CP2112 Errata

This document contains information on the CP2112 errata. The latest available revision of this device is revision F03.

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 CP2112 and all unresolved errata in revision F03 of the CP2112.

Table 1.1. Errata History Overview

Designator	Title/Problem	Exists on Revision:			
		Data Sheet revision 1.2 and below	F01	F02	F03
CP2112_E101	GetReadResponse May Return Incorrect Data	—	X	X	—
CP2112_E102	ROM Programming Voltage	X	—	—	—
CP2112_E103	No Enumeration if I2C Stuck Low	—	X	X	—
CP2112_E104	Address Read Request with NAK	—	X	X	—
CP2112_E105	Occasional Double START on Read Request	—	X	X	—
CP2112_E106	Addressed Read Format	—	X	—	—
CP2112_E107	Multimaster Applications	—	X	—	—
CP2112_E108	Failure to Enumerate	—	X	X	—
CP2112_E109	HID reports for Manufacturing/Product strings missing last byte	—	X	X	X
CP2112_E110	Get_Report (Feature Report 0x01) causes device reset	—	X	X	X
CP2112_E111	Spurious Input Reports sent after device enumerates	—	X	X	X

Table 1.2. Active Errata Status Summary

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
1	CP2112_E109	HID reports for Manufacturing/Product strings missing last byte	Yes	F03	—
2	CP2112_E110	Get_Report (Feature Report 0x01) causes device reset	Yes	F03	—
3	CP2112_E111	Spurious Input Reports sent after device enumerates	Yes	F03	—

2. Detailed Errata Descriptions

2.1 CP2112_E109 – HID reports for Manufacturing/Product strings missing last byte

Description of Errata
When read using a USB Get_Report (Feature Report) request, the CP2112 Manufacturing and Product strings do not contain the last 0x00 byte (i.e. the MSB of the UTF-16 character is not sent). The last 0x00 byte is present when the Manufacturing and Product strings are read using the USB Get_Descriptor request.
Affected Conditions / Impacts
The effect of the missing 0x00 byte depends on the behavior of the host driver or application software.
Workaround
The Manufacturing and Product strings can be read using the USB Get_Descriptor request.
Resolution
There is currently no resolution for this issue.

2.2 CP2112_E110 – Get_Report (Feature Report 0x01) causes device reset

Description of Errata
As defined in AN495 (CP2112 Interface Specification), Feature Report 0x01 is an output-only report that is used to reset the device. However, issuing a Get_Report for this feature report will also cause a device reset.
Affected Conditions / Impacts
Prior to Linux kernel version 4.12, the Linux usbhid driver performed a Get_Report request on all Feature Reports declared in a device's HID descriptor. The Get_Report request for Feature Report 0x01 causes the CP2112 to reset, which results in a failed enumeration.
Workaround
Update to Linux kernel v4.12 or later. Alternatively, recompiling the Linux kernel with the 'HID_QUIRK_NO_INIT_REPORTS' quirk added to the CP2112 driver ('hid-cp2112 driver.c') will prevent Linux from reading all Feature Reports from the CP2112 at enumeration.
Resolution
There is currently no resolution for this issue.

2.3 CP2112_E111 – Spurious Input Reports sent after device enumerates

Description of Errata
During enumeration the CP2112 may spuriously send these Input Reports to the host. These reports are defined in Application Note AN495 (CP2112 Interface Specification): • Data Read Response (Report ID 0x13) • Transfer Status Response (Report ID 0x16)
Affected Conditions / Impacts
There have been no observed or reported problems due to these spurious reports.
Workaround
If necessary, the host driver or application software should ignore these spurious reports.
Resolution
There is currently no resolution for this issue.

3. Errata History

This section contains the errata history for CP2112 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 CP2112.

