

Aluno: Edwino Alberto Lopes Stein

Matrícula: 1201324411

Disciplina: Introdução a Sistemas Embarcados (2016.2)

Professor: Herbert Oliveira Rocha

Instituição: Universidade Federal de Roraima

1. Defina sistemas embarcados e apresente 3 exemplos com justificativa.

São sistemas computacionais designados a um propósito específico, e geralmente integram um sistema maior. Em relação a sistemas tradicionais (desktops, laptops, servidores), são equipados geralmente com microcontroladores ao invés de microprocessadores, isso permite mais simplicidade, otimização e barateamento do hardware com o intuito de que funcionem de forma ubíqua (invisível) e pervasiva (todo lugar), onde atuam sob ambientes reativos e geralmente com muitas restrições (energética, temporal, etc).

Exemplos:

- **Navegação aeronáutica:** Sistemas críticos que auxiliam na pilotagem de aeronaves em condições diversas: transponder; sistema de medição de altitude; radar.
- **Computadores de bordo automotivo:** Conjunto de sistemas embarcados e sensores com a finalidade de controlar e supervisionar sistemas automatizados do veículo: ABS, sistema de detecção de chuva do limpador de para-brisa, central de multimídia.
- **Equipamentos hospitalares:** Sistemas críticos para monitoramento e suporte de vida para pacientes: monitor cardíaco, máquina de raios X.

2. Apresente uma comparação entre sistemas embarcados e sistemas tradicionais.

Enquanto o foco dos equipamentos de propósito geral (desktops, laptops, servidores) é garantir um alto desempenho para realização de tarefas complexas e sem um contexto específico (editar textos, navegar na internet, reproduzir áudio e vídeo), os sistemas embarcados são desenvolvidos com um foco específico e geralmente único, isso permite que exista uma otimização de hardware e software e consequentemente um barateamento, pois geralmente não será necessário um grande poder computacional e recursos em abundância para executar tarefas de controle. Outro ponto divergente é o fato de que sistemas embarcados trabalham sob restrições mais severas, onde são obrigados a garantir o tratamento de eventos relativos a eventos externos em tempo real e satisfazer requisitos de segurança, durabilidade, autonomia e confiabilidade.

3. Descreva os que são restrições temporais e de consumo de energia.

Restrições temporais: São restrições que definem que as tarefas devem ser executadas em tempo hábil e previsível, pois o tempo de resposta é crucial para o efetivo funcionamento do sistema como um todo. Um exemplo é um radar de controle de voo, pois o tempo de resposta pode ser decisivo para evitar uma possível colisão entre aeronaves em zonas muito movimentadas, como um aeroporto.

Restrições de energia: São restrições que devem ser levado em consideração principalmente quando o sistema necessita de uma autonomia consideravelmente longa, pois em muitos casos a manutenção e reposição do equipamento podem ser dificultadas por diversas restrições do projeto (distância da área de atuação, acessibilidade). Um exemplo seria uma estação meteorológica no meio da Amazônia, pois o sistema embarcado deve ser capaz de ter uma autonomia e eficiência energética grande, pois será muito custoso repor as fontes de energia, como baterias.

4. O que é hardware/software codesign? Descreva as etapas (fases)?

É uma metodologia de desenvolvimento em que o foco é especificar a estrutura e comportamento de sistemas em uma máquina finita de estados possibilitando assim uma série de testes, simulações e verificações formais para determinar quais componentes de hardware e software serão sintetizados no sistema final. Componentes de hardware são geralmente traduzidos em portas lógicas e circuitos integrados, e componentes de software são traduzidos em linguagem de programação de baixo nível. É geralmente composto pelas etapas:

