

UNIVERSIDADE METODISTA DE SÃO PAULO
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

Danilo Maurílio
Eduardo de Siqueira Correa
Jonas Fabrício Kozamekinas

**CORREÇÃO AUTOMÁTICA DE ROTAS DE
VEÍCULOS POR GPS (SISTEMA DE
POSICIONAMENTO GLOBAL)**

SÃO BERNARDO DO CAMPO
2009

Danilo Maurílio

Eduardo de Siqueira Correa

Jonas Fabrício Kozamekinas

CORREÇÃO AUTOMÁTICA DE ROTAS DE VEÍCULOS POR GPS (SISTEMA DE POSICIONAMENTO GLOBAL)

Trabalho apresentado ao curso de Graduação
em Engenharia da Computação, Universidade
Metodista de São Paulo, como requisito parcial
para obtenção do título de Bacharel em
Engenharia da Computação
Orientador: Prof. Dr. Mario Francisco Guerra
Boaratti.

SÃO BERNARDO DO CAMPO
2009

Danilo Maurílio
Eduardo de Siqueira Correa
Jonas Fabrício Kozamekinas

CORREÇÃO AUTOMÁTICA DE ROTAS DE VEÍCULOS POR GPS (SISTEMA DE POSICIONAMENTO GLOBAL)

Trabalho apresentado ao curso de Graduação
em Engenharia da Computação, Universidade
Metodista de São Paulo, como requisito parcial
para obtenção do título de Bacharel em
Engenharia da Computação
Orientador: Prof. Dr. Mario Francisco Guerra
Boaratti.

Orientador:

Dr. Mario Francisco Guerra Boaratti

FICHA CATALOGRÁFICA

	Título
M447c	Correção automática de rotas de veículos por GPS (Sistema de Posicionamento Global) / Danilo Maurílio, Eduardo de Siqueira Correa, Jonas Fabrício Kozamekinas. 2009. 84 f.
Monografia (graduação em Engenharia da Computação) -- Faculdade de Exatas e Tecnologia da Universidade Metodista de São Paulo, São Bernardo do Campo, 2009.	
Orientação: Mario Francisco Guerra Boaratti	
1. Sistema de posicionamento global - GPS 2. Software 3. Microcontrolador I. Correa, Eduardo de Siqueira II. Kozamekinas, Jonas Fabrício III. Maurílio, Danilo	
CDD 001.5	

Dedicamos esse trabalho à nossa família e
namoradas, por apoiarem e compreenderem a
importância deste trabalho em nossa formação
acadêmica.

AGRADECIMENTOS

Ao nosso orientador, pela sua dedicação, competência e
confiança, que possibilitou a conclusão
deste trabalho, nosso muito obrigado.

Aos Professores com admiração.

Aos nossos amigos e aos colegas de sala.

RESUMO

Esse trabalho descreve sobre algumas tecnologias presentes em nosso cotidiano para construção de um protótipo de veículo autônomo, com sistema de correção de rotas pré-programadas composto por módulo receptor GPS e microcontrolador PIC. O veículo com correção automática de rota pré-programada utiliza conceitos de programação, eletrônica e posicionamento por satélite. A proposta é de um circuito de malha fechada com retorno de informação de posicionamento por GPS, e programação avançada com microcontrolador para tratativa de informação para correção da rota pré-programada. Foram criadas regras de negócio, caso de uso e requisitos funcionais e não funcionais.

Este trabalho teve sucesso em seus objetivos e descreve propostas para um melhor aproveitamento do protótipo desenvolvido.

Palavras-chave: GPS; microcontrolador.

ABSTRACT

This work describes some of these technologies in our present days for construction of a prototype autonomous vehicle, with correction system pre-programmed routes composed of GPS receiver module and PIC microcontroller. The vehicle with automatic correction of pre-programmed route uses programming concepts, electronic and satellite positioning. The proposal is for a circuit with closed loop feedback of GPS positioning, and advanced programming with microcontroller for transacting the information to correct pre-programmed route. Were created business rules, use case and functional requirements and nonfunctional.

This work has succeeded in its objectives and describes proposals for a better use of the prototype.

Keywords: GPS; microcontroller.

LISTA DE FIGURAS

Figura 1: Pinagem do Microcontrolador PIC16F877A	24
Figura 2: Diagrama de Blocos do Microcontrolador PIC16F877A.....	25
Figura 3: Pinagem do Microcontrolador PIC16F628A	27
Figura 4: Diagrama de Blocos do Microcontrolador PIC16F628A.....	28
Figura 5: Diagrama de blocos de um sistema de malha fechada	29
Figura 6: Sensor Óptico por barreira	31
Figura 7: Sensor Óptico por difusão	31
Figura 8: Sensor Óptico por reflexão	32
Figura 9: Utilização de um sensor capacitivo de deslocamento.....	33
Figura 10: Jipe em escala 1:20 (Brinquedo)	38
Figura 11: Microcontrolador PIC 16F877A.....	39
Figura 12: Microcontrolador PIC 16F628A.....	39
Figura 13: Módulo GPS ME-1000RW	40
Figura 14: Circuito Integrado L293D	40
Figura 15: Experimento com Microcontrolador.....	45
Figura 16: Experimento com Microcontrolador 2.....	45
Figura 17: Experimento com GPS	47
Figura 18: Placa Universal	48
Figura 19: Detalhe do circuito do protótipo	49
Figura 20: Circuito do protótipo e veículo em movimento.....	49
Figura 21: Veículo sem a Placa controladora.....	51
Figura 22: Circuito Ponte H	51
Figura 23: Circuito Integrado L293D (ponte H) em conjunto com o veículo	51
Figura 24: Circuito Integrado Ponte H em conjunto com microcontrolador	52
Figura 25: Teste de transmissão com MAX232N	53
Figura 26: Teste de transmissão com GPS, microcontrolador e MAX232N.....	53
Figura 27: Foto do protótipo da placa na placa universal	54
Figura 28: Tabela ASCII.....	55
Figura 29: Circuito de simulação montado no <i>Proteus Isis</i>	56
Figura 30: Tela de compilação do código no <i>CCS Compiler</i>	57
Figura 31: Fluxograma de funcionamento do microcontrolador.....	60
Figura 32: Esquema eletrônico do circuito do veículo feito no <i>Proteus Isis</i>	65
Figura 33: Esquema eletrônico do circuito montado na placa universal de fenolite.....	66

LISTA DE TABELAS

Tabela 1: Especificações Técnicas do módulo receptor GPS	21
Tabela 2: Descrição da pinagem do PIC16F877A	26
Tabela 3: Lista de Materiais adquiridos para construção do protótipo	41

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO	13
1.2	MOTIVAÇÃO	13
1.3	METODOLOGIA	14
2	FUNDAMENTAÇÃO TEÓRICA.....	15
2.1	SISTEMA DE POSICIONAMENTO GLOBAL - GPS	15
2.2	LATITUDE E LONGITUDE.....	17
2.3	ALTITUDE	17
2.4	PADRÃO NMEA 0183.....	18
2.5	MÓDULO RECEPTOR GPS MODELO ME-1000RW.....	19
2.6	MICROCONTROLADOR.....	22
2.6.1	Arquitetura Harvard	23
2.6.2	MICROCONTROLADOR PIC 16F877A	23
2.6.3	MICROCONTROLADOR PIC 16F628A	27
2.7	SENSORES	29
2.7.1	Sensores Ópticos	30
2.7.2	Sensores Capacitivos	32
2.7.3	Sensores de Proximidade Ultrassônicos	34
3	DESENVOLVIMENTO.....	36
3.1	PRIMEIRA FASE: Iniciação (Definições, Levantamento de Requisitos, Casos de Uso) ..36	36
3.1.1	O Propósito do Projeto.....	36
3.1.2	Colaboradores e Usuário (Diretos e Indiretos).....	37
3.1.3	Restrições Mandatárias (Cronograma, Materiais e Orçamento).....	37
3.1.4	Cronograma:.....	37
3.1.5	Materiais.....	38
3.1.6	Orçamento.....	41
3.1.7	Fatos e Suposições	41
3.1.8	Levantamento de Requisitos, Regras de Negócio e Casos de Uso.....	42
3.2	SEGUNDA FASE: Construção (Teste, desenvolvimento e adaptações).....	44
3.2.1	Primeiro experimento:.....	44
3.2.2	Segundo experimento:.....	44
3.2.3	Terceiro experimento:.....	45
3.2.4	Quarto experimento:	46
3.2.5	Quinto experimento:.....	47
3.2.6	Sexto experimento:.....	48
3.2.7	Sétimo experimento:.....	50
3.2.8	Oitavo experimento:.....	52
3.2.9	Nono Experimento	54
3.2.10	Algoritmo do Protótipo:.....	59
3.3	TERCEIRA FASE: Finalização (Montagem, Programação Funcionamento).....	59

3.3.1	Fluxograma de Funcionamento do Microcontrolador	60
3.3.2	Programação do Microcontrolador	60
3.3.3	Sistema Eletrônico do Protótipo	64
3.3.4	Protótipo	66
4	CONCLUSÃO	67
4.1	<i>SUGESTÕES PARA TRABALHOS FUTUROS.</i>	68
5	REFERÊNCIAS BIBLIOGRÁFICAS.....	69
6	GLOSSÁRIO.....	72
7	APÊNDICE.....	78
8	ANEXOS.....	79

1 INTRODUÇÃO

O GPS (Sistema de Posicionamento Global) foi criado em 1960 pelo Departamento de defesa americano e totalmente operacionalizado em 1995, tornando-se uma poderosa ferramenta na atualidade para localização no globo terrestre, e com aplicação em diversos campos (militar, comercial, doméstico, entre outros).

É com a perspectiva desta ferramenta, mais o advento da automação, robótica e programação, que o trabalho de conclusão de curso é desenvolvido. Ele reúne todo o conhecimento adquirido na universidade e novos conhecimentos em seu desenvolvimento, a fim de trazer à realidade a idéia de um veículo autônomo com correção automática de rota pré-programada.

1.1 OBJETIVO

O objetivo deste trabalho visa o estudo e o entendimento das tecnologias contidas no GPS (Sistema de Posicionamento Global), nos microcontroladores e os protocolos de comunicação necessários para utilização no desenvolvimento de um protótipo que integra um módulo GPS a um veículo, com autonomia para chegar ao seu destino com base em uma rota pré-programada.

A proposta é de criação de uma plataforma que permita a movimentação do veículo utilizando as coordenadas do GPS como ponto de referência para chegar ao seu destino final, utilizando um microcontrolador programado em linguagem C para este objetivo.

1.2 MOTIVAÇÃO

O desafio de desenvolver um protótipo devido à complexidade de integração e interação dos diversos componentes, aliado a real possibilidade de utilização das linguagens de programação (na inteligência do protótipo), que foram matérias bem enfatizadas durante o curso de graduação e de aplicar, de modo abrangente, o conhecimento adquirido durante a formação acadêmica, para tornar a idéia de um carro autônomo viável em nossa sociedade.

1.3 METODOLOGIA

Trata-se de uma pesquisa teórica com trabalho de campo empírico e de laboratório, como também implementação do protótipo com um trabalho que combinará as várias formas de tecnologia e de pesquisa.

Para isso, foram feitas pesquisas bibliográficas em livros, artigos de imprensa especializada, sites e revistas, e para a parte prática serão feitos testes em laboratório, simulações realizadas com o auxílio do software *NI Multisim*, além da montagem do circuito em *Protoboard* para testes antes da montagem final dos componentes eletrônicos em placa de circuito impresso.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SISTEMA DE POSICIONAMENTO GLOBAL - GPS

Segundo GORGULHO (2004), a tecnologia atual permite que qualquer pessoa possa se localizar no planeta com uma precisão nunca imaginada por navegantes e aventureiros há até bem pouco tempo. O sofisticado sistema GPS – Global Positioning System (Sistema de Posicionamento Global) – foi concebido pelo Departamento de Defesa dos EUA no início da década de 1960, sob o nome de ‘projeto NAVSTAR’. O sistema foi declarado totalmente operacional apenas em 1995, e seu desenvolvimento custou 10 bilhões de dólares. Testes realizados em 1972 mostraram que a pior precisão do sistema era de 15 metros e a melhor de 1 metro. Preocupados com o uso inadequado, os militares americanos implantaram duas opções de precisão: para usuários autorizados (eles mesmos) e usuários não-autorizados (civis). Cada satélite emite um sinal que contém: código de precisão (P); código geral (CA) e informação de status. Como outros sistemas de rádio-navegação, todos os satélites enviam seus sinais de rádio exatamente ao mesmo tempo, permitindo ao receptor avaliar o lapso entre emissão/recepção. A potência de transmissão é de apenas 50 Watts.

A hora-padrão GPS é passada para o receptor do usuário. Receptores GPS em qualquer parte do mundo mostrarão a mesma hora, minuto, segundo, etc, até milissegundo. A hora-padrão é altamente precisa, porque cada satélite tem um relógio atômico, com precisão de nanosegundo (GORGULHO 2004).

O receptor tem que reconhecer as localizações dos satélites. Uma lista de posições conhecida como almanaque, é transmitida de cada satélite para os receptores. Controles em terra rastreiam os satélites e mantém seus almanaques acurados (GORGULHO 2004).

Cada satélite tem códigos P e CA únicos, e o receptor pode distingui-los. O código P é mais complexo que o CA, quase impossível de ser alterado e somente militares tem acesso garantido a ele. Receptores civis medem os lapsos de tempo entre a recepção dos sinais codificados em CA. O conceito da rádio-navegação depende inteiramente da transmissão simultânea de rádio-sinais. O controle de terra pode interferir, fazendo com que alguns satélites enviem seus sinais CA ligeiramente antes ou depois dos outros. A interferência deliberada introduzida pelo

Departamento de Defesa dos EUA é a fonte da Disponibilidade Seletiva – *Availability Selective* (AS). Os receptores de uso civil desconhecem o valor do erro, que é alterado aleatoriamente e está entre 15 e 100 metros. Os receptores militares não são afetados. Existe outra fonte de erro que afeta os receptores civis: a interferência ionosférica. Quando um sinal de rádio percorre os elétrons livres na ionosfera, sofre um certo atraso. Sinais de freqüências diferentes sofrem atrasos diferentes. Para detectar esse atraso, os satélites do sistema enviam o código P em duas ondas de rádio de diferentes freqüências, chamadas L1 e L2. Receptores caros rastreiam ambas as freqüências e medem a diferença entre a recepção dos sinais L1 e L2, calculam o atraso devido aos elétrons livres e fazem correções para o efeito da ionosfera. Receptores civis não podem corrigir a interferência ionosférica porque os códigos CA são gerados apenas na freqüência L1 (1575,42 MHz). Existem receptores específicos, conhecidos como não-codificados, que são super acurados (GORGULHO 2004).

Como desconhecem os valores do código P, obtém sua precisão usando técnicas especiais de processamento. Eles recebem e processam o código P por um número de dias e podem obter uma posição fixa com precisão de 10 mm. É ótimo para levantamento topográfico. O sinal gerado pelos satélites contém um "código de identidade" (ou pseudo-randômico), dados efêmeros (de status) e dados do almanaque. O código de identidade (*Pseudo-Random Code – PRN*) identifica qual satélite está transmitindo. Usa-se como referência dos satélites seus *PRN*, de 1 a 32. O código pseudo-randômico permite que todos os satélites do sistema compartilhem a mesma freqüência sem interferências. É um sistema engenhoso que torna o *GPS* prático e relativamente barato de se usar. Ao contrário dos satélites de TV, que estão em órbitas geosíncronas (estacionários no céu) e transmitem poderosos sinais para refletores parabólicos em terra, o satélite GPS envia sinais com poucas informações e de baixa potência para antenas do tamanho do dedo polegar. De fato, os sinais GPS são tão fracos que não são maiores que o ruído de fundo (de rádio) inerente à Terra. O princípio do código pseudo-randômico, que significa literalmente "aparentemente aleatório", se baseia em uma comparação realizada em muitos ciclos de um sinal, que é demorada e incômoda se comparada com um sinal de TV. O padrão para comparação do código pode ser alterado (apenas código CA), permitindo que o governo americano controle o acesso ao sistema do satélite (GORGULHO 2004).

Os dados efêmeros (de status) são constantemente transmitidos e contém informações de status do satélite (operacional ou não), hora, dia, mês e ano. Os dados de almanaque dizem ao receptor onde procurar cada satélite a qualquer momento do dia. Com um mínimo de três satélites, o receptor pode determinar uma posição Latitude / Longitude – que é chamada posição fixa 2D – bi-dimensional. (Deve-se entrar com o valor aproximado da altitude para melhorar a precisão). Com a recepção de quatro ou mais satélites, um receptor pode determinar uma posição 3D, isto é, Latitude / Longitude / Altitude. Pelo processamento contínuo de sua posição, um receptor pode também determinar velocidade e direção do deslocamento (GORGULHO 2004).

2.2 LATITUDE E LONGITUDE

Segundo DILÃO (2009), latitude é a distância ao Equador medida ao longo do meridiano de Greenwich. Esta distância mede-se em graus, podendo variar entre 0° e 90° para Norte ou para Sul.

A longitude é a distância ao meridiano de Greenwich medida ao longo do Equador. Esta distância mede-se em graus, podendo variar entre 0° e 180° para Leste ou para Oeste.

As siglas internacionais para os pontos cardeais são N=Norte, S=Sul, E=Leste, W=Oeste.

A forma de representação das coordenadas geográficas é de Graus – Minutos – Segundos, onde cada grau é dividido em 60 minutos, que por sua vez se subdividem, cada um, em 60 segundos. A partir daí, os segundos podem ser divididos decimalmente em frações cada vez menores. Exemplo: $22^{\circ} 54' 21.64''S$ $47^{\circ} 03' 38.06''W$.

2.3 ALTITUDE

A Terra é aproximadamente esférica, com um ligeiro achataamento nos pólos. Para se definir a altitude de um ponto sobre a Terra define-se uma esfera (geóide) com um raio de 6378 km. A altitude num ponto da Terra é a distância na vertical à superfície deste geóide. (DILÃO, 2009)

2.4 PADRÃO NMEA 0183

Todo sistema de transmissão e recepção de informações utiliza padrões de comunicação. O mesmo ocorre com os módulos GPS, que utilizam o padrão NMEA para comunicação.

Segundo informações disponibilizadas pela *National Marine Electronics Association* (Associação Nacional de Eletrônica Marítima) em seu site, o padrão NMEA 0183 (ou simplesmente NMEA) é uma interface padrão que define requisitos de sinais elétricos, protocolo de transmissão de dados e tempo, e especifica formatos de sentenças para um canal de transmissão de dados serial. Cada canal pode ter apenas um falante, porém vários ouvintes. Este dado é do tipo ASCII imprimível e pode incluir informações com posição, velocidade, profundidade, atribuição de freqüência, entre outras.

Os dados são transmitidos em forma de sentenças. Cada sentença inicia com o caracter “\$”, dois caracteres para a identificação do falante, três caracteres para a identificação da sentença, seguido por um número de campos de dados separados por vírgula, e finalizado por um *checksum* opcional, e um retorno de carro/alimentação de linha (*CR/LF – carriage return / line feed*). A sentença pode conter até 82 caracteres incluindo o “ \$ ” e o CR/LF. (BENNET, 2009)

Se os dados para um campo não estiverem disponíveis, o campo é simplesmente omitido, mas as vírgulas que o delimitam ainda são enviadas, sem espaço entre elas. Desde que alguns campos tenham tamanho variado, ou possam ser omitidos, o receptor deve localizar os campos de dados desejados contando a quantidade de vírgulas, e não pela posição do caracter com a sentença. (BENNET, 2009)

O campo de verificação (*checksum*) opcional consiste em um “ * ” (asterisco) e dois dígitos hexadecimais que representam uma operação OU exclusiva entre todos os caracteres , exceto os caracteres “ \$ ” e “ * ”. Um *checksum* é requisitado em algumas sentenças. (BENNET, 2009)

O padrão permite que fabricantes individuais definam formatos de sentenças proprietárias. Estas sentenças iniciais com “ SP ”, então uma identificação do fabricante com três letras, seguido por qualquer dado que o fabricante deseja, seguindo o formato geral das sentenças padrões. (BENNET, 2009)

2.5 MÓDULO RECEPTOR GPS MODELO ME-1000RW

A definição deste módulo receptor para o projeto foi devido ser um equipamento didático, destinado ao uso por profissionais que estão desenvolvendo outros produtos, e que possui representante no Brasil. Ele deve ser conectado a uma placa de circuito impresso que possua ao menos um microcontrolador, para tratar as informações recebidas do módulo. Pelo fato de também ter saída serial, pode ser ligado diretamente à porta serial do computador e sua saída observada com algum programa de comunicação serial, como por exemplo, o *Hiperterminal* (*Microsoft Windows*).

