



**SC667/8 Light Management Unit
Software User's Guide and
I²C Interface Specification**

Table of Contents

Introduction.....	4
Applicability	4
I ² C General Specification	4
Supported Formats.....	4
Software Integration.....	6
System Requirements	6
Files	6
General Integration.....	6
Macro Definitions	6
SetSemHigh() and SetSemLow()	6
SC667/8_SLAVE_ADDRESS	6
Semtech_Read I2C (slaveAddress, data, length)	6
Semtech_Write I2C (slaveAddress, data, length)	6
Application Programming Interface.....	7
General Interface.....	7
WriteRegister	7
ReadRegister.....	8
ResetAllRegisters.....	9
EnableDevice	10
DisableDevice.....	10
Backlight Interface	11
EnableBacklights.....	11
DisableBacklights	11
EnableBanks	12
DisableBanks.....	12
SetBacklightCurrentAndEffect	13
LDO Interface	15
SetLDO1Voltage	15
SetLDO2Voltage	15
SetLDO3Voltage	16
SetLDO4Voltage	16

Backlight Effects	17
SetLightAssignments.....	17
SetEffectRate	18
SetEffectTimeGrp1	19
SetEffectTimeGrp2	19
Lighting Assignments	20
SetAdcEnables	20
SetAdcOut	21
SetAdcRiseFall	22
SetAutoDropProtectFunctions.....	22

Introduction

Applicability

This Software User Guide is applicable to both the SC667 and SC668. The SC667 is configured with 7 LED backlights and the SC668 is configured with 8 LED backlights. When using this document with the SC667 device ignore all references to backlight 8.

I²C General Specification

The SC667/8 is a read-write slave-mode I²C device and complies with the Philips I²C standard Version 2.1, dated January 2000. The SC667/8 has 23 user-accessible internal 8-bit registers. The I²C interface has been designed for program flexibility, supporting direct format for write operation. Read operations are supported on both combined format and stop separated format. While there is no auto increment/decrement capability in the SC667/8 I²C logic, a tight software loop can be designed to randomly access the next register independent of which register you begin accessing. The start and stop commands frame the data-packet and the repeat start condition is allowed if necessary.

SC667/8 Limitations to the I²C Specifications

The SC667/8 only recognizes seven bit addressing. This means that ten bit addressing and CBUS communication are not compatible. The device can operate in either standard mode (100kbit/s) or fast mode (400kbit/s).

Slave Address Assignment

The seven bit slave address is 1110 000x. The eighth bit is the data direction bit. E0h is used for a write operation, and E1h is used for a read operation.

Supported Formats

The supported formats are described in the following subsections.

Direct Format — Write

The simplest format for an I²C write is direct format. After the start condition [S], the slave address is sent, followed by an eighth bit indicating a write. The SC667/8 I²C then acknowledges that it is being addressed, and the master

responds with an 8 bit data byte consisting of the register address. The slave acknowledges and the master sends the appropriate 8 bit data byte. Once again the slave acknowledges and the master terminates the transfer with the stop condition [P].

Combined Format — Read

After the start condition [S], the slave address is sent, followed by an eighth bit indicating a write. The SC667/8 I²C then acknowledges that it is being addressed, and the master responds with an 8 bit data byte consisting of the register address. The slave acknowledges and the master sends the repeated start condition [Sr]. Once again, the slave address is sent, followed by an eighth bit indicating a read. The slave responds with an acknowledge and the 8 bit data from the previously addressed register; the master then sends a non-acknowledge (NACK). Finally, the master terminates the transfer with the stop condition [P].

Stop Separated Reads

Stop-separated reads can also be used. This format allows a master to set up the register address pointer for a read and return to that slave at a later time to read the data. In this format the slave address followed by a write command are sent after a start [S] condition. The SC667/8 then acknowledges it is being addressed, and the master responds with the 8-bit register address. The master sends a stop or restart condition and may then address another slave. After performing other tasks, the master can send a start or restart condition to the SC667/8 with a read command. The device acknowledges this request and returns the data from the register location that had previously been set up.

Serial Interface (continued)

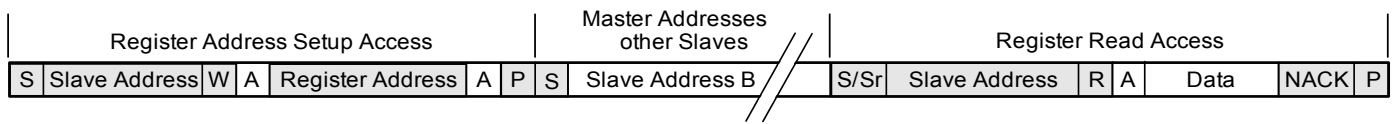
I²C Direct Format Write



S – Start Condition
 W – Write = '0'
 A – Acknowledge (sent by slave)
 P – Stop condition

