

Informe de Análisis y Diseño Parcial 1

**YESIKA MILENA CARVAJAL DÍAZ
NICOLL CAROLINE CHAZATAR
ANDRÉS FELIPE ZULUAGA ORTIZ**

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	3
4.1. El proyecto se divide en 6 bloques:	3
4.2. Análisis del parcial	3
5. DESARROLLO	4
5.1. Circuito integrado 74HC595	4
5.2. Comunicación entre Arduinos	5
5.3. Sistema de desenscriptación.	10
5.4. Sistema completo	12
5.4.1. Arduino transmisor	12
5.4.2. Arduino transmisor y Circuito 74HC595	16
5.4.3. Integrado 74HC595 y circuito de desenscriptación	16
5.4.4. Arduino receptor	18
6. CONCLUSIONES	22
7. REFERENCIAS	22

1. INTRODUCCIÓN

En este documento se presenta el informe completo y detallado del sistema de transmisión y recepción de datos del parcial 1 del curso de Informática II. Este documento es la implementación al proyecto parcial al cual se le hizo el análisis anteriormente.

Se realiza un sistema de transmisión y recepción de datos utilizando un circuito integrado 74HC595 para la paralelización de datos mandados previamente desde el arduino transmisor utilizando una señal de reloj para la recolección de estos; se utilizan circuitos integrados de compuertas lógicas inversoras y AND para la verificación de la clave, por tanto la generación de la señal de bandera que se ingresa al arduino receptor.

En el arduino transmisor se generan la señal de los datos, y dos señales para la recolección adecuada de los datos, una que da razón del tiempo por bit y la otra del tiempo por Byte, y en el circuito receptor se almacena el dato solicitado, en este caso es el mayor dato de los cinco números anteriores a la clave.

2. RESÚMEN

El informe busca obtener herramientas y conocimiento que ayuden a realizar el proyecto del curso de Informática II. Se analiza a profundidad el sistema de comunicación completo, logrando establecer la transmisión de los datos entre transmisor y receptor mediante la señal de reloj, haciendo uso del lenguaje en C++, Arduino, la plataforma Tinkercad y circuitos integrado para ello.

3. OBJETIVOS

3.1. GENERAL

Adquirir conocimientos de sistemas de transmisión por medio de sistemas integrados.

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

3.2. ESPECÍFICOS

- Desarrollar la capacidad de solución de problemas.
- Adquirir conocimientos en la utilización de Arduino.
- Entender sobre la interconexión entre Arduinos y entre arduinos con circuitos integrados.
- Manejar adecuadamente la plataforma de Tinkercad.

- Entender el uso del circuito integrado 74HC595.
- Comprender la utilización y la utilidad de las compuertas lógicas y de sus circuitos integrados.

4. METODOLOGÍA

4.1. 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.

4.2. Análisis del parcial

Primero se crea el código con el cual generaremos los datos que saldrán desde el PC1 hacia el primer Arduino, El arduino de transmisión recibe los datos y los transforma a binario.

Luego se crea la conexión entre ambos arduinos en Tinkercad y de manera simultánea se conecta el circuito integrado 74HC595 para paralelizar el binario. Después se crea el código de cada arduino. Posteriormente se hace uso de las compuertas lógicas para verificar la bandera que recibe el arduino 2; esto se hace en el código de arduino.

Luego de conectar el PC2 (Pantalla LCD), y completar el circuito, se implementa una serie de pruebas para verificar el correcto funcionamiento del sistema de comunicación.

5. DESARROLLO

Inicialmente para del desarrollo del proyecto se realiza prototipos de cada sección del sistema para posteriormente unir estas secciones y lograr resolver el proyecto de manera mas analítica.

5.1. Circuito integrado 74HC595

El circuito integrado es útil para 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 desencriptación.

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.

