



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

CCET- CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

ENGENHARIA DE COMPUTAÇÃO

Tacômetro

CURITIBA

2012

Marcelo Emanuel Melani Camati

Renan Nepomoceno Pinto

Gustavo Raphael Stein

Conta-Giros

Projeto apresentado como requisito de avaliação parcial do programa de aprendizado em Microprocessadores I, do curso de Engenharia de Computação da Pontifícia Universidade Católica do Paraná.

Orientador: Prof^o. Afonso Ferreira Miguel.

CURITIBA

2012

RESUMO

O projeto Tacômetro, referente ao quinto período da Engenharia da Computação da Pontifícia Universidade Católica do Paraná, consiste no desenvolvimento de um tacômetro digital para carros, em que funciona baseado no número de vezes em que a bobina é acionada, assim contando a cada instante a quantidade de acionamentos da bobina, mostrando para o usuário o número de rotações por minuto (RPM) do seu veículo naquele dado momento.

Palavra-chave: Conta-Giros, Carro, bobina, vela, Tacômetro.

ABSTRACT

The Project Tacômetro, made in fifth period of the course of Computer Engineering at the Pontifícia Universidade Católica of Paraná, is the development of a simple and digital dynamometer for cars, its works based on the number of times the ignition coil is triggered in a certain time, thus showing to the user the rpm at that specific time.

Keywords: Dynamometer, coil, car, spark plug

1. Sumário

2.	Introdução.....	6
3.	Objetivo.....	8
4.	Não está incluso no escopo deste projeto.....	9
5.	Módulos	10
6.	Funcionamento eletrônico.....	11
7.	Os Resultados.....	14
8.	A equipe de desenvolvimento	15
9.	Conclusão.....	18
10.	Anexos.....	20
11.	Referências Bibliográficas	78

2. Introdução

A disciplina de Micro-processadores do curso de Engenharia de Computação tem como intuito desenvolvimento de projetos, desde a documentação completa, organogramas, cronogramas, apresentações e a conclusão do projeto nas mais corretas formas, além do trabalho em grupo. Capacitando o aluno a gerenciar um projeto para quando entrar no mercado de trabalho não ter tantas dificuldades em realizar qualquer tipo do mesmo. A intenção do desenvolvimento deste projeto, também é por em prática o conhecimento adquirido acerca de eletrônica digital e analógica, micros controladores e micros processadores. O projeto Tacômetro é um dispositivo capaz de contar pulsos elétricos e mostrar em um display de sete segmentos o valor contado em um determinado instante. Os pulsos são captados diretamente do pino w da bobina, o que facilita muito o desenvolvimento do projeto, pois basta contarmos quantos pulsos que são emitidos em um minuto e mostrar no display. O projeto também conta com um simulador de bobina, no caso um dispositivo para gerar pulsos similares a de uma bobina, facilitando testes e possibilitando a apresentação final do projeto. O grupo formado para o desenvolvimento do projeto do quinto período do curso de Engenharia de Computação é formado pelos seguintes integrantes: Gustavo Stein, Marcelo Camatie Renan N. Pinto.

1.1 Justificativas

O Tacômetro será desenvolvido para que possamos interagir com os conceitos de eletrônica digital, eletrônica analógica, micro controlador e micro processadores dentro

dos conceitos da engenharia de computação. Em que pese o projeto tenha fins didáticos o sistema a ser desenvolvido possuirá uma grande quantidade de vantagens aos demais disponíveis no mercado, dentre elas estão o consumo reduzido e a grande precisão.

1.1 Metodologia

O projeto é um sistema embarcado, ou seja, micro-controlado sem que haja necessidade de um computador. Usamos um micro controlador AT89s52 baseado na arquitetura do clássico 8051 da Intel. O projeto conta com um software desenvolvido em linguagem assembly. Também contamos com aparelhos de instrumentação básica como: multi-teste, osciloscópios, gerador de funções, protoboard, estação de solda, fluxo e estanho e componentes eletrônicos como: resistores, capacitores, indutores, transistores, placas de fenolite entre outros. O primeiro item a ser posto em prática será o desenvolvimento e implementação da placa de circuitos onde estará o micro controlador, esta placa conterá quatro display de sete segmentos, o micro controlador at89s52, um módulo de conversão de onda senoidal para onda quadrada e o módulo de entrada de dados. O segundo item será o simulador sinal regulável através de um potenciômetro, este circuito simulador conta com um CI chamado 555.

1.2 As Responsabilidades

Para que o projeto obtivesse tamanho sucesso em seu desenvolvimento foi necessária a participação ativa de todos os participantes do grupo e também dos professores, é necessário muita responsabilidade, seriedade e muita força de vontade em todos os eixos do grupo para que o projeto fosse bem desenvolvido. Cada integrante teve a sua responsabilidade e cumpriu com o máximo de comprometimento para com ele. Os professores estavam aptos a responder todas as nossas dúvidas em relação ao projeto, e nos ajudar, dar novas idéias e apoio. E também dependemos das estruturas da PUC, que se tornou a principal responsabilidade, pois são nos laboratórios que conseguimos montar projeto e testá-lo, levando o projeto adiante.

3. Objetivo

O projeto Tacômetro tem como objetivos, “ensinar” o grupo a fazer pesquisas, documentações, cronogramas, apresentações, ou seja, tudo que envolve um bom gerenciamento de projetos, requisitos desse projeto, assim como utilizar teorias usadas em sala de aula já em aplicações mais próximas da engenharia propriamente dita, como a utilização da disciplina de Micro controladores em Engenharia, mecânica em geral, circuitos elétrico, micro processador e sistemas digitais para uma compreensão melhor da eletrônica analógica e digital e das tecnologias que serão utilizadas.

2.1-Geral

Com base nos programas de aprendizagem de Eletrônica I e Microcontroladores I, criar um projeto que utilize os conceitos aprendidos em sala de aula colocando-os em pratica.

2.2- Específico

- Estudar e aplicar pulsos quadráticos.
- Estudar o funcionamento de micro controlador e aplicá-lo na prática
- Utilizar a linguagem Assembly para hardware poder fazer a contagem dos pulsos e imprimi-los no display
- Construir uma maquete para o conta-giros ilustrando seu funcionamento

4. Não está incluso no escopo deste projeto

O Projeto tacômetro poderá receber futuros *upgrades*, que não estarão disponíveis na versão 1.0. Dentre estes *upgrades* está à opção de fazer leitura para veículos que atinjam mais de 9999 RPM, opção de tacômetro com shift light regulável, a implementação de uma bateria de suporte que manteria o sistema ativo (caso a alimentação fosse cortada), opção de gerenciamento de registros para o shift light.

3.1 Materiais Utilizados

O projeto conta com uma placa de circuito impresso de fenolite, um micro controlador AT89S52 baseado na arquitetura do processador 8051 da Intel, um shimetrigger, um display quádruplo de sete segmentos, um CI 555, potenciômetro, resistores, diodos, zeners, regulador de tensão, capacitores, pushbottons, leds e transistores.

5. Módulos

4.1 Alimentação

A alimentação do projeto será feita através de baterias, o módulo principal utilizara a própria bateria do veiculo 12V, e o módulo simulador deverá ser alimentado por pilhas.

4.2 Processador

O processamento de dados será feito através do micro controlador AT89S52, que utilizará um software desenvolvido em linguagem assembly.

4.4 Sensor

O módulo sensor contará com um circuito regulador de tensão e um shime trigger para que a onda senoidal da bobina seja convertida em onda quadrada, facilitando os cálculos que serão efetuados pelo processador.

4.5 Painel

O painel conta com quatro displays de led de sete segmentos, onde a informação virá diretamente de uma das portas do micro controlador. O micro controlador também utiliza outros quatro bits para chavear os quatro displays com ajuda de quatro transistores.

4.6 Software

O software será desenvolvido em linguagem assembly, este software deverá controlar as saídas para o display, e efetuar cálculos de largura de pulso e enviar o valor correto para o display.

6. Funcionamento eletrônico

O circuito já alimentado tem seu início no módulo sensor que irá receber os pulsos da bobina, o sinal passa através de um circuito regulador de tensão que conta

com um zener. O sinal passa pelo shime trigger e então a onda que era senoidal até então, passa a ser quadrada. O sinal entra pela porta int0 do micro controlador, neste momento o controlador medira a largura dos pulsos, e após uma série de cálculos matemáticos o micro controlador deverá enviar o valor correto para os displays que serão chaveados pelo próprio micro controlador.

