

# Universidade de São Paulo

## Escola de Engenharia de São Carlos

Documentação Odômetro e Medidor de Velocidades

Prof.: Edson Gesualdo

Leonardo Campadelli	8606452
Raphael Kohn	8552011
Guilherme Simonetti de Arruda	8551879
Carlos Vinicius Fernandes	8551671
Vinicius Soares de Angelis	8532552
Denis Cosmo	8006821

<b>1. Definição do Problema</b>	<b>3</b>
<b>2. Justificativas da Solução Adotada</b>	<b>3</b>
<b>3. Metodologia</b>	<b>3</b>
3.1 Implementação do código	3
3.2 Construção de um sensor indutivo	4
3.2.1 Oscilador de Colpitts	7
3.2.2 Bomba de carga e Filtro Passa - Baixas	8
3.2.3 Circuito Amplificador e circuito comparador	8
<b>4. Resultados</b>	<b>10</b>
4.1 Código implementado	10
4.2 Sensor indutivo	11
<b>Anexos</b>	<b>15</b>

# 1. Definição do Problema

O intuito deste documento é poder apresentar melhorias para o sistema de medição de velocidades e poder implementar um odômetro, sistema responsável pela medição da distância percorrida pelo carro. Inicialmente, o sistema de medição de velocidade do BAJA não apresenta boa precisão na captura de dados visto que existem circunstâncias em que o carro sofre derrapagem devido ao percurso irregular. Já para o sistema do odômetro, a equipe ainda não possui esse tipo de medição. Logo, será acrescentado ao carro do BAJA.

## 2. Justificativas da Solução Adotada

Para reduzir os erros causados pela derrapagem do carro, foi escolhido obter a média de velocidade nas rodas do veículo. Para isso, será necessário instalar sensores nas rodas dianteiras do veículo podendo, assim, obter a média de velocidades para que o dado apresentado pelo velocímetro não seja tão distante do comportamento real. Com a melhoria do medição de velocidade, é possível acrescentar o sistema de odômetro ao carro já que a captação da informação da distância percorrida depende da velocidade medida pelo veículo.

## 3. Metodologia

### 3.1 Implementação do código

Para a implementação do medidor de velocidade e do odômetro, foi utilizado como base do projeto o seguinte algoritmo:

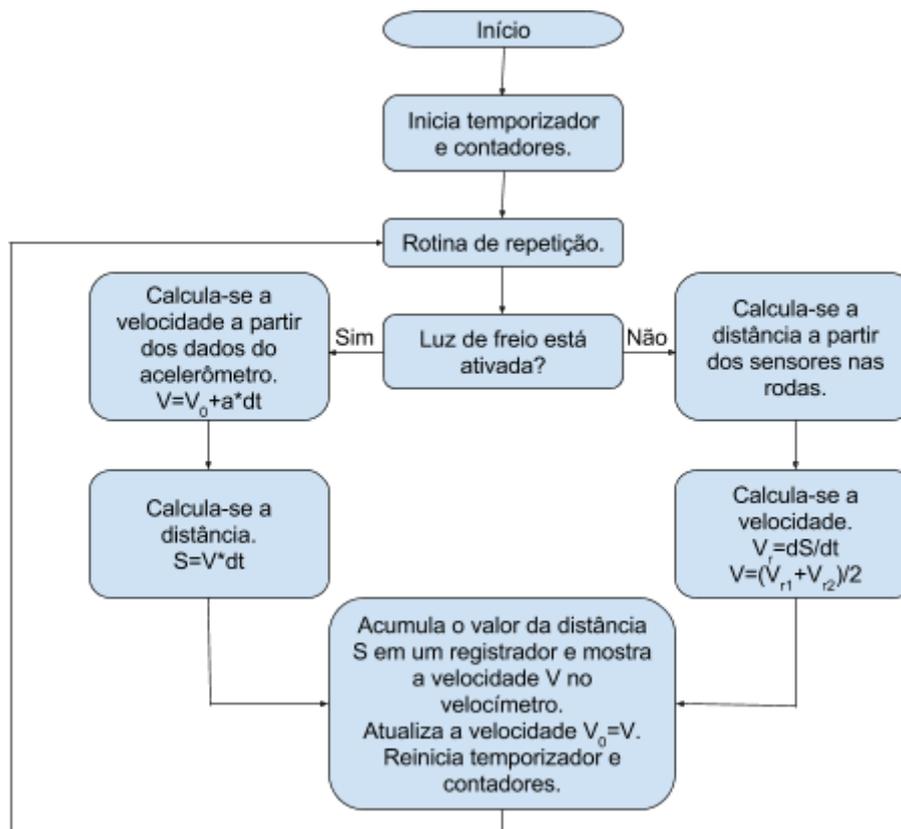


Figura 1: Algoritmo de implementação.

A partir do algoritmo da Figura 1, é possível utilizar a linguagem de programa que achar mais conveniente para implementação.

