

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

RAFAEL HUMBERTO MACHADO DA FONSECA

**SISTEMA DE MONITORAMENTO DE SENSORES DE
ROMPIMENTO DE FIO DE UMA URDIDEIRA UTILIZANDO
MICROCONTROLADORES**

VIÇOSA
2018

RAFAEL HUMBERTO MACHADO DA FONSECA

**SISTEMA DE MONITORAMENTO DE SENSORES DE ROMPIMENTO
DE FIO DE UMA URDIDEIRA UTILIZANDO
MICROCONTROLADORES**

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

Orientador: Prof. Dr. André Gomes Tôres.

VIÇOSA
2018


RAFAEL HUMBERTO MACHADO DA FONSECA

**SISTEMA DE MONITORAMENTO DE SENSORES DE
ROMPIMENTO DE FIO EM UMA URDIDEIRA UTILIZANDO
MICROCONTROLADORES**

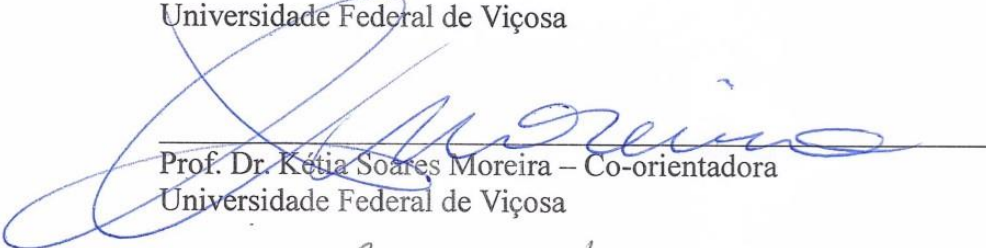
Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 06 dezembro de 2018.

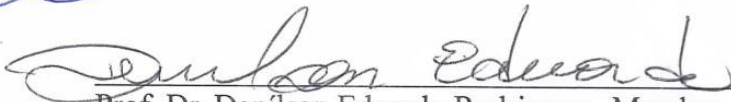
COMISSÃO EXAMINADORA



Prof. Dr. André Gomes Torres - Orientador
Universidade Federal de Viçosa



Prof. Dr. Kétia Soares Moreira – Co-orientadora
Universidade Federal de Viçosa



Prof. Dr. Denilson Eduardo Rodrigues - Membro
Universidade Federal de Viçosa

*“Irmão, você não percebeu
Que você é o único representante
Do seu sonho na face da terra
Se isso não fizer você correr, chapa
Eu não sei o que vai.”*

Música: Levanta e Anda

Emicida

Aos meus pais José Geraldo e Levina Edefátima.

Agradecimentos

Agradeço primeiramente a Deus, por ser minha fonte de fé e força, fornecer sabedoria e paciência durante minhas caminhadas e por ser o meu guia para todas as minhas ações e desafios. Agradeço ao meu pai, José Geraldo, e minha mãe, Levina Edefátima, por sempre me apoiarem, me incentivarem a estudar e possibilitarem tudo o que aconteceu de bom na minha vida. Eles sempre me deram a sustentação para seguir meus sonhos e buscar a minha própria felicidade, mesmo que distantes. São eles também que alimentam minha alma com todo amor e carinho que posso receber.

Agradeço à Vilma que sempre me olhou e torceu por mim, mesmo do céu. Ao Diego, Matheus, Gabriel e Francis que moraram comigo durante o meu tempo em Viçosa e se mostraram como verdadeiros irmãos perante a mim. A todos os amigos que fiz em Viçosa que me deram tantas alegrias nesta cidade, especialmente à Ana Clara, Lara, Leandro e Silas pois foram realmente uma família para mim durante todos esses anos de UFV. Ao Fidelis e Anailson, por me auxiliarem de forma técnica nesse trabalho. Aos meus amigos de Sete Lagoas, especialmente Marina. Ao pessoal da diElétrica, da equipe Baja Pererecas UFV e do Nero por permitirem me transformar em um profissional e uma pessoa melhor. Aos meus familiares, que de alguma forma contribuíram para a realização desse sonho.

Agradeço aos meus professores que foram grandes mestres com seus ensinamentos sobre engenharia e sobre a vida, principalmente ao professor André Gomes Tôrres, por me auxiliar nesse trabalho. Aos meus colegas da Cedro Têxtil que permitiram durante meu estágio, realizar este projeto. E por fim, agradeço à Universidade Federal de Viçosa, que me ensinou a engenharia elétrica, me permitiu amadurecer e a crescer.

Resumo

A indústria têxtil está inserida em um mercado competitivo, o que leva a necessidade do desenvolvimento tecnológico para sobreviver nesse setor. Neste contexto a Cedro Têxtil de Sete Lagoas solicitou o desenvolvimento de um sistema de baixo custo para monitoramento dos seiscentos sensores de rompimento de fio de uma urdideira utilizando microcontroladores. Interesse motivado em detectar um fio rompido no processo de urdimento e permitir, através dessa informação, que operador atue o mais rápido possível e libere mais tempo para o trabalhador desenvolver atividades de maior valor agregado para a companhia. Para que o sistema desenvolvido atendesse as necessidades da empresa foram traçados requisitos de manutenção, operação, físicos e de integração que o projeto deveria seguir.

Este trabalho visa apresentar todas as etapas de desenvolvimento do projeto e justificá-las abordando a elaboração do hardware através da seleção e do dimensionamento dos seus componentes, a escolha das estratégias de comunicação, processo de construção do software, as simulações e testes realizados. A partir do layout do sistema estudou-se o custo de fabricação e implementação do projeto.

As simulações e os testes realizados mostraram a eficiência do sistema e foram capazes de validá-lo. Esses testes também mostraram que todos os requisitos foram alcançados. Por fim, o projeto apresentou um custo de execução bem abaixo das outras alternativas para o monitoramento dos sensores de rompimento de fio quando comparada com o uso de CLP's para a tarefa.

Sumário

1	Introdução.....	14
1.1	Indústria Têxtil	14
1.2	O Tecido	14
1.3	Urdimento.....	17
1.4	Urdideira.....	18
1.5	Objetivo	25
2	Revisão Bibliográfica.....	27
2.1	Microcontrolador	27
2.2	CLP (Controlador Lógico Programável).....	28
2.3	Padrões de Comunicações	28
3	Desenvolvimento do Projeto	30
3.1	Hardware	30
3.1.1	Seleção da Rede.....	30
3.1.2	Conversor RS-232 e RS-485	31
3.1.3	Comunicação Ethernet.....	31
3.1.4	Endereçamento dos Microcontroladores	32
3.1.5	Display.....	33
3.1.6	Seleção do Microcontrolador	34
3.1.7	Leitura dos Sensores	36
3.1.8	Sistema de Alimentação, Proteção e Seleção dos Conectores	38
3.2	Software.....	40
3.2.1	Código Mestre	40
3.2.2	Código Servo	43
3.3	Simulação e ensaio	45
3.4	Layout das Placas de Circuito Impresso.....	48
4	Resultados	50
5	Considerações Finais.....	52
	Referências Bibliográficas	53

Apêndice A – Código Mestre..... 55

Apêndice B – Código Servo..... 62

Lista de Figuras

Figura 1- Tecido Plano, Tecido de Malha e Tecido Não Tecido	15
Figura 2- Etapas de fabricação de um tecido.....	15
Figura 3-Processo de urdimento	18
Figura 4-Esquemático de uma urdideira.....	18
Figura 5-Gaiola Em V vista superior em CAD.	19
Figura 6-Vista lateral da gaiola.	20
Figura 7- Gaiola e seus elementos.....	20
Figura 8- Interior da gaiola.....	21
Figura 9-Tensor	21
Figura 10-Lamela, o fio quando está correndo faz a trava ficar na posição mostrada	22
Figura 11- Quando o fio rompe, a trava cai e aciona, fechando um contato.	23
Figura 12- Lamela	23
Figura 13 -Chave magnética, quando fio rompe fecha essa trava.	24
Figura 14- Pente de distribuição.....	24
Figura 15 - O rolo de urdume no cabeçote da urdideira.....	25
Figura 16 - Rolo de urdume no início do processo de urdimento.....	25
Figura 17 - Exemplo de um microcontrolador.	27
Figura 18 - Rede RS-485 com o MAX485.....	31
Figura 19- Módulo Ethernet (ENC28J60).....	32
Figura 20- Microcontrolador na posição 10, do lado direito.....	33
Figura 21 - Esquema para o controle dos display de 7 segmentos.....	34
Figura 22 - PIC 18f4550 e PIC16f887a.....	36
Figura 23- Sistema de leitura do sensor pelo microcontrolador.....	38
Figura 24 -Esquema de ligação do LM7805.	38

Figura 25- Conectores seleccionados.	39
Figura 26 - Fluxograma do código mestre.	41
Figura 27 - Mensagem enviada para o microcotrolador 13.....	42
Figura 28 -Fluxograma do código servo.	44
Figura 29- Mensagem enviada pelo microcotrolador 13 informando o status dos seus sensores.	45
Figura 30 - Simulação do Servo.	46
Figura 31- Simulação da Rede RS-485	47
Figura 32- Teste na matriz de contatos.....	48
Figura 33- Layout da placa de circuito integrado mestre e do servo.	49

Lista de Tabelas

Tabela 1 - Características técnicas do projeto para o mestre.....	35
Tabela 2 -Características técnicas do projeto para o servo.....	36

1 Introdução

1.1 Indústria Têxtil

A produção de tecidos é baseada na confecção dos fios de origem vegetal, animal ou petroquímico, para a fabricação de tecidos com finalidades mais diversas e aplicações. Os artigos têxteis, como mercadorias submetidas ao processo industrial de produção, representam a mais antiga atividade organizada enquanto indústria [1]. A indústria têxtil, devido sua importância para a humanidade, foi a pioneira e serviu como palco de experimentação da primeira revolução industrial. Logo, a história da indústria têxtil é a própria narrativa da industrialização, transitando pelas suas diversas fases.

A primeira revolução industrial mecanizou a fiação e a tecelagem, trazendo consigo a notável dádiva da força motriz sob a forma da máquina a vapor [2]. A segunda revolução industrial surgiu a partir do desenvolvimento da indústria química, elétrica e do aço, com inúmeros impactos no processo de fabricação desse ramo industrial, o qual inseriu novos equipamentos como os motores elétricos. A digitalização do mundo através da informática e o desenvolvimento da automação, mais uma vez, mudou modo de trabalho nas fábricas de tecido. Hoje, percebe-se novamente que a indústria têxtil será impactada com a nova revolução industrial, graças ao apoio de entidades, como o Centro de Tecnologia da Indústria Química e Têxtil (CETIQT), sempre atento ao desenvolvimento tecnológico e científico. É a partir dessa vertente que se compreende a importância do entendimento e acompanhamento dos processos da indústria têxtil.

1.2 O Tecido

Para entender a indústria têxtil é necessário conhecer o seu principal produto: o tecido. O tecido é definido como um artigo a base de fios de origem natural, artificial ou sintética, que composto de diversas formas se torna uma cobertura capaz de formar roupas e outras vestimentas [3]. Os tecidos podem ser classificados em: plano, malha e não tecido. O tecido plano é o entrelaçamento de fios de urdume (fios longitudinais) e de trama (fios transversais). O tecido malha é sucessão de fios laçados consecutivamente no sentido do comprimento e da largura do

tecido. O tecido não tecido é feito pelas fibras unidas dos fios, dispostos de forma a não apresentar uma orientação. Na Figura 1 é possível observar os diferentes tipos de tecido.



Figura 1- Tecido Plano, Tecido de Malha e Tecido Não Tecido

O processo de fabricação do tecido moderno, a base de algodão, pode ser subdividido em quatro setores, que são: a fiação, a preparação, a tecelagem e o acabamento. A Figura 2 exhibe as etapas de fabricação de um tecido.

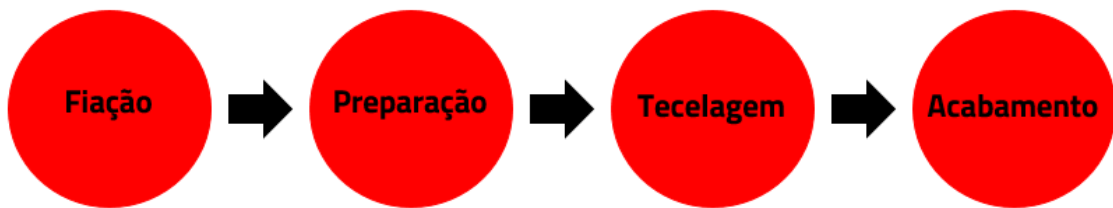


Figura 2- Etapas de fabricação de um tecido.

A fiação é a etapa da cadeia têxtil responsável por transformar a matéria-prima fibrosa, previamente tratada, em um fio, com relação de massa por unidade de comprimento, desejada por meio de um conjunto de operações previamente determinadas [4].

O primeiro passo desse setor é chamado de “Abertura”, no qual fardos de algodão são inseridos em um “Abridor de Fardos”, equipamento que apanha porções de cada fardo, os misturam e executam processos que retiraram do algodão galhos, folhas e outras impurezas, através de batedores, tendo flocos de algodão como produto. A cardagem é a próxima etapa, na qual os flocos de algodão são inserido na Carda, onde pentes de metal e processos mecânicos terminam a retirada de impurezas, iniciando o estiramento e torção dos fios, a fim de obter uma mistura maior das fibras e a fita de carda, como produto final.

Com o objetivo de uniformizar o peso por unidade de comprimento das fibras, é necessário paralelizá-las, utilizando um “Passador”, que também fará a junção de várias fitas e

fornecerá uma fita com a composição e espessura próprias. Por fim, essas fitas são inseridas em um “Filatório Open End” que produzirá de forma rápida inúmeros fios com características, espessura e qualidade especificadas.

No setor de preparação os fios, os mesmos são preparados para sofrerem o agressivo processo de tecelagem. A primeira etapa desse setor é o urdimento, que é a operação realizada na urdideira, a fim de retirar as inúmeras cargas de fios, de modo paralelo, para a uma única bobina de urdume. Essa bobina é uma carga do tear, que irá fornecer os fios de urdume na operação da tecelagem. A quantidade de fios que compõem o rolo de urdume e a largura desse rolo dependem do tipo de tecido que será confeccionado. Logo em seguida, o rolete de urdume vai para etapa de engomagem, na engomadeira, e tem a finalidade de impregnar os fios com uma película de goma, com o propósito de torná-los mais lisos e mais resistentes, assegurando um melhor trabalho no tear [5].

Depois da “Preparação”, os rolos de urdume engomados seguem para tecelagem, para a transformação dos fios em tecido, nos teares, na operação de tecelagem. Os teares mecânicos conseguem, sem interrupções, entrelaçar os fios de urdume e trama com uma alta produtividade. O produto desse setor já é o tecido, mas este precisa seguir para seção de acabamento, com finalidade de equiparar-se aos tecidos encontrados no mercado.

No “Acabamento”, a primeira etapa do tecido é desengomagem, realizada pela desengomadeira, com o intuito de tirar a goma posta pela engomadeira antes do processo de tecelagem. Em seguida, o tecido passa pela mercerizadeira que, com adição de soda cáustica e umectantes, passa por uma zona de impregnação, depois uma fase de estabilização e de lavagem. Esse processo trata o algodão sob condições determinadas de tensão longitudinal do fio, tempo, temperatura e alcalinidade [6]. Isso provoca a paralização das fibras dos fios, logo o tecido apresenta uma estabilidade dimensional, brilho, resistência e um maior rendimento tintorial. No final da etapa, é necessário que o tecido saia com um pH neutro.

