - Transformador 12 V 1 A, 60 V 5 A.

Para determinar a chave de seleção do nível de corrente é necessário observarmos o tipo de ligação feita no motor de passo. Esta seleção deve ter um valor de corrente acima do escrito no *datasheet* do motor de passo. Selecionamos inicialmente um valor de corrente de 4,5 A selecionando K5, K6 e K7 como OFF e K8 como ON.

Para determinar a seleção de micropassos, as chaves K1, K2, K3 e K4 no *driver* do motor de passo tem a função de determinar quantos passos o usuário quer para o uso no controle do motor de passo. Posicionando K1, K2, e K3 como ON e K4 como OFF um número de divisão de passos por 2 é posicionando, proporcionando o controle para uma volta do motor de passo em 400 posições.

Demais informações podem ser encontradas no *datasheet* do *driver*.

5.3 SENSOR EMISSOR-RECEPTOR INFRAVERMELHO

Para utilização no projeto, este tipo de sensor é composto por dois componentes infravermelhos. Um LED irá emitir uma luz infravermelha (IR) e um fototransistor a detecta. Quando ocorrer algum tipo de interrupção entre o feixe do LED e do fototransistor, uma variação na tensão é sentida em relação à tensão que o sensor tinha anteriormente.

5.3.1 LED Infravermelho - TIL32

O TIL32 é um LED que emite uma luz infravermelha, com comprimento de onda de 940 nm. Este parâmetro é extremamente importante, visto que o detector precisa trabalhar nesta mesma frequência. Possui uma corrente máxima de 40 mA. A Figura 19 mostra este LED.



Figura 19 - LED Infravermelho TIL32

Fonte: <http://www.datasheetarchive.com/>

5.3.2 Fototransistor Infravermelho - TIL78

O TIL78 é um fototransistor que possui dois terminais, correspondendo ao coletor e emissor de um transistor, como mostra a Figura 20. A base é ativada pela luz, e quando uma quantidade suficiente de luz infravermelha é captada o transistor conduz, permitindo a passagem de corrente do coletor para o emissor. Com a falta de claridade, o transistor não conduz e o coletor e emissor ficam isolados. Assim, a saída do fototransistor fica em 5 V.

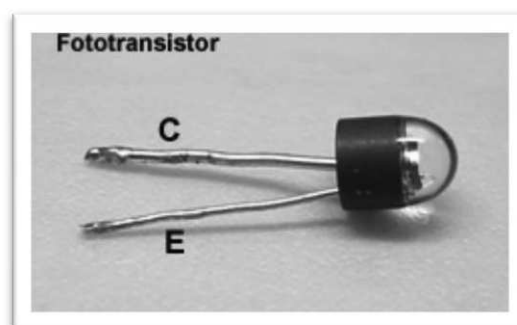


Figura 20 - Fototransistor Infravermelho TIL78

Fonte: <http://proesi.com.br/>

A leitura destes dados será feito pelo conversor AD, pela variação no nível de tensão do fototransistor. Usando um valor de referência, valores acima

serão entendidos como resposta negativa para detecção de grãos e valores abaixo serão usados para respostas positivas.

Utilizando o PIC 16F877A, um sinal de entrada analógico será convertido para um binário de 10 bits, no qual é armazenada a saída. Este resultado de 10 bits é gerado por aproximações sucessivas ao valor de entrada para um valor analógico que melhor o corresponde. Ele será armazenado nos registradores ADRESH e ADRESL. O conversor AD pode gerar uma interrupção, uma vez concluída a conversão.

Os pinos de conversão AD devem ser colocados como entrada, antes de iniciar a conversão no registrador TRIS, assim como no registrador ANSEL/ANSELH.

O registrador ADCON0 também deve ser configurado. Nele selecionamos o canal do conversor, seleção do *clock* da conversão, status da conversão e habilitação do módulo ADC. O *datasheet* do microcontrolador pode ser consultado através do *site* do fornecedor.

Os bits CHS3: CHS0 são usadas para a configuração do canal na qual o conversor irá se conectar. Só será habilitado o canal RA0, tendo como resposta os bits 0000. A seleção do *clock* da conversão acontece pelos bits ADCS1:ADCS0. Como é usado um cristal de 4 MHz, precisamos usar a seleção de bits 10, para que assim um tempo de conversão AD seja de 1,6 μ s que é o mínimo que o conversor pode atuar. A habilitação dos bits GO/DONE e ADON são feitas durante a conversão conforme mostra o Apêndice C.

O registrador ADCON1 deve ser usado para sabermos se o formato do resultado da conversão será justificado a esquerda ou à direita assim como para sabermos a quantidade de portas A/D usadas. Confira o *datasheet* para configuração.

Para o projeto, foi ocupado o formato de justificado a direita e o bit ADFM foi colocado em nível alto, assim como os bits PCFG3: PCFG0 no qual foi colocado a resposta de 1110, habilitando apenas a porta RA0 como entrada analógica. Para um melhor entendimento de como essa resposta é justificada para direita ou esquerda a Figura 21 é usada para esclarecimento.

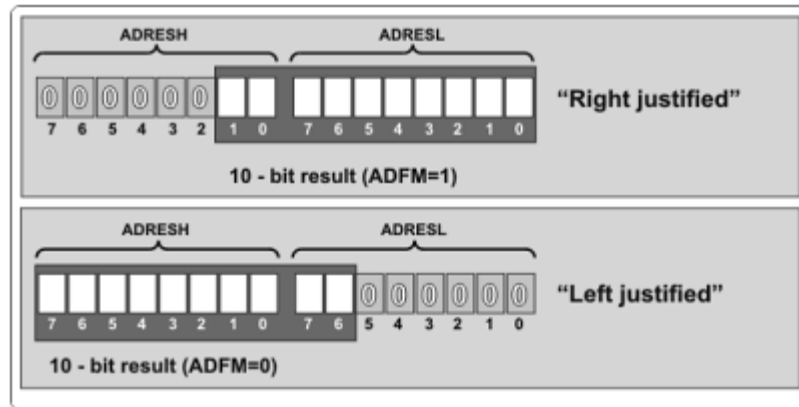


Figura 21 - Resposta justificada a esquerda ou a direita

Fonte: <http://microcontrolandos.blogspot.com.br/>

Os registradores ANSEL/ANSELH são colocados como 00000001 e 00000000 respectivamente, para apenas a seleção da porta AN0 ocorra.

5.4 DISPLAY LCD

O *display* LCD que foi usado neste projeto, é o modelo 16x2, conforme mostrado na Figura 22. Devido sua facilidade de configuração quando usado em conjunto aos microcontroladores PIC sua utilização se torna muito simples. Por possuir 8 pinos de dados, 3 pinos de controle, 3 de alimentação e 2 onde pode se definir o *backlight*, podemos acessar todo o *display* usando apenas 11 pinos.



Figura 22 - Exemplo de Display LCD 16x2

Fonte: <http://www.marcelomaciel.com/>

Usando este *display*, as funções de cada porta são descritas na Tabela 1.

Tabela 1 - Descrição das Funções dos Pinos do Display LCD

Pino	Nome	Função
1	Vss	Terra
2	Vdd	Positivo (5 V)
3	Vo	Contraste do LCD. Chamado de Vee
4	RS	Register Select
5	R/W	Read/Write
6	E	Enable
7	D0	Bit 0 do dado a ser escrito no LCD(ou lido dele).
8	D1	Bit 1 do dado a ser escrito no LCD(ou lido dele).
9	D2	Bit 2 do dado a ser escrito no LCD(ou lido dele).
10	D3	Bit 3 do dado a ser escrito no LCD(ou lido dele).
11	D4	Bit 4 do dado a ser escrito no LCD(ou lido dele).
12	D5	Bit 5 do dado a ser escrito no LCD(ou lido dele).
13	D6	Bit 6 do dado a ser escrito no LCD(ou lido dele).
14	D7	Bit 7 do dado a ser escrito no LCD(ou lido dele).
15	A	Anodo do <i>backlight</i>
16	K	Catodo do <i>backlight</i>

Os pinos de 1 a 3 são usados para fazer o contraste do *backlight* do LCD. Geralmente, é usado um potenciômetro para que este contraste seja feito. Em nosso projeto um potenciômetro de 10 k Ω foi usado para fazer este contraste.

Os pinos 4,5 e 6 são usados como entradas do *display*. No projeto o pino 5 R/W não foi ativado, pois como será usado apenas a função de escrita na tela não precisamos conectá-lo ao microcontrolador. Assim, o pino 5 será aterrado. Os pinos 4 e 6, são chamados de pinos de comandos, no qual o pino RS é quem define se os bits enviados para o LCD serão configurações ou dados. Quando nível baixo, o LCD interpreta os bits como configurações do *display*, limpar tela, cursor a direita por exemplo. Já em nível alto o controlador interpreta os bits como dados e escreve no *display*. O pino ENABLE deve ser enviado quando informamos o controlador que os pinos de dados precisam ser

lidos, pois existe uma configuração ou comando para ser executado. Os pinos RB4 e RB5 recebem respectivamente ENABLE e RS.

Os pinos de dados foram de D4 a D7, ou RB0 a RB3, fazendo com que dois *nibbles*, ou seja, para que um byte seja lido é preciso primeiramente ler os quatro primeiros bits e depois os quatro últimos, fazendo assim com que um comando ou uma configuração seja enviado completamente.

5.5 TECLADO MATRICIAL 4x4

Este teclado foi utilizado, pois é necessário se interagir com o controlador, para que o processo tenha um ponto de partida e um ponto de desligamento. Este teclado, conforme mostra a Figura 23 é composto por 4 linhas e 4 colunas.

