UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ELETRÔNICA
CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL

FELIPE SCHEIFER
JOÃO HENRIQUE MACIEL DO NASCIMENTO
ROMEO IRAN CALLASSA JUNIOR

DESENVOLVIMENTO DE SISTEMA PARA CONTROLE E SUPERVISÃO DE UMA MÁQUINA MULTIHEAD WEIGHER BASEADO EM HARDWARE MICROCONTROLADO

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA
2014

FELIPE SCHEIFER

JOÃO HENRIQUE MACIEL DO NASCIMENTO

ROMEO IRAN CALLASSA JUNIOR

DESENVOLVIMENTO DE SISTEMA PARA CONTROLE E SUPERVISÃO DE UMA MÁQUINA MULTIHEAD WEIGHER BASEADO EM HARDWARE MICROCONTROLADO

Trabalho de conclusão de curso apresentado à Coordenação de Eletrônica no Campus Ponta Grossa da Universidade Tecnológica Federal do Paraná como requisito parcial para a obtenção da conclusão do curso Superior de Tecnologia em Automação Industrial.

Orientador: Profº. Frederic Conrad Janzen

PONTA GROSSA

2014

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa

Diretoria de Graduação e Educação Profissional

TERMO DE APROVAÇÃO

DESENVOLVIMENTO DE SISTEMA PARA CONTROLE E SUPERVISÃO DE UMA MÁQUINA MULTIHEAD WEIGHER BASEADO EM HARDWARE MICROCONTROLADO.

por

FELIPE SCHEIFER, JOÃO HENRIQUE MACIEL DO NASCIMENTO E ROMEO IRAN CALLASSA JUNIOR.

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 21 de fevereiro de 2014 como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial. Os candidatos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Frederic Conrad Janzen
Prof. Orientador

Angelo Marcelo Tusset
Membro titular

Julio Cesar Guimarães
Membro titular

Julio Cesar Guimarães
Responsável pelos Trabalhos
de Conclusão de Curso

Márcio Mendes Casaro
Coordenador do Curso
UTFPR - Campus Ponta Grossa

Dedicamos...

A todos os amigos e familiares que nos acompanharam nessa caminhada.

AGRADECIMENTOS

Em primeiro lugar, agradecemos a Deus, pela benção da inteligência, saúde, amizade e principalmente por jamais ter nos deixados sozinhos na procura dos nossos sonhos.

Um agradecimento muito especial as nossas famílias pelo apoio, incentivo e compreensão na realização deste trabalho.

Nosso agradecimento ao Profº. Frederic Conrad Janzen pela paciência e atenção que teve para conosco ao longo deste projeto.

E finalmente, aos nossos amigos e professores que, de forma direta ou indireta, contribuíram para a realização desse projeto.

RESUMO

JUNIOR, Romeo Iran Callassa; NASCIMENTO,João Henrique Maciel; SCHEIFER, Felipe. **Desenvolvimento de sistema para controle e supervisão de uma máquina Multihead Weigher baseado em hardware microcontrolado.** 2014. 70 f. Trabalho de Conclusão de Curso de Tecnologia em Automação Industrial – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

Este trabalho apresenta o desenvolvimento de um sistema de controle e supervisão de uma máquina Multihead Weigher, utilizando hardware microcontrolado como alternativa de automação do equipamento. Apresenta os principais conceitos relacionados a programação de microcontroladores, protocolos de comunicação, controle de motor de passo e aquisição de dados analógicos de um sistema de pesagem, utilizando-se de célula de carga. Demonstra em etapas os protótipos desenvolvidos em laboratório, com base no sistema de controle proposto, bem como os resultados obtidos durante os testes.

Palavras-chave: Multihead Weigher. Sistema Microcontrolado. Protocolo CAN. Conversor Analógico-Digital. Célula de Carga.

ABSTRACT

JUNIOR, Romeo Iran Callassa; NASCIMENTO,João Henrique Maciel; SCHEIFER, Felipe. **Development of a control and supervision system for a Multihead Weigher machine based on microcontroller hardware.** 2014. 70 f. Trabalho de Conclusão de Curso de Tecnologia em Automação Industrial – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

This paper presents the development of a control and supervision system of a Multihead Weigher machine using a microcontroller hardware as an alternative in the automation of the equipment. Introduces key concepts related to microcontrollers programming, communication protocols, control pitch and acquisition of analog data of a weighing system, using load cell engine. Demonstrates in steps the prototypes developed in the laboratory, based on the proposed control system and the results obtained during the tests.

Keywords: Multihead Weigher. Microcontrolled system. CAN protocol. Analog-to-Digital Converter. Load cell.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

USART – Universal Synchronous Asynchronous Receiver Trasmitter
USB – Universal Serial Bus
CAN – Controller Area Network
SPI – Serial Peripheral Interface
I2C – Inter Integrated Circuit
PC – Computador Pessoal (do original Personal Computer)
CPU – Central Processing Unit
I/O – Entradas/Saídas (do original Inputs/Outputs)
IHM – Interface homem máquina
Conversor A/D – Conversor Analógico/Digital
CLP – Controlador Lógico Programável
PIC – Peripheral Interface Controller
RISC – Reducer Instruction Ser Computer
ROM – Real Only Memory
OTP – One Time Programmable
EPROM – Erasable Programmable Read-Only Memory
SAE – Society of Automotive Engineers
ISO – International Organization for Standardization
OSI – Open Systems Interconnection
CAN_H – Controller Area Network High
CAN_L – Controller Area Network Low
CSMA – Carrier Sense Multiple Access
NDBA – Non-Destructive Bitwise Arbitration
RTR – Remote Transmit Request
SRR – Substitute Remote Request
IDE – Integrated Development Environment
DLC – Data Length Code
ACK – Acknowledgment
SCK – Clock
MSSP – Master Synchronous Serial Port
CCP – Capture Compare PWM

mV – Milivolt

CI – Circuito Integrado

CC – Corrente continua

CA – Corrente Alternada

V – Volt

A – Ampere

LED – Light

SUMÁRIO

INTRODUÇÃO

Na constante busca por qualidade e produtividade dentro da manufatura industrial, a tecnologia veio para revolucionar o modo de se produzir. A partir da implantação de novas tecnologias foi possível o melhor aproveitamento de recursos físicos e humanos, refletindo no desempenho e lucratividade das empresas que investem em tecnologia. Devido a essa demanda, a necessidade de inovar tornou-se essencial.

Dentro do ambiente de automação os grandes avanços se devem em grande parte ao desenvolvimento e a evolução dos microcontroladores e microprocessadores, possibilitando uma forma de controle rápido, seguro e preciso das máquinas desenvolvidas.

Hoje os microcontroladores possuem diversas funções integradas e permitem uma vasta gama de aplicações. Desde simples cálculos numéricos internamente, a controles de processos fabris inteiros, recolhendo dados de sensores na planta (digitais ou analógicos), controle de motores através de módulo PWM, comunicação em rede com demais dispositivos da planta (USART, USB, CAN, SPI, I2C), além de oferecer inúmeras velocidades e capacidades de memória em seus diversos modelos, tornando-se extremamente úteis no desenvolvimento de projetos de automação.

Em virtude da necessidade de automação dos processos de dosagens de produtos sólidos, surgiu em 1972 o primeiro Multihead Weigher (Balança de múltiplos cabeçotes), um equipamento automatizado, que permitia a dosagem precisa de embalagens de produtos sólidos.

Com o uso largamente difundido hoje nas indústrias, o Multihead Weigher possui seu controle baseado em um sistema de balanças, dispostas de forma circular em torno de uma plataforma de dosagem (a quantidade de balanças varia dependendo do modelo, geralmente em torno de 8 a 32). O produto destinado a envase cai na plataforma através de um sistema de tubulação e é dosado nas balanças através de um sistema de vibração da plataforma de pesagem. O sistema então realiza um cálculo entre todas as balanças, selecionado a combinação de balanças que deverão ser dosadas na embalagem final, de forma a manter o peso dentro do desejado. O equipamento repete ciclicamente esses passos (dosagem

nas balanças, cálculo, dosagem na embalagem final), realizando assim uma dosagem rápida e precisa.

O presente projeto visa desenvolver um sistema distribuído microcontrolado para comando de um Multihead Weigher, integrando o recolhimento de dados das balanças (leituras de peso) através de placas escravas, interligando à uma placa central de controle, através de rede de comunicação baseada no protocolo CAN (Controller Area Network).

1.1 TEMA DA PESQUISA

Desenvolvimento de sistema para controle e supervisão de uma máquina Multihead Weigher baseado em hardware microcontrolado.

1.1.1 Delimitação do Tema

O projeto será desenvolvido propondo um sistema distribuído de controle do Multihead Weigher, utilizando-se de microcontroladores integrados via rede CAN. Para tal será necessário o desenvolvimento de placas de aquisição de dados escravas, responsáveis pela leitura de dados das balanças e comando de motores de dosagem, conectadas através da rede CAN à placa central mestre, responsável pelo cálculo e controle lógico do sistema.

A interface homem-máquina será realizada através de um PC (Personal Computer), comunicando pela porta serial com o microcontrolador central.

1.2 PROBLEMA

A utilização de um sistema de automação tradicional, com controladores lógicos programáveis, acarreta um custo elevado no desenvolvimento. A compra de *hardware* (CPU, placas I/O analógicas e digitais, IHM), bem como o de *software* dedicado para programação, refletiria em um alto preço do equipamento final, tornando-se assim fora da realidade do mercado de máquinas industriais.

1.3 HIPÓTESE / PREMISSA

Com a implantação de um sistema microcontrolado, acredita-se que é possível o desenvolvimento de um sistema de controle e supervisão de uma máquina Multihead Weigher de baixo custo de automação, sem perdas de eficiência ou precisão do equipamento.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Desenvolvimento de sistema de controle automatizado de um Multihead Weigher.

1.4.2 Objetivos Específicos

- Levantamento da bibliografia sobre microcontroladores;
- Levantamento da bibliografia sobre rede CAN;
- Levantamento da bibliografia sobre sistemas de pesagem;
- Levantamento da bibliografia sobre Supervisórios;
- Levantamento do hardware necessário;
- Programação microcontrolador escravo (comando de motores e tratamento de dados do conversor A/D);
- Programação do microcontrolador mestre;
- Programação da rede CAN (mestre-escravos);
- Programação do supervisório;
- Confecção das placas mestre e escravos, para testes em bancada;

1.5 JUSTIFICATIVA

O motivo para o desenvolvimento do projeto é a elaboração de um sistema de automação para Multihead Weigher que possibilite uma alta confiabilidade, velocidade de processamento da operação, além de um baixo custo de implantação. O desenvolvimento utilizando microcontroladores mostra-se o mais viável em todos os aspectos, pois além de possuírem todas as funcionalidades necessárias ao projeto, tem custo reduzido de desenvolvimento.

Um sistema utilizando CLP apresenta as vantagens de módulos prontos, além da confiabilidade e facilidade de desenvolvimento do programa, porém tem um custo que excede o esperado para uma automação desse nível. Nesse ponto vemos a vantagem dos microcontroladores: é possível alcançar o mesmo nível de automação, montando um sistema robusto e preciso, dependendo apenas da programação elaborada e da qualidade do *hardware* desenvolvido.

Além das vantagens na implantação do projeto, temos também a facilidade de manutenção nesse sistema, por se tratar de um sistema distribuído, e composto de simples placas de controle, montadas com componentes eletrônicos de baixo custo.

1.6 MÉTODO DA PESQUISA

Este projeto será uma pesquisa aplicada com o propósito de desenvolver uma automação específica a um equipamento.

No processo de pesquisa haverá busca por informações junto a profissionais, artigos, livros, e obras já publicadas.

Serão realizados testes via *softwares* e em laboratório, para garantir que ao ser implementado apresente todas as funcionalidades predefinidas.

.

2 DESENVOLVIMENTO

2.1 MULTIHEAD WEIGHER

O Multihead Weigher é um equipamento de pesagem combinatória, composto por uma série de módulos de pesagem independentes, chamados de cabeçotes, conectados a uma unidade central de processamento, responsável pelo controle lógico e cálculo combinatório, conectada a uma *interface* homem-máquina.

Figura 1: Multihead Weigher.
Fonte: [YAMATO, 2014].

Geralmente um sistema de pesagem combinatória é composto por um conjunto de cabeçotes idênticos, sendo cada um equipado com sua própria célula de carga, para realização da pesagem do produto, além de um atuador eletromecânico (comumente motor de passo) para o controle do sistema mecânico de dosagem. Um modelo típico prevê os cabeçotes dispostos de forma circular, sendo a alimentação de produto para pesagem realizada radialmente do centro da máquina, utilizando-se de plataforma vibratória. A combinação dos pesos contida nos recipientes de pesagem dos módulos que mais se aproxima ao peso necessário é descarregado

em um dispositivo de transferência, por exemplo, um cone ou funil, o qual transporta o produto para um sistema de embalagem.

Na figura 2 podemos visualizar a representação geral dos módulos de um Multihead Weigher típico:

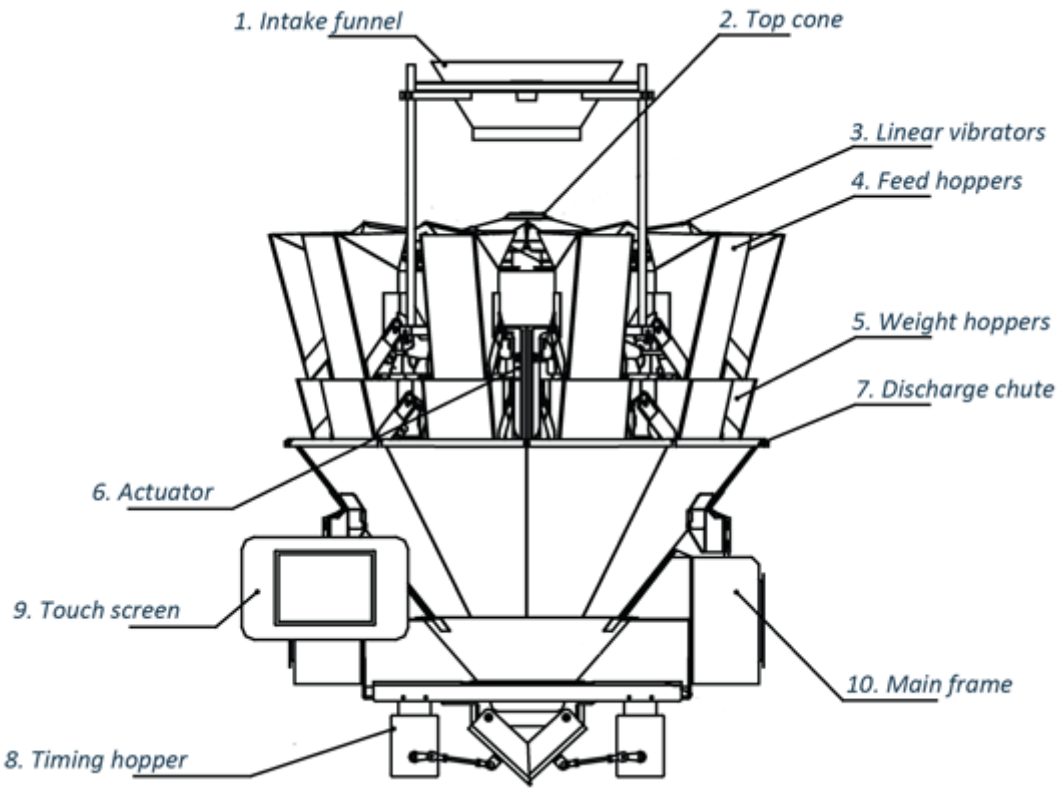


Figura 2: Diagrama Multihead Weigher.
Fonte: [YAMATO, 2014].

