ViperBoard User Guide

Nano River Technologies
October 2010

Table of Contents

Congratulations on the purchase of the ViperBoard from Nano River Technologies! We trust that this board will be a valuable asset in the development or test of your own electronics.

… Nano River Technologies development team

ABBREVIATIONS

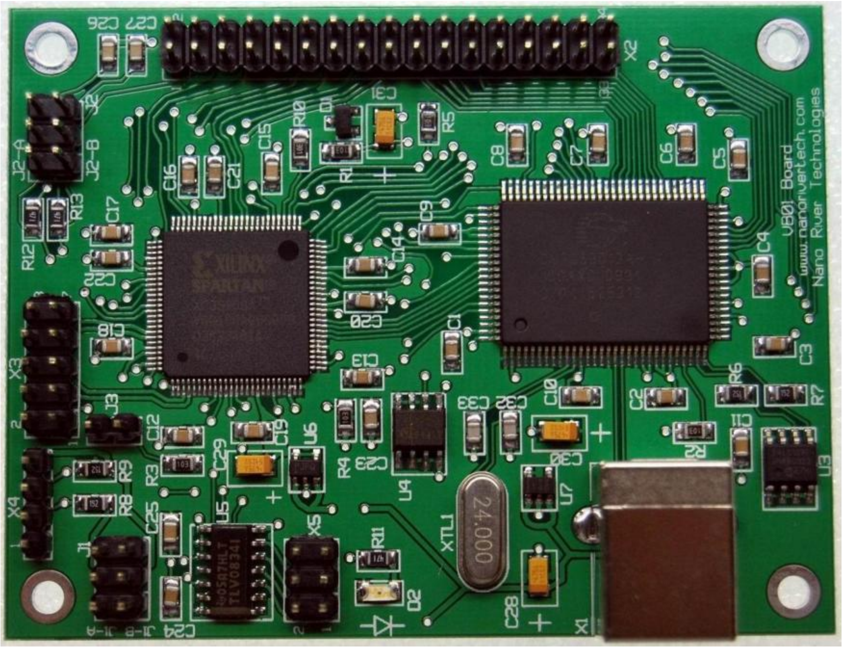
API	Application Programming Interface
EEPROM	Electrically Erasable Programmable Read Only Memory
G++	C++ compiler for Linux (part of GCC)
GCC	GNU Compiler Collection
GPIO	General Purpose IO
GPIOA	ViperBoard GPIO Port A (advanced GPIO interface)
GPIOB	ViperBoard GPIO Port B (digital IO interface)
GUI	Graphical User Interface
I2C	Inter-Integrated Circuit
IIC	Inter-Integrated Circuit (same as I2C)
IO	Input / Output
LED	Light Emitting Diode
Master	An interface which supplies the clock like the SPI master or I2C master on ViperBoard
NRT	Nano River Technologies
Slave	An interface which receives the clock like the SPI slave or I2C slave on ViperBoard
SPI	Serial Peripheral Interface
USB	Universal Serial Bus

1. Preliminaries

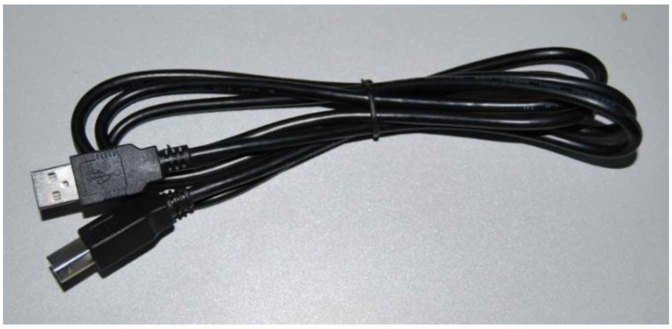
1.1. Packaging List

Your ViperBoard comes standard with the following items.

- One ViperBoard circuit board assembly



- One USB cable – type A to type B



PC software, example applications and documentation for use with this product is available downloadable from our web-site.

www.nanorivertech.com

1.2. Getting Help

Most questions relating to installation and usage of the ViperBoard are available from our web-site.

www.nanorivertech.com

If however you have further questions, then please do not hesitate to contact us via email.

support@nanorivertech.com

For any sales related questions, please contact:

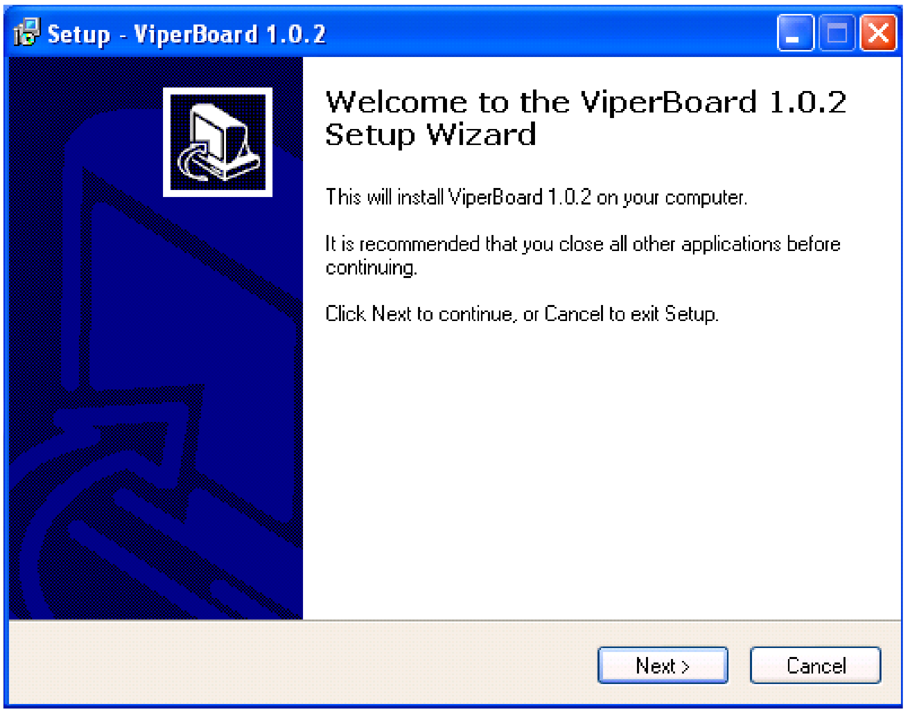
sales@nanorivertech.com

2. Installation (Windows 2000 and XP)

2.1. ViperBoard Installer

The ViperBoard installer activates from the software download page available from:

www.nanorivertech.com



Setup - ViperBoard 1.0.2

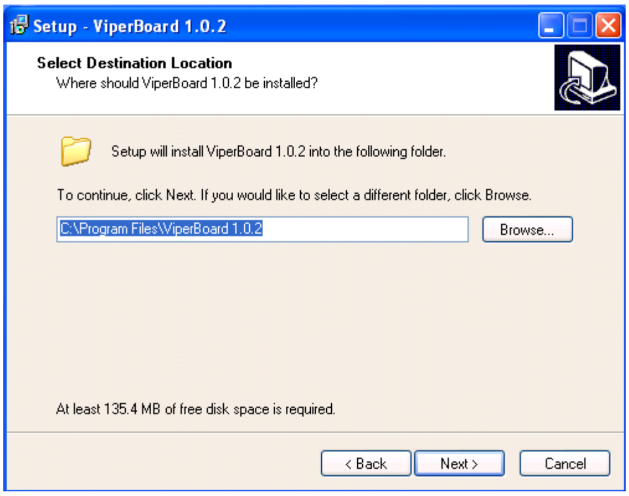
Welcome to the ViperBoard 1.0.2 Setup Wizard

This will install ViperBoard 1.0.2 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

The installer starts by downloading all ViperBoard documentation and software examples.


Setup - ViperBoard 1.0.2

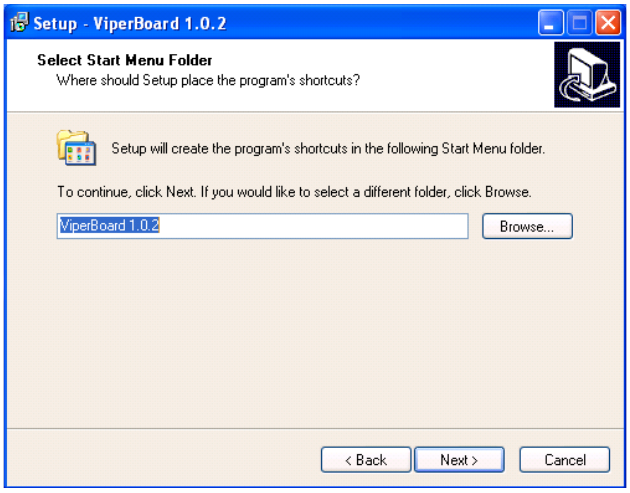
Select Destination Location
Where should ViperBoard 1.0.2 be installed?

Setup will install ViperBoard 1.0.2 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\ViperBoard 1.0.2

At least 135.4 MB of free disk space is required.

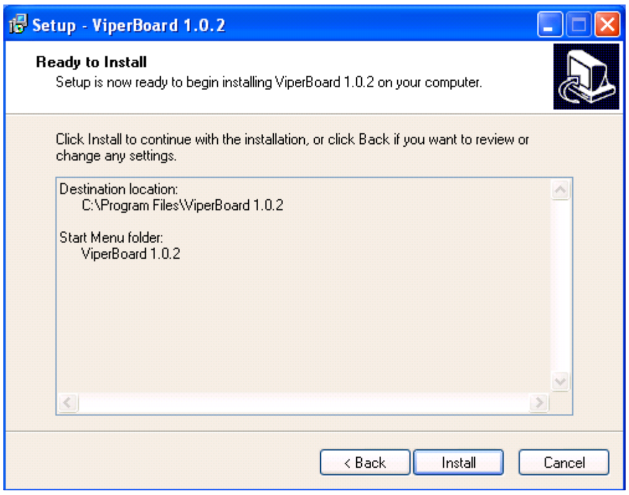

Setup - ViperBoard 1.0.2

Select Start Menu Folder
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

ViperBoard 1.0.2

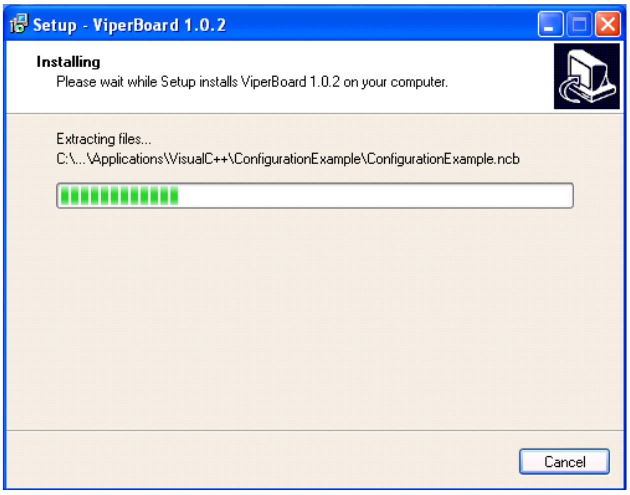

Setup - ViperBoard 1.0.2

Ready to Install
Setup is now ready to begin installing ViperBoard 1.0.2 on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files\ViperBoard 1.0.2

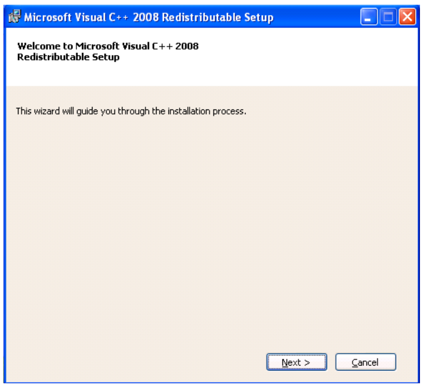
Start Menu folder:
ViperBoard 1.0.2


Setup - ViperBoard 1.0.2

Installing
Please wait while Setup installs ViperBoard 1.0.2 on your computer.

