**18/04/2021**

**Ideas**

* Hacer menú
* Verificación: Prender todos los LEDS al mismo tiempo.
* Prender LEDS uno a uno sin apagar el anterior, siguiendo el ejemplo dado en el parcial:



**Análisis y propuesta de solución:**

Se plantea el uso de 2 Circuitos Integrados 74HC595 para recibir la información de los leds que se deben encender, diferenciados por filas y columnas. Esta diferenciación se hace con el fin de poder optimizar el proceso, formando parejas entre el número de fila y columna para poder trabajar la matriz como una matriz de 8 filas x 8 columnas, donde cada elemento de la misma representa un led. Adicionalmente, se implementa este sistema para reducir el número de resistencias, conexiones y circuitos integrados, aumentando así la eficiencia.

Foto de las primeras conexiones:

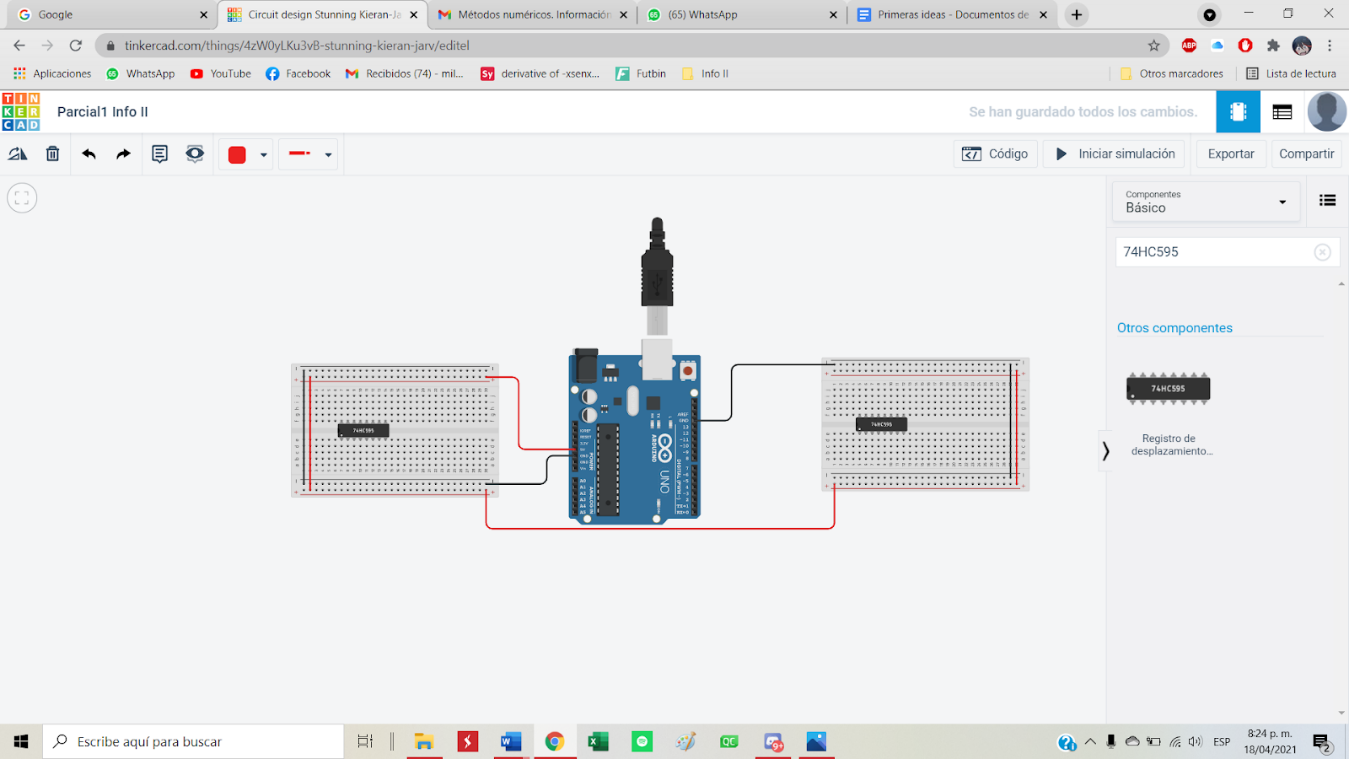
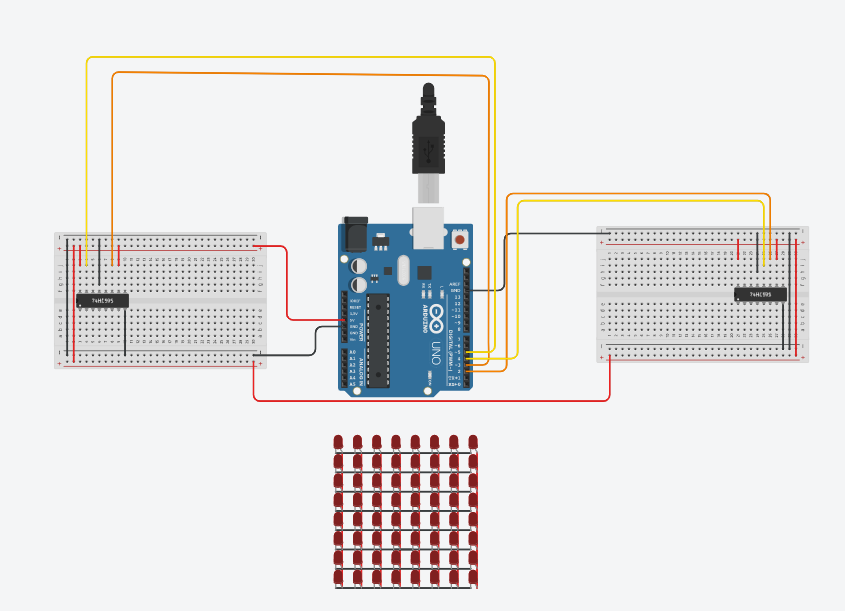
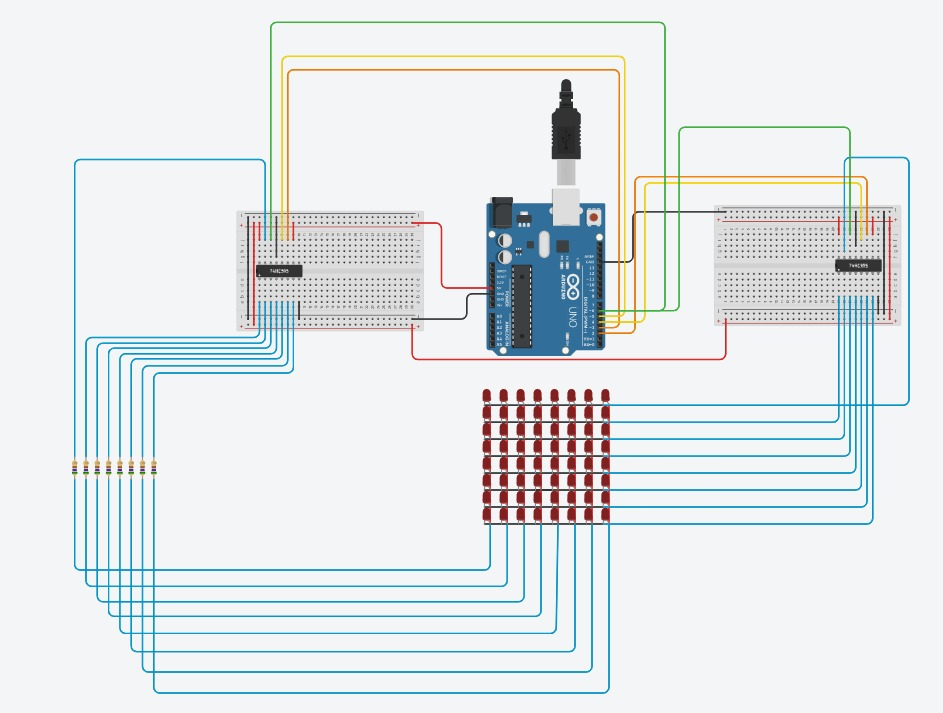


