

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

RODRIGO PORTO ANDRÉ

PROTÓTIPO DE UM CRONOTACÓGRAFO DIGITAL

CRICIÚMA

2017

RODRIGO PORTO ANDRÉ

PROTÓTIPO DE UM CRONOTACÓGRAFO DIGITAL

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel, no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof.Esp. Valter Blauth Junior

CRICIÚMA

2017

RODRIGO PORTO ANDRÉ

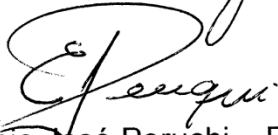
PROTÓTIPO DE UM CRONOTACÓGRAFO DIGITAL

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Microcontrolador.

Criciúma, 20 de Junho de 2017

BANCA EXAMINADORA


Prof. Valter Blauth Junior – Especialista – UNESC - Orientador


Prof. Ênio José Peruchi – Especialista - UNESC


Prof. Rogério Antonio Casagrande – Mestre - UNESC

RESUMO

O cronotacógrafo foi inventado por um engenheiro civil alemão, para o controle da condução dos trens. Após décadas se tornou o uso obrigatório em veículos de transporte de cargas e passageiros no Brasil, com o intuito de reduzir os números de acidentes. O dispositivo registra velocidades, distância e tempo de operação, servindo como uma caixa-preta dos veículos. Existem hoje cronotacógrafos mecânicos, eletrônicos e digitais que gravam as informações em discos diagrama ou fitas diagrama. Este trabalho possui o objetivo de construir um protótipo de um cronotacógrafo digital. No protótipo foi abordado o uso da tecnologia dos microcontroladores PIC e Atmel, comunicações I2C e Serial TTL. O protótipo construído registra velocidade, distância percorrida e tempo de condução de veículos em uma EEPROM. Foi construído um hardware que possui uma interface com o usuário. O protótipo possui um LCD e botões que possibilita cadastrar motorista, mostrar a máxima velocidade registrada, quilometragem total do veículo, data e hora e ainda contém uma impressora térmica que possibilita a impressão dos dados registrados das últimas 24 horas. Para o funcionamento dos microcontroladores foram desenvolvidos códigos utilizando a linguagem C.

Palavras-chave: Microcontrolador. Cronotacógrafo. Tacógrafo. Memória.

ABSTRACT

The *chromotachograph* was invented by a German civil engineer to help control the train conduction. Over the years, in Brazil, it became compulsory the utilization on load and passengers transportation vehicles, to help manage and reduce the number of accidents. The device works registering the speed, distance and operation time, like a "black box" would do. Nowadays, mechanic, electronic and digital *chromotachograph* can be found as well, which also are capable to record the information on diagram discs and diagram tapes. This project has, as main purpose, building a digital *chromotachograph*. In the prototype, was approached the use of PIC and Atmel microcontrollers technology, I2C communications and TTL. The built prototype records speed, traveled distance and driving time of the vehicles in an EEPROM. Was also developed a communication hardware with an interface to the user. The prototype has LCD display and buttons that allows to register the driver, shows maximum speed, total vehicle mileage, date and time and still contains a thermal printer, which allows the printing of data from the last 24 hours. For the microcontrollers operation, were developed codes using C language.

KEY WORDS - microcontrollers. *chromotachograph*. *tachograph*. memory.

LISTA DE ILUSTRAÇÕES

Figura 1 - Cronotacógrafo mecânico	15
Figura 2 - Cronotacógrafo eletrônico	15
Figura 3 - Cronotacógrafo digital	16
Figura 4 - Agulhas cronotacógrafo	16
Figura 5 - Ensaio.....	17
Figura 6 - Redutores angular	17
Figura 7 - Disco diagrama diário	18
Figura 8 - Disco diagrama	19
Figura 9 - Fita diagrama	20
Figura 10 - Sensor indutivo	20
Figura 11 - Sensor conversor.....	21
Figura 12 - Unidade Central de Processamento.....	23
Figura 13 - Símbolo ULA.....	23
Figura 14 - Comunicação I2C	31
Figura 15 - Transmissão	31
Figura 16 - Ligação entre um mestre e escravos no barramento SPI.....	32
Figura 17 - Roda fônica ou roda dentada	39
Figura 18 - Osciloscópio teste sensor	39
Figura 19 - Sensor indutivo	40
Figura 20 - Osciloscópio acionamento sensor.....	41
Figura 21 - PIC16f887.....	42
Figura 22 - Impressora térmica	43
Figura 23 - Esquema elétrico	44
Figura 24 - Esquema elétrico	46
Figura 25 - Esquema RTC	47
Figura 26 - Esquema entrada sensor	48
Figura 27 - Esquema fonte de alimentação.....	49
Figura 28 - Layout placa principal e fonte de alimentação.....	49
Figura 29 - Gravura das trilhas.....	50
Figura 30 - Placa corroída.....	51
Figura 31 - Protótipo	51
Figura 32 - Código leitura sensor	52
Figura 33 - Código gravar velocidade e deslocamento.....	53

Figura 34 - Código leitura DS1307	53
Figura 35 - Pacote de data e hora	54

LISTA DE TABELAS

Tabela 1 - Lista de materiais utilizados	45
Tabela 2 - Endereço 24c64	46

LISTA DE ABREVIATURAS E SIGLAS

CI	Circuito Integrado
CONTRAN	Conselho Nacional de Trânsito
CPU	Unidade Central de Processamento
CTB	Código de Trânsito Brasileiro
DENATRAN	Departamento Nacional de trânsito
EEPROM	<i>Electrically-Erasable Programmable Read-Only</i>
EPROM	<i>Erasable Programmable Read-Only</i>
I2C	<i>Inter Integrated Circuit</i>
INMETRO	Instituto Nacional De Metrologia Normalização
LCD	<i>Liquid Crystal Display</i>
MCU	Microcontrolador
RAM	<i>Random Access Memory</i>
RIIVT	Registrador Instantâneo e Inalterável de Velocidade Tempo
RISC	<i>Reduced Instruction Set Computer</i>
ROM	<i>Read Only Memory</i>
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SPI	<i>Serial Peripheral Interface</i>
PIC	<i>Peripheral Interface Controller</i>
PROM	Memória ROM Programável
UIC	Unidade de Instrução e Controle
ULA	Unidade Lógica e Aritmética
USART	<i>Universal SynchronousAsynchronousReceiverTransmitter</i>
RTC	Relógio de Tempo Real

SUMÁRIO

1 INTRODUÇÃO	10
1.1 OBJETIVO GERAL	11
1.2 OBJETIVO _s ESPECIFICOS	11
1.3 JUSTIFICATIVA	11
1.4 ESTRUTURA DA PESQUISA	12
2.1 CRONOTACÓGRAFO	13
2.1.1 Cronotacógrafo mecânico	16
2.1.2 Cronotacógrafo eletrônico	18
2.1.3 Cronotacógrafo digital	19
2.2 MICROCONTROLADOR	21
2.2.1 Unidade Central de Processamento	22
2.2.2 Sistema de clock	24
2.2.3 Unidade de Memória	24
2.2.3.1 Memória de programa	25
2.2.3.2 Memória <i>Random Acces Memory</i>	25
2.2.3.3 Memória <i>Electrically Erasable Programmble</i>	26
2.2.4 Unidades de entradas e saídas	26
2.2.5 Periféricos MCUs	27
2.2.5.1 Temporizadores	27
2.2.5.2 Interrupção	27
2.2.5.3 Conversor analógico-digital	27
2.2.5.4 Universal Asyanchronous Receiver Transmitter	27
2.3 COMUNICAÇÃO DE DADOS	28
2.3.1 Modos de transmissão	28
2.3.1.1 Transmissão paralela	28
2.3.1.2 Transmissão serial	29
2.3.2 Modos de Operação	29
2.3.3 Ritmos de transmissão	29
2.3.3.1 Transmissão assíncrona	29
2.3.3.2 Transmissão síncrona	30
2.3.4 Protocolo de comunicação	30
2.3.4.1 Protocolo de comunicação <i>Inter-Integrated Circuit</i>	30

2.4.4.2 Protocolo de comunicação SPI	31
2.4.4.3 Protocolo de comunicação RS232	32
2.5 SENSORES	33
3 TRABALHOS CORRELATOS	35
3.1 PROPOSTA DE UM SISTEMA ELETRÔNICO EMBARCADO PARA FISCALIZAÇÃO AUTOMÁTICA DE VEÍCULOS RODOVIÁRIOS DE CARGA	35
3.2 A REDE BRASILEIRA DE METROLOGIA LEGAL E SEU PAPEL NA VERIFICAÇÃO DE CRONOTACÓGRAFOS	36
3.3 FERRAMENTA PARA VERIFICAÇÃO DE JORNADA DE TRABALHO DE CONDUTOR DE VEÍCULOS PESADOS POR ANÁLISE DE TACÓGRAFO	37
4 DESENVOLVIMENTO DO CRONOTACÓGRAFO	38
4.1 METODOLOGIA	38
4.1.1 Seleção dos componentes	38
4.1.3 Construção do protótipo	44
4.1.3.1 Circuito principal	45
4.1.3.2 Circuito eletrônico fonte de alimentação	48
4.1.4.2 Confeção da placa de circuito impresso e montagem do protótipo	49
4.1.4.3 Desenvolvimento do software	51
4.1.4.5 Teste do protótipo	55
5. CONCLUSÃO	58
REFERÊNCIAS	59
APÊNDICES	61
APÊNDICE A -Código do microcontrolador Atmel	62
APÊNDICE B -Fita diagrama 0 km/h.....	67
APÊNDICE C - Fita diagrama com velocidade simulada	68
APÊNDICE D-Código PIC16F887.....	69
APÊNDICE E – Artigo final	85

1 INTRODUÇÃO

O Brasil é um país fortemente dependente do transporte rodoviário, seus veículos de carga e de transporte de passageiros desempenham um importante papel socioeconômico e uma posição estratégica na matriz de transportes do país.

O cronotacógrafo é um equipamento eficiente e confiável na obtenção dos dados de condução e do condutor e é um poderoso aliado das políticas de segurança de trânsito, pois proporciona um controle efetivo e prático sobre a operação dos veículos, auxilia o gerenciamento da operação no transporte de passageiros e cargas, podendo ainda, ser utilizado na identificação de possíveis causas de acidentes.

No Brasil desde 23 setembro de 1997 é obrigatório o uso de cronotacógrafo com o principal intuito de reduzir os números de acidentes. Este equipamento constitui-se de um único aparelho mecânico, eletrônico ou composto por um conjunto computadorizado. Os cronotacógrafos computadorizados podem ter duas interfaces para inserção de cartões, estes cartões armazenam dados de condução do veículo e dados do condutor (IMTT, 2010).

Com o avanço rápido na tecnologia da computação e eletrônica, surgiram algumas formas de construir o cronotacógrafo digital, com custo reduzido, com maior número de funcionalidades, maior segurança na leitura de dados e identificação e controle da jornada de trabalho do motorista. Sendo assim, este trabalho tem como propósito desenvolver um protótipo de cronotacógrafo digital, que atende às regulamentações do Instituto Nacional de Metrologia, Normalização e Qualidade Industrial (INMETRO). O dispositivo deve armazenar todos os dados monitorados do veículo (velocidade, distância percorrida e tempo) e dados do condutor do veículo, em uma memória não volátil, e equipamento será provido de uma impressora de fita diagrama destinado a disponibilizar a qualquer momento os relatórios dos dados monitorados pelo cronotacógrafo.

A utilização de microcontroladores trás vantagens, pois os circuitos ficam compactos, tem seu custo reduzido, permite processamento de sinais analógicos ou digitais, possibilita a alteração de funções do equipamento sem alteração de hardware e facilita a integração do circuito com computadores ou outros dispositivos (SOUZA, 2003).

No desenvolvimento do protótipo foi utilizado um microcontrolador para processar os dados, fazer leitura do sensor de rotação, comunicação com *Liquid Crystal Display* (LCD), memória e outro microcontrolador para comunicar com mini impressora térmica. Alguns modelos de microcontroladores possuem capacidades de ler tanto valores analógicos como digitais, e possui memória interna que possibilita o armazenamento dos dados.

1.1 OBJETIVO GERAL

Desenvolver um hardware microcontrolado, capaz de armazenar, de forma instantânea e inalterável os dados de distância percorrida, velocidade e quantidade de horas de direção e dados do condutor do veículo de carga ou de passageiros.

1.2 OBJETIVOS ESPECIFICOS

Os objetivos específicos desta pesquisa consistem em:

- a) entender o funcionamento dos cronotacógrafos e suas normas;
- b) realizar o estudo de microcontroladores;
- c) realizar o estudo de comunicação serial;
- d) compreender sensores;
- e) desenvolver um protótipo do cronotacógrafo digital.

1.3 JUSTIFICATIVA

Atualmente, o Brasil possui uma frota de mais de 4 milhões de veículos de carga e de transporte de passageiros com o uso obrigatório do cronotacógrafo (BRASIL, 2015).

Na Europa, a principal forma de controle da jornada de trabalho dos condutores de veículos pesados é através do cronotacógrafo, este dispositivo é montado por trás do velocímetro que registra velocidade, distância, tempo e modo de trabalho do veículo (ANDERSON, 1998, tradução nossa).

De acordo com o INMETRO (2015), a importância do uso do cronotacógrafo vem crescendo ao longo dos anos em consequência do número de

acidentes fatais, envolvendo ônibus e caminhões. O instrumento inibe os excessos e ajuda a reduzir acidentes.

Os microcontroladores oferecem uma variada combinação de projetos altamente eficientes e integrados, possuem arquiteturas otimizadas para baixo consumo de energia e suporte avançado para interfaces, os microcontroladores permitem aos desenvolvedores criarem soluções completas para o mercado industrial, consumidor, de segurança, de comunicações, computação e automotivo (ATMEL, 2015).

Sendo assim, justifica a pesquisa deste projeto, pois o protótipo cronotacógrafo digital desenvolvido com microcontrolador, tem um custo baixo, circuitos mais compactos, baixo consumo de energia, com maior durabilidade. Com este equipamento será possível fazer a gestão da frota de forma mais eficaz e segura, controlar pontos como: jornada de trabalho, desgaste do veículo (banguela, tempo de marcha), consumo de combustível (velocidade, distância percorrida, tempo) e uma maior segurança no trânsito.

1.4 ESTRUTURA DA PESQUISA

O trabalho é dividido em cinco capítulos, o primeiro trata da contextualização da pesquisa, apresentando a introdução, objetivo geral e os objetivos específicos, assim como a justificativa para a realização deste projeto.

O segundo capítulo aplica-se ao estudo em cronotacógrafo, microcontrolador, comunicação de dados e sensores que será utilizado no protótipo.

O terceiro capítulo detalha os trabalhos correlatos pesquisados para obter um melhor levantamento de informações sobre as pesquisas já realizadas na área de estudo.

O quarto capítulo apresenta cada etapa do desenvolvimento do protótipo cronotacógrafo, os testes realizados e os resultados obtidos.

O último capítulo é composto pela conclusão do projeto e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 CRONOTACÓGRAFO

O cronotacógrafo é um Registrador Instantâneo e Inalterável de Velocidade e Tempo (RIIVT), conhecido popularmente como tacógrafo, foi criado no século XIX, pelo alemão Max Maria Von Weber, sendo aplicado inicialmente para fiscalização dos condutores de trens (SOARES, 2015).

O cronotacógrafo apresenta o monitoramento contínuo do tempo de operação, distância percorrida e velocidades do veículo. Através desses dados é possível obter informações tais como: tempo de condução e descanso do motorista, paradas não programadas, limites de velocidade, horário de saída e chegada, sendo assim, este equipamento é de grande valia no deslinde das causas de acidentes de trânsito, e é um equipamento adequado para ajudar no gerenciamento das operações no transporte de passageiros ou de carga (PRADO, 2006).

Segundo o Conselho Nacional de Trânsito (CONTRAN, 1999), usando da competência que lhe confere o Inciso I, do Art.12, da Lei 9.503, de 23 setembro de 1997, que instituiu Código de Trânsito Brasileiro (CTB) e conforme o Decreto 2.327 de 23 setembro de 1997, que trata da coordenação do Sistema Nacional de Trânsito, considera que os veículos de transporte passageiros com mais de dez lugares e nos veículos de carga com peso bruto total superior a 4536 kg, para circular em vias públicas, os veículos deverão ser dotados do Registrador Instantâneo de Velocidade e Tempo.

Conforme o CONTRAN (1999), o RIITV deverá apresentar e disponibilizar a qualquer momento os dados das últimas vinte e quatro horas, pelo menos as velocidades desenvolvidas, distância percorrida pelo veículo, tempo de movimentação e suas interrupções, data e hora de início da operação, identificação do veículo, identificação dos condutores e identificação de abertura do compartimento que contém o disco ou emissão da fita diagrama. O RIIVT pode constituir-se de um único aparelho mecânico, eletrônico ou compor um conjunto computadorizado que, além das funções específicas, exerça outros controles. Para a aprovação pelo INMETRO ou entidade equivalentes o dispositivo e seu disco ou fita diagrama deverá conter os seguintes requisitos:

- a) possuir registrador próprio, em meio físico adequado, de espaço percorrido, velocidades desenvolvidas e tempo de operação do veículo, no período de vinte e quatro horas;
- b) assegurar a inviolabilidade e inalterabilidade do registro de informações;
- c) possuir lacre de proteção das ligações necessárias ao seu funcionamento e de acesso interno ao equipamento;
- d) dispor de indicação de violação;
- e) ser constituído de material compatível para o fim a que se destina;
- f) totalizar toda distância percorrida pelo veículo;
- g) ter os seus dispositivos indicadores iluminados adequadamente, com luz não ofuscante ao motorista;
- h) utilizar como padrão as seguintes unidades de medida e suas frações: quilômetro por hora (Km/h), para velocidade; hora (h) para tempo e quilômetro (km) para espaço percorrido;
- i) situar-se na faixa de tolerância máxima de erro nas indicações;
- j) possibilitar leitura fácil, direta e sem uso de instrumental próprio no local de fiscalização, nos dados registrados no meio físico (INMETRO, 2004).

Segundo o Departamento Nacional de Trânsito (DENATRAN, 2012), Resolução nº405, o tempo de jornada de trabalho do motorista profissional, na condução dos veículos de transporte de passageiros e de carga, pode ser fiscalizado pela análise do disco ou fita diagrama do RIIVT.

Os cronotacógrafos podem ser encontrados em três tecnologias distintas: mecânico, eletrônico e digital (PRADO, 2006).

Os cronotacógrafos mecânicos possuem um relógio e um ponteiro que identifica a velocidade do veículo, sua aparência lembra a de um velocímetro, na figura 1, um cronotacógrafo da marca VDO.

Figura 1 - Cronotacógrafo mecânico



Fonte: Taco1000 (2013).

O cronotacógrafo eletrônico é um equipamento que possui um compartimento, que permite o acesso ao disco de diagrama, com um display que mostra a hora, data e quilometragem do veículo, conforme figura 2.

Figura 2 - Cronotacógrafo eletrônico



Fonte: Taco1000 (2013).

O Cronotacógrafo digita da marca Seva é um dispositivo eletrônico, que possui um sistema de impressão incorporado, conforme figura 3.

Figura 3 - Cronotacógrafo digital



Fonte: Taco1000 (2013).

2.1.1 Cronotacógrafo mecânico

O cronotacógrafo mecânico tem seu funcionamento totalmente mecânico, o mesmo dispõe de um conjunto de três agulhas ou sondas metálicas, como dispositivo de impressão, o registro é por pressão das agulhas sobre o disco de diagrama, o disco dispõe de uma camada de cera, a sonda inferior registra a distância percorrida pelo veículo, a sonda média registra as atividades do condutor em cada momento e a sonda superior registra a velocidade do veículo em cada instante (PRADO, 2006), conforme figura 4.

Figura 4 - Agulhas cronotacógrafo



Fonte: VDO (2013).

Os Cronotacógrafos mecânicos possui um cabo tacógrafo que é acoplado na parte traseira do equipamento e na saída da caixa de câmbio, o equipamento possui um mecanismo que faz com que quanto maior a rotação do cabo, maior o deslocamento da agulha de registro de velocidade (PRADO, 2006).

No processo de calibração, o veículo é colocado sobre um equipamento com rolos que simula uma pista, conforme a figura 5, simulando um percurso de 2 km com uma velocidade média de 50km/h, com esse processo é possível identificar o valor da constante "W", a constante "W" é o número de voltas por km, quando a constante "W" é diferente da constante "K" do cronotacógrafo, existe a necessidade da instalação de um redutor angular, o redutor é um componente mecânico que é acoplado à caixa de câmbio, e torna compatível a constante W obtida na saída de caixa de câmbio com a constante K do cronotacógrafo, a figura 6 demonstra alguns modelos de redutores angulares (FERNANDEZ, 2010).

Figura 5 - Ensaio



Fonte: Sostacógrafos(2016).

Figura 6- Redutores angular



Fonte: Soseletrica(2015).

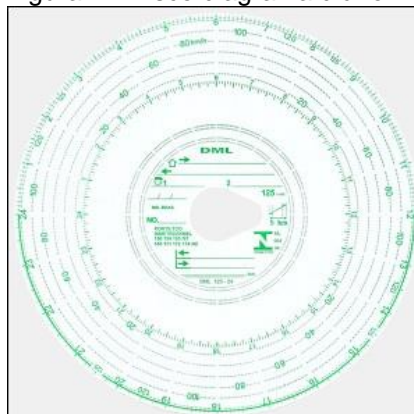
2.1.2 Cronotacógrafo eletrônico

O cronotacógrafo eletrônico tem como principal diferença em relação ao mecânico, a utilização de cabo elétrico, que liga o equipamento a um sensor instalado na saída da caixa de câmbio. O sensor acoplado na caixa de câmbio converte a rotação mecânica em pulsos elétricos, de acordo com a rotação da saída da caixa de câmbio, o sensor envia pulsos elétricos em uma frequência para o cronotacógrafo. Neste equipamento é dispensado o uso do redutor angular, o ajuste da constante "W" é efetuado no interior do cronotacógrafo eletrônico (DEMCZUK, 2010).