Slave Address – 7-bit
 Register address – 8-bit
 Data – 8-bit

I²C Stop Separated Format Read



S – Start Condition
 W – Write = '0'
 R – Read = '1'
 A – Acknowledge (sent by slave)
 NAK – Non-Acknowledge (sent by master)
 Sr – Repeated Start condition
 P – Stop condition

Slave Address – 7-bit
 Register address – 8-bit
 Data – 8-bit

I²C Combined Format Read



S – Start Condition
 W – Write = '0'
 R – Read = '1'
 A – Acknowledge (sent by slave)
 NAK – Non-Acknowledge (sent by master)
 Sr – Repeated Start condition
 P – Stop condition

Slave Address – 7-bit
 Register address – 8-bit
 Data – 8-bit

Software Integration

System Requirements

All API functions must be called from within a single thread. Although the SC667/8 is intended for multi threaded systems, the APIs are not designed to be simultaneously called from multiple threads. The SC667/8 APIs are written with the assumption that 8-bit data accesses are atomic for the processor that is executing the code. This means that 8-bit values are written to or read from system memory within a single CPU instruction.

Files

The drivers are contained within three files: `SC668I2C.h`, `SC668I2C.c`, and `semtech_integration.h`. Usage of these files are described in the following table.

File	Usage
<code>sc668I2C.h</code>	Contains all prototypes and data types required to use the SC667/8 drivers. This file is intended to be included within the files that access the SC667/8 APIs.
<code>sc668I2C.c</code>	Contains the implementation of the SC667/8 APIs and drivers. This file does not need to be modified for typical usage models.
<code>semtech_integration.h</code>	Contains macros that must be defined in order to allow the SC667/8 drivers to access the system hardware and resources. This file must be modified during the system integration process. This file should not be included within any file other than <code>sempulse.c</code> .

General Integration

Macro Definitions

The macros in this section must be implemented for successful operation.

SetSemHigh() and SetSemLow()

The `SetSemHigh()` and `SetSemLow()` macros are used to drive the EN pin of the Semtech chip high and low respectively. This enables or disables the entire chip.

SC667/8_SLAVE_ADDRESS

The `SC668_SLAVE_ADDRESS` sets the slave address of the SC667/8.

Semtech_Read I2C (slaveAddress, data, length)

`Semtech_ReadI2C` is a macro that reads from the current I²C address into `data`. `slaveAddress` and `length` should be of type `unsigned char` and `data` should be an `unsigned char *`. A call to `Semtech_WriteI2C` is used to set the address before calling `Semtech_ReadI2C`.

Semtech_Write I²C (slaveAddress, data, length)

`Semtech_WriteI2C` is a macro that writes `length` bytes of data over I²C. `slaveAddress` and `length` should be of type `unsigned char` and `data` should be an `unsigned char *`.

Application Programming Interface

General Interface

The SC667/8 is provided with reference drivers facilitating complete control of the device. A low-level interface provides the means to directly write registers. A high-level interface allows control of features and is abstracted from the register-level accesses required to enact the features. All names specific to the drivers are shown in the *Courier New* font. When SC667/8 is shown in the text use one of the following — SC667 or SC668 .

WriteRegister

The `WriteRegister` function is the lowest level API provided. All functionality of the chip may be accessed through this single function which writes data to the specified register. The function verifies that the register is within the valid range before attempting the write. In addition, all reserved bits are masked off before the data is transferred.

Prototype

```
Semtech_StatusCode WriteRegister(SC667/8_Register address, unsigned char data)
```

Parameters

address	<p>The register to write to is the lowest level API provided. This must be one of the following values:</p> <pre> BACKLIGHT_1TO6_ENABLE BACKLIGHT_7TO8_AND_BANK_ENABLE BANK1_CONTROL BANK2_CONTROL BANK3_CONTROL BANK4_CONTROL BANK1_SETTINGS BANK2_SETTINGS BANK3_SETTINGS BANK4_SETTINGS LDO1_CONTROL LDO2_CONTROL LDO3_CONTROL LDO4_CONTROL LIGHTING_ASSIGNMENTS EFFECT_RATE_SETTINGS EFFECT_RATE1_TIME EFFECT_RATE2_TIME ADC_ENABLE ADC_OUTPUT ADC_RISE ADC_FALL ADP_OLE_CONTROLS </pre>
Data	The data to be written to the register.

Return Value

SEMTECH_SUCCESS	The data has been written to the register of the SC667/8.
SEMTECH_INVALID_ADDRESS	The register is outside of the valid range and the data will not be sent to the SC667/8.
SEMTECH_RESERVED_BITS_EXCLUDED	The data contains bits that are reserved. These bits were cleared and the remaining portion of the data has been written to the register of the SC667/8.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

Application Programming Interface (continued)

ReadRegister

The `ReadRegister` function is the lowest level API provided. All functionality of the chip may be accessed through this single function which writes data to the specified register. The function verifies that the register is within the valid range before attempting the read. In addition, all reserved bits are masked off before the data is transferred.

