Informe Arduino y C++

Circuito Tinkercad

Juan Diego Carrera Quintero
Juan José Arboleda Cardona

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

Índice

1. Análisis del problema

1.1. El Problema

Crear un circuito e implementar un programa para que en una matriz 8x8 de leds se muestre un patrón ingresado por una persona; además de eso, el programa debe ser capaz de lo siguiente:

1. Mostrar que los 64 leds funcionen correctamente, encendiéndose por completo.
2. Recibir un patrón por consola y replicarlo tal cual en los leds, como en la figura 1.
3. Mostrar una cadena de patrones, como en la figura 2, en un tiempo, para eso deber recibir:
   - Un número N el cual determina la cantidad de patrones difrentes.
   - N veces los patrones.
   - Un número T el cual determina el tiempo antes de mostrar el siguiente patrón.

1.1.1. Ejemplos

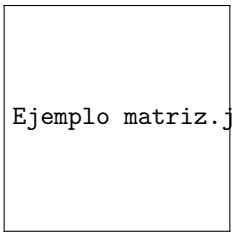

Figura 1: Matriz 8x8

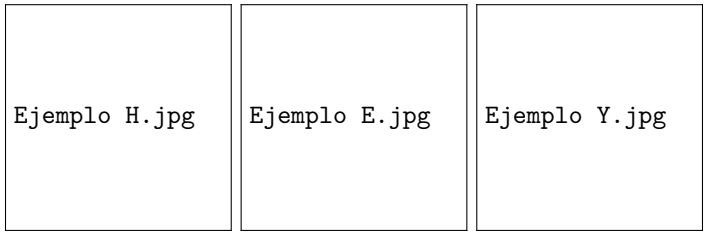

Figura 2: Patrón H-E-Y

1.2. Solución

Con ayuda de un video en **YouTube** [**?**] y un poco de tiempo se logra hacer las conexiones necesarias para que el circuito funcione y día tras día se fue preparando el código para que el circuito funcione, se hacían pruebas en qt, se pasaba a Tinkercad y así hasta llegar al resultado final, figura 3.

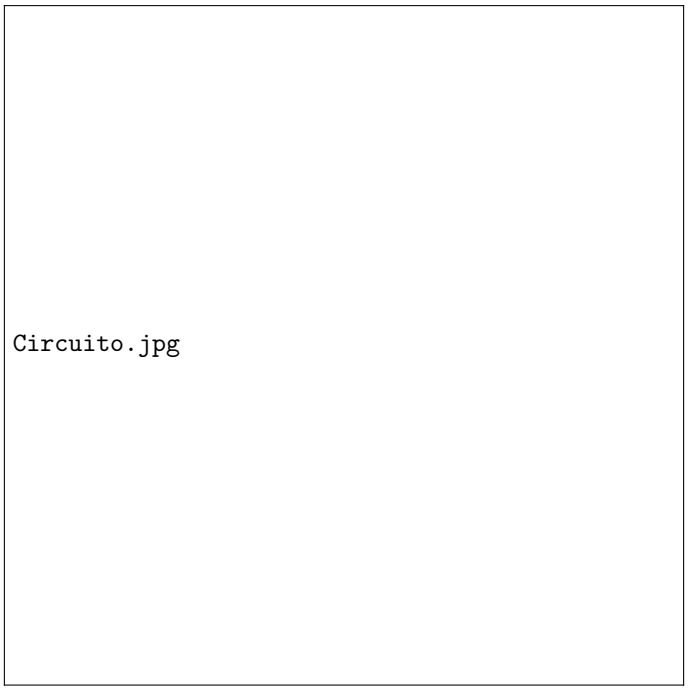

Figura 3: Circuito Solución Final

2. Desarrollo del Algoritmo

Una vez leímos el problema y buscamos información en las clases pasadas e internet montamos el circuito en Tinkercad, primero empezamos con un solo integrado y 8 leds, una vez montado el circuito escribimos el código para que los leds se encendieran, no importaba el orden, solo que encendieran. Luego de varias pruebas, volver a armar el circuito y reescribir el código los primeros 8 leds se encendieron y empezamos a mover números para darle un orden y así poder continuar. con el circuito montado empezamos a hacer el código por partes

Cuando el circuito estaba funcionando correctamente empezamos con la codificación en Qt y cuando éste funcione lo pasamos a Tinkercad. Para realizar las funciones "Verificacion", Ïmagenƨ "Publikrecurrimos a Qt creator e hicimos una simulación de cómo funcionaría el programa en Tinkercad pero con las li-

mitaciones de Qt. Empezamos creando un código que genere la matriz 8x8 el cuál nos serviría para hacer las funciones, una vez el generador de matriz estaba completo empezamos con las funciones. La primera y la más fácil fue la funcion "Verificacion", en Qt para simular los leds encendidos hicimos que en consola imprimiera 1 y para los leds apagados 0, entonces para esta función al momento de generar la matriz hicimos que cada elemento sea igual a 1 y al momento de imprimirla saldría la matriz 8x8 de 1.

