



**INSTITUTO POLITÉCNICO NACIONAL**

---

**ESCUELA SUPERIOR DE CÓMPUTO**

**PRÁCTICAS 1ER DEPARTAMENTAL:**

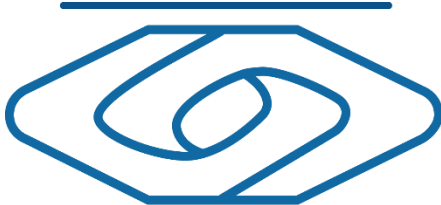
Introducción a los Microcontroladores

Grupo: 3CM8

Alumnos:

Cruces Luna Oscar Francisco

Pérez Hernández Julio



**ESCOM**

## Contenido

Practica 1: Sumador con cambio de velocidad .....	4
Introducción.....	4
Material .....	4
Desarrollo .....	5
Conclusiones.....	5
Bibliografía.....	6
Practica 2: Decodificador .....	6
Introducción.....	6
Material .....	6
Desarrollo .....	6
Conclusiones.....	8
Bibliografía.....	8
Practica 3: Sumador .....	9
Introducción.....	9
Material .....	9
Desarrollo .....	9
Conclusiones.....	10
Bibliografía.....	10
Practica 4: Contador de sumas .....	10
Introducción.....	10
Material .....	11
Desarrollo .....	11
Conclusiones.....	12
Bibliografía.....	12
Practica 5: Ventanillas .....	12
Introducción.....	12
Material .....	12
Desarrollo .....	13
Conclusiones.....	15

Bibliografía .....	16
Practica 5: Examen (NIBBLES) .....	16
Introducción .....	16
Material .....	16
Desarrollo .....	16
Conclusiones .....	18
Bibliografía .....	18

## Practica 1: Sumador con cambio de velocidad

### Introducción

Como primera práctica es imprescindible saber lenguaje ensamblador y conocer el microcontrolador con el que se trabajará durante el resto del curso, el cual es el Microcontrolador ATMEGA8535:

El microcontrolador AVR RISC de 8 bits de Microchip de alto rendimiento y bajo consumo de energía combina 8 KB de memoria flash programable, 544B SRAM, 512B EEPROM y un convertidor A / D de 8 canales y 10 bits. El dispositivo admite un rendimiento de 16 MIPS a 16MHz y funciona entre 4.5-5.5 voltios.

Por otro lado, ejecuta instrucciones en un solo ciclo de reloj, el dispositivo logra rendimientos que se acercan a 1 MIPS por MHz, equilibrando el consumo de energía y la velocidad de procesamiento.

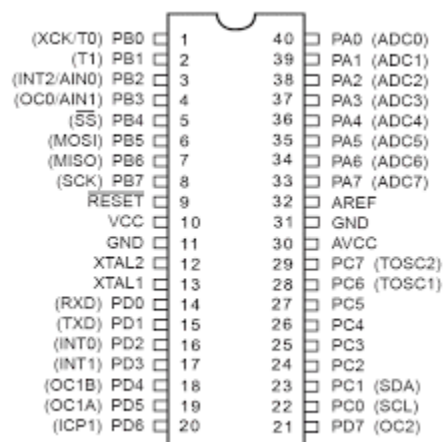


Diagrama de pines ATMEGA8535

Lenguaje ensamblador: es un lenguaje de programación de bajo nivel. Consiste en un conjunto de mnemónicos que representan instrucciones básicas para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables. Implementa una representación simbólica de los códigos de máquina binarios y otras constantes necesarias para programar una arquitectura de procesador y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador.

