Informe Parcial Informática II Andres Felipe Agudelo Zapata

**Análisis de los problemas:**

* **Análisis Problema 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.*

* **Análisis Problema 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.

* **Análisis Problema 3:** Para darle la libertad al usuario de imprimir lo que desee, vamos a preguntarle al usuario sobre el estado de cada led es decir sobre si desea que este encendido o apagado. Y al final terminara por imprimir la imagen deseada
* **Análisis Problema 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 uno en uno mientras el contador de espacios disminuye de uno a uno, esto hasta que el contador de espacios es 0 y el contador de leds sea 8 en cuyo caso se revierte la situación haciendo que incrementen los espacios y disminuyan los leds activos, mas esta fila completamente iluminada se repite.

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). Este patrón nos muestra un ciclo de cuadrados 2x2 que dejan un espacio en blanco entre cada cuadrado, para facilitar la programación se tomó que la tercer y cuarta fila de cuadrados son una combinación de la primera y de la segunda, en el que la segunda fila (o cuarta) esta corrida una porción a la derecha respecto a la primera(tercera). Para dibujar este patrón es necesario aparte de los dos ciclos “*for*” que recorren la matriz el de diversos condicionales que evalúen si la fila es par, (esto debido a que las filas impares son simplemente una copia de su fila pasada) y también es necesario el utilizar una variable que nos permita evaluar si se debe dibujar en esa celda o si por el contrario se debe de dejar en blanco.

d). Si se analiza a profundidad el patrón se observa que este ejercicio es similar al clásico ejercicio en el que se debe imprimir una pirámide invertida, solo que en este ejercicio se invierten las filas con las columnas por lo que la matriz esta volteada, más esto se soluciona fácil al invertir las filas y las columnas también en la asignación, nótese además que en el ejercicio la base inicial es 6 y esta corresponde a la cantidad de leds apagados por lo que los extremos equivalen a los leds prendidos y el centro son los leds apagados, esta cantidad va disminuyendo de a uno mientras al mismo tiempo la cantidad de leds prendidos va en aumento también de a uno, cuando los leds encendidos equivalen a 8, se invierten las posiciones pasando los leds apagados a los extremos y los leds encendidos al centro mientras al mismo tiempo los leds apagados van en aumento y los leds encendidos disminuyen.

* **Análisis Problema 5**

El problema 5 no es mas que el diseño de un menú en el monitor serial del Arduino que nos permita invocar las funciones desarrolladas con anterioridad siendo la parte a la invocación del problema 2, la función b la invocación del problema 3, y la función c la invocación de las 4 funciones del problema 4