USB TO 8CH TTL

From Waveshare Wiki
Jump to: navigation, search

> If you are using MacOS, please access via the WCHSerialPort software only; other access methods are not currently supported on MacOS

Overview

USB TO 8CH TTL



(https://www.waveshare.com/usb-to-8ch-tt.htm)

UART, TTL

Introduction

USB TO 8CH TTL, an industrial UART TO TTL converter with an aluminum alloy case, features the original CH348L chip and built-in protection circuits such as the self-recovery fuse and TVS. This converter is easier to operate due to its fully automatic transceiver circuit with no delay. Boasting characteristics of fast communication, stability, reliability, and safety, it is an ideal choice for various industrial control devices and applications with high communication requirements.

Features

- Extending USB to 8CH TTL (UART) with hardware flow control, allows concurrent and independent communication for each channel.
- Onboard resettable fuse and protection diodes, ensure the current/voltage stable outputs, provide over-current/over-voltage proof, improving shockproof performance.
- Onboard TVS (Transient Voltage Suppressor), effectively suppresses surge voltage and transient spike voltage in the circuit, lightningproof & anti-electrostatic.
- Onboard voltage translator circuit, with anti-surge and ESD protection, safe and stable communication.
- Onboard TTL serial 3.3V/5V voltage translator, config the TTL level via the switch.
- Aluminium alloy enclosure with oxidation dull-polish surface, CNC process opening, solid and durable, well-crafted.

- 4x external LEDs for indicating the power and transceiver status.

Parameters

Product	Industrial USB to TTL converter	
Host Interface	USB	
Device Interface	TTL (UART)	
USB Connector	Operating Level	5V
	Connector	USB-B
	Protection	200mA self-recovery fuse, ESD protection
Power Port	Connector	5V DC Power Port
	Protection	Over-voltage protection and anti-reverse protection
TTL (UART)	Connector	50PIN Anti-reverse Port
	Protection	TVS diode, surge protection & ESD protection
	UART TTL Level	3.3V/5V (Adjustable)
Indicator	PWR	Power indicator, connects to USB, lights up red when voltage is detected
	ACT	Status indicator, lights up green when the driver is detected
	TXD	TX indicator, lights up when the USB port sends data
	RXD	RX indicator, lights up when the device ports send data back
Operating System	Mac, Linux, Android, Windows 11/10/8.1/8/7	

Onboard Interface



(/wiki/File:USB_TO_8CH_TTL_Interface.jpg)

Indicator	
PWR	Power indicator, connect to USB, red light is on when the voltage is detected
ACT	Status indicator, green light is on when the driver is detected
TXD	Transmitting indicator, turn on when data is sent from the USB port
RXD	Receiving indicator, turn on when receiving data from the corresponding port

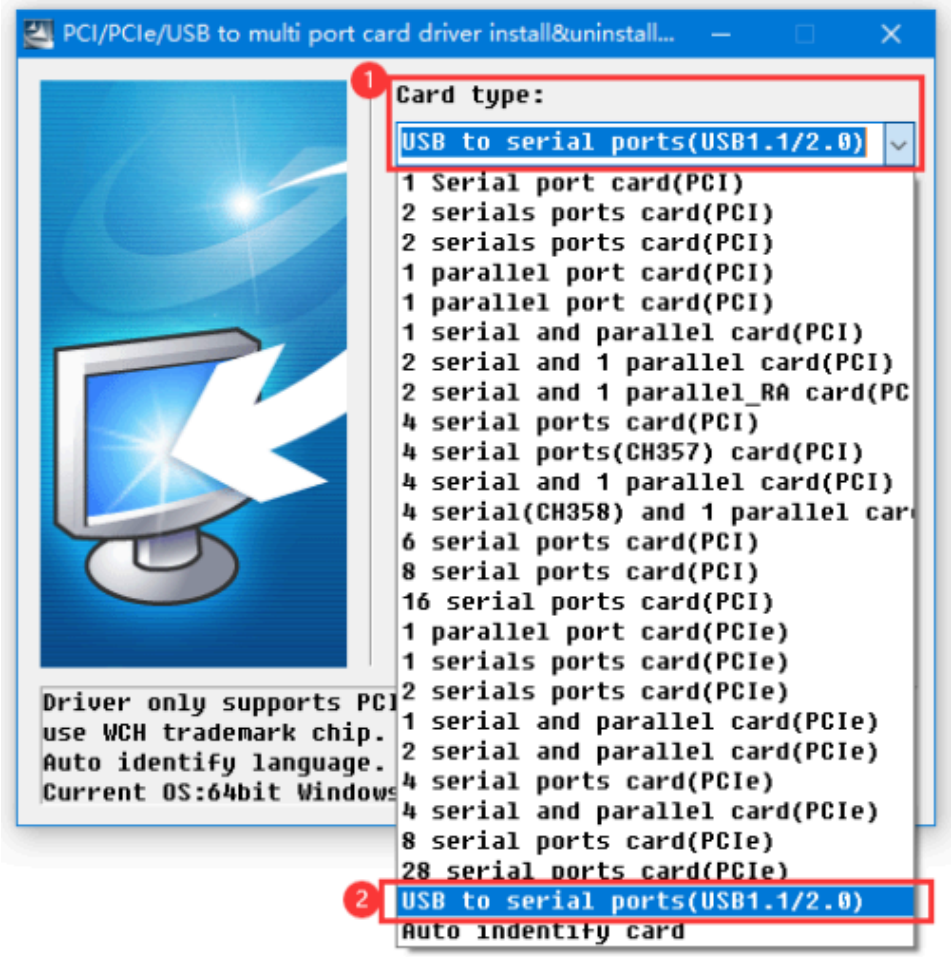
Dimensions



(/wiki/File:USB_TO_8CH_TTL_Dim.jpg)

