

See the Yellow Marked text below for installation of the drivers

RELEASE NOTES

=====

Silicon Labs Virtual COM Port (VCP) Universal Driver for Windows 10 version 1803 and later, windows 11

Package version: 11.4.0

Release date: 2024-12-19

This package contains a Virtual COM Port Universal driver for Microsoft Windows 10 version 1803 and later, windows 11 versions of Windows for use with Silicon Labs VCP USB Serial Bridges. These devices include the CP2102N, CP2102, CP2103, CP2104, CP2105, CP2108, CP2109.

OS VERSIONS

This package provides the CP210x VCP driver for the following versions of Windows:

- Windows 10 version 1803 and later (x64, x86)
- Windows 11 (x64)

CHOOSING VCP DRIVER VERSION

VCP drivers of version 11.1.0 and later are released for Microsoft Windows 10 version 1803 and later, windows 11 versions of Windows as Universal Drivers. This means they do not work on older version of Windows such as Window 7 and 8.1. You should use this driver if you only need support for Windows 10 version 1803 and later, windows 11, or you need the Universal driver. If you require support on older versions of Windows, you should use the legacy VCP driver which can be found on the Silicon Labs web site. The legacy Windows VCP drivers have version numbers less than 10.0.0.

CONTENTS

The VCP driver package contains the following files and directories:

- silabser.inf -- Driver information file
- SLAB_License_Agreement_VCP_Windows.txt -- Software License
- CP210x_Universal_Windows_Driver_ReleaseNotes.txt -- this file

Additional files needed by installer

- silabser.cat
- x64/silabser.sys
- x86/silabser.sys

INSTALLING

The VCP driver is automatically installed by Windows Update for CP210x with alternate PID values of 0xEA63, 0xEA7A, and 0xEA7B. These are not default values.

Otherwise, unzip the .zip file and you will have a directory containing the driver installer files.

Prompted install

If Windows prompts you to install a driver for a CP210x device (for example, if you plug a CP210x into your computer):

1. Use the dialog to browse for the driver location
2. Locate the driver folder (that you previously unzipped)
3. Follow the instructions

Manual install

1. Using Windows File Explorer, locate the driver folder (that you previously unzipped)
2. Right click on the silabser.inf file and select Install
3. Follow the instructions

UNINSTALLING

1. The device must be plugged in
2. Run Device Manager
3. Locate the device you want to uninstall
4. Right click on the device and choose "uninstall"
5. Follow the instructions

SETUP REGISTRY KEY

Registry keys are used for setting some parameters of the serial port
For setting these parameters, we have 2 ways:

- Modify registry value using registry editor of Windows
- Modify registry value using INF file

The detail of each method as following.

Modify registry value using registry editor of Windows

1. Edit the file UpdateParameters.reg according to your settings, save the file.
Registry keys are allocated at 3 position:
 - Parameters keys: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\silabser
 - Software keys: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\<ClassID>\
 - Hardware keys: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB\deviceID\
2. Right-click to UpdateParam.bat -> run as administrator, then your settings have been updated.
3. Restart you computer.

Modify registry value using INF file

1. To enable the registry key, Please remove character ';' to uncomment the statement
2. Registry key is allocated at 3 position:
 - Parameters keys: [silabser.AddService.AddReg] directives
 - Software keys: [silabser.AddReg] directives
 - Hardware keys: [SiLabsDDInstallSection.HW.AddReg] directives

SUPPORT

For technical support please visit the Silicon Labs support site:

<https://www.silabs.com/support>

RELEASE NOTES FOR 11.4.0 (2024-12-19)

Changes for VCP Driver

=====

- MCUSW-1436 | Fixed issue: Blue screen, Chip: CP2102N-A02-GQFN28R, baud: 3Mbaud
| software: Putty with debug log enabled, message: "ATTEMPTED_SWITCH_FROM_DPC".
- MCUSW-1437 | Fixed an issue about suspend mode behaves incorrectly when using
| CP210x Universal Windows Driver v11.3.0
- MCUSW-1464 | Microsoft Windows Certified ("WHQL Certified") for Windows 11
| version 24H2, Windows 10 21H2 and Windows 10 RS5.

