

VLSI System Design (VSD)

vsdsquadron

The vsdsquadron is an educational kit with general-purpose interfaces that enables you to evaluate features of RISC-V ISA

Contents

1	Getting Started	4
1.1	Kit Contents	4
1.2	Block Diagram	4
1.3	Web Resources	5
1.4	Board Overview	5
1.4.1	Form Factor	6
1.4.2	Caravel RISC-V SoC IO Bank Assignment for Individual Interfaces	6
1.4.3	The following table lists the important components of the vsdsquadron RISC-V SoC Kit	7
1.5	Handling the Board	7
1.6	Operating Temperature	8
1.7	Powering Up the Board	8
2	Installation and Settings	9
2.1	Software Settings	9
2.1.1	Toolchain setup	9
2.1.2	Programming flow to run gpio_test	9
2.2	Hardware Settings	13
2.3	Power Sources, GPIOs and FTDI	14
3	Board Component Placement	15
3.1	vsdsquadron top view	15
3.2	vsdsquadron bottom view	15
4	Appendix: Transmitting data over UART	16
4.1	Installing and running serial terminal to see messages	16
4.2	Flashing UART code and viewing data	16
5	Revision History	17
	The VSD Website	18
	Product Change Notification Service	19
	Customer Support	20
	VSD Devices Intellectual Property Protection	21
	Legal Notice	22

List of Tables

1	Kit Contents	4
2	Caravel Bank Assignment	6
3	vsdsquadron RISC-V SoC Kit Components	7
4	Pin definitions and settings	13
5	LEDs for Power supply, GPIOs and FTDI	14
6	Revision History	17

List of Figures

1	vsdsquadron RISC-V SoC Kit Block Diagram	5
2	vsdsquadron RISC-V SoC Kit Board image	6
3	Micro-B end of micro USB cable connected to board	8
4	Data Flow - Compiling	10
5	Type-A end of micro USB cable connected to laptop USB port	10
6	Flash programming message	11
7	Data Flow - Flashing	11
8	Data Flow - Executing	13
9	Silkscreen top view	15
10	Silkscreen bottom view	15

1 Getting Started

The vsdsquadron RISC-V SoC Kit supports the following interfaces:

- Caravel chip with VexRISC V processor
- 10 Mhz External Oscillator
- 38 General purpose Input Output pins
- UART
- SPI
- 4Mb External Flash
- USB Programming support over FTDI

The vsdsquadron RISC-V SoC device available on Kit is programmed using the onboard flash programmer through Caravel's housekeeping SPI interface. Connect the vsdsquadron board using a USB micro B connector to program the Caravel chip.

1.1 Kit Contents

The following table lists the contents of the vsdsquadron RISC-V SoC Kit.

Item	Quantity
vsdsquadron SoC Kit featuring the RISC-V Caravel device with RV32I instruction set architecture	1
Micro USB cable	1
Quickstart card	1
Memory Stick	1

Table 1: Kit Contents

1.2 Block Diagram

The following block diagram shows the key components of the vsdsquadron RISC-V SoC Kit.

