

EA075

Processadores de Propósito Geral: Microcontrolador ATmega328P



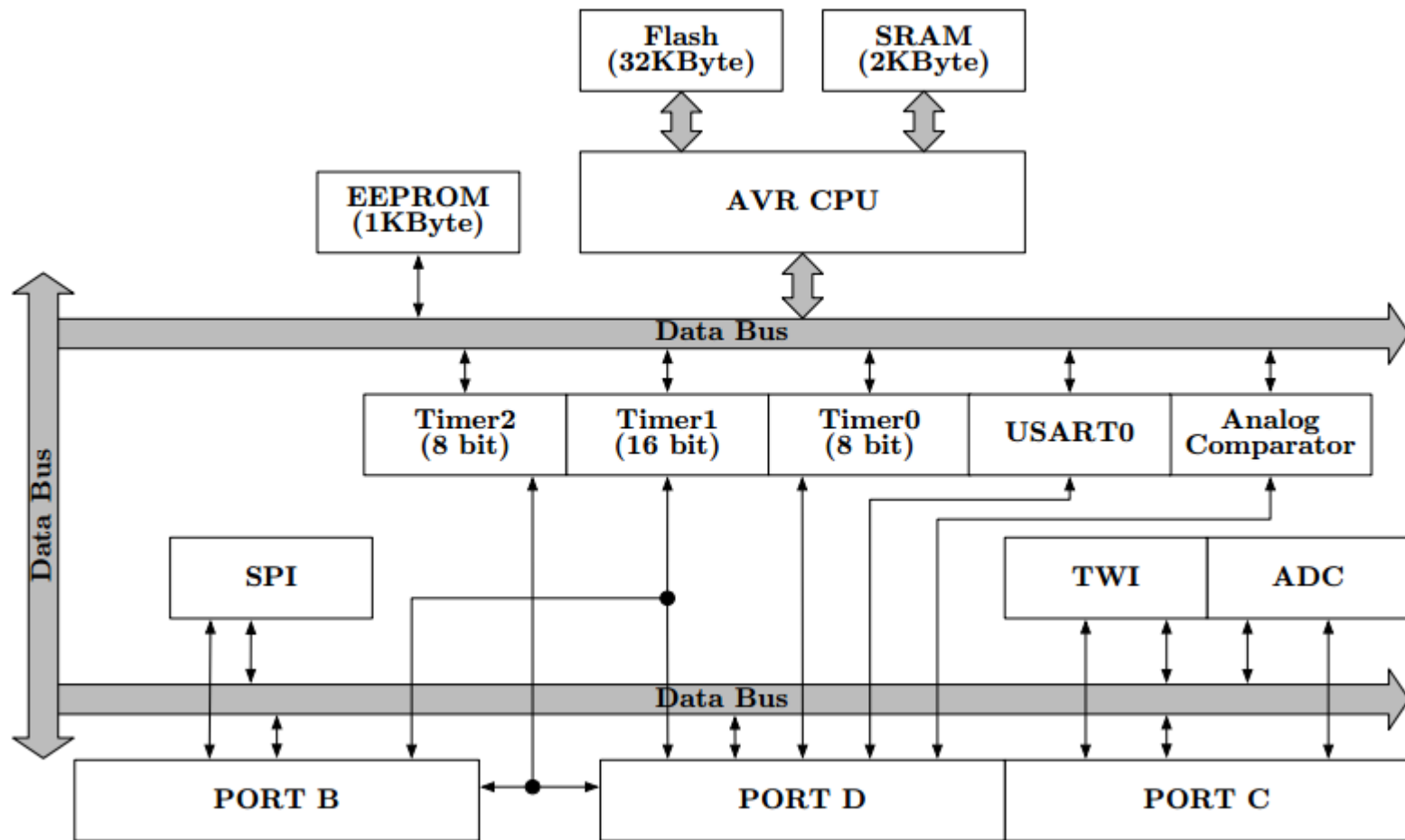
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)

Prof. Rafael Ferrari

Microcontrolador ATmega328P

- Microcontrolador de 8 bits.
- Baseado em um processador AVR RISC (*Reduced Instruction Set Computing*).
- Arquitetura Harvard (memórias de programa e dados independentes).
- 32kB de memória flash para armazenamento de programas.
- 2kB de memória RAM estática para armazenamento de dados.
- 1kB EEPROM para armazenamento não-volátil.
- 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.