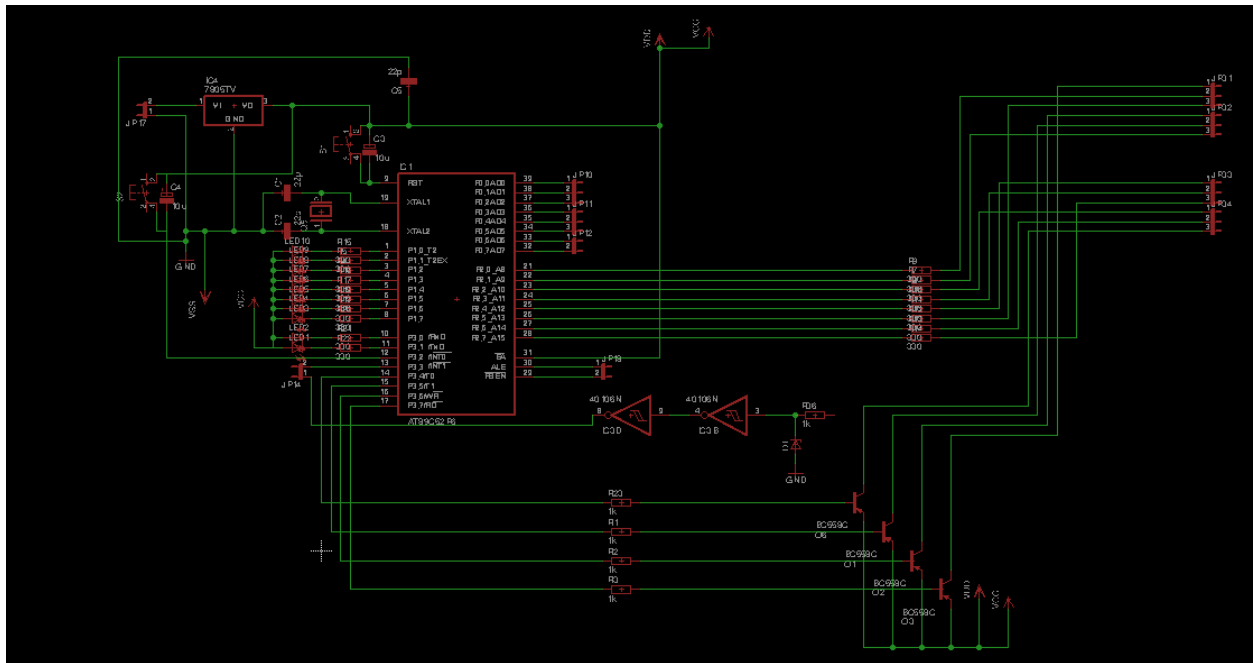


Figura 01: Esquema elétrico

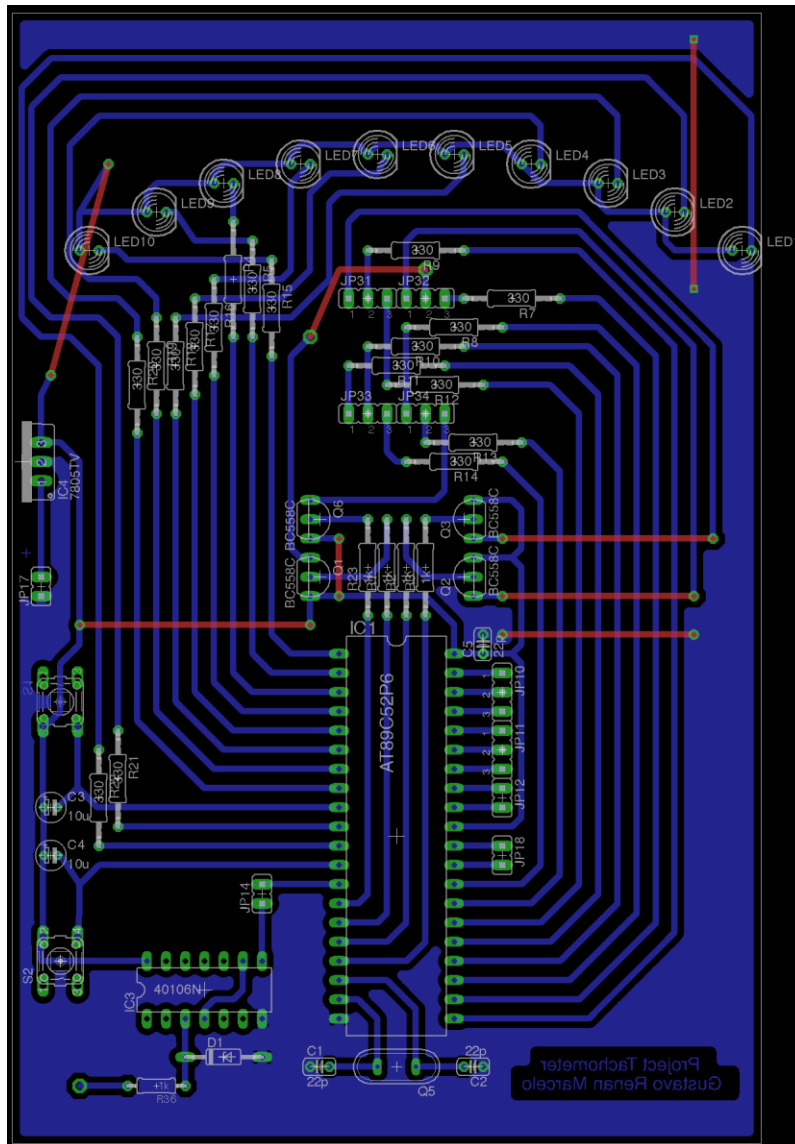


Figura 2: Layout da PCI

7. Os Resultados

Como resultados deste projeto, serão apresentados ao professor os seguintes itens / funcionalidades:

1. Protótipo funcionando dos Módulos;

- Placa circuito impresso;

- Alimentação;

- Processador;

- Sensor;

- Software;

2. Software do Microcontrolador AT89S52;

3. CD com arquivos, fotos, desenhos, códigos-fonte, esquemáticos, diagramas e modelos dos módulos implementados;

4. Vídeo do funcionamento;

5. Documentação do projeto dos itens acima.

6. Mostra do funcionamento aos professores e colegas.

8. A equipe de desenvolvimento

Renan Nepomoceno Pinto.

Responsabilidades: Acompanhar e propor soluções às dificuldades do projeto. Deverá ainda auxiliar os demais nas atividades para cumprir o cronograma e o plano de trabalho. Igualmente, executará tarefas pedidas pelo conselho deliberativo (Professor Afonso Miguel e o cliente externo). Finalmente, realizará atividades mediante pedido dos superiores imediatos (Professores Afonso Miguel).

Marcelo Camati.

Responsabilidades: Gerenciar os processos de qualidade, selecionar fornecedores, acompanhar processos de acabamento e desenvolvimento de novas técnicas e ferramentas. Igualmente, deverá auxiliar os demais nas atividades para o cumprimento do cronograma e do plano de trabalho. Por fim, executar atividades compatíveis mediante pedido do superior imediato.

Gustavo Stein.

Responsabilidades: Gerenciar os processos de tecnologia da informação do projeto, verificar estado do parque de máquinas (hardware), pesquisar novas ferramentas (softwares e hardwares) para a diminuição de custos do projeto e prestar suporte a equipe. De igual modo, auxiliará os demais nas atividades para cumprir o

cronograma e o plano de trabalho. Por fim, executar atividades compatíveis mediante o pedido do superior imediato.

Designação de Atividades.

Atividade

Colaborador

Anteprojeto

Estado da arte	Marcelo Camati
Levantamento de alternativas	Renan N. Pinto
Seleção das alternativas	Gustavo Stein
Definição do escopo	Marcelo Camati
Recursos necessários	Renan N. Pinto
Custos	Gustavo Stein

Projeto

Teste do Módulo de Alimentação

Aquisição de componentes	Marcelo Camati
Pesquisa e desenvolvimento	Gustavo Stein
Teste Módulo Finalizado	Renan N. Pinto

Teste do Módulo Software

Aquisição de componentes	Renan N. Pinto
--------------------------------	----------------

Pesquisa e desenvolvimentoGustavo Stein
Teste Módulo FinalizadoMarcelo Camati

Teste do Módulo Sensor

Aquisição de componentesRenan N. Pinto
Pesquisa e desenvolvimentoGustavo Stein
Teste Módulo FinalizadoMarcelo Camati

Teste do Módulo Processador

Aquisição de componentesRenan N. Pinto
Pesquisa e desenvolvimentoGustavo Stein
Teste Módulo FinalizadoMarcelo Camati

Teste do Módulo Painel

Aquisição de componentesRenan N. Pinto
Pesquisa e desenvolvimentoGustavo Stein
Teste Módulo FinalizadoMarcelo Camati

Teste Maquete

Aquisição de componentesRenan N. Pinto
Pesquisa e desenvolvimentoGustavo Stein
Teste Módulo FinalizadoMarcelo Camati
ImplementaçãoGustavo Stein

Documentação	Marcelo Camati
Apresentação	Gustavo Stein
Ajustes Finais	Renan N. Pinto
Acompanhamento	Renan N. Pinto
Conclusão do projeto.....	Marcelo Camati, Gustavo Stein, Renan N. Pinto.

Vale ressaltar que embora as atividades estejam separadas individualmente, todos estarão envolvidos para o cumprimento destas, a separação apenas indica o gerente da atividade.

9. Conclusão

Com a finalização deste projeto, que durou cerca de cinco meses, podemos dizer com convicção que conseguimos atender todas as metas levantadas no início do mesmo. O projeto tendo como objetivo principal a integração das diversas disciplinas do curso, relacionando assim teoria com prática. Pode-se dizer que o objetivo do mesmo foi alcançado com êxito no projeto. Apesar dos problemas que tivemos no início do projeto em decidir que tipo de projeto faríamos, no final acabou dando certo e optamos por igual em fazer algo que tem um valor no mercado. Alguns acontecimentos devem ser ressaltados, estes que fizeram muita diferença na execução e conclusão do projeto. O Projeto Tacômetro se destaca por ser um projeto acadêmico que é inspirado nos sistemas de conta giros existentes, porém com a tecnologia que vai crescendo e

com as mudanças nos padrões atuais, esse mesmo projeto acadêmico pode ter várias melhorias e competir de igual a igual com sistemas de alto nível, pois a idéia basicamente é a mesma, tem-se um micro controlador que controla todas as funcionalidades do sistema. Durante todo o projeto contamos com a ajuda direta do Prof. Afonso Ferreira Miguel, que nos apoiaram, nos ajudaram e sempre estiveram dispostos a tirar nossas dúvidas, sejam elas até em momentos importunos. Os Professores sempre tiveram a parte de todos os acontecimentos durante o projeto, tenham sido eles bons ou ruins, sempre nos orientando a qual caminho deveríamos tomar para obter êxito em tal quesito. Contamos também com o apoio de outros Professores, amigos, familiares e funcionários da PUCPR que sempre nos deram um apoio e incentivo para continuar sempre disposto a atingir os nossos objetivos. O trabalho em grupo foi de suma importância, pois para avançarmos em uma etapa, fazia-se necessário a conclusão de outra etapa, que muitas vezes eram desempenhadas por somente um membro do grupo, gerando assim um senso de responsabilidade e comprometimento maior para com o assunto que estava tratando. O projeto serviu para vermos os vários conceitos passados em sala de aula na prática, pois o conceito passa apenas uma noção básica sobre o assunto, já na prática é outra coisa, deve-se tomar cuidado com isso e aquilo, tem que pensar na interferência que um pode ocasionar no outro, entre outras situações mais que devem sempre ser levadas em conta, onde que muitas vezes nem pensávamos que era assim. Ficamos imensamente felizes em termos feito esse trabalho e esperamos pelos próximos que ainda estão por vir.

10. Anexos

9.1 Anexos I– Dicionário de termos técnicos

Para facilitar a leitura para quem possa a vir a fazer a leitura do mesmo, foifeito este pequeno dicionário básico contendo alguns termos usados neste projeto. Este dicionário serve apenas para dar uma noção básica sobre do que se trata cada palavra em específico.

Capacitor

Um capacitor ou condensador é um componente que armazena energia num campo elétrico, acumulando um desequilíbrio interno de carga elétrica.

Os formatos típicos consistem em dois eletrodos ou placas que armazenam cargas opostas. Estas duas placas são condutoras e são separadas por um isolante ou por um dielétrico. A carga é armazenada na superfície das placas, no limite com o dielétrico. Devido ao fato de cada placa armazenar cargas iguais, porém opostas, a carga total no dispositivo é sempre zero.

Capacitância

A propriedade que estes dispositivos têm de armazenar energia elétrica sob a forma de um campo eletrostático é chamada de capacitância ou capacidade (C) e

é medida pelo quociente da quantidade de carga (Q) armazenada pela diferença de potencial ou tensão (V) que existe entre as placas:

$$C = \frac{Q}{V}$$

Pelo Sistema Internacional de Unidades (SI), um capacitor tem a capacitância de um Farad (F) quando um Coulomb de carga causa uma diferença de potencial de um volt (V) entre as placas. O farad é uma unidade de medida considerada muito grande para circuitos práticos, por isso, são utilizados valores de capacitâncias expressos em microFarads (μF), nanoFarads (nF) ou picoFarads (pF). A equação acima é exata somente para valores de Q muito maiores que a carga do elétron ($e = 1,602 \times 10^{-19}\text{C}$). Por exemplo, se uma capacitância de 1pF fosse carregada a uma tensão de 1 μV , a equação perderia uma carga $Q = 10^{-19}\text{C}$, mas isto seria impossível já que seria menor do que a carga em um único elétron. Entretanto, as experiências e as teorias recentes sugerem a existência de cargas fracionárias. A capacitância de um capacitor de placas paralelas constituído de dois eletrodos planos idênticos de área A separados à distância constante d é aproximadamente igual a:

$$C = \epsilon_0 \epsilon_r \frac{A}{d}$$

Onde C é a capacitância em Farads ϵ_0 é a permissividade eletrostática do vácuo ou espaço livre.

