

Microcontrolador e Microprocessador

MCMP

Prof^o José W. R. Pereira

jose.pereira@ifsp.edu.br

josewrpereira.github.io/docs



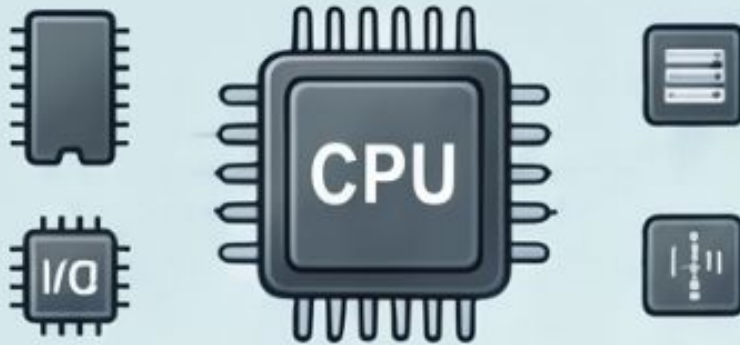
Índice

Arduino: Introdução à Arquitetura e à Plataforma

1. Visão Geral: Microprocessadores vs. Microcontroladores
2. O "Cérebro" do Uno: O ATmega328P
3. Anatomia do Arduino Uno (Pinagem)
4. Estrutura de um Programa (Sketch)
5. Software de Desenvolvimento (IDE)

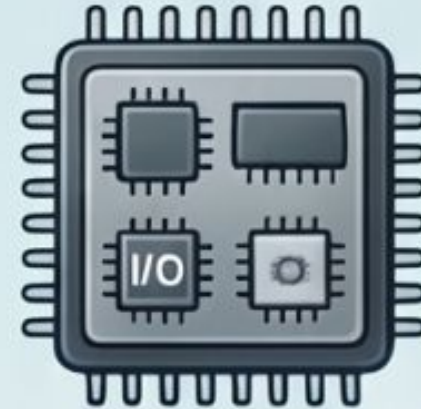
1. Visão Geral: Microprocessadores vs. Microcontroladores

Visão Geral: Microprocessadores vs. Microcontroladores



Microprocessador (MPU)

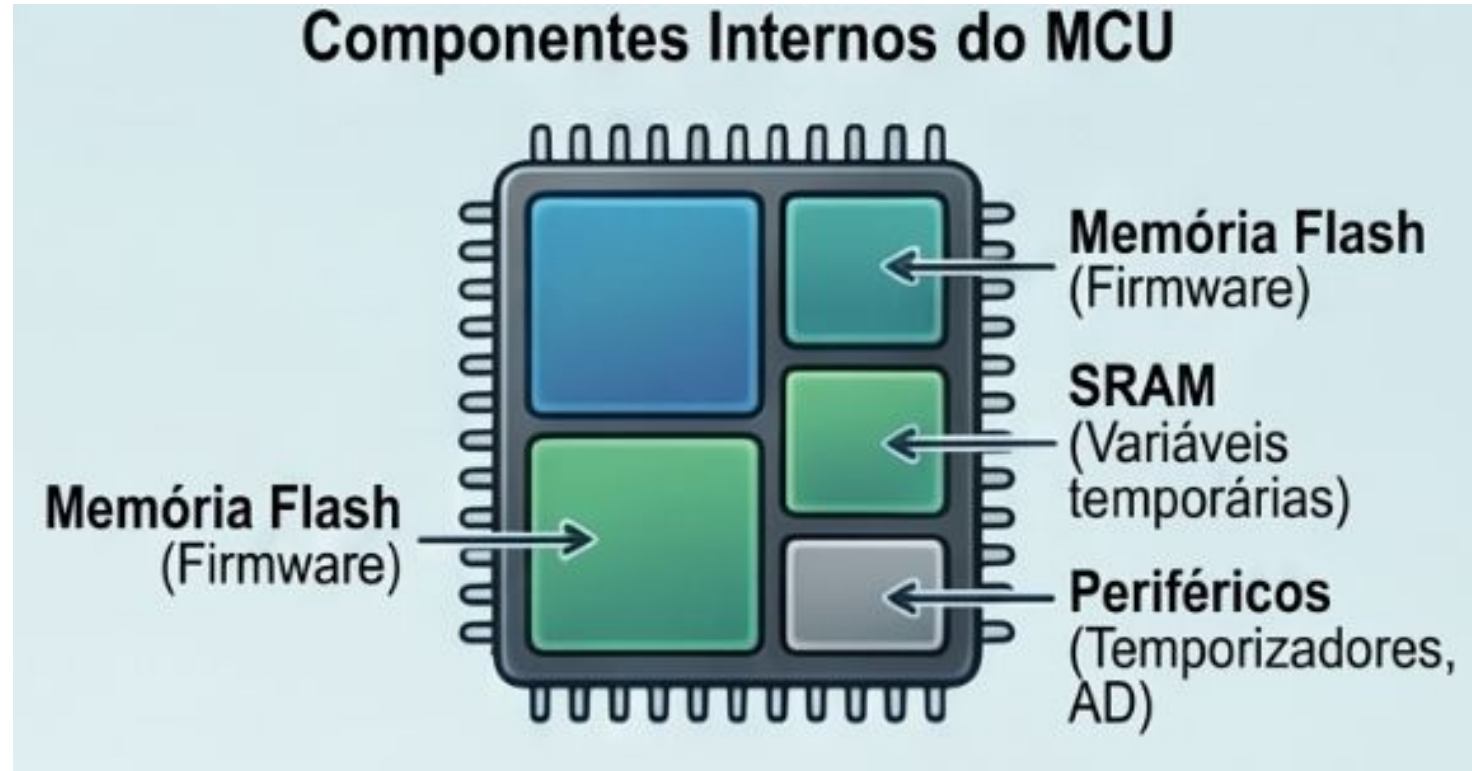
MPU é apenas a CPU e requer componentes externos (memória, I/O)



