

Tutorial: Mega 2560 – Separate Serial vs Serial1 ("USB only" test)

This guide explains the exact wiring, upload, and verification steps for the sketch below so you can prove that Serial (USB/UART0) is independent from Serial1 (UART1). Use this when debugging multiple UARTs or USB-TTL adapters on an Arduino Mega 2560.

```
void setup() {  
    Serial.begin(9600);    // USB (UART0)  
    Serial1.begin(9600);   // UART1 (pins 19 RX1, 18 TX1)  
    delay(500);  
    Serial.println("USB only should be visible");  
}  
void loop() {  
    Serial.println("HIII");  
    delay(1000);  
    Serial1.println("THIS SHOULD NOT APPEAR ON USB");  
    delay(1000);  
}
```

1) What you'll learn

- How Serial (USB/UART0) is routed to your computer's /dev/ttyACM* port.
- How Serial1 (UART1 on pins 19/18) is not visible on USB unless you connect a separate USB-TTL cable.
- How to monitor two ports at the same time and confirm output separation.

2) Hardware & ports

- Board: Arduino Mega 2560

UARTs:

- Serial: UART0: pins 0 (RX0), 1 (TX0) : also the onboard USB interface shows up as /dev/ttyACM0 (Linux)
- Serial1: UART1: pins 19 (RX1), 18 (TX1) no automatic USB. Use a USB-TTL adapter to see it:/dev/ttyUSB0`

Important: For this tutorial, you can leave UART1 unwired. We are proving that its prints do not appear on USB.

3) IDE setup

1. Board: *Arduino Mega or Mega 2560*, Processor: *ATmega2560*
2. Port: select the Mega's USB port (e.g., /dev/ttyACM0).
3. Close all serial monitors before uploading.
4. Make sure nothing is connected to pins 0/1 during upload.

Linux tips:

- If permission errors: sudo usermod -aG dialout \$USER (log out/in) or temporary sudo chmod a+rw /dev/ttyACM0.
- If uploads hang, ensure ModemManager isn't grabbing the port: sudo systemctl stop ModemManager.

4) Upload the code

Upload the sketch above using the Mega's USB port. When it finishes:

- Open the Serial Monitor (or Serial Plotter) on /dev/ttyACM0 at 9600 baud.

You should see:

USB only should be visible

HIII

HIII

HIII

...

You should not see the line:

THIS SHOULD NOT APPEAR ON USB

Because that is printed to Serial1, which is not routed to the USB port.

5) Watch Serial1

If you connect a USB-TTL adapter to UART1, you can view Serial1 output too:

Wiring (Mega 2560 USB-TTL):

- GND to GND (Black)
- TX1 (pin 18) ↔ RX of adapter (White)
- RX1 (pin 19) ↔ TX of adapter (Green)
- Do not connect the adapter's 5V if the Mega is already USB-powered. (Red)

Then open another terminal/monitor on the adapter's port (e.g., /dev/ttyUSB0) at 9600 baud. You'll see:

THIS SHOULD NOT APPEAR ON USB

THIS SHOULD NOT APPEAR ON USB

...

Now you've proven the separation: Serial: /dev/ttyACM0,
Serial1: /dev/ttyUSB0.

6) Why the delay(500)?

On some systems, the USB virtual COM port takes a moment to enumerate and be ready. The short delay makes the first line (USB only should be visible) more likely to appear after the host is ready. It isn't strictly required, but it avoids "missing" the very first print.

7) Troubleshooting checklist

- Seeing both messages on the USB monitor?
 - Check for jumpers/shields accidentally linking pins 18/19 to 0/1.
 - Make sure your monitor is actually on /dev/ttyACM0 (the Mega), not the USB-TTL port.
- Upload timeouts (stk500v2_ReceiveMessage(): timeout)?
 - Close all serial monitors; select /dev/ttyACM0 (Mega) as upload port; ensure pins 0/1 are free; try a different USB cable/port.
- Nothing on /dev/ttyUSB0?
 - Verify adapter appears in dmesg; check TX/RX swapped; confirm common GND; set 9600 baud; ensure no other process holds the port.

8) Variations to try

- Change the Serial1.begin(9600) to a different baud and match your TTL monitor—proves baud rate matters.
- Comment out the Serial1.println(...) line—USB output continues unchanged.

- Swap to Serial2/Serial3 on the Mega (pins 16/17 and 14/15) to practice multiple UARTs.

9) Key takeaways

- On the Mega 2560, Serial goes to USB and pins 0/1; Serial1 is separate on pins 19/18.
- Without a USB-TTL adapter (or another MCU), Serial1 prints are invisible on the computer.
- Keep pins 0/1 free during uploads; open the correct port at the correct baud.