

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

**DARNEY JABS**

**PLACA MICROCONTROLADA PARA CONTROLE  
SUPERVISÓRIO COM INTERFACE WEB**

**TRABALHO DE CONCLUSÃO DE CURSO**

**TOLEDO  
2015**

**DARNEY JABS**

**PLACA MICROCONTROLADA PARA CONTROLE  
SUPERVISÓRIO COM INTERFACE WEB**

Trabalho de Conclusão de Curso  
apresentado como requisito à obtenção  
do título de Bacharel em Engenharia  
Eletrônica, da Coordenação de  
Engenharia Eletrônica, da Universidade  
Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Paulo de Tarso  
Neves Junior

TOLEDO  
2015



Ministério da Educação  
**Universidade Tecnológica Federal do Paraná**  
Câmpus Toledo  
Coordenação do Curso de Engenharia Eletrônica



---

## TERMO DE APROVAÇÃO

Título do Trabalho de Conclusão de Curso Nº 017

### **Placa Microcontrolada para Controle Supervisório com Interface Web**

por

**Darney Jabs**

Esse Trabalho de Conclusão de Curso foi apresentado às 14:40 h do dia **25 de novembro de 2015** como requisito parcial para a obtenção do título **Bacharel em Engenharia Eletrônica**. Após deliberação da Banca Examinadora, composta pelos professores abaixo assinados, o trabalho foi considerado **APROVADO**.

---

Prof. Dr. Felipe Walter Dafico Pfrimer  
(UTFPR-TD)

---

Prof. Dr. Vilson Luiz Dalle Mole  
(UTFPR-TD)

---

Prof. Dr. Paulo de Tarso Neves Junior  
(UTFPR-TD)  
Orientador

**Visto da Coordenação**

---

Prof. M. Alessandro Paulo de Oliveira  
Coordenador da COELE

## **AGRADECIMENTOS**

Agradeço meu pai, Artur, e à minha mãe Lucila. Ao meu irmão Darios.

Agradeço as minhas irmãs Débora e Denny pelo incentivo e ajuda durante todo o curso.

Agradeço aos meus amigos e pessoas que convivi durante todos os anos de faculdade.

Agradeço ao meu orientador Prof. Dr. Paulo de Tarso Neves Junior.

## RESUMO

JABS, Darney. **Placa Microcontrolada para Controle Supervisório com Interface Web**. 2015. 74f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná, 2015.

O presente trabalho apresenta o desenvolvimento de um dispositivo microcontrolado com o protocolo TCP/IP embarcado capaz de conectar-se a Internet para controlar processos e fornecer dados para supervisão em tempo real capaz de se comunicar com o usuário através de uma interface *web*. A placa microcontrolada, baseada no microcontrolador PIC18F4620 e no protocolo TCP/IP embarcado, oferece ao usuário versatilidade e baixo custo no controle de processos. O desenvolvimento das páginas *web* é realizado em códigos HTML e CSS com funções *JavaScript*. Como exemplo de aplicação, foi projetada uma placa de circuito impresso microcontrolada com funções de entradas e saídas digitais, modulação por largura de pulso (PWM) e conversões analógica/digital controlada por uma página *web*.

**Palavras Chaves:** TCP/IP Embarcado, Acionamento e Monitoramento, Microservidor

## ABSTRACT

JABS, Darney. **Micro-controlled Device to Supervisory Control with Web Interface**. 2015. 74f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica) – Federal University of Technology - *Paraná*. Toledo, 2015.

This work presents the development of a microcontrolled device with the TCP/IP protocol embedded able to connect to the Internet to control processes and provide data for supervision in real time able to communicate with the users through a web interface. The microcontrolled board, based on the PIC18F4620 microcontroller and TCP/IP protocol embedded, offering for the user versatility and low cost. The development of the web pages is done in HTML and CSS codes with JavaScript functions. As an application example, it was designed a microcontrolled printed-circuit board with digital functions of inputs and outputs, pulse-width modulation (PWM) and digital/analog conversion controlled by a web page.

**Keywords:** *Embedded TCP/IP, Activation and Monitoring, Microserver*

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>11</b>
1.1.OBJETIVO GERAL .....	12
1.2.OBJETIVOS ESPECÍFICOS.....	13
1.4.JUSTIFICATIVA .....	14
<b>2. REVISÃO BIBLIOGRÁFICA</b> .....	<b>16</b>
2.1.PILHA TCP/IP .....	17
2.1.1. PROTOCOLO HTTP .....	18
2.1.1.1. REQUISIÇÕES HTTP E MÉTODO GET .....	19
2.1.1.2. O Common Gateway Interface (CGI) .....	20
2.1.2. TEORIA DE OPERAÇÃO DO PROTOCOLO ETHERNET .....	21
2.1.3. PROTOCOLO TCP ( <i>TRANSMISSION CONTROL PROTOCOL</i> ) .....	22
2.1.4. PROTOCOLO IP ( <i>INTERNET PROTOCOL</i> ) .....	24
2.1.5. PROTOCOLO ICMP ( <i>INTERNET CONTROL MESSAGE PROTOCOL</i> ) ....	28
2.1.6. A Pilha TCP/IP da MICROCHIP .....	29
2.2.CONTROLE SUPERVISÓRIO .....	30
2.2.1. Estrutura de um Sistema SCADA.....	31
2.2.2. Componentes Físicos de um Sistema SCADA .....	33
2.2.3. Sistemas SCADA Baseados na Internet.....	34
2.3.A LINGUAGEM HTML .....	36
<b>3. MATERIAIS E MÉTODOS</b> .....	<b>38</b>
3.1. O MICROCONTROLADOR .....	38
3.2. CARACTERÍSTICAS DO PIC18F4620 .....	39
3.2.1. Entradas/Saídas digitais.....	40
3.2.2. Conversor Analógico/Digital .....	41
3.3. O CONTROLADOR ETHERNET .....	42
3.4. A INTERFACE DE REDE.....	44
3.5. O SENSOR DE TEMPERATURA LM35.....	45
3.6. SOFTWARES UTILIZADOS .....	45
3.6.1. MPLAB IDE .....	46
3.6.2. O Compilador MPLAB C18.....	47
3.7. SOFTWARE DO SERVIDOR WEB .....	47
3.7.1. Interpretação pelo Servidor HTTP .....	48
3.7.2. Geração de Páginas Web .....	49
3.7.3. Configuração da Rede .....	50
3.8.DESENVOLVIMENTO DA PLACA DE CIRCUITO IMPRESSO NO SOFTWARE EAGLE® .....	51

<b>4. RESULTADOS .....</b>	<b>56</b>
<b>5. CONCLUSÃO .....</b>	<b>68</b>
<b>6. CONSIDERAÇÕES FINAIS.....</b>	<b>69</b>



## LISTA DE FIGURAS

Figura 1: Arquitetura de Controle com Rede Ethernet .....	13
Figura 2: Camadas do Modelo da Pilha TCP/IP da MICROCHIP® .....	17
Figura 3: Interface Gerada por index.htm .....	20
Figura 4: Cabeçalho do Protocolo TCP .....	23
Figura 5: Cabeçalho do Protocolo IP .....	25
Figura 6: Roteador com NAT .....	28
Figura 7: Cabeçalho do Protocolo ICMP .....	29
Figura 10: Comparação Entre os Serviços TCP/IP com a Referência Modelo .....	30
Figura 8: Sistema SCADA Integrado.....	32
Figura 9: Sistema SCADA Baseado na Internet.....	33
Figura 11: Conexões do ENC28J60 .....	43
Figura 12: Conexão dos Conversores de Nivel .....	44
Figura 13: Módulo Ethernet ENC28J60 da HanRum.....	44
Figura 14: Ambiente de Trabalho do MPLAB IDE v8.92 .....	46
Figura 15: Criação das Imagens com MPFS.exe .....	50
Figura 16: Schematic Produzido para o Projeto .....	51
Figura 17: Board Desenvolvido para o Projeto.....	52
Figura 18: Módulo Microcontrolado Desenvolvido .....	53
Figura 19: Placa Conectada ao Módulo da Hanrum.....	53
Figura 20: Caixa de Distribuição com a Placa .....	54
Figura 21: Vista Frontal da Caixa .....	54
Figura 22: Teste de Ping .....	56
Figura 23: Página Inicial.....	58
Figura 24: Página Acionamento Remoto.....	59
Figura 25: Página PWM .....	60
Figura 26: Duty-cycle de 90% e 10% Obtidos.....	63
Figura 27: Duty-cycle de 50% e 10% Obtidos.....	63
Figura 28: Gráfico de Temperatura .....	65
Figura 29: Diagrama de Blocos da Leitura de Temperatura.....	66

## LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledge
ARP	<i>Ethernet Address Resolution Protocol</i>
CAT3	Categoria 3
CLPs	Controladores Lógicos Programáveis
CGI	Interface comum de porta de ligação (do inglês, <i>Common Gateway Interface</i> )
CSMA/CD	<i>Carrier Sense Multiple Access With Collision Detection</i>
CSS	Cascade Style Sheet
DHCP	<i>Dynamic Host Configuration Protocol</i>
FTP	<i>File Transfer Protocol</i>
HTTP	<i>Hypertext Transfer Protocol</i>
ICMP	<i>Internet Control Message Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IHM	Interface Homem Máquina
IP	<i>Internet Protocol</i>
LAN	<i>Local Area Network</i>
LED	Diodo emissor de luz (do inglês, <i>Light Emissor Diode</i> )
MCU	Microcontrolador
MAC	<i>Media Access Control</i>
MPFS	<i>Microchip File System</i>
ODVA	<i>Open DeviceNet Vendors Association</i>
RAM	<i>Random Access Memory</i>
RFC	<i>Request for Coments</i>
ROM	<i>Read-Only Memory</i>
SNMP	<i>Simple Network Management Protocol</i>
TCP	<i>Transmission Control Protocol</i>
TFTP	<i>Trivial File Transfer Protocol</i>
TMR3	Timer 3
UDP	<i>User Datagram Protocol</i>
SPI	<i>Serial Periferal Interface</i>

## 1. INTRODUÇÃO

As necessidades de automação na indústria e nos mais diversos processos geralmente estão ligadas às complexidades e variações das operações, requerendo um grande número de controles e mecanismos de regulação para permitir decisões mais ágeis e, portanto, aumentar os níveis de produtividade e eficiência do processo produtivo.

Desta forma, torna-se mais notória a necessidade do desenvolvimento de processos produtivos automatizados, cujo fluxo de informações passe do chão de fábrica aos sistemas corporativos de forma fácil e sem problemas de integração. Essas tais necessidades vêm ao encontro do crescente uso da tecnologia de supervisão e controle pela Internet.

Atualmente os serviços de controle envolvem o intercâmbio de dados de tempo crítico entre os dispositivos como Controladores Lógicos Programáveis, *Soft-Starters* ou placas microcontroladas, para acionamentos de motores com velocidade variável, acionamento em rampa de tensão, inversão de rotação, leitura de sensores, controle de atuadores, além de estes estarem interligados em um sistema de controle supervisório.

Existem várias possibilidades de uso da automação em um processo produtivo. Pode-se, por exemplo, realizar o monitoramento e controle da iluminação, climatização, segurança e das variáveis do processo em si (temperatura, velocidade, corrente, tensão, etc.).

Uma das principais vantagens do uso da automação é a possibilidade de que este controle e monitoramento possa ser feito à distância. Em situações nas quais as pessoas esqueceram-se, por um motivo qualquer, de desligar alguma máquina ou há uma falha na máquina que venha a interferir no resultado final do processo ou a danificação permanente de algum equipamento, a automação remota também é de grande utilidade. Estas atividades podem ser realizadas automaticamente por meio de um sistema de automação com agendamento de tarefas, seguindo assim um roteiro pré-estabelecido pelo operador ou usuário, comandando diversas variáveis do processo, dando assim, a impressão de que uma pessoa estivesse observando o processo. O intuito é facilitar a vida das pessoas, liberando mais tempo que pode ser dedicado para outras tarefas.

Dessa forma, o presente trabalho propõe a implementação de um dispositivo microcontrolado conectado em uma rede Internet capaz de controlar processos e fornecer dados para supervisão em tempo real através de uma página *web*. Fornecendo uma arquitetura de automação que permite a coleta de dados para fins de exibição em uma interface interativa e que apresente análises e tendências, e/ou solução de problemas e manutenção.

O trabalho visa iniciar os estudos para o desenvolvimento de um sistema microcontrolado capaz de observar e controlar grandezas físicas como temperatura de um determinado lugar ou acionamento à distância com uma placa microcontrolada, e disponibilizar a supervisão e modificação dos parâmetros deste controle através de uma rede Ethernet utilizando um navegador padrão de Internet.

O projeto está baseado no microcontrolador (MCU) PIC18F4620 e a pilha<sup>1</sup> TCP/IP (Transmission Control Protocol/Internet Protocol) versão 3.75 da Microchip®, que oferece o conjunto de protocolos necessários a implementação de uma rede *Ethernet*. Para exemplificar, torna-se necessário a descrição das configurações do modo de funcionamento da pilha e o desenvolvimento de uma placa de circuito impresso com funções de entradas e saídas digitais, modulação por largura de pulso (PWM) e conversões analógica/digital (AD) controladas por uma página *web*.

## 1.1.OBJETIVO GERAL

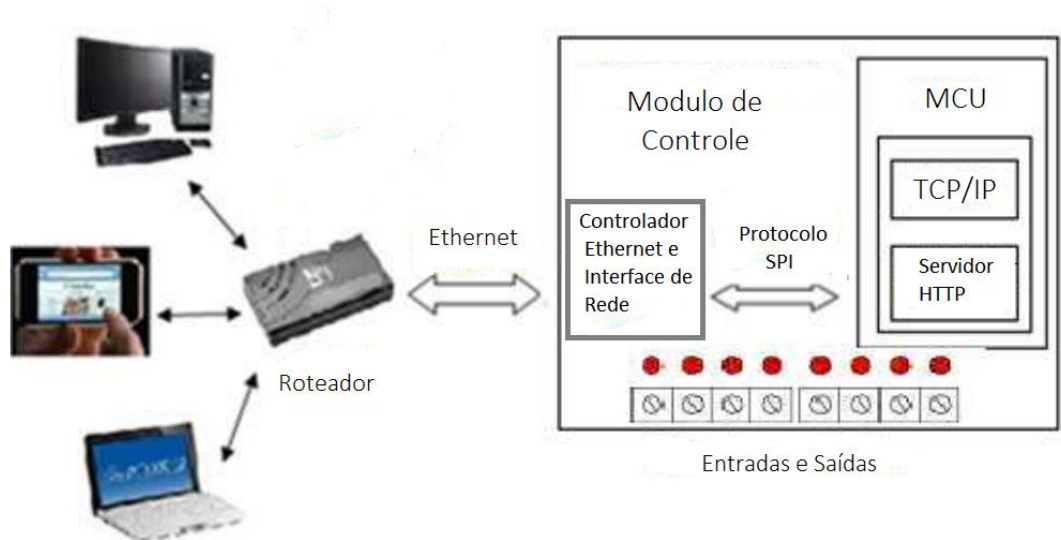
O objetivo deste projeto é o desenvolvimento de um dispositivo microcontrolado conectado em uma rede *Ethernet* capaz de controlar funções e fornecer dados para supervisão em tempo real (Figura 1), como também determinar o *hardware* e configurações do *software* embarcado no microcontrolador (MCU) necessárias para criar uma rede de comunicação *Ethernet* com um computador.

Foi desenvolvido um serviço de comunicação via *web* que é responsável por guardar as páginas *web* no MCU. Tais páginas apresentam a interface onde

---

<sup>1</sup> Pilha é a tradução da palavra inglesa Stack, e neste caso, significa um conjunto de arquivos ordenados

serão controladas as funções como acionar LEDs, utilizar as funções PWM e as funções das conversões analógica/digital do MCU.



**Figura 1: Arquitetura de Controle com Rede Ethernet**  
**Fonte: Autoria Própria**

## 1.2. OBJETIVOS ESPECÍFICOS

O desenvolvimento desse trabalho se configura nos seguintes objetivos específicos:

- Analisar e descrever os principais protocolos implementados pela pilha TCP/IP v3.75, disponibilizada pela MICROCHIP®.
- Descrever as configurações do software da pilha necessárias para funcionamento do servidor *web* embarcado no MCU.

## 1.3. ESTRUTURA DO TRABALHO

O primeiro capítulo apresenta um resumo sobre a importância dos dispositivos embarcados com possibilidade de acesso a rede Ethernet, assim como as justificativas que levaram ao desenvolvimento deste trabalho.

O segundo capítulo faz uma revisão bibliográfica que abrange desde o desenvolvimento das redes Ethernet e da Internet, além da descrição do modelo de referência TCP/IP (comparando-o com o modelo da pilha TCP/IP da Microchip) e um resumo das funções das camadas do modelo de referência.

O terceiro capítulo traz a implementação do projeto, com a programação da página web e adaptação da pilha TCP/IP da Microchip® à aplicação. É descrito também os softwares utilizados no projeto, assim como a estruturação física do hardware relacionando os componentes utilizados para montagem do mesmo.

O quarto capítulo apresenta os materiais e métodos utilizados.

O quinto capítulo explica o funcionamento da aplicação web junto ao módulo embarcado, além da descrição do funcionamento do módulo controlado a cada comando enviado. No sexto e sétimo capítulo encontra-se a conclusão do trabalho e a sugestão para trabalhos futuros.

