Proyecto Parcial 1 Informática 2

ACTIVIDAD

Matriz de Led ’s 8x8 Programable

(Informe)



Facultad de Ingeniería

Departamento de Ingeniería Electrónica y

Telecomunicaciones

**Estudiante:**

Emiliano De Jesús Lince Diaz

Universidad de Antioquia

Medellín

2021-2

Tabla de contenido

[Objetivos 2](#_Toc146568576)

[Introducción 3](#_Toc146568577)

[Enlaces 6](#_Toc146568578)

[Informe 6](#_Toc146568579)

[Circuito. 6](#_Toc146568580)

[Software o Programa 7](#_Toc146568581)

[Problemas afrontados 12](#_Toc146568582)

[Conclusión 13](#_Toc146568583)

[Webgrafía 13](#_Toc146568584)

# Objetivos

1. Desarrollar la capacidad de solución de problemas en los estudiantes enfrentándolos a problemáticas de la vida cotidiana.

2. Identificar si el estudiante adquirió las destrezas y conocimientos fundamentales de la programación con C++, en donde resaltamos estructuras de programación (iterativas, secuenciales y decisión), tipos de datos, apuntadores, arreglos y funciones.

3. Evaluar la capacidad del estudiante para trabajar con Arduino e integrar la programación en C++, además de usar de manera adecuada las funciones de la plataforma.

# **Introducción**

Para esta primera parte nuestra meta es lograr encontrar una manera de crear una matriz de leds totalmente funcional sin tener que usar muchos pines digitales del Arduino o varios Arduinos, Pero ¿cómo logramos eso?, Ahí entra un Microprocesador el cual nos ayudara a facilitar esta tarea y no tener que usar sino tan solo 3 pines digitales del Arduino, este “Milagroso” aparato es un registro de desplazamiento, de esta manera nos ayudara a controlar matrices de led son Arduino:

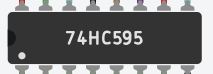


Imagen 74HC595.

Pero para llevar a cabo todo el proceso y llegar a lo que queremos debemos investigar e ir probando con el 74HC595 para comprender bien cual es su funcionamiento y como aplicarlo para nuestra meta final.

[](https://www.youtube.com/embed/ROKculksEr0?feature=oembed)Primero veamos un video donde explican el uso del 74Hc595 con un arduino:

Video sobre 74HC595 con Arduino.

También podemos entrar a esta pagina donde nos explica el registro de desplazamiento del 74HC595:

|  |  |
| --- | --- |
| [1] | L. d. V. Hernández, «Programarfacil.com,» 2020. [En línea]. Available: https://programarfacil.com/blog/74hc595-registro-de-desplazamiento-arduino/. |

Y además también veamos un Datasheet sobre este elemento y asi tener bien claro el cómo usarlo.

<https://pdf1.alldatasheet.com/datasheet-pdf/view/12198/ONSEMI/74HC595.html>.

Enlace Datasheet del 74HC595.

Bueno ahora conociendo un poco del 74HC595 podemos llevarlo a la practica y empezar por hacer proyectos sencillos como el mostrado en el video al principio de la introducción, igualmente al final de la introducción montare un montón de enlaces donde estará varios videos de YouTube donde nos podemos guiar para aprender a usar este aparato electrónico muy interesante.

Para el primer proyecto “Conociendo el 74HC595” conectaremos el microprocesador a una línea de leds y la pondremos a prueba:

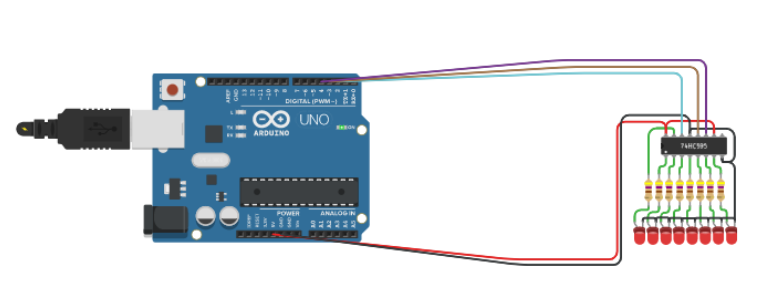


Imagen del circuito a armar en tinkercad.

