

Parcial 1 Info2

Integrantes:

**Daniel Andres Agudelo Garcia
Andres Felipe Rendon Villada
Esteban Felipe Guiza Piñeros**

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Marzo de 2021

Índice

| | |
|---|----------|
| 1. Introduccion: Analisis del problema | 2 |
| 2. Desarrollo | 3 |
| 2.1. Esquema de desarrollo | 3 |
| 2.2. Algoritmo implementado | 4 |
| 2.3. Problemas que se presentaron | 4 |
| 2.4. Solucion a problematicas presentadas | 5 |
| 3. Conclusión | 7 |

1. Introduccion: Analisis del problema

Leimos el informe del problema y las soluciones que se necesitaban, se hizo un analisis con todos las posibles soluciones que sean factibles para seguir el paso a paso estipilado del informe.

Obervamos la problematica a tener en cuenta, en la cual planteamos diferentes soluciones y elegimos la mas optima, pensando en un desarrollo ideal a la propuesta hecha, formando una idea de proyecto y estructurando las mecanicas y/o codigos a implementar, tomando como eje principal que el usuario pueda ingresar el numero que desee de patrones a generar y cualquier patron.

Al tener las soluciones planteadas, buscamos e investigamos todos los conceptos que vamos a necesitar para el montaje del circuito en tinkercar, al igual que entender el funcionamiento de todos sus componentes para tener conciencia de como lo vamos a utilizar.

2. Desarrollo

2.1. Esquema de desarrollo

Buscamos la solución de programación en c++, implementando las ideas pensadas y realizando un funcionamiento óptimo, cumpliendo las problemáticas propuestas. La solución propuesta es hacer que el usuario sea quien incorpore

el patrón que desea mostrar en los leds, ingresando cada valor uno por uno, el cual le dará libertad de crear cualquier figura que desee.

Luego pensamos el montaje del arduino y todos sus componentes en el cual por medio de un transistor que estará configurado como un switch haciendo que los leds se enciendan o apaguen, dependiendo de lo ingresado por el usuario, estipulado por una determinada señal, con el objetivo de mostrar el patrón ingresado. Por consiguiente se tiene el código funcionando en c++, empezamos

a plantear la construcción del sistema en Tinkercad, cambiando el código en lenguaje de c++ al lenguaje de programación que se maneja en Tinkercad.

2.2. Algoritmo implementado

Codigo c++ *definicion de los puertos* *funciones y configuracion de pines*

2.3. Problemas que se presentaron

Al empezar el montaje del sistema en tinkercar, hubo muchos errores en la conexión de los componentes, obligandonos a recrear en muchas ocasiones la reestructuración del sistema, debido que a veces funcionaba pero para la implementación del código manejarlo con una matriz no sería efectivo y traería futuros problemas.

(Problemas presentados)

Al principio íbamos a usar 8 integrados para las filas y 8 integrados para controlar las señales de las columnas pero luego de analizarlo, nos dimos cuenta que no sería una forma factible para su desarrollo, ya que era más complicado la manipulación del pulso y los datos de entrada, a través del pin de datos "SER" No entendimos como conectar la matriz al comienzo, por lo que buscábamos los diagramas esquemáticos de los leds. *Imagen del arduino con sus componentes de

fondo negro*

Tuvimos dificultades sobre como encender los leds, y luego buscando como mantenerlos encendidos.

Hubo dificultades en el entendimiento del integrado, ya que no contábamos con la experiencia de la electrónica y con un programa como arduino.

Implementacion del codigo en c++, ya que se hicieron dos versiones, una con un triple arreglo y la otra con memoria dinamica, un triple puntero, haciendo que el codigo al implementarlo en arduino nos diera fallas de manejar los datos de la matriz, en c++ el codigo funcionaba con lo esperado pero en arduino no fue asi.

Luego de crear la interfaz, se hicieron las funciones de la matriz para que el usuario fuera rellenandolas con los datos en cada posicion, ya sea que los quiera

(Problema) Posteriormente pensamos implementar transistores con el objetivo de poder switchear la señal para indicar cual led iba a prender, pero se nos fue indicado que no era permitido.

Los patrones ingresados por el usuario quedan guardados en la matriz pero al momento de hacer el recorrido de las filas/columnas y que envíen los pulsos de salida para que se encienda o apague el led esta fallando.

2.4. Solucion a problemáticas presentadas

En la implementacion de las soluciones y empezar a crear el codigo en c++ con arreglo de dos dimensiones, pero el usuario solo podia ingresar un patron, y en caso de que quisiera entrar "Hola" no se podria, por ende se necesito hacer un arreglo de tres dimensiones o un triple puntero, para que pueda guardar cada patron y al final mostrar todos los punteros.

Luego decidimos usar 8 integrados para controlar las 8 filas de la matriz, usando 3 pines digitales del arduino para el control de la señal, para controlar las señales que recibe el integrado. Cada integrado usara la salida inversa de la anterior como dato de entrada del integrado siguiente, excepto el ultimo que tiene el pin de salida inversa libre. Mas adelante se empezo a realizar la codificacion y configuration de los puertos del arduino como salidas digitales.

Al darnos cuenta que el programa no funcionaba ingresando dato por dato de la matriz, en la funcion "publik" habra un arreglo de 8 elementos los cuales son las filas, y cada elemento sera un numero real, que se convertira en binario, enviando los pulsos de 1 y 0 para encender/apagar led.

Despues de montar la idea de trabajar con sistema binario, y montar el codigo en c++, pasarlo a arduino nos trajo multiples errores, ya que no se trabaja igual los datos de entrada y las funciones trabajadas no sirvieron.

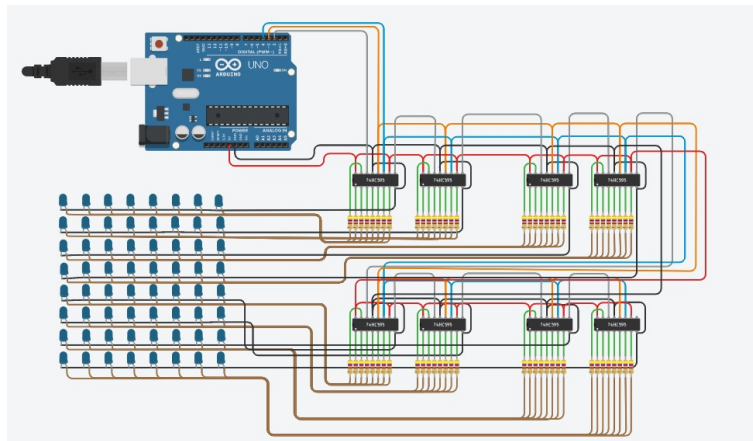


Figura 1: Tinkercar

Pines integrado 1 y 2

3. Conclusión