#### 1.4. JUSTIFICATIVA

Com a diminuição de preço dos microcontroladores e o aumento da capacidade de processamento dos mesmos, além da crescente utilização de equipamentos como sensores, controladores lógicos programáveis e atuadores que dispõem de algum protocolo de comunicação em rede, surgem cada vez mais processos que precisam ser automatizados a um baixo custo. Destaca-se assim, a tecnologia Ethernet e a rede global Internet, hoje disponíveis em praticamente todos os sistemas operacionais (CARVALHO, 2005).

As empresas de manufatura estão cada vez mais expandindo suas operações globais para enfrentar novas oportunidades e reduzir custos operacionais. Essas estão procurando melhorar continuamente a eficiência e reduzir os custos de instalações e processos existentes. Os fabricantes precisam melhorar a resposta a eventos que ocorrem no chão de fábrica,

independentemente do local, com operações mais flexíveis e ágeis para reagir às rápidas mudanças das condições de mercado.

Já os sistemas que possuem alta conectividade e acesso via Ethernet são, em geral, muito caros para aplicação em pequenas plantas, inviabilizando seu uso. E é esta lacuna entre os robustos sistemas supervisórios com conexão *web* e as pequenas plantas industriais que a tecnologia Internet embarcada vem preencher com baixo custo e grande facilidade de implantação.

Uma instalação típica de uma rede Ethernet pode se estender para toda a planta industrial e ser conectado à rede mundial através da Internet. Funções como enviar dados para os sistemas, servir páginas *web* da intranet, realizar o controle de supervisão e fornecer conectividade para interfaces de operação, eventos e alarmes exigem um alto rendimento e fácil acesso, características que a tecnologia Ethernet oferece.

Cada vez mais é possível ter uma rede Ethernet com conexão a Internet segura e cada vez mais confiável, desenvolvendo-a para fins de controle de processos. Com a necessidade de um maior compartilhamento de dados e um melhor tempo de resposta entre as informações compartilhadas, ela tem sido a alternativa cada vez mais utilizada como protocolo de comunicação no setor industrial.

## 2. REVISÃO BIBLIOGRÁFICA

O interesse em se conectar pequenos dispositivos a uma rede *Ethernet* já existente (como a Internet global), tem aumentado consideravelmente. A fim de ser capaz de comunicar através da Internet, é necessária a implementação da pilha de protocolo TCP/IP.

Com o sucesso da Internet, o conjunto de protocolos TCP/IP tornou-se um padrão global de comunicação.

A pilha completa TCP/IP consiste de vários protocolos que vão desde protocolos de baixo nível, como ARP<sup>2</sup> (*Address Resolution Protocol* – que traduz endereços IP para endereços MAC<sup>3</sup> – *Media Access Control*), como para protocolos de nível de aplicação, tais como HTTP, que é utilizado para transferir páginas da *web* (RAJBHARTI, 2002).

O sucesso da internet e o bom funcionamento das redes devem-se principalmente no desenvolvimento e aperfeiçoamento do protocolo de comunicação de computadores TCP/IP. Esse protocolo de comunicação melhorou bastante a capacidade de transmissão e segurança. O que existe atualmente é uma rede rápida, a nível mundial, e única baseada em padrões abertos (ZIMMER, 2014).

Os sistemas embarcados, sendo capaz de executar o TCP/IP, podem ser ligados diretamente à Internet. A pilha TCP/IP permite que um microcontrolador se conecte a internet como se fosse um computador, a fim de estabelecer a conexão lógica com uma máquina remota em um determinado segmento de rede.

Descrever os principais protocolos da pilha torna-se necessário para melhor compreensão do funcionamento da mesma, podendo assim retirar as melhores funcionalidades para a aplicação desenvolvida.

---

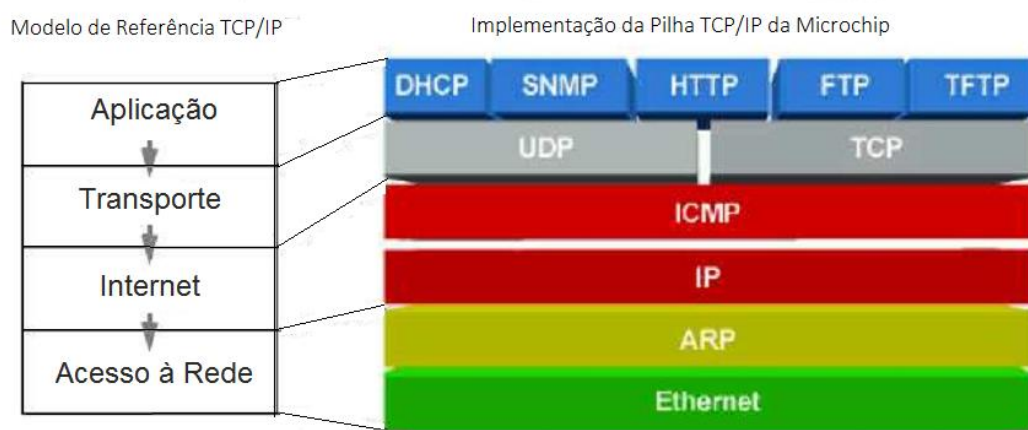
<sup>2</sup> Address Resolution Protocol – Protocolo responsável por fazer a associação do endereço IP da rede local com o endereço MAC (Media Access Control) do dispositivo. Cada dispositivo possui um endereço MAC único.

<sup>3</sup> O endereço MAC é um identificador único de 48 bits designado para a maioria dos dispositivos, traz a informação de seus respectivos fabricantes para fins de identificação.



## 2.1. PILHA TCP/IP

A sigla TCP/IP identifica a família de protocolos utilizada na rede Internet e nas intranets. Os protocolos presentes na família encontram-se agrupados em quatro camadas: camada de aplicação, camada de transporte, camada internet e camada de acesso à rede (Figura 2).



**Figura 2: Camadas do Modelo da Pilha TCP/IP da MICROCHIP®**  
Fonte: MICROCHIP®, 2015.

A seguir é apresentado um breve resumo das funções reservadas a cada camada no modelo de referência TCP/IP (MICROCHIP®, 2011. COSTA, 2015):

- Camada de Aplicação: Camada que a maioria dos programas de rede usa para se comunicar através da rede. A camada define os protocolos de aplicativos, os quais provêm serviços que suportam diretamente aplicações do usuário. O pacote relacionado à camada de aplicação é chamado mensagem ou dados. Os protocolos mais comuns que nela trabalham são o HTTP( Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), o FTP (File Transfer Protocol), o SNMP (Simple Network Managment Protocol), o DNS (Domain Name System) e o Telnet.
- Camada de Transporte: Onde é determinado para qual aplicação um dado qualquer é destinado. Camada responsável pela garantia da entrega de

dados e da integridade dos mesmos. Sua finalidade é colher os dados enviados pela camada de aplicação e transformá-los em pacotes, que serão repassados na Internet. Os protocolos que operam nessa camada são o TCP (Transmission Control Protocol) e o UDP (User Datagram Protocol).

- Camada de Internet: A tarefa dessa camada é entregar pacotes onde for necessário, ou seja, ela permite que dispositivos *host* enviem pacotes em qualquer rede e deve garantir que eles trafeguem independentemente até o destino. Os protocolos que operam nesta camada são o IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol) e RARP (Reverse Address Resolution Protocol).
- Camada de Acesso à Rede: Essa camada tem a responsabilidade de detalhar o modo como os dados serão enviados fisicamente pela rede, trata das características elétricas e mecânicas do meio, como tipos de conectores e cabos utilizado para estabelecer uma comunicação.

### 2.1.1. PROTOCOLO HTTP

O protocolo de transferência de hipertexto HTTP (*Hypertext Transfer Protocol*) é um protocolo da camada de aplicação. É utilizado em varias tarefas, tais como servidores, sistema de gerenciamento de objetos, através de seus métodos de requisição HTTP (BEOK et. al., 2015).

Geralmente o HTTP usa uma relação cliente-servidor e é muito utilizado para transferência de documentos HTML entre servidores da internet e um cliente *web*. A maneira mais simples para que um cliente estabeleça uma comunicação com um servidor é enviando o identificador único do dado, que especifica o caminho do arquivo a ser interpretado pelo servidor, transferindo assim os dados deste arquivo para o cliente (ABUSAR, 2015).

Como exemplo, as Intranets, que se caracterizam pelo uso da tecnologia *web*, onde os usuários acessam as informações armazenadas em bancos de dados e nos próprios servidores *web* através de navegadores. Os navegadores são instalados nas máquinas dos usuários e através deles é possível apresentar

os documentos transferidos a partir dos servidores *web*. Entre os navegadores mais populares temos o Internet Explorer e o Chrome. (BEOK et. al., 2015).

#### 2.1.1.1. REQUISIÇÕES HTTP E MÉTODO GET

O HTTP define um conjunto de mensagens chamadas de métodos de requisição HTTP, que são usados pelo cliente para enviar uma requisição ao servidor.

São definidos oito métodos de requisição HTTP, para indicar a ação a ser realizada por cada recurso específico (BEOK et. al., 2015). As funções são:

- GET – Solicita, integral ou parcialmente, algum recurso ou informação;
- HEAD – Solicita os cabeçalhos de algum recurso;
- POST – Envia dados para serem processados pelo recurso indicado;
- PUT – Envia um determinado recurso;
- DELETE – Exclui um determinado recurso;
- TRACE – Ecoa uma solicitação para fins de depuração;
- OPTIONS – Recupera os métodos HTTP aceitos pelo servidor;
- CONNECT – Utilizado para criar, por exemplo, uma conexão segura.

Os dois métodos mais comuns para troca de dados são o *GET* e o *POST*. O método GET tem a finalidade de recuperar informações do servidor. Definido como um método seguro e que produz o mesmo resultado se repetido várias vezes. O GET somente envia os dados através do cabeçalho, como o URL<sup>4</sup>(*Uniform Resource Locators*), enquanto o método *POST* define que os dados são transferidos através de entradas e saídas padronizadas, que além do cabeçalho envia anexos como arquivos, planilhas (LCJV, 2007).

O método GET quando envia dado, adiciona-os no URL. Por exemplo, ao descrevermos um arquivo *index.htm* com o seguinte código HTML:

```
<html>
<body>
<form method="get">
<input name="comando1" type="hidden" value="valor1" />
```

---

<sup>4</sup> Um URL é uma sigla que se refere ao mapeamento de um endereço da rede Internet.

```
<input name="comando2" type="hidden" value="valor2" />
<input type="submit" />
</form>
</body>
</html>
```

Ao abrir o arquivo index.htm em um browser web a imagem da Figura 3 é gerada.



**Figura 3: Interface Gerada por index.htm**  
**Fonte: Autoria Própria**

Ao clicar no botão *Enviar*, que é *GET* ("`<form method="get">`"), o seguinte cabeçalho HTTP é enviado para o servidor:

```
GET /index.htm?comando1=valor1&comando2=valor2 HTTP/1.1
```

É o mesmo que digitar a URL (`http://site/index.htm`) no browser, ou seja, toda vez que for solicitada alguma informação o servidor irá interpretar estes dados. Com esta interface, o cliente HTTP pode invocar uma função e receber os resultados sob a forma de uma página *web* (LCJV, 2007).

#### 2.1.1.2. O Common Gateway Interface (CGI)

O CGI é uma interface que facilita o uso de aplicações *web* em qualquer linguagem de programação com suporte para leitura de variáveis, interpretação de dados e retorno de resultados. O CGI permite requisitar dados de um programa executado no servidor pelo cliente através do navegador.

Pode ser descrita em C/C++, Java, Python, Visual C/C++, dependendo do sistema operacional usado no servidor. Possibilita a execução de programas sob um servidor de informações, geralmente na forma de scripts, que recebem requisição de informação, e retornam um documento ou respostas com os resultados correspondentes (ROCHA, 2014).

Um servidor *web* capaz de implementar a especificação CGI é capaz de executar aplicações solicitadas remotamente (browser). Pode ser qualquer aplicação. O uso mais comum do CGI é como intermediário para aplicações mais complexas, principalmente de acesso e tratamento de dados dinamicamente. Por exemplo, suponha que um servidor *web* instalado em uma máquina ou rede que controlem o acionamento e desligamento de um motor. Se for possível construir uma aplicação local nessa máquina ou rede que ligue e desligue o motor, será possível construir uma aplicação CGI e instalá-la no servidor *web* para que um cliente remoto, através de uma página recebida em seu browser, possa também acionar e desligar o motor remotamente (ROCHA, 2014).

Servidores que suportam CGI podem ser configurados a executarem programas externos ao servidor, passar-lhes parâmetros recebidos de uma requisição do browser e redirecionar sua saída como uma resposta HTTP.

### 2.1.2. TEORIA DE OPERAÇÃO DO PROTOCOLO ETHERNET

Tem havido muita discussão recentemente sobre a aplicabilidade do uso de Ethernet em vários níveis da hierarquia de controle. O Ethernet é tão presente no escritório ou é frequentemente usado como a rede da empresa para controle ao nível corporativo e que está surgindo cada vez mais no nível de controle supervisorio ou até mesmo no nível do dispositivo. Os argumentos para a sua aplicação incluem baixo custo, boa conectividade e simples migração para redes de maior velocidade.

Ethernet é um protocolo de enlace de dados e da camada física definido pela especificação IEEE 802™. O protocolo se apresenta de vários modos, sendo os principais definidos pela taxa máxima de transmissão de bits, pelo modo de transmissão e pelo meio de transmissão física adotados (RAJBHARTI, 2002).

Este protocolo define o cabeamento e o modo de transmissão dos sinais elétricos na camada física, o formato de pacotes e os protocolos adotados para a camada de controle de acesso ao meio, e é definida em três principais camadas: Controle de Link Lógico – CLL (IEEE 802.2), Controle de Acesso ao Meio – MAC (IEEE 802.3), e Camada Física (ROCHA, 2014).

- Controle do Link Lógico (LLC, IEEE 802.2): inclui informações dos protocolos de nível mais alto como o TCP/IP ou uma aplicação, que entrega o pacote de dados a ser transmitido. Com isso, a máquina receptora tem como saber para qual protocolo de nível mais alto ela deve entregar os dados, por exemplo, o TCP/IP.
- Controle de Acesso ao Meio (MAC, IEEE 802.3): Monta o quadro de dados a ser transmitido pela camada física, incluindo cabeçalhos próprios dessa camada (como o endereço MAC), aos dados recebidos da camada de Controle do Link Lógico.
- Física: Transmite os quadros entregues pela camada de Controle de Acesso ao Meio usando o método CSMA/CD. Define as características dos equipamentos utilizados para a rede e a especificação para conectores e pinos padrões.

Os dados são enviados pela rede através de um meio de transmissão escolhidos, que pode ser por cabos (UTP<sup>5</sup>, STP<sup>6</sup> ou Fibra Óptica), ou através do ar (*Wireless*) (CALVERT; DONAHO, 2002).

A escolha do modo de transmissão influi diretamente na qualidade e na segurança dos dados transmitidos, sendo a transferência usando cabos mais segura e com maior taxa de transmissão de bits do que através do ar.

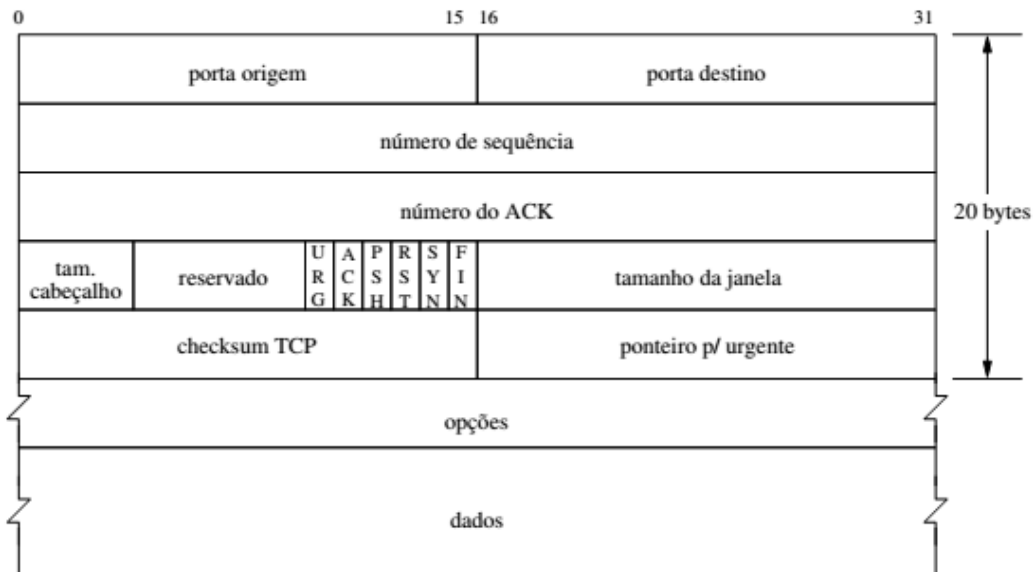
### 2.1.3. PROTOCOLO TCP (*TRANSMISSION CONTROL PROTOCOL*)

TCP é um protocolo de camada de transporte padrão descrito na RFC 793 (1981). Fornece fluxos confiáveis em redes não confiáveis (não determinística), e fornece a base para os protocolos HTTP, SMTP, e outros. O cabeçalho TCP é apresentado na Figura 4.

---

<sup>5</sup> UTP é abreviação das palavras inglesas Unshielded Twisted Pair (Par Trançado sem Blindagem).

<sup>6</sup> STP é a abreviação das palavras inglesas Shielded Twisted Pair (Par Trançado Blindado).



