

E209

Sistemas Microcontrolados e Microprocessados

Prof. João Magalhães

Inatel

CAMINHOS
QUE CONECTAM
COM O FUTURO

Microcontroladores AVR

Introdução

Os microcontroladores AVR RISC da fabricante ATMEL são de 8 bits e 32 bits, com arquitetura Harvard.

O código fonte (programa – firmware) para o microcontrolador necessita ser escrito, compilado, depurado e gravado.

Todas estas tarefas são realizadas com o suporte de softwares adequados.

32kB de memória flash para armazenamento de programas.

2kB de memória RAM estática para armazenamento de dados.

1kB EEPROM para armazenamento não-volátil.



CAMINHOS
QUE CONECTAM
COM O FUTURO

Microcontroladores AVR

Introdução

23 linhas de entrada/saída de propósito geral (GPIO).

32 registradores de propósito geral.

3 temporizadores/contadores.

USART (Universal Synchronous/Asynchronous Receiver Transmitter).

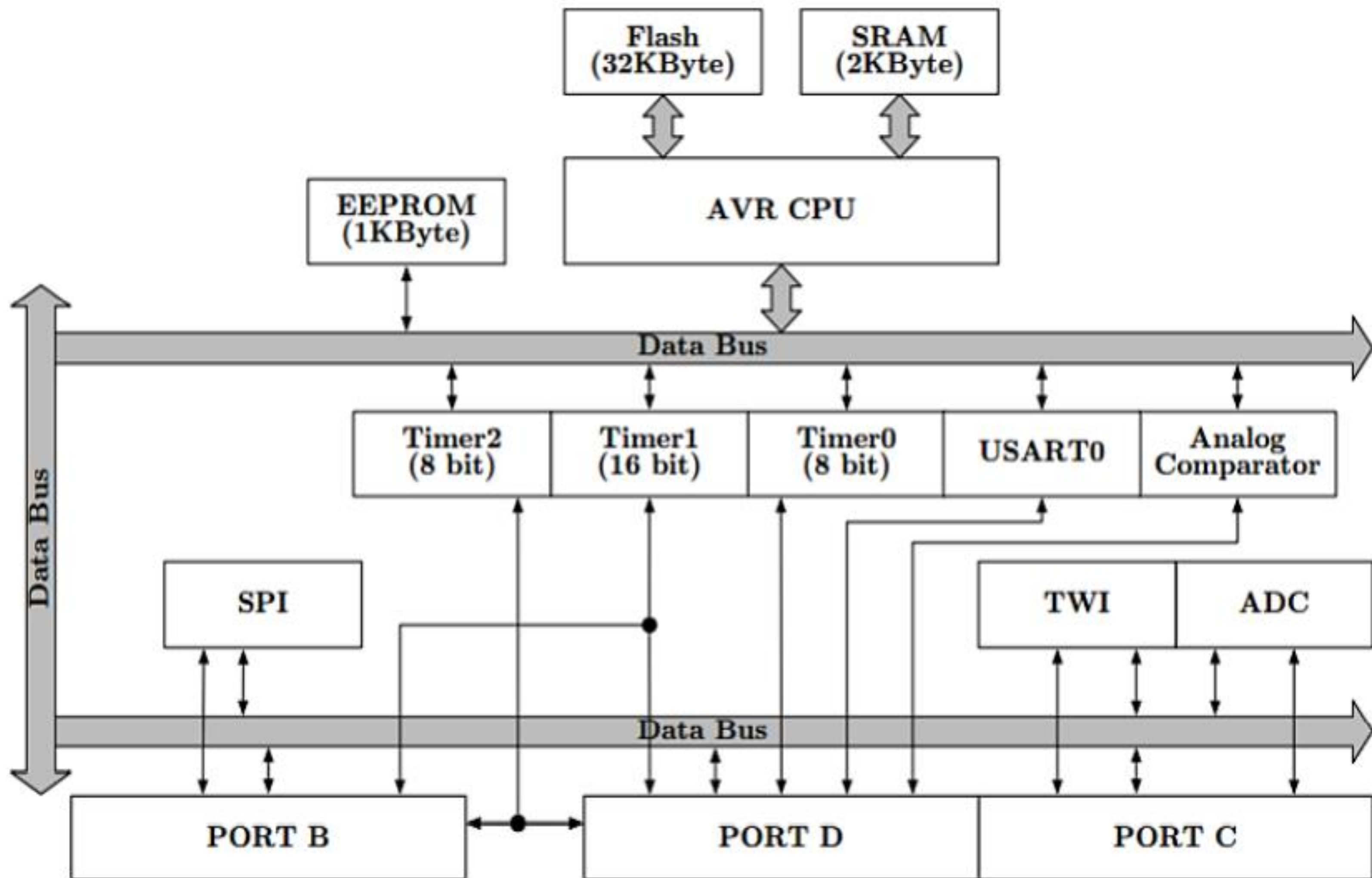
Porta serial I2C (Inter-Integrated Circuit), também chamada de TWI (Two Wire Interface).