RELEASE NOTES FOR 11.3.0 (2023-05-26)

Changes for VCP Driver

=====

- MCUSW-1386 | Fix bug in get Modem status stress test
- MCUSW-1404 | Change the .inf file to enable the purge buffer command
- MCUSW-1405 | Create the .bat and .reg files for the customer to easily update
| registry parameters VCP driver
- MCUSW-1395 | Microsoft Windows Certified ("WHQL Certified") for Windows 11
| version 22H2, Windows 10 21h2 and Windows 10 RS5.

RELEASE NOTES FOR 11.2.0 (2022-10-21)

Changes for VCP Driver

=====

- MCUSW-415 | Remove code paths for XP vs not XP which is used for
| UsingWindows2000.
- MCUSW-416 | Improve debug statement readability by adding device specific
| information.
- MCUSW-523 | add instruction how to setting up registry in .inf file.
- MCUSW-532 | Fix Automatic handshaking on RTS/CTS, and driver buffers
| overran at high baud rate.
- MCUSW-1285 | add DisableIoctl_IOCTL_SERIAL_PURGE in .inf file to prevent data
| lost in close.

RELEASE NOTES FOR 11.1.0 (2022-03-21)

Changes for VCP Driver

=====

- MCUSW-803 | CP210x device needs to send Xoff to the other side device
| (FTDI for example) when Windows goes to sleep
- MCUSW-1038 | Fix DfX problem on Windows 11
- MCUSW-1039 | Do HLK DfX testing for Windows VCP driver version 11.1.0
- MCUSW-1045 | Update file version and driver version for Windows VCP
| driver version 11.1.0
- MCUSW-1047 | Make the Windows VCP driver pass SDV
- MCUSW-1048 | Microsoft Windows Certified ("WHQL Certified") for Windows 11
| version 21H2
- MCUSW-1049 | Using Dynamic api to make Windows VCP driver to be compatible
| with different versions of Windows (Mininum required is Windows
| 10 version 1803)

RELEASE NOTES FOR 11.0.0 (2021-11-05)

Changes for VCP Driver

=====

- MCUSW-176 | Add report SERIAL_ERROR_QUEUEOVERRUN when driver-side receive
| buffer is overrun
- MCUSW-889 | Fixed HLK Test error: Static Tools Logo Test
- MCUSW-749 | Fixed issue that RTS is not set to high when Windows goes to
| sleep to make sure incoming data to CP210x won't be lost when
| Windows is in sleeping state
- MCUSW-841 | Removed installer to make a standalone driver package. Users
| need to use INF file to install the driver instead
- MCUSW-906 | Add Microsoft Windows Certified ("WHQL Certified") for Windows 11
| version 21H2

RELEASE NOTES FOR 10.1.10 (2021-01-13)

Changes for VCP Driver

=====

- MCUSW-651 | Support baud rate up to 2187500 for CP2102, CP2104, CP2109
- MCUSW-685 | Fixed the Blue Screen issue when running Windows Hardware Lab
| Kit for customized driver
- MCUSW-783 | Microsoft Windows Certified (aka "WHQL Certified") for Windows
| 10 version 20H2 (2020-10-20)

RELEASE NOTES FOR 10.1.9 (2020-07-30)

Changes for VCP Driver

=====

- MCUSW-566 | Remove sanity/error checking on IOCTL_SERIAL_SET_BAUD_RATE for
| CP2103 devices.
- MCUSW-596 | Microsoft Windows Certified (aka "WHQL Certified") for Windows
| 10 version 2004 (2020-05-27)
- MCUSW-619 | Support for Directed PoFx (DFx - Directed Power Management
| Framework).

