Parcial practico 1

Proceso de análisis, diseño e implementación

Rolman David Echavarria Prince

Despartamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Septiembre de 2023

Índice

1. Objetivos **2**
1.1. Circuito 2
1.2. Algoritmo 2

2. Introducción **3**

3. Marco teórico **4**
3.1. Funcionamiento circuito integrado 74HC595 4
3.1.1. Descripcion 4
3.1.2. Funcionamiento 6

4. Análisis del problema **7**
4.1. Materiales para la implementación del sistema 7
4.1.1. Arduino uno 7
4.1.2. 74HC595 7
4.1.3. Resistores 8
4.1.4. Diodo LED color verde 8
4.2. Desarrollo de algoritmos 8
4.3. Algoritmos implementados 9
4.4. Problemas de desarrollo 13
4.5. Evolución de la solución y consideraciones a tener en cuenta en la implementación. 14

5. Diseño de la solución **15**

6. Marco experimental **15**

7. Conclusiones **16**

1. Objetivos

Objetivo del Informe: Implementación de un sistema de matriz 8x8 de diodos LED, que permita al usuario ejecutar su funcionamiento. El circuito se diseñará para funcionar con Arduino y se complementará con el integrado recomendado 74HC595 para multiplicar las salidas. Además, se desarrollará un código basado en C++ para controlar la matriz de LEDs.

1.1. Circuito

Establecer un sistema de software que integre puertos, circuitos integrados, resistores y diodos LED como solución al problema en el cual el usuario realiza pruebas a través del menú del algoritmo. El objetivo es minimizar la caída de voltaje y corriente que afecta la luminosidad de la matriz, garantizando una legibilidad óptima de los patrones sin exceder los límites que puedan dañar los componentes.

1.2. Algoritmo

Establecer un mensaje, símbolo o caracteres mediante una matriz 8x8, con requisitos que incluyen varias funciones disponibles a través de un menú de opciones, como verificación, imágenes y patrones. Diseñar una solución en C++ que utilice estructuras de programación, variables, arreglos, punteros, asignación dinámica de memoria y funciones, y que incluya una revisión y depuración exhaustivas del algoritmo.

2. Introducción

Se lleva a cabo un proyecto que consiste en un circuito de una matriz de 64 LEDs. El usuario podrá verificar el funcionamiento de todos los LEDs con la función de 'verificación'. A través de la función 'imagen', podrá introducir por comunicación serial el mensaje o carácter que desea ver reflejado en la matriz. La función 'secuencias' mostrará patrones de forma alterna. Todo esto se logra mediante una función pública que actúa como un menú para las funciones mencionadas anteriormente.

3. Marco teórico

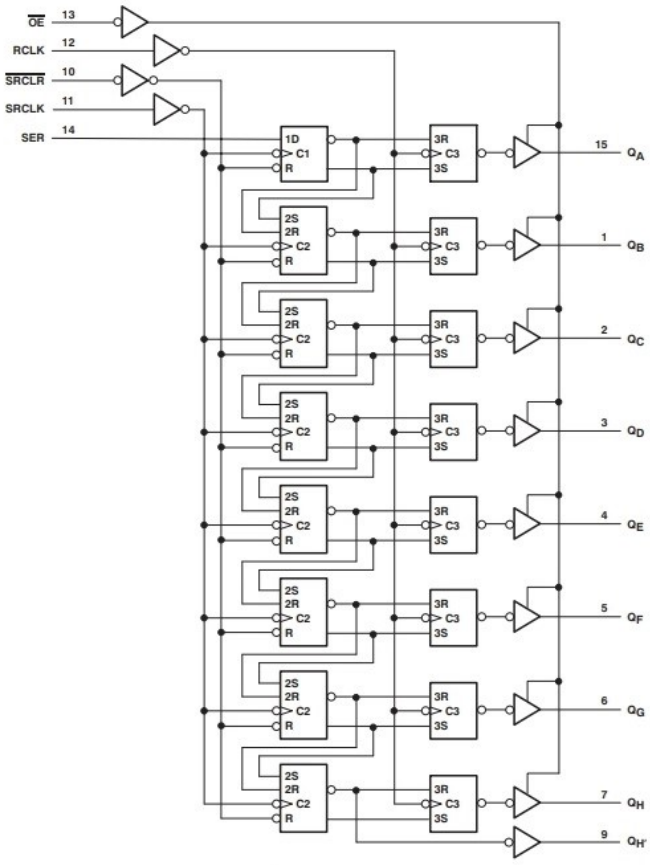
3.1. Funcionamiento circuito integrado 74HC595

