



Instituto Politécnico Nacional

Escuela superior de Cómputo

Introducción a los microcontroladores

Pérez Pérez José Juan

Práctica 4: Contador de 0 a 99

Esquivel Pérez Jonathan Alfredo

Hernández López Ángel Zait

Salgado Gallegos Jesús

Sánchez Pizano Irving Daniel

3CM8

Junio 2020

Objetivo

El objetivo de esta práctica es realizar un contador de 0 a 99, utilizando las salidas del microcontrolador para poner un número binario a una salida BCD de 7 segmentos para observar el número en binario.

Introducción

El código BCD (Binary-Coded Decimal o Decimal codificado en binario) es un estándar para representar números decimales en el sistema binario, en donde cada dígito decimal es codificado con una secuencia de 4 bits. Con esta codificación especial de los dígitos decimales en el sistema binario, se pueden realizar operaciones aritméticas como suma, resta, multiplicación y división.

El BCD es muy común en sistemas electrónicos donde se debe mostrar un valor numérico, especialmente en los sistemas digitales no programados (sin microprocesador o microcontrolador). Utilizando el código BCD, se simplifica la manipulación de los datos numéricos que deben ser mostrados por ejemplo en un visualizador de siete segmentos. Esto lleva a su vez una simplificación en el diseño físico del circuito (hardware).

Material y equipo utilizado

- Protoboard
- Push button
- Cable
- Pinzas de corte
- Resistencias
- 2 display de 7 segmentos
- Microcontrolador ATMEGA8535
- Software AVR Studio

Desarrollo de la práctica

En esta práctica se debe hacer un contador de 0 a 99, no se usa un contador que binario de 0 a 99, si no que se utilizarán dos contadores, uno será para las unidades, y otro para las decenas. Cada vez que el contador de unidades sea igual a 10, este reiniciará su contador en cero y aumentará en uno el contador de los decimales, es decir, por lo que a la salida se verá como si el contador que está en binario contará en decimal.

Circuito electrónico

En esta parte se muestra un pequeño diagrama (Figura 1) de cómo se vería más o menos en un protoboard, este circuito está hecho en Proteus.