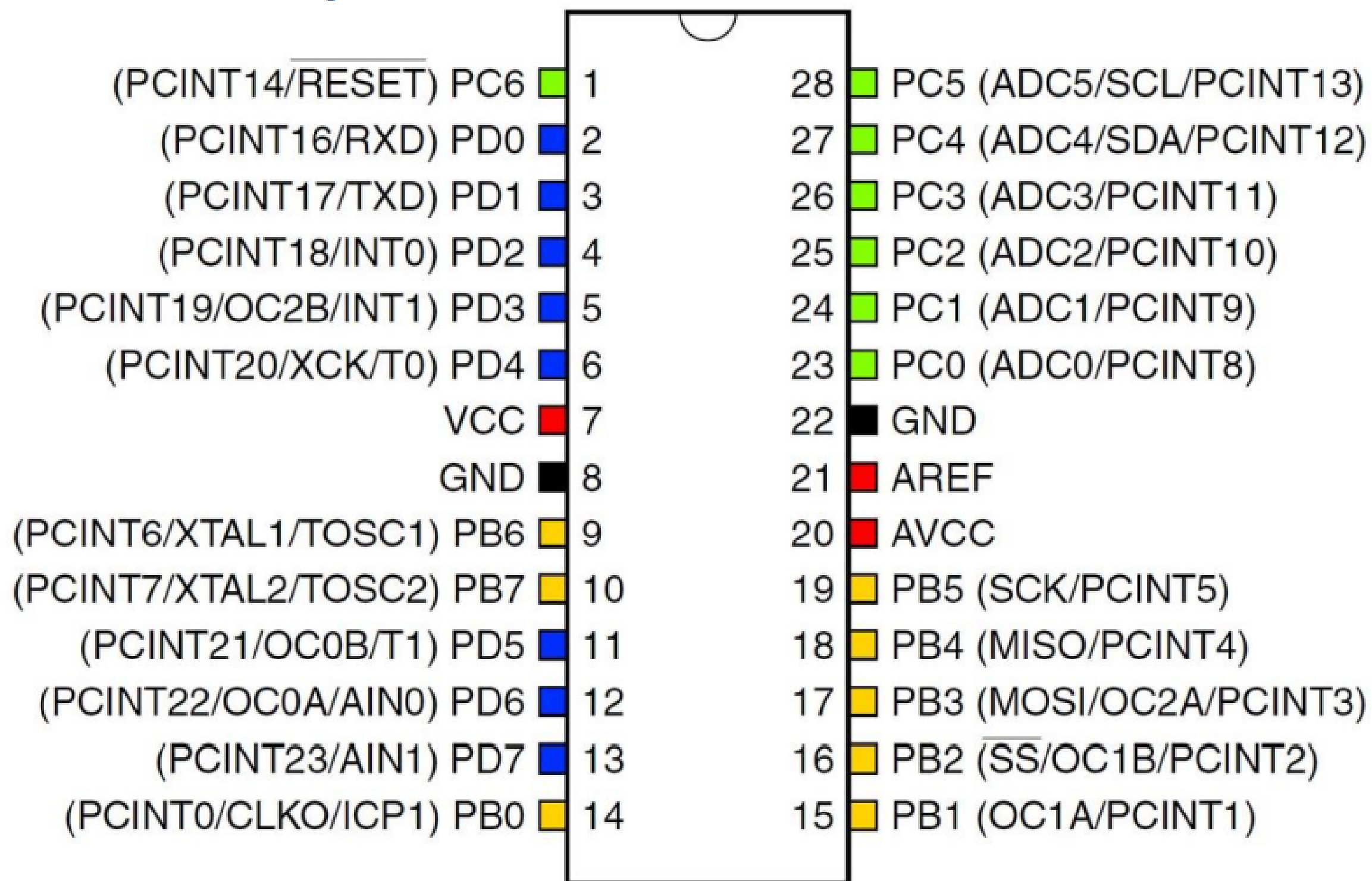
Porta serial SPI (Serial Peripheral Interface).

