TEXAS INSTRUMENTS

TVP5150AEVM
User's Guide

User's Guide

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products & application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments

Post Office Box 655303 Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated

Read This First

About This Manual

This manual is intended to be a complete reference on how to use the TVP5150AEVM. The manual describes how to configure and use the EVM.

How to Use This Manual

This document contains the following chapters:

❑ Chapter 1 – Functional Description

❑ Chapter 2 – Setup and Configuration

❑ Chapter 3 – System Operation

❑ Chapter 4 – Troubleshooting the TVP5150AEVM

❑ Chatper 5 – TVP5150AEVM Schematics and Board Layout

FCC Warning

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

Contents

Figures

Tables

Functional Description

The TVP5150AEVM evaluation module is a printed circuit board designed for demonstration of the TVP5150A video decoder. The board is designed with a 120-pin connector which allows a connection to multiple backends. The EVM is shipped with a professional encoder module. The board is designed to provide ease of use, while allowing full evaluation of the video decoder.

Topic   **Page**

1.1 Description Overview

The TVP5150EVM uses the PC parallel port to emulate the I^{2}C bus, which provides communication with the TVP5150A video decoder and the video encoder. The WinVCC4 application software that communicates with the devices via the I^{2}C is provided on the EVM CD-ROM.

The analog video inputs supported by the TVP5150AEVM include composite video and S-video. More detail about the video inputs is discussed in the *Video Input Description* section. In general, the video decoder converts the analog video input signal into digital component data. This data and the associated clocks from the video decoder are sent to the video backend. The video backend then converts the digital data back into analog video. The analog video outputs supported by the EVM include: composite video, S-video, and component video. These are all output simultaneously.

To experiment with the programmable features of the TVP5150A video decoder and the video encoder, the parallel port of the TVP5150A is connected to the parallel port of a PC. WinVCC4 (Windows Video Control Center), a Windows™ compatible application provides the user interface for performing register-level and high-level control of the TVP5150A video decoder and the video encoder.

1.1.1 Board-Level Description

The TVP5150AEVM consists of the TVP5156AEVM module (see Figure 1–1) and the encoder EVM module (see Figure 1–2). A four row 120-pin connector connect the boards. The block diagram of the EVM set is shown in Figure 1–3.
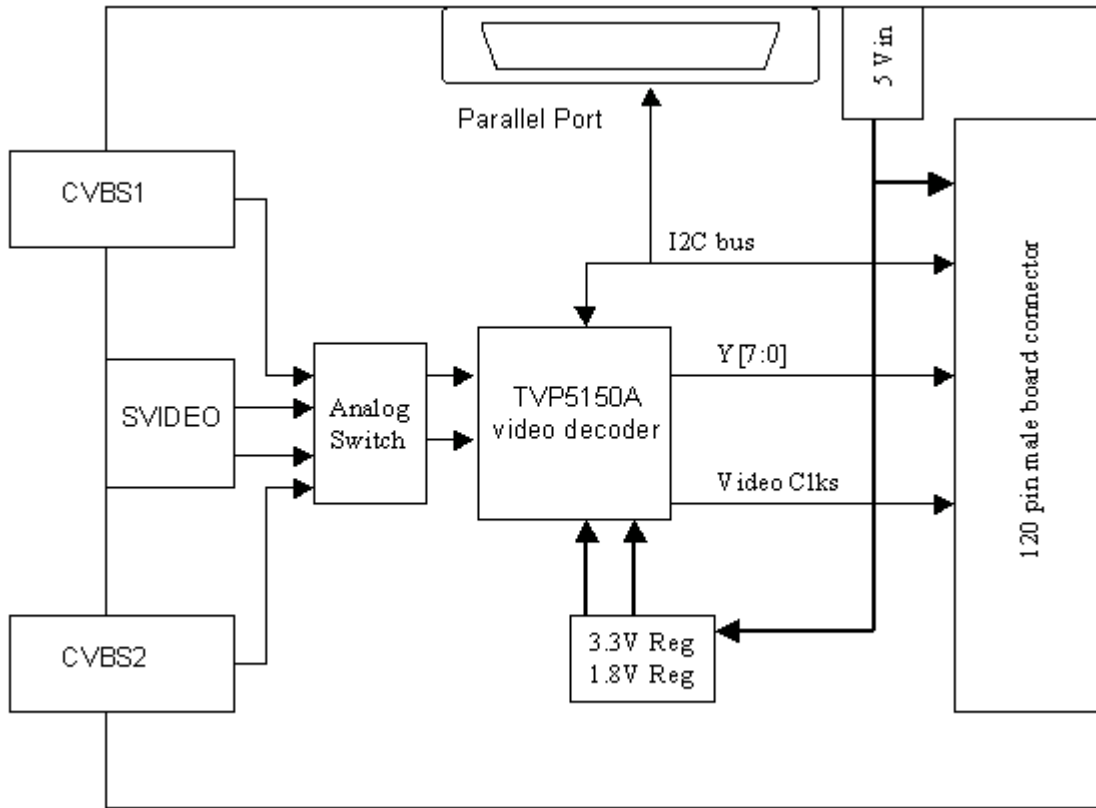
Figure 1−1. TVP5150A Module
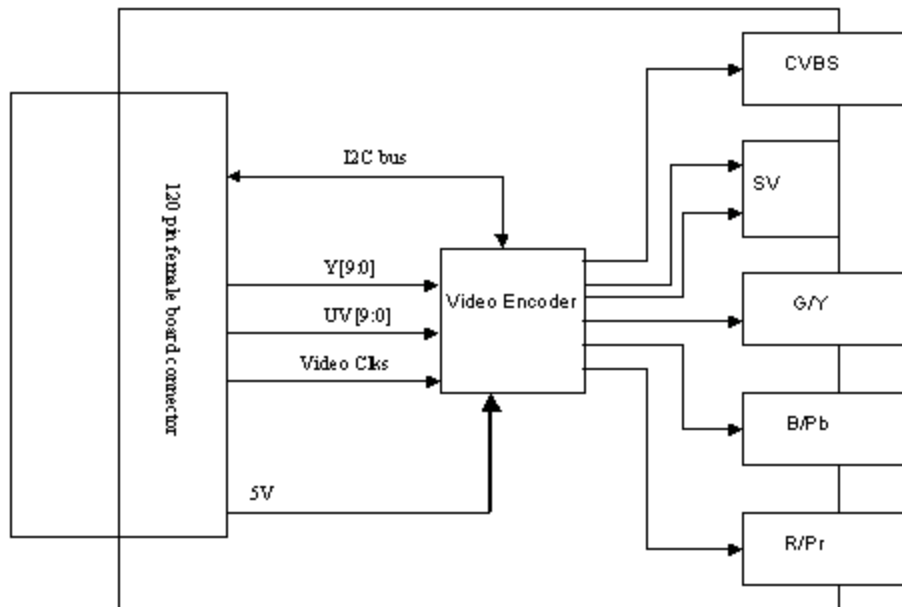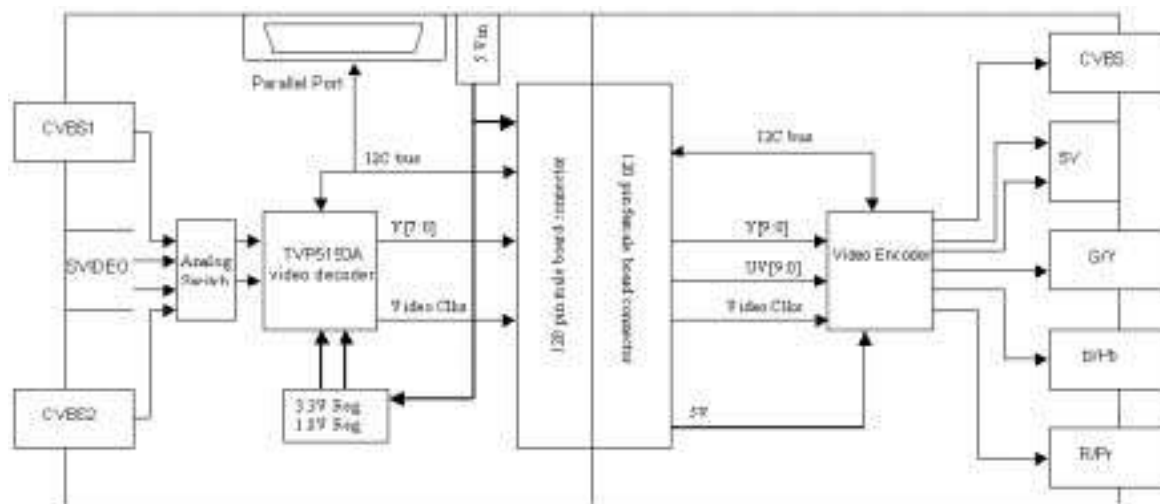

Figure 1−2. Encoder Module

Figure 1−3. TVP5150AEVM

1.1.2 Testpoints and Jumpers

The TVP5150AEVM was designed with test-points and jumpers to help in evaluation and troubleshooting. There are test-points for SCL, SDA, 3.3 V, and 1.8 V. All digital video data are brought out to a dual row header which allows easy hookup to test equipment. The I²C address selection is made with a shunt jumper, which is only read after a reset or at power up. The default address is 0xB8. If the address needs to be changed, the TVP5150A must receive a reset.

1.1.3 Decoder to Encoder Connection

The TVP5150AEVM uses a four row 120-pin connector to share common signals and the 5V power supply between the boards. This connection allows multiple backends to be connected to the TVP5150AEVM. The EVM package is shipped with an encoder module. This connector shares all digital video bits (Y[7:0]), all video clocks (VS, HS, GLCO/FID, and DATACLK), 5V, ground, I²C bus (SCL and SDA), and reset.

1.1.4 Video Input Description

The TVP5150AEVM decoder has an analog input channel that accepts two video inputs. The decoder supports a maximum input voltage range of 0.75 V, therefore, an attenuation of one-half is needed for most input signals with a peak-to-peak variation of 1.5 V. The parallel termination before the input to the device is 75 Ω. The two analog inputs can be connected as either two selectable composite video inputs, or one S-video input.

The TVP5150AEVM allows the user to have up to two composite inputs and one S-video input. This is achieved by using an analog video switch, which uses the GPICL pin to control the switch. The EVM software sets up and controls this pin.

The EVM has a resistor network on each of the inputs before it goes into the switch. These networks attenuate the signal and allow a 75-Ω resistor to ground to be placed after the anti-alias filter for termination.

Each input has an anti-alias filter that can be in-circuit or bypassed by jumpers (JP1, JP2, JP7–JP10). To select the filter, the shunts need to be positioned to short positions 1–3 and 2–4. To bypass the filter, the shunts need to be moved to short positions 1–2 and 3–4 as shown in Figure 1–4. The boards are shipped with the filter in circuit.

Figure 1–4. Filter Selection
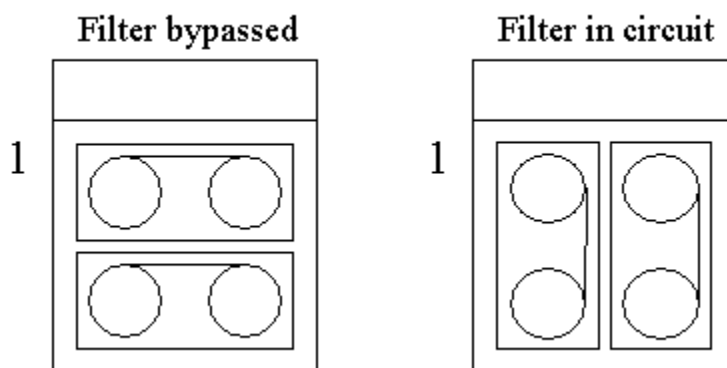
1.1.5 Video Output Description

The 8-bit digital video output of the TVP5150A is routed to the 120-pin edge connector along with all video clocks on the TVP5150A module. The encoder module connects to the 120-pin connector and is capable of receiving digital video with or with out embedded syncs. The analog outputs of the encoder module are composite, S-video, and component. All of these outputs come out of the encoder module simultaneously for the user's convenience.