**Figura 4: Cabeçalho do Protocolo TCP**  
**Fonte: PEREIRA, 2003.**

A descrição do cabeçalho TCP possui as seguintes características (COSTA, 2015):

- Porta origem e porta destino: Identificam os pontos terminais da conexão.
- Número de sequência: Identifica a posição deste segmento no fluxo de dados e cada conexão possui um fluxo de dados particular.
- Número de ACK: Utilizado para confirmar o recebimento de segmentos enviados anteriormente e especifica o próximo segmento aguardado.
- Tamanho do cabeçalho: Tamanho do cabeçalho TCP (números de palavras de 32 bits).
- Tamanho da janela: Indica o tamanho (disponível) do buffer do receptor e é usado pelo receptor para indicar ao transmissor que diminua o fluxo de transmissão de dados.
- *Checksum* TCP: Verificação de erros.
- *Urgent pointer* : Usado pela origem para indicar onde se encontra algum dado urgente dentro do segmento opções.

As conexões TCP são feitas em um processo de três vias, estabelecimento da conexão pelo emissor, a confirmação da conexão pelo receptor e o envio dos dados, garantido uma conexão de um aplicativo emissor

até o aplicativo receptor. Assim que um nó transmite algum dado, nós remotos anunciam a quantidade de dados que eles estão prontos para receber, e todos os dados enviados devem ser reconhecidos, senão ele é automaticamente retransmitido, garantindo que todos os erros de rede, como pacotes perdidos, danificados ou fora de ordem sejam corrigidos (DUNKELS, 2003; RFC 793, 1981).

Para que o processo de transmissão ocorra em sua totalidade o TCP deve operar em um *buffer*. Uma vez que não tenha mais espaço para receber dados, ocorre a interrupção da transmissão dos dados até que o nó remoto reconheça o recebimento.

Para a pilha TCP/IP da Microchip<sup>®</sup>, a rotina deve retornar ao *loop*<sup>7</sup> principal da pilha para que as funções, responsáveis pela verificação da chegada de pacotes de dados, leiam os dados e limpe o *buffer*. Da mesma forma se a rotina não retornar que recebeu os dados, o nó remoto não pode transmitir mais dados até que o dispositivo local tenha informado a recepção e o *buffer* esteja livre (RAJBHARTI, 2002).

E pode-se compreender TCP como sendo a conexão entre dois aplicativos que estão em dispositivos diferentes conectados à rede. Os dados são transmitidos pela conexão e ambos os lados devem manter o controle das informações enviadas e recebidas, para que se possa detectar as falhas ou as duplicações no fluxo de dados (DUNKELS, 2003).

#### 2.1.4. PROTOCOLO IP (*INTERNET PROTOCOL*)

O IP foi elaborado como um protocolo que somente fornece as funções necessárias para enviar um pacote de uma origem a um destino por um sistema de redes. O IP não foi elaborado para rastrear e gerenciar o fluxo dos pacotes. Estas funções são realizadas por outros protocolos das outras camadas.

As partes de informação são identificadas por endereços de comprimento fixo denominado endereço IP, como não há uma garantia de uma transmissão de dados confiável de ponta a ponta, o cabeçalho IP é protegido por uma soma de

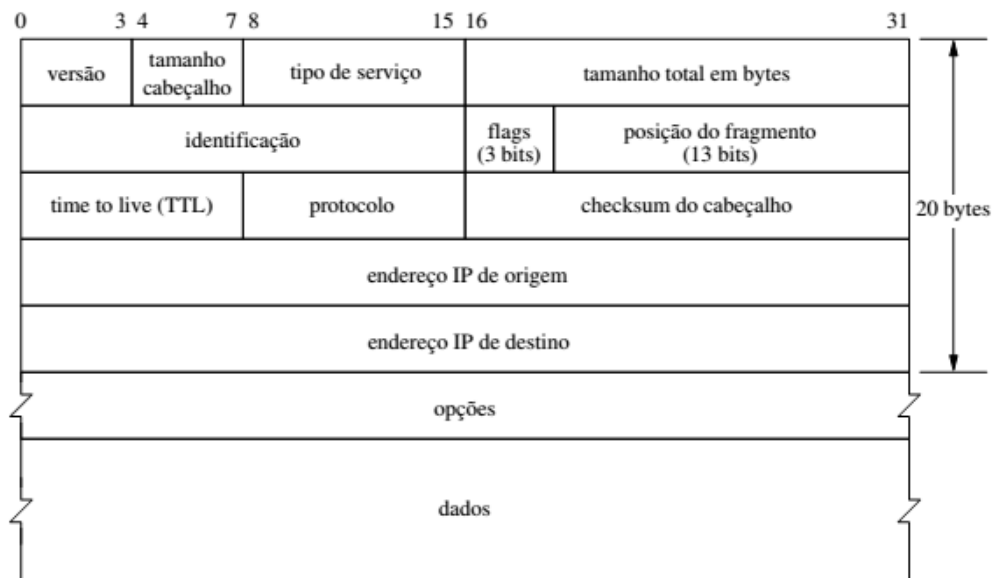
---

<sup>7</sup> Loop é uma palavra que significa um código de programa que se repete.



verificação e cada pacote IP possui um campo do protocolo indicando qual protocolo da camada superior é utilizado para transmissão dos dados (TCP ou UDP) (RFC 791, 1981).

A Figura 5 apresenta o cabeçalho do protocolo IP.



**Figura 5: Cabeçalho do Protocolo IP**  
**Fonte: PEREIRA, 2003.**

As definições do cabeçalho IP são os seguintes (GUERBER, 2007):

- Versão - Contém o número da versão IP (4 para IPv4 e 6 para IPv6).
- Tamanho do Cabeçalho - Especifica o tamanho do cabeçalho do pacote.
- Tipo de Serviço - O campo Tipo de Serviço contém um valor binário de 8 bits que é usado para determinar a prioridade de cada pacote. Este valor permite que um mecanismo de Qualidade de Serviço (QoS) seja aplicado aos pacotes com alta prioridade. O roteador que processa os pacotes pode ser configurado para decidir qual pacote será encaminhado com base no valor do Tipo de Serviço.
- Tamanho Total - Este campo fornece o tamanho total do pacote em bytes, incluindo o cabeçalho e os dados.
- Identificação - Este campo é usado principalmente para identificar unicamente os fragmentos de um pacote IP original.

- *Flag*<sup>8</sup> Mais Fragmentos - A *flag* Mais Fragmentos (MF) é um único bit no campo *Flag* usado com o Deslocamento de Fragmentos na fragmentação e reconstrução de pacotes. O bit da *flag* Mais Fragmentos é configurado, o que significa que ele não é o último fragmento de um pacote. Quando um host de destino identifica um pacote com MF = 1, ele examina o Deslocamento de Fragmentos para ver onde este fragmento deve ser colocado no pacote reconstruído. Quando um host de destino recebe um quadro com MF = 0 e um valor diferente de zero no Deslocamento de Fragmentos, ele designa este fragmento como a última parte do pacote reconstruído. Um pacote não fragmentado possui todas as informações de fragmentação iguais a zero (MF = 0, deslocamento de fragmentos = 0).
- *Flag* Não Fragmentar - A *flag* Não Fragmentar (DF) é um único bit no campo *Flag* que indica que a fragmentação do pacote não é permitida. Se o bit da *flag* Não Fragmentar for configurado, a fragmentação do pacote não será permitida. Se um roteador precisar fragmentar um pacote para permitir que ele passe para a camada de enlace de dados e o bit DF estiver definido como 1, o roteador descartará o pacote.
- *Bit R* – Bit reservado, tem que ser 0.
- Deslocamento de Fragmento - Um roteador pode precisar fragmentar um pacote ao encaminhá-lo de um meio físico para outro. Quando ocorre a fragmentação, o pacote IPv4 usa o campo Deslocamento de Fragmento e a *flag* MF no cabeçalho IP para reconstruir o pacote quando ele chega ao host de destino. O campo deslocamento de fragmento identifica a ordem na qual o fragmento do pacote deve ser colocado na reconstrução.
- Time to Live (TTL) - O TTL é um valor binário de 8 bits que indica o "tempo de vida" restante do pacote. O valor TTL diminui em pelo menos um a cada vez que o pacote é processado por um roteador. Quando o valor chega a zero, o roteador descarta ou abandona o pacote, ou seja, ele é removido do fluxo de dados da rede. Este mecanismo evita que os pacotes que não conseguem chegar a seus destinos sejam encaminhados indefinidamente entre roteadores em um loop de roteamento. Se os loops de roteamento tivessem permissão para continuar, a rede ficaria congestionada com os

---

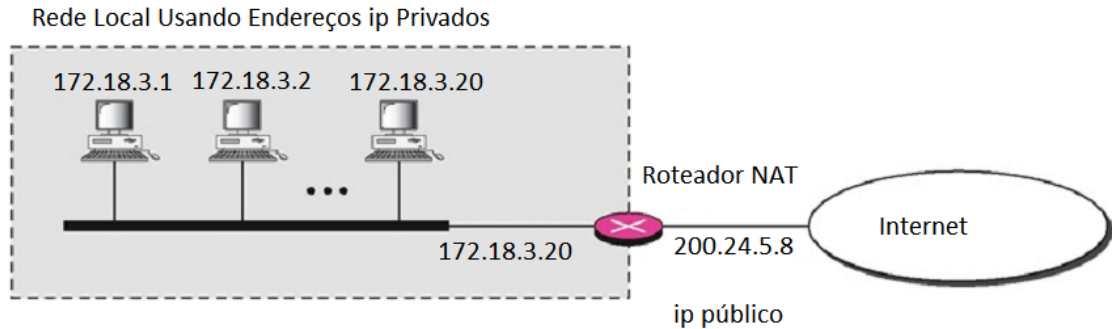
<sup>8</sup> Flag é uma palavra inglesa e significa bandeira, sinalizador.

pacotes de dados que nunca chegariam a seus destinos. A diminuição do valor de TTL a cada salto assegura que ele chegue a zero e que o pacote com um campo TTL expirado seja descartado.

- Protocolo - O valor binário de 8 bits indica o tipo de valor dos dados que o pacote está carregando. O campo Protocolo possibilita que a camada de rede passe os dados para o protocolo apropriado das camadas superiores. Alguns exemplos de valores: 01 para ICMP - 06 para TCP - 17 para UDP.
- *Checksum* do Cabeçalho - O campo de checksum é usado para a verificação de erros no cabeçalho do pacote.
- Endereço IP de Origem - O Endereço IP de Origem contém um valor binário de 32 bits que representa o endereço do host de origem do pacote.
- Endereços IP de Destino - O Endereço IP de Destino contém um valor binário de 32 bits que representa o endereço do host de destino do pacote.
- Opções – Disponíveis para funções adicionais no cabeçalho IPv4, mas eles raramente são utilizados.

O IP usa endereços que identificam o dispositivo que originou o pacote e o dispositivo ao qual o pacote se destina. Estes endereços são denominados endereços IP, têm 32 bits e são divididos em três campos: identificador da classe de endereços, endereço da rede e endereço local da máquina.

Os endereços IP são organizados em três conjuntos de endereços ip privados. O primeiro é destinado a grandes redes, que possui os endereços 10.0.0.0 a 10.255.255.255, o segundo para redes com tamanho moderado inclui os endereços 172.16.0.0 a 172.31.255.255 e o terceiro para redes pequenas do endereço 192.168.0.0 a 192.168.255.255. Para que estas redes possam estar interligadas à internet precisam passar por um roteador com software que será responsável pela conversão dos endereços de rede local para endereços públicos (Figura 6). A solução para isso é a chamada NAT (*Network Address Translation*) - tradução de endereços de rede (ROCHA, 2014).



**Figura 6: Roteador com NAT**  
**Fonte: Autoria Própria**

Uma rede local deve ter apenas uma única conexão para a Internet por meio de um roteador que roda o software NAT. Neste projeto o dispositivo terá um endereço IP local, neste caso 192.168.1.20. Ao registrar um domínio DNS como projetoiot.ddns.net para acessar o dispositivo pela Internet, basta configurar na rede local para que toda solicitação com endereço projetoiot.ddns.net seja direcionada para o IP 192.168.1.20.

#### 2.1.5. PROTOCOLO ICMP (*INTERNET CONTROL MESSAGE PROTOCOL*)

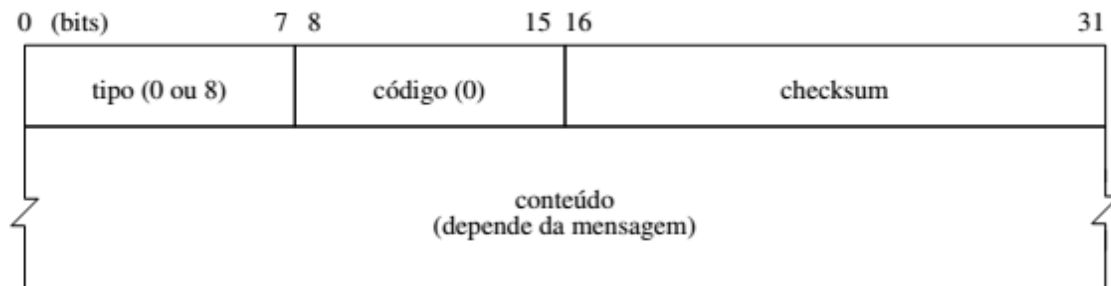
O protocolo de controle de transmissão ICMP é um protocolo de camada de rede da pilha TCP/IP. É muito seguro e orientado a uma conexão host para host, é um dos mais aceitos para a internet. (RFC 792, 1981)

Na pilha TCP/IP, o ICMP é normalmente usado para enviar e receber status de erros, sendo mais útil para fins de diagnóstico da conexão. Ativa-se o mesmo para responder aos testes de “ping”, para assim identificar a presença do servidor por parte do cliente ou host remoto (RAJBHARTI, 2002).

A Figura 7 apresenta o cabeçalho ICMP, e possui as seguintes características:

- Tipo: Tem valor 0 para envio de mensagem de *ping*. E valor 8 para resposta a uma mensagem de *ping*.
- Código: 0.
- *Checksum*: Soma de verificação;

Há varias outras formas de utilização do protocolo ICMP mas somente as requisições de *ping* serão descritas aqui.



**Figura 7: Cabeçalho do Protocolo ICMP**  
Fonte: PEREIRA, 2003.

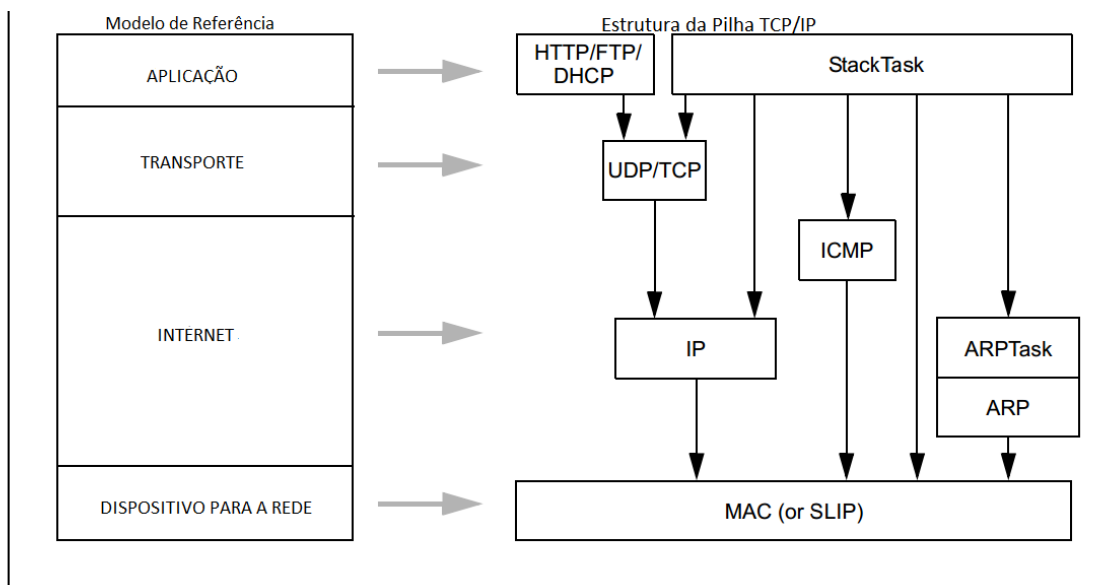
Com o teste de “ping” é possível obter várias informações referentes à conexão com a Internet como o tempo necessário para enviar um comando e receber uma resposta na conexão (latência), o tempo de vida (TTL) do dado enviado e obter a quantidade de pacotes perdidos na transmissão durante o teste. Estas informações são muito importantes e devem ser levadas em conta no desenvolvimento de um sistema de monitoramento *web* pela Internet.

#### 2.1.6. A Pilha TCP/IP da MICROCHIP

A empresa Microchip® oferece uma pilha TCP/IP livre, otimizada para microcontroladores. A implementação da pilha é dividida em várias camadas, em que cada camada acessa serviços de uma ou mais camadas diretamente adjacentes à mesma. Por especificações, muitas das camadas TCP/IP são em tempo real, no sentido de que elas não só agem quando um serviço é solicitado, mas também quando eventos como tempo limite ou nova chegada de pacotes ocorre (RAJBHARTI, 2002).

A pilha da Microchip® (Figura 10), possui módulos de suporte para desenvolvimento de aplicações incluindo HTTP (*Hypertext Transfer Protocol*), o SMTP (*Simple Mail Transport Protocol*) e o SNMP (*Simple Network Management Protocol*).

Além disso, a pilha inclui implementações leves e de alto desempenho para os protocolos TCP e UDP, como também, módulos de suporte como IP e o ICMP. A maioria das funcionalidades da camada física, como *Media Access Control* (MAC) é fornecida pelos chips utilizados com a pilha, como por exemplo, o controlador Ethernet ENC28J60 (MICROCHIP®, 2011).