6 canais de 10 bits para conversão A/D



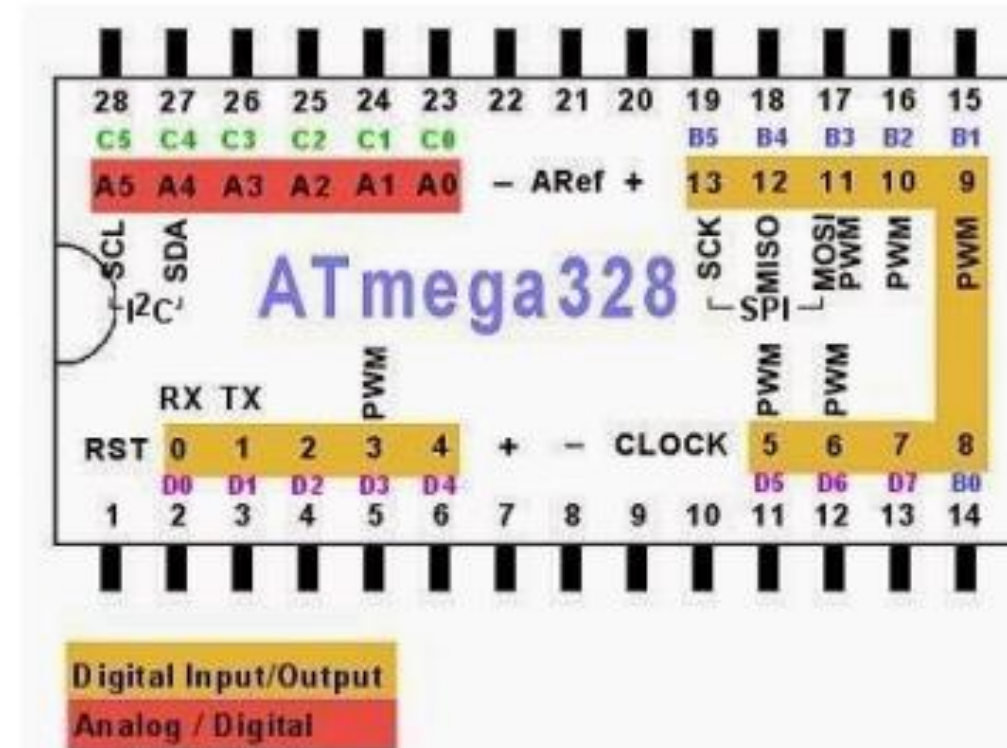
CAMINHOS
QUE CONECTAM
COM O FUTURO





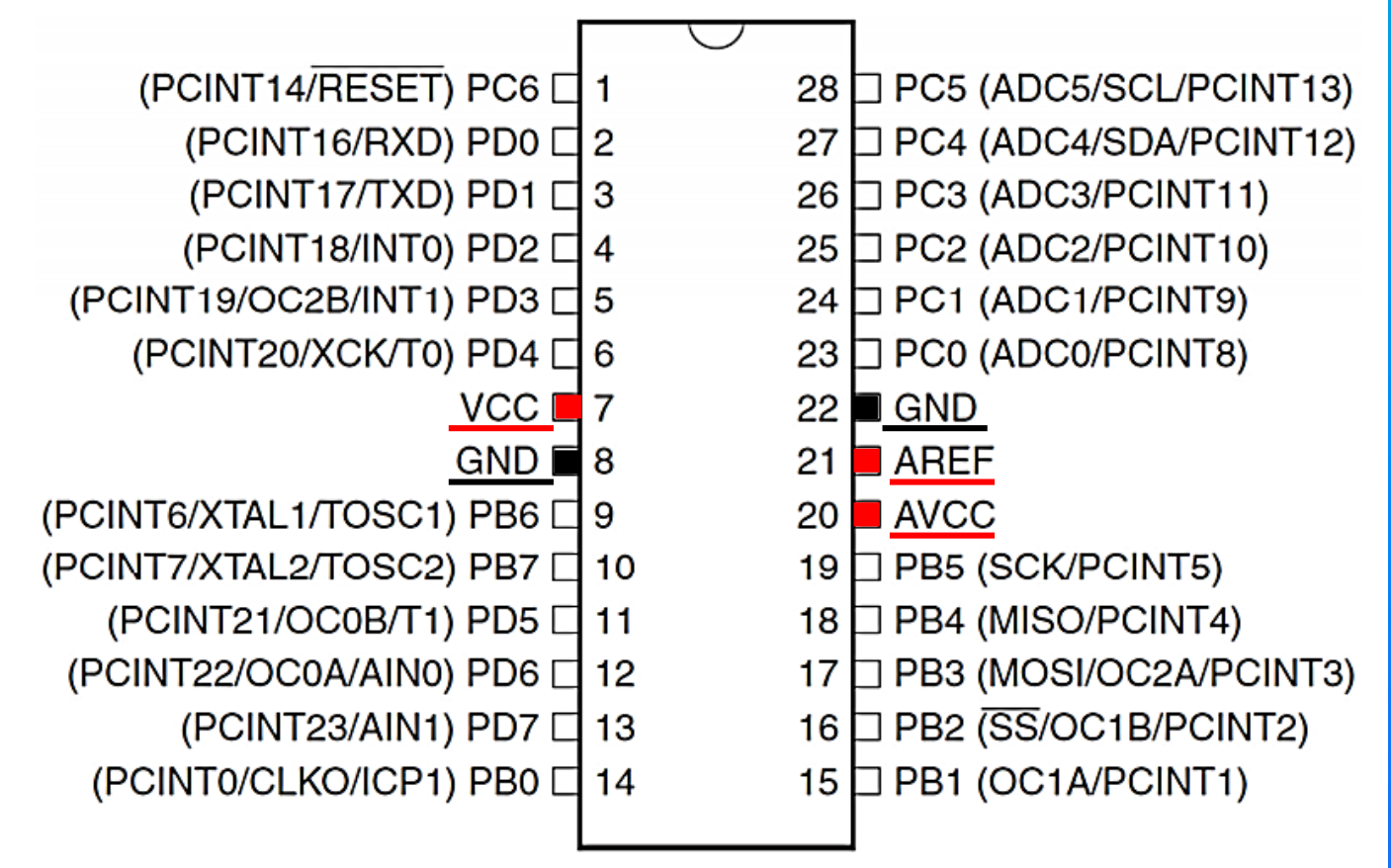
Correspondência entre ATMEGA e Arduino

| Arduino | ATmega328 | Arduino | ATmega328 | Arduino | Atmega328 |
|------------------|-----------|---------|-----------|---------|-----------|
| <i>Analog In</i> | PORTC | | PORTD | | PORTB |
| A0 | PC0 | 0 | PD0 | 8 | PB0 |
| A1 | PC1 | 1 | PD1 | 9 | PB1 |
| A2 | PC2 | 2 | PD2 | 10 | PB2 |
| A3 | PC3 | 3 | PD3 | 11 | PB3 |
| A4 | PC4 | 4 | PD4 | 12 | PB4 |
| A5 | PC5 | 5 | PD5 | 13 | PB5 |
| | | 6 | PD6 | | |
| | | 7 | PD7 | | |



Descrição dos Pinos

Pinos de Alimentação



PINOS DE ALIMENTAÇÃO

| | |
|------|--|
| VCC | Tensão de alimentação. |
| AVCC | Pino para a tensão de alimentação do conversor AD. Deve ser externamente conectado ao VCC, mesmo se o ADC não estiver sendo utilizado. |
| AREF | Pino para a tensão de referência analógica do conversor AD. |
| GND | Terra. |



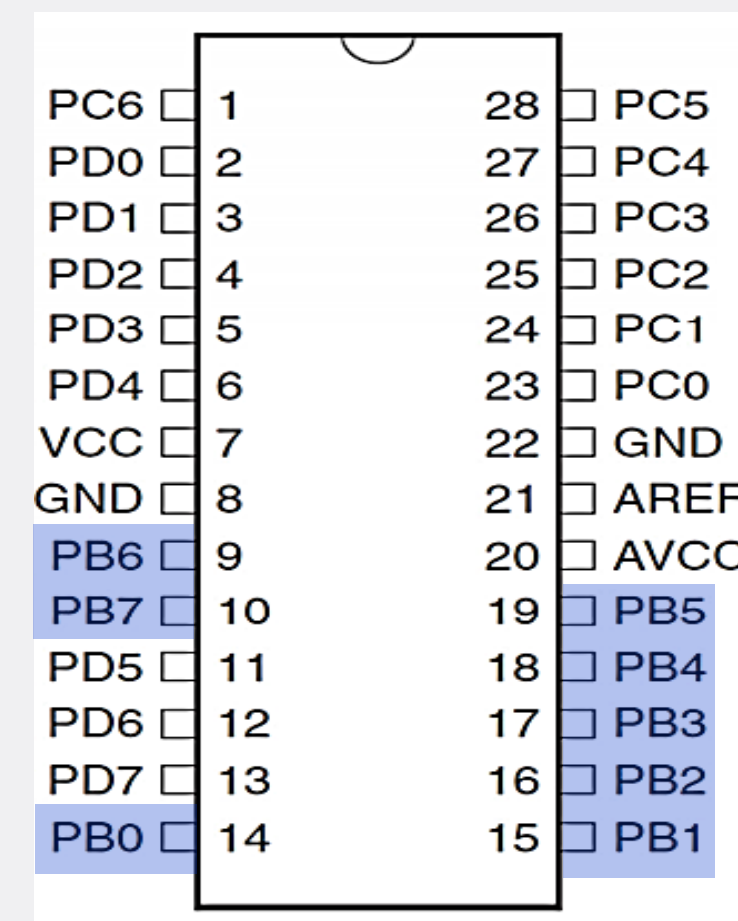
Descrição dos Pinos

Port B

É uma porta bidirecional de 8 bits, com resistores *pull-up* internos, selecionáveis para cada pino.

As funções alternativas para os pinos da PORTB são:

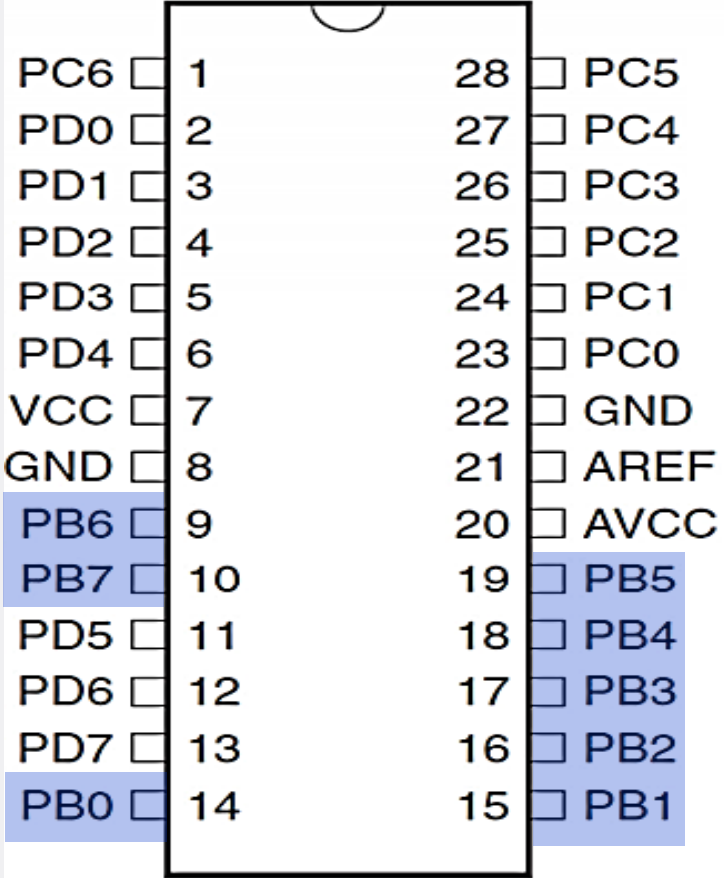
- XTAL
- SPI
- Comparadores de saída (*Output Compare*) para temporizadores



