

# Desafío 1

**WANERGE ALMANZA VELASQUEZ**

docente

**Augusto Salazar Jimenez**

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

# Índice

<b>1. Análisis general del problema</b>	<b>2</b>
1.1. Registro de desplazamiento 74HC595 . . . . .	2
1.1.1. ¿Para qué Sirve? . . . . .	2
1.1.2. Utilidad en el desarrollo del desafío . . . . .	2
1.1.3. Ejemplo de implementacion en arduino . . . . .	2
1.2. Comunicación entre dos Arduinos . . . . .	4
<b>REFERENCIAS</b>	<b>5</b>

## **1. Análisis general del problema**

Para la solución del problema es necesario la investigación e implementación del chip 74HC595 y algunos circuitos integrados de lógica. Primero que todo, la información a descryptar se recibirá por medio del puerto serial o de manera manual. En consecuencia, dicha información será almacenada y por medio de pines digitales ser trasladada en su forma binaria al sistema de paralelización (74HC595), después por medio de circuitos de lógica se descifrá su código y se enviará al destino un aviso de la forma de descryptación. Luego de transferir toda la información, el arduino de destino se encargada de separar la información irrelevante con la verdadera. consecuentemente, se mostrará en una pantalla LCD el mensaje oculto.

### **1.1. Registro de desplazamiento 74HC595**

#### **1.1.1. ¿Para qué Sirve?**

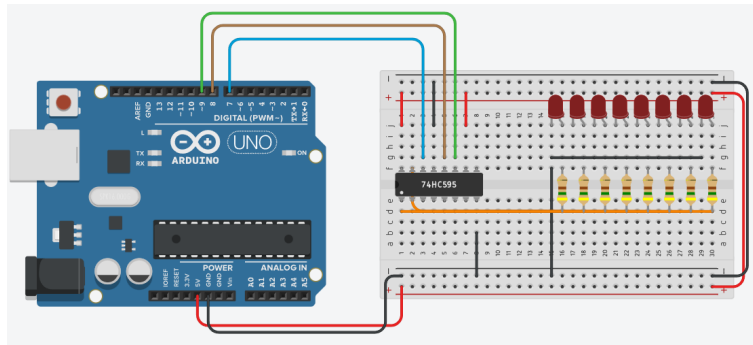
El chip 74HC595 sirve para ampliar el número de pines digitales. De forma más específica, por medio de 3 pines digitales que son: entrada, reloj de almacenamiento, reloj de salida, se puede simular un sin fin de pines digitales de salida. De este modo, el pin de entrada es el encargado de la lectura de los datos en serie y el de reloj de almacenamiento el que determina el avance temporal de dichos datos. Por último, el pin de reloj de salida se encarga de trasladar los datos almacenados hacia las salidas de los pines del 74HC595. [1]

#### **1.1.2. Utilidad en el desarrollo del desafío**

El chip 74HC595 servirá como forma de almacenamiento de información de 8 bits (1 byte), siendo este el sistema encargado de paralelizar los datos, para luego determinar si dicha información será devuelta o no.

#### **1.1.3. Ejemplo de implementacion en arduino**

En este caso se montó un ejemplo típico del funcionamiento del 74HC595, donde en un ciclo se recorre ascendentemente números enteros entre 0 y 255. Para luego ser convertidos a binarios, y después por medio del pin de entrada y el reloj de almacenamiento ingresar los bits del más significativo al menos significativo del número actual. Al mismo tiempo a través del reloj de salida solo se encienden los leds si el número es par, donde los leds representan la forma binaria del numero entero. [2]



Listing 1: Ejemplo código Arduino

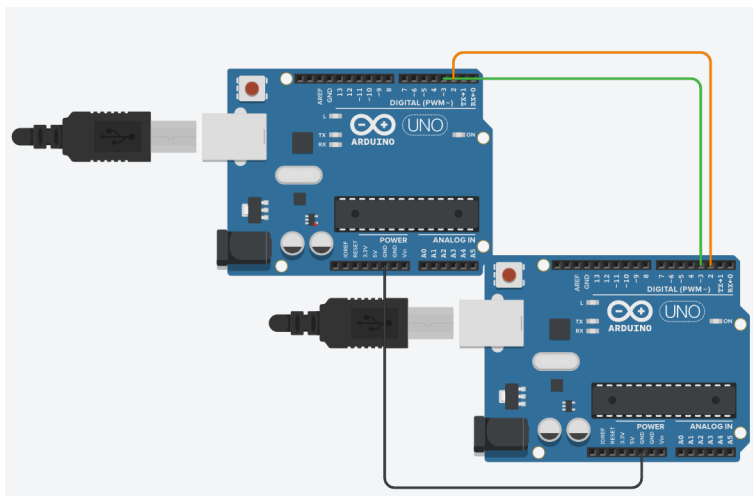
```
#define Entrada 7
#define reloj_salida 8
#define reloj_almace 9

void setup()
{
  pinMode(Entrada , OUTPUT);
  pinMode(reloj_salida , OUTPUT);
  pinMode(reloj_almace , OUTPUT);
}

void loop()
{
  for(int i = 0; i < 256; i++){
    digitalWrite(reloj_salida , LOW);
    shiftOut(Entrada , reloj_almace , MSBFIRST, i);
    if((i%2) == 0 ){
      digitalWrite(reloj_salida , HIGH);
    }
    delay(500);
  }
}
```

## 1.2. Comunicación entre dos Arduinos

La comunicación entre los dos arduinos se hace a través de dos puertos digitales, donde un puerto se utiliza para la transferencia de datos y el otro determina el tiempo de lectura de los datos(reloj). En primer lugar, el arduino 1 lee los datos ingresados por el puerto serial, luego en un ciclo se procesará el dato (byte) comparando y determinando si se enviara un bit alto (1) o bajo (0), después se activará el reloj un tiempo limitado, que es el que indicara cuando el bit se puede leer. En segundo, tenemos al arduino 2 que constantemente está leyendo si el puerto del reloj está activado (1), y si es así, tomara de referencia en qué estado está el puerto de lectura, empezando desde el bit menos significativo al más significativo, y se irán sumando para formar el byte. por último, se imprimirá en el arduino 2 el byte completo, y el proceso se repetirá hasta que no haya más datos. [3]



## Referencias

- [1] B. Ar. 74hc595 registro de desplazamiento (shift register). [Online]. Available: <https://www.youtube.com/watch?v=LFqIA3ZvZE8>
- [2] L. del Valle Hernández. Utilizar pulsadores en arduino. [Online]. Available: <https://programarfacil.com/blog/utilizar-pulsadores-en-arduino/>
- [3] AbrahamG. Comunicación entre dos arduinos con el puerto serie. [Online]. Available: <https://www.automatizacionparatodos.com/comunicacion-entre-dos-arduinoss-con-el-puerto-serie/>