

TVP5146EVM

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

Preface

Read This First

About This Manual

This manual is intended to be a complete reference on how to use the TVP5146EVM. 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 TVP5146EVM
- ☐ Chapter 5 – TVP5146EVM 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

1	Functional Description	1-1
1.1	Description Overview	1-2
1.1.1	Board-Level Description	1-2
1.1.2	Video Input Description	1-4
1.1.3	Video Output Description	1-4
1.1.4	System-Level Description	1-5
2	Setup and Configuration	2-1
2.1	Software Installation	2-2
2.1.1	Software Update	2-2
2.2	Hardware Configuration	2-2
3	System Operation	3-1
3.1	Power Up	3-2
3.1.1	Starting the WinVCC4 Application Program	3-2
3.1.2	WinVCC4 Configuration Dialog Box	3-3
3.1.3	I ² C System Test	3-4
3.1.4	Real-Time Polling	3-5
3.2	Main Menu	3-6
3.2.1	System Initialization	3-6
3.2.2	Adding a Custom Data Set	3-7
3.2.3	Command Files	3-8
3.2.4	Register Editing	3-10
3.2.5	Property Sheets	3-14
4	Troubleshooting the TVP5146EVM	4-1
4.1	Troubleshooting Guide	4-2
4.2	Resolving I ² C Communication Problems	4-3
4.2.1	Corrective Action Dialogs	4-4
4.2.2	Setting the PC's Parallel Port Mode	4-5
4.2.3	General I ² C Error Report	4-6
5	TVP5146EVM Schematic and Board Layout	5-1

Figures

1-1	TVP5146EVM Module	1-3
1-2	Encoder Module	1-4
1-3	TVP5146EVM	1-4
1-4	Filter Selection	1-5
1-5	TVP5146EVM System-Level Block Diagram	1-6
3-1	Multiple Instances Running Message	3-2
3-2	WinVCC4 Configuration	3-3
3-3	I ² C System Test Report	3-4
3-4	Real-Time Polling Dialog	3-5
3-5	Main Menu	3-6
3-6	System Initialization	3-7
3-7	Example CMD File With One Data Set	3-8
3-8	Register Map Editor	3-10
3-9	Encoder Module Editor	3-12
3-10	Generic I ² C Register Editor	3-12
3-11	Indirect Register Editor	3-13
3-12	Property Sheets	3-15
4-1	I ² C System Test Report	4-4
4-2	Cable Connected / Power On Message	4-5
4-3	Connect Cable / Turn On Power Message	4-5
4-4	Parallel Port Mode Message	4-6
4-5	General I ² C Error Report	4-6
4-6	Register Map Editor	4-6

Tables

3-1	Main Menu Summary	3-6
3-2	Register Map Editor Controls	3-10
3-3	Indirect Register Editor Controls	3-12
3-4	Use of Property Sheet Controls	3-15
3-5	Property Sheet Button Controls	3-16
4-1	Troubleshooting Guide	4-2
4-2	Register Map Editor Controls	4-7

Functional Description

The TVP5146EVM evaluation module is a printed circuit board designed for demonstration of the TVP5146 video decoder. The board is designed with a 120-pin connector with all video clocks and data, which allows a connection to multiple back ends. 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	1-2

1.1 Description Overview

The TVP5146EVM uses the PC parallel port to emulate the I²C bus, which provides communication with the TVP5146 video decoder and the video encoder. The WinVCC4 application software that communicates with the devices via I²C is provided.

The analog video inputs supported by the TVP5146EVM include composite video, S-video, component video (RGB and YPbPr), and SCART (CVBS + component). More detail about the video inputs are 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 our EVM include composite video, S-video, and component video. These are all output simultaneously.

To experiment with the programmable features of the TVP5146 video decoder and the video encoder, the parallel port of the TVP5146 is connected to the parallel port of a PC. A Windows compatible application provides the user interface for performing register-level and high-level control of the TVP5146 video decoder and the video encoder. This application is called WinVCC4 (Windows Video Control Center).

1.1.1 Board-Level Description

The TVP5146EVM consists of the TVP5146EVM module (see Figure 1-1) and the Encoder EVM module (see Figure 1-2). A four row 120-pin connector connects the boards. The block diagram of the EVM set is shown in Figure 1-3.

Figure 1–1. TVP5146 Module

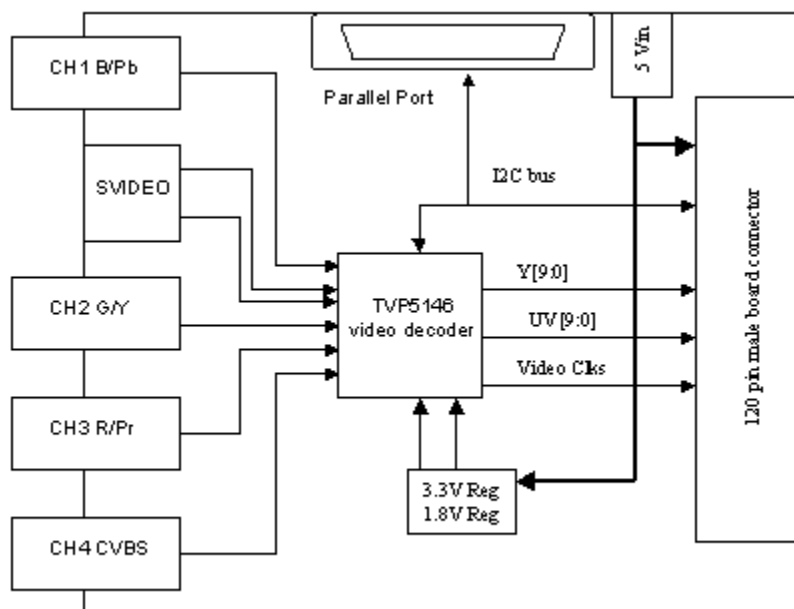


Figure 1–2. Encoder Module

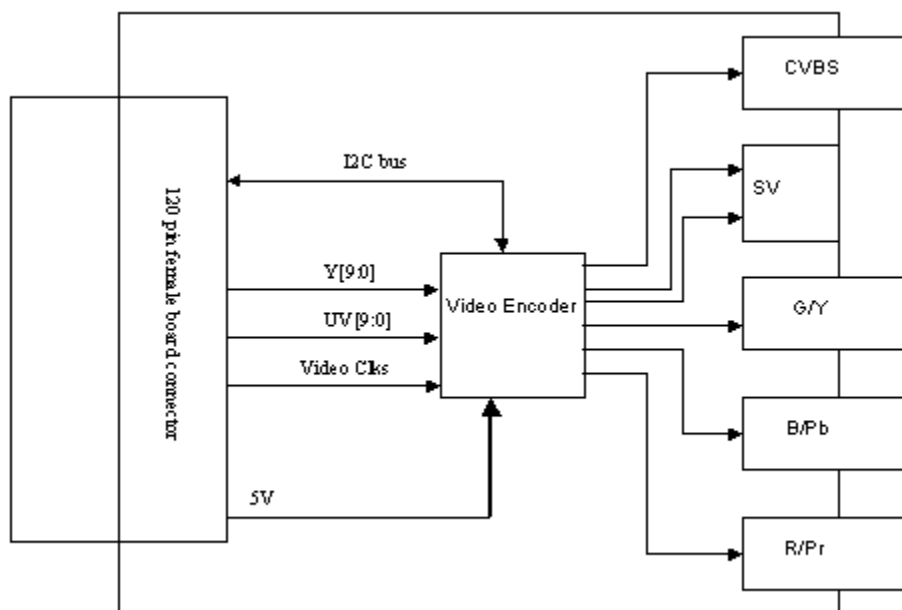
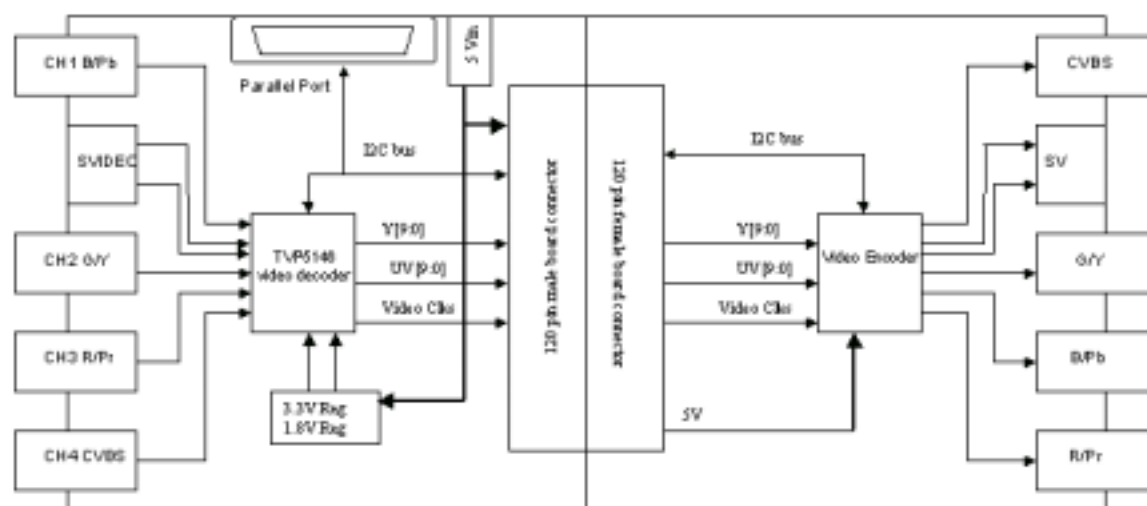


