WiLink™ 8 Adaptor
Getting Started Guide
Version 1.6
## Prior Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>October 30, 2014</td>
<td>Initial version - Rev.A board</td>
</tr>
<tr>
<td>1.1</td>
<td>October 31, 2014</td>
<td>Content changes, Rev.A board</td>
</tr>
<tr>
<td>1.2</td>
<td>December 02, 2014</td>
<td>Content changes, Rev.B board</td>
</tr>
<tr>
<td>1.3</td>
<td>December 04, 2014</td>
<td>Updated photos, Bluetooth quick start added</td>
</tr>
<tr>
<td>1.4</td>
<td>December 10, 2014</td>
<td>Updates - startup script section and photos</td>
</tr>
<tr>
<td>1.5</td>
<td>December 15, 2014</td>
<td>Miscellaneous corrections (WiLink™ 8)</td>
</tr>
<tr>
<td>1.6</td>
<td>December 22, 2014</td>
<td>BT changes (Vivado and Test), link corrections</td>
</tr>
</tbody>
</table>
Contents

1 Getting started with WiLink™ 8 and Zynq SoC boards .................................................. 4
2 What is needed to develop with WiLink™ 8 on Zynq? ..................................................... 5
3 What’s on the Web? ............................................................................................................ 6
  3.1 Official Documentation.................................................................................................. 6
  3.2 Tutorials and Reference Designs.................................................................................. 6
  3.3 Training and Videos...................................................................................................... 6
4 Overview of the WiLink™ 8 Wireless Solution ................................................................. 7
  4.1 Features....................................................................................................................... 7
  4.2 Applications................................................................................................................ 7
  4.3 TI Module Key Benefits.............................................................................................. 7
5 WiLink™ 8 Embedded Linux Development Environment ................................................ 9
  5.1 Install VM Image containing MicroZed Yocto recipes .................................................. 9
  5.2 Configure and Build the Device Tree.......................................................................... 12
  5.3 Configure and Build the Linux Kernel......................................................................... 12
  5.4 Startup Script for auto-connect to a specific WLAN Access Point ............................. 12
  5.5 Lighttp Webserver and the adding of Custom Web Pages ........................................... 14
  5.6 Hardware Setup.......................................................................................................... 15
6 Getting Help and Support .................................................................................................. 17
  6.1 Avnet Support............................................................................................................. 17
  6.2 Xilinx Support............................................................................................................ 17
7 Appendix A: Installing the ArchiTech Yocto SDK .......................................................... 18
8 Appendix B: Building the Hardware Platform using Vivado ............................................. 20
  8.1 IP Integrator: Adding the WLAN interface (SD1 and MIO GPIO)............................... 21
  8.2 IP Integrator: Adding Bluetooth (UART0 and EMIO GPIO)....................................... 22
  8.3 Vivado: Implementation, Bitstream and Export to SDK ............................................ 23
  8.4 SDK: Generate an updated Zynq Boot Image ......................................................... 23
9 Appendix C: Wi-Fi Quick Start instructions (MicroZed only, pre-built Linux files) .......... 24
10 Appendix D: Bluetooth Quick Start instructions ............................................................ 26
11 Appendix E: Demo List on WiLink™ 8 Wiki Pages ........................................................ 27
  11.1 WiLink™ 8 Wiki: Landing Page.................................................................................. 27
  11.2 Wi-Fi Use Cases....................................................................................................... 27
12 Appendix F: App Development in ArchiTech’s Yocto SDK .......................................... 28
1 Getting started with WiLink™ 8 and Zynq SoC boards

- The WiLink8 Adaptor is a small COM8 to dual Pmod adaptor card that facilitates use of TI’s WL1835MODCOM8 module with “Zed-series” Avnet boards
- TXS0108E level-shifters on the adaptor board ensure that only 1.8V signaling is applied to the I/O pins of the wireless module (this permits the adaptor to be plugged-into any Pmod connector, regardless of which Vccio voltage is in use)
- WLAN uses the PS Pmod interface for best possible throughput, via a predefined MIO pinout for 4-bit wide SDIO interface clocked at 50MHz
- BT/BLE uses a slower, UART based PL interface, implemented on the Pmod-JK connector located adjacent to PS Pmod connector on MicroZed’s IO Carrier Card.
- The WL1835 COM8 module includes on-board chip antennas (and uFL connectors for optional external antennas) and is FCC/ICC certified.