**Figura 8: Comparação Entre os Serviços TCP/IP com a Referência Modelo**  
**Fonte: RAJBHARTI, 2002**

## 2.2. CONTROLE SUPERVISÓRIO

O controle supervisão é um termo geral para a definição do controle de muitos controladores individuais ou de um circuito de controle, quer seja por um sistema de controle automático ou manual, embora quase todos os sistemas reais sejam uma combinação de ambos (automático e manual). Um uso mais específico do termo é a denominação “sistema SCADA” (*Supervisory Control and Data Acquisition* - Controle Supervisório e Aquisição de Dados).

No controle supervisão um operador humano monitoriza um sistema complexo e intermitentemente executa algum nível de controle sobre um processo. Durante o controle de supervisão, um operador pode planejar uma

atividade que é mediada pelo computador, instrui o computador através de uma série de comandos para executar o plano desejado, em seguida, monitora para assegurar que a ação é executada, intervém quando o computador quer cometer um erro ou precisa da informação de novos parâmetros.

Um sistema de controle de supervisão e aquisição de dados (SCADA) refere-se a um sistema de controle para a indústria; é um sistema de automação de processos, que é utilizado para reunir dados de sensores e instrumentos localizados em locais remotos para transmitir dados a um site central tanto para fins controle ou monitoramento. Os dados coletados são geralmente vistos em um ou mais computadores centrais (host SCADA). Com base nas informações recebidas pelas estações remotas, comandos automatizados ou comandos de supervisão baseados em um operador podem ser transmitidas aos dispositivos de controle da estação remota, que são muitas vezes denominados como dispositivos de campo (STOUFFER et. al., 2006).

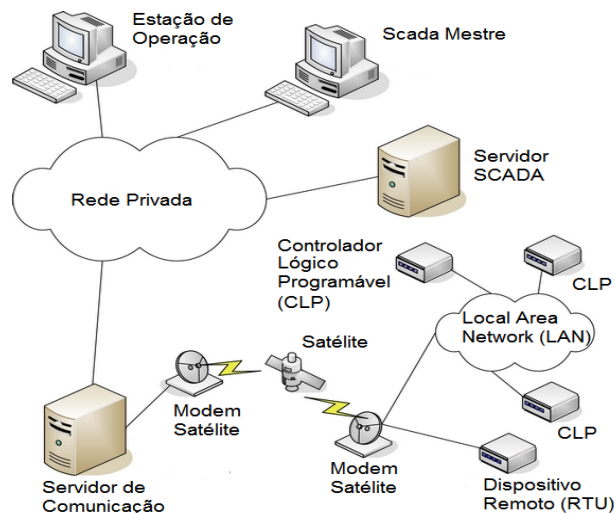
### 2.2.1. Estrutura de um Sistema SCADA

O sistema SCADA geralmente consiste dos seguintes subsistemas (SILVA; SALVADOR, 2005):

- A Interface Homem-Máquina ou IHM: é o dispositivo que apresenta os dados de processo para um operador humano em uma interface gráfica e através da qual monitora e controla o processo.
- Um Sistema de Supervisão (computador): reúne os dados obtidos sobre o processo e responsável também por envio de comandos de controle para o processo.
- Unidades Terminais Remotas (UTRs): Estes se conectam aos sensores no processo, convertendo os sinais do sensor em dados digitais e os envia para o sistema de supervisão.
- Dispositivo Programável: utilizado como dispositivos de campo, responsável por oferecer a versatilidade e flexibilidade do sistema por sua capacidade de alteração via software.

- Infraestrutura de Comunicação: que liga o sistema de supervisão para as Unidades Terminais Remotas.

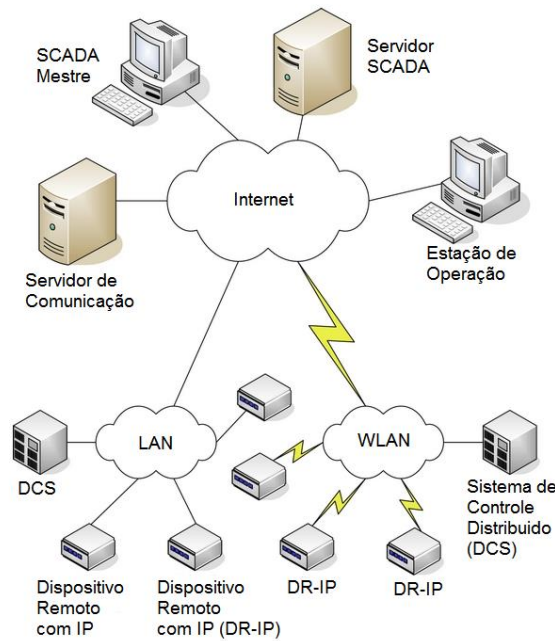
Atualmente os sistemas SCADA adotaram os modelos e infraestrutura de rede fazendo com que haja maior integração entre as diversas unidades controladas. A Figura 1 mostra um exemplo de um sistema SCADA integrado.



**Figura 9: Sistema SCADA Integrado**  
**Fonte: (STOUFFER et. al., 2006)**

Sistemas SCADA antigos com padrões proprietários estão sendo substituídos pelo padrão abertos com os protocolos TCP/IP. Uma das principais razões pelas quais o Protocolo da Internet (IP) é um enorme sucesso é que ele pode ser usado em praticamente qualquer mídia física. Em arquiteturas SCADA complexas, há uma variedade de protocolos e meios de transmissão com cabos ou sem fios envolvidos na obtenção de dados para enviar ao local de monitoramento central. Por isso é possível desenvolver redes SCADA robusta baseadas no IP utilizando sistemas de celular, satélite, telefone fixo ou sistemas mistos, como na Figura 9 (STOUFFER et. al., 2006)..





**Figura 10: Sistema SCADA Baseado na Internet**  
**Fonte: (STOUFFER et. al., 2006)**

Comunicações SCADA podem ser empregadas de diversas maneiras com a utilização de cabos (linhas telefônicas, fibra óptica, ADSL, cabos) e a utilização de meios de comunicação sem fio (rádio, celular, ou via satélite). A escolha depende de inúmeros fatores que caracterizam a infraestrutura de comunicação já existente. Fatores como o equipamento existente, conexões, comunicações disponíveis nos locais isolados, as taxas de dados a serem enviados e a frequência de acesso à informação, distância do local, orçamento da instalação e capacidade para ampliação para futuras instalações geram impacto na decisão final para a arquitetura SCADA a ser definida (BARR et. al. , 2004).

### 2.2.2. Componentes Físicos de um Sistema SCADA

De um modo geral, um sistema SCADA é constituído pelos seguintes componentes. Primeiros instrumentos analíticos, que detectam variáveis de processo e equipamentos operacionais ligados a instrumentos (Sensores). Em segundo lugar, um ou mais dispositivos de campo que fazem a interface de dados, esses se ligam aos sensores no processo e realizam a conversão dos

sinais de sensor em dados digitais e os enviam para o sistema de supervisão. Em terceiro lugar, a infraestrutura de comunicação usada para transferir os dados entre dispositivos de campo de interface de dados com as unidades de controle e o sistema de supervisão (STOUFFER et. al., 2006).

Os sensores são dispositivos conectados aos equipamentos controlados e monitorados pelos sistemas SCADA, são responsáveis por converterem parâmetros físicos tais como velocidade, nível de água e temperatura, para sinais analógicos e digitais legíveis pela estação remota. Os atuadores são utilizados para atuar sobre o sistema, ligando e desligando determinados equipamentos (SILVA; SALVADOR, 2005).

O processo de controle e aquisição de dados se inicia nas estações remotas, PLCs (*Programmable Logic Controllers*) e RTUs (*Remote Terminal Units*), com a leitura dos valores atuais dos dispositivos que a ele estão associados e seu respectivo controle. Os PLCs e RTUs são unidades computacionais específicas, utilizadas nas instalações fabris (ou qualquer outro tipo de instalação que se deseje monitorar) para a funcionalidade de ler entradas, realizar cálculos ou controles, e atualizar saídas.

A rede de comunicação é a plataforma por onde as informações fluem dos PLCs/RTUs para o sistema SCADA e, levando em consideração os requisitos do sistema e a distância a cobrir, pode ser implementada através de cabos Ethernet, fibras ópticas, linhas dedicadas, rádio modems, etc.

O sistema de monitoração central são as unidades principais dos sistemas SCADA, sendo responsáveis por recolher a informação gerada pelas estações remotas e agir em conformidade com os eventos detectados, podendo ser centralizadas num único computador ou distribuídas por uma rede de computadores, de modo a permitir o compartilhamento das informações coletadas.

### 2.2.3. Sistemas SCADA Baseados na Internet

Existem várias arquiteturas diferentes baseadas na Web para um sistema SCADA. Os sistemas são bastante diferentes em design e funcionalidade. Cada um tem seu próprio conjunto de vantagens e desvantagens (SILVA; SALVADOR, 2005).

Um desses sistemas é o equipamento remoto com um servidor *web* embarcado diretamente no dispositivo. Este tipo de dispositivo é adequado para pequenos sistemas autônomos remotos que não exigem um grande controlador, mas precisam de acesso remoto via a Internet. O equipamento está conectado diretamente à Internet e tem pré configurado páginas *web* armazenadas no servidor da *web* embarcado (STOUFFER et. al., 2006).

Usando o conjunto de protocolos TCP/IP, é possível construir um sistema de controle em forma modular, onde os dispositivos microcontrolados estão conectados com uma planta usando o TCP/IP através de sistemas supervisórios SCADA mediante software para conexão com a Internet (SILVA; SALVADOR, 2005).

O sistema de um servidor *web* embarcado em um MCU oferece todas as funcionalidades do controle supervisório baseada na *web* que envolvem a troca de dados em tempo real, pelos quais é possível gerar telas gráficas, fazer análise de tendências, armazenar as informações em um sistema de banco de dados ou que podem ser visualizados em um navegador *web*. Além disso, também oferece as funcionalidades do acionamento remoto e gestão remota dos equipamentos, gerando economia de tempo e custos, através da possibilidade de realizar ajustes remotos, aumentando a produtividade no trabalho e possibilitando a redução de deslocamento de pessoal.

A Internet oferece um meio cada vez mais importante para a distribuição de informações em todo o mundo, informações que podem ser exibidas numérica e graficamente em qualquer dispositivo com conexão com a Internet. Ela permite que os usuários finais possam acessar os dados online e controlar os instrumentos através de um navegador *web* (BARR et. al., 2004).

Não há limite para o número teórico de estações de uma rede Ethernet. A taxa de transferência de dados de um sistema de comunicação serial RS232 é cerca de 19.2 kbps. O padrão Ethernet 10Base-T fornece 500 vezes mais que o de RS232. Ao olhar para a distância de cabeamento(sem repetição) para RS232, o limite é de 23m. Ethernet com o cabo CAT3 é de 91m. Ethernet sem fio pode se estender 96km ou mais com repetidores via ligação em ponte (RUTHERFORD, 2012).

## 2.3. A LINGUAGEM HTML

Inventada por Tim Berners-Lee no Laboratório CERN, é composta por marcações através dos quais é possível definir a aparência e a estrutura de documentos (ALBURQUERQUE, 2001).

Os documentos HTML são organizados como hipertextos e interligados através de links, possibilitando maior agilidade no acesso às informações. Os links são representados por textos ou imagens que, uma vez selecionados, possibilitam que o usuário se desloque para outro documento (PEDROSO et. al., 2007).

Os documentos HTML são arquivos ASCII, criados com um editor convencional ou com editores específicos para esta tarefa. A linguagem HTML usa sequências de caracteres chamados *tags* que informam como os documentos devem ser apresentados. Os *tags* descrevem a aparência do documento. Nem todos os navegadores entendem todos os *tags*, sendo ignorados os *tags* não entendidos. A seguir, os formatos genéricos dos *tags* serão apresentados (PEDROSO et. al., 2007).

- *Tag* isolado:

```
<tag [opcoes]>
```

- *Tag* em par:

```
<tag [opcoes]>objeto</tag>
```

Um documento HTML é delimitado pelos *tags* `<HTML>` e `</HTML>` e dividido entre cabeçalho e corpo. O cabeçalho é delimitado pelos *tags* `<HEAD>` e `</HEAD>`, o corpo pelos *tags* `<BODY>` e `</BODY>`. O corpo ocupa a maior parte do documento e contém informações variadas.

São diversas *tags* que podem ser utilizadas na página e as mesmas podem ser combinadas. Como no exemplo a seguir:

```
<html>
  <head><title> Exemplo de Título HTML </title></head>
  <body style=" background:#FFFFFF">
    <h1> Primeiro nível de cabeçalho </h1>
    <p>Este é o primeiro parágrafo
    <p>Este é o segundo
    
  </body>
</html>
```

Como se pode observar, os conteúdos das páginas *web* são identificados através das *tags*. Cada *tag* terá dentro de si as diretivas, que são os comandos relacionados a um determinado conteúdo. A maioria das *tags* da HTML são utilizadas de forma a indicar o início de uma área da página, com o formato <COMANDO>, e o final da mesma, com o formato </COMANDO> (tag em par), ou somente o primeiro (tag isolado) como na inserção de uma imagem.

As páginas *web* são desenvolvidas utilizando também diversas outras linguagens como Java, PHP, ASP, dentre outras. A escolha de uma linguagem vai depender da capacidade de interpretação do servidor embarcado e também da necessidade, podendo ser HTML puro ou combinado com outras.

### 3. MATERIAIS E MÉTODOS

Para efetuar um acionamento ou monitoramento, anteriormente aos servidores *web* embarcados, era utilizado um modelo com uma IHM e uma placa de aquisição de dados. Na proposta deste projeto o servidor *web* será embarcado no MCU.

Para que um sistema desenvolvido possa conectar-se à rede Ethernet, o mesmo deve possuir os seguintes componentes:

- Microcontrolador;
- Pilha TCP/IP para sistemas embarcados;
- Controlador Ethernet;
- Interface de rede.

Levando em consideração o uso de um microcontrolador, o mesmo terá que ter condições de manipular uma pilha TCP/IP (nesse caso é desenvolvida para uso em microcontroladores e será gravada em sua memória *flash*) para que possa se conectar a rede como se fosse um computador normal, estabelecendo conexão lógica com uma máquina remota em um determinado segmento de rede.

No modelo embarcado, o MCU é que possui um servidor *web* implementado que se comunica com a aplicação realizando as ações solicitadas. Utilizando a internet embarcada, não existe mais a necessidade de manter um computador dedicado (PEREIRA, 2014).

#### 3.1. O MICROCONTROLADOR

O microcontrolador selecionado para utilização neste projeto é fabricado pela Microchip e pertence à Família PIC18. Os microcontroladores desta família possuem processamento de 10 a 16 MIPS (milhões de instruções por segundo). Possuem até 128kB de memória de programa e uma faixa de 18 a 100 pinos. Possuem um conjunto de periféricos tais como: Entradas e saídas digitais, PWM, USART (*Universal Synchronous Receiver Transmitter*), entradas analógicas, MSSP (*Master Synchronous Serial Port*), dentre outros. A família PIC 18

também permite a implementação em “firmware” de tecnologias como Ethernet, USB e CAN e/ou utilização, em conjunto, com “chips” externos que implementam essas tecnologias.

Para a utilização neste projeto foi selecionado o microcontrolador PIC 18F4620 junto com o controlador Ethernet ENC28J60 que possui a interface Ethernet integrada. O PIC18F4620 é um microcontrolador avançado de 8 bits caracterizado pelo conversor analógico/digital de 10 bit's e por utilizar a tecnologia nano watt que opera com tensões de alimentação de 5V. Este microcontrolador será detalhado a seguir.

### 3.2. CARACTERÍSTICAS DO PIC18F4620

Conforme Microchip (2008a, p. 3), as principais características deste microcontrolador são:

- 33 pinos de E/S: o desenvolvedor pode selecionar cada pino individualmente como entrada ou saída de dados;
- Apresentado na forma DIP (*Dual In-line Package*) 40 pinos.
- 3 módulos de temporização: módulos que atuam como temporizadores ou contadores;
- 13 canais analógicos/digitais: canais que podem ser configurados para atuarem de forma analógica;
- Memória flash avançada: permite o microcontrolador escrever em sua própria memória de programa;
- Alimentação de 5 volts.
- Frequência de operação até 40MHz.
- Multiplicação por Hardware;

O Quadro 1 apresenta de um modo geral e simplificado as características de memória do microcontrolador PIC18F4620.

Memória de Programa		Memória de Dados	
Flash(bytes)	Instruções	RAM(bytes)	EEPROM(bytes)
64k	32768	3968	1024

**Quadro 1: Memória de Programa e Dados Disponível**  
**Fonte: Introdução ao PIC 18F4620 (NEVES, 2011).**

Em termos de flexibilidade e estrutura do oscilador, o PIC 18F4620 permite a configuração do oscilador em quatro modos distintos de cristal, sendo possível utilizar cristais até 25 MHz e recorrendo ao *Phase Lock Loop* (PLL) é possível multiplicar por 4 a frequência de oscilação do cristal ou do oscilador interno. A frequência máxima de oscilação está limitada a 40 MHz incluindo o uso do PLL, possui ainda um oscilador interno que pode operar desde 31kHz até 32MHz. O microcontrolador também permite o uso de um oscilador secundário de 32kHz, que poderá ser usado para gerar um relógio em tempo real independente do oscilador principal (NEVES, 2011).

Neste projeto o PIC18F4620 opera com um cristal de 10MHZ. Essa frequência é quadruplicada usando a opção HSPLL que habilitará o PLL interno, fazendo com que o MCU atinja uma frequência de operação de 40MHZ.