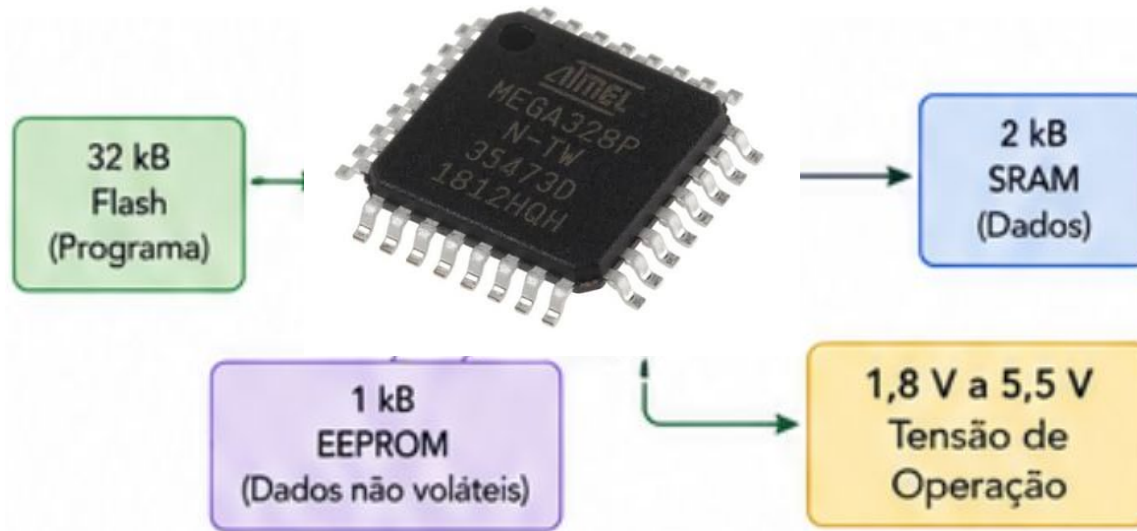
Microcontrolador (MCU)

MCU é um sistema computacional completo integrado em um único chip

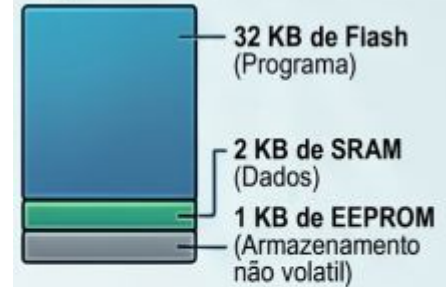
Visão Geral: Microprocessadores vs. Microcontroladores



Visão Geral: Microprocessadores vs. Microcontroladores



Capacidade de Memória



Tensão e Clock

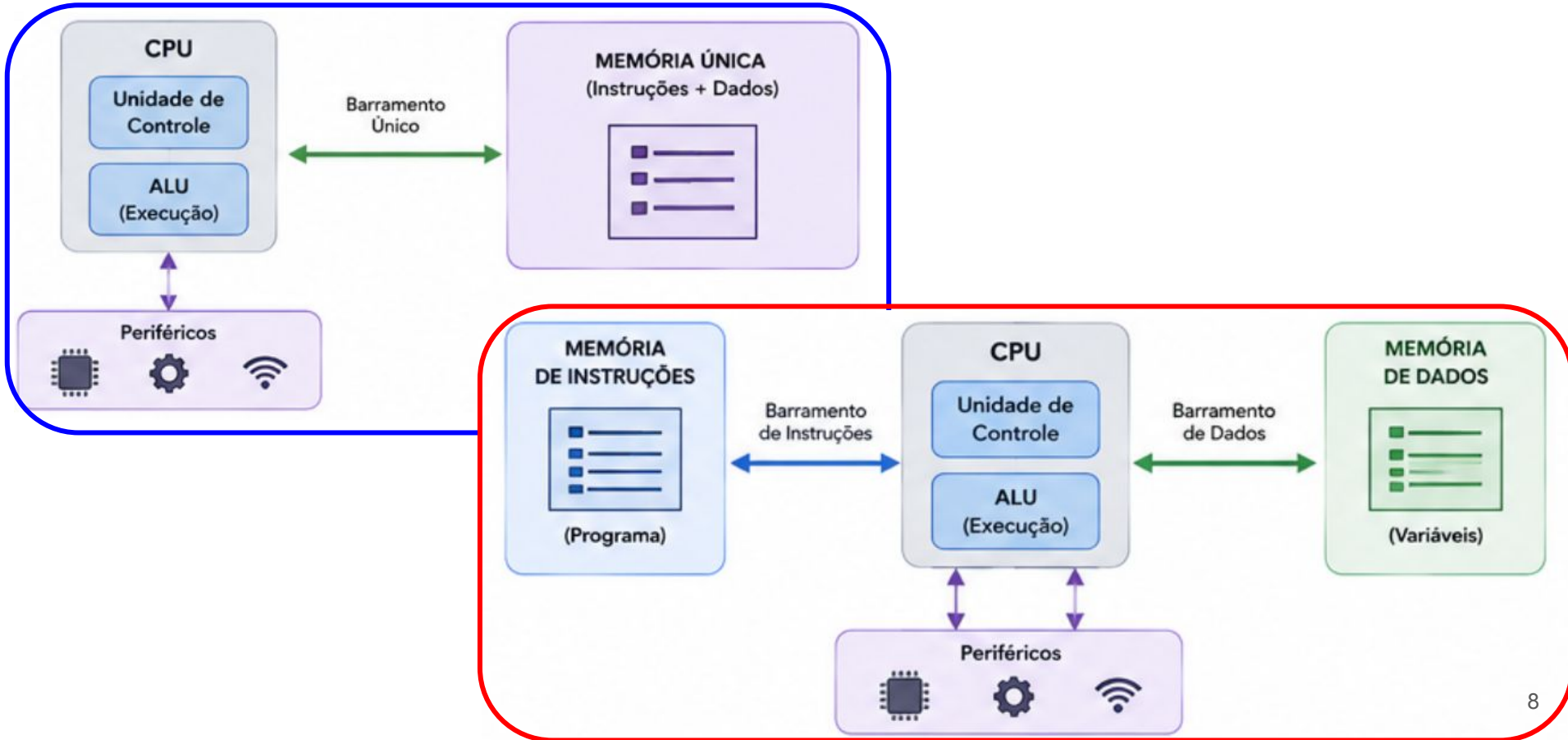
16  Opera entre 1,8 V e 5,5 V (padrão 5 V no Uno) com um ritmo de trabalho ditado por um cristal de 16 MHz