Figure 1–3. TVP5146EVM



1.1.2 Testpoints and Jumpers

The TVP5146EVM 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 TVP5146 must receive a reset.

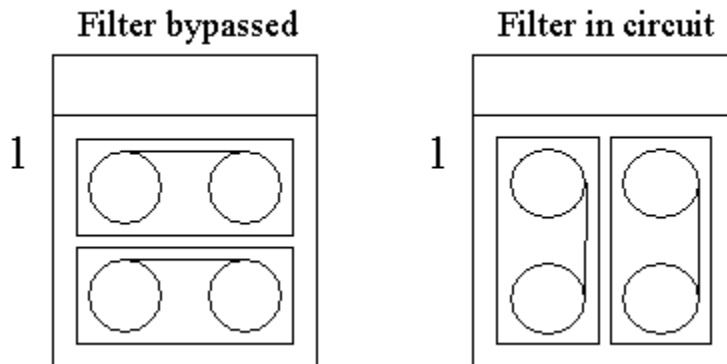
1.1.3 Decoder to Encoder Connection

The TVP5146EVM uses a four row 120-pin connector to share common signals and the 5-V power supply between the boards. This connection allows multiple backends to be connected to the TVP5146EVM. The EVM package is shipped with an encoder module. This connector shares all digital video bits (Y[9:0] and C[9:0]), all video clocks (VS, HS, GLCO, AVID, FID, and DATACLK), 5 V, ground, I²C bus (SCL and SDA), and reset.

1.1.4 Video Input Description

The TVP5146 has four analog channels that accept up to 10 video inputs that can be configured with a combination of RGB, YPbPr, CVBS, or S-video video inputs. There are many different combinations that can be configured; however, the TVP5146EVM only has four BNC connectors and one S-video jack. The four BNC connectors allow the user to input four composite video inputs, or one composite and one component input. Each input has an anti-alias filter that can be in-circuit (default), or bypassed by jumpers (JP1, JP2, JP7–JP10). To select the filter, the shunts need 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 below in Figure 1–4. The boards are shipped with the filter in circuit.

Figure 1–4. Filter Selection



1.1.5 Video Output Description

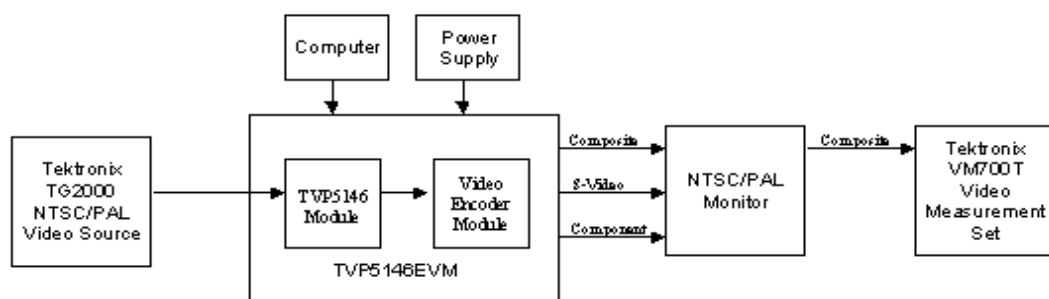
The TVP5146's digital outputs are routed to the 120-pin edge connector along with all video clocks on the TVP5146 module. The encoder module connects to the 12-pin connector and is capable of receiving digital video with or without embedded syncs. The analog outputs of the encoder module are composite, 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 TVP5146 is shown in Figure 1–5. Typical test equipment is also shown. The primary features of this configuration are listed below.

- ☐ Power is provided by a single 5-V power supply provided with the EVM and is shared between both modules via the 120-pin connector.
- ☐ Supported analog inputs include composite video, S-video, and component video.
- ☐ Reencoded composite video and S-video are output via the encoder module.
- ☐ Component (YPbPr) video is output via the encoder module.
- ☐ I²C bus initialization of the video devices via a PC parallel port
- ☐ TVP5146 video decoder performance parameters may be measured with a video analyzer.

Figure 1–5. TVP5146EVM System Level Block Diagram



Setup and Configuration

The following summarizes the steps for setup and operation of the EVM. 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)

Topic	Page
2.1 Software Installation	2-2
2.2 Hardware Configuration	2-2

2.1 Software Installation

All necessary software for the TVP5146EVM is provided on 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 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 this driver, run PROT95NT.exe, which is located in the root directory of the installation CD-ROM.
- 5) Go to Start->Programs->TVP5146EVM Software to start WinVCC4 and to browse documentation.
- 6) 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 our web site at www.ti.com for the latest version of the EVM software and the device software.

2.2 Hardware Configuration

Figure 1-3 shows the TVP5146EVM layout and indicates the location of the power supply jack and the appropriate connectors. All connectors are labeled according to their function. To prepare the EVM for evaluation, connect the following:

- ☐ TVP5146 module to encoder module
- ☐ Parallel port cable from a PC to the TVP5146EVM
- ☐ Analog video in
- ☐ Analog video out
- ☐ Power supply

The system comes with the anti-alias filters connected. If you would like to bypass the filters, you need to set the filters as stated in section 1.1.2 *Video Input Description*.

The I²C address LSB can be set 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 I²C address is now 0xBA. If you want to change the address, you need to ensure that WinVCC4 is using the selected address.

System Operation

This chapter describes the system operation.

Topic	Page
3.1 Getting Started	3-2
3.2 Main Menu	3-6

3.1 Power Up

After *Software Installation* (section 2.1) and *Hardware Configuration* (Section 2.2) have been completed, the TVP5146EVM 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. Run *WinVCC4* from the Windows™ Start Menu:

Start->Programs->TVP5146EVM 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 WinVCC4 or WinVCC4 CONFIGURATION. Then restart WinVCC4.

