

Prof. Me. Renato Alves Ferreira  
email: [renato.ferreira@fmu.br](mailto:renato.ferreira@fmu.br)

Disciplina:

*MICROCONTROLADORES e IoT*

# Semana 1

## Apresentação

- **Professor**
- **Disciplina – Microcontroladores e IoT**
- **Método de Avaliação**

Composição da média final =  $(N1 \cdot 0,4) + (N2 \cdot 0,6)$  - Aprovação média  $\geq 6$

- **Controle da frequência 75%** (+ - 5 dias de ausência)

## Orientações gerais

## Introdução a Microcontroladores

### AVALIAÇÃO

A Média Final (MF) da disciplina considera os seguintes elementos e valores:

N1	N2
A1 – Avaliação(ões) a ser(em) definida(s) de acordo com os objetivos de aprendizagem  [nota de 0 a 10]	A2 – Avaliação(ões) a ser(em) definida(s) de acordo com os objetivos de aprendizagem (9,0 pontos) + APS – Atividade Prática Supervisionada (1,0 ponto)  [nota de 0 a 10]  <b>OU</b>  SUB – Avaliação Substitutiva  [nota de 0 a 10]

A Média Final (MF) é calculada por meio da média ponderada das duas notas, N1 e N2, com peso, respectivamente de 40% e 60%, resultante da seguinte equação:

$$MF = (N1 \cdot 0,4) + (N2 \cdot 0,6)$$

Para aprovação, a Média Final deverá ser igual ou superior a 6,0 (seis), além da necessária frequência mínima de 75% nas aulas.

O estudante que não realizar a A2 ou não atingir a média final 6,0 (seis) na disciplina, poderá realizar uma Avaliação Substitutiva (SUB), cuja nota substituirá a nota de A2 obtida, caso seja maior.

# O que esperar da disciplina

## Ementa

Aborda a evolução da arquitetura e organização de microcontroladores. Discute questões relacionadas ao conjunto básico de instruções, programação, modos de endereçamento, pilhas, organização de memórias e interfaces de comunicação.

## Objetivos

Analisar e projetar hardware e software de sistemas microcontrolados e suas interfaces. Explicar o princípio de funcionamento dos microprocessadores/microcontroladores 8051, 8085, ATMEGA 328, ARM. Desenvolver programas em Assembly e/ou Linguagem C, C++, utilizando ambientes de simulação. Utilizar ferramentas de análise, desenvolvimento e depuração de programas para o microprocessador/microcontrolador. Entender, Projetar e Configurar cenários com elementos da IoT.

# Referências

## **BIBLIOGRAFIA BÁSICA**

Gimenez, S. P. Microcontroladores 8051: conceitos, operações, fluxogramas e programação. [recurso eletrônico, Minha Biblioteca]. 1ª ed. Erica, 2015.

Gimenez, S. P. Microcontroladores 8051: teoria e prática. [recurso eletrônico, Minha Biblioteca]. 1ª ed. Erica, 2010.

Monk, S. Internet das Coisas: Uma Introdução com o Python - Série Tekne. Link: <https://integrada.minhabiblioteca.com.br/books/9788582604793>

## **BIBLIOGRAFIA COMPLEMENTAR**

Nicolosi, D. E. C. Microcontrolador 8051 detalhado. [recurso eletrônico, Minha Biblioteca]. 9ª ed. Erica, 2013

Oliveira, C. L. V.; Zanetti, H. A. P. Arduino descomplicado: como elaborar projetos de eletrônica. [recurso eletrônico, Minha Biblioteca]. 1ª ed. Erica, 2015.

Monk, S. Programação com Arduino : começando com sketches. [recurso eletrônico, Minha Biblioteca]. 2ª ed. Bookman, 2015.

Monk, S. Programação com Arduino II: passos avançados com sketches. [recurso eletrônico, Minha Biblioteca]. 2ª ed. Bookman, 2017.

Monk, S. 30 Projetos com Arduino. [recurso eletrônico, Minha Biblioteca]. 2ª ed. Bookman, 2014.