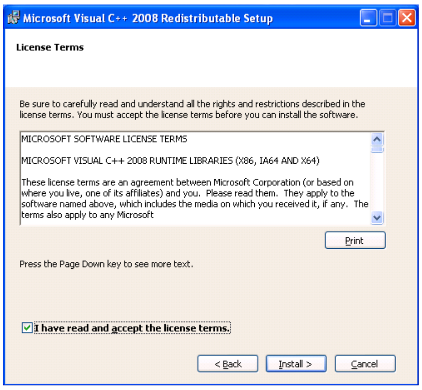
Extracting files...
C:\...\Applications\VisualC++\ConfigurationExample\ConfigurationExample.ncb

To run the applications without MicroSoft Visual C++ installed the Microsoft Visual C++ 2008 redistribution pack is required. This is installed automatically next. Allow all default directories and file names.



Welcome to Microsoft Visual C++ 2008 Redistributable Setup

This wizard will guide you through the installation process.



License Terms

Be sure to carefully read and understand all the rights and restrictions described in the license terms. You must accept the license terms before you can install the software.

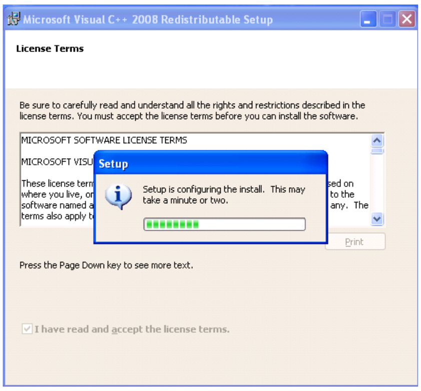
MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL C++ 2008 RUNTIME LIBRARIES (X86, IA64 AND X64)

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

Press the Page Down key to see more text.

☑ I have read and accept the license terms.



License Terms

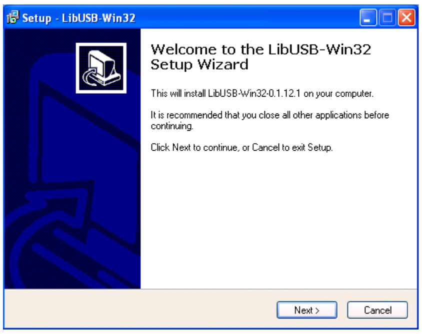
Be sure to carefully read and understand all the rights and restrictions described in the license terms. You must accept the license terms before you can install the software.

Setup is configuring the install. This may take a minute or two.

Press the Page Down key to see more text.

☑ I have read and accept the license terms.

Next the open source LibUSB-Win USB driver is installed. Allow all default directories and file names.

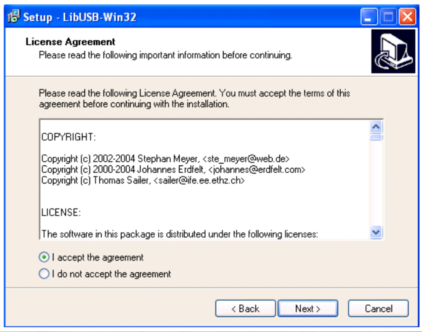


Welcome to the LibUSB-Win32 Setup Wizard

This will install LibUSB-Win32-0.1.12.1 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

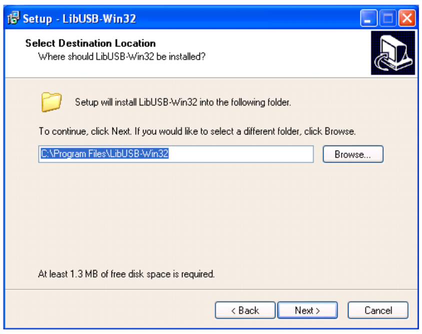
COPYRIGHT:

Copyright (c) 2002-2004 Stephan Meyer, <ste_meyer@web.de>
Copyright (c) 2000-2004 Johannes Erdfelt, <johannes@erdfelt.com>
Copyright (c) Thomas Sailer, <sailer@ife.ee.ethz.ch>

LICENSE:

The software in this package is distributed under the following licenses:

● I accept the agreement
○ I do not accept the agreement



Select Destination Location

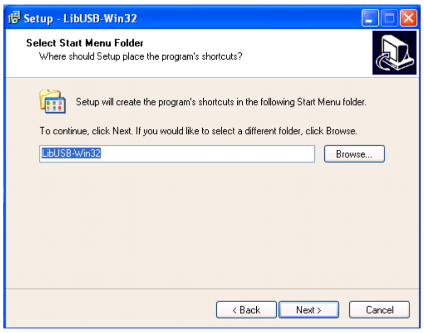
Where should LibUSB-Win32 be installed?

Setup will install LibUSB-Win32 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\LibUSB-Win32

At least 1.3 MB of free disk space is required.



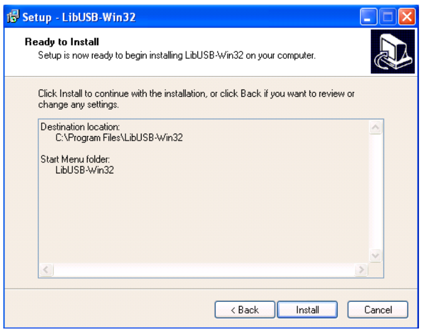
Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

LibUSB-Win32



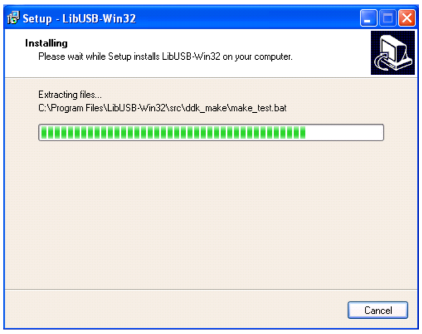
Ready to Install

Setup is now ready to begin installing LibUSB-Win32 on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files\LibUSB-Win32

Start Menu folder:
LibUSB-Win32

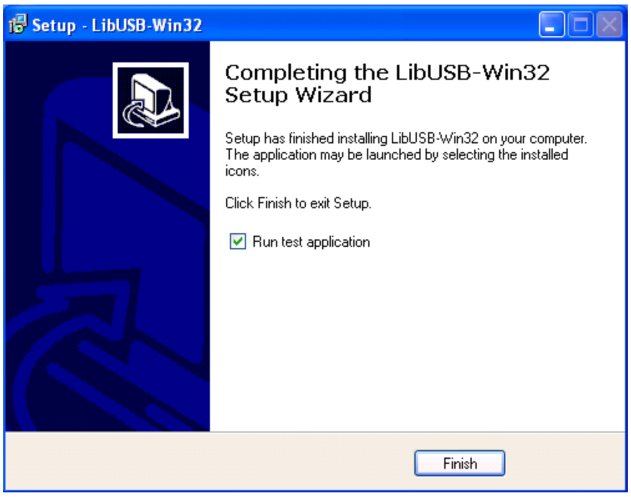


Installing

Please wait while Setup installs LibUSB-Win32 on your computer.

Extracting files...
C:\Program Files\LibUSB-Win32\src\ddk_make\make_test.bat

At the conclusion of the LibUSB-Win USB driver installation there will be the opportunity to run a test program to ensure it installed properly. Running this is optional.

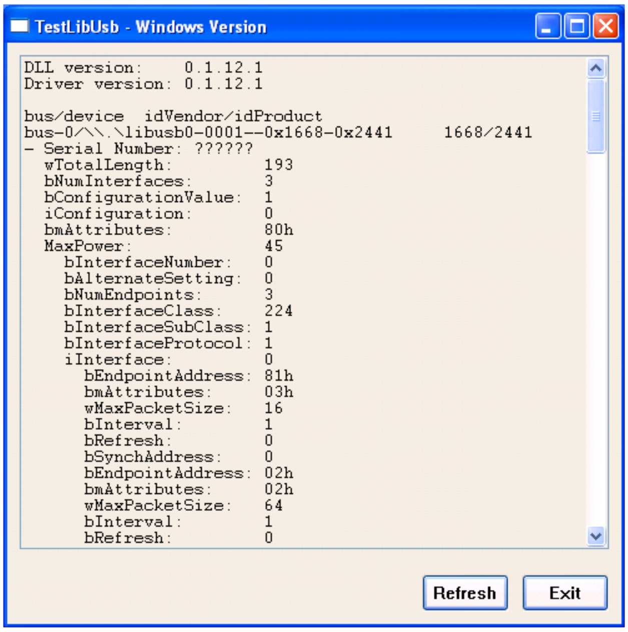


Completing the LibUSB-Win32 Setup Wizard

Setup has finished installing LibUSB-Win32 on your computer. The application may be launched by selecting the installed icons.

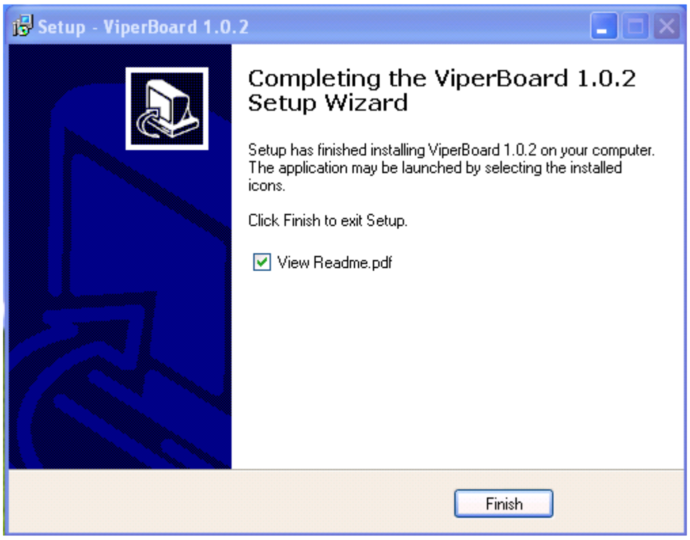
Click Finish to exit Setup.

☑ Run test application



```
DLL version:    0.1.12.1
Driver version: 0.1.12.1

bus/device  idVendor/idProduct
bus-0/\\.\libusb0-0001--0x1668-0x2441     1668/2441
- Serial Number: ??????
  wTotalLength:         193
  bNumInterfaces:       3
  bConfigurationValue:  1
  iConfiguration:       0
  bmAttributes:         80h
  MaxPower:             45
    bInterfaceNumber:   0
    bAlternateSetting:  0
    bNumEndpoints:      3
    bInterfaceClass:    224
    bInterfaceSubClass: 1
    bInterfaceProtocol: 1
    iInterface:         0
      bEndpointAddress: 81h
      bmAttributes:     03h
      wMaxPacketSize:   16
      bInterval:        1
      bRefresh:         0
      bSynchAddress:    0
      bEndpointAddress: 02h
      bmAttributes:     02h
      wMaxPacketSize:   64
      bInterval:        1
      bRefresh:         0
```

At the conclusion of the driver install, a Readme file will be displayed containing an overview of the ViperBoard product.



Completing the ViperBoard 1.0.2 Setup Wizard

Setup has finished installing ViperBoard 1.0.2 on your computer. The application may be launched by selecting the installed icons.

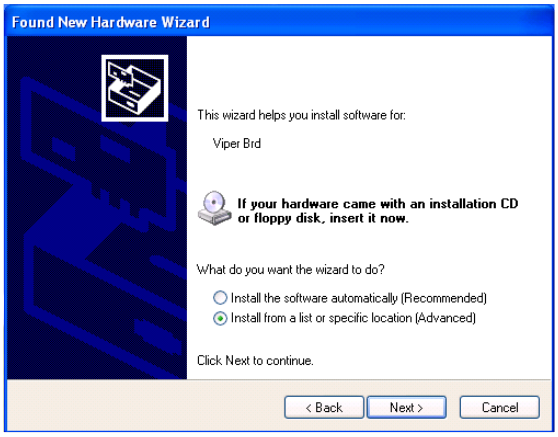
Click Finish to exit Setup.

☑ View Readme.pdf

2.2. Plugging in the ViperBoard

When the driver is installed and the ViperBoard plugged into a USB port for the first time, the ″Found New Hardware″ wizard will appear.

Select "Install from a list or specific location".