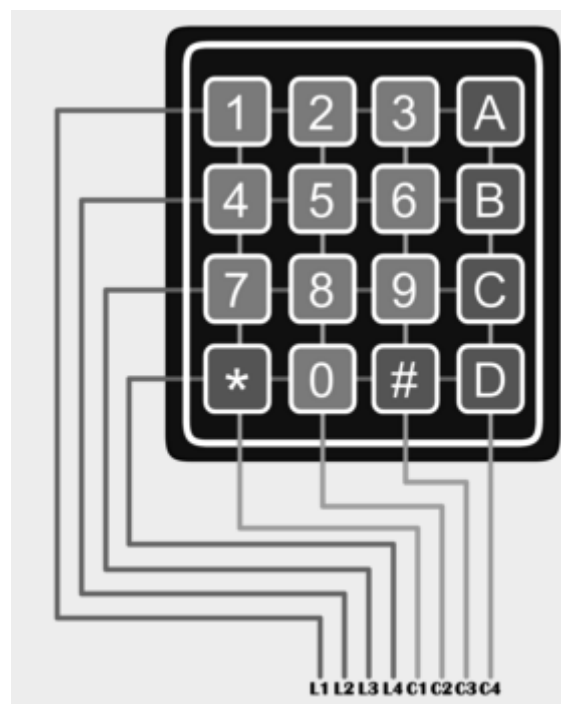


Figura 23 - Teclado matricial 4x4

Fonte: <http://blog.filipeflop.com/>

No teclado matricial se usa a técnica de multiplexação, na qual compartilham de um mesmo barramento para vários dispositivos, todavia apenas um irá ser utilizado por vez. Neste caso os barramentos serão as linhas

do teclado e os dispositivos as colunas. Para que ocorra a leitura correta, apenas uma coluna deve ser lida por vez.

Para que isto ocorra, devemos colocar as colunas como entradas, ou seja, RD0 a RD3 são entradas do sistema, assim como, RC0 a RC3 são as saídas. Nas colunas, colocamos resistores em *pull-down* mantendo em nível baixo quando não houver acionamento nas teclas.

6 METODOLOGIA

Segundo Saraiva (1998), todo o sistema da agricultura de precisão é composto por quatro grandes etapas: a coleta de dados, o gerenciamento da informação, a aplicação em taxa variável, e a avaliação dos resultados. A Figura 24 ilustra estas etapas de uma maneira mais simples.

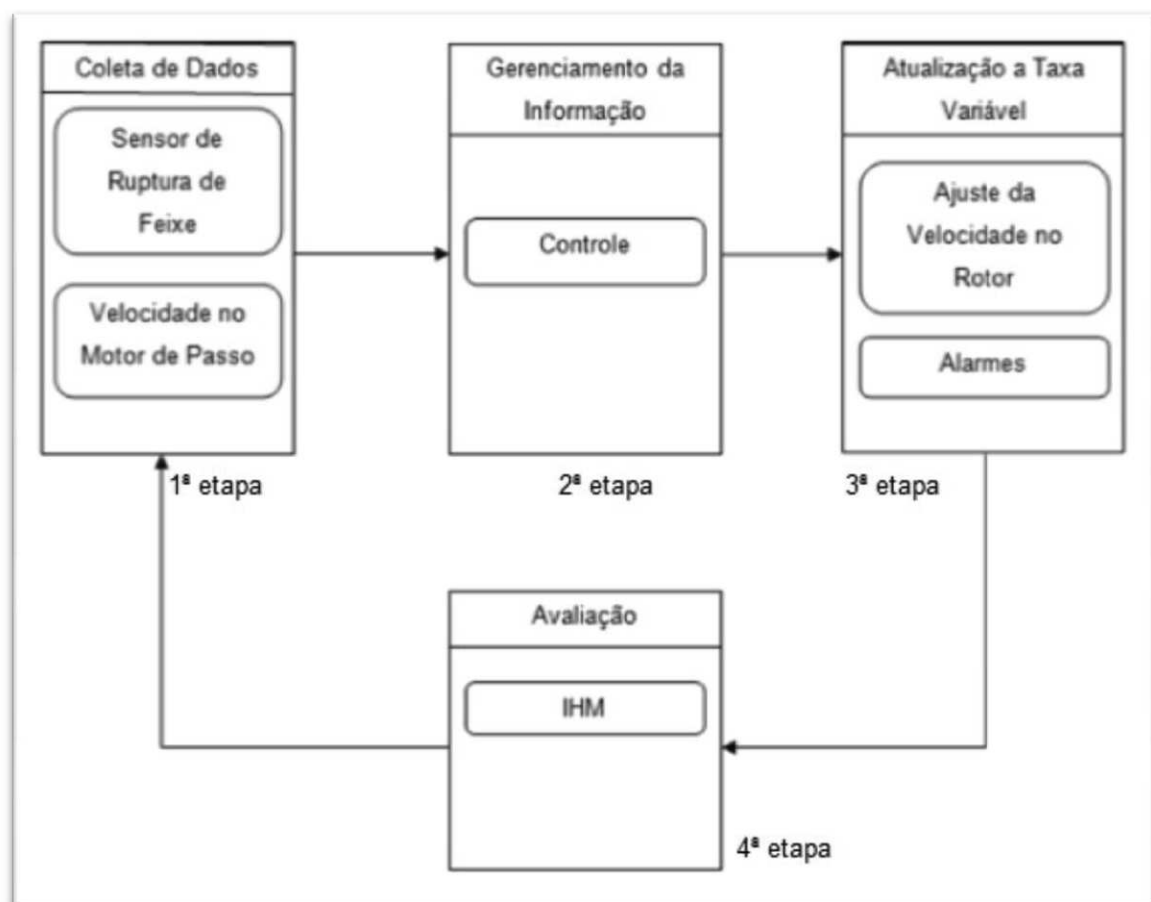


Figura 24 - Lógica do Sistema aplicado a AP

A primeira etapa identifica a variabilidade existente no campo, por isso todas as informações que possam indicar essa variância devem ser retiradas tendo em comparação as informações anteriormente passadas da própria produção.

No gerenciamento da informação, leva-se em consideração o processamento dos dados com apoio de um aparato computacional. Varias informações relevantes podem ser retiradas, visando corrigir os erros apresentados na coleta de dados, ou reparar pequenas mudanças ocorridas. Assim, as possíveis causas do que está acontecendo serão apontadas.

A atualização a taxa variável, tem como finalidade efetivar o processo que está sendo controlado, fazendo com que o ajuste ocorra de forma automática. Com isto, uma melhora de desempenho é esperada quando comparado ao que havia acontecendo.

A última etapa é a avaliação dos resultados e a visualização das informações importantes ocorridas. Na verdade, este processo está distribuído durante todo o ciclo, mas permanece sempre realimentado, devido a novos cenários que possam se apresentar.

Com base nestas etapas foi escolhido o software de programação, assim como a lógica de programação para o controle do motor de passo e a montagem do sistema protótipo.

6.1 SOFTWARE PARA PROGRAMAÇÃO

Para que o controle do distribuidor de sementes ocorresse foi usado o *software* MPLAB X IDE à versão v3.00, tendo como linguagem de programação, a linguagem C. Devido a ser uma linguagem de fácil compreensão, esta se torna mais acessível do que outras linguagens, como Assembly, por exemplo. Nesta ultima linguagem citada, a programação é feita *bit a bit*, sendo conhecida por ser uma linguagem de baixo nível, por causa deste método usado, proporciona uma dificuldade quanto ao uso dos seus registradores e a execução da lógica de programação.

Este software destinado exclusivamente a microcontroladores da *Microchip Technology* e juntamente com o compilador XC8, no qual permite a programação em linguagem C, foram as bases para que a programação fosse feita.

A dificuldade em encontrar tutoriais, fóruns ou pessoas especializadas no funcionamento desta linguagem tornou-se um ponto importante no projeto.

Informações quanto ao uso de interrupções, uso do conversor AD e o funcionamento da lógica de programação tornaram o projeto muito mais complicado do que o imaginado, devido à falta de informações precisas quanto aos seus usos, referentes a linguagem usada.

6.2 LÓGICA DE PROGRAMAÇÃO

Com a ideia de fazer um controle total de um distribuidor de sementes e a compensação do erro se a sucção da semente no disco do distribuidor não ocorrer, a lógica criada para o processo se baseou em dois problemas descritos a seguir:

Tendo em vista que o distribuidor de sementes é composto por um disco com 22 furos, onde as sementes são succionados para que eles possam ter um correto posicionamento para a semeadura, um motor de passo faz com que este distribuidor gire em uma velocidade proporcional.

Esta velocidade proporcional se dá pelo numero de pulsos que o motor de passo recebe por segundo. Com base no *datasheet* do *driver* do motor de passo está velocidade proporcional dada em RPM, foi encontrada pela equação (1):

$$RPM = \frac{PPS \times 60}{PPR} \quad (1)$$

Onde:

PPR = 200x(nº de divisão de passo)

PPS – Passo por Segundo

PPR – Passo por Revolução

Outro referencial importante e que só foi possível obter valores depois de testes realizados na Empresa Renz e Cia., foi o tempo de sucção de cada semente, ou seja, o tempo para que uma semente colocada no furo **A** passasse pelo furo **B**, como mostra a Figura 25, quando girando no sentido horário. Assim deu-se a velocidade com que o sensor IR respondeu ao sistema, isto é, o tempo da passagem de grãos no disco do distribuidor de sementes.

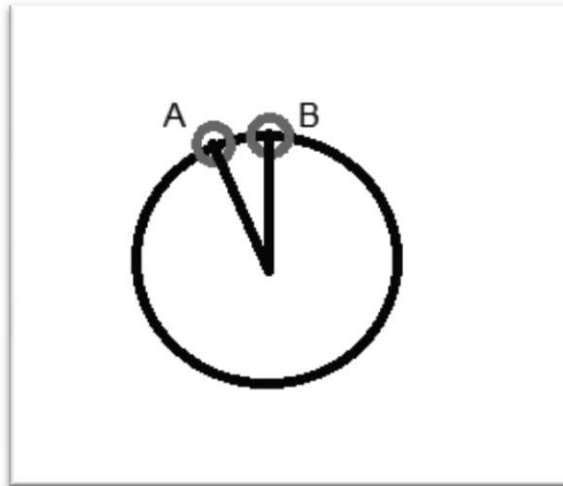


Figura 25 - Tempo de sucção dos grãos, para que A chegue em B.