Descrição dos Pinos

Port B

| PORTB | |
|-------|--|
| PB0 | ICP1 – entrada de captura para o Temporizador/Contador 1. CLKO – saída de <i>clock</i> do sistema. PCINT0 – interrupção 0 por mudança no pino. |
| PB1 | OC1A – saída da igualdade de comparação A do Temporizador/Contador 1 (PWM). PCINT1 – interrupção 1 por mudança no pino. |
| PB2 | SS – pino de seleção de escravo da SPI (<i>Serial Peripheral Interface</i>). OC1B – saída da igualdade de comparação B do Temporizador/Contador 1 (PWM). PCINT2 – interrupção 2 por mudança no pino. |
| PB3 | MOSI – pino mestre de saída e escravo de entrada da SPI. OC2A – saída da igualdade de comparação A do Temporizador/Contador 2 (PWM). PCINT3 – interrupção 3 por mudança no pino. |
| PB4 | MISO – pino mestre de entrada e escravo de saída da SPI. PCINT4 – interrupção 4 por mudança no pino. |
| PB5 | SCK – pino de <i>clock</i> da SPI. PCINT5 – interrupção 5 por mudança no pino. |
| PB6 | XTAL1 – entrada 1 do oscilador ou entrada de <i>clock</i> externa. TOSC1 – entrada 1 para o oscilador do temporizador (RTC). PCINT6 – interrupção 6 por mudança no pino. |
| PB7 | XTAL2 – entrada 2 do oscilador. TOSC2 – entrada 2 para o oscilador do temporizador (RTC). PCINT7 – interrupção 7 por mudança no pino. |



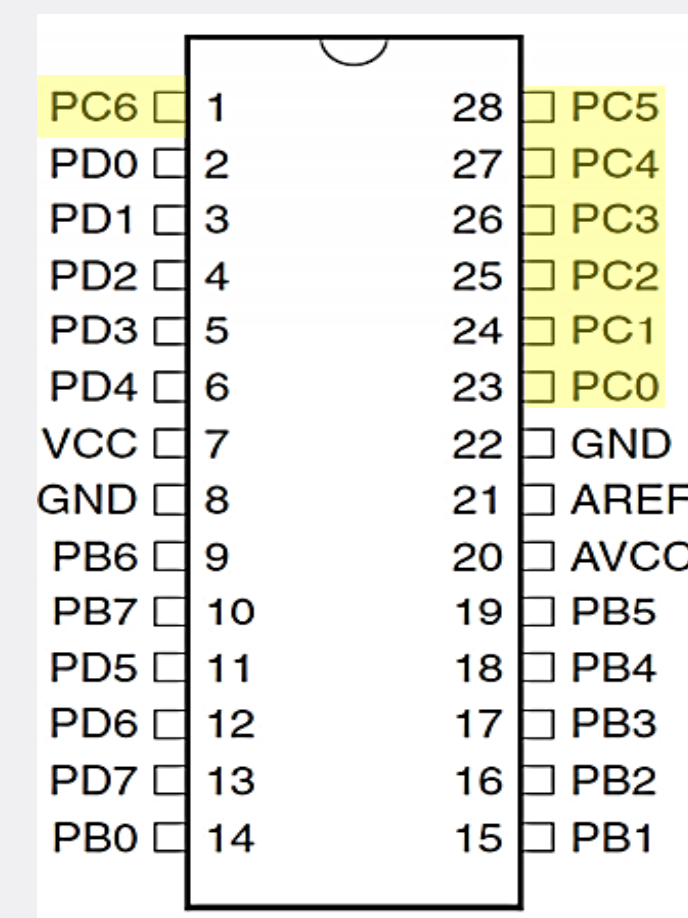
Descrição dos Pinos

Port C

É uma porta bidirecional de 7 bits, com resistores pull-up internos, selecionáveis para cada pino.

As funções alternativas para os pinos do PORTC são:

- Entradas analógicas (ADC)
- I2C.



Descrição dos Pinos

Port C

| PORTC | |
|-------|--|
| PC0 | ADC0 – canal 0 de entrada do conversor AD. PCINT8 – interrupção 8 por mudança no pino. |
| PC1 | ADC1 – canal 1 de entrada do conversor AD. PCINT9 – interrupção 9 por mudança no pino. |
| PC2 | ADC2 – canal 2 de entrada do conversor AD. PCINT10 – interrupção 10 por mudança no pino. |
| PC3 | ADC3 – canal 3 de entrada do conversor AD. PCINT11 – interrupção 11 por mudança no pino. |
| PC4 | ADC4 – canal 4 de entrada do conversor AD. SDA – entrada e saída de dados da interface a 2 fios (TWI – I2C). PCINT12 – interrupção 12 por mudança no pino. |
| PC5 | ADC5 – canal 5 de entrada do conversor AD. SCL – <i>clock</i> da interface a 2 fios (TWI – I2C). PCINT13 – interrupção 13 por mudança no pino. |
| PC6 | RESET – pino de inicialização. PCINT14 – interrupção 14 por mudança no pino. |

| | | | |
|-----|----|----|------|
| PC6 | 1 | 28 | PC5 |
| PD0 | 2 | 27 | PC4 |
| PD1 | 3 | 26 | PC3 |
| PD2 | 4 | 25 | PC2 |
| PD3 | 5 | 24 | PC1 |
| PD4 | 6 | 23 | PC0 |
| VCC | 7 | 22 | GND |
| GND | 8 | 21 | AREF |
| PB6 | 9 | 20 | AVCC |
| PB7 | 10 | 19 | PB5 |
| PD5 | 11 | 18 | PB4 |
| PD6 | 12 | 17 | PB3 |
| PD7 | 13 | 16 | PB2 |
| PB0 | 14 | 15 | PB1 |



