Parcial I: Informe

Informática II

Por:

Raul Daza Liñan.

Luis David Muñoz Jurado.

2023

Departamento de Ingeniería electrónica y telecomunicaciones

Facultad de Ingeniería

Universidad de Antioquia

Análisis del problema y consideraciones

El problema a resolver es realizar un sistema que permita mostrar ciertos patrones en una matriz de 8x8, para solucionarlo se hará uso de la plataforma Tinkercad hacer la simulación del sistema, en la simulación a grandes rasgos se hará uso de la plataforma Arduino Uno, leds para matriz 8x8 y el integrado 74HC595 para la activación y desactivación de los leds de la matriz.

Para llevar a cabo el proyecto se dividirá en 5 etapas:

1. Comprensión de implementación matriz 8x8.
2. Funcionamiento 74HC595.
3. Diagrama de flujo.
4. Codigos en c++.
5. Implementación de los códigos y circuitos Tinkercad.

Comprensión de implementación matriz 8x8.

Las dos configuraciones típicas se pueden ver en la figura 1. En anodo común para encender un led se envía un voltaje positivo en la respectiva fila y tierra en la respectiva columna, mientras que en catodo común es lo contrario, tierra en la respectiva fila y un voltaje positivo en la respectiva columna.

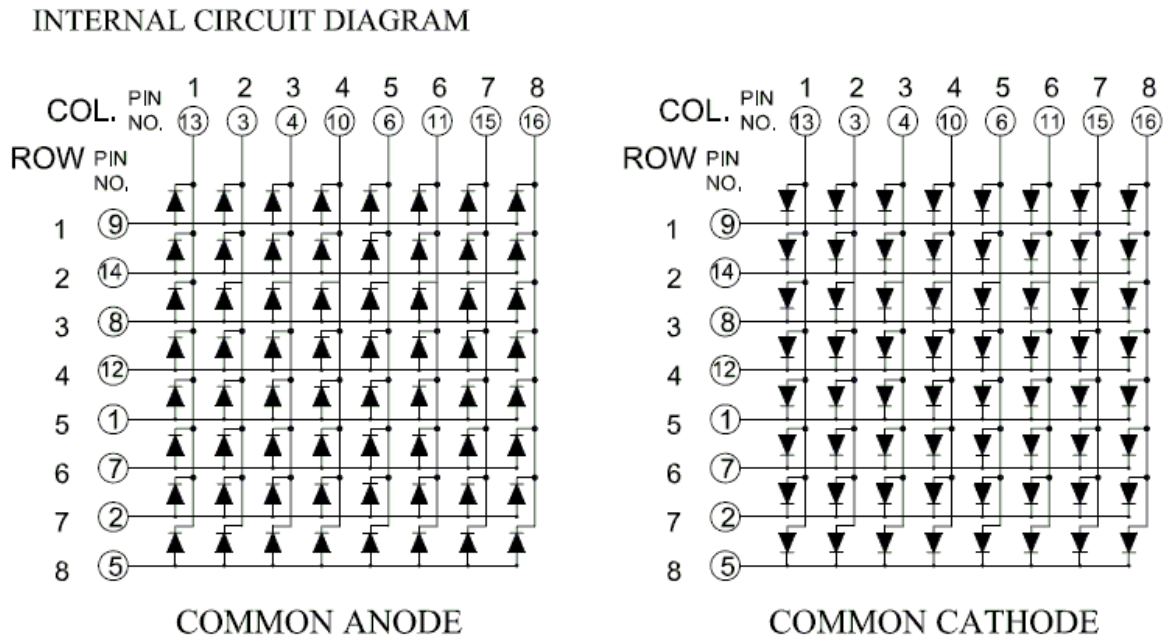

Figura 1. Configuración Ánodo común y Cátodo común.

Funcionamiento 74HC595.

El integrado 74HC595 es un registro de desplazamiento de 8 bits, su funcionamiento consiste en ingresar de forma serial informacion y desplegarla en paralelo, es util para implementarlo en sistemas donde se tiene que desplegar varios bits en paralelo pero se está limitado en la cantidad de pines que dan salida de informacion.

En la figura 2 se puede ver la configuración de pines del integrado. VCC es el pin de alimentacion, puede ir de 2v a 6v, Serial/Data (DS) es el pin por donde ser recibe la informacion secuencialmente, Ouput Enable (OE) es el pin que habilita la salida de datos, Latch (ST) es el pin que actualiza los datos de salida, SH (Clock) es el pin que actualiza los datos de que van guardando serialmente, Master Reset (MR) es el pin que resetea los datos guardados, QX son los pines de salida, Q7' es un pin que permite conectar varios 74HC595. Cabe resaltar que OE y MR estan negadas.

