

2024

# Comunicación de Datos

COMUNICACIONES CABLEADAS

EQUIPO: KUHN LAUTARO, PALOMEQUE MATEO, CHIABO FRANCO

DOCENTES: MG. ING. MARTIN PICO, ING. MILTON POZZO

## Contenido

Comunicaciones Cableadas .....	2
Objetivos a cumplir .....	2
Comunicación en Paralelo .....	2
Implementación del Hardware Emisor .....	2
Implementación del Software Emisor .....	3
Implementación del Hardware Receptor .....	4
Implementación del Software Receptor .....	5
Comunicación en Serie .....	6
Implementación de Hardware Emisor .....	7
Implementación de Hardware Receptor .....	7
Implementación de Software Emisor y Receptor .....	7
Dificultades encontradas .....	8
Conclusión y agradecimiento .....	8
Bibliografía .....	8

## Comunicaciones Cableadas

### Objetivos a cumplir

El objetivo de este trabajo práctico es poder realizar conexiones en paralelo y en serie de, en este caso, dos placas Arduino de forma cableada. Estos nos permitirán obtener una comunicación de 8 bits para enviar datos desde un dispositivo al otro.

### Comunicación en Paralelo

#### Implementación del Hardware Emisor

Comenzamos con la comunicación en paralelo y para esta instancia se utilizó un Arduino Mega 2560 como el dispositivo 1, y un Arduino Nano como dispositivo 2. Este último se conecta a un protoboard con los demás componentes, tales como, los pulsadores, las resistencias y el led RGB de cátodo común.

La comunicación se da mediante cables que salen desde los pines digitales del dispositivo 1, los cuales serían desde el pin 2 hasta el 9 e inclusive el pin 10 para el reloj (clock), y se conectan de forma directa a los pines de entrada digital del dispositivo 2 que serían desde el pin 02 para el clock hasta el 08, el pin 012 y 013. También se usan los pines analógicos, PWM, 09, 010 y 011 asociados a los colores del led.

En el protoboard se colocan dos pulsadores, cada uno con una resistencia de 1kohm. Estas se conectan a los pines digitales 18 y 19 del dispositivo 1. Estos pines permiten las interrupciones externas por ese motivo se los asoció con los botones debido a la función que se les da. El pulsador en el pin 18 es el encargado de indicar, al dispositivo 2, el encendido o apagado del led RGB y el otro pulsador, en el pin 19, cumplirá la función de avisar al Arduino Nano el cambio de color del led. Ambas interrupciones son externas para que de esta forma se pueden ejecutar siempre que se presionen los botones, pero solo una interrupción a la vez y cuando termine el canal de comunicación quedará libre para recibir otra interrupción.

