Portada: (lista en overleaf)

**Conceptos:**

-Arduino:

Es una placa electrónica donde se pueden conectar periféricos de entrada y salida de un microcontrolador, es de código abierto, el dispositivo permite crear códigos de diferentes tipos de microcontroladores.

-Puerto Serie:

El término serial se refiere a los datos enviados mediante un solo hilo. Los bits se

envían uno detrás de otro como se muestra en el siguiente esquema (figura 1) en el cual

está pasando información desde el equipo A hacia el equipo B con una conexión

en serie.

-Puerto Paralelo:

Es un tipo de puerto que permite el intercambio de paquetes simultáneos (en ambos sentidos) de datos por medio de hilos o cables. La forma en que funciona este puerto es a nivel físico, este tiene un cable por cada bit que se envía, por eso se llama puerto paralelo.

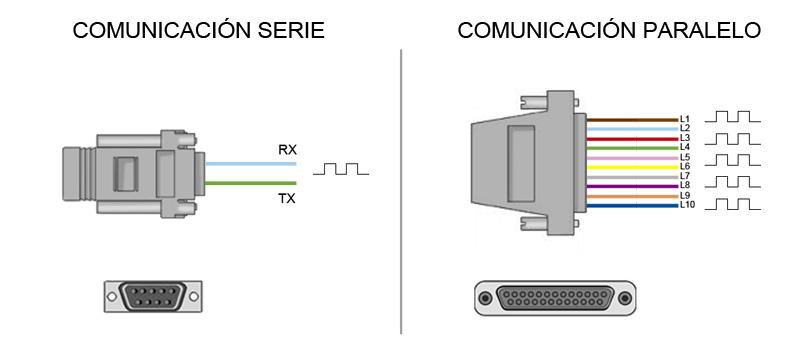


Figura 1. Comunicación serie y paralelo

-74HC595:

**Características:**

* Maneja un voltaje mínimo de -0.6V a un máximo de +5 V
* Tiene 8 biestables para almacenar 8 bits. El pin de salida o patillas, Vcc y GND para la alimentación, y Q que son las de datos. El resto corresponden a señales de reloj/control.
* Las entradas la tiene en serie y la salida en paralelo,con una sola entrada, se pueden controlar a la vez esas 8 salidas. Solo necesitarás tres pines del microcontrolador usado para manejarlo. Esas son Latch, Clock y Data. Latch es el pin 13 en este caso. Clock puede estar en el 11 u otros, y el bit de datos es el 14.
* La señal de reloj alimenta el circuito, es el ritmo, los datos cambian el comportamiento del chip, pasa de LOW a HIGH , con esto se consigue es grabar la posición actual donde se encuentre el desplazamiento el valor ingresado por este pin de datos. Si repites esto 8 veces, entonces habrás grabado las 8 posiciones y tener un byte almacenado (Q0-Q7).
* Salida de datos de tres estados de salida paralela de 8 bits (tri-estado: nivel bajo, nivel alto y estado de alta impedancia)
* Con frecuencia de cambio de 100MHz.
* La salida en serie puede controlar el chip en cascada del siguiente nivel

**Aplicaciones:**

* Conmutadores de red
* Infraestructura de energía
* Pantallas LED
* Servidores

**Descripción:**

El 74HC595 es un chip de registro de desplazamiento de 8bit de entrada serial y salida serial y paralela que normalmente se utiliza como chip de expansión de los cuales se usan 3 pines de la MCU y este nos permite usar 8 pines del chip.

**Configuración de pines y funciones:**

74HC595

PIN SÍMBOLO DESCRIPCIÓN

1 QB Salida paralela 1

2 QC Salida paralela 2

3 QD Salida paralela 3

4 QE Salida paralela 4

5 QF Salida paralela 5

6 QG Salida paralela 6

7 QH Salida paralela 7

8 GND Conexión a tierra

9 QH’ Conexión a otro chip 74HC595

10 SRCLR Resetea el registro de desplazamiento (activa con nivel bajo)

11 SRCLK Reloj de sincronización de los datos

12 RCLK Reloj registro de almacenamiento

13 OE Habilita las salidas (se activa con un nivel bajo)

14 SER Ingreso de los datos serial

15 QA Salida paralela 0

16 VCC Alimentación del chip (5v)

**Observación:** El circuito integrado 74HC595 es un registro de desplazamiento que cuenta con entrada en serie y salida en paralelo de 8 bits (SIPO). Se alimenta un registro de almacenamiento de tipo D (datos).

El 74HC595 es de gran ayuda cuando se requiere ampliar la cantidad de salidas digitales ya que se puede conectar 8 LEDs con tan solo 3 pines del microcontrolador o Arduino.

Resumen:

Introducción:

Metodología:

Procedimos a leer, analizar y discutir el texto del desafío, y decidimos como grupo de trabajo que cada quien se instruya sobre los conocimientos necesarios para llevar a cabo el desarrollo del desafío, como lo es Arduino, c++, manejo de repositorios, chip 74HC595, puertas lógicas, con el fin de hacer uso de estos, aplicándolos al desarrollo.

De forma grupal, nos guiamos de algunos ejemplos de Arduino que utilizan el chip 74HC595, con el fin de entender de una mejor manera el uso de este chip en la vida real.