1- Funil de admissão;
2- Prato radial de separação de produto;
3- Módulo vibratório linear;
4- Cabeça de alimentação;
5- Cabeça de Pesagem;
6- Atuador sistema de dosagem;
7- Calha de descarga;
8- Controle de saída de produto (temporizado);
9- Interface homem-máquina;
10- Quadro principal;

Na figura 3 vemos a representação gráfica exemplificando o sistema de cálculo combinatório. Nota-se que a dosagem possui um range de tolerância ajustável (100g~105g), de forma a manter as características de velocidade do equipamento, dentro do limite aceitável do processo.

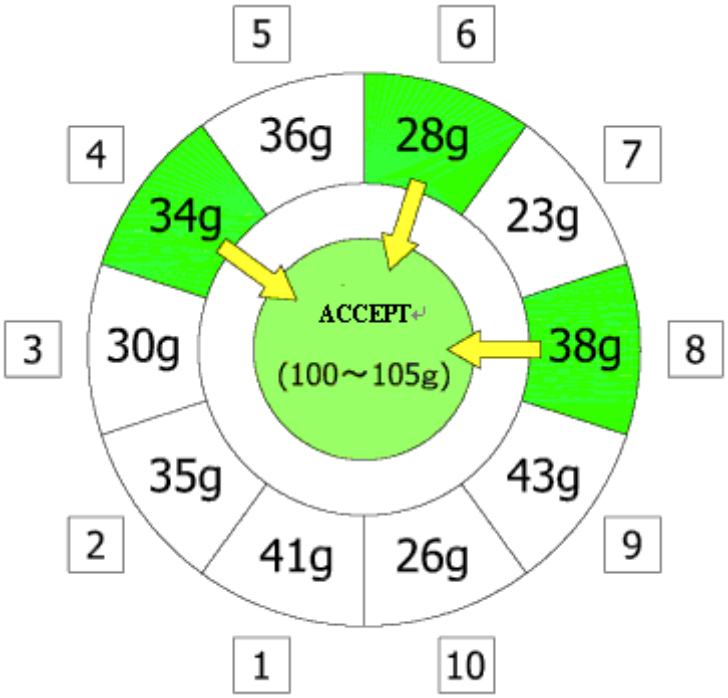


Figura 3: Cálculo combinatório.
Fonte: [YAMATO, 2014].

2.2 MICROCONTROLADOR PIC

A partir da década de 70, para uma melhor eficiência no processamento de dados, começou-se a utilizar microprocessadores em computadores. O microprocessador da Intel foi um dos precursores. A partir daí, cada vez mais os micro componentes começaram a ser utilizados buscando reduzir circuitos maiores em um único componente.

Com base na arquitetura de um microprocessador e seus periféricos, surgiram os microcontroladores que comportavam todo um sistema equivalente a um microprocessador e seus periféricos, o diferencial de um microcontrolador em

relação a um computador é que o microcontrolador processa um único programa com propósito específico que fica armazenado na memória de programa [PEREIRA, 2007].

Os microcontroladores são chips que consistem num circuito processador que possui entradas, saídas e uma memória, para projetar com esse componente é preciso saber programar o circuito para que ele faça o que desejamos [BRAGA, 2010].

No início dos anos 90, foi fundada a Microchip, que desenvolve os microcontroladores PIC (Peripheral Interface Controller), que tinha intuito de conseguir um microcontrolador barato, pequeno e prático.

2.2.1 Arquitetura do Microcontrolador PIC

A empresa Microchip utiliza dois tipos de arquiteturas nos seus chips, a RISC e a Harvard, na primeira o microcontrolador faz tudo usando poucas instruções básicas, pois cada instrução pode ser executada em apenas um ciclo do *clock*, na arquitetura Harvard segundo [BORGES; PAIVA; PIEDADE, 2008], a leitura pode ser feita ao mesmo tempo em que as instruções são executadas, o sistema fica o tempo todo executando instruções que acarreta em um ganho significativo de velocidade, enquanto uma instrução é executada a seguinte já está sendo lida.

2.2.2 Memórias

As memórias disponíveis nos microcontroladores podem ser de dois tipos: memórias de programa e memórias de dados.

2.2.2.1 Memórias de programa

Memórias de programa tem como função armazenar o *software* a ser executado, são memórias não voláteis, ou seja, o programa não é perdido quando se retira a alimentação do sistema, sendo assim cada vez que o sistema é desligado e colocado em funcionamento não há necessidade de programar novamente, são utilizados memórias do tipo ROM, OTP, EPROM e Flash.

Segundo [PEREIRA, 2005] a memória de programa é mapeada de forma que cada endereço tenha 8 *bits*, porém as instruções armazenadas na memória de programa tem 16 ou 32 *bits*, assim cada instrução ocupa dois endereções de memória e o Contador de Programa ao ser executado incrementa de dois em dois endereços e pelo fato do barramento de instrução conter 16 bits a leitura dos dois endereços é simultânea, formando uma instrução de 16 bits (instruções curtas) ou 32 bits (instruções longas).

2.2.2.2 Memórias de dados

Memória de dados é onde são armazenados os dados a serem processados pelo computador, como é constantemente alterada a memória utilizada é do tipo RAM, sendo uma memória volátil, ou seja, quando a alimentação dela é cortada os dados são perdidos [ADRIANO; MARÇANO, 2009].

2.2.3 Famílias

Pelo fato de efetuar tarefas especificas, A Microchip possui uma variedade muito grande de microcontroladores da série PIC no mercado, diferenciando-se pelo número de entradas e saídas, pelos recursos de periféricos do dispositivo e pelo meio de comunicação que ele disponibiliza para com outros equipamentos, por este motivo os microcontroladores da Microchip são divididos em famílias, cada família tem vários componentes com tamanhos e recursos diferentes:

- PIC10;
- PIC12;
- PIC14;
- PIC16;
- PIC17;
- PIC18;
- PIC24F / PIC24H
- PIC32;

2.3 PROTOCOLO CAN

2.3.1 Histórico

A rede CAN foi desenvolvida pela empresa Robert Bosch na década de 80 cujo objetivo era promover a interconexão entre dispositivos de controle em automóveis. Em Fevereiro de 1986, a Robert Bosch GmbH introduziu o barramento CAN no congresso Society of Automotive Engineers (SAE), onde foi apresentada a versão 1.0. Com a rápida expansão e aceitação do protocolo no setor industrial, surgiu a necessidade de maior flexibilidade no protocolo, culminando na publicação da versão 2.0 em 1991 [AZEVEDO; BATISTA; VARELA, 2009].

A versão publicada associa as versões 2.0A e 2.0B, sendo a 2.0A (11 *bits*), com 2032 identificadores, e a versão 2.0B(29 *bits*) com identificadores expandidos, chegando a 537 milhões de identificadores possíveis.

O interesse gerado acabou levando, em 1993, à elaboração da ISO 11898, cobrindo as duas camadas inferiores do modelo de referência para comunicações ISO/OSI – Open Systems Interconection [SANCHO,2009].

2.3.2 Características

O protocolo CAN tem sido bastante utilizado em aplicações de controle em tempo real, possui a característica de ser multi-mestre, ou seja, onde todos os módulos ligados a ela podem se tornar mestre em um momento e escravo num outro e trabalhar com mensagem *multicast,* onde todas as mensagens são recebidas por todos os módulos da rede (figura 4) [AZEVEDO; BATISTA; VARELA, 2009]

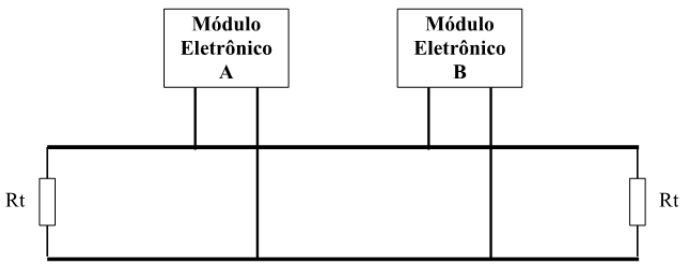


Figura 4: Modelo barramento CAN.
Fonte: [AZEVEDO; BATISTA; VARELA, 2009].

Entre vantagens da rede CAN está a velocidade da transmissão de dados de 1 Mbps considerando um comprimento de barramento de até 40 metros, excelente detecção de erros, mensagens curtas de até 8 bytes por mensagem e controle da rede por prioridade nas mensagens [GUIMARÃES, SARAIVA, 2002].

2.3.3 Modelo OSI

No modelo OSI, o protocolo CAN encontra-se posicionado em duas camadas: a camada física e a camada de ligação de dados [SANCHO, 2009].

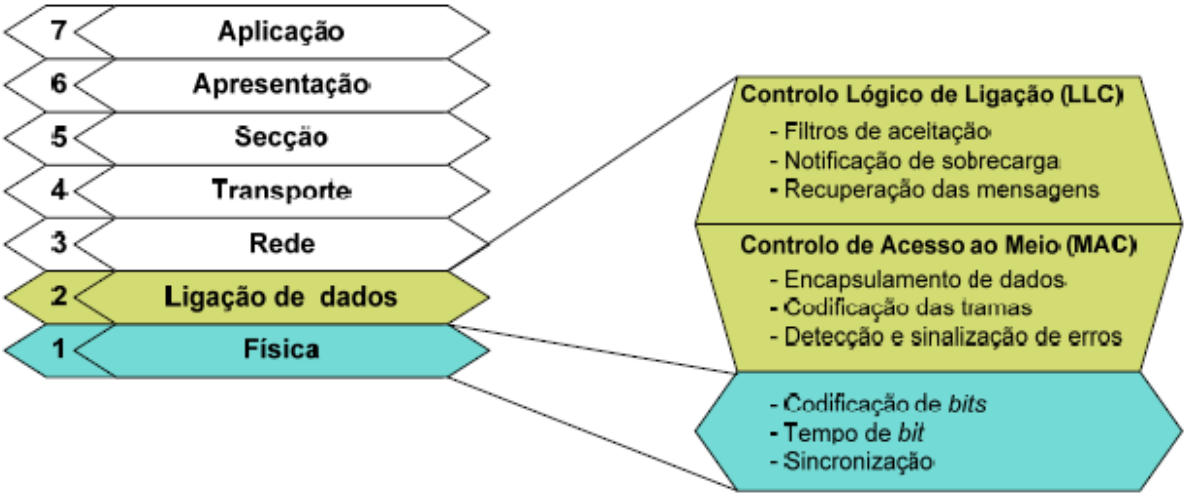


Figura 5: Modelo camada OSI.
Fonte: [SANCHO, 2009].

2.3.4 Camada Física

A camada física trata de aspectos como a temporização, a codificação de *bits*, a sincronização dos mesmos e quais os cabos e conectores que serão utilizados na instalação da rede [SANCHO, 2009].

O barramento CAN pode ser montado com apenas dois fios, CAN_H (CAN *High*) e CAN_L (CAN *Low*). A comunicação é baseada na transmissão diferencial, onde a diferença entre os dois fios é avaliada para definição do nível lógico do *bit.* Numa rede CAN os sinais elétricos digitais são representados pelos níveis recessivo ou dominante [BARBOSA; 2003]. A figura 6 ilustra os níveis de tensão para bits recessivos ou dominantes.

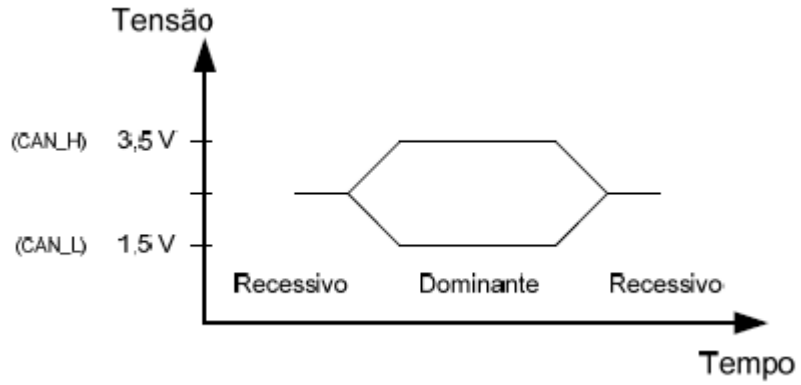


Figura 6: Níveis de tensão bit recessivo ou dominante.
Fonte: [BARBOSA, 2003].

2.3.5 Camada de Ligação de Dados

No nível da camada de ligação de dados são definidas a construção das mensagens, a manipulação, controle de transmissão, identificação do pacote de dados, controle de acesso ao barramento e verificação de erros nas transmissões ou conteúdo das mensagens [SANCHO, 2009].

Para evitar a colisão de dados durante as transmissões de dados entre os dispositivos da rede, é utilizado o método de arbitragem CSMA/NDBA (Carrier Sense Multiple Access with Non-Destructive Bitwise Arbitration), que realiza um processo de definição de prioridades entre os dados transmitidos dos dispositivos [SOUZA, 2002].

Os erros e problemas de transmissão são armazenados em contadores, em um determinado dispositivo, como propósito de limitar a quantidade de erros. Caso um dispositivo da rede exceda o número de erros ele pode ser desligado automaticamente da rede [SANCHO, 2009].

2.3.6 Processo de Arbitragem

O método CSMA define como base de operação que para um nó transmitir, é necessário aguardar que o barramento CAN esteja desocupado. Iniciado a transmissão, é verificado ao mesmo tempo se outro não iniciou simultaneamente uma transmissão, através dos sinais do barramento. Entra então em ação o método NDBA, onde ocorre a comparação bit-a-bit do identificador de mensagens, ou seja,

em cada nó que é disputado a transmissão, o *bit* do identificador é comparado com o *bit* presente no barramento, se for igual a transmissão continua. Quando um nó transmite um *bit* recessivo (1 lógico) e no barramento se encontra um *bit* dominante (0 lógico), este aborta de imediato a sua transmissão e aguarda que o barramento fique livre para iniciar nova transmissão [SOUZA,2002]

.

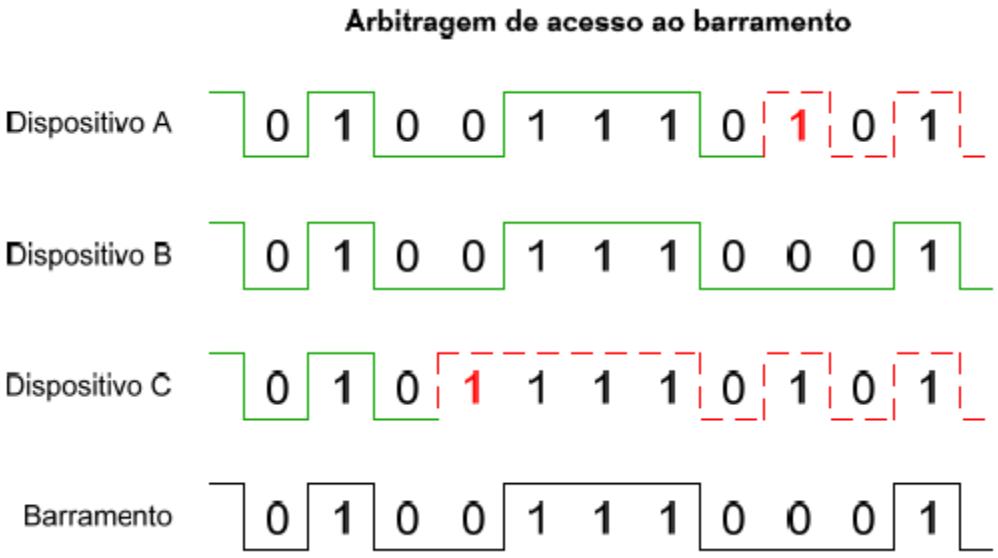


Figura 7: Arbitragem de acesso ao barramento.
Fonte: [SOUZA, 2002].

