MiWi Library

Microchip Libraries for Applications (MLA)

Table of Contents

MiWi Library

1 MiWi Library

1.1 Introduction

MiWi™ Development Environment with MiMAC and MiApp Interfaces

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 are able to 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 (MRF24J40, MRF49XA and MRF89XA for this release) are called by protocol layers via MiMAC interfaces. Configuration files are also presented in each layer. Following diagram shows the architecture of Microchip MiWi™ DE.

The details of the major modifications can be found in the Release Notes.

1.2 Legal Information

This software distribution is controlled by the Legal Information at www.microchip.com/mla_license

1.3 Release Notes

Microchip MiWi(TM) Development Environment Software Stack * Version 5.35.0, March 2017

Microchip MiWi™ Development Environment (MiWi™ DE) protocol stack provides simple wireless connectivity for short-range, low data rate and low power applications. Microchip MiWi™ DE protocol is royalty free as long as implemented on Microchip PIC microcontroller and radio frequency transceiver. Please refer to the attached license agreement for details.

The MiWi™ DE source code is released with applications to demonstrate communications between two RF devices. The source code for each device is located in individual directories under "apps/miwi". In addition, the directory "miwi" is for MiWi™ stack source code.

The main updates from earlier versions are:

- MiApp_UnicastStar() function is updated to use End Device Index as input. Now a End Device can send message to another End Device using the index of the Destination End Device. In early versions of MiWi Star release, it was the responsibility of the Application Developer to include the Destination End Device address and the forward packet command as part of Data Payload. In this version of stack , application developer can simply use the destination end device index as the input parameter to MiApp_UnicastStar() function and send a Data Packet to another End Device connected to Pan Coordinator.

Note : Detailed changes and features supported in the new library based MiWi Mesh Stack are available in the example project folder of the MiWi Mesh folder.

Resource Usage

Peripherals

Type/Use	Specific/Configurable	Limitations
UART for hyper- terminal/Teraterm output	Select via pin definitions in system_config.h	None
Timer for protocol timing when using a 16 bit microcontroller Timer for protocol Timing when using a 8 bit microcontroller	16bit Timer	Timer is preferred to be configured to represent 16us for one tick Timer is preferred to be configured to represent 4 us for one tick
SPI for RF transceiver	Select via pin definitions in system_config.h	Both hardware SPI or software big-bang can be used.
Digital I/O pins to RF transceiver	Select via pin definitions in system_config.h	Must be able to be configured as external interrupt pin or interrupt-on-change pin; must have a pull-up

Limitations

1. Microchip C18, C30, XC32 compilers are no longer supported with this release. If support is required for XC32, please contact us.
2. High data rate for MRF49XA may require MCU running at faster speed. This is due to the nature of 16-bit RX buffer used in MRF49XA.

3. MRF24XA under proprietary mode supports data rates up to 2 Mbps and while using higher data rates it is advised to run MCU at faster speed.
4. Previous MiWi Mesh and MiWi Mesh Pro have been deprecated , new library based MiWi Mesh stack has been added.

1.3.1 Online References and Resources

This section includes useful links to online MiWi development resources.

Description

Note: Newer versions of the documents below may be available. Please check www.microchip.com/miwi for the latest version.

MiWi Design Center

http://www.microchip.com/miwi

User Guides and Reference Documentation

MiWi® Demo Kit – User's Guide (2.4 GHz MRF24J40, 868 MHz MRF89XA, 915 MHz MRF89XA)

8-bit Wireless Development Kit User's Guide

MRF24J40MA/MB PICtail/PICtail Plus Daughter Board User's Guide

MRF89XAMxA PICtail/PICtail Plus Daughter Board User's Guide

Application Notes

AN1284 – Microchip Wireless (MiWi) Application Programming Interface - MiApp

AN1283 – Microchip Wireless (MiWi) Media Access Controller - MiMAC

AN1066 – MiWi Wireless Networking Protocol Stack

AN1371 – Microchip MiWi PRO Wireless Networking Protocol

AN1204 - Microchip MiWi P2P Wireless Protocol

General RF/ Wireless Application Notes

AN1192 - MRF24J40 Radio Utility Driver Program

AN1340 - MRF89XA Radio Utility Driver Program

AN1629 - Simple Link Budget Estimation and Performance Measurements of Microchip 2.4 GHz Radio Modules

AN1631 - Simple Link Budget Estimation and Performance Measurements of Microchip Sub-GHz Radio Modules

2.4 GHz and Sub-GHz Wireless Development Tools

(For Debugging and Sniffing)

ZENA Wireless Adapter User's Guide

Wireless Development Studio Help

1.4 Library Interface

1.4.1 Configuring the Library

MiWi(TM) Development Environment uses configuration files to regulate the behavior of the stack. There are three layers of configurations in application, protocol stack and RF transceivers respectively.

1.4.1.1 Application

Configuration in application layer defines the basic functionality of the wireless node. Usually, those configurations may differ among different wireless nodes in the same application, depending on the wireless node's role in the network and application.

Configurations in application layer include following categories:

* Choice of wireless protocol

* Choice of RF transceiver

* System Resources Definitions

* Enable/Disable functionalities according to application needs

* Application specific information

1.4.1.2 Wireless Protocol

Configurations in wireless protocol layer can be used to fine tune the behavior of wireless protocol. The possible configurations differ between different protocols.

1.4.1.2.1 MiWi(TM) P2P/star Network Protocol

Configurations can be used to fine tune the behavior of MiWi(TM) P2P wireless protocol.

Configurations can be used to fine tune the behavior of MiWi(TM) Star wireless protocol.

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

1. headers/systemconfig/hardware_configuration/miwi_config.h
2. headers/systemconfig/hardware_configuration/miwi_config_p2p.h

Note : MiWi Star Protocol is a extension of MiWi P2P protocol.

1.4.1.2.2 MiWi(TM) Mesh Network Protocol

Configurations can be used to fine tune the behavior of MiWi(TM) P2P wireless protocol.

Configurations can be used to fine tune the behavior of MiWi(TM) Star wireless protocol.

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

1. headers/systemconfig/hardware_configuration/miwi_config.h
2. headers/systemconfig/hardware_configuration/miwi_config_p2p.h

Note : MiWi Star Protocol is a extension of MiWi P2P protocol.

1.4.1.3 RF Transceivers

Configurations for RF transceivers specifies how RF transceiver work in MiMAC layer. The configurations in this layer may define frequency band, data rate and other RF related parameters. Those configurations differ between different RF transceivers.

1.4.1.3.1 MRF24J40 IEEE 802.15.4 Compliant 2.4GHz Transceiver

Configurations can be used to regulate Microchip MRF24J40 IEEE 802.15.4 compliant 2.4GHz transceiver.

Location of Header Files to regulate Microchip MRF24J40 IEEE 802.15.4 compliant 2.4GHz:

1. headers/systemconfig/hardware_configuration/config_24j40.h

1.4.1.3.2 MRF24XA IEEE 802.15.4 Compliant 2.4GHz Transceiver

Configurations can be used to regulate Microchip MRF24XA IEEE 802.15.4 compliant 2.4GHz transceiver.