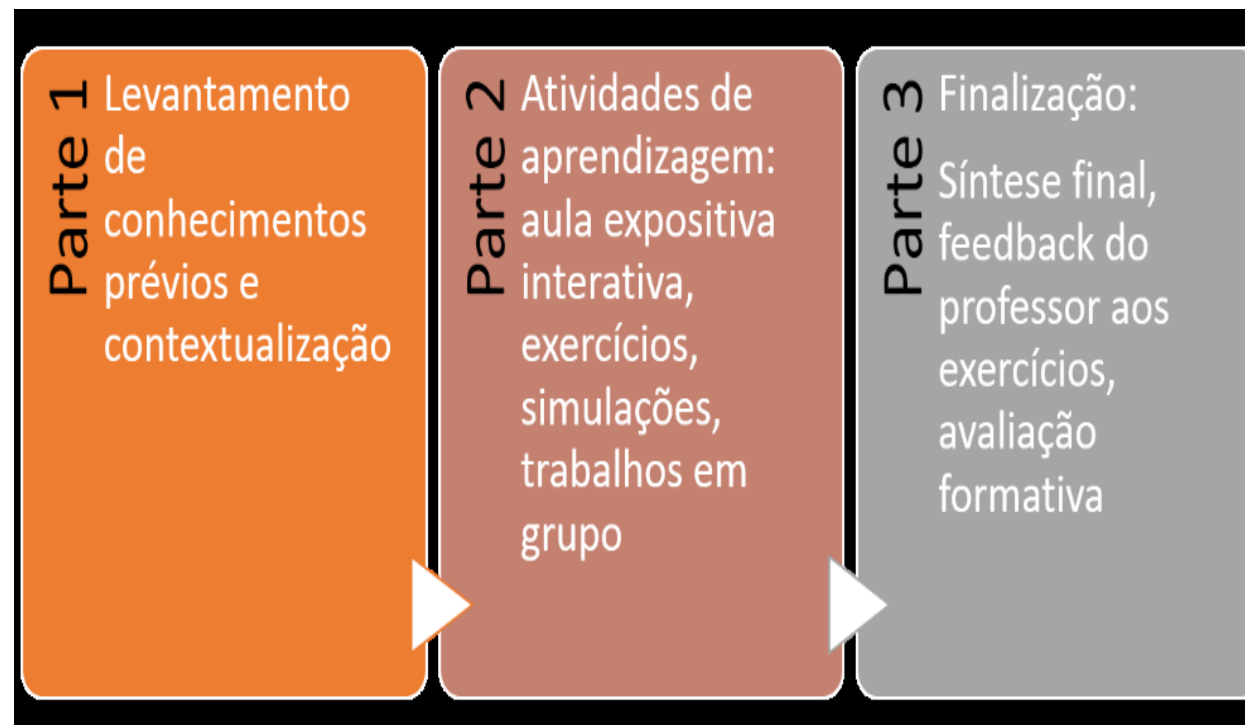
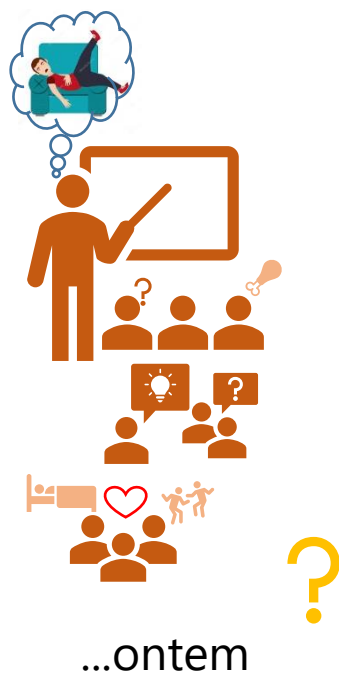
Stallings, W. Arquitetura e Organização de Computadores [recurso eletrônico, Minha Biblioteca]. 8ª ed. Pearson, 2010.

## ORIENTAÇÕES ACADÊMICAS (portfólio do estudante)

“Para tornar-se um profissional competente naquilo que faz, o mercado de trabalho exige que você mantenha uma atitude de buscar aprender sempre, de modo cada vez mais ativo e autônomo.”

Pensando nisso, suas aulas utilizam **metodologias ativas**, que buscam levá-lo(a) a envolver-se nas atividades e fomentar uma aprendizagem realmente significativa.

As aulas são estruturadas em 3 partes:



# COMO APRENDEMOS



A pirâmide de aprendizagem  
de Willian Glasser



# Indicações e referências

- ❖ <https://scratch.mit.edu/>;
- ❖ <https://code.org/>;
- ❖ <https://www.arduino.cc/>
- ❖ <http://blog.novaeletronica.com.br/piinagem-ci-atmega328-arduino/>
- ❖ <https://www.arduino.cc/reference/pt/>
- ❖ <https://www.tinkercad.com/>
- ❖ [https://www.cisco.com/c/pt\\_br/index.html](https://www.cisco.com/c/pt_br/index.html)
- ❖ <http://www.logiccircuit.org/>
- ❖ <http://embarcados.com.br>
- ❖ Overdose de Youtube com conteúdos pertinentes;
- ❖ Pesquisas diversas;



# Introdução: Conceitos iniciais

- Rever conceitos de Sistemas computacionais
- Analisar as linguagens de programação associadas aos microcontroladores
- Rever a Estrutura básica de um computador
- Diferentes tipos de arquitetura de computadores  
(Processadores, Microcontroladores, PIC 16F877A, Arduino, Raspberry, z80)

# Linguagens de programação para microcontroladores

- Assembly
- C/C++
- Linguagem padrão do Arduino baseada em C++
- Python

# Simuladores de circuitos e testes de protótipos

- Tinkercad (<https://www.tinkercad.com/things>)
- EDSIM51 (<https://www.edsim51.com/>)
- Packet Tracer ([https://www.cisco.com/c/pt\\_br/index.html](https://www.cisco.com/c/pt_br/index.html))
- Circuito Lógico (<http://www.logiccircuit.org/>)
- Proteus (<https://www.labcenter.com/>)

# Revisão sobre os Sistemas Computacionais

- ❖ Hardware
- ❖ Software (Sistemas operacionais e sistemas dedicados)
- ❖ Infraestrutura



