

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
ENGENHARIA DE COMPUTAÇÃO**

PROJETO AUTO CAR

**CURITIBA
2009.**

**ANDERSON GEREZ
CÉSAR MENDES CHRUSCIEL**

PROJETO AUTO CAR

Este projeto será apresentado à disciplina de Microprocessadores I do Curso de Engenharia de Computação do Centro de Ciências Exatas e de Tecnologia da Pontifícia Universidade Católica do Paraná, como parte integrante da nota do primeiro semestre. A finalidade deste projeto é o trabalho com microprocessadores 8051

Professores orientadores: Afonso Ferreira Miguel.

CURITIBA

2009.

SUMÁRIO

1. INTRODUÇÃO.....	03
1.1 JUSTIFICATIVAS.....	03
1.2 METODOLOGIA.....	04
1.3 RESPONSABILIDADES.....	04
2. OBJETIVOS.....	05
3. NÃO ESTÁ INCLUSO NO ESCOPO DESTE PROJETO.....	05
4. O PROJETO.....	06
5. OS RESULTADOS.....	08
6. A EQUIPE DE DESENVOLVIMENTO.....	09
7. FOTOS.....	10
8. CONCLUSÃO.....	10
9. ANEXOS.....	11
9.1 ANEXOS I – DICIONÁRIO DE TERMOS TÉCNICOS.....	11
9.2 ANEXOS II – CÓDIGO FONTE DO SOFTWARE DO 8051.....	26
9.3 ANEXOS III – DATASHEETS.....	42

1. INTRODUÇÃO

Projeto de Microprocessadores do curso de Engenharia de Computação tem como intuito iniciar o desenvolvimento de projetos, desde a documentação completa, organogramas, cronogramas, apresentações e a conclusão do projeto nas mais corretas formas, além do trabalho com microprocessadores 8051 em sistemas embarcados. Capacitando o aluno a gerenciar um projeto para quando entrar no mercado de trabalho não ter tantas dificuldades em realizar qualquer tipo do mesmo.

O grupo formado para o desenvolvimento do Projeto de Microprocessadores do quinto período do curso de Engenharia de Computação é formado pelos seguintes integrantes: Anderson Gerez e César Mendes Chrusciel.

A idéia do projeto surgiu depois de muitos diálogos entre o grupo, entre algumas delas estavam: caixa que controlava os remédios de pacientes, teclado com tela LCD e muitas outras idéias, mas uma delas foi aceita com mais entusiasmo pelo grupo, que foi a idéia do Auto Car, um sistema que controla todas as funções do carro, sendo um sistema que de automação do carro e ao mesmo tempo um sistema anti-furto.

A partir desse dia o projeto passou a se chamar “Auto Car”.

1.1 JUSTIFICATIVAS

O projeto “Auto Car” se destaca por ser um tipo de sistema embarcado, no qual tem uma aplicabilidade enorme, como neste caso foi feito um sistema para carro, o mesmo sistema, com algumas modificações poderia ser implantado numa casa.

Com a crescente tecnologia do mundo de hoje e da variedade de novos sistemas de automação existentes, mais projetos como estes surgem, dando um novo rumo à tecnologia, juntando informatização e automação em sistemas cada vez mais inteligentes, no qual executam tarefas, controlam, monitoram, etc, apenas possuindo um microprocessador simples e um circuito que o faz funcionar.

O Projeto Auto Car é sistema barato, que pode ser vendido comercialmente por um preço competitivo, pois para sua montagem utiliza materiais que apresentam um baixo custo, podendo assim ser um concorrente no mercado atual.

1.2 METODOLOGIA

A metodologia de desenvolvimento do “Auto Car” foi estruturada de uma forma a gerar poucos erros, pois o tempo é curto. O plano foi em primeiro lugar fazer pesquisas de componentes eletrônicos, programação Assembly, microprocessadores 8051, pesquisa de preços, pesquisa dos princípios de funcionamento de uma automação de um carro e tudo que envolveu o projeto.

Nas primeiras semanas começamos a montagem mecânica, com a fixação dos motores dos vidros elétricos, esquerdo e direito, rádio, sirene, centralina dos vidros, travas das portas, computador de bordo e sistema representativo da abertura da porta, ignição, lanternas e bomba injetora de combustível. Pensando também no design da giga de teste com a fixação dos seus componentes de modo a gerar uma boa visualização do seu funcionamento, e em paralelo a essas tarefas anteriores foi desenvolvido o software em linguagem Assembly para o microprocessador 8051, no qual foi escolhido esse microprocessador para atender as exigências do Projeto Integrado de Microprocessadores I. E também foi feito testes simulando o roubo do carro, abertura da porta e o não destravamento correto do sistema antifurto, o qual corta a bomba de combustível e dispara o alarme após um determinado tempo, testes também de disparo do alarme, controle do rádio, etc.

Os equipamentos que utilizamos foi osciloscópio para exibir os formatos de ondas gerados pelo circuito elétrico, gravadora universal para a gravação do programa em Assembly no microprocessador 8051, multímetros para medição de correntes, tensões e outros testes, fontes de alimentação, além de uma bateria de 12V – 50A para alimentação dos motores do vidro elétrico, sirene, lanternas, etc.

1.3 AS RESPONSABILIDADES

Para que o projeto obtivesse tamanho sucesso em seu desenvolvimento foi necessário 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 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 fazer a gravação do microprocessador 8051e com os devidos equipamentos que conseguimos levar o projeto adiante.