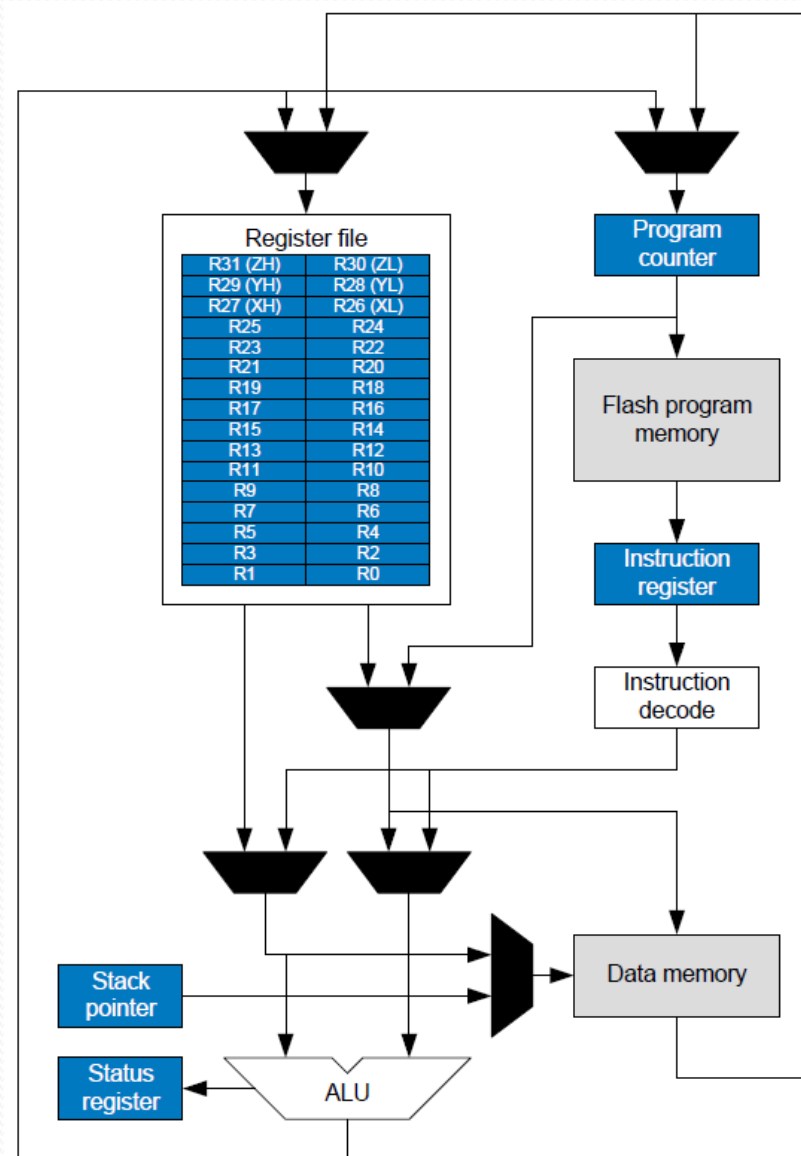
ATmega328P: Arquitetura



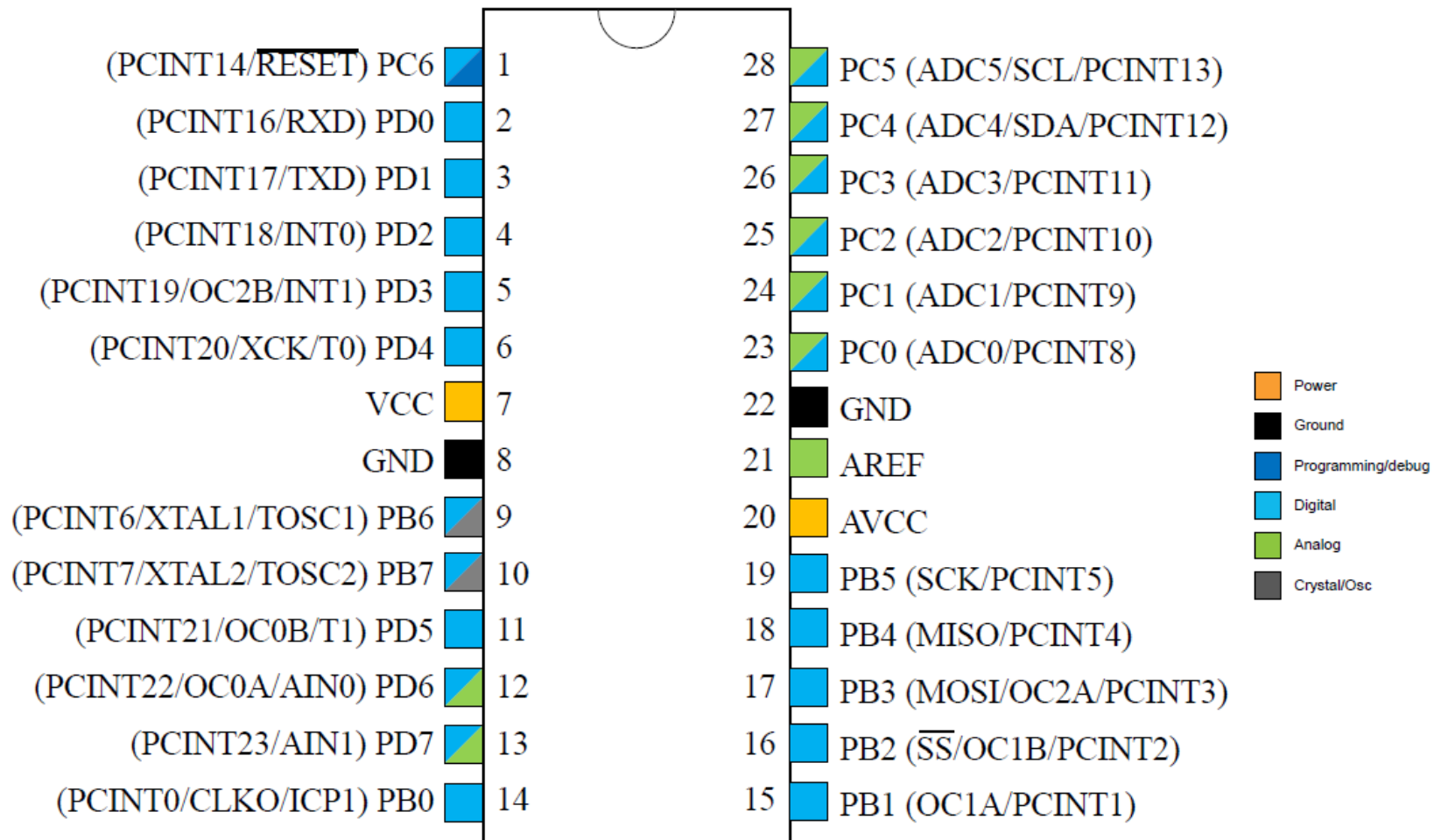
Microcontrolador ATmega328P

- 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).