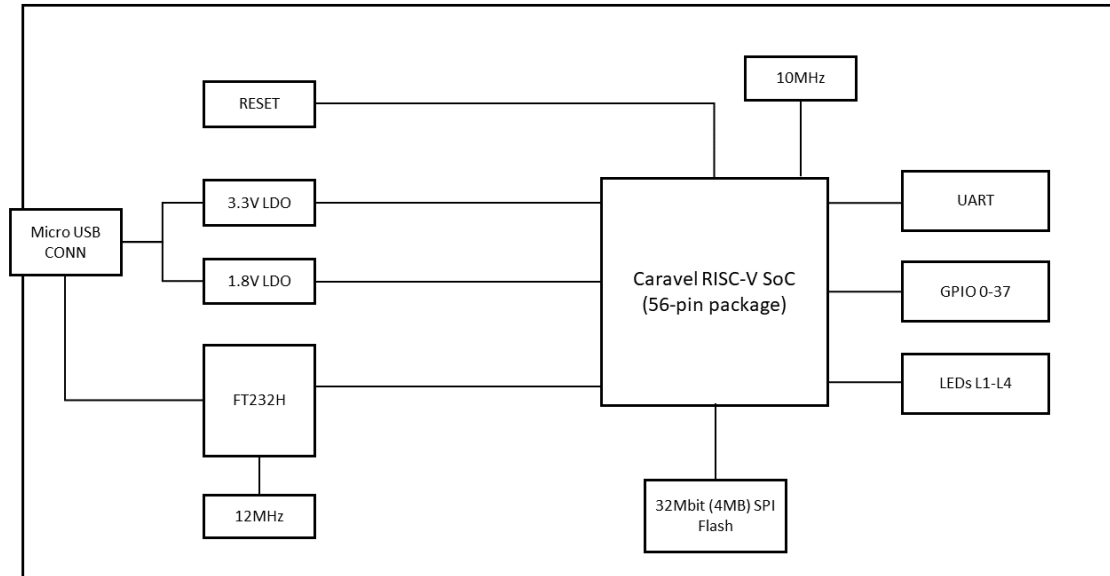


Figure 1: vsdsquadron RISC-V SoC Kit Block Diagram

1.3 Web Resources

For more information about the vsdsquadron RISC-V SoC device, refer to [Caravel SoC page](#)

1.4 Board Overview

The vsdsquadron RISC-V SoC Kit features a Caravel RISC-V SoC with the following capabilities:

- 10 Mhz External Oscillator
- 38 General Purpose Input Output pins
- UART
- SPI
- 4MB External Flash
- USB Programming support over FTDI

The following illustration highlights various components of the vsdsquadron RISC-V SoC Kit.

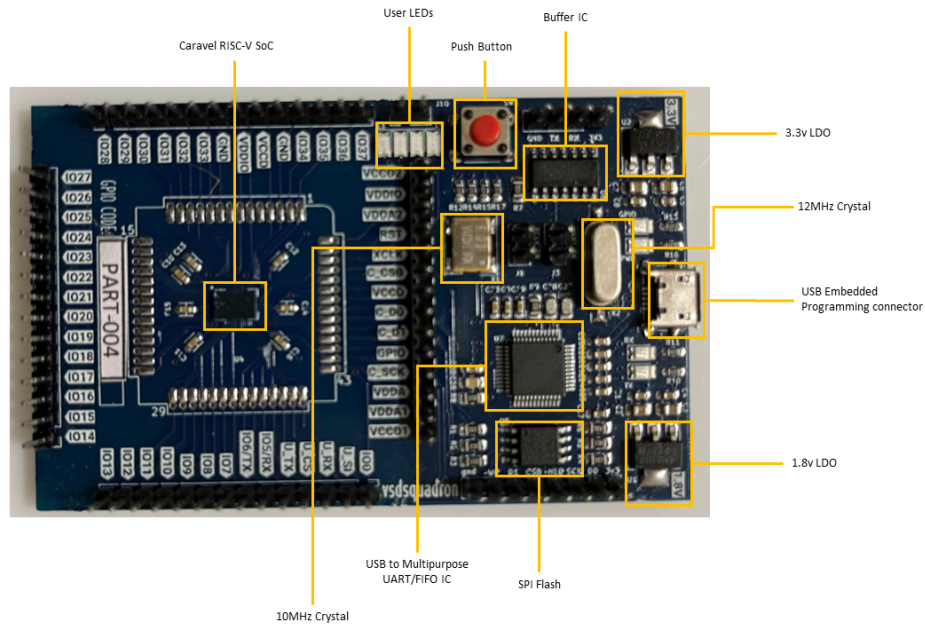


Figure 2: vsdsquadron RISC-V SoC Kit Board image

1.4.1 Form Factor

The following are the dimensions of the vsdsquadron RISC-V SoC Kit.