3.1.1. Descripcion

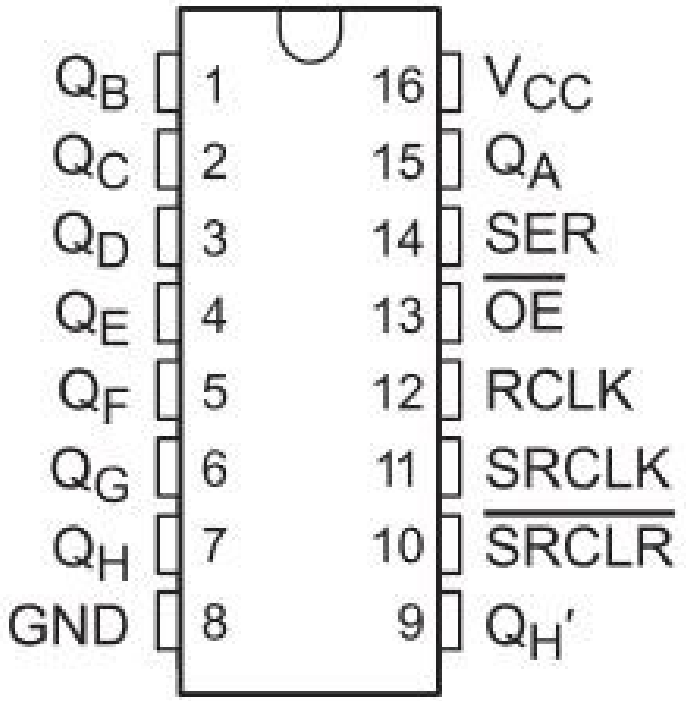
El 74HC595 es conocido cómo registro de desplazamiento de 8 bits y sirve para expandir los puertos de un microcontrolador y también para almacenar estados lógicos, los datos ingresan de forma serial y a la salida obtenemos los datos en paralelo.

Este circuito integrado consta principalmente de dos arreglos de flip-flops, el de la izquierda son 8 flip-flops encadenados en serie que permite el desplazamiento de los bits y el segundo arreglo copia los bits del primero para posteriormente mostrarlos en las salidas.

En el siguiente diagrama se pueden apreciar los dos arreglos:



[1]



[2]

Vamos a definir un pulso cómo el cambio de estado alto (5v) al estado bajo (0v).

Para su funcionamiento disponemos los siguientes pines:

- Pin 8 (SER): Este pin recibe los datos de entrada en forma serial 1 o 0.
- Pin 11 (RCLK): Reloj de registro de salida, en cada pulso se copia los 8 bits correspondientes de forma simultánea a un segundo conjunto de flip-flops que están conectados a las 8 salidas del circuito integrado.
- Pin 12 (SRCLK): Reloj de registro de desplazamiento. En cada pulso se copiará el bit que se tenga en el pin SER e irá empujando el dato al siguiente registro en serie.

3.1.2. Funcionamiento

Para escribir un 0 se establece en bajo (0 voltios) el pin 8(SER) y a continuación se manda el pulso de reloj al pin 12(SRCLK), ahora el 0 está almacenado en el primer flip-flop. Si se quiere escribir un 1 entonces el pin 8(SER) se establece en alto (5 voltios) y nuevamente se manda el pulso de reloj al pin 12(SRCLK), lo que sucede es que el cero que había al inicio se desplaza al siguiente flip-flop y ahora el 1 ocupa el primero. Siguiendo esta lógica se pueden almacenar hasta 8 bits. Lo más importante es que el nuevo dato empuja a los demás una posición.

Una vez se tengan los 8 bits en su posición se procede a enviar un pulso al pin 11(RCLK) copiando estos 8 bits al segundo arreglo de flip-flops de forma simultánea y así poder visualizarlos en las salidas del circuito integrado
.