### Material

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

## Desarrollo

```
;-INSTITUTO POLITECNICO NACIONAL-;  
;---ESCULA SUPERIOR DE COMPUTO---;  
  
;-----CRUCES LUNA OSCAR FRNACISCO  
;-----PEREZ HERNANDEZ JULIO  
.include "m8535def.inc"  
.def data = r18  
.def aux = r17  
.def datb = r19  
.def cont = r20  
  
ldi r17, low(RAMEND)  
out SPL,r17  
ldi r17, high(RAMEND)  
out SPH,r17  
  
ser aux  
clr cont  
out ddra,aux  
out portb,aux  
out ddrc,aux  
  
loop:  
    in datb,pinb  
    call delay  
    inc cont  
    out porta,cont  
    rjmp loop  
  
delay:  
    mov r21, datb  
    ldi r22, 148  
L1: dec r22  
    brne L1  
    dec r21  
    brmi exit  
    brne L1  
exit:  
    ret
```

## Conclusiones

**Cruces Luna Oscar Francisco:** En esta práctica utilizamos instrucciones para corrimiento de bits y para sumar registros, además claro de quedar clara la configuración de los puertos para que funcionen como entradas o salidas.

**Pérez Hernández Julio:** Para esta práctica se implementaron puertos como entradas y salidas, y su configuración nos quedó clara; además quedó clara la importancia de realizar operaciones lógicas como parte de un programa en ensamblador.

## Bibliografía

[https://es.wikipedia.org/wiki/Lenguaje\\_ensamblador](https://es.wikipedia.org/wiki/Lenguaje_ensamblador)

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

## Practica 2: Decodificador

### Introducción

Un decodificador es un circuito combinatorio combinacional, cuya función es convertir un código binaria de entrada de N bits de entrada y M líneas de salida, tales que cada línea de salida será activada para una sola de las combinaciones posibles de entrada.

Para este caso en especifico se prendé programar un decodificador de BCD para un display de 7 segmentos, el cual mostrará el número que se introduce por medio de interruptores por uno de los puertos del microcontrolador.

### Material

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

### Desarrollo

```
;--INSTITUTO POLITECNICO NACIONAL--  
;---ESCULA SUPERIOR DE COMPUTO---  
  
;-----CRUCES LUNA OSCAR FRNACISCO  
;-----PEREZ HERNANDEZ JULIO  
.include "m8535def.inc"  
.def datb = r16  
  
ldi r17, low(RAMEND)  
out SPL,r17  
ldi r17, high(RAMEND)  
out SPH,r17  
  
ser r17  
out ddra,r17  
out ddrc,r17  
out ddrd,r17  
out portb,r17
```

```

read:
    in datb,pinb

    mov r18,datb
    rcall module
    out porta,r18

    rjmp read

module:
    ldi r20,0
menos:
    cpi r18,$a
    brlo xxxx
    inc r20
    subi r18,$a
    rjmp menos
xxxx:
    rcall decodificar
    mov r21,r18
    mov r18,r20
    rcall decodificar
    mov r20,r18
    mov r18,r21
    ret

decodificar:
    cpi r18,$0
    breq dispzero
    cpi r18,$1
    breq dispuno
    cpi r18,$2
    breq dispdos
    cpi r18,$3
    breq disptres
    cpi r18,$4
    breq dispquatro
    cpi r18,$5
    breq dispquinco
    cpi r18,$6
    breq dispseis
    cpi r18,$7
    breq dispseven
    cpi r18,$8
    breq dispchocho
    cpi r18,$9
    breq dispnueve

    dispzero:
        ldi r18,$3f
        rjmp endDecodificar
    dispuno:

```

```

        ldi r18,$06
        rjmp endDecodificar
dispdos:
        ldi r18,$5b
        rjmp endDecodificar
disptres:
        ldi r18,$4f
        rjmp endDecodificar
dispcuatro:
        ldi r18,$66
        rjmp endDecodificar
dispcinco:
        ldi r18,$6d
        rjmp endDecodificar
dispseis:
        ldi r18,$7d
        rjmp endDecodificar
dispsiete:
        ldi r18,$27
        rjmp endDecodificar
dispocho:
        ldi r18,$7f
        rjmp endDecodificar
dispnueve:
        ldi r18,$6f

endDecodificar:
    ret

```

## Conclusiones

**Cruces Luna Oscar Francisco:** En esta práctica quedó claramente comprendido el uso de instrucciones se saltos condicionales, así como la implementación y uso de subrutinas como parte del programa para modularizar partes del programa como en la verificación del número a desplegar.