Esto para probar el funcionamiento del registro de desplazamiento del 74HC595, luego podremos ver que se puede hacer algo interesante con este elemento electrónico y es que se pueden conectar entre si formando una línea mas larga de leds y la podemos poner a prueba, los videos y los enlaces a tinkercad estarán al final de este documento.

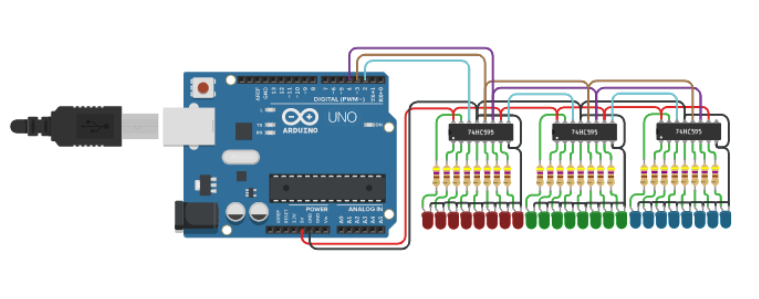


Imagen con 3 “74HC595”.

Imagen que contiene circuito, electrónica

Descripción generada automáticamente

Imagen con 4 “74HC595”.

Por último, armaremos una matriz de 64 leds y para ello tendremos que usar a 8 “74HC595” conectados entre si mismos y asi solo tendríamos que usar 3 pines digitales de nuestro Arduino:

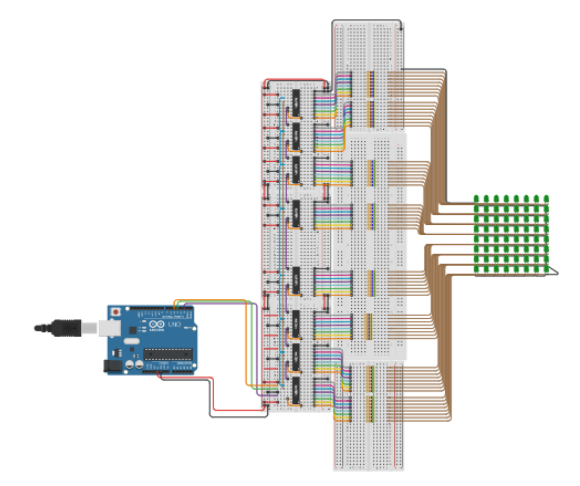


Imagen con la matriz completa. (Imagen CF)

En este último solo esta codificado para comprobar el correcto encendido de todos los leds nada más.

Aquí encontraremos los enlaces para poder llevar a cabo dichas pruebas y proyecticos para aprender a usar este microprocesador, también habrá para armarlos en físico por si queremos hacerlo de manera física si disponemos de un Arduino:

## Enlaces

Enlaces tinkercad**.**

<https://www.tinkercad.com/things/5dwhZNOAhox?sharecode=72GIrhW27O96lesZSqiW8xXWUO_ibv0zxrT9G_aMpi0>.

<https://www.tinkercad.com/things/cdbm6Lv3R2x?sharecode=9vLRdRZq2IHgUdNyRie6wzDbc90FOxNKSy5nRgrMx0Q>.

<https://www.tinkercad.com/things/6KTbxrBXTqu?sharecode=Q411Exxsr7Py0EtP4CnDuL70QFWl_xSV-tW7Am_Ovj8>.

<https://www.tinkercad.com/things/fjnr0wLvULA?sharecode=6Gq8ReMpSraqa0KPscm5Dk0ag4gTbh0myU3o0GsCdRg>.

Enlaces videos YouTube.

<https://youtu.be/pBdjMayFK5Y?si=fzc5JCjtUdeZ5vFK>.

<https://youtu.be/LFqIA3ZvZE8?si=JizNVPRnpFzjjJdc>.

<https://youtu.be/JyCXS2bl3tI?si=-O_vVDCWRuLqyzYt>.

<https://youtu.be/vnBsBwAp9Rg?si=nne9nUz0h7F3OyxK>.