Figure 3–1. WinVCC4 Multiple Occurrences Error Message

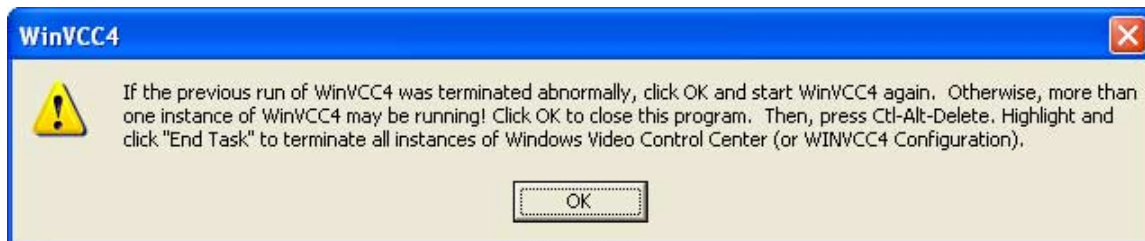


Figure 3–2. WinVCC4 I²C Address Configuration

The image shows a Windows-style dialog box titled "WinVCC4 Configuration". It is divided into two main sections: "I2C SYSTEM CONFIGURATION" and "PROGRAM OPTIONS".

I2C SYSTEM CONFIGURATION

DEVICE FAMILY	SPECIFIC DEVICE	I2C SLAVE ADDR (W)
VID_DEC (Video Decoder)	TVP5146	☒ B8h ☐ BAh
VID_ENC (Video Encoder)	Encoder Module	☒ 54h ☐ 56h
THS8200 (HDTV/SDTV/RGB DAC)	NOT USED	☒ 40h ☐ 42h

PROGRAM OPTIONS

☒ ENABLE	I2C System Test (after clicking OK).
☒ ENABLE	I2C Acknowledge Checking
☒ ENABLE	Auto-Update from Device (when opening or selecting a window).

At the bottom of the dialog box are two buttons: "OK" and "Exit Program".

3.1.2 WinVCC4 Configuration Dialog Box

The *I²C Address 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 I²C slave addresses for each device must match the I²C slave address selected by jumper on the TVP5146EVM. These jumpers are set by the factory to use B8h for the video decoder and 54h for the encoder.

It is also important to select the correct specific devices. The TVP5146 and encoder module must be selected for this EVM.

The Program Options should all be enabled. Disabling these options should only be required if you are debugging a problem with the I²C bus itself.

Clicking *OK* begins I²C communication with the selected devices.

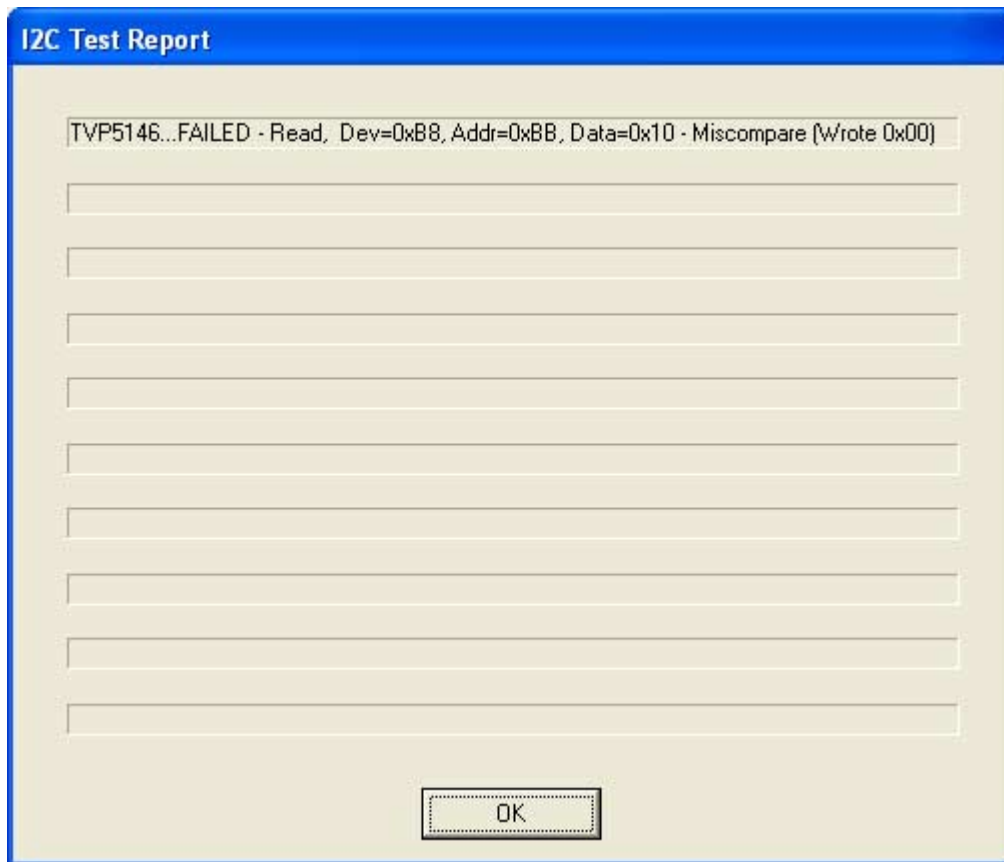
3.1.3 I²C System Test

The I²C system test of selected registers runs immediately after closing the *WinVCC4 Configuration* dialog box with *OK* (unless the I²C system test program options button was disabled).

If the I²C system test passes, no message appears. If the test failed, a dialog box like Figure 3–3 appears. See section 4.1 on resolving I²C communication problems.

The I²C system test can be run at anytime by clicking *Run System I²C Test* in the *Tools* menu.

Figure 3–3. I²C System Failure



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 TVP5146 is VIDEO-STANDARD AUTO-SWITCH POLLING.

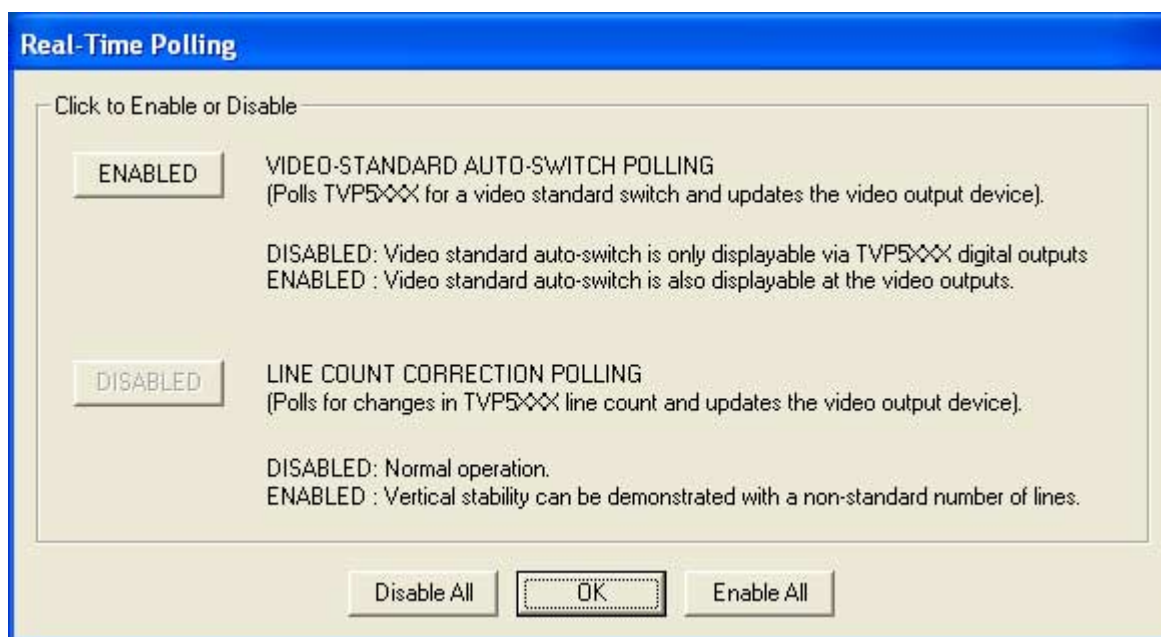
When TVP5146 detects a change in the input video standard, it automatically switches to operation in the detected standard (which includes all necessary

I²C register initialization) for proper decoding of the input video. To enable auto-switch on the TVP5146, the *Set Video Standard* register must be set to auto-switch mode (Reg 0x02 = 0x00).