### 3.2.1. Entradas/Saídas digitais

Como visto, o PIC 18F4620 possui 33 pinos de Entrada/Saída distribuídos em 5 grupos chamados PORTs. Todas elas podem ser configuradas como Entradas/Saídas digitais, quer dizer, podem assumir nível lógico 1 (+5 V) ou nível lógico 0 (0 V). Cada PORT possui 3 registradores associados à sua operação, o TRIS, o PORT e o LAT.

O registrador TRIS é responsável por definir o sentido de fluxo dos dados. Se configurado em 0, o pino correspondente funcionará como uma saída. Se configurado como 1, o pino correspondente funcionará como entrada;

O registrador PORT é utilizado para leitura do nível lógico do pino correspondente;



Registrador LAT é utilizado em operações de leitura, modificação e escrita nos valores dos pinos correspondentes. Deve-se configurar corretamente cada um dos pinos utilizados no projeto de acordo com sua função de Entrada ou Saída de sinal.

### 3.2.2. Conversor Analógico/Digital

Dos 33 pinos de Entrada/Saída que o microcontrolador PIC 18F4620 disponibiliza, 13 deles podem ser configurados para trabalhar como entrada analógica. Quando configurados, estes pinos são conectados internamente ao módulo conversor analógico/digital do microcontrolador através de um multiplexador. Este módulo permite a conversão de um sinal analógico presente em cada uma destas entradas em um sinal digital de 10 bits. O mesmo módulo controla todas as 13 entradas que devem ser selecionadas antes da leitura. Para leitura de outra entrada analógica, deve-se modificar novamente a entrada selecionada através de seus registradores.

Os registradores utilizados para configuração do módulo conversor analógico/digital são:

- ADRESH;
- ADRESL;
- ADCON0;
- ADCON1;
- ADCON2;

Os registradores ADRESH e ADRESL são responsáveis por manter o valor do resultado da conversão. Cada um deles possui 8 bits e, como o conversor A/D possui 10 bits, o registrador ADRESL foi configurado para armazenar os bits menos significativos e o registrador ADRESH para armazenar os bits mais significativos.

Para realizar uma conversão A/D, em primeiro lugar deve-se selecionar o canal desejado através do registrador ADCON0. Depois setar o bit GO/DONE (bit 0 do ADCON0) em nível lógico "1". Em seguida aguardar que o bit GO/DONE

retorne ao nível lógico “0”, o que indica que a conversão já foi concluída e os registradores ADRESL e ADRSH já possuem o resultado da última conversão.

### 3.3. O CONTROLADOR ETHERNET

O controlador Ethernet é um hardware que possui um módulo que é responsável por receber as informações enviadas pela Interface de Rede e codificar no padrão IEEE 802.3 (Ethernet) as informações trocadas entre o meio externo e o microcontrolador. Ele é responsável pelo controle e fluxo de pacotes de dados entre o microcontrolador e outros dispositivos Ethernet aos quais está conectado. Possui ainda uma unidade hardware dedicada somente para correção de erros de comunicação. (MICROCHIP®, 2004).

O ENC28J60 é um controlador Ethernet autônomo com uma Interface Periférica Serial padrão (SPI™). Ele é projetado para servir como uma interface de rede Ethernet para qualquer controlador equipado com SPI (MICROCHIP, 2004).

As principais características do controlador são:

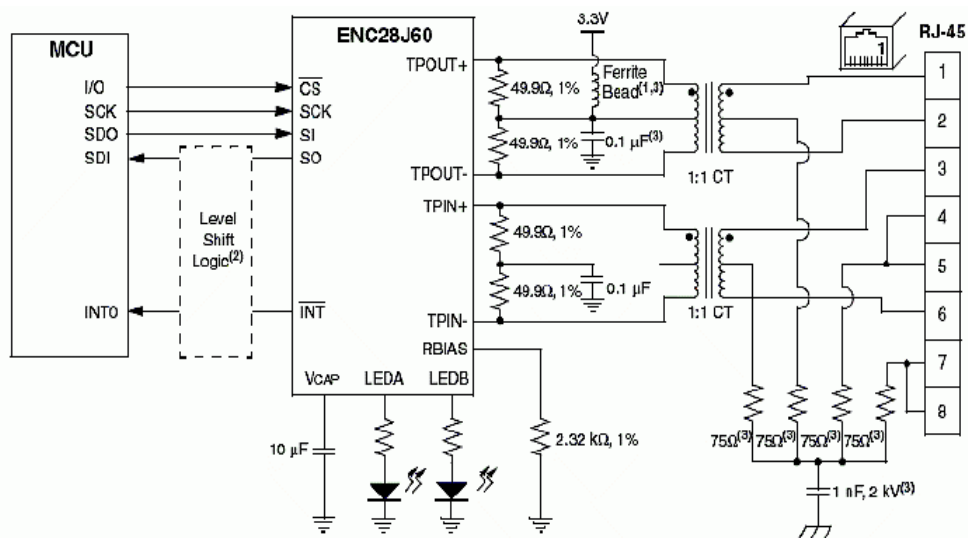
- Controle Ethernet compatível com o padrão IEEE 802.3™;
- Totalmente compatível com redes 10BASE-T;
- Possui integrado o protocolo Media Access Control (MAC);
- Possui o protocolo PHY 10Base-T<sup>9</sup>;
- Suporta uma porta 10Base-T com detecção de polaridade automática e correção de erros;
- Suporta os modos de transmissão *full* e *half-duplex*;
- Retransmissão automática programável em caso de colisão de dados;
- Interface SPI com velocidades de *clock* de até 20 MHz;
- Buffer: 8 kbytes de transmissão/recepção de pacotes;
- Endereço MAC: Suporta pacotes *Unicast*, *Multicast* e *Broadcast*;
- Tensão de funcionamento de 3,1 V a 3,6 V (3,3 V típica);

---

<sup>9</sup> PHY 10BASE-T é o protocolo que define a taxa de transmissão de dados, a codificação do sinal e do tipo do meios de transmissão que interligam dois nós da rede (10BASE-T refere-se a 10Mbps, Fast Ethernet para 100 Mbps e Gigabit Ethernet para 1000 Mbps).

O ENC28J60 opera com 3,3 V, por isso oferece em sua saída quando em nível alto apenas 3,3 V. Pode-se usar um MCU que opere também com 3,3 V (como um PIC18LF4620), mas isso limita a frequência de *clock* máximo suportado, reduzindo também o *clock* máximo SPI gerado, reduzindo a taxa de transmissão de dados. (MICROCHIP®, 2004).

Embora o controlador opere com 3,3 V ele foi desenvolvido para ser facilmente integrado ao MCU de 5 V. Por isso suas entradas, CS, SCK, RESET suportam 5 V, mas as saídas só oferecem 3,3 V, sendo necessário um conversor de nível de tensão. O conversor de nível de tensão utilizado neste projeto foi o 74HC125D (*Quad Tri-state Buffer*). Esse tem a função de ler uma entrada de 3,3 V como nível lógico 1 e oferecer na sua saída um nível lógico 1 em 5 V. Os circuitos de conexão recomendados para o ENC28J60 são mostrados na Figura 11 e 12 a seguir (MICROCHIP®, 2004).



**Figura 11: Conexões do ENC28J60**  
**Fonte: Datasheet ENC28J60 (MICROCHIP, 2004)**

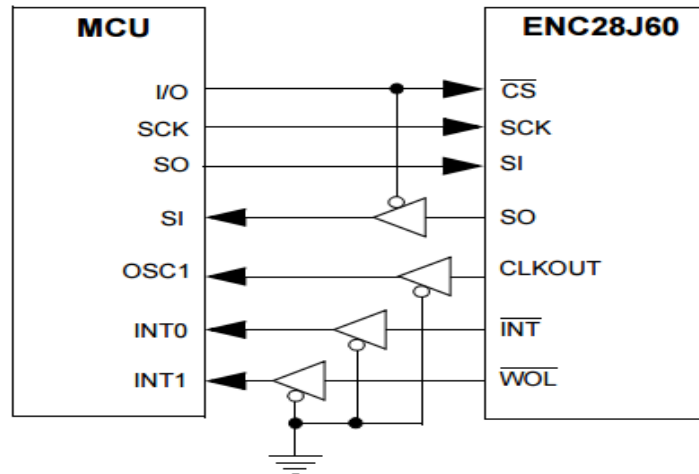


Figura 12: Conexão dos Conversores de Nível  
 Fonte: Datasheet ENC28J60 (MICROCHIP, 2004)

### 3.4. A INTERFACE DE REDE

Para a construção do protótipo foi adquirido o módulo Ethernet ENC28J60, que apresenta o conector RJ45 HR911105A da HanRun e o controlador ENC28J60 da Mircrochip®, conforme Figura 13.



Figura 13: Módulo Ethernet ENC28J60 da HanRun  
 Fonte: Arduinobr, 2015.

O módulo possui um conector RJ45 com transformador magnético, o ENC28J60 junto com todos os outros componentes necessários para o correto funcionamento. A saída é uma interface SPI, sendo somente necessária a conversão de nível de tensão e a conexão ao MCU. Como a placa opera com 3,3 V é necessário fazer um acoplamento de tensão, com um regulador de tensão de 5 V para 3,3 V (HPSPIN, 2015, MICROCHIP®, 2011).

### 3.5. O SENSOR DE TEMPERATURA LM35

O sensor LM35 é um sensor de precisão, que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontrar, tendo em sua saída um sinal de 10 mV para cada Grau Celsius de temperatura. Sendo assim, apresenta uma boa vantagem com relação aos demais sensores de temperatura calibrados em Kelvin, não necessitando nenhuma subtração de variáveis para que se obtenha uma escala de temperatura em graus Celsius (*Datasheet* LM35, 2015).

O LM35 não necessita de qualquer calibração externa para fornecer com exatidão dentro da faixa de temperatura de  $-55\text{ }^{\circ}\text{C}$  à  $150\text{ }^{\circ}\text{C}$ . Este sensor tem saída com baixa impedância, tensão linear e calibração inerente precisa, fazendo com que a interface de leitura seja simples, barateando todo o sistema em função disto.

Este sensor poderá ser alimentado com alimentação simples ou simétrica de 4-20 V, dependendo da tensão que se desejar como sinal de saída, mas independentemente disso, a variação da tensão na saída continuará sendo de 10 mV/ $^{\circ}\text{C}$ .

### 3.6. SOFTWARES UTILIZADOS

Os principais softwares utilizados na implementação do sistema são apresentados a seguir.

### 3.6.1. MPLAB IDE

O MPLAB IDE é um ambiente de desenvolvimento integrado, que permite fácil integração com o compilador MPLAB C18 ou outro tipo de compilador que gere um arquivo hexadecimal para gravar no microcontrolador, além de ser uma ferramenta extremamente poderosa, pois ela gerencia o projeto, compila, simula e em muitos casos grava no chip (MICROCHIP®, 2011).

Desenvolvido e fornecido gratuitamente pela Microchip®, o MPLAB é executado como um aplicativo de 32 bits no sistema operacional Microsoft® Windows, é de fácil utilização e possui um conjunto de ferramentas para desenvolvimento e depuração de forma rápida e ágil de aplicações embarcadas baseadas nos produtos da Microchip®. O ambiente do MPLAB IDE v8.92 utilizado nesse trabalho pode ser visto na Figura 14 (MICROCHIP®, 2011).

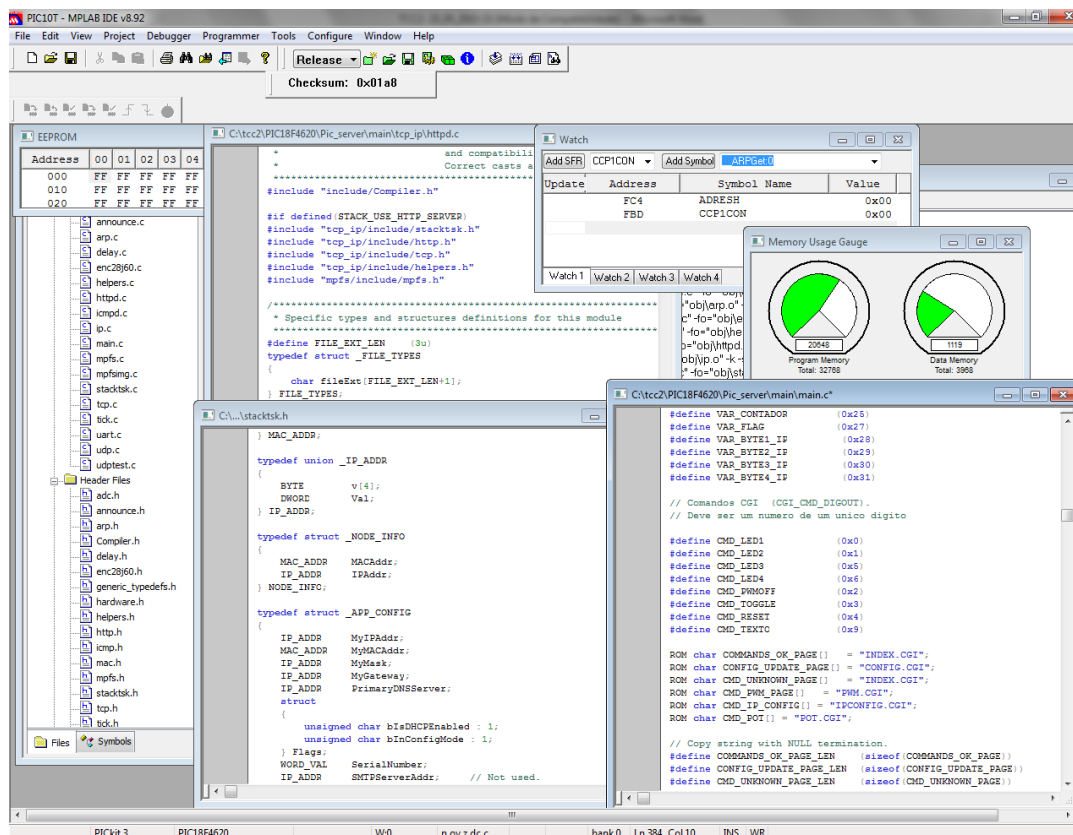


Figura 14: Ambiente de Trabalho do MPLAB IDE v8.92  
Fonte: Autoria Própria

### 3.6.2. O Compilador MPLAB C18

O C18 é um compilador em linguagem C desenvolvido com funções específicas para os microcontroladores disponibilizado pela Microchip® gratuitamente, que é executado em um PC e produz o código que pode ser executado em MCUs PIC18XXXX da Microchip®.

A grande vantagem do Compilador C18 é a sua integração com o MPLAB, ou seja, pode-se desenvolver os códigos em C diretamente no MPLAB, anteriormente desenvolvido em códigos Assembly. O C18 faz a conversão dos dados diretamente em arquivos .hex no momento da compilação, e pode-se gravar o MCU sem a necessidade da importação do arquivo .hex no MPLAB, necessário quando a utilização de outros compiladores, resultando em maior agilidade na construção e execução de projetos.

Todas as pilhas TCP/IP da Microchip são escritos em C para o compilador C18, inclusive a geração das páginas *web* pelo programa “MPFS.exe” em tabelas com valores hexadecimal.

### 3.7. SOFTWARE DO SERVIDOR WEB

Como já foi descrito, a pilha TCP/IP da Microchip possui os principais protocolos implementados para que seja possível estabelecer conexões à rede mundial de computadores, a internet.

O sistema de arquivos MPFS da Microchip é uma implementação de um servidor *web* com as páginas *web* necessário para o armazenamento dos arquivos na memória do MCU para serem acessadas pelos usuários do sistema.

A implementação do servidor *web* disponibilizado pela Microchip atende aos pedidos de conexão e requisições HTTP para que as páginas *web* possam ser exibidas no navegador utilizado pelo usuário.

As páginas *web* armazenadas possuem implementações CGI para possibilitar ao usuário efetuar pedidos de envio de comandos para o PIC gerenciado e permitir a exibição de variáveis do servidor *web*, que representam os

estados dos periféricos do microcontrolador monitorado em tempo real. Os pedidos de envio de comandos através da página *web* são interpretados pela função `HTTPExecCmd(BYTE** argv, BYTE argc)` do servidor *web* e a busca dos valores das variáveis a serem exibidas nas páginas é feita pela função `HTTPGetVar()` toda vez que um caractere '%' é encontrado nas páginas *web*.

### 3.7.1. Interpretação pelo Servidor HTTP

Quando um navegador executa um método GET com mais de um parâmetro, o servidor HTTP deve analisá-lo e chamar o aplicativo principal com o código do método CGI correto, um comando, com seus respectivos parâmetros.

A função responsável por esta tarefa é a função "`HTTPExecCmd(BYTE** argv, BYTE argc)`". O parâmetro "*argv*" é a matriz que contém todos os dados enviados. O "*argc*" contém o tamanho da matriz de dados recebidos. Assim, todas as requisições da página e envios de comando são executados (LCJV, 2007).

No servidor HTTP da Microchip® foi definido por padrão, que o número de argumentos (ou campos de formulário) e que o número máximo de caracteres de um argumento é limitada a 5 e 80, respectivamente.

As funções específicas desenvolvidas pela Microchip®, que tratam da interpretação de comandos recebidos, atendimento a pedidos de leitura de variáveis pela página *web*, solicitação de envio de comandos para o PIC monitorado, dentre outras, são as seguintes:

- `#pragma interruptlow LowISR void LowISR(void)`: faz o tratamento das interrupções de baixo nível. São desempenhadas as funções de PWM através das interrupções do Timer3 previamente configuradas.