Prototype

```
unsigned char ReadRegister(SC667/8_Register address)
```

Parameter

address	<p>The register to write to is the lowest level API provided. This must be one of the following values:</p> <ul style="list-style-type: none"> BACKLIGHT_1TO6_ENABLE BACKLIGHT_7TO8_AND_BANK_ENABLE BANK1_CONTROL BANK2_CONTROL BANK3_CONTROL BANK4_CONTROL BANK1_SETTINGS BANK2_SETTINGS BANK3_SETTINGS BANK4_SETTINGS LDO1_CONTROL LDO2_CONTROL LDO3_CONTROL LDO4_CONTROL LIGHTING_ASSIGNMENTS EFFECT_RATE_SETTINGS EFFECT_RATE1_TIME EFFECT_RATE2_TIME ADC_ENABLE ADC_OUTPUT ADC_RISE ADC_FALL ADP_OLE_CONTROLS
---------	---

Return Value

The data read from the register

Application Programming Interface (continued)

ResetAllRegisters

The `ResetAllRegisters` function sets all of the registers of the SC667/8 back to the initial power up state. The reset state of the registers is as follows.

Address	Register Name	Reset Value
0x00	Backlight enable for BL1 — BL6	0x00
0x01	Bank enable, plus backlight enable for BL7 — BL8 (see note)	0x00
0x02	Fade enable and bank #1 backlight current	0x00
0x03	Fade enable and bank #2 backlight current	0x00
0x04	Fade enable and bank #3 backlight current	0x00
0x05	Fade enable and bank #4 backlight current	0x00
0x06	Blink/breathe bank #1 target backlight settings	0x00
0x07	Blink/breathe bank #2 target backlight settings	0x00
0x08	Blink/breathe bank #3 target backlight settings	0x00
0x09	Blink/breathe bank #4 target backlight settings	0x00
0x0A	LDO1 voltage settings	0x00
0x0B	LDO2 voltage settings	0x00
0x0C	LDO3 voltage settings	0x00
0x0D	LDO4 voltage settings	0x00
0x0E	Lighting effects assignments, banks & groups	0x00
0x0F	Effect rates for group #1 and group #2	0x00
0x10	Target time and start time for group #1	0x00
0x11	Target time and start time for group #2	0x00
0x12	ALS function	0x00
0x13	ADC output AD_{OUT}	0x00
0x14	ADC rising threshold AD_{RISE}	0x00
0x15	ADC falling threshold AD_{FALL}	0x00
0x16	Other lighting effects — auto-dim full or partial, auto-dim enable	0x00

NOTE: BL8 not available on SC667

Prototype

```
Semtech_ResultCode ResetAllRegisters(void)
```

Parameters

None

Return Value

SEMTECH_SUCCESS	The registers of the SC667/8 have been reset.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and some of the registers on the SC667/8 have not been reset. This result code is only possible in Scheduled mode.

Application Programming Interface (continued)

EnableDevice

The `EnableDevice` function pulls the enabling pin of the SC667/8 high for at least 1ms.

Prototype

```
Semtech_ResultCode EnableDevice (void)
```

Parameters

None

Return Value

SEMTECH_SUCCESS	The SC667/8 is enabled.
-----------------	-------------------------

DisableDevice

The `DisableDevice` function pulls the enable pin of the SC667/8 low for at least ten milliseconds.

Prototype

```
Semtech_ResultCode DisableDevice (void)
```

Parameters

None

Return Value

SEMTECH_SUCCESS	The SC667/8 is disabled.
-----------------	--------------------------

Backlight Interface

EnableBacklights

The `EnableBacklights` function is used to enable a subset of backlights without affecting the state of the others.

Prototype

```
Semtech_ResultCode EnableBacklights(unsigned char backlights)
```

Parameter

<code>backlights</code>	<p>The bits corresponding to the backlights which are to be enabled are ORed together to create the <code>backlights</code> parameter. The state of all backlights, not included in this parameter are unaffected. This parameter must be set to the bitwise ORed results of none, any, or all of the following values.</p> <ul style="list-style-type: none"> <code>BACKLIGHT_8</code> <code>BACKLIGHT_7</code> <code>BACKLIGHT_6</code> <code>BACKLIGHT_5</code> <code>BACKLIGHT_4</code> <code>BACKLIGHT_3</code> <code>BACKLIGHT_2</code> <code>BACKLIGHT_1</code>
-------------------------	--

Return Value

<code>SEMTECH_SUCCESS</code>	The backlight settings have been successfully changed.
<code>SEMTECH_BUFFER_ERROR</code>	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

DisableBacklights

The `DisableBacklights` function is used to disable a subset of backlights without affecting the state of the others.