Com base no tempo de deslocamento da semente entre os furos **A** e **B**, uma velocidade proporcional adequada foi colocada para que os grãos das sementes tivessem um espaçamento em relação ao solo com uma margem de 10 cm entre um e outro.

Um fluxograma foi criado para facilitar o entendimento do processo, como mostra a Figura 26.

O processo inicia a partir do teclado matricial 4x4, apertando o botão A. A partir deste momento as sementes serão colocadas de forma contínua no disco por meio de sucção, conforme o número de grãos passados para semeadura pelo usuário. Com base no PASSO e DIREÇÃO recebidos pelo microcontrolador, o motor de passo irá girar em uma determinada velocidade.

Tendo os parâmetros de PASSO e DIREÇÃO aceitos, um sensor emissor-receptor infravermelho é posicionado para detectar a falta ou não das sementes no disco do distribuidor. A partir de um valor limiar determinado, se a diferença de potencial sentida no receptor for maior que este valor limiar, a passagem de sementes acontece de forma correta e a velocidade com que o motor de passo está atuando pode-se manter constante.

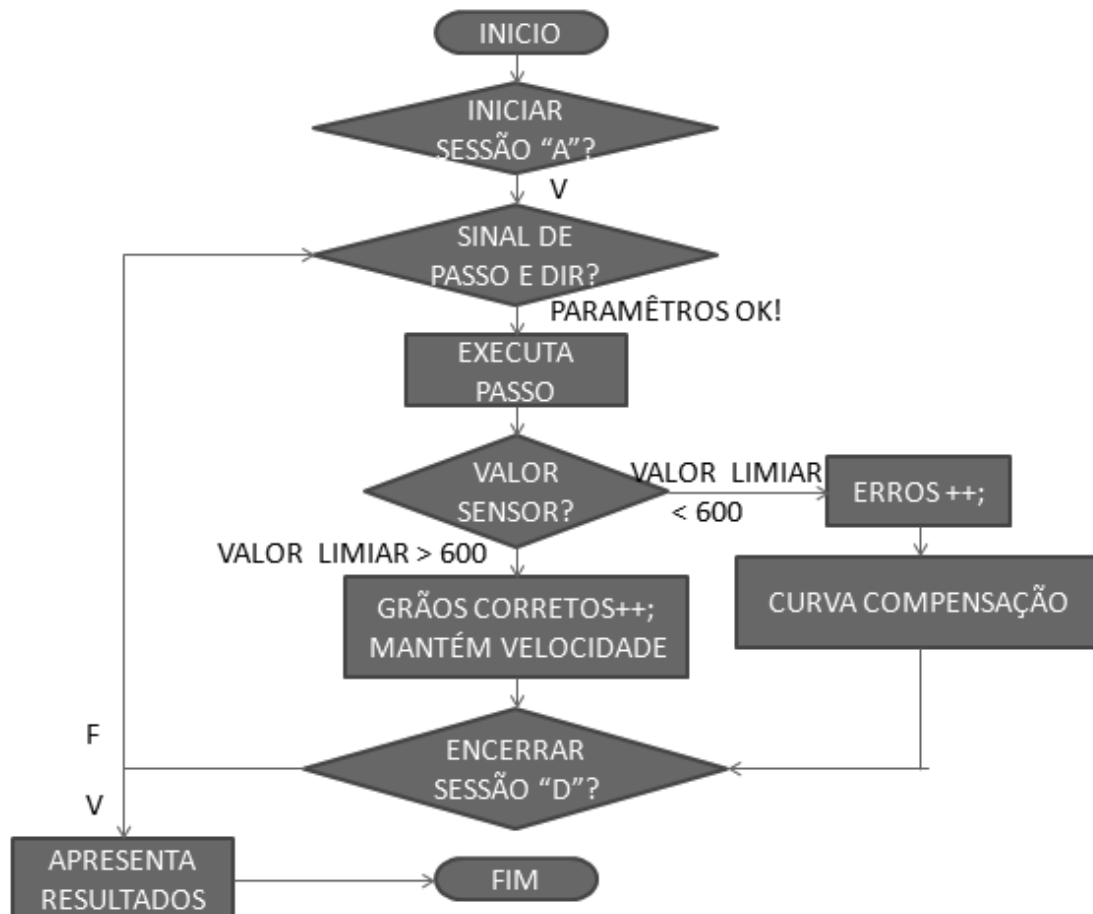


Figura 26 - Fluxograma do Sistema Controlador do Distribuidor de Sistemas

Se um erro ocorrer, ou seja, um valor abaixo do valor limiar chegar ao sensor receptor infravermelho, uma curva de compensação é criada para que o erro possa ser compensado, fazendo com que esta falha possa ser corrigida. Esta curva de compensação fará com que o motor de passo aumente sua velocidade e depois diminua, até voltar à velocidade constante anteriormente determinada. A Figura 27 ilustra como será esta curva de compensação.

Apertando o botão D do teclado 4x4 a sessão é encerrada e os resultados são apresentados ao usuário. Para se iniciar uma nova sessão é necessário apenas que o usuário aperte A e novos resultados poderam ser determinados após se encerrar a sessão novamente.

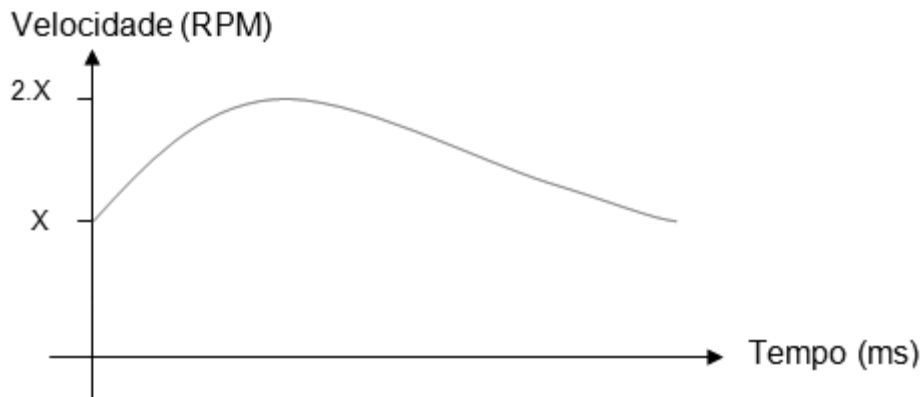


Figura 27 - Curva de Compensação criada para o sistema

Após a lógica do sistema de controle do motor de passo, passamos ao processo de geração do código para esta implantação.

Duas partes importantes do código são apresentadas individualmente para um melhor entendimento.

Primeiramente, para a geração da velocidade passada para o motor de passo, assim como para a conversão AD, se faz necessário o uso de Interrupções no MPLAB X IDE.

Interrupções são eventos externos que causam ao processador parar a execução do programa corrente e desviam a execução para um bloco de código chamado de interrupção, que contém normalmente operações de entrada e saída de dados. Terminando o tratamento da interrupção, o controle retorna ao programa interrompido exatamente no mesmo estado em que estava quando ocorreu a interrupção.

Conforme mostra o Apêndice B e explicado anteriormente no capítulo 5, sinais digitais são mandados com informações ao *driver* do motor de passo. Para que a velocidade do motor de passo seja enviada corretamente ao driver, o sinal de PASSO deve ser enviado da seguinte maneira:

- Uma interrupção de TIMER1 é criada, sabendo que ela pode tanto ser usada como temporizador, como para contador, criamos uma lógica dentro do evento da interrupção para que a interrupção só seja realmente ativa e provoque um sinal para alguma porta, após um período determinado de tempo. Para isso, se usa um contador, no qual fará com que o evento de interrupção ocorra certo número de vezes, para só depois habilitar o uso em alguma porta de entrada ou saída de dados.

É permitido que este sinal digital seja enviado para sua porta, a partir de uma condição inicial, fazendo com que o motor de passo se encontre desligado se o botão A não for apertado para inicialização.

Os sinais digitais que devem ser enviados são os de PASSO, DIREÇÃO e OPTO, como explicado na seção 5.2.2.

O sinal digital de DIREÇÃO, como o motor só irá girar em um sentido, não sendo necessário seu uso. Assim, podemos colocar este sinal diretamente em nível baixo, mandando para a porta RC4 um sinal 5 us adiantado do sinal de PASSO. Este sinal adiantado é enviado para um melhor rendimento do motor de passo.

O sinal OPTO foi colocado em nível alto, servindo como fonte de alimentação de circuitos de comando. Já o sinal de PASSO é enviado pela porta RC5. Quando enviado um sinal digital em nível alto e outro em nível baixo, um pulso será gerado. A velocidade do motor de passo será proporcional à quantidade de pulsos por segundo, como mostrado a equação na seção 6.2.

Também teremos interrupção de dois outros tipos: de conversão AD, e, de UART. Estas duas interrupções ajudam muito na visualização do resultado no simulador Proteus ISIS, utilizado para simular o sistema criado. Quando usado no sistema original, apenas a interrupção do conversor AD será usada, visto que uma interrupção pode ser gerada quando a conversão se encerrar.

Os códigos para as funções de conversão AD e UART são apresentadas no Apêndice C.

A segunda parte esclarece como foi feito a lógica de controle do processo. Isto só foi possível escrevendo códigos para que o *display* LCD e o teclado matricial 4x4, apresentassem e fossem acionados, respectivamente, somente quando necessários. A leitura correta do teclado 4x4, e a configuração do display, proporciona uma melhor visualização sobre o que está acontecendo no projeto, além de fazer com que o processo seja inicializado e finalizado pelo próprio usuário.

A função do teclado 4x4 como já explicado na seção 5.5 é apenas de inicializar e finalizar o processo quando o usuário quiser. Já no caso do *display* LCD, é de mandar configurações na forma de dados ou instruções de forma

correta. O Apêndice D apresenta o código para o controle do *display* LCD, e o Apêndice E mostra o código da leitura correta do teclado 4x4.

