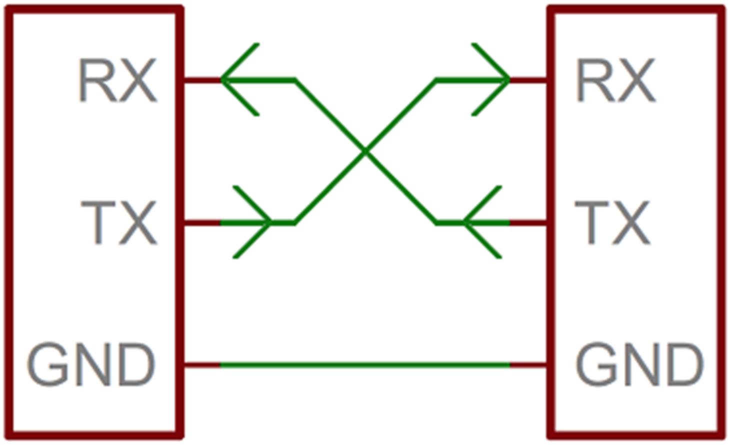
BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
EEE-F411: Internet of Things
Lab 1: Serial Monitor

Serial Communication

A serial bus consists of just two wires - the transmitter TX wire for sending data and receiver RX wire for receiving data.



Serial Communication on Arduino/ NodeMCU - Serial Monitor

Some basic functions:

- Serial.begin(baudrate) - This function is used to set the speed of transferring data at a specific baud rate. Put this inside the setup function.
- Serial.print() - This function converts the data in the ASCII text which is easily readable by human beings and prints it on the serial monitor.
- Serial.println() - This function works similarly to print() but in addition, it adds a new line
- Serial.write() - Writes binary data to the serial port. This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the print() function instead.
- Serial.read() - This function is used to read incoming serial data (i.e. can read input data from user)
- Serial.available() - This function gets the number of bytes available for reading from the serial port. We will usually use this inside an if statement to check whether we have data to read or not. For eg: if(Serial.available() > 0) { … }.

Execute the codes provided in the examples below and see the results-

Example 1: Print what's sent in the serial monitor

```
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
   }
}
```

Example 2: Serial Read

```
void setup() {
   Serial.begin(9600);
}
void loop() {
   while (Serial.available() == 0);
   int val = Serial.read() -'0';
   Serial.println(val);
}
```

Example 3: In this example, we will learn how to send messages to a computer via a USB port.

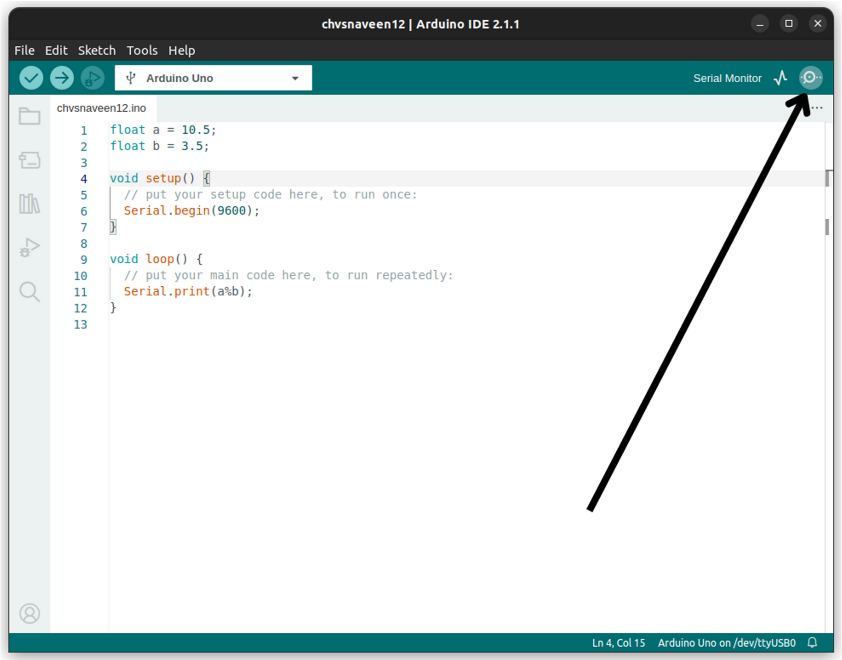
Code:

```
// serial_print

int number;

void setup() {
  number = 0;
  Serial.begin(9600);  // Initialize serial communication at 9600 bits per second
}

void loop() {
  number++;
  Serial.print("Hello World! ");  // Next print would happen on the same line
  Serial.println(number % 10);    // Next print would happen on a new line
  delay(100);                    // We don't want to flood the PC with messages
}
```