De acordo com informações disponibilizadas no manual do equipamento, o ME-1000RW é um módulo receptor GPS com elevado grau de integração, que possui uma mini-antena GPS de cerâmica. A antena é conectada ao módulo por um amplificador de baixo nível de ruído (*LNA – Low Noise Amplifier*). O módulo possui 51 canais de aquisição e 14 canais de trajetos, que são capazes de receber sinais de até 65 satélites GPS e transferir a informação com precisão de posição e tempo de informação que podem ser lidos em qualquer porta *UART* (*Universal Asynchronous Receiver/Transmitter*), que é responsável pela implementação do processo de comunicação assíncrona, adicionando bits adicionais para garantir a integridade do sinal, tanto como transmissor e como receptor (codificando e decodificando dados). O *UART* não somente controla a transferência dos dados, mas também a velocidade em que comunicação se estabelece. O tamanho pequeno e alta gama de funcionalidades é disponibilizado com um baixo consumo de energia. Ambos os níveis TTL e interface de sinal RS232 são disponibilizados em um conector da interface, e trabalha com fonte de alimentação entre 3,3 VCC e 6,0 VCC.

O ME-1000RW utiliza um *chipset* (conjunto de circuitos integrados) que o torna altamente sensível, estável e com ótima qualidade de recepção, podendo detectar sinais de satélites de até 158dBm, e possibilitando uma partida à frio inferior a 35 segundos.

Possui led indicador de aquisição e possui bateria de backup interna para manter os dados automaticamente, mesmo que desligado. Fazendo com que seja rápida qualquer posterior reaquisição, mesmo com partida a frio.

Dentre suas diversas aplicações, podemos citar:

- Navegação Marinha e Automotiva
- Rastreamento Veicular
- Localizador de Emergência
- Levantamento Geográfico
- Localização Pessoal
- Recreação e Esporte
- Robótica
- Espionagem
- Dataloggers

A saída de dados deste GPS é padrão NMEA-0183 V3.01. Por ser padrão NMEA, este módulo é compatível com praticamente qualquer programa de GPS.

Quando utilizado em conjunto com um microcontrolador, é possível utilizar qualquer biblioteca compatível com o padrão NMEA.

Possui taxa de atualização de 1 Hz, ou seja, ele atualiza sua posição a cada segundo. Sua conexão de dados é de 9600 bits por segundos, com 8 bits dados, sem paridade e com 1 bit de parada (9600 8 N 1). O relógio interno, bem como informações de satélites, é mantido por uma bateria interna que já vem nele.

Como este módulo já tem antena embutida, sendo desnecessário o uso de antena externa. Isto simplifica o uso e reduz o tamanho projeto.

As seguintes sentenças NMEA (e suas respectivas informações) são fornecidas por padrão por este GPS:

- GGA - Dados de localização, como: horário, latitude, longitude, número de satélites em uso, precisão, altitude em relação ao nível do mar, dados sobre o uso de DGPS;
- GSA - Modo de operação do receptor GPS, satélites usados para navegação e valores de diluição de sinal;
- RMC - Horário, data, posição, direção/sentido da navegação, velocidade;
- VTG - Direção/sentido do deslocamento (se houver deslocamento) e velocidade;
- GSV - Informação individual e coletiva sobre os satélites, como: Número de satélites a vista, identificação de cada um, localização no céu e nível de sinal;

Com relação à antena, vários fatores influenciam a qualidade da recepção. Para uma qualidade de recepção adequada, podemos afirmar que:

- A antena do GPS deve estar apontada para o céu;
- Módulos GPS com sensibilidade como a deste, são feitos para serem utilizados em ambientes a céu aberto (*outdoor*). Ao tentar utilizá-lo em ambientes fechados, a qualidade da recepção ficará seriamente comprometida e poderá inclusive impossibilitar quase que por completo a recepção de informações e a consequente utilidade do produto.
- Barreiras metálicas e de outros tipos de material podem blindar o sinal dos satélites e impedir que o módulo GPS os capte.
- Ambientes com fortes ruídos eletromagnéticos podem causar a perda de sinal do GPS.
- Alterações ou flutuações de voltagem podem prejudicar o bom funcionamento do GPS.

Informações detalhadas do módulo estão descritas abaixo:

Tabela 1: Especificações Técnicas do módulo receptor GPS

Chipset	Skytraq V634LPX
Tipo de Receptor	65 canais de paralelização, código L1 C/A
Velocidade	0.1m/sec
Tempo de Início	Partida Quente (<i>hot start</i>) < 10 segundos Partida Morna (<i>warm start</i>) < 32 segundos Partida Fria (<i>cold start</i>) < 35 segundos
Reaquisição	1s
Sensibilidade	-158dBm aquisição -161dBm rastreamento
Taxa de Atualização	1Hz
Dinâmicas	4G (39.2m/s ²)
Limites Operacionais	Altitude < 18,000m ou velocidade < 515m/s (limite COCOM, um ou outro pode ser excedido, mas não ambos)
Interface Serial	Dupla saída serial: 1 Nível CMOS 1 Nível RS-232
Protocolo	NMEA-0183 V3.01 GPGGA, GPGSA, GPRMC, GPVTG, GPGSV 9600 baud, 8, N, 1 (Default)
Datum	Padrão WGS-84
Interface de Conexão	Conector Molex (part number 51021-0600) Plugue macho (part number 53261-0671)
Tensão de Entrada	3,3VCC ~ 6,0VCC

Consumo	Quando rastreando <= 35mA a 3,3V Máximo de 76mA a 3,3V
Dimensões	33,9 mm x 33,9 mm x 9,2 mm
Peso	14g
Temperatura	Armazenamento: -40°C ~ +80°C Operação: -20°C ~ +65°C
Umidade	5% ~ 95%
Outros	Led indicador de recepção Bateria de backup para partida rápida

2.6 MICROCONTROLADOR

Segundo GIMENEZ (2005), os microcontroladores são semicondutores em forma de CI (circuito integrado), que integram todas as partes básicas de um microcomputador – microprocessador (*CPU*), memórias não-voláteis (*ROM / PROM / EPROM / EEPROM*), memórias voláteis (*RAM / SRAM, DRAM, Flash RAM*), portas de entrada e saída (portas de comunicação paralela, portas de comunicação serial, conversores analógico/digital, conversores digital/analogico, entre outros). Ele é conhecido como um microcomputador implementado em um único CI. Geralmente, é limitado em termos de quantidade de memória, principalmente no que se diz respeito à memória de dados, e é utilizado em aplicações específicas, ou seja, naqueles que não necessitam armazenar grandes quantidades de dados, como automação residencial (fornos microondas, máquinas de lavar, telefones, alarmes residências, automação de portão, entre outros), em automação predial (elevadores, controladores de energia elétrica), em automação industrial (robótica, controladores lógicos programáveis (CLPs), controladores de acesso restrito, relógios de ponto) e na automação embarcada (computadores de bordo, alarmes). Tem a vantagem de apresentar um custo baixo. São vários os fornecedores de microcontroladores. Os principais, atualmente, são: *Motorola, Microchip, Mitsubishi, NEC, Philips, SGS, Intel, Hitachihi, Toshiba*.

2.6.1 Arquitetura Harvard

Os microcontroladores PIC se sustentam em uma arquitetura, denominada Harvard. Baseia-se na existência de duas memórias independentes, uma dedicada a guardar instruções e outra a guardar dados.

Esta independência permite realizar acessos simultâneos a ambas as memórias e buscar uma instrução enquanto se está escrevendo um dado, além disso, a capacidade de cada memória pode se ajustar à necessidade precisa da informação que se guarda, desta forma, a capacidade da memória das instruções pode ser muito maior que a de dados, como convém na maioria das aplicações com microcontroladores. Também se pode diferenciar e adequar às especificações, a longitude nos bits das posições de cada memória.

No caso do PIC16F628A, segundo dados do manual do fabricante, a memória de programa, que é a que guarda as instruções, tem uma capacidade de 2048 posições de 14 bits de comprimento cada uma, e está construída com tecnologia FLASH, que permite escrevê-las e apagá-las eletricamente. A memória de dados tem uma capacidade muito menor e suas posições têm um tamanho de 8 bits, típico para os dados. É volátil, de tipo RAM, embora se complemente com uma pequena memória de dados auxiliares não volátil do tipo EEPROM, para guardar dados de forma permanente, o qual é necessário em algumas aplicações. A independência das duas memórias exige canais de entrada diferenciados, pelo qual, a memória de programa se controla com um canal de entrada de senhas, com linhas suficientes para discriminar 2 K posições e um canal de entrada para transferência das instruções de 14 bits. O canal de entrada de senhas dos dados tem menos linhas porque a memória de dados é de menor capacidade e o canal de entrada pelo qual circulam os dados têm 8 linhas para suportar os bits de cada posição.

2.6.2 MICROCONTROLADOR PIC 16F877A

Para o projeto, inicialmente, havia-se considerado o microcontrolador 16F877A. Conforme o decorrer do desenvolvimento, houve a necessidade de mudança para o microcontrolador 16F628A.

A definição por este microcontrolador foi devido ser um equipamento didático, com muito material de pesquisa e exemplos em livros técnicos e sites especializados no assunto na *Internet*.

O microcontrolador PIC 16F877A, de acordo com as informações disponibilizadas no manual do fabricante, é um microcontrolador da família de 8 bits e núcleo de 14 bits fabricado pela *Microchip Technology*. Possui memória flash de programa com 8192 palavras de 14 bits, memória RAM com 368 bytes e memória EEPROM com 256 bytes. Sua freqüência de operação (clock) vai até 20MHz. Seu conjunto de instruções RISC é composto de 35 instruções. Pode funcionar com alimentação de 2V a 5,5V. Sua pinagem é de 40 pinos.

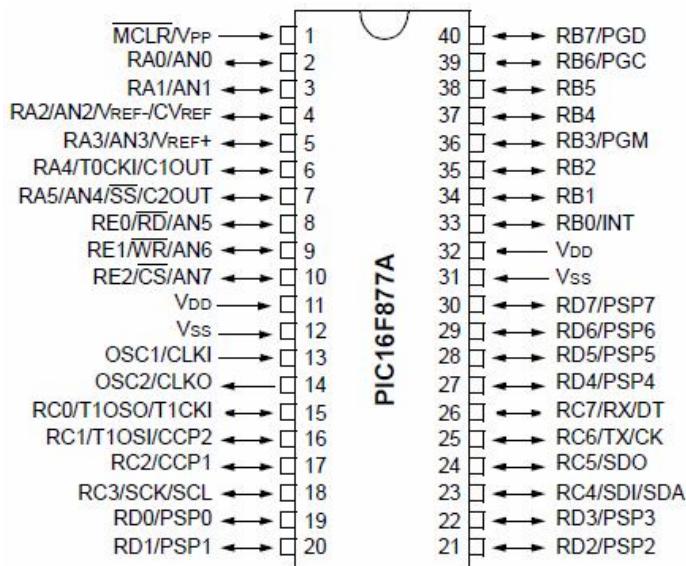


Figura 1: Pinagem do Microcontrolador PIC16F877A

Fonte: Manual do Microcontrolador PIC16F877A

A figura 1 mostra a descrição dos pinos do PIC16F877A, e a figura abaixo (figura 2) mostra o diagrama de blocos do mesmo.

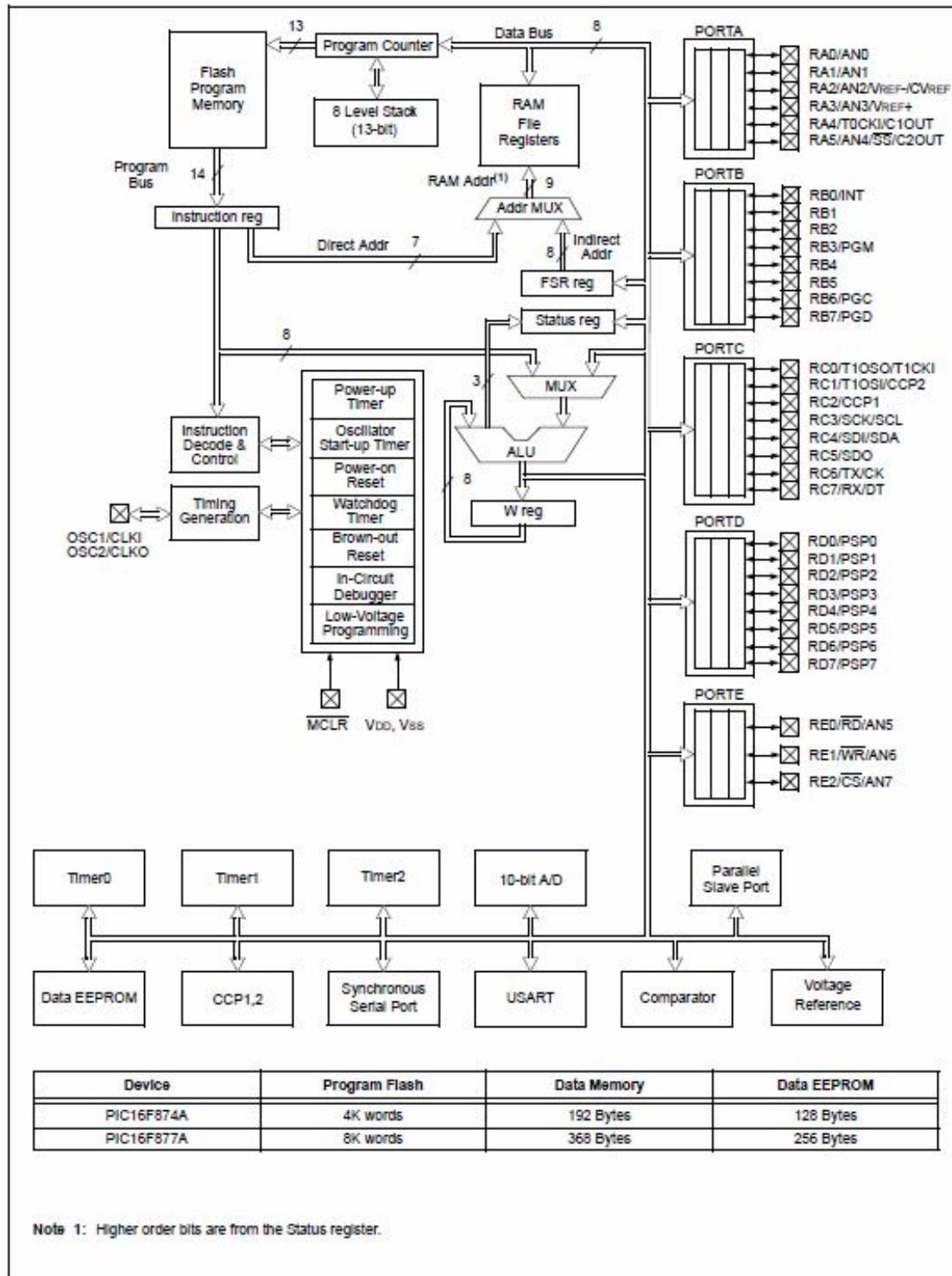


Figura 2: Diagrama de Blocos do Microcontrolador PIC16F877A

Fonte: Manual do Microcontrolador PIC16F877A

Informações detalhadas do microcontrolador estão descritas abaixo:

Tabela 2: Descrição da pinagem do PIC16F877A

Nome do Pino	Pino# PDIP, SOIC, SSOP	Pino# QFN	Tipo E/S/T	Tipo do Buffer	Descrição
OSC1/CLKI OSC1 CLKI	9	6	E E	ST/CMOS(3)	Cristal oscilador ou entrada de <i>clock</i> externo. Entrada de cristal oscilador ou fonte de entrada de <i>clock</i> externo. Buffer ST quando configurado em modo RC; CMOS invertido. Fonte de entrada de <i>clock</i> externo. Sempre associado com o pino de função SC1 (ver pinos OSC1/CLKI, OSC2/CLKO).
OSC2/CLKO OSC2 CLKO	10	7	S S	-	Cristal oscilador ou saída de <i>clock</i> . Saída de cristal oscilador. Conecta ao cristal ou ressonador em modo de cristal oscilador. Em modo RC, pino OSC2 tem 1/4 da freqüência de OSC1 e denota a taxa de ciclo da instrução.
MCLR/VPP MCLR V _{PP}	1	26	E T	ST	Master Clear (entrada) ou tensão de programação (saída). Entrada de Master Clear (Reset). Este pino é um baixo Reset ativo para o equipamento. Entrada de tensão de programação.
RA0/AN0 RA0 AN0 RA1/AN1 RA1 AN1 RA2/AN2/V _{REF} /CV _{REF} RA2 AN2 V _{REF} - CV _{REF} RA3/AN3/V _{REF} + RA3 AN3 V _{REF} + RA4/T0CKI/C1OUT RA4 T0CKI C1OUT RA5/AN4/SS/C2OUT RA5 AN4 SS C2OUT	2 3 4 5 6 7	27 28 1 2 3 4	E/S E E/S E E/S E E S E/S E E E/S E S E/S E E S	TTL TTL TTL TTL ST TTL	PORTA é uma porta de E/S bidirecional. E/S Digital. Entrada analógica 0. E/S Digital. Entrada analógica 1. E/S Digital. Entrada analógica 2. Entrada de tensão (baixa) de referência A/D. Saída do comparador V _{REF} . E/S Digital. Entrada analógica 3. Entrada de tensão (alta) de referência A/D. E/S Digital – Coletor aberto quando configurado como saída. Entrada de <i>clock</i> externo Timer0 Saída do comparador 1. E/S Digital. Entrada analógica 4. Entrada de seleção escravo SPI Saída do comparador 2.

Legenda: E = entrada; S = saída; E/S = entrada/saída; T = tensão; — = Não utilizado;

TTL = entrada TTL; ST = entrada Trigger Schmitt

Nota 1: Este buffer é uma entrada Trigger Schmitt quando configurado com interrupção externa.

Nota 2: Este buffer é uma entrada Trigger Schmitt quando utilizado no modo de Programação Serial.

Nota 3: Este buffer é uma entrada Trigger Schmitt quando configurado no modo Oscilador RC e uma entrada invertida CMOS.

2.6.3 MICROCONTROLADOR PIC 16F628A

O microcontrolador PIC 16F628A, de acordo com as informações disponibilizadas no manual do fabricante, é um microcontrolador da família de 8 bits e núcleo de 14 bits fabricado pela *Microchip Technology*. Possui memória flash de programa com 2048 palavras de 14 bits, memória RAM com 224 bytes e memória EEPROM com 128 bytes. Sua freqüência de operação (clock) vai até 20MHz. Seu conjunto de instruções RISC é composto de 35 instruções. Pode funcionar com alimentação de 2V a 5,5V. Sua pinagem é de 18 pinos.

