

CP2112 INTERFACE SPECIFICATION

1. Introduction

The Silicon Laboratories CP2112 is a USB device that adheres to the USB-defined Human Interface Device class specification. HID devices communicate with a USB host through the use of reports. This document is a specification for the reports supported by the CP2112 and describes the configurable parameters.

Silicon Laboratories provides dynamic libraries that adhere to this specification for the following operating systems:

- Windows®
- Mac®

This document is intended for the following:

- Users who are using an operating system that is not supported by the dynamic libraries and who need to implement their own interface.
- Users who want to integrate the device interface into their application.

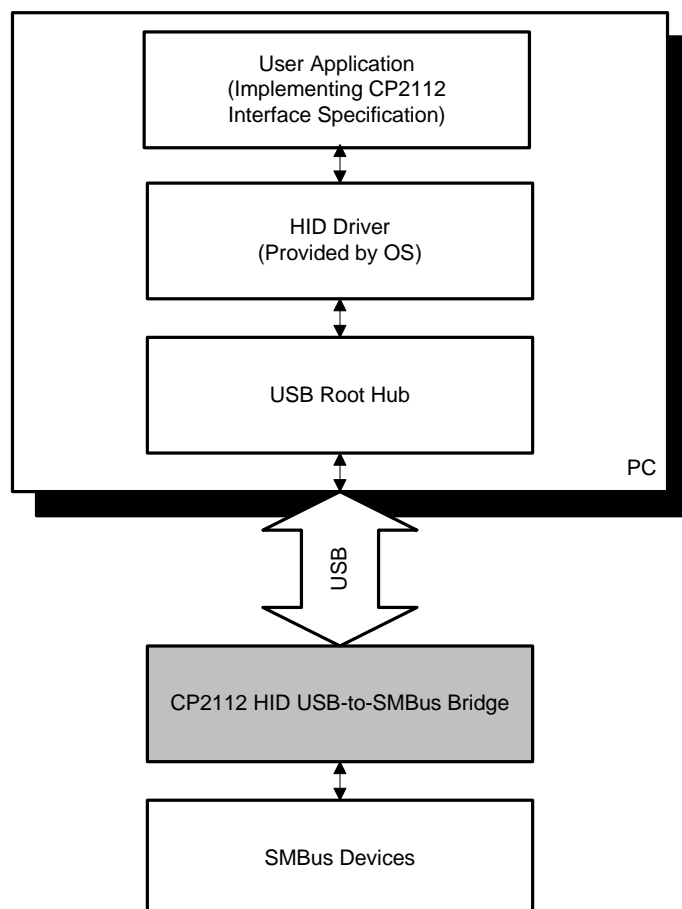


Figure 1. System Architecture Diagram

1.1. Additional Documentation

- CP2112 data sheet:
<http://www.silabs.com/products/interface/hidusbtoSMBus>
- Dynamic libraries that adhere to this CP2112 Interface specification are available for the following operating systems:
 - Windows 7/Vista/XP
 - Mac OS X
- AN496: CP2112 HID USB-to-SMBus API Specification documents the Windows and Mac DLL. The document and libraries are available at:
<http://www.silabs.com/products/interface/hidusbtoSMBus>
- HID Device Class Definition:
<http://www.usb.org/developers/hidpage/>

1.2. Default Values for Parameters Stored in PROM and Pin Settings

Table 1 lists the default values for the one-time configurable parameters stored in the PROM of the CP2112. Table 2 lists the default values for the GPIO, Suspend, and SMBus pins. The GPIO pins are configured in RAM and should be reconfigured each time the device is reset. The suspend and SMBus pins are non-configurable.

