



MiWi™ v6.1 Quick Start Guide

Introduction

This document is intended as a starting point for software engineers prototyping, implementing, testing, and deploying wireless sensor networks using MiWi™ Protocol based on the Microchip software platform.

This document describes how to start quickly with the MiWi™ by installing development environment, setup hardware and program devices with reference applications.

Chapter 1 provides an overview of the MiWi™ v6.1 package, lists supported platforms.

Chapter 2 describes the documentation set available for MiWi™ v6.1.

Chapter 3 and Chapter 4 gives instructions on Development tools setup and Hardware environment setup.

Chapter 5 and 6 describes the MiWi™ P2P Protocol applications – Simple_Example_P2P, Chat Demo.

Chapter 7 describes the MiWi™ Star Protocol application – Simple_Example_Star.

Chapter 0 describes the MiWi™ Mesh Protocol application - WSNDemo

Table of Contents

1.	OVERVIEW	4
1.1	MiWi™ v6.1 Release Contents.....	4
1.2	Supported hardware platforms and IDEs.....	5
2.	MIWI™ DOCUMENTATION.....	5
3.	DEVELOPMENT ENVIRONMENT SETUP	6
3.1	IDE installation	6
3.1.1	Atmel Studio	6
3.1.2	IAR Embedded Workbench	6
3.2	Stack Configurations	6
3.3	Building applications in Atmel Studio.....	7
3.3.1	Opening Project from ASF	7
3.3.2	Switching/Changing the configuration using ASF Wizard	10
3.3.3	Building application from IDE.....	11
3.3.4	Building application from command line	12
3.4	Building applications in IAR Embedded Workbench.....	12
3.4.1	IDE build procedure	13
3.4.2	Command line build procedure.....	13
4.	HARDWARE ENVIRONMENT SETUP	13
	Supported platform and boards	13
5.	SIMPLE_EXAMPLE_P2P REFERENCE APPLICATION	17
6.	CHAT_DEMO P2P REFERENCE APPLICATION	20
7.	SIMPLE_EXAMPLE_STAR REFERENCE APPLICATION	23
8.	WSN DEMO MESH REFERENCE APPLICATION	27
9.	OTAU IN WSN DEMO MESH APPLICATION.....	32
9.1	Required Software.....	32
9.2	OTAU Client	32
9.2.1	Memory Layout on Client	32
9.2.2	Steps to Program Client Firmware	32
9.3	OTAU Server.....	34
1.1	Atmel WiDBG for OTAU	35
1.2	Starting the Session	36
1.2.1	Notification Settings	37
1.3	Main Window.....	39
1.3.1	Table View.....	40
1.3.2	Saving the Table View.....	41
1.3.3	Searching the Nodes	41
1.3.4	Network Topology View	43

1.3.5	Output View	45
1.4	Upgrading the Node	46

1. Overview

MiWi™ Development Environment (MiWi™ DE) is developed by Microchip to support a wide range of wireless applications. The backbone of MiWi™ DE is MiMAC and MiApp interfaces, which link the support of multiple RF transceivers as well as wireless communication protocols together as a well-defined simple but robust Microchip proprietary wireless development environment.

Within MiWi™ DE, application developers can switch between RF transceivers and wireless protocols with little or no modification in the application layer. By providing such easy migration capability in MiWi™ DE, as well as simple but robust interfaces, the firmware development risk has been reduced to a level that has never been observed in the industry before. MiWi™ DE is defined in three layers: application layer, protocol layer and RF transceiver layer. The three layers are linked together by MiMAC and MiApp interfaces. Application layer uses MiApp interfaces to talk to the protocol layer. In protocol layer, there are implementations of MiWi™ P2P, MiWi™Star and MiWi™ Mesh wireless communication protocols available. The drivers for Microchip RF transceivers (AT86RF233, AT86RF212B for this release) are called by protocol layers via MiMAC interfaces. Configuration files are also presented in each layer.



The MiWi™ stack offers a significantly smaller foot-print relative to the open standard based ZigBee® compliant protocol stack. This enables operation in microcontrollers with smaller memory / lower cost.

- MiWi™ P2P/Star – Simple Peer-to-Peer or Star network requiring minimal code size.
- MiWi™ Mesh – A True Routing Mesh network topology. This is perfect for a department store HVAC or alarm application or a large smart home application.

1.1 MiWi™ v6.1 Release Contents

MiWi™ is released as part of ASF. The main items provided as part of MiWi™ Release are:

- Implementation of MiWi™ P2P and Star protocol in form of sources and API header files. The same source is used for all MiWi™ P2P and Star applications.
- Implementation of MiWi™ Mesh protocol in form of libraries and API header files. The same library is used for all MiWi™ Mesh applications.

Source code and IDE projects for reference applications:

- Simple Example P2P – MiWi™ P2P reference application
- Simple Example Star – MiWi™ Star reference application
- WSN Demo – MiWi™ Mesh reference application

Documentation files

- Quick Start Guide

- Migration Guide
- Release Notes

1.2 Supported hardware platforms and IDEs

Microcontroller	RF Transceiver	Supported Evaluation Kit	Supported IDEs
SAMR21G18A(SIP)	RF233(in SIP)	SAMR21 ZLLEK SAMR21 XPRO	Atmel Studio v7.0 IAR Embedded Workbench® for ARM 7.4
SAMR30G18A(SIP)	RF212B (in SIP)	SAMR30 XPRO SAMR30M XPRO	Atmel Studio v7.0 IAR Embedded Workbench for ARM 7.4

2. MiWi™ documentation

This Chapter describes the documentation set available for MiWi™. It is intended to help user understand where to find information required during application evaluation and development.

ASK Documentation:

- ASF Documentation <http://www.microchip.com/webdoc/asf/index.html>
- ASF Getting Started
<http://www.microchip.com/webdoc/asf/asf.GettingStarted.html>
- ASF Wizard
<http://www.microchip.com/webdoc/asf/asf.ModuleExplorerView.html>

MiWi™ Documentation/Tools in web:

- MiWi™ P2P and Star Protocol Application Note
<http://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en536181>
- [AN1284 - Microchip Wireless \(MiWi™\) Application Programming Interface - MiApp](#)
- [AN1283 - Microchip Wireless \(MiWi™\) Media Access Control Interface - MiMAC](#)
- MiWi™ Protocol Sniffer
<http://www.microchip.com/SWLibraryWeb/product.aspx?product=Wireless%20Development%20Studio>

MiWi™ Documentation in ASF package:

Title	Description
Quick Start Guide	This document. User guide to quickly start with MiWi™
Migration Guide	List the migration guidelines to use the current version of MiWi™
Release Notes	Provides information on release features and enhancements

3. Development environment setup

This chapter provides instructions on how to setup MiWi™ software package as well as supported IDEs. It also describes the structure of the MiWi™ package and includes references to hardware setup of the supported platforms.

3.1 IDE installation

3.1.1 Atmel Studio

Atmel Studio can be used to develop and debug applications for AVR- and ARM-based platforms. Atmel Studio is equipped with the GCC compiler and does not require any additional external tools to compile and debug MiWi™ applications.

Installation procedure:

- Download and install latest Atmel Studio version, if not already installed on your PC.
- Add path to the folder containing the ARMGCC compiler to the Path Windows environment variable. The compiler is located in the \Atmel\Studio\7.0\toolchain\arm\arm-gnu-toolchain\arm-none-eabi\bin directory of the Atmel Studio installation directory. This step is necessary for command line compilation (with makefiles).

3.1.2 IAR Embedded Workbench

IAR Embedded Workbench for ARM can be used to develop and debug applications on ARM-based platforms. IAR IDEs support editing of application source code, compiling source files, linking object modules with libraries and application debugging.

Installation procedure:

- Download and install IAR Embedded Workbench for ARM, if not already installed on your PC
- Add path to the folder containing the IAR RAM compiler to the Path Windows environment variable. The compiler is located in the \IAR Systems\Embedded Workbench 7.2\arm\bin directory of the IAR installation directory. This step is necessary for command line compilation (with makefiles).

3.2 Stack Configurations

MiWi™ uses configuration files to regulate the behavior of the stack.

Location of Header Files to fine tune the behavior of P2P application:

1. thirdparty\wireless\miwi\apps\simple_example_p2p\miwi_config.h
2. thirdparty\wireless\miwi\apps\simple_example_p2p\miwi_config_p2p.h

Location of Header Files to fine tune the behavior of Star application:

1. thirdparty\wireless\miwi\apps\simple_example_star\miwi_config.h
2. thirdparty\wireless\miwi\apps\simple_example_star\miwi_config_p2p.h

Location of Header Files to fine tune the behavior of Mesh application:

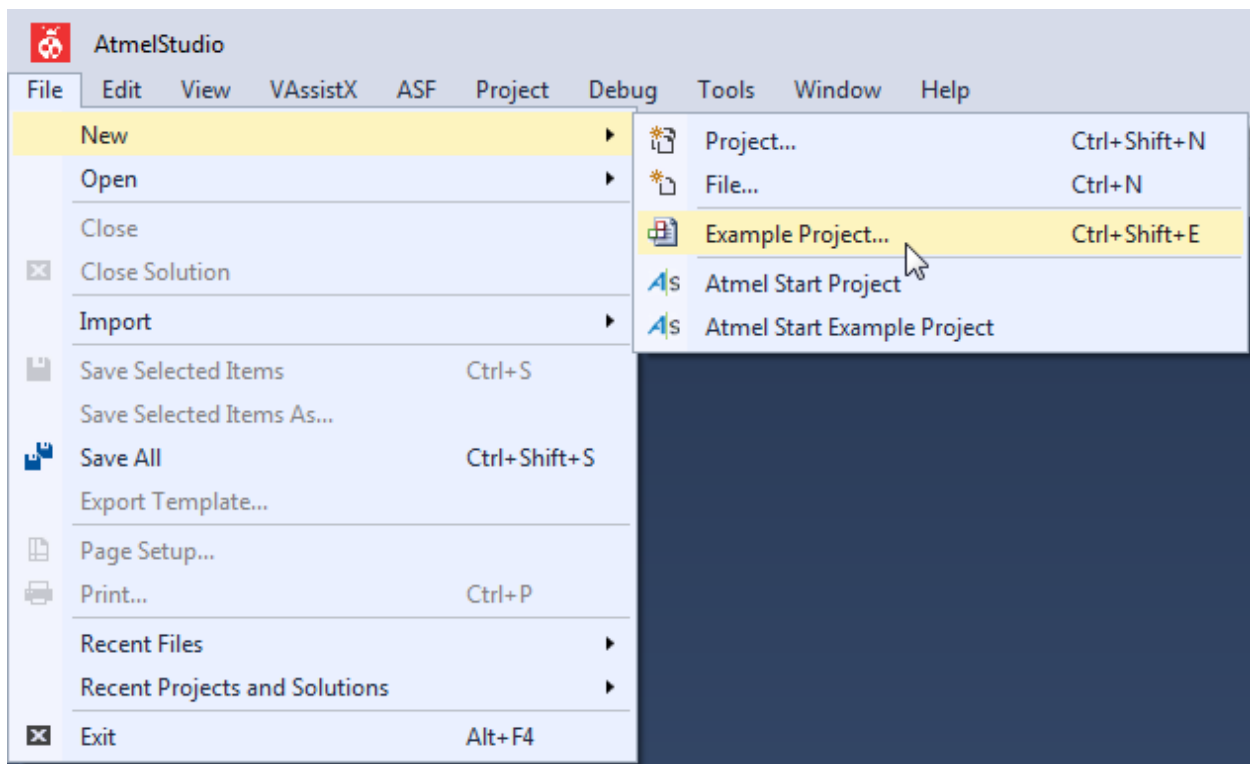
1. thirdparty\wireless\miwi\apps\wsn_demo\miwi_config.h
2. thirdparty\wireless\miwi\apps\wsn_demo\miwi_config_mesh.h

3.3 Building applications in Atmel Studio

Atmel Studio can be used to develop and build MiWi™ applications. Reference applications include Atmel Studio project files located in the \as5_arm subdirectory of the application root directory. These projects rely on the configurations given by external low-level makefiles.