- `static void ProcessIO(void)`: é chamada periodicamente na aplicação principal para realizar a tarefa específica de aquisição do valor das entradas analógicas e convertê-las para uma *string* para enviá-las nas requisições das páginas *web*.

- `WORD HTTPGetVar(BYTE var, WORD ref, BYTE* val)`: Retorna para a página *web* o valor obtido de um periférico do PIC monitorado. É ativada por uma requisição CGI quando encontrado o caractere '%' na página *web*. Uma vez que esta função só permite enviar um caractere por vez, o `HTTPServer()` chama essa função várias vezes até que a aplicação principal indicar que não há mais dados



ou todo o valor da variável foi enviado, se for uma *string*, ele envia um caractere por vez e chama esta função até todo o dado for enviado.

- void HTTPExecCmd(BYTE\*\* argv, BYTE argc): Esta é uma chamada de resposta de uma requisição HTTP para o aplicativo principal, como resultado de uma chamada remota. Responsável por enviar comandos para o PIC. Esta função é chamada pelo servidor *web* e ativada pela página *web* quando ocorre uma requisição com o conteúdo do comando a ser enviado, como o *click* em um botão na tela com função específica.

### 3.7.2. Geração de Páginas Web

O server HTTP pode alterar as páginas e substituir informações toda vez que for solicitado a requisição da página *web*, como status de entrada/saída. Para incorporar esta informação em tempo real, o arquivo correspondente CGI (\*.cgi) deve conter a combinação de caracteres, "%xx", onde o caractere '%' serve como um código de controle e 'xx' representa um identificador de uma variável qualquer de dois dígitos. Quando o server encontra esta sequência de texto, ele remove o caractere '%' e chama a função HTTPGetVar().

Considere a pagina exemplo.cgi que esta sendo servida pelo servidor HTTP, e contém a seguinte linha

```
<td>S3=%04</td><td>D6=%01</td><td>D5=%00</td>
```

Ele irá procurar pelas variáveis "04", "01" e "00" que estão depois do caractere %, e estas são declaradas como variáveis HTTP na aplicação principal, e retorna a página com o valor respectivo de cada, pode ser o estado de um bit, um valor inteiro, ou até uma cadeia de caracteres.

A execução do programa MPFS se dá pelo terminal de comando especificando o caminho ou pasta de destino onde se encontra o programa através da linha de comando `cd \tcc2\Pic_server` (caso o MPFS esteja na pasta `C:\tcc2\Pic_server`, por exemplo). Em seguida deve-se gerar a imagem .c através do comando `MPFS \tcc2\Pic_server\pag mpfsimg.c /c` (Figura 15). O arquivo `mpfsimg.c` irá conter as paginas *web* desenvolvidas no formato que o servidor *web* embarcado consegue interpretar (LJCV, 2007).

```
C:\Windows\system32\cmd.exe
MPFS Size so far 8251...
C:\tcc2\Pic_server>
C:\tcc2\Pic_server>
C:\tcc2\Pic_server>
C:\tcc2\Pic_server>
C:\tcc2\Pic_server>MPFS \tcc2\Pic_server\pag mpfsimg.c /c
Adding '\tcc2\Pic_server\pag\INDEX.CGI' ...
MPFS Size so far 3901...
Adding '\tcc2\Pic_server\pag\INDEX.HTM' ...
MPFS Size so far 4229...
Adding '\tcc2\Pic_server\pag\IPCONFIG.CGI' ...
MPFS Size so far 5053...
Adding '\tcc2\Pic_server\pag\MCHP.GIF' ...
MPFS Size so far 7186...
Adding '\tcc2\Pic_server\pag\PWM.CGI' ...
MPFS Size so far 8251...

C:\tcc2\Pic_server>_
```

**Figura 15: Criação das Imagens com MPFS.exe**  
**Fonte: Autoria Própria**

### 3.7.3. Configuração da Rede

Todos os dispositivos em uma rede TCP/IP deve ter atribuído a estes um endereço IP. Considerando que o endereço MAC é o endereço de hardware do dispositivo, o endereço IP é um endereço de software. Normalmente, qualquer dispositivo Ethernet industrial (novo ou de substituição) precisa de um endereço IP atribuído a ele. Muitas instalações de produção possuem atribuição estática de endereços, onde alguém tem que decidir o endereço e configurar os dispositivos finais com os respectivos endereços IP.

Todas as placas homologadas pela Microchip® vem com o protocolo DHCP, que permite atribuição de IP automaticamente. A placa desenvolvida foi configurada com um IP fixo. O IP escolhido foi 192.168.1.20, pode ser qualquer valor IP disponível na rede local, geralmente fica entre os valores 192.168.1.2 à 192.168.1.254. O IP escolhido foi 192.168.1.20.

### 3.8. DESENVOLVIMENTO DA PLACA DE CIRCUITO IMPRESSO NO SOFTWARE EAGLE®

O software EAGLE oferece uma interface com vários tipos de soluções fáceis de usar, ferramentas bastante práticas e acessíveis para o projeto de placa de circuito impresso, incluindo geração de esquemático, layout da placa e a técnica de Auto Router, que gera automaticamente as trilhas dos circuitos. É fácil de aprender e fácil de usar e tem as mesmas funções básicas dos pacotes de softwares comerciais.

No Eagle foram desenvolvidos dois módulos, *Schematic* (Figura 16), e o *Board* (Figura 17).

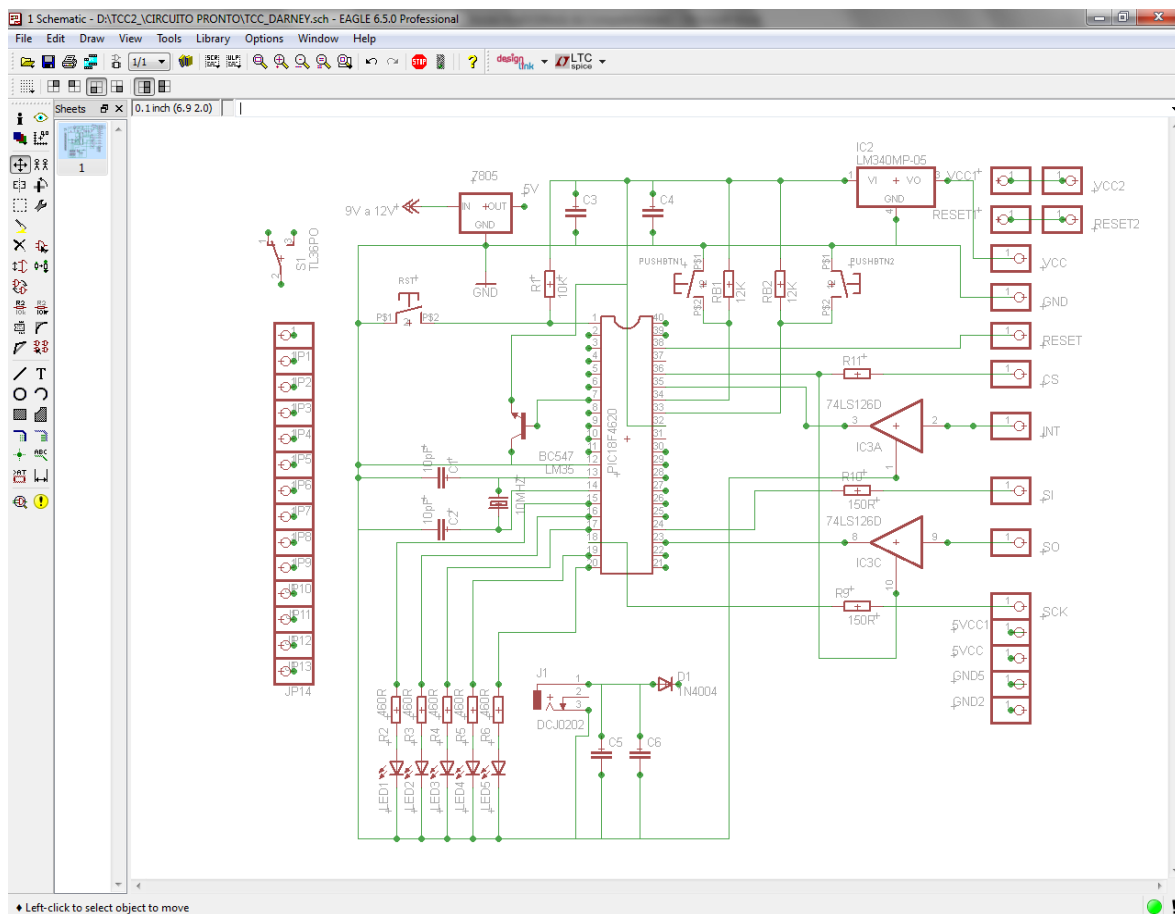
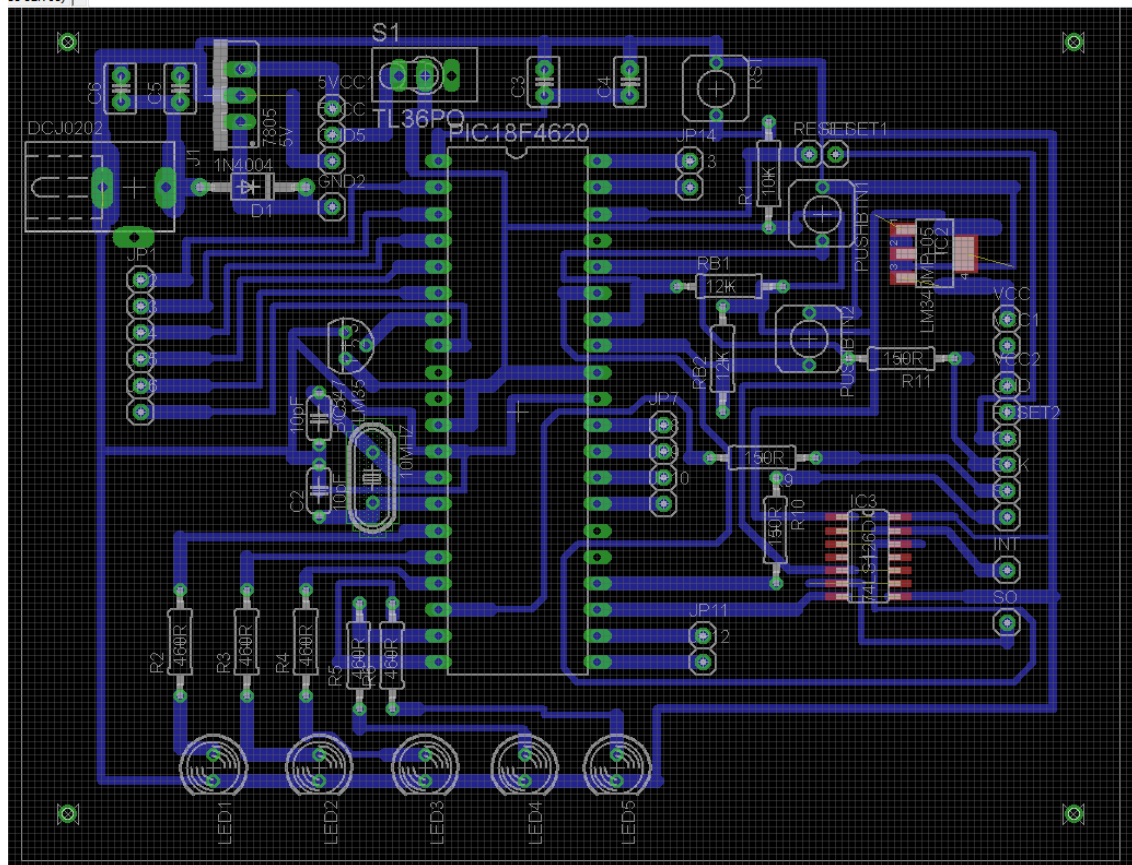
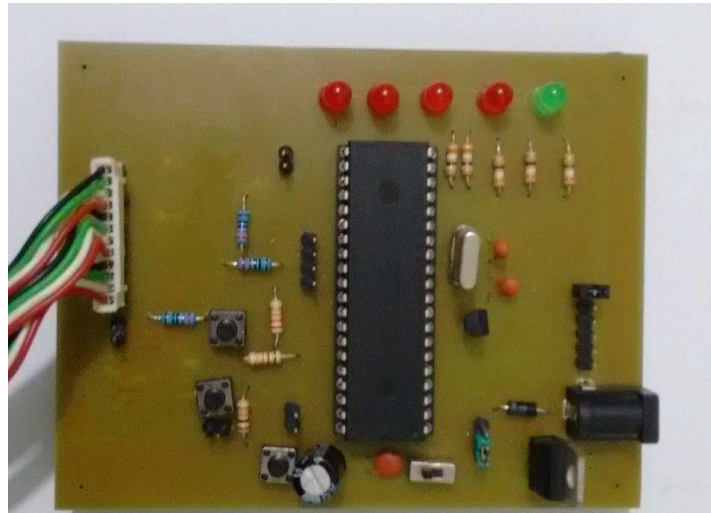


Figura 16: Schematic Produzido para o Projeto  
Fonte: HPSPIN, 2015.

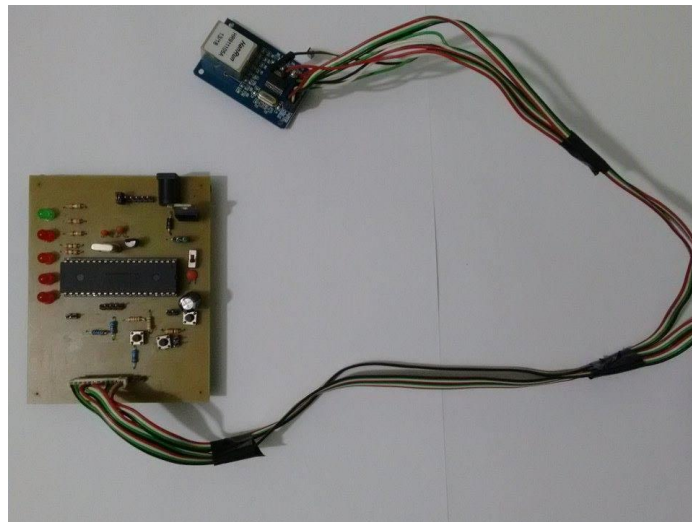


**Figura 17: Board Desenvolvido para o Projeto**  
**Fonte: Autoria Própria**

O hardware desenvolvido se configura numa placa de circuito impresso (PCI) de face simples, onde em um dos lados ficam os componentes (Figura 18) e no outro lado fica a solda e as trilhas de cobre. A conexão com o módulo para interface com a rede é apresentado na Figura 19.

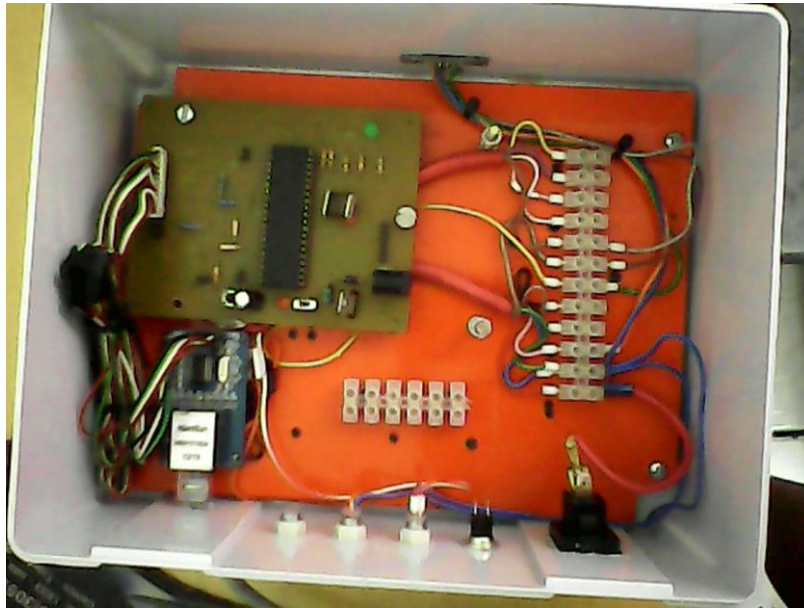


**Figura 18: Módulo Microcontrolado Desenvolvido**  
**Fonte: Autoria Própria**



**Figura 19: Placa Conectada ao Módulo da Hanrum**  
**Fonte: Autoria Própria**

Para que o dispositivo possa ser instalado em campo foi utilizada uma caixa de distribuição onde as saída e entradas do PIC foram conectadas a bornes (Figura 20 e Figura 21), onde se pode facilmente desconectar e conectar sensores, relés e outros componentes.



**Figura 20: Caixa de Distribuição com a Placa**  
Fonte: Autoria Própria



**Figura 21: Vista Frontal da Caixa**  
Fonte: Autoria Própria

### 3.9. CUSTOS DO PROJETO

Os componentes utilizados para a montagem do hardware e os custos relacionados à implementação do *hardware*, é apresentado na Tabela 1.