O tecido então passa por processos de alvejamento, o que facilita o seu tingimento. No processo de tingimento, os artigos tomam banhos químicos em altas temperaturas, alguns recebendo banhos diferentes, um sobre o outro, permitindo uma maior variação de tonalidades para a peça confeccionada. Ainda no acabamento, o tecido tingido irá sofrer diversos processos físicos e químicos para oferecer maciez, solidez no tingimento, maior brilho e textura. Para agregar maior valor ao produto, pode-se ainda passar por procedimentos de estamparia e banhos químicos, que podem dar características antichamas, proteção contra agentes químicos, que

inibem a proliferação de microrganismos e até a propriedade de repelir insetos. Máquinas chamadas de endireitadoras de tramas através do processo de termofixação, conferem ao tecido estabilização e proporciona o controle final de largura para, em seguida, passar pela safonizadeira e sofrer a sanfonização, que consiste no encolhimento do tecido no sentido dos fios de urdume. São essas duas etapas finais que garantem o tamanho final do tecido e que garantem que as roupas confeccionadas com ele não encolham durante a lavagem caseira.

Antes do tecido ir para os consumidores, ele passa por testes de laboratórios e revisões técnicas para identificar defeitos, atestar sua qualidade e classificá-los.

1.3 Urdimento

Para a compreensão desse trabalho é importante focar no processo de urdimento. Esse processo consiste basicamente na passagem dos fios que formarão o urdume do tecido, transferindo-os de seus suportes iniciais para o rolete do tear [4]. Então, utiliza-se urdideira para executar essa tarefa, máquina que será mais detalhada nesse projeto.

Na urdideira, carretéis contendo fios são fixados na gaiola e fio a fio são levados até um pente, o qual determinará a largura do tecido a ser produzido na tecelagem e a distância entre cada linha. Depois de passar nesse pente, os fios são tracionados e conectados a bobina de urdume. Ao dar partida na máquina, o rolo começa a girar pelo acionamento dos motores, que tem a sua velocidade controlada pelo operador, realizando assim a transferência dos fios para o rolete da urdideira. Sempre que um fio arrebenta, sensores são ativados e param a máquina imediatamente, forçando o trabalhador a detectar o fio rompido, emendá-lo e retornar o processo. Essa atividade ocorrerá até que seja indicado que se alcançou o comprimento desejado para fabricação do tecido. Quando é acionado as tesouras que cortarão os fios e, se necessário, é acionado um sistema o qual substitui os quadros externos da gaiola pelos internos. Isso permite que o operador recarregue as gaiolas enquanto o processo transcorre. A Figura 3 mostra o processo de urdimento.



Figura 3-Processo de urdimento.

1.4 Urdideira

A urdideira é a responsável pelo urdimento, uma máquina composta basicamente por uma estrutura chamada de gaiola, um pente em cruz, um pente de distribuição e por um cabeçote, onde se localiza o motor. A Figura 4 exibe um esquemático da urdideira.

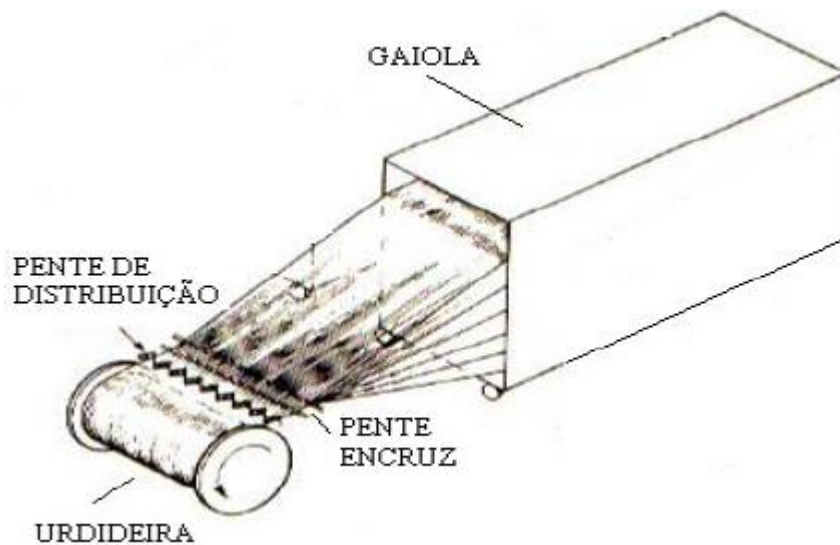


Figura 4-Esquemático de uma urdideira

A gaiola é uma estrutura onde ficam armazenadas as cargas de fios que alimentam o equipamento, sendo a capacidade da gaiola responsável por ditar a quantidade de fios que podem ser urdidos, além de garantir uma tensão constante nos fios e também monitorá-los. Ela pode apresentar diferentes concepções em sua estrutura e é composta por vários elementos como: suportes, tensores, sensores de rompimento de fio e tesouras.

Os suportes da gaiola são pinos metálicos que armazenam os carretéis dos fios e permitem que eles se desenrolem à medida que o processo segue. A gaiola estudada no trabalho é composta de seiscentos pinos, divididos em cinquenta colunas, com seis suportes para cada lado do sistema, sendo todos os suportes igualmente espaçados entre si. Pode-se então urdir seiscentos fios simultaneamente. A estrutura é chamada de “gaiola em V”, como mostra a Figura 5 a vista superior da máquina. Essa configuração é interessante, pois enquanto o processo ocorre nos quadros externos, o operador recarrega os quadros internos da referida gaiola. Isso permite que o processo possa recomeçar rapidamente quando se acabam os fios, pois, é necessário somente que responsável pela urdideira acione um mecanismo, o qual permite que instantaneamente os quadros sejam rotacionados e os suportes internos carregados passam a ser os suportes externos. A Figura 5 exibe a gaiola numa vista superior, a Figura 6 apresenta a gaiola numa vista lateral em CAD, a Figura 7 a gaiola numa vista lateral, mas exibindo suas estruturas e na Figura 8 tem-se a vista dos lados internos da gaiola.

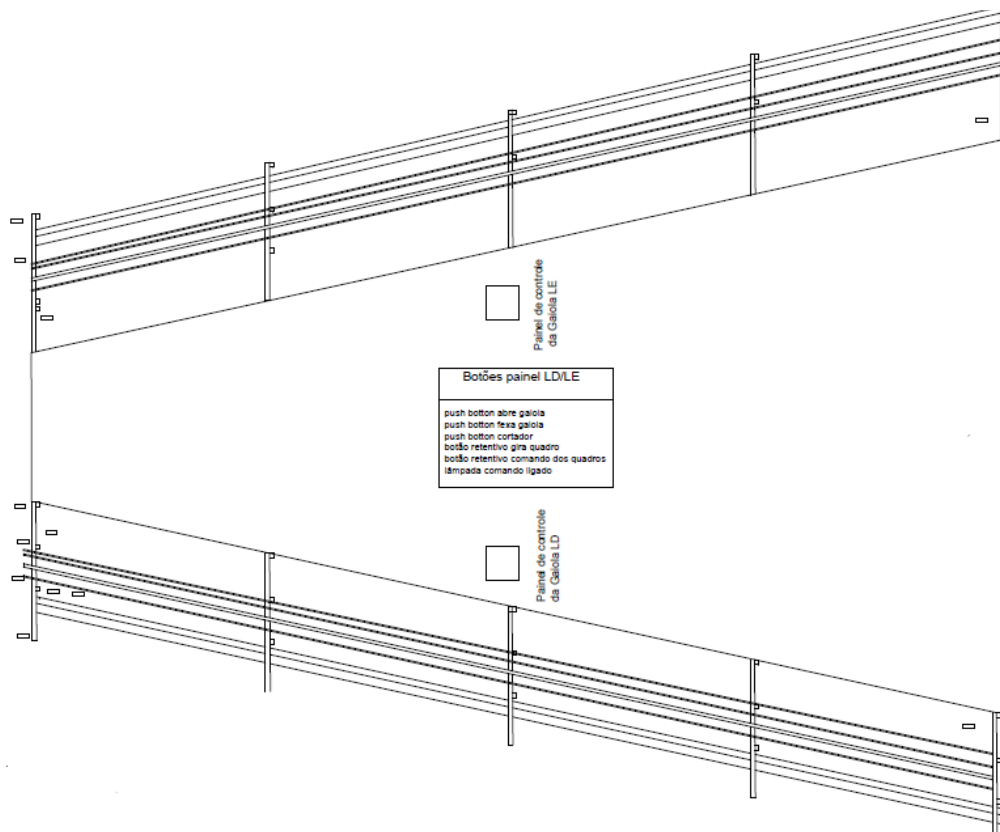


Figura 5-Gaiola Em V vista superior em CAD.

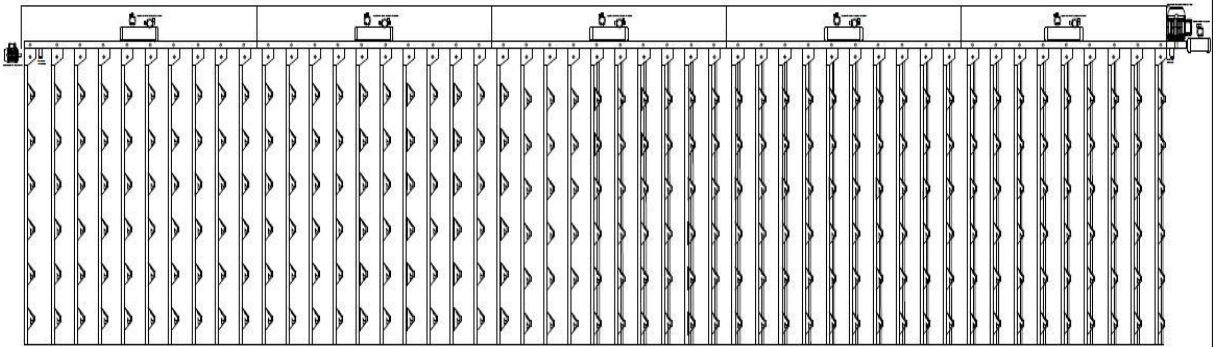


Figura 6-Vista lateral da gaiola.

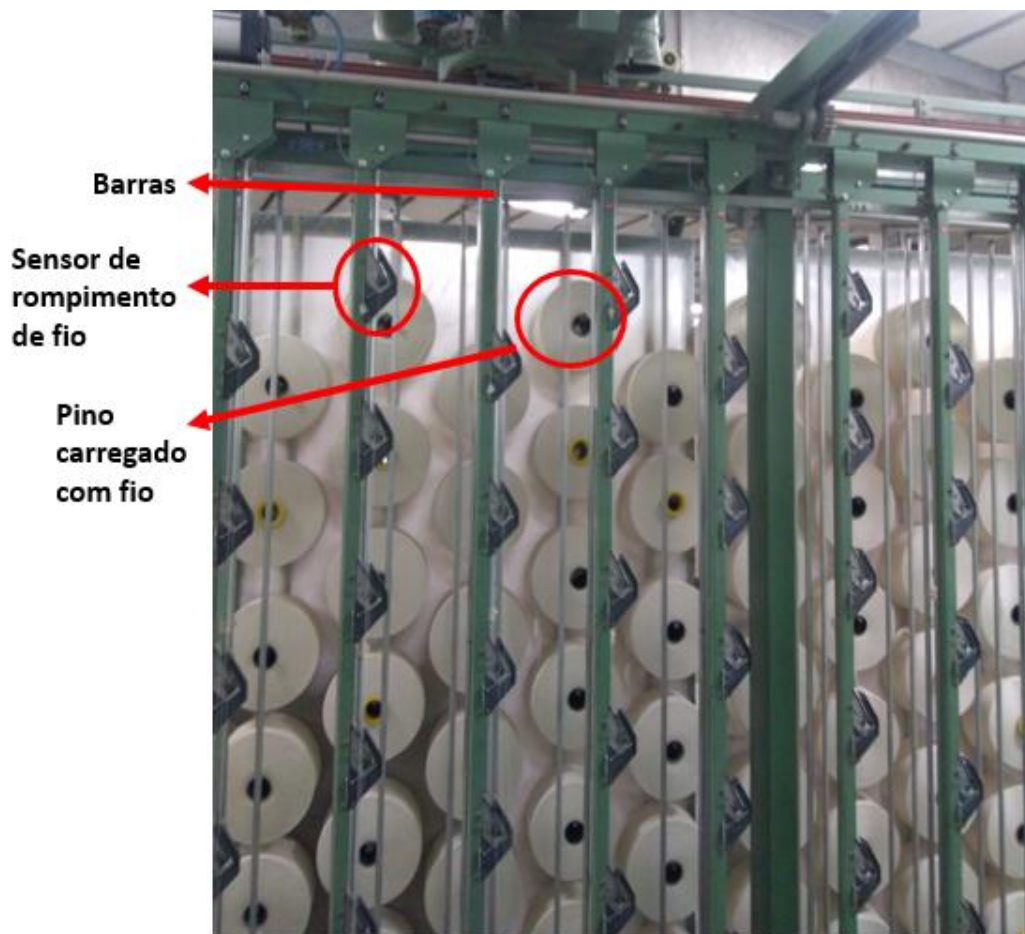


Figura 7- Gaiola e seus elementos.



Figura 8- Interior da gaiola.

Há também os tensores, que são mecanismos mecânicos ou eletrônicos, os quais prendem os fios e dão a tensão necessária a eles para que fiquem retos e não se embolem entre si durante o processo. Na Figura 9 é exibido um tensor.

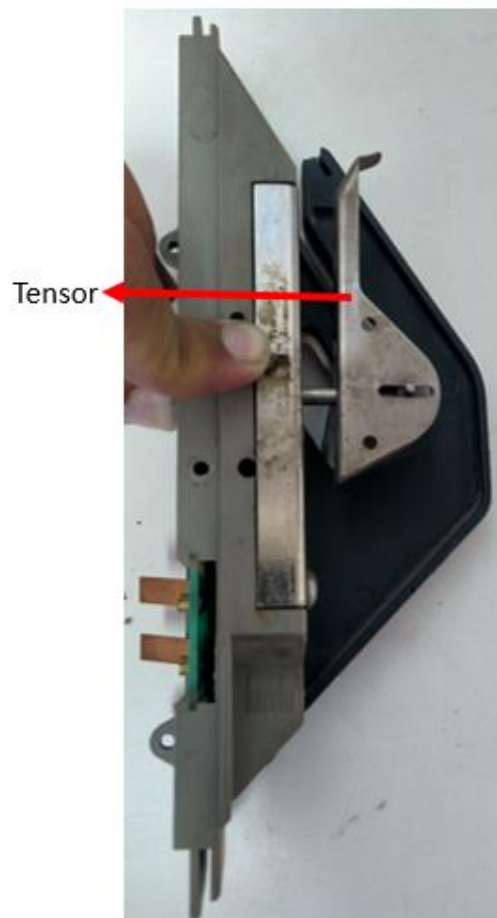


Figura 9-Tensor

Outro elemento muito importante são os sensores de rompimento de fios, também chamados de lamelas. Existe uma lamela para cada fio, logo, a máquina possui um total de seiscentas lamelas. Os sensores de rompimento de fio têm a importante função de detectar o rompimento de um fio e parar imediatamente o motor da máquina, interrompendo o processo e impedindo que o rolo de urdume tenha fios partidos ou faltantes, fato que impacta no processo de tecelagem.