Figure 1 – WL1835 Eval board plus WiLink™ 8 Adaptor (Rev.B board)

This Getting Started Guide briefly describes the hardware setup and then details the steps required to build the necessary support for the two wireless interfaces into a MicroZed reference design, based on the Xilinx Open Source Linux distribution.
2 What is needed to develop with WiLink™ 8 on Zynq?

The recommended full test configuration (using an IO Carrier Card) facilitates development and test of applications that use Wi-Fi and/or Bluetooth:

A. Texas Instruments WL1835MODCOM8B wireless evaluation board
B. Avnet WiLink™ 8 Pmod Adaptor Board (only the WLAN connector is attached)
C. MicroZed IO Carrier Card (optional for WLAN, required if Bluetooth to be used)
D. MicroZed 7Z010 or 7Z020 SOM board

In addition the following will also be required:

- micro-USB-B to USB-A cable
- Universal Power Adapter (only if MicroZed IO Carrier Card is used)
- Vivado Design Edition - Device Locked
- WiLink™ 8 Adaptor Quick Start Card
- Downloadable documentation and reference designs
3 What’s on the Web?
MicroZed is a community-oriented kit, with all materials being made available through the www.microzed.org community website.

3.1 Official Documentation
- Getting Started Guide
- Hardware User Guide
- Schematics
- Bill of materials
- PCB Layout

3.2 Tutorials and Reference Designs
- Design examples based on MicroZed, including:
  - WiLink™ 8 Wi-Fi Webserver using xlx-linux and lighttpd
  - WiLink™ 8 BT/BLE using xlx-linux and BlueZ stack (available Q1, 2015)
- Community projects

For the latest tutorials and reference designs:
1. Go to www.microzed.org
2. Click Support ➔ Reference Designs/Tutorials
3. Click on the Wi-Fi and Bluetooth View All button.

3.3 Training and Videos
- Overview of MicroZed
- Introduction to Zynq
- Implementing Linux on the Zynq-7000 SoC

For the latest training and videos:
- Go to www.microzed.org
- Click Support ➔ Trainings and Videos
4 Overview of the WiLink™ 8 Wireless Solution

4.1 Features

The WiLink8 Adaptor facilitates use of TI’s WL1835MODCOM8B Wi-Fi & BT/BLE module. Features of this wireless solution include the following:

- WLAN, Bluetooth, BLE on a module board
- WLAN 2.4 GHz SISO (20- and 40-MHz channels), 2.4-GHz MIMO (20-MHz channels)
- Support for BLE dual mode
- Seamless integration with Xilinx Zynq, TI Sitara and other application processors
- Can be used with Avnet Zynq based boards or TI AM335X general-purpose EVM
- WLAN and Bluetooth, BLE cores are software and hardware compatible with prior WL127x, WL128x and CC256x offerings, for smooth migration to device.
- Shared HCI transport for Bluetooth and BLE over UART and SDIO for WLAN.
- Wi-Fi / Bluetooth single antenna co-existence
- Built-in chip antennas plus U.FL RF connectors for optional external antennas
- Supports wide input voltage range (4.8-V to 2.9-V)
- VIO in the 1.8-V domain

4.2 Applications

- Internet of Things (IoT) Multimedia
- Industrial and Home Automation
- Video Camera, Security and M2M Industrial Applications

4.3 TI Module Key Benefits

- Reduces Design Overhead: Single WiLink8™ Module Scales Across Wi-Fi and Bluetooth.
- WLAN High Throughput: 80 Mbps (TCP), 100 Mbps (UDP)
- Bluetooth 4.0 + BLE (Smart Ready, supports BT dual mode)
- Low Power (30–50% Less than Previous Generation)
- Built-in chip antenna plus U.FL RF connectors for optional external antennas
- Dual-band 2.4GHz/5GHz plus Industrial temperature version modules also available
- Available as Easy-to-Use FCC, ETSI, and Telec Certified Module
- Lower Manufacturing Costs, Saving Board Space and Minimizing RF Expertise
- Linux Reference Design accelerates development and Time to Market
Figure 3 – WiLink™ 8 module details

Figure 4 – WiLink™ 8 Adaptor - Block Diagram
5 WiLink™ 8 Embedded Linux Development Environment

A scripted build is provided for use with the Yocto SDK from ArchiTech (Avnet Silica) Use of Yocto significantly reduces user effort in building the Embedded Linux solution. Supplied as a VirtualBox VM image download, the SDK is provided at no cost and is composed of several components including:

- Ubuntu Linux host platform (setup of a separate Linux host is no longer required)
- Pre-installed Yocto (plus meta-layers from ArchiTech, Xilinx and other key suppliers),
- Yocto HOB tool (GUI tool to configure the build and manage package selection)
- Pre-configured BSPs for multiple boards (ArchiTech BSPs are ‘Open Source’ and can be modified for your own custom board)
- Pre-installed SDK Tools, pre-configured Eclipse Software Development IDE (eliminates complicated setup and Application development can be started immediately)
- Qt Creator and Qt libraries,
- Java

ArchiTech SDK provides:

- A common development flow across multiple boards from different vendors
- Quick and easy creation of boot loader, kernel and root file system
- Reduced effort to reconfigure Linux kernel and rootfs to your own requirements

5.1 Install VM Image containing MicroZed Yocto recipes

   See Appendix A of this document for notes on download plus installation of:
   a. VirtualBox application (plus VirtualBox Extension Pack)
   b. ArchiTech Yocto SDK VirtualBox VM image
   Note: the Virtual Machine that you create must be named “architech”!
2. Using the “2.2 Build” section of the “Quick Start” Wiki pages as guidance, proceed with steps 1 through 9 of the Quick Start Wiki (do not run step 10!)
3. Launch the Virtual Machine by clicking Start on the toolbar:

   ![VirtualBox Manager](image)

4. Once the VM has loaded, double-click the ArchiTech SDK icon on the Ubuntu VM desktop to bring-up the ArchiTech Splash Screen application.
5. When prompted with ArchiTech or 3rd Party, choose ArchiTech.
6. Select MicroZed as the board you want to work on
7. Click the “Run bitbake” button on the Splash screen. This step configures the default settings and launches a terminal from which you can enter commands.
8. From the command line open Microzed’s local.conf file in the gedit texteditor:
   gedit conf/local.conf
9. Add the following 2 lines to the end of this file: `EXTRA_IMAGE_FEATURES_append = "tools-debug debug-tweaks" IMAGE_INSTALL_append = "tcf-agent"

Save the file and close gedit

10. **Do not** run the command-line bitbake build at this time! (step 10 of the Build Wiki)

11. The VM was created with an English UK keyboard layout by default. This may be changed as follows:
   - Select System Settings (from ‘Gear wheel’ symbol at top right of screen)
   - Select Keyboard Layout
   - Press the "+" button in lower left corner to add a new layout
   - Select **English US** (or whatever your preference) then click the "Add" button
   - Select the previous (no longer needed) English UK keyboard layout
   - Press the "-" button in lower left corner to delete the old keyboard layout

5.1.1 **File downloads and running the script files**

12. From the **WiLink™ 8 Pmod Adaptor** product area of the MicroZed.org website:
   - [http://microzed.org/pmodwifi-wilink8](http://microzed.org/pmodwifi-wilink8)
   - [http://microzed.org/product/wilink-8-adaptor](http://microzed.org/product/wilink-8-adaptor)

download the script files to the ~/Documents staging area on your Linux host.

<table>
<thead>
<tr>
<th>Script File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>microzed-wilink8-ramdisk-</td>
<td>quick SD card creation using precompiled files</td>
</tr>
<tr>
<td>microzed-wilink8-ramdisk.sh</td>
<td>master script file</td>
</tr>
</tbody>
</table>

13. Make sure these files are executable as follows: `chmod +x microzed-wilink8-ramdisk-SDcard.sh` `chmod +x microzed-wilink8-ramdisk.sh`

14. `microzed-wilink8-ramdisk-SDcard.sh`

   For a quick check of functionality using WiLink™ 8 and MicroZed, use this script file to download and program the necessary files onto a microSD card. Launch as follows:
   - `cd ~/Documents`
   - `.microzed-wilink8-ramdisk-SDcard.sh 7010`

   (Command line parameter of 7010 or 7020 must be specified, to designate which BOOT.bin file is written to the microSD card. - See Appendix C for full instructions)

15. **microzed-wilink8-ramdisk.sh**

   **Caution:** The first time execution of this “master” script is very time-consuming (*several hours* required). It is partitioned into 8 sections (some of which launch other script files). A better approach is to selectively run one section at a time, check the console messages are clear of errors, then proceed to the next section.

16. For the lengthy first build, it is important that you:
   - Allocate an uninterrupted time (eg. overnight) for the timeconsuming sections
   - Reduce the build time by:
     - a) ensuring that you have a fast internet connection and
     - b) maximizing the number of CPUs and threads that are used.
17. Multiprocessing is optimized using the following two methods:
   a. Change the VM machine settings to make use of all available CPU cores (*must* be done prior to launching the VM!)