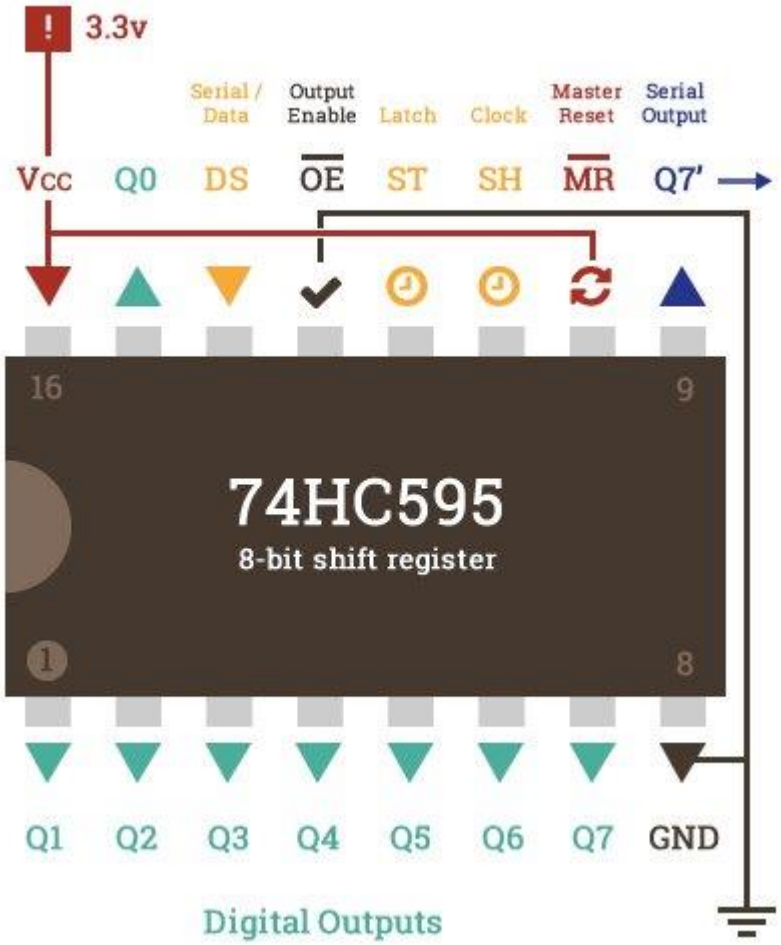

Figura 2. Integrado 74HC595.

Diagrama de flujo.

El diagrama de flujo se presenta en la figura3.