Foto de interconexiones entre los LEDS y conexiones del circuito integrado:

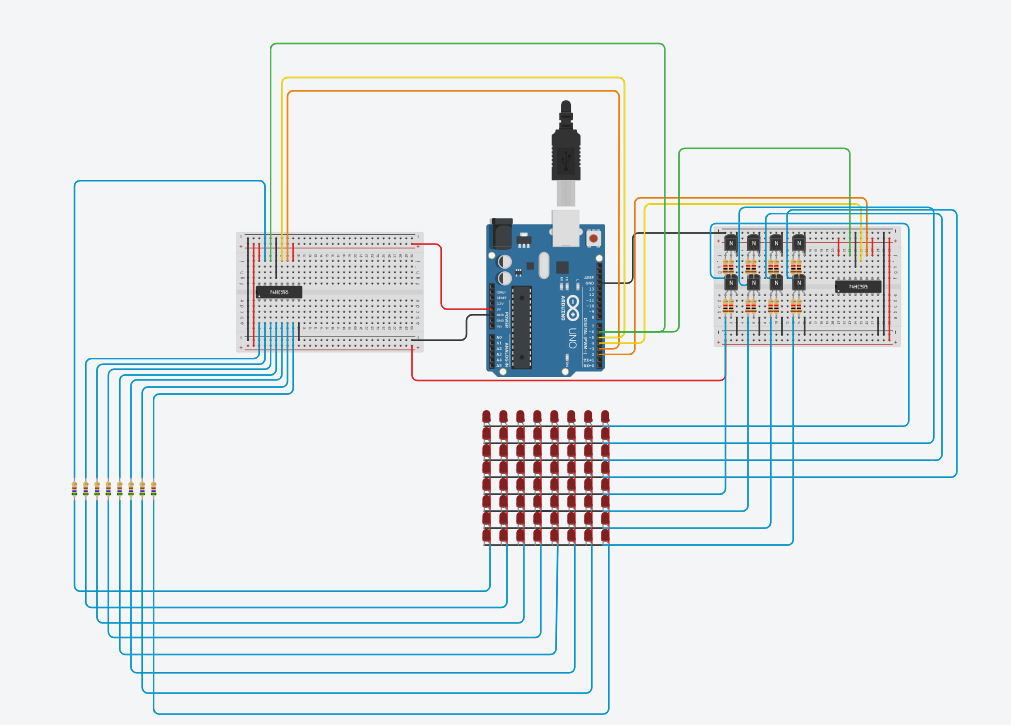


**Avances del lunes**

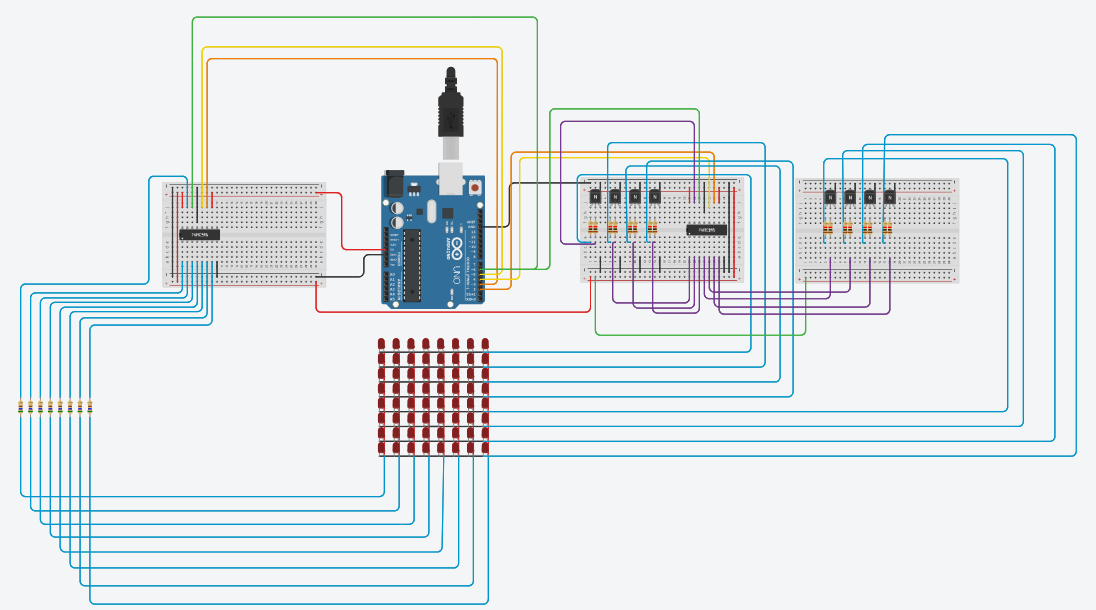
Conexiones iniciales del lunes con el uso de transistores



Continuación conexiones (Se guarda para mantener referencia)



Parte circuital finalizada:



Primera versión de código:

#define Reloj1F = 2

#define Reloj2 = 4

#define Reloj1C = 3

#define Reloj2C = 5

#define Serial = 6

void setup()

{

Serial.begin(9600);

pinMode(Reloj1F, OUTPUT);

pinMode(Reloj2F, OUTPUT);

pinMode(Reloj1C, OUTPUT);

pinMode(Reloj2C, OUTPUT);

}

void loop()

{

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Escoja una de las opciones del menu";

Serial.print("1. Secuencia de VERIFICACION del estado de los LED")

Serial.print("2. Ingrese y visualice su propia IMAGEN");

Serial.print("3. Ingrese sus patrones a visualizar")

delay(1000);

caso = Serial.parseint();

switch(caso){

case 1:

//Funcion VERIFICACION para probar que todos los led estan buenos

break;

