

SERIAL COMMUNICATION



What is our GOAL for this CLASS?

In this class, we learned about a new controller “Arduino” and how machines talk to each other using Serial communication.

What did we ACHIEVE in the class TODAY?

- We learned how to connect 2 Arduinos with each other.
- We used serial communication protocol to exchange data between the 2 Arduinos.

Which CONCEPTS/ CODING BLOCKS did we cover today?

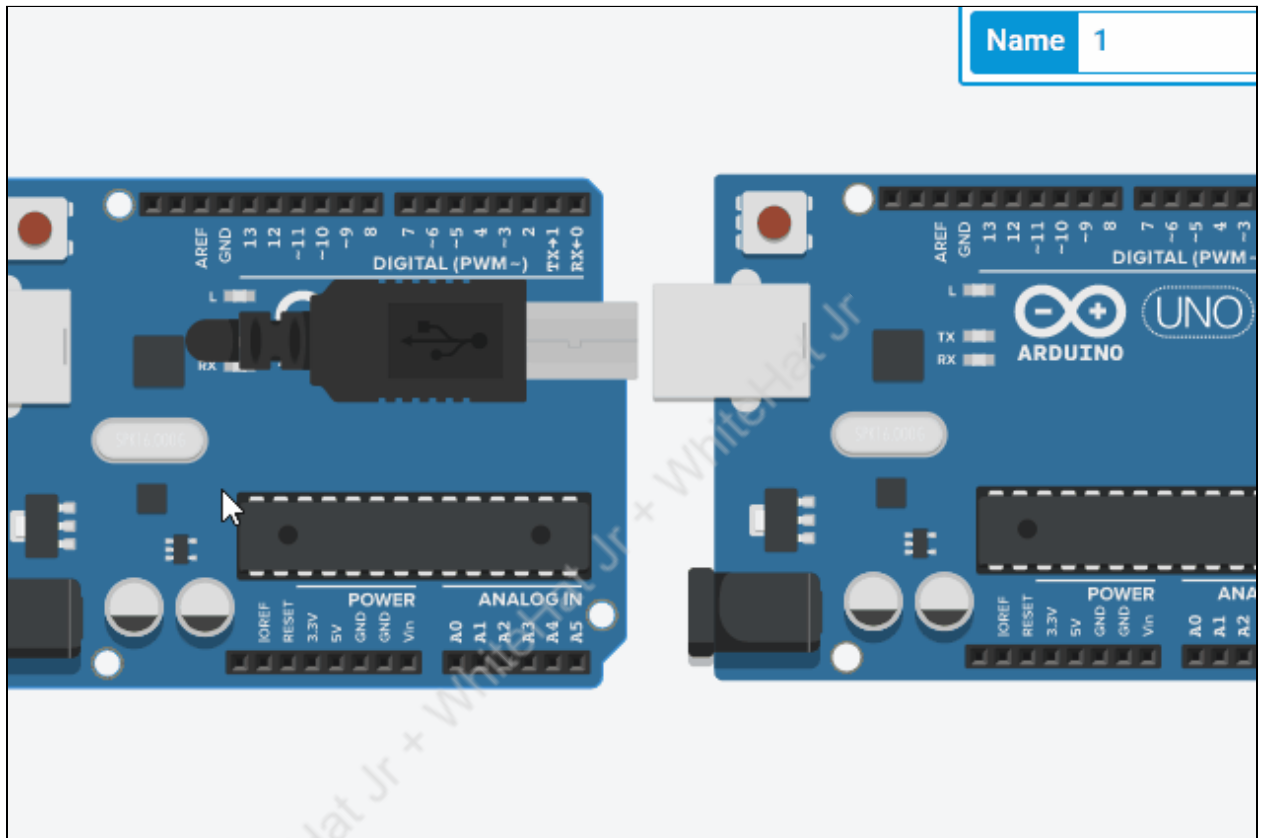
- Concepts: Transmitting data, Receiving data, Infinite loops, Sequencing of code, Controlling LED brightness using a potentiometer, Mapping 2 ranges.
- Coding blocks: Serial.begin(), Serial.print(), Serial.available(), delay(), Serial.readString(), Serial.setTimeout(), pinMode(), digitalWrite(), String keyword.

