ZS1310, USB CAN Adaptor

User Manual

1.2, 7 Feb 2025

ZILOGIC

Table of Contents

Chapter 1. Revision History

Revision	Date	Comment
1.0	Jul 2023	Initial Revision.
1.1	Dec 2024	Add enclosure assembly and removal procedure.
1.2	Feb 2025	Product code updated.

Chapter 2. Introduction

ZS1310, USB CAN Adaptor is a signal and power-isolated CAN transceiver that allows a system with a USB to connect with a Controller area network (CAN). It is based on the open-source CANtact hardware and candleLight firmware.

2.1. Applications

- Troubleshoot CAN based systems by sniffing messages on a CAN bus.
- Automate testing of CAN based systems.
- Exploring and rapid prototyping of CAN devices.

2.2. Features

- Host Interface USB 2.0
- High-speed CAN connection (ISO 11898-2)
- Bit rates from 5 kbit/s up to 1 Mbit/s
- CAN Transceiver Analog Devices ADM3053
- CAN termination can be activated through a jumper
- Voltage supply via USB
- Temperature range from 0°C to 70°C

2.3. System Requirements

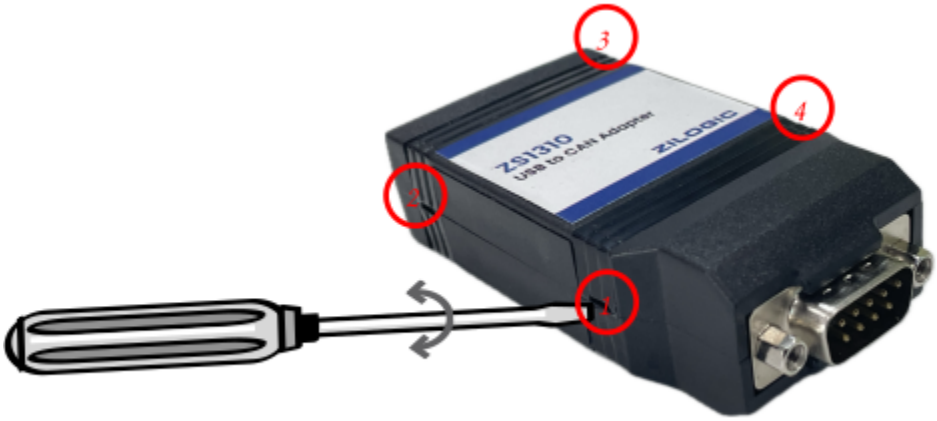
Computer with

- Linux Kernel >= 4.5
- Ubuntu 20.04 and above
- USB 2.0 port

Chapter 3. HW Settings

3.1. Enclosure Assembly and Removal

- The default jumper pin settings might need to be modified as per the usage requirements of the USB CAN Adaptor device. The enclosure needs to be removed to modify the default jumper settings as mentioned in Section 3.2, “120 ohm Termination” and Section 3.3, “CAN / OBD-II Pinout”.
- The diagram below illustrates the procedure for removing the enclosure.



- Insert a 3mm flat head screwdriver into any of the 4 holes present at the sides of the enclosure and rotate it clockwise or anti-clockwise to open the enclosure.
- After changing the pin settings, fit the enclosure by aligning the removed top cover over the bottom and pressing it. A click sound is heard on assembling the enclosure correctly.

3.2. 120 ohm Termination

There are 2 possible termination settings:

1. Termination Enable [Default]
2. Termination Disable

For each of the termination settings, the jumper positions are shown in the following diagrams.

Figure 3.1. Termination Enable

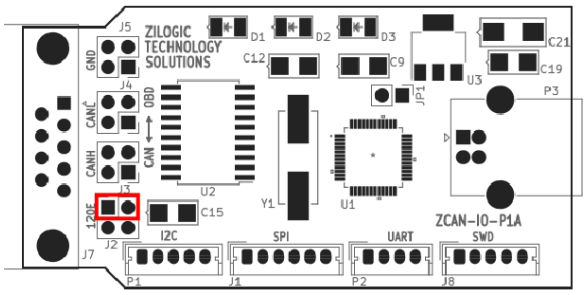
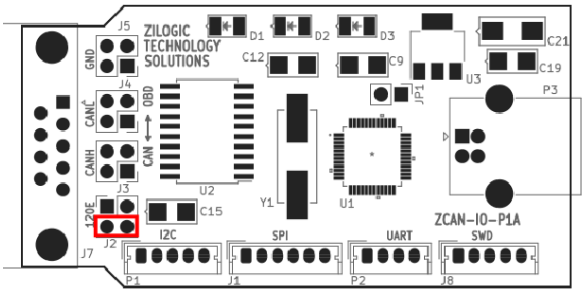


Figure 3.2. Termination Disable



3.3. CAN / OBD-II Pinout

There are 2 possible pinout settings:

1. CAN Pinout [Default]
2. OBD-II Pinout

For each of the pinout settings, the jumper positions are shown in the following diagrams.