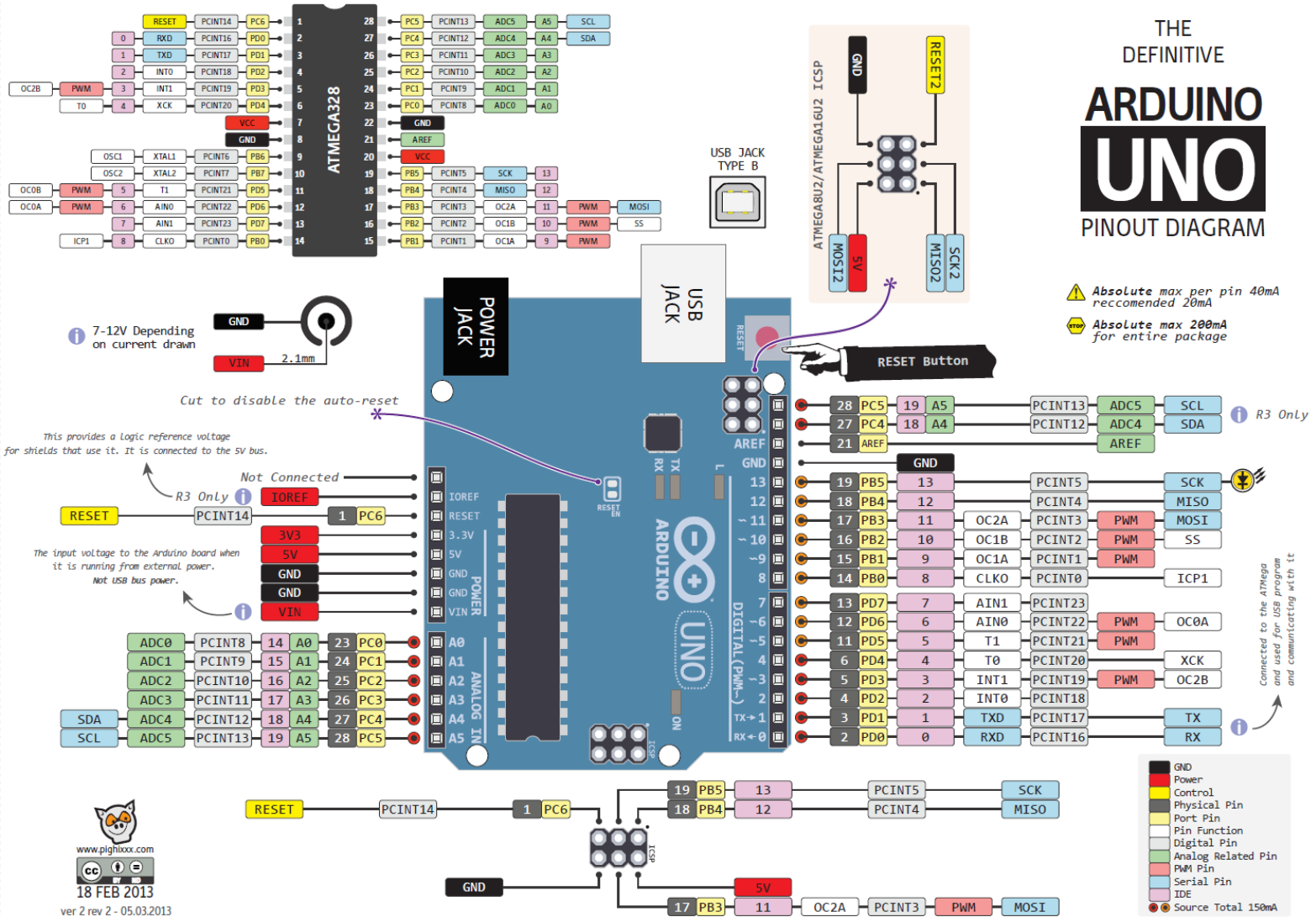
CPU AVR



Microcontrolador ATmega328P



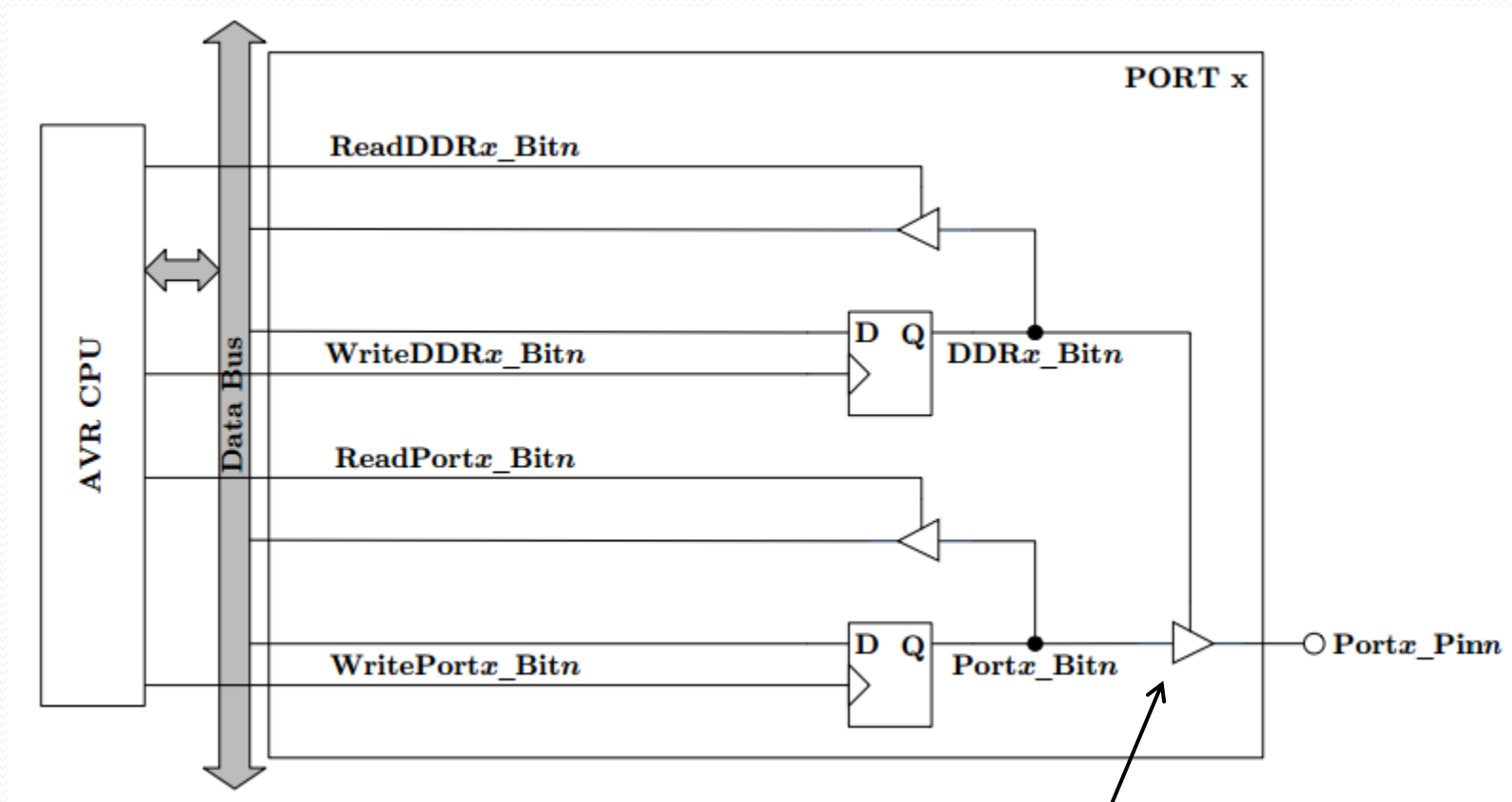
Arduino Uno



GPIO

- As portas de entrada e saída de propósito geral (*General-Purpose 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.

