

**Using the Intersil XLABVIEW01 Hardware
with
Intersil XDCCP ICs**

**Release 3.2.5
11/20/07**

1.0 System Requirements

- Windows 98/NT/2000/XP
- Parallel port available

2.0 Hardware Installation

- Connect the board to the parallel port of a PC using a standard parallel port cable
- Verify the parallel port address on your PC (default is set for 0x378, but it could also be set to 0x278 or 0x3BC)
- If the device you're controlling operates at 3.3V (rather than 5V) move the "Vcc Select" jumper to the 3.3V position

3.0 Connections

- Connect a 5V power supply to the +5V and Gnd terminals. The red LED next to the terminal should light up. If not, verify that your power supply is working properly.
- Prepare the device under test (connect its power and any other pins that need to be connected to make it operational, such as Chip Select, Address lines etc.)

2-wire Bus

- Connect the CLK output of the ProgramIC board to the 2-wire Clock input pin on the chip (SCL pin).
- Connect the SDA output of the ProgramIC board to the 2-wire data input pin on the chip (SDA pin)

SPI Bus

- Connect the CLK output of the ProgramIC board to the SPI Clock input pin on the chip (CLK pin)
- Connect the SI output of the ProgramIC board to the SPI data input pin on the chip (SI pin).
- Connect the SO input of the ProgramIC board to the SPI data output pin on the chip (SO pin)

3-wire Bus

- Connect the CS output of the ProgramIC board to the CS input pin on the chip (CS pin)
- Connect the UD output of the ProgramIC board to the U/D data input pin on the chip (U/D pin).
- Connect the INC output of the ProgramIC board to the INC input pin on the chip (INC pin)

To connect to the device use the wire jump between corresponding pins of the JP1 header and device. See **Appendix 1** below for an example of how to setup the X9241A chip.

Additionally you may use the other available lines on the ProgramIC card to connect the address lines or chip select line on your device. Four address lines (A0 through A3) are available for use with 2-wire or SPI devices.

Note: For the Intersil X9252 or X9455 device, you must connect the following additional lines:

From Master (Intersil XLABVIEW01 Board)		To Slave (Xicor device)
A1	Connect to	DS1
A0	Connect to	DS0

The Potentiometer Address window in the GUI will now control these two bits, with A0 being the LSB.

4.0 Software Installation

4.1 Run the setup.exe from the CD-ROM you received. This will install the following files or programs:

- Xicor XDCP Test Program (Xicor.exe)
- Documentation (Xicor XDCP Test Utility Manual.doc)
- LabVIEW 6.0.2 Run-Time System
- LabWindows/CVI 5.5 Run-Time System
- Low Level LabVIEW driver for the XLABVIEW01 board
- Bus Level LabVIEW drivers for the XLABVIEW0board (2-wire and SPI)
- Chip Level LabVIEW drivers for the Intersil devices
- Xicor.ini configuration file

4.2 There is an upgrade available for recent Intersil devices, V3.2.5. It is contained in a ZIP file, "XDCP Update V3.2.5". This upgrade is installed on top of the original software program as follows:

4.2.1 Double click on the filename.

4.2.2 Click Extract on the menu bar.

4.2.3 Choose the folder C:\Program Files\Xicor.

4.2.4 Make sure the boxes "Overwrite Existing Files" and "Use Folder Names" are checked.

4.2.5 Click OK. The update files will automatically be loaded into the appropriate folders.

4.3 Click on the Xicor.exe file or the desktop shortcut to start the program.

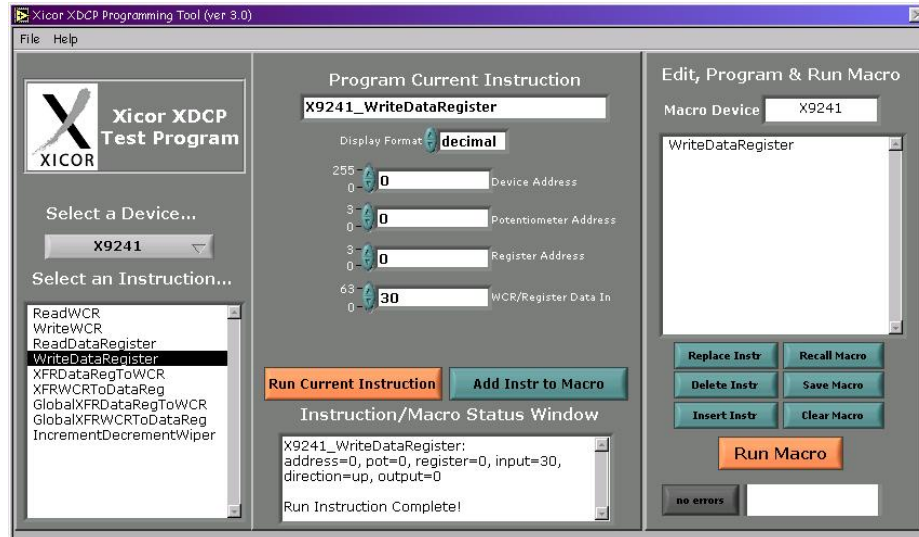
The XDCP Test Program Software supports the following devices:

3-wire devices	2-wire devices (I2C)	SPI devices
X9C102	X4023x	X9110
X9C103	X60250	X9111
X9C104	X9118	X9250
X9C303	X9119	X9251
X9C503	X9221A/41A	X9271
X9015	X9252	X9400
X9313	X9258/59	X9410
X9315	X9408/09	X9420
X9317	X9418/19	X9430
X9318	X9428	X9440
X9319	X9438	ISL224x6/9 family
X93154	X9448	ISL224x4 family
X90100	X9525	
ISL95310	X952x	
ISL95710	X95820/40	
ISL23710	ISL95810	
	ISL95311	
	ISL95711	
	ISL23711	
	ISL90460/1/2 (2-wire U/D)	
	ISL96017	
	ISL90726/7/8	
	ISL223x6/9 family	
	ISL223x3 family	

Check www.intersil.com for the latest release and updates to the LabVIEW drivers.

5.0 Using the Xicor XDCP Test Utility software

Run the *Intersil XDCP Test Utility* program. You will be asked first to select the address of the parallel port on your PC. It is usually set to 0x378, which is the Windows OS default. If you're not sure about the parallel port address, check the Device Manager in Windows.



1.0 Start by first selecting a device from the drop down box on the left.

2.0 As soon as you select a device, the program will list the available instructions for that device.

3.0 Select any of the instructions from the list. When you select an instruction, the **Program Current Instruction** window automatically displays the available inputs/output for that instruction.

4.0 Select any values for your input parameters and then run the instruction by clicking "Run Current Instruction" button. The Status Window below displays the status and shows the input/output values that were sent/received from the device. The information on the Status window is also saved in a file called "debug_out.txt" which resides in the same directory as the executable. The file is overwritten every time you run an instruction.

Note: The Status Window always displays a fixed list of parameters (address, pot, register, input, direction and output). Not all these parameters are used for a typical XDCP instruction. Ignore the parameters that you did not set for that particular instruction

5.0 On the right-hand side you have the option of building a macro. A macro consists of a list of instructions which will be executed in the order they are listed. A macro can be saved and recalled later. You can use this feature to build small programs that can be run over and over again. If you want to save the current instruction to the macro, click the "Add Instr to Macro" button. The instruction will be saved in the macro window on the right.

Note: The input parameters that you've selected for that instruction are also saved in the macro although only the instruction name is shown on the macro window. You can add as many instructions as you want to the macro. The only restriction is that you cannot mix commands from different devices in one macro. As soon as you try to select a new device, the program will ask you if you want to clear the current macro.

6.0 You can edit the macro by inserting, deleting and replacing instructions. When you select any of the instructions from the macro window, the **Program Current Instruction** window will automatically update

and display the parameters saved for that instruction. You can then modify any of these values and re-save the instruction with the new values.

7.0 When you're done with creating the macro, you can run the macro and verify the input/output parameters in the Status window. The Status window will show a list of all commands that were executed along with their input/output values. The information on the Status window is also saved in a file called "debug_out.txt" which resides in the same directory as the executable. The file is overwritten every time you run a macro or a single instruction.

8.0 You can save the macro and recall it later by selecting the "Save Macro" and "Recall Macro" buttons on the right-hand side. You can always use this program to recall and run an existing macro.