This wizard helps you install software for:

Viper Brd

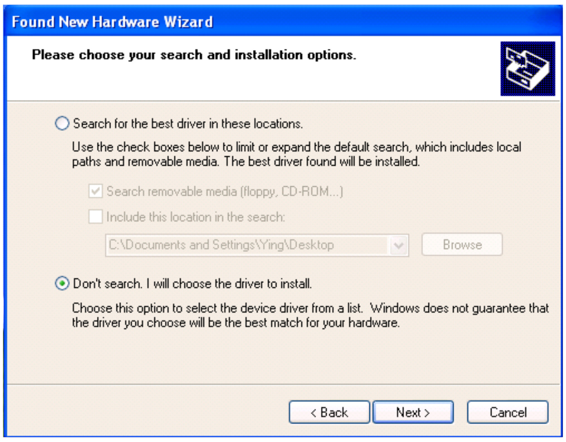
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)
● Install from a list or specific location (Advanced)

Click Next to continue.

Select ″Don′t search. I will choose the driver to install″.



Please choose your search and installation options.

○ Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

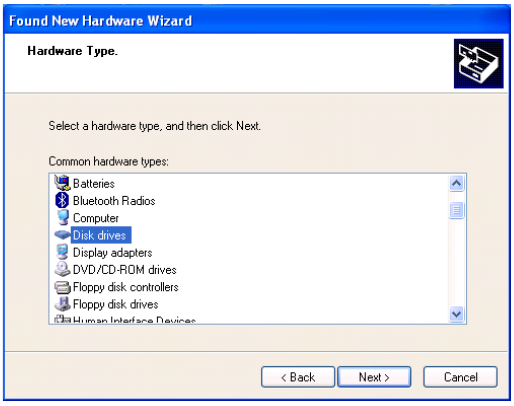
☑ Search removable media (floppy, CD-ROM...)
□ Include this location in the search:

C:\Documents and Settings\Ying\Desktop

● Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Navigate to the hard disk.

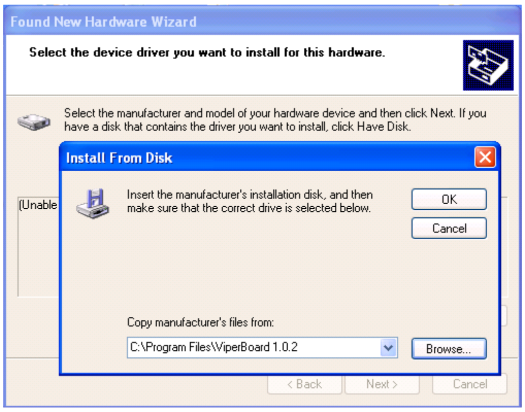

Found New Hardware Wizard

Hardware Type.

Select a hardware type, and then click Next.

Common hardware types:

Navigate to the install directory for ViperBoard eg `C:\Program Files\ViperBoardInstaller_1.0.2.`


Found New Hardware Wizard

Select the device driver you want to install for this hardware.

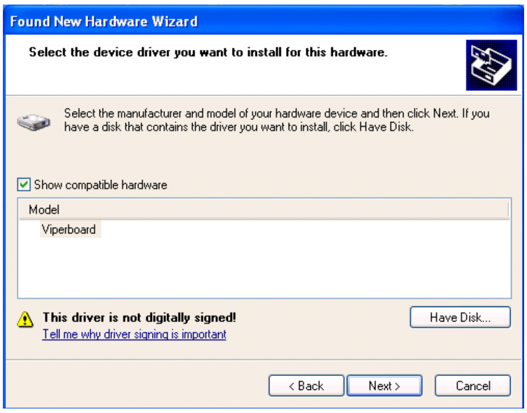
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Install From Disk

Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from:

C:\Program Files\ViperBoard 1.0.2


Found New Hardware Wizard

Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

☑ Show compatible hardware

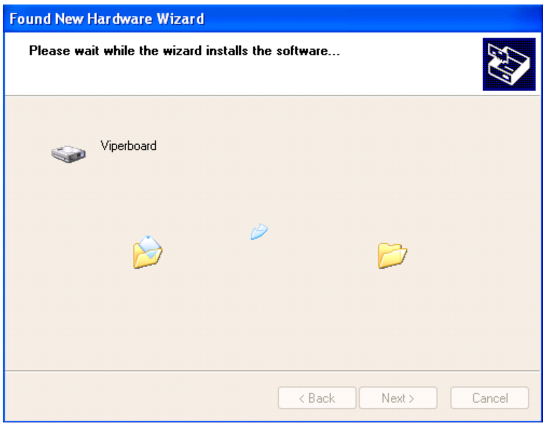
Model

Viperboard

This driver is not digitally signed!

Tell me why driver signing is important

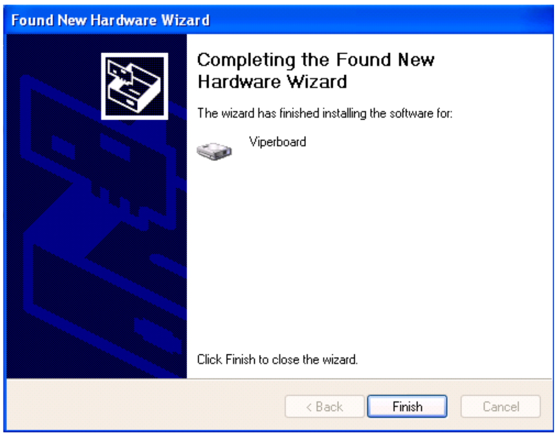
The Wizard will then find the driver setup file and start to link to the ViperBoard driver.


Found New Hardware Wizard

Please wait while the wizard installs the software...

Viperboard

Installation is then complete. The PC can now talk to a ViperBoard.


Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

Viperboard

Click Finish to close the wizard.

3. Installation (Linux)

Installation of ViperBoard under Linux consists of two steps. First OpenSource LibUsb for Linux is installed. Secondly the ViperBoard example applications and documentation is installed.

3.1. Step 1: LibUsb Installation

LibUsb is an OpenSource USB driver available for usage for commercial applications. The ViperBoard class library will make low level USB calls to LibUsb to implement the desired high level functions available through the ViperBoard API.

The LibUsb installation is contained within "libusb-0.1.12.tar.gz".

Installation consists of the following:

a) One should first ensure that GNU g++ is available on the Linux machine. This will be needed during the installation.

b) In a working directory untar the file using

```
linux%> gzip -cd libusb-0.1.12.tar.gz | tar xvf -
```

c) Move to the created directory using

```
linux%> cd libusb-0.1.12
```

d) Run the configure script

```
linux%> ./configure
```

e) Run make (become root if necessary) to build LibUsb

```
linux%> make
```

f) Install LibUsb

```
linux%> make install
```

At this point you should have installed LibUsb for Linux. Unfortunately in most cases it seems that the library can only be seen by the root user. This means that applications would only work as root. To fix this one needs to edit one of the rule files in UDEV.

g) Change to be root

```
linux%> sudo bash
```

h) Edit for example your `/etc/udev/rules.d/40-basic-permissions.rules` to include the following two new lines (in red). This makes the ViperBoard USB connection able to be seen by someone other than root.

```
# USB devices (usbfs replacement)
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", MODE="0664"
SUBSYSTEM=="usb_device",              MODE="0664"
#New lines for Viperboard
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device",SYSFS{idVendor}=="04b4",
SYSFS{idProduct}=="1004", MODE="0666"
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device",SYSFS{idVendor}=="2058",
SYSFS{idProduct}=="1005", MODE="0666"
```

The above change to permissions shows how to change USB permissions for Ubuntu Linux. For other kernels permissions may need to be changed in other files. For example in Federa 11, one needs to make the modification to `/lib/udev/rules.d/50-udev-default.rules`.

i) You can return from being root now and you are ready to install the ViperBoard examples and documentation.

3.2. Step 2: ViperBoard Examples and Documentation Installation

a) Documents and the example applications are all contained in ViperBoard_Installation_1.0.2.tar.gz. Untar this in some suitable place.

```
linux%> tar xvfz ViperBoard_Installation_1.0.2.tar.gz
```

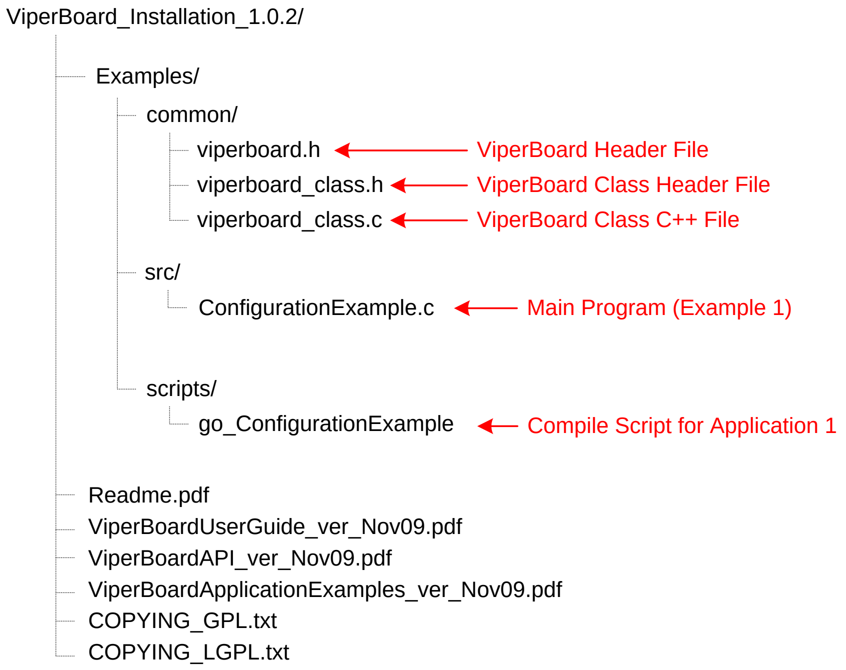
b) Move to the created directory.

```
linux%> cd ViperBoard_Installation_1.0.2
```

c) You will see the following documents to read:

Readme.pdf
ViperBoardUserGuide_ver_Nov09.pdf
ViperBoardAPI_ver_Nov09.pdf
ViperBoardApplicationExamples_ver_Nov09.pdf

d) You will find the application examples in the Examples/ directory. Read **ViperBoardApplicationExamples_ver_Nov09.pdf** to learn about running these. The file structure should appear as below.


ViperBoard_Installation_1.0.2/
Examples/
common/
viperboard.h
ViperBoard Header File
viperboard_class.h
ViperBoard Class Header File
viperboard_class.c
ViperBoard Class C++ File
src/
ConfigurationExample.c
Main Program (Example 1)
scripts/
go_ConfigurationExample
Compile Script for Application 1
Readme.pdf
ViperBoardUserGuide_ver_Nov09.pdf
ViperBoardAPI_ver_Nov09.pdf
ViperBoardApplicationExamples_ver_Nov09.pdf
COPYING_GPL.txt
COPYING_LGPL.txt

4. Installation (MacOS)

Installation of ViperBoard for MacOS consists of two steps. First OpenSource LibUsb for MacOS is installed. Secondly the ViperBoard example applications and documentation is installed.

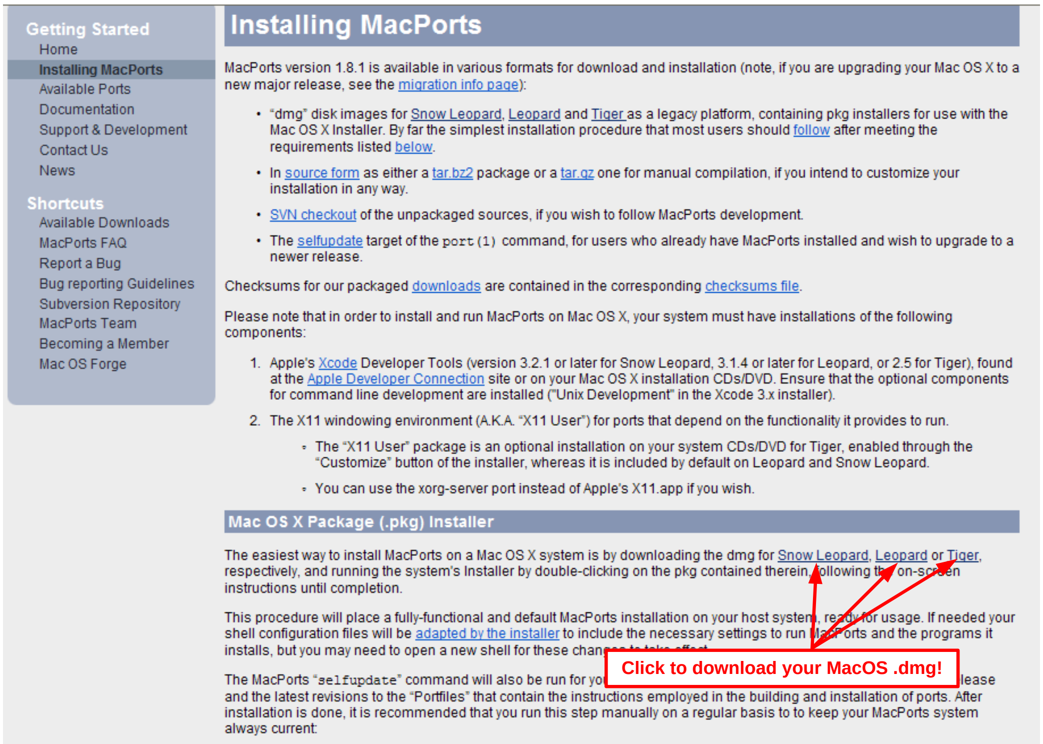
4.1. Step 1: LibUsb Installation

LibUSB 0.1.12 must be first installed under MacOS. The simplest way to achieve this is to make use of the installation already included in MacPorts.

