

# Temperature Sensor Board

*User Manual*

---

1.0, Oct 2013



This work is licensed under the Creative Commons Attribution-Share Alike 2.5 India License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/in/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

---

# Table of Contents

1. Temperature Sensor Board .....	1
1. Overview .....	1
2. Board Features .....	1
3. Locating Components .....	1
4. Description .....	1
5. Power Supply .....	2
6. Remote Sensor .....	2
7. Connectors and Headers .....	3
8. Specifications .....	4
2. Board Usage .....	6
1. I <sup>2</sup> C Bus Overview .....	6
2. Accessing Device Registers .....	6
3. Common Usage Scenarios .....	6
A. Legal Information .....	10
1. Copying .....	10
2. Limited Hardware Warranty .....	10

# Chapter 1. Temperature Sensor Board

## 1. Overview

The temperature sensor board is based on NXP's SA56004 temperature sensor, which can be interfaced through the I<sup>2</sup>C bus. Temperatures can be measured with an accuracy of  $\pm 1$  °C.

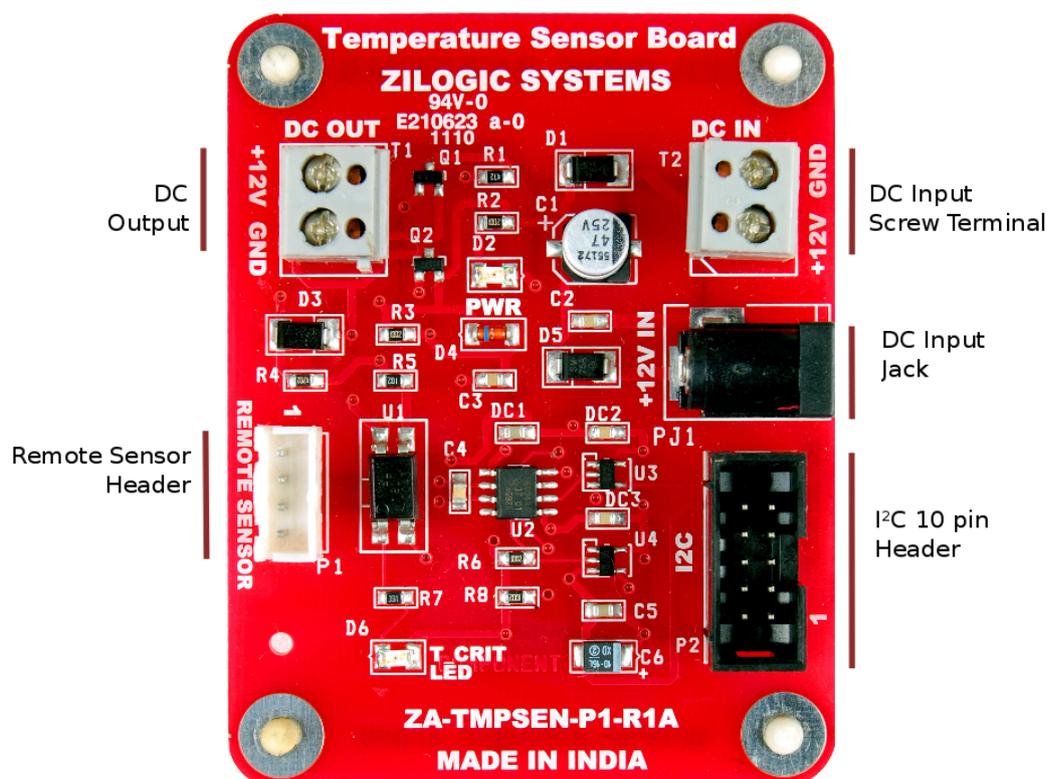
## 2. Board Features

- SA56004, I<sup>2</sup>C temperature sensor
- Remote temperature measurement
- Over temperature alert interrupt
- External device control, without software intervention
- LED indications for testing/debugging
- Can measure temperatures from -40 °C to 125 °C.