Location of Header Files to regulate Microchip MRF24XA IEEE 802.15.4 compliant 2.4GHz transceiver:

1. headers/systemconfig/hardware_configuration/config_24xa.h

1.4.1.3.3 SubGHz Transceivers

Microchip SubGHz transceivers share some common configurations.

1.4.1.3.3.1 MRF89XA SubGHz Transceiver

Configurations can be used to regulate Microchip MRF89XA subGHz transceiver.

Location of Header Files to regulate Microchip MRF89XA subGHz transceiver:

1. headers/systemconfig/hardware_configuration/config_89xa.h

1.4.1.3.3.2 MRF49XA SubGHz Transceiver

Configurations can be used to regulate Microchip MRF49XA subGHz transceiver.

Location of Header Files to regulate Microchip MRF49XA subGHz transceiver:

1. headers/systemconfig/hardware_configuration/config_49xa.h

1.4.2 Using API

In this section, MiMAC and MiApp interfaces are defined in detail.

1.4.2.1 MiApp Interfaces

1.4.2.1.1 Call Back Functions

Callback Functions *

MiApp callback functions are called from the protocol stack to application layer. In most of the cases, the callback functions are defined as macros. If developer choose to implement the function, the macro can be commented out and replaced by a function call in the application layer.

Refer: AN1284 – Microchip Wireless (MiWi) Application Programming Interface - MiApp

1.4.2.2 MiMAC Interfaces

MiMac callback functions are called from the PHY and MAC layer. In most of the cases, the callback functions

are defined as macros. If developer see these functions being used in the MAC/Data link layer.

Refer: AN1283 – Microchip Wireless (MiWi) Media Access Controller - MiMAC

1.5 Advanced Features

Advanced Features

MiWi DE supports advanced features , such as Network Freezer, Enhanced Data Request and Time Synchronization. Some of these features are not documented in the Application Notes. In this section, we describe those new features in detail to help users o understand them. All these features can be enabled in any demo, which is included in the release package.

1.5.1 Network Freezer

Motivation

Occasionally, a wireless network may lose power. After power is restored, in most of the cases, the wireless nodes might form a different network through different joining procedures. This is particularly obvious for MiWi protocol, which uses 16-bit network address in communication. After the power cycle, a wireless node in MiWi network may be assigned with a different network address. As the result, the application layer may have to dedicate more efforts to handle the power cycle scenario. It is important to develop a feature which can release the application layer from handling power cycle.

Solution

Network Freezer feature is developed to solve this problem. It saves critical network information into the Non-Volatile Memory (NVM) and restore them after power cycle. In this way, the application does not need to worry about the power cycle scenario and the network can be restored to the state before the power cycle without any message exchange after the power cycle.

Interface

Network Freezer feature can be enabled by defining ENABLE_NETWORK_FREEZER in configuration file of application layer: ConfigApp.h. In the demo, this feature has been enabled for feature demo.

Network Freezer feature is invoked by calling the MiApp function MiApp_ProtocolInit. The only input boolean parameter bNetworkFreezer indicates if Network Freezer feature should be invoked. When this parameter is TRUE, the network information will be restored from NVM; otherwise, the network information in NVM will be erased and the wireless node start from scratch.

Additional Notes

Network Freezer feature requires NVM to store the critical network information. NVM can be data EPROM in MCU, external EEPORM connected to MCU via SPI, or programming space, if enhanced flash is used in MCU. Choosing the correct form of NVM can be configured in hardware configuration file HardwareProfile.h. The possible options are:

- USE_DATA_EEPROM
- USE_EXTERNAL_EEPROM
- USE_PROGRAMMING_SPACE

For each selection, there are a few minor configurations which can be found in NVM configuration file NVM.h.

Note : These features are only supported for MiWi P2P and Star Protocols.

1.5.2 Enhanced Data Request

Motivation

In a lot of practical network, most of the devices are sleeping devices, which are connected to a few Full Function Devices (FFDs). Usually, the sleeping devices wake up periodically, asking data from the FFDs and report information back to the network.

For most of the applications, it is critical to provide long battery life for the sleeping devices. A majority portion of power is consumed when the sleeping device is active, asking for data and sending data in the duty cycle. Since the power consumption in active mode is around ten thousand times higher than in sleep mode, lower the total active time plays an important role to prolong the battery life.

Solution

According to IEEE 802.15.4 specification, there are typically three message exchanges after a sleeping device wakes up:

1. Data Request command from the sleeping device to FFD, asking for indirect message from FFD.
2. Indirect message from FFD to sleeping device
3. Message from sleeping device to FFD.

In order to save battery power, we can combine message 1 and 3 together, attach message 3 as payload of message 1. In this way, there are only two messages transmitted, saving the time in CSMA-CA detection and put the device into sleep earlier. Our tests show that the total active time could be lowered 20-30%.

Interface

To enable Enhanced Data Request feature, ENABLE_ENHANCED_DATA_REQUEST must be defined in configuration files for protocols: ConfigP2P.h or ConfigMiWi.h. The reason of enabling this feature in protocol layer instead of application layer is that both sleeping devices and FFDs must enable this feature at the same time. While configuration files in application layer is for each individual devices and configuration file in protocol layer is for every devices, it makes sense to enable/disable this feature in the protocol layer to avoid any mismatch in feature enabling.

There is no special function call for the Enhanced Data Request feature. However, the application function call procedure is different with or without Enhanced Data Request feature.

For applications **without** Enhanced Data Request feature, the procedure after MCU waking up is following:

1. Wake up the transceiver by calling *MiApp_TransceiverPowerState* with parameter POWER_STATE_WAKEUP_DR. It will wake up the transceiver as well as asking FFD for indirect message by sending out Data Request command.
2. Send data from sleeping device to FFD.

For application **with** Enhanced Data Request feature, the procedure after MCU waking up is revised, as shown below:

1. Send data from sleeping device to FFD. However, the data is just queue up in the memory. Actual data is not sent yet.
2. Wake up the transceiver by calling *MiApp_TransceiverPowerState* with parameter POWER_STATE_WAKEUP_DR. It will wake up the transceiver and send Data Request command for indirect message. The data that is sent in step 1 will be the payload of Data Request command. At FFD side, it will handle such message by spliting it into Data Request command as well as the individual message.

The implementation of Enhanced Data Request feature can be found in feature demo in the release package.

Additional Notes

Enhanced Data Request feature can be used to transmit unicast message from the sleeping device to the FFD, but broadcast message still depends on normal message delivery method, because broadcast message and Data Request command may have different destination address.

Because Enhanced Data Request is a brand new feature, ZENA sniffer program has not been updated to decode it. However, this limitation would not affect the operation of the stack.

1.5.3 Time Synchronization

Motivation

In a practical wireless network, large number of sleeping node may be connected to a single Full Function Device (FFD). All sleeping devices can sleep for a while and wake up and request indirect messages in a duty cycle. If multiple sleeping devices wake up around the same time and send Data Request to the FFD, some of those packets may collide and get lost, or have to try multiple times before a positive acknowledgement can be received. This scenario also put burden on the FFD to handle multiple requests almost at the same time.