RELEASE NOTES FOR 10.1.8 (2019-06-24 (DriverVer = 06/20/2019,10.1.8.2466))

Changes for VCP Driver

=====

- MCUSW-412 | Certified for Windows 10 RS6.
- MCUSW-484 | WDFVERIFY() added to the driver. Configurable via Registry.
- MCUSW-493 | Improved the tagging/tracking and parent-hierarchy of framework
| objects and framework object reference counts.
- MCUSW-495 | Clear RTS on CloseHandle().
- MCUSW-508 | IOCTL_SERIAL_GET_COMMSTATUS refactored to reduce USB-bus
| bandwidth, while still offering timely reporting.
- MCUSW-518 | WDFVerifier parameter ForceLogsInMiniDump Registry parameter
| added for improved supportability.
- MCUSW-519 | Resolve an SDV-identified issue: do not double-complete a
| Request on error paths.
- MCUSW-526 | Creates the framework USB device object with client contract
| version USB_CLIENT_CONTRACT_VERSION_602/0x602 (i.e. Register
| with "new" (to Windows 8) Windows USB driver stack).

RELEASE NOTES FOR 10.1.7 (2019-03-26 (DriverVer = 03/26/2019,10.1.7.2399))

Changes for VCP Driver

=====

- MCUSW-473 | Fix length of nul-terminated legacy Dos Name string written to
| HARDWARE\DEVICEMAP\SERIALCOMM Registry key.
- MCUSW-483 | Fixed bug for Max Baud rate for CP2101, CP2104, CP2105 and
| CP2108 devices.

RELEASE NOTES FOR 10.1.6 (2019-03-19 (DriverVer = 03/15/2019,10.1.6.2388))

Changes for VCP Driver

=====

- MCUSW-445 | Fix a blue-screen system crash bug with handling of ioctl
| IOCTL_SERIAL_WAIT_ON_MASK during power state changes.

RELEASE NOTES FOR 10.1.5 (2019-02-19 (DriverVer = 02/19/2019,10.1.5.2368))

Changes for VCP Driver

=====

MCUSW-414 | Implement an undocumented Registry-configured feature to ignore
| IOCTL_SERIAL_PURGE.

MCUSW-417 | Read many of the Registry "parameter"-type settings first from
| the "parameters key" and then from the "hardware key", with the
| "hardware key" taking precedence (allows "driver wide, all
| devices" setting, but provides the capability to specify per-
| device settings (before we only supported per-device
| settings)).
|

MCUSW-439 | Fixed a bug whereby Close()-Open() on a CP2102N could get USB
| data toggle confused (I.e. issue USB_CLEAR_FEATURE(HALT) to
| bulk endpoints to reset data toggle back to DATA0).

MCUSW-449 | Fix a Max Baud Rate bug (use device-specific MaxBauds (from the
| data-sheets): 3 Mbaud for CP2102N, 1 Mbaud for all other CP210x
| devices.).

MCUSW-450 | Fix Verification BSOD (race condition in the VCP device driver
| that can cause double completions): Only call
| DetachForCompletionUncancelable on success of UNmark as
| cancelable in handling of IOCTL_SERIAL_WAIT_ON_MASK-related
| requests.

RELEASE NOTES FOR 10.1.4 (2018-12-07 (DriverVer = 11/26/2018,10.1.4.2290))

Changes for VCP Driver

=====

MCUSW-134 | Fixed an issue with handling of IOCTL_SERIAL_SET_WAIT_MASK,
| IOCTL_SERIAL_WAIT_ON_MASK, with the SERIAL_EV_RXCHAR event and
| data arriving from the device into the driver. Previously, an
| indication of arriving data was made before making the data
| available, causing, in some conditions, subsequent requests for
| data after notification of data available to not deliver any
| data.

MCUSW-144 | Onecore-based editions of Windows 10 (I.e. Windows 10 IoT Core)
| do not, by design, create PortName entries in the Registry to
| specify the device's legacy DOS name (I.e. COM<n>).
| Consequently, VCP enhanced to not treat this as a failing
| condition and to continue without creating legacy COM<n> DOS
| devices.

MCUSW-148 | Rearchitected into two VCP driver packages:
| 1) Windows 7 and later Desktop Editions x86 and x64 driver
| package, and
| 2) Windows Universal (10 and later; Desktop, tablet, mobile,
| embedded) x86, x64, ARM, ARM64 driver package.

