

# Sistemas Embarcados (C213)

Prof. Samuel Baraldi Mafra



Créditos: Prof. Yvo Marcelo Chiaradia Masselli

## Objetivo geral

O objetivo desse curso é apresentar métodos e etapas de projeto e desenvolvimento aplicados aos sistemas embarcados. Arquiteturas de software em sistemas embarcados. Aplicações

## Programa

- Sistemas Embarcados;
- Modelo Conceitual de Arquitetura em Camadas de Hardware/Software;
- Hardware e Software para Sistemas Embarcados;
- Sistemas Dinâmicos;
- Sistemas de Controle;
- Projeto de Controladores PID Digitais;
- Lógica Difusa;
- Projeto de controlador baseado em Lógica Difusa embarcado

## Bibliografia

- DORF, Richard C. (Richard C. Dorf); BISHOP, Robert H. (Robert H. Bishop); SILVA FILHO, Bernardo Severo da (Bernardo Severo da Silva Filho), Sistemas de controle modernos. 12 ed. Rio de Janeiro, RJ: LTC, 2013, 814 p. ISBN 978-85-216-1995-6.
- MALVINO, Albert Paul (Albert Paul Malvino); LASCHUK, Anatolio (Anatolio Laschuk), Microcomputadores e microprocessadores. São Paulo, SP: McGraw-Hill, 1985, 578 p. ISBN 0-07-450303-0.
- SILVA JÚNIOR, Vidal Pereira (Vidal Pereira Silva Júnior), Microcontrolador 8051. São Paulo, SP: Érica, 1990, 143 p. ISBN 85-7194-036-3.

## Bibliografia

- CAMPOS, Mario Cesar M. Massa de (Mario Cesar M. Massa de Campos); TEIXEIRA, Herbert C. G., Controles típicos de equipamentos e processos industriais. 2 ed. São Paulo, SP: Editora Blucher, 2010, 396 p. ISBN 978-85-212-0552-4.
- GIMENEZ, Salvador P. (Salvador P. Gimenez), Microcontroladores 8051: Teoria de hardware e software, aplicações em controle digital, laboratório/simulação. São Paulo, SP: Prentice Hall, 2002, 253 p. ISBN 85-87918-28-1.
- NICOLOSI, Denys Emílio Campion (Denys Emílio Campion Nicolosi), Microcontrolador 8051 detalhado. 4 ed. São Paulo, SP: Érica, 2000, 221 p. ISBN 85-7194-721-x.
- OGATA, Katsuhiko (Katsuhiko Ogata); MAYA, Paulo Alvaro (Paulo Alvaro Maya); LEONARDI, Fabrizio (Fabrizio Leonardi), Engenharia de controle moderno. 5 ed. São Paulo, SP: Pearson Prentice Hall, 2010, 809 p. ISBN 978-85-7605-810-6.
- PEREIRA, Fábio (Fábio Pereira), Microcontroladores PIC: programação em C. 7 ed. São Paulo, SP: Érica, 2007, 358 p. ISBN 978-85-7194-935-5.

## Avaliação

- A avaliação será feita através de dois projetos.
- Serão desenvolvidos dois projetos práticos ao longo do semestre com notas de 0 a 100 pontos descritas a seguir:  
NPJ1: Implementação de um controlador PID Digital embarcado.  
NPJ2: Implementação de um controlador baseado em Lógica Difusa embarcado.  
$$NP1=NP2=NPL= (NPJ1*0,3 + NPJ2*0,7)$$

As aulas teóricas vão ocorrer nas terças feiras das 19:30 -21:10.

**Horário de atendimento:**

Terça-Feira 17:30 - 19:30

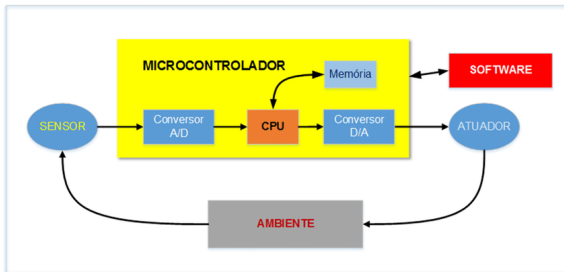
IoT Research Group Prédio II

Laboratórios:

