

# Lista 1

Pedro Daniel da Silva Gohl

Dezembro 2016

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

### 1.1 Sistemas Embarcados

Um sistema embarcado é um sistema embutido feito para um uso específico incorporado a outro sistema, controlado por instruções armazenadas dentro de um chip, as instruções armazenadas na ROM são executadas por um microprocessador.

### 1.2 Microondas:

- **Painel de entrada:** responsável por receber entradas;
- **Controlador display de LCD:** responsável por exibir as informações dadas pelo usuário;
- **Controlador de micro-ondas:** responsável por ativar e desativar as micro-ondas que aquecem a comida.

### 1.3 Geladeira:

- **Painel de frontal:** responsável por receber comandos;
- **Controlador de temperatura:** responsável por diminuir e aumentar a temperatura e desativar o compressor;

### 1.4 Ar-Condicionado:

- **Controlador de temperatura:** responsável por diminuir e aumentar a temperatura;

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

### 2.1 Sistema Embarcado x Sistemas Tradicionais

#### Sistema embarcados:

- **Preço:** Sistemas embarcados são mais baratos que sistemas tradicionais;
- **Praticidade:** Sistemas embarcados usam micro-controladores, que por sua vez já contém processador, memória e interface externa;
- **Limitado:** Sistemas embarcados possuem memória e processamento limitados;
- **Restritos:** Sistemas embarcados estão restringidos a trabalhar somente para aquilo que foi desenvolvido.

#### Sistemas tradicionais:

- **Poder:** Sistemas tradicionais conseguem atingir níveis de processamento muito maiores em comparação à um sistema embarcado;
- **impraticidade** Sistemas tradicionais usam micro-processadores, que precisam de memória externa, e outros componentes para funcionar adequadamente;
- **Funcionalidade:** Sistemas tradicionais podem ser usados em diversas áreas ao mesmo tempo.

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

### 3.1 Restrições temporais

O sistema precisa levar em consideração a criticidade do sistema e ser preciso. Em sistemas estritos, todas as restrições temporais tem de ser satisfeitas para evitar falha do sistema, onde o custo da falha é muito alta. Em Sistemas brandos, podem eventualmente não atender essas restrições.

### 3.2 Consumo de energia

Os componentes do sistema devem levar em consideração eficiência de energia, o processador é um dos maiores consumidores de energia, escolher um com boa autonomia em conjunto com uma boa lógica. conhecer o componente é ideal para implementações com boa eficiência, levando em consideração os *trade-offs*.

## 4 O que é hardware/software codesign? Descreva as etapas.

### 4.1 Hardware / Software Codesign

Na construção do sistema, devem ser implementados algumas metodologias de desenvolvimento, tendo isso em consideração, podemos fazer o levantamento das especificações do sistema, testes e validações até sua síntese usando HW / SW codesign.

As verificações e mapeamento são particionadas em vários passos intermediários conduzidos por ferramentas. Os modelos são descritos com alto nível de abstração e refinado até os níveis mais baixos, as propriedades são verificadas em cada nível.

- **Especificação:** Descrição dos requisitos do sistema com alto nível de abstração e especificação de requisitos funcionais e não-funcionais;
- **Particionamento:** Descrição dos componentes a serem implementados em hardware e/ou software levando em consideração as métricas;
- **Co-Síntese:** Descrição dos conjuntos de módulos comunicantes. Usando o resultado do particionamento para criação de um protótipo real, baseado no protótipo virtual, de forma que todas as restrições sejam satisfeitas;
- **Análise e validação:** Podendo ser feita após cada fase, a análise e validação, consistem em prover métricas de qualidade.

## 5 Defina Microprocessador e Microcontrolador.

### 5.1 Microprocessador:

Microprocessadores é um circuito integrado que contém milhares de transistores. Os transistores trabalham para armazenar e manipular dados, facilitando para execução de uma gama variada de funções. Para poder funcionar o microprocessador precisa de outros periféricos. Sua Unidade de lógica aritmética pode ser mais poderosa, porém utiliza-se de componentes externos, diferente da de um Microcontrolador.

### 5.2 Microcontrolador:

Microcontrolador é um microprocessador de propósito especial, contendo todos os periféricos necessários para seu funcionamento básico: controlar. a Unidade de lógica aritmética não é tão potente como a de um microprocessador, porém tem tudo que precisa para funcionar no chip.

## 6 Descreva as funções da interrupção por Timers/Counters no micro-controlador 8051.

### 6.1 Timers/Counters

O timer é um temporizador preciso, toda vez que o timer conclui sua temporização o programa é interrompido e então a função definida é executada. Seu acionamento é feito de forma indireta por um cristal ligado ao microcontrolador, esse *clock* é dividido por 12 antes de entrar nos Timers. O oscilador interno e o cristal geram os *ciclos de máquina*, cada ciclo de oscilação gera um pulso e a cada dois pulsos, contabilizam um *State*, uma sequência de seis estados contabilizam um *ciclo de máquina*.