Solution

To solve this kind of problem, requiring each sleeping device to report in a predefined interval is preferred. This approach is somewhat similar to beacon network which is defined in IEEE 802.15.4. However, beacon network in IEEE 802.15.4 only support star topology and require extensive hardware assistant. Our solution is simpler and requires far more less system resources. It is also suitable to be implemented in transceivers that is not IEEE 802.15.4 compliant.

Our solution have the FFD to control the timing of the sleeping device when to wake up and check in next time. As the result, the timing information will be attached to the indirect message response time. As the result, the indirect message has been changed to the following format:

Name	MAC Command	Rough Timing Info	Precise Timing Info	Indirect Message
Length (BYTE)	1	2	2	various
Description	Time Sync Data Packet Command (0x8A) for data indirect message. Time Sync Command Packet Command (0x8B) for command indirect message.	Timeout times on timers. Timers timeout roughly once per 16 seconds.	Timer ticks for precise timing control. One timer tick is configured to be around 244 us.	The indirect message itself.

Interfaces

To enable Time Synchronization feature, ENABLE_TIME_SYNC must be defined in configuration files for protocols: ConfigP2P.h or ConfigMiWi.h. The reason of enabling this feature in protocol layer instead of application layer is that both sleeping devices and FFDs must enable this feature at the same time. While configuration files in application layer is for each individual devices and configuration file in protocol layer is for every devices, it makes sense to enable/disable this feature in the protocol layer to avoid any mismatch in feature enabling.

Additional configuration for Time Synchronization is the total number of slots, TIME_SYNC_SLOTS, supported in the wake up interval of sleeping devices. **As the rule, the TIME_SYNC_SLOTS must be higher or equal to number of sleeping devices that connects to the FFD, so that every sleeping device can have at least one time slot.** Same as ENABLE_TIME_SYNC, TIME_SYNC_SLOTS is defined in protocol configuration files ConfigP2P.h or ConfigMiWi.h for the same reason above. Another configuration is the frequency for the external crystal that connects to the 16-bit asynchronized

counter.

Apart from the configurations in protocol layer, there is no special requirement for function calls on application layer. There are additional hardware requirement for this feature. The details of additional hardware requirement can be found in the next section.

Additional Notes

Time Synchronization feature requires hardware support. The MCU needs a 16-bit timer working as asynchronized counter mode on a 32KHz external crystal. The timer will be able to run when the MCU is in sleep mode and wake up the MCU once it reaches the preset interval.

When Time Synchronization feature is enabled, the minimum time slot depends on the primary oscillator accuracy, 32 KHz external crystal accuracy as well as random time delay caused by CSMA-CA on the environment noise. On standard Microchip demo board, the time slot can be lowered to 100 millisecond.

1.5.4 Star Network Features

Information on MiWi Star Network:

The MiWi P2P protocol stack has been upgraded and a new topology has been added that is called STAR topology. The traditional P2P topology remains as the default, and the new STAR topology is available via a compiler option. In the STAR topology, all the devices join a central PAN Coordinator, and all messages are routed through the PAN Coordinator, for a maximum of 2 hops. In the STAR topology the network range is extended up to 2 radio-range in distance,.

1)Link Status Command:

A Link Status command is added to the stack. Its purpose is to provide of means for the PAN Coordinator to monitor the status of the network. By default, each End Node, sends a Link Status message to the PAN Coordinator every 15 seconds, indicating its alive and communicating on the network. The frequency of the Link Status message is a compiler option,

2)Network Leave Command:

A Network Leave Command has been added to the stack. Devices voluntarily or involuntarily leaving the network shall broadcast this message prior to leaving the network. Other network devices receiving the Leave Command use it to update their Connection Table.

3)Periodic Broadcast of Connection Table:

PAN Coordinator periodically broadcasts the Connection Table to all nodes in the network. This feature is added so that all FFD is made aware of all the other devices that have joined or have left the network.

4)MiAPP API Expansion:

The MiApp_UnicastStar API has been added to the Application Layer suite of interface functions. Its purpose is to send a message from source End Node to Destination End Node in a STAR network.

1.6 Demos

1.6.1 Required Hardware

To run this project, you will need two wireless nodes. Each of the nodes can be any of setups of listed hardware.

P2P -> To run a P2P demo , minimal of 2 demo kits are required.

Star -> To run a Star demo , minimal of 3 demo kits are required.

Mesh -> To run a Mesh demo , minimal of 5 demo kits are required.

1.6.1.1 Hardware Sets

Hardware Setups

Hardware Set 1:

- Demo Board:
  - PICDEM Z 2.4 Demo Kit (DM163027-4 OR DM163027-5) OR
  - PICDEM Z Mother Board (AC163027-1)
- RF Board:
  - MRF24J40
    - PICDEM Z MRF24J40 2.4GHz Daughter Card (AC163027-4) OR
    - MRF24J40MA PICDEM™ Z 2.4GHz RF Card (AC163028)

Hardware Set 2:

- Demo Board:
  - PIC18 Explorer with PIC18F87J11 PIM (DM183032)
- RF Board:
  - MRF24J40
    - MRF24J40MA PICtail (AC164134-1)
  - MRF49XA
    - MRF49XA PICtail 434MHz (AC164137-1) OR
    - MRF49XA PICtail 868/915MHz (AC164137-2)
  - MRF89XA
    - MRF89XA PICtail 868MHz OR
    - MRF89XA PICtail 915MHz
  - MRF24XA
    - MRF24XA PICtail 2.4GHz

Hardware Set 3:

- Demo Board:

- Explorer 16 (DM240001)
- PIC24FJ128GA010 Plug-In-Module (PIM) (MA240011)

- RF Borad
  - MRF24J40
    - PICDEM Z MRF24J40 2.4GHz Daughter Card (AC163027-4) OR
    - MRF24J40MA PICtail™ Plus 2.4GHz RF Card (AC164134) OR
    - MRF24J40MA PICtail (AC164134-1)
  - MRF49XA
    - MRF49XA PICtail 434MHz (AC164137-1) OR
    - MRF49XA PICtail 868/915MHz (AC164137-2)
  - MRF89XA
    - MRF89XA PICtail 868MHz OR
    - MRF89XA PICtail 915MHz
  - MRF24XA
    - MRF24XA PICtail 2.4GHz

Hardware Set 4:

- Demo Board:
  - 8-bit Wireless Development Kit
- RF Board:
  - MRF24J40
    - MRF24J40MA PICtail (AC164134-1)
  - MRF49XA
    - MRF49XA PICtail 434MHz (AC164137-1) OR
    - MRF49XA PICtail 868/915MHz (AC164137-2)
  - MRF89XA
    - MRF89XA PICtail 868MHz OR
    - MRF89XA PICtail 915MHz
  - MRF24XA
    - MRF24XA PICtail 2.4GHz

Hardware Set 5:

