USBXpress™ Family
CP2102N Errata

This document contains information on the CP2102N errata. The latest available revision of this device is revision A02.

For errata on older revisions, refer to the errata history section for the device. The revision information is typically specified in or near the trace code on the device. Refer to the package marking information in the data sheet for more information.

Errata effective date: March, 2019.

1. Active Errata Summary

These tables list all known errata for the CP2102N and all unresolved errata in revision A02 of the CP2102N.

Table 1.1. Errata History Overview

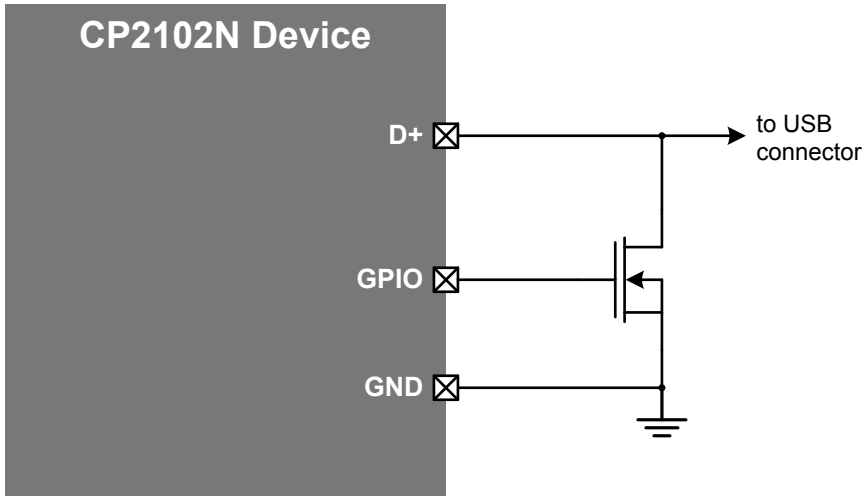
Designator	Title/Problem	Exists on Revision:		
		A01	A01 devices with a date code earlier than 1639	A02
CP2102N_E101	Failure to Power On or Reset	—	X	—
CP2102N_E102	Race Condition on Error Flags	—	X	—
CP2102N_E103	Self-Powered Device May Not Enter Suspend	—	X	—
CP2102N_E104	IO Exception in .NET Applications when Manually Controlling RTS	X	X	—
CP2102N_E105	RS485 DE Signal Not Held Long Enough	X	X	—
CP2102N_E106	DTR/RTS behavior when port is closed	X	X	—
CP2102N_E107	SUSPEND and SUSPENDb state during enumeration	X	X	—
CP2102N_E108	Failure to Enumerate	X	X	—
CP2102N_E109	Failure to Enumerate on Windows 7 when the String Descriptors are multiples of 64 bytes	X	X	—
CP2102N_E110	USB D+/D- Power-On Reset	X	X	X

Table 1.2. Active Errata Status Summary

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
1	CP2102N_E110	USB D+/D- Power-On Reset	Yes	A02	—

2. Detailed Errata Descriptions

2.1 CP2102N_E110 – USB D+/D- Power-On Reset

Description of Errata
During and immediately following power-on reset, the USB D+ and D- pins are undefined. These pins can be logic high, logic low, or mid-supply. This behavior typically lasts for ~15 ms, at which point the D+ and D- pins operate normally. The USB specification provides a 100 ms settling/debounce period after power-on.
Affected Conditions / Impacts
Some USB host devices may require the USB D+ and D- pins be in a known state during or immediately following power-on reset to start properly.
Workaround
Adding an external circuit to the USB D+ and D- pins can force the desired state on these pins during and immediately following power-on reset. The rising edge of RSTb signals the end of power-on reset. An example circuit is shown below:    This circuit uses an NMOS transitor and a GPIO pin to short D+ to ground during power-on. The GPIO pin will be open drain and pulled high through a weak pull-up during reset. The GPIO should be configured so that the [**Reset Latch**] value is [**Low**], which turns off the transistor after reset and enables the device to enumerate normally. GPIO can be configured using Xpress Configurator in Simplicity Studio v4 (www.silabs.com/simplicity).
Resolution
There is currently no resolution for this issue.

