



UNIVERSIDAD LINDA VISTA

EX-FINCA STA CRUZ #1 PUEBLO NUEVO SOLISTAHUACÁN, CHIAPAS

INGENIERÍA EN DESARROLLO DE SOFTWARE

SISTEMAS DIGITALES

NOMBRE DEL ALUMNO: JOSE MOISES MARTINEZ HERNANDEZ

SINDY FABIOLA PERDOMO RAPALO

DOCENTE: NELSON ORTIZ LÓPEZ

ACTIVIDAD DE APRENDIZAJE:

LABORATORIO: COMUNICACIÓN SERIAL EN ARDUINO

FECHA DE ENTREGA:

28/01/25

Introducción

En el mundo actual, la comunicación entre dispositivos es muy importante para el desarrollo de programas y de la sociedad. Esta interconexión permite la creación de sistemas complejos que facilitan la automatización, el intercambio de información y la innovación tecnológica.

En este contexto, la comunicación serial juega un papel crucial al permitir el envío y recepción de datos de manera secuencial, lo que es esencial para el funcionamiento de muchos dispositivos electrónicos. La siguiente práctica, se analiza la comunicación serial utilizando Arduino, usando el Monitor Serial, el cual permite introducir comandos y muestra los resultados ya sea una cadena de texto o algún actuador. Esto ayudará a comprender cómo funciona la comunicación entre dispositivos y cómo poder controlarlos.

Resultados

A continuación se muestran los resultados de los ejercicios realizados, tanto los resultado el Arduino IDE como en el simulador [Tinkercad](https://www.tinkercad.com/things/am8HT1LcoCk-contadorarduino) por medio de links los cuales contienen los códigos usados.

Recibir datos serial desde el Arduino

<https://www.tinkercad.com/things/am8HT1LcoCk-contadorarduino>

```
1  int cont=0;
2
3  void setup() {
4    // put your setup code here, to run once:
5    Serial.begin(9600);
6
7  }
8
9  void loop() {
10   // put your main code here, to run repeatedly:
11   Serial.println("Contador = " + String(cont++));
12   if(cont == 10){
13     cont=0;
14   }
15   delay(1000);
16 }
17
```

Output Serial Monitor x

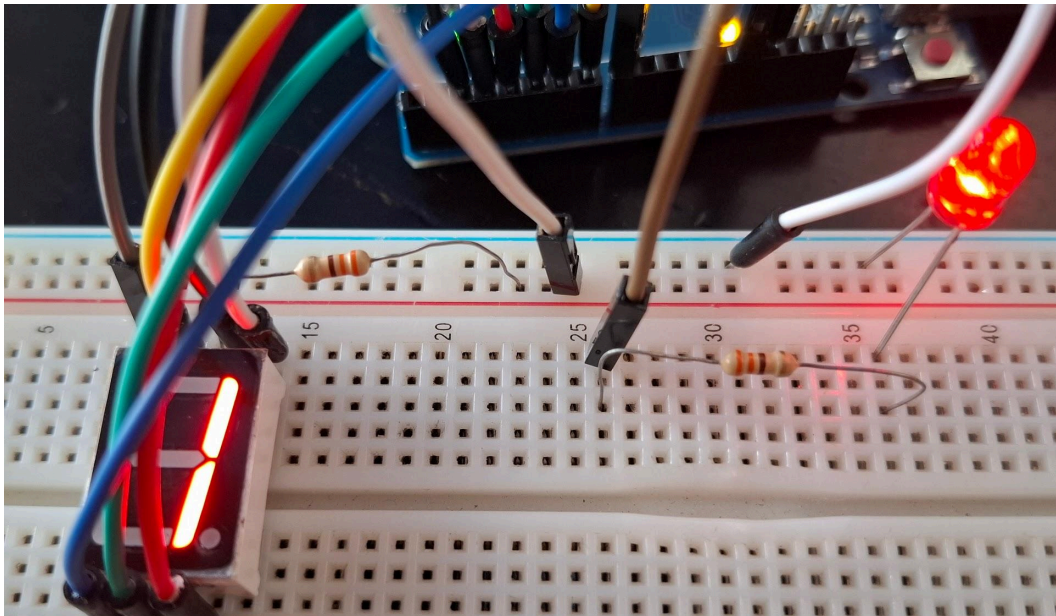
Message (Enter to send message to 'Arduino Uno' on 'COM5')

```
Contador = 0
Contador = 1
Contador = 2
Contador = 3
Contador = 4
Contador = 5
Contador = 6
Contador = 7
Contador = 8
Contador = 9
Contador = 0
Contador = 1
Contador = 2
Contador = 3
Contador = 4
Contador = 5
```

En este planteamiento, se configuró al Arduino para que contara de 0 a 9 y al llegar a este último se reiniciara a 0 y se repitiera el conteo.