   ![Image of architech-sdk_110 - Settings](image)

   b. When editing the **local.conf** file for your MicroZed build (see Appendix A), edit the following two lines according to how many CPU cores are available:

   ```
   BB_NUMBER_THREADS = "4"
   PARALLEL_MAKE = "-j 4"
   ```

   The applicable local.conf can be opened in an editor using the following command:

   ```
   gedit ~/architech_sdk/architech/microzed/yocto/build/conf/local.conf
   ```

18. **Note:** If all sections of the master script have previously been executed, future builds will be much quicker, only requiring execution of the latter sections of this file.

The downloaded script files called during the build process are as follows:

<table>
<thead>
<tr>
<th>Script File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>microzed-wilinx-</td>
<td>master script file</td>
</tr>
<tr>
<td>uz_wl8_TI_software_pack</td>
<td>WL18xx software package script (WiLink™ 8 driver and firmware)</td>
</tr>
<tr>
<td>build_wl18xx.sh</td>
<td>WL18xx WLAN System Build Utilities script</td>
</tr>
<tr>
<td>uz_wl8_file_system.sh</td>
<td>patch local_conf, then bitbake a core-image-minimal file system</td>
</tr>
<tr>
<td>uz_wl8_boot_loader.sh</td>
<td>u-boot build (xilinx-v2014.2) from u-boot-xlnx repository</td>
</tr>
<tr>
<td>uz_wl8_kernel.sh</td>
<td>kernel build (xilinx-v2014.2.01) from linux-xlnx repository</td>
</tr>
<tr>
<td>uz_wl8_ramdisk.sh</td>
<td>ramdisk build: adds startup script, webserver pages, Wilink8 firmware</td>
</tr>
</tbody>
</table>

The Linux OS distribution and versions used at this time are as follows:

<table>
<thead>
<tr>
<th>Target OS</th>
<th>TI driver version</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux-xlnx (xilinx-v2014.2.01 checkout tag)</td>
<td>ol_r8.a9.14</td>
</tr>
</tbody>
</table>
5.2 Configure and Build the Device Tree

The `uz_wl8_kernel.sh` script automatically downloads the .dts source and builds a preconfigured devicetree (that includes the additional SDIO interface and 2 GPIOs)

If changes need to be made to the Device Tree, the .dts source file that was downloaded to the `~/Documents/linux-xlnx` folder may be edited, then re-compiled by executing the following from within this folder:

```
./scripts/dtc/dtc -I dts -O dtb microZed-devicetree.dts -o devicetree.dtb
```

5.3 Configure and Build the Linux Kernel

The `uz_wl8_kernel.sh` script also downloads a preconfigured .config file that is used to configure and build a Kernel that includes WiLink™ 8 WLAN support.

The key Kernel configuration changes (for Wi-Fi) are shown below...

<table>
<thead>
<tr>
<th>Networking support</th>
<th>(CONFIG_NET [=y])</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wireless</td>
<td>(CONFIG_WIRELESS [=y])</td>
</tr>
<tr>
<td>cfg80211 - wireless configuration API</td>
<td></td>
</tr>
<tr>
<td>Generic IEEE 802.11 Networking Stack (mac80211)</td>
<td>(CONFIG_FW_LOADER)</td>
</tr>
<tr>
<td>Include in-kernel firmware blobs in kernel binary</td>
<td>(CONFIG_FIRMWARE_IN_KERNEL [=n])</td>
</tr>
<tr>
<td>Network device support</td>
<td>(CONFIG_NETDEVICES [=y])</td>
</tr>
<tr>
<td>Wireless LAN</td>
<td>(CONFIG_WLAN [=y])</td>
</tr>
</tbody>
</table>

5.4 Startup Script for auto-connect to a specific WLAN Access Point

Shown below is an example of untethered operation where the Lighttp webserver is accessed from an Internet browser running on a tablet or SmartPhone which in this case also provides a Mobile Hotspot (Wireless Access Point).

Connection to this hotspot is taken care of using a `startup.sh` script file (which sequences the network configuration and connection commands), ensuring automatic connection to it's local WLAN network on power-up.

MicroZed’s USR LED signals when connectivity has been established and MicroZed is now ready to serve-up the webpages from it's microSD card.
This example test configuration (where an IO Carrier Card is not used) is as follows:

A. Avnet WiLink™ 8 Pmod Adaptor Board (only the WLAN connector is attached)
B. Insulator between boards (a folded Business Card)
C. Texas Instruments WL1835MODCOM8B wireless evaluation board
D. Avnet MicroZed SOM board
E. Lithium Battery Pack
F. Internet Browser (on an Android tablet)

Placing your own custom `startup.sh` script file, in the `~/Documents/wl8_scripts` folder allows the user to specify what exactly gets automatically executed on startup.