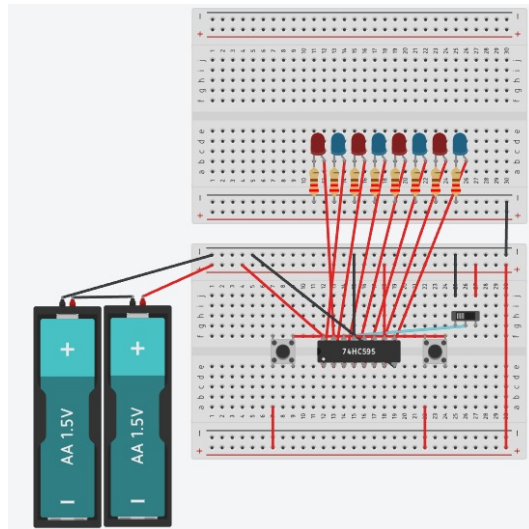


Figura 1: Circuito independiente 74HC595.

[2]

5.2. 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 t2 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.

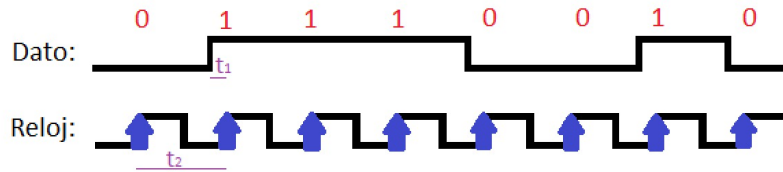


Figura 2: Señales para la primera prueba de la señal de dato

Para el t_2 de la prueba, su valor es de 250 milisegundo. Se utilizan 125 milisegundos para el valor en alta y 125 milisegundos en baja. Para t_1 se utiliza un valor de 25 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.

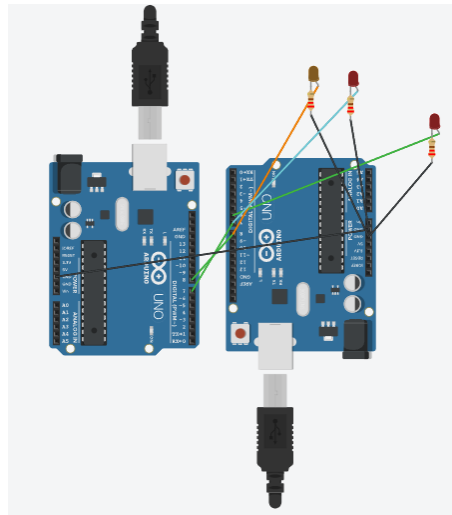


Figura 3: Interconexión de los Arduinos

[3]

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

```
// 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.

[3]


```
//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.

[3]

```

    //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.

[3]

```

// 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.

[3]

5.3. Sistema de descriptación.

Para el sistema de descriptación se utiliza un circuito integrado 74HC04 y dos 74HC08, los cuales corresponden a un circuito integrado de compuertas inversoras y de compuertas AND respectivamente.

Se usan 3 compuertas inversoras y 7 compuertas AND, siguiendo el esquema de la figura 8. Se utilizaron compuertas AND debido a que estas son compuertas que sólo obtienen un '1' para un caso, por lo que para otros casos en que no sean el solicitado arrojan '0'.

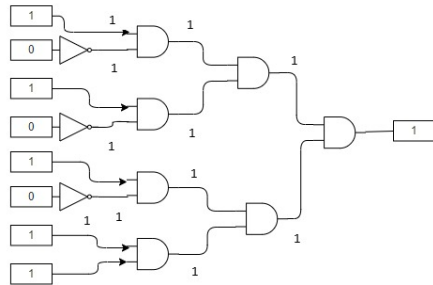


Figura 8: Compuertas del sistema de descriptación.

El esquema de arduino para la implementación del esquema de la figura anterior junto con las compuertas lógicas es el de la figura 9, y el código que se

implementó para observar el buen funcionamiento del circuito es el de la figura 10, en el cual se mandan los bits correspondientes a la clave individual del grupo, el número 171, el cual en bits corresponde al 10101011.

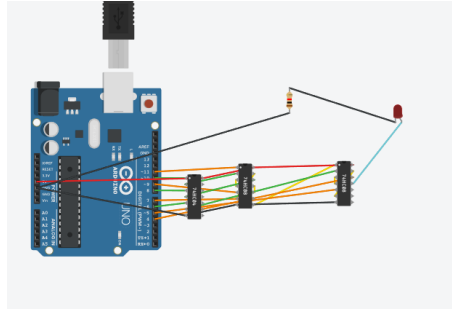


Figura 9: Sistema de descryptación mediante compuertas lógicas.