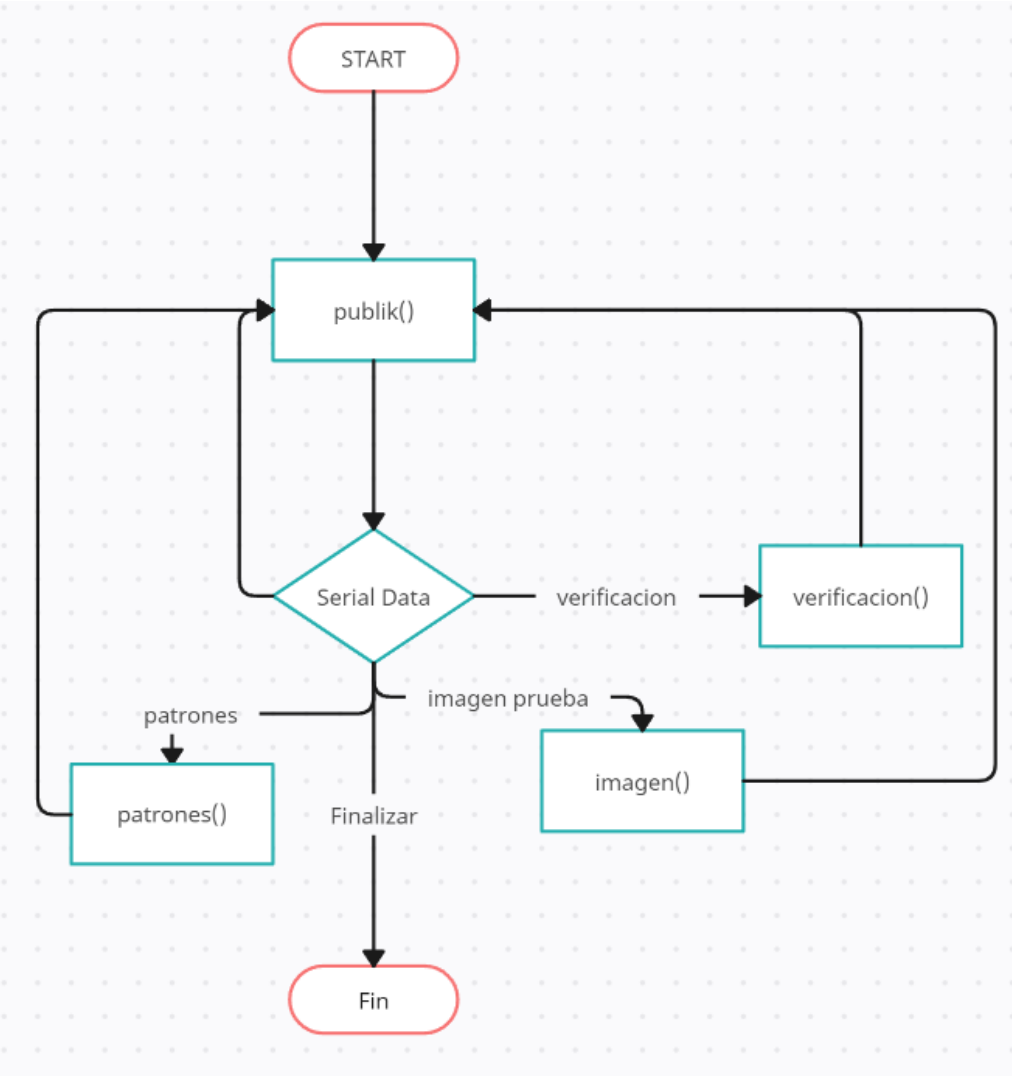

Figura 3. Diagrama de flujo del programa.

1. Primero se ejecuta la función publik () que se encarga de pedir al usuario que es lo que desea hacer.
2. Se ejecuta la función correspondiente a lo que eligió el usuario.
   - Verificación: ejecuta la función verificación (), que permite verificar que los leds de la matriz están funcionando correctamente. Los leds se encenderán y apagarán cada cierto tiempo.
   - Imagen prueba: ejecuta la función imagen (), esta le pide al usuario que ingrese un patrón a mostrar en la matriz y luego la muestra en la matriz de leds.
   - Patrones: ejecuta la función patrones (), esta muestra de forma alternada 4 patrones diferentes.
   - Finalizar: Termina la ejecución del programa.
3. Si no se recibe el comando Finalizar al terminar la ejecución de la función que realiza lo pedido por el usuario se vuelve a la función publik () para que el usuario vuelva a ingresar otro comando para realizar alguna función. Si el usuario ingresa un comando invalido también volverá a la función publik.

Procedimiento.

Se opto por trabajar la secuencia ánodo común, es decir que un led se encenderá cuando el ánodo este en 1 y el cátodo este 0.
Se implementaron 2 integrados 74HC595, con los cuales se controla las 8 filas y 8 columnas respectivamente, por lo tanto, todos los leds de cada fila fueron conectados por sus ánodos y cada fila será controlada por cada salida digital del integrado teniendo en cuenta las posiciones de mayor potencia, similarmente, todos los cátodos de cada columna fueron conectados entre sí, y además fue necesario agregar una resistencia a cada columna de la matriz para así evitar que el circuito se quemase, el valor de esta resistencia fue de 600 ohm.

Los integrados se conectaron entre si mediante la salida invertida del integrado que controla los cátodos y la entrada del integrado que controla los ánodos y así usar solamente 3 pines digitales del Arduino para cumplir con las especificaciones de la práctica.

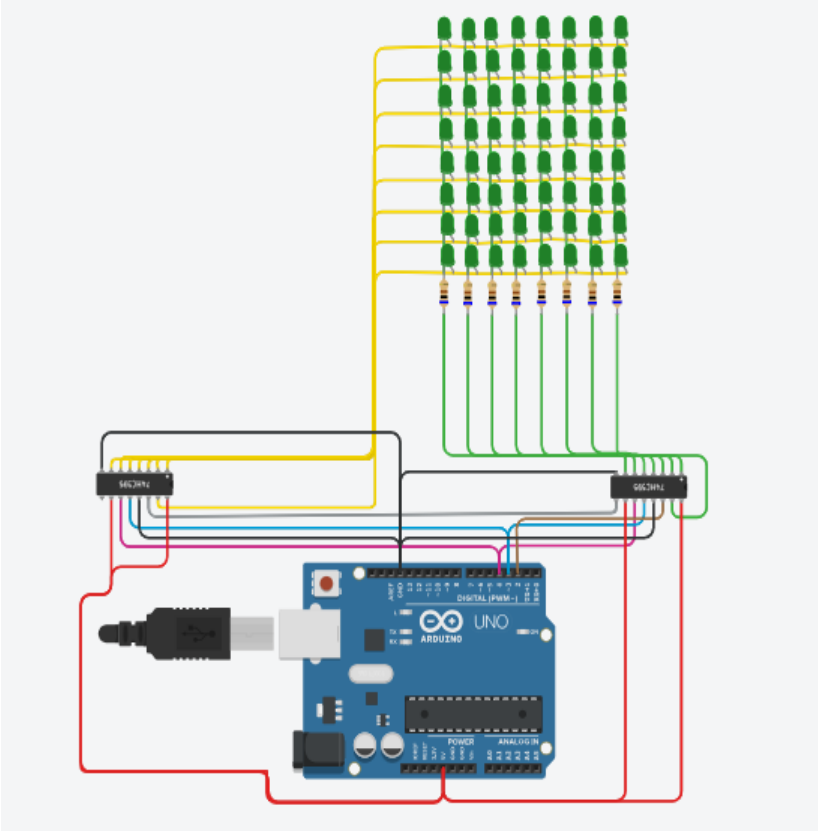

Figura 4. Montaje en Tinkercad.

