

INFORME DE ANÁLISIS Y DISEÑO

74HC595

Julian Montenegro Pinzón

Jaime Moncada Londoño

Juan Esteban Grisales Gallego

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Febrero de 2022

Índice

1. Sección introductoria	2
2. Objetivos	2
2.1. objetivo General	2
2.2. objetivos Específicos	2
3. Marco Teórico	2
4. Análisis de desafío	3
5. Aplicación	4
6. diseño de circuito	4
6.1. Candidato 1 (Juan Esteban Grisales)	4
6.2. Candidato 2 (Julian Montenegro)	6
6.3. Avance diseño	7
6.4. Avance sistema de decodificación	7
6.5. Tener en cuenta para el diseño	9
6.6. Observación	10
7. Conclusiones	10
8. Sección de contenido	11
8.1. Citación	11
9. Inclusión de imágenes	11

1. Sección introductoria

Porque es importante el uso del circuito integrado para resolver nuestro desafío, el funcionamiento del circuito integrado 74HC595 nos permite hacer una ampliación de los puertos del arduino y llevar a que se tengan más salidas con solo 3 entradas, el cual al realiza la ejecución de funciones que conllevan a la optimización de espacios y usos. Utilizaremos palabras propias para la explicación de su implementación y lograr así los objetivos planteados en el informe. Sin embargo la aparición de problemas en el análisis del trabajo que lleva a la preocupación del equipo por tal vez no realizar un buen trabajo, visto de otro modo con el entendimiento del microprocesador y los temas vistos en clases se deben llevar todos los esfuerzos a la realización del desafío.

sabemos que el éxito del trabajo será gracias a nuestros conocimientos y a la eficiencia de nuestro ingenio. en la práctica se verá el esfuerzo y la organización desempeñada por los roles que cumplirá cada uno de los integrantes.

2. Objetivos

2.1. objetivo General

Investigar y entender cómo es el funcionamiento del circuito integrado 74HC595 y revisar las distintas aplicaciones del mismo, por consiguiente entender cómo podríamos utilizarlo en nuestro desafío.

2.2. objetivos Específicos

1. Revisar la hoja de datos del microprocesador, así entender cómo se toman los datos y cómo los podemos pasar a la salida.
2. Analizar el desarrollo para el desafío planteado en el parcial 1 del curso de informática II. utilizar palabras propias para la definición de los resultados de la investigación del uso y función del micro controlado.
3. Identificar posibles problemas y soluciones en el desarrollo del desafío.
4. Análisis de todas las pines del microcontrolador y como hacer uso de ellos.

3. Marco Teórico

Uno de los registros de desplazamiento más comúnmente usados es el 595, usando solo tres entradas digitales de nuestro arduino podremos aumentar la cantidad de salidas en 8 y procesar datos a conveniencia, el circuito integrado 74HC595 basa su funcionamiento en el pulso del reloj, cada vez que aplicamos un cambio en este pulso y este un alto (1), los registros que están en el puerto serial pasar a paralelo y se desplazan hacia la izquierda, por ejemplo: en el momento en que nuestro registro de desplazamiento detecta un pulso alto (1)

en su registro de reloj los datos en sus pines de salida se desplazan un pin hacia la izquierda, el dato que antes estaba en el pin 0 ahora se encuentra en el pin 1, el dato que antes estaba en el pin 1 ahora está en el pin 2 y así sucesivamente.

Los datos ingresan por el puerto SER, pero para poder pasar el dato hay que hacer una señal de reloj con los puertos SRCLK el cual al hacer el flanco (0,1,0) pasa hacer un registro de almacenado del dato, pero si hacemos un nuevo flanco (0,1,0) con el puerto RCLK será pasado al registro de salida y así sucesivamente podemos hacer el corrimiento de los datos. Y que estos sean reflejados en las salidas.

Ahora bien, si el flanco lo hacemos solo con el puerto SRCLK el corrimiento se hace, pero los datos no se van a ver reflejados en los puertos de salida, por eso la importancia de hacer los flancos también con el puerto RCLK.