O código principal para o controle do distribuidor de sementes foi criado com base no fluxograma mostrado na Figura 26. O Apêndice F mostra o código criado para esta implementação e como citado anteriormente um controle total do processo se dá por meio desta função.

6.3 MONTAGEM DO PROTÓTIPO DO SISTEMA

A montagem do protótipo se iniciou em uma placa de ensaio, *protoboard*, onde os primeiros testes foram feitos.

Nestes testes iniciais, foi usado um oscilador externo de 20 MHz, juntamente com dois capacitores de 22 pF para diminuir perdas por interferências e ruídos. Mesmo com o uso de capacitores e devido à alta frequência do *clock* no microcontrolador, o sinal que chegava até o microcontrolador se perdia, ou seja, não proporcionava o funcionamento do microcontrolador quando alimentado.

Por meio de conversas com pessoas da área e com visitas a fóruns de eletrônica, foi constatado que se diminuíssemos a frequência passada ao microcontrolador, o circuito não receberia tantos estímulos em um certo determinado tempo, o que facilitaria ao controlador, a leitura correta do cristal externo. Mesmo com um cristal externo de 4 MHz o microcontrolador não conseguia fazer a leitura correta da frequência recebida.

A partir desse ponto, testes foram feitos para medir a continuidade de todo o sistema para termos a certeza do que estava provocando erros na leitura do microcontrolador. Feito os testes de continuidade em todo o sistema, foi constatado que o problema era a placa de ensaio. A Figura 28 mostra a *protoboard* usada.

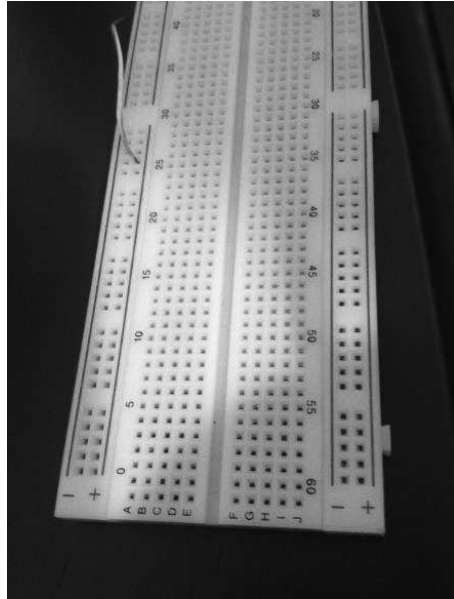


Figura 28 - Placa de ensaio usada inicialmente no controlador do sistema

Perdas e interferências na própria *protoboard* faziam com que o controlador não conseguisse ler a frequência correta passada pelo cristal externo. Assim, passou-se a usar uma placa de circuito impresso universal pré-furada do tipo Ilha, como mostra o modelo da Figura 29.

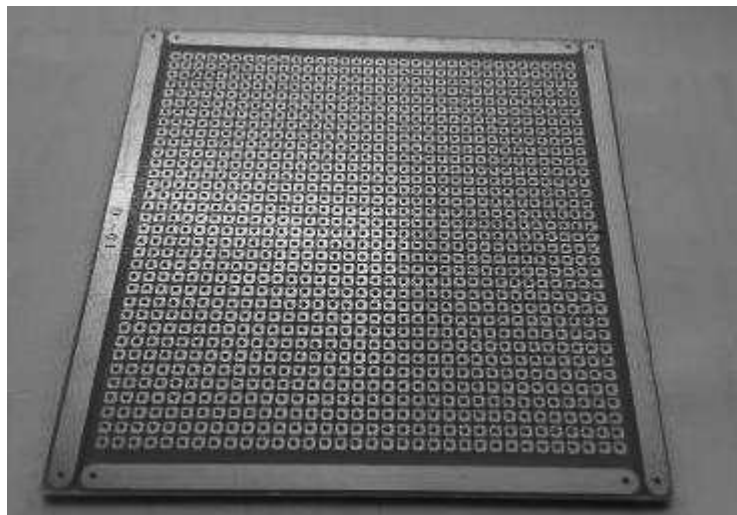


Figura 29 - Modelo da Placa Ilha utilizada para o protótipo do sistema

Com o uso da Placa Ilha, foi possível adquirir uma leitura correta da frequência de *clock* no controlador, e assim, montado o protótipo.

Tendo como base o circuito apresentado no Apêndice A e explicado durante o capítulo 5, assim como a lógica de programação usada na seção 6.2, a Figura 30 mostra o protótipo.

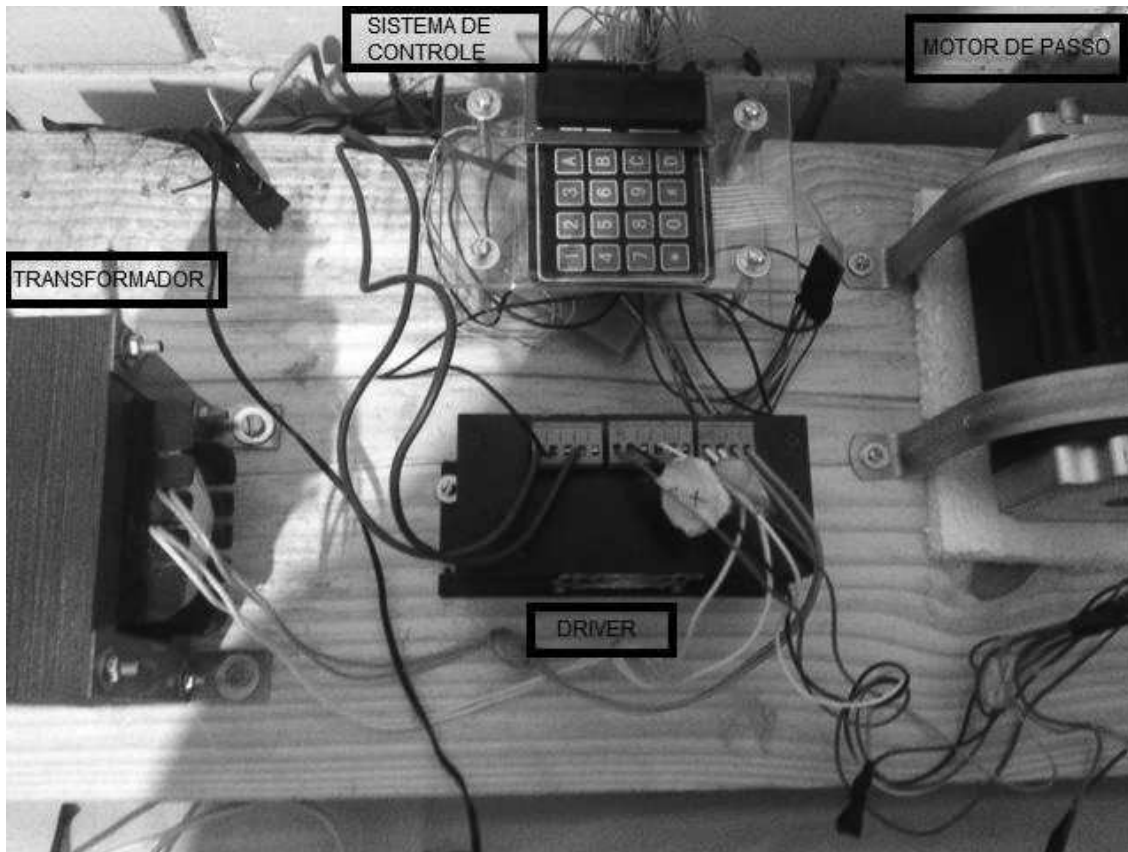


Figura 30 - Protótipo usado para o Controlador do Distribuidor de Sementes

O sensor emissor-receptor infravermelho foi colocado de forma a se assemelhar com o posicionamento no qual eles foram inseridos no distribuidor de sementes da empresa Renz e Cia.. A Figura 31 mostra o posicionamento em uma posição similar ao instalado na empresa.

Neste posicionamento, conforme testes, tomou-se como referência a inexistência de interferências no sensor emissor-receptor, uma contagem de semente corretas, ou seja, as sementes passam como desejadas pelo distribuidor de sementes. A partir do momento, em que um objeto opaco passe entre o emissor e receptor, uma contagem de erros acontece.

Com esta montagem foi possível fazer varias sessões de testes em cima deste posicionamento.

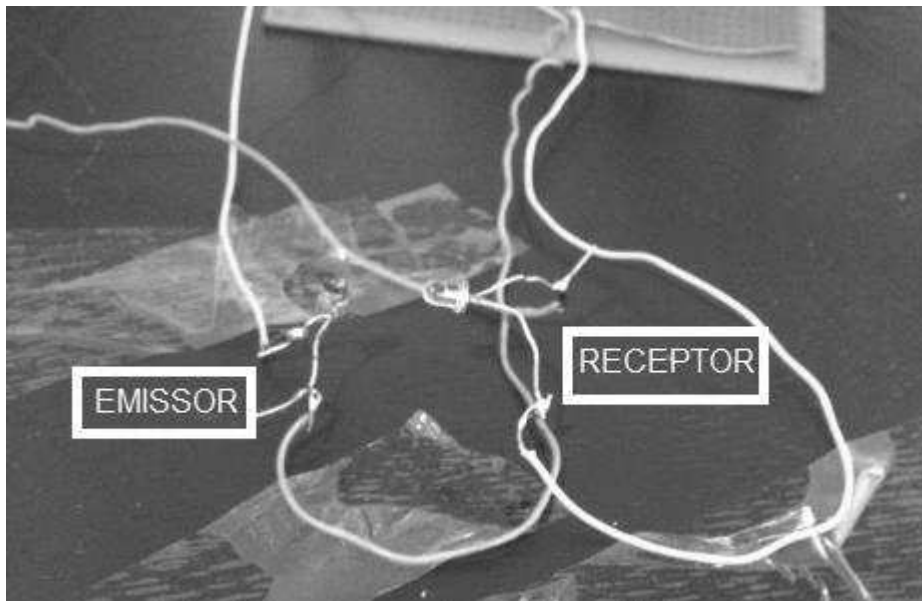


Figura 31 - Sensor emissor-receptor IR instalado

O motor de passo foi ligado a partir do momento em que o *driver* do motor de passo recebeu os sinais digitais e que a alimentação do *driver* foi feita a partir de um transformador exclusivo para o uso, como explicado na seção 5.2.2. A Figura 32 mostra esta ligação.