Energia

A energia (no SI, medida em Joules) armazenada em um capacitor é igual ao trabalho feito para carregá-lo. Considere um capacitor com capacitância C , com uma carga $+q$ em uma placa e $-q$ na outra. Movendo um pequeno elemento de carga q de uma placa para a outra contra a diferença de potencial $V = q/C$ necessita de um trabalho dW :

$$dW = \frac{q}{C}dq$$

Nós podemos descobrir a energia armazenada em um capacitor integrando essa equação. Começando com um capacitor descarregado ($q=0$) e movendo carga de uma placa para a outra até que as placas tenham carga $+Q$ e $-Q$, necessita de um trabalho W :

$$W_{\text{carregando}} = \int_0^Q \frac{q}{C}dq = \frac{1}{2} \frac{Q^2}{C} = \frac{1}{2} CV^2 = E_{\text{armazenada}}$$

Capacitores Comuns

Apresenta-se com tolerâncias de 5 % ou 10 %. Capacitores são freqüentemente classificados de acordo com o material usados como dielétrico. Os seguintes tipos de dielétricos são usados: **cerâmica** (valores baixos até cerca de $1\mu\text{F}$) C0G or NP0 - tipicamente de $4,7 \text{ pF}$ a $0,047 \text{ uF}$, 5 %. Alta tolerância e performance de temperatura. Maiores e mais caros X7R - tipicamente de 3300 pF a $0,33 \text{ uF}$, 10 %. Bom para

acoplamento não-crítico, aplicações com timer. Z5U - tipicamente de 0,01 uF a 2,2 uF, 20 %. Bom para aplicações em bypass ou acoplamentos. Baixo preço e tamanho pequeno.

poliestireno(geralmente na escala de picofarads).

poliéster(de aproximadamente 1 nF até 1000000 µF).

polipropileno(baixa perda. alta tensão, resistente a avarias).

tântalo(compacto, dispositivo de baixa tensão, de até 100 µF aproximadamente).

eletrolítico(de alta potência, compacto mas com muita perda, na escala de 1 µF a 1000 µF)

Propriedades importantes dos capacitores, além de sua capacitância, são a máxima tensão de trabalho e a quantidade de energia perdida no dielétrico. Para capacitores de alta potência a corrente máxima e a Resistência em Série Equivalente (ESR) são considerações posteriores. Um ESR típico para a maioria dos capacitores está entre 0,0001 ohm e 0,01 ohm, valores baixos preferidos para aplicações de correntes altas. Já que capacitores têm ESRs tão baixos, eles têm a capacidade de entregar correntes enormes em circuitos curtos, o que pode ser perigoso. Por segurança, todos os capacitores grandes deveriam ser descarregados antes do

manuseio. Isso é feito colocando-se um resistor pequeno de 1 ohm a 10 ohm nos terminais, isso é, criando um circuito entre os terminais, passando pelo resistor.

Capacitores também podem ser fabricados em aparelhos de circuitos integrados de semicondutores, usando linhas metálicas e isolantes num substrato. Tais capacitores são usados para armazenar sinais analógicos em filtros chaveados por capacitores, e para armazenar dados digitais em memória dinâmica de acesso aleatória (DRAM).

Diferentemente de capacitores discretos, porém, na maior parte do processo de fabricação, tolerâncias precisas não são possíveis (15 % a 20 % é considerado bom).

Corrente elétrica

Na Física, corrente elétrica é o fluxo ordenado de partículas portadoras de carga elétrica. Sabe-se que, microscopicamente, as cargas livres estão em movimento aleatório devido a agitação térmica. Apesar desse movimento desordenado, ao estabelecermos um campo elétrico na região das cargas, verificasse um movimento ordenado que se apresenta superposto ao primeiro. Esse movimento recebe o nome de movimento de deriva das cargas livres. Raios são exemplos de corrente elétrica, bem como o vento solar, porém a mais conhecida, provavelmente, é a do fluxo de elétrons através de um condutor elétrico, geralmente metálico. O símbolo convencional para representar a intensidade de corrente elétrica (ou seja, a quantidade de carga Q que flui por unidade de tempo t) é o I , original do alemão *Intensität*, que significa intensidade.

$$I = \frac{\Delta Q}{\Delta t}$$

A unidade padrão no SI para medida de intensidade de corrente é o ampère. A corrente elétrica é também chamada informalmente de amperagem. Embora seja um termo válido, alguns engenheiros repudiam o seu uso.

Circuito integrado

Um circuito integrado, também conhecido por chip, é um dispositivomicroeletrônico que consiste de muitos transistores e outros componentesinterligados capazes de desempenhar muitas funções. Suas dimensões sãoextremamente reduzidas, os componentes são formados em pastilhas de materialsemicondutor.

A importância da integração está no baixo custo e alto desempenho, além dotamanho reduzido dos circuitos aliado à alta confiabilidade e estabilidade defuncionamento. Uma vez que os componentes são formados ao invés de montados,a resistência mecânica destes permitiu montagens cada vez mais robustas a choques e impactos mecânicos, permitindo a concepção de portabilidade dosdispositivos eletrônicos.

No circuito integrado completo ficam presentes os transistores, condutores deinterligação, componentes de polarização, e as camadas e regiões isolantes oucondutoras obedecendo ao seu projeto de arquitetura.No processo de formação do chip, é fundamental que todos os componentessejam implantados nas regiões apropriadas da pastilha. É necessário que a isolaçãoseja perfeita, quando for o caso.

Isto é obtido por um processo chamado difusão, que se dá entre os componentes formados e as camadas com o material dopado com fósforo, e separadas por um material dopado com boro, e assim por diante. Após sucessivas interconexões, por boro e fósforo, os componentes formados ainda são interconectados externamente por uma camada extremamente fina de alumínio, depositada sobre a superfície e isolada por uma camada de dióxido de silício.

Diodo

Diodo semicondutor é um dispositivo ou componente eletrônico composto de cristal semicondutor de silício ou germânio numa película cristalina cujas faces opostas são dopadas por diferentes gases durante sua formação. É o tipo mais simples de componente eletrônico semicondutor, usado como retificador de corrente elétrica. A dopagem no diodo é feita pela introdução de elementos dentro de cristais tetraivalentes, normalmente feitos de silício e germânio.

Dopando esses cristais com elementos trivalentes, obterá átomos com sete elétrons na camada de valência, que necessitam de mais um elétron para a neutralização (cristal P). Para a formação do cristal P, utiliza-se principalmente o elemento Índio. Dopando os cristais tetraivalentes com elementos pentavalentes, obter-se-á átomos neutralizados (com oito elétrons na camada de valência) e um elétron excedente (cristal N). Para a formação do cristal N, utiliza-se principalmente o elemento Fósforo. Quanto maior a intensidade da dopagem, maior será a condutibilidade dos cristais, pois suas estruturas apresentarão um número maior de portadores livres

(lacunas e elétrons livres) e poucas impurezas que impedem a condução da corrente elétrica. Outro fator que influencia na condução desses materiais é a temperatura. Quanto maior for sua temperatura, maior será a condutibilidade pelo fato de que a energia térmica tem a capacidade de quebrar algumas ligações covalentes da estrutura, acarretando no aparecimento de mais portadores livres para a condução de corrente elétrica.

Depois de dopadas, cada face dos dois tipos de cristais (P e N) terá uma determinada característica diferente da oposta, gerando regiões de condução no cristal, uma com excesso de elétrons, outra com falta destes (lacunas), e entre ambas, haverá uma região de equilíbrio por recombinação de cargas positivas e negativas, chamada de região de depleção (à qual possui uma barreira de potencial).

Micro controlador

Um microcontrolador (também denominado MCU ou μC) é um computador num chip, contendo um processador, memória e funções de entrada/saída. É um microprocessador que enfatiza a alta integração, em contraste com os microprocessadores de uso geral (do tipo usado em computadores pessoais). Além dos componentes lógicos e aritméticos usuais dum microprocessador de uso geral, o Microcontrolador integra elementos adicionais tais como memória RAM, EEPROM ou Memória flash para armazenamento de dados ou programas, dispositivos periféricos e interfaces de E/S que podem ir de um simples pino digital do componente a uma interface USB ou Ethernet nos mais avançados (como o ARM LPC2368).

Com frequências de clock de poucos MHz ou ainda mais baixas microcontroladores são considerados lentos se comparados aos microprocessadores modernos, mas isso é perfeitamente adequado para aplicações típicas. Eles consomem relativamente pouca energia (miliwatts), e geralmente possuem a capacidade de "hibernar" enquanto aguardam que aconteça algum evento interessante provocado por um periférico, tal como o pressionar dum botão, que os colocam novamente em atividade. O consumo de energia enquanto estão "hibernando" pode ser de nanowatts, tornando-os ideais para aplicações de baixa energia e que economizem bateria.

De forma oposta aos microprocessadores, onde se super dimensiona ao máximo tendo como limite o preço que o usuário deseja investir, a escolha do Microcontrolador é feita pelo projetista do equipamento. É erro de projeto superdimensionar. Cada desperdício será multiplicado pelo número de equipamentos fabricados (às vezes milhões). Por isso existem duas linhas de pesquisa paralelas, mas opostas uma criando microcontroladores mais capazes, para atender produtos de mais tecnologia como os novos celulares ou receptores de TV digital e outra para criar microcontroladores mais simples e baratos, para aplicações elementares (como um chaveiro que emite sons).

De forma diferente da programação para microprocessadores, que em geral contam com um sistema operacional e um BIOS, o programador ou projetista que desenvolve sistemas com microcontroladores tem que lidar com uma gama muito grande de desafios, fazendo muitas vezes todo o processo construtivo do aparelho: BIOS, firmware e circuitos.

Microprocessador

Todos os computadores pessoais e um número crescente de equipamentos eletrônicos baseiam-se num tipo especial de circuito eletrônico chamado de microprocessador.

O microprocessador moderno é um circuito integrado formado por uma camada chamada de mesa epitaxial de silício, trabalhada de modo a formar um cristal de extrema pureza, laminada até uma espessura mínima com grande precisão, depois cuidadosamente mascarada por um processo fotográfico e dopada pela exposição a altas temperaturas em fornos que contêm misturas gasosas de impurezas. Este processo é repetido tantas vezes quanto necessário à formação da micro arquitetura do componente. Responsável pela execução das instruções num sistema, o microprocessador, escolhido entre os disponíveis no mercado, determina, em certa medida a capacidade de processamento do computador e também o conjunto primário de instruções que ele compreende. O sistema operativo é construído sobre este conjunto.