- ✓ **Arquitetura Harvard:** barramentos separados para instruções e dados, maior velocidade.
- ✓ **Arquitetura RISC Avançada:** 131 instruções, maioria executada em 1 ciclo de clock.

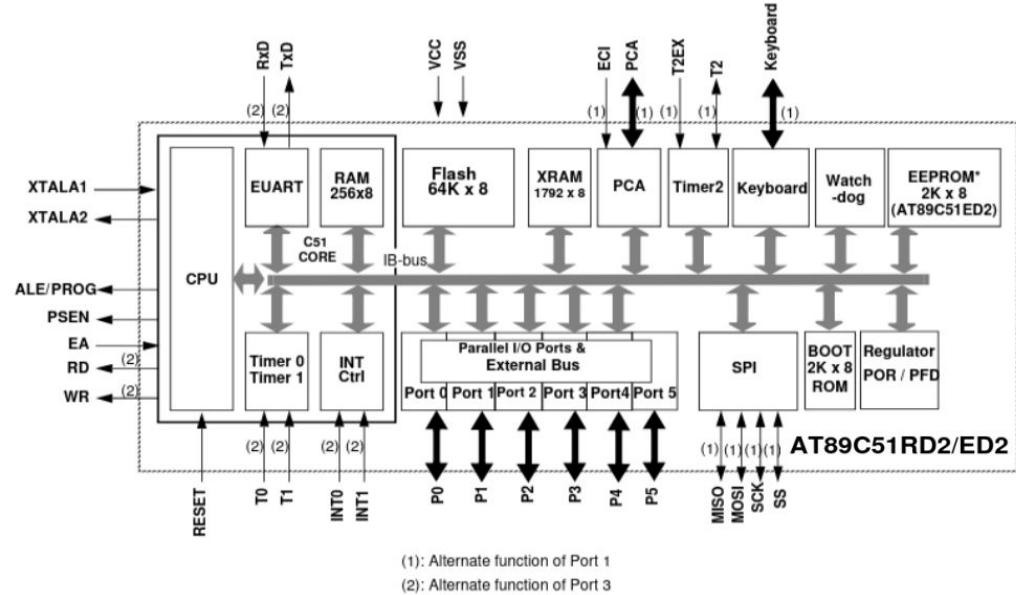
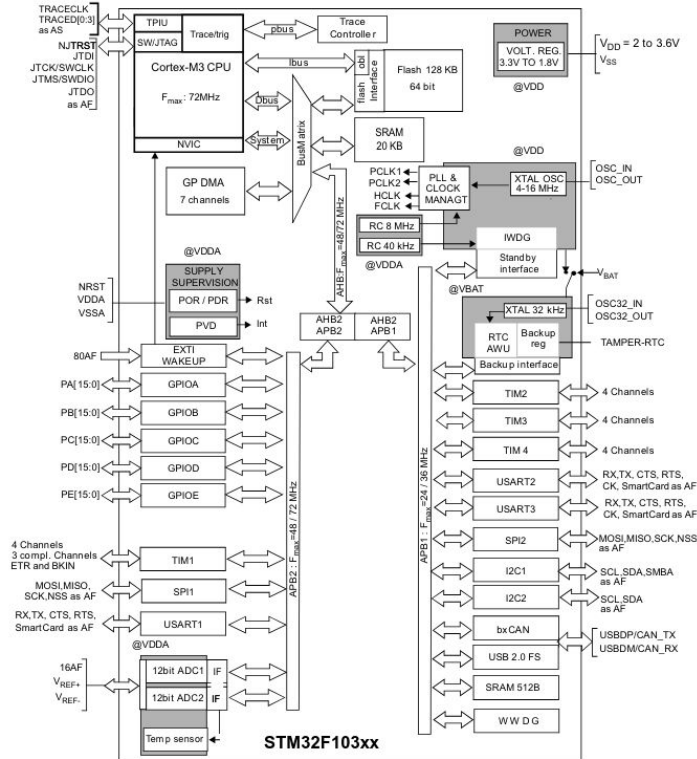
- ✓ **Capacidade de Memória:** 32 kB Flash, 2 kB SRAM, 1 kB EEPROM
- ✓ **Tensão de Operação:** 1,8 V a 5,5 V (na placa Arduino: 5 V via USB ou 7 V a 12 V via conector jack)

