

Parcial I Informtica II

Informe

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	17
2.4. Solucion a problematicas presentadas	19
3. Conclusion	24
4. Manual de uso	25

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 consecuente se tiene el código funcionando en c++, empezamos a planear 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

Primer codigo EN C++, con memoria dinamica

El primer codigo que se penso, para que el usuario ingresara cada posicion con 1 y 0, utilizando memoria dinamica.

```
#include <iostream>

using namespace std;
int ***matriz_dinamica;
int a;
void InicializacionMemoria();
void datos_matriz();
void imprimir();
void imagen();
void imprimir();

int main()
{

    cout << "PARCIAL INFORMATICA II 2021" << endl;
    cout<<"Cantidad de caracteres: ";cin>>a;
    InicializacionMemoria();
    datos_matriz();
    imprimir();
    imagen();
    imprimir();

    delete [] matriz_dinamica;
}

void InicializacionMemoria(){
    matriz_dinamica=new int**[a];
    for(int i=0;i<8;i++){
        matriz_dinamica[i]=new int*[8];
        for(int j=0;j<8;j++){
            matriz_dinamica[i][j]=new int[8];
        }
    }
}
```

```

}

void datos_matriz(){
    int b=1;//Ingresar los datos para el patron
    for(int k=0;k<a;k++){ //Cantidad de caracteres
        cout<<"Ingrese el patron del caracter #"<<b<<endl;
        b++;
        for(int i=0;i<8;i++){ //Filas
            for(int j=0;j<8;j++){ //Columnas
                cout<<"Ingrese el valor de la posicion: "<<"[ "<<i<<" ]"<<"[ "<<j<<" ]: ";
                cin>>matriz_dinamica[k][j][i]; //Irls colocando
            }
        }
        cout<<endl<<"-----"<<endl;
    }
}

void imprimir(){ //imprimir
    int b=1;
    for (int k=0; k<a; k++){//Cantidad de caracteres
        cout<<"Patron del caracter #"<<b<<endl;
        b++;
        for(int i=0;i<8;i++){ //Filas
            for(int j=0;j<8;j++){ //Columnas
                cout<<"["<<matriz_dinamica[k][i][j]<<"]"; //muestre lo que hay en la matriz
            }
            cout<<endl;
        }
        cout<<endl<<"-----"<<endl;
    }
}

void imagen(){ //Patron
    for(int k=0; k<a; k++){ //cantidad de caracteres
        for( int i=0; i<8;i++){ //flas
            for(int j=0; j<8 ; j++){ //columnas
                if (matriz_dinamica[k][i][j]==49) matriz_dinamica[k][i][j]='*'; //convertir en *
                else matriz_dinamica[k][i][j]=' ';
            }
        }
    }
}

```

Manejo de pines en arduino

Funcionamiento e los pines en la conexion del arduino, notas de como se usan.

Datasheet 74HC595
pines de conexion
pin 16 VCC entrada de voltaje positivo
pin 8 tierra
pin 10 SCLR a POSITIVO
pin 11, 12 y 14? entrada digital del arduino
resistencia de 560 Y LEDS
5 señales de control
RCLK ESTA (conectada a una entrada digital del arduino)
SER ESTA (conectada a una entrada digital del arduino)
SRCLK ESTA PARA CONTROL (conectada a una entrada del arduino)
SRCLR ALTO (POSITIVO)
OE BAJO (NEGATIVO)

conexiones de entrada
SRCLK (SRCLK,RCLK) TIPO DE SEÑAL (0,1,0)
SER TIPO DE SEÑAL (0 o 1)
SRCLR

LOS DATOS SE INGRESAN DE IZQ A DER por el ser
ACTIVACION POR FLANCO (0,1,0)

Codigo de prueba para el funcionamiento de la matriz, en todas sus entradas, y enviar los pulsos para entrar o salir