First visit the MacPorts installation page:

http://www.macports.org/install.php

Download the MacPorts installation for your version of MacOS by selecting Snow Leopard, Leopard or Tiger. A .dmg file will be downloaded (it will be called something like `MacPorts-1.8.1-10.6-SnowLeopard.dmg`).



Installing MacPorts

MacPorts version 1.8.1 is available in various formats for download and installation (note, if you are upgrading your Mac OS X to a new major release, see the migration info page):

- "dmg" disk images for Snow Leopard, Leopard and Tiger as a legacy platform, containing pkg installers for use with the Mac OS X Installer. By far the simplest installation procedure that most users should follow after meeting the requirements listed below.
- In source form as either a tar.bz2 package or a tar.gz one for manual compilation, if you intend to customize your installation in any way.
- SVN checkout of the unpackaged sources, if you wish to follow MacPorts development.
- The selfupdate target of the `port(1)` command, for users who already have MacPorts installed and wish to upgrade to a newer release.

Checksums for our packaged downloads are contained in the corresponding checksums file.

Please note that in order to install and run MacPorts on Mac OS X, your system must have installations of the following components:

1. Apple's Xcode Developer Tools (version 3.2.1 or later for Snow Leopard, 3.1.4 or later for Leopard, or 2.5 for Tiger), found at the Apple Developer Connection site or on your Mac OS X installation CDs/DVD. Ensure that the optional components for command line development are installed ("Unix Development" in the Xcode 3.x installer).
2. The X11 windowing environment (A.K.A. "X11 User") for ports that depend on the functionality it provides to run.
   - The "X11 User" package is an optional installation on your system CDs/DVD for Tiger, enabled through the "Customize" button of the installer, whereas it is included by default on Leopard and Snow Leopard.
   - You can use the xorg-server port instead of Apple's X11.app if you wish.

Mac OS X Package (.pkg) Installer

The easiest way to install MacPorts on a Mac OS X system is by downloading the dmg for Snow Leopard, Leopard or Tiger, respectively, and running the system's Installer by double-clicking on the pkg contained therein, following the on-screen instructions until completion.

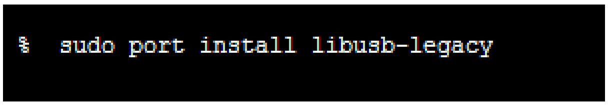
This procedure will place a fully-functional and default MacPorts installation on your host system, ready for usage. If needed your shell configuration files will be adapted by the installer to include the necessary settings to run MacPorts and the programs it installs, but you may need to open a new shell for these changes to take effect.

The MacPorts "`selfupdate`" command will also be run for yo lease and the latest revisions to the "Portfiles" that contain the instructions employed in the building and installation of ports. After installation is done, it is recommended that you run this step manually on a regular basis to to keep your MacPorts system always current:

Click to download your MacOS .dmg!

Double click on the downloaded file to mount it and execute the package file (.pkg) to install it.

In a command tool install version 0.1.12 of LibUsb using the following command:



```
%  sudo port install libusb-legacy
```

At this point you should have installed LibUsb for MacOS and you are ready to install the ViperBoard examples and documentation.

4.2. Step 2: ViperBoard Examples and Documentation Installation

a) Documents and the example applications are all contained in ViperBoard_Installation_1.0.2.tar.gz which comes from the ViperBoard software downloads page for MacOS. Untar this in some suitable place.

```
macos %> tar xvfz ViperBoard_Installation_1.0.2.tar.gz
```

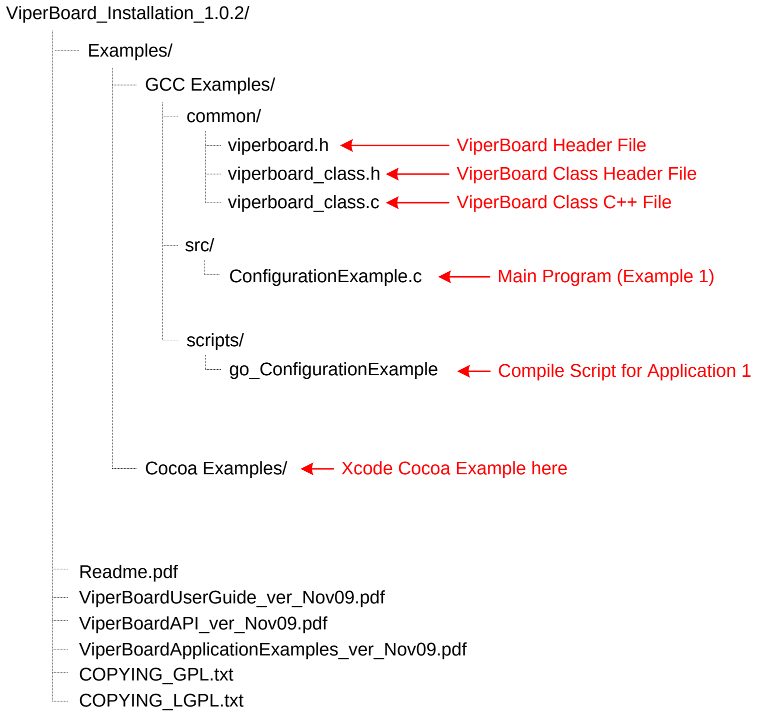
b) Move to the created directory.

```
macos %> cd ViperBoard_Installation_1.0.2
```

c) You will see the following documents to read:

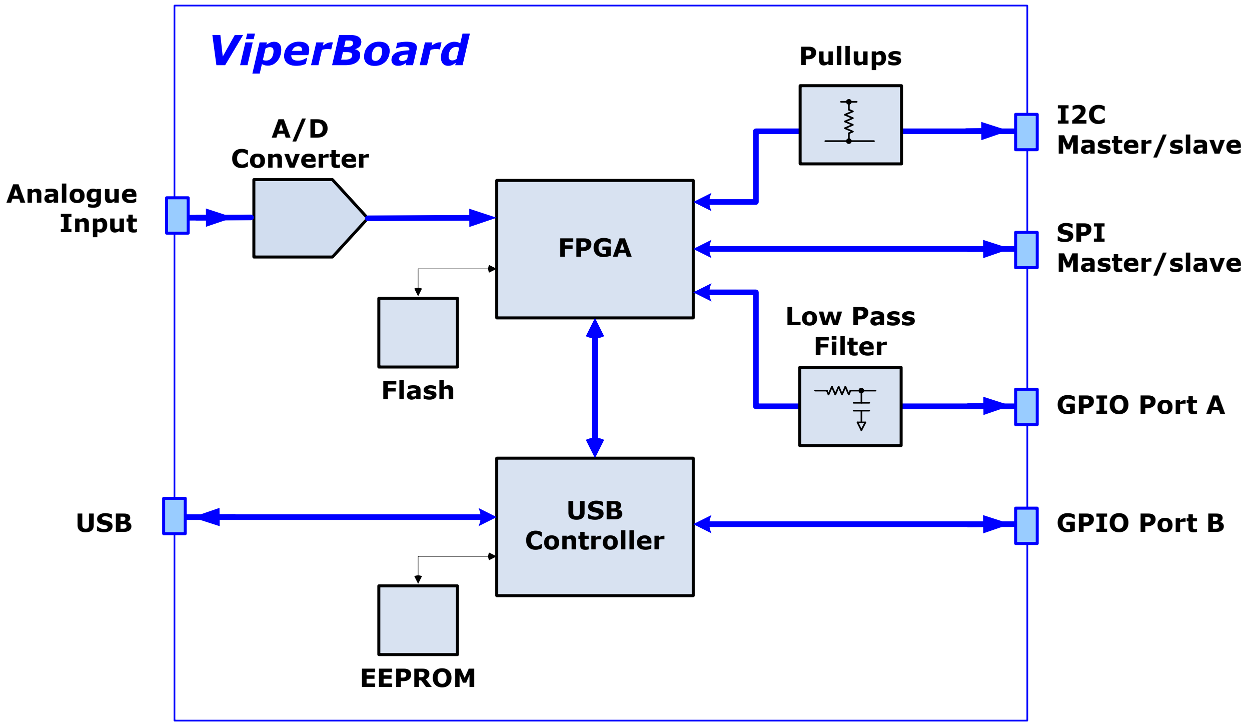
Readme.pdf
ViperBoardUserGuide_ver_Nov09.pdf
ViperBoardAPI_ver_Nov09.pdf
ViperBoardApplicationExamples_ver_Nov09.pdf

d) You will find the application examples in the Examples/ directory. Read **ViperBoardApplicationExamples_ver_Nov09.pdf** to learn about running these. The file structure should appear as below.



5. ViperBoard Hardware

5.1. Overview



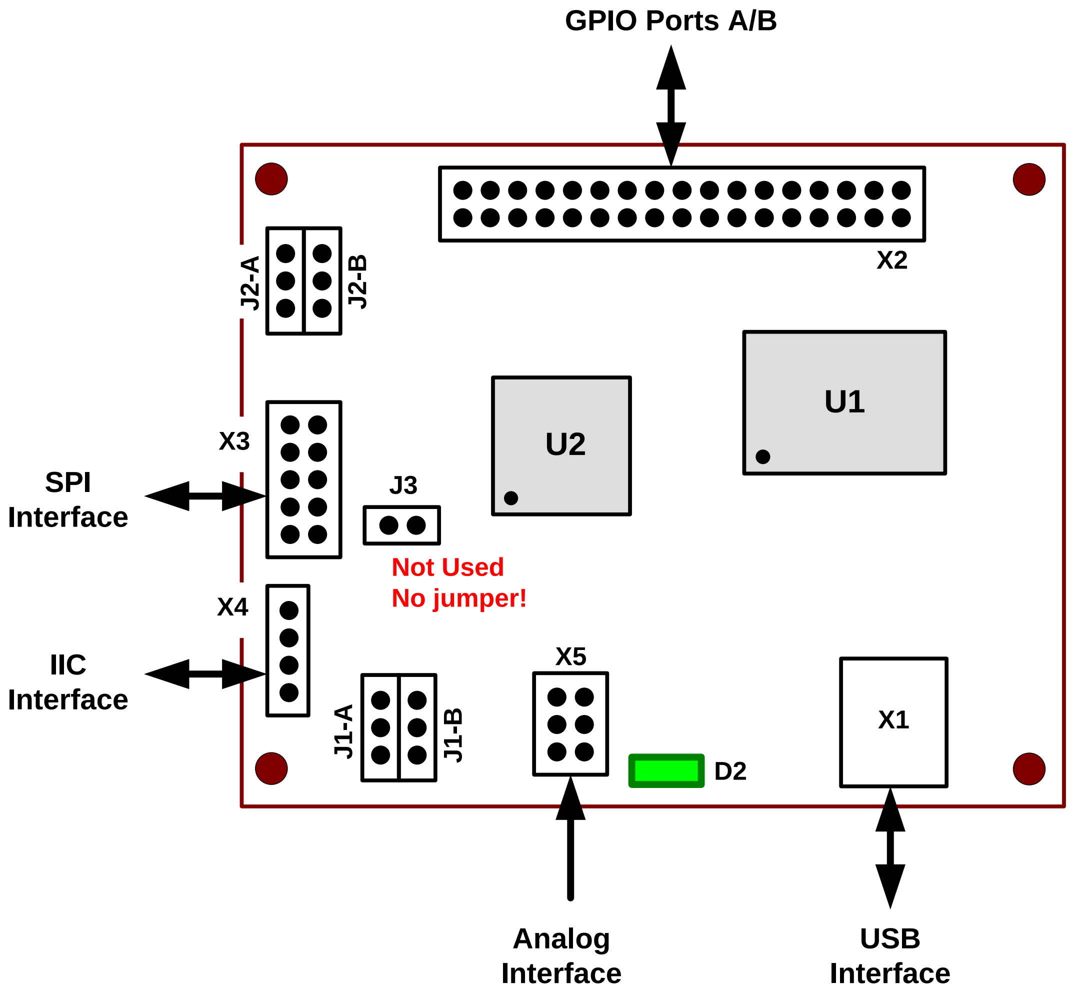
Viperboard is built around a Cypress USB controller chip and FPGA. It accepts commands across the USB bus and then translates these into I2C, SPI and GPIO bus protocols as required. Key features of the Viperboard include:

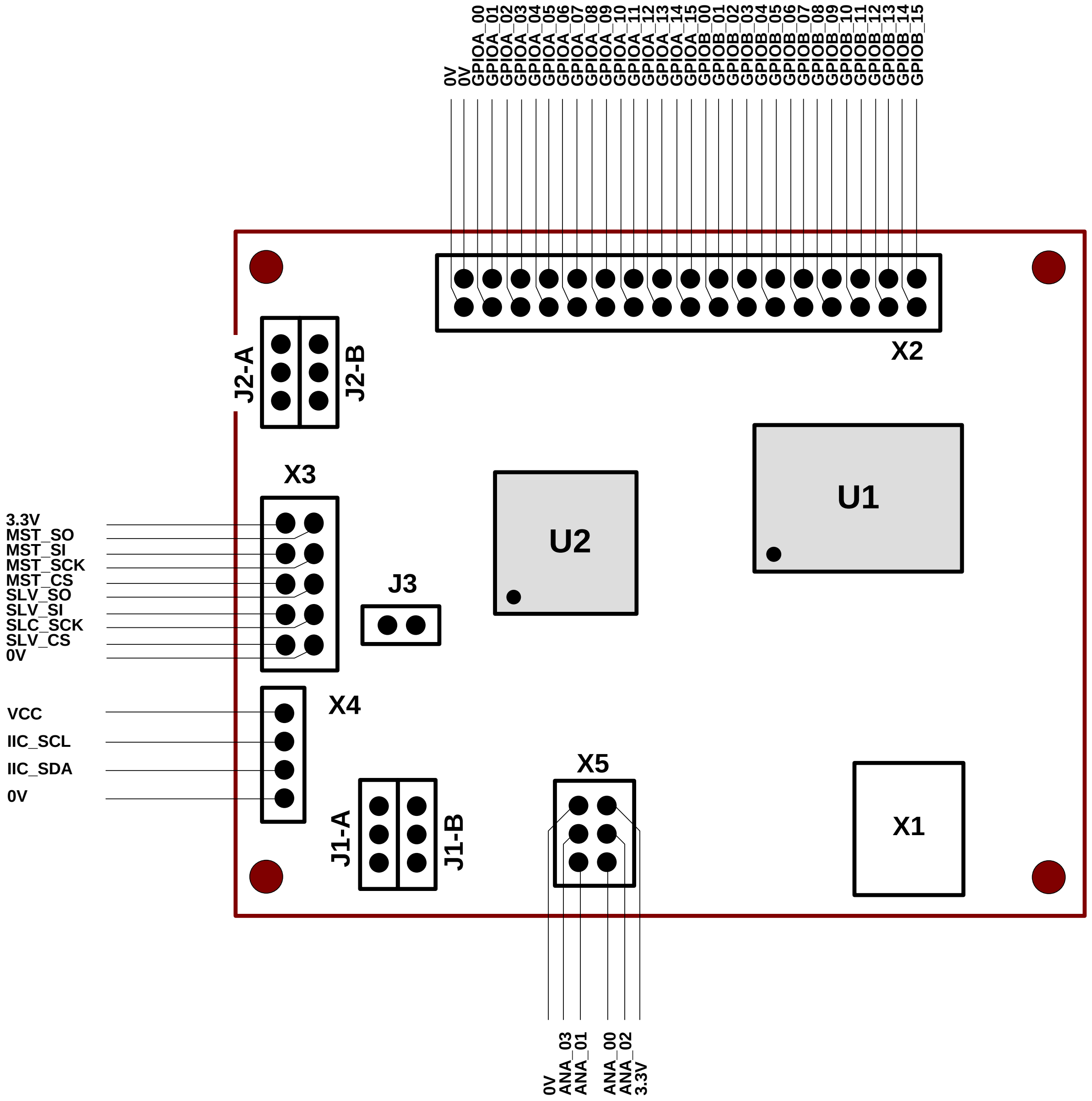
- high speed SPI master and slave (up to 17 channels)
- high speed I2C master and slave with optional pull-ups
- 16 bit GPIO Port A capable of digital I/O, interrupts, PWM
- optional low pass filtering for 2 GPIO Port A pins for analogue output
- 16 bit GPIO Port B capable of digital I/O
- 4 bit analogue input

A full schematic for the Viperboard can be seen in Appendix A.

5.2. Interfaces

Separate connectors are provided for each of the I2C, SPI, GPIO and Analogue interfaces. With pins as specified in the following diagrams.




0V
0V
GPIOA_00
GPIOA_01
GPIOA_02
GPIOA_03
GPIOA_04
GPIOA_05
GPIOA_06
GPIOA_07
GPIOA_08
GPIOA_09
GPIOA_10
GPIOA_11
GPIOA_12
GPIOA_13
GPIOA_14
GPIOA_15
GPIOB_00
GPIOB_01
GPIOB_02
GPIOB_03
GPIOB_04
GPIOB_05
GPIOB_06
GPIOB_07
GPIOB_08
GPIOB_09
GPIOB_10
GPIOB_11
GPIOB_12
GPIOB_13
GPIOB_14
GPIOB_15
X2
J2-A
J2-B
X3
U1
U2
J3
3.3V
MST_SO
MST_SI
MST_SCK
MST_CS
SLV_SO
SLV_SI
SLC_SCK
SLV_CS
0V
VCC
IIC_SCL
IIC_SDA
0V
X4
X5
J1-A
J1-B
X1
0V
ANA_03
ANA_01
ANA_00
ANA_02
3.3V

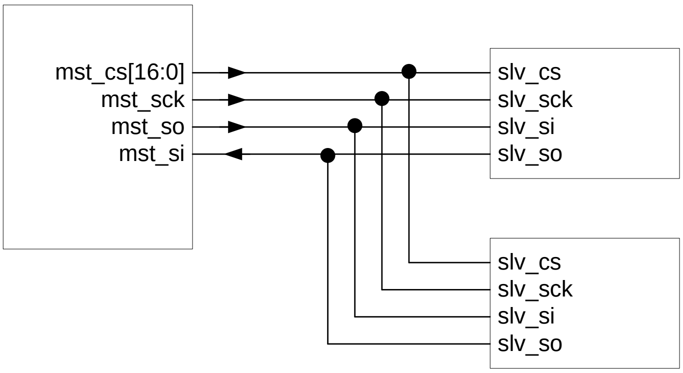
5.3. SPI Interface

Connector X3 contains the SPI interface.

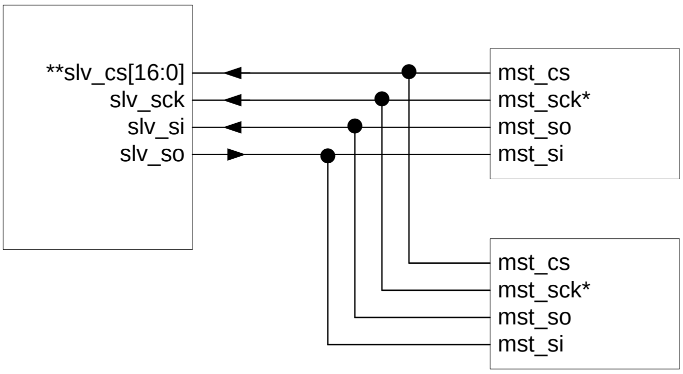
MST_CS[0]	Output	SPI master chip select output (channel 0).
MST_SCK	Output	SPI master serial clock output.
MST_SO	Output	SPI master serial data output.
MST_SI	Input	SPI master serial data input.
SLV_CS[0]	Input	SPI slave chip select input (channel 0).
SLV_SCK	Input	SPI slave serial clock input.
SLV_SO	Output	SPI slave serial data output.
SLV_SI	Input	SPI slave serial data input.
3.3V	Power	Supply
0V	Power	Ground

The interface includes separate master and slave SPI interfaces. Pins of the GPIOA port can be sacrificed to either SPI master or SPI slave chip selects. These then form the remaining bits of the chip select buses mst_cs[16:0] or slv_cs[16:0]. In general three usages of the SPI interface become possible, as shown over:

Master Mode



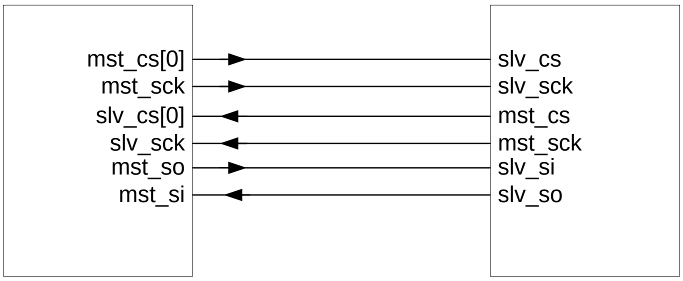
Slave Mode



*) When connecting multiple slaves, care needs to be taken to make sure the clock outputs can be connected. For example they could be externally gated or made open drain.

**) Always make sure that for channels configured to be SPI slaves always have their chip select inputs driven and not left floating.

