



Instituto Politécnico Nacional
Escuela Superior de Cómputo
Departamento de Ciencias e Ingeniería
de la Computación



Sistemas en Chip

Práctica 15

“Vólmetro de 0.0 a 5.0 Volts y de 0.0 a 30.0 Volts”

Profesor: Fernando Aguilar Sánchez

Grupo: 6CM1

Equipo 3

Alumnos:

Ocampo Téllez Rodolfo

Patlani Mauricio Adriana

Ruvalcaba Flores Martha Catalina

Sandoval Hernández Eduardo

Fecha de entrega: 15/01/2023

Objetivo

Al término de la sesión, los integrantes del equipo contarán 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 Teórica

El microcontrolador ATMEGA8535 fabricante ATMEL. En la figura 1 se muestra el microcontrolador ATMEGA8535 la cual maneja datos de 8 bits es decir su bus de datos de 8 bits. Aunque este contiene tres registros los cuales son el x, y y z, los cuales manejan datos de 16 bits.

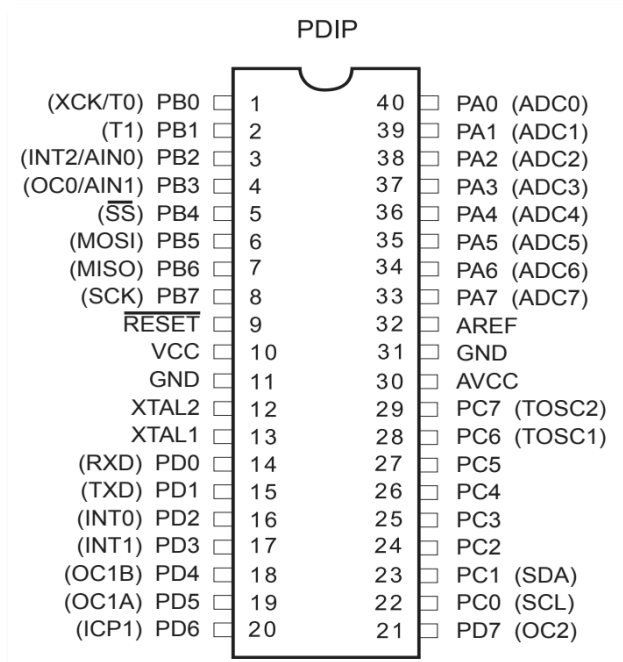


Figura 1. Configuración de pines ATmega8535

La comunicación interna del microcontrolador se categoriza en 4 puertos, en la figura 1 se puede analizar la etiquetación de los puertos PA0 al PA7, PB0 al PB7, PC0 al PC7 y PD0 al PD7. Cabe recordar que maneja datos de 8 bits.

Posee un oscilador interno de 1MHz, sin embargo, como es un oscilador RC, es susceptible a variar la frecuencia con respecto a las variaciones de temperatura. Por otro lado, puede conectarse un oscilador de 0 a 8 MHz o de 0 a 16 MHz.

Divisor de Voltaje

Un divisor de voltaje es un arreglo de 2 impedancias, comúnmente resistencias que dividen el voltaje y la corriente de salida. La división es proporcional a las resistencias involucradas en el divisor. Un divisor de voltaje se configura para tener una salida de potencial determinada, esta se puede calcular con una simple ecuación o fórmula.

La resistencia que va a tierra forma un paralelo con la resistencia de carga lo cual modifica el voltaje de la entrada de la carga. Usualmente los divisores como fuentes se implementan en conjunto con diodos zener y/o amplificadores operacionales. Esto considerando aplicaciones básicas. La configuración básica de un divisor de voltaje es la siguiente, en donde por convención vamos a llamar R1 y R2 a las resistencias. R1 la que va a la fuente y R2 la que va a tierra. Vi es el voltaje de entrada y Vo es el de salida del divisor.

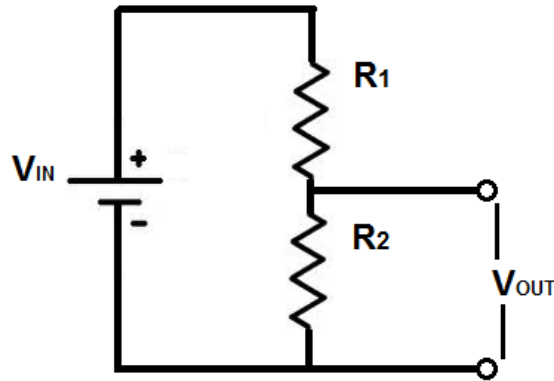


Figura 2. Configuración del divisor de voltaje.