O cronotacógrafo mecânico ou eletrônico registra os dados através da pressão de três agulhas com pontas de safira, sobre um disco diagrama que remove a cera depositada na folha. Este disco de papel gira em velocidade constante, de tal forma que uma volta completa leva 24 horas, armazenando assim os dados. Alguns modelos de cronotacógrafo trabalham com disco diário que armazena dados de 24 horas e outros modelos trabalham com arranjo de sete discos, que armazenam dados no período de 168 horas, sem a necessidade de troca de disco (PRADO, 2006).

O disco diagrama possui uma escala de velocidade no formato circular com velocidade máxima de 125km/h, dividida e numerada a cada 20km/h, uma escala de tempo de formato circular com capacidade de registro de 24horas, com divisões de 5 minutos e outra escala de distância percorrida, também no formato circular composta de 4 linhas tracejadas, na figura 7, um disco diário (INMETRO,2015).

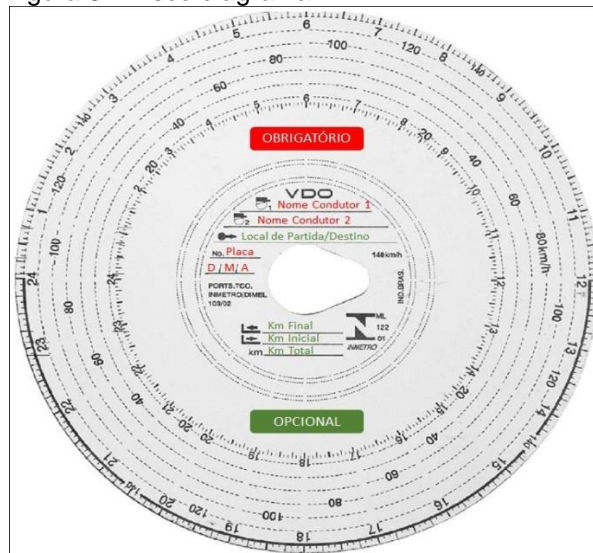
Figura 7 - Disco diagrama diário



Fonte: VDO (2013).

Segundo o DENATRAN (2016) o disco diagrama constitui-se de um item importante para o cronotacógrafo mecânico e eletrônico, pois é no disco que é armazenado de forma automática os dados de velocidade, tempo de condução e distância percorrida e outros dados devem ser preenchidos manualmente. O disco possui campos obrigatórios para preenchimentos, tais como: nome do motorista, identificação do veículo (placa), data de colocação do disco, além de outros dados de preenchimento não obrigatório, como: dados de destino de viagem, quilometragem final, quilometragem inicial, na figura 8, um exemplo de preenchimento.

Figura 8 - Disco diagrama



Fonte: VDO (2016).

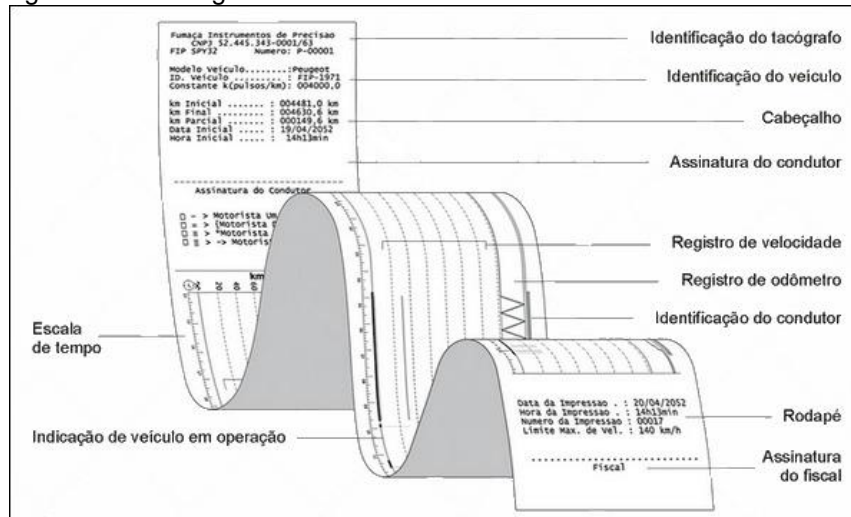
2.1.3 Cronotacógrafo digital

O cronotacógrafo digital ou computadorizado é semelhante ao eletrônico, com o diferencial de que os dados são armazenados em uma memória eletrônica não volátil e podem ser transferidos a outros formatos de mídia eletrônica. O mesmo não utiliza o disco diagrama como o cronotacógrafo mecânico e eletrônico, o RIITV digital tem um equipamento emissor de fita diagrama para disponibilizar as informações registradas (PRADO, 2006).

De acordo com Seva (2016), a fita diagrama é constituída de bobina de papel térmico, este papel é sensível à temperatura, a cabeça de impressão da impressora aquece o papel em postos específicos, e assim grava as informações permanentemente.

Ainda segundo Seva (2016), uma grande vantagem do cronotacógrafo digital, é poder imprimir as informações quantas vezes forem necessárias e a qualquer momento, sem preocupação e a inconveniência da troca diária ou semanal do disco de diagrama. Pode-se observar na figura 9, um exemplo de fita de diagrama.

Figura 9 - Fita diagrama



Fonte: FIP (2005).

O cronotacógrafo digital utiliza para medida de deslocamento do veículo, um sensor instalado na caixa de câmbio, que converte a rotação da caixa em pulsos elétricos e envia para o cronotacógrafo, que processa esses pulsos e transforma em dados de deslocamento. O sensor pode sair de fábrica instalado na caixa de câmbio ou ser adaptado no cabo mecânico do velocímetro. Na figura 10 tem-se um sensor indutivo acoplável na caixa de câmbio, na figura 11, tem-se um sensor conversor adaptável ao cabo do velocímetro (FERNANDEZ, 2010).

Figura 10 - Sensor indutivo



Fonte: VDO (2016).

Figura 11 - Sensor conversor



Fonte: VDO (2016).

A respeito da segurança das informações do cronotacógrafo digital, em caso de acidente com o veículo, o mesmo deverá conter as informações das últimas vinte e quatro horas. Os dados deverão permanecer intactos na memória do dispositivo, independente do fornecimento de energia elétrica, por um período mínimo de um ano, e seu relógio eletrônico interno deverá manter-se em funcionamento normal por um período maior a cinco anos. No processo de aferição para alteração do valor da constante “K” o equipamento deverá ter uma senha alfanumérica de pelo menos 8 caracteres (CONTRAN, 2003).

2.2 MICROCONTROLADOR

O microcontrolador é um componente eletrônico programável, com tamanho reduzido, e encapsulado em uma única pastilha de silício. Conhecido popularmente por CI ou CHIP, este componente é capaz de controlar processos lógicos, estes processos devem ser entendidos como controle de periféricos, tais como: botões, display, resistências, reles, sensores. Sua lógica de operação é estruturada na forma de um programa, é gravada no microcontrolador (SOUZA, 2003).

Os microcontroladores (MCU) também podem ser chamados de computadores embutidos, são indicados para controle específico e não tem grande capacidade de processamento. Seu custo varia de acordo de quantos bits possui, de que tipo é sua memória, capacidade, além de outros fatores. Os microcontroladores são encontrados em diversos aparelhos eletrônicos, como: eletrodomésticos,

brinquedos, aparelhos de reprodução de imagem, equipamentos médicos, sistema de armamentos militares e outros (TANENBAUM, 2007).

Há inúmeros fabricantes e modelos que podem ser encontrados no mercado, alguns deles: Família 8051(Intel ou Atmel), AVR (Atmel), 68hc196KB (Motorola), MSPs (Texas), *Peripheral Interface Controller* (PIC) (BATISTA,2012).

Os microcontroladores PIC possuem uma estrutura de máquina Harvard, que é caracterizada por ter dois barramentos internos, um barramento passa as instruções e o outro trafega dados, com isso é possível executar uma instrução por ciclo de *clock*, sua velocidade de processamento é maior que a arquitetura Von-Neumann, que possui um único barramento de dados, geralmente de 8 bits. A arquitetura Harvard necessita maior número de linhas de código para executar a mesma tarefa que em uma arquitetura Von-Neumann. Os PIC utilizam uma tecnologia chamada Computador com Set de Instruções Reduzidos (RISC) (SOUZA,2003).

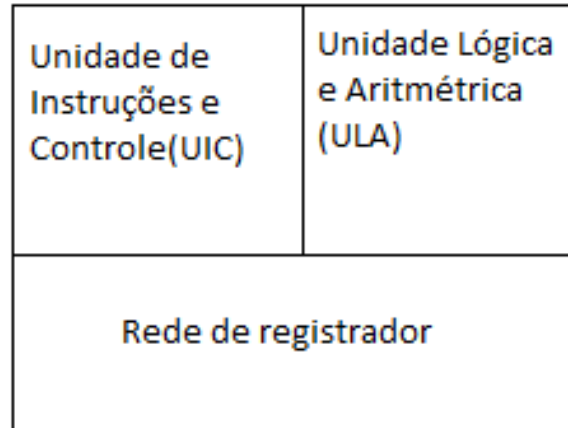
Na arquitetura Harvard no instante que uma instrução está sendo processada, a outra instrução está sendo buscada na memória de programa, isso só é possível por arquitetura possuir dois barramentos, o processamento é rápido. Os microcontroladores Harvard possuem CPU do tipo RISC, com seu *set* de instruções reduzido, aumenta seu poder de processamento, outra vantagem é o fato de possuir poucas instruções, o que facilita o desenvolvimento das aplicações (ZANCO, 2005).

Os MCU são um sistema computacional completo, no qual estão incluídos: Unidade Central de Processamento (CPU), Sistema de *clock*, Memória, unidades entradas e saídas, programa, além de outros possíveis periféricos (BATISTA, 2012).

2.2.1 Unidade Central de Processamento

A CPU é encarregada de executar e gerenciar todas as tarefas chamadas de instruções, para que seja possível o gerenciamento da CPU é preciso que a mesma se comunique com a memória e com os dispositivos de entrada e saída. A CPU é composta basicamente por três módulos: Unidade Lógica e Aritmética (ULA), Unidade de Instrução e Controle (UIC) e rede de registradores, como demonstrado na figura 12 (ZANCO, 2005).

Figura 12 - Unidade Central de Processamento

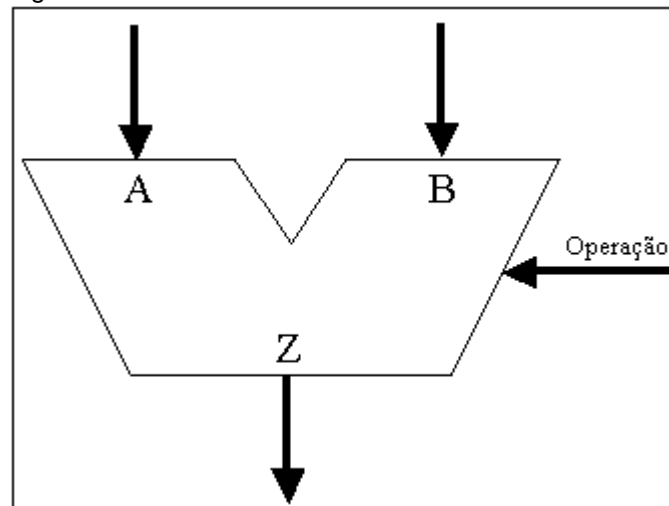


Fonte: Zanco (2005).

A ULA é a área do MCU onde é executada as operações lógicas e aritméticas. Algumas ULAs podem executar operações com sinal positivo ou negativo. Na figura 13, uma demonstração típica de uma ULA, onde "A" e "B" são operados e "R" é a saída (IBRAHIM, 2012, tradução nossa).

Segundo Souza (2003), quanto mais poderosa a ULA do componente, maior sua capacidade de processamento, sendo que todas as operações matemáticas e lógicas são executadas na ULA.

Figura 13 - Símbolo ULA



Fonte: Zanco (2005).

A UIC é responsável pelo controle das ações a serem realizadas no interior do microcontrolador, ela busca as instruções na unidade de memória, uma de cada vez, e as interpreta, e então gera os sinais apropriados para todas as outras unidades (TOCCI, 2007).

Redes de Registradores é uma pequena memória de alta velocidade usada para armazenar dados temporários, que serão manipulados pelo programa em algum momento (TANEMBAUM, 2007).

2.2.2 Sistema de clock

O Sistema de *clock* é um circuito eletrônico que gera um sinal de *clock*, que nada mais é do que uma onda quadrada. Este sinal é necessário para que a CPU possa usar como referência de tempo, para executar uma ou mais instruções por ciclo de *clock* (ZANCO, 2005).

Nos microcontroladores PIC, o *clock* pode ser obtido basicamente a partir de cinco configurações de oscilador:

- a) oscilador RC: são construídos com resistência e um capacitor, tornando-o um oscilador de custo baixo. São indicados para sistemas pouco sensíveis a frequência de operação, já que o mesmo trabalha com frequência menor que 4 MHz. A operação depende dos valores de R_{EXT} e C_{EXT} ;
- b) oscilador LP: são osciladores construídos com um cristal e dois capacitores cerâmica. Esta opção permite ao PIC utilizar um cristal de até 200kHz;
- c) oscilador XT: são osciladores construídos com um cristal e dois capacitores cerâmica, esta opção permite ao PIC utilizar um cristal de até 4MHz;
- d) oscilador HS: a configuração HS deve ser utilizada com cristais ou ressonadores de alta frequência. Geralmente *clock's* maiores que 8MHz utilizam esta configuração;
- e) oscilador externo: é utilizar um oscilador externo, neste caso, o sinal de *clock* é inserido no pino OSC1/CLKIN. O pino OSC2/CLKOUT deve ficar aberto (MICROCHIP, 2005).

2.2.3 Unidade de Memória

Unidade de memória é a parte do microcontrolador cuja função é de armazenar dados. A unidade de memória é organizada em espaços de memórias e

endereçamento, sendo que para acessar um determinado espaço de memória, precisamos selecionar o seu endereço. Nos MCUs existem três tipos de memória, conforme citado abaixo.

2.2.3.1 Memória de programa

Na MCU da família PIC, existem quatro diferentes tipos de tecnologia de memória de programa. A utilização de uma ou outra depende das características da aplicação a ser desenvolvida:

- a) *Read - Only Memory* (ROM) do tipo máscara: o microcontrolador já sai de fábrica com o programa gravado. O custo inicial de produção é elevado, pois a produção da máscara é elevada. Se comprados em grandes quantidades, na casa de milhares de microcontroladores, seu custo se torna viável;
- b) *Programmable Read – Only Memory* (PROM): este tipo de memória permite que o microcontrolador seja gravado apenas uma vez. São utilizados em sistemas que não exista a necessidade de atualização do programa;
- c) *Erasable Programmable Read Only Memory* (EPROM): é um componente encapsulado com uma janela de vidro. Essa memória pode ser regravada. Para apagar os programas gravados, requer exposição a luz ultravioleta;
- d) *Flash - ROM*: este componente permitem a gravação e regravação milhares de vezes por meio de sinais elétricos. Devido a facilidade de regravação do programa, é a opção mais indicada para teste de programas e implementação final dos projetos eletrônicos (ZANCO, 2005).

2.2.3.2 Memória *Random Acces Memory*

Random Acces Memory (RAM) são utilizadas para armazenar dados temporários que serão manipulados pelo programa e usados durante a operação do microcontrolado. Este tipo de memória é volátil, que significa que quando é

interrompido a alimentação de energia do microcontrolador, seus dados são apagados (MAZIDI, 2011).

2.2.3.3 Memória *Electrically Erasable Programmable*

A Memória *Electrically Erasable Programmable* (EEPROM) permite que seu conteúdo possa ser alterado durante a operação, e é do tipo não volátil: ou seja, mesmo na ausência de energia, seu conteúdo não se perde (MAZIDI, 2011, tradução nossa).

2.2.4 Unidades de entradas e saídas

Unidade de entradas e saídas são os meios que o microcontrolador utiliza para controlar dispositivos externos ou para receber pulsos, estes dispositivos tais como: teclados, display, reles, sensores entre outros (MARINHO, 2001, tradução nossa).

Nos microcontroladores, de maneira geral, a mais importante especificação dos pinos I/O é a corrente máxima que os mesmos podem suportar. A corrente de que cada pino suporta depende do modelo do microcontrolador. Geralmente esta corrente é de 5 a 25 mA, o que é suficiente para acionar um LED, um transistor ou outros dispositivos similares de baixa corrente. Outro ponto importante que deve ser observado é qual a corrente máxima que o microcontrolador suporta, para que sejam determinados quantos pinos simultâneos podem estar circulando corrente (SENA, 2005).

As portas são configuradas por meio de instruções, ou seja, a porta pode se comportar como entrada ou saída, da mesma forma, que os valores dos pinos podem estar em estado alto (1) ou baixo (0). Os registradores responsáveis pelos pinos de I/O nos microcontroladores PIC são: PORTx registrador de dados da porta, DDRx registrador de direção da porta e PINx registrador de entrada da porta (ZANCO, 2005).

2.2.5 Periféricos MCUs

2.2.5.1 Temporizadores

O temporizador ou contador é um registro, cujo valor, está continuamente a ser incrementado até o número 255. São utilizados para gerar pulsos de duração arbitrária, medição do tempo ou para contar pulsos externos. Nos PICs, conforme o seu modelo, o microcontrolador pode ter mais os menos temporizadores e são dados como: TMR0, TMR1 e TMR2 (MIYADAIRA, 2009).

2.2.5.2 Interrupção

As interrupções desempenham um papel importante em sistemas microcontrolados, elas fazem com que o microcontrolador responda a eventos externos ou internos. As fontes de interrupções podem ser a mudança de estado de um pino de entrada ou eventos internos, como exemplo: um estouro de Timer, onde o microcontrolador sai do fluxo normal de execução e passa para uma parte do programa denominada como: rotina de serviço de interrupção (MIYADAIRA, 2009).

2.2.5.3 Conversor analógico-digital

Os MCUs não compreendem sinais diferentes do digital (1 ou 0), por essa razão, é necessário converter o sinal analógico para o digital. Esta tarefa é executada por intermédio de um bloco destinado a conversão analógica-digital. O valor da precisão da conversão depende da resolução de conversão, que varia de acordo do modelo do microcontrolador (MIYADAIRA, 2009).

2.2.5.4 Universal Asynchronous Receiver Transmitter

A Universal Asynchronous Receiver Transmitter (USART), é uma interface de comunicação serial, que pode ser configurada para trabalhar no modo assíncrono ou síncrono. Na configuração assíncrona pode trabalhar em regime *full-duplex* e no modo síncrona, trabalha no modo *half-duplex*, ou seja, nos dois sentidos, mas um sentido de cada vez (MICROCHIP,2003).

2.3 COMUNICAÇÃO DE DADOS

Comunicação de dados é a troca de informação entre dois dispositivos através de alguma forma de meio de comunicação. Os dispositivos de comunicação devem ser parte de um sistema de comunicações feito da combinação de hardware e software (FOROUZAN, 2006).

De acordo Percilia (2015), a eficiência de um sistema de comunicação de dados depende de três características básicas:

- a) entrega: o sistema deve entregar os dados somente ao destino correto, apenas o dispositivo destino deve receber os dados;
- b) confiabilidade: o sistema deve garantir a entrega dos dados;
- c) tempo de atraso: os dados devem ser entregues em um tempo finito e predeterminado pelo sistema, sendo que deve ser o mais eficiente possível.

Conforme Alencar (2010), para um sistema de comunicação básico, é necessário no mínimo, cinco componentes, sendo eles:

- a) mensagem: é a informação a ser transmitida em um formato protocolado;
- b) transmissor: é o dispositivo que envia a mensagem. Exemplo: computador, um telefone, um microcontrolador, entre outros;
- c) receptor: é o dispositivo que recebe a mensagem, como: um computador, um telefone, um microcontrolador, entre outros;
- d) meio: é o caminho físico por onde passa a mensagem, do transmissor até o receptor;
- e) protocolo: é um conjunto de regras que padroniza a comunicação de dados.

2.3.1 Modos de transmissão

2.3.1.1 Transmissão paralela

Na transmissão paralela, o dispositivo pode enviar ou receber um conjunto de bits em um único ciclo de *clock*, em geral, esse conjunto de dados são

compostos de 8 bits compondo assim, um byte. A transmissão paralela requer "n" fios para transmitir o byte, por haver a interferência de linha cruzada entre fios, a distância é limitada (FOROUZAN, 2006).

2.3.1.2 Transmissão serial

Na transmissão serial, a transferência de dados ocorre bit por bit, através de uma única linha de dados. Teoricamente, a transmissão serial é mais lenta do que a paralela. As mesmas comportam maiores distâncias do que a paralela (FOROUZAN, 2006).

2.3.2 Modos de Operação

A transmissão pode ser classificada de três maneiras, de acordo com o fluxo de dados a serem trafegados (SILVEIRA,1991):

- a) *simplex*: a comunicação é unidirecional. Pode existir um único transmissor para vários receptores. Exemplo: transmissão do sinal de televisão e rádios;
- b) *duplex*: a comunicação ocorre nos dois sentidos, a transmissão é bidirecional, porém, não é simultânea, transmitindo em um sentido de cada vez. Exemplo: a comunicação entre radioamadores;
- c) *full-duplex*: a comunicação é possível em ambos os sentidos simultaneamente, compartilhando a capacidade do meio de transmissão. Exemplo: a comunicação entre celulares.

