



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



Sistemas en Chip

Práctica 17

“Matriz 7x5”

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 para manejar una matriz de leds de 7x5.

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.

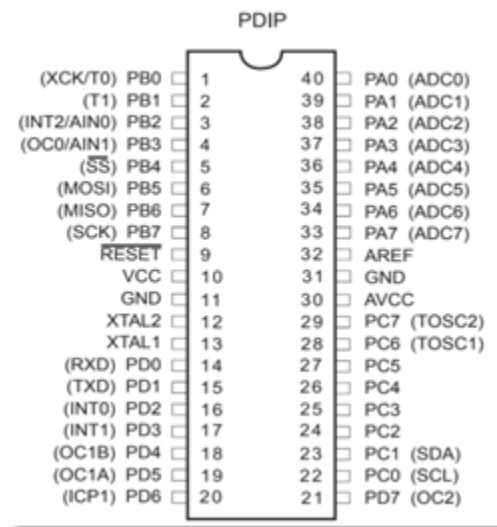


Figura 1. Configuración de pines ATmega8535.

El microcontrolador utiliza una arquitectura cerrada, es decir, aquel que es inmodificable por los programadores ajenos a la compañía propietaria del código fuente. Por lo tanto, a este sistema no se le pueden colocar dispositivos periféricos, solo se usa el hardware de la compañía propietaria ya que los dispositivos ajenos a dicha compañía no son compatibles.

El microcontrolador ATMEGA8535 utiliza un encapsulado DIP-40, común en la construcción de circuitos integrados que consiste en un bloque con dos hileras paralelas de pines, observar la figura 2.



Figura 2. Microcontrolador atmega853.

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.

El microcontrolador se alimenta de las terminales 10 y 11 como lo muestra la figura 1, los cuales son el VCC (5 V y una tolerancia de $\pm 0.5V$) y GND. Sin embargo, el convertidor se alimenta de forma externa en la terminal 31 y 32, los cuales son GND y AREF. 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.

Matriz de leds 7x5

La matriz de leds de 7x5 no es más que un arreglo de LEDs agrupados dentro de un encapsulado, los cuales se encuentran agrupados en forma de matriz. El Display de matriz de puntos 7X5 de led ideal para mostrar mensajes, imágenes, caracteres o símbolos. Fácilmente controlable mediante Arduino, PIC, AVR, ARM, microcontrolador, entre otras.

Matriz de LED de 7 x 5, 35 LEDs ideal para proyectos electrónicos. Una matriz de LEDs consiste en un arreglo de LEDs que pueden ser encendidos y apagados individualmente desde un microcontrolador es como una pantalla de pocos pixeles en los cuales pueden presentar gráficos y textos, tanto estáticos como en movimiento y hacer tus proyectos o trabajos electrónicos.

La Matriz de Led 7 x 5 consume una baja cantidad de corriente, además con poco voltaje te permitirá crear letreros luminosos que puedes adaptar para tus proyectos escolares y prototipos.

Aplicaciones

- Pueden ser encendidos y apagados individualmente desde un microcontrolador
- Aplicaciones: Para proyectos escolares.

Especificaciones

- Tipo: Matriz 5 x 7
- Polaridad: Ánodo común a renglón
- Número de LEDs: 35
- Tipo de LEDs: 5 mm
- Voltaje de alimentación: 3 V
- Corriente de alimentación: 30 mA
- Tipo de luz del LED: Ultra brillante
- Número de columnas: 5
- Número de renglones: 7
- Dimensiones: 2.3 cm x 4 cm x 0.7 cm
- Color de la luz del LED: Rojo
- Modelo: GMM-12057ASB

Materiales y Equipo empleado

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

Desarrollo Experimental

1.- Diseñe un programa para visualizar en una matriz de leds de 7x5 los números del 0 al 9 tal y como lo muestra la figura 3.

