INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Introducción a los microcontroladores

3CM16

Practica 07

Contador de 00 a 99 activado por IR

Equipo 7

Integrantes:

- Bocanegra Heziquio Yestlanezi
- Dominguez Durán Alan Axel
- Hernandez Mendez Oliver Manuel
- Martinez Cruz José Antonio

Profesor: Aguilar Sanchez Fernando

Índice

Índice de imágenes

Objetivo

Al término de la sesión, los integrantes del equipo contaran con la habilidad de realizar un contador sin rebotes de 00 a 99 mostrado en un par de displays activado con infrarrojo.

Introducción

Contador de 00 a 99 activado por IR

Un contador de 00 a 99 activado por IR es un circuito digital que utiliza una señal infrarroja (IR) para actualizar su estado y contar de forma ascendente o descendente. Este tipo de contador se utiliza comúnmente en aplicaciones de automatización, tales como el control de inventarios, el control de accesos, o en dispositivos de seguridad que requieren el conteo de objetos o personas [1].

La señal IR es emitida por un emisor IR y recibida por un receptor IR, el cual convierte la señal en un pulso eléctrico que es enviado al contador. Cada vez que la señal IR es recibida, el contador actualiza su estado y aumenta o disminuye su cuenta, dependiendo de si se ha configurado para contar hacia arriba o hacia abajo. [2].

Los contadores de 00 a 99 activados por IR son una herramienta útil en la automatización y el control de procesos, ya que permiten la medición precisa y el registro de eventos que ocurren a una velocidad alta, con un alto nivel de precisión. Además, el uso de señales IR ofrece varias ventajas en comparación con otros métodos de detección, como la detección por contacto o la detección por ultrasonido, ya que no requieren contacto físico y son menos susceptibles a la interferencia externa. [2].

Material y equipo empleado

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATmega 8535
- 2 Display cátodo común
- 15 Resistores de 330 Ω a 1⁄4 W
- 1 Resistor de 1K Ω a 1⁄4 W
- 1 Resistor de 680 Ω a 1⁄4 W
- 1 transistor 2N2222

Desarrollo experimental

1. Diseñe un programa en el que coloque dos Displays, uno en el Puerto A y el otro en el Puerto B y con una terminal del Puerto D.

Circuito armado

Con los conocimientos obtenidos en las diferentes materias de electrónica, procedemos a realizar el montado del circuito en la protoboard como se muestra en la imagen 1.