Hybrid Master-Slave Mode

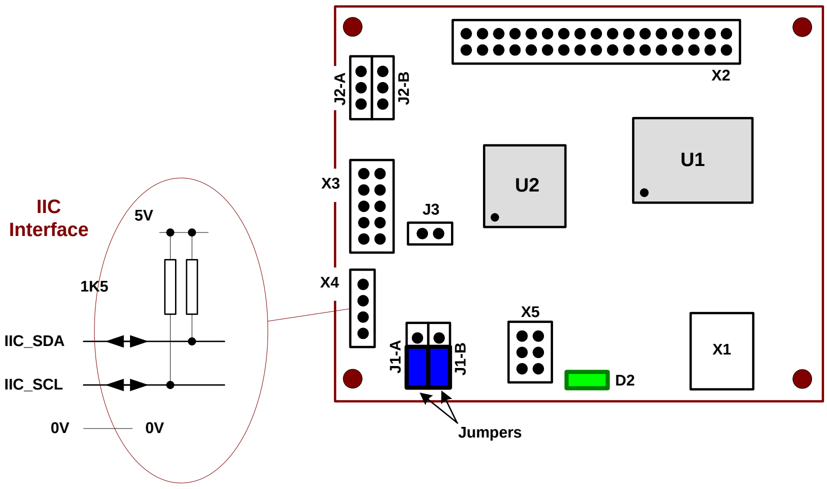


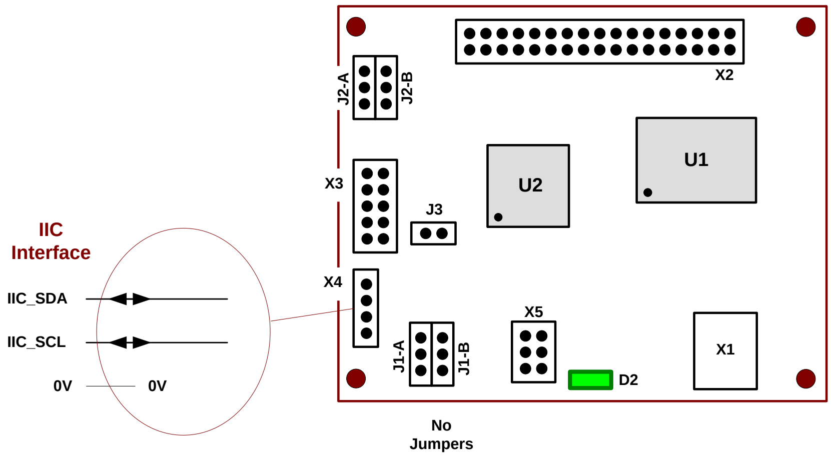
5.4. IIC Interface

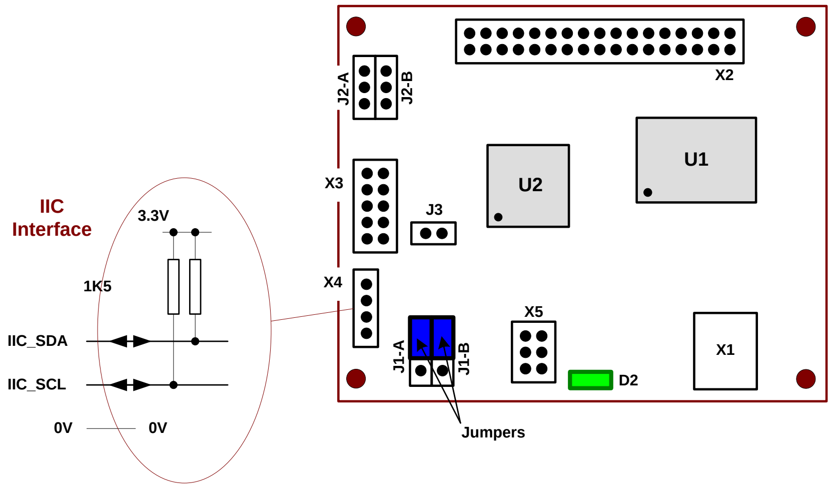
Connector X4 contains the I2C interface.

IIC_SCL	Input/Output	IIC serial clock.
IIC_SDA	Input/Output	IIC serial clock.

Configuration jumpers J1-A and J1-B can be used to switch in/out on-board pull-up resistors for the I2C lines up to either 3.3 volts or 5 volts as per the following drawings.


IIC
Interface
5V
1K5
IIC_SDA
IIC_SCL
0V
0V
X2
J2-A
J2-B
X3
J3
U2
U1
X4
X5
J1-A
J1-B
D2
X1
Jumpers


IIC
Interface
IIC_SDA
IIC_SCL
0V
0V
X2
J2-A
J2-B
X3
J3
U2
U1
X4
X5
J1-A
J1-B
D2
X1
No
Jumpers


IIC
Interface
3.3V
1K5
IIC_SDA
IIC_SCL
0V
0V
X2
J2-A
J2-B
X3
J3
U2
U1
X4
X5
J1-A
J1-B
D2
X1
Jumpers

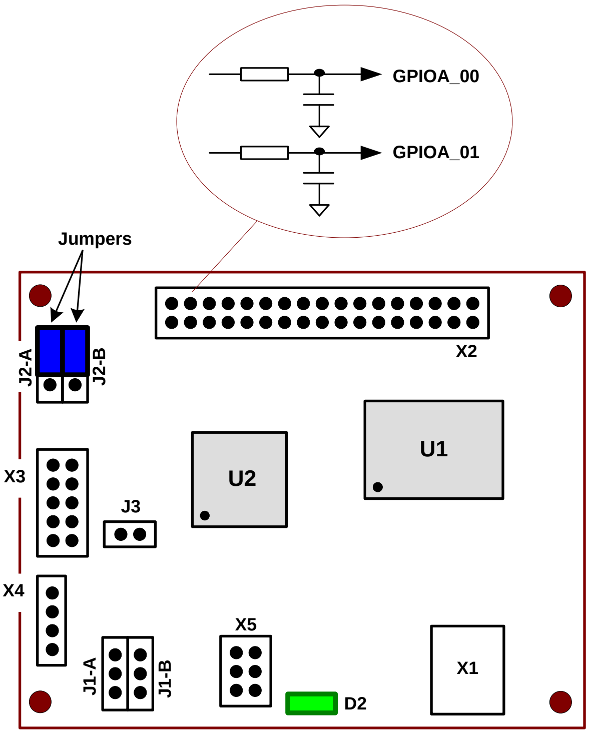
5.5. GPIO Port A Interface

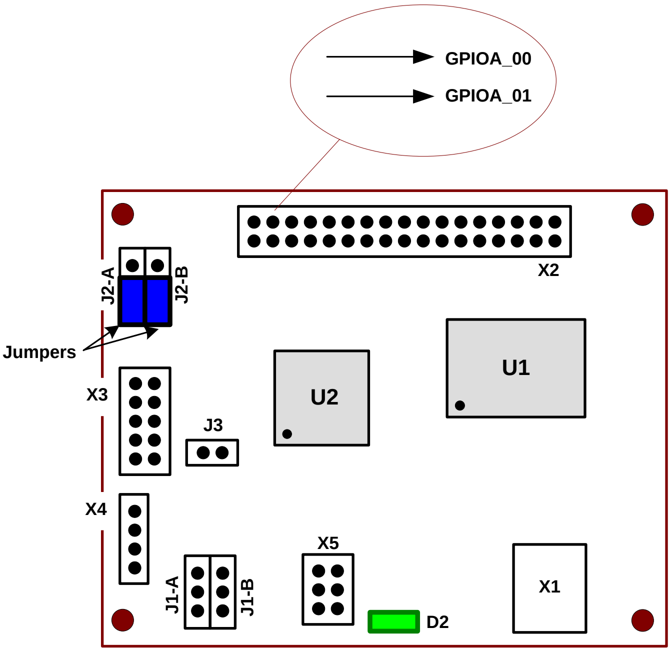
Connector X2 contains GPIO Port A interface.

GPIOA_00	Input/Output	Digital I/O, PWM, interrupt or pulsed. Can be filtered to form an analogue output.
GPIOA_01	Input/Output	Digital I/O, PWM, interrupt or pulsed. Can be filtered to form an analogue output.
GPIOA_02	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_03	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_04	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_05	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_06	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_07	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_08	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_09	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_10	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_11	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_12	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_13	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_14	Input/Output	Digital I/O, PWM, interrupt or pulsed.
GPIOA_15	Input/Output	Digital I/O, PWM, interrupt or pulsed.
0V	Supply	

The GPIO pins can all be separately configured as digital input, digital output, PWM outputs, pulsed output or interrupt input. These pins can also be sacrificed as SPI chip selects as either a SPI master or SPI slave as described in section 5.3.

For GPIOA bits 0 and 1 in PWM mode the output can be low pass filtered to form an analogue output using jumpers J2-A and J2-B.





5.6. GPIO Port B Interface

Connector X2 contains GPIO Port B interface.

GPIOB_00	Input/Output	Digital I/O.
GPIOB _01	Input/Output	Digital I/O.
GPIOB _02	Input/Output	Digital I/O.
GPIOB _03	Input/Output	Digital I/O.
GPIOB _04	Input/Output	Digital I/O.
GPIOB _05	Input/Output	Digital I/O.
GPIOB _06	Input/Output	Digital I/O.
GPIOB _07	Input/Output	Digital I/O.
GPIOB _08	Input/Output	Digital I/O.
GPIOB _09	Input/Output	Digital I/O.
GPIOB _10	Input/Output	Digital I/O.
GPIOB _11	Input/Output	Digital I/O.
GPIOB _12	Input/Output	Digital I/O.
GPIOB _13	Input/Output	Digital I/O.
GPIOB _14	Input/Output	Digital I/O.
GPIOB _15	Input/Output	Digital I/O.
0V	Supply	

5.7. Analogue Interface

Connector X5 the analogue interface.

ANA_00	Input	Analogue input.
ANA_01	Input	Analogue input.
ANA_02	Input	Analogue input.
ANA_03	Input	Analogue input.
3.3V	Supply	
0V	Supply	

5.8. LED Indicator

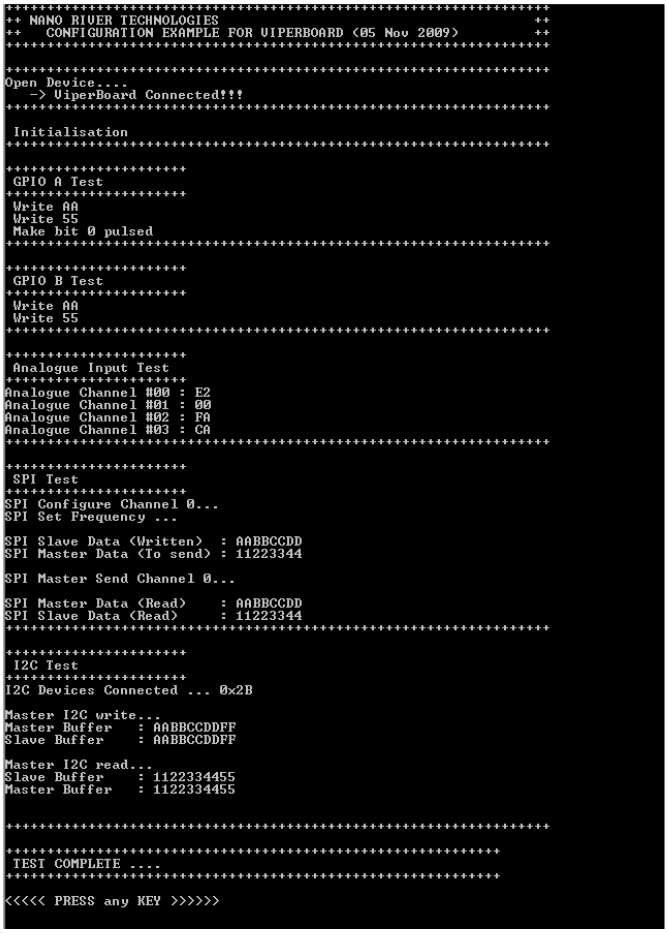
The ViperBoard includes a green LED indicator to indicate that the circuit is powered correctly.

6. Getting Started

Use of the ViperBoard typically involves writing a C/C++ console application or a Visual C++ Windows GUI application which calls functions and tasks in the ViperBoard API provided by Nano River Technologies. A full definition of the API is provided in the **"ViperBoard API Specification",** provided on the web-site.

In order to rapidly build up applications and see how to link to the API, eight example applications have been built up. The example applications provided are described below. These are all fully documented in the **"ViperBoard Example Applications"** document, provided on the web-site.

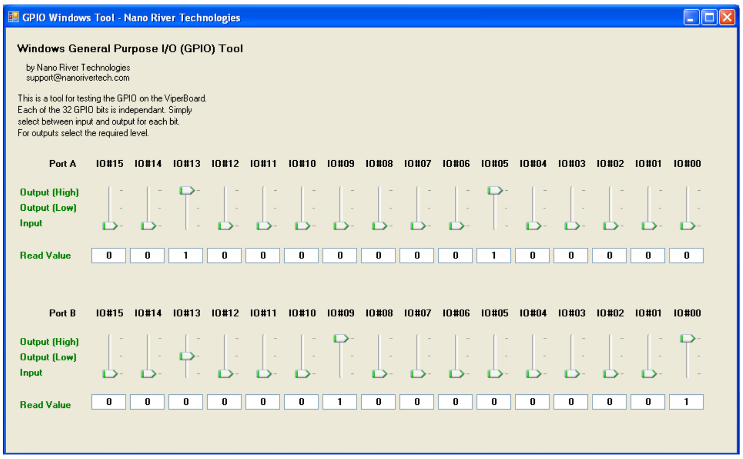
Configuration Example (for Windows)



```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++ NANO RIVER TECHNOLOGIES                                      ++
++   CONFIGURATION EXAMPLE FOR VIPERBOARD (05 Nov 2009)         ++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Open Device....
   -> ViperBoard Connected!!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 Initialisation
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 GPIO A Test
++++++++++++++++++++++
 Write AA
 Write 55
 Make bit 0 pulsed
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 GPIO B Test
++++++++++++++++++++++
 Write AA
 Write 55
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 Analogue Input Test
++++++++++++++++++++++
Analogue Channel #00 : E2
Analogue Channel #01 : 00
Analogue Channel #02 : FA
Analogue Channel #03 : CA
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 SPI Test
++++++++++++++++++++++
SPI Configure Channel 0...
SPI Set Frequency ...

SPI Slave Data (Written)  : AABBCCDD
SPI Master Data (To send) : 11223344

SPI Master Send Channel 0...

SPI Master Data (Read)    : AABBCCDD
SPI Slave Data (Read)     : 11223344
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 I2C Test
++++++++++++++++++++++
I2C Devices Connected ... 0x2B

Master I2C write...
Master Buffer   : AABBCCDDFF
Slave Buffer    : AABBCCDDFF

Master I2C read...
Slave Buffer    : 1122334455
Master Buffer   : 1122334455


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 TEST COMPLETE ....
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

<<<<< PRESS any KEY >>>>>>
```

This example is very simple C/C++ console application for Windows showing how to call basic functions associated with SPI, I2C, GPIO and analogue IO. The example is a good starting point if you want to try the ViperBoard for the first time and want to see a simple console application.