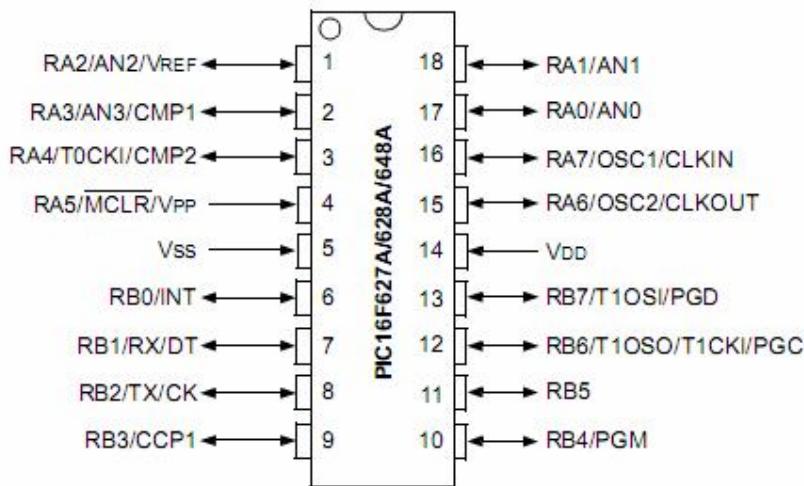


Figura 3: Pinagem do Microcontrolador PIC16F628A

Fonte: Manual do Microcontrolador PIC16F628A

A figura 3 mostra a descrição dos pinos do PIC16F628A, e a figura abaixo (figura 4) mostra o diagrama de blocos do mesmo.

FIGURE 3-1: BLOCK DIAGRAM

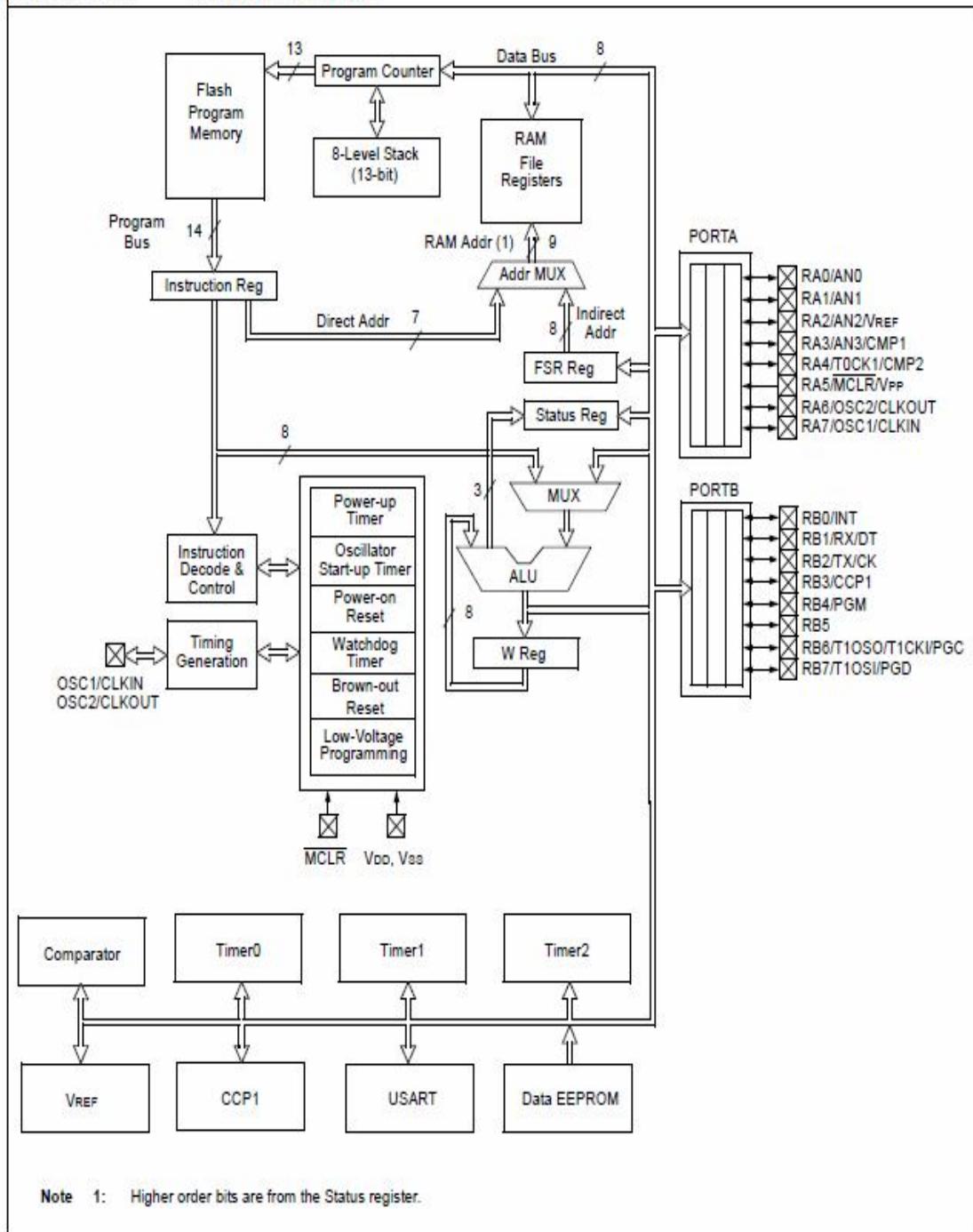


Figura 4: Diagrama de Blocos do Microcontrolador PIC16F628A

Fonte: Manual do Microcontrolador PIC16F628A

2.7 SENsores

Segundo BRESSIATI, PERINI, SILVA, E GANZER, 2009, os sensores são dispositivos capazes de detectar movimentos e ações que ocorrem em processos e projetos eletro-eletrônicos. Em geral, esses elementos são dotados e encarregados de gerar informações para sistemas de automação.

Os sensores são elementos de extrema importância no controle moderno, pois fecham a malha de um circuito, ou seja, eles são capazes de detectar a margem de erro de um sistema e enviar um sinal para um atuador. Com isto, é possível que o processo responda de uma forma mais desejada e pode-se então controlar mais eficientemente um processo (BRESSIATI, PERINI, SILVA, E GANZER, 2009). Um esquema resumido desse fato está mostrado na figura 5, juntamente com a função de transferência.

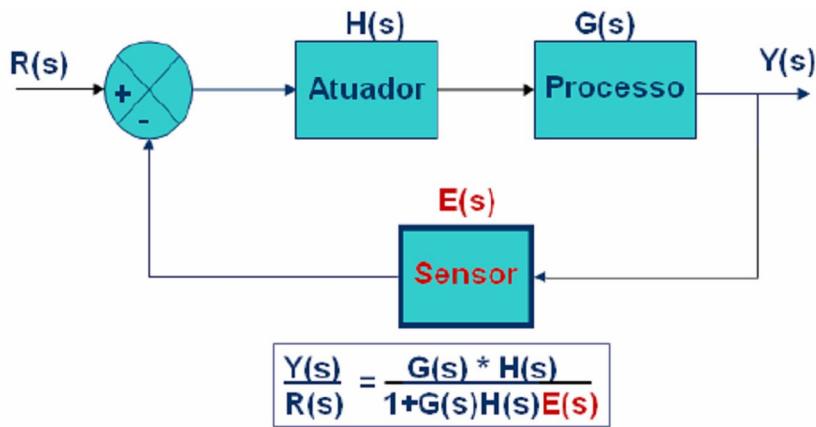


Figura 5: Diagrama de blocos de um sistema de malha fechada

Fonte: BRESSIATI, PERINI, SILVA, E GANZER, 2009

Os sensores tentam emular os cinco sentidos humanos: a visão, a audição, o tato, o olfato e o paladar. No caso dos sensores ópticos, estes emulam basicamente a visão. Os sensores ópticos são fabricados segundo a tecnologia da emissão e recepção de irradiação infravermelha modulada. Eles trabalham em uma faixa de onda correspondente aos raios infravermelhos, ou seja, comprimentos de onda na faixa de 700 nm a 750 nm (BRESSIATI, PERINI, SILVA, E GANZER, 2009).

De modo geral, estes sensores apresentam uma vida útil praticamente infinita e são mais precisos quando comparados a outros tipos de sensores. O princípio de funcionamento geral de um sensor óptico consiste na colocação de emissores e

receptores de luz em uma parte móvel de uma régua graduada, fixa à base do movimento. Isto permite a passagem ou a reflexão apenas de feixes seletivos de luz, que são detectados pelos receptores e assim, podem indicar a posição da parte móvel em relação à parte fixa. Em vista de fatores, como a precisão, por exemplo, estes sensores possuem um custo elevado de fabricação e consequentemente são mais caros que outros sensores (BRESSIATI, PERINI, SILVA, E GANZER, 2009).

2.7.1 Sensores Ópticos

Segundo BRESSIATI, PERINI, SILVA, E GANZER, 2009, os sensores ópticos podem ser divididos em dois tipos, os infravermelhos ativos e os passivos. Dentro dos infravermelhos ativos, podemos caracterizar três grupos principais, de acordo com o seu funcionamento:

- Sistema por Barreira (ópticas alinhadas);
- Sistema por Difusão;
- Sistema por Reflexão.

Um sensor é dito ativo quando este possui um emissor, por onde é emitida uma onda infravermelha, esta invisível a visão humana, e por um receptor que detecta esta onda. Sua utilização é recomendada tanto para ambientes internos como para ambiente externos, pois possuem uma faixa de operação considerável (podem detectar objetos a mais de 100 metros de distância) (BRESSIATI, PERINI, SILVA, E GANZER, 2009).

Os sensores são ditos passivos quando estes possuem apenas receptores, ou seja, eles não emitem ondas infravermelhas, apenas detectam a movimentação destas nas suas áreas de atuação. Ao se detectar um sinal infravermelho, este geralmente é transformado numa variação de tensão ou de corrente e é interpretado por um circuito eletrônico. Seu uso é recomendável principalmente em ambientes fechados, pois a faixa de operação efetiva destes sensores é inferior a dos ativos, e raramente ultrapassa 20 metros (BRESSIATI, PERINI, SILVA, E GANZER, 2009).

Dentro dos infravermelhos ativos, temos aqueles que funcionam no sistema por barreira, onde o elemento transmissor de irradiação infravermelha é alinhado frontalmente a um receptor infravermelho, a uma distância pré-determinada para

cada tipo de sensor. Se ocorrer alguma interrupção desta irradiação, ocorrerá um chaveamento eletrônico, pois não haverá sinal recebido pelo receptor (BRESSIATI, PERINI, SILVA, E GANZER, 2009). Um esquema de um sensor por barreira é mostrado na figura 6.

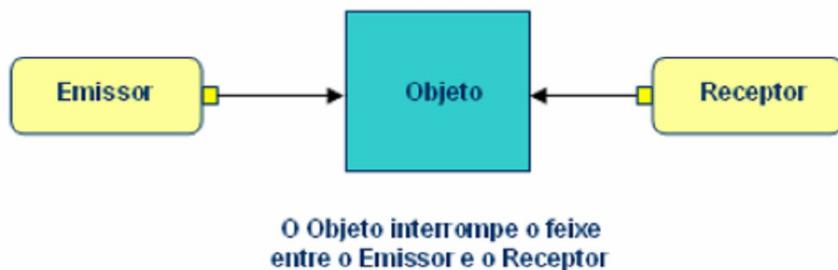


Figura 6: Sensor Óptico por barreira

Fonte: BRESSIATI, PERINI, SILVA, E GANZER, 2009

Os sistemas por difusão e por reflexão caracterizam-se pelos emissores e receptores estarem dispostos lado a lado em um mesmo conjunto óptico (BRESSIATI, PERINI, SILVA, E GANZER, 2009).

Nos sistemas por difusão, os raios infravermelhos emitidos pelo emissor, incidem diretamente sobre um objeto, o qual retorna um feixe de luz em direção ao receptor. De acordo com o tempo de resposta, determina-se a distância na qual o objeto está disposto e então ocorre um chaveamento eletrônico correspondente (BRESSIATI, PERINI, SILVA, E GANZER, 2009). Um esquema é mostrado na figura 7.

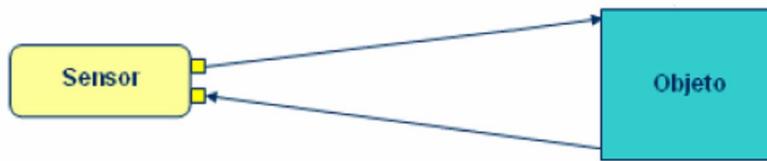


Figura 7: Sensor Óptico por difusão

Fonte: BRESSIATI, PERINI, SILVA, E GANZER, 2009

No sistema por reflexão, os raios infravermelhos são emitidos em direção a um espelho prismático. Neste caso, o chaveamento ocorrerá quando se retirar o espelho ou quando interromper-se a barreira dos raios infravermelhos entre o sensor

e o espelho com um objeto ou corpo de qualquer natureza, fazendo com que o receptor não receba mais um sinal (BRESSIATI, PERINI, SILVA, E GANZER, 2009). Este esquema é mostrado na figura 8.

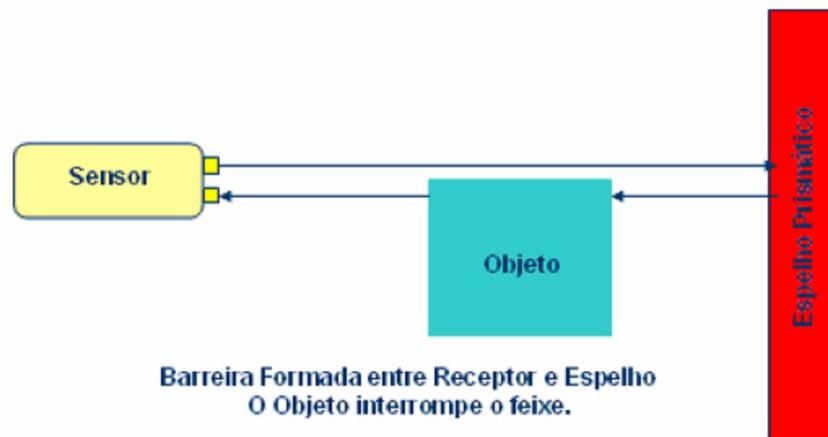


Figura 8: Sensor Óptico por reflexão

Fonte: BRESSIATI, PERINI, SILVA, E GANZER, 2009

2.7.2 Sensores Capacitivos

Segundo CARREIRA E FONSECA, 2009, um sensor ou transdutor capacitivo é um condensador que exibe uma variação do valor nominal da capacidade em função de uma grandeza não elétrica. Uma vez que um condensador consiste basicamente num conjunto de duas placas condutoras separadas por um dielétrico, as variações no valor nominal da capacidade podem ser provocadas por redução da área frente a frente e da separação entre as placas, ou por variação da constante dielétrica do material.

Os sensores capacitivos permitem medir com grande precisão um grande número de grandezas físicas, tais como a posição, o deslocamento, a velocidade e a aceleração linear ou angular de um objeto; a umidade, a concentração de gases e o nível de líquidos ou sólidos; a força, o torque, a pressão e a temperatura; mas também detectar a proximidade de objetos, a presença de água e de pessoas, etc (CARREIRA E FONSECA, 2009).

Hoje em dia existe uma grande variedade de aplicações que utilizam sensores capacitivos, de forma discreta ou integrada. Por exemplo, são bastante comuns os sensores capacitivos de pressão, (caso dos microfones), de aceleração, de fluxo de gases ou líquidos, de umidade, de compostos químicos como o

monóxido de carbono, dióxido de carbono, azoto, de temperatura, de vácuo, de nível de líquidos, de força, de deslocamento, etc., uns detectando as variações na espessura do dielétrico, outros na constante dielétrica. A detecção da variação da capacidade é geralmente efetuada através da medição da carga acumulada, por exemplo através da aplicação de uma tensão constante, ou então indiretamente através da variação da freqüência de oscilação ou da forma de onda à saída de um circuito, do qual o sensor é parte integrante (CARREIRA E FONSECA, 2009). Na Figura 9 apresentam-se os esquemas simplificados de alguns dos sensores capacitivos mais utilizados.

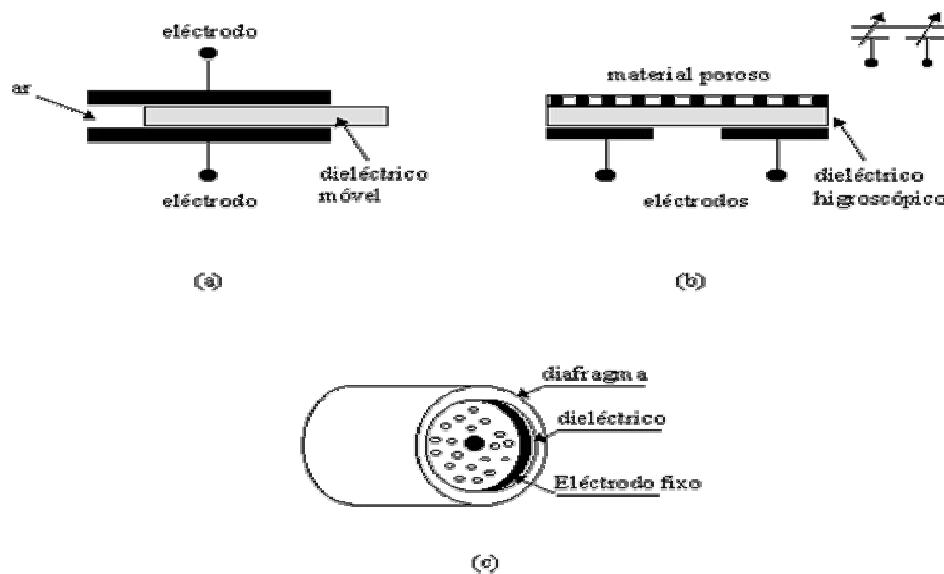


Figura 9: Utilização de um sensor capacitivo de deslocamento

Fonte: CARREIRA E FONSECA, 2009

Neste sensor os dois elétrodos são fixos e estão separados por uma película fina de um material cuja constante dielétrica é superior à unidade ($\epsilon_r > 1$), que se pode deslocar lateralmente em conjunto com o objeto cujo movimento se pretende medir. O deslocamento da película altera a proporção entre as partes dos elétrodos separadas por ar e pela película de material dielétrico, que se traduz numa variação linear da constante dielétrica do conjunto e, em consequência, da capacidade do condensador. Na prática existem diversas variantes deste princípio básico, utilizadas por exemplo na construção de transdutores em rotores e estatores de motores (CARREIRA E FONSECA, 2009)

2.7.3 Sensores de Proximidade Ultrassônicos

Segundo as informações técnicas da empresa NIVETEC, este instrumento tem a função de detectar ou medir a posição de determinado objeto através da emissão de pulsos de ultra-som.

A superfície plana do alvo reflete o ultra-som, que retorna ao sensor como um eco. A variável medida é o tempo de trajeto (de ida e volta) do ultra-som entre a face do sensor e a superfície do objeto, cujo resultado é utilizado no cálculo da distância (levando-se ainda em conta que a velocidade de propagação do ultra-som no ar é praticamente constante) (NIVETEC).

A faixa de medição máxima que o sensor pode monitorar é de 1 ou 6 m, dependendo do modelo escolhido. Apresenta versões com saída PNP (ON/OFF) ou analógica de 4-20 mA ou 0-10 VCC e com invólucro em material termoplástico ou metálico (Aço Inox) (NIVETEC).

O ponto de atuação do sensor com saída PNP pode ser ajustado em qualquer valor da faixa nominal de medição. Do mesmo modo, as versões com saídas analógicas permitem ter a faixa ajustada (valor mínimo e máximo) em quaisquer pontos dentro da faixa nominal de medição, podendo inclusive ser invertidas.

Apresentam proteção contra inversão de polaridade, pequeno ângulo de abertura do feixe de ultra-som (5°) e fácil instalação e configuração, com o auxílio de um magneto de contato (NIVETEC).