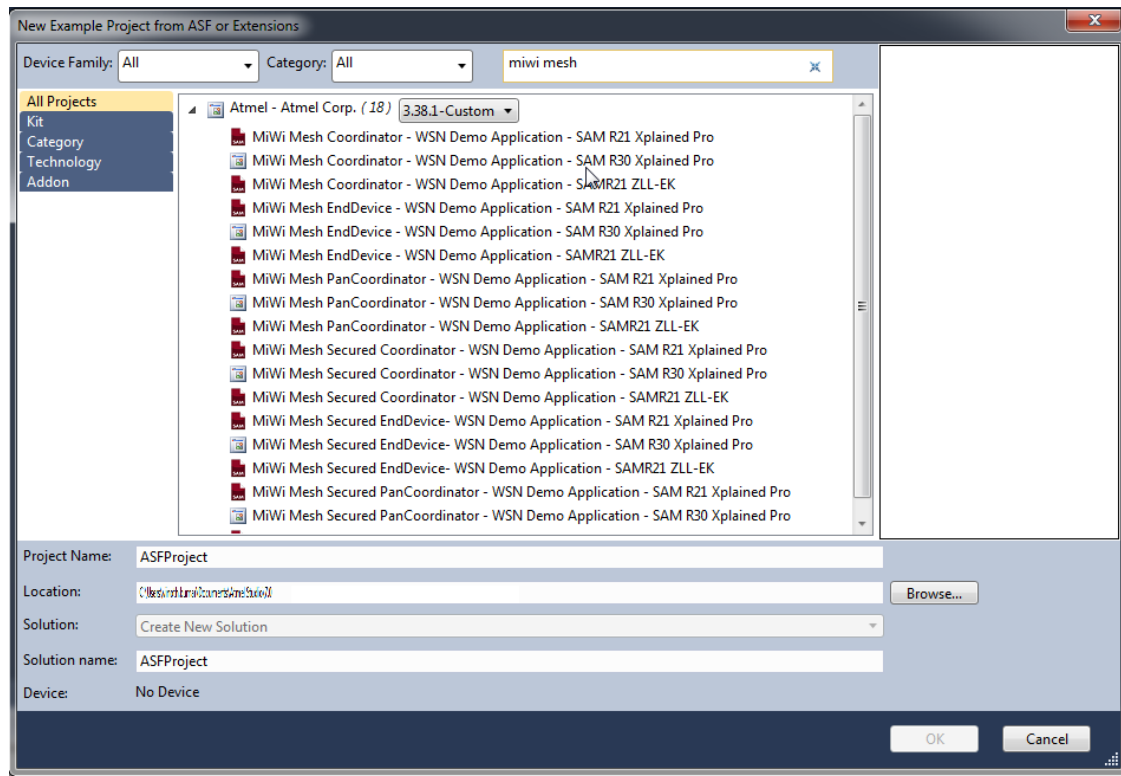
3.3.1 Opening Project from ASF

Open Atmel Studio. Click File->New->Example Project as shown below.

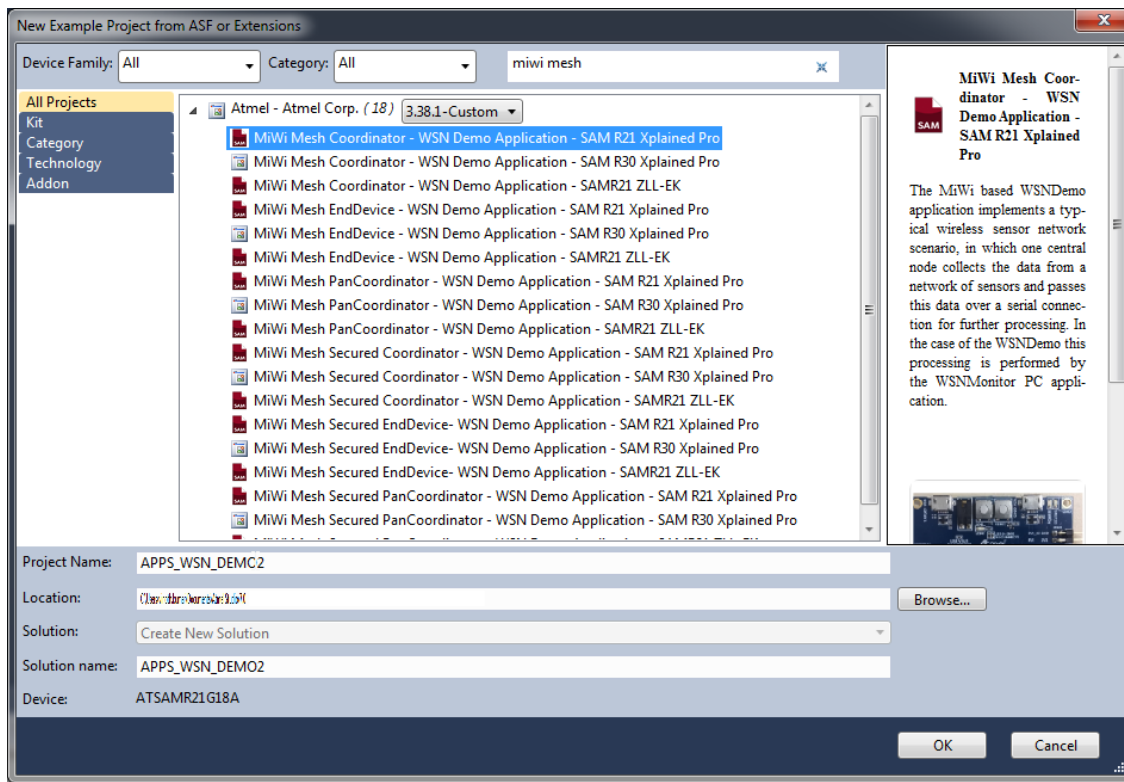


In the opened window, Type “MiWi Mesh” to list the list of available projects in MiWi Mesh as shown below

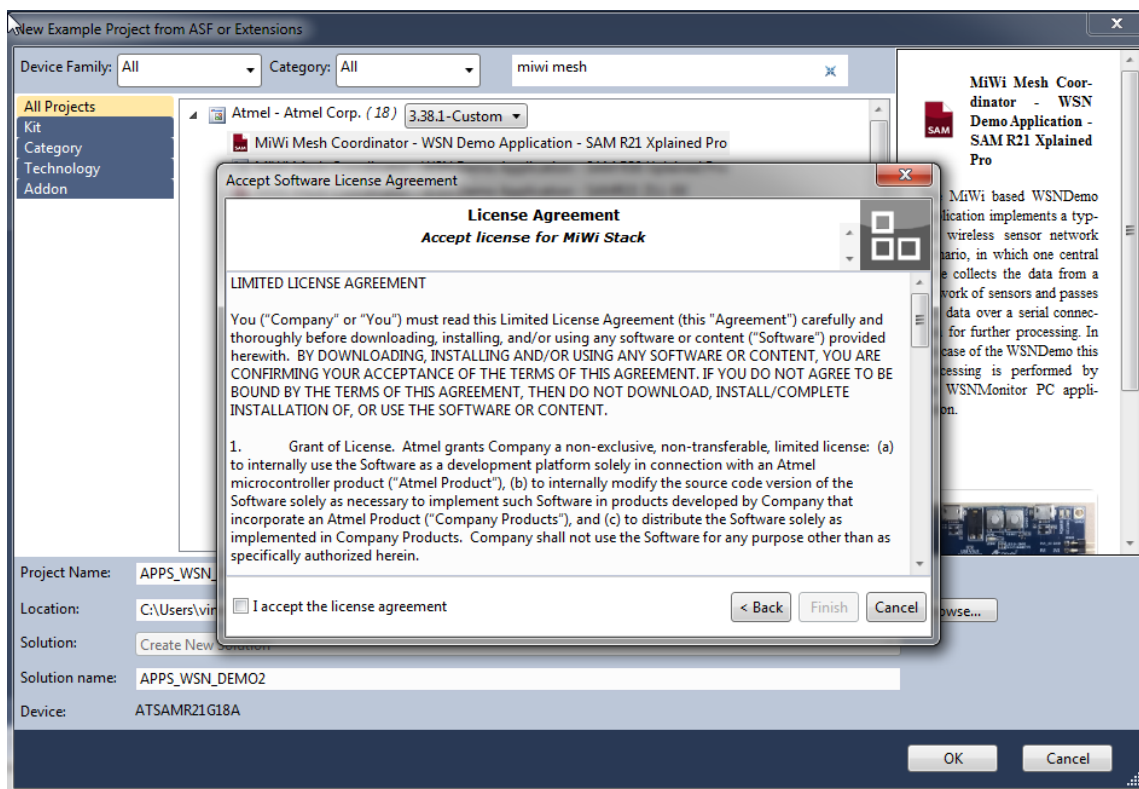
Select a project and Click Ok to open the project.



For Example, choose MiWi Mesh PAN Coordinator WSN Demo project.

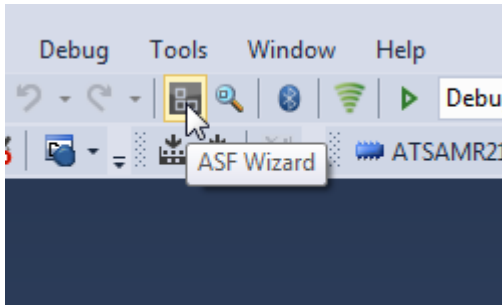


Click 'OK' and accept license agreement, click 'Finish'.

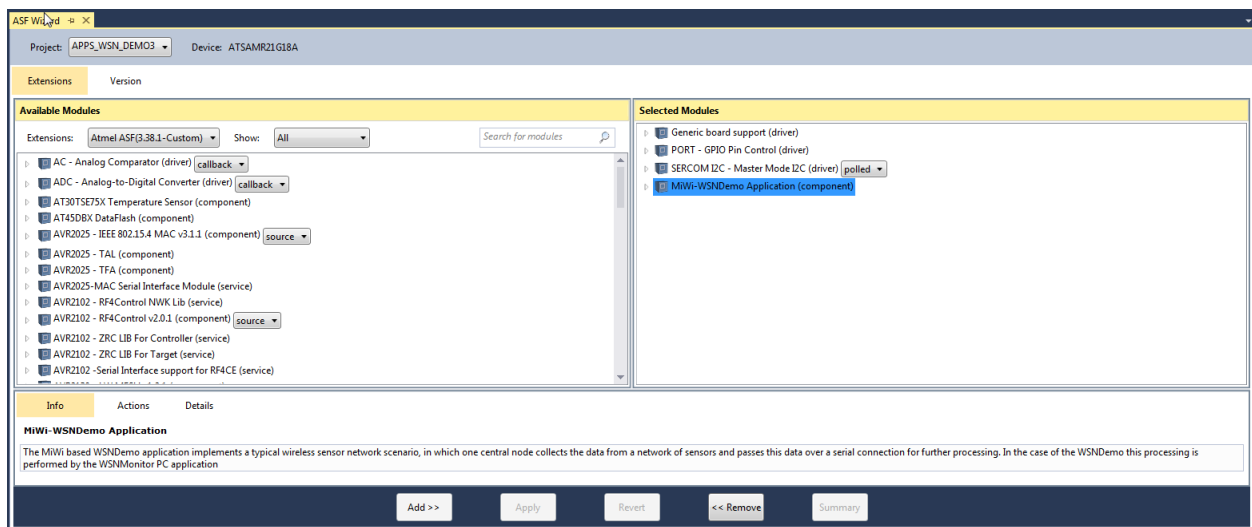


3.3.2 Switching/Changing the configuration using ASF Wizard

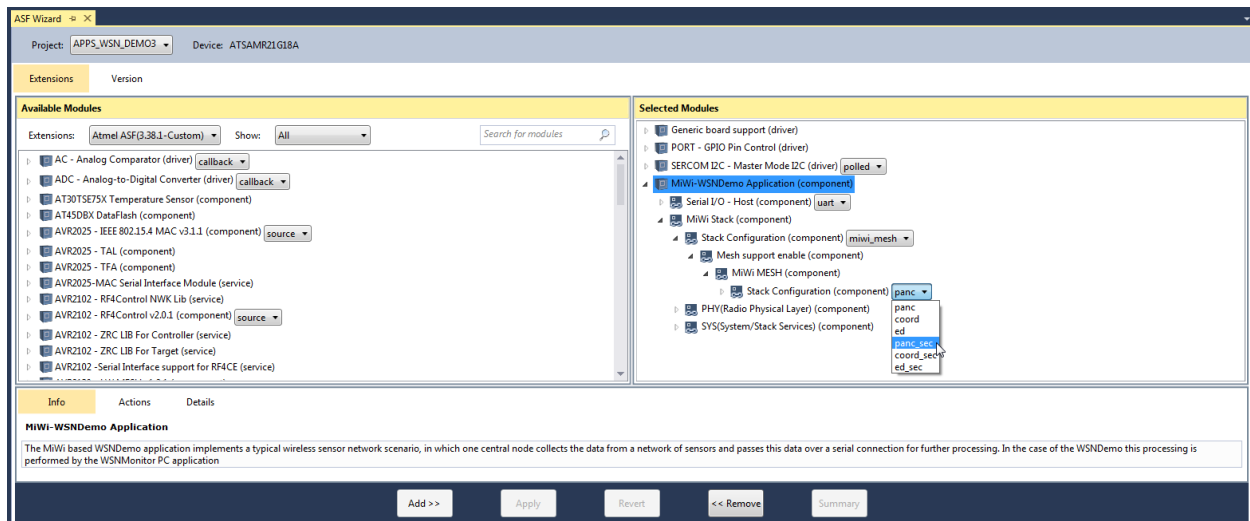
For example, currently you are using MiWi Mesh coordinator and need to switch to MiWi Mesh PANC Security, the follow the below steps. Step 1: Go to ASF Wizard



