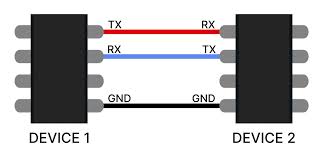
### UART - (Universal Asynchronous Receiver-Transmitter)

UART is an asynchronous serial communication protocol that allows communication between devices using only two wires: TX (Transmit) and RX (Receive). UART is commonly used for communication with devices such as GPS modules, Bluetooth modules, and other microcontrollers. Every Arduino board comes equipped with at least one port for UART.



Arduino UART Communication

The following code will make Arduino send hello world when it starts up.

Here using in-build Serial Library and Default Rx Tx Pins

void setup() {

Serial.begin(9600); //set up serial library baud rate to 9600

Serial.println("hello world"); //print hello world

delay(1000);

}

void loop() {

}

The following code returns whatever it receives as an input.

void setup() {

Serial.begin(9600); //set up serial library baud rate to 9600

}

void loop() {

if(Serial.available()) //if number of bytes (characters) available for reading from {

serial port

Serial.print("I received:"); //print I received

Serial.write(Serial.read()); //send what you read

}

}

Notice that **Serial.print** and **Serial.println** will send back the actual ASCII code, whereas **Serial.write** will send back the actual text. See ASCII codes for more information.

Also we Can Read the received String:

**void** **setup**() {  
    Serial.begin(9600);  
}  
  
**void** **loop**() {  
  // Check if there is any incoming data  
  **if** (Serial.available() > 0) {  
    // Read the incoming data and print it to the serial monitor  
    **String** incomingData = Serial.readString();  
    Serial.println(incomingData);  
  }  
}

The Arduino Uno operates on a 5V logic level while a computer's RS232 port uses a +/-12V logic level.

Directly connecting an Arduino Uno to an RS232 port can and will damage your board.

What is SoftwareSerial?

The Arduino SoftwareSerial library was developed to emulate UART communication, allowing serial communication through any two digital pins on Arduino boards. It's useful when the hardware UART is already in use by other devices.

To set up SoftwareSerial,

first include the SoftwareSerial library in the sketch.

#**include** <SoftwareSerial.h>

Then create an instance of the SoftwareSerial object by specifying the **RX** and **TX** pins to be used for communication.

SoftwareSerial **mySerial**(2, 3); // RX, TX pins

Here is an example code for Arduino that demonstrates the use of SoftwareSerial:

#**include** <SoftwareSerial.h>  
SoftwareSerial **mySerial**(2, 3); // RX, TX pins  
**void** **setup**() {  
  Serial.begin(9600); // start hardware serial  
  mySerial.begin(9600); // start soft serial  
}  
  
**void** **loop**() {  
  **if** (mySerial.available()) {  
    Serial.write(mySerial.read()); // send received data to hardware serial  
  }  
  **if** (Serial.available()) {  
    mySerial.write(Serial.read()); // send data from hardware serial to soft serial  
  }  
}

**The Serial Library**

The Serial library is a powerful tool in Arduino that allows communication between the microcontroller and a computer or other devices via a serial connection. Some common functions include:

| **Function** | **Description** |
| --- | --- |
| **Serial.begin(speed)** | Initializes serial communication with a specified data rate. |
| **Serial.print(data)** | Sends data to the serial port for transmission as ASCII text. |
| **Serial.write(data)** | Sends raw binary data over the serial port. |
| **Serial.available()** | Returns the number of bytes available to read from the serial buffer. |
| **Serial.flush()** | Waits for outgoing serial data to complete transmission before continuing. |
| **Serial.read()** | Reads the first byte of incoming serial data and returns it as an integer. |