



DOMAIN TECHNOLOGIES

Users Guide version 1.1

DTSLIC2L

Telephony Daughterboard

DTSLIC2L Daughterboard Users Guide October 2005

Domain Technologies, Inc.
811 E Plano Pkwy, Suite 115
Plano, TX 75074

Tel.: +1-972-578-1121

Fax: +1-972-578-1086

<http://www.domaintec.com>

email: support@domaintec.com

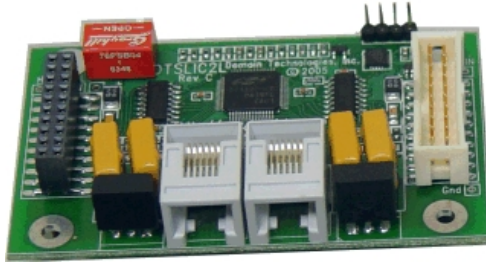
Contents:

1. INTRODUCTION..... 4
2. HARDWARE SETUP..... 5
3. SAMPLE SOFTWARE. 6
4. MICROCONTROLLER COMMUNICATION PROTOCOL. 9
APPENDIX A: DTCLIC2L SCHEMATICS..... 11
APPENDIX B: DTSLIC2L SILKSCREEN WITH COMPONENTS LOCATION..... 13
APPENDIX C: DTCLICPWR SCHEMATICS 14
APPENDIX D: DTSLICPWR SILKSCREEN WITH COMPONENTS LOCATION 15

1. Introduction.

The DtSlicGui.EXE application illustrates communication with the DTSLIC2L from a PC through the ZSP host port. The DtSlicGui.EXE program is designed for use with any ZSP development board featuring a SPORT connector. Domain Technologies' EB403LC is featured in this manual.

DTSLIC2L



The Dual Subscriber – Line Interface Circuit (SLIC) card features a complete dual analog phone line. The DTSLIC2L contains two powered line interface jacks and a compact, economical dual-channel analog telephone interface: the SI3220-FQ Dual ProSLIC(R). The Si3220 is controlled by an 8-bit micro-controller which interfaces to the ZSP through a 2 line I²C interface. The SI3220 is fully programmable and has on-chip functions for DTMF generation/decoding, FSK caller ID generation, and modem tone detection help; in addition, its integrated test and line monitoring feature allows remote subscriber loop and line card diagnostics to be performed without centralized test equipment. Programmable internal ring generation is provided also.

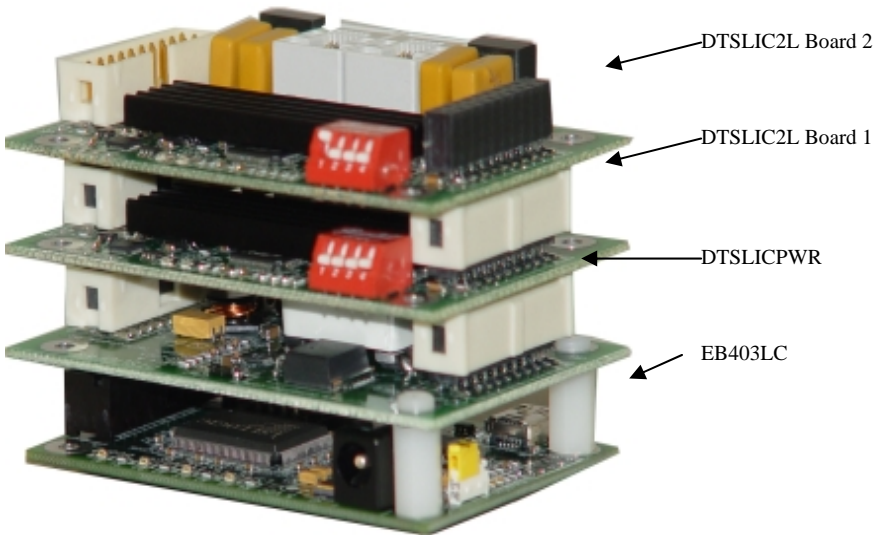
DTSLICPWR



The DTSLIC2L requires the DTSLICPWR board to supply power. Up to 4 DTSLIC2L boards may be powered by one DTSLICPWR. The DTSLICPWR board requires a 7.5V DC power supply.