Na figura 7, temos o exemplo de três dispositivos que iniciam simultaneamente a transmissão no barramento. Entra em ação então o método NDBA, onde os dispositivos comparam o dado no barramento ao seu próprio identificador. No quarto bit transmitido, o dispositivo C encontra um bit dominante, enquanto o seu identificador possui um bit recessivo, cessando então imediatamente a sua transmissão e entrando em modo de aguardo do barramento livre. O mesmo acontece com o dispositivo A quando o bit número nove comparado. O dispositivo B, que possui o menos identificador, ganha acesso ao barramento e continua então sua transmissão na rede.

2.3.7 Tipo de Quadros (*Frame*)

Dentro do protocolo CAN a transferência de dados pode ser feita através de 4 tipos de quadros:

- *Data Frame*
- *Remote Frame*
- *Error Frame*
- *Overload Frame*

A seguir serão detalhados os tipos de quadros definidos.

2.3.7.1 *Data Frame*

O *data frame* ser usado tanto no formato padrão (2.0A) ou estendido (2.0B) e é o único que permite o envio de informação de até 8 *bytes*(64 *bits*). È o tipo de quadro utilizado para transferência de informação entre os dispositivos da rede [HUBERT,2001].

O início de quadro é composto de um *bit* dominante (formato padrão ou estendido), marcando o início do *data frame.*

O campo de arbitragem no formato padrão é composto por um identificador de 11 *bits* e o bit RTR (Remote Transmit Request), que diferencia entre *data frame* (dominante) e *remote frame* (recessivo).

Com o formato estendido é formado por 29 *bits* de identificador, o *bit* SRR (Substitute Remote Request), que garante prioridade para mensagens de formato padrão caso ambas tenham o mesmo identificador base. Possui também o *bit* IDE (dominante para formato padrão e recessivo para estendido) e o *bit* RTR [HUBERT, 2001].

O campo de controle é um campo de 6 *bits*, seja para o formato padrão ou estendido. Para o padrão o primeiro *bit*, é o *bit* IDE, que deve ser dominante, enquanto que para estendido é o *bit* R1, que deve ser dominante. Segue-se, em ambos os formatos, o *bit* reservado R0, que deve ser enviado como dominante, seguindo-se de quatro *bits* que compõem o DLC (*Data Lenght Code*), que indica o tamanho do campo de dados em *bytes*, que pode variar de 0 a 8 [SANCHES, 2009].

O campo de dados corresponde à informação que se deseja enviar, comportando de 0 a 8 bytes (64 bits).

O *checksum* é composto por 15 bits, e permite a detecção de erros na mensagem.

O campo ACK (Acknowledgment) é formado por 2 *bits*, ACK SLOT e ACK Delimiter. O dispositivo que enviar uma mensagem coloca estes dois *bits* como

recessivos, qualquer dispositivo que receba a mensagem reescreve o *bit* ACK *SLOT* como dominante, sinalizando assim ao emissor que a mensagem foi recebida pelo menos por um dispositivo.

O último campo é o fim de quadro, composto por uma sequência de 7 bits recessivos [HUBERT,2001].

2.3.7.2 *Remote Frame*

Utilizado para solicitação de informação de um dispositivo conectado a rede CAN. O nó solicitado responde enviando seu quadro de dados, com o mesmo identificador do quadro remoto.

Possui formato semelhante ao do *data frame*, porém sem o campo de dados, e o bit RTR é enviado como recessivo.

2.3.7.3 *Error Frame*

Toda vez que um erro é detectado esse quadro é transmitido por qualquer nó da rede. Possui os campo de erro, que é uma sequência de 6 bits consecutivos, e o campo delimitador que marca o fim da error frame.

O envio de 6 *bits* consecutivos (dominantes ou recessivos), ocasiona a quebra da regra de *Bit Stuffing*, que permite o envio de no máximo 5 bits consecutivos de mesmo valor. Com isso os contadores da rede são incrementados [SANCHO,2009].

2.3.7.4 *Overload Frame*

Em geral são utilizadas com o objetivo de atrasar o próximo envio de um *data frame* ou *remote frame*, evitando erros nas transmissões.

Compostas por dois campos: *flag* de sobrecarga (6 *bits* dominantes) e delimitador de sobrecarga (8 *bits* recessivos)[SANCHO,2009].

2.3.8 Filtragem de mensagens

No protocolo CAN é possível programar um sistema de filtragem de mensagens, onde o dispositivo programado ignora a mensagem que não possua o filtro correto. É realizado através do identificador de mensagens, sendo possível utilizar o identificador completo, ou então mascarar o identificador, selecionando apenas os bits que se deseja filtrar, permitindo a filtragem de determinados grupos de mensagens [HUBERT, 2001].

2.4 PROTOCOLO SPI (*SERIAL PERIPHERAL INTERFACE*)

O protocolo SPI (*Serial Peripheral Interface)* é um protocolo síncrono (com clock), trata-se de uma comunicação serial que utiliza um dispositivo mestre para iniciar a comunicação com um dispositivo escravo e os dados são trocados entre eles, permite a comunicação entre dois ou mais dispositivos em alta velocidade com 8 *bits* de dados [SOUZA; LAVINIA, 2005].

No protocolo SPI cada dispositivo tem duas linhas de dados, uma para entrada e outra para saída, estas trocas de dados são controlados pela linha de *clock* (SCK), a qual é controlada pelo dispositivo mestre que gera o *clock* de sincronismo.

A comunicação SPI é feita apenas entre dois dispositivos, pois não permite endereçamento. O diagrama abaixo mostra a ligação entre dois componentes que podem ser, por exemplo, dois PICs:

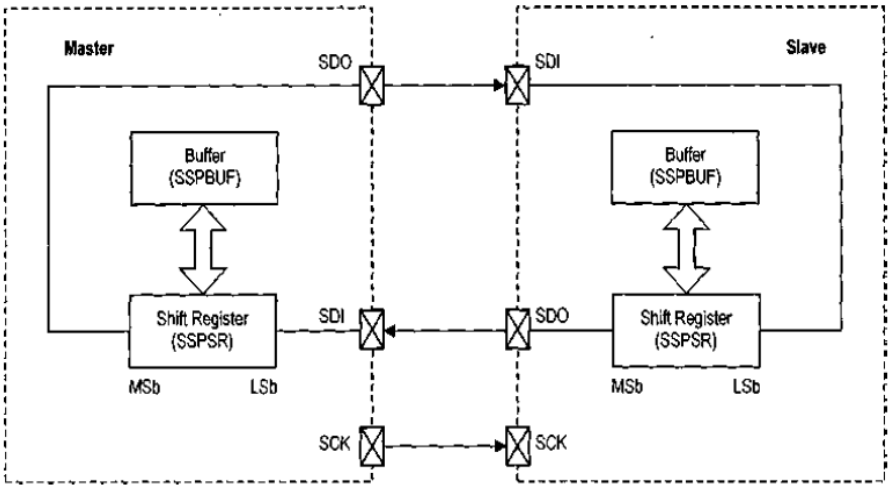


Figura 8: Diagrama de ligação SPI entre dois componentes.
Fonte: [SOUZA, LAVINIA, 2005].

A comunicação ocorre da seguinte maneira: quem manda na comunicação é o mestre, um valor qualquer é escrito no registrador, então o valor é movido para o registrados SSPSR, em seguida oito pulsos são gerados na saída de *clock* SCK, cada *clock* transmite 1 *bit*, ao término dos oito pulsos o valor que estava no SSPSR mestre é trocado com o valor que estava no SSPSR escravo. Tanto de um lado quanto do outro o valor de SSPSR é então movido para o registrador SSPBUF e são ativados *flags* para informar ao sistema o fim da comunicação, e os dados então podem ser acessados novamente em SSPBUF [SOUZA; LAVINIA, 2005].

Em comunicação com mais de dois escravos, existe uma linha dedicada para selecionar o dispositivo escravo com qual se quer comunicar (CS ou SS), chamado *chip select*, a comunicação não é feita por meio de endereçamento.

2.5 COMUNICAÇÃO SERIAL (USART)

A comunicação serial USART (*Universal Synchronous Asynchronous Receiver Trasmitter*) é dividida em dois modos distintos de trabalho: o sincronizado e não-sincornizado.

2.5.1 Comunicação Serial Assíncrona

A comunicação serial assíncrona é feita somente com duas vias, uma utilizada para transmissão e outra para recepção do sinal, possibilitando que as informações sejam enviadas e recebidas ao mesmo tempo, cada uma em sua via, que recebe o nome de *Full Duplex* [SOUZA; LAVINIA, 2005].

A comunicação assíncrona por *hardware* feita em dispositivos que contam com uma Usart interna, que é o caso de PICs, pode utilizar um simples conjunto de funções para acesso a ela e deixa o programa principal livre para tarefas mais importantes [PEREIRA, 2005].

2.5.2 Comunicação Serial Síncrona

Igual ao modo assíncrono também é utilizado duas vias, porém na comunicação síncrona uma via é utilizada para dados e outra para *clock,* isto impossibilita a transmissão e recepção simultânea de dados, essa comunicação é chamada *Half Duplex* [SOUZA; LAVINIA, 2005].

2.6 LINGUAGENS DE PROGRAMAÇÃO

2.6.1 Linguagem de Programação Assembly

A linguagem Assembly é uma forma de representação de código de máquina usando mnemônicos, ou seja, abreviações de termos usuais que descrevem a operação efetuada pelo código de máquina. A conversão dos mnemônicos em códigos binários executáveis pela máquina é feita por um tipo de programa chamado Assembler (montador) [PEREIRA, 2005].

Alguns modelos da série PIC16 e praticamente todas as famílias anteriores a ela utilizavam apenas programas feitos em linguagem Assembly. O desenvolvimento de programa nesta linguagem eleva o tempo e o custo de criação de uma aplicação, devido ser uma linguagem de baixo nível, ou seja, seus comandos são muito próximos da linguagem de máquina e consequentemente mais complexos.

No entanto, os programas em Assembly são muito eficientes, devido à proximidade com o *hardware* do microcontrolador, sendo muito mais rápidos que os programas feitos em outras linguagens.

2.6.2 Linguagem de Programação C

A linguagem de programação C mais utilizada em microcontroladores nos dias atuais foi criada em 1972, por Dennis Ritchie, ela consiste em uma linguagem de programação genérica desenvolvida para ser eficiente, rápida, bem estruturada e lógica [PEREIRA, 2007].

A utilização de uma linguagem de alto nível como C faz com que o programador preocupe-se mais com a programação da aplicação em si, já que o compilador cuida das tarefas como o controle e localização das variáveis, operações matemáticas e lógicas, verificação de banco de memórias, etc [PEREIRA, 2005].

2.7 CÉLULA DE CARGA

Em 1856, o professor da Universidade de Coimbra Royal Society of London, William Thomson (Lord Kelvin) notou que a resistência elétrica de um condutor, fio de cobre e/ou ferro, variava quando eles era submetidos a uma deformação elástica.

As observações de Kelvin foram decorrentes da relação da resistência elétrica do condutor com algumas de suas propriedades físicas, segundo a equação:

$$R = \rho.L / A,$$

onde:

R = resistência elétrica

ρ = constante de condutividade

L = comprimento do condutor

A = área da seção transversal do condutor

Sendo assim, quando o condutor é esticado, ele também diminui o volume, ou seja, a área da seção transversal, como o aumento do comprimento é diretamente proporcional a resistência elétrica e a área é inversamente proporcional, esticando o condutor aumenta sua resistência à passagem de corrente elétrica.

Apenas a partir da década de 1930 a 1940, que foi aplicado a construção de extensômetros (strain gages)(figura 9), um resistor feito de um finíssimo material condutor colocado sobre um composto isolante, que tem seu princípio de funcionamento voltado para a variação de resistência quando é submetido a uma pequena deformação. Tem como características principais a alta precisão de medição e o baixo custo, dentre outras, até hoje é utilizado na construção de células de carga.

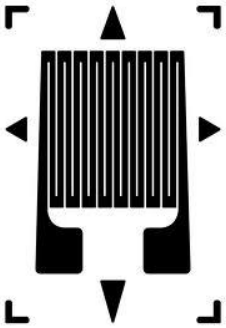


Figura 9: Representação de um Strain Gauge.
Fonte: [ANDOLFATO; CAMACHO; BRITO, 2004].

As balanças mecânicas foram substituídas pelas balanças eletrônicas, elas que possuíam braços mecânicos e réguas graduadas, passaram a utilizar células de cargas e displays, as balanças eletrônicas facilitaram o manuseio e diversidade de áreas aplicadas com sua tecnologia de última geração [MOREIRA 2005].

Os extensômetros são utilizados para a confecção das células de cargas, através da ponte de Wheatstone, onde temos uma tensão de entrada e podemos calcular a tensão de saída conforme a resistência gerada pelo circuito (figura 10).

A ponte é circuito mais usado, pois apresenta maior precisão para medidas de pequenas variações de resistência [ANDOLFATO; CAMACHO; BRITO, 2004].

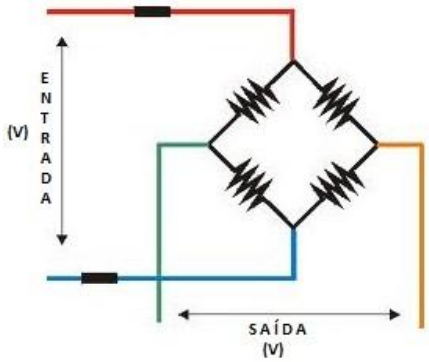


Figura 10: Ponte de Wheatstone completa.
Fonte: [ANDOLFATO; CAMACHO; BRITO, 2004].

Encontramos 4 tipos diferentes de ponte montadas com extensômetros, ¼ de ponte, ½ ponte assimétrica, ½ ponte simétrica e ponte completa (figura 11). Para montagem de transdutores esta ultima é a mais recomendada devido a uma maior precisão no processo de medição.

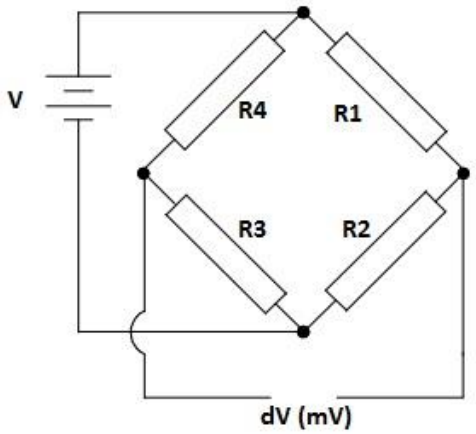


Figura 11: Ponte de Wheatstone completa.
Fonte: [ANDOLFATO; CAMACHO; BRITO, 2004].

Na figura 11 podemos observar que se alimenta a ponte com uma tensão (no máximo 15 VDC/VCA), R1, R2, R3 e R4 são resistores extensômetros, quando R4.R2 = R1.R3 e dV=0, a ponte de Wheatstone está calibrada, quando elas são submetidos a uma força, sofrem deformação e geram um sinal de saída em mV proporcional à força sofrida pelos extensômetros. É através da medição desse sinal de saída que se obtém o valor da força aplicada.