- Demo Board:
  - MiWi Demo Kit
- RF Board:
  - MRF24J40
    - MiWi Demo Kit 2.4 Ghz MRF24J40 (DM182016-1)
  - MRF89XA
    - MiWi Demo Kit 915 Mhz MRF89XA (DM182016-3) OR
    - MiWi Demo Kit 868 Mhz MRF89XA (DM182016-2)

1.6.2 Configuring the Hardware

This section describes how to set up the various configurations of hardware to run this demo:

Configuration 1: PICDEM Z Demo Kit

Configuration 2: PIC18 Explorer demo board

Configuration 3: Explorer 16 demo board, with PIC24FJ128GA010 or PIC32MX360F512L PIM

Configuration 4: 8-bit Wireless Development Kit

Configuration 5: MiWi Demo Kit

Serial Port Support on Configuration 1 , 2, 3, 4 only

Terminal Emulator Software :

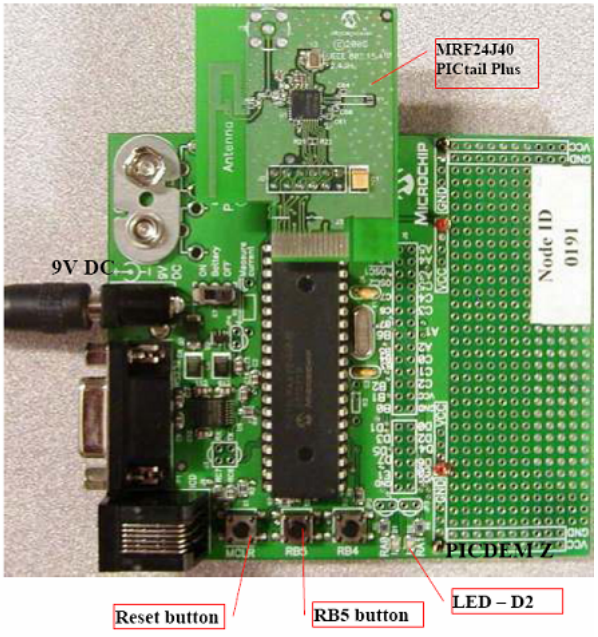
Windows Machine : Tera Term

MAC : Cool Term

1.6.2.1 PICDEM Z

Configuration 1: PICDEM Z

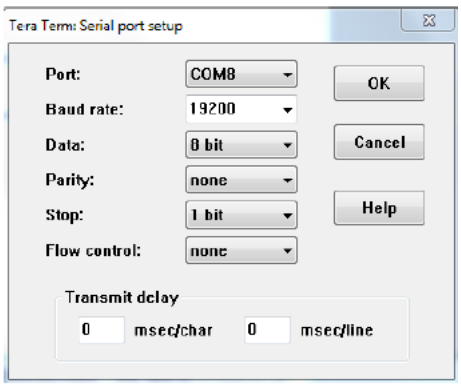
Connect the MRF24J40 2.4GHz RF Card tot he PICDEM Z demo board as shown in the picture



Before running the demos, it is highly recommended to connect a serial cable to both demo boards and connect the ZENA sniffer hardware to monitor the operating of the network.

PICDEM Z demo board only support Microchip MRF24J40 transceiver, which is compliant with IEEE 802.15.4 specification.

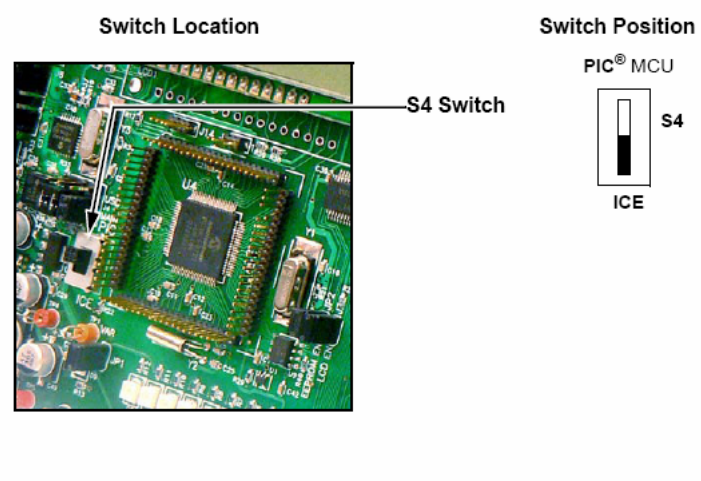
Serial Port Setup on Terminal:



1.6.2.2 PIC18 Explorer

Configuration 2: PIC18 Explorer

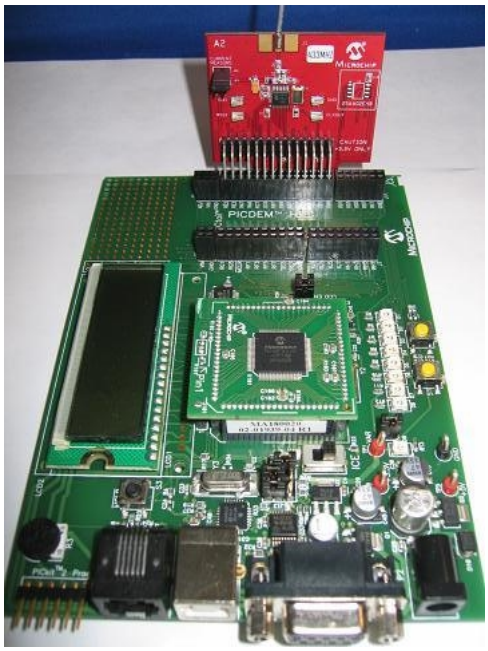
1) Set the S4 switch on PIC18 Explorer at the position of ICE.



2) Before connecting the PIM to the PIC18 Explorer board, remove all attached cables. Connect the PIM to the PIC18 Explorer board. Be careful when connecting the boards to insure that no pins are bent or damaged during the process. Also ensure that the PIM is not shifted in any direction and that all of the headers are properly aligned.

3) Connect the either the MRF24J40, MRF49XA or MRF89XA RF board to the PICTail connector. Be aware that the transceiver chip should face the PIM and the first pin should be plugged into the hole labeled "RE2".

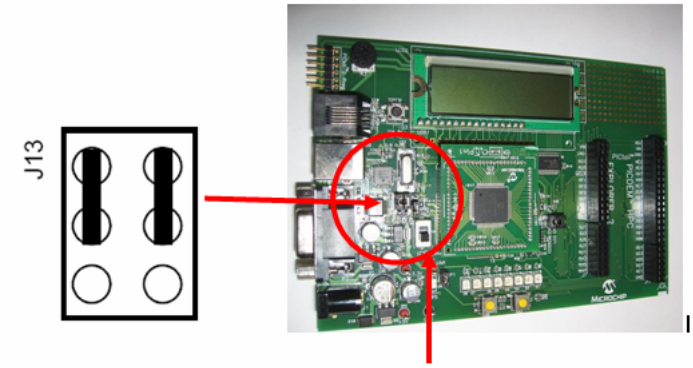
The configured hardware setup for PIC18 Explorer board should look like following picture.