## 3. Locating Components

The location of the components on the board is indicated in the following diagrams.

**Figure 1.1. Front View**



## 4. Description

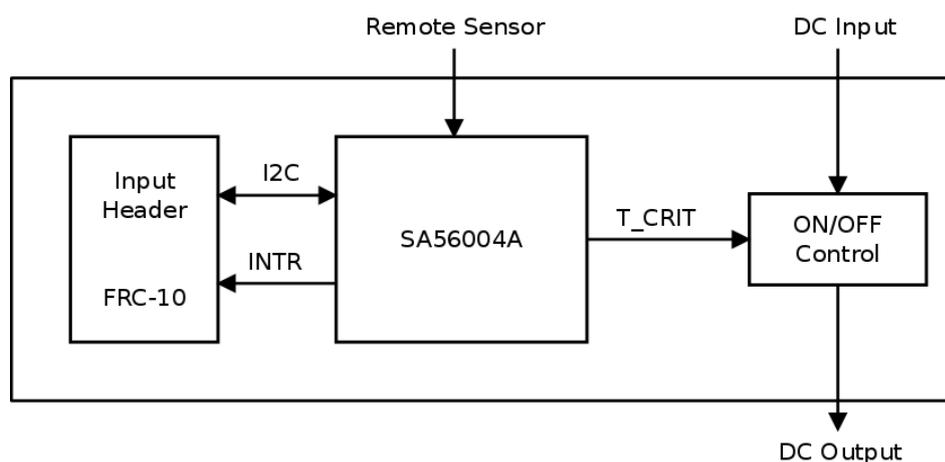
A block diagram representation of the board is shown below. The I<sup>2</sup>C temperature sensor can be accessed through the 10 pin FRC connector from the motherboard. The temperature sensor can

measure the local IC temperature, and a remote temperature. For measuring remote temperatures a transistor connected to the `REMOTE SENSOR` port is used as the sensing element.

An alert interrupt is raised when the temperature exceeds a configurable set point. The signal `T_CRIT` is raised when the temperature exceeds a configurable critical set point. This is useful for automatic temperature regulation applications, where the system is to be shutdown or a fan is to be switched `ON`, without software intervention.

The board provides a isolated 12V/1Amp power switch. The board switches `ON` the 12V/1Amp `DC OUT` on `T_CRIT` assertion and switches `OFF` the `DC OUT` on `T_CRIT` de-assertion. `DC OUT` can be connected to a 12V fan to switch it on and off automatically on thermal critical points.

**Figure 1.2. Block Diagram**



## 5. Power Supply

The temperature sensor board is powered from the motherboard through FRC-10 header.

The `DC OUT` can be used for controlling an external device, like a fan or a relay. If external device control is required, an external 12V/1Amp DC power should be provided through the `DC IN` screw terminal or through the `+12V IN` power jack. This will be driven out through `DC OUT` on `T_CRIT` assertion.

Detailed power supply specifications are available in section Specifications.

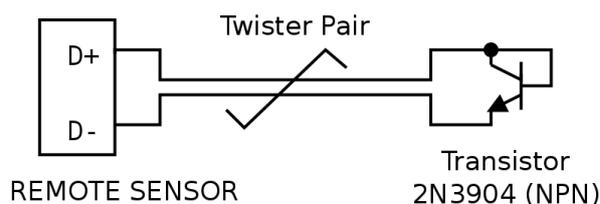
## 6. Remote Sensor

For remote temperature measurement, the Emitter-Base diode of a `BJT`, is used as the sensing element. Note that two-lead diodes cannot be used, since the ideal factor of two-lead diodes is incompatible with the temperature sensor. The recommended transistors are listed below.

- 2N3904 (NPN)
- 2N3906 (PNP)

An example of an `2N3904` NPN transistor connected to the `REMOTE SENSOR` connector is shown in the following diagram.