4. Análisis de desafío

La manera en la que se piensa abordar el desafío es tomando provecho de las 8 salidas digitales que nos ofrece el circuito integrado 74HC595, de manera serial estaremos recibiendo números enteros, estos números enteros deberán pasar por un proceso de conversión a binario de 8 bits, en el primer proceso los primeros 8 bits (correspondientes al primer número entero) estarán ingresando uno a uno a nuestro registro de desplazamiento, en el momento en que el programa detecte que nuestro pulso de reloj estuvo en alto 8 veces daremos por entendido que el número estará disponible en las 8 salidas digitales, este número ahora será convertido a entero y almacenado en una variable (num1), nuestro sistema iniciara de nuevo un proceso de 8 pulsos de reloj para poder procesar nuestro siguiente número y así almacenarlo en una segunda variable (num2), una vez tenemos datos almacenados en estas dos variables el sistema procederá a comparar dichos datos y hacer gestión de ellos según como se necesite en el proceso de descriptación.

El sistema continuará procesado los números que ingresen por el puerto serial y en cada ciclo el valor de nuestra segunda variable (num2) pasará a ser el valor de la primera (num1) para que la segunda variable (num2) quede disponible para el número que va ingresando.

Como aún no se ha definido cuál es la clave, ni cómo vamos a tomar los valores que nos van a servir dentro del arreglo (derecha o izquierda de la clave), se ha pensado en recorrer el arreglo con apuntadores para poder desplazarnos los lugares que sean necesarios para luego tomar la candidas de valores indicados que son los datos correctos para la descriptación, por ejemplo, si la clave es 49 seguido de 50. nos desplazamos hacia la izquierda o derecha de acuerdo con las reglas para tomar los valores correctos, pasarlos y poder descriptar los valores que sean de nuestro interés para así poderlos pasar de binarios a enteros y de enteros a carácter y que nos sirva para mostrarlo en la pantalla LCD.

5. Aplicación

Al utilizar el circuito integrado 74HC595, ocupamos solamente 3 salidas digitales en la placa Arduino, pero obtenemos 8 salidas digitales adicionales. Este 3 x 8 en pines no parece una gran mejora (ganamos 5 salidas), pero si se agregan más chips, la ampliación puede llegar a ser importante.

6. diseño de circuito

Entre los miembros del grupo se dio una lluvia de ideas para el montaje de la primera parte del circuito.