**Vinicius Bottini Jardim**

- Sistema computacional: Processador + Memória + Periféricos
- Embarcado:
  - Faz parte de outro sistema;
  - Exemplos: aeronave, automóvel, eletrodoméstico, equipamento agrícola, equipamento médico, equipamento de telecomunicações, etc.
- Reage a eventos externos e internos





- Um sistema embarcado pode ser microprocessado ou microcontrolado , no qual o "computador" é completamente encapsulado ou dedicado ao dispositivo ou sistema que o mesmo controla.
- É desenvolvido para uma tarefa específica. Por questões como segurança e usabilidade, pode possuir restrições para computação em tempo real.
- O software escrito para sistemas embarcados é muitas vezes chamado **firmware** , e armazenado em uma memória ROM ou memória flash ao invés de um disco rígido.

- Por vezes o sistema também é executado com recursos computacionais limitados: sem teclado, sem tela e com pouca memória.
- Em geral os sistemas embarcados possuem uma capacidade de processamento reduzida em comparação com computadores pessoais (notebooks e desktops).

## Aplicações

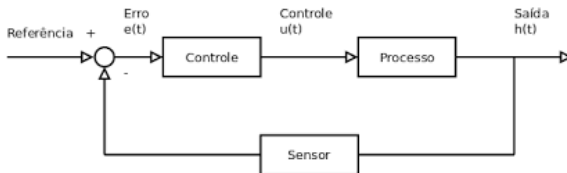
- Processamento de sinais : aplicações que envolvem um grande volume de informação a ser processado em curto espaço de tempo.
- Os sinais a serem tratados são digitalizados através de ADs , processados, e novamente convertidos em sinais analógicos por DAs .
- Exemplos: compressores de vídeo, radares e sonares, etc.

## Aplicações

- Comunicações e redes: chaveamento e distribuição de informações.
- Exemplo: Sistemas de telefonia e telecomunicações, equipamentos de redes e internet.



- Sistemas de controle: controles em malha fechada com realimentação em tempo real.
- Geralmente são as aplicações mais robustas, com placas dedicadas e múltiplos sensores de entrada e saída.
- Muitas vezes fornecem pouca interação com o usuário.
- Usados nos motores de automóveis, processos químicos, controle de vôo , usinas nucleares, etc.

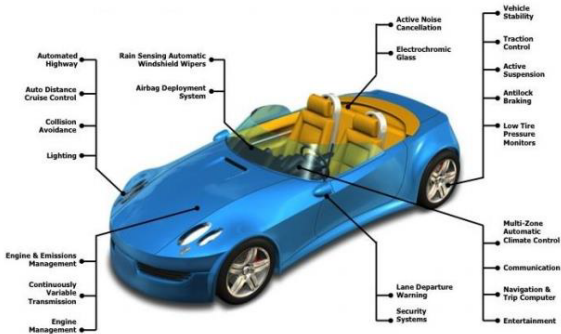


## Aplicações de propósito geral versus aplicações de propósito específico

- Aplicações de propósito geral: são as aplicações mais parecidas com os computadores de mesa, mas em embalagens embarcadas .
- Nelas costumam haver grande interação entre os usuários e o sistema, geralmente através de terminais de vídeo ou monitores. Como exemplo tem se os vídeo games, caixas de bancos, centrais multimídia automotivas e etc.