# Informe

## Circuito.

Para el desarrollo como tal de esta practica o proyecto tenemos que examinar primero la parte de introducción y haber comprendido el uso del 74HC595 y haber concretado el diseño final del circuito con la matriz completa “Imagen CF”, ya teniendo el diseño podemos empezar con la parte de la programación, pero voy a dar una corta introducción de las conexiones del circuito final:

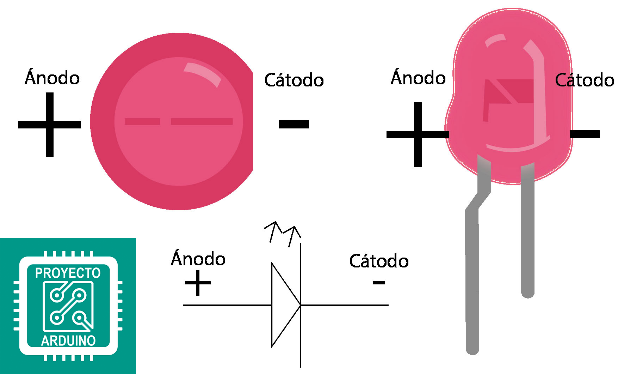
Se usará para el desarrollo de esta práctica la página web de Tinkercad, donde usaremos un Arduino uno para enviar las señales a la matriz 8x8 por medio de tres pines digitales, para ello tendremos que usar un microprocesador llamado 74HC595, en el caso mostrado en este informe es usado 8 de estos conectados en serie, ya que cada uno de los 74HC595 tiene para conexión de 8 leds podemos tener conectada toda la matriz de manera individual, ósea cada led por separado, ¡¡Importante!! Tan solo están conectados individualmente por el ánodo, pero todos los leds están en configuración cátodo común.

Imagen led.

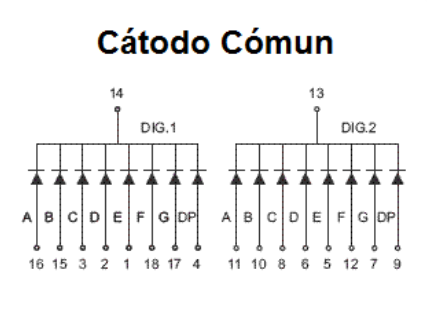
Eso significa que están conectados en serie por el cátodo.

Imagen ejemplo leds conectados cátodo común.

Con esto ultimo ya tendremos la matriz de leds y conectadas al 74HC595, si no esta muy clara como es la conexión de los leds a los 74HC595 y la conexión que hay entre ellos junto con la del Arduino, te invito a ver la introducción y los videos de ejemplo, igualmente junto con este informe será anexado un enlace a un video de YouTube donde explicare el funcionamiento del circuito y sus conexiones, además del programa.

## Software o Programa

**Introducción.**

Bueno ya teniendo claras las conexiones y teniendo todo el circuito montado, vamos con lo mas importante que es el programa a ejecutar para que el circuito haga las funciones que nosotros le pidamos y queramos, para este caso voy a efectuar tres funciones; la primera que me encienda todos los leds de la matriz, la segunda que me genere un patrón al que el usuario va a ingresar por medio de consola y tercero una serie de 4 patrones donde irán mostrándose uno por uno seguidos.

Para finalizar hare una función que será mi menú para escoger la opción de lo que quiero que haga la matriz de leds.

Primero mostrare el código completo y les voy explicando cada parte para que vayan entendiendo como es el funcionamiento de cada punto del código.

**Primera parte.**

Para esta primera parte tenemos lo más básico:

const int dataPin = 2; // Pin de datos (DS)

const int clockPin = 5; // Pin de reloj (SHCP)

const int latchPin = 4; // Pin de almacenamiento (STCP)

const int numFilas = 8; // Número de filas de LEDs

const int numColumnas = 8; // Número de columnas de LEDs

byte matrizLEDs[numFilas]; // Matriz para almacenar el estado de los LEDs

void setup() {

pinMode(dataPin, OUTPUT);

pinMode(clockPin, OUTPUT);

pinMode(latchPin, OUTPUT);

Serial.begin(9600);

}

void loop() {

publik();

}

Podemos observar que se instancian varias “variables” como “const int” de esta manera definimos que son constantes y son enteros, luego definimos la matriz como un arreglo de variable byte de esta manera guardara como datos solo 1 y 0 luego viene la función void setup() donde definimos los pines digitales como salida del Arduino, y luego la entrada serial, para el ingreso de datos e información por medio del monitor serial y ya por último la función repetitiva void loop() donde llamamos a la función publik() la cual es como nuestro menú de selección para escoger entre que queremos que ejecute nuestro programa, que función llamar y utilizar, esta función la podremos ver a lo ultimo del código, mas adelante explico de ella.