Table 1. Default PROM Values

| Parameter | Default Value |
|----------------------|--------------------------------|
| VID | 0x10C4 |
| PID | 0xEA90 |
| Power | 0x32 (100 mA) |
| Power Mode | Bus Powered |
| Release Number | 0x0100 (Release Version 01.00) |
| Manufacturing String | Silicon Laboratories |
| Product String | CP2112 HID USB-to-SMBus Bridge |
| Serial String | 0001 |
| Lock Byte | 0xFF* |

***Note:** The lock bytes indicate which parameters have already been programmed. See "7.1. Get/Set Lock Byte" on page 17 for more details.

Table 2. Default Pin Settings

| Parameter | Default Config Value |
|-----------------------------|----------------------|
| GPIO0_TXT | Open-Drain |
| GPIO1_RXT | Open-Drain |
| GPIO2 | Open-Drain |
| GPIO3 | Open-Drain |
| GPIO4 | Open-Drain |
| GPIO5 | Open-Drain |
| GPIO6 | Open-Drain |
| GPIO7_CLK | Open-Drain |
| SDA | Open-Drain |
| SCL | Open-Drain |
| Suspend | Push-Pull Output |
| $\overline{\text{Suspend}}$ | Push-Pull Output |

1.3. Default SMBus Configuration

Table 3 lists the default settings for the SMBus. These settings are used at device power-up or reset.

Table 3. SMBus Settings

| Parameter | Default Value |
|------------------|----------------------|
| Clock Speed | 0x0186A0 (100 kHz) |
| Device Address | 0x02 |
| Auto Send Read | 0x00 (Disabled) |
| Write Timeout | 0x00 (Disabled) |
| Read Timeout | 0x00 (Disabled) |
| SCL Low Timeout | 0x00 (Disabled) |
| Retry Time | 0x00 (No Limit) |

2. Report Overview

Communication with the CP2112 is performed using HID Reports as defined in the HID Device Class Definition. The class definition is available for download from <http://www.usb.org/developers/hidpage/>.

2.1. Reports Response

The CP2112 responds to reports in different ways, depending on whether the report configures a parameter on the device using a Set Report or requests data from the device using a Get Report. A list of all supported reports is available in "3. Report ID List" on page 6.

2.1.1. Set Reports

In response to a Set Report, the CP2112 will not return any report or acknowledgement of a report. To verify that a report has completed successfully, use the corresponding Get Report to obtain the data. The delay imposed by the HID protocol between reports guarantees that there will be no race condition between the execution of a Set Report and Get Report verification. A Set Report will always complete before the device receives the Get Report.

2.1.2. Get Reports

If a report requests data from the device and the report is valid, the device will return a report with the requested data. If the report is invalid, the device will stall.

2.2. Data Format

In all of the reports, the first byte of the data portion of the payload is the Report ID. In the report definitions in this document, the Report ID is stored in index 0 of the payload and is not explicitly listed in the table. All data content in the report starts at index 1. The reports have a maximum length of 64 bytes, indexed from 0 to 63. For any multi-byte values sent in the reports, the values are sent most-significant byte first.

3. Report ID List

Table 4. Report IDs

| Report ID | Report Name | Page |
|---|------------------------------|------|
| Device Configuration (Feature Request) | | |
| 0x01 | Reset Device | 7 |
| 0x02 | Get/Set GPIO Configuration | 8 |
| 0x03 | Get GPIO | 9 |
| 0x04 | Set GPIO | 9 |
| 0x05 | Get Version Information | 9 |
| 0x06 | Get/Set SMBus Configuration | 10 |
| Data Transfer (Interrupt Transfer) | | |
| 0x10 | Data Read Request | 11 |
| 0x11 | Data Write Read Request | 11 |
| 0x12 | Data Read Force Send | 12 |
| 0x13 | Data Read Response | 12 |
| 0x14 | Data Write | 13 |
| 0x15 | Transfer Status Request | 13 |
| 0x16 | Transfer Status Response | 14 |
| 0x17 | Cancel Transfer | 15 |
| USB Customization (Feature Requests) | | |
| 0x20 | Get/Set Lock Byte | 17 |
| 0x21 | Get/Set USB Configuration | 18 |
| 0x22 | Get/Set Manufacturing String | 18 |
| 0x23 | Get/Set Product String | 19 |
| 0x24 | Get/Set Serial String | 19 |