When building the ramdisk root filesystem the `uz_wl8_ramdisk.sh` script file will automatically include this file plus the auto-start mechanism in the new `ramdisk.image.gz` file
The startup script file used in this example is shown below:

```bash
sleep 2
echo " "
echo "***********************************************************************
*** WiLink™ 8: Enabling WLAN interface..."
ifconfig wlan0 up
sleep 5
echo " "
echo "*** WiLink™ 8: Scan for local SSIDs..." iw wlan0 scan | grep SSID
sleep 5
echo " "
echo "*** WiLink™ 8: Connect to Wireless A/P (X-Fest2014-IoT)..." iw wlan0 connect X-Fest2014-IoT
sleep 5
echo " "
echo "*** WiLink™ 8: Verify connection to Wireless A/P..." iw wlan0 link
sleep 5
echo " "
sleep 5
```

5.5 Lighttp Webserver and the adding of Custom Web Pages

A lighttp webserver is included in the scripted Linux build. When building the ramdisk file system, the `uz_wl8_ramdisk.sh` script file will automatically include whatever webpage files are located in the `~/Documents/wl8_webserver_demo` folder when building the `ramdisk.image.gz` file
5.6 Hardware Setup

19. A terminal program is required (Windows 7 does not come pre-installed with a terminal program.) Tera Term was used in this example and can be downloaded from the SourceForge Japan page: [ttssh2.sourceforge.jp](http://ttssh2.sourceforge.jp).

20. If not previously installed, go to [www.microzed.org](http://www.microzed.org) to download and install the Silicon Labs CP2104 USB-to-UART driver, [www.microzed.org/support/documentation/1519](http://www.microzed.org/support/documentation/1519).


22. Attach Texas Instruments WL1835MODCOM8B wireless eval board to J2 of Avnet WiLink™ 8 Wi-Fi Adaptor.

23. Attach WLAN Pmod connector (J3) of Adaptor Board to MicroZed’s PS-Pmod connector (J5). The adaptor should be oriented so that the Green WiLink™ 8 board is nearest MicroZed’s RJ45 ethernet connector. NB: Insulate protruding jumpers on WiLink board from the MicroZed board (eg. by means of a business card).

24. Connect a microUSB cable from computer to USB-UART port of MicroZed (J2) (MicroZed will power on, the Green Power Good LED (D5) should illuminate).

25. Launch a Serial console (115200 baud) on the host and watch MicroZed’s Linux boot messages for warnings or errors.

26. Login as root at the microzed login command-line prompt.

27. At the `root@microzed:~` command prompt, enter the commands shown in **bold text** below (the **yellow** highlighted text indicates typical response where this is relevant to clarifying the procedure):

```
# Enable WiLink™ 8’s WLAN radio:
ifconfig wlan0 up

wlcore: PHY firmware version: Rev 8.2.0.0.212
wlcore: MAC firmware booted (Rev 8.9.0.0.17)

# Scan for SSIDs of Wireless Access Points:
iw wlan0 scan | grep SSID
SSID: X-Fest2014-IoT

# Connect to the desired Wireless A/P (use SSID reported from previous step)
iw wlan0 connect X-Fest2014-IoT

# Verify connection to Wireless A/P
iw wlan0 link

# Request IP address (DHCP) from Wireless A/P
udhcpc -i wlan0
Lease of 192.168.1.101 obtained

# Ping this IP address from host computer (or other device on same subnet), eg.
ping 192.168.1.101
A response such as the following should be observed:
64 bytes from 192.168.1.101: seq=3 ttl=64 time=0.248 ms
```
28. Once connectivity with WiLink has been verified, paste this same IP address into the address field of an internet browser (on host computer, tablet or smartphone provided they are connected to same Wireless A/P). This will display webpage output from the lighttpd webservice app that is executing on MicroZed.

29. Additional Linux-based tests for different use-cases are described at:
   http://processors.wiki.ti.com/index.php/WiLink8_WLAN_Demo:_Linux

   It is recommended that you explore Texas Instrument’s WiLink™ 8 Wiki pages for further insights and detail on getting the most utility out of your WiLink™ 8 Wireless board:

   Especially useful are the following sub-pages:
   http://processors.wiki.ti.com/index.php/WiLink8_WLAN_Demo:_Linux
6  Getting Help and Support

6.1  Avnet Support
MicroZed is a versatile development kit as well as a SOM ready to be adopted into your next design. All technical support is offered through the MicroZed.org support forums. Users are encouraged to participate in forums and offer help to others where possible.