2.3.3 Ritmos de transmissão

2.3.3.1 Transmissão assíncrona

Em uma transmissão assíncrona, não há nenhum mecanismo de sincronização relativa ao emissor. Os bits do caractere são transmitidos em sequência, utilizando um elemento de sinalização para indicar o início do caractere e outro para indicar o término do caractere (SOUSA,1999).

2.3.3.2 Transmissão síncrona

Uma transmissão síncrona é quando há um mecanismo de sincronização, os bits são transmitidos em sequência, não existindo um elemento de sinalização para indicar o início e término do caractere (SILVEIRA, 1991).

Para manter o sincronismo, é necessário um sinal chamado de clock. Ele opera como um relógio entre o transmissor e o receptor, e determina o início e o fim de cada transmissão de bit 0 ou 1. Os dados são agrupados em blocos de tamanho constantes (SOUZA, 1999).

Segundo Forouzan (2006), a principal vantagem é a velocidade, sem a necessidade de bits de sinalização ou intervalos de repouso da linha de transmissão, a transferência de dados se torna mais rápida.

2.3.4 Protocolo de comunicação

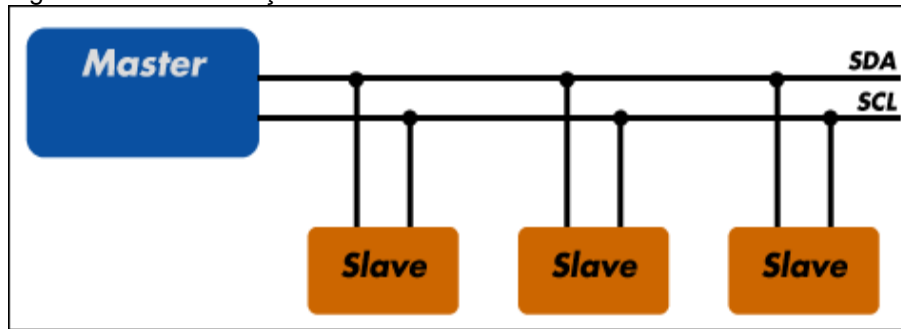
Protocolo é um conjunto de regras destinado a organizar a comunicação de dados, nelese define o que é comunicado, de que forma é comunicado e quando será a comunicação entre transmissor e receptor(FOROUZAN, 2006).

2.3.4.1 Protocolo de comunicação *Inter-Integrated Circuit*

O protocolo *Inter-Integrated Circuit* (I2C) desenvolvido pela Philips Semicondutores nos meados de 1996, é uma comunicação serial bidirecional síncrona, com taxas de transferências de até 400kb/s, ela permite a comunicação entre diversos dispositivos através de um barramento de duas vias: uma via corresponde ao relógio (*clock*) de comunicação chamada *Serial Clock Line* (SCL) e a outra via é responsável pela comunicação bidirecional de dados *Serial Data Line* (SDA) (ATMEL, 2015).

No protocolo I2C tem dois tipos de dispositivos: mestre (*master*) e os escravos (*slave*). Onde o *master* é o dispositivo que inicia a comunicação e podem existir vários dispositivos *slave*, como mostra a figura 14.

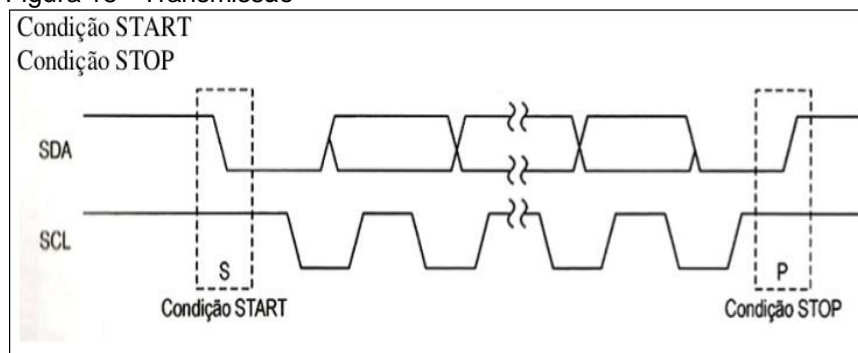
Figura 14 - Comunicação I2C



Fonte: Arduino (2013)

Para inicializar a comunicação, o dispositivo *master* trás a linha SDA para nível lógico baixo durante a fase alta da linha SCL, indicando aos dispositivos *slave* que uma nova transmissão de dados terá início. Para finalizar uma transmissão, o *master* trás a linha SDA de nível lógico baixo para alto, durante a fase alta da linha SCL. A figura 15 demonstra o início e a parada da transmissão

Figura 15 - Transmissão



Fonte: Arduino (2013).

Com a utilização do I2C, a principal vantagem é a economia de portas I/O do microcontrolador, pois, para seu funcionamento são necessários apenas dois pinos, tendo a possibilidade de conectar em um mesmo barramento, de modo geral, 128 dispositivos sem a necessidade de inclusão de um hardware adicional (LIMA, 2012).

2.4.4.2 Protocolo de comunicação SPI

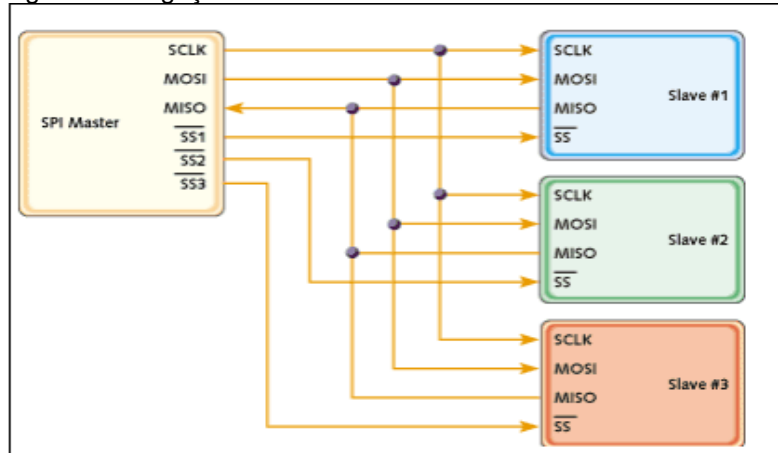
O protocolo de comunicação *Serial Peripheral Interface* (SPI) desenvolvido pela Motorola é um protocolo de comunicação série síncrono, usado por microcontroladores, para comunicar com um ou mais dispositivos periféricos em

curtas distâncias. O SPI funciona com um dispositivo mestre, que geralmente é um microcontrolador, que controla um ou mais dispositivos periféricos denominados escravos (ARDUINO, 2016, tradução nossa).

A comunicação entre o dispositivo mestre e os escravos mostrada na figura 16 contém: o mestre (*master*) e três escravos (*slave*) com quatro condutores:

- Master In Slave Out In (MISO)*: pino de dados do escravo para o mestre;
- Master Out Slave In (MOSI)*: pino de dados do mestre para o escravo;
- Serial Clock (SCK)*: *clock* de sincronização para transmissão de dados entre o mestre e o escravo;
- Slave Select (SS)*: seleciona qual o escravo receberá os dados (MICROCHIP, 2007).

Figura 16 - Ligação entre um mestre e escravos no barramento SPI



Fonte: Embarcados (2015).

A comunicação SPI tem como vantagem ser *full-duplex*, ou seja, não necessita de resistor de *pull-up*. Os dispositivos escravos não precisam de um endereço único, por outro lado, que requer mais pinos do mestre, que geralmente é um microcontrolador. Suporta apenas um dispositivo mestre e trabalha em curtas distâncias em relação a outras formas de comunicação (MICROCONTROLANDOS, 2013).

2.4.4.3 Protocolo de comunicação RS232

De acordo Canzian (2002), o padrão de comunicação RS232 foi criado em meados 1969, pela *Electronic Industries Association* (EIA) e surgiu para

padronizar a comunicação entre microcomputador e periféricos. Com o passar dos anos, sofreu algumas revisões e, a partir delas, foi formulada uma nova norma: a EIA RS232C, que definiu as características elétricas e mecânicas.

Ainda segundo Canzian (2002), transmissão de dados utilizado neste padrão é assíncrona, a transmissão ocorre um bit por vez com um ou mais bits de start e stop. A taxa de comunicação chamada de *baud-rate* informa quanto bits no período de 1 segundo serão transferidos na linha. As taxas de transmissão mais comuns são 2400,4800 e 9600 bps.

Na RS232C existem apenas dois níveis lógicos possíveis (0 ou 1), a norma prevê níveis de tensão entre 3 à 15 volts para nível lógico 0 e para nível lógico 1 o nível de tensão de -15 à -3 volts.

2.5 SENSORES

Os sensores são dispositivos sensíveis a alguma forma de energia, como: luminosa, térmica, cinética e tem a capacidade de transformar em sinal elétrico uma grandeza física a ser medida, como: temperatura, pressão, velocidade, corrente, aceleração, posição entre outras. O sinal de saída do sensor deve ser manipulado antes de sua leitura no sistema de controle (THOMAZINI, 2005).

Os sensores transformam as grandezas em sinais elétricos que podem ser sinais analógicos ou sinais digitais (FIALHO, 2010).

Sensores analógicos geram respostas elétricas que variam dentro de uma faixa de operação, e podem ter qualquer valor dentro dela. O sinal da saída mesmo limitada entre dois valores de tensão pode assumir infinitos valores de tensão intermediários (THOMAZINI, 2005).

Sensores digitais diferentes dos analógicos podem assumir apenas dois valores no seu sinal de saída ao longo do tempo, no entanto não existem naturalmente grandezas físicas que assumam apenas dois valores, no entanto com a utilização de um conversor analógico digital é possível ter apenas dois valores. Quando a conversão é realizada, parte do sinal é perdida e pode haver pequenas distorções na grandeza realmente medida (THOMAZINI, 2005).

2.5.1 Sensor Indutivo

Os sensores indutivos são dispositivos de estado sólido que detectam objetos metálicos sem contato direto. O sensor possui um oscilador conectado a uma bobina que forma um campo magnético, quando um metal é aproximado há uma interferência no campo magnético, essa variação do campo faz com que o circuito do sensor comute.

3 TRABALHOS CORRELATOS

Os trabalhos correlatos se fazem necessários para obtenção de conhecimento em tecnologias e metodologias utilizadas em outras experiências dentro da mesma área de pesquisa, tornando possível a comparação dos resultados obtidos.

São apresentados trabalhos que fazem parte das tecnologias estudadas e utilizadas na fundamentação teórica, metodologia e resultados obtidos nos projetos.

3.1 PROPOSTA DE UM SISTEMA ELETRÔNICO EMBARCADO PARA FISCALIZAÇÃO AUTOMÁTICA DE VEÍCULOS RODOVIÁRIOS DE CARGA

Este projeto é uma tese, para obtenção do título de doutor em engenharia mecânica, apresentada ao Departamento de Engenharia de São Carlos da Universidade de São Paulo no ano de 2010. A pesquisa teve como objetivo propor um sistema eletrônico embarcado que integre as funções previstas em lei de um registrador instantâneo de velocidade e tempo, e enviar os dados para autoridades competentes por um processo de comunicação sem fio (FERNANDEZ, 2010).

Para o desenvolvimento do sistema embarcado foi utilizada uma placa de processamento (*beaglebord*), esta placa trata de um projeto *open source* de um computador completo, o sistema operacional utilizado é o Linux e a distribuição adotada é Ubuntu, versão 10.04. Como periféricos foi utilizado um leitor de cartões, display, módulo GPS, adaptador de rede sem fios Realtek 8187b, todos os componentes com comunicação USB (FERNANDEZ, 2010).

O software residente no registrador foi desenvolvido em linguagem de programação Python e o banco de dados utilizado no armazenamento de informações é o SQLite3. O software foi dividido em três diferentes programas: programa de comunicação, registro e análise. Com os programas em execução, o registrador de velocidade e tempo, registra a posição geográfica do veículo e o seu instante preciso, uma vez por segundo, através de cálculos é possível determinar a sua velocidade, distância percorrida, tempo de operação e suas interrupções, data e hora do início e término da operação (FERNANDEZ, 2010).

Pode-se concluir que o protótipo funcionou perfeitamente, os testes realizados mostraram que não há necessidade de modificação nas características

físicas do veículo para uma comunicação sem fios com o padrão IEEE 802.11. Com uma velocidade do veículo de até 82 Km/h, a privacidade e autenticidade das informações enviadas pelo veículo foram garantidas através de algoritmos públicos de criptografia. O custo aproximado do equipamento ficou 52% do valor de um registrador instantâneo de velocidade e tempo comercial convencional (FERNANDEZ, 2010).

3.2 A REDE BRASILEIRA DE METROLOGIA LEGAL E SEU PAPEL NA VERIFICAÇÃO DE CRONOTACÓGRAFOS

O trabalho é uma monografia, apresentada como requisito para obter o título de Especialista em metrologia legal pela Universidade Federal do Paraná, em Curitiba, no ano de 2010. O trabalho tem como propósito estudar a importância da atuação da Rede Brasileira de Metrologia Legal e Qualidade quanto ao controle metrológico do cronotacógrafo, identificar os principais pontos de possíveis fraudes no sistema de funcionamento do cronotacógrafo e obter dados estatísticos referentes ao serviço realizado pelos postos de selagem e ensaios (DEMCZUK, 2010).

Este estudo foi desenvolvido em três etapas: na primeira foi realizado um levantamento estatístico das atividades realizadas em ensaio de cronotacógrafo, como: número de aprovações e reprovações dos serviços pelos órgãos metrológicos. O segundo momento consistiu em pesquisa de campo a um posto de selagem, onde foram coletadas informações pertinentes ao funcionamento e manutenção do cronotacógrafo, também foram obtidas informações referentes à forma com que os motoristas demonstravam conhecimentos em relação ao cronotacógrafo. Na última etapa do trabalho as informações obtidas foram organizadas e avaliadas (DEMCZUK, 2010).

Teve como resultados da pesquisa que fosse observado que a Rede Brasileira de Metrologia Legal e Qualidade do Inmetro possui papel importante no controle legal dos cronotacógrafos, tornando o equipamento confiável, tendo como consequência, um maior respeito quanto aos limites nas rodovias, aumentando assim a segurança no setor de transporte (DEMCZUK, 2010).

3.3 FERRAMENTA PARA VERIFICAÇÃO DE JORNADA DE TRABALHO DE CONDUTOR DE VEÍCULOS PESADOS POR ANÁLISE DE TACÓGRAFO

Esta pesquisa é um Trabalho de Conclusão de Curso submetido à Universidade do Planalto Catarinense para obtenção dos créditos de disciplina de Metodologia da Pesquisa, no curso de Sistema de Informação, em Lages/ SC, no ano de 2013. O trabalho teve como propósito, desenvolver uma ferramenta para leitura de disco de tacógrafo pra verificação da jornada de um condutor de veículos pesados, utilizando o processamento de imagem em dispositivos móveis com Android (HOFFMANN, 2013).

Para o desenvolvimento do aplicativo proposto foi utilizadas duas linguagens de programação, Java e C++ e uma biblioteca para o real processamento da imagem a OpenCV. A leitura do disco é realizada após manipulação da imagem, a imagem é capturada pela câmera e redesenhada, passando por um ajuste centralizado do disco e em seguida, a imagem com forma circular é transformada em um retângulo e passa por um filtro de *threshold* para posteriormente fazer a leitura do disco (HOFFMANN, 2013).

Pode se concluir que não existe literatura de disco de tacógrafo, poucas informações dos fabricantes dos discos, ainda que, com algumas dificuldades foi possível desenvolver uma ferramenta para fazer a leitura de um disco tacógrafo, utilizando o sistema Android com o auxílio das bibliotecas Mare e OpenCv, o que poderia ser melhorado a ergonomia do aplicativo(HOFFMANN,2013).

4 DESENVOLVIMENTO DO CRONOTACÓGRAFO

O protótipo do cronotacógrafo digital deve ser capaz de registrar a velocidade, distância e tempo de condução do veículo e que seja possível identificar o veículo e o condutor. O dispositivo possui um microcontrolador que interpreta o sinal do sensor, e armazena em uma memória não volátil, por um período de 24 horas, que posteriormente poderão ser impressas.

4.1 METODOLOGIA

Para o desenvolvimento deste projeto de pesquisa, inicialmente foi realizado um levantamento do material bibliográfico, por meio de pesquisa nas bases científicas e na biblioteca da UNESCO, após a primeira etapa de pesquisa e aprendizado, iniciou o desenvolvimento do cronotacógrafo digital. Primeiramente, fez-se a seleção dos componentes necessários para montagem do protótipo.

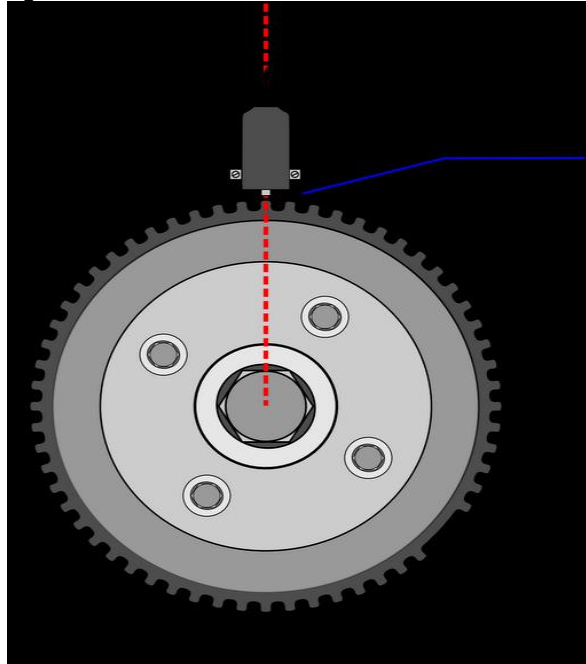
A etapa seguinte baseou-se na montagem do protótipo e na programação do microcontrolador, onde foram colocados em prática, os conceitos abordados durante o projeto de pesquisa.

Na última etapa do projeto foram realizados testes no protótipo e demonstrados os resultados obtidos.

4.1.1 Seleção dos componentes

Para a seleção dos componentes para a construção do protótipo, em primeiro momento, realizou-se uma análise do funcionamento de um veículo Mercedes Benz, chassi OH-1628, assim, determinou-se que seria necessário a utilização de um sensor, que é capaz de efetuar a contagem de pulso de uma roda fônica. O modelo do veículo analisado possui uma roda fônica interna na caixa de câmbio, que com a utilização de um sensor indutivo conta os dentes da roda fônica e envia pulsos elétricos para o cronotacógrafo. Na figura 17 tem-se a ilustração de uma roda fônica com um sensor.

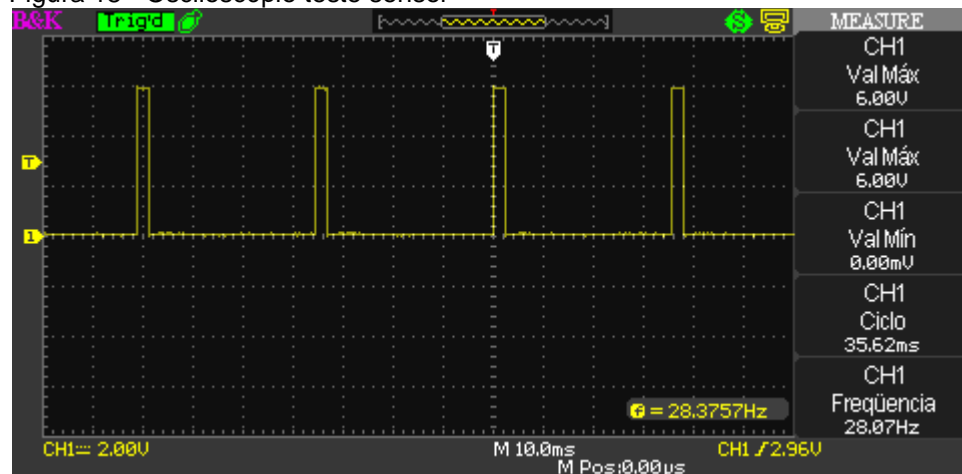
Figura 17 - Roda fônica ou roda dentada



Fonte: Dantas (2014).

Na análise do funcionamento do veículo, fez-se a leitura do sensor indutivo para determinar a quantidade de pulsos que o sensor envia para cada metro percorrido. A leitura dos pulsos foi realizada com o auxílio de um osciloscópio, coletando os dados referentes aos pulsos enviados pelo sensor. No primeiro teste com o veículo, com a velocidade de 20 km/h, observou-se como mostra a figura 18, que o sensor enviou pulsos de pico a pico de 6 volts, na frequência de 28.07 Hz. No segundo teste, com a velocidade de 30 km/h, o sensor envia pulsos de pico a pico de 5,68 volts, na frequência de 42.11 Hz.

Figura 18 - Osciloscópio teste sensor



Fonte: Do autor.

Com os dados obtidos no teste com o sensor conclui-se que com 50,5 pulsos, o veículo desloca 10 metros, e com uma velocidade de 150 Km/h, o sensor envia 211 pulsos por segundo. Essas informações são valores aproximados, por essa razão, o dispositivo cronotacógrafo deve possuir uma forma de calibração. Existem hoje, no Brasil, postos credenciados pelo INMETRO que fazem aferição e calibragem dos cronotacógrafos. A calibração de um equipamento é executada alterando-se o valor da constante “k”.

O sensor deve ser capaz de ser conectado a caixa de câmbio do veículo e enviar pulsos ao microcontrolador, o sensor utilizado foi um sensor indutivo, da marca VDO, modelo 216/9/1, conforme figura 19, o sensor possui quatro terminais sendo: dois pinos de alimentação e dois pinos de saída de pulso elétrico. Por ser um sensor indutivo, seu acionamento é sem contato físico, sua maior vantagem é a alta durabilidade, enquanto sua maior desvantagem é o fato de que nem todas as caixas de câmbio possui uma roda fônica, impossibilitando a utilização de um sensor indutivo.