4. Device Configuration Reports

4.1. Reset Device

Report ID: 0x01

Direction: Feature Request Out

| Name | Offset | Size | Value | Description |
|------------|--------|------|-------|---------------------------|
| Reset Type | 1 | 1 | 0x01 | Reset with re-enumeration |

Reset Device is used to restart the device from the USB host. The device re-enumerates on the USB bus, and all SMBus configuration settings are reset to their default values. If any value other than 0x01 is sent with the *Reset Type* field of the command, the command will be ignored.

For certain operating systems, such as Windows, initiating a device reset and re-enumerating will make the device's handle stale. The user application is responsible for handling this "surprise disconnect" event. See "AN496: CP2112 HID USB-to-SMBus API Specification" for more information regarding surprise disconnects.

4.2. Get/Set GPIO Configuration

Report ID: 0x02

Direction: Feature Request In/Out

| Name | Offset | Size | Value | Description |
|---------------|--------|------|-------|---|
| Direction | 1 | 1 | 0x00 | 0 = input 1 = output |
| Push-Pull | 2 | 1 | 0x00 | 0 = open-drain 1 = push-pull |
| Special | 3 | 1 | 0x00 | Enables special functions of GPIO pins. See Table 5. |
| Clock Divider | 4 | 1 | 0x00 | 0 = 48 MHz Clock Output All other values will output a clock signal determined by Equation 1 below. |

Get/Set GPIO Configuration is used to configure the eight GPIO pins as input/output and open-drain/push-pull through the *Direction* and *Push-Pull* fields. GPIO0 corresponds to bit zero (the least significant bit), and GPIO7 corresponds to bit seven (the most significant bit). For pins that are configured as an input, push-pull mode is ignored.

Special is used to enable special functionality on GPIO0_TXT, GPIO1_RXT, and GPIO7_CLK. By setting a bit in the *Special* byte of this report, the functionality is enabled. Table 5 has more information regarding the special functionality.

Table 5. Special Function Summary

| Bit in Special Byte of Get/Set GPIO Configuration | GPIO Pin | Value | Description |
|---|-----------|--------|---|
| Bit 0 | GPIO7_CLK | 0 1 | GPIO Pin Clock Output - Push-Pull Output |
| Bit 1 | GPIO0_TXT | 0 1 | GPIO Pin TX Toggle - Push-Pull Output |
| Bit 2 | GPIO1_RXT | 0 1 | GPIO Pin RX Toggle - Push-Pull Output |
| Bit 3 - Bit 7 | — | — | Not Used |

Clock Divider specifies the clock output frequency on GPIO7_CLK when the clock output is enabled. If *Clock Divider* is 0, a 48 MHz clock will be output on GPIO7_CLK. Otherwise, Equation 1 can be used to determine the clock output.

$$\text{GPIO7_CLK Clock Frequency} = \frac{48 \text{ MHz}}{2 \times \text{ClockDivider}}$$

Equation 1. GPIO Clock Frequency Output

4.3. Get GPIO Values

Report ID: 0x03

Direction: Feature Request In

| Name | Offset | Size | Value | Description |
|-------------|--------|------|-------|----------------------|
| Latch Value | 1 | 2 | * | Current latch values |

If a pin is configured as a GPIO input pin, the corresponding *Latch Value* bit represents the input value. If a pin is configured as a GPIO output pin, the corresponding *Latch Value* bit represents the logic level driven on the pin. GPIO0_TXT corresponds to bit zero (the least significant bit), and GPIO7_CLK corresponds to bit seven (the most significant bit).