## 7 Raspberry PI Zero / Arduino Uno.

### 7.1 Raspberry PI Zero

| Raspberry Pi Zero (J8 Header) |                      |    |  |                      |       |
|-------------------------------|----------------------|----|--|----------------------|-------|
| GPIO#                         | NAME                 |    |  | NAME                 | GPIO# |
|                               | 3.3 VDC Power        | 1  |  | 5.0 VDC Power        | 14    |
| 8                             | GPIO 8 SDA1 (I2C)    | 3  |  | 5.0 VDC Power        | 15    |
| 9                             | GPIO 9 SCL1 (I2C)    | 5  |  | Ground               | 16    |
| 7                             | GPIO 7 GPCLK0        | 7  |  | GPIO 15 TXD (UART)   | 15    |
|                               | Ground               | 9  |  | GPIO 16 RXD (UART)   | 16    |
| 0                             | GPIO 0               | 11 |  | GPIO 1 PCM_CLK/PWM0  | 1     |
| 2                             | GPIO 2               | 13 |  | Ground               | 2     |
| 3                             | GPIO 3               | 15 |  | GPIO 4               | 4     |
|                               | 3.3 VDC Power        | 17 |  | GPIO 5               | 5     |
| 12                            | GPIO 12 MOSI (SPI)   | 19 |  | Ground               | 6     |
| 13                            | GPIO 13 MISO (SPI)   | 21 |  | GPIO 10 CE0 (SPI)    | 10    |
| 14                            | GPIO 14 SCLK (SPI)   | 23 |  | GPIO 11 CE1 (SPI)    | 11    |
|                               | Ground               | 25 |  | SCL0 (I2C ID EEPROM) | 31    |
| 30                            | SDA0 (I2C ID EEPROM) | 27 |  | Ground               | 26    |
| 21                            | GPIO 21 GPCLK1       | 29 |  | GPIO 25 PWM0         | 25    |
| 22                            | GPIO 22 GPCLK2       | 31 |  | Ground               | 27    |
| 23                            | GPIO 23 PWM1         | 33 |  | GPIO 27              | 27    |
| 24                            | GPIO 24 PCM_FSPWM1   | 35 |  | GPIO 28 PCM_DIN      | 28    |
| 25                            | GPIO 25              | 37 |  | GPIO 29 PCM_DOUT     | 29    |
|                               | Ground               | 39 |  |                      |       |

**Attention!** The GPIO pin numbering used in this diagram is intended for use with WiringPi / P432. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Figura 1: Raspberry PI zero Pinout

---

### Algoritmo 1 Flashing LED Python

---

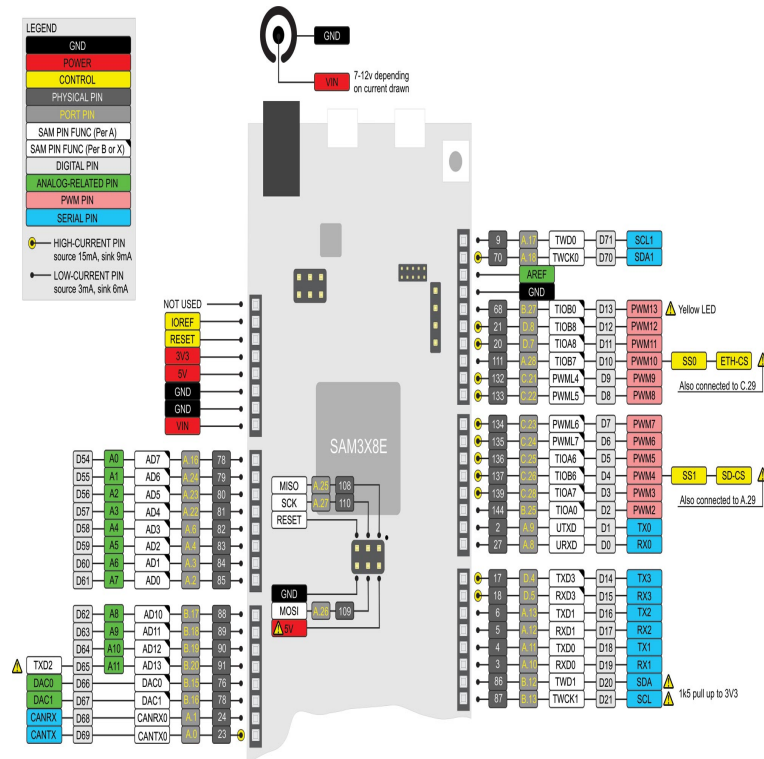
```

from gpiozero import LED
from time import sleep
led = LED(17)
enquanto 1 faça:
    função LED.ON
    fim função
    função SLEEP(1)
    fim função
    função LED.OFF
    fim função
    função SLEEP(1)
    fim função
fim enquanto

```

---

## 7.2 Arduino Uno



---

**Algoritmo 2** Flashing LED C

---

```
função VOID SETUP
    função PINTMODE(LED-BUILTIN, OUTPUT)
    fim função
fim função
função VOID LOOP
    função DIGITALWRITE(LED-BUILTIN, HIGH)
    fim função
    função DELAY(1000)
    fim função
    função DIGITALWRITE(LED-BUILTIN, LOW)
    fim função
    função DELAY(1000)
    fim função
fim função
```

---