

# **Parcial 1: Informática II**

Análisis y diseño.

**JOSE MIGUEL GOMEZ MONSALVE**

**ERIKA DAYANA LEÓN QUIROGA**

**DAVID AGUDELO OCHOA**

Departamento de Ingeniería Electrónica y Telecomunicaciones

Universidad de Antioquia

Medellín

Febrero 2022

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Integrado 74HC595.</b>	<b>3</b>
2.1. Características del integrado. . . . .	3
2.2. Funcionamiento. . . . .	4
2.3. Aplicaciones. . . . .	7
<b>3. Comunicación entre Arduinos</b>	<b>7</b>
3.1. Intento número 1 . . . . .	7
3.1.1. arduino emisor . . . . .	7
3.1.2. arduino emisor y receptor . . . . .	9
<b>4. Cibergrafía</b>	<b>10</b>

## 1. Introducción

## 2. Integrado 74HC595.

El integrado 74HC595 hace parte de la familia de dispositivos SNx4HC59, la cual contienen un registro de desplazamiento de 8 bits de entrada en serie y salida en paralelo, el registro de almacenamiento tiene salidas de 3 estados paralelos. Se proporcionan relojes separados para el registro de desplazamiento y el de almacenamiento. El registro de desplazamiento tiene una entrada de anulación directa (SRCLR), una entrada en serie (SER) y salidas en serie para la conexión en cascada. Tienen una amplia corriente de funcionamiento de 2 V a 6 V, y las salidas de 3 estados de alta corriente pueden controlar hasta 15 cargas LSTTL. Los dispositivos tienen un bajo consumo consumo de 80-A (máximo) ICC.

### 2.1. Características del integrado.

- Entrada serial, salida paralela, o salida serial que permite la conexión en cascada de varios integrados.
- Registro de desplazamiento de 8 bits que alimenta a un registro de almacenamiento.
- Entradas de reloj separadas para el registro de desplazamiento y el de almacenamiento con activación por flanco de subida.

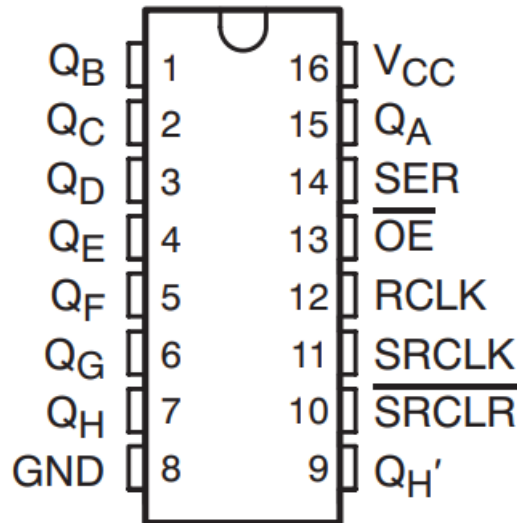


Figura 1: Integrado 74HC595.

#### Configuración de pines.

- Los pines de  $Q_B$  (pin 1) a  $Q_H$  (pin 7), añadiendo  $Q_A$  (pin 15) representan las salidas del integrado.
- $V_{CC}$  (pin 16) es la alimentación y GND (pin 8) se conecta a tierra.
- $Q_{H'}$  (pin 9) se utiliza para conectar otro integrado 74HC595 y generar un efecto de cascada.
- El pin 14 o SER es el pin donde se envían los datos.
- $\overline{OE}$  (pin 13) llamado Output Enable, habilita las salidas y se activa con un nivel bajo, por lo cual, para que siempre esté activo se conecta a GND.

- El RCLK (pin 12) es el reloj del registro de almacenamiento y se utiliza para actualizar los datos a los pines de salida.
- El SRCLK (pin 11) es el reloj que sincroniza la carga de datos.
- El  $SR\bar{C}LR$  (pin 11) llamado Shift Register Clear, reestablece el registro de desplazamiento.

## 2.2. Funcionamiento.

El objetivo principal es pasar el número dado de un formato serial a uno paralelo. Para explicar el funcionamiento del integrado tomaremos un número cualquiera de 8 bits, este número se irá guardando bit por bit en cada uno de los cuadros que se pueden ver en la Figura 2, también podemos ver en esta misma figura que la entrada de los datos es en serie (uno por uno), y la salida de ellos es en paralelo (8 bits).