PIC18 Explorer support both RS232 serial port and USB to connect to the PC for monitoring. PIC18 Explorer by default is configured to use the RS232 serial port to communicate with PC. Following steps setup the PIC18 Explorer to use USB port.

1. Hardware Setup

Configure the Explorer PIC18 demo board to use USB connection by setting jumper J13 according to the following diagram

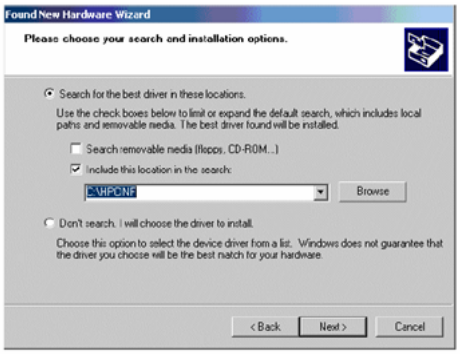


Make sure toggle switch is in the DOWN – ICE position. This switch activates the PIC18 on the PIM. DO NOT remove the PIM or the board Vdd will be 5V, which may damage the RF module.

2. USB Driver Install

- Connect the PIC18 Explorer demo board to the PC using a USB cable.
- Power up PIC18 Explorer demo board, following pop up window will appear

- Select "Install from a list or specific location" option and click "Next". Following pop up window appear

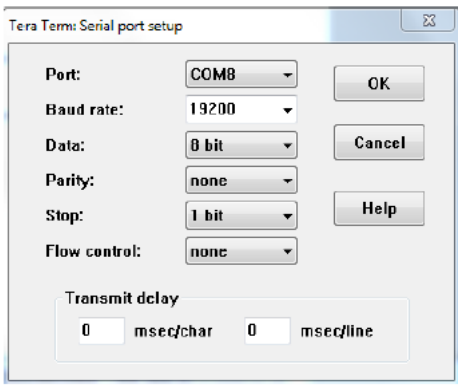


- Select the check box "Include this location in the search", in the text box, browse to "<Install Directory>\PC Software" folder. This is the location of the "mchpcdc.inf" driver.
- Click "Next". There may be warning from Windows operating system about installing a driver without digital signature. Please ignore that warning and continue. After the driver is installed properly, the following screen will appear

- Click "Finish". USB port is ready to be used.

3. Open Tera Term Terminal

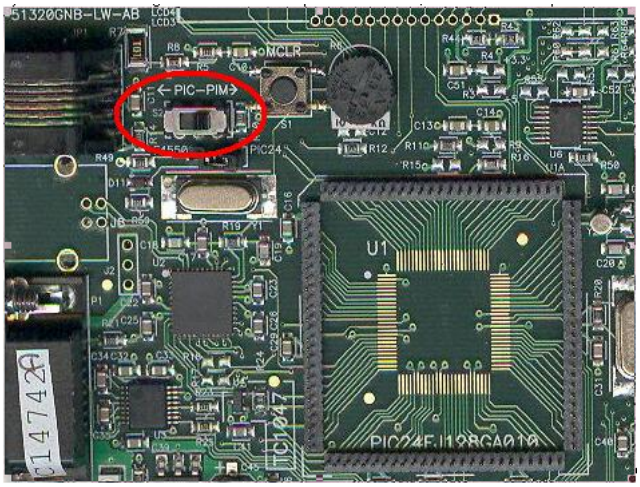
Once the RS232 serial or USB cable is connected between the demo board and PC, launch a hyper terminal to display the information from the demo board. The hyper terminal configuration is baud rate 19200, Data bit 8, Parity None, Stop bits 1 and Flow control None, as shown below.



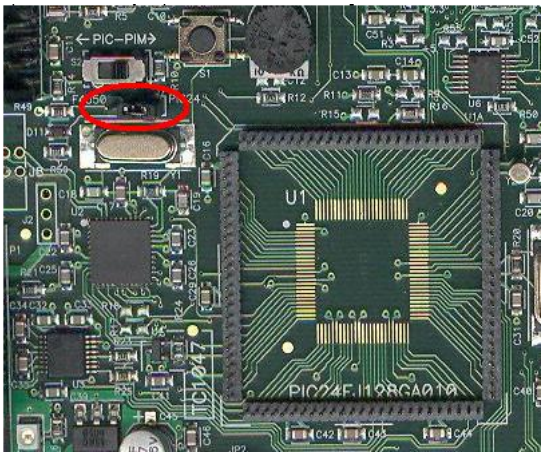
1.6.2.3 Explorer 16

Configuration 3: Explorer 16

1) Before attaching the PIM to the Explorer 16 board, ensure that the processor selector switch (S2) is in the “PIM” position as seen in the image below:

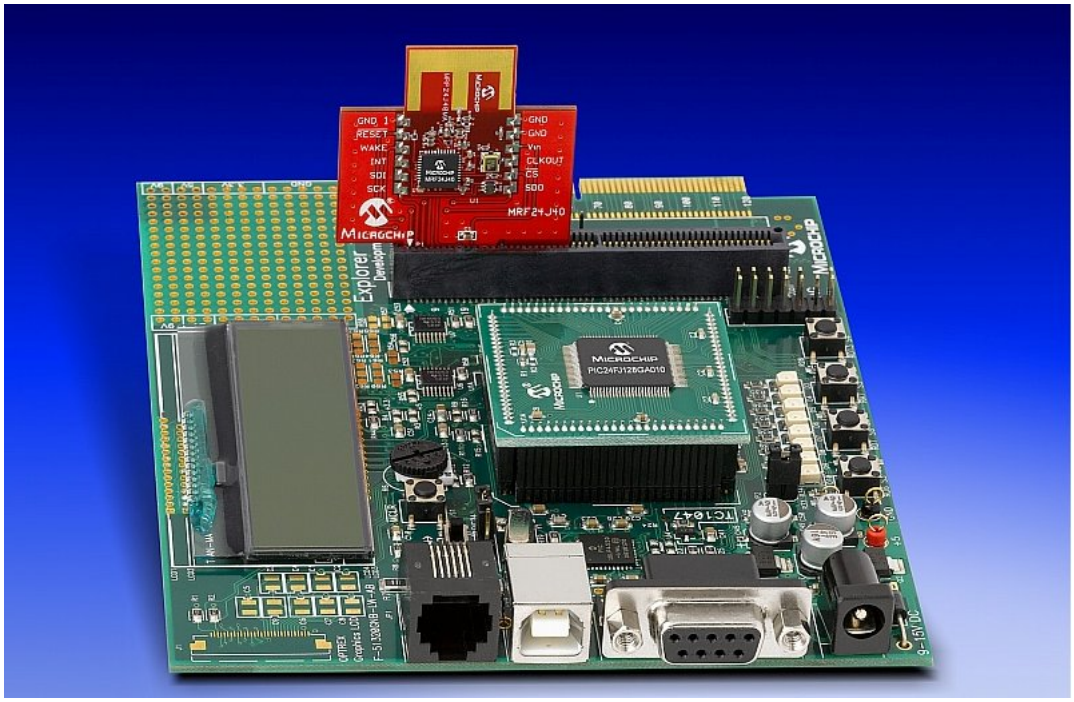


2) Short the J7 jumper to the "PIC24" setting

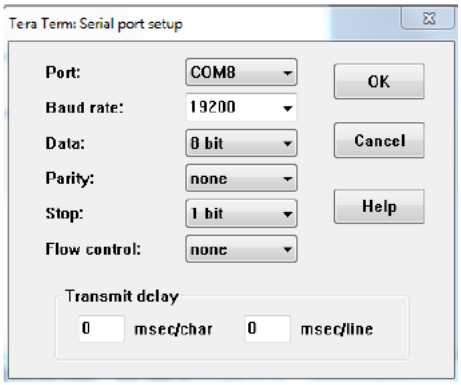


3) Before connecting the PIM to the Explorer 16 board, remove all attached cables. Connect the PIM to the Explorer 16 board. Be careful when connecting the boards to insure that no pins are bent or damaged during the process. Also ensure that the PIM is not shifted in any direction and that all of the headers are properly aligned.