Enviar datos serial al Arduino

<https://www.tinkercad.com/things/htvPiefyxdO-decode-bcdserialin>



```
27
28 void loop() {
29   if (Serial.available() > 0) { // Verifica si hay datos disponibles
30     char dato = Serial.read(); // Lee el dato recibido
31
32     if (dato == 'E') { // Si el dato es 'E'
33       digitalWrite(ledPin, HIGH); // Enciende el LED
34       Serial.println("LED ON");
35     }
36     else if (dato == 'A') { // Si el dato es 'A'
37       digitalWrite(ledPin, LOW); // Apaga el LED
38       Serial.println("LED OFF");
39     }
40     else if (dato >= '0' && dato <= '9') { // Si es un número entre 0 y 9
41       int num = dato - '0'; // Convierte el carácter a número
42       mostrarNumero(num); // Muestra el número en el display
43       Serial.print("DATO: ");
44       Serial.println(num); // Envía el dato al monitor serial
45     }
46   }
47 }
```

Output Serial Monitor ✕

Message (Enter to send message to 'Arduino Uno' on 'COM5')

```
DATO: 9
DATO: 8
DATO: 7
DATO: 6
DATO: 5
DATO: 4
DATO: 3
DATO: 2
DATO: 1
LED ON
```

Se configuró un Display de 7 segmentos y un LED, E y A para encender y apagar el led correspondientemente y del 0 al 9 para mostrar un número en el display.

Escribir desde PC y Arduino Contesta con cadena de caracteres

<https://www.tinkercad.com/things/1Uuno8GUgWP-escribirpccontestaarduino>

*Este ejercicio solo funciona en Arduino IDE.

```
saludo_lab_2.ino
4 void setup() {
5   Serial.begin(9600); // Inicia la comunicación serie a 9600 baudios
6 }
7
8 void loop() {
9   if (inputComplete) {
10    inputSerial.trim(); // Eliminar espacios en blanco y el salto de línea
11
12    // Comparar el contenido de inputSerial
13    if (inputSerial == "Hola") {
14      Serial.println("Hola que tal!"); // Respuesta para "Hola"
15    } else if (inputSerial == "Adios") {
16      Serial.println("Hasta luego!"); // Respuesta para "Adios"
17    } else {
18      Serial.println("El dato recibido es inválido!!"); // Respuesta para cualquier otro dato
19    }
20
21    // Reiniciar variables para la próxima entrada
22    inputSerial = ""; // Limpiar la variable
23    inputComplete = false; // Marcar la entrada como no completa
24  }
25 }
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM5')

Hola que tal!
Hasta luego!
El dato recibido es inválido!!

Aquí, Arduino responde a “Hola” con “Hola que tal” y “Adios” con “Hasta luego”, se usó sentencia if para asignar a cada palabra una respuesta.

Encender y Apagar LED desde la PC mediante cadena de caracteres por el Puerto Serie

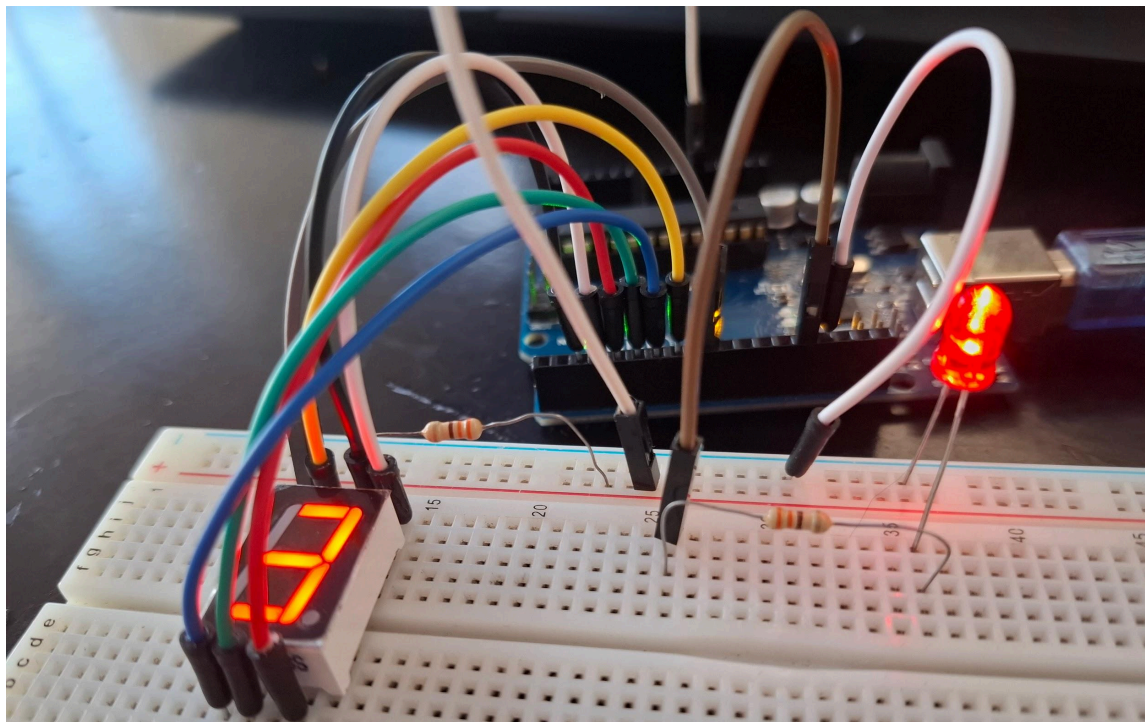
<https://www.tinkercad.com/things/ifYGwIVUydB-decode-bcdserialingreetings>