If the WinVCC4 auto-switch polling function is enabled, the detected video standard status from the TVP5146 is polled until a change in the input video standard (or in the TVP5146 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 as 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 Box



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. WinVCC4 Main Menu

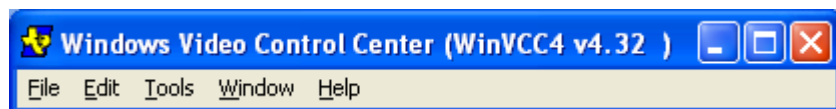


Table 3–1. Main Menu Summary

Menu	Contents
File	Exit WinVCC4
Edit	Register Map Editor Encoder Module 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 below 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).

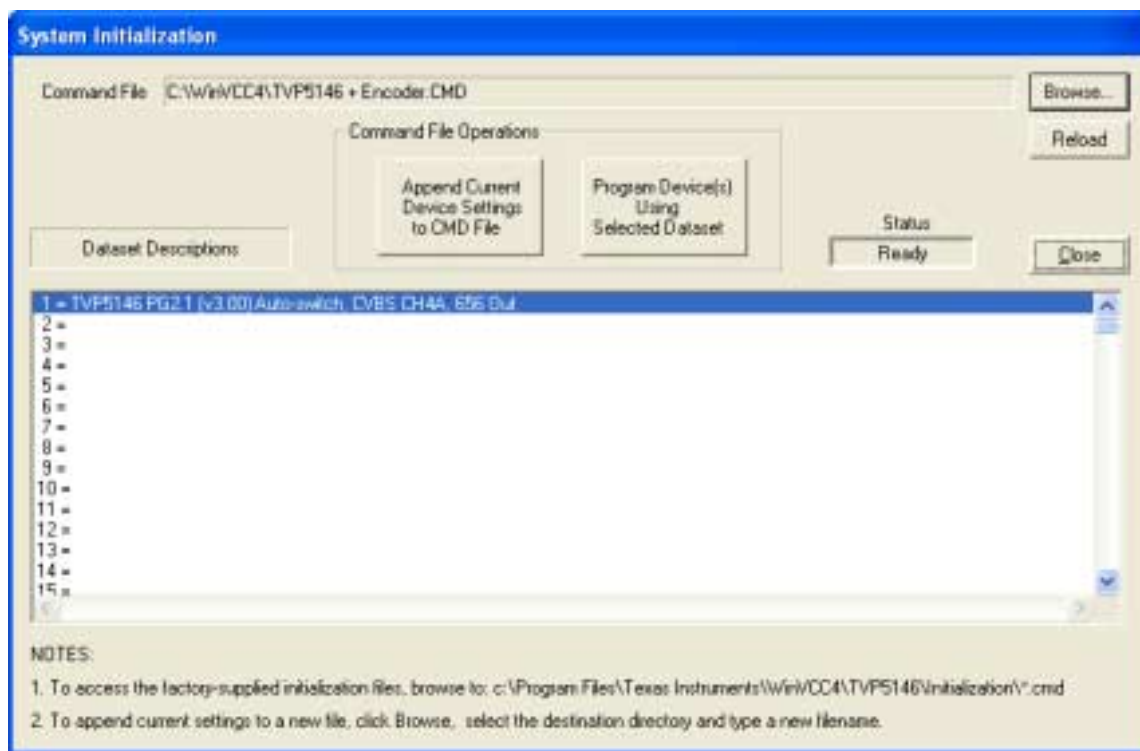
The command file is opened using the *Browse...* button. Once the command file is opened, a text list displays descriptions of the individual data sets contained within the command file.

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 I²C 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 I²C bus is not communicating. See chapter 4: *Troubleshooting the TVP5146EVM*, 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

After you program the EVM via the *System Initialization* tool using the factory-supplied command file through the *Property Sheets* tool, 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 from the decoder and encoder will be 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: The command file (.CMD) must be saved as plain text.

3.2.3 Command Files

The command file is a text file that can be generated using any common editor; however, it must be saved as plain text. Command files are especially useful for quickly switching between the various system configurations. These .CMD files are unrelated to the typical Windows .CMD files.

A default command file has been provided on the CD-ROM. This command file should contain most of the desired setups. This command file is located at:

*c:\Program Files\Texas Instruments\WinVCC4\TVP5146\Initialization
Files\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 below. 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. Command File Example

```
BEGIN_DATASET
DATASET_NAME,"TVP5146 PG2.1 v3.00 Auto-switch, CVBS CH4A, 656 Out"
// Initialize video encoder using an include file
INCLUDE, EncoderNTSC656_RTC.INC
// Program TVP5146 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 hexadecimal (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 datasets is 250.
- 5) The dataset 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 encoders

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 will be 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 four available modes of register editing: *Register Map Editor*, *Encoder Module Editor*, *Generic I²C Register Editor*, 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. If the intended device type is not shown, use the Windows menu to activate the existing window.

Figure 3–8. Register Map Editor

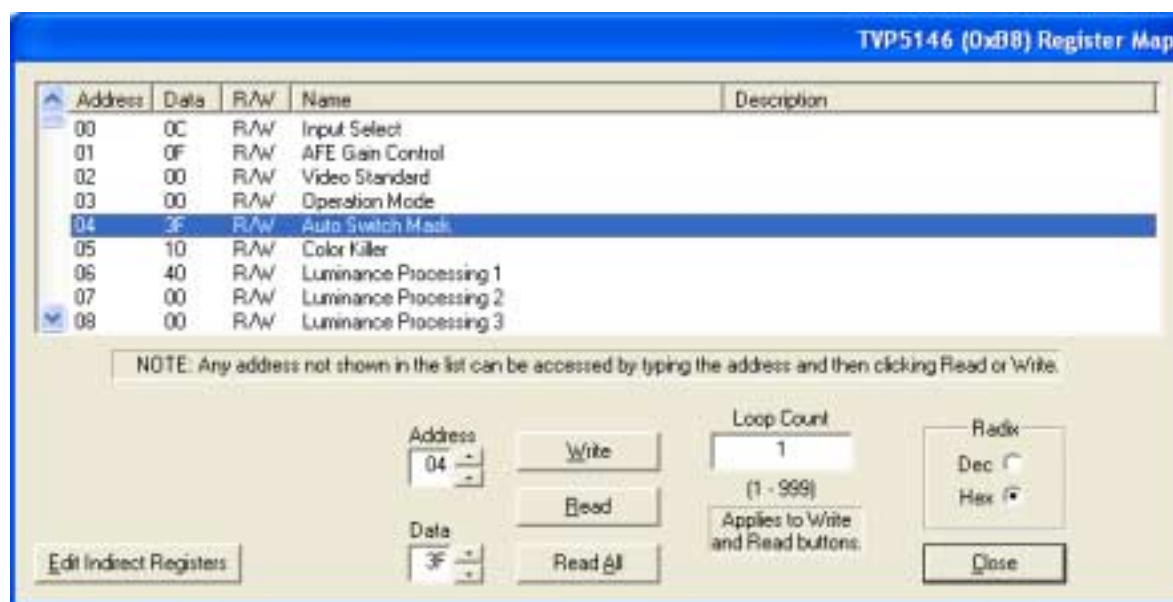


Table 3–2. Register Map Editor Controls

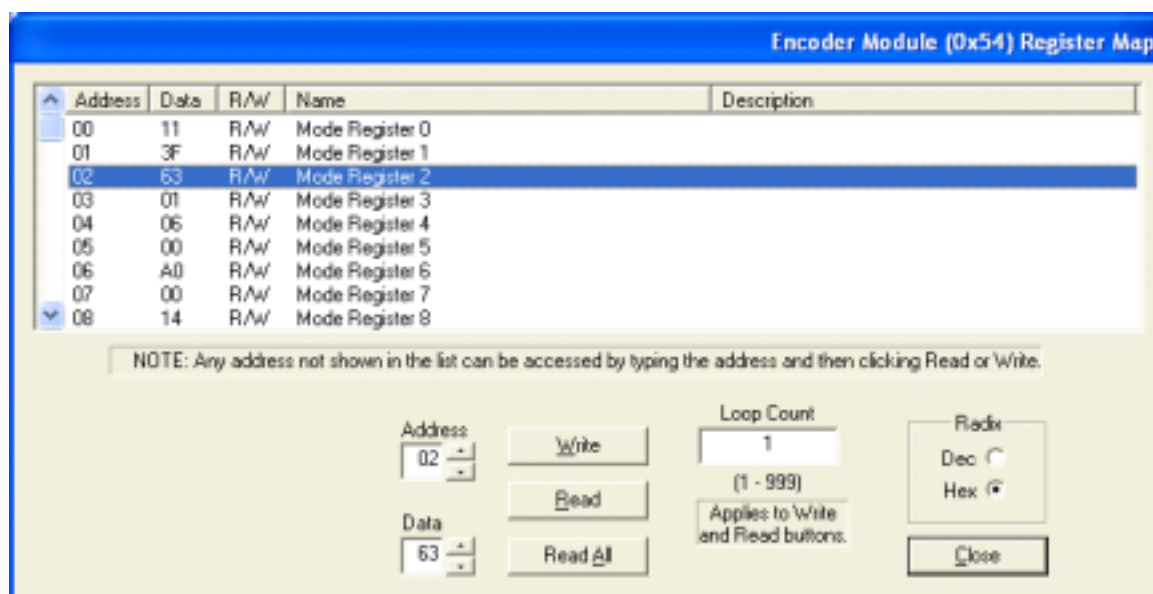
Control	Definition
Register Window	Scrolling text box that displays the address and data for the I ² C registers that are defined for the device.
Address Edit Box	This contains the I²C sub-address that will be accessed using the <i>Write</i> and <i>Read</i> buttons. Clicking on a row selects an address, which then appears in the address edit box. NOTE: After clicking on a row, the <i>Data Edit</i> box contains the data that was in the register window. The device has not yet been read. The address up/down arrows are used to jump to the next/previous sub-address that is defined for the device. If an address is not defined for the device, it can still be accessed by typing the sub-address in the <i>Address Edit</i> box.
Data Edit Box	This contains the data which will be written to or was read from the I²C sub-address. The data up/down arrows incr/decr the data value by 1.
Write Button	Writes the byte in the <i>Data Edit</i> box to the address in the <i>Address Edit</i> box. The I²C 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 <i>Address Edit</i> box into the <i>Data Edit</i> box and the register window.
Read All Button	Reads all defined readable registers from the device and updates the register window.
Hex Button	Converts all values in the register window and address and data edit boxes to hexadecimal.
Dec Button	Converts all values in the register window and address and data edit boxes to decimal.
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 TVP5146.

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



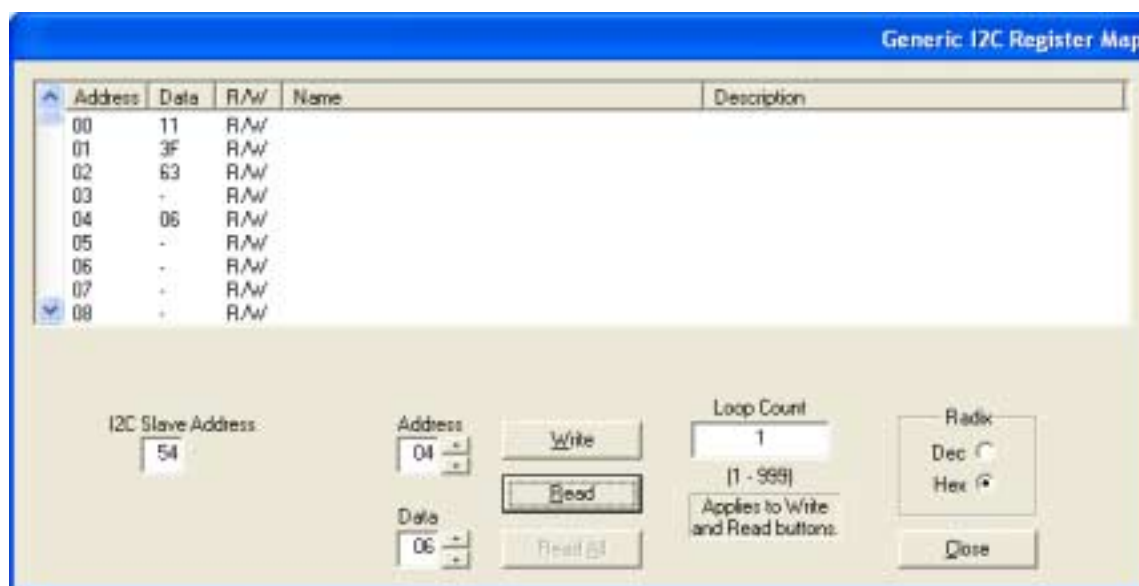
3.2.4.3 Generic I²C Register Editor