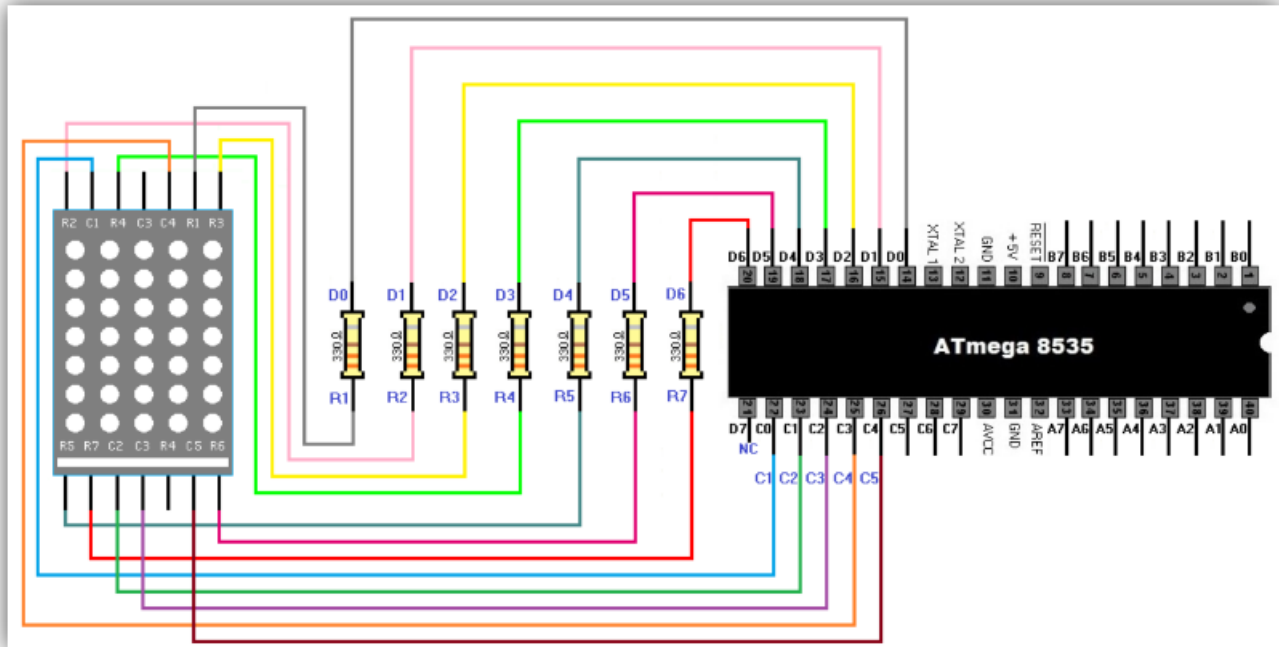


Figura 3. Circuito para conectar las Filas (D0-D7) y las Columnas (C0-C4) de la matriz 7x5.

Matriz de LEDs

La matriz de LEDs no es más que un arreglo de LEDs agrupados dentro de un encapsulado, los cuales se encuentran agrupados en forma de matriz. Este acomodo nos ayuda para poder generar cualquier cosa que nosotros queramos siempre y cuando se pueda representar dentro de la matriz, observar la figura 4.

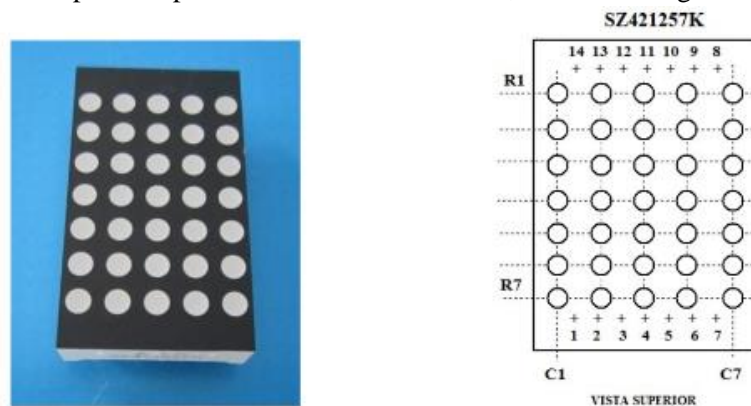


Figura 4. Vista real de la matriz 7x5.

La matriz de LEDs que se usara en este ejemplo es una como la de la foto superior, esta es de 5 columnas por 7 filas, las columnas son representadas por una C y las filas por una R, en la imagen inferior (figura 5), podemos ver como se encuentran distribuidos los pines de la matriz a usar.

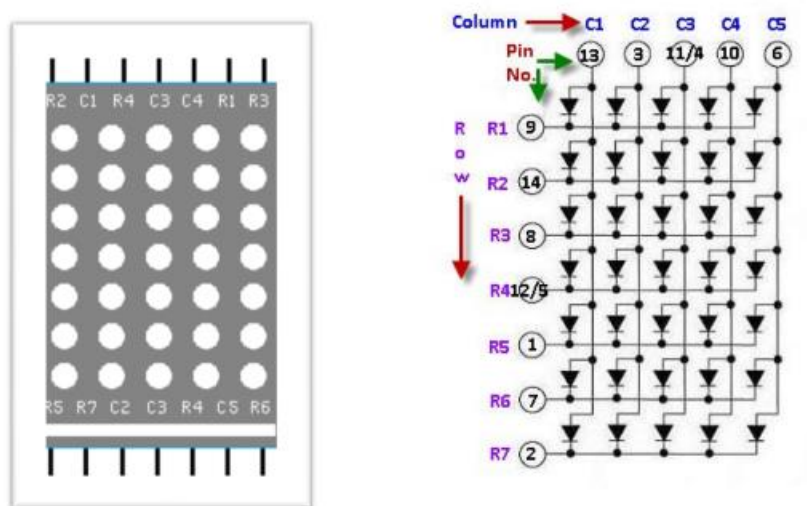


Figura 5. Asignación de pines de la matriz 7x5.

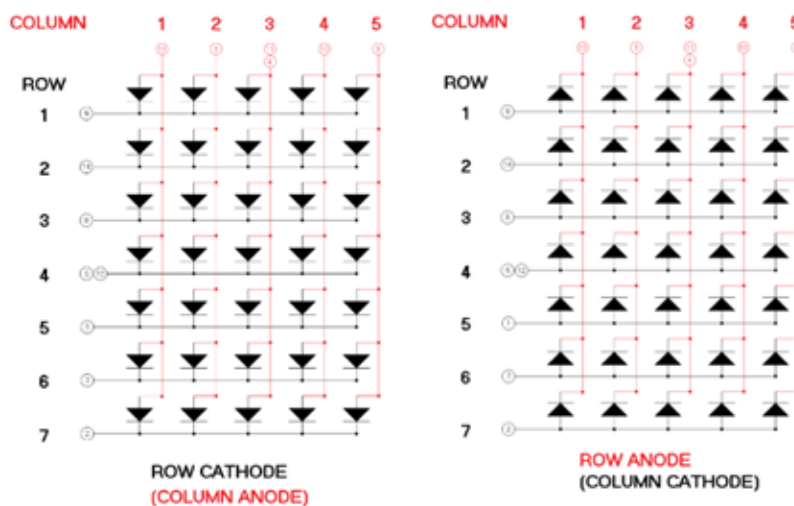


Figura 6. Conexiones para la matriz Fila Cátodo-Columna Ánodo y conexiones para la matriz Fila Ánodo-Columna Cátodo.