How did we DO the activities?

1. Open the [tinkercad](#) simulator and create a new **Circuit Project**.
2. Drag out **2 Arduinos** and connect them using the instructions given in the table below, or you can refer to the following gif as well.

Arduino 1	Arduino 2
RX	TX

TX	RX
GND	GND



<https://s3-whjr-curriculum-uploads.whjr.online/feeaa55a-d3e2-4705-ae50-1773cb5434cd.gif>

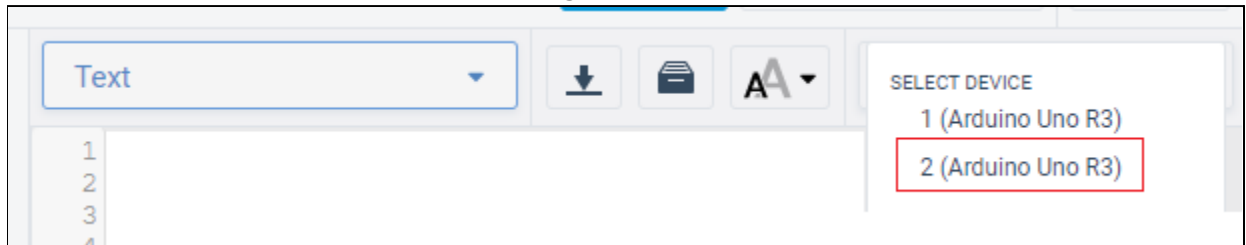
3. Arduino 1 will act as a **transmitter**. To do so, write code for Arduino 1. In the **setup()** method, initialize the communication using the **.begin()** method.

```
void setup()
{
  Serial.begin(9600);
}
```

4. In the **loop()** method, we will send a string **"hello i am arduino 1"** towards the **Arduino 2** using the serial communication protocol at an interval of **1 second**. To do so, use the **Serial.print("hello i am arduino 1")** and **delay(1000)** methods.

```
void loop()
{
  Serial.print("hello i am arduino 1");
  delay(1000);
}
```

5. Now our Arduino 2 will act as a **receiver**. To do so, write code for Arduino 2. Change the controller to **2 (Arduino Uno R3)**, using the controller menu.



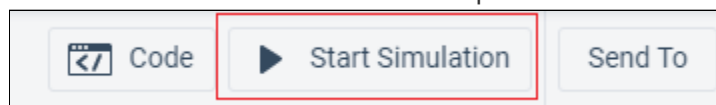
6. In the **setup()** method,
- Initialize the communication using the **.begin()** method
 - Set the serial wait time to **100 ms** using the **.setTimeout()** method.
 - Configure **pin 13** as **output** using the **pinMode()** method.

```
void setup()
{
  Serial.begin(9600);
  Serial.setTimeout(100);
  pinMode(13, OUTPUT);
}
```