O próprio microprocessador subdivide-se em várias unidades, trabalhando em altas frequências. A ALU (Arithmetic and Logical Unit), unidade responsável pelos cálculos aritméticos e lógicos e os registradores são parte integrante do microprocessador na família x86, por exemplo.

Embora seja a essência do computador, o microprocessador diferente do

Microcontrolador está longe de ser um computador completo. Para que possa interagir com o utilizador precisa de: Memória, E/S Entradas/Saídas, um clock, controladores e conversores de sinais entre outros. Cada um desses circuitos de apoio interage de modo peculiar com os programas e, dessa forma, ajuda a moldar o funcionamento do computador.

Resistor

Um resistor (chamado de resistência em alguns casos) é um dispositivo elétrico muito utilizado em eletrônica, com a finalidade de transformar energia elétrica em energia térmica (efeito joule), a partir do material empregado, que pode ser por exemplo carbono.

Um resistor ideal é um componente com uma resistência elétrica que permanece constante independentemente da tensão ou corrente elétrica que circular pelo dispositivo. Os resistores podem ser fixos ou variáveis. Neste caso são chamados de potenciômetros ou reostatos. O valor nominal é alterado ao girar um eixo ou deslizar uma alavanca. O valor de um resistor de carbono pode ser facilmente determinado de acordo com as cores que apresenta na cápsula que envolve o material resistivo, ou então usando um ohmímetro. Alguns resistores são longos e finos, com o material resistivo colocado ao centro, e um terminal de metal ligada em cada extremidade. Este tipo de encapsulamento é chamado de encapsulamento axial. A fotografia a direita mostra os resistores em uma tira geralmente usados para a pré-formatação dos terminais. Resistores usados em computadores e outros dispositivos

são tipicamente muito menores, freqüentemente são utilizadas tecnologia de montagem superficial (Surface-mount technology), ou SMT, esse tipo de resistor não tem perna de metal. Resistores de potência maior são feitos mais robustos para dissipar calor de maneira mais eficiente, mas eles seguem basicamente a mesma estrutura.

Os resistores são sim como parte de um circuito elétrico e incorporados dentro de dispositivos microeletrônicos ou semicondutores. A medição crítica de um resistor é a resistência, que serve como relação de voltagem para corrente é medida em ohms, uma unidade SI. Um componente tem uma resistência de 1 ohm se uma voltagem de 1 volt no componente fazer com que percorra, pelo mesmo, uma corrente de 1 Ampère, o que é equivalente à circulação de 1 Coulomb de carga elétrica, aproximadamente 6.241506×10^{18} elétrons por segundo. Qualquer objeto físico, de qualquer material é um tipo de resistor. A maioria dos metais são materiais condutores, e opõe baixa resistência ao fluxo de corrente elétrica. O corpo humano, um pedaço de plástico, ou mesmo o vácuo têm uma resistência que pode ser mensurada. Materiais que possuem resistência muito alta são chamados isolantes ou isoladores

A relação entre tensão, corrente e resistência, através de um objeto é dada por uma simples equação, Lei de Ohm:

$$R = \frac{V}{I}$$

Onde V é a voltagem em volts, I é a corrente que circula através de um objeto em Ampères, e R é a resistência em ohms. Se V e I tiverem uma relação linear -- isto é, R é constante -- ao longo de uma gama de valores, o material do objeto é chamado de ôhmico. Um resistor ideal tem uma resistência fixa ao longo de todas as freqüências e

amplitudes de tensão e corrente. Materiais supercondutores em temperaturas muito baixas têm resistência zero. Isolantes (tais como ar, diamante, ou outros materiais não-condutores) podem ter resistência extremamente alta (mas não infinita), mas falham e admitem que ocorra um grande fluxo de corrente sob voltagens suficientemente altas.

A resistência de um componente pode ser calculada pelas suas características físicas. A resistência é proporcional ao comprimento do resistor e à resistividade do material (uma propriedade do material), e inversamente proporcional à área da secção transversal. A equação para determinar a resistência de uma seção do material é:

$$R = \frac{\rho L}{A}$$

Onde ρ é a resistividade do material, L é o comprimento, e A é a área da secção transversal. Isso pode ser estendido a uma integral para áreas mais complexas, mas essa fórmula simples é aplicável a fios cilíndricos e à maioria dos condutores comuns. Esse valor está sujeito a mudanças em altas frequências devido ao efeito skin, que diminui a superfície disponível da área. Resistores padrões são vendidos com capacidades variando desde uns poucos milôhms até cerca de um gigaôhms; apenas uma série limitada de valores, chamados valores preferenciais, estão disponíveis. Na prática, o componente discreto vendido como "resistor" não é um resistor perfeito como definido acima. Resistores são freqüentemente marcados com sua tolerância (a variação máxima esperada da resistência marcada). Em resistores codificados com cores, uma faixa mais à direita demonstra uma tolerância de 10%, uma faixa dourada significa 5% de tolerância, uma faixa vermelha marca 2% e uma

faixa marrom significa 1% de tolerância. Resistores com tolerância menores, também chamados de resistores de precisão, também estão disponíveis.

Um resistor tem uma tensão e corrente máximas de trabalho, acima das quais a resistência pode mudar (drasticamente, em alguns casos) ou o resistor pode se danificar fisicamente (queimar, por exemplo). Embora alguns resistores tenham tensões de tensão e corrente especificadas, a maioria deles são taxadas em função de sua potência máxima, que é determinada pelo tamanho físico. As tensões mais comuns para resistores de composição de carbono e filme de metal são 1/8 watt, 1/4 watt e 1/2 watt. Resistores de filme de metal são mais estáveis que os de carbono quanto a mudanças de temperatura e a idade. Resistores maiores são capazes de dissipar mais calor por causa de sua área de superfície maior. Resistores dos tipos wire-wound e sand-filled são usados quando se necessita de tensões grandes de potência, como 20 Watts. Além disso, todos os resistores reais também introduzem alguma indutância e capacitância, que mudam o comportamento dinâmico do resistor da equação ideal.

Resistor Variável

O resistor variável é um resistor cujos valores podem ser ajustados por um movimento mecânico, por exemplo, rodando manualmente. Os resistores variáveis podem ser dos baratos, de volta simples, ou de múltiplas voltas com um elemento helicoidal. Alguns têm um display mecânico para contar as voltas. Tradicionalmente, resistores variáveis são não-confiáveis, porque o fio ou o metal podem se corroer ou se

desgastar. Alguns resistores variáveis modernos usam materiais plásticos que não corroem. Outro método de controle, que não é exatamente um resistor, mas se comporta como um, envolve um sistema sensor fotoelétrico que mede a densidade ótica de um pedaço de filme. Desde que o sensor não toque o filme, é impossível haver desgaste.

Reostato é um resistor variável com dois terminais, sendo um fixo e o outro deslizante. Geralmente são utilizados com altas correntes.

Potenciômetro

É um tipo de resistor variável comum, sendo comumente utilizado para controlar o volume em amplificadores de áudio.

Metal Óxido Varistor ou M.O.V. / Varistores

É um tipo especial de resistor que tem dois valores de resistência muito diferentes, um valor muito alto em baixas voltagens (abaixo de uma voltagem específica), e outro valor baixo de resistência se submetido a altas voltagens (acima da voltagem específica do varistor). Ele é usado geralmente para proteção contra curtos-circuitos em extensões ou pára-raios usados nos postes de ruas, ou como "trava" em circuitos eletro motores.

Termistores

São resistências que variam o seu valor de acordo com a temperatura a que estão submetidas. A relação geralmente é direta, porque os metais usados têm um coeficiente de temperatura positivo, ou seja se a temperatura sobe, a resistência também sobe. Os metais mais usados são a platina, daí as designações Pt100 e Pt1000 (100 porque à temperatura 20°C, têm uma resistência de 100ohm, 1000 porque à temperatura 20°C, têm uma resistência de 1000ohm) e o Níquel (Ni100) os termistores PTC e NTC, são um caso particular, visto que em vez de metais usam semicondutores. Alguns autores não consideram resistências pelo fato de usarem semicondutores.

9.2 ANEXOS II – CÓDIGO FONTE DO SOFTWARE EM ASSEMBLY VERSÃO

FINAL

```
org 0000h
```

```
jmp inicio
```

```
org 0003h
```

```
jmp led_rti
```

```
org 0013h
```

```
jmp int1_rti
```

```
org 001bh
jmp window_RTI
```

led_rti:

```
mov A,pulses
mov B,#10d
div AB
mov divs,A
reti
```

int1_RTI:

```
inc r0
reti
```

window_RTI:

```
djnz r1,not_reached
push acc
push psw
;clr ea ;desabilita interrupcoes
clr tr1 ;desliga timer1
mov th1,#03ch ;zera contador
mov tl1,#0b0h ;zera contador
```

```

mov    Pulses,R0        ;coloca pulsos contados em sua variavel
mov    r0,#00h          ;zera r0
setb   tr1              ;ativa timer 1
;setb  ea               ;ativa interrupcoes
mov    r1,#04h          ;reinicia contagem de tempo 4 x 50ms =
200ms
pop    psw
pop    acc
reti

```

not_reached:

```

clr    tr1              ;desliga timer1
mov    th1,#03ch        ;zera contador
mov    tl1,#0b0h        ;zera contador
setb   tr1              ;ativa timer 1
reti

```

inicio:

```

mov    p1,#00h          ;leds
setb   p3.0             ;leds
setb   p3.1             ;leds
mov    p2,#00h          ;inicialização das portas e reg que serão
utilizados

```

```

mov   TMOD,#00010001b ;Conf timers
setb  ex0
setb  ex1              ;Ativa a interrupcao no pino INT1
setb  et1              ;ativa a interrupcao do timer 1
setb  it1              ;especifica int1 em falling-edge
setb  ea                ;Habilita interrupcoes
mov   r1,#04h          ;seta contador de tempo
mov   r0,#00h          ;zera contador de pulso

mov   th1,#03ch        ;zera contador
mov   tl1,#0b0h        ;zera contador
setb  tr1              ;ativa contador
mov   pulses,#00h      ;zera var RPM

mov   num0,#11000000b ;leds do bcd para cada numero
mov   num1,#11111001b
mov   num2,#10100100b
mov   num3,#10110000b
mov   num4,#10011001b
mov   num5,#10010010b
mov   num6,#10000011b
mov   num7,#11111000b