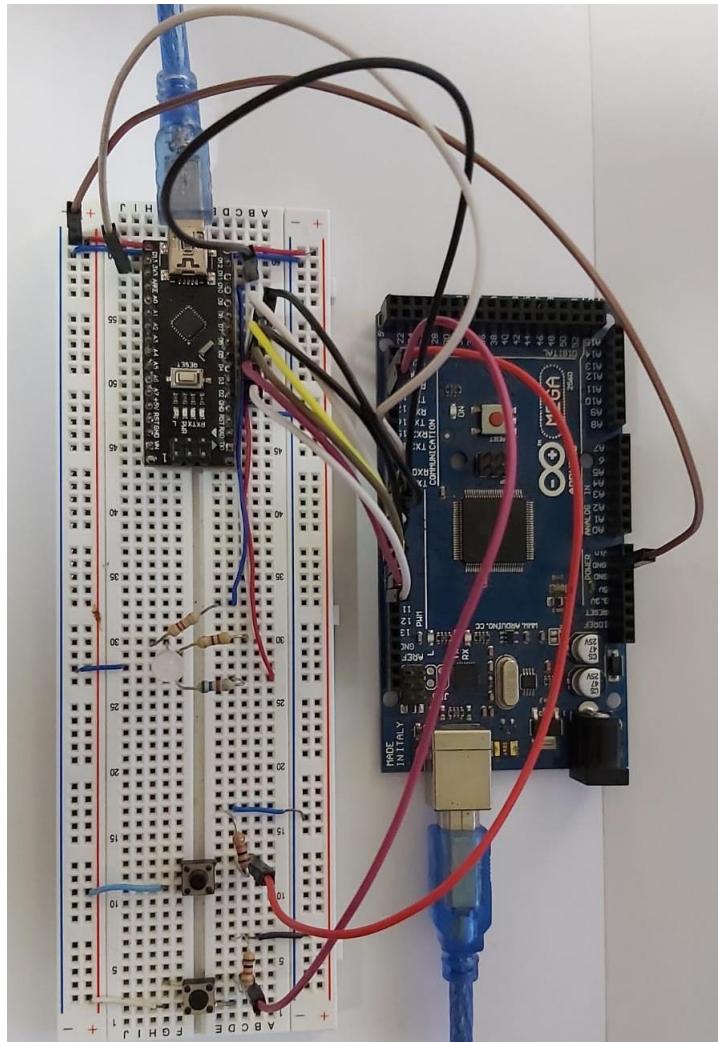


Figura 1. Conexiones del Emisor y Receptor

#### Implementación del Software Emisor

Para que esta comunicación pueda ser llevada a cabo, se necesita de un código para programar los dispositivos que intervienen. Se realizará un programa para cada dispositivo, el emisor y el receptor. En el primero (dispositivo 1) se tendrá poder enviar los tres colores aleatorios que conformarán el color final del led, esto mediante un botón que cada vez que se presione genera valores aleatorios distintos. También se configura un botón que enciende o apaga el led, que este encendido significa que los colores serán enviados al dispositivo 2 y apagado es que se enviará (0,0,0) como valores para que el led este apagado y no podrá recibir otros valores hasta que se vuelva a presionar el pulsador.

Se logró usando dos interrupciones externas, como se mencionó anteriormente, para que el programa en bucle solo continue cuando sea necesario o se lo permita. Estas interrupciones están ligadas a pulsadores conectados a pines digitales específicos que reconocen un cambio de los niveles de voltaje como flancos

ascendentes o flancos descendentes. Esto es útil para ahorrar tiempo y agilizar el funcionamiento del microcontrolador permitiendo una respuesta rápida y efectiva a eventos externos.

En la generación de valores aleatorios para el color del led, se utiliza el comando “random()” donde dentro del paréntesis de coloca el valor máximo que puede generar, en este caso 255 debido a que usamos 8 bits. Este procedimiento se realiza tres veces, uno para cada color del led, rojo (R), verde (G) y azul (B). Para que lo pueda recibir el dispositivo 2, se usó el comando “bitRead()”, esto nos permite pararle un valor en decimal y mediante un bucle convertirlo en binario y enviar un valor, 0 o 1, por cada puerto de salida del dispositivo. El procedimiento se hace para cada color, es decir, se efectúa tres veces, una tras otra y el dispositivo receptor los recibirá en binario y lo almacenará en decimal para pasarlo como los valores del led.

Para esta instancia se explicó el procedimiento y hardware del dispositivo emisor, Arduino Mega, su función se limita a enviar información y datos para que el dispositivo receptor los reciba y ejecute las acciones.

#### Implementación del Hardware Receptor

Como se mencionó antes el dispositivo 2, Arduino Nano, está conectado a conectado a una protoboard y mediante unos cables desde sus pines digitales, mencionados con anterioridad, a los pines del dispositivo 1. En el protoboard también se encuentran los pulsadores conectados al Arduino Mega, y el led RGB al Arduino Nano. Este led se conecta a los pines analógicos 09, 010 y 011, cada uno a un color en específico, Red (09), Green (010) y Blue (011), y la pata metálica más larga se conecta a tierra (GND). Para evitar quemar el led se les añade resistencias entre la placa y las patas del led, cada color tiene un valor distinto de resistencia, en el rojo usamos 560ohm, en verde y azul 470ohm cada una.