[4]

[4]

```

// C++ code
//
void setup()
{
  pinMode(4, OUTPUT);    //1
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);    //5
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);   //8
}

void loop()
{
  digitalWrite(11, HIGH);    //bit 7
  digitalWrite(10, LOW);
  digitalWrite(9, HIGH);
  digitalWrite(8, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(4, HIGH);    //bit 0
}

```

Figura 10: Código de prueba para la bandera del sistema de descriptación.

5.4. Sistema completo

5.4.1. Arduino transmisor

Para el arduino transmisor, se utilizaron 3 salidas, correspondientes a la señal de datos (pin 8), señal de reloj (pin 7) y una señal que se activa antes del noveno flanco de subida de la señal de reloj y se desactiva 25 milisegundos después (pin 6).

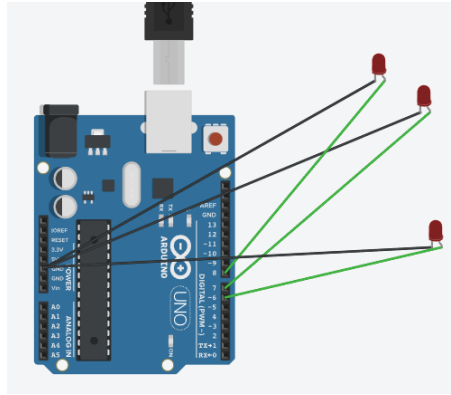


Figura 11: Circuito con Arduino configurado como transmisor.

[5]

La señal del pin 8 es la señal que entrega los datos en bits de los valores del arreglo entregado. La señal del pin 7 (señal de reloj) controla el almacenamiento de cada bit de la señal de datos, y la señal del pin 6 controla la señal que se ingresa al circuito integrado 74HC595 que da cuenta de cuándo se deben enviar los bits almacenados a la salida del integrado en forma paralela. La realización de estas señales se observa en el código para el circuito de transmisión en las figuras 12 y 13. Se tiene en cuenta que la señal de datos debe tomar el valor deseado antes del flanco de subida del reloj, para esta se almacene adecuadamente, y que la señal del pin 6 debe generarse antes del noveno flanco de subida de la señal del reloj, para mostrar la cadena de 8 bits antes de recoger el noveno dato.

```

2 void setup()
3 {
4     Serial.begin(9600);
5     pinMode(8, OUTPUT);
6     pinMode(7, OUTPUT);
7     pinMode(6, OUTPUT); //Pin para la señal que manda los datos
8     pinMode(LED_BUILTIN, OUTPUT);
9 }
10
11 void loop(){
12     int datos []={2, 43, 171, 209, 53, 204, 171, 87, 90, 80, 60};
13
14     for(long long unsigned i=0; i<11;i++){ //Cambiar la longitud
15
16         int entero=datos[i];
17         String binario;
18         binario=int2bin(entero);
19         Serial.println(entero);
20         digitalWrite(6, HIGH); //Pin para la señal que manda lo
21     }
22 }
23
24 String int2bin(int entero)
25 {
26     String binario="AAAAAAAA";
27     int f=0;
28     for(int i=7; i>=0; i--){
29         if(entero>=pow(2, i)){
30             binario[f]='1';
31             digitalWrite(8, HIGH);
32             f++;
33             entero=entero-pow(2,i);

```

Figura 12: Primera parte del código del Arduino transmisor.

[5]

```

24 String int2bin(int entero)
25 {
26     String binario="AAAAAAAA";
27     int f=0;
28     for(int i=7; i>=0; i--){
29         if(entero>=pow(2, i)){
30             binario[f]='1';
31             digitalWrite(8, HIGH);
32             f++;
33             entero=entero-pow(2,i);
34         }
35         else{
36             binario[f]='0';
37             digitalWrite(8, LOW);
38             f++;
39         }
40
41         delay(25);
42         digitalWrite(7, HIGH); //señal del reloj
43         digitalWrite(6, LOW); //Bajamos la señal que lleva los 8
44         delay(125);
45         digitalWrite(7, LOW);
46         delay(125);
47     }
48     return binario;
49 }
50

```

Figura 13: Segunda parte del código del Arduino transmisor.