Actividad 1

Genere la siguiente secuencia en la matriz para que se observen las columnas como muestra la figura 7.

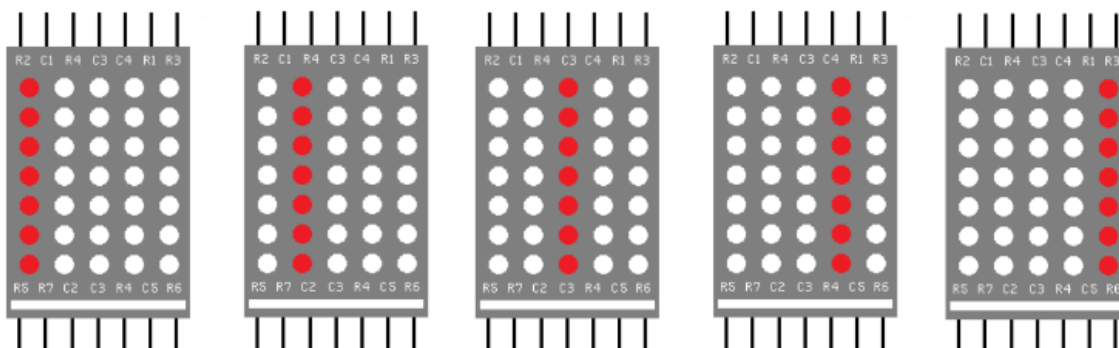


Figura 7. Visualización de las columnas.

Actividad 2

Genere la siguiente secuencia en la matriz para que se observen las filas como muestra la figura 8.

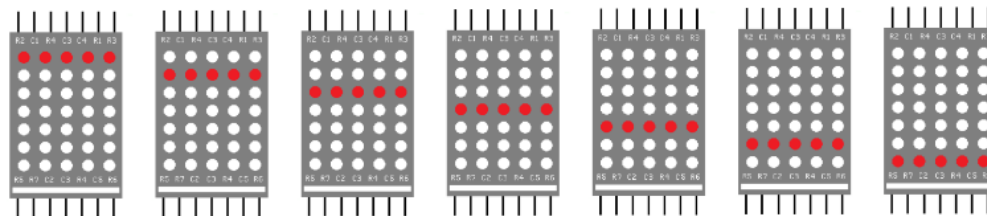


Figura 8. Visualización de las filas.

Actividad 3

Genere la siguiente secuencia en la matriz para que se observen las filas como muestra la figura 9.

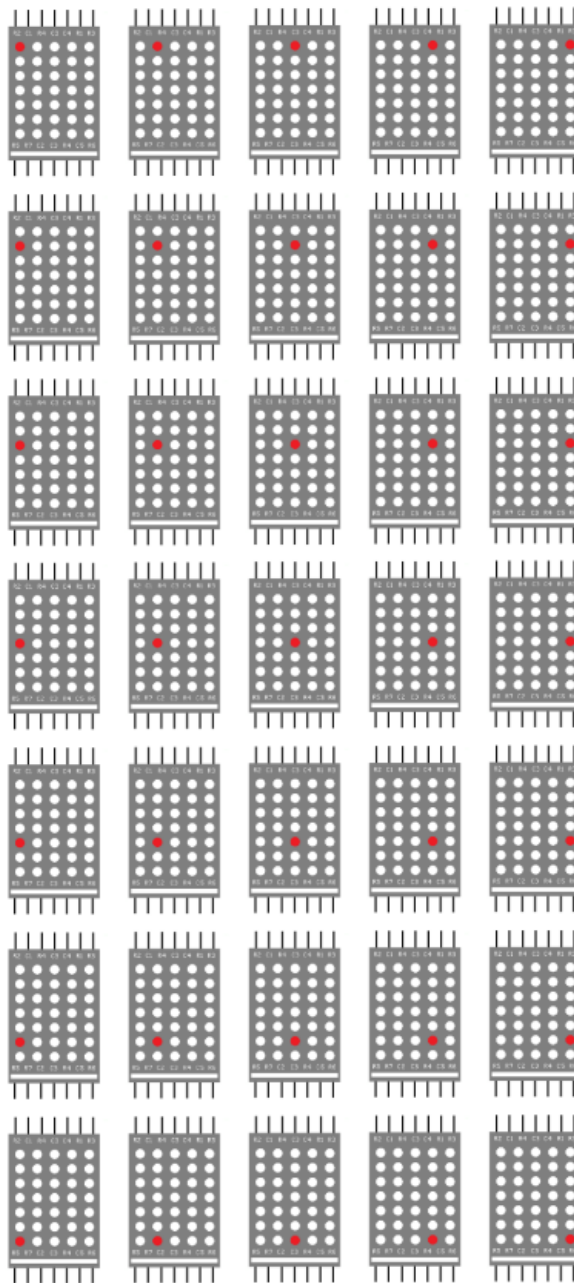


Figura 9. Visualización de todos los LEDs.