### Figure 1.3. Connecting Sensing Element



The wires connecting the sensing element can be 8 inches long. For longer wire lengths a shielded twisted pair is recommended. Please see the recommendations under the section "Mounting" of the SA56004X data sheet.

## 7. Connectors and Headers

### 7.1. I2C Connector

The temperature sensor board can be interfaced to a motherboard through the I2C, 10 pin FRC, connector. The signal details are given below.

**Table 1.1. FRC-10 Pin Connector**

Pin #	Signal	Signal Type
1	VCC	Supply from motherboard
2	Not Used	-
3	Not Used	-
4	SCL	Open Collector
5	SDA	Open Collector
6	Not Used	-
7	Not Used	-
8	Not Used	-
9	INTR#	Open Collector
10	GND	-

### 7.2. REMOTE SENSOR Connector

When a remote temperature has to be measured, the remote temperature sensing element is connected to the REMOTE SENSOR header.

**Table 1.2. REMOTE SENSOR Connector**

Pin #	Signal
1	GND
2	D+
3	D-
4	GND

### 7.3. DC OUT and DC IN Connector

This **DC OUT** connector provides a 12V DC from the **DC IN** connector when **T\_CRIT** is asserted. This is typically used for controlling a fan, or triggering a system shutdown, when the temperature exceeds a configured critical set point.

**Table 1.3. DC IN Connector**

Signal	Signal Type
+12V	+12V from external supply
GND	Ground

**Table 1.4. DC OUT Connector**

Signal	Signal Type
+12V	+12V to external device
GND	Ground

## 8. Specifications

**Table 1.5. Specifications**

Parameter	Value	Condition
<b>VCC</b>		
Voltage	5V	
Max.Current	5mA	
<b>External Power Supply</b>		
Voltage	12V	
Max. Current	1A	
Polarity		
<b>Temperature</b>		
Measurement Range	-40 °C - 125 °C	
Local Temp. Error	±2 °C	+60 °C to +100 °C
	±3 °C	-40 °C to +125 °C
Remote Temp. Error	±1 °C	+60 °C to +100 °C
	±3 °C	-40 °C to +125 °C
<b>I<sup>2</sup>C Bus</b>		
Input High Voltage	2.2V - 5.0V	
Input Low Voltage	0.0V - 0.8V	
Max. SCL frequency	400kHz	

**Table 1.6. I<sup>2</sup>C Device Address**

Address	Part Number	Marking Code on Chip
0x48	SA56004-AD	ARS

---

<b>Address</b>	<b>Part Number</b>	<b>Marking Code on Chip</b>
0x49	SA56004-BD	ART
0x4A	SA56004-CD	ARU
0x4B	SA56004-DD	ARV
0x4C	SA56004-ED	ARW
0x4D	SA56004-FD	ARX
0x4E	SA56004-GD	ARY
0x4F	SA56004-HD	ARX

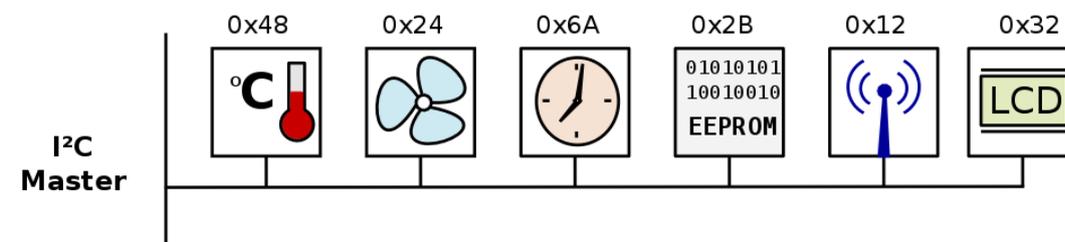
---

# Chapter 2. Board Usage

## 1. I<sup>2</sup>C Bus Overview

The I<sup>2</sup>C is a simple two wire interface for on-board communication. One of the wires is data and the other is clock. The I<sup>2</sup>C is a master slave bus. The master initiates the communication and the slaves respond to it. Each slave on the bus has an address to uniquely identify itself.