- Form factor is 84.00 x 52.00 mm
- Maximum height of the component at the top side: 8mm
- Maximum height of the component at the bottom side: 1mm

1.4.2 Caravel RISC-V SoC IO Bank Assignment for Individual Interfaces

Interface	Caravel Bank Allocation
SPI	U_SI , U_RX, U_TX, U_CS
User-defined LEDs	IO0, U_SI, IO36, IO37
UART	IO5, IO6

Table 2: Caravel Bank Assignment

1.4.3 The following table lists the important components of the vdsquadron RISC-V SoC Kit

Component	Label on Board	Description
Featured Device		
Caravel RISC-V SoC	U4	Caravel is a template SoC for Efabless Open MPW and chipIgnite shuttles based on the 130nm node from Skywater Technologies
Power Supply		
5V power supply input	J2	The board is powered by a 5V power source using USB which internally gets converted to 1.8V and 3.3V using LDOs
Clocks		
On board 10MHz HC-MOS Clock Oscillator	X1	10MHz clock to Caravel RISC-V SoC device
12MHz Two pin crystal	X2	12MHz clock which is an input to the FTDI Module
RISC-V programming and debugging		
USB Programming and Reset	J2,SW1	Micro USB connector and reset switch to program the vdsquadron RISC-V SoC device using RISC-V GCC
Memory chips		
SPI flash	U5	32M-bit W25Q32FV Winbond Serial Flash memory device connected to SPI pins of vdsquadron RISC-V SoC device
General Purpose I/O		
Light-emitting diodes (LEDs)	L1-L4	Four active-high LEDs connected to some of the user I/Os for debugging

Table 3: vdsquadron RISC-V SoC Kit Components

1.5 Handling the Board

To avoid causing any damage or malfunctions, it is important to be mindful of the following points when handling or operating the board:

- To prevent any damage, make sure to handle the board while taking electrostatic discharge (ESD) precautions.
- Power down the board by disconnecting the board from USB port

1.6 Operating Temperature

Designed for Room Temperature. The standard range for room temperature in Celsius is typically considered to be between 20 to 25 degrees Celsius (or 68 to 77 degrees Fahrenheit).

1.7 Powering Up the Board

Connect the Micro-B end of micro USB cable to the board as shown in below image and refer to [Installation and Settings section](#) for programming the board

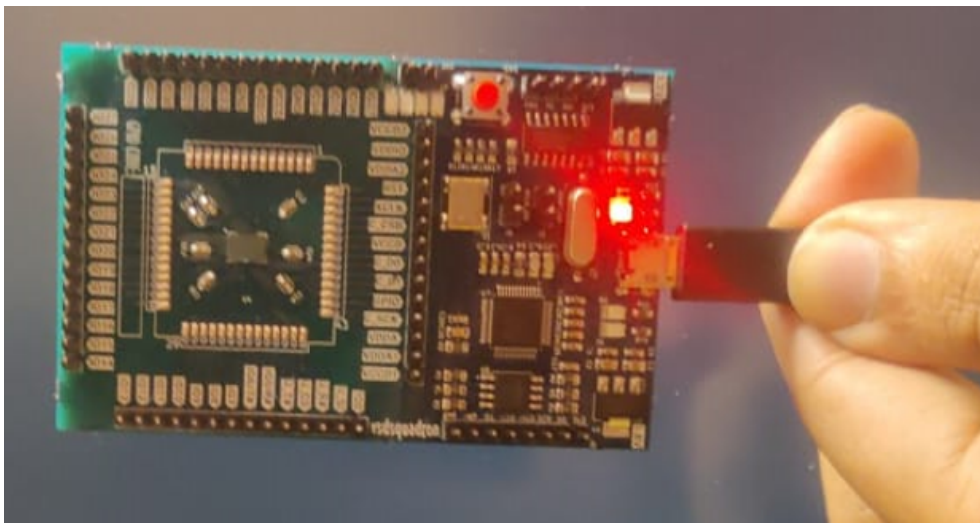


Figure 3: Micro-B end of micro USB cable connected to board

2 Installation and Settings

This section provides information about the software and hardware settings required to run `gpio_test` on the vsdsquadron RISC-V SoC Kit in Linux Ubuntu 20.04 machine