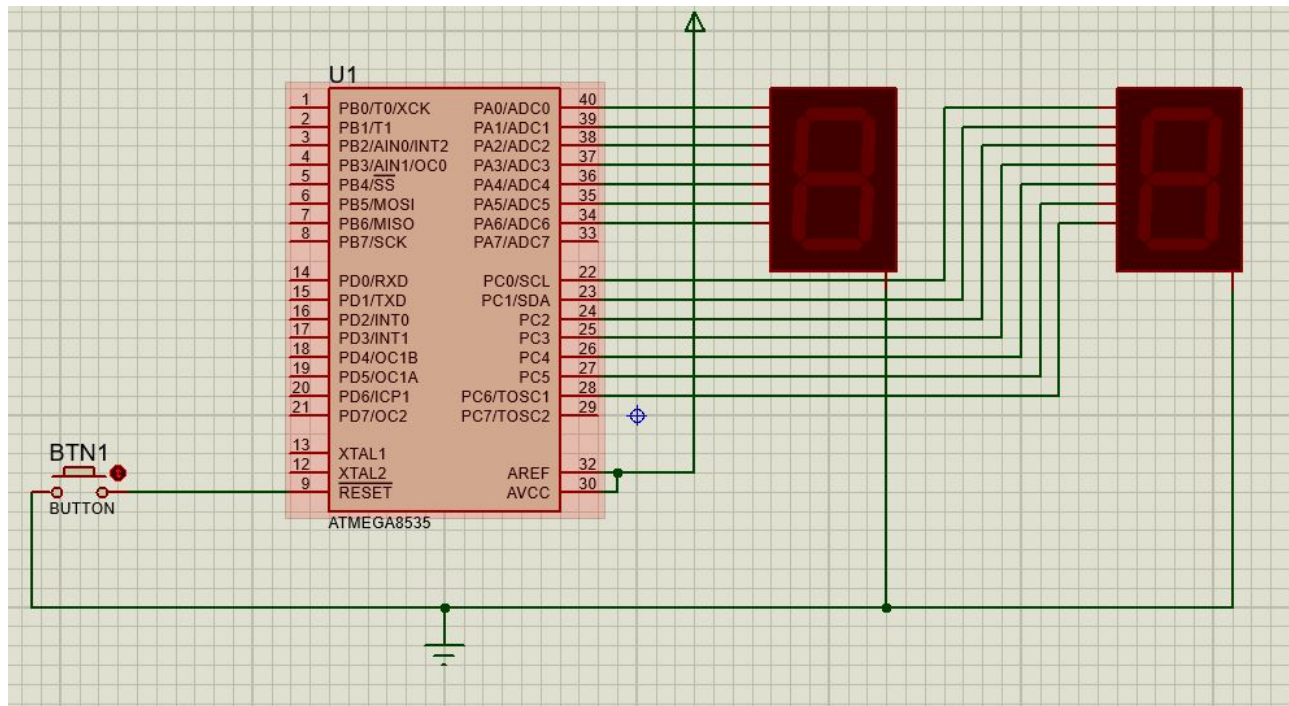


Figura 1. Circuito de la práctica

Diagrama de flujo

En este diagrama (Figura 2) se puede observar la idea principal de este contador.

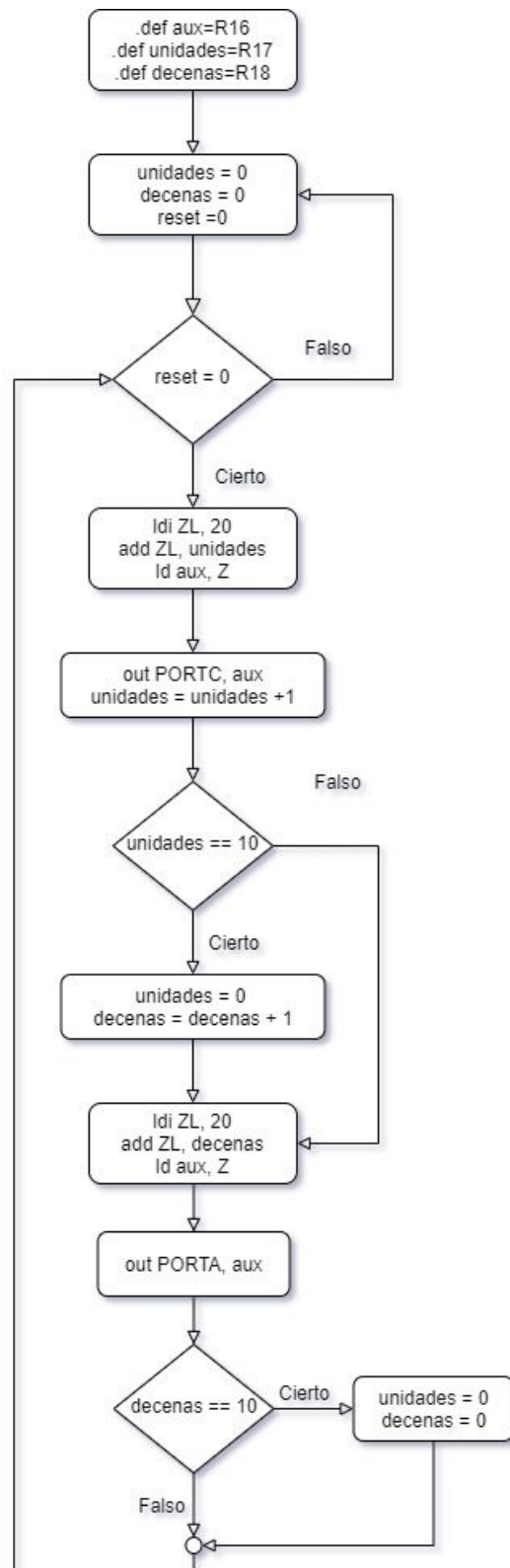


Figura 2. Diagrama de flujo

Código comentado

El código se realizó en lenguaje ensamblador, con ayuda de Atmel Studio 7.0, el código es el siguiente:

```
1. .include "m8535def.inc"
2. .def aux=R16
3. .def unidades=R17
4. .def decenas=R18
5.
6. rjmp main
7. .org $009          ; Salto al registro $009
8. rjmp aumen
9.
10.main:
11.    ldi aux, low(RAMEND)
12.    out spl, aux
13.    ldi aux, high(RAMEND)
14.    out sph, aux      ; Inicializacion de la pila
15.    ser aux
16.    out DDRC, aux
17.    out DDRA, aux
18.    ldi aux, 5
19.    out tccr0, aux
20.    ldi aux, 1
21.    out tmsk, aux
22.    sei
23.
24.load:
25.    ldi R20, $3F      ; 0
26.    ldi R21, $06      ; 1
27.    ldi R22, $5B      ; 2
28.    ldi R23, $4F      ; 3
29.    ldi R24, $66      ; 4
30.    ldi R25, $6D      ; 5
31.    ldi R26, $7D      ; 6
32.    ldi R27, $27      ; 7
33.    ldi R28, $7F      ; 8
34.    ldi R29, $6F      ; 9
35.    clr unidades
36.    clr decenas
37.    clr ZH
38.
39.loop:
40.    rcall BCD_unidades
41.    out PORTC, aux     ; Ponemos el valor de unidades en el puerto C
42.    rcall BCD_decenas
43.    out PORTA, aux     ; Ponemos el valor de decenas en el puerto A
44.    rjmp loop
45.
```