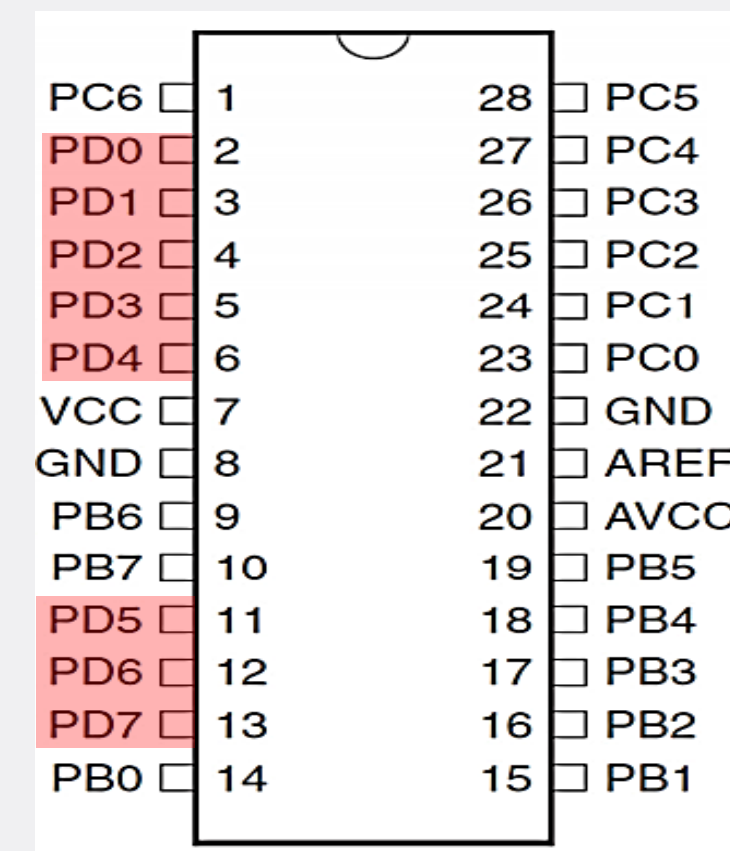
Descrição dos Pinos

Port D

É uma porta bidirecional de 8 bits, com resistores pull-up internos, selecionáveis para cada pino.

As funções alternativas para os pinos do PORTD são:

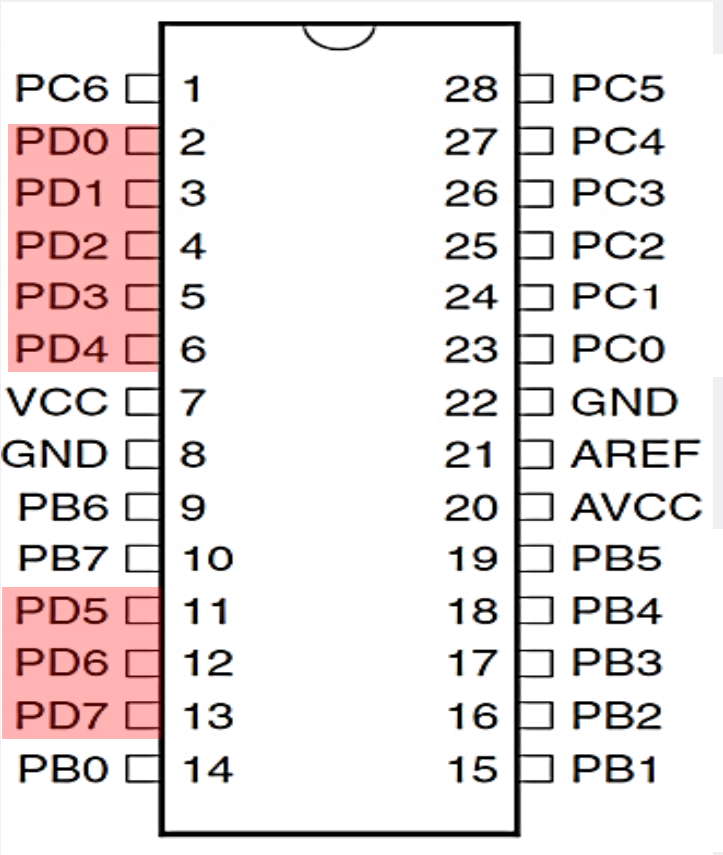
- Porta serial do USART.
- Interrupções externas INT0 e INT1.
- Comparadores de saída para temporizadores.



Descrição dos Pinos

Port D

| PORTD | |
|-------|---|
| PD0 | RXD – pino de entrada (leitura) da USART. PCINT16 – interrupção 16 por mudança no pino. |
| PD1 | TXD – pino de saída (escrita) da USART. PCINT17 – interrupção 17 por mudança no pino. |
| PD2 | INT0 – entrada da interrupção externa 0. PCINT18 – interrupção 18 por mudança no pino. |
| PD3 | INT1 – entrada da interrupção externa 1. OC2B – saída da igualdade de comparação B do Temporizador/Contador 2 (PWM) PCINT19 – interrupção 19 por mudança no pino. |
| PD4 | XCK – <i>clock</i> externo de entrada e saída da USART. T0 – entrada de contagem externa para o Temporizador/Contador 0. PCINT 20 – interrupção 20 por mudança no pino. |
| PD5 | T1 – entrada de contagem externa para o Temporizador/Contador 1. OC0B – saída da igualdade de comparação B do Temporizador/Contador 0 (PWM). PCINT 21 – interrupção 21 por mudança no pino. |
| PD6 | AIN0 – entrada positiva do comparador analógico. OC0A – saída da igualdade de comparação A do Temporizador/Contador 0 (PWM). PCINT 22 – interrupção 22 por mudança no pino. |
| PD7 | AIN1 – entrada negativa do comparador analógico. PCINT 23 – interrupção 23 por mudança no pino. |



Características

- 131 Instruções poderosas, a maioria executada em um único ciclo de relógio.
- Um banco de 32x8 registros de uso geral.
- Até 20 MIPS (Milhões de instruções por segundo) a 20 MHz.
- Um multiplicador de hardware on-chip de 2 ciclos.