4) Connect the RF board for MRF24J40 or MRF49XA to the first slot of edge card connector, as shown in the following picture.



5) Before running the demos, it is highly recommended to connect a serial cable to both demo boards and connect the ZENA sniffer hardware to monitor the operating of the network. Once the serial cable is connected between the demo board and PC, launch a terminal emulator to display the information from the demo board. The terminal configuration is baud rate 19200, Data bit 8, Parity None, Stop bits 1 and Flow control None, as shown below.



1.6.2.4 8-bit Wireless Demo Board

Configuration 4: 8-bit Wireless Demo Board

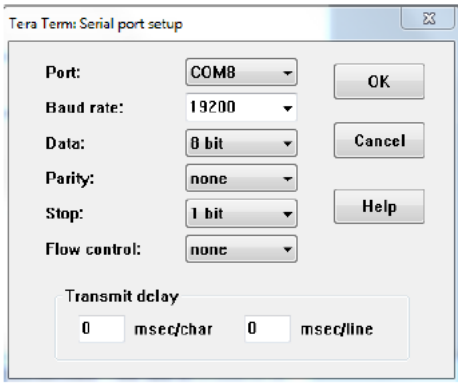
The 8-bit Wireless Development Kit (Part Number : DM182015-1) allows developers to evaluate and experiment with the 2.4 GHz RF solutions from Microchip. The Wireless Development Kit provides two RF hardware nodes which can be used to create a simple two-node wireless network. Figure 2 shows two set of 8-bit WDK development boards along with the

MRF24J40 PICtail boards which comes as part of the 8-bit Wireless Development Kit. More nodes can be added to the network by using 8-Bit Wireless Development Kit or individual boards. The DM182015-1 comes with MRF24J40MA based PICtail boards to support 2.4 GHz communication. The kit includes all hardware needed to rapidly prototype wireless applications. For more information on 8-bit Wireless Development Kit refer 8-bit Wireless Development Kit User's Guide from

www.microchip.com\DM182015-1

1) Connect the 8-bit Wireless Demo Board in the following way that is shown in the picture. Be aware that the jumper "JP1" on the LCD daughter board should be removed to work with RS232 daughter board.

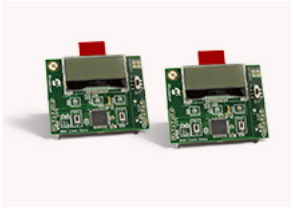
2) Before running the demos, it is highly recommended to connect a serial cable to both demo boards and connect the ZENA sniffer hardware to monitor the operating of the network. Once the serial cable is connected between the demo board and PC, launch a terminal emulator to display the information from the demo board. The terminal configuration is baud rate 19200, Data bit 8, Parity None, Stop bits 1 and Flow control None, as shown below.



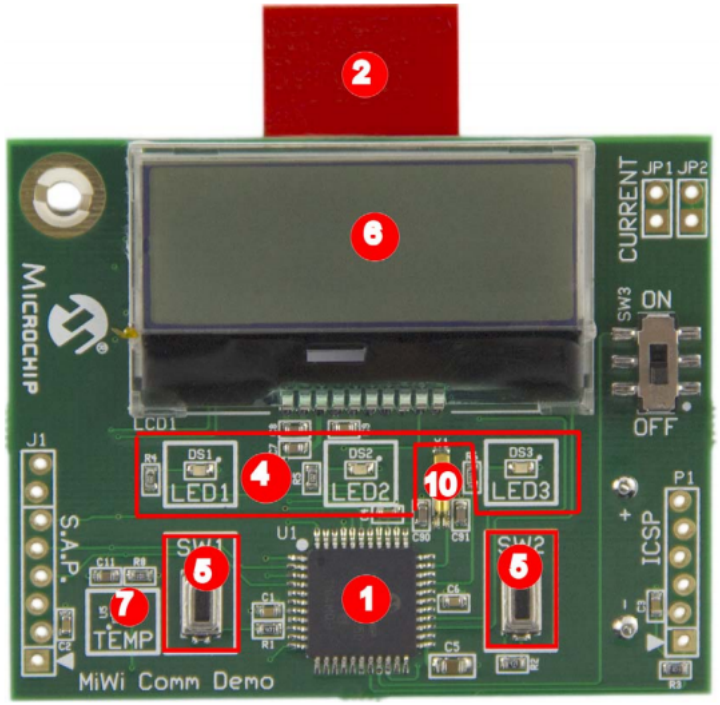
1.6.2.5 MiWi Demo Kit

Configuration 5:The MiWi Demo Kit (Part Number: DM182016-1) enables developers to evaluate and experiment with 2.4 GHz RF solutions from Microchip. The MiWi Demo Kit contains two hardware nodes (MiWi Demo Boards integrated with MRF24J40MA modules) which can be used to create a simple two node MiWi wireless network. Figure 1 shows two MiWi Comm Boards which comes as part of the MiWi Demo Kit. More hardware nodes to the same network can be added using MiWi Demo Kits or individual boards. The kit includes all hardware needed to rapidly prototype wireless applications. For more information on MiWi Demo Boards refer MiWi Demo Kit User's Guide from www.microchip.com\DM182016-1

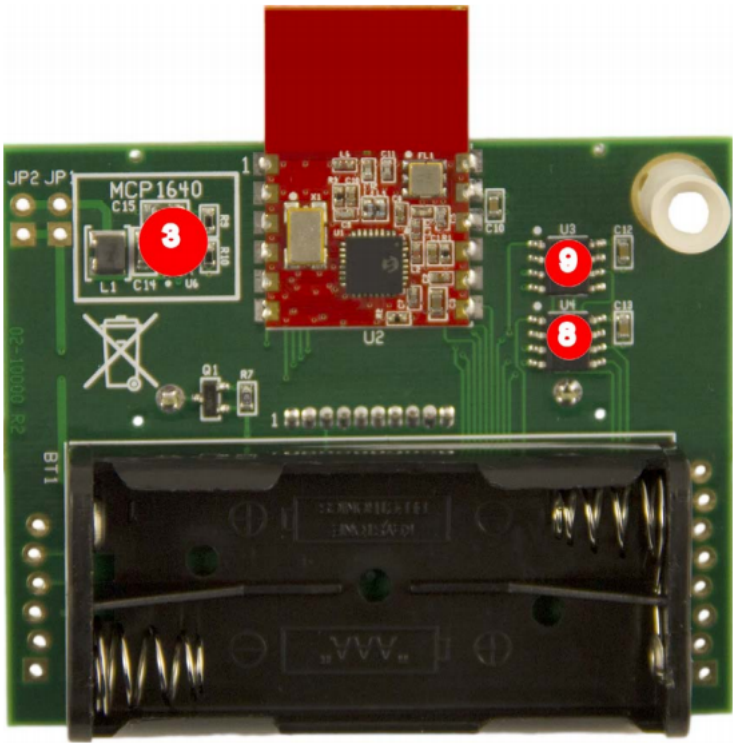
MiWi Demo kit:



Frontal View of MiWi Demo Kit:



Rear View of MiWi Demo Kit:



Key Features of MiWi Demo Board:

1. PIC18F46J50 8-bit XLP microcontroller

2. MRF24J40MA RF Transceiver Module or MRF89XAM8A RF Transceiver

Module or MRF89XAM9A RF Transceiver Module

3. +3.3V Boost Regulator (MCP1640)

4. Three status indicator LEDs (Red, Yellow and Green)

5. Two push-button switches (SW1 and SW2) for user input

6. 2 X 16 LCD Character Display

7. MCP9700 Temperature Sensor

8. 2K SPI EEPROM with a unique MAC Address

9. 1 Mbit SPI Serial Flash

10. 32 KHz Crystal for Sleep Mode

Programmers Compatible to MiWi Demo Kit:

PG164130 - PICkit 3 In-Circuit Debugger

DV164035 - MPLAB ICD 3 In-Circuit Debugger

DV244005 - MPLAB REAL ICE PROBE KIT

Power Supply:

Two AAA batteries are required with each MiWi Demo Kit as power supply.

Programming and Debugging using ICSP pin:

Connect the available Microchip programmer/debugger to the ICSP port/interface (P1 in MiWi Demo Kit and of the development board.

1.6.3 Firmware

Firmware

To run this project, you will need to load the corresponding firmware into the devices. There are two methods available for loading the demos: Precompiled demos and source code projects.

1.6.3.1 Precompiled HEX Files

Precompiled HEX Files

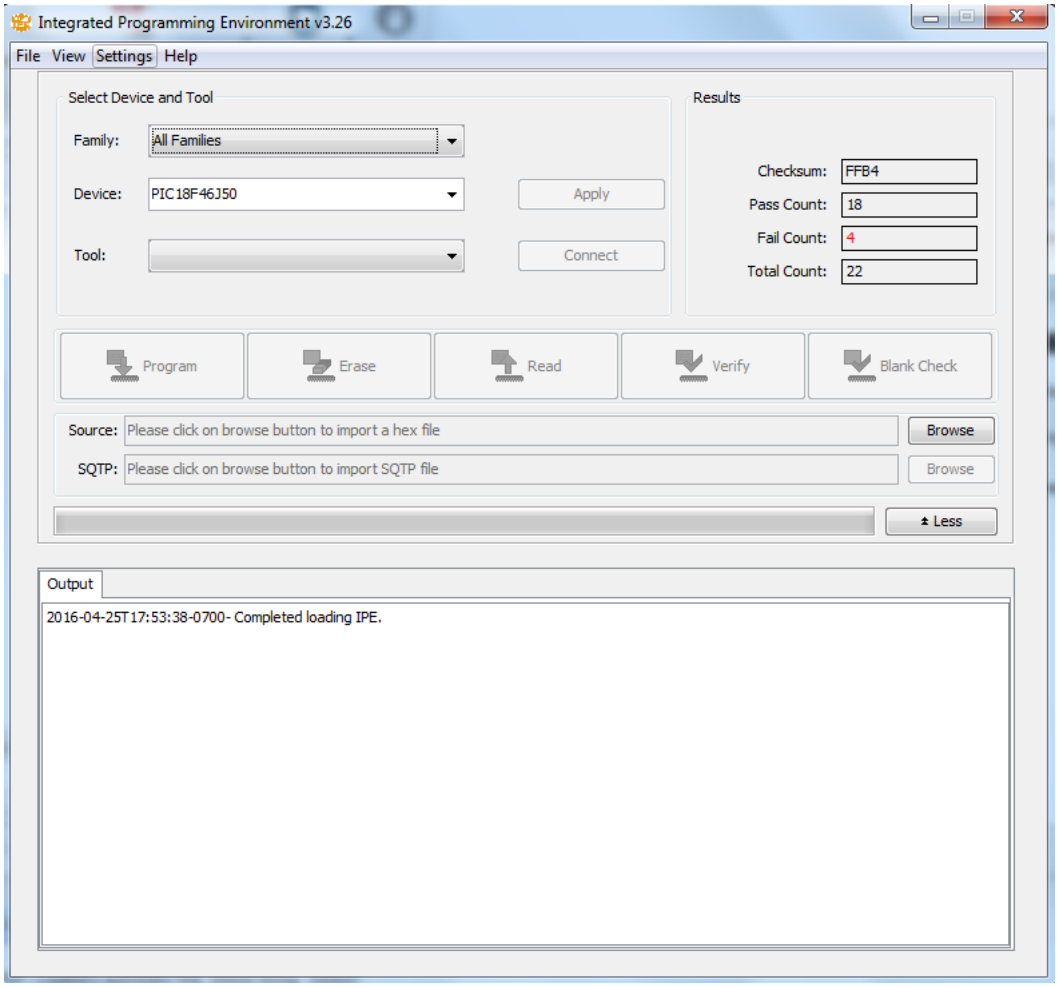
Precompiled Demos are available in the “<Install Directory>\apps\miwi\miwi_p2p\simple_example_p2p\firmware\hex\hardware_configuration\.hex”. A hex file is provided for each hardware configuration and each available RF transceiver under the two directories.Import the corresponding hex file and then program the hex to the demo board.

Process for Programming different Hex files remains the same for all MiWi Demo Projects covering P2P , STAR and MESH networks.

Note : Directory for Hex files will change based on the project chosen.