```
31 inputSerial = Serial.readString(); // Leer entrada del puerto serie
32 inputSerial.trim(); // Eliminar espacios o saltos de línea
33
34 if (inputSerial == "E") { // Si el dato es 'E'
35     digitalWrite(ledPin, HIGH); // Enciende el LED
36     Serial.println("LED ON");
37 }
38 else if (inputSerial == "A") { // Si el dato es 'A'
39     digitalWrite(ledPin, LOW); // Apaga el LED
40     Serial.println("LED OFF");
41 }
42 else if (inputSerial >= "0" && inputSerial <= "9") { // Si es un número entre 0 y 9
43     int num = inputSerial.toInt(); // Convierte el carácter a número
44     mostrarNumero(num); // Muestra el número en el display
45     Serial.print("DATO: ");
46     Serial.println(num); // Envía el dato al monitor serial
47 }
48 else if (inputSerial == "Hola") {
49     Serial.println("Hola que tal!"); // Respuesta para "Hola"
50 }
51 else if (inputSerial == "Adios") {
52     Serial.println("Hasta luego!"); // Respuesta para "Adios"
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM5')

Hola que tal!
Hasta luego!
LED OFF
LED ON
DATO: 0
DATO: 9
DATO: 8
DATO: 7
DATO: 3



En este ejercicio se combinó el ejercicio 2 y 3, pues Arduino contesta a los saludos del ejercicio 3 y puede mostrar número en el display y apagar o encender el led, todo esto con comando y el Serial Monitor.

Controlar una secuencia de Led's desde la PC mediante cadena de caracteres por el Puerto Serie

