



# 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**

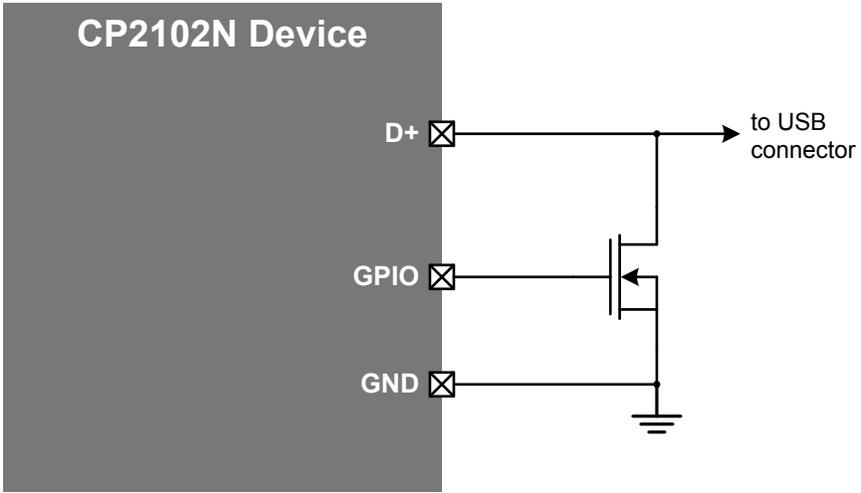
Designator	Title/Problem	Exists on Revision:		
		A01	A01 devices with a date code earlier than 1639	A02
CP2102N_E101	Failure to Power On or Reset	—	X	—
CP2102N_E102	Race Condition on Error Flags	—	X	—
CP2102N_E103	Self-Powered Device May Not Enter Suspend	—	X	—
CP2102N_E104	IO Exception in .NET Applications 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	<a href="#">USB D+/D- Power-On Reset</a>	Yes	A02	—

## 2. Detailed Errata Descriptions

### 2.1 CP2102N\_E110 – USB D+/D- Power-On Reset

<b>Description of Errata</b>
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.
<b>Affected Conditions / Impacts</b>
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.
<b>Workaround</b>
<p>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.</p> <p>An example circuit is shown below:</p> <div style="text-align: center;">  <p>The diagram shows a grey rectangular block labeled 'CP2102N Device' on the left. Three pins are indicated: 'D+', 'GPIO', and 'GND'. The 'D+' pin is connected to a horizontal line that leads to an arrow pointing right, labeled 'to USB connector'. The 'GPIO' pin is connected to the gate of an NMOS transistor. The 'GND' pin is connected to the source of the NMOS transistor. The drain of the NMOS transistor is connected to the line between the 'D+' pin and the 'to USB connector' arrow. The gate of the transistor is also connected to a vertical line that goes up to the 'D+' pin, forming a feedback loop that shorts D+ to ground when the transistor is turned on.</p> </div> <p>This circuit uses an NMOS transistor 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 (<a href="http://www.silabs.com/simplicity">www.silabs.com/simplicity</a>).</p>
<b>Resolution</b>
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

<b>Description of Errata</b>
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.
<b>Affected Conditions / Impacts</b>
Systems using the CP2102N may see devices fail to respond until a power-on reset.
<b>Workaround</b>
If a device fails to respond properly, remove and replace power until the device properly responds.
<b>Resolution</b>
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

<b>Description of Errata</b>
Devices can fail to notify the host of an error flag if an error occurs while the host is reading the UART status.
<b>Affected Conditions / Impacts</b>
The following error conditions may be missed if they occur while the host is reading the UART status: <ul style="list-style-type: none"> <li>• Set Break</li> <li>• Hard overrun</li> <li>• Queue overrun</li> <li>• Parity error</li> </ul>
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
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

<b>Description of Errata</b>
A device in the self-powered configuration may not enter Suspend mode properly if the USB host is disconnected (i.e. cable unplugged).
<b>Affected Conditions / Impacts</b>
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.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
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

<b>Description of Errata</b>
The CP2102N uses the incorrect byte of the SERIAL_HANDFLOW structure ( <a href="https://msdn.microsoft.com/en-us/library/windows/hardware/jj680685(v=vs.85).aspx">https://msdn.microsoft.com/en-us/library/windows/hardware/jj680685(v=vs.85).aspx</a> ) 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.
<b>Affected Conditions / Impacts</b>
Applications written in .NET will see IO exceptions when attempting to manually set or clear RTS on the CP2102N.
<b>Workaround</b>
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:  <a href="https://www.silabs.com/community/interface/knowledge-base.entry.html/2017/11/10/cp2102n_e104_io_ex-YqAX.html">https://www.silabs.com/community/interface/knowledge-base.entry.html/2017/11/10/cp2102n_e104_io_ex-YqAX.html</a>  If the serial port is configured in C++, directly set the DCB XON/XOFF limits to 0x00 as a workaround for this issue.
<b>Resolution</b>
This issue is resolved in A02 devices.

### 3.7 CP2102N\_E105 – RS485 DE Signal Not Held Long Enough

<b>Description of Errata</b>
The CP2102N RS485 signal is not held long enough.
<b>Affected Conditions / Impacts</b>
Devices configured in RS485 mode will not communicate properly.
<b>Workaround</b>
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 ( <a href="http://www.silabs.com/simplicity">www.silabs.com/simplicity</a> ).
<b>Resolution</b>
This issue is resolved in A02 devices.

### 3.8 CP2102N\_E106 – DTR/RTS behavior when port is closed

<b>Description of Errata</b>
CP2102N does not reset DTR/RTS when port is closed.
<b>Affected Conditions / Impacts</b>
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.
<b>Workaround</b>
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.
<b>Resolution</b>
This issue is resolved in A02 devices.

**3.9 CP2102N\_E107 – SUSPEND and SUSPENDb state during enumeration**

<b>Description of Errata</b>
SUSPEND and SUSPENDb will not assert until after the device completes a USB configuration.
<b>Affected Conditions / Impacts</b>
CP2102N's SUSPEND and SUSPENDb pins deassert during power on for 3ms, instead of asserting until the USB enumeration process completes.
<b>Workaround</b>
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.
<b>Resolution</b>
This issue is resolved in A02 devices.

**3.10 CP2102N\_E108 – Failure to Enumerate**

<b>Description of Errata</b>
The CP2102N may fail to enumerate.
<b>Affected Conditions / Impacts</b>
When connecting the CP2102N to a USB host, the CP2102N may fail to enumerate.
<b>Workaround</b>
If a device fails to enumerate, disconnect and reconnect the device to the USB host until the device successfully enumerates.
<b>Resolution</b>
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**

<b>Description of Errata</b>
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.
<b>Affected Conditions / Impacts</b>
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.
<b>Workaround</b>
Change the string descriptor so that it no longer has a problematic length. Devices may enumerate correctly when disconnected and reconnected.
<b>Resolution</b>
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.

Silicon Labs

# Simplicity Studio™4



## 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](http://www.silabs.com/IoT)



**SW/HW**  
[www.silabs.com/simplicity](http://www.silabs.com/simplicity)



**Quality**  
[www.silabs.com/quality](http://www.silabs.com/quality)



**Support and Community**  
[community.silabs.com](http://community.silabs.com)

### Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

### Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR®, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, ISOModem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



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

<http://www.silabs.com>