2. OS OBJETIVOS

O projeto “Auto Car” tem como objetivos, “ensinar” o grupo a fazer pesquisas, documentações, cronogramas, apresentações, ou seja, tudo que envolve um bom gerenciamento de projetos, além do trabalho com o microprocessador 8051 e da linguagem Assembly, requisitos desse Projeto Integrado, assim como utilizar teorias usadas em sala de aula já em aplicações mais próximas da engenharia propriamente dita, como a utilização da arquitetura de computadores para o trabalho com o microprocessador e principalmente com a linguagem Assembler, a própria disciplina mesmo de microprocessadores o qual nos mostrou como funciona o 8051, mecânica em geral, circuitos elétricos e sistemas digitais para uma compreensão melhor da eletrônica analógica e digital e das tecnologias que serão utilizadas.

3. NÃO ESTÁ INCLUSO NO ESCOPO DESTA PROJETO

O projeto tem uma ampla diversidade de tecnologias que poderiam ser feitas, mas que não serão implementadas nessa versão, mas talvez nos próximo semestre no “Auto Car Two”.

As tecnologias que poderiam ser feitas são: controle remoto ao invés de controle por computador de bordo, uso de outro microprocessador como, por exemplo, o MSP430, controle de mais funcionalidades do carro, ao invés de senha, controle da liberação da direção por reconhecimento da impressão digital do proprietário, etc, mas tudo isso depende de custos, capacitação, tempo e muito, muito estudo.

4. O PROJETO

O Projeto “Auto Car” faz a simulação de uma automação em automóveis, no qual é toda controlada por um computador de bordo, sendo este um circuito embarcado aonde trabalha com o microprocessador 8051.

Todo o controle do projeto fica a cargo do computador de bordo, ele que gerencia o funcionamento de todas as funções que a giga de teste apresenta. O “Auto Car” faz o controle do rádio, de automação dos vidros, corte de ignição do carro, trava e destrava as portas, dispara sirene e as lanternas (piscas).

Quando o sistema é ligado o carro entra cortado, tendo em vista que um ladrão pode cortar a ligação do circuito elétrico com a alimentação (bateria), neste caso, o ladrão não teria como ligar o automóvel, mesmo que tente fazer pegar no “tranco”, pois o sistema de ignição do carro se encontrava cortado.

A primeira frase que aparece no visor de LCD do computador de bordo é para efetuar o cadastro, onde que é feito de modo simples, é só digitar um código e precisar o tecla enter. Com o cadastro efetuado, quando o usuário for usar o veículo se faz necessário que ele digite a sua senha, caso ele digite errado, aparece no visor à frase de senha incorreta e na seqüência o sistema informará no visor para que o usuário digite a senha novamente. Digitado a senha corretamente é atracado um relé, o qual faz com que a ignição do carro seja liberada.

Com o carro funcionando o motorista pode controlar os vidros, o travamento das portas e o rádio do carro. O sistema funciona da seguinte maneira com os botões do teclado:

- 0 = Liga o som do carro;
- 9 = Desliga o som do carro;

- 4 = Abaixa os vidros do carro (direita e esquerda simultaneamente);
- 2 = Sobre o vidro esquerdo;
- 5 = Desce o vidro esquerdo;
- 3 = Sobre o vidro direito;
- 6 = Desce o vidro direito;
- 7 = Travas as portas;
- 8 = Destrava as portas;

O funcionamento do vidros tem duas etapas, com apenas um toque, os vidros descem / sobem até o final, porém caso seja dado mais um toque no teclado quando os vidros estiverem em funcionamento o mesmo para na posição que se encontra no momento.

No caso de um assalto relâmpago em um semáforo, tem duas chaves que fazem a simulação para essa representação, tem uma chave I/O que representa a porta do motorista, onde que o ladrão abre e fecha a porta para entrar no carro e mais uma chave I/O que representam a chave de ignição do carro. Quando o ladrão bater a ignição do carro ele terá 30 segundos para digitar a senha, depois desse tempo o alarme do carro dispara e mais 10 segundos é cortado a ignição do carro. Para que a ignição seja liberada novamente e o alarme desligado, faz-se necessário digitar a senha.

5. OS RESULTADOS

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

1. Protótipo funcionando da Giga de Teste;
 - Placa circuito impresso;
 - Motores dos vidros elétricos instalados
 - Rádio instalado;
 - Travas da portas instaladas;
 - Sirene instalada;
 - As centralinas instaladas;
 - Chaves de Simulação da porta e da ignição instaladas;
 - Lanternas instaladas;
 - Representação da Bomba Injetora instalada;
 - Computador de Bordo funcionando.
2. Software do Computador de Bordo em Assembly;
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.

6. A EQUIPE DE DESENVOLVIMENTO

A equipe de desenvolvimento contará com Anderson Gerez e César Mendes Chrusciel que serão responsáveis por realizar determinadas tarefas específicas. As tarefas serão distribuídas da seguinte maneira:

❖ **Anderson Gerez:**

- Circuito eletrônico;
- Esquemáticos e diagramas em bloco dos módulos eletrônicos;
- Desenho e produção da PCI;
- Estrutura mecânica em geral;
- Criação da IHM (Interface Homem-Máquina);
- Desenvolvimento do Software para o Microprocessador;
- Pesquisa de componente eletrônicos.

❖ **César Mendes Chrusciel:**

- Digitação da Documentação do Projeto;
- Pesquisas em geral sobre o projeto;
- Pesquisa de preços e locais para compra de componentes em geral;
- Interface gráfica;
- Criação da IHM (Interface Homem-Máquina);
- Circuito eletrônico;
- Estrutura mecânica em geral;
- Compra de componentes em geral.

7. FOTOS

8. 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.

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.

Não tivemos muitos problemas no desenvolvimento do projeto, somente foram queimados alguns 8051, porém, tendo em vista de outros projetos que já fizemos, esse já apresentamos uma maturidade maior na escolha do projeto tanto na escolha de material, montagem de PCI e demais.

Um dos problemas que mais preocupou o grupo foi agora na última semana, onde que faltando apenas 5 dias para a apresentação do projeto, tivemos um programa no software, no qual tendo como objetivo uma otimização do software, perdemos alguns dados e gravamos no 8051. Fixado no computador de bordo esse não executava as funções que lhe eram informadas, porém, fizemos um debug no programa e encontramos o erro, corrigindo-o.