Actividad 4

Para poder formar algo en la matriz, es necesario realizar un barrido en las columnas para controlarlas de manera independiente, cada columna tendrá su código, por lo que debemos formar la figura, número o letra que necesitemos separando la misma en 5 columnas. A continuación, veremos cómo se forma el número 3 en la figura 10, el cual prestando atención al valor de las R's las cuales forman el código deseado mientras que las C's generan un barrido de las columnas. Es importante destacar que en las R's el LED prende con un 0 lógico, lo cual está dado ya que la columna correspondiente está habilitada con un 1 lógico (Vcc), el led en las R's prendera con la diferencia de voltaje, por lo tanto, en las R's se usa el 0 como prendido.

Ejemplo de la formación de un número 3.

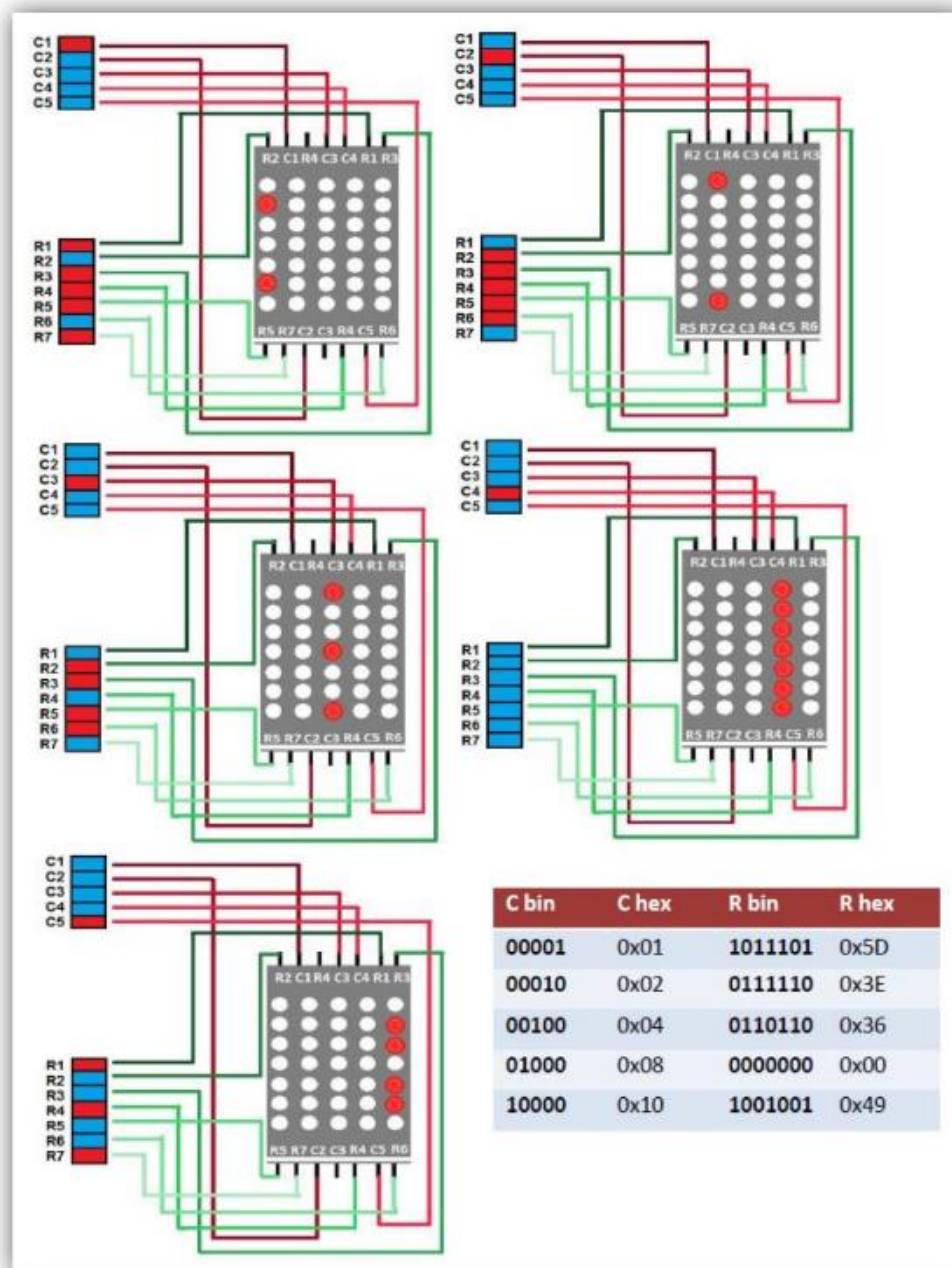


Figura 10. Ejemplo de formación del número 3.

Como se puede ver en la imagen anterior, el número 3 se formó en base a la combinación de controlar las C's y las R's, trabajo que le asignaremos al microcontrolador, este proceso se repetirá varias veces a una velocidad lo suficientemente alta, como para no alcanzar a percibir los cambios, y tener la idea de que todos los LEDs deseados se encuentran prendidos a la vez. Después de hacer un proceso similar, se obtuvo una tabla con los valores de cada número deseados, en este caso los números se crearon de determinada forma, la cual puede cambiar dependiendo las necesidades de cada persona, ya sea que se necesite mostrar letras o caracteres distintos.