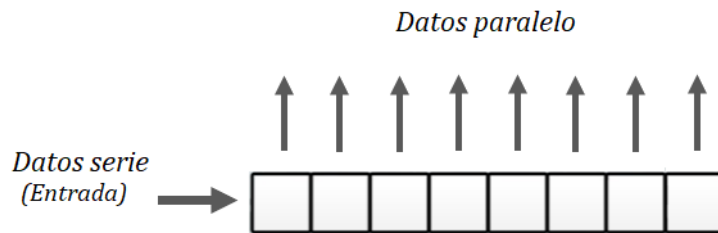


Figura 2: Representación del funcionamiento del circuito integrado 74HC595.

Para que la toma de los datos sea exitosa es necesario un reloj que por medio de pulsos, controlará en qué momento ingresa al integrado el bit presente en la entrada. Tomaremos de ejemplo la representación binaria del número 49, la cual es 00110001.

El primer paso para transformar la información que se encuentra en serie a paralelo es realizar el desplazamiento de los bits dentro del integrado iniciando por el bit más significativo (MSB por sus siglas en inglés), en nuestro ejemplo es un cero, que se encuentra presente en la entrada y que en el primer pulso del reloj ingresa a la primera posición del registro de desplazamiento. Figura 3

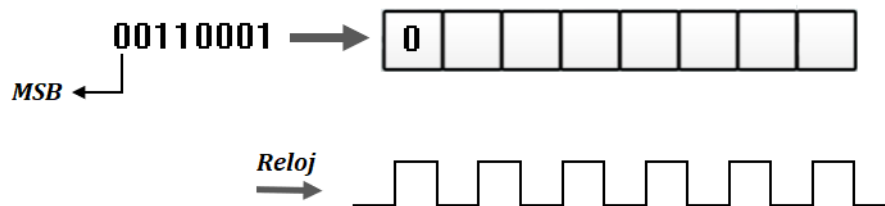


Figura 3: Entrada del MSB al integrado.

En el próximo pulso del reloj el MSB, ya dentro del integrado, se correrá una posición a la derecha en el registro de desplazamiento, mientras que el número a la derecha del MSB, en la entrada, se posicionará en la primera posición del registro de desplazamiento.

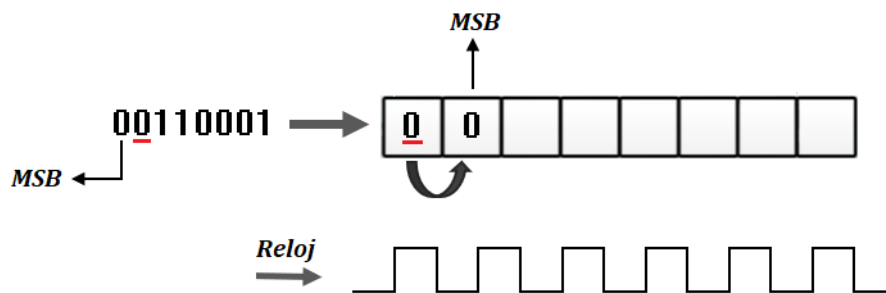


Figura 4: Desplazamiento de los bits dentro del integrado.

Este proceso se repetirá hasta que se ingrese al registro de desplazamiento el último bit del número (Figura 6).

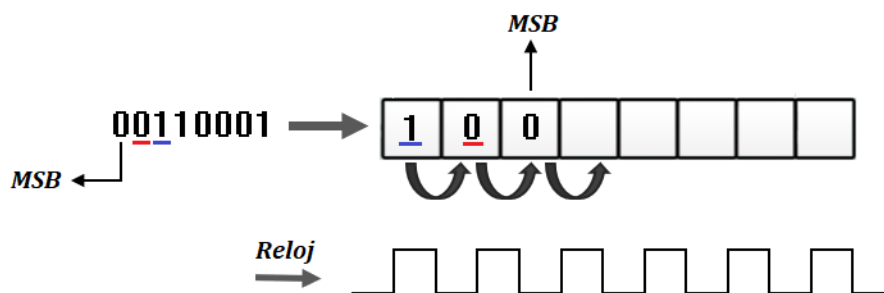


Figura 5: Desplazamiento de los bits dentro del integrado.

