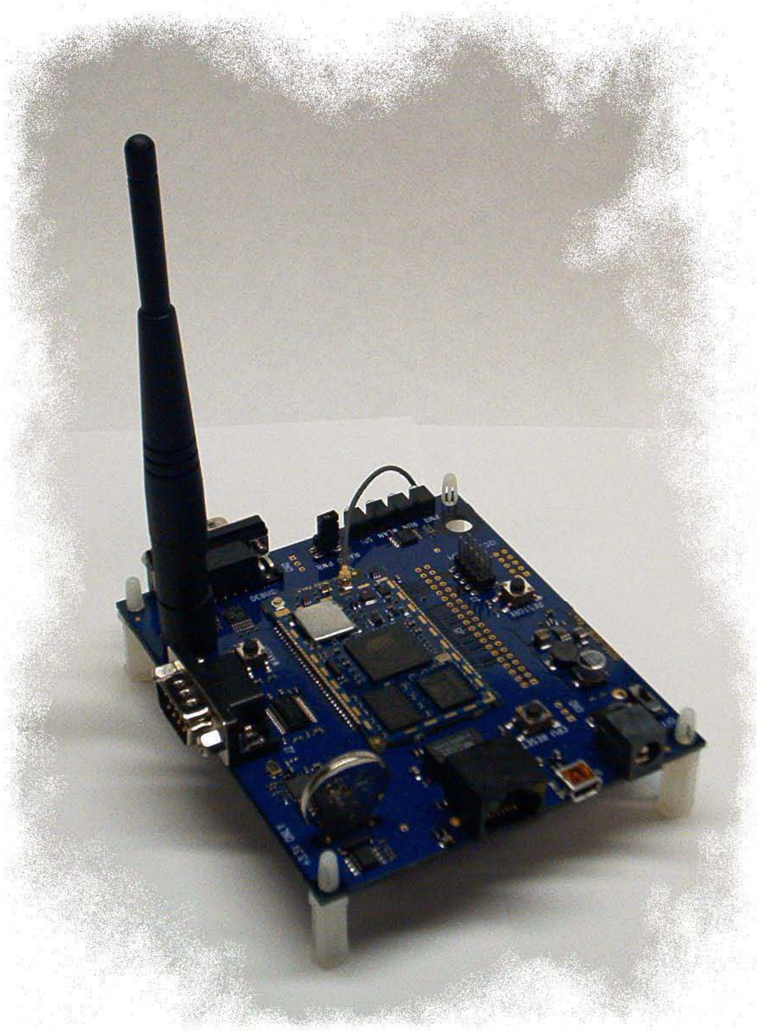
DVK-WLM400/402



Developer Kit Manual

Innovative **Technology**

for a **Connected** World

Laird Technologies is the world leader in the design and manufacture of customized, performance-critical products for wireless and other advanced electronics applications. Laird Technologies partners with its customers to find solutions for applications in various industries such as:

Network Equipment
Telecommunications
Data Communications
Automotive Electronics
Computers
Aerospace
Military
Medical Equipment
Consumer Electronics

Laird Technologies offers its customers unique product solutions, dedication to research and development, as well as a seamless network of manufacturing and customer support facilities across the globe.

Change History

Issue	Change	Author	Date
00a	Document Created	John Talley	Dec-1-2009
00b	Preliminary version	John Talley	April-2-2010
00d	Distributed to Sales Team	John Talley	April-25-2010
1.00	Initial Product Release	John Talley	June-2-2010

Glossary of Terms

Acronym	Term	Description
dboard	dboard.exe	Performance tool which creates a UDP socket on a remote PC to receive packets and measure system throughput. **dboard.exe IS ACCESSABLE AFTER THE INSTALLATION BY…** **Start→All Programs→Wism+→DashBoard Performance Monitor**
DVK	Dev Kit	This refers to the Developers Kit PCB which is designed to provide all power supply and interface support hardware for a WISM+
WAPI	WISM+ API	This is the API command language used between a host and WISM+ for configuration and data transfer
WIF	WIF.exe	This stands for **W**ism+ **I**nter**F**ace program. This is the PC based executable program for configuring and running performance tests for the WISM+/Dev Kit. **WIF.EXE IS ACCESSABLE AFTER THE INSTALLATION BY…** **Start→All Programs→Wism+→Wism+ Config Tool**
WISM+		This refers to a WLM400 (for US) or WLM402 (for CE)

Table Of Contents

1 OVERVIEW

This Dev Kit consists of a WLM400/WLM402 WISM+ (**W**i-Fi **I**ntelligent **S**erial **M**odule) wireless module mounted to a support board (Dev Kit) which allows you to connect a PC to the WISM+ using your choice of interface. A configuration/performance tool (WIF.exe) is provided so you can set up custom WISM+ configurations, then and measure the performance of the WISM+ wireless module from Laird Technologies.

*The download and installation of this package provides you with the **following executables** by clicking…*

Start→All Programs→Wism+→

- **DashBoard Performance Monitor** (this is actually **dboard.exe** performance tool)
- **Wism+ Config Tool** (this is actually **WIF.exe** configuration tool)

*The download and installation of this package provides you with the **following documentation** by clicking…*

Start→All Programs→Wism+→Documentation→

- **Developer Kit Manual** -this manual that you are reading
- **Release Notes** -important notes about this particular release
- **SDK API Reference** -description of the Software Developer Kit source code
- **WISM+ User Manual** -the User Manual for WISM+ module (WLM400 & WLM402)
- **Wireless API Reference** -description of the low level WISM API (WAPI) commands

*The download and installation of this package also provides you the **SDK (Software Developer Kit) source code**. This source code (written in c) represents a working sample of how to communicate with the WISM+. Use the "***SDK API Reference"** *as a guide to understanding the C functions. The directory of source files can be found at the following path…*

C:\Program Files\Laird Technologies\Wism+\wapi

There are several choices for which interface to use between the PC and the Dev Kit, but the easiest to use right out of the box, is the USB. This is the suggested way to get started…

- Get familiar with the Dev Kit hardware (see section: "**Hardware**").
- Set up the WISM+ USB driver (see section: "**Configuring PC USB port for WIF.exe**").
- Launch WIF.exe (see section: "**Starting WIF.exe over the PC USB port**").
- Become familiar with the configuration tool (see section: "**WIF.exe Operation**").
- Use WIF.exe's Performance Tool (UDP 802.11b mode) with the program "dboard.exe" to measure the performance of the WISM+ (see section: "**Using the Performance Tool in Ad-Hoc Mode**").
- Use WIF.exe's Performance Tool (UDP 802.11b/g mode) in infrastructure mode with the program "dboard.exe" between a laptop with an access point and another laptop with a WISM+ see section: "**Using the Performance Tool in Infrastructure Mode**".
- Create an infrastructure mode (802.11b/g mode) network between a laptop with an access point and another laptop with a WISM+. Share a folder on each, and then copy a large file across the wireless network, and measure the performance of the WISM+… see section: "**Using WISM+ in Bridge Mode w/Access Point**".

Note: All PC specific references are made to the Windows XP operating system.

1.1 Parts List

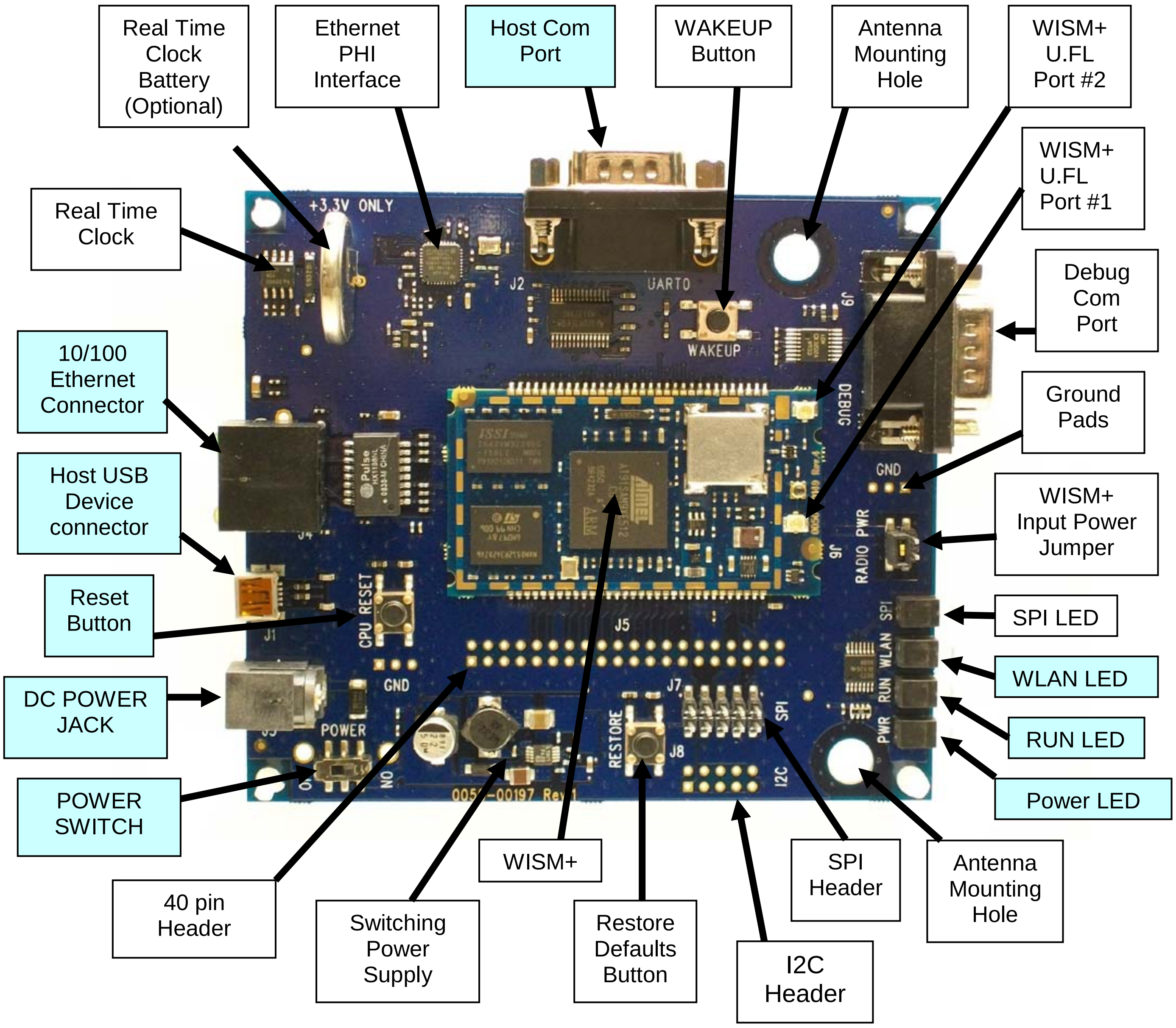
The WISM+ Dev Kit includes

- Dev Kit PCB with a WLM400 (FCC/IC) or a WLM402 (CE) WISM+ mounted
- Two Antennae installed on the Dev Kit and connected to the WISM+ antenna ports
- 120VAC/7.5VDC wall brick power adapter
- USB cable “type A to mini” for WISM+ Dev Kit connection to connect to a PC
- RS-232 serial cable (female-female) for WISM+ COM port connection to a PC serial port

Note: an Ethernet cable is not provided.

2 Hardware

2.1 Picture/Mechanical Drawing detailing high level components



Full schematics and assembly drawing are provided (see section: "**Dev Kit Design**").

🟦 ← Blocks in this colour are essential things to become familiar with.

2.2 Host Interfaces

2.2.1 Host Com Port

This is a DCE COM port which operates at RS-232 levels. It is a male DB9 connector which is meant to be connected to a PC serial port via the serial cable (female to female) provided with the Dev Kit.

DB9	DB9		WISM+	WISM+	
PIN#	PIN NAME	I/O	PIN #	PIN NAME	DESCRIPTION
1	DB9_DCD	O	59	$\overline{\text{CONNECTED}}$	The WISM+ $\overline{\text{CONNECTED}}$ pin is translated to RS-232 levels (and drives pin 1 of the DB9) and indicates to the PC that we have joined a network. When "connected" WISM+ drives pin 59 pin low and is level translated to a positive RS-232 level at the DB9.
2	DB9_TXD	O	6	UART_TXD	This pin is transmit data from the WISM+ back to the PC driven through a RS-232 level translator.
3	DB9_RXD	I	9	UART_RXD	This is receive data from the PC to the WISM+. An RS-232 receiver converts RS-232 levels at the connector to CMOS levels for WISM+.
4	DB9_DTR	I	12	$\overline{\text{UART\_DTR}}$	This is an output from the PC and is converted to CMOS levels to drive the WISM+ $\overline{\text{UART\_DTR}}$ input. **CURRENTLY THIS PIN HAS NO FUNCTION**.
5	GROUND	-	-	-	This is tied to Dev Kit GROUND.
6	DB9_DSR	O	8	$\overline{\text{DSR\_READY}}$	When the WISM+ cannot take any more data (all buffers are full) it will assert $\overline{\text{DSR\_READY}}$ pin low—which is translated to a high on DB9 pin 6. - **This pin becomes DTR at the PC**-
7	DB9_RTS	I	10	$\overline{\text{UART\_RTS}}$	This is converted to CMOS levels to drive the WISM+ $\overline{\text{UART\_RTS}}$ input. **The host can hold off the WISM+ from transmitting serial data when asserted- ONLY IF FLOW CONTROL IS ENABLED ON WISM+.**
8	DB9_CTS	O	7	$\overline{\text{UART\_CTS}}$	WISM+ $\overline{\text{UART\_CTS}}$ pin drives an RS-232 level translator to signal the host that it is ready to accept serial data**. WISM+ drives this pin low when it can accept serial input from the host (creating a positive RS-232 level voltage to the PC). If WISM+ cannot accept serial input, it drives the pin high (creating a negative RS-232 voltage).**
9	DB9_RI	0	-	-	This pin should be ignored at the host.

The host com port can be used for the following things…

- HyperTerminal can be used on a PC to send AT commands to the WISM+ and display responses.
- WIF.exe can be configured to operate over host com port after an “at+quit”.
- The entire WAPI command interface can be implemented over the host com port after an “at+quit”.

2.2.2 SPI Header

This is the host communications SPI interface to the WISM+. This is a dual-row 10 pin male header which mates with the Ardvark USB to SPI module http://www.totalphase.com/products/aardvark_i2cspi/. If you purchase this device, and load the drivers, WIF.exe can be configured to operate over the SPI port. The following table indicates the pin out of the SPI header…

Pin 1 is the lower left pin. The bottom row of pins are numbered 1,3,5…etc (with even pin numbers on top).

SPI Header PIN#	SPI Header PIN NAME	I/O	WISM+ PIN #	WISM+ PIN NAME	DESCRIPTION
1	$\overline{\text{SPI\_REQ}}$	O	19	$\overline{\text{SPI\_REQ}}$	This is an output from the WISM+ which is asserted (low) when the WISM+ has data to be transferred back to the host.
2	GND				Ground pin connected to the ground plane of the Dev Kit PCB.
3	RESERVED				This pin does go to a WISM+ port pin, but has no function.
4	N/C				This pin is not connected
5	SPI_MISO	O	28	SPI_MISO	This is the MISO (Master In Slave Out) data line for the SPI interface. This should be connected to the MISO of a SPI Master device.
6	N/C				This pin is not connected
7	SPI_CLK	I	30	SPI_CLK	This is the clock input for the SPI interface. This is an input to the WISM+ because the WISM+ is always a SPI SLAVE device, intended to be connected to the clock output of a SPI Master device.
8	SPI_MOSI	I	29	SPI_MOSI	This is the MOSI (Master Out Slave In) data line for the SPI interface. This should be connected to the MOSI of a SPI Master device.
9	$\overline{\text{SPI\_SS}}$	I	31	$\overline{\text{SPI\_SS}}$	This is the SPI Slave Select input to the WISM+. This is driven low by a SPI Master device to frame the data transfer.
10	GND				Ground pin connected to the ground plane of the Dev Kit PCB.