# **Introdução a Microcontroladores e IoT**

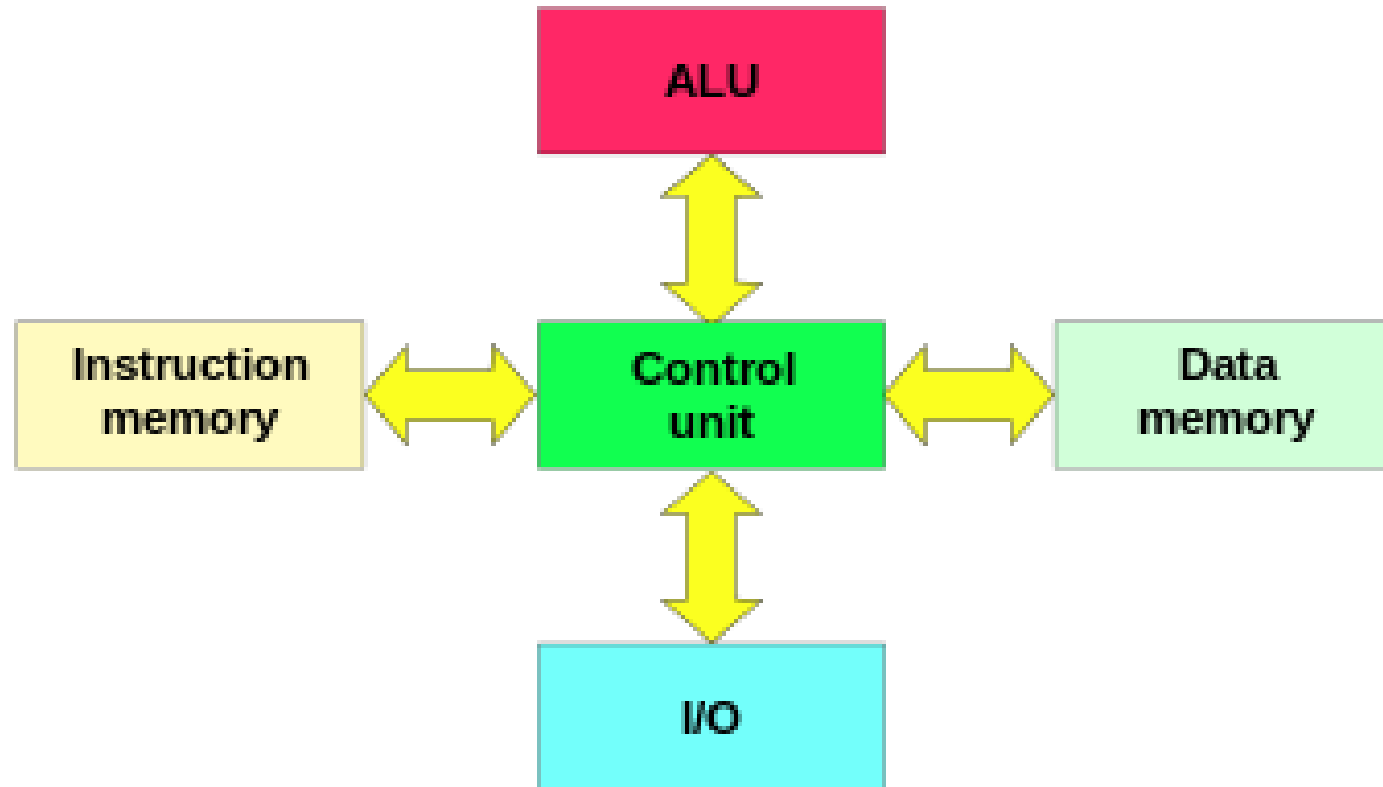
# Estrutura básica do Computador

## Arquitetura Von Neumann



# Estrutura básica do Computador

## Arquitetura de Harvard



# Von Neumann x Harvard

**RISC** (Reduced Instruction Set Computer) e **CISC** (Complex Instruction Set Computer)

## Von Neumann:

- Arquitetura mais simples;
- Mais lento pois não permite acesso simultâneo às memórias;
- Geralmente CISC

Exemplo:

- 4004 – 46 instruções
- 8080 – 78 instruções
- 8051 – 111 instruções
- 8085 – 150 instruções
- Z80 – Mais de 500 instruções

## Harvard:

- Arquitetura mais complexa;
- Mais rápido, pois permite acesso simultâneo às memórias;
- Geralmente RISC
- Permite o Pipelining

Exemplo:

- Intel 8086, 8088
- Microchip PIC – 35 instruções

	CISC			RISC	
	IBM370	VAX 11/780	8086	SPARC I	MIPS I
Ano	1973	1973	1978	1981	1983
Nº Instr.	208	303	100	39	55
Microc.	54Kb	400Kb	11Kb	0	0
Instr. (bytes)	2-6	2-57	1-17	4	4



