INTERRUPCAO

http://www.englishandculture.com/blog/bid/97032/Business-Meeting-English-Skills-How-to-Interrupt

© Prof. Engº esp Luiz Antonio Vargas Pinto
www.vargasp.com
Rev. junho de 2025

© Prof. Vargasp

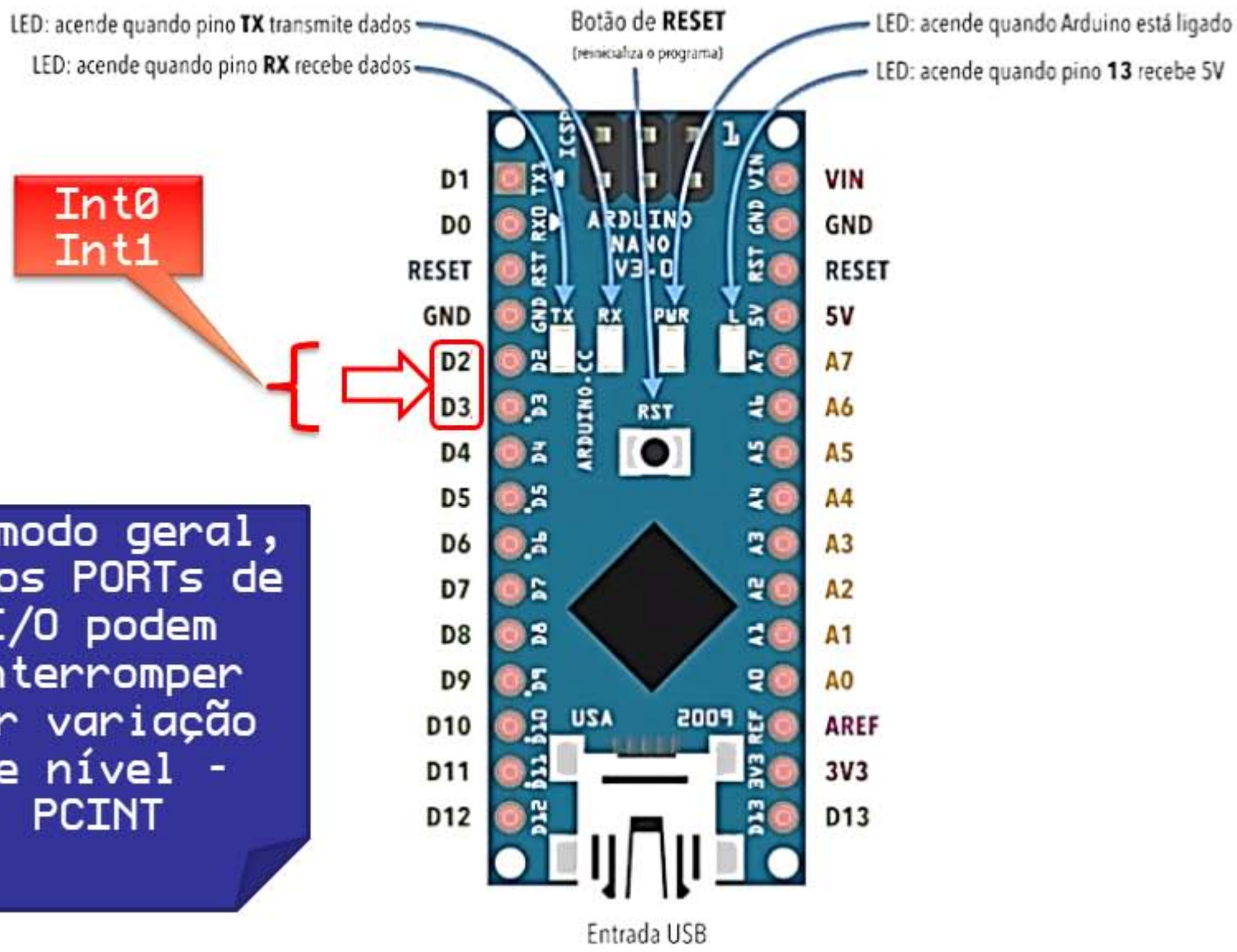
Tabela de vetores

Table 11-1. Reset and Interrupt Vectors in ATmega328P

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

© Prof. Vargasp

Interrupções



De modo geral, todos PORTs de I/O podem interromper por variação de nível - PCINT