2.2.3 I2C Header

This is the WISM+ I2C two-wire interface. It is also connected to the real time clock on the Dev Kit board. These pins are also compatible with the Ardvark USB to SPI/I2C device, but currently, the only function for the I2C bus is the interface to the Real-Time-Clock populated on the Dev Kit PCB. In the future, this interface might be set up so the WISM+ can interface to other defined I2C sensors.

I2C Header PIN#	I2C Header PIN NAME	I/O	WISM+ PIN #	WISM+ PIN NAME	DESCRIPTION
1	I2C_CLK	I/O	23	I2C_CLK	This is the I2C clock line.
3	I2C_DAT	I/O	22	I2C_DAT	This is the I2C data line.
2,10	GND				These pins are tied to the ground plane of the Dev Kit PCB
4,5,6,7,8,9	N/C				These pins are not connected.

2.2.4 10/100 Ethernet Connector

This is a standard RJ-45 Ethernet connector and is another host interface to the WISM+. A standard Ethernet cable can be connected between a PC and the Dev Kit and WIF.exe can be configured to operate over the PC's Ethernet port.

2.2.5 USB Device "MINI" Connector

This is a standard USB mini connector which can be used to connect the Dev Kit to a PC over the host USB interface. A USB cable is provided to connect to a PC. By default, WIF.exe will use the USB port for WISM+ communications. This is the recommended interface to use for the Dev Kit right out of the box.

On the Dev Kit PCB, the USB connector is wired to the WISM+ USB device pins: WISM+ pin 4 (USB_DDM) and WISM+ pin 5 (USB_DDP).

2.2.6 Debug Com Port

This is an RS-232 DCE serial port. This port outputs internal debug information which can be used by Laird Technologies to attempt to debug WISM+ operations. There is no hardware flow control. DB9 Pin 2 is transmit data to the PC, and DB9 pin 3 is receive data from the PC.

2.3 RF

2.3.1 WISM+ U.FL Port#1 & #2

These are the two WISM+ RF ports. They utilize a miniature U.FL type non-standard RF connector per FCC/IC rules. The WISM+ Dev Kit comes with two antennae connected. This gives you the ability to test antenna diversity, or to configure a single antenna.

NOTE: See section: "**Configure WISM+ Antenna**" for the details on the procedure to change the WISM+ antenna configuration.

2.3.2 Antenna Mounting Holes

These holes are used to mount antennae. Both mounting holes can have antennae installed and connected to WISM+ RF ports #1 & #2 U.FL connectors. The Dev Kit is shipped stock with both antennae and is configured out of the box to use antenna diversity.

2.4 POWER/GND

2.4.1 DC Power Jack

The DC power jack accepts a 2mm plug from the AC/DC wall-brick power supply adapter. It is recommended that you use the wall-brick power supply which is shipped with the Developer Kit, although, the input voltage range is 5 to 25 Volts (plus on the center).

2.4.2 Power Switch

The power switch when in the ON position will connect the DC power jack to the input of the switching power supply, providing power to the entire development kit and the WISM+.

2.4.3 Switching Power Supply

The input to this switching power supply is provided from the DC power jack and is protected for reverse polarity with a series diode. The output of the switching power supply provides power (3.3V) to the entire development board circuitry as well as the WISM+.

2.4.4 WISM+ Input Power Jumper

This is a 4 position, dual row header. The two lower pins are connected to the WISM+ VCC input pins, and the two upper pins are connected to the output of the 3.3V Dev Kit switching power supply. For normal operation, only one jumper need be installed (shorting an upper pin to a lower pin).

To make WISM+ current measurements, remove the jumper and connect an ammeter between an upper and a lower pin.

To insert an external DC power input, connect a power supply ground to one of the ground pads, and the positive side of the power supply to one of the lower input power pins.

2.4.5 Ground Pads

These three thru-hole vias are made available simply to allow the user to have access to a system ground.

2.5 LEDs/INDICATORS

2.5.1 Power LED

This LED indicates DC power is being supplied to the WISM+ (DC power connection is made to the Dev Kit board and the ON/OFF switch is in the ON position.

2.5.2 RUN LED

This LED is the WISM+ heartbeat. It will blink fast at boot time, pause, and then blink at a 1Hz rate once the WISM+ processor system has booted. This LED is driven by the WISM+ “HEARTBEAT” pin (WISM+ pin# 57).

2.5.3 WLAN LED

This LED will be lit when the WISM+ is wirelessly connected to another device (or joined a network). This LED is driven by the WISM+ “$\overline{\text{CONNECTED}}$” pin (WISM+ pin#59)

2.5.4 SPI

This LED will be lit whenever the SPI port is being used as the host interface, and communications are active over the SPI port.

2.6 BUTTONS

2.6.1 Reset Button

This provides a hardware reset to the WISM+. The WISM+ will re-boot after this has been pressed. This button drives WISM+ $\overline{\text{RESET\_IN}}$ (pin# 21) to ground when it is pressed.

2.6.2 Restore Defaults Button

The Restore Defaults Button will restore the WISM+ with the saved environment which was set up at Laird production/test time. The WISM+ must perform a boot while the Restore Defaults Button is pressed and remain pressed until the heartbeat LED begins flashing in order for this function to work. This button is connected to the WISM+ $\overline{\text{RESTORE}}$ pin (pin# 56). $\overline{\text{RESTORE}}$ is pulled LOW when the button is pressed.

2.6.3 WAKEUP Button

This button is used to wake the WISM+ from a deep sleep. A WAPI API command is used over the host interface to enter deep sleep. This button is connected to the WISM+ pin WAKEUP (pin# 63). When the button is pressed the WISM+ WAKEUP pin is pulled to 3.3V which initiates a wake from deep sleep.

2.7 CLOCK

2.7.1 Real Time Clock

This is a Texas Instruments BQ32000 real time clock chip connected to the WISM+ I2C bus. WISM+ API commands can be executed to both set and read the current time and date. This is only required in certain situations when using WPA Enterprise security.

2.7.2 Real Time Clock Battery

This is the backup battery (not installed) for the real time clock on the Dev Kit board. The Real Time Clock chip will be powered when the Dev Kit is powered up -even if the battery is not installed.

This battery is a standard lithium 3.0V 220mAH tabbed coin cell. If you want battery backup for the Clock, we recommend part# CR2032RV-MFR (made by Renata) available at Mouser.

2.8 WISM+

Depending on which development kit which has been purchased, this will be either:

- WLM400 –for North America carrying FCC and IC approvals
- WLM402 –for Europe with CE regulatory compliance

2.9 40 Pin Header

The 40 pin header allows access to certain WISM+ pins. The pin out of the header (J5) is defined in the Dev Kit board schematics- See Section: "**Schematics**"

3 Using AT Commands with PC Serial Port

A PC serial port can be used to execute a set of simple AT commands using HyperTerminal. The following section shows you how to set up a HyperTerminal session to communicate with the WISM+ using its host COM port.

Refer to the **“*AT Command Reference*”** for descriptions of the AT command set.

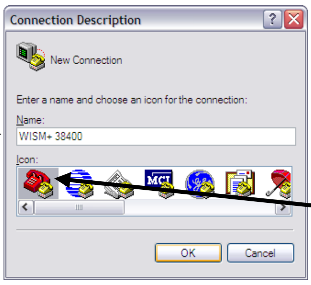
The WISM+ module (WLM400/402) implements a serial port at 3.3V levels. The WISM+ Dev Kit has industry standard RS-232 level translators and provides a male DB-9 connector so you can connect to a PC serial port using the female-female serial cable provided with the Dev Kit. The WISM+ dev kit is viewed as a DCE (Data Communications Equipment). If you use a standard serial cable (male to female), you will need a gender changer (not a Null-Modem adapter) at the WISM+ Dev Kit end.

If your PC does not have a serial port, you’ll need a USB to Serial converter. Most USB to Serial converters will require a driver to be loaded. You must first follow the instructions which came with your USB to Serial converter and install the driver as they recommend.

Once your serial port is in place and you know the COM port number (COM6, COM1 etc.) you can perform the following steps to set up HyperTerminal to communicate from your PC to the WISM+ Dev Kit hardware over the serial port…

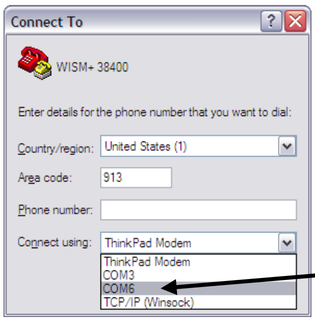
- Launch HyperTerminal by clicking…
- Start Button→All Programs→Accessories→Communications→HperTerminal and you will see the following…

We have entered a name for this HyperTerminal Session (“WISM+ 38400”).

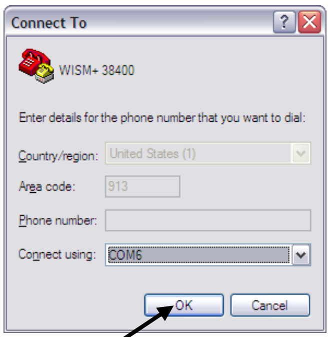

Connection Description
New Connection
Enter a name and choose an icon for the connection:
Name: WISM+ 38400
Icon:

Select the “Telephone” icon and in the “Name” field,

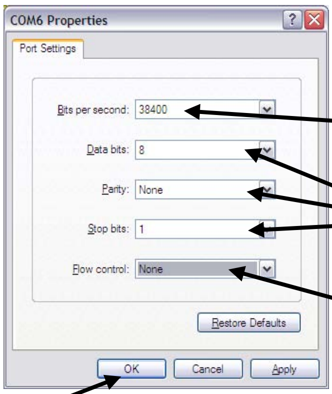
Click “OK” and you will see the following screen…


Connect To
WISM+ 38400
Enter details for the phone number that you want to dial:
Country/region: United States (1)
Area code: 913
Phone number:
Connect using: ThinkPad Modem

Select the COM port connected your to the WISM+ and you will you will see this


Connect To
WISM+ 38400
Enter details for the phone number that you want to dial:
Country/region: United States (1)
Area code: 913
Phone number:
Connect using: COM6

Click “OK” and the following COM port properties screen will appear…



COM6 Properties

Port Settings

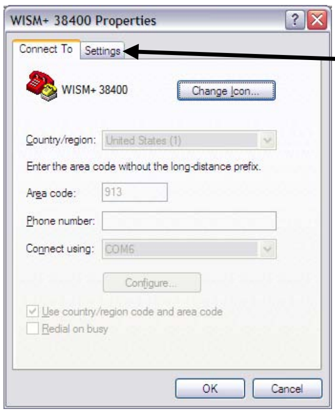
Bits per second: 38400
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Select 38400 for the baud rate (WISM+ defaults to 38400 baud on its COM port)

Select 8 bits, No Parity, and 1 Stop Bit

Select NO Hardware Flow Control because WISM+ is set by default to NOT use Hardware Flow Control

- Click "OK"
- At the top (HyperTerminal menu) click…**File→Properties** and the following screen will appear…



WISM+ 38400 Properties

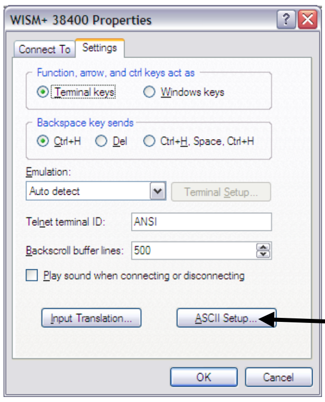
Connect To | Settings

WISM+ 38400

Country/region: United States (1)
Enter the area code without the long-distance prefix.
Area code: 913
Phone number:
Connect using: COM6

[✓] Use country/region code and area code
[ ] Redial on busy

Click on the "Settings" Tab and you will see this

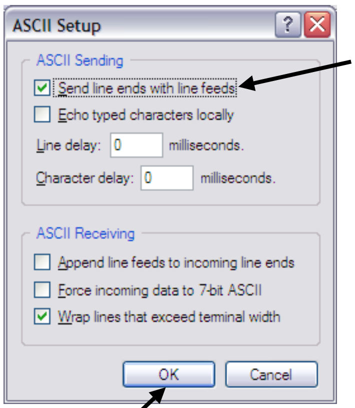


WISM+ 38400 Properties

Connect To | Settings

Function, arrow, and ctrl keys act as: [✓ Terminal keys] Windows keys
Backspace key sends: [✓ Ctrl+H] Del Ctrl+H, Space, Ctrl+H
Emulation: Auto detect
Telnet terminal ID: ANSI
Backscroll buffer lines: 500
[ ] Play sound when connecting or disconnecting

Click the "ASCII Setup" tab and you will see the ASCII Setup screen



ASCII Setup

ASCII Sending
[✓] Send line ends with line feeds
[ ] Echo typed characters locally
Line delay: 0 milliseconds.
Character delay: 0 milliseconds.

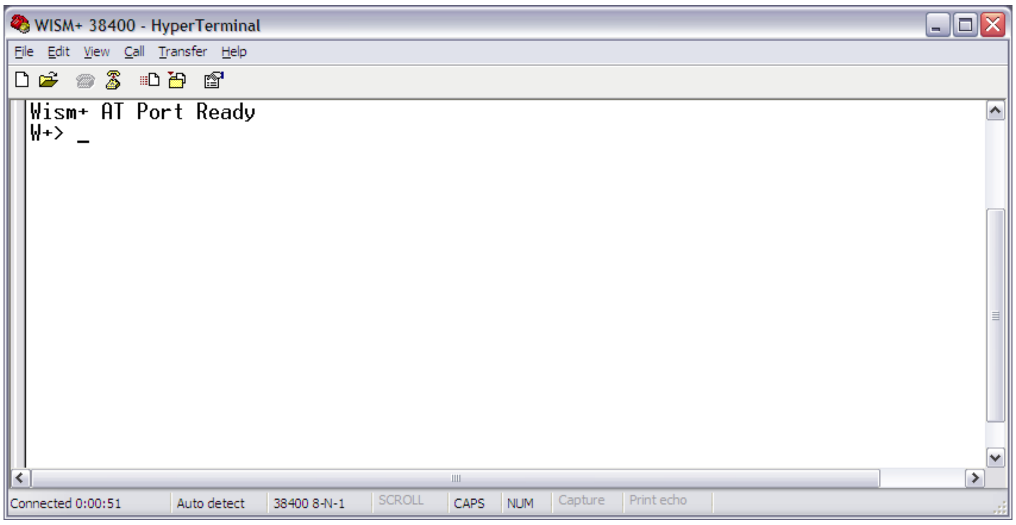
ASCII Receiving
[ ] Append line feeds to incoming line ends
[ ] Force incoming data to 7-bit ASCII
[✓] Wrap lines that exceed terminal width

Select the "Send line ends with line feeds" checkbox. This will ensure that when you press the "ENTER" key while typing in HyperTerminal, that a Line End character is sent with the string to the WISM+.

Click the "OK" button and your COM port configuration is complete.