Memoria dinámica: Para realizar los diferentes patrones de visualización y se creó una matriz dinámicamente en memoria utilizando punteros a punteros para almacenar la dirección de memoria de la matriz, se utiliza el comando new para reservar memoria en el heap. Posteriormente cuando se deje de usar la matriz es necesario liberar la memoria que fue reservada y se realizo mediante el comando delete.

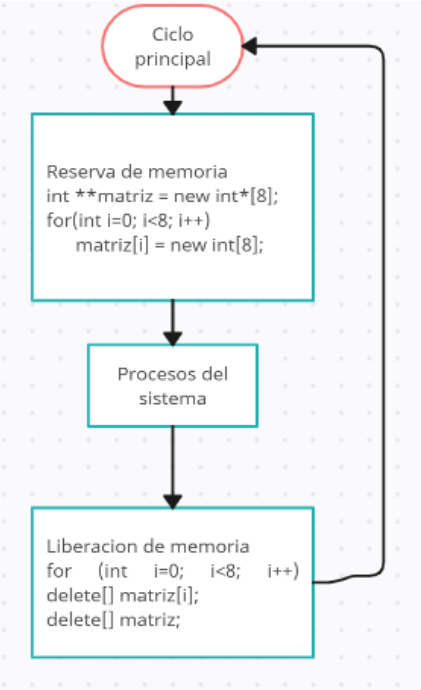

Figura 4. Memoria dinámica.

Función publik (): La función publik se la realizo tipo entero para así poder retornar un valor ingresado por el usuario. Esta función muestra un menú en pantalla para interactuar con el usuario y a la vez este pueda escoger la opción correspondiente a la tarea que quiera realizar entre las tareas disponibles por el sistema.

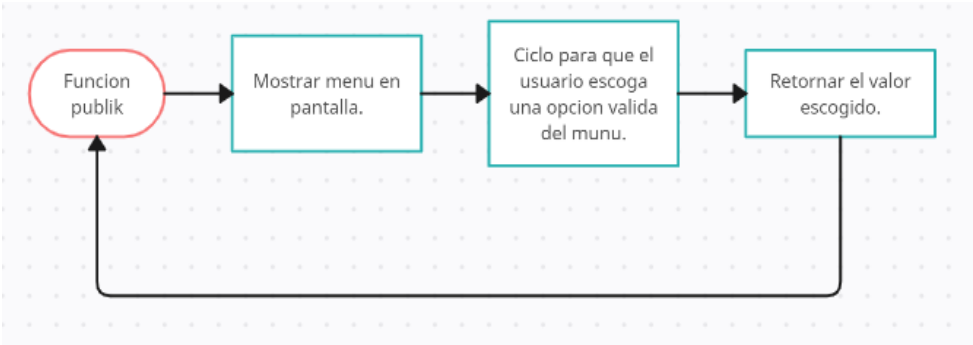

Figura 5. Función publik.

Función H595: Esta función tipo void que recibe dos datos tipo byte decimales los cuales corresponden al ánodo y cátodo respectivamente y distribuye la información (1-0) a cada fila y cada columna según corresponda a los leds que se van a encender o apagar.
Estos valores decimales corresponden en ánodo y cátodo a la suma de las potencias de cada fila y columna respectivamente donde se van a encender leds teniendo en cuenta que se trabajó con ánodo común.

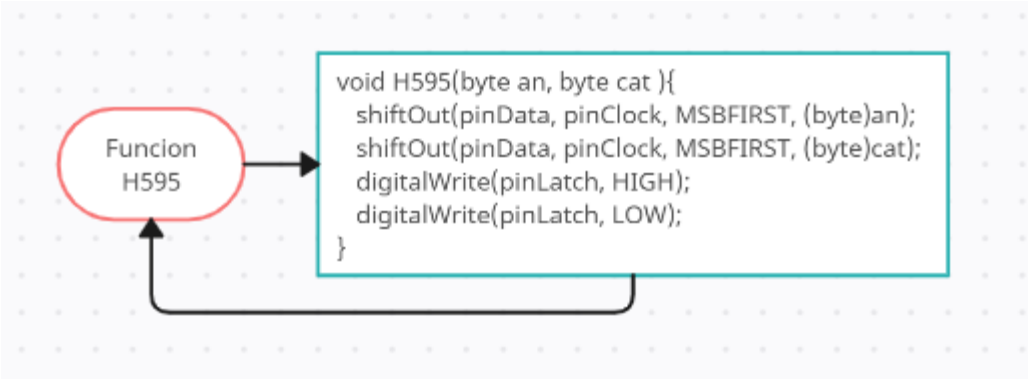

Figura 6. Función H595.