The *Generic I²C Editor*, as shown in Figure 3–10, allows the display and editing of any device on the I²C bus. This editor works like the *Register Map Editor*, except that the I²C 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 TVP5146. Then, click the *Edit Indirect Registers* button. The operation of the controls of the indirect register editor is explained in Table 3–3.

Figure 3–11. Indirect Register Editor

TVP5146 (0xB8) Indirect I2C Registers

Base Address Selector
80 0500 - VBI Data Processor Control

MSB <----- DATA -----> LSB

BASE ADDR: 800500
ADDR OFFSET: 4
LOOP COUNT: 1 (1 - 999)

Read Write Close

NOTES:
=> LSB data is at the lowest address.
=> Base Address Selector allows quick input of base addresses by functional block.
=> Any base address can be typed directly into the BASE ADDR edit box.
=> Loop count applies to both read and write buttons.

Table 3–3. Indirect Register Controls

Control	Definition
Base Address Selector	The indirect registers use a 24-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 TVP5146.
Base Address Edit Box	This allows the 24-bit base address to be typed in directly.
Address Offset Edit Box	This contains the lowest byte of the 24-bit internal address. The full 24-bit address is formed by adding the base address to the address offset. The address up/down arrows increment/decrement the address offset by 4.
Data Edit Boxes	This contains the 32-bit data word that will be 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 (4) bytes in the <i>Data Edit</i> boxes starting at the 24-bit indirect address BASE+OFFSET.
Read Button	Reads (4) consecutive data bytes starting at the 24-bit indirect address BASE+OFFSET to the <i>Data Edit</i> 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.

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.

3.2.5.1 Reading the Register Map

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 (by clicking on it).

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), the user can 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.

Figure 3–12. Property Sheets

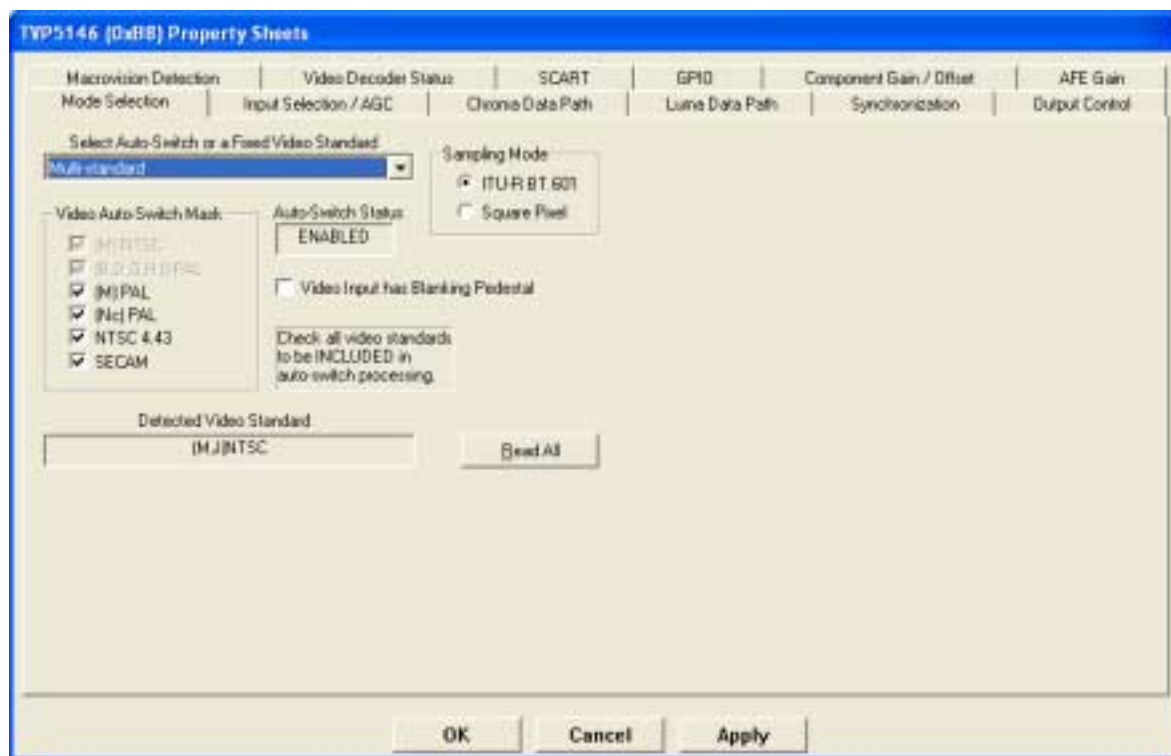


Table 3–4. Use of Property Sheet Controls

Property Sheets	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 <i>Apply</i>
Drop-Down List	Select from a text list	After <i>Apply</i>
Edit Box	Type a number	After <i>Apply</i>
Edit Box with Up/Down arrows	Use up/down arrows or type a number	Up/Down Arrows: Immediately Type a number: After <i>Apply</i>
Slider	Slide a lever	Immediately
Pushbutton	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 <i>Apply</i> 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.

Troubleshooting the TVP5146EVM

This chapter discusses ways to troubleshoot the TVP5146EVM.

Topic	Page
4.1 Trouble Shooting Guide	4-2
4.2 Resolving I²C Communication Problems	4-3

4.1 Troubleshooting Guide

If you are experiencing problems with the TVP5146EVM hardware or the WinVCC4 software, see Table 4–1 for available solutions.

Table 4–1. TVP5146 Troubleshooting

Symptom	Cause	Solution