**Figure 2.1. I<sup>2</sup>C Bus**



There are typical two operations that the master can perform `i2c_read` and `i2c_write`. During the `i2c_read` operation the master specifies the slave address and reads one or more bytes. During the `i2c_write` operation the master specifies the slave address and the bytes to be written.

## 2. Accessing Device Registers

The devices usually have a set of registers. Each register is given an address. When a register is to be read or written, the address is first written to the device and then data is read or written to the register.

The register address specified is stored within the device. For successive reads to the same register the address need not be specified again.

## 3. Common Usage Scenarios

This section shows the pseudo-code for various common usage scenarios. The I<sup>2</sup>C operations are represented by the following two functions.

- `i2c_read()`
- `i2c_write()`

```
data = i2c_read(dev_addr);
```

The `dev_addr` is the 7-bit I<sup>2</sup>C address of the device to read from. The byte read from the device is returned by `i2c_read()`.

```
i2c_write(dev_addr, data);
```

The `dev_addr` is the 7-bit I<sup>2</sup>C address of the device to write to. The `data` is the byte to be written.

**Scenario I: Measuring Local Temperature.** To measure the local temperature the following pseudo code can be used.

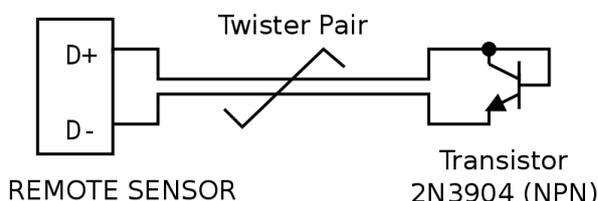
```
lthb_reg = 0x0;
```

```
i2c_write(dev_addr, lthb_reg); ❶
temp = i2c_read(dev_addr); ❷
```

- ❶ Write the register address of the local temperature register (**LTHB**), 0x0.
- ❷ Read the contents of **LTHB**. The read byte is the local temperature in degree Celsius.

**Scenario II: Measuring Remote Temperature.** The temperature sensing element, is first connected to the **SENSOR PORT** as shown in the following diagram.

**Figure 2.2. Connecting Sensing Element**



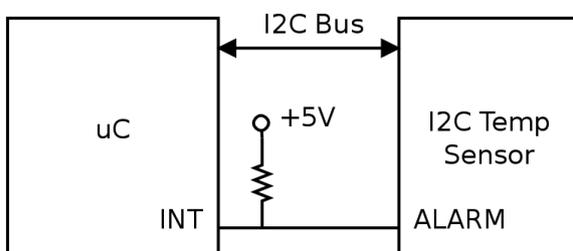
The following pseudo code is used to measure the remote temperature.

```
rthb_reg = 0x1;
i2c_write(dev_addr, rthb_reg); ❶
temp = i2c_read(dev_addr); ❷
```

- ❶ Write the register address of the remote temperature register (**RTHB**), 0x1.
- ❷ Read the contents of **RTHB**. The read byte is the remote temperature in degree Celsius.

**Scenario III: Interrupt Notification.** In certain applications, an action has to be taken when the temperature exceeds a specific temperature range. This can be done by periodically reading the temperature register. But a much more efficient method is possible. The temperature sensor can interrupt the CPU, when the temperature exceeds a specified range. The CPU then performs the required action, and acknowledges the interrupt.