Como informado no próprio *datasheet* deste *driver*, os sinais digitais emitidos pelo microcontrolador devem estar a mais de 10 cm do *driver*, pois assim interferências e possíveis ruídos possam ser diminuídos a partir desta distância.

A Figura 33 apresenta as ligações feitas para o correto uso do teclado matricial 4x4 e do *display* LCD, conforme explicado nas seções 5.4 e 5.5.

Com esta montagem e com a execução da lógica do programa para o controle do distribuidor de sementes, foi possível iniciar a coleta de resultados para o sistema.

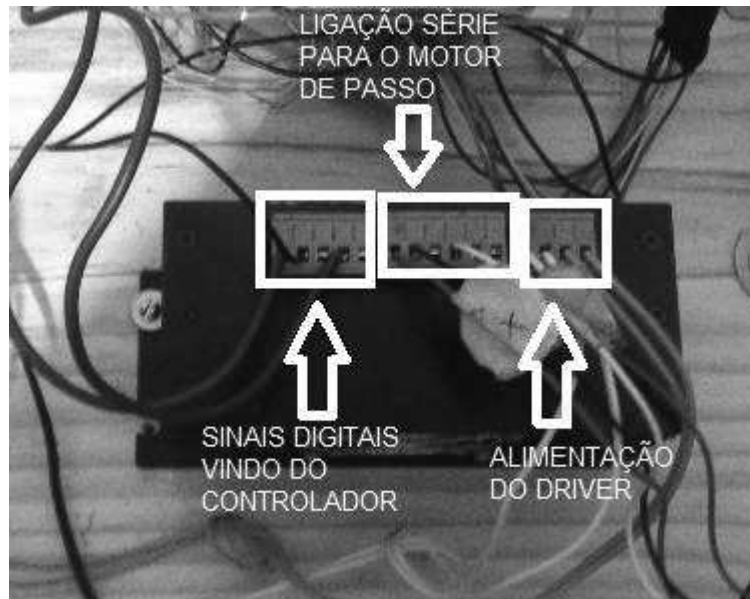


Figura 32 - Ligação do *Driver* e do Motor de Passo



Figura 33 - Ligações do Teclado 4x4 e do *Display* LCD

7 RESULTADOS

Com a implantação do código de programação para o microcontrolador baseada na linguagem C, e com a correta ligação de todos os componentes na Placa do Tipo Ilha os testes realizados buscaram:

- Verificar se o controle de todo o sistema do distribuidor de sementes estava acontecendo;

- Verificar se, a partir do momento em que um erro ocorresse pela falta de sementes no disco do distribuidor, o sistema compensador criado aceleraria a velocidade do motor de passo, com finalidade de diminuir ou anular tal erro.

Assim, testes foram feitos tanto em laboratórios, quanto na empresa Renz e Cia. para avaliação da eficiência do sistema.

7.1 CONTROLE TOTAL DO DISTRIBUIDOR DE SEMENTES

De uma forma geral, o controle total do distribuidor de sementes obteve bons resultados.

A partir da lógica de programação feita para o microcontrolador e da implantação do sistema no distribuidor de sementes da empresa, os testes realizados na Renz, mesmo com interferências no sensor emissor-receptor infravermelho devido a vibrações e interferências externas, proporcionou bons resultados quanto à coleta de dados a respeito do número de falhas detectadas.

Nos primeiros testes feitos na empresa o sensor IR não estava conseguindo suportar tanta vibração, devido ao alto torque do motor de passo e a má colocação do sistema emissor-receptor. Com os primeiros testes feitos, foi possível adequar o sistema para que os resultados pudessem ser qualificados. Assim, os seguintes resultados foram obtidos:

Tabela 2 - Resultados na empresa Renz e Cia.

Sessões de Testes	Sementes Corretas	Falhas do Sistema e Falhas Provocadas	Total de Sementes e Erros
1	264	4	268
2	247	2	249
3	387	3	390
4	97	1	98
5	149	2	151
6	323	1	324
7	197	3	200
8	328	2	330
9	193	1	194
10	346	2	348

Como é possível analisar na Tabela 2, o distribuidor de sementes da empresa possui um elevado nível de acerto, quanto à sucção correta de sementes no disco do distribuidor já que os erros presentes foram provocados, ou seja, as sementes foram subtraídas propositalmente para poder testar o sistema de controle.

O controlador a partir deste momento fez a contagem correta tanto na contagem de sementes corretas, quanto na contagem dos erros. O numero de sementes colocadas no distribuidor deu-se através da quantidade retirada, possuindo valores aleatórios. Quando um erro acontecia, um LED na Placa Ilha era aceso, informando ao usuário que um erro tinha ocorrido.

Os testes realizados em laboratórios mostraram ainda mais a eficiência do controlador aplicado a este distribuidor de sementes. Conforme explicado na seção 6.3, foi colocado o sensor emissor-receptor infravermelho de forma que quando um objeto opaco ou qualquer objeto interferisse entre o emissor e o receptor um contador de erro era acionado e quando o sensor não sofria interferência um contador de grãos fazia a contagem da quantidade de grãos corretamente posicionados. Neste caso, usamos um tempo maior entre as passagens dos grãos, assim uma maior facilidade para visualização dos erros

e acertos foi possível. A Tabela 3 mostra os resultados obtidos em testes feitos em laboratório.

Tabela 3 - Resultados em Laboratórios da universidade

Sessões de Testes	Sementes Corretas	Falhas Provocadas	Total de Sementes e Erros
1	97	8	105
2	56	5	61
3	89	7	96
4	77	3	80
5	176	6	182
6	45	5	50
7	67	7	74
8	123	6	129
9	143	4	147
10	152	8	160
11	88	3	91
12	91	7	98

Estes testes nos mostraram a eficiência quanto ao controle do processo. A taxa elevada de erros provocados foi feita para que fosse possível comprovar a eficiência do processo quanto à captura de falhas. Os erros foram cometidos de forma intencional para validar o sistema. Informações como quantidade de sementes corretamente posicionadas, erros e total de sementes foram mostradas ao usuário para autenticar o processo.

Com isso pode-se concluir, que o controle do distribuidor de sementes por meio do microcontrolador mostrou-se viável, já que é um controlador de baixo custo quando se comparado com outros controladores utilizados no mercado.

7.2 COMPENSAÇÃO DO ERRO NO DISTRIBUIDOR DE SEMENTES

O sistema compensador de erros quando utilizado na empresa Renz e Cia. apresentou resultados que puderam validar este sistema compensador criado.

Este sistema criado, quando testado no simulador Proteus ISIS obteve resultados muito satisfatórios em sua simulação. Nesta simulação, foi usado um potenciômetro em vez do sensor emissor-receptor infravermelho, como forma para mostrar ao sistema um erro. O Apêndice A mostra como foi feita esta simulação. A partir do momento em que o potenciômetro ultrapassasse um valor limiar previamente colocado, o sistema compensava este erro corretamente através da lógica criada.

Quando está lógica de programação foi passada ao microcontrolador para testes laboratoriais, o sistema conseguiu compensar eventuais erros que acontecia.

Os resultados obtidos em testes em laboratórios da própria universidade, nas amostras mostradas na Tabela 3, tanto no tempo de passagem de grãos desejado pela empresa, ou seja, o tempo em que o grão passa no disco para a sucção, para que ele seja colocado no solo 10 cm um em relação ao outro, quanto em tempos maiores se mostraram satisfatórios. Estes tempos maiores para a sucção do grão, significa que uma diminuição na velocidade do motor de passo proporciona uma maior distância dos grãos em relação ao solo. A leitura dos dados pelo microcontrolador é mais precisa quando tempos maiores entre um grão e outro ocorre.

Já nos testes feitos na empresa Renz e Cia., quando colocado o valor desejado de tempo para a passagem de grãos no disco do distribuidor de sementes, alguns erros aconteciam. Devido ao tempo para a sucção da semente no disco ser muito baixo, ou seja, uma semente colocada no disco a cada poucos milissegundos, fez com que os sensores encontrassem algumas dificuldades quanto a sua leitura.

Em alguns processos da Tabela 2, o erro era informado, mas a compensação não ocorria devido a esta velocidade elevada na leitura do sensor, no qual fazia com que ele se perdesse em uma ou duas posições, não

conseguindo fazer uma correta coleta de dados para o microcontrolador, proporcionando falhas na compensação do erro.

Este problema na coleta de dados, não ocorria todas às vezes. Em determinados erros que ocorreram no processo, o sistema conseguiu compensar esta falha, reduzindo ou anulando o erro acontecido.

Quando se aumentou o tempo da passagem do grão no disco do distribuidor, melhores resultados foram obtidos. A leitura correta do sensor IR permitiu ao sistema que todos os erros acontecidos fossem compensados, proporcionando uma ótima eficiência quando submetido a estas características.

7.3 PROJETOS FUTUROS

Com a ideia de melhorar o funcionamento do controle do distribuidor de sementes e tendo em vista os ótimos resultados obtidos, algumas partes do projeto ainda podem receber melhorias.

O uso do microcontrolador PIC16F877A no controle do sistema tende a minimizar o custo do controlador e soluções antes caras com controladores já prontos no mercado, podem se tornar muito mais viáveis com o aperfeiçoamento deste trabalho, fazendo um algoritmo específico para tal uso.

Melhorias tanto na comunicação entre o homem e o processo, quanto no sistema de controle na compensação do erro, assim como em outras partes podem ser feitas.