Características:

- Funcionamento baseado na emissão de pulsos de ultra-som
- Faixas de medição de 1 e 6 m
- Saídas PNP, 4-20 mA ou 0-10 VCC
- Fácil configuração
- Invólucro em PP ou Aço Inox

Aplicações:

- Detecção de presença ou ausência de objetos em esteiras
- Medição da altura de objetos

- Medição da distância em empilhadeiras
- Monitoramento de continuidade
- Aviso de aproximação.

Na decorrência do protótipo, não foi implementado o uso de sensores, porém existe a sua base teórica para uma futura implementação.

3 DESENVOLVIMENTO

O projeto de correção de rotas de veículos autônomos por GPS foi dividido em três fases: Iniciação, Construção (testes, desenvolvimento e adaptações ao protótipo) e Finalização.

Como base para o desenvolvimento da primeira fase foi utilizado o método Volere (método de levantamento de informações para construção de software que pode ser encontrado em <http://www.volere.co.uk>) com algumas adequações conforme o desenvolvimento do sistema embarcado (*Software e Hardware*).

3.1 PRIMEIRA FASE: Iniciação (Definições, Levantamento de Requisitos, Casos de Uso)

3.1.1 O Propósito do Projeto

A atual tecnologia está trazendo perspectivas jamais imaginadas pelo ser humano, e tornando realidade o que antes eram somente sonhos em imagens de filmes de ficção. A construção desse protótipo poderia ser considerada, até pouco tempo atrás, como impossível e irreal. Porém, com os avanços tecnológicos, será demonstrado um conceito que poderá contribuir muito para o meio acadêmico e científico.

Mediante esse desafio, foi dado início à construção do protótipo de um veículo autônomo que utilizará um módulo receptor GPS (com informações do posicionamento atual do veículo baseada em latitude e longitude), microcontroladores (executando as operações de movimento do veículo), sensores (evitando a colisão do veículo com outro objeto que possa interferir no seu deslocamento) e a programação do software embarcado responsável pelo envio de comandos ao veículo. A proposta é que o estudo aqui realizado sirva como base para ser aplicada em diversos campos futuramente, e que contribua com as necessidades e conforto do ser humano.

3.1.2 Colaboradores e Usuário (Diretos e Indiretos)

Por se tratar de um protótipo de algo que ainda não é encontrado no mercado, não se optou em procurar parceiro e nem especificar um tipo de usuário, já que a idéia é despertar o interesse de diferentes públicos.

3.1.3 Restrições Mandatárias (Cronograma, Materiais e Orçamento)

Durante todo o estudo e construção do protótipo foram definidas algumas restrições como cronograma, custos e ambiente por onde o veículo autônomo deve operar.

3.1.4 Cronograma:

Fevereiro¹: pesquisa de atualidades e tendências;

Fevereiro²: Definição do Tema e escolha do orientador;

Março¹: Levantamento de informações sobre o tema escolhido;

Março²: Reuniões com orientador e entrega do primeiro relatório de atividades;

Abril¹: Entrega da proposta de TCC, apresentação para banca e Relatório das Recomendações da Banca de Qualificação (RRBQ) sobre a apresentação do Pré-projeto;

Abril²: Levantamento de mais informações e aquisição dos componentes eletrônicos para construção do protótipo;

Maio¹: Desenvolvimento da primeira versão do TCC baseado nas informações e desenvolvimentos ocorrido até o momento;

Maio²: Entrega da Primeira Versão Prévia;

Junho¹: Testes individuais dos componentes e desenvolvimento do software;

Junho²: Testes individuais dos componentes e desenvolvimento do software;

Julho¹: Desenvolvimento de Hardware com integração dos componentes;

Julho²: Desenvolvimento de Hardware com integração dos componentes;

Agosto¹: Testes Integrados de Hardware (Componentes Eletrônicos, PIC e Modulo GPS);

Agosto²: Testes Integrados de Hardware e entrega do relatório de atividades 2 (RA

2 – 31/08/2009);

Setembro¹: Desenvolvimento de software, documentação das atividades realizadas;

Setembro²: Desenvolvimento de software, documentação das atividades e finalização da versão prévia 2 do TCC;

Outubro¹: Teste integrado de Hardware e Software e entrega da versão prévia 2 do TCC (05/10/2009);

Outubro²: Teste integrado de Hardware e Software. Escrita do trabalho final e melhorias do projeto;

Novembro¹: Escrita do trabalho final, revisão e entrega do mesmo (16/11/2009);

Novembro²: Ajuste (caso necessário) e apresentação do TCC para Bancada.

3.1.5 Materiais

- Compra de veículo de controle remoto.

Esse Jipe (figura 10) em escala 1:20 foi produzido com suspensão nas quatro rodas e pneus aderentes. É comandado por rádio controle possui 7 funções e pode atingir altas velocidades.



Figura 10: Jipe em escala 1:20 (Brinquedo)

Fonte: www.candide.com.br

Foi feita a abertura da parte superior do mesmo e levantamento do circuito eletrônico junto ao nosso orientador. O CI utilizado no mesmo é o modelo SM6135W. Seu manual pode ser encontrado no link:

<http://www.alldatasheet.com/datasheet-pdf/pdf/152941/ETC/SM6135W.html>.

- Compra do microcontrolador PIC16F877A



Figura 11: Microcontrolador PIC 16F877A

Fonte: www.labtools.com.br

Com base em estudos iniciais, ficou definido que o microcontrolador a ser utilizado seria o PIC16F877A (figura 11), por suas características técnicas e por ser muito utilizado didaticamente. Seu manual pode ser encontrado no link: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>.

- Compra do microcontrolador PIC16F628A

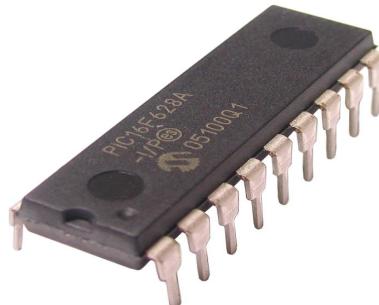


Figura 12: Microcontrolador PIC 16F628A

Fonte: www.microchip.com

Após novas avaliações, foi reconsiderado o uso do microcontrolador PIC16F628A (figura 12), por suas características técnicas e por ser mais simples de trabalhar, atendendo as necessidades do projeto até esta etapa. Seu manual pode ser encontrado no link:

<http://ww1.microchip.com/downloads/en/DeviceDoc/40044F.pdf>.

- Compra do módulo receptor GPS

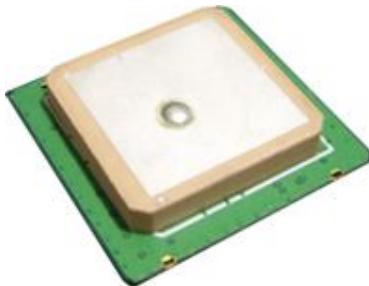


Figura 13: Módulo GPS ME-1000RW

Fonte: Manual do fabricante

A definição deste módulo receptor GPS (figura 13) foi devido ser um dos poucos comercializados e com suporte pelo distribuidor no Brasil. Seu manual pode ser encontrado no link: <http://res.trilha21.com/001000608/files/ME-1000RW.pdf>.

Outros equipamentos como computador, gravador de microcontrolador e software de programação em linguagem C, também fazem parte da composição dos nossos instrumentos utilizados neste projeto.

- Compra do circuito integrado L293D



Figura 14: Circuito Integrado L293D

Fonte: <http://www.active-robots.com/products/components/chips-ics/l293d.jpg>

O circuito integrado L293D (figura 14) foi escolhido para acionar os motores do veículo, e eliminando assim todo o circuito do veículo escolhido para os testes (deixando apenas os motores para os movimentos de direita /esquerda, frente/trás).

Este CI trabalha com tensões de 5 VCC até 36 VCC, e corrente máxima de 600 mA por porta. A excitação desta porta é feita com nível TTL (5 VCC), o que garante a compatibilidade com microcontroladores. Seu manual pode ser encontrado em: <http://www.datasheetcatalog.org/datasheet/stmicroelectronics/1330.pdf>

3.1.6 Orçamento

Tabela 3: Lista de Materiais adquiridos para construção do protótipo

Material	Valor	Fornecedor
Módulo GPS ME-1000RW	R\$ 137,90	www.Trilha21.com
Carro de Controle Remoto	R\$ 49,90	www.americanas.com
Microcontrolador PIC16F877A	R\$ 29,90	www.labtools.com.br
Gravador de PIC USB	R\$ 110,00	www.mercadolivre.com.br
Cabo conversor USB / Serial	R\$ 45,00	Eletrotel (Santo André, SP)
Outros (comp. eletrônicos, solda, etc)	R\$ 160,00	Eletrotel (Santo André, SP)
Circuito Integrado L293D	R\$ 10,00	Santa Efigênia
Circuito Integrado Max232N	R\$ 6,00	Santa Efigênia
Livro PIC: Programação em C	R\$ 106,00	www.submarino.com.br
2x microcontroladores PIC16F628A	R\$ 22,00	www.mercadolivre.com.br
Total Gasto	R\$ 676,70	

3.1.7 Fatos e Suposições

Na utilização do módulo GPS é necessário um ambiente aberto para que ele possa sincronizar com os satélites presentes na órbita da do planeta Terra. Também é preciso levar em consideração todos os meios naturais que possam interferir no deslocamento do veículo, já que o mesmo trabalha com freqüências de recepção da informação do GPS.

Como não é utilizada uma interface gráfica (mapas de ruas), as coordenadas podem ser de destino com ou sem obstáculos, mas como amostragem, utilizou-se um ambiente livre dos mesmos.

Há uma extensa procura por componentes e programas que facilitem a construção e programação do protótipo (para programação do PIC e construção dos desenhos técnicos é sempre optado por ferramentas de softwares livres, ou seja, sem custo de aquisição).

3.1.8 Levantamento de Requisitos, Regras de Negócio e Casos de Uso

Com as regras de negócio e requisitos, foram estipuladas algumas diretrivas e caminhos a serem seguidos no desenvolvimento prático do projeto. Já os casos de uso descrevem de forma abreviada as ações necessárias para atingir o objetivo.

Requisitos funcionais

- O protótipo deve permitir que um usuário cadastre as coordenadas de destino do veículo no software de programação do microcontrolador (*CCS Compiler*).

Regras de Negócio

- O software embarcado do protótipo deve estar armazenado no PIC e pronto para executar as rotinas e tarefas (chegada ao destino informado).
- O local de teste deve ser em uma área aberta, para que o módulo receptor GPS consiga se comunicar com os satélites da órbita terrestre.
- Por se tratar de um protótipo (feito a partir da estrutura de um carrinho de controle remoto), o veículo deverá possuir limitações no que diz respeito à distâncias percorridas (no máximo 20 metros).
- O software embarcado deverá considerar uma margem de erro (para mais ou para menos) nas coordenadas de destino do veículo.
- O veículo deverá tomar a decisão de destino para a longitude.
- É necessária a implantação de sensores para evitar a colisão do veículo (para o projeto não foram implementados sensores).

Casos de Uso

Número: 01

Caso de uso: Movimentação do Veículo

Descrição: Esse caso de uso descreve as ações que o veículo deve tomar durante a sua busca do endereço (longitude) para qual foi designado.

Autor: Veículo autônomo

Pré-Condições: Ter as coordenadas de destino previamente cadastradas

Pós-Condição: O veículo chegar ao seu destino

Fluxo Principal:

1. Sistema compara a coordenada atual com a coordenada de Longitude informada;
2. Sistema inicia movimento do veículo;
3. Veículo atinge a coordenada de Longitude;
4. Veículo para de se movimentar.

Fluxo Alternativo:

- 1.1. Sistema identifica que já está no local da latitude de destino;
- 1.2. Vai para o passo 4.

- 3.1. Sistema identifica que ultrapassou a latitude de destino;
- 3.2. Vai para o passo 4.

3.2 SEGUNDA FASE: Construção (Teste, desenvolvimento e adaptações).

3.2.1 Primeiro experimento:

Foram instalados e configurados os *drivers* e software do gravador de PIC, onde a operação foi simples e sem nenhum problema.

Em seguida, foram instalados os compiladores MikroC e CCS (que instalam uma biblioteca com diversos exemplos, incluindo o modelo do microcontrolador utilizado).

Foi utilizado o software de gravação que acompanha o gravador de PIC, chamado *PicBurner*, para transferir o arquivo .HEX (compilado) do arquivo de exemplo de led piscando (que acompanha a biblioteca do software CCS). O processo ocorreu sem nenhum problema.

3.2.2 Segundo experimento:

Após a gravação do programa do microcontrolador, foi montado o circuito correspondente no *protoboard* para verificar o funcionamento do mesmo.

Não foi obtido sucesso nesta etapa, e depois da verificação em sites especializados e literatura técnica, foi descoberto que o tipo de oscilador externo que foi utilizado não era correto.

Após a correção do oscilador externo no *protoboard*, houve problemas em iniciar o PIC16F877A em conjunto do cristal oscilador e os capacitores do circuito de oscilação. Resumindo, o microcontrolador não iniciou o programa de teste de piscar um led.

Foi utilizado o PIC16F628A que veio de brinde com a placa do gravador de PIC. Com este microcontrolador, foi possível iniciar o programa teste no *protoboard*.

A instrução que o programa executou foi acionar uma saída do microcontrolador com tempo especificado (2 segundos) para ligar o LED e depois desligar a saída por um segundo, conforme mostra as figuras 15 e 16.



Figura 15: Experimento com Microcontrolador

Fonte: Foto tirada durante o experimento

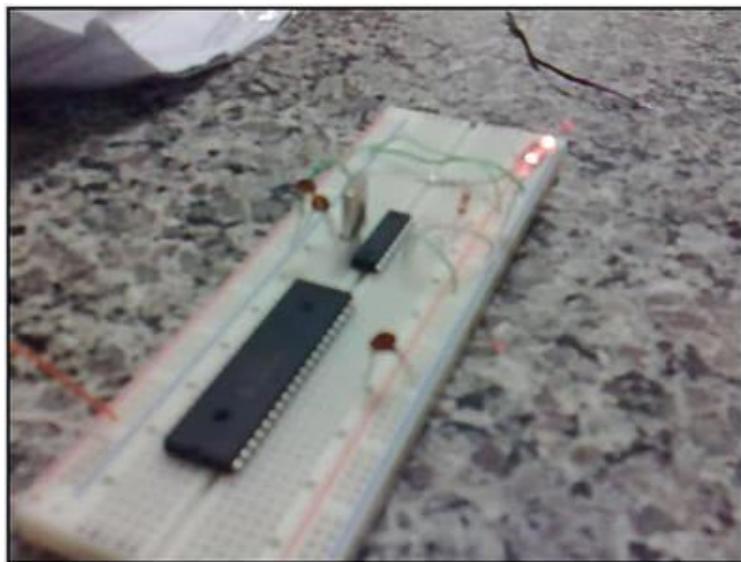


Figura 16: Experimento com Microcontrolador 2

Fonte: Foto tirada durante o experimento

3.2.3 Terceiro experimento:

Foi realizada a verificação da pinagem dos I/O's do microcontrolador utilizado no veículo com o *datasheet* do mesmo. Para isso, foi utilizado um multímetro na escala de tensão (corrente contínua), foi colocado a ponta de teste do comum no referencial de terra e a ponta de teste do positivo no pino correspondente

de ‘virar para esquerda’ e acionamos o movimento em questão no controle do veículo.

Foi verificado que o esquema de pinagem do microcontrolador estava correto e que a tensão de saída do mesmo é 2,5VCC. Foi repetido este passo para os demais comandos (virar para a direita, ir para frente e para trás).

3.2.4 Quarto experimento:

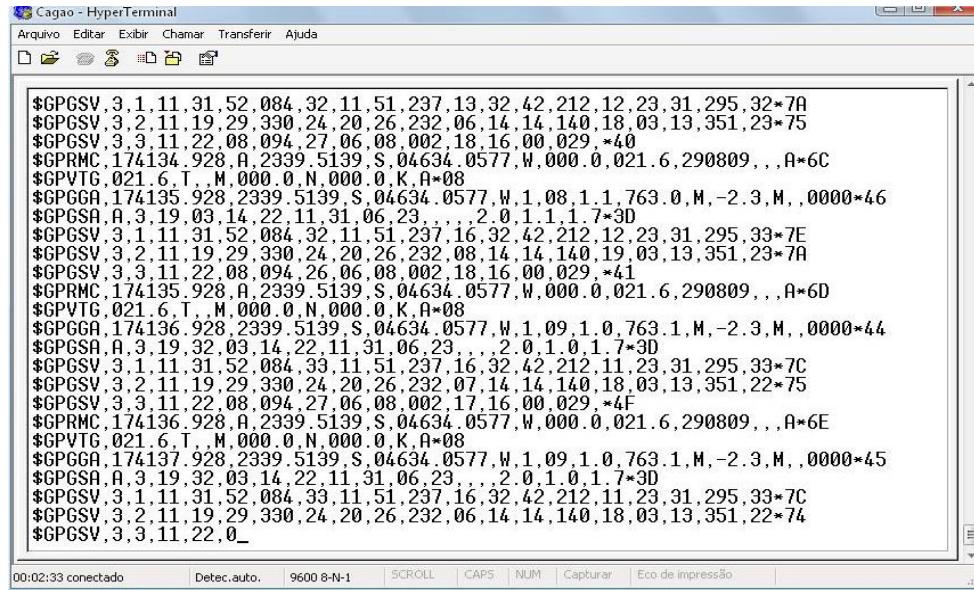
Foi ligado o GPS (energizando com 4,4VCC) e conectado os pinos de TX, RX e GND do equipamento aos pinos 2, 3 e 5 de um conector DB9 fêmea. Ligando o mesmo ao conversor USB / Serial e conectamos ao PC.

Em seguida, foi configurado o software *HyperTerminal* do *MS Windows XP Pro* para comunicação com o GPS. Para isso, foi selecionado a porta serial do PC (que é emulada através do software do conversor USB / Serial), configurado a velocidade de *baud rate* em 4800, bits de dados em 8, sem paridade, um bit de parada e sem controle de fluxo.

O resultado obtido foi a leitura de caracteres em linguagem de máquina, impossibilitando a interpretação dos dados.

Em contato com o fabricante e o mesmo informou que o *datasheet* informava a velocidade de comunicação errada: ao invés de 4800bps, o correto era 9600bps.

Foi modificado essa configuração e foi obtido com sucesso na aquisição dos dados do GPS pelo *HyperTerminal* do *MS Windows* em formato NMEA, como é demonstrado na figura 17.