[3]

4. Análisis del problema

Se necesita por medio del puerto serial del Arduino navegar en el menú de opciones de las diferentes funciones para el encendido y apagado de ciertos diodos LEDs.

4.1. Materiales para la implementación del sistema

4.1.1. Arduino uno

Se encargará de generar la información a través del puerto SER y en el que escribirá el programa.

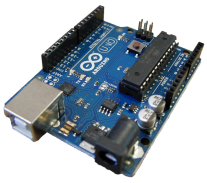


Figura 1: Arduino uno [4]

4.1.2. 74HC595

Recibe los datos generados por el Arduino uno a través de un puerto y sus dos relojes, estos ocho puertos adicionales serán la fila o columna de la matriz de LEDs.

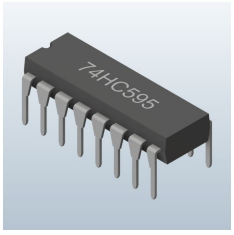


Figura 2: 74HC595 [5]

4.1.3. Resistores

resistores con la resistencia de 450 omh, para la corriente toral del integrado no supere el limite.

4.1.4. Diodo LED color verde

Este diodo puede encender a un color estándar con un limite de 1,8v y 0.015A y con un color brillante con un limite de 3v y 0.02A.

Figura 3: diodo LED [6]

4.2. Desarrollo de algoritmos

Con el circuito ya establecido y funcionalmente básico. Para completar los diferentes objetivos exigidos, se requerían diferentes funciones.

- función verificación: donde el usuario ingresa el tiempo de encendido de los 64 LEDs en esta parte cada dato por el puerto SER sera de 1, y con los respectivos pulsos de ambos relojes.
- función imagen: se reserva filas de un arreglo bidimensional en el HEAP, se ingresan datos decimales que por corrimiento de bits y comparación AND con el bit menos significativo se guarda en la ultima posición; de los 8 datos, los enteros serán 8 representaciones en byte.
- función patrones: En esta función se reusan diferentes funciones y el arreglo para trabajar con los 4 patrones y cada patrón se trabaja como imagen en el cual esta preestablecida ya sus valores decimales para la representación binaria.
- función publik: Esta función permite el reuso de código con excepción de imágenes y patrones aleatorios indeterminados con tiempo especifico.

las diferentes actividades o esquema fue:

- diseño del algoritmo: el objetivo es trabajar con funciones evitando así mucho código en el loop, un algoritmo que permita el reuso de codigo.

- implementación: estableciendo c++ en el ámbito de Arduino, con los diferentes objetivos para encender o apagar de maneras distintas la matriz de 64 LEDs con ayuda de funcion verificacion(boolean) o imagen(boolean **). Se realizaron diferentes pruebas a medida que se avanzaba en cada objetivo, buscando posibles errores y como simplificar.
- se documento parte esencial del código fuente, siendo más técnica y como manual para el usuario.
- Algunos aspectos a mejorar son la velocidad de la simulación ya que el Arduino físico trabaja a velocidades reales, una navegación con el menú más entendible, simplificar código ayuda al procesamiento de la maquina.

