Introduction

The evaluation board is designed to help the customer evaluate the 9ZXL1951D. The device is programmable through an SMBus interface. This user guide details the board set and connection, as well as the companion GUI installation for communicating to the device. The board has a self-contained USB to SMBus interface.

Board Overview

Use Figure 1 and Table 1 to identify: USB connector, input and output frequency SMA connectors.

Figure 1. Evaluation Board Overview

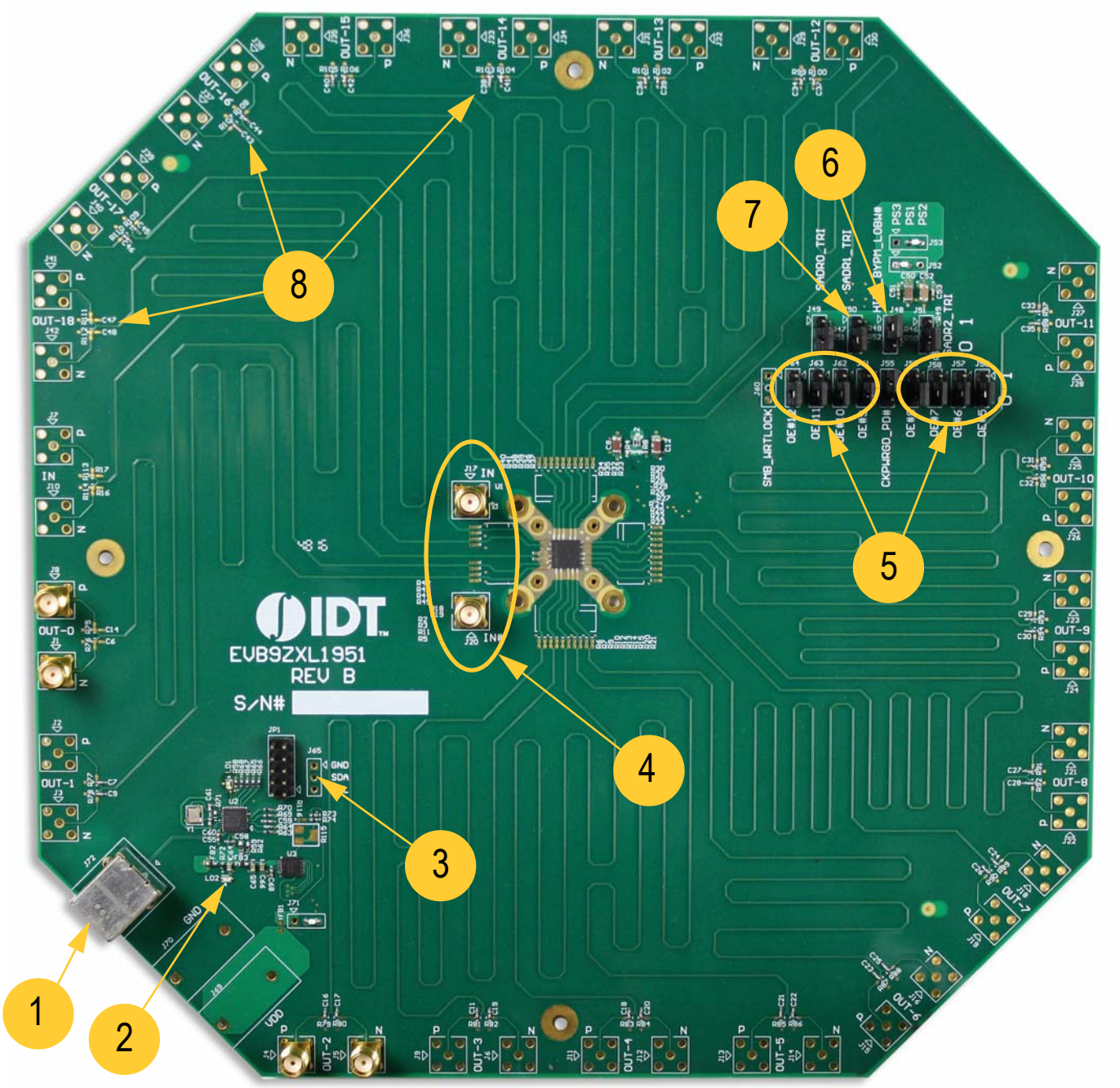


Table 1. Evaluation Board Pins and Functions

Label Number	Name	On-board Connector Label	Function
1	USB Interface	J72	Used for power-up of the device and connection with a PC, and for interaction with the IDT PCIe GUI. On-board USB to SMBus connection.
2	Power LED	LD2	Power from USB connector.
3	SMBus Header	J1	External SMBus connection.
4	Input Clock	J20, J17	Input clock SMA connector.
5	Jumper Setting	J64, J63, J62, J61 J59,J58, J57, J56	Output Enable for: OUT12, OUT11, OUT10, OUT9, OUT8, OUT7, OUT6, OUT5.
6	PLL Operation	J48	▪ PLL High Bandwidth mode: Connect J48 pin 1 and pin 2. ▪ PLL Bypass mode: Remove jumper in J48. ▪ PLL Low Bandwidth mode: Connect J48 pin 2 and pin 3. Note: Jumper J48 setting needs to power-cycle the board to take effect.
7	Address Select	J49, J50	SMB_A1 / SMB_A0 / SMB_Addr 0 / 0 / 0xD8 0 / M / 0xDA 0 / 1 / 0xDE M / 0 / 0xC2 M / M / 0xC4 M / 1 / 0xC6 1 / 0 / 0xCA 1 / M / 0xCC 1 / 1 / 0xCE
8	Output Ports	J1 to J42	HCSL clock output.

Board Power Supply

By default, the board is set to be used with the USB power supply.

USB Power Supply

When the board is connected to a PC through a USB cable, the on-board voltage regulators will generate 3.3V for the device.

Connecting the Board

The board is connected to a PC through a USB connector for configuring and programming the device. The USB interface also provides a +5V power supply to the board, from which on-board voltage regulators generate various voltages for the core as well as for each output. The LD2 power LED will light up to indicate a successful connection.