1.1.6 System-Level Description

A system-level block diagram incorporating the TVP5150A is shown in Figure 1–5. Typical, commercially available test equipment is also shown. The primary features of this configuration are listed below.

❏ Power is provided by a single 5V power supply provided with the EVM and is shared between both modules via the 120-pin connector.

❏ Supported analog inputs include composite video and S-video.

❏ Re-encoded composite video and S-video are output via the encoder module.

❏ Component (YPbPr) video is output via the encoder module.

❏ I^{2}C bus initializes the video devices via a PC parallel port.

❏ The TVP5150A video decoder performance parameters may be measured with a video analyzer.

Figure 1–5. TVP5150AEVM System-Level Block Diagram

Chapter 2
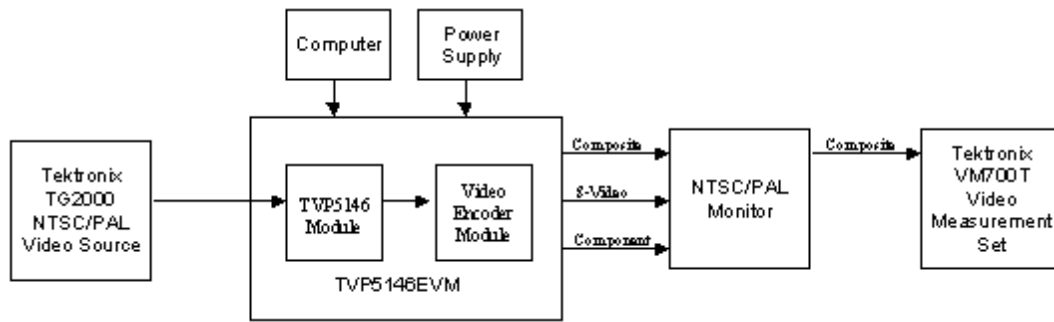
Setup and Configuration

The following summarizes the steps for setup and operation of the TVP5150AEVM. Follow the steps in the order shown.

❏ Software Installation (Section 2.1) or Updating the Software (Section 2.1.1)

❏ Hardware Configuration (Section 2.2)

2.1 Software Installation

All necessary software for the TVP5150AEVM is provided on the enclosed CD-ROM. Both the EVM software and the device files must be installed on the PC emulating the I²C bus via the parallel port.

1) Insert the CD-ROM into the computer that will emulate the I²C bus via the parallel port.

2) Run the SETUP.EXE file to install WinVCC4 and documentation.

3) Click *Next* at all prompts and finally click *Finish* to complete the installation process.

4) A message may appear instructing you to install the DriverLINX port I/O driver (if it was not previously installed). This must be installed for WinVCC4 to run. To install the driver, run PORT95NT.EXE, which is located in the root directory of the installation CD-ROM.

5) At the root directory of the installation CD is a zip file named Application_Reports.zip, which contains several application documents.

2.1.1 Software Update

Visit www.ti.com to obtain the latest version of the EVM software and the device initialization files.

2.2 Hardware Configuration

Figure 1–3 shows the TVP5150AEVM layout and indicates the location of the power supply and the appropriate connectors. All connectors are labeled according to their function. To prepare the EVM for evaluation, connect the following:

❑ TVP5150A module to encoder module

❑ Parallel port cable to a PC

❑ Analog video in

❑ Analog video out

❑ Power supply

The system comes with the anti-alias filters connected. To bypass the filters, you must rotate the appropriate jumpers (JP1, JP2, JP7–JP10) as described in section 1.1.5.

The I²C slave address can be selected with JP4. The default setting for this jumper is for the shunt to short pins 2–3, which selects 0xB8. This is connected to pin 37, which is read at power up. If you move the jumper to positions 1–2 and reset the board, the video decoder now responds to I²C slave address

0xBA. If you choose to change the address, you must exit, restart WinVCC4, and configure the software to use the new I^{2}C slave address.

Chapter 3

System Operation

This chapter describes the system operation.

3.1 Getting Started

After software installation (Section 2.1) and hardware configuration (Section 2.2) have been completed, the TVP5150AEVM may be powered up.

3.1.1 Starting the WinVCC4 Application Program

The *WinVCC4* (Windows™ Video Control Center) application program must have been previously installed on the PC. This is described in section 2.1. Run *WinVCC4* from the Windows™ Start Menu:

Start−>Programs−>TVP5150AEVM Software−>WinVCC4

In case the dialog shown in Figure 3−1 pops up, it means one of two things:

1) WinVCC4 did not run to completion the last time it ran.

In this case, click *OK* to exit the program and restart WinVCC4.

2) There is more than one instance of WinVCC4 running at the same time!

In this case, click *OK* to exit the program. Then, press *CTRL−ALT−DELETE* to bring up the *Windows Close Program* dialog. Select and click *End Task* on all occurrences of *Windows Video Control Center* (or *WinVCC4 Configuration*). Then, restart WinVCC4.

Figure 3−1. Multiple Instances Running Message

WinVCC4

If the previous run of WinVCC4 was terminated abnormally, click OK and start WinVCC4 again. Otherwise, more than one instance of WinVCC4 may be running! Click OK to close this program. Then, press Ctl-Alt-Delete. Highlight and click "End Task" to terminate all instances of Windows Video Control Center (or WINVCC4 Configuration).

OK
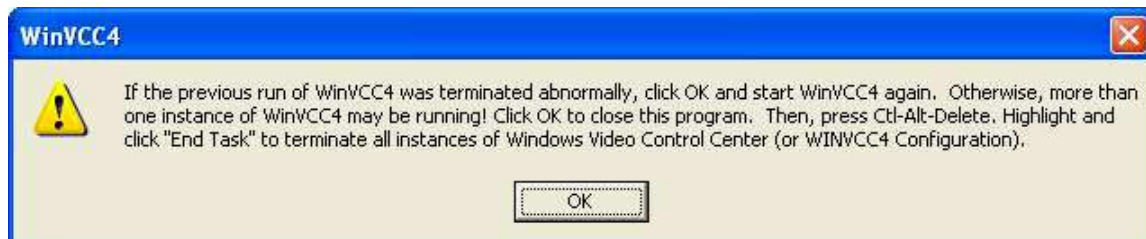
Figure 3–2. WinVCC4 Configuration
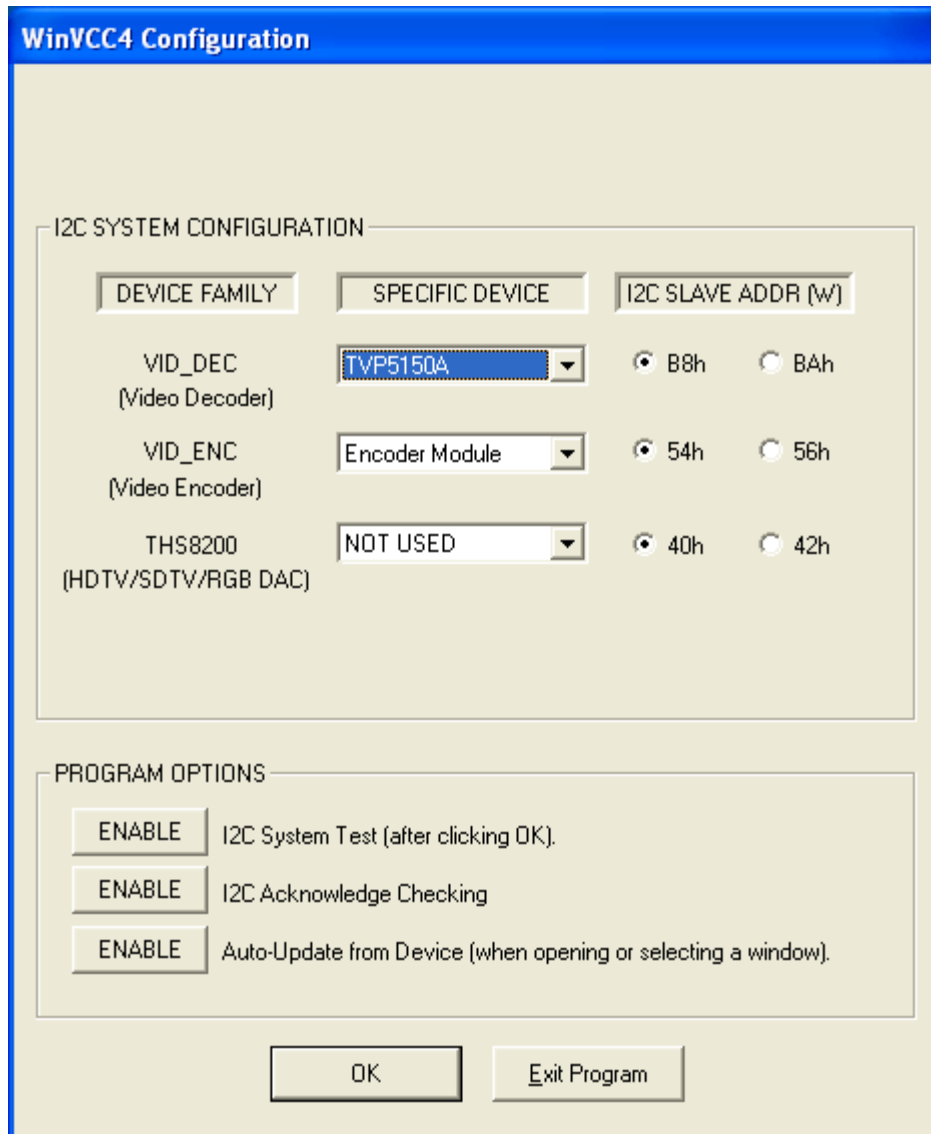
3.1.2 WinVCC4 Configuration Dialog Box

The WinVCC4 Configuration dialog box, as shown in Figure 3–2, should now be visible. All settings from this dialog box are stored in the Windows registry and are restored the next time the program is started. After initial installation, VID_DEC will be set to TVP5146 and VID_ENC will be set to encoder module.

The I^2C slave addresses for each device must match the I^2C slave address selected by the jumper on the TVP5150AEVM. This jumper is set by the factory to use 0xB8 for the TVP5150A video decoder.

It is also important to select the correct *Specific Device* type for the video decoder. TVP5150A and encoder module must be selected for this EVM.

The program options should normally all be set to ENABLE. Disabling these options should only be required if you are debugging a problem with the I^2C bus itself.

Clicking *OK* begins I^{2}C communication with the selected devices.

3.1.3 I^{2}C System Test

The I^{2}C system test of selected registers runs immediately after closing the WinVCC4 Configuration dialog box with *OK* (unless the *I^{2}C System Test* program options button was disabled).

If the I^{2}C system test passes, no message appears. If the test failed, a dialog box like Figure 3–3 appears. See Section 4.1 on resolving I^{2}C communication problems.

The I^{2}C system test can be run at anytime by clicking *Run System I^{2}C Test* in the *Tools Menu*.