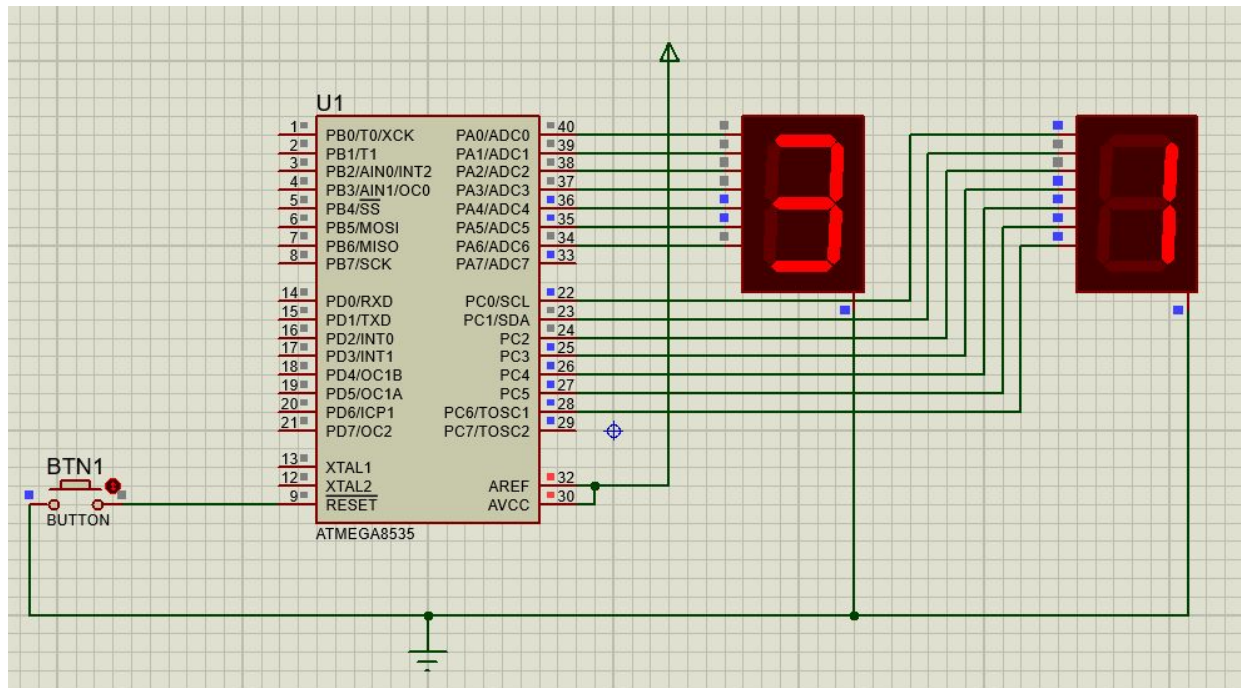
```

46.BCD_unidades:          ; Le damos a las unidades el valor BCD
47.    ldi ZL, 20
48.    add ZL, unidades
49.    ld aux, Z
50.    ret
51.
52.BCD_decenas:           ; Le damos a las decenas el valor BCD
53.    ldi ZL, 20
54.    add ZL, decenas
55.    ld aux, Z
56.    ret
57.
58.aumen:                 ; Se acciona el contador cada vez que se acabe un ciclo
59.    inc unidades        ; Se incrementa en uno unidades
60.    cpi unidades, 10
61.    breq a_dece
62.    reti
63.
64.a_dece:                ; Si las unidades es igual a 10
65.    clr unidades        ; Se pone unidades en cero
66.    inc decenas         ; Se incrementa en uno decenas
67.    cpi decenas, 10
68.    breq restart
69.    reti
70.
71.restart:               ; Si decenas es igual a diez
72.    clr unidades        ; Se inicia el contador en cero
73.    clr decenas         ; Es decir, se reinicia todo
74.    reti

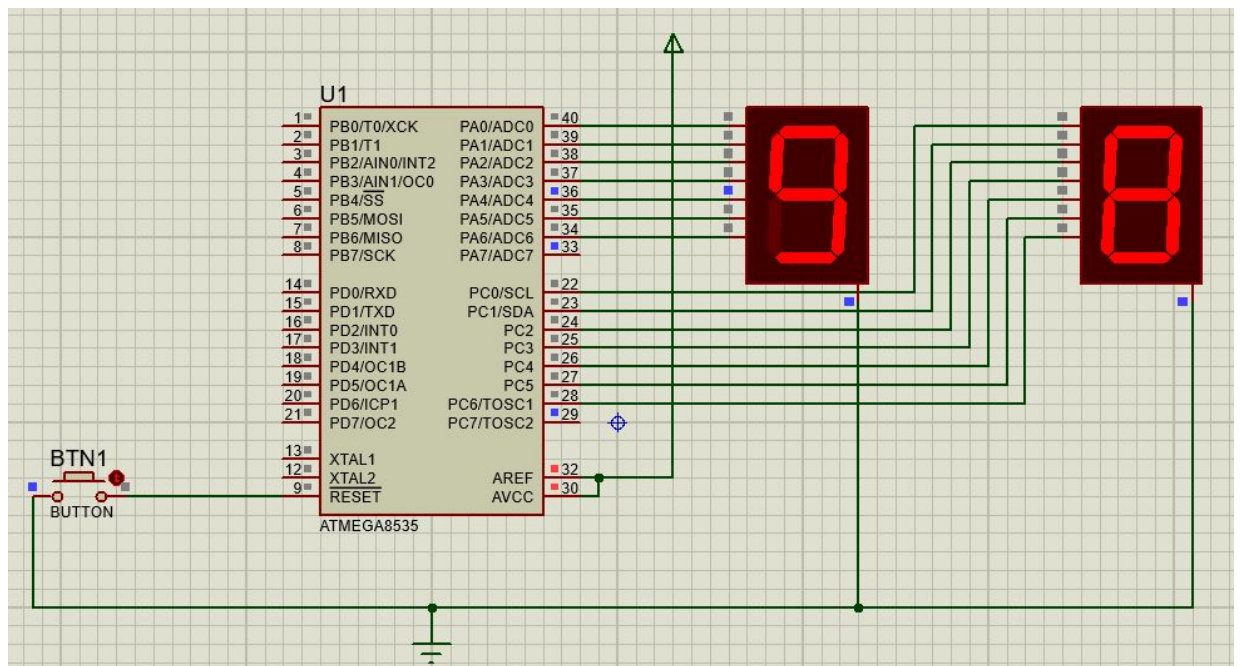
```