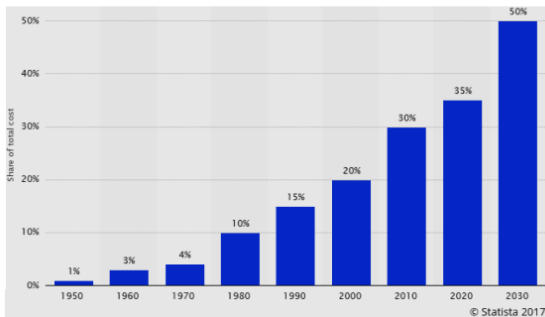
## Aplicações de propósito específico

### Sistemas Automotivos (Drive-by-Wire)



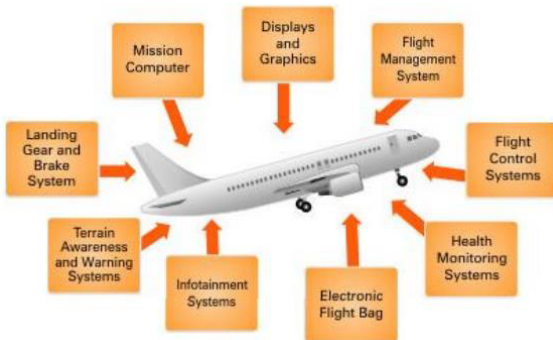


# Eletrônica Automotiva Embarcada

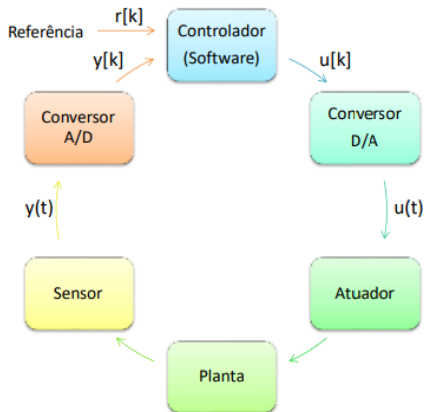


Aplicações de propósito específico

## Sistemas Aviônicos (Fly-by-Wire)

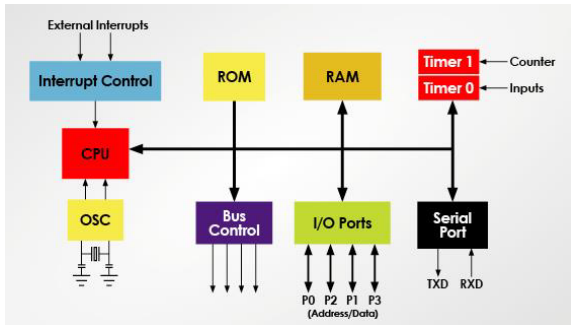


# Sistema de Controle Digital



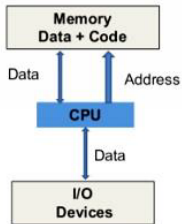
## Microcontrolador

- Um microcontrolador pode ser visto como um computador em um único chip.
- Esse chip contém um processador (Unidade Lógica e Aritmética ULA), memória, periféricos de entrada e de saída, temporizadores, dispositivos de comunicação serial, dentre outros.

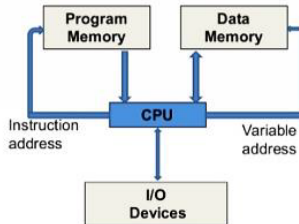


## Arquiteturas Harvard e Von Neuman

- Quando um sistema de processamento de dados (processadores e microcontroladores) possui uma única área de memória na qual ficam armazenados os dados (variáveis) e o programa a ser executado (software), dizemos que esse sistema segue a arquitetura de Von Neuman.
- No caso em que os dados (variáveis) ficam armazenados em uma área de memória e o programa a ser executado (software) fica armazenado em outra área de memória, dizemos que esse sistema segue a arquitetura Harvard.

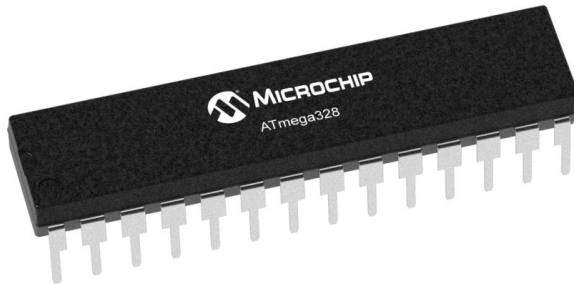


Von Neumann Machine

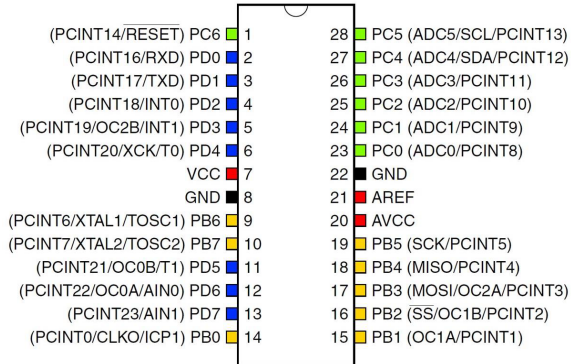


Harvard Machine

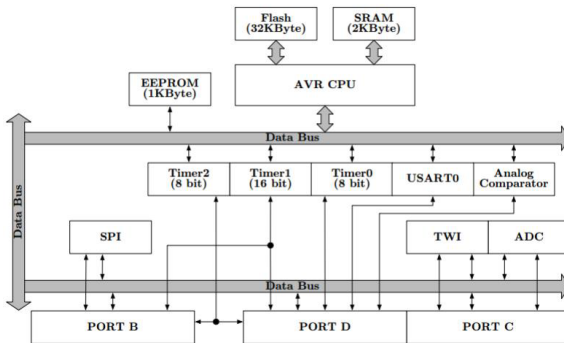
## Hardware do ATMEGA328







## Arquitetura de microcontrolador



Cada componente tem uma função específica:

- a) CPU (Unidade de Processamento Central) tem a finalidade de interpretar as instruções de programa.
- b) Na memória somente de leitura (ROM) na qual são gravadas as instruções do programa.
- c) A memória RAM (Memória de Acesso Aleatório) é utilizada para memorizar as variáveis utilizadas pelo programa.
- d) O conjunto de LINHAS de I/O é utilizado para controlar dispositivos externos ou receber impulsos de sensores, interruptores, etc.
- e) O conjunto de dispositivos auxiliares dão suporte ao funcionamento do componente, ou seja, gerador de clock, contadores, UASART para comunicação, etc.

- Cada periférico é controlado e configurado através de um conjunto de registradores específicos.
- Cada registrador é mapeado em uma posição da memória de dados, ou seja, sua manipulação se dá através de operações de escrita e leitura em memória.
- Cada linha das portas B, C e D (GPIOs) pode ser configurados como entrada ou saída de maneira independente.
- A memória flash de programa pode ser programada através da conexão serial SPI ou por meio de um boot loader (Arduino).



## GPIO

- As portas de entrada e saída de propósito geral (GeneralPurpose Input/Output GPIO) são provavelmente os componentes dos microcontroladores mais empregados em sistemas embarcados.
- GPIOs são linhas digitais conectadas aos pinos externos do microcontrolador usadas no controle e acionamento de dispositivos (LEDs, botões, motores, etc).
- Cada GPIO pode ser configurada independentemente como um pino de entrada ou de saída.
- Via software, pode se escrever um nível 0 ou um nível 1 em uma GPIO de saída ou pode se ler valores digitais em uma GPIO de entrada.

- Temporizador : circuito capaz de medir intervalos de tempo.
  - Pode gerar eventos temporais por exemplo, para manter o sinal verde em um semáforo por 10 s.
  - Pode medir o tempo decorrido entre a ocorrência de eventos por exemplo, para computar a velocidade de um carro.
- Como medir o tempo?
  - Contando os pulsos que aparecem em um sinal de relógio de entrada que possui um período/frequência conhecido.
  - Exemplo: Se a frequência do relógio é igual a 1 MHz (período  $1\ \mu\text{s}$ ) e são contabilizados 2000 pulsos, o tempo decorrido é igual a 2 ms.

### Exemplo:

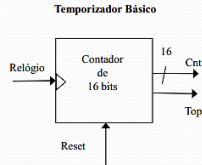
➤ **Faixa:** máximo intervalo de tempo que o temporizador consegue medir.

Exemplo:  $65535 \cdot 1 \times 10^{-6} = 65,535 \text{ ms}$

➤ **Resolução:** mínimo intervalo de tempo que o temporizador consegue medir (período do relógio).

Exemplo:  $1 \mu\text{s}$

➤ O sinal top indica que a contagem máxima foi alcançada

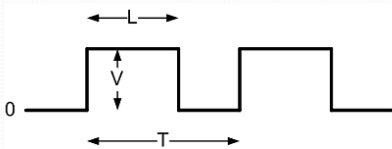




## Modulação por largura de pulso (PWM)

- Modular : inserir informações em um sinal mediante a modificação de suas características. Exemplos: AM e FM.
- Modulação por largura de pulso ou Pulse Width Modulation (PWM) ajusta o tempo que um sinal periódico com dois níveis (digital), alto e baixo, permanece em nível alto de acordo com a informação a ser inserida.
- Tempo de nível alto : largura do pulso
- Duty cycle : razão entre a largura do pulso (tempo em nível alto) e período da onda.