Figure 3–3. I^{2}C System Test Report
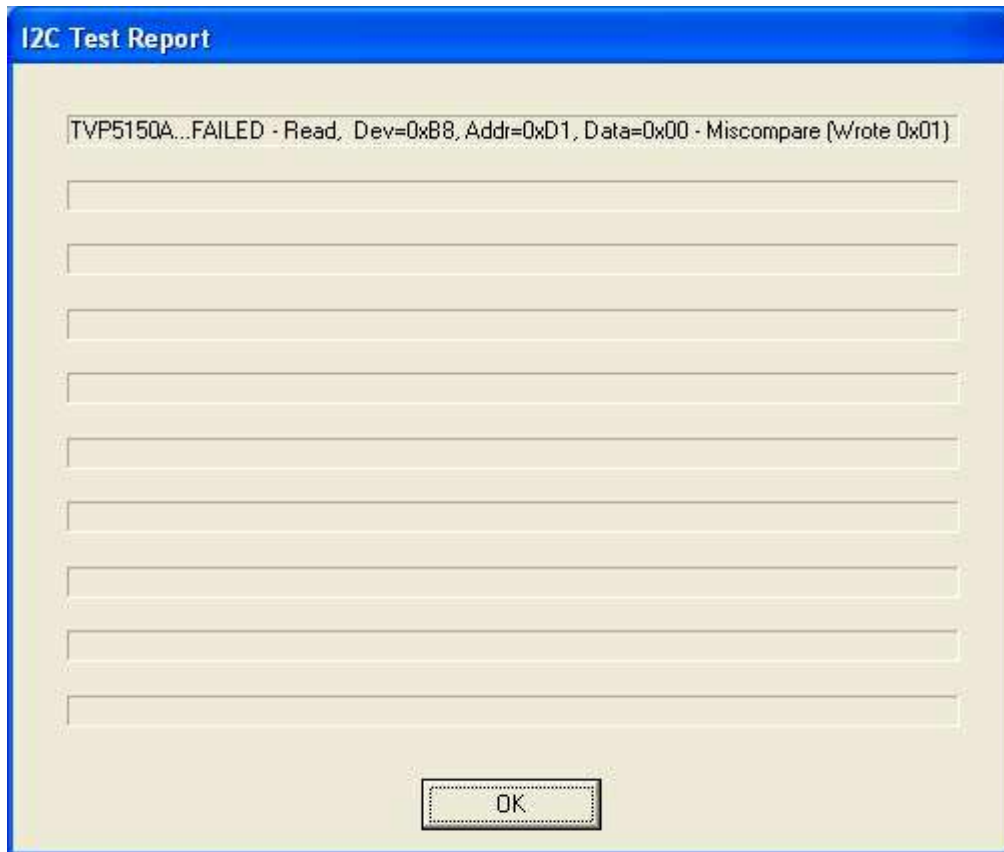
3.1.4 Real-Time Polling

Real-time polling provides polling functions that execute in the background continuously, when enabled via the Real-Time Polling dialog. There are two polling functions. The function that applies to the TVP5150A is VIDEO-STANDARD AUTO-SWITCH POLLING.

When TVP5150A detects a change in the input video standard, it automatically switches to operation in the detected standard (which includes all necessary I^2C register initialization) for proper decoding of the input video. To enable auto-switch on the TVP5150A, the *Set Video Standard* register must be set to auto-switch mode (Reg 0x28 = 0x00).

If the WinVCC4 auto-switch polling function is enabled, the detected video standard status from the TVP5150A is polled until a change in the input video standard (or in the TVP5150A sampling mode) is detected. When a change is detected, the video encoder is reprogrammed as needed for the detected standard. Using this feature, the video source can change its video standard and the system will display using the new standard without user intervention.

To enable auto-switch polling (recommended), the video-standard auto-switch polling function must be enabled in the real-time polling dialog shown in Figure 3–4. The real-time polling dialog can also be accessed once WinVCC4 is up and running by clicking *Real-Time Polling* in the *Tools* menu.

Figure 3–4. Real-Time Polling Dialog
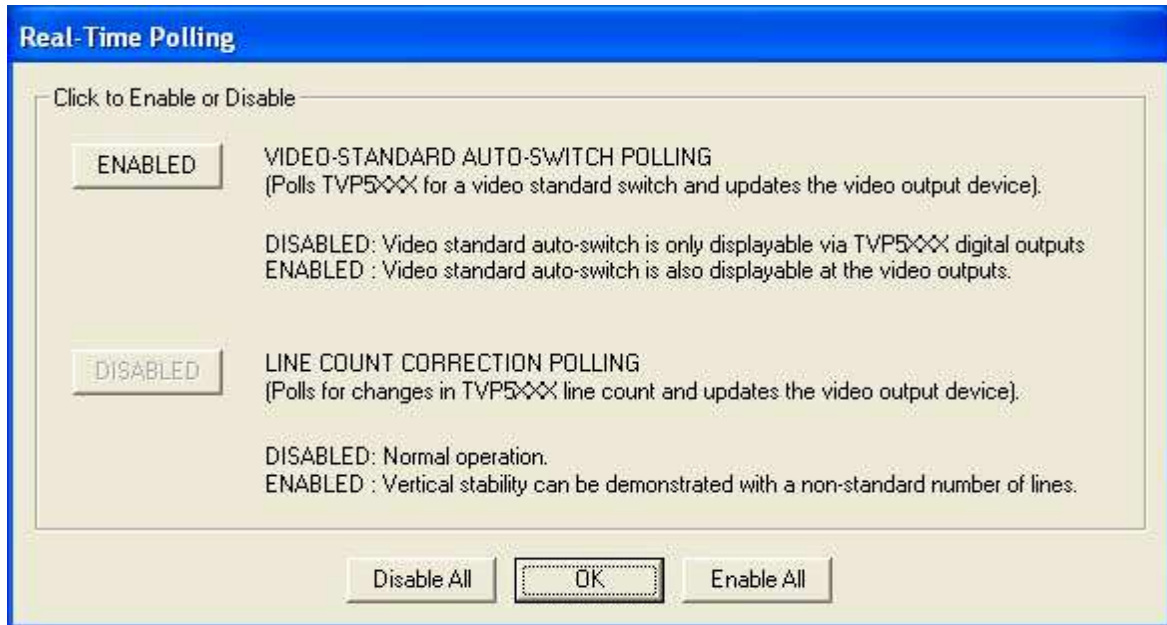
3.2   Main Menu

After closing the real-time polling dialog, the Main Menu is displayed as shown in Figure 3–5. The menus, which are used to operate WinVCC4, are *File*, *Edit*, *Tools*, *Window* and *Help*. The *File* menu's only function is *Exit*, which terminates the program. Table 3–1 is a summary of the main menu contents.

Figure 3−5. Main Menu

Table 3−1. Main Menu Summary

Menu	Contents
File	Exit WinVCC4
Edit	Register Map Editor
	Generic I^{2}C Editor
	Property Sheets
Tools	System Initialization
	Real-time Polling
	Multiple-Byte I^{2}C Transfers
	Set I^{2}C Bit Rate
	Run System I^{2}C Test
	Run Continuous I^{2}C Test
	Read VBI FIFO
Window	Allows selection of the active window. Multiple windows can be open at the same time. Initially the Window menu is empty.
Help	Displays program version

3.2.1   System Initialization

Clicking *System Initialization* in the *Tools* menu displays the dialog shown in Figure 3–6. This provides the means for initializing the video decoder and/or video encoder for a particular video mode. The details of the initialization are contained in the command file (with a CMD file extension).

The command file is opened using the *Browse…* button. Once the command file is opened, a text list displays descriptions of the individual "data sets" contained within the command file.

Click once on the desired data set description to select it. Click the *Program Device(s) Using Selected Dataset* button to run the selected data set, which loads the devices via the I^{2}C bus. When the device initialization has completed, the status indicator reads *Ready*.

NOTE: If Ready does not display, then the devices are not initialized and the I^{2}C bus is not communicating. See section 4.1 *Troubleshooting the TVP5150AEVM* for a solution.

Click the *OK* button to close the dialog box. Each time the system initialization dialog is closed, the initialization file pathname and the data set selection

number are saved in the Windows™ registry to allow these settings to be retained for the next time *WinVCC4* runs.

Figure 3−6. System Initialization

3.2.2 Adding a Custom Data Set

Program the EVM via the system initialization tool using the factory-supplied command file. Through the property sheets tool (see section 3.2.5), you can customize the device register settings to suit your needs. To save your custom settings:

1) Reopen the *System Initialization* dialog via the *Tools* menu.

2) Click the *Append Current Device Settings to Command File* button. A dialog requesting a description of the new data set appears.

3) Optionally, click the drop-down box and select one of the existing descriptions.

4) Modify the description text or type your own description.

5) Click *OK*. All nondefault register values are read from the decoder and encoder is appended to the current command file as an additional data set.

Now, you can select your custom data set and send it with a press of the *Program…* button.

NOTE: After editing the command file (.CMD) using a standard editor, the file must be saved as plain text.

3.2.3 Command Files

The command file is a text file which can be generated using any common editor; however, it must be saved as plain text. Command files are especially useful for testing and comparing various changes in the device setup.

A default command file, which should contain most of the desired setups, has been installed with the EVM software. After installation, when the system initialization tool is opened, the default command file is selected. This command file is located at:

C:\Program Files\Texas Instruments\WinVCC4\TVP5150A\Initialization\Initialize.cmd

A command file can contain up to 250 data sets. A data set is a set of register settings to initialize the video decoder and/or video encoder for a particular video mode. Each data set includes a description that is displayed in one row of the dataset descriptions list. The register settings may be located in the command file itself and/or may be stored in separate include file(s) (with an INC file extension) and be included into the command file using the INCLUDE statement.

3.2.3.1 Example Command File

An example of one data set within a command file is shown in Figure 3−7. Each command file may contain individual *write to register* (WR_REG) commands or include these commands from a separate INC file or both. The purpose of the *INCLUDE* feature is to avoid repeating a long list of register settings many times in a command file and having to maintain that list.

Figure 3−7. Example CMD File With One Data Set

```
BEGIN_DATASET
DATASET_NAME,"TVP5150A, Auto-switch, CVBS CH4A, 656 Out"
// Initialize video encoder using an include file
INCLUDE, EncoderNTSC656_RTC.INC
// Program TVP5150A registers
WR_REG,VID_DEC,0x01,0x00,0x0C // Input: CVBS CH4
WR_REG,VID_DEC,0x01,0x02,0x00 // Auto-switch mode
WR_REG,VID_DEC,0x01,0x04,0x3F // Auto-switch MASK
WR_REG,VID_DEC,0x01,0x06,0x40 // Pedestal off
WR_REG,VID_DEC,0x01,0x31,0x05 // RTC mode
WR_REG,VID_DEC,0x01,0x33,0x40 // ITU 601 (extended range)
WR_REG,VID_DEC,0x01,0x34,0x11 // OUTPUTS ACTIVE
WR_REG,VID_DEC,0x01,0x35,0xFA // GLCO and FID ACTIVE
WR_REG,VID_DEC,0x01,0x36,0xAF // HS and VS ACTIVE
END_DATASET
```

3.2.3.2 Command File Syntax

1) The comment indicator is the double-slash "//".

2) The command file is not case-sensitive and ignores all *white-space* characters.

3) All numbers can be entered as hexidecimal (beginning with 0x) or as decimal.

4) Every data set in a command file begins with BEGIN_DATASET and ends with END_DATASET. The maximum number of data sets is 250.

5) The data set text description is entered between double quotes using the DATASET_NAME command. The enclosed text can be up to 128 characters in length. This text appears in the system initialization dialog when the command file is opened.

6) The INCLUDE command inserts the contents of an include file (with an INC file extension) in-line in place of the INCLUDE command. Therefore, the include file should not contain the BEGIN_DATASET, END_DATASET, and DATASET_NAME commands.

NOTE: All included files must be located in the same directory as the command (CMD) file.