Windows GPIOTool (for Windows)



GPIO Windows Tool - Nano River Technologies

Windows General Purpose I/O (GPIO) Tool

by Nano River Technologies
support@nanorivertech.com

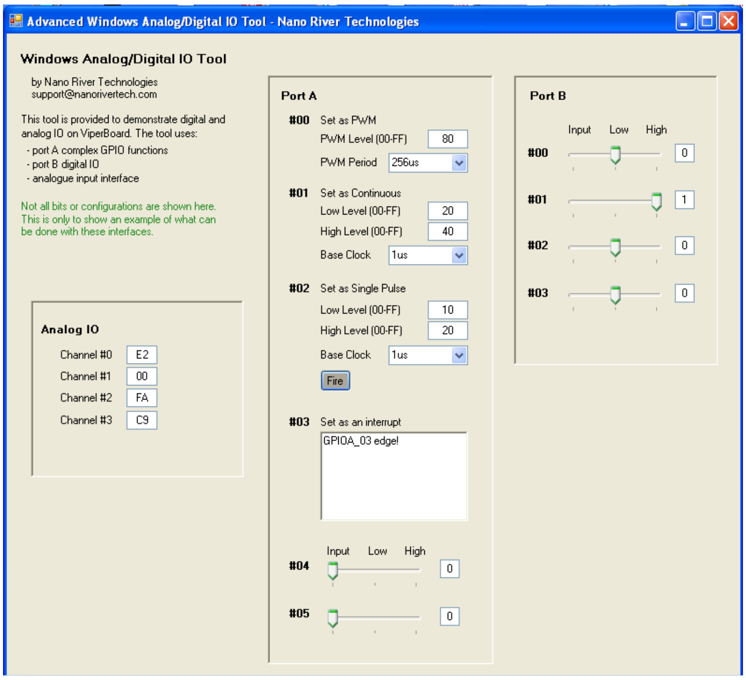
This is a tool for testing the GPIO on the ViperBoard.
Each of the 32 GPIO bits is independant. Simply
select between input and output for each bit.
For outputs select the required level.

Port A	IO#15	IO#14	IO#13	IO#12	IO#11	IO#10	IO#09	IO#08	IO#07	IO#06	IO#05	IO#04	IO#03	IO#02	IO#01	IO#00
Read Value	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Port B	IO#15	IO#14	IO#13	IO#12	IO#11	IO#10	IO#09	IO#08	IO#07	IO#06	IO#05	IO#04	IO#03	IO#02	IO#01	IO#00
Read Value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

This example application is a very simple Visual C++ GUI application showing how to interface to the general purpose IO (GPIO) pins. This includes both port A and port B GPIOs.

Advanced Windows Analog/Digital IO Tool



Advanced Windows Analog/Digital IO Tool - Nano River Technologies

Windows Analog/Digital IO Tool

by Nano River Technologies
support@nanorivertech.com

This tool is provided to demonstrate digital and analog IO on ViperBoard. The tool uses:
- port A complex GPIO functions
- port B digital IO
- analogue input interface

Not all bits or configurations are shown here. This is only to show an example of what can be done with these interfaces.

Analog IO

Channel #0	E2
Channel #1	00
Channel #2	FA
Channel #3	C9

Port A

#00 Set as PWM
PWM Level (00-FF) 80
PWM Period 256us

#01 Set as Continuous
Low Level (00-FF) 20
High Level (00-FF) 40
Base Clock 1us

#02 Set as Single Pulse
Low Level (00-FF) 10
High Level (00-FF) 20
Base Clock 1us

#03 Set as an interrupt
GPIOA_03 edge!

#04 0
#05 0

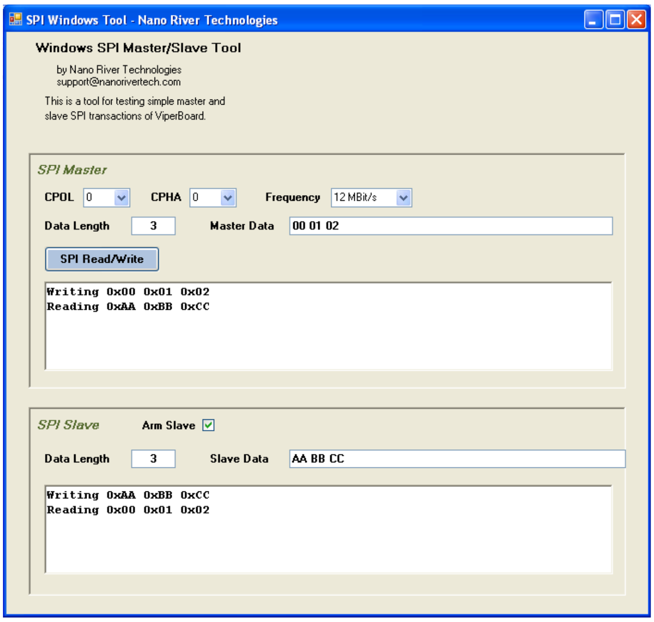
Port B

#00 0
#01 1
#02 0
#03 0

This is a more complex Visual C++ GUI application showing some of the more complex GPIO and analog input functions.

Windows SPI Tool

The Windows SPI tool is a windows GUI application in Visual C++ used to demonstrate both master and slave operation of the SPI interface for ViperBoard.



SPI Windows Tool - Nano River Technologies

Windows SPI Master/Slave Tool

by Nano River Technologies
support@nanorivertech.com

This is a tool for testing simple master and slave SPI transactions of ViperBoard.

SPI Master

CPOL: 0 · CPHA: 0 · Frequency: 12 MBit/s

Data Length: 3 · Master Data: 00 01 02

```
Writing 0x00 0x01 0x02
Reading 0xAA 0xBB 0xCC
```

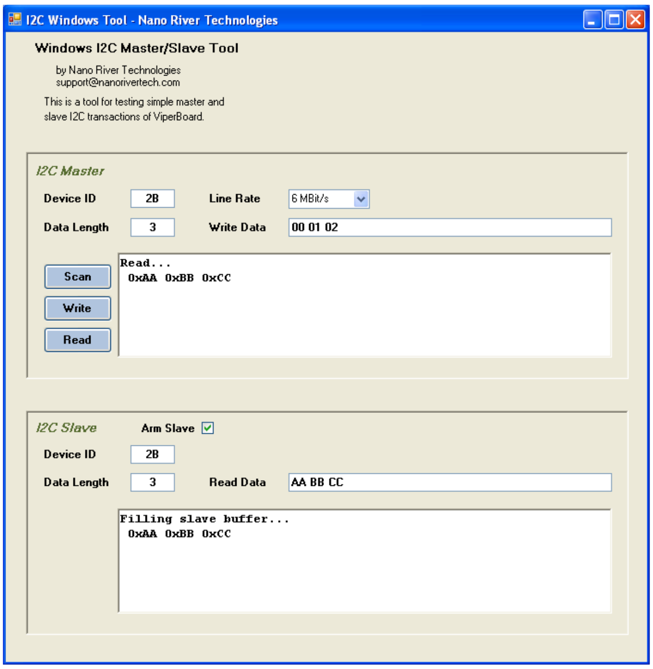
SPI Slave · Arm Slave ☑

Data Length: 3 · Slave Data: AA BB CC

```
Writing 0xAA 0xBB 0xCC
Reading 0x00 0x01 0x02
```

Windows I2C Tool

The Windows I2C tool is a windows GUI application in Visual C++ used to demonstrate both master and slave operation of the I2C interface for ViperBoard.



I2C Windows Tool - Nano River Technologies

Windows I2C Master/Slave Tool

by Nano River Technologies
support@nanorivertech.com

This is a tool for testing simple master and slave I2C transactions of ViperBoard.

I2C Master

Device ID: 2B · Line Rate: 6 MBit/s

Data Length: 3 · Write Data: 00 01 02

```
Read...
 0xAA 0xBB 0xCC
```

I2C Slave · Arm Slave ☑

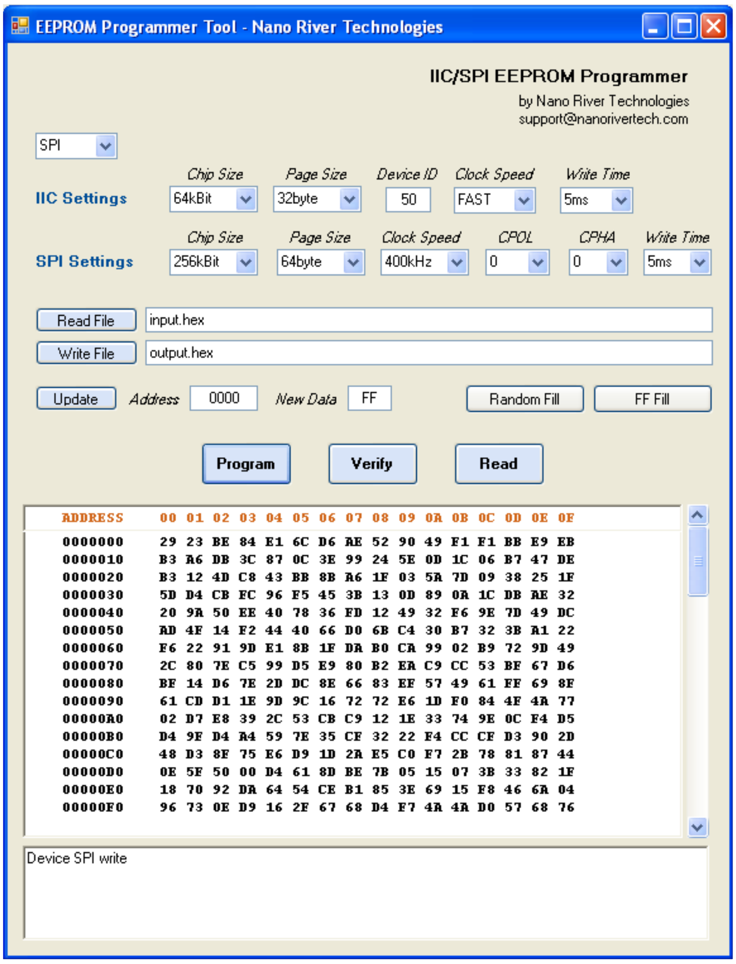
Device ID: 2B

Data Length: 3 · Read Data: AA BB CC

```
Filling slave buffer...
 0xAA 0xBB 0xCC
```

Windows EEPROM Programmer

This is a simple I2C and SPI EEPROM programmer written as a windows GUI application in Visual C++. It illustrates how to build a more complex application and also how to link to both I2C and SPI functions in the API.