La expresión que define el divisor de voltaje es la siguiente:

$$V_o = V_i \left(\frac{R_2}{R_1 + R_2} \right)$$

Despejando los elementos, de la expresión nos quedan los siguientes valores.

$$R_1 = R_2 \left(\frac{V_i - V_o}{V_o} \right)$$

$$V_i = V_o \left(\frac{R_2 + R_1}{R_2} \right)$$

En definitiva, sirve para obtener una tensión más pequeña partiendo de una tensión mayor. Por ejemplo si queremos alimentar una bombilla a 6V y solo disponemos de una pila de 10V. ¿Qué se puede hacer? Un divisor de tensión que nos divida la tensión de 10 voltios de la pila (Ventrada) en dos, una de 6V (Vsalida) y otra de 4V.

Voltímetro

El voltímetro es un instrumento de medición que se encarga de calcular la diferencia que existe en un potencial eléctrico entre dos puntos o lo que seguro habrás escuchado como voltaje, el cual es representado a través de los voltios.

En sí el voltímetro mide tanto las corrientes positivas como las negativas, pues ambas son importantes para poder determinar este diferencial en el potencial eléctrico. Este instrumento tiene además la capacidad de hacer estas mediciones tanto en la corriente alterna como en la continua.

El voltímetro es un instrumento que cuenta con una infinidad de funciones, lo cual lo hace un instrumento esencial en todo procedimiento eléctrico. Entre las más destacadas están:

- Realiza con mucha precisión mediciones en una amplia variedad de aparatos que funcionan con corriente eléctrica.
- Se encarga como te mencioné anteriormente de medir los voltajes o las diferencias de potencial.
- Miden el voltaje igualmente si existe corriente continua o corriente alterna.
- Permite calcular la continuidad de la corriente eléctrica.
- Al necesitar medir el flujo que circula por una resistencia, es de esperar que además del voltaje calcule la resistencia del circuito.
- Permite probar la capacidad que tienen las baterías.
- Además de medir las resistencias, pueden medir transistores eléctricos.



Figura 3. Voltímetro.

Materiales y Equipo empleado

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATmega 8535
- 2 Display Cátodo común
- 8 Resistores de 330Ω a $\frac{1}{4} W$
- 2 Resistores de $1K \Omega$ a $\frac{1}{4} W$
- 2 Transistores BC547 o 2N222.

