Informática 2

Parcial 1

Johan Hernández Vargas

Kevin Jimenez Rincon

Manuela Gutierrez Rodriguez

Despartamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Abril de 2021

Índice

1. Análisis del problema y consideraciones para la alternativa de solución propuesta.

El problema principal es identificar en el codigo la matriz de leds 8x8, ademas de esto, definir 3 funciones, una para verificar el funcionamiento de los 64 leds, otra para la impresion de un simbolo en esta matriz por el monitor serial y una más para imprimir un patron ingresado desde el mismo monitor serial. Además de esto la elaboracion de un manual completo y detallado en el cual, usuarios y programadores, puedan entender el funcionamiento de este circuito sin necesidad de tener que acceder al código.

2. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo.

De acuerdo al análisis que se realizó, se establecieron una serie de tareas a realizar con el fin de solucionar el problema de forma ordenada.

1. Determinar que componenetes vamos a usar para conectar el circuito.
2. Probar cual de las formas de conexión es más efectiva.
3. Conectar los 8 pines entre ellos de tal forma que trabajen en conjunto como uno solo.
4. Reducir al máximo posible la cantidad de pines utilizados.
5. Inicializar el puerto serial y configurar las entradas, los registros de dezplazamiento y los registros de salida.
6. Verificar que todo esté conectado correctamente y verificar que los 64 LEDs funcionan.
7. Crear una función que nos permita recibir una matriz y almacenarla.
8. Crear una función que nos permita imprimir un patrón preestablecido por la matriz.
9. Crear una función que nos permita ingresar diversos patrones y almacenarlos en la cantidad de matrices necesarias.
10. Imprimir cada matriz de forma consecutiva según el tiempo establecido por el usuario.

3. Algoritmo implementado.

El algortimo que se implementó consiste en un menú principal que se encuentra desntro de la función **loop();** en el cual el usuario puede decidir que desea realizar al ingresar la opción por el monitor en serie.

Cuando el usuario ingresa la primera opción se invoca a la función **verificacion();** que consiste en un algoritmo que envia la señal de encendido a cada uno de los led, dentro de esta función se encuentra otra llamada **relojregistro();** que tiene como función enviar un flanco al reloj de registro de salida o al reloj de registro de desplazamiento. Luego de esto, se establece un tiempo de duración de los led prendidos para que después el algoritmo envíe una señal low y los led se apaguen.

Cuando el usuario ingresa la segunda opción, en primer lugar se invoca la función **definematrizbidi();**, que es la encargada de definir una matriz dinámica de 8x8 a través de punteros y asignando a cada posición de la matriz el valor 0. La segunda función invocada es **RecibirMatrizbidi();**, que es la encargada de pedirle al usuario fila por fila y posteriormente almacenar cada dígito de la fila en su posición correspondiente dentro de la matriz. Por último, se invoca la función **imagen();**, que es la encargada de recorrer la matriz y enviar a un led en una posición determinada su valor correspondiente en la matriz, de esta forma el led se puede mostar como encendido o apagado.

En la última opción nos encontramos con una serie de inputs que le pide al usuario la cantidad de patrones que desea mostar y el tiempo en milisegundos que desea que se muestre cada patron, la primera función que se encuentra aquí es **definematriztridi(cpatrones);** que recibe como parámetro la cantidad de patrones que se desea ingresar y crea un arreglo dinámico de matrices 8x8, es decir, una matriz de matrices e inicializa cada elemento con el valor de 0. La segunda función presente es RecibirMatriztridi(cpatrones); que es la encargada de pedir las filas de cada una de las matrices dentro del arreglo y almacena el valor ingresado en su posición correspondiente. Por último se encuentra la función **publik(tpatrones, cpatrones);** que además de recibir como parámetro la cantidad de patrones, también recibe el tiempo de duración de cada patrón.

4. Problemas de desarrollo que presentó.

Uno de los principales problemas fue el uso de los pines digitales, teniendo en cuenta las restricciones planteadas. Además, contamos con problemas de distribucion en el TinkerCad de cableado, leds y otros componentes.

Además de esto, hemos presentado inconvenientes en la programación de las funciones, ya que contábamos con poco conocimiento previo de Arduino y ha sido muy complejo traducir el codigo de C++ a lenguaje Arduino. Por otra parte, la plataforma tinkercad cuenta con ciertas dificultades que hacen más complicado el desarrollo del proyecto, entre ellas se encuentra el hecho de que la plataforma no guarda bien el proyecto y se eliminan componentes o líneas de código, además de que su debugger no es el más óptimo y a la hora de querer

imprimir algo en lugar de aparecer el error se imprimen caracteres extraños, que muchas veces no se encuentran codificados en el código ASCII.

También, se han presentado inconvenientes a la hora de buscar información respecto a dudas con tinkercad, ya que no se encuentra mucho al respecto. Por otra parte, también hemos tenido inconvenientes a la hora de querer recibir datos por el monitor en serie, ya que algo que no se dice mucho es que cada vez que se ingresa un carácter se lee como tipo char.

Otro de los grandes problemas fue a la hora de querer crear matrices dinámicas a través de punteros, ya que en clase se explicó como hacerlo con un array simple, y a la hora de crear un arreglo bidimensional o tridimensional se vuelve más complicado.

5. Evolución del algoritmo y consideraciones a tener en cuenta en la implementación.

Al principio de la solución del parcial, estuvimos experimentando diferentes maneras de verificación de funcionamiento de los 64 leds, esto nos permitió adentrarnos a la sintaxis de este lenguaje y empezamos a definir funciones vitales para el correcto desarrollo del programa. Además de esto, estuvimos experimentando con el monitor serial de tinkercad, haciendo inputs/outputs, reconociendo la sintaxis de esta función (Serial) ya que esta es muy importante.

En los días posteriores, comenzamos a definir funciones para ingresar un patrón, pero esto lo hacíamos sin uso de punteros, debido a que teníamos poco conocimiento sobre ello. Luego de indagar en foros y preguntar a monitor y tutores, logramos usar punteros para el display de un patrón.

Finalmente, logramos mostrar un patrón de matrices reutilizando código de la función imagen(), supimos separar espacio de memoria para un array tridimensional y pudimos finalizar de manera certera el programa.

Respecto a las consideraciones, el usuario deberá ingresar en la función publik() valores reales, los cuales la maquina pueda procesar, es impensable ingresar 1000 matrices y un tiempo de ejecucion de 1 mes, debido al uso de memoria. Además, respecto al input de las filas, el usuario debe saber que este input es de 8 caracteres.