Figure 3.3. CAN Selection

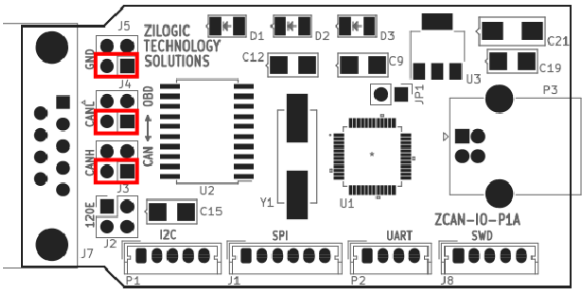
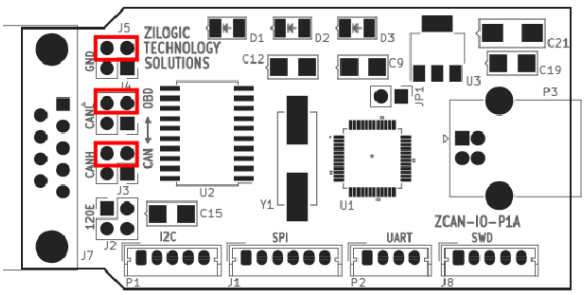


Figure 3.4. OBD-II Selection



Chapter 4. Usage

4.1. Connector

The pinout details of the connector in CAN mode and OBD-II mode is shown in the following tables.

Figure 4.1. D-Sub Male Pin Numbering

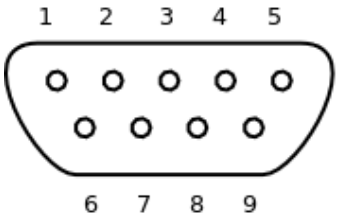


Table 4.1. CAN Pinout

Pin	Description
1	-
2	CANL
3	GND
4	-
5	-
6	-
7	CANH
8	-
9	-

Table 4.2. OBD-II Pinout

Pin	Description
1	GND
2	-
3	CANH
4	-
5	CANL
6	-
7	-
8	-
9	-

4.2. LEDs

Component	Color	Description
D1	Red	TX Indication
D2	Green	RX Indication

Component	Color	Description
D3	Red	Power Indication

4.3. Linux Commands

4.3.1. Dependencies

Install CAN utils package in Ubuntu using the following command.

```
apt-get install can-utils
```

4.3.2. Setup

The CAN device is visible as a network interface. After connecting the USB CAN Adaptor device to the system, check the name assigned to the device using the following command.

```
$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
9: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT group d
    link/can
10: can1: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT group
    link/can
```

The CAN device can be then configured and enabled using the following command.

```
$ ip link set can0 up type can bitrate 500000
```

4.3.3. Transmit and Receive

The `cansend` command transmits a CAN message. In the following example the `cansend` command transmits using the `can0` interface a message with identifier `0x1B` and 8 bytes of data.

```
cansend can0 01b#1122334455667788
```

The received data can be viewed through the `candump` command. The following command displays the data received on the interface `can0`.

```
candump can0
```

4.4. GUI Application - CANdevStudio

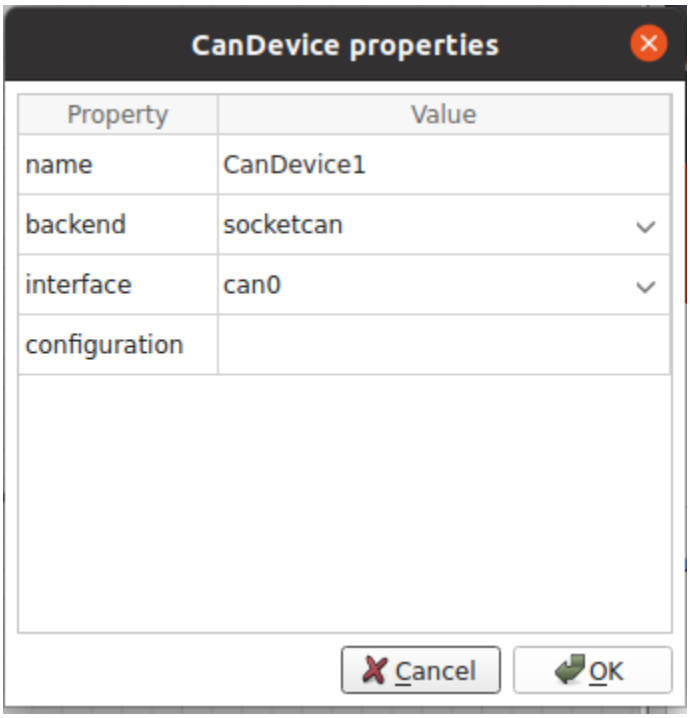
CANdevStudio is designed to support the development and testing of CAN (Controller Area Network) based systems. CANdevStudio is cost-effective replacement for CAN simulation software.

