Product Document

Published by
ams OSRAM Group

TAOS Inc.

is now

ams AG

The technical content of this TAOS document is still valid.

Contact information:

Headquarters:
ams AG
Tobelbader Strasse 30
8141 Premstaetten, Austria
Tel: +43 (0) 3136 500 0
e-Mail: ams_sales@ams.com

Please visit our website at **www.ams.com**

DESIGNER'S NOTEBOOK

TSL258x: Accurate ADC Readings after Enable

Florencio Villasenor Jr.

August 2012
Keywords: TSL2581, TSL2583

Abstract

This Designer's Notebook will describe how to quickly get valid data from the TSL258x after enabling the device. This process also will demonstrate how to set the integration time, set the gain, set the threshold values, set the interrupt persistence values to any desired value, clear the interrupt, and read the ADC data.

System Considerations

There are three system design considerations to take into account when using this device (from the data sheet):

- The initial Channel 0 and Channel 1 ADC values (0x14~0x17) after ADC_EN is asserted should be discarded because the first ADC values are affected by transients associated with the power up of the analog circuitry.
- It is recommended that the ADC_EN be disabled before changing the analog gain (GAIN) in the Analog register (0x07) because the change will affect the integration in progress, yielding an indeterminate result.
- The integration time (ITIME) in the Timing register (0x01) can be changed at any time; however, the change will take effect only upon completion of the current ALS cycle.

Pseudo Code Description

In the following pseudo code, **`read`** and **`write`** are I2C functions that read and write to a slave device at an address of 0x29, 0x39 or 0x49 depending on whether the ADDR SEL (pin 2) is tied to GND, floating or VDD.

I2C WRITE

The I2C write will send the device address (shifted one bit to the left) with the least significant bit (lsb) set to zero to indicate a write.

Next, the CONTROL register is written. The most significant bit (msb) is set (0x80) when addressing the CONTROL register. The TRANSACTION field is set to 0x20 to auto-increment. Auto-increment will write the first data byte to the starting register address that is in the ADDRESS field, and write subsequent bytes to the subsequent register addresses. The ADDRESS field will have the starting register address.

To clear the ADC_INTR bit in the CONTROL register (bit 5), the TRANSACTION field must be 0x60 and the ADDRESS field must be 0x01.

I2C READ

To do an I2C read, the start register address must be written as described in I2C write above. Then the slave device address is written, shifted one bit to the left, with the msb set to one to indicate a read. If the TRANSACTION field is set to 0x20 (auto-increment mode), the device will return bytes starting with the register address written during the write portion, otherwise a single byte is returned.

Description of Initialization

After power up, set THL and THH registers are set to zero which will ensure an interrupt, set the TIMING register to the minimum integration cycle, set the ANALOG GAIN register to maximum value, and set the device to interrupt after two integration cycles out of range. Immediately after enabling the ADC, the integration cycle time can be changed to the desired value. The device will interrupt after the desired integration cycle time plus the minimum integration cycle time (2.7 ms). After the first interrupt, change the threshold values and the INTERRUPT register to the desired values.

Pseudo Code Example

```
-- Pseudo code to initialize TSL258x/CT81x to output readings after 100 ms integration.
-- Will interrupt after five integration cycles are out of range.
-- The initialization data may be written in a single I2C transaction.

Write 0x01 to the CONTROL (r0x00) register, set the POWER bit (b1)
Write 0xFF to the TIMING (r0x01) register, set to 2.7 ms
Write 0x12 to the INTERRUPT register (r0x02), set the PERSIT (b3-0)to 2 and set the INTR(b5-4) to 1,
interrupt after two integration cycles out of range
Write 0x0000 to the THL (r0x03, r0x04) registers, to ensure an interrupt
Write 0x0000 to the THH (r0x05, r0x06) registers, to ensure an interrupt
Write 0x03 to the ANALOG (r0x07) register, set the GAIN (b2-0) to 111x
Write 0x03 to the CONTROL (r0x00) register, set the ADC_EN (b1) bit in leaving the POWER (b1) bit set
(Enable ADC + Power on)
Write 0xDB to the TIMING (r0x01) register, set to 100 ms or to the desired value

-- Wait for interrupt (102.7 ms, maximum, depends on TIMING register value)

Write 0xE1 to COMMAND register to clear the interrupt

Read DATA0 registers (r0x014, r0x15, auto increment mode, two bytes)
Read DATA1 registers (r0x016, r0x17, auto increment mode, two bytes)

-- Set to THL, THH and PERSIST to the desired value
-- For example, PERSIST = 5, THL = 10000 and THH = 28000

Write 0x15 to INTERRUPT register (r0x02), interrupt after five integration cycles out of range
Write 10000 (0x2710) to THL registers (r0x03, r0x04, autoincrement mode)
Write 28000 (0x6D60) to THH registers (r0x05, r0x06, autoincrement mode)

A:
-- Wait for interrupt (500 ms, minimum, depends on TIMING, INTERRUPT, THL and THH register values)

Write 0xE1 to COMMAND register to clear interrupt

Read DATA0 registers (r0x014, r0x15, auto increment mode, two bytes)
Read DATA1 registers (r0x016, r0x17, auto increment mode, two bytes)

goto A:
```

The following is a timing diagram of the previous sequence.