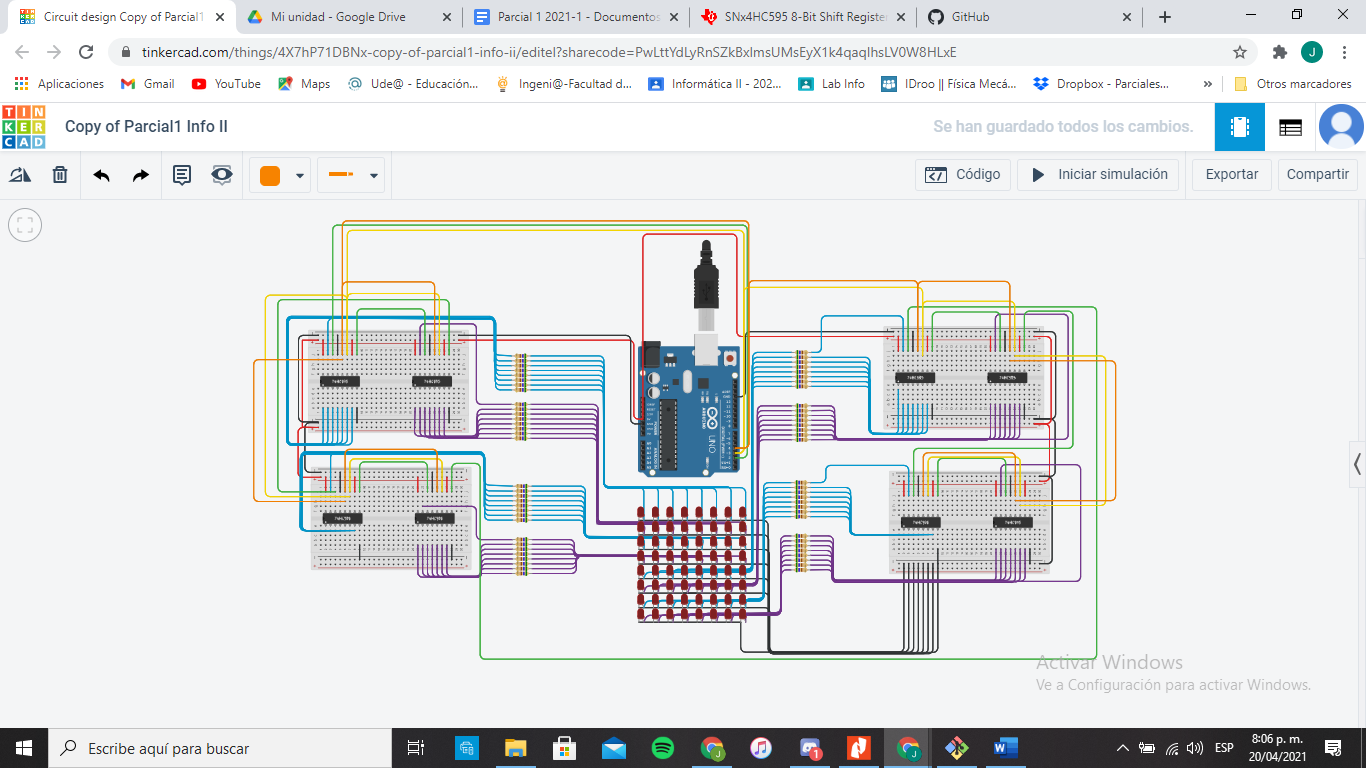
}

}

}

**Avances 20/04/2020**

Hoy el profesor nos informó que no estaba permitido utilizar transistores en la solución del circuito, por lo que nos adaptamos a una nueva estrategia donde se emplea un integrado 595 para el control independiente de cada columna, el cual representa un circuito más grande pero que requiere menor cantidad de pines del Arduino.



VERSION DE HOY PARA EL CODIGO

#define SERIAL 2

#define CLOCK1 3

#define CLOCK2 4

#define tiempo 1

void setup()

{

Serial.begin(9600);

pinMode(SERIAL, OUTPUT);

pinMode(CLOCK1, OUTPUT);

pinMode(CLOCK2, OUTPUT);

}

//EMPIEZO A CREAR MIS FUNCIONES

void verf(int p1, int p2, int p3, int p4, int p5, int p6, int p7, int p8){

//Le voy metiendo los datos unoa uno al integrado

//Activando el reloj 1 para dejar entrar cada dato que

//Entra por el serial

shiftOut(SERIAL, CLOCK1, MSBFIRST, p1);

shiftOut(SERIAL, CLOCK1, MSBFIRST, p2);

shiftOut(SERIAL, CLOCK1, MSBFIRST, p3);

shiftOut(SERIAL, CLOCK1, MSBFIRST, p4);

shiftOut(SERIAL, CLOCK1, MSBFIRST, p5);

shiftOut(SERIAL, CLOCK1, MSBFIRST, p6);

shiftOut(SERIAL, CLOCK1, MSBFIRST, p7);

shiftOut(SERIAL, CLOCK1, MSBFIRST, p8);

//Le activo el reloj 2 al integrado para que me saque todos los datos

//También esto me activa la salida de la salida invertida 8

digitalWrite(CLOCK2, HIGH);

digitalWrite(CLOCK2, LOW);

}

void verificacion(){

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

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

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

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

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

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

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

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

verf(0,0,0,0,0,0,2,0); delay(tiempo);

verf(0,0,0,0,0,2,0,0); delay(tiempo);

verf(0,0,0,0,2,0,0,0); delay(tiempo);

verf(0,0,0,2,0,0,0,0); delay(tiempo);

verf(0,0,2,0,0,0,0,0); delay(tiempo);

verf(0,2,0,0,0,0,0,0); delay(tiempo);

verf(2,0,0,0,0,0,0,0); delay(tiempo);

verf(0,0,0,0,0,0,0,4); delay(tiempo);

verf(0,0,0,0,0,0,4,0); delay(tiempo);

verf(0,0,0,0,0,4,0,0); delay(tiempo);

verf(0,0,0,0,4,0,0,0); delay(tiempo);

verf(0,0,0,4,0,0,0,0); delay(tiempo);

verf(0,0,4,0,0,0,0,0); delay(tiempo);

verf(0,4,0,0,0,0,0,0); delay(tiempo);

verf(4,0,0,0,0,0,0,0); delay(tiempo);

verf(0,0,0,0,0,0,0,8); delay(tiempo);

verf(0,0,0,0,0,0,8,0); delay(tiempo);

verf(0,0,0,0,0,8,0,0); delay(tiempo);

verf(0,0,0,0,8,0,0,0); delay(tiempo);

verf(0,0,0,8,0,0,0,0); delay(tiempo);

verf(0,0,8,0,0,0,0,0); delay(tiempo);

verf(0,8,0,0,0,0,0,0); delay(tiempo);

verf(8,0,0,0,0,0,0,0); delay(tiempo);

verf(0,0,0,0,0,0,0,16); delay(tiempo);

verf(0,0,0,0,0,0,16,0); delay(tiempo);

verf(0,0,0,0,0,16,0,0); delay(tiempo);

verf(0,0,0,0,16,0,0,0); delay(tiempo);

verf(0,0,0,16,0,0,0,0); delay(tiempo);

verf(0,0,16,0,0,0,0,0); delay(tiempo);

verf(0,16,0,0,0,0,0,0); delay(tiempo);

verf(16,0,0,0,0,0,0,0); delay(tiempo);

verf(0,0,0,0,0,0,0,32); delay(tiempo);

verf(0,0,0,0,0,0,32,0); delay(tiempo);

verf(0,0,0,0,0,32,0,0); delay(tiempo);

verf(0,0,0,0,32,0,0,0); delay(tiempo);

verf(0,0,0,32,0,0,0,0); delay(tiempo);

