**Informe de Conexión para Matriz de LEDs 8x8 con Registros de Desplazamiento 74HC595 y razonamiento para código**

**Alejandro Naranjo Naranjo**

**Introducción:**

El objetivo de este informe es proporcionar una guía detallada sobre cómo se llevó a cabo el análisis y diseño de la conexión para una matriz de LEDs 8x8 utilizando registros de desplazamiento 74HC595. Esta matriz se utilizará para mostrar patrones de iluminación en una variedad de proyectos electrónicos. La elección de utilizar registros de desplazamiento en cascada permite un control eficiente de una gran cantidad de LEDs con un número limitado de pines de salida en un microcontrolador.

**Pasos del Análisis y Diseño:**

A continuación, se detallan los pasos que se siguieron para llegar a la conexión final de la matriz de LEDs 8x8:

**Paso 1: Definición de Requisitos:**

El primer paso fue definir los requisitos del proyecto, que incluían:

* Tamaño de la matriz: 8x8 LEDs.
* Uso de registros de desplazamiento 74HC595.
* Control desde un Arduino.
* Capacidad para mostrar patrones y animaciones personalizados.

**Paso 2: Selección de Componentes:**

Se seleccionaron los siguientes componentes:

* 2 registros de desplazamiento 74HC595 para ampliar las salidas.
* Matriz de LEDs 8x8 de cátodo común.
* Resistencias limitadoras adecuadas para los LEDs.
* Un Arduino como microcontrolador.

**Paso 3: Investigación:**

Se realizaron investigaciones para comprender el funcionamiento de los registros de desplazamiento 74HC595, la disposición de la matriz de LEDs 8x8 y cómo se interconectan.

**Paso 4: Esquema de Conexión:**

Se creó un esquema de conexión preliminar que mostraba cómo se conectarían los registros de desplazamiento y la matriz de LEDs. Esto incluía la conexión de los pines DS, SHCP, y STCP de los registros de desplazamiento al Arduino y la conexión de las filas y columnas de la matriz de LEDs.  
  
**Uso de 74HC595**

Empezamos eligiendo el registro de desplazamiento 74HC595 debido a su capacidad para controlar múltiples LEDs con solo unos pocos pines de salida de Arduino. Esto nos permite ahorrar pines y simplificar la conexión.

**Abstracción de la Matriz**

Uno de los principios fundamentales de este proyecto es la abstracción de la matriz de LEDs. En lugar de representar explícitamente cada LED con un estado "encendido" o "apagado", optamos por una representación más abstracta. Usamos una matriz de datos que representa el estado lógico de cada LED, pero sin especificar directamente los valores. Cada elemento de la matriz es un "bit" que puede ser 1 (encendido) o 0 (apagado).

**Funciones para Crear Patrones**

En lugar de definir cada patrón de LED de manera estática, creamos funciones que generan patrones dinámicamente. Por ejemplo, una función podría generar una estrella, otra una flecha y así sucesivamente. Estas funciones manipulan la matriz de datos utilizando punteros y arreglos para establecer qué LEDs deben estar encendidos o apagados.

**Manipulación de Memoria Dinámica**

Para permitir una mayor flexibilidad en la creación de patrones, utilizamos memoria dinámica. Algunos patrones pueden requerir un tamaño variable en memoria, y el uso de punteros y funciones nos permite asignar y liberar memoria según sea necesario.