Step 2: Explore MiWi™ in the selected Modules

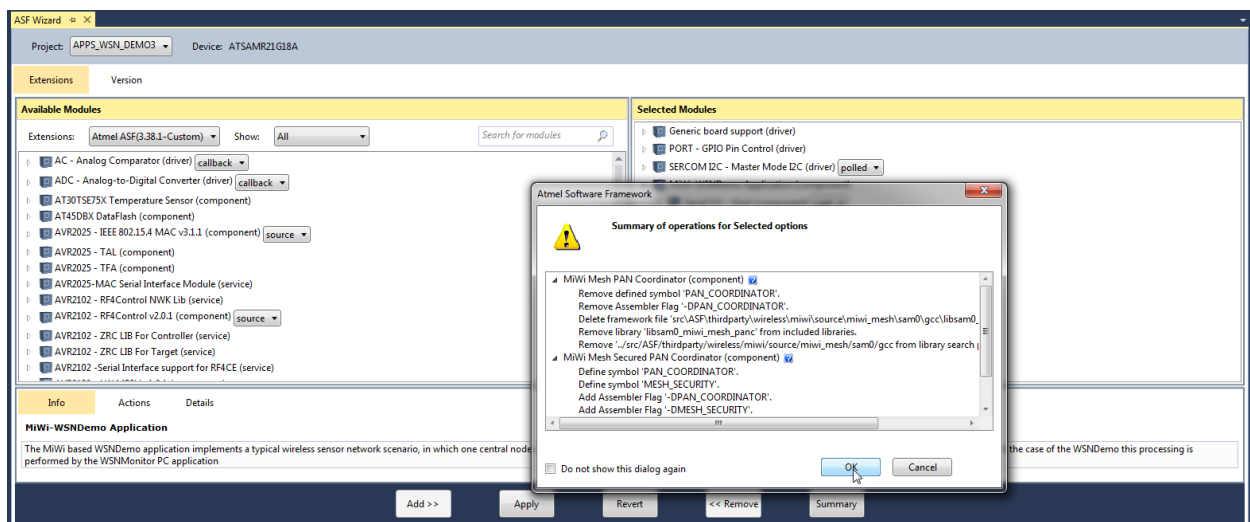


Step 3: Explore to Stack Configuration and select Panc_Sec



Step 4: Click 'Summary' to understand the components change

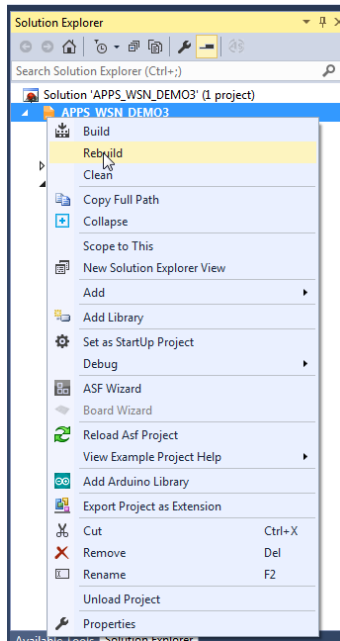
Step 5: Click 'Apply' to apply the changes.



3.3.3 Building application from IDE

Open an appropriate `.atsln` project file from the `<appName>\..\as5_arm` directory with Atmel Studio. Solution Explorer tab. This provides access to the application source files as well as stack components that compile together with application.

Example structure of Atmel Studio application project.



From the main menu execute *Build => Rebuild All*.

Once the build process is completed, some of the .hex, .srec, .bin, and .elf image files will be generated, depending on the platform configuration that has been chosen. Use the .hex file for programming devices via EDBG. The .elf file is used for debugging.

3.3.4 Building application from command line

compile the application by running the make utility, executing
make clean all

It is possible to run the make utility from Atmel Studio by selecting Tools > Command Prompt. This will guarantee that the make utility provided with Atmel Studio is used. Otherwise, the path to the folder containing the make utility can be added to the Path environment variable. In this case, run the make utility in the command line from the application's root directory.

3.4 Building applications in IAR Embedded Workbench

IAR Embedded Workbench can be used to develop and build MiWi™ applications. All reference applications include IAR project files located in the \iar subdirectory of the application root directory.

As mentioned above, a part of stack components and drivers are compiled with the application. For convenience reasons, source files for these components are included in the IAR projects, so they are effectively a part of the application.

For compilation from the command line with the IAR compiler, makefiles are used exactly the same way as described above.

3.4.1 IDE build procedure

Open the .eww file in the iar_projects subdirectory of the appropriate application directory (for WSNDemo, the APP_WSN_Demo.eww file from the apps\WSN_Demo\..\iar subdirectory) with IAR Embedded Workbench, execute the Rebuild All item from the Project menu.

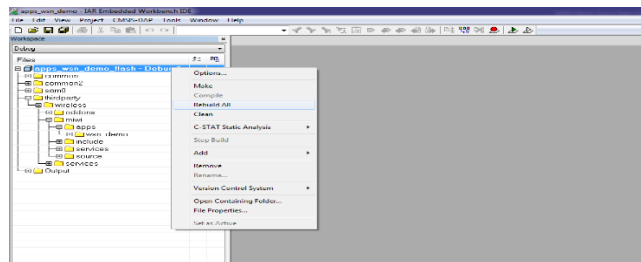
By default, the .a90 file (for WSNDemo, APP_WSN_Demo.a90) will be generated in the \iar\Debug\exe subdirectory (for WSNDemo, in the apps\WSN_Demo\iar\Debug\exe directory) with format as specified in Linker Output Options of the IAR project.

3.4.2 Command line build procedure

Compile the application by running the make utility, executing make clean all

Some of the .hex, .srec, .bin, and .elf image files will then be generated, depending on the platform configuration that has been chosen.

Project's structure in IAR Embedded Workbench.

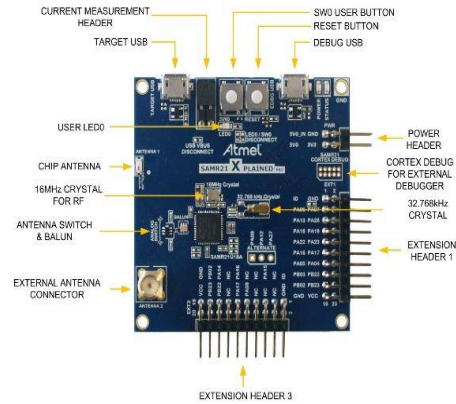


4. Hardware environment setup

Supported platform and boards

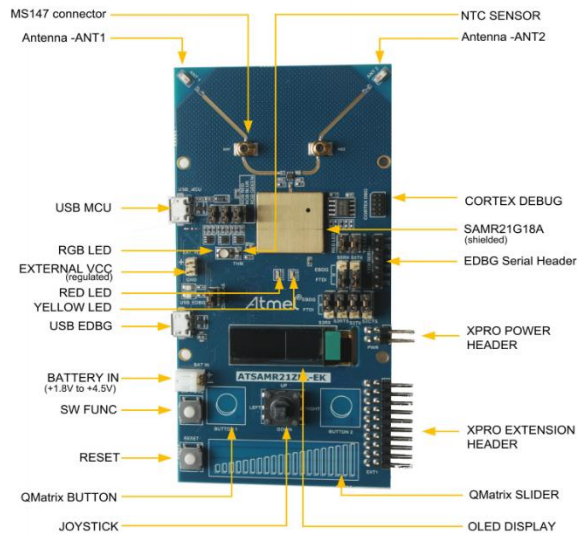
The following boards are used/supported in this release.

1. SAMR21 Xplained Pro



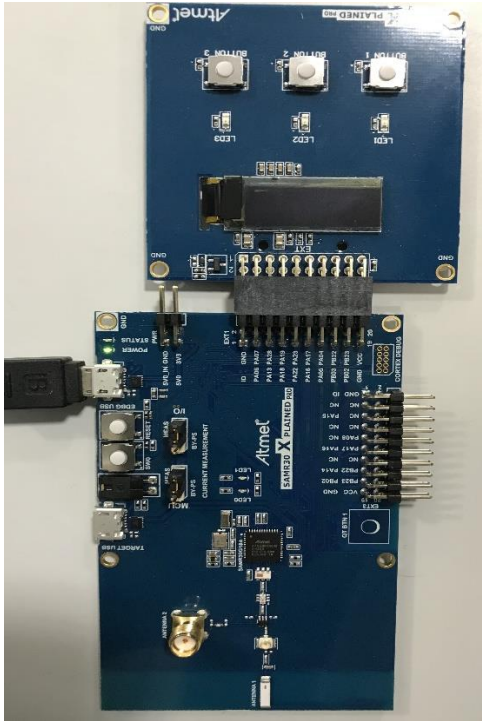
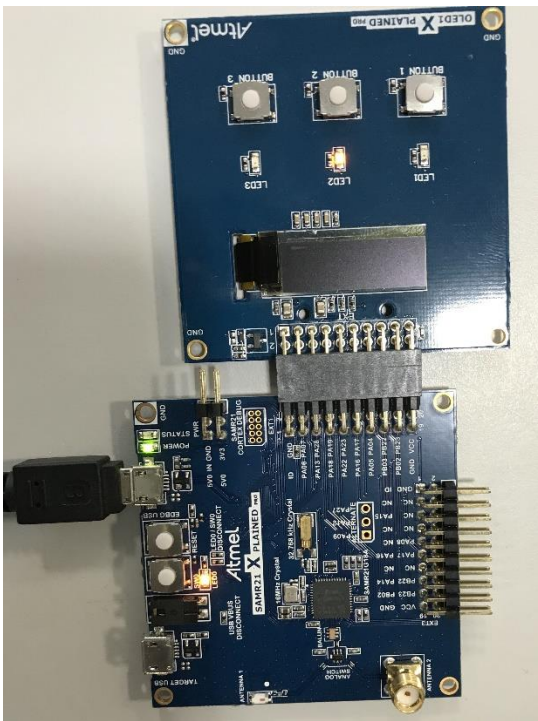
User Guide is available @ http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42243-SAMR21-Xplained-Pro_User-Guide.pdf

2. SAMR21 ZLLEK



User Guide is available @ http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42462-ATSAMR21ZLL-EK_UserGuide.pdf

3. SAMR30 Xplained Pro

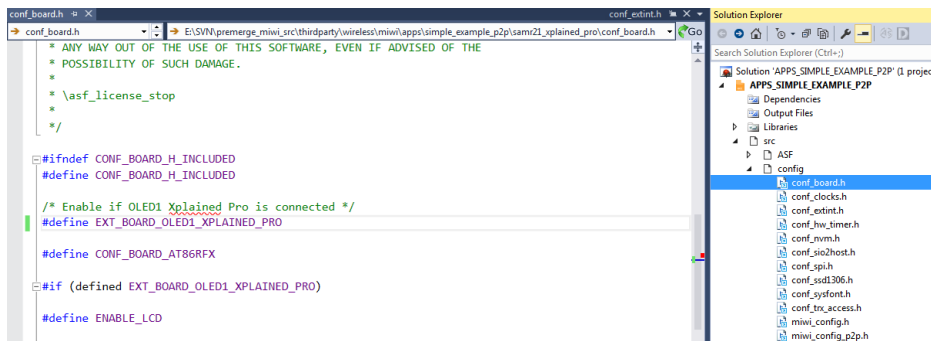


5. Simple_Example_P2P Reference application

The simple example application code focuses on the simplicity of the MiWi™ DE protocol stack application programming interfaces. It provides a clean and straightforward wireless communication between two devices with less than 30 lines of effective C code to run the stack in application layer for both devices. In this application, following features of MiWi™ DE protocol stack has been demonstrated:

- Establish connection automatically between two devices
- Broadcast a packet
- Unicast a packet
- Apply security to the transmitted packet

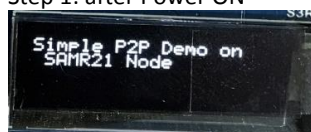
To run full featured Simple_Example_P2P application on SAMR21 XPRO or SAMR30 XPRO, User needs to connect OLED1 XPlained PRO. When you use OLED1 Xplained PRO along with SAMR21 XPRO or SAMR30 XPRO, then user needs to Enable the macro “`#define EXT_BOARD_OLED1_XPLAINED_PRO`” in `conf_board.h` file.



To run the simple example application, follow the instructions given below

1. Program node 1 and node 2 with Simple_Example_P2P application firmware.
2. Power on node 1 and node 2 respectively
3. Wait a few seconds, until the first LED on both nodes light up. This means a connection has been established automatically.
 - For the details of connection establishment, please refer to section “VARIATIONS FOR HANDSHAKING” in application note AN1204 “Microchip MiWi™ P2P Wireless Protocol” if MiWi™ P2P protocol is used, or section "MAC Function Description" in IEEE 802.15.4 specification if MiWi™ protocol is used.
 - If the demo is running on SAMR21 ZLLEK (or) SAMR21 XPRO with ATOLED1-XPRO (or) SAMR30 XPRO with ATOLED1-XPRO, critical information will be shown on the LCD of the demo board. It first shows the demo name, RF transceiver and node number, then connecting information and channel information will be shown before the LCD shows the demo instruction

Step 1: after Power ON



Step2: Connecting to a network if found

Connecting Peer on
Channel 26

Step3: Connected to a network (This will not come for first device since it will start network when it is unable to find network to join)

Connected Peer on
Channel 26

Step4: Demo Instruction will be displayed.