2. Hardware Setup.

To illustrate the functionality of the DTSLIC2L telephony board, the EB403LC development board will be used to execute the sample application *slic.out*. User control will be provided by the DtSlicGui program running on the PC. Our example will utilize two DTSLIC2L telephony boards as well as the DTSLICPWR board.



Begin by inspecting the EB403LC. Attach the DTSLICPWR board to the EB403LC. Be sure to center the boards when aligning to connect the 20 pin header and socket correctly. If the jumper shunt next to the power connector is installed, the DTSLICPWR will also power the EB403LC. If the jumper is not installed, the EB403LC can be powered by either the USB port or the EB403LC external power supply.

Next, connect the two DTSLIC2L boards to the DTSLICPWR board. The switches on the bottom DTSLIC2L board should be all down; this indicates it is the first SLIC board. The switches on the top DTSLIC2L board should have the left most switch up and the additional 3 switches down; this indicates it is the second SLIC adapter. Finally, connect the 7.5 volt power supply to the DTSLICPWR board. The telephony circuit is powered by the 7.5 volt power supply with current capacity greater than 1 amp.

Note, when using the DT402EB board instead of the EB403LC board: the DT402EB rev. A board must be powered by the USB port or its own external power supply. Do not use the jumper shunt next to the power connector on the DTSLICPWR to provide power to the DT402EB rev. A board.

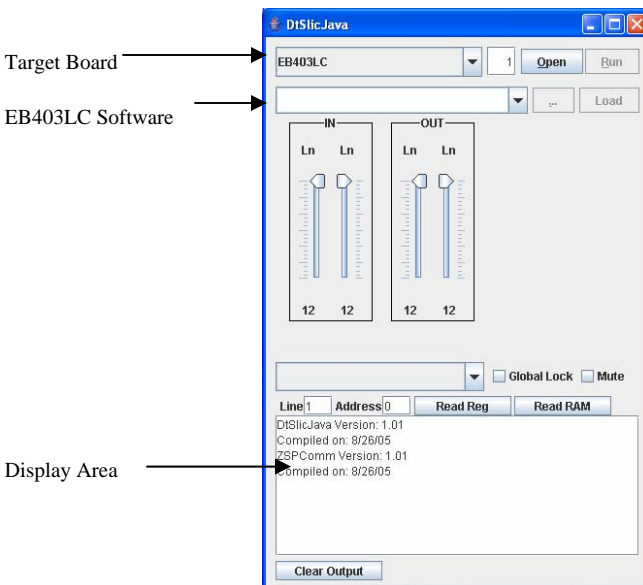
3. Sample Software.

The software installation CD distributed with your ZSP development board contains sample applications to illustrate the functionality of the DTSLIC2L telephony board. The BoxView IDE has following two projects:

Project name	Description
DtSlicGui	Java application providing GUI on PC
DtSlicZsp	Sample ZSP application

DtSlicGui Program

Execute the DtSlicGui application from the BoxView IDE and the following window is displayed:



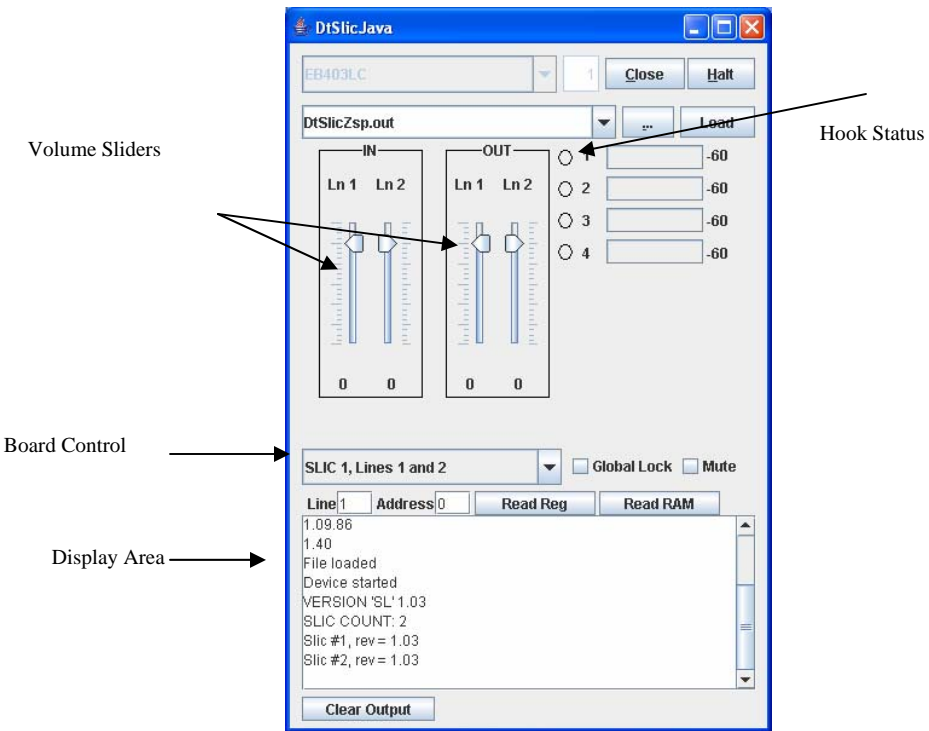
Begin by defining the target ZSP board; in this example, the *EB403LC* is used. Press the *Open* button to initialize the target. Note, the *Open* button will become a *Close* button when the target device has successfully been opened. The *Display area* in the GUI will indicate a target device has been found, the target device's handle and serial number.