O sensor é composto por uma pequena trava, a qual é presa pelo fio quando tencionado. Ao ocorrer o rompimento do fio, estas travas desarmam e fecham uma chave magnética que energiza um barramento de 24V, interrompendo o funcionamento do equipamento imediatamente. Sempre que a totalidade dos fios permitidos pela gaiola não é utilizada, é fundamental que desligue o sensor, feito através do bloqueio da trava que fecha a chave magnética. As Figura 10, Figura 11, Figura 12 e Figura 13 mostram as estruturas dos sensores de rompimento de fio.



Figura 10-Lamela, o fio quando está correndo faz a trava ficar na posição mostrada.



Figura 11- Quando o fio rompe, a trava cai e aciona, fechando um contato.



Figura 12- Lamela.

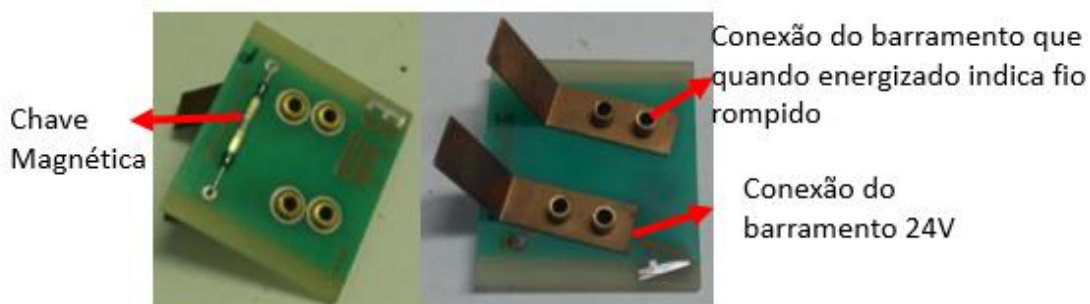


Figura 13 -Chave magnética, quando fio rompe fecha essa trava.

O pente de distribuição é um tipo de pente pelo qual os fios passam entre os seus dentes, que são totalmente reguláveis. Esse pente tem o propósito de determinar a densidade de fios e a largura do rolo de urdume, ao ajustar a distância entre os seus dentes. Além dessas atribuições, ele é importante para evitar o embaraçamento com os fios adjacentes e facilitar identificação de qual fio foi rompido no processo. A Figura 14 mostra o pente de distribuição.



Figura 14 - Pente de distribuição.

O cabeçote da máquina é a estrutura onde localizam-se os motores, que são responsáveis por girar o rolo de urdume no processo de transferência dos fios dos carretéis. Também, é nessa estrutura que encontra-se o painel de controle, pelo qual o operador pode acionar a partida da máquina, pará-la, ajustar parâmetros do processo, bem como: velocidade, metragem de fios que devem ser enrolados e etc. Ainda, é pelo painel de controle que acompanha-se a sinalização fornecida pela máquina, como em processo e/ou finalizado, rompimento de fio, alguns sinalizadores de defeitos e quantos metros de urdume fora fabricado. A Figura 15 mostra rolo de urdeume no cabeçote da máquina e a Figura 16 mostra o início do processo de urdimento.



Figura 15 - O rolo de urdume no cabeçote da urdideira.



Figura 16 - Rolo de urdume no início do processo de urdimento.

1.5 Objetivo

Esse trabalho tem como objetivo desenvolver um projeto para a Companhia de Fiação e Tecido Cedro e Cachoeira, Cedro Têxtil, de Sete Lagoas. O projeto consiste em elaborar um sistema de monitoramento dos sensores de rompimento de fio para a urdideira Santa Clara, da Cedro Têxtil, com um custo baixo de modo que ele seja capaz de comunicar com CLP (Controlador Lógico Programável), o qual controla todo o processo. O projeto visa facilitar o

monitoramento e a intervenção do operador da máquina, de modo que o processo corra de forma mais rápida e disponibilize mais tempo para o operador se dedicar às atividades que gerem maior valor para a companhia. Assim, o operador será informado sobre qual o fio foi arrebitado, através de uma IHM (Interface Homem Máquina), sem a necessidade da identificação de modo manual. Além disso, o monitoramento dos sensores possibilitará o fornecimento de informações para uma possível análise de dados e inserir esta indústria têxtil cada vez mais na indústria 4.0.

É necessário, então, que o sistema seja capaz de monitorar os seiscentos sensores de rompimento de fio presentes na urdideira, sendo possível de identificar quais fios foram rompidos, fornecer esses dados para o CLP e para a interface da máquina. Também, garantir a interrupção do funcionamento do equipamento, de forma rápida no caso de rompimento de um fio, para uma eficiente ação do operador.

A partir da análise do processo, reuniões com os técnicos de automação e elétrica, conversas com o engenheiro de automação e através das opiniões e relatos dos operadores da máquina, foram determinados alguns requisitos técnicos que o projeto deveria seguir. Os requisitos obrigatórios foram:

- **Requisitos Físicos (Elétricos e Mecânicos):**
 - ❖ Ler os sinais de 24V dos sensores;
 - ❖ Robustez necessária para o ambiente industrial;
 - ❖ Atenção ao cabeamento da máquina.
- **Requisitos de Operação:**
 - ❖ Processo não pode ser interrompido se um sensor ficar off-line;
 - ❖ Indicação luminosa na máquina informando o sensor rompido.
- **Requisitos de Manutenção:**
 - ❖ Um microcontrolador para cada duas barras;
 - ❖ Baixo custo;
 - ❖ Facilidade de manutenção;
 - ❖ Componentes de fácil reposição;
 - ❖ Conexões simples e robustas entre os fios;
 - ❖ Identificação de defeitos;
 - ❖ Sistema deve funcionar em módulos para simplificar a substituição e manutenção dos mesmos.
- **Requisitos de Comunicação/Integração:**

- ❖ Comunicar com CLP;
- ❖ Informar qual sensor foi rompido.

2 *Revisão Bibliográfica*

2.1 *Microcontrolador*

Um microcontrolador é um circuito integrado composto de uma unidade de processamento, memória, periféricos de entrada e saída (digitais e analógicas), temporizadores e de comunicação. Os microcontroladores surgiram como uma evolução natural dos circuitos digitais devido ao aumento da complexidade das necessidades atuais, chega um ponto em que é mais barato e mais compacto, substituir a lógica das portas digitais por um conjunto de processador e software [7]. Na Figura 17 tem-se um exemplo de um microcontrolador.



Figura 17 - Exemplo de um microcontrolador.

Esses dispositivos foram revolucionários por serem capazes de serem programados através de um código estruturado, para executar diversas funções, além de poderem ser regravados inúmeras vezes. O microcontrolador é formado basicamente por uma CPU (Unidade de Processamento Central), responsável por calcular e executar o código inserido no mesmo, uma memória PROM (Memória Programável Somente de Leitura), na qual são armazenadas as instruções do programa, uma memória RAM (Memória de Acesso Aleatório), cuja finalidade é armazenar as variáveis utilizadas pelo programa. Ainda, possui também periféricos para receberem sinais digitais e saídas digitais, conversores analógicos-digitais, para interpretar sinais analógicos e permitir que tenham saídas analógicas. Podem contar também com periféricos de comunicação para os mais diferentes protocolos de comunicação, comparadores, saídas PWM (*Pulse Width Modulation*), entre outros.

Os microcontroladores, devido ao seu tamanho e características computacionais, são muito utilizados em sistemas embarcados. Os sistemas embarcados são sistemas microprocessados, no qual a unidade de processamento (CPU) é completamente dedicada ao dispositivo ou sistema que ele controla [8]. A evolução dos sistemas digitais, a qual permitiu a fabricação desses dispositivos com baixo custo, viabilizou o uso dos mesmos nos diversos equipamentos, mudando a forma de interação do homem com os sistemas digitais e hoje facilita o processo que chamamos de internet das coisas.

2.2 CLP (Controlador Lógico Programável)

O controlador lógico programável (CLP) é um computador dedicado à automação industrial, tendo como campo aplicações em processos industriais de manufatura. Entretanto, pode ser aplicado a qualquer sistema que se deseja tornar automático. O CLP, então, é definido como um aparelho eletrônico digital, que utiliza uma memória programável para armazenar instruções e para implementar funções específicas, tais como: lógica, sequenciamento, temporização, contagem e aritmética, controlando, por meio de módulos de entrada e saída, vários tipos de máquinas ou processos [9].

Um CLP é constituído por módulos de entrada e de saída (hardware), onde as funções disponíveis podem ser programadas em uma memória interna (software), através de uma linguagem de programação que possui um padrão internacional chamado IEC 1131-3, uma fonte de alimentação e uma CPU [10].

2.3 Padrões de Comunicações

Existem vários padrões de comunicação utilizados por microcontroladores, criados para atenderem diferentes objetivos e finalidades, que utilizam diversas técnicas e estratégias para cumprir as suas propostas.

O padrão RS-232 é um padrão de comunicação serial assíncrona. Ela é utilizada em uma rede ponto a ponto, podendo trabalhar em full-duplex. Cada ponto recebe um nome, um lado é chamado de DTE (*Data Terminal Equipment*) e o outro DCE (*Data communication Equipment*). Essa nomenclatura remete à ligação entre um computador e o modem [11]. Esse padrão apresenta um *baud rate* comum de 300, 1200, 2400, 4800, 9600 e 19200 bits/s. É indicada para ser utilizada em pequenas distâncias para que não haja a perda de dados.

O padrão de comunicação RS-485 é bastante utilizado na indústria para as mais diversas aplicações. Esse padrão utiliza uma estratégia de sinais diferenciais o que permite uma filtragem eficiente dos ruídos captados pelo cabo ao longo do seu comprimento, tornando o padrão RS-485 robusto a interferências [12]. Também utiliza um par de fios trançados para a transmissão do sinal a fim de aproveitar a características dos campos eletromagnéticos e transformar a rede ainda mais blindada aos efeitos eletromagnéticos. Esses atributos transforma a rede RS-485 atraente aos ambientes industriais que contém inúmeras cargas indutivas e ruídos, também garantem eficiência na transmissão de dados de até 1219m. Uma rede construída nesse padrão deve ser construída na topologia de barramento e funciona no modo half-duplex, podendo atuar no modo full-duplex com a adaptações na rede. Essa rede também exige adição de resistores de terminação no intuito de impedir a reflexão dos sinais e limita o seu projeto a utilização de até trinta e dois dispositivos.

O padrão SPI (*Serial Peripheral Interface*) surgiu como uma alternativa a comunicação RS-232 entre os periféricos de um dispositivo, ao permitir que todos os elementos fossem inseridos em uma única rede, no qual cada um teria um endereço próprio. O SPI é um padrão de comunicação síncrona o que garante uma grande velocidade de comunicação, mas exige um dispositivo como mestre para controlar a rede e ditar a frequência do *clock* de comunicação. Para o controle da rede é necessário que cada elemento esteja conectado ao mestre através de um cabo independente, já que ele será usado para o controle da rede. Outra característica é que toda troca de dados acontece sempre em ambas as direções, em outras palavras, cada bit trocado entre o mestre e um servo traz um bit do servo para o mestre. Dessa forma, definimos que a comunicação é sempre full- duplex [13].

O padrão I²C (*Inter Integrated Circuit*) é uma comunicação serial síncrona criada para a comunicação de periféricos pela Philips. A sua rede necessita de dois barramentos: um barramento de *clock* e um outro barramento onde será enviado a mensagem, todos os dispositivos devem ser conectados a essa rede. A comunicação I²C se limita a 127 nós que corresponde ao tamanho máximo do endereço que consegue receber, que é de sete bits e a uma distância de até um metro. Esse método de comunicação serial é um dos mais práticos, simples e de baixo custo disponível no mercado. Pois sistemas uma vez complexos, podem ser construídos através de sistemas mais simples, mas também ajuda na manutenção, uma vez que cada dispositivo pode ser protegido separadamente [14].

3 *Desenvolvimento do Projeto*

3.1 *Hardware*

3.1.1 *Seleção da Rede*

A primeira parte do projeto foi selecionar o protocolo de comunicação entre os microcontroladores e, conseqüentemente, a topologia de rede que seria utilizada. Analisaram-se os periféricos de comunicação mais comuns nos microcontroladores que foram, o I²C, SPI e RS-232.

A comunicação I²C apresentava inúmeras vantagens, mas apresentou uma característica que inviabilizou a sua utilização, já que ela opera somente a uma distância máxima de 1m e apenas um lado da gaiola da urdideira apresenta uma distância de 15m.

O protocolo SPI também se mostrou interessante, mas foi eliminado por necessitar de 3+n dispositivos de fios para construir o seu barramento, porém, cada microcontrolador iria monitorar somente duas barras. Seria necessário, então cinquenta servos em toda a máquina. Logo o excesso de fios correndo na urdideira, exigido para sua instalação, o inviabilizou por dificultar a sua implementação física e o serviço de manutenção.

A comunicação RS-232 também não era viável (por ser um protocolo de comunicação ponto a ponto), mas a existência de circuitos integrados que a convertem para o protocolo RS-485 mostrou uma solução significativa. O protocolo RS-485 é uma rede industrial, robusta a interferências e funcionaria perfeitamente na distância de 15m, requerida no projeto. Mas a RS-485 suporta no máximo trinta e dois dispositivos conectados a ela, sendo que a máquina contaria com cinquenta. Como solução, decidiu-se construir duas redes RS-485 para cada lado do equipamento, cada rede contando com vinte e cinco servos monitorados por um mestre.

A **rede RS-485** foi selecionada para fazer comunicação entre os microcontroladores, sendo cada servo necessita de um periférico de comunicação de RS-232 e o mestre de dois desse.

3.1.2 Conversor RS-232 e RS-485

Para fazer a conversão entre os protocolos de RS-232 e RS-485 para a confecção da rede, selecionou o circuito integrado **MAX485** da MAXIM. O MAX485 é um receptor/transmissor de baixa potência, para uma taxa de transmissão de até 250kbps no modo Half-duplex. Ele apresenta um consumo de $120\mu\text{A}$ até $500\mu\text{A}$ e com uma temperatura máxima de 160°C , além da seleção do seu modo de funcionamento demandar 40ns, segundo o seu *dataSheet* [15]. Essas características o fizeram uma ótima opção para solucionar o problema, necessitando somente de dois resistores de 120Ω de terminação na rede e par de fios trançados, como mostra a Figura 18. É necessário, também, que o sistema faça o controle dos seus pinos \overline{RE} e DE, para habilitar e desabilitar a sua função de transmissor ou receptor, de modo a garantir que não haja colisões na rede e que cada microcontrolador consiga receber e enviar as mensagens no momento correto.