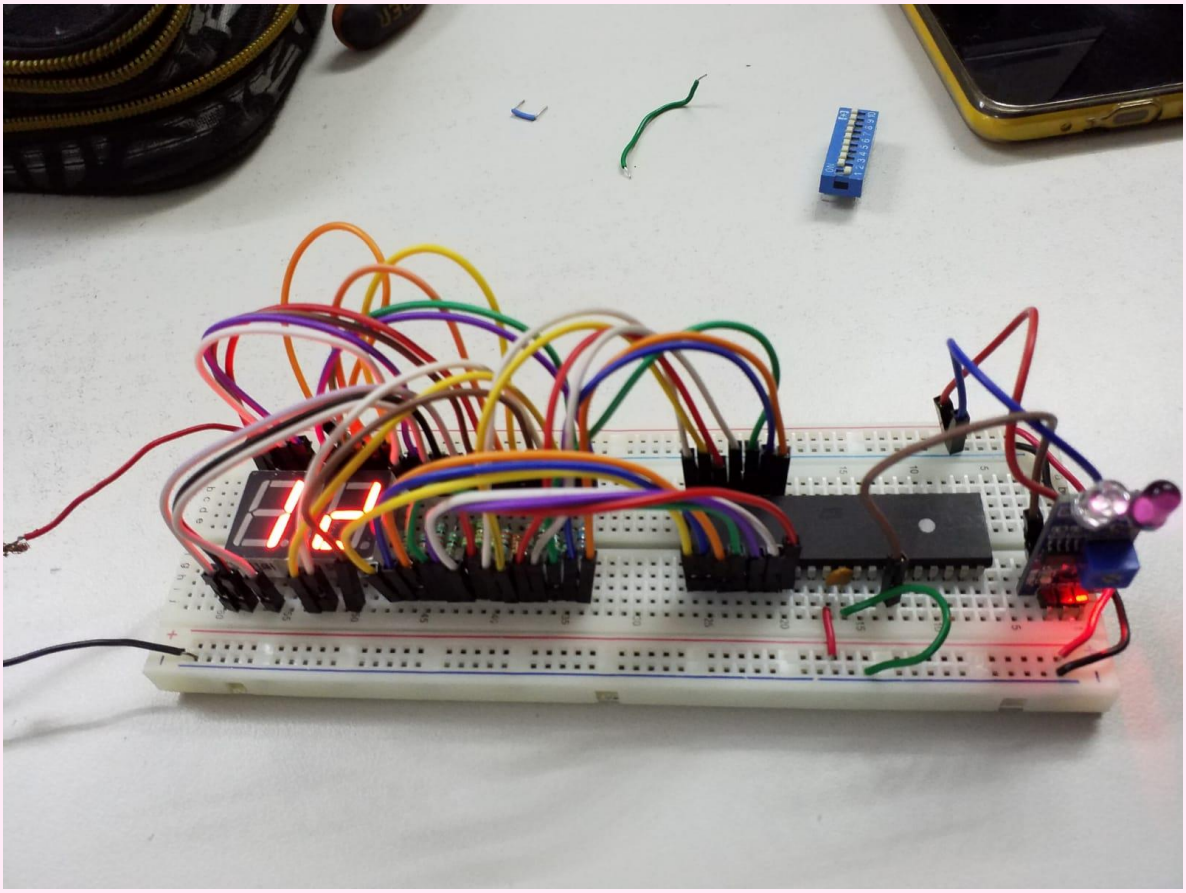

Imagen 1 contador de 00 a 99 activado por IR armado en protoboard

Estructura del programa

Código CodeVisionAVR

```
/*****************
This program was created by the CodeWizardAVR V3.51
Automatic Program Generator
© Copyright 1998-2023 Pavel Haiduc, HP InfoTech S.R.L.
http://www.hpinfotech.ro

Project :
Version :
Date    : 20/04/2023
Author  :
Company :
Comments:


Chip type               : ATmega8535
Program type            : Application
AVR Core Clock frequency: 1.000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 128
*****************/

// I/O Registers definitions

#include <mega8535.h>
#include <delay.h>


#define RESET_BTN PIND.0
#define STOP_BTN PIND.1
#define START_BTN PIND.2

const char mem[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x6F};
unsigned char unidad, decena, decima;
bit debe_avanzar;

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
```

```
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out
DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) |
(1<<DDA1) | (1<<DDA0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) |
(0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out
DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) |
(1<<DDB1) | (1<<DDB0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) |
(0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) |
(1<<DDC1) | (1<<DDC0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) |
(0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) |
(0<<DDD1) | (0<<DDD0);
// State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) |
(1<<PORTD2) | (1<<PORTD1) | (1<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) |
(0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
(0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) |
(0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) |
(0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
(0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);
```

```
// USART initialization
// USART disabled
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
(0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) |
(0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)
      {
        //Revisar por los botones
        if (!START_BTN) debe_avanzar = 1;
        if (!STOP_BTN) debe_avanzar = 0;

        if (!RESET_BTN) {
            decima = 0;
            unidad = 0;
            decena = 0;
            debe_avanzar = 0;
        }

        //Actuar seg?n el estado
        if (decena == 6) debe_avanzar = 0;
```

```
        if (debe_avanzar) {
            decima++;

            if (decima == 10) {
                decima = 0;
                unidad++;
            }

            if (unidad == 10) {
                unidad = 0;
                decena++;
            }
        }

        PORTA = mem[unidad];
        PORTB = mem[decena];
        PORTC = mem[decima];
        delay_ms(30);
      }
}
```