SAMR21ZLLEK

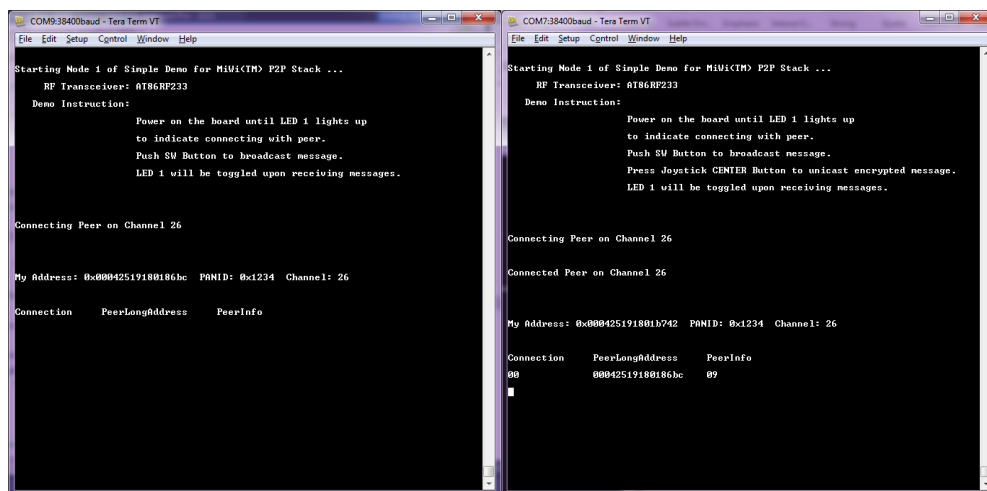
SW FUNC : Broadcast
JOYSTICKCenter: Unicast

or

SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO

SW BUTTON1: Broadcast
BUTTON1: Unicast

- If a hyper terminal has been opened to monitor firmware output, you should be able to see the information about the peer device printed out from both nodes.



4. Press SW/SW FUNC on one node will toggle the second LED on the other node
 - This shows how a broadcast packet has been transmitted.
 - If the demo is running on SAMR21 ZLLEK (or) SAMR21 XPRO with ATOLED1-XPRO (or) SAMR30 XPRO with ATOLED1-XPRO, the total number of transmitted and received messages will be shown on the LCD.

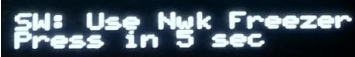
Tx Messages: 0
Rx Messages: 2

- If hyper terminal has been used, on the receiving end (the device that has LED2 toggled), you should be able to see the print out of broadcast packet source address, signal strength and the packet payload. The packet payload is the one line of bit map of "MiWi™". Press the SW button at an interval of 2 seconds between each press on one end will display the complete bit map of "MiWi™".

6. By default, Network Freezer is enabled in the application. Network Freezer feature is used to store critical network information into the non-volatile memory. When the device reboots or power is recycled, it will try to restore the persistent data from the non-volatile memory and may continue to operate in the same network with existing network information.

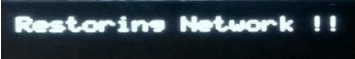
If Power is recycled on an Node, the description given below may be used to restore the persistent data if available in the node using the Network Freezer.

- After Step1, the below message will be shown on LCD. User must press SW button within 5 seconds to boot the application from stored network parameters or User can press and hold the SW button immediately after power recycle till the LCD shows the below message.



SW: Use Nwk Freezer
Press in 5 sec

- If SW button is pressed properly within the timeout, application will restore the data from Network Freezer and the below message will be shown on LCD.



Restoring Network !!

6. Chat_Demo P2P Reference application

The chat demo P2P application code focuses on the simplicity of the MiWi™ DE protocol stack application programming interfaces. It provides a clean and straightforward wireless communication between two devices with less than 30 lines of effective C code to run the stack in application layer for both devices. In this application, following features of MiWi™ DE protocol stack has been demonstrated:

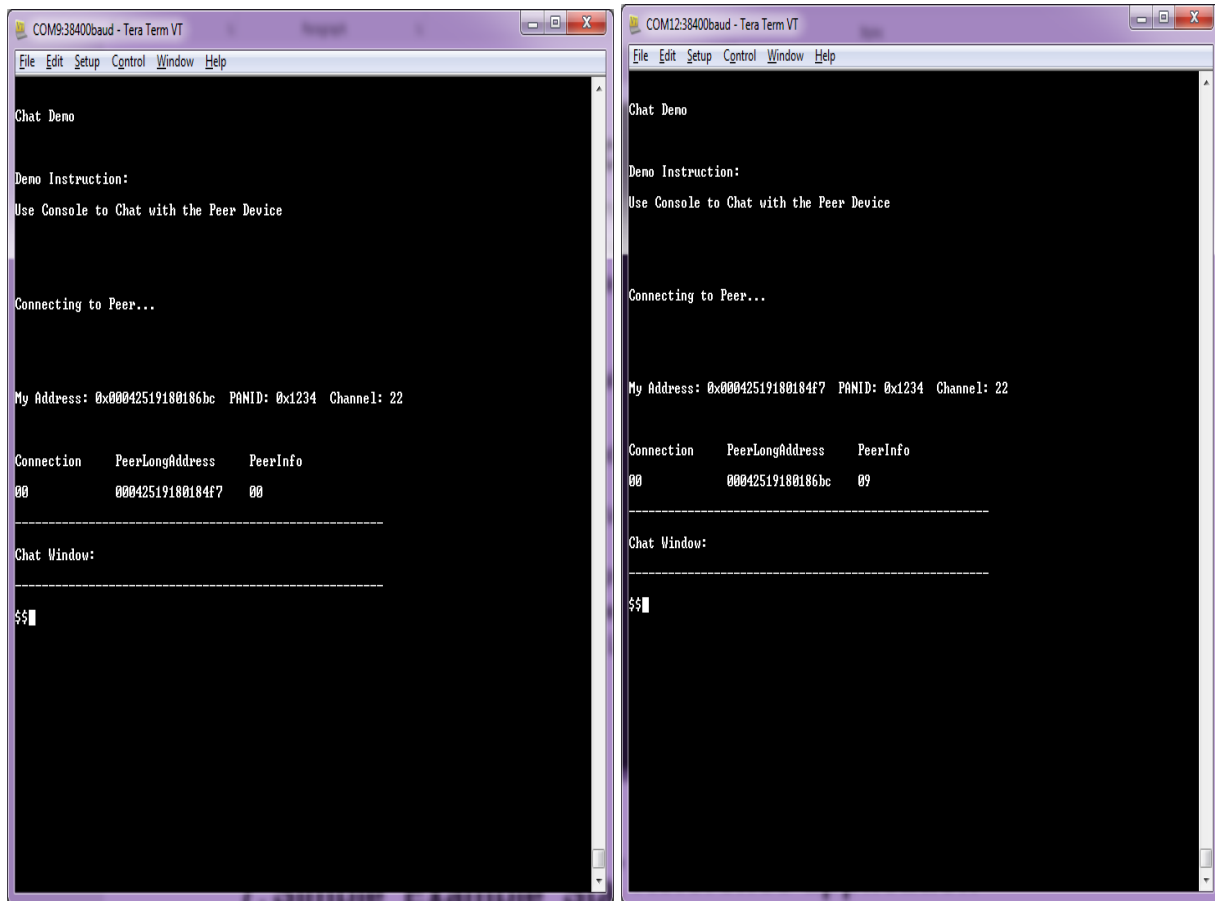
- Establish connection automatically between two devices
- Unicast a packet
- Apply security to the transmitted packet

To run the chat demo application, follow the instructions given below

1. Program node 1 and node 2 with proper firmware.
2. Power on node 1 and node 2 respectively, a hyper terminal must be opened to monitor firmware output.
3. Wait a few seconds, until the first LED on both nodes light up. These are the steps to establish connections between two devices.
4. Now you should be able to see the information about the peer device printed out from both nodes.

Node 1

Node 2



- Once the devices are connected, start typing and press enter. For example, type 'Hello World!' on node 1 and hit enter, you should be able to see the entered text on node 2 like below.

Node 1

Node 2

```
COM9:38400baud - Tera Term VT
File Edit Setup Control Window Help

Chat Demo

Demo Instruction:
Use Console to Chat with the Peer Device

Connecting to Peer...

My Address: 0x00042519180186bc PANID: 0x1234 Channel: 22

Connection PeerLongAddress PeerInfo
00 00042519180184f7 00
-----
Chat Window:
-----
$$Hello World!
$$
```

```
COM12:38400baud - Tera Term VT
File Edit Setup Control Window Help

Chat Demo

Demo Instruction:
Use Console to Chat with the Peer Device

Connecting to Peer...

My Address: 0x00042519180184f7 PANID: 0x1234 Channel: 22

Connection PeerLongAddress PeerInfo
00 00042519180186bc 09
-----
Chat Window:
-----
$$
----- From [04251918186bc] : Hello World!
$$
```

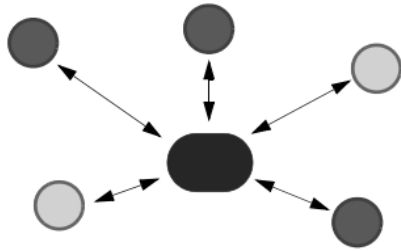
7. Simple_Example_Star Reference application

The simple example application code focuses on the simplicity of the MiWi™ DE protocol stack application programming interfaces. It provides a clean and straightforward wireless communication between two devices with less than 30 lines of effective C code to run the stack in application layer for both devices. In this application, following features of MiWi™ DE protocol stack has been demonstrated:

- Establish connection automatically between Pan Coordinator and End device
- Broadcast a packet
- Unicast a packet from one end device to another end device through Pan Coordinator.
- Apply security to the transmitted packet.

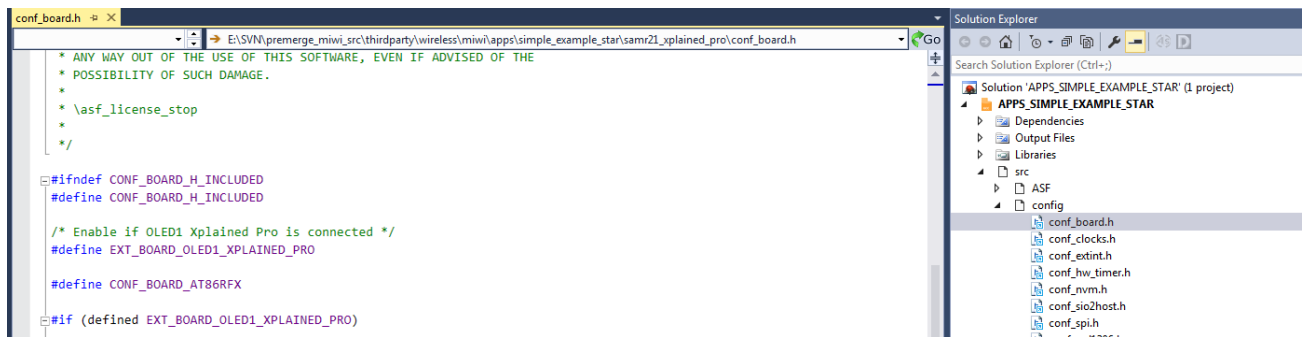
Star Network is supported on 3 different Hardware Development Boards and works the same way as discussed below for all the configurations.

Star Network Configuration

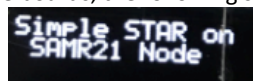


DEMO OPERATION

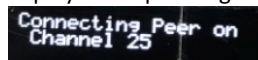
To run full featured Simple_Example_Star application on SAMR21 XPRO or SAMR30 XPRO, User needs to connect OLED1 Xplained PRO. When you use OLED1 Xplained PRO along with SAMR21 XPRO or SAMR30 XPRO, then user needs to Enable the macro “`#define EXT_BOARD_OLED1_XPLAINED_PRO`” in `conf_board.h` file.



1. On powering the boards, the following splash screen message will be displayed on the LCD screen.



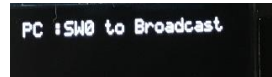
2. The LCD screen displays the operating channel



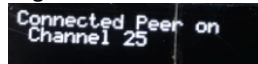
3. If no network is found, the node creates its own network and acts as a MiWi™ PAN Coordinator. The LCD display changes to:

SAMR21ZLLEK

SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO

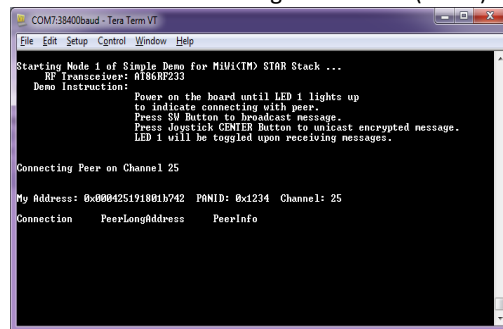


4. If any other MiWi™ PANCO node is found in the vicinity, then it gets connected to the PAN Coordinator as End Node. The LCD display changes to

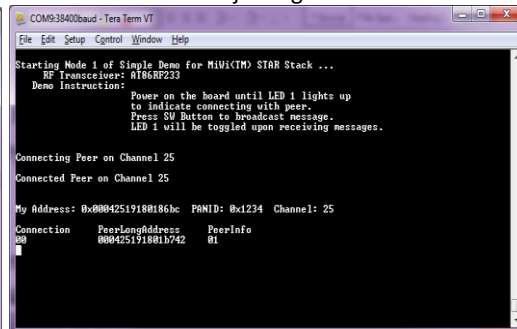


Hyper terminal shows below upon successful connection.