- Download CANdevstudio from https://github.com/GENIVI/CANdevStudio and extract the file.
- Run script CANdevStudio.sh to start application.
- Enable CAN interface as indicated in Section 4.3.2, “Setup”.

4.4.1. Sending using CANdevStudio

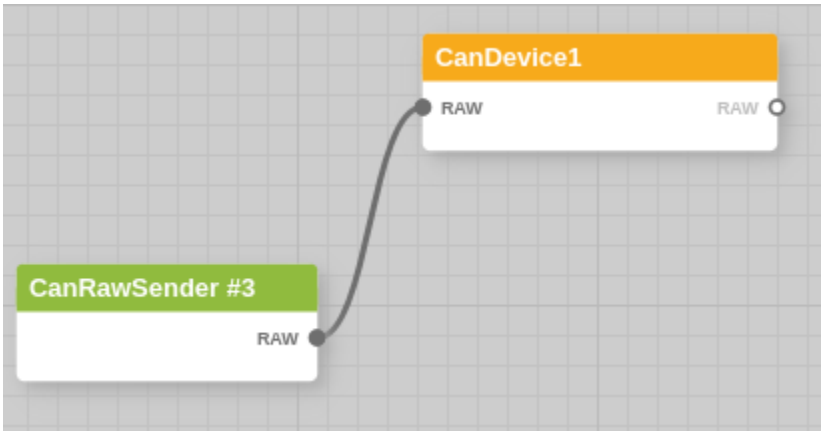
- Go to project and select New Project.
- Drag candevice from device layer and canrawsender from rawlayer.
- Configure the CanDevice as shown below.

Figure 4.2. CandevStudio Candevice Configuration



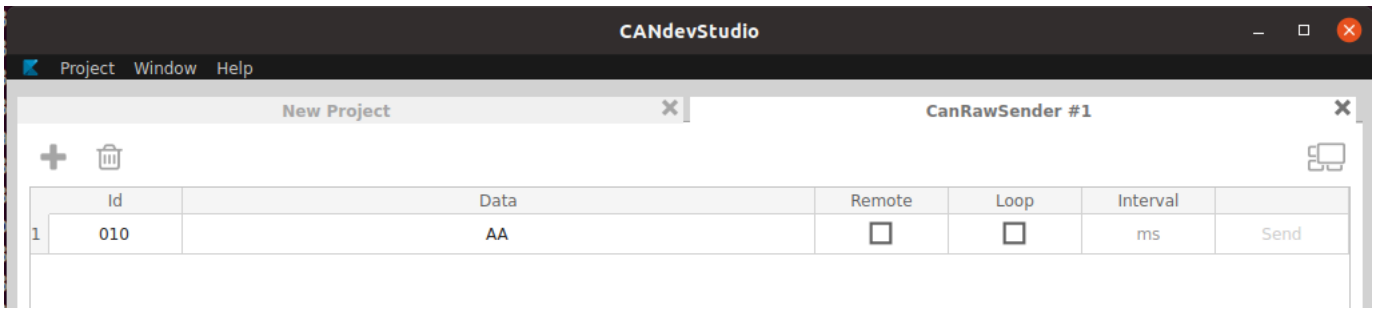
- Connect the raw points, as shown in the following diagram.

Figure 4.3. CandevStudio Send



- Open canrawsender and click on the `+` sign to add the CAN message to send. Specify the Id, Data for the CAN message to be sent, as shown in the following screenshot.