Número	C1	C2	C3	C4	C5
0	0X41	0X2E	0X36	0X3A	0X41
1	0X3F	0X3D	0X00	0X3F	0X3F
2	0X3D	0X1E	0X2E	0X36	0X39
3	0X5D	0X3E	0X36	0X36	0X49
4	0X67	0X6B	0X6D	0X00	0X7F
5	0X58	0X3A	0X3A	0X3A	0X46
6	0X43	0X35	0X36	0X36	0X4F
7	0X7C	0X7E	0X0E	0X76	0X78
8	0X49	0X36	0X36	0X36	0X49
9	0X79	0X66	0X66	0X56	0X69

Tabla 1. Tabla para generar los números del 0 al 9 para la matriz 7x5.

Estructura del programa

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

```
#include <mega8535.h>
#include <delay.h>
#include <math64.h>

int i = 0;
int j = 0;
int k = 0;
long cont = 0;
const char secuencia3_columna[5] = {0x01,0x02,0x04,0x08,0x10};
const char secuencia3_fila[7] = {0x7e,0x7d,0x7b,0x77,0x6f,0x5f,0x3f};
const char numeros[10][5] = {
    {0x41,0x2e,0x36,0x3a,0x41},
    {0x3f,0x3d,0x00,0x3f,0x3f},
    {0x3d,0x1e,0x2e,0x36,0x39},
    {0x5d,0x3e,0x36,0x36,0x49},
    {0x67,0x6b,0x6d,0x00,0x7f},
    {0x58,0x3a,0x3a,0x3a,0x46},
    {0x43,0x35,0x36,0x36,0x4f},
    {0x7c,0x7e,0x0e,0x76,0x78},
    {0x49,0x36,0x36,0x36,0x49},
    {0x79,0x36,0x36,0x36,0x41}
};
```



```
const char escum[30] =
{0x00,0x36,0x36,0x36,0x36,0x7f,0x31,0x36,0x36,0x36,0x46,0x7f,0X41,0X3e,0X3e,0X3e,0X3e,0
x7f,0X41,0X3e,0X3e,0X3e,0X41,0x7f,0x00,0x7d,0x7b,0x7d,0x00,0x7f};

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=(1<<PORTA7) | (1<<PORTA6) | (1<<PORTA5) | (1<<PORTA4) | (1<<PORTA3) | (1<<PORTA2)
| (1<<PORTA1) | (1<<PORTA0);
// Port B initialization
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1)
| (0<<DDB0);
PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) | (1<<PORTB2)
| (1<<PORTB1) | (1<<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=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) | (1<<DDD1)
| (1<<DDD0);
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2)
| (0<<PORTD1) | (0<<PORTD0);
...
...
...
while (1)
{
//Primera secuencia
PORTD = 0x00;
for(j = 0; j < 5; j++){
PORTC = secuencia3_columna[j];
delay_ms(100);
}
//Segunda secuencia
PORTC = 0x1f;
for(j = 0; j < 7; j++){
PORTD = secuencia3_fila[j];
delay_ms(100);
}
//Tercera secuencia
for(j = 0; j < 7; j++){
```

```
        PORTD = secuencia3_fila[j];
        for(i = 0; i < 5; i++){
            PORTC = secuencia3_columna[i];
            delay_ms(100);
        }
    }
    //Cuarta secuencia
    for(i = 0; i < 10; ){
        for(j = 0; j < 5; j++){
            PORTC = secuencia3_columna[j];
            PORTD = numeros[i][j];
            delay_ms(5);
        }
        cont++;
        if (cont == 10){
            i++;
            cont = 0;
        }
    }
    //Quinta secuencia
    for(i = 0; i < 25; ){
        for(j = 0; j < 5; j++){
            PORTC = secuencia3_columna[j];
            PORTD = escum[j+k];
            delay_ms(5);
        }
        cont++;
        if (cont == 10){
            i++;
            k++;
            cont = 0;
        }
        if(k == 25)
            k = 0;
    }
}
```

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.