http://eletronicaparaartistas.com.br/arduino-2-configuracao-do-arduino-nano/

© Prof. Vargasp

Interrupções nos boards

Board	Pinos digitais usados para interrupção
Uno, Nano, Mini, e 328-based	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo, e 32u4-based	0, 1, 2, 3, 7
Zero	Todos pinos digitais exceto o 4
MKR1000 Rev.1	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Due	Todos pinos digitais
101	Todos pinos digitais (somente os pinos 2, 5, 7, 8, 10, 11, 12, 13 trabalham com alteração)

Board	int.0	int.1	int.2	int.3	int.4	int.5
Uno, Ethernet, Nano	2	3				
Mega2560	2	3	21	20	19	18
32u4 based (Leonardo, Micro)	3	2	0	1	7	
Due,Zero,MKR1000,101	Nº da interrupção = número do pino					

© Prof. Vargasp

Sobre o ISR(1)

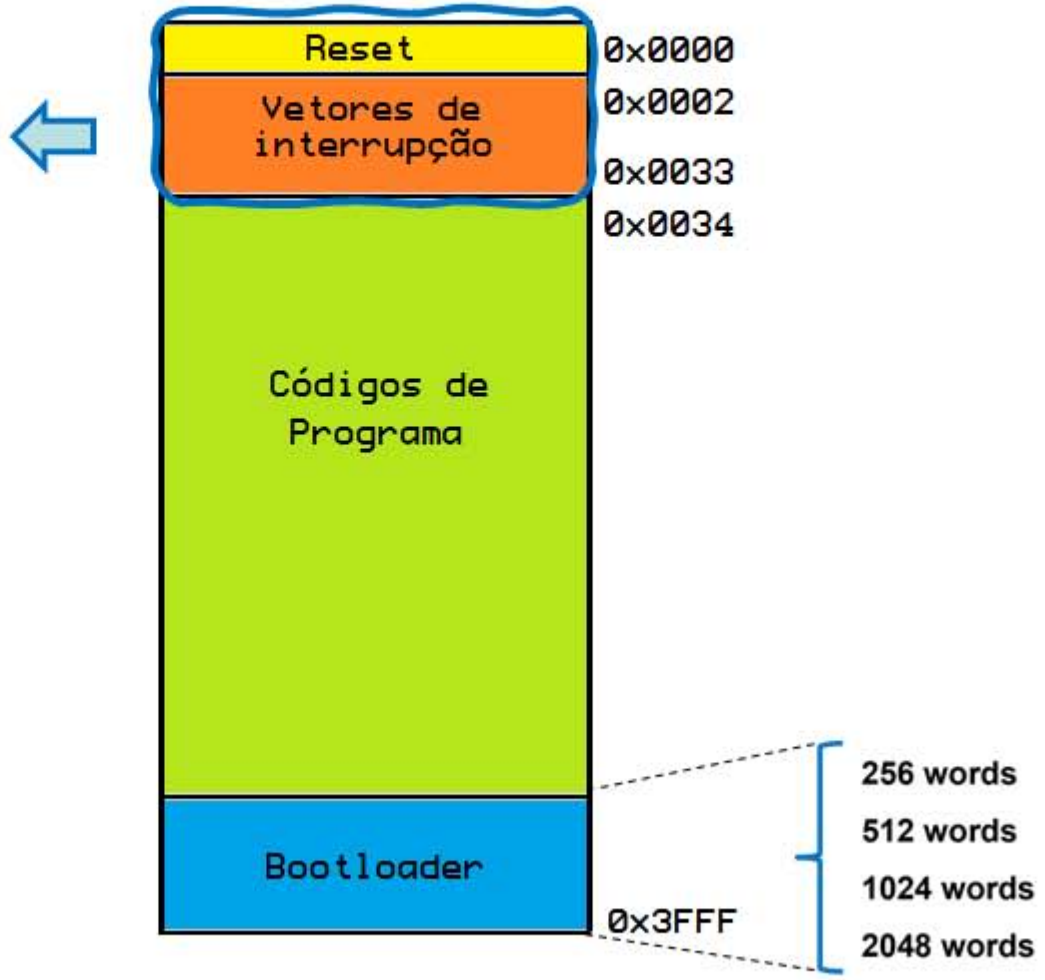
- Para que o Arduino atenda uma interrupção, devemos configurá-lo com o comando attachInterrupt().
- **Sintaxe**: attachInterrupt (INT, ISR, MODO), onde:
  - **INT**: Número da porta usada para a interrupção.
    - `0 = porta digital 2` e `1 = porta digital 3`
  - **ISR**: Nome da função que será chamada e executada quando ocorrer a interrupção;
  - **MODO:** Define o modo:
    1. LOW: Dispara a interrupção com nível `0`
    2. CHANGE: Dispara por mudança no pino
    3. RISING: Dispara na borda de subida
    4. FALLING: Dispara na borda de descida