4.4. Set GPIO Values

Report ID: 0x04

Direction: Feature Request Out

| Name | Offset | Size | Value | Description |
|-------------|--------|------|-------|-------------------------------|
| Latch Value | 1 | 1 | * | Latch value |
| Latch Mask | 2 | 1 | * | Pin to set to new latch value |

Set GPIO Values sets the values for GPIO pins.

The desired value for the pin is configured in *Latch Value*. To drive a “1” on an output pin, the corresponding bit should be set to “1”. To drive a “0” on an output pin, the corresponding bit should be set to “0”.

The Report sets new values only for output pins that have a “1” in the corresponding bit position in *Latch Mask*. If the corresponding bit in *Latch Mask* is set to “0”, a new pin value will not be set, even if the pin is configured as an output pin.

The Report does not affect any pins that are not configured as outputs.

4.5. Get Version Information

Report ID: 0x05

Direction: Feature Request In

| Name | Offset | Size | Value | Description |
|----------------|--------|------|--------|--------------------|
| Part Number | 1 | 1 | 0x0C | Device part number |
| Device Version | 2 | 1 | Varies | |

Part Number indicates the device part number. The CP2112 returns 0x0C.

Device Version is the version of the device. This value is not programmable over the HID interface.

4.6. Get/Set SMBus Configuration

Report ID: 0x06

Direction: Feature Request In/Out

The values in **bold** are the default values.

| Name | Offset | Size | Value | Description |
|-----------------|--------|------|---------------------|---|
| Clock Speed | 1 | 4 | 0x186A0 * | 100 KHz SMBus Clock Speed in Hertz |
| Device Address | 5 | 1 | 0x02 | Bits 7–1 make up device address (least significant bit is masked) |
| Auto Send Read | 6 | 1 | 0x00 0x01 | Disabled Enabled |
| Write Timeout | 7 | 2 | 0x0000 * | No Timeout 0–1000 ms timeout value |
| Read Timeout | 9 | 2 | 0x0000 * | No Timeout 0–1000 ms timeout value |
| SCL Low Timeout | 11 | 1 | 0x00 0x01 | Disabled Enabled |
| Retry Time | 12 | 2 | 0x0000 * | No Limit 0–1000 retries |

Values from the Set Report are not stored in PROM. These parameters must be initialized after every power-on or device reset.

Clock Speed is the frequency of the SMBus clock in hertz (Hz). It is stored as a big-endian 4-byte unsigned number. For example, to run the SMBus at 400 kHz, the value, 0x61A80, would need to be written to four clock speed bytes in this report. If a value of zero is written, this parameter is ignored.

Device Address is the 7-bit slave address of the CP2112. The CP2112 will ACK this address, but the CP2112 will not respond to any read or write requests. If the least significant bit is set in this byte (read/write bit), this parameter is ignored.

Write Timeout and *Read Timeout* are the time limits in milliseconds before the CP2112 automatically cancels a transfer that has been initiated. If a transfer is canceled by a write or read timeout, the status byte of the *Transfer Status Response* command is set appropriately. If this value is set to 0x0000, the device continues to attempt the transfer until the transfer completes or until the *Cancel Transfer* command is sent. Any value greater than 1000 is ignored.

SCL Low Timeout is a timeout that resets the SMBus if the SCL line is held low for more than 25 ms. If enabled and an SCL Low Timeout occurs, the status byte of the *Transfer Status Response* command will be set appropriately. Any value other than 0x00 and 0x01 is ignored.

Retry Time is the number of attempts that the CP2112 attempts to complete a transfer before terminating the transfer. Any value greater than 1000 is ignored.

5. Data Transfer Reports

5.1. Data Read Request

Report ID: 0x10

Direction: Interrupt Out

| Name | Offset | Size | Value | Description |
|---------------|--------|------|-------|--|
| Slave Address | 1 | 1 | * | Must be between 0xF7 - 0x02. Least significant bit is read/write bit and must be zero. |
| Length | 2 | 2 | * | Number of bytes (1-512 bytes) to read back |

Slave Address is the 7-bit address of the slave device from which data is being read. The address must be between 0xF7 and 0x02 (the least significant bit is the read/write bit and must be zero). All other values are ignored.