ATmega328 Features

| | |
|------------------------|--|
| No. of Pins | 28 |
| CPU | RISC 8-Bit AVR |
| Operating Voltage | 1.8 to 5.5 V |
| Program Memory | 32KB |
| Program Memory Type | Flash |
| SRAM | 2048 Bytes |
| EEPROM | 1024 Bytes |
| ADC | 10-Bit |
| Number of ADC Channels | 8 |
| <u>PWM</u> Pins | 6 |
| Comparator | 1 |
| Packages (4) | 8-pin PDIP32-lead TQFP28-pad QFN/MLF32-pad QFN/MLF |
| Oscillator | up to 20 MHz |
| Timer (3) | 8-Bit x 2 & 16-Bit x 1 |



Características

- Memória de programa FLASH de 32 KB, programável dentro do sistema.
- Memória SRAM interna de 2 KBytes.
- Memória EEPROM de 1 KByte.
- 2 Temporizadores / Contadores de 8 bits.
- 1 Temporizador / Contador de 16 bits.

| | |
|----------------------------------|----------------|
| Enhanced Power on Reset | Yes |
| Power Up Timer | Yes |
| I/O Pins | 23 |
| Manufacturer | Microchip |
| SPI | Yes |
| I2C | Yes |
| Watchdog Timer | Yes |
| Brown out detect (BOD) | Yes |
| Reset | Yes |
| USI (Universal Serial Interface) | Yes |
| Minimum Operating Temperature | -40 C to +85 C |

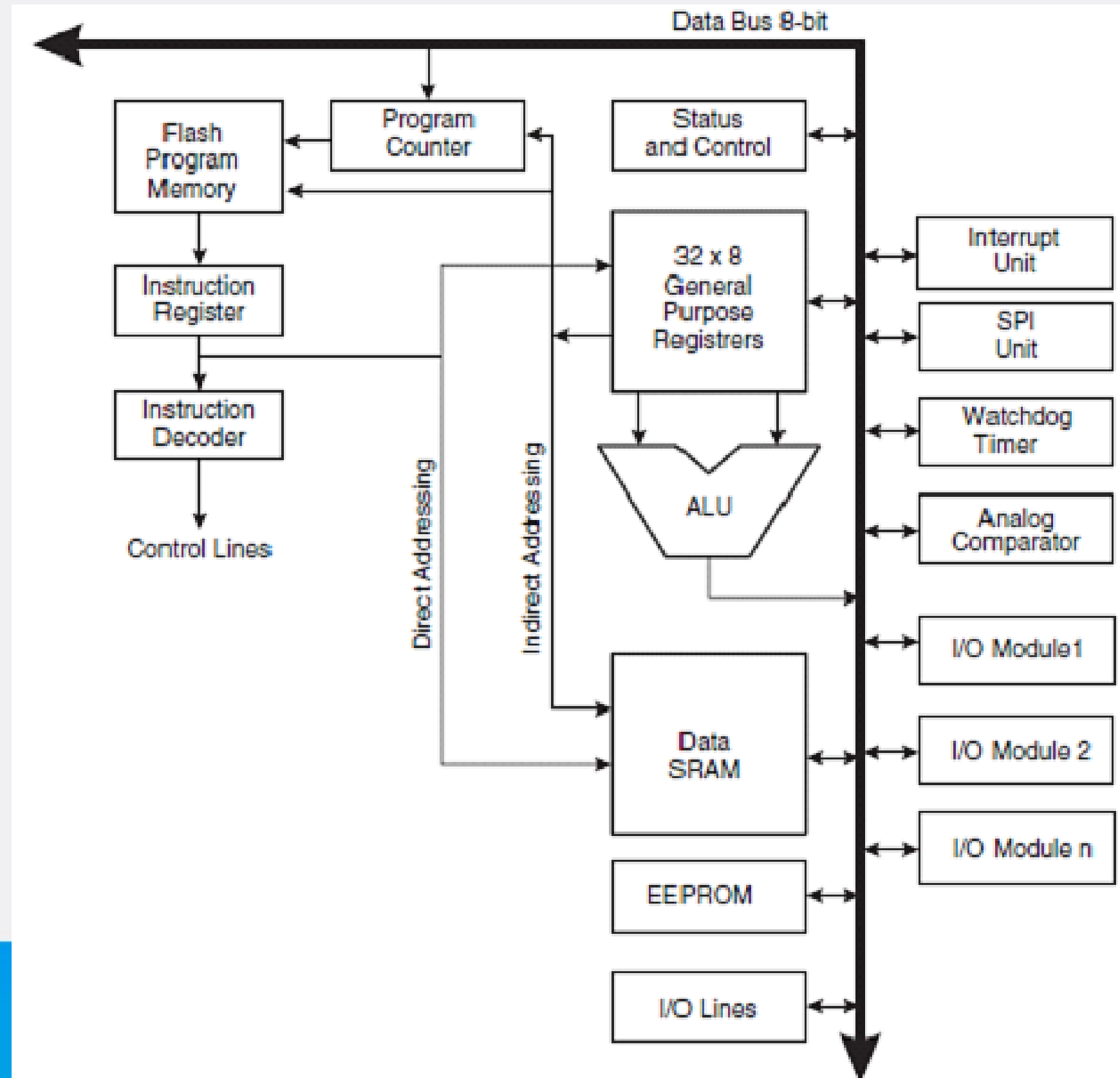


Características

- 6 canais PWM.
- 6 canais analógicos para o ADC.
- 1 porta serial USART.
- 1 interface serial SPI.
- 1 interface serial de 2 fios, compatível com I2C.
- 1 temporizador de vigilância.
- 1 Um comparador analógico on-chip.
- Interrupções.
- Vários modos de baixo consumo



Arquitetura



Linguagens de Programação

Assembly

O assembly é uma linguagem de baixo nível e permite obter o máximo desempenho de um microcontrolador, gerando o menor número de bytes de programa combinados a uma maior velocidade de processamento.

Todavia, o assembly só será eficiente se o programa estiver bem estruturado e empregar algoritmos adequados.

Programar em assembly exige muito esforço de programação.