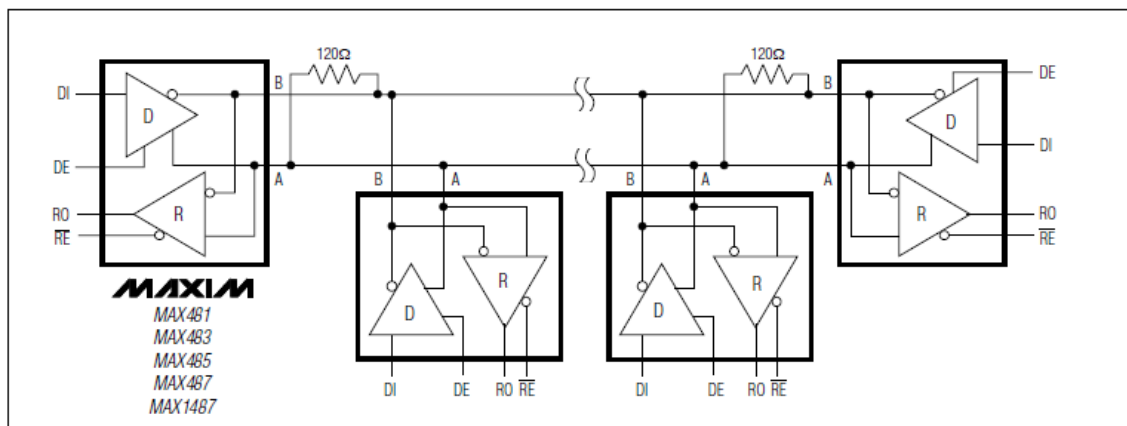


Figura 18 - Rede RS-485 com o MAX485.

Fonte: (DataSheet: Maxim, Low-Power, Slew-Rate-Limited RS-485/RS-422 [15])

3.1.3 Comunicação Ethernet

A comunicação Ethernet entre o microcontrolador mestre e o CLP pode ser feita através do **ENC28J60**, que é um *chip* conversor da comunicação SPI para o protocolo TCP/IP. O ENC28J60 é comercializado na internet através de uma interface inteiramente pronta, onde todo esquema eletrônico exigido pelo seu *dataSheet* [16] já é montado, exigindo somente que seja alimentado e a ligação SPI entre ele e o microcontrolador, conector ethernet, que já vem inserida nessa interface, como mostra a Figura 19. Logo é necessário que o mestre apresente também um periférico **SPI** para seu funcionamento.



Figura 19- Módulo Ethernet (ENC28J60).

3.1.4 Endereçamento dos Microcontroladores

Na intenção de atender ao critério de modulação do projeto, facilitar a manutenção, permitir a identificação de cada dispositivo na comunicação entre eles, percebeu-se a necessidade de cada um ser endereçado com um ID próprio.

O endereçamento poderia ser feito via software ou via hardware. Optou-se por fazê-lo via hardware, pois, por este modo existiria somente um programa genérico para todos os servos, ao invés de um programa específico para cada um. Quando uma placa estragasse, bastaria substituí-la por uma nova, inserindo somente de forma física o ID nela ou, se fosse trocado sua posição, exigiria apenas que colocasse o novo endereço, sem a necessidade, nesses dois casos, de reprogramar essas placas. Estratégia que simplifica o processo de manutenção.

O endereço de cada dispositivo era dado pela sua posição na gaiola, sendo que o primeiro receberia o ID 1 e assim por diante. Logo, o dispositivo 1 monitorava as barras 1 e 2, o dispositivo 2, as barras 3 e 4 e assim por diante, até a posição 25. Então, o endereçamento via hardware foi feito com uma chave **Dip Switch de 6 vias**. Essa chave recebia um número binário referente à sua posição nas 5 primeiras chaves seletoras e a última chave indicava se ele estava do lado direito ou esquerdo da máquina (o estado baixo para o lado direito e o alto para o lado esquerdo). Portanto, isso exigia do microcontrolador mais **seis pinos**. A Figura 20 mostra a chave *Dip Switch* para um dispositivo na posição 10, do lado direito.

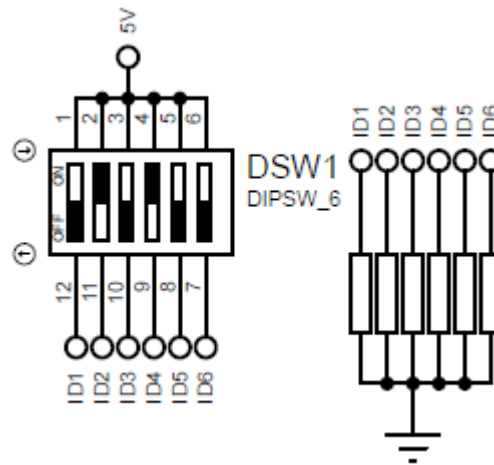


Figura 20- Microcontrolador na posição 10, do lado direito.

3.1.5 Display

Facilitar a identificação dos fios rompidos era um requisito do projeto, então, inspirados em outras urdideiras do mercado, estipulou instalar **displays de sete segmentos**, em cima das barras da gaiola, que mostrariam qual barra e a lamela que teve o fio rompido. Os dois primeiros displays seriam responsáveis por exibir a barra que teve o fio rompido e o terceiro ficou com a incumbência de indicar a lamela que houve o rompimento. Selecionou o display de 7 segmentos, catodo comum de 0,56” da Cromatek, um elemento de rápida resposta (ns), consumo de corrente direta de 30mA e uma corrente de pico de 100mA, para uma frequência de operação de até 1kHz, apresentando um baixo consumo de 60mW e uma temperatura de operação de até 90°C [17].

Para o display de 7 segmentos funcionar perfeitamente exige o controle dos seus sete pinos para cada um, como eram três displays totalizava vinte e um pinos, fato que inviabiliza sua utilização. Então, para reduzir o número de pinos, pensou-se em utilizar a estratégia de multiplexação dos displays, que consiste no acionamento de cada dígito com uma velocidade superior à da percepção do olho humano, produzindo o efeito de que os dígitos estão sendo exibidos simultaneamente. Essa tática exige o controle de sete pinos e mais um para cada display.

Para o controle de cada display usou-se o transistor **BC847B**. Esse transistor é capaz de suportar uma tensão de 45V entre coletor e emissor, uma corrente do coletor de até 100mA, com uma potência de até 250mW, um ganho de corrente de 150 vezes, uma frequência de chaveamento máxima de 100MHz e a temperatura de funcionamento limite de 150°C [18]. Esse transistor funcionaria como uma chave controlada pelo microcontrolador, habilitando ou

desabilitando o display associado a esse transistor. Para o perfeito dimensionamento, avaliou-se um resistor smd de $10k\Omega$ e $0,25W$ para o controle da corrente de base do transistor.

Usou-se também um circuito integrado, BCD (Binary Coded Decimal) na intenção de reduzir a quantidade de pinos do microcontrolador para controlar os dígitos. O BCD é um dispositivo que recebe na sua entrada um número binário e o converte em código decimal, para o controle de displays de 7 segmentos.

O componente BCD selecionado foi **CD4511**, que apresenta para até $100^{\circ}C$ uma potência de $500mW$, com uma tensão de alimentação de até $20V$ e possui em suas saídas uma tensão de $5V$, suportando uma corrente de porta de $10mA$ [19]. Baseado nessas características e nos outros elementos para a construção do sistema na utilização dos displays, foi necessário dimensionar resistores smd de 330Ω e $0,25W$ nas suas saídas para limitar a corrente consumida por ele, a corrente de alimentação dos LED's e a corrente no transistor.

A Figura 21 o mostra sistema usado para o acionamento dos displays e que ele utiliza **sete pinos** do microcontrolador para funcionar.

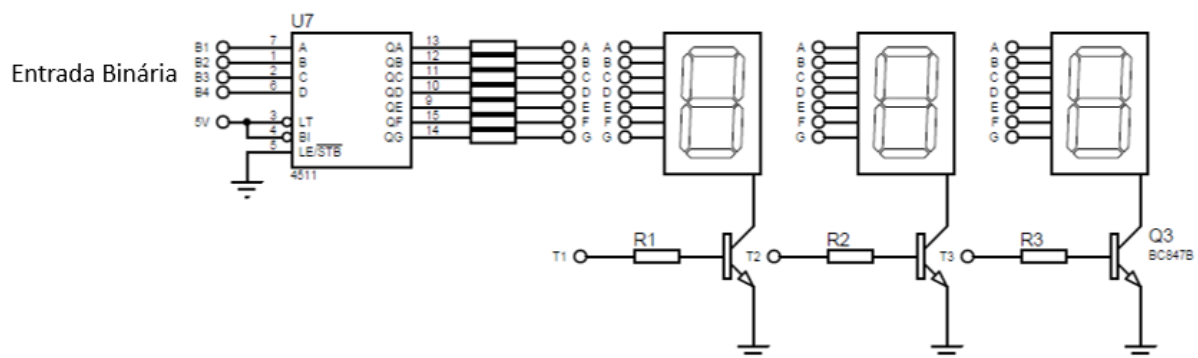


Figura 21 - Esquema para o controle dos displays de 7 segmentos.

3.1.6 Seleção do Microcontrolador

A seleção dos microcontroladores se deram baseadas nos objetivos traçados para o projeto e os requisitos técnicos exigidos a medida que o projeto foi se desenvolvendo.

O microcontrolador mestre, pelas análises já feitas, deveria ter duas entradas de comunicação RS-232, que requeria do dispositivo quatro pinos no total, mais duas saídas digitais para o controle da rede RS-485 e um periférico de comunicação SPI para instalar o modulo ethernet, exigindo mais quatro pinos. Uma saída digital para um LED que deveria ficar piscando, indicando o funcionamento desse microcontrolador e a leitura de sua programação.

Mais quatro entradas/saídas digitais, sendo uma entrada para receber uma saída do CLP, uma saída digital para enviar um sinal para o CLP parar a máquina e os outros dois pinos ficariam disponíveis para melhorias no projeto. Na Tabela 1 apresenta as características técnicas do projeto para selecionar o microcontrolador mestre.

Tabela 1 - Características técnicas do projeto para o mestre.

Função	Nº de Pinos
Comunicação RS-232	2
Monitoramento da rede RS-485	1
Leitura dos sensores (I/O)	12
Endereçamento (I/O)	6
Display	7
Total	28

A partir dessas informações foi feito um levantamento dos microcontroladores vendidos em três sites, escolhidos arbitrariamente, dedicados à venda de componentes eletrônicos, sendo eles Eletrodex (<https://www.eletrodex.com.br/>), FilipFlop (<https://www.filipeflop.com/>) e RoboCore (<https://www.robocore.net/>). O microcontrolador escolhido deveria ser vendido simultaneamente nos três sites, ter os periféricos exigidos, citados no parágrafo acima, o menor preço, vasto material nos fóruns de desenvolvedores e uma boa capacidade de memória. A necessidade de serem vendidos nesses três sites mostra sua facilidade de serem encontrados no mercado, o que conseqüentemente facilita a manutenção e o menor preço atende ao requisito de se ter o menor custo possível para o cliente. A exigência de ter um grande número de arquivos em fóruns de desenvolvedores facilita o avanço do projeto e obter soluções técnicas. A capacidade de memória do dispositivo é uma exigência técnica para suportar o programa que será inscrito nele. Logo, a partir dessas considerações o microcontrolador escolhido foi o **PIC 18f4550**, da MicroChip, para o mestre.

A escolha do microcontrolador servo exigia um periférico de comunicação RS-232, uma porta para monitorar a rede RS-485, doze entradas digitais para o monitoramento dos sensores de rompimento de fio, outras seis entradas para inserir o endereço do dispositivo, quatro pinos para exibir os dígitos, três saídas para o controle dos transistores responsáveis pela multiplexação dos displays. Utilizou-se a mesma metodologia para selecionar o mestre, atendendo à disponibilidade do microcontrolador nos mesmos sites, as funcionalidades exigidas, o menor preço, capacidade de memória e materiais e arquivos nos conceituados fóruns

de desenvolvedores. Então, o microcontrolador servo escolhido foi o **PIC 16f877a**. Na Tabela 2 apresenta as características técnicas do projeto para selecionar o microcontrolador servo.

Tabela 2 -Características técnicas do projeto para o servo.

Função	Nº de Pinos
Comunicação RS-232 (direito)	2
Comunicação RS-232 (esquerdo)	2
Controle da rede RS-485	2
Comunicação SPI	4
LED espia (I/O)	1
Portas Digitais (I/O)	4
Total	15

Na Figura 22 vemos os dois microcontroladores selecionados, com as suas respectivas pinagens.

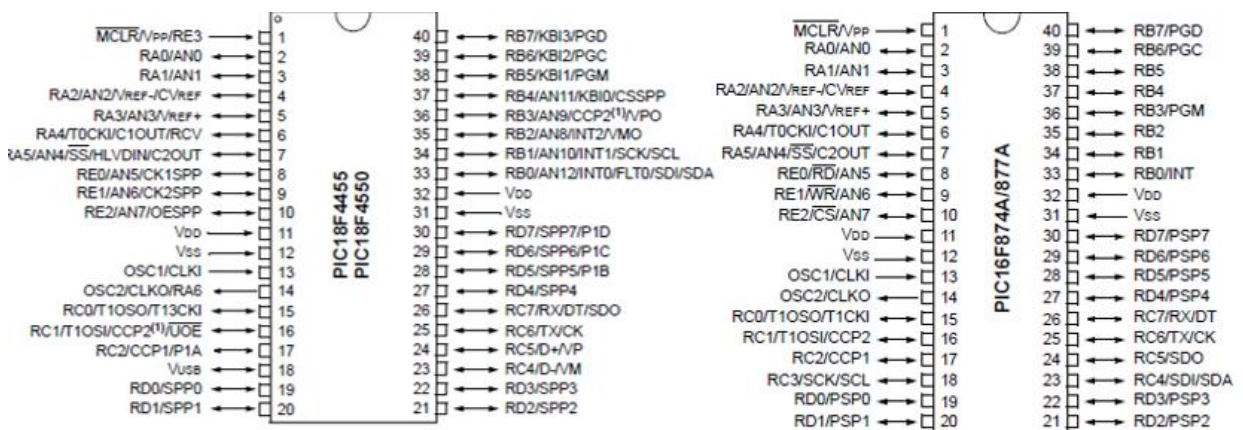


Figura 22 - PIC 18f4550 e PIC16f887a

3.1.7 Leitura dos Sensores

Ao detectarem o rompimento de um fio, as lamelas enviam imediatamente um sinal de 24V, mas o microcontrolador suporta no máximo um nível de tensão de 5V. Então, era necessário traçar estratégias que convertessem o sinal dos sensores para o nível de tensão aceito pelo sistema. A inspiração veio das mesmas técnicas usadas nos CLP's que trabalham com sinais de 24V, mas as suas CPU's precisam que o sinal seja convertido para 5V.

A técnica dos CLP's consiste em utilizar optoacopladores para a tarefa. O optoacoplador é simplesmente uma cápsula que contém um LED infravermelho e um fotodetector associado aos transistores. Quando se energiza o LED, o sinal é transferido por um feixe de luz ao

transistor, os optoacopladores são projetados com tempos de respostas tão pequenos que podem ser usados para transmitir dados na faixa de mega-hertz [20]. Como não existe contato elétrico entre os elementos do optoacoplador, ele permite a isolação elétrica entre circuitos e, também, a interferência de ruídos.

O optoacoplador escolhido foi o **4N25**, um dispositivo com capacidade de isolação de 5000V e uma temperatura de operação de até 125°C, o seu LED capaz de suporta uma corrente de 60mA e uma potência de 100mW e o transistor capaz de suporta uma corrente de 100mA no pino coletor, uma tensão de 70V entre o coletor e o emissor e dissipar uma potência de 150mW [21].