Prototype

```
Semtech_ResultCode DisableBacklights (unsigned char backlights)
```

Parameter

<code>backlights</code>	<p>The bits corresponding to the backlights which are to be disabled are ORed together to create the <code>backlights</code> parameter. The state of all backlights, not included in this parameter are unaffected. This parameter must be set to the bitwise ORed results of none, any, or all of the following values.</p> <ul style="list-style-type: none"> <code>BACKLIGHT_8</code> <code>BACKLIGHT_7</code> <code>BACKLIGHT_6</code> <code>BACKLIGHT_5</code> <code>BACKLIGHT_4</code> <code>BACKLIGHT_3</code> <code>BACKLIGHT_2</code> <code>BACKLIGHT_1</code>
-------------------------	---

Return Value

<code>SEMTECH_SUCCESS</code>	The backlight settings have been successfully changed.
<code>SEMTECH_BUFFER_ERROR</code>	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

Backlight Interface (continued)

EnableBanks

The `EnableBanks` function is used to enable a subset of banks.

Prototype

```
Semtech_ResultCode EnableBanks (unsigned char bank)
```

Parameter

bank	The bits corresponding to the banks which are to be enabled are ORed together to create the <code>bank</code> parameter. The state of all banks, not included in this parameter are unaffected. This parameter must be set to the bitwise ORed results of none, any, or all of the following values. <code>BANK_4_ENABLE</code> <code>BANK_3_ENABLE</code> <code>BANK_2_ENABLE</code> <code>BANK_1_ENABLE</code>
------	--

Return Value

<code>SEMTECH_SUCCESS</code>	The bank settings have been successfully changed.
<code>SEMTECH_INVALID_PARAMETER</code>	The <code>bank</code> parameter contains bits which should not be sent. These bits have been masked off and the remaining bits have been used.
<code>SEMTECH_BUFFER_ERROR</code>	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

DisableBanks

The `DisableBanks` function is used to enable a subset of banks.

Prototype

```
Semtech_ResultCode DisableBanks (unsigned char bank)
```

Parameter

bank	The bits corresponding to the banks which are to be disabled are ORed together to create the <code>bank</code> parameter. The state of all banks, not included in this parameter are unaffected. This parameter must be set to the bitwise ORed results of none, any, or all of the following values. <code>BANK_4_ENABLE</code> <code>BANK_3_ENABLE</code> <code>BANK_2_ENABLE</code> <code>BANK_1_ENABLE</code>
------	---

Return Value

<code>SEMTECH_SUCCESS</code>	The bank settings have been successfully changed.
<code>SEMTECH_INVALID_PARAMETER</code>	The <code>bank</code> parameter contains bits which should not be sent. These bits have been masked off and the remaining bits have been used.
<code>SEMTECH_BUFFER_ERROR</code>	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

Backlight Interface (continued)

SetBacklightCurrentAndEffect

The `SetBacklightCurrentAndEffect` function sets the backlight current and target current and also enables fade, blink, breathe, or no effect on a specified bank.

Prototype

```
Semtech_ResultCode SetBacklightCurrentAndEffect (BankType bank, unsigned char current, unsigned char targetCurrent, unsigned char effect)
```

Parameters

bank	<p>The specified bank used for enabling current, target current and light effect.</p> <p>BANK_1 BANK_2 BANK_3 BANK_4</p>
current	<p>Sets the backlight current. This parameter must be set to one of the following values.</p> <p>BACKLIGHT_CURRENT_0_0 BACKLIGHT_CURRENT_0_0_5 BACKLIGHT_CURRENT_0_1 BACKLIGHT_CURRENT_0_2 BACKLIGHT_CURRENT_0_5 BACKLIGHT_CURRENT_1_0 BACKLIGHT_CURRENT_1_5 BACKLIGHT_CURRENT_2_0 BACKLIGHT_CURRENT_2_5 BACKLIGHT_CURRENT_3_0 BACKLIGHT_CURRENT_3_5 BACKLIGHT_CURRENT_4_0 BACKLIGHT_CURRENT_4_5 BACKLIGHT_CURRENT_5_0 BACKLIGHT_CURRENT_6_0 BACKLIGHT_CURRENT_7_0 BACKLIGHT_CURRENT_8_0 BACKLIGHT_CURRENT_9_0 BACKLIGHT_CURRENT_10_0 BACKLIGHT_CURRENT_11_0 BACKLIGHT_CURRENT_12_0 BACKLIGHT_CURRENT_13_0 BACKLIGHT_CURRENT_14_0 BACKLIGHT_CURRENT_15_0 BACKLIGHT_CURRENT_16_0 BACKLIGHT_CURRENT_17_0 BACKLIGHT_CURRENT_18_0 BACKLIGHT_CURRENT_19_0 BACKLIGHT_CURRENT_20_0 BACKLIGHT_CURRENT_21_0 BACKLIGHT_CURRENT_23_0 BACKLIGHT_CURRENT_25_0</p>