6.1. Candidato 1 (Juan Esteban Grisales)

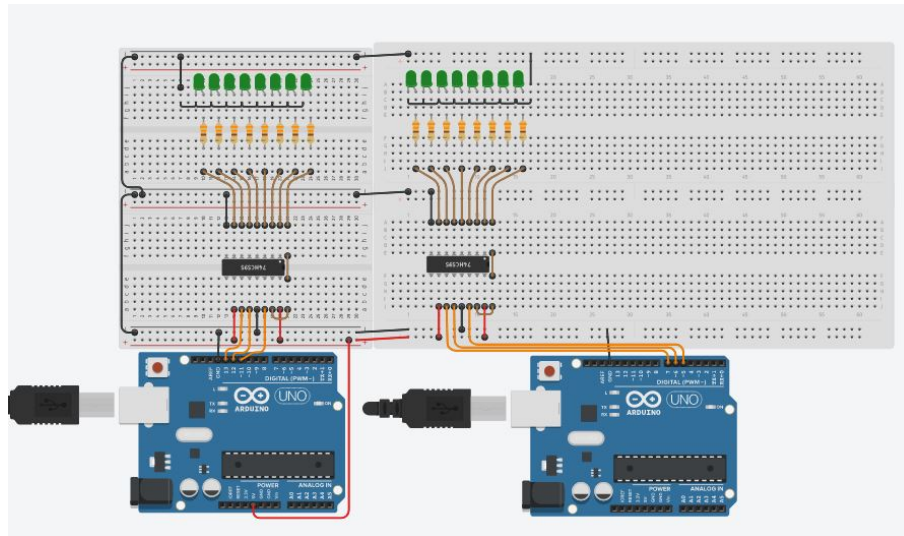


Figura 1: diseño inicial

```
const int SER = 11;
const int RCLK = 12;
const int SRCLK = 13;
#define TIEMPO 100

void reloj(int clk);

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);

}

void loop()
{
    digitalWrite(SER,1);
    /*int bits[8]={0,1,1,0,0,0,0,1};
    int i=0;
    int aux=0;
    while(i<8){
        aux=bits[i];
        digitalWrite(SER,aux);*/

        ////////////////////////////////////

        /*digitalWrite(SRCLK,0);
        digitalWrite(SRCLK,1);
        digitalWrite(SRCLK,0);*/
        reloj(SRCLK);

        ////////////////////////////////////

        /*digitalWrite(RCLK,0);
        digitalWrite(RCLK,1);
        digitalWrite(RCLK,0);*/
        reloj(RCLK);
        //i++;

    delay(TIEMPO);

}

void reloj(int clk){
    digitalWrite(clk,0);
    digitalWrite(clk,1);

```

```

    digitalWrite( clk ,0);
}

```

6.2. Candidato 2 (Julian Montenegro)

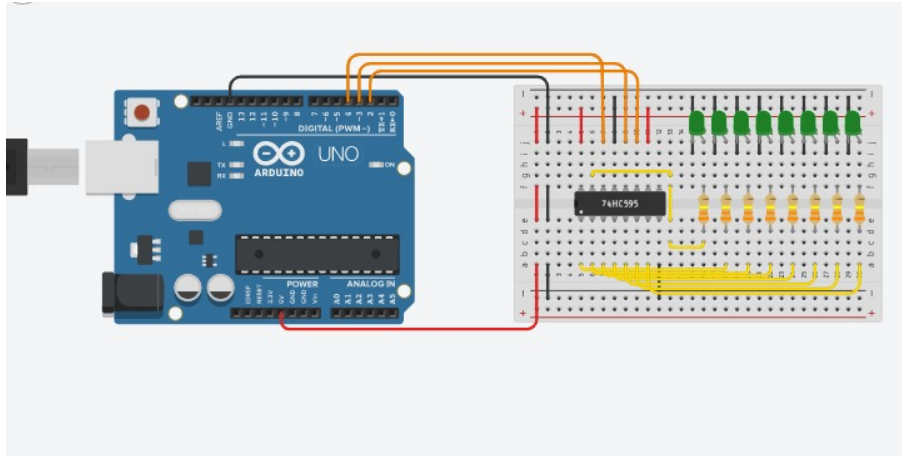


Figura 2: diseño inicial

```

int SER= 2;
int RCLK= 3;
int SRCLK= 4;

void setup()
{
    for(int i=2;i<=4;i++){
        pinMode(i ,OUTPUT);
    }
}

void loop()
{
    for(int i= 0;i<256;;i++){
        digitalWrite(RCLK,0);
        shiftOut(SER,SRCLK,LSBFIRST,i);
        digitalWrite(RCLK,1);
        delay(1000);
    }
}

```

6.3. Avance diseño

Se utiliza el circuito integrado 74HC595 para el ingreso de la clave la cual se mantendra fija, y se ira comparando con los numeros enteros de 8 bits que iran ingresando por el segundo circuito integrado 74HC595, de esta manera cuando el ultimo led (amarillo) se encienda sabremos que los numeros no son iguales.

Se utilizan dos compuertas XOR (74HC86) para comparar bit a bit la clave con el numero que va ingresando, al tener el resultado de cada bit, se comparan los bits entre si con ayuda una compuerta OR (74HC32) para determinar si alguno de ellos es diferente.