7) The *write to register* command is written as follows:

WR_REG, <DeviceFamily>, <Number of data bytes (N)>, <sub-address>, <Data1>,…,<DataN>

or

WR_REG,<Literal slave address>,<Number of data bytes (N)>,<sub–address>, <Data1>,…,<DataN>

The valid device family mnemonics are:

❏ VID_DEC for the video decoders

❏ VID_ENC for the video encoding

❏ THS8200 for the THS8200 device

WinVCC4 translates the device family mnemonic to the slave address that was selected on the WinVCC4 configuration dialog upon program startup. This eliminates having to edit command files if the alternate slave address must be used.

If the literal slave address method is used, the slave address entered is used directly. This method is normally used for programming the video encoder.

8) A delay may be inserted between commands using the WAIT command, which is written as follows:

WAIT,<# milliseconds>

3.2.4 Register Editing

The next sections describe the two available modes of register editing: register map editors and property sheets. Each of these functions can be selected from the *Edit* menu.

3.2.4.1 Register Map Editor

The register map editor, as shown in Figure 3–8, allows the display and editing of the entire used register space of the device within a simple scrolling text box. To open this, click on *Edit Register Map* in the *Edit* menu and click on the device type to edit. Table 3–2 describes how to use each of the controls in the register map editor.

Figure 3–8. Register Map Editor
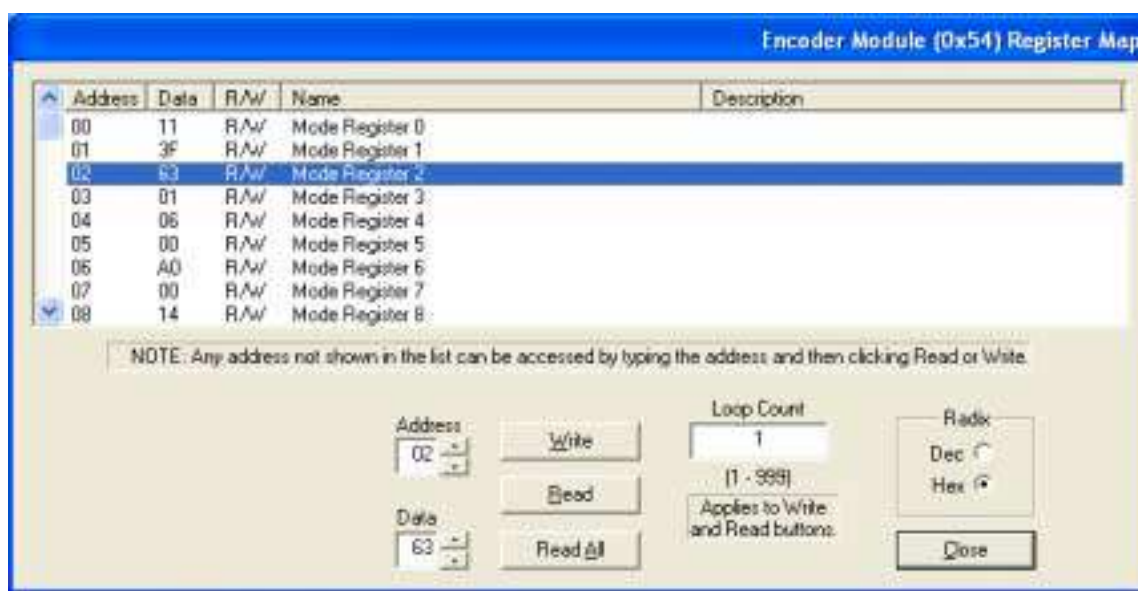
Table 3–2. Register Map Editor Controls

Control	Definition
Register Window	Scrolling text box that displays the address and data for the I^2C registers that are defined for the device.
Address Edit Box	This contains the I^2C sub-address that is accessed using the *Write* and *Read* buttons. Clicking on a row selects an address, which then appears in the address edit box.
	NOTE: After clicking on a row, the *Data Edit Box* contains the data that was in the register window. This data may not correspond to the actual data in the device (until *Read* or *Read All* is clicked).
	The address up/down arrows are used to jump to the next/previous sub-address that is defined for the device.
	NOTE: If an address is not defined for the device, it can still be accessed by typing the sub-address in the *Address Edit Box*.
Data Edit Box	This contains the data which will be written to or was read from the I^2C sub-address.
	The data up/down arrows increases / decreases the data value by 1.

Write Button	Writes the byte in the *Data Edit Box* to the address in the *Address Edit* box. The I^{2}C register is written to whether or not the data is different from the last time the register was read.
Read Button	Reads the data from the address in the *Address Edit Box* into the *Data Edit* box and the register window.
Read All Button	Reads all defined readable registers from the device and updates the register window.
Hex Button	Displays all values in the register window and address and data edit boxes in hexadecimal format.
Dec Button	Displays all values in the register window and address and data edit boxes in decimal format.
Close Button	Closes the dialog. **NOTE:** Multiple edit register map windows can be open at the same time (one for each device). Use the Window menu to navigate.
Loop Count	Causes subsequent write or read operations to be performed N times. N is entered as a decimal number from 1 to 999.
Edit Indirect Registers	Opens the indirect register editor of the TVP5150A.

3.2.4.2 Encoder Module Editor

The encoder module editor, as shown in Figure 3−9, allows the display and editing of the video encoder registers. This editor works like the register map editor.

To open this, click on *Edit Register Map* in the *Edit* menu and click on *Encoder Module*.

Figure 3−9. Encoder Module Editor
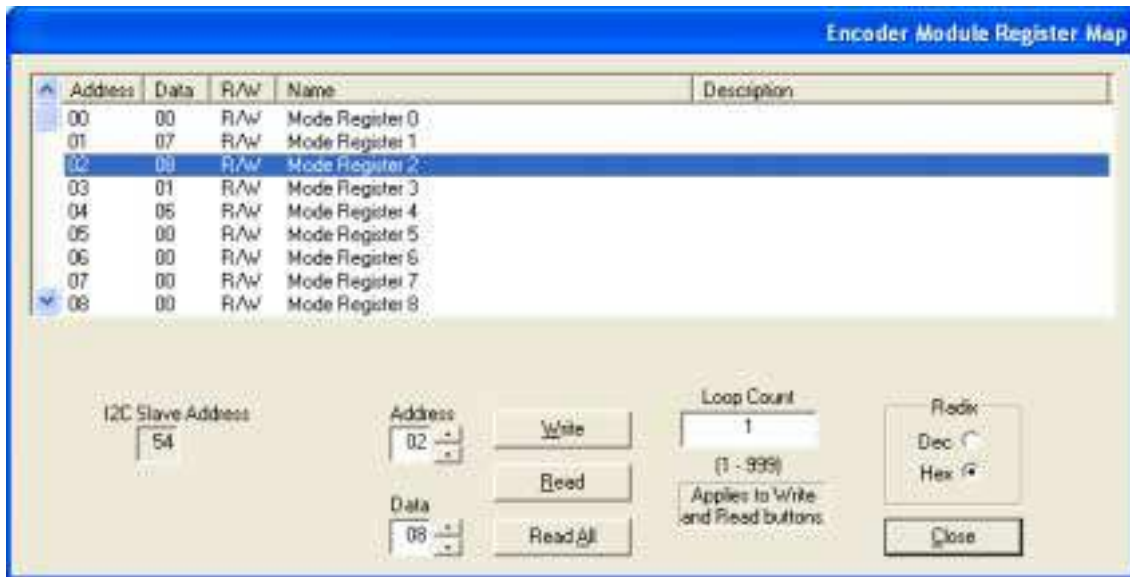
3.2.4.3 Generic I^{2}C Register Editor

The *Generic I^{2}C Editor*, as shown in Figure 3−10, allows the display and editing of any device on the I^{2}C bus. This editor works like the *Register Map Editor*, except that the I^{2}C slave address must be entered and the *Read All* button is disabled.

To open this, click on *Edit Register Map* in the *Edit* menu and then click on *Generic I²C*.

The video encoder module registers can be edited using I²C sub-address 0x54 (default) or 0x56 if the alternate slave address is being used.

Figure 3–10. Generic I²C Register Editor
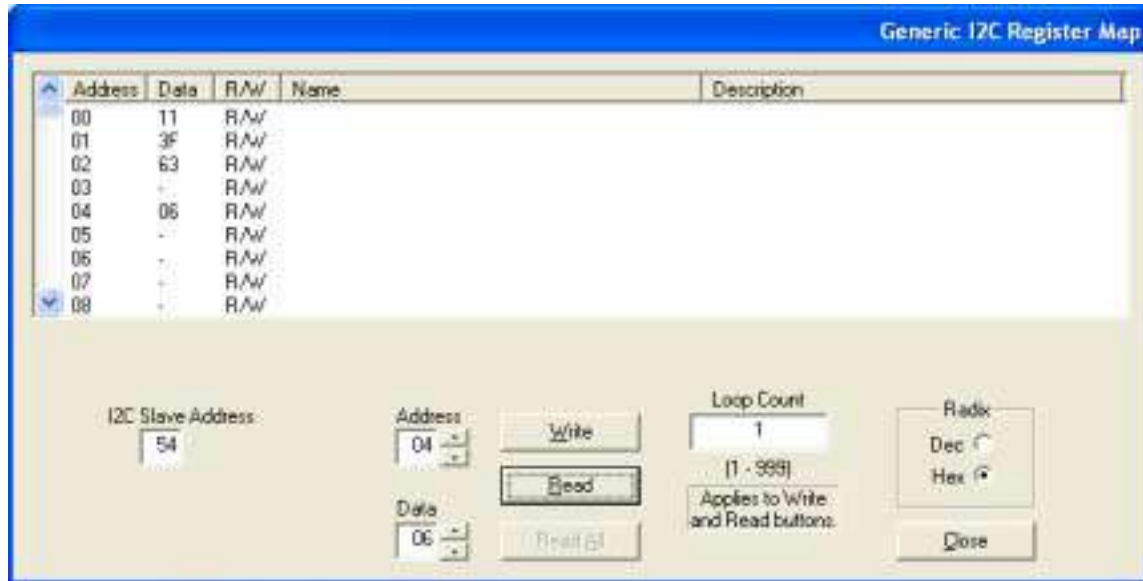
3.2.4.4 Indirect Register Editor

The indirect register editor, as shown in Figure 3–11, allows the display and editing of the indirect registers (or hardware registers) of the device.

To open this, first open the *Register Map Editor* in the *Edit* menu for the TVP5150A. Then, click the *Edit Indirect Registers* button. The operation of the indirect register editor controls are explained in Table 3–3.