**Tabela 1. Custos dos componentes utilizados no projeto**

Componente	Quantidade	Valor unitário	Custo
Placa PCI Simples 10x10	1	5,00	5,00
Microcontrolador PIC18F4620	1	26,00	26,00
Módulo Ethernet	1	31,00	31,00
Diodos 1N4007	1	1,00	1,00
Sensor de Temperatura LM35	1	3,50	3,50
LED's	5	0,20	1,00
Botões ( <i>pushbutons</i> )	3	0,50	1,50
Resistores 12k $\Omega$ , de 220 $\Omega$ , de 10k $\Omega$ , de 370 $\Omega$	11	0,1	1,1
Conector para fonte de 12V	1	1,50	1,50
Regulador de tensão L7805	1	0,80	0,80
Regulador LM1117	1	3,75	3,75
Capacitores eletrolíticos de 10 $\mu$ F	1	1,80	1,80
Capacitores cerâmicos de 15 pF e 1 de 100nF	3	0,20	0,60
Conectores de I/O	1	2,50	2,50
Cabo com conectores Modu	1	3,00	3,00
Caixa de Distribuição	1	30,00	30,00
Total			114,00

O desenvolvimento de dispositivos que possuem conectividade com a Internet se mostrou bastante viável já que é uma tendência mundial a conexão de pequenos aparelhos a Internet. As integrações das informações fornecidas pelas máquinas conectadas com a Internet com o sistema corporativo apresentam vantagens por causa do tratamento e armazenamento dos dados, como a utilização dos mesmos para identificar tendências e padrões, identificar falhas, para manutenção preventiva e preditiva, por exemplo.

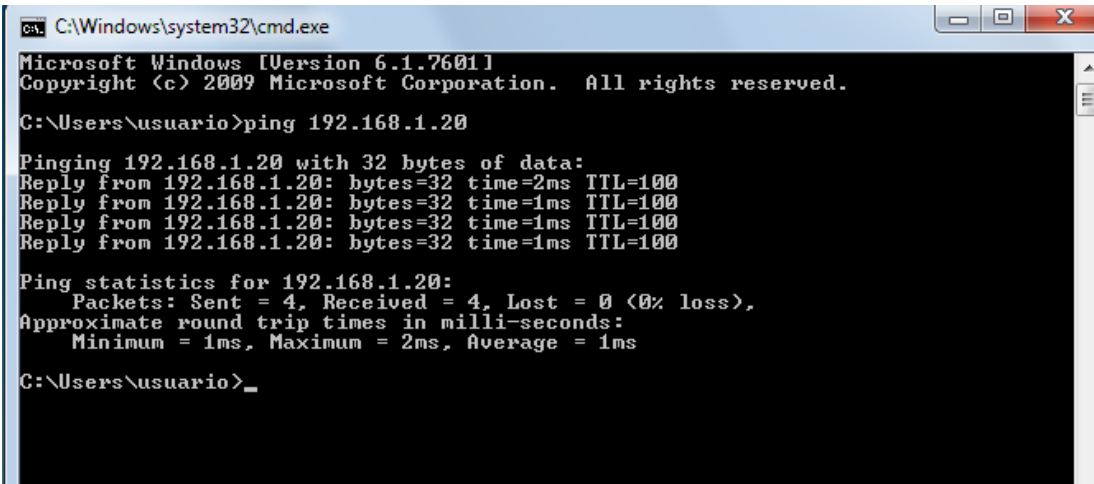


## 4. RESULTADOS

Para utilizar o sistema, primeiramente é necessário estabelecer uma conexão entre o servidor *web* e o PIC monitorado através da conexão da placa e de um computador a um *switch*.

Antes de qualquer uso do sistema, deve-se ter certeza de que o microcontrolador e o módulo Ethernet estejam conectados pelas suas respectivas portas seriais padrão SPI. Também, o servidor *web* deverá estar ligado a algum ponto de rede disponível no local, com um IP disponível e conhecido. Nos exemplos foi configurado e será utilizado o IP 192.168.1.20 para o servidor *web*.

O passo seguinte é estabelecer a comunicação com a placa de controle via rede Ethernet. Para evitar a utilização de cabos e prover acesso ao sistema por dispositivos portáteis, foi instalado um roteador conectado à placa de controle. Em seguida, através de um computador conectado à mesma rede, foi aberto o “prompt de comando” do Windows e executado o comando “PING” para o endereço da placa de comando (Figura 22).



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\usuario>ping 192.168.1.20

Pinging 192.168.1.20 with 32 bytes of data:
Reply from 192.168.1.20: bytes=32 time=2ms TTL=100
Reply from 192.168.1.20: bytes=32 time=1ms TTL=100
Reply from 192.168.1.20: bytes=32 time=1ms TTL=100
Reply from 192.168.1.20: bytes=32 time=1ms TTL=100

Ping statistics for 192.168.1.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\usuario>_
```

Figura 22: Teste de Ping  
Fonte: Autoria Própria

Depois de conectado pode-se escolher a página que se deseja interagir. O servidor enviará à página *web* com os respectivos valores das variáveis observadas nas páginas de monitoramento. Devido o modo de operação



escolhido para o servidor as páginas, que necessitam do valor de variáveis de leitura pelo *hardware*, são atualizadas a cada 10 segundos, enquanto as páginas em que o usuário envia algum dado por escrito ou a ação de um *click* somente respondem aos comandos produzidos por botões disponibilizados na interface *web*.

O software em execução no microcontrolador PIC pode ser compreendido em duas partes: uma relacionada ao sistema do servidor HTTP e outra à execução de tarefas críticas de controle de periféricos, para sua intervenção no meio em que está inserido. Em outras palavras, a primeira seria um módulo do programa do microcontrolador que entra em execução assim que é estabelecido uma conexão entre o microcontrolador e a internet através da página *web*. Esse módulo do firmware é o módulo de comunicação que é executado em conjunto com a segunda parte do programa do microcontrolador que desempenha funções dependentes do ambiente onde o chip está inserido.

O arquivo *index.html* possui uma interface simples responsável pela inicialização e todas as interfaces desenvolvidas apresentam uma estrutura de elementos tratados por folhas de estilo “CSS<sup>10</sup>” com as técnicas de páginas *web* responsivas. Os sinais de entrada aplicados no servidor são provenientes da chegada de requisições pelo navegador geradas pela ação do usuário junto à página *web*, que são botões e/ou figuras clicáveis. Estas imagens são modificadas, acompanhando o estado real em hardware.

A página *web* principal do sistema é mostrada na Figura 23. É a partir dela que se pode acessar as outras páginas para monitoramento e acionamento de entradas (Figura 24), configuração do *duty-cycle* do PWM (Figura 25) e gráfico de temperatura (Figura 28).

---

<sup>10</sup> Cascading Style Sheets (ou simplesmente CSS) é a linguagem para descrever a apresentação de páginas da Web, incluindo cores, layout e fontes. Ele permite adaptar a apresentação para diferentes tipos de tamanho de tela. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.



**Figura 23: Página Inicial**  
**Fonte: Autoria Própria**

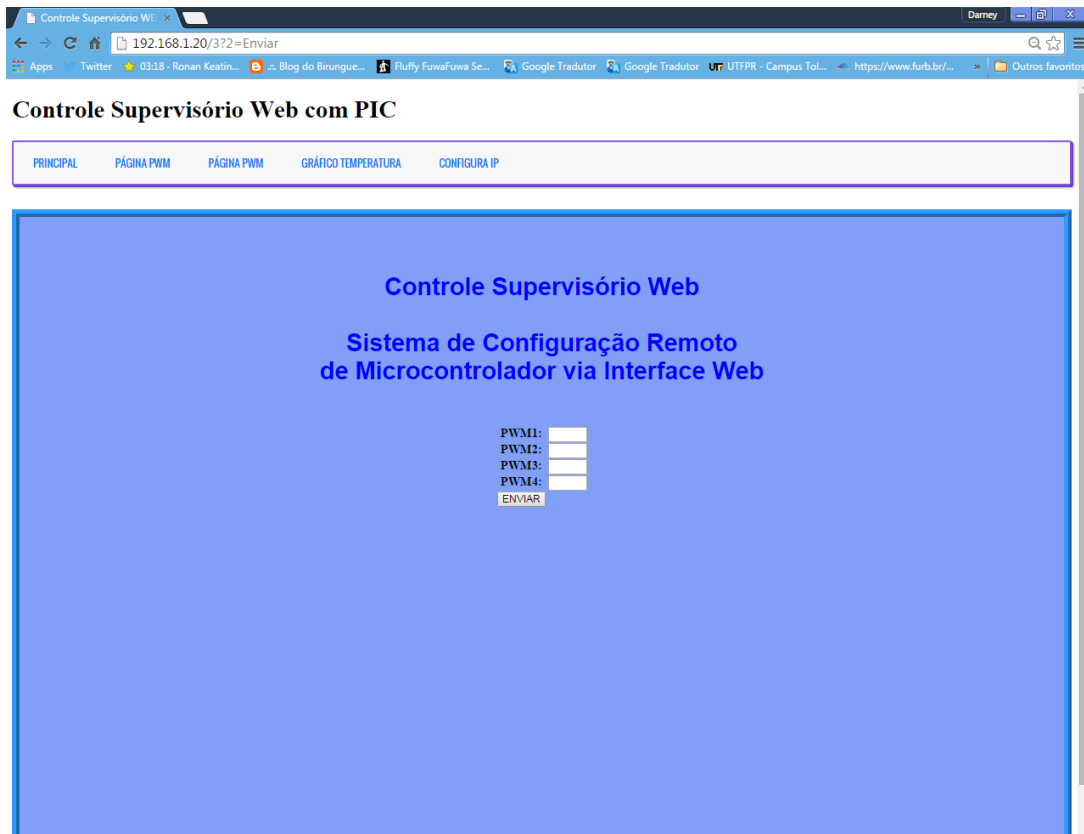
Assim, feita a conexão, o sistema estará sendo monitorado constantemente e esperando por recebimento de comandos para execução de tarefas. O acesso as outras interfaces são feitas por um menu na parte superior.

Na página de acionamento remoto (Figura 24), ao se clicar sobre os botões de liga (LED1ON) e desliga (LED1OFF) será enviado um comando ao MCU que irá retornar a figura que apresenta uma lâmpada ligada ou uma lâmpada desligada, correspondendo ao nível lógico nos pinos em que os LEDs estão conectados. O acionamento remoto demonstra o controle das entradas e saídas do microcontrolador à distância, podendo acionar qualquer dispositivo que possa ser acoplado ao módulo com acopladores ópticos, relés ou qualquer outro componente eletrônico capaz de ser acionado com o nível de tensão e corrente oferecido pelo MCU em seus terminais.



**Figura 24: Página Acionamento Remoto**  
**Fonte: Autoria Própria**

Os programas e sistemas de acesso remoto oferecem muitas vantagens a empresas de qualquer tamanho e setor, entre elas a gestão remota de seus equipamentos. Essas soluções permitem que os profissionais com alta mobilidade acessem e controlem seus equipamentos conectados à internet de qualquer parte, como se estivessem em seus escritórios. Também admitem a organização de mini reuniões, ações de marketing para venda dos produtos monitorados, o gerenciamento e supervisão de sistemas de difícil acesso e até a impressão de dados fornecidos em um computador.



**Figura 25: Página PWM**  
**Fonte: Autoria Própria**

A página PWM (Figura 25) apresenta outro tipo de comando a ser enviado ao PIC, que seria a ativação de um serviço, como a geração de sinal PWM (Pulse Width Modulation - Modulação por Largura de Pulso). O sinal PWM possui uma parcela de seu período de sinal de oscilação em que fica em nível lógico alto (+5V). O *duty cycle* é a relação  $Ton/T$ , onde  $Ton$  é o tempo em que o sinal fica em nível lógico alto (ligado) e  $T$  é o tempo total do sinal. Portanto, ao enviar um comando para alterar o *duty cycle* do PWM, o PIC recebe o comando, interpreta e, caso o PWM nos LED's esteja desativado, o mesmo é ativado e seu *duty-cycle* é alterado para o valor desejado.

A descrição das funções PWM1, PWM2, PWM3, PWM4 representam configurações que estão diretamente ligadas à variação do *duty cycle*, ou seja, o usuário entra com valores em porcentagem (0 à 100%) que deseja para o *duty-cycle* ativo dos LEDs.

Como não foi desenvolvido nenhuma aplicação específica controlada por PWM, esta funcionalidade foi testada nas saídas em que estão conectados os LEDs, porque a operação pode ser visualizada a "olho nu" através da variação da

intensidade luminosa dos LEDs. O Quadro 2 apresenta como foi desenvolvido o código de tratamento da interrupção pelo módulo Timer3 do MCU para geração do PWM.

```

#pragma interruptlow LowISR
void LowISR(void)
{
    if (PIR2bits.TMR3IF) // É Interrupcao de Timer3 ?
        if (pwm_LED1)
        {
            if (le1==1)
            {
                LED1_IO =1;
                led1ON--;
                if (led1ON==0)
                {
                    le1=0;
                    led1ON=100-led1OFF;
                }
            }
            else
            {
                LED1_IO =0;
                led1OFF--;
                if (led1OFF==0)
                {
                    le1=1;
                    led1OFF=100-led1ON;
                }
            }
        }
        if (pwm_LED2)
        {
            ...
        }
        ...
    }
    TMR3H      = 0xFC;      // timer3 high byte
    TMR3L      = 0x17;      // timer low byte max -> overflow 0.1ms
    PIR2bits.TMR3IF=0; // limpa TMR3 overflow flag
}
    TickUpdate();// Função da Interrupção de Timer0 da pilha TCP/IP
}

#pragma code lowVector=0x18
void LowVector(void)
{
    _asm goto LowISR _endasm
}
#pragma code high_vector=0x08