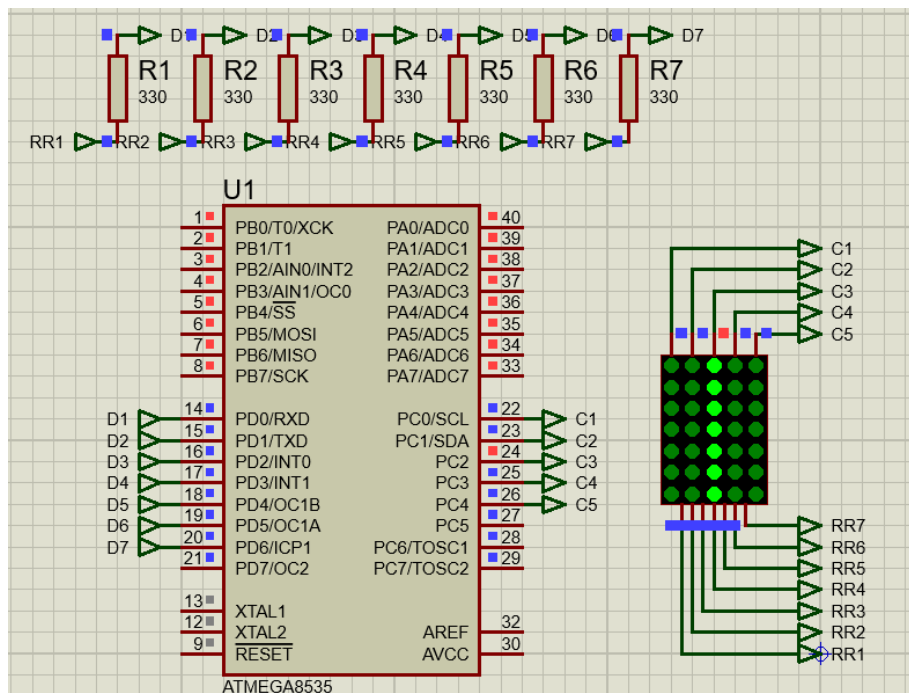


Figura 11. Simulación de la Actividad 1.

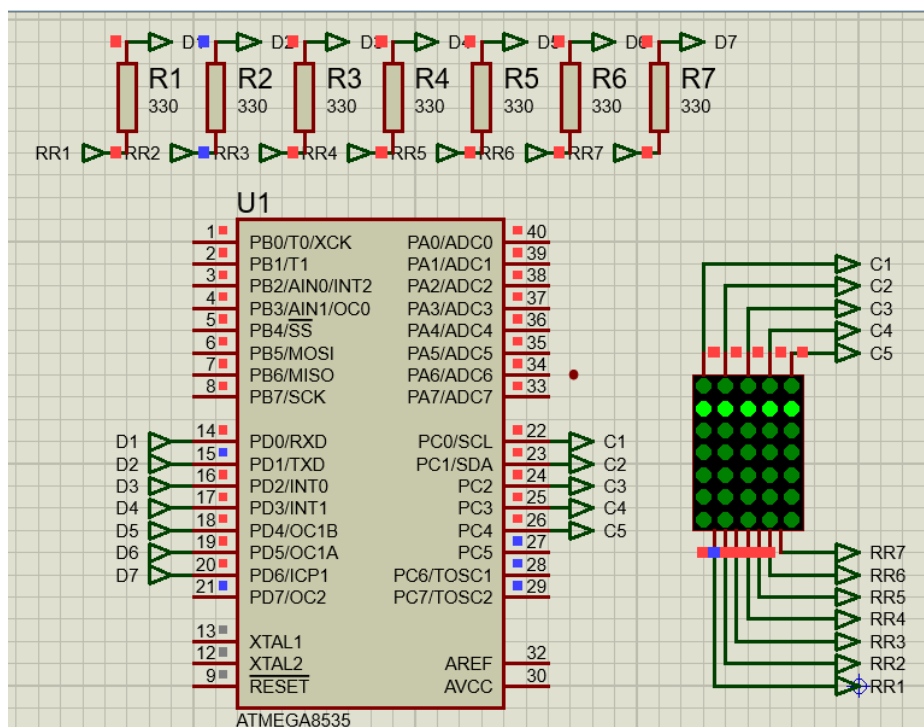


Figura 12. Simulación de la Actividad 2.

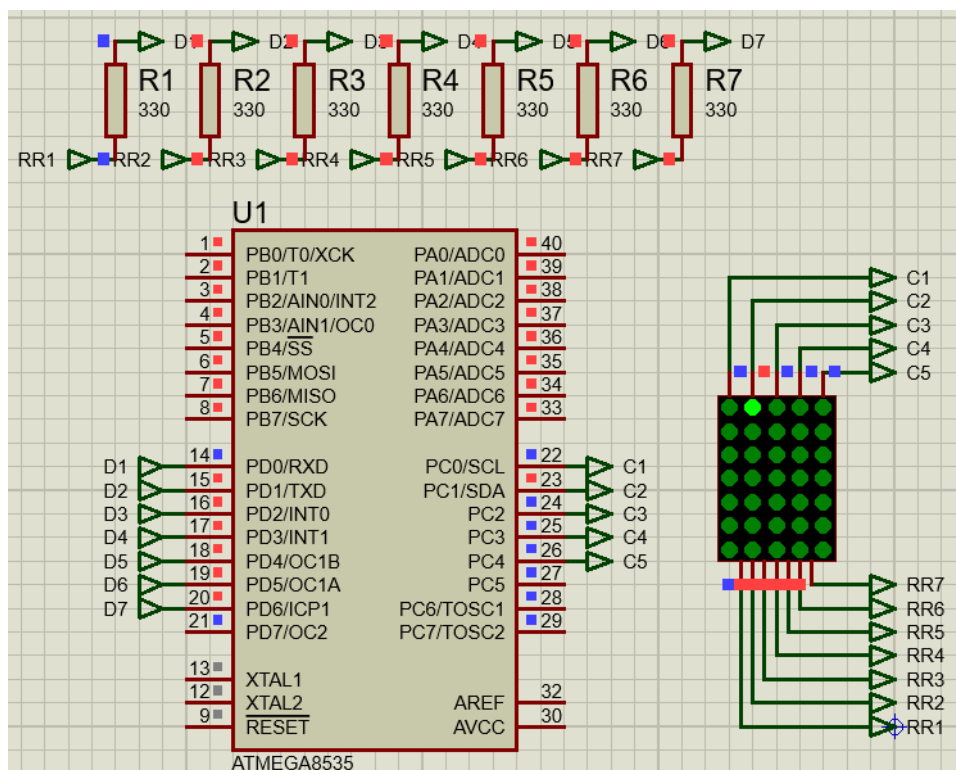


Figura 13. Simulación de la Actividad 3.