Length is the number of bytes being requested from the slave device. The CP2112 can request between 1 and 512 bytes from a slave device. Any value outside that range is ignored.

5.2. Data Write Read Request

Report ID: 0x11

Direction: Interrupt Out

| Name | Offset | Size | Value | Description |
|-----------------------|--------|------|-------|---|
| Slave Address | 1 | 1 | * | Must be between 0xF7 and 0x02. Least significant bit is the read/write bit and must be zero. |
| Length | 2 | 2 | * | Number of bytes (1–512 bytes) to read back |
| Target Address Length | 4 | 1 | * | Number of bytes in target address (from 0x01 to 0x10) |
| Target Address | 5 | 16 | * | Address of device to be read. The number of bytes in this field must match the number of bytes specified in the <i>target length address</i> above. |

Slave Address is the 7-bit address of the slave device from which data is being read. The address must be between 0xF7 and 0x02 (the least significant bit is the read/write bit and must be zero). All other values are ignored.

Length is the number of bytes that are being requested from the slave device. The CP2112 can request between 1 and 512 bytes from a slave device. Any value outside that range is ignored.

Target Address Length is the number of bytes in the target address of the slave device from which data is being read. The target address length must be between 1 and 16. Any value outside that range is ignored.

Target Address is the address of the memory location being read on the slave device. The number of bytes in the target address must match the number of bytes in target address length. Target Address Length specifies the number of valid address bytes in Target Address starting with the byte at Offset position 5.

This type of report is used when a repeated start condition is used, which is common when accessing an EEPROM or something similar.

5.3. Data Read Force Send

Report ID: 0x12

Direction: Interrupt Out

| Name | Offset | Size | Value | Description |
|--------|--------|------|-------|----------------------------|
| Length | 1 | 2 | * | Number of valid data bytes |

This forces the device to send a *Data Read Response* report when the next Interrupt IN token arrives. This is essentially polled mode reading. The PC should poll using *Transfer Status Request* first to determine whether data is ready. The number of bytes requested can be 1 to 512. If the number of bytes requested is greater than the number of valid bytes in the CP2112's received bytes buffer, only the valid bytes will be returned.

This command should only be used when *Auto Send Read* is set to 0x00. This command is ignored when *Auto Send Read* is set to 0x01. If a transfer is not in progress or if no data is in the buffer, this command performs no action. This command can be used while a read is in progress to retrieve the data received so far.

5.4. Data Read Response

Report ID: 0x13

Direction: Interrupt In

| Name | Offset | Size | Value | Description |
|--------|--------|------|------------------------------|---|
| Status | 1 | 1 | 0x00 0x01 0x02 0x03 | Idle Busy Complete (once read, reverts to 0x00) Complete with error (once read, reverts to 0x00) |
| Length | 2 | 1 | * | Number of valid data bytes |
| Data | 3 | 61 | * | Data being returned from the SMBus slave device |

Data for *Data Read Request*, *Data Write Request*, and *Data Read Force Send* will be returned to the host with this report ID.

Status shows the current status of the SMBus transmission.

Length is the number of valid data bytes being returned to the host.

Data is the actual data being returned to the host. Only the number of bytes specified in the *Length* field will be valid.

5.5. Data Write

Report ID: 0x14

Direction: Interrupt Out

| Name | Offset | Size | Value | Description |
|---------------|--------|------|-------|--|
| Slave Address | 1 | 1 | * | Must be between 0xF7 - 0x02. Least significant bit is read/write bit and must be zero. |
| Length | 2 | 1 | * | Number of valid data bytes |
| Data | 3 | 61 | * | Data being returned from the SMBus slave device |

Slave Address is the 7-bit address of the slave device to which data is being sent. The address must be between 0xF7 and 0x02 (the least significant bit is the read/write bit and must be 0). All other values are ignored.