MCUSW-214 | Windows 10 Universal driver release (I.e. no Windows 7, 8 or
| 8.1 support)

MCUSW-352 | Workaround to issue where close during receipt of serial data
| caused CP2102N to hang.
MCUSW-369 | Certified for Windows 10 RS5.
MCUSW-393 | Fix bug due to some BaudRate-related math causing a divide-by-
| zero.
MCUSW-396 | Be defensive against the impossible case of getting a NULL
| object in the object completion routine.

RELEASE NOTES FOR 10.1.3 (2018-05-29)

Changes for VCP Driver

=====

MCUSW-326 | Clear DTR on CloseHandle().

RELEASE NOTES FOR 10.1.2 (2018-05-14)

Changes for VCP Driver

=====

MCUSW-134 | Fixed an issue with handling of IOCTL_SERIAL_SET_WAIT_MASK,
| IOCTL_SERIAL_WAIT_ON_MASK, with the SERIAL_EV_RXCHAR event and
| data arriving from the device into the driver. Previously, an
| indication of arriving data was made before making the data
| available, causing, in some conditions, subsequent requests for
| data after notification of data available to not deliver any
| data.

MCUSW-211 | Windows 10 Universal driver release (i.e. no Windows 7, 8 or
| 8.1 support)

MCUSW-288 | Fix IOCTL_SERIAL_GET_PROPERTIES.

MCUSW-291 | Fix IOCTL_SERIAL_GET_PROPERTIES/SERIAL_COMMPROP for CP210x-like
| "Voice MB v2.0" device.

MCUSW-315 | Certified for RS4/1803.

RELEASE NOTES FOR 10.1.1 (DriverVer=11/15/2017,10.1.1.1951)

Changes for VCP Driver

=====

MCUSW-134 | Fixed an issue with handling of IOCTL_SERIAL_SET_WAIT_MASK,
| IOCTL_SERIAL_WAIT_ON_MASK, with the SERIAL_EV_RXCHAR event
| and data arriving from the device into the driver. Previously,
| an indication of arriving data was made before making the data
| available, causing, in some conditions, subsequent requests
| for data after notification of data available to not deliver
| any data.

LEGACY HISTORY

Everything below is the shared development history before creation of the Windows 10 Universal driver. Everything above is development history of the Universal driver.

RELEASE NOTES FOR 6.7.5 (2017-10-13)

Changes for VCP Driver

=====

- MCUSW-17 | IOCTL_SERIAL_GET_STATUS TransmittedCount fixed.
- MCUSW-19 | CP210x_GET_CONFIG_DESCRIPTOR_IOCTL returned byte count fixed.
- MCUSW-21 | Revert IOCTL_SERIAL_GET_COMMSTATUS behaviour to return cached
| information (reverse slowdown reported with v6.7.4 in some
| applications)
|
| Improve windbg logging, w.r.t. doing less when logging is not
| enabled (say, Release builds), or when particular subsystem is
| not enabled for logging.
|
- MCUSW-110 | Windows 10 certified against RS2 (Creators Update).
- MCUSW-207 | In a failed initialization, discard a reference to dynamically
| allocated memory once memory has been freed.
- MCUSW-234 | timestamp SHA2 Signatures with SHA2 timestamps

RELEASE NOTES FOR 6.7.4.261 (2017/01/30)

- Device Guard compliant
- CP210XFW-747: Add 3 new parallel PIDs one each for the single-, dual- & quad-port CP210x devices (to be distributed via Windows Update)
- CP210XFW-741: Fix driver to use correct interface string index for the multi-interface CP210x devices (CP2105 & CP2108). Return iProduct string for other CP210x devices.
- CP210XFW-713: Add protection against IRP_MJ_CLOSE coming before IRP_MJ_CLEANUP. In such case EvtFileCleanup callback was called with 0 FileObject, causing a crash. Serenum is suspected of causing this, even though there is no proof except that it's present, was introduced recently and this crash was never registered before.
- CP210XFW-691: Fixed race between timer activation and request cancelation during processing of read rwquests.
- Changed raw WDF read request timer to our wrapper.
- CP210XFW-693: bugfix: WMI command "wmic path Win32_SerialPort get Availability^, DeviceID^, Name" is unable to enumerate the port.
- CP210XFW-686: EvtRequestCancel for wait mask IRP crashed because the request was already completed. It didn't serialize well with other places that can complete it. It looked like XOff IRP may have the same problem. This patch is adding an extra check into *all* other EvtRequestCancel functions to make sure this doesn't happen again.
- Changed IOCTLs returning polled-cached MDMSTS to explicitly read the

status. Use reg value DisableHwAccessInModemStatusIoctl to revert this new functionality to old behavior.