Simulación

En esta parte se muestra unas imágenes de cómo funciona el circuito en proteus (Figuras 3), para ver el funcionamiento, se anexará el proyecto para hacer pruebas si son necesarias.



a)



b)

Figura 3. a) Contador en 31; b) Contador en 98

Conclusiones

Esquivel Pérez Jonathan Alfredo:

En la presente práctica se implementó un BCD el cual recupera los datos guardados en la pila de microcontrolador para su posterior uso y visualización. En la práctica pudimos hacer uso del llamado a más de un registro y mediante el uso de una rutina pudimos aumentar en una decena cada vez que el dígito requería cambiar en decimales.

Hernández López Ángel Zait:

En esta práctica se pudo usar más de una variable al mismo tiempo, Se nos dificultó al principio el cómo poner los datos en código BCD, hasta que vimos que era necesario ya tenerlos guardados y poder jugar con la pila para poder ingresar a los valores BCD dependiendo el valor de la unidad o la decena para poder verse en los display de 7 segmentos.

Salgado Gallegos Jesús:

La practica cumplio el objetivo de desarrollar el contador de 00 a 99 con código BCD aunque hubo altibajos como el llamado a más de un registro y el uso de la pila donde necesitaba almacenar y sacar datos para poner el valor requerido en el display.

Sánchez Pizano Irving Daniel:

En esta práctica lo único que fue complicado fue el código BCD ya que no estábamos tan familiarizados con el, pero al investigar un poco nos dimos cuenta que teníamos que ya tenerlos guardados para posteriormente ser utilizados. Pero de ahí en fuera no tuvimos otro problema y todo se entendió correctamente.