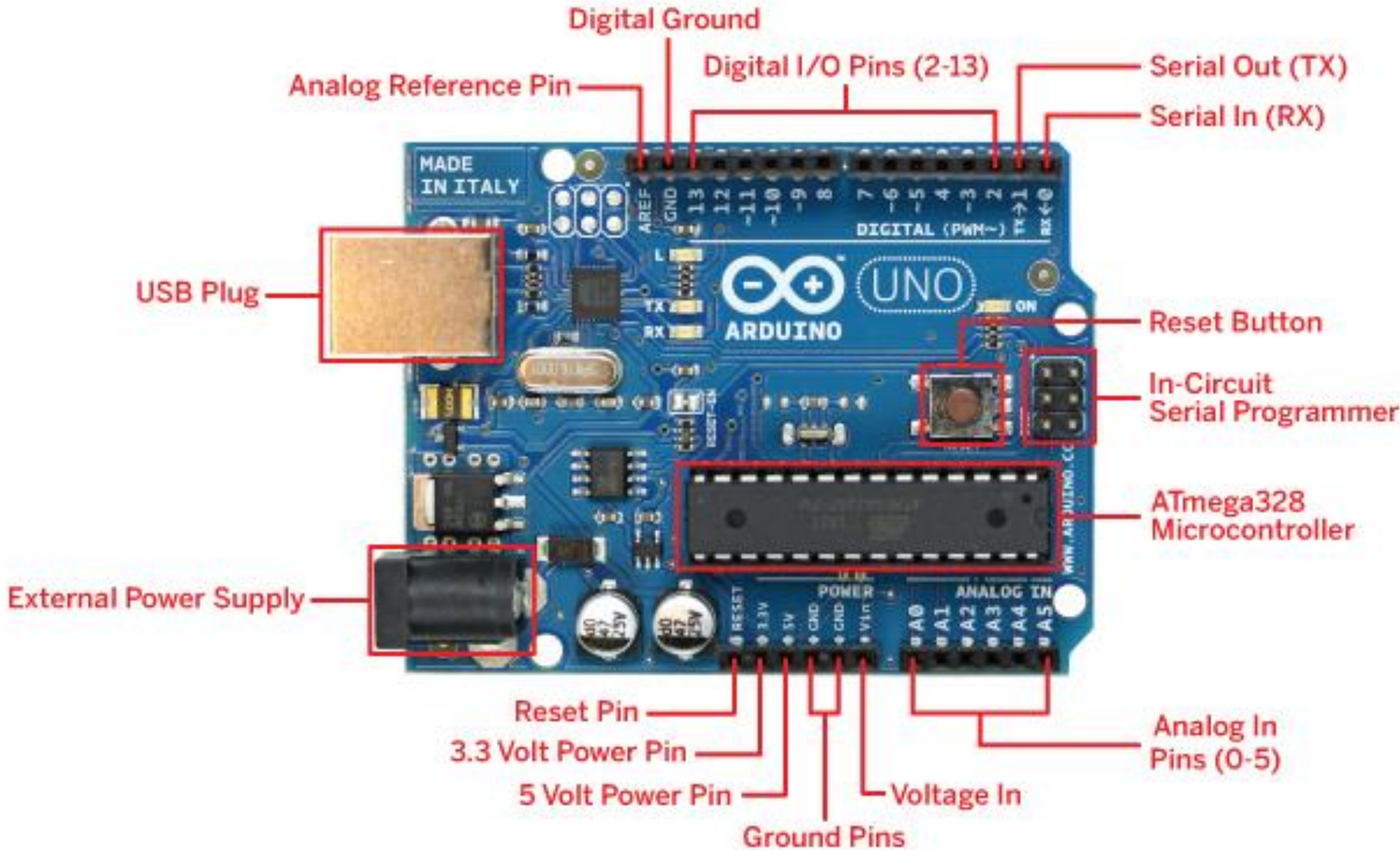
## Diferentes arquitetura de processadores e controladores

- Microprocessadores (8080, Z80, 80286, 80486, Pentium)
- Microcontroladores (ARM, ATMEL AVR, 8051, PIC16F877A, 16F628A)