PCIe GUI Installation Setup

Download the PCIe GUI software. The drivers should automatically install. If they do not, follow the instructions below.

First, the GUI requires a driver for the FTDI IC to interface between the USB and SMBus interfaces.

1. Unzip the files from the PCIe GUI archive on your PC.
2. Extract the FTDI windows driver from the PCIe GUI archive or go to the FTDI website to download the latest driver and install on your computer (see Figure 2).

Note: For non-Windows operating systems, download the respective driver from the FTDI website.

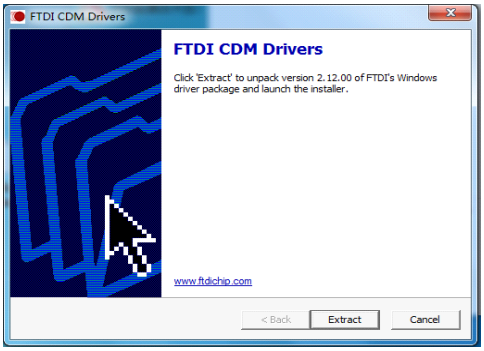
Figure 2. FTDI Currently Supported D2XX Drivers

Currently Supported D2XX Drivers:

		Processor Architecture							
Operating System	Release Date	x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4	Comments
Windows*	2014-09-29	Available as setup executable Contact support1@ftdichip.com if looking to create cusomised drivers		-	-	-	-	-	2.12.00 WHQL Certified Available as setup executable Release Notes
Windows RT	2014-07-04	1.0.2	-	-	1.0.2	-	-	-	A guide to support the driver (AN_271) is available here
Linux	2012-06-29	1.1.12	1.1.12	-	1.1.12 Suitable for Raspberry Pi	-	-	-	ReadMe
Mac OS X	2012-10-30	1.2.2	1.2.2	1.2.2	-	-	-	-	Requires Mac OS X 10.4 (Tiger) or later ReadMe
Windows CE 4.2-5.2**	2014-22-04	1.0.1.10	-	-	1.0.1.10	1.0.1.6	1.0.1.6	1.0.1.6	
Windows CE 6.0/7.0	2014-22-04	1.0.1.10 CE 6.0 CAT CE 7.0 CAT	-	-	1.0.1.10 CE 6.0 CAT CE 7.0 CAT	1.0.1.6	1.0.1.6	1.0.1.6	For use of the CAT files supplied for ARM and x86 builds refer to AN_319

3. Double click the executable file to install the driver (see Figure 3).

Figure 3. FTDI CDM Drivers Executable File Window



4. Connect the 9ZXL1951D board to the computer using the supplied USB cable. Double-click on the application file *ClockCtl.exe* to start the PCIe GUI support application. The PCIe Clock/Buffer GUI main window appears (see Figure 4). If no board is connected, the following message will appear:

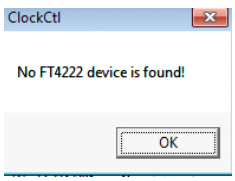


Figure 4. PCIe Clock/Buffer GUI Main Window

See Table 2 for descriptions.

