
TN8000.22

Technical Note

*Using the XE3005 Audio CODEC
with the XE8807A*

Table of Contents

1	INTRODUCTION	3
2	HARDWARE DESCRIPTION	3
2.1	XE3005 overview	3
2.1.1	General description	3
2.1.2	Features	3
2.2	XE8807A overview.....	4
2.2.3	General description	4
2.2.4	Features	5
2.3	I/O connections	5
3	SOFTWARE DESCRIPTION.....	6
3.1	Project structure.....	6
3.2	InitMicro.....	6
3.3	InitCodec.....	6
3.4	InitSamplingEvent	6
3.5	Asm_Send_Receive_Sample	7
3.5.1	Purpose	7
3.5.2	Different possible implementations	8
3.5.3	Routine structure	9
3.5.4	Register use	10
4	CONCLUSION.....	11
5	REFERENCES	11

1 INTRODUCTION

The purpose of this document is to help customers easily build their application containing one of the XE3000 series audio devices driven by one of the XE8000 series microcontrollers.

The hardware and software implementations described in this document have been developed for the XE8807A driving the XE3005. From this example, it is easy to adapt hardware connections and software routines to drive any XE3000 series audio devices with any XE8000 series microcontroller.

This technical note describes and provides the classical low level routines for configuring the XE3005 through the Serial Programming Interface (SPI) and, most important, the assembly optimized routines for communicating efficiently with the XE3000's Serial Audio Interface (SAI).

2 HARDWARE DESCRIPTION

2.1 XE3005 OVERVIEW

2.1.1 General description

The XE3005 is a fully integrated voice / audio CODEC. It contains all functions required for voice / audio microphone drive and amplification; 16 bit sigma-delta ADC, 16-bit PWM DAC and power amplifier. The Serial Audio Interface supports (SFS) Short Frame Synchronization and (LFS) Long Frame Synchronization digital audio formats. For more detailed information please refer to the XE3005 datasheet.

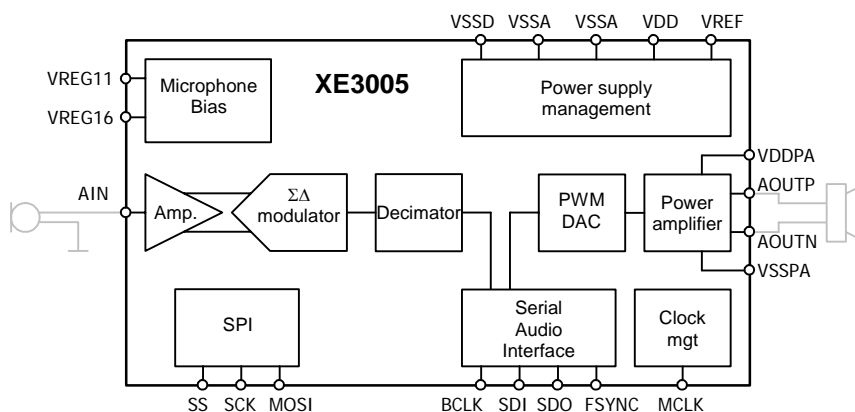


Figure 1 : XE3005 block diagram

2.1.2 Features

- 16 bit sigma-delta ADC, 78 dB SNR
- ADC sampling rate 4 - 48 kHz
- 16 bit PWM DAC, 78 dB SNR
- DAC sampling rate 4 - 48 kHz
- Microphone supply 1.6 V and 1.1 V / 50 uA
- On-chip power amplifier, peak current 100 mA max.
- 1.8 - 3.6V, typ 0.25 mA @ 1.8V, fs=20 kHz
- TSSOP 20, Ultra CSP ®

2.2 XE8807A OVERVIEW

2.2.3 General description

The XE8807A is a transceiver interface IC based on the XE8000 architecture. It exploits the full bandwidth of the transceiver thanks to its BitJockey™ and has extremely low power modes, with 1 Hz hibernation mode. The XE8807A has 4k Instruction of program memory and operates at 5 sustained MIPS (one clock per instruction for all instruction and all addressing modes). The XE8807A is a pin-compatible faster version of the XE8806A. For more detailed information please refer to the XE8807A datasheet.