```

```
mov num8,#10000000b
```

```
mov num9,#10010000b
```

```
mov A,#67d
```

```
mov B,#10d
```

```
div AB
```

```
mov divs,A
```

main:

```
call separate
```

```
call convert_buffers
```

```
call calc_led
```

```
call imprime_bcd
```

```
jmp main
```

imprime_bcd:

```
mov p2,buf4 ;seta numero desejado na porta 2
```

```
clr bcd4 ;liga o bcd que será impresso o numero
```

```
call checktime ;da um tempinho ligado
```

```
nop
```

```
nop
```

```
nop
```

```
setb bcd4 ;desliga o bcd setado e vai pra proxima
```

nop

mov p2,buf3

clr bcd3

call checktime

nop

nop

nop

setb bcd3

nop

mov p2,buf2

clr bcd2

call checktime

nop

nop

nop

setb bcd2

nop

mov p2,buf1

clr bcd1

call checktime


```

nop
nop
nop
setb  bcd1
nop
ret

```

checktime:

```

clr    tf0           ;baixa a flag
;MOV  th0,#0ffh     ;debug
;MOV  TL0,#0f0h     ;debug
MOV    TH0,#0dfh    ;d8f0 = 10ms @ 12Mhz
MOV    TL0,#0f0h    ;
setb   TR0          ;liga t0
jnb    tf0,$        ;aguarda o tempo
clr    TR0          ;desliga t0
ret

```

separate: ;Quando invocada quebra o valor de Pulses

```

mov    buf1,#00h
mov    pdec,#00h
mov    punit,#00h
mov    A,Pulses    ;este numero nao pode ser maior que 66 = 9900 RPS

```

;= 330 Hz

```

jz    fim
mov   B,#10d
div   AB
mov   pdec,A
mov   punit,B
mov   A,punit
mov   B,#05d           ;este 5 vem do 150 cabalístico
mul   AB
mov   B,#10d           ;para separar valor do carry
div   AB
mov   buf2,B
mov   b,punit           ;soma unidade com o carry pois seria
;multiplicação por 1
add   A,B               ;do 150 cabalístico
mov   B,#10d
div   AB
mov   buf3,B
mov   buf4,A
mov   A,pdec
jz    fim               ;teste para ver se tem unidade no num. de pulso
mov   B,#05d           ;este 5 vem do 150 cabalístico novamente
mul   AB
mov   B,#10d           ;para dividir decimal de unidade

```

```

div    AB
mov    aunit,B
mov    adec,A
mov    a,buf3
add    A,B
mov    B,#10d
div    AB
mov    buf3,B
mov    B,pdec
add    A,B
mov    B,adec
add    A,B
mov    B,buf4
add    A,B
mov    buf4,A           ;Atualiza o valor de buf4
mov    aunit,#00h      ;zera os valores auxiliares
mov    adec,#00h
fim:   ret

```

convert_buffers:

```

mov    A,buf4
call   convert_bcd     ;retorna em A o codigo BCD do numero
mov    buf4,A          ;coloca o codigo na variavel dele

```

```

mov  A,buf3
call convert_bcd      ;retorna em A o codigo BCD do numero
mov  buf3,A          ;coloca o codigo na variavel dele
mov  A,buf2
call convert_bcd      ;retorna em A o codigo BCD do numero
mov  buf2,A          ;coloca o codigo na variavel dele
mov  A,buf1
call convert_bcd      ;retorna em A o codigo BCD do numero
mov  buf1,A          ;coloca o codigo na variavel dele
ret

```

convert_bcd:

```

    cjne  a,#0,um      ;Ã© zero o 2 digito? (primeiro digito cravado
sempre em 0)
    mov  a,num0        ;sim logo move zero
    ret
um:   cjne  a,#1,dois
    mov  a,num1
    ret
dois: cjne  a,#2,tres
    mov  a,num2
    ret
tres: cjne  a,#3,quatro
    mov  a,num3

```

```

        ret
quatro:  cjne  a,#4,cinco
        mov  a,num4
        ret
cinco:   cjne  a,#5,seis
        mov  a,num5
        ret
seis:    cjne  a,#6,sete
        mov  a,num6
        ret
sete:    cjne  a,#7,oito
        mov  a,num7
        ret
oito:    cjne  a,#8,nove
        mov  a,num8
        ret
nove:    mov  a,num9      ;nenhum anterior? soh pode ser o nove
        ret

calc_led:
        mov  A,pulses    ;move pulsos lidos
        mov  B,divs      ;move divisor calculado
        div  ab

```

```
jz    zero
cjne  A,#10d,tmu ;todos menos um
mov   p2,#00h   ;leds
clr   p3.0      ;leds
clr   p3.1      ;leds
ret
```

```
tmu:  cjne  A,#09d,soh8
      mov   p2,#00h   ;leds
      clr   p3.0      ;leds
      setb  p3.1      ;leds
      ret
```

soh8:

```
mov   numled,A    ;Salva numero de leds que tem que acender
mov   a,#0ffh
setb  p3.0
setb  p3.2
```

soh8_2:

```
rl    A
dec   A
djnz  numled,soh8_2
mov   p1,A
```

```

        ret
zero:  setb  p3.0
        setb  p3.1
        mov  p1,#0ffh
        ret

```

```

;begin defines

```

```

buf1  equ  1Ah      ;Posicoes que a rotina imprime_bcd le
buf2  equ  1Bh      ;em ordem buf1 unidade 2 dezena
buf3  equ  1Ch      ;buf3 centena e buf4 milhar
buf4  equ  1Dh
punit equ  1Eh      ;parte unitaria do pulso
pdec  equ  1Fh      ;parte decimal do pulso se for maior que 9
aunit equ  20h      ;unidade auxiliar
adec  equ  21h      ;decimal auxiliar

Pulses    equ  22h
Pul_old   equ  23h
Divs      equ  24h
numled    equ  25h

bcd1  equ  p3.4      ;portas do chaveamento bcd

```

bcd2 equ p3.5

bcd3 equ p3.6

bcd4 equ p3.7

led1 bit p1.0

led2 bit p1.1

led3 bit p1.2

led4 bit p1.3

led5 bit p1.4

led6 bit p1.5

led7 bit p1.6

led8 bit p1.7

led9 bit p3.0

led10 bit p3.1

num0 equ 10h ;leds do bcd para cada numero

num1 equ 11h

num2 equ 12h

num3 equ 13h

num4 equ 14h

num5 equ 15h

num6 equ 16h

num7 equ 17h

num8 equ 18h

num9 equ 19h

END

Pré-Programa e Software's de Testes

bcd1 bit p3.4

bcd2 bit p3.5 ;LEMBRAR DE MUDAR ESTA PORTA POIS ELE

ESTA PEGANDO OS PULSOS VINDOS DELA

bcd3 bit p3.6

bcd4 bit p3.7

multiplier EQU 30h ;RAM Location

MULTIPLIER_DEFAULT DATA 1 ;2 x 50ms = 100 ms

test equ 40h ;RPS (rotação por segundo) logo não

passara de 200 RPS pois $200 \cdot 60 = 12000$ RPM (limite proposto no projeto) assim 8 bits

dara conta de 200 decimal sendo o maximo 256 decimal em 8bits e definido pelos

integrantes 166RPS (por causa dos 4 BCD apenas $167 = 10020$ RPM) ;MEMORIA

IMPORTANTE posição 18,19 e 1A terão o valor de RPS 8 mais significativo e A

menos significativo posição 10h,11h,12h,13h e 14h serão o valor quebrado de

RPM 10h mais significativo e 14h menos significativo

espera equ 43h ;aqui ele ira fazer uma espera
mostrando no bcd e leds o valor adquirido depois desta espera de (espera x (2 x 50ms))

= tempo q ira ficar mostrando no bcd e leds

num0 equ 50h ;armazena valores formato bcd puro!

num1 equ 51h

num2 equ 52h

num3 equ 53h

num4 equ 54h

num5 equ 55h

num6 equ 56h

num7 equ 57h

num8 equ 58h

num9 equ 59h

led1 bit p1.0

led2 bit p1.1

led3 bit p1.2

led4 bit p1.3

led5 bit p1.4

led6 bit p1.5

led7 bit p1.6

led8 bit p1.7

led9 bit p3.0

led10 bit p3.1

```

ORG 0000h

    jmp começa

org 13h

    jmp int1_rti

org 01bh

    jmp window_RTI

int1_RTI:

    inc r3

    reti

window_RTI:

    push acc

    push psw

    clr ea                ;desabilita interrupcoes

    clr tr1              ;desliga timer1

    mov th1,#00h        ;zera contador

    mov tl1,#00h        ;zera contador

    mov a,r3

    jz zero

    mov a,#5d           ;numero cabalistico que gera RPS

    mov b,r3            ;salva pulsos contados em r0

```

```

mov r3,#00h          ;zera r0
mul ab              ;multiplica A por B - result em RPs
mov test,A          ;coloca RPS em sua variavel
zero: setb tr1      ;ativa timer 1
setb ex1
setb et1
setb ea            ;ativa interrupcoes
pop psw
pop acc
pop a              ;tiram os o endereço que haviamos parado nos
queremos recalculer tudo novamente! e nao continuar da onde paramos!
pop b
mov a,#00h
mov b,#9ch        ;posicao em hexa do conta_pulsos forçamos o
program counter a ir nesta posição ao invéz de voltar aonde havia parado antes da
interrupção -----ATENCAO lembrar de mudar isto se adicionar mais qualquer linha de
codigo!!!!!!!!!!!! pois o valor em hexa IRA mudar!
push b
push a
reti
testa_zero:
inc R4            ;vemos se em rps existe 3 numero com zero
se for entao rps = 0 logo rpm nao precisa ser convertido é 0 tambem

```

```
ret
```

```
rpm_zerado:
```

```
mov 10h,#0h ;zeramos o rpm pois o rps jah deu zero sem  
necessidade de converção e nao gasta tempo!
```

```
mov 11h,#0h
```

```
mov 12h,#0h
```

```
mov 13h,#0h
```

```
mov 14h,#0h
```

```
ret
```

```
comeca:
```

```
mov num0,#11000000b ;carregando valores bcd puro para  
memoria
```

```
mov num1,#11111001b
```

```
mov num2,#10100100b
```

```
mov num3,#10110000b
```

```
mov num4,#10011001b
```

```
mov num5,#10010010b
```

```
mov num6,#10000011b
```

```
mov num7,#11111000b
```

```
mov num8,#10000000b
```

```
mov num9,#10010000b
```

```
mov multiplier,#MULTIPLIER_DEFAULT
```

```

mov TMOD,#00010001b    ;Timer 0 Mode 1(16 bit mode) e Timer 1
Mode 1 tambem (contar tempo em nivel logico 1)
mov TH0,#0E5h          ;0.05 sec
mov TL0,#0F5h
mov th1,#00h           ;zera contador
mov tl1,#00h           ;zera contador
setb ex1               ;Ativa a interrupcao no pino INT1
setb et1               ;ativa a interrupcao do timer 1
setb ea                ;Habilita interrupcoes
;setb px1

conta_pulsos:
clr f0                 ;desligamos a flag F0 para poder
refazer as contas dos leds
mov r4,#0h             ;zeramos o r4 para contar os zeros do
rps
mov espera,#02h        ;quantas vezes ele ira fazer o
loop de imprimir no bcd?
setb tr1               ;ativa contador
quebra_valor_3:        ;acha a 3 casa decimal
mov A,test
mov B,#100
div ab
mov 18h,A