Instructions:

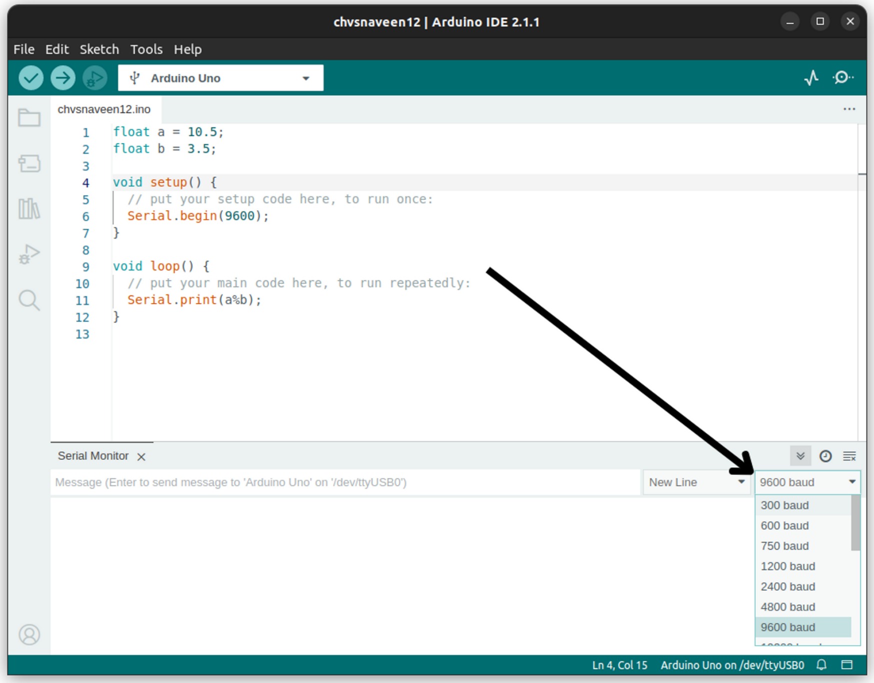
Upload the program to our **ESP32** board as per the previous instructions.

- Click on the **Serial Monitor** icon on the top right hand side, or you could press `Ctrl` + `Shift` + `M`.



```
float a = 10.5;
float b = 3.5;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print(a%b);
}
```

- Make sure to select the correct baud rate.



```
float a = 10.5;
float b = 3.5;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print(a%b);
}

```

Message (Enter to send message to 'Arduino Uno' on '/dev/ttyUSB0')

