
CP210x Troubleshooting Techniques

Question

I'm having a problem with my CP210x system. How do I find and resolve the issue?

Answer

Troubleshooting Utilities

UVCView:

Microsoft: <http://www.microsoft.com/whdc/device/stream/vidcap/UVCview.msp>

UVCView displays the USB information in a device (like VID, PID, Product String, descriptors, etc) con device does not need to be installed or loaded for UVCView to read the information, as the utility uses th device. This is useful for when a CP210x is not enumerating because of an incorrectly programmed PID

PortMon:

SysInternals: <http://www.sysinternals.com/Utilities/Portmon.html>

PortMon is a serial port monitoring program that records the transactions between the host application an useful for determining if invalid settings are accidentally being used with the CP210x or if the data is bei

CSDiff:

Component Software, Inc: <http://www.componentsoftware.com/Products/CSDiff/>

CSDiff is a file comparison program. This can be used to compare the input to the CP210x to the output

Terminal:

<http://bray.velenje.cx/>

Terminal is a basic serial communication program that can be used to connect to the CP210x VCP or the

SnoopyPro:

<http://sourceforge.net/projects/usbsnoop>

SnoopyPro is a USB software monitoring program that records the transactions between the CP210x driv driver. Much like PortMon, this program can be used to verify that the data is being properly sent to the

Suggested CP210x Troubleshooting Steps

In order for the CP210x to work properly:

- the CP210x has to enumerate properly on the USB bus
- the correct driver has to install
- the host application has to properly open, send data to, and close the handle to the CP210x

CP210x Enumeration

Run UVCView on the PC to verify if the CP210x properly enumerated. If all of the data in the selected C reported, then the device has enumerated. Here is an example of proper CP2101 enumeration:

The screenshot displays the 'USB Video Class Device Viewer' window. The left pane shows a tree view of USB devices, with 'CP210x USB Composite Device' selected under the first Intel USB Universal Host Controller. The right pane shows the following device information:

```

---->>Device Information<<----
English product name: "CP2101 USB to UART Bridge Controller"

ConnectionStatus:
Current Config Value:      0x01  -> Device Bus Speed: Full
Device Address:           0x01
Open Pipes:               2

***>>Device Descriptor<<***
bLength:                  0x12
bDescriptorType:          0x01
bcdUSB:                   0x0110
bDeviceClass:              0x00  -> This is an Interface Class Defined Device
bDeviceSubClass:          0x00
bDeviceProtocol:          0x00
bMaxPacketSize0:          0x40  = (64) Bytes
idVendor:                 0x10C4  = Cygnal Integrated Products, Inc.
idProduct:                0xEA60
bcdDevice:                0x0100
iManufacturer:            0x01
                           English (United States) "Silicon Labs"
iProduct:                 0x02
                           English (United States) "CP2101 USB to UART Bridge Controller"
iSerialNumber:            0x03
                           English (United States) "AbCd1234"
bNumConfigurations:       0x01

***>>Configuration Descriptor<<***
bLength:                  0x09
bDescriptorType:          0x02
wTotalLength:             0x0020  -> Validated
bNumInterfaces:          0x01
bConfigurationValue:     0x01
iConfiguration:          0x00
bmAttributes:             0x80  -> Bus Powered
MaxPower:                 0x32  = 100 mA

***>>Interface Descriptor<<***
bLength:                  0x09
bDescriptorType:          0x04
bInterfaceNumber:         0x00
bAlternateSetting:        0x00
bNumEndpoints:           0x02
bInterfaceClass:          0xFF  -> pre-release Video Interface Class
bInterfaceSubClass:       0x00
*!CAUTION: This appears to be an invalid bInterfaceSubClass
bInterfaceProtocol:       0x00
iInterface:               0x02
                           English (United States) "CP2101 USB to UART Bridge Controller"
*!ERROR: 0xFF is the pre-release USB Video Class ID

***>>Endpoint Descriptor<<***
bLength:                  0x07
bDescriptorType:          0x05
bEndpointAddress:         0x03  -> Direction: OUT - EndpointID: 3
bmAttributes:             0x02  -> Bulk Transfer Type
vMaxPacketSize:           0x0040 = 0x40 bytes
bInterval:                0x00

***>>Endpoint Descriptor<<***
bLength:                  0x07
bDescriptorType:          0x05
bEndpointAddress:         0x83  -> Direction: IN - EndpointID: 3
bmAttributes:             0x02  -> Bulk Transfer Type
vMaxPacketSize:           0x0040 = 0x40 bytes
bInterval:                0x00
    
```