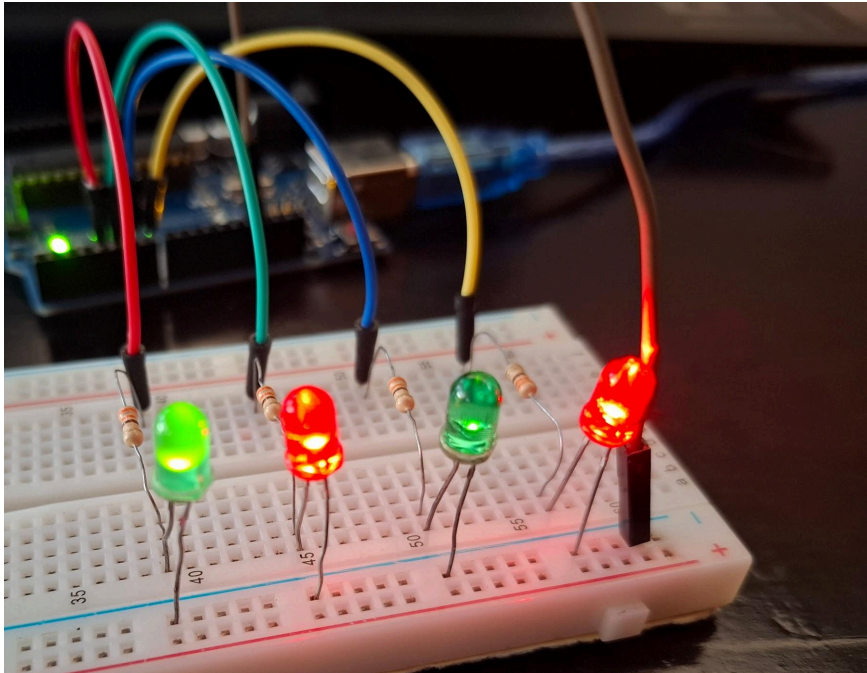
<https://www.tinkercad.com/things/b9lXSIM4X4n-secuencialeds>

```
22 void loop(){
23   if (Serial.available() > 0){
24     inputSerial = Serial.readString(); // Leer entrada del puerto serie
25     inputSerial.trim(); // Eliminar espacios o saltos de línea
26
27     Serial.print("Comando ingresado: ");
28     Serial.println(inputSerial);
29
30     if(inputSerial == "serie 1"){
31       Serial.println("Ejecutando Serie 1");
32       serieUno(); // Llamada a la función serieUno
33     }else if (inputSerial == "serie 2"){
34       Serial.println("Ejecutando Serie 2");
35       serieDos(); // Llamada a la función serieDos
36     }else if(inputSerial == "serie 3"){
37       Serial.println("Ejecutando Serie 3");
38       serieTres(); // Llamada a la función serieTres
39     }else{
40       Serial.println("Comando no reconocido. Intenta de nuevo.");
41     }
42   }
```

Output Serial Monitor ✕

Message (Enter to send message to 'Arduino Uno' on 'COM5')

```
Comando ingresado: Serie 1
Comando no reconocido. Intenta de nuevo.
Comando ingresado: serie 1
Ejecutando Serie 1
Serie 1 terminada.
Comando ingresado: serie 2
Ejecutando Serie 2
Serie 2 terminada.
Comando ingresado: serie 3
Ejecutando Serie 3
Serie 3 terminada.
```

En este ejercicio, se usó sentencia if que iban acorde con el número de sentencia que se quería que los LEDs se prendieran o apagaran, y estas a cumplir las condiciones, llamaban funciones que contenían la dinámica a realizar.

Conclusiones

En esta práctica, hemos descubierto el poder de la comunicación serial, que nos permite enviar datos de manera secuencial, bit a bit. Esto ha sido especialmente emocionante al trabajar con Arduino y el Monitor Serial, ya que hemos podido enviar comandos y ver cómo responden los dispositivos, ya sea a través de mensajes de texto o controlando LEDs.

Una de las cosas más interesantes ha sido cómo podemos combinar la comunicación serial con estructuras de control como las sentencias if, lo que nos permite dar instrucciones específicas a los dispositivos y obtener respuestas precisas. Además, hemos podido aplicar conocimientos previos sobre el manejo del tiempo y el uso de funciones, lo que ha enriquecido nuestra comprensión de cómo se pueden crear sistemas más complejos y útiles.

Lo que más nos ha gustado es darse cuenta de que no solo podemos interactuar con Arduino a través del código, sino que también podemos hacerlo de manera más directa y tangible. Esto nos ha abierto los ojos a las posibilidades de automatizar tareas cotidianas de manera más inteligente. Con esto se puede crear un sistema que ajuste el ambiente según tu estado de ánimo, reproduciendo la música que te gusta, encendiendo las luces en el lugar adecuado o adaptando el clima de casa bajo ciertas condiciones.

Esta práctica fomenta el trabajo en equipo y la creatividad, ya que los integrantes colaboraron mutuamente, explicando conceptos y resolviendo dudas en conjunto. La experiencia de seguir las instrucciones sin omitir pasos y alcanzar los resultados de manera colaborativa fue gratificante. Estas habilidades serán valiosas en el futuro, permitiéndonos enfrentar problemas, proponer soluciones o implementarlas para beneficiar a la sociedad.