Desarrollo Experimental

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

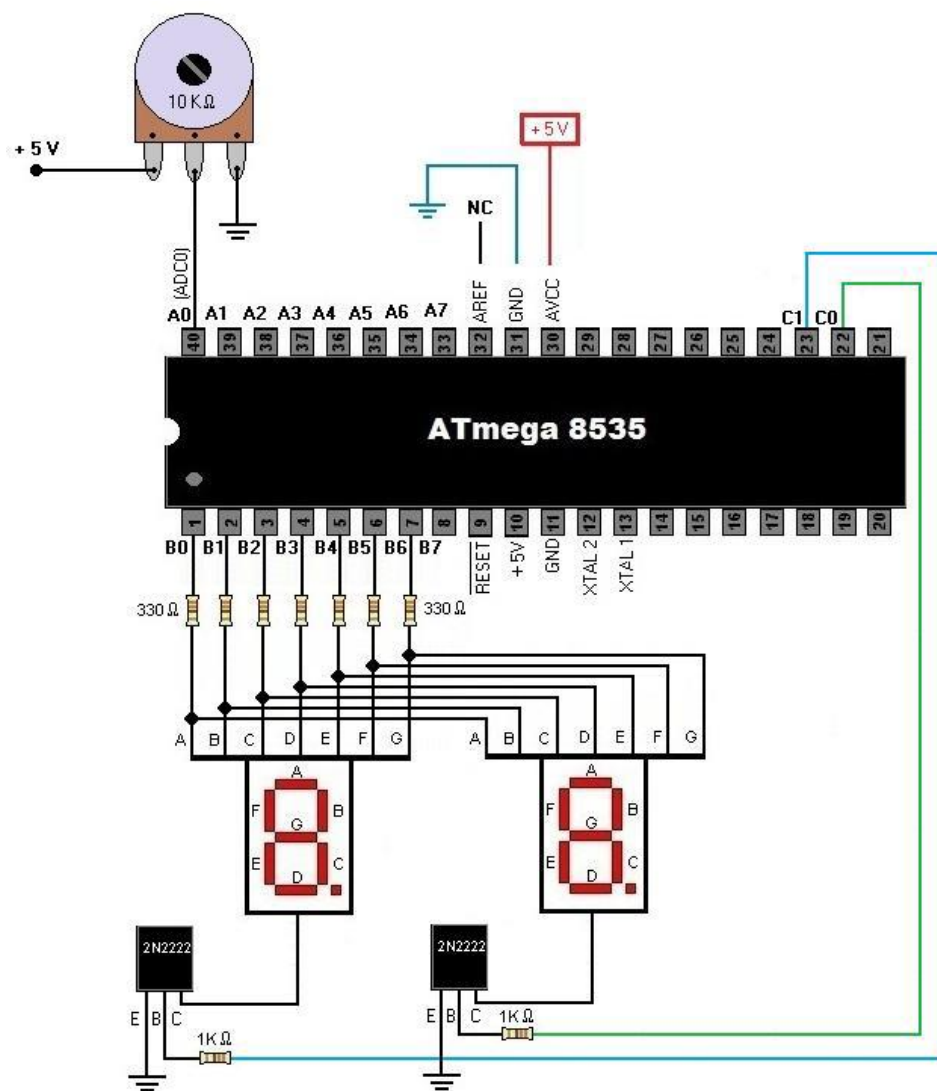


Figura 4. Circuito para el Voltímetro.

2.- Cálculo de las resistencias para el divisor de voltaje utilizado en la segunda parte de la práctica:

$$R_1 = R_2 \left(\frac{V_i - V_o}{V_o} \right)$$

Con una R_2 propuesta de $10\text{k}\Omega$, un voltaje de entrada de 30 V , y un voltaje de salida requerido de 5 V ,

$$R_1 = 10k\Omega \left(\frac{30V - 5V}{5V} \right) \rightarrow R_1 = 50k\Omega$$

Estructura del programa

Código de la configuración de los periféricos utilizados y código del programa principal en C para el voltmetro de 5V, proporcionados por el IDE de CodeVision.

```
#include <mega8535.h>
#include <delay.h>
// Voltage Reference: AREF pin
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (1<<ADLAR))

const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};
bit mux = 0;
unsigned char valorADC;
unsigned char valorFinal;

unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | ADC_VREF_TYPE;
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=(1<<ADSC);
    // Wait for the AD conversion to complete
    while ((ADCSRA & (1<<ADIF))==0);
    ADCSRA|=(1<<ADIF);
    return ADCH;
}

void main(void)
{
    // Input/Output Ports initialization
    // Port A initialization
    DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1)
    | (0<<DDA0);
    PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2)
    | (0<<PORTA1) | (0<<PORTA0);
    // Port B initialization
    DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1)
    | (1<<DDB0);
    PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2)
    | (0<<PORTB1) | (0<<PORTB0);
    // Port C initialization
    DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<DDC1)
    | (1<<DDC0);
    PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2)
    | (0<<PORTC1) | (0<<PORTC0);
```

```
// Port D initialization
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1)
| (0<<DDD0);
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2)
| (0<<PORTD1) | (0<<PORTD0);
...
...
...
while (1)
{
    valorADC = read_adc(0);
    valorFinal = 50 * valorADC / 255;

    if(mux){
        PORTB = tabla7segmentos [valorFinal/10];
        PORTC = 0x02;
    }
    else{
        PORTB = tabla7segmentos [valorFinal%10];
        PORTC = 0x01;
    }
    mux = ~mux;
    delay_ms(5);
}
```