Figura 19 - Sensor indutivo

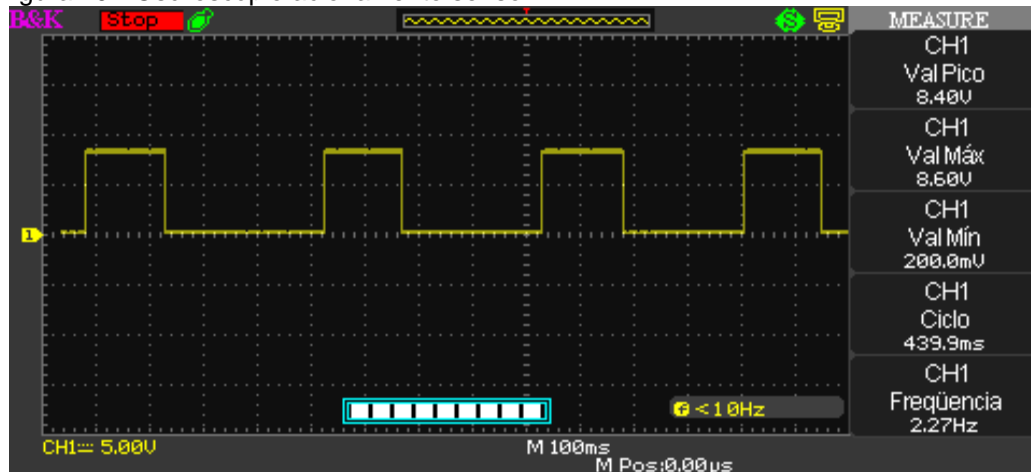


Fonte: VDO (2017).

Após o estudo do sensor, foi determinado a função de cada terminal, sendo: o pino 1 - GND, o pino 2 - VCC, o pino 3 - pulsos e o pino 4 - pulsos. O sensor deve ser alimentado, para maior confiabilidade, com uma alimentação de 8 volts. Após teste com o osciloscópio, como mostra a figura 20, observou-se que o pulso do pino 3 está normalmente em nível baixo (0 volts), e com o acionamento do sensor passa para nível alto (8 volts). Simulando o seu funcionamento com o auxílio de um metal, pode-se observar uma forma de onda quadrada, onde a largura da

onda é alterada pelo tempo que o metal fica próximo do sensor. No pino 4, por sua vez, está normalmente em nível alto (8 volts) e quando o sensor é acionado, seu nível passa para nível baixo (0 volts).

Figura 20 - Osciloscópio acionamento sensor



Fonte: Do autor.

O cronotacógrafo digital, segundo as normas do INMETRO, deve ter seu relógio independente do corte da alimentação do cronotacógrafo e, o seu relógio interno, tem que se manter em funcionamento por um período superior a 5 anos. Após pesquisa se optou pela utilização do DS1307, por ser de fácil aquisição e com grande quantidade de documentação do mesmo. O DS1307 é um componente de Relógio de Tempo Real (RTC) com calendário completo, sendo capaz de fornecer informações como: segundo, minutos, dias, semana, mês e ano. Ele possui uma interface de comunicação I2C, com um consumo menor de 500 nA no modo bateria. Segundo *Maxim Integrated*, com uma bateria de lítio com 48mAh ou superior, o DS1307 manterá seu relógio em funcionamento por um período superior a 5 anos.

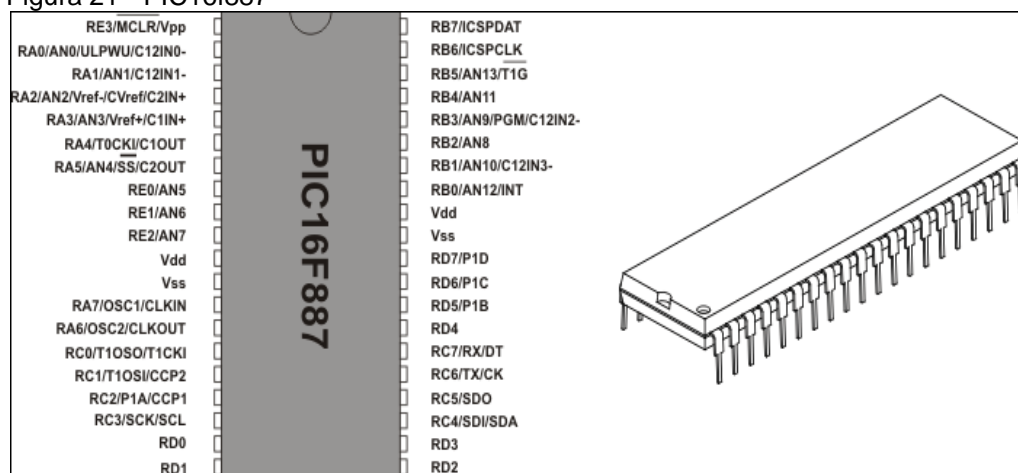
Outro componente necessário para o protótipo realizar o armazenamento dos dados de velocidade do veículo é a utilização de uma memória não volátil. Após o cálculo da quantidade necessária para o armazenamento dos dados do equipamento, chegou-se a conclusão que seria necessário um componente capaz de armazenar 2880 bytes. Sendo armazenado o valor da velocidade do veículo a cada um minuto, 2880 bytes serão capazes de armazenar 24 horas de informação de velocidade. Para essa finalidade foi utilizada a memória EEPROM 24c64 pelas seguintes razões: custo relativamente baixo, é um componente utilizado em vários equipamentos eletrônicos, é de fácil aquisição, possui capacidade de armazenamento de 64K bits, possui uma interface de comunicação I2C, com

retenção dos dados maior que 10 anos, e uma capacidade de 10000000 ciclos de escrita e leitura dos dados. Com isso, a vida útil do componente no protótipo é superior a 10 anos. Outra vantagem a ser destacada é que em um mesmo barramento de dados pode ser conectado 8 chips, e futuramente poderia comportar novas funcionalidade implementadas no cronotacógrafo digital.

Para a interface do protótipo foi empregado 4 micros interruptores e um LCD de 16 colunas por 2 linhas, com a finalidade de fazer a interação entre o usuário e o cronotacógrafo. O display opera com uma tensão de 5V, com comunicação paralela, possui o controlador hd44780, um dos mais utilizados na indústria de display LCDs. Sua principal vantagem é ser um display de baixo custo e alta eficiência.

O item escolhido para ser o responsável por fazer o processamento dos dados e fazer a comunicação dos componentes já citados, foi o microcontrolador PIC 16f887, por o acadêmico já possuir o mesmo. É um microcontrolador da família de 8 bits fabricado pela *Microchip Technology*, ele funciona com alimentação de 2v a 5,5v, possui 256 bytes de memória de dados EEPROM, com 5 conjuntos de portas de entrada e saída totalizando 33 portas. Na figura 21 tem-seoPIC16f887, o mesmo suporta trabalhar com temperaturas de - 40 a 125 C°, disponibiliza periféricos de comunicação UART,SPI e I2C, com dois temporizadores de 8 bits e um de 16 bits, em um microcontrolador que atende as necessidades do protótipo, por ter os periféricos UART para comunicação serial, o I2C para a comunicação da memória e RTC, o temporizador que será utilizado para receber os pulsos do sensor e com as portas digitais para fazer a comunicação com o display LCD e os botões.

Figura 21 - PIC16f887



Fonte: Microchip (2016).

O último componente escolhido para o cronotacógrafo foi a impressora térmica, pois as normas exigem que todo dispositivo cronotacógrafo esteja provido de um disco tacógrafo ou uma fita de diagrama. A impressora adquirida para o projeto é uma mini impressora térmica modelo JP-QR701, conforme figura 22. A escolha deste modelo foi por ter um tamanho compacto, operar com tensão de 5 V a 9 V, com um consumo de 1,5A, ter uma comunicação serial TTL 5 volts, possibilitando a ligação direta ao microcontrolador. Por ser uma impressora térmica, não necessita de tinta ou tonner. Ela utiliza bobina térmica com largura de 57 mm, a mesma utilizada em máquinas de cartão de crédito. Ela imprime com uma resolução de 8 pontos/mm e 384 pontos por linha.

Figura 22 - Impressora térmica



Fonte: Webtronico (2017).

Após a aquisição do equipamento, com alguns testes, foi obtido um melhor desempenho de funcionamento com uma alimentação de 8 Volts, com corrente de 1,5 A. Sua comunicação é serial TTL, com uma taxa de transferência 9600 bits por segundo, podendo ser conectada diretamente nos microcontroladores que possuem o periférico de comunicação UART. Com novos testes para impressão de pontos em determinadas posições na linha, não se obteve sucesso. Em uma pesquisa com o objetivo de encontrar alguma biblioteca para auxiliar a impressão de pontos na linha, que possibilitasse a impressão do gráfico de velocidade com o microcontrolador PIC16f887, não se obteve sucesso. Na pesquisa, foi encontrada uma biblioteca AdafruitThermalPrinter, que foi desenvolvida para o microcontrolador Atmel, utilizadas no Arduíno. Após estudos, se optou em utilizar o microcontrolador Atmega 328 para fazer a interface entre o PIC16f887 e a

impressora. O microcontrolador Atmega 328 recebe os dados a serem impressos, processa esses dados, e envia a impressora.

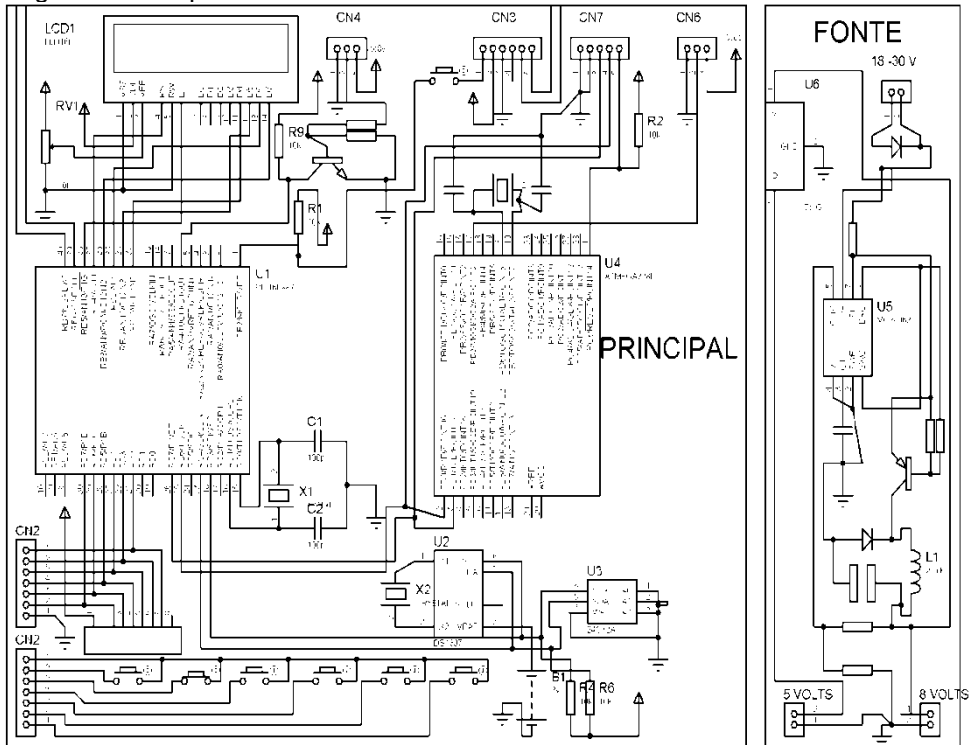
Outros componentes eletrônicos serão citados no decorrer do trabalho, tais como: resistores, cristais, transistores, trimpot, capacitores e reguladores de tensão.

4.1.3 Construção do protótipo

Para o desenvolvimento do protótipo foi utilizada uma ferramenta de simulação e construção de circuitos eletrônicos, chamada Proteus, na versão demo. No Proteus estão disponíveis vários componentes e ferramentas de laboratório, como: microcontroladores, LCD, resistores, capacitores, osciloscópio, geradores de sinais. Uma ferramenta que disponibiliza simulações de circuitos analógicos e digitais ou híbridos. O uso do Proteus foi essencial na execução dos primeiros testes funcionais para encontrar possíveis erros antes da montagem do protótipo.

Com o uso do Proteus, foi desenvolvido o esquema eletrônico do protótipo. O esquema eletrônico foi dividido em duas partes: fonte de alimentação e circuito principal, como pode-se observar na figura 23.

Figura 23 - Esquema elétrico



Fonte: Do autor.

4.1.3.1 Circuito principal

No circuito principal foram utilizados os materiais descritos na tabela 1, sendo descritos brevemente a funcionalidades de cada componente utilizado.

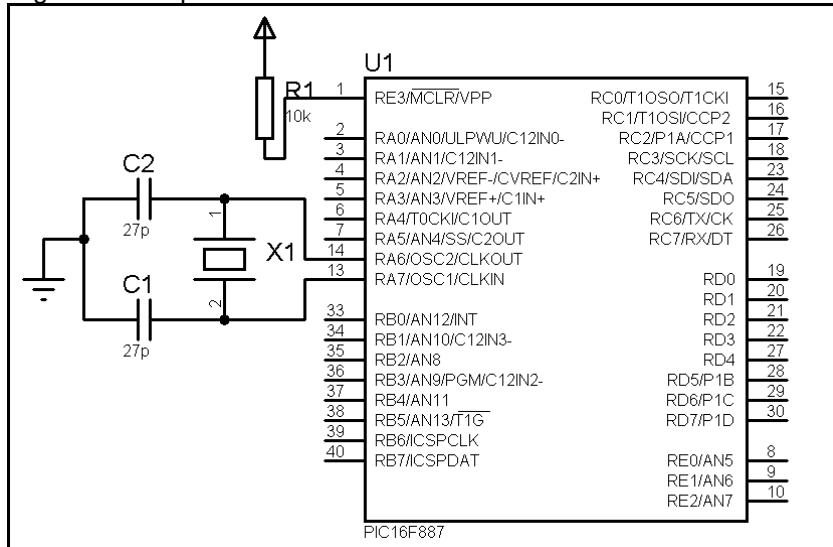
Tabela 1 - Lista de materiais utilizados

Quantidade	Descrição
1	Microcontrolador-PIC 16f887
1	Atmega 328
1	Cristal - 8MHz
1	Cristal - 16MHz
1	Cristal-32768KHz
2	Capacitores- 27pF
2	Capacitores- 22pF
2	Trimpot- 10K
11	Resistor-10K
1	Resistor-1M
6	Resistor-1K
4	Micro chaves
1	Display LCD 16x2
1	Mini impressora térmica
2	DS1307
1	Suporte bateria 2032
5	Conectores
2	Placa circuito impresso
2	BC547
1	MC34063
1	Indutor
2	Diodo 1n5819
1	LM7805

Fonte: Do autor.

O microcontrolador PIC 16f887 é um circuito microprocessador que possui entradas e saídas. Para seu funcionamento é indispensável um circuito oscilador. O oscilador escolhido para o protótipo possui um cristal de 8 Mhz e dois capacitores, formando o circuito oscilador, essa configuração de oscilador foi escolhida por ser mais eficiente e com baixo custo. O componente utilizado para formar o circuito *reset* do microcontrolador foi um resistor 10kohm, esse tem a função de manter em nível alto no pino 1 do microcontrolador, na figura 24 tem-se o circuito básico para o funcionamento do microcontrolador PIC16f887.

Figura 24- Esquema elétrico



Fonte: Do autor.

O LCD é um componente com 16 terminais, sua alimentação é de 5 volts, pino 1 GND, 2 VDD, 3 é o ajuste do contraste do LCD que foi conectado um trimpot de 10kohm, responsável pelo ajuste do contraste do display. Os pinos 15 e 16 são responsáveis por alimentar a luz de fundo que está conectada ao conector CN1, através do resistor que limita a corrente dos leds do display. Para a comunicação com o microcontrolador com o LCD, foi utilizado 6 terminais, operando com uma comunicação paralela de 4 bits.

No protótipo do cronotacógrafo, para o armazenamento dos dados de condução do veículo, velocidade e distância percorrida, foi utilizado uma memória 24c64, na placa foi colocado 3 soquetes possibilitando colocar 3 memórias. Essas memórias estão conectadas ao microcontrolador nos pinos SCL e SDA, utilizando o protocolo de comunicação serial I2C, cada memória possui um endereço físico composto de 7 bits sendo que: 4 bits já estão pré-configurados no componente e, os outros 3 bits foram configurados fisicamente nos componentes, através de 3 terminais, a tabela 24 apresenta a configuração das memórias.

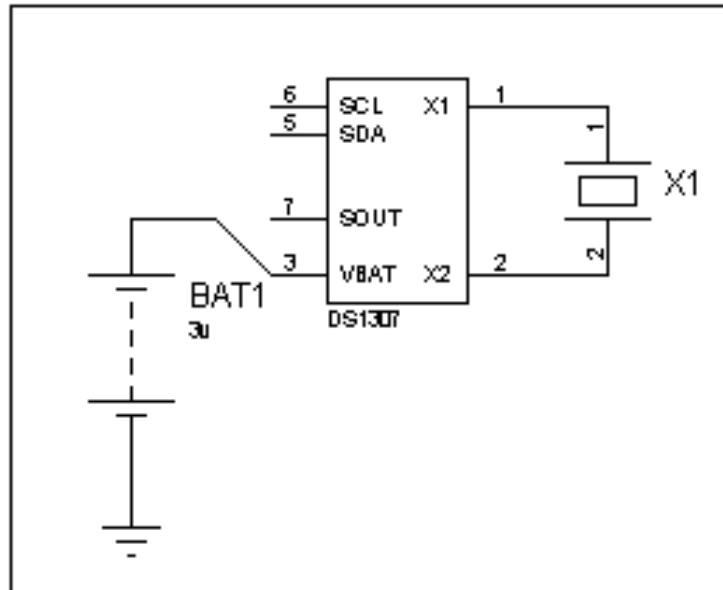
Tabela 2- Endereço 24c64

Componentes	A0	A1	A2
U1	0	0	0
U2	0	1	0
U3	0	0	1

Fonte: Do autor

O RTC é um circuito destinado para manter o controle do relógio do protótipo, na figura 25 está o circuito DS1307, com um cristal de 32768 KHz. Este circuito ainda disponibiliza de uma bateria de 3V, que mantém o funcionamento do DS1307 quando o pino 8 está com nível de tensão abaixo de 4,5 volts. O DS1307 está conectado ao microcontrolador pelo mesmo barramento das memórias 24c64, protocolo I2C.

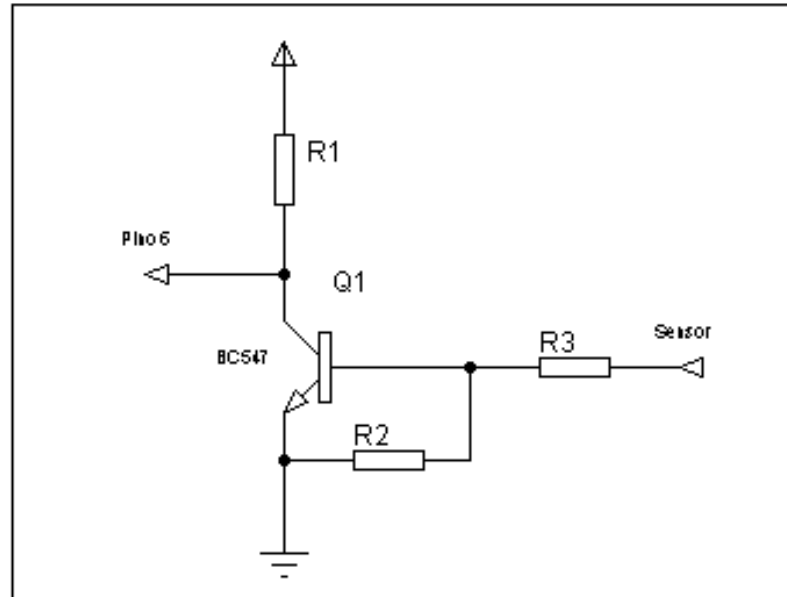
Figura 25- Esquema RTC



Fonte: Do autor.

O terminal 6 do PIC16F887 tem como função no protótipo receber o sinal do sensor indutivo, a figura 26 mostra o circuito de entrada de sinal do sensor, o resistor R1 conectado ao pino 6 ao 5 volts tem como função elevar o nível de tensão do pino 6 para evitar flutuação do mesmo. Um transistor BC547, com seu emissor ligado ao GND, o coletor ligado ao pino 6 MCU, sua base está conectada ao GND através de resistor R2, este com função de manter o nível baixo da base do transistor, quando o mesmo não esta recebendo pulso do sensor, o R3 que está ligado a base e ao sensor, este por sua vez tem a função de limitar a corrente da base do transistor. Quando o sensor é acionado, envia um pulso elétrico de 6 volts ao transistor, o BC547 polariza fazendo com que o nível de tensão do pino 6 MCU fique em nível baixo, sendo possível que internamente no MCU a interrupção do *timer 0* seja disparado.

Figura 26- Esquema entrada sensor



Fonte: Do autor.

