

PRACTICA N° 4

INTERRUPCIONES EXTERNAS

DE UN MICROCONTROLADOR ATMEGA16

OBJETIVOS.

HABILITAR Y USAR LAS INTERRUPCIONES EXTERNAS DEL MICROCONTROLADOR

INTRODUCCION

Un microcontrolador de 8 bits cuenta con al menos una interrupción externa, en el caso concreto de un ATMEGA328, este tiene 2 interrupciones externas, la interrupción externa 0 (INT0) se encuentra en el pin 4 (PD2) y la interrupción externa 1 (INT1) se tiene en el pin 5 (PD3). Estas interrupciones contienen cuatro formas de disparo para la INT0 e INT1, los registros relacionados con las interrupciones externas son los siguientes:

BIT							
7	6	5	4	3	2	1	0
-	-	-	-	ISC11	ISC10	ISC01	ISC00

Registro de control de la Interrupciones Externa A (EICRA)

Los bits 0, 1, 2 y 3 de este registro se usan para elegir alguno de los cuatro formas de disparo de las interrupciones INT0 e INT1, estos se describen en la tabla 4.1, los bits para INT0 son ISC01 (bit 1) e ISC00 (bit 0) y para INT1 son ISC11 (bit 3) e ISC10 (bit 2).

Tabla 4.1 formas de disparo de las interrupciones INT0 e INT1.

ISC11 / ISC01	ISC10 / ISC00	DESCRIPCION
0	0	Se genera una llama a interrupción (INT0 / INT1) cuando hay un nivel bajo.
0	1	Se genera una llamada a interrupción (INT0 / INT1) con cualquier cambio de estado lógico.
1	0	Se genera una llamada a interrupción (INT0 / INT1) con un flanco de bajada.
1	1	Se genera una llamada a interrupción (INT0 / INT1) con un flanco de subida.

Para activar cada una de las interrupciones se emplea el Registro de Mascara de Interrupción Externa (EIMSK), cada una de las interrupciones se activan cuando el usuario pone a uno el bit que habilita la interrupción. Por ejemplo si se desea habilitar solo la interrupción cero el bit que se deberá habilitar será el bit 0 de este registro.

BIT							
7	6	5	4	3	2	1	0
-	-	-	-	-	-	INT1	INT0

Registro de Mascara de Interrupción Externa (EIMSK)

Cuando se genera la interrupción al ser originada por un evento externo, la bandera de dicha interrupción se pondrá a uno y volverá a cero una vez que se haya atendido la interrupción. Estas banderas se encuentran en el Registro de Banderas de Interrupciones Externas (EIFR)

BIT							
7	6	5	4	3	2	1	0
-	-	-	-	-	-	INTF1	INTF0

Registro de Banderas de Interrupción Externa (EIFR)

DESARROLLO

Planteamiento del problema

Se requiere escribir un programa que realice normalmente una rutina de corrimiento y cuando llegue un evento externo vía INT0 se despliegue 5 ocasiones la rutina de desplegar los datos 0xAA, 0x55 y después borrar el puerto con sus respectivos retardos de un segundo entre cada valor. La INT0 se dispara por flanco de subida.

Procedimiento

CODIGO EN LENGUAJE C.

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
int i=0, j=1, bandera=0, a=0x55, b=0xAA, c= 0x00;
ISR (INT0_vect) {
    for(i=0;i<5;i++){
        PORTB = a;
        _delay_ms(1000);
        PORTB = b;
        _delay_ms(1000);
        PORTB = c;
        _delay_ms(1000);
    }
}

int main(void){
    cli();
    DDRB = 0xFF;
    DDRD = 0x00;
    EIMSK |= (1<<INT0);
    EICRA |= (1<<ISC01)|(1<<ISC00);
    sei();
    while(1)
    {
        if (bandera==0) {
            _delay_ms(500);
            j=(j*2);
            if(j==0x40){
                bandera=1;}
        }

