

Informe de Análisis y Diseño Parcial 1

**YESIKA MILENA CARVAJAL DÍAZ
NICOLL CAROLINE CHAZATAR
ANDRES FELIPE ZULUAGA ORTÍZ**

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

Índice

1. INTRODUCCIÓN	2
2. RESÚMEN	2
3. OBJETIVOS	2
3.1. GENERAL	2
3.2. ESPECÍFICOS	2
4. METODOLOGÍA	2
4.1. Circuito integrado 74HC595	3
4.2. Comunicación entre Arduinos	4
4.3. Análisis del parcial	9
5. DESARROLLO	9
6. CONCLUSIONES	9
7. REFERENCIAS	10

1. INTRODUCCIÓN

En este informe se habla sobre el análisis realizado al proyecto parcial que se entregará próximamente. Se estudia el funcionamiento del circuito integrado 74HC595, se profundiza en el funcionamiento del Arduino y en comandos para su funcionamiento, a su uso en la plataforma de Tinkercad y se observa la interconexión entre arduinos.

2. RESÚMEN

El informe busca obtener herramientas y conocimiento que ayuden a realizar el proyecto del curso de Informática II. Se analiza el proyecto de forma general y además se profundiza en los puntos referentes al circuito integrado 74HC595, en las señales de dato y de reloj y en la interconexión entre arduinos, además de sus códigos. Se concluye la utilidad de las herramientas Tinkercad, Arduino, C++ y del circuito integrado 74HC595.

3. OBJETIVOS

3.1. GENERAL

Adquirir herramientas y conocimientos de apoyo para el análisis teórico del desarrollo del proyecto parcial.

3.2. ESPECÍFICOS

- Desarrollar la capacidad de solución de problemas.
- Evaluar la capacidad del estudiante para trabajar con Arduino.
- Usar de manera adecuada las funciones de la plataforma Arduino explicadas en clase que permiten controlar el puerto serial y los puertos digitales.

4. METODOLOGÍA

El proyecto se divide en 6 bloques:

1. PC1: Recibe los datos vía serial y es el encargado de enviar los datos al primer Arduino.
2. Sistema de generación de información serial: Este es el primer Arduino, el cual recibe la información que envía el PC1, y envía señal de datos y reloj. Los cuáles reciben el segundo Arduino, y el sistema de paralelización.

3. Sistema que paraleliza los datos: En este bloque, se paraleliza la información de un byte, es decir, esta recoge un byte y lo separa en 8 bits independientes. Los 8 bits independientes entran paralelamente al circuito de verificación de la banda (sistema de desenscriptación). La paralelización de los datos se realiza por medio del circuito integrado 74HC595.
4. Sistema de desenscriptación: Este sistema recibe los datos binarios paralelizados y con respecto a estos se genera de salida una bandera. Si la bandera es verdadera, significa que el próximo byte será el mensaje real. Este sistema se crea con base al byte que verifica el ingreso del mensaje real, y se realiza con compuertas lógicas.
5. Sistema de recepción: Este sistema tiene tres entradas: los datos, la señal del reloj y la bandera de verificación del mensaje real. Los datos y la señal del reloj se ingresan desde el sistema de generación de información serial, y la bandera entra desde el sistema de desenscriptación. El sistema de recepción almacena el dato correspondiente al mensaje real para posteriormente entregarlo al Arduino 2. La señal del reloj sirve para indicarle al sistema de recepción en qué momento empieza la recolección del mensaje real, la cual es de 8 flancos de subida o de bajada del reloj después de la activación de la bandera. Este bloque se encuentra en el Arduino 2. PC2: Este bloque recibe la señal del mensaje real, desde el sistema de recepción, y lo reproduce en una pantalla LCD.

4.1. Circuito integrado 74HC595

El circuito integrado es útil en la realización de la práctica en la creación del bloque Sistema que paraleliza los datos. El 74HC595 nos ayuda a paralelizar los bytes en sus 8 puertos de salida para que estos entren al Sistema de desenscriptación.

Los pines de alimentación del integrado son el pin 8 para tierra y 16 para Vcc.

Se debe generar una diferencia de potencial en los pines MA y OE para habilitar las salida del integrado, por lo que el pin OE se conecta a tierra y MA a Vcc.