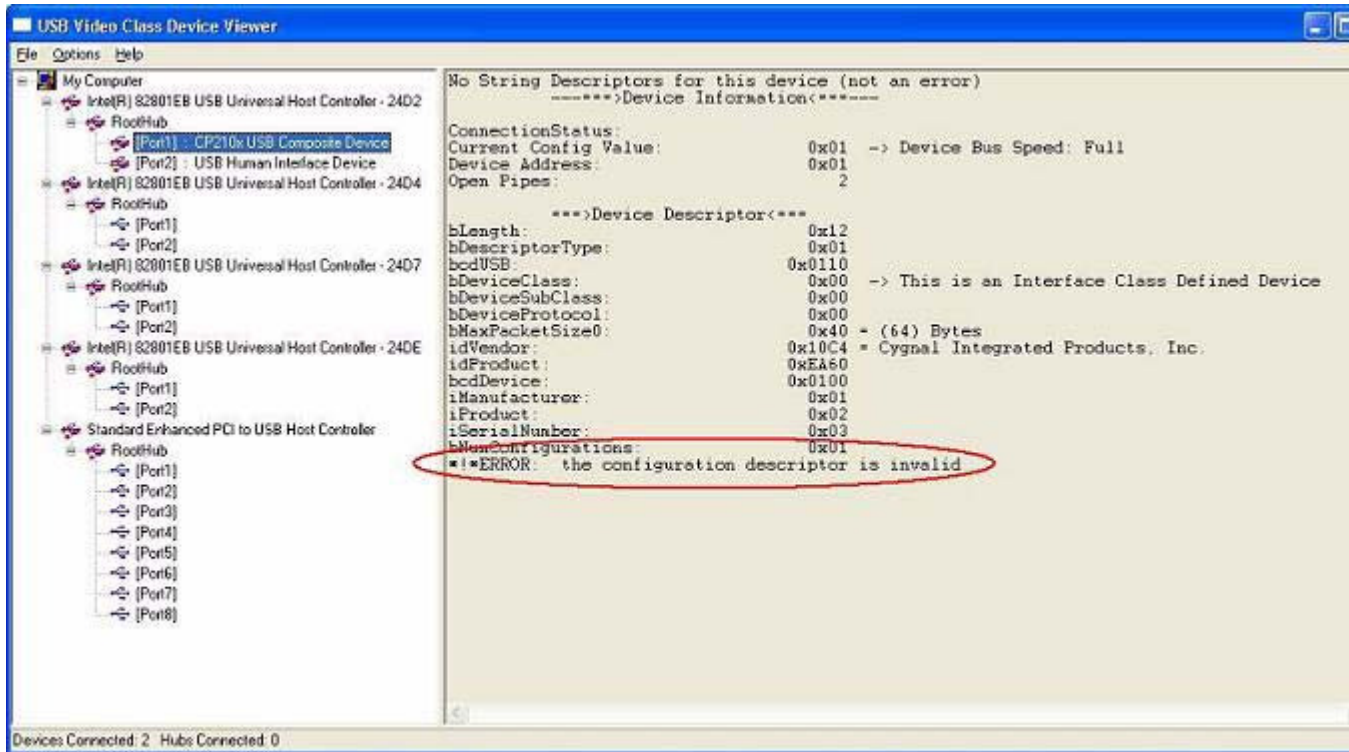
Annotations in the image include:

- Red circles around `idVendor: 0x10C4` and `idProduct: 0xEA60`, with a label "VID and PID".
- Red circles around `iSerialNumber: 0x03` and `English (United States) "AbCd1234"`, with a label "Serial Number String (AbCd1234)".
- Blue circles around `bInterfaceSubClass: 0x00` and `*!CAUTION: This appears to be an invalid bInterfaceSubClass`, with a label "This caution is okay".
- Blue circles around `*!ERROR: 0xFF is the pre-release USB Video Class ID`, with a label "This error is okay".

At the bottom left, a blue text box states: "The above caution and error are shown because UVCView was designed to check USB Video Class devices; They do not apply to CP210x devices."

However, if all the data is zeroes (0x0000, for example) or if an error is reported, the device did not prop

example of improper enumeration.