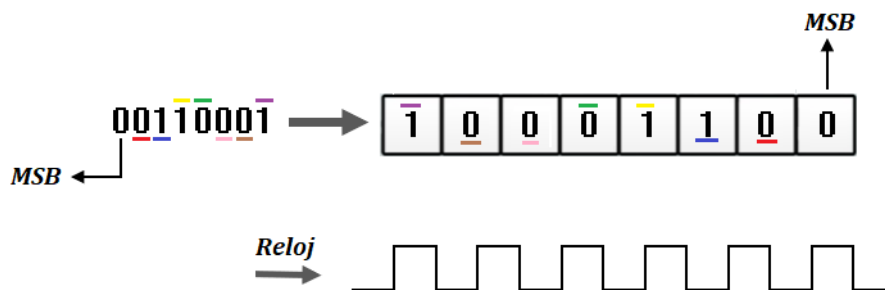


Figura 6: Registro de desplazamiento lleno. Se utilizaron colores encima o abajo de los bits para identificar fácilmente cuál es cuál en el registro de desplazamiento.

Para que las salidas no vayan cambiando mientras que el registro de desplazamiento se llena, el integrado hace uso del registro de almacenamiento y el RCLK (Rejol del registro de almacenamiento). Figura 7

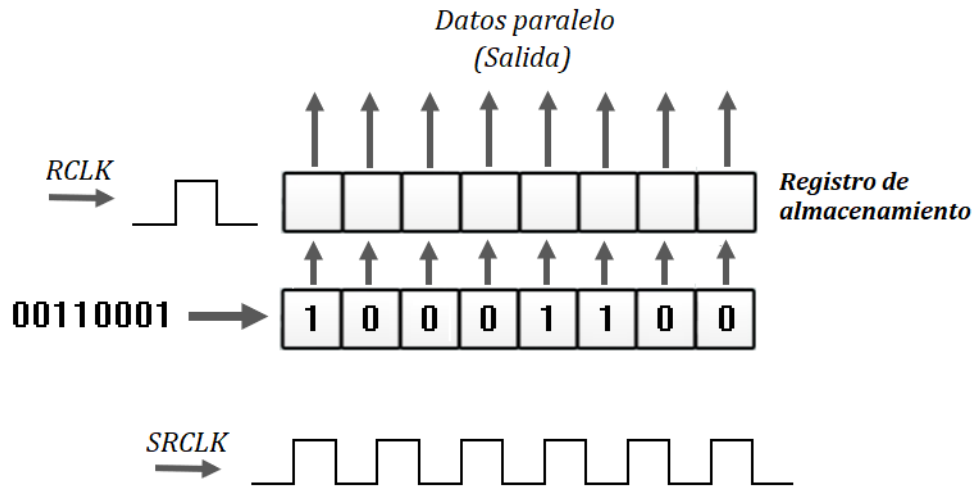


Figura 7: Representación del registro de almacenamiento.

Una vez se finalice la carga de datos en el registro de desplazamiento, con un único pulso del RCLK se cargan todos los datos del registro de desplazamiento al registro de almacenamiento y se muestran en la salida. De esta forma se alcanzará el objetivo principal, transformar los datos de entrada en serie a datos de salida paralelos.

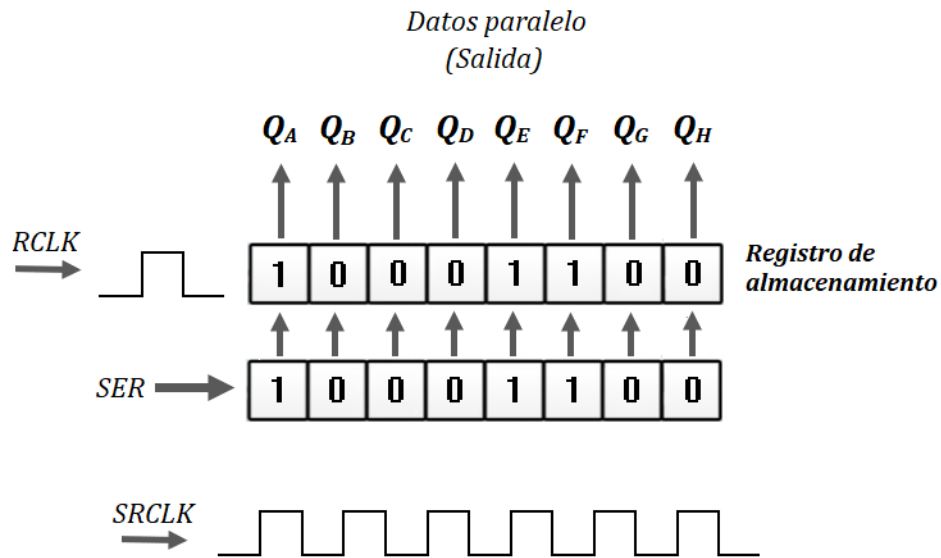


Figura 8: Representación del registro de almacenamiento.

