

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA

ANGELO ABDALLAH PELISSON

**INSTRUMENTAÇÃO BÁSICA E SISTEMA DE TELEMETRIA
PARA BAJA**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2018

ANGELO ABDALLAH PELISSON

**INSTRUMENTAÇÃO BÁSICA E SISTEMA DE TELEMETRIA
PARA BAJA**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina Trabalho de Conclusão de Curso 2, do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná - UTFPR, Campus Cornélio Procópio, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Luís Fernando Caparroz Duarte

CORNÉLIO PROCÓPIO
2018



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento de Engenharia Elétrica
Curso de Engenharia Elétrica



FOLHA DE APROVAÇÃO

Angelo Abdallah Pelisson

Instrumentação básica e sistema de telemetria para baja

Trabalho de conclusão de curso apresentado às 15:50hs do dia 13/06/2018 como requisito parcial para a obtenção do título de Engenheiro Eletricista no programa de Graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

Prof(a). Dr(a). Luis Fernando Caparroz Duarte - Presidente (Orientador)

Prof(a). Dr(a). Luiz Francisco Sanches Buzachero - (Membro)

Prof(a). Dr(a). Pedro Carvalhaes Dias - (Membro)

Prof(a). Dr(a). Anderson Wedderhoff Spengler - (Membro)

A folha de aprovação assinada encontra-se na coordenação do curso.

Dedico esse trabalho à minha família.

AGRADECIMENTOS

Agradeço aos meus pais e irmãos, por terem acreditado em mim, pelo apoio e incentivo nos momentos difíceis dessa caminhada, e por sempre estarem do meu lado quando precisei.

Ao Professor Luís, pela excelente didática, mesmo à distância sempre sanou minhas dúvidas, procurando ensinar ao invés de fazer, sendo literalmente um orientador.

À todos os amigos que fiz durante o tempo de graduação, dentre eles Marcelo, Félix, Carnero, Benicio, Bruno, Laercio, Luiz, Igol, Buneco, Dionizio, Fabiano, Camilo, Alyson, Giovani e Bolacha. Principalmente ao Luiz Gustavo Roncon, que sempre esteve comigo desde a primeira aula do curso e que agora não está mais conosco, levarei amizade e parceria de todos pro resto da vida.

À UTFPR por proporcionar um ensino de boa qualidade, fazendo-me tornar um profissional preparado para o mercado de trabalho.

RESUMO

PELISSON, Angelo Abdallah. Instrumentação básica e sistema de telemetria para Baja. 2018. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

O presente trabalho apresenta o desenvolvimento de uma instrumentação básica e sistema de telemetria para baja que é um veículo off- road, com o objetivo de implementar uma comunicação de dados via rádio frequência entre o carro e um computador. É um projeto que visa melhorar o desempenho da Equipe ProcoBaja em competições, aprimorando a eletrônica já existente no veículo.

Palavras-chave: Telemetria, Rádio Frequência, Baja.

ABSTRACT

PELISSON, Angelo Abdallah. Basic instrumentation and telemetry system for Baja. 2018. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

This work presents the development of a basic instrumentation and telemetry system for Baja, which is an off-road vehicle, in order to implement radio frequency data communication between the car and a computer and write this data to a flash memory. It is a project that aims to improve the performance of ProcoBaja Team in competitions, improving the existing electronics in the vehicle.

Keywords: Telemetry, Radio Frequency, Baja.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Diagrama de blocos do sistema | 17 |
| Figura 2 - Configuração SPI..... | 19 |
| Figura 3 – Diagrama de blocos de comunicação | 20 |
| Figura 4 - Conexão SPI..... | 21 |
| Figura 5 - Sensor de temperatura LM35 | 22 |
| Figura 6 - Pinagem do LM35..... | 23 |
| Figura 7 - Sensor de relutância magnética variável MTE-THOMSON | 24 |
| Figura 8 - Sinal gerado pelo sensor de rotação..... | 25 |
| Figura 9 - PIC18F4520..... | 26 |
| Figura 10 - PIC18F2550..... | 27 |
| Figura 11 - Cartão de memória SD..... | 29 |
| Figura 12 - Módulo adaptador cartão MicroSD..... | 30 |
| Figura 13 - Circuito divisor de tensão..... | 30 |
| Figura 14 - Pinagem MX-FS-03V | 31 |
| Figura 15 - Pinagem MX-05V | 32 |
| Figura 16 - Módulo Conversor USB TTL CP2102 Serial RS232 | 33 |
| Figura 17- Simulação LM35 | 36 |
| Figura 18 -Simulação velocidade e RPM | 37 |
| Figura 19 - Gravação cartão SD..... | 38 |
| Figura 20 - Transmissão de dados..... | 39 |
| Figura 21 - Sinal gerado pelo sensor de rotação..... | 40 |
| Figura 22 - Circuito Tx..... | 41 |
| Figura 23 - Montagem sensor e disco | 42 |
| Figura 24 - Tx e furadeira ligados..... | 43 |
| Figura 25 - Circuito Rx ligado no computador | 43 |
| Figura 26 - Resultados | 44 |
| Figura 27 - PCI transmissão..... | 45 |
| Figura 28 - PCI recepção | 46 |
| Figura 29 - PCI transmissão Tx..... | 47 |
| Figura 30 - PCI Recepção Rx..... | 47 |

LISTA DE SIGLAS

PIC – Microcontrolador PIC18F4520.

PCI – Placa de circuito impresso.

RF – Rádio frequência.

A/D – Analógico/Digital.

ADC – Conversão analógica/digital.

D/A – Digital/Analógico.

E/S – Entrada/Saída.

AT – Atenção.

UART – Comunicação serial assíncrona.

LM35 – Sensor de temperatura.

SUMARIO

| | |
|--|----|
| 1 INTRODUÇÃO | 12 |
| 2 PROBLEMA E OBJETIVOS | 14 |
| 2.1 Problemas..... | 14 |
| 2.2 Objetivos..... | 14 |
| 2.2.1 Objetivos gerais..... | 14 |
| 3 MATERIAIS E MÉTODOS | 16 |
| 3.1 MikroC PRO for PIC | 17 |
| 3.1.1 MultiMediaCard Libray | 18 |
| 3.1.2 SPI Libray | 18 |
| 3.2 Comunicação do PIC18F4520 com cartão SD | 19 |
| 3.2.1 Comunicação SPI..... | 20 |
| 3.3 Sensores..... | 22 |
| 3.3.1 LM35 | 22 |
| 3.3.2 Sensor de relutância magnética variável..... | 23 |
| 3.4 Microcontrolador | 25 |
| 3.4.1 PIC18F4520 | 25 |
| 3.4.2 PIC18F2550 | 27 |
| 3.5 Cartões de memória flash..... | 28 |
| 3.5.1 Cartões de memória SD | 28 |
| 3.5.2 Módulo adaptador cartão MicroSD | 29 |
| 3.6 Módulo Transmissor e Receptor de Radiofrequência em 433 MHz..... | 31 |
| 3.6.1 Transmissor MX-FS-03V..... | 31 |
| 3.6.2 Receptor MX-05V..... | 32 |
| 3.7 Comunicação do PIC18F2550 com PC | 32 |

| | |
|---|----|
| 4 DESENVOLVIMENTO DO PROJETO | 34 |
| 4.1 Aquisições dos dados..... | 34 |
| 4.1.1 Temperatura..... | 34 |
| 4.1.2 Velocidade e RPM..... | 36 |
| 4.2 Gravação cartão SD | 37 |
| 4.3 Transmissões de dados..... | 38 |
| 5 RESULTADOS | 40 |
| 5.1 Testes de bancada | 40 |
| 5.2 Placas de circuito impresso | 45 |
| 6 CONCLUSÃO | 48 |
| 7 REFERÊNCIAS | 50 |
| APÊNDICE A - Código implementado no circuito transmissor | 53 |
| APÊNDICE B – Código implementado no circuito receptor. | 61 |

1 INTRODUÇÃO

O Baja é um projeto realizado pela SAE - Society of Automotive Engineers (Sociedade de Engenheiros da Mobilidade), que oferece aos estudantes de engenharia uma oportunidade de aplicar na prática os conhecimentos adquiridos nas aulas teóricas, com o objetivo de prepará-los para o mercado de trabalho (SAE BRASIL,2015). O envolvimento direto com o desenvolvimento do projeto estimula os estudantes a absorver conhecimentos práticos sobre projeto, construção, montagem, manutenção de máquinas, ergonomia, trabalho em equipe, liderança, administração e marketing.

Esse projeto foi criado em uma universidade nos Estados Unidos, a Universidade da Carolina do Sul, sob a direção do Dr. John F. Stevens, em que a primeira competição ocorreu em 1976. No Brasil, a competição foi realizada pela primeira vez em 1995 e cresce mais a cada ano que passa se tornando uma atividade extracurricular importante para alunos de engenharia.

O Baja SAE é um veículo off-road que leva apenas um piloto, é pequeno, ágil e leve, com uma estrutura do tipo gaiola, formada por tubos. Projetado para utilização em pistas de condições variadas de terreno, especialmente na terra, onde são bastante acidentados. É um veículo robusto, mas não deixa de possuir características de conforto e boa dirigibilidade.

Projetado de acordo com as regras e especificações da SAE, o carro possui limitações quanto ao motor e as dimensões da estrutura. Deve possuir um motor de combustão interna padronizado de 10 HP e ser capaz de transportar uma pessoa (o piloto) com até 1,90m de altura e com peso de até 113,4 kg (SAE BRASIL,2009). O resto do projeto é puro desenvolvimento e trabalho dos alunos de engenharia integrantes da equipe.

Telemetria é um processo de comunicação automatizado, que surgiu no ano de 1845 com o primeiro sistema de transmissão de dados entre Winter Palace e o quartel da armada russa. Em 1874 foi desenvolvida uma rede de sensores atmosféricos em Mont Blanc, França, com transmissão de dados em tempo real para Paris. Mais tarde, em 1906 foram montadas umas séries de estações sísmicas na Rússia, com transmissão de dados para o Pulkovo Observatory (Queirós, 2011).

O sistema consiste na medição e coleção de dados em pontos remotos ou móveis, geralmente em pontos de difícil acesso, onde é inviável estabelecer cabeamento, com os dados são transmitidos para equipamentos de recepção e processamento para o devido monitoramento. É uma tecnologia usada em diversas áreas da ciência, desde a medicina e biologia, até a sua aplicação mais conhecida na indústria automotiva: a Fórmula 1 (F1).

A construção de um sistema de telemetria através do desenvolvimento deste trabalho visa informar o piloto sobre como melhorar o seu desempenho e obter melhores resultados nas diversas provas de uma competição, através da comunicação de dados via rádio frequência em tempo real com um computador e armazenamento desses dados em um cartão portátil de memória flash removível (Micro SD), onde enquanto o carro estiver em movimento ou no meio de uma competição, a sua equipe poderá avaliar o desempenho do carro.

2 PROBLEMA E OBJETIVOS

2.1 Problemas

A eletrônica existente no veículo apenas mostra os dados para o piloto, que, ainda assim, não consegue ler os dados facilmente. O aprimoramento da eletrônica com a implementação de uma comunicação de dados via rádio frequência e a gravação desses dados em uma memória flash pode fornecer à equipe dados coletados durante eventos.

Informações como a velocidade em certos trechos de um percurso, podem proporcionar uma melhora no desempenho do piloto, que será informado pela equipe das mudanças que deve fazer durante a volta para ganhar mais tempo.

2.2 Objetivos

2.2.1 Objetivos gerais

Realizar a construção de um sistema de instrumentação para um carro Baja SAE e implementar uma comunicação de dados via rádio frequência entre o carro e um computador, para enviar os dados de temperatura do óleo, RPM do motor e velocidade do carro.

2.2.2 Objetivos específicos

- Identificar as medidas que serão realizadas.
- Definir os sensores que serão utilizados para realizar a aquisição dos dados.
- Desenvolver os circuitos de condicionamento para os sinais gerados pelos sensores de velocidade.
- Interpretar os sinais gerados com um microcontrolador.
- Gravar os dados lidos pelo microcontrolador em uma memória flash.