O “Auto Car” se destaca por ser um projeto acadêmico que é inspirado nos sistemas de alarmes 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 computador de bordo que controla todas as funcionalidades do sistema.

Esse projeto serviu de grande valia no sentido de ser trabalhar com um sistema embarcado que possui um microprocessador operando as informações e também com a linguagem de programação Assembler que “dá vida” ao microprocessador, interpretando as informações e codificando-as.

9. ANEXOS

9.1 ANEXOS I – DICIONÁRIO DE TERMOS TÉCNICOS

Para facilitar a leitura para quem possa a vir a fazer a leitura do mesmo, foi feito 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 ($e = 1,602 \times 10^{-19}$ C). Por exemplo, se uma capacitância de 1 pF fosse carregada a uma tensão de 1 μ V, a equação perderia uma carga $Q = 10^{-19}$ 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 dq 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 μF)

C0G or NP0 - tipicamente de 4,7 pF a 0,047 μF , 5 %. Alta tolerância e performance de temperatura. Maiores e mais caros

X7R - tipicamente de 3300 pF a 0,33 μF , 10 %. Bom para acoplamento não-crítico, aplicações com timer.

Z5U - tipicamente de 0,01 μF a 2,2 μF , 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, verifica-se 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 dispositivo microeletrônico que consiste de muitos transistores e outros componentes interligados capazes de desempenhar muitas funções. Suas dimensões são extremamente reduzidas, os componentes são formados em pastilhas de material semicondutor.

A importância da integração está no baixo custo e alto desempenho, além do tamanho reduzido dos circuitos aliado à alta confiabilidade e estabilidade de funcionamento. 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 dos dispositivos eletrônicos.

No circuito integrado completo ficam presentes os transistores, condutores de interligação, componentes de polarização, e as camadas e regiões isolantes ou condutoras obedecendo ao seu projeto de arquitetura.

No processo de formação do chip, é fundamental que todos os componentes sejam implantados nas regiões apropriadas da pastilha. É necessário que a isolação seja 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 tetravalentes, 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 tetravalentes 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.

Após dopadas, cada face dos dois tipos de cristais (P e N) terá uma determinada característica diferente da oposta, gerando regiões de condução do 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).

MICROCONTROLADOR

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 super dimensionar. Cada desperdício será multiplicado pelo numero 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.

MICROPROCESSADOR 8051

O Intel 8051 faz parte de uma popular família de microcontroladores de 8 bits lançada pela Intel em 1977. É conhecido por sua facilidade de programação, em linguagem assembly graças ao seu poderoso conjunto de instruções. É tido como o microcontrolador mais popular do mundo, pois existem milhares de aplicações para o mesmo, e existem pelo menos dois mil fabricantes produzindo variantes e clones do modelo. Atualmente possui diversos modelos clones sendo produzidos por empresas diversas à Intel. Por ser um microcontrolador CISC, oferece um conjunto

de instruções muito vasto que permite executar desde um simples programa que faz piscar um LED até um programa de controle de acesso controlado por rede

O 8051 possui uma memória ROM que faz parte da arquitetura interna do chip, na qual será armazenado exclusivamente o programa que a CPU executará, não os dados, pois esses serão gravados em outra memória (RAM), que pode ser interna ou externa. A memória ROM tem a característica de poder ser gravada apenas uma vez, em geral, na fábrica. Este fato inviabiliza que os projetistas utilizem o 8051 em sua bancada. Para realizar projetos, normalmente utiliza-se o 8031 que não possui esta memória interna de programa (ROM), somente a de dados (RAM). Neste caso o programa é gravado numa memória externa muito conhecida e barata chamada EPROM e a gravação é feita por um equipamento também popular, chamado gravador de EPROM. Além disso, pode-se utilizar, uma RAM estática com excelente resultado, pois com ela, é possível criar o programa num computador pessoal e enviá-lo ao microcontrolador através de um cabo serial, sem a necessidade de qualquer outro equipamento auxiliar.

Uma outra versão (8751) também pode ser utilizada para desenvolvimento de projetos por apresentar uma EPROM como memória de programa integrada ao chip. Atualmente esta versão já está praticamente preterida por versões que apresentam memória FLASH, por exemplo o 8252. Mas enfim, se o leitor compreender o funcionamento básico do 8031, estará formando conhecimento didático para operar todos os membros desta família, sejam do fabricante original ou dos clones.

8051

O 8051 é o microcontrolador mais usado atualmente. É um dispositivo simples, mas de grande aplicação. Podemos encontrá-lo em circuitos lógicos que atuam na área da automação industrial, rádios, telefones celulares, microondas e etc. Fabricado pela Intel, o 8051 mostra-se como uma solução de baixo custo e facilidade de uso. Sua programação é feita em Assembly, e tem dois modos de funcionamento: 1) Modo Mínimo: onde são usados somente os recursos internos, não havendo a necessidade de outros componentes externos, possibilitando o uso das quatro portas para controle de I/O. 2) Modo Expandido: onde as memórias ROM e RAM são expandidas usando-se CIs externos. Mas tem a desvantagem de perder duas de suas portas para comunicação com as memórias externas.

RELÉ

Relé é um dispositivo eletromecânico ou não, com inúmeras aplicações possíveis em comutação de contatos elétricos. Servindo para ligar ou desligar dispositivos. É normal o relé estar ligado a dois circuitos. No caso do Relé eletromecânico, a comutação é realizada alimentando-se a bobina do mesmo. Quando uma corrente originada no primeiro circuito passa pela bobina, um campo eletromagnético é gerado, acionando o relé e possibilitando o funcionamento do segundo circuito. Sendo assim, uma das aplicabilidades do relé é utilizar-se de baixas correntes para o comando no primeiro circuito, protegendo o operador das possíveis altas correntes que irão circular no segundo circuito (contatos). Os tipos de relés existentes e suas aplicações tem uma grande diversidade em várias áreas como no setor de energia, por exemplo, um dos principais nichos do mercado de relés. Ao contrário do que a grande maioria das pessoas pensam, os relés não se limitam ao uso em carros. Pelo contrário, são largamente utilizados na indústria. Há também aplicações em automações residenciais e comerciais.