A função da célula de carga é converter pressão física em pulsos elétricos. Os extensômetros são colocados em uma peça de alumínio ou liga de aço, denominada corpo da célula, a força a ser medida é aplicada ao corpo da célula, logo sua deformação é transmitida aos extensômetros, que medirão sua intensidade. A forma e características do corpo da célula variam conforme a aplicação, hoje se encontra uma grande variedade no mercado, elas podem ser utilizadas com várias funções diferentes, em prensas para se calcular pressão, em balanças comerciais, em cabo de aço para controle do peso da carga, automatização e controle de processos industriais, dosagem, enchimento, dentre várias outras aplicações. Abaixo um exemplo de célula de carga a MR-XX da Micro Análise, estas são destinadas as mais variáveis aplicações, podendo trabalhar em sistemas de pesagens em geral, com média e alta precisão, são fabricadas em aço contra corrosão, com sensibilidade de 2mV/V e precisão de +/- 0,1% inoxidável (figura 12).

Figura 12: Célula de Carga MR-XX.
Fonte: [MICRO ANALISE].

2.8 MOTOR DE PASSO

Os Motores de Passo são dispositivos eletromecânicos que convertem pulsos elétricos em movimentos mecânicos que geram variações angulares discretas. O rotor ou eixo de um motor de passo é rotacionado em pequenos incrementos angulares, denominados “passos”, quando pulsos elétricos são aplicados em uma determinada sequencia nos terminais deste. A rotação de tais motores é diretamente relacionada aos impulsos elétricos que são recebidos, bem como a sequencia a qual tais pulsos são aplicados reflete diretamente na direção a

qual o motor gira. A velocidade que o rotor gira é dado pela frequência de pulsos recebidos e o tamanho do angulo rotacionado é diretamente relacionado com o numero de pulsos aplicados [BRITES; SANTOS, 2008].

Dividem-se em três tipos:

Relutância Variável: consiste em um rotor e um estator normalmente construído com material ferromagnético laminado, possuindo pólos salientes em ambos para que ocorra a magnetização. Os polos do estator possuem enrolamentos, já no rotor não há enrolamentos nem imã permanente. No estator os polos diametralmente opostos são energizados em série para produzir um campo magnético que passe por ambos gerando o movimento. Graças aos avanços tecnológicos, a redução de preço dos microcontroladores, e principalmente a sua simplicidade e robustez, os motores de relutância variável vem sendo muito utilizados englobando desde eletrodomésticos a equipamentos industriais [HENRIQUES, 2004].

Imã Permanente: para utilizar imãs permanentes no rotor dos motores, se faz necessário entender as propriedades magnéticas do material ferromagnético que constitui a estrutura do estator. Isso é devido à limitação da densidade de fluxo magnético no ferro, que não deve exceder aproximadamente 1,6-1,7 T [HENDERSHOT; MILLER, 1994], faixa onde ocorre rapidamente o decrescimento de permeabilidade magnética. Com passos típicos de 7,5 a 15 graus possuem uma baixa resolução e um baixo custo também. O rotor é construído com imas permanentes, porem, não possui dentes como no de relutância variável. É magnetizado radialmente e provem uma maior intensidade de fluxo magnético possuindo uma melhor característica de torque [BRITES; SANTOS, 2008].

Híbridos: vem da combinação das melhores características dos dois citados acima e por consequência é o melhor também, pois provem um melhor desempenho em relação à resolução de passo, torque e velocidade. Seu rotor é multidentado como no motor de relutância variável e contem uma ima permanente ao redor de seu eixo. Para que o rotor avance um passo é necessário que a polaridade magnética de um dente do estator se alinhe com a polaridade magnética oposta de um dente do rotor [SOUZA, 2011].

A forma de controle depende do que se deseja controlar e qual será a necessidade da aplicação, no caso o torque, a velocidade ou a precisão que são as características apresentadas por um motor de passo.

Passo Completo (Full-Step), uma bobina é energizada a cada passo, não possui grande torque e consome pouca energia, porém tem uma grande velocidade. Passo Completo 2 (Full-Step), onde duas bobinas são energizadas simultaneamente a cada passo dando a ele maior torque, porém, consumindo mais energia.

Motor de Meio Passo (Half-Step), onde ocorre a combinação dos dois primeiros citados. Ele consome mais energia que os anteriores, é muito mais preciso, seu torque é próximo ao do Motor de Passo Completo 2 onde as bobinas são energizadas simultaneamente, porem a velocidade é bem inferior a dos outros dois.

Sequência Para Controlar um Motor de Passo:

Nº do Passo	B3	B2	B1	B0
1	+	0	0	0
2	0	+	0	0
3	0	0	+	0
4	0	0	0	+

Figura 13: Passo Completo (Full-Step) motor unipolar.

Fonte: [BRITES; SANTOS, 2008].

Nº do Passo	B3	B2	B1	B0
1	+	0	0	0
2	+	+	0	0
3	0	+	0	0
4	0	+	+	0
5	0	0	+	0
6	0	0	+	+
7	0	0	0	+
8	+	0	0	+

Figura 14: Meio Passo (Half-Step) motor unipolar.

Fonte: [BRITES; SANTOS, 2008].

Nº do Passo	B3	B2	B1	B0
1	+	-	-	+
2	+	-	+	-
3	-	+	+	-
4	-	+	-	+

Figura 15: Passo completo (Full-Step) motor bipolar.

Fonte: [BRITES; SANTOS, 2008].

+ = Fluxo de Corrente Positiva
- = Fluxo de Corrente Negativa

Estados de funcionamento de um motor de passo:

Desligado: Não há alimentação no motor. Não existe consumo de energia, e todas as bobinas estão desligadas.

Na maioria dos circuitos este estado ocorre quando a fonte de alimentação é desligada.

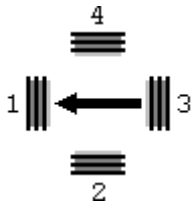


Figura 16: motor desligado.

Fonte: [BRITES; SANTOS, 2008].

Parado: Pelo menos uma das bobinas fica energizada e o motor permanece estático num determinado sentido. Nesse caso há consumo de energia, mas em compensação o motor mantém alinhado numa posição fixa.

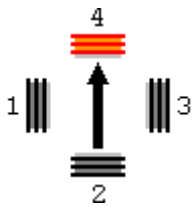


Figura 17: motor parado.

Fonte: [BRITES; SANTOS, 2008].

Rodando: As bobinas são eletrizadas em intervalos de tempos determinados, impulsionando o motor a girar numa direção.

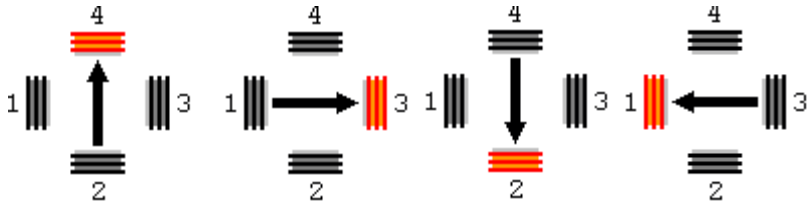


Figura 18: motor rodando.

Fonte: [BRITES; SANTOS, 2008].

3 DESENVOLVIMENTO DO SISTEMA

Neste capítulo serão apresentadas as atividades desenvolvidas na elaboração do sistema de controle do Multihead Weigher, descrevendo detalhadamente o sistema proposto, as experiências realizadas em laboratório e as dificuldades encontradas.

3.1 SISTEMA PROPOSTO

Inicialmente realizou-se o levantamento dos dados principais sobre os Multihead Weigher existentes no mercado, bem como suas funcionalidades e ranges de trabalho.

Para obter tais informações foi utilizado como referência o manual de um equipamento ao qual se tinha acesso, bem como consulta e pesquisa junto aos principais fabricantes.

Com base nas informações levantadas, optou-se por desenvolver o protótipo de um sistema distribuído conforme o diagrama de blocos abaixo:

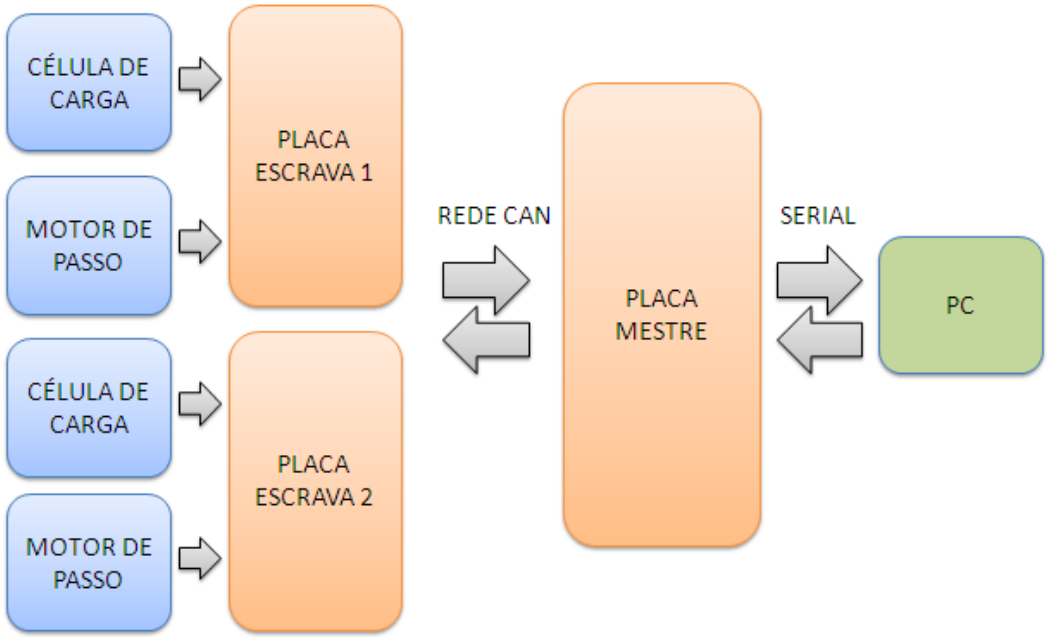


Figura 19: Diagrama processo.

Fonte: [Autoria própria].

As placas escravas do sistema são responsáveis cada qual por um cabeçote de dosagem, sendo cada cabeçote composto por uma célula de carga (pesagem do produto), um motor de passo (controle de dosagem), além de um sensor óptico para posicionamento do motor.

Para controle dos dados recolhidos da célula, comando do motor e comunicação de dados com a placa mestre é utilizado o microcontrolador PIC18F4580, desenvolvido pela Microchip. Para interface entre o microcontrolador e a rede física CAN é utilizado o componente MCP2551, que realiza o ajuste dos níveis de tensão necessários à comunicação.

Em conjunto com a célula de carga é utilizado um conversor A/D de 24bits, de forma a melhorar a resolução passível de ser obtida pelo sistema, sem afetar o desempenho do equipamento quanto a velocidade e precisão de dosagem. Para o projeto foi escolhido o conversor A/D da Analog Devices, o AD7731, que cumpre as especificações necessárias para o protótipo.

As placas escravas possuem também o circuito integrado L298N, que atua como drive de corrente, fornecendo os níveis de tensão/corrente necessários para o funcionamento do motor de passo.

O controle lógico do equipamento, definição de prioridades, controle do fluxo de dados na rede CAN e comunicação com o supervisório é realizado pela placa mestre. Ela é composta de um microcontrolador PIC18F4580, juntamente com os componentes de interface MAX232 (protocolo serial RS-232) e MCP2551(protocolo CAN).

Para visualização gráfica e interface com o operador da máquina foi desenvolvido um *software* de supervisão em Processing, uma linguagem baseada em Java, utilizada pelo ambiente de desenvolvimento de mesmo nome. A comunicação entre supervisório e placa mestre ocorre via porta serial RS-232 do PC.

As próximas seções detalham cada um dos módulos do projeto Desenvolvido.

3.2 MICROCONTROLADOR PIC18F4580

A família PIC18 uma das mais utilizadas composta por microcontroladores de 8 bits, possui várias subfamílias que se diferem pela quantidade de memória RAM, de memória EEPROM (que pode ser apagada através de luz ultra violeta), memória Flash (eletricamente apagada, de acesso para leitura rápida), numero de pinos (que pode ser 18,28,40,etc...), frequência máxima de *clock* e periféricos, que são dados fornecidos no *datasheet* de cada componente.

A escolha do microcontrolador PIC18F4580 deve-se ao fato deste possuir integrados todos os módulos de *hardware* necessários à comunicação CAN, USART e SPI, grande velocidade de processamento e capacidade de memória, além de apresentar um baixo custo e facilidade de programação.

O microcontrolador possui um módulo ECAN, que é completamente compatível com os protocolos CAN 2.0A e 2.0B, definidos pelas especificações da BOSCH.

As principais especificações desse microcontrolador são listadas a seguir:

- Velocidade de processamento 40mhz;
- Memória flash de programa de 32K bytes;
- 36 pinos configuráveis I/O (entradas e saídas);
- Módulo Master Synchronous Serial Port (MSSP), comunicação SPI e I2C;
- Módulo comunicação USART (RS-232, RS-485);
- Módulo ECAN, suportando taxas de comunicação de 1Mbps;
- Conversor A/D interno de 11 bits, com até 11 canais.
- Módulo CCP (Capture,Compare,PWM);

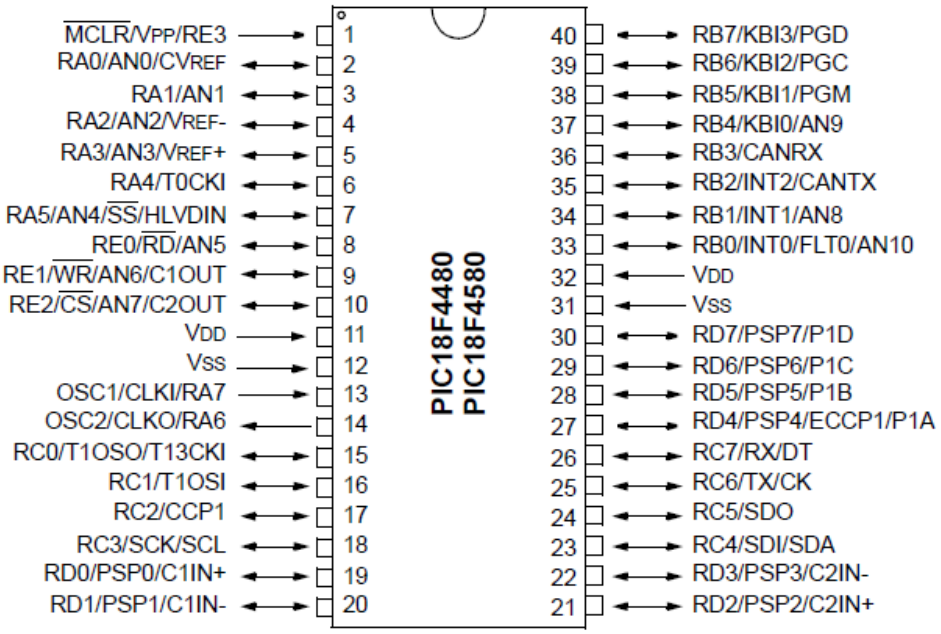


Figura 20: Diagrama de pinos do PIC18F4580.

Fonte: [MICROCHIP, 2009].

3.3 CONVERSOR ANALÓGICO/DIGITAL AD7731

O AD7731, componente da empresa Analog Devices, além de um conversor A/D de 24 bits com dois canais diferenciais de entrada, contém módulos analógicos para desenvolvimento de medidores de pressão e balança. Ele recebe sinais de baixa amplitude do transdutor, e os resultados da conversão analógica digital, são disponibilizados através de uma interface de saída serial. O conversor opera com uma fonte de alimentação de 5V, e trabalha com as escalas para entrada analógica unipolar 10mV, 20mV, 40mV e 80mV, ou para bipolar ±10mV, ±20mV, ±40mV e ±80mV.