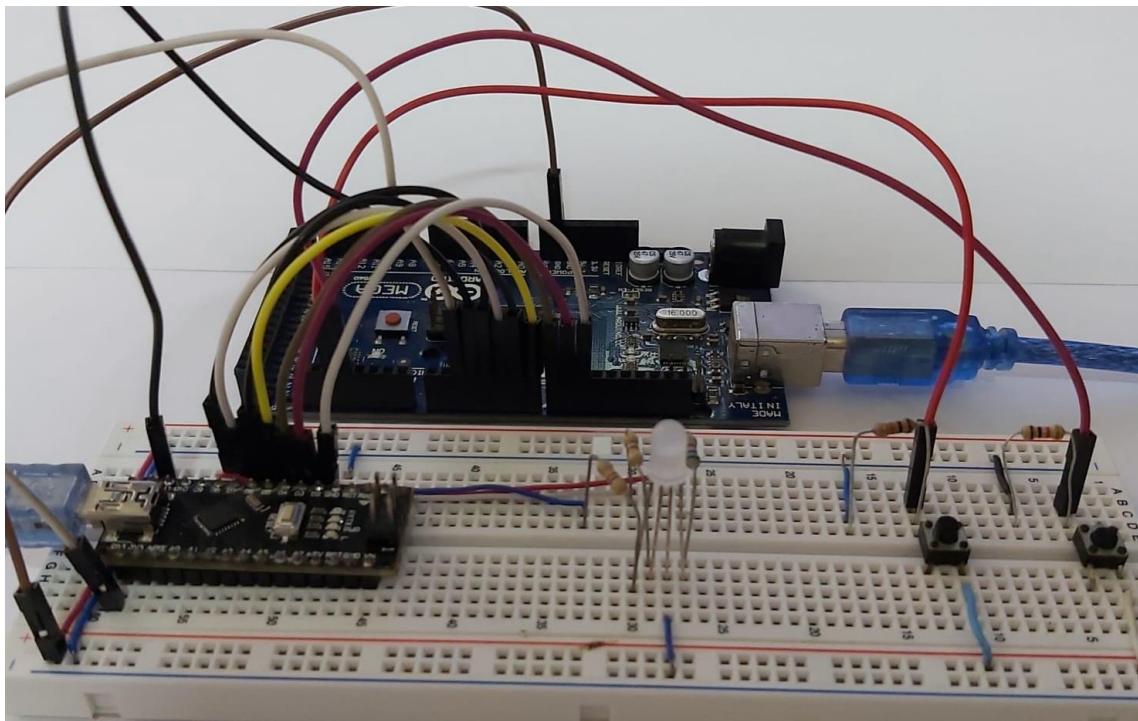


Figura 2. Enfoque principal de la conexión

#### Implementación del Software Receptor

Al igual que en el caso del emisor, se realiza un código para que puede efectuar la tarea deseada. Para esta instancia se declaran tres listas vacías, una para cada color, estas se llenaron mediante la obtención de los estados de los 8 bits enviados desde el dispositivo 1. Para recibir adecuadamente los valores se declara una interrupción controlada por el clock cuando el pin pasa de bajo a alto. Luego en el loop además de guardar los valores en los arrays, se los reconstruyen de binario a decimal para que sean los parámetros del led RGB.

De esta forma se realizó la comunicación en paralelo entre dos dispositivos programables, explicando el hardware y software del trabajo práctico. A continuación, se adjunta un diagrama con las partes, funcionalidad y conexiones en específico de dicha comunicación.

