



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

PRÁCTICAS 2DO DEPARTAMENTAL:

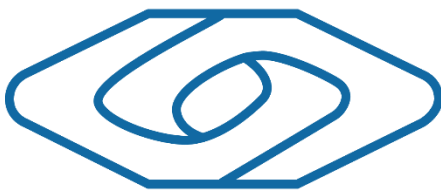
Introducción a los Microcontroladores

Grupo: 3CM8

Alumnos:

Cruces Luna Oscar Francisco

Pérez Hernández Julio



ESCOM

Contenido

Practica 1: Contador de 0 a 99.....	3
Introducción.....	3
Material	3
Desarrollo	3
Conclusiones.....	5
Bibliografía.....	5
Practica 2: Hola multiplexado.....	5
Introducción.....	5
Material	5
Desarrollo	6
Conclusiones.....	6
Practica 3: Hola scroll.....	7
Introducción.....	7
Material	7
Desarrollo	7
Conclusiones.....	11

Practica 1: Contador de 0 a 99

Introducción

En esta práctica realizamos un circuito en el cual se muestra mediante dos display un conteo del 0 al 99, el conteo es automático usando un delay.

Material

- 1 microcontrolador ATMEGA8535
- Fuente de voltaje
- Protoboard
- Cable
- Leds
- Resistencias

Desarrollo

```
;---ESCUOLA SUPERIOR DE COMPUTO---
```

```
;-----CRUCES LUNA OSCAR FRNACISCO
```

```
;-----PEREZ HERNANDEZ JULIO
```

```
.include "m8535def.inc"
```

```
.def datb = r17
```

```
ldi r16, low(RAMEND)
```

```
out SPL,r16
```

```
ldi r16, high(RAMEND)
```

```
out SPH,r16
```

```
ser r16
```

```
out ddra,r16
```

```
out ddrc,r16
```

```
out ddrd,r16
```

```
out portb,r16
```

```
leer:
```

```
in datb,pinb
```

```
mov r19,datb
```

```
rcall modulo
```

```
out portd,r19
```

```
out portc,r20
```

```
rjmp leer
```

```
modulo:
```

```
ldi r20,0
```

```
restar:
```

```
    cpi r19,$a
```

```
    brlo aaaa
```

```
    inc r20
```

```
    subi r19,$a
```

```

        rjmp restar
aaaa:
        rcall decodificar
        mov r21,r19
        mov r19,r20
        rcall decodificar
        mov r20,r19
        mov r19,r21
ret

```

```

decodificar:
        cpi r19,$0
        breq dispzero
        cpi r19,$1
        breq dispuno
        cpi r19,$2
        breq dispdos
        cpi r19,$3
        breq disptres
        cpi r19,$4
        breq dispquatro
        cpi r19,$5
        breq dispquinco
        cpi r19,$6
        breq dispseis
        cpi r19,$7
        breq dispseven
        cpi r19,$8
        breq dispocho
        cpi r19,$9
        breq dispnueve

dispzero:
        ldi r19,$3f
        rjmp endDecodificar
dispuno:
        ldi r19,$06
        rjmp endDecodificar
dispdos:
        ldi r19,$5b
        rjmp endDecodificar
disptres:
        ldi r19,$4f
        rjmp endDecodificar
dispquatro:
        ldi r19,$66
        rjmp endDecodificar
dispquinco:
        ldi r19,$6d
        rjmp endDecodificar
dispseis:
        ldi r19,$7d
        rjmp endDecodificar
dispsiete:

```

```

        ldi r19,$27
        rjmp endDecodificar
dispocho:
        ldi r19,$7f
        rjmp endDecodificar
dispnueve:
        ldi r19,$6f

endDecodificar:
        ret

```

Conclusiones

Cruces Luna Oscar Francisco: Esta práctica me resulto relativamente sencilla y un poco redundante, ya que se utilizo lo aprendido en prácticas anteriores.

Pérez Hernández Julio: En esta práctica continuamos usando contadores y manejo de registros, me pareció una práctica que ayuda a reforzar lo que aprendimos de contadores y decodificadores.