4.3. Algoritmos implementados

```
//definicion y asignaacion de valores puertos
const int SER = 8;
const int RCLK = 11;
const int SRCLK = 12;

char opcion;//v:verificacion,i:imagen,y:usuario,
int tiempo;
boolean **matriz=new boolean *[8];
void reloj(int c);
void verificacion(boolean);
void publik();
void imagen(boolean **);
void patrones(boolean **,int);
void reservaFilas(boolean **);
void datosMatriz(boolean **);
void intaBinary(int, boolean *);
void liberarMatriz(boolean **);
void datosMatrizP(boolean **,int []);


void setup()
{
  //paso 1: configuracion de puertos digitales como salida
    pinMode(SER, OUTPUT );
    pinMode(RCLK, OUTPUT );
    pinMode(SRCLK, OUTPUT );
  //inicializar en bajo
    digitalWrite(SER,0);
    digitalWrite(RCLK,0);
    digitalWrite(SRCLK,0);
  //inicio de puerto serial
    Serial.begin(9600);
  //mostrar menu una sola vez
    Serial.print("*****menu*****\n");
  Serial.print("v:verificacion\n");
    Serial.print("i:imagen\n");
    Serial.print("p:patrones\n");
    Serial.print("n:publik\n");
    Serial.print("**************\n");
```

```
    Serial.print("ingrese opcion: ");
  //incializacion de numeros semi-aleatorios
    randomSeed(analogRead(0));
}

void loop()
{
  if(Serial.available()>0){
    //lee dato caracter en monitor
    opcion=Serial.read();
    switch (opcion){
      case 'v':
        Serial.print("tiempo encendido-apagado: ");
      while (!Serial.available()) {}
      tiempo=Serial.parseInt();
      Serial.println(tiempo);
      //se establece el tiempo
        verificacion(1);
        delay(tiempo);
        verificacion(0);
        delay(tiempo);
      break;
      case 'i':
        reservaFilas(matriz);//reservacion de arreglo dinamico
      datosMatriz(matriz);//
        imagen(matriz);
        delay(1000);
      break;
      case 'p':
        reservaFilas(matriz);
      for(int p=0;p<4;p++){
        patrones(matriz,p);
        delay(1);
        imagen(matriz);
        delay(1500);
      }
      break;
      case 'n':
        Serial.println(opcion);
        publik();
      break;
      default:
      Serial.println("opcion invalida");
    }
    Serial.print("*****menu*****\n");
  Serial.print("v:verificacion\n");
    Serial.print("i:imagen\n");
    Serial.print("p:patrones\n");
    Serial.print("n:publik\n");
    Serial.print("**************\n");
    Serial.print("ingrese opcion: ");
  }


}

void publik(){
```

```
97
98  int secuencias;
99  char opcionP;
100
101 Serial.print("*****menu_publik*****\n");
102 Serial.print("v:verificacion\n");
103 Serial.print("i:imagen\n");
104 Serial.print("p:patrones\n");
105 Serial.print("**************\n");
106 Serial.print("ingrese opcion_publik: ");
107 while (!Serial.available()) {}
108 opcionP=Serial.read();
109 Serial.println(opcionP);
110
111 switch (opcionP){
112   case 'v':
113     Serial.print("tiempo encendido-apagado: ");
114     while (!Serial.available()) {}
115     tiempo=Serial.parseInt();
116     Serial.println(tiempo);
117     Serial.print("numero secuencias: ");
118     while (!Serial.available()) {}
119     secuencias=Serial.parseInt();
120     Serial.println(secuencias);
121     for(int i=0;i<secuencias;i++){
122         verificacion(1);
123         delay(tiempo);
124         verificacion(0);
125         delay(tiempo);
126     }
127
128   break;
129   case 'i':
130     reservaFilas(matriz);
131     datosMatriz(matriz);
132     Serial.print("tiempo encendido-apagado: ");
133       while (!Serial.available()) {}
134     tiempo=Serial.parseInt();
135     Serial.println(tiempo);
136       while(true){   //bucle indeterminado
137             imagen(matriz);
138               delay(tiempo);
139               verificacion(0);
140               delay(tiempo);
141       }
142   break;
143   case 'p':
144     reservaFilas(matriz);
145     Serial.print("tiempo encendido-apagado: ");
146       while (!Serial.available()) {}
147     tiempo=Serial.parseInt();
148     Serial.println(tiempo);
149     while(true){//bucle indeterminado
150         patrones(matriz,random(5));
151         delay(1);
152         imagen(matriz);
153         delay(tiempo);
```

```
      }
    break;
    default:
      Serial.print("opcion invalida_p\n");
  }

}

void verificacion(boolean valor){
    for(int j=0;j<64;j++){
        digitalWrite(SER,valor);//envia al SER el dato valor
    reloj(SRCLK);//pulso de reloj
      reloj(RCLK);//pulso de reloj
    }
}

void imagen(boolean **m){
    //llevar matriz al serial(SER)
    for(int y=0;y<8;y++){
      for(int x=0;x<8;x++){
          digitalWrite(SER, m[y][x]);
      reloj(SRCLK);
        reloj(RCLK);
      }
  }
}
void reservaFilas(boolean **m){
    for (int i=0;i<8;i++ ) {
        m[i]=new boolean [8];//reserva de elementos dinamicos
    }
}
void datosMatriz(boolean **m){
  //usuario ingresa fila por fila dato en decimal ->byte
    int valor=0;
    for(int i=0;i<8;i++){
      Serial.print("Ingrese dato de fila " + String(i) + " en
    decimal: ");
        while (!Serial.available()) {}
      valor=Serial.parseInt();
      Serial.println(valor);
        intaBinary(valor,m[i]);//convierte un entero a binario
    }
}
void datosMatrizP(boolean **m,int a[]){
  for(int i=0;i<8;i++){
    intaBinary(a[i],m[i]);//convierte un entero a binario
  }
}

void patrones(boolean **m,int rand){
  int array[4][8]={
  {24,60,126,255,255,126,60,24},
  {129,66,36,24,24,36,66,129},
  {219,219,109,109,219,219,109,109},
  {240,120,60,30,30,60,120,240},
  };//son los datos de los diferentes patrones
  switch (rand){
```

```
    case 0:
      datosMatrizP(m,array[0]); //patron 1
    break;
    case 1:
      datosMatrizP(m,array[1]); //patron 2
    break;
    case 2:
      datosMatrizP(m,array[2]); //patron 3
    break;
    case 3:
      datosMatrizP(m,array[3]); //patron 4
    break;
    default:
    Serial.print("\n");
  }
}
void liberarMatriz(boolean **m){
    //liberacion de memoria
    for(int i=0;i<8;i++){
        delete [] m[i];
    }
    delete []m;
    m = nullptr;
}

void intaBinary(int valor, boolean *ptr){
  for (int i = 0; i < 8; i++) {
    // extrae el ultimo bit del valor utilizando
    //desplazamiento a la derecha y operaci n and.
    int bit = (valor >> i) & 1;
    // almacena el bit en el arreglo de booleanos,
    //invirtiendo el orden para que el bit  m  s  significativo
    // del entero se almacene en la  posici n 7 del arreglo.
    ptr[7 - i] = boolean(bit);
  }
}

void reloj (int c){
  //pulso reloj
  digitalWrite(c,0);
  digitalWrite(c,1);
  digitalWrite(c,0);
}
```