The Node which is starting the network(PANC)



The Node which is joining the network



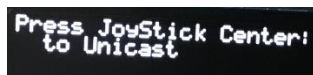
Note:

For every 15 seconds, PAN CO broadcasts the connection table to all the end nodes and End nodes sends the link status to PAN CO. After the PAN Coordinator has established a network, power on a second node and follow the instructions in step #5 above to join the PAN Coordinator. This process may be repeated to add any number of Nodes to the network.

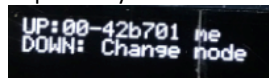
5. After getting connected to PAN Coordinator, the LCD displays options to Unicast a message to either the Pan Coordinator or to another Node in the network.

SAMR21ZLLEK

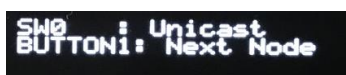
SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO



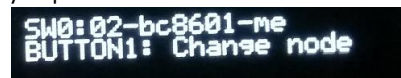
If you press Joystick Center button



Pressing Joystick **UP** sends the unicast message.



If you press BUTTON 1 will show the next node to choose

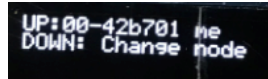


Pressing SW0 button sends the unicast message.

8. Pressing Joystick DOWN push button on the Node on SAMR21ZLLEK or Pressing BUTTON 1 on SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO, displays the address of the next node in the Unicast address selection list. The LCD Screen will display the three bytes of short address followed by “me” keyword indicating its own MAC address in the list, or MAC address of the next

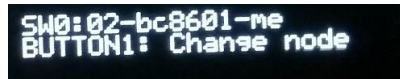
node in the Unicast address selection list. Depending on the location within the list, the LCD screen will show one of the following text:

SAMR21ZLLEK



UP: 00-42b701 me
DOWN: Change node

SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO



SW0: 02-bc8601-me
BUTTON1: Change node

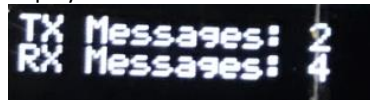
7. Now if Joystick UP push button on SAMR21ZLLEK or SW0 on SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO is pressed at End Node, a unicast message is sent to PAN Coordinator in the case of

xx-xxxxxx-me or to the destination node as indicated by xx-xxxxxx. After a successful transmission, the TX value

gets incremented at the source End Node. The RX value gets incremented at the destination node (PAN CO) and three

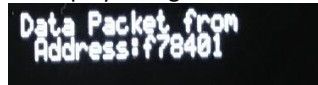
bytes of source MAC address are displayed.

In Nodes, LCD will display the number of transmitted and received messages:



TX Messages: 2
RX Messages: 4

After One second the display changes to



Data Packet from
Address: f78401

8. If Joystick DOWN push button on SAMR21ZLLEK or BUTTON 1 on SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO is pressed at End Node, the LCD displays the next node present in the connection table.

UP: 01-XXXXXX

DOWN: Change Node

Note: At End Nodes, UP push button on SAMR21ZLLEK or SW0 button on SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO is used to unicast message to the selected node. DOWN push button on SAMR21ZLLEK or BUTTON 1 on SAMR21XPRO+OLED1XPRO or SAMR30XPRO+OLED1XPRO is used to change/select the other destination node of the unicast message.

9. Press SW button at PAN CO to broadcast message to all the End nodes in the network. The LCD displays

the incremented TX values.

TX: xx , RX: yy

Message Count

Note: Whenever the destination node receives the message from source (End Nodes or PAN CO) RX value gets incremented. The respective nodes display the three bytes of source MAC address from which they have received the message.


10. After few seconds the display reverts to following messages on the LCD Display

At PAN CO:



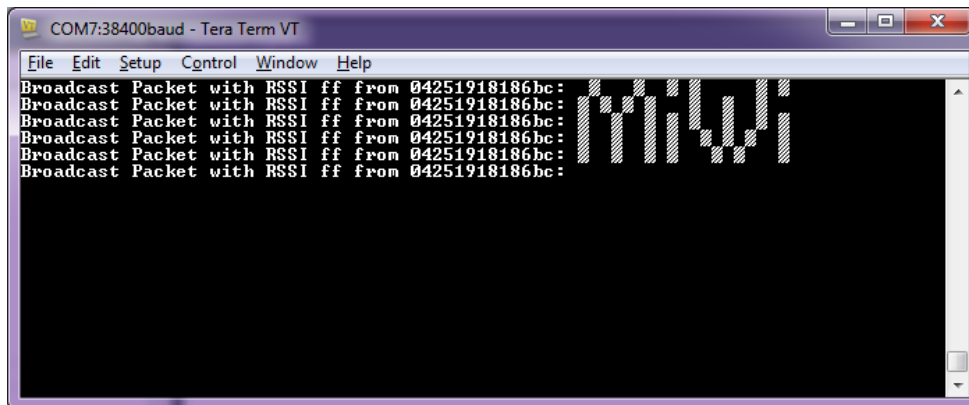
PC :SW FUNC
to Broadcast

At Source or Destination End Nodes:



Press Joystick Center:
to Unicast

Hyper terminal shows the below upon many broadcasts from PAN Coordinator.



11. By default, Network Freezer is enabled in the application. Network Freezer feature is used to store critical network information into the non-volatile memory. When the device reboots or power is recycled, it will try to restore the persistent data from the non-volatile memory and may continue to operate in the same network with existing network information.

If Power is recycled on an Node, the description given below may be used to restore the persistent data if available in the node using the Network Freezer.

- After Step1, below message will be shown on LCD. User must press SW button within 5 seconds to boot the application from stored network parameters or User can press and hold the SW button immediately after power recycle till the LCD shows the below message.



SW: Use Nwk Freezer
Press in 5 sec

- If SW button is pressed properly within the timeout, application will restore the data from Network Freezer and the below message will be shown on LCD.



Restoring Network !!

8. WSN Demo Mesh Reference application

8.1 Overview

The network and radio frequency performance of the hardware components is demonstrated with the WSNDemo application, which is based on the MiWi™ API.

This application consists of the embedded firmware, which supports functions for PAN Coordinator, Coordinator, and End device, and the GUI visualization application, WSNMonitor, which is run on a PC. In WSNDemo, the nodes communicate based on a proprietary messaging protocol.

The package includes the WSNMonitor PC application in binary format and the WSNDemo embedded application is available in binary format and source code.

The source code for the WSNDemo application can be modified and extended, making it possible to develop WSN applications for a variety of application scenarios.

End devices, coordinators, and the PAN coordinator devices emulate the sensor data reading for light and temperature sensors, and forward collected data to the WSNMonitor application for visualization.

End devices follow a duty cycle to transmit data to the coordinator. Using the serial connection, the PAN coordinator transmits the received packets, along with its own sensor data (or emulated sensor data), to the WSNMonitor application. Those transmitted values are displayed on WSNMonitor panes as temperature, light, and battery level measurements.

WSNMonitor also visualizes network topology by drawing a tree of nodes that have joined the network. For each of the nodes, parameters like node address, node sensor information, and link quality data are displayed.

RSSI indicates a link's current condition and is measured in dBm. The RSSI resolution is 3dBm. LQI, a numeric parameter defined within the 0 to 255 range, is used to measure the link quality. Larger values mean a better link, while values close to zero indicate a poor connection.

By default, Network Freezer is enabled in WSN Demo application. Network Freezer feature is used to save or store critical network information into the non-volatile memory.

When the device reboots or power is recycled, it will try to restore the persistent data from the non-volatile memory and may continue to operate in the same network with existing network information.

By default, Sleep Feature is enabled in End device application. Both MCU and Transceiver are put into sleep when device is idle.

8.2 Launching the demo

1. Assemble devices
2. Program devices with firmware images. One node shall be programmed as PAN coordinator, others as coordinators or end devices.
3. Connect the PAN coordinator node to the PC using the serial interface.

4. Run WSNMonitor

Use the following setting for the serial connection of the WSN Monitor:

BAUD RATE: 38400

PARITY: None

DATA BITS: 8

STOP BITS: 1

FLOW CONTROL: Off (On for the XPRO board)

5. Observer coordinator node in the WSN Monitor.

6. Power on the rest of the nodes and observe them displayed in the WSNMonitor.

7. Select any coordinator node and click on the bulb icon next to it, observe the device to blink its LEDs.

8. SW Button on the board is used to erase all the persistent items in the memory and make the device as a factory new device.

8.3 Network startup

The PAN coordinator organizes the wireless network automatically. Upon starting, every node informs the network of its role. When the PAN coordinator is powered on, it switches to an active state even though no child node is present. This is normal, and it indicates that the PAN coordinator is ready and that child nodes can join the network with the coordinator's PAN ID. By default, the coordinator uses PAN ID 0x1234, which is recognized by all coordinators. The PAN ID can be modified by the user through the application's configuration file.

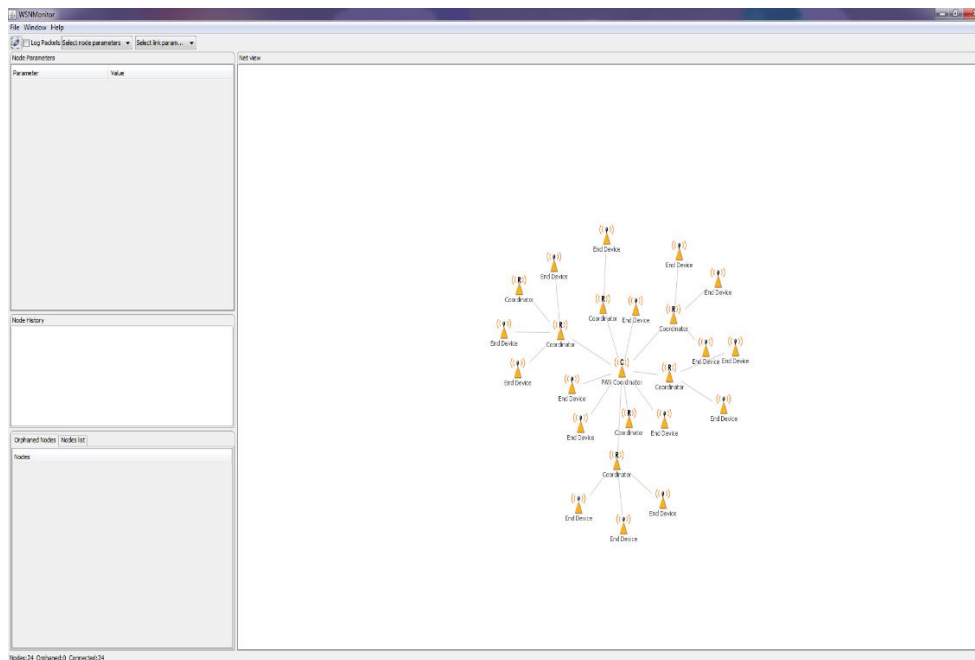
If the PAN coordinator is absent or has not been turned on, the coordinators and end devices will remain in the network search mode. In this mode, coordinators scan the channels specified in the channel mask in search of a network. By default, the channel mask in application provided with the release contains a single channel. In rare cases, if the frequency corresponding to the radio channel is busy, the coordinator node may stay in the network search mode. If this happens, it may become necessary to change the application's channel mask to select another channel by changing the application's configuration file and recompiling the application.

Network health can be monitored through the WSN Monitor application described in the next section.

8.4 WSNMonitor

WSNMonitor is a PC counterpart to the WSNDemo embedded application, and can be used to display MiWi™ Mesh network topology and other information about a wireless sensor network. A typical WSN Monitor screen is shown below. It contains topology, sensor data, and node data panes and application toolbars.