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 formação 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 miliôhms 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 voltagem 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 as taxas de voltagem e corrente especificadas, a maioria deles são taxadas em função de sua potência máxima, que é determinada pelo tamanho físico. As taxas 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 taxas 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 eletromotores.

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 uma coeficiente de temperatura positivo, ou seja se a temperatura sobe, a resistência também sobe. Os metais mais usado são a platina, daí as designação 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.

TRANSISTOR

O transistor (ou transistor) é um componente eletrônico que começou a se popularizar na década de 1950 tendo sido o principal responsável pela revolução da eletrônica na década de 1960, e cujas funções principais são amplificar e chavear sinais elétricos. O termo vem de transfer resistor (resistor de transferência), como era conhecido pelos seus inventores.

O processo de transferência de resistência, no caso de um circuito analógico, significa que a impedância característica do componente varia para cima ou para baixo da polarização pré-estabelecida. Graças à esta função, a corrente elétrica que passa entre coletor e emissor do transistor varia dentro de determinados parâmetros pré-estabelecidos pelo projetista do circuito eletrônico; esta variação é feita através da variação de corrente num dos terminais chamado base, que conseqüentemente ocasiona o processo de amplificação de sinal.

Entende-se por "amplificar" o procedimento de tornar um sinal elétrico mais fraco em mais forte. Um sinal elétrico de baixa intensidade, como os sinais gerados por um microfone, é injetado em um circuito eletrônico (transistorizado por exemplo), cuja função principal é transformar este sinal fraco gerado pelo microfone em sinais elétricos com as mesmas características mas com potência suficiente para excitar os alto-falantes, a este processo todo se dá o nome de ganho de sinal.

9.2 ANEXOS II – CÓDIGO FONTE DO SOFTWARE DO 8051

Auto_Car

```

;definição de constantes
;barramento de dados do LCD
#define LCDRS_OUT P1
;quando o LCDRS está em 0 ele coloca o LCD para receber comando, LCDRS=1
escreve dados no display
#define LCDRS P3.0
#define LCDHABILITA P3.1
#define COLUNA3 P0.1
#define COLUNA2 P0.2
#define COLUNA1 P0.3
#define LINHA4 P0.4
#define LINHA3 P0.5
#define LINHA2 P0.6
#define LINHA1 P0.7
#define INTERRUPT P3.5
#define CHAVE P0.0
#define SIRENE P3.4
#define CORTE P2.0

;início do programa
    org 0
;inicia timer0
início:
    mov    TMOD,#2d    ;
    mov    TCON,#10h   ;
    mov    TH0,#0FDh   ;
    mov    TL0,#0FDh   ;
    mov    P2,#0
    clr    SIRENE
    clr    CORTE
    acall  init_display
    mov    ArmazenaSenha,#0
    mov    ValidaSenha,#0
    mov    RegSenha,#0
    mov    TeclaPres,#0
    mov    Libera,#0
    mov    BitChave,#0
    mov    Contagem,#0
    mov    T255ms,#0
    mov    T40s,#0
    mov    T60s,#0
    mov    r2,#0
    mov    StatusPorta,#0
    mov    DPTR,#Cadastrar
    acall  Limpa_posiona
    acall  Imprime
    acall  Posiciona_linha2
volta:

    jnb    TF0,volta
    clr    TF0
/*

```

```

;quando a porta for aberta, após fechada e a ignição ligada, inicia-se a
contagem do tempo
;para que a senha seja digitada, caso não seja, inicia o processo de corte,
após 40s a sirene
;é disparada, ao chegar em 60s a ignição é cortada
*/
;Verifica se a variável contagem é igual a 1, caso for inicia contagem
    mov     a,Contagem
    cjne   a,#1,ContVerifica
;variável que contém 255ms
    mov     a, T255ms
    inc     a
    mov     T255ms,a
    cjne   a,#50,ContVerifica;255
;variável que contém 40s
    mov     a, T40s
    inc     a
    mov     T40s,a
;zerá o contador de 255ms
    mov     T255ms,#0
;verifica se a contagem chegou a 156, equivalente 40s, caso chegou dispara
a sirene
    cjne   a,#20,ContMaxima
    ;dispara a sirene
    setb   SIRENE
ContMaxima:
;verifica se a contagem chegou a 235, equivalente a 60s, caso chegou faz o
corte da ignição
    cjne   a,#30,ContVerifica
    clr    CORTE
    setb   P2.4
    mov    Contagem,#0
    mov    T40s,#0
;caso a contagem chega em 60s e a senha não foi digitada, cancela os
comandos
    mov    Libera,#0
;caso o comando foi liberado, poderá executar os comandos
ContVerifica:
    mov    a,Libera
    cjne   a,#1,MaqEstados
VerificaStatus:
;o bitChave indica que a porta foi aberta->fechada->chave ligada
    mov    a,BitChave
    cjne   a,#1,MaqStatus
    mov    Libera,#0
    mov    Contagem,#1
    mov    RegSenha,#0
    mov    ValidaSenha,#0
    mov    StatusPorta,#0
    mov    T255ms,#0
    mov    T40s,#0
;muda a máquina de estados para verificar senha
    mov    r2,#1
    mov    DPTR,#DigiteSenha
    acall  Limpa_posiona
    acall  Imprime
    acall  Posiciona_linha2
    jmp    volta
MaqStatus:
    acall  EsperaComando