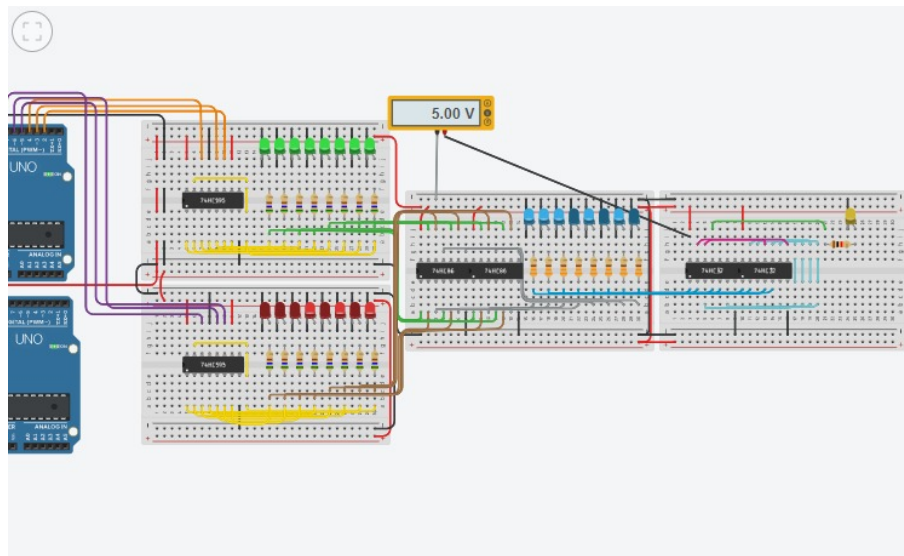


Figura 3: Se acopla sistema de comparación

6.4. Avance sistema de decodificación

Con ayuda del aplicativo Qt creator se elvara una base para el sistema de decodificacion teniendo las reglas previamente dictaminadas con el profesor (clave 21, el dato valido esta 21 posiciones después de la clave y el anterior a éste sea par”).

```
int main() {  
  
    int cadena[500];  
    int num, contador=0, con, clave =21,aumento=0;  
    char mensaje[20], letra;  
    cout<<" Crearemos_cadena_de_caracteres"<<endl;  
    for (int i=0;i!='\0' || i<=500;i++){  
        cout<<" Por_favor_ingrese_un_numero_del_1_al_255:_"<<endl;
```



```

cin>>num;
cadena[i]=num;
cout<<endl<<endl;
for(int j=0;j<=i;j++){
    if(j==i) cout<<cadena[j]<<endl;
    else cout<<cadena[j]<<" , ";
}
contador++;
cout<<endl;
cout<<" -----" <<endl<<endl;

if( contador==40){

    cout<<" Oprima 1 si no desea agregar otro numero" <<endl<<endl;
    cin>>con;
    if( con==1) break;
    else contador=0;
}
}

for(int i=0;i<=500;i++){
    int contadorchar=0;
    if( cadena[i]==clave){

        for(int j=i+1,contador2=1;contador2<=clave;contador2++,j++){

            if( contador2==clave){
                if( cadena[j-1]%2==0){
                    letra=cadena[j];
                    mensaje[contadorchar+aumento]=letra;
                    aumento++;
                }
            }

        }

    }

}

}

cout<<mensaje<<endl<<endl;

}

```

También se pudo avanzar en la parte de recibir los datos ingresados por el puerto serial, los datos se van a recibir como una cadena de caracteres y se

va pasar a una cadena de números o un array, para esto había que tomar cada valor hasta la coma y poderlo convertir a número con la función `atoi`. El siguiente código nos muestra como se hace la conversión de carácter a alfanumérico.

NOTA: nos queda falta para terminar esta parte, un arreglo dinámico que pueda ir almacenando los valores numéricos que nos entrega la función en cada iteración.

```
#include <iostream>
#include <iostream>
#include <sstream>

using namespace std;

string cadena_Array(string cadena);

int main()
{
    string cadena = "56,74,95,13,64,73,89,67,34,75,27,84";
    cadena_Array(cadena);
}

string cadena_Array(string cadena){
    int aux[20]={};
    int l=0,i=0;
    string lectura;
    // Convierte la cadena a un stream
    stringstream input_stringstream(cadena);
    //tomar los valores que estan antes de la coma o el salto de linea.
    while (getline(input_stringstream , lectura ,
    ','))||getline(input_stringstream , lectura , '\n'))
    {
        // funcion que coge cada caracter del arreglo
        // y lo pasa a un numero entero
        l=atoi(lectura.c_str());
        aux[i]=l;
        cout << "Un_valor:_ " << lectura << endl;
        cout << "Un_aux[i]:_" << *aux << endl;
        i++;
    };
    return 0;
};
```