Figure 3−11. Indirect Register Editor
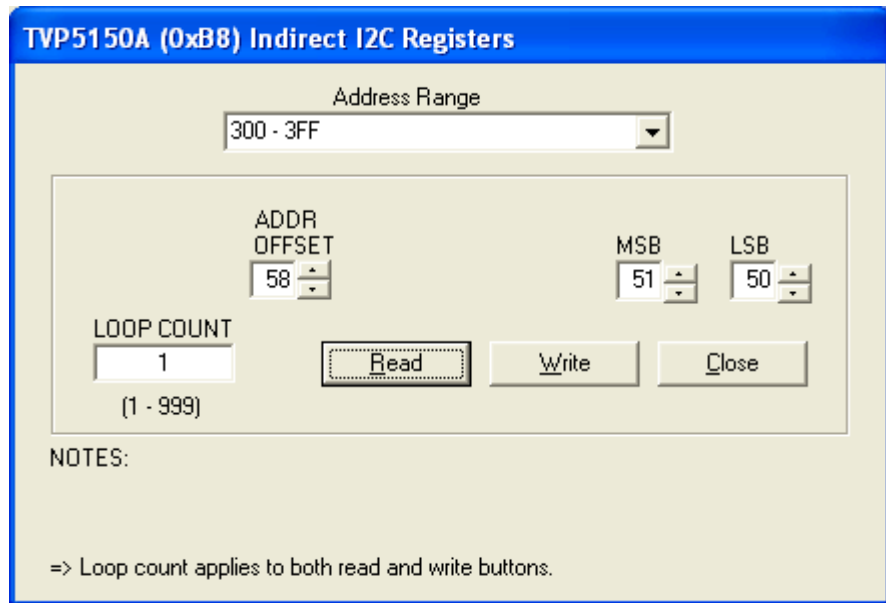
Table 3−3. Indirect Register Editor Controls

Control	Definition
Base Address Selector	The indirect registers use a 10-bit address internally. The base address selector allows quick entry of the base address. The list contains base addresses for the major functional blocks of the TVP5150A.
Address Offset Edit Box	This contains the lowest byte of the 10-bit internal address. The full 10-bit address is formed by adding the base address to the address offset. The address up/down arrows increment/decrement the address offset by 1.
Data Edit Boxes	This contains the 16-bit data word that is written to or read from the indirect address. The LSB data is at the lowest address. The data is written/read LSB first. The data up/down arrows increment/decrement the data value by 1.
Write Button	Writes the (2) bytes in the data edit boxes starting at the 10-bit indirect address BASE+OFFSET.
Read Button	Reads (2) consecutive data bytes starting at the 10-bit indirect address BASE+OFFSET to the data edit boxes.
Loop Count Edit Box	Causes subsequent write or read operations to be performed N times. N is entered as a decimal number from 1 to 999.
Close Button	Closes the dialog. **NOTE:** The indirect register editor can remain open with other windows. Use the Window menu to navigate.

3.2.5 Property Sheets

The property sheets represent the register data in a user-friendly format. The data is organized by function, with each function having its own page and being selectable via tabs at the top. See Figure 3−12.

To open this, click on *Edit Property Sheets* in the *Edit* menu and select the device type to edit.

When the property sheet function is started, or whenever you tab to a different page, all readable registers in the device are read from hardware to initialize

the dialog pages. Values on the page are changed by manipulating the various dialog controls as described in Table 3–4.

There are *OK*, *Cancel*, and *Apply* buttons at the bottom of each property page. These are explained in Table 3–5.

Figure 3–12. Property Sheets
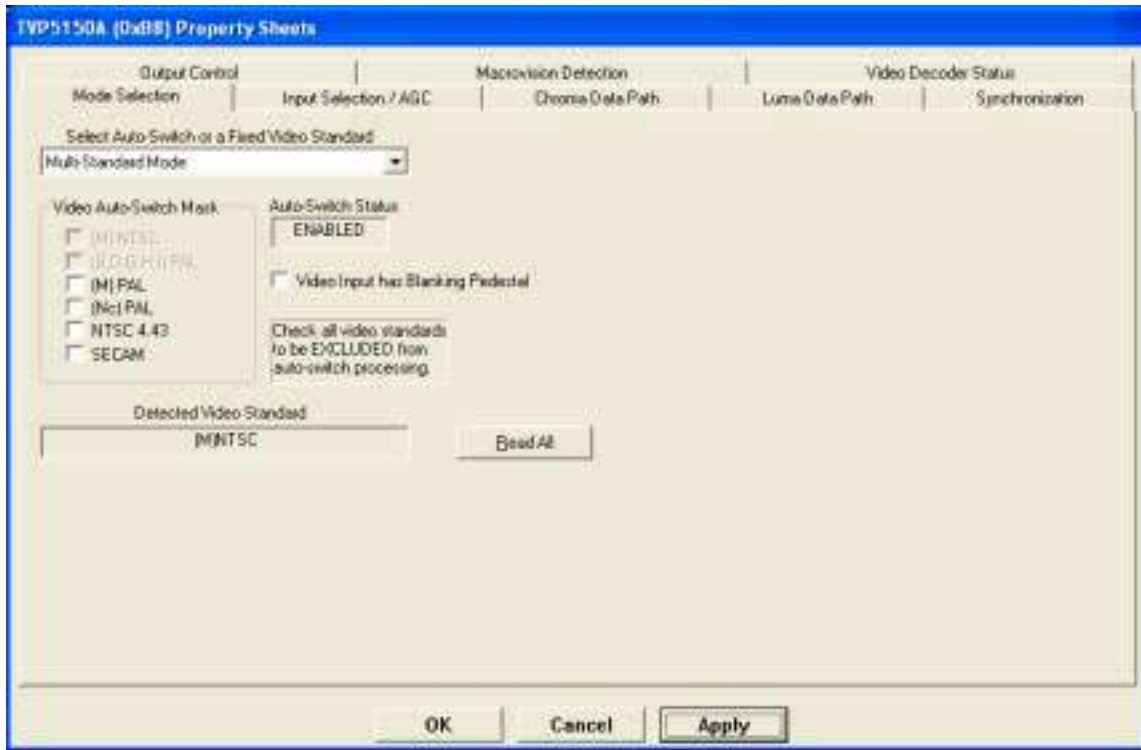
Table 3–4. Use of Property Sheet Controls

Dialog Control	What Do I Do With It?	When is Hardware Updated?
Read-only edit box	Read status information	N/A
Check box	Toggle a single bit	After *Apply*
Drop-down list	Select from a text list	After *Apply*
Edit box	Type a number	After *Apply*
Edit box with up/ down arrows	Use up/down arrows or type a number	Up/Down arrows: Immediately Type a number: After *Apply*
Slider	Slide a lever	Immediately
Push button	Initiate an action	Immediately

Table 3–5. Property Sheet Button Controls

Button Control	Definition
OK	Writes to all writeable registers whose data has changed. A register is flagged as changed if the value to be written is different from the value last read from that address. Closes the dialog.
Cancel	Causes all changes made to the property page since the last *Apply* to be discarded. Changes made to dialog controls with *immediate hardware update* are not discarded, since they have already been changed in hardware. Does not write to hardware. Closes the dialog.
Apply	Writes to all writeable registers whose data has changed. A register is flagged as changed if the value to be written is different from the value last read from that address.

3.2.5.1 Property Sheet Refresh

The property sheets were designed so that the data displayed is always current. Certain actions cause the entire register map to be read from the device and to update the property sheets. This happens when:

1) Property sheets are initially opened.

2) When tabbing from one page to another.

3) When *Read All* is clicked.

4) When making *the Property Sheets* window the active window (by clicking on it).

5) When making a register map editor window the active window.

3.2.5.2 Auto-Update from Device

Items 4 and 5 above are referred to as the auto-update feature. Auto-update can be disabled by setting its program option button to DISABLED. This button is located on the initial dialog box (WinVCC4 Configuration).

With auto-update enabled (default), when you open both the property sheets and the register map editor at the same time, changes made to the property sheets (and applied) are updated in the register map window as soon as the register map window is clicked on. It also works the other way; changes made in the register map editor are updated in the property sheets as soon as the property sheets window is clicked on.

Troubleshooting the TVP5150AEVM

This chapter discusses ways to troubleshoot the TVP5150AEVM.

Topic **Page**

4.1 Troubleshooting Guide

If you are experiencing problems with the TVP5150AEVM hardware or the WinVCC4 software, see Table 4−1 for available solutions.

Table 4−1. Troubleshooting Guide

Symptom	Cause	Solution
At startup the error message *Cannot find DLL file DLPORTIO.DLL* appears.	The parallel port driver has not been installed.	Run Port95NT.exe on the CD-ROM to install the driver
Blank screen	Wrong video input is selected.	Go to Edit−>Property Sheets−>TVP5150A, input selection/AGC page, select the correct video input(s) and click *Apply*. (The composite video input 4A is selected by default.)
	Source is connected to the wrong input connector.	Connect source to the correct input connector.
	YCbCr outputs or clock output is disabled.	Go to Edit−>Property Sheets−>TVP5150A, output page, check the enable YCbCr outputs and enable clock outputs check boxes and click *Apply*.
No color	Wrong mode selected for color subcarrier genlock output.	Go to Edit−>Property Sheets−>TVP5150A,synchronization page, set the Fsc control format to RTC and click *Apply*.
	GLCO pin is not set to output the GLCO signal.	Go to Edit−>Property Sheets−>TVP5150A, output page, set the drop down box labeled "FID/GLCO (pin 23)" to genlock output (GLCO) and click *Apply*.
Screen colors are only magenta and green.	Wrong YCbCr output format.	Go to Edit−>Property Sheets−>TVP5150A, output page, set the YCbCr output format to 8-bit 4:2:2 YCbCr w/ITU−R BT.656 embedded sync mode and click *Apply*.
Video standard auto-switch does not work on the video decoder side.	Auto-switch masks are not set correctly.	Go to Edit−>Property Sheets−>TVP5150A, mode selection page, **uncheck** all standards to be included in auto-switch processing and click *Apply*.
	Video decoder is not in auto-switch mode.	Go to Edit−>Property Sheets−>TVP5150A, mode selection page, set the drop-down box to multi-standard and click *Apply*.
Video standard auto-switch does not work on the video encoder side.	Auto-switch polling is not enabled.	Click real-time polling in the Tools menu. Click *Enable All* and *OK*.

Table 4−1. Troubleshooting Guide (Continued)

Symptom	Cause	Solution
No I2C communication	Decoder I2C slave address is wrong.	Make sure I2C slave address jumper on the TVP5150A decoder module is across pins 2 and 3.
	Encoder I2C slave address is wrong.	Slave address is hard coded to be 0x54 in the command file. Make sure the I2C slave address jumper on encoder module is across pins 2 and 3.
	Parallel cable is not connected from PC parallel port to the TVP5150A decoder module DB25 connector.	Connect cable
	EVM is not powered on.	Power supply must be plugged into a 100 V – 240 V/47 Hz – 63-Hz power source and the cord must be plugged into the power connector on the EVM.
	Wrong type of parallel cable.	Some parallel cables are not wired straight through pin-for-pin. Use the cable supplied with the EVM.
	Device was placed in power-down mode.	Press the *Reset* button on the TVP5150A Decoder Module.
	EVM was configured for an external I2C master.	Reinstall 0-Ω resistors R5 and R6. Control EVM using the PC parallel port.
	PC parallel port mode is not set correctly.	DO THIS AS A LAST RESORT. Reboot PC, enter BIOS setup program, and set parallel port LPT1 mode (Addr 378h) to ECP mode or bidirectional mode (sometimes called PS/2 mode or byte mode). If already set to one of these two modes, switch to the other setting. See Section 4.2.2.
	Still no I2C communication	PC may not be capable of operating in the required parallel port mode. This is true of some laptop computers. Use a different computer, preferably a desktop PC.

4.2 Resolving I2C Communication Problems

When WinVCC4 is started and the WinVCC4 Configuration dialog box is closed with *OK*, the I2C system test is performed (unless the I2C system test program options button was disabled).

If the I2C system test fails, a dialog box like Figure 4−1 appears. This example reports that a read from TVP5150A failed, using slave address 0xB8, sub-address 0xD1. The data read was 0x00. The expected data was 0x01.

After noting which device had a problem, click *OK* to continue. Next, the *corrective action dialogs* appears to help fix the problem.