Para limitar a corrente de entrada do LED no 4N25, vindo do sinal de 24V dos sensores, colocou em série um resistor de 3,3kΩ de 0,25W. Já no lado do transistor, o pino coletor foi ligado na alimentação de 5V e no emissor foi conectado à entrada digital do PIC, para fazer a leitura do sinal e um resistor em série para limitar a sua corrente. O circuito montado apresentaria então uma saída em nível alto sempre que o LED fosse energizado. Aproveitando o acionamento do sinal do sensor quando rompesse, o fio colocou na sua saída também um resistor em série, de 3,3kΩ e 1W, com um LED de alto brilho, para acender sempre que um fio fosse rompido. Esse LED de alto brilho deveria ficar posicionado na própria gaiola da urdideira, perto da sua respectiva lamela, para quando o fio se rompesse, o operador chegasse perto da barra que teve o sensor ativado e fosse guiado em maior velocidade para a lamela, através dessa sinalização visual.

Esse sistema projetado, é mostrado na Figura 23, atende os requisitos de operação e manutenção, pois indica através de um LED o fio rompido e isola eletricamente os circuitos, garantindo uma robustez necessária para o ambiente industrial.

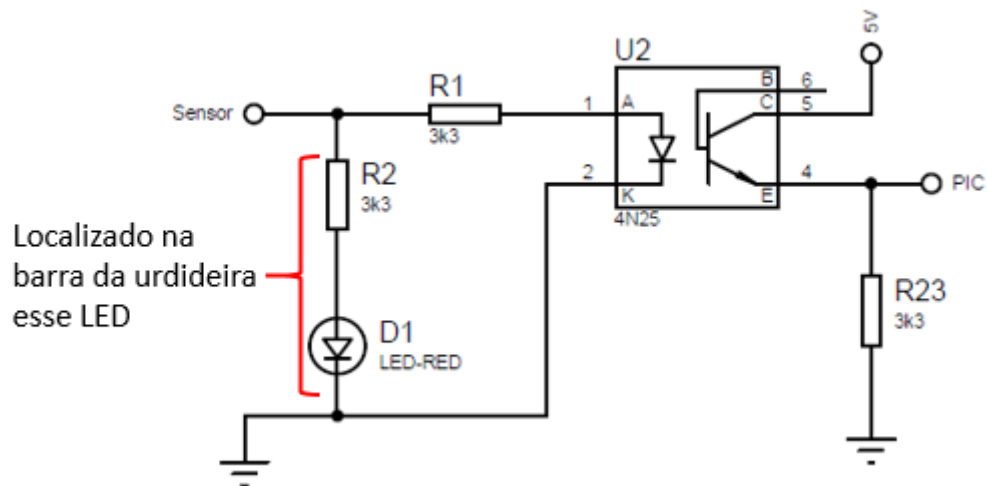


Figura 23- Sistema de leitura do sensor pelo microcontrolador.

3.1.8 Sistema de Alimentação, Proteção e Seleção dos Conectores

O sistema embarcado da urdideira funciona totalmente em uma tensão de 5V e o sistema de automação da máquina era alimentado por uma fonte contínua de 24V. Então, surgiu a necessidade de converter o nível de tensão da fonte para o nível do sistema.

Essa tarefa foi feita utilizando o **LM7805**. Esse circuito integrado recebe um nível de tensão contínua na entrada e tem na saída 5V. Possui como características técnicas uma entrada máxima de 25V, uma saída de 5V e uma corrente de até 1,5A [22]. O LM7805 teve ainda um capacitor cerâmico de **0,33 μ F**, em paralelo à sua entrada, e um outro de **0,1 μ F**, em paralelo a sua saída, como mostra a Figura 24 retirada do seu dataSheet [22].

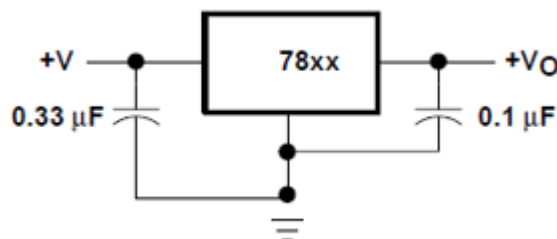


Figura 24 -Esquema de ligação do LM7805.

Fonte:(DataSheet: 7800 SERIES, Positive-Voltage Regulators [22])

O sistema de proteção foi concebido para aumentar vida útil dos dispositivos do projeto e atender aos requisitos de manutenção.

Para impedir que a inversão de polaridade danificasse os equipamentos, colocou-se um diodo antes a entrada do LM7805. Assim, impediria que uma corrente fluísse pelo circuito e o

danificasse, já que estaria trabalhando na polarização reversa quando ligado ao sistema com polaridade invertida. Selecionou o diodo **1N4007**, o qual apresenta uma tensão de ruptura de 1000V e suporta uma corrente de até 1A [23].

Inseriu também na entrada do sistema um fusível 500mA, para impedir que uma corrente de curto circuito queimasse o microcontrolador e os demais componentes.

Estabeleceu que ao se montar a placa de circuito impresso do projeto, deveria ser inserida uma malha de terra, afim de criar uma blindagem eletroestática e proteger o funcionamento dos circuitos integrados e da comunicação no ambiente fabril.

O propósito de utilizar bons conectores tem como objetivo diminuir o tempo de reparo e a substituição de dispositivos pelo setor de manutenção, além de diminuir problemas causados por mau contato elétrico. Os conectores escolhidos foram os bornes para placa de circuito impresso **FFKDSA1/H**, da Phoenix Contact. Esses bornes suportam uma corrente de 6A, uma tensão nominal de 160V, conexão *push-in* por mola [24]. A conexão *push-in* por mola permite uma conexão rápida dos cabos, sem a necessidade de ferramentas, pois esforço do contato definido assegura um contato estável por longo tempo. Já para a conexão ethernet usou o *plug* **RJ45** industrial, para fazer a interface ethernet da placa. A Figura 25 exhibe os conectores utilizados.

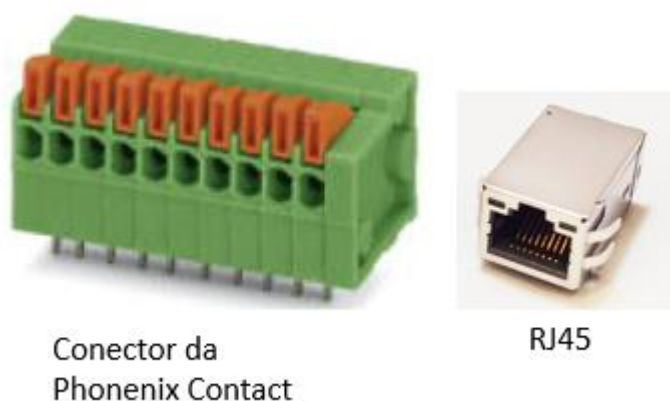


Figura 25- Conectores selecionados.

3.2 Software

3.2.1 Código Mestre

A programação do microcontrolador PIC 18f4550 funciona como o mestre e foi feita no CCS em código C.

O microcontrolador mestre recebe do CLP, via ethernet, o número de barras que serão utilizadas no processo de urdimento e no início/fim do processo. Ao receber o número de barras que serão utilizados no lado direito e esquerdo, ele irá configurar cada dispositivo que será utilizado, até obter o comando de partida. A partir daí, ele começa a monitorar a rede RS-485 do lado direito. Primeiro o mestre habilita esta rede para que todos o escutem e envia a pergunta para o primeiro elemento, habilita a rede para que servos respondam e aguarda essa resposta. Quando detecta o fim da resposta, o mestre novamente habilita o canal para que os dispositivos o ouçam e ele pergunta para o próximo microcontrolador, repetindo o processo para todos os elementos da rede. Ao acabar de verificar todas os pontos desta rede, ele começa a monitorar os elementos da rede RS-485, do lado esquerdo.

À medida que o mestre vai recebendo a resposta de cada microcontrolador, ele imediatamente processa essa resposta para identificar se houve um fio rompido, informando para o CLP exatamente qual fio foi rompido e em qual barra. O mestre espera até 14ms para que cada elemento responda. Se não responder dentro desse tempo ele automaticamente passa para o seguinte. Se um dispositivo ficar mais de três vezes seguidas sem responder, é informado um defeito nesse servo para o CLP. O mestre também coloca em estado alto um pino ligado no CLP sempre que detecta o rompimento de um fio. Ao se finalizar o processo, a máquina informar isso ao 18f4550 e ele imediatamente começa informa isso para todos os elementos da rede.

Na Figura 26 é possível ver o fluxograma montado para escrever o código que se encontra no Anexo A.

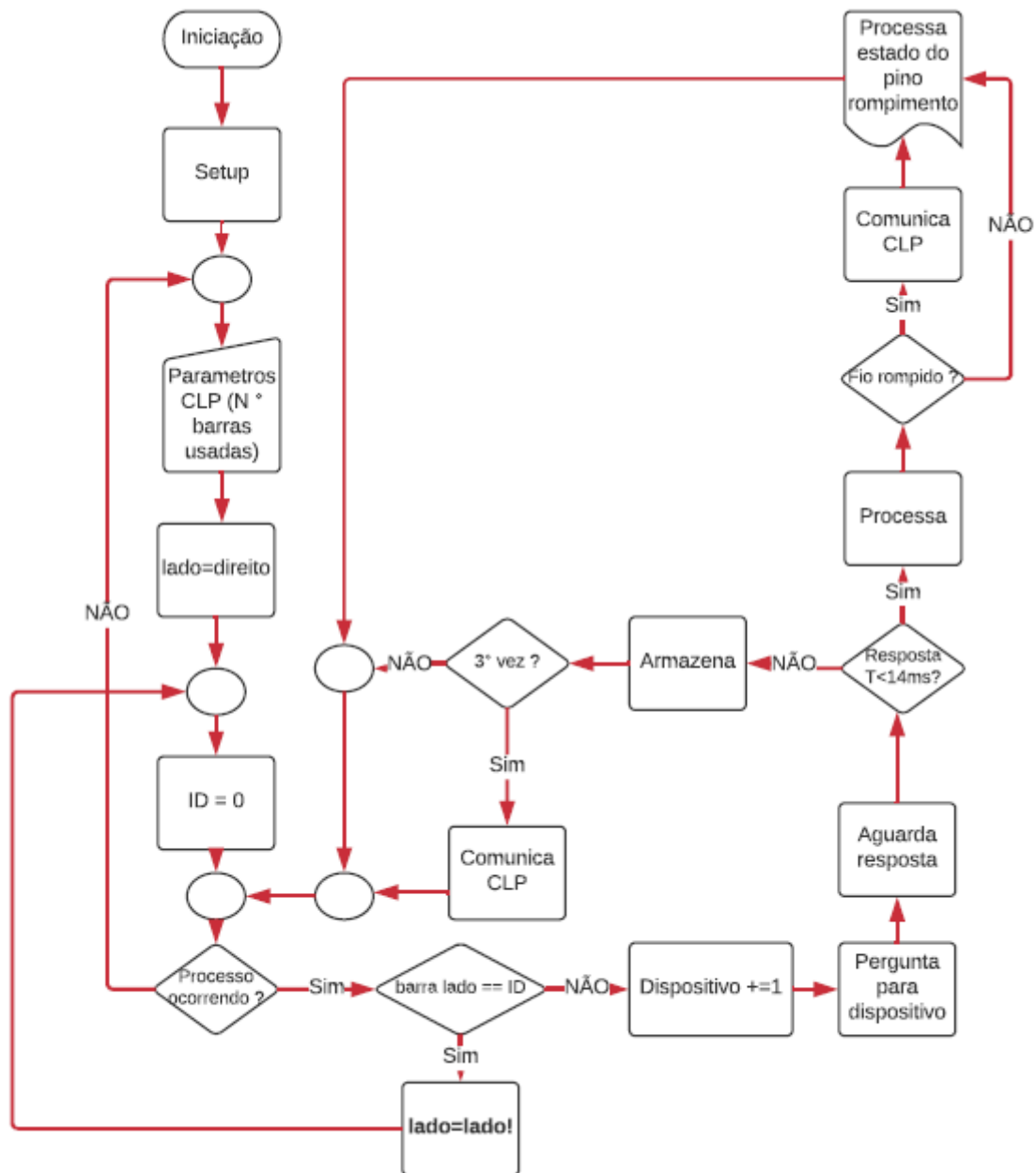


Figura 26 - Fluxograma do código mestre.

O código mestre começa configurando o hardware do microcontrolador, ajustando o tempo do *clock* e outras características físicas. É adicionado as bibliotecas que permitam a utilização do modulo ENC28J60 para a comunicação ethernet, é ajustado os canais de RS-485 para a construção da rede com um band rate de 19200bits/s. Todas as variáveis são declaradas e são definidos o nome dos pinos utilizados para auxiliar na programação e nas possíveis alterações que o projeto sofra. Os *timers* também são configurados e é onde também são escritas as instruções que fazem um LED piscar, indicando o funcionamento correto do sistema.

O programa não funciona enquanto não recebe os parâmetros do CLP com todas as configurações necessárias para que o processo ocorra. Quando isso acontece essas informações são salvas e repassadas para as próximas funções.

A função responsável por controlar a rede, ativa um contador que irá informar qual elemento da rede está trabalhando, esse contador varia à medida que se muda de dispositivo. Primeiro, um pino que controla a rede é colocado em nível alto, para que o dispositivo endereçado possa receber a mensagem. Então, a mensagem é enviada e altera-se o estado do pino para que o servo responda. A mensagem enviada pelo mestre de quatro caracteres demora 1,7ms para ser toda enviada e a resposta do servo gasta 6,3ms, logo o processo de pergunta e resposta demanda 8ms. O mestre espera até 14ms, caso não receba a resposta passa para o próximo elemento, quando isso acontece essa informação é salva em um vetor e caso se repita três vezes consecutivas para o mesmo microcontrolador é imediatamente informado ao CLP, informação que pode significar defeito nesse elemento. Esse processo de pergunta é repetido para todos os dispositivos que estão participando do processo. A verificação completa de um lado da rede leva 200ms e as duas redes 400ms. Quando acaba a verificação do lado direito, parte para o lado esquerdo da gaiola. Na Figura 27 vemos o formato da mensagem enviada para os microcontroladores.



Figura 27 - Mensagem enviada para o microcontrolador 13

À medida que o mestre vai recebendo os dados de cada servo, também vai os analisando. Primeiro, é verificado o endereço do microcontrolador que enviou e, depois, cada elemento dessa mensagem em busca de identificar um fio rompido. Quando isso acontece, a partir do endereço do servo, mais o estudo do vetor que contém a resposta, é possível informar exatamente qual lamela foi ativada. Em seguida, essa mensagem é informada e um pino em contato com o CLP fica em estado alto, até que o mestre receba uma mensagem da rede

informando que o problema foi resolvido. Esses algoritmos se repetem enquanto o urdimento estiver ocorrendo.

3.2.2 Código Servo

A programação dos microcontroladores PIC 16f877A que funcionariam como servos também foi feito no CCS, em código C.