Next, define the location of the EB403LC software which will execute on the ZSP; the built ZSP application is located in the DtSlicGui folder. To browse for the application use “...” button.

The ZSP application can be modified within the BoxView IDE, but this operation requires permanent or evaluation license. The Java portion of the GUI is operational without a license.

Choose the *Load* button to load the software on the EB403LC. The Display area of the GUI will be updated with the text “File Loaded.”

Continue by pressing the *Run* button. The ZSP application takes few moments to initialize SLIC daughter board. When the SLIC initialization is finished, the *Display* area will indicate the version information of the executed code. The ZSP code interrogates the SLIC and finds the total number of adaptors as well as the microcontroller firmware revisions. The ZSP communicates with each microcontroller via the I²C interface. Once initialization finishes, each board is queried to determine how many controllers it has on a bus; this information is passed back to the PC and displayed in the *Display* area of the GUI. Note, the *Run* button will be replaced with a *Halt* button.



The DtSlicGui provides volume control with the *Volume Sliders* for 2 lines only. The sliders can be locked for simultaneous control by checking the *Lock A/B* checkbox. Select the SLIC board to control through the *Board Control* combo box. The *Hook Status Indicator* become checked once a line is off hook. Anytime a line is off hook, the software will ring the other line on the same SLIC board. When both lines are answered, voice is sent between the two lines. Selecting the *Mute* checkbox will set all amplifications to their minimum values.

Low level SLIC registers can be read by choosing the *Read Reg* button; the *Display* area will be updated with the 8 bit register information associated with the *Address* field register. The *Write Reg* button will update the register the *Address* field's register with the *Value* field's data.

16-bit RAM information can also be provided in the *Display* area. Provide the appropriate data in the *Address* entry field and select the *Read RAM* button; to write RAM, provide the associated data in the *Value* entry field. Note, RAM 71 and 72 are not displayed by DtSlicGui; they are used for volume control and are utilized by the GUI display. For more information on RAM and Register manipulation, review the [Si3220/Si3225 Programmer's Guide, Application Note \(AN\) 58](#).

Use the *Clear* button at the bottom of the interface to clean the contents from the *Display* area. The *GetVersion* button will obtain the software version information, the total number of adaptors as well as the microcontroller firmware revisions; all will be shown in the *Display* area.

Once testing is complete, choose the *Halt* button to stop activity on the target; the *Halt* button will become replaced by the *Run* button. Choose the *Close* button to close the connection to the target; the *Display* area will indicate the USB Target is closed and the *Close* button will be replaced by a *Open* button.

4. Microcontroller Communication protocol.

Each DTSLIC2L daughterboard needs to be configured at a different address. This is accomplished by settings of the 4 position DIP switch. The I2C address is equal of the switch value * 2 + 0x20. Bit 0 indicates Read/Write operation, bits 6 and 7 of the address value are cleared.

Host processor communicates with the DTSLIC2L microcontroller by sending byte commands through the I2C interface. Some commands are followed by the parameters, for some command microcontroller returns response of 1, 2 or 4 bytes. Details for commands are in the table below.

Selection of the SLIC A/B for each command for register or RAM access is provided by bit 0x20 of each command.