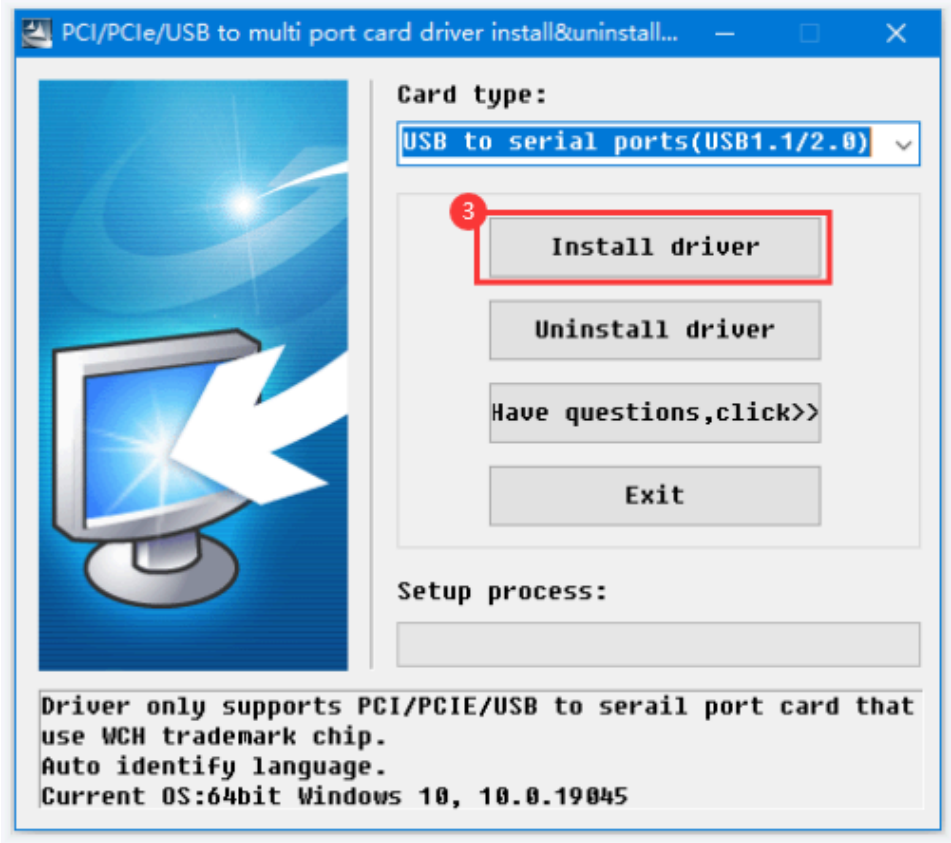
Driver Installation

- First, download the driver file USBMESR (https://files.waveshare.com/wiki/USB-TO-8CH-TTL/USBMSER.rar).
- Double-click **USBMESR.exe** and install it by steps.
- Select **USB to serial ports (USB 1.1/2.0)** as card type.



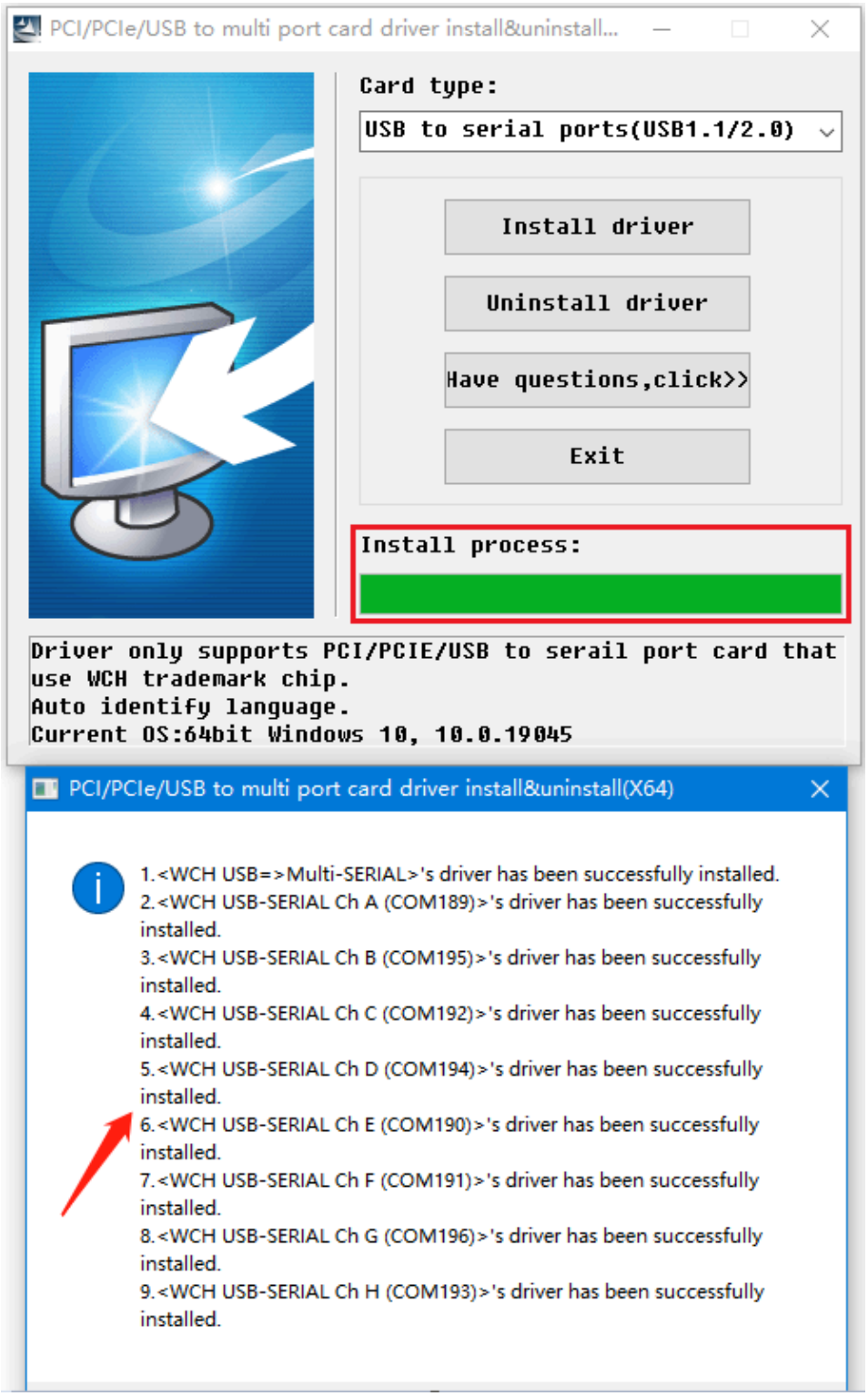
(/wiki/File:USB_TO_8CH_TTL_Driver_2.png)