Listing 1: Algoritmo de circuito

4.4. Problemas de desarrollo

En el hardware, enfrenté problemas con el cálculo de resistencias para obtener una corriente y voltaje adecuados para una iluminación apreciable. También encontré conexiones faltantes.

En el software, fue importante discernir cuándo leer enteros y cuándo leer caracteres. Además, tuve que tener en cuenta las funciones y realizar ajustes para simplificar el código. No es posible diseñar sin comprender la teoría detrás

del integrado 74HC595. También fue crucial trabajar de manera efectiva con la matriz dinámica y transmitir esos datos al SER.

4.5. Evolución de la solución y consideraciones a tener en cuenta en la implementación

Para avanzar en la solución, primero fue necesario comprender cómo trabajar con los integrados, en este caso, se optó por organizarlos por filas. Luego, se resumieron los objetivos y se diseñaron estructuras de control necesarias para cumplir con los requisitos. Las funciones escritas pueden reutilizarse en tareas futuras.

Es importante abordar la solución paso a paso, desde lo más básico hasta el objetivo final. Los códigos binarios son fundamentales en este proceso, al igual que el desplazamiento de bits.

5. Diseño de la solución

Con los materiales ya mencionados se establece tres puestos de salida SER puerto 8, SRCLK puerto 12, RCLK puerto 11, la potencia (5v) y la tierra, el ser se conecta a la entrada del primer 74HC595 de este IC salen 8 puertos que serán la fila octava o inferior de la matriz, además la salida QH' o salida invertida se convierte en la entrada del siguiente integrado que en continuación sera la séptima fila de la matriz. Todas las 64 salidas estarán conectadas a un resistor independiente.