Terminando la funcion "Verificar"seguía Ïmagen", copiamos el código de "Verificación"para posteriormente modificarlo, pero aquí las cosas se complicaron un poco porque debíamos pensar cómo recibir el patrón, entonces decidimos que la mejor forma era recibirlo fila por fila, es decir para cada patrón recibiría 8 números, ejemplo: 10011001. Luego de saber cómo recibir el patrón seguía pensar cómo separar cada número y lo primero que se nos ocurrió es que reciba un número entero de 8 cifras así despues aplicando módulo y divisiones se podría acceder a cada elemento, el detalle es que organizaba los números en el orden contrario en que eran ingrsados por fila, así que para solucionar eso alteramos la forma de imprimir las matrices para que se volviera a invertir el orden y así se imprima en la posición original. Entonces metimos la entrada de datos en el primer ciclo de creación de matriz para que así lo pida 8 veces y poder tener el patrón deseado.

Lo último era la función Publik así que copiamos nos guiamos del código de Ïmagenz esta vez usando memoria dinámica creamos las matrices, la diferencia es que primero se crea un arreglo de N elementos que son ingresados por el usuario, despues en ese arreglo creamos las N matrices de 8x8, el resto es igual a Ïmagen"solo que ahora hay que especificar en qué matriz se crea el patrón. Además de pedir el número de patrones a ingresar también se pide un número T que define el tiempo que pasa entre patrón. Aunque el Qt no aplicamos el "delayïmplementé el código que pide T para tenerlo como guía al implementar todo el código en Tinkercad.

Ya teniendo el esqueleto en Qt nos pusimos a pasarlo a Tinkercad, el código es prácticamente el mismo, solo que ahora debíamos cambiar la forma de las entradas, quitar la parte que imprime en consola que sería reemplazada por las funciones de encendr los leds y en la función Publik crear un ciclo infinito el cual repite los patrones con una intermitencia de T milisegundos. Una vez el código estaba en Tinkercad solo nos faltaba una cosa, cómo recibir datos así que le preguntamos a nustros profesores y volvimos a ver las grabaciones para solucionar eso. Cuando encontramos cómo ingresar los datos lo implementamos en el código y nos aseguramos que funcione correctamente. Finalmente con el código funcionando agregamos los comentarios faltantes y más código el cual tiene como función imprimer en el serial datos como un menu, un cartel que diga qué función se está implementando y otros que indiquen cuándo ingresar datos.

3. Algoritmo Solución

El código empieza definiendo los puertos del arduino, el serial como puerto 2, los relojes como 3 y 4 y una variable menu la cuál será usada más adelante. En el setup inicialisamos el serial y configuramos los pines, seguidamente en el loop imprimos algunos carteles que dan información sobre qué ingresar y un menu el cuál dependiendo de lo ingresado por el usuario llamrá una función diferente.

Si el ususario ingresa 1 en el menú, invocará la función Verificar, esta función crea una matriz 8x8 y en cada elemento pone el número 1, después de que la función se creó encenderá todos los leds llamando a la función FilaLeds. Luego de que se enciendan los leds y sale de la función. Despues de que pasen 5 segundos llama a la función LedsOff que apaga los leds. Finalmente vuelve a entrar en el menú esperando a que el usuario ingrese otra opción.

Si el usuario ingresa 2 en el menú, invocará la función Imagen, la cual pide un patrón al usuario para replicarlo encendiendo o apagando los leds. Al entrar en esta función primero se inicializan dos variables, res y led y se crea una matriz 8x8, le especifica al usuario cómo funcionan los leds, si ingresa 1 se enciende, y si ingresa 0 se apaga, sabiendo esto el usuario deberá ingresar un número de 8 cifras que corresponden a los 8 leds de la primera fila el cual se almacena en la variable led y lo sigue haciendo 7 veces más para así llenar la matriz. Después de pedir cada vez los números entra en otro ciclo donde al número ingresado le aplica el opeprador módulo 10 para obtener el último dígito del número y lo guarda en la variable res, después a led lo divide entre 10 y el resultado queda en led eliminando así el número anterior, finalmente el valor de res queda el matriz en la posición 1 1 y se repite ocho veces para obtener la primera fila, después vulve a pedir al usuario que ingrese otro número de 8 dígitos y el proceso se repite hasta que se llena la matriz. Despues vuelve a llamar la función FilaLeds ocho veces para replicar la matriz en los leds. Despues sale de la función, espera 5 segundos e invoca la función LedsOff y vuelve a entrar en el menu.