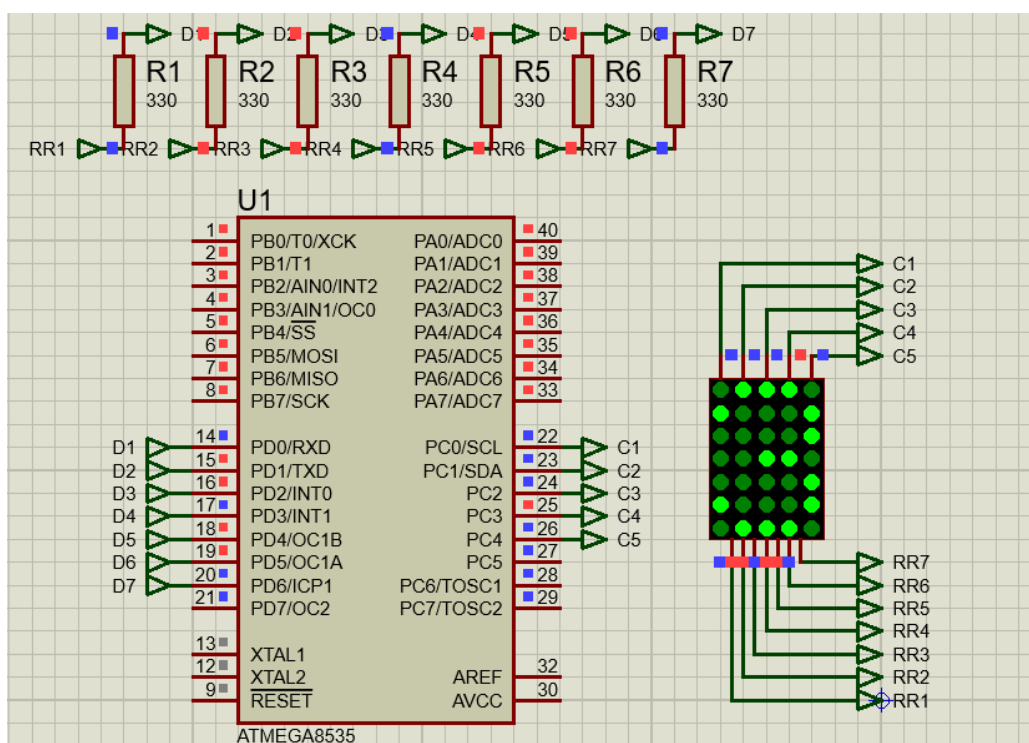


Figura 14. Simulación de la Actividad 4.

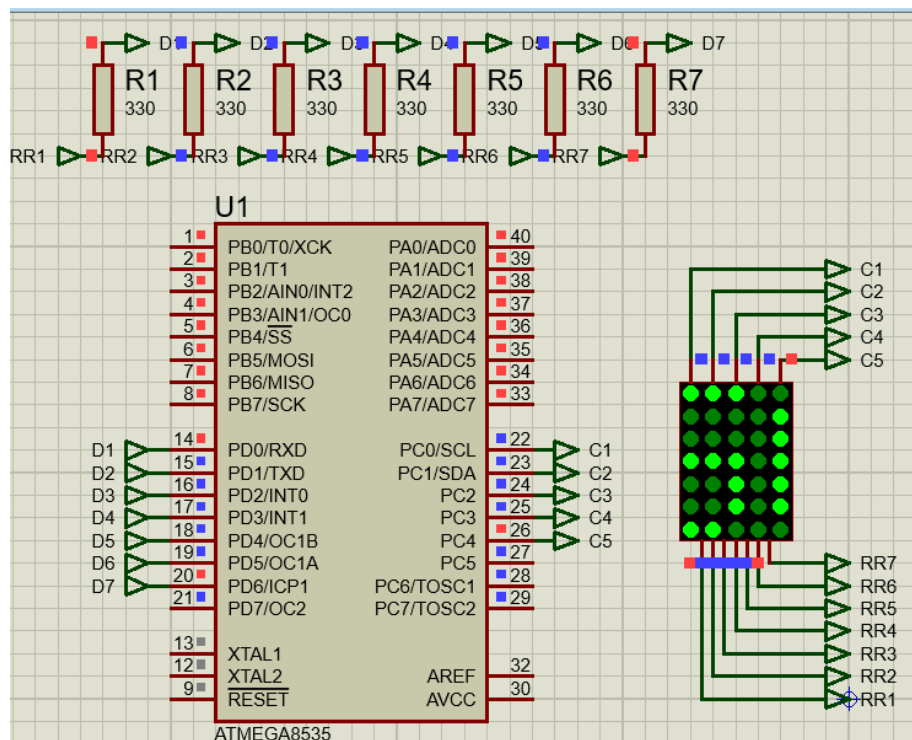


Figura 15. Simulación de la Actividad 5.

Fotografía del circuito armado

A continuación, se muestra la figura 16 la evidencia sobre la realización y prueba de la practica 17.