2. O "Cérebro" do Uno: O ATmega328P

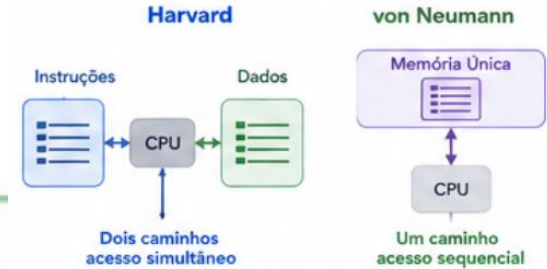
Arquiteturas: Von Neumann x Harvard



Arquiteturas: Von Newmann x Harvard



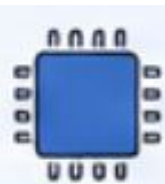
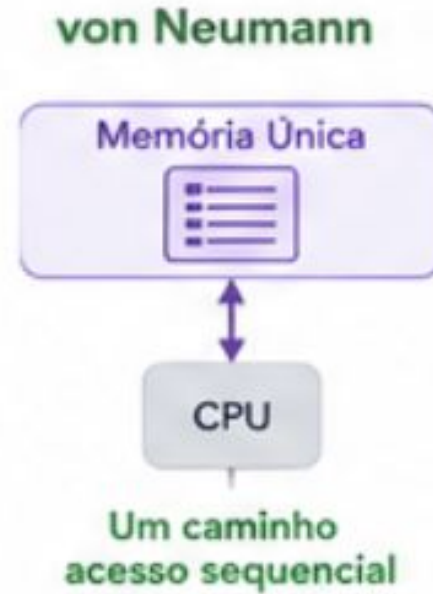
Arquiteturas: Von Neumann x Harvard



Comparação Rápida

Característica	Harvard	von Neumann
Memória	Duas memórias separadas	Uma memória única
Barramento	Dois barramentos (instr. e dados)	Um barramento compartilhado
Desempenho	Alto (busca e acesso simultâneos)	Menor (gargalo de acesso)
Complexidade	Maior	Menor
Custo	Maior	Menor
Uso Típico	MCUs, DSPs, sistemas embarcados, IoT, controle em tempo real	PCs, sistemas gerais, aplicações de propósito geral

Arquiteturas: Von Neumann x Harvard



Harvard

Sistemas embarcados, microcontroladores (Arduino, PIC, ESP32), DSPs, aplicações em tempo real e IoT.

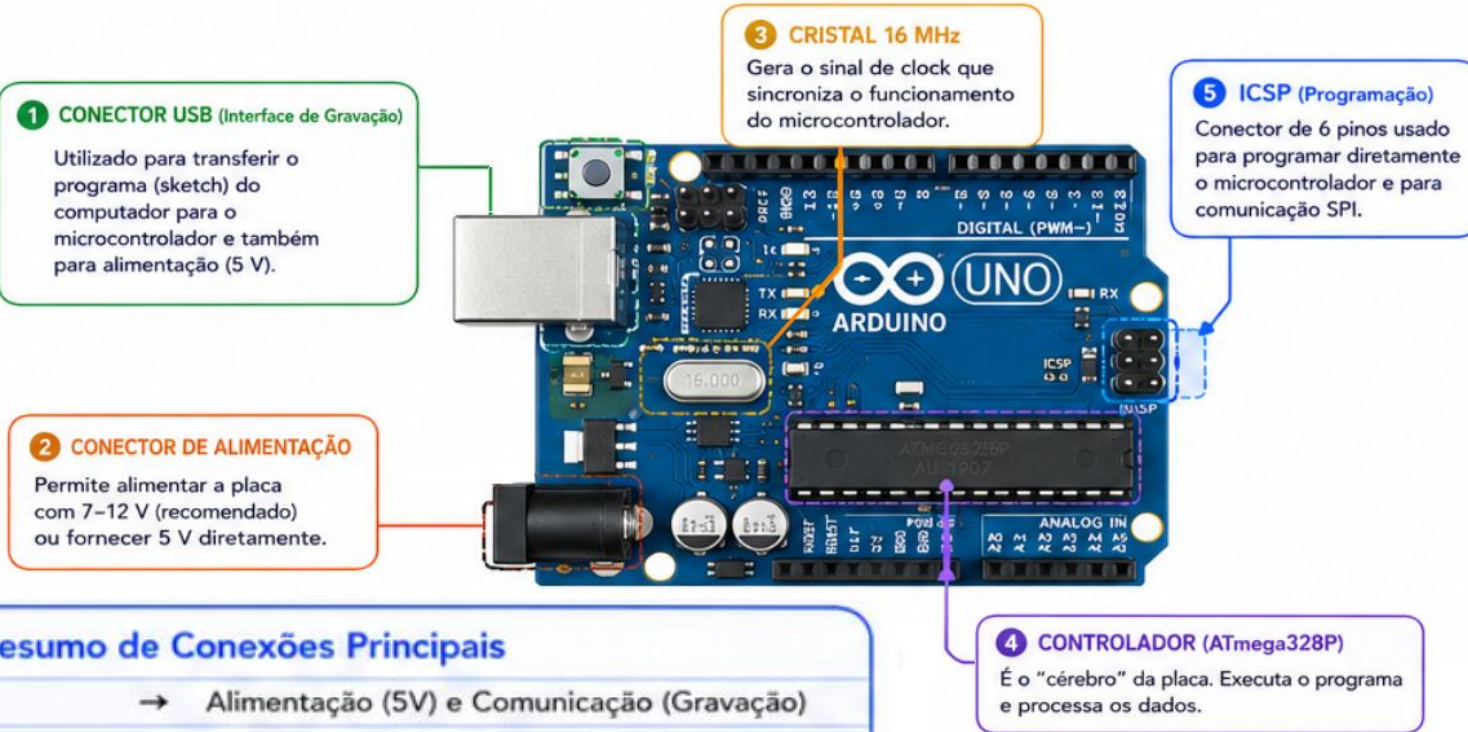


von Neumann

Computadores pessoais, servidores, sistemas operacionais e aplicações de uso geral.