A possible cause for improper enumeration is soldering. Please see the attached “SolderingGuidelines” P

CP210x Driver Installation

If the CP210x driver is not installing, check:

- that any older CP210x drivers are uninstalled by going to Add or Remove Programs in the Control Panel to “UART Bridge Controller” BEFORE installing the latest driver
- that the CP210x VID and PID match the VID and PID of the driver using UVCView and the Application Customization software. Application Notes can be found on the Silicon Labs Applications webpage: <http://www.silabs.com/products/microcontroller/applications.asp>.
- that another driver of the same VID and PID is not already installed
- the PreInstaller has been run prior to connecting the CP210x. This is especially important for Windows (Windows will not automatically find the driver for the device if the device is connected before the driver is installed work again).

If the Add New Hardware wizard appears, simply hit Next several times to install the driver. If an error appears before attempting to install the driver again. If the driver is still not installing correctly, please contact MCU assistance.

CP210x Operation

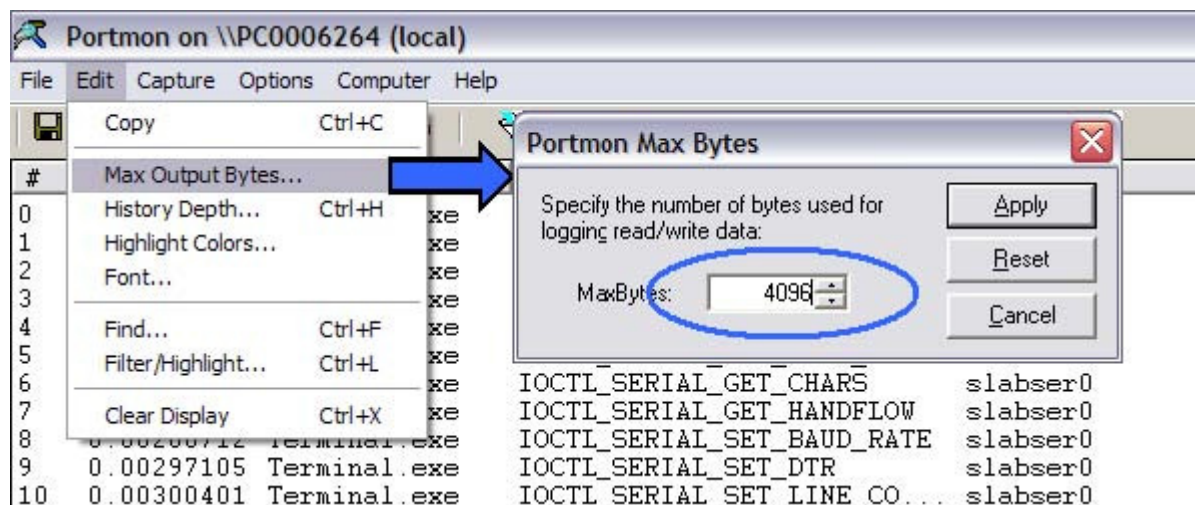
Once the CP210x hardware is properly enumerating and the CP210x driver is properly installed, the host can open, access, and close the CP210x handle. If the CP210x is not properly transferring data:

- Verify that the latest driver (contact MCU Tools at mcutools@silabs.com for the latest driver version

Manager -> right click on the driver and select Properties -> click the Driver tab. Also, verify that the app CP210x by checking the appropriate datasheet. For example, not all baud rates supported on the CP2101, the application may be attempting to use an invalid baud rate for a CP2101.

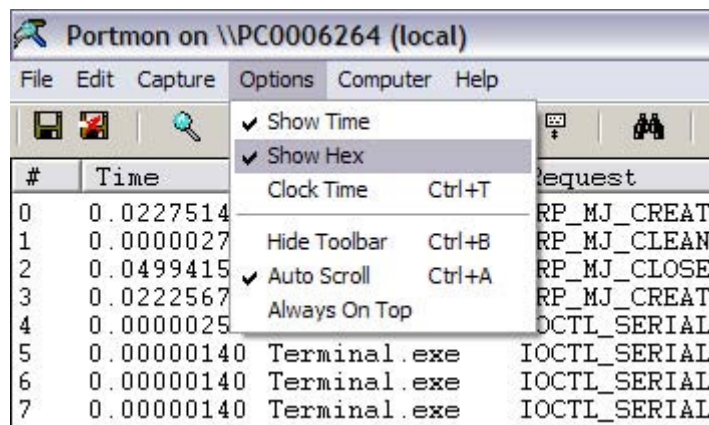
- In our testing, the CP210x device and driver do not typically drop data (the USB Bulk protocol guarantees that the application is losing data, how is the application handling the VCP? Is some form of handshaking being used (CTS/RTS, XON/XOFF)? Is the COM port being closed prematurely (before all data is sent out the COM port), the data being dropped (driver/USB bus/CP210x)?
- Use PortMon to view the transactions on the CP210x COM port. Are there any irregularities in the log immediately after writing the last data? Is some data not being transferred for one reason or another? Use the Tools for review. Check whether the application works with a real COM port. How do the real COM port logs look? CP210x PortMon logs?