verf(0,0,32,0,0,0,0,0); delay(tiempo);

verf(0,32,0,0,0,0,0,0); delay(tiempo);

verf(32,0,0,0,0,0,0,0); delay(tiempo);

verf(0,0,0,0,0,0,0,64); delay(tiempo);

verf(0,0,0,0,0,0,64,0); delay(tiempo);

verf(0,0,0,0,0,64,0,0); delay(tiempo);

verf(0,0,0,0,64,0,0,0); delay(tiempo);

verf(0,0,0,64,0,0,0,0); delay(tiempo);

verf(0,0,0,64,0,0,0,0); delay(tiempo);

verf(0,0,64,0,0,0,0,0); delay(tiempo);

verf(0,64,0,0,0,0,0,0); delay(tiempo);

verf(64,0,0,0,0,0,0,0); delay(tiempo);

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

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

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

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

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

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

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

}

void loop()

{

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("\n");

Serial.print("Escoja una de las opciones del menu");

Serial.print("\n");

Serial.print("1. Secuencia de VERIFICACION del estado de los LED");

Serial.print("\n");

Serial.print("2. Ingrese y visualice su propia IMAGEN");

Serial.print("\n");

Serial.print("3. Ingrese sus patrones a visualizar");

Serial.print("\n");

int caso = Serial.parseInt();

delay(10000);

switch(caso){

case 1:

//Funcion VERIFICACION para probar que todos los led estan buenos

//for(int i = 0; i<5 ; i++) {

verificacion();

//delay(4000);

//}

break;

}

}

}

AVANCES DEL 21/04/2021

Se perfecciona la función VERIFICACION con una versión más corta y sin la función shirtOut()

#define SER 2 //ENTRADA SERIAL

#define RCLK 3 //RELOJ REGISTRO DESPLAZAMIENTO

#define SRCLK 4 //REGISTRO DE SALIDA

//funcion que recibe 4 variables tipo int, la primera el valor que se le dara a cada led y las siguientes los puertos analogos del arduino

void Verificacion(int num, int serial,int despla,int salida);

void setup()

{

//configuracion de puertos digitales de forma: OUTPUT

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

//Inicializar la entrada serial y los relojes en bajo

digitalWrite(SER, 0);

digitalWrite(RCLK, 0);

digitalWrite(SRCLK, 0);

for(int i = 1; i<=65; i++)

{

Verificacion(i,SER,RCLK,SRCLK);

}

}

void loop()

{

}

void Verificacion(int num, int serial ,int dezpla,int salida)

{

digitalWrite(serial, num);//se le da el valor para al serial el cual pasara el numero

digitalWrite(salida, 0);

digitalWrite(salida, 1);

digitalWrite(salida, 0);

digitalWrite(dezpla, 0);

digitalWrite(dezpla, 1);

digitalWrite(dezpla, 0);

}

PRIMERA IDEA CON AVANCE DE LA FUNCIÓN IMAGEN

Se va a implementar un doble puntero que permita recorrer un arreglo de ocho elementos, cada uno a su vez con ocho elementos. Inicialmente estos van a ser todos ceros y se le va a pedir al usuario decir cuales posiciones de LED deben activarse para formar su “imagen” deseada.

#define SER 2 //PUERTO SERIAL

#define RCLK 3 //RELOJ DE REGISTRO DESPLAZAMIENTO

#define SRCLK 4 //RELOJ DE REGISTRO DE SALIDA

int \*pmatriz, filas[8], columnas[8] = {0,0,0,0,0,0,0,0};

void Matrizcero(){

pmatriz = filas;

for(int i = 0 ; i<8 ; i++){

pmatriz[i] = columnas;

}

}

void Mostrar (){

for(int f = 0 ; f<8 ; f++){

for(int c = 0 ; c<8 ; c++){

Serial.print(\*((\*(pmatriz+f))+c));

Serial.print(" ");

}

Serial.print("\n");

}

}

void setup()

{

//Inclizamos el serial para poder recibir datos por consola.

Serial.begin(9600);

//configuracion de puertos digitales

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

//Se inicializa el serial y los relojes en LOW.

digitalWrite(SER, 0);

digitalWrite(RCLK, 0);

digitalWrite(SRCLK, 0);

}

void loop()

{

Matrizcero();

Mostrar();

}

VERSION 2

#define SER 2 //PUERTO SERIAL

#define RCLK 3 //RELOJ DE REGISTRO DESPLAZAMIENTO

#define SRCLK 4 //RELOJ DE REGISTRO DE SALIDA

int \*\*matrizCero;//Puntero doble que me recorrela matriz

matrizCero = new int\*[8]; // Hice que el puntero tome un arreglo de 8 punteros(que serán las filas)

for (int f = 0; f < 8; f++)

matrizCero[f] = new int[8]; // Los 8 elementos serán las filas

void Imagen (){

//Le pido al usuario que me diga cuales led prender y cuales no

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

Serial.print("\n");

Serial.print("Fila: ");

Serial.print(fila-'0');

Serial.print("\n");

Serial.print("Columna: ");

Serial.print(columna-'0');

Serial.print("Ingrese 1 si el led debe prenderse y 0 si no: ");

int dato = Serial.parseInt();

delay(3000);

matrizCero[fila][columa]=dato;

}

}

void EjecutarImagen(matrizCero);{

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

ingreso = \*(\*(matrizCero+fila)+columna) ;

digitalWrite(serial, ingreso);//Se le da el valor al serial, que es el que va pasando los numeros.

digitalWrite(salida, 0);

digitalWrite(salida, 1);

digitalWrite(salida, 0);

}

digitalWrite(dezpla, 0);

digitalWrite(dezpla, 1);

digitalWrite(dezpla, 0);

}

void setup()

{

//Inclizamos el serial para poder recibir datos por consola.