Los pines SH-CP(11), DS(14), ST-CP(12) controlan el ingreso de los bits que se reproducen en las salidas del integrado. El pin DS da el valor del bit, el SH-CP es la señal para tomar el bit de DS, y ST-CP muestra los bits almacenados previamente en las salidas del integrado. Las señales reproducidas inician desde la salida Q0 hasta Q7, que corresponden respectivamente a los pines 15 y 1-7. [1]

Para el uso del circuito integrado independiente se realiza el monje de la figura 1, en donde el switch nos da el valor del bit, el botón izquierdo almacena el bit, y el botón derecho manda la señal al circuito para que los bits sean visibles.

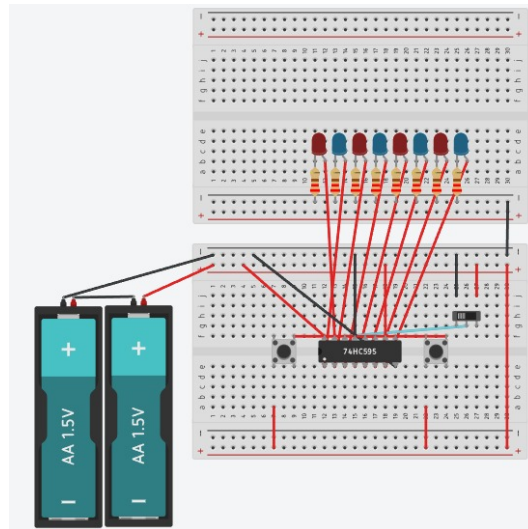


Figura 1: Circuito independiente 74HC595.

4.2. Comunicación entre Arduinos

Para realizar la comunicación entre arduinos realizamos dos señales en el arduino transmisor (Arduino 1), las cuales son la señal de datos y la del reloj.

Para las señales de datos se utilizaron varias señales de prueba. Las señales de prueba para los datos fueron las siguientes:

- 01110010
- 00110101
- 10001100
- 11111101010100101

La lectura de los datos se hará con cada flanco de subida del reloj.

Las señales de prueba se realizaron en el loop de Arduino y se utilizaron funciones delay. Los tiempos de delay se utilizan para entender cuándo se deben hacer modificaciones tanto en las señales de datos como en las del reloj.

Por ejemplo; para la primera señal de prueba para los datos, las señales de datos y de reloj se pueden observar como las señales de la figura 2, en donde t_2 corresponde al tiempo entre la generación del bit de la señal de dato y el flanco de subida del reloj, y t_1 corresponde al periodo de la señal de reloj.

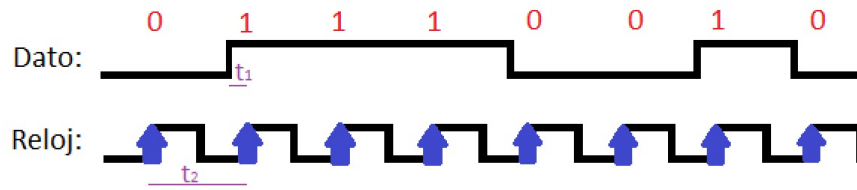


Figura 2: Señales para la primera prueba de la señal de dato

Para el t_2 de la prueba, su valor es de 1 segundo. Se utilizan 500 milisegundos para el valor en alta y 500 milisegundos en baja. Para t_1 se utiliza un valor de 250 milisegundos.

Para la interconexión entre los arduinos se realiza el montaje de la figura 3, en donde en el arduino 1 el puerto 6 es la señal del reloj, el puerto 7 es la del bit a recolectar, y el puerto 6 del arduino 2 es el bit que se recibió visto como flanco.

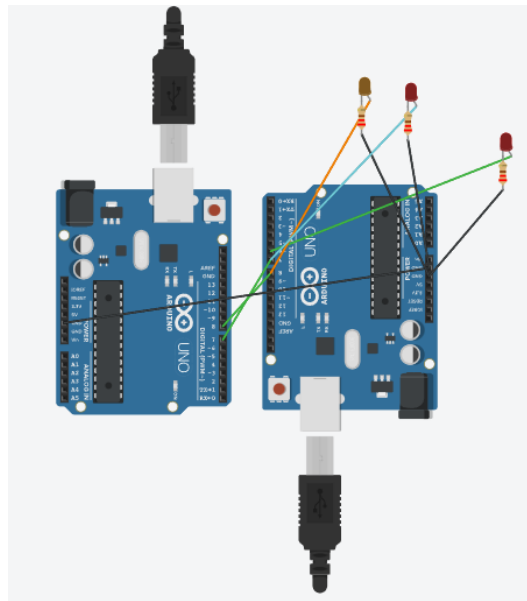


Figura 3: Interconexión de los Arduinos

Basados en la figura 2, se realizan los códigos en Arduino.