La ShiftOut es una función propia del integrado que hace un corrimiento hacia la izquierda con el bit más significativo (MSBFIRST) del valor del ánodo o cátodo que se le ingrese respectivamente, cada corrimiento es almacenado en las posiciones digitales del integrado respectivamente cuando el pinClock este en alto, una vez este almacenado todo el corrimiento en todos los pines del integrado, es decir información almacenada (pinData), se procede a desplegar esa información simultáneamente de los dos integrados hacia la matriz de leds, este proceso se realiza poniendo el pin de despliegue de información en alto (digitalWrite(pinLatch)) y luego es necesario dejar de desplegar información para desplegar la información de las siguientes filas.

Función vector2sum: Es la función que se encarga de calcular la suma de filas y columnas para llamar a la función H595 y mostrar los leds encendidos el tiempo que especifica el usuario.

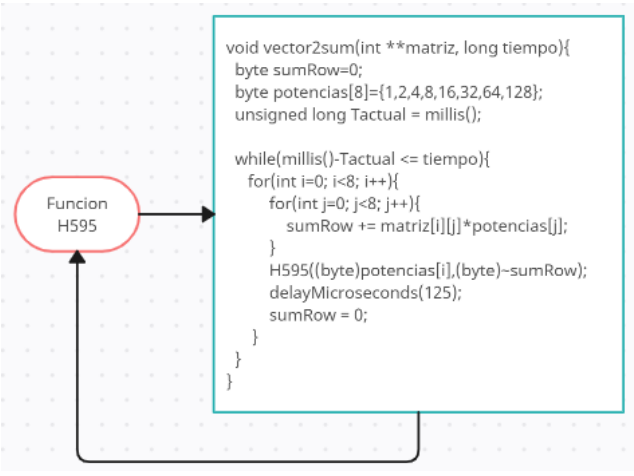

Figura 6. Función vector2sum.

En esta función se crea un arreglo de 8 posiciones que contiene las potencias de 2 correspondientes a cada posición del arreglo o a su vez a cada posición de las filas de la matriz, este arreglo nos permite calcular la suma de cada fila en la matriz.
Se emplea la función millis () que es propia de Arduino, esta función da el tiempo en milisegundos desde que arranco el programa hasta donde esta sea invocada, esta función se emplea en un ciclo para controlar el tiempo que permanezca encendida la matriz, ese tiempo es digitado por el usuario.

Mediante 2 ciclos anidados se recorre la matriz para calcular la suma en cada fila, posteriormente se llama la función H595 para pasarle los parámetros correspondientes al ánodo y cátodo, la suma se la pasa en la posición del cátodo y de manera negada ya que se necesita tener en cero los cátodos de los leds que van a encender.

Función verificación: La función verificación se dividió en 2 funciones por separado donde cada función prede y apaga los leds de la matriz respectivamente, esta función a su vez llama la función vector2sum para encender la matriz.

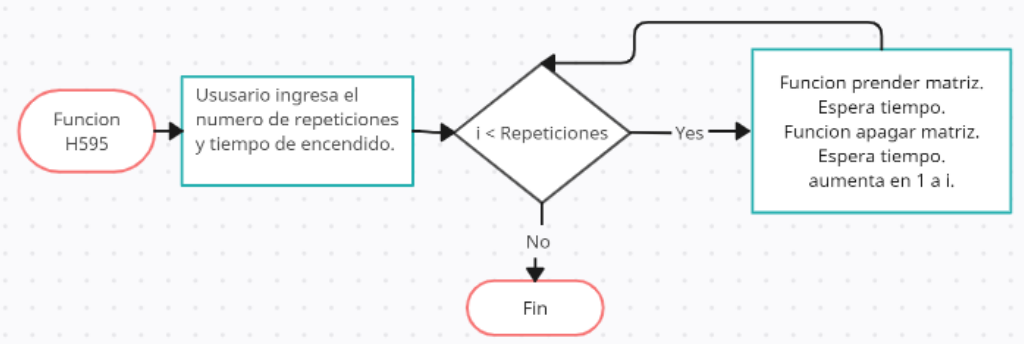

Figura 7. Función verificación.

Las funciones prender y apagar llaman por separado a la función H595 para así apagar y prender todos los leds teniendo en cuenta que se debe poner todas las filas el ánodo en 1 y todas las columnas el cátodo en cero para prender todos los leds, y para apagar todos los leds se debe poner tantos el ánodo como el cátodo en cero en todos los leds.

Función imagen: Esta función esta encargada de interactuar con el usuario para que este ingrese la matriz posición por posición de los leds que se van a encender y posteriormente se llama la función vector2sum para desplegar la matriz en los leds.

Función patrones: Esta función se encarga de llamar a los 4 patrones creados en funciones aparte y desplegarlos en pantalla uno a uno.