Length is the number of bytes that are being sent to the slave device. The host can transmit 1 to 61 bytes to the CP2112. Any value outside that range is ignored.

Data is the actual data being sent over the SMBus to the slave device. The number of data bytes being sent in this field should match the number of bytes specified in the *Length* field.

5.6. Transfer Status Request

Report ID: 0x15

Direction: Interrupt Out

| Name | Offset | Size | Value | Description |
|---------|--------|------|-------|-------------------------------|
| Request | 1 | 1 | 0x01 | Request SMBus transfer status |

Request is used to request the current SMBus transfer status. If any value other than 0x01 is sent with the transfer status request, the command will be ignored.

AN495

5.7. Transfer Status Response

Report ID: 0x16

Direction: Interrupt In

| Name | Offset | Size | Value | Description |
|----------|--------|------|------------------------------|---|
| Status 0 | 1 | 1 | 0x00 0x01 0x02 0x03 | Idle Busy Complete (once read, reverts to 0x00) Complete with error (once read, reverts to 0x00) |
| Status 1 | 2 | 1 | * | Specific conditions based on <i>Status 0</i> (see below) |
| Status 2 | 3 | 2 | * | Number of retries before completing, being canceled, or timing out |
| Status 3 | 5 | 2 | * | Number of received bytes |

Transfer Status Response returns information on the SMBus interface.

Status 0 returns the general status of the CP2112 SMBus interface.

If *Status 0* is 0x00, the rest of the status bytes are not valid.

Status 1 returns specific information based off of *Status 0* and is dependent on the value of *Status 0*.

Status 2 returns the number of retries before completing, being cancelled, or timing out.

Status 3 returns the bytes successfully read.

| Status 0 | Status 1 | Description |
|---------------|--|--|
| 0x01 | 0x00 0x01 0x02 0x03 | Address ACKed Address NACKed Data read in progress Data write in progress |
| 0x02 and 0x03 | 0x00 0x01 0x02 0x03 0x04 0x05 | Timeout address NACKed Timeout bus not free (SCL Low Timeout) Arbitration lost Read incomplete Write incomplete Succeeded after <i>Status 2</i> retries |

For example, if this report returned the following data (in hexadecimal):

16 02 05 00 05

This corresponds to:

16: Report ID (Transfer Status Response)

02: Transmission complete without errors

05: Succeeded after the following bytes retry

00: Indicates zero retries

05: Five bytes were received

5.8. Cancel Transfer

Report ID: 0x17

Direction: Interrupt Out

| Name | Offset | Size | Value | Description |
|--------|--------|------|-------|--|
| Cancel | 1 | 1 | 0x01 | Will cancel the current transfer. All other values are ignored |

Cancel is used to cancel the current SMBus transfer. If any value other than 0x01 is sent with the reset command, the command is ignored.

6. Programmable USB Parameters

The following parameters are programmable on the device. Five different reports are provided to program these parameters. Each parameter modified with these five reports can only be modified once.

| Name | Description |
|---------------------|---|
| VID | USB Vendor ID |
| PID | USB Product ID |
| Power | Power request in mA/2 |
| Power Mode | Bus Powered (0x00) Self Powered - Regulator Off (0x01) Self Powered - Regulator On (0x02) |
| Release Version | Major and Minor release version |
| Manufacturer String | Product Manufacturer |
| Product String | Product Description |
| Serial String | Serialization String |
| Lock Byte | Indicates programmed values |

VID is the USB Vendor ID.

PID is the USB Product ID.

Power is the current requested by the device from the USB host in bus-powered mode. The units for this value are milliamps / 2. For example, if the device is configured to request 200 mA, the value for *Power* is 100. The maximum setting for *Power* is 500 mA, or a value of 250. Unpowered USB hubs are limited to providing 100 mA per port.