3. Errata History

This section contains the errata history for CP2102N devices.

For errata on latest revision, refer to the beginning of this document. The device data sheet explains how to identify chip revision, either from package marking or electronically.

3.1 Errata History Summary

This table lists all resolved errata for the CP2102N.

Table 3.1. Errata History Status Summary

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
1	CP2102N_E101	Failure to Power On or Reset	Yes	A01 devices with a date code earlier than 1639	A02, A01 devices with a date code of 1639 or later
2	CP2102N_E102	Race Condition on Error Flags	No	A01 devices with a date code earlier than 1639	A02, A01 devices with a date code of 1639 or later
3	CP2102N_E103	Self-Powered Device May Not Enter Suspend	No	A01 devices with a date code earlier than 1639	A02, A01 devices with a date code of 1639 or later
4	CP2102N_E104	IO Exception in .NET Applications when Manually Controlling RTS	Yes	A01	A02
5	CP2102N_E105	RS485 DE Signal Not Held Long Enough	Yes	A01	A02
6	CP2102N_E106	DTR/RTS behavior when port is closed	Yes	A01	A02
7	CP2102N_E107	SUSPEND and SUSPENDb state during enumeration	Yes	A01	A02
8	CP2102N_E108	Failure to Enumerate	Yes	A01	A02
9	CP2102N_E109	Failure to Enumerate on Windows 7 when the String Descriptors are multiples of 64 bytes	Yes	A01	A02

3.2 Detailed Errata Descriptions

3.3 CP2102N_E101 – Failure to Power On or Reset

Description of Errata
Devices can intermittently fail to power-on or reset properly. When this failure occurs, devices will hang indefinitely and not enumerate or respond until the next power-on cycle.
Affected Conditions / Impacts
Systems using the CP2102N may see devices fail to respond until a power-on reset.
Workaround
If a device fails to respond properly, remove and replace power until the device properly responds.
Resolution
This issue is resolved in revision A01 devices with a date code of 1639 or later and A02 devices.

3.4 CP2102N_E102 – Race Condition on Error Flags

Description of Errata
Devices can fail to notify the host of an error flag if an error occurs while the host is reading the UART status.
Affected Conditions / Impacts
The following error conditions may be missed if they occur while the host is reading the UART status: • Set Break • Hard overrun • Queue overrun • Parity error
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision A01 devices with a date code of 1639 or later and A02 devices.

3.5 CP2102N_E103 – Self-Powered Device May Not Enter Suspend

Description of Errata
A device in the self-powered configuration may not enter Suspend mode properly if the USB host is disconnected (i.e. cable unplugged).
Affected Conditions / Impacts
A device may draw additional current on the order of Normal Operation mode (~10 mA) when not connected to USB and in the self-powered configuration.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision A01 devices with a date code of 1639 or later and A02 devices.

3.6 CP2102N_E104 – IO Exception in .NET Applications when Manually Controlling RTS

Description of Errata
The CP2102N uses the incorrect byte of the SERIAL_HANDFLOW structure (https://msdn.microsoft.com/en-us/library/windows/hardware/jj680685(v=vs.85).aspx) to control the RTS signal. Instead of looking at the first byte of FlowReplace, the device is reading the first byte of the XonLimit and interpreting that as the first byte of FlowReplace. Applications written in .NET set the Xon/Xoff limits to 160, equal to 0xA0, which the CP2102N interprets as hardware flow control, and so it returns an error when manually setting RTS.
Affected Conditions / Impacts
Applications written in .NET will see IO exceptions when attempting to manually set or clear RTS on the CP2102N.
Workaround
For .NET applications, it is possible to create a software workaround for this issue by setting the XON/XOFF limits to 0x00. Because of additional .NET limitations, the workaround also has to enable hardware flow control, enable the port, then switch to none. These changes enable the CP2102N to properly set or clear the RTS signal manually. An example demonstrating this can be found in the following Knowledge Base article: https://www.silabs.com/community/interface/knowledge-base.entry.html/2017/11/10/cp2102n_e104_io_ex-YqAX.html If the serial port is configured in C++, directly set the DCB XON/XOFF limits to 0x00 as a workaround for this issue.
Resolution
This issue is resolved in A02 devices.

