



# Instituto Politécnico Nacional



Escuela Superior de Cómputo

## **Alumnos:**

Esquivel Pérez Jonathan Alfredo  
Hernández López Ángel Zait  
Salgado Gallegos Jesús  
Sánchez Pizano Irving Daniel

## **Profesor:**

Pérez Pérez José Juan

## **Unidad de aprendizaje:**

Introducción a los microcontroladores

## **Práctica 3**

Decodificador Hexadecimal

**Grupo:** 3CM8

## Objetivo

Implementar un codificador en el cual se recibe una entrada binaria, y se muestra su número equivalente en hexadecimal en una salida representada por un display de 7 segmentos usando el Microcontrolador Atmega8535.

## Introducción

### Sistema Hexadecimal

El sistema hexadecimal es el sistema de numeración posicional que tiene como base el 16. Su uso actual está muy vinculado a la informática y ciencias de la computación donde las operaciones de la CPU suelen usar el byte u octeto como unidad básica de memoria; y, debido a que un byte representa  $2^8$  valores posibles, y esto puede representarse como:

$$2^8 = 2^4 \cdot 2^4 = 16 \cdot 16 = 1 \cdot 16^2 + 0 \cdot 16^1 + 0 \cdot 16^0$$

que equivale al número en base 16 al número 100, dos dígitos hexadecimales corresponden exactamente a un byte.

En principio, dado que el sistema usual de numeración es de base decimal y, por ello, sólo se dispone de diez dígitos, se adoptó la convención de usar las seis primeras letras del alfabeto latino para suplir los dígitos que nos faltan. El conjunto de símbolos es el siguiente:

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$

Se debe notar que  $A = 10$ ,  $B = 11$ ,  $C = 12$ ,  $D = 13$ ,  $E = 14$  y  $F = 15$ . En ocasiones se emplean letras minúsculas en lugar de mayúsculas. Como en cualquier sistema de numeración posicional, el valor numérico de cada dígito es alterado dependiendo de su posición en la cadena de dígitos, quedando multiplicado por una cierta potencia de la base del sistema.

### Decodificador

El decodificador es un circuito combinacional diseñado para convertir un número binario (entrada) en word de «unos» y «ceros» (niveles altos y bajos de voltaje) con un orden distinto, para ejecutar un trabajo especial. En otras palabras, el word que sale es diferente al word que entró, aunque tenga la misma cantidad de bits. En Electrónica Digital es a menudo necesario pasar un número binario a otro formato, tal como el requerido para energizar los siete segmentos de los display hechos con diodos emisores de luz, en el orden adecuado para que se ilumine la figura de un individual número en otra base.

## Material y equipo utilizado

- Protoboard
- Dip Switch 8 entradas
- Cable
- Pinzas de corte
- Resistencias de 100 Ohms
- Display 7 Segmento
- Microcontrolador ATMEGA8535
- Software AVR Studio

## Que se realizará en la práctica

Se implementará un codificador hexadecimal del 0 al F el cual mostrará el resultado dependiendo de la activación en el dip switch del numero binario. Una vez ingresada la combinación se verá reflejado en el display de 7 segmentos donde se podrá apreciar el número binario en su formato hexadecimal.

## Desarrollo de la práctica

Primero se realizo el código correspondiente a la practica como se muestra a continuación:

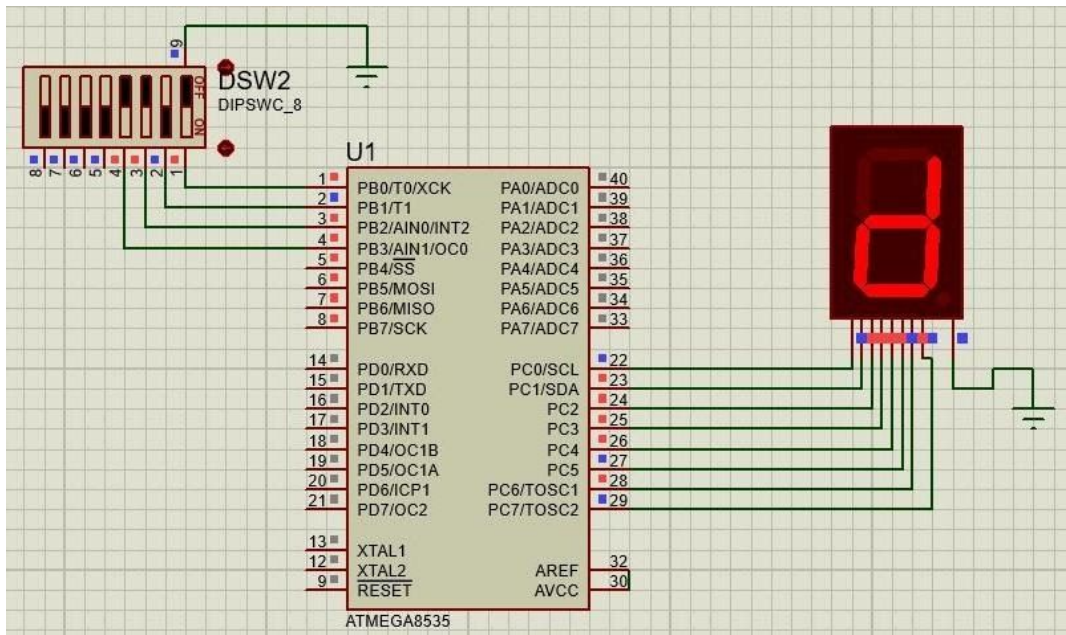
```
1. .include "m8535def.inc"
2. .def aux = r28
3. .def dato = r29
4.
5. reset:
6.     ser aux
7.     out DDRA, aux    ; Se habilita puerto A como salida
8.     out DDRC, aux    ; Se habilita puerto C como salida
9.     out portb, aux    ; Se habilita puerto B como entrada
10.
11. load:
12.     ldi aux, $3f
13.     mov r12,aux ;0
14.     ldi aux, 6
15.     mov r13,aux ;1
16.     ldi aux, $5b
17.     mov r14,aux ;2
18.     ldi aux, $4f
19.     mov r15,aux ;3
20.     ldi r16,$66 ;4
21.     ldi r17,$6d ;5
22.     ldi r18,$7d ;6
23.     ldi r19,$27 ;7
24.     ldi r20,$7f ;8
25.     ldi r21,$6f ;9
26.     ldi r22,$77 ;A
27.     ldi r23,$7c ;B
28.     ldi r24,$39 ;C
29.     ldi r25,$7e ;D
30.     ldi r26,$79 ;E
```

```

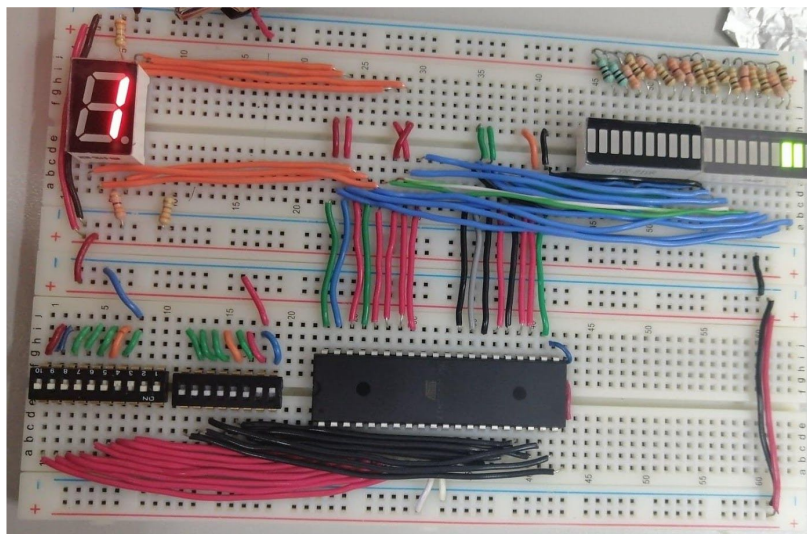
31.    ldi r27,$71 ;F
32.
33. otro:
34.    ldi ZL,12      ; Se carga a la parte de la pila ZL el valor 12
35.    in dato, pinb   ; Se carga a dato la entrada del puerto B
36.    andi dato,$0f   ; Se hace una and a de (registro & constante)
37.    out porta, dato ; Se pone a la salida dato
38.    add ZL,dato     ; Se suma el valor de dato a ZL
39.    ld dato,Z       ; Se carga a dato el valor de Z
40.    out portc,dato  ; Se pone en la salida C el valor de Z (BCD)
41.    rjmp otro