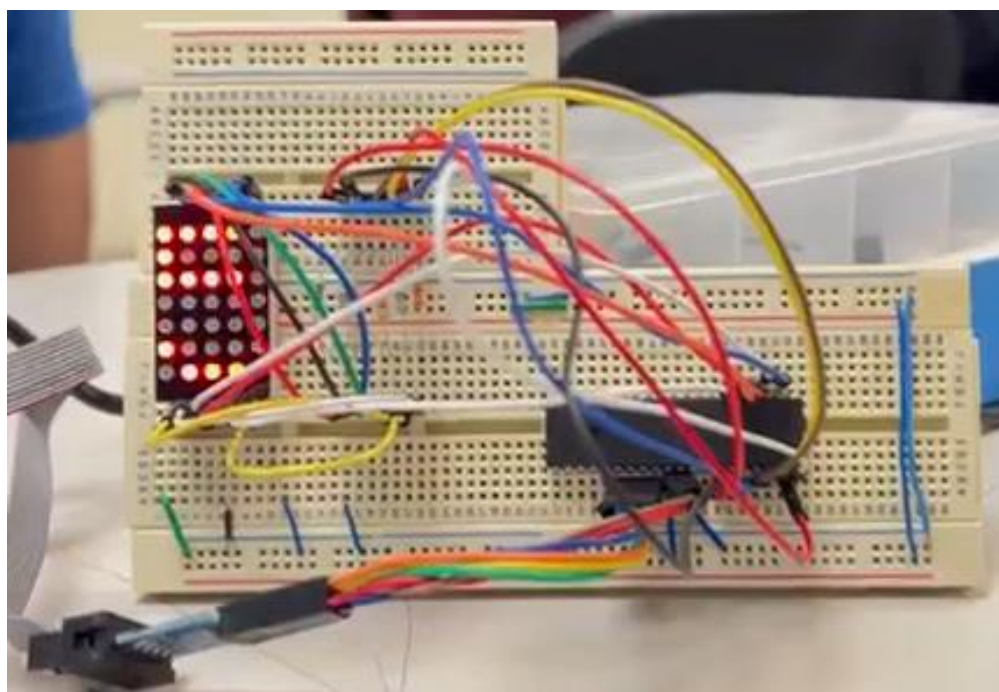


Figura 16. Circuito armado y funcionamiento de la práctica 17.

Observaciones y conclusiones Individuales

Ocampo Téllez Rodolfo

Esta práctica sirvió para familiarizarnos con el funcionamiento de una matriz de leds 7x5, y aunque se tuvieron que realizar 5 distintas secuencias, el desarrollo del código fue relativamente sencillo una vez que se comprendió el procedimiento de encendido de LEDs en la matriz por medio de agrupación en nibbles. Una vez que se establecieron todos los arreglos que contenían todos los valores hexadecimales necesarios, cada secuencia se ejecutó mediante ciclos for, en las dos últimas se necesitó anidar ciclos y reducir el valor de delay, en la primera puesto que se necesitaba encender todas las columnas para formar un número y en el segundo porque además se necesitaba dar sensación de movimiento.

Patlani Mauricio Adriana

Esta práctica sirvió bastante para aprender y comprender el manejo de una matriz de leds de 7x5, el armado del circuito fue sencillo, aunque el manejo no tanto, ya que fue algo tardado en realizar las 5 secuencias, pero finalmente se comprendieron gracias a los recursos dados en clase. Fue el saber manejar filas y columnas para realizar las secuencias

Ruvalcaba Flores Martha Catalina

El desarrollo de esta práctica fue relativamente sencillo, pues en esta se logra entender el funcionamiento de la matriz de 7x5, y gracias a la página proporcionada por el profesor sobre el generador matricial, se pudo configurar y programar las filas y columnas de forma sencilla para la elaboración de las actividades solicitadas en dicha práctica.

Sandoval Hernández Eduardo

El circuito de esta práctica no representó mayor problema al momento de armarlo, lo complicado fue al momento de codificar la práctica pues se tuvo que pensar demasiado en cómo hacer las letras y números en la matriz de leds, y posteriormente como hacer las secuencias que se solicitaban para la práctica, aun así, se pudo solucionar y como resultado la matriz muestra sin mayor problema cada una de las secuencias. Esta práctica además sirvió para darnos una idea de la práctica 20.

Bibliografía

F. Aguilar. (2020, Diciembre 12). 17 Práctica de Matriz 7x5. Introducción a los Microcontroladores. [Online]. Disponible en: <https://youtu.be/9jr5PDJ-7Yc>

- 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. Cereijo. (2004, Diciembre 28). Microcontroladores PIC. [Online]. Disponible en: https://drive.google.com/file/d/1mAydVlkZYqtYXhFQ0Tnr_UjZyjiwIp90/view

- F. Aguilar. (Mayo, 2020). Microcontroladores Matrix 7x5. [Online]. Disponible en: <https://youtu.be/9jr5PDJ-7Yc>
- La Red Electrónica. (s. f.). Matriz LED Rojo 5X7 5mm. [Online]. Disponible en: <https://laredelectronica.com/producto/matriz-de-de-led-rojo-7x5-3mm/>
- Carrod. (s. f.). Matriz de LED 5 mm Anodo de 5 x 7. [Online]. Disponible en <https://www.carrod.mx/products/matriz-de-led-5-mm-anodo-de-7x5-gmm-12057asb>