void high_vector(void)
{
    _asm GOTO high_isr _endasm
}
#endif
#pragma code // Retorna ao código

```

**Quadro 2: Tratamento da Interrupção do Módulo Timer3 para PWM**

**Fonte: Autoria Própria**

Foram desenvolvidos testes em laboratório nos pinos em que os LEDs estão conectados, e com auxílio de um osciloscópio obteve-se os resultados caso os campos da Figura 25 forem preenchidos com 10 e 90 (Figura 26), 50 e 10 (Figura 27). A frequência de operação foi previamente definida em 80 Hz.

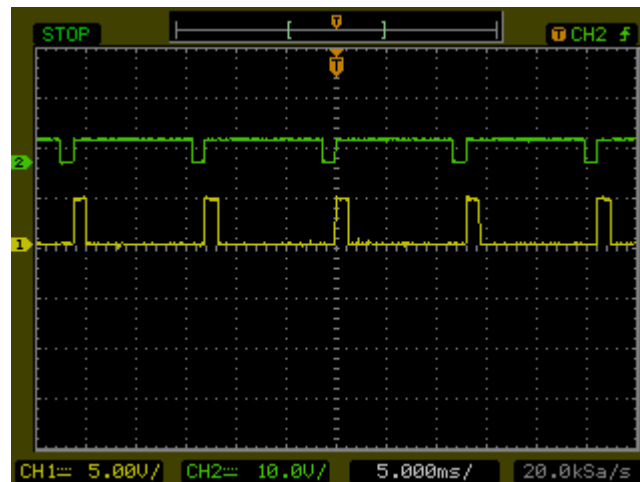


Figura 26: Duty-cycle de 90% e 10% Obtidos  
Fonte: Autoria Própria

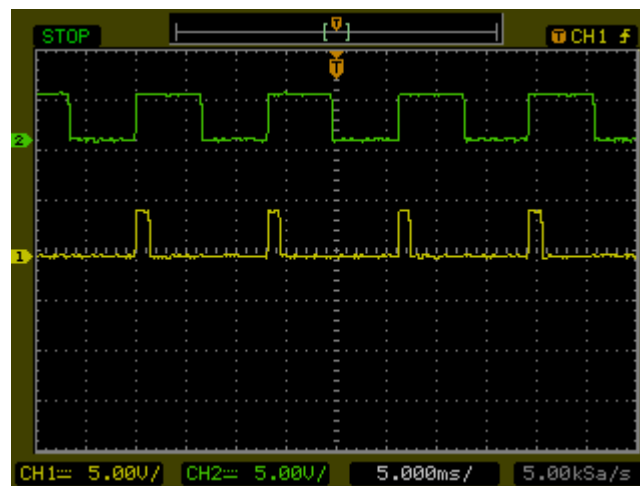


Figura 27: Duty-cycle de 50% e 10% Obtidos  
Fonte: Autoria Própria

As fontes chaveadas, os controles de potência, inversores de frequência e muitos outros circuitos eletrônicos utilizam a tecnologia do PWM. A tecnologia PWM surgiu da necessidade de haver maior controle sobre a potência transmitida, que é mais difícil de obter com controladores lineares, como circuitos

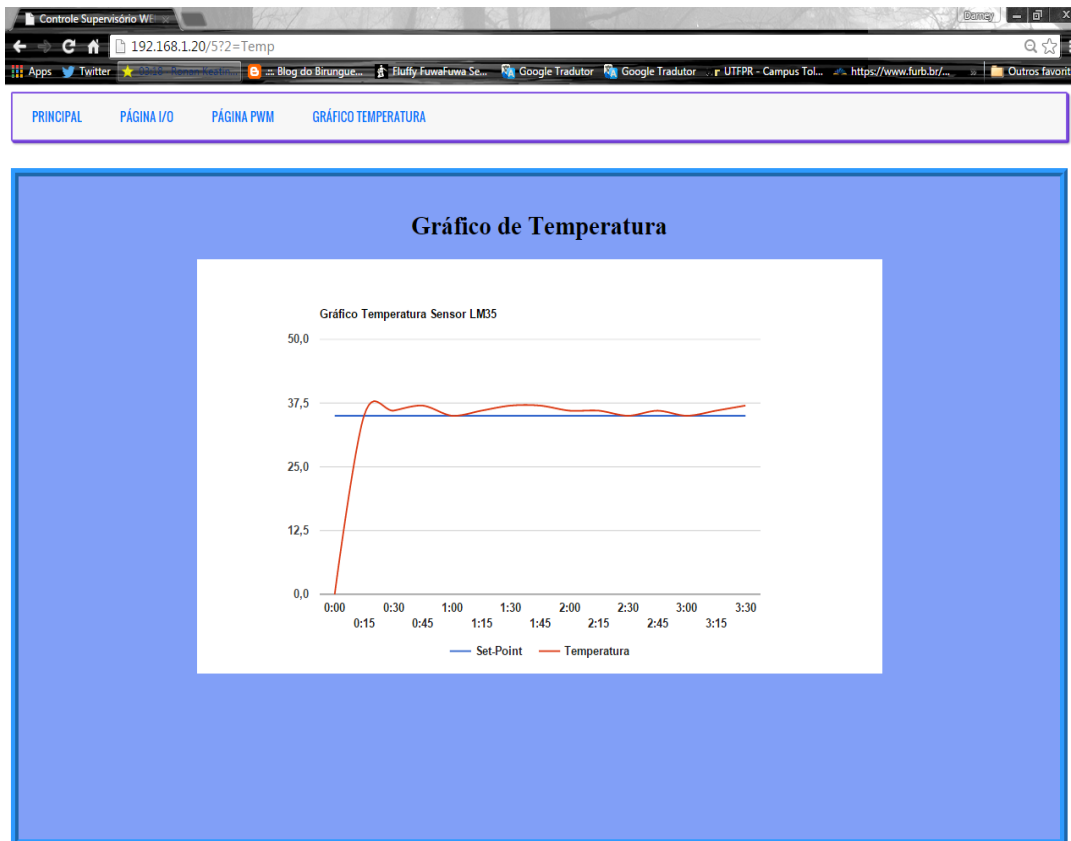
resistivos. A grande vantagem do uso circuitos PWM como controladores contra os circuitos resistivos é quanto à eficiência. Enquanto o PWM trabalha com eficiência quase 1 (menos de 1% de perda), para um circuito resistivo trabalhando a 50% da carga, 50% vai realmente para alimentação da carga, mas uma parte é perdida no aquecimento dos resistores.

A transmissão de informação por PWM pode ser utilizada de várias formas, uma delas é onde o sinal transmitido representa a potência transmitida a carga pela largura do pulso.

Outra forma é a que pode transmitir a carga a potência transmitida a ela e o sentido da corrente na mesma, como exemplo, pode-se definir que o PWM opere em um range de tensão  $-V$  e  $+V$ , ao se aplicar 50% do *duty cycle* a potência média transmitida pelo sinal é 0. Variando-se este *duty cycle* para mais ou para menos passa-se a ter uma predominância do sinal positivo ou negativo, respectivamente, fazendo assim a definição da potência e também do sentido da corrente na carga controlada. Esse tipo de controle é interessante quando se trata de motores elétricos, porque o sentido da corrente define o sentido de rotação.

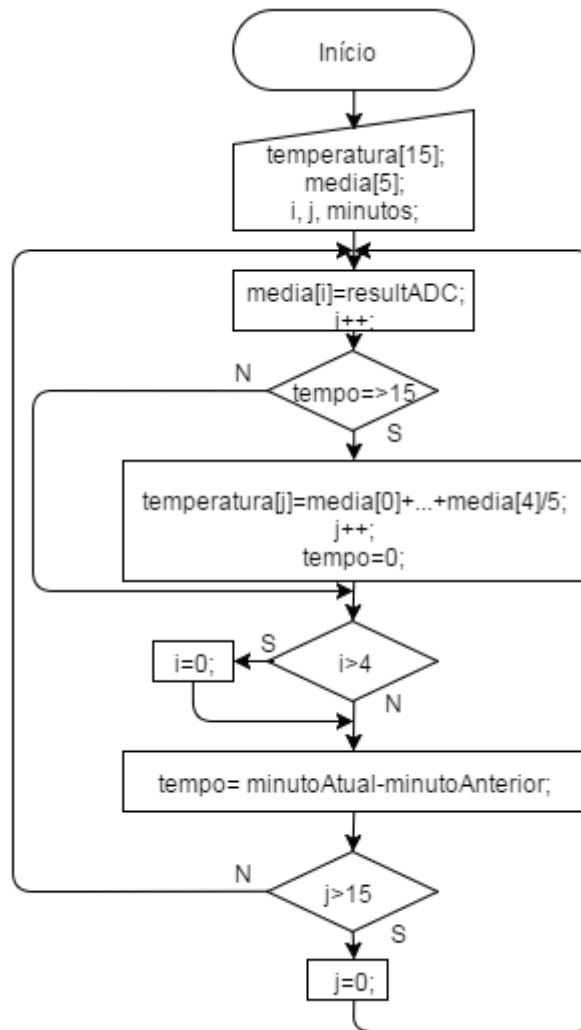
A figura 28 representa a leitura dos valores de temperatura do sensor LM35 durante um período determinado e apresentado em um gráfico. Para representar a variação da temperatura em forma de gráfico, foram criadas 15 variáveis que possuem o valor da temperatura, combinado com a utilização da ferramenta *Google Charts Tools*, onde foi possível ter um gráfico que representa a dinâmica da temperatura durante o tempo.





**Figura 28: Gráfico de Temperatura**  
**Fonte: Autoria Própria**

Os valores apresentados correspondem a uma leitura a cada 15 minutos e num total de 15 amostras representam o comportamento da temperatura durante 3 horas e trinta minutos. O diagrama de blocos da leitura da temperatura é apresentado na figura 29. Nessa figura, “ResultADC” representa o resultado da conversão AD já convertido no valor da temperatura em graus Celsius correspondente, “minutoAtual” e “minutoAnterior” é a referência ao relógio interno criado para contar os minutos, que também utiliza o modulo de interrupção do Timer3 utilizado nas funções PWM. A média se faz necessária para evitar as variações que possam ocorrer nas conversões AD que ocorrem constantemente durante a execução do programa do MCU, as variáveis “i” e “j” são utilizadas como contadores.



**Figura 29: Diagrama de Blocos da Leitura de Temperatura**  
**Fonte: Autoria Própria.**

O gráfico é uma ferramenta muito importante nos processos de controle supervisório. Como o sistema supervisório destina-se à capturar e armazenar em um banco de dados, informações sobre um processo de produção, essa ferramenta disponibiliza as informações que vêm de sensores que capturam dados específicos (conhecidos como variáveis de processo) da planta industrial.

O sistema faz a leitura e armazena as informações em variáveis HTTP e em seguida mostra o resultado em telas no computador do usuário toda vez que uma requisição HTTP for realizada, através da atualização periódica e automática da interface *web*.

Os sistemas de controle supervisório e aquisição de dados tem a sua forma mais usual a geração de gráficos que envolvem registros cronológicos regulares (dia-a-dia, hora-a-hora, etc) de uma ou mais características (por exemplo, média,

amplitude, proporção, etc) calculadas em amostras obtidas das medições em fases apropriadas do processo. Estes valores podem ser armazenados em bancos de dados centrais e dispostos, pela sua ordem, em um gráfico que possui características próprias e inerentes ao processo analisado, como por exemplo, possuir uma linha central e dois limites, denominados limites de controle ou de operação, considerados os extremos aceitáveis para as variáveis medidas.

A possibilidade de obter estes dados diretamente no processo e disponibiliza-los na Internet permite fazer uma análise mais precisa do processo por possibilitar distinguir as causas comuns de operação e das causas especiais (anormais) de variação e indicar se o problema é local ou merece atenção gerencial, evitando frustrações e o custo de erros no direcionamento da solução de problemas. Embora o microcontrolador disponha de pouca memória para armazenar dados por períodos maiores, estes podem ser enviados para servidores centrais disponíveis na empresa ou até em servidores disponibilizados em “nuvem” (*Cloud Servers*).

## 5. CONCLUSÃO

Esse trabalho tratou da implementação de um módulo, microcontrolado com protocolo Ethernet embarcado e com possibilidade de interação através da Internet. Tal sistema possibilita o monitoramento e acionamento de cargas, conectadas ao mesmo, desde que se tenha acesso a uma rede Ethernet e um navegador *web*.

Com o sistema pode-se ter um controle e, ao mesmo tempo, um monitoramento de eventos ocorridos em um microcontrolador através de uma interface *web* utilizando um navegador que poderá estar desde em um computador Desktop até celulares.

Com isso, o sistema pode ser utilizado para que os recursos do microcontrolador sejam utilizados de forma remota sem a necessidade da manipulação direta do dispositivo microcontrolador. Pode-se programar o PIC para efetuar tarefas específicas, com a obtenção de dados ou variáveis, de forma a retornar ao servidor *web* os resultados obtidos com sua execução. Desta forma, o servidor *web* poderá avaliar esses resultados, tomar decisões e enviar comandos ao PIC, tudo de modo automático e de qualquer parte do mundo através do usuário do sistema em um navegador *web*.

Pode-se observar nos exemplos que o sistema é bastante genérico quanto ao monitoramento e controle dos recursos. Quando o sistema for utilizado para gerenciar um microcontrolador que esteja realizando tarefas específicas basta efetuar algumas alterações para um melhor entendimento dos eventos ocorridos. Um exemplo seria, em um controle de irrigação, colocar identificadores nas páginas *web* dos atuadores e sensores ligados ao PIC, como motores de pivôs e sensores de umidade e pressão.

## 6. CONSIDERAÇÕES FINAIS

Hoje em dia, “Ethernet” é quase sinônimo de rede. Por ser um padrão aberto, qualquer fabricante pode fabricar placas e outros componentes de rede e desenvolver soluções, o que aumenta a concorrência e o volume produzido, derrubando os preços.

Entre muitas vantagens estão à facilidade de montar, de gerenciar, manter, a capacidade de interconectividade e acesso remoto. Por tudo isso a Ethernet é a tecnologia de rede local (LAN) mais popular e mais utilizada no mundo em aplicações de rede de comunicação comercial, e mais recentemente em aplicações industriais. Ela se encontra disposta nos mais variados ambientes sejam eles educacionais, públicos, residenciais, comerciais, industriais, etc. Deve ser lembrado que as redes Ethernet possuem a importante característica de conexão a Internet.

O desenvolvimento desse sistema baseado na tecnologia dos microcontroladores possibilitou a produção de um dispositivo flexível, pois é fácil realizar adaptações no módulo para que o mesmo venha atender as diversas necessidades que se apresentem, maximizando seu poder de conexão a diversos equipamentos e sistemas a baixos custos. Tudo isso se soma ao momento atual onde está claro que a quantidade de equipamentos com tecnologias de comunicações embarcadas tende a crescer, aumentando o número de dispositivos ligados à internet.

## REFERÊNCIAS

ABUSAR, 2015. “**HTTP**”. Associação Brasileira dos Usuários de Acesso Rápido. Disponível em <<http://www.abusar.org.br/ftp/pitanga/Intranet/http.PDF>> Acesso dia 12/05/2015.

Arduinobr. “**Colocando o Arduino na rede com o ENC28J60 Ethernet Shield**”. Disponível em : <[http://www.arduinobr.com/arduino/arduino\\_shield/colocando-o-arduino-na-rede-com-o-enc28j60-ethernet-shield/](http://www.arduinobr.com/arduino/arduino_shield/colocando-o-arduino-na-rede-com-o-enc28j60-ethernet-shield/)> Acesso dia 28/04/2015.

BARR, Dale. FONASH, Peter. “**Supervisory Control and Data Acquisition (SCADA) Systems**”, TECHNICAL INFORMATION BULLETIN 04-1. NATIONAL COMMUNICATIONS SYSTEM, 2004.

BEOK, Leandro, R. P; Cibela, F. C; Leandro, R. L; Eder, P; **Protocolo HTTP**. Disponível em: <http://img.vivaolinux.com.br/imagens/artigos/comunidade/Protocolo%20HTTP.pdf>. Acesso dia 12/04/15.

CALVERT, Kenneth L.; DONAHOO, Michael J. “**TCP/IP Sockets in Java: Practical Guide for Programmers**”. 1 ed. Editora Morgan Kaufmann Publishers, 2002.

CARVALHO, Marco, A. S; **Um sistema de monitoramento remoto de pacientes usando rede sem fio**. Dissertação (Mestrado) – Programa de Pós-graduação em Ciências da Computação da Universidade Federal de Minas Gerais. Belo Horizonte, 2005.

Code-Google.com, RSRAHUL. “**PIC**”. Disponível em <<https://code.google.com/p/my-home-projects/source/browse/trunk/PIC/?r=12>> Acesso dia 10/04/2015.

COSTA, Jeferson; **Internet e arquitetura TCP/IP**. Disponível em: <http://www.jeffersoncosta.com.br/TCP.pdf> Acesso dia 12/04/15

CROWLEY, Patrick. “**TCP/IP on Host Processor Project**”. Disponível em: <[http://www.arl.wustl.edu/~sailesh/tcp\\_research\\_page/tcp.htm](http://www.arl.wustl.edu/~sailesh/tcp_research_page/tcp.htm)> Acesso Dia: 23/05/2015

GUERBER, 2007; **Protocolo IP**. Disponível em: <http://www.mfa.unc.br/info/carlosrafael/rco/ip.pdf>. Acesso dia 12/04/15

HPSPIN. "**WebServer\_v04**". Disponível em: [http://www.hp-spin.com.br/site1/circuitos/picwebserver/webserver\\_v04.zip](http://www.hp-spin.com.br/site1/circuitos/picwebserver/webserver_v04.zip)> Acesso dia 20/05/2015.

HPSPIN. "**Web Server no PIC18F4620**" disponível em <<http://www.hp-spin.com.br/site1/circuitos/picwebserver/>> Acesso dia 18/05/2015.

KILIAN, Christopher T. Modern Control Technology: Components and Systems, 2a edição, Delmar Thomson Learning, 2000. Disponível em <<http://bit.ly/1n89A0G>>

LJCV. "**Building the modified Microchip TCP/IP Stack Version 3.75.6**", LJCV Eletronics. 2007. Disponível em <<http://www.ljcv.net/library/en/en002b.pdf>> Acesso dia 16/04/2015

LJCV. "**Internetworking with Microchip Microcontrollers**". Disponível em <<http://www.ljcv.net/projects/pic10t/index.html>> Acesso dia 16/05/2015.

MICROCHIP. Disponível em <[www.microchip.com/tcpip](http://www.microchip.com/tcpip)> Acesso dia 16/05/2015. 2015, Microchip Technology.

MICROCHIP. "**Microchip TCP/IP Stack Help**". 2011, Microchip Technology.

MICROCHIP. "**Stand-Alone Ethernet Controller with SPI™ Interface**". ENC28J60 Data Sheet, 2004.

NEVES, Bernardino. "**Introdução ao PIC18F4620**". CET em Automação, Robótica e Controle Industrial, 4ª Edição. 2011.

ODVA. "**EtherNet/IP Adaptation of CIP Specification**", The CIP Networks Library Volume 2 ed. 1.4, Publication Number: PUB00002; Open DeviceNet Vendor Association, Inc. (ODVA), 2007.

ODVA. "**A Guide for Ethernet/IP Developers**", Publication Number: PUB00213R0; Open DeviceNet Vendor Association, Inc. (ODVA), 2008.

RAFIQ, Abdul. "**AN1108: Microchip TCP/IP Stack with BSD Socket API**" Microchip Technology Inc. 2007

RAJBHARTI, Nilesh. "**AN833: The Microchip TCP/IP Stack**" Microchip Technology Inc. 2002

RFC 791. “**IP: INTERNET PROTOCOL**”, DARPA Internet Program. Protocol Specification, Setembro 1981. Disponível em <<http://www.ietf.org/rfc/rfc791.txt>> Acesso Dia 11/07/14.

RFC 793. “**TCP: Transmission Control Protocol**”. DARPA Internet Program. Protocol Specification, Setembro 1981. Disponível em <<http://www.ietf.org/rfc/rfc793.txt>> Acesso Dia 11/07/14.

RFC 792. “**ICMP: INTERNET CONTROL MESSAGE PROTOCOL**”, DARPA Internet Program. Protocol Specification, Setembro 1981. Disponível em <<http://www.ietf.org/rfc/rfc792.txt>> Acesso Dia 12/07/14.

ROCHA, Helder. “**A Plataforma Web**”. 2014. Disponível em <<http://www.argonavis.com.br/cursos/web/AplicacoesWebCGI.pdf>> Acesso dia 12/03/2015.

RUTHERFORD, Denis. “**Ethernet for Scada Systems**”. 2012. Disponível em <[http://www.schneiderelectric.com/solutions/ae/en/med/20340548/application/pdf/1484\\_se-whitepaper-letter-ethernetforscada-v004.pd.pdf](http://www.schneiderelectric.com/solutions/ae/en/med/20340548/application/pdf/1484_se-whitepaper-letter-ethernetforscada-v004.pd.pdf)>. Acesso dia 12/10/2015.

SILVA, Ana P. G., SALVADOR, Marcelo. “**O que são sistemas supervisórios**”. 2005. Disponível em <[http://www.wectrus.com.br/artigos/sist\\_superv.pdf](http://www.wectrus.com.br/artigos/sist_superv.pdf)>. Acesso dia 12/10/2015.

STEVENS. W. Richard; FALL Kevin R. “**TCP/IP Illustrated**” Volume 1, 2ª edição. Editora Pearson, 2012.

STOUFFER, Keith. FALCO, Joe. KENT, Karen. “**Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security**”. Recommendations of the National Institute of Standards and Technology, 2006.

SOUZA, Thallis, A. **Sistema embarcado aplicado ao monitoramento e controle de irrigação artificial em condições semiáridas**. 2012. Monografia apresentada para obtenção do diploma de Engenheiro da Computação. Montes Claros – MG, 2012.

ZIMMER, Alan Valente. “**Sistema de Automação Residencial Controlado Via Rede Doméstica**”. Trabalho de Graduação em Engenharia Elétrica– Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2014.