```

## Diagrama Electrico

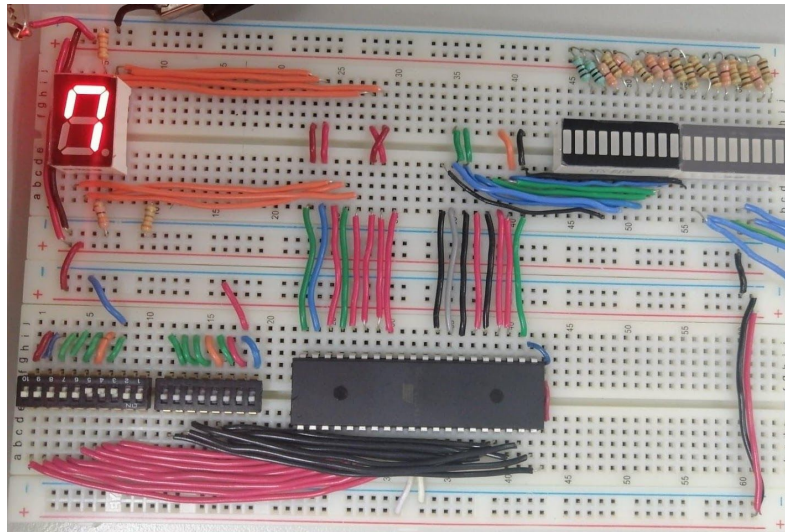


## Circuito y funcionamiento

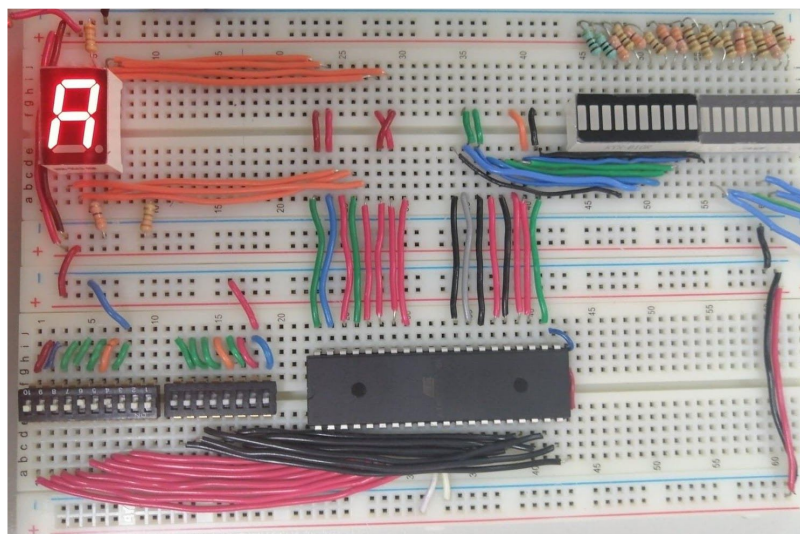


Decodificación de el numero 1

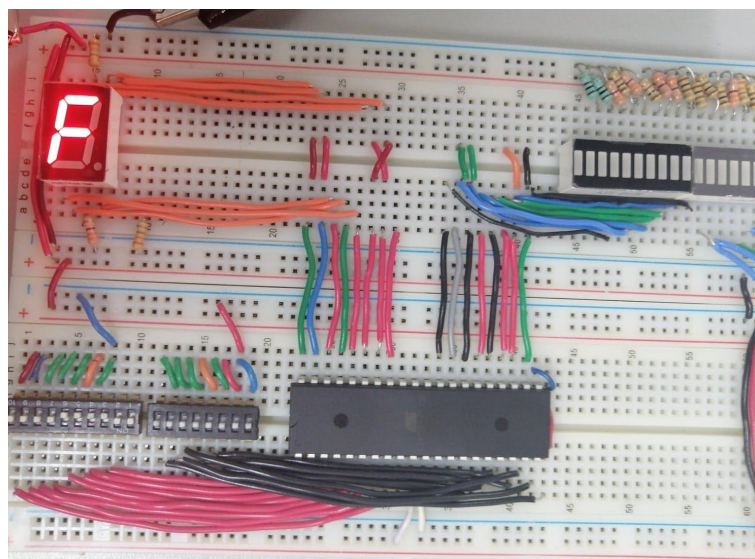




Decodificación de el numero 7



Decodificación de el numero 10 (A)



Decodificación de el numero 15 (F)

## Conclusiones

**Esquivel Pérez Jonathan Alfredo:** Al realizar esta práctica aprendimos en funcionamiento de los registros de ZH y ZL en la pila. Dado que las instrucciones para movimiento son distintas para cada uno de éstos su implementación resultó un poco compleja.

### **Hernández López Ángel Zait:**

En esta práctica se observó de nuevo el funcionamiento de la codificación del BCD; se tuvo dificultades con funcionamiento de la pila y los registros ZH y ZL, ya que las instrucciones de cada uno son diferentes para poder mover los datos entre ellos.

### **Salgado Gallegos Jesús**

En esta práctica hubo cierta dificultad en aplicar los registros altos y bajos del registro Z, la codificación de los números para que se muestran en el display de 7 segmentos es fácil, pero la lógica de ZH y ZL fueron más complicados de lo esperado, por lo cual fue necesario utilizar otro tipo de instrucciones para pasar registros bajos a altos.

### **Sánchez Pizano Irving Daniel:**

Esta práctica fue un poco complicado lo de la codificación en bcd, pero lo que se nos complicó un poco más fueron los registros de ZH y ZL, ya que su funcionamiento es diferente, por lo que tuvimos que también tuvimos que utilizar otras instrucciones y al final si se logró hacer el funcionamiento de la práctica.