- Enviar através da comunicação por rádio frequência os dados colhidos para outro microcontrolador.
- Passar os dados recebidos pelo microcontrolador para um computador através da comunicação serial assíncrona.
- Exibir os dados coletados na tela do computador.

3 MATERIAIS E MÉTODOS

Neste trabalho foi realizado primeiramente a identificação das medidas a serem realizadas e em seguida a definição dos sensores. Para realizar a medida de temperatura do óleo do motor utilizou-se o circuito integrado LM35 como sensor de temperatura, que gera um sinal de tensão linear na saída e com a leitura da mesma é possível definir a temperatura.

A velocidade do carro é medida na roda de freio dianteira, por um sensor de relutância magnética variável. Outro sensor igual foi colado na CVT - Sistema de Transmissão Contínua do veículo para medir a velocidade de rotação por minutos do motor. Os sinais gerados pelos sensores de rotação passarão por um circuito que será responsável por condicionar esse sinal.

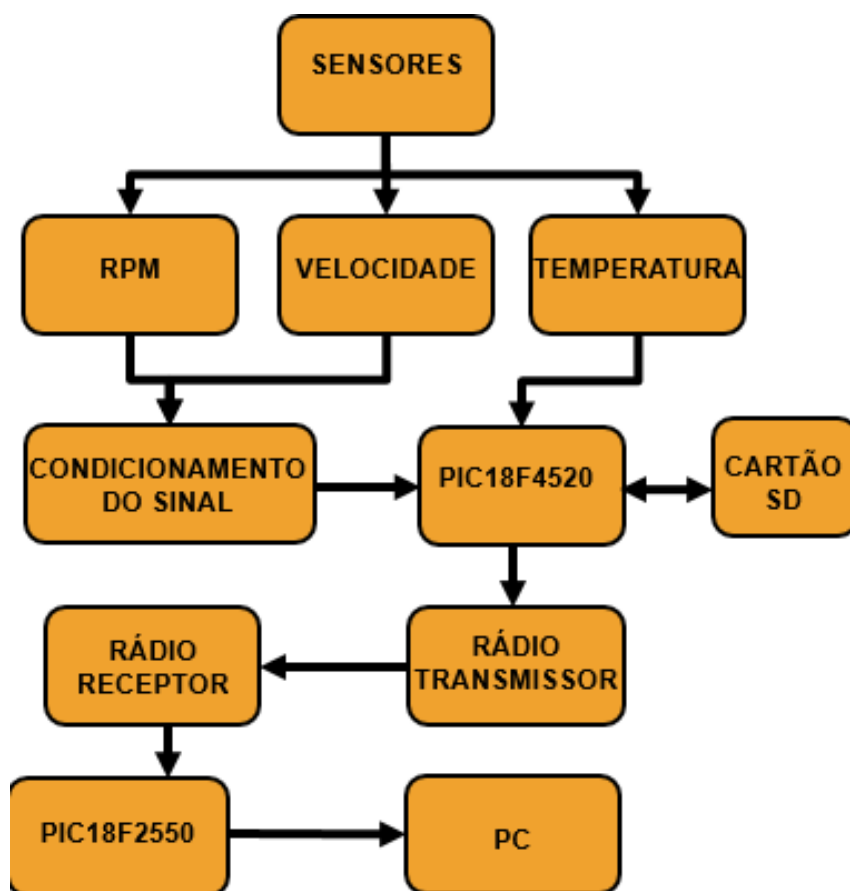
O microcontrolador PIC utilizado faz a leitura dos sinais da saída desses sensores através de um algoritmo que será implementado pelo software MikroC PRO for PIC, programando assim o PIC para que seja capaz de fazer a leitura correta dos dados.

Posteriormente esses dados coletados serão gravados em um cartão SD, sendo assim possível ler os dados no computador com facilidade. Além de serem gravados no cartão, os dados são transmitidos através de um módulo de rádio frequência transmissor para um receptor do modelo, onde há um outro PIC capaz de receber esses dados em tempo real e fazer uma comunicação serial assíncrona com o computador.

Realizados todos estes procedimentos, foram feitos testes neste sistema de telemetria, para que se possa saber se o sistema pode ser utilizado de forma correta e precisa.

A Figura 1 representa um fluxograma geral do trabalho que mostra de forma sucinta os passos a serem seguidos para chegar ao objetivo principal.

Figura 1 – Diagrama de blocos do sistema



Fonte: Autoria própria.

3.1 MikroC PRO for PIC

O MikroC PRO for PIC é um software da Microelektronika de programação para dispositivos PIC da Microchip. Dispõe de várias bibliotecas de hardware e software além de ferramentas adicionais e exemplos de códigos que facilitam seu uso. (MikroC, 2015).

Algumas bibliotecas do software utilizadas ao longo do desenvolvimento do projeto são citadas abaixo:

- SPI Library: Biblioteca para comunicação SPI em modo mestre, e para comunicação do microcontrolador com outros dispositivos, como: conversores A/D e D/A, MAX7219, LTC1290 e cartões de memória flash.

- EEPROM Library: Biblioteca para execução de leitura/escrita em determinado endereço.
- UART Library: Biblioteca utilizada para comunicação com outros dispositivos através do protocolo RS-232.
- MMC Library: Biblioteca para acessar dados em MultiMediaCard via comunicação SPI. Suporta cartões de memória SD e de alta capacidade SDHC.

3.1.1 MultiMediaCard Library

As rotinas de gerenciamento de arquivos podem ser usadas apenas com o sistema de arquivos FAT16, lendo e criando arquivos apenas a partir de um diretório raiz. A biblioteca usa o módulo SPI para comunicação, que deve ser inicializado antes de usar a biblioteca. Ela funciona apenas com a família PIC18F, que são microcontroladores de 8bits.

Abaixo algumas das principais funções da biblioteca (MikroC, 2015):

- Mmc_Fat_Init: inicializa o cartão MMC/SD, lê o setor de inicialização MMC/SD FAT16 e extrai dados necessários pela biblioteca.
- Mmc_Fat_Assign: atribui arquivo para operações como ler, escrever e excluir. Todas as operações de arquivos subsequentes serão aplicadas em um arquivo atribuído.
- Mmc_Fat_Append: abre o arquivo atualmente atribuído para anexar. Após esta função, os ponteiros do arquivo de execução serão posicionados após o último byte no arquivo, então qualquer operação posterior de gravação de arquivo começará a partir daí.
- Mmc_Fat_Write: grava o número solicitado de bytes para o arquivo atualmente atribuído aberto para gravação.

3.1.2 SPI Libray

Abaixo algumas funções da biblioteca (MikroC, 2015):

- SPI1_Init (): Configura e aciona o módulo SPI.
- SPI1_Init_Advanced (**unsigned short** master_slav, **unsigned short** data_sample, **unsigned short** clock_idle, **unsigned short** transmit_edge): os parâmetros mode, data_sample e clock_idle configuram o módulo e possuem as seguintes opções (Figura 2):

Figura 2 - Configuração SPI.

| Description | Predefined library const |
|---|--------------------------|
| SPI work mode: | |
| Master clock = Fosc/4 | _SPI_MASTER_OSC_DIV4 |
| Master clock = Fosc/16 | _SPI_MASTER_OSC_DIV16 |
| Master clock = Fosc/64 | _SPI_MASTER_OSC_DIV64 |
| Master clock source TMR2 | _SPI_MASTER_TMR2 |
| Slave select enabled | _SPI_SLAVE_SS_ENABLE |
| Slave select disabled | _SPI_SLAVE_SS_DIS |
| Data sampling interval: | |
| Input data sampled in middle of interval | _SPI_DATA_SAMPLE_MIDDLE |
| Input data sampled at the end of interval | _SPI_DATA_SAMPLE_END |
| SPI clock idle state: | |
| Clock idle HIGH | _SPI_CLK_IDLE_HIGH |
| Clock idle LOW | _SPI_CLK_IDLE_LOW |
| Transmit edge: | |
| Data is valid on low to high edge | _SPI_LOW_2_HIGH |
| Data is valid on high to low edge | _SPI_HIGH_2_LOW |

Fonte – MikroC, 2015.

3.2 Comunicação do PIC18F4520 com cartão SD

A comunicação entre o cartão e o microcontrolador é realizada através de um protocolo de comunicação serial síncrona e a biblioteca de sistema de arquivos

MultiMediaCard Library. Na Figura 3 tem-se um fluxograma que demonstra de forma simplificada como vai ocorrer essa comunicação entre o PIC e o cartão SD.

Figura 3 – Diagrama de blocos de comunicação.



Fonte – Autoria própria.

3.2.1 Comunicação SPI

O PIC18F4520 possui um módulo MSSP (Master Synchronous Serial Port - Porta de Comunicação Serial Síncrona Mestre). Esse módulo permite a implementação de dois protocolos muito usados em comunicação entre circuitos integrados, o SPI (Serial Peripheral Interface) e o I2C (Inter-Integrated Circuits).

O SPI é um protocolo utilizado para interconectar circuitos integrados. É um método que sustenta a comunicação de sistemas digitais complexos, como A/D e D/A, memórias EEPROM e flash, real time clocks (RTC), sensores com interface de comunicação, cartões de memória, controladores LCD e unidades de transmissão e recepção de dados UART.

A comunicação SPI é do tipo que possui um mestre e um escravo e pode ser implementada de uma forma que os dados possam ser transmitidos e recebidos

simultaneamente (full-duplex), e é configurado através da biblioteca SPI Library. Esse método foi escolhido devido a sua compatibilidade com cartões de memória SD.

São especificados três pinos de comunicação de dados: clock (SCK), entrada de dados (SDI) e saída de dados (SDO). Além destes, cada periférico SPI deve ter um sinal de habilitação baixo ativo, normalmente chamado de chip select (CS) ou chip enable (CE). A Tabela 1 abaixo apresenta os sinais, suas funções e os pinos correspondentes no PIC18F4520.

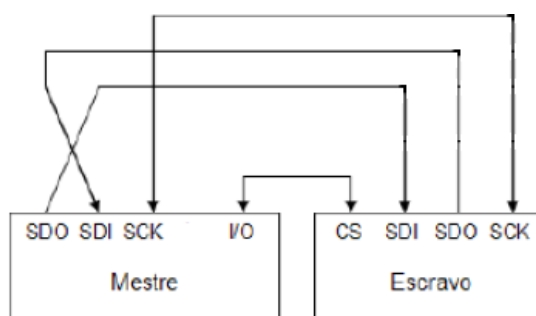
Tabela 1 – Sinais, pinos e funções PIC18F4520.

| SINAL | PINO | FUNÇÃO |
|------------|------|---|
| SDO | RC5 | SAÍDA DE DADOS |
| SDI | RC4 | ENTRADA DE DADOS |
| SCK | RC3 | MODO MESTRE: SAÍDA DE CLOCK MODO ESCRAVO: ENTRADA DE CLOCK |
| CS | RC0 | HABILITAÇÃO NO MODO ESCRAVO |

Fonte - Aatoria Própria.

A conexão entre os dispositivos SPI deve ser feita conforme o diagrama da Figura 4 abaixo. A saída do mestre (SDO) é ligada na entrada dos escravos (SDI) e a saída dos escravos (SDO) é ligada na entrada do mestre (SDI). Alguns fabricantes utilizam também a nomenclatura MISO (Master Input, Slave Output) e MOSI (Master Output, Slave Input) para simbolizar os pinos de dados.

Figura 4 - Conexão SPI



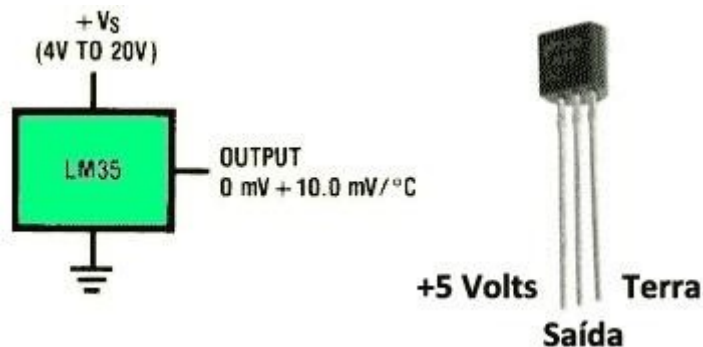
Fonte: Sistemas microcontrolados (2013).

de temperatura com facilidade e sem ajustes tornando custos mais baixos, com base nessas características o sensor foi escolhido.