Si el usuario ingresa 3 entra en la función Publik, aquí aquí se inicializan las variables led y res al igual que anteriormente, pepro también están size, time y T que vale 8.

Primero le pide al usuario que ingrese cuántos patrones va a meter y los guarda en la variable size, después le pide un tiempo en milisegundos el cual determina cuánto tiempo hay entre patrón y patron y lo guarda en time. Usando memoria dinámica crea un arreglo de tamaño size y dentro crea tantas matrices 8x8 de acuerdo a size. Aquí se repite lo mismo que en Imagen, se crean dos ciclos, en uno pide un número de 8 dígitos y en otro organiza los 1 y 0 en las matrices.

después de crear el arreglo con size matrices entra a un ciclo infinito donde replica en los leds el primer patron, espera time milisegundos y pasa al segundo, vuelve a esperar time milisegundo y así sucesivamente hasta replicar todos los patrones. Al terminar el último patron la variable i del ciclo vuelve a 0 para volver al primer patron. Este ciclo se ejecuta hasta que se detenga la simulacion.

Finalmente tenemos tres funciones:

1. Funcion LedsOff: Al igual que Verificación genera una matriz 8x8, pero esta vez cada elemento es igual a 0, así cuando llama a FilaLeds todos se quedan apagados.

2. Funcion FilaLeds: esta función al ser ejecutada enciende o apaga los leds de una sola fila, por eso se llama ocho veces, la diferencia de esta función del resto es que ésta recibe o parámtetros, uno por cada led de la fila, así enciende unos y apaga otros; esta función trabaja correctamente porque lo hace en conjunto con LedsOn.

3. LedsOn: Esta función enciende un solo led, por eso recibe un solo parámetro, sea 1 o 0. El número que reciba la función va se envía por el pin serial haciendo que el led se encienda o apague, despuéstan los dos relojes, estos hacen que el número que le llegue a la función se mueva al siguiente led cada vez que le llega un número nuevo.

4. Problemas de Desarrollo

Durante el desarollo del programa se presentaron dificultades en:

1. Montar el Arduino en tinkerkat con sus respectivas conexiones para el funcionamiento de los 64 leds utilizando el Integrado 74HC959 sin salir de los limites de las restricciones.

2. Implementar una funcion que pudiera suplir los requerimientos pedidos en la necesidad planteada , como son la funcion 'verificacion', 'imagen' y 'publik'.

3. Implementacion del algoritmo solucion en la plataforma tinkerkat.

5. Evolución del Algoritmo

Una vez implementadas las conexiones en el arduino desarrollamos e ideamos soluciones para la implementacion de un algoritmo solucion evolucionando asi:

1. Para el ingreso de los patrones se pensó hacer 2 arreglos de dos dimensiones , uno tendria todas las letars del abecedario con sus respectivos patrones para la representacion en los leds y otro contendria los patrones para los respectivos numeros de la misma forma. para diferenciar caracteres se utilizarian los valores ASCII para saber cual arreglo de patrones corresponde un caracter, ya sea letra o numero, y en que poscicion de el arreglo se encontraria.
   Esta idea Se descarto debido a la excesiva memoria que utilizaria (se penso mas a favor de el Arduino) , ademas que este restringe el ingreso a solo letras y numeros, no caracteres especiales (como caritas felices o caracteres especiales).

2. Luego se planteo implementar un tablero de botones de 8 x 8 ,respectivos a cada led, en el cual el usuario, al momento de interactuar con un boton, ingresaria el patron deseado marcando cual led queria encender y a su vez guardar el patron para su debida visualizacion.
   Este planteamiento no se implemento debido a que se salia de las restricciones para la implementacion del circuito, ya que no incluia el uso de botones u otros elemntos de los circuitos de tinker kat diferentes a los explicados.

3. Se implemento una funcion en c++ de un arreglo de 2 dimensiones, el cual contendria 'n' arreglos de 8 x 8 (n es ingresado por el Usuario) , luego por el puerto ingresaria por filas 1 ó 0 , para prender un led o apagarlo y asi ir formando el patron deseado por el usuario sin restringir a un patro pre-definido.
   esta idea luego se mejoró, ya que el ingreso de 64 numeros , uno por uno, es un tanto complejo y hasta enredado.

4. teniendo en cuenta la funcion anterior y el proceso que llevaba esta para ingresar terminos, se cambio el ingreso numero por numero a ingresar directamente una linea entera de leds y la funcion haria el proceso de sacar numero por numero e irlo guardando en los arreglos .
   gracias a esto se pasó de ingresar 64 x n veces numeros a ingresar linea por linea de leds y asi ir formando los n patrones que el usuario requiera, usando la misma sintaxis que en la primera version de la funcion con 0 y 1 para los estados de los leds.