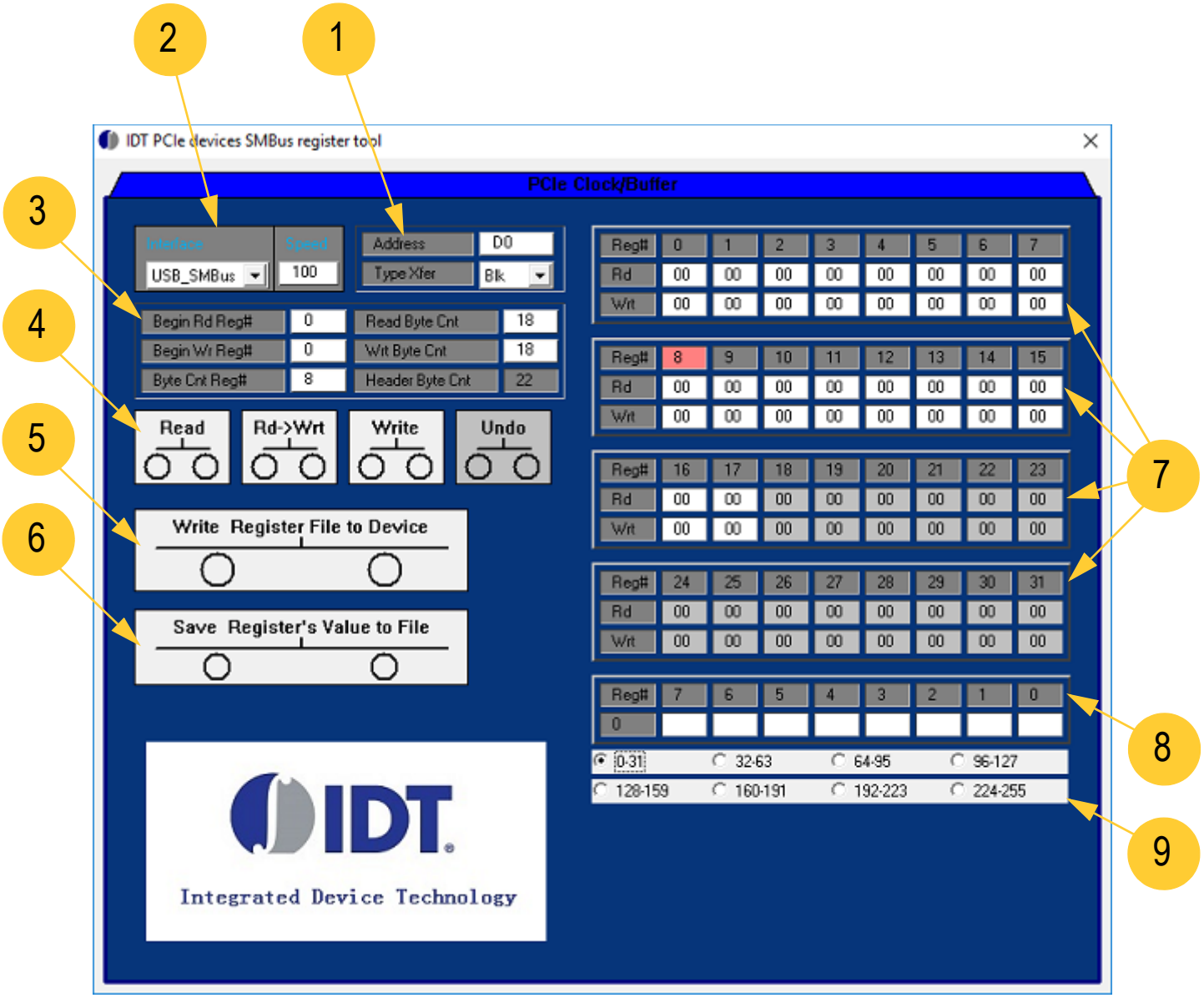


Table 2. PCIe Clock/Buffer GUI Main Window Label Descriptions

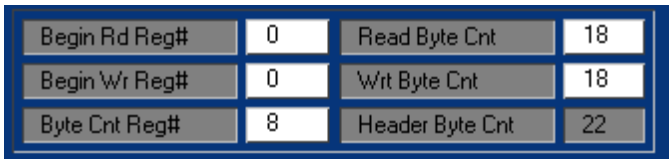
Label #	Name	Description
1	Slave Address/TypeXfer	The address is 7-bit slave address combined with “0” in LSB. For example, if the slave address is 1101000, “D0” should be filled.  There are three modes for transfer type: “Blk”, “Byte” and “Word”. The device is SMBus block (Blk) mode protocol by default. 
2	SMBus Interface	Only USB to SMBus is available. The SMBus speed (in kHz) can be changed. Note that the speed of SMBus is from 10kHz to 100kHz. 
3	Begin Reg# and Byte Count	▪ *Begin Rd Reg#* is the begin register address of a read operation. ▪ *Read Byte Cnt* is the byte count of a read operation. ▪ *Begin Wr Reg#* is the begin register address of a write operation. ▪ *Wrt Byte Cnt* is the byte count of a write operation. 

Table 2. PCIe Clock/Buffer GUI Main Window Label Descriptions (Cont.)

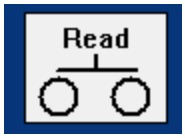
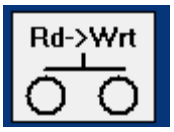
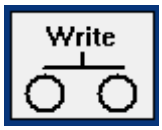
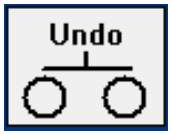
Label #	Name	Description