### 3.2 Construção de um sensor indutivo

Um sensor indutivo comercial pode ser construído a partir de circuitos básicos de instrumentação. Existem diversas técnicas aplicadas para sua construção, porém, todas elas apresentam o mesmo princípio de funcionamento e

os mesmos componentes básicos, ilustrados a seguir:

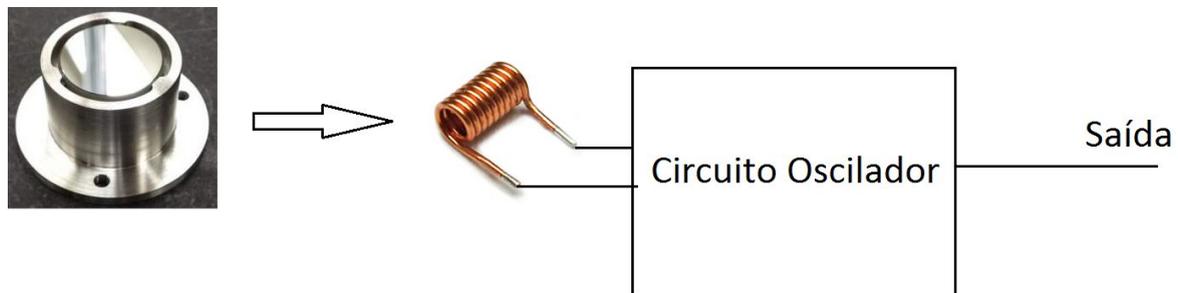


Figura 2: Componentes básicos de um sensor indutivo.

Na Figura 2, o oscilador é responsável por aplicar uma corrente alternada nos terminais da bobina, a qual gera um campo magnético (por apresentar propriedades indutivas). Com a aproximação do metal em direção a bobina, o campo magnético gerado induz a criação de correntes na superfície do metal (Correntes 'Eddy') que circulam em tal sentido de modo a diminuir o campo magnético total. A variação de campo magnético, por sua vez, pode ser percebida observando a mudança de tensão ou de frequência do sinal na saída do oscilador.

Primeiramente construiu-se a bobina com auxílio de um cilindro (17,5 mm de diâmetro e 48,0 mm de comprimento) e fio de cobre (0,5 mm de diâmetro). Enrolou-se o fio no cilindro até completarem-se aproximadamente 100 voltas, isso conferiu a bobina uma indutância de aproximadamente  $67\mu\text{H}$  e resistência série de 0,9 Ohm. A bobina pode ser vista na Figura 3.

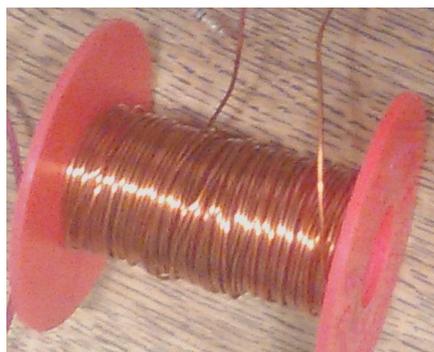


Figura 3: Bobina construída.

Optou-se por verificar a variação de frequência do oscilador. Para isso, utilizou-se a topologia de *Colpitts*. A Figura 4 mostra o diagrama de blocos do sensor implementado. À seguir, é descrito o funcionamento de cada bloco. Para o funcionamento do circuito, é necessário uma fonte simétrica de +9V/-9V no estágio de amplificação. Como solução para uma fonte desse tipo, propõe-se o circuito da figura 5, o qual utiliza um regulador de tensão de 9V 1,5A (LM7809) e o circuito integrado IC7660 para gerar as tensões simétricas a partir da bateria de 12V do carro.

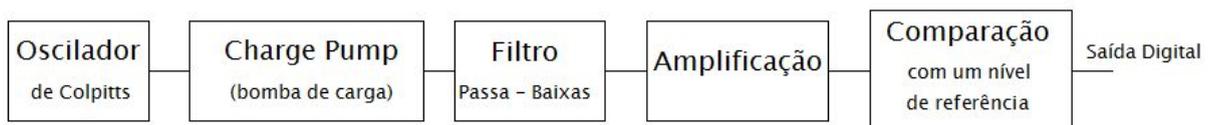


Figura 4: Diagrama de blocos do sensor indutivo implementado.