At startup, the error message <i>Cannot find DLL file DLPORTIO.DLL</i> appears.	The parallel port driver supplied with the EVM 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->TVP5146, Input Selection/AGC page, select the correct video input(s) and click <i>Apply</i> . (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->TVP5146, Output page, check the <i>Enable YCbCr Outputs</i> and <i>Enable Clock Outputs</i> check boxes and click <i>Apply</i> .
No color	Wrong mode selected for color subcarrier genlock output.	Go to Edit->Property Sheets->TVP5146, Synchronization page, set the Fsc control format to RTC and click <i>Apply</i> .
	GLCO pin is not set to output the GLCO signal.	Go to Edit->Property Sheets->TVP5146, GPIO page, set the drop down box labeled "GLCO/I2CA Terminal" to Genlock Output and click <i>Apply</i> .
Screen colors are only magenta and green.	Wrong YCbCr output format.	Go to Edit->Property Sheets->TVP5146, Output page, set the YCbCr output format to 10-bit 4:2:2 ITU-R BT.656 with embedded syncs mode and click <i>Apply</i> .
Video standard auto-switch does not work on the video decoder side.	Auto-switch masks are not set correctly.	Go to Edit->Property Sheets->TVP5146, Mode Selection page, check all standards to be included in auto-switch processing and click <i>Apply</i> .
	Video decoder is not in auto-switch mode.	Go to Edit->Property Sheets->TVP5146, Mode Selection page, set the drop-down box to <i>Multi-Standard</i> and click <i>Apply</i> .
Video standard auto-switch does not work on the video encoder side.	Auto-switch polling is not enabled.	Click <i>Real-Time Polling</i> in the Tools menu. Click <i>Enable All</i> and <i>OK</i> .

Table 4–2. Table 4–2. I²C Troubleshooting

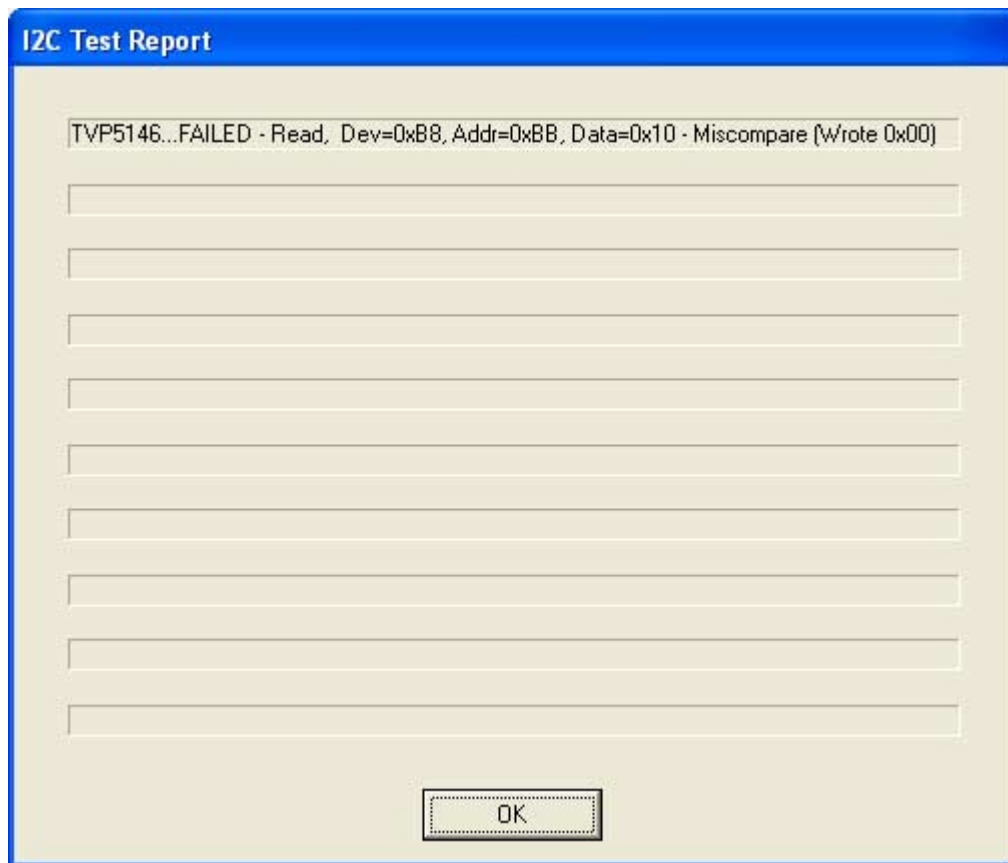
Symptom	Cause	Solution
No I ² C communication	I ² C slave address is wrong.	Close and restart WinVCC4. Choose the alternate slave address in the WinVCC4 Configuration dialog.
	Parallel cable is not connected from PC parallel port to the EVM DB25 connector.	Connect cable.
	EVM is not powered on.	The power 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.
	PC parallel port mode is not set correctly.	Reboot PC, enter BIOS setup program, 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 4.2.1.
	Device was placed in power-down mode.	Press the reset button on the TVP5146EVM.
	EVM was configured for an external I ² C master.	Reinstall 0-Ω resistors R5 and R6. Control EVM using the PC parallel port.
	Still no I ² C communication	The 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.

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 TVP5146 failed, using slave address 0xB8, sub-address 0xBD. 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 Dialog* box (see Figure 4–2) appears to help fix the problem.

Figure 4–1. I²C System Failure Dialog Box



4.2 Corrective Action Dialogs

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

Figure 4–2. Corrective Action Dialog Box