- $duty\ cycle = \frac{L}{T}$



- Tensão média:

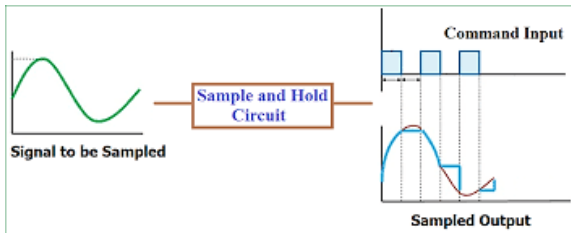
$$V_m = \frac{1}{T} \int_{t-T/2}^{t+T/2} V(t) dt = \frac{1}{T} LV = V \times duty\ cycle$$

## Aplicações:

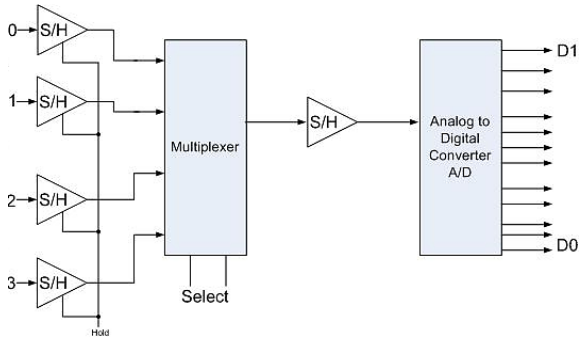
- Controle de velocidade de motores CC, como, por exemplo, ventoinhas de computadores.
- Controle de luminosidade de lâmpadas em dimmers.
- Posicionamento de servomotores em robótica e em dispositivos radio controlados.

## Conversor A/D

- O conversor analógico digital (A/D) efetua a conversão de um sinal analógico para a sua representação digital de 10 bits. A primeira etapa no processo de conversão é a amostragem e retenção.

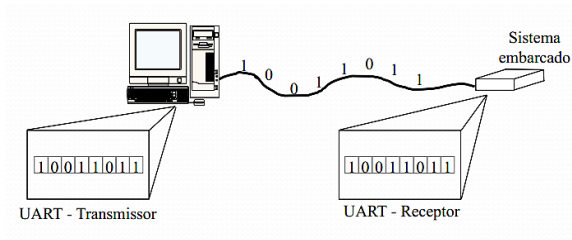


A saída do circuito de amostragem e retenção é ligado à entrada do conversor A/D de  $n$  bits. Onde  $n$  é a resolução do conversor.

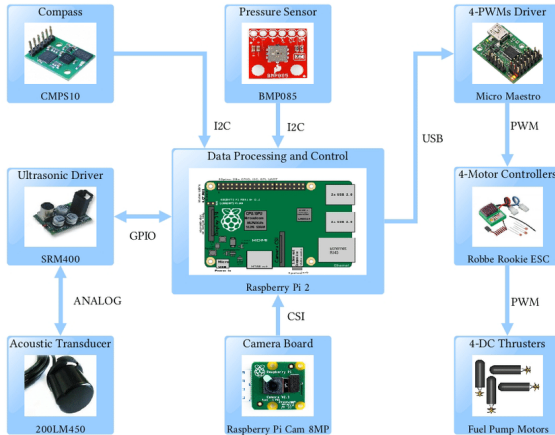


## UART

- UART (Universal Asynchronous Receiver/Transmitter): receptor/transmissor universal assíncrono.
- Recebe dados seriais e os armazena como dados paralelos (usualmente um byte); também recebe dados paralelos e os transmite de forma serial.
- Útil para comunicação entre dispositivos bastante afastados ou quando há poucos pinos de entrada/saída disponíveis.
- Baud rate = determina a velocidade com que dados são trocados entre duas UARTs; mede o número de mudanças de sinal por segundo que ocorrem na linha serial.

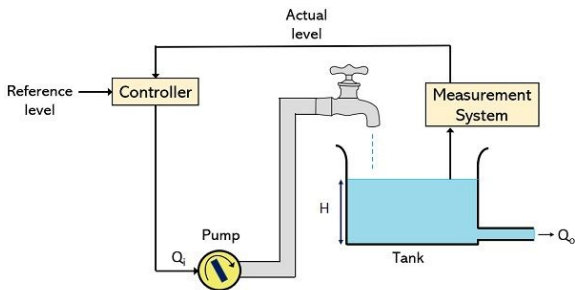


# Raspberry-Pi





## Controle de nível



Liquid Level Control System

Electronics Coach

## Aplicação IoT