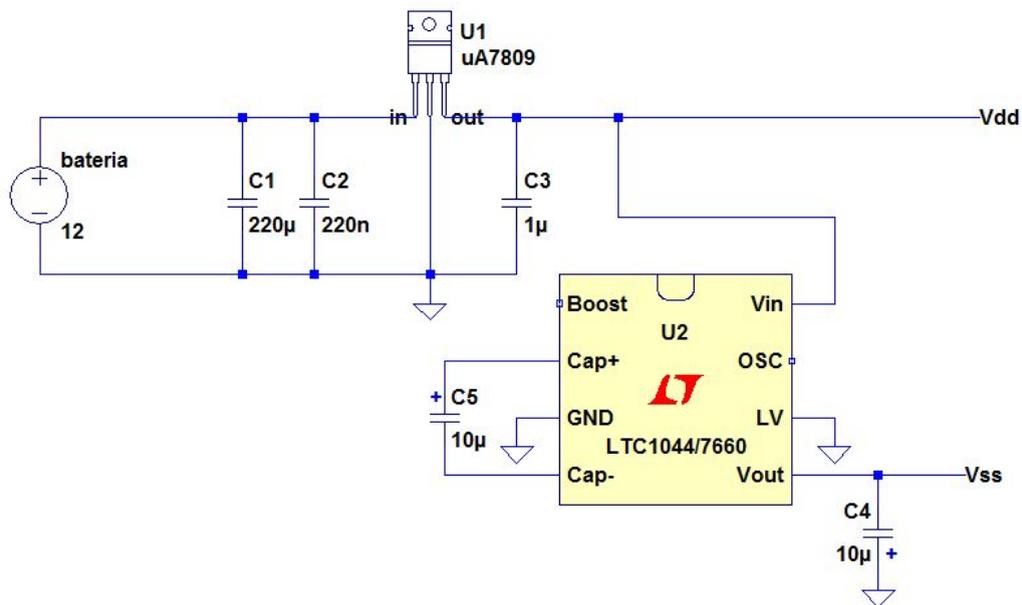


Figura 5: Circuito da fonte simétrica proposta. Figura produzida no software *LTSpice IV*.

### 3.2.1 Oscilador de *Colpitts*

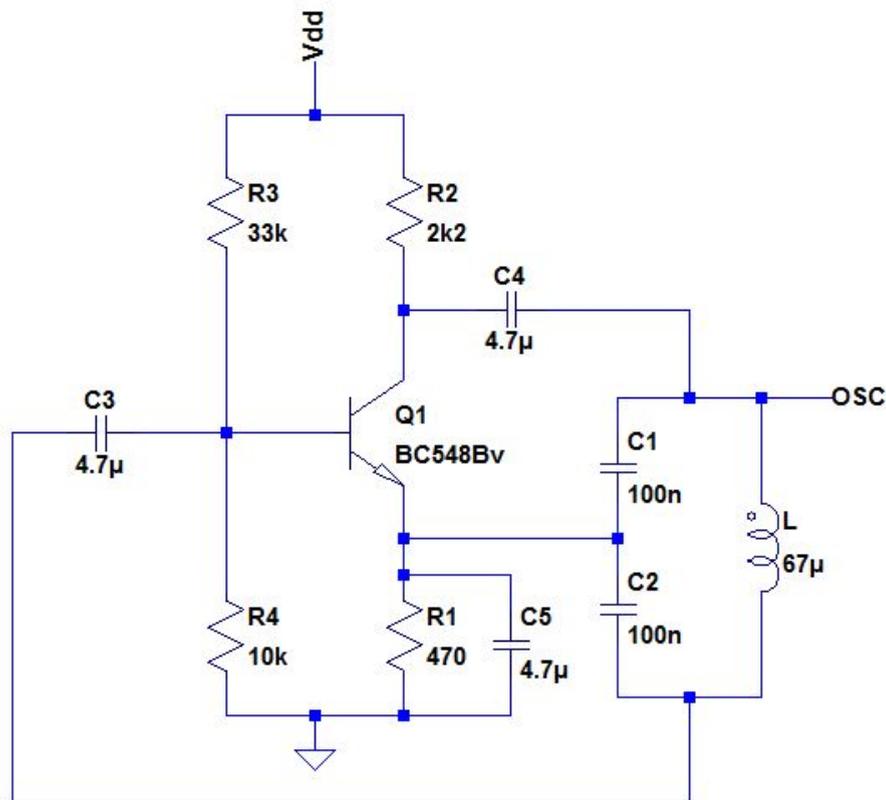


Figura 6: Oscilador de *Colpitts* implementado. Figura produzida no software *LTSpice IV*.

O oscilador de *Colpitts* (Figura 6) constitui em um circuito tanque composto pelos capacitores C1 e C2 e a bobina L. Para que as oscilações não se amortecessem, um amplificador emissor comum com o transistor bipolar BC548B é utilizado. Na presença de uma peça metálica, a indutância da bobina diminui, fato que pode ser verificado pelo aumento da frequência de oscilação. O decréscimo na indutância aumenta a corrente que flui na bobina e, desse modo, diminui a corrente que é injetada na bomba de carga (sinal de oscilação OSC).