A pinagem do LM35 é mostrada na Figura 6 e as suas características técnicas (Baú da eletrônica,2014) são:

- Calibrado diretamente em graus Celsius (Centígrados).
- Linear + 10,0 mV/°C (fator de escala).
- Precisão de 0,5°C (a 25°C).
- Faixa de operação de -55°C até 150°C.
- Adequado para aplicações remotas.
- Baixo custo / Não necessita de calibração no circuito.
- Alimentação 4V-20V.
- Menos de 60µA de corrente de passagem.
- Baixo auto aquecimento, 0,08°C em ar ambiente.
- Não linearidade apenas mais ou menos ¼°C.
- Baixa impedância de saída, 0,1 Ohm por 1mA de carga.

Figura 6 - Pinagem do LM35.



Fonte: Nova eletrônica (2013).

3.3.2 Sensor de relutância magnética variável

Os sensores de relutância variável são usados para medir a posição e velocidade de componentes metálicos em movimento. O sensor a ser utilizado é o MTE-THOMSON mostrado na Figura 7. Este sensor é composto por um ímã

permanente, um polo ferromagnético e uma bobina. Quando a roda executa uma rotação, a relutância do fluxo dentro da bobina muda, bem como o fluxo de ligação que dentro dela se encontra, cujos resultados, refletidos numa mudança de tensão, são medidos por um circuito externo (Oshiro e Kague, 2012).

Para gerar um sinal o sensor não precisa de uma alimentação, como no sensor de efeito de Hall. O sinal é gerado por indução eletromagnética. Quando o dente da roda dentada se aproxima do sensor, a tensão aumenta devido à variação do fluxo magnético. Quando o dente da roda dentada se alinha perfeitamente com o sensor, a tensão é nula. Quando deixa de estar alinhado com o sensor, a tensão cai devido à variação do fluxo magnético. Quando o dente da roda dentada estiver totalmente desalinhado com o sensor, a tensão volta a ser nula.

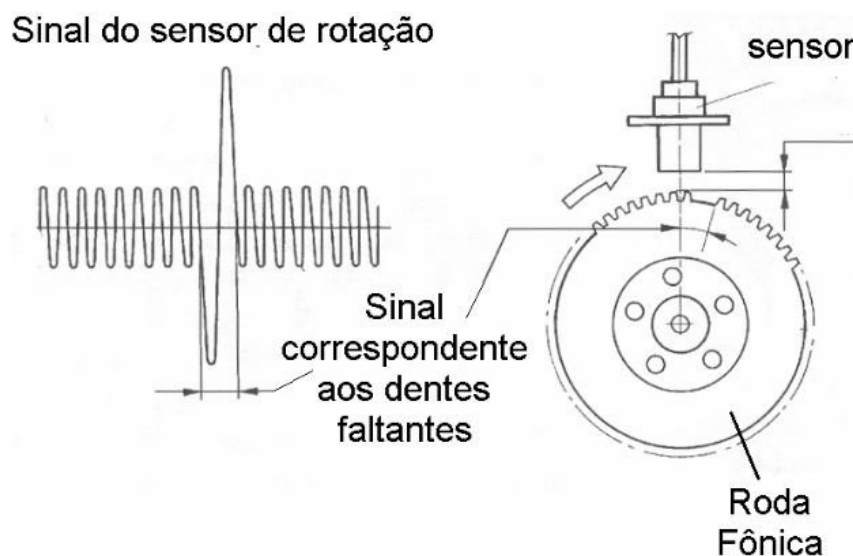
Figura 7 - Sensor de relutância magnética variável MTE-THOMSON.



Fonte: MTE-THOMSON.

O sinal padrão gerado pelo sensor e o modo como ele é instalado estão expostos na Figura 8.

Figura 8 - Sinal gerado pelo sensor de rotação.



Fonte: Giga de testes para sensor de relutância magnética variável (2012).

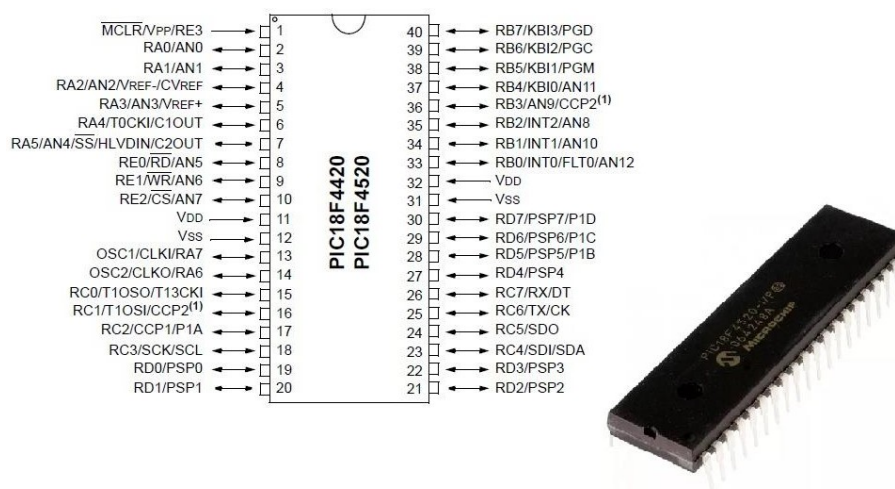
3.4 Microcontrolador

Um microcontrolador é um pequeno computador, que possui um processador, pinos de entrada e saída, periféricos e memória. Através da programação dos microcontroladores podemos controlar suas saídas, tendo como referência as entradas e um programa interno. O que diferencia os diversos tipos de microcontroladores, são as quantidades de memória interna (programa e dados), velocidade de processamento, quantidade de pinos de entrada e saída (I/O), alimentação, periféricos, arquitetura e set de instruções (Martins e Torres,2011).

3.4.1 PIC18F4520

O PIC é um circuito integrado produzido pela Microchip Technology Inc., que pertence a categoria dos microcontroladores, ou seja, um único dispositivo contém todos os circuitos necessários de um sistema digital programável.

Figura 9 - PIC18F4520.



Fonte: Microchip PIC18F4520.

É um microcontrolador de 8bits com Arquitetura RISC (set de instruções reduzido) que possui 40 pinos (mostrado na Figura 9), suas principais características são (Microchip Technology,2009):

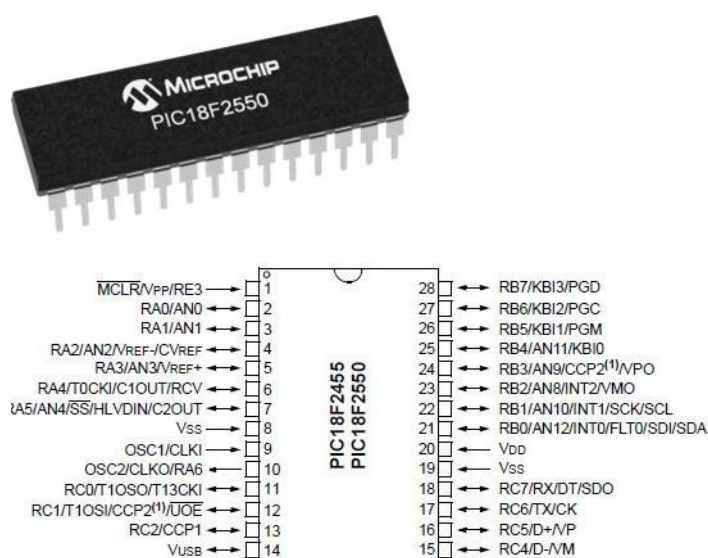
- Memória flash para armazenamento de programa 32 Kbytes.
- Memória EEPROM de dados 256 bytes.
- 36 portas configuráveis como entradas ou saídas digitais.
- 13 portas configuráveis como canais de entradas analógicas.
- 4 temporizadores de 8 e 16 bits.
- Frequência de operação de até 40 MHz.
- 2 Módulos CCP (Capture/Compare/PWM).
- Porta de comunicação serial, Porta de comunicação USB 2.0.
- Temperatura de trabalho: -40°C a +85°C.
- Tensão de Operação: de 2,0V à 5,5V.

3.4.2 PIC18F2550

É um microcontrolador de 8bits que possui 28 pinos (Figura 10), e suas principais características são (Microchip Technology,2009):

- Memória flash para armazenamento de programa 32 Kbytes.
- Memória EEPROM de dados 256 bytes.
- 25 portas configuráveis como entradas ou saídas digitais.
- 10 portas configuráveis como canais de entradas analógicas.
- 4 temporizadores de 8 e 16 bits.
- Frequência de operação de até 48 MHz.
- 2 Módulos CCP (Capture/Compare/PWM).
- Porta de comunicação serial, Porta de comunicação USB 2.0.
- Temperatura de trabalho: -40°C a +85°C.
- Tensão de Operação: de 2,0V à 5,5V.

Figura 10 - PIC18F2550



Fonte – Autoria Própria.

3.5 Cartões de memória flash

A memória flash é uma memória não volátil, capaz de preservar os dados armazenados por um longo tempo sem precisar de alimentação elétrica.

Devido à sua velocidade elevada de escrita e leitura, durabilidade e o seu baixo consumo, a memória flash é ideal para numerosas aplicações, como por exemplo:

- As máquinas fotográficas digitais.
- Celulares.
- Impressoras.
- Computadores portáteis.
- Dispositivos de leitura ou de registro sonoro como um MP3 Player.

Além disso, este tipo de memória não possui elementos mecânicos, o que lhes confere uma grande resistência a choques (Kioskea, 2012).

Existe um grande número de formatos de cartões de memória, não compatíveis entre si, criados por vários fabricantes. Os cartões de memória mais utilizados são:

- Compact Flash.
- Secure Digital (chamadas SD Card).
- Memory Stick.
- SmartMedia.
- MMC (MultimediaCard).

3.5.1 Cartões de memória SD

O Secure Digital (SD) (Figura 11) foi lançado em 1999 e são fruto de uma parceria entre SanDisk, Panasonic e Toshiba. Trata-se de uma variação do tipo MMC, possuindo quase as mesmas dimensões: 24 mm x 32 mm x 2,1 mm. Entre seus diferenciais estão: compatibilidade com determinações de segurança da Secure

Digital Music Initiative (SDMI), que visa evitar a distribuição ilegal de músicas; uma pequena trava de segurança que impede a eliminação de dados do dispositivo; e melhor desempenho na transferência de dados (Alecrim, 2017).

Cartões SD são, atualmente, o tipo mais popular de memória de armazenamento de dados, sendo amplamente utilizados em celulares, câmeras digitais, consoles de videogame e outros dispositivos portáteis. Tamanho sucesso se deve, em parte, às suas variações: os cartões miniSD e microSD, que possuem dimensões reduzidas, como indicam seus nomes (20 mm x 21,5 mm x 1,4 mm e 11 mm x 15 mm x 1 mm, respectivamente). Independente da categoria, os cartões SD podem ser encontrados atualmente com capacidades bastante altas, como 32 ou 64 GB.

Figura 11 - Cartão de memória SD.



Fonte: Cartão de memória flash (2014).

3.5.2 Módulo adaptador cartão MicroSD

Para conectar o cartão com o microcontrolador foi utilizado um módulo adaptador de cartão MicroSD (Figura 12), que permite a comunicação do cartão com o PIC. Ele foi escolhido por ser prático de usar e simples de inserir no projeto. Esse módulo possui uma entrada para cartões MicroSD com os 6 pinos necessários para a comunicação com o PIC por meio dos pinos SPI:

- GND – Terra.
- VCC – Alimentação 5V.
- MISO (Master Input Slave Output) ou SDO (Slave Data Output).
- MOSI (Master Output Slave Input) ou SDI (Slave Data Input).
- CS (Chip Select).
- SCK (Serial Clock).