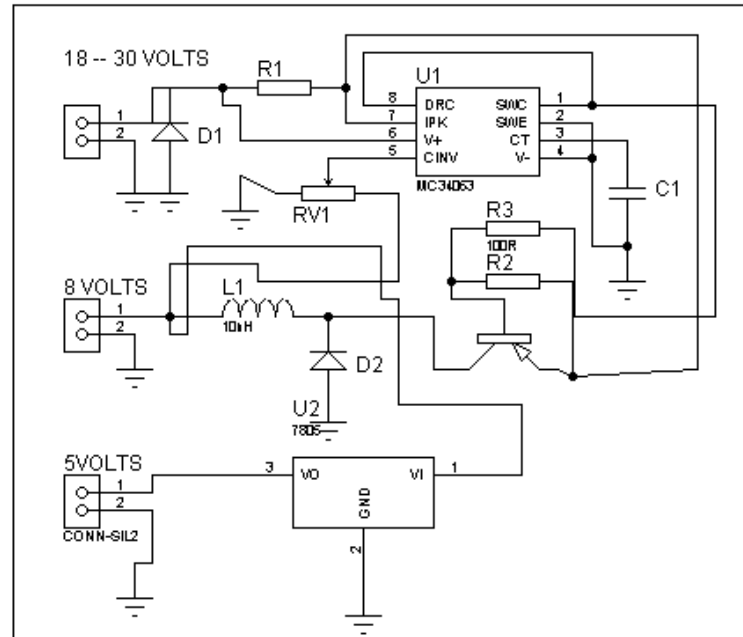
O Atmega 328 tem como papel no circuito fazer a Comunicação do PIC16f887 à mini impressora térmica, ele possui um software internamente destinado a interpretar os dados enviado pelo PIC, estes dados são processados e enviados para a impressora, que esta por sua vez, imprime as informações do veículo, dados do condutor e velocidade registrada nas últimas 24 horas.

4.1.3.2 Circuito eletrônico fonte de alimentação

A fonte de alimentação tem a função de fornecer energia com nível de tensão adequado para os componentes eletrônicos, no protótipo cronotacógrafo digital ela opera com tensão de entrada de 18 a 30 volts, sendo necessário dois níveis de tensões, 5 volts e 8 volts. Para a fonte de 8 volts foi escolhido o CI mc34063, com ele é possível fazer um conversor DC-DC *step-down* com poucos componentes, esta fonte é responsável por fornecer energia para a impressora térmica.

A fonte de alimentação de 5volts, com a função de alimentar os microcontroladores, memórias e DS1307, foi utilizado o CI 7805 que é um regulador de 5 volts, a alimentação do mesmo é fornecida pela fonte de 8 volts, na figura 27 tem-se a fonte do protótipo.

Figura 27- Esquema fonte de alimentação

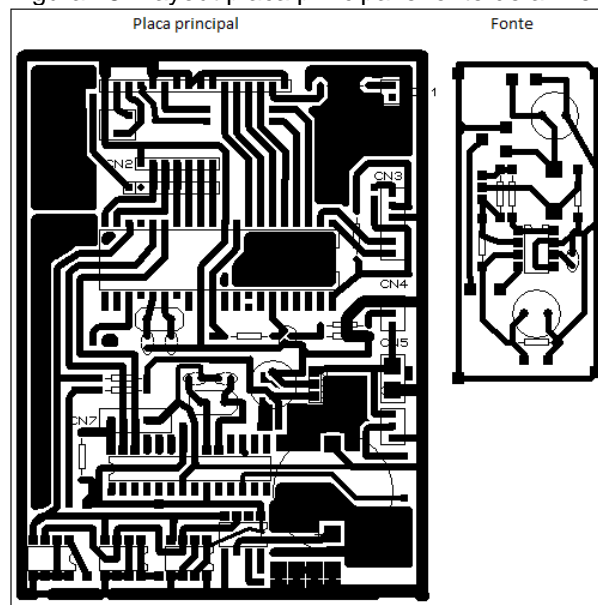


Fonte: Do autor.

4.1.4.2 Confeção da placa de circuito impresso e montagem do protótipo

Para o desenvolvimento do layout da placa se utilizou a ferramenta Ares do Proteus, se optou em dividir o circuito em duas placas: placa principal e placa fonte. Para evitar ruído na placa principal causado pela fonte de alimentação, o posicionamento dos componentes e roteamento das trilhas foram feito manualmente, na figura 28 tem-se o layout das placas.

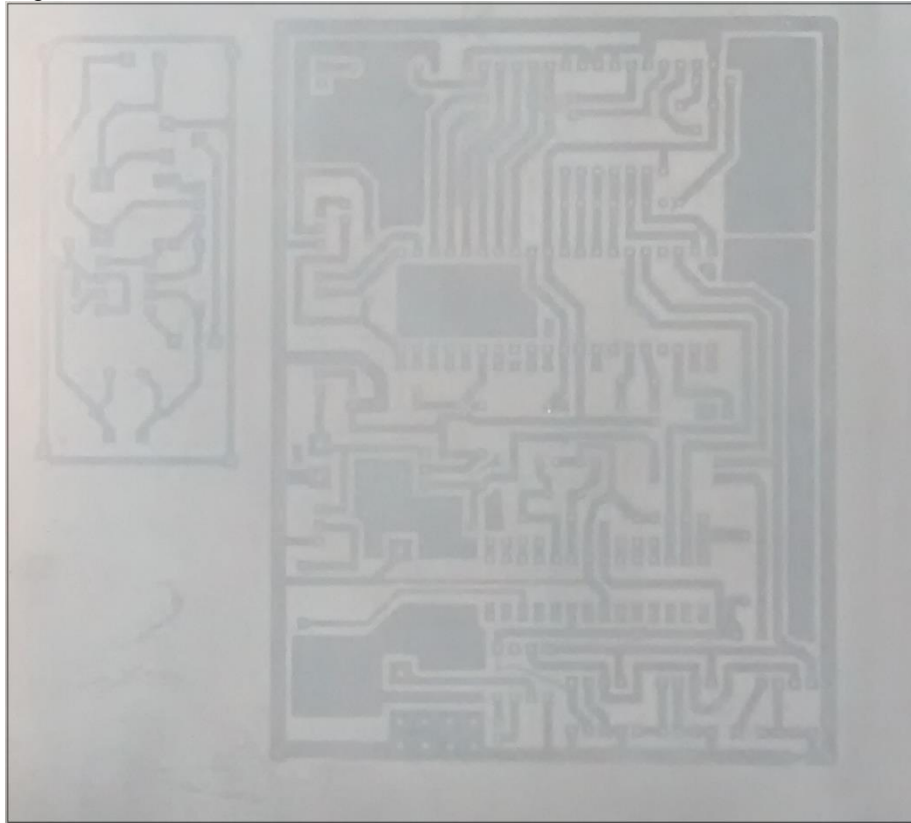
Figura 28- Layout placa principal e fonte de alimentação



Fonte: Do autor.

Para a confecção das placas de circuito impresso foi utilizado o método de serigrafia para fazer a impressão do desenho das trilhas no cobre da placa de fibra de vidro. O processo de serigrafia consiste em uma tela de nylon com a gravura do que se deseja imprimir, para a gravação da gravura na tela, se aplica uma camada de uma emulsão fotossensível, após a secagem da emulsão coloca-se o fotolito e expõe a luz na tela. Nos pontos transparentes do fotolito, a emulsão impermeabiliza e nos pontos não impermeabilizados é possível retirar com água. Com a tela devidamente limpa e seca, é possível observar a gravura na tela, na figura 29 pode se observar a tela gravada com a gravura das trilhas da placa do projeto.

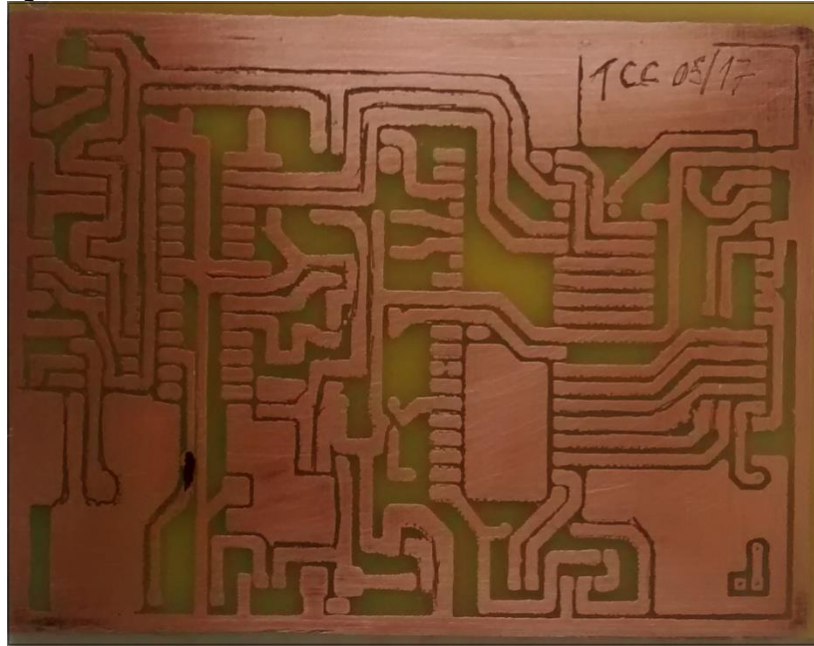
Figura 29- Gravura das trilhas



Fonte: Do autor.

Após a impressão do desenho das trilhas no cobre, foi utilizado o percloro de ferro para a corrosão do cobre, onde o cobre é exposto ao percloro ocorre a corrosão, no cobre com a proteção da tinta, não ocorre a corrosão, o resultado obtido pelo processo é mostrado na figura 30.

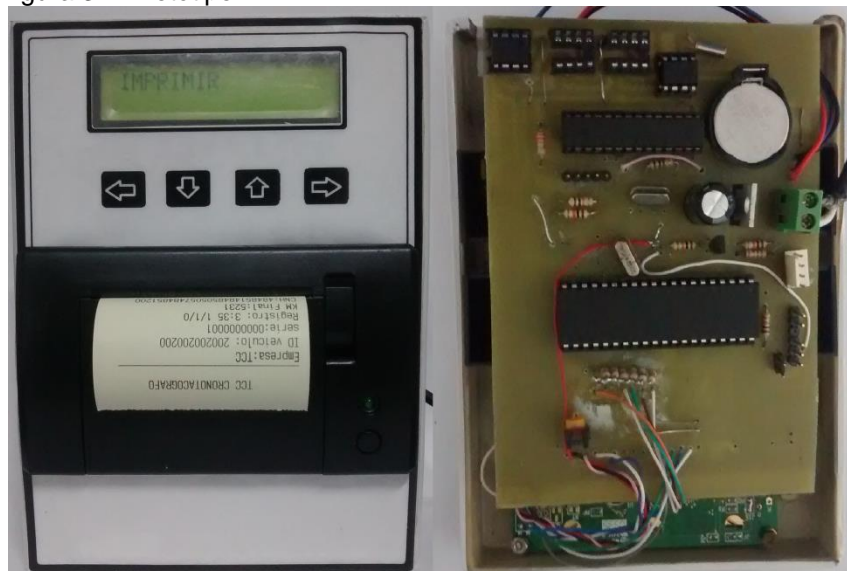
Figura 30- Placa corroída



Fonte:Do autor.

Com a placa corroída e perfurada, foram adicionados os componentes eletrônicos e a solda, após o processo de confecção da placa eletrônica foi realizado o processo de montagem na caixa do protótipo. Na figura 31 tem-se o protótipo montado.

Figura 31- Protótipo



Fonte: Do autor

4.1.4.3 Desenvolvimento do software

Para o desenvolvimento do código para o microcontrolador PIC, foi utilizado o MikroC versão demo, este é um software destinado a programação para

microcontroladores PIC, o mesmo dispõe de várias bibliotecas facilitando assim o desenvolvimento. Para o desenvolvimento do código se utilizou a lógica de armazenar todos os dados de velocidade, distância percorrida, registro do tempo percorrido e os dados do veículo em uma EEPROM 24c64. Para a impressão, os dados são lidos da EEPROM e enviados a outro microcontrolador Atmel através da comunicação serial. O Atmel tem a função de transformar os dados recebidos, em protocolo reconhecido pela a impressora. Os dados de velocidade e distância são armazenados a cada minuto.

Para transformar os pulsos do sensor em velocidade, utilizou-se o contador do microcontrolador: o timer0, que conta os pulsos em um intervalo de 712 milissegundos. Este valor foi calculado a partir dos dados coletados no veículo Mercedes Benz, que a cada 1 segundo se recebe 28,09 pulsos, o veículo está com uma velocidade de 20 Km/h, sendo assim, se receber 20 pulsos em 712 milissegundos, o veículo está a uma velocidade de 20 Km/h. Com base nesse cálculo, com uma leitura de 5056 pulsos, o veículo desloca 1 Km. Na figura 32 demonstra-se o código utilizado para leitura e transformação dos pulsos em velocidade e distância.

Figura 32 - Código leitura sensor

```

//=====ler pulsos=====
numflancos=TMRO;
pulsos = (pulsos + numflancos);
TMRO=0; //Inicializa o registro TMR0 com o valor 0.
Delay ms(714); //Conta durante 714 ms.
numflancos=TMRO; //numflancos=velocidade em km/h.
Gravar_velocidade(numflancos);
pulsos+=TMRO;
TMRO=0;
//=====

```

Fonte: Do autor.

Com os pulsos já transformados em velocidades, são armazenados na EEPROM, o protótipo armazena os dados das ultimas 24 horas, para a leitura e escrita na EEPROM foi criado duas funções: Ler_EEPROM e Grava_EEPROM. A função Ler_EEPROM recebe o parâmetro de endereço e o dado a ser armazenado, Grava_EEPROM recebe o parâmetro de endereço e retorna o byte lido. Para criar as funções se utilizou a biblioteca I2C do MikroC, que já tem a função de *start*, *stop* e escrita do dado no barramento I2C. O endereço a ser gravado o dado, é calculado com base na hora e minuto que a velocidade foi recebida, se utilizou a lógica de calcular o endereço multiplicando a hora por 100 e adicionar os minutos e adicionar 400, para que os dados sejam armazenados de uma forma organizada a partir do endereço 400. Assim, se for 23 horas e 22 minutos, o endereço será 2722,

desta forma é possível posteriormente, saber a hora e o minuto que o dado foi gravado. Para armazenar o deslocamento do veículo se utilizou da mesma lógica. Na figura 32 demonstra-se as funções Gravar_velocidade e Gravar_deslocamento.

Figura 33 - Código gravar velocidade e deslocamento

```

//=====
void Gravar velocidade(int valor){
  char horas,min;
  horas =Ler 24c64(303);
  min=Ler_24c64(302) ;
  Gravar_24c64(((horas*100)+min+400),valor );
}

void Gravar deslocamento(int val){
  char horas,min;
  horas = Ler 24c64(303);
  min= Ler 24c64(302);
  Gravar_24c64(((horas*100)+min+3000),val);
}
//=====

```

Fonte: Do autor.

Para identificar o período que veículo está em condução, se armazena um valor pré-definido na EEPROM, o valor 200 é armazenado nas mesmas posições de memória que as velocidades serão gravadas, com isso, se o veículo está desligado o valor na EEPROM não é modificado. Se o veículo está em funcionamento, logo o valor 200 é sobrescrito pela velocidade. O microcontrolador quando é ligado faz uma atualização nos valores armazenados na memória, se o veículo ficou desligado por um período x, todas as posições da EEPROM correspondente ao período x do desligamento do veículo são atualizadas para o valor 200.

No protótipo é utilizado o DS1307, que informa data e hora para o microcontrolador, o DS1307 se comunica com o microcontrolador pelo protocolo I2C, na figura 34 tem-se o código de leitura dos dados do DS1307.

Figura 34 - Código leitura DS1307

```

//=====
void lerds1307(){
  char horas, minutos, segundos, semana, dia, mes, ano;
  I2C1 Stop();
  I2C1 Start();
  I2C1_Wr(0xd0); //endereço no barramento e indicação de leitura ds1307
  I2C1_Wr(0x00); //endereço inicial
  I2C1 Repeated Start(); //restart no barramento
  I2C1_Wr(0xd1); //endereço no barramento e indicação de leitura
  segundos=Bcd2Dec(I2C1_Rd(1)); //segundos
  minutos=Bcd2Dec(I2C1_Rd(1)); //minutos
  horas=Bcd2Dec(I2C1_Rd(1)); //horas
  semana=Bcd2Dec(I2C1_Rd(1)); //semana
  dia =Bcd2Dec(I2C1_Rd(1)); //dia
  mes=Bcd2Dec(I2C1_Rd(1)); //mes
  ano=Bcd2Dec(I2C1_Rd(1)); //ano
  I2C1 Stop();
  Delay 100ms;
  Gravar 24c64(301,segundos); //segundos
  Gravar 24c64(302,minutos); //minuto
  Gravar 24c64(303,horas); //horas
  Gravar_24c64(304,semana); //semana
  Gravar 24c64(305,dia); //dia
  Gravar 24c64(306,mes); //mes
  Gravar_24c64(307,ano); //ano
}
//=====

```

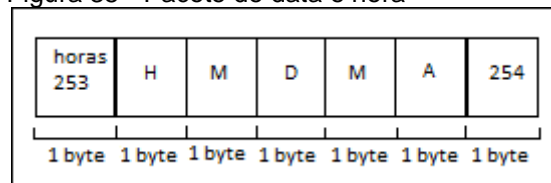
Fonte: Do autor

Para a interação com o dispositivo foi utilizado um *display* LCD 16x2 e quatro botões, para enviar os dados para o display foi utilizado a biblioteca do MikroC que possui as funções de inicializar, limpar, escrever no *display* e entre outras. Para leitura das portas que estão ligadas aos botões se utilizou a biblioteca Buttonn.

Após tentativas de desenvolver uma função para fazer com que impressora térmica imprimisse um gráfico utilizando somente o microcontrolador PIC16f887, não se obteve resultados positivos, foi utilizado o microcontrolador Atmel, o mesmo utilizado no Arduíno Uno, por já possuir uma biblioteca destinada para a impressão utilizando mini impressora térmica, com isso foi utilizada a IDE do Arduíno para o desenvolvimento do código para receber os dados do microcontrolador PIC através da porta serial. No apêndice A tem-se o código completo do microcontrolador Atmel.

Um protocolo básico foi criado, para enviar os dados para o Atmel, na figura 35 tem-se o modelo do pacote de dados data. O byte com valor decimal 253 identifica que vai ser enviado uma sequência de *bytes* que representa a data e hora, um *byte* com valor 254 identifica o fim do pacote.

Figura 35 - Pacote de data e hora



Fonte: Do autor

Para o envio da velocidade, utiliza-se o primeiro byte para informar que uma sequência de byte vai ser enviada, esse byte informa também a hora que os valores foram armazenados, após o primeiro byte ser enviado, é enviado uma sequência de bytes que representa os valores de velocidade, período de condução e distância percorrida, após essa sequência é enviado o byte com valor 254 que representa o fim do pacote. Com os dados recebidos pelo Atmel são processados e são enviados a impressora, uma função foi criada para receber a velocidade, distância percorrida e tempo de condução, essa função processa os dados e envia para a impressora.

O código contém outras funções que auxiliam o funcionamento do protótipo, como ajuste de hora, gravação do último registro, leitura de

quilometragem, gravação de quilometragem, dentre outras. No apêndice D encontra o código no microcontrolador PIC.

4.1.4.5 Teste do protótipo

Com o protótipo montado e os códigos desenvolvidos se iniciou os testes de funcionamento do protótipo. A avaliação do funcionamento do protótipo foi realizada em bancada. Os primeiros teste foram referente as funções: ajuste de hora e data, o funcionamento do *display* LCD e botões, após a primeira etapa de teste, se realizou o teste do sensor indutivo, com o auxílio de um metal se observou as velocidades mostradas, logo se observou que seria inviável o teste de velocidade máxima do protótipo de uma forma manual, induzindo o funcionamento do sensor. Para realizar os testes sem o sensor, se construiu um simulador de sensor, em primeiro momento, o simulador de sensor é construído com CI 555, o simulador de sensor oscila de 1 a 300 Hz.

O segundo teste realizado foi conferir as velocidades mostradas se correspondiam os pulsos que o simulador enviava, nessa primeira etapa, não se encontrou problema do funcionamento de leitura e visualização da velocidade.

No terceiro teste foi realizado a avaliação da impressão, com o protótipo em funcionamento num período de 24 horas com uma velocidade fixa em 0 Km/h, se realizou o teste de impressão, no apêndice B tem-se a fita de diagrama, nesse teste não se observou nenhuma irregularidade dos dados armazenados e impresso.

O quarto teste foi realizado o funcionamento do protótipo com uma velocidade fixa em 90 km/h, também não se observou nenhuma irregularidade dos dados.

No quinto teste foi realizado a simulação do funcionamento do protótipo com variação de velocidade, para simular a variação de velocidade e período de condução, foi construído outro simulador com Arduíno mega, esse simulador construído possui uma função randômica que alterava a velocidade de funcionamento do protótipo. O primeiro teste com outro simulador de velocidade, após a impressão com o protótipo em funcionamento durante uma hora, se observou um problema nos dados impressos.

Para identificar onde estava sendo geradas as informações incorretas, utilizou um leitor de EEPROM, o fluxo de funcionamento do protótipo é dividido em:

armazenar os dados na EEPROM e leitura da EEPROM. Para o armazenamento, os pulsos do sensor são transformados em informações, mostrados no display e são gravados na EEPROM. Para a impressão, os dados são lidos da EEPROM e enviados ao Atmel, que por sua vez, processa e envia a impressora.

Como a velocidade mostrada no display estava correta, se concluí que a etapa de leitura do sensor estava correta, com a leitura da EEPROM, se descobriu que os dados estavam sendo armazenados de uma forma incorreta. O armazenamento de velocidade da forma correta é do endereço 400 até o endereço 2759. Com a análise dos dados lidos na EEPROM com auxílio de leitor externo, se observou que os dados eram sobrescritos a cada 4 horas e 12 minutos. Com o teste realizado no código do microcontrolador PIC se identificou um problema na conversão do endereço da EEPROM na função de gravar_EEPROM, a função não estava gerando o endereço alto do endereço corretamente. Após correção do código, o protótipo armazenou e imprimiu da forma correta. No apêndice C tem-se a fita diagrama com os dados impressos.