Simulación – Proteus

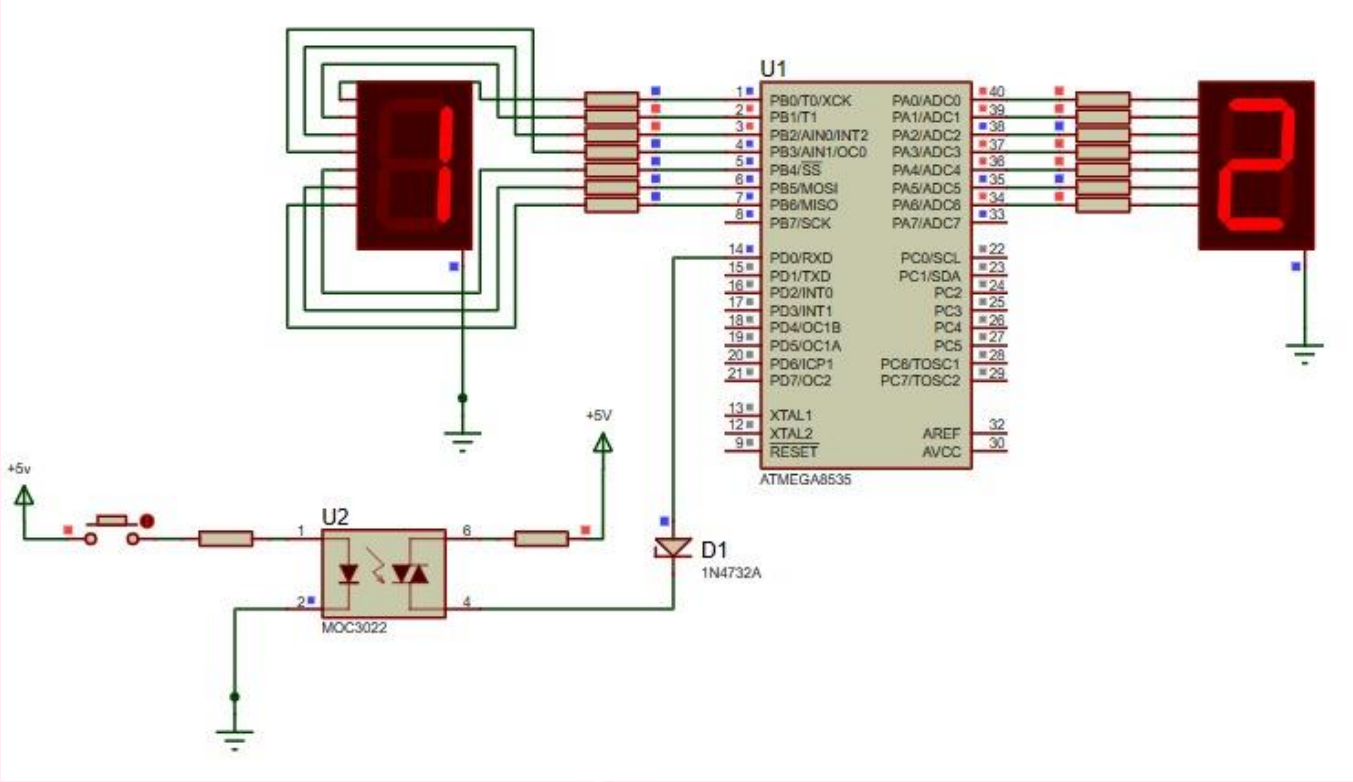

Imagen 2 Circuito simulado en proteus

Conclusiones

Bocanegra Heziquio Yestlanezi

En conclusión, de la practica 7 un contador de 00 a 99 activado por IR es un circuito digital que utiliza una señal infrarroja (IR) para contar objetos o personas en un proceso automatizado. Este contador es útil en aplicaciones de control de inventarios, control de accesos, sistemas de seguridad, entre otros. se activa por la señal IR, emitida por un emisor IR y recibida por un receptor IR. Cada vez que la señal IR es recibida, el contador actualiza su estado y aumenta o disminuye su cuenta, según la configuración del contador.

Domínguez Durán Alan Axel

Al termino de esta práctica se implementó un contador muy similar al que se controlaba con pulsos de push button; en este caso se utilizaron los pulsos de un sensor infrarrojo, lo que da pie a que se puedan implementar circuitos que puedan ser controlados de forma inalámbrica, aumentando las posibilidades de implementar circuitos para resolver problemas complejos.

Hernández Méndez Oliver Manuel

En particular esta práctica fue mi primer acercamiento al uso de sensores inalámbricos en un circuito armado, la implementación del sensor infrarrojo se asemejaba bastante al uso de los push buttons, solo que en este caso solo era necesario acercar la mano para ver el contador subiendo. Sin duda este tipo de hardware aumenta el panorama para la implementación de circuitos.

Martínez Cruz José Antonio

Gracias al uso del infrarrojo con sus circuitería necesaria ya implementada logramos realizar la práctica de manera correcta, realizando el conteo desde el 0 hasta el 99, así mismo este se reiniciaba al cero para comenzar de nuevo. Aunque existieron pequeños destalles en su construcción, se lograron superar.

Referencias

[1] M. Morris Mano and Charles R. Kime, "Logic and Computer Design Fundamentals," 4th ed. Upper Saddle River, NJ, USA: Pearson, 2008, pp. 542-544.

[2] T. L. Floyd, "Digital Fundamentals," 11th ed. Boston, MA, USA: Pearson, 2015, pp. 518-520.