4	Register Operations	▪ *Read* operation Clicking the *Read* button initiates a read. If a chipset is used for reading, the byte count is determined by the value in the device byte count register. The byte count cannot be larger than 32 dec. Non-read locations in the read grid will be grayed out.  ▪ *Rd->Wrt* operation Clicking the Rd>Wrt button copies all of the read cells to the write cell contents.  ▪ *Write* operation *Write* button operation. If the chipset is used for writing, the byte count is controlled by the value in the GUI panel byte count register. Registers that will not be written because of the starting location setting and byte count will be grayed out. The hex values for data to be written will be in a cell with a white background.  ▪ *Undo* operation Reverts back to the last performed operation. 

Table 2. PCIe Clock/Buffer GUI Main Window Label Descriptions (Cont.)

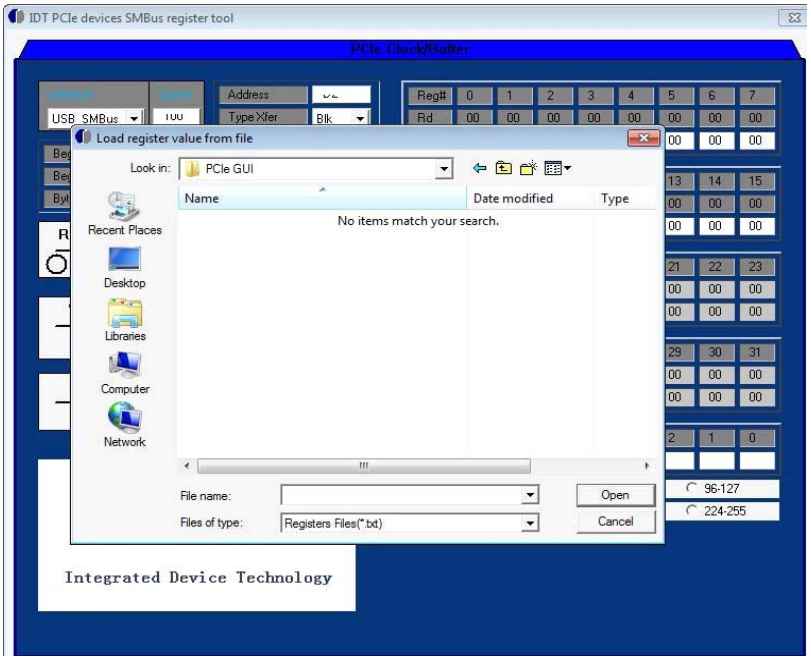
Label #	Name	Description
5	Write Register File to Device	Clicking the “Write Register File to Device” button writes the register from file.A pop-up a window appears allowing the selection of the file path and the file name. Once selected, click “Open”. The GUI will read all the register’s value from the file and then download to the device.  