Backlight Interface (continued)

<p>targetCurrent</p>	<p>Sets the target backlight current. This parameter must be set to one of the following values.</p> <ul style="list-style-type: none"> BACKLIGHT_CURRENT_0_0 BACKLIGHT_CURRENT_0_0_5 BACKLIGHT_CURRENT_0_1 BACKLIGHT_CURRENT_0_2 BACKLIGHT_CURRENT_0_5 BACKLIGHT_CURRENT_1_0 BACKLIGHT_CURRENT_1_5 BACKLIGHT_CURRENT_2_0 BACKLIGHT_CURRENT_2_5 BACKLIGHT_CURRENT_3_0 BACKLIGHT_CURRENT_3_5 BACKLIGHT_CURRENT_4_0 BACKLIGHT_CURRENT_4_5 BACKLIGHT_CURRENT_5_0 BACKLIGHT_CURRENT_6_0 BACKLIGHT_CURRENT_7_0 BACKLIGHT_CURRENT_8_0 BACKLIGHT_CURRENT_9_0 BACKLIGHT_CURRENT_10_0 BACKLIGHT_CURRENT_11_0 BACKLIGHT_CURRENT_12_0 BACKLIGHT_CURRENT_13_0 BACKLIGHT_CURRENT_14_0 BACKLIGHT_CURRENT_15_0 BACKLIGHT_CURRENT_16_0 BACKLIGHT_CURRENT_17_0 BACKLIGHT_CURRENT_18_0 BACKLIGHT_CURRENT_19_0 BACKLIGHT_CURRENT_20_0 BACKLIGHT_CURRENT_21_0 BACKLIGHT_CURRENT_23_0 BACKLIGHT_CURRENT_25_0
<p>effect</p>	<p>Enables fade, breathe, blink or no effect. This parameter must be set to one of the following values.</p> <ul style="list-style-type: none"> ENABLE_FADE ENABLE_BREATHE ENABLE_BLINK NO_EFFECT

Return Value

<p>SEMTECH_SUCCESS</p>	<p>The settings have been successfully changed.</p>
<p>SEMTECH_INVALID_PARAMETER</p>	<p>The bank, current, or targetCurrent parameter contains invalid bits. The register has not been changed.</p>
<p>SEMTECH_BUFFER_ERROR</p>	<p>A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.</p>

LDO Interface

SetLDO1Voltage

The `SetLDO1Voltage` function sets the voltage output of LDO1. The LDO may be set in increments between 1.5V and 3.3V or turned off.

Prototype

```
Semtech_ResultCode SetLDO1Voltage (LDO_Voltage voltage)
```

Parameter

voltage	<p>Sets the voltage of LDO1. This parameter must be set to one of the following values.</p> <pre>LDO_OFF LDO_VOLTAGE_3_3 LDO_VOLTAGE_3_2 LDO_VOLTAGE_3_1 LDO_VOLTAGE_3_0 LDO_VOLTAGE_2_9 LDO_VOLTAGE_2_8 LD_VOLTAGE_2_7 LDO_VOLTAGE_2_6 LDO_VOLTAGE_2_5 LDO_VOLTAGE_2_4 LDO_VOLTAGE_2_2 LDO_VOLTAGE_1_8 LDO_VOLTAGE_1_7 LDO_VOLTAGE_1_6 LDO_VOLTAGE_1_5</pre>
---------	---

Return Value

SEMTECH_SUCCESS	The LDO settings have been successfully changed.
SEMTECH_INVALID_PARAMETER	The <code>voltage</code> parameter is outside of the valid range. The LDO voltage has not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

SetLDO2Voltage

The `SetLDO2Voltage` function sets the voltage output of LDO2. The LDO may be set in 0.1V increments between 1.2V and 1.8V or turned off.

Prototype

```
Semtech_ResultCode SetLDO2Voltage (LDO2_Voltage voltage)
```

Parameter

voltage	<p>Sets the voltage of LDO2. This parameter must be set to one of the following values.</p> <pre>LDO2_OFF LDO2_VOLTAGE_1_8 LDO2_VOLTAGE_1_7 LDO2_VOLTAGE_1_6 LDO2_VOLTAGE_1_5 LDO2_VOLTAGE_1_4 LDO2_VOLTAGE_1_3 LDO2_VOLTAGE_1_2</pre>
---------	--

LDO Interface (continued)

SetLDO2Voltage (continued)

Return Value

SEMTECH_SUCCESS	The LDO settings have been successfully changed.
SEMTECH_INVALID_PARAMETER	The voltage parameter is outside of the valid range. The LDO voltage has not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

SetLDO3Voltage

The `SetLDO3Voltage` function sets the voltage output of LDO3. The LDO may be set in increments between 1.5V and 3.3V or turned off.