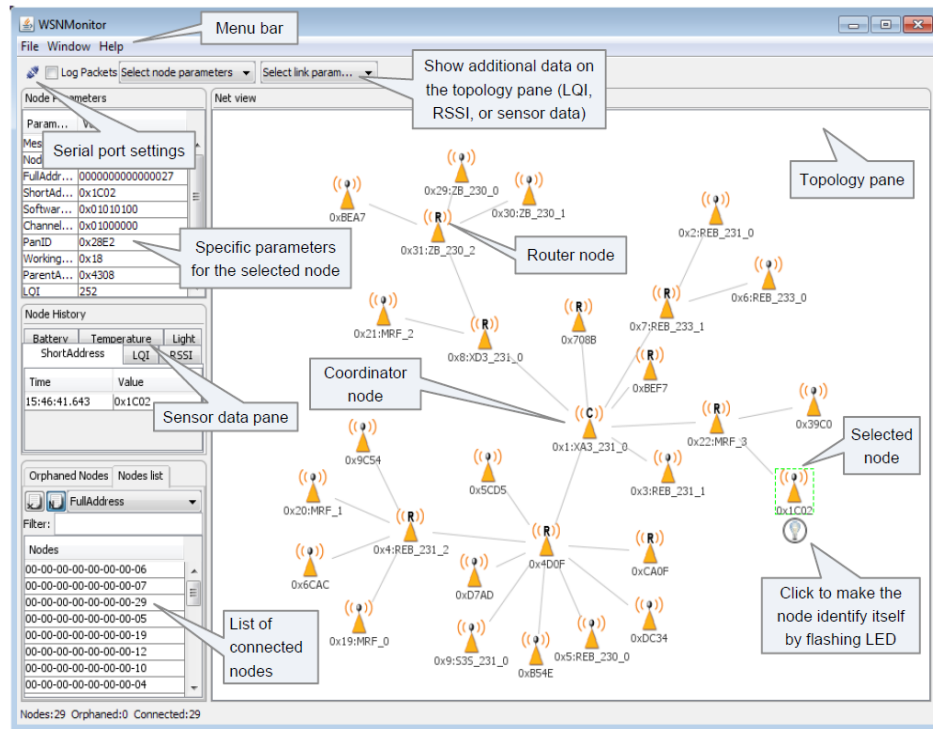
Example Topology of MiWi™ Mesh protocol is shown below:



The topology pane displays the network topology in real time, which helps the user monitor the formation of and dynamic changes in the network while nodes join, send data across, or leave the network. The network topology is constructed on the basis of next-hop information for each of the nodes, and each link is also tipped with RSSI and LQI values. Each of the nodes displayed is depicted by an icon, with the node's address or name below and sensor readings to the right of the icon, if required by settings.

The sensor data pane displays data coming from onboard sensors of the selected node. It is presented in graph and table form. Other parameters can be observed for each node in table form. The node data pane includes a sensor selection combo-box, which is used to switch between sensor types.

By default, in the topology pane, nodes are labeled with their short addresses. However, another title can be assigned to any desired node by a double click. If "Cancel" is pressed in the opened window, the node's title is set back to the short address.



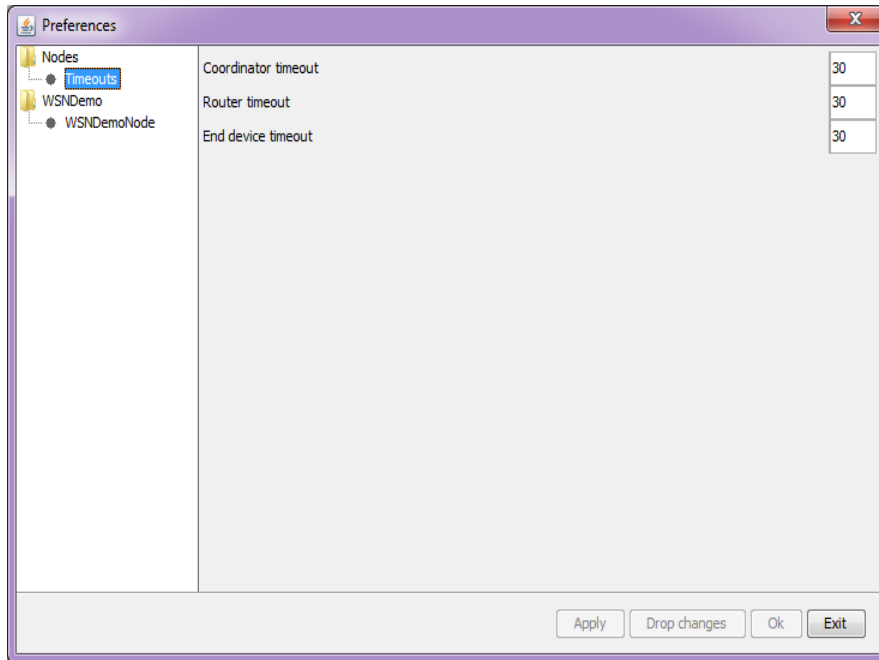
8.5 Identifying nodes

When a user clicks a node in the topology pane a button that can be used to identify the node appears under the node's icon. When the user clicks this button WSNMonitor sends a command, which is delivered to the PAN coordinator through the serial connection and wirelessly to the target node. The target node, receiving the command, blinks with its LED for several seconds.

8.6 Node timeouts

The Window/Preferences menu of WSNMonitor contains a number of parameters used to control application behavior. Timeouts are used to tune visualization of PAN Coordinator, Coordinator, and end devices as the nodes disappear from the network each time a connection is lost, power is down, or a reset has occurred. A node timeout corresponds to the time the WSNMonitor application waits for a packet from a particular node before assuming that the node is no longer part of the network. Note that this value does not correspond to the frequency with which data are transmitted by each type of device. To get smooth topology visualization, setting timeouts to 20 seconds is recommended for PAN Coordinator, and Coordinator, and 30 seconds is recommended for an end device. Assuming a default application configuration, these timeouts cover three periods between sending a packet, and so at least three packets would need to be lost before a node is removed from the WSNMonitor topology pane.

WSN Monitor preferences menu



8.7 Sensor data visualization

Each board sends temperature/light/battery sensor readings (or emulated values) to the PAN coordinator, which in turn sends it to the PC. WSNMonitor displays the readings from onboard sensors next to a node icon inside the topology pane. A corresponding option can be selected in the node/link parameters from the quick settings toolbar.

The user can select any node in the topology pane to monitor the node's activity and see the node data in one of three different forms:

Text table

Chart

The onboard sensor's data displayed next to each node in the topology pane. These values are also tipped with arrows indicating whether the value increased or decreased in relation to the previous sample. A given node is selected when it is clicked on and a dashed frame is visible around it.

The same values are shown on the sensor data pane, enabling the user to observe how the values change over a period of time. The sensor data pane includes a sensor selection combo-box. Use the button on the sensor control toolbar to display the desired types of sensor data.

9. OTAU in WSN Demo Mesh application

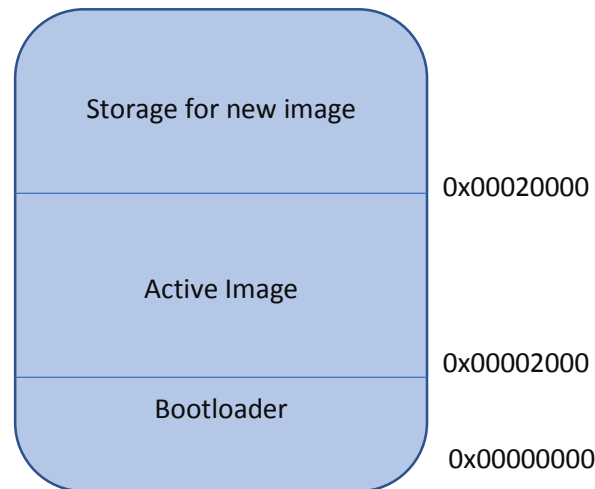
9.1 Required Software

1. Bootloader PC Tool, Available in <ProjectDir>\thirdparty\wireless\miwi\services\otau\tools
2. WiDBGSetup.msi, Available in <ProjectDir>\thirdparty\wireless\miwi\services\otau\tools

9.2 OTAU Client

9.2.1 Memory Layout on Client

In order to upgrade the application from WiDBG UI, the serial bootloader in AVR2054 is modified to support internal flash swap functionality. The last page (of size 64bytes) of flash is used for storing the information for bootloader to perform flash swap. The memory layout of the client is as shown below...



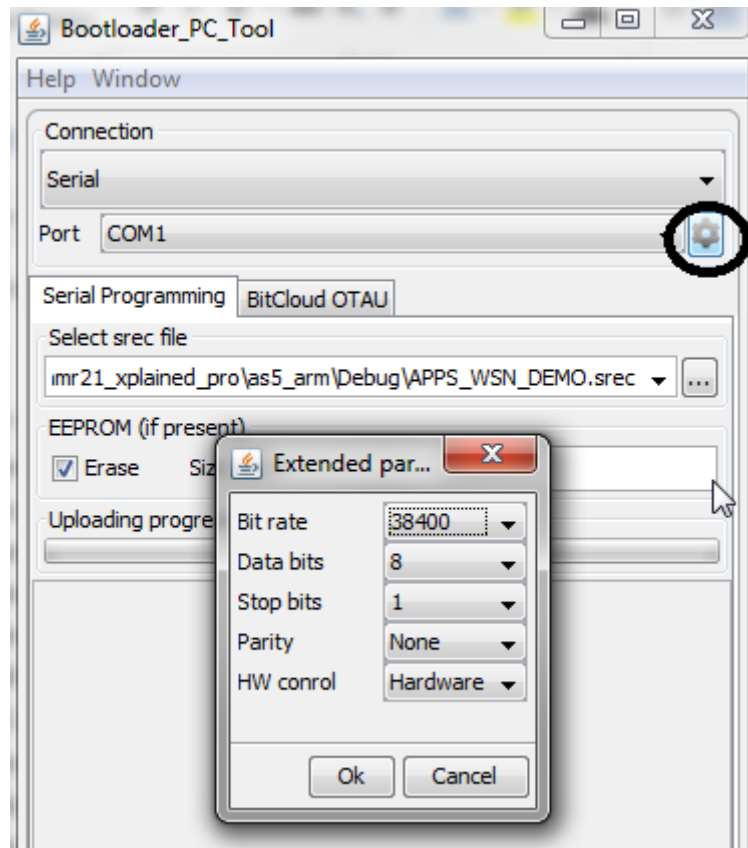
9.2.2 Steps to Program Client Firmware

1. Select a WSN Demo application with required device type as mentioned in Section-3.
2. Open
“<ProjectDir>\thirdparty\wireless\miwi\services\pds\src\wl\linkerscripts\samr21\gcc\samr21g18a_flash.ld”.
3. Modify the Memory region of rom from 0x00000000 to 0x00002000 as shown below...

```
/* Memory Spaces Definitions */  
MEMORY  
{  
    rom      (rx)  : ORIGIN = 0x00002000, LENGTH = 0x00040000  
    ram      (rwx) : ORIGIN = 0x20000000, LENGTH = 0x00008000  
}
```

4. Rebuild the project.

5. Program the Client node with bootloader elf file available in
<ProjectDir>\thirdparty\wireless\miwi\services\otau\tools.
6. Open "Bootloader PC Tool" and configure the serial settings as shown below...

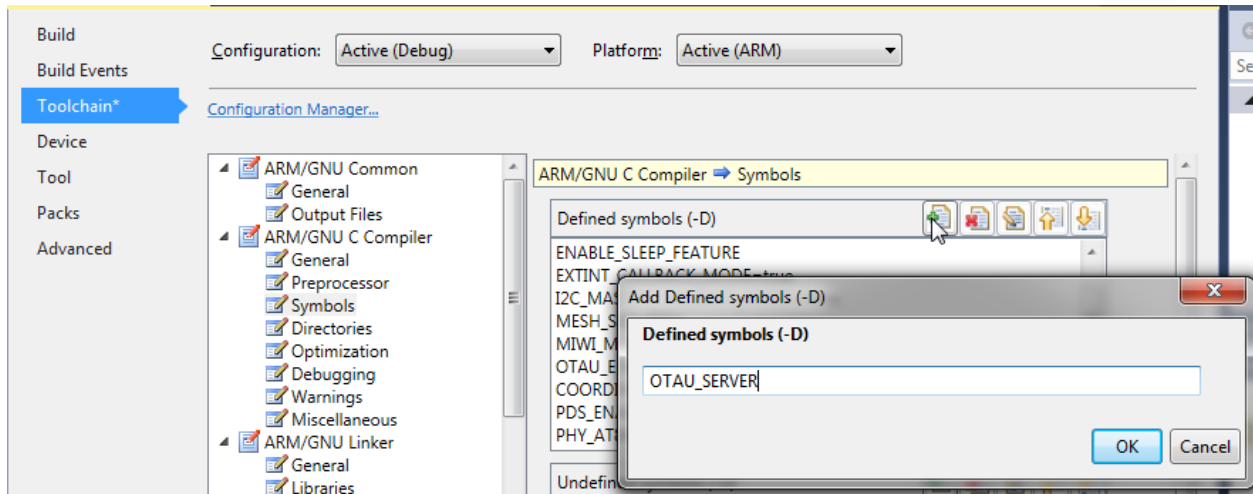


7. Select the built. srec file from the project directory and click on Upload button.
8. In order to start upload, reset the client node.