```

```

                jnz nop                ;testamos se os 3 valores do rps
eh zero ou nao

                call testa_zero

quebra_valor_2:                ;acha a 2 casa decimal

    nop:  mov A,test

                mov B,#10

                div ab                ;exemplo 256 ã© o RPS logo
256/10 =25 queremos apenas o decimal 5 e nã£o o 2

                mov R5,A

                mov a,18h

                mov b,#10

                mul ab                ;aqui teremos a retirada do
decimal 2 que jã£i temos porem temos q faze-lo voltar a forma de 20

                xch a,b                ;realoca a->b e vise versa

                mov a,R5

                clr c

                subb a,b                ;25-20 = 5 que desejamos

                mov 19h,a                ;finalmente guardamos esse
valor sobrescrevendo o anterior 25

                jnz nop1

                call testa_zero

quebra_valor_1:                ;acha a primeira casa decimal

    nop1: mov A,18h

```

```

        mov B,#100
        mul ab
        mov R7,A                ;recoloca apenas a 3 casa
decimal 256 apos isso 200
        mov A,19h
        mov b,#10
        mul ab                  ;achou apenas a 2 casa
decimal se fosse 56 agora ã© 50
        mov R6,A
        mov A,test
        mov b,R7
        clr c                    ;limpa o carry pois o sub pode ter
o "emprestado" e não ã© objetivo aqui
        subb A,B                ;como no exemplo anterior 256 -
200 = 56
        clr c
        mov b,R6
        subb a,b                ;finalmente aqui 56-50 = 6 que
nos desejamos, a primeira casa decimal!
        mov 1ah,a              ;move-se para o lugar apropriado
        jnz nop2
        call testa_zero

```

RPS_RPM:

nop2: cjne r4,#3h,continua ;se o r4 nao tiver 3 ou seja 3 zeros do rps
significando rps = 0 entao faça a converçao se nao pule e imprima os zeros

jmp loop

continua: clr c ;limpamos o carry e auxiliar
tambã©m

clr ac

mov a,1ah

mov b,#3

mul ab ;aqui teremos a primeira

mul do primeiro digito do RPS

mov r0,a ;teremos que guardar o resultado

da multiplicaçao com a sobra que ã© o segundo digito

mov b,#10 ;iremos dividir por 10 para

descobrir a sobra e o valor da multiplicaçao desta casa em especial

div ab

mov r1,a ;por definiçao colokaremos r1

como a sobra

mov b,#10

mul ab

xch a,b

mov a,r0

clr c

clr ac

```
subb a,b
```

```
mov 013h,a ;achamos o segundo digito
```

do RPM como o primeiro esta cravado em 0 nÃ£o precisa calcular e Ã© o endereÃ§o 14h

```
mov a,19h
```

```
mov b,#3
```

```
mul ab
```

```
add a,r1 ;aqui teremos a multiplicaÃ§Ã£o
```

para achar o terceiro digito E a sobra da multiplicaÃ§Ã£o anterior R1 !!!

```
mov r0,a ;guardar o valor com sobra no r0
```

para futuro calculo do digito em si desta posiÃ§Ã£o

```
mov b,#10
```

```
div ab
```

```
mov r1,a ;achamos a sobra para o proximo
```

digito da multiplicaÃ§Ã£o deixamos ele guardado

```
mov b,#10
```

```
mul ab
```

```
xch a,b
```

```
mov a,r0
```

```
clr c
```

```
clr ac
```

```
subb a,b ;apartir daqui encontramos o
```

valor da posiÃ§Ã£o que desejamos

```

mov 12h,a ;guardamos esse valor de RPM
da 3 casa

mov a,18h
mov b,#3
mul ab
add a,r1 ;adicionamos a sobra anterior ao
valor da multiplicaÃ§ao atual

mov r0,a ;teremos a multiplicaÃ§ao do
ultimo digito de RPS por 6

mov b,#10
div ab
mov r1,a ;achamos a sobra da ultima
multiplicaÃ§ao da ultima casa aqui guardaremos a sobra pois ela serÃ¡ a ultima casa
do RPM

mov b,#10
mul ab ;teremos q tirar o resto do
digito desta posiÃ§ao logo sub

xch a,b
mov a,r0
clr c
clr ac
subb a,b ;achamos o digito desta
posiÃ§ao

```

```
mov 11h,a
```

```
mov 10h,r1 ;e movemos tb o resto que sera
```

neste ultimo caso o ultimo digito do RPM

loop:

```
jnb f0,led_main ;adotamos a flag F0 (reservada
```

para propositos gerais) para saber se precisa ou não mudar os leds poupa MUITO tempo de processamento!!!

```
mov p2,num0 ;cravado em zero pois Y (rps) x 60 (min)
```

= #####0 (rpm)<- vai ter sempre 0 no começo

```
clr bcd1
```

```
call check_timer
```

```
setb bcd1
```

```
mov p2,num0 ;segundo digito cravado em zero
```

também

```
clr bcd2
```

```
call check_timer
```

```
setb bcd2
```

```
mov a,12h ;definimos qual digito queremos
```

converter

```
call convert_bcd ;funcao para convercao de numeros para formato bcd
```

```
mov p2,a
```

```
clr bcd3
```

```

        call    check_timer
        setb   bcd3
        mov    a,11h
call    convert_bcd
        mov    p2,a
        clr    bcd4
call    check_timer
        setb   bcd4
        djnz   espera,loop        ;ficara no loop até o tempo definido

```

no espera

lcall conta_pulsos;apos o numero de vezes definidas saira e
pegara amostras novamente

check_timer:

```

        setb   TR0                ;Start Timer 0
        jnb   TF0,$              ;wait for timer to over flow
        clr   TR0                ;Stop Timer 0
        clr   TF0                ;clear TIMER 0 flag

        mov   TH0,#0E5h ;0.05 sec, Reload timer 0
        mov   TL0,#0F5H
        djnz  multiplier,check_timer ;Decrement

```

multiplier by 1 & check if its 0

```

        mov   multiplier,#multiplier_default
        ret

```

led_main:

mov a,test ;testamos se for zero

jz ok5

mov a,test ;usamos 16 pois o maximo sera

166rps = 9960 RPM bcd apenas suporta 4 numeros

mov b,#16

div ab ;iremos dividir para saber se tem

cociente ou nao

jz ok6 ;soh ira pular para instruÃ§ao

led1 se houver cociente da divisao ou seja (RPS>16 logo acende 1 led)

mov a,test

mov b,#32

div ab

jz ok7

mov a,test

mov b,#48

div ab

jz ok8

mov a,test

mov b,#64

div ab

jz ok0

jmp led_4

mov a,test

mov b,#80

div ab

jz ok

mov a,test

mov b,#96

div ab

jz ok1

mov a,test

mov b,#112

div ab

jz ok2

mov a,test

mov b,#128

div ab

jz ok3

mov a,test

mov b,#144

div ab

jz ok4

mov a,test

jmp led_10

;se nao Ã© nenhum anterior soh pode

ser todos os leds entao!

ret

ok8: jmp led_3

ok7: jmp led_2

ok6: jmp led_1

ok5: jmp led_0

ok0: jmp led_4

ok: jmp led_5

ok1: jmp led_6

ok2: jmp led_7

ok3: jmp led_8

ok4: jmp led_9

led_0:

setb led1

setb led2

setb led3

setb led4

setb led5

setb led6

setb led7

setb led8

setb led9

setb led10

setb F0


```

                                ajmp  loop
led_1:
                                clr   led1           ;acender leds por categoria do RPS
                                setb  led2
                                setb  led3
                                setb  led4
                                setb  led5
                                setb  led6
                                setb  led7
                                setb  led8
                                setb  led9
                                setb  led10
                                setb  F0             ;essa sera a forma de nos dizer para o
processamento que o led foi calculado e soh ira tirar a flag quando for pra refazer os
calculos de RPS!
                                ajmp  loop           ;depois de achar quantos leds iram acender
nÃ£o precisa voltar para main_led volta para loop porem agora com a flag F0 setada
ele nao ira voltar para led_main

```

led_2:

```

                                clr   led1
                                clr   led2
                                setb  led3
                                setb  led4

```

```
setb led5  
setb led6  
setb led7  
setb led8  
setb led9  
setb led10  
setb F0  
ajmp loop
```

led_3:

```
clr led1  
clr led2  
clr led3  
setb led4  
setb led5  
setb led6  
setb led7  
setb led8  
setb led9  
setb led10  
setb F0  
ajmp loop
```

led_4:

```
clr led1
```

```
clr led2
clr led3
clr led4
setb led5
setb led6
setb led7
setb led8
setb led9
setb led10
setb F0
ajmp loop
```

led_5:

```
clr led1
clr led2
clr led3
clr led4
clr led5
setb led6
setb led7
setb led8
setb led9
setb led10
setb F0
```

```
                                ajmp  loop
led_6:
                                clr   led1
                                clr   led2
                                clr   led3
                                clr   led4
                                clr   led5
                                clr   led6
                                setb  led7
                                setb  led8
                                setb  led9
                                setb  led10
                                setb  F0
                                ajmp  loop
led_7:
                                clr   led1
                                clr   led2
                                clr   led3
                                clr   led4
                                clr   led5
                                clr   led6
                                clr   led7
                                setb  led8
```

```
setb led9
setb led10
setb F0
ajmp loop
```

led_8:

```
clr led1
clr led2
clr led3
clr led4
clr led5
clr led6
clr led7
clr led8
setb led9
setb led10
setb F0
ajmp loop
```

led_9:

```
clr led1
clr led2
clr led3
clr led4
clr led5
```

```
clr led6
clr led7
clr led8
clr led9
setb led10
setb F0
ajmp loop
```

led_10:

```
clr led1
clr led2
clr led3
clr led4
clr led5
clr led6
clr led7
clr led8
clr led9
clr led10
setb F0
ajmp loop
```

convert_bcd:

```
cjne a,#0,um ;Ã© zero o 2 digito? (primeiro digito cravado
```

sempre em 0)

```

        mov  a,num0 ;sim logo move zero
        ret
um:     cjne a,#1,dois
        mov  a,num1
        ret
dois:   cjne a,#2,tres
        mov  a,num2
        ret
tres:   cjne a,#3,quatro
        mov  a,num3
        ret
quatro:        cjne a,#4,cinco
        mov  a,num4
        ret
cinco:   cjne a,#5,seis
        mov  a,num5
        ret
seis:   cjne a,#6,sete
        mov  a,num6
        ret
sete:   cjne a,#7,oito
        mov  a,num7
        ret

```

```
oito:      cjne  a,#8,nove
           mov  a,num8
           ret
nove:     mov  a,num9 ;nenhum anterior? soh pode ser o nove
           ret
end
```


9.3 Anexos III– Placa Geradora de Onda Quadrada

Modulo utilizado no projeto para gerar ondas, e fazer o teste no tacômetro para visualização da placa principal em funcionamento sem a necessidade de ter um alternador real em conjunto. A placa simplesmente gera ondas quadradas com a frequência que o usuário desejar (65 Hz ~600 Hz).

