In some of my videos I mentioned that I used Putty for debugging of my Arduino or ESP projects. Some viewers asked, how this is done. This is the reason for this short video.

We are all aware of the very useful “Serial Monitor” function of our Arduino IDE. It helps us during debugging by printing some messages or values on the screen. For most cases, this is sufficient.

The Serial monitor connects through the USB cable to a chip on all arduinos. These chips transform the USB to Serial signals. These chips are always connected to RX and TX or RX0 and TX0 pins.

Sometimes, you need an additional serial connection and sometimes, you also want to monitor this second serial connection. For these situations, I will show you how this can be done.

Let us assume you want to connect a GPS module to an Arduino or an ESP. If you use an Arduino Mega, this is absolutely no problem. Because the Mega has four serial connections, you connect the GPS module to Serial1. Like that, you still can use your Serial monitor on Serial0.

If you want to use a UNO we only have one Serial connection available. Fortunately, we have a possibility to create a so called Software Serial on two other pins using a readily available library.

Now, we connect the GPS to the Software serial and we have the same situation as with the Mega. Just pay attention. Software serial is only rated to 57600 bps.

What do we have to do to monitor both connections in parallel? We can use a simple USB to serial adapter, sometimes also called FTDI adapters. These devices come in different shapes and sizes and have the same function as the chip on the Arduinos. If you use one of these, the most important decision is the voltage: If your processor board is 3.3 volt (as the ESP8266 or some of the newer Arduino boards) you need an adapter which is capable to provide this voltage. A 5v device can destroy your microcontroller. This is only important if you want to connect the RX pin of your microcontroller.

These FTDI adapters are also connected via a USB cable to your PC or directly into a USB hub. To connect them to your microcontroller, you connect the RX pin to the TX pin and vice versa. In addition, you have to connect ground. In most of the cases, we only want to use this adapter to monitor and not to enter data. Then, we can omit one connection.

But How can we now see what happens on the second serial connection? Serial monitor is only capable to show us one line. And usually, we keep it on Serial because the upload is also done through this connection.

So, we need a second Serial monitor. For that purpose I usually use Putty because it is simple and free. You can download it from putty.org and it does not need to be installed. Just click on the file and you see this window. I will not explain the different possibilities. There are other videos which do that. I just concentrate on debugging Arduinos or ESPs.

Let’s now return to the USB to serial adapter. If we connect it the first time to our PC it installs its driver and we find the connection in the device manager under “Ports (COM & LPT)”. Pay attention if you already have your Arduino or NodeMCU board connected to your computer. Then, you find this connection also in your device manager and you have to find out which is which. This can be done by disconnecting and reconnecting one of the two devices.

Now we know our port, which is COM plus a number. Attention: This port can change even with the same adapter if you reconnect it or if you boot your computer. So, always check before you start.

For our purpose, we only need to choose “serial connection” and enter the com port number from before. Speed 9600 is usually ok for monitoring. If we press “connect” we get a new window.

Now we can go to our Arduino or ESP board and write a small test sketch.

Because the ESP boards are slightly different, I use an NodeMCU board for my demonstration. The ESP8266 modules have two Serial connections. Serial1, however, only has one usable pin. Fortunately, this is the TX1 pin. So we do not need software serial if we only want to monitor.

On the NodeMCU board, TX1 is on D4 and we connect D4 to RX of the USB to serial connector. The sketch is very simple: We print “Hello World” to Serial and Serial1. To do that, we have to “begin” both serials in our setup. The speeds have to match the respective monitor, but can be different. And here we are. We see the messages on both channels.

You even can connect two USB to serial adapters if you need to monitor two traffic directions This culd be the case for Nextion displays where you see then the command from the display and the answer of the Arduino.

I hope, this video was useful or interesting for you. Bye.