For questions regarding the MicroZed community website, please direct questions to: webmaster@microzed.org

To access the most current collateral for MicroZed, visit the community support page at: http://www.MicroZed.org/content/support

Once on the MicroZed.org support page:

To access latest MicroZed documentation, click on the Documentation link:

To access latest reference designs for MicroZed, click on the following link:

To access the MicroZed technical forums, click on the following link:

6.2  Xilinx Support
For questions regarding products within the Product Entitlement Account, send an e-mail message to the Customer Service Representative in your region:

Canada, USA and South America - isscs_cases@xilinx.com

Europe, Middle East, and Africa - eucases@xilinx.com

Asia Pacific including Japan - apaccase@xilinx.com

For technical support including the installation and use of the product license file, contact Xilinx Online Technical Support at www.xilinx.com/support. The following assistance resources are also available on the website:

– Software, IP and documentation updates
– Access to technical support web tools
– Searchable answer database with over 4,000 solutions
– User forum
7 Appendix A: Installing the ArchiTech Yocto SDK

1. Open the “QuickStart” page of the online ArchiTech Wiki pages for more detailed guidance than the quick checklist provided below:

2. Download and Install VirtualBox (version 4.2.10 or higher)
   https://www.virtualbox.org/wiki/Downloads

3. Download the latest VM image for the ArchiTech Yocto SDK
   http://downloads.architechboards.com/sdk/virtual_machine/download.html
   Note: A Download Manager must be used for download of this large file (5.7 GB) eg. free version of Speedbit’s Download Accelerator App (for Windows)

4. Create a Virtual Machine as follows
   a. Name = “architech” (do not use a different name)
   b. Type = Linux
   c. Version = Ubuntu
   d. Memory = 2048 MB
   e. Use an existing virtual hard drive (select the downloaded architech .vdi file)

5. Click on Settings, then
   a. System/Processor: select number of CPUs available
   b. Network/Advanced/Port Forwarding: add a port forwarding rule
   c. Shared Folder: add a shared folder
   d. Click ‘OK’ to save settings

6. Once the VM is running, Install Guest Additions (when prompted or from Devices menu) which mounts the applicable VBOXADDITIONS .ISO file as a virtual drive (this adds additional Linux device drivers and useful Apps)
   a. The password is ‘architech’

Note! The online ArchiTech Wiki pages detail the installation and use of the SDK
The ArchiTech Yocto SDK installation on the host computer has the following folder structure (only the relevant sections and the microzed branch are shown):

```
Home
  |__ Documents
  |   |__ build-utilities
  |   |   |__ linux-xl
  |   |__ output
  |   |__ wl8_scripts
  |   |   |__ wl8_webserver_demo
  |   |__ u-boot-xl
  |   |   |__ architech_sdk
  |   |   |   |__ architech
  |   |   |   |   |__ microzed
  |   |   |   |   |   |__ conf
  |   |   |   |   |   |   |__ docs
  |   |   |   |   |   |   |__ eclipse
  |   |   |   |   |   |__ java
  |   |   |   |   |__ qtcreator
  |   |   |   |__ splashscreen
  |   |   |   |__ sysroot
  |   |   |__ toolchain
  |   |__ workspace
  |   |   |__ eclipse
  |   |   |__ qt
  |   |   |__ yocto
  |   |   |   |__ build
  |   |   |   |__ meta-microzed
  |   |   |   |__ meta-openembedded
  |   |   |   |__ meta-xilinx
  |   |   |__ poky
  |   |   |__ splashscreen
  |   |   |   |__ splashscreen-interface
  |   |   |__ architech-manifest
```
Appendix B: Building the Hardware Platform using Vivado

The WLAN host interface needs to be added to the Zynq H/W platform definition.