Serial.begin(9600);

//configuracion de puertos digitales

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

//Se inicializa el serial y los relojes en LOW.

digitalWrite(SER, 0);

digitalWrite(RCLK, 0);

digitalWrite(SRCLK, 0);

}

int main(){

Imagen();

}

void loop()

{

}

AVANCES 22/04/2021

Se adjunta la función imagen terminada, pero sin la optimización

#define SER 2 //PUERTO SERIAL

#define RCLK 3 //RELOJ DE REGISTRO DESPLAZAMIENTO

#define SRCLK 4 //RELOJ DE REGISTRO DE SALIDA

void setup(){

//Inclizamos el serial para poder recibir datos por consola.

Serial.begin(9600);

//configuracion de puertos digitales

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

//Se inicializa el serial y los relojes en LOW.

digitalWrite(SER, 0);

digitalWrite(RCLK, 0);

digitalWrite(SRCLK, 0);

}

int \*\*CrearM(){

int \*\*matrizCero;//Puntero doble que me recorrela matriz

matrizCero = new int\*[8]; // Hice que el puntero tome un arreglo de 8 punteros(que serán las filas)

for (int f = 0; f < 8; f++)

matrizCero[f] = new int[8]; // Los 8 elementos serán las filas

return matrizCero;

}

int \*\*Imagen (int \*\*matrizCero){

//Le pido al usuario que me diga cuales led prender y cuales no

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

Serial.print("\n");

Serial.print("Fila: ");

Serial.print(fila,DEC);

Serial.print("\n");

Serial.print("Columna: ");

Serial.print(columna,DEC);

Serial.print("\n");

Serial.print("Ingrese 1 si el led debe prenderse y 0 si no: ");

while(!Serial.available()>0);

int dato = Serial.parseInt();

matrizCero[fila][columna]=dato;

}

}

return matrizCero;

}

void EjecutarImagen(int \*\* matrizCero){

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

int ingreso = \*(\*(matrizCero+fila)+columna) ;

digitalWrite(SER, ingreso);//Se le da el valor al serial, que es el que va pasando los numeros.

digitalWrite(RCLK, 0);

digitalWrite(RCLK, 1);

digitalWrite(RCLK, 0);

}

digitalWrite(SRCLK, 0);

digitalWrite(SRCLK, 1);

digitalWrite(SRCLK, 0);

}

}

void loop()

{

int \*\*inicial = CrearM();

int \*\*final = Imagen(inicial);

EjecutarImagen(final);

}

**FUNCION IMAGEN OPTIMIZADA**

#define SER 2 //PUERTO SERIAL

#define RCLK 3 //RELOJ DE REGISTRO DESPLAZAMIENTO

#define SRCLK 4 //RELOJ DE REGISTRO DE SALIDA

void setup(){

//Inclizamos el serial para poder recibir datos por consola.

Serial.begin(9600);

//configuracion de puertos digitales

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

//Se inicializa el serial y los relojes en LOW.

digitalWrite(SER, 0);

digitalWrite(RCLK, 0);

digitalWrite(SRCLK, 0);

}

int \*\*CrearM(){

int \*\*matrizCero;//Puntero doble que me recorre la matriz

matrizCero = new int\*[8]; // Hice que el puntero tome un arreglo de 8 punteros(que serán las filas)

for (int f = 0; f < 8; f++)

matrizCero[f] = new int[8]; // Los 8 elementos serán las filas

return matrizCero;

}

int \*\*Imagen (int \*\*matrizCero){

//Le pido al usuario que me diga cuales led prender y cuales no

for (int fila = 0; fila<8; fila++){

Serial.print("\n");

Serial.print("Fila: ");

Serial.print(fila+1,DEC);

Serial.print("\n");

Serial.print("Ingrese 1 si el led debe prenderse y 0 si debe estar apagado (respectivamente por fila): ");

while(!Serial.available()>0);

long long int dato = Serial.parseInt();

for (int columna = 0; columna<7; columna++){

int individual = dato%10;

dato = dato/10;

matrizCero[fila][columna]=individual;

}

matrizCero[fila][7]=dato;

}

return matrizCero;

}

void EjecutarImagen(int \*\* matrizCero){

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

int ingreso = \*(\*(matrizCero+fila)+columna) ;

digitalWrite(SER, ingreso);//Se le da el valor al serial, que es el que va pasando los numeros.

digitalWrite(RCLK, 0);

digitalWrite(RCLK, 1);

digitalWrite(RCLK, 0);

}

digitalWrite(SRCLK, 0);

digitalWrite(SRCLK, 1);

digitalWrite(SRCLK, 0);

}

}

void loop()

{

int \*\*inicial = CrearM();

int \*\*final = Imagen(inicial);

EjecutarImagen(final);

}

AVANCES 23/04/2021

FUNCIÓN PUBLIK LISTA

#define SER 2 //PUERTO SERIAL

#define RCLK 3 //RELOJ DE REGISTRO DESPLAZAMIENTO

#define SRCLK 4 //RELOJ DE REGISTRO DE SALIDA

void setup(){

//Inclizamos el serial para poder recibir datos por consola.