Table 3.1. Errata History Status Summary

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
1	CP2112_E101	GetReadResponse May Return Incorrect Data	Yes	F02	F03
2	CP2112_E102	ROM Programming Voltage	Yes	Data Sheet revision 1.2 and below	Data Sheet revision 1.3
3	CP2112_E103	No Enumeration if I2C Stuck Low	No	F02	F03
4	CP2112_E104	Address Read Request with NAK	Yes	F02	F03
5	CP2112_E105	Occasional Double START on Read Request	No	F02	F03
6	CP2112_E106	Addressed Read Format	No	F01	F02
7	CP2112_E107	Multimaster Applications	No	F01	F02
8	CP2112_E108	Failure to Enumerate	No	F02	F03

3.2 Detailed Errata Descriptions

3.2.1 CP2112_E101 – GetReadResponse May Return Incorrect Data

Description of Errata
If the `autoReadRespond` feature is used with the `HidSmbus_AddressReadRequest()` API, `HidSmbus_GetReadResponse()` may return incorrect data where the first bytes of the response are overwritten by the target address.
Affected Conditions / Impacts
Systems using the `autoReadRespond` feature along with `HidSmbus_AddressReadRequest()` may see incorrect data when using `HidSmbus_GetReadResponse()`.
Workaround

To work around this problem, disable the `autoReadRespond` feature and use the `HidSmbus_ForceReadResponse()` function. For example, with `autoReadRespond` enabled using the `HidSmbus_SetSmbusConfig()` function:
`HidSmbus_AddressReadRequest()`
`HidSmbus_GetReadResponse() // device automatically sends any data; many`
`// reports won't be full`
`…`
`HidSmbus_GetReadResponse()`
If `autoReadRespond` is disabled, the recommended sequence is as follows:
`HidSmbus_AddressReadRequest()`
`// poll transfer status until transfer is done using`
`// HidSmbus_TransferStatusRequest() and HidSmbus_GetTransferStatusResponse()`
`HidSmbus_ForceReadResponse()`
`HidSmbus_GetReadResponse() // reports will generally be full`
`…`
`HidSmbus_GetReadResponse()`

Resolution
This issue is resolved in revision F03 devices.

3.2.2 CP2112_E102 – ROM Programming Voltage

Description of Errata
The data sheet incorrectly indicates that VDD must remain at 3.3 V or higher to successfully write to the configuration ROM. Instead, the voltage on the VIO pin must remain at 3.3 V or higher when writing to the configuration ROM.
Affected Conditions / Impacts
For systems that connect VDD and VIO together, there is no impact. For systems that have a separate voltage source for VIO and are configuring the ROM in-system, VIO must remain at 3.3 V while programming is in progress.
Workaround
For systems that connect VDD and VIO together, keep both power supplies above 3.3 V when programming. For systems that have a separate voltage source for VIO and are configuring the ROM in-system, VIO must remain at 3.3 V while programming is in progress.
Resolution
This issue is resolved in data sheet revision 1.3 or later.

3.2.3 CP2112_E103 – No Enumeration if I2C Stuck Low

Description of Errata
The CP2112-F02 behaves as follows if its I2C data pin (SDA) is in the low logic state when the CP2112-F02 is reset or power-cycled. When this occurs: • The CP2112-F02 will not enumerate as long as SDA remains low. • The CP2112-F02 will enumerate when SDA becomes high. However, if SDA was low at reset or power-up, the device's I2C, GPIO, and SUSPEND pins will remain in the high-impedance state and cannot be driven low by the CP2112-F02. In this state, the device correctly reads the logic state of GPIO inputs.
Affected Conditions / Impacts
Systems that use the CP2112 where the data pin (SDA) could be stuck low while the CP2112 is resetting or starting up will not see the device enumerate properly. When this enumeration failure occurs and after SDA becomes a logic high, the CP2112 I2C, GPIO, and SUSPEND pins will not drive low as normal.
Workaround
Connecting an unused GPIO input to the SUSPEND pin allows the host to detect the state in which the CP2112-F02 outputs cannot be driven low (i.e. SDA was low at reset/powerup but has since gone high). After enumeration, the SUSPEND pin should be low while in Active Mode (i.e. not suspended). If the GPIO input indicates the SUSPEND pin is high during Active Mode, the host should reset the CP2112-F02 by calling the *HidSmbus_Reset* library function or by issuing HID Feature Request 0x01. If SDA is high at reset, the CP2112-F02 will then behave normally. See *AN495: CP2112 Interface Specification* and *AN496: CP2112 HID USB-to-SMBus API Specification* for details on the CP2112 library function calls and interface specification. Application Notes can be found in Simplicity Studio in the Documentation area or on the website at www.silabs.com/interface-appnotes.
Resolution
This issue is resolved in revision F03 devices.