**Segunda parte.**

En esta segunda parte viene nuestras funciones extras o mejor dicho funciones de apoyo:

void actualizarMatriz() {

digitalWrite(latchPin, LOW); // Preparar para la carga en paralelo

for (int fila = 0; fila < numFilas; fila++) {

shiftOut(dataPin, clockPin, LSBFIRST, matrizLEDs[fila]);

}

digitalWrite(latchPin, HIGH); // Actualizar los registros de almacenamiento

}

void limpiarMatriz() {

for (int fila = 0; fila < numFilas; fila++) {

matrizLEDs[fila] = 0;

}

}

Estas dos funciones nos van a ser de mucha utilidad ya que con actualizarMatriz() es actualizar la matriz de manera que nos muestre la figura que pedimos en la siguiente opción que escojamos, y la función limpiarMatriz() para que nos ponga toda la matriz en 0 y apague todos los leds para la próxima utilización de la matriz, ambas son importantes y se complementan en si ya que nos ayudan a que nos muestren una imagen o patrones limpios y no con algún led mal encendido.

**tercera parte.**

Para esta tercera parte traigo ya las funciones principales como tal, la idea es observar como es el funcionamiento de cada una y probarlo en nuestra configuración de matriz de leds hecha en Tinkecad.

Primero mostrare una parte que es la función verificación () donde lo que se hace es rellenar la matriz de leds de 1 para comprobar que enciendan todos los leds, lo que hace es recorrer cada fila del arreglo hasta el ultimo y encenderlos todos, luego tenemos la función imagen que lo que va a hacer es encender los leds de la configuración de matriz que ingresemos por el monitor serial con 1 y 0.

//--------------------------------------------------------------------------------------------------------------------------//

void verificacion() {

digitalWrite(latchPin, LOW); // Preparar para la carga en paralelo

for (int i = 0; i < 8; i++) {

shiftOut(dataPin, clockPin, LSBFIRST, 255); // 255 enciende todos los bits

}

digitalWrite(latchPin, HIGH); // Actualizar los registros de almacenamiento

}

//--------------------------------------------------------------------------------------------------------------------//

void imagen() {

limpiarMatriz(); // Limpiar la matriz antes de definir un nuevo patrón

Serial.println("Ingrese el patrón por filas (8 filas de 8 bits cada una):");

for (int fila = 0; fila < numFilas; fila++) {

Serial.print("Fila "); // Muestra el número de fila actual

Serial.print(fila + 1); // Suma 1 para mostrar el número de fila en base 1

Serial.print(": ");

while (!Serial.available()) {

// Esperar a que el usuario ingrese la fila

}

// Leer la fila ingresada por el usuario

byte filaBits = 0;

for (int bit = 0; bit < 8; bit++) {

while (!Serial.available()) {

// Esperar a que el usuario ingrese cada bit de la fila

}

char bitChar = Serial.read();

filaBits |= (bitChar == '1') ? (1 << bit) : 0;

}

matrizLEDs[fila] = filaBits;

}

// Mostrar el patrón ingresado en la matriz

actualizarMatriz();

}

//-----------------------------------------------------------------------------------------------------------------//

La otra parte del código es mas larga por lo tanto no la anexare aquí, pero este informe será subido junto con un archivo texto donde estará el código en C++ completo.

Igual aquí explico que lo que hace esas funciones son generar 4 patrones diferentes y luego una función que muestre una secuencia con cada patrón. Esa última función con la secuencia es la que será llamada en el menú ósea en la función publik(), cada función se llamaba formarpatron1(), formarpatron2(), formarpatron3(), formarpatron4() y la ultima que es la secuencia se llama MostrarPatrones().

