**Primer examen parcial**

**1 - Planteamiento del problema:**

Se requiere construir una matriz de leds con dimensiones de 8 filas por 8 columnas que sea capaz de mostrar una serie de patrones. Para ello se debe hacer uso de un microcontrolador Arduino UNO y el lenguaje de programación C++ además de los conocimientos adquiridos en clase e investigación autónoma.

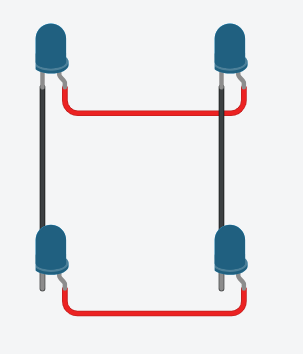
**2 - Análisis del problema:**

Al abordar el problema no hemos percatado de la siguiente situación: El problema es muy particular y tanto el hardware como el software están muy vinculados entre sí. Esto nos ha llevado a tener presentes ambos frentes de tal manera que se pueda generar una solución adecuada, evitando perder tiempo en reestructurar gran parte del análisis porque las soluciones de hardware y software son incompatibles entre sí.

**2.1 – Abordando el hardware:**

Una matriz es un conjunto de elementos distribuidos en forma de filas y columnas, en este caso, leds. Para controlar los leds de manera individual se pueden organizar sus terminales eléctricas de modo que estas formen un plano ordenado, de otra forma se podría contar con un número muy alto de terminales eléctricas (64 terminales positivas). Para solucionar esto, pueden conectar todos los ánodos de una misma fila teniendo así una única terminal positiva por cada fila de leds y conectar los cátodos de una misma columna para tener una sola terminal negativa por columna. (figura 1).

(figura 1. Rojo: ánodo, negro: cátodo)



Con esta disposición se reduce de 64 terminales positivas a tan solo 16 terminales entre positivas y negativas, que deben de poder ser controladas de manera individual. Esto genera un nuevo dilema pes la plataforma Arduino UNO posee una cantidad de conectores limitados y para el desarrollo de este parcial solo pueden ser usadas un máximo de 7 salidas digitales.

Este nuevo problema se puede resolver utilizando el **registro de desplazamiento** **74HC595,** el cual nos permite transformar una **entrada de ocho bits en serie** a ocho **salidas individuales en paralelo,** es decir, el IC (circuito integrado) recibe un byte y lo transforma en ocho bits individuales.

Esto supone una gran ventaja, como explica del Valle (s.f.) en “*Programafacil.com”*, nos permite transformar tres salidas digitales del Arduino UNO en ocho salidas digitales, además de poder conectarse varios de estos en serie aumentando en 8n salidas digitales con solo 3 salidas digitales del Arduino UNO.

Para resolver el problema inicial, se hará uso de 2 registros de desplazamiento 74HC595, uno controlará los ánodos de los leds y el otro los cátodos.

**2.2 – Abordando el software:**

Para controlar la imagen se planea generar un conjunto de 6 arreglos tipo *char* ( de un byte de longitud) con ocho caracteres de longitud cada uno. Cada arreglo representará una figura distinta, el sexto arreglo tendrá los valores = {128, 64, 32, 16, 8, 4, 2, 1} que representan las filas a encender en un momento determinado generando un efecto “barrido”, esto se debe a que la solución de hardware no puede mostrar la imagen completa en un solo instante, requiriendo de varios ciclos para realizarlo.

Para evitar la *solución trivial*, estos arreglos serán generados en tiempo de ejecución:

* Se harán 5 funciones, una por cada patrón, diseñadas para generar el arreglo con la información de la columna requerida para reproducir en la matriz cada patrón según sea solicitado (se incluye la función *“verificación”* solicitada en la guía).
* Se hará una función llamada *dibujador*, encargada de recibir una tupla de bytes y dibujarla en la matriz.
* Se hará una función llamada “imagen” que se comunicará con la función *“publik”*, *dibujador* y una de las *“funciones patrón”* .
* Se hará un pequeño programa que permita seleccionar entre distintos modos (Verificar funcionamiento de los leds, elegir un patrón, mostrar los patrones de forma alternada), este programa será la función *“publik”*.

**2.3 Análisis de cada problemática y especificaciones a detalle:**

* + - *Especificación 1:*
* Se debe de implementar un circuito en tinkercard que nos permite controlar 64 leds de la forma más eficientemente posible y con menor uso de pines digitales posibles, siendo además capaces de controlar estos leds y poder decidir sobre el estado de cada uno de ellos. Se decidió implementar una matriz de ánodo común conectando las filas a los 5v y las columnas a tierra (GND). Para el código se generará una matriz 8\*8 en C++ y se le asignará un puntero a su posición (0,0), que permita recorrerla por dos ciclos *“for”.*
  + - *Especificación 2:*

Se debe de implementar una función que de la orden a los 64 leds de prenderse a la misma vez y de apagarse también en sincronía esto con el fin de poder verificar que todos los leds estén funcionando correctamente.

En el tinkercard se pude lograr esto mediante el integrado 74HC595, que al recibir una entrada en binario divide esta secuencia de 8 bits en 8 salidas con la que cada una va a una fila, nótese que al ingresar el numero 255 (en binario: 1111111) esta manda una señal de 1 a cada fila, permitiendo prender los 64 leds de la matriz.

* + - *Especificación 3:*
    - *Especificación 4:*

a). Para generar este patrón se nos ocurre el generar dos ciclos anidados con 2 variables controladoras, el contador de espacios y el contador de leds, en un inicio el contador de leds va aumentando de dos en dos mientras el contador de espacios disminuye de a dos, esto hasta que el contador de espacios es 0 y el contador de les es 8 en cuyo caso se revierte la situación haciendo que incrementen los espacios y disminuyan los leds activos.

b). Para generar este patrón es necesario dar corriente a las columnas que cumplan que se encuentran en la misma posición de la fila evaluada y en las que cumplan la ecuación (8-j) en la que j es la posición de la fila evaluada iniciando desde j=0.

c).

* + - *Especificación 5:*