4.2 RESULTADOS OBTIDOS

A construção do protótipo do cronotacógrafo digital atendeu as expectativas esperadas, o hardware construído apresentou seu funcionamento de uma forma estável e eficiente. É um circuito relativamente simples, que é possível armazenar os dados do veículo de uma forma simples, a placa do hardware já foi construída pensando na possibilidade de receber outros dispositivos, o microcontrolador Atmel possui 17 pinos que podem ser ligados outros periféricos, dois soquetes com o endereço pré-configurado para outras duas memórias 24cXX que já estão disponíveis no protótipo. Com outras memória é possível fazer o *backup* dos dados.

A comunicação I2C se mostrou eficiente para a comunicação da memória e o RTC com o microcontrolador, não se teve problema com hardware por interferência, alguns problemas encontrados foram decorrentes da construção do código.

A comunicação serial TTL 9600bits por segundos entre microcontroladores teve seu funcionamento esperado. Os dados armazenados na EEPROM e enviados para o microcontrolador Atmel foram recebidos e impressos

corretamente. Por o código não tratar de possíveis erros na comunicação, se houver alguma interferência na comunicação os microcontroladores pode entrar em loops infinitos.

A impressora térmica utilizada atendeu as expectativas de impressão, é um componente de pouca documentação de funcionamento, o pouco conhecimento adquirido para o uso da impressora foi através da análise da biblioteca da mesma, nesse ponto em ter que adicionar outro microcontrolador para a comunicação entre PIC e impressora, não demonstra uma forma eficiente para realidade comercial do dispositivo, para nível acadêmico se disponibilizou a possibilidade da comunicação entre microcontroladores e manipulação de dados.

Com a construção do protótipo cronotacógrafo digital aliada hardware e software alcançaram os resultados esperados que atingindo os objetivos propostos para nível acadêmico, para a utilização como cronotacógrafo, há a necessidade de alteração de software e hardware.

5. CONCLUSÃO

A realização deste trabalho de conclusão de curso exigiu do autor estudos e levantamentos, a fim de conhecer todos os aspectos necessários dos cronotacógrafos disponíveis no mercado, para então, iniciar o planejamento e construção do protótipo. O trabalho cumpriu com os objetivos propostos de forma eficaz e ainda possibilitou a criação do protótipo com vantagens relevantes como: o seu baixo custo de manutenção e a comodidade proporcionada ao condutor (por dispensar a troca do disco-diagrama).

Para a homologação junto ao INMETRO faz-se necessárias inúmeras adequações no projeto, o uso do protótipo não é proibido na utilização para controle de veículos com peso inferior a 4536 kg. Devida a impossibilidade de sua homologação junto ao INMETRO, a comparação como um produto final diante de outros modelos disponíveis no mercado tornou-se inviável.

No desenvolvimento do protótipo foi indispensável conhecimento técnico em eletrônica bem como na programação de microcontroladores. Durante a construção do hardware não foi encontrada qualquer dificuldade, por outro lado, em nível de software, algumas dificuldades foram detectadas.

Dentre as dificuldades enfrentadas na construção do projeto, devem-se destacar as falhas de comunicação com a impressora, que foram superadas com o auxílio de uma biblioteca para Arduino, que abstraiu a comunicação com a impressora e tornou possível a criar uma função para a impressão dos gráficos de velocidade.

O protótipo apresentou um funcionamento esperado. O dispositivo possibilita cadastrar o motorista, obter a maior velocidade registrada no período das 24 horas, ou a impressão de velocidades, tempo de operação e distância percorrida no período das 24 horas.

À partir da finalização deste projeto, pode-se observar as possibilidades de aprimoramentos para trabalhos futuros, por exemplo: a implantação de um leitor RFID ao hardware para uma fácil identificação do motorista e a implantação de um módulo de comunicação sem fio, para dispositivos smartphones, com o objetivo de receber os dados registrados no veículo junto a um software gestor que auxiliaria na análise de dados.

REFERÊNCIAS

ALENCAR, Márcio Aurélio dos Santos. **Fundamentos de Redes de Computadores**. Disponível

em: <http://redeetec.mec.gov.br/images/stories/pdf/eixo_infor_comun/tec_man_sup/081112_fund_redes_comp.pdf>. Acesso em: 28 de outubro de 2015.

ANDERSON, Ross. **On the security of digital tachographs**. In: European Symposium on Research in Computer Security. Springer Berlin Heidelberg, 1998. p. 111-125.

ATMEL (Org.). **Tecnologias de microcontroladores comprovadas e inovadoras com baixo consumo de energia para as suas aplicações exclusivas**. Disponível em: <<http://www.atmel.com/pt/br/products/microcontrollers/default.aspx>>. Acesso em: 22 de maio 2015.

BRASIL. DENATRAN(Org.). **Frota de veículos**. Disponível em: <<http://www.denatran.gov.br/frota2015.htm>>. Acesso em: 22 de maio 2015.

CANZIAN, Edmur. **Comunicação Serial – RS232**. São Paulo: Editora da Escola Técnica CNZ de Cotia, 2002.

DEMCZUK, Thiago. **A rede brasileira de metrologia legal e seu papel na verificação de cronotacógrafo**. 2010.70f. Monografia(Especialista em Metrologia Legal) – Universidade Federal do Paraná, Curitiba.

FERNANDEZ, Breno Ortega. **Proposta de um sistema eletrônico embarcado para fiscalização automática de veículos rodoviários de carga**.2010.131f.Tese(Doutorado em Engenharia Mecânica).Área de Concentração: Dinâmica de Máquinas e Sistemas – Universidade de São Paulo, São Carlos.

FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores**.3.ed. Porto Alegre: Artmed, 2006.

IBRAHIM, Dogan. **Using LEDs, LCDs and GLCDs in microcontroller projects**. John Wiley& Sons, 2012.

IMTT. Instituto da mobilidade e dos transportes. Disponível em: <<http://www.imtt.pt/sites/imtt/Portugues/TransportesRodoviaros/TacografoDigital/Paginas/TacografoDigital.aspx>>.Acesso em 5 de nov. 2015.

INMETRO. **O que é cronotacógrafo**: História do cronotacógrafo. Disponível em: <<http://cronotacografo.rbmlq.gov.br/o-que-e-cronotacografo>>. Acesso em: 20 maio 2015.

LIMA E SILVA, Itallo Santos et al. **Aplicações do protocolo I2C em sistemas microcontrolados**. In: VII CONNEPI-Congresso Norte Nordeste de Pesquisa e Inovação. 2012.

MAZIDI, M et al. **The AVR microcontroller and embedded system: using assembly and c**. New Jersey: Prentice Hall, 2011.

MIYADAIRA, Alberto Noboru. **Microcontroladores PIC18: Aprenda e programe em linguagem C.4.ed.**São Paulo:Érica,2009.

PERCILIA, Eliene. **Comunicação de Dados**. Disponível em: <<http://www.brasilecola.com/informatica/comunicacao-dados.htm>>. Acesso em: 01 de novembro de 2015.

PRADO, Marcelo Vinaud. **Data Warehouse para apoio a gestão da operação em Empresas do Transporte rodoviário Interestadual de Passageiros**. 2006. 111 f. Dissertação (Mestrado) - Curso de Faculdade de Tecnologia, Engenharia Civil e Ambiental, Universidade de Brasília:Brasília, 2006.

SEVA. **Liberdade é poder imprimir relatórios a qualquer momento e sempre que quiser**. Disponível em:<<http://www.seva.com.br/produto/10/SVT3000-A/?query=fitadiagrama>>. Acesso em: 25 de maio 2016.

SILVEIRA, Jorge Luis da. **Comunicação de dados e sistema de teleprocessamento**.São Paulo: Makron, 1991.

SOARES, Luciana. **Dúvidas! Modelos de cronotacógrafos**. Disponível em: <<https://logisticasemmedo.wordpress.com/category/cronotacografos/>>. Acesso em: 22 de maio 2015.

SOUSA, Lindeberg Barros de. **Redes de Computadores: dados, voz e imagem**. 4. ed. São Paulo: Érica, 1999.

SOUZA, David. **Desbravando o PIC: Ampliado e atualizado para PIC16F628A**. 8. ed. São Paulo: Editora Érica, 2003.

TANENBAUM, Andrew S.**Organização estruturada de computação**. 5. ed. Rio de Janeiro: Pearson Prentice, 2007.

TOCCI, R. J.; WIDMER, N.S.; MOSS, G.L. **Sistemas Digitais: Princípios e Aplicações**.11. ed. São Paulo: Pearson Prentice Hall, 2007.

ZANCO, Wagner da Silva. **Microcontroladores PIC: Uma abordagem prática e objetiva**. São Paulo: Érica, 2005.

APÊNDICES

APÊNDICE A -Código do microcontrolador Atmel

```

#include "Adafruit_Thermal.h"
#include "SoftwareSerial.h"
SoftwareSerialSerialImpressora(10, 11); // RX TX
SoftwareSerialmySerial(0, 1); // RX, TX
Adafruit_Thermalprinter(&SerialImpressora); // Passaddrtoprinterconstructor
String placa = "error", horas = "error", hodometro = "error", ch = "error", caractere;
int velocidade[130],tempo = 0,hora = 0,x = 0,contadorhoras = 0,contador = 0;
intaux,dados, estado = 1, velocidadeanterior=0, teste=0, v=0, horasleitura;
char buf;

voidimprimevelocidade(intvelocidadeatual,intconducao, int distancia){
    uint8_t data[]={0x00, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, //30
                    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //60
                    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //90
                    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //120
                    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //150
                    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                    0x00, 0x00, 0x00,0x00,0x00
    };
    velocidadeatual=velocidadeatual+15;
    velocidadeanterior=velocidadeanterior+15;
    intvalor_ anterior[6]={0x03,0xff,0x3f,0x1f,0x0f,0x03};
    intvalor_ atual[6]={0x00,0x3f,0x0f,0x07,0x03,0x00};
    int y = (velocidadeatual / 5) ;
    int x = (velocidadeanterior / 5);
    intrestoa = velocidadeanterior % 5;
    intrestob = velocidadeatual % 5;
    if(velocidadeanterior<velocidadeatual){
        data[x]=valor_ anterior[restoa];
        for (inti =(x+1); i< 35; i++) {
            data[i] = 0xff;
        }
        data[y]=data[y]-valor_ atual[restob];
        for (inti =(y+1); i< 35; i++) {
            data[i] = 0x00;
        }
    }elseif(velocidadeanterior>velocidadeatual){
        data[y]=valor_ anterior[restob];
        for (inti =(y+1); i< 35; i++) {
            data[i] = 0xff;
        }
        data[x]=data[x]-valor_ atual[restoa];
        for (inti =(x+1); i< 35; i++) {
            data[i] = 0x00;
        }
    }else data[x]=valor_ anterior[restoa]-valor_ atual[restoa];
}

```

```

if (tempo > 4) {
    tempo = 0;
    hora++;
    data[0] = 0xff;
}
tempo++;
x++;
if(conducao==1){
    data[40]=0xff;
}else data[40]=0x02;

switch (distancia) {
    case 0:data[45]=0x00; break;
    case 1:data[45]=0xc0; break;
    case 2:data[45]=0xe0; break;
    case 3:data[45]=0xf0; break;
case 4:data[45]=0xf8; break;
    case 5:data[45]=0xfc; break;
case 6: data[45]=0xfe; break;
    case 7: data[45]=0xff; break;
    case 8: data[45]=0xff; data[46]=0x80; break;
    case 9: data[45]=0xff; data[46]=0xe0; break;
    case 10:data[45]=0xff; data[46]=0xf0;break;
}
printer.printBitmap(397, 1, data, false);
for(int i=0;i<46;i++) data[i]=0x00;
data[9]=0x03; data[15]=0x03; data[21]=0x03; data[27]=0x03;
data[33]=0x03;printer.printBitmap(397, 1, data, false);
velocidadeanterior=(velocidadeatual-15);

}

voidcabecalho(String placa, String hora, Stringhodometro, Stringcnh) {
    /*- Fabricante e modelo do equipamento
    - Identificação do veículo
    - Identificação do condutor
    - Indicação inicial do hodômetro
    - Data e hora do início do registro gráfico
    - Número de série do equipamento
    - Número de ordem da fita diagrama impressa.
    */
    Stringauxtexto;
    printer.println(F("      TCC CRONOTACOGRAFO"));
    printer.println(F("_____"));
    printer.println(F("Empresa:TCC"));
    auxtexto = "ID veiculo: ";
    printer.print(auxtexto);
    printer.println(placa);
    printer.println("serie:0000000001");
    printer.print("Registro: ");
    printer.println(hora);

```

```

printer.print("KM Final:");
printer.println(hodometro);
printer.print("CNH:");
printer.println(cnh);
printer.println(F("_____"));
printer.println(F("KMO 30 60 90 120 150 T D"));

```

```

}
String lercnh() {
int dados[12], f = 1,x = 0;
while (f) {
if (mySerial.available()) {
dados[x] = mySerial.read();
if (dados[x] == 254) f = 0;
x++;
}
}
String ch;
for (inti = 0;i<x-1; i++)
ch += dados[i];

```

```

return ch;
}

```

```

//=====
=====

```

```

intlerhodometro() {
int f = 1;
inti = 0;
int dados[8];
while (f) {
if (mySerial.available()) {
dados[i] = mySerial.read();
if (dados[i] == 254) f = 0;
i++;
}
}
}
inthodometro;
hodometro =(dados[0]*256)+dados[1];
return hodometro;
}

```

```

String lerhora() {
int f = 1;
inti = 0;
int dados[5];
while (f) {
if (mySerial.available()) {
dados[i] = mySerial.read();

```

```

if (dados[i] == 254) f = 0;
i++;

```

```

    }
}

```

String hora;

```

    hora += dados[0];
    hora += ':';
    hora += dados[1];
    hora += ' ';
    hora += dados[2];
    hora += '/';
    hora += dados[3];
    hora += '/';
    hora += dados[4];
return hora;
}

```

```

String lerplaca() {
int f = 1;
int x = 0;
int dados[9];
while (f) {
    if (mySerial.available()) {
        dados[x] = mySerial.read();
        if (dados[x] == 254) f = 0;
        x++;
    }
}
}

```

```

String placa;
for (inti = 0; i < x-1; i++)
placa += dados[i];
return placa;
}

```

```

void setup() {
    SerialImpressora.begin(9600); // inicializacao porta serial
    printer.begin(); // Init printer (same regardless of serial type)
    mySerial.begin(9600);
    printer.feed(2);
    printer.sleep(); // Tell printer to sleep
        delay(3000L); // Sleep for 3 seconds
    printer.wake(); // MUST wake() before printing again, even if reset
    printer.setDefault(); // Restore printer to defaults
}

```

```

}
void loop() {
    int c ;
    if (mySerial.available()) {
        c = mySerial.read();
}

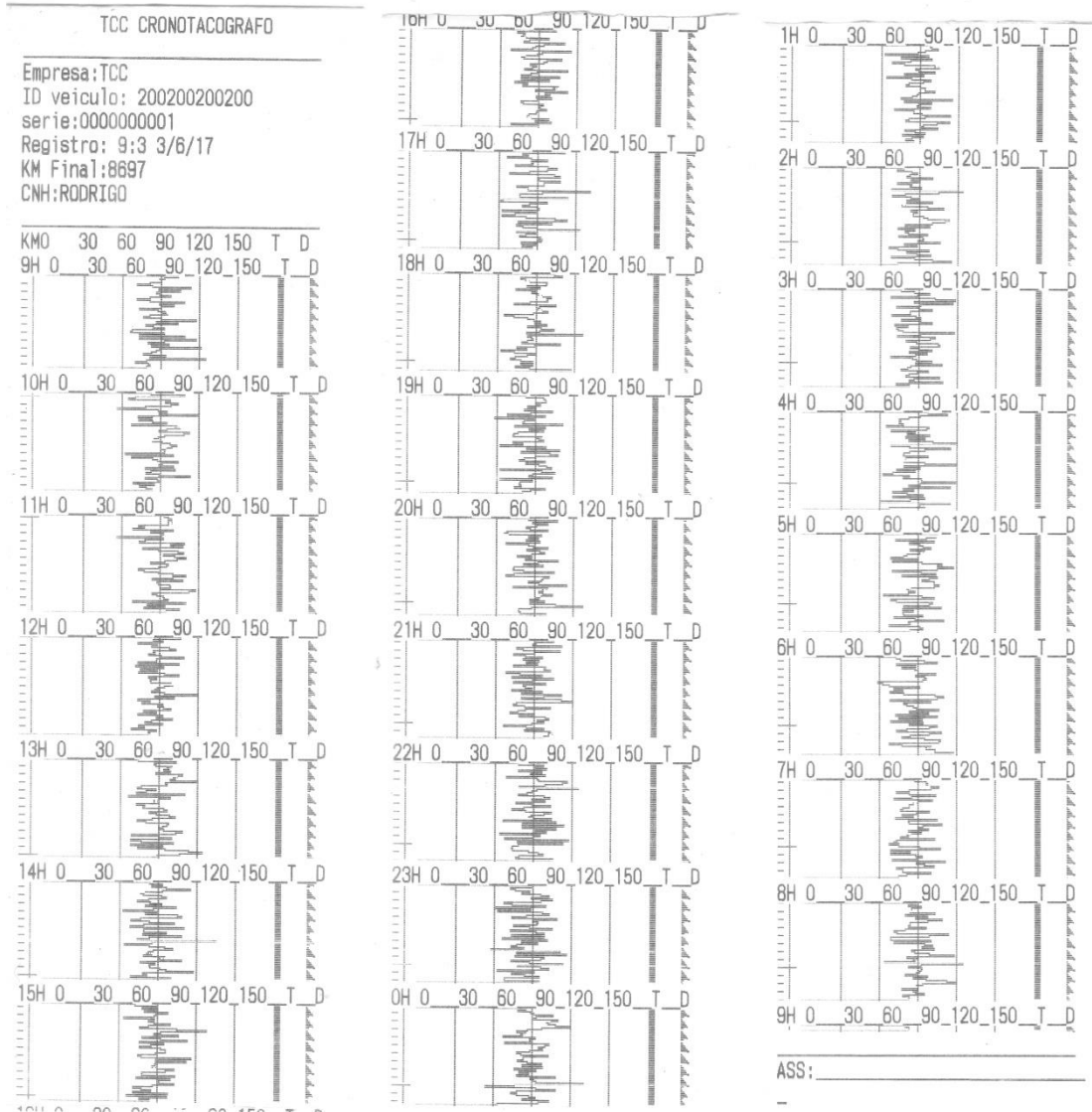
```

```

    }
    if (c == 255) placa = lerplaca();
    if (c == 253) horas = lerhora();
    if (c == 252) hodometro = lerhodometro();
    if (c == 251) {
        ch = lercnh();
        cabecalho(placa, horas,hodometro, ch);
        horasleitura=1;
        mySerial.write(horasleitura);
        v=0;
    }
    for(int i=0;i<130;i++)velocidade [i] ==0;
    if((c>219)&&(c<251)){
        int f = 0, i = 0;
        horasleitura=(c-220);
        c=0;
        while (f==0) {
            v=0;
            if (mySerial.available()) {
                i++;
                velocidade[i]=mySerial.read();
                if (velocidade [i] == 254) f = 1;
                if(velocidade [i] == 249) f = 2;
            }
        }
    }
    printer.print(horasleitura);
    printer.println(F("H 0__30__60__90__120__150__T__D"));
    intaux=0;
    for(int x=0;x<=i;x=x+2){
        aux=auxx+1;
        if(auxx>10)auxx=0;
        delay(10);
        if((velocidade[x+1])< 150){
            imprimevelocidade(velocidade[x+1],1,velocidade[x]);
        }else{
            imprimevelocidade(0,0,velocidade[x]);
        }
    }
    if(f==2){
        printer.println(F("_____"));
        printer.println(F("ASS:_____"));
        printer.println();
        printer.println();
        mySerial.write(25);
    }
    mySerial.write(horasleitura);
}
}
}

```


APÊNDICE C - Fita diagrama com velocidade simulada



APÊNDICE D-Código PIC16F887

```

#include "24c64.h"
#include "ds1307.h"
#include "menu.h"
//=====
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D7_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB0_bit;
bit flag1,flag2,ok;
int numflancos=0; // numero de pulso
int pulsos=0;
unsigned long km=0;
char deslocamento=0;
//=====
char* codetxt_to_ramtxt(const char* ctxt){
    static char txt[20];
    char i;
    for(i=0; txt[i] = ctxt[i]; i++);
    return txt;
}
//=====
void main() {
    char menu=0;
    flag1=1;flag2=1,ok=1;
    UART1_Init(9600);I2C1_Init(100000);
    OPTION_REG=0b10111111;
    TRISD=0b11111111; ANSEL = 0;ANSELH = 0;C1ON_bit = 0;C2ON_bit = 0;
    Lcd_Init();
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Cmd(_LCD_CURSOR_OFF);
    alfabeto();
    km=Ler_KM();
    lerds1307();
    deslocamento=ler_deslocamento();
    ler_km();
    tcc();Delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    msgAjuste();
    while(ok){
        menus(menu);
        if ((flag1) && (Button(&PORTd, 4, 1, 0))) {