En este codigo se tenia como premisa, que el usuario ingresara cada componente, y si es 1 enciende el led y si es 0 apaga el led.

```
int pinData = 2;//SER
int pinLatch = 3;//RCLK
int pinClock = 4;//SRCLK
#define tiempo 2
```

```

void ledWrite(int p1, int p2, int p3, int p4, int p5, int p6, int p7, int p8){
    shiftOut(pinData, pinClock, MSBFIRST, p1);
    shiftOut(pinData, pinClock, MSBFIRST, p2);
    shiftOut(pinData, pinClock, MSBFIRST, p3);
    shiftOut(pinData, pinClock, MSBFIRST, p4);
    shiftOut(pinData, pinClock, MSBFIRST, p5);
    shiftOut(pinData, pinClock, MSBFIRST, p6);
    shiftOut(pinData, pinClock, MSBFIRST, p7);
    shiftOut(pinData, pinClock, MSBFIRST, p8);
    digitalWrite(pinLatch, HIGH);
    digitalWrite(pinLatch, LOW);
}

void setup(){
    pinMode(pinData, OUTPUT);
    pinMode(pinLatch, OUTPUT);
    pinMode(pinClock, OUTPUT);
}

void loop(){

    ledWrite(0,0,0,0,0,0,0,1); delay(tiempo);
    ledWrite(0,0,0,0,0,0,0,2); delay(tiempo);
    ledWrite(0,0,0,0,0,0,0,4); delay(tiempo);
    ledWrite(0,0,0,0,0,0,0,8); delay(tiempo);
    ledWrite(0,0,0,0,0,0,0,16); delay(tiempo);
    ledWrite(0,0,0,0,0,0,0,32); delay(tiempo);
    ledWrite(0,0,0,0,0,0,0,64); delay(tiempo);
    ledWrite(0,0,0,0,0,0,0,128); delay(tiempo);
    ledWrite(0,0,0,0,0,0,1,0); delay(tiempo);
    ledWrite(0,0,0,0,0,0,2,0); delay(tiempo);
    ledWrite(0,0,0,0,0,0,4,0); delay(tiempo);
    ledWrite(0,0,0,0,0,0,8,0); delay(tiempo);
    ledWrite(0,0,0,0,0,0,16,0); delay(tiempo);
    ledWrite(0,0,0,0,0,0,32,0); delay(tiempo);
    ledWrite(0,0,0,0,0,0,64,0); delay(tiempo);
    ledWrite(0,0,0,0,0,0,128,0); delay(tiempo);
    ledWrite(0,0,0,0,0,1,0,0); delay(tiempo);
    ledWrite(0,0,0,0,0,2,0,0); delay(tiempo);
    ledWrite(0,0,0,0,0,4,0,0); delay(tiempo);
    ledWrite(0,0,0,0,0,8,0,0); delay(tiempo);
    ledWrite(0,0,0,0,0,16,0,0); delay(tiempo);
    ledWrite(0,0,0,0,0,32,0,0); delay(tiempo);
    ledWrite(0,0,0,0,0,64,0,0); delay(tiempo);
    ledWrite(0,0,0,0,0,128,0,0); delay(tiempo);

```



```

ledWrite(0,0,0,0,1,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,2,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,4,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,32,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,64,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,128,0,0,0); delay(tiempo);
ledWrite(0,0,0,1,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,2,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,4,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,32,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,64,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,128,0,0,0,0); delay(tiempo);
ledWrite(0,0,1,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,2,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,8,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,16,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,64,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,128,0,0,0,0,0); delay(tiempo);
ledWrite(0,1,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,4,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,8,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,16,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,32,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,128,0,0,0,0,0,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(2,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(4,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(8,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(16,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(32,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(64,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,0); delay(tiempo);

ledWrite(0,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(0,0,0,0,0,0,1,0); delay(tiempo);
ledWrite(0,0,0,0,0,1,0,0); delay(tiempo);
ledWrite(0,0,0,0,1,0,0,0); delay(tiempo);
ledWrite(0,0,0,1,0,0,0,0); delay(tiempo);