A resolução pico a pico que pode ser obtida com o componente é de 1 em 230.000 divisões, dependendo da faixa de tensão de entrada que pode ser selecionada e da utilização ou não dos filtros internos, esta resolução pode cair para 30.000 divisões.

Na figura 21, o digrama funcional de blocos do componente:

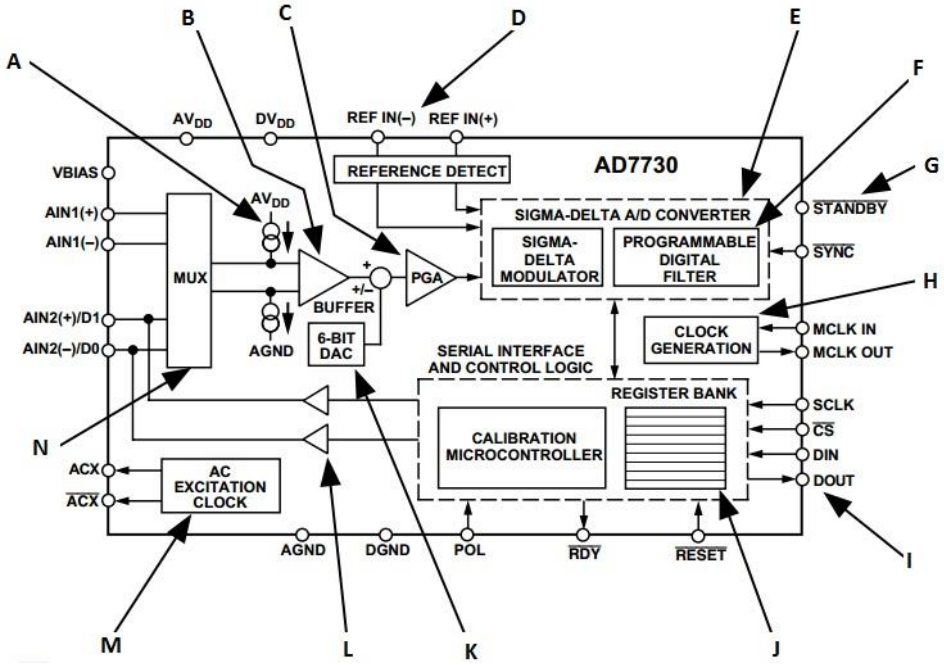


Figura 21: Diagrama de blocos do AD7731.
Fonte: [ANALOG DEVICES, 1998].

Breve explicação do diagrama funcional de blocos:

A) Fontes de corrente *Burnout*: duas fontes de corrente Burnout de 100nA permitem ao usuário detectar se o transdutor do canal selecionado está em curto circuito ou circuito aberto;

B) Amplificador *Buffer:* O amplificador *buffer* apresenta uma alta impedância para o estágio de entrada analógico;

C) Amplificador de Ganho Programável: O amplificador de ganho programável permite 4 faixas de entrada unipolares ou bipolares de 10mV a 80mV;

D) Referência Diferencial: A entrada de referência é diferencial, a tensão pode ser selecionada para ser nominalmente 2,5V ou 5V;

E) Conversor A/D Sigma-Delta: A arquitetura sigma-delta assegura 24 bits de resolução. A entrada do conversor sigma-delta pode ser chaveada;

F) Filtro Digital Programável: É um filtro de duplo estágio que permite programação de atualização da taxa de saída;

G) Modo de Espera: O modo de espera reduz o consumo de energia para 5m;

H) Circuito de Clock: A fonte de *clock* pode ser fornecida por uma fonte externa, conectados aos seus pinos de clock;

I) Interface Serial: Utiliza apenas três fios, todas as funções do componente, podem ser acessadas a partir desta interface;

J) Bando de Registradores: Treze registradores controlam todas as funções na peça, também fornecem o status do sistema e resultado da conversão;

K) Conversor D/A para Remoção do Offset: Permite gerar uma tensão programada que pode ser somada ou subtraída do sinal analógico de entrada antes que este seja aplicado ao PGA (Amplificador de Ganho Programável);

L) Drivers de Saída: O segundo canal de entrada analógico pode ser reconfigurado para funcionar como duas portas digitais de saída. A programação é feita através da interface serial;

M) Circuito de Excitação AC: A saída ACX fornece sinais usados para chavear a polaridade da tensão de excitação de circuito em ponte para aplicações utilizando excitação AC;

N) Multiplexador Analógico: Seleciona um dos dois canais diferenciais de entrada disponíveis para o *buffer.* É controlado através da interface serial [ANALOG DEVICES, 1998].

A comunicação entre o AD7731 e um dispositivo externo, microcontrolador ou placa dedicada a programação do componente é feita através do padrão SPI, sendo esta a forma de programação dos filtros de calibração e a saída dos dados por ele convertidas.

A interface serial do componente pode ser configurada com duas opções de calibração para operação com microcontroladores: calibração de zero e fundo de escala interna ou calibração de zero fundo de escala do sistema.

No processo de calibração de escala interna, os valores de tensão de referência são gerados para o ponto de zero e para o ponto de fundo de escala. Para o ponto de zero, o próprio componente curto-circuita a entrada analógica de sinal para zerar a tensão de referência de zero volt. Para o ponto de fundo de escala, o AD7731 utiliza uma tensão de referência gerada internamente.

Na calibração de zero e fundo de escala do sistema, executam-se os mesmos passos realizados no processo de calibração de zero e fundo de escala interna, porém, as tensões de referência (de zero e fundo de escala) são introduzidas pelo usuário. Durante o processo de calibração de zero e fundo de escala do sistema, as tensões aplicadas na entrada do canal selecionado devem permanecer estáveis para que sejam obtidos resultados satisfatórios.

Leitura de registradores on-chip do conversor pode ser na forma de uma única leitura ou leitura contínua. Uma única leitura de um registrador consiste de uma escrita para o registrado Communications (com RW1 = 0 e RW0 = 1), seguida pela leitura do registrador especificado. Para executar a leitura contínua de um registrador, deve-se escrever no registrador Communications (com RW1 = 1 e RW0 = 0) para colocar o dispositivo em modo de leitura contínua. O registrador especificado pode ser lido de forma contínua até que uma operação de escrita para o registrador Communications (com RW1 = 1 e RW0 = 1) ocorra, o que encerra o modo de leitura contínua. Quando estiver operando em modo de leitura contínua, o conversor está monitorando continuamente sua linha DIN. Portanto, a linha DIN deve ser permanentemente nível baixo (Vss = 0V) para permitir que a parte continue em modo de leitura contínua. A interface serial do AD7731 consiste em cinco sinais, CS, SCLK, DIN, DOUT e RDY. A linha DIN é usada para transferência de dados para os registradores on-chip, enquanto a linha DOUT é usada para acessar os dados a partir dos registradores. O SCLK é a entrada de clock serial do dispositivo e todas as transferências de dados (DIN ou DOUT) ocorrem com relação a este sinal.

A interface serial do componente pode ser configurada com duas opções de calibração para operação com microcontroladores: auto calibração de zero e fundo de escala ou calibração de zero fundo de escala do sistema.

No processo de calibração de escala, valores de tensão de referência são gerados para o ponto de zero e para o ponto de fundo de escala. Para o ponto de zero, o próprio componente curto-circuita a entrada analógica de sinal para zerar a tensão de referência de zero volt. Para o ponto de fundo de escala, o AD7731 utiliza uma tensão de referência gerada internamente.

Na calibração de zero e fundo de escala do sistema, executam-se os mesmos passos realizados no processo de auto calibração de zero e fundo de escala, porém, as tensões de referência (de zero e fundo de escala) são introduzidas pelo usuário. Durante o processo de calibração de zero e fundo de escala do sistema, as tensões aplicadas na entrada do canal selecionado devem permanecer estáveis para que sejam obtidos resultados satisfatórios.

3.3.1 Registrador Communications

Todas as operações para outros registradores são iniciadas através do registrador Communications. Ele controla se as operações subsequentes são de leitura ou escrita e também seleciona o registrador para qual a operação subseqüente ocorrerá. A maioria das operações subseqüentes devolve o controle ao registrador Communications exceto para a operação do modo de leitura contínua.

3.3.2 Registrador Status

Fornece informações de status sobre conversões, calibrações, modo de *standby* e condição da tensão de referência.

3.3.3 Registrador Data

Fornece o mais recente resultado da conversão do dispositivo. O tamanho do registrador pode ser programado para ser de 16 ou 24 bits.

3.3.4 Registrador Mode

Controla funções como modo de operação, operação unipolar/bipolar, controla a função de AIN3/D1 e AIN4/D0, corrente de *burnout* e o tamanho da palavra do registrador Data. Ele também contém o *bit* de seleção de referência, os bits de seleção de range e os bits de seleção de canal.

3.3.5 Registrador Filter

Controla a quantidade média de filtro no primeiro estágio, seleciona o modo *fast step* e modo *skip* e controla os modos de *chopping* no dispositivo.

3.3.6 Registrador Offset

Contém uma palavra de 24 bits, que é o coeficiente de calibração de *offset* para o conversor. O conteúdo deste registrador são utilizados para fornecer a

correcção de *offset* na saída do filtro digital. Há três registradores de *offset* e estes estão associados os pares de canais de entrada disponíveis.

3.3.7 Registrador Gain

Contém uma palavra de 24 bits, que é o coeficiente de calibração de ganho para o conversor. No conteúdo deste registrador são utilizados para fornecer a correcção de ganho na saída do filtro digital. Há três registradores de ganho e estes estão associados os pares de canais de entrada disponíveis.

3.4 CIRCUITO INTEGRADO MCP2551

O circuito integrado MCP2551 é um transceptor CAN de alta velocidade, responsável pela interface entre o controlador CAN e o barramento físico. Suporta taxas de transmissão de até 1Mbps, tem alta imunidade a ruídos devido a implementação diferencial e implementa as exigências de camada física da norma ISO-11898.

Cada nó do sistema possui um MCP2551 para converter os sinais digitais gerados pelo respectivo controlador CAN para sinais adequados para transmissão através do barramento (tensão diferencial de saída). Ele também atua como uma barreira entre o controlador CAN e picos de alta tensão que podem ser gerados no barramento CAN por fontes externas (EMI, ESD, transientes elétricos, etc.).

3.5 CIRCUITO INTEGRADO MAX232

O MAX232 é um CI, desenvolvido pela Maxim Integrated Products, que realiza a conversão de nível de sinais de uma porta serial RS-232 para níveis adequados para uso em dispositivos TTL. Ele é um transmissor/receptor duplo que converte entradas RS-232 para níveis de 5V TTL/CMOS(receptor), e entradas 5V TTL/CMOS para níveis RS-232(transmissor).

Esse circuito integrado converte os níveis dos sinais RX, TX, CTX e RTS, e dispensa o uso de uma fonte externa de +12Vdc, sendo alimentado por uma única tensão +5Vdc.

3.6 DRIVE DE ALTA TENSÃO L298N

O drive de alta tensão e alta corrente L298N é utilizado para fornecer a relação tensão/corrente necessária ao acionamento do motor de passo. Este CI pode operar com até 46V de alimentação e conduzir até 2A(por canal) para a carga. È projetado para aceitar níveis lógicos padrão TTL e acionar cargas indutivas como relés, solenoides, motores CC e motores de passo.

4 TESTES EM BANCADA

De forma a validar a *performance* do sistema e sua viabilidade foram realizados testes em bancada, utilizando de *protoboard* para a montagem dos circuitos necessários.

Os testes foram divididos em quatro etapas distintas e posteriormente integrados, realizando-se uma simulação do funcionamento do sistema de controle. As etapas foram:

- Protótipo de controle de motor de passo;
- Desenvolvimento de software supervisório e comunicação serial com placa mestre;
- Protótipo de comunicação CAN mestre-escravos;
- Protótipo do módulo de leitura da célula de carga;

A programação do microcontrolador foi realizada em linguagem *Assembly* inicialmente (controle de motor de passo, comunicação com supervisório e leitura de célula de carga), porém devido as dificuldades encontradas na implantação do protocola CAN o software foi migrado para linguagem C, utilizando-se do programa MikroC PRO for PIC. As vantagens da utilização desse programa foram principalmente o vasto número de bibliotecas encontradas (CAN, SPI, USART), além de amplo material de referência na internet, facilitando a programação em linguagem de alto nível.

A seguir são detalhados o *hardware* e *software* dos testes realizados, bem como os resultados obtidos.

4.1 CONTROLE DE MOTOR DE PASSO

O teste de controle de motor de passo foi realizado utilizando-se de um cabeçote de Multihead Weigher ao qual se tinha acesso, visto na figura 22.

Figura 22: Cabeçote Multihead Weigher.
Fonte: [Autoria própria].

O cabeçote utilizado possui dois motores de passo, sendo um para controle de abertura/fechamento da cabeça de alimentação das balanças e outro para o controle da cabeça de dosagem das balanças na calha de descarga. O sistema mecânico de atuação é semelhante nos dois funis, sendo ele composto de um acoplamento com formato de meio espiral e um braço mecânico, que reage de forma proporcional a rotação do acoplamento, como pode ser visto em detalhe na figura 23 abaixo.

Figura 23: Sistema mecânico de atuação.
Fonte: [Autoria própria].

A proporção de abertura é dada pela rotação do motor de passo, portanto é necessário o controle do número de passo dados, relacionado ao ângulo de rotação da espiral de acoplamento. Para o referenciamento de ponto zero do motor, o acoplamento é dotado de um sensor óptico, que realiza a detecção de um pequeno recorte no acoplamento, indicando o ponto de parada do motor.

Figura 24: Referenciamento de ponto zero do motor.
Fonte: [Autoria própria].

O motor de passo utilizado é do tipo bipolar à 4 fios, com passo de 1,8º, tensão de alimentação de 12V, consumindo 0,33A por fase aproximadamente. Não foi possível obter maiores dados (fabricante, torque), devido a placa de identificação do motor estar danificada.