GPIO – Saída



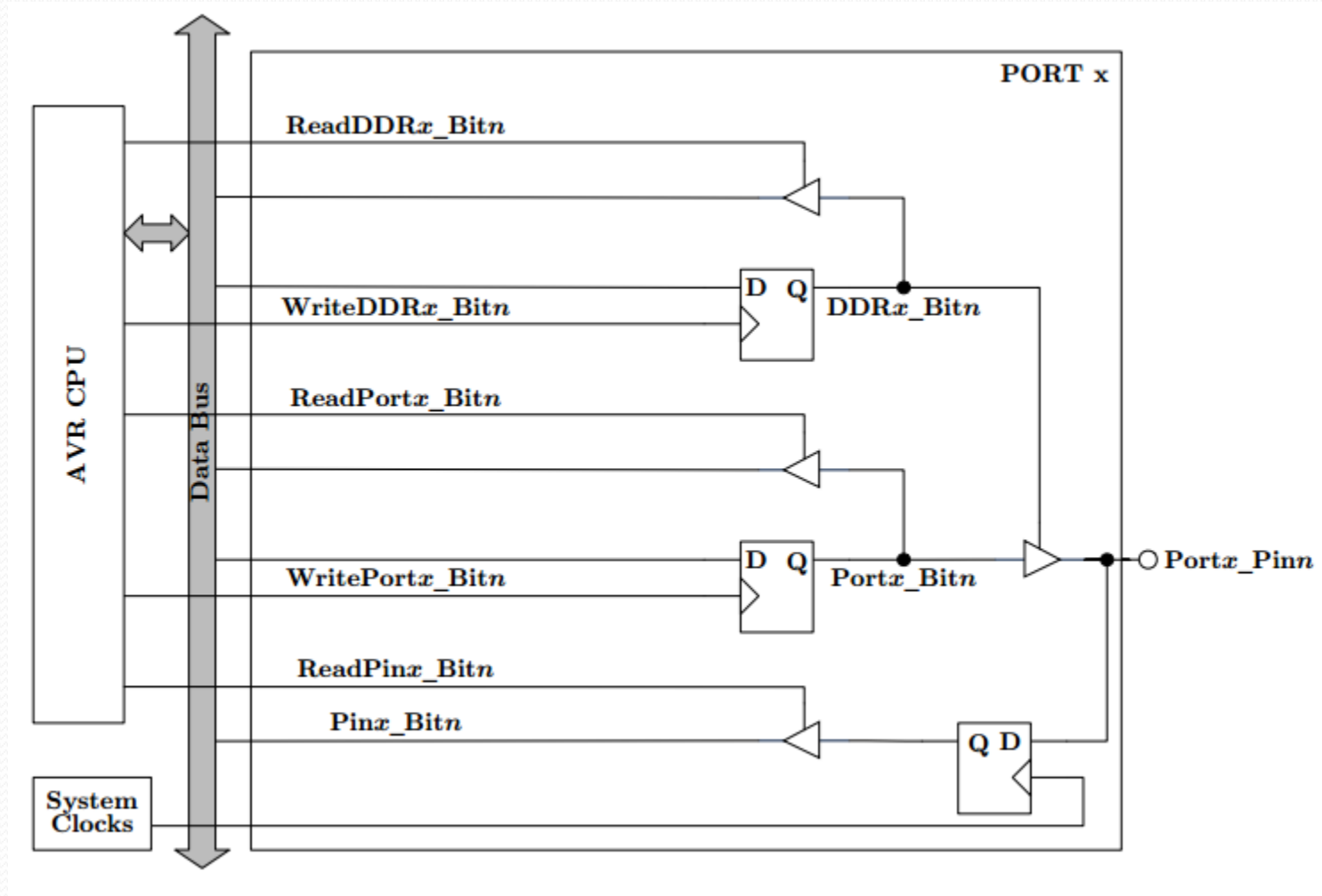
Buffer com saída tri-state

Enable = 1	Output = Input
Enable = 0	Output = Alta impedância

GPIO – Saída

- Dois registradores (flip-flops tipo D):
 - *Data Direction Register* (DDR)
 - Registrador de dados
- O registrador DDR habilita a conexão entre o registrador de dados e o pino do microcontrolador.
- O registrador de dados armazena o nível lógico de saída.
- Procedimento para escrever no pino de saída n da porta x :
 1. Escrever um nível lógico “1” em DDR_x_Bitn
 2. Escrever um nível lógico “1” ou “0” em $Portx_Bitn$

GPIO – Entrada



GPIO – Entrada

- Acrescenta-se ao circuito um registrador (flip-flop tipo D), denominado Pin, com sua entrada de dados conectada ao pino do microcontrolador.
- Se o conteúdo do registrador DDR é “0”, o registrador de saída é desconectado do pino e um dispositivo de entrada pode ser conectado com segurança ao pino do microcontrolador. Ou seja, o pino se transforma em uma entrada.
- A cada pulso do relógio, o nível lógico no pino é amostrado e armazenado no registrador Pin e seu conteúdo pode ser lido pelo processador.
- Procedimento para leitura do pino n da porta x :
 1. Escrever um nível lógico “0” em DDR_x_Bitn .
 2. Ler o conteúdo da linha Pin_x_Bitn (conteúdo do registrador Pin).

