



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**



Introducción a los microcontroladores  
3CM16

Practica 17

Matriz 7x5

Entrega 17-06-2023

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.....	3
Objetivo.....	4
Introducción.....	4
Material y equipo empleado .....	5
Desarrollo experimental.....	6
Circuito armado.....	6
Estructura del programa .....	7
Código CodeVisionAVR.....	7
Simulación – Proteus.....	11
Conclusiones .....	12
Bocanegra Heziquio Yestlanezi.....	12
Domínguez Durán Alan Axel.....	12
Hernández Méndez Oliver Manuel .....	12
Martínez Cruz José Antonio .....	12
Referencias.....	13



## Índice de imágenes

Imagen 1 Matriz 7x5.....	6
Imagen 2 Circuito simulado en proteus.....	11



## Objetivo

Al término de la sesión, los integrantes del equipo contarán con la habilidad para manejar una matriz de leds de 7x5.

## Introducción

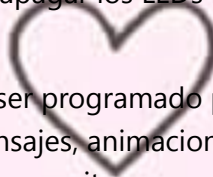
### Termómetro 0 a 50 °C (32 a 122 °F)

Una matriz de LED 7x6 es un conjunto de 42 LEDs organizados en una matriz de 7 filas y 6 columnas. Estos LEDs pueden ser controlados individualmente para mostrar patrones, caracteres, gráficos u otra información visual.

En una práctica de construcción de una matriz de LED 7x6 utilizando microcontroladores, se utiliza un microcontrolador para controlar y activar los LEDs de la matriz. El microcontrolador es un circuito integrado programable que contiene una unidad central de procesamiento (CPU), memoria y periféricos de entrada/salida, y es capaz de controlar múltiples dispositivos electrónicos[1].

El diseño de la matriz de LED implica la conexión de cada LED a un pin de salida del microcontrolador. Al manipular los pines de salida de manera adecuada, el microcontrolador puede encender o apagar los LEDs individualmente para crear patrones y visualizaciones deseadas[2].

Además, el microcontrolador puede ser programado para controlar la matriz de LED de diferentes formas, como mostrar mensajes, animaciones o incluso interactuar con otros sensores o dispositivos externos. Esto permite una amplia gama de aplicaciones, desde paneles de visualización hasta señalización, juegos y arte interactivo.



## Material y equipo empleado

- ♥ CodeVision AVR
- ♥ AVR Studio 4
- ♥ Microcontrolador ATmega 8535
- ♥ 3 Display Cátodo común
- ♥ 8 Resistores de  $330\ \Omega$  a  $1/4\ W$
- ♥ 1 Matriz de leds de 7x5

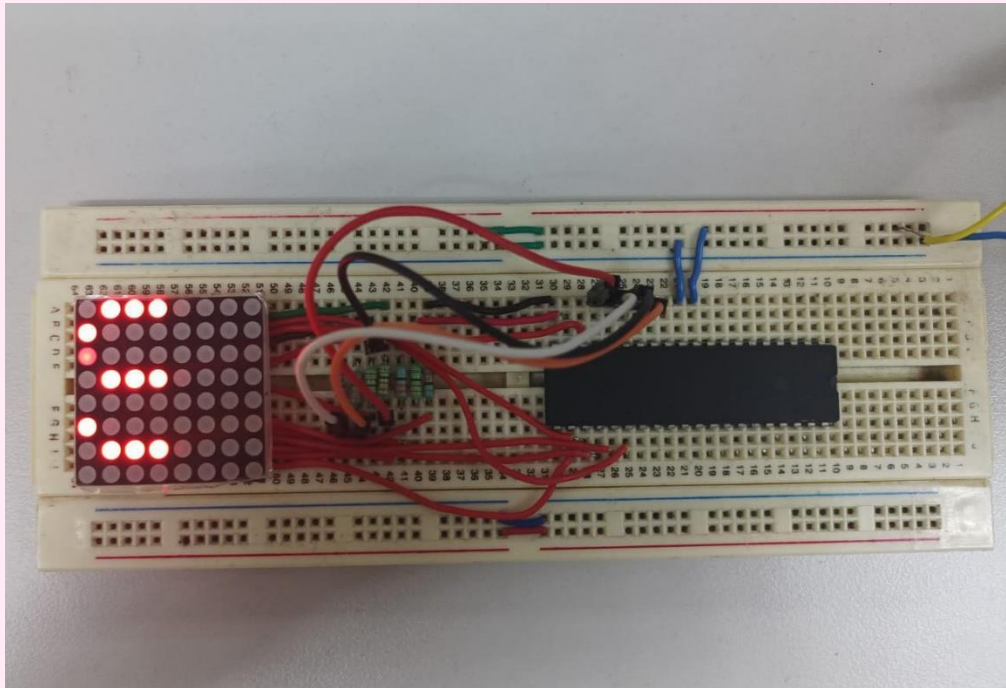


## Desarrollo experimental

Diseñe un programa para visualizar en una matriz de leds de 7x5 los número del 0 al 9 tal y como lo muestra la figura 1.