Figure 4–1. I²C System Test Report
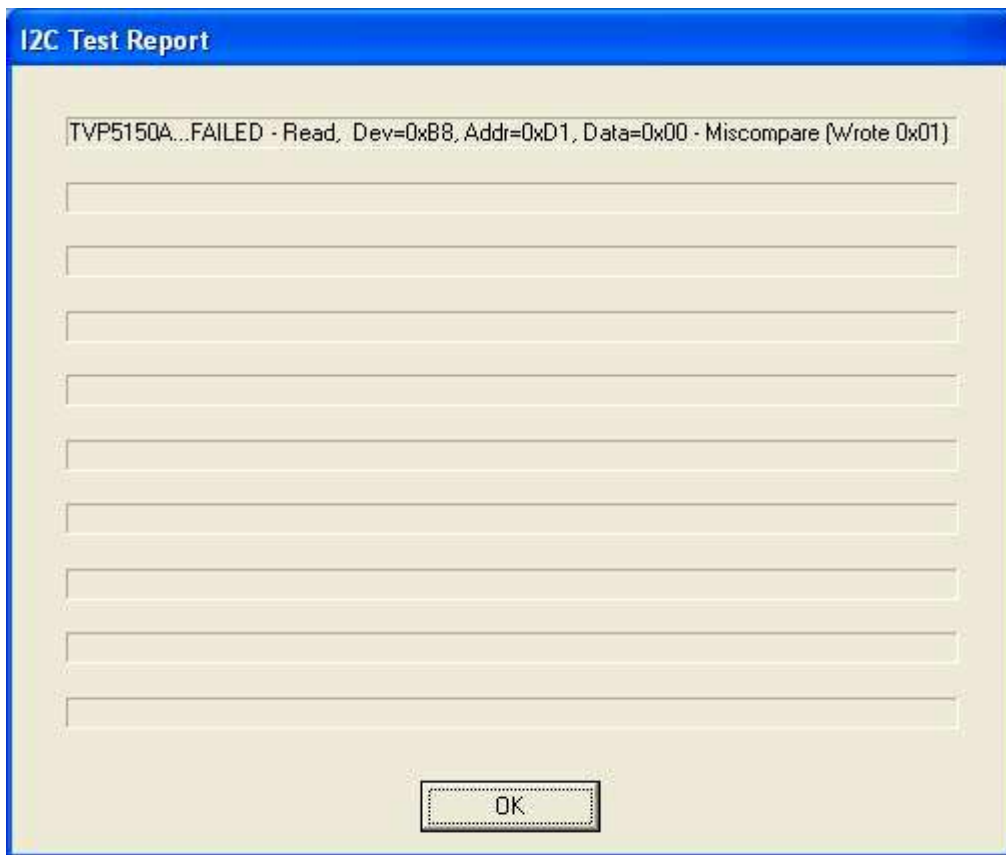
4.2.1 Corrective Action Dialogs

After closing the I²C system test report dialog box, the dialog in Figure 4–2 appears.

1) If the cable is NOT connected from the PC parallel port to the TVP5150AEVM or if the EVM power is not on:

   a) Click *NO*.

   b) The dialog shown in Figure 4–3 appears instructing you to correct the problem.

   c) Correct the problem.

   d) Click *OK* to continue. The *Real-Time Polling* dialog should appear. See Figure 3–4 in section 3.1.4.

2) If the cable IS connected from the PC parallel port to the TVP5150AEVM AND the EVM power IS on:

   a) Click *Yes*.

   b) The dialog shown in Figure 4–4 appears. This dialog appears if the PC parallel port mode setting MAY need to be changed.

   NOTE: Only run the PC BIOS setup program if the I²C communication problem CANNOT BE RESOLVED IN ANOTHER WAY (correct slave address settings, reset or power cycle the EVM and/or check that the device type selected was TVP5150A).

   Click *OK* to continue.

     c) The real-time polling dialog appears. Click *OK* to close it and get to the main menu.

     d) Click *Exit* in the *File* menu to exit the program.

Figure 4–2. Cable Connected / Power On Message
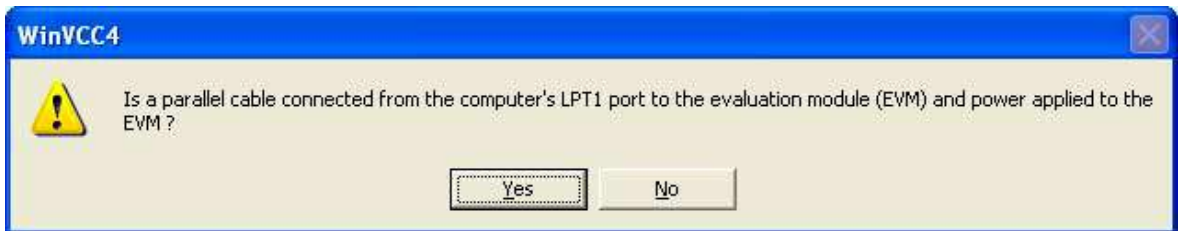
Figure 4–3. Connect Cable / Turn On Power Message

4.2.2 Setting the PC's Parallel Port Mode

NOTE: Only run the PC BIOS setup program if the I^{2}C communication problem CANNOT BE RESOLVED IN ANOTHER WAY (correct slave address settings, reset or power cycle the EVM and/or check that the device type selected was TVP5150A).

1) Restart the PC.

2) During the boot process, enter the BIOS setup program by pressing the required key (usually the initial text screen tells you which key to press).

3) Find the place where the parallel port settings are made.

4) Set the parallel port LPT1 at address 378h to DCP mode or bidirectional mode (sometimes called PS/2 mode or byte mode). If one of these two modes is already selected, change to the opposite mode.

5) Exit and save changes.

Figure 4−4. Parallel Port Mode Message
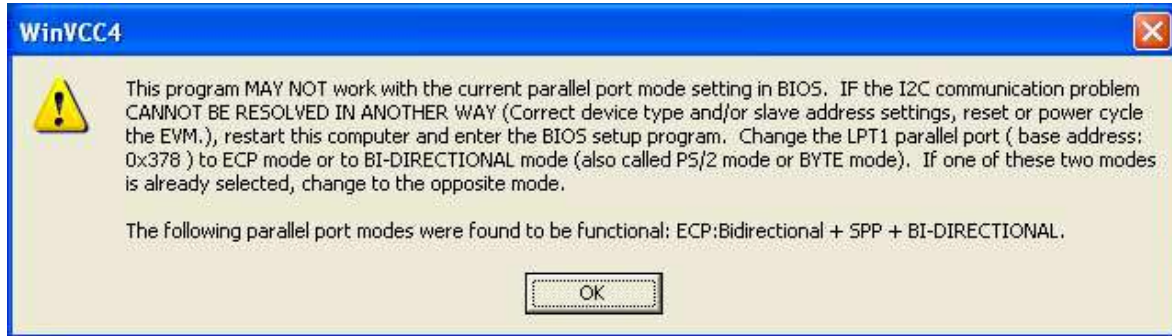
4.2.3 General I^{2}C Error Report

The error report shown in Figure 4−5 appears when an I^{2}C error occurs at any time other than after the I^{2}C system test. In this example, there was an acknowledge error at slave address 0x54 (the video encoder module). The error occurred on r*ead cycle phase 1* on the device address (slave address) byte.

Figure 4−5. General I^{2}C Error Report
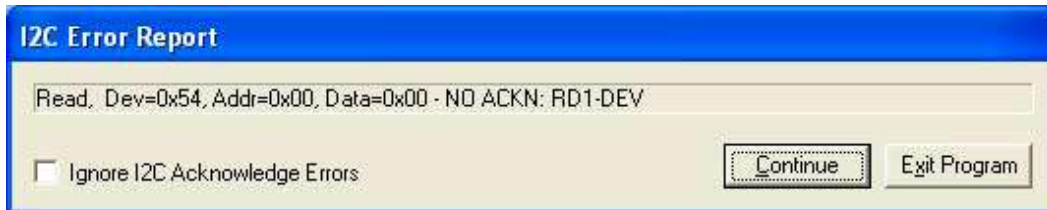
TVP5150AEVM Schematics

This chapter contains the TVP5150AEVM schematics.

TVP5150EVM REV 1.1

I2C
I2C

Power
Power

SDA
SCL

/RESET

Connector
Connector

TVP5150
TVP5150

CVBS1

SCL
SDA

SCL
SDA

Y[7..0]

Y[7..0]

SV

PCLK_SCLK
HSYNC
VSYNC_PALI
FID_GLCO
AVID
INTREQ_GPCL
/RESET

PCLK_SCLK
HSYNC
VSYNC_PALI
FID_GLCO
AVID
INTREQ_GPCL

CVBS2

Testpoints
Testpoints

SCL
SDA
Y[7..0]
PCLK_SCLK
HSYNC
VSYNC_PALI
FID_GLCO
AVID
INTREQ_GPCL
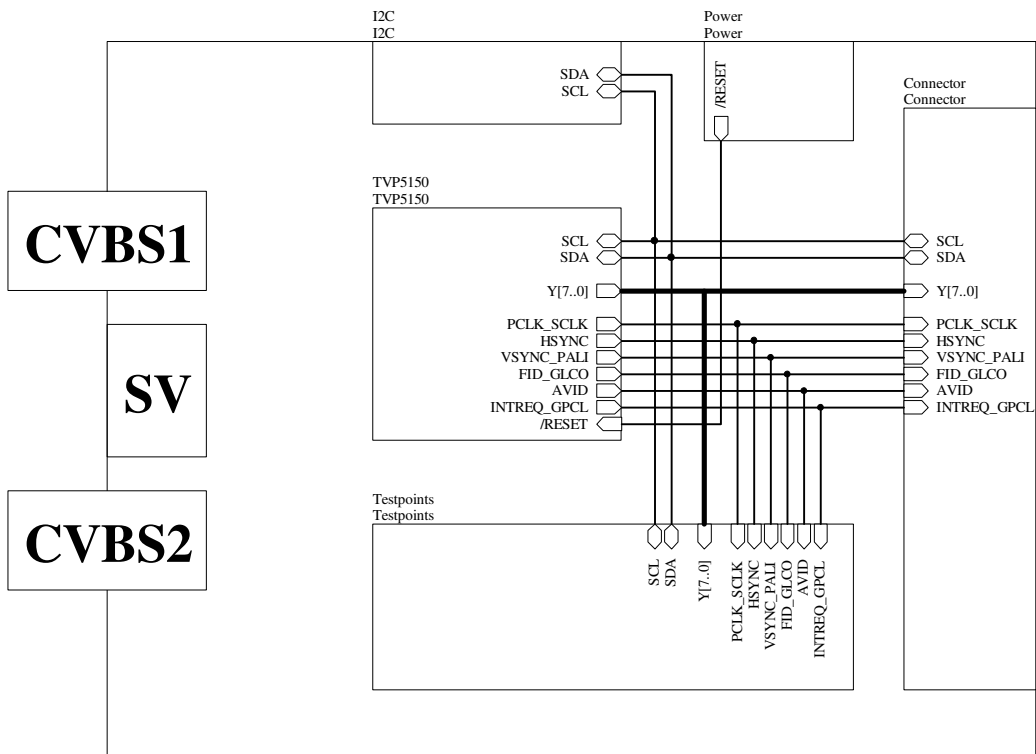
Title		
TVP5150EVM - Block Diagram		
Size A	Number	Revision REV 1.1
Date: 16-Oct-2003		Sheet of
File: C:\Userdata\EVMs\MODULARIZED_EVMS\TVP5150\TVP5150EVM REV 1_1 customer sch\TVP5150EVM_REV 1_1 customer.Ddb		

5V, 3.0A DC INPUT

D5V

F1 FUSE   L1   AS5V

P1
PJ-002BH

H1
3
2
1
HEADER 3

D1
ZENER
SS26

+ C33
47uF

C34
22uF

POWER ON LED (+5V)