Cmd	Name	Params	Returns	Description
0	GET_VER	-	1 byte	Get code version
1	READ_REG	1 byte	1 byte	Read 8 bit register value
2	WRITE_REG	2 bytes	-	Write 8 bit register
3	READ_RAM	1 byte	2 bytes	Read 16-bit RAM value
4	WRITE_RAM	3 bytes	-	Write 16-bit RAM location
5	READ_CTRL	-	1 byte	Read control bits
6	WRITE_CTRL	1 byte	-	Set control bits
7	CLEAR_MON	-	-	Clear active monitors
8	ADD_MON	4 bytes	-	Add monitor function
9	GET_MON	1 byte	4 bytes	Get monitor settings
10	GET_MON_CNT	1 byte	1 byte	Get count for a monitor
11	READ_STATUS	-	1 byte	Read monitors status
12	SET_LED	1 byte	-	Set led a/b on/off
13	GET_I2C_ADDR	-	1 byte	Get I2C address (debug)

Get Version – retrieve a byte with current version of the microcontroller firmware.

Read Register addr – read the SLIC register at address.

Write Register addr val– write 8-bit value into the SLIC register at address

Read RAM addr – read SLIC RAM location – 16-bit value at address

Write RAM addr valL valH– write SLIC RAM location with a 16-bit value at address

Read Control – get current control settings

Bit(s)	Name	Function
0	RESET	State of the RESET signal
1	VENA	Enable for the 3.3V voice regulator
2	LEDA	Led A on state
3	LEDB	Led B on state

Write Control ctrl – bit controlled: bit 0 – RESET, bit 1- Vcc enable.

Clear Monitors – disable all hardware monitors

Add Monitor oper addr patt last – add monitoring function, up to eight available

Byte	Name	Function
0	OPER	Selects monitor function
1	ADDR	Specifies register address to read
2	PATTERN	Bit mask for bits to check
3	LAST	Value to compare, if different – increment counter

Get Monitor index – get settings for the specified monitor (0..7)

Get Monitor Count index – get events counter for the specified monitor

Get Monitor Status – get status of all monitors and clear it

Set LED ctrl – bit 0 – 0-LedA, 1-LedB, bit 1 – 0-OFF, 1-ON

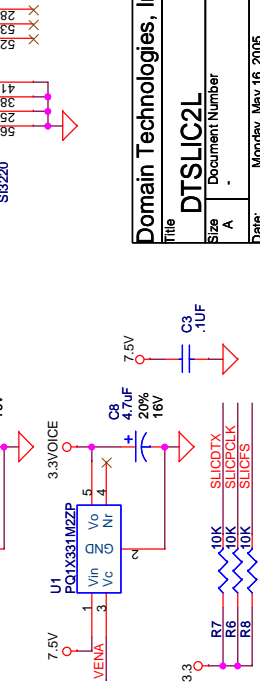
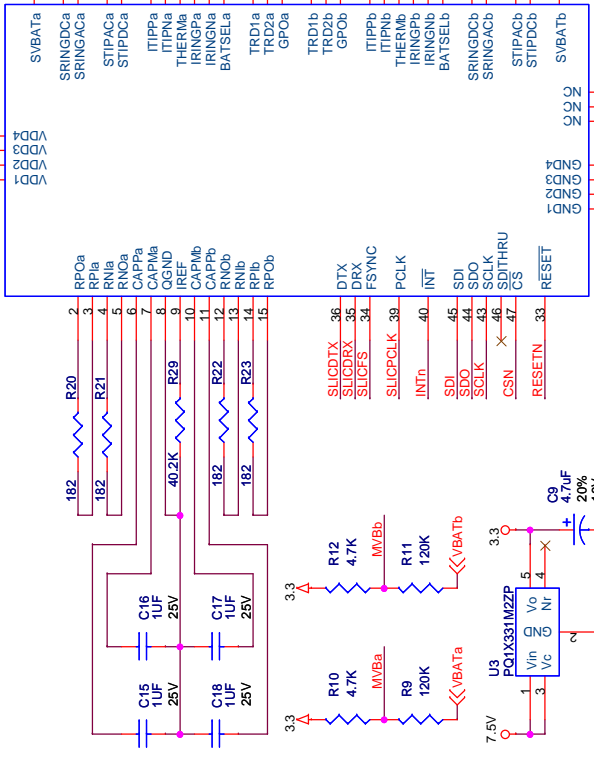
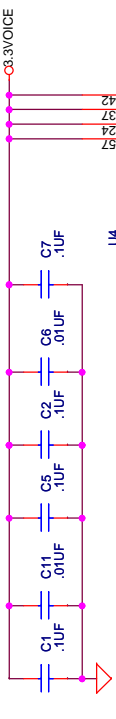
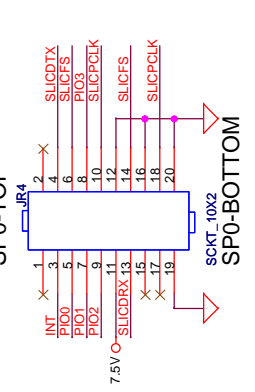
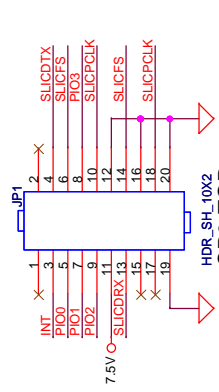
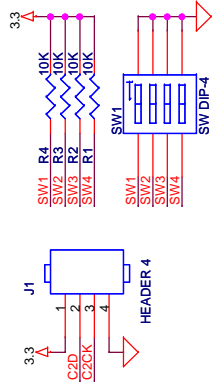
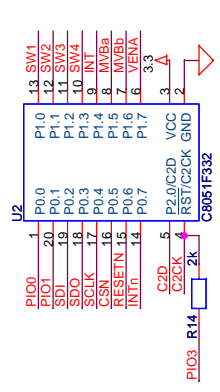
Get I2C Address – get current address – for debug purpose