GPIO

- O microcontrolador Atmega328P possui 23 linhas GPIO, divididas entre três portas:
 - Porta B: 8 linhas
 - Porta C: 7 linhas
 - Porta D: 8 linhas
- Cada porta possui três registradores:
 - DDR x : define a direção da porta (“0” entrada, “1” saída).
 - PORT x : armazena o dado de saída.
 - PIN x : armazena o dado de entrada amostrado.

(OBS: x = B, C ou D)

- Os registradores são mapeados em memória e podem ser acessados através de operações de leitura e escrita em memória.

GPIO – Porta B

PORTB - THE PORT B DATA REGISTER

Bit	7	6	5	4	3	2	1	0
0x25	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

- PORTB7-0: GPIO data value stored in bit n .

DDRB - THE PORT B DATA DIRECTION REGISTER

Bit	7	6	5	4	3	2	1	0
0x24	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

- DDRB7-0: selects the direction of pin n . If DDRB n is written '1', then PORTB n is configured as an output pin. If DDRB n is written '0', then PORTB n is configured as an input pin.

PINB - THE PORT B INPUT PINS ADDRESS

Bit	7	6	5	4	3	2	1	0
0x23	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Read/Write	R	R	R	R	R	R	R	R
Default	-	-	-	-	-	-	-	-

- PINB7-0: logic value present on external pin n .

Endereços
dos
registradores

GPIO

- Para alterar o conteúdo dos registradores $DDRx$ e $PORTx$ é necessário realizar uma operação de escrita de um byte completo, mesmo que se deseje alterar apenas um dos bits.
- A fim de evitar a modificação indesejada dos bits do registrador que não devem ser alterados, o seguinte procedimento deve ser adotado:
 - Leitura e armazenamento do conteúdo do registrador em uma variável temporária.
 - Modificação do conteúdo de acordo com os novos valores que os bits devem assumir.
 - Escrita da variável modificada no registrador.

GPIO – Programando em C

- Usando **ponteiros**:

```
// criar o ponteiro e atribuir o endereço do registrador
unsigned char *portB_DDR;
portB_DDR = (unsigned char*) 0x24;
```

```
// Leitura do conteúdo armazenado no registrador DDRB
unsigned char dado;
dado = (*portB_DDR);
```

```
// Escrita
*portB_DDR = 0xFF; // todos os bits são de saída
```

```
// Alterando somente o bit 5
#define BIT5_MASK 0x20 // 0010 0000
```

```
// atribuir nível baixo ao bit 5
*portB_DDR = (*portB_DDR) & (~BIT5_MASK);
```

```
// atribuir nível alto ao bit 5
*portB_DDR = (*portB_DDR) | BIT5_MASK;
```


GPIO – Programando em C

- Usando variáveis predefinidas:

```
// Leitura do conteúdo armazenado no registrador DDRB  
dado = DDRB; // DDRB corresponde a um ponteiro para 0x24
```

```
// Escrita  
DDRB = 0xFF; // todos os bits são de saída
```

```
// Alterando somente o bit 5  
# define BIT5_MASK 0x20 // 0010 0000
```

```
// atribuir nível baixo ao bit 5  
DDRB = DDRB & (~BIT5_MASK);
```

```
// atribuir nível alto ao bit 5  
DDRB = DDRB | BIT5_MASK;
```

- Na IDE do Arduino, as variáveis predefinidas geralmente seguem a nomenclatura adotada no datasheet (DDRB, PORTB, PINB, etc).

GPIO – Programando em C

- Usando funções predefinidas da IDE (biblioteca Wiring):

```
int output_pin = 13;
int input_pin = 12;
int val;

void setup() {
    pinMode(output_pin, OUTPUT);
    pinMode(input_pin, INPUT);
}

void loop() {

    // Escrita: atribui nível alto ao pino 13 do Arduino
    digitalWrite(output_pin, HIGH);

    // Leitura do pino 12 do Arduino
    val = digitalRead(input_pin);
}
```

- Note que nesse caso nos referimos diretamente aos pinos da placa de desenvolvimento Arduino e não aos registradores do ATmega328P.

GPIO – Características Eléctricas

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0mA
DC Current V_{CC} and GND Pins	200.0mA

GPIO – Características Eléctricas

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0mA
DC Current V_{CC} and GND Pins	200.0mA