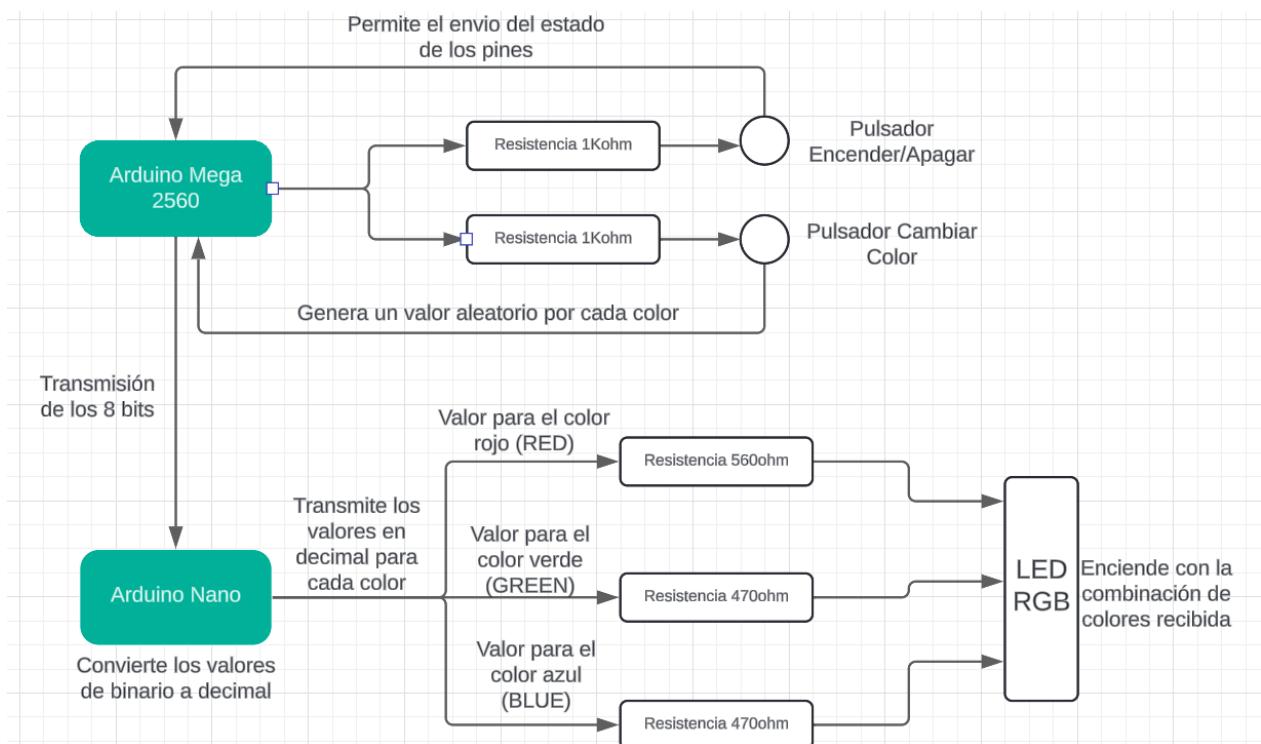


Figura 3. Diagrama de bloques Comunicación en Paralelo.

### Comunicación en Serie

Se implementa una comunicación en serie entre dos dispositivos programables, nuevamente se utiliza un Arduino Mega 2560 (dispositivo 1) y un Arduino Nano (dispositivo 2). A diferencia de la comunicación en paralelo, la transmisión de datos es secuencial, es decir, que los bits se transmiten uno por uno.

La conexión alámbrica se lleva a cabo mediante un cable que se conecta al pin TX1 del dispositivo que transmite los datos, al pin RX1 del dispositivo que recibe los datos. Como en este proyecto se necesita de una comunicación bidireccional, esta conexión se realiza en ambas placas por igual.

