INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Introducción a los microcontroladores

3CM16

Practica 15

Vólmetro de 0 a 5 Volts y Vólmetro de 0 a 30 Volts

Entrega 08-03-2023

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 hacer uso del convertidor analógico digital del microcontrolador implementando un Vólmetro de 0.0 a 5.0 Volts mostrado en dos displays de siete segmentos multiplexados.

Introducción

Convertidor Analógico-Digital

El proyecto de un voltímetro de 0.0 a 5.0 volts es una aplicación común en el campo de los microcontroladores. Un voltímetro es un dispositivo que se utiliza para medir la diferencia de potencial eléctrico o voltaje en un circuito. En este caso, el proyecto se centra en diseñar un voltímetro que pueda medir voltajes en el rango de 0.0 a 5.0 volts.

El objetivo principal de este proyecto es desarrollar un sistema de medición preciso y confiable utilizando un microcontrolador y otros componentes electrónicos. El microcontrolador se encargará de realizar la conversión analógico-digital (ADC) de la señal de voltaje, y posteriormente mostrará el valor medido en una pantalla LCD o mediante alguna otra interfaz de visualización [1].

Para implementar este proyecto, se requerirá el uso de componentes electrónicos como resistencias, condensadores y un convertidor analógico-digital (ADC) compatible con el microcontrolador seleccionado. Además, será necesario programar el microcontrolador para leer y procesar la señal de voltaje entrante, y luego presentar el resultado de manera legible[2].

Material y equipo empleado

- ♥ CodeVision AVR
- ♥ AVR Studio 4
- ♥ Microcontrolador ATmega 8535
- ♥ 2 Display Cátodo común
- ♥ 8 Resistores de 330 Ω a 1⁄4 W
- ♥ 2 Resistores de 1K Ω a 1⁄4 W
- ♥ 2 Transistores BC547 o 2N222.

Desarrollo experimental

Diseñe un programa para mostrar en dos Displays voltajes entre 0.0V a 5.0 V.

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.