The screenshot shows a window titled "Cagao - HyperTerminal" with a menu bar: Arquivo, Editar, Exibir, Chamar, Transferir, Ajuda. The main area displays a series of GPS data messages in a monospaced font. The messages include fields such as \$GPGSV, \$GPGSV, \$GPRMC, \$GPVTG, \$GPGGA, \$GPGSA, \$GPGSV, \$GPGSV, \$GPRMC, \$GPVTG, \$GPGGA, \$GPGSA, \$GPGSV, \$GPGSV, and \$GPGSV. The data includes coordinates (lat, lon), date/time, and various status flags. At the bottom of the terminal window, there are several status indicators: 00:02:33 conectado, Detec. auto., 9600 N-1, SCROLL, CAPS, NUM, Capturar, and Eco de impressão.

```
$GPGSV,3,1,11,31,52,084,32,11,51,237,13,32,42,212,12,23,31,295,32*7A
$GPGSV,3,2,11,19,29,330,24,20,26,232,06,14,14,140,18,09,13,351,23*75
$GPGSV,3,3,11,22,08,094,27,06,08,002,18,16,00,029,*40
$GPRMC,174134,928,A,2339,5139,S,04634,0577,W,0000.0,021.6,290809,,,A*6C
$GPVTG,021,6,T,M,000.0,N,000.0,K,A*08
$GPGGA,174135,928,2339,5139,S,04634,0577,W,1,08,1,1,763.0,M,-2,3,M,,0000*46
$GPGSA,A,3,19,03,14,22,11,31,06,23,,,2,0,1,1,1,7*3D
$GPGSV,3,1,11,31,52,084,32,11,51,237,16,32,42,212,12,23,31,295,33*7E
$GPGSV,3,2,11,19,29,330,24,20,26,232,08,14,14,140,19,03,13,351,23*7A
$GPGSV,3,3,11,22,08,094,26,06,08,002,18,16,00,029,*41
$GPRMC,174135,928,A,2339,5139,S,04634,0577,W,0000.0,021.6,290809,,,A*6D
$GPVTG,021,6,T,M,000.0,N,000.0,K,A*08
$GPGGA,174136,928,2339,5139,S,04634,0577,W,1,09,1,0,763.1,M,-2,3,M,,0000*44
$GPGSA,A,3,19,32,03,14,22,11,31,06,23,,,2,0,1,0,1,7*3D
$GPGSV,3,1,11,31,52,084,33,11,51,237,16,32,42,212,11,23,31,295,33*7C
$GPGSV,3,2,11,19,29,330,24,20,26,232,07,14,14,140,18,03,13,351,22*75
$GPGSV,3,3,11,22,08,094,27,06,08,002,17,16,00,029,*4F
$GPRMC,174136,928,A,2339,5139,S,04634,0577,W,0000.0,021.6,290809,,,A*6E
$GPVTG,021,6,T,M,000.0,N,000.0,K,A*08
$GPGGA,174137,928,2339,5139,S,04634,0577,W,1,09,1,0,763.1,M,-2,3,M,,0000*45
$GPGSA,A,3,19,32,03,14,22,11,31,06,23,,,2,0,1,0,1,7*3D
$GPGSV,3,1,11,31,52,084,33,11,51,237,16,32,42,212,11,23,31,295,33*7C
$GPGSV,3,2,11,19,29,330,24,20,26,232,06,14,14,140,18,03,13,351,22*74
$GPGSV,3,3,11,22,0
```

Figura 17: Experimento com GPS

Fonte: Imagem da tela gerada durante os testes

3.2.5 Quinto experimento:

Após os testes no *protoboard*, foi decidido utilizar uma placa universal (figura 18) para soldar os componentes, devido às falhas de mau contato com o *protoboard*.

Foi elaborado outro programa em linguagem C no compilador CCS, ativando 4 portas do microcontrolador para simular movimentos (aleatórios) com o veículo.

O microcontrolador foi gravado e foi testado com o multímetro, onde observou-se a tensão variando nas portas utilizadas entre 0 volt e 5 volts (conforme programado).

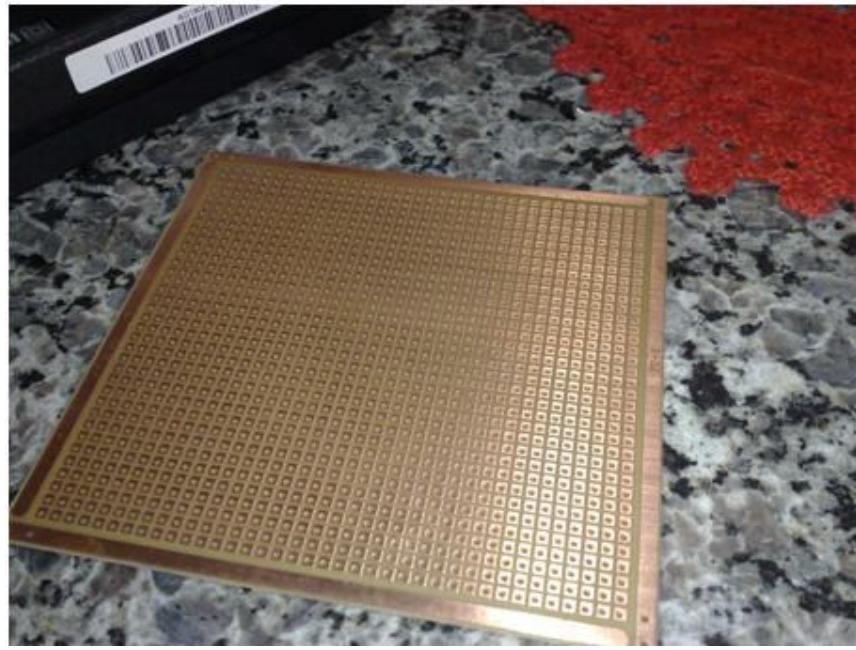


Figura 18: Placa Universal

Fonte: Imagem retirada durante experimento

3.2.6 Sexto experimento:

Foi realizada a conexão entre as placas do veículo e do protótipo e houve um problema, pois foram utilizadas as saídas de movimento do microcontrolador do veículo (a fim de utilizar o circuito do mesmo para alimentação do motor) com as saídas das portas do microcontrolador do protótipo, porém cada placa opera em uma tensão diferente. A placa do protótipo trabalha com 5,0 VCC e do veículo trabalha com 3,6 VCC.

Devido à diferença de tensão encontrada, foram utilizados os transistores BC548 (NPN) para realizar o chaveamento em cada porta de saída do protótipo para placa do veículo, onde foi obtido sucesso no teste como demonstra na figura 19 o circuito construído, e na figura 20 o protótipo em movimento.

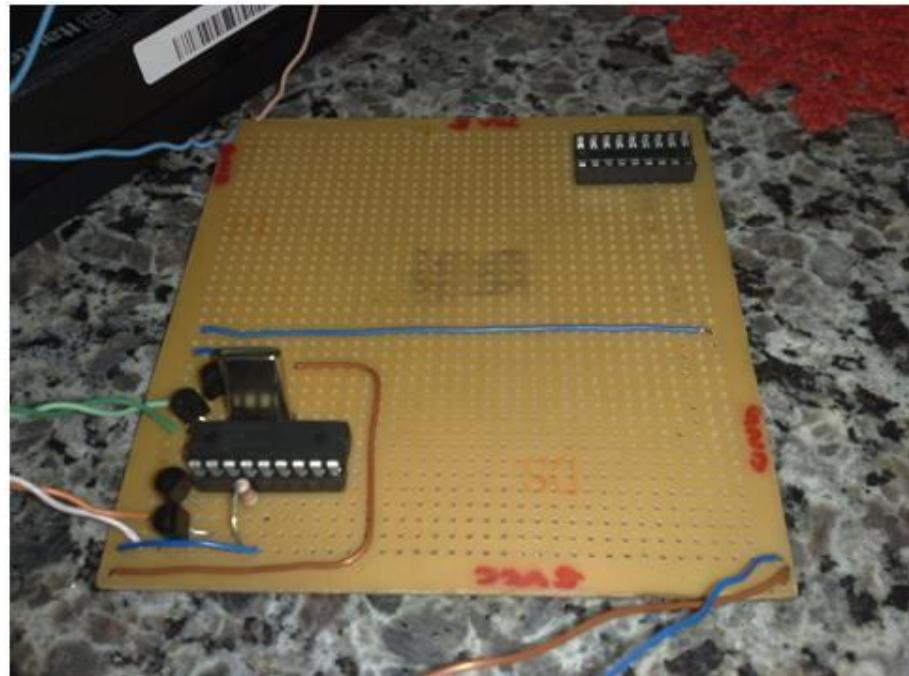


Figura 19: Detalhe do circuito do protótipo

Fonte: Imagem retirada durante experimento

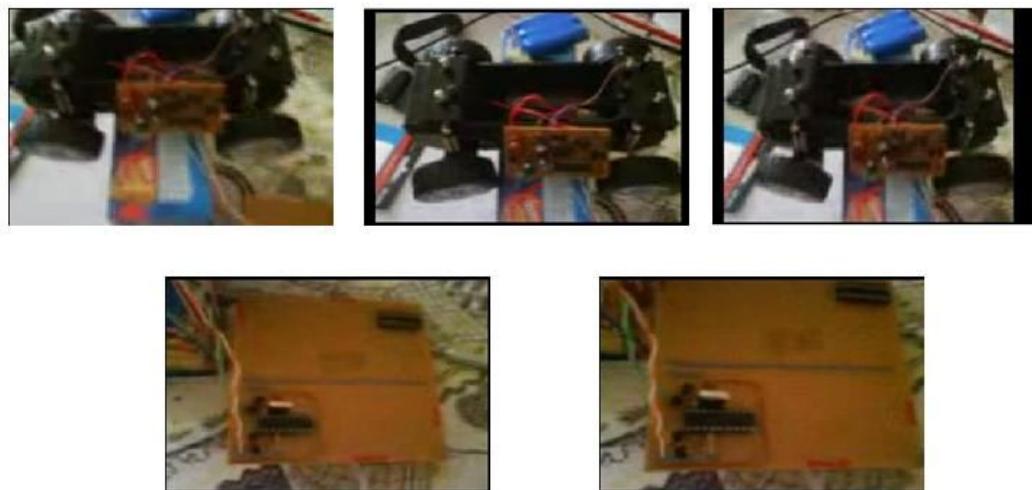


Figura 20: Circuito do protótipo e veículo em movimento

Fonte: Imagens retirada de vídeo gravado durante experimento

3.2.7 Sétimo experimento:

Decidiu-se remover a placa eletrônica original do veículo, a fim de eliminar a necessidade de chaveamento de tensão, e manter a originalidade do protótipo.

Foi realizada a remoção da placa do veículo (figura 21), e foi constatada a necessidade da construção de duas pontes H, que invertem a polaridade no motor elétrico, onde uma ponte H faz o veículo andar para frente e para trás, e outra ponte H faz o veículo virar para direita e esquerda.

Foi utilizado um tutorial do site Maxwell Bohr para construção da mesma (<http://www.maxwellbohr.com.br/downloads/Tutorial%20Eletronica%20-%20Montagem%20de%20uma%20Ponte%20H.pdf>), utilizando-se 2 transistores BC558 PNP e 2 transistores NPN BC548 em conjunto com 4 diodos para proteção do circuito (figura 22). Ao testar a ponte em conjunto com o motor do veículo, foi constatado que a tensão de saída da ponte H resultou em uma tensão de 2.6V, e a corrente foi de 150mA, através da medição com o multímetro em escala de corrente ligado em série ao fio do motor na saída da ponte H. O resultado obtido foi de movimentação fraca dos eixos esquerda e direita, e baixa rotação do motor de aceleração.

Para solução do problema foram realizadas pesquisas na internet e consultas aos *datasheets* dos componentes envolvidos. Encontrou-se então o circuito integrado L293D, que realiza a função de 2 pontes H, garantindo a corrente de 500mA que alimenta corretamente o motor do veículo.

Realizando testes com o CI L293D em conjunto com o microcontrolador (figuras 23 e 24), o veículo respondeu aos comandos pré-programados de aceleração por 2 segundos e virar para esquerda e para direita por 1 segundo.

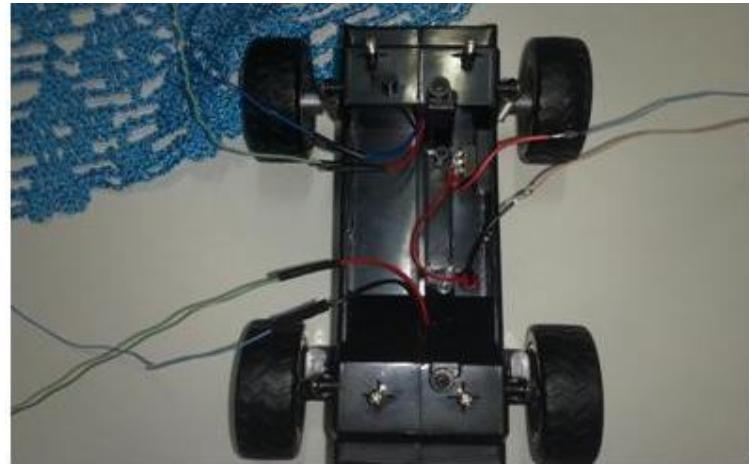


Figura 21: Veículo sem a Placa controladora

Fonte: Imagem retirada durante experimento



Figura 22: Circuito Ponte H

Fonte: Imagem retirada durante experimento



Figura 23: Circuito Integrado L293D (ponte H) em conjunto com o veículo

Fonte: Imagem retirada de vídeo gravado durante experimento

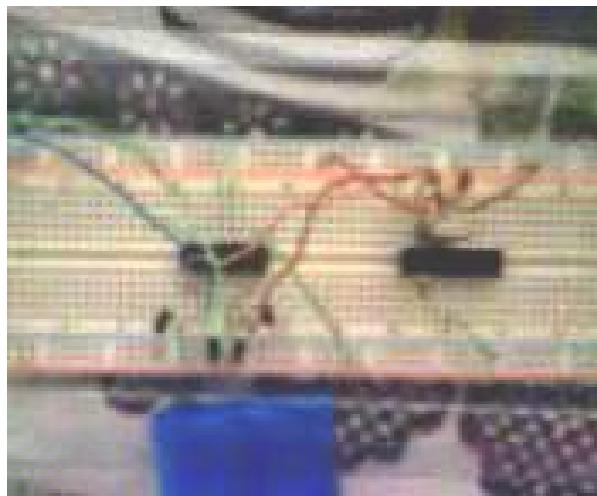


Figura 24: Circuito Integrado Ponte H em conjunto com microcontrolador

Fonte: Imagem retirada de vídeo gravado durante experimento

3.2.8 Oitavo experimento:

Após a conclusão dos testes de controle entre o microcontrolador e o veículo, foram realizados testes de comunicação entre o microcontrolador e o GPS.

Como o microcontrolador possui porta serial com protocolo USART, foi necessário a utilização do circuito integrado MAX232N para converter o sinal TTL do microcontrolador para RS-232, possibilitando a comunicação com um PC pela porta serial, utilizando-se o software *HyperTerminal* do sistema operacional *Windows XP*.

Para isso, foi criado um programa que imprimia um texto quando o programa entrava em modo de execução. Utilizou-se uma biblioteca de autoria de Fábio Pereira, retirada do livro Microcontroladores PIC: Programação em C, para o funcionamento da comunicação serial.

Com o circuito montado, o microcontrolador foi gravado com o programa de teste da nova biblioteca, e foi montado um cabo serial padrão para interligação do *protoboard* ao computador (figura 25). Após configurar o *HyperTerminal* com a velocidade de 9600bps, sem paridade, 1 bit de parada, 8 bits de dados e sem controle de hardware, a conexão com o circuito eletrônico foi bem sucedida. Ao energizar o *protoboard*, a mensagem começou a aparecer no *HyperTerminal*. Em seguida, foram realizados alguns testes baseados em programação de laços

infinitos, para a verificação da recepção da mensagem, e obtiveram-se resultados satisfatórios.

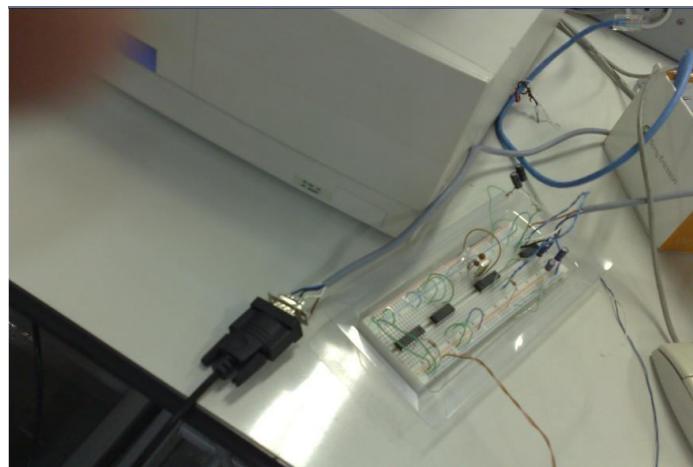


Figura 25: Teste de transmissão com MAX232N

Fonte: Imagem retirada durante experimento

Em seguida foi ligado o pino de TX do GPS ao pino de RX do microcontrolador. A configuração de transmissão do microcontrolador para o CI MAX232 foi mantida, para a transmissão serial de dados para o computador. Manteve-se a biblioteca utilizada nos testes anteriores, para a recepção dos dados transmitidos pelo GPS. O teste procedeu com sucesso, e todas as informações do GPS foram visualizadas pelo *HyperTerminal* (figura 26), sendo atualizadas a cada segundo, e conforme deslocou-se o GPS, pode-se ter certeza que o mesmo alterava as informações enviadas ao microcontrolador, e o mesmo na transmissão para o PC.

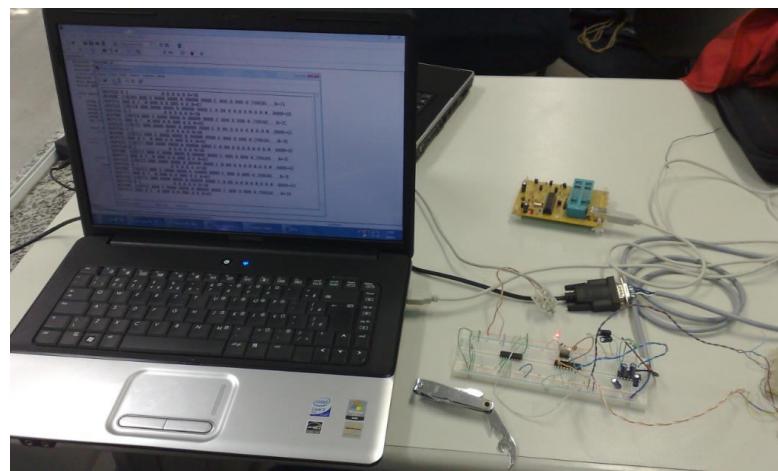


Figura 26: Teste de transmissão com GPS, microcontrolador e MAX232N

Fonte: Imagem retirada durante experimento

Após os testes de comunicação com o GPS, iniciou-se a programação para executar a filtragem das informações necessárias para que o algoritmo do veículo funcionasse como planejado, com a extração da latitude e longitude. Porém, ao religar o circuito no *protoboard* para os testes do filtro criado, começou a apresentar falhas no *HyperTerminal*.

As informações do GPS que eram tratadas no microcontrolador e retransmitidas ao PC começaram a falhar, trazendo caracteres incompreensíveis no *HyperTerminal*. Após um bom tempo de pesquisa e testes realizados no circuito, foi constatado que o problema era de aterramento ou mau contato, pois ao tocar com o dedo em um terminal terra de qualquer parte do circuito os caracteres voltavam ao normal.

Decidiu-se então montar o circuito em uma placa universal de fenolite (conforme figura 27), pois até então os testes estavam sendo executados com o circuito montado no *protoboard*. O circuito foi montado na placa universal, e o problema de aterramento ou mau contato foi resolvido.

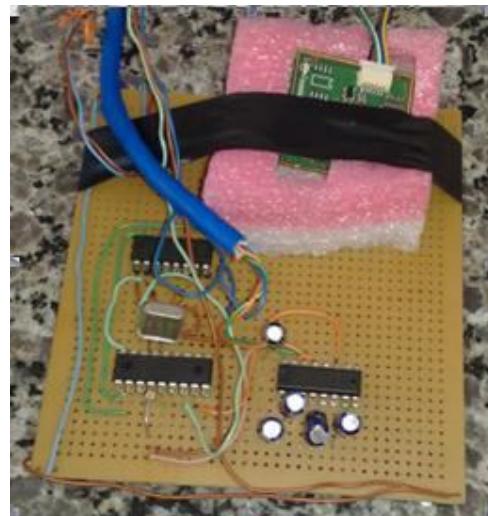


Figura 27: Foto do protótipo da placa na placa universal
Fonte: Imagem retirada durante experimento

3.2.9 Nono Experimento

Neste experimento, iniciaram-se os testes de diversos algoritmos para filtrar dos dados enviados pelo GPS a sentença que iniciasse com a '\$GPGGA', que

contém as informações de latitude e longitude, necessárias para funcionamento do algoritmo proposto inicialmente.