3.2.4 CP2112_E104 – Address Read Request with NAK

Description of Errata
If software issues an Addressed Read Request with an invalid address, the command is correctly NAKed after the address is sent. If this NACKed Address Read Request is followed with a regular Read Request, the CP2112 device correctly sends the ADDR, which is ACKed, and then starts to write the target address again, as if the Read Request is another Addressed Read Request.
Affected Conditions / Impacts
Systems switching between Addressed Read Requests and Read Requests may observe the Read Request treated like an Addressed Read Request if the previous Address Read Request is NACKed.
Workaround
For systems switching between Addressed Read Requests and Read Requests, ensure the Address Read Request uses a valid address that is ACKed.
Resolution
This issue is resolved in revision F03 devices.

3.2.5 CP2112_E105 – Occasional Double START on Read Request

Description of Errata
When using standard Read Requests, the device will occasionally send two STARTs when the clock rate is set to less than or equal to 100 kHz.
Affected Conditions / Impacts
Systems using standard Read Requests may see an occasional request with two STARTs.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision F03 devices.

3.2.6 CP2112_E106 – Addressed Read Format

Description of Errata
In F01 devices, addressed read requests are performed by issuing a start on the bus, followed by a slave address (write), logical address to read, stop, start, and slave address (read). These devices do not use a repeated start, which may be incompatible with some SMBus slaves.
Affected Conditions / Impacts
Systems using addressed reads may not be able to communicate with all SMBus slaves.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision F02 devices.

3.2.7 CP2112_E107 – Multimaster Applications

Description of Errata
F01 devices can hold the SDA line low for approximately 3 ms if the Set SMBus Configuration command (Report ID 0x06) is received by one CP2112 master during the middle of a separate master device's transaction. A fix is implemented on F02 devices to eliminate this behavior.
Affected Conditions / Impacts
Systems with multiple SMBus masters including the CP2112 may experience communication issues.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision F02 devices.

3.2.8 CP2112_E108 – Failure to Enumerate

Description of Errata
Devices can fail to enumerate properly on initial power on, after a device reset, or when connected to a USB port. In the case of a failure, the device will lock up until the next reset or power on reset. The failure rate is intermittent and will vary from device to device.
Affected Conditions / Impacts
The device can fail to enumerate on initial power on, after a device reset, or when connected to a USB port.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision F03 devices.

4. Revision History

Revision 0.4

March, 2019

- Updated the errata to latest revision, CP2112-F03.
- Added CP2112_E109, CP2112_E110, and CP2112_E111.
- Updated 4. Revision History to adhere to new schema.

Revision 0.3

February, 2017

- Updated the resolutions to CP2112_E101, CP2112_E103, CP2112_E104, and CP2112_E105.
- Added CP2112_E106 and CP2112_E107 to document revision F01 device behaviors in the errata history.