        if (bandera==1){
            j=(j/2);
            _delay_ms(500);
            if(j==1){
                bandera=0;}
        }
        PORTB=j;
    }
    return(0);
}
```

INTERUPCION POR CAMBIO DE PIN.

En las interrupciones externas que vimos anteriormente solo tiene dos y cada una con un solo pin que las activa, en el caso de la interrupción por cambio de pin o PCINT tenemos un grupo de pines que pueden causar una interrupción, para el caso específico del ATMEGA328 el primer grupo que puede causar la interrupción PCINT0 va del PCINT0 o PB0 a PCINT7 o PB7 como se muestra en la figura 4.1, uno puede escoger dentro de este grupo que pin puede activar la interrupción con un cambio lógico, a diferencia de las interrupciones externas esta no se puede seleccionar el tipo de señal que active la interrupción, ya que se activa solamente con cualquier cambio lógico. El siguiente grupo que activa la interrupción PCINT1 es el que lo conforma PCINT8 o PC0 a PCINT14 o PC6 y por último la interrupción PCINT2 la integran los pines PCINT16 al PCINT23, como se observa cada grupo corresponde a cada puerto del microcontrolador.

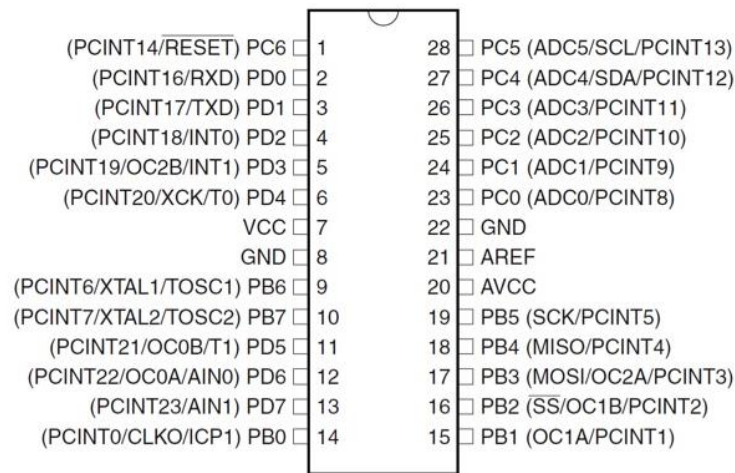


Fig 4.1 Pines del microcontrolador ATMEGA 328.

Los registros involucrados en esta interrupción son PCICR en el cual se habilita las interrupciones, con el bit 0 o PCIE0 se habilita con un uno la interrupción PCINT0, con el bit 1 o PCIE1 se habilita con un uno la interrupción PCINT1, y por último con el bit 2 o PCIE2 se habilita con un uno la interrupción PCINT2,

BIT							
7	6	5	4	3	2	1	0
-	-	-	-	-	PCIE2	PCIE1	PCIE0

Registro de Control de la Interrupción de Cambio de Pin (PCICR)

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLOGIAS AVANZADAS
INSTRUCTIVO DE PRACTICAS DE LABORATORIO DE MICROCONTROLADORES AVANZADOS

Cuando hay un cambio lógico que produzca una interrupción, se pondrá a uno la bandera de la interrupción que fue activada.

BIT							
7	6	5	4	3	2	1	0
-	-	-	-	-	PCIF2	PCIF1	PCIF0

Registro de Banderas de la Interrupción de Cambio de Pin (PCIFR)

Los siguientes tres registros sirven para designar el pin o los pines que ejecutan la interrupción, con un uno se dará de alta para ejecutar la interrupción, los registros se muestran a continuación.

BIT							
7	6	5	4	3	2	1	0
PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0

Registro para Enmascarar el Cambio de Pin (PCMSK0)

BIT							
7	6	5	4	3	2	1	0
-	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8

Registro para Enmascarar el Cambio de Pin (PCMSK1)

BIT							
7	6	5	4	3	2	1	0
PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16

Registro para Enmascarar el Cambio de Pin (PCMSK2)

Por último mostramos el siguiente programa empleando la interrupción por cambio de pin.

```
define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
int i=0, bandera=0, j=1, a=0x55, b=0xAA, c= 0x00;
ISR (PCINT1_vect) {
    for(i=0;i<5;i++){
        PORTB = a;
        _delay_ms(1000);
        PORTB = b;
        _delay_ms(1000);
        PORTB = c;
        _delay_ms(1000);
    }
}

int main(void){
    cli();
    DDRB = 0xFF;
    PCICR |= (1<<PCIE1);
    PCMSK1 |= (1<<PCINT8)|(1<<PCINT11);
    sei();
    while(1)
    {
        if (bandera==0) {
            _delay_ms(500);
            j=(j*2);
            if(j==0x40){
                bandera=1;}
        }

        if (bandera==1){
            j=(j/2);
            _delay_ms(500);
            if(j==1){
                bandera=0;}
        }
        PORTB=j;
    }
    return(0);
}
```