Integrated Programming Environment:

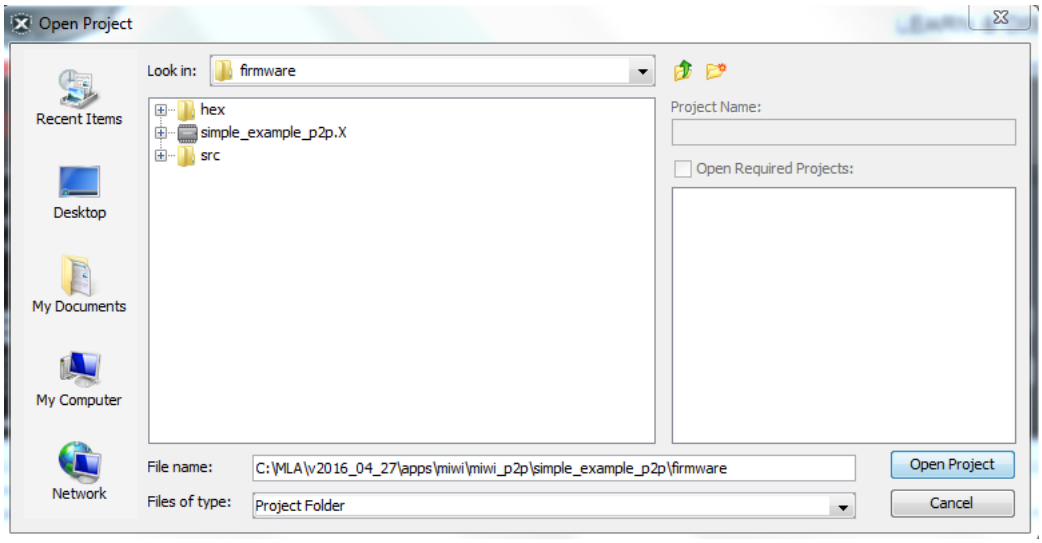
MPLAB IPE v3.xx

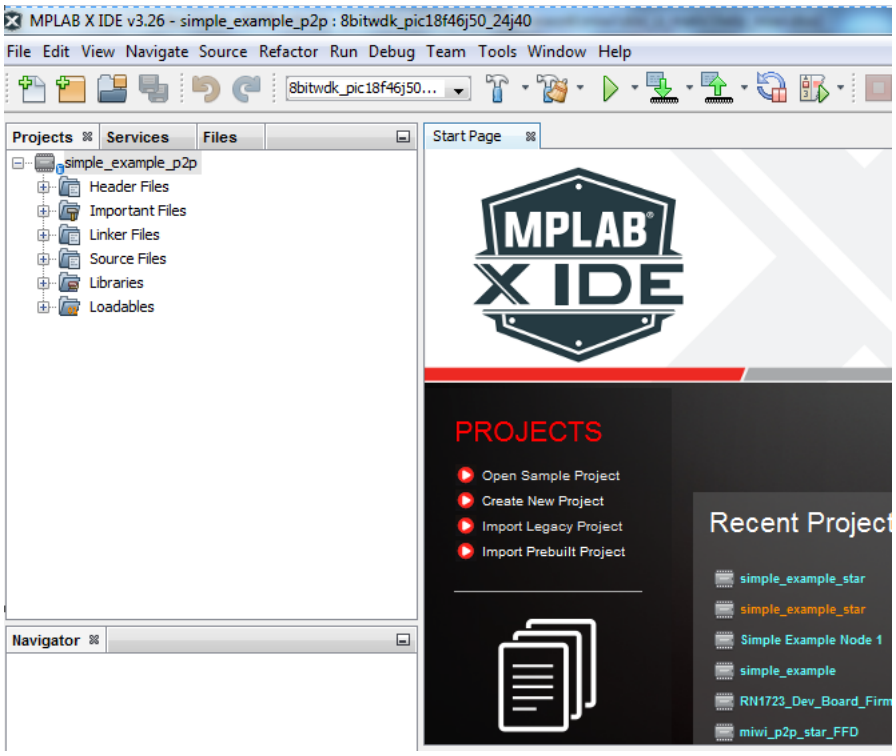


1.6.3.2 Demo Source Code Project

Demo Source Code Modification

The source code for this demo is available in the “<Install Directory>\apps\miwi\miwi_p2p\simple_example_p2p\firmware\project.x” directory. In these directories, you will find all of the application level source and header files as well as project files for each of the hardware platforms. To open the project, select “Project” from the main menu of MPLAB IDE and then “Open…” option. A window will pop out and request the project file. The snap-shot of the opening a mplabx project can be found below:





To run the demo, both nodes must be configured to use the same RF transceiver with the same settings and the same wireless protocol. However, both nodes do not have to be the same demo board.

To compile the demo, following working environment must be established:

• PICDEM Z, PIC18 Explorer , 8 bit wireless Dev Kit and MiWi Demo Kit and RFD Board Configuration: XC8 v 1.35 or higher

• Explorer 16 and RFD Board Configuration: xc16 v 1.26 or higher.

Compile and program the demo code into the hardware platform. For more help on how to compile and program projects, please refer to the MPLAB® IDE help available through the help menu of MPLAB IDE (Help->Topics…->MPLAB IDE).

By default, there are 4 projects ,

1) miwi_demo_kit (miwi mesh)

Supported Dev Boards: MiWi Demo Kit

2) simple_example_p2p (miwi_p2p)

Supported Dev Boards: PICDEM Z , PIC18 Explorer , Explorer 16 , 8-bit Wireless Dev Kit , MiWi Demo Kit

3) simple_example_star (miwi_star)

Supported Dev Boards: PICDEM Z , PIC18 Explorer , Explorer 16 , 8-bit Wireless Dev Kit , MiWi Demo Kit

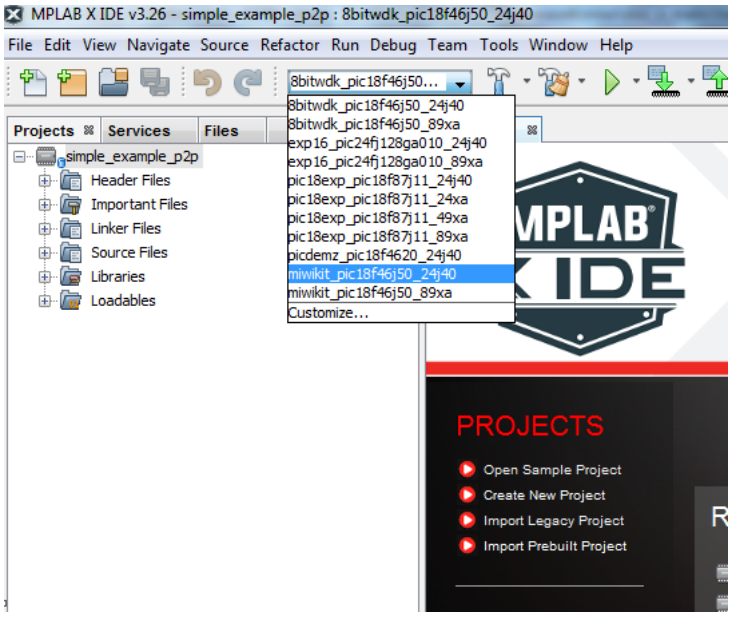
4) chat_demo (miwi_p2p)

Supported Dev Boards : Supported Dev Boards: 8-bit Wireless Dev Kit

Each demo can be configured to work with a different combination of dev kit and tranceiver. With minor configuration modification, user can compile and run the demo on any Microchip RF transceivers, any Microchip proprietary wireless protocols on any supported standard demo boards. This section demonstrates how to migrate the demo among all supported options.

Example Project : simple_example_p2p

Change in Hardware configuration



1.6.3.2.1 MiWi P2P

1.6.3.2.1.1 PICDEM Z Demo Board for MiWi P2P