This consists of:

SDIO (SD1), which provides a 4bit wide data transport interface

WL_IRQ input, is the Out of Band hardware Interrupt from WLAN subsystem WL_EN output, controls the WLAN subsystems power

![Figure 5 – Adding the WLAN host interface in Vivado](image)

<table>
<thead>
<tr>
<th>Pmod Pin#</th>
<th>PS MIO Pin</th>
<th>MicroZed Zynq pin#</th>
<th>WiLink8 Pmod Signal Name</th>
<th>Pmod Pin#</th>
<th>PS MIO Pin</th>
<th>MicroZed Zynq pin#</th>
<th>WiLink8 Pmod Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MIO 13</td>
<td>E8</td>
<td>SDIO_D1</td>
<td>7</td>
<td>MIO 0</td>
<td>E6</td>
<td>WL_IRQ</td>
</tr>
<tr>
<td>2</td>
<td>MIO 10</td>
<td>E9</td>
<td>SDIO_D0</td>
<td>8</td>
<td>MIO 9</td>
<td>B5</td>
<td>WLAN_EN_SOC</td>
</tr>
<tr>
<td>3</td>
<td>MIO 11</td>
<td>C6</td>
<td>SDIO_CMD</td>
<td>9</td>
<td>MIO 14</td>
<td>C5</td>
<td>SDIO_D2</td>
</tr>
<tr>
<td>4</td>
<td>MIO 12</td>
<td>D9</td>
<td>SDIO_CLK</td>
<td>10</td>
<td>MIO 15</td>
<td>C8</td>
<td>SDIO_D3</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>GND</td>
<td></td>
<td>11</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>VCC</td>
<td></td>
<td>12</td>
<td></td>
<td>VCC</td>
<td></td>
</tr>
</tbody>
</table>

*Predefined PS MIO peripheral function for SDIO

![Figure 6 – PS MIO pinout for the WLAN interface](image)
Note: Vivado 2014.2 project files targeting MicroZed 7010 and 7020 boards are available at www.microzed.org/product/WiLink™-adapter (in the Reference Design / Tutorials section)

8.1 IP Integrator: Adding the WLAN interface (SD1 and MIO GPIO)

Figure 7 – Adding the WLAN interface in Vivado IP Integrator
8.2 IP Integrator: Adding Bluetooth (UART0 and EMIO GPIO)

Figure 8 – Adding the Bluetooth interface in Vivado IP Integrator

Figure 9 – IP Integrator Block Diagram showing Bluetooth I/O signals
8.3 Vivado: Implementation, Bitstream and Export to SDK
- Run Implementation
- Generate Bitstream
- Export Hardware Design to SDK
  (File → Export → Export Hardware [Include Bitstream])

8.4 SDK: Generate an updated Zynq Boot Image
Because the bitstream has changed, a new BOOT.bin file needs to be generated. Launch the Create Zynq Boot Image tool in Vivado’s SDK to create this file Select Import from existing BIF file and use the provided BIF file
  (located in the ..\SD_image\ramdisk_version folder)

Once completed, copy the new BOOT.bin file to the ~/Documents/output staging area on your Ubuntu Linux host system for writing to the SD card later.
Appendix C: Wi-Fi Quick Start instructions (MicroZed only, pre-built Linux files)

1. Prepare a FAT32 formatted microSD card (4GB+) with first partition labeled BOOT
2. Mount the microSD card (via an SD card-reader) onto your Linux host computer
3. Download the latest SD card creation script file from [http://microzed.org/pmodwifi-wilink8](http://microzed.org/pmodwifi-wilink8)
4. In the home folder of your Linux host system, create a temporary folder to which this script file should be copied.
5. From this folder, launch the SD card script (host must be connected to Internet!) with the following command: `/microzed-wilink8-ramdisk-SDcard.sh 7010`
   (this script downloads 6 files and copies them to BOOT partition of microSD card)
6. Safely unmount the microSD card from the Linux host computer
7. Insert the microSD card into MicroZed’s microSD connector
8. Set the boot mode (JP3-JP1) jumpers on MicroZed for SD card mode
9. Attach Texas Instruments WL1835MODCOM8B wireless eval board to J2 of the Avnet WiLink™ 8 Wi-Fi Adaptor
10. Attach WLAN Pmod connector (J3) of Adaptor Board to MicroZed’s PS-Pmod connector (J5),
    *Note! The adaptor should be oriented so that the Green WiLink™ 8 board is nearest MicroZed’s RJ45 ethernet connector (Insulate the protruding jumpers on Wilink board from the MicroZed board, e.g. by means of a business card)*
11. Connect a microUSB cable from computer to USB-UART port of MicroZed (J2) (MicroZed will power on, the Green Power Good LED (D5) should illuminate)
12. Launch a Serial console (115200 baud) on the host and observe MicroZed’s Linux boot messages
13. Login as root at the microzed login command-line prompt
14. At the `root@microzed:`~# command prompt, enter the commands shown in bold text below (the yellow highlighted text indicates typical response where this is relevant to clarifying the procedure)

```bash
# Enable WiLink™ 8’s WLAN radio:
ifconfig wlan0 up
    wlcore: PHY firmware version: Rev 8.2.0.0.212
    wlcore: MAC firmware booted (Rev 8.9.0.0.17)

# Scan for SSIDs of Wireless Access Points:
iw wlan0 scan | grep SSID
SSID: X-Fest2014-IoT
```
# Connect to the desired Wireless A/P
(using the SSID reported from the previous command), eg.
iw wlan0 connect X-Fest2014-IoT