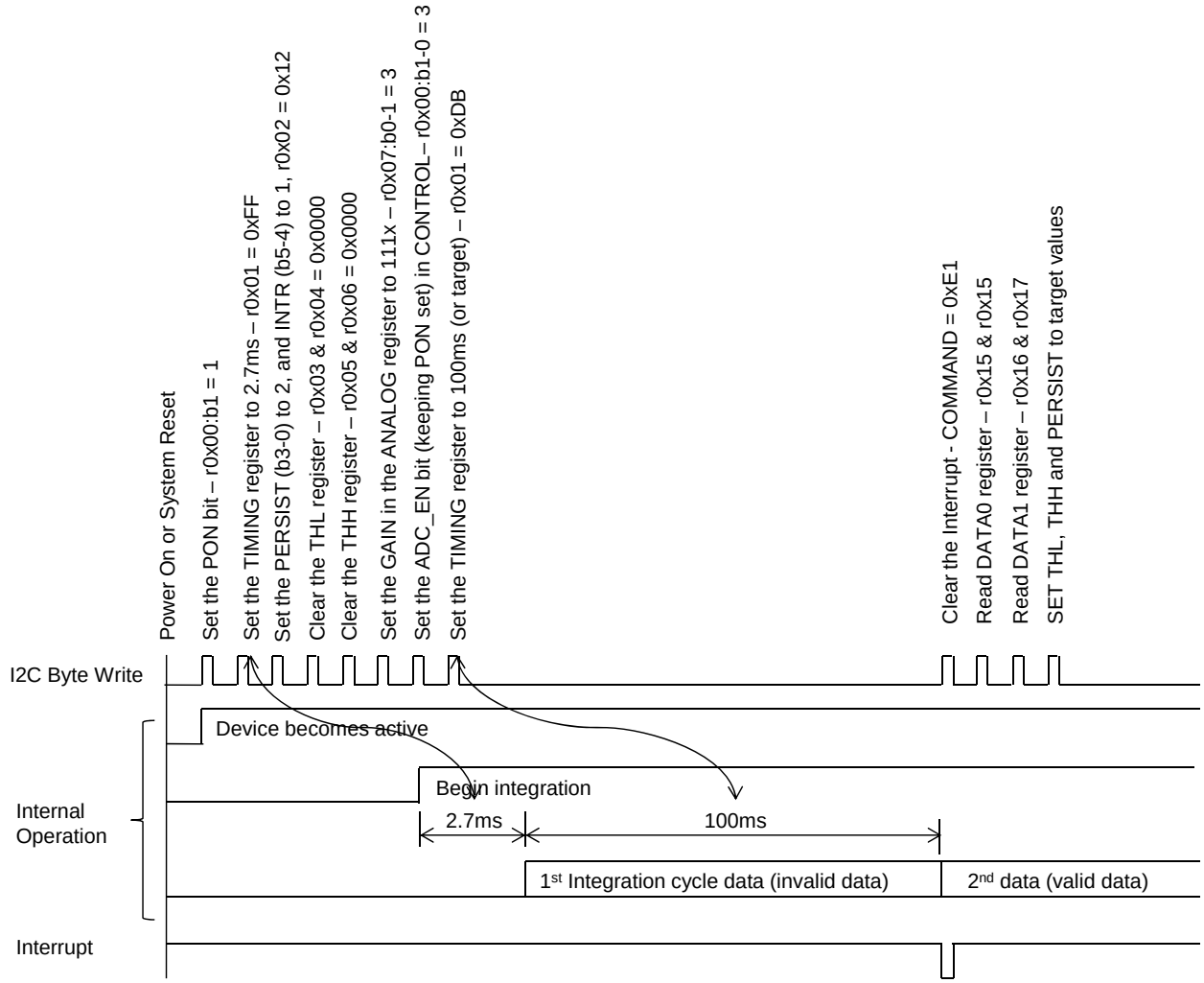

Figure 1: Timing Diagram of Pseudo Code