6.0 Special Device for Customizing Interface Signals

The X9XXX device can be selected from the pull-down menu and has the following options:

I2C_SendStop – an I2C stop sequence
I2C_SendStart – an I2C start sequence
I2C_GiveAcknowledge – master sends an acknowledge that the input data is received
I2C_ReceiveAcknowledge – master is checking for the acknowledge that the output data is received by the slave
I2C_SendByte – data output is the byte you want to send
I2C_ReceiveByte – data input is the byte read from the device
Set_Addr_Lines_A0-A3 – set the slave address lines either 0 (LOW) or 1 (HIGH)
Set_CS_Line – set the CS line either 0 (LOW) or 1 (HIGH)
Set_SO_Line – set the SO line either 0 (LOW) or 1 (HIGH)
Set_INC_Line – set the INC line either 0 (LOW) or 1 (HIGH)
Set_UD_line – set the UD line either 0 (LOW) or 1 (HIGH)
Set_CLK_Line – set the CLK line either 0 (LOW) or 1 (HIGH)
Set_SDA_Line – set the SDA line either 0 (LOW) or 1 (HIGH)
Read_SI_Line – the output data is either 0 (LOW) or 1 (HIGH)
Read_IN0_Line – the output data is either 0 (LOW) or 1 (HIGH)
Read_IN1_Line – the output data is either 0 (LOW) or 1 (HIGH)

These instructions can be used individually or combined in series using the Macros window to talk to virtually any device with serial bus communications. Note that the bus speed is limited by the parallel port and may be slower than the required minimum of some devices. It is possible to use the I2C and SPI commands to do "bit banging" to an IC if no other software tool is available.

7.0 Using the Xicor LabVIEW drivers in a custom application

In addition to the XDCP Test Utility program, the current release of the software offers a set of LabVIEW drivers that can be used to program the Intersil chips (or any 2-wire or SPI device) from a LabVIEW application.

Developers who would like to integrate the Intersil instructions into a larger, custom application (such as an automated test program or a calibration program) can use these drivers in the LabVIEW development environment (both LabVIEW 6 and LabVIEW 5.1.1 are supported in the current release)

Three levels of drivers are provided for this purpose:

1. Low Level

The Low Level drivers can be found in the "Low Level" directory under the program directory. You can use these drivers to toggle the ProgramIC lines high or low. Any of the input/output lines on the ProgramIC board can be controlled with these self-explanatory routines.

2. Bus Level

The Bus Level drivers can be found in the "Bus Level" directory under the program directory. You can use these drivers to **communicate with any 2-wire or SPI device**. The driver implements commands such as I2C_Write, I2C_Read, I2C_Acknowledge, SPI_Write and SPI_Read.

The commands are very flexible and can be used to send any number of bits and bytes to an SPI or 2-wire device.

3. Chip Level

The Chip Level drivers can be found in the "Chip Level" directory under the program directory. These drivers are designed specifically for the Intersil XDCP devices listed above. These drivers are called by the main XDCP Test Utility at run-time. Since most Intersil XDCP devices have very similar programming interfaces, many of them can share the same drivers. You will find a set of directories under the "Chip Level" directory named after each main Intersil device. Each directory contains the Intersil Instructions for that particular device and any other device that shares the same instructions.

For additional information on these drivers, contact Cal-Bay Systems at info@calbay.com. A separate manual describes the usage of these drivers in more detail.

8.0 Troubleshooting

Most problems encountered with the ProgramIC board are a result of the PC parallel port. Some parallel port configurations can block the driver from communicating with the device. If you're having trouble communicating with the 2-wire or SPI device verify the following:

1. First install a low-level hardware access driver (AccessHw.zip) which can be downloaded from the following web sites (a copy of this driver is also included with the Xicor XDCP Software Upgrade distribution)

<ftp://ftp.calbay.com/downloads/ProgramIC/>

or

<http://zone.ni.com/devzone/devzoneweb.nsf/opendoc?openagent&2851D0A06E3185BC8625683A000AC730&cat=034E59A8530E09DF862568900018707A>