Prototype

```
Semtech_ResultCode SetLDO3Voltage (LDO_Voltage voltage)
```

Parameter

<p>voltage</p>	<p>Sets the voltage of LDO3. This parameter must be set to one of the following values.</p> <ul style="list-style-type: none"> LDO_OFF LDO_VOLTAGE_3_3 LDO_VOLTAGE_3_2 LDO_VOLTAGE_3_1 LDO_VOLTAGE_3_0 LDO_VOLTAGE_2_9 LDO_VOLTAGE_2_8 LDO_VOLTAGE_2_7 LDO_VOLTAGE_2_6 LDO_VOLTAGE_2_5 LDO_VOLTAGE_2_4 LDO_VOLTAGE_2_2 LDO_VOLTAGE_1_8 LDO_VOLTAGE_1_7 LDO_VOLTAGE_1_6 LDO_VOLTAGE_1_5
----------------	--

Return Value

SEMTECH_SUCCESS	The LDO settings have been successfully changed.
SEMTECH_INVALID_PARAMETER	The voltage parameter is outside of the valid range. The LDO voltage has not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

SetLDO4Voltage

The `SetLDO4Voltage` function sets the voltage output of LDO4. The LDO may be set in increments between 1.5V and 3.3V or turned off.

Prototype

```
Semtech_ResultCode SetLDO4Voltage (LDO_Voltage voltage)
```


LDO Interface (continued)

SetLDO4Voltage (continued)

Parameter

voltage	<p>Sets the voltage of LDO4. This parameter must be set to one of the following values.</p> <pre> LDO_OFF LDO_VOLTAGE_3_3 LDO_VOLTAGE_3_2 LDO_VOLTAGE_3_1 LDO_VOLTAGE_3_0 LDO_VOLTAGE_2_9 LDO_VOLTAGE_2_8 LDO_VOLTAGE_2_7 LDO_VOLTAGE_2_6 LDO_VOLTAGE_2_5 LDO_VOLTAGE_2_4 LDO_VOLTAGE_2_2 LDO_VOLTAGE_1_8 LDO_VOLTAGE_1_7 LDO_VOLTAGE_1_6 LDO_VOLTAGE_1_5 </pre>
---------	--

Return Value

SEMTECH_SUCCESS	The LDO settings have been successfully changed.
SEMTECH_INVALID_PARAMETER	The voltage parameter is outside of the valid range. The LDO voltage has not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

Backlight Effects

SetLightAssignments

The `SetLightAssignments` function assigns individual lights to four different banks and then assigns each bank to a group.

Prototype

```
Semtech_ResultCode SetLightAssignments(unsigned char bankAssignment, unsigned char groupAssignment)
```

Parameters

bankAssignment	<p>Use the following values to assign lights to banks. The first value assigns backlights 1 through 8 to bank #1. The second value assigns backlights 2 through 8 to bank #1 and backlight 1 to bank #2 etc.</p> <pre> BANK1_BL1_BL8 BANK1_BL2_BL8_BANK2_BL1 BANK1_BL3_BL8_BANK2_BL1_BL2 BANK1_BL3_BL8_BANK2_BL2_BANK3_BL1 BANK1_BL4_BL8_BANK2_BL1_BL3 BANK1_BL4_BL8_BANK2_BL3_BANK3_BL2_BANK4_BL1 BANK1_BL5_BL8_BANK2_BL3_BL4_BANK3_BL2_BANK4_BL1 BANK1_BL6_BL8_BANK2_BL3_BL5_BANK3_BL2_BANK4_BL1 </pre>
----------------	---

Backlight Effects (continued)

groupAssignment	<p>Use the following values to assign banks to two different groups. The first value assigns bank #1 to group #1 and bank #2 through bank #4 to group #2. The second value assigns bank #1 and bank #2 to group #1 and bank #3 and bank #4 to group #2, etc.</p> <pre>GROUP1_BK1_GROUP2_BK2_BK3_BK4 GROUP1_BK1_BK2_GROUP2_BK3_BK4 GROUP1_BK1_BK2_BK3_GROUP2_BK4 GROUP1_BK1_BK2_BK3_BK4</pre>
-----------------	--

Return Value

SEMTECH_SUCCESS	The light assignment settings have been successfully changed.
SEMTECH_INVALID_PARAMETER	The bankAssignment and/or groupAssignment parameter is outside of the valid range. The light assignment has not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

SetEffectRate

The SetEffectRate function sets the breathe and fade effect rates for group #1 and group #2 lights.