Mediante esta comunicación se busca poder encender un led RGB, el cual, los colores serán enviados al dispositivo 2 por el Arduino Mega. Para que se cumpla esta condición, la comunicación deberá encenderse mediante un pulsador, luego desde el Arduino Nano se conecta un pulsador en donde envía una señal solicitando la conexión con el dispositivo 1, si esto se concreta se deberá encender un led para verificarlo. Cuando se envíen los datos, se encenderá otro led indicando que el dispositivo 2 recibió los bits del dispositivo 1. Cabe aclarar que para el cambio de color en el led RGB se conecta un segundo pulsador al Arduino Mega para generar tres valores aleatorios que conformarán la combinación de colores.

### Implementación de Hardware Emisor

En esta conexión se emplea un cable que conecte el pin TX1 del Arduino Mega al pin RX0 del Arduino Nano y otro cable que parte del TX1 del Nano al RX1 del Mega, de esta manera se establece el canal bidireccional donde serán enviados los datos de ambas partes.

Al dispositivo 1 se le adjuntará dos pulsadores, al igual que en la conexión anterior, uno para encender y apagar la comunicación, y el otro es para el cambio de color del led RGB. Para esta instancia también se colocan un par de leds, ambos conectados a sus respectivas resistencias.

### Implementación de Hardware Receptor

Para el dispositivo 2 se le conectarán el led RGB a los pines analógicos 09, 010 y 011, de igual forma que en la comunicación en paralelo, se respetarán las mismas resistencias y color para cada pin. Como pasa con la emisión, habrá algún que otro cambio en la recepción, en este caso, un pulsador extra el cual tendrá la función de solicitar al emisor el permiso para realizar la conexión entre ambos dispositivos.

### Implementación de Software Emisor y Receptor

Como se habló con anterioridad, se utilizan dispositivos programables los cuales para realizar su función requieren de un código de programación, uno para emisor y otro para receptor. Como no contamos con el tiempo necesario para realizar la programación adecuada, no podremos compartir avances de esta parte. No obstante, se explicará nuestra visión y posible conclusión a esta comunicación, la cual será realizada en brevedad.

Compartiremos la idea física de la conexión de los componentes y los problemas por los cuales no pudimos concretar dicha consigna.

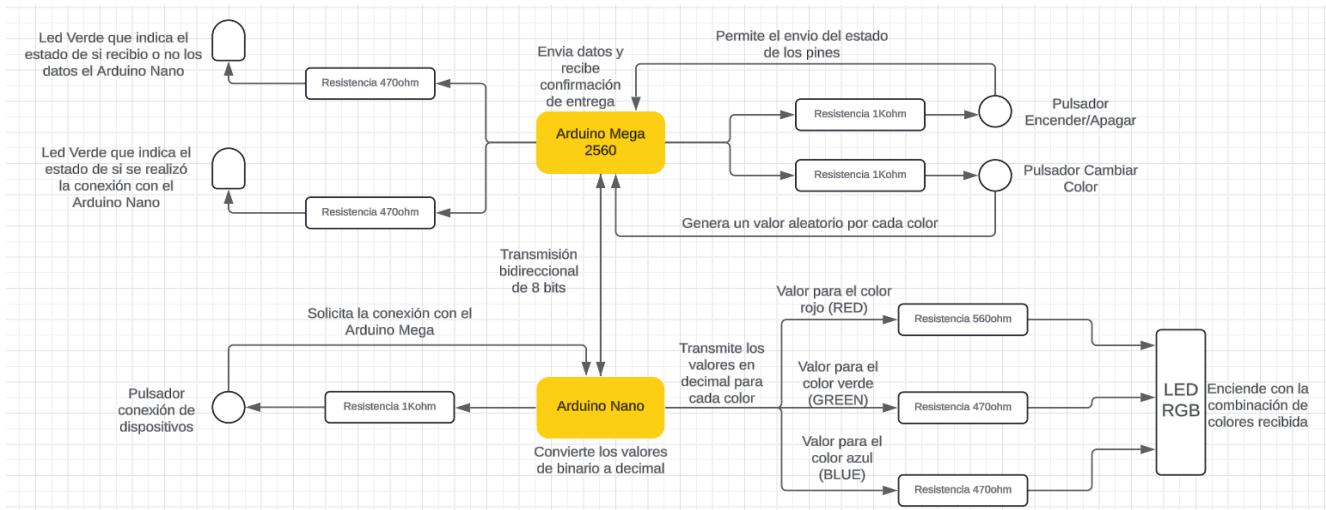


Figura 4. Diagrama de bloques Comunicación en Serie.