EEPROM Programmer Tool - Nano River Technologies
IIC/SPI EEPROM Programmer
by Nano River Technologies
support@nanorivertech.com
Read File
Write File
Update
Random Fill
FF Fill
Program
Verify
Read

SPI

IIC Settings: Chip Size 64kBit, Page Size 32byte, Device ID 50, Clock Speed FAST, Write Time 5ms

SPI Settings: Chip Size 256kBit, Page Size 64byte, Clock Speed 400kHz, CPOL 0, CPHA 0, Write Time 5ms

input.hex

output.hex

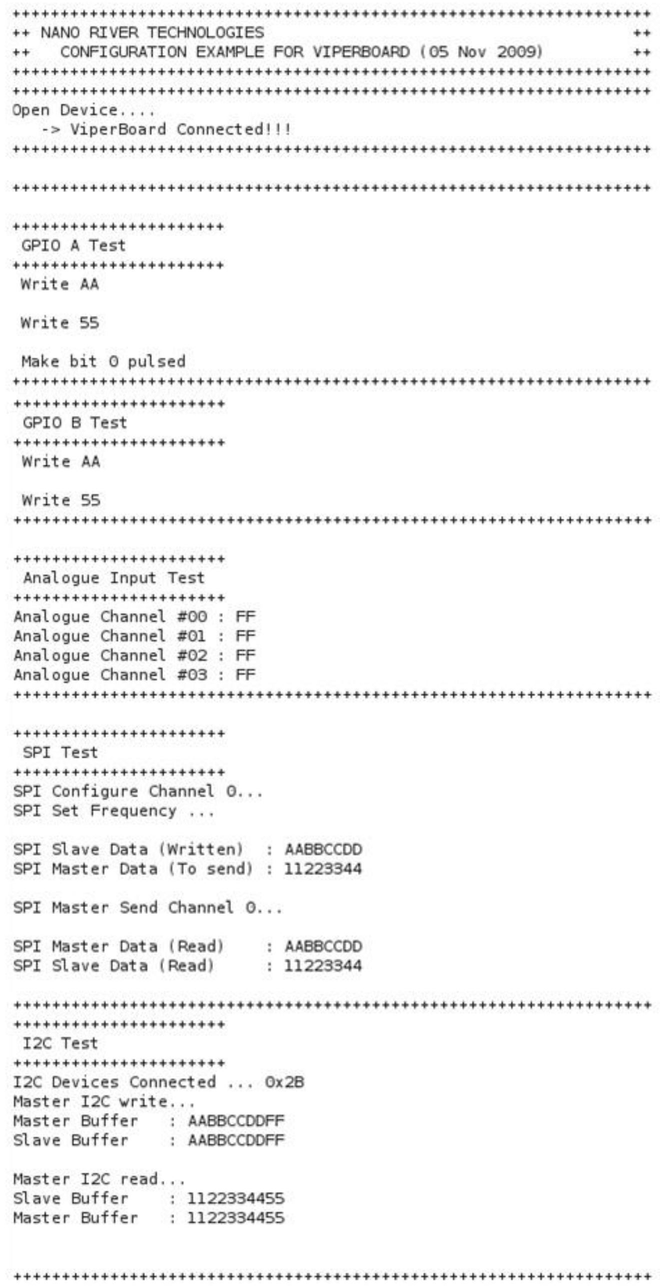
Address 0000 New Data FF

```
ADDRESS    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000    29 23 BE 84 E1 6C D6 AE 52 90 49 F1 F1 BB E9 EB
0000010    B3 A6 DB 3C 87 0C 3E 99 24 5E 0D 1C 06 B7 47 DE
0000020    B3 12 4D C8 43 BB 8B A6 1F 03 5A 7D 09 38 25 1F
0000030    5D D4 CB FC 96 F5 45 3B 13 0D 89 0A 1C DB AE 32
0000040    20 9A 50 EE 40 78 36 FD 12 49 32 F6 9E 7D 49 DC
0000050    AD 4F 14 F2 44 40 66 D0 6B C4 30 B7 32 3B A1 22
0000060    F6 22 91 9D E1 8B 1F DA B0 CA 99 02 B9 72 9D 49
0000070    2C 80 7E C5 99 D5 E9 80 B2 EA C9 CC 53 BF 67 D6
0000080    BF 14 D6 7E 2D DC 8E 66 83 EF 57 49 61 FF 69 8F
0000090    61 CD D1 1E 9D 9C 16 72 72 E6 1D F0 84 4F 4A 77
00000A0    02 D7 E8 39 2C 53 CB C9 12 1E 33 74 9E 0C F4 D5
00000B0    D4 9F D4 A4 59 7E 35 CF 32 22 F4 CC CF D3 90 2D
00000C0    48 D3 8F 75 E6 D9 1D 2A E5 C0 F7 2B 78 81 87 44
00000D0    0E 5F 50 00 D4 61 8D BE 7B 05 15 07 3B 33 82 1F
00000E0    18 70 92 DA 64 54 CE B1 85 3E 69 15 F8 46 6A 04
00000F0    96 73 0E D9 16 2F 67 68 D4 F7 4A 4A D0 57 68 76
```

Device SPI write

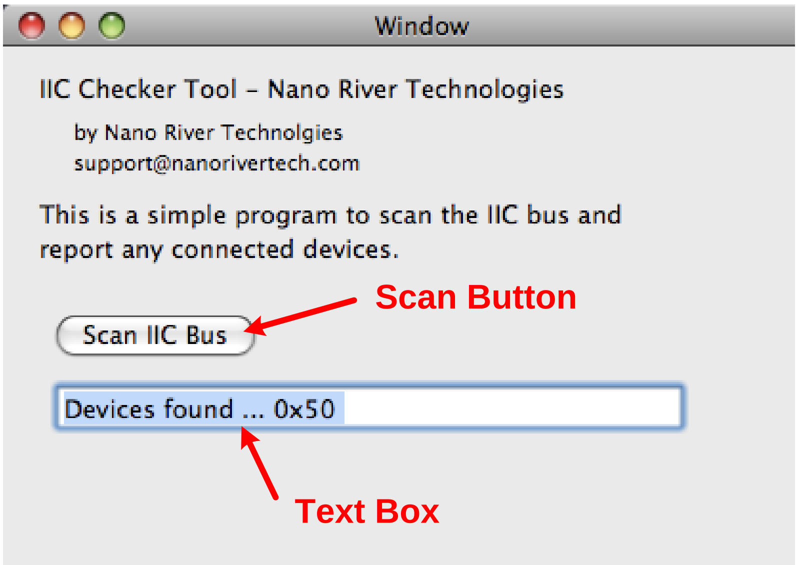
Configuration Example (for Linux and MacOS)

This is a simple GCC program for Linux and MacOS to show basic setup of GPIO, I2C, SPI and Anlogue IO interfaces.



```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++ NANO RIVER TECHNOLOGIES                                       ++
++   CONFIGURATION EXAMPLE FOR VIPERBOARD (05 Nov 2009)          ++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Open Device....
   -> ViperBoard Connected!!!
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 GPIO A Test
++++++++++++++++++++++
 Write AA

 Write 55

 Make bit 0 pulsed
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++
 GPIO B Test
++++++++++++++++++++++
 Write AA

 Write 55
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 Analogue Input Test
++++++++++++++++++++++
Analogue Channel #00 : FF
Analogue Channel #01 : FF
Analogue Channel #02 : FF
Analogue Channel #03 : FF
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++
 SPI Test
++++++++++++++++++++++
SPI Configure Channel 0...
SPI Set Frequency ...

SPI Slave Data (Written)  : AABBCCDD
SPI Master Data (To send) : 11223344

SPI Master Send Channel 0...

SPI Master Data (Read)    : AABBCCDD
SPI Slave Data (Read)     : 11223344

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++
 I2C Test
++++++++++++++++++++++
I2C Devices Connected ... 0x2B
Master I2C write...
Master Buffer   : AABBCCDDFF
Slave Buffer    : AABBCCDDFF

Master I2C read...
Slave Buffer    : 1122334455
Master Buffer   : 1122334455


+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

I2C Checker Tool *(for MacOS Xcode Cocoa)*



IIC Checker Tool – Nano River Technologies

by Nano River Technolgies
support@nanorivertech.com

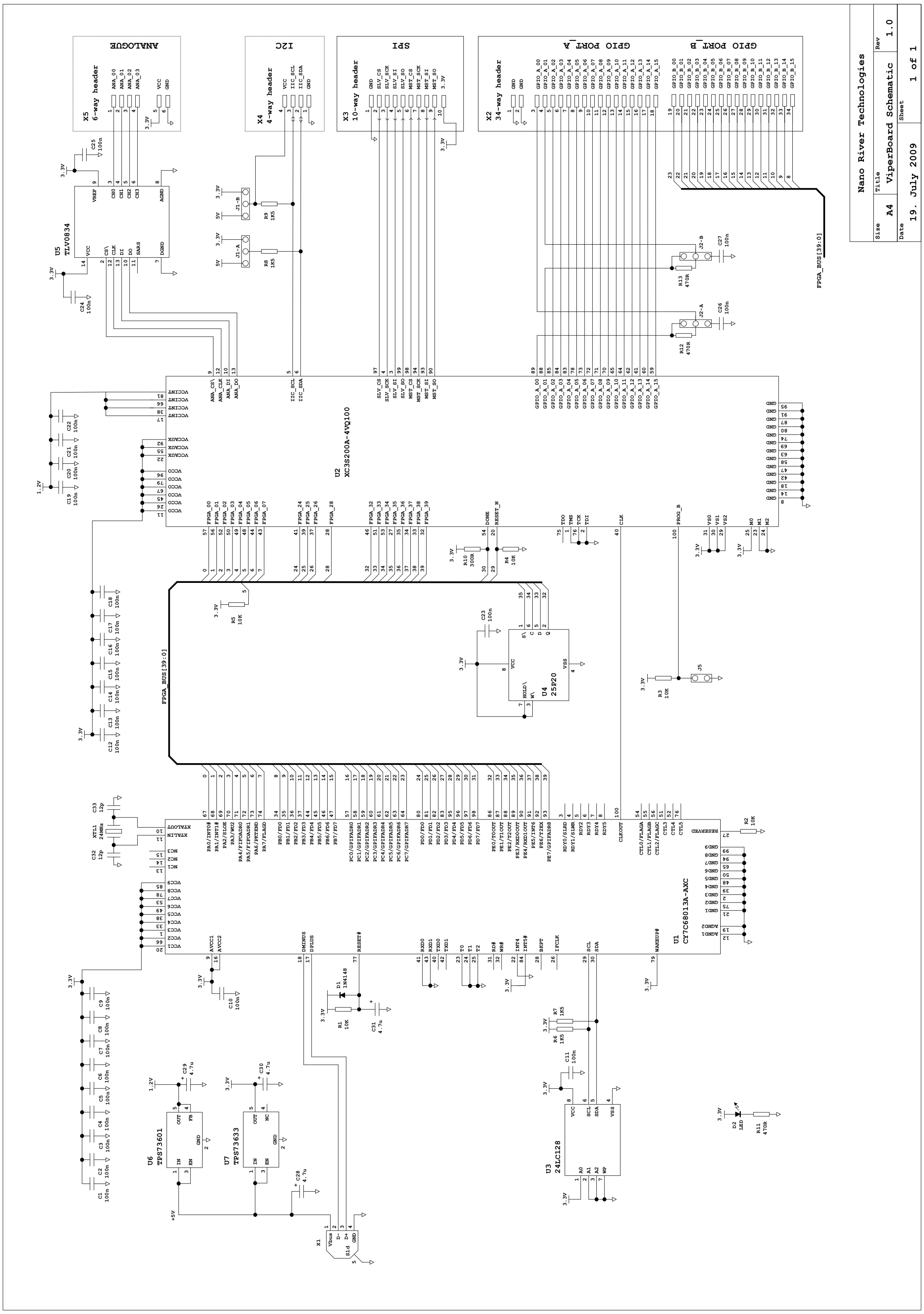
This is a simple program to scan the IIC bus and report any connected devices.

Devices found ... 0x50

This is a simple GUI program for MacOS showing how to use ViperBoard to scan IIC devices from Xcode Cocoa.

7. Appendix A : Schematic

See over.


Nano River Technologies
Size A4
Title ViperBoard Schematic
Rev 1.0
Date 19. July 2009
Sheet 1 of 1
ANALOGUE
X5 6-way header
I2C
X4 4-way header
SPI
X3 10-way header
GPIO PORT_A
GPIO PORT_B
X2 34-way header
U5 TLV0834
J1-A
J1-B
J2-A
J2-B
J5
U2 XC3S200A-4VQ100
U4 25P20
FPGA_BUS[39:0]
U1 CY7C68013A-AXC
XTL1 24MHz
U6 TPS73601
U7 TPS73633
U3 24LC128
X1
D1 1N4148
D2 LED
3.3V
1.2V
+5V