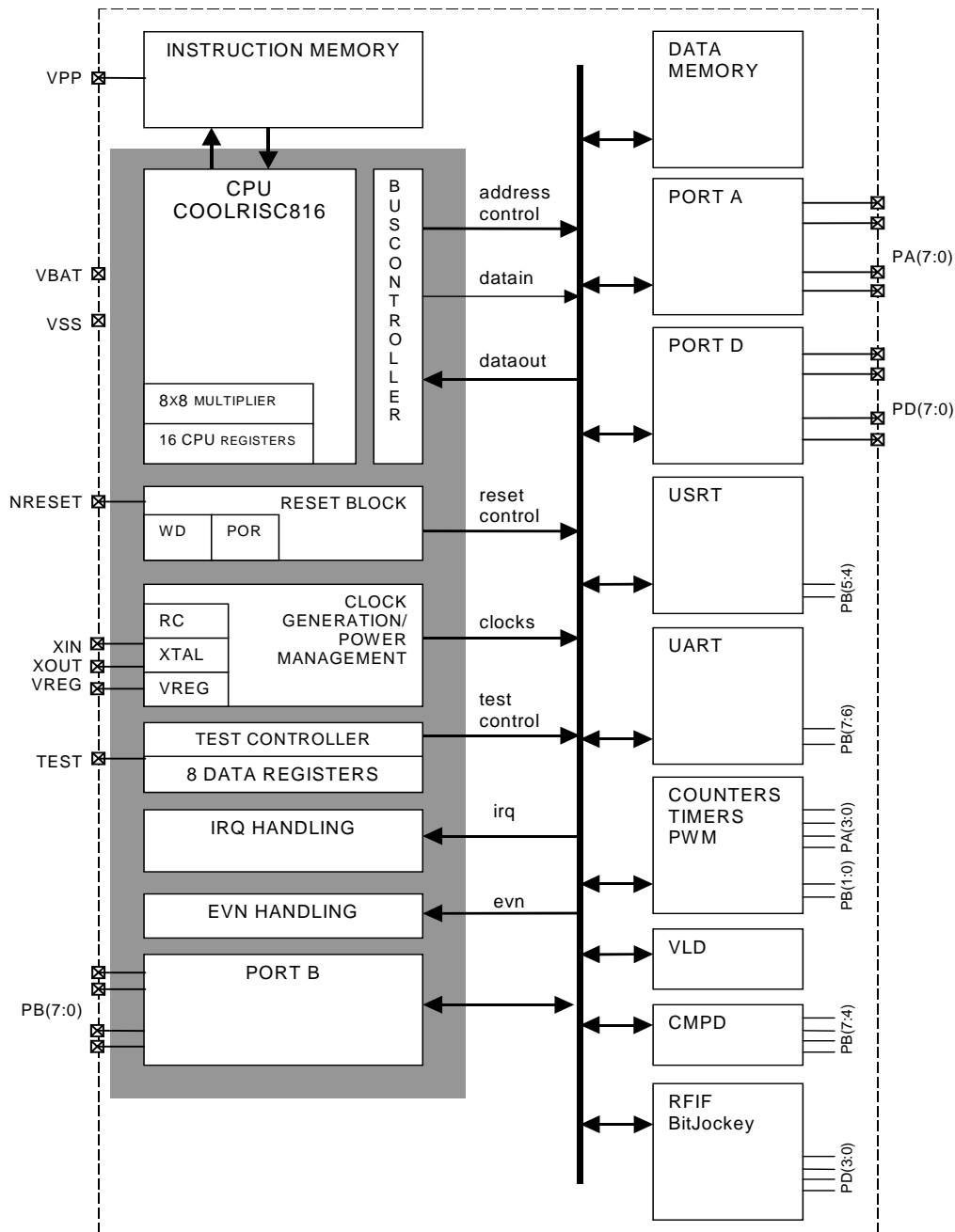


Figure 2 : XE8807A block diagram

2.2.4 Features

- Transceiver interface BitJockey™
 - up to 156 kbit/s
 - built-in FIFO
 - built-in Manchester and Miller encoder/decoder
- Low-voltage low-power controller operation
 - 5 sustained MIPS at 2.4V to 5.5V supply voltage
 - 300 uA at 1 MIPS, 2.4V to 5.5V supply
- 11 kByte (4 kInstruction) MTP, 520 Byte RAM
- RC and crystal oscillators
- 5 reset, 16 interrupt, 8 event sources

2.3 I/O CONNECTIONS

The figure below describes the connections recommended and assumed between the XE8807A and the XE3005 for the whole document and its associated software.