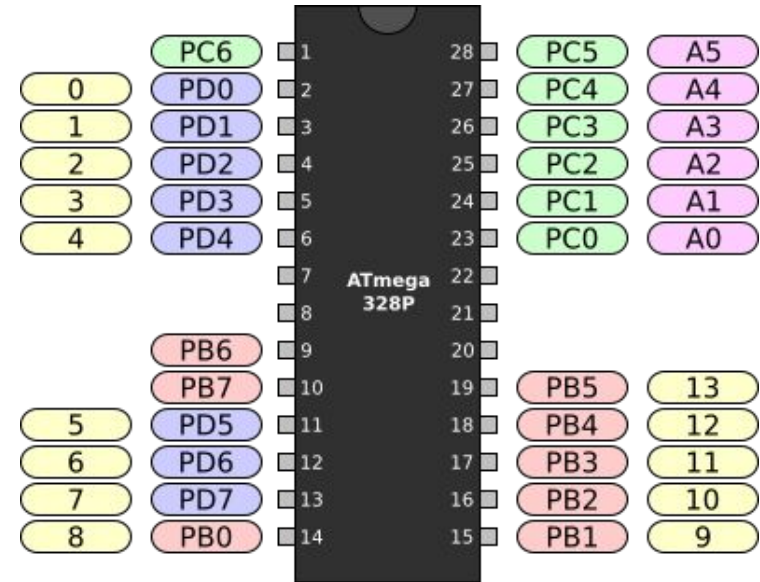
3. Anatomia do Arduino Uno (Pinagem)

Arduino Uno



Resumo de Conexões Principais

USB	→ Alimentação (5V) e Comunicação (Gravação)
Jack DC	→ Alimentação (7–12V)
Pinos Digitais 0–13	→ PORTD (0–7) e PORTB (8–13)
Pinos Analógicos A0–A5	→ PORTC → Conversor AD (10 bits)
ICSP	→ Programação e comunicação SPI
Cristal 16 MHz	→ Clock do sistema



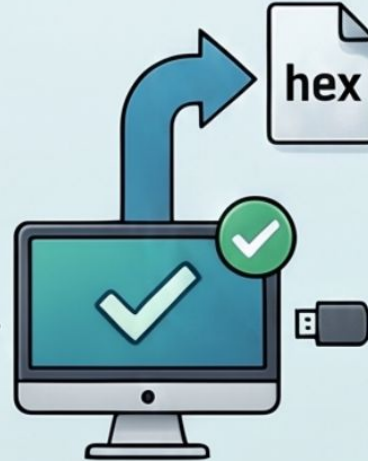
- 14

4. Estrutura de um Programa (Sketch)

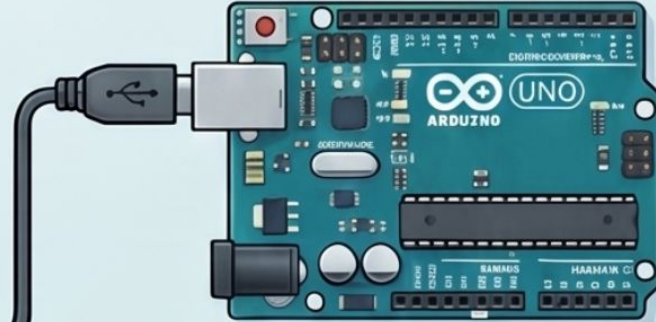
Estrutura do Programa

Estrutura do Software (O Sketch) e Fluxo de Desenvolvimento (IDE)

```
G/C++ +
1 void setup() {
2   Bloco executado uma única vez
3   para configurações iniciais
4   (GPIO entrada/saída)
5 }
6
7 void loop() {
8   Laço infinito que executa a
9   lógica principal do sistema de
10  forma ininterrupta logo após o
11  término do setup
12 }
```



Escrita e Compilação
O código é escrito em C/C++ e a compilação (Verify) o traduz para linguagem de máquina (.hex), verificando erros de sintaxe



Gravação (Upload)
Transferência do arquivo binário compilado para a memória Flash do ATmega328P via cabo USB

O Ecossistema Arduino
"Arduino" é a união de Hardware, Software e Comunidade, mas o foco profissional deve ser o domínio do Microcontrolador por trás da placa

Fluxo de Desenvolvimento de Firmware

1. ESCRITA

Escrevemos o código na IDE em linguagem baseada em C/C++.



2. COMPILAÇÃO (VERIFY)

O código é traduzido para linguagem de máquina (.hex) e verificado para erros de sintaxe.



3. GRAVAÇÃO (UPLOAD)

O arquivo compilado é enviado à memória Flash do ATmega328P via cabo USB.