Prototype

```
Semtech_ResultCode SetEffectRate(unsigned char effectGrp1, unsigned char effectGrp2)
```

Parameters

effectGrp1	<p>Use the following values to set the breathe and fade rates on group #1 assigned lights. The first value sets the breathe and fade rates to zero. The second value sets the breathe rate to 4ms and sets the fade rate to 1ms, etc.</p> <pre>BREATH_0_FADE_0 BREATH_4_FADE_1 BREATH_8_FADE_2 BREATH_16_FADE_4 BREATH_24_FADE_6 BREATH_32_FADE_8 BREATH_48_FADE_12 BREATH_64_FADE_16</pre>
effectGrp2	<p>Use the following values to set the breathe and fade rates on group #2 assigned lights. The first value sets the breathe and fade rates to zero. The second value sets the breathe rate to 4ms and sets the fade rate to 1ms, etc.</p> <pre>BREATH_0_FADE_0 BREATH_4_FADE_1 BREATH_8_FADE_2 BREATH_16_FADE_4 BREATH_24_FADE_6 BREATH_32_FADE_8 BREATH_48_FADE_12 BREATH_64_FADE_16</pre>

Return Value

SEMTECH_SUCCESS	The effect rate register 0Fh has been successfully changed.
SEMTECH_INVALID_PARAMETER	The effectGrp1 and/or effectGrp2 parameter is outside of the valid range. The effect rate has not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

Backlight Effects (continued)

SetEffectTimeGrp1

The `SetEffectTimeGrp1` function sets the time durations that the backlights will stay at the start value and at the target value. This feature is an additional way to customize the breathe, and blink lighting effects, by pausing the brightness at the beginning and at the end of each lighting effect sequence. This function only effects group #1 assigned lights.

Prototype

```
Semtech_StatusCode SetEffectTimeGrp1(unsigned char startTime, unsigned char targetTime)
```

Parameters

<code>startTime</code>	<p>Use one of the following values to set the start time to one of eight values between 32ms and 4096ms. The first value assigns <code>startTime</code> to 32ms. The second value assigns <code>startTime</code> to 64ms, etc.</p> <ul style="list-style-type: none"> <code>TIME_32</code> <code>TIME_64</code> <code>TIME_256</code> <code>TIME_512</code> <code>TIME_1024</code> <code>TIME_2048</code> <code>TIME_3072</code> <code>TIME_4096</code>
<code>targetTime</code>	<p>Use one of the following values to set the target time to one of eight values between 32ms and 4096ms. The first value assigns <code>targetTime</code> to 32ms. The second value assigns <code>targetTime</code> to 64ms, etc.</p> <ul style="list-style-type: none"> <code>TIME_32</code> <code>TIME_64</code> <code>TIME_256</code> <code>TIME_512</code> <code>TIME_1024</code> <code>TIME_2048</code> <code>TIME_3072</code> <code>TIME_4096</code>

Return Value

<code>SEMTECH_SUCCESS</code>	<p>The target and start time register for group #1 (10h) has been successfully changed.</p>
<code>SEMTECH_INVALID_PARAMETER</code>	<p>The <code>startTime</code> and/or the <code>targetTime</code> parameter is outside of the valid range. The effect time has not been changed.</p>
<code>SEMTECH_BUFFER_ERROR</code>	<p>A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.</p>

SetEffectTimeGrp2

The `SetEffectTimeGrp2` function sets the time durations that the backlights will stay at the start value and at the target value. This feature is an additional way to customize the breathe, and blink lighting effects, by pausing the brightness at the beginning and at the end of each lighting effect sequence. This function only effects group #2 assigned lights.

Prototype

```
Semtech_StatusCode SetEffectTimeGrp2(unsigned char startTime, unsigned char targetTime)
```

SetEffectTimeGrp2 (continued)

Backlight Effects (continued)

Parameters

startTime	Use one of the following values to set the start time to one of eight values between 32ms and 4096ms. The first value assigns <code>startTime</code> to 32ms. The second value assigns <code>startTime</code> to 64ms, etc. TIME_32 TIME_64 TIME_256 TIME_512 TIME_1024 TIME_2048 TIME_3072 TIME_4096
targetTime	Use one of the following values to set the target time to one of eight values between 32ms and 4096ms. The first value assigns <code>targetTime</code> to 32ms. The second value assigns <code>targetTime</code> to 64ms, etc. TIME_32 TIME_64 TIME_256 TIME_512 TIME_1024 TIME_2048 TIME_3072 TIME_4096

Return Value

SEMTECH_SUCCESS	The target and start time register for group #2 (11h) has been successfully changed.
SEMTECH_INVALID_PARAMETER	The <code>startTime</code> and/or the <code>targetTime</code> parameter is outside of the valid range. The effect time has not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

Lighting Assignments

SetAdcEnables

The `SetAdcEnables` function sets the ALS function register 12h. It allows the user to enable or disable the ADC output register 13h. It also lets the user enable or disable automatic brightness control of bank #1 assigned lights. It is also used to enable or disable the `AD_SATEN` bit. When set, this bit allows an underflow or overflow to generate an interrupt which sets the `AD_INT` bit.