**Pérez Hernández Julio:** Para implementar un decodificador dentro del microcontrolador fue necesario hacer múltiples comprobaciones, para lo cual utilizamos sentencias de salto condicional durante la mayor parte de la codificación de la práctica.

## Bibliografía

[https://www.academia.edu/14925237/ELECTRONICA\\_CODIFICADORES\\_Y\\_DECODIFICADORES](https://www.academia.edu/14925237/ELECTRONICA_CODIFICADORES_Y_DECODIFICADORES)

[http://ikastaroak.ulhi.net/edu/es/IEA/ELEC/ELEC02/es\\_IEA\\_ELEC02\\_Contenidos/website\\_541\\_decodificador\\_bcd\\_de\\_7\\_segmentos.html](http://ikastaroak.ulhi.net/edu/es/IEA/ELEC/ELEC02/es_IEA_ELEC02_Contenidos/website_541_decodificador_bcd_de_7_segmentos.html)



## Practica 3: Sumador

### Introducción

Un sumador es un circuito que realiza la suma de dos palabras binarias. Es distinta de la operación OR, con la que no nos debemos confundir. La operación suma de números binarios tiene la misma mecánica que la de números decimales.

Por lo que en la suma de números binarios con dos o más bits, puede ocurrir el mismo caso que podemos encontrar en la suma de números decimales con varias cifras: cuando al sumar los dos primeros dígitos se obtiene una cantidad mayor de 9, se da como resultado el dígito de menor peso y "me llevo" el anterior a la siguiente columna, para sumarlo allí.

### Material

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

### Desarrollo

```
;-INSTITUTO POLITECNICO NACIONAL-;
;---ESCULA SUPERIOR DE COMPUTO---;

;-----CRUCES LUNA OSCAR FRNACISCO
;-----PEREZ HERNANDEZ JULIO
.include "m8535def.inc"
.def data = r17
.def aux = r16
.def datb = r18
.def pbutton = r20
.def cont = r19

ser aux
clr cont
out ddra,aux
out portb,aux
out portd,aux
out ddrc,aux

loop:
    in pbutton,pind
    cpi pbutton,0
    breq loop

    in datb,pinb
```

```

    in data,pina

    add data,datab
    inc cont

    out porta,data
    out portc,cont

aqui:
    in pbutton,pind
    cpi pbutton,0
    brne aqui
    rjmp loop

```

## Conclusiones

**Cruces Luna Oscar Francisco:** Esta práctica sencilla nos ayudó como instrucción al lenguaje ensamblador comprendiendo conceptos básicos como la configuración de puertos, subrutinas y manejo de registros. Me pareció muy interesante la pequeña rutina para el censado del push botón, ya que se cicla a menos que detecté el cambio.

**Pérez Hernández Julio:** A pesar de ser una práctica introductoria, me ayudo a dejar claro la inicialización de puertos, y así mismo además de utilizar instrucciones para la suma binaria, se hizo el incremento de registros en este caso para un contador.

## Bibliografía

[http://educativa.catedu.es/44700165/aula/archivos/repositorio/4750/4923/html/7\\_sumadores.html](http://educativa.catedu.es/44700165/aula/archivos/repositorio/4750/4923/html/7_sumadores.html)  
[https://personales.unican.es/manzanom/Planantiguo/EDigital/SumG2\\_09.pdf](https://personales.unican.es/manzanom/Planantiguo/EDigital/SumG2_09.pdf)

## Practica 4: Contador de sumas

### Introducción

En programación, se llama contador a una variable cuyo valor se incrementa o decrementa en un valor fijo (en cada iteración de un bucle). Un contador suele utilizarse para contar el número de veces que itera un bucle. Pero, a veces, se utiliza para contar, solamente, aquellas iteraciones de un bucle en las que se cumpla una determinada condición.

El contador se utiliza para llevar la cuenta de determinadas acciones que se pueden solicitar durante la resolución de un problema.

En las instrucciones de preparación se realiza la inicialización del contador o contadores. La inicialización consiste en poner el valor inicial de la variable que representa al contador. Generalmente se inicializa con el valor 0.