2.1 Software Settings

2.1.1 Toolchain setup

- Install PyFTDI library support. PyFTDI relies on PyUSB, which requires a native dependency: `libusb 1.x`.

```
sudo apt-get install libusb-1.0 libusb-1.0-0-dev
sudo pip3 uninstall setuptools
sudo apt-get install python3-setuptools
sudo pip3 install pyftdi
```

Listing 1: Commands to install PyFTDI library support

- Install the riscv toolchain using below command

```
sudo apt-get install gcc-riscv64-unknown-elf
```

Listing 2: Command to install RISC-V toolchain

- In case this does not work, you can get the sources yourself [here](#)
- Git clone the vsdsquadron software repository using below command

```
git clone https://github.com/yathAg/vsdsquadron_software.git
```

Listing 3: Command to git clone vsdsquadron repository

2.1.2 Programming flow to run `gpio_test`

GPIO test is a simple script to toggle all the GPIOs on the board.

- Compiling: The commands and image below illustrate the process of compiling code.

```
cd vsdsquadron_software/firmware/gpio_test
make PART=<part_id>
```

Listing 4: Command to compile code

Data Flow – Compiling

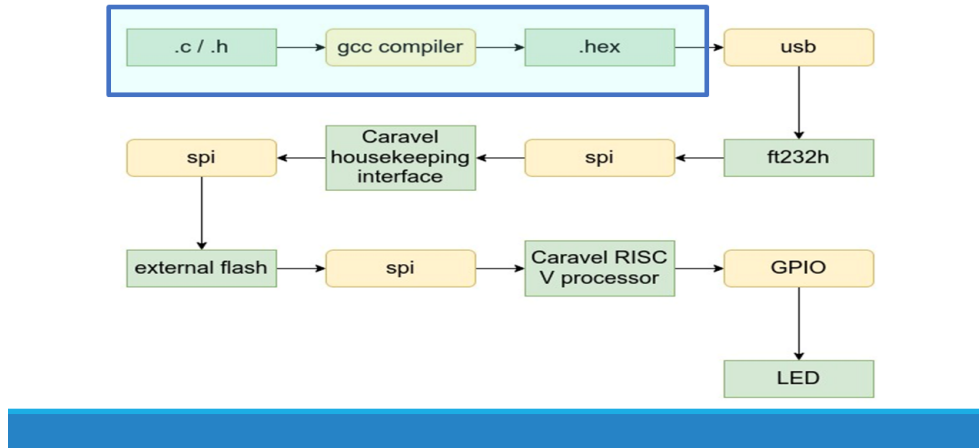


Figure 4: Data Flow - Compiling

- Flashing: The commands and image below illustrate the process of flashing vdsquadron with the compiled code.
 - Make sure Jumper J3 is removed
 - Connect the Type-A end of micro USB cable to the laptop or Desktop (as shown in below image) while holding down the reset button



Figure 5: Type-A end of micro USB cable connected to laptop USB port

- Run the command without releasing the button

```
sudo make flash
```

Listing 5: Command to flash code

- Release the button once you see the following message

```
python3 ../util/caravel_hkflash.py gpio_test.hex
Success: Found one matching FTDI device at ftdi://ftdi:232h:1:a/1

Caravel data:
  mfg      = 0456
  product  = 11
  project ID = 00000000

Resetting Flash...
status = 0x00

JEDEC = b'ef4016'
Erasing chip...
```

Figure 6: Flash programming message

- The below image illustrates the process of flashing vsdsquadron with the compiled code