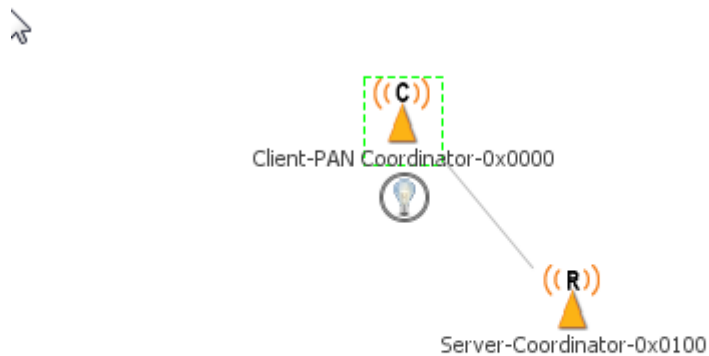
9.3 OTAU Server

Note: PAN Coordinator cannot be used as OTAU Server as PAN Coordinator uses serial interface to communicate to WSN Monitor. However, PAN Coordinator can be configured as OTAU Client.

Configure the WSN Demo application to Coordinator as explained in Section 3 and add the OTAU_SERVER in symbol as shown below and Click OK.



Now compile and program the OTAU server node. When the node joins the network with available coordinator, the WSN monitor will be updated as shown below...



1.1 Atmel WiDBG for OTAU

Once the installation of WiDBGSetup.msi is finished, you can see the shortcut created on your Desktop. Double click and open the WiDBG now.

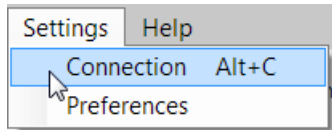


Figure 1 Opening WiDBG Tool

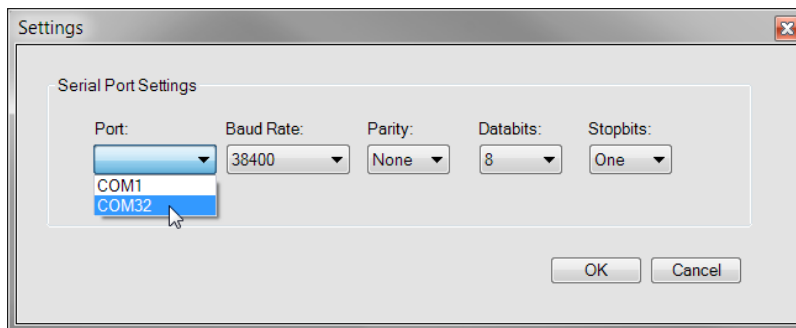
1.2 Starting the Session

This section explains about how to start the session using this tool.

Make sure you have connected the Server node to the Host Machine (PC) via USB. Once the application is opened, go to Settings menu and Select Connection.

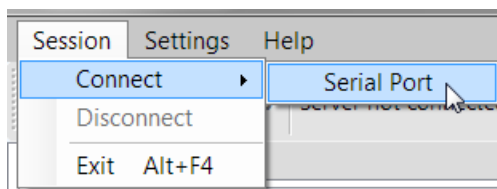


Connection Settings



Configuring the Serial Port

- Select the desired COM port and click OK to save the settings.



Connecting the OTAU Server

- Now Go Session-> Connect-> Serial Port.
- Once it successfully connected, we will get the status as successfully connected.
- Select the desired Channel number from the drop-down box available.

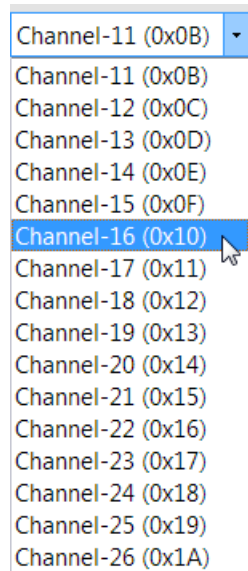


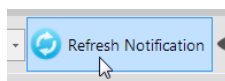
Figure 2 Setting the Channel number

- After selecting, Click **Start**  to start the debugging session.

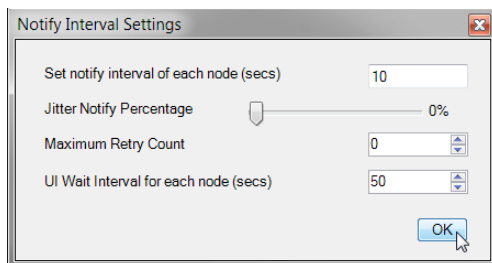
1.2.1 Notification Settings

Once the session is started, all the OTAU clients will start notifying after configuring Notification interval. In this section, we will explain how to configure the notification intervals of the clients.

Depending on the number of client nodes, user can adjust the interval so that the traffic on the channel can be minimized for effective debugging during the session. By default, all clients will notify the server for every 5 seconds. This can be changed at any time only if the number of nodes is becoming higher.



Refresh Notification



Notify Interval Settings for the Client

Click on Refresh Notification button as shown in Figure 7 and you will get a window as like shown in the Figure 8. Here, we have understood and set the following properties as described below.

- **Notify Interval**

Set notify interval of each node (secs)

Client Notify Interval Setting

This is the interval that we are assigning it on each client node of the network. Once it receives the value from the tool then it will start sending the notification to us according to the specified interval.



TIP

For example: Let's say, we have 5 clients connected to us and if we set 10 seconds, then each client will send a notification to WiDBG on the 10th, 20th ... and so on after it receives the value from tool until we change or stop the session.

- **Jitter Notify Percentage**

Jitter Notify Percentage  0%

Jitter Notify Percentage Setting

This is the percentage value we are assigning to each client to randomize the time to avoid the collision when the other client is notifying at the same time. By default, there will be no randomization is applied to clients.



TIP

**Increase the percentage if network is Large
Decrease the percentage if it is Small**

- **Maximum Retry Count**

Maximum Retry Count

Retry Count Setting

If the notification is still missed out by the tool due to heavy traffic, force the client to retry the notification for specified count value.

- **UI Wait Interval**

UI Wait Interval for each node (secs)

UI Wait Interval Setting

Also, the user should set the minimum wait time for the tool to decide a node whether it's active or not using **UI Wait Interval** settings from **Preferences** window.



TIP

User should manage to set the timing accordingly at UI Wait Interval based on Notify Interval Value

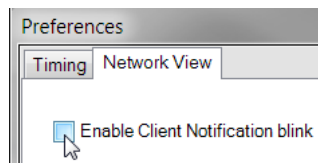
By default, the timing is configured as.,
Notify Interval for Client = 5 seconds
UI Wait Interval for WiDBG = 15 seconds

So, WiDBG waits for 15 seconds which is 3 times the client notify interval setting to decide a node as ACTIVE or not.



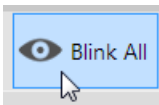
TIP

Users can see the client's notification by enabling the checkbox from Settings -> Preferences -> Network View Settings.



Enabling Client Notification Blink on Network View

Once it is enabled, the node will blink as GREEN color for a moment when it is responding to the tool.



Identifying all the Clients using Identify All button

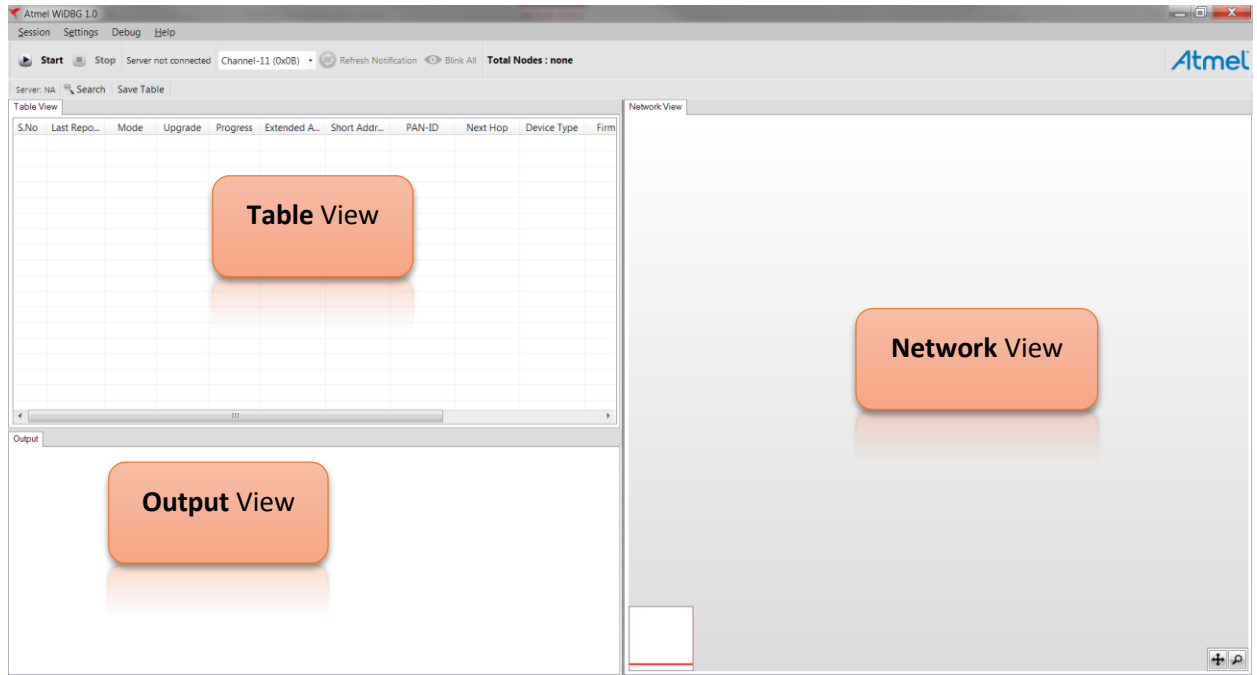
At any time, we can use the **Blink All** button to blink all the Nodes connected to the network. This eventually blinks by toggling its board LED.

1.3 Main Window

The Main window summarizes you about every information around your connected network. Once the session started, this window tells the user about the stability of the system at any point of time. It basically provides following information:

1. *Table View*
2. *Network Topology View*
3. *Output View*

Each view is designed to provide useful information about the connected clients. The network view and table view are interconnected to each other. It gives clarity to the user to perform analyzing tasks on each and every node. Also, both the views are lively in nature and reflects the status of nodes very immediately with the time stamp. We will discuss more about the usage in detail in later sections of the document.



Main Window

1.3.1 Table View

The Table view will list down all the nodes that are connected to the network. It holds useful information about the node. Each node occupies a row with several columns as described below:

S.No	Last Repo...	Mode	Upgrade	Progress	Extended Address	Short Address	Device Type	Parent	LQI	Supported ...	Firmware	Firmware v...	Board	Board versi...
1	17:08:53	APP	Upgrade		0x1122334455667788	0x1								
2	17:11:40	APP	Upgrade		0x4250000005002	0x5002	Router	0x1	228	0x3	LWMesh	1.2.1	SAMR21XPro	1.0.0
3	17:11:35	APP	Upgrade		0x4250000005032	0x5032	Router	0x5019	236	0x3	LWMesh	1.2.1	SAMR21XPro	1.0.0
4	17:11:36	APP	Upgrade		0x4250000005027	0x5027	Router	0x1	220	0x3	LWMesh	1.2.1	SAMR21XPro	1.0.0
5	17:12:17	APP	Upgrade		0x4250000005015	0x5015	Router	0x4	244	0x3	LWMesh	1.2.1	SAMR21XPro	1.0.0
6	17:12:24	APP	Upgrade		0x4250000005011	0x5011	Router	0x5027	204	0x3	LWMesh	1.2.1	SAMR21XPro	1.0.0
7	17:12:36	APP	Upgrade		0x4250000005016	0x5016	Router	0x501C	172	0x3	LWMesh	1.2.1	SAMR21XPro	1.0.0
8	17:12:41	APP	Upgrade		0x6	0x6	Router	0x5026	244	0x3	LWMesh	1.2.1	RFR2XPro	1.0.0
9	17:12:50	APP	Upgrade		0x4	0x4	Router	0x6	240	0x3	LWMesh	1.2.1	RFR2XPro	1.0.0
10	17:12:59	APP	Upgrade		0x3	0x3	Router	0x5030	252	0x3	LWMesh	1.2.1	RFR2XPro	1.0.0
11	17:13:03	APP	Upgrade		0x4250000005018	0x5018	Router	0x1	208	0x3	LWMesh	1.2.1	SAMR21XPro	1.0.0

Table View

Each row will be updated whenever the client sends data to WiDBG and the last reported time will be updated. Each node supports two operational modes called PHY mode and APP mode. By default, all the nodes will operate on APP mode, User can switch to PHY mode when the node is not responding to the tool.

User can easily upgrade the node using the button provided and it will show the live progress in the progress bar as like below figure.

Upgrade	Upgrade Progress	Extended Address
Upgrade		0x112233445566...
Upgrade		0x42500000001EB
Upgrade		0x42500000001EF

Upgrade button

The remaining columns explains more about the node's detailed information.

Extended Address	Short Address	Device Type	PAN-ID	Parent	Firmware	Firmware version	Board	Board version
------------------	---------------	-------------	--------	--------	----------	------------------	-------	---------------

Table columns explaining Node's detail

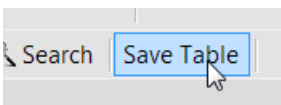
When any node goes IDLE or Not-Responding state, the Short-Address column will turn in to Red color to indicate its state.

d Address	Short Address	Device Type
3445566...	0x0	
000001EB	0x1EB	Router
000001EF	0x1EF	Router
000001F1	0x1F1	Router
000001EC	0x1EC	Router
000001EE	0x1EE	Router
000001F3	0x1F3	Router
000001ED	0x1ED	Router

Idle state indication at Table

1.3.2 Saving the Table View

On any time, the contents of the table view can be saved as CSV file by using **Save Table** option.