```

```

ledWrite(0,0,1,0,0,0,0,0); delay(tiempo);
ledWrite(0,1,0,0,0,0,0,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,2); delay(tiempo);
ledWrite(0,0,0,0,0,0,2,0); delay(tiempo);
ledWrite(0,0,0,0,0,2,0,0); delay(tiempo);
ledWrite(0,0,0,0,2,0,0,0); delay(tiempo);
ledWrite(0,0,0,2,0,0,0,0); delay(tiempo);
ledWrite(0,0,2,0,0,0,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,0,0); delay(tiempo);
ledWrite(2,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,4); delay(tiempo);
ledWrite(0,0,0,0,0,0,4,0); delay(tiempo);
ledWrite(0,0,0,0,0,4,0,0); delay(tiempo);
ledWrite(0,0,0,0,4,0,0,0); delay(tiempo);
ledWrite(0,0,0,4,0,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,0,0,0); delay(tiempo);
ledWrite(0,4,0,0,0,0,0,0); delay(tiempo);
ledWrite(4,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,8); delay(tiempo);
ledWrite(0,0,0,0,0,0,8,0); delay(tiempo);
ledWrite(0,0,0,0,0,8,0,0); delay(tiempo);
ledWrite(0,0,0,0,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,0,0,0,0); delay(tiempo);
ledWrite(0,0,8,0,0,0,0,0); delay(tiempo);
ledWrite(0,8,0,0,0,0,0,0); delay(tiempo);
ledWrite(8,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,16); delay(tiempo);
ledWrite(0,0,0,0,0,0,16,0); delay(tiempo);
ledWrite(0,0,0,0,0,16,0,0); delay(tiempo);
ledWrite(0,0,0,0,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,0,0,0,0); delay(tiempo);
ledWrite(0,0,16,0,0,0,0,0); delay(tiempo);
ledWrite(0,16,0,0,0,0,0,0); delay(tiempo);
ledWrite(16,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,32); delay(tiempo);
ledWrite(0,0,0,0,0,0,32,0); delay(tiempo);
ledWrite(0,0,0,0,0,32,0,0); delay(tiempo);
ledWrite(0,0,0,0,32,0,0,0); delay(tiempo);
ledWrite(0,0,0,32,0,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,0,0,0); delay(tiempo);
ledWrite(0,32,0,0,0,0,0,0); delay(tiempo);
ledWrite(32,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,64); delay(tiempo);
ledWrite(0,0,0,0,0,0,64,0); delay(tiempo);
ledWrite(0,0,0,0,0,64,0,0); delay(tiempo);

```

```

ledWrite(0,0,0,0,64,0,0,0); delay(tiempo);
ledWrite(0,0,0,64,0,0,0,0); delay(tiempo);
ledWrite(0,0,64,0,0,0,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,0,0,0); delay(tiempo);
ledWrite(64,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(0,0,0,0,0,0,128,0); delay(tiempo);
ledWrite(0,0,0,0,0,128,0,0); delay(tiempo);
ledWrite(0,0,0,0,128,0,0,0); delay(tiempo);
ledWrite(0,0,0,128,0,0,0,0); delay(tiempo);
ledWrite(0,0,128,0,0,0,0,0); delay(tiempo);
ledWrite(0,128,0,0,0,0,0,0); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,0); delay(tiempo);

```

```

ledWrite(0,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,2); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,4); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,8); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,16); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,32); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,64); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(0,0,0,0,0,0,128,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,64,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,32,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,16,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,8,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,4,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,2,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,1,0); delay(tiempo);
ledWrite(0,0,0,0,0,1,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,2,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,4,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,8,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,16,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,32,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,64,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,128,0,0); delay(tiempo);
ledWrite(0,0,0,0,128,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,64,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,32,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,4,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,2,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,1,0,0,0); delay(tiempo);

```

```

ledWrite(0,0,0,1,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,2,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,4,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,32,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,64,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,128,0,0,0,0); delay(tiempo);
ledWrite(0,0,128,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,64,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,16,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,8,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,2,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,1,0,0,0,0,0); delay(tiempo);
ledWrite(0,1,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,4,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,8,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,16,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,32,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,128,0,0,0,0,0,0); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(64,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(32,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(16,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(8,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(4,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(2,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,0); delay(tiempo);

ledWrite(0,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(0,0,0,0,0,0,1,0); delay(tiempo);
ledWrite(0,0,0,0,0,1,0,0); delay(tiempo);
ledWrite(0,0,0,0,1,0,0,0); delay(tiempo);
ledWrite(0,0,0,1,0,0,0,0); delay(tiempo);
ledWrite(0,0,1,0,0,0,0,0); delay(tiempo);
ledWrite(0,1,0,0,0,0,0,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(2,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,2,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,2,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,2,0,0,0); delay(tiempo);