Figura 12 - Módulo adaptador cartão MicroSD

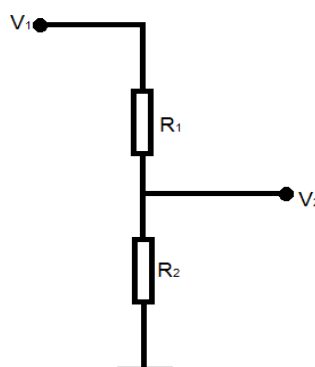


Fonte – Autoria própria.

Uma observação importante é que mesmo que a alimentação do módulo seja de 5V, os sinais SDI, SDO e SCK são de 3.3V. Então é necessário utilizar três divisores de tensão para que a tensão de saída dos sinais do PIC baixe de 5V para 3.3V do lado do cartão. Portanto é necessário fazer um divisor de tensão, mostrado na Figura 13, onde V_1 é a tensão dos pinos do PIC e V_2 a tensão desejada. Adotando um valor de R_2 igual a 3,3k Ω e R_1 igual a 2,2k Ω utilizamos a Equação 1 para calcular o valor da tensão de saída.

$$V_2 = V_1 * \frac{R_2}{R_2 + R_1} = 5 * \frac{3300}{3300 + 2200} = 3 \text{ V} \quad (1)$$

Figura 13 - Circuito divisor de tensão.



Fonte – Autoria própria.

Como o resultado dado pela Equação 1 é 3V que é um valor de tensão satisfatório para o circuito, então os resistores R_1 e R_2 usados nos pinos SDI, SDO e SCK serão de 2,2k Ω e 3,3k Ω respectivamente.

3.6 Módulo Transmissor e Receptor de Radiofrequência em 433 MHz

O módulo RF 433 MHz utilizado atua em uma faixa de frequência usada por rádio amadores, é uma das opções mais baratas que temos para montar equipamentos que precisam se comunicar sem fio.

Seu modo de operação é AM – Modulação em Amplitude, é utilizado em sistemas de alarme, controles remotos, aquisição de dados e robótica em geral.

3.6.1 Transmissor MX-FS-03V

É um módulo AM de pequeno tamanho, possui uma potência de 10mW sob uma tensão de operação de 3,5V. Fácil de usar e pode ser conectado direito ao microcontrolador. A Figura 14 mostra a sua pinagem e abaixo principais características são:

- Frequência de operação 433/315 MHz.
- Taxa de transferência: 4Kb/s.
- Tensão de operação entre 3,5 – 12V.
- Distância de transmissão entre 20 – 200m.
- Baixo custo.

Figura 14 - Pinagem MX-FS-03V

| PINO | |
|------|--------|
| 1 | DATA |
| 2 | VCC |
| 3 | GND |
| 4 | ANTENA |



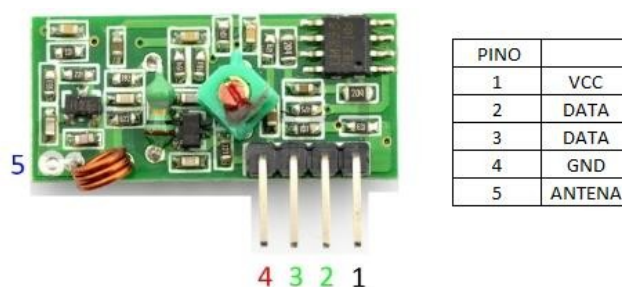
Fonte – Autoria própria.

3.6.2 Receptor MX-05V

O módulo MX-05V é um receptor de sinal RF de baixo custo e baixo consumo de corrente, funcionando bem em conjunto com o transmissor MX-FS-03V. A Figura 15 mostra sua pinagem e suas principais características são:

- Tensão de operação: 5V.
- Corrente de operação: 4mA.
- Frequência de recepção: 433MHz.
- Sensibilidade: -105dB.

Figura 15 - Pinagem MX-05V



Fonte – Autoria própria.

3.7 Comunicação do PIC18F2550 com PC

A comunicação entre o microcontrolador e o PC é realizada através de uma comunicação serial assíncrona, para realizar a comunicação foi usado o Módulo Conversor USB TTL CP2102 Serial RS232.

Esse módulo (Figura 16) foi utilizado para simplificar e facilitar a comunicação serial do PIC18F2550 computador. Ele integra o chip CP2102 responsável por converter uma serial USB para RS232 em nível TTL, além de alimentar o circuito. Suas principais características são:

- Baud Rate de 300Kbps.
- Taxa de transmissão de 1Mbps.

- Tensão de trabalho: 4V~5.25 (Alimentado pela porta USB).
- Tensão de saída: 3,3V ou 5V.
- LED's Indicadores: Alimentação (Power) e TXD (Transmissão).
- Corrente limite: 100mA (Com proteção).

Figura 16 - Módulo Conversor USB TTL CP2102 Serial RS232



Fonte – Autoria própria.

4 DESENVOLVIMENTO DO PROJETO

O início do desenvolvimento desse trabalho foi dado a partir do momento em que foram identificadas as medidas a serem realizadas. A partir da aquisição dos dados, foi analisada a necessidade do condicionamento do sinal gerado pelo sensor de relutância.

Após a interpretação dos sinais gerados pelos sensores pelo microcontrolador, foi realizado um estudo para a gravação dos dados no cartão de memória SD. Optou-se por usar o software MikroC PRO for PIC pelo fato desse software oferecer mais recursos nas bibliotecas de comunicação SPI e de acesso ao cartão de memória SD. São respectivamente as bibliotecas SPI Library e MultiMediaCard Library.

Com a aquisição e gravação dos dados pelo PIC18F4520, iniciou-se o desenvolvimento da segunda parte do projeto, onde os dados são enviados através de um módulo RF transmissor, para um outro receptor, onde o microcontrolador PIC18F2550 realiza a leitura destes dados e em seguida exibe os mesmos na tela do computador.

4.1 Aquisições dos dados

O primeiro passo do projeto após a identificação das medidas a serem realizadas e a definição dos sensores, foi a simulação desse processo de aquisição dados pelo software Proteus.

4.1.1 Temperatura

Para o LM35, foi feito um código simples em que sua saída é ligada no pino AN0 do PIC, que é configurado como uma entrada analógica. Assim é feita a leitura de tensão em mV e ocorre uma ADC de 10bits, isso significa que o valor analógico é convertido em um número digital de 10bits, que pode ter 2^{10} valores, ou seja, vai variar entre 0 e 1023. A relação entre o resultado da conversão e a leitura máxima possível e a relação entre a tensão de entrada e a tensão máxima de entrada são

iguais, isso acontece porque a tensão de entrada é uma fração da tensão máxima possível e o resultado da saída do conversor é a mesma fração do valor máximo possível, que é 1023.

Supondo que o LM35 faça uma leitura de 35°C, o conversor recebe uma tensão de 0,350 V, assim temos o resultado da conversão com a Equação 2.

$$\begin{aligned} \text{[bits]} &= \frac{\text{[tensão]} * \text{[max]}}{\text{[max]}} \tag{2} \\ &= \frac{0,350 * 1023}{5} \\ &= 71,61 \end{aligned}$$

Se o processo for invertido utilizando a relação linear do sensor em que cada 10 mV temos 1°C, encontramos então o resultado final, que é a temperatura lida pelo sensor através da Equação 5. A simulação com o software Proteus é mostrada na Figura 17.

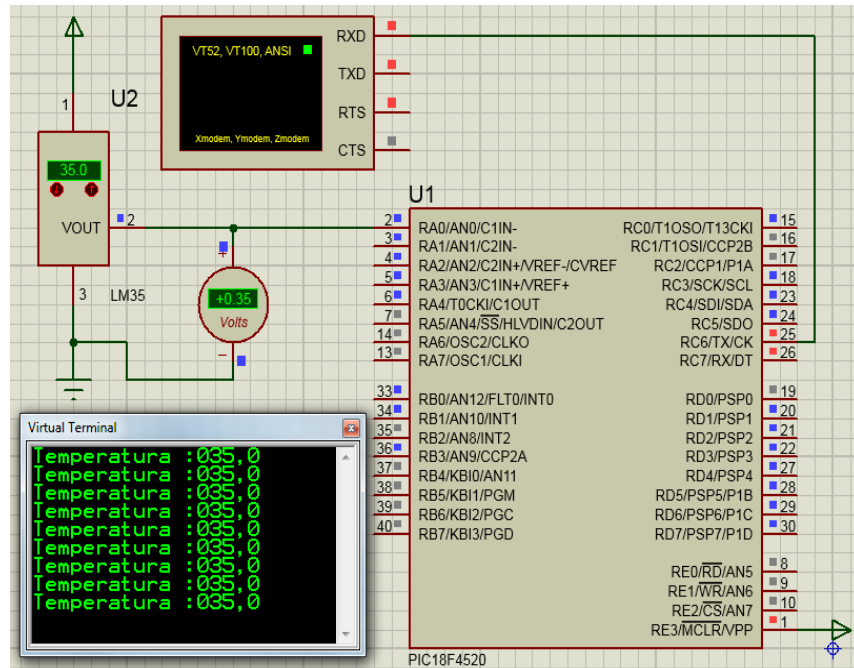
$$\begin{aligned} \text{[bits]} &= \frac{\text{[bits]}}{10} \tag{3} \\ &= 0,01 * \text{[bits]} \end{aligned}$$

$$\begin{aligned} \text{[bits]} &= \frac{0,01 * \text{[bits]} * \text{[max]}}{\text{[max]}} \tag{4} \end{aligned}$$

$$\begin{aligned} \text{[bits]} &= \frac{\text{[bits]} * \text{[max]}}{0,01 * \text{[max]}} \tag{5} \end{aligned}$$

$$\begin{aligned} \text{[bits]} &= \frac{71,61 * 5}{0,01 * 1023} = 35^\circ \tag{6} \end{aligned}$$

Figura 17- Simulação LM35



Fonte – Autoria própria.

4.1.2 Velocidade e RPM

A velocidade será medida no disco de freio da roda dianteira do veículo. Esse disco possui 8 furos, toda vez que o sensor passa por um furo gera um pico na tensão, como mostrado no sinal da Figura 8. Então foi utilizado o modo de captura (CCP) do PIC, para que a cada pico de tensão, ou seja, cada borda de subida gerasse uma interrupção e fosse contado um pulso, e assim 8 bordas de subida seriam uma volta, devido a disposição de 8 furos na roda.

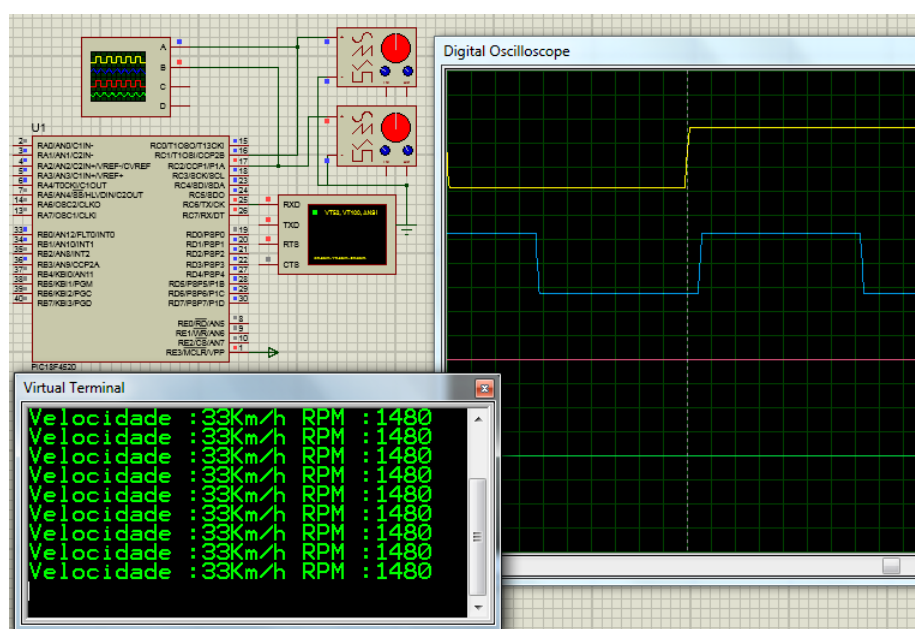
Com o número de voltas é possível calcular a quantidade de rotações por minuto da roda, e em seguida calcular a velocidade em Km/h através do valor do raio dinâmico da roda ($r = 1,627847$), que é um valor constante que o raio assume quando está em movimento. Supondo que foram contados 45 pulsos, a Equação 7 calcula o número de rotações por minuto e a Equação 8 faz a conversão para Km/h.

$$RPM = \left(\frac{45}{8} \right) * 60 = 337,5 \quad (7)$$

$$\frac{33 \text{ km/h}}{h} = \frac{1,627847}{1000} * 337,5 \quad (8)$$

A velocidade rotação do motor é medida da mesma forma que a velocidade do carro, a única diferença é que foi colocado na CVT do veículo três imãs, e os pulsos aconteciam quando o sensor passava pelo vão entre os imãs. Portanto como na Equação 6, ao invés de dividirmos 60 por 8, dividiríamos por 3. A Figura 18 demonstra simulação da velocidades e RPM no software Proteus.