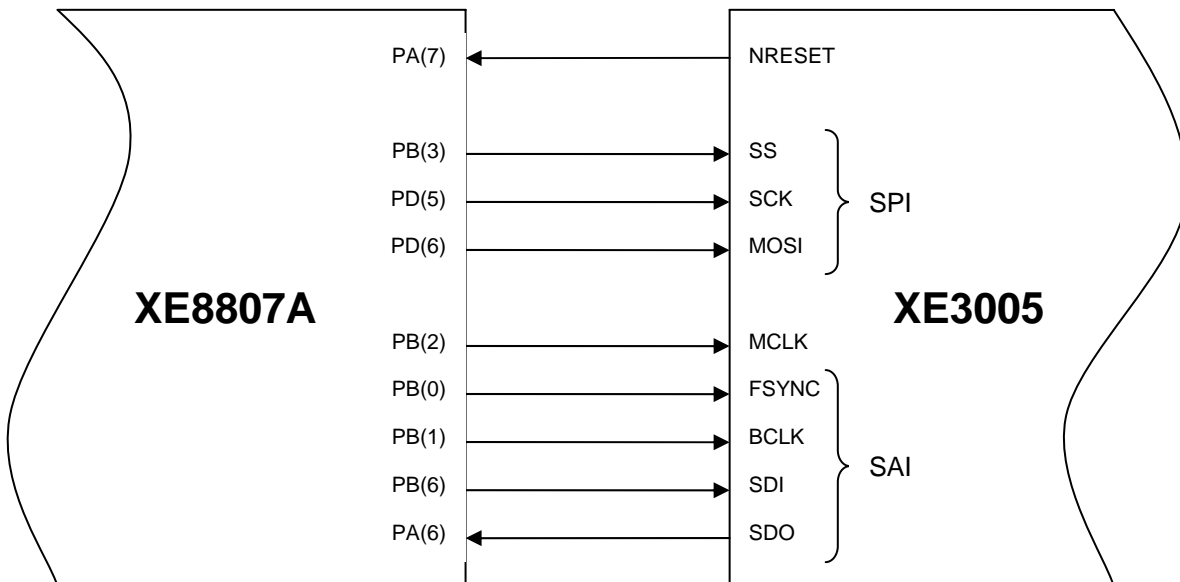


Figure 3 : Connections diagram

The connections diagram minimizes the number of reserved outputs of the microcontroller (ports B and D) by connecting CODEC outputs (NRESET and SDO) on port A (input only). Also note that, in this configuration, the pins corresponding to the BitJockey™ (PD(3,0)) remain available for a potential RF extension.

3 SOFTWARE DESCRIPTION

3.1 PROJECT STRUCTURE

The project is built from the RIDE template. The main function calls several sub functions which are described in the next paragraphs.

3.2 INITMICRO

This function performs various operations to initialize the microcontroller such as CPU frequency setting and ports initialization.

In this example, CPU frequency has been set to 3686400Hz.

3.3 INITCODEC

This function initializes the CODEC through different steps:

1. Initialization of signal values.
2. Generation of the master clock MCLK of the CODEC.
3. Temporization while internal pre-initialization is done (NRESET signal goes high)
4. Programming of the various configuration registers with the appropriate values using the Write_mosi function.

3.4 INITSAMPLINGEVENT

As described in the XE3005/6 datasheet, the whole communication process of the Serial Audio Interface is synchronized on the FSYNC signal, whose frequency has to be equal to the sampling frequency programmed internally in the CODEC.

This function's purpose is to configure the periodic generation of the FSYNC signal through the association of a timer and an event.

The timer uses the concatenation of counters A and B (B = MSB, A =LSB).

The values programmed in register RegCntB and RegCntA should be calculated using the following formula:

$$\mathbf{RegCntBA = (Fcpu / Fsampling) - 1}$$

Ex : Fcpu = 3686400Hz and Fsampling = 7200Hz

- ⇒ RegCntBA = (3686400/7200) – 1
- ⇒ RegCntBA = 511
- ⇒ RegCntBA = 0x01FF
- ⇒ RegCntB = 0x01 and RegCntA = 0xFF

3.5 ASM_SEND_RECEIVE_SAMPLE

3.5.1 Purpose

This routine implements the CODEC's optimized LFS mode communication routine as described in the XE3005 datasheet and in the figure below.