### 3.2.2 Bomba de carga e Filtro Passa - Baixas

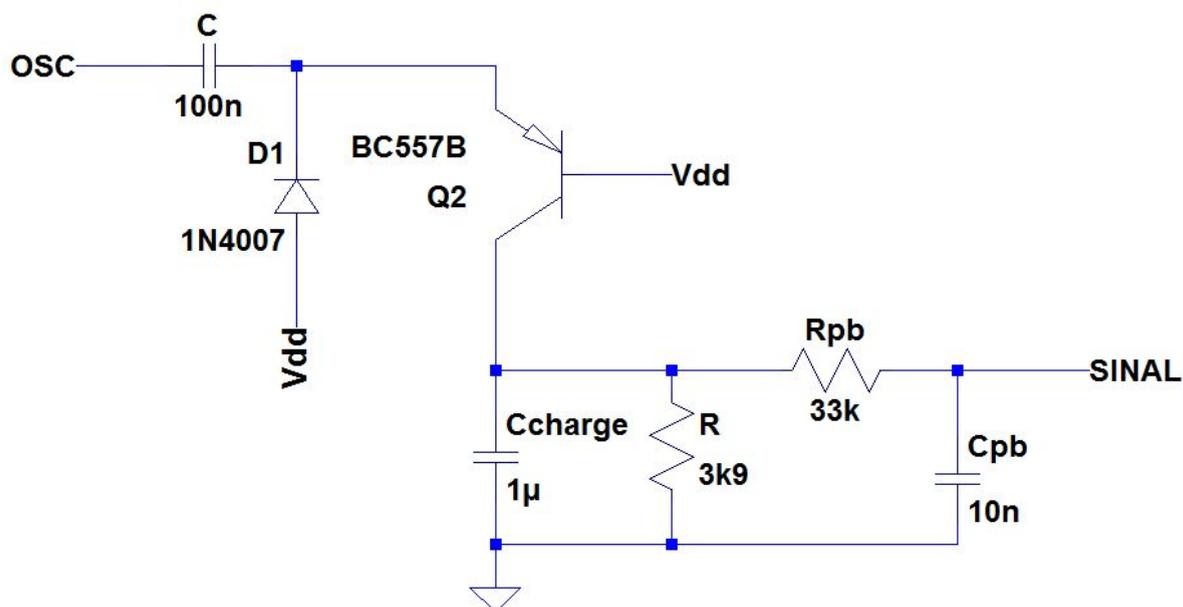


Figura 7: Bomba de Carga e Filtro passa-baixas implementado. Figura produzida no software LTSpice IV.

O princípio de funcionamento da Figura 7 é semelhante ao de uma bomba de carga (*Charge Pump*), com a peculiaridade de carregar o capacitor de carga ( $C_{charge}$ ) com o ciclo negativo da corrente de entrada  $OSC$ , devido à presença do diodo  $D1$ . A tensão final no capacitor de carga é proporcional à corrente do sinal de oscilação  $OSC$ , sendo menor, portanto, na presença de uma peça metálica. Na saída da bomba de carga, tem-se um filtro passa-baixas ( $R_{pb} - C_{pb}$ ) com frequência de corte de 481 Hz, responsável por atenuar as altas frequências da tensão no capacitor  $C_{charge}$ , mantendo um sinal bem regulado.

### 3.2.3 Circuito Amplificador e circuito comparador

Como o sinal de saída do filtro passa-baixas ( $SINAL$ ) é da ordem de mV, ele passa por um estágio de amplificação através de um amplificador não-inversor com ganho de 9,33 V/V, como mostrado na figura 8. Em seguida, o sinal amplificado ( $S\_AMPLIF$ ) é comparado com um sinal de referência  $V_{ref}$  (vide figura 8), o qual

pode ser variado através de um potenciômetro, de acordo com uma distância de detecção pré-estabelecida. Caso o sinal amplificado seja menor que  $V_{ref}$ , ou seja, presença de peça metálica a partir de uma certa distância, o sinal de saída  $OUT$  vai para 5V, acendendo o LED de indicação D3.