// Código .ino

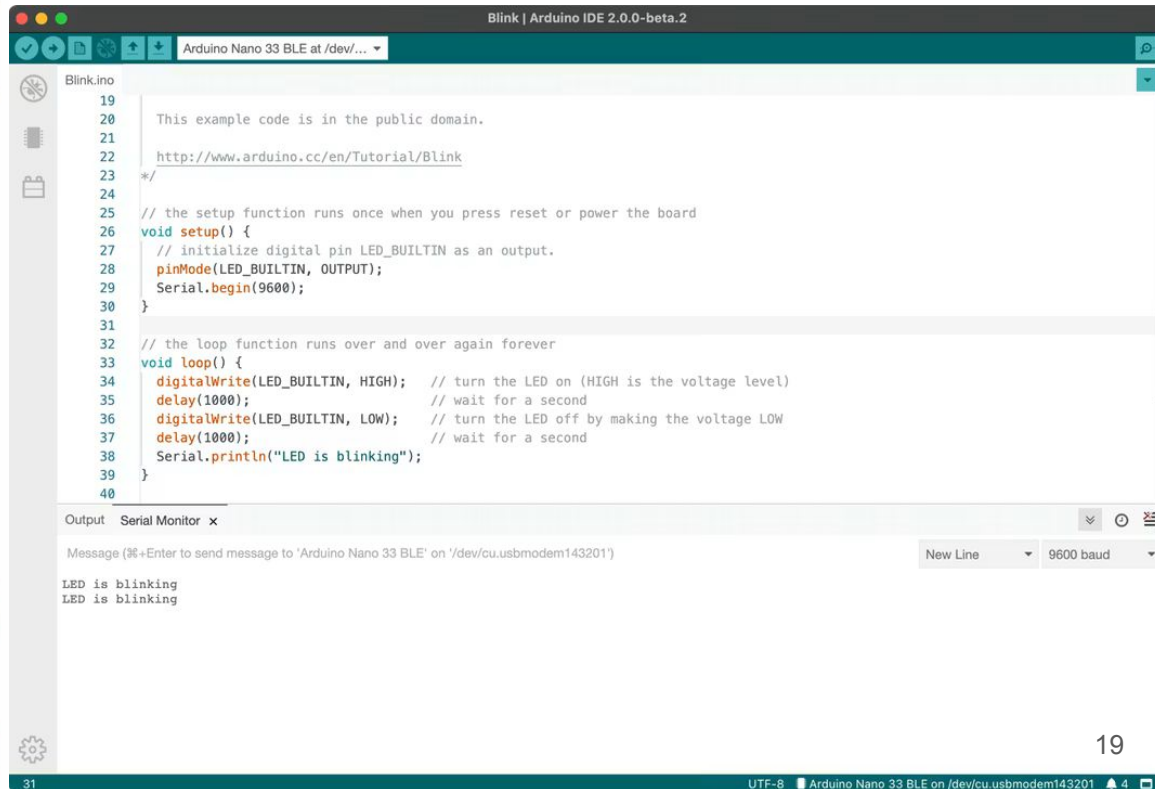
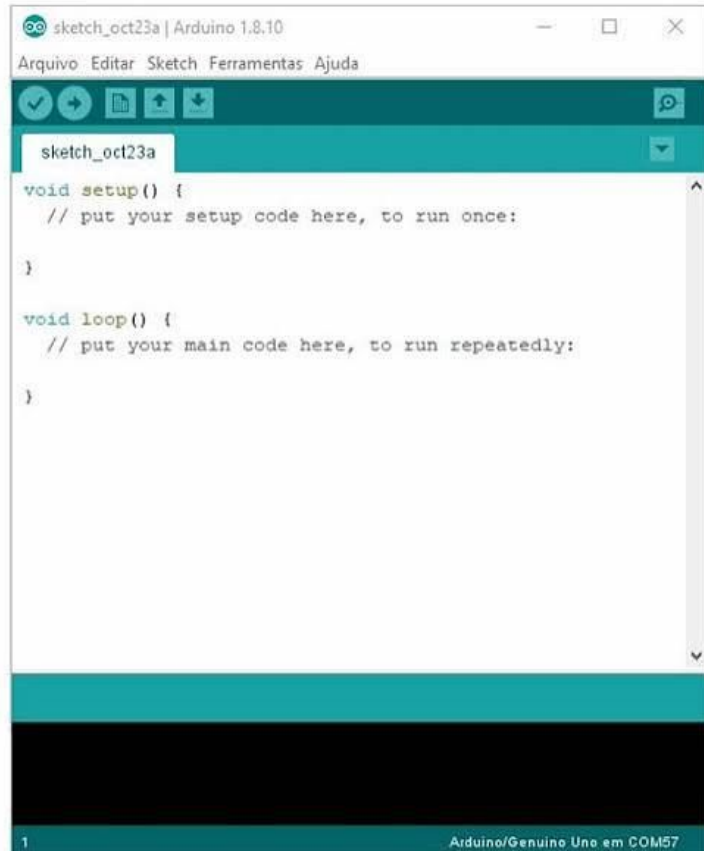
```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode( 8, INPUT);
}

void loop()
{
  if( digitalRead(8) == 1 )
  {
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
    delay(500);
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    delay(500);
  }
  else
  {
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
  }
}
```

```
1 :10000000C946100C947300C947300C947300B6
2 :10001000C947300C947300C947300C94730094
3 :10002000C947300C947300C947300C94730084
4 :10003000C947300C947300C947300C94730074
5 :10004000C9458010C947300C947300C9473007E
6 :10005000C947300C947300C947300C94730054
7 :10006000C947300C947300C947300C9473004001
8 :100070002900000000 26 :10018000111080C0D095DE23DC938F8FDF91CF9192
9 :10008000250028002B 27 :100190001F910895DE2BFCFCF93DF9390EFC0101
10 :100090000202020203 28 :1001A000E656FF4F24918A579F4FFC018491892384
11 :1000A0004080010204 29 :1001B000D1F090E080F991FFC01EE58FF4FA591F9
12 :1000B0000008000201 30 :1001C000B491FC01E458FF4FC591D49161108C068
13 :1000C000000011241F 31 :1001D0009F87F8948C91E22FE0958E238C9328121
14 :1000D000A0E0B1E001 32 :1001E000E223E8839FBDF91CF9108958F87F89402
15 :1000E000A2010C9436 33 :1001F000EC91E22BEC938F8F6CF3F87F894890150
16 :1000F000813099F082 34 :10020000050190910601A0910701B09108010196A11DED
17 :10010000C9F08430B1 35 :10021000A89E05C021 49 :1003000090910601A0910701B09108010196A11DED
18 :1001100080008F780 36 :10022000BA2F9A2F96 51 :10031000B11D8093050190930601A0930701B0934E
19 :10012000809584B58F 37 :10023000811D911D44 52 :1003200000801BF91AF919F819F913F912F910F90B5
20 :10013000B000089580 38 :10024000D1F7089581 53 :100330000F8E0F901F90189526E23F0296A11D5F
21 :10014000DF93282F30 39 :10025000EE92FF920F 54 :10034000B11D2CF789484B5826E084B084B58160BC
22 :10015000E656FF4FD4 40 :10026000DD24D394E1 55 :10035000E4B085B5826E085B5816085B5809190
23 :10016000A9F0162F81 41 :100270008A9598096A 56 :10036000E60816080936E00109281008091810008
24 :10017000FF1FE458FF 42 :10028000C21AD108E1 57 :10037000E26809381008091810081608093810000
43 :10029000A11CB11CCE 58 :10038000809180008160809380008091B1008460C2
44 :1002A0007A00846080937A0080917A0082608093E2
45 :1002B00001F920F9201 59 :10039000809381008091B00081608093800809123
46 :1002C00080680937A001092C10061E08DE00E9405
47 :1002D0000301B09104 60 :1003A0007A00846080937A0080917A0082608093E2
48 :1002E000058F50196A1 61 :1003B0007A0080917A00816080937A0080917A003F
49 :1002F00090930201A 62 :1003C00080680937A001092C10061E08DE00E9405
63 :1003D000CC00C6EBD0E02EA10E09EE8E92E90E0F7
64 :1003E000F92E20E0C22E0E0D22EFE18491F801D9
65 :10040000A490F701B490BB2041F181110E947500C6
66 :10041000E2BDF0E0EE0FF1F859FF4FA591B491CF
67 :100420008C91A822D1F061E08CE00E949E060E0F7
68 :100430008C91A822D1F061E08CE00E949E060E0F7
69 :100440009E061E08DE00E949E00E942201C11486
70 :10045000D10499F20E94000D0CF60E08CE00E94AD
71 :100460009E060E08DE00E949E00F1CF894FFCFE7
72 :00000001FF
```