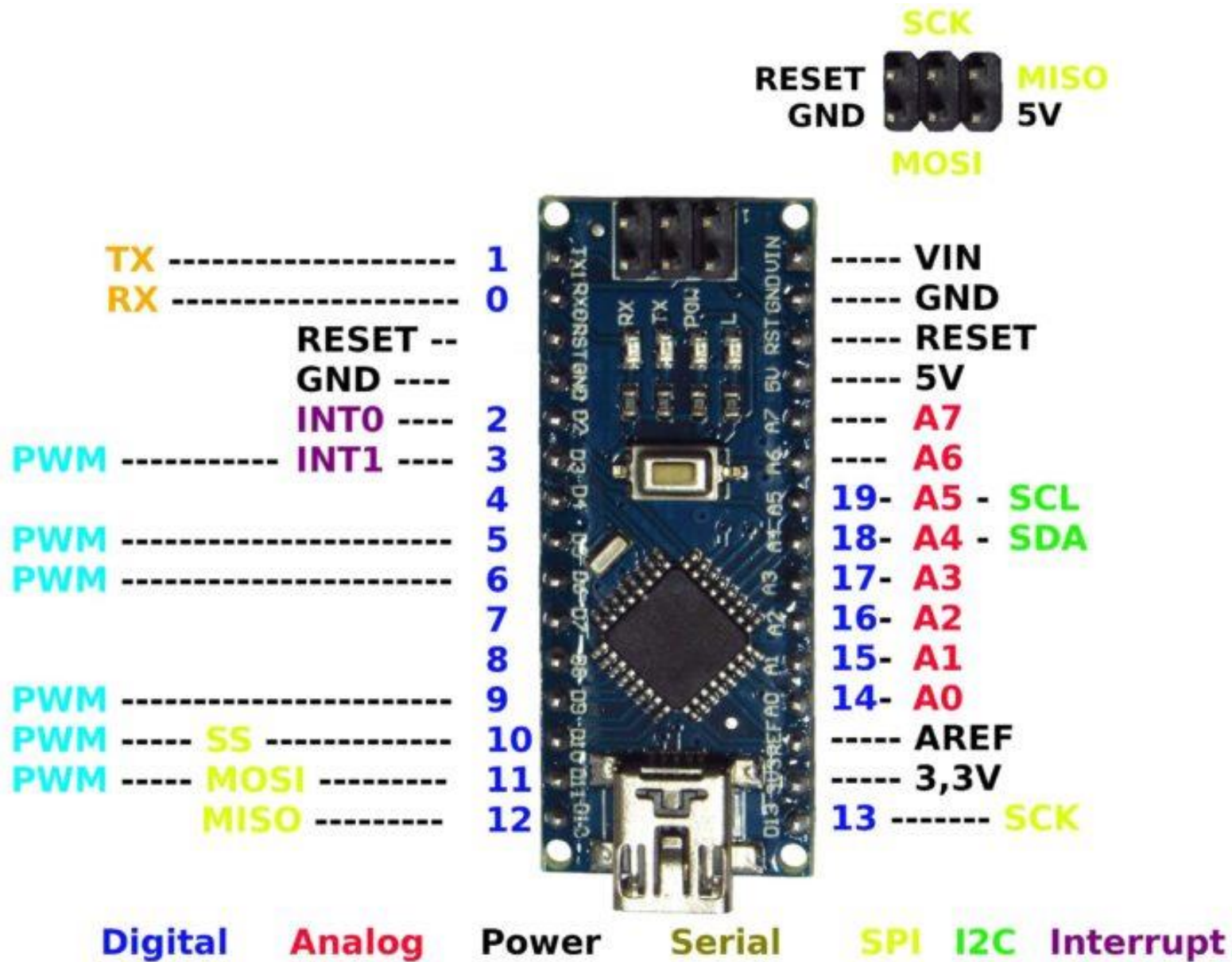
## Plataformas de prototipagem (embarcados)

- Arduino
- Raspberry Pi
- Arquitetura proprietária (Microchip, Multipic, etc)