```

```

ledWrite(0,0,0,0,0,2,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,2,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,2); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,4); delay(tiempo);
ledWrite(0,0,0,0,0,0,4,0); delay(tiempo);
ledWrite(0,0,0,0,0,4,0,0); delay(tiempo);
ledWrite(0,0,0,0,4,0,0,0); delay(tiempo);
ledWrite(0,0,0,4,0,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,0,0,0); delay(tiempo);
ledWrite(0,4,0,0,0,0,0,0); delay(tiempo);
ledWrite(4,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(8,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,8,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,8,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,8,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,8,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,8); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,16); delay(tiempo);
ledWrite(0,0,0,0,0,0,16,0); delay(tiempo);
ledWrite(0,0,0,0,0,16,0,0); delay(tiempo);
ledWrite(0,0,0,0,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,0,0,0,0); delay(tiempo);
ledWrite(0,0,16,0,0,0,0,0); delay(tiempo);
ledWrite(0,16,0,0,0,0,0,0); delay(tiempo);
ledWrite(16,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(32,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,32,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,32,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,32,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,32,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,32,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,32); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,64); delay(tiempo);
ledWrite(0,0,0,0,0,0,64,0); delay(tiempo);
ledWrite(0,0,0,0,0,64,0,0); delay(tiempo);
ledWrite(0,0,0,0,64,0,0,0); delay(tiempo);
ledWrite(0,0,64,0,0,0,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,0,0,0); delay(tiempo);
ledWrite(64,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,128,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,128,0,0,0,0,0); delay(tiempo);

```

```

ledWrite(0,0,0,128,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,128,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,128,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,128,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,128); delay(tiempo);

```

```

ledWrite(0,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(0,0,0,0,0,0,2,0); delay(tiempo);
ledWrite(0,0,0,0,0,4,0,0); delay(tiempo);
ledWrite(0,0,0,0,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,0,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,0,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,0,0,0); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(0,0,0,0,0,64,0); delay(tiempo);
ledWrite(0,0,0,0,0,32,0,0); delay(tiempo);
ledWrite(0,0,0,0,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,0,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,0,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,0,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(0,0,0,0,0,0,2,0); delay(tiempo);
ledWrite(0,0,0,0,0,4,0,0); delay(tiempo);
ledWrite(0,0,0,0,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,0,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,0,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,0,0,0); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,0); delay(tiempo);
ledWrite(0,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(0,0,0,0,0,64,0); delay(tiempo);
ledWrite(0,0,0,0,0,32,0,0); delay(tiempo);
ledWrite(0,0,0,0,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,0,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,0,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,0,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,0); delay(tiempo);

```

```

ledWrite(1,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(0,2,0,0,0,0,2,0); delay(tiempo);
ledWrite(0,0,4,0,0,4,0,0); delay(tiempo);
ledWrite(0,0,0,8,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,16,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,32,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,64,0); delay(tiempo);

```

```

ledWrite(128,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(0,64,0,0,0,0,64,0); delay(tiempo);
ledWrite(0,0,32,0,0,32,0,0); delay(tiempo);
ledWrite(0,0,0,16,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,8,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,4,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,2,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,1); delay(tiempo);
ledWrite(0,2,0,0,0,0,2,0); delay(tiempo);
ledWrite(0,0,4,0,0,4,0,0); delay(tiempo);
ledWrite(0,0,0,8,8,0,0,0); delay(tiempo);
ledWrite(0,0,0,16,16,0,0,0); delay(tiempo);
ledWrite(0,0,32,0,0,32,0,0); delay(tiempo);
ledWrite(0,64,0,0,0,0,64,0); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(128,0,0,0,0,0,0,128); delay(tiempo);
ledWrite(0,64,0,0,0,0,64,0); delay(tiempo);
ledWrite(0,0,32,0,0,32,0,0); delay(tiempo);
ledWrite(0,0,0,16,16,0,0,0); delay(tiempo);
ledWrite(0,0,0,8,8,0,0,0); delay(tiempo);
ledWrite(0,0,4,0,0,4,0,0); delay(tiempo);
ledWrite(0,2,0,0,0,0,2,0); delay(tiempo);
ledWrite(1,0,0,0,0,0,0,1); delay(tiempo);
}

```

CODIGO CON DIFERENTE FORMA DE ENTRADA Y DE GUARDAR EN LA MATRIZ: C++

Este fue un codigo que se tenia como iniciativa que el usuario ingresara los datos de entrada en forma binaria y los llevara y los almacenara en la matriz, convirtiendolos en decimales y la funcion verificar del programa de arduino los leyerá, por falta de tiempo no se pudo incluir para que el usuario entrara la figura con 1 y 0.

```

#include <iostream>

using namespace std;
void Inicializacion(int );
int Conversion(int );
int **ptr;