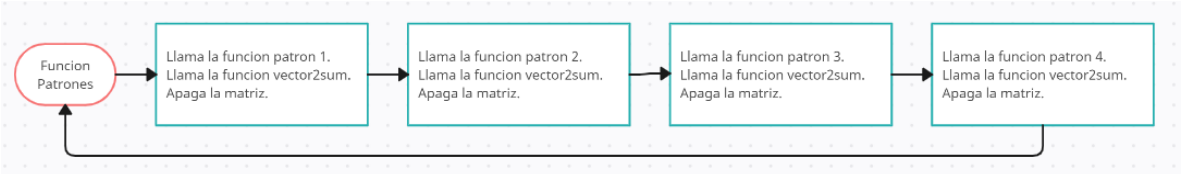

Figura 8. Función patrones.

RESULTADOS.

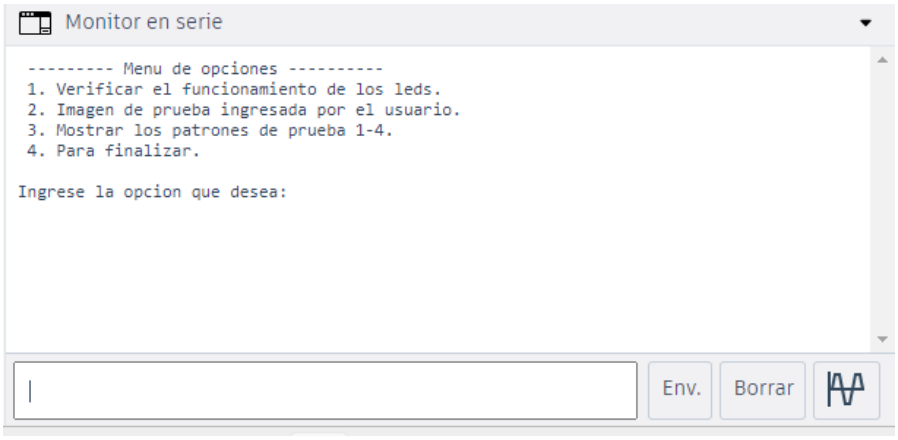

Figura 9. Menú de interacción.

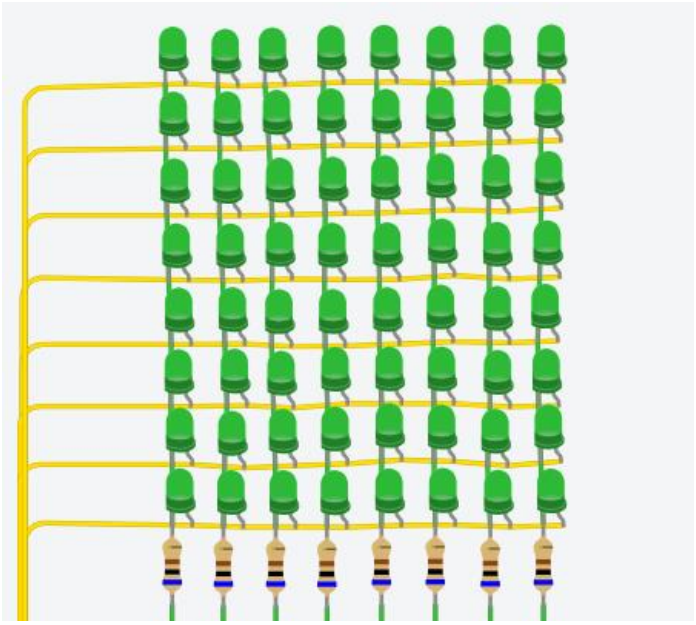

Figura 9. Matriz encendida.

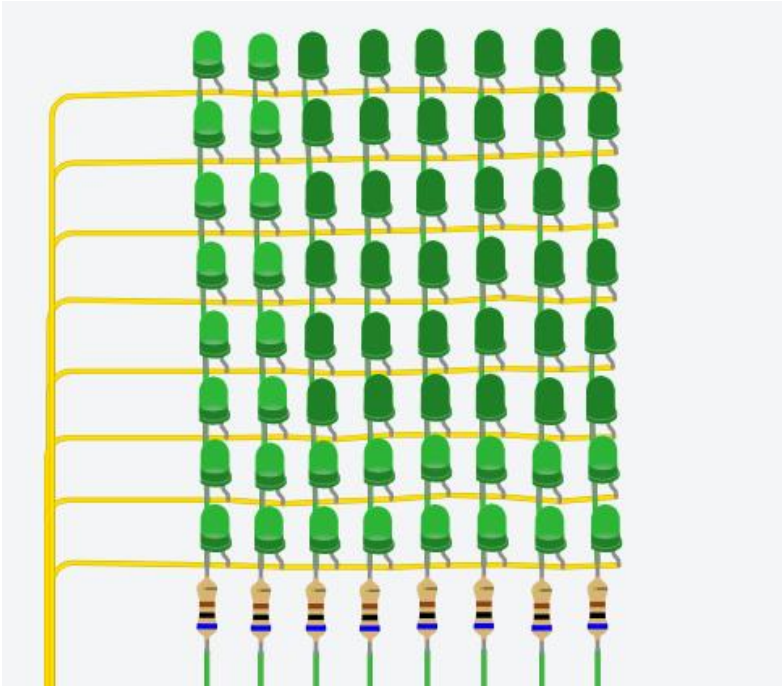

Figura 10. Matriz ingresada por el usuario, letra L.