# Arduino Uno Atmega328

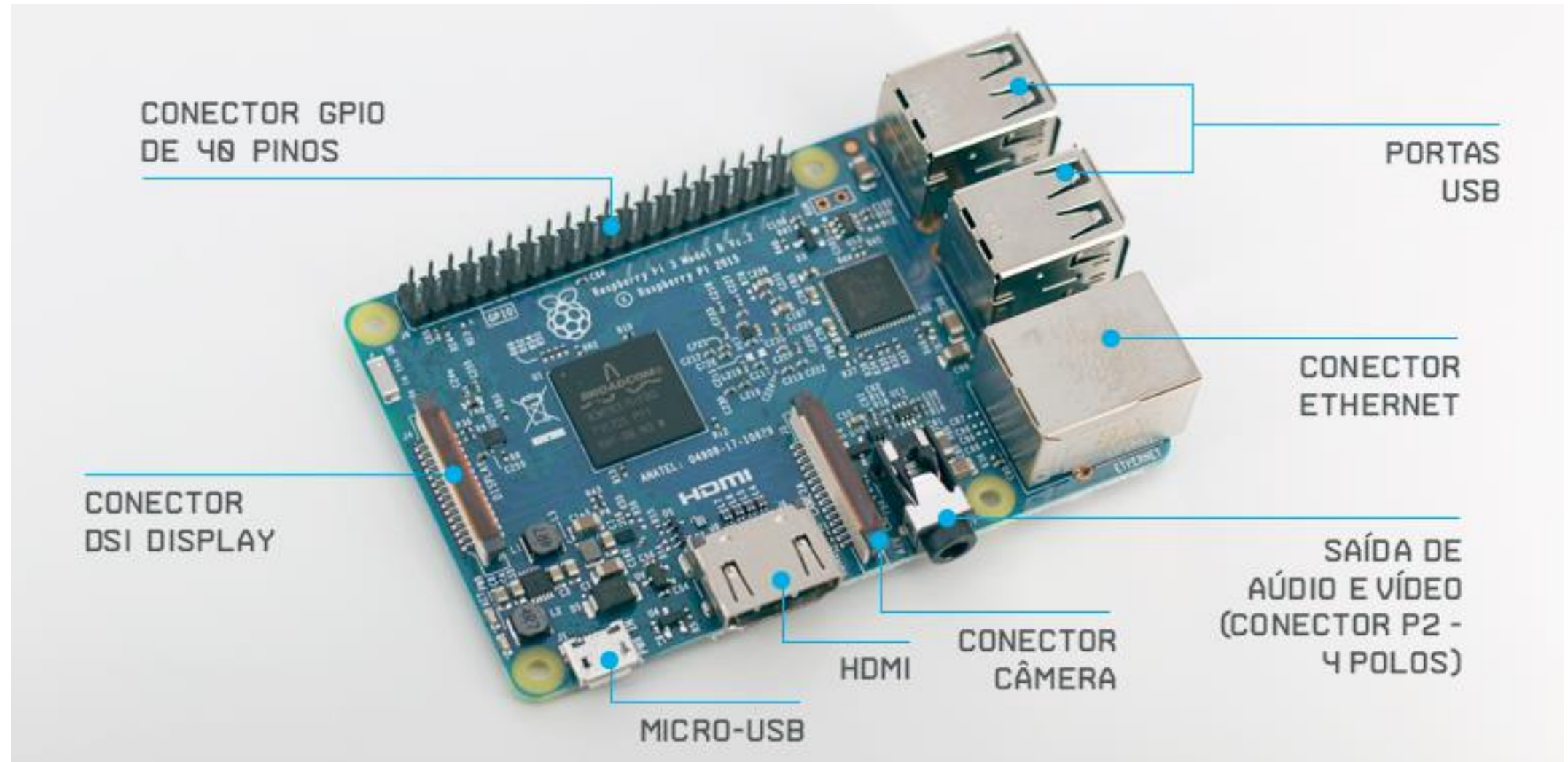


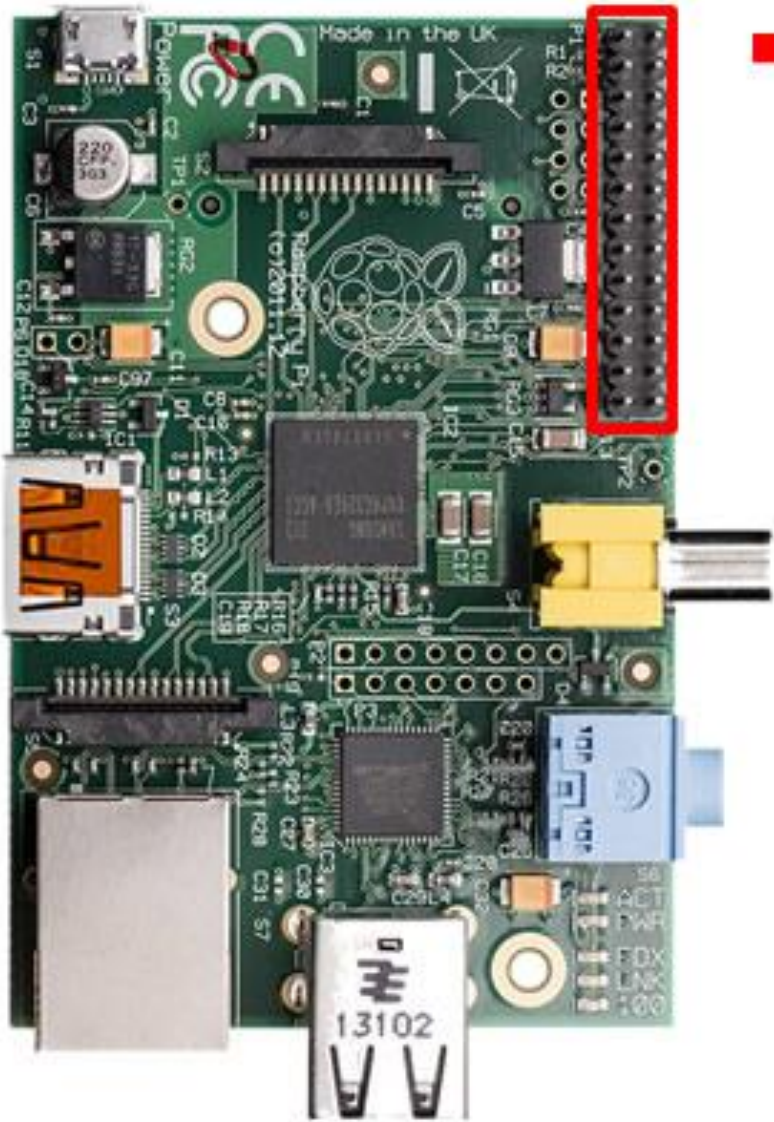
# Arduino Pro Mini Atmega328





# Raspberry Pi





Pinagem Raspberry Pi



- I2C, RTC
- Input1
- INT MAX3140
- LED2/Out2
- Input5
- SPI MAX3140

left		right	
bottom	top	bottom	top
P1-01	P1-02	P1-25	P1-26
3V3 Power	5V Power		
GPIO 0 (SDA)	--		
GPIO 1 (SCL)	Ground		
GPIO 4 (GPCLK0)	GPIO 14 (TXD)		
--	GPIO 15 (RXD)		
GPIO 17	GPIO 18 (PCM_CLK)		
GPIO 21 (PCM_DOUT)	--		
GPIO 22	GPIO 23		
--	GPIO 24		
GPIO 10 (MOSI)	--		
GPIO 9 (MISO)	GPIO 25		
GPIO 11 (SCKL)	GPIO 8 (CE0)		
	GPIO 7 (CE1)		