Table 2. PCIe Clock/Buffer GUI Main Window Label Descriptions (Cont.)

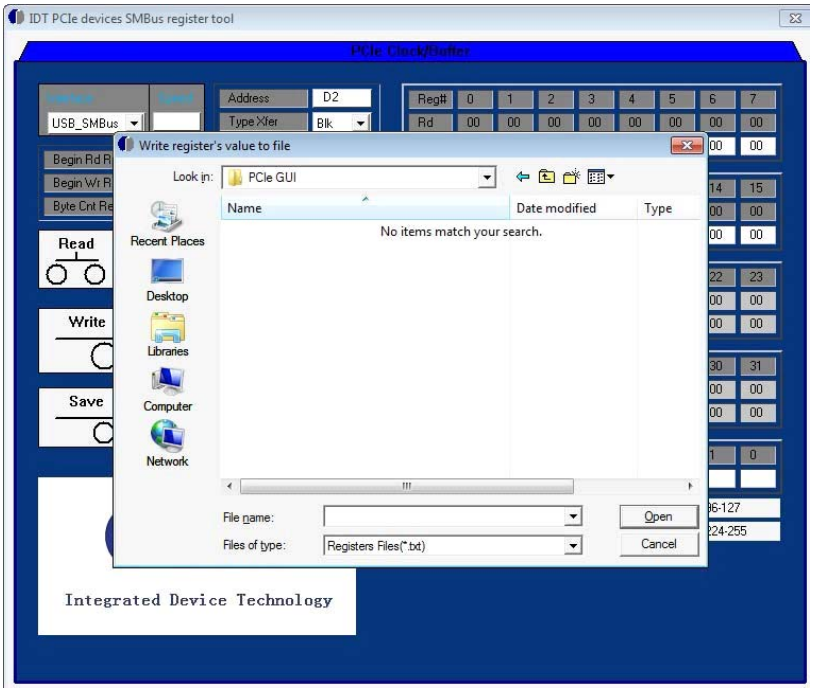
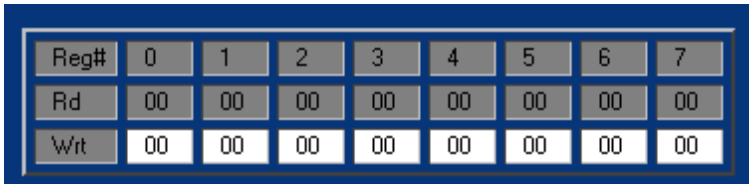
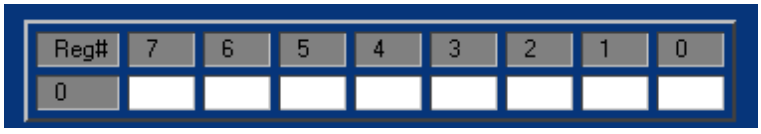
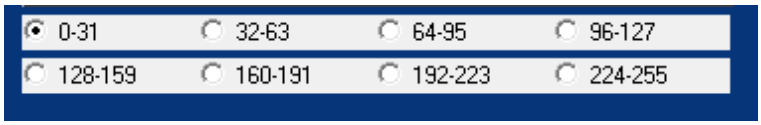
Label #	Name	Description
6	Save Register's Value to File	Clicking the "Save Registers Value to File" button saves the registers to the file, A pop-up window is displayed. Select the file path and enter a file name, then click "Save". The GUI will dump all the register's value then save to the file.  
7	Register Value Field	The hexadecimal *read* information appears as a grayed background, meaning that it cannot be altered. Hexadecimal *write* information appears on a white background. 
8	Binary Display Table	Clicking on a *Reg#* or *Rd* window displays the binary decode of the hex value. This may be used for entering binary data instead of hexadecimal data. 

Table 2. PCIe Clock/Buffer GUI Main Window Label Descriptions (Cont.)

Label #	Name	Description
9	Byte Count Range Switch	A 32-byte value is the default display. If the byte count exceeds 32, select the radio button that coincides with the new range. 