# Verify connection to Wireless A/P
iw wlan0 link

# Request IP address (DHCP) from Wireless A/P
udhcpc -i wlan0
Lease of 192.168.1.101 obtained

# Ping this IP address from Linux host computer command prompt (or another device on same subnet), eg.
ping 192.168.1.101
A response such as the following should be observed:
64 bytes from 192.168.1.101: seq=3 ttl=64 time=0.248 ms

15. Once connectivity with Wilink has been verified, paste this same IP address into the address field of an internet browser (on host computer, tablet or smartphone - provided they are connected to same Wireless A/P). This will display webpage output from the lighttp webserver app that is executing on MicroZed

16. Additional Linux-based tests for different use-cases are described at:
http://processors.wiki.ti.com/index.php/WiLink8_WLAN_Demo:_Linux
10 Appendix D: Bluetooth Quick Start instructions

Use the following commands to enable Bluetooth and initiate a scan (a basic verification of BlueZ stack and HCI functionality)

# Enable the Bluetooth radio by setting the BT_EN pin high
echo 54 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio54/direction
echo 1 > /sys/class/gpio/gpio54/value

# Bluetooth modules (drivers) do not require loading, - they are already built into the kernel

# activate BlueZ
hciattach -t 30 -s 115200 /dev/ttyPS1 texas 3000000 flow hciconfig

# scan for bluetooth devices
hcitool -i hci0 scan

# get device info (change the address shown to that of the device you wish to test with)
hcitool -i hci0 info 20:13:E0:BC:36:95

# check communication with device
l2ping 20:13:E0:BC:36:95

# scan for BLE devices
hcitool lescan

<table>
<thead>
<tr>
<th>Pmod Pin#</th>
<th>PL pin Name</th>
<th>MicroZed Zynq pin#</th>
<th>WiLink8 Pmod Signal Name</th>
<th>Pmod Pin#</th>
<th>PL pin Name</th>
<th>MicroZed Zynq pin#</th>
<th>WiLink8 Pmod Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EMIO</td>
<td>Y18</td>
<td>BT_HCI_CTS</td>
<td>7</td>
<td>EMIO</td>
<td>N17</td>
<td>BT_EN</td>
</tr>
<tr>
<td>2</td>
<td>EMIO</td>
<td>Y19</td>
<td>BT_HCI_TX</td>
<td>8</td>
<td>EMIO</td>
<td>P18</td>
<td>BT_WAKE_HOST</td>
</tr>
<tr>
<td>3</td>
<td>EMIO</td>
<td>Y16</td>
<td>BT_HCI_RX</td>
<td>9</td>
<td>EMIO</td>
<td>P15</td>
<td>WL_RS232_TX</td>
</tr>
<tr>
<td>4</td>
<td>EMIO</td>
<td>W16</td>
<td>BT_HCI_RTS</td>
<td>10</td>
<td>EMIO</td>
<td>P16</td>
<td>WL_RS232_RX</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>GND</td>
<td></td>
<td>11</td>
<td></td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>VCC</td>
<td></td>
<td>12</td>
<td></td>
<td>VCC</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10 – PL EMIO pinout for Bluetooth interface
11 Appendix E: Demo List on WiLink™ 8 Wiki Pages

11.1 WiLink™ 8 Wiki: Landing Page

11.2 Wi-Fi Use Cases
WiLink8 WLAN Demos: Linux

Connect to Secure AP using WPA Supplicant
WiFi Direct-P2P
mBSSID
Portable Wireless Gateway
Wireless Hard Drive with fast access to content
WiFi Provisioning for Headless Devices
Configurable Low Power WiFi
Internet Video over WiFi with Audio over BT A2DP
Appendix F: App Development in ArchiTech’s Yocto SDK

Application code development and debug for Avnet boards or 3rd party custom boards may be done from the Eclipse IDE and QT Creator tools that are provided within the ArchiTech Yocto SDK installation. More detail on this topic is available at:

http://architechboards-microzed.readthedocs.org/en/latest/quick.html#code