Algumas implementações como fazer do teclado matricial 4x4 uma interface entre homem e processo mais completa, podendo extrair mais informações, tais como, posicionamento dos erros durante o processo servindo como arquivo para o usuário, pode ser uma implementação que irá ajudar muito na melhoria do sistema. Mesmo com a leitura correta do teclado, para a função de inicialização e finalização de cada sessão do distribuidor, uma melhoria seria trazer mais informações para que o usuário possa interagir mais com o processo.

Um aperfeiçoamento no sistema compensador de erros, traria também ao sistema uma maior confiabilidade, podendo fazer o sistema trabalhar em diversas velocidades pelo motor de passo.

8 CONSIDERAÇÕES FINAIS

Os resultados obtidos no capítulo 7 demonstraram que a implementação do código de controle para o distribuidor de sementes e a execução do protótipo funcionaram conforme esperado. Em geral, o controle do distribuidor de sementes teve êxito na maioria dos seus processos, mostrando-se preciso quanto a quantidade de sementes corretamente colocados e o número de erros devido a falta de semente.

A exceção ocorreu quando o controlador foi colocado no distribuidor de sementes da empresa e implementado numa alta taxa de grãos colocados no disco em pouco tempo. Neste caso, erros ocorreram devido a um conflito de informações no sensor IR. Este problema está associado ao uso de interrupções na lógica de controle usada no microcontrolador. As interrupções fazem com que o código principal seja parado, e só depois do tratamento da interrupção, o controle retorne ao programa exatamente no ponto onde parou o código corrente. Devido a este fator e a alta velocidade na sucção de grãos no disco em um tempo pequeno, o sensor não conseguiu em alguns casos passar a informação correta.

Com os excelentes resultados apresentados na simulação através do Proteus ISIS, onde todo o sistema de controle se comportou conforme esperado e com os bons resultados que aconteceram nos testes junto ao distribuidor, os eventuais erros que ocorreram na compensação de sementes podem ser eliminados através do aperfeiçoamento da lógica de programação ou com um reposicionamento dos sensores junto ao distribuidor de sementes. Desta maneira, os resultados poderiam se assemelhar ainda mais ao simulado.

O uso de outro *software* de programação ou outra linguagem, assim como um aperfeiçoamento no esquemático criado e na lógica de controle do processo, pode ser uma das soluções para que a falha que ocorre em alguns casos na compensação do erro possa ser corrigida.

9 REFERÊNCIAS BIBLIOGRÁFICAS

AKIYAMA, **MANUAL DRIVE PARA MOTOR DE PASSO MODELO AKDMP5-5.0A**. 2015, <http://www.multcomercial.com.br/pdf/driver/5.0A.pdf>. Acesso em: 2015-17-09.

BLOG FILIPEFLOP, **COMO USAR O TECLADO MATRICIAL 4x4 COM ARDUINO**. 2014, <http://blog.filipeflop.com/arduino/teclado-matricial-4x4-arduino.html>. Acesso em: 2015-19-09.

BRAIN, Marshall. **O QUE É UM MICROCONTROLADOR**. 2000, <http://tecnologia.hsw.uol.com.br/microcontroladores1.htm>. Acesso em: 2015-13-05.

EMBRAPA, **REDE DE AGRICULTURA DE PRECISÃO II**. 2015, <http://www.macroprograma1.cnptia.embrapa.br/redeap2>. Acesso em: 2015-05-11.

JESUS, Pedro Henrique de. **Fundamentos e Aplicações de microcontroladores pic® utilizando mplab® x ide – xc8 compiler**. 1. ed. São Paulo, 2015.

MACIEL, Marcelo. **CONECTANDO UM DISPLAY DO LCD NO PIC**. 2012, <http://www.marcelomaciel.com/2012/03/configuracao-display-lcd-no-pic.html>. Acesso em: 2015-09-18.

MESSIAS, Antônio. **CONTROLE DE MOTOR DE PASSO ATRAVÉS DA PORTA PARALELA.** 2006, <http://www.rogercom.com/pparalela/IntroMotorPasso.htm>. Acesso em: 2015-18-08.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. **AGRICULTURA DE PRECISÃO:** Folder. Brasília, 2013.

NEOYAMA, **CONVERSOR A/D.** 2013, <http://microcontrolandos.blogspot.com.br/2013/03/pic-conversor-ad.html>. Acesso em: 2015-17-09.

ORIENTAL MOTOR DO BRASIL, **VISÃO GERAL DO MOTOR.** 2014, <http://www.orientalmotor.com.br/produtos/motores-de-passo.html>. Acesso em: 2015-11-09.

PIC16F877A, **PIC16F87XA.** 2013, <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>. Acesso em: 2015-13-05.

PINOTTI, Elvio Brasil. **AGRICULTURA DE PRECISÃO.** 2011, http://www.fatecpompeia.edu.br/arquivos/arquivos/aula_n4_maquinas.pdf. Acesso em: 2015-12-05.

PINTO, Francisco. **SEMEADORAS.** 2002, Slides Universidade Federal de Viçosa - UFV. Arquivo em: 2015-05-18.

PORTELA, José Antônio. **PLANTIO DE PRECISÃO: O DESAFIO DO SÉCULO XXI**. 1999, http://www.cnpt.embrapa.br/biblio/p_co25.htm. Acesso em: 2015-12-05.

ROMERO, André. **DESENVOLVIMENTO DE UM CONTROLE PARA MOTOR DE PASSO**. 2014, <http://stoa.usp.br/andreromero/files/-1/11859/Motor-Passo.pdf>. Acesso em: 2015-10-09.

SACCO, Francesco. **CONFIGURATION BITS DO PIC16F877A**. 2014, <http://www.embarcados.com.br/configuration-bits-do-pic16f877a/>. Acesso em: 2015-09-12.

SARAIVA, A. M.; CUGNASCA, C. E.; HIRAKAWA, A. R et al. **Aplicação em taxa variável de fertilizantes e sementes**. Viçosa: UFV, 2000.

SILVA, Cláudio da. **ELETRÔNICA EMBARCADA E OS COMPUTADORES DE CAMPO**. 2012, <http://blogdaagriculturadeprecisao.blogspot.com.br/2012/04/eletronica-embarcada-e-os-computadores.html>. Acesso em: 2015-11-05.

STARA, **AGRICULTURA DE PRECISÃO**. 2015, <http://www.stara.com.br/produtos/agricultura-de-precisao-pt-br/>. Acesso em 2015-05-13.

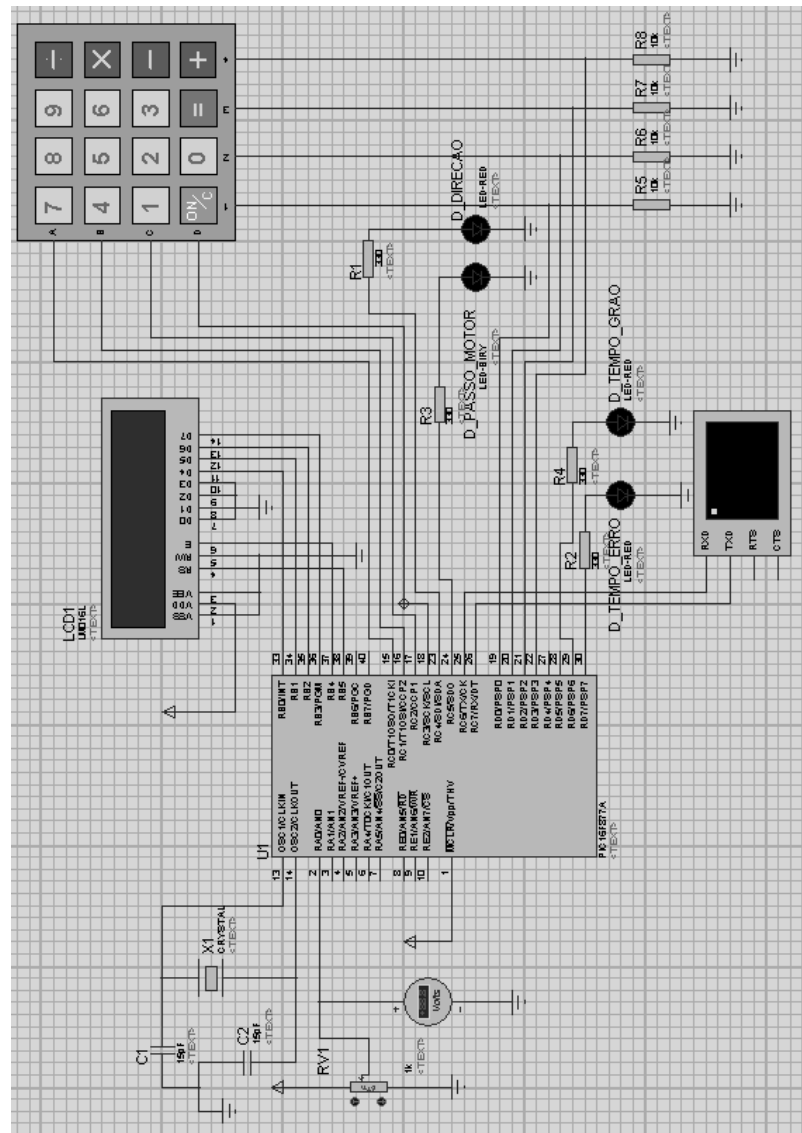
THOMAZINI, Daniel. **Sensores Industriais**, 2007. ALBUQUERQUE, Pedro U **Sensores Industriais Fundamentos e Aplicações**. São Paulo: Érica, 2007.

TIL32, **LIGHT EMITTING DIODES (LEDs)**. 2013,
<http://www.datasheetarchive.com/dlmain/Datasheets-X2/DSA1206000-174.pdf>.
Acesso em: 2015-13-05.

TIL78, **FOTOTRANSISTORES DE SILICIO**. 2012, <http://proesi.com.br/phft380-til78-foto-transistor-receptor-3mm.html>. Acesso em: 2015-13-05.