Read/Write Operations

Read

Pressing the read button initiates a read. If a chip set is used for reading, the byte count is determined by the value in the device byte count register. The byte count cannot be larger than 32 dec. Non-read locations in the read grid will be grayed out.

Rd->Wrt

Pressing the Rd->Wrt button will copies all of the read cells to the write cell contents.

Write

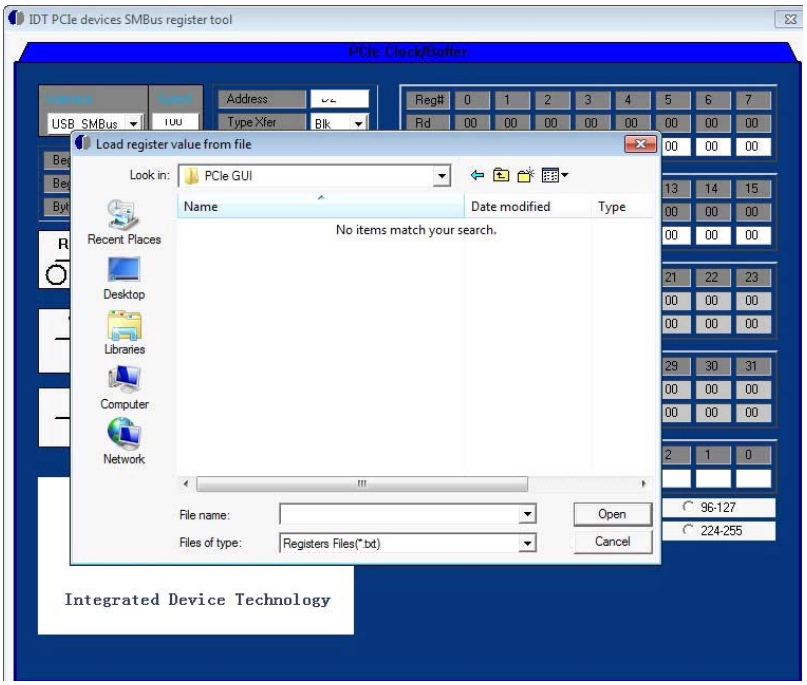
Write button operation. If the chip set is used for writing, the byte count is controlled by the value in the GUI panel byte count register. Registers that will not be written because of the starting location setting and byte count will be grayed out.

The hex values for data to be written will be in a cell with a white background.

Read/Write from File

To Write register from file, click the “Write Register File to Device” button. A pop-up window is displayed (see Figure 5). Select the file path and enter a file name, then click “Open”. The GUI will read all register’s value from the file, then download to the device.

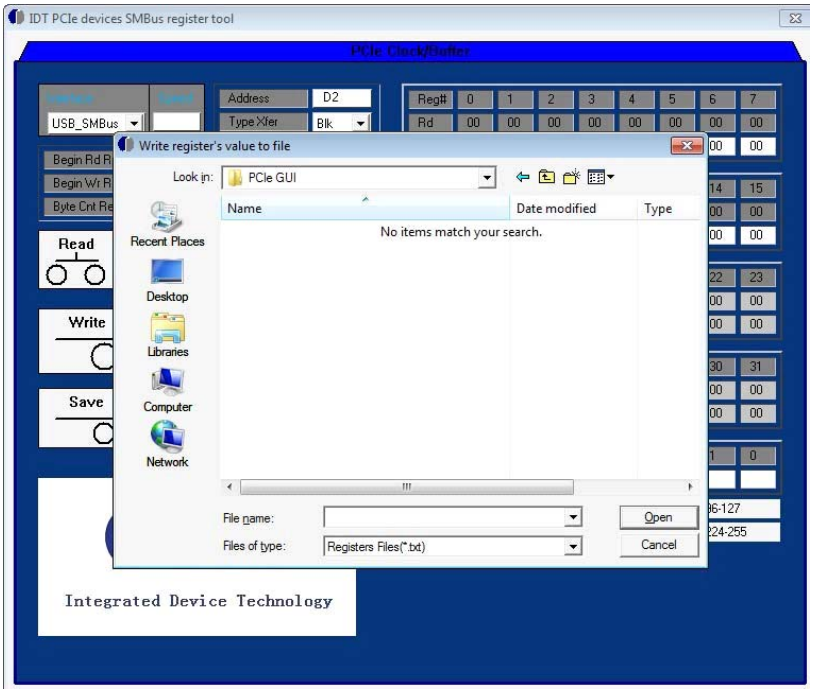
Figure 5. Load Register Value from File Pop-up Window



Clicking the “Save Registers Value to File” button saves the registers to a file. A pop-up window is displayed (see Figure 6). Select the file path and enter a file name, then click “Save”. The GUI will dump all register’s value then save to the file.