- 1) If the cable is NOT connected from the PC parallel port to the TVP5146EVM or if the EVM power is not on:
 - a) Click NO.
 - b) The dialog shown in Figure 4–3 appear instructing you to correct the problem.

- c) Correct the problem.
- d) Click *OK* to continue. The real-time polling dialog should appear. See section 3.2.3.

Figure 4–3. Corrective Action Required

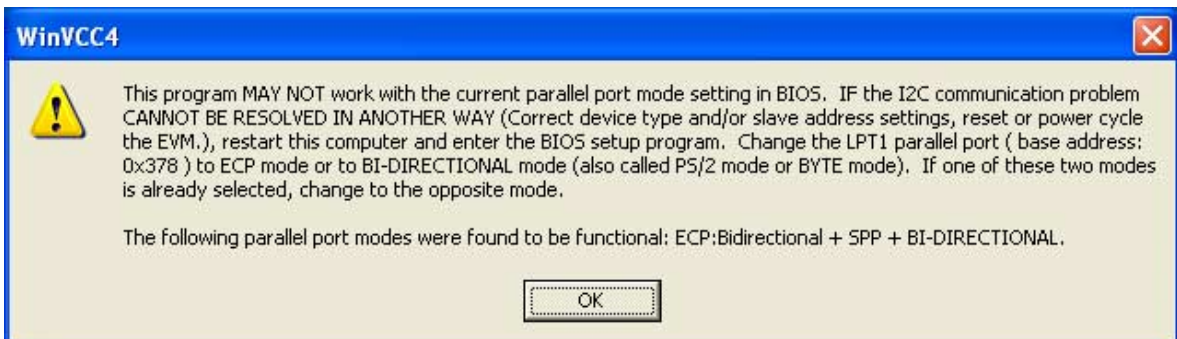


- 2) If the cable IS connected from the PC parallel port to the TVP5146EVM AND the EVM power IS on:
 - a) Click Yes.
 - b) The dialog shown in Figure 4–4 appears. This dialog appears if the PC parallel port mode setting may need to be changed.

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 TVP5146).

- c) Click *OK* to continue.
- d) The real-time polling dialog appears. Click *OK* to close it and get to the main menu.
- e) Click *Exit* in the *File* menu to exit the program.
- f) See trouble shooting guide above.

Figure 4–4. Corrective Action Required



4.2.1 Setting the PC's Parallel Port Mode

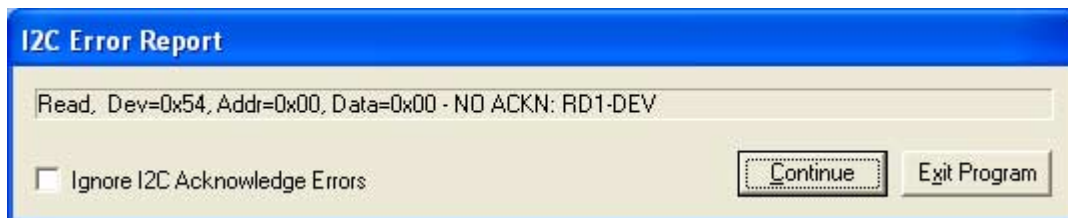
NOTE: Only run the PC BIOS setup program if the I²C communication problem CANNOT BE RESOLVED IN ANOTHER WAY. (Correct slave address settings, reset or power cycle the EVM, and/or check that the device type selected was TVP5146).

- 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 ECP 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.

4.2.2 General I²C Error Report

The error report shown in Figure 4–5 appears when an I²C error occurs at any time other than after the I²C system test. In this example, there was an acknowledge error at slave address 0x54 (the video decoder module). The error occurred on *Read Cycle Phase 1* on the device (slave) address byte.

Figure 4–5. I²C Error

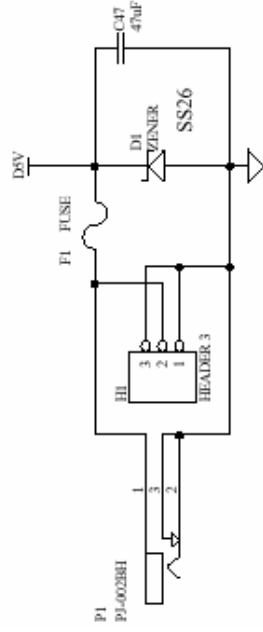


TVP5146EVM Schematics

This chapter contains the TVP5146EVM schematics.

Title					
TVP5146EVM					
Size	Number		Revision		
A B			REV 1.1		
Date:	10 Jul 2018		Status of		
C:\ProgramData\Texas Instruments\TVP5146EVM\TVPEVM_Rev1.1.dwg					

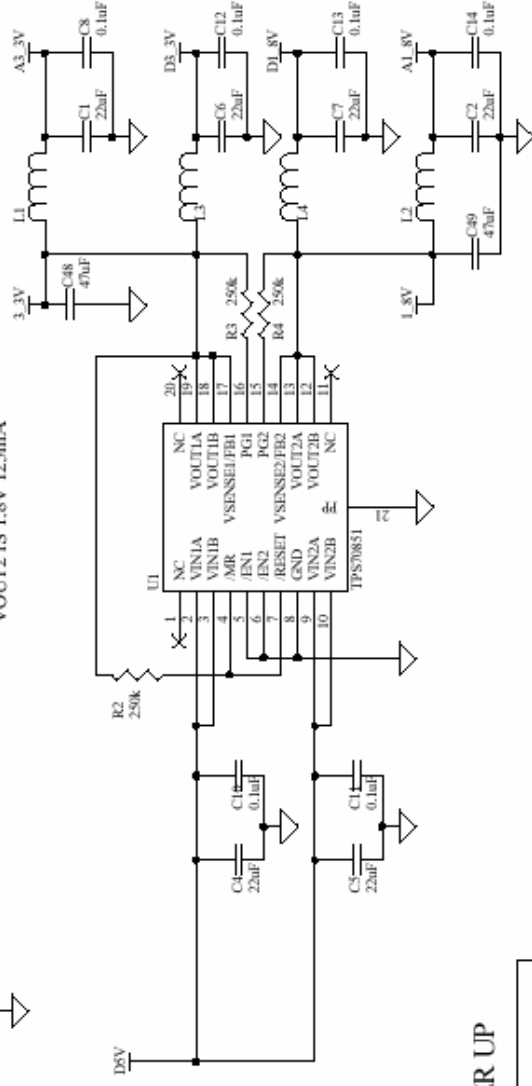
5V, 3.0A DC INPUT



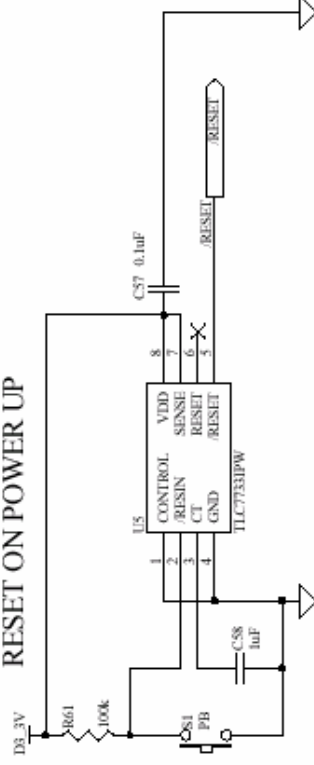
TVP5146 POWER SUPPLY

VOUT1 IS 3.3V 250mA
VOUT2 IS 1.8V 125mA

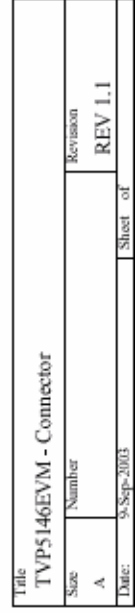
POWER ON LED (+5V)

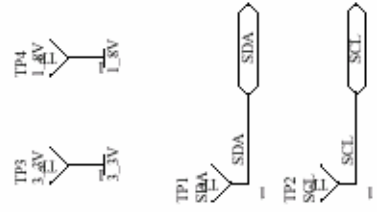
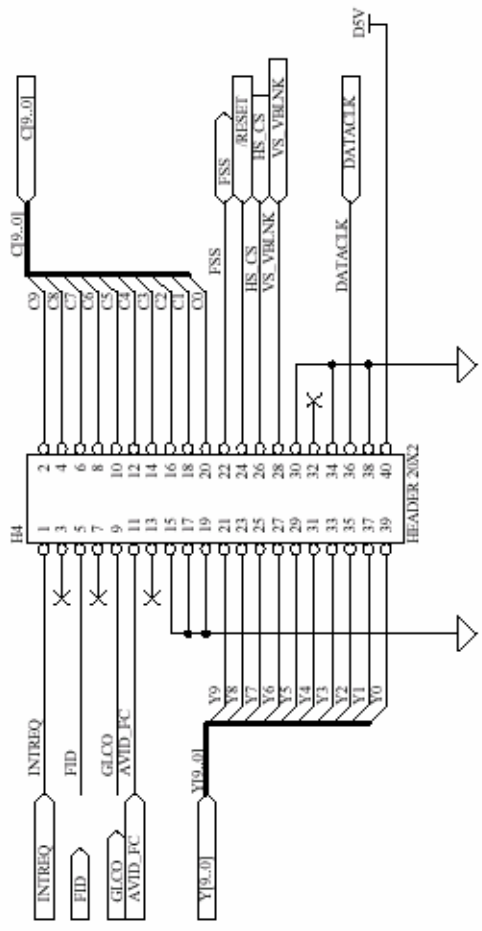


RESET ON POWER UP

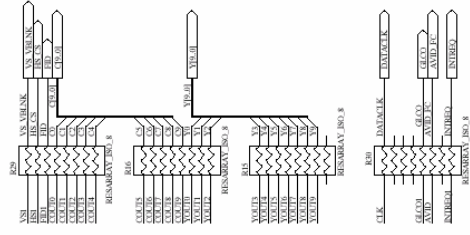
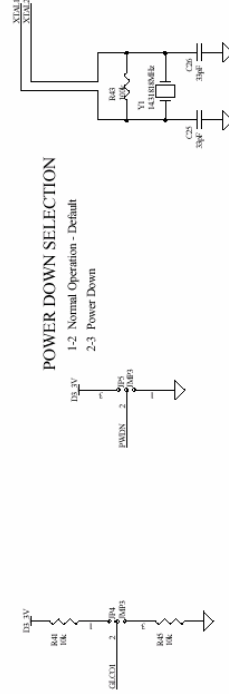
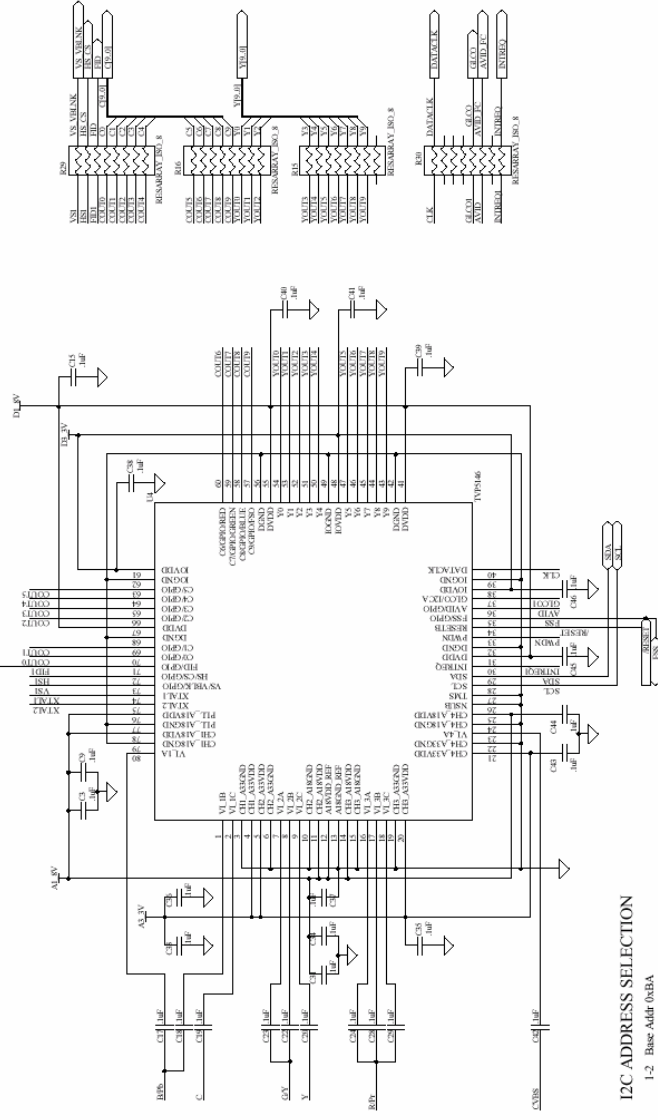
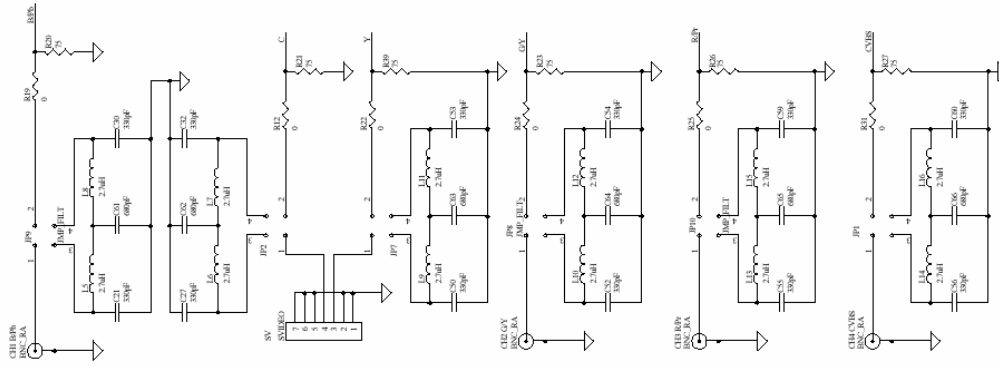


Title		TVP5146EVM - Power	
Size	Number	Revision	REV 1.1
A			
Date:	9-Sep-2003	Sheet	of

Title
TVP5146EVM - Connector



Title			
TVP5146EVM - Testpoints			
Size	Number	Revision	
A		REV 1.1	
Date:	9-Sep-2003	Sheet	of



Title		Revision	
TVP5146EVM - Main		REV 1.1	
Size	Number		
C			
Date:	3 Sep 2013	Sheet of	
File:	C:\projects\TVP5146EVM\REV 1.1\TVP5146EVM REV 1.1.schematic.sch		