

7.1 COMPARADOR ANALÓGICO.

Un comparador analógico es un dispositivo que se utiliza cuando se requiere saber el cambio de nivel de una señal con respecto a una referencia, y visto desde un punto de vista simplificado podemos imaginarlo como un amplificador operacional en lazo abierto, y puede ser utilizado para detectar cuando algún nivel de voltaje rebasa un nivel peligroso para nuestra aplicación, o cuando queremos saber cual es el comportamiento de dos señales la una con respecto a la otra.

El principio de operación es, suponiendo que se tienen dos señales A y B, dadas en términos de niveles de voltaje analógicos la señal de control o de salida del comparador se disparará si:

- La señal A es más grande que la señal B.
- La señal B es más grande que la señal A.
- Hubo una basculación de las señales A y B (es decir un cambio de nivel alto a nivel bajo con respecto a la otra señal que funciona como referencia).

7.2 CONFIGURACIÓN DEL COMPARADOR ANALÓGICO.

El comparador analógico del microcontrolador AVR ATmega48, compara los valores de entrada en su pin positivo (AIN0) y negativo (AIN1). Cuando el voltaje del pin positivo es mayor que el del pin negativo, la salida de comparador analógico (ACO) se activa.

7.2.1 Registro de funciones especiales de entradas y salidas, SFIOR.

# de bit	7	6	5	4	3	2	1	0	
	-	-	-	-	ACME	PUD	PSR2	PSR10	SFIOR
Read/write	R	R	R	R	R/W	R/W	R/W	R/W	
Valor inicial	0	0	0	0	0	0	0	0	

Figura 7.1 Registro SFIOR

Bit 3 – ACME: Habilitación de multiplexor para comparador analógico. Cuando se activa este bit y el convertidor analógico-digital (ADC) está deshabilitado (ADEN=0 en ADCSRA) el ADC selecciona la entrada negativa del comparador analógico.

7.2.2 Registro de estatus y control del comparador analógico, ACSR.

# de bit	7	6	5	4	3	2	1	0	
	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Valor inicial	0	0	N/A	0	0	0	0	0	

Figura 7.2 Registro ACSR

Bit 7 – ACD: Deshabilita comparador analógico. Un uno en este bit, apaga el comparador analógico.

Bit 6 – ACBG: Selección de Bandgap, activar este bit, reemplaza la entrada positiva del comparador por el voltaje de bandgap (aprox. 1.23V @ 5V de V_{CC}).

Bit 5 – ACO: Salida de comparador analógico.

Bit 4 – ACI: Bandera de interrupción de comparador analógico. Este bit se activa por el comportamiento de interrupción definido por los bits ACIS1 y ACIS0.

Bit 3 – ACIE: Habilitación de interrupción de comparador analógico. Escribir un uno a este bit, activa la interrupción por comparador analógico.

Bit 2 – ACIC: Habilitación de evento de captura por comparador analógico. Escribir un uno a este bit, activa la función del timer/counter1 para que esta sea activada por el comparador analógico.

Para que se active el evento de captura, necesita estar en uno el bit TICIE del registro TIMSK.

Bit 1:0 – ACIS1:ACIS0: Selección de modo de interrupción. Determinan que evento activará la interrupción del comparador analógico. Las diferentes combinaciones se muestran en la tabla 7.1.

ACIS1	ACIS0	Modo de Interrupción.
0	0	Interrupción en un cambio en el bit de salida.
0	1	Reservado
1	0	Interrupción en un flanco de bajada en el bit de salida.
1	1	Interrupción en un flanco de subida en el bit de salida.

Tabla 7.1 Combinaciones de ACIS1:ACIS0

7.3 PRÁCTICA DE USO DEL COMPARADOR ANALÓGICO.

7.3.1 Objetivo.

Conocer el funcionamiento y los registros que intervienen en la configuración del comparador analógico.

7.3.2 Desarrollo.

Arme el circuito de la figura 7.3.

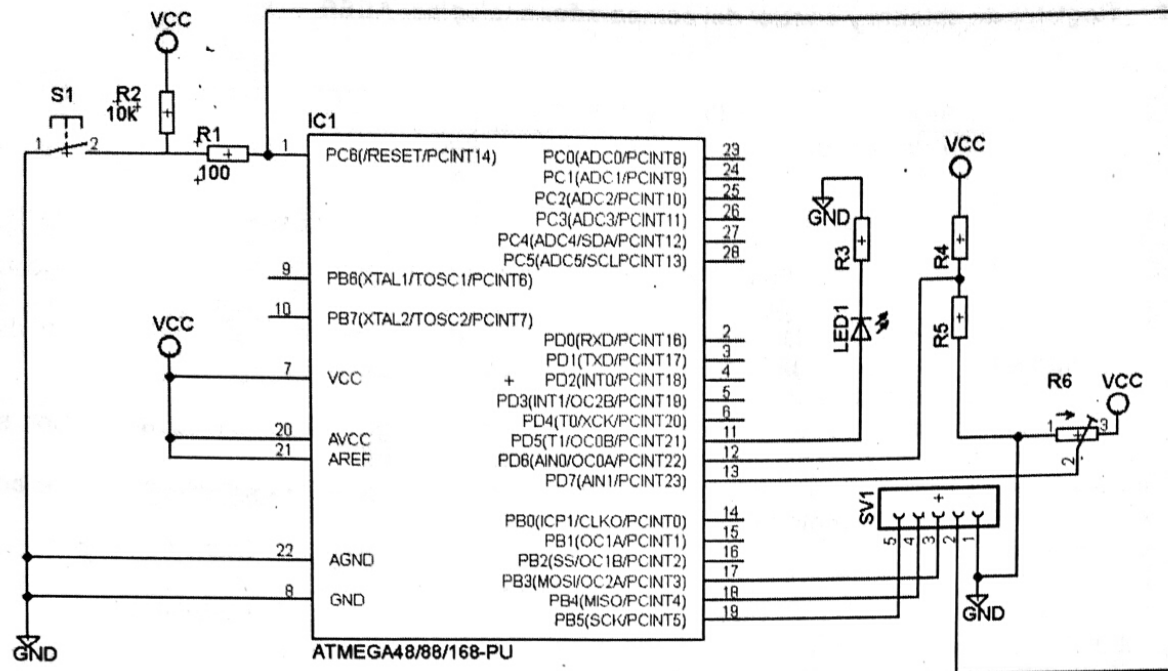


Figura 7.3 Circuito para la practica de comparador analogico.

7.3.3 Código.

```
//----- INICIO DE PROGRAMA

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

void init_comparador(void){
    DDRD=0b00111111;           // PD6 y PD7 como entradas
    ACSR=_BV(ACIE);            // Interrupcion en cambio AIN0 vs. AIN1
    sei();
}

ISR(ANA_COMP_vect){           // Interrupcion por comparador analogico
    if(PORTD==0x20)            // Si el LED esta prendido, apagarlo
        PORTD=0;
    else if(PORTD==0x00)       // Si el LED esta apagado, prenderlo
        PORTD=_BV(PD5);
}

int main(void){
    init_comparador();
    while(1);
}

//----- FIN DE PROGRAMA
```