O hardware montado pode ser visualizado na figura abaixo:

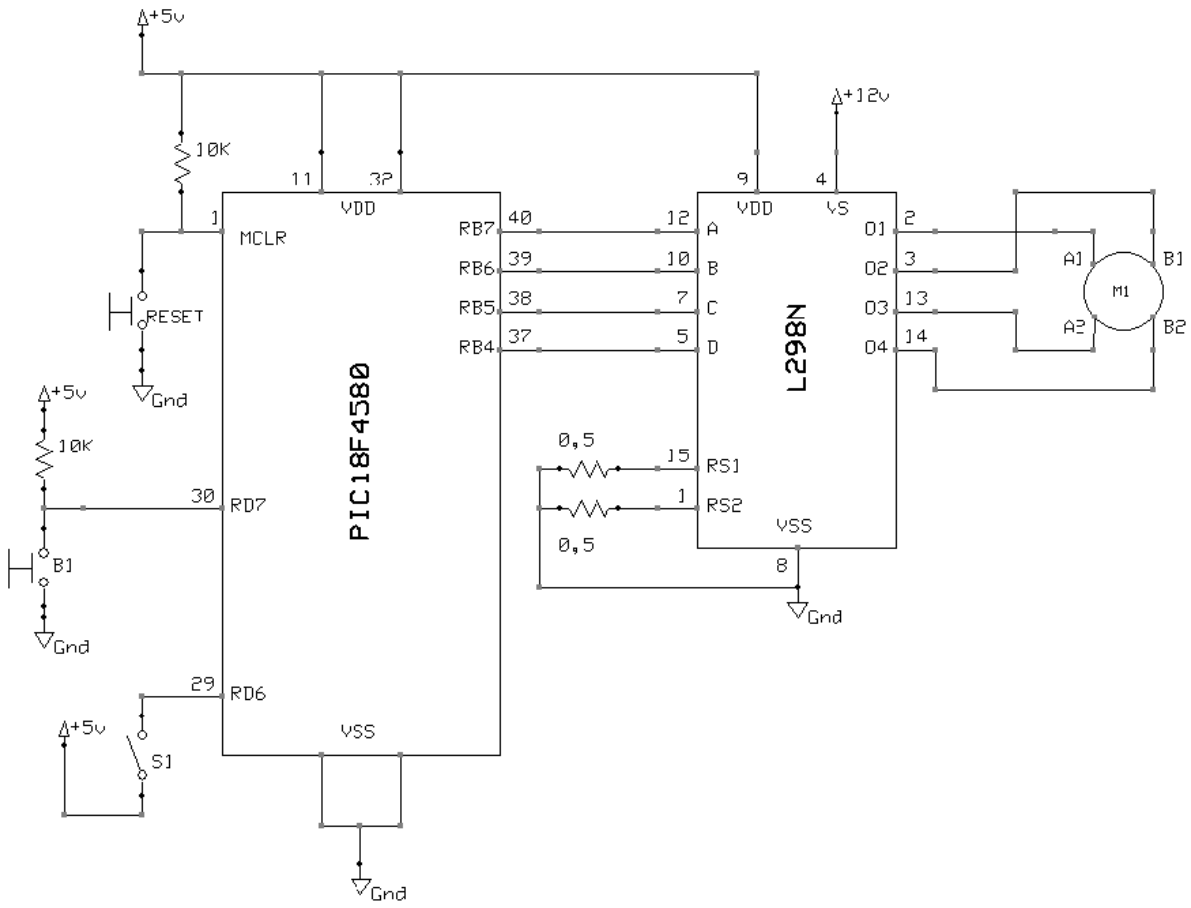


Figura 25: Diagrama elétrico controle do motor de passo.
Fonte: [Autoria própria].

O microcontrolador teve sua PORTB configurada como saída, sendo os pinos RB4 à RB7 utilizados como sinal de comando nas entradas do L298N. Como sinal de entrada temos o sensor óptico de referenciamento (S1) e um botão não-retentivo (B1), que foram ligados respectivamente aos pino 6 e 7 da PORTD (RD6/RD7), que foram configuradas como entrada no microcontrolador.

Toda vez que um pulso é dado nas entradas do L298N, a saída de mesmo número atua, sendo alimentada com a tensão de alimentação (Vs = 12V), ocorrendo a alimentação da bobina do motor de passo ligada ao pino.

O software programado no microcontrolador monitora o estado do botão B1 ciclicamente, assim que um pulso do botão é sentido ele executa a rotina de comando do motor de passo. A rotina utilizada leva em conta o modelo de acionamento de meio-passo, acionando portando as saídas (RB4 à RB7) sequencialmente a partir do modelo definido.

Os testes realizados demonstraram que o ângulo de rotação necessário para uma abertura total do sistema mecânico era de aproximadamente 76º. Portanto o número de meio- passos (1,8º/2 = 0,9º) necessário era de 84 ciclos aproximadamente. O programa elaborado realiza a contagem dos ciclos de passo realizados e assim que a contagem estoura, o sentido de giro do motor é invertido, até que o sinal do sensor óptico atue na entrada do microcontrolador.

Os resultados obtidos foram satisfatórios, sendo a abertura do sistema mecânico realizada com velocidade e precisão.

4.2 SISTEMA SUPERVISÓRIO

4.2.1 *Processing*

Para o desenvolvimento do *software* supervisório foram buscados referências de *softwares* gratuitos, que oferecessem as funcionalidades desejadas, aliadas a boa capacidade gráfica. Acabou optando-se pelo *software Processing*, desenvolvido em 2001 pelos estudantes do MIT Media Lab, Ben Fry e Casey Reas.

Inicialmente desenvolvido como um *software sketchbook* e para o ensino dos fundamentos de programação de computadores dentro de um contexto visual, o *Processing* evoluiu para uma ferramenta de desenvolvimento profissional. Utilizando uma linguagem baseada em Java, de nome *Processing*, ele alia a capacidade gráfica do Java à simplificação dos comandos de programação [PROCESSING,2013].

Sua capacidade gráfica e ambiente amigável de programação serviram como motivação para o desenvolvimento de um software supervisório para controle do nosso sistema de *Multihead Weigher*.

4.2.2 O *software* desenvolvido

O *software* tem a função de oferecer ao usuário uma *interface* de controle e monitoração amigável e enviar o comando efetuado pelo usuário para a porta serial do microcomputador, que por sua vez será enviado ao microcontrolador.

4.2.3 Tela Inicial

Figura 26: Tela Inicial do software desenvolvido.
Fonte: [Autoria própria].

A tela inicial do supervisório oferece os principais comandos, e visualização dos principais parâmetros ajustados para o processo. Temos o tipo de produto sendo envasado, código de produto, peso alvo, número de dosagens por pacote, peso mínimo e máximo tolerado e o contagem de produção. Todos esses parâmetros são alteráveis pelo operador na tela “Parâmetros de Produção”.

Na parte superior da tela temos os botões de Start e Stop de produção, juntamente com os botões de acessos as demais telas do supervisório. As telas de Comandos e Calibração só são acessíveis caso a máquina encontre-se em modo de Stop, sendo a que a para acesso a tela de Calibração é necessário um login a nível de técnico (Usuário e Senha).

4.2.4 Tela *Login*

Figura 27: Tela de login do software desenvolvido.
Fonte: [Autoria própria].

Na tela de *login* é possível alterar o nível de acesso ao supervisório, alterando entre os usuários programados. Inicialmente foram definidos 2 tipos de usuários: Operador e Engenharia. O primeiro possui acesso às telas de parâmetros e comandos e o segundo acesso às telas de calibração.

Caso seja inserido senha ou usuário incorreto, o sistema exibe uma mensagem de alerta ao operador.

4.2.5 Tela de calibração

A tela de calibração exibe uma representação gráfica de todas as cabeças de pesagem, juntamente com o peso atual presente em cada balança. O usuário tem a opção de realizar a calibração de cada balança individualmente, selecionando as desejadas através do botão de seleção, logo acima do peso da balança. Quando definido as balanças desejadas basta clicar sobre o botão calibra, que o comando é enviado a placa mestre, que executa a lógica de calibração dos conversores A/D presentes nas placas escravas.

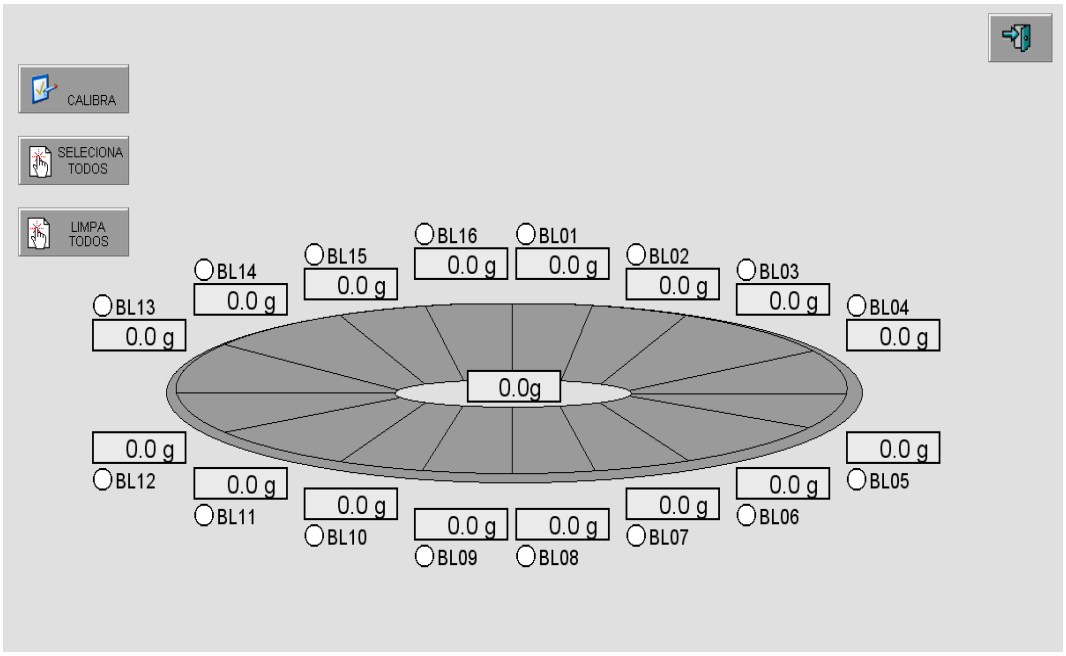


Figura 28: Tela de calibração do software desenvolvido.

Fonte: [Autoria própria].

4.2.6 Tela de comandos

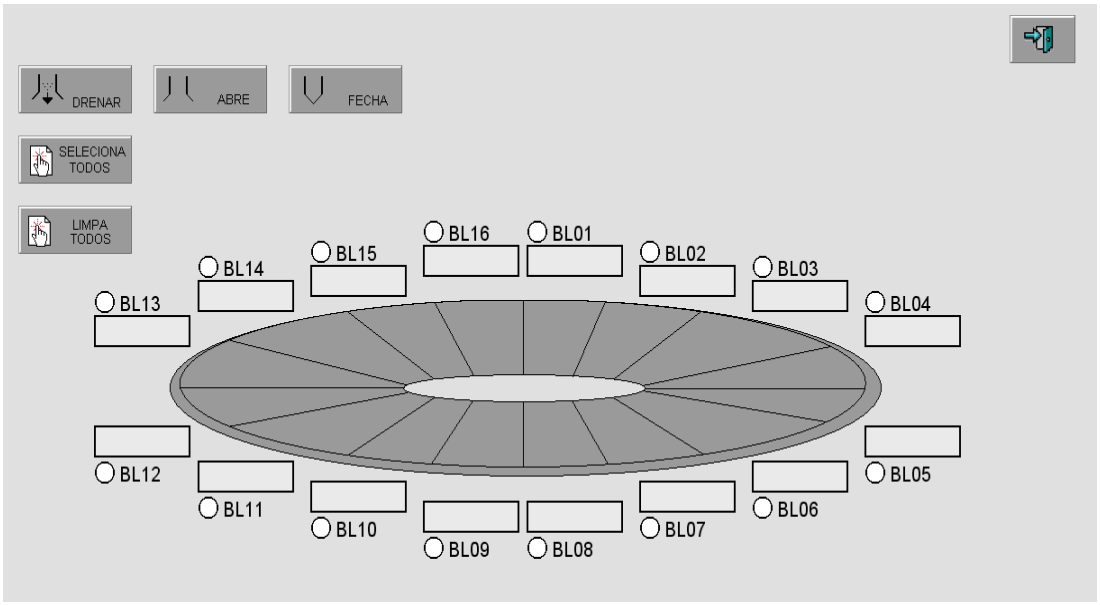


Figura 29: Tela de comandos do software desenvolvido.

Fonte: [Autoria própria].

A tela de comandos permite ao usuário o comando manual dos dispositivos eletromecânicos do equipamento. Semelhante a tela de calibração, ela mostra uma representação gráfica das cabeças de pesagem, sendo possível selecionar a cabeça

desejada para controle. Nesse protótipo foram definidos os comando apenas dos motores de passo das cabeças de pesagem, sendo configurado os comandos de abertura, fechamento e de drenagem das balanças (abertura e fechamento temporizados).

4.2.7 Tela de parâmetros

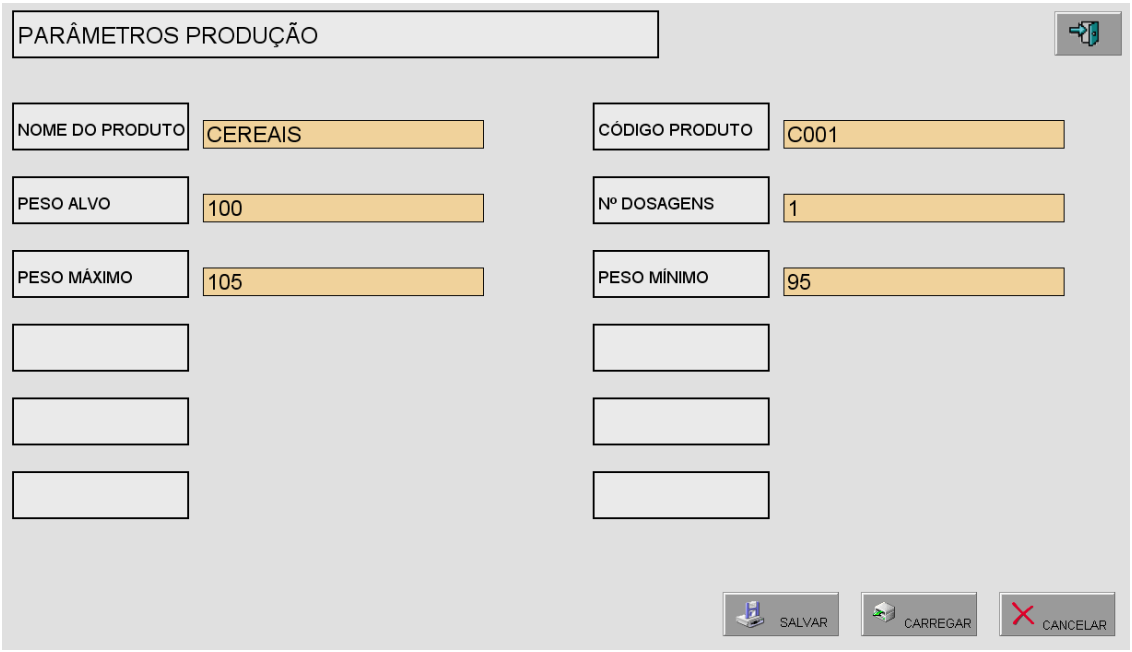


Figura 30: Tela de parâmetros do software desenvolvido.
Fonte: [Autoria própria].

Na tela de parâmetros é possível a alteração dos parâmetros de produção do equipamento, bem como o carregamento e criação de receitas pré-definidas. A alteração só é executada pela máquina quando clica-se no ícone “SALVAR”, caso seja necessário, é possível cancelar as alterações clicando no ícone “CANCELAR”. Assim que o comando de salvar é executado, o supervisório envia através da serial os novos dados para o microcontrolador mestre, que realiza o controle lógico de operação baseado nos novos parâmetros.

Um banco de dados para salvamento de receitas de produção foi criado, e é acessado através do botão carregar, como visto na figura 31.