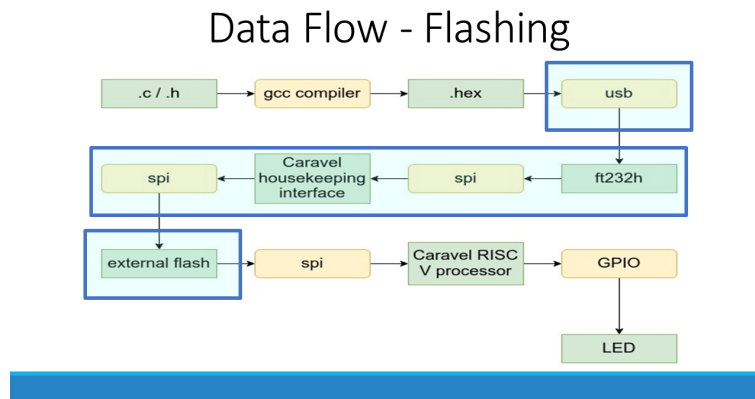


Figure 7: Data Flow - Flashing

- Executing: The above steps should flash the board with the below message. This should result in L1, L2, and GPIO Led blinking with a frequency of 2Hz

```
python3 ../util/caravel_hkflash.py gpio_test.hex
Success: Found one matching FTDI device at
  ftdi://ftdi:232h:1:a/1

Caravel data:
  mfg      = 0456
  product  = 11
  project ID = 00000000

Resetting Flash...
status = 0x00
```

```
JEDEC = b'ef4016'  
Erasing chip...  
done  
status = 0x0  
setting address to 0x0  
addr 0x0: flash page write successful  
addr 0x100: flash page write successful  
addr 0x200: flash page write successful  
addr 0x300: flash page write successful  
addr 0x400: flash page write successful  
addr 0x500: flash page write successful  
addr 0x600: flash page write successful  
addr 0x700: flash page write successful  
addr 0x800: flash page write successful  
setting address to 0x900  
addr 0x900: flash page write successful  
  
total_bytes = 2552  
status reg_1 = 0x0  
status reg_2 = 0x2  
*****  
verifying...  
*****  
status reg_1 = 0x0  
status reg_2 = 0x2  
setting address to 0x0  
addr 0x0: read compare successful  
addr 0x100: read compare successful  
addr 0x200: read compare successful  
addr 0x300: read compare successful  
addr 0x400: read compare successful  
addr 0x500: read compare successful  
addr 0x600: read compare successful  
addr 0x700: read compare successful  
addr 0x800: read compare successful  
setting address to 0x900  
addr 0x900: read compare successful  
  
total_bytes = 2552  
pll_trim = b'00'  
  
python3 ../util/caravel_hkstop.py  
Success: Found one matching FTDI device at  
ftdi://ftdi:232h:1:a/1
```

Listing 6: Command to flash code

The below image illustrates the process of executing vsdsquadron with the compiled code

stored in flash

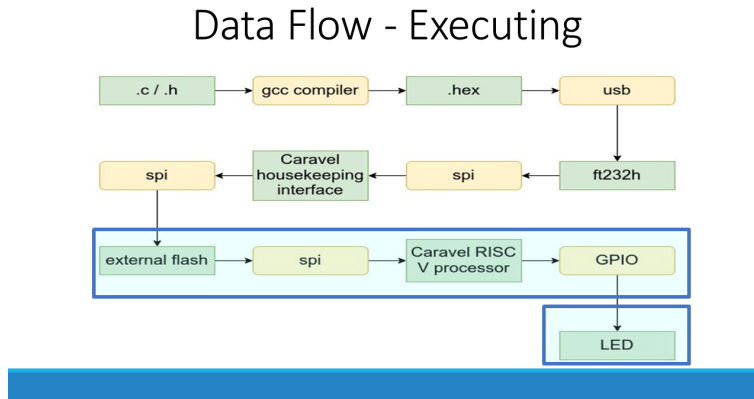


Figure 8: Data Flow - Executing

2.2 Hardware Settings

This section provides information about jumper settings, switches, and LEDs on the vsdsquadron RISC-V SoC Kit.

