PRACTICA # 12 "USO DEL CONVERTIDOR ANALOGICO DIGITAL"

ADMUX=0X00;

CLR R16

OUT ADMUX,R16

ADCSRA=\_BV(ADEN)|\_BV(ADFR)|\_BV(ADIE)|\_BV(ADPS2)\_BV(ADPS0)|\_BV(ADSC);

ATMEGA8535 ADCSRA :ADEN,ADSC,ADATE,ADIF,ADIE,ADPS2,ADPS1,ADPS0

RESET: 0 0 0 0 0 0 0 0

MODO CARRERA LIBRE: 1 1 1 0 1 1 0 1 =$ED

ADEN: HABILITACION DEL ADC

ADSC: INICIO DE CONVERSION

ADATE: CARRERA LIBRE

ADIF: BANDERA DE INTERRUPCION POR CONVERSION

ADIE:HABILITACION DE INTERRUPCION POR ADC

ADPS2-0: PRE-ESCALA DEL RELOJ DEL ADC

POR LO TANTO PARA TENER HABILITADO EL MODO CARRERA LIBRE Y GENERAR UNA INTERRUPCION POR FIN DE CONVERSION , SE DEBE CARGAR AL REGISTRO ADCSRA CON EL VALOR:

ADCSRA= 0b11101101

PARA TENER HABILITADA LA INTERRUPCION POR FIN DE CONVERSION, EL MODO CARRERA LIBRE, USAR LA ENTRADA ADC0 CON UN FACTOR DE DIVISION DE 4 Y USAR ALINEACION A LA IZQUIERDA PARA PODER USAR CONVERSIONES DEL TAMAÑO DE UN BYTE :

Practica #12

1. Armar el circuito del teclado analógico, comprobar de forma práctica cada uno de los voltajes al pulsar cada botón y calcular teóricamente cada uno de estos voltajes, comprobar que deben ser muy similares ambos valores.
2. Escribir un programa en ensamblador para poder leer el valor a travez del ADC y mostrarlo de forma binaria en 2 puertos, comprobar teóricamente estos valores con la formula indicada en el manual del microcontrolador, tomar nota de estos valores ya que son los que se usaran para el circuito final
3. El circuito final consiste en conectar el teclado analógico y 6 displays multiplexados, y deberá escribirse un programa que muestre los dígitos decimales (0-9) correspondientes a los botones pulsados.

**Nota: para considerar aprobada esta práctica, se deben cubrir los 3 incisos, y el teclado debe ser el descrito (ANALOGICO)**

PROGRAMA PARA EL PUNTO 2

EN ESTE PROGRAMA SE DEJO ADMUX=$00 SI SE QUIERE TENER ALINEACION A LA DERECHA SE DEBERA AGREGAR: LDI R16,$20 Y OUT ADMUX,R16 (ADLAR=1)

.INCLUDE"m8535def.inc"

rjmp Start

.ORG $0E

RJMP CONV

Start:

LDI R16,LOW(RAMEND)

OUT SPL,R16

LDI R16,HIGH(RAMEND)

OUT SPH,R16

SER R16

OUT DDRD,R16

OUT DDRB,R16

LDI R16,$ED

OUT ADCSRA,R16

SEI

Loop:

OUT PORTD,R17

OUT PORTB,R16

rjmp Loop

CONV:

IN R16,ADCL

IN R17,ADCH

RETI