**Figure 2.3. Alert Interrupt**



The temperature sensor has an **ALERT** signal that can be used as an interrupt signal. The set points for local and remote temperatures can be configured individually. An example circuit configuration is shown in Figure 2.3, "Alert Interrupt". Note that since the **ALERT** is an open collector output a pull-up resistor (typically 10k) is required. The pseudo code for enabling interrupt notification when the local temperature exceeds the range 10 °C - 60 °C is given below.

```
lhs_reg = 0x0B;
i2c_write(dev_addr, lhs_reg); ❶
```

```
temp = 60;
i2c_write(dev_addr, temp); ❷

lls_reg = 0x0C;
i2c_write(dev_addr, lls_reg); ❸

temp = 10;
i2c_write(dev_addr, temp); ❹
```

- ❶ Write register address of the Local High Setpoint register (**LHS**), 0x0B.
- ❷ Write local high setpoint temperature 60 °C to **LHS**.
- ❸ Write register address of the Local Low Setpoint register (**LLS**), 0x0C.
- ❹ Write local low setpoint temperature 10 °C to **LLS**.

The pseudo code for the interrupt handler is given below.

```
lthb_reg = 0x0;
i2c_write(dev_addr, lthb_reg);

temp = i2c_read(dev_addr); ❶

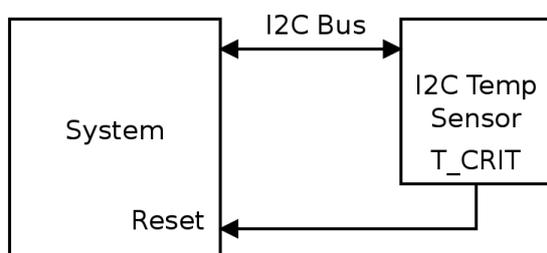
sr_reg = 0x2;
i2c_write(dev_addr, sr, reg);

status = i2c_read(dev_addr); ❷
```

- ❶ Read the local temperature.
- ❷ Read the status register to acknowledge the interrupt.

**Scenario IV: Thermal Watchdog.** In certain applications, the temperature sensor is required to autonomously control an external device, when the temperature exceeds a setpoint. This must be done without the intervention of the software. One such application is the thermal watchdog, where the temperature sensor shuts down the system being monitored or switches a cooling fan when the temperature overshoots the setpoint.

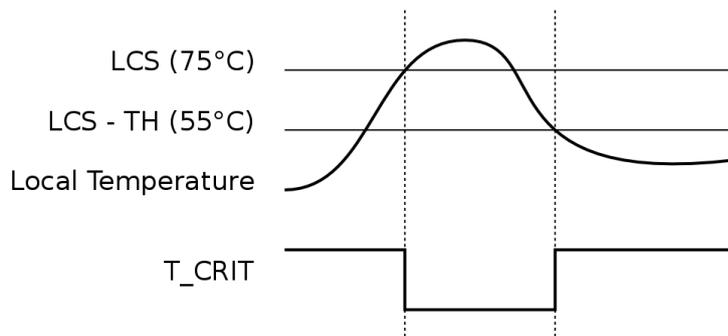
#### Figure 2.4. Thermal Watchdog



The temperature sensor can act like a thermal watchdog using the **T\_CRIT** signal. The **T\_CRIT** becomes low when the temperature exceeds the **T\_CRIT** setpoint. The **T\_CRIT** is held low till the temperature drops by a configurable hysteresis amount.

The **T\_CRIT** signal can in-turn control a cooling fan. The **T\_CRIT** setpoint can be configured individually for local and remote temperatures. An example circuit configuration is shown in Figure 2.4, "Thermal Watchdog". Hysteresis on the **T\_CRIT** signal is shown in the figure below.

## Figure 2.5. Hysteresis



The default `T_CRIT` setpoint is 85 °C. The default `T_CRIT` hysteresis is 10 °C. The following pseudo-code sets the `T_CRIT` setpoint to 75 °C, and the hysteresis to 20 °C.

```
lcs_reg = 0x20;  
i2c_write(dev_addr, lcs_reg); ❶  
  
temp = 75;  
i2c_write(dev_addr, temp); ❷  
  
th_reg = 0x21;  
i2c_write(dev_addr, th_reg); ❸  
  
hyst = 20;  
i2c_write(dev_addr, hyst); ❹
```

- ❶ Write the register address of the local `T_CRIT` register (`LCS`), 0x20.
- ❷ Write the setpoint to `LCS`. The local temperature setpoint in degree Celsius, is written to the register.
- ❸ Write the register address of the `T_CRIT` hysteresis register (`TH`), 0x21.
- ❹ Write the hysteresis amount to `TH`. The hysteresis amount in degree Celsius, is written to the register.