Las figuras 4, 5 y 6 muestran el código en el arduino de transmisión.

```
// Código de interconexión de los
//Arduinos para el Arduino 1.

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}

void loop()
{
  delay(500);
  digitalWrite(8, LOW);
  delay(250);
  digitalWrite(7, LOW);
  delay(250);
  digitalWrite(8, HIGH);
  delay(500);

  //SEGUNDO BIT
  digitalWrite(8, LOW);
  delay(250);
  digitalWrite(7, HIGH);
  delay(250);
  digitalWrite(8, HIGH);
```

Figura 4: Primera parte del código del Arduino transmisor.

```
//TERCER BIT
delay(500);
digitalWrite(8, LOW);
delay(250);
digitalWrite(7, HIGH);
delay(250);
digitalWrite(8, HIGH);

//CUARTO BIT
delay(500);
digitalWrite(8, LOW);
delay(250);
digitalWrite(7, HIGH);
delay(250);
digitalWrite(8, HIGH);

//QUINTO BIT
delay(500);
digitalWrite(8, LOW);
delay(250);
digitalWrite(7, LOW);
delay(250);
digitalWrite(8, HIGH);
```

Figura 5: Segunda parte del código del Arduino transmisor.


```

        //SEXTO BIT
        delay(500);
        digitalWrite(8, LOW);
        delay(250);
        digitalWrite(7, LOW);
        delay(250);
        digitalWrite(8, HIGH);

        //SEPTIMO BIT
        delay(500);
        digitalWrite(8, LOW);
        delay(250);
        digitalWrite(7, HIGH);
        delay(250);
        digitalWrite(8, HIGH);

        //OCTAVO BIT
        delay(500);
        digitalWrite(8, LOW);
        delay(250);
        digitalWrite(7, LOW);
        delay(250);
        digitalWrite(8, HIGH);
    }

```

Figura 6: Tercera parte del código del Arduino transmisor.

La figura 7 muestra el código en el arduino de recepción.

```

// Código de interconexión de
// los Arduinos para el Arduino 2.
void setup()
{
  pinMode(6, OUTPUT);
  pinMode(7, INPUT);
  pinMode(8, INPUT);
}

void loop()
{
  bool a;
  if(digitalRead(8)==HIGH){
    a= digitalRead(7);
  }
  digitalWrite(6,a);

  delay(550);
}

```

Figura 7: código del Arduino receptor.

4.3. Análisis del parcial

Primero se crearía el código con el cual generaremos los datos que saldrán desde el PC1 hacia el primer Arduino, el cual transformará los datos recibidos en binario. Luego creamos la conexión entre ambos arduinos en tinkercad y de manera simultánea conectamos el circuito integrado 74HC595 para paralelizar el binario. Después crearemos el código de cada arduino. Posteriormente haremos uso de las compuertas lógicas para verificar la bandera que recibirá el arduino 2; esto se hará en un código de C++, que luego será adaptado al arduino. Luego de conectar el PC2(Pantalla LCD), y completar el circuito, se implementará una serie de pruebas para verificar el correcto funcionamiento del sistema de comunicación.

5. DESARROLLO

Esta sección se presentará en el informe de implementación.

6. CONCLUSIONES

- La plataforma de Tinkercad, el microcontrolador Arduino, el lenguaje C++ y el circuito integrado 74HC575 son piezas claves tanto para la

realización de este proyecto como para la realización de otros proyectos prácticos.

- El microcontrolador Arduino es útil para el envío de información a otras plataformas para otros lenguajes de programación, y también a otros arduinos.
- El uso del circuito integrado es útil para el envío paralelo de los datos.

7. REFERENCIAS

Referencias

- [1] ELECTROALL. Funcionamiento y aplicación del circuito integrado 74hc595—pcb fabricado por jlcpb. [Online]. Available: <https://www.youtube.com/watch?v=G1c9xtP9FDY>