Código de la configuración de los periféricos utilizados y código del programa principal en C modificado para el **voltmetro de 30V**, proporcionados por el IDE de CodeVision.

```
#include <mega8535.h>
#include <delay.h>
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (1<<ADLAR))

const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};
int mux = 0;
unsigned long valorADC;
unsigned long valorFinal;

unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | ADC_VREF_TYPE;
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=(1<<ADSC);
```

```
// Wait for the AD conversion to complete
while ((ADCSRA & (1<<ADIF))==0);
ADCSRA|=(1<<ADIF);
return ADCH;
}

void main(void)
{
// Input/Output Ports initialization
// Port A initialization
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1)
| (0<<DDA0);
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2)
| (0<<PORTA1) | (0<<PORTA0);
// Port B initialization
DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1)
| (1<<DDB0);
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2)
| (0<<PORTB1) | (0<<PORTB0);
// Port C initialization
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<DDC1)
| (1<<DDC0);
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2)
| (0<<PORTC1) | (0<<PORTC0);
// Port D initialization
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1)
| (0<<DDD0);
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2)
| (0<<PORTD1) | (0<<PORTD0);
...
...
...
while (1)
{
    valorADC = read_adc(0);
    valorFinal = 300 * valorADC / 255;

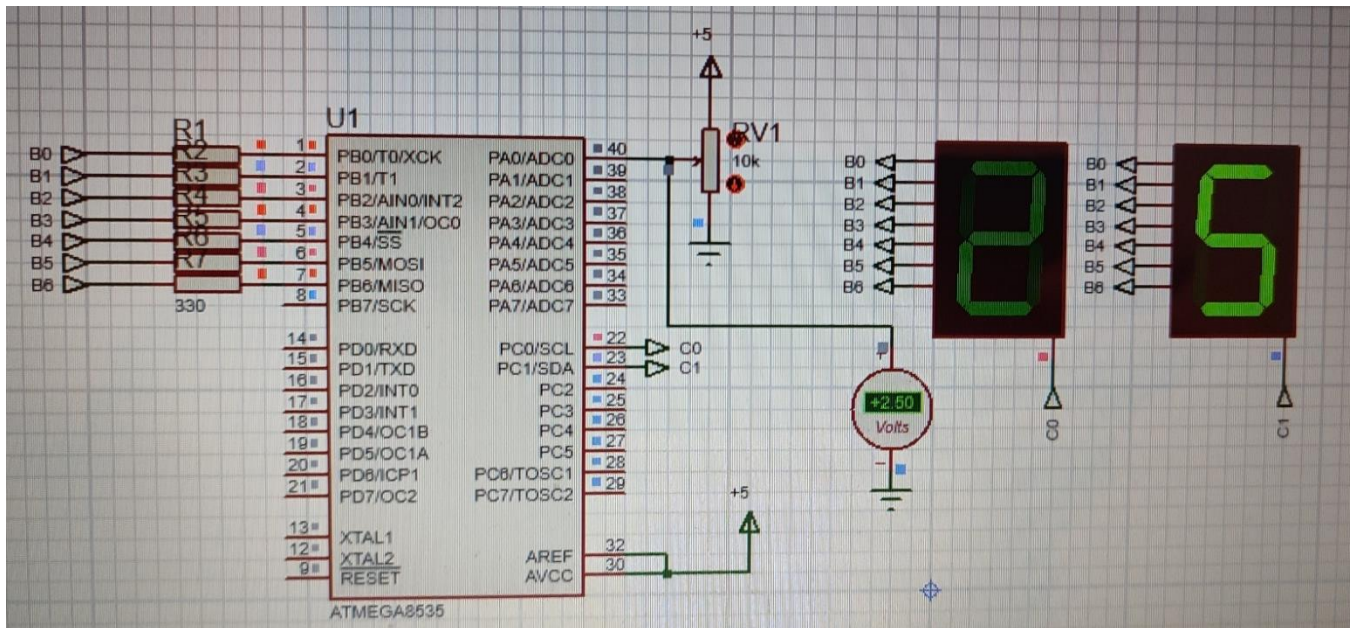
    if(mux == 0){
        PORTB = tabla7segmentos [valorFinal/100];
        PORTC = 0x06;
    }
    else if(mux == 1){
        PORTB = tabla7segmentos [valorFinal%100/10];
        PORTC = 0x05;
    }
    else if(mux == 2){
```