# Appendix A. Legal Information

## 1. Copying

This work is licensed under the Creative Commons Attribution-Share Alike 2.5 India License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/in/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

## 2. Limited Hardware Warranty

The warranties provided by Zilogic Systems in this Limited Hardware Warranty apply only to Hardware Products you purchase for your use, and not for resale. The term "Hardware Product" means a computing device with a specific function and limited configuration ability.

### 2.1. LIMITED HARDWARE WARRANTY

Zilogic Systems warrants that the hardware components of its Hardware Product shall be free from material defects in design, materials, and workmanship and will function, under normal use and circumstances, in accordance with the documentation provided, for a period of one (1) year from the date of purchase of the Hardware Product.

Your sole and exclusive remedy, and Zilogic Systems' sole and exclusive liability for defective hardware components, shall be that Zilogic Systems, subject to the terms and conditions of this Section, and solely upon confirmation of a defect or failure of a hardware component to perform as warranted, shall at its sole option, either repair or replace the nonconforming hardware component. All replacement parts furnished to you under this warranty shall be refurbished and equivalent to new, and shall be warranted as new for the remainder of the original warranty period. All defective parts, which have been replaced, shall become the property of Zilogic Systems. All defective parts that have been repaired shall remain your property.

### 2.2. EXCLUSIONS

The foregoing warranties and remedies shall be void as to any Hardware Products damaged or rendered unserviceable by one or more of the following: (1) improper or inadequate maintenance by anyone other than Zilogic Systems or Zilogic Systems' authorized engineers, (2) interfacing supplied by anyone other than Zilogic Systems, (3) modifications, alterations or additions to the Hardware Products by personnel not certified by Zilogic Systems or Zilogic Systems' authorized engineers to perform such acts, or other unauthorized repair, installation or other causes beyond Zilogic Systems' control, (4) unreasonable refusal to agree with engineering change notice programs, (5) negligence by any person other than Zilogic Systems or Zilogic Systems' authorized engineers, (6) misuse, abuse, accident, electrical irregularity, theft, vandalism, fire, water or other peril, (7) damage caused by containment and/or operation outside the environmental specifications for the Hardware Products, (8) alteration or connection of the Hardware Products to other systems, equipment or devices (other than those specifically approved by Zilogic Systems) not in accordance to the board and on-board device specifications (9) any use that is inconsistent with the user manual supplied with the Hardware Product. The warranty period is not extended if Zilogic Systems repairs or replaces a warranted product or any parts. Zilogic Systems may change the availability of limited hardware warranties, at its discretion, but any changes will not be retroactive.

### 2.3. HARDWARE RETURN PROCEDURES

If a Hardware Product or one of its component parts does not function as warranted during the warranty period, and such nonconformance can be verified by Zilogic Systems, Zilogic Systems, at

its election, will provide either return and replacement service or replacement with a refurbished part/unit for the Hardware Product under the type of warranty service Zilogic Systems designates for that Hardware Product. A defective Hardware Product or one of its component parts may only be returned to Zilogic Systems upon Zilogic Systems' prior written approval. Any such approval shall reference an RMA number issued by an authorized Zilogic Systems service representative. If you do not register the Hardware Product with Zilogic Systems, you may be required to present proof of purchase as evidence of your entitlement to warranty service. The Hardware Product's serial number will be required for all RMA cases.

Transportation costs, if any, incurred in connection with the return of a defective item to Zilogic Systems shall be borne by You. Any transportation costs incurred in connection with the redelivery of a repaired or replacement item to You by Zilogic Systems shall be borne by Zilogic Systems; provided, however, that if Zilogic Systems determines, in its sole discretion, that the allegedly defective item is not covered by the terms and conditions of the warranty or that a warranty claim is made after the warranty period, the cost of the repair by Zilogic Systems, including all shipping expenses, shall be reimbursed by You.