```

```

        menu++;
        Lcd_Cmd(_LCD_CLEAR);
        flag1=0;
        if (menu>5){
menu=0;
}
        } else if(Button(&PORTd, 4, 0, 1)){ flag1=1;}

        if ((flag2) &&( Button(&PORTd, 7, 1, 0))) {
                flag2=0;
Lcd_Cmd(_LCD_CLEAR);
if (menu<1){
menu=6;
}
menu--;
        } else if(Button(&PORTd, 7, 0, 1)){flag2=1;}

if((menu==2)&&(Button(&PORTd, 6, 1, 0))) ok=0;
}
limpar_eeprom();
while(1){
    char texto1[7];
lerds1307();
Gravar_deslocamento(deslocamento);
Gravar_registro();
//=====lerpulsos=====
numflancos=TMR0;
pulsos = (pulsos + numflancos);
TMR0=0;          //Inicializa o registro TMR0 com o valor 0.
Delay_ms(714);   //Conta durante 714 ms.
numflancos=TMR0; //numflancos=velocidade em km/h.
Gravar_velocidade(numflancos);
pulsos+=TMR0;
TMR0=0;
//=====
if(pulsos>5056){
pulsos=0;
        km++;
if(deslocamento<10){
        deslocamento++;
        }else deslocamento=0;
Gravar_Km(km);
}
Lcd_Cmd(_LCD_CLEAR);
IntToStr(numflancos,texto1); //Transforma rpm em texto.
Lcd_Out(2,1,codetxt_to_ramtxt("KM/h")); //Escreve no LCD
Lcd_Out_CP(texto1);
Lcd_Out(1,1,codetxt_to_ramtxt("KM")); //Escreve no LCD
IntToStr(km,texto1);
Lcd_Out_CP(texto1);
IntToStr(deslocamento,texto1);

```

```
Lcd_Out(1,10,texto1);
}
}
```

```
//=====
//=====DS1307=====
#include "msg.h"
```

```
void lerds1307(){
    char horas, minutos,segundos,semana,dia,mes,ano;
    I2C1_Stop();
    I2C1_Start();
    I2C1_Wr(0xd0);//endereço no barramento e indicação de leitura ds1307
    I2C1_Wr(0x00);//endereço inicial
    I2C1_Repeated_Start();//restart no barramento
    I2C1_Wr(0xd1);//endereço no barramento e indicação de leitura
    segundos=Bcd2Dec(I2C1_Rd(1));//segundos
    minutos=Bcd2Dec(I2C1_Rd(1));//minutos
    horas=Bcd2Dec(I2C1_Rd(1));//horas
    semana=Bcd2Dec(I2C1_Rd(1));//semana
    dia =Bcd2Dec(I2C1_Rd(1));//dia
    mes=Bcd2Dec(I2C1_Rd(1));//mes
    ano=Bcd2Dec(I2C1_Rd(1));//ano
    I2C1_Stop();
    Delay_100ms;
    Gravar_24c64(301,segundos);//segundos
    Gravar_24c64(302,minutos);//minuto
    Gravar_24c64(303,horas);//horas
    Gravar_24c64(304,semana);//semana
    Gravar_24c64(305,dia);//dia
    Gravar_24c64(306,mes);//mes
    Gravar_24c64(307,ano);//ano
}
void mostra_hora_data(){
    char texto1[7];
    lerds1307();
    IntToStr(Ler_24c64(303),texto1);
    Lcd_Chr(1,1,texto1[4]);
    if(texto1[4]==' ')Lcd_Chr(1,1,'0');
    Lcd_Chr(1,2,texto1[5]);
    Lcd_Chr(1,3,':');

    IntToStr(Ler_24c64(302),texto1);
    Lcd_Chr(1,4,texto1[4]);
    if(texto1[4]==' ')Lcd_Chr(1,4,'0');
    Lcd_Chr(1,5,texto1[5]);
    Lcd_Chr(1,6,');

    IntToStr(Ler_24c64(305),texto1);
    Lcd_Chr(1,7,texto1[4]);
    if(texto1[4]==' ')Lcd_Chr(1,7,'0');
```

```

    Lcd_Chr(1,8,texto1[5]);
    Lcd_Chr(1,9,');

    IntToStr(Ler_24c64(306),texto1);
    Lcd_Chr(1,10,texto1[4]);
    if(texto1[4]==' ')Lcd_Chr(1,10,'0');
    Lcd_Chr(1,11,texto1[5]);
    Lcd_Chr(1,12,');

    IntToStr(Ler_24c64(307),texto1);
    Lcd_Chr(1,13,texto1[4]);
    Lcd_Chr(1,14,texto1[5]);
}
char ajustehoras(){
    char horas, minutos,segundos,semana,dia,mes,ano,ok=0,coluna=0;
    char texto1[7];
    horas= Ler_24c64(303);minutos=Ler_24c64(302);dia=Ler_24c64(305);
    mes=Ler_24c64(306);ano=Ler_24c64(307);
    while(ok<20){
        Delay_ms(200);
        ok++;
        Lcd_Cmd(_LCD_CLEAR);
        if(Button(&PORTd, 6, 1, 0)){
            coluna++;
            if(coluna>4){
                coluna=0;
            }
            ok=0;
        }
        if(Button(&PORTd, 5, 1, 0)){
            if(coluna<1){
                coluna=5;
            }
            coluna--;
            ok=0;
        }

        switch(coluna){
            case 0:{
                if (Button(&PORTd, 7, 1, 0)) {
                    horas++;
                    if(horas>24){
                        horas=1;
                    }
                }
                ok=0;
            }
            if (Button(&PORTd, 4, 1, 0)) {
if(horas<1){
                horas=25;

```

```

        }
        horas--;
        ok=0;
    }
    msgAjuste() ;

    IntToStr(horas,texto1);
    Lcd_Out(2,1,texto1);
    break;
    }
    case 1:{

        if (Button(&PORTd, 7, 1, 0)) {
            minutos++;
            if(minutos>59) minutos=1;
            ok=0;
        }

        if (Button(&PORTd, 4, 1, 0)) {
            if(minutos<1)minutos=60;
            minutos--;
            ok=0;
        }
        msgAjusteminuto();
        IntToStr(minutos,texto1);
        Lcd_Out(2,1,texto1);
        break;
    }
    case 2:{
        if (Button(&PORTd, 7, 1, 0)) {
            dia++;
            if(dia>31)dia=1;
            ok=0;
        }
        if (Button(&PORTd, 4, 1, 0)) {
            if(dia<1)dia=32;
            dia--;
        }
        ok=0;
    }
    msgAjustedia();
    IntToStr(dia,texto1);
    Lcd_Out(2,1,texto1);
    break;
    }
    case 3:{

        if (Button(&PORTd, 7, 1, 0)) {
            mes++;
            if(mes>12){
                mes=1;
            }
            ok=0;
        }
    }

```

```

        }
        if (Button(&PORTd, 4, 1, 0)) {
            if(mes<1) mes=13;
            mes--;
        }
    ok=0;
}
msgAjustemes();
IntToStr(mes,texto1);
Lcd_Out(2,1,texto1);
break;
}
case 4:{
    if (Button(&PORTd, 7, 1, 0)) {
        ano++;
        if(ano>99){
            ano=0;
        }
        ok=0;
    }
}
    if (Button(&PORTd, 4, 1, 0)) {
        if(minutos<1){
            ano=100;
        }
        ano--;
        ok=0;
    }
}
msgAjusteano();
IntToStr(ano,texto1);
Lcd_Out(2,1,texto1);
break;
}
}
}
I2C1_Start();
I2C1_Wr(11010000);
I2C1_Wr(0x00);
I2C1_Wr(Dec2Bcd(0));//segundos
I2C1_Wr(Dec2Bcd(minutos));
I2C1_Wr(Dec2Bcd(horas));
I2C1_Wr(Dec2Bcd(0));//dia da semana
I2C1_Wr(Dec2Bcd(dia));
I2C1_Wr(Dec2Bcd(mes));
I2C1_Wr(Dec2Bcd(ano));
I2C1_Stop();
Delay_100ms;
return 0;
}

```

```

//=====24C64.H=====
char horas,minutos,dia,mes,ano;
void Gravar_24c64(unsigned intendereco, int dados){
    unsigned int temp;
    I2C1_Stop();
    I2C1_Start();
    I2C1_Wr(0b10100000);
temp=endereco>>8;
    I2C1_Wr(temp);
    I2C1_Wr(endereco);
    I2C1_Wr(dados);
I2C1_Stop();
Delay_ms(10);
}
char Ler_24c64(unsigned intendereco){
    char dados;
    unsigned int temp;
    I2C1_Stop();
    I2C1_Start();
    I2C1_Wr(0b10100000);
temp=endereco>>8;
    I2C1_Wr(temp);
I2C1_Wr(endereco);
    I2C1_Repeated_Start();
I2C1_Wr(0b10100001);
    dados=I2C1_Rd(0);
I2C1_Stop();
Delay_ms(5);
    return dados;
}
char ler_deslocamento(){
intend=(Ler_24c64(333)*100)+(Ler_24c64(334))+3000;
return Ler_24c64(end);
}
void Gravar_Km(long km){
char kma= km>>8;

    Gravar_24c64(331,kma);
    Gravar_24c64(332,km);
}
voidGravar_registro(){
    Gravar_24c64(333,(Ler_24c64(303)));
    Gravar_24c64(334,Ler_24c64(302));
    Gravar_24c64(335,Ler_24c64(305));
    Gravar_24c64(336,Ler_24c64(306));
    Gravar_24c64(337,Ler_24c64(307));
}
voidGravar_velocidade(int valor){
    char horas,min;
    horas =Ler_24c64(303);
    min=Ler_24c64(302) ;
}

```

```

    Gravar_24c64(((horas*100)+min+400),valor );
}
voidGravar_deslocamento(intval){
    char horas,min;
    horas = Ler_24c64(303);
    min= Ler_24c64(302);
    Gravar_24c64(((horas*100)+min+3000),val);
}
longLer_KM(){
return (Ler_24c64(331)<<8)+(Ler_24c64(332));
}
voiddecrementa_tempo(){
if(minutos>0){
    minutos--;
} //if(minutos)
else{
    minutos=59;
if(horas>0){
    horas--;
} //ifhoras
else{
    horas=23;
if(dia>1){
    dia--;
} //if dia
else{
    if(mes>1){
mes--;
} //if mes
else{
mes=12;
ano--;
} //else mes
if(mes==2){
dia=28;
} //if mes==2
else if((mes==1)|| (mes==3)|| (mes==5)|| (mes==7)|| (mes==8)
|| (mes==10)|| (mes==12)){
dia=31;
} //else if mes==2
else{
    dia==30;
}
} //else dia
} //else horas
} //else minutos
}
voidlimpar_eeprom(){
inti,ends,enderec;char texto1[7];
    horas=Ler_24c64(303);minutos=Ler_24c64(302);dia=Ler_24c64(305);
mes=Ler_24c64(306);ano=Ler_24c64(307);

```

```

        for(i=0;i<1500;i++){
if((Ler_24c64(333)==horas)&&(Ler_24c64(334)==minutos)&&
    (Ler_24c64(335)==dia)&&(Ler_24c64(336)==mes)&&
    (Ler_24c64(337)==ano)){
    i=1500;
    }else{
endrec=((horas*100)+(minutos)+400);
ends=((horas*100)+(minutos)+3000);
IntToStr(horas,texto1);
Lcd_Out(1,1,texto1);
IntToStr(minutos,texto1);
Lcd_Out_CP(texto1);
    Gravar_24c64(endrec,200);
Delay_ms(10);
    Gravar_24c64(ends,0);
decrementa_tempo();
    }
    }
}
void alfabeto(){
    Gravar_24c64('A','A');Gravar_24c64('B','B');Gravar_24c64('C','C');
    Gravar_24c64('D','D');Gravar_24c64('E','E');Gravar_24c64('F','F');
    Gravar_24c64('G','G');Gravar_24c64('H','H');Gravar_24c64('I','I');
    Gravar_24c64('J','J');Gravar_24c64('K','K');Gravar_24c64('L','L');
    Gravar_24c64('M','M');Gravar_24c64('N','N');Gravar_24c64('O','O');
    Gravar_24c64('Q','Q');Gravar_24c64('R','R');Gravar_24c64('S','S');
    Gravar_24c64('T','T');Gravar_24c64('U','U');Gravar_24c64('V','V');
    Gravar_24c64('W','W');Gravar_24c64('X','X');Gravar_24c64('Y','Y');
    Gravar_24c64('Z','Z');Gravar_24c64('0','0');Gravar_24c64('1','1');
    Gravar_24c64('2','2');Gravar_24c64('3','3');Gravar_24c64('4','4');
    Gravar_24c64('5','5');Gravar_24c64('6','6');Gravar_24c64('7','7');
    Gravar_24c64('8','8');Gravar_24c64('9','9');

}
//=====msg.h=====
#include "ds1307.h"
void tcc(){
Lcd_Chr(1,7,Ler_24c64('T'));
Lcd_Chr_cp(Ler_24c64('C'));
Lcd_Chr_cp(Ler_24c64('C'));
Lcd_Chr(2,1,Ler_24c64('C'));
Lcd_Chr_cp(Ler_24c64('R'));
Lcd_Chr_cp(Ler_24c64('O'));
Lcd_Chr_cp(Ler_24c64('N'));
Lcd_Chr_cp(Ler_24c64('O'));
Lcd_Chr_cp(Ler_24c64('T'));
Lcd_Chr_cp(Ler_24c64('A'));
Lcd_Chr_cp(Ler_24c64('C'));
Lcd_Chr_cp(Ler_24c64('O'));
Lcd_Chr_cp(Ler_24c64('G'));
Lcd_Chr_cp(Ler_24c64('R'));

```

```

Lcd_Chr_cp(Ler_24c64('A'));
Lcd_Chr_cp(Ler_24c64('F'));
Lcd_Chr_cp(Ler_24c64('O'));
    }
voidmsgAjuste(){
Lcd_Chr(1,1,Ler_24c64('A'));
Lcd_Chr_cp(Ler_24c64('J'));
Lcd_Chr_cp(Ler_24c64('U'));
Lcd_Chr_cp(Ler_24c64('S'));
Lcd_Chr_cp(Ler_24c64('T'));
Lcd_Chr_cp(Ler_24c64('E'));
Lcd_Chr_cp(' ');
}
voidmsgmax(){
Lcd_Chr(1,1,Ler_24c64('V'));
Lcd_Chr_cp(Ler_24c64('E'));
Lcd_Chr_cp(Ler_24c64('L'));
Lcd_Chr_cp(Ler_24c64('O'));
Lcd_Chr_cp(Ler_24c64('C'));
Lcd_Chr_cp(Ler_24c64('I'));
Lcd_Chr_cp(Ler_24c64('D'));
Lcd_Chr_cp(Ler_24c64('A'));
Lcd_Chr_cp(Ler_24c64('D'));
Lcd_Chr_cp(Ler_24c64('E'));
Lcd_Chr_cp(' ');
Lcd_Chr_cp(Ler_24c64('M'));
Lcd_Chr_cp(Ler_24c64('A'));
Lcd_Chr_cp(Ler_24c64('X'));
}
voidmsgAjustehoras(){
msgAjuste();
Lcd_Chr_cp(Ler_24c64('H'));
Lcd_Chr_cp(Ler_24c64('O'));
Lcd_Chr_cp(Ler_24c64('R'));
Lcd_Chr_cp(Ler_24c64('A'));
}
voidmsgAjustemes(){
msgAjuste();
Lcd_Chr_cp(Ler_24c64('M'));
Lcd_Chr_cp(Ler_24c64('E'));
Lcd_Chr_cp(Ler_24c64('S'));
}
voidmsgAjustedia(){
msgAjuste();
Lcd_Chr_cp(Ler_24c64('D'));
Lcd_Chr_cp(Ler_24c64('I'));
Lcd_Chr_cp(Ler_24c64('A'));
}
voidmsgAjusteano(){
msgAjuste();
Lcd_Chr_cp(Ler_24c64('A'));
}

```

```

Lcd_Chr_cp(Ler_24c64('N'));
Lcd_Chr_cp(Ler_24c64('O'));
}
voidmsgAjustehora(){
msgAjuste();
Lcd_Chr_cp(Ler_24c64('H'));
Lcd_Chr_cp(Ler_24c64('O'));
Lcd_Chr_cp(Ler_24c64('R'));
Lcd_Chr_cp(Ler_24c64('A'));
}
voidmsgAjusteminuto(){
msgAjuste();
Lcd_Chr_cp(Ler_24c64('M'));
Lcd_Chr_cp(Ler_24c64('I'));
Lcd_Chr_cp(Ler_24c64('N'));
}
void msgcnh1(){
Lcd_Chr(1,1,Ler_24c64('C'));
Lcd_Chr_cp(Ler_24c64('N'));
Lcd_Chr_cp(Ler_24c64('H'));
Lcd_Chr(2,1,Ler_24c64(308));
Lcd_Chr_CP(Ler_24c64(309));
Lcd_Chr_cp(Ler_24c64(310));
Lcd_Chr_cp(Ler_24c64(311));
Lcd_Chr_cp(Ler_24c64(312));
Lcd_Chr_cp(Ler_24c64(313));
Lcd_Chr_cp(Ler_24c64(314));
Lcd_Chr_cp(Ler_24c64(315));
Lcd_Chr_cp(Ler_24c64(316));
Lcd_Chr_cp(Ler_24c64(317));
Lcd_Chr_cp(Ler_24c64(318));
}
voidmsgcadastroCNH(){
Lcd_Chr(1,1,Ler_24c64('C'));
Lcd_Chr_cp(Ler_24c64('A'));
Lcd_Chr_cp(Ler_24c64('D'));
Lcd_Chr_CP(Ler_24c64('A'));
Lcd_Chr_cp(Ler_24c64('S'));
Lcd_Chr_cp(Ler_24c64('T'));
Lcd_Chr_cp(Ler_24c64('R'));
Lcd_Chr_cp(Ler_24c64('O'));
Lcd_Chr_cp(' ');
Lcd_Chr_cp(Ler_24c64('C'));
Lcd_Chr_cp(Ler_24c64('N'));
Lcd_Chr_cp(Ler_24c64('H'));
}

```

```

//=====menu.h=====
#include "calculos.h"
#include "imprimir.h"
char cadastrcnh(){
    bit ok;
    char cnh[]={48,48,48,48,48,48,48,48,48,48,48};
    char coluna=0;
    char caracter=48;
    char i;
    char motorista=1 ;
ok=1;
Lcd_Cmd(_LCD_CLEAR);
    msgcnh1();
    while(ok>0){
        if (Button(&PORTd, 7, 1, 0)) {
            if(motorista<2){
motorista++;
                }//if
            }//if
            if (Button(&PORTd, 4, 1, 0)) {
                if(motorista>1){
motorista--;
                    }//if
                }//if
                if(motorista==1) Lcd_Out(1,1,"MOTORISTA 1");
                if(motorista==2) Lcd_Out(1,1,"MOTORISTA 2");
Delay_ms(500);
                if(Button(&PORTd, 6, 1, 0))ok=0;
            }//while
            ok=1;
            while(ok>0){
                if(Button(&PORTd, 6, 1, 0)){
                    if(coluna<11)coluna++ ;
                    if(coluna>10) ok=0;
                }//if
                if(Button(&PORTd, 5, 1, 0)){
                    if(coluna>0){coluna--;}
                }//if
            }
            caracter=cnh[coluna];
            if (Button(&PORTd, 7, 1, 0)) {
                caracter= caracter+1;
                if(caracter>57) caracter=48;
            }//if
            if (Button(&PORTd, 4, 1, 0)) {
                if(caracter<49) caracter=58;
                caracter= caracter-1;
            }//if
            cnh[coluna]=Ler_24c64(caracter);
            Delay_ms(300);
            Lcd_Chr(2,coluna,' ');
            Delay_ms(50);

```

```

        for(i=0;i<11;i++)Lcd_Chr(2,i,cnh[i]);
    }//while
    for(i=0;i<11;i++){
if(motorista=1)Gravar_24c64((i+308),cnh[i]);
else Gravar_24c64((i+320),cnh[i]);
    } //for
    Lcd_Cmd(_LCD_CLEAR);
    return 0;
}//cadastro
void menus(char menu){
char texto[7];
    switch(menu){
        case 0:{
mostra_hora_data();
            break;
        }
        case 1:{
msgAjuste();
            if(Button(&PORTd, 6, 1, 0)){menu= ajustehoras();} // ds1307
            break;
        }
        case 2:{
msgcnh1();
            break;
        }
        case 3:{
msgcadastrocnh();
            if(Button(&PORTd, 6, 1, 0)){
                menu=0;
            }
            break;
        }
        case 4: {
msgmax();
            if(Button(&PORTd, 6, 1, 0)){

                IntToStr(velocidademaxima(),texto);
                Lcd_Out(2,1,"MAX ");
                Lcd_Out_CP(texto);
                };
            break;
        }
        case 5: {
Lcd_Out(1,1,"IMPRIMIR ");
            if(Button(&PORTd, 6, 1, 0)){
                imprimir();
            }
            break;
        }
    }
} //switch

```

```

} //menus

//=====================================================imprimir.h=====
void Enviar_Velocidade(int hora, int minutos, int aux minutos, int fim){
int valor = 220 + hora;
    UART1_Write(valor);
    for(; minutos < aux minutos; minutos++){
int end=0;
end=(hora*100)+minutos+400;
    UART1_Write(Ler_24c64(end));
Delay_ms(50);
    UART1_Write(Ler_24c64(end+2600));
Delay_ms(50);
    } //for
    if(fim == 0){
        UART1_Write(254);
    } else UART1_Write(249);
Delay_ms(1500);
} //envia velocidade
void imprimir(){
int i; char texto[7], rxdados, aux horas, horas, aux minutos, minutos, aux min, fim=0;
rxdados=30;
//=====================================================
Lcd_Out(2,1,"Id Veiculo");
    UART1_Write(255);
Delay_ms(10);
    for(i=338; i<342; i++){
        UART1_Write(Ler_24c64(i));
Delay_ms(10);
    }
    UART1_Write(254);
Delay_ms(1000);
Lcd_Out(2,1,"horas");
    UART1_Write(253);
Delay_ms(10);
    UART1_Write(Ler_24c64(303)); // hora
Delay_ms(10);
    UART1_Write(Ler_24c64(302)); // minuto
Delay_ms(10);
    UART1_Write(Ler_24c64(305)); // dia
Delay_ms(10);
    UART1_Write(Ler_24c64(306)); // mes
Delay_ms(10);
    UART1_Write(Ler_24c64(307)); // ano
Delay_ms(10);
    UART1_Write(254);
Delay_ms(1000);
//=====================================================
=====
Lcd_Out(2,1,"Enviando km");
    UART1_Write(252);

```



```
//=====cálculos.h=====
char velocidademaxima(){
    char velocidade=0,h,m,aux,texto[7];
    for(h=0;h<24;h++){
        for(m=0;m<60;m++){
            aux=Ler_24c64((h*100)+m+400);
            if((aux>velocidade)&&(aux<150))velocidade=aux;
            IntToStr(h,texto);
            Lcd_Out(2,1,texto);
        }//for
    }//for
    Lcd_Cmd(_LCD_CLEAR);
    return velocidade;
} //velocidademaxima
```

APÊNDICE E – Artigo final

Protótipo de Um Cronotacógrafo Digital

Prof. Esp. Valter Blauth Junior¹, Rodrigo Porto André²

¹Professor Especialista do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – Universidade do Extremo Sul Catarinense (UNESC) – Caixa Postal 3.167 – 88806-000 – Criciúma – SC - Brasil

²Acadêmico do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – Universidade do Extremo Sul Catarinense (UNESC) – Caixa Postal 3.167 – 88806-000 – Criciúma – SC - Brasil

valterblauth@unesc.net, rpaeletronica@hotmail.com

Abstract. For this work, a prototype chromotachograph was developed. In Brazil, the chromotachograph is mandatory for load and passengers vehicles with the purpose of reducing the number of accidents. In the prototype was approached the use of PIC and Atmel microcontrollers, I2c communication and serial TTL. The built-in device stores the speed, distance and time informations from the last 24 hours, allowing printing those informations through a thermal printer available on the device.