Los datos importantes del código del arduino transmisor son la función `int2bin` (figura 13), la cual recoge el entero y lo transforma en string por medio de un `for`, el cual evalúa si su valor es mayor o igual a 2 a la séptima potencia, de ser así, se obtendrá un bit '1' en el bit más significativo. Así mismo se realizan con los demás bits. Además, en esta misma función se mandará por medio del puerto 8 el bit correspondiente a '1' o a '0' según es el caso. Al final de cada iteración del `for`, por tanto de cada envío de señal del dato en el puerto 8, se hace un `delay` de 25 milisegundos para generar la señal de subida del reloj. Además, al salir de la función (figura 12) se realiza el envío de la señal del puerto 6, para dar cuenta de que ya se cumplieron los envíos de los 8 bits.

Por medio de las 3 señales antes mencionadas y de la forma antes explicada se realiza una toma adecuada de los datos para el funcionamiento adecuado del circuito integrado 74HC595 y del arduino receptor.

5.4.2. Arduino transmisor y Circuito 74HC595

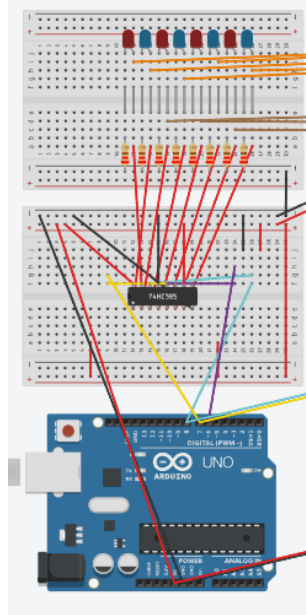


Figura 14: Conexión entre el Arduino transmisor y el circuito integrado 74HC595.

[6]

En la interconexión del arduino con el circuito integrado 74HC595, se interconectar el pin 8, 7 y 6 del arduino con los pines 14, 11 y 12 del 74HC595 respectivamente.

Se agregan LEDs para observar la serie de bits que arroja el integrado. Estos bits están ordenados por orden de peso de bit.

5.4.3. Integrado 74HC595 y circuito de descryptación

Para realizar el circuito de descryptación, se sigue el esquema de la figura 15. Se modifica este esquema respecto al numeral 1 en que se cambia un circuito de 4 AND de 2 entras (74HC08) por un circuito de 2 AND de 4 entradas (74HC21).

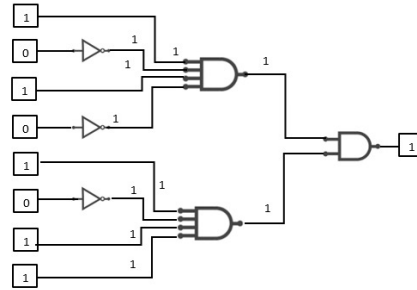


Figura 15: Compuertas lógicas para el sistema completo.

[6]

La conexión del circuito integrado 74HC595 y del sistema de descriptación se observa en la figura 16, en donde se observa con cables cafés que los bits 1, 3 y 5 (derecha a izquierda) se dirigen al integrado correspondiente al de las compuertas lógicas inversoras. Esto es lo observado en el esquema de la figura anterior.

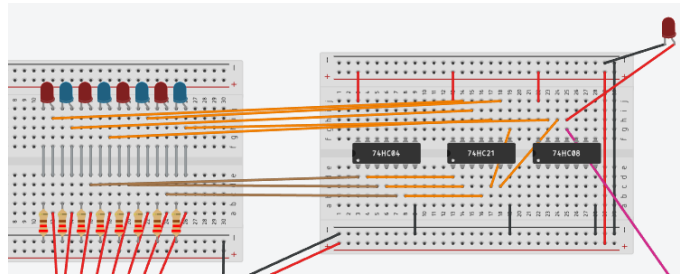


Figura 16: Interconexión del sistema de paralelización y del sistema de descriptación.