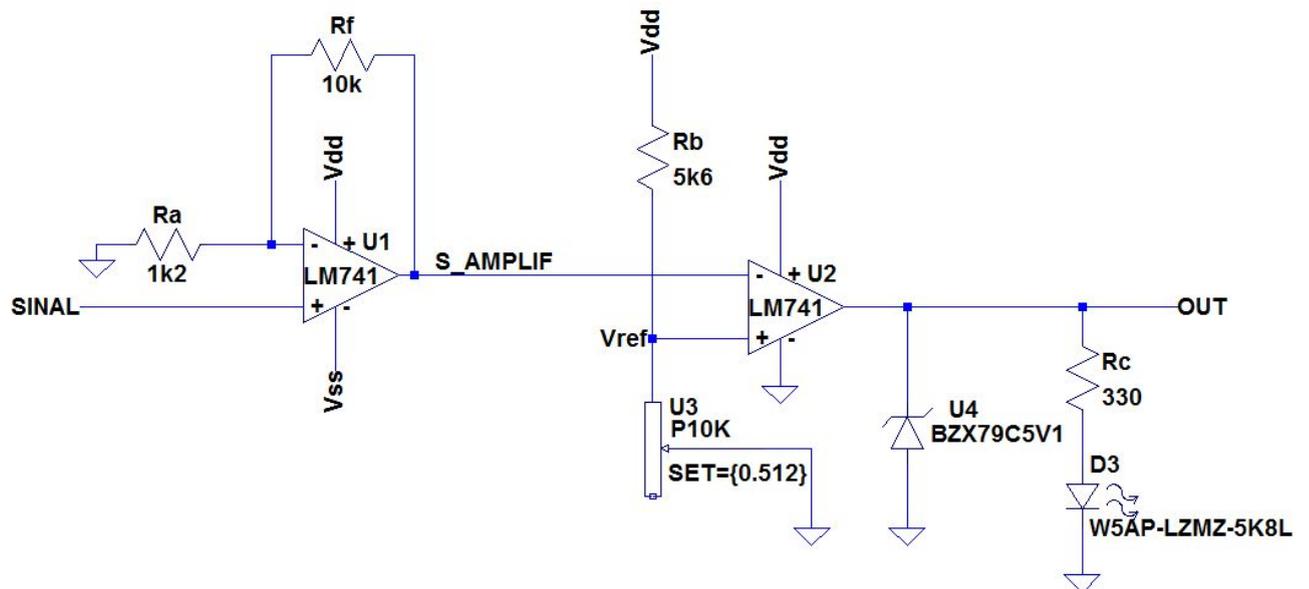


Figura 8: Circuito Amplificador não - inversor e circuito comparador implementados. Figura produzida no software *LTSpice IV*.

## 4. Resultados

### 4.1 Código implementado

Com o algoritmo da Figura 1, foi escolhido implementá-lo em linguagem C para controlador PIC obtendo o seguinte código. O código em anexo ao final deste documento.

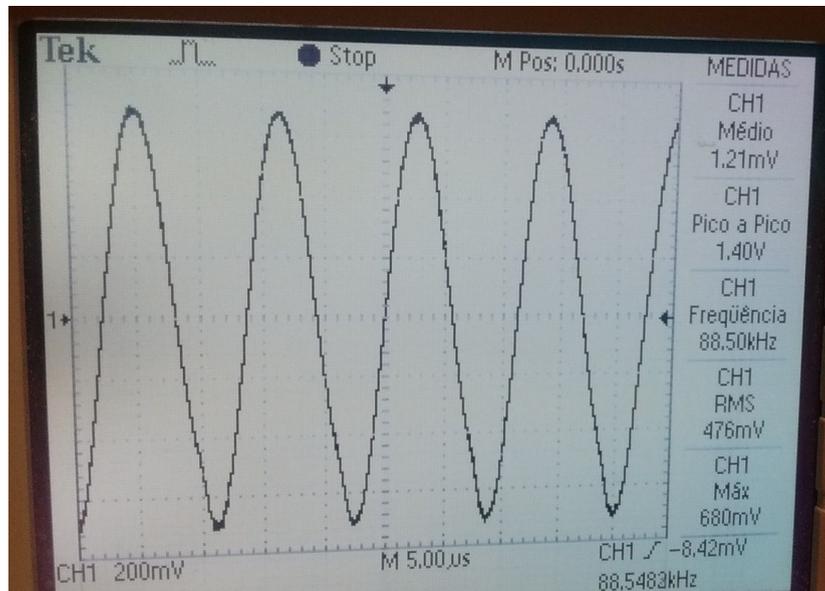
No código são feitas as seguintes considerações:

- É utilizado um dos timers do PIC para um contagem de 1ms, a proposta é que esse seja o mesmo 1ms utilizado pelo para atualização de dados proposto pelo grupo responsável pela implementação do GPS e do acelerômetro. É necessário adaptação do valor do timer dependendo do valor de clock utilizado pelo Baja.
- Considera-se que os dados do acelerômetro lidos já foram convertidos para  $m/s^2$  e que a janela de tempo é o pequena o bastante pra que a aceleração seja considerada à instantânea.
- Dois contadores são utilizados para a contagem de pulsos dos sensores indutivos, no código são eles os timers 1 e 3. Caso já estejam em uso pela equipe, contadores externos necessitam ser implementados.
- Não é especificado qual memória ROM é implementada para o odômetro (do próprio PIC ou externa), ou circuito de reset para a mesma.