### Dificultades encontradas

Los problemas por los cuales atravesamos y complicaron la finalización momentánea del práctico son por parte de los componentes y por problemas en la programación. Al inicio de este proyecto se compro leds RGB pensando que eran de cátodo común, pero al momento de probarlos nos percatamos de que no eran los necesarios y que teníamos en nuestro poder leds RGB de ánodo común. Esto atrasó la continuidad de las pruebas, por lo que tuvimos que encontrar una solución rápida, en primer lugar, teníamos la intención de cambiarlos por los de cátodo común y continuar tal y como lo teníamos previsto, en caso de esto no fuera posible se planteo la idea de utilizar transistores, lo cual complejiza el hardware del proyecto. En el transcurso de un par de días pudimos conseguir los leds RGB necesarios y continuar con el trabajo.

Otra dificultad fue un error a la hora de cargar los programas a la placa de Arduino Nano. El problema era que el bootloader, que es la herramienta que carga el software del sistema en el dispositivo y determina la prioridad de los procesos que se ejecutan, estaba corrompido, por lo que no podíamos subir ningún programa que hicimos. La única solución que encontramos fue quemar el bootloader y reinstalar sus drivers. Luego de llevarlo a cabo pudimos solucionarlo sin más inconveniente.

A la hora de implementar los programas de Arduino nos encontramos con algunas dificultades que, si bien nos llevo algo de tiempo corregir y perfeccionar, pudimos solucionarlas en un día de pruebas.

### Conclusión y agradecimiento

Como conclusión podemos declarar que el aprendizaje de conceptos y práctica sobre comunicaciones fue de gran ayuda ya que pudimos solucionar y practica con los diferentes modos de transmisión de datos, placas de Arduino y componentes electrónicos. Agradecemos de antemano cualquier intervención que puedan ayudar a mejorar el escrito y/o el práctico.

### Bibliografía

Rey Canul. (8 de mayo 2021). *Ejemplo de Comunicación Serial Y Paralela*.  
<https://www.youtube.com/watch?v=lTmOwKMefBM&t=2s>

Jorge APC. (25 de diciembre 2017). *Arduino conversión decimal a binario bitRead()*  
Jorge APC. <https://www.youtube.com/watch?v=hxvi8m141A>

Master Manual Mx. (17 de diciembre 2020). *INTERRUPCIONES en Arduino // IngeDonManual*. <https://www.youtube.com/watch?v=pZaYKm6zcw0>

MecatrónicaEs. (3 de agosto 2021). *ARDUINO NO CARGA PROGRAMA / QUEMAR BOOTLOADER / RESETEAR ARDUINO / SOLUCIÓN*.  
<https://www.youtube.com/watch?v=wJ6Y7X0IIIU&t=5s>

*tecno-consejos. (28 de enero 2022). Arduino Nano NO funciona!! SOLUCIÓN!!*  
<https://www.youtube.com/watch?v=-S7QTFiUt9s&t=3s>

*Héctor Pérez. (17 de septiembre 2019). ¿Cómo instalar el driver de una placa arduino genérica o china? <https://www.youtube.com/watch?v=zaeggzQylXE&t=1s>*

*Opentronika. (9 de julio 2021). Toda la librería serial de arduino en un solo video.*  
[https://www.youtube.com/watch?v=\\_jQj0Y04TX0&t=2s](https://www.youtube.com/watch?v=_jQj0Y04TX0&t=2s)