Now the PC serial port is configured to communicate with the WISM+ host com port. Now follow the steps below…

- Ensure that the WISM+ Dev Kit power switch is in the “OFF” position.
- Plug in the AC/DC power adapter to 120AC power outlet
- Plug the AC/DC power adapter’s DC power plug into the Dev Kit DC power jack
- Make sure that the serial cable supplied with the Dev Kit is connected between the PC (or PC USB to Serial adapter).
- In HyperTerminal, make sure the “hook” is picked up (HyperTerminal is connected)
- Switch the WISM+ Dev Kit power switch to the “ON” position
- Wait 6 or 8 seconds for the WISM+ to boot
- You will see the “Heartbeat" LED” (labelled “RUN” on the WISM+ Dev Kit) begin to flash at a 1Hz rate
- You should then see the following on the HyperTerminal



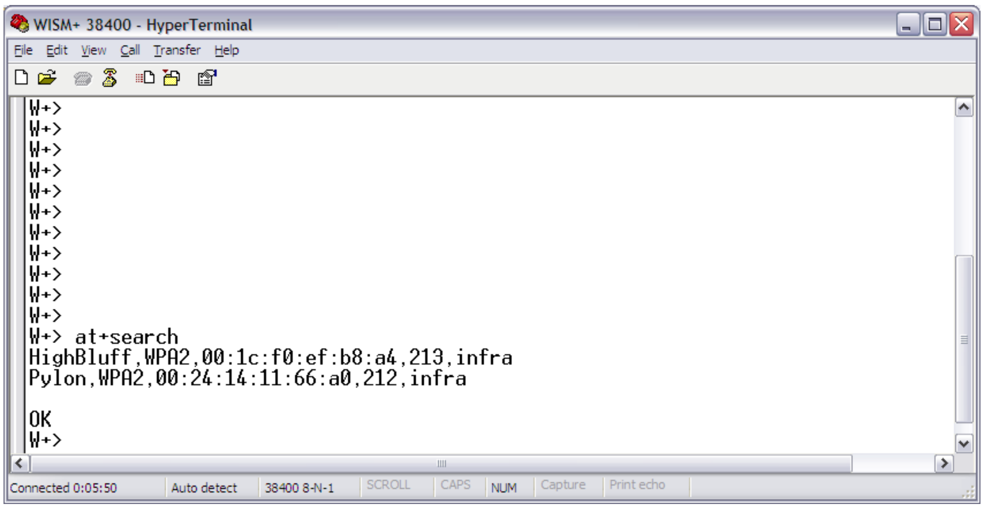
```
Wism+ AT Port Ready
W+> _
```

The HyperTerminal screen shot above shows the notification from WISM+ (“WISM+ AT Port Ready”) that it has successfully booted. Below the boot message, you can see the “W+>” prompt sent by WISM+ indicating that it is ready for you to type an “at+xxxx” command. By simply pressing the “ENTER” key, another “W+>” prompt should be displayed on the next line showing that the COM port connection is working in both directions.

If you **do not** see a new “W+>” prompt each time you press “ENTER”, you could have one of the following…

- You may not have selected the proper COM port in HyperTerminal
- Your USB to serial adapter may not be performing properly
- You may not have set up HyperTerminal COM port properties properly… 38400, 8bits, no parity, 1 stop bit, NO hardware flow control
- You are not using the supplied serial cable between the Dev Kit and the PC serial port
- You may not have selected “Send line ends with line feed” in the HyperTerminal ASCII setup

With the PC to WISM+ com port communications working properly, the following screen shot shows the result of pressing “ENTER” multiple times, and then the execution of an “at+search” command…


WISM+ 38400 - HyperTerminal
File Edit View Call Transfer Help
Connected 0:05:50 Auto detect 38400 8-N-1 SCROLL CAPS NUM Capture Print echo

```
W+>
W+>
W+>
W+>
W+>
W+>
W+>
W+>
W+>
W+>
W+>
W+> at+search
HighBluff,WPA2,00:1c:f0:ef:b8:a4,213,infra
Pylon,WPA2,00:24:14:11:66:a0,212,infra

OK
W+>
```

Above, you can see the resulting response from the WISM+ “at+search” command. Two access points have been found: HighBluff, and Pylon.

By typing “at+help” and “ENTER” at the WISM+ prompt, you will see a list of all of the AT commands which can be exercised.

Refer to the “***AT Command Reference***” for descriptions of the AT commands.

4 Software- WIF

WIF.exe has been installed by running the setup file from the download and is launched by clicking…

Start→All Programs→Wism+→Wism+ Config Tool

WIF.exe actually exists in the following folder… *c:\Program Files\Laird Technologies\Wism+*

The WIF.exe tool is used for the following:

- basic WISM+ configuration (over USB, Serial, Ethernet, or SPI)
- performance tool to measure WISM+ to Wireless Access Point/Router data through-put

By default, WIF.exe uses the USB interface to a Windows PC.

NOTE: WIF.exe can be made to communicate over any of the WISM+ host interfaces,

There are some specific things which need to be done depending upon which host interface of the WISM+ Dev Kit which you are going to use. USB is the default, but Ethernet, RS-232, and SPI will also work.

The required preparation for using each of these interfaces will be described in the following sections.

4.1 Setting up PC ports for WIF-WISM+ communications

4.1.1 Configuring PC USB port for WIF.exe

To use the PC USB port to communicate with the WISM+ Dev Kit, a USB driver must be installed. This driver exists on your PC after running SETUP.EXE. By default, WIF.exe will look for an active USB connection from the PC to the WISM+.

Before starting the WIF.exe software, perform the following steps…

- Set the Dev Kit power switch to the OFF position
- Plug the AC/DC wall brick power adapter into an AC power outlet, and the other end into the Dev Kit's DC power jack.
- Attach the USB cable between the Dev Kit USB jack and your PC.
- Switch the Dev Kit power switch to the "ON" position
- Wait for the WISM+ to boot- the "Heartbeat LED" (labelled "RUN" on the WISM+ Dev Kit) will begin to flash at a 1Hz rate and then you will hear the USB device register with the PC.
- Your PC will notify you that new hardware has been found
- When Window's "New Hardware Wizard" pops up, instruct it to look for the appropriate driver **on your PC**. The wizard will find the USB driver and install it.
- Once the driver installation is complete, you are ready to proceed.
- You are now ready to start Wif.exe…

Now go to section: "**Starting WIF.exe**"

4.1.2 Configuring PC Serial port for WIF.exe

By default at boot time, the WISM+ host COM port only interprets AT commands. You can use WIF.exe configuration/performance tool over the WISM+ host port, but you must first send an “**at+quit**” command. After this command is sent to the WISM+, there will be no response from WISM+, but the WISM+ host COM port will now be in WAPI mode and is compatible with the WISM+ API commands used by WIF.exe.

To use WIF.exe over the WISM+ host COM port, perform the following steps:

- Follow the steps in section: “**Using AT Commands with PC Serial Port**” in order to set up your PC’s serial port for communications.
- Issue an “**at+quit**” AT command using HyperTerminal (and there will be no response).
- Now quit the HyperTerminal session (or hang up the hook) which will release the serial port for WIF.exe.
- Now go to section: “**Starting WIF.exe over the PC Serial port**”

4.1.3 Configuring PC Ethernet port for WIF.exe

WIF.exe can be configured to operate over the PC Ethernet port. There are a couple of things which must be done first, in order to allow proper Ethernet operation…

- WinPcap must be installed on your PC.
- Any PC service which traps out of the ordinary network traffic must be halted.

WinPcap

WinPcap is a networking package which MUST be installed on your PC in order for WIF.exe to communicate to the WISM+ over Ethernet. **You would have been warned at install time if WinPcap didn't exist on your PC.** This package is used by programs like "WireShark" to monitor network traffic (just one example). You must install this (if you don't have it) before attempting to use WIF.exe over the Ethernet port. It allows WIF.exe to capture and transmit network packets bypassing the windows protocol stack.

To install WinPcap on your PC, go to "**http://www.winpcap.org**". Click on the option on the left of the screen called "Get WinPcap". Under the "WinPcap 4.1.1 download" section, click on the link…

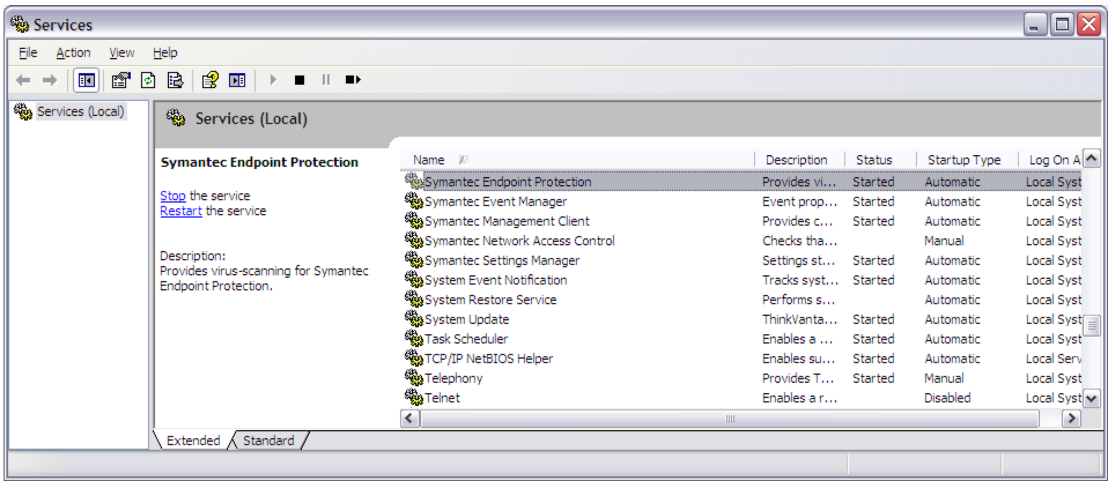
"WinPcap auto-installer (driver +DLLs)"

Download the installer and launch it on your PC. Follow the installation instructions so that it will automatically start up at boot time.

Stopping services which interfere with network traffic

At Laird, we use Symantec virus protection and network traffic monitoring. Part of its job is to look for potentially harmful network traffic which might be hacking into your computer. The WIF to WISM+ communications look potentially harmful to these types of services, and the services will interfere and capture the network traffic thus keeping WIF.exe from communicating with WISM+ over the Ethernet interface.

Click…Start→Control Panel and double-click on the "Administrative Tools" icon. Next, double-click on the "Services" icon. Once the "Services" window is open, scroll down to the virus protection services. The following screen shot is an example of the Symantec services which are running (if you use Symantec virus protection). Over to the right, under the "Status" column, you can see the services which have been started. Right click on the service, and click "Stop" on all the Symantec services which indicate "Started". Your PC will be vulnerable to attack, but you will only be connected to the WISM+ Dev Kit. When you are finished using WIF.exe, turn the services back on to connect back to your network, or simply re-boot.



Symantec Endpoint Protection

Stop the service
Restart the service

Description:
Provides virus-scanning for Symantec Endpoint Protection.

Name	Description	Status	Startup Type	Log On A
Symantec Endpoint Protection	Provides vi…	Started	Automatic	Local Syst
Symantec Event Manager	Event prop…	Started	Automatic	Local Syst
Symantec Management Client	Provides c…	Started	Automatic	Local Syst
Symantec Network Access Control	Checks tha…		Manual	Local Syst
Symantec Settings Manager	Settings st…	Started	Automatic	Local Syst
System Event Notification	Tracks syst…	Started	Automatic	Local Syst
System Restore Service	Performs s…		Automatic	Local Syst
System Update	ThinkVanta…	Started	Automatic	Local Syst
Task Scheduler	Enables a …	Started	Automatic	Local Syst
TCP/IP NetBIOS Helper	Enables su…	Started	Automatic	Local Serv
Telephony	Provides T…	Started	Manual	Local Syst
Telnet	Enables a r…		Disabled	Local Syst

4.2 Starting WIF.exe

Since WIF.exe uses the USB as the default port, an active USB connection is checked for at launch time, otherwise a port selection screen will pop up. See the following sections for specific instructions for using USB, serial, Ethernet & SPI.

4.2.1 Starting WIF.exe over the PC USB port

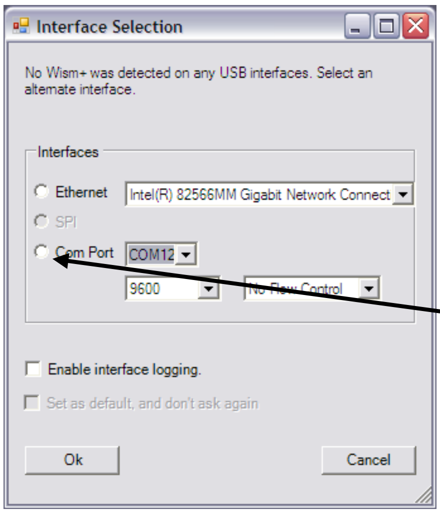
With the USB driver successfully installed, (see section: "**Configuring PC USB port for WIF.exe**") the USB cable connected between the WISM+ Dev Kit and the PC, WISM+ Dev Kit powered up, and the USB interface recognized by the PC, launch WIF by clicking…

Start→All Programs→Wism+→Wism+ Config Tool

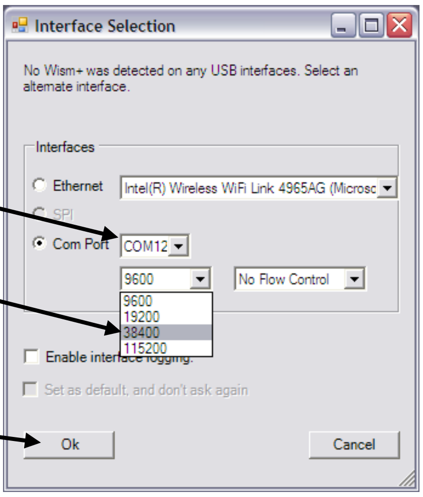
If everything is working properly, you will see the WIF front screen populated with the current state saved in WISM+ constant memory. Now go to section: "**WIF.exe Operation**".

4.2.2 Starting WIF.exe over the PC Serial port

- Since WIF.exe is looking for the USB connection upon launch, MAKE SURE THE USB CABLE IS DISCONNECTED.
- Make sure that you have exited AT command mode for the WISM+ host COM port as described in section: "**Configuring PC Serial port for WIF.exe**" (issue the "at+quit" command).
- Now, launch WIF.exe by clicking…**Start→All Programs→Wism+→Wism+ Config Tool**
- The following interface selection screen will appear…


Interface Selection
No Wism+ was detected on any USB interfaces. Select an alternate interface.
Interfaces
Ethernet
Intel(R) 82566MM Gigabit Network Connect
SPI
Com Port
COM12
9600
No Flow Control
Enable interface logging.
Set as default, and don't ask again
Ok
Cancel

Click on "Com Port" and then pull down the COM port selection box to select you PCs COM port. The default settings for WISM+ is 38400 with No Flow Control


Interface Selection
No Wism+ was detected on any USB interfaces. Select an alternate interface.
Interfaces
Ethernet
Intel(R) Wireless WiFi Link 4965AG (Microsc
SPI
Com Port
COM12
9600
No Flow Control
9600
19200
38400
115200
Enable interface logging.
Set as default, and don't ask again
Ok
Cancel

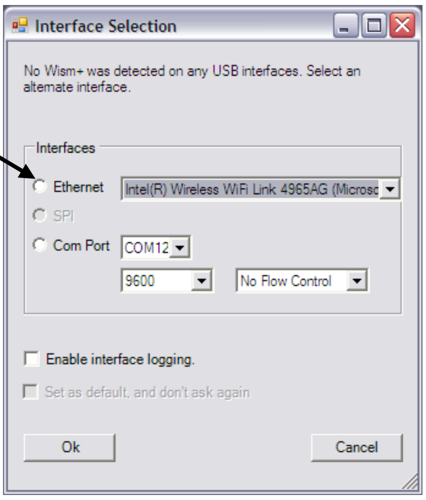
After setting up the COM port parameters click "OK".

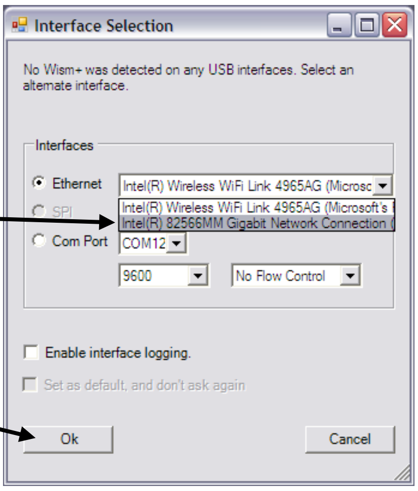
If everything is working properly, you will see the WIF front screen populated with the current state saved in WISM+ constant memory.

Now go to section: "**WIF.exe Operation**".