Power Mode indicates whether the device is operating in Bus-powered (0x00), Self-powered (0x01, voltage regulator disabled) or Self-powered (0x02, voltage regulator enabled) mode. If the device is configured for Self-powered mode, the value programmed for *Power* is not used during USB enumeration.

Release Version is a user-programmable value. The most significant byte is the Major revision number. The least significant byte in the report is the Minor revision number. Both bytes can be programmed to any value from 0 to 255.

Manufacturing String is a 62-byte string in which the first two bytes must be set according to the USB specification (length, 0x03).

Product String is a 62-byte string in which the first two bytes must be set according to the USB specification (length, 0x03).

Serial String is a 62-byte character array used to provide a unique serial number/string for the device. The first two characters must be set according to the USB specification (length, 0x03).

7. PROM Programming Reports

7.1. Get/Set Lock Byte

Report ID: 0x20

Direction: Feature Request In/Out

| Name | Offset | Size | Value | Description |
|-------------|--------|------|-----------|--|
| Lock Status | 1 | 1 | See Below | Shows which fields have already been programmed. |

The device has a 1-byte field that indicates which of the customizable fields have been programmed. The following table shows the values of the bits:

| Bit Position | Customizable Field |
|--------------|---------------------|
| Bit 0 | VID |
| Bit 1 | PID |
| Bit 2 | Max Power |
| Bit 3 | Power Mode |
| Bit 4 | Release Version |
| Bit 5 | Manufacturer String |
| Bit 6 | Product String |
| Bit 7 | Serial String |

If the bit value is set to 1, the corresponding field has not been customized. If the bit value is set to 0, the field has been customized or locked and can no longer be changed for this device.

Using the Set Lock Byte Report, any bit value set to 0 will lock the corresponding field. Send 0x00 to lock all parameters and prevent future customization.

7.2. Get/Set USB Configuration

Report ID: 0x21

Direction: Feature Request In/Out

| Name | Offset | Size | Value | Description |
|---------------|--------|------|----------------------|--|
| VID Low Byte | 1 | 1 | * | VID Low Byte |
| VID High Byte | 2 | 1 | * | VID High Byte |
| PID Low Byte | 3 | 1 | * | PID Low Byte |
| PID High Byte | 4 | 1 | * | PID High Byte |
| Power | 5 | 1 | * | Power requested in mA/2 |
| Power Mode | 6 | 1 | 0x00 0x01 0x02 | Bus Powered Self Powered - Regulator Off Self Powered - Regulator On |
| Release Major | 7 | 1 | * | Release Version Major Value |
| Release Minor | 8 | 1 | * | Release Version Minor Value |
| Mask | 9 | 1 | * | Mask for what fields to program |

Get USB Configuration returns the values for the various fields and also the *Mask* value. The *Mask* value is equal to the most significant byte value returned in Report *Get Lock Byte*. If the corresponding *Mask* bit is set to “0”, the corresponding field has been programmed, and any *Set USB Configuration* function operating on that field is ignored.

Set USB Configuration is used to customize these fields. The corresponding *Mask* bit should be set to “1” to program the field. If the field has already been programmed once, an attempt to reprogram it is ignored. If a field is being programmed with the current value, the programmed bit will still be set.

See "7.1. Get/Set Lock Byte" on page 17 for the definition of *Mask*.

If a parameter is being modified using this report, the corresponding bit in *Mask* must be set to a 1. For example, if the VID needed to be programmed, bit 0 of *Mask* would need to be set to a 1, and the values in *VID Low Byte* and *VID High Byte* would be written to the CP2112.