Após muitos testes, foi criado o filtro da primeira frase do GPS em um vetor de *strings* de 41 posições (o qual não suportaria toda a frase devido a pouca quantidade de memória *RAM* no microcontrolador, mas que era suficiente para coletar a latitude e a longitude, sempre apresentadas em formato padrão), que apresentou informações no *HyperTerminal*. Porém, observou-se que as primeiras informações eram a de inicialização do chip do GPS, e que não era necessário armazená-las no vetor.

O algoritmo construído estabelece que para que o filtro da sentença seja acionado, é feita a verificação caractere a caractere (recebidos pela porta serial do microcontrolador conectada ao GPS) dos cinco primeiros caracteres após a verificação do caractere '\$'. Caso a verificação corresponda a 'GPGGA', o algoritmo armazena um vetor de caracteres de latitude e longitude.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	NUL	(null)	32	20	040	 	Space	64	40	100	@	Ø	96	60	140	`	`
1	1 001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2 002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3 003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4 004	EOT	(end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5 005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6 006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7 007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8 010	BS	(backspace)	40	28	050	({	72	48	110	H	H	104	68	150	h	h
9	9 011	TAB	(horizontal tab)	41	29	051)	}	73	49	111	I	I	105	69	151	i	i
10	A 012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B 013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C 014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D 015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E 016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F 017	SI	(shift in)	47	2F	057	/	:	79	4F	117	O	O	111	6F	157	o	o
16	10 020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11 021	DCL1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12 022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13 023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14 024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15 025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16 026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17 027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18 030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19 031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A 032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B 033	ESC	(escape)	59	3B	073	;	/	91	5B	133	[[123	7B	173	{	{
28	1C 034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D 035	GS	(group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E 036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F 037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Figura 28: Tabela ASCII

Fonte: <http://www.virose.pt/ml/blogs/a2m/?p=471>

Quando o vetor de caracteres é completamente preenchido, é realizada uma conversão de caracteres para inteiro, onde foi consultada a tabela ASCII (figura 28) para o cálculo. Os valores de latitude e longitude (tanto a parte inteira, quanto a

parte fracionária) são armazenados em variáveis de números inteiros, em seguida são convertidos os quatro números da parte fracionária, armazenando os novos valores em variáveis de números reais.

O processo de teste era muito dispendioso, sempre era necessário remover o microcontrolador da placa universal, encaixar no soquete do gravador, gravar o novo programa, remover e inseri-lo novamente no soquete específico da placa universal. Realizou-se então pesquisa em sites especializados e revistas técnicas, onde encontrou-se o software *Proteus Isis*, um simulador de circuitos eletrônicos que possibilita a simulação do circuito em conjunto com a programação interna do microcontrolador, apenas apontando o caminho do arquivo .HEX gerado na compilação do código-fonte no compilador CCS. O mesmo simula a rotina programada no microcontrolador fielmente, mostrando-se muito útil e prático. O único problema é que a base de dados de componentes do software simulador não possui GPS. Para contornar isso, foi colocado um terminal virtual para que se pudesse simular o envio de dados que seriam tratados pelo microcontrolador. Para visualizar a maneira como o software faria o cálculo, foi inserido um *display* de cristal líquido (*LCD – Liquid Crystal Display*) para visualização das informações. O circuito de testes montado pode ser observado na figura 29.

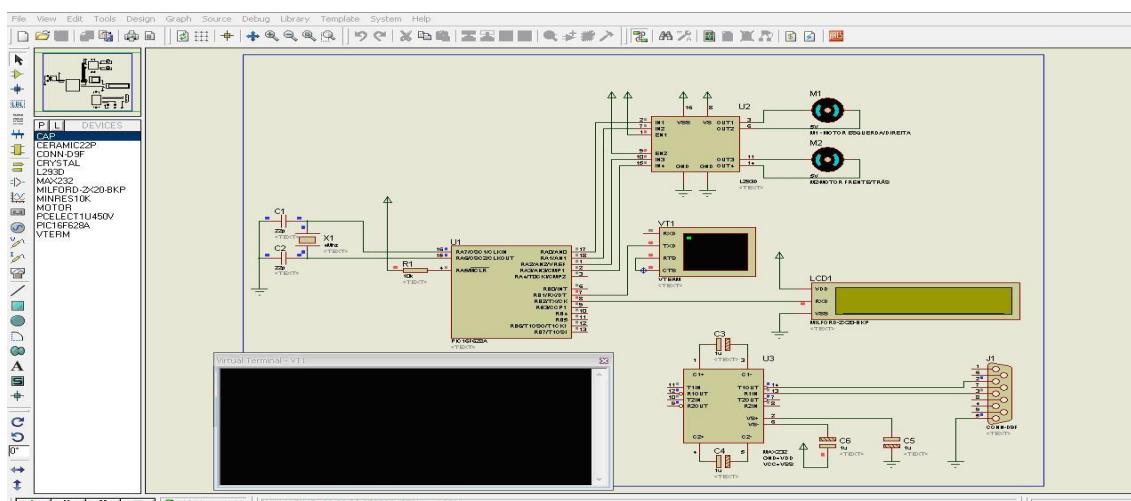


Figura 29: Circuito de simulação montado no *Proteus Isis*

Fonte: Imagem da tela gerada durante os testes

O software desenvolvido foi testado com comando de impressão (*printf*) em linguagem de programação C, para verificar se o microcontrolador estava filtrando a sentença do GPS corretamente. Este teste do software funcionou, porém ao realizar operações lógicas de comparação para realizar os movimentos dos motores com variáveis de números reais, verificou-se que o mesmo estava travando devido ao pequeno tamanho da memória, conforme pesquisa na parte teórica e em sites especializados.

Foi alterado o código para tratar a parte fracionária como inteiro apenas para comparações lógicas para a movimentação (só tratados apenas os quatro últimos dígitos de cada coordenada, pois os mesmos dão a precisão em metros para os movimentos de até 20 metros) no *Proteus Isis*. O algoritmo funcionou apenas com o *WatchDog* habilitado (resetando o programa a todo momento), constatando-se que alguma condição do programa estava entrando em erro.

Após inúmeras melhorias na construção do código fonte do microcontrolador, obteve-se sucesso na codificação, tratando as informações de latitude e longitude do GPS, utilizando as partes inteira e fracionária, transformando as mesmas em números inteiros de 32 bits. Vale ressaltar que com as melhorias no código, o uso de memórias (RAM e ROM) ficou menor que 50% da capacidade do PIC16F628A, conforme se pode verificar na figura 30.

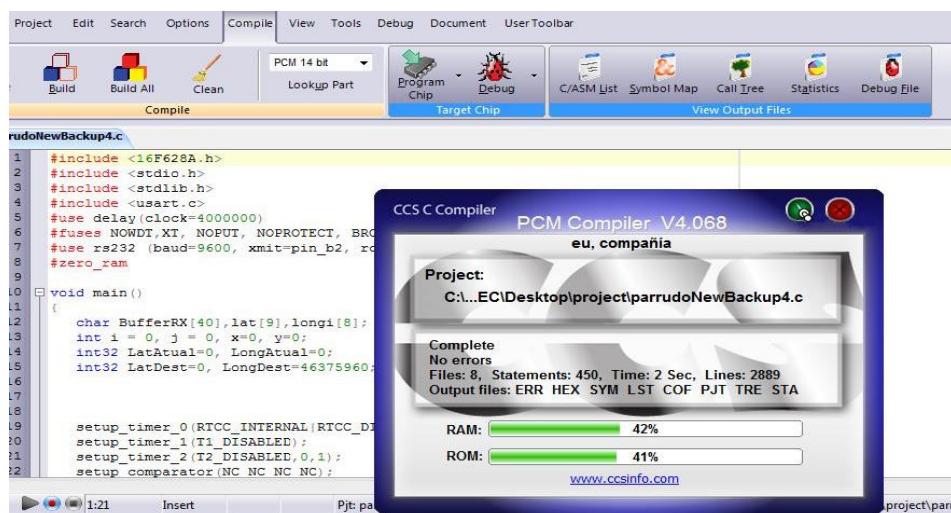


Figura 30: Tela de compilação do código no CCS Compiler

Fonte: Imagem da tela gerada durante os testes

Décimo experimento

Após a finalização do software, foi iniciado o teste no protótipo, gravando o microcontrolador com o teste básico de movimentação que faz o veículo andar para frente por 3 segundos e para trás por mais 3 segundos. Neste teste foi identificado um defeito, pois os motor traseiro de aceleração não iniciava, causando o sobreaquecimento do microcontrolador. Ao executar os testes com o motor frontal (movimentos de esquerda e direita), tudo funcionava normalmente.

Após vários testes, identificou-se que no momento em que o motor traseiro partia, havia um pico de corrente elétrica, e este pico de corrente fazia com que o microcontrolador reiniciasse a programação e nunca partia os motores. O problema foi solucionado adicionando-se diodos para proteção do circuito (evitar corrente reversa no circuito), e resistores com baixo valor de resistência (5 Ohms), para evitar o pico de corrente elétrica na saída da ponte H.

Solucionado o problema, foi decidido então reconstruir a placa do protótipo, integrando o circuito já proposto com um regulador de tensão (para garantir a tensão de 5VCC no circuito) já que utilizou-se duas baterias de 3,6 VCC em série (totalizando 7,2 VCC), os diodos de proteção, um fusível e os resistores nas saídas da ponte H.

Ao interligar o circuito eletrônico com o veículo, deparou-se com um novo problema. Estava tudo funcionando corretamente, porém o veículo não tinha torque para se deslocar no chão. Decidiu-se então colocar a tensão maior (7,2VCC) no pino de alimentação dos motores na ponte H. Com isso, o veículo ficou com mais torque nos motores, e foi possível deslocar-se no chão.

Executou-se então os testes do veículo com valores de longitude conhecidos como destino, e a partir do ponto de partida, o veículo deslocou-se até o ponto destino e parou, conforme estava programado no microcontrolador, obtendo-se sucesso no teste final.

3.2.10 Algoritmo do Protótipo:

Em paralelo com teste de hardware, foi elaborada a lógica de ação do veículo:

- No início do programa, é feita a verificação do ponto inicial e do ponto de destino de longitude;
- Efetua pré-teste antes do primeiro movimento;
- Coleta a informação do GPS;
- Filtra e converte os valores de longitude;
- Compara se a longitude atual é igual, maior ou menor que a de destino;
- Efetua um movimento, caso exista diferença (maior ou menor) a ser corrigida;
- Caso a condição seja satisfeita, o veículo entra em modo de repouso.

3.3 TERCEIRA FASE: Finalização (Montagem, Programação Funcionamento)

Após todo o estudo teórico e a gama de testes realizada, foram obtidos os resultados finais da implantação do módulo GPS em um veículo autônomo, gerando um código fonte do programa gravado no microcontrolador, o fluxograma de funcionamento do mesmo, o circuito eletrônico e a montagem do protótipo.

3.3.1 Fluxograma de Funcionamento do Microcontrolador

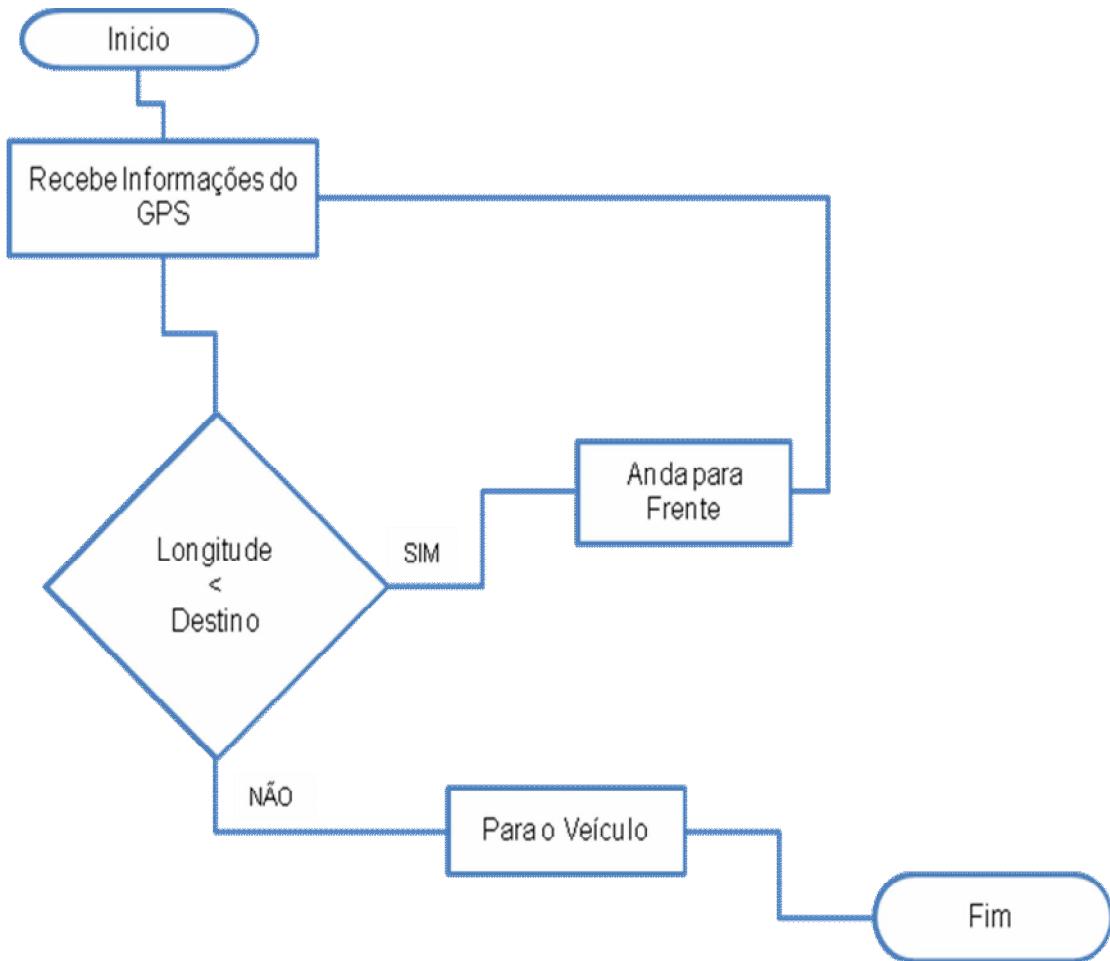


Figura 31: Fluxograma de funcionamento do microcontrolador

3.3.2 Programação do Microcontrolador

Após a definição do comportamento do veículo, foi gerado a programa do microcontrolador, responsável pelo acionamento e desligamento dos motores conforme as informações recebidas do GPS.

```

//*****
//*
/* USART.C
*/
/* Biblioteca de manipulação da USART
/* Suporta o modo assíncrono
*/
/* Autor: Fábio Pereira
/* Livro: Microcontroladores Pic: Programação em C
*/
//*****
  
```

```

char usart_rx, usart_tx, txreg, rcreg, spbrg;

struct rcsta_reg
{
    int rx9d : 1;
    int oerr : 1;
    int ferr : 1;
    int aden : 1;
    int cren : 1;
    int sren : 1;
    int rx9 : 1;
    int spen : 1;
} rcsta;

struct txsta_reg
{
    int tx9d : 1;
    int trmt : 1;
    int brgh : 1;
    int xxx : 1;
    int sync : 1;
    int txen : 1;
    int tx9 : 1;
    int csra : 1;
} txsta;

// define os endereços das variáveis
#define rcsta = 0x18
#define txreg = 0x19
#define rcreg = 0x1a
#define txsta = 0x98
#define spbrg = 0x99
#define r_pirl = 0x0c // define o registrador r_pirl
#define flag_rc = r_pirl.5 // define o flag_rc
// definições utilizadas nas funções

void usart_inicializa ( int vel, boolean brgh )
/*
O valor dos parâmetros vel e brgh deve ser retirado a partir das
tabelas de baud rate fornecidas pela Microchip ou no livro:
Microcontroladores PIC: Técnicas Avançadas.
*/
{
    txsta.brgh = brgh; // seleciona o modo do gerador de baud rate
    spbrg = vel; // configura o gerador de baud rate

    // configura os pinos da USART como entradas !!!!!
    input (pin_b1); //RX
    input (pin_b2); //TX

    txsta.sync = 0; // seleciona o modo assíncrono
    rcsta.spen = 1; // habilita a USART
    txsta.tx9 = 0; // seleciona o modo de 8 bits
    txsta.txen = 1; // ativa o transmissor da USART
    rcsta.cren = 1; // modo de recepção contínua
}

```

```

void usart_transmite (char dado)
{
    while (!txsta.trmt); // aguarda o buffer de transmissão esvaziar
    txreg = dado; // coloca novo caractere para transmissão
}

char usart_recebe (void)
{
    while (!flag_rc); // aguarda a recepção de caracteres
    return rcreg; // retorna o caractere recebido
}

//*****
//*
// * FiltroGPS.C - Novembro 2009
// *
// * Trata as informações do GPS, convertendo os valores
// * de latitude e longitude, e executa os passos para
// * atingir a longitude de destino (caso necessite)
// * Autores: Danilo Maurilio,
// *           Eduardo de Siqueira Correa,
// *           Jonas Fabricio Kozamekinas
// * 10º Semestre Eng. Computação, Universidade Metodista de SP
// *
//*****
#include <16F628A.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <uart.c>
#use delay(clock=4000000)
#fuses NOWDT,XT, NOPUT, NOPROTECT, BROWNOUT, MCLR, LVP, NOCPD
#use rs232 (baud=9600, xmit=pin_b2, rcv=pin_b1)
#zero_ram

void main()
{
    char BufferRX[40],lat[9],longi[9];
    int i = 0, j = 0, x=0, y=0;
    int32 LatAtual=0, LongAtual=0;
    int32 LatDest=0, LongDest=46345875;

    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    set_tris_a(0x00); //Configura Porta A como saída

    usart_inicializa (25,1);

    while(true)
    {
        LatAtual=0;
        LongAtual=0;
        x=0;
        y=0;

        BufferRX[i]=usart_recebe();
}

```

```

if(BufferRX[i]=='$')
{
    for(i=0;i<5;i++)
    {
        BufferRX[i]=uart_recebe();
    }

if((BufferRX[0]=='G')&&(BufferRX[1]=='P')&&(BufferRX[2]=='G')&&(BufferRX[3]
=='G')&&(BufferRX[4]=='A'))
{
    for(j=0;j<34;j++)
    {
        BufferRX[j]=uart_recebe();

        if((j>11)&&(j<16))
        {
            lat[x] = BufferRX[j];
            usart_transmite(lat[x]);
            x++;
        }

        if((j>16)&&(j<21))
        {
            lat[x] = BufferRX[j];
            usart_transmite(lat[x]);
            x++;
        }

        if((j>24)&&(j<29))
        {
            longi[y] = BufferRX[j];
            usart_transmite(longi[y]);
            y++;
        }

        if((j>29)&&(j<34))
        {
            longi[y] = BufferRX[j];
            usart_transmite(longi[y]);
            y++;
        }
    }

    LongAtual=atoi32(longi); //conversão de string para inteiro de
32 bits

    LatAtual=atoi32(lat); //conversão de string para inteiro de 32
bits

    if (LongAtual < LongDest) //se for menor, o veículo se desloca
    {
        output_high(PIN_A2);
        output_low(PIN_A3);
    }

    else //caso contrário, para
    {
        output_low(PIN_A2);
        output_low(PIN_A3);
    }
}

```

} } }

3.3.3 Sistema Eletrônico do Protótipo

O circuito eletrônico do veículo possui duas baterias recarregáveis de 3,6VCC (uma com corrente de 600mAh, e outra com corrente de 500mAh) ligadas em série, totalizando 7,2VCC. Esta tensão vai para o pino de alimentação dos motores do circuito integrado L293D (ponte H), e para um regulador de tensão (LM7805), que regula a tensão para 5VCC, e sua saída alimenta a parte lógica do circuito.