## 2.3. Aplicaciones.

- Network switches
- Power infrastructure
- LED displays
- Servers

## 3. Comunicación entre Arduinos

### 3.1. Intento número 1

A continuación se presenta el código y montaje de arduinos que se usaron para probar el tema de comunicación entre dos arduinos. Cabe aclarar que la implementación de estos se dio antes de las indicaciones y especificaciones de la clase del día 17 de febrero, por lo que no reflejan el futuro diseño a implementar, el cual contará con una gran participación del integrado 74HC595 y se sospecha que también del protocolo I2C para la señal de reloj.

#### 3.1.1. arduino emisor

montaje:

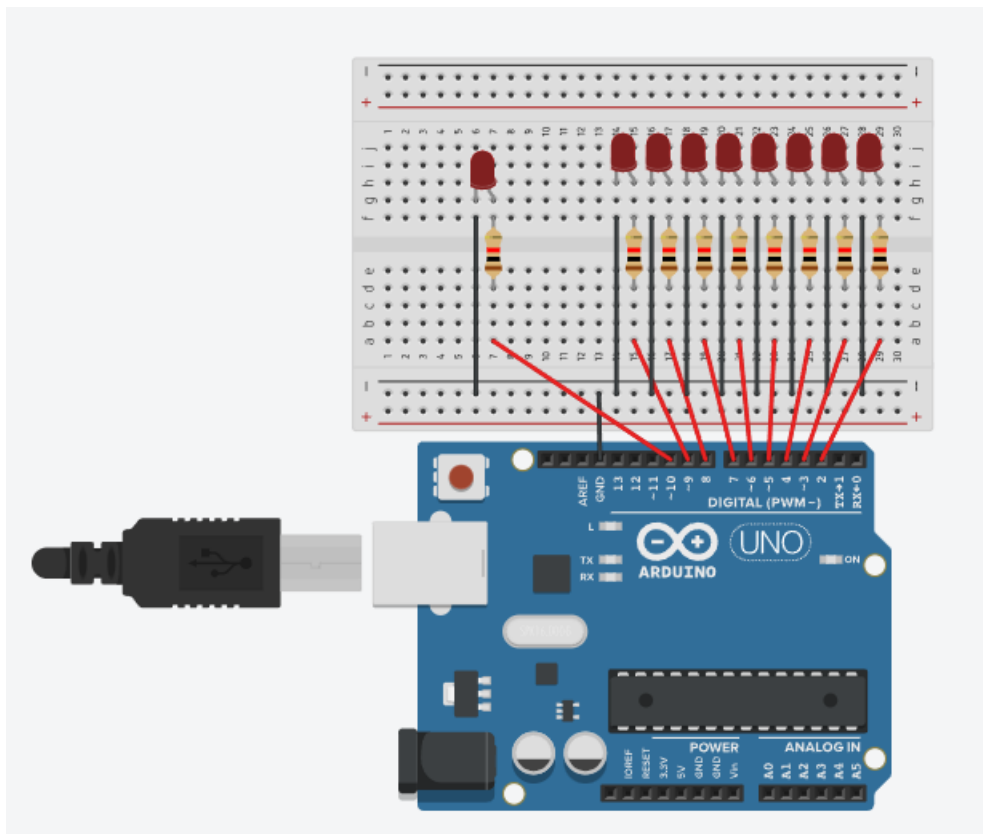


Figura 9: Prueba de emisión de bits y una señal de reloj por medio de 9 leds.

código:



```

1  #define tiempo 1000
2  byte codigo[8]={1,2,3,4,5,6,7,8}; //A byte stores an 8-bit unsigned
   number, from 0 to 255.
3  //-----
4  //prototipo de las funciones
5
6  //-----
7  //setup
8  void setup()
9  {
10     for(int i=2;i<11;i++)
11     {
12         pinMode(i, OUTPUT);
13     }
14 }
15
16 //-----
17 //loop
18 void loop()
19 {
20     for(int i=0;i<8;i++)
21     {
22         digitalWrite(10, LOW);
23         delay(25); // Wait for 1000 millisecond(s)
24         for(int j=0;j<8;j++)
25         {
26             digitalWrite(j+2, bitRead(codigo[i], j));
27         }
28         delay(tiempo-25);
29         digitalWrite(10, HIGH);
30         delay(tiempo);
31     }
32 }
33
34 }
35 //-----
36 //cuerpo de las funciones

```

### 3.1.2. arduino emisor y receptor

montaje:

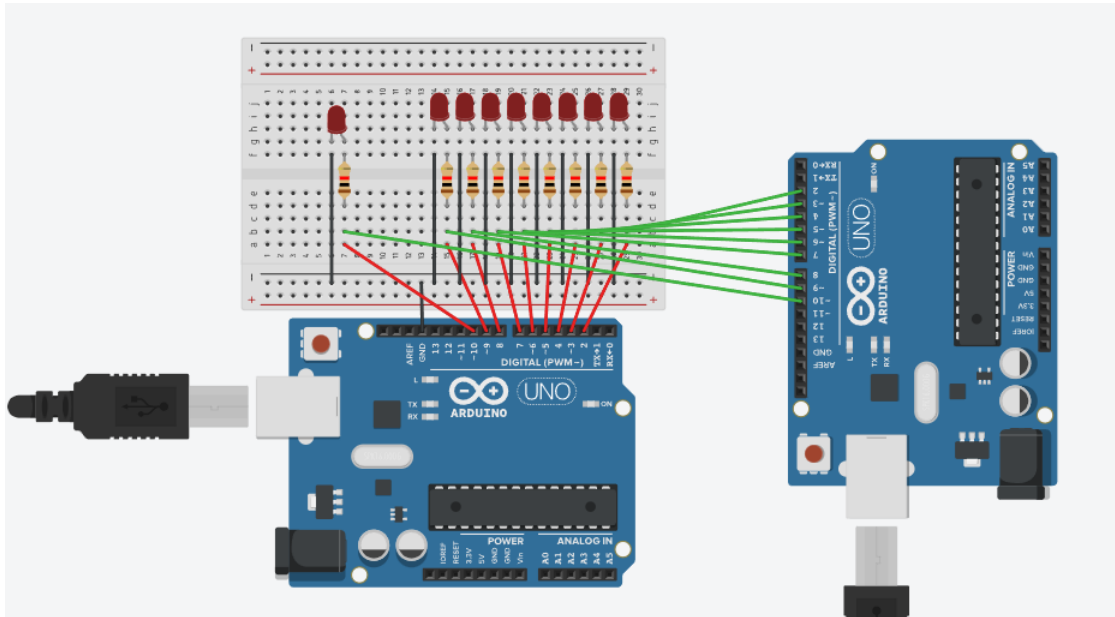


Figura 10: Prueba de emisión de bits y una señal de reloj por medio de 9 leds y su recepción a otro arduino.

código arduino emisor:

```
37 //directivas de preprocesamiento
38 #define tiempo 250
39 byte codigo[8]={2,1,3,4,5,6,7,8}; //A byte stores an 8-bit unsigned
    number, from 0 to 255.
40 //-----
41 //prototipo de las funciones
42
43 //-----
44 //setup
45 void setup()
46 {
47     for(int i=2;i<11;i++)
48     {
49         pinMode(i, OUTPUT);
50     }
51 }
52 //-----
53 //loop
54 void loop()
55 {
56     for(int i=0;i<9;i++)
57     {
58         digitalWrite(10, LOW);
59         delay(50); // Wait for 1000 millisecond(s)
60         for(int j=0;j<8;j++)
```

```

62     {
63         digitalWrite(j+2, bitRead(codigo[i], j));
64     }
65     delay(tiempo-50);
66     digitalWrite(10, HIGH);
67     delay(tiempo);
68 }
69
70
71 }
72 //-----
73 //cuerpo de las funciones

```

código arduino receptor:

```

75 // C++ code
76 //
77 int entero=0;
78 bool primerCero=false;
79 void setup()
80 {
81     Serial.begin(9600);
82     for(int i=2;i<11;i++)
83     {
84         pinMode(i, INPUT);
85     }
86 }
87
88 void loop()
89 {
90     if(primerCero==true)
91     {
92         for(int i=0;i<8;i++)
93         {
94             entero=entero+((int)digitalRead(i+2))*pow(2,i);
95         }
96         Serial.println(entero);
97         entero=0;
98         primerCero=false;
99     }
100     if(primerCero==false)
101     {
102         while(digitalRead(10)==true)
103         {
104             primerCero=true;
105         }
106     }
107 }
108 }

```

## 4. Cibergrafía

(PONERLA CORRECTAMENTE) <https://www.electrogeekshop.com/como-funciona-el-74hc595-shift-register-y-su-interfaz-con-arduino/>

(PONERLA CORRECTAMENTE) <https://github.com/trihedral/ArduinoLatexListing>