Note: LED LD1 will light up on every SDATA operation.

Figure 6. Write Register Value from File Pop-up Window



Schematics

Figure 7. 9ZXL1951D Connections

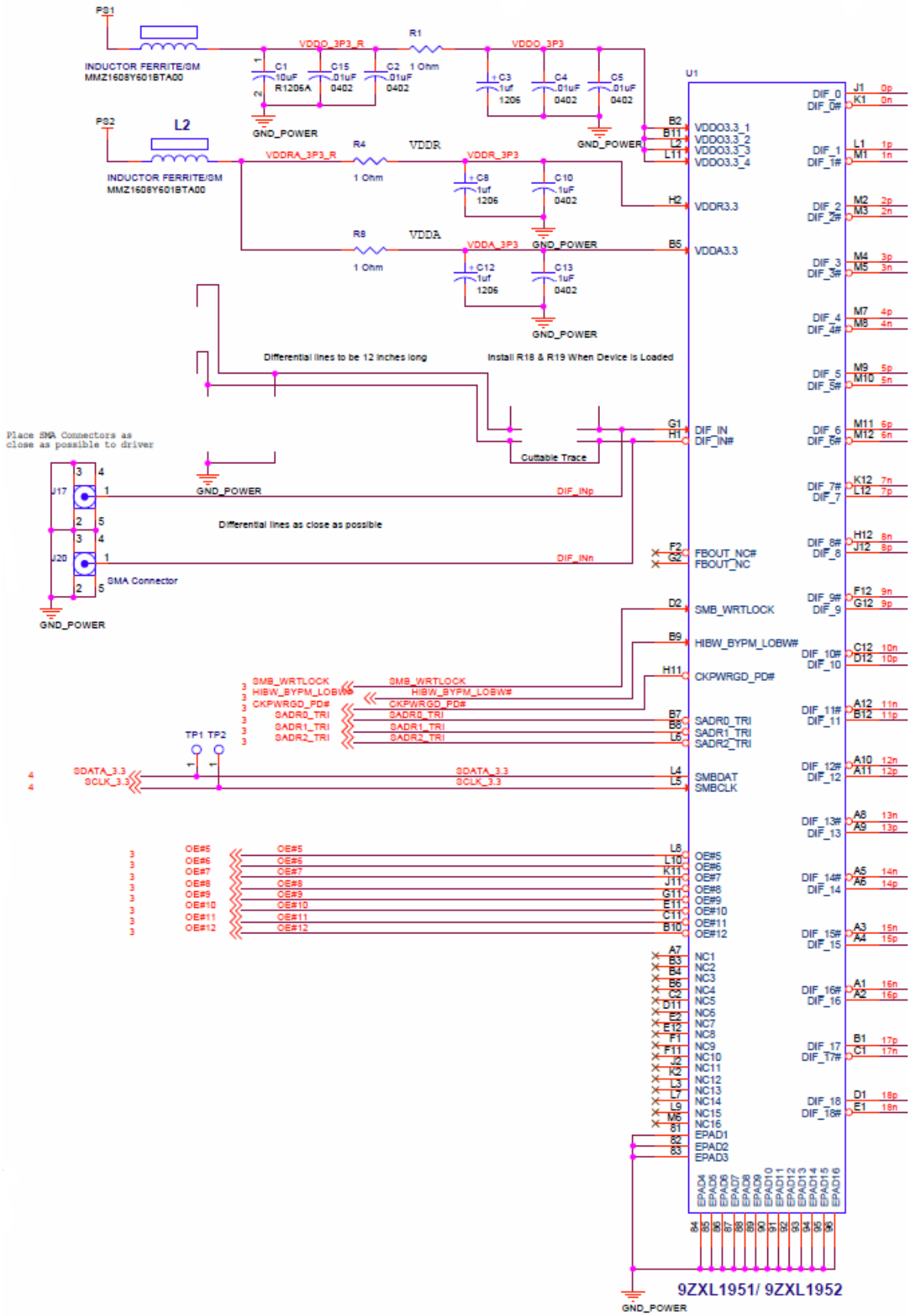
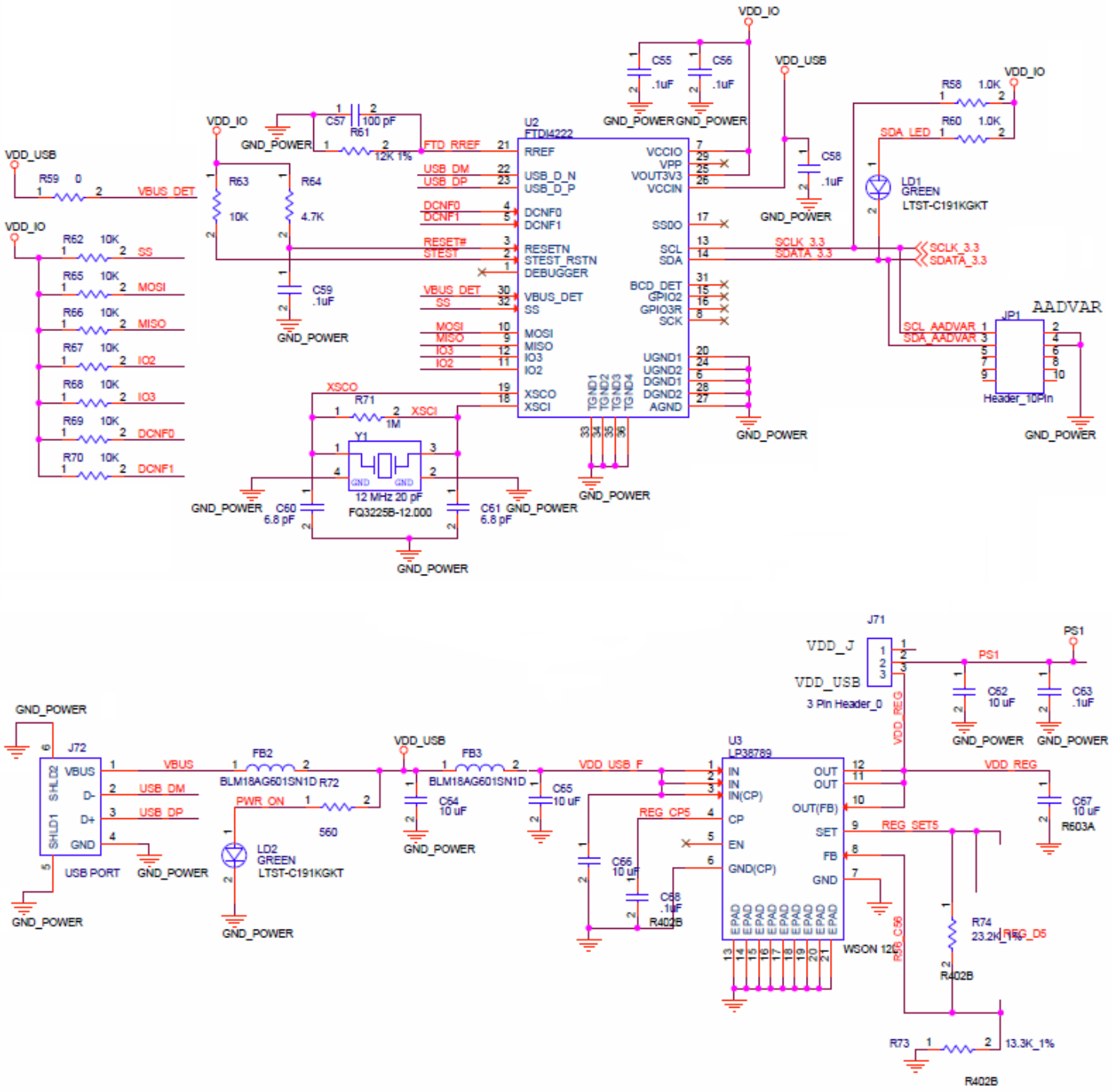


Figure 8. USB Interface and Power



Ordering Information

Orderable Part Number	Description
EVK9ZXL1951D	9ZXL1951D Evaluation Kit

Revision History

Revision Date	Description of Change
March 23, 2018	Initial release.