

# **Desafio 1**

Informa2 S.A.S

**Victor Manuel Jimenez Garcia**  
**Jose Miguel Jaramillo Sanchez**  
**Sebastian Garcia Morales**

Departamento de Ingeniería Electrónica y  
Telecomunicaciones  
Universidad de Antioquia  
Medellín  
Febrero 17 de 2022

# Índice

<b>1. Objetivos</b>	<b>2</b>
<b>2. Introduccion</b>	<b>2</b>
<b>3. Marco Teorico</b>	<b>2</b>
3.1. Conocimientos previos . . . . .	2
<b>4. Analisis del problema</b>	<b>5</b>
<b>5. Conclusiones</b>	<b>5</b>

## 1. Objetivos

- Aplicar los conocimientos adquiridos a lo largo del curso, demostrando apropiación de los fundamentos básicos del lenguaje de programación C++.
- Desarrollar habilidades de investigación y redacción que permitan la adquisición de nuevos conocimientos con el fin de solucionar problemas de la vida real.
- Demostrar la importancia y utilidad de la programación por hardware, así como el uso de módulos físicos para optimizar el uso de software en un diseño
- Diseñar un aplicativo en la plataforma de Arduino integrando programación de C++ para solucionar un desafío propuesto.

## 2. Introduccion

## 3. Marco Teorico

### 3.1. Conocimientos previos

A la hora de enfrentarse a un desafío lo más recomendable es dividirlos en varias etapas para trabajarlo más fácilmente, una primera etapa sería realizar una investigación de conceptos y componentes propuestos en el desafío. En este caso es necesario investigar el concepto de transferir información de forma serial y paralela cómo también identificar características, funcionalidades arquitectura, conexiones, alcances y limitaciones del circuito integrado 75HC595, por otro lado, ¿qué es un Arduino? y ¿cómo unirlo al circuito integrado mencionado anteriormente para lograr solucionar el desafío completo?.

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla (principalmente con cables dupont). [1]

Este dispositivo es el que nos permitirá recibir los datos ingresados por el usuario y realizar la conversión a binario, además de funcionar tanto transmisor como receptor en el sistema de encriptación.

La comunicación entre arduinos se realizará de forma serial, que es el proceso de enviar datos de carácter binario un bit a la vez, esto provee la ventaja de mantener la interfaz transmisor-receptor de forma simple y eficiente. [2]  
Por lo tanto, para desencriptar, es necesario paralelizar dicha secuencia de bits

que luego serán las entradas de un circuito de lógica combinacional encargado de comparar los datos de acuerdo a los parámetros de descryptación.

Paralelizar no es más que llevar la secuencia de bits que se desplazan como una sola fila, y transformarla en una columna. De esta forma si se tiene una secuencia serial de  $n$  bits, al paralelizar, el resultado es una columna de bits de  $n$  filas.

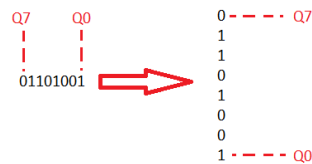


Figura 1: Ejemplo de paralelización

Esta acción de paralelizar la llevará a cabo el circuito integrado 74HC595 también conocido como Registro de desplazamiento. Este chip de 16 pines, recibe una secuencia de 8 bits en un solo pin, y los va almacenando en cada una de las salidas para luego ser liberados como 8 señales independientes.

Dos definiciones que se deben tener en cuenta para entender mejor el funcionamiento de todo el sistema son: comunicación síncrona y comunicación asíncrona.

**Comunicación síncrona:** Se da cuando el intercambio de mensajes sucede en tiempo real. Requiere que las dos partes (emisor y receptor) estén presentes en el mismo tiempo y espacio, ya sea físico o virtual. Por ejemplo, las llamadas telefónicas, las reuniones en la oficina o las videoconferencias.

**Comunicación asíncrona:** Sucede cuando los mensajes se intercambian sin importar el tiempo. Es decir, que no necesitan la atención inmediata del receptor, quien puede responder en el momento que decida o pueda hacerlo. Estamos hablando de medios como el correo electrónico, foros en línea, chats, mensajes de texto y documentos colaborativos. [3]

En este caso, la comunicación se da de forma síncrona con ayuda de un pulso de reloj. En electrónica y especialmente en circuitos digitales síncronos, una señal de reloj es una señal usada para coordinar las acciones de dos o más circuitos, esta señal oscila entre estado alto y bajo, también conocido como flanco de subida y de bajada, respectivamente, y gráficamente toma la forma de una onda cuadrada. [4]

#### Entradas:

**GND** (pin 8): conexión a tierra (0 V)

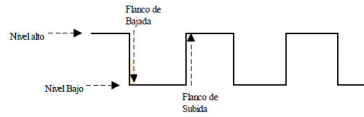


Figura 2: Diagrama de tiempo de una señal de reloj

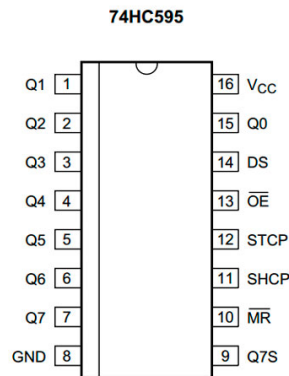


Figura 3: Pines IC 74HC595

**GND** (pin 10): reinicio del registro (activo bajo)  
**SHCP** (pin 11): señal de reloj  
**STCP** (pin 12): pulso para liberar los datos  
**GND** (pin 13): habilitar salida del registro (activo bajo)  
**DS** (pin 14): entrada de datos serial  
**VCC** (pin 16): conexion a fuente de voltaje (5 V)

#### Salidas:

**Q0-Q7** (pines 1-7 y 15 ): salida de datos  
**Q7S** (pin 9): salida de datos serial

El funcionamiento es el siguiente, la informacion serial entra por el **DS** (pin 14), el integrado recibe cada bit cuando ocurre un flanco de subida por el **SHCP** (pin 11) y lo almacena en la salida de mas baja valor **Q0** (pin 15), a medida que van entrando mas bits, los datos que habian almacenados anteriormente se van desplazando desde **Q0** hasta **Q7** hasta completar el byte. Una vez hecho esto, se manda un flanco de subida en **STCP** (pin 12), que se encargan de liberar los datos almacenados.

De esta forma el primer bit que entra, queda en la salida **Q7** y el ultimo en la salida **Q0**. Para ingresar un nuevo byte se debe borrar la informacion del registro, esto se hace mandando un flanco de bajada al pin **MR** (pin 10) y

luego activando la salida del registro (pin 12).

## **4. Analisis del problema**

## **5. Conclusiones**

## **Referencias**

- [1] Arduino.cl. ¿qué es arduino? [Online]. Available: <https://arduino.cl/que-es-arduino/>
- [2] O. Weis. Guía completa de especificaciones del puerto serie. [Online]. Available: <https://www.virtual-serial-port.org/es/article/what-is-serial-port/>
- [3] E. editorial de Indeed. Diferencias entre comunicación sincrónica y asincrónica. [Online]. Available: <https://mx.indeed.com/orientacion-profesional/desarrollo-profesional/diferencias-comunicacion-sincronica-asincronica>
- [4] injgonzalez. El concepto de señal de reloj. [Online]. Available: <https://sistemasdigitales2.files.wordpress.com/2011/11/semana3.pdf>