Cada microcontrolador servo deveria ler e identificar o próprio endereço físico e monitorar a rede RS-485, em busca da sua configuração inicial e a sua ativação, que seriam enviadas pelo mestre. Quando identificasse a sua configuração e fosse habilitado a trabalhar, ele deveria começar a monitorar o estado dos sensores e aguardar as mensagens endereçadas a ele. Toda vez que o mestre perguntasse o estado dos sensores, deveria imediatamente responder pela rede a pergunta. Quando identificasse na sua rotina de monitoramento o rompimento de um sensor, colocaria em estado alto um barramento, indicando o rompimento e mostrando no seu display a barra que teve o fio rompido, assim como o fio que foi rompido. A utilidade de colocar um barramento em estado alto é permitir que o mestre ou CLP monitorasse esse barramento e diminuísse o tempo de resposta da máquina quando um fio fosse rompido. Esse processo deveria se repetir até que o mestre informasse que o processo havia acabado. Era importante também que o servo só respondesse quando fosse solicitado, para evitar colisões e a perda de dados.

Com o propósito de desenvolver um código que cumprisse as funções citadas, foi feito um fluxograma do algoritmo. A partir desse mapa, o código exibido no Anexo B seria elaborado. Na Figura 28 é apresentado o tal fluxograma.

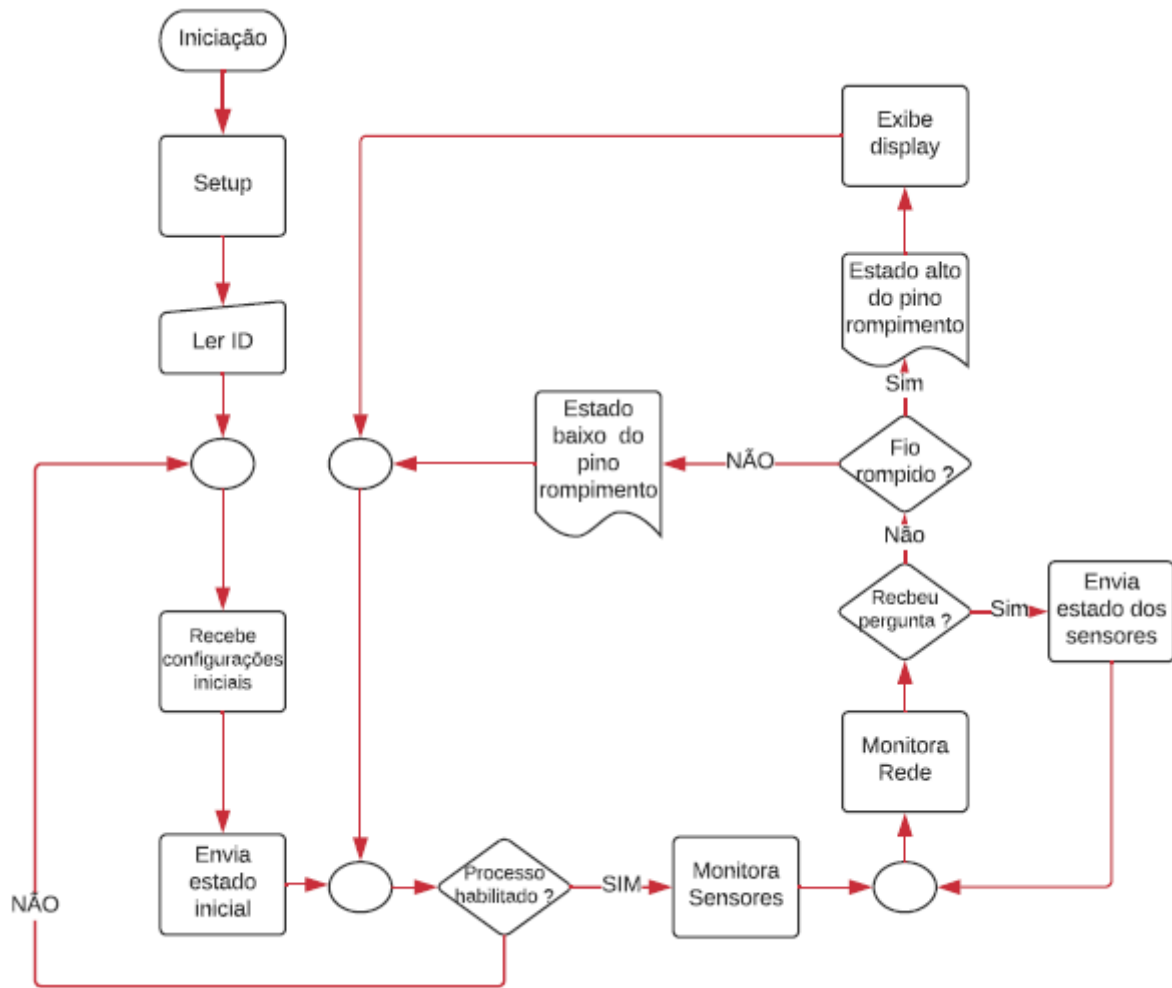


Figura 28 -Fluxograma do código servo.

Na primeira parte, o microcontrolador recebe as configurações de hardware, como a frequência e o método do *clock*, o status do *watch dog*, entre outras configurações de hardware. Nessa parte também, o protocolo de comunicação é configurado e o *baud rate* é ajustado em uma frequência de 19200bits/s, assim como as configurações de *timer* e as interrupções são ajustadas. Para, em seguida, as variáveis serem declaradas, assim como o nome dos pinos, com a finalidade de facilitar a programação.

A configuração do endereço era a próxima parte e, é feita a partir dos pinos que recebiam o endereço binário, para posteriormente convertê-lo em um número decimal.

A leitura do canal de comunicação é feita através da interrupção RDA, que sempre que detectar alguma informação no *buffer* de comunicação é ativada. Essa interrupção pega o dado nesse *buffer* e o transfere para um vetor. A partir daí inicia-se uma série de operações para receber os dados. Quando era detectado o fim da mensagem, avaliava se a mensagem era endereçada para o dispositivo. Se a mensagem fosse para o microcontrolador, começava a

e comunicar com todos os dispositivos. No servo primeiro avaliou a leitura do endereço, depois a rotina de monitoramento, a exibição dos displays e as rotinas de comunicação. Em cada simulação que era feita eram feitos ajustes e adequações sempre que necessário.

Depois dessa primeira fase de testes, começou a rodar a simulação completa do mestre, com todos os módulos unificados e o com seu algoritmo completo (com exceção da comunicação ethernet). De maneira semelhante, os testes validaram o projeto, atentando às adequações necessárias. O mesmo procedimento foi repetido para avaliar e analisar os microcontroladores servos e, mais uma vez, as alterações requeridas foram realizadas.

Por fim, foi simulada somente a comunicação entre os microcontroladores em uma rede RS-485, com um mestre e mais dois outros elementos na rede. Isso aconteceu devido às limitações computacionais, as quais impediam o computador de executar uma simulação completa do projeto. Mas, assumiu que se a rede composta de um mestre e mais dois servos tivesse êxito, funcionariam para n elementos. Novamente, as mudanças que foram exigidas para que os resultados fossem satisfatórios foram executadas. A Figura 30 mostra a simulação do microcontrolador servo e a Figura 31 mostra simulação da rede RS-485 com três elementos.

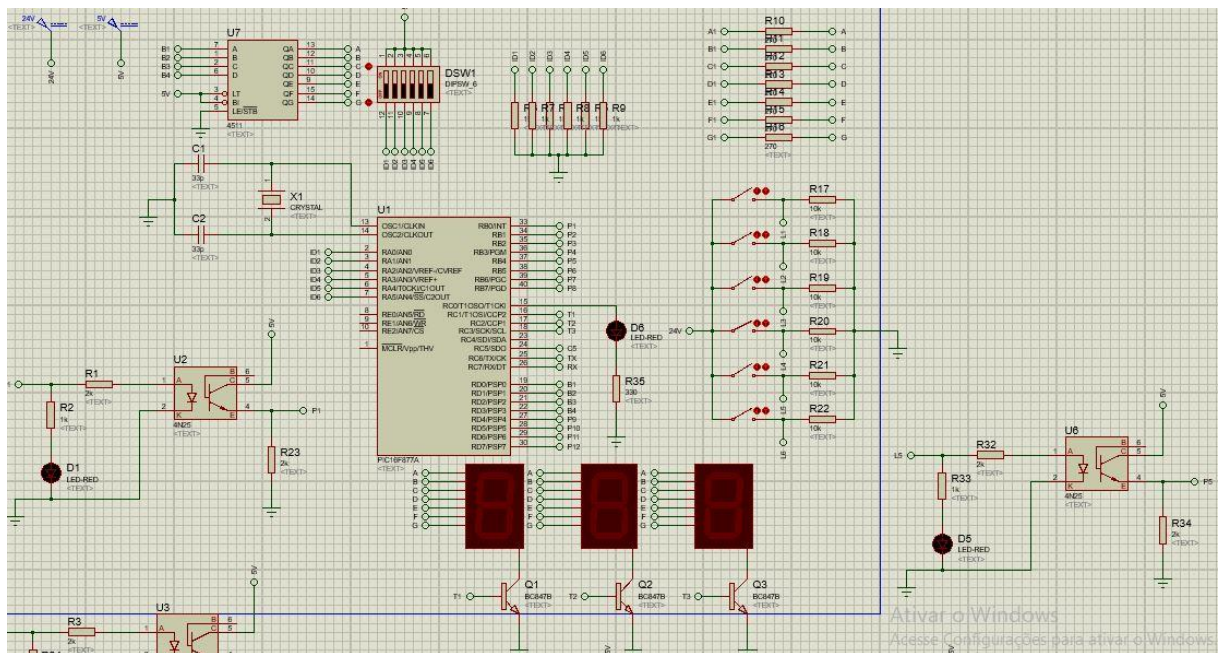


Figura 30 - Simulação do Servo.

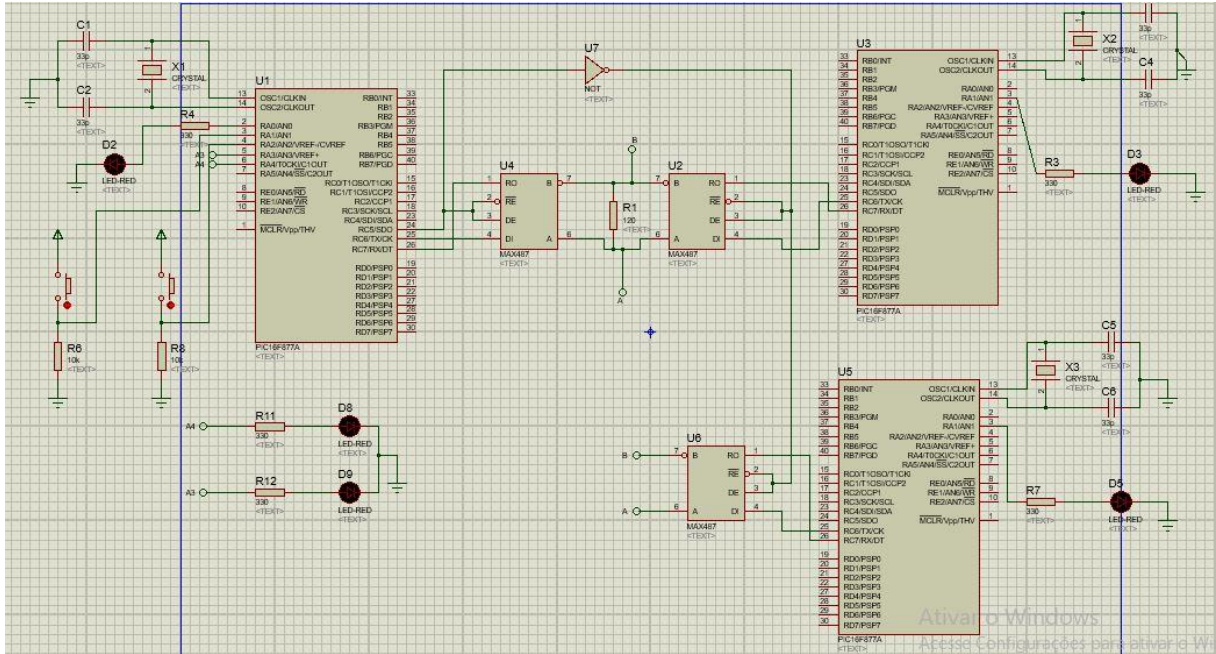


Figura 31- Simulação da Rede RS-485

Os ótimos resultados encontrados nas simulações permitiram que o projeto prosseguisse e protótipos fossem construídos.

Em uma placa de ensaio foram testados todos os circuitos, inicialmente, de forma independente. Primeiro, experimentou o circuito de leitura dos sensores, o sistema de endereçamento e de displays. À medida que os experimentos eram conduzidos, avaliações e adequações eram feitas para otimizar o máximo possível os resultados do projeto. O ensaio da rede foi feito com o 18f4550, para o controle da rede, e com dois 16f77a, como servos. Conseguiu-se provar a capacidade de gerenciamento da rede pelo mestre e sua capacidade de perguntar e responder, também, testou a capacidade de os servos responderem e perguntarem, sem que ocorresse a colisão de dados. Estipulou ainda, que se essa rede com três elementos funcionasse, ela funcionaria para uma rede com n elementos por inferência. Em conjunto, a comunicação Ethernet foi testada também na matriz de contatos, entre o mestre e um computador capaz de emular a comunicação ethernet de um CLP da Siemens. Na Figura 32 exibe o teste realizado na matriz de contatos.

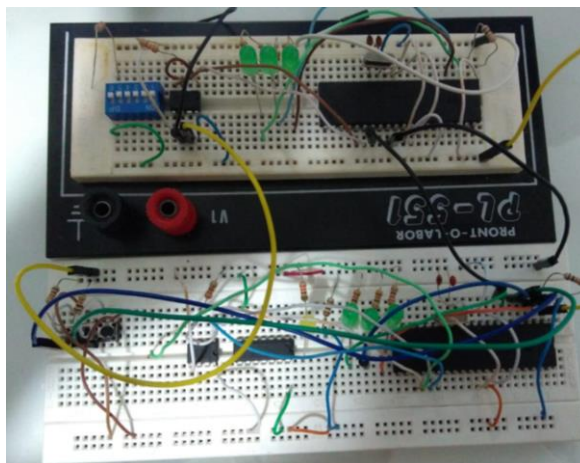


Figura 32- Teste na matriz de contatos.

Quando alcançou-se os resultados esperados, permitiu que o projeto seguisse para sua fase final, que era o esquema elétrico das placas de circuitos impressos para se obter o produto final.

3.4 Layout das Placas de Circuito Impresso

Depois de dimensionar todos os sistemas do projeto, validá-los e testá-los começou a se projetar as placas de circuito impresso, com o software ARES, programa projetado para o desenvolvimento de layouts de placas de circuitos integrados.

Primeiro, atentou-se para dimensões máximas de cada placa. A placa mestre deveria ter no máximo uma dimensão 10cm x 8cm e as placas dos servos, um tamanho máximo de 12cm x 65cm. A partir das dimensões das placas, começou-se a projetar os layouts, atentando sempre à algumas características, como a construção de uma malha de terra em todo o aparato, para proporcionar uma proteção eletromagnética e reduzir uma menor impedância para as correntes de retorno. Tentou-se agrupar circuitos de mesma funcionalidade como a seção de alimentação, as entradas e saída digitais de baixa frequência e os sinais de comunicação de alta frequência. O grande objetivo era impedir um que circuito interferisse no outro. Para diminuir as emissões de ondas eletromagnéticas, atentou-se por manter o sinal de clock perto dos microcontroladores, assim como os sinais de comunicação percorressem o menor caminho possível, para que sinais diferenciais ficassem próximos, com intuito de otimizar as vantagens do cancelamento dos campos magnéticos.