- Click on **Install Driver**.



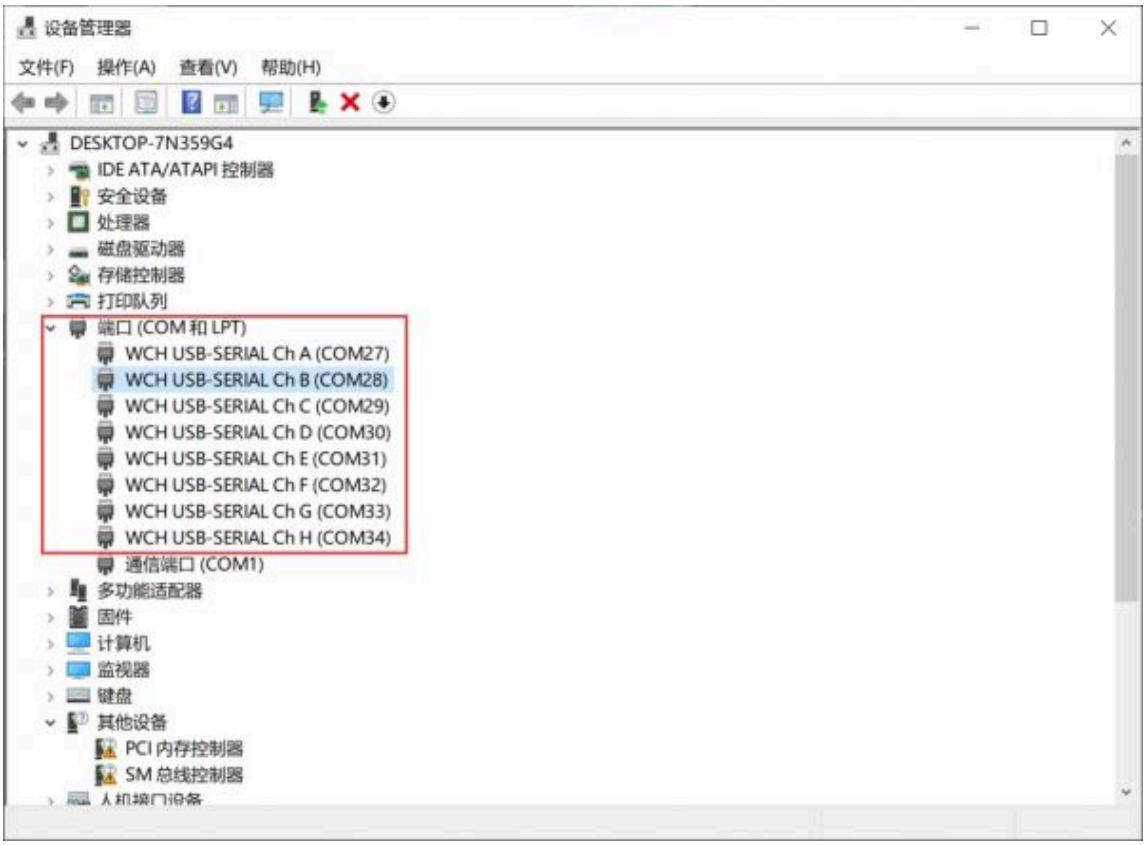
(/wiki/File:USB_TO_8CH_TTL_Driver_3.png)

- When the progress bar is loaded, a pop-up window will appear to indicate that the driver has been installed successfully.



(/wiki/File:USB_TO_8CH_TTL_Driver_4.png)

- After connecting to the computer, you can find the available ports and their corresponding COM port. (Also, you can assign the COM port number through the serial port manager tool).



(/wiki/File:USB_TO_8CH_TTL_Driver_5.jpg)

Communication Operation

- Open SSCOM software (https://files.waveshare.com/upload/2/20/Cktszsss32.zip).
- In general, the identification of 8 COM port numbers usually proceeds sequentially from Port A to Port H, starting from the smallest.