Pin Header	Function	Active Low/High
J1	Caravel Clock Enable	Active LOW
J2	Micro USB connector	
J3	UART enable	Active HIGH
J4	UART header	
J5	FLASH Header	
J10	User LED Input	

Table 4: Pin definitions and settings

2.3 Power Sources, GPIOs and FTDI

The following table lists the LEDs for power supply, GPIOs and FTDI on the vsdsquadron RISC-V SoC Kit.

LED	Function
PWR LED	3.3V OK
GPIO LED	Management GPIO
RX LED	FTDI Receive signal
TX LED	FTDI Transmit signal
L1	Caravel GPIO 0
L2	Caravel GPIO 37
L3	User GPIO 1
L4	User GPIO 2

Table 5: LEDs for Power supply, GPIOs and FTDI

3 Board Component Placement

The following figure shows the placement of various components on the vsdsquadron RISC-V SoC Kit silkscreen.

3.1 vsdsquadron top view

The following figure shows the top view of the vsdsquadron RISC-V SoC Kit silkscreen.

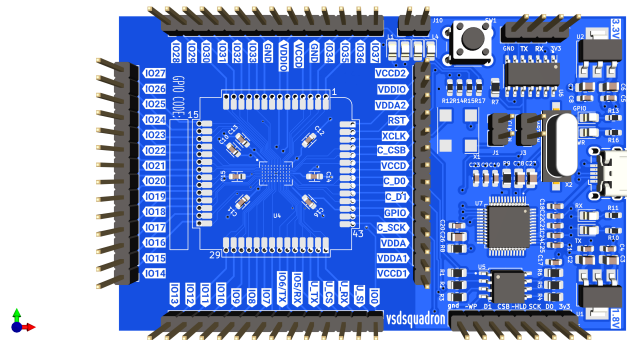


Figure 9: Silkscreen top view

3.2 vsdsquadron bottom view

The following figure shows the top view of the vsdsquadron RISC-V SoC Kit silkscreen.

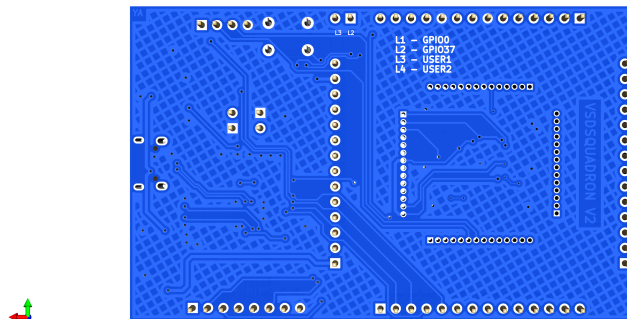


Figure 10: Silkscreen bottom view

4 Appendix: Transmitting data over UART

4.1 Installing and running serial terminal to see messages

- To install picocom

```
sudo apt install picocom
```

Listing 7: Command to install picocom

- Launch picocom using

```
picocom -b 9600  
/dev/serial/by-id/usb-FTDI_Single_RS232-HS-if00-port0
```

Listing 8: Command to install picocom

- -b defines the baud rate
- /dev/serial/by-id/usb-FTDI_Single_RS232-HS-if00-port0 is the serial id of the board it can also be found using the below command:
ls /dev/serial/by-id/

4.2 Flashing UART code and viewing data

- Compiling: The commands and image below illustrate the process of compiling code.

```
cd vsdsquadron_software/firmware/uart_test  
make PART=<part_id>
```

Listing 9: Command to compile code

- Flashing: The commands and image below illustrate the process of flashing vsdsquadron with the compiled code. Refer to [Installation and Settings section](#) for more details on how to program the board

- Make sure Jumper J3 is removed
- Connect the Type-A end of micro USB cable to the laptop or Desktop while holding down the reset button
- Run the command without releasing the button

```
sudo make flash
```

Listing 10: Command to flash code

- Release the button once you see the "Erasing chip" message
- Connect J3 and Power Cycle the board (Turn off and then Turn On)
- Open serial terminal to see the message "Hello from VSDSQUADRON"

5 Revision History

The document's revision history provides a record of the alterations made to it, listed in chronological order, with the most recent revision first.