New Line

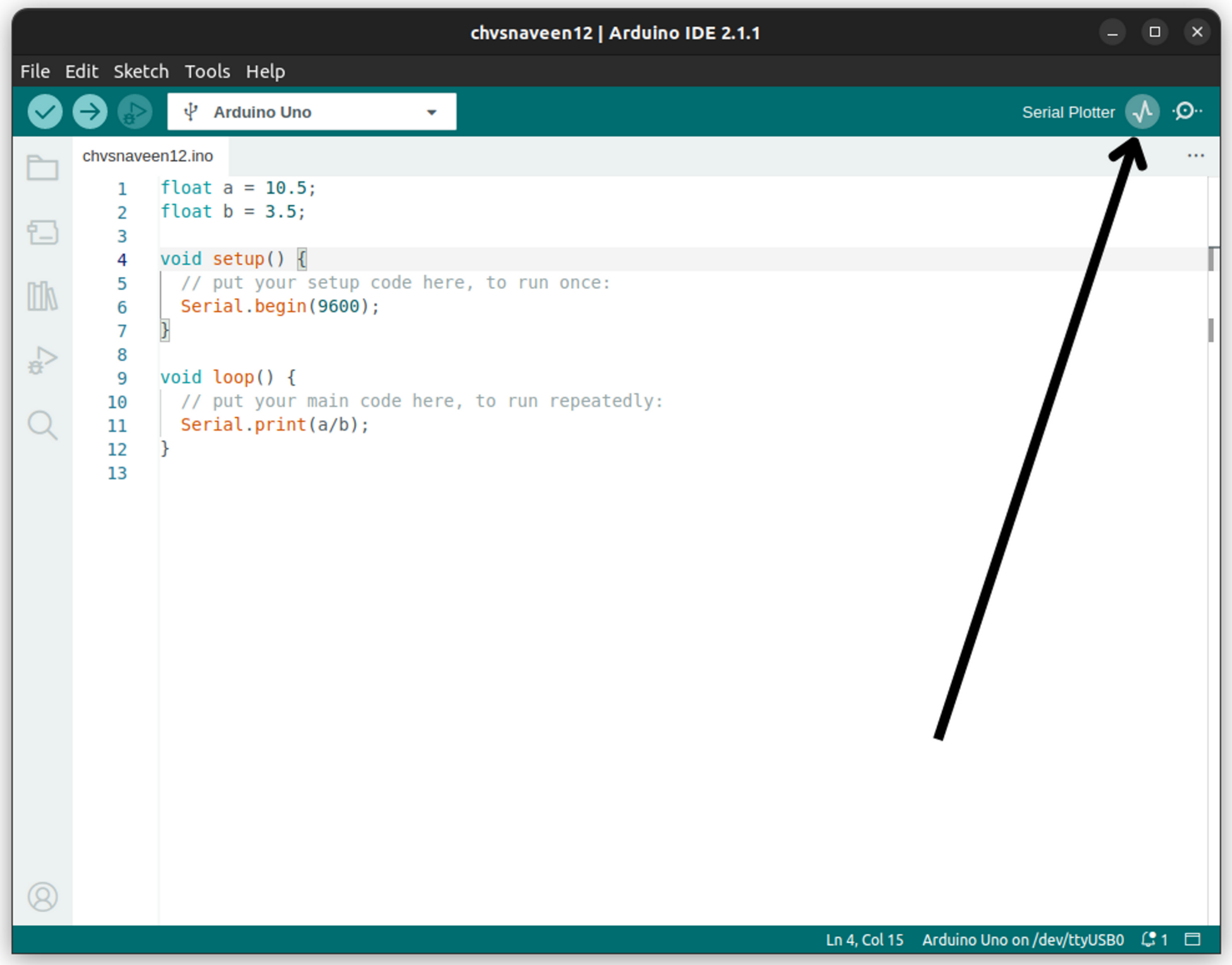
9600 baud

In this case it is 9600 baud

- You should be able to see the printed messages.

- To visualize numerical data, you can use the **Serial Plotter** tool in the **Arduino IDE**. This plots numerical values inside messages with respect to time automatically.