O microcontrolador utiliza um cristal de 4MHz, fazendo com que seu clock interno seja de 1MHz. Foi utilizado um resistor de 10KΩ no pino de *Master Clear*, que reseta o microcontrolador na energização, garantindo que o mesmo inicie sempre da primeira linha de programação. O pino 7 (RX) foi interligado ao pino de TX do GPS. O pino 8 (TX) do microcontrolador foi ligado ao circuito integrado MAX232N, que é responsável por converter o sinal do microcontrolador para ser interpretado pelo PC. Os pinos 1,2,17 e 18 são utilizados como saídas digitais, para o acionamento das entradas do circuito integrado L293D.

O circuito L293D foi ligado conforme o manual do fabricante, adicionando-se os diodos de proteção e os resistores de 5Ω , para evitar os picos de corrente na partida dos motores.

O circuito integrado MAX232N foi ligado conforme o manual do fabricante.

A figura 32 mostra o esquema proposto para o funcionamento do veículo.

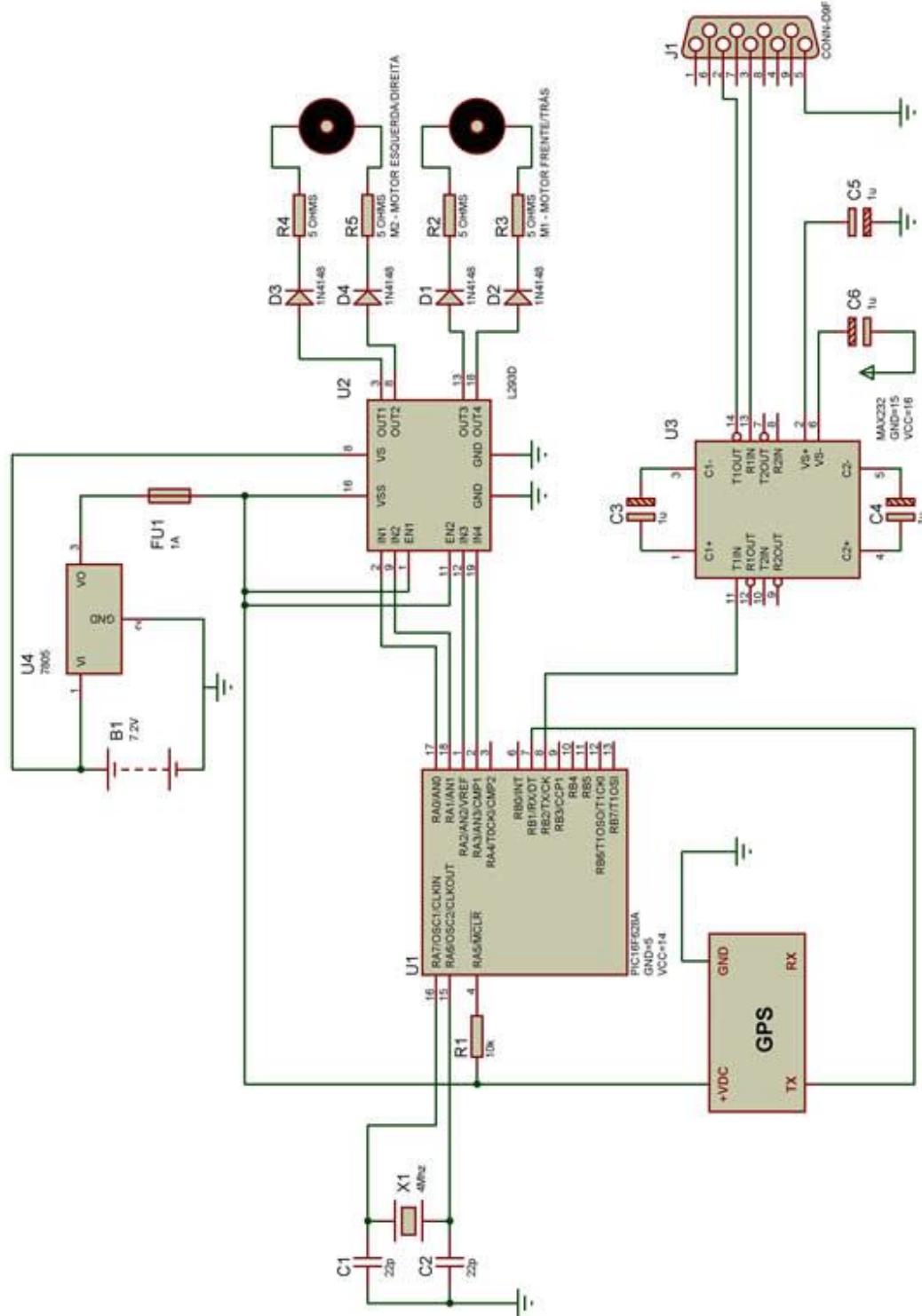


Figura 32: Esquema eletrônico do circuito do veículo feito no *Proteus Isis*

3.3.4 Protótipo

O circuito da figura anterior foi montado na placa universal de fenolite (figura 33), interligou-se os fios do GPS nos respectivos bornes (parte superior da figura), e os fios das baterias nos bornes do outro lado da placa (parte inferior da figura).

Segue a descrição dos itens numerados:

- 1 – Módulo receptor GPS;
- 2 – Microcontrolador PIC16F628A;
- 3 – Circuito integrado L293D;
- 4 – Circuito integrado MAX232N.

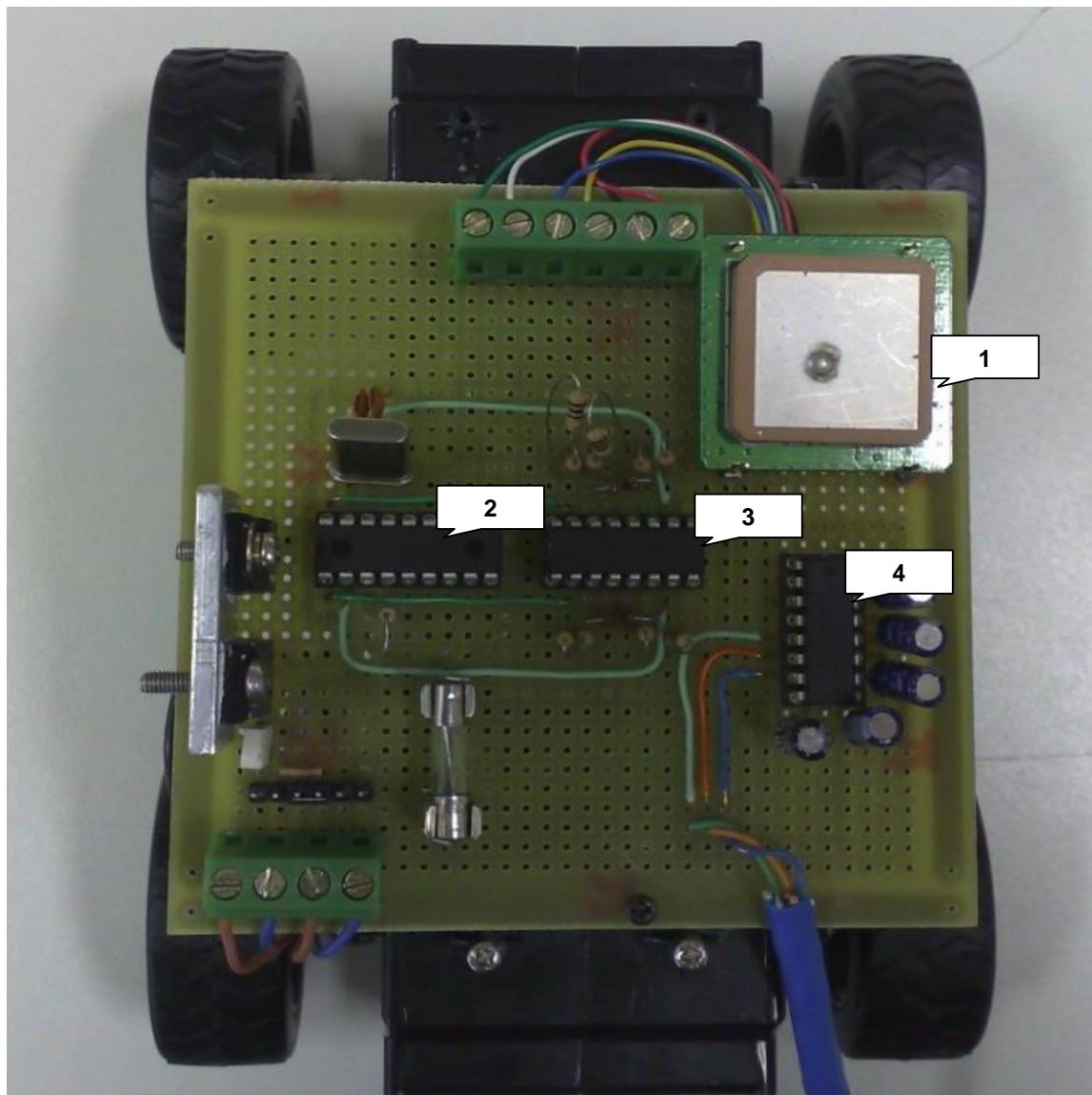


Figura 33: Esquema eletrônico do circuito montado na placa universal de fenolite

4 CONCLUSÃO

O projeto de correção automática de rotas de veículos por GPS mostrou-se eficaz, devido aos resultados obtidos com sucesso nas buscas pelas informações, e aos equipamentos necessários para o desenvolvimento de um protótipo que utilizam conceitos de eletrônica e lógica de programação.

Durante a construção do protótipo foram encontradas muitas dificuldades, como a falta de conhecimento na tecnologia escolhida (GPS) e no funcionamento prático dos microcontroladores, também houve contratemplos na aquisição dos componentes e erros na programação no microcontrolador. Por isso, alguns itens da proposta inicial tiveram que ser postergados, outros adaptados, no intuito de focalizar as atividades com a parte principal do projeto.

A cada evolução, deparou-se com a necessidade de buscar mais conhecimento técnico e novas metodologias para a concretização do projeto, o que ocasionou maior investimento de tempo do que o previsto no cronograma proposto.

Apesar de todos os problemas técnicos encontrados ao longo do desenvolvimento, é importante ressaltar que a experiência de aplicar na prática o que foi aprendido na universidade é um ato muito prazeroso, empolgante e ao mesmo tempo importante para o futuro profissional. Isto se tornou um motivador para que este trabalho mostrasse o emprego de diversos conceitos adquiridos ao longo do curso de Engenharia da Computação na Universidade Metodista de São Paulo.

Utilizou-se a ferramenta *Proteus Isis* no ambiente de testes virtuais do projeto. Filtrou-se as mensagens fornecidas pelo módulo GPS e, através de condições matemáticas e lógica, construiu-se o software embarcado que foi transferido ao microcontrolador. O software desenvolvido funcionou como esperado acionando o motor de partida até o encontro do seu destino.

Por fim, este TCC demonstrou ser um projeto com a necessidade de um tempo maior do que o proposto inicialmente por seus integrantes, para que pudessem ser implementadas todas a propostas iniciais. Porém, a satisfação de atingir o objetivo principal recompensa todo o esforço, as noites sem dormir e fins de semanas de dedicação, para a concretização deste projeto, tornando o mesmo uma fonte de

consulta para muitas outras pesquisas e desenvolvimentos com utilização de GPS, e controle automático de rotas.

4.1 SUGESTÕES PARA TRABALHOS FUTUROS

Diante do cenário existente e exposto durante o trabalho de conclusão de curso, a seguir apresentam-se as sugestões para trabalhos futuros:

- A implementação de sensores (ópticos, capacitivos, ultrassônicos) para evitar colisões com obstáculos móveis ou fixos;
- Desenvolvimento de interface gráfica para inserir as coordenadas de latitude e longitude por comunicação serial;
- Desenvolvimento de monitoramento à distância com interfaces gráficas (mapas cartográficos existentes) usando as tecnologias GSM e/ou 3G, para o rastreamento e intervenções (mudança de rota) do veículo autônomo;
- Implementação de um *DATALOGGER* (dispositivo eletrônico que armazena dados ao longo do tempo, utilizando-se de sensores ou instrumentos de medição), utilizando a mesma arquitetura e implementando um microcontrolador mais potente e com mais memória ROM.

5 REFERÊNCIAS BIBLIOGRÁFICAS

APLICAÇÃO DO SISTEMA DE POSICIONAMENTO GLOBAL (GPS) NA COLETA DE DADOS. Sistema de Posicionamento Global (GPS). Técnicas de posicionamento do GPS: fundamentos e descrições.

Disponível em:

<<http://www.rc.unesp.br/igce/aplicada/DIDATICOS/LANDIM/textogps.pdf>>

Acesso em 20/03/2009

BENNET, P. - NMEA-0183 and GPS Information

Disponível em:

<<http://vancouver-webpages.com/peter/nmeafaq.txt>>

Acesso em 27/04/2009

BRESSIANI, D., PERINI, I., SILVA, J. N., GANZER, J. Sensor de medição de distância – Universidade de Caxias do Sul (UCS)

Disponível em:

<<http://www.ucs.br/ccet/demc/vjbrusam/inst/dist61.pdf>>

Acesso em 20/04/2009

DAVID, José de Souza. Desbravando o PIC: Ampliado e Atualizado para PIC16F628A. São Paulo: Editora Érica, 2008. 12ª Edição.

DILÃO, R. – Sistema de Posicionamento Global.

Disponível em:

<<http://www.cienciaviva.pt/latlong/anterior/gps.asp>>

Acesso em 25/03/2009

FUNDAMENTOS GPS. SISTEMA DE POSICIONAMENTO GLOBAL. Fonte: Curso de GPS e Cartografia Básica – Instituto CEUB de Pesquisa e Desenvolvimento

Disponível em:

<<http://www.ltc.ufes.br/GEOGRAPHIC/Modulo%20GPS.pdf>>

Acesso em 15/03/2009

GIMENES, Salvador Pinillos. Microcontroladores 8051: Teoria do hardware e do software / Aplicações em controle digital / Laboratório e simulação. São Paulo: Pearson Prentice Hall, 2002.

HYPERTERMINAL – Componente do *Windows XP*

Disponível em:

<<http://pt.wikipedia.org/wiki/HyperTerminal>>

Acesso em 08/10/2009

MAX232 – Circuito Integrado MAX232

Disponível em:

<<http://pt.wikipedia.org/wiki/MAX232>>

Acesso em 08/10/2009

ME-1000RW: Manual do ME-1000RW - ver. 1.2

Disponível em:

<<http://res.trilha21.com/001000608/files/ME-1000RW.pdf>>

Acesso em 15/05/2009

PEREIRA, Fábio. Microcontroladores PIC: Programação em C. São Paulo: Editora Érica, 2009. 7ª Edição.

PIC16F628A – Manual do Microcontrolador PIC16F628A

Disponível em:

<<http://ww1.microchip.com/downloads/en/DeviceDoc/40044F.pdf>>

Acesso em 25/03/2009

PIC16F877A – Manual do Microcontrolador PIC16F877A

Disponível em:

<<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>>

Acesso em 23/03/2009

National Marine Electronics Association

Disponível em:

<http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp>

Acesso em 27/04/2009

CARREIRA R., FONSECA, P. Sensores Capacitivos. Sebenta Multimedia – Instituto Superior Técnico de Lisboa

Disponível em:

<http://www.estiq.ipbeja.pt/~lmqt/cee/misc/Sebenta_Online/cap_07/senscapa.htm>

Acesso em 20/04/2009

SENSORES DE PROXIMIDADE ULTRASSÔNICOS

Disponível em:

<<http://www.nivete.com.br/htm/outrsensordproximidadeultrasonico.htm>>

Acesso em 20/04/2009

SOARES, Márcio José. Revista Eletrônica Total nº102. Editora Saber Ltda, 2004.

TTL – Lógica Transistor-Transistor

Disponível em:

http://pt.wikipedia.org/wiki/Transistor-Transistor_Logic

Acesso em 08/10/2009

VOLERE – Método de levantamento de informações

Disponível em:

<<http://www.volere.co.uk>>

Acesso em 01/09/2009

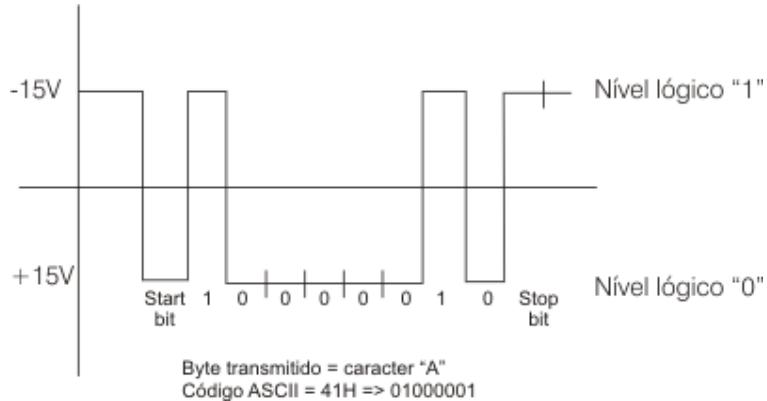
6 GLOSSÁRIO

Circuito Integrado MAX232N: O MAX232 é um circuito eletrônico que converte sinais de uma porta serial para sinais adequados para uso em circuitos microprocessados, por exemplo. O MAX232 amplifica/reduz sinais RX, TX, CTS e RTS. A discrepância de voltagem (acima de $\pm 12\text{VCC}$ do RS232 para 3.3VCC TTL) é gerada por capacitores (normalmente de 10 nF). O MAX232 é um transmissor/receptor duplo que fornece níveis de voltagem TIA/EIA-232-F de uma única fonte de tensão de 5VCC . Cada receptor converte entradas TIA/EIA-232-F para níveis de 5VCC TTL/CMOS . Estes receptores têm um limiar típico de 1.3VCC , uma histerese típica de 0.5VCC e pode aceitar $\pm 30\text{VCC}$ de entrada. Cada transmissor converte níveis de entrada TTL/CMOS em níveis TIA/EIA-232-F.

Compilador CCS (CCS Compiler): O compilador CCS é uma plataforma de programação em linguagem C da empresa *Custom Computer Services*. Ele incorpora ferramentas de software projetadas para os microcontroladores e dispositivos da empresa Microchip. O compilador CCS possui operadores padrões da linguagem C, além de bibliotecas embutidas que são específicas para acesso a recursos de hardware. A versão do software utilizada neste TCC é a 4.0.69.

Comunicação Serial (RS-232): O padrão RS-232 é uma dos mais difundidos no mundo da automação e controle. Hoje em dia muitos equipamentos fazem uso do mesmo. O tipo de transmissão é serial, ou seja, transmitimos os bytes através de uma única via bit a bit e não byte a byte como em uma porta paralela, por exemplo.

Os sinais elétricos na porta serial (RS-232) possuem níveis de tensão entre $+15\text{VCC}$ e -15VCC . Na figura abaixo, temos a representação de um sinal RS-232 na transmissão de um byte. Na RS-232, o “1” lógico tem nível de tensão entre -15VCC à -3VCC e o “0” lógico tem nível entre $+3\text{VCC}$ à $+15\text{VCC}$.



Toda comunicação assíncrona depende de alguns fatores, tais como: velocidade, número de bits, paridade e stop bits.

A velocidade de comunicação é medida em bps (bits por segundo). Ela também é fundamental para o sucesso na comunicação entre equipamentos. Ambos os equipamentos (transmissor e receptor) devem ser configurados para operar na mesma velocidade, sempre. Esta velocidade pode ser configurada de 300 bps a 115000 bps nos dias atuais, dependendo exclusivamente do equipamento.