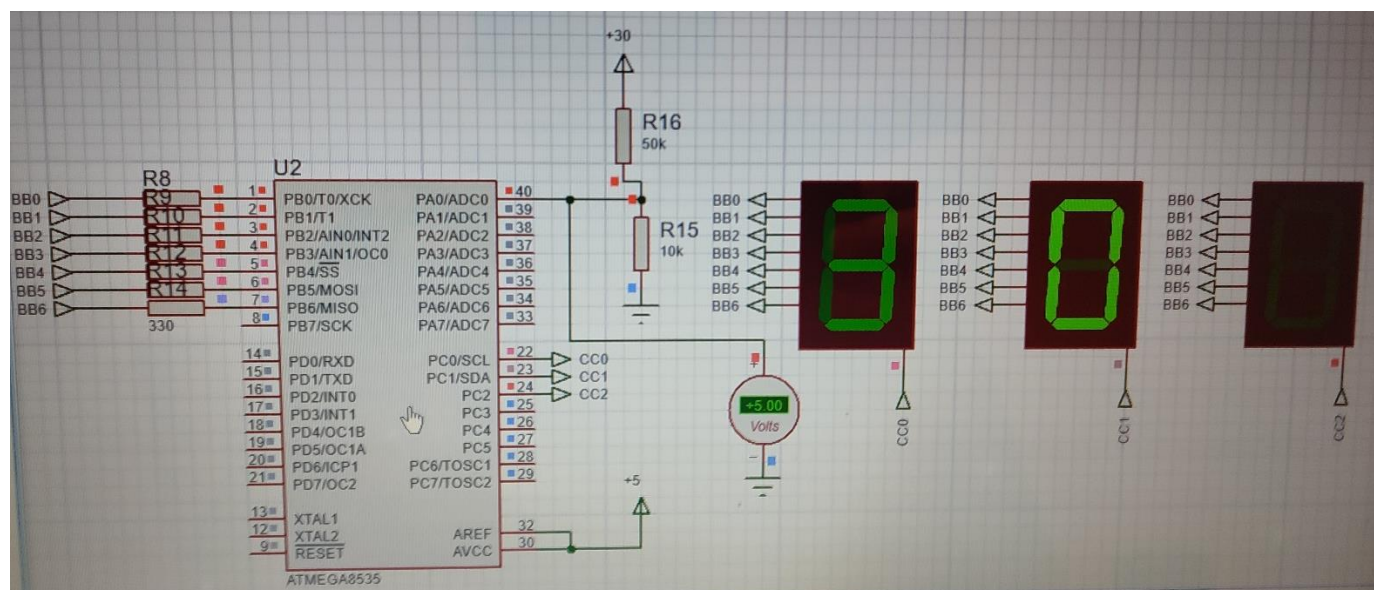


```
    PORTB = tabla7segmentos [valorFinal%10];  
    PORTC = 0x03;  
}  
mux++;  
if(mux == 3)  
    mux = 0;  
delay_ms(10);  
}  
}
```

Simulaciones

La simulación se realizó en el software Proteus Design Suite, previamente cargado el programa en formato “.hex” obtenido del software CodeVision AVR, el cual es un software para el microcontrolador Microchip AVR y sus contrapartes XMEGA.