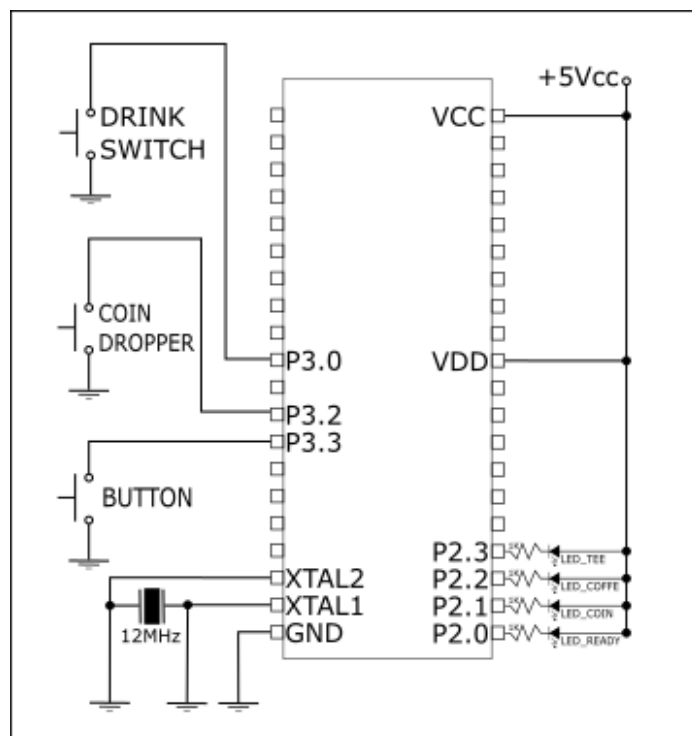
- **Especificação:** Descrição dos requisitos do sistema com alto nível de abstração e descrição dos requisitos funcionais e não-funcionais;
- **Particionamento:** Define quais componentes de hardware e software devem ser implementados de acordo com as métricas (custo, tempo de execução, consumo de energia, área de memória, tamanho de programas, etc);
- **Co-síntese:** Também chamado de protótipo virtual, é o resultado do particionamento, onde são apresentados os módulos comunicantes de hardware e software que são capazes de satisfazer as restrições;
- **Análise e Validação:** Consiste em prover métricas de qualidade.

5. Defina Microprocessador e Microcontrolador.

Microprocessador: Circuito integrado composto por transistores que trabalham em conjunto com o intuito de possibilitar a realização de uma grande variedade de tarefas que serão descritas através de um software. Um microprocessador geralmente necessita de alguns periféricos (memória RAM, timers, etc) para poder funcionar adequadamente.

Microcontrolador: É um microprocessador de propósito específico e geralmente contém todos os componentes necessários para funcionar (SoC, System on a Chip), e com foco na execução de softwares de controle.

6. Dado o fluxo de estados abaixo de uma máquina de café e chá. Implemente este projeto no micro-controlador 8051, apresentando: Uma descrição da pinagem utilizada para o projeto; a máquina de estados de Mealy usando a tabela de transição de estados e o diagrama; o código da máquina de Mealy no micro-controlador 8051. Descrever a sua solução com o máximo de detalhes possível, por exemplo, apresentando testes com simulador, inclusive descreva a forma como os testes foram feitos.



Esquema do circuito.

Descrição: O sistema espera que o usuário selecione a bebida através do interruptor conectado em P3.0 (ligado para café representado pelo LED P2.2, e desligado para chá representado pelo LED P2.3), e aguarda a inserção da moeda pelo sensor conectado em P3.2. Quando inserida, o LED P2.1 irá acender para indicar a presença da moeda. Após a inserção, o usuário poderá pressionar o botão conectado em P3.3 para que a bebida selecionada seja servida. O processo será indicado pelo piscar do LED P2.0 e estará completo quando o mesmo parar de piscar e permanecer ligado, indicando que o sistema está pronto para a próxima bebida.

Executando o programa: O código fonte se encontra no arquivo "src/questao6.zip". Para executá-lo, basta extrair os arquivos, e abrir o projeto no ambiente de desenvolvimento Keil, e compilá-lo. Para os testes consideramos que os LEDs P2.0, P2.1, P2.2 e P2.3 representam respectivamente: sistema pronto (aceso) ou em

atividade (piscando); moeda presente ou ausente, café selecionado; chá selecionado. O controle da bebida será feita através do interruptor P3.0 (ligado para café e desligado para chá). Para detectar a moeda, basta mudar desativar a porta P3.2 e o LED da bebida correspondente será ligado (P2.2 ou P2.3). Para iniciar o processo de servir a bebida, basta ligar a porta P3.3. Enquanto a bebida é servida, o LED 2.0 ficará piscando durante algum tempo até parar de piscar e o sistema será reiniciado.

7. Pesquise e selecione 2 micro-controladores ou plataformas com microprocessadores (exceto o 8051) e apresente o esquema de pinagem (onde está localizado e sua função) para programação. Adicionalmente, apresente 1 exemplo de código utilizando a pinagem da placa para cada escolha.