[6]

Se agregó un LED a la señal de salida del esquema para comprobar el valor de la bandera. Esto se observa en la figura 16.

5.4.4. Arduino receptor

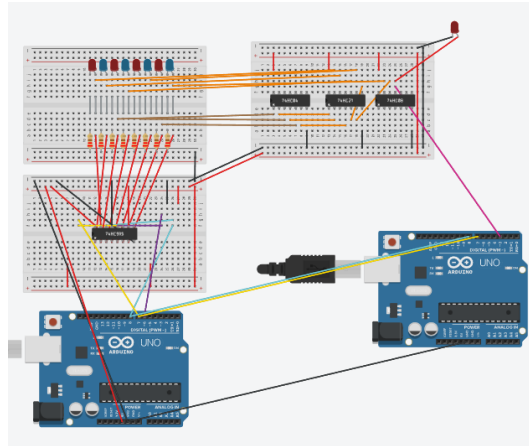


Figura 17: Sistema completo.

[6]

El Arduino receptor es el encargado de realizar el proceso en el que se identifica cual es el número mayor entre los últimos 5 números antes de que aparezca la bandera, es por ello por lo que el código del Arduino receptor es el observado en las figuras 18, 19 y 20:

```

2   String uno="AAAAAAA";
3   String dos="AAAAAAA";
4   String tres="AAAAAAA";
5   String cuatro="AAAAAAA";
6   String cinco="AAAAAAA";
7   String seis="AAAAAAA"; //String de más que guarda la clave
8   String siete="AAAAAAA";
9
10  int uno_ent=0;
11  int dos_ent=0;
12  int tres_ent=0;
13  int cuatro_ent=0;
14  int cinco_ent=0;
15
16  int mayor=0;
17  //
18  void setup()
19  {
20    Serial.begin(9600);
21    pinMode(3, INPUT); //Señal de bandera
22    pinMode(7, INPUT); //Señal de reloj
23    pinMode(8, INPUT); //Señal de datos
24  }
25
26  void loop()
27  {
28
29    if(digitalRead(3)){
30      //Cambie los String uno-cinco por enteros
31      uno_ent= str2int(uno);
32      dos_ent=str2int(dos);

```

Figura 18: Primera parte del código del Arduino receptor.

[6]

```

26 void loop()
27 {
28     if(digitalRead(3)){
29         //Cambie los String uno-cinco por enteros
30         uno_ent= str2int(uno);
31         dos_ent=str2int(dos);
32         tres_ent=str2int(tres);
33         cuatro_ent=str2int(cuatro);
34         cinco_ent=str2int(cinco);
35
36         //Compare los enteros uno-cinco para saber cuál es mayor
37         mayor=uno_ent;
38         if(dos_ent>mayor)      mayor=dos_ent;
39         if(tres_ent>mayor)     mayor=tres_ent;
40         if(cuatro_ent>mayor)   mayor=cuatro_ent;
41         if(cinco_ent>mayor)    mayor=cinco_ent;
42
43         Serial.println(mayor);
44     }
45
46     uno=dos;
47     dos=tres;
48     tres=cuatro;
49     cuatro=cinco;
50     cinco=seis;
51     seis=siete;
52
53     int j=0;
54     while(j<8){
55         if(digitalRead(7)){ //Si el reloj es 1.
56             if(digitalRead(8)){ //Si el dato es 1.
57                 siete[j]='1';
58             }

```

Figura 19: Segunda parte del código del Arduino receptor.

[6]

```

53   int j=0;
54   while(j<8){
55       if(digitalRead(7)){           //Si el reloj es 1.
56           if(digitalRead(8)){       //Si el dato es 1.
57               siete[j]='1';
58           }
59       } else{                       //Si el dato es 0
60           siete[j]='0';
61       }
62       j++;
63       delay(137);                  //Para asegurar que en la próxima lectur
64   }
65 }
66
67
68 int str2int(String binario){
69     int entero=0;
70     int suma=128;
71     if(binario[0]=='A'){
72         entero=0;
73     }
74     else{
75         for(int i=0; i<8; i++){
76             if(binario[i]=='1'){
77                 entero=entero+suma;
78             }
79             suma=suma/2;
80         }
81     }
82     return entero;
83 }
84

```

Figura 20: Tercera parte del código del Arduino receptor.

[6]

La estructura del circuito receptor es la siguiente:

Inicialmente hay siete variables tipo string almacenadas en cada una de ellas la letra 'A', 8 veces, con el fin de capturar los 8 bits de entrada de cada dato; es decir, las primeras variables se crean para almacenar 5 datos antes de que aparezca la bandera, el string seis se crea para almacenar la clave, y el string siete almacena el dato posterior a la clave. El dato, como se mencionó antes, es 171.

En la función *setup* en el código de Arduino se inicializa la comunicación serial, se define los pines 3, 7, 8 como entrada que corresponde a la señal de bandera, de reloj y de datos respectivamente. En la función *loop* se observa si el bit de la bandera está en '1'. De ser así, se cambia el tipo de dato de los datos de string a entero (de las variables de uno a siete a las de uno-ent a siete-ent) con la función *str2int*, y posteriormente se hacen una serie de if para tomar el valor de la variable mayor. En la función de *str2int* se toma en cuenta que los strings que no tengan bits sino un 'A' toma el valor de entero de 0, para que al comparar con el valor más grande este no se tome en cuenta.

Fuera de la condición de que la bandera sea '1', se realiza la toma de datos. Para esto, primero se trasladan las variables, para actualizar el dato más actual y desechar el dato más antiguo. Mediante un ciclo *while* se establece la simulación

del reloj y se toma los datos en cada flanco de subida, generando un delay de 550 milisegundos para dar el espacio de tiempo pertinente para que el reloj vuelva al valor de bajo y así almacenar en la variable siete el dato de entrada; esto se realiza comparando si lo que llega es un '1' o '0', y este valor se agrega en la posición pertinente del string siete.

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 y recepción de información a diferentes sistemas y también a otros arduinos.
- En un sistema de transmisión y recepción de datos, es importante tener una señal de reloj que de cuenta del dato presente.
- En el sistema de comunicación, los datos generados deben producirse antes del flanco de referencia del reloj, para asegurar que estos datos sean los correctos, ya que si se genera al mismo tiempo del flanco del reloj, puede tomar el dato anterior y no el actual.
- La señal del pin 6 se debe generar antes del noveno flanco de subida de la señal del reloj, para mostrar la cadena de 8 bits antes de que el reloj recoga el dato del noveno bit.
- El uso del circuito integrado 74HC595 es útil para el envío paralelo de los datos en sus 8 puertos de salida.
- El arduino se programa en lenguaje de C++, siempre y cuando se respeten funciones principales setup y loop del arduino.
- Con las compuertas lógicas se pueden crear sistemas para el reconocimiento de bits especificados.

7. REFERENCIAS

Referencias

- [1] ELECTROALL. Funcionamiento y aplicación del circuito integrado 74hc595—pcb fabricado por jlpcb. [Online]. Available: <https://www.youtube.com/watch?v=G1c9xtP9FDY>
- [2] yesika.carvajal. Prueba con botones del 74hc595. [Online]. Available: <https://www.tinkercad.com/things/1V5SJiUvgR8-surprising-fyyran-leelo/editel?tenant=circuits>

- [3] ——. Interconexión entre arduinos. [Online]. Available: <https://www.tinkercad.com/things/byA928yFaPF-stunning-turing-densor/editel?tenant=circuits>
- [4] ——. Compuertas lógicas. [Online]. Available: <https://www.tinkercad.com/things/5nJrKYKeH00-spectacular-juttuli>
- [5] ——. Circuito transmisor. [Online]. Available: <https://www.tinkercad.com/things/9q8DKDR45Gn-ingenuous-lahdi>
- [6] ——. Sistema de comunicación. [Online]. Available: <https://www.tinkercad.com/things/6gT6oBgz0bW-copy-of-copy-of-circuito-completo/editel?tenant=circuits>