Informe

Wilmer Banquez, David Arango, y Juliana Martínez

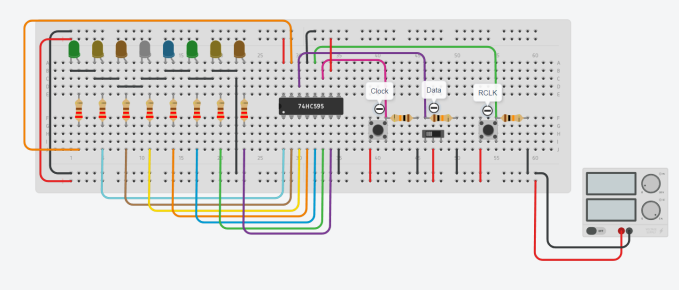
***Index Terms*—Arduino, Integrado 74HC595, etc.**

I. Introducción

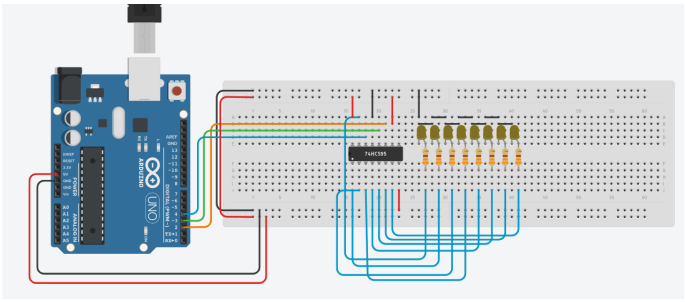
EL chip 74HC595 es un circuito impresor, concretamente es un registro de desplazamiento de 16 pines que sirve para convertir datos de serie a paralelo de una forma eficaz y sencilla. En el siguiente informe se encuentran dos ejemplos de cómo utilizar dicho circuito, después detallamos la utilidad que podemos darle para solucionar la implementación del sistema de encriptación en los dos arduinos. Por último se plantea un posible diseño.

II. Circuito integrado 74HC595

Como primera medida, se propuso el ejercicio de representar números binarios de forma manual usando una protoboard, un suministro de energía, un switch deslizante, dos pulsadores, luces led, el circuito integrado 74HC595, resistencias y cables. La primera etapa del ejercicio se basa en la estructuración general de los cables para efectuar la conexión entre el circuito integrado, los pulsadores, el switch y demás componentes. El funcionamiento constaba de un switch deslizante (data) que dependiendo del estado en el que se encontrara (encendido o apagado) comunicaba cierta información (1 o 0), a continuación el primero de los pulsadores recibe esta información las veces que fuera accionado. Por último, el segundo pulsador toma el registro y lo expone a la salida en este caso reflejándolo en las led.



https://www.tinkercad.com/things/fdPmct90xWs.
En el segundo ejemplo aplicado se utilizaron luces leds, resistencias, un arduino, una protoboard, cables y obviamente el circuito integrado 74HC595. El objetivo a cumplir era convertir un caracter ingresado por el usuario a codigo binario, principalmente mediante hardware. Una vez más se llevaron a cabo todos los cableados necesarios para que la conexión entre las componentes fuera óptima.



https://www.tinkercad.com/things/ajHYnGZvnA2.
A continuación se usaron las resistencias y las LED para representar la conversión del caracter a código binario realizada por el propio circuito. En lo que respecta al código, se tomaron medidas específicas como restringir el rango de los caracteres ya que si éste se encontraba fuera de 0 y 255 no correspondería a código ASCII lo cual arrojaría un error.

```
#define PIN_CLOCK 2
#define PIN_RCLK 3
#define PIN_DATA 4
void setup() {
  pinMode(PIN_CLOCK, OUTPUT);
  pinMode(PIN_RCLK, OUTPUT);
  pinMode(PIN_DATA, OUTPUT);

  Serial.begin(9600);

}
void loop() {

  if (Serial.available()) {
    char caracter = Serial.read(); //Si es que si, lo leemos. En este caso la variable será de tipo char
    if (caracter >= 0 && caracter < 256) { //No podemos representar carácteres cuyo código no esté entre 0 y 255
      digitalWrite(PIN_RCLK, LOW); //Le decimos que vamos a escribir algo...
      shiftOut(PIN_DATA, PIN_CLOCK, MSBFIRST, caracter); //Escribimos el carácter que el usuario proporcionó
      digitalWrite(PIN_RCLK, HIGH) ; //Y le indicamos que lo exponga a la salida dando un pulso del reloj
    }
  }
}
```

III. Utilidad

Sabiendo que el circuito integrado paraleliza los datos que recibe, éste facilitaría la labor de desencriptar la información mediante el método que se requiera.

IV. Conclusiones

¿Qué sucede con las tablas?
En la documentación se encuentra el código, explicado y con ejemplos, para esto.

Referencias

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.