(1) ISR = Interrupt Service Routines

© Prof. Vargasp

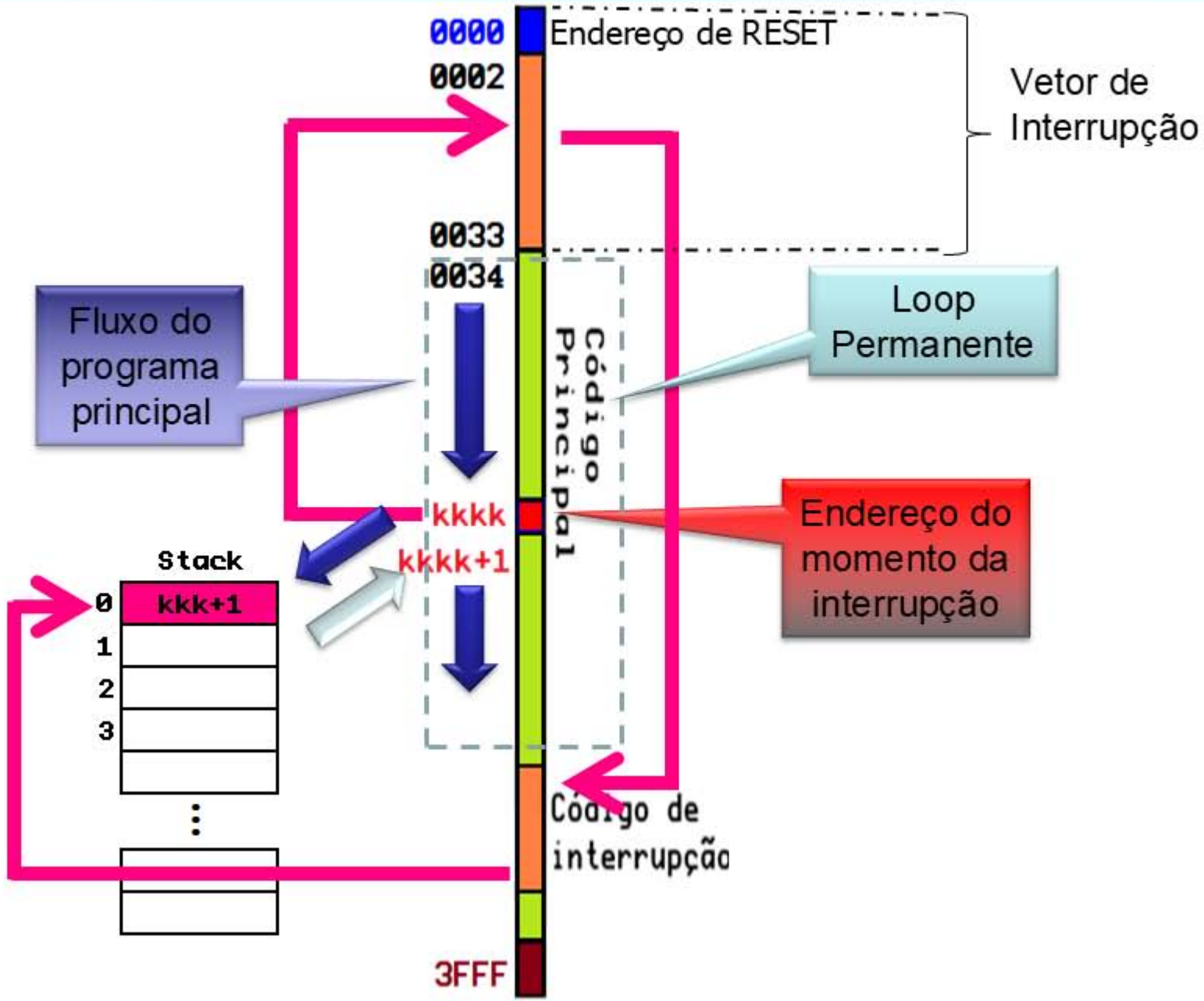
Estrutura da FlashRom

Vector No.	Program Address	Source
1	0x0000	RESET
2	0x002	INT0
3	0x0004	INT1
4	0x0006	PCINT0
5	0x0008	PCINT1
6	0x000A	PCINT2
7	0x000C	WDT
8	0x000E	TIMER2 COMPA
9	0x0010	TIMER2 COMPB
10	0x0012	TIMER2 OVF
11	0x0014	TIMER1 CAPT
12	0x0016	TIMER1 COMPA
13	0x0018	TIMER1 COMPB
14	0x001A	TIMER1 OVF
15	0x001C	TIMER0 COMPA
16	0x001E	TIMER0 COMPB
17	0x0020	TIMER0 OVF
18	0x0022	SPI, STC
19	0x0024	USART, RX
20	0x0026	USART, UDRE
21	0x0028	USART, TX
22	0x002A	ADC
23	0x002C	EE READY
24	0x002E	ANALOG COMP
25	0x0030	TWI
26	0x0032	SPM READY



© Prof. Vargasp

Processo de interrupção sem bootloader



© Prof. Vargasp

No programa

```
/*-------------------------------------------------------------------
  Programa : Interrupt.ino
  Data     : 25/02/2025
  Autor    : Prof. Vargas
  Descrição: Obter o controle On/Off do LED5 por meio de tratamento
           | de interrupção.  Foi escolhido o Int1 (digital3) do
           | Arduino Nano.
           | O Led pisca-pisca está no port de I/O digital 6
           | Os LEDs acendem com 0 e apagam com 1
           | Botão solto envia 1 ... Botão apertado envia 0
  -------------------------------------------------------------------*/

//Declaração de ports por meio de constante
const byte Lednormal = 6;   //O port de I/O digital 6 será o Lednormal
const byte Ledint = 5;

char Estado;

void Tratint()
{
  Estado = digitalRead(Ledint);
  if(Estado==0)
  {
    digitalWrite(Ledint,HIGH);
  }else
  {
    digitalWrite(Ledint,LOW);
  }
}

void setup()
{
  pinMode(Ledint, OUTPUT);
  pinMode(Lednormal, OUTPUT);
  digitalWrite(Ledint,HIGH);
  attachInterrupt(1, Tratint, FALLING); //Configura a interrupção digital3
}

void loop() //Ação normal repetitiva
{
  digitalWrite(Lednormal,LOW);
  delay(250);
  digitalWrite(Lednormal,HIGH);
  delay(250);
}
```

Ativa a interrupção

© Prof. Vargasp

Considerações sobre o Stack

- **De acordo com o datasheet do ATMEGA328pu**

"During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The stack pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture."

- **O STACK** está em SRAM e portanto é preciso cuidado com a quantidade de interrupções que usaremos

© Prof. Vargasp

Até a próxima !