4.2.3 Starting WIF.exe over PC Ethernet port

- Since WIF.exe is looking for the USB connection upon launch, MAKE SURE THE USB CABLE IS DISCONNECTED.
- Connect a network cable between the WISM+ Dev Kit Ethernet port and the PC's Ethernet port.
- Make sure you have followed the steps in section: "**Configuring PC Ethernet port for WIF.exe**".
- Power up the WISM+ Dev Kit (or press the reset button) and wait for the "Run Led" to begin flashing at a 1Hz rate.
- Now launch Wif.exe by clicking…**Start→All Programs→Wism+→Wism+ Config Tool**.
  You will then see WIF.exe's interface selection screen as shown below…

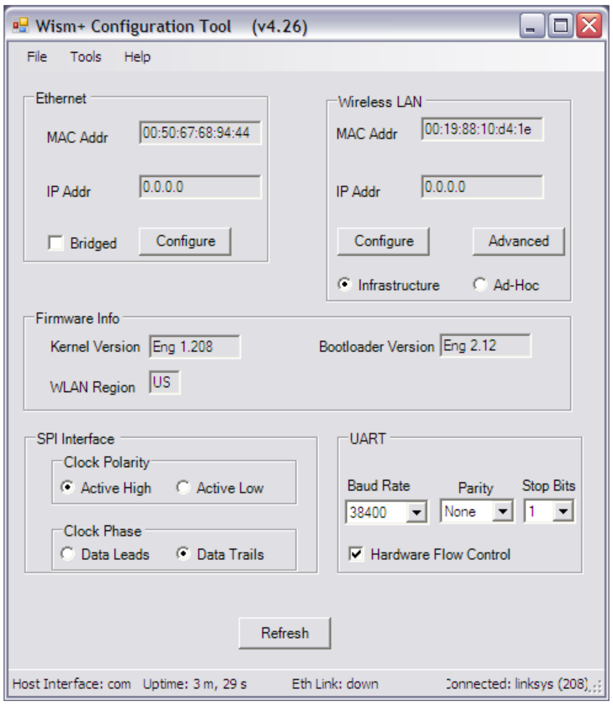

Select Ethernet
Interface Selection
No Wism+ was detected on any USB interfaces. Select an alternate interface.
Interfaces
Ethernet
Intel(R) Wireless WiFi Link 4965AG (Microsc
SPI
Com Port
COM12
9600
No Flow Control
Enable interface logging.
Set as default, and don't ask again
Ok
Cancel


Select the Ethernet adapter associated your PC's wired network connection,
Click OK.
Interface Selection
No Wism+ was detected on any USB interfaces. Select an alternate interface.
Interfaces
Ethernet
Intel(R) Wireless WiFi Link 4965AG (Microsc
Intel(R) Wireless WiFi Link 4965AG (Microsoft's
Intel(R) 82566MM Gigabit Network Connection
SPI
Com Port
COM12
9600
No Flow Control
Enable interface logging.
Set as default, and don't ask again
Ok
Cancel

If everything is working properly, you will see the WIF front screen populated with the current state saved in WISM+ constant memory.

Now go to section: "**WIF.exe Operation**".

4.3 WIF.exe Operation


Wism+ Configuration Tool (v4.26)

File Tools Help

Ethernet
MAC Addr: 00:50:67:68:94:44
IP Addr: 0.0.0.0
Bridged [ ]

Wireless LAN
MAC Addr: 00:19:88:10:d4:1e
IP Addr: 0.0.0.0
[✓ Infrastructure] Ad-Hoc

Firmware Info
Kernel Version: Eng 1.208
Bootloader Version: Eng 2.12
WLAN Region: US

SPI Interface
Clock Polarity: [✓ Active High] Active Low
Clock Phase: Data Leads [✓ Data Trails]

UART
Baud Rate: 38400
Parity: None
Stop Bits: 1
Hardware Flow Control [✓]

Host Interface: com Uptime: 3 m, 29 s Eth Link: down Connected: linksys (208)

This is the front page presented by WIF.exe. It gives you access to three top level menus…

- File
- Tools
- Help

The front page also gives you information on…

- Wired Ethernet connection MAC & IP addresses
- 802.11 wireless MAC and IP addresses
- Firmware & Bootloader Versions.
- Region Code (US/CE)
- SPI configuration parameters
- UART configuration parameters

You may click the "Refresh" button at the bottom at any time to read the WISM+ again.

This front page also allows you to select or de-select Bridged mode.

It also allows selection of Infrastructure/Ad-Hoc mode.

You can set the SPI bus and UART parameters.

NOTE:

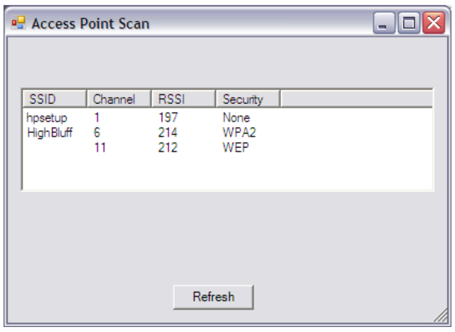
This tool reads and writes the state of the configured parameters stored in the WISM+. Any time you modify parameters, they are written to the WISM+ upon exit of the WIF.exe program. You must then reset the WISM+ for them to take affect, and re-start WIF.exe to view the changes by the configuration tool.

4.4 WIF Access Point Scan

From the WIF front page, pull down the "Tools" menu option and select "Access Point Scan". WISM+ will then go looking for all access points within range and present you with a window like what is shown below…

You will see…

- All access points and Ad-Hoc connections
- Their SSID's
- Their channel
- Associated Signal level (RSSI)
- What type of security

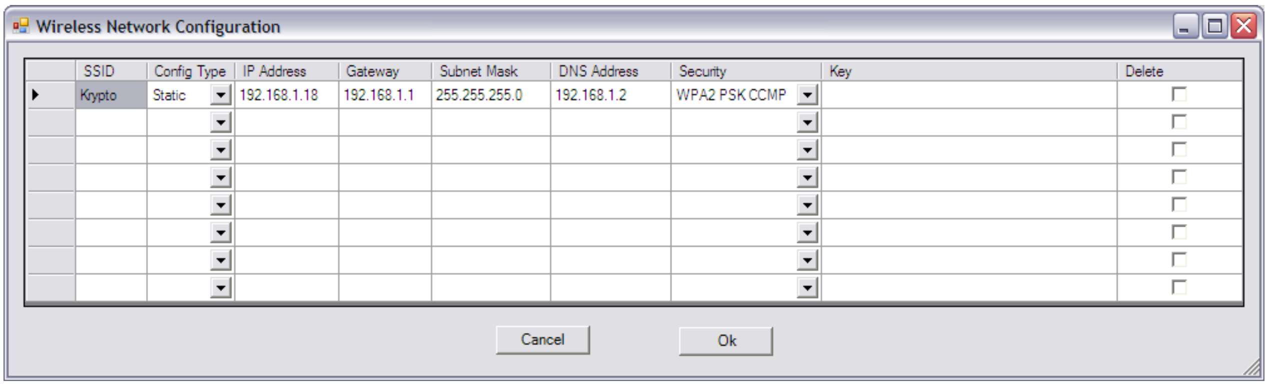

Access Point Scan

SSID	Channel	RSSI	Security
hpsetup	1	197	None
HighBluff	6	214	WPA2
	11	212	WEP

4.5 Configuring WISM+ Access Point Table

From the WIF front page, in the “Wireless LAN” section (upper right), press the “Configure” button and WIF.exe will display the contents of WISM+’s Access Point Table as shown below…

Access Point Table



Wireless Network Configuration

SSID	Config Type	IP Address	Gateway	Subnet Mask	DNS Address	Security	Key	Delete
Krypto	Static	192.168.1.18	192.168.1.1	255.255.255.0	192.168.1.2	WPA2 PSK CCMP		☐
								☐
								☐
								☐
								☐
								☐
								☐
								☐

This page will allow you to describe multiple access points which WISM+ could connect to. SSID, static IP or DHCP, Gateway, Subnet Mask, DSN Address, Security type and Security key parameters can be configured.

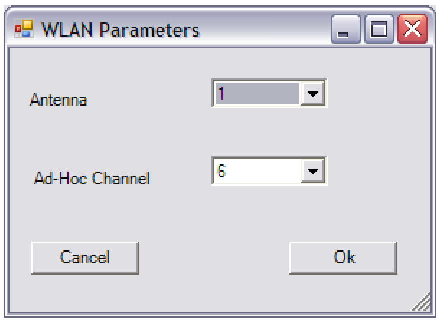
To delete an entry, click the “Delete” box (on the right) and press OK.

Note: Only one Ad-Hoc connection can be described and MUST be configured as the FIRST entry in the table.

By clicking OK, and then exiting WIF.exe, the parameters will be saved to WISM+’s access point table in Non-Volatile memory.

4.6 Configure WISM+ Antenna

From the WIF.exe front page, in the Wireless LAN section (upper right), click the “Advanced” button. You will see the following “WLAN Parameters” window (on the left)…

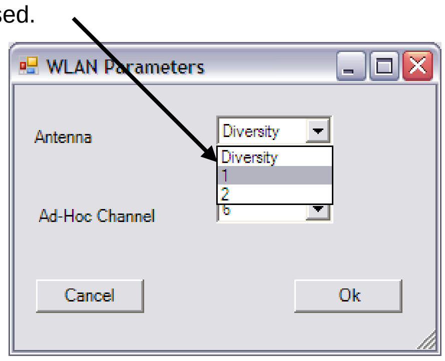


WLAN Parameters

Antenna: 1

Ad-Hoc Channel: 6

By pulling down the Antenna selection box, you can select U.FL port #1, U.FL port #2 or the diversity option in which two antennae MUST be used.



WLAN Parameters

Antenna: Diversity

Once a selection has been made, click OK and exit WIF. Re-start the WISM+ and the selection is permanent.

NOTE: When diversity is selected, the Marvell 8686 antenna diversity algorithm will select which antenna is used for both transmitting and receiving. It constantly evaluates signal to noise ratio and packet error rates and will make decisions to switch antennae. By default, antenna #1 is configured to be used for transmit and receive.

5 Perform a Wireless Ping PC→WISM+

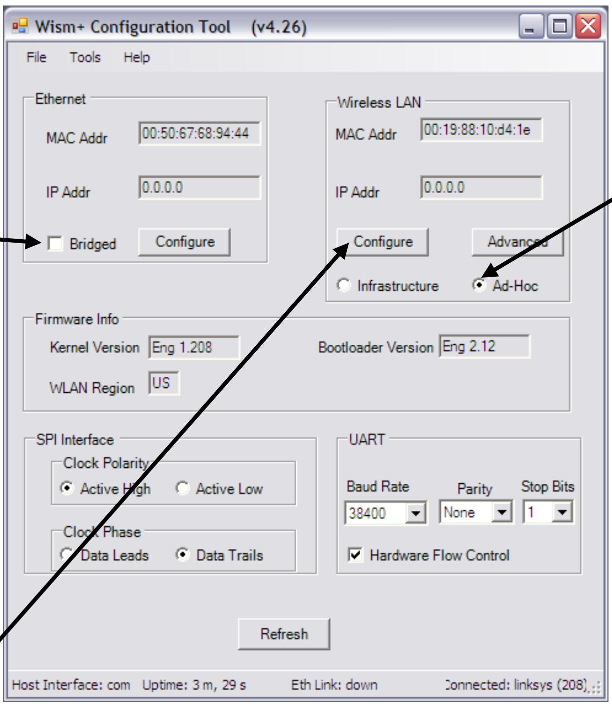
The following section will walk you through the procedure of how to create an Ad-Hoc connection between a WISM+ Dev Kit and a wireless enabled PC in order to perform a wireless network Ping.

The WISM+ can operate either in infrastructure mode (communication with a Wireless Access Point or Router) or in Ad-Hoc mode (with another WISM+, wireless PC in Ad-Hoc mode or any other 802.11b device capable of creating an Ad-Hoc Connection). **An Ad-Hoc mode connection is always 802.11b.**

Launch WIF.exe using the USB connection, and then select the following…

Not "Bridged"

Select "Ad-Hoc"



Ethernet
MAC Addr 00:50:67:68:94:44
IP Addr 0.0.0.0
[ ] Bridged

Wireless LAN
MAC Addr 00:19:88:10:d4:1e
IP Addr 0.0.0.0
Infrastructure [✓ Ad-Hoc]

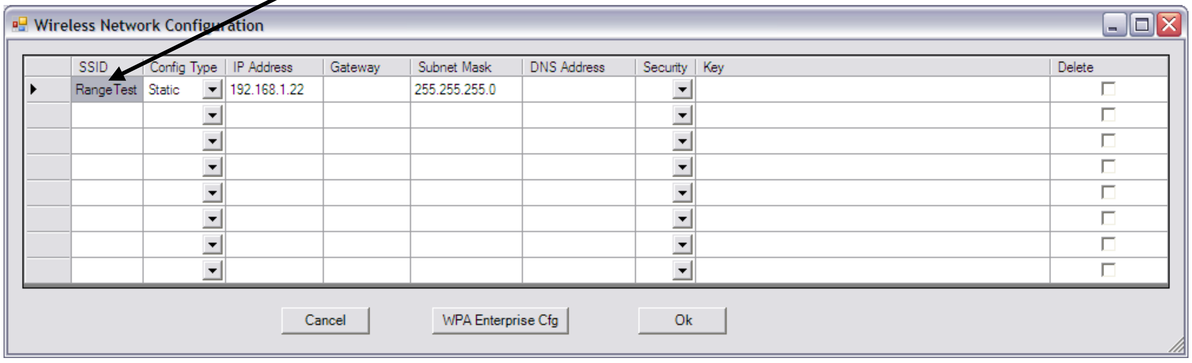
Firmware Info
Kernel Version Eng 1.208
Bootloader Version Eng 2.12
WLAN Region US

SPI Interface
Clock Polarity: [✓ Active High] Active Low
Clock Phase: Data Leads [✓ Data Trails]

UART
Baud Rate 38400
Parity None
Stop Bits 1
[✓] Hardware Flow Control

Host Interface: com Uptime: 3 m, 29 s Eth Link: down Connected: linksys (208)

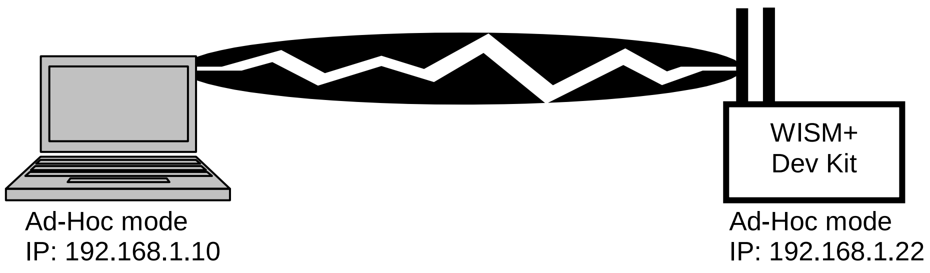
Now click "Configure", and define the following Ad-Hoc static network connection as shown below…



SSID	Config Type	IP Address	Gateway	Subnet Mask	DNS Address	Security	Key	Delete
RangeTest	Static	192.168.1.22		255.255.255.0				[ ]
								[ ]
								[ ]
								[ ]
								[ ]
								[ ]
								[ ]
								[ ]

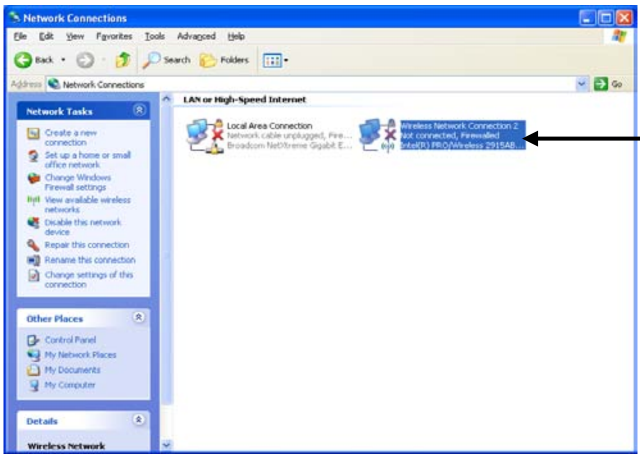
The first access point table entry is the only entry in the table which can be used for an Ad-Hoc connection, and we have set it up with an SSID of "Range Test" with a "Static" IP address of 192.168.1.22 & subnet mask 255.255.255.0. Now click "OK" and then close WIF. Now, reset the WISM+ in order for the changes to be in effect.