O número de bits define quantos bits compõe o byte a ser enviado. Nos dias de hoje, para a maioria dos equipamentos ele tem oito bits. Porém é possível a comunicação de 5 a 8 bits.

A paridade é o modo mais antigo e simples de se verificar um erro durante a transmissão. Ela considera o número de bits em estado lógico igual a "1". Assim a unidade transmissora "conta" o número de bits do byte transmitido em estado lógico "1" e envia um bit informando isso. Se o número de bits for par, o bit de paridade assume o estado lógico "1". Se ele for ímpar seu estado lógico será "0".

Existem três formas de controle de paridade: par, ímpar e nenhuma. A diferença está em indicar se o número de bits no estado lógico "1" é sempre par, ímpar ou nenhum. Nesse caso o bit de paridade assume estado lógico "1" de acordo com a opção selecionada. Se a escolha for nenhum, o bit de paridade é ignorado pelo equipamento receptor (Obs.: Apesar do bit de paridade poder informar um erro, não existe fator de correção. Neste caso seria necessário via software verificar a paridade e se detectado um erro, promover um novo envio do byte danificado.).

O Stop bit é uma forma de informar o agente receptor sobre o final do byte. Ele pode assumir "1" ou "2" bits de parada.

O Start bit serve para informar sobre o inicio de uma transmissão.

Sendo assim a configuração de uma porta RS-232 exige as seguintes informações:

- velocidade;
- número de bits;
- paridade;
- número de stop bits.

A configuração para velocidade de comunicação, número de bits, paridade e número de stop bits deve ser a mesma nos equipamentos transmissores e receptores, caso contrário a comunicação não será estabelecida corretamente. (SOARES, 2004)

HyperTerminal: O HyperTerminal é o aplicativo de comunicações mais antigo do sistema operacional *Windows* (Microsoft). Lançado em 1985, o *HyperTerminal*, que até 1995 se chamava Terminal, permitia conectar o computador para uma outra pessoa.

Em 1995, com o lançamento do *Windows 95*, o componente passa a se chamar *HyperTerminal*, em parceria entre a Hilgraeve e a Microsoft. Lançou novos recursos, como a criação de um servidor, entre outros.

Em 2007, com o lançamento do *Windows Vista*, o *HyperTerminal* não é mais incluído como parte dos Acessórios do Windows, agora o *HyperTerminal* está disponível apenas no site oficial da Hilgraeve (<http://www.hilgraeve.com>)

Instrução: Define uma única ação que o microcomputador pode executar por vez. As ações das instruções podem corresponder a operações de leitura e escrita nos conteúdos dos registradores ou nas posições de memória, a operações lógicas e aritméticas. (GIMENEZ, 2005)

Memória de Armazenamento de Informações: Essa memória deve permitir a escrita e leitura de informações. O seu tipo é volátil, ou seja, não grava de modo permanente os dados nela contidos. Quando a alimentação do sistema é cortada, tudo que foi guardado é perdido. As memórias que apresentam essas características são memórias RAM (*Random Access Memory*). O sistema é bastante útil para o processamento de dados, pois disponibiliza espaço para informações cruciais, que

podem ser acessadas de maneira quase imediata, ao contrário de outras formas de armazenamento, como discos rígidos, CDs ou DVDs. (GIMENEZ, 2005)

Memória de Armazenamento de Programa: todo microcomputador deve ter uma área de memória na qual será armazenado o programa que definirá as tarefas que o microprocessador deve executar. Esse programa define a função de um microcomputador e também ao microprocessador, instrução por instrução, o que ele deve fazer ao longo do tempo. É necessário a implementação de um programa para que o microcomputador funcione, caso contrário ele não irá funcionar. Esse programa não pode ser perdido na ausência de energia elétrica, pois isso acarretaria a perda da funcionalidade desse microcomputador, para o qual foi projetado. Assim, o tipo de memória em que esse programa deve ser armazenado é a memória não-volátil. Dessa maneira, sempre que o microcomputador for ‘desenergizado’, esse programa não poderá ser perdido e quando ele for energizado novamente, deverá voltar a funcionar com a mesma característica funcional. Memórias não-voláteis normalmente utilizadas em microcomputadores são as memórias ROM (*Read Only Memory*), PROM/OTP (*Programmable Read Only Memory / One Time Programming*), EPROM (*Erasable Programmable Read Only Memory*) e EEPROM (*Electrical, Erasable and Programmable, Read Only Memory*). Geralmente, essas memórias são utilizadas somente para leitura e consequentemente não é possível realizar operações de escrita. (GIMENEZ, 2005)

Microcomputador: equipamento compacto cujo objetivo principal é simular (imitar) eletronicamente o sistema chamado ser humano. Suas principais funções são grande velocidade de processamento das informações e alta confiabilidade nas atividades executadas. Ele é implementado por três blocos básicos: Unidade Central de Processamento, Unidade de Memória e Unidade de Entrada e Saída de Informações. (GIMENEZ, 2005)

Microprocessador ou Unidade Central de Processamento (CPU): mais conhecido como CPU (*Central Processing Unit*). Fazendo analogia com sistema ser humano, o microprocessador de um microcomputador corresponde ao cérebro de um ser humano. Fisicamente é um dispositivo semicondutor (círcuito integrado) constituído por milhões de transistores que implementam uma variedade de circuitos

(registradores, máquinas seqüenciais, circuitos lógicos). Ele é responsável pela busca de programa na memória e por sua execução. (GIMENEZ, 2005)

MultiSIM: é um software de uso profissional da empresa National Instruments, para o desenvolvimento de placas de circuito impresso (PCI) com ferramentas globais para captura esquemática, simulações interativas, moldura de placas e teste integrados.

Proteus Isis VSM: O software Proteus Isis VSM (Virtual System Modelling, ou Sistema de Modelagem Virtual) integra simulação de circuitos em modo SPICE (*Simulated Program with Integrated Circuits Emphasis*, ou Programa de Simulação com Ênfase em Circuitos Integrados), animação de componentes e modelos completos de microcontroladores para facilitar a co-simulação. É um software muito útil para desenvolver e testar protótipos antes de sua construção física. É fabricado pela empresa Labcenter Electronics. A versão do software utilizada neste TCC é a 7.3 SP3.

Protoboard: é uma placa com milhares de furos e conexões condutoras para montagem de circuitos elétricos experimentais. A grande vantagem do protoboard na montagem de circuitos eletrônicos é a facilidade de inserção de componentes, uma vez que não necessita soldagem. As placas variam de 800 furos até 6000 furos, tendo conexões verticais e horizontais. Na superfície de uma matriz de contato há uma base de plástico em que existem centenas de orifícios onde são encaixados os componentes. Em sua parte inferior são instalados contatos metálicos que interligam eletricamente os componentes inseridos na placa.

TTL: A Lógica Transistor-Transistor (Transistor-Transistor Logic) é uma classe de circuitos digitais construídos de transistores de junção bipolar (BJT), e resistores. É chamado lógica transistor-transistor porque ocorrem ambas as funções porta lógica e de amplificação pelos transistores. Isso é notável por ser uma família difundida de circuitos integrados (CI) usada por muitas aplicações como computadores, controle industrial, eletrônica de consumo, sintetizadores etc. Por causa do grande uso desta família lógica, sinais de entrada e saída de equipamentos eletrônicos, pode ser

chamada entrada ou saída "TTL", significantemente compatível com os níveis de voltagem usados.

Os circuitos digitais TTL têm como principal característica a utilização de sinais de 5 volts para níveis lógico altos. Seus circuitos integrados são constituídos basicamente de transistores, o que os torna pouco sensíveis à eletricidade estática.

Unidade de Entrada e Saída (E/S): também chamada pela abreviação de I/O (*Input / Output*), é responsável pelo interfaceamento das informações entre o microcomputador e o mundo externo. Geralmente são circuitos integrados capazes de ler e armazenar as informações vindas do mundo externo, por exemplo, informações do teclado, do canal de comunicação serial de um microcomputador, e que também pode definir informações para o mundo externo como alto-falantes, *leds* (diodo emissor de luz) indicadores, interfaces de comunicação seriais e paralelas. (GIMENEZ, 2005)

Unidades de Memória: são divididas em duas partes: memória de armazenamento de programas e memória de armazenamento de informações (dados, *bytes*, *bits*). (GIMENEZ, 2005)

USART: O Transmissor/ Transmissor/Receptor Universal Síncrono e Assíncrono (UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER) é um formato padrão de comunicação de dados de forma serial. A maioria dos microcontroladores disponíveis no mercado possui uma USART. A USART do microcontrolador PIC16F628A possui capacidade de comunicação RS-232 no modo assíncrono com velocidade de até 115000 bps (bits por segundo), dependendo exclusivamente do cristal utilizado no clock de operação do microcontrolador. Externamente, esta USART se comunica através de dois pinos, TX e RX (RB2 e RB1 respectivamente, no PIC16F628A). Por padrão, não existem outros pinos RS-232 (CTS, RTS, CD, etc) e caso exista a necessidade de seu uso, os mesmos poderão ser implementados com os outros pinos de I/O, presentes no microcontrolador. Um outro ponto importante sobre os pinos do PIC é que quando se utiliza a USART interna, os pinos de TX e RX não podem mais ser utilizados como pinos de I/O comuns (digitais). (SOARES, 2004)

7 APÊNDICE

APÊNDICE A - Relatório de Recomendações de Qualificação da Banca.

8 ANEXOS

ANEXO A – MANUAL DO CI SM6135W

V-chip voltage regulator function remote control cars SM6135W

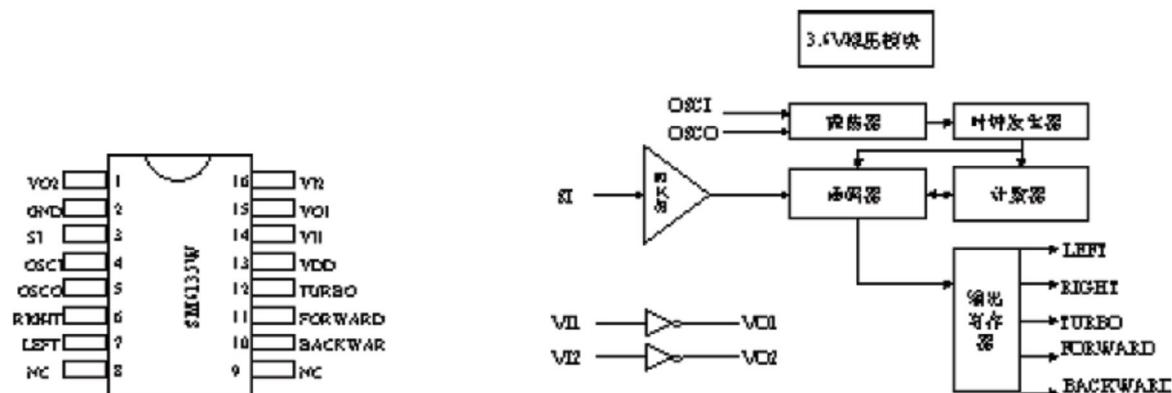
Features

Wide voltage range: 2.4V ~ 12.0V
 Superior drive capability
 Very few external components required

Overview

SM6135W remote control car is designed for large-scale integrated circuit design. There are five control keys to control the remote control car campaign, they are: forward, backward, right, left and accelerate the function of this 5. Chip self - 3.6V voltage regulator modules.

Chip pin and the internal block diagram are as follows:



Pin	Name	Functional explanation
1	VO2	Gate 2 output pin
2	GND	0 volt
3	SI	Coded signal input pin
4	OSCI	Oscillation input pin
5	OSCO	Oscillation output pin
6	RIGHT	Output pin turn right
7	LEFT	Output pin turn left
8	NC	-
9	NC	-
10	BACKWARD	Output pin backward
11	FORWARD	Output pin forward
12	TURBO	Output pin accelerate
13	VDD	Power supply
14	VI1	Gate 1 input pin
15	VO1	Gate 1 output pin
16	VI2	Input NAND Gate 2 input pin

Electrical performance indicators

SM6135W (VDD=4.0V, FOSC=128KHZ, T=25°C)

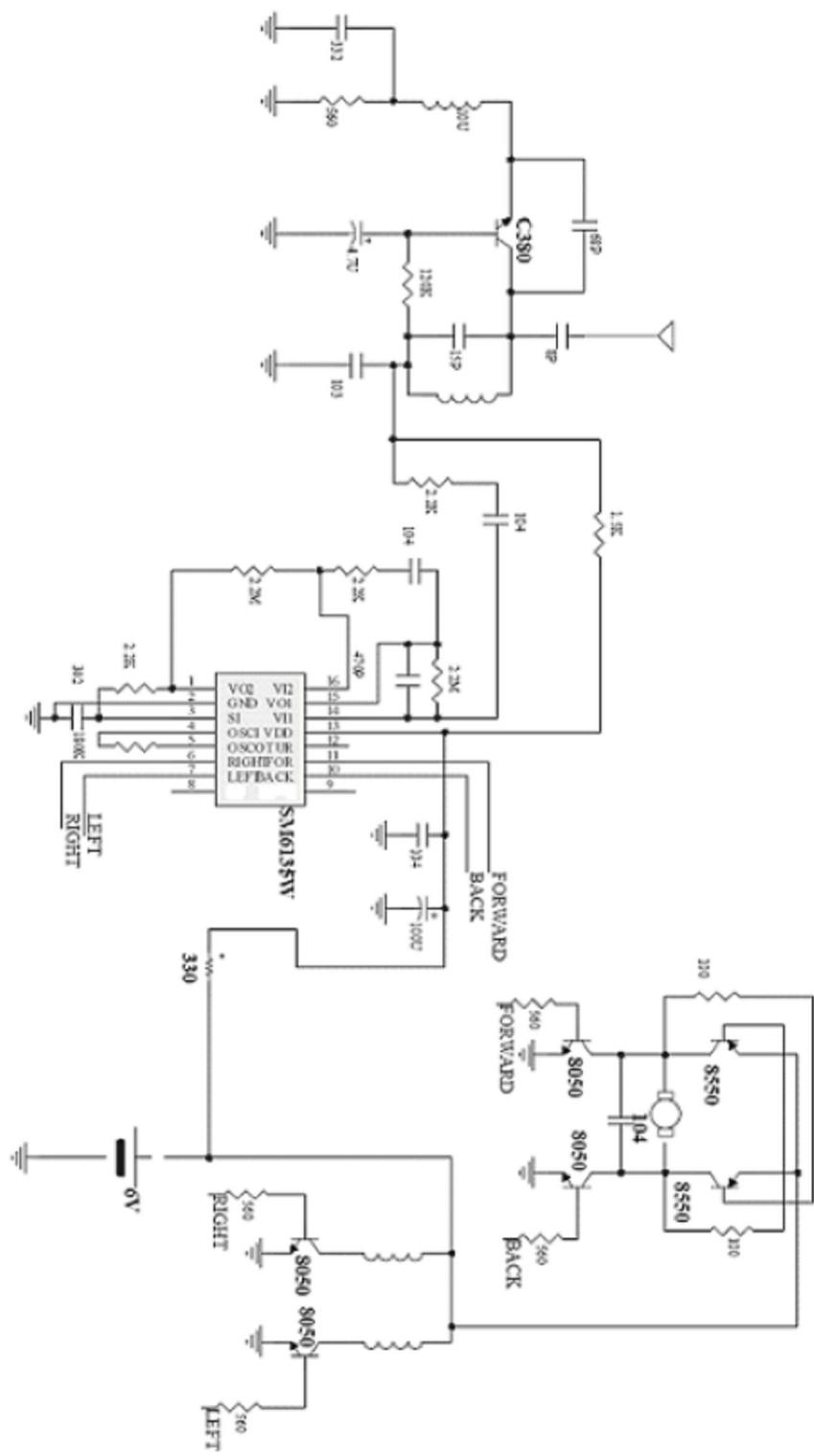
Item	Symbol	Minimum	Typical	Max
Operating voltage	VDD	2.4V	9V	12V
Current	IDD		1mA	
Output drive current	IDR		10mA	
Frequency deviation of the scope permitted	FT	-20%		+20%
Voltage regulator	FW	3.6V		4.2V

PAD Maps and coordinates

Pin	X	Y
VDD	630.10	822.70
VI1	468.80	822.65
V01	265.75	822.65
VI2	82.50	797.50
V02	82.30	656.80
GND	82.40	516.00
SI	365.85	72.85
OSCI	529.90	72.25
OSCO	697.30	73.10
RIGHT	867.55	114.25
LEFT	866.20	335.20
BACKWARD	868.95	491.50
FORWARD	868.50	642.65
TURBO	867.60	823.40

The pad map diagram shows the layout of pins on the SM6135W package. The central label 'SM6135W' is surrounded by various pins. On the left side, there are three vertical columns of pads: 'V12' at the top, 'V02' in the middle, and 'GND' at the bottom. To the right of these are four horizontal rows of pads: 'V01', 'VI1', 'VDD', and 'TURBO' in the top row; 'FORWARD' and 'BACKWARD' in the second row; 'LEFT' and 'RIGHT' in the third row; and 'SI', 'OSCI', 'OSCO', and 'RIGHT' in the bottom row.

Application Chart:



APÊNDICE A – RELATÓRIO DE RECOMENDAÇÕES DE QUALIFICAÇÃO DA BANCA

Recomendações:

- Definir se a pesquisa é somente teórica ou teórica e empírica. Já que os testes são empíricos, pois muitas vezes o ajuste só é alcançado com resultados de formulação de testes práticos, e não com teoria ou matematicamente.

FEITA PELO PROFESSOR: Luiz D'Amore

PARECER DO ORIENTADOR

(X) DE ACORDO

() NÃO VIÁVEL

COMENTÁRIO DO ORIENTADOR: *Entendo que o projeto é teórico (com pesquisas e formulações de hipóteses) e prático (com o desenvolvimento de um protótipo).*

- Ficar atento às tentativas que não funcionar, documentar todo o processo passo a passo, documentar todas as decisões de projeto.

FEITA PELO PROFESSOR: Marcelo Módolo

PARECER DO ORIENTADOR

(X) DE ACORDO

() NÃO VIÁVEL

COMENTÁRIO DO ORIENTADOR: *Isto é parte das atividades prática e experimental e deve ser feito.*

- Melhorar o objetivo, detalhar melhor para que haja um melhor entendimento, enriquecer a descrição do objetivo.
- O objetivo não deve ser somente desenvolver o protótipo, mas fazer um estudo de caso, detalhando os objetos de estudo que são integrados neste projeto.

FEITA PELO PROFESSOR: Marcelo Módolo

PARECER DO ORIENTADOR

(X) DE ACORDO

() NÃO VIÁVEL

COMENTÁRIO DO ORIENTADOR: **O objetivo é o ponto chave, pois é nele que vocês vão se comprometer com o leitor.**

- Documentar tudo que não der certo, para pessoas que quiserem dar continuidade ao projeto ou aplicá-lo em outro projeto não cometam os mesmos erros, ou que possam questionar os métodos utilizados ou que possam vir à substituí-los.

FEITA PELO PROFESSOR: D'Amore

PARECER DO ORIENTADOR

(X) DE ACORDO

() NÃO VIÁVEL

COMENTÁRIO DO ORIENTADOR: **Isto é parte das atividades prática e experimental e deve ser feito.**

- Não é escopo do projeto utilizar inteligência artificial. Há um estudo de implementar sensores para evitar colisão, mas serão adotados padrões, não um sistema com inteligência artificial.

FEITA PELO PROFESSOR: Marcelo Módolo

PARECER DO ORIENTADOR

(X) DE ACORDO

() NÃO VIÁVEL

COMENTÁRIO DO ORIENTADOR: *Não tínhamos pensado em utilizar IA.*

Realmente IA é aplicado nestes casos de controle de rota.