| Resultado de imagen para logo ipn | Instituto Politécnico Nacional  Escuela Superior de Cómputo | Resultado de imagen para logo escom |
| --- | --- | --- |

**Ejercicio : Clientes Eeprom**

**Microcontroladores**

Grupo: 3CM16

Alumnos:

Cazares Martínez Maximiliano

Lozano Rivera Oscar

Ramos Nieves Adrian

Profesor.

Pérez Pérez José Juan

**Ejercicio**

Para el circuito usado en el ejercicio de contador de clientes, agregar 2 displays bcd 7 seg. en el puerto D, en estos displays se mostrará la cuenta de clientes ganadores. La primera vez que funcione el circuito mostrará “00” y por cada cliente ganador se irá incrementado en forma decimal hasta “99”. Cada vez que se incrementa esta cuenta se debe guardar en la EEPROM este valor, si se apaga el sistema o se pulsa “RESET” debe iniciar la cuenta con el último valor guardado, si se pulsa INT2 la cuenta deberá iniciar se a “00”.

**Código de AVR Studio**

.include"m8535def.inc"

.def aux = r16

.def msk = r17

.def ini = r18

.def tc1h = r19

.def tc1l = r20

.def auc = r21

.def dirh = r22

.def dirl = r23

.def dato = r24

.def datodecimal = r25

.def diezhex = r26

.macro eeprw

ldi dirh,@0

ldi dirl,@1

rcall EEPROM\_write

.endm

.macro eeprr

ldi dirh,@0

ldi dirl,@1

rcall EEPROM\_read

.endm

.macro convdecimal

clr datodecimal

clr aux

add aux, dato

comprobar:

cpi aux,10

brlo terminar

restar:

add datodecimal, diezhex

subi aux, 10

rjmp comprobar

terminar:

add datodecimal, aux

.endm

reset:

rjmp main

.org $004

rjmp onda

.org $008

rjmp tmpo

rjmp cliente

.org $012

rjmp interrupciondos

main:

ldi aux,low(ramend)

out spl,aux

ldi aux,high(ramend)

out sph,aux

ser aux

out ddra,aux

out ddrc,aux

out ddrd,aux

out portb,aux

ldi aux,6

out tccr0,aux

ldi aux,0b01000101

out timsk,aux

ldi aux,0b01000000

out mcucsr,aux

ldi aux,0b00100000

out gicr,aux

sei

ldi aux,250

out tcnt0,aux

ldi msk,1

ldi ini,256-141

ldi tc1h,$B3

ldi tc1l,$B5

out tcnt1h,tc1h

out tcnt1l,tc1l

ldi diezhex, $10

eeprr $00, $00

convdecimal

cpi dato,$64 ;100 en hexadecimal

brne nada

ldi dato,$00

convdecimal

eeprw $00,$00

nada:

in auc,tcnt0

clr aux

sub aux,auc

out portc,aux

out portd,datodecimal

rjmp nada

onda:

out tcnt2,ini

in aux,pina

eor aux,msk

out porta,aux

reti

tmpo:

ldi aux,0

out tccr2,aux

out tccr1b,aux

ldi aux,250

out tcnt0,aux

reti

cliente:

ldi aux,2

out tccr2,aux

ldi aux,4

out tccr1b,aux

out tcnt1h,tc1h

out tcnt1l,tc1l

inc dato

cpi dato,$64 ;100 en hexadecimal

brne escribirdato

ldi dato,$00

escribirdato:

convdecimal

eeprw $00,$00

reti

interrupciondos:

ldi dato,$00

convdecimal

eeprw $00,$00

reti

EEPROM\_write:

; Wait for completion of previous write

sbic EECR,EEWE

rjmp EEPROM\_write

; Set up address (r18:r17) in address register

out EEARH, dirh

out EEARL, dirl

; Write data (r16) to Data Register

out EEDR,dato

; Write logical one to EEMWE

sbi EECR,EEMWE

; Start eeprom write by setting EEWE

sbi EECR,EEWE

ret

EEPROM\_read:

; Wait for completion of previous write

sbic EECR,EEWE

rjmp EEPROM\_read

; Set up address (r18:r17) in Address Register

out EEARH, dirh

out EEARL, dirl

; Start eeprom read by writing EERE

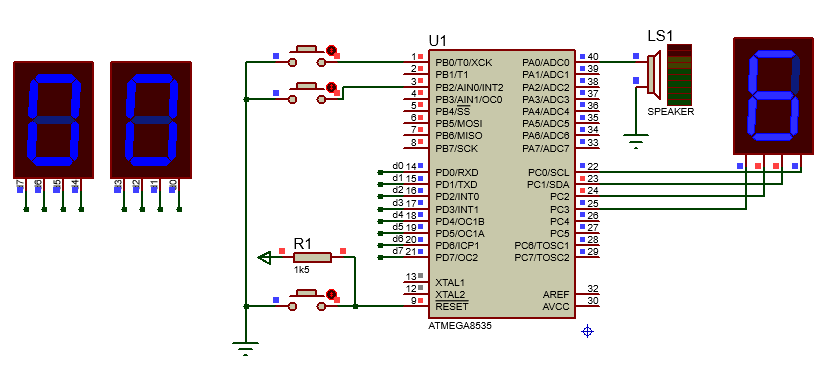
sbi EECR,EERE

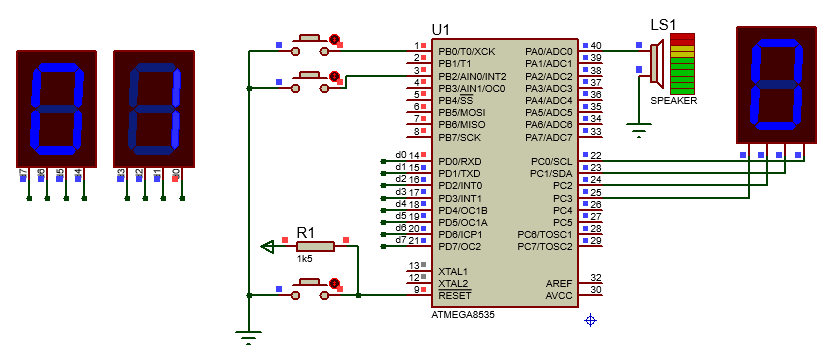
; Read data from Data Register

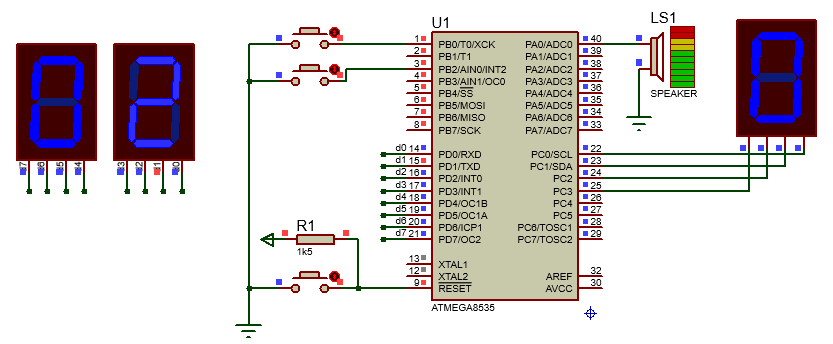
in dato,EEDR

ret

**Pruebas en Proteus**

****

****

****

**Conclusiones**

**Maximiliano Cazares Martínez:**

Los microcontroladores parecidos al ATMEGA8535 son de mucha utilidad para resolver problemas sencillos como el anteriormente planteado, ya que no requieren de mucho procesamiento. El uso de la EEPROM de un microprocesador hace que sus aplicaciones sean cada vez mayores. El desarrollo de esta práctica fue un poco complicado debido al uso de la EEPROM, sin embargo, al final pudimos terminarla.

**Oscar Lozano Rivera:**

La modificación de esta práctica respecto a la anterior nos permitió interactuar con un elemento sumamente importante, la memoria no volátil o EEPROM, la cual nos permite almacenar datos sin que se necesite energía para que la información persista. Con esta aplicación del microcontrolador podemos encender y apagar el sistema con el que interactúe el microcontrolador y retomar los procesos desde el punto donde se detuvo la energía.

**Adrian Ramos Nieves:**

En esta práctica modificamos el ejercicio de la práctica anterior, guardando la cantidad de clientes ganadores directamente en la EEPROM, que es la memoria no volátil del microprocesador para que la información almacenada no se pierda cuando se apague, esta función nos muestra que podemos guardar distinta información y sea persistente. Después se mostró en otros dos displays de 7 segmentos la cantidad de ganadores, que se pueden almacenar hasta 99.