We will be setting up the following scenario…



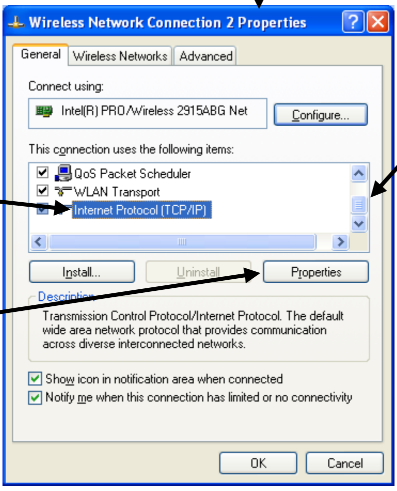
Since the WISM+ has been configured and reset into Ad-Hoc mode, now we have to go to the PC side and configure its Ad-Hoc wireless connection.

Click…Start→Control Panel→Network Connections…


Network Connections
LAN or High-Speed Internet
Local Area Connection
Network cable unplugged, Fire...
Broadcom NetXtreme Gigabit E...
Wireless Network Connection 2
Not connected, Firewalled
Intel(R) PRO/Wireless 2915AB...
Network Tasks
Other Places
Details

Right Click on the Wireless Network Connection and select "Properties".

The screen below will appear…

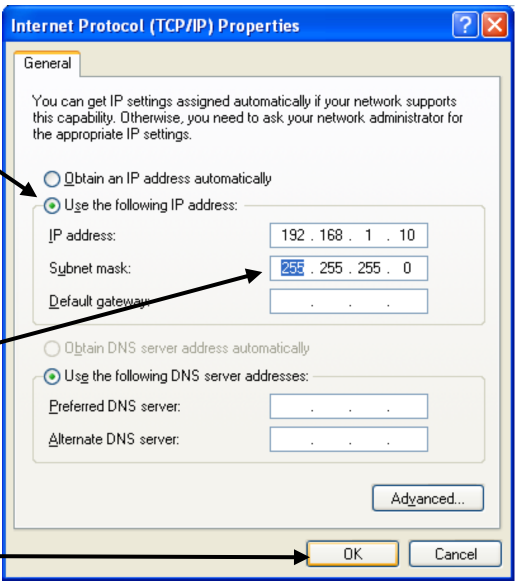

Wireless Network Connection 2 Properties
General
Wireless Networks
Advanced
Connect using:
Intel(R) PRO/Wireless 2915ABG Net
Configure...
This connection uses the following items:
QoS Packet Scheduler
WLAN Transport
Internet Protocol (TCP/IP)
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
Show icon in notification area when connected
Notify me when this connection has limited or no connectivity
OK
Cancel

First, pull the bar down to get to the end of the list

Next, Select "Internet Protocol (TCP/IP)"

Now click "Properties"

The screen to the right will appear, and select "Use the following IP address:"



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
◉ Use the following IP address:

IP address: 192 . 168 . 1 . 10
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
◉ Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

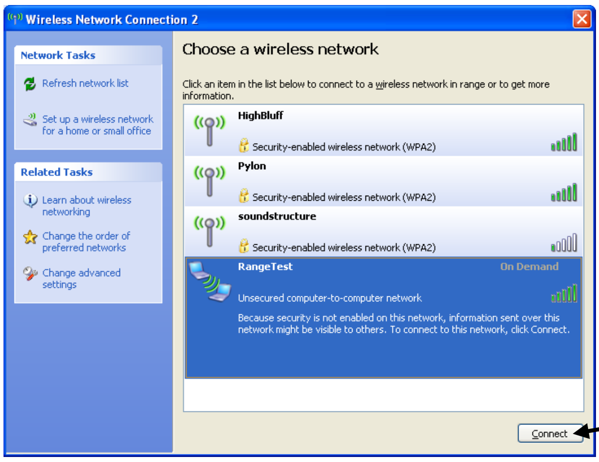
At this point need to give the PC wireless connection its own static IP address (instead of requesting one from a DHCP server).

Fill in the IP address and Subnet mask as shown

When complete, click "OK", and then close the Network connections window.

Now that we have given our wireless connection a static IP address, we are ready to allow the PC and WISM+ to join the Ad-Hoc connection.

Click Start→Connect To, and then select the wireless connection (indicated by the SSID: "Range Test")…



Wireless Network Connection 2

Choose a wireless network

Click an item in the list below to connect to a wireless network in range or to get more information.

HighBluff
Security-enabled wireless network (WPA2)

Pylon
Security-enabled wireless network (WPA2)

soundstructure
Security-enabled wireless network (WPA2)

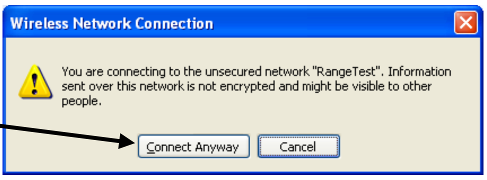
RangeTest — On Demand
Unsecured computer-to-computer network
Because security is not enabled on this network, information sent over this network might be visible to others. To connect to this network, click Connect.

Network Tasks
Refresh network list
Set up a wireless network for a home or small office
Related Tasks
Learn about wireless networking
Change the order of preferred networks
Change advanced settings
Connect

Note:

The Dev Kit must be powered up this point (with WISM+ in Ad-Hoc mode) and must be in range of the PC

Press Connect, and you will get the following warning…

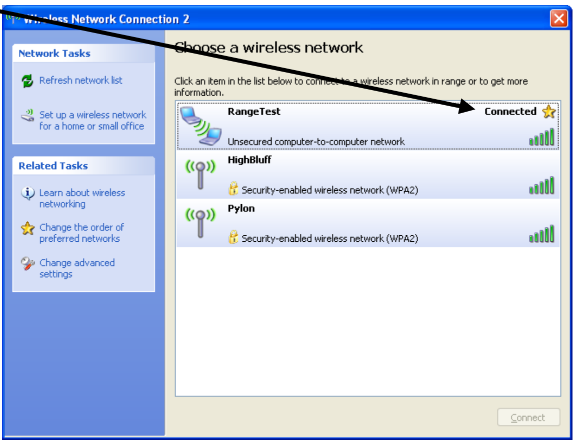


Wireless Network Connection

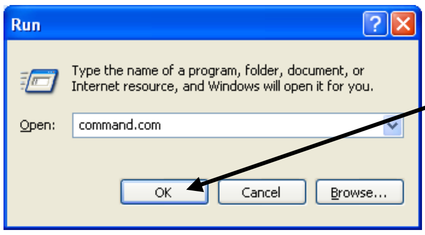
You are connecting to the unsecured network "RangeTest". Information sent over this network is not encrypted and might be visible to other people.

Click "Connect Anyway" and you are now connected to the WISM+ Ad-Hoc connection as shown below…

The following screen appears which confirms that the PC has joined an Ad-Hoc connection with the WISM+.

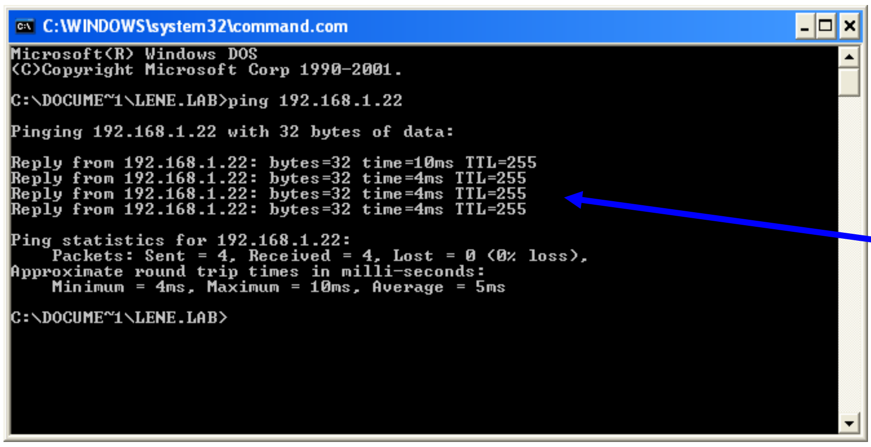

Wireless Network Connection 2
Network Tasks
Refresh network list
Set up a wireless network for a home or small office
Related Tasks
Learn about wireless networking
Change the order of preferred networks
Change advanced settings
Choose a wireless network
Click an item in the list below to connect to a wireless network in range or to get more information.
RangeTest
Connected
Unsecured computer-to-computer network
HighBluff
Security-enabled wireless network (WPA2)
Pylon
Security-enabled wireless network (WPA2)
Connect

Now close this Wireless Network Connection window and Click Start→Run, and type command.com as shown below…


Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
command.com
OK
Cancel
Browse...

Click "OK"

At the DOS command prompt, type "ping" followed by the IP address of the WISM+, and a network ping will be performed between the PC and the WISM+ over the 802.11b connection as shown below…



```
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-2001.

C:\DOCUME~1\LENE.LAB>ping 192.168.1.22

Pinging 192.168.1.22 with 32 bytes of data:

Reply from 192.168.1.22: bytes=32 time=10ms TTL=255
Reply from 192.168.1.22: bytes=32 time=4ms TTL=255
Reply from 192.168.1.22: bytes=32 time=4ms TTL=255
Reply from 192.168.1.22: bytes=32 time=4ms TTL=255

Ping statistics for 192.168.1.22:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 10ms, Average = 5ms

C:\DOCUME~1\LENE.LAB>
```

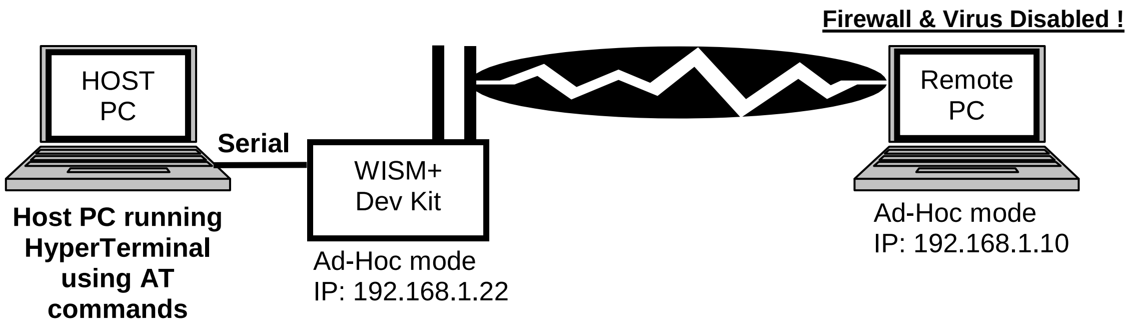
Four pings will be performed by default.

You can also type "ping –help" and see all of the options allowed with the ping command. You can set the number of pings to a large number and then perform range testing.

6 AT mode Wireless Ping WISM+→PC

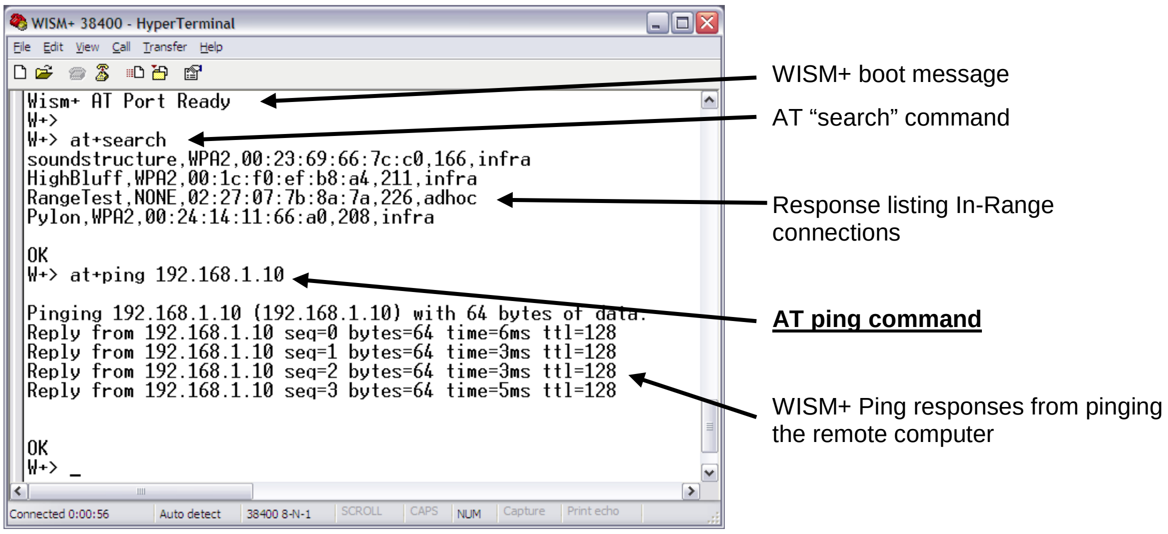
Once you have set up the Ad-Hoc wireless connection as shown in section:

"**Perform a Wireless Ping PC→WISM+**" we can now turn the situation around and connect the WISM+ Dev Kit to a PC over the serial port and use AT commands to ping the PC as shown below…



To set this up…

- Connect the WISM+ Dev Kit Host Com port to the PC serial port (or USB to serial adapter).
- Follow the steps outlined in section: "**Using AT Commands with PC Serial Port**" in order to create a HyperTerminal session which can communicate with the WISM+ over a serial connection.
- Ensure that the HyperTerminal session is successfully communicating with WISM+ by executing a simple AT command.
- **DISABLE THE REMOTE COMPUTER'S FIREWALL**. You may also have to disable your virus protection as described in section: "**Configuring PC Ethernet port for WIF.exe**" on the remote PC.
- Execute the AT Ping command as shown below…


WISM+ 38400 - HyperTerminal
File Edit View Call Transfer Help
Connected 0:00:56
Auto detect
38400 8-N-1
SCROLL
CAPS
NUM
Capture
Print echo
WISM+ boot message
AT "search" command
Response listing In-Range connections
AT ping command
WISM+ Ping responses from pinging the remote computer

```
Wism+ AT Port Ready
W+>
W+> at+search
soundstructure,WPA2,00:23:69:66:7c:c0,166,infra
HighBluff,WPA2,00:1c:f0:ef:b8:a4,211,infra
RangeTest,NONE,02:27:07:7b:8a:7a,226,adhoc
Pylon,WPA2,00:24:14:11:66:a0,208,infra

OK
W+> at+ping 192.168.1.10

Pinging 192.168.1.10 (192.168.1.10) with 64 bytes of data.
Reply from 192.168.1.10 seq=0 bytes=64 time=6ms ttl=128
Reply from 192.168.1.10 seq=1 bytes=64 time=3ms ttl=128
Reply from 192.168.1.10 seq=2 bytes=64 time=3ms ttl=128
Reply from 192.168.1.10 seq=3 bytes=64 time=5ms ttl=128


OK
W+> _
```

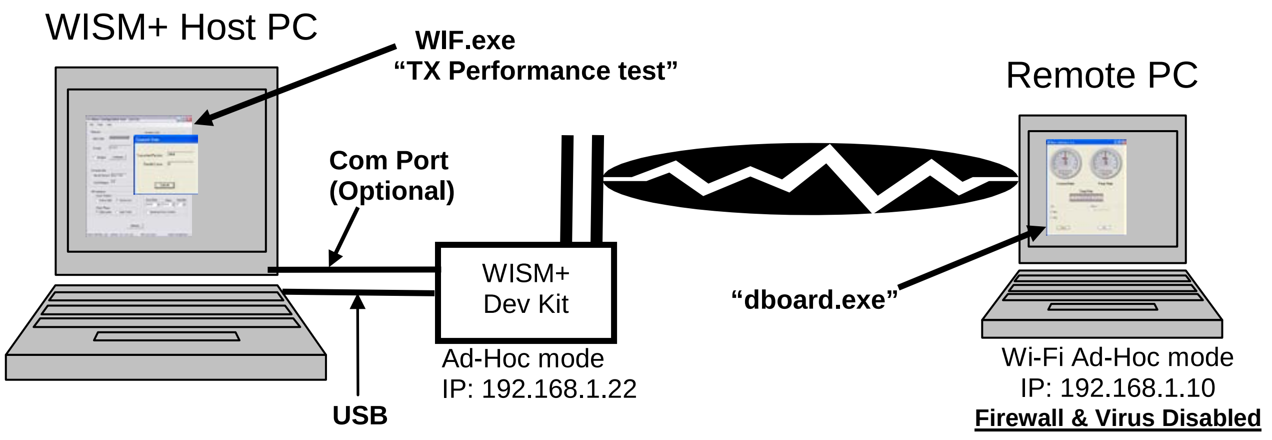
NOTE: In the ping command (shown above), the target of the 'PING' is the IP address of the remote PC configured in Ad-Hoc mode.