TRUCCO, Victor. **OS MOTORES DE PASSO**. 2013,
<http://www.victortrucco.com/CNC/MotoresdePasso/MotoresdePasso.asp>.
Acesso em: 2015-09-09.

APÊNDICE A – Simulação Proteus ISIS do Sistema de Controle



APÊNDICE B – Código das Funções de Interrupção para o controle do motor de passo

```

void interrupt interrupcoes(void)//vetor de interrupção
{
    if(PIR1bits.TMR1IF){

        if(flag_libera_inicio == 1){
            if(flag_curva_interrupt == 0) //se erro aconteceu?
                TMR1 = 40536; // 0,05s
            else{//(flag_curva_interrupt = 1)
                if(flag_sobe_desce == 0) //curva subida
                    TMR1 = 59286;//0,025s
                else // curva descida
                    TMR1 = 53036; //0,0125s
            }

            if(cont > 4){ //0,05s x 5 = 0,25s
                PORTCbits.RC5 = !PORTCbits.RC5;
                cont = 0;
            }
            cont++;
        }
        PIR1bits.TMR1IF = 0;
    }

    if(RCIF)
    {
        //se interrupção de recepção da serial
        //aqui não será usado essa função
        caracter = RCREG;
        flag_interrupcao = 1;
        RCIF = 0;// limpa flag de interrupção de recepção
    }
    if(ADIF)
    {
        //se interrupção do modulo analogico!
        ADIF = 0;//limpa flag
    }
}

```

APÊNDICE C – Código das Funções para conversão AD e UART

```

//////////funções usadas pela uart //////////
void inicializa_RS232(long velocidade,int modo)
{
//por padrão é usado o modo 8 bits e sem paridade, mas se necessario
//ajuste aqui a configuração desejada.
//verifique datasheet para ver a porcentagem de erro e se a velocidade
//é possivel para o cristal utilizado.
    RCSTA = 0X90;//habilita porta serial,recepção de 8 bit em modo
    //continuo,assincrono.
    int valor;
    if(modo == 1)
    {
//modo = 1 ,modo alta velocidade
        TXSTA = 0X24;//modo assincrono,transmissao 8 bits.
        valor =(int) (((_XTAL_FREQ/velocidade)-16)/16);//calculo do valor do
        //gerador de baud rate
    }
    else
    {
//modo = 0 ,modo baixa velocidade
        TXSTA = 0X20;//modo assincrono,transmissao 8 bits.
        valor =(int) (((_XTAL_FREQ/velocidade)-64)/64);//calculo do valor do
        //gerador de baud rate
    }
    SPBRG = valor;
    RCIE = 1;//habilita interrupção de recepção
    TXIE = 0;//deixa interrupção de transmissão desligado(pois corre se o
    //risco de ter uma interrupção escrita e leitura ao mesmo tempo)
}

void escreve(char valor)
{
    TXIF = 0;//limpa flag que sinaliza envio completo.
    TXREG = valor;
    while(TXIF ==0);//espera enviar caracter
}

void imprime(const char frase[])
{
    char indice = 0;
    char tamanho = strlen(frase);
    while(indice < tamanho ) ///veja que o programa pode travar se aqui não
    //tiver as duas aspas
    {
        escreve(frase[indice]);
        indice++;
    }
}

```

```

void int_to_char(int quant)
{
    char convert_char5='0';
    char convert_char6='0';
    char convert_char7='0';
    char convert_char8='0';
    char convert_char9='0';

    while(quant>=10000)
    {
        quant=quant-10000;
        convert_char5++;
    }

    while(quant>=1000)
    {
        quant=quant-1000;
        convert_char6++;
    }
    while(quant>=100)
    {
        quant=quant-100;
        convert_char7++;
    }

    while(quant>=10)
    {
        quant=quant-10;
        convert_char8++;
    }
    while(quant>=1)
    {
        quant=quant-1;
        convert_char9++;
    }

    escreve('\n');
    escreve('\r');
    escreve(convert_char5);
    escreve(convert_char6);
    escreve(convert_char7);
    escreve(convert_char8);
    escreve(convert_char9);
}

```

```
////////////////////////////////////funções usadas pelo conversor AD////  
void inicializa_AD()  
{  
    ADCON0 = 0X10;//Tosc = 32 (20MHZ , veja datasheet) ;habilita modulo de  
    ADCON1 = 0X8E;//leitura do valor justificado a direita, apenas RA0 será  
    ADIE = 1;//não habilita interrupção de conversão analogica  
}  
int leitura()  
{  
    long valor;  
    ADCON0 = 5;//solicita leitura  
    while(ADCON0 == 5);//espera terminar leitura  
    return((ADRESH<<8)+ ADRESL);  
}
```

APÊNDICE D – Código de controle do Display LCD

```

#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include "hardware.h"
#include "pic16.h"

//pinos display
#define TR_DADO_4  TRISBbits.TRISB0
#define TR_DADO_5  TRISBbits.TRISB1
#define TR_DADO_6  TRISBbits.TRISB2
#define TR_DADO_7  TRISBbits.TRISB3
#define TR_ENABLE  TRISBbits.TRISB4
#define TR_RS      TRISBbits.TRISB5
#define LCD_DADO_4  PORTBbits.RB0
#define LCD_DADO_5  PORTBbits.RB1
#define LCD_DADO_6  PORTBbits.RB2
#define LCD_DADO_7  PORTBbits.RB3
#define LCD_ENABLE  PORTBbits.RB4
#define LCD_RS      PORTBbits.RB5

#define lcd_type 2          // 0=5x7, 1=5x10, 2=2 lines
#define lcd_line_two 0x40 // LCD RAM address for the 2nd line

const unsigned char LCD_INIT_STRING[4]=
{
20 | (lcd_type << 2),    // Func set: 4-bit, 2 lines, 5x8 dots
0xc,                    // liga display
1,                      // limpa display
6                       // incrementa cursor
};

void init_lcd_4bit(void);
void lcd_envia_nibble(unsigned char n);
void lcd_envia_byte(unsigned char address, unsigned char n);
void lcd_gotoxy(unsigned char x, unsigned char y);
void lcd_putc(unsigned char c);
void lcd_escreve_p(const unsigned char *n, ...);
void lcd_escreve_s(const unsigned char *y);

```

```

//inicializa LCD
void init_lcd_4bit(void)
{
    unsigned char i;

    TR_ENABLE=0;
    TR_RS=0;
    LCD_ENABLE=0;
    LCD_RS=0;
    TR_DADO_4=0;
    TR_DADO_5=0;
    TR_DADO_6=0;
    TR_DADO_7=0;
    LCD_DADO_4=0;
    LCD_DADO_5=0;
    LCD_DADO_6=0;
    LCD_DADO_7=0;

    __delay_ms(15);

    for(i=0; i<3; i++)
    {
        lcd_envia_nibble(0x03);
        __delay_ms(5);
    }
    lcd_envia_nibble(0x02);

    for(i=0; i < sizeof(LCD_INIT_STRING); i++)
    {
        lcd_envia_byte(0, LCD_INIT_STRING[i]);
    }
}

//envia nibble para LCD
void lcd_envia_nibble(unsigned char n)
{
    LCD_DADO_4 = !(n & 1);
    LCD_DADO_5 = !(n & 2);
    LCD_DADO_6 = !(n & 4);
    LCD_DADO_7 = !(n & 8);

    __delay(1);
    LCD_ENABLE=1;
    __delay_us(2);
    LCD_ENABLE=0;
}

```



```

//envia byte para o LCD
void lcd_envia_byte(unsigned char address, unsigned char n)
{
    LCD_RS=0;
    __delay_us(60);

    if(address) LCD_RS=1;
    else LCD_RS=0;

    _delay(1);

    LCD_ENABLE=0;

    lcd_envia_nibble(n >> 4);
    lcd_envia_nibble(n & 0xf);
}

//posiciona cursor
void lcd_gotoxy(unsigned char x, unsigned char y)
{
    unsigned char address;

    if(y != 1) address = lcd_line_two;
    else address=0;

    address += x-1;
    lcd_envia_byte(0, 0x80 | address);
}

//envia caracter para display
void lcd_putc(unsigned char c)
{
    switch(c)
    {
        case '\f':
            lcd_envia_byte(0,1);
            __delay_ms(2);
            break;

        case '\n':
            lcd_gotoxy(1,2);
            break;

        case '\b':
            lcd_envia_byte(0,0x10);
            break;
    }
}

```

```

        default:
            lcd_envia_byte(1,c);
            break;
    }
}

//escreve parametros
void lcd_escreve_p(const unsigned char *n, ...)
{
    va_list pa;

    unsigned char i=0,j=0,caracter[6];
    const unsigned char *ptr;
    unsigned int k=0;

    ptr=n;

    while(*n!='\0')
    {
        if(*n=='%')j++;
        n++;
    }

    va_start(pa,j);

    while(*ptr!='\0')
    {
        if(*ptr=='%')
        {
            ptr++;
            if(*ptr=='i')
            {
                k=va_arg(pa,int);
                itoa(caracter,k,10);
                if(caracter[0]!='\0')lcd_putc(caracter[0]);
                if(caracter[1]!='\0')lcd_putc(caracter[1]);
                if(caracter[2]!='\0')lcd_putc(caracter[2]);
                if(caracter[3]!='\0')lcd_putc(caracter[3]);
                if(caracter[4]!='\0')lcd_putc(caracter[4]);
                for(i=0; i<6; i++)caracter[i]='\0';
            }
            if(*ptr=='h')
            {
                lcd_putc('0');
                lcd_putc('x');
                k=va_arg(pa,int);
                itoa(caracter,k,16);
            }
        }
        else
        {
            lcd_putc(*ptr);
            ptr++;
        }
    }
}