7.3. Get/Set Manufacturing String

Report ID: 0x22

Direction: Feature Request In/Out

| Name | Offset | Size | Value | Description |
|----------------------|--------|------|-------|-----------------------------------|
| String Length | 1 | 1 | * | Length of manufacturer string + 2 |
| USB Required | 2 | 1 | 0x03 | Required Value |
| Manufacturing String | 3 | 60 | * | First 60 bytes of string |

The *Set Manufacturing String* Report can only be used once to set the Manufacturing String. Any subsequent calls to *Set Manufacturing String* are ignored.

The maximum value for *String Length* is 60. The first two bytes are allocated for *String Length* and the value 0x03, meaning the actual length of the pstring is 62 bytes. The device will ignore the report if *String Length* is too long.

7.4. Get/Set Product String

Report ID: 0x23

Direction: Feature Request In/Out

| Name | Offset | Size | Value | Description |
|----------------|--------|------|-------|------------------------------|
| String Length | 1 | 1 | * | Length of product string + 2 |
| USB Required | 2 | 1 | 0x03 | Required Value |
| Product String | 3 | 60 | * | First 60 bytes of string |

The *Set Product String* Report can only be used once to set the Product String. Any subsequent calls to *Set Product String* are ignored.

The maximum value for *String Length* is 60. The first two bytes are allocated for *String Length* and the value 0x03, meaning the actual length of the payload part of the string is 62 bytes. The device will ignore the Report if *String Length* is too long.

7.5. Get/Set Serial String

Report ID: 0x24

Direction: Feature Request In/Out

| Name | Offset | Size | Value | Description |
|---------------|--------|------|-------|-----------------------------|
| String Length | 1 | 1 | * | Length of serial string + 2 |
| USB Required | 2 | 1 | 0x03 | |
| Ser String | 3 | 60 | * | 60 bytes of string |

The *Set Serial String* Report can only be used once to set the Product String. Any subsequent calls to *Set Serial String* are ignored.

The maximum value for *String Length* is 60. The first two bytes are allocated for *String Length* and the value 0x03, meaning the actual length of the payload part of the string is 62 bytes. The device will reject the Report if *String Length* is too long.

Some of the pins of the CP2112 are configurable as inputs, open-drain outputs, or push-pull outputs. These options are configured when the device has enumerated and is operating in a normal mode. When the CP2112 is in USB suspend, all of the configurable pins are limited to be open-drain or push-pull outputs. The following describes the differences between open-drain and push-pull, and the difference in behavior in Suspend mode. See the CP2112 data sheet for the electrical specifications of the GPIO pins.

- **GPIO Input**

When a pin is configured as a GPIO input, the pin can read a logic high or logic low value. Internally, the GPIO pin is connected to the VIO pin through a resistor. If the pin is not connected externally, it will return a logic high or 1. Any voltages connected to the pin should conform to data sheet specifications.

- **Open-Drain Output**

When a pin is configured as a GPIO open-drain output, the pin can output a logic high or logic low value. The default value is logic high, and a logic high value is created by internally connecting the GPIO pin to the VIO pin through a resistor. In this mode, the pin is unable to source any current when driving a logic high. If the Set GPIO Values Report is used to change the output to a logic low, the pin is internally connected to GND.

- **Push-Pull Output**

When a pin is configured as a GPIO push-pull output, the pin can output a logic high or logic low value. When driving a logic high value, the pin is directly connected to the VIO pin internally and can source current for devices, such as LEDs. When driving a logic low value, the pin is internally connected to GND.

- **Suspend Mode**

When the device is in Suspend mode, all of the GPIO pins are forced to be open-drain or push-pull outputs. The GPIO pins retain their state during suspend mode.

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- Updated "4.2. Get/Set GPIO Configuration" on page 8 with Special Functions.
- Changed names of GPIO0 to GPIO0_TXT, GPIO1 to GPIO1_RXT, and GPIO7 to GPIO7_CLK.

CONTACT INFORMATION

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Please visit the Silicon Labs Technical Support web page:
<https://www.silabs.com/support/pages/contacttechnicalsupport.aspx>
and register to submit a technical support request.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.