

LDH ATMEGA328P INTERRUPCIONES

Manuel J. Bellido Díaz
Germán Cano Quiveu

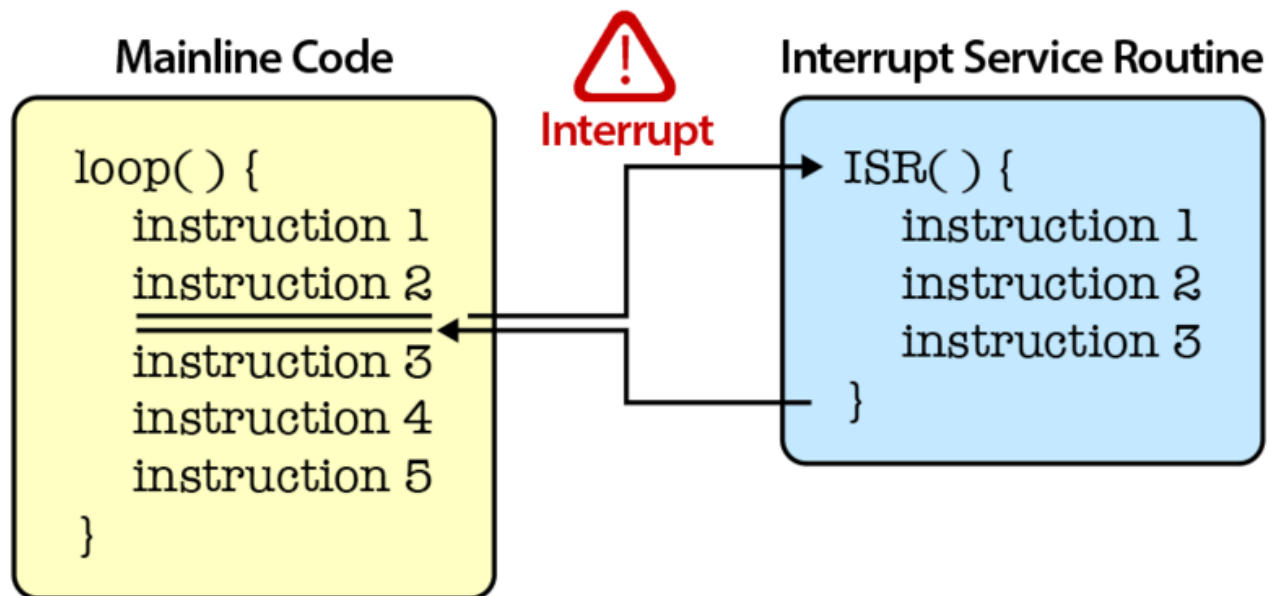


REFERENCIAS

- http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- <https://www.arduino.cc/reference/en/language/functions/interrupts/interrupts/>

INTERRUPCIONES

- Las interrupciones se ejecutan al activarse determinada señal, interrumpe el código principal y ejecuta una subrutina.



INTERRUPCIONES

INTERRUPCIONES

- Esto es especialmente útil para realizar tareas en segundo plano con determinados sensores los cuales se activan cuando se cumple cierto requisito, o es especialmente importante cuando se une con el diseño de bajo consumo.

INTERRUPCIONES (I)

- En el ATMEGA328P cuando una interrupcion ocurre se deshabilitan por defecto el resto de subrutinas.
 - Este comportamiento se podria cambiar para anidar interrupciones
 - Ejemplo: la funcion `delay()` llama a una ISR (`TIMER0_OVF_vect`) por lo que esta función dentro de otra ISR deshabilitará la ejecución de la misma.
 - Sin embargo la funcion `delayMicroseconds()` no usa ISR

INTERRUPCIONES (III)

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	0x0002	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

Arduino References

- Existen algunas interrupciones, las mas comunes, dentro de las librerias de arduino
 - AttachInterrupt()
 - INT0 (pin2) e INT1 (pin3), interrupciones externas

INTERRUPCION TIMER

```
#include <avr/io.h>
#include <avr/interrupt.h>

void activar_timer1(){
    TCCR1A = 0;
    TCCR1B = 0;
    //habilitar timer1 overflow interrupt
    TIMSK1 = (1<<TOIE1);
    //habilitar timer con frecuencia clk_IO/1
    TCCR1B |= ((0<<CS12) | (0<<CS11) | (1<<CS10));
}

void cancelar_timer1(){
    TIMSK1 &= (0<<TOIE1);
}

ISR(TIMER1_OVF_vect){
    //se ejecuta cada 0xFFFF+1 ciclos de reloj
    // se ejecuta cada 65336/Fq = X s
}
```

ANALOG COMP

- Compara los voltajes de los pines AIN0 y AIN1
 - AIN0 es el Pin Digital 6 en el Arduino
 - AIN1 es el Pin Digital 7 en el Arduino
 - Cuando $AIN0 > AIN1 \Rightarrow ACO = 1$
 - Las interrupciones variaran segun ACO
 - RISE
 - FALL
 - TOGGLE

INTERRUPCION COMPARADOR

```
//LIBRERIAS NECESARIAS PARA INTERRUPCIONES EN MODO AVR
```

```
#include <avr/io.h>
```

```
#include <avr/interrupt.h>
```

```
//BLOQUE PARA ACTIVAR EL COMPARADOR
```

```
void activar_comparador(){
```

```
    //habilitar interrupt
```

```
    ACSR = (1<<ACIE);
```

```
    //habilitar modo, ejecutar ISR solo cuando AIN0 sea mayor que AIN1 (RISING)
```

```
    ACSR |= ((1<<ACIS1) | (1<<ACIS0));
```

```
}
```

```
//
```

```
void cancelar_timer1(){
```

```
    ACSR &= (0<<ACIE);
```

```
}
```

```
ISR(ANALOG_COMP_vect){
```

```
    .....
```

```
}
```