Revision	Date	Description
1.0	-	This is the first publication of this document

Table 6: Revision History

The VSD Website

VSD's website, <https://www.vlsisystemdesign.com/>, serves as a platform for providing customers with online support and access to a range of resources. These resources include:

- **Product Support:** Customers can access a variety of helpful resources such as data sheets, errata, application notes, sample programs, design resources, user's guides, and hardware support documents. They can also find information on the latest software releases and archived software.
- **General Technical Support:** A range of resources are available on the website to help customers with their technical queries and support requests. These resources include FAQs, technical support requests, and online discussion groups.
- **Business of VSD:** The website also features information about the business of VSD, including product selectors and ordering guides, the latest VSD press releases, seminar and event listings, and a directory of VSD sales offices, distributors, and factory representatives.

Product Change Notification Service

The service offered by VSD for notifying customers about changes to their products is designed to ensure that customers stay informed about the latest updates. Those who subscribe to this service will receive email notifications whenever changes, updates, revisions, or errata related to a specific product family or development tool that they are interested in occur.

Customer Support

VSD offers various channels for users of their products to receive assistance. These include:

- Product Specialist
- Field Application Engineer (FAE)
- Online Support
- Knowledge Base
- Community Forums

Customers are encouraged to reach out to their distributor, representative or FAE for support. Local sales offices are also available to assist customers. A list of sales offices and their locations can be found in the document.

Additionally, technical support is available to customers through [VSD's website](#). This can be accessed 24/7 and includes helpful resources such as product manuals, FAQs, and troubleshooting guides.

If customers require further assistance, they can also submit a support request through the website. VSD's technical support team will respond to these requests as quickly as possible to help resolve any issues that may arise.

Overall, VSD is committed to providing excellent customer service and support to ensure that users of their products have a positive experience.

VSD Devices Intellectual Property Protection

- The design of VSD products is based on the specifications outlined in their corresponding data sheets.
- According to VSD, their products are considered secure under normal conditions and when used as intended.
- VSD acknowledges that there may be attempts to breach the Intellectual Property protection features of their devices using dishonest and potentially illegal methods. VSD asserts that these methods often involve using their products outside of the operating specifications listed in their data sheets, and any attempt to circumvent these Intellectual Property protection features may result in a violation of VSD's intellectual property rights.
- VSD is dedicated to collaborating with customers who express concerns regarding the security of their Intellectual Property.
- It should be noted that no semiconductor manufacturer, including VSD, can guarantee absolute security of their Intellectual Property. Although Intellectual Property protection measures are implemented, they are constantly evolving to address emerging threats. VSD is committed to continually enhancing their Intellectual Property protection features.
- Trying to bypass VSD's Intellectual Property protection feature could be considered a breach of the Digital Millennium Copyright Act. If such actions lead to unauthorized access to copyrighted works or software, legal action may be pursued under the Act.

Legal Notice

The information provided in this publication is intended solely for designing and using VSD products. Information related to device applications is provided for convenience purposes only and may be subject to updates. It is the responsibility of the user to ensure that their application meets their specifications.

VSD provides this information "as is" and makes no representations or warranties of any kind, whether express or implied, written or oral, statutory or otherwise, related to the information. This includes but is not limited to any implied warranties of non-infringement, merchantability, and fitness for a particular purpose or warranties related to its condition, quality, or performance.

VSD shall not be liable for any indirect, special, punitive, incidental, or consequential loss, damage, cost, or expense of any kind whatsoever related to the information or its use, even if VSD has been advised of the possibility of such damages or if the damages are foreseeable. To the fullest extent allowed by law, VSD's total liability on all claims related to the information or its use will not exceed the amount of fees, if any, paid directly to VSD for the information.

The use of VSD devices in life support and/or safety applications is at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless VSD from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any VSD intellectual property rights unless otherwise stated.