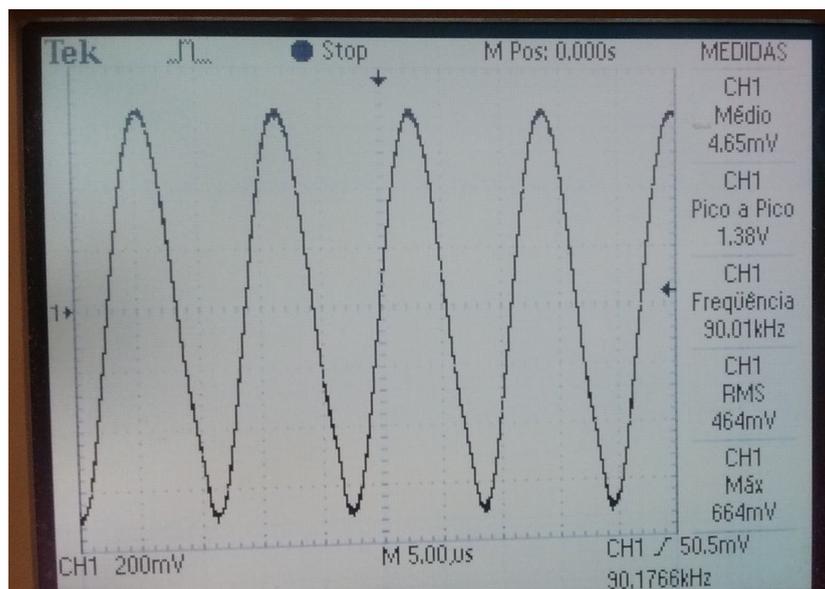
Considerações sobre a janela de tempo:

- Supondo raio da “roda” partindo do centro do eixo até a borda dos pneus, e que os pneus não se deformam consideravelmente, para um raio de 19,5cm é necessário que o valor mínimo da janela seja de 140ms, afim de registrar pulsos o suficientes para contar 1km/h. Portanto, é sugerido que o valor de janela fique entre 150 ms e 200 ms para o registro de velocidades baixas.
- Janelas adaptativas podem ser implementadas.

## 4.2 Sensor indutivo



(a)



(b)

Figura 9: Variação da frequência do oscilador com e sem a placa. (a) Sem a presença da placa, resultando em uma frequência gerada de 88,5 kHz e (b) com a placa o mais perto possível da bobina, resultando em uma frequência de 90,0 kHz. Fonte: Elaborada pelo autores.

Para realização dos testes, utilizou-se uma placa de metal majoritariamente composta de alumínio de largura 38,0 mm e altura de 72,0 mm, conforme a figura 12 (foto do lado direito). A variação da frequência do circuito oscilador perante tal placa pode ser ilustrada na figura 9.

Nota-se que a frequência variou aproximadamente 1,5 kHz com a presença da placa para a distância mínima (a placa estava praticamente encostada na bobina), ou seja, a excursão máxima de frequência é de aproximadamente 1,5 kHz. Ao se afastar um pouco mais a placa da bobina, a frequência de oscilação decaía, porém não linearmente. A saída amplificada da bomba de carga ( $S\_AMPLIF$  na figura 8) sem a presença de placa e, posteriormente, com a placa à aproximadamente 4,5mm de distância, é mostrada na figura 10.

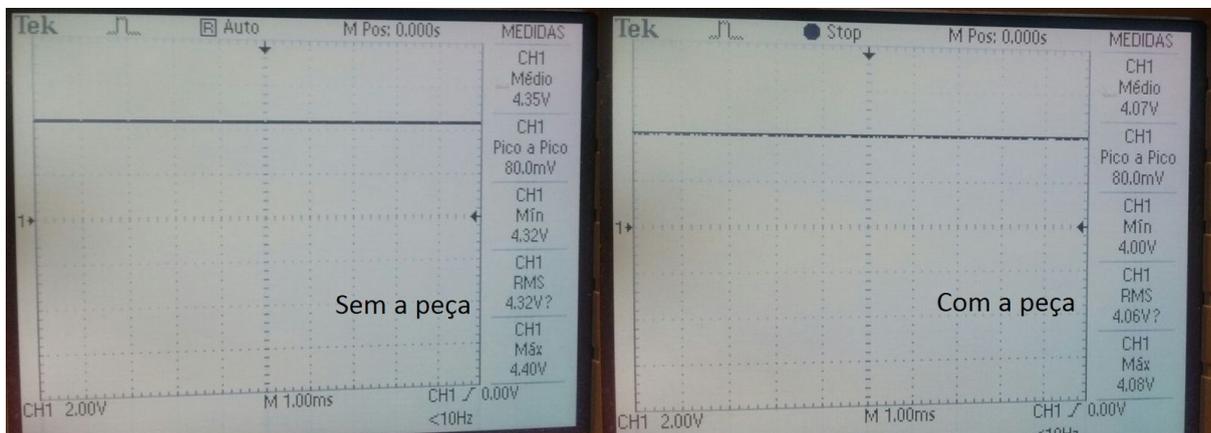


Figura 10: Saída amplificada da bomba de carga com e sem a presença da placa de teste. Fonte: Elaborada pelo autores.

Ao se aumentar ou diminuir a distância entre a bobina e a placa, notou-se que a tensão do sinal amplificado mudava, acompanhando a variação da distância. Assim, foi possível traçar uma curva que relacionasse a distância da peça de metal em relação a bobina com o valor de saída do conversor. Tal curva pode ser visualizada na figura 11.