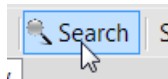


Saving Table View

1.3.3 Searching the Nodes

To locate the node on a Table and Topology view, user can search using the node's addresses.

Click **Search** available above the Table View.



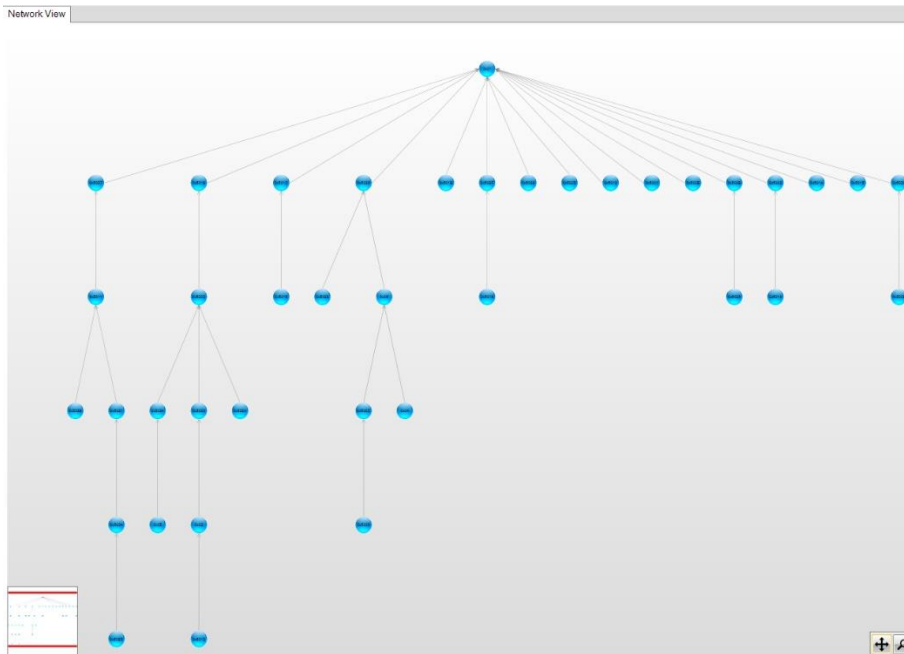
Enter either any one of the address of the Node to be located.

<input type="text" value="425000000502E"/>	Or	<input type="text" value="Type it's Short Address here.."/>
<div><div>425000000500A</div><div>425000000502E</div><div>4250000005031</div></div>		<div><div>0x500A</div><div>0x5031</div><div>0x5</div></div>

Searching the Node

1.3.4 Network Topology View

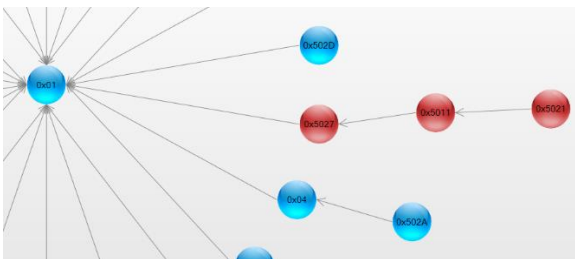
The network view is an advanced diagrammatic representation provided to the user to view the topology of the network.



Network Topology View

As soon as the session started, this view gets active and starts populating the view. Each node will join with its parent and forms the network. It shows the hopping immediately when it happens in the network. Also, when the node on the table is left-clicked on mouse, the corresponding node in this view turns orange and user can zoom in/out or pan the diagram, when the network becomes huge.

As like the table, this view also actively reflects the node status as red or blue color based on the node state.

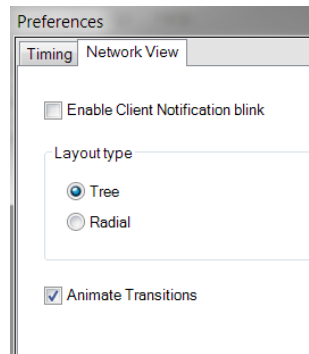


Idle state indication at Network View

1.3.4.1 Layout Type

The topology view can be configured as two different layout types.

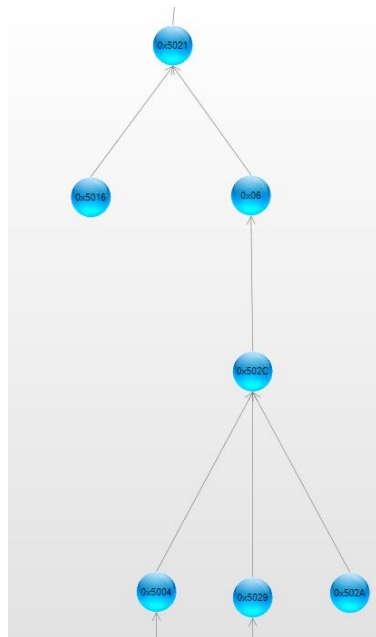
Go to **Settings -> Preferences -> Network View**



Layout Type Setting

- **Tree**

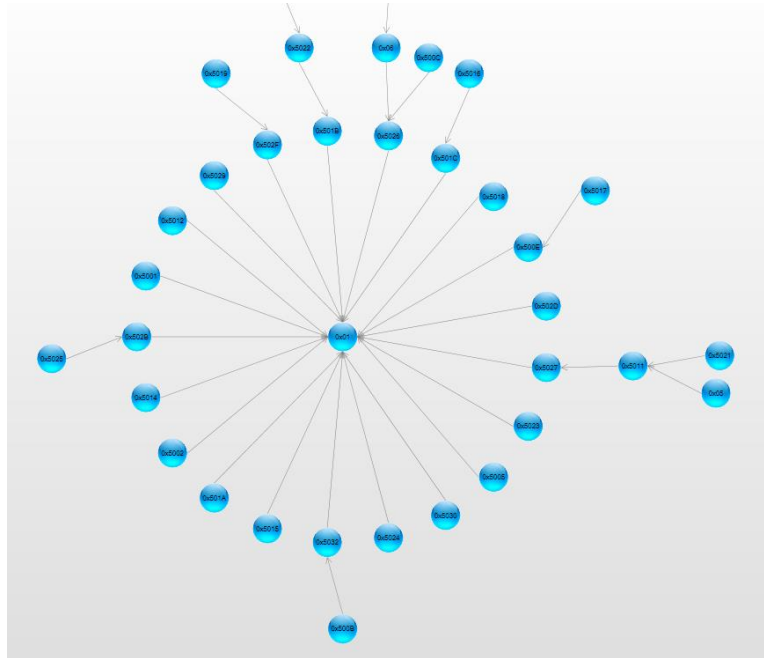
It is the default layout view when the tool runs for the first time. It populates the view in a tree like structure as shown in the Figure 19 below.



Tree Layout View

- **Radial**

It forms the nodes in a circular fashion as shown in the Figure below.



Radial Layout View

1.3.5 Output View

This view is used to understand all the activities happening in the tool. It logs all the information that the user requested and provides the status or progress about it. All the success / failure messages can be seen using this view.

Output

```
20:36:08 : WiDBG has started collecting data..
20:36:08 : Getting server information...
20:36:08 : Server : 0x1122334455667788 has responded.
20:36:08 : Waiting for the client nodes from Channel-21 (0x15)
20:36:08 : Please wait...
20:36:10 : Node : 0x42500000001F1 has joined the network
20:36:11 : Node : 0x42500000001F2 has joined the network
20:36:11 : Node : 0x42500000001F4 has joined the network
20:36:11 : Node : 0x42500000001EF has joined the network
20:36:11 : Node : 0x42500000001EE has joined the network
20:36:12 : Node : 0x42500000001F3 has joined the network
20:36:12 : Node : 0x42500000001EB has joined the network
20:36:12 : Node : 0x42500000001F0 has joined the network
20:36:12 : Node : 0x42500000001ED has joined the network
20:36:12 : Node : 0x42500000001EC has joined the network
```

Output View

1.4 Upgrading the Node

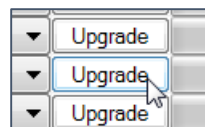
As we discussed previously, a node can be upgraded using the table view. If we want to upgrade a node, one has to follow the below steps.

- First the tool should know what is the firmware version is currently running on the node. So, user should right-click on the row and should first get the information like below.

0x42500000001EF	0x1EF	Router
0x42500000001F5	0x1F5	Router
0x4250000000000		Router
0x4250000000000		Router
0x4250000000000		Router
0x42500000001ED	0x1ED	Router

Getting the Firmware and Board information

- Then the server node will ask the client node about its version details and populates it on the respective columns.
- Now, Click the Upgrade button of the corresponding node.



Doing Upgrade on the respective node

- Select the HEX file using the Browse button.

- Details about the HEX file can be sent to client using below configuration but this is not handled in MiWi for this release. As the same can be modified in `"otauClientInfoIndication_t clientInfo"` in `client_notify.c` of client project to see the upgraded image's information after upgrading

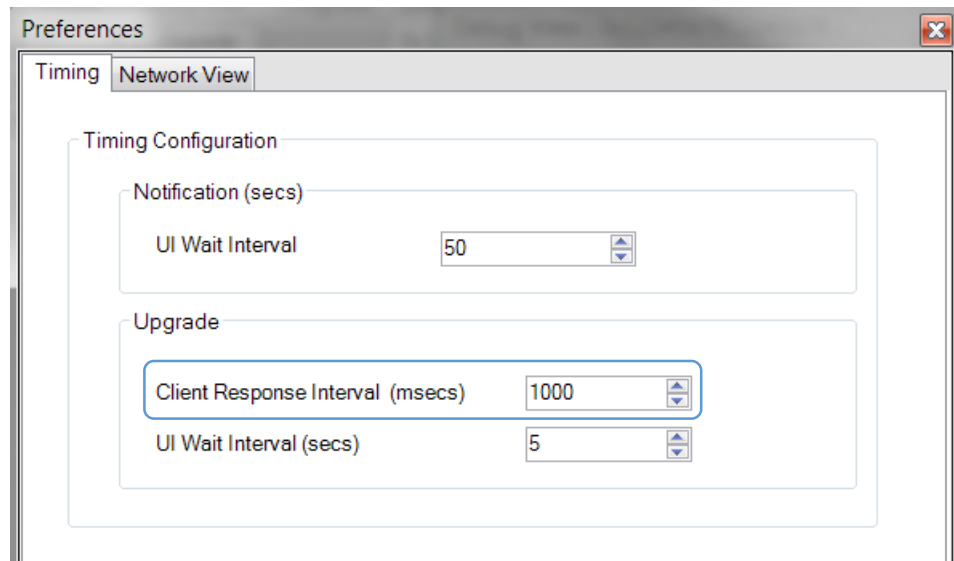
Selecting the HEX file and entering the details about the firmware



INFO

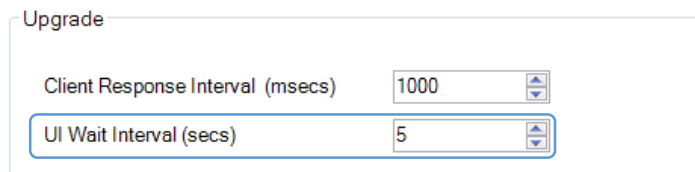
The details about the HEX file on each field should not exceed the length of `client_information` sequence of the `user_config.xml` file which is explained in detail at XML Configuration section.

- Click OK now. The upgrade will begin and progress bar will start increasing once the client receives the hex file data.
- The upgrade data will be sent to the client for every 1000 milliseconds by default. If we want to modify this interval, Open **Preferences** from **Settings** menu and change the *Client Response Interval* under the tab Timing.



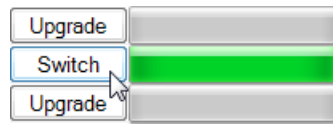
Setting the Client Response Interval for Upgrade

- If we did not receive any response from the client then the tool will stop the upgrade process after the certain period of time. This can be configured again from the same Preferences window, *UI Wait Interval* value at the Upgrade section.



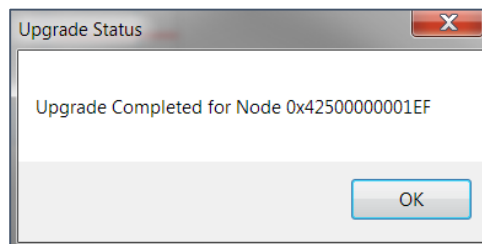
Setting the UI Wait Interval for Upgrade

- Once the full image sent to the client, it will be ready to switch the image as per the new image file. The user can switch to the new image whenever wanted.



Switching to new image

- If we click switch, then the client writes into its flash memory and restarts as per the new image file selected during this process.
- Once it is successfully switched, then we will get a notification about the status like below figure.



Upgrade completed status window

- Now the upgrade is completed.