Interrupções

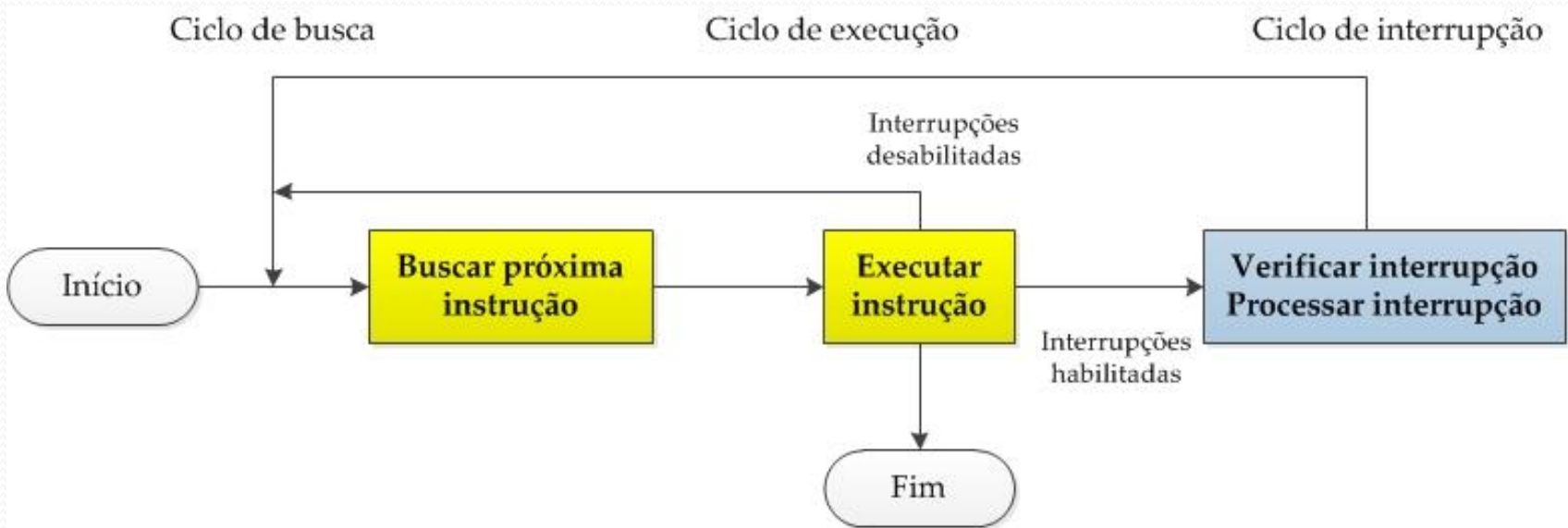
- Considere que o microcontrolador deve ascender um LED quando um botão é pressionado.
- O instante em que o botão é pressionado é absolutamente imprevisível da perspectiva do microprocessador / programa em execução.
- Há duas abordagens para se tratar um evento de natureza assíncrona:
 - Varredura
 - Interrupção

Interrupções

- **Primeira opção:** varredura (*polling*) – durante a execução do programa, ocorrem chamadas para uma rotina que verifica se a chave foi pressionada.
 - Fácil implementação.
 - Desperdício de ciclos de execução do processador.
- **Alternativa:** o processador suporta interrupções – um sinal de interrupção indica para o processador quando uma solicitação de atendimento foi feita por um dispositivo externo (periférico).
 - Neste caso, o processador suspende o programa em execução e desvia para a rotina de serviço de interrupção (*interrupt service routine, ISR*).

Interrupções

- O mecanismo de interrupção passa a fazer parte do ciclo de instrução do processador.



Interrupções: ATmega328p

- Cada periférico presente no microcontrolador pode gerar uma ou mais interrupções internas.
- Há também interrupções associadas a eventos externos.
- **Interrupções mascaráveis:** cada interrupção possui um bit de habilitação que precisa ser ativado juntamente com o *Global Interrupt Enable* bit no registrador de status para que interrupções ocorram.
- **Interrupções fixas:** cada interrupção é mapeada para um endereço fixo de memória.
- As interrupções são desabilitadas automaticamente quando uma ISR está sendo executada. Entretanto, é possível habilitá-las manualmente dentro da ISR e assim permitir que interrupções sejam atendidas durante a execução da ISR.

Interrupções: ATmega328P

Pri.	Address	Interrupt Source	ISR C Function Name	Description
1	0x0000	RESET		System reset (power-on)
2	0x0002	INT0	INT0_vect	External Interrupt Request 0
3	0x0004	INT1	INT1_vect	External Interrupt Request 1
4	0x0006	PCINT0	PCINT0_vect	Pin Change Interrupt Request 0
5	0x0008	PCINT1	PCINT1_vect	Pin Change Interrupt Request 1
6	0x000A	PCINT2	PCINT2_vect	Pin Change Interrupt Request 2
7	0x000C	WDT	WDT_vect	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	TIMER2_COMPA_vect	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	TIMER2_COMPB_vect	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	TIMER2_OVF_vect	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	TIMER1_CAPT_vect	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	TIMER1_COMPA_vect	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	TIMER1_COMPB_vect	Timer/Counter1 Compare Match B
14	0x001A	TIMER1 OVF	TIMER1_OVF_vect	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	TIMER0_COMPA_vect	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	TIMER0_COMPB_vect	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	TIMER0_OVF_vect	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI_STC_vect	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART_RX_vect	USART Receive Complete
20	0x0026	USART, UDRE	USART_UDRE_vect	USART Data Register Empty
21	0x0028	USART, TX	USART_TX_vect	USART Transmit Complete
22	0x002A	ADC	ADC_vect	ADC Conversion Complete
23	0x002C	EE READY	EE_READY_vect	EEPROM Ready
24	0x002E	ANALOG COMP	ANALOG_COMP_vect	Analog Comparator
25	0x0030	TWI	TWI_vect	2-wire Serial Interface
26	0x0032	SPM READY	SPM_READY_vect	Store Program Memory Ready

A prioridade da interrupção é determinada por sua posição na tabela: quanto menor o endereço maior é a prioridade.