5. Software de Desenvolvimento (IDE)

Software de Desenvolvimento (IDE)



Framework x Bare metal (C e Asm)

// Código .ino

```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode( 8, INPUT);
}
```

void loop()

```
{
  if( digitalRead(8) == 1 )
  {
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
    delay(500);
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    delay(500);
  }
  else
  {
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
  }
}
```

// Código .c

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
  DDRB = 0b00110000;
  PORTB = 0;
  while (1)
  {
    if( PINB & (1<<0))
    {
      PORTB |= (1<<4);
      PORTB &= ~(1<<5);
      _delay_ms(500);
      PORTB ^= (1<<4 | 1<<5);
      _delay_ms(500);
    }
    else
    {
      PORTB &= ~(1<<4);
      PORTB &= ~(1<<5);
    }
  }
}
```

; Código .asm

; Inclusão das definições de pinos fornecidas pelo fabricante
.include "m328pdef.inc"

; Programa alocado no endereço 0 da memória de código.
.ORG 0x0000

; PB5, PB4: out; PB0: in; 0b00110000 = 0x30
LDI R16,0x30
OUT DDRB,R16

DESLIGA_LEDS:

LDI R16,0x00
OUT PORTB,R16

INICIO:

IN R16,PINB
SBRS R16,0
RJMP DESLIGA_LEDS

RCALL DELAY
SBI PORTB,PB5
CBI PORTB,PB4
RCALL DELAY
CBI PORTB,PB5
SBI PORTB,PB4
RJMP INICIO

DELAY:

LDI R16,0
LDI R17,0
LDI R18,41

DELAY_LOOP:

INC R16
BRNE DELAY_LOOP
INC R17
BRNE DELAY_LOOP
DEC R18
BRNE DELAY_LOOP
RET

Framework (Arduino) x Bare metal (C e Asm)

Arduino 3136 Bytes