```

```

        mov     a,StatusPorta
        cjne   a,#0,PortaFechar
        acall  PortaAberta
        jmp    volta
PortaFechar:
        cjne   a,#1,ChaveLigar
        acall  FechaPorta
        jmp    volta
ChaveLigar:
        acall  LigaChave
        jmp    volta

PortaAberta:
        jnb   INTERRUPT,EsperaFechar
        ret
EsperaFechar:
        mov   StatusPorta,#1
        ret
FechaPorta:
        jb   INTERRUPT,Fechou
        ret
Fechou:
        mov   StatusPorta,#2
        ret
LigaChave:
        jnb  CHAVE,DigitarSenha
        jnb  INTERRUPT,Reinicia
        ret
DigitarSenha:
        mov  BitChave,#1
        ret
Reinicia:
        mov          StatusPorta,#0
        ret

MaqEstados:
        cjne   r2,#0,Estado2
        acall  VerificaEstado1
Estado2:
        cjne   r2,#1,Estado3
        acall  VerificaEstado2
Estado3:
        cjne   r2,#2,Estado4
        jmp    volta
Estado4:
        jmp    volta

;o primeiro estado faz o cadastro de uma senha e armazena-a
VerificaEstado1:
        acall  Verifica_teclado
        ;verifica se alguma tecla foi pressionada, caso não foi, return
        mov   a,TeclaPres
        cjne  a,#0, TeclaPressionada
        ret
TeclaPressionada:
;espera soltar a tecla pressionada
        acall  EsperaSoltarTecla
        mov   a,RegSenha
        mov   TeclaPres,#0
        //

```

```

//
cjne    a, '#', VerificaCancela
mov     RegSenha, #0
;incrementa a máquina de estados
mov     r2, #1
;pede para o usuário digitar a senha
mov     DPTR, #DigiteSenha
acall   Limpa_posiona
acall   Imprime
acall   Posiciona_linha2
ret
VerificaCancela:
;caso a tecla digitada foi a cancela, limpa a senha armazenada e espera um
novo cadastro
cjne    a, '*', ArmazenaDigito
;pede para o usuário efetuar um novo cadastro
mov     DPTR, #Cadastrar
acall   Limpa_posiona
acall   Imprime
acall   Posiciona_linha2
;apaga registrado de digitos e variavel que armazena a senha
mov     RegSenha, #0
mov     ArmazenaSenha, #0
ret
ArmazenaDigito:
mov     a, ArmazenaSenha
add     a, RegSenha
mov     ArmazenaSenha, a
mov     BitIndica, #1
;registrador r1 é utilizado para enviar caracteres a ser escrito
mov     r1, '*'
acall   escreve_display
ret
;////////////////////////////////////
////////////////////////////////////
VerificaEstado2:
;após ter efetuado o cadastro, espera o usuário validar.
acall   Verifica_teclado
;verifica se alguma tecla foi pressionada, caso não foi, return
mov     a, TeclaPres
cjne    a, #0, ContinuaValidacao
ret
ContinuaValidacao:
;espera soltar a tecla pressionada
acall   EsperaSoltarTecla
mov     TeclaPres, #0
mov     a, RegSenha
;verifica se a tecla pressionada é o enter, caso seja, faz a validacao da
senha
cjne    a, '#', VerificaCancela2
mov     a, ValidaSenha
;faz uma subtração para ver se a senha digita é igual a senha cadastrada
subb    a, ArmazenaSenha
;caso seja o comando foi liberado
jz      ComandoLiberado
mov     RegSenha, #0
mov     ValidaSenha, #0
mov     DPTR, #SenhaIncorreta
acall   Limpa_posiona
acall   Imprime

```

```

    mov     a,#0
loopTime:
    inc     a
    cjne   a,#200,loopTime
    mov     a,#0
    mov     DPTR,#DigiteSenha
    acall  Limpa_posiona
    acall  Imprime
    acall  Posiciona_linha2
    ret
ComandoLiberado:
;incrementa a máquina de estados, pois a senha foi válida
    mov     r2,#2
    mov     ValidaSenha,#0
;a variável Libera indica que o comando foi liberado, ou seja, a senha foi
válida
    mov     Libera,#1
    mov     DPTR,#EntreComando
    acall  Limpa_posiona
    acall  Imprime
    mov     Contagem,#0
;libera comandos
    mov     BitChave,#0
;desliga a sirene
    clr     SIRENE
;libera a ignição para ligar o carro
    setb   CORTE
;zera contador 40 a 60s
    clr     P2.4
    mov     T40s,#0
;zera contados 255ms
    mov     T255ms,#0
    ret
VerificaCancela2:
    mov     a,RegSenha
;verifica se a tecla apertada foi CANCELA, caso seja zero registradores
    cjne   a,'#*',ArmazenaDigito2
;pede para o usuário digitar novamente a senha
    mov     DPTR,#DigiteSenha
    acall  Limpa_posiona
    acall  Imprime
    acall  Posiciona_linha2
    mov     RegSenha,#0
    mov     ValidaSenha,#0
    ret
ArmazenaDigito2:
    mov     a,ValidaSenha
    add    a,RegSenha
    mov     ValidaSenha,a
    mov     BitIndica,#1
    mov     r1,'#*'
    acall  escreve_display
    ret
;esta função verifica se alguma tecla ainda continua pressionada
EsperaSoltarTecla:
    clr     COLUNA1
    clr     COLUNA2
    clr     COLUNA3
    jb     LINHA1,PresLinha2
    jmp    EsperaSoltarTecla