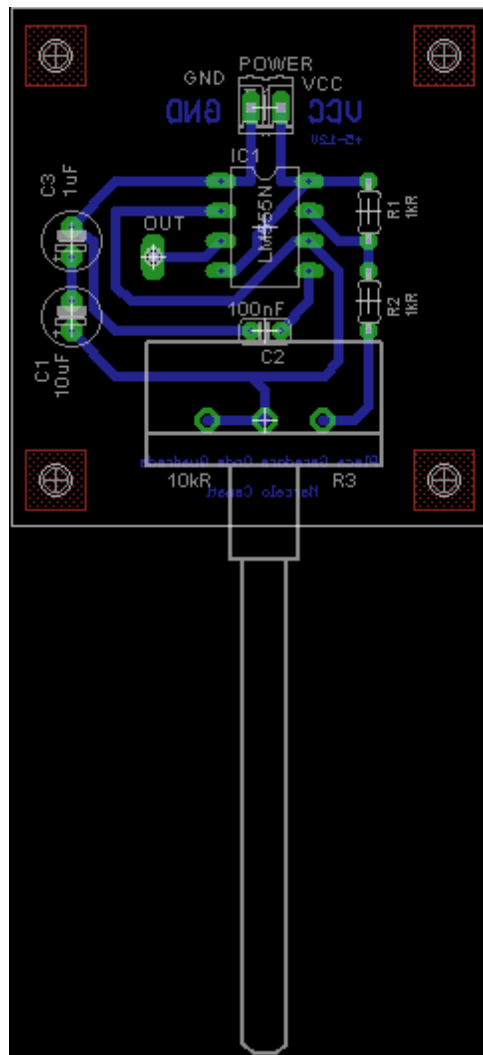


Figura 03: Esquema da Placa Simuladora de Onda

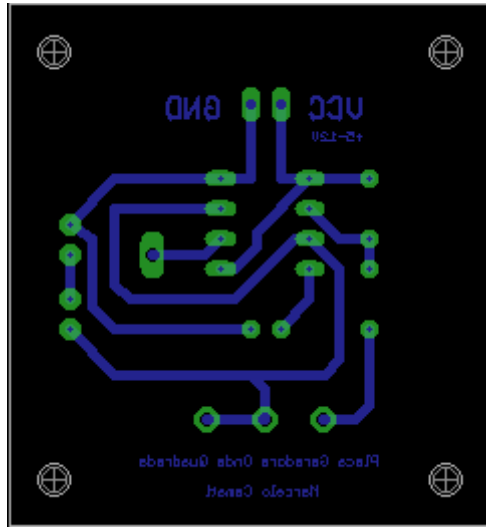


Figura 04: Placa Simuladora de Onda

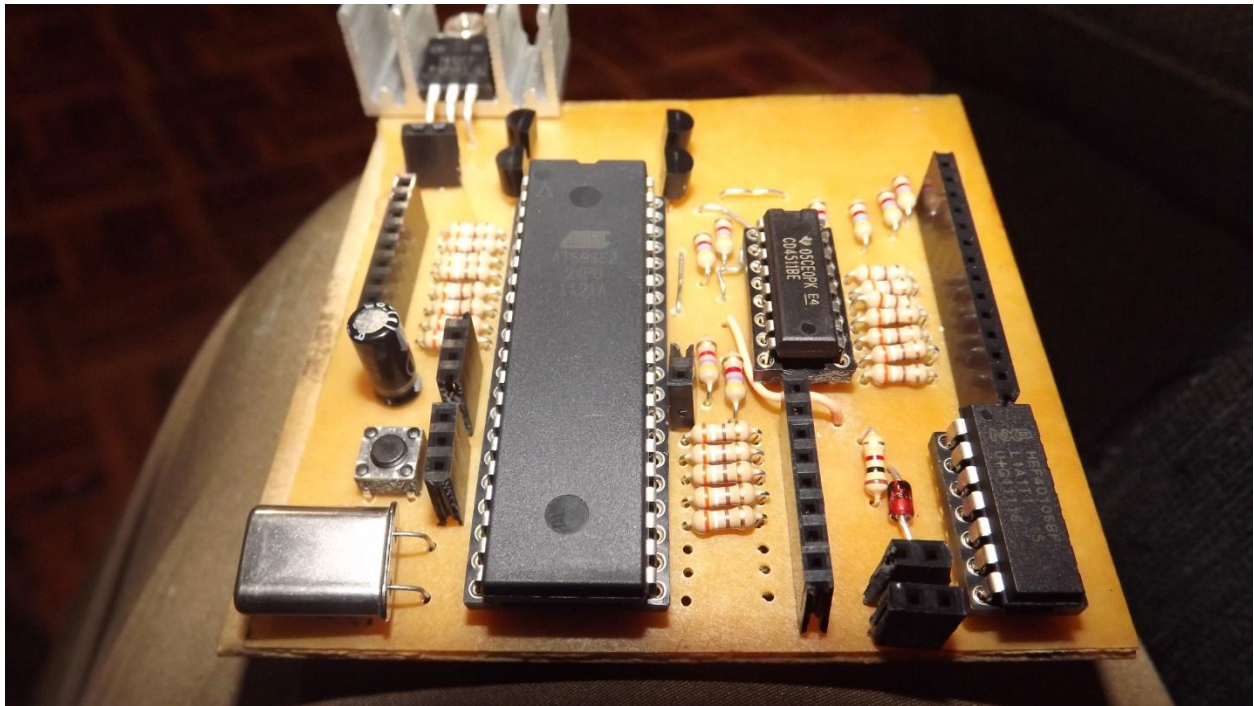


Figura 05: Primeira Versão do Projeto

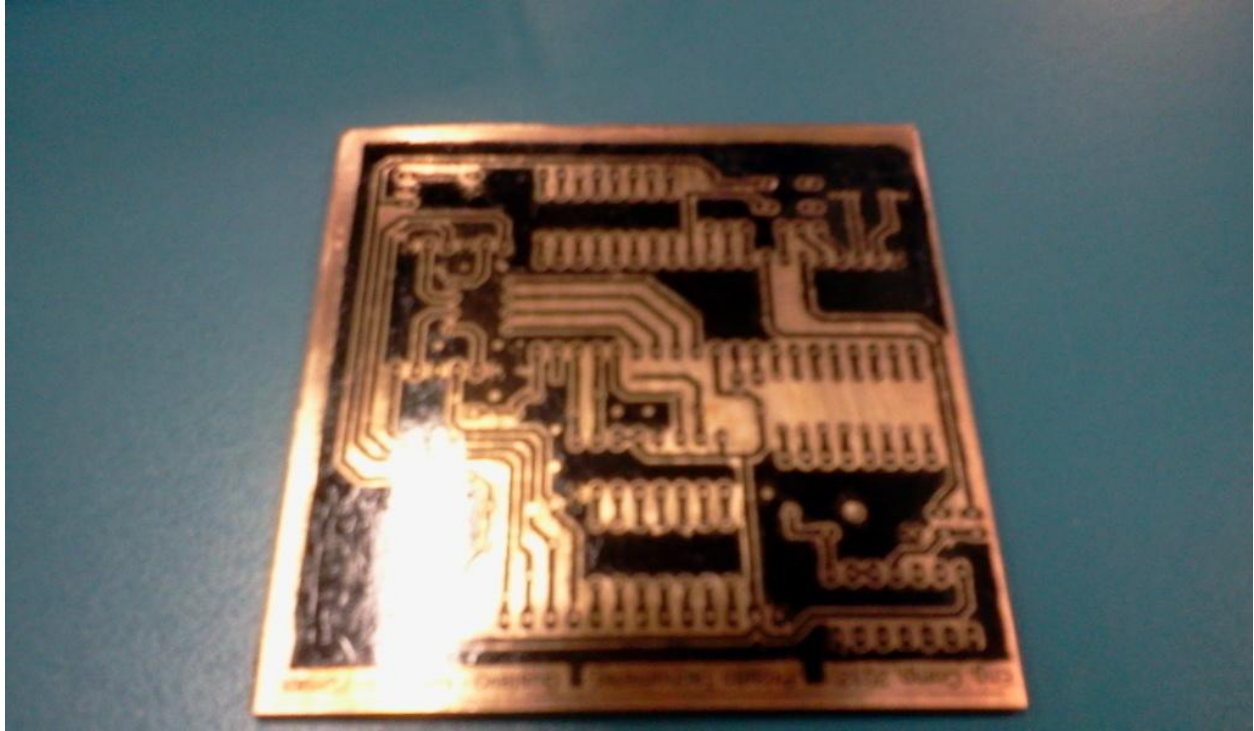


Figura 06: Primeira Versão Pronta para Corroer

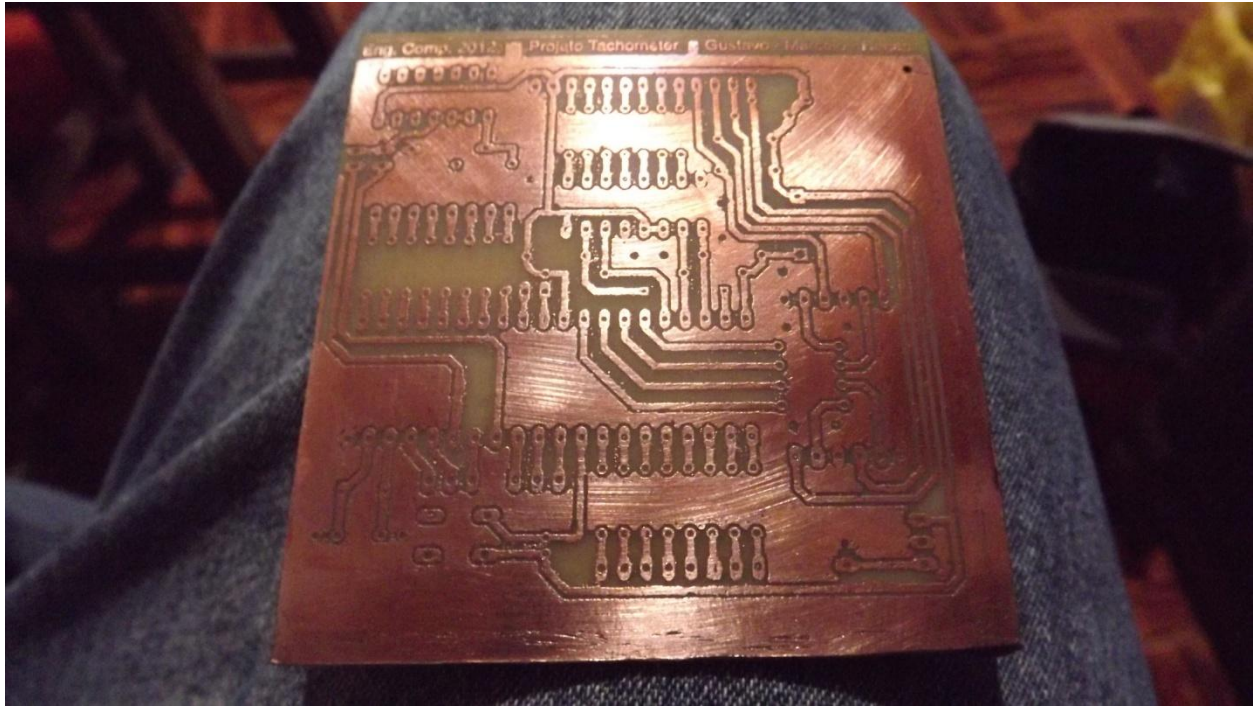


Figura 07: Primeira Versão Corroída e Furada

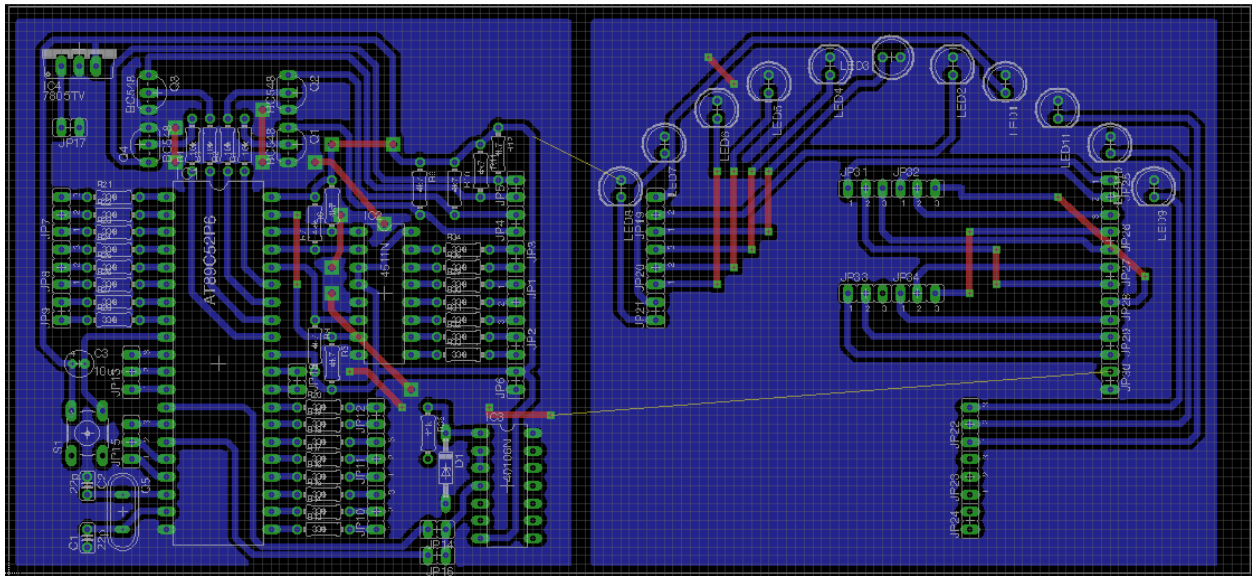


Figura 09: Primeira Versão do Projeto da Placa

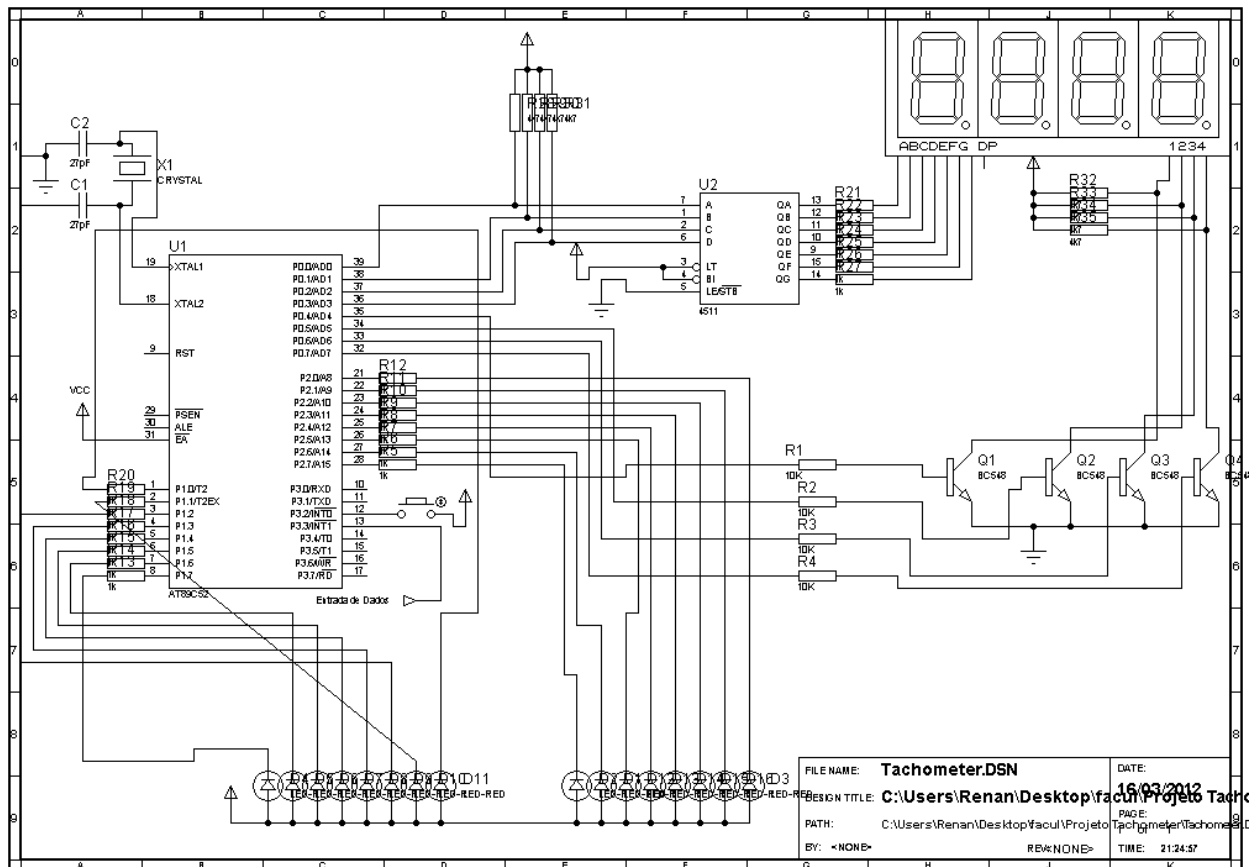


Figura 09: Esquemático da Primeira Versão do Projeto

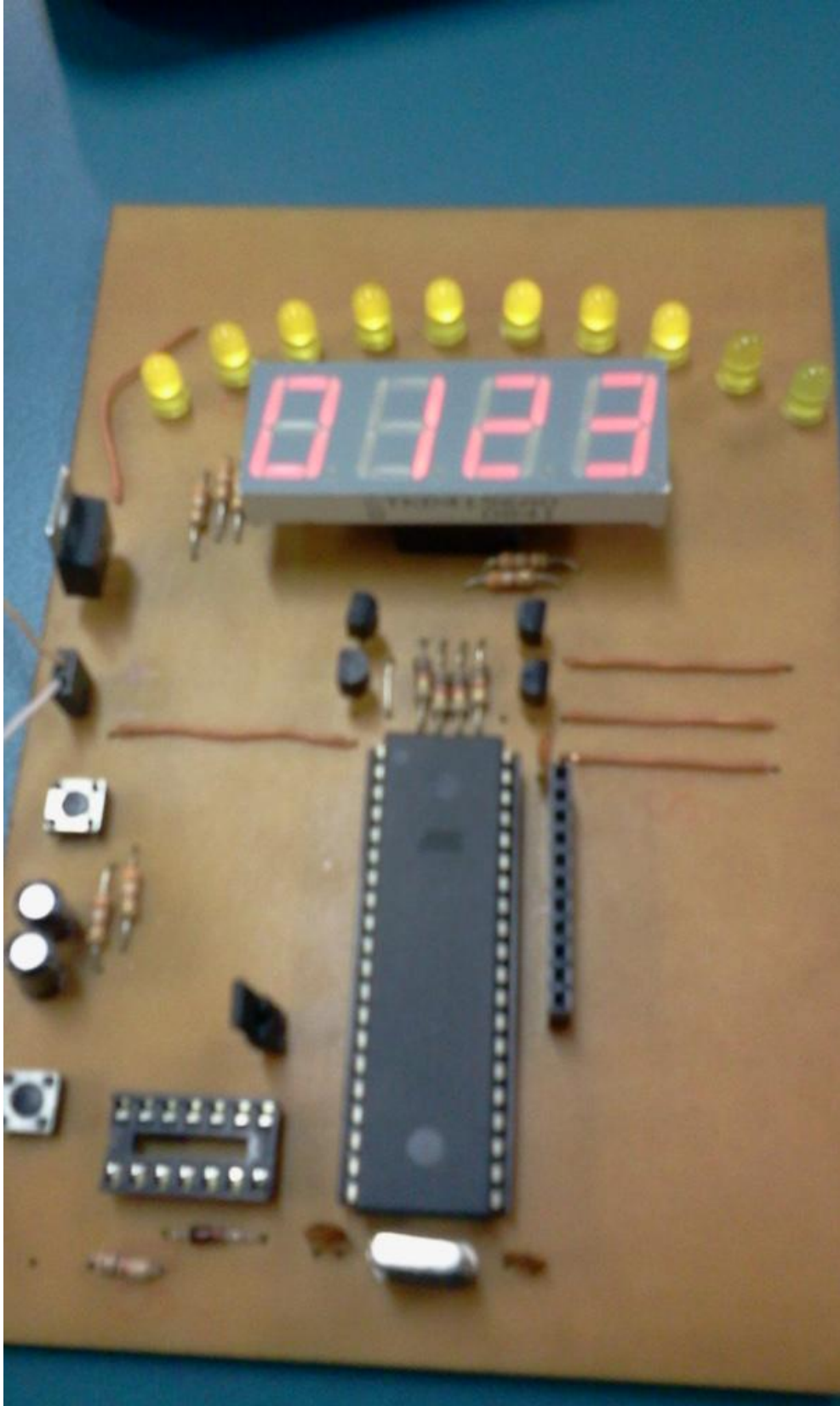


Figura 10: Versão Final Pronta

11. Referências Bibliográficas

<http://afonsomiguel.com> Acessado em junho de 2012

<http://www.freemansgarage.com/tach.html> Acessado em junho de 2012

<http://pt.wikipedia.org/wiki/Tac%C3%B4metro> Acessado em junho de 2012

http://pt.wikipedia.org/wiki/Painel_de_instrumentos Acessado em junho de 2012

<http://www.atmel.com/Images/doc1919.pdf> Acessado em junho de 2012

http://www.datasheetcatalog.net/pt/datasheets_pdf/A/T/8/9/AT89S52-24AI.shtml Acessado em junho de 2012

<http://www.keil.com/dd/chip/3411.htm> Acessado em junho de 2012

<http://www.asic-world.com/digital/tutorial.html> Acessado em junho de 2012

<http://www.clubedohardware.com.br/>

Acessado em junho de 2012