This optimized mode of operation requires the CODEC to be set to Slave mode, and consequently, the purpose of this routine is to properly generate the signals FSYNC, BCLK and SDI and read data on SDO.

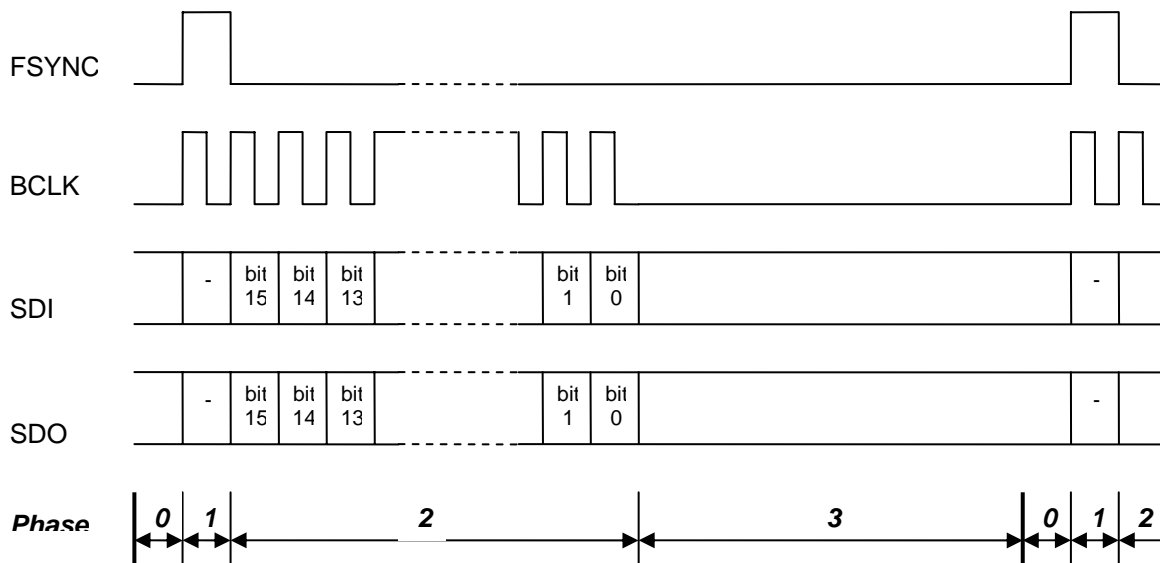


Figure 4 : Optimized LFS mode

Phase 0: This phase starts when the routine is called, it can start anytime between phase 2 and 3 after the other tasks the software has to do. This duration must be positive in order not to miss an event.

Phase 1: This phase corresponds to the handling of the first channel in the LFS mode. In this optimized implementation, we only use the second channel.

Phase 2: This phase corresponds to the handling of the second channel in the LFS mode. This channel is the one on which we exchange data and consequently takes most of the time of the routine. The routine ends up at end of phase 2.

Phase 3: This phase corresponds to the available CPU resources left by the CODEC routine for implementing the rest of the software without overlapping with the next sample communication.

3.5.2 Different possible implementations

Several different implementations are possible for this communication routine. The diagram below illustrates three of them and the link between code size and speed of execution.

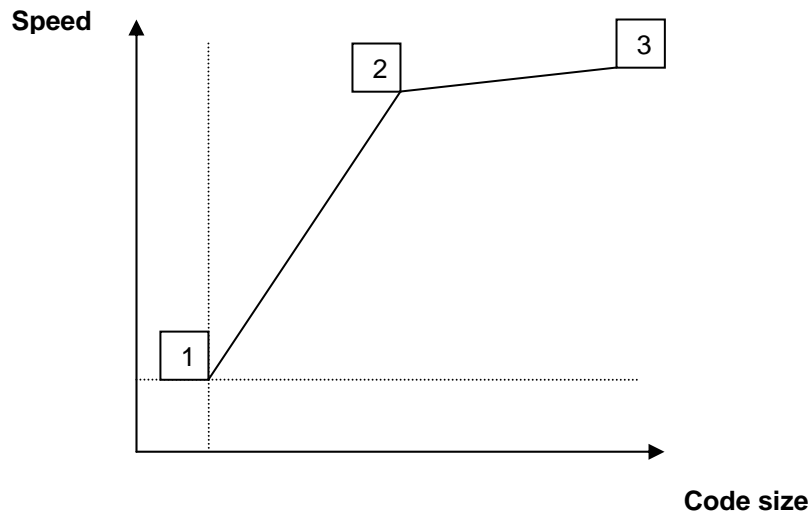


Figure 5 : Implementations diagram

Implementation 1: Code size optimized

This implementation of the subroutine would correspond to the most natural one which would consist of having a loop with a **bit** counter for the 16bits word.