```

```

PresLinha2:
    jb     LINHA2,PresLinha3
    jmp    EsperaSoltarTecla
PresLinha3:
    jb     LINHA3,PresLinha4
    jmp    EsperaSoltarTecla
PresLinha4:
    jb     LINHA4,ReturnLinha
    jmp    EsperaSoltarTecla
ReturnLinha:
    ret

;função que verifica teclas do teclado
Verifica_teclado:
    ;mov    P0,#0Eh
loop1:
    clr    COLUNA1
    jnb    LINHA1,UM
    jnb    LINHA2,QUATRO
    jnb    LINHA3,SETE
    jnb    LINHA4,CANCELA
    setb   COLUNA1
loop2:
    clr    COLUNA2
    jnb    LINHA1,DOIS
    jnb    LINHA2,CINCO
    jnb    LINHA3,OITO
    jnb    LINHA4,ZERO
    ;coluna pra nível 1 as colunas
    setb   COLUNA2
loop3:
    clr    COLUNA3
    jnb    LINHA1,TRES
    jnb    LINHA2,SEIS
    jnb    LINHA3,NOVE
    jnb    LINHA4,ENTER
    setb   COLUNA3
loop4:
    ret
UM:
    mov    RegSenha,#'1'
    mov    TeclaPres,#1
    acall  Delay
    ret
DOIS:
    mov    RegSenha,#'2'
    mov    TeclaPres,#1
    acall  Delay
    ret
TRES:
    mov    RegSenha,#'3'
    mov    TeclaPres,#1
    acall  Delay
    ret
QUATRO:
    mov    RegSenha,#'4'
    mov    TeclaPres,#1
    acall  Delay
    ret
CINCO:

```

```

        mov     RegSenha,#'5'
        mov     TeclaPres,#1
        acall   Delay
        ret
SEIS:
        mov     RegSenha,#'6'
        mov     TeclaPres,#1
        acall   Delay
        ret
SETE:
        mov     RegSenha,#'7'
        mov     TeclaPres,#1
        acall   Delay
        ret
OITO:
        mov     RegSenha,#'8'
        mov     TeclaPres,#1
        acall   Delay
        ret
NOVE:
        mov     RegSenha,#'9'
        mov     TeclaPres,#1
        acall   Delay
        ret
CANCELA:
        mov     RegSenha,#'*'
        mov     TeclaPres,#1
        acall   Delay
        ret
ZERO:
        mov     RegSenha,'#0'
        mov     TeclaPres,#1
        acall   Delay
        ret
ENTER:
        mov     RegSenha,'##'
        mov     TeclaPres,#1
        acall   Delay
        ret
Limpa_posiona:
; Apaga display
        mov     BitIndica,#0           ;escrevendo comando
        mov     r1,#01h                ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;
        mov     r1,#080h                ;linha0 coluna0
        acall   escreve_display
        ret
Posiciona_linha2:
        mov     BitIndica,#0           ;escrevendo comando
        mov     r1,#0C0h
        acall   escreve_display
        ret
;função que imprime no display
Imprime:
        MOV     R0,#0
loop:
        MOV     A,R0
        MOVC   A,@A+DPTR
        JZ     FIM

```

```

        INC     R0
        mov     r1,a
        mov     BitIndica,#1h      ;o BitIndica diz se é comando ou dado
        acall   escreve_display;
        JMP     loop
FIM:
        ret
init_display:
; barramento tem 8 bits
        mov     BitIndica,#0h      ;o BitIndica diz se é comando ou dado
        mov     r1,#30h           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;

; barramento tem 8 bits
        mov     r1,#30h           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;

; barramento tem 8 bits
        mov     r1,#30h           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;

; Function set (8-bit interface, 2 linhas,matriz de 5*7)
        mov     r1,#38h           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;

; Desliga o cursor
        mov     r1,#0Ch           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;

; Apaga display
        mov     r1,#01h           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;

; Seleção de modo de entrada (Entry mode set )
        mov     r1,#06h           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display ;

        mov     r1,#80h           ;o acumulador tem o dado a
ser escrito no LCD
        acall   escreve_display;

        ret

escreve_display:
        mov     P1,r1             ;P3 é o barramento do LCD
        mov     a,BitIndica       ;recupera valor do BitIndica
;se o acumulador é zero, então a escrita é de comando
        jz     comando           ;
;LCDRS=1 quando é dados
        setb    P3.1              ;pino RS
        jmp    enable            ;
comando:
;LCDRS=0 quando é comando

```

```

        clr    P3.1                ;pino RS
enable:
;dá uma borda de descida no pino habilita, para autenticar comando
        setb  P3.0                ;pino E
        clr   P3.0                ;pino E
        ret
;esta função fica esperando algum comando
EsperaComando:
        mov   TeclaPres,#0
        acall Verifica_teclado
;verifica se alguma tecla foi pressionada, caso não foi, return
        mov   a,TeclaPres
        cjne  a,#0, VerificaComando
        ret
VerificaComando:
;espera soltar a tecla pressionada
        acall EsperaSoltarTecla
        mov   a,RegSenha
;caso a tecla # tenha sido pressionada, o menu de funções será mostrado
        subb  a,#'#'
        jz    Menu
        jmp   VerProximo
Menu:
        acall VerMenu
        ret
VerProximo:
;caso a tecla 1 foi pressionada, o vidro irá subir os vidros
        mov   a,RegSenha
        cjne  a,#'1',prox
        jmp   SobeVidro
;caso a tecla 4 foi pressionada, o vidro irá descer os vidros
prox:
        cjne  a,#'4',prox1
        jmp   DesceVidros
;caso a tecla 2 foi pressionada, o botao1_0 será acionado
prox1:
        cjne  a,#'2',prox2
        jmp   AcionaBotao1_0
;caso a tecla 5 foi pressionada, o botao1_1 será acionado
prox2:
        cjne  a,#'5',p4
        jmp   AcionaBotao1_1
;caso a tecla 3 foi pressionada, o botao2_0 será acionado
p4:
        cjne  a,#'3',p5
        jmp   AcionaBotao2_0
;caso a tecla 6 foi pressionada, o botao2_1 será acionado
p5:
        cjne  a,#'6',p6
        jmp   AcionaBotao2_1
;caso a tecla 7 foi pressionada, o travaPorta será acionado
p6:
        cjne  a,#'7',p7
        jmp   Trava
;caso a tecla 8 foi pressionada, o Destrava será acionado
p7:
        cjne  a,#'8',p8
        jmp   Destrava
;caso a tecla 9 foi pressionada, o LDRádio será acionado
p8:

```

```

        cjne    a,#'9',p9
        jmp     LDRadio
;caso a tecla 0 foi pressionada, o DesRadio será acionado
p9:
        cjne    a,#'0',p10
        jmp     DesRadio
p10:
        ret
;funções relacionadas aos comandos
SobeVidro:
        mov     RegSenha,#0
        mov     DPTR,#FSOBE
        acall   Posiciona_linha2
        acall   Imprime
        setb    P2.6
        acall   Delay
        clr     P2.6
        ret
DesceVidros:
        mov     RegSenha,#0
        mov     DPTR,#FDESCE
        acall   Posiciona_linha2
        acall   Imprime
        setb    P2.5
        acall   Delay
        clr     P2.5
        ret
AcionaBotao1_0 :
        mov     RegSenha,#0
        mov     DPTR,#FBOTAO1_0
        acall   Posiciona_linha2
        acall   Imprime
        setb    P3.7
        acall   Delay
        clr     P3.7
        ret
AcionaBotao1_1 :
        mov     RegSenha,#0
        mov     DPTR,#FBOTAO1_1
        acall   Posiciona_linha2
        acall   Imprime
        setb    P3.6
        acall   Delay
        clr     P3.6
        ret
AcionaBotao2_0 :
        mov     RegSenha,#0
        mov     DPTR,#FBOTAO2_0
        acall   Posiciona_linha2
        acall   Imprime
        setb    P3.2
        acall   Delay
        clr     P3.2
        ret
AcionaBotao2_1 :
        mov     RegSenha,#0
        mov     DPTR,#FBOTAO2_1
        acall   Posiciona_linha2
        acall   Imprime
        setb    P3.3

```

```

    acall Delay
    clr P3.3
    ret
Trava :
    mov RegSenha,#0
    mov DPTR,#FTRAVA
    acall Posiciona_linha2
    acall Imprime
    setb P2.3
    acall Delay
    clr P2.3
    ret
Destrava :
    mov RegSenha,#0
    mov DPTR,#FDESTRAVA
    acall Posiciona_linha2
    acall Imprime
    setb P2.2
    acall Delay
    clr P2.2
    ret
LDRadio :
    mov RegSenha,#0
    mov DPTR,#FDESRADIO
    acall Posiciona_linha2
    acall Imprime
    setb P2.1
    ret
DesRadio :
    mov RegSenha,#0
    mov DPTR,#FLIGARADIO
    acall Posiciona_linha2
    acall Imprime
    clr P2.1
    ret
VerMenu:
    mov RegSenha,#0
    mov DPTR,#FFDESRADIO
    acall Limpa_posiona
    acall Imprime
    acall Posiciona_linha2
    mov DPTR,#FFSOBE
    acall Imprime
    acall Delay1
    mov DPTR,#FFBOTA01_0
    acall Limpa_posiona
    acall Imprime
    acall Posiciona_linha2
    mov DPTR,#FFBOTA02_0
    acall Imprime
    acall Delay1
    mov DPTR,#FFDESCE
    acall Limpa_posiona
    acall Imprime
    acall Posiciona_linha2
    mov DPTR,#FFBOTA01_1
    acall Imprime
    acall Delay1
    mov DPTR,#FFBOTA02_1
    acall Limpa_posiona

```

```

acall  Imprime
acall  Posiciona_linha2
mov    DPTR,#FFTRAVA
acall  Imprime
acall  Delay1
mov    DPTR,#FFDESTRAVA
acall  Limpa_posiona
acall  Imprime
acall  Posiciona_linha2
mov    DPTR,#FFLIGARADIO
acall  Imprime
acall  Delay1

mov    DPTR,#EntreComando
acall  Limpa_posiona
acall  Imprime
acall  Posiciona_linha2
ret

;função delay
Delay:
mov    a,#0
EsperaDelay:
jnb    TF0,EsperaDelay
clr    TF0
inc    a
cjne  a,#150,EsperaDelay
ret

;função delay
Delay1:
mov    r4,#0
mov    r3,#0
EsperaDelay1:
jnb    TF0,EsperaDelay1
clr    TF0
mov    a,r4
inc    a
mov    r4,a
cjne  r4,#255,EsperaDelay1
mov    a,r3
inc    a
mov    r3,a
mov    r4,#0
cjne  r3,#4,EsperaDelay1
mov    r3,#0
ret

Digite_Senha:          DB    'DIGITE A SENHA',0
Cadastrar:            DB    'EFETUE CADASTRO',0
CadastroEfetuado:    DB    'CADASTRO EFETUADO',0
EntreComando:        DB    'DIGITE O COMANDO',0
SenhaIncorreta:      DB    'SENHA INCORRETA',0
DigiteSenha:          DB    '*DIGITE A SENHA*',0
FFSOBE:               DB    '1 - SOBE VIDRO',0
FFDESCE:              DB    '4 - DESCE VIDRO',0
FFBOTA01_0:           DB    '2 - *S. V. ESQ*',0
FFBOTA01_1:           DB    '5 - *D. V. ESQ*',0
FFBOTA02_0:           DB    '3 - *S. V. DIR*',0
FFBOTA02_1:           DB    '6 - *D. V. DIR*',0
FFLIGARADIO:         DB    '9 - *DES. RADIO',0