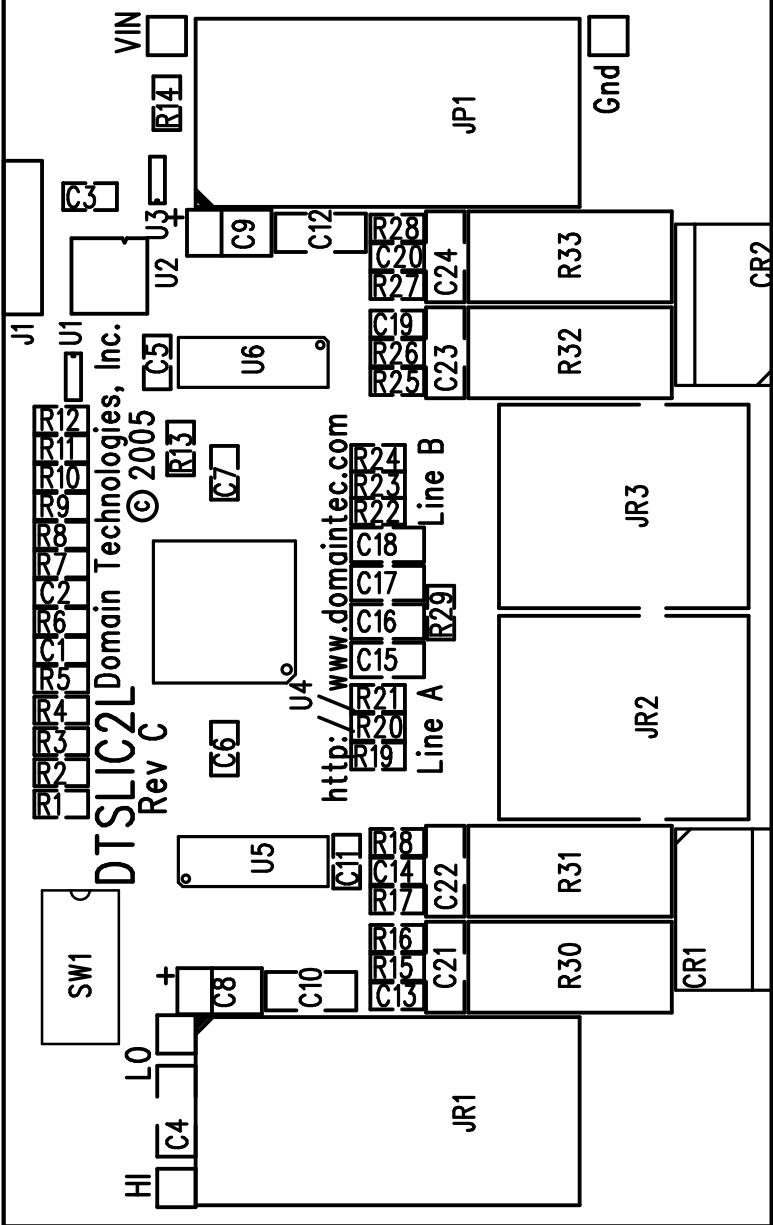
Microcontroller can monitor SLICs operation, and report to the host ZSP detected events by asserting the interrupt, or updating the status byte. This saves ZSP’s processing time, so it does not have to poll the register values continuously.