This implementation would result in a very small code size but at the expense of speed of execution due to the counter and its associated tests at every bit.

This implementation could be used to get the smallest code size in a system where the sampling frequency is low or where there is almost no other processing power needed by the rest of the software.

Implementation 2: Code size/Speed compromise

This implementation unrolls an 8 bits loop and uses a **byte** counter to handle MSByte and LSByte.

Because of the 8-bit architecture of the CPU, this implementation provides almost the highest speed reachable while keeping code size reasonable.

This is the implementation provided and described in this document.

Implementation 3: Speed optimized

This implementation completely unrolls the 16 bits loop by duplicating the code of implementation 2 and avoiding the use of any byte or bit counter.

This provides the fastest speed of execution but at the expense of a higher code size.

This implementation may be chosen in applications where speed of execution is the absolute priority and where code size is not so important.

3.5.3 Routine structure

3.5.3.1 Global

At function call, r2 and r3 contain the sample to be sent (r3=MSB; r2=LSB). At the end of the subroutine, r2 and r3 now contain the received sample (r3=MSB; r2=LSB).

Figure below describes the global structure of `Asm_Send_Receive_Sample`.

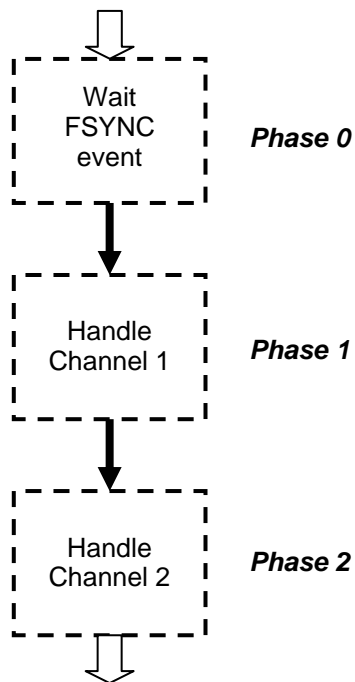


Figure 6 : Routine global structure

In this example project we implement, in `main.c`, an infinite loop on `Asm_Send_Receive_Sample`, creating an audio loopback by sending the previously received sample.

3.5.3.2 Wait FSYNC event

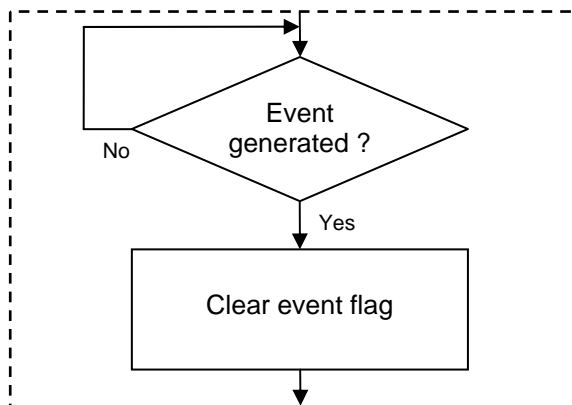


Figure 7 : Synchronization mechanism

3.5.3.3 Handle Channel 1 (no data)

Channel 1, which corresponds to FSYNC = '1', is not used in the LFS optimized mode and is consequently shortened to one BCLK period.