6.5. Tener en cuenta para el diseño

Se investigo acerca de cómo se haría la comunicación entre los dos Arduinos y se encontró que hay una librería que nos ayuda hacer la comunicación entre

estos, esta librería se llama "Wire.h", la cual nos permite que un Arduino quede como Maestro, que es el que controla la información que se le envía al otro Arduino que queda como Esclavo. Esto es posible gracias a que el Arduino tiene integrado un bus de comunicación en serie llamado I2C (Inter-Integrated Circuit). Esta comunicación se hace por 3 puertos llamados SDA,SCL,GND. SDA: por donde se envían los datos SCL: Por donde se envía la señal de reloj GND: La tierra, la cual los dos Arduino deben de estar conectados a la misma masa. Ejemplo de funcionamiento internamente.

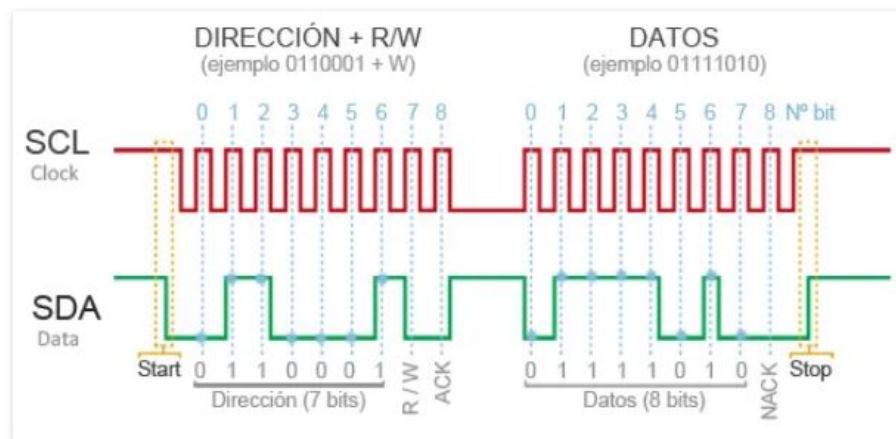


Figura 4: sincronización SCL-SDA

Cuando los datos son enviados por SDA, los pulsos de reloj son enviados por SCL para mantener el Maestro y el Esclavo sincronizados. Y los datos son enviados por bit en cada pulso de reloj. Cuando la comunicación finaliza el maestro transmite una condición de stop para dejar libre el bus.

6.6. Observación

Luego de tener claro como se hace la comunicación de estos Arduino, se determinó que el diseño que teníamos inicialmente y todo lo que se había pensado de como solucionar el problema cambia radicalmente, pero estamos trabajando para lograr el objetivo.

7. Conclusiones

1. Luego de haber investigado como es el funcionamiento del circuito integrado 74HC595, podemos concluir que al ganar 5 salidas del Arduino podremos hacer aplicaciones más complejas con menos Arduino, como hacer tiras de leds o aplicaciones donde podamos mostrar patrones o letras,

por ejemplo, para dar información, como en el metro de medellín que utilizan leds para mostrar un mensaje y en otras funciones donde se hace uso de los leds.

2. Según los resultados de la investigación vemos que el microprocesador será de gran ayuda para el desarrollo del trabajo, permitiéndonos reducir usos y espacios, ayudándonos en funciones necesarias. También conocimos diversas tareas que realiza este pues es capaz de desplazar, almacenar y arrojar información siendo una herramienta útil a la hora del uso de arduino.
3. Después del análisis del desafío que nos llevó a diversos problemas y posibles soluciones que aunque fueron varias, estamos buscando la correcta para plantear, llevarla a cabo y así garantizar un buen trabajo.