3.7 CP2102N_E105 – RS485 DE Signal Not Held Long Enough

Description of Errata
The CP2102N RS485 signal is not held long enough.
Affected Conditions / Impacts
Devices configured in RS485 mode will not communicate properly.
Workaround
The CP2102N has support for RS485 Hold Time configuration. To work around this issue, add 2 bits of Hold Time to the device manually using Xpress Configurator in Simplicity Studio (www.silabs.com/simplicity).
Resolution
This issue is resolved in A02 devices.

3.8 CP2102N_E106 – DTR/RTS behavior when port is closed

Description of Errata
CP2102N does not reset DTR/RTS when port is closed.
Affected Conditions / Impacts
When a USB host closes the CP2102N's virtual comm port, the DTR and RTS flow control signals will remain in the same state as they were before port closure, instead of being deactivated.
Workaround
Applications interfacing with the virtual comm port must manually deactivate DTR and RTS signals through comm APIs unless the installed host driver performs this step automatically. The Windows 10 VCP Universal driver version 10.1.3 and later and Windows VCP driver version 6.7.6 and later automatically deactivate DTR and RTS when the port is closed.
Resolution
This issue is resolved in A02 devices.

3.9 CP2102N_E107 – SUSPEND and SUSPENDb state during enumeration

Description of Errata
SUSPEND and SUSPENDb will not assert until after the device completes a USB configuration.
Affected Conditions / Impacts
CP2102N's SUSPEND and SUSPENDb pins deassert during power on for 3ms, instead of asserting until the USB enumeration process completes.
Workaround
Embedded devices monitoring SUSPEND or SUSPENDb state should delay any actions based on SUSPEND and SUSPENDb signals until signal state persists for at least 3 ms.
Resolution
This issue is resolved in A02 devices.

3.10 CP2102N_E108 – Failure to Enumerate

Description of Errata
The CP2102N may fail to enumerate.
Affected Conditions / Impacts
When connecting the CP2102N to a USB host, the CP2102N may fail to enumerate.
Workaround
If a device fails to enumerate, disconnect and reconnect the device to the USB host until the device successfully enumerates.
Resolution
This issue is resolved in A02 devices.

3.11 CP2102N_E109 – Failure to Enumerate on Windows 7 when the String Descriptors are multiples of 64 bytes

Description of Errata
On Windows 7 host PCs, the OS may fail to properly detect the CP2102N. This is caused by a race condition in the Windows 7 USB stack improperly handling string descriptors that are a multiple of 64 bytes long. This issue does not occur on Windows 10 PCs. Because the string descriptors are formatted using UCS-2, strings that are 31, 63, 95, or 127 characters long will encounter this issue.
Affected Conditions / Impacts
The CP2102N will not completely enumerate and will show up in the Ports section of the Device Manager with an exclamation point, or device driver installation will fail.
Workaround
Change the string descriptor so that it no longer has a problematic length. Devices may enumerate correctly when disconnected and reconnected.
Resolution
This issue is resolved in A02 devices.

4. Revision History

Revision 0.4

March, 2019

- Added CP2102N_E110.
- Added CP2102N_E109 to the errata history section.
- Resolved CP2102N_E104, CP2102N_E105, CP2102N_E106, CP2102N_E107 and CP2102N_E108 and moved to the errata history section.
- Updated resolutions to CP2102N_E101, CP2102N_E102, CP2102N_E103, CP2102N_E104, CP2102N_E105, CP2102N_E106, CP2102N_E107, CP2102N_E108 and CP2102N_E109.
- Updated the description of errata, affected conditions/impacts and workaround in CP2102N_E105.

Revision 0.3

November, 2018

- Added CP2102N_E108.
- Added CP2102N_E106 and CP2102N_E107.

