INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

PRACTICA 11

INTRODUCCIÓN A LOS MICROCONTROLADORES

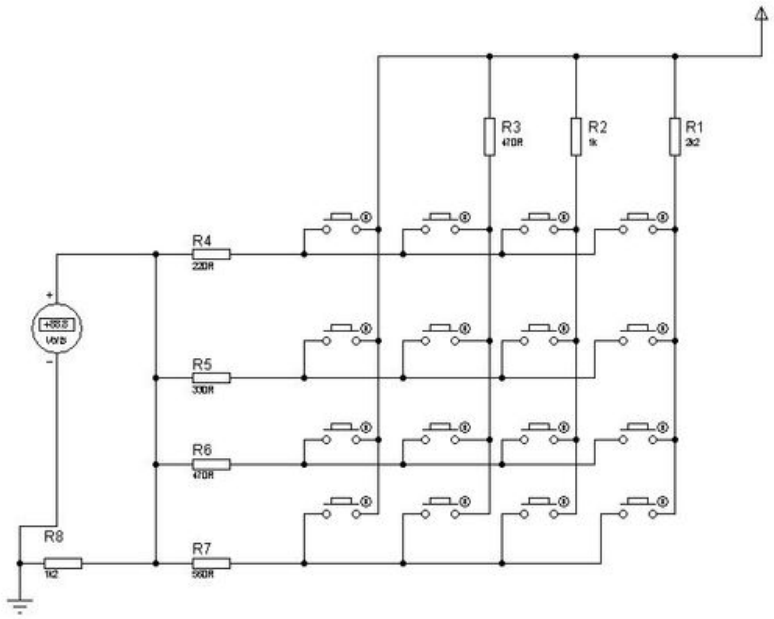
INTEGRANTES:
HERRERA HERNÁNDEZ ANGEL SALVADOR
LÓPEZ ROJAS GUILLERMO EDER
MATA CORTÉS VALERIA
GRUPO: 3CM6
PROFESOR: PÉREZ PÉREZ JOSÉ JUAN

1 | DESARROLLO

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 hexadecimales (0-F) correspondientes a los botones pulsados.



2 | CÓDIGO

```
.include "m8535def.inc"
.def ADH = r16
.def d = r17
.def col = r18
.def aux2 = r19

rjmp main
.org $08
rjmp scroll
rjmp barre

.org $00E
rjmp conv

main:
    ldi d,low (RAMEND)
    out spl, d

    ldi d, high (RAMEND)
    out sph, d

    ldi d, $20

    out ADMUX, d
    ldi d, $ED
    out ADCSRA, d
    ser d
    out ddrb, d
    out ddrd, d
    sei
    out ddrc,d
    ser d
    out porta,d

    ldi d,$00;
    clr r20
    mov r5,d
    mov r4,d
    mov r3,d
    mov r2,d
    mov r1,d
    mov r0,d
```

```
    ldi d,1
    out TCCR0, d
    ldi d,2
    out TCCR1B, d
    ldi d,5
    out TIMSK, d
    sei
    clr zh
    ldi zl,5
    ldi col,$20
    ldi r27, 0
    clr r20


uno:
    out portd, ADH
    out portb, d
    in r25, pind

    si2:
    out portc,col
    ld d,z
    out portb,d
    rjmp uno

barre:
    ldi aux2,1
    out TCNT0,aux2
    out portc,zh
    lsr col
    brcs dos
    dec zl


reti
```

```
dos:
    ldi col,$20
    ldi zl, 5
reti

scroll:


    cpi r25,$50;
    brbc 1,tr1S

    ldi r27,$0C
    rcall mover


    tr1S:
    cpi r25,$74;
    brbc 1,tr2S

    ldi r27,$B6
    rcall mover

    tr2S:
    cpi r25,$88;
    brbc 1,tr3S

    ldi r27,$9E
    rcall mover


    tr3S:
    cpi r25,$4E;
    brbc 1,tr4S

    ldi r27,$CC
    rcall mover
```

```
tr4S:
cpi r25,$6E;
brbc 1,tr5S

ldi r27,$DA
rcall mover


tr5S:
cpi r25,$80;
brbc 1,tr6S

ldi r27,$FA
rcall mover


tr6S:
cpi r25,$4B;
brbc 1,tr7S

ldi r27,$0E
rcall mover


tr7S:
cpi r25,$68;
brbc 1,tr8S

ldi r27,$FE
rcall mover


tr8S:
cpi r25,$78;
brbc 1,tr9S

ldi r27,$CE
rcall mover
```

```
tr9S:
cpi r25,$65;
brbc 1,trAS

ldi r27,$7E
rcall mover


trAS:
cpi r25,$D2;
brbc 1,trBS

ldi r27,$EE
rcall mover


trBS:
cpi r25,$C0;
brbc 1,trCS

ldi r27,$F8
rcall mover


trCS:
cpi r25,$AE;
brbc 1,trDS

ldi r27,$72
rcall mover


trDS:
cpi r25,$A4;
brbc 1,trES

ldi r27,$BC
rcall mover
```

```
    trES:
    cpi r25,$49;
    brbc 1,trFS

    ldi r27,$F2
    rcall mover


    trFS:
    cpi r25,$73;
    brbc 1,trOTROS

    ldi r27,$E2
    rcall mover


    trOTROS:


reti


conv:
    in ADH, ADCH
reti


mover:

    mov r0, r1
    mov r1, r2
    mov r2, r3
    mov r3, r4
    mov r4, r5
    mov r5, r27


ret
```

3 | CONCLUSIONES

El principio de funcionamiento es sencillo y es similar a lo que haríamos para multiplexar leds o dipslays de 7 segmentos. El programa configura el puerto B del PIC de la siguiente forma: RB4 a RB7 funcionan como salidas y la otra mitad (RB0-RB3) como entradas. Las filas (horizontal) del teclado matricial se conectan a los bits más significativos que funcionan como salidas, mientras que las columnas (vertical) se conectan a los bits menos significativos del puerto que funcionan como entradas con resistencias pull-down. Cualquier tecla que se oprima en una columna causará que uno de los bits menos significativos del puerto (RB0 – RB3) cambie de un estado lógico bajo a un estado alto.

El microcontrolador escanea en forma sucesiva los pines de salida, mientras lee las entradas en la parte baja del puerto, de manera que puede detectar que teclas están oprimidas en cada fila. Ahora solo falta escribir nuestro codigo en C que implemente los procedimientos mencionados anteriormente y nos devuelva un valor de la tecla oprimida, por ejemplo, mediante un número binario.