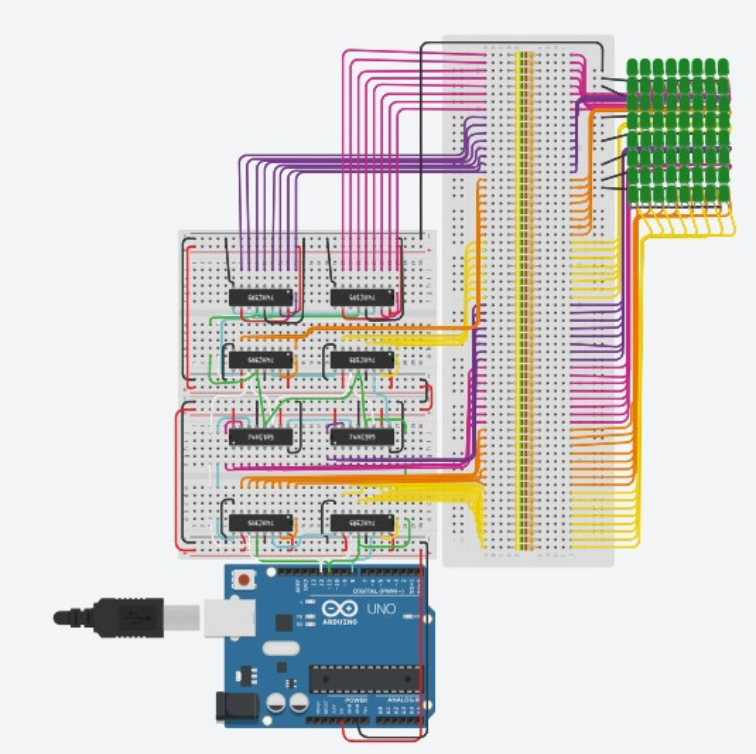


Figura 4: diseño Tinkercad

6. Marco experimental

versión funcional con algoritmo básico de comprobación.

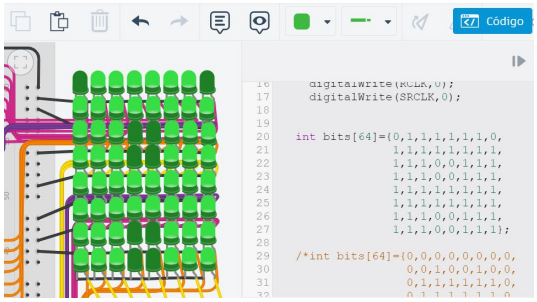


Figura 5: prueba1

7. Conclusiones

Cada componente brinda una solución y en el caso del integrado es brindarnos un ahorro de puertos de los cuales tenemos un limite, estos IC tienen muchas aplicaciones en diferentes sectores. Con prueba y error se comprende funcionalidad y en el caso de la matriz de 8x8 reflejar los mensajes.

En el transcurso del proyecto se realizaron diferentes experimentos para comprender el uso de los integrados, Tinkercad como plataforma en que se trabajó y complementando con diferentes fuentes de información, se logra apreciar como un código en ejecución trabaja en lo físico dando la satisfacción de ver como funciona, aunque se presenten dificultades ya sea en el uso de herramientas o en como solucionar de forma algorítmica o electrónicamente un problema.

Referencias

[1] T. Instruments. Snx4hc595 8-bit shift registers with 3-state output registers. [Online]. Available: https://www.ti.com/lit/ds/symlink/sn74hc595.pdf

[2] L. del Valle Hernández. 74hc595 registro de desplazamiento con arduino. [Online]. Available: https://programarfacil.com/blog/74hc595-registro-de-desplazamiento-arduino/

[3] Dan. How to use a 74hc595 shift register. [Online]. Available: https://www.marginallyclever.com/2017/02/use-74hc595-shift-register/

[4] JotaCartas. (2011) Arduino-uno-perspective-whitw. [Online]. Available: https://es.wikipedia.org/wiki/Archivo:Arduino-uno-perspective-whitw.jpg

[5] megaeshop. (2023) Sn74hc595n. [Online]. Available: https://megaeshop.pk/product/sn74hc595n-74hc595n-74hc595-dip-16-shift-register-ic.html

[6] Steren. Led de 5 mm, color verde claro. [Online]. Available: https://www.steren.com.co/led-de-5-mm-color-verde-claro.html