- RS232 TX
- RS232 RX
- LED1/Out1
- Input4
- Input3
- Input2
- CS MAX3140



# Encapsulamentos

Apresentação



DIP



PGA



SOIC



TSOP



SOJ



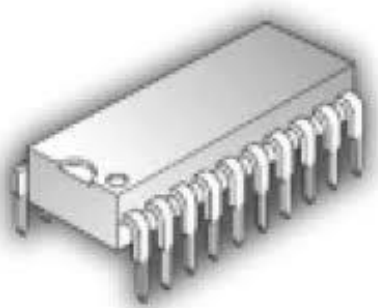
PLCC



QFP



BGA



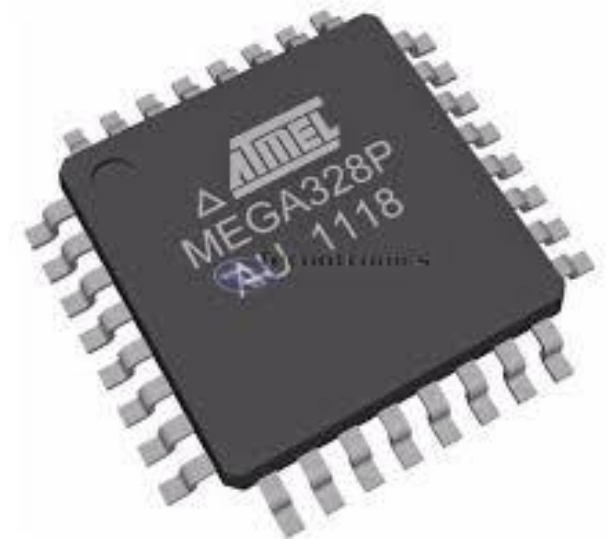
DIP



PDIP Package



TQFP Package



# Famílias de Microcontroladores

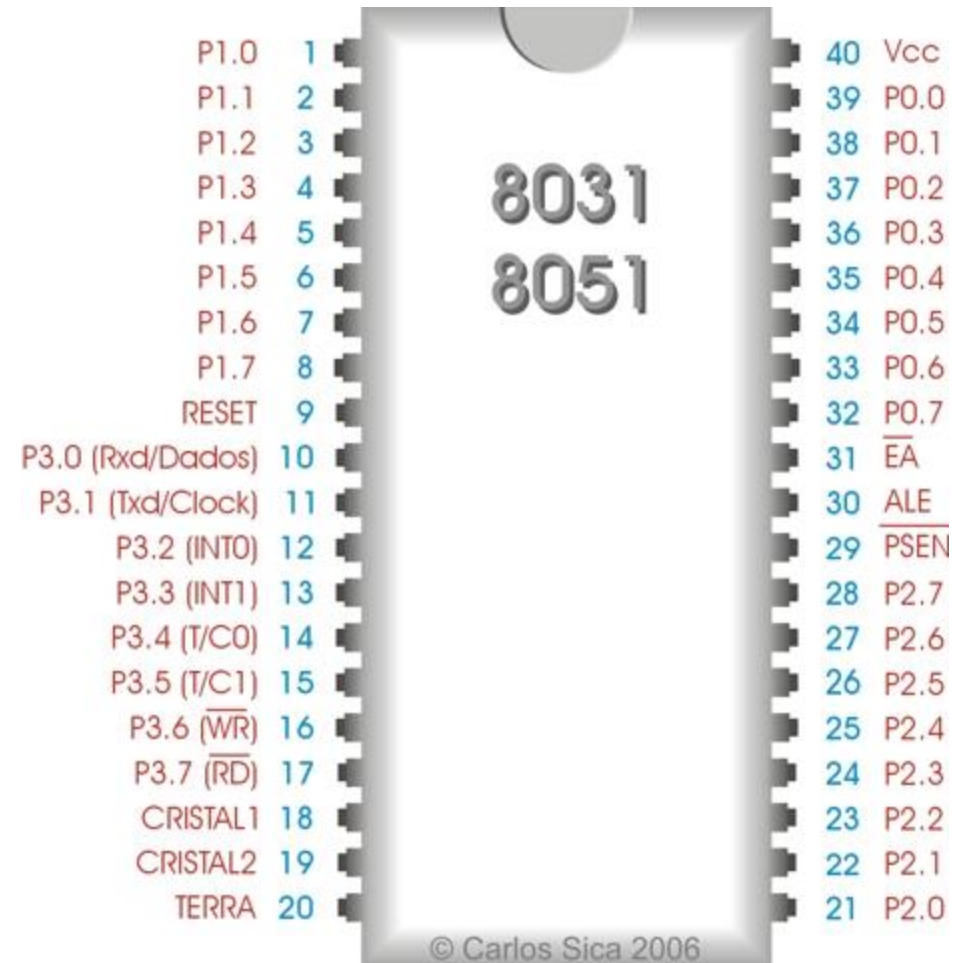
- **PIC** - fabricados pela [Microchip](#)
- **Atmel AVR** - fabricados pela [Atmel](#)
- **Intel MCS** - fabricados pela [Intel](#)