Bibliografía

https://es.wikipedia.org/wiki/Lenguaje_ensamblador

<https://www.microchip.com/wwwproducts/en/ATmega8535>

Practica 2: Hola multiplexado

Introducción

En esta práctica se realizó un circuito con un microcontrolador programado que mostró por medio de 6 displays la palabra -HOLA-, para esto es necesario primero multiplexar un puento para que encienda cada display con diferentes letras, para ello se utilizaron 2 timers en el programa, modificando la frecuencia para que se vea de manera clara.

Material

- Computadora con software AVR Studio
- 1 microcontrolador ATMEGA8535
- Fuente de voltaje
- Protoboard
- Cable
- Leds
- Resistencias
- 6 display 7 segmentos

Desarrollo

```
;--INSTITUTO POLITECNICO NACIONAL--;  
;---ESCULA SUPERIOR DE COMPUTO---;  
  
;-----CRUCES LUNA OSCAR FRNACISCO  
;-----PEREZ HERNANDEZ JULIO  
.include "m8535def.inc"  
.def col=r17  
.def dato=r16  
ldi dato,low(RAMEND)  
out SPL,dato  
ldi dato,high(RAMEND)  
out SPH,dato  
out ser dato  
out DDRC,dato  
out DDRB,dato  
ldi dato,$01  
mov R0,dato  
ldi dato,$6f  
mov R1,dato  
ldi dato,$70  
mov R2,dato  
ldi dato,$7e  
mov R3,dato  
ldi dato,$6d  
mov R4,dato  
ldi dato,$01  
mov R5,dato  
clr ZH  
  
uno: clr ZL  
        ldi col,$04  
dos: out PORTB,col  
        ld dato,Z+  
        out PORTC,dato  
        rcall delay  
        out PORTC,ZH  
        LSL col  
        brcc dos  
        rjmp uno  
        delay:  
        ldi r18,4  
        ldi r19,229  
L1: dec r19  
        brne L1  
        dec r18  
        brne L1  
        ret
```

Conclusiones

Cruces Luna Oscar Francisco: En esta práctica exploramos una nueva utilidad dentro del microcontrolador, llamado Timer, el cual nos permitió desarrollar un programa más complejo desplegando letras en diferentes displays con una velocidad variable de acuerdo a la frecuencia.

Pérez Hernández Julio: Con esta práctica conocimos un aspecto más del microcontrolador, implementando timers y manejando sus frecuencias realizamos una práctica que va más allá de las que hemos estado realizando.

Practica 3: Hola scroll

Introducción

En esta práctica se realizó un circuito similar al anterior con un microcontrolador programado que mostró por medio de 6 displays la palabra -HOLA-, pero esta vez la palabra debe hacer un recorrido de un lado hacia al otro.

Material

- Computadora con software AVR Studio
- 1 microcontrolador ATMEGA8535
- Fuente de voltaje
- Protoboard
- Cable
- Leds
- Resistencias
- 6 display 7 segmentos
- 1 inversor 7406

Desarrollo

```
;---ESCULA SUPERIOR DE COMPUTO---;

;-----CRUCES LUNA OSCAR FRNACISCO
;-----PEREZ HERNANDEZ JULIO
#include "m8535def.inc"
.macro modi
    modiMacroLabel:
        cpi @0,@1
        brlo exitModiMacro

        subi @0,@1
        rjmp modiMacroLabel

    exitModiMacro:

.endmacro

.macro lsli
    push r16
    ldi r16,@1

    lsliMacroLabel:
        cpi r16,0
        breq exitLsliMacro
```

```

        lsl @0
        dec r16
        rjmp lsliMacroLabel

exitLsliMacro:
        pop r16

.endmacro

.macro lsir
        push r16
        mov r16,@1

        lsirMacroLabel:
            cpi r16,0
            breq exitLsirMacro

            lsl @0
            dec r16
            rjmp lsirMacroLabel

        exitLsirMacro:
            pop r16

.endmacro

.def temp = R16
.def cont = R17 ;
.def char = R18 ;

.set NUM_DISPLAYS = 6 ;
.set vector = $64; >= $60
.set string_size = 7; 9 ; tamaño de la palabra

START: ; Main program start
        ldi temp, high(RAMEND)
        out SPH,temp ; Set Stack Pointer to top of RAM
        ldi temp, low(RAMEND)
        out SPL,temp

        ser temp
        out DDRA,temp ; se declara el puerto A como salida
        out DDRC,temp ; se declara el puerto C como salida
        out DDRB,temp ; se declara el puerto B como salida
        out DDRD,temp ; se declara el puerto D como salida
        ;out PORTB,temp ; se declara el puerto B como entrada y pull ups
        ;out PORTD,temp ; se declara el puerto D como entrada y pull ups

        clr cont
        clr char

        rcall initVector