7 Using the Performance Tool in Ad-Hoc Mode

The performance of the WISM+ to a Wi-Fi enabled PC in a UDP socket configuration (in 802.11b mode) can be measured (over the USB interface) using WIF.exe's Tx Performance tool feature. From WIF.exe's top menu, select…

Tools→Tx Performance Test

Packets are handed to the WISM+ over the wired active host PC interface, and then transmitted by WISM+ to be received by a remote PCs Wi-Fi (set up with an Ad-Hoc connection) using "dboard.exe" (a separate stand alone program running). Below is a diagram of the setup…



To begin, set up the Ad-Hoc connection between the WISM+ and the remote PC by following the procedure outlined in section: "**Perform a Wireless Ping PC→WISM+**".

Now that the WISM+\Remote PC connection is in place, **launch "dboard.exe" on the remote PC by clicking**…

Start→All Programs→Wism+→DashBoard Performance Monitor

dboard.exe will capture packets & measure throughput performance. Since this the TX Performance tool uses a UDP socket, there is no guaranteed delivery of packets (as with TCP). There may be more total packets transmitted from the WISM+ Host PC Tx Performance test than packets received on the remote PC by dboard.exe.

About dboard.exe…

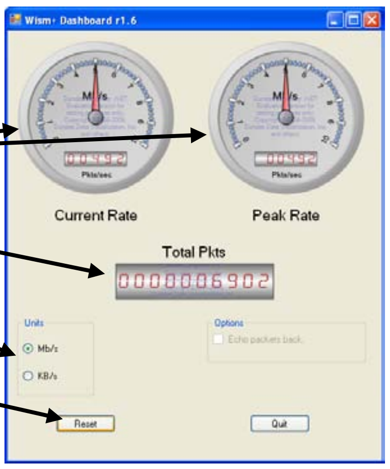
"dboard.exe" is the program which runs on the remote PC and is use to capture the packets sent from the host PC/WISM+ over the air.

It displays instantaneous data rate ("Current Rate") and a latched Peak Rate.

It displays the total number of 1500 byte packets received.

You can scale the units of the meters in kb/s or Mb/s.

You can also reset the peak rate meter and the packet count by pressing the "Reset" button

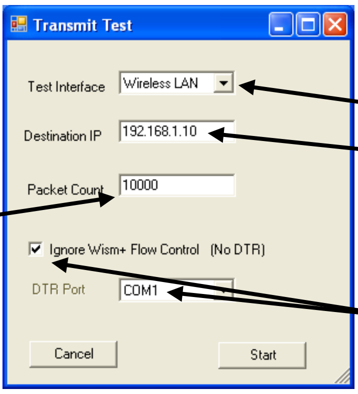


On the WISM+ Host PC side…

Launch WIF.exe with a USB connection (see section: "Software- WIF")

From WIF.exe's top menu, select…**Tools→Tx Performance Test**

The TX Performance test window will appear



Test Interface: Wireless LAN
Destination IP: 192.168.1.10
Packet Count: 10000
☑ Ignore Wism+ Flow Control (No DTR)
DTR Port: COM1

Select "Wireless LAN"

Enter the Remote PCs IP address

Enter the number of packets to transmit

These selections are used to select the PCs com port DTR to wait for the WISM+'s $\overline{\text{DSR\_READY}}$ hold off.

NOTE: See DTR hold off discussion below…

DTR Hold Off function

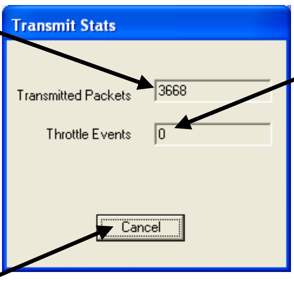
The PCs Host COM port can be set up to use DTR (connected through the Dev Kit to WISM+'s $\overline{\text{DSR\_READY}}$).

When the WISM+ input buffers become "mostly full" of transmit packets, WISM+ drives the $\overline{\text{DSR\_READY}}$ pin to an invalid state. The WISM+ host PC (running WIF.exe "TX Performance Test") will hold off from transmitting packets to the WISM+ until the $\overline{\text{DSR\_READY}}$ becomes "ready". ***If you do not have the serial connection, click the box "Ignore WISM+ Flow Control (No DTR)"***

WIF.exe's Performance TX test "Transmit Status Window

Once you press START (in the above Transmit Test window), packet transmission beings. The following Transmit Status window will be displayed while the packets are being transmitted.

The number of packets successfully transmitted



Transmitted Packets: 3668
Throttle Events: 0

When WIF.exe attempts to transmit a packet to the WISM+ *-and the DTR hold off is active (the "Ignore WISM+ Flow Control" check box in the Transmit test window above- **is not checked**)* -it will be recorded as a **"Throttle Event"**. The will be quickly incremented as long as the DTR Hold-Off condition exists.

Click to Cancel the transmit operation at any time during the test.

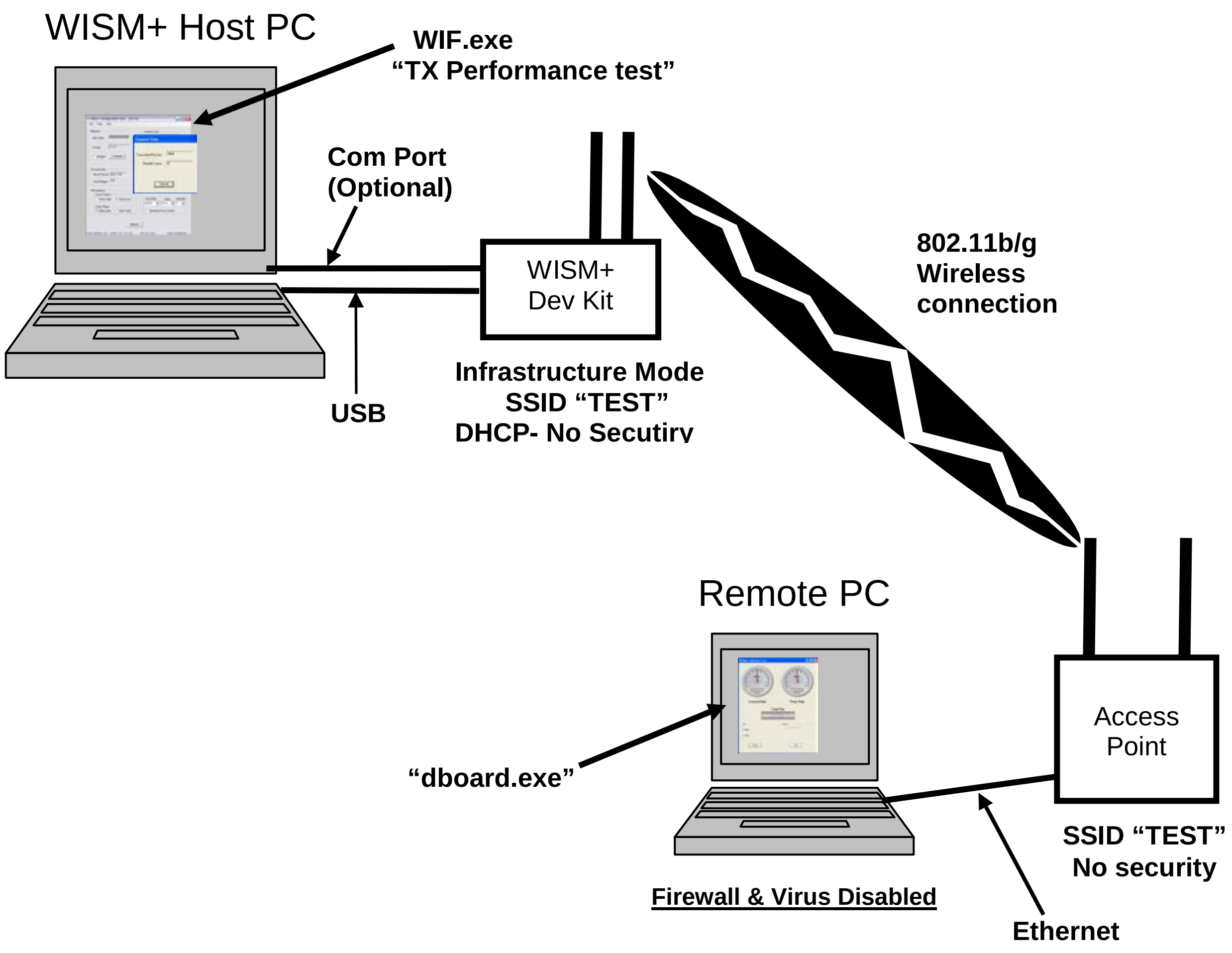
NOTE:

Although the TX performance test can operate using any host interface, serial is very slow, and Ethernet currently has packet transmit problems due to the WinPcap package. SPI works as well, but in all cases, the throughput is held back due to how fast the packets can actually be given to the WISM+.

8 Using the Performance Tool in Infrastructure Mode

The TX Performance tool in WIF.exe is used to send packets to a UDP socket at a destination IP address.

In the example shown below, the WISM+ (in infrastructure mode NOT bridged) is wirelessly connected to the Access Point which has an Ethernet connection to the Remote PC. The Remote PC is running dboard.exe.



See the following page on how to set up the WISM+ configuration.

There are too many combinations of the above diagram to discuss. The Access Point could be a Router, the Remote PC might not be connected directly to the Access Point or Router, but over a network. The PCs could have Static IP addresses, etc.

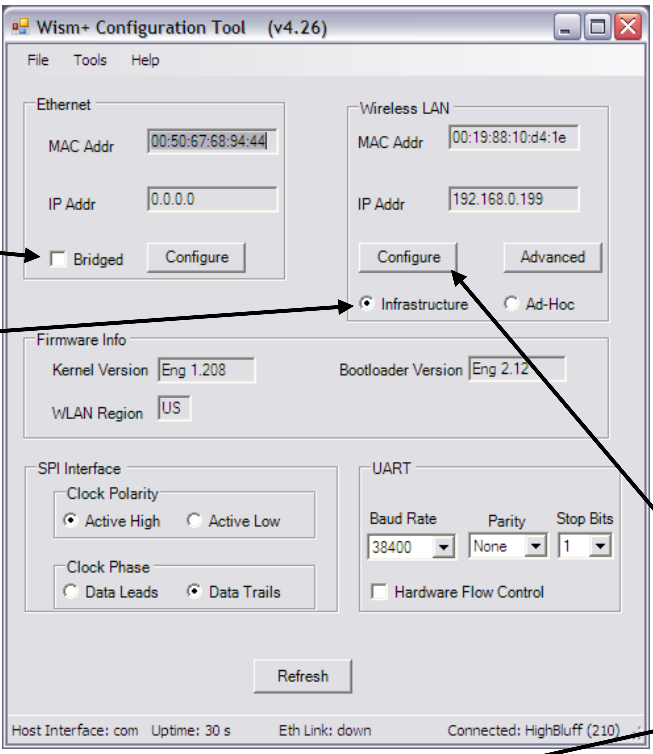
Refer to section: "Using the Performance Tool in Ad-Hoc Mode" for a description of using WIF.exe to transmit packets on the WISM+ Host PC and dboard.exe operation on the Remote PC.

To Configure the WISM+,

Launch WIF.exe (over USB) to configure the WISM+ in “Infrastructure mode” with “Bridge Mode” de-selected as shown below…

Make sure “Bridged” is not selected

Select “Infrastructure”



Ethernet
MAC Addr: 00:50:67:68:94:44
IP Addr: 0.0.0.0
Bridged [ ]

Wireless LAN
MAC Addr: 00:19:88:10:d4:1e
IP Addr: 192.168.0.199
[✓ Infrastructure] Ad-Hoc

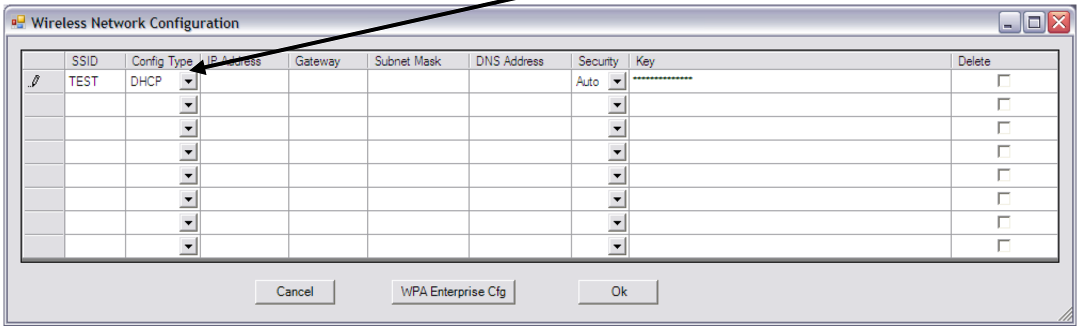
Firmware Info
Kernel Version: Eng 1.208
Bootloader Version: Eng 2.12
WLAN Region: US

SPI Interface
Clock Polarity: [✓ Active High] Active Low
Clock Phase: Data Leads [✓ Data Trails]

UART
Baud Rate: 38400
Parity: None
Stop Bits: 1
Hardware Flow Control [ ]

Host Interface: com Uptime: 30 s Eth Link: down Connected: HighBluff (210)

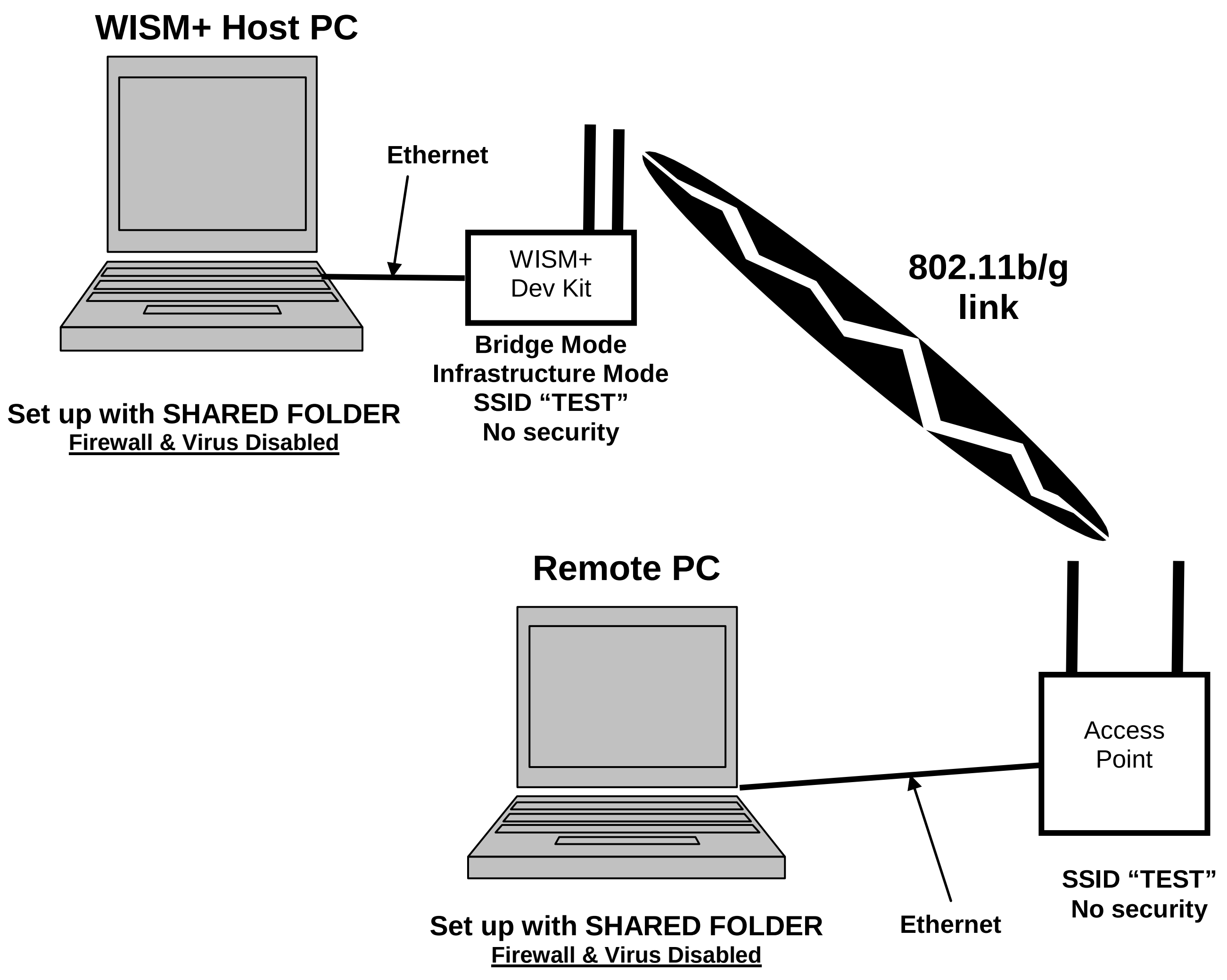
Once Bridged is un-selected and Infrastructure is selected, press the configure button, and set the first access point table entry to match the SSID of the access point with DHCP as shown below...