## Circuito armado

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



*Imagen 1 Matriz 7x5*

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

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

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

    PORTC=0x10;
    PORTD=0x00;
    delay_ms(10);
}
}

```

## Simulación – Proteus

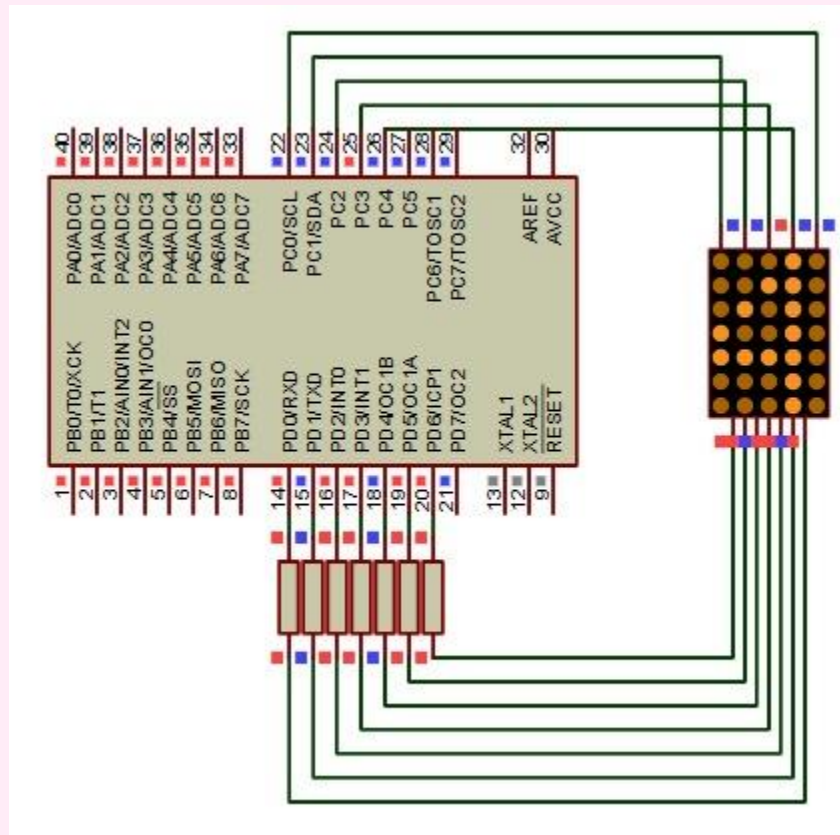


Imagen 2 Circuito simulado en proteus

## Conclusiones

### Bocanegra Heziquio Yestlanezi

En conclusión, se diseñó un programa utilizando CodeVision AVR y AVR Studio 4 para visualizar los números del 0 al 9 en una matriz de LEDs de 7x5. Para llevar a cabo esta tarea, se emplearon los siguientes materiales y equipo: microcontrolador ATmega 8535, 3 displays de cátodo común, 8 resistores de 330  $\Omega$  a 1/4 W y 1 matriz de LEDs de 7x5.

La matriz de LEDs se configuró en forma de matriz de 5 columnas por 7 filas, y cada columna se controló de manera independiente a través de un barrido. Para formar los números en la matriz, se utilizó un código que se dividió en 5 columnas, aprovechando la disposición de las filas (R) y las columnas (C) en la matriz. Se tuvo en cuenta que en las R, el LED se enciende con un valor lógico de 0 debido a la diferencia de voltaje con la columna habilitada con un valor lógico de 1 (Vcc).

Este programa permite generar y mostrar los números del 0 al 9 en la matriz de LEDs, lo que amplía las capacidades del microcontrolador ATmega 8535 y brinda una herramienta para la visualización de información de manera clara y concisa. Esta experiencia fortalece el conocimiento y las habilidades en el diseño y control de matrices de LEDs, y proporciona una base sólida para futuros proyectos que involucren visualización gráfica en este tipo de dispositivos.

### Domínguez Durán Alan Axel

Al termino de esta práctica aprendimos a cablear e implementar una matriz de leds para poder ser utilizada por el microcontrolador; en este caso imprimimos una gran cantidad de letras y figuras. La implementación de esta herramienta queda como preámbulo para poder realizar el proyecto final, ya que el problema propuesto requiere de una matriz de leds más grande. A pesar de ser un circuito relativamente simple, alambrar correctamente la matriz de leds puede llegar a ser algo tedioso.

### Hernández Méndez Oliver Manuel

En conclusión, la práctica de construir una matriz de LED 7x6 utilizando microcontroladores ofrece un amplio potencial creativo y funcional. El uso de microcontroladores permite un control preciso de cada LED en la matriz, lo que brinda la posibilidad de crear patrones, mensajes y visualizaciones personalizadas. Esta práctica puede ser aplicada en una variedad de campos, como señalización, paneles de visualización, juegos y arte interactivo. La combinación de la matriz de LED y los microcontroladores proporciona flexibilidad y versatilidad en el diseño, permitiendo adaptar el sistema a diferentes necesidades y aplicaciones específicas.

### Martínez Cruz José Antonio

Esta práctica conto con dos partes, la primera era mostrar diferentes sucesiones que nos permitiera observar cómo se comporta los leds en situaciones particulares. Para la segunda parte, el desarrollo de una marquesina nos permitió como es que se hacen este tipo de

movimientos y permitirnos observar mayor contenido dentro de mismo espacio de la matriz. Aunque al momento de conectar los pins de la matriz de leds, logramos mostrar de manera satisfactoria las dos partes de esta práctica.

## Referencias

- [1] "Design and Implementation of a 7x6 LED Matrix Display System using Microcontroller" por Chitrang Dixit, Bhargav Bhatnagar y Saurabh Kapoor. Puedes encontrar este artículo en la revista International Journal of Science, Engineering and Technology Research (IJSETR). Ofrece detalles sobre el diseño y la implementación de un sistema de visualización con matriz de LED 7x6 utilizando un microcontrolador.
- [2] "Building a 7x6 LED Matrix Display with a Microcontroller" por Jeremy Blum. Este es un proyecto disponible en el sitio web "Exploring Arduino". Proporciona un tutorial paso a paso sobre cómo construir una matriz de LED 7x6 utilizando un microcontrolador Arduino. Incluye diagramas de conexión, código fuente y explicaciones detalladas.