Outras medidas para alcançar um bom designer foram de não aceitar ângulos retos nas trilhas, a fim de evitar a reflexão dos sinais. Usou-se capacitores de desacoplamento nas

entradas de alimentação para impedir que ruídos de alta frequência alterassem o funcionamento dos dispositivos. Por fim, pensou em utilizar uma blindagem metálica para impedir que os sinais eletromagnéticos do ambiente prejudicassem o sistema.

Com essas informações e atento às recomendações, desenhou os layouts da placa de circuito impresso do mestre e do servo, que são mostradas na Figura 33.

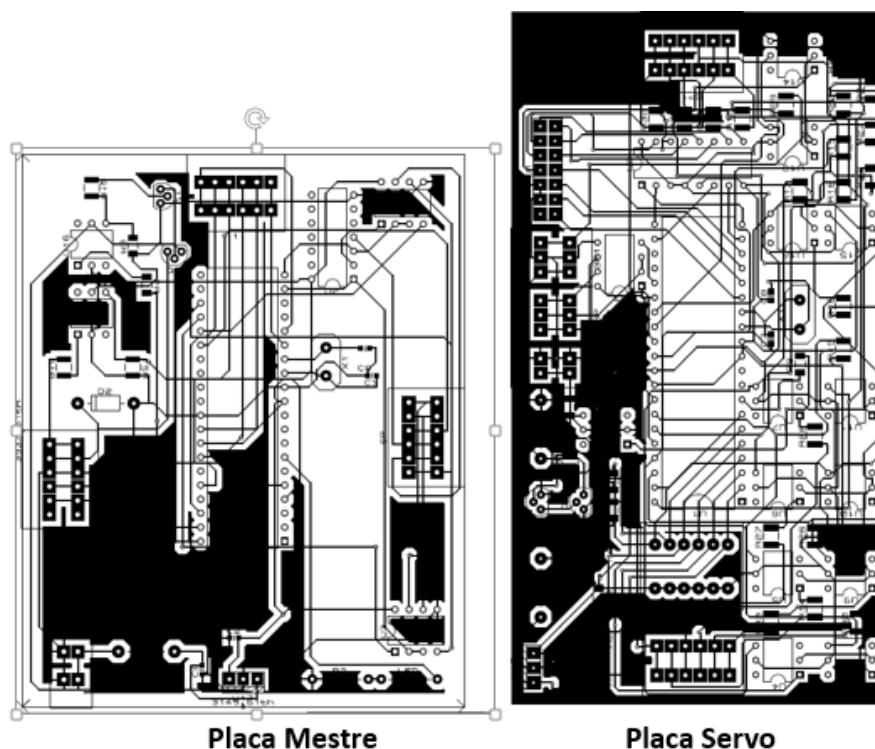


Figura 33- Layout da placa de circuito integrado mestre e do servo.

Já com os arquivos gerber das placas, a lista de materiais, os códigos usados e os orçamentos de empresas do ramo de eletrônica, solicitou que essas revisassem o projeto com relação ao hardware e ao software, confeccionassem duas placas mestre e cinquenta e cinco placas servos em um encapsulamento metálico. O custo estimado ficou no valor de R\$7788,00 no total, sendo para parte é destinado a consultoria no valor de R\$3000,00 para revisão do esquema elétrico e do layout e para a confecção das placas a soma de R\$4788,00, um valor de R\$84,00 por unidade para produzir as 55 placas servos e as 2 placas mestre.

4 *Resultados*

O desenvolvimento gradual desse projeto permitiu avaliar os seus resultados e desempenho à medida que ia se desenvolvendo, assim como os seus objetivos traçados.

Para obter um projeto robusto e confiável, dimensionaram-se todos os componentes baseados nas suas características elétricas encontrados nas suas folhas de dados, somados aos conhecimentos empregados de engenharia e eletrônica. É esse fato que justifica a adoção de par de fios trançados evitar perdas de comunicação, obedecendo ao padrão RS-485 e sua escolha, por ser uma rede industrial. A utilização da malha de terra na placa de circuito integrado e a blindagem metálica para elas protegerem os dispositivos e o funcionamento dos circuitos de comunicação. A adoção de bons conectores, para evitar falhas elétricas. Logo, no que tange ao projeto alcançou-se através dos estudos a garantia que o objetivo que exigia a robustez do projeto e sua confiabilidade fossem atingidos.

O requisito para que os sinais de 24V dos sensores fossem lidos pelos microcontroladores era meta importante para que os mínimos de alterações físicas fossem realizadas na máquina, até por causa dos outros sistemas de automação presentes nela. Que foi alcançado com a utilização dos optoacopladores, circuito que foi validado tanto nas simulações quanto experimentalmente. A escolha da rede RS-485 ocorreu por se mostrar a rede com o menor uso de fios e ser bastante utilizada nas indústrias. Como consequência, salientou que o projeto a todo tempo se mostrou cuidadoso com o número de fios que correriam pela máquina, elemento que dificulta a manutenção. Então, os requisitos físicos foram respeitados e atingidos.

A programação do mestre para que não interrompesse o processo quando um microcontrolador parasse de responder, é justificado pelo fato de que, caso um dispositivo estragasse, a urdideira não ficasse parada sem produzir, até que ele fosse substituído. Logo, outro objetivo também foi respeitado. Quando colocou-se um LED para acender perto da lamela que tivesse o fio rompido e displays que indicassem a barra e o fio rompido, tinha-se em mente garantir que o operador chegasse mais rápido possível ao fio rompido. Esses dois objetivos foram alcançados e, avaliados no ISIS e na matriz de contatos, consequentemente os requisitos de operação também foi alcançado.

Quando se exigiu um microcontrolador para cada duas barras, buscou-se que nenhum ficasse sobrecarregado, facilitando a operação de monitoramento realizada por eles e o defeito

em um, não compromettesse um alto número de barras, requisito este que também foi acatado como mostra o projeto. O uso de bons conectores, além de impedir problemas de mau contato, diminui o tempo gasto pelos técnicos no processo de manutenção, ao simplificar essa tarefa. Os microcontroladores e seus componentes estão passíveis de defeitos e devem ser substituídos, então, o projeto deveria usar elementos que fossem fáceis de encontrar no mercado, para que o sistema nunca ficasse inoperante por falta de componentes. Por isso também o projeto foi feito em módulos, porque no de caso defeito, bastaria endereça-lo e fazer a reposição de uma placa, o que facilitaria o serviço de manutenção. O mestre também era capaz de identificar em sua programação a ausência de um servo para o operador e, assim, medidas fossem tomadas para evitar que o processo ocorresse por muito tempo fora de sua plenitude. O conjunto dessas características facilita o serviço de manutenção, diminui o seu custo e tempo gasto, logo, todos os requisitos de manutenção foram cumpridos, resultado que se manteve até quando o projeto foi posto a prova nas simulações e nos ensaios físicos.

Como o projeto aqui desenvolvido faz parte de um maior sistema de automação, ele deveria ser capaz comunicar com um CLP, via Ethernet, para garantir a integração total da máquina. As simulações e os testes realizados nas placas de ensaios mostraram que o PIC 18f4550 era capaz de receber as informações vindas dos outros dispositivos, identificar exatamente o sensor rompido e informar isso pelo modulo ethernet, para o CLP. Barramentos que ligavam o sistema projetado ao CLP também eram estratégias e meios de comunicação, para agilizar atuação do CLP no processo de urdimento. Desse modo, a máquina tinha capacidade de agir rapidamente para evitar problemas no processo e mostrar, por uma IHM, exatamente o estado dos sensores e dos microcontroladores. Então, o requisito de comunicação também foi atingido.

O custo era um fator importantíssimo para a empresa e, por isso, também foi muito levado em consideração. A confecção das placas aqui desenvolvidos teve como custo total um valor de R\$7788,00. Outra alternativa para monitorar os sensores era o emprego de CLP, o que exigiria para monitorar os seiscentos sensores pelo menos 5 CPU's do modelo Siemens 7s 1200 com 8 cartões de 16 entrada cada CPU. Considerado o emprego de CLP's da Siemens, o projeto custaria como um todo o valor de R\$34000,00, contando que as 5 CPU's custam R\$1200 cada e cada cartão de 16 entradas um valor de R\$700,00. Portanto, é possível notar a existência de uma economia no valor de R\$26212,00 entre a estratégia do emprego do CLP e dos microcontroladores, exemplificando que o objetivo o qual exigia um baixo custo para o projeto foi plenamente alcançado.

5 *Considerações Finais*

O princípio de inovação é algo inerente à engenharia, o qual busca aplicar os conhecimentos científicos para melhorar a vida das pessoas e gerar valor para sociedade. Nesse contexto, a Cedro Têxtil, motivada em um mercado competitivo, objetivando melhorar seu processo de urdimento, aumentar a velocidade dos operadores na detecção dos fios rompidos, para que assim se tenha mais tempo para atividades de maior valor agregado, solicitou um projeto de um sistema para monitorar os seiscentos sensores de rompimento de fio com microcontroladores, em uma urdideira deles.

O desenvolvimento do projeto ocorreu obedecendo os requisitos do cliente e tentando alcançar os objetivos traçados por ele. Então, primeiramente, delimitaram-se estratégias para resolver os desafios impostos e começou-se assim a projetar a rede RS-485 para a comunicação dos microcontroladores, uma rede ethernet entre os dispositivos e o CLP da urdideira, o dimensionamento dos componentes do projeto no que se refere à hardware e a confecção de códigos que realizassem as tarefas propostas. Observando sempre as características físicas e de custo para justificar todas as escolhas realizadas.

Logo, através dos estudos, simulações e experimentos conseguiu-se um projeto para monitorar os seiscentos sensores da urdideira com microcontroladores e que atendesse todos os objetivos traçados e fosse mais viável economicamente que outras soluções abordadas. Projetou e se preocupou para que o sistema fosse simples e de fácil acesso para os técnicos da manutenção, conseguindo indicar com rapidez os sensores que tivesse os fios rompidos para os operadores da urdideira.

Referências Bibliográficas

- [1] URBAN, MARIA L. P., Desenvolvimento da produção de têxteis de algodão no Brasil. *Informações Econômicas*, SP, v.25, n12, dez. 1995.
- [2] SILVA, SILVIO, ITT- Introdução à Tecnologia Têxtil. SENAI, SC, Florianópolis, p. 7, 2002.
- [3] PEREIRA, GISLAINE S., Materiais e Processos Têxteis. Instituto Federal de Educação, Ciência e Tecnologia, SC, Campus Araranguá, p.42, 2009.
- [4] PEREIRA, GISLAINE S., Introdução à Tecnologia Têxtil – Curso Têxtil em Malharia e Confecção. Modulo 2, CEFET/SC, SC, Araranguá, p. 11, p. 40, 2012.
- [5] RIBEIRO, LUIZ G., Introdução à Indústria Têxtil. v. 2, CETIQT/SENAI, RJ, Rio de Janeiro, p.4, 1984.
- [6] Mercerização de Tecido Plano de Algodão, ..., p.9
- [7] PENIDO, ÉDILUS C. C.; TRINDADE, RONALDO S., Microcontroladores. Ouro Preto: Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais; Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, p.15, 2013.
- [8] OLIVEIRA JÚNIOR, MARCONI; DUARTE, RICARDO O., Apostila sobre Introdução ao Projeto com Microcontroladores e Programação de Periféricos. Universidade Federal de Minas Gerais, MG, Belo Horizonte, p.9, 2010.
- [9] ZACAN, MARCOS D., Controladores programáveis. 3. Ed. Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, RS, Santa Maria, p. 16, 2011.
- [10] MARTINS, G. M., Princípios de Automação Industrial. S.1.: s.n, 2007.
- [11] SOUZA, FÁBIO, RS-232/RS-485 – Hardware, Disponível em: <<https://www.embarcados.com.br/arduino-rs-232-rs-485-hardware/>>. Acesso em: 23/09/2018.
- [12] FREITAS, CARLOS M., Redes de Comunicação em RS-485, Disponível em: <<https://www.embarcados.com.br/redes-de-comunicacao-em-rs-485/>>. Acesso em: 26/09/2018.

- [13] SACCO, FRANCESCO, Comunicação SPI – Parte 1, Disponível em: <<https://www.embarcados.com.br/spi-parte-1/>>. Acesso em: 26/09/2018.
- [14] CAMARA, ROMULO, Protocolo I²C. Univasf. 11/2013.
- [15] MAXIM INTEGRATED PRODUCTS, DataSheet: Maxim, Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers: MAX481/MAX483/MAX485/MAX487-MAX491/MAX1487. Rev. 8, 03/2010.
- [16] MICROCHIP TECHNOLOGY INC. DataSheet: ENC28J60, Stand-Alone Ethernet Controller with SPI Interface. 2008.
- [17] CROMATEK ELETRÔNICA LTDA. DataSheet: Display 7 Segmentos, Série-0,52-Catodo Comum. Garulhos, SP, 06/03/2009.
- [18] PHILIPS SEMICONDUCTORS. DataSheet: BC846; BC847; BC848, NPN general purpose transistors. 04/02/2002.
- [19] TEXAS INSTRUMENTS INCORPORATED. DataSheet: CD4511 Types, CMOS BCD-to-7-Segment Latch Decoder Drivers. 07/2003.
- [20] BOYLESTAD, ROBERT L., Dispositivos eletrônicos e teoria de circuitos.8^a Ed., Prentice Hall, São Paulo, SP., p.631, 2004.
- [21] VISHAY SEMICONDUCTORS. DataSheet: 4N25, 4N26, 4N27, 4N28, Optocoupler, Phototransistor Output, with Base Connection. 07/01/2010.
- [22] TEXAS INSTRUMENTS INCORPORATED. DataSheet: 7800 SERIES, Positive-Voltage Regulators. 05/2003
- [23] VISHAY SEMICONDUCTORS. DataSheet: 1N4001 thru 1N4007, General Purpose Plastic Rectifier. 23/02/2011.
- [24] PHOENIX CONTACT. DataSheet: Borne para paca de circuito impresso – FFDSA1/H-2,54-3-1789317. 29/05/2018.