D5V

R1
330

LED1
LED
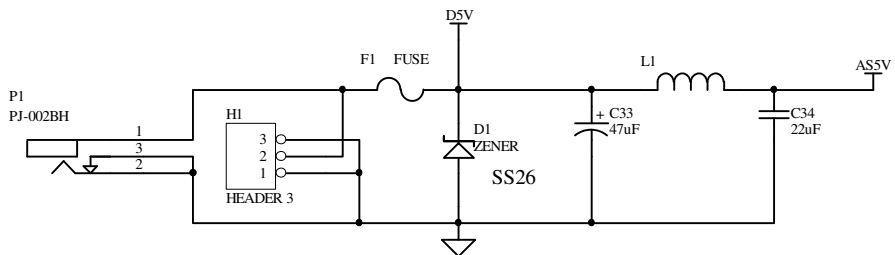
TVP5150 POWER SUPPLY

VOUT1 IS 3.3V 250mA

VOUT2 IS 1.8V 125mA

D5V

R2
250k

U1

1	NC	NC	20
2	VIN1A	VOUT1A	19
3	VIN1B	VOUT1B	18
4	/MR	VSENSE1/FB1	17
5	/EN1	PG1	16
6	/EN2	PG2	15
7	/RESET	VSENSE2/FB2	14
8	GND	VOUT2A	13
9	VIN2A	VOUT2B	12
10	VIN2B	NC	11

PP   21

TPS70851

C4
22uF

C10
0.1uF

C5
22uF

C11
0.1uF

+ C1
47uF

L3   D3_3V

R3 250k

C6
22uF

C12
0.1uF

R4 250k

L4   D1_8V

C7
22uF

C13
0.1uF

L5   A1_8V

+ C2
47uF

C8
22uF

C14
0.1uF

RESET ON POWER UP

D3_3V

R12
100k

S1
PB

C36
1uF

U5

1	CONTROL	VDD	8
2	/RESIN	SENSE	7
3	CT	RESET	6
4	GND	/RESET	5

TLC7733IPW

C35 0.1uF

/RESET

Title
TVP5150EVM - Power

Size
A

Number

Revision
REV 1.1

Date: 16-Oct-2003   Sheet of
File: C:\Userdata\EVMs\MODULARIZED_EVMS\TVP5150\TVP5150EVM REV 1_1 customer sch\TVP5150EVM_REV 1_1 customer.Ddb

P2

Pin	Signal
30	5V
29	5V
28	GND
27	GND
26	GND
25	SCL/PHI_ACK
24	SDA/PHI_RWW
23	PHI_DS/RD
22	PHI_CS
21	PHI_A1
20	PHI_A0
19	PHI_D7
18	PHI_D6
17	PHI_D5
16	PHI_D4
15	PHI_D3
14	PHI_D2
13	PHI_D1
12	PHI_D0
11	GND
10	CLK5/M1
9	FPDAT/VSYA/M2
8	FFRSTW/CBFLAG
7	FSY/HC/HSYA/~BLNK
6	VGAV/SYNC_T
5	FFIE/CCVALID
4	FFWE/DVALID
3	FFRSTWIN/~SCLK
2	FFRE/DIG_H
1	FFOE/DIG_V

Pin	Signal
60	5V
59	5V
58	GND
57	GND
56	GND
55	AMXCLK
54	ALRCLK
53	ASCLK
52	AMCLK
51	GND
50	GPIO7
49	GPIO6
48	GPIO5
47	GPIO4
46	GND
45	GPIO3
44	GPIO2
43	GPIO1
42	GPIO0
41	INTREQ
40	GPCL
39	GND
38	ITRDY
37	SOGOUT
36	VACTIVE
35	D_SCLK
34	D_RDY
33	D_PREF
32	D_HS
31	D_VS

Pin	Signal
90	5V
89	GND
88	GND
87	RCr9
86	RCr8
85	RCr7
84	RCr6
83	RCr5
82	RCr4
81	RCr3
80	RCr2
79	RCr1
78	RCr0
77	GND
76	D9
75	D8
74	D7
73	D6
72	D5
71	D4
70	D3
69	D2
68	D1
67	D0
66	GND
65	RESET
64	PALI
63	HSYNC
62	AVID
61	PREF

Pin	Signal
120	5V
119	GND
118	GND
117	BCb9
116	BCb8
115	BCb7
114	BCb6
113	BCb5
112	BCb4
111	BCb3
110	BCb2
109	BCb1
108	BCb0
107	GND
106	GY9
105	GY8
104	GY7
103	GY6
102	GY5
101	GY4
100	GY3
99	GY2
98	GY1
97	GY0
96	GND
95	FID
94	GLCO
93	VSYNC
92	PCLK
91	SCLK

SAMTEC_TMMS_120PIN_M_RA

D5V D5V D5V D5V

SCL   SCL
SDA   SDA

INTREQ   GPCL   R31 0
R30 0

Y[7..0]   Y[7..0]
Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0

HSYNC   R32 0
R44

VSYNC PALI   R24 0
HSYNC   HSYNC
AVID

FID_GLCO   R27 0
R28 0
PCLK_SCLK   R26 0
R29 0

R25 0

TVP5150EVM - I2C

Size	Number		Revision
A			REV 1.1

Date: 16-Oct-2003 Sheet of
File: C:\Userdata\EVMs\MODULARIZED_EVMS\TVP5150EVM\TVP5150EVM REV 1_1 customer sch\TVP5150EVM_REV 1_1 customer.Ddb
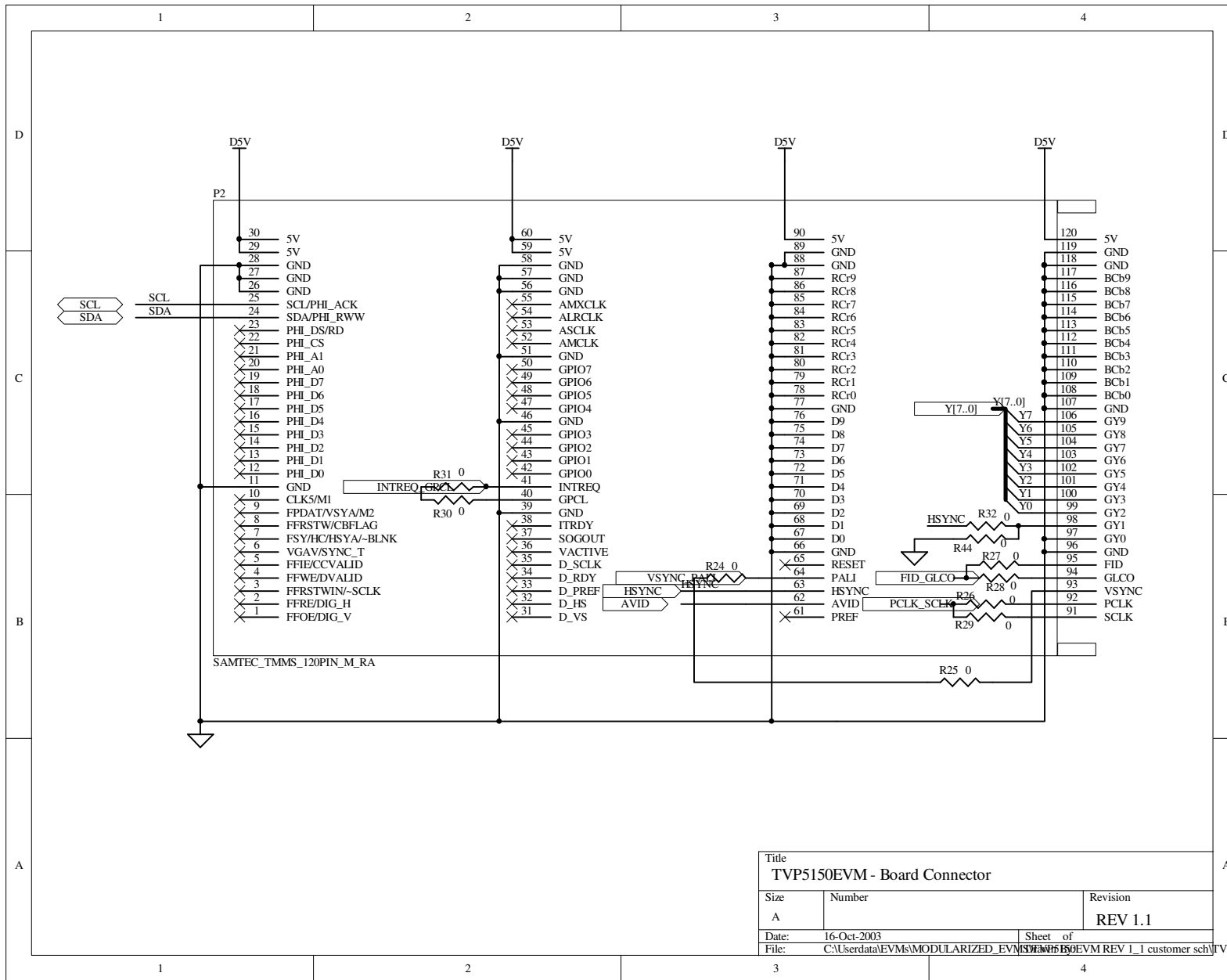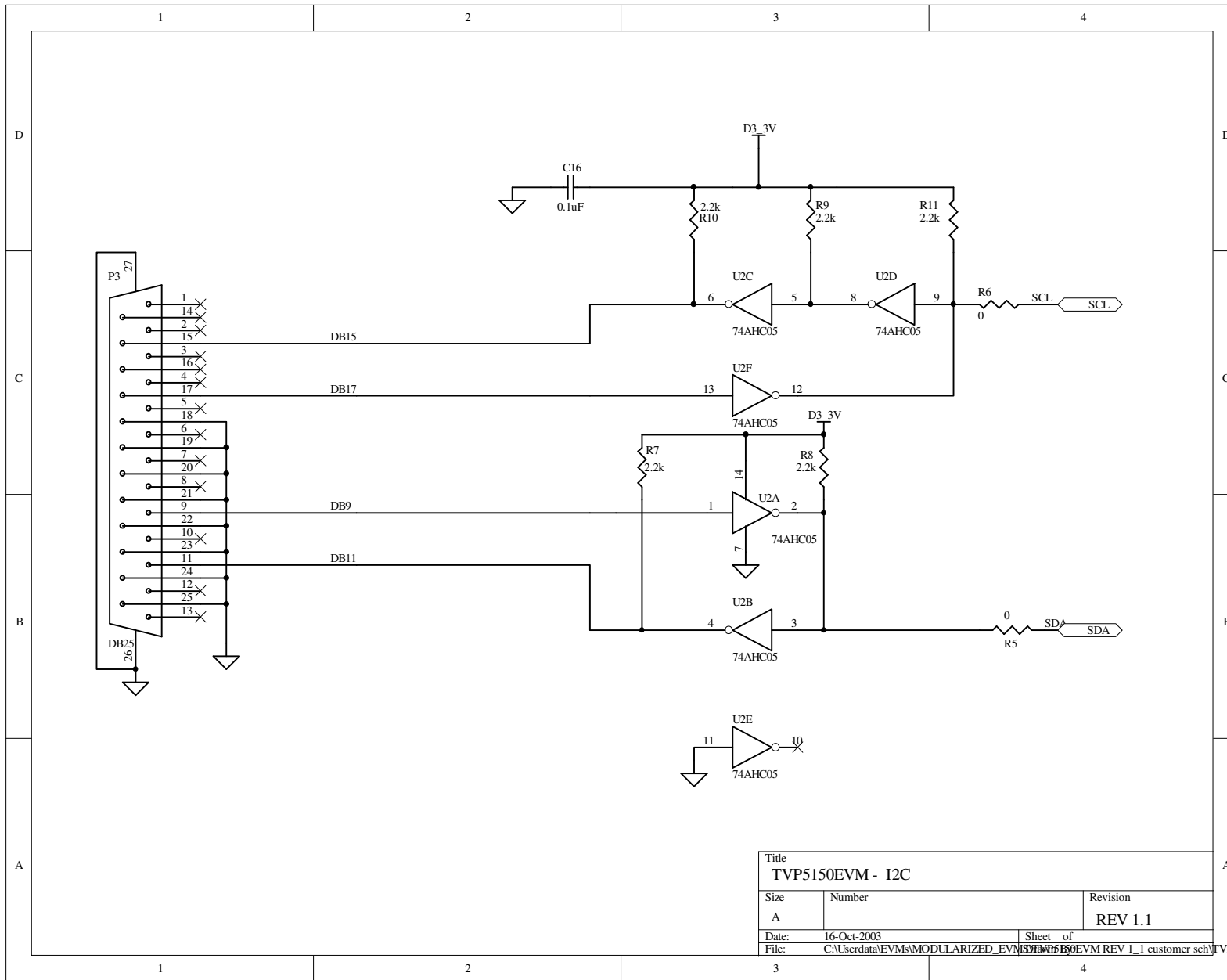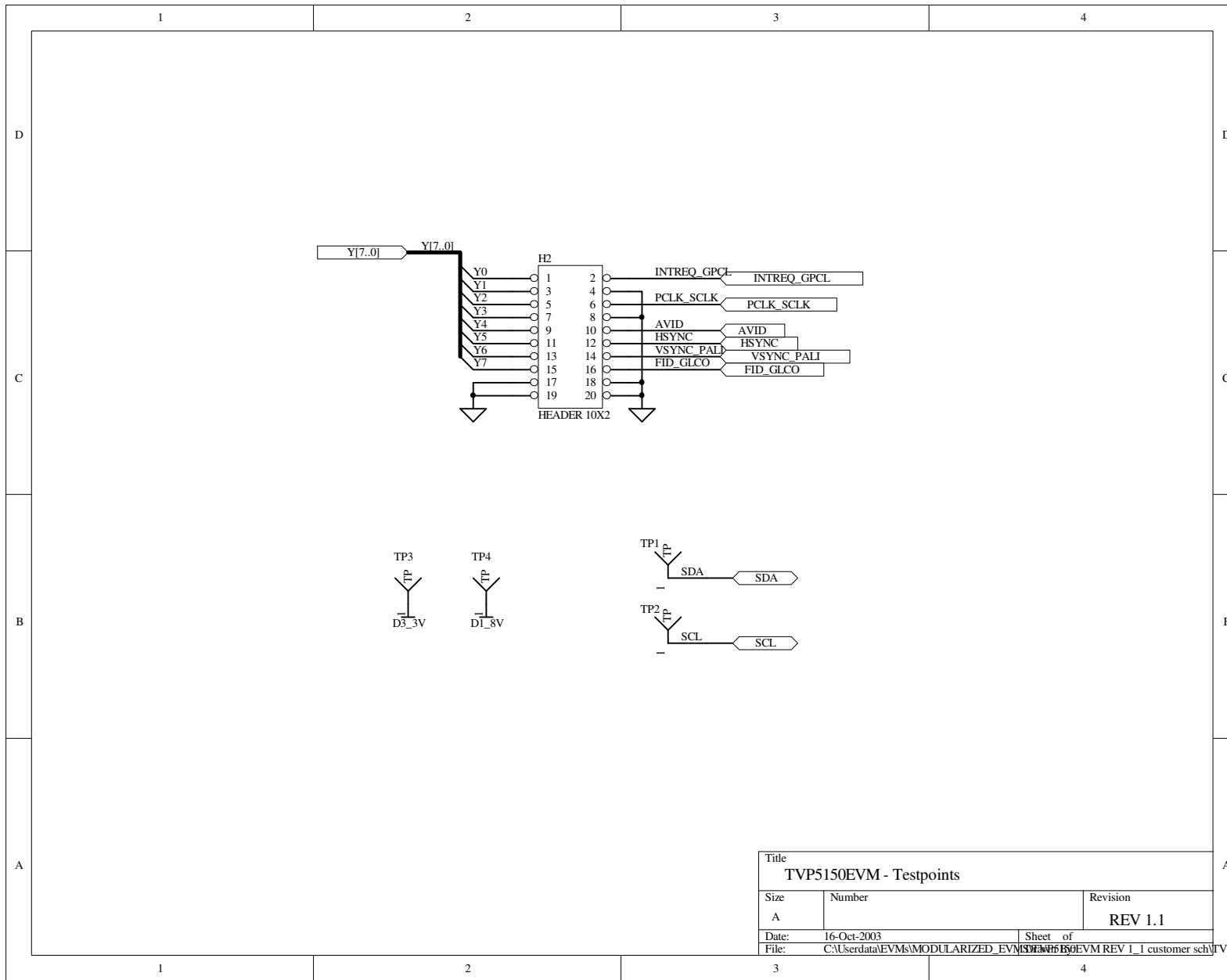
D