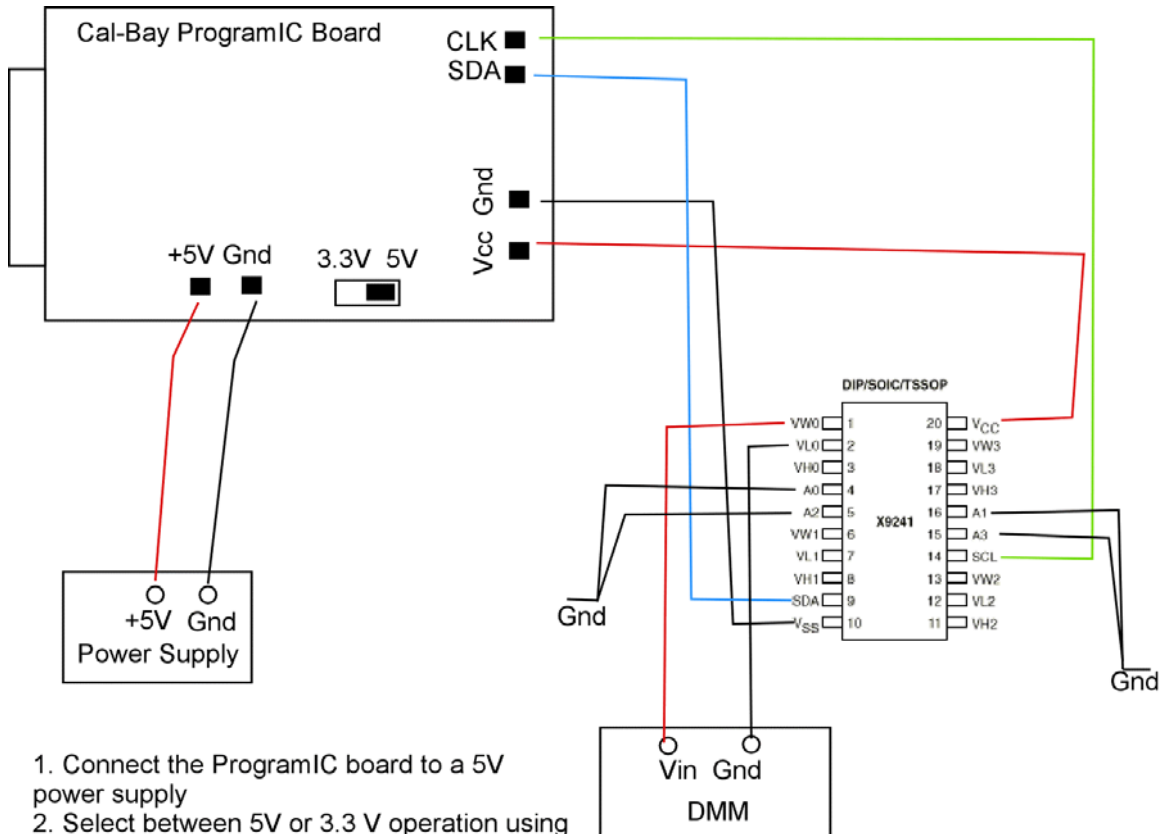
Download the AccessHw.zip file, unzip it and follow the instructions that come with it.

2. Using Device Manager on your PC check the parallel port address. Try changing the parallel port address to a different value.
3. Use the "Set Output Line.vi" in the LowLevel directory to toggle the clock line high or low. The LED at the top of the board should go on or off depending on the state of the clock line. If you can toggle this LED, then you're communicating successfully with the device.
4. If you still have difficulty communicating with the device, check the BIOS setup of your PC. Under the parallel port section you may see a setting for "Auto" or a specific value. Choose the "Auto" setting. Go back to step 2 and try the communication test again. If you still have communication problems, check the BIOS again for a setting called "Bi-Directional", "Compatible" or "Output Only". Try switching between these 3 settings and run the test on step 2 again.

5. Make sure you disable any printers that may be connected to the parallel port (verify this in Control Panel -> Printers)

If you are programming the Intersil XDCP using the low level drivers provided, and if you can access the parallel port, but still can't communicate with the device, try adding some delays in your LabVIEW program. Slowing down the execution of the instruction will prevent the software from violating the setup & hold times for the device.

Appendix 1: Xicor X9241A Chip (2-wire interface)



1. Connect the ProgramIC board to a 5V power supply
2. Select between 5V or 3.3 V operation using the jumper on the board
3. Connect power to the IC using the Vcc and Gnd outputs of the ProgramIC board
4. Connect CLK output to SCL input of the 9241 (green)
5. Connect SDA output to the SDA input of the 9241 (blue)
6. Connect the A0,A1,A2 and A3 inputs to any Ground in the system This will set the chip address to 0.
7. To test Potentiometer 0:
 - connect VL0 to Ground
 - connect VH0 to Vcc
 - connect DMM Voltage input to VW0
 - connect DMM Ground input to VL0