- Improved debug prints.
- SerialDbgPrintEx used 1K of stack for the string, reduced to 256.
- Resolved many static Code Analysis warnings
- Added accounting for data in chip's TX queue. Added translation of COMM_STATUS::ulErrors into SERIAL_EV_BREAK and SERIAL_EV_ERR.
- CP210XFW-549, CP210XFW-536, CP210XFW-546: Generate events for IOCTL_SERIAL_SET_WAIT_MASK from COMM_STATUS::ulErrors. Can be disabled by DisableCommStatusPolling reg value.
- Added spinlock around ModemStatus change detection. Refactored modem status event signalling to use common function.
- Move passive level functions into the paged section.
- MCUFW-740: Made device close operation wait for chip's Tx queue to empty. Added registry settings DisableTxEmptyCheckInClose and TxEmptyTimeoutOnCloseInMs.
- Improve control transfer debug prints and clean up code, otherwise no functional changes.
- MCUFW-740: Fixed EV_TXEMPTY to account for bytes in chip's Tx queue.
- MCUFW-740: New functions to sync access to comm error storage from multiple threads.
- Added DisableTxEmptyCheckInGetCommStatusIoctl reg value.
- Refactor control request I/O. MCUFW-740: Fixed IOCTL_SERIAL_GET_COMMSTATUS.
- MCUFW-740: Made device close operation wait for chip's Tx queue to empty. Added registry settings DisableTxEmptyCheckInClose and TxEmptyTimeoutOnCloseInMs.
- CP210XFW-648: Set a three second timeout on WdfUsbTargetDeviceSendControlTransferSynchronously().

RELEASE NOTES FOR 6.7.3 (2016/03/03)

- Down-rev the version of KMDF the driver uses (CP210XFW-591. Use version of KMDF natively included on Windows release)
- Bug fix of blue-screen system crash (CP210XFW-575. driver tries to clear a Cancel Routine in a Request that doesn't have it.)

RELEASE NOTES FOR 6.7.2 (2015/11/23)

- Unified Windows 7, 8, 8.1 & 10 support (XP, Vista and Server 2003 available in version 6.7).
- ARM support added for Windows 10 and Windows 10 IoT Core.
- Silicon Labs-specific Serial Enumeration driver silabenm removed in favor of the Windows "in-box" serenm.
- UpperFilter Serial Enumeration driver not enabled by default
- Updated to latest Microsoft build tools
- Resolved /W4 compilation warnings
- Resolved code analysis warnings
- Added a registry value ControlHandShakeForcedSetBits that can force certain bits in the ControlHandShake variable to always be set.
- Add SAL-annotations to API header files.

- Bugfix an invalid memory access on device stop.
- Standardize branding Properties resources across all deliverables.
- Digitally sign all signable deliverables.
- Rewrote driver synchronization for data transmission paths.
- Resolve bugs arising from extended stress test.
- Added diagnostic variables to device extension.
- Bugfix a Write IRP stuck forever causing bugcheck 9f-4 if call to USB returns with error.
- Bugfix "packetized write" timeout may fail to act. And improved debug prints.
- New reg value DisableS0Idle (default: 0) to replace EnablePowerManagement (default: 0 which actually meant "enable"). Also, DisableS0Idle=1 now forces AllowIdleWhileOpen to 0.
- Driver ignores BreakOnEntry, DebugLevel and DebugFlag in registry.
- New CP2108 bug workarounds - GET_LINE_CTL byte swap and tx_empty().
- Fixed crash in read path.
- Simplified, commented and synchronized with NEWSTUFF EvtIoStop and EvtIoResume.
- Fixed memory leak on device removal when multiple devices are present.
- Bugfix Read IRP may never complete if other read IRPs are already waiting when it arrives.
- Add CP2102N part numbers. Handle new part numbers in WriteLatch()/ReadLatch().
- Resolve HCK INF warnings that became errors on HLK.
- Bugfix: Potential invalid memory access on device stop.
- Fixed possible memory overrun when writing CP2108 latch.
- Cumulative collection of improvements (due to code analysis, application of latest toolset, etc.) and bug fixes.