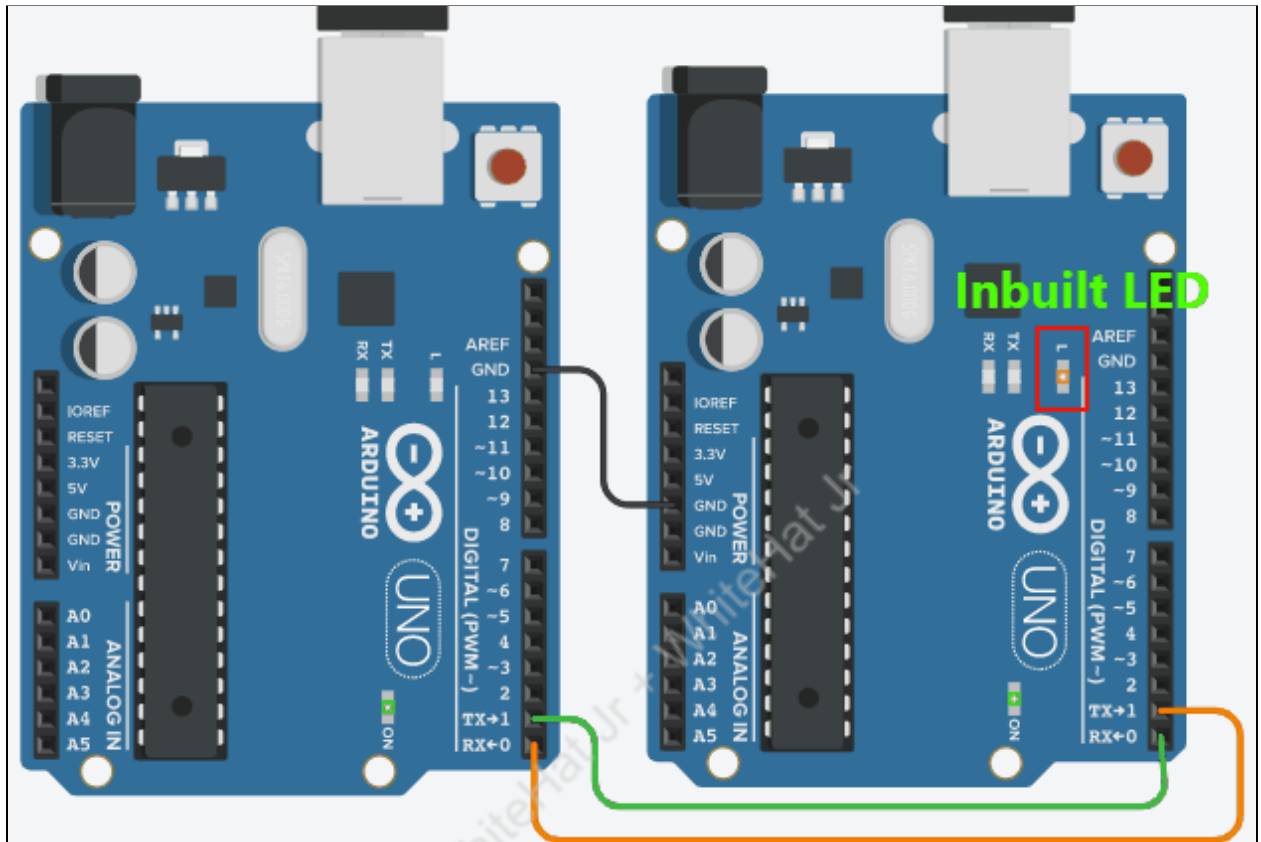
7. In the **loop()** method,
- Check if the data is available using the **.available()** method
 - If there is data, read it into a string variable using the **.readString()** method.
 - Check if the message received is the **same** as the one, sent by Arduino 1, using an **if statement**.
 - If yes, turn on the **inbuilt LED** of **Arduino 2**, connected to **pin 13** internally, so that we can verify that Arduino 2 has received the message successfully, using the **digitalWrite(13, HIGH)** method.

```
void loop()
{
  if (Serial.available()){
    String data = Serial.readString();
    if (data == "hello i am arduino 1"){
      digitalWrite(13,HIGH);
      delay(500);
      digitalWrite(13,LOW);
    }
  }
}
```