# **Exemplos Microcontroladores**

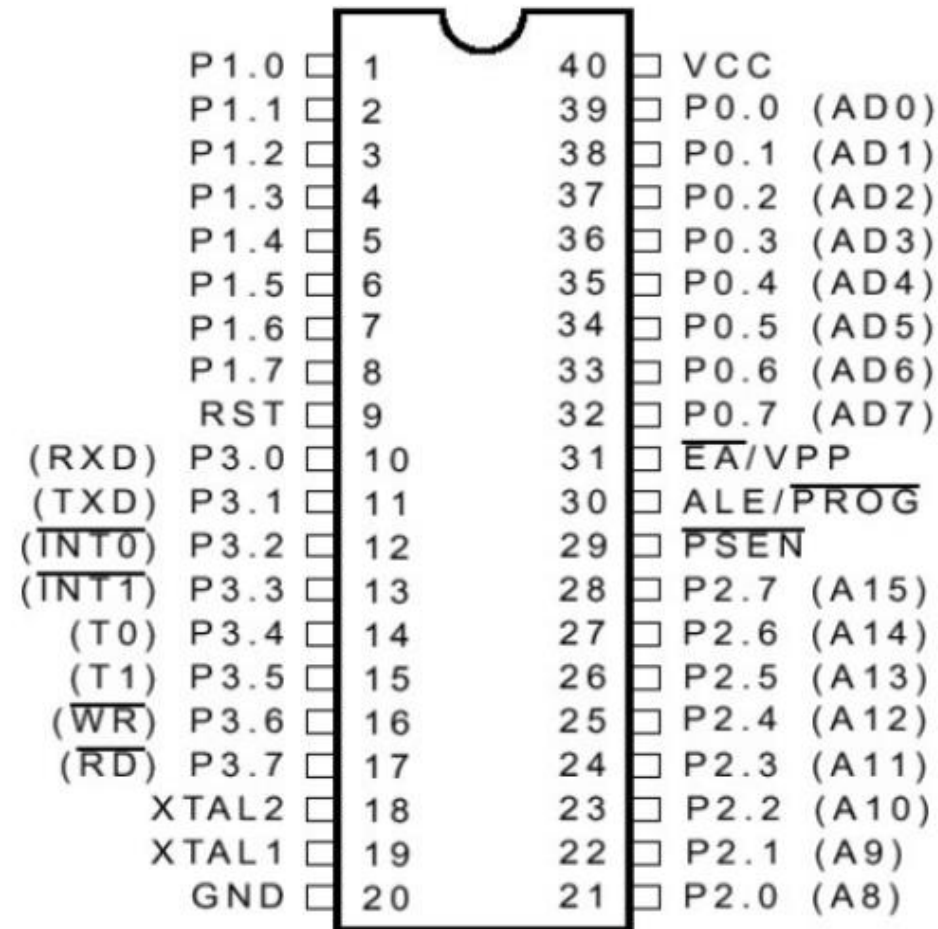


# Microcontrolador 8031 e 8051

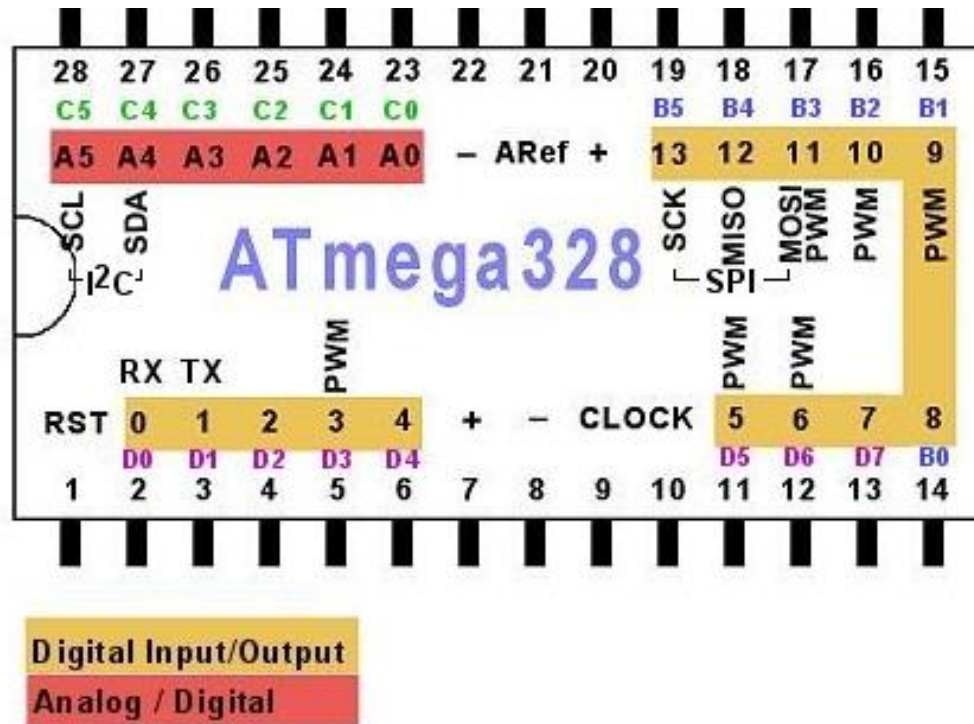


# Microcontrolador 8051

Pinagem para um encapsulamento de 40 pinos:



# Pinagem atmega328 Arduino



Porta B tem pinos B0 para B5

Porta C tem pinos C0 a C5

Port D tem pinos D0 a D7





# Proposta para APS – Atividade Prática Supervisionada

## Desenvolvimento de projeto prático em grupo

- *Projetar circuitos microcontrolados com Arduino ou para IoT*
- *Programar com Arduino ou para Internet das coisas*
- *Testar e demonstrar os códigos desenvolvidos*

**Te espero na próxima aula!**