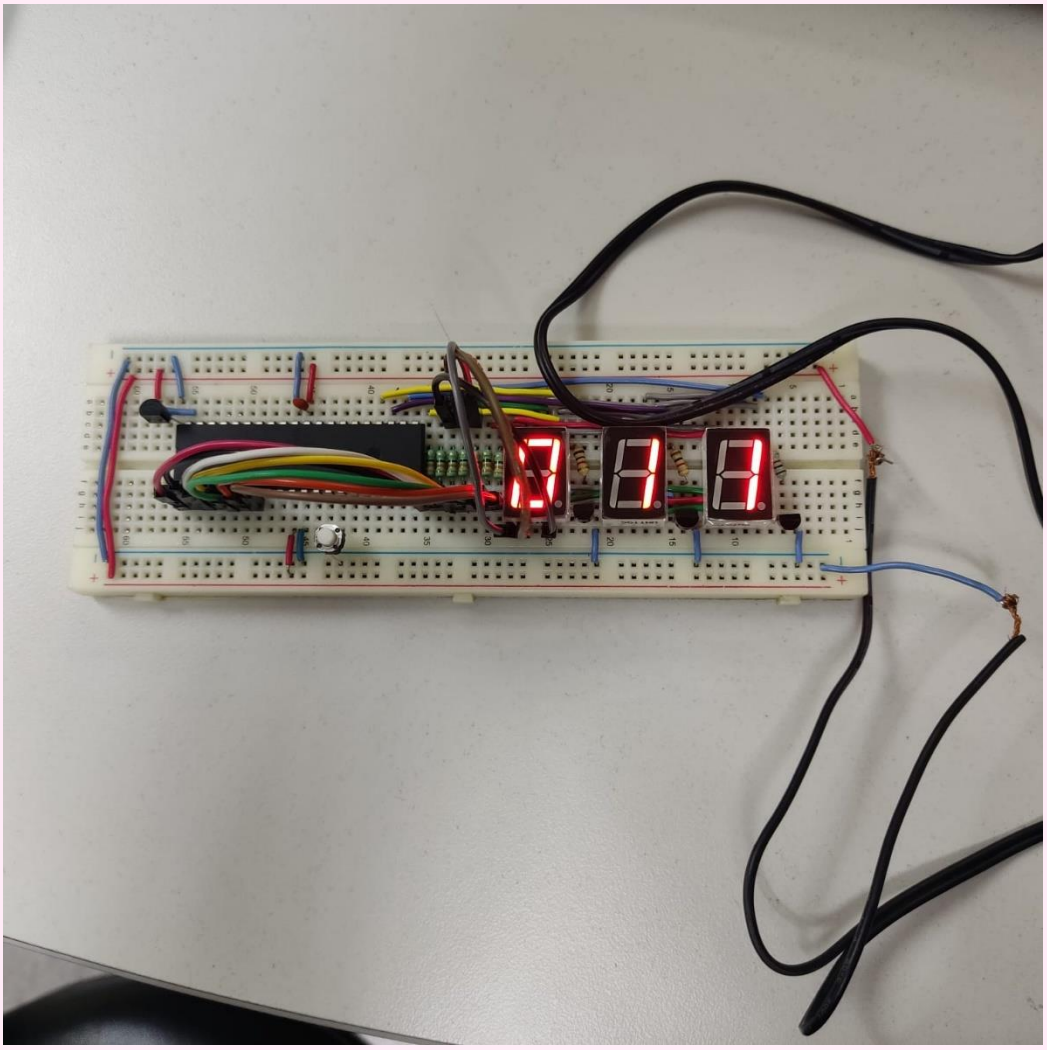

Imagen 1 Vólmetro de 0 a 5 Volts y Vólmetro de 0 a 30 Volts

Estructura del programa

Código CodeVisionAVR

```
/*****************************************************
This program was created by the CodeWizardAVR V3.48b
Automatic Program Generator
© Copyright 1998-2022 Pavel Haiduc, HP InfoTech S.R.L.
http://www.hpinfotech.ro

Project :
Version :
Date    : 13/11/2022
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>

// Declare your global variables here

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

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) |
(0<<DDA1) | (0<<DDA0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) |
(0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
```

```
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) |
(0<<DDB1) | (0<<DDB0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
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=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out
DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) |
(1<<DDD1) | (1<<DDD0);
// State: Bit7=1 Bit6=1 Bit5=1 Bit4=1 Bit3=1 Bit2=1 Bit1=1 Bit0=1
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)
      {
      // Place your code here
        PORTC=0x01;
        PORTD=0x00;
        delay_ms(10);
        PORTC=0x00;

        PORTC=0x02;
        PORTD=0x00;
        delay_ms(10);
        PORTC=0x00;

        PORTC=0x04;
        PORTD=0x00;
        delay_ms(10);
        PORTC=0x00;

        PORTC=0x08;
        PORTD=0x00;
        delay_ms(10);
        PORTC=0x00;

        PORTC=0x10;
        PORTD=0x00;
```

```
        delay_ms(10);
        PORTC=0x00;

      }
}
```