```

```

FFDESRADIO:          DB '0 - *LIGA RADIO',0
FFTRAVA:             DB '7 - TRAVA PORTA',0
FFDESTRAVA:         DB '8 - *DEST PORTA',0

FSOBE:               DB '***SOBE VIDRO***',0
FDESCE:             DB '***DESCE VIDRO***',0
FBOTAO1_0:          DB '*SOBE VIDRO ESQ*',0
FBOTAO1_1:          DB 'DESCE VIDRO ESQ',0
FBOTAO2_0:          DB '*SOBE VIDRO DIR*',0
FBOTAO2_1:          DB 'DESCE VIDRO DIR',0
FLIGARADIO:         DB 'CD PLAYER LIGADO',0
FDESRADIO:          DB 'CD PLAYER DESLIG',0
FTRAVA:             DB 'PORTAS TRAVADAS*',0
FDESTRAVA:         DB 'PORTAS DESTRAVA*',0

```

```

        DSEG AT 0x40
BitIndica: DS 1
ArmazenaSenha: DS 1
ValidaSenha: DS 1
RegSenha: DS 1
TeclaPres: DS 1
T255ms: DS 1
T40s: DS 1
T60s: DS 1
StatusPorta: DS 1
Confirma: DS 1
Libera: DS 1
BitChave: DS 1
Contagem: DS 1
        end

```

LCD

```

;definição de constantes
;barramento de dados do LCD
#define LCDRS_OUT P3
;quando o LCDRS está em 0 ele coloca o LCD para receber comando, LCDRS=1
escreve dados no display
#define LCDRS P3.0
#define LCDHABILITA P3.1

;início do programa
        org 0
;inicia timer0
início:
        mov TMOD,#2d ;
        mov TCON,#10h ;
        mov TH0,#6d ;
        acall init_display;
;inicializa o display
inicializa_horas:
        mov Horas,#0h ;zera horas

inicializa_minutos:
        mov Minutos, #0h ;zera minutos

inicializa_segundos:
        mov Segundos, #0h ;zera segundos

```

```

mostra_display:
    acall    atualiza_display

volta:
    jnb     TF0,volta
    clr     TF0

atualiza_display:
    mov     a,#84h                ;posiciona cursor
    mov     BitIndica,#0          ;escrevendo comando
    acall   escreve_display
;a partir deste ponto o que será escrito no display é dados
    mov     BitIndica,#1
    mov     a,Horas
    mov     r6,a                  ;mostra horas
    acall   ConverteValor_SendSerial
    mov     BitIndica,#1
    mov     a,#':'                ;mostra :
    acall   escreve_display
    mov     BitIndica,#1
    mov     a,Minutos
    mov     r6,a                  ;mostra minutos
    acall   ConverteValor_SendSerial
    mov     BitIndica,#1
    mov     a,#':'                ;mostra :
    mov     r0,#1d
    acall   escreve_display
    mov     BitIndica,#1
    mov     a,Segundos
    mov     r6,a                  ;mostra segundos
    acall   ConverteValor_SendSerial
    ret

ConverteValor_SendSerial:
;antes de escrever no display tem que converter o número pra hexadecimal
    acall   Hexa_Decimal
    mov     a,r7                  ;Envia o resultado
    mov     BitIndica,#1
    acall   escreve_display
    mov     a,r6                  ;Recupera resto
    mov     BitIndica,#1
    acall   escreve_display      ;Envia o resto
    ret

Hexa_Decimal:
    mov     r7,#25d
    mov     a,r7
loop_div_10:
    rl     a
    rl     a
    add    a,r7
    rl     a
    mov    r0,a
    mov    a,r6
    subb  a,r0
    jc    decrementa_r7
;conversao para ascii
    add    a,'#0'
    mov    r6,a
    mov    a,r7
    add    a,'#0'

```

```

        mov     r7,a
        ret
decrementa_r7:
    dec     r7
    mov     a,r7
    jmp     loop_div_10
init_display:
; barramento tem 8 bits
    mov     BitIndica,#0h      ;o BitIndica diz se é comando ou dado
    mov     a,#30h             ;o acumulador tem o dado a
ser escrito no LCD
    acall   escreve_display;

; barramento tem 8 bits
    mov     a,#30h             ;o acumulador tem o dado a
ser escrito no LCD
    acall   escreve_display;

; barramento tem 8 bits
    mov     a,#30h             ;o acumulador tem o dado a
ser escrito no LCD
    acall   escreve_display;

; Function set (8-bit interface, 2 linhas,matriz de 5*7)
    mov     a,#38h             ;o acumulador tem o dado a
ser escrito no LCD
    acall   escreve_display;

; Desliga o cursor
    mov     a,#0Ch             ;o acumulador tem o dado a
ser escrito no LCD
    acall   escreve_display;

; Apaga display
    mov     a,#01h             ;o acumulador tem o dado a
ser escrito no LCD
    acall   escreve_display;

; Seleção de modo de entrada (Entry mode set )
    mov     a,#06h             ;o acumulador tem o dado a ser
escrito no LCD
    acall   escreve_display ;
    ret

escreve_display:
    mov     LCDRS_OUT,a        ;P3 é o barramento do LCD
    mov     a,BitIndica        ;recupera valor do BitIndica
;se o acumulador é zero, então a escrita é de comando
    jz     comando            ;
;LCDRS=1 quando é dados
    setb    LCDRS              ;pino RS
    jmp     enable            ;
comando:
;LCDRS=0 quando é comando
    clr     LCDRS              ;pino RS
enable:
;dá uma borda de descida no pino habilita, para autenticar comando
    setb    LCDHABILITA        ;pino E
    clr     LCDHABILITA        ;pino E
    ret

```

```
        DSEG AT 0x40
BitIndica: DS    1
Horas:      DS    1
Minutos:    DS    1
Segundos:   DS    1
        end
```

9.3 ANEXOS III – DATASHEETS