TTL Communication

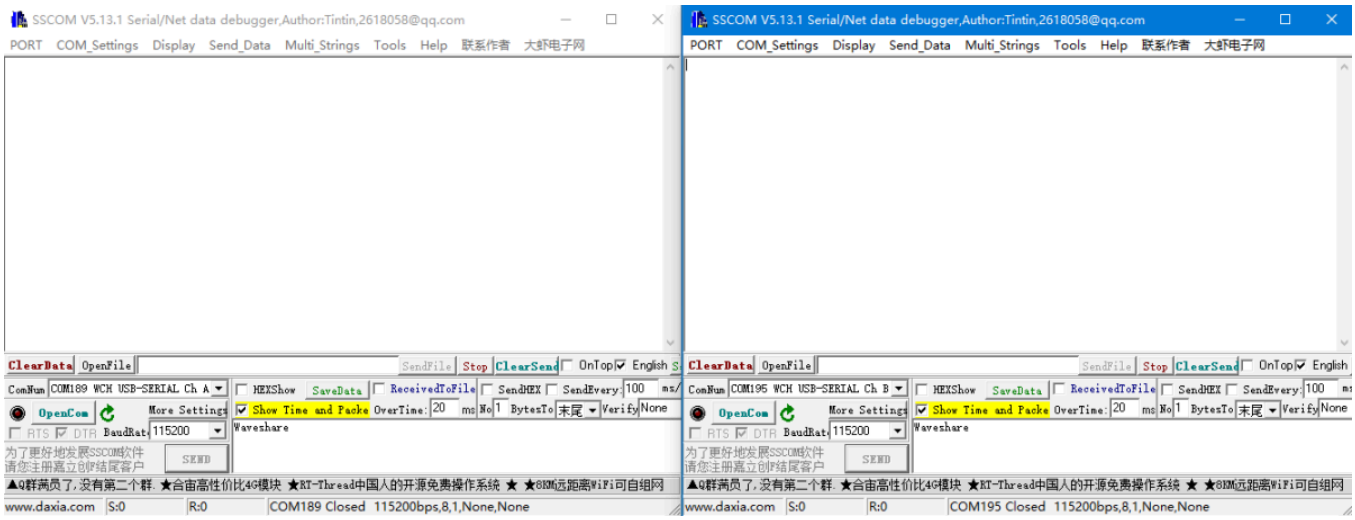
The following displays how to perform the communication between the UART 0 and UART 1 of the product.

Hardware Connection

USB TO 8CH TTL - UART 0	USB TO 8CH TTL - UART 1
UART 0 - TXD	UART 1 - RXD
UART 0 - RXD	UART 1 - TXD
UART 0 - GND	UART 1 - GND

Software Operation

- Open two SSCOM interfaces.
- Select the corresponding COM port of the UART0 and UART 1, respectively.



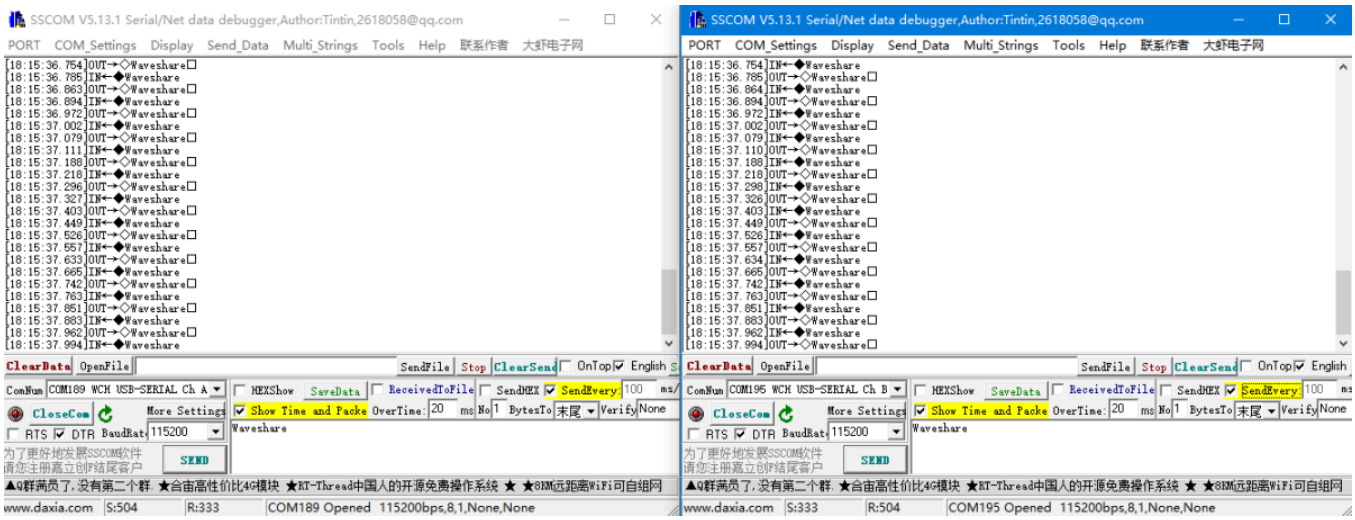
(/wiki/File:USB_TO_8CH_TTL_communication-1.png)

- Select the baud rate as **115200**, input the characters you want to send, check **Show time and packet**, and click on **Open COM**.