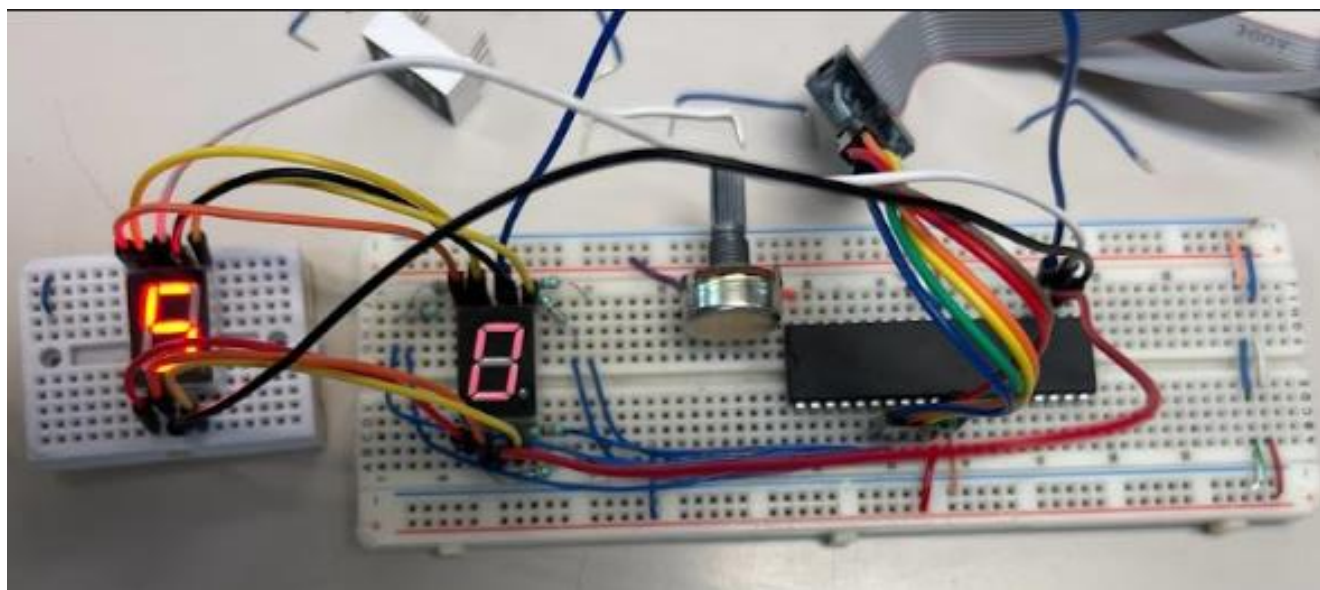


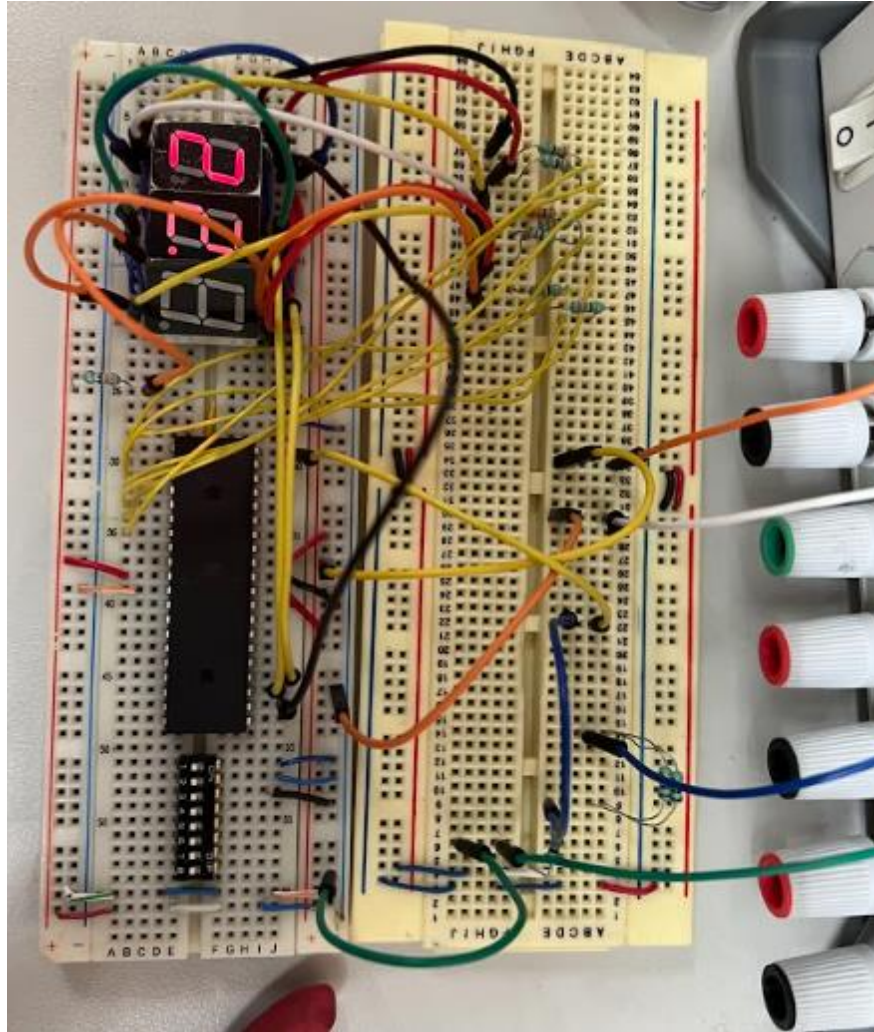


Figuras 5 (a y b). Simulación de la práctica 15.

Fotografía del circuito armado

A continuación, se muestra las figuras 6 la evidencia sobre la realización y prueba de la practica 15.





Figuras 6 (a y b). Circuito armado y funcionamiento de la práctica 15.