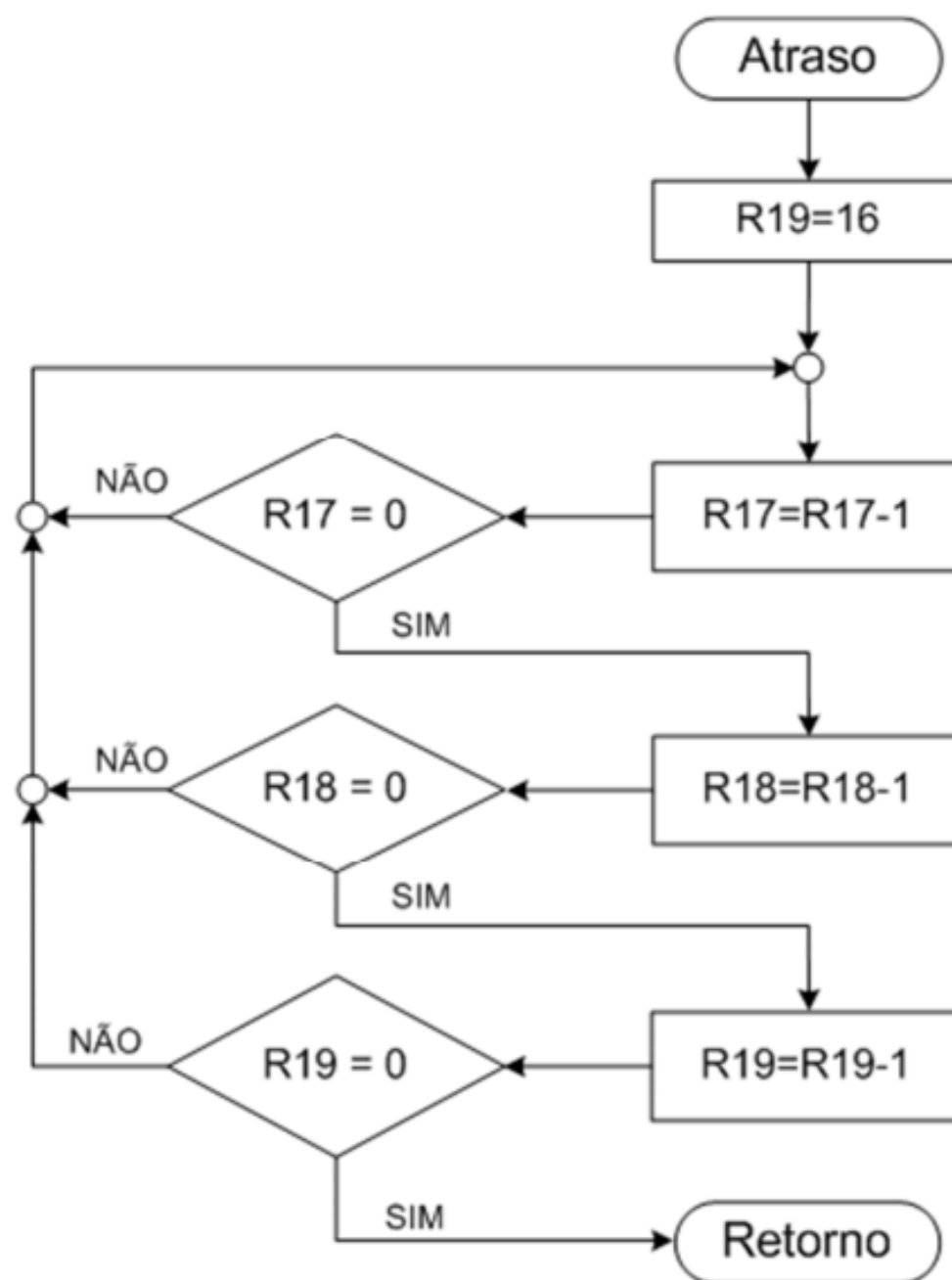
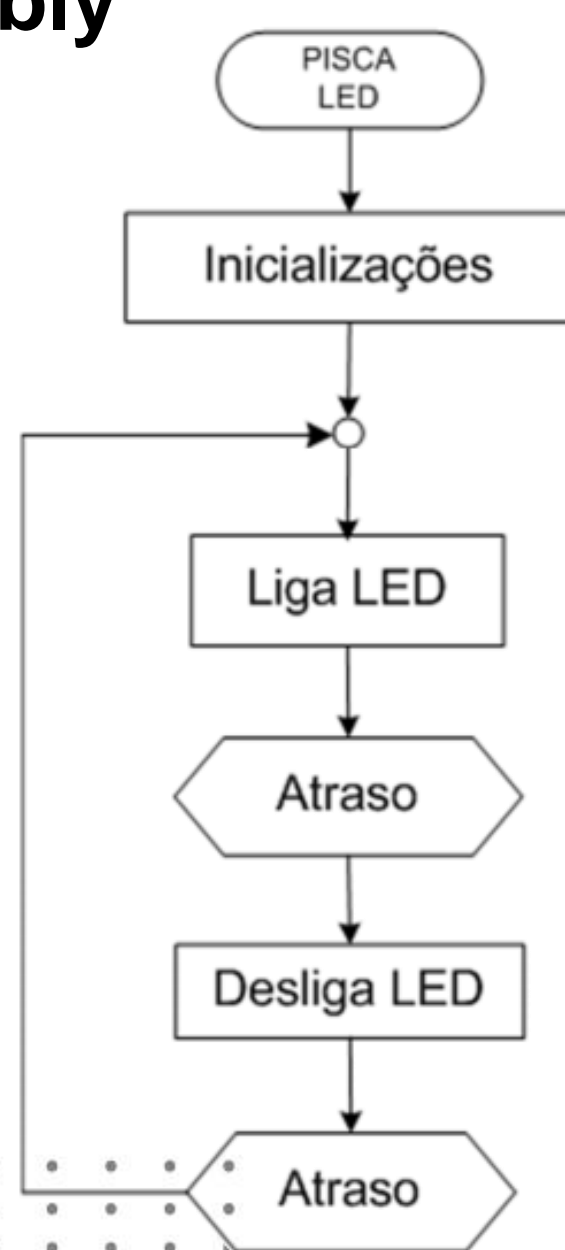
**Assembly é a linguagem da CPU do
microcontrolador!**



CAMINHOS
QUE CONECTAM
COM O FUTURO

Linguagens de Programação

Assembly



Linguagens de Programação

Assembly

```

.equ LED    = PB5           //LED é o substituto de PB5 na programação
.ORG 0x000                  //endereço de início de escrita do código

INICIO:
    LDI R16,0xFF             //carrega R16 com o valor 0xFF
    OUT DDRB,R16             //configura todos os pinos do PORTB como saída

PRINCIPAL:
    SBI PORTB, LED           //coloca o pino PB5 em 5V
    RCALL ATRASO             //chama a sub-rotina de atraso
    CBI PORTB, LED           //coloca o pino PB5 em 0V
    RCALL ATRASO             //chama a sub-rotina de atraso
    RJMP PRINCIPAL           //volta para PRINCIPAL

ATRASO:                      //atraso de aprox. 200ms (16 MHz)
    LDI R19,16
volta:
    DEC R17                  //decrementa R17, começa com 0x00
    BRNE volta               //enquanto R17 > 0 fica decrementando R17
    DEC R18                  //decrementa R18, começa com 0x00
    BRNE volta               //enquanto R18 > 0 volta decrementar R18
    DEC R19                  //decrementa R19
    BRNE volta               //enquanto R19 > 0 vai para volta
    RET

```

```

. . . . .
. . . . .
. . . . .

```

30 Bytes
15 instruções

Inatel

**CAMINHOS
QUE CONECTAM
COM O FUTURO**

Linguagens de Programação

Linguagem C

Com a evolução tecnológica (compiladores), o Assembly foi quase que totalmente substituído pela linguagem C.

As vantagens do uso do C são numerosas:

- Redução do tempo de desenvolvimento.
- O reuso do código é facilitado.
- Facilidade de manutenção.
- Portabilidade.