Figura 18 -Simulação velocidade e RPM



Fonte – Autoria própria.

4.2 Gravação cartão SD

A comunicação com o cartão SD foi feita através do protocolo SPI utilizando a biblioteca SPI Library, após iniciar o protocolo SPI, com a biblioteca MultiMediaCard é possível inicializar o cartão com o comando Mmc_Fat_Init. É verificado o retorno do comando Mmc_Fat_Init, e for igual a 1 é porque o cartão não foi inicializado, já se for igual a 0 o LED verde acende sinalizando que o cartão foi reconhecido e inicializado.

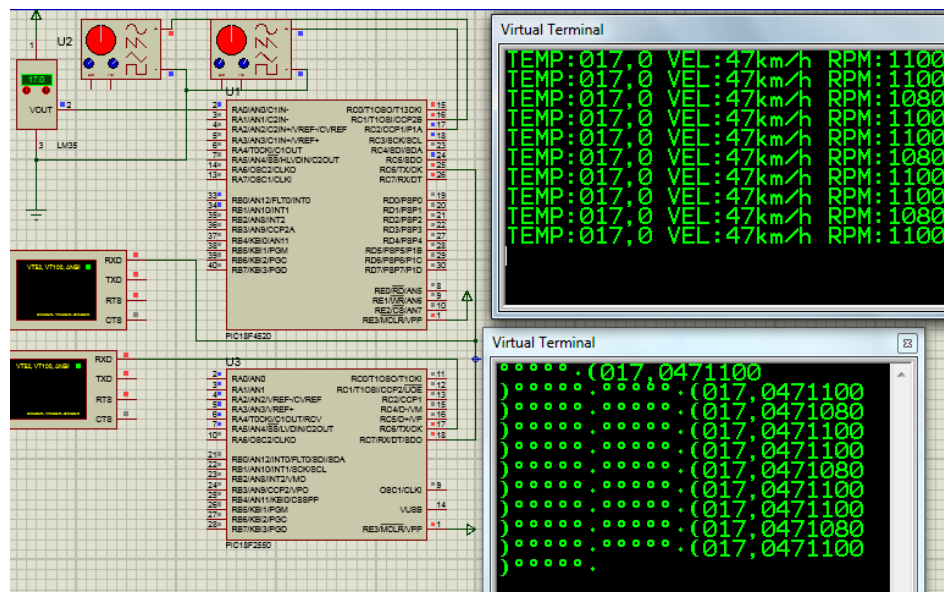
Após a iniciação do cartão executa-se o comando Mmc_Fat_Assign, que cria um arquivo de texto chamado "DATA.txt" caso ele não exista, se existir o comando

Na prática o pino RC6/TX do PIC18F4520 é ligado no módulo transmissor e o pino RC7/RX do PIC18F2550 no receptor, dessa forma a comunicação é feita com a própria biblioteca da UART.

A comunicação foi realizada através de um protocolo de comunicação desenvolvido no projeto que decodificou os dados, para que fosse obtido o menor número de ruídos e erros de transmissão entre os módulos RF. O receptor demora alguns segundos para ajustar seu ganho, então qualquer dado transmitido deve começar com um preâmbulo antes dos dados principais, pois ai o receptor terá um tempo para ajustar seu ganho. Preâmbulo é um delimitador entre um pacote e outro de dados, é formado por 7bytes idênticos “10101010” e mais um último byte com o valor “10101011”.

Após o envio do preâmbulo, são utilizados 2 bytes um para iniciar e outro para finalizar o envio do pacote de dados principais. O start_byte = ‘(‘ e o stop_byte = ‘)’ , entre esses dois bytes estão os dados que interessam. Após o envio do “stop_byte”, é enviado novamente o preâmbulo e a cada 3 segundos esse processo se repete e dados são transmitidos. A recepção é feita pelo PIC18F2550, que foi programado para ler apenas os dados que estivessem entre o “start_byte” e “stop_byte”, e trata-los para serem enviados para o computador.

Figura 20 - Transmissão de dados



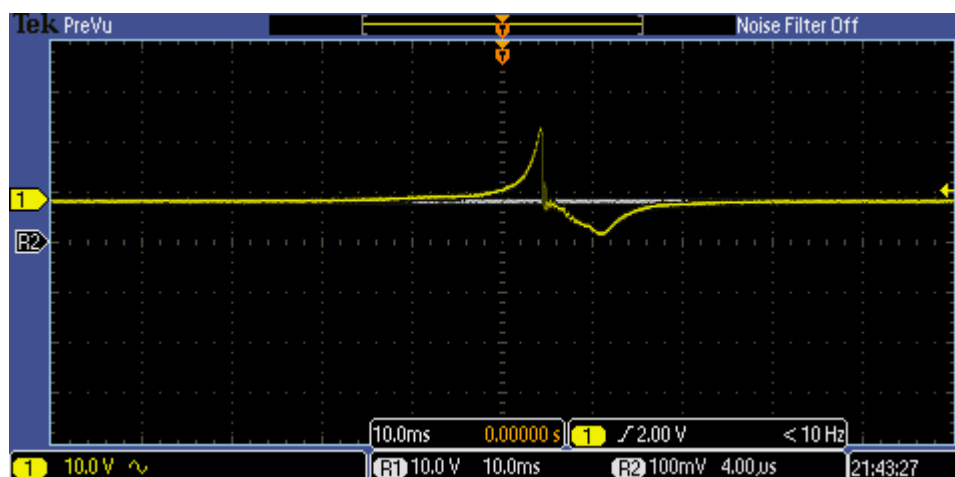
Fonte - Autoria própria.

5 RESULTADOS

5.1 Testes de bancada

Primeiramente realizou-se um teste com o sensor de rotação no laboratório para que pudesse saber a forma do seu sinal e se haveria a necessidade de um condicionamento do sinal gerado. Porém um problema surgiu, não houve a possibilidade de se utilizar um osciloscópio na oficina no Baja, pois a universidade não permite que o osciloscópio saia do laboratório. Então foi feito o teste improvisadamente no laboratório P111 do campus da universidade com o sensor e um ímã. Passando e retirando o ímã pelo sensor, foi obtido o sinal da Figura 21.

Figura 21 - Sinal gerado pelo sensor de rotação



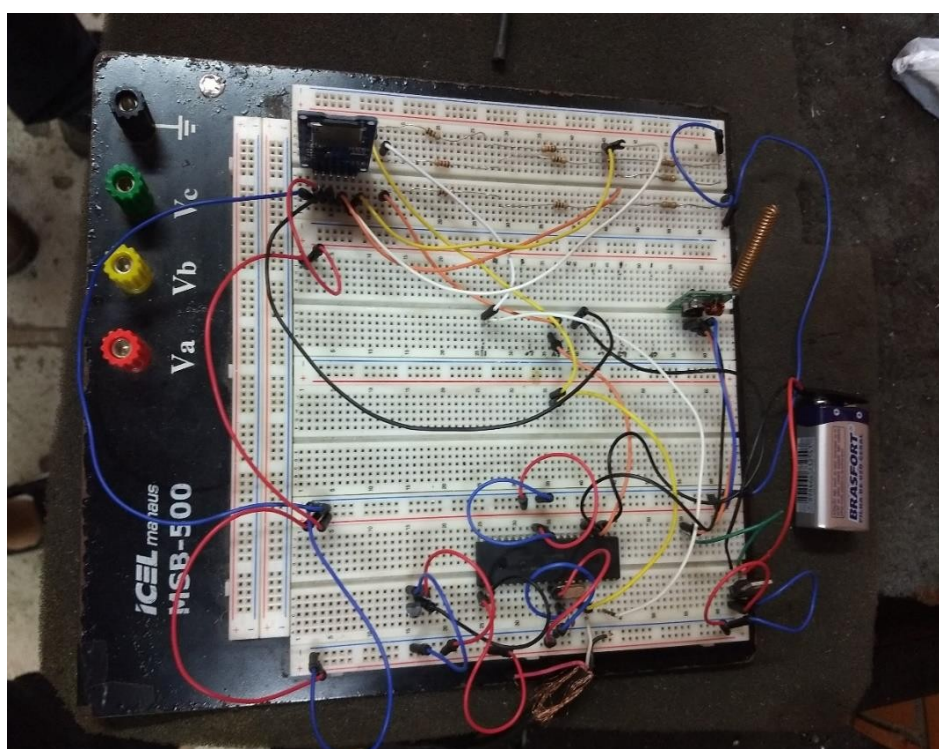
Fonte – Autoria própria.

O próximo passo foi a montagem do circuito responsável pela aquisição, gravação e transmissão dos dados no protoboard (Figura 22), chamaremos ele de Tx. Para a montagem de Tx, foram necessários os seguintes componentes:

- 3 resistores de 1k Ω .
- 3 resistores de 1,2k Ω .
- 3 resistores de 3,3k Ω .
- 1 LM7805.

- 1 LM35.
- 1 cristal 20MHz.
- 2 capacitores 22pF.
- 1 Led verde.
- 1 Led vermelho.
- 1 módulo adaptador de cartão microSD.
- 1 transmissor MX-FS-03V.
- 1 bateria de 9V com conector.
- 1 sensor de rotação.
- 1 cartão microSD.
- 1 PIC18F4520.

Figura 22 - Circuito Tx

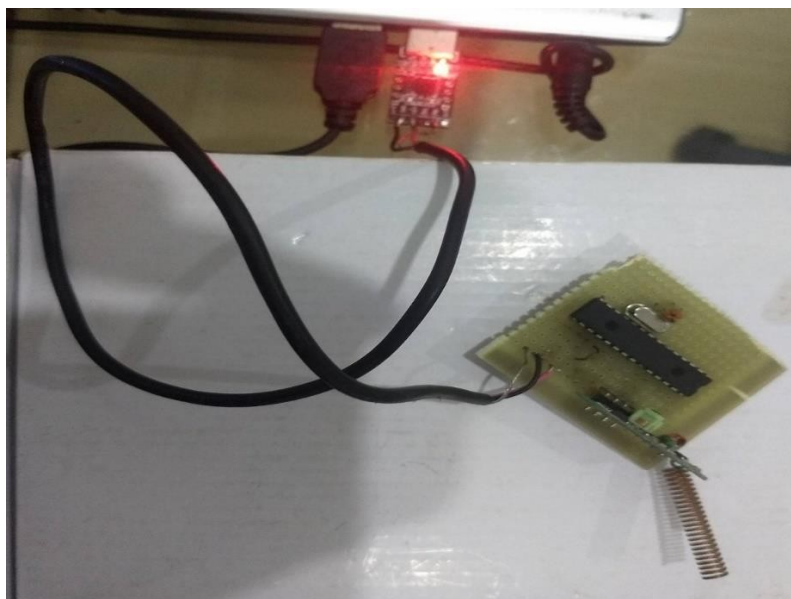


Fonte – Autoria própria.