Com o ajuste do potenciômetro, foi possível controlar a distância mínima para detecção do metal. Através do gráfico da figura 11, basta variar a resistência do

potenciômetro de modo que a tensão sobre o mesmo tenha o valor correspondente à distância desejada.

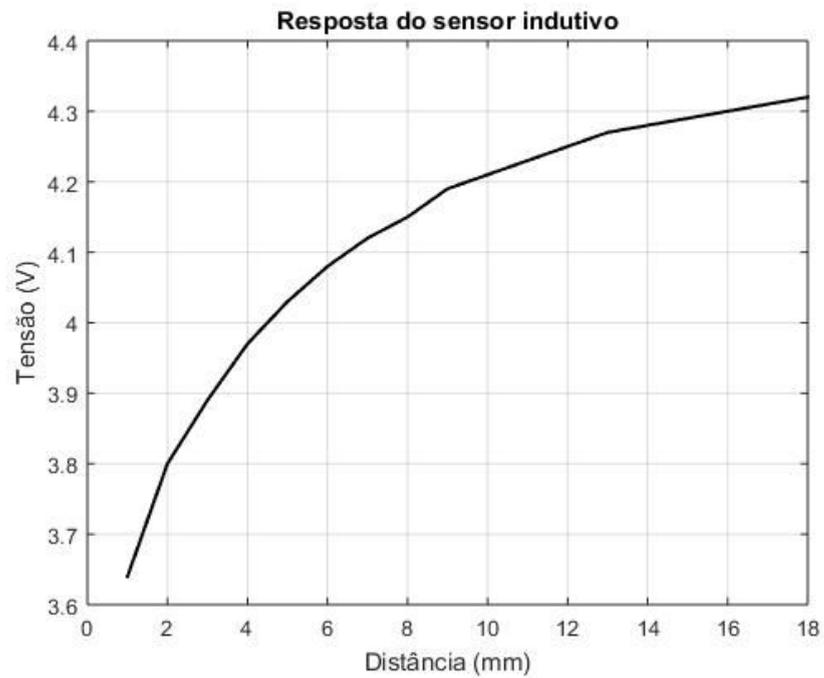


Figura 11: Tensão do sinal amplificado em relação a distância placa - bobina. Figura produzida no software *Matlab R2015a*.

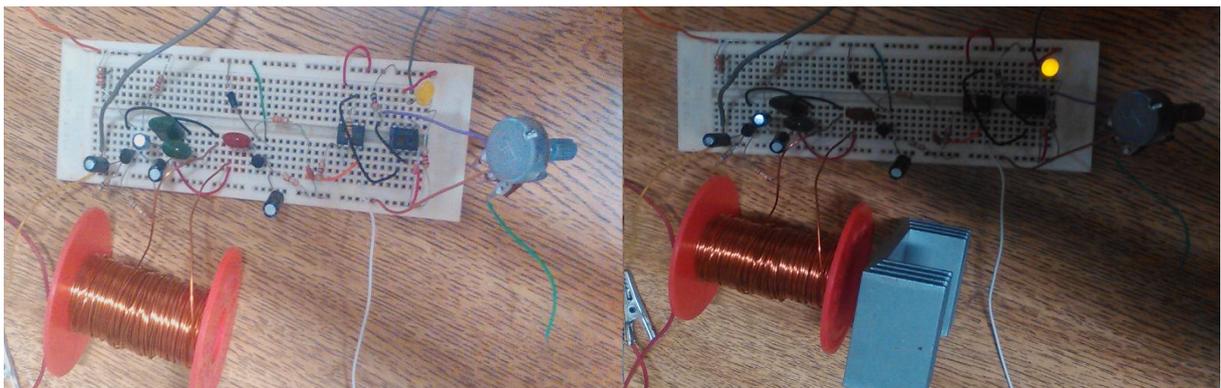


Figura 12: Circuito final para a presença de uma placa metálica e ausência da mesma.

Fonte: Elaborada pelo autores.

A saída do circuito completo, para as situações de presença e ausência da placa se ilustra na figura 12. O nível de referência foi ajustado em  $V_{ref} = 4,19 \text{ V}$ , o que corresponde no gráfico da figura 11 a uma distância de 9 mm aproximadamente.

Vale ressaltar que o sensor indutivo pode ser reduzido de tamanho. Para tanto, é necessário inserir um núcleo de ferrite. Com isso é possível obter uma indutância de 67  $\mu\text{H}$  com um cilindro de diâmetro menor e com menos voltas. Essa possibilidade acarreta uma vantagem sobre a implementada, já que com um número menor de voltas, a resistência da bobina é menor e, portanto, a corrente que alimenta a bomba de carga é maior, aumentando o nível de tensão após o filtro passa-baixas (*SINAL* na figura 7). Dessa forma, seria possível comparar tal sinal com o nível de referência diretamente, sem a necessidade de uma amplificação.