Serial.begin(9600);

//configuracion de puertos digitales

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

//Se inicializa el serial y los relojes en LOW.

digitalWrite(SER, 0);

digitalWrite(RCLK, 0);

digitalWrite(SRCLK, 0);

}

int \*\*\*ArregloPublik( int patrones){

int \*\*\*ArregloP = new int \*\*[patrones];

for(int p = 0; p<patrones ; p++){

ArregloP[p] = new int\*[8];

for(int f = 0; f<8 ; f++){

ArregloP[p][f] = new int[8];

}

}

return ArregloP;

}

int \*\*\*MeterP (int \*\*\*ArregloP, int patrones){

//Le pido al usuario que me diga cuales led prender y cuales no

for(int patron = 0; patron<patrones ; patron++){

for (int fila = 0; fila<8; fila++){

Serial.print("\n");

Serial.print("Numero de patron:");

Serial.print(patron+1,DEC);

Serial.print("\n");

Serial.print("Fila: ");

Serial.print(8-fila,DEC);

Serial.print("\n");

Serial.print("Ingrese 1 si el led debe prenderse y 0 si debe estar apagado (ocho numeros por fila): ");

while(!Serial.available()>0);

long long int dato = Serial.parseInt();

for (int columna = 0; columna<7; columna++){

int individual = dato%10;

dato = dato/10;

ArregloP[patron][fila][columna]=individual;

}

ArregloP[patron][fila][7]=dato;

}

}

return ArregloP;

}

void EjecutarPub(int \*\*\* MeterP, int patrones, int secpatron){

for(int patron = 0; patron<patrones ; patron++){

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

int ingreso = \*(\*(\*(MeterP+patron)+fila)+columna) ;

digitalWrite(SER, ingreso);//Se le da el valor al serial, que es el que va pasando los numeros.

digitalWrite(RCLK, 0);

digitalWrite(RCLK, 1);

digitalWrite(RCLK, 0);

}

digitalWrite(SRCLK, 0);

digitalWrite(SRCLK, 1);

digitalWrite(SRCLK, 0);

}

int time = secpatron\*1000;

delay(time);

}

}

void Publik(){

Serial.print("Ingrese la cantidad de patrones que desea imprimir: ");

while(!Serial.available()>0);

int patrones = Serial.parseInt();

Serial.print("Ingrese el tiempo (en segundos) entre cada impresion de patrones: ");

while(!Serial.available()>0);

int secpatron = Serial.parseInt();

int \*\*\*InicialP = ArregloPublik(patrones);

int \*\*\*MedioP = MeterP (InicialP,patrones);

EjecutarPub(MedioP,patrones,secpatron);

}

void loop()

{

Publik();

}

CODIGO FINAL

#define SER 2 //PUERTO SERIAL

#define RCLK 3 //RELOJ DE REGISTRO DESPLAZAMIENTO

#define SRCLK 4 //RELOJ DE REGISTRO DE SALIDA

//int menuIn = 0;

void setup()

{

//Inclizamos el serial para poder recibir datos por consola.

Serial.begin(9600);

//configuracion de puertos digitales

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

//Se inicializa el serial y los relojes en LOW.

digitalWrite(SER, 0);

digitalWrite(RCLK, 0);

digitalWrite(SRCLK, 0);

}

//COMIENZO A AGREGAR LAS FUNCIONES PROPIAS

//Funcion que crea el doble puntero para la funcion imagen

int \*\*CrearM(){

int \*\*matrizCero;//Puntero doble que me recorre la matriz

matrizCero = new int\*[8]; // Hice que el puntero tome un arreglo de 8 punteros(que serán las filas)

for (int f = 0; f < 8; f++)

matrizCero[f] = new int[8]; // Los 8 elementos serán las filas

return matrizCero;

}

//Esta el la funcion que recibe los datos de un patron y arma el doble puntero para la funcion Imagen

int \*\*Imagen (int \*\*matrizCero){

//Le pido al usuario que me diga cuales led prender y cuales no

for (int fila = 0; fila<8; fila++){

Serial.print("\n");

Serial.print("Fila: ");

Serial.print(8-fila,DEC);

Serial.print("\n");

Serial.print("Ingrese 1 si el led debe prenderse y 0 si debe estar apagado (respectivamente por fila): ");

while(!Serial.available()>0);

long long int dato = Serial.parseInt();

for (int columna = 0; columna<7; columna++){

int individual = dato%10;

dato = dato/10;

matrizCero[fila][columna]=individual;

}

matrizCero[fila][7]=dato;

}

return matrizCero;

}

//Esta es la funcion que manda los datos del doble puntero hacia los integrados

void EjecutarImagen(int \*\* matrizCero){

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

int ingreso = \*(\*(matrizCero+fila)+columna) ;

digitalWrite(SER, ingreso);//Se le da el valor al serial, que es el que va pasando los numeros.

digitalWrite(RCLK, 0);

digitalWrite(RCLK, 1);

digitalWrite(RCLK, 0);

}

digitalWrite(SRCLK, 0);

digitalWrite(SRCLK, 1);

digitalWrite(SRCLK, 0);

}

}

//Funcion de verificacion

void Verificacion(int num, int serial ,int dezpla,int salida)//Num lo que hace es recorrer la matriz de LEDS, la segunda variable es para mandar los datos al serial y las otras 2 ajustan los relojes.