Raspberry PI (A e B original):

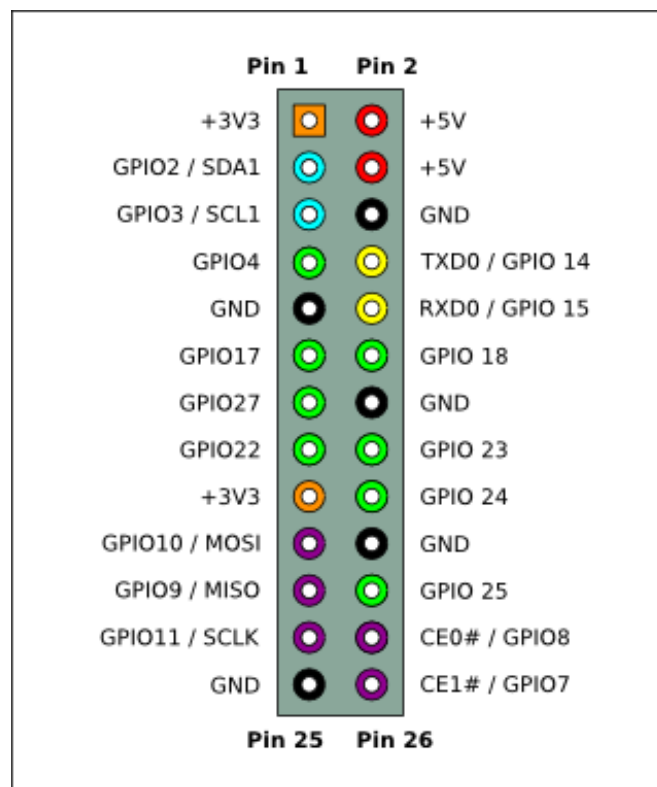


Imagem disponível em <http://elinux.org/RPi_Low-level_peripherals#Model_A_and_B_.28Original.29>

```
1 from gpiozero import LED
2 from time import sleep
3
4 led = LED(4)      # Define o LED no pino 4
5 led.on()          # Liga o LED
6 sleep(5)          # Espera 5 segundos
7 led.off()         # Desliga o LED
```

Script Phyton para piscar um LED.

Arduino Uno R2:

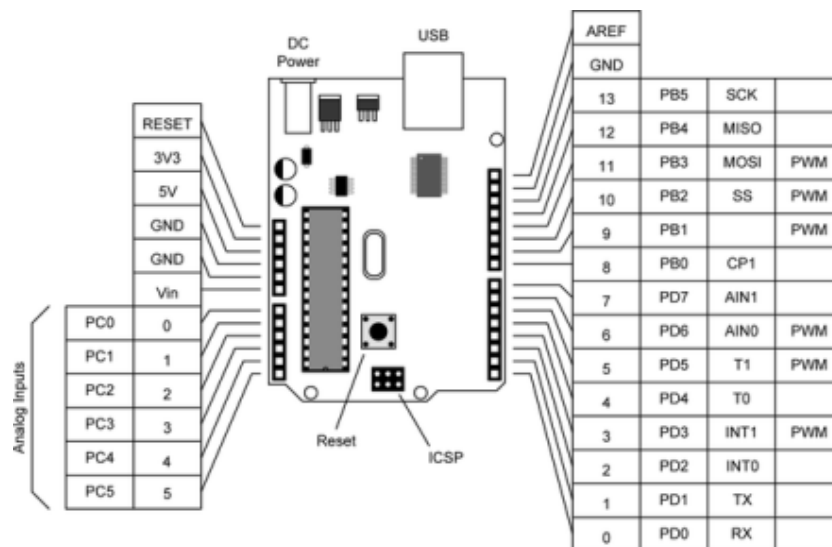


Imagem disponível em <https://www.safaribooksonline.com/library/view/arduino-a-technical/9781491934319/ch04.html#baseline_boards_table>

```

1
2 // Inicializa a placa
3 void setup() {
4     // Define o pino LED_BUILTIN (13) como output
5     pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 // Programa principal
9 void loop() {
10    // Liga o LED
11    digitalWrite(LED_BUILTIN, HIGH);
12    // Espera 5 segundos
13    delay(5000);
14    // Desliga o LED
15    digitalWrite(LED_BUILTIN, LOW);
16 }

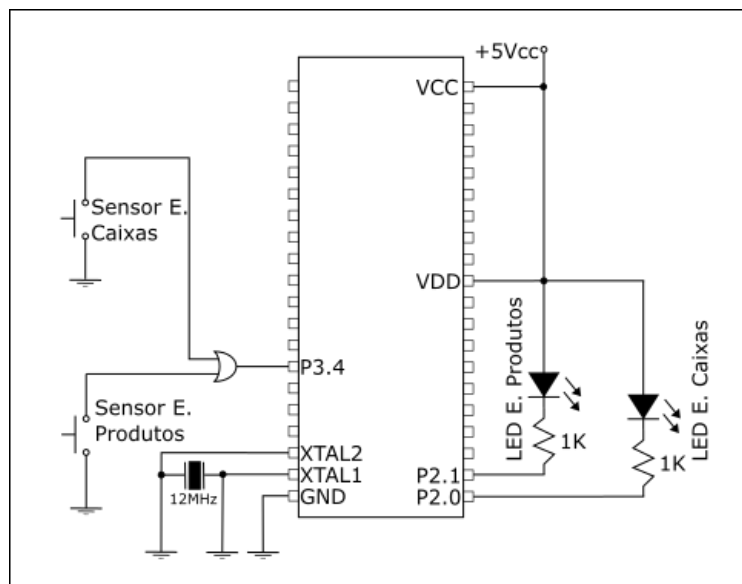
```