3.5.3.4 Handle Channel 2 (data)

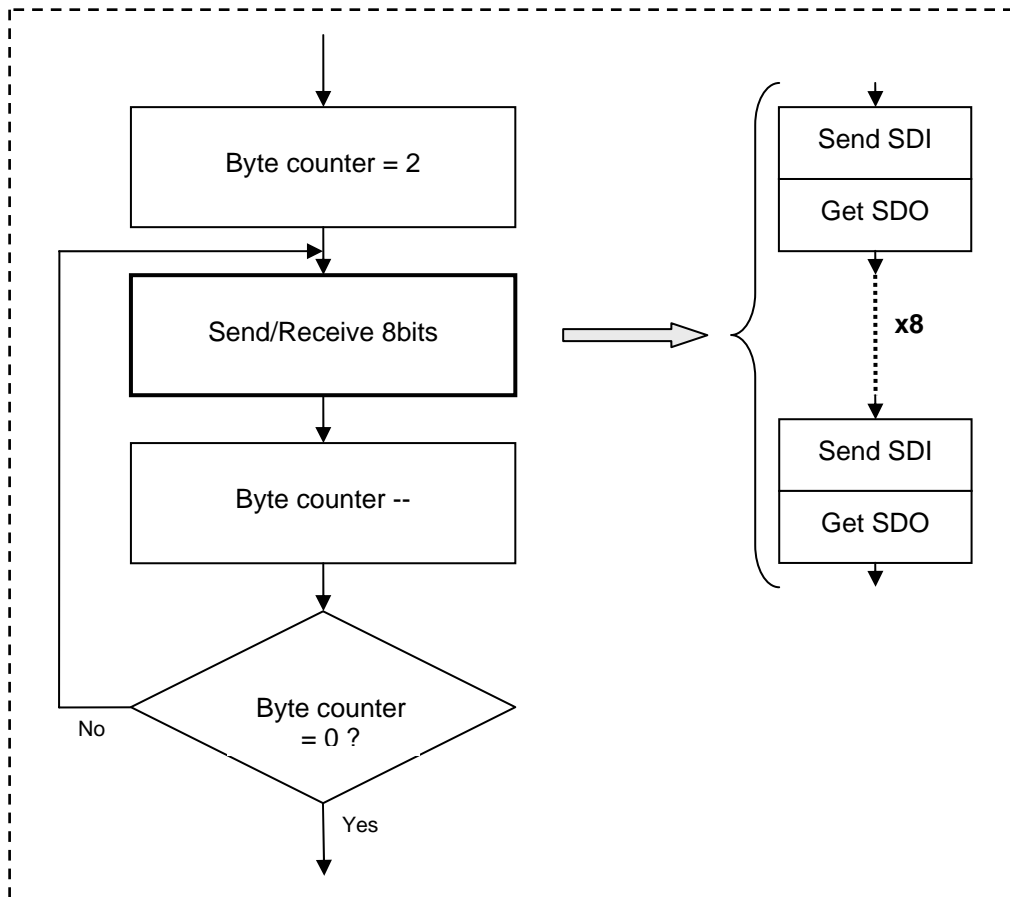


Figure 8 : Data channel handling

3.5.4 Register use

Some registers of the CoolRISC are used during the routine processing. These registers are only used internally and can be overwritten once the routine is completed.

Name	Function in Asm_Send_Receive_Sample
a	SDO reading
r0	RegPBOut generation
r2	Sample's Least Significant Byte
r3	Sample's Most Significant Byte
i0h	Byte counter
i1l	Temporary storage of sample's Least Significant Byte
i1h	Temporary storage of sample's Most Significant Byte

4 CONCLUSION

Asm_send_receive_sample performance:

Code size: 113 instructions

Time of execution (Phases 1 and 2): Max 50us @ Fcpu = 3686400Hz

⇒ 36% of processor time @ Fsampling = 7200Hz

⇒ 1.33 MIPS

5 REFERENCES

- XE3005 / XE3006 Datasheet
<http://www.semtech.com>
- XE8806A / XE8807A Datasheet
<http://www.semtech.com>
- CoolRISC® 816 Databook
<http://www.semtech.com>

© Semtech 2006

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.

Contact Information

Taiwan	Tel: 886-2-2748-3380 Fax: 886-2-2748-3390	Switzerland	Tel: 41-32-729-4000 Fax: 41-32-729-4001
Korea	Tel: 82-2-527-4377 Fax: 82-2-527-4376	United Kingdom	Tel: 44-1794-527-600 Fax: 44-1794-527-601
Shanghai	Tel: 86-21-6391-0830 Fax: 86-21-6391-0831	France	Tel: 33-(0)169-28-22-00 Fax: 33-(0)169-28-12-98
Japan	Tel: 81-3-6408-0950 Fax: 81-3-6408-0951	Germany	Tel: 49-(0)8161-140-123 Fax: 49-(0)8161-140-124

Semtech International AG is a wholly-owned subsidiary of Semtech Corporation, which has its headquarters in the U.S.A