{

digitalWrite(serial, num);//Se le da el valor al serial, que es el que va pasando los numeros.

digitalWrite(salida, 0);

digitalWrite(salida, 1);

digitalWrite(salida, 0);

digitalWrite(dezpla, 0);

digitalWrite(dezpla, 1);

digitalWrite(dezpla, 0);

}

//Crea el triple puntero para la funcion Publik

int \*\*\*ArregloPublik( int patrones){

int \*\*\*ArregloP = new int \*\*[patrones];

for(int p = 0; p<patrones ; p++){

ArregloP[p] = new int\*[8];

for(int f = 0; f<8 ; f++){

ArregloP[p][f] = new int[8];

}

}

return ArregloP;

}

//Le pide al usuario los datos para armar el triple puntero de la funcion Publik

int \*\*\*MeterP (int \*\*\*ArregloP, int patrones){

//Le pido al usuario que me diga cuales led prender y cuales no

for(int patron = 0; patron<patrones ; patron++){

for (int fila = 0; fila<8; fila++){

Serial.print("\n");

Serial.print("Numero de patron:");

Serial.print(patron+1,DEC);

Serial.print("\n");

Serial.print("Fila: ");

Serial.print(8-fila,DEC);

Serial.print("\n");

Serial.print("Ingrese 1 si el led debe prenderse y 0 si debe estar apagado (ocho numeros por fila): ");

while(!Serial.available()>0);

long long int dato = Serial.parseInt();

for (int columna = 0; columna<7; columna++){

int individual = dato%10;

dato = dato/10;

ArregloP[patron][fila][columna]=individual;

}

ArregloP[patron][fila][7]=dato;

}

}

return ArregloP;

}

//Envia los datos del triple puntero a los integrados

void EjecutarPub(int \*\*\* MeterP, int patrones, int secpatron){

for(int patron = 0; patron<patrones ; patron++){

for (int fila = 0; fila<8; fila++){

for (int columna = 0; columna<8; columna++){

int ingreso = \*(\*(\*(MeterP+patron)+fila)+columna) ;

digitalWrite(SER, ingreso);//Se le da el valor al serial, que es el que va pasando los numeros.

digitalWrite(RCLK, 0);

digitalWrite(RCLK, 1);

digitalWrite(RCLK, 0);

}

digitalWrite(SRCLK, 0);

digitalWrite(SRCLK, 1);

digitalWrite(SRCLK, 0);

}

int time = secpatron\*1000;

delay(time);

}

}

//Funcion principal Publik

void Publik(){

Serial.print("\n");

Serial.print("\n");

Serial.print("Ingrese la cantidad de patrones que desea imprimir: ");

while(!Serial.available()>0);

int patrones = Serial.parseInt();

Serial.print("\n");

Serial.print("Ingrese el tiempo (en segundos) entre cada impresion de patrones: ");

while(!Serial.available()>0);

int secpatron = Serial.parseInt();

int \*\*\*InicialP = ArregloPublik(patrones);

int \*\*\*MedioP = MeterP (InicialP,patrones);

EjecutarPub(MedioP,patrones,secpatron);

}

void loop()

{

//Para que no se note el menu anterior en la consola

for(int n=0;n<10;n++) Serial.print("\n");

//Se imprime el menú para escoger ls fuciones

Serial.print(" BIENVENIDO");

Serial.print("\n");

Serial.print("Escoja una de las opciones del menu");

Serial.print("\n");

Serial.print("0. Detener lo que se esta ejecutando");

Serial.print("\n");

Serial.print("1. Secuencia de VERIFICACION del estado de los LED");

Serial.print("\n");

Serial.print("2. Ingrese y visualice su propia IMAGEN");

Serial.print("\n");

Serial.print("3. Ingrese varios patrones a visualizar");

Serial.print("\n");

delay(3000);

if (Serial.available()>0){

Serial.flush();

int caso = Serial.parseInt();

switch(caso){

case 0 :{

Serial.print("\n\n\n\n");

Serial.print("DETENIENDO LO QUE SE ESTA EJECUTANDO");

Serial.print("\n");

for(int i = 1; i<=65; i++) {

Verificacion(0,SER,RCLK,SRCLK);}

Serial.print("\n");

Serial.print("Para ejecutar otra opcion ingrese cualquier letra");

Serial.print("\n");

while(!Serial.available()>0);

break;}

case 1 :

{

//Para ejecutar la funcion de verificacion

Serial.print("\n\n\n\n");

Serial.print("Ejecutando funcion de Verificacion");

for(int i = 1; i<=65; i++) {

Verificacion(i,SER,RCLK,SRCLK);}//Este for es el que permite recorrer los 64 LEDS.

// delay(8000);

Serial.print("\n");

Serial.print("Para ejecutar otra opcion ingrese cualquier letra");

Serial.print("\n");

while(!Serial.available()>0);

break;}

case 2 :

{

//Para ejecutar la funcion de imagen

int \*\*inicial = CrearM();

int \*\*final = Imagen(inicial);

EjecutarImagen(final);

Serial.print("\n");

Serial.print("Para ejecutar otra opcion ingrese cualquier letra");

Serial.print("\n");

while(!Serial.available()>0);

break;

}

case 3 :{

Serial.print("\n");

Serial.print("\n");

Publik();

Serial.print("\n");

Serial.print("Para ejecutar otra opcion ingrese cualquier letra");

Serial.print("\n");

while(!Serial.available()>0);

break;}

}//Del switch

}//Del if

}//Del loop