

# **Elaboración de Informe del parcial 1**

**Angel Graciano Espitia  
Juan Diego Castro**

**Informatica 2  
Grupo Team  
Profesores: Augusto Enrique Salazar - Jonathan Ferney  
Gomez  
21 de febrero de 2022**

# Índice

<b>1. Abstract</b>	<b>2</b>
<b>2. Introducción</b>	<b>2</b>
<b>3. Marco Teórico</b>	<b>2</b>
<b>4. Análisis del Problema</b>	<b>2</b>
4.1. Transmisión . . . . .	2
4.2. Paralelización y Desencriptación . . . . .	2
4.3. Recepción . . . . .	3
4.4. Visualización . . . . .	3

## 1. Abstract

## 2. Introducción

## 3. Marco Teórico

## 4. Análisis del Problema

### 4.1. Transmisión

En la parte de transmisión tenemos que hay un Arduino emisor que va a enviar datos al Arduino receptor, también va a enviar datos al circuito 74hc595 que se encargara de capturar la trama de bits. Para el emisor tenemos 3 salidas. Una señal digital serial, una señal de reloj y el latch el cual va a captar ese byte para el integrado de registro de desplazamiento. A su vez, a señal serial y la de reloj irán a un pin digital de por cada señal en al Arduino receptor. Para obtener una señal digital de los datos que queremos hacemos una conversión de entero a binario, este para eso utilizamos en lenguaje de C++ que nos permite utilizar el Arduino. Gracias a la ayuda del operador lógico “” para poder realizar a lo que se le llama mascarar, así obtenemos el bit menos significativo que nos permitirá poner en un arreglo binario, después los ponemos a iterar para sacarlo del Arduino, así con cada uno de los elementos de los arreglos char. También es importante decir que después de realizar esta acción, hicimos la de la señal de reloj, para recibir un bit de la señal, y después que terminara con el arreglo de binarios, se emitió el latch para el 74hc595.

### 4.2. Paralelización y Desencriptación

Después de que el integrado de registro de desplazamiento paralelizaba los datos que le llegaban desde el pin serial del Arduino emisor, se tenía que hacer la comparativa

para la desencriptar el número, para ello se tenía utilizamos otro 74hc595, pero en este caso para en el Arduino receptor y para paralelizar lo que sería la bandera, en este caso se aplica el mismo código, pero solo para un número. Ya una vez paralelizado la bandera y el número, a través de 2 circuitos integrados 74hc86 fuimos comparando bit a bit para saber si eran iguales, como este circuito integrado consta de 4 compuertas XOR esperábamos cuatro salidas, que en caso de ser diferentes números esperábamos un 1 en la salida y 0 en el caso contrario. Recordemos que utilizamos 2 de estas, porque son 8 bits, ya las salidas de estas los comparamos con un circuito integrado 75hc32 que son compuertas OR, y lo que esperábamos que no hubiera un uno en alguna salida, sino sabríamos que ya hay algún bit diferente a otro, por ende, los números serian diferentes. El proceso de las OR se fue repitiendo hasta que solo hubiera una salida, que era la que determinaba si los números eran iguales o diferentes, en caso en el que marcara 1 los números serian diferentes, caso contrario si la salida marcara 0.

### **4.3. Recepción**

Recordemos que en el Arduino receptor recibe el reloj y los datos del primer Arduino, además también recibe la señal que va a indicar si los números son iguales. Lo que hace el programa del Arduino receptor es interpretar de la manera mas correcta posible los datos del primer Arduino y los convierte a char nuevamente gracias a la conversión de binario a entero, estos datos char va cambiando mediante se ingrese un nuevo dato al sistema. Ya después se imprime en pantalla el dato descifrado.

### **4.4. Visualización**

Esta parte fue una de las mayores dificultades de nuestro programa, ya que encontrábamos muchos errores de sincronización, errores de la función pow que se utiliza para las potencias. Requiere tiempo para hallar este último error, que es estaba incluso dañando la salida en el monitor serial. Al final Tratamos lo más posible para que esos errores no sucedieran. Para realizar este paso tuvimos que buscar como se conecta un LCD en Arduino, también se utilizó la librería de `¡LiquidCrystal!`, y se configuro los pines del Arduino en la función LiquidCrystal.