Em seguida montou-se o circuito que recebe os dados enviados por Tx e envia para o computador, esse circuito é chamado de Rx (Figura 23). Como não havia a disponibilidade de outro protoboard esse circuito foi soldado diretamente em uma placa de fenolite perfurada. Os componentes necessários para a montagem, foram:

- 1 cristal de 20MHz.
- 2 capacitores de 15pF.
- 1 receptor MX-05V.
- 1 Módulo Conversor USB TTL CP2102 Serial RS232.
- 1 PIC18F2550.

Figura 23 - Circuito Rx ligado no computador



Fonte – Autoria própria.

Para realização do teste foi utilizado um disco semelhante ao que seria usado no veículo, a única diferença entre os dois é o raio dinâmico. O disco foi colocado para rodar em uma furadeira de bancada e em seguida foi posicionado o sensor de rotação como mostrado na Figura 24. Então em seguida ligou-se a furadeira, os circuitos Tx e Rx respectivamente nas Figuras 25 e 23 para que fosse iniciado o teste. O circuito receptor Rx estava a 15m de distância do transmissor.

Figura 24 – Montagem sensor e disco



Fonte – Autoria própria.

Figura 25 - Tx e furadeira ligados



Fonte – Autoria própria.

O resultado do teste foi como o esperado, houve a comunicação entre os módulos RF e a gravação dos dados no cartão. A Figura 26 representa os dados obtidos por Rx que foram enviados para o computador do lado esquerdo, e do lado direito os dados gravados no cartão SD pelo Tx.

Figura 26 - Resultados

```

PuTTY (inactive)
TEMP:031,0 VEL:57km/h RPM:0000
TEMP:
% VEL: km/h RPM:A>
TEMP:032,0 VEL:56km/h RPM:0000
TEMP:031,0 VEL:57km/h RPM:0000
TEMP:032,0 VEL:56km/h RPM:0000
TEMP:030,0 VEL:57km/h RPM:0000
TEMP:e VEL: km/h RPM:DL~
TEMP:034,0 VEL:56km/h RPM:0000
TEMP:032,0 VEL:57km/h RPM:0000
TEMP:031,0 VEL:56km/h RPM:0000
TEMP:027,0 VEL:57km/h RPM:0000
TEMP:032,0 VEL:56km/h RPM:0000
TEMP:032,0 VEL:56km/h RPM:0000
TEMP:030,0 VEL:57km/h RPM:0000
TEMP:e VEL: km/h RPM:DD^
TEMP:u' VEL: km/h RPM:0000
TEMP:R1,0 VEL:56km/h RPM:0000
TEMP:0DD,0 VEL:57km/h RPM:0000
TEMP:0W' VEL:>km/h RPM:
TEMP:' VEL: km/h RPM:N~
TEMP:032,0 VEL:56km/h RPM:0000
TEMP:023,0 VEL:57km/h RPM:0000
TEMP:030,0 VEL:56km/h RPM:0000
TEMP:J VEL: km/h RPM:L3
TEMP:jYe VEL: km/h RPM:a
TEMP:e VEL: km/h RPM:DLn
TEMP:030,0 VEL:56km/h RPM:0000
TEMP:031,0 VEL:56km/h RPM:0000
TEMP:030,0 VEL:57km/h RPM:0000
TEMP:028,0 VEL:57km/h RPM:0000
TEMP:z VEL: km/h RPM::3
TEMP:026,0 VEL:56km/h RPM:0000
TEMP:\' VEL: km/h RPM:
TEMP:031,0 VEL:56km/h RPM:0000
TEMP:$4 VEL: km/h RPM:D
TEMP:031,0 VEL:56km/h RPM:0000
TEMP:030,0 VEL:57km/h RPM:0000
TEMP:027,0 VEL:57km/h RPM:0000
TEMP: VEL: km/h RPM:
TEMP:030,0 VEL:56km/h RPM:0000
TEMP:027,0 VEL:57km/h RPM:0000
TEMP:029,0 VEL:56km/h RPM:0000
DATA - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
TEMP=034,0 VEL:00 km/h RPM: 0000|
TEMP=034,0 VEL:00 km/h RPM: 0000
TEMP=033,0 VEL:56 km/h RPM: 0000
TEMP=032,0 VEL:56 km/h RPM: 0000
TEMP=031,0 VEL:57 km/h RPM: 0000
TEMP=032,0 VEL:56 km/h RPM: 0000
TEMP=031,0 VEL:57 km/h RPM: 0000
TEMP=032,0 VEL:56 km/h RPM: 0000
TEMP=030,0 VEL:57 km/h RPM: 0000
TEMP=032,0 VEL:56 km/h RPM: 0000
TEMP=034,0 VEL:56 km/h RPM: 0000
TEMP=032,0 VEL:57 km/h RPM: 0000
TEMP=031,0 VEL:56 km/h RPM: 0000
TEMP=027,0 VEL:57 km/h RPM: 0000
TEMP=032,0 VEL:56 km/h RPM: 0000
TEMP=032,0 VEL:56 km/h RPM: 0000
TEMP=030,0 VEL:57 km/h RPM: 0000
TEMP=031,0 VEL:56 km/h RPM: 0000
TEMP=027,0 VEL:57 km/h RPM: 0000
TEMP=032,0 VEL:56 km/h RPM: 0000
TEMP=023,0 VEL:57 km/h RPM: 0000
TEMP=030,0 VEL:56 km/h RPM: 0000
TEMP=030,0 VEL:56 km/h RPM: 0000
TEMP=031,0 VEL:56 km/h RPM: 0000
TEMP=030,0 VEL:57 km/h RPM: 0000
TEMP=028,0 VEL:57 km/h RPM: 0000
TEMP=026,0 VEL:56 km/h RPM: 0000
TEMP=030,0 VEL:57 km/h RPM: 0000
TEMP=031,0 VEL:56 km/h RPM: 0000
TEMP=030,0 VEL:57 km/h RPM: 0000
TEMP=027,0 VEL:57 km/h RPM: 0000
TEMP=030,0 VEL:56 km/h RPM: 0000
TEMP=027,0 VEL:57 km/h RPM: 0000
TEMP=029,0 VEL:56 km/h RPM: 0000
TEMP=030,0 VEL:56 km/h RPM: 0000
TEMP=029,0 VEL:57 km/h RPM: 0000
TEMP=027,0 VEL:57 km/h RPM: 0000
TEMP=029,0 VEL:56 km/h RPM: 0000
TEMP=025,0 VEL:57 km/h RPM: 0000
TEMP=030,0 VEL:56 km/h RPM: 0000
TEMP=029,0 VEL:57 km/h RPM: 0000
TEMP=027,0 VEL:57 km/h RPM: 0000
TEMP=027,0 VEL:56 km/h RPM: 0000
TEMP=029,0 VEL:56 km/h RPM: 0000
TEMP=028,0 VEL:57 km/h RPM: 0000

```

Fonte – Autoria própria.

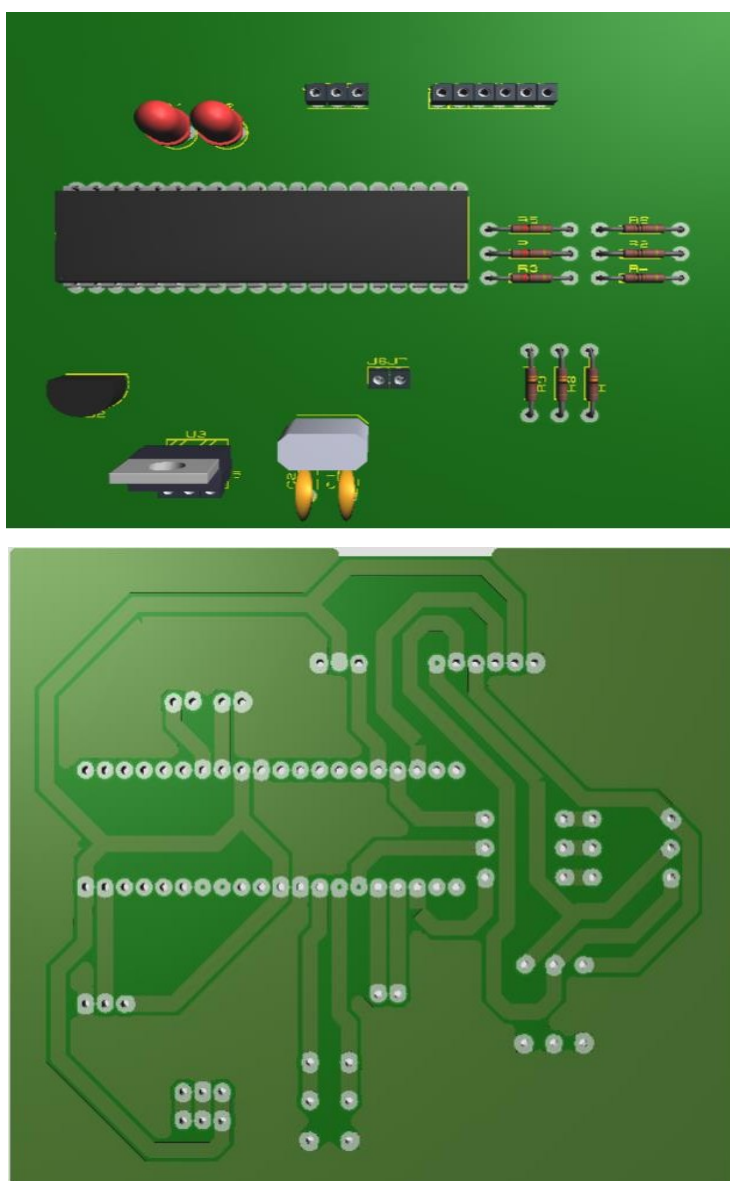
Analisando agora o valor da velocidade em Km/h, vemos que variou entre 56 e 57 Km/h. Para verificar esse valor a utilizou-se a Equação 7, e substitui-se a variável RPM pelo valor mostrado no tacômetro da Figura 24 que é 627,7.

$$\begin{aligned}
 \frac{v}{h} &= \frac{v_{\text{RPM}} * 60}{1000} = \frac{1,627847 * 627,7 * 60}{1000} \quad (9) \\
 &= 61 \text{ km/h}
 \end{aligned}$$

5.2 Placas de circuito impresso

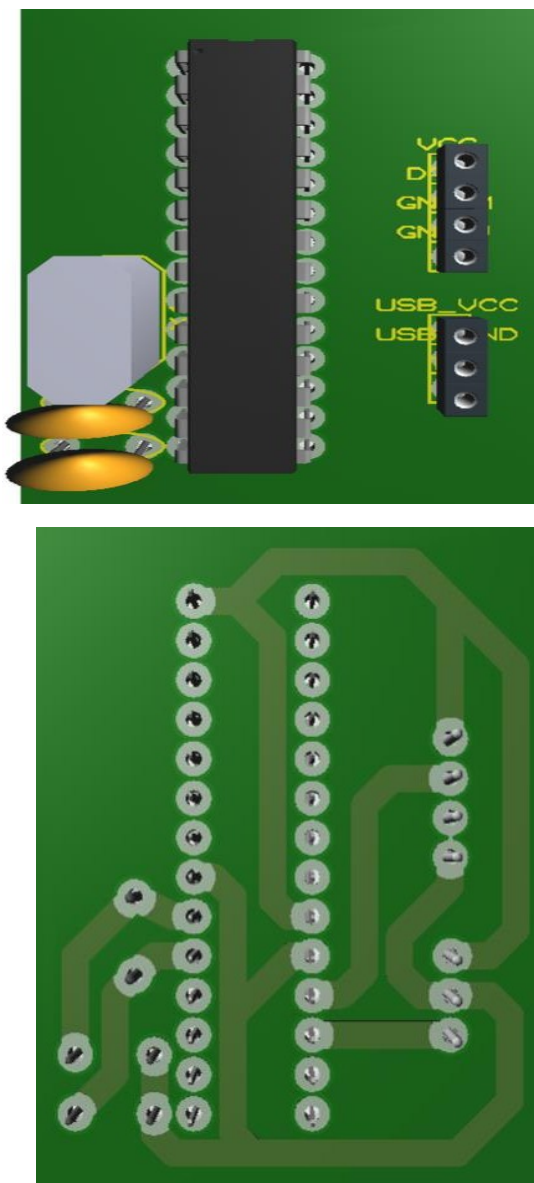
A placa de circuito impresso foi gerada no Proteus após os testes de bancada e as simulações no próprio software. As Figuras 27 e 28 mostram o modelo 3D das duas placas e as trilhas, de transmissão e recepção respectivamente.