Programa para piscar um LED.

8. Descreva as funções da interrupção por Timers/Counters no microcontrolador 8051.

São periféricos que funcionam indiretamente pelo cristal de clock e serve como contador de ciclos de máquina, e pode ser utilizado como uma forma de medir o tempo com certa precisão. A cada ciclo de máquina, é incrementado o registrador contador de 16 bits (registradores TL0 e TH0, ambos de 8 bits), e a cada estouro (overflow de 65535 + 1 ciclos de máquina) do contador, é gerada uma interrupção, onde a execução do programa é pausada para que seja executada outro trecho de código definido pelo programador. Outra funcionalidade é a contagem de pulsos recebidos pela porta 3.4.

9. Imagine um sistema de embalagem de produtos. O sistema controla esteiras, uma com produtos e outra com caixas de embalagem. No final da esteira de produtos, o produto é "derrubado" dentro de uma caixa que está na esteira logo abaixo. A caixa tem certa capacidade de produtos. Um sensor no final da esteira de produtos (um feixe de luz, por exemplo) detecta a queda do produto e envia um pulso a cada produto. Esse sinal deve ser conectado ao pino que mapeia a entrada do contador (timer/counter), o qual deve ser ajustado para a capacidade da caixa. Após o enchimento da caixa é gerada uma interrupção. O tratamento dessa interrupção deve então: parar a esteira de produtos; andar a esteira de caixas para posicionar uma nova caixa; e acionar novamente o contador e a esteira de produtos. Fazer um programa para o micro-controlador 8051 de tal forma que o sistema fique livre para quaisquer outras tarefas enquanto a caixa não está cheia, ou seja, evitar o busy-waiting. Descrever a sua solução com o máximo de detalhes possível, inclusive a forma como os testes foram feitos. Fazer quaisquer suposições que se fizerem necessárias, por exemplo, mesmo não tendo o hardware do sensor, supor que o mesmo envia o pulso, o qual é lido por um pino do micro-controlador.



Esquema do circuito.

Descrição: Enquanto a máquina está trabalhando, o sistema fica aguardando uma caixa passar pelo sensor de caixas conectado na porta 3.4, quando isso ocorre, é disparada a interrupção de estouro de contador, que por sua vez interrompe a atividade da esteira de caixas na porta 2.0, e então reinicia o contador para a quantidade definida de produtos que será incrementado através do sensor de produtos, também conectado a porta 3.4, então é iniciada a atividade da esteira de produtos na porta 2.1 e o sensor de produtos ficará aguardando os produtos passar para incrementar o contador. Quando ocorre o estouro do contador de produtos, a atividade da esteira de produtos será interrompida e o contador será reiniciado para esperar a próxima caixa na esteira de caixas que será religada e então o processo reinicia.

Executando o programa: O código fonte se encontra no arquivo “src/questao9.zip”. Para executá-lo, basta extrair os arquivos e abrir o projeto no ambiente de desenvolvimento Keil, e compilá-lo. Para os testes consideramos que os LEDs P2.0 e P2.1 significam o funcionamento das esteiras das caixas e produtos respectivamente. Quando o LED da esteira das caixas (P2.0) está ativo, significa que a esta esteira está em movimento e o sensor de caixas está esperando uma caixa, ou seja, ela irá parar quando o houver um pulso (desliga e liga) em P3.4. Quando o LED da esteira de produtos (P2.1) estiver ativo, significa esta esteira está em movimento e que P3.4 estará esperando pulsos para incrementar o contador (definido para 5 pulsos). Quando for atingido o limite de pulsos (produtos), o processo reinicia.