SSID	Config Type	IP Address	Gateway	Subnet Mask	DNS Address	Security	Key	Delete
TEST	DHCP					Auto	**************	[ ]
								[ ]
								[ ]
								[ ]
								[ ]
								[ ]
								[ ]
								[ ]

9 Using WISM+ in Bridge Mode w/Access Point

In the following scenario, we are going to connect two PCs using a wireless connection between the WISM+ Dev Kit and an access point in 802.11b/g infrastructure mode. The WISM+ will be in Bridged mode and the TCP stack in the PCs will guarantee delivery of the file across the network. We will then copy a large file from one PC to the other across the WISM+/Access point link in order to test network throughput.



To set this up

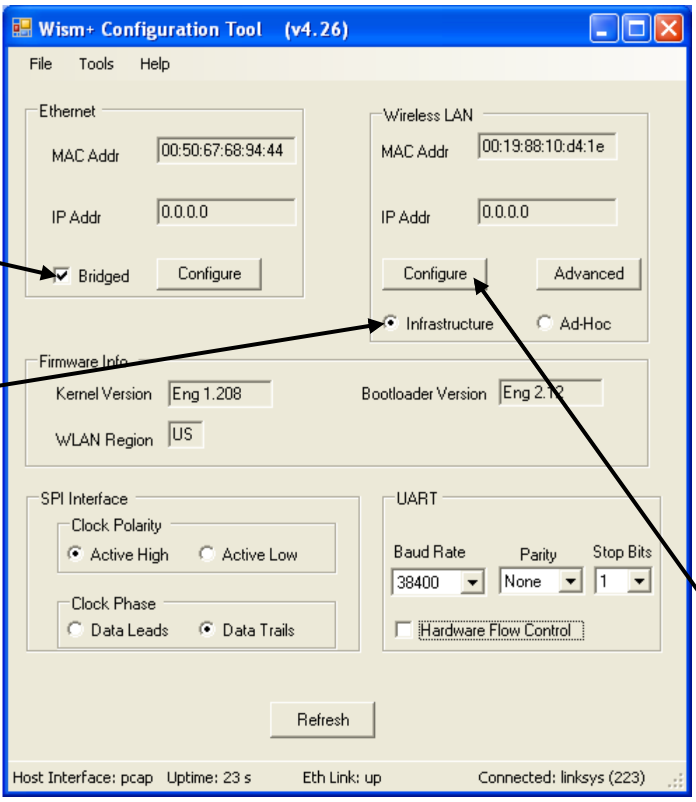
- Connect the access point to the remote PC with an Ethernet cable. Firewall and Virus protection may have to be disabled on both host and remote PCs. Set up a shared folder on both PCs, so each will be able to access the folder on the opposite PC once the wireless connection is established.
- Make sure that both laptops have their TPC properties (for the wired Ethernet connection) set up to obtain an IP address automatically (DHCP).
- Configure the access point to have an SSID (in this example "TEST") and no security.

To Configure the WISM+,

Launch WIF.exe (over USB) to configure the WISM+ in “Infrastructure mode” with “Bridge Mode” selected as shown below…

Select “Bridged”

Select… “Infrastructure”



Ethernet
MAC Addr 00:50:67:68:94:44
IP Addr 0.0.0.0
[✓ Bridged]

Wireless LAN
MAC Addr 00:19:88:10:d4:1e
IP Addr 0.0.0.0
[✓ Infrastructure] Ad-Hoc

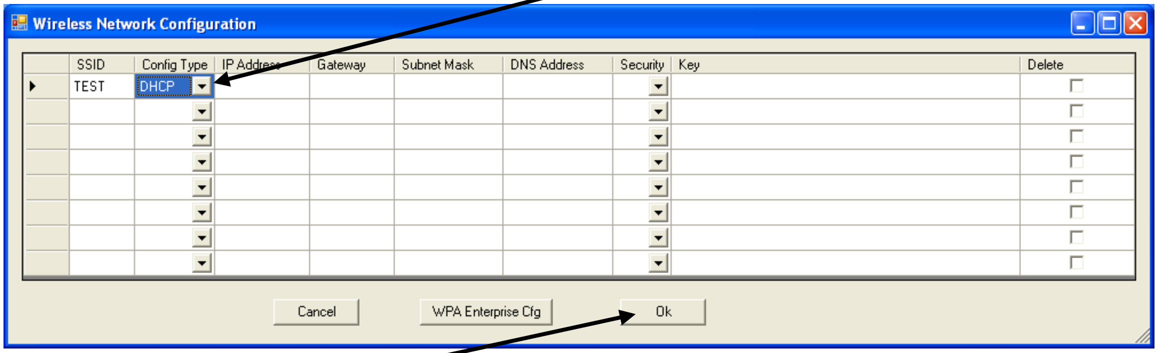
Firmware Info
Kernel Version Eng 1.208
Bootloader Version Eng 2.12
WLAN Region US

SPI Interface
Clock Polarity: [✓ Active High] Active Low
Clock Phase: Data Leads [✓ Data Trails]

UART
Baud Rate 38400
Parity None
Stop Bits 1
Hardware Flow Control

Host Interface: pcap Uptime: 23 s Eth Link: up Connected: linksys (223)

Once Bridged and Infrastructure modes are selected, press the configure button, and set the first access point table entry to match the SSID of the access point with DHCP as shown below...



SSID	Config Type	IP Address	Gateway	Subnet Mask	DNS Address	Security	Key	Delete
TEST	DHCP							

Now, press “Ok”, and then exit WIF.exe, and then re-boot the WISM+ by pressing RESET, or cycling power.

Now connect the WISM+ Dev Kit to the **WISM+ HOST PC** via an Ethernet cable.

When WISM+ connects to the Access point (joins the wireless network) its “WLAN LED” will be lit.

Looking at IP addresses

You can click **“Start→Run** and type **command.com”** to open a dos window on both laptops.

Type **“ipconfig”** at the DOS prompt, and the system will display the IP address which was issued to it…

- The access point will give the “Remote PC” an IP address over the wired Ethernet connection
- The access point will issue the “WISM+ Host PC” an IP address after the WISM+ in bridge mode joins the wireless network.

Performing a file copy to measure throughput

- Create a large file (maybe a hundred Mega-Bytes) and place it in the shared folder.
- On the opposite PC, use Windows Explorer and navigate to view the shared folder on the other PC.
- Drag and drop the large file from the remote shared drive to the desktop and time the network file transfer.
- Now compute the throughput of the system by multiplying the size of the file (in bytes) times 8 (bits per byte) and then dividing by the transfer time in seconds.
- This test uses 802.11g/b and TCP stack on the PCs, performing network level retries and guaranteeing the file delivery.

10 Regulatory Considerations

10.1 US and Canada

The WLM400 module has been approved for FCC & IC. The approvals have been filed with proper US and Industry Canada government agencies (see USER MANUAL for details). The WLM400 module has been tested and found to comply with limits for a Class B digital device. The Dev Kit has been found to comply with limits for a Class A digital device.

NOTE:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense

This Class A digital apparatus complies with Canadian ICES-003.

[Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.]

10.2 Europe

The WLM402 (on the Dev Kit board) has been tested and found to comply with the following CE standards…

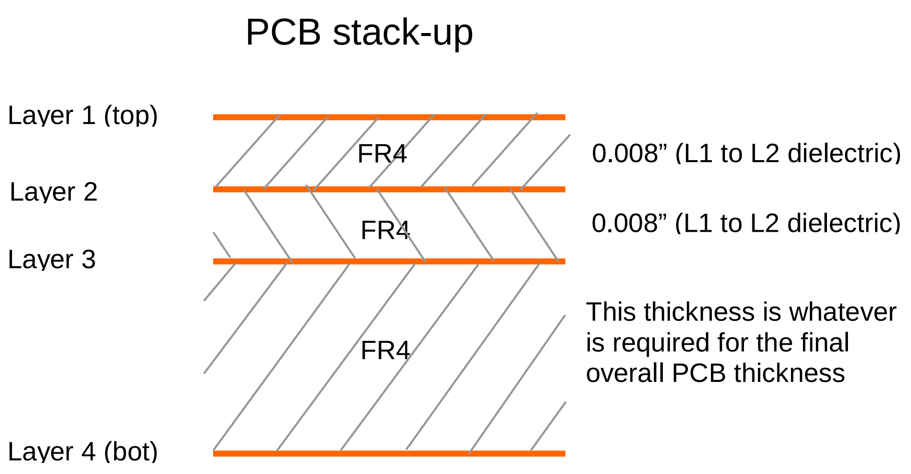
- ETSI EN 301 489-1 V1.8.1 (2008-04)
- ETSI EN 301 489-17 V1.2.1 (2002-08)
- ETSI EN 300 328-2 V1.7.1 (2006-05)

The test reports are available upon request.

11 Dev Kit Design

11.1 PCB Stack-Up

The Dev Kit Printed Circuit Board (PCB) is a 4 layer PCB made with standard FR4 material. The solder mask technology is SMOBC. The diagram below shows the basic PCB stack-up for the 4 layer PCB.

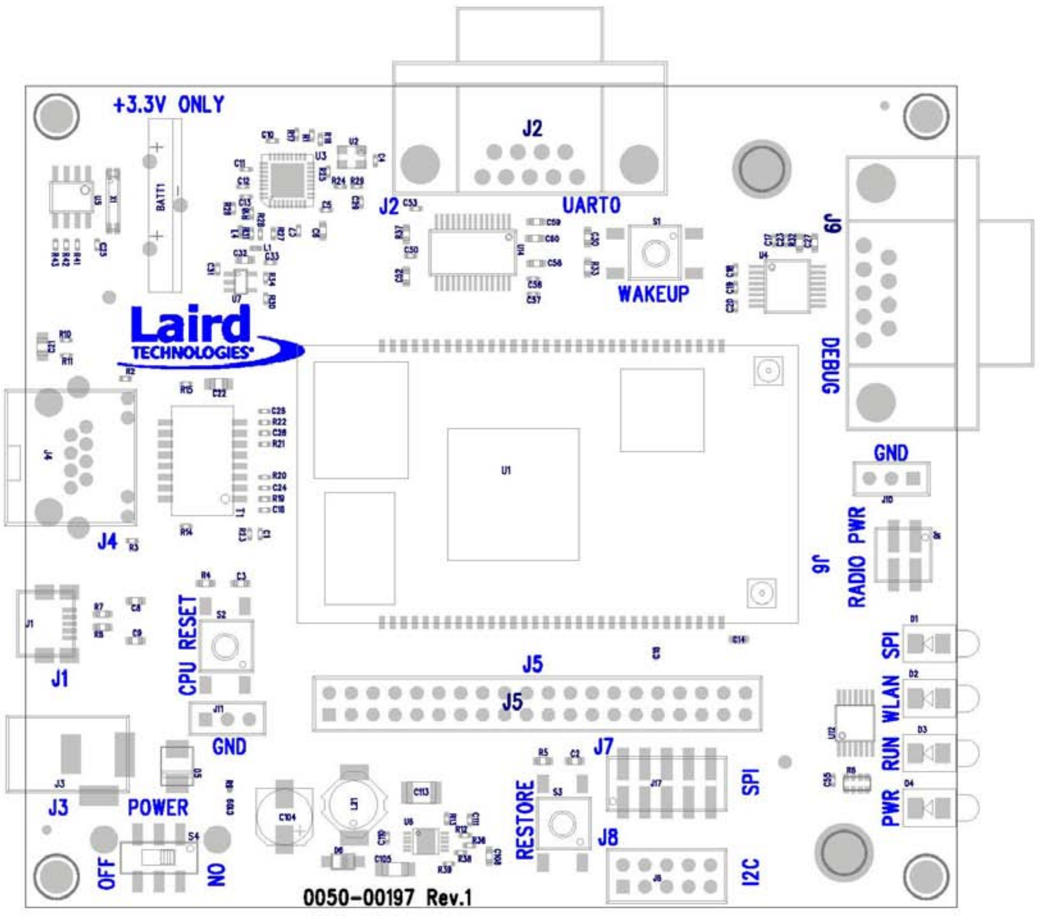

PCB stack-up
Layer 1 (top)
Layer 2
Layer 3
Layer 4 (bot)
FR4
FR4
FR4
0.008" (L1 to L2 dielectric)
0.008" (L1 to L2 dielectric)
This thickness is whatever is required for the final overall PCB thickness

The Ethernet PHI interface requires good PCB design practices in order to yield acceptable EMI suppression. Use of ground plane which encloses noisy signals and clocks is imperative in order to produce a quiet design.