(/wiki/File:USB_TO_8CH_TTL_communication-2.png)

- Select 100ms intervals in two SSCOM interfaces, and you can see they transmit and receive data normally, the effects as shown below:



(/wiki/File:USB_TO_8CH_TTL_communication-3.png)

Resource

Datasheet

- CH348L Manual (https://files.waveshare.com/wiki/USB-TO-8CH-TTL/CH348DS1.PDF)

Software and Drivers

- SSCOM Assistant (https://files.waveshare.com/upload/2/20/Cktszsss32.zip)
- Windows USBMESR (https://files.waveshare.com/wiki/USB-TO-8CH-TTL/USBMSER.rar) driver (or download from WCH website (https://www.wch.cn/downloads/USBMSER_exe.html))
- ComPortManager (https://files.waveshare.com/wiki/USB-TO-8CH-TTL/ComPortManager.rar)
- Linux Driver (https://files.waveshare.com/wiki/USB-TO-8CH-RS485/LINUX.zip)

FAQ

Software Driver

Question: Why does the device appear as an unknown device?

Answer:

- USB cable failure – possible damage, poor contact, or charging-only cable. Replace the USB cable to troubleshoot.
- USB port issue – hardware failure or insufficient power, especially on older computers with aging USB ports and low voltage. Try a different port.
- Computer system driver issue – test with another computer or reinstall the system. It is also recommended to install the CH341SER driver (https://files.waveshare.com/wiki/common/CH341SER.zip).

Question: During communication, there is garbled data or no reply. How to solve it?

Answer:

- The product has an onboard level translation chip, and there is a data conversion rate issue. Therefore, during communication, avoid excessive data conversion in the middle of the communication process.

Question: After connecting the device, why does Linux not show a serial device descriptor (COM port number)?

Answer:

- This is a driver issue. Download and unzip the Linux driver (https://files.waveshare.com/wiki/USB-TO-8CH-TTL/USB-8CH-485-linux.zip).

```
wget https://files.waveshare.com/wiki/USB-TO-8CH-TTL/USB-8CH-485-linux.zip (https://files.waveshare.com/wiki/USB-TO-8CH-TTL/USB-8CH-485-linux.zip)
sudo unzip USB-8CH-485-linux.zip
cd ch9344ser_linux-main/driver
```

- Compile and load the USB device descriptor using:

```
sudo make install
```

- The figure below is tested on a Raspberry Pi with the 240704 64-bit Bookworm system (https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2024-07-04/2024-07-04-raspios-bookworm-arm64.img.xz). Different systems may vary; please modify the operations according to your actual situation:

```
pi@xl2025:~/Music/LINUX/driver $ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 1a86:55d9 QinHeng Electronics USB2.0 To Multi Serial Ports
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@xl2025:~/Music/LINUX/driver $ sudo make clean
rm -rf *.mk .tmp_versions Module.symvers *.mod.c *.o *.ko .*.cmd Module.markers modules.order *
.a *.mod
pi@xl2025:~/Music/LINUX/driver $ sudo make install
make -C /lib/modules/6.6.51+rpt-rpi-2712/build M=/home/pi/Music/LINUX/driver
make[1]: Entering directory '/usr/src/linux-headers-6.6.51+rpt-rpi-2712'
  CC [M]  /home/pi/Music/LINUX/driver/ch9344.o
  MODPOST /home/pi/Music/LINUX/driver/Module.symvers
  CC [M]  /home/pi/Music/LINUX/driver/ch9344.mod.o
  LD [M]  /home/pi/Music/LINUX/driver/ch9344.ko
make[1]: Leaving directory '/usr/src/linux-headers-6.6.51+rpt-rpi-2712'
insmod ch9344.ko || true
mkdir -p /lib/modules/6.6.51+rpt-rpi-2712/kernel/drivers/usb/serial/ || true
cp -f ./ch9344.ko /lib/modules/6.6.51+rpt-rpi-2712/kernel/drivers/usb/serial/ || true
depmod -a
pi@xl2025:~/Music/LINUX/driver $ ls /dev
autofs           gpiomem4      net      serial0     tty26  tty51          ttySC0   vcsu4
block            hwrng         null     shm         tty27  tty52          ttySC1   vcsu5
btrfs-control    i2c-1         port     snd         tty28  tty53          uhid     vcsu6
bus              i2c-11        ppp      spidev0.1   tty29  tty54          uinput   vcsu7
cachefiles       i2c-12        pps0     spidev10.0  tty3   tty55          urandom  vga_arbiter
cec0             initctl       ptmx     stderr      tty30  tty56          v4l      vhci
cec1             input         ptp0     stdin       tty31  tty57          vcio     vhost-net
ch9344_iodev0    kmsg          pts      stdout      tty32  tty58          vc-mem   vhost-vsock
char             kvm           ram0     tty         tty33  tty59          vcs      video19
console          log           ram1     tty0        tty34  tty6           vcs1     video20
cpu_dma_latency  loop0         ram10    tty1        tty35  tty60          vcs2     video21
cuse             loop1         ram11    tty10       tty36  tty61          vcs3     video22
disk             loop2         ram12    tty11       tty37  tty62          vcs4     video23
dma_heap         loop3         ram13    tty12       tty38  tty63          vcs5     video24
dri              loop4         ram14    tty13       tty39  tty7           vcs6     video25
fd               loop5         ram15    tty14       tty4   tty8           vcs7     video26
full             loop6         ram2     tty15       tty40  tty9           vcsa     video27
fuse             loop7         ram3     tty16       tty41  ttyAMA0        vcsa1    video28
gpiochip0        loop-control  ram4     tty17       tty42  ttyAMA10       vcsa2    video29
gpiochip10       mapper        ram5     tty18       tty43  ttyCH9344USB0  vcsa3    video30
gpiochip11       media0        ram6     tty19       tty44  ttyCH9344USB1  vcsa4    video31
gpiochip12       media1        ram7     tty2        tty45  ttyCH9344USB2  vcsa5    video32
gpiochip13       media2        ram8     tty20       tty46  ttyCH9344USB3  vcsa6    video33
gpiochip14       mem           ram9     tty21       tty47  ttyCH9344USB4  vcsa7    video34
gpiomem0         mmcblk0       random   tty22       tty48  ttyCH9344USB5  vcsu     video35
gpiomem1         mmcblk0p1     rfkill   tty23       tty49  ttyCH9344USB6  vcsu1    watchdog
gpiomem2         mmcblk0p2     rtc      tty24       tty5   ttyCH9344USB7  vcsu2    watchdog0
```

