

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
5. DESARROLLO	5
6. CONCLUSIONES	5

1. INTRODUCCIÓN

2. RESÚMEN

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 descriptación). La paralelización de los datos se realiza por medio del circuito integrado 74HC595.
4. Sistema de descriptación: 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 descriptación. 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 descricptación.

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.

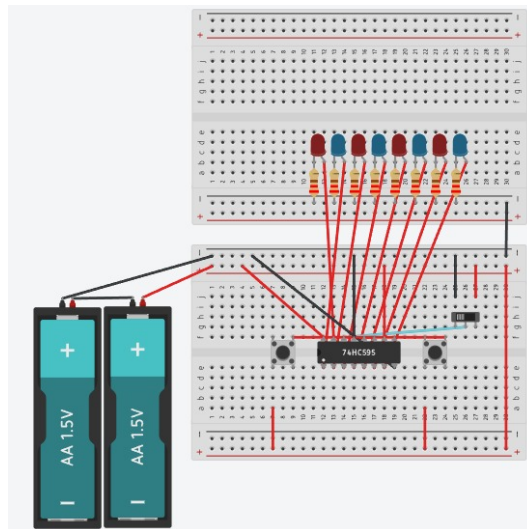


Figura 1: Circuito independiente 74HC595.

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
- 1111101010100101

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.

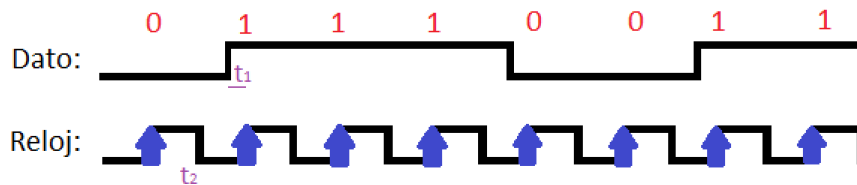


Figura 2: Señales para la primera prueba de la señal de dato

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.

La figura 3 muestra la interconexión entre los arduinos de transmisión (arduino 1) y de recepción (arduino 2).

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.

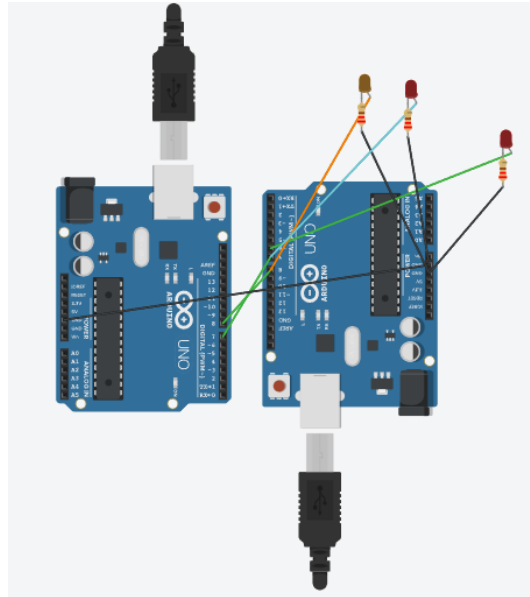


Figura 3: Interconexión de los Arduinos

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

5. DESARROLLO

Esta sección se presentará en el informe de implementación.

6. CONCLUSIONES

```
// 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.

```

// 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.