8. Sección de contenido

Esta sección es para agregar toda la información correspondiente con código, citas, etc.

8.1. Citación

9. Inclusión de imágenes

.

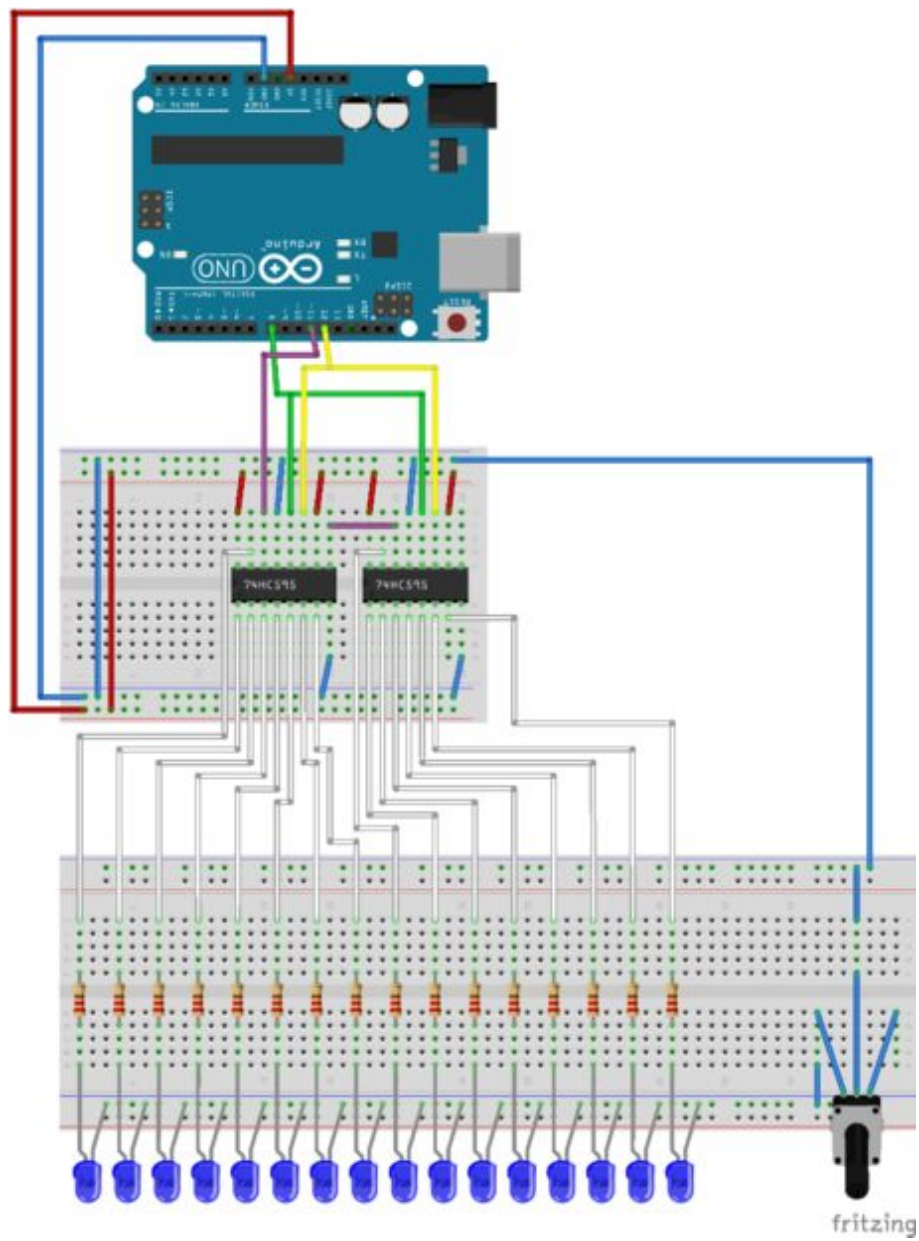


Figura 5: implementación 74HC595