Y[7..0]   Y[7..0]

H2

Y0	1	2	INTREQ_GPCL
Y1	3	4	
Y2	5	6	PCLK_SCLK
Y3	7	8	
Y4	9	10	AVID
Y5	11	12	HSYNC
Y6	13	14	VSYNC_PALI
Y7	15	16	FID_GLCO
	17	18	
	19	20	

INTREQ_GPCL

PCLK_SCLK

AVID
HSYNC
VSYNC_PALI
FID_GLCO

HEADER 10X2

C

TP3   TP4   TP1
TP

SDA   SDA

TP
D3_3V   TP
D1_8V

TP2
TP

SCL   SCL

B

A

JP3 SELECTION

1-2 TEST

2-3 Normal Operation - Default

I2C ADDRESS SELECTION

1-2 Base Addr 0xBA

2-3 Base Addr 0xB8 - Default

POWERDOWN

1-2 Normal Operation - Default

2-3 Powerdown

D1_8V

JP3

I/O VDD

D3_3V

IO_DVDD

C21
0.1uF

R16
RESARRAY_ISO_8

INTREQ1   INTREQ_GPCL   INTREQ_GPCL
AVID1   AVID   AVID
HS1   HSYNC   HSYNC
VS1   VSYNC_PALI   VSYNC_PALI
FID1   FID_GLCO   FID_GLCO
YOUT0   Y0   Y[7..0]
YOUT1   Y1
YOUT2   Y2

R41
10k

D3_3V

Y7   JP4
         I2C ADR

R45
10k

D1_8V   D3_3V

C17   C18
0.1uF   0.1uF

D3_3V

R42
10k

~PDN   JP5
          /PDN

C22   C24
1uF   1uF

C23
1uF

REFM   REFP

R15

YOUT3   Y3
YOUT4   Y4
YOUT5   Y5
YOUT6   Y6
YOUT7   Y7

SCLK1   PCLK_SCLK   PCLK_SCLK

RESARRAY_ISO_8

INPUT VIDEO DIVIDER NETWORK
FOR 0-0.75V INPUT RANGE

The resistor network divides the
video input by 2 while allowing the
source and the filter to see 75 ohms

BNC1
BNC_RA   R17
          56.25   CVBS1

R50   R20
225   225

JP2 CONTROL

1-2 CVBS1, CVBS2 - Default

2-3 SV

ANTI-ALIAS FILTER SELECTION

1-3 and 2-4 Filter in video path - Default

1-2 and 3-4 Filter bypassed

A1_8V   A1_8V

C3
0.1uF

C9
0.1uF

REFM
REFP
~PDN
INTREQ1
AVID1
HS1

SV1
SVIDEO

7   R51   R21
6   225   225
5   R18
4   56.25   C   Y
3   R19
2   56.25
1   R53   R39
     225   225

A-SW CONTROL

1-2 GPCL - Default

2-3 JP2

AS5V

R40
10k

JP2

JMP3   JP1 JMP3

JMP3   1   INTREQ_GPCL

L9   L10
2.7uH   2.7uH

C28   C29   C31
330pF   680pF   330pF

R46

C19
0.1uF

R47
75

C20
0.1uF

U3

AIP1A
AIP1B

AIP1A
AIP1B
PLL_AGND
PLL_AVDD
XTAL1/OSC
XTAL2
NSUB
/RESET

CH1_AVDD
CH1_AGND
REFM
REFP
~PDN
INTREQ/GPCL
AVID
HSYNC

VSYNC/PALI   VS1
FID/GLCO   FID1
SDA   SDA   SDA
SCL   SCL   SCL
DVDD
DGND
YOUT0   YOUT0
YOUT1   YOUT1

TVP5150

D1_8V

CVBS1   U4
CVBS2

IN1   D1
S1A   S1B
GND   VDD
S2A   S2B
IN2   D2

CH1A
Y
C
CH1B

AS5V

ADG736

C15
0.1uF

JP6

JP7

R48

R49
75

R43
100k

Y1
14.3181.8MHz

C25   C26
33pF   33pF

/RESET

BNC2
BNC_RA   R22 56.25   CVBS2

R52   R23
225   225

L8   L11
2.7uH   2.7uH

C27   C30   C32
330pF   680pF   330pF

YOUT2   YOUT2
YOUT3   YOUT3
YOUT4   YOUT4
YOUT5   YOUT5
YOUT6   YOUT6
YOUT7   YOUT7

SCLK1   SCLK1

IO_DVDD

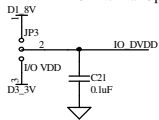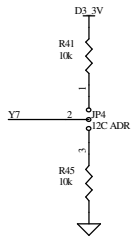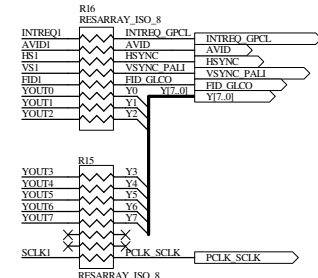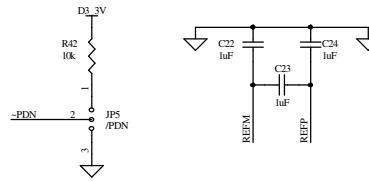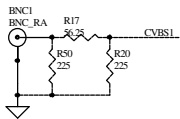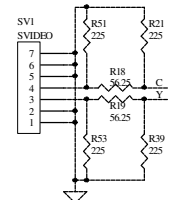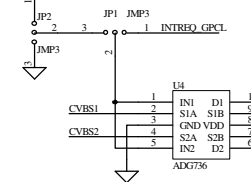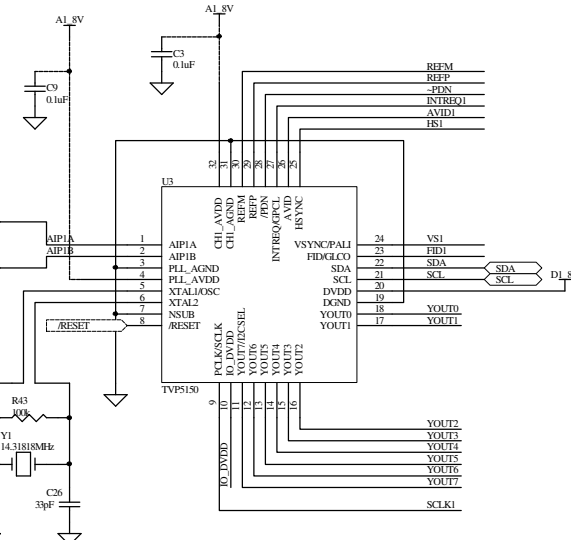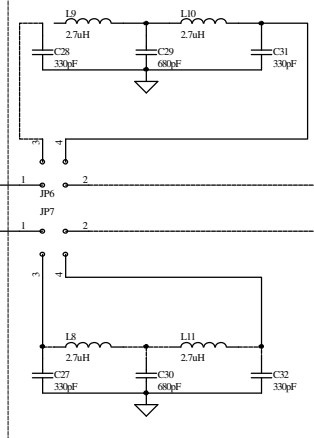| Title	TVP5150EVM - Main		
Size B	Number		Revision REV 1.1
Date: 16-Oct-2003		Sheet of	
File: C:\Userdata\EVMs\MODULARIZED_EVMS\TVP5150EVM REV 1_1 customer sch\TVP5150EVM_REV 1_1 customer.Ddb			