```

```

int cont=0;

int main()
{
    int a, binario=0;
    cout<<"Ingrese numero de patrones: "; cin>>a;
    Inicializacion(a);
    for(int k=0;k<a;k++){
        cout<<"El patron # "<<k<<endl;
        for(int i=0;i<8;i++){
            cout<<"Ingrese un numero binario: "; cin>>binario;
            ptr[k][i]=Conversion(binario);
        }
    }
    cout<<endl<<"-----"<<endl;
    for(int k=0;k<a;k++){
        cout<<"El patron # "<<k<<endl;
        for(int i=0;i<8;i++){
            cout<<ptr[k][i]<<"-";
        }
        cout<<endl;
    }
}

void Inicializacion(int a){
    ptr=new int*[a];
    for(int i=0;i<a;i++){
        ptr[i]=new int[8];
    }
}

int Conversion(int binario){
    int resultado, resto=0, digito[8];
    for (int i=0; i<8; i++){
        digito[i]=binario%10;
        binario/=10;
    }
    for (int i=7; i>=0;i--){
        resultado=(resto*2)+digito[i];
        resto=resultado;
    }
    cout<<endl<<" Conversion es: "<<resultado<<endl;
    return resultado;
}

```



```

1 int pinData = 2; //SER
2 int pinLatch = 3; //RCLK
3 int pinClock = 4; //SRCLK
4 #define tiempo 2
5
6 void verificacion(int p1, int p2, int p3, int p4, int p5, int p6, int p7, int p8){
7     shiftOut(pinData, pinClock, MSBFIRST, p1);
8     shiftOut(pinData, pinClock, MSBFIRST, p2);
9     shiftOut(pinData, pinClock, MSBFIRST, p3);
10    shiftOut(pinData, pinClock, MSBFIRST, p4);
11    shiftOut(pinData, pinClock, MSBFIRST, p5);
12    shiftOut(pinData, pinClock, MSBFIRST, p6);
13    shiftOut(pinData, pinClock, MSBFIRST, p7);
14    shiftOut(pinData, pinClock, MSBFIRST, p8);
15    digitalWrite(pinLatch, HIGH);
16    digitalWrite(pinLatch, LOW);
17 }
18
19 void setup(){
20     pinMode(pinData, OUTPUT);
21     pinMode(pinLatch, OUTPUT);
22     pinMode(pinClock, OUTPUT);
23 }
24

```

Figura 1: Definicion de los puertos

```

void verificacion(int p1, int p2, int p3, int p4, int p5, int p6, int p7, int p8){
    shiftOut(pinData, pinClock, MSBFIRST, p1);
    shiftOut(pinData, pinClock, MSBFIRST, p2);
    shiftOut(pinData, pinClock, MSBFIRST, p3);
    shiftOut(pinData, pinClock, MSBFIRST, p4);
    shiftOut(pinData, pinClock, MSBFIRST, p5);
    shiftOut(pinData, pinClock, MSBFIRST, p6);
    shiftOut(pinData, pinClock, MSBFIRST, p7);
    shiftOut(pinData, pinClock, MSBFIRST, p8);
    digitalWrite(pinLatch, HIGH);
    digitalWrite(pinLatch, LOW);
}

void setup(){
    pinMode(pinData, OUTPUT);
    pinMode(pinLatch, OUTPUT);
    pinMode(pinClock, OUTPUT);
}

```

Figura 2: 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.

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.

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.

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

Luego de crear la interfaz, se hicieron las funciones de la matriz para que el usuario fuera rellenándolas con los datos en cada posición, para imprimir el patrón que deseara.

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 estaba fallando.

Los datos que ingresados no se interpretan de la lógica estudiada, por ejemplo si se ingresa un dato entero el programa lo toma en número ASCII.

2.4. Solucion a problematicas 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.

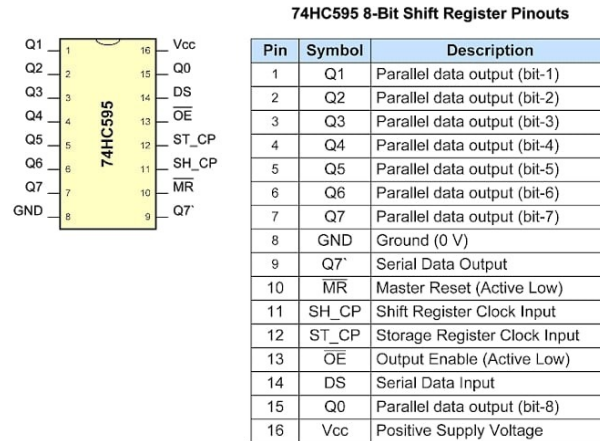


Figura 3: Pines del integrado

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 cumplieron llo esperado.