Apêndice A – Código Mestre

```

#include <18F4620.h>
#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES WDT128         //Watch Dog Timer uses 1:128 Postscale
#FUSES HS             //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPROTECT     //Code not protected from reading
#FUSES IESO           //Internal External Switch Over mode enabled
#FUSES NOBROWNOUT    //No brownout reset
#FUSES BORV21        //Brownout reset at 2.1V
#FUSES NOPUT         //No Power Up Timer
#FUSES NOCPD         //No EE protection
#FUSES STVREN        //Stack full/underflow will cause reset
#FUSES NODEBUG       //No Debug mode for ICD
#FUSES NOLVP         //No low voltage prgming, B3(PIC16) or B5(PIC18) used for
I/O
#FUSES NOWRT         //Program memory not write protected
#FUSES NOWRTD       //Data EEPROM not write protected
#FUSES NOEBTR       //Memory not protected from table reads
#FUSES NOCPB        //No Boot Block code protection
#FUSES NOEBTRB      //Boot block not protected from table reads
#FUSES NOWRTC       //configuration not registers write protected
#FUSES NOWRTB       //Boot block not write protected
#FUSES FCMEN        //Fail-safe clock monitor enabled
#FUSES NOXINST      //Extended set extension and Indexed Addressing mode
disabled (Legacy mode)
#FUSES PBADEN       //PORTB pins are configured as analog input channels on
RESET
#FUSES LPT1OSC      //Timer1 configured for low-power operation
#FUSES NOMCLR       //Master Clear pin used for I/O

#use delay(clock=2000000)
#use rs232(baud=19200,parity=N,xmit=PIN_C4,rcv=PIN_C5,bits=8,stream=direita)
#use rs232(baud=19200,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=esquerda)

#define PIN_ENC_MAC_SO    PIN_C0 // Conectar com PIN MISO do ENC28J60.
#define PIN_ENC_MAC_SI    PIN_B0 // Conectar com PIN MOSI do ENC28J60.
#define PIN_ENC_MAC_CLK   PIN_B1 // Conectar com PIN SCK do ENC28J60.
#define PIN_ENC_MAC_CS    PIN_B3 // Conectar com PIN CS do ENC28J60.
#define PIN_ENC_MAC_RST   PIN_B5 // Conectar com PIN RST do ENC28J60.
#define PIN_ENC_MAC_INT   PIN_C1 // Conectar com PIN INT do ENC28J60.

```

```

#define PIN_ENC_MAC_WOL    PIN_D3 // Conectar com PIN WOL do ENC28J60.
//Protocolos a utilizar.
#define STACK_USE_MCPENC TRUE
//#define STACK_USE_DHCP TRUE //Sem serviço DHCP
#define STACK_USE_ARP TRUE
#define STACK_USE_ICMP TRUE
#define STACK_USE_TCP TRUE
#include "tcpip/stacktsk.c"
#define MY_HTTP_SOCKET 80

```

```

#define rompimento1 pin_a0
#define rompimento2 pin_a1
#define espia pin_a5
#define comunicacao pin_c5

```

```

// Tempo de resposta-----
-----
unsigned int8 delay = 0;           // Auxiliar para tempo de resposta de cada escravo

```

```

// Comunicação -----
-----

```

```

#define BUFFER_SIZE 16 // tamanho do buffer
BYTE buffer[BUFFER_SIZE];
int next_in = 0;
int next_out = 0;
int t= 0;
char recebe[7]="xxxxxxx";
int cont=0; //tamanho da mensagem recebida
int hp=0; //habilita processamento

```

```

char inf[6] = "abcde$";
int i=0;

```

```

int offline = 150; //numero de vez necessarias para o mestre pular a comunicação
int ok=1;

```

```

int idto=0;
int ndemicro=20;

```

```

int aux0= 0;
int aux1 =0;

```

```

int j=0;
int w=2;

```

```

int estatus[25]; // falta inciar esse vetor com valores 0

```



```

int idoff=0;

int dir_esq=0; // 0 é direita e 1 esquerda

int contler=0;
int auxler=0;

#int_RDA
void RDA_isr(void)
{
    buffer[next_in]=getc();
    t=next_in;
    next_in=(next_in+1) % BUFFER_SIZE;
    if(next_in==next_out){
        next_in=t;        // Buffer full !!
    }
}

#define bkbhit (next_in!=next_out)// 1 se next_out diferente de next_in

BYTE bgetc() {
    BYTE c;

    while(!bkbhit) ;
    c=buffer[next_out];
    next_out=(next_out+1) % BUFFER_SIZE;
    return(c);
}

#int_SSP
void SSP_isr(void)
{
}

#int_TIMER2
void Timer2_isr(void)                // A cada x ms no PIC
{
    delay+=1;
}

void envia(){
    output_high(comunicacao);

    for(i=0;i<=5;++i){
        putc(inf[i]);
    }
}

```

```
    delay_ms(5);
    output_low(comunicacao);
}

void insereID(){
    aux0=(int)(idto/10);
    aux1=(int)(idto%10);
    inf[0]= aux0+'0'; // converte para string a dezena do Id de destino
    inf[1]= aux1+'0'; // converte para string a unidade do Id de destino
}

void enviarompimento(){
//envia para CLP
}

void processa(){

    for(j=2;j=13;j++){
        if(recebe[j]=='1'){
            enviarompimento();
        }
    }
}

void enviaestatus(){
//envia CLP
}

void fimdeprocesso(){

}

void enviasensor(){

}

void MACAddrInit(void)
{
    MY_MAC_BYTE1=0x00;
    MY_MAC_BYTE2=0x04;
    MY_MAC_BYTE3=0xA3;
    MY_MAC_BYTE4=0x00;
    MY_MAC_BYTE5=0x00;
    MY_MAC_BYTE6=0x01;
}

void IPAddrInit(void)
{
//IP Direto
```

```

MY_IP_BYTE1=192;
MY_IP_BYTE2=168;
MY_IP_BYTE3=0;
MY_IP_BYTE4=20;

//Porta de Enlace
MY_GATE_BYTE1=192;
MY_GATE_BYTE2=168;
MY_GATE_BYTE3=0;
MY_GATE_BYTE4=1;

//Mascara de Subred
MY_MASK_BYTE1=255;
MY_MASK_BYTE2=255;
MY_MASK_BYTE3=255;
MY_MASK_BYTE4=0;
}

void main()
{
    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_CLOCK_DIV_2|ADC_TAD_MUL_0);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SLAVE|SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DIV_BY_16,20,16);
    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    enable_interrupts(INT_RDA);
    enable_interrupts(INT_SSP);
    enable_interrupts(GLOBAL);
//Setup_Oscillator parameter not selected from Intr Oscillator Config tab

    MACAddrInit();
    IPAddrInit();
    StackInit();

    // TODO: USER CODE!!
    while(true){

```

```

if(ok==1){
    ok=0;
    idto+=1;
    insereID();
    envia();
    delay=0;
    if(idto==ndemicro){
        idto=0;
        if(dir_esq==0)
            dir_esq=1;
        else
            dir_esq=0;
    }
}
if(delay>=offline){
    ok=1;
    idoff= idto-1;
    estatus[idoff]=estatus[idoff]+1;
    if(estatus[idoff]==3){
        enviaestatus();
    }
}

while(bkbhit){
    recebe[cont]=bgetc();
    cont+=1;
    if(recebe[(cont-1)]=='$'){
        hp=1;
        contler= cont-1;
        cont=0;
        ok=1;
    }
}
if(hp==1){
    for (auxler=2;auxler<=contler;++auxler){
        if(recebe[auxler]=='1'){
            enviasensor();
        }
    }
    hp=0;
}
}

```

}

Apêndice B – Código Servo

```
#include <16F877A.h>

#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES HS              //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPUT          //No Power Up Timer
#FUSES NOPROTECT      //Code not protected from reading
#FUSES NODEBUG        //No Debug mode for ICD
#FUSES NOBROWNOUT     //No brownout reset
#FUSES NOLVP          //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOCPD          //No EE protection
#FUSES NOWRT          //Program memory not write protected
#FUSES RESERVED       //Used to set the reserved FUSE bits

#use delay(clock=2000000)
#use rs232(uart1,baud=19200,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)

//#priority timer2

// Mapeamento de Hardware -----
-----

#define Display1 pin_c1 //display dezena
#define Display2 pin_c2 //display unidade
```

```
#define Displaylamela pin_c3 //display de lamela

// Entradas das Lamelas -----
---

#define L1 pin_b0
#define L2 pin_b1
#define L3 pin_b2
#define L4 pin_b3
#define L5 pin_b4
#define L6 pin_b5
#define L7 pin_b6
#define L8 pin_b7
#define L9 pin_d4
#define L10 pin_d5
#define L11 pin_d6
#define L12 pin_d7

// Entrada do ID -----

#define ID1 pin_a0 // endereco de 0 a 32
#define ID2 pin_a1
#define ID3 pin_a2
#define ID4 pin_a3
#define ID5 pin_a4
#define ID6 pin_a5 // 0 para lado direito e 1 para lado esquerdo

// indica um fio rompido -----
-----

#define rompido pin_e0 // pino que manda um pulso para central se um fio romper

// Variáveis -----
```

```

// ID -----
int l_endereco[6]={0,0,0,0,0,0}; // vetor que recebe o endereço do PIC
int nID = 0; // numero em decimal do PIC
int x = 0; // selecionador da barra impar ou par
char IDp[2]="00"; // string do ID
int aux0= 0;
int aux1 =0;

// Comunicação -----
---

#define BUFFER_SIZE 16 // tamanho do buffer
BYTE buffer[BUFFER_SIZE];
int next_in = 0;
int next_out = 0;
int t= 0;
char recebe[7]="xxxxxxx"; // qual o tamanho deve ser esse vetor ??
int cont=0; //tamanho da mensagem recebida
int hp=0; //habilita processamento

// 7 Segmentos -----
-
unsigned int8 delay = 0; // Auxiliar para tempo de acionamento dos displays
int dezena = 0; // numero dezena da barra
int unidade = 0; // parte unitaria da barra
int nlamela = 0; // lamela utilizada
int nbarra = 0; // numero decimal da barra
int atvdisplay = 0;

```



```

// Monitorar sensores -----
-

int rompimento=0; // indica o rompimento de um fio

int i=0; // auxiliar para resetar o estado dos sensores

int1 sensores[12]={0,0,0,0,0,0,0,0,0,0,0,0}; // vetor q armazena o estado dos sensores

int habilita = 0;

// Envia -----

int k=0; // auxiliar o ID

int j=0; // auxiliar da mensagem

// Funções Auxiliares -----
-

void lerID(){ // Ler o ID setado no PIC

    endereco[0]=input(ID1); // endereco de 0 a 32

    endereco[1]=input(ID2);

    endereco[2]=input(ID3);

    endereco[3]=input(ID4);

    endereco[4]=input(ID5);

    endereco[5]=input(ID6); //0 para lado direito e 1 para lado esquerdo

    nID= endereco[0]*1; //conversão binario para decimal do ID

    nID= nID+endereco[1]*2;

    nID= nID+endereco[2]*4;

    nID= nID+endereco[3]*8;

    nID= nID+endereco[4]*16;

```

```

}

void display(){
  nbarra= nID*2;
  nbarra= nbarra-x; // ler a barra do sensor
  dezena=(int)(nbarra/10); // pega o digito da dezena do sensor
  unidade=(int)(nbarra%10); // pega a unidade da barra
  nlamela= nlamela; // pega o lamela rompida

  switch (delay){
  case 1: output_d (dezena); // Apresenta Digito[dezena_da_barra]
          output_high (Display1); // Habilita display 1
          break;
  case 4: output_low (Display1); // Desabilita display 1
          output_d (unidade); // Apresenta Digito[unidade_do_barra]
          output_high (Display2); // Habilita display 2
          break;
  case 7: output_low (Display2); // Desabilita display 2
          output_d (nlamela); // Apresenta N da lamela
          output_high (Displaylamela); // Habilita o Display da lamela
          break;
  case 10: output_low (Displaylamela); // Desabilita display da Lamela
           delay=0;
           break;
  }
}

```

```
}
```

```
void envia (){  
    for(k=0;k<=1;++k){  
        putc(IDp[k]);    // envia o ID do PIC  
    }  
    for(j=0;j<=11;++j){  
        putc(sensores[i]); // envia a mensagem do PIC  
    }  
    putc('$');    // fim da mensagem  
}
```

```
void monitora(){  
    rompimento=0;    // reseta a indicação de um fio rompido  
    for (i=0;i<=11;i++){ // reseta o estado dos sensores  
        sensores[i]=0;  
    }  
  
    if(input(L1)==1){  
        rompimento=1;  
        sensores[0]=1;  
        nlamela=1;  
        x=1;  
    }  
    if(input(L2)==1){  
        rompimento=1;  
        sensores[1]=1;
```

```
    nlamela=2;
    x=1;

}
if(input(L3)==1){
    rompimento=1;
    sensores[2]=1;
    nlamela=3;
    x=1;

}
if(input(L4)==1){
    rompimento=1;
    sensores[3]=1;
    nlamela=4;
    x=1;
}
if(input(L5)==1){
    rompimento=1;
    sensores[4]=1;
    nlamela=5;
    x=1;

}
if(input(L6)==1){
    rompimento=1;
    sensores[5]=1;
```

```
    nlamela=6;
    x=1;

}
if(input(L7)==1){
    rompimento=1;
    sensores[6]=1;
    nlamela=0;
    x=1;
}
if(input(L8)==1){
    rompimento=1;
    sensores[7]=0;
    nlamela=2;
    x=1;
}
if(input(L9)==1){
    rompimento=1;
    sensores[8]=1;
    nlamela=3;
    x=0;
}
if(input(L10)==1){
    rompimento=1;
    sensores[9]=1;
    nlamela=4;
    x=0;
```

```
}  
if(input(L11)==1){  
    rompimento=1;  
    sensores[10]=1;  
    nlamela=5;  
    x=0;  
}  
if(input(L12)==1){  
    rompimento=1;  
    sensores[11]=1;  
    nlamela=6;  
    x=0;  
}  
}  
  
#int_RDA  
void serial_isr(){  
    buffer[next_in]=getc();  
    t=next_in;  
    next_in=(next_in+1) % BUFFER_SIZE;  
    if(next_in==next_out){  
        next_in=t;    // Buffer full !!  
    }  
}  
  
#define bkbhit (next_in!=next_out)// 1 se next_out diferente de next_in
```

```
BYTE bgetc() {  
    BYTE c;  
  
    while(!bkbhit) ;  
    c=buffer[next_out];  
    next_out=(next_out+1) % BUFFER_SIZE;  
    return(c);  
  
}  
  
#int_TIMER2  
void Timer2_isr(void)           // A cada 2 ms no PIC  
{  
    if(atvdisplay==1){  
        delay+=1;  
        display();  
    }  
}  
  
// Programa Principal -----  
-  
void main()  
{  
  
    setup_adc_ports(NO_ANALOGS);  
    setup_adc(ADC_CLOCK_DIV_2);  
    setup_psp(PSP_DISABLED);  
    setup_spi(SPI_SS_DISABLED);
```

```

setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DIV_BY_16,48,13);      // Estouro a cada 2ms no pic
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);

enable_interrupts(INT_RDA);
enable_interrupts(INT_TIMER2);
enable_interrupts(GLOBAL);

lerID();
aux0=(int)(nID/10);
aux1=(int)(nID%10);
IDp[0]= aux0+'0'; // converte para string a dezena da barra
IDP[1]= aux1+'0'; // converte para string a unidade da barra

// TODO: USER CODE!!

while(true){

while(bkbhit){
recebe[cont]=bgetc();
cont+=1;
if(recebe[(cont-1)]=='$'){ //localizar o fim da mensagem
hp=1; // habilita o processamento de dados
cont=0; // inicializa o contador do vetor
}
}
}

```



```

if(hp==1){ //habilita o processamento
  if(recebe[0]==IDp){
    if(recebe[1]==IDp){ //verifica se a mensagem é para ele
      if(recebe[2]=='e'){
        habilita =1;
      } else{
        habilita=0;
      }
      envia(); //envia a situação do sensores
    }
  }

  hp=0;
}

if(habilita==1){
  monitora(); // função que monitora o estado dos sensores
  output_bit(rompido,rompimento); // ativa um reler que indica um fio rompido
  if(rompimento==1){
    atvdisplay=1; //display que somente é chamado quando um fio arrebenta
  }else{
    atvdisplay=0;
  }
}
}
}
}

```