```

1 :10000000C946100C947300C947300C947300B6
2 :10010000C947300C947300C947300C94730094
3 :10002000C947300C947300C947300C94730084
4 :10003000C947300C947300C947300C94730074
5 :10004000C9458010C947300C947300C9473007E
6 :10005000C947300C947300C947300C94730054
7 :10006000C947300C947300C947300C94730050
8 :100070002900000000 25 :10018000111108C0D095DE23DC938F8DF91CF9192
9 :10008000250028002B 26 :100190001F910895DE2BF8CF93DF9390E0FC0101
10 :100090000202020203 27 :1001A000E656FF4F24918A579F4FC018491882384
11 :1000A0004080010204 28 :1001B000D1F090E0880F991FC01EE58FF4FA591F8
12 :1000B0000008000201 29 :1001C000B491FC01E458FF4FC591D49161110EC068
13 :1000C000000011241F 30 :1001D0009FB7F8948C91E22FE0958E238C93288121
14 :1000D000A0E0B1E001 31 :1001E000E223E8839FBDF91CF9108958F7F89402
15 :1000E000A2010C9436 32 :1001F000EC91E22EC938FBF6CF3FB7F894809150
16 :1000F000813099F082 33 :10020000050190910601A0910701B091080126B562
17 :10010000C9F08430B1 34 :10021000A89B05C021 49 :1003000090910601A0910701B09108010196A11DED
18 :1001100080008F7780 35 :10022000BA2FA92896 50 :100310000B11D8093050190930601A0930701B0934E
19 :10012000089584B5F8 36 :10023000811D911D4: 51 :100320000801BF91AF919F918F913F912F910F90B5
20 :10013000B000089580 37 :10024000D1F7089581 52 :100330000FBE0F901F90189526823F0296A11D5F
21 :10014000DF93282F30 38 :10025000EF92FF920F 53 :10034000B11DD2CF789484B5826084B84B58160BC
22 :10015000E656FF4FD4 39 :10026000DD24D394E1 54 :1003500084B84B5826084B84B58160BC809190
23 :10016000A9F6162F81 40 :100270008A099B096F 55 :1003600086E0816080936E00109281008091810008
24 :10017000FF1FE458FF 41 :10028000C21AD108E1 56 :100370008260809381008091810081608093810000
42 :10029000A11CB11CC1 57 :10038000809180008160809380008091B1008460C2
43 :1002A000EF90DF90C1 58 :100390008093B1008091B00081608093B000809123
44 :1002B0001F920F920F 59 :1003A0007A00846080937A0080917A0082608093E2
45 :1002C0009F93AF93B1 60 :1003B0007A0080917A00816080937A0080917A003F
46 :1002D0000301B091A4 61 :1003C000806880937A001092C10061E08DE00E9405
47 :1002E00058F50196A1 62 :1003D000CC061E08CE00E94CC0060E088E00E94C7
48 :1002F00090930201A6 63 :1003E000CC0C6EBD0E02EA10E09E8E92E90E0F7
64 :1003F000F92E0EC22E2E0ED22EFE018491F801D9
65 :10040000A490F701B490EB2041F181110E947500C6
66 :10041000EB2DF0E0E0EFF1FE859FF4FA591B491CF
67 :100420008C91A822D1F061E08CE00E949E0060E0F7
68 :100430008C91A822D1F061E08CE00E949E0060E0F7
69 :100440009E061E08DE00E949E00E942201C11486
70 :10045000DD10499F20E9400DD0CF60E08CE00E94AD
71 :100460009E060E08DE00E949E00F1CFB94FFCFE7
72 :00000001FF

```

C 712 Bytes

```

1 :10000000C943400C946C00C946C00C946C00F8
2 :10001000C946C00C946C00C946C00C946C00B0
3 :10002000C946C00C946C00C946C00C946C00A0
4 :10003000C946C00C946C00C946C00C946C0090
5 :10004000C946C00C946C00C946C00C946C0080
6 :10005000C946C00C946C00C946C00C946C0070
7 :10006000C946C00C946C00C946C00C946C0060
8 :10007000DEBFCDF0E94400C946100F894FFCFLA
9 :1000800080E384B9158690E3189B18C02C9A2D987A
10 :100090002FEF39E688E1215030408040E1F700C081
11 :1000A00000085B1892785B92FEF39E688E1215015
12 :1000B00030408040E1F700C0000E6CF2C982D983A
13 :0200C000E3CF8C
14 :1000C200EC010E9471000E947000CE010E94720039
15 :0600D200CE010E946E0049
16 :0400D800C940000084
17 :0400DC000E943E0040
18 :0200E000089581
19 :0200E20008957F
20 :0200E40008957D
21 :00000001FF

```

Asm 165 Bytes

```

1 :020000020000FC
2 :100000000E304B900E005B903B100FFBFCF06D05F
3 :10001000D29A2C9E03D02D982C9AF6CF00E010E062
4 :1000200029E20395F1F71395E1F72A95D1F70895A1
5 :00000001FF

```



Referências

LIMA, Charles Borges de; VILLAÇA, Marco Valério Miorim. **AVR e Arduino**: técnicas de projeto. 2. ed. Florianópolis: Edição dos Autores, 2012.

MARGOLIS, Michael. **Arduino Cookbook**. 2. ed. Sebastopol: O'Reilly Media, 2012.

PURDUM, Jack. **Beginning C for Arduino**: learn C programming for the Arduino and compatible microcontrollers. New York: Apress, 2012.

PEREA, Francis. **Arduino Essentials**. Birmingham: Packt Publishing, 2015.

LB TECNOLOGIA.  **Programação com Arduino**. [S. l.: s. n.], 2025. 1 playlist (27 vídeos). Disponível em: <https://youtube.com/@LuizBitencourt>. Acesso em: 23 jul. 2026.

Este material foi produzido com auxílio de
ferramentas de Inteligência Artificial:



Microcontroladores e Microprocessadores - MCMP

Profº José W. R. Pereira

jose.pereira@ifsp.edu.br

josewrpereira.github.io/docs