## **2.4. HARDWARE REPLACEMENT PROCEDURES**

Zilogic Systems will attempt to diagnose and resolve your problem over the phone or e-mail. Upon determination of the hardware issue is related to a malfunction of one of the Hardware Product components, an RMA process will be initiated by Zilogic Systems.

For Warranty Replacement service, it is required that you deliver the faulty unit to a location Zilogic Systems designates, and provide courier name and tracking number to Zilogic Systems. After the Faulty unit is returned to Zilogic Systems, Zilogic Systems will use commercially reasonable efforts to ship the replacement hardware within fourteen (14) business days. Actual delivery times may vary depending on availability of the spares and customer's location.

## **2.5. ADDITIONAL RESPONSIBILITIES**

You agree:

- To provide Zilogic Systems or its partner with sufficient and safe access to your facilities to permit Zilogic Systems to fulfill its obligations.
- To ship back the faulty Hardware Product (or replaceable unit) suitably packaged, quoting the RMA number, to the Zilogic Systems designated location.
- You shall ship the faulty Hardware Product once Zilogic Systems approves the RMA and provide the courier name and tracking number.
- To securely erase from any Hardware Product you return to Zilogic Systems for any reason all programs and data not provided by Zilogic Systems with the Hardware Product. You acknowledge that in order to perform its responsibilities under this Limited Hardware Warranty, Zilogic Systems may ship all or part of the Hardware Product or its software to third party locations around the world, and you authorize Zilogic Systems to do so.

## **2.6. LIMITATION OF LIABILITY**

Zilogic Systems' development kits are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, not in applications where failure or malfunction of a Zilogic Systems product can reasonably be expected to result in personal injury, death or severe property or environmental damage.

NOTWITHSTANDING ANYTHING ELSE IN THIS AGREEMENT OR OTHERWISE, NEITHER ZILOGIC SYSTEMS NOR ITS SUPPLIERS WILL BE LIABLE WITH RESPECT TO ANY SUBJECT MATTER OF THIS AGREEMENT UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY, OR OTHER LEGAL

OR EQUITABLE THEORY, REGARDLESS OF WHETHER ZILOGIC SYSTEMS OR ITS SUPPLIERS WERE ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, FOR: (i) ANY PUNITIVE, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR LOST DATA OR LOST PROFITS; OR (ii) FOR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES; OR (iii) FOR ANY CLAIMS BASED ON ANY ERROR, DEFECT OR NONCONFORMITY IN THE PRODUCTS OR SERVICE, FOR ANY AMOUNT IN EXCESS OF THE PRICE PAID TO ZILOGIC SYSTEMS FOR SUCH DEFECTIVE PRODUCT(S) OR SERVICE; OR (IV) FOR ALL OTHER CLAIMS NOT RELATED TO AN ERROR, DEFECT OR NONCONFORMITY IN THE PRODUCTS, ANY AMOUNTS IN EXCESS IN THE AGGREGATE OF THE AMOUNT PAID TO ZILOGIC SYSTEMS HEREUNDER DURING THE THREE (3) MONTHS PRECEDING THE DATE THE CAUSE OF ACTION AROSE.

WARRANTY DISCLAIMER. EXCEPT AS STATED HEREIN, ZILOGIC SYSTEMS MAKES NO WARRANTIES WITH RESPECT TO any PRODUCT, license or SERVICE AND DISCLAIMS ALL Statutory or IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, or arising from a course of dealing or usage of trade and any WARRANTIES OF NONINFRINGEMENT. ZILOGIC SYSTEMS DOES NOT WARRANT THAT THE ZILOGIC SYSTEMS PRODUCT(S) WILL MEET any REQUIREMENTS or THAT THE OPERATION OF ZILOGIC SYSTEMS PRODUCTS WILL BE UNINTERRUPTED OR ERROR FREE.