Revision 0.2

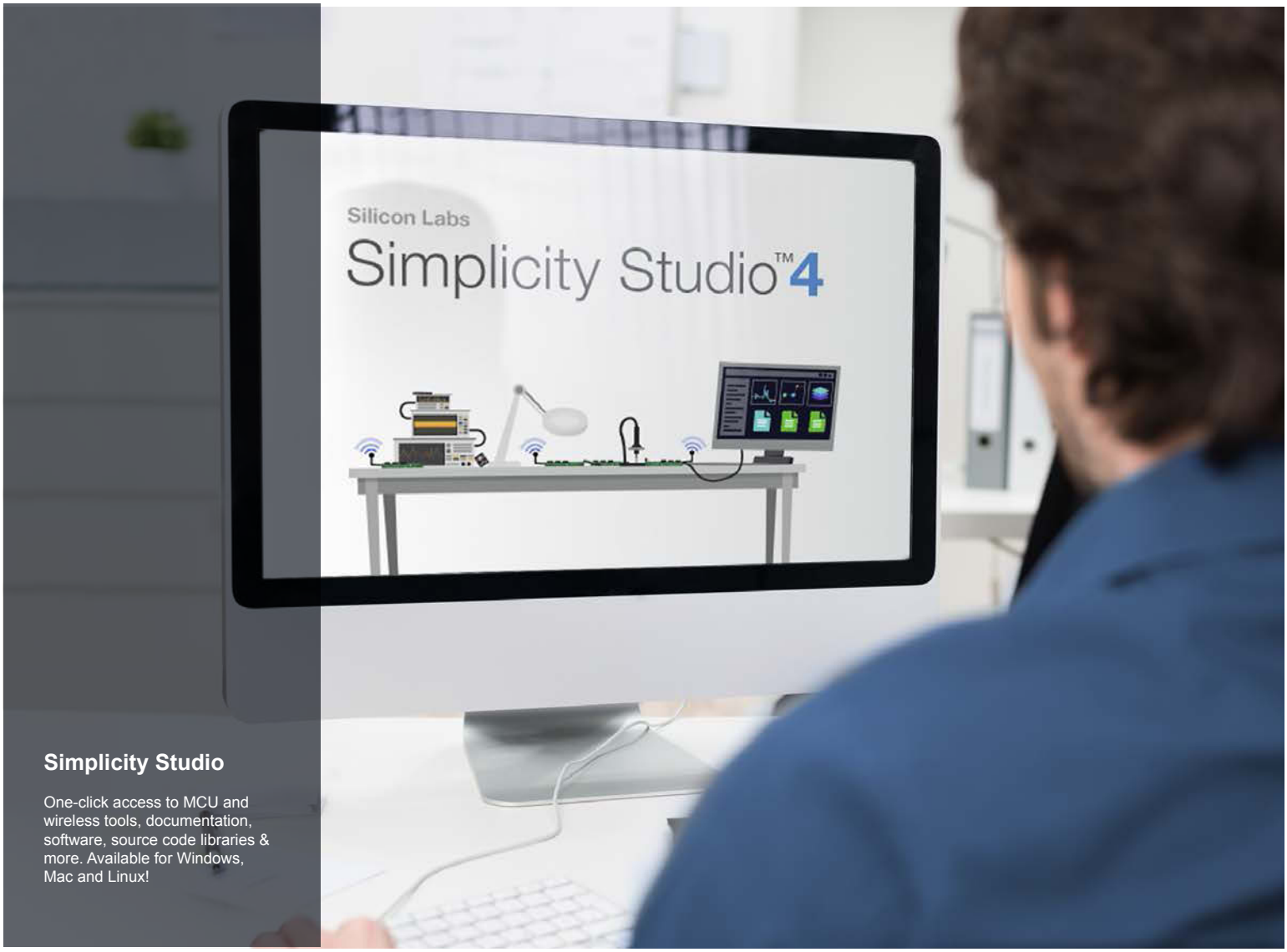
November, 2016

- Added CP2112_E101, CP2112_E103, CP2112_E104, and CP2112_E105.
- Updated the resolution for CP2112_E102.

Revision 0.1

April, 2015

- 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