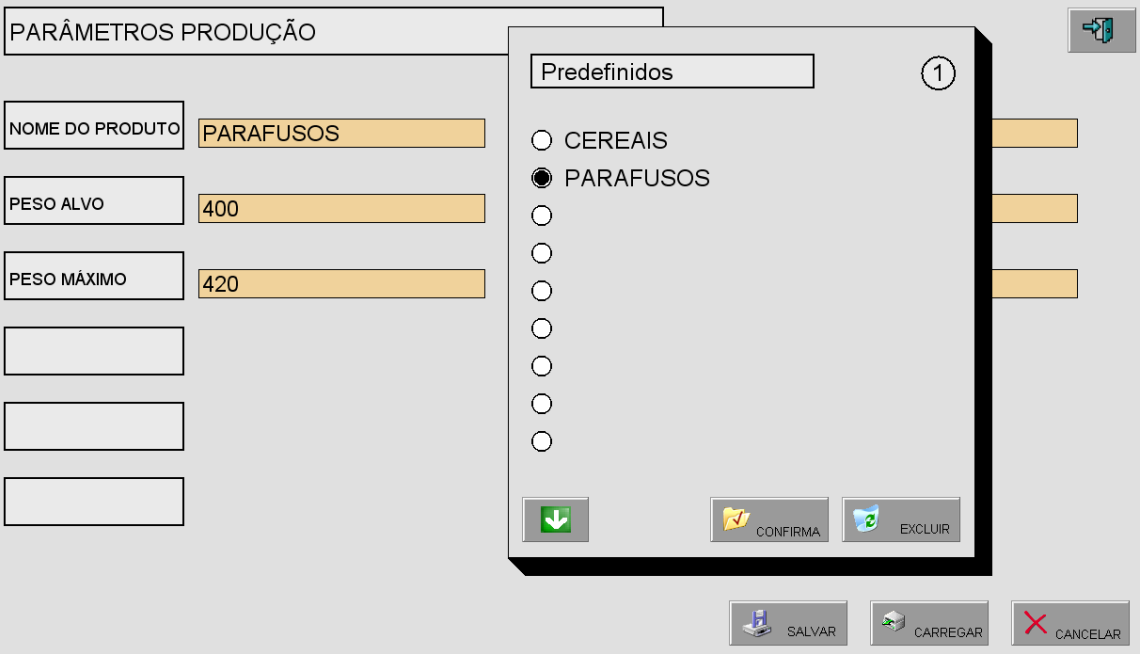


Figura 31: Tela de predefinidos do software desenvolvido.

Fonte: [Autoria própria].

Para selecionar uma das receitas já existente basta selecionar o ícone correspondente a receita desejada e clicar em “CONFIRMAR”. É possível excluir a receita, clicando no botão “EXCLUIR”. Caso se deseje criar uma nova receita, basta seleciona um dos campos vazios, clicar em confirmar e inserir os parâmetros desejados nos campos correspondentes. O banco de dados criado aceita inicialmente até 20 receitas pré-definidas, alteráveis pelo usuário.

Este *software* supervisório foi desenvolvido com o intuito de demonstrar as funcionalidades necessárias ao equipamento, bem como realizar um teste de processamento pelo computador da geração gráfica e da comunicação mestre-supervisório.

Para realização dos testes em *protoboard* o microcontrolador mestre foi programado para enviar um dado correspondente a cada comando, sinalizando a correta transmissão de dados, pois devidos a problemas de prazos no projeto, a lógica de tomada de decisões a partir dos dados recebidos não havia sido programada.

A comunicação serial foi definida para 9600bps, não sendo constatado interferências, ruídos ou perda de dados durante os testes realizados.

Abaixo temos o diagrama elétrico de conexão entre mestre-PC.

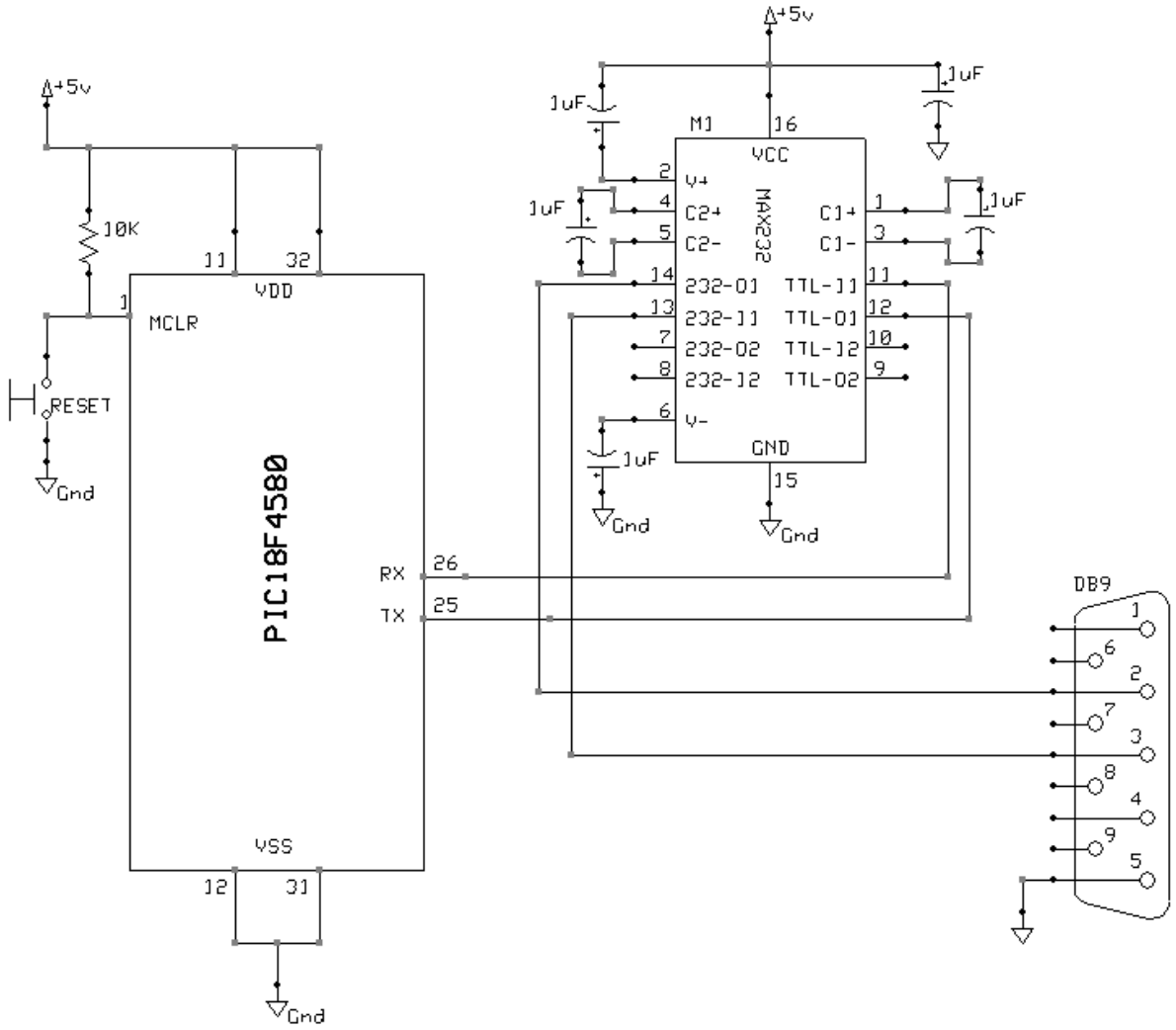


Figura 32: Diagrama elétrico comunicação serial.

Fonte: [Autoria própria].

4.2.8 Protótipo de comunicação CAN

Os teste para implementação de protocolo CAN foram realizados em *protoboard,* promovendo a conexão entre PC-mestre (RS-232) mestre-escravo (CAN), segundo o hardware no esquema figura 33.

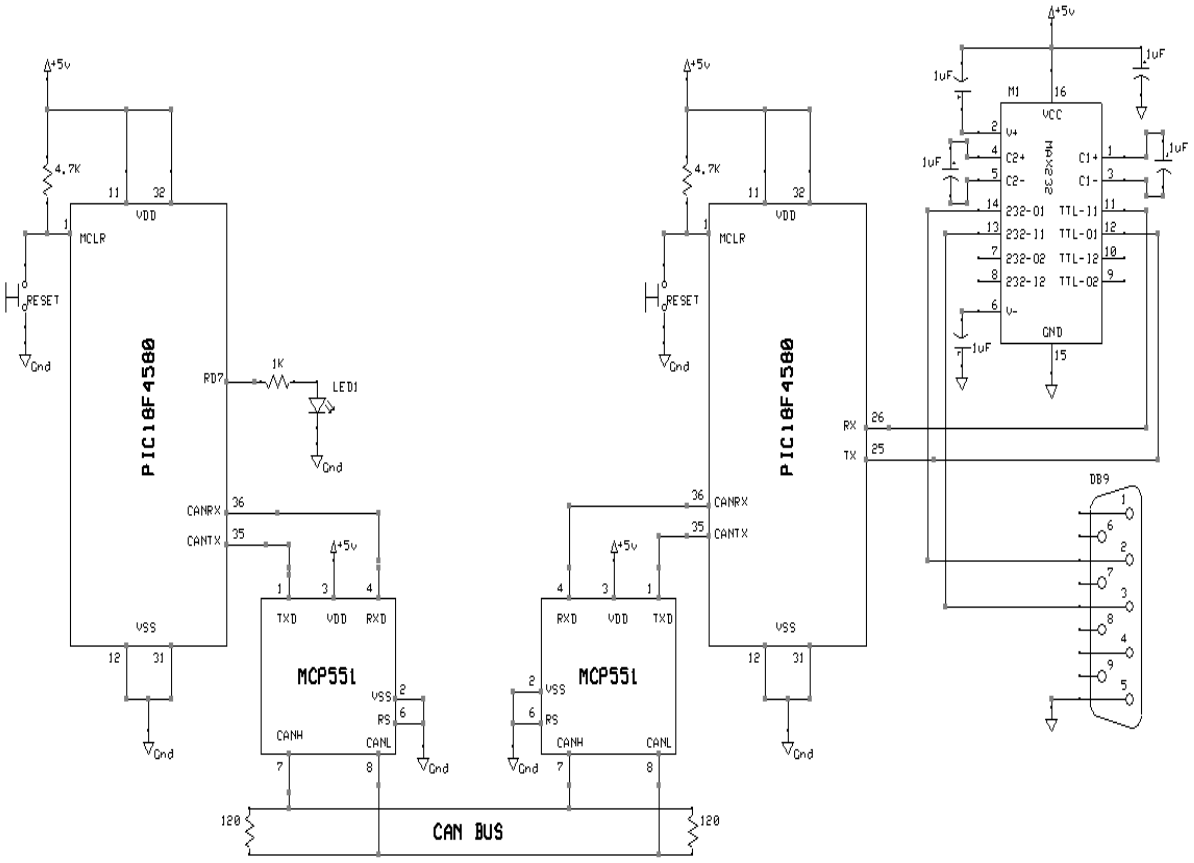


Figura 33: Diagrama elétrico comunicação rede CAN.
Fonte: [Autoria própria].

O barramento CAN tem em sua terminação dois resistores de 120Ohm, um em cada extremidade, conforme a norma ISO-11898. A conexão entre PC e placa mestre é feita utilizando conectores DB9, ligados ao componente MAX232 para interface. Para esse teste foi desenvolvido um software de supervisão em Processing simplificado, responsável por mostrar o dado recebido do mestre e enviar um comando para ser repassado ao dispositivo escravo.

A interface com o barramento CAN é feita em cada dispositivo da rede utilizando o MCP2551 para adequação de níveis de tensão. No pino RD7 do dispositivo escravo temos um LED (light emission diode), conectado de forma a sinalizar que o comando definido foi recebido.

Com um frequência de *clock* de 8Mhz no microcontrolador e velocidade da rede CAN em 100Kb/s, foram definidas as seguintes especificações da rede :

- Time quanta (Tq) = 500nS;
- Nominal Bit Time = 20Tq;
- Propagation Delay (Prop_Seg) = 6 Tq;

- Phase Segment 1 (Phase_Seg1) = 6Tq;
- Phase Segment 2 (Phase_Seg2) = 7Tq;
- Syncro Jump Width (SJW) = 1Tq;
- Sample Point : 65% do tempo de bit;
- Mascaramento desabilitado;
- ID filtro de mestre : 1
- ID filtro de escravo : 20

Para as especificações de parâmetros de tempo da rede CAN foi utilizado o software CAN Bit Timing Calculator, fornecido pela Microchip.

No circuito montado o mestre envia por rede CAN, uma mensagem contendo o ID de filtro do dispositivo escravo e um dado “A”, como forma de solicitar uma resposta do dispositivo escravo. Assim que recebe a mensagem de resposta o mestre envia através da porta Serial o dado para o software de supervisão, que o exibe na tela do PC. Quando o mestre recebe o comando do supervisório, ele envia na rede CAN o ID de filtro do dispositivo escravo e um dado “B”.

O PIC escravo quando recebe um dado via CAN, faz uma comparação: caso seja o valor do dado seja “A” ele envia como resposta o valor “23” na CAN. Caso o dado recebido seja “B”, ele alimenta a saída RD7, ligando o LED conectado ao pino.

O protótipo montado em *protoboard* permitiu a avaliação do funcionamento da rede CAN, sendo possível analisarmos a velocidade do tráfego de dados e a imunidade à ruídos da rede.

4.2.9 Protótipo leitura célula de carga

Os testes iniciais com a célula de carga foram realizados utilizando-se da cabeça de pesagem ao qual se tinha acesso figura 22. Por ser parte integrante de um Multihead Weigher operante oferecia a vantagem do sistema de balança pronto, sendo necessária a leitura do sinal da célula de carga. Consultando o manual do equipamento foi verificado que a célula de carga era especificado para até 2Kg, fornecendo 2mV/V na sua saída. Com base nas consultas realizadas ao fabricante foi verificado que os equipamentos existentes no mercado ofereciam uma alta resolução para esse range de célula de carga, sendo em geral de 0,1g.

Com um range de 2000g e resolução de 0,1g, era necessário um conversor de 20000 divisões (2000g/0,1g) no mínimo. O PIC18F4580 oferece um conversor

A/D de 11 bits máximos de resolução, ou aproximadamente 2048 divisões (2e11), mostrando-se insuficiente para a aplicação desejada.

Foi optado então pela utilização de um conversor A/D externo, no caso o AD7731, que possui resolução de 24bits, sendo a resolução de pico a pico, livre de ruídos, de até 230.000 divisões(aproximadamente 18bits), além de apresentar funcionalidades como calibração automática do sistema e gerar em seus registradores de saída o dado já tratado, não sendo necessário qualquer circuito de filtragem adicional.

O hardware utilizado para testes pode ser visto na figura 34 abaixo.