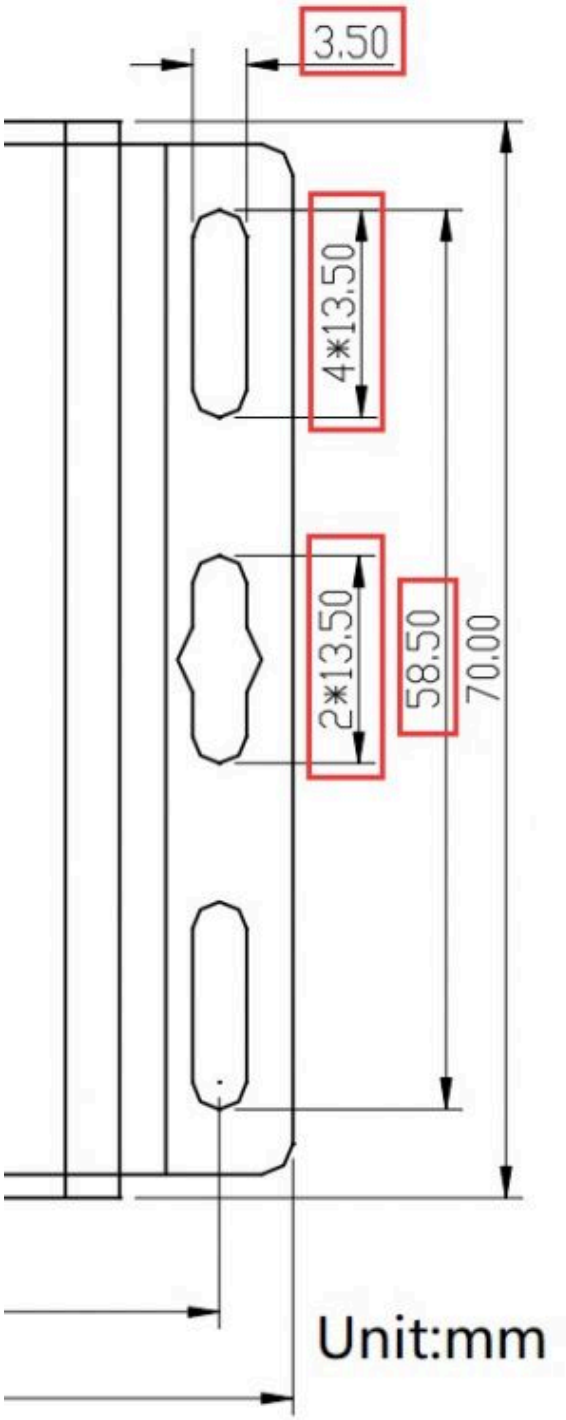
(/wiki/File:8-ttl-485-linux.png)

- It is recommended to use tools like cutecom to access the serial port; minicom may experience lag issues

Question: What is the size of the side mounting holes on this product?

Answer:

- The left and right side mounting holes are symmetrical and of the same size. The dimensions are shown below:



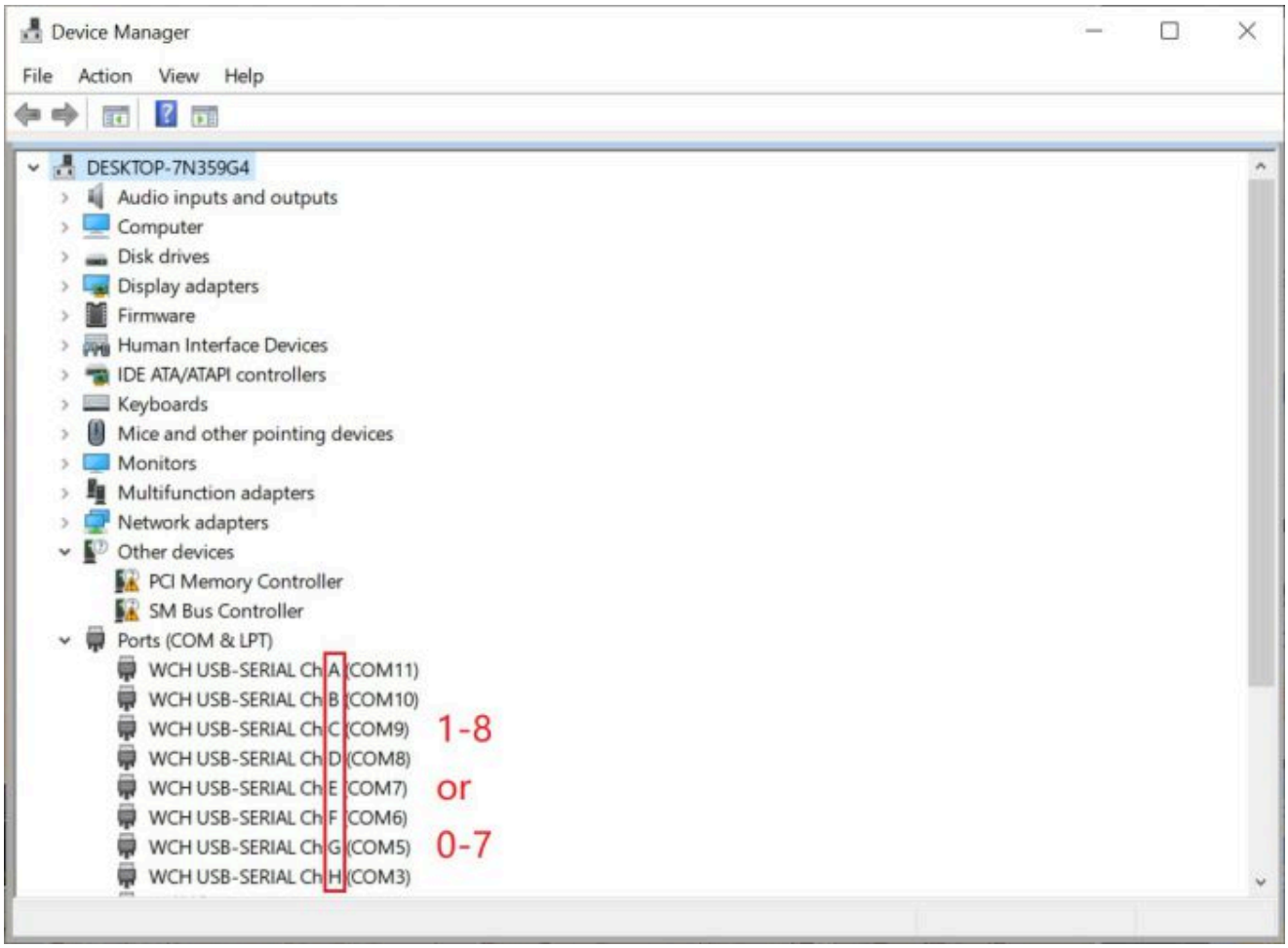
(/wiki/File:USB_TO_8CH_TTL_FAQ-4.jpg)

Question: How to check the port numbers after connecting the device to a computer?

Answer:

1. After connecting the device to the computer via USB, open Device Manager.

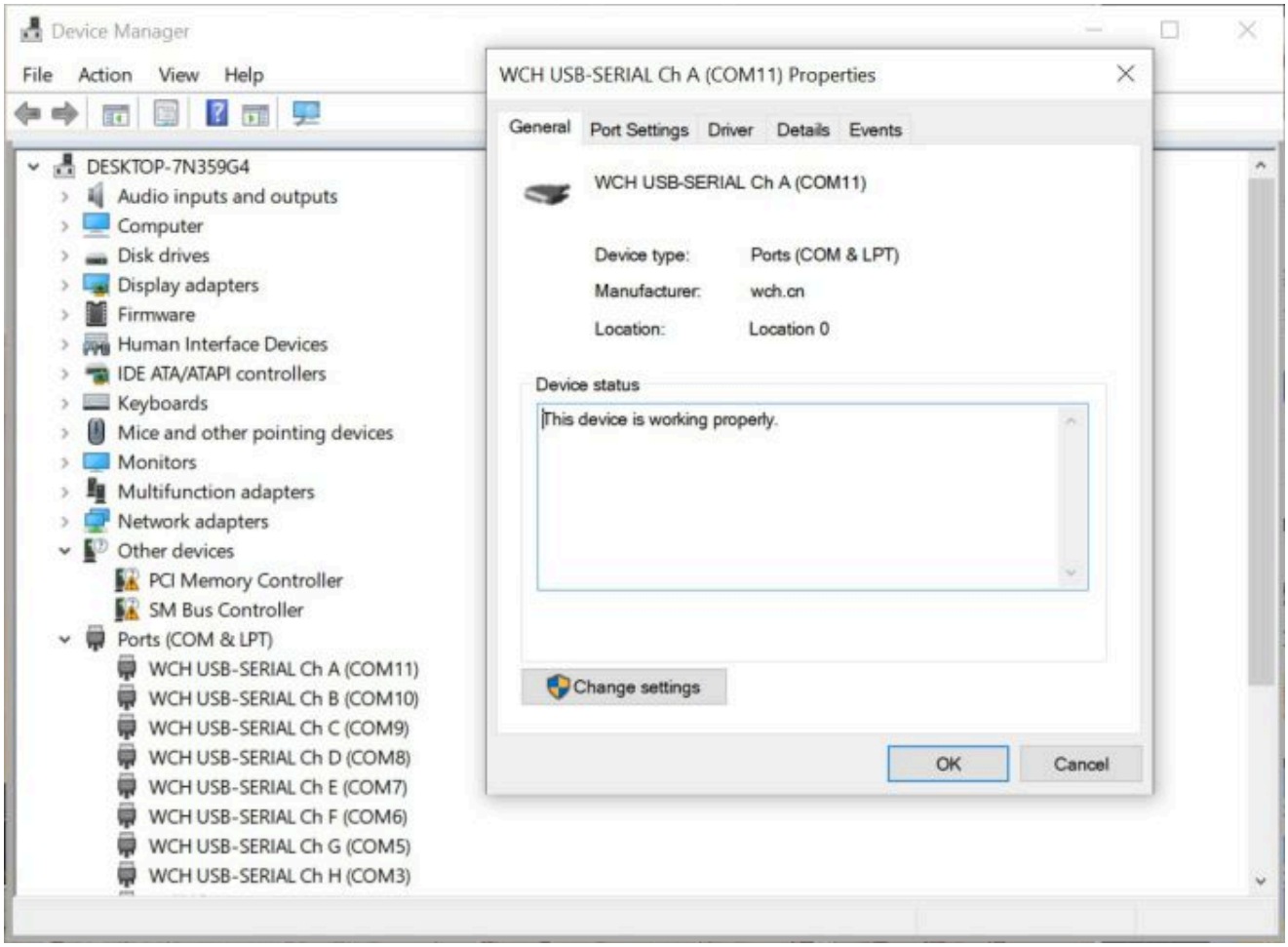
2. Under the "Ports" section, you will see the corresponding serial port numbers.