Entrar numeros binarios y hacer la conversion a entero para que el arreglo lo leyera no fue posible por falta de tiempo, entonces buscamos que el usuario entrara los numeros decimales, que es el equivalente a los 8 bits, para que fuera leyendo fila por fila.

Como el arreglo era leido de derecha a izquierda y se nos estaba agotando el tiempo para la entrega, se le dio un giro de 180 grados al arduino para que imprimiera el patron en la forma indicada.

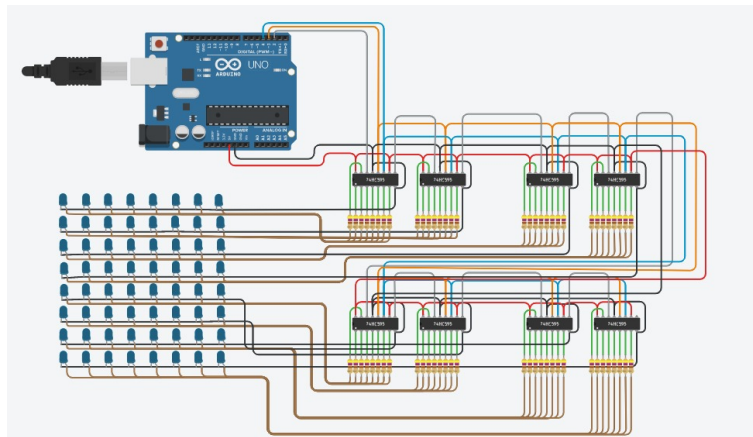


Figura 4: Tinkercad

Codigo de tinkercad listo

CODIGO DEFINITIVO

```
int pinData = 2;//SER
int pinLatch = 3;//RCLK
int pinClock = 4;//SRCLK
#define tiempo 400
```

```
int **Patrones=NULL;
int n=0;
```

```
void publik(int &n,int **patrones){
    for(int i=0;i<n;i++){
        for(int j=0;j<8;j++){
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][0]);delay(tiempo);
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][1]);delay(tiempo);
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][2]);delay(tiempo);
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][3]);delay(tiempo);
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][4]);delay(tiempo);
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][5]);delay(tiempo);
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][6]);delay(tiempo);
            shiftOut(pinData, pinClock, MSBFIRST, Patrones[i][7]);delay(tiempo);
            digitalWrite(pinLatch, HIGH);
            digitalWrite(pinLatch, LOW);
        }
        shiftOut(pinData,pinClock,MSBFIRST, 0);
        shiftOut(pinData,pinClock,MSBFIRST, 0);
        shiftOut(pinData,pinClock,MSBFIRST, 0);
        shiftOut(pinData,pinClock,MSBFIRST, 0);
        shiftOut(pinData,pinClock,MSBFIRST, 0);
        shiftOut(pinData,pinClock,MSBFIRST, 0);
        shiftOut(pinData,pinClock,MSBFIRST, 0);
        shiftOut(pinData,pinClock,MSBFIRST, 0);
    }
}
```

```
void imagen(int &n){
    Serial.println("Bienvenido a la funcion Imagen ingrese el valor de la primer posicion: ");
    //while(Serial.available()<=0){
    //}
    Patrones=new int*[n];
    for(int i=0;i<n;i++){
        Patrones[i]=new int[8];
        for(int j=0;j<8;j++){
            while(Serial.available()<=0){
```

```

    }
    Serial.print("Ingrese el valor de: ");
    Serial.print("[");
    Serial.print(i);
    Serial.print("]");
    Serial.print("[");
    Serial.print(j);
    Serial.print("]");
    Serial.print(": ");
    Serial.println();
    Patrones[i][j]=Serial.parseInt();
    Serial.println(Patrones[i][j]);
}

}

}

int verificacion(int p1,int p2,int p3,int p4,int p5, int p6, int p7, int p8){
    shiftOut(pinData, pinClock, MSBFIRST, p1);
    shiftOut(pinData, pinClock, MSBFIRST, p2);
    shiftOut(pinData, pinClock, MSBFIRST, p3);
    shiftOut(pinData, pinClock, MSBFIRST, p4);
    shiftOut(pinData, pinClock, MSBFIRST, p5);
    shiftOut(pinData, pinClock, MSBFIRST, p6);
    shiftOut(pinData, pinClock, MSBFIRST, p7);
    shiftOut(pinData, pinClock, MSBFIRST, p8);
    digitalWrite(pinLatch, HIGH);
    digitalWrite(pinLatch, LOW);
}

void setup()
{
    pinMode(pinData,OUTPUT);
    pinMode(pinLatch,OUTPUT);
    pinMode(pinClock,OUTPUT);
    Serial.begin(9600);
}

void loop(){
    Serial.println("MENU DE FUNCIONAMIENTO");
    Serial.println("1. FUNCION DE VERIFICACION");
    Serial.println("2. FUNCION IMAGEN");
    Serial.println("3. FUNCION PUBLIK");
    while(Serial.available()==0){
    }
}