```

```
        if (character[0] != '\0') lcd_putc (character[0]);
        if (character[1] != '\0') lcd_putc (character[1]);
        if (character[2] != '\0') lcd_putc (character[2]);
        if (character[3] != '\0') lcd_putc (character[3]);
        if (character[4] != '\0') lcd_putc (character[4]);
        for (i=0; i<6; i++) character[i]='\0';
    }
    if (*ptr=='c')
    {
        k=va_arg (pa, char);
        lcd_putc (k);
    }
}
else
{
    lcd_putc (*ptr);
}
ptr++;
}
}

//escreve strings
void lcd_escreve_s(const unsigned char *y)
{
    while (*y!='\0')
    {
        lcd_putc (*y);
        y++;
    }
}
```

APÊNDICE E – Leitura do Teclado matricial 4x4

```

#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
//#include "hardware.h"
//#include "pic16.h"
#include "lcd_4bit.h"

//pinos teclado 4x4
#define RD0 PORTDbits.RD0
#define RD1 PORTDbits.RD1
#define RD2 PORTDbits.RD2
#define RD3 PORTDbits.RD3

char findKey(unsigned int a, unsigned int b);
char readKeyboard();

char findKey(unsigned int a, unsigned int b) //generating key character
{
    if(b == 0)
    {
        if(a == 3)
            return '*';

        else if(a == 2)
            return '7';
        else if(a == 1)
            return '4';
        else if(a == 0)
            return '1';
    }
    else if(b == 1)
    {
        if(a == 3)
            return '0';
        else if(a == 2)
            return '8';
        else if(a == 1)
            return '5';
        else if(a == 0)
            return '2';
    }
}

```

```
else if(b == 2)
{
    if(a == 3)
        return '#';
    else if(a == 2)
        return '9';
    else if(a == 1)
        return '6';
    else if(a == 0)
        return '3';
}
else if(b == 3)
{
    if(a == 3)
        return 'D';
    else if(a == 2)
        return 'C';
    else if(a == 1)
        return 'B';
    else if(a == 0)
        return 'A';
}
}

char readKeyboard()
{
    unsigned int i = 0;
    for(i=0;i<4;i++)
    {
        if(i == 0)
            PORTC = 0x01;
        else if(i == 1)
            PORTC = 0x02;
        else if(i == 2)
            PORTC = 0x04;
        else if(i == 3)
            PORTC = 0x08;

        if(RD0)
            return findKey(i,0);
        if(RD1)
            return findKey(i,1);
        if(RD2)
            return findKey(i,2);
        if(RD3)
            return findKey(i,3);
    }
    return ' ';
}
```

APÊNDICE F – Código principal para o Controle do Distribuidor de Sementes

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "hardware.h"
#include "pic16.h"
#include "lcd_4bit.h"
#include "teclado4x4.h"
/*
 *
 */
//////////variaveis globais//////////
int cont = 0;
char caracter;
bit flag_interrupcao = 0;

static bit flag_libera_inicio = 0;
static bit flag_sobe_desce = 0;
static bit flag_curva_interrupt = 0;

//////////Rotina principal//////////
void main(void) {

    TRISC = 0x80;//configura portC C7 (pino RX) como entrada
    PORTC = 0x00; // limpar as portas que estão configuradas como saidas
    TRISA = 0xFF;//configura portA como entrada
    inicializa_RS232(9600,1);//modo de alta velocidade
    inicializa_AD();
    init_lcd_4bit();

    INTCONbits.PEIE = 1;//habilita interrupção de perifericos do pic
    INTCONbits.GIE = 1; //GIE: Global Interrupt Enable bit
    //INTCONbits.INTE = 1;
    //imprime("Voltmetro usando a serial! \n\r");

    //USANDO O TIMER1
    PIE1bits.TMR1IE = 1; //habilita interrupcao por overflow de TIMER1
    T1CONbits.TMR1ON = 1; //habilita interrupcap tmr1
    //T1CONbits.TMR1CS = 1; //clock internal(Fosc/4)
    T1CONbits.T1CKPS0 = 1;
    T1CONbits.T1CKPS1 = 0; //prescaler 1/4

```

```

int valorAD;
char teclado = 0;
char flag_fecha_inicio = 0;
char flag = 0;
char flag_fim_sessao = 1;
int contador_erro = 0;
//int total = 0;
int aux = 0;
int contador_graos = 0;
int cont_sessao = 1;
char flag_proxima = 0;
char flag_curva = 0;
int cont_curva = 0;

int cont_subida = 0;
int cont_descida = 0;
while(1){

    while((teclado == 32 || teclado == 0) && flag_fecha_inicio == 0){
        //espera o teclado ser apertado em A e flag é inicializada em zero.
        if(cont_sessao < 2)
            //Se primeira sessao
            lcd_escreve_s("\fAGUARDA INICIO.. ");
        //escreve no LCD
        else{
            // se mais de uma sessao ja tiver ocorrido
            if(flag_proxima == 0){
                //flag para esperar sessoes intermediarias

                lcd_escreve_s("\fRESULTADOS");
                //escreve no LCD
                __delay_ms(3000); //tempo 3s
                lcd_escreve_p("\fCORRETOS: %i", contador_graos);
                //escreve no LCD
                __delay_ms(3000); // tempo 3s
                lcd_escreve_p("\fERROS: %i", contador_erro);
                //escreve no LCD
                __delay_ms(3000); //tempo 3s
                lcd_escreve_p("\fTOTAL: %i", (contador_erro+contador_graos));
                //escreve no LCD
                __delay_ms(3000); //tempo 3s
                if(contador_erro == 0)
                    lcd_escreve_s("\fpor100: 0"); //quando nao ter erro
                else // como o codigo sempre ira se encerrar sem semente,
                    //para o calculo
                    lcd_escreve_p("\fpor100: %i",
                        ((contador_erro-1)*100)/(contador_erro+contador_graos));
            }
        }
    }
}

```

```

        __delay_ms(3000); // tempo 3s
        flag_proxima = 1; //faz flag_proxima = 1,
        //para aguardar nova sessao
    }
}
if(flag_proxima == 1)
    lcd_escreve_s("\fPROXIMA SESSAO..");
    //escreve no LCD
    //enquanto nao ser apertado A

teclado = readKeyboard();
//verifica se teclado é pressionado
if(teclado == 65){ // se A?
    teclado = 0;
    // teclado recebe 0 para poder ser pressionado novamente
    __delay_us(5);
    flag_libera_inicio = 1;
    //comeca o controle do sistema
    flag_fecha_inicio = 1;
    //para nao entrar no laço while anterior
    flag_fim_sessao = 1;
    flag_proxima = 0;
    // flag_proxima para zero, esperando que a sessao se encerre
    contador_graos = 0;

    // zera contador graos
    contador_erro = 0;
    // zera contador erros
    //total = 0; //zera total de graos
}
else
    teclado = 0;
// teclado recebe 0 para poder ser pressionado novamente
}
teclado = readKeyboard();
//verifica se teclado é pressionado
if(teclado == 68){ // se D
    flag_fecha_inicio = 0;
    //faz com que a tecla A possa ser pressionada
    flag_fim_sessao = 1;
    flag_libera_inicio = 0; //encerra-se sessao
    cont_sessao++;
    //conta a sessao para apresentar escrita correta
    teclado = 0;
    // teclado recebe 0 para poder ser pressionado novamente
}
else
    teclado = 0;
// teclado recebe 0 para poder ser pressionado novamente

if(flag_libera_inicio == 1){ //inicia sessao quando flag = 1;
    int_to_char(leitura());
    valorAD = leitura(); //pega o valor AD do conversor

```



```

//lcd_escreve_p("\fVALOR AD: %i", valorAD);
if(valorAD > 600){ // se valor AD maior que 600 ? ERRO!
    PORTCbits.RC4 = 1; //seta o valor de PORTC4 para 1,mostra erro!
    aux++; //aux para controlar erros,
    if(aux < 2 ){
        contador_erro++; // soma o contador de erros
        if(flag == 0) // começa sem semente e espera a primeira
            lcd_escreve_s("\fCHEGANDO GRAO..");
        // nao retorna a esta funcao
        else{ // apresenta o erro
            flag_curva = 1; //flag que indica a compensacao do erro
            cont_curva = 0; //zera contador
            //total++; // total de erros + acertos
            lcd_escreve_p("\fERROS: %i", contador_erro);
            //escreve no LCD
        }
    }
}
else{ // se valor AD < 600 ? CORRETO!
    if(contador_erro == 1 && flag == 0){
        //faz com que quando começa sem semente
        flag = 1; // nao conte erros atoa
        contador_erro = 0;
    }

    else{ //
        PORTCbits.RC4 = 0; //zera PORTC4 que acende LED para erro
        contador_graos++; //conta os graos corretamente colocados
        //total++; // total de erros + acertos
        lcd_escreve_p("\fCORRETOS: %i", contador_graos);
        //escreve no LCD
        cont_curva++; // cont_curva para começar compensacao
        cont_subida++; // contador para vel maxima da curva
        cont_descida++; //contador de vel intermediaria da curva
        flag = 1; // se caso começar com semente
        //para apresentar mensagem correta qndo erro acontecer
        aux = 0; // zera aux, para contagem correta dos erros
    }
}
if(flag_curva == 1 ) //começa a compensacao
{
    if(cont_curva == 3) //espera tres graos, para começar a curva
    {
        flag_curva_interrupt = 1; //habilita curva
        flag_curva = 0; //sai do laço
        cont_subida = 0; //começa a contar subida
    }
}
}

```

```
if(flag_curva_interrupt == 1)//comecando a curva
{
    if(flag_sobe_desce == 0){//entra com aumento da vel
        if(cont_subida == 5){ //por 5 posicoes
            cont_subida = 0;// zera o contador
            flag_sobe_desce = 1; //comeca a descer a curva
            cont_descida = 0; //zera contador
        }
    }

    if(flag_sobe_desce == 1) // curva descendo
    {
        if(cont_descida == 5) // por 5 posicoes
        {
            cont_descida = 0; // zera contador
            flag_sobe_desce = 0; // sai da curva,
            flag_curva_interrupt = 0; //para uma vel constante
            //pré-fixada anteriormente
        }
    }
}

__delay_ms(1000);
//Tempo em que o grao passa no disco do distribuidor
}
```