RELEASE NOTES FOR 6.7.1 (2015/06/29)

- Added certification for Windows 10 Preview (Note: Windows 10 only; use v6.7 for Windows prior to Windows 10)

RELEASE NOTES FOR 6.7 (2014/04/11)

- Added vendor-specific commands to the driver that do not affect normal operation with standard CP210x devices.
- Modified write behavior to packetize data as a work around for a rare USB 2.0 hub behavior seen only in Windows XP (x86).

RELEASE NOTES FOR 6.6.1 (2013/10/24)

- Added certification for Windows 8.1
- Added support for CP2108 devices
- Corrected bug where OS would hang on repetition of many rapid open/closes on certain systems.

RELEASE NOTES FOR 6.6

- Moving installer format to DPInst
- Supported under Windows 8
- Corrected bug in total write timeout calculation
- Corrected bug where EV_BREAK event could be misreported based on incoming data
- Corrected another BSOD corner case regarding write cancellation

RELEASE NOTES FOR 6.5.3

- Corrected BSOD corner case for a write cancellation
- Corrected memory leak that occurs when a device is plugged/unplugged rapidly enough to allow driver startup function to fail

RELEASE NOTES FOR 6.5

- Corrected BSOD corner case issue for handling control requests
- Corrected issue where USB 3.0 hubs would deliver data to the driver out of order

RELEASE NOTES FOR 6.4

- Corrected multiple BSOD issues by updating request handling for control requests to the USB device, and the write completion method
- Corrected device removal routine to also undo the COM port naming in the registry at removal time instead of when the device is closed after the removal.

RELEASE NOTES FOR 6.3a

- Updated to 3.2 DriverInstaller to correct a bug seen when updating from an old driver installer version

RELEASE NOTES FOR 6.3

- Corrected a bug introduced in 6.2 which caused BSOD during surprise removal.
- Corrected a bug found in I/O cancellation corner case

RELEASE NOTES FOR 6.2

- Corrected return value for DeviceIoControl() and other COM API functions once the device is removed to properly show ERROR_ACCESS_DENIED (0x05) instead of ERROR_BAD_COMMAND in GetLastError()
- Corrected bug in DTR/RTS reporting latency seen if you set DTR/RTS and ask for it before the device has reported it to the host - now when it is written it reads back immediately
- Corrected several USB requests that were improperly defined as device requests when they are actually interface requests - this doesn't exhibit and bugs in current single interface devices but was changed to be complete and correct
- Added support for CP2104 and CP2105 devices

RELEASE NOTES FOR 6.1

- Corrected a memory leak which could bog the system down after extended use of a continuously opened COM port, and conditionally yield bluescreens on certain systems
- Corrected a problem where an IO request would sometimes return a busy status to user mode, instead the queue is restarted if necessary before adding an IO request to the queue
- Corrected a condition which would blue screen on cancelling write request that hasn't been fully sent out USB
- Corrected the Capabilities return value, which incorrectly reported that timeouts are not supported
- Corrected several Queue size return values, which affected behavior in the MSCOMM control
- Corrected DTR/RTS value on device insertion, visible when Serial Enumeration is disabled
- Corrected the ability to override Baud Rate and Line Control from the INF file

RELEASE NOTES FOR 6.0

- Corrected multiple blue screens and driver hangs related to race conditions in the driver
- Corrected problem where driver hangs when 4 or more devices are connected to a single transaction translator hub
- Corrected bugs that prohibited serial enumeration
- Corrected problem where IO requests were not completed/cancelled on a close
- Corrected problem with dialing out or PPP connections
- Created IO queueing mechanism so that multiple reads, writes, etc. can be queued and waited on