Simulación – Proteus

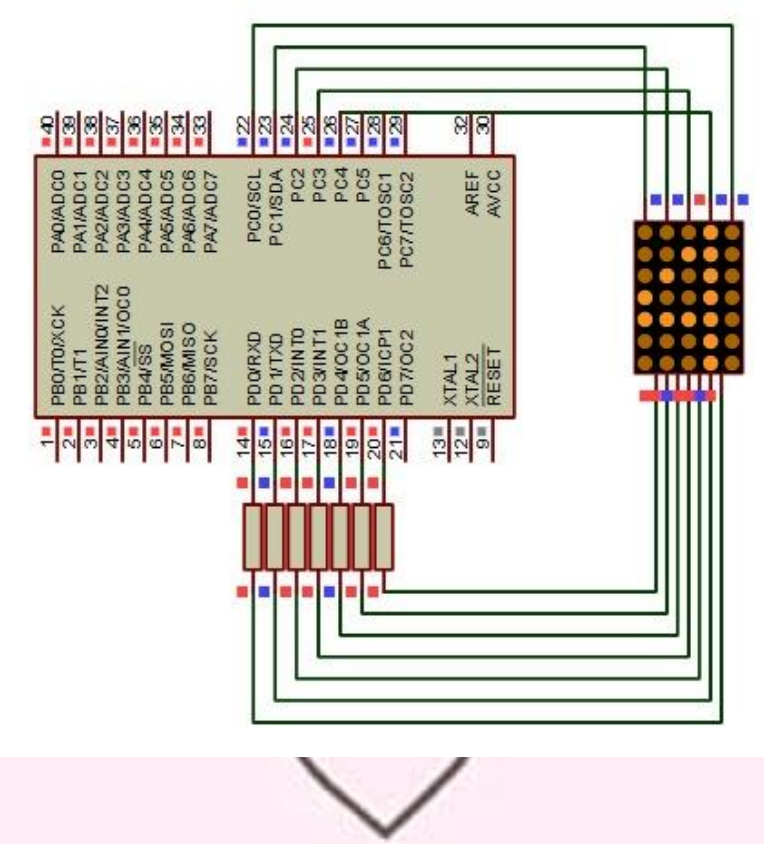

Imagen 2 Circuito simulado en proteus

Conclusiones

Bocanegra Heziquio Yestlanezi

Al concluir la practica adquirimos la capacidad de utilizar el convertidor analógico-digital del microcontrolador ATmega8535 para implementar un Vólmetro de 0.0 a 5.0 Volts mostrado en dos displays de siete segmentos multiplexados. Además, Utilizando CodeVision AVR y AVR Studio 4, programamos el microcontrolador de manera efectiva para convertir las señales analógicas de voltaje en datos digitales. Los 2 displays de siete segmentos, junto con los resistores asociados, permitirán mostrar de forma visual y legible los valores de voltaje medidos en el Vólmetro.

Por lo que ahora somos capaces de diseñar y construir un sistema funcional que mida y muestre el voltaje con precisión en dos displays de siete segmentos.

Domínguez Durán Alan Axel

Para esta práctica, igualmente utilizando el ADC, logramos captar las señales de entrada, en este caso son entradas analógicas de voltaje, y esta información la procesa el micro, de tal forma que se pueden mostrar cuanto voltaje esta entrando al puerto. De esta forma podemos mostrar como el micro puede funcionar como sensor de distintas fuentes y estímulos exteriores

Hernández Méndez Oliver Manuel

En mi opinión, la práctica de construir y utilizar un voltímetro de 0 a 5 volts y otro de 0 a 30 volts ha sido una experiencia muy enriquecedora en mi aprendizaje de microcontroladores, la construcción y calibración de los voltímetros fue un proceso desafiante pero gratificante. A medida que avanzaba en la práctica, fui capaz de superar obstáculos técnicos y obtener resultados más precisos. Aprendí la importancia de realizar pruebas y ajustes en cada etapa del proyecto para asegurarme de que el voltímetro funcionara correctamente y ofreciera mediciones exactas.

Martínez Cruz José Antonio

Para el desarrollo de esta práctica fue necesario comprender de manera clara el uso correcto del ADC, ya que puede pasar desapercibido el valor del voltaje de referencia. Esto puede generarnos valores erróneos y no nos permitiría alcanzar el objetivo de nuestra práctica. No existieron problemas al momento de implementar esta práctica, aunque cabe resaltar que para la implantación del volmetro de 30 volts solo se hizo una modificación al código para hacer la regla de tres correctamente, así mismo todos los displays mostraban los valores correctamente.

Referencias

[1] Boylestad, R. L., & Nashelsky, L. (2010). Electronic devices and circuit theory (11th ed.). Pearson.

[2] Bhadra, S., & Sen, D. (2017). Design of Voltmeter Circuit using Microcontroller ATmega328. International Journal of Electrical and Electronics Engineering Research (IJEEER), 9(2), 289-297.