8. Click on the **Start Simulation** button to see the output.



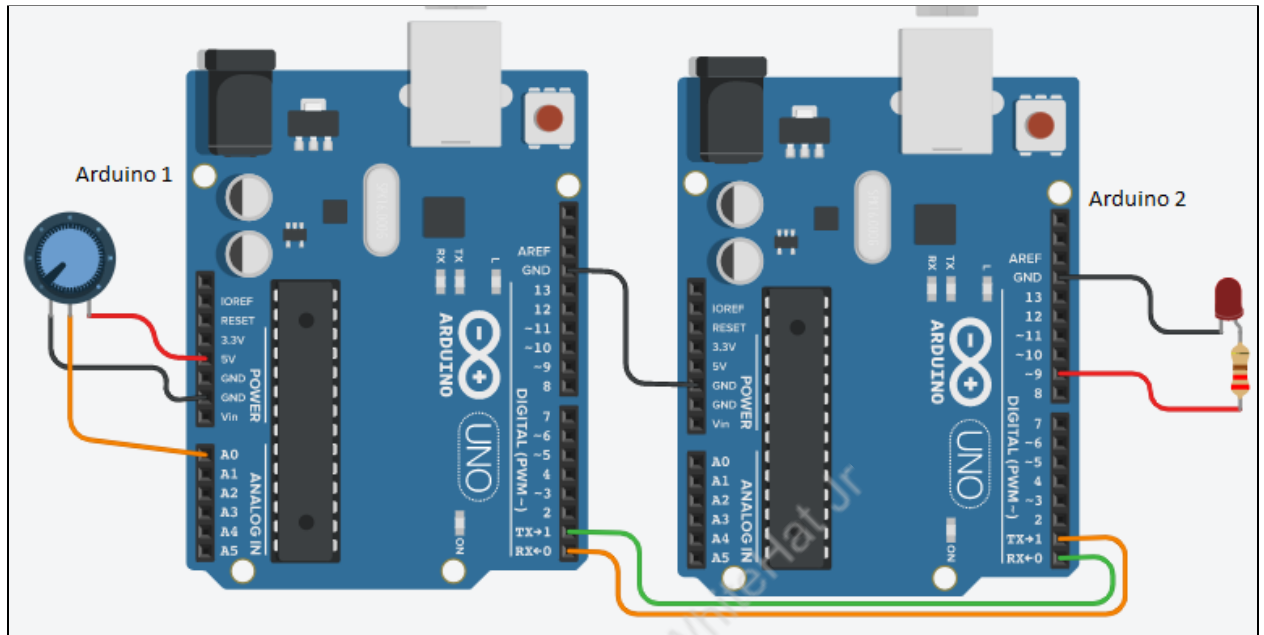
9. The output would look like this,



<https://s3-whjr-curriculum-uploads.whjr.online/127c78ea-507d-49d0-88f9-4e4e242f0b39.gif>

10. Now, it's time to use this principle. We will control the brightness of an LED that is connected with **pin 9** of the **2nd Arduino** using a **potentiometer** that is connected with **Arduino 1**. To do so, create the following circuit using the instructions given in the graphics below. Same colored terminals should be connected together.

Arduino 1	Arduino 2	Potentiometer	LED	Resistor
5 volts	Ground 1	Terminal 1	Cathode (negative leg)	Terminal 1
A0	RX (Pin 0)	Wiper (Middle pin)	Anode (positive leg)	Terminal 2
Ground1	TX (Pin 1)	Terminal 2		
Ground 2	Ground 2			
TX (Pin 1)	Pin 9			
RX (Pin 0)				



11. Arduino 1 will act as a **transmitter**. To do so, write code for Arduino 1. In the **setup()** method, initialize the communication using the **.begin()** method.

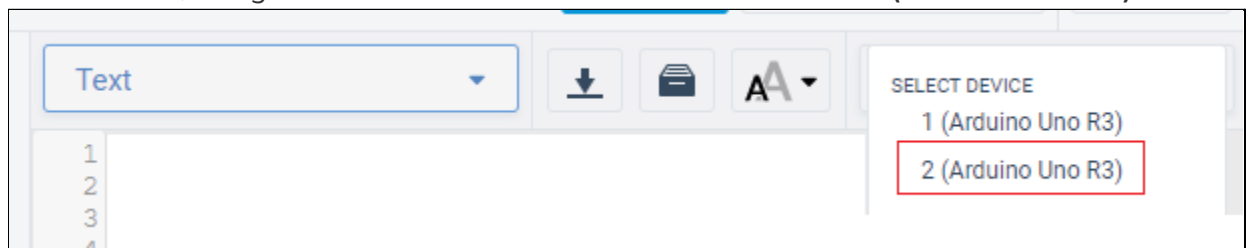
```
void setup()
{
  Serial.begin(9600);
}
```

12. In the **loop()** method,

- We will read the state of the **A0 pin** using the **analogRead()** method.
- Convert the data into string format using the **String** keyword.
- Send this **potentiometer** data towards **Arduino 2** using the **.print()** method.
- Wait for **500 ms** using the **delay()** method.

```
void loop()
{
  String pot_val = String(analogRead(A0));
  Serial.print(pot_val);
  delay(500);
}
```

13. Now our Arduino 2 will act as a **receiver**. To do so, write code for Arduino 2. Change the controller, using the controller menu. Select the controller as **2 (Arduino Uno R3)**.