A rotina de interrupção em C deve ser definida na forma:

```
ISR (INT0_vect) {
```

ATmega328P: Interrupções Externas

- São interrupções geradas por dispositivos externos ao microcontrolador.
- Há dois tipos:
 - Interrupções geradas nos pinos INT0 e INT1: permitem um número maior de configurações e têm maior prioridade. Podem ser ativadas nas bordas de subida, descida, em ambas ou por nível lógico baixo do sinal de interrupção. São configuradas através dos registradores EICRA e EIMSK. Quando disparadas, ativam flags no registrador EIFR.
 - *Pin Change Interrupts*: têm menor prioridade que as anteriores e podem ser ativadas quando há uma mudança de nível em um dos 23 pinos PCINT. São configuradas por meio dos registradores PCMSK0, PCMSK1, PCMSK2 e PCICR. Flags no registrador PCIFR indicam quando essas interrupções são ativadas.

ATmega328P: Interrupções Externas

- EICRA – External Interrupt Control Register A
 - Define como INT0 e INT1 são ativadas (borda(s) ou nível)

Bit	7	6	5	4	3	2	1	0	
(0x69)	–	–	–	–	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 13-1. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

Table 13-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

ATmega328P: Interrupções Externas

- EIMSK – External Interrupt Mask Register
 - Habilita INT0 e INT1.

EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	–	–	–	–	–	–	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **INT0 - External Interrupt Request 0 Enable:** quando em nível alto, habilita a geração de interrupções a partir do pino INT0 (É necessário que a interrupção global também esteja ativa).
- **INT1 - External Interrupt Request 1 Enable:** quando em nível alto, habilita a geração de interrupções a partir do pino INT1 (É necessário que a interrupção global também esteja ativa).

ATmega328P: Interrupções Externas

- EIFR – External Interrupt Flag Register
 - Sinaliza quando INT0 e INT1 foram disparadas.

EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	–	–	–	–	–	–	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **INTF0 - External Interrupt Flag 0:** quando ocorre a condição de ativação de INT0, de acordo com as configurações em EICRA, esse bit assume nível alto.
- **INTF1 - External Interrupt Flag 1:** quando ocorre a condição de ativação de INT1, de acordo com as configurações em EICRA, esse bit assume nível alto.

ATmega328P: Interrupções Externas

- PCICR – Pin Change Interrupt Control Register
 - Habilita as interrupções PCIE0, 1 e 2.

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **PCIE0 – Pin Change Interrupt Enable 0:** habilita interrupções geradas pelos pinos PCINT 0 a 7.
- **PCIE1 – Pin Change Interrupt Enable 1:** habilita interrupções geradas pelos pinos PCINT 8 a 14.
- **PCIE2 – Pin Change Interrupt Enable 2:** habilita interrupções geradas pelos pinos PCINT 16 a 23.

ATmega328P: Interrupções Externas

PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
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	

- **Bit 7:0 – PCINT[7:0]: Pin Change Enable Mask 7...0**

Each PCINT[7:0] bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

ATmega328P: Interrupções Externas

PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Reserved**

This bit is an unused bit in the ATmega48A/PA/88A/PA/168A/PA/328/P, and will always read as zero.

- **Bit 6:0 – PCINT[14:8]: Pin Change Enable Mask 14...8**

Each PCINT[14:8]-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[14:8] is set and the PCIE1 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[14:8] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

ATmega328P: Interrupções Externas

PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
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	

- **Bit 7:0 – PCINT[23:16]: Pin Change Enable Mask 23...16**

Each PCINT[23:16]-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[23:16] is set and the PCIE2 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[23:16] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

ATmega328P: Interrupções Externas

- PCIFR – Pin Change Interrupt Flag Register
 - Sinaliza quando PCI0, PCI1 e PCI2 foram disparadas.

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **PCIF0 – Pin Change Interrupt Flag 0:** quando ocorre uma mudança de nível lógico em qualquer um dos pinos PCINT[7:0] dispara uma interrupção, PCIF0=1.
- **PCIF1 – Pin Change Interrupt Flag 1:** quando ocorre uma mudança de nível lógico em qualquer um dos pinos PCINT[14:8] dispara uma interrupção, PCIF1=1.
- **PCIF2 – Pin Change Interrupt Flag 2:** quando ocorre uma mudança de nível lógico em qualquer um dos pinos PCINT[23:16] dispara uma interrupção, PCIF2=1.

Referências

- Datasheet do processador ATmega328P
- David Russell, *Introduction to Embedded Systems using ANSI C and the Arduino Development Environment*.