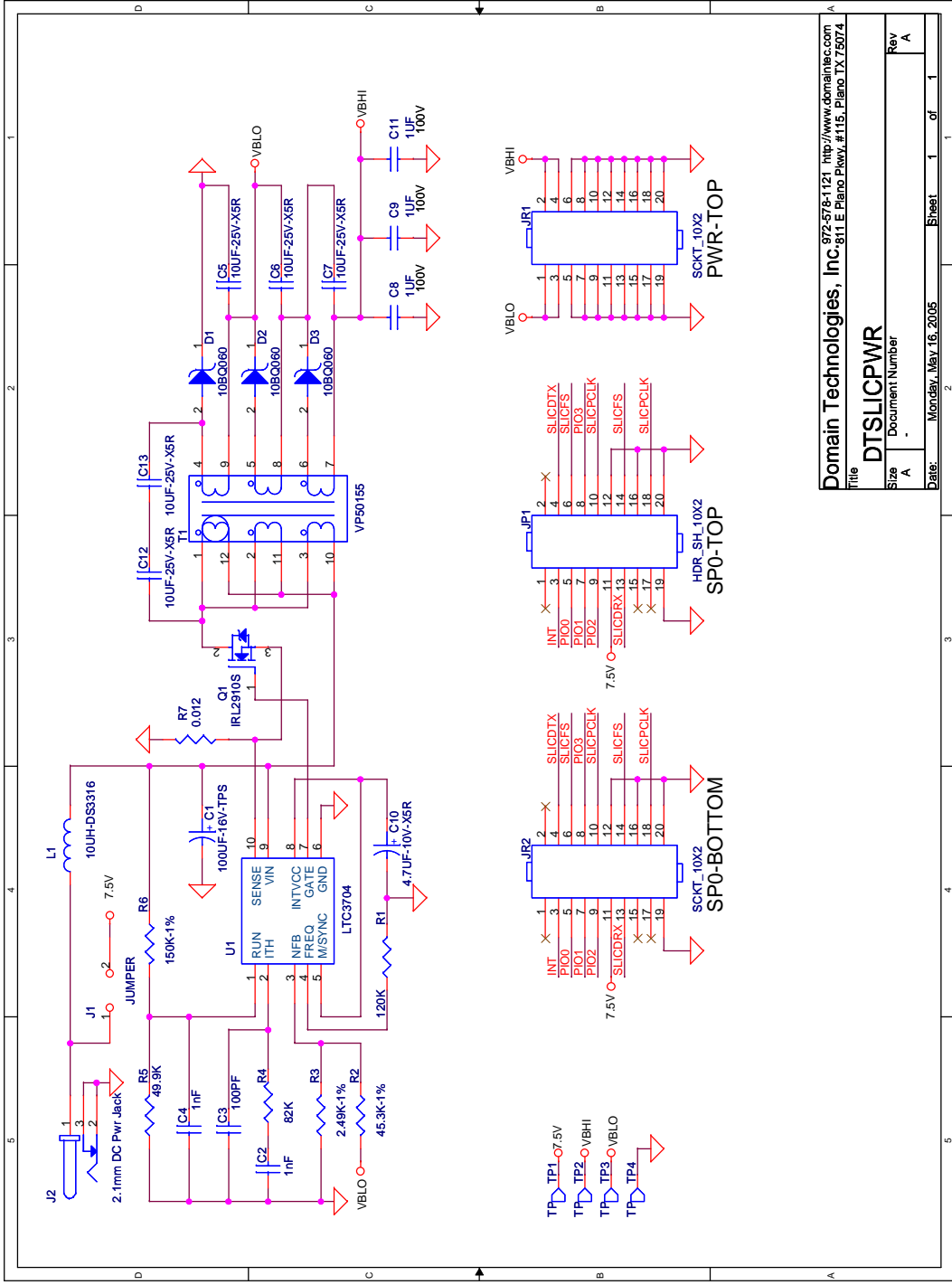
Monitor operations are bit encoded as follows:

Bit	Name	Function
0	RAM	0 – monitor register, 1 – monitor RAM
1	HIGH_RAM	0 – check low byte, 1 – check high byte
2	SLICB	0 – SLIC A, 1 – SLIC B
3	SET	Trigger if masked value is non zero
4	CLEAR	Trigger if masked value is zero
5	SET_INT	Asset Interrupt on event
6	SET_LED	Set LED on – for debug
7	CLR_LED	Set LED off – for debug

For each monitor, register or RAM value is read from the specified address, value is masked with the specified bit mask, and if value matches specified event, interrupt is generated. On each change of the value, the event counter is incremented.







Domain Technologies, Inc. 972-576-1121 <http://www.domaintec.com>
 811 E Plano Pkwy, #115, Plano, TX 75074
 Title: DTSLICPWR
 Size: A Document Number
 Date: Monday, May 16, 2005 Sheet 1 of 1

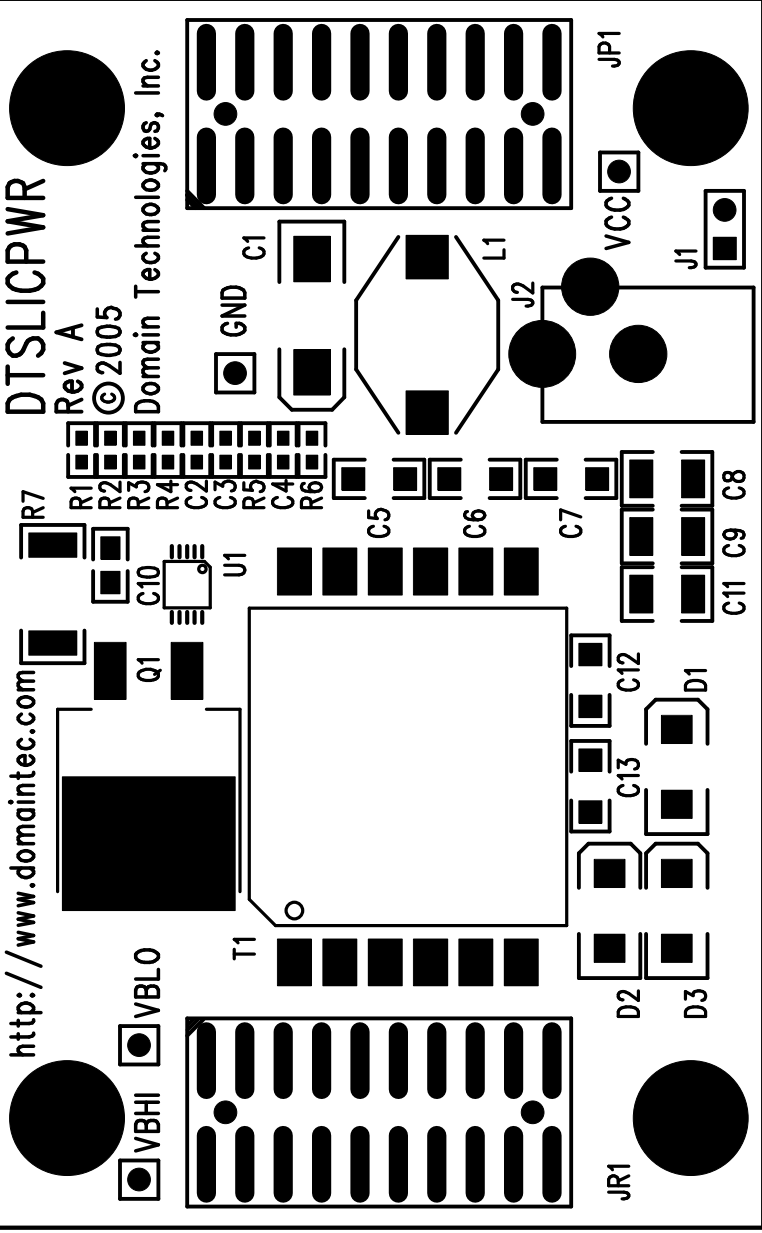
<http://www.domaintec.com>

DTSLICPWR

Rev A

©2005

Domain Technologies, Inc.



Domain Technologies, Inc.

<http://www.domaintec.com>