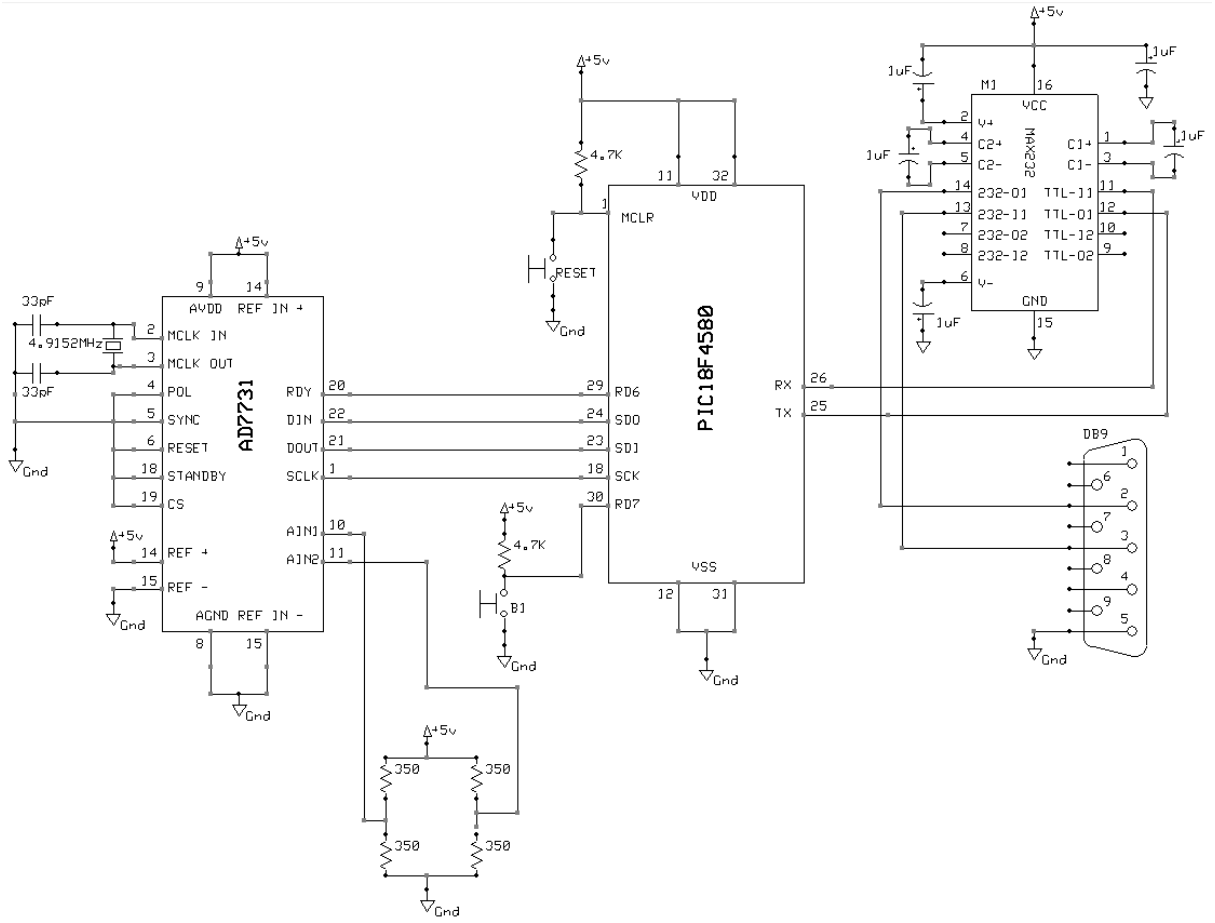


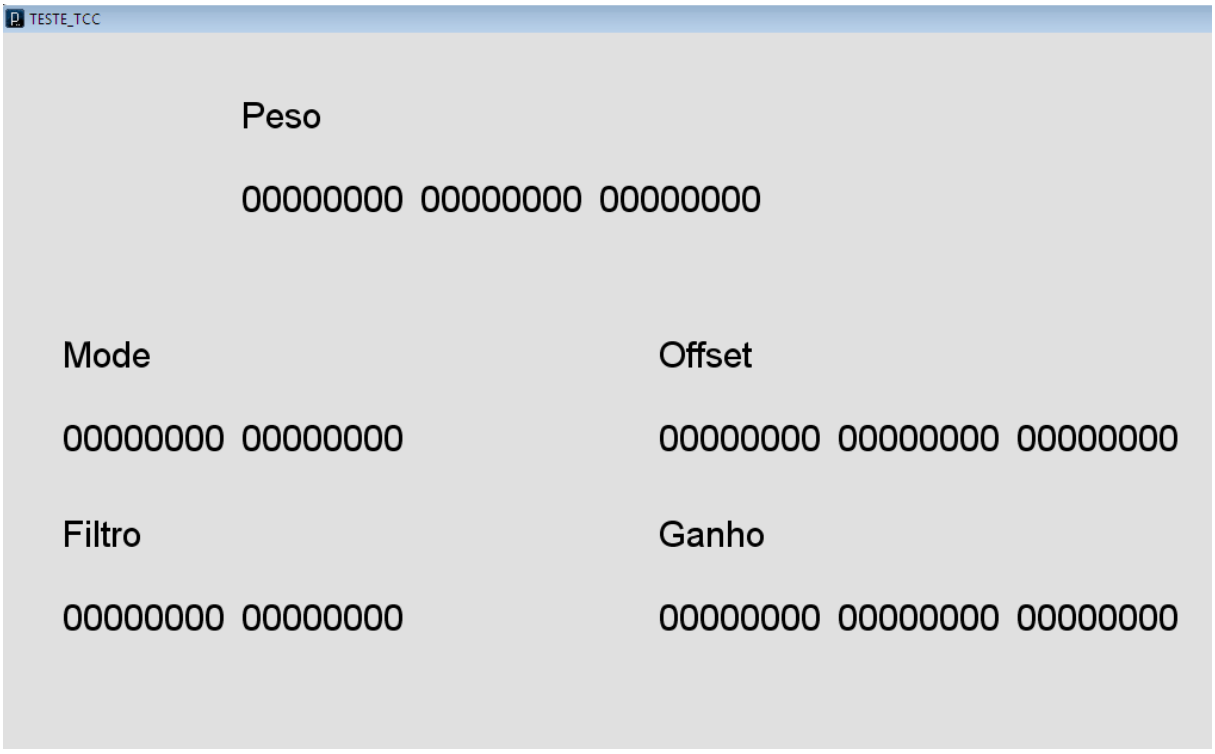
Figura 34: Diagrama elétrico conversor A/D.

Fonte: [Autoria própria].

Como podemos ver no diagrama o microcontrolador é interligado ao conversor analógico digital utilizando a porta de comunicação SPI. Para estabelecer a comunicação com o conversor foram definidos as seguintes configurações no microcontrolador:

- *Clock* oscilador do microcontrolador = 8Mhz;
- *Clock* oscilador SPI = 125Khz (Oscilador/64);
- Amostragem de dado no meio período (SMP *bit* = 0);
- Polaridade de *clock* inativo = 0V(CKP *bit* = 0);
- Transmissão ocorre na transição de estado inativo para ativo (CKE *bit* = 0);

O programa desenvolvido para o microcontrolador envia sequencialmente os dados programados, via SPI, para o conversor A/D, que escreve os valores recebidos em seus registradores internos. Foi elaborada também na rotina a leitura de determinados registradores antes e depois das configurações, de formar a validar o estado inicial de *reset* do conversor e o correto recebimento de dados. Para essa verificação do correto funcionamento da comunicação SPI, e da configuração do conversor, foi desenvolvido uma tela de supervisão em Processing, figura 35, que



permitia a visualização dos valores nos registradores principais do conversor A/D.

Figura 35: Supervisório de teste de comunicação conversor A/D.

Fonte: [Autoria própria].

O *software* desenvolvido para o microcontrolador aguarda inicialmente o pressionamento do botão ligado ao pino RD7. Assim que o botão é pressionado ele

executa uma leitura do valor do registrador *mode* e do registrador filtro do conversor, e envia o valor recebido para visualização na IHM. Pressionado o botão novamente ele realiza a escrita do valor definido para o registrador filtro e realiza logo em seguida uma leitura do valor que acabou de ser escrito e envia para o supervisório. Um novo pulso no botão envia o comando de calibração interna de fundo de escala para o conversor A/D, o controlador aguarda então um pulso no pino RD6, que é o sinal RDY do conversor, sinalizando o fim da calibração. Após um atraso de tempo ele executa o mesmo processo para o comando de calibração interna de zero.

Realizadas as calibrações ele executa a leitura dos registradores de *offset*, ganho e filtro e envia para visualização no supervisório, sendo possível assim avaliar se a comunicação ocorreu sem perdas de dados e se é possível iniciar o processo de leitura dos dados do conversor A/D ou se é necessário um *reset* no sistema e realizar o processo de configuração novamente.

Um novo pulso no botão faz com que o microcontrolador envie os comandos para o conversor A/D, colocando em modo de conversão contínua e de leitura contínua, entrando em seguida num loop de programa para leitura dos dados e envio serial para o supervisório.

4.3 DIFICULDADES ENCONTRADAS

Os testes realizados em *protoboard* para comunicação com o conversor A/D não apresentaram resultados satisfatórios. Problemas de ruído na *protoboard* ocasionavam perda de dados durante a transmissão, comprometendo todo o sistema.

Os primeiros testes realizados ocorreram configurando apenas alguns registradores do conversor A/D, ocorrendo sem maiores problemas. O problema surgiu quando a configuração final e o processo de leitura dos dados dos registradores foram feitos. Após um número variável de envios na SPI, o conversor retornava valores diferentes do esperado, como sequências de bits 1 ou 0, não sendo possível recomeçar a comunicação do ponto de erro, tendo como única solução um reset no microcontrolador e conversor.

Como já citado o problema se mostrava intermitente, em algumas poucas ocasiões o processo de configuração ocorreu sem problemas, porém após alguns

minutos de leitura contínua da célula de carga o conversor retornava sequências de *bits* 1, evidenciando uma perda de comunicação.

Diversas soluções foram tentadas sem sucesso. Alterações de *clock*, velocidade de transmissão, estado de polaridade, borda de envio e recebimento de dados foram testados e não forneceram bons resultados. Altas taxas de transmissão evidenciavam ainda mais o problema, enquanto que alterações na borda de envio e recebimento SPI ocasionavam a perda total de comunicação.

As mesmas configurações de SPI foram utilizadas para uma comunicação de teste entre dois microcontroladores. Neste teste realizado ela ocorreu sem perda de dados ou sinais de interferência.

Acredita-se que o conversor A/D tenha uma alta sensibilidade à ruídos no seu módulo interno de comunicação, tornando inviável os testes em *protoboard*.

5 CONCLUSÃO

O desenvolvimento deste projeto possibilitou a visualização das inúmeras aplicações ao qual o uso de *hardware* microcontrolado aparece como uma alternativa viável, de forma a reduzir os custos na aplicação de automação em processos ou equipamentos.

A partir dos protótipos montados, foi possível verificar a robustez da rede CAN como opção de sistema distribuído de controle de máquinas, mostrando-se compatível com exigências que o ambiente industrial exige, tanto em nível de velocidade de processamento de dados, como no controle do tráfego de informações entre os dispositivos.

O *software* Processing, surge como opção de ferramenta de desenvolvimento de *interfaces* homem-máquina. Apesar de não ser um programa dedicado a essa aplicação, sua capacidade gráfica, liberdade de estruturação de programa e interconexão com diversos recursos comuns à maioria dos PC´s, apresenta um atrativo para o desenvolvimento de projetos de *software* supervisório.

Igualmente, podemos notar também as dificuldades e desafios que surgiram durante o desenvolvimento de um *hardware* microcontrolado. Desde a busca por bibliografias especializadas, até a elaboração de um sistema robusto para aplicações industriais. Os níveis de ruído e interferências externas notado nos testes em *protoboard* demonstram que para se trabalhar com leitura de dados analógicos através de conversores externos, o *hardware* deve ser visto como um ponto crucial ao processo, de forma a amenizar as interferências externas durante a leitura do sinal analógico e a comunicação dos dados.

Apesar do projeto não ter alcançado o resultado esperado, espera-se que este trabalho sirva como referencia e motive outros acadêmicos e profissionais da área a dar continuidade ao projeto, explorando alternativas aos problemas enfrentados, visto o potencial que o desenvolvimento do sistema proposto oferece.

REFERÊNCIAS

[MICROCHIP, 2009] MICROCHIP. **PIC18F2480/2580/4480/4580.** Datasheet, 2009. Disponível em: *www.microchip.com.* Acessado em: 23 de dezembro de 2013.

[YAMATO, 2014] YAMATO Scale Dataweigh (UK) Ltd. Disponível em: *www.yamatoscale.co.uk*. Acessado em: 23 de dezembro de 2013.

[ROCKWELL, 1997] ROCKWELL. **DeviceNet Product overview.** Publication DN-2.5, 1997.

[BORGES; PAIVA; PIEDADE, 2008] BORGES, J.D. ; PAIVA,L. N. ; PIEDADE, T. S. S.. **Sistema de posicionamento microcontrolado utilizando comunicão serial.** UNIS-MG, Varginha 2008.

[AZAVEDO; BATISTA; VARELA] AZAVEDO, J. V. C.; BATISTA, I. J. L.; VARELA, A. T.;**Implementação de uma rede CAN aplicada a Robôe Móveis.** Instituto Federal de Educação, Ciência e Tecnologia do Ceará.

[BRAGA, 2010] BRAGA, Newton C. **Como funcionam os Conversores A/D.** Instituto Newton C. Braga, 2010. Disponível em: *www.newtoncbraga.com.br*. Acessado em: 04 de janeiro de 2014.

[GUIMARÃES; SARAIVA, 2002] GUIMARÃES, Alexandre de Almeida; SARAIVA, Antônio Mauro.**O Protocolo CAN: Entendendo e Implementando uma Rede de Comunicação Serial de Dados baseada no Barramento "Controller Area Network".** Universidade Estadual de São Paulo, 2002.

[ANALOG DEVICES, 1998] ANALOG DEVICES. **Bridge Transducer ADC (AD7731).** Datasheet, 1998. Disponível em: *www.analog.com*. Acessado em: 04 de janeiro de 2014.

[MOREIRA, 2005] MOREIRA, Odair. **Estudo Sobre Sistemas de Pesagem.** Universidade de São Judas Tadeu – Monografia Apresentada a Pós-Graduação, São Paulo, 2005.

[SOUZA; LAVINIA, 2005] SOUZA, David José de; LAVINIA, Nicolas César. **PIC16f877A. Conectando o PIC – Recursos Avançados.** 3 Ed. São Paulo,Editora Érica, 2006.

[ANDOLFATO; CAMACHO; BRITO, 2004] ANDOLFATO, R. P.; CAMACHO, J. S.; BRITO, G. A. de. **Extensômetria Básica.** UNESP, Ilha Solteira, 2004.

[MICRO ANÁLISE] MICRO ANÁLISE. **Catálago – Célula de Carga MR-XX Aplicação Geral.** Disponível em: *www.microanalise.com.br/catalogo/cat11.html*. Acessado em: 10 de janeiro de 2014.

[ADRIANO; MARÇANO, 2009] ADRIANO, J. D.; MARÇANO, R. de O. **XM118 – Microncontroladores PIC18.** Exsto Tecnologia Lyda. Santa Rita do Sapucaí - MG, 2009.

[SANCHO, 2009] SANCHO, Rui Jorge Cunha. **Sistema de controlo Distribuído Baseado em Barramento CAN.** Universidade de Aveiro. Aveiro - Portugal, 2009.

[PEREIRA, 2007] PEREIRA, Fábio. **Microcontroladores PIC – Programação em C**. 7 Ed. São Paulo, Editora Érica, 2007.

[PEREIRA, 2005] PEREIRA, Fábio. **Microcontroladores PIC – Técnicas Avançadas**. 6 Ed. São Paulo, Editora Érica, 2005.

[HUBERT, 2001] HUBERT, Marco Kasdorf. **O Protocolo CAN como solução para aplicações distribuídas, baseadas em objetos entre PCs e microcontroladores**. Universidade Federal de Pelotas. Pelotas – RS, 2001.

[SOUZA, 2002] SOUZA, Rafael Vieira de. **CAN (Controller Area Network): Uma abordagem para automação e controle na área agrícola.** Universidade de São Paulo. São Carlos – SP, 2002.

[BARBOSA, 2003] BARBOSA, Luiz Roberto Guimarães. **Rede CAN.** Escola de Engenharia da Universidade Federal de Minas Gerais. Belo Horizonte – MG, 2003.

[BRITES; SANTOS, 2008] BRITES, Felipe Gonçalves; SANTOS, Vinicius Puga de Almeida. **Motor de Passo.** Universidade Federal Fluminense. Niterói -RJ, 2008.

[HENRIQUES, 2004] HENRIQUES, Luís Oscar de Araújo Porto. **Implementação de estratégia de minimização de oscilações de torque e remoção de sensor de posição para um acionamento de relutância variável usando técnica Neuro-Fuzzy.** Universidade Federal do Rio de Janeiro. Rio de Janeiro – RJ, 2004.

[SOUZA, 2011] SOUZA, Jhonathan Junio de. **Motores de passo.** Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa. Ponta Grossa – PR, 2011.