Revision 0.2

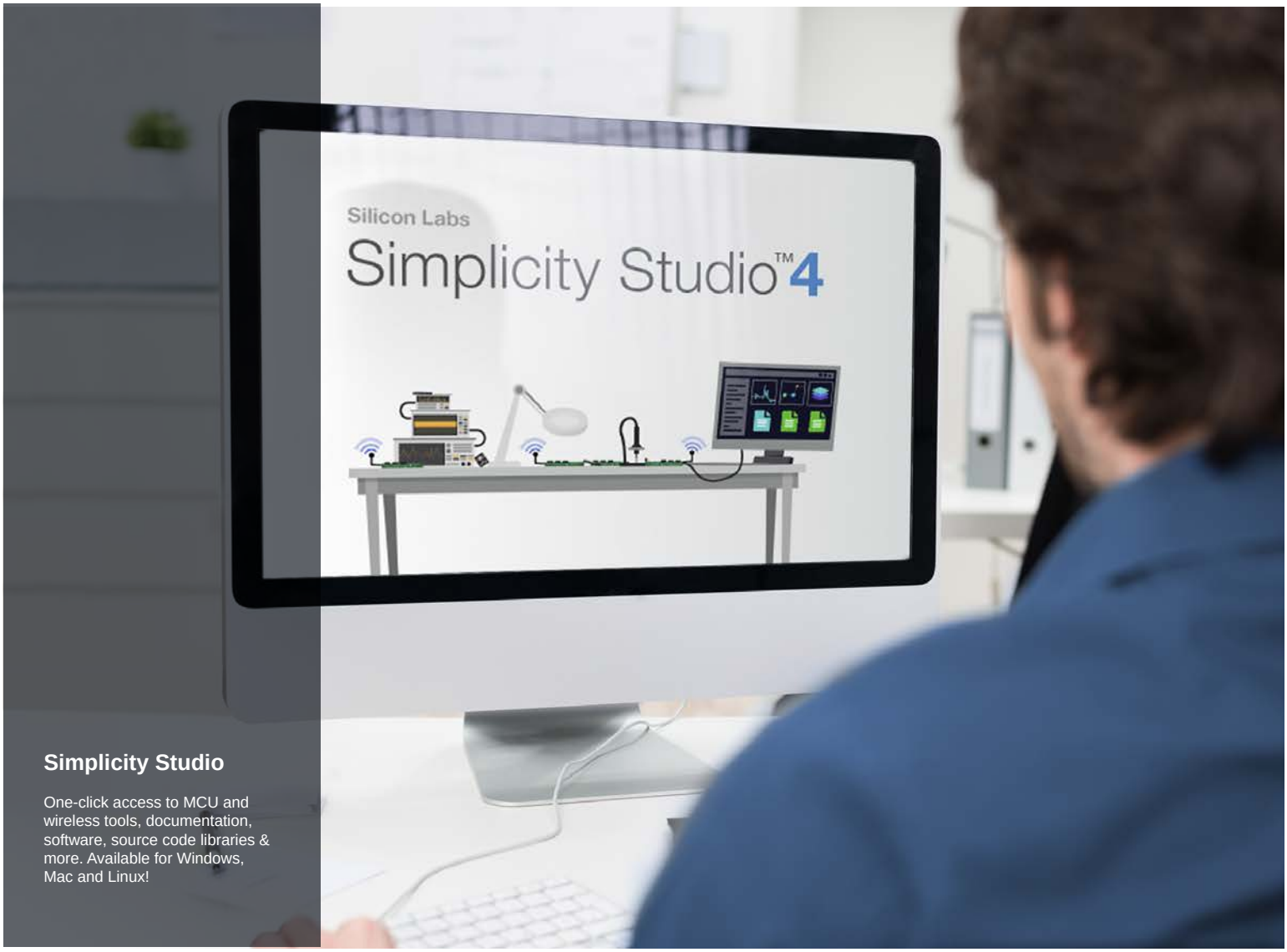
November, 2017

- Added CP2102N_E104 and CP2102N_E105.
- Moved CP2102N_E101, CP2102N_E102 and CP2102N_E103 to the errata history section.
- Merged errata history and errata into one document.
- Updated revision history format.

Revision 0.1

September, 2016

- Initial release.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com