Figura 27 - PCI transmissão



Fonte – Autoria própria.

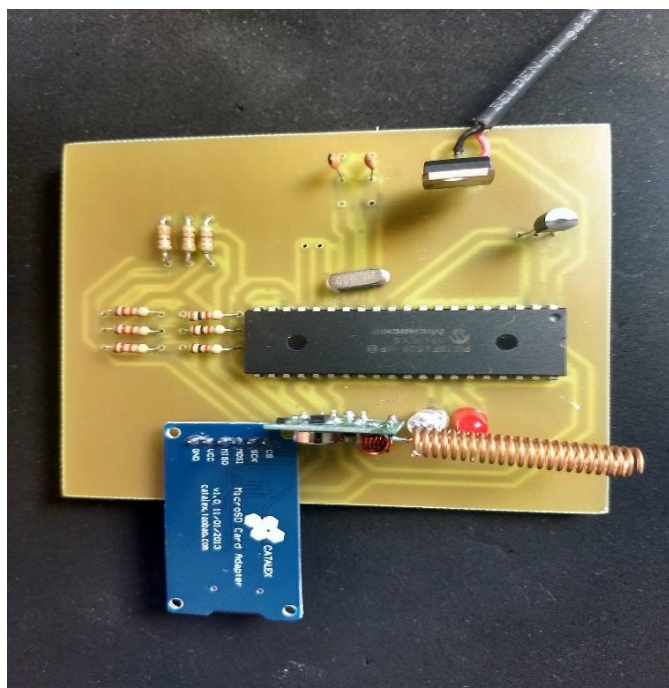
Figura 28 - PCI recepção



Fonte – Autoria própria.

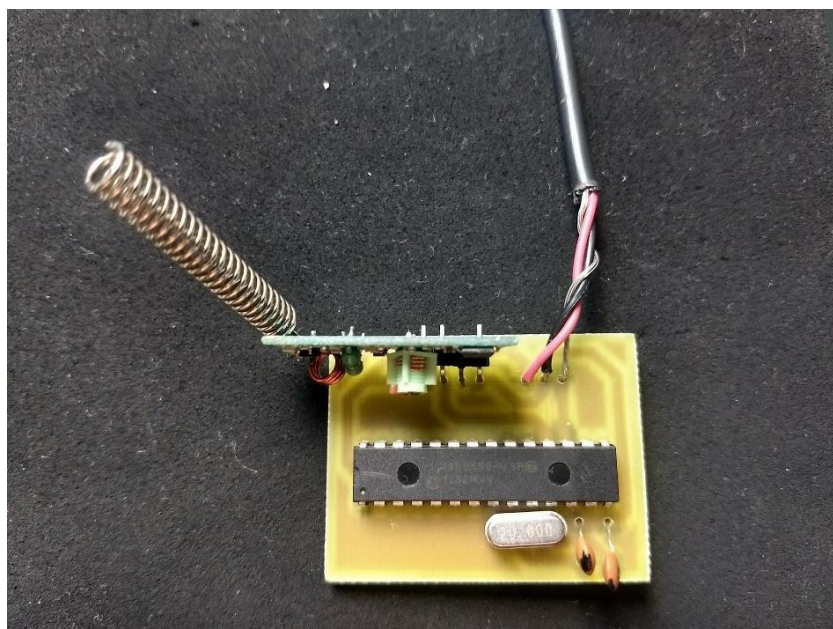
Após a simulação da placa, foram enviados os modelos para serem confeccionados na prototipadora da UTFPR – Câmpus Cornélio Procópio, onde as placas foram impressas e em seguida foi feita a solda dos componentes nas placas de transmissão Tx e recepção Rx mostradas na Figuras 29 e 30 respectivamente.

Figura 29 - PCI transmissão Tx



Fonte - Autoria própria.

Figura 30 - PCI Recepção Rx



Fonte - Autoria própria.

6 CONCLUSÃO

O trabalho realizado apresentou a instrumentação básica e sistema de telemetria para baja. A realização desse projeto envolveu conhecimentos sobre eletrônica analógica e digital, sistemas embarcados, comunicação via radiofrequência e leitura de sensores de velocidade e temperatura.

Os objetivos gerais e específicos do projeto foram realizados, porém nem todos como planejados. Algumas mudanças em relação a proposta inicial tiveram que ser realizadas, como a mudança do módulo de rádio Digi Xbee-PRO 900HP para um modelo de rádio com uma potência inferior, o que interferiu diretamente no projeto, que era criar um sistema de comunicação RF que atingisse uma distância maior que 1km, para que tivesse uma utilidade maior nas competições para a Equipe ProcoBaja em provas que necessitam de dados em tempo real.

Para o teste de bancada mostrada no Capítulo 5 foi utilizado uma lógica para que o PIC transmitisse e enviasse os dados a cada 5 segundos, para que depois pudessemos comparar os dados enviados com os gravados. Como podemos ver algumas linhas de dados enviadas para o computador são de caracteres ilegíveis, provenientes de ruídos que o módulo receptor lê no intervalo de 5 segundos, onde Tx não está enviando nada.

É possível perceber que a partir da quarta linha do arquivo DATA.txt na Figura 26, os dados enviados e gravados são os mesmos, isso aconteceu devido ao intervalo de ligação dos dois circuitos, onde o Tx foi ligado primeiro e gravou os dados, porém o Rx não recebeu e conseqüentemente não enviou para o computador pois ainda estava desligado.

A Equação 9 nos dá um valor com 4 Km/h a mais que o calculado pelo microcontrolador. Isso acontece devido a diferença do raio dinâmico dos discos. O disco utilizado no teste possui um raio menor do que o que será realmente usado no veículo, e ainda assim a variação de quilometragem não é grande, observando que não foi usado nem um tipo de condicionamento do sinal gerado pelo sensor, e como mostra os cálculos o valor é aceitável para a aplicação.

Foi decidido então não usar nenhum tipo de condicionamento de sinal na saída do sensor de rotação, devido também a falta de uma informação mais completa

sobre o sinal gerado pelo mesmo, pois não houve possibilidade da utilização do osciloscópio na oficina.

Depois do teste de bancada e a confecção das placas de circuito impresso, a intenção era fazer uma instalação simples da placa de transmissão no veículo e realizar mais alguns testes de campo. Porém uma enorme dificuldade encontrada em todo desenvolvimento do trabalho foi o andamento do cronograma da Equipe ProcoBaja com o do desenvolvimento do trabalho, não possibilitando os testes de campo, em que no momento que foi necessário, o carro não estava completamente montado com todos os sensores e sem ter como andar.

Por fim, os resultados obtidos foram satisfatórios, os dados dos sensores lidos pelo PIC18F4520, como o de velocidade ficou muito próximo ao calculado. A comunicação dos rádios e foi estabelecida, permitindo a transmissão de dados contínua. A gravação dos dados no cartão microSD também foi realizada com sucesso, concluindo então que o sistema funciona.

Como sugestões para trabalhos futuros:

- Montar um painel para o motorista do veículo, que mostre todos os dados que forem lidos.
- Utilizar componentes eletrônicos do tipo SMD.
- Utilizar o módulo de rádio Digi Xbee-PRO 900HP.
- Adicionar um GPS a telemetria.

7 REFERÊNCIAS

ALBACORE. **Módulos ZigBee e RF.** Disponível em: <<http://www.albacore.com.br/index.php/produtoss/solucoesembarcaveis/modulos-zigbee-e-rf/modulos-rf-p>> Acesso em: 20 de Nov. 2016.

ALECRIM, Emerson. **Cartões SD e Micro SD: velocidades, tipos e capacidades.** Disponível em: <https://www.infowester.com/sd_microsd.php> Acesso em: 15 de out. 2016.

ARAÚJO, Marcelo Ferreira de. **Gerador de Sinais Microcontrolados.** 2013. 51 f. TCC (Graduação) - Curso de Tecnologia em Automação Industrial, Departamento de Eletrotécnica, Universidade Tecnológica Federal do Paraná - Ufpr, Cornélio Procópio, 2013.

BAU DA ELETRÔNICA. **Sensor de temperatura LM35.** Disponível em: <<http://www.baudaeletronica.com.br/sensor-de-temperatura-lm35.html>> Acesso em: 15 de out. 2016.

Bosch - **Manual de Tecnologia Automotiva** – 25ª Edição.

Catálogo MTE THOMSON de Injeção Eletrônica.

DANTAS, Diogo Holanda. **Sistema de Telemetria para Kart.** 2012. 133 f. Monografia (Especialização) - Curso de Engenharia de Computação, Uniceub, Brasília, 2012.

DATASHEET **PIC18F2420/2520/4420/4520** , Microchip Technology, 2009.

DATASHEET **PIC18F2455/2550/4455/4550** , Microchip Technology, 2009.

DATASHEET **SINGLE-CHIP USB TO UART BRIDGE**, Silicon Labs.

DATASHEET **XBee®/XBee-PRO® RF Modules**, Digi International Inc, 2009.

ELM CHAN. **FatFs – Generic Fat File System Module.** Disponível em: <http://elm-chan.org/fsw/ff/00index_e.html> Acesso em: 30 de out. 2016.

GOLDSMITH, Andrea. **Wireless communications.** EUA: Cambridge University Press, 2005.

IBRAHIM, D. **Advanced Pic Microcontroller Projects in C.** Edição 1ª. Editora Newnes, 2010.

Interfacing Catalex Micro SD Card Module with Arduino. Disponível em <
[http://www.vishnumaiea.in/projects/hardware/interfacing-catalex-micro-sd-card-
module](http://www.vishnumaiea.in/projects/hardware/interfacing-catalex-micro-sd-card-module) > Acesso em: 07 de mar. 2018.

KIOSKEA. **Cartões de memória (memória Flash)**. Junho, 2012.

MARTINS, Henrique. R; TORRES, Fernando. E. **Apostila Curso de Sistemas Microcontrolados**, baseado no PIC18F4550. [S.L:s.n], 2011.

MIKROC. MikroC PRO for PIC. 2015. Disponível em <
<https://www.mikroe.com/mikroc/> > Acesso em: 29 de set. 2017.

MORÃES, Martin José Fagonde. **Comunicação de Dados Por Rádio Frequência**. 2001. 107 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 2000.

NOVA ELETRÔNICA. **LM35 - O sensor de temperatura mais popular**. Disponível em:<
[http://blog.novaeletronica.com.br/lm35-o-sensor-de-temperatura-
mais-popular/](http://blog.novaeletronica.com.br/lm35-o-sensor-de-temperatura-mais-popular/) > Acesso em: 15 de out. 2016.

OPEN LAB. **Raw SD Read/Write Using PIC18F4550**. Disponível em:<
<https://openlabpro.com/guide/raw-sd-readwrite-using-pic-18f4550/> > Acesso em: 15 de fev. 2017.

OPEN LAB. **SPI Module in PIC18F4550**. Disponível em:<
<https://openlabpro.com/guide/spi-module-in-pic18f4550/> > Acesso em: 15 de fev. 2017.

OSHIRO, Fábio Kenji; KAGUE, Sérgio Mitsuo. **Giga de Testes para Sensores de Relutância Magnética**. 2012. 48 f. Monografia (Especialização) - Curso de Tecnologia em Eletrônica Automotiva, Fatec, Santo André, 2012.

QUEIRÓS, J. M. R. **Sistema de Sensorização e Telemetria de um VEC (Veículo Eléctrico de Competição)**. 2011. 117 f. Dissertação (Mestrado Integrado em Engenharia Eletrotécnica e de Computadores Major Automação) - Faculdade de Engenharia da Universidade do Porto. Cidade do Porto. 2011.

Regulamento SAE BRASIL 2009. Disponível em <
<http://www.saebrasil.org.br> > Acesso em: 02 de nov. 2017.