Resumo. Neste trabalho foi desenvolvido um protótipo de um cronotacógrafo. No Brasil, o uso do cronotacógrafo é obrigatório em veículos de transporte de cargas e passageiros, com o intuito de reduzir os números de acidentes. No protótipo foi abordado o uso da tecnologia dos microcontroladores PIC e Atmel, comunicação I2C e serial TTL. O cronotacógrafo construído armazena as informações de velocidade, distância e tempo de condução das últimas 24 horas permitindo a impressão dessas informações através de uma impressora térmica disponível no dispositivo.

1. Introdução

O Brasil é um país fortemente dependente do transporte rodoviário, seus veículos de carga e de transporte de passageiros desempenham um importante papel socioeconômico e uma posição estratégica na matriz de transportes do país. Desde 23 setembro de 1997 é obrigatório o uso de cronotacógrafo com o principal intuito de reduzir os números de acidentes. Este equipamento constitui-se de um único aparelho mecânico, eletrônico ou composto por um conjunto computadorizado.

Com o avanço rápido na tecnologia da computação e eletrônica, surgiram algumas formas de construir o cronotacógrafo digital, com custo reduzido, com maior número de funcionalidades, maior segurança na leitura de dados e identificação e controle da jornada de trabalho do motorista.

A utilização de microcontroladores trás vantagens: os circuitos ficam compactos, tem seu custo reduzido, permite o processamento de sinais analógicos ou digitais, possibilita a

alteração de funções do equipamento sem alteração de hardware e facilita a integração do circuito com computadores ou outros dispositivos (SOUZA, 2003).

No desenvolvimento do protótipo do cronotacógrafo digital foi utilizado um microcontrolador PIC para processar os dados, fazer leitura do sensor de rotação, efetuar a comunicação com o Liquid Crystal Display (LCD), memória, ds1307 e um microcontrolador Atmel para realizar a comunicação entre a mini impressora térmica e o microcontrolador PIC.

2. Justificativa

Atualmente, o Brasil possui uma frota de mais de 4 milhões de veículos de carga e de transporte de passageiros com o uso obrigatório do cronotacógrafo (BRASIL, 2015).

Na Europa, a principal forma de controle da jornada de trabalho dos condutores de veículos pesados é através do cronotacógrafo. Este dispositivo é montado por trás do velocímetro que registra velocidade, distância, tempo e modo de trabalho do veículo (ANDERSON, 1998, tradução nossa).

De acordo com o INMETRO (2015), a importância do uso do cronotacógrafo vem crescendo ao longo dos anos em consequência do número de acidentes fatais, envolvendo ônibus e caminhões. O instrumento inibe os excessos e ajuda a reduzir acidentes.

Os microcontroladores oferecem uma variada combinação de projetos altamente eficientes e integrados, possuem arquiteturas otimizadas para baixo consumo de energia e suporte avançado para interfaces, os microcontroladores permitem aos desenvolvedores criarem soluções completas para o mercado industrial, consumidor, de segurança, de comunicações, computação e automotivo (ATMEL, 2015).

O protótipo cronotacógrafo digital desenvolvido com microcontrolador, tem um circuito compacto, baixo consumo de energia e maior durabilidade. Com este equipamento será possível fazer a gestão da frota de forma mais eficaz e segura, controlar pontos como: jornada de trabalho, consumo de combustível (velocidade, distância percorrida, tempo) e uma maior segurança no trânsito.

3. Desenvolvimento do Cronotacógrafo

O protótipo do cronotacógrafo digital deve ser capaz de registrar a velocidade, distância e tempo de condução do veículo além da identificação do veículo e do condutor. O dispositivo possui um microcontrolador que interpreta o sinal do sensor e armazena em uma memória não volátil, por um período de 24 horas, que posteriormente poderão ser impressas.

3.1 Seleção dos Componentes

Para a seleção dos componentes para a construção do protótipo, em primeiro momento, realizou-se uma análise do funcionamento de um veículo Mercedes Benz, chassi OH-1628, assim, determinou-se que seria necessário a utilização de um sensor indutivo capaz de efetuar a contagem dos dentes de uma roda fônica e enviar pulsos elétricos para o protótipo.

Os cronotacógrafos digitais, segundo as normas do INMETRO, devem ter seu relógio independente do corte da alimentação do cronotacógrafo e o seu relógio interno, tem que se manter em funcionamento por um período superior a 5 anos. Após pesquisa, se optou pela utilização do DS1307, por ser de fácil aquisição e com grande quantidade de documentação

do mesmo. O DS1307 é um componente de Relógio de Tempo Real (RTC) com calendário completo, sendo capaz de fornecer informações como: segundo, minutos, dias, semana, mês e ano.

Outro componente necessário para o protótipo realizar o armazenamento dos dados de velocidade do veículo é a utilização de uma memória não volátil. Após o cálculo da quantidade necessária para o armazenamento dos dados do equipamento, chegou-se a conclusão que seria necessário um componente capaz de armazenar 2880 bytes. Sendo armazenado o valor da velocidade do veículo a cada um minuto, 2880 bytes serão capazes de armazenar 24 horas de informação de velocidade. Para essa finalidade foi utilizada a memória EEPROM 24c64 pelas seguintes razões: custo relativamente baixo, é um componente utilizado em vários equipamentos eletrônicos, é de fácil aquisição, possui capacidade de armazenamento de 64K bits, possui uma interface de comunicação I2C, retém os dados por um período maior que 10 anos, e conta com uma capacidade de 10000000 ciclos de escrita e leitura dos dados. Outra vantagem a ser destacada é que em um mesmo barramento de dados pode ser conectado 8 chips, e futuramente poderia comportar novas funcionalidade implementadas no cronotacógrafo digital.

Para a interface do protótipo foi empregado 4 micros interruptores e um LCD de 16 colunas por 2 linhas, com a finalidade de fazer a interação entre o usuário e o cronotacógrafo. O display opera com uma tensão de 5V, com comunicação paralela, possui o controlador HD44780, um dos mais utilizados na indústria de display LCDs. Sua principal vantagem é ser um display de baixo custo e alta eficiência.

O item escolhido para ser o responsável por fazer o processamento dos dados e fazer a comunicação dos componentes já citados, foi o microcontrolador PIC 16f887, por o acadêmico já possuir o mesmo. É um microcontrolador da família de 8 bits fabricado pela Microchip Technology, ele funciona com alimentação de 2V a 5,5V, possui 256 bytes de memória de dados EEPROM, com 5 conjuntos de portas de entrada e saída totalizando 33 portas. O PIC16f887 suporta trabalhar com temperaturas de - 40 a 125 C°, disponibiliza periféricos de comunicação UART, SPI e I2C, com dois temporizadores de 8 bits e um de 16 bits. Este é um microcontrolador que atende as necessidades do protótipo, por ter os periféricos UART para comunicação serial, o I2C para a comunicação da memória e RTC, o temporizador que será utilizado para receber os pulsos do sensor e com as portas digitais para fazer a comunicação com o display LCD e os botões.

Para a impressão das informações armazenadas no cronotacógrafo foi utilizada uma impressora térmica. A impressora adquirida para o projeto é uma mini impressora térmica modelo JP-QR701. A escolha deste modelo foi por ter um tamanho compacto, operar com tensão de 5V a 9V, com um consumo de 1,5A, ter uma comunicação serial TTL 5 volts, possibilitando a ligação direta ao microcontrolador. Por ser uma impressora térmica, não necessita de tinta ou tonner. A impressora utiliza bobina térmica com largura de 57 mm, a mesma utilizada em máquinas de cartão de crédito. Por a impressora não possuir uma biblioteca destinada para os microcontroladores PIC foi utilizado um microcontrolador Atmel para fazer a interface entre o PIC e a impressora, sendo que o Atmel já possui uma biblioteca destinada para a mini impressora térmica JP-QR701.

3.2 Construção do protótipo

Para o desenvolvimento do protótipo foi utilizada uma ferramenta de simulação e construção de circuitos eletrônicos, chamada Proteus, na versão demo. Com o uso do Proteus, foi desenvolvido o esquema eletrônico do protótipo.

O protótipo possui um microcontrolador PIC16F887 que faz o processamento do cronotacógrafo, para seu funcionamento é indispensável um circuito oscilador. No protótipo o oscilador é constituído de um cristal de 8 MHz e dois capacitores. O terminal 6 do PIC16F887 tem como função no protótipo receber o sinal do sensor indutivo. O sensor está acoplado na caixa de câmbio, este sensor envia pulsos elétricos para o microcontrolador que processa esses pulsos e transforma em informação de velocidade e distância percorrida. Quando o sensor é acionado, envia um pulso elétrico de 6 volts ao transistor BC547 que polariza fazendo com que o nível de tensão do pino 6 MCU fique em nível baixo, sendo possível que internamente a interrupção do timer 0 do microcontrolador seja disparado.

No protótipo do cronotacógrafo os dados de condução do veículo, velocidade e distância percorrida são armazenados em uma memória 24c64, a memória esta conectada ao microcontrolador nos pinos SCL e SDA, utilizando o protocolo de comunicação serial I2C.

O RTC é um circuito destinado para manter o controle do relógio do protótipo, o circuito DS1307 disponibiliza de uma bateria de 3V, que mantém o funcionamento do DS1307 quando o pino 8 está com nível de tensão abaixo de 4,5 volts. O DS1307 está conectado ao microcontrolador pelo mesmo barramento das memórias 24c64, protocolo I2C.

O protótipo utiliza um microcontrolador Atmega 328 que tem como papel no circuito fazer a comunicação do PIC16f887 à mini impressora térmica, ele possui um software internamente destinado a interpretar os dados enviado pelo PIC, estes dados são processados e enviados para a impressora, que por sua vez, imprime as informações do veículo, dados do condutor e velocidade registrada nas últimas 24 horas.

Um circuito com um LCD e com quatro botões fazem a interface entre usuário e protótipo cronotacógrafo, na figura 1 tem-se o protótipo montado.

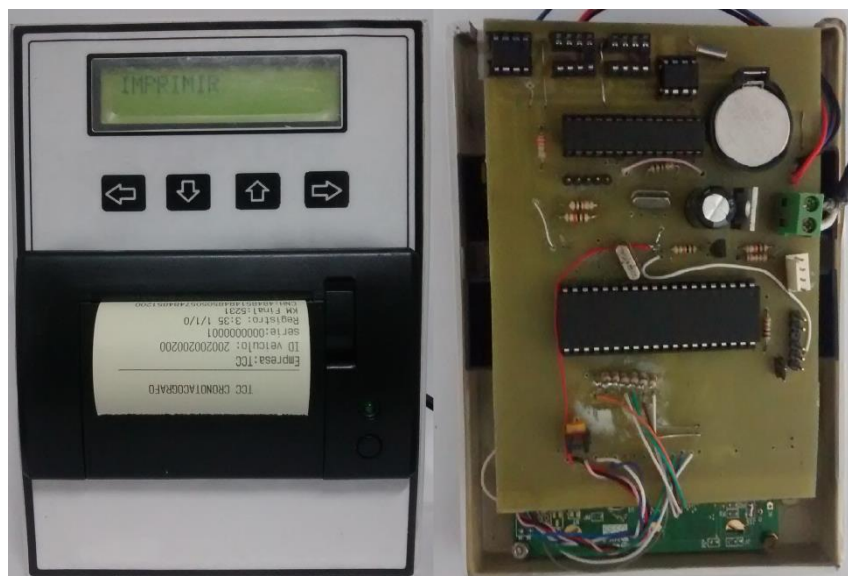


Figura 1. Protótipo cronotacógrafo digital

3.3 Desenvolvimento do software

No desenvolvimento do software para o microcontrolador PIC foi utilizado o MikroC em sua versão demo. Este disponibiliza várias bibliotecas destinadas a programação para microcontroladores, facilitando assim, o desenvolvimento.

Para o desenvolvimento do código se utilizou a lógica de armazenar todos os dados de velocidade, distância percorrida, registro do tempo percorrido e os dados do veículo em uma EEPROM 24c64.

Os pulsos enviados pelo sensor indutivo são transformados em informações de velocidade, através de uma função criada utilizando o periférico timer0 do microcontrolador, o timer conta os pulsos em um intervalo de 712 milissegundos. Este valor foi calculado a partir dos dados coletados no veículo Mercedes Benz, que a cada 1 segundo se recebe 28,09 pulsos, o veículo está com uma velocidade de 20 Km/h, sendo assim, se receber 20 pulsos em 712 milissegundos, o veículo está a uma velocidade de 20 Km/h. Com base nesse cálculo a quantidade de pulsos que se receber em um intervalo de 712 milissegundos será a velocidade que o veículo está em deslocamento. Com os pulsos já transformados em velocidades, eles são armazenados na EEPROM, o protótipo armazena os dados das últimas 24 horas, o endereço que a velocidade é armazenada representa a hora que foi processada a velocidade.

Para a impressão dos dados se utilizou uma impressora térmica. O PIC 16f887 está conectada a impressora através de um microcontrolador Atmega, que faz o processamento dos dados enviados pelo PIC para um protocolo de dados reconhecido pela impressora.

3.4 Teste do protótipo

Com o protótipo montado e os códigos desenvolvidos se iniciou os testes de funcionamento do protótipo. A avaliação do funcionamento do protótipo foi realizada em bancada, não foi possível realizar o teste no veículo Mercedes OH1628, pois o mesmo possui um sensor indutivo com lacres do INMETRO. Os primeiros testes foram referente as funções: ajuste de hora e data e funcionamento do display LCD e botões. Após a primeira etapa de teste, se realizou o teste do sensor indutivo, com o auxílio de um metal se observou as velocidades mostradas, logo se observou que seria inviável o teste de velocidade máxima do protótipo de uma forma manual, induzindo o funcionamento do sensor. Para realizar os testes sem o sensor, se construiu um simulador de sensor, em um primeiro momento, o simulador de sensor é construído com CI 555, o simulador de sensor oscila de 1 a 300 hz.

O segundo teste realizado foi conferir as velocidades mostradas se correspondiam aos pulsos que o simulador enviava. Nessa etapa, não foi encontrado problema de funcionamento de leitura e visualização da velocidade.

No terceiro teste foi realizado a avaliação da impressão, com o protótipo em funcionamento num período de 24 horas com uma velocidade fixa em 0 Km/h, se realizou o teste de impressão, nesse teste não se observou nenhuma irregularidade dos dados armazenados e impresso.

O quarto teste foi realizado com o funcionamento do protótipo em uma velocidade fixa de 90km/h, também não se observou nenhuma irregularidade dos dados.

No quinto teste foi realizado a simulação do funcionamento do protótipo com variação de velocidade. Para simular a variação de velocidade e o período de condução, foi construído outro simulador com Arduino Mega. Esse simulador construído possui uma função randômica que alterava a velocidade de funcionamento do protótipo. Após a impressão com o protótipo em funcionamento durante uma hora, sendo o primeiro teste com outro simulador de velocidade, se observou um problema nos dados impressos.

Para identificar onde estava sendo geradas as informações incorretas, utilizou um leitor de EEPROM, o fluxo de funcionamento do protótipo é dividido em: armazenar os dados na EEPROM e leitura da EEPROM. Para o armazenamento, os pulsos do sensor são transformados em informações, mostrados no display e são gravados na EEPROM. Para a impressão, os dados são lidos da EEPROM e enviados ao Atmel, que por sua vez, processa e envia a impressora.

Como a velocidade mostrada no display estava correta, se conclui que a etapa de leitura do sensor estava correta, com a leitura da EEPROM, se descobriu que os dados estavam sendo armazenados de uma forma incorreta. O armazenamento de velocidade da forma correta é do endereço 400 até o endereço 2759. Com a análise dos dados lidos na EEPROM com auxílio de leitor externo, se observou que os dados eram sobrescritos a cada 4 horas e 12 minutos. Com o teste realizado no código do microcontrolador PIC se identificou um problema na conversão do endereço da EEPROM na função de gravar_EEPROM, a função não estava gerando o endereço alto do endereço corretamente. Após correção do código, o protótipo armazenou e imprimiu da forma correta a fita de diagrama.

4. Resultados Obtidos

A construção do protótipo do cronotacógrafo digital atendeu as expectativas esperadas, o hardware construído apresentou seu funcionamento de uma forma estável e eficiente. É um circuito relativamente simples, que é possível armazenar os dados do veículo de uma forma simples, a placa do hardware já foi construída pensando na possibilidade de receber outros dispositivos, o microcontrolador Atmel possui 17 pinos que podem ser ligados outros periféricos, dois soquetes com o endereço pré-configurado para outras duas memórias 24cXX que já estão disponíveis no protótipo. Com outras memória é possível fazer o backup dos dados.

A comunicação I2C se mostrou eficiente para a comunicação da memória e o RTC com o microcontrolador, não se teve problema com hardware por interferência, alguns problemas encontrados foram decorrentes da construção do código.

A comunicação serial TTL 9600 bits por segundos entre microcontroladores teve seu funcionamento esperado. Os dados armazenados na EEPROM e enviados para o microcontrolador Atmel foram recebidos e impressos corretamente. Por o código não tratar de possíveis erros na comunicação, se houver alguma interferência na comunicação os microcontroladores pode entrar em loops infinitos.

A impressora térmica utilizada atendeu as expectativas de impressão, é um componente de pouca documentação de funcionamento, o pouco conhecimento adquirido para o uso da impressora foi através da análise da biblioteca da mesma. Com a construção do protótipo cronotacógrafo digital aliado ao hardware e software foram atingidos os resultados esperados referentes aos objetivos propostos para nível acadêmico. Para a utilização do protótipo como cronotacógrafo, há a necessidade de alteração de software e hardware.

5. Conclusão

A realização deste trabalho de conclusão de curso exigiu do autor estudos e levantamentos, a fim de conhecer todos os aspectos necessários dos cronotacógrafos disponíveis no mercado, para então, iniciar o planejamento e construção do protótipo. O trabalho cumpriu com os objetivos propostos de forma eficaz e ainda possibilitou a criação do protótipo com vantagens

relevantes como: o seu baixo custo de manutenção e a comodidade proporcionada ao condutor (por dispensar a troca do disco-diagrama).

Para a homologação junto ao INMETRO faz-se necessárias inúmeras adequações no projeto, o uso do protótipo não é proibido na utilização para controle de veículos com peso inferior a 4536 kg. Devida a impossibilidade de sua homologação junto ao INMETRO, a comparação como um produto final diante de outros modelos disponíveis no mercado tornou-se inviável.

No desenvolvimento do protótipo foi indispensável conhecimento técnico em eletrônica bem como na programação de microcontroladores. Durante a construção do hardware não foi encontrada qualquer dificuldade, por outro lado, em nível de software, algumas dificuldades foram detectadas.

O protótipo apresentou um funcionamento esperado. O dispositivo possibilita o cadastro do motorista, obter a maior velocidade registrada no período das 24 horas, ou a impressão de velocidades, tempo de operação e distância percorrida no período das 24 horas.

À partir da finalização deste projeto, pode-se observar as possibilidades de aprimoramentos para trabalhos futuros, por exemplo: a implantação de um leitor RFID ao hardware para uma fácil identificação do motorista e a implantação de um módulo de comunicação sem fio, para dispositivos smartphones, com o objetivo de receber os dados registrados no veículo junto a um software gestor que auxiliaria na análise de dados.

Referências

ANDERSON, Ross. **On the security of digital tachographs**. In: European Symposium on Research in Computer Security. Springer Berlin Heidelberg, 1998. p. 111-125.

ATMEL (Org.). **Tecnologias de microcontroladores comprovadas e inovadoras com baixo consumo de energia para as suas aplicações exclusivas**. Disponível em: <<http://www.atmel.com/pt/br/products/microcontrollers/default.aspx>>. Acesso em: 22 de maio 2015.

BRASIL. DENATRAN(Org.). **Frota de veículos**. Disponível em: <<http://www.denatran.gov.br/frota2015.htm>>. Acesso em: 22 de maio 2015.

INMETRO. **O que é cronotacógrafo**: História do cronotacógrafo. Disponível em: <<http://cronotacografo.rbmlq.gov.br/o-que-e-cronotacografo>>. Acesso em: 20 maio 2015.

SOUZA, David. **Desbravando o PIC**: Ampliado e atualizado para PIC16F628A. 8. ed. São Paulo: Editora Érica, 2003.