# **[Demo 2: How to use multiple Serial ports on Arduino ESP32](http://www.iotsharing.com/2017/05/how-to-use-serial-arduino-esp32-print-debug.html)**

**1.Introduction**  
Arduino ESP32 use Serial port to flash software and print information on Terminal. ESP32 supports 3 Serial ports so you need not to use SoftwareSerial as in general Arduino. In this tutorial we only care about using **How to use multiple Serial port on Arduino ESP32 to print the debug information to Terminal**.

**2. Hardware**  
You do not need any extra hardware.

**3. Software**  
We use "**HardwareSerial**" class for Serial communication. It has some important interfaces:  
- **HardwareSerial(int uart\_nr)**: this is the constructor of HardwareSerial where uart\_nr is 0, 1 or 2 so we have maximum 3 Serial ports.

- **void begin(unsigned long baud, uint32\_t config=SERIAL\_8N1, int8\_t rxPin=-1, int8\_t txPin=-1):**initialize Serial port with baudrate, Serial mode (default is **SERIAL\_8N1**), rxPin and txPin (if you leave these parameters empty library will use default pins).

|  |
| --- |
| void HardwareSerial::begin(unsigned long baud, uint32\_t config, int8\_t rxPin, int8\_t txPin)  {  if(\_uart\_nr == 0 && rxPin < 0 && txPin < 0) {  rxPin = 3;  txPin = 1;  }  if(\_uart\_nr == 1 && rxPin < 0 && txPin < 0) {  rxPin = 9;  txPin = 10;  }  if(\_uart\_nr == 2 && rxPin < 0 && txPin < 0) {  rxPin = 16;  txPin = 17;  }  \_uart = uartBegin(\_uart\_nr, baud, config, rxPin, txPin, 256, false); |

- **available():** Get the number of bytes (characters) available for reading from the serial port.

- **print():** Prints data to the serial port as human-readable ASCII text.

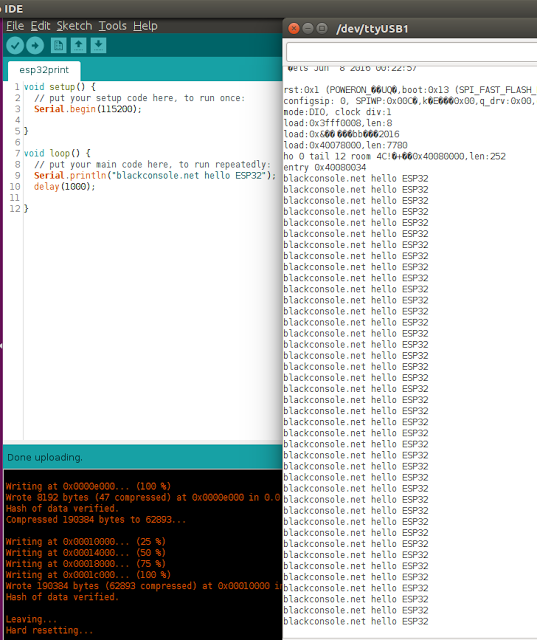
- **println():** Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n').

- **read():** Reads incoming serial data on Rx pin.

- **readStringUntil():** reads characters from the serial buffer into a string until facing terminator character.  
- Because Arduino library created a default instance **HardwareSerial Serial(0)**, so you can use created Serial object directly (in example below) without create an instance by yourself.  
- In order to use more Serial port, you just create another instance of **HardwareSerial**such as: **HardwareSerial Serial1(1)**or **HardwareSerial Serial2(2)**and then use them as usual.

We will make a simple demo that print string “hello ESP32” every 1 second on Terminal. This demo is easy. To monitor “printed” data just go to **Tools > Serial Monitor**

**4. Result**

[](https://2.bp.blogspot.com/-JoKT7bR94EU/WRkSCzwlK-I/AAAAAAAADzw/xOcqXi9OXH46kGDMGTbtTdC0wNfN-9ZfACPcB/s1600/serial1.png)

**Figure: ESP32 using Serial to print debug information**