**Ultima parte.**

void publik() {

Serial.println("Selecciona una opción:");

Serial.println("1 Verificar funcionamiento de LEDs.");

Serial.println("2 Mostrar imagen de prueba.");

Serial.println("3 Mostrar patrones alternados 1 al 4.");

while (!Serial.available()) {

// Espera a que el usuario envíe un comando por el puerto serial.

}

char opcion = Serial.read();

switch (opcion) {

case '1': // Implementa la funcion que verifica el encendido de todos los leds

verificacion();

delay(1000); // Espera 1 segundo antes de encender los LEDs nuevamente

break;

case '2':

// Implementa la funcion que muestra el patron ingresado de la persona.

imagen();

break;

case '3':

// Implementa la funcion que muestra los 4 patrones definidos.

MostrarPatrones();

break;

default:

Serial.println("Opción no válida.");

}

}

Aquí muestro la última función la del menú, la función publik(), lo que hace es preguntar al usuario que opción va a escoger y mostrarle las opciones, dependiendo lo que escoja ejecuta el case y llama la función que fue colocada en ese case.

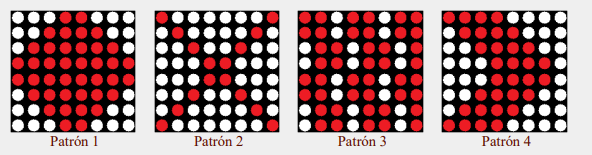
Una disculpa en la parte de los patrones que va a mostrar la matriz son estos 4 patrones:

Imagen patrones a mostrar en la función MostrarPatrones().

## Problemas afrontados

Para este proyectos afrente demasiados problemas y uno de ellos fue el del diseño del circuito ya que yo tenia pensado en buscar una manera más eficaz de desarrollarlo sin necesidad de tantos elementos “74HC595” con tan solo dos de ellos, lo que se hacia es que se conectaban la matriz por columnas y filas a los anodos y a los catodos de cada una, los catodos van a uno de los 74HC595 y los ánodos a los pines del otro, de esta manera lo que se hace es se envían bits a las filas y columnas, los bits que coinciden en 1 se enciende el led en esa posición y los bits que no se apagan los leds.

El circuito se observaba más o menos así:

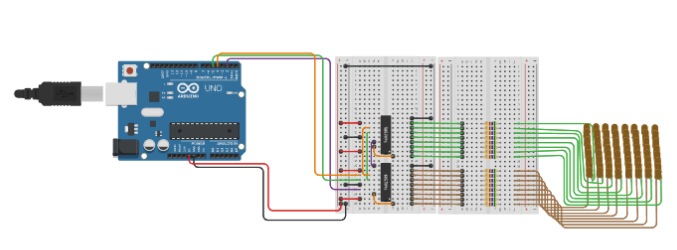


Imagen circuito falló.

Pero al final no funciono, el enlace a Tinkercad estará subido junto a este documento. Por si se encuentra alguna solución al problema.

Los demás problemas son con respecto al código ya que aveces no llegaba al patron deseado o llegaba, pero quedaba al revés, lo que hacía es que volvía a planear el código y lo volvía a ejecutar y probar hasta que funcionara y me diera el patrón que deseaba. Pero el código aquí mostrado y el que será montado con un archivo texto junto con este documento está totalmente funcional.

# Conclusión

El desarrollo del circuito requiere tiempo y esfuerzo, junto con una muy buena planeación, espero que al final les funcione correctamente el circuito como a mí, fue complicado, pero al final se logro llegar a lo que se requería y cumplir los objetivos propuestos. Aquí seguido subo enlaces de apoyo para el desarrollo de esta práctica.

# Webgrafía

<https://www.arduino.cc/reference/en/language/functions/advanced-io/shiftout/>.

<https://drive.google.com/drive/folders/18ygeodNWQcUq9rYGwrPzMp5RsIEUDmBm>.

<https://www.youtube.com/watch?v=Unw4Hfdr7pI>.

<https://www.youtube.com/watch?v=QsHCZ-jUD9E>.

<https://www.youtube.com/results?search_query=matriz+led+8x8+arduino+y+74HC595>. Varios videos que sirven de ayuda.