Figure 4.4. CandevStudio Send configuration

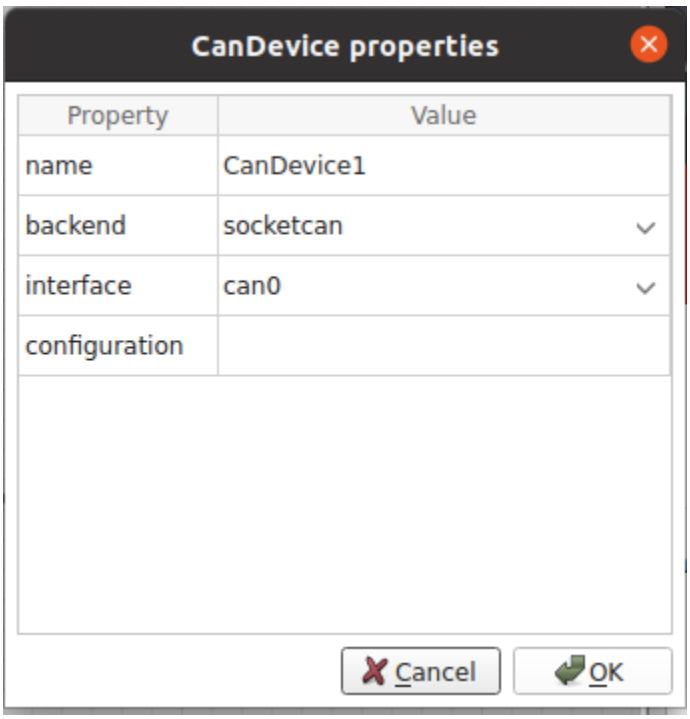


- Click play button to run the project. The application will send the CAN message, through the CAN interface. Enable loop and specify the interval to send the message periodically.

4.4.2. Receiving using CANdevStudio

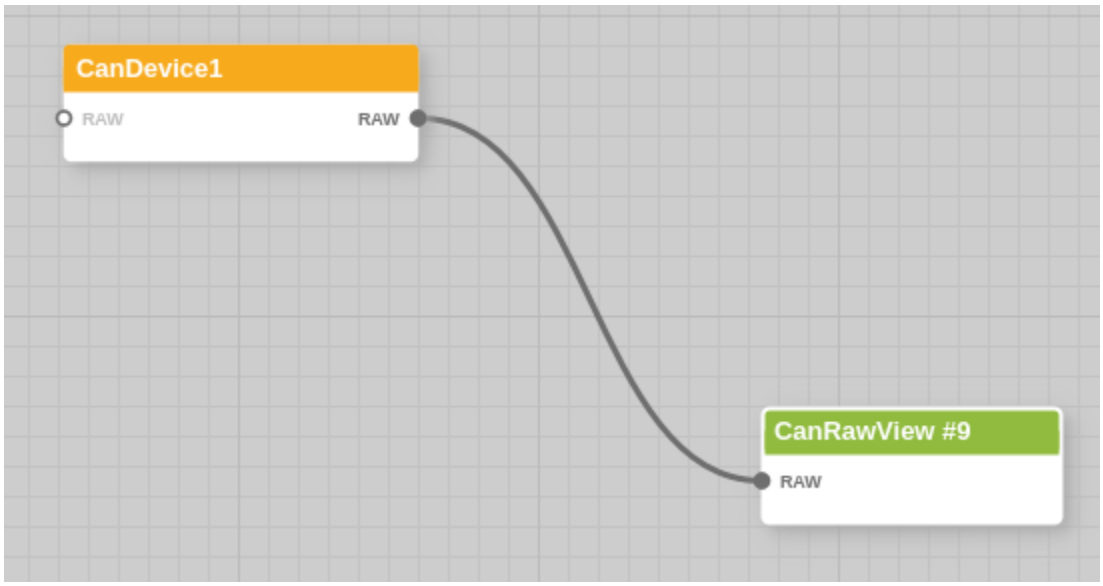
- Go to project and select New Project.
- Drag candevice from device layer and canrawview from rawlayer.
- Configure the CanDevice as shown below.

Figure 4.5. CandevStudio Candevice Configuration



- Connect the raw points, as shown in the following diagram.

Figure 4.6. CandevStudio Receive



- Click play button to run the project. Open the "canrawview", to view the received CAN messages.

4.5. Python API

USB CAN Adaptor can be accessed using the python-can module. The module can be installed using the following command.

```
pip install python-can
```

The module can be import using the following import command.

```
import can
```

To access the CAN device, create a CAN bus object as shown below. Specify the name of the CAN device in `channel` argument.

```
bus = can.interface.Bus(bustype="socketcan",
                        channel="can0",
                        bitrate=5000,
                        receive_own_messages=True)
```

Messages can be created and sent using the `bus.send()` method, as shown below.

```
msg_data = [0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88]
msg = can.Message(arbitration_id=0x1b, data=msg_data, is_extended_id=False)
bus.send(msg)
```

Messages can be received from the can bus using `bus.recv()` method, as shown below.

```
timeout_sec = 5
bus.recv(timeout_sec):
print(f"Received msg: ID={msg.arbitration_id}, Data={msg.data}")
```

For more information on the python-can package, please see https://python-can.readthedocs.io/en/stable/

Appendix A. 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.

A.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.

A.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.

A.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.

A.3.1. 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.

A.4. 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.

A.5. 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 resonably 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.