Linguagens de Programação

Linguagem C

- O problema de um código em C é que o mesmo pode consumir muita memória e reduzir a velocidade de processamento.
- Os compiladores tentam traduzir da melhor forma o código para o Assembly (antes de se tornarem código de máquina), mas esse processo não consegue o mesmo desempenho de um código escrito exclusivamente em Assembly.
- Como os compiladores C são eficientes para a arquitetura do AVR, a programação dos microcontroladores ATMega é feita em C.

Só existe a necessidade de se programar puramente em Assembly em casos críticos.



CAMINHOS
QUE CONECTAM
COM O FUTURO

Linguagens de Programação

Linguagem C

```

#define F_CPU 16000000UL    //define a frequência do microcontrolador 16MHz
#include <avr/io.h>          //definições do componente especificado
#include <util/delay.h>      //biblioteca para o uso das rotinas de delay

//Definições de macros
#define set_bit(Y,bit_x) (Y|=(1<<bit_x))    //ativa o bit x da variável Y
#define clr_bit(Y,bit_x) (Y&=~(1<<bit_x))    //limpa o bit x da variável Y
#define tst_bit(Y,bit_x) (Y&(1<<bit_x))      //testa o bit x da variável Y
#define cpl_bit(Y,bit_x) (Y^=(1<<bit_x))     //troca o estado do bit x da variável Y

#define LED PB5              //LED é o substituto de PB5 na programação

//-----
int main( )
{
    DDRB = 0xFF;             //configura todos os pinos do PORTB como saídas

    while(1)                 //laço infinito
    {
        set_bit(PORTB,LED); //liga LED
        _delay_ms(200);     //atraso de 200 ms
        clr_bit(PORTB,LED); //desliga LED
        _delay_ms(200);     //atraso de 200 ms
    }
}
//-----

```

216 Bytes

7,2 vezes maior



CAMINHOS
QUE CONECTAM
COM O FUTURO

Linguagens de Programação

Linguagem “Arduino”

```
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(200);               // wait for a second
  digitalWrite(led, LOW);   // turn the LED off by making the voltage LOW
  delay(200);               // wait for a second
}
```

| |
|-------------------------------|
| 30 bytes Assembly |
| 216 bytes C |
| 1084 bytes IDE Arduino |



Inatel

**CAMINHOS
QUE CONECTAM
COM O FUTURO**



Uso de Registradores

Introdução

Quando desejamos programar um microcontrolador, é fundamental o conhecimento de suas características (arquitetura).

Um dos pontos iniciais de grande importância é a manipulação de registradores.

Um registrador é um tipo de memória de pequena capacidade porém muito rápida, contida na CPU, utilizado no armazenamento temporário de dados durante o processamento.

Os registradores estão no topo da hierarquia de memória, sendo desta forma o meio mais rápido e de menor custo para armazenar um dado.

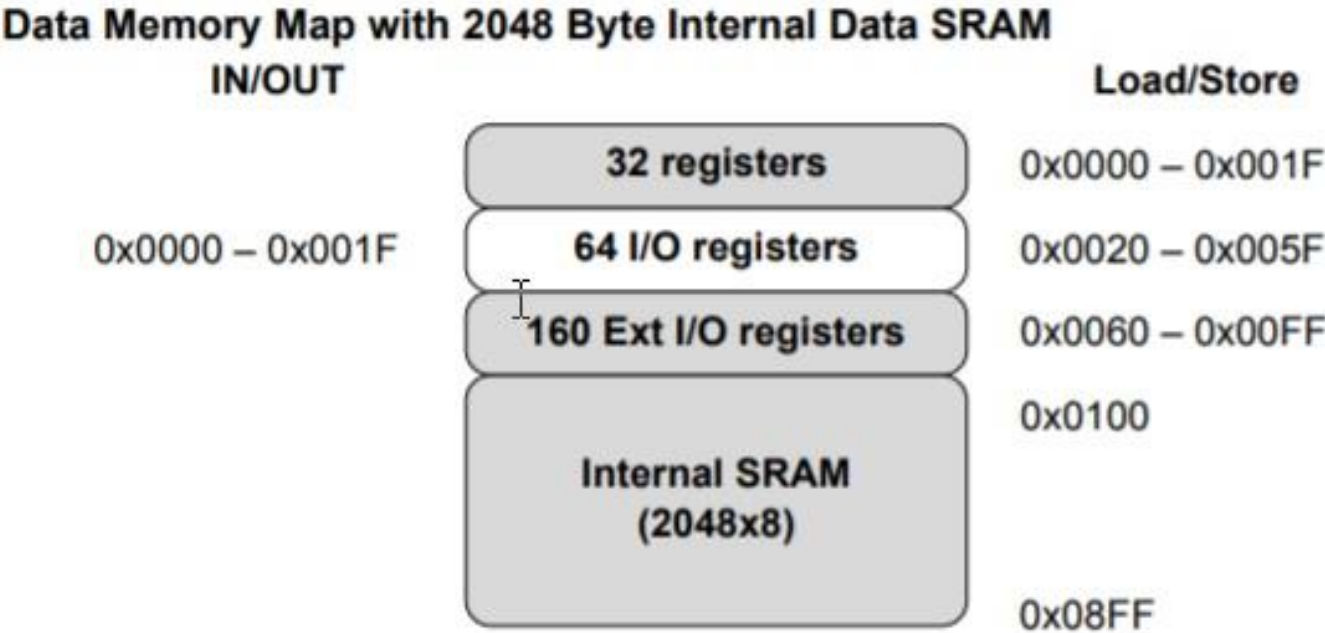


Uso de Registradores

Introdução

Os mesmos podem ser divididos em registradores de propósito geral ou de função específica (SFR).

Os registradores (SFR – Special Function Registers) recebem nomes específicos e têm função bem definida: guardar a configuração e o estado de funcionamento atual do microcontrolador.





Uso de Registradores

Introdução

Normalmente, cada bit do registrador tem uma função específica. Assim, temos um registrador para definir se as portas são de entrada ou de saída, ativar e desativar interrupções, apresentar o estado da CPU, etc.

Os registradores de I/O são o painel de controle dos Microcontroladores pois todas as configurações de trabalho, incluindo acesso às entradas e saídas, se encontram nessa parte da memória.



Uso de Registradores

Introdução

PORTB – The Port B Data Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0x05 (0x25) | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | PORTB |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Prof. João Magalhães

Horário de Atendimento:

- Segunda-feira: 17h30
- Quinta-feira: 19h30

E-mail: joao.magalhaes@inatel.br

Celular: (35) 99895-4450

Linkedin: <https://www.linkedin.com/in/joaomagalhaespaiva/>