En el primer arduino (figura 2), entendimos a partir del código (figura 3) que el chip lo podemos usar para convertir información serial a paralela y entendimos la forma de desplazamiento de los datos, evidenciando esto con la iluminación de cada led.

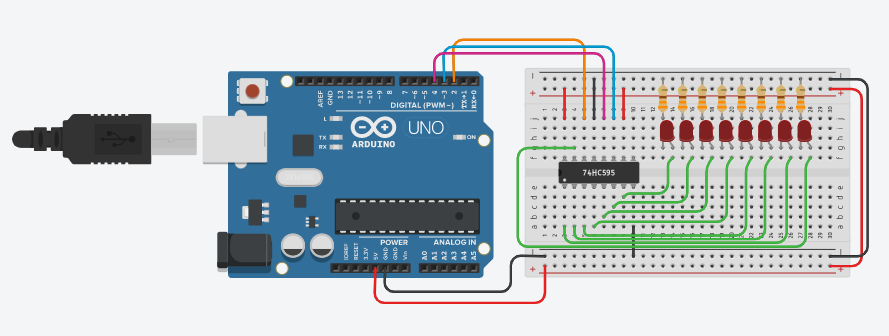


Figura 2. Ejemplo arduino 1

| #define pinDATA 2 //SER  #define pinCLOCK 3 //SRCLK  #define pinLATCH 4 //RCLK  void setup() {  Serial.begin(9600);  pinMode(pinDATA, OUTPUT);  pinMode(pinCLOCK, OUTPUT);  pinMode(pinLATCH, OUTPUT);  digitalWrite(pinCLOCK,LOW); //SE INICIA EN BAJO POR SI NO SE HA DECIDIDO EL DATO QUE SE VA ENVIAR  digitalWrite(pinLATCH,LOW); //SE INICIA EN BAJO POR SI TIENE ALGO EN MEMORIAL  }  void loop()  {  byte j = B00000001; // ES IGUAL A 1 EN BINARIO 00000001  for(int i = 0; i < 8; i++)  {  escribir(j);  j<<= 1;  Serial.println(j);  delay(500);  }  }  //FUNCION ESCRIBIR EN CHIP LO QUE SE VA A MANDAR  void escribir(byte data)  {  byte h = B10000000; // h es ugual a 128 en binario 1000000  for(int i = 0; i < 8; i++)  {  if(data == h){  digitalWrite(pinDATA,HIGH);//DATA=1 // INDICAR QUE ES UN 1 LO QUE VA A ENVIAR  }  else  {  digitalWrite(pinDATA,LOW);//DATA=0 // INDICAR QUE ES UN 0 LO QUE VA A ENVIAR  }    h >>= 1;  digitalWrite(pinCLOCK,HIGH);//CLOCK=1 se envía el valor que se eligió de pinDATA  digitalWrite(pinCLOCK,LOW);//CLOCK=0  }  digitalWrite(pinLATCH,HIGH);//LATCH=1  digitalWrite(pinLATCH,LOW);//LATCH=0  } |
| --- |

Figura 3. Código arduino 1

En el segundo arduino (figura 4), a partir del código (figura 5), que podemos controlar de manera manual la iluminación de cada led.

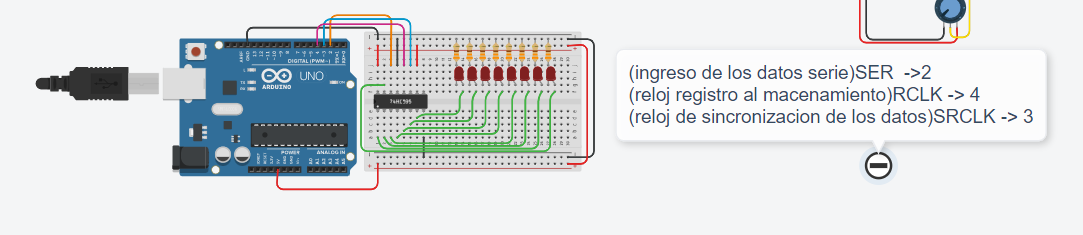


Figura 4. Ejemplo arduino 2

| #define SER 2 //ingreso de los datos serial  #define RCLK 4 //reloj registro de almacenamiento  #define SRCLK 3 //reloj de sincronizacion de los datos  #define LED0 1  #define LED1 2  #define LED2 4  #define LED3 8  #define LED4 16  #define LED5 32  #define LED6 64  #define LED7 128  void setup() {  pinMode(SER, OUTPUT);  pinMode(RCLK, OUTPUT);  pinMode(SRCLK, OUTPUT);  }  void loop()  {  digitalWrite(RCLK, LOW);  shiftOut(SER,SRCLK, MSBFIRST, LED0);  digitalWrite(RCLK, HIGH);  delay(500);  } |
| --- |

Figura 5. Código arduino 2

Análisis:

Se debe enviar de un arduino a otro una trama de datos que se encuentra encriptada, se debe leer la trama, verificar qué datos son los reales y mostrarlos en otro arduino. Para esta comunicación se utilizará un chip 74HC595.

Conclusiones:

Referencias:

<https://www.ti.com/lit/ds/symlink/sn74hc595.pdf?ts=1644870203944&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FSN74HC595>

<https://programarfacil.com/blog/74hc595-registro-de-desplazamiento-arduino/>

<https://www.hwlibre.com/74hc595/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+hwlibreweb+%28Hardware+libre%29>