14. In the **setup()** method,

- Initialize the communication using the **.begin()** method
- Set the serial wait time to **100 ms** using the **.setTimeout()** method.
- Configure **pin 9** as **output** using the **pinMode()** method.

```
void setup()
{
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  Serial.setTimeout(100);
}
```

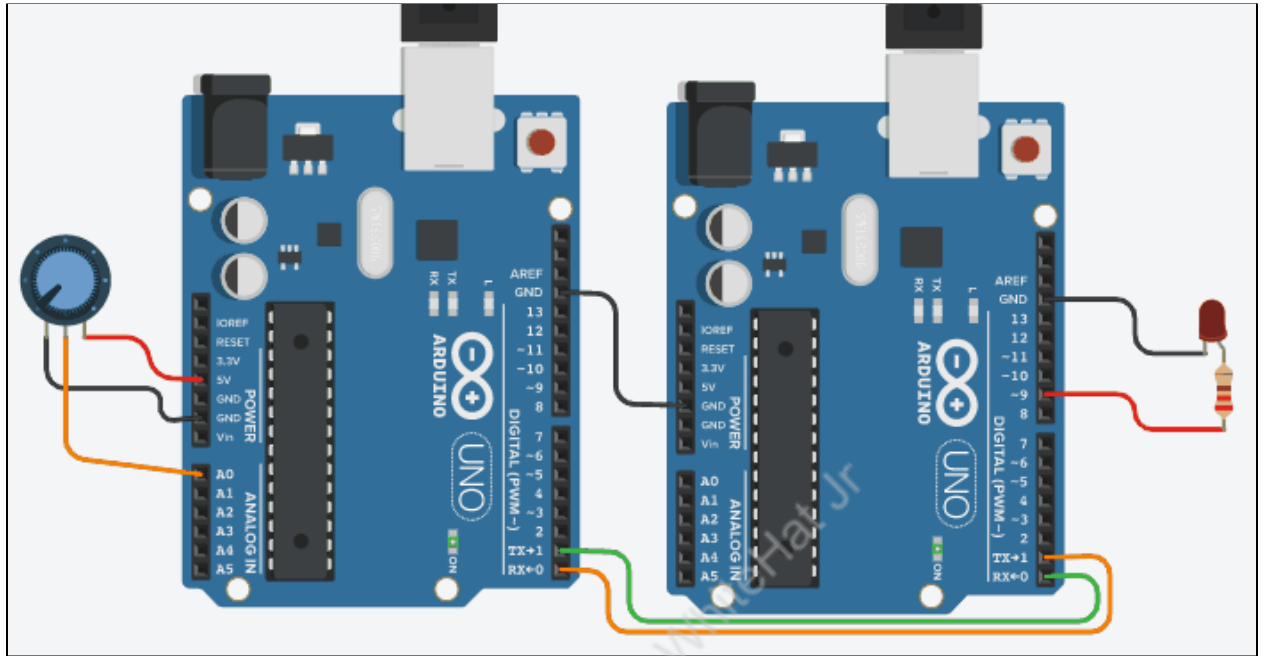
15. In the **loop()** method,

- Check if the data is available using the **.available()** method
- If there is data, read it in a string variable using the **.readString()** method.
- Convert that value to an integer using the **.toInt()** method.
- Map the value from **0 to 255** range using the **map()** method.
- Control the LED **brightness** using the **analogWrite()** method.

```
void loop()
{
  if (Serial.available()){
    int data = Serial.readString().toInt(); // 0-1023 range

    // mapping data from 0-255
    data = map(data , 0 , 1023 , 0 , 255);
    analogWrite(9 , data);
  }
}
```

16. The output would look like this,



<https://s3-whjr-curriculum-uploads.whjr.online/3ae789e3-7084-42e3-8448-65798a28e257.gif>

What's NEXT?

In the **next class**, we will learn about the RTC (**Real-time clock**) module.

Expand Your Knowledge

To know more about Serial communication, [click here](#).