Prototype

```
Semtech_ResultCode SetAdcEnables(unsigned char enable, unsigned char autoBrightControl, unsigned char sat)
```

SetAdcEnables (continued)

Parameters

Lighting Assignments (continued)

enable	This parameter enables or disables the ADC output. If <code>AD_ENABLE</code> is selected, the value can be read out of the ADC output register 13h. Use one of the following values. <code>AD_ENABLE</code> <code>AD_DISABLE</code>
autoBrightControl	This parameter enables or disables automatic control to the brightness of bank #1 assigned lights. Use the following values. <code>AD_AUTO_ENABLE</code> <code>AD_AUTO_DISABLE</code>
sat	This parameter enables or disables the <code>AD_SATEN</code> bit. If enabled, <code>AD_SATEN</code> allows <code>AN_INT</code> to indicate if an overflow or underflow occurs in the ADC output register 13h. Use one of the following values. <code>AD_SAT_ENABLE</code> <code>AD_SAT_DISABLE</code>

Return Value

<code>SEMTECH_SUCCESS</code>	The ALS register 12h has been successfully changed.
<code>SEMTECH_INVALID_PARAMETER</code>	The <code>enable</code> and/or the <code>autoBrightControl</code> and/or the <code>sat</code> parameter is outside of the valid range. The rise and fall threshold registers have not been changed.
<code>SEMTECH_BUFFER_ERROR</code>	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

SetAdcOut

The `SetAdcOut` function sets the ADC output. This function can only be used when the `AD_EN` bit is 0 in register 12h. Before using this function, use the `SetAdcEnables` function to set parameter `enable` to `AD_DISABLE`. This is necessary to disable ADC values from being written to register 13h by the SC667/8.

Prototype

```
Semtech_StatusCode SetAdcOut(unsigned char adcOut)
```

Parameter

adcOut	Set the 8-bit value of ADC output in register 13h.
--------	--

Return Value

<code>SEMTECH_SUCCESS</code>	Register 13h has been successfully changed.
<code>SEMTECH_INVALID_PARAMETER</code>	The <code>enable</code> and/or the <code>autoBrightControl</code> and/or the <code>sat</code> parameter is outside of the valid range. The rise and fall threshold registers have not been changed.
<code>SEMTECH_BUFFER_ERROR</code>	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

SetAdcRiseFall

The `SetAdcRiseFall` function sets the ADC rise and fall thresholds.

Lighting Assignments (continued)

Prototype

```
Semtech_ResultCode SetAdcRiseFall(unsigned char rise, unsigned char fall)
```

Parameters

rise	Set the 8-bit rise threshold of the ADC.
fall	Set the 8-bit fall threshold of the ADC.

Return Value

SEMTECH_SUCCESS	Register 14h and 15h has been successfully changed.
SEMTECH_INVALID_PARAMETER	The rise and/or fall parameter is outside of the valid range. The rise and fall threshold registers have not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

SetAutoDropProtectFunctions

The `SetAutoDropProtectFunctions` sets ADP (Automatic Dropout Prevention) parameters of register 16h.

Prototype

```
Semtech_ResultCode SetAutoDropProtectFunctions(unsigned char autoDropActivation, unsigned char currentReduceTime, unsigned char enable, unsigned char lightEffect, unsigned char pwm)
```

Parameters

autoDropActivation	Activates the ADP feature. Use one of the following values. ADP_ACTIVATE ADP_DEACTIVATE
currentReduceTime	The time elapsed before backlight current reduces as a result of activation started. Use one of the following values. MSEC_4_DELAY MICROSEC_256_DELAY
enable	Enables or disables the ADP. Use one of the following values. ADP_ENABLE ADP_DISABLE
lightEffect	Sets lighting effects on group #1 or group #2 assigned lights. Use one of the following values. OLE_NONE_GRP1 OLE_FULL_AUTO_DIM_GRP1 OLE_PART_AUTO_DIM_GRP1 OLE_SYNCH_GRP1 OLE_NONE_GRP2 OLE_FULL_AUTO_DIM_GRP2 OLE_PART_AUTO_DIM_GRP2 OLE_SYNCH_GRP2
pwm	Enables the PWM bit. Use one of the following values. PWM_ENABLE PWM_DISABLE

Return Value

SEMTECH_SUCCESS	Register 14h and 15h have been successfully changed.
-----------------	--

Lighting Assignments (continued)

SEMTECH_INVALID_PARAMETER	The <code>rise</code> and/or <code>fall</code> parameter is outside of the valid range. The rise and fall threshold registers have not been changed.
SEMTECH_BUFFER_ERROR	A buffer overflow has occurred within the driver and the data has not been sent to the SC667/8. This result code is only possible in Scheduled mode.

© Semtech 2010

All rights reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights. Semtech assumes no responsibility or liability whatsoever for any failure or unexpected operation resulting from misuse, neglect improper installation, repair or improper handling or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified range.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

Contact Information

Semtech Corporation
Power Management Products Division
200 Flynn Road, Camarillo, CA 93012
Phone: (805) 498-2111 Fax: (805) 498-3804

www.semtech.com