Before generating any PortMon logs, change the default “Max Output Bytes” PortMon setting, as follows:

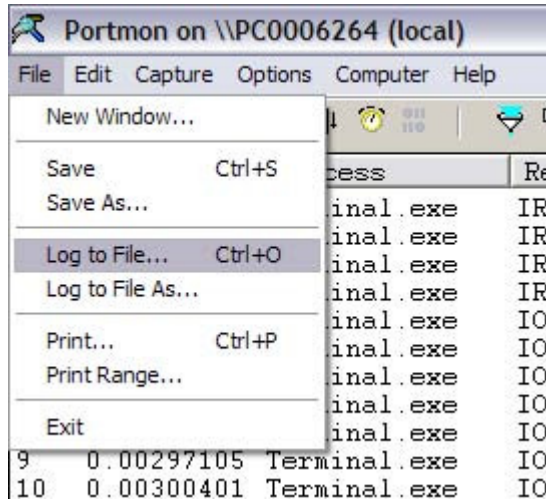


The maximum value for this setting is 4096. Changing this value will change the number of bytes that are read and write. By setting this to the maximum setting, it may be easier to see if data is not being sent.

Additionally, the data can be shown as hexadecimal values by checking the “Show Hex” option.



If the CP210x application is causing the PC to crash, then the “Log to File” option allows PortMon to save the log or received before the crash. This option is available in the File menu.



The PortMon logs display timestamps, driver and application names, IOCTL commands, and the data example section of a PortMon log displays the opening of a port. As shown in the log, the port is open stop bit, no parity, and hardware handshaking (Shake: 9... no handshaking is Shake: 1).

Request	Port	Result	Other
IOCTL_SERIAL_GET_LINE_CO...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_CHARS	slabser0	SUCCESS	
IOCTL_SERIAL_GET_HANDFLOW	slabser0	SUCCESS	
IOCTL_SERIAL_SET_BAUD_RATE	slabser0	SUCCESS	Rate: 115200
IOCTL_SERIAL_SET_DTR	slabser0	SUCCESS	
IOCTL_SERIAL_SET_LINE_CO...	slabser0	SUCCESS	StopBits: 1
IOCTL_SERIAL_SET_CHAR	slabser0	SUCCESS	EOF:0 ERR:0
IOCTL_SERIAL_SET_HANDFLOW	slabser0	SUCCESS	Shake:9 Repl
IOCTL_SERIAL_PURGE	slabser0	SUCCESS	Purge: TXABO
IOCTL_SERIAL_SET_QUEUE_SIZE	slabser0	SUCCESS	InSize: 2048
IOCTL_SERIAL_SET_TIMEOUTS	slabser0	SUCCESS	RI:1 RM:0 RC
IOCTL_SERIAL_CLR_DTR	slabser0	SUCCESS	
IOCTL_SERIAL_CLR_RTS	slabser0	INVALID PARAMETER	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IRP_MJ_READ	slabser0	TIMEOUT	Length 0:
IRP_MJ_READ	slabser0	TIMEOUT	Length 0:
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IRP_MJ_READ	slabser0	TIMEOUT	Length 0:
IRP_MJ_READ	slabser0	TIMEOUT	Length 0:
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IOCTL_SERIAL_GET_MODEMST...	slabser0	SUCCESS	
IRP_MJ_READ	slabser0	TIMEOUT	Length 0:
IRP_MJ_READ	slabser0	TIMEOUT	Length 0:

In this example, the IOCTL_SERIAL_CLR_RTS command receives an “INVALID PARAMETER”

IRP_MJ_READ commands return with “TIMEOUT” instead of “SUCCESS.” These types of return values may be occurring. Please contact MCU Tools for help with deciphering the PortMon logs.

- If the USB device is getting spurious resets or USB disconnects, adding a pull-up can help increase the internal pull-up of ~100 k Ω . To get better noise immunity, add a pull-up on the /RST pin as shown in