(/wiki/File:USB_TO_8CH_TTL_FAQ-5.jpg)

3. Right-click on the port you want to view and select "Properties".

4. After entering, you will see more status information about the port, as shown below.



(/wiki/File:USB_TO_8CH_TTL_FAQ-6.jpg)

Question: Does it support a baud rate of 921600 bps? The product page says the maximum is 2 Mbps.

Answer:

- The baud rates are not continuous. The following baud rates are supported:

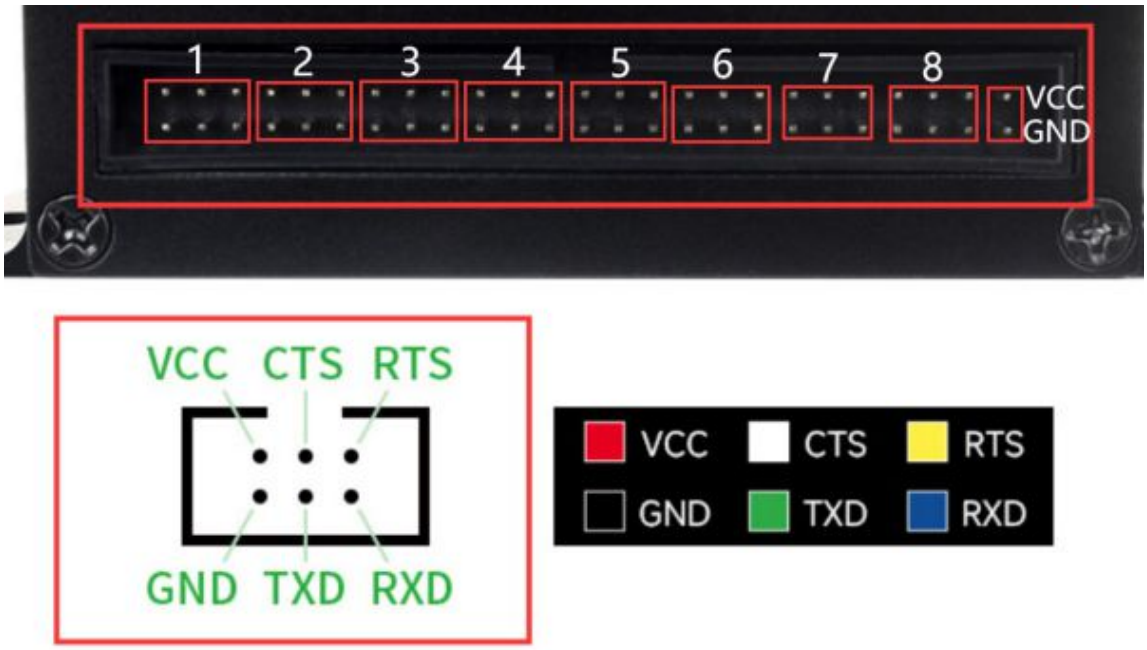
```
1200, 1800, 2400, 3600, 4800, 9600, 14400, 19200, 28800, 33600, 38400, 56000, 57600, 76
800, 115200, 128000, 153600, 230400, 460800, 921600, 1M, 1.5M, 2M
```

Hardware Interfaces

Question: The product interface connector is 50PIN. How do I distinguish the interfaces?

Answer:

- The interfaces are grouped as **2*3PIN** per serial port, repeating from left to right, for a total of 8 groups. The color corresponding to each interface matches the color of the 50PIN cable when connected. The specific identification is shown in the figure:



(/wiki/File:USB_TO_8CH_TTL_FAQ-1.jpg)

Question: In general, is USB power sufficient? When is DC power needed?

Answer:

- Generally, USB power is sufficient. For some older computers with insufficient USB power, DC power can be used.
- If multiple channels are used simultaneously, multiple devices are connected, and the communication distance is long, causing insufficient drive capability, DC power is needed. TTL power consumption is relatively low.

Question: For the USB TO 8CH TTL, the DC interface is 5V powered. Is there a current limit?

Answer:

- The current output of a 5V regulated power supply is not constant; it supplies whatever the load draws. It can be 0.5A, 1A, 2A, etc. There is no limit on the adapter's current rating. The module itself consumes less than 1A.
- USB itself can also provide sufficient power. If an external power adapter is used, a 5V / 1A specification is recommended, which can stably cover the full current demand of the module.
- Similarly, 5V phone chargers are also constant-voltage controlled; the current varies continuously: high current during fast charging at the beginning, medium current during

normal charging in the middle, and gradually reduced to a tiny trickle charge when approaching full charge.

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.
Working Time: 9 AM - 6 PM GMT+8 (Monday to Friday)

Submit Now (https://service.waveshare.com/)

Retrieved from "https://www.waveshare.com/w/index.php?title=USB_TO_8CH_TTL&oldid=110396 (https://www.waveshare.com/w/index.php?title=USB_TO_8CH_TTL&oldid=110396)"