```

```

int opcion=Serial.parseInt();
switch(opcion){
case 1:
    Serial.println("EJECUTANDO VERIFICACION");
    //for(int j=0;j<9;j++){
        //int pot=pow(2,j);
        verificacion(0,0,0,0,0,0,0,255);delay(tiempo);
        verificacion(0,0,0,0,0,0,255,0);delay(tiempo);
        verificacion(0,0,0,0,0,255,0,0);delay(tiempo);
        verificacion(0,0,0,0,255,0,0,0);delay(tiempo);
        verificacion(0,0,255,0,0,0,0,0);delay(tiempo);
        verificacion(0,255,0,0,0,0,0,0);delay(tiempo);
        verificacion(255,0,0,0,0,0,0,0);delay(tiempo);
    //}
    verificacion(0,0,0,0,0,0,0,0);
    break;
case 2:
    Serial.println("cantidad de patrones que desea ingresar: ");
    while(Serial.available()<=0){
    }
    n=Serial.parseInt();
    Serial.println(n);
    imagen(n);
    break;
case 3:
    publik(n,Patrones);
    verificacion(0,0,0,0,0,0,0,0);
    break;
}
}

```


3. Conclusion

Se soluciono el problema planteado con todas las problemáticas estipuladas y siguiendo los parámetros señalados. Se necesitó de mucha investigación para la estructura del arduino, en el uso de sus componentes, conociendo su manejo e infraestructura, al igual que la sintaxis que maneja arduino, que es diferente a c++, que fue donde planteamos la idea de desarrollo.

Se falló en múltiples ocasiones en la infraestructura del arduino y en la implementación del código, pero fue necesario para comprender su funcionamiento.

4. Manual de uso

El programa sirve para usted pueda ingresar patrones y por medio de 64 leds va a mostrar dichos patrones.

MENU

En el menu habran 4 opciones y el usuario debe ingresar el numero de la opcion a ejecutar :

1. Verificar: Comprueba que todos los leds enciendan.
2. Imagen: Agregar el patron que desea mostrar en los leds.
3. Publik: Mostrar el patron ingresado en "Imagen" (en caso que no halla ingresado el patron no se encendera ningun led).
- 4.Reiniciar: Eliminar los patrones ingresados en la opcion 2.

EL USUARIO DESEA INGRESAR PATRON(ES) Y OBSERVARLOS EN LOS LEDS

Opcional: Darle en la opcion 1, para verificar que todos los leds enciendan.

1. Darle en el menu la opcion 2, para empezar a ingresar el/los patrones.
2. El programa le va a pedir la cantidad de patrones a ingresar.
3. El programa le va a empezar a pedir un numero entero entre 0 y 255, 8 veces, que son las 8 filas de los leds.
4. Tomando como base que 1 es encendido y 0 es apagado, ingrese el numero entero que signifique en codigo binario los leds que desea tener encendidos o apagados, teniendo en cuenta que se lee de derecha a izquierda.
Ejemplo: si en la primera fila desea encender las columnas 3,5,7 eso significaria en binario: 01010100 , pero el usuario deberia ingresar su equivalencia en entero que en este caso seria: 84.

5. En caso de que haya pedido mas de 1 patron a dibujar, apenas termine de ingresar los 8 elementos, el programa le seguira pidiendo los otros 8 elementos, hasta que termine el numero de patrones que se ingreso en un principio.

6. El programa le retornara al menu, e ingrese la opcion 3 (Publik) donde le empezara a encender los leds correspondientes a los patrones ingresados.