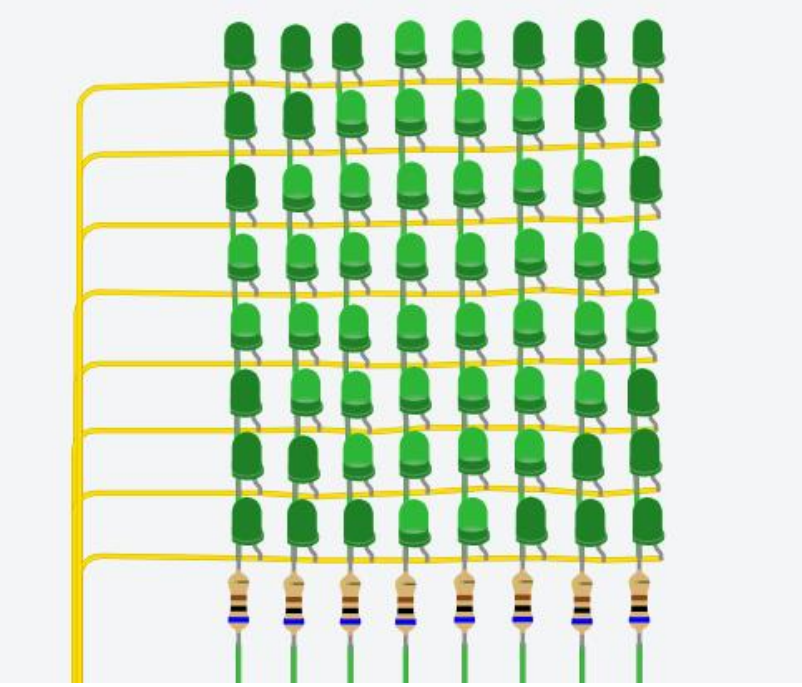

Figura 11. *Patrón 1 desplegado.*

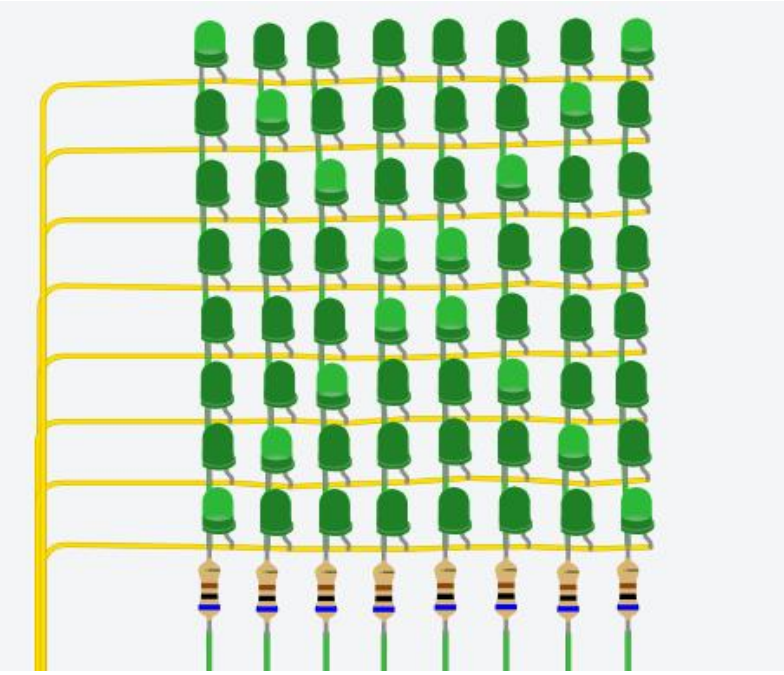

Figura 12. *Patrón 2 desplegado.*

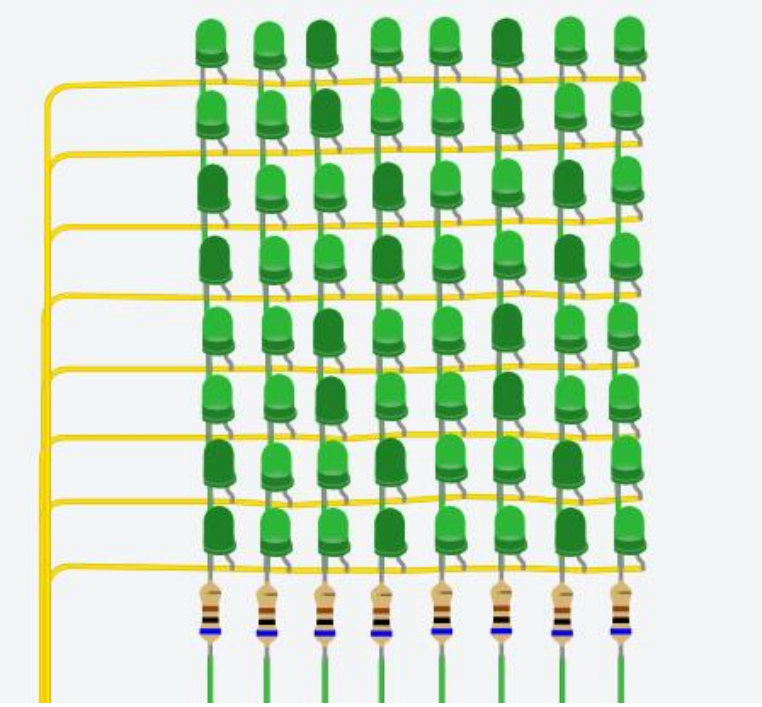

Figura 13. Patrón 3 desplegado.

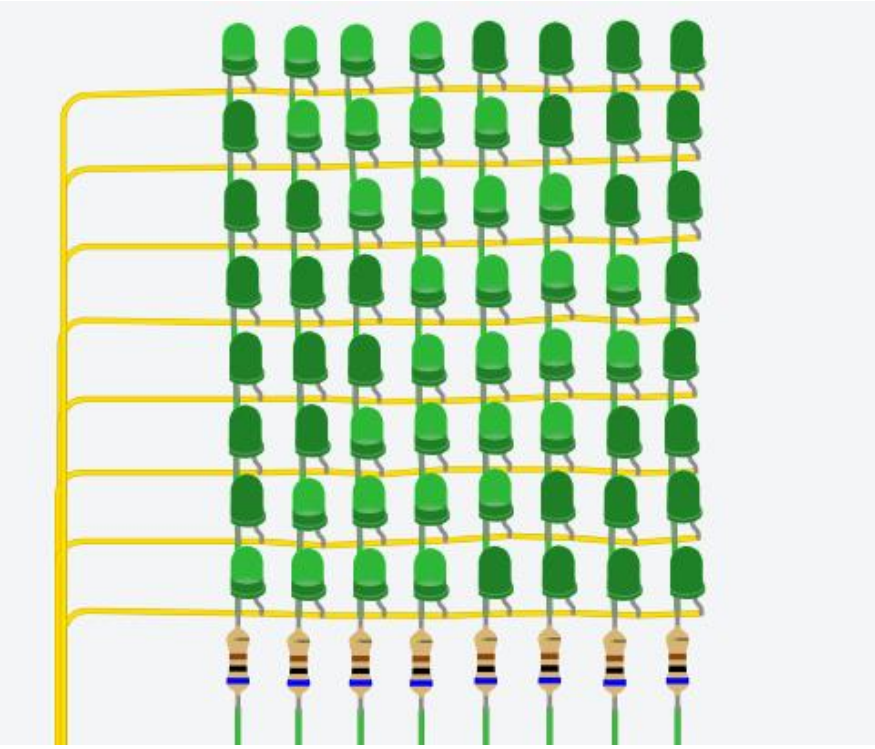

Figura 14. Patrón 4 desplegado.

Conclusiones.

Las placas de desarrollo pueden tener limitantes de hardware que pueden ser superadas con integración de hardware externo, en este caso se uso un registro de desplazamiento para superar la necesidad de múltiples salidas digitales.

Es necesario tener buenas técnicas de programación en conjunto con un lenguaje de alto desempeño para los sistemas embebidos, todo eso se puede llevar a cabo con C++. Para estos sistemas se necesita velocidad de ejecución del firmware en procesadores limitados, además que se requiere confiabilidad en el código que se entrega ya que la idea del firmware es que no se necesite ser modificada.

Referencias.

EASY TECH. *How to make Led Matrix | 8X8 Led Matrix*. 2018. Link: https://www.youtube.com/watch?v=U3ccR3xbJSQ

BenJi. *Cómo usar Matriz 8x8 con 74HC595 y Arduino (Efecto Scrolling).* Link: https://www.youtube.com/watch?v=Unw4Hfdr7pI

Luis del Valle Hernández. 74HC595 registro de desplazamiento con Arduino. Link: https://programarfacil.com/blog/74hc595-registro-de-desplazamiento-arduino/

Parzibyte. *Tutorial Arduino: 74hc595 y leds.* Link: https://parzibyte.me/blog/2017/11/07/tutorial-arduino-74hc595-leds/