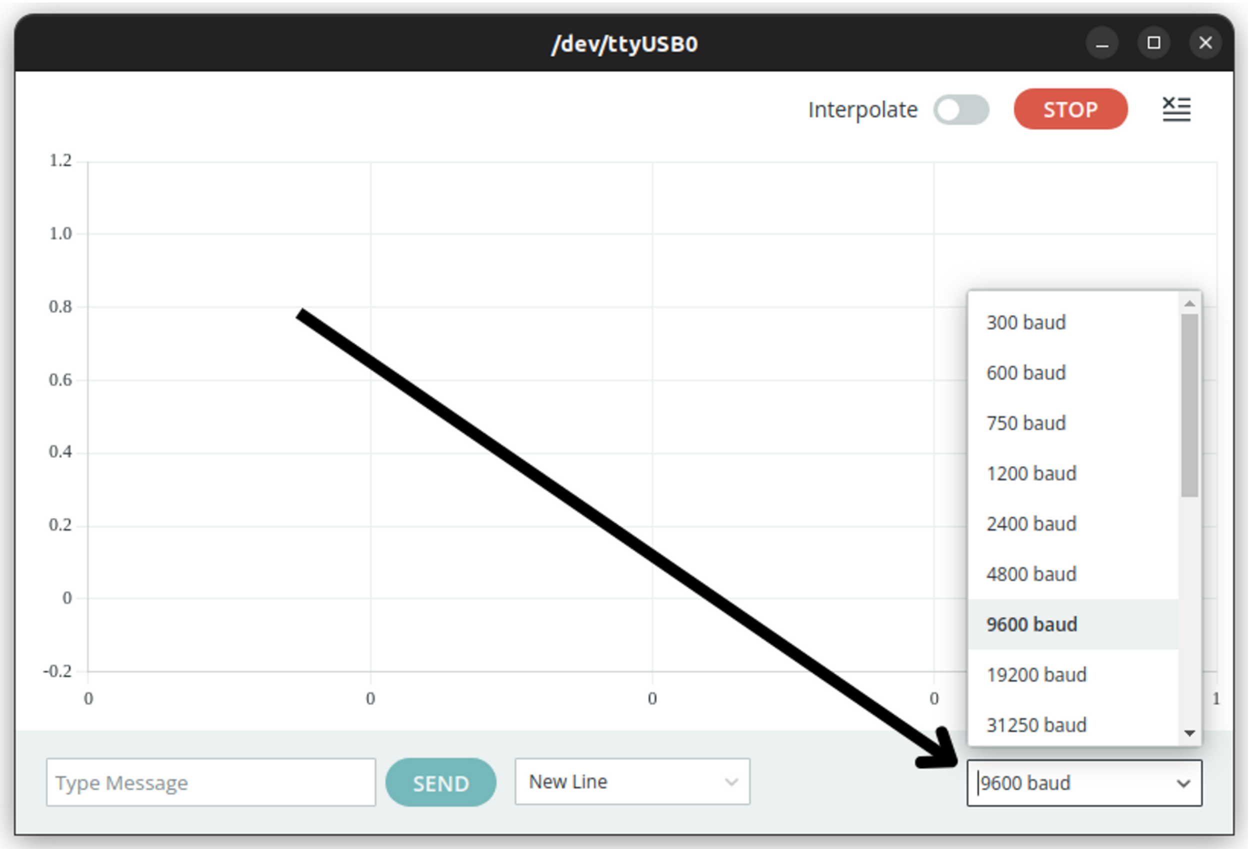
- To open it, click on its icon in the top right hand side.



chvsnaveen12 | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
Serial Plotter
chvsnaveen12.ino
Ln 4, Col 15 Arduino Uno on /dev/ttyUSB0 1

```
float a = 10.5;
float b = 3.5;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print(a/b);
}

```

- Select the right baud rate, and you should be able to see the plot of the number being printed.



Challenges (1+1 Marks)

(a) Use Serial Read to control the LED ON/OFF.
If Serial Read value is 1, turn on the LED
If Serial Read value is 0, turn off the LED

(b) Use a potentiometer to control the brightness of an LED and display the corresponding brightness on Serial Monitor.
(Note that range of pot is 0-1023 and range of led brightness is 0-255.)