Observaciones y conclusiones Individuales

Ocampo Téllez Rodolfo

En esta práctica de nuevo se volvió a dar un uso al ADC del microcontrolador, en esta ocasión para obtener los valores analógicos de voltaje y mostrarlos digitalmente en un voltímetro cuyos valores se veían reflejados en displays de 7 segmentos. La práctica se dividió en 2 partes, en ambas se utilizaron 8 bits de resolución para el ADC, es decir, utilizando los valores de 0 a 255. Para la primera parte en la que se midieron valores de 0.0 a 5.0 V, el desarrollo fue sencillo siguiendo el diagrama del circuito y mediante la ayuda de una fórmula mencionada por el profesor en el video de explicación. Mientras que para la segunda parte en la que se midieron valores mucho mayores de 0.0 a 30.0 V se requirió modificar el circuito agregando un display más y aplicando un divisor de voltaje para evitar daños en este, ya que el Atmega 8535 funciona con 5 V, asimismo se la fórmula anterior modificada, en este caso se estableció con 300 valores diferentes, obviamente realizando la asignación de decenas, unidades y decimas correspondiente a cada display. Asimismo, se utilizó un delay bajo ya que al ocupar los displays multiplexados se requería dar una sensación de continuidad a la hora de observar los valores.

Patlani Mauricio Adriana

Continuando con el uso del convertir analógico a digital del microcontrolador, esta fue una de las aplicaciones más complicadas, puesto que se debe obtener valores analógicos de voltaje para convertir sus valores a digital usando valores de 0 a 255 (ocho bits), funciones como `read_adc(0)` y `CN`. El armado del circuito fue extenso al usar varios displays y también el uso de un generador de voltaje para medir valores de 0.0 a 5.0V y de 0.0 a 30.0v.

Ruvalcaba Flores Martha Catalina

Para el desarrollo de la práctica, se dividió en dos partes, primeramente, se realizó un voltmetro de 0 a 5 volts generado por un potenciómetro el cual, a través de un ADC realiza la conversión de analógico a digital para obtener los valores binarios de la señal analógica, con base al video se utilizó la función `read_adc(0)` para capturar dichos valores de 0 a 255 ya que se escogió de 8 bits. Y con el resultado de dicha función, esta se utilizó en la fórmula `CN` (Código Nuevo) la cual sirvió para capturar el valor del voltaje, generando un valor del 0 al 50, y así tener un resultado desde el 0.0 hasta el 5.0 volts, ya por último para poder reflejar en el display las unidades, decenas y centenas se aplicaron sus respectivas operaciones para obtener las equivalencias de dichas cantidades. Por otro lado, la segunda parte de la práctica se realizó en el laboratorio, ya que se requirió de una fuente. Para realizar el voltmetro de 0 a 30 volts, se tuvo que realizar un divisor de voltaje, con el fin de evitar que se dañara el microcontrolador, ya que este solamente puede recibir 5 volts de entrada.

Sandoval Hernández Eduardo

Al comienzo parecía que esta práctica sería sencilla pues es similar a las 2 prácticas anteriores, sin embargo, al momento de comenzar a realizarlas nos dimos cuenta de que era más complicado ya que se tenían que realizar 2 circuitos uno de ellos con 3 displays multiplexados lo cual fue un tanto laborioso de armar en la protoboard, el código para el circuito se complicó un poco debido a los cálculos que habían que realizar para convertir la entrada analógica a un valor digital, además de multiplexar los displays para mostrar los valores adecuados. Fue una práctica complicada pero útil para las prácticas siguientes.

Bibliografía

- Atmel Corporation. (2006). 8-bit Microcontroller with 8K Bytes In-System Programmable Flash. [Online]. Disponible en: <https://drive.google.com/file/d/1acpQaDlsyLHr3w3ReSgrlXRbshZ5Z-lb/view>
- F. Aguilar. (Mayo, 2020). Microcontroladores Voltmetro 30 Volts. [Online]. Disponible en: <https://youtu.be/2-rlcp7bmXA>
- H. Torres. (2018). Divisor de Voltaje o Tensión Formulas y Ejemplos. [Online]. Disponible en: <https://hetpro-store.com/TUTORIALES/divisor-de-voltaje/>
- Área Tecnología. (s.f.). Divisor de tensión. [Online]. Disponible en: <https://www.areatecnologia.com/electronica/divisor-de-tension.html>
- Materiales de Laboratorio. (s.f.). Voltímetro. [Online]. Disponible en: <https://materialeslaboratorio.com/voltimetro/>