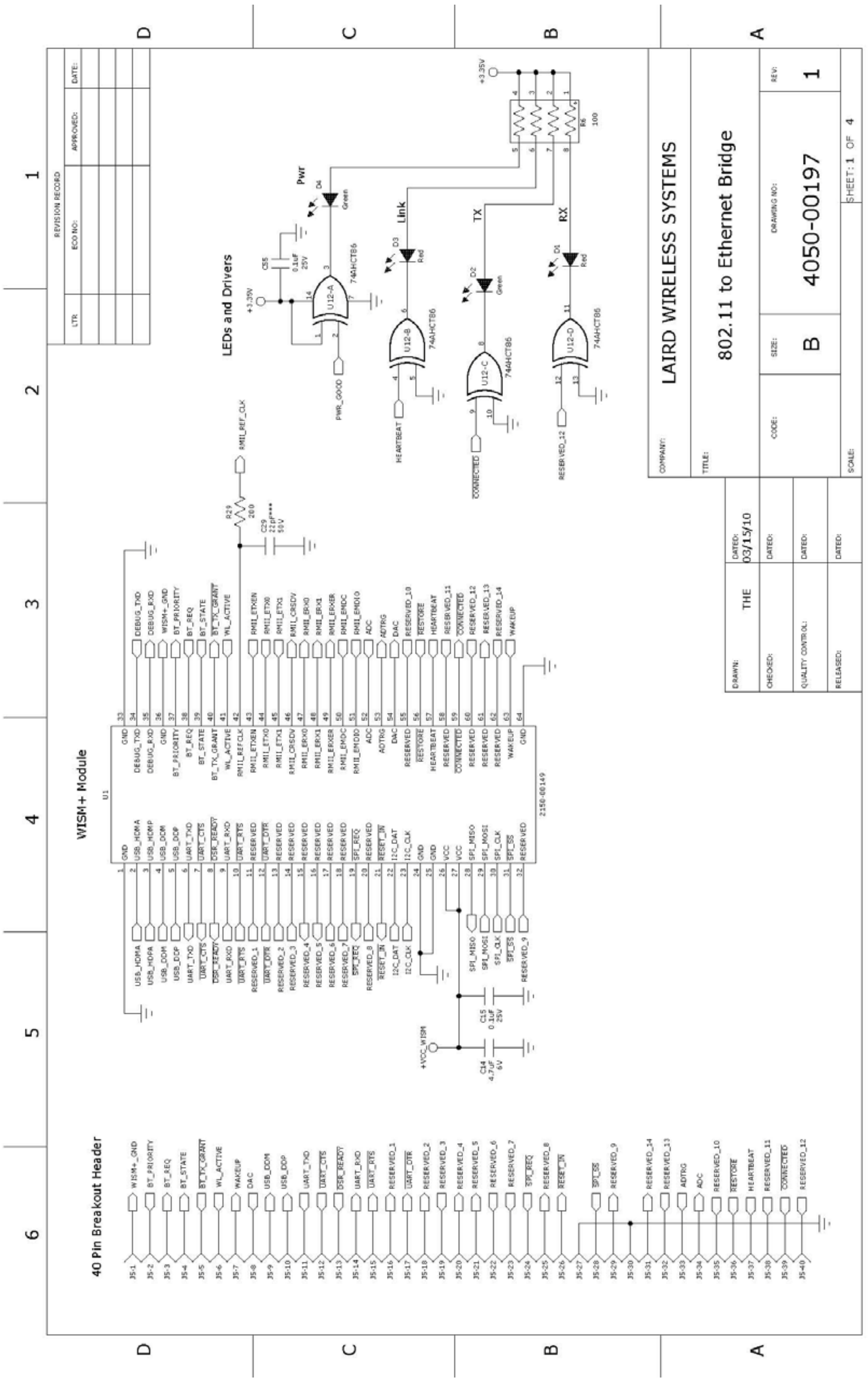
Bill of Materials and Gerber Files are available upon request.

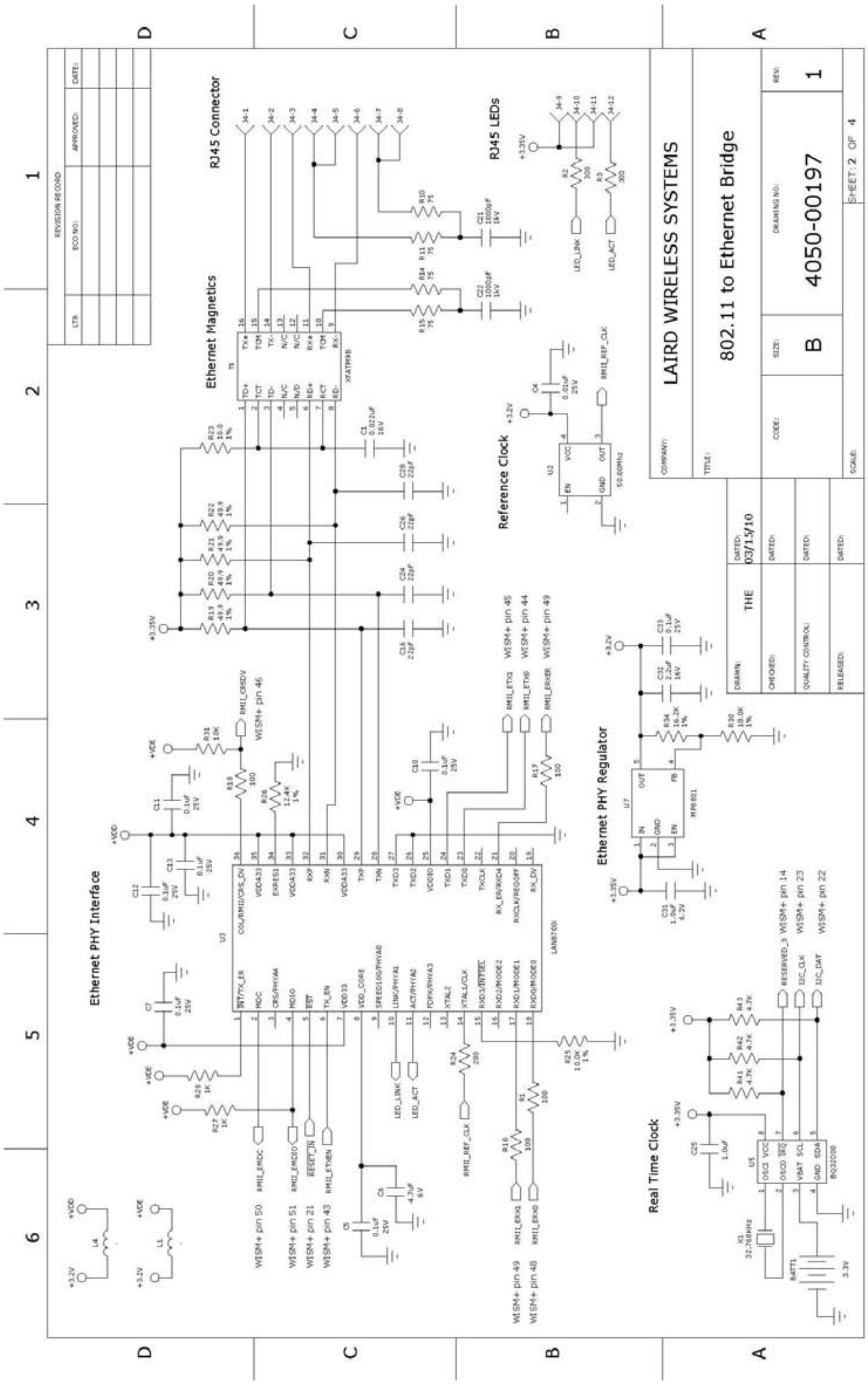
11.2 PCB Layout Drawing

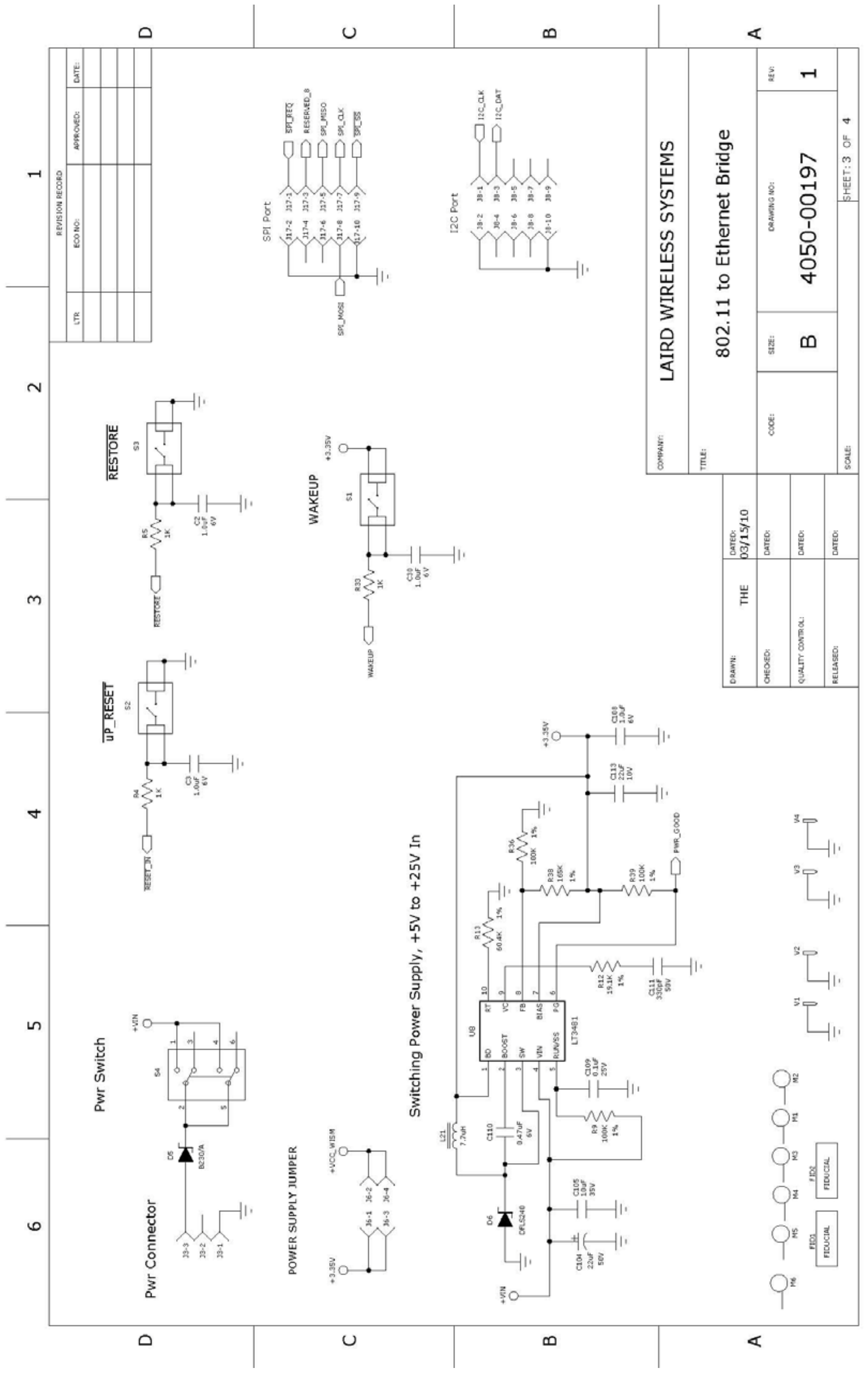
The following is an assembly drawing of the Dev Kit PCB indicating reference designators and parts placements. The corresponding schematics are shown in the following section.

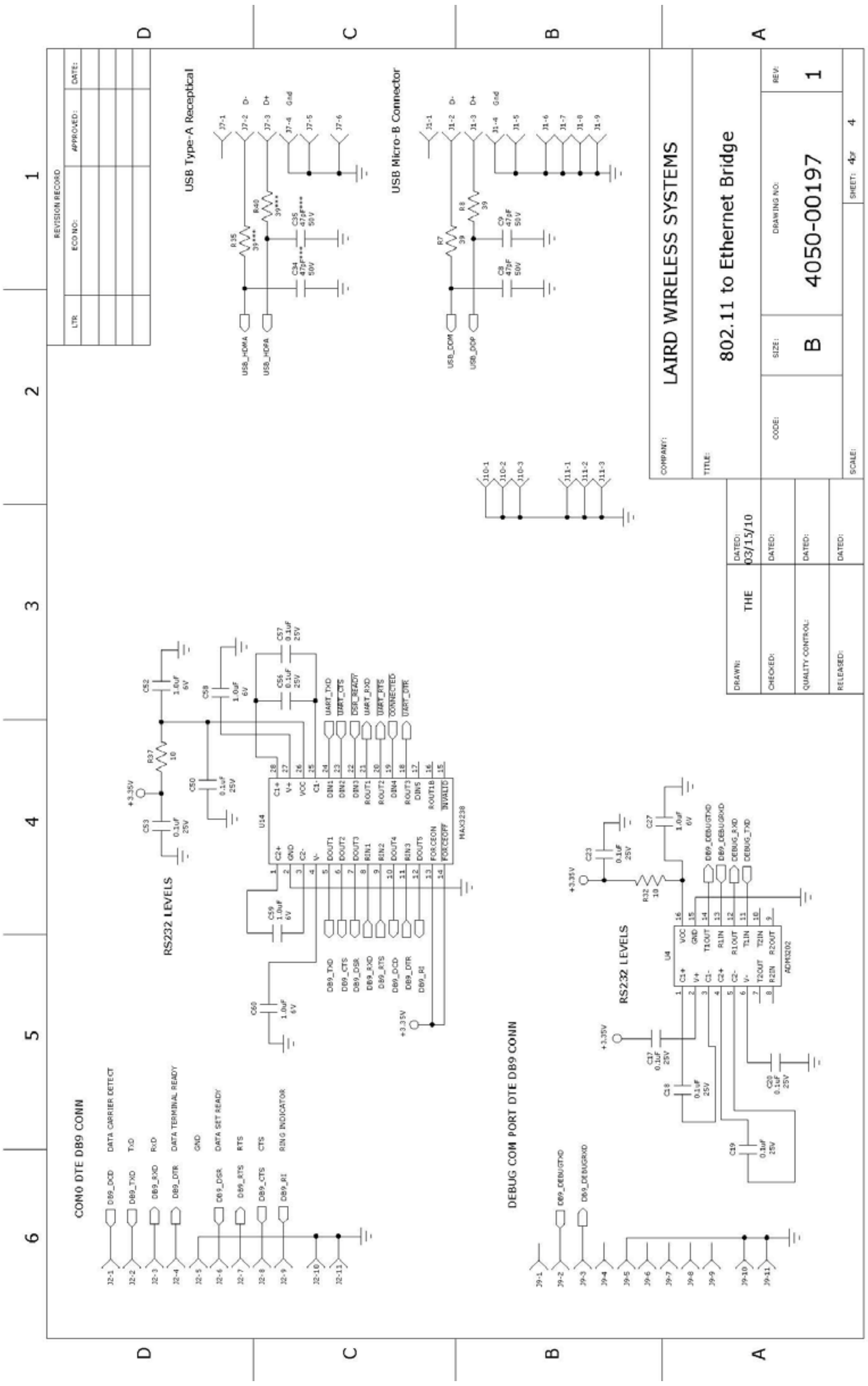


11.3 Schematics




RJ45 Connector
RJ45 LEDs
Ethernet Magnetics
Reference Clock
Ethernet PHY Interface
Ethernet PHY Regulator
Real Time Clock
LAIRD WIRELESS SYSTEMS
802.11 to Ethernet Bridge
4050-00197
SHEET: 2 OF 4
THE
03/15/10
LAN8700i
XFATM9B
MP8801
BQ32000
50.00Mhz
32.768KHz


LAIRD WIRELESS SYSTEMS
802.11 to Ethernet Bridge
DRAWING NO: 4050-00197
SIZE: B
REV: 1
SHEET: 3 OF 4
DRAWN: THE
DATED: 03/15/10
REVISION RECORD
SPI Port
SPI_REQ
RESERVED_8
SPI_MISO
SPI_CLK
SPI_SS
SPI_MOSI
I2C Port
I2C_CLK
I2C_DAT
RESTORE
uP_RESET
WAKEUP
RESET_IN
+3.35V
Pwr Switch
Pwr Connector
POWER SUPPLY JUMPER
+VIN
+VCC_WISM
Switching Power Supply, +5V to +25V In
U8
LT3481
PWR_GOOD
FIDUCIAL


LAIRD WIRELESS SYSTEMS
802.11 to Ethernet Bridge
DRAWING NO: 4050-00197
SIZE: B
REV: 1
SHEET: 4 OF 4
DRAWN: THE
DATED: 03/15/10
REVISION RECORD
USB Type-A Receptical
USB Micro-B Connector
RS232 LEVELS
U14 MAX3238
U4 ADM3202
CON0 DTE DB9 CONN
DEBUG COM PORT DTE DB9 CONN
DATA CARRIER DETECT
TxD
RxD
DATA TERMINAL READY
GND
DATA SET READY
RTS
CTS
RING INDICATOR