SAE BRASIL. **Baja SAE Brasil**. Disponível em: <
<http://portal.saebrasil.org.br/programas-estudantis/baja-sae-brasil> > Acesso em: 20 de out. 2

SAE BRASIL. **Programas estudantis, Baja 2015**. Disponível em:
<http://www.saebrasil.org.br/eventos/programas_estudantis/baja2015/oquee.aspx>
Acesso em: 20 de out. 2016.

Sensor de rotação indutivo. Disponível em <
<https://www.doutorie.com.br/blog/sensor-magnetico-indutivo-ou-de-relutancia-variavel-principio-de-funcionamento/>> Acesso em: 02 de mar. 2018.

SOBRINHO, André. S. F. **Sistemas Microcontrolados**, baseado no PIC18F4550, 2013.

SOUZA, D. J. **Desbravando o PIC**. Edição 8^a. Editora Érica, 2005

SOUZA, D. J. **Desbravando o PIC**. Edição 8^a. Editora Érica, 2005.

Tanenbaum, Andrew S. **Redes de Computadores**. 4^a.ed. Campus, 2003.

VEIGA, Gilberto Gil Dias; ANDRADE, Sérgio Filipe n. **Sensores no automóvel**: Injeção Eletrônica. 2008. 9 f. Monografia (Especialização) - Curso de Engenharia Eletrotécnica, Departamento de Engenharia Eletrotécnica, Instituto Superior de Engenharia do Porto, Porto, 2008.

APÊNDICE A - Código implementado no circuito transmissor.

```

#define buffer 32
// MMC module connections
sbit Mmc_Chip_Select      at LATC0_bit;
sbit Mmc_Chip_Select_Direction at TRISC0_bit;

unsigned int valor_ad = 0;
unsigned long tensao = 0, pulsos, ciclos, total, tot;
unsigned char mil,cem,dez,un, error;
unsigned int valor, sd_card=0, uart=0, count=0, x, j=0, sec=0, temp=0, timer=0;
unsigned char filename[] = "data.txt";
unsigned char temperatura[buffer];
unsigned char m_total = 0, c_total = 0, d_total = 0, u_total = 0, Kmh = 0, Kmh_disp=0;
unsigned char m1_total = 0, c1_total = 0, d1_total = 0, u1_total = 0 , m_Kmh=0, c_Kmh=0, d_Kmh=0 ,
u_Kmh=0;

void interrupt (void) {
  if(PIR1.CCP1IF)
  {
    PIR1.CCP1IF = 0;      // Limpa a flag para nova captura
    pulsos ++;
  }
  if (PIR2.CCP2IF){
    PIR2.CCP2IF = 0;
    ciclos++;
  }
  else if(INTCON.TMR0IF){
    PIE2.CCP2IE = 0; // Desabilita interrupção - periféricos CCP2
    PIE1.CCP1IE = 0; // Desabilita interrupção - periféricos CCP1
    TMR0H = 0x67;
    TMR0L = 0x6A;
    sd_card++;
    timer++;
    uart++;
    INTCON.TMR0IF = 0; //Zera flag timer 1
    PIE2.CCP2IE = 1; // Habilita interrupção - periféricos CCP2
    PIE1.CCP1IE = 1; // Habilita interrupção - periféricos CCP1
  }
}

void send_preamble(){

```

```

int cha;
for(cha=0;cha<5;cha++){
UART1_Write(0xF8);
}
UART1_Write(0xFA);
}
void start_byte(){
UART1_Write('(');
}
void stop_byte(){
UART1_Write(')');
}

void main() {
TRISC = 0b00000110;
TRISA.RA0 = 1; // AN0 como entrada
TRISB.RB0 = 0;
TRISB.RB1 = 0;

T3CON.T3CCP1=0;
T3CON.T3CCP2=0;
INTCON.GIE = 1; // Habilita chave geral das interrupções
INTCON.PEIE = 1; // Habilita Interrupção dos periféricos

PIE2.CCP2IE = 1;
PIE1.CCP1IE = 1; // Habilita Interrupção do módulo CCP
//T1CON = 0b10111001; // Liga TIMER1, prescaler 1:8, Clock interno
T0CON = 0b10000110; // Liga TIMER1, prescaler 1:8, Clock interno
CCP1CON = 0b00000101; // CCP1 em modulo CAPTURA e borda de SUBIDA
CCP2CON = 0b00000101;
//CCP1CON = 0b00000100; // CCP1 em modulo CAPTURA e borda de DESCIDA
PIR1.CCP1IF = 0; // Limpa a flag para nova captura
PIR2.CCP2IF = 0; // Limpa a flag para nova captura
INTCON.TMR0IE = 1; // Habilita a interrupção para o Timer 0
INTCON.TMR0IF = 0; // Zera a flag do Timer 0
INTCON2.TMR0IP = 1; // Habilita prioridade para o Timer 0

// T = Tcy * Prescaler * (65536 - x)
// 1s = 0,2*10^-6 * 128 * (65536 - x)

```

```

// x = 676A
TMR0H = 0x67;
TMR0L = 0x6A;

T0CON = 0b10010110; // Habilita Timer 0, 16 bits, borda de subida, prescaler de 128
INTCON.TMR0IE = 1; // Habilita a interrupção para o Timer 0
INTCON.TMR0IF = 0; // Zera a flag do Timer 0
INTCON2.TMR0IP = 1; // Habilita prioridade para o Timer 0

RCON.IPEN= 1; // Habilita prioridades de interrupção
INTCON.GIEH= 1; // Habilita interrupção de alta prioridade
INTCON.GIEL= 1; // Habilita interrupção de baixa prioridade

// Configuração do conversor A/D
CMCON = 0x07; // desabilita comparadores
ADCON0 = 0b00000011; // Seleciona AN0 e habilita A/D
ADCON1 = 0b00001110; //Habilita AN0 como analógico, Vref+ =Vcc e Vref-=Vss
ADCON2 = 0b10010101; //Alinhado a direita, TACQ = 4TAD TAD = FOSC/16
Delay_ms(100); // 0,1s

// Inicializando a comunicação serial UART

UART1_Init(1200);
Delay_ms(100); // Para estabilizar

SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV16, _SPI_DATA_SAMPLE_MIDDLE,
_SPI_CLK_IDLE_LOW, _SPI_LOW_2_HIGH);
Delay_us(10);
error = MMC_Fat_Init();
while(error == 1)
{
error = MMC_Fat_Init();
}
if(error==0){
PORTB.RB0 = ~PORTB.RB0;
MMC_Fat_Init();
SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV4, _SPI_DATA_SAMPLE_MIDDLE,
_SPI_CLK_IDLE_LOW, _SPI_LOW_2_HIGH);
Mmc_Fat_Assign(&filename,0x80);

```



```
    }

while(1){

tensao=ADC_Read(0);
tensao = (tensao*500/1023)*10; // Alimentando o pic em 5V
valor = tensao;
mil = valor/(1000);
valor = valor%1000;
cem = valor/100;
valor = valor%100;
dez = valor/10;
un = valor%10;

if (timer == 1)
    {
timer = 0;

total = pulsos *(60/3) ;
tot= ciclos * (60/8);

Kmh=(tot*1.627847*60)/1000;
Kmh_disp = Kmh;

m_Kmh = Kmh_disp/1000;
Kmh_disp = Kmh_disp%1000;
c_Kmh = Kmh_disp/100;
Kmh_disp = Kmh_disp%100;
d_Kmh = Kmh_disp/10;
u_Kmh = Kmh_disp%10;

m1_total = tot/1000;
tot = tot%1000;
c1_total = tot/100;
tot = tot%100;
d1_total = tot/10;
u1_total = tot%10;

m_total = total/1000;
```

```
total = total%1000;
c_total = total/100;
total = total%100;
d_total = total/10;
u_total = total%10;

ciclos=0;
pulsos = 0;
}
else if (timer > 1)
{
timer = 0;
}
if(uart==3){
uart =0;
while(UART1_Tx_Idle()==1){
send_preamble();
start_byte();

UART1_Write(mil + 48);
UART1_Write(cem + 48);
UART1_Write(dez + 48);
UART1_Write(44);
UART1_Write(un+48);
UART1_Write(d_kmh + 48);
UART1_Write(u_kmh + 48);
UART1_Write(m_total+48);
UART1_Write(c_total+48);
UART1_Write(d_total+48);
UART1_Write(u_total+48);
UART1_Write('\n');
UART1_Write('\r');

stop_byte();
send_preamble();}
Delay_ms(100);

}
else if(uart>3){
```

```
uart =0;
}
if(sd_card==10){
PORTB.RB1 = ~PORTB.RB1;
sd_card=0;

Temperatura[0] = 'T';
Temperatura[1] = 'E';
Temperatura[2] = 'M';
Temperatura[3] = 'P';
Temperatura[4] = '=';
Temperatura[5] = mil+48;
Temperatura[6] = cem+48;
Temperatura[7] = dez+48;
Temperatura[8] = ',';
Temperatura[9] = un+48;
Temperatura[10] = ' ';
Temperatura[11] = 'V';
Temperatura[12] = 'E';
Temperatura[13] = 'L';
Temperatura[14] = ':';
Temperatura[15] = d_kmh+48;
Temperatura[16] = u_kmh+48;
Temperatura[17] = ' ';
Temperatura[18] = 'k';
Temperatura[19] = 'm';
Temperatura[20] = '/';
Temperatura[21] = 'h';
Temperatura[22] = ' ';
Temperatura[23] = 'R';
Temperatura[24] = 'P';
Temperatura[25] = 'M';
Temperatura[26] = ':';
Temperatura[27] = ' ';
Temperatura[28] = m_total+48;
Temperatura[29] = c_total+48;
Temperatura[30] = d_total+48;
Temperatura[31] = u_total+48;
Temperatura[32] = '\r';
```

```
Temperatura[33]='\n';  
Mmc_Fat_Append();  
Mmc_Fat_Write(Temperatura,36);  
Delay_ms(100);
```

```
PORTB.RB1 = 0;  
}  
else if(sd_card>10){  
sd_card=0;  
}  
}  
}
```

APÊNDICE B – Código implementado no circuito receptor.

```

#define red_led RA0_bit
unsigned char ch[15], x;
int i, j;
void interrupt(void){
  if(UART1_Data_Ready()==1){
    x =UART1_Read();
    if(x=='('){
      j=1;
      i=0;
    }
    else if(j==1 && x==''){
      i=0;
      j=0;
    }
    else if(j==1){

      ch[i]= x;
      i++;
    }
  }
}
void main() {
  TRISA=0;
  TRISC.RC6 = 0;
  TRISC.RC7 = 1;
  ADCON1 = 0x0F;
  CMCON = 7;      // Desabilita os comparadores
  PIR1.RCIF= 0 ;//habilita interrupção
  PIE1.RCIE = 1 ;//zera flag
  RCON.IPEN= 0; // Interrupções sem prioridade
  INTCON.PEIE = 1;      // Habilita Interrupção dos periféricos
  INTCON.GIE = 1;      // Habilita chave geral das interrupções

  UART1_Init(1200);      // Inicializa UART
  delay_ms(10);
  while(1){

    if(i>0){

```

```
UART1_Write('T');
UART1_Write('E');
UART1_Write('M');
UART1_Write('P');
UART1_Write(':');
UART1_Write(ch[0]);
UART1_Write(ch[1]);
UART1_Write(ch[2]);
UART1_Write(ch[3]);
UART1_Write(ch[4]);
UART1_Write(' ');
UART1_Write('V');
UART1_Write('E');
UART1_Write('L');
UART1_Write(':');
UART1_Write(ch[5]);
UART1_Write(ch[6]);
UART1_Write('k');
UART1_Write('m');
UART1_Write('/');
UART1_Write('h');
UART1_Write(' ');
UART1_Write('R');
UART1_Write('P');
UART1_Write('M');
UART1_Write(':');
UART1_Write(ch[7]);
UART1_Write(ch[8]);
UART1_Write(ch[9]);
UART1_Write(ch[10]);
UART1_Write(' ');
UART1_Write('\n');
UART1_Write('\r');
i=0;
}

}}
```