O circuito do sensor indutivo apresenta baixo custo, já que é composto por componentes eletrônicos e circuitos integrados básicos. O núcleo de ferrite compõe o componente de mais alto custo do projeto. Além disso, ele é versátil, pois pode ser utilizado em projetos que necessitem saber a distância de objetos metálicos, como por exemplo, para separá-las de acordo com a distância. Em outras palavras, o sensor construído pode ser utilizado como sensor de proximidade para materiais metálicos.

# Anexos

## código implementado

```
/**
//*****
// Considerações Teóricas:
// - Necessário criar uma janela de tempo pois ciclo de máquina muito mais rápido
//   que tempo entre pulsos
// - Dividir o valor do timer por 4 e olhar o resto em uma tabela
//*****

//*****
//Headers necessários
//Também é necessário a inclusão de header para funcionamento da eeprom e do
acelerometro*/

#include <pl8f452.h> // PIC a ser utilizado
#include <delays.h> // Adiciona a biblioteca do delay
#include <timers.h> // Adiciona a biblioteca dos timers
#include <stdio.h> // Adiciona a biblioteca padrão de entradas e saídas
#include <adc.h> //Adiciona a biblioteca de conversão analógico-digital
#include <usart.h> //Adiciona a biblioteca de comunicação serial
#include <capture.h> //Adiciona a biblioteca de capture

//*****

//*****
//Variáveis utilizadas
int x,y,z; // componentes de aceleração, se possível utilizar as mesmas do acelerometro
unsigned int janela = 50; //janela de tempo em milisegundos
unsigned float raio = 0.195; //raio roda - considera-se roda do centro até a borda do pneu
unsigned int dist_percorrida = 0; // endereço alocado a distância percorrida - odometro
unsigned int contVel = 0;

//Variáveis já declaradas no código original do baja
#define botao_frenagem PORTDbits.RD2
#define Fosc 20 // Define a frequência do oscilador :20Mhz

//*****

//*****
//Funções auxiliares

void INT_TIMER0(void)//Interrupção Timer a cada lms para incrementar variáveis de tempo
{
    contVel++;
    WriteTimer0 (0x1E84);
}

//*****

void calculo_velocidade()//Cálculo velocidades
{
    unsigned int velocidade1, velocidade2, rod1, rod2, velocidade_media, aceleração;

    if(contVel>=janela)
    {
        contVel = 0;
    }
}

```

```

if(botao_frenagem=1) //Freio ativado
{
    adxl_readAccel(&x, &y, &z); //funcao do acelerometro
    //aceleracao = x^2 + y^2; // se não estiver calibrado, caso contrário, utilizar
    //aceleracao = sqrt(aceleracao); // somente um eixo e não o módulo
    aceleracao = y; //Considerando o eixo y do acelerometro apontado para a frente do carro
    velocidade_media = velocidade_media + aceleracao*janela;
    WriteTimer1(0); //Resetando contadores dos sensores indutores
    WriteTimer3(0);
}

else //Freio nao ativado

{
    rodal= ReadTimer1();
    roda2= ReadTimer3();
    velocidade1 = 3.6*rodal*dist/(4*(janela*0.001));
    velocidade2 = 3.6*roda2*dist/(4*(janela*0.001));
    velocidade_media = (velocidade1 + velocidade2)/2;

    EEPROM_Write(dist_percorrida, 0.001*(rodal + roda2)*dist/2 + EEPROM_Read(dist_percorrida));
    // atualizacao do odometro ATENTAR-SE!!!!

    WriteTimer1(0); //Resetando contadores dos sensores indutores
    WriteTimer3(0);
}
}

}

//*****

//*****
//Programa principal
void main(void){

//configuraa do timer 0 (Janela de lms valor inicial do timer = 7812.5)
OpenTimer0
(
    TIMER_INT_ON & //Liga interrupcao do timer
    T0_16BIT & //Contador de 16bits
    T0_SOURCE_INT & //Fonte de clock interna
    T0_EDGE_RISE & //Clock de subida
    T0_PS_1_256 //Prescale
);

WriteTimer0 (0x1E84);

//configuracao dos timers que serao usados nos sensores
OpenTimer1
(
    TIMER_INT_OFF &
    T1_16BIT_RW &
    T1_SOURCE_EXT &
    T1_PS_1_1 &
    T1_OSC1EN_OFF &
    T1_SYNC_EXT_OFF
);

OpenTimer3
(
    TIMER_INT_OFF &
    T3_16BIT_RW &
    T3_SOURCE_EXT &
    T3_PS_1_1 &
    T3_OSC1EN_OFF &
    T3_SYNC_EXT_OFF
);

dist = (2*3.14*raio)/8 ; //distancia percorrida pela roda em uma volta/8,correspondente a 4 pulsos

while(1){

    calculo_velocidade();

};

}

//*****

```