```



```

main: ; BUCLE INFINITO

ldi r20, 120; 8ms * 120 = 1s aprox
L1:

    rcall showChar
    rcall delay8ms

    ; clear halo
    ldi temp,0
    out PORTA,temp

    ; next iteration
    inc cont
    modi cont,NUM_DISPLAYS

    dec r20
    brne L1
    nop

; termino 1s
inc char
modi char,string_size

rjmp main

```

```

showChar:
    push r19
    push r20
    push yh
    push yl

    ; obtener indice de caracter
    mov r19,cont
    add r19,char
    modi r19,string_size

    ; obtener posicion de caracter
    ldi r20,vector
    add r19,r20

    ; obtener caracter
    ldi yh,0
    mov yl,r19
    ld r19,y ; el caracter esta R19

    ; obtener display
    ldi r20,1
    lslr r20,cont

```

```

; show char to selected display
out PORTC,R20
out PORTA,R19

pop yl
pop yh
pop r20
pop r19

ret ;; RETURN FROM showChar

```

```

delay8ms:
    push r20
    push r21

    ; Generated by delay loop calculator
    ; at http://www.bretmulvey.com/avrdelay.html
    ;
    ; Delay 7 996 cycles
    ; 7ms 996us at 1 MHz

        ldi r20, 11
        ldi r21, 97
L0:    dec r21
        brne L0
        dec r20
        brne L0
        rjmp PC+1

    pop r21
    pop r20

    ret ;; RETURN FROM delay8ms

```

```

delay500ms:
    push r20
    push r21
    push r22

    ; Generated by delay loop calculator
    ; at http://www.bretmulvey.com/avrdelay.html
    ;
    ; Delay 499 996 cycles
    ; 499ms 996us at 1 MHz

        ldi r20, 3
        ldi r21, 138
        ldi r22, 85
L2:    dec r22
        brne L2
        dec r21
        brne L2
        dec r20

```

```

        brne L2
        nop

    pop r22
    pop r21
    pop r20

    ret ;; RETURN FROM delay500ms

initVector:
    ldi xh,0
    ldi xl,vector

    ldi r19,$00 ; ' '
    st x+,r19
    ;ldi r19,$40 ; '-'
    ;st x+,r19
    ldi r19,$4f ; 'E'
    st x+,r19
    ldi r19,$5b ; 'S'
    st x+,r19
    ldi r19,$0f ; 'C'
    st x+,r19
    ldi r19,$3f ; 'O'
    st x+,r19
    ldi r19,$1e;$37 ; 'n'
    st x+,r19
    ldi r19,$3c;$37 ; 'n'
    st x+,r19
    ;ldi r19,$40 ; '-'
    ;t x+,r19

    ret

```

Conclusiones

Cruces Luna Oscar Francisco: En esta práctica continuamos con el uso de timers, y recorrido de biots para poder recorrer las letras, sin embargo a mi me pareció una de las practicas más difíciles durante el curso, sobre todo la parte de entender como pasar un letra de un display a otro.

Pérez Hernández Julio: Con esta práctica practicamos aun más el uso de los timers y configurar su frecuencia de trabajo, requirió un trabajo extra para hacer el scroll respecto a la practica anterior.