En esta practica a diferencia del la anterior se realiza la suma no con ayuda de un push button, sino una vez que el microcontrolador detecta un cambio en los valores de entrada,

## Material

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

## Desarrollo

```
;--INSTITUTO POLITECNICO NACIONAL--;  
;---ESCULA SUPERIOR DE COMPUTO---;  
  
;-----CRUCES LUNA OSCAR FRNACISCO  
;-----PEREZ HERNANDEZ JULIO  
.include "m8535def.inc"  
.def data = r17  
.def aux = r16  
.def datb = r18  
.def cont = r19  
  
ser aux  
clr cont  
out ddra,aux  
out portb,aux  
out ddrc,aux  
  
loop:  
    in datb,pinb  
    cpi datb,0  
    breq loop  
  
    in data,pina  
    add data,darb  
    inc cont  
    out porta,data  
    out portc,cont  
aki:  
    in datb,pinb
```

```
    cpi datb,0  
    brne aki  
    rjmp loop
```

## Conclusiones

**Cruces Luna Oscar Francisco:** Esta práctica me sirvió de refuerzo ya que a diferencia de la anterior se tuvo que hacer un manejo diferente de los registros para retener el valor de la suma así como para detectar el cambio de los valores de la entrada.

**Pérez Hernández Julio:** Fue interesante ver como con una simple instrucción es posible determinar el estado de un bit en los puertos de entrada, lo que nos ayudo a detectar el cambio y realizar el contador de manera similar al anterior.

## Bibliografía

<https://sites.google.com/site/fprogati/contadores-y-acumuladores>

[http://www.carlospes.com/minidiccionario/variable\\_contador.php](http://www.carlospes.com/minidiccionario/variable_contador.php)

## Practica 5: Ventanillas

### Introducción

Ciertas veces es necesario llevar el control del personal en algunos establecimientos, una de las soluciones que se pueden proponer es implementar un contador de turnos que asigne a las personas a distintas ventanillas.

Para ello esta práctica fue necesario integrar dos practicas pasadas agregando algunos cambios; por un lado, un contador que se incrementa desde 0 a 99 al presionar botones que imitan a las personas, y por otro lado una rutina que controle la asignación a tres diferentes ventanillas. Por último hacer uso del decodificador para mostrar las salidas en displays.

### Material

- 1 microcontrolador ATMEGA8535
- Fuente de voltaje
- Protoboard
- Cable
- Leds
- Resistencias
- 3 push button
- 4 displays de 7 segmentos
- 1 Dip.switch

## Desarrollo

```
;-INSTITUTO POLITECNICO NACIONAL-;
;---ESCULA SUPERIOR DE COMPUTO---;

;-----CRUCES LUNA OSCAR FRNACISCO
;-----PEREZ HERNANDEZ JULIO
.include "m8535def.inc"
.def datb = r17
.def vent = r18
.def turno = r16

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

clr vent
clr turno

leer:
    in datb,pinb
    cpi datb,0
    breq leer

    sbic pinb,0
    rjmp cero
    sbic pinb,1
    rjmp uno
    sbic pinb,2
    rjmp dos
    sbic pinb,3
    rjmp tres
    sbic pinb,4
    rjmp cuatro

aki:
    in datb,pinb
    cpi datb,0
    brne aki
    rjmp leer

cero:
    ldi vent,1
    rjmp aumentar
uno:
    ldi vent,2
    rjmp aumentar
dos:
```

```

        ldi vent,3
        rjmp aumentar
tres:
        ldi vent,4
        rjmp aumentar
cuatro:
        ldi vent,5

aumentar:
        inc turno

        mov r19,vent
        rcall decodificar
        out porta,r19

        mov r19,turno
        rcall modulo
        out portc,r19
        out portd,r20

        rjmp aki

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 dispquince
        cpi r19,$6

```

```

breq dispseis
cpi r19,$7
breq dispsiete
cpi r19,$8
breq dispocho
cpi r19,$9
breq dispnueve

dispcero:
    ldi r19,$3f
    rjmp endDecodificar
dispuno:
    ldi r19,$06
    rjmp endDecodificar
dispdos:
    ldi r19,$5b
    rjmp endDecodificar
disptres:
    ldi r19,$4f
    rjmp endDecodificar
dispcuatro:
    ldi r19,$66
    rjmp endDecodificar
dispcinco:
    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 requirió de encontrar la manera de integrar el conocimiento que adquirimos en prácticas pasadas, por un lado un contador, una rutina para detectar el cambio de estado en los pines de entrada del puerto B, para poder determinar la ventanilla asignada y por último implementar el decodificador a siete segmentos.

**Pérez Hernández Julio:** En esta sesión de laboratorio, se puso en práctica instrucciones que previamente usamos en ejercicios anteriores, por lo que la dificultad recayó en encontrar la forma de conectar los módulos del programa.

## Bibliografía

## Practica 5: Examen (NIBBLES)

### Introducción

Un nibble es la mitad de un byte, o sea, 4 bits.

Un nibble es una colección de cuatro bits, esto no representaría una estructura interesante si no fuera por dos razones: El Código Binario Decimal (BCD por sus siglas en inglés) y los números hexadecimales. Se requieren cuatro bits para representar un sólo dígito BCD ó hexadecimal.

Con un nibble se pueden representar 16 valores diferentes, en el caso de los números hexadecimales, cuyos valores 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, y F son representados con cuatro bits. El BCD utiliza diez dígitos diferentes (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) e igualmente se requiere de cuatro bits. De hecho se puede representar 16 elementos diferentes con un sólo nibble pero los dígitos hexadecimales y BCD son los principales representados por un nibble.

Para este examen se pidió dividir un byte en nibbles, dependiendo del estado de un bit sumar y mostrar en un puerto distinto.

### Material

- 1 microcontrolador ATMEGA8535
- Fuente de voltaje
- Protoboard
- Cable
- Resistencias
- 1 barra de leds
- 1 Dip.switch

### Desarrollo

```
;-INSTITUTO POLITECNICO NACIONAL-;  
;---ESCULA SUPERIOR DE COMPUTO---;  
  
;-----CRUCES LUNA OSCAR FRNACISCO  
;-----PEREZ HERNANDEZ JULIO  
.include "m8535def.inc"  
.def data = r17  
.def aux = r16  
.def datb = r18
```



```

.def cont = r19
.def a = r20
.def b = r21
.def operacion = r22
.def pos = r23
.def nega = r24
.def result = r25

ser aux
clr cont
; salidas
out ddra,aux
out ddrc,aux
; entradas
out portb,aux
out portd,aux

loop: in datb,pinb
      cpi datb,0
      breq loop

      mov aux, datb

      ;Obtencion de nibbles
      mov a,darb
      andi a,$0F
      mov b,darb
      swap b
      andi b,$0F

      in operacion,pind
      cpi operacion,0

      breq res
      add a,b
      rjmp sig

res:  sub a,b

sig:  brmi casoneg
      inc pos
      rjmp result

casoneg:  inc nega

result:  out porta,a

      mov result, nega
      andi result,$0F
      swap result
      or result, pos

      out portc, result

```

```
aqui: in datb, pinb  
      cpse datb, aux  
      rjmp loop  
      rjmp aqui
```

## Conclusiones

**Cruces Luna Oscar Francisco:** Esta práctica me enseñó como usar nibbles a nivel ensamblador, para lo cual una simple comparación lógica y la instrucción swap nos ayudó a obtener ambos nibbles en registros distintos, me parece interesante que el uso de nibbles se relaciona mucho con el sistema hexadecimal.

**Pérez Hernández Julio:** usar nibbles en cualquier lenguaje de programación resulta práctico y conveniente, y con tres instrucciones nos fue suficiente para obtener los nibbles de un puerto y con ello realizar operaciones. Se puso en práctica todo lo aprendido durante sesiones pasadas.

## Bibliografía

<https://www.lawebdelprogramador.com/diccionario/956-Nibble.html>