

# Informe de Implementación

**Andrés Salazar Hoyos**  
**Santiago Arcila Gómez**  
**Jeisson Arley Alvarez Giraldo**

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

## Contents

|          |                       |          |
|----------|-----------------------|----------|
| <b>1</b> | <b>Solución</b>       | <b>2</b> |
| <b>2</b> | <b>Inconvenientes</b> | <b>2</b> |
| <b>3</b> | <b>Experiencia</b>    | <b>3</b> |
| <b>4</b> | <b>Pruebas</b>        | <b>3</b> |
| <b>5</b> | <b>Referencias</b>    | <b>4</b> |

# 1 Solución

## 2 Inconvenientes

A la hora de resolver el problema se presentaron diferentes inconvenientes pero se lograron solucionar todos gracias a una buena investigación.

Mientras se realizaba la implementación en tinkercad del ejercicio, llegamos a un problema con la cantidad de los puertos del arduino, ya que no eran suficientes para las conexiones de la pantalla lcd y los integrados, pero se pudo solucionar conectando otro integrado adicional, con las señales del segundo integrado (el encargado de recibir la información), y de esta manera disponer de más puertos para controlar la pantalla lcd. A su vez para que funcionara con estas modificaciones tocó consultar e implementar unas funciones de código para reestructurar los pines del integrado y mandar la información a la pantalla.

El equipo encargado de recibir la información lo hace mediante la función "pow" de arduino, pero esta nos estaba generando un problema, ya que al momento de entregar el número lo entregaba en decimal y un valor aproximado, por ejemplo, si debía entregar un 15 entregaba 14.999 que es un número muy cercano pero que al convertirlo a entero no era el número que se necesitaba. Se llegó a la conclusión de que el problema era la función de arduino y para solucionarla se creó una mediante código desde cero.

En un inicio del proyecto habíamos tomado la decisión de que por eficiencia, el circuito integrado que se encargaba de enviar la información, se encargaría a su vez de realizar el trabajo de enviar solo la información correcta, es decir, el mismo integrado se encargaba de descifrar la clave y enviar los datos válidos al receptor; Después de una asesoría con el profesor, se tuvo en cuenta que al realizar el proyecto de esta forma estábamos enviando los datos sin ningún tipo de protección. La solución que se nos ocurrió fue hacer que el sistema encargado de recibir la información fuera a su vez el que separara los datos correctos de los que no.

El inconveniente más difícil que se presentó durante la solución del problema planteado, fue hacer un correcto uso del reloj tanto para enviar como para recibir la información. En un principio utilizamos los mismos tiempos en la programación, para enviar y recibir los datos, pero al momento de mostrarlos en la pantalla LCD por un tiempo más prolongado, debíamos dejar en espera al equipo que se encargaba de recibir la información, mientras que el equipo encargado de enviarla no se detenía, por lo cual había una pérdida de paquetes debido a que el equipo encargado de recibir se encontraba en otro proceso; La forma en que se solucionó este inconveniente fue asignándole tiempos de ejecución diferentes a ambos equipos que coincidieran con los tiempos requeridos para todo el proceso, a su vez, encontramos una forma de poner en espera al

sistema encargado de enviar la informacion, siempre y cuando el sistema encargado de recibirla tambien lo estuviera.

Durante la fase de programación de los circuitos integrados y realización de pruebas de funcionamiento, se llegó a un punto muerto, en el cual el sistema estaba enviando información totalmente distinta a lo que se tenía programado, se revisó todo el código y se comparó con los implementados en Qt, no se encontraba el error del porqué se recibían datos erroneos, tambien se revisaron las conexiones del arduino, y todo se encontraba en orden; Después de un par de horas en este punto muerto se eliminó el integrado de la protoboard y se puso uno nuevo sin modificar nada más y el problema se soluciono, al hacer esto se llegó a la conclusión de que posiblemente se trataba de un bug de la plataforma de Tinkercad.

### 3 Experiencia

Este parcial supuso un desafio bastante exigente, no solo por el problema a resolver, sino tambien por el tiempo dado para ello. Entre el afán de resolverlo rápido y la inexperiencia de trabajo con este data-shift surgieron varios problemas e inconvenientes que con una investigación adecuada lograron ser resueltos. Fue un proyecto que definitivamente requirió dedicación exhaustiva, pero que fue bastante interesante, y su proceso de resolución fue placentero

### 4 Pruebas

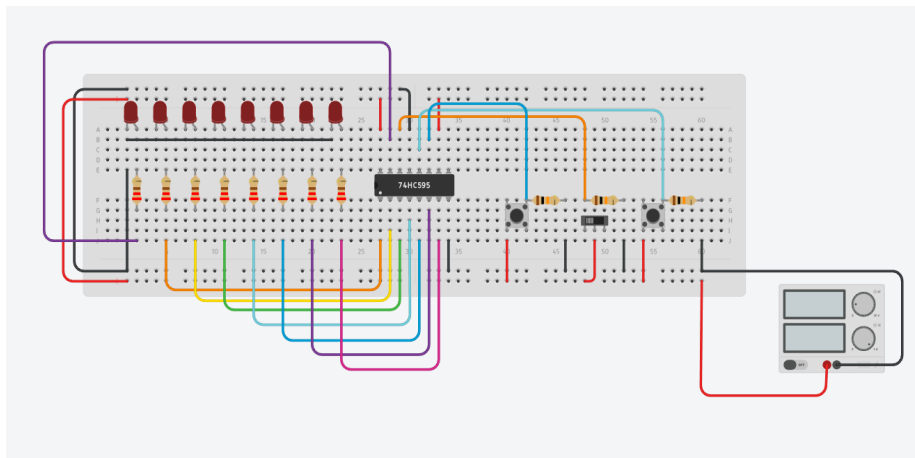


Figura 1: Integrado con pulsadores y switch

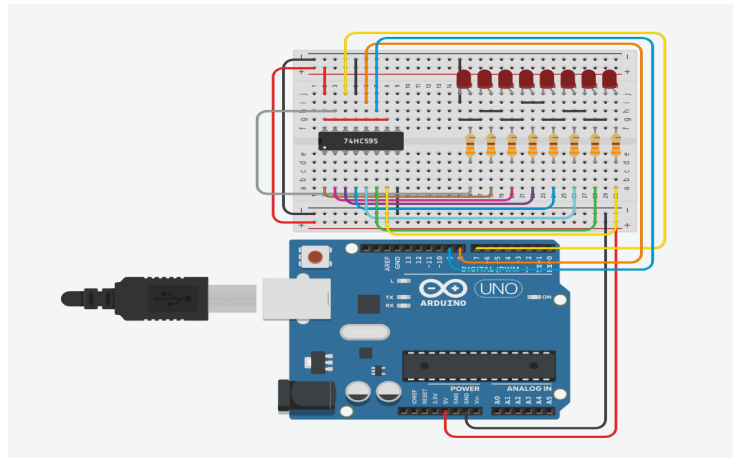


Figura 2: Integrado con arduino

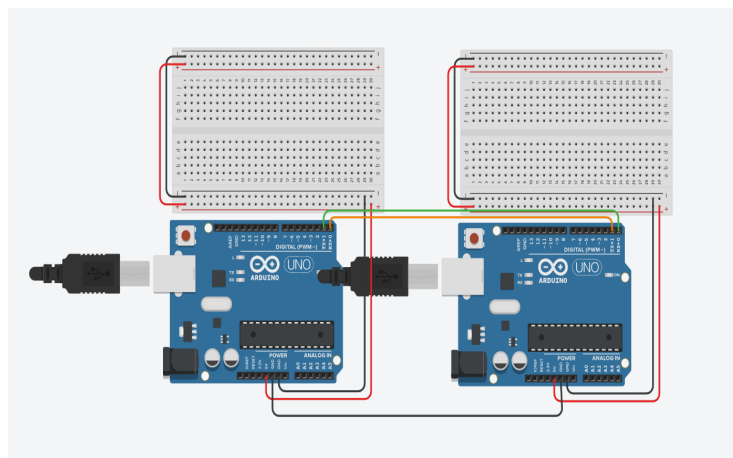


Figura 3: Comunicación entre arduinos

## 5 Referencias

<https://youtu.be/-AtKQSwgnSM>

<https://youtu.be/LFqIA3ZvZE8>

<https://github.com/matmunk/LiquidCrystal.74HC595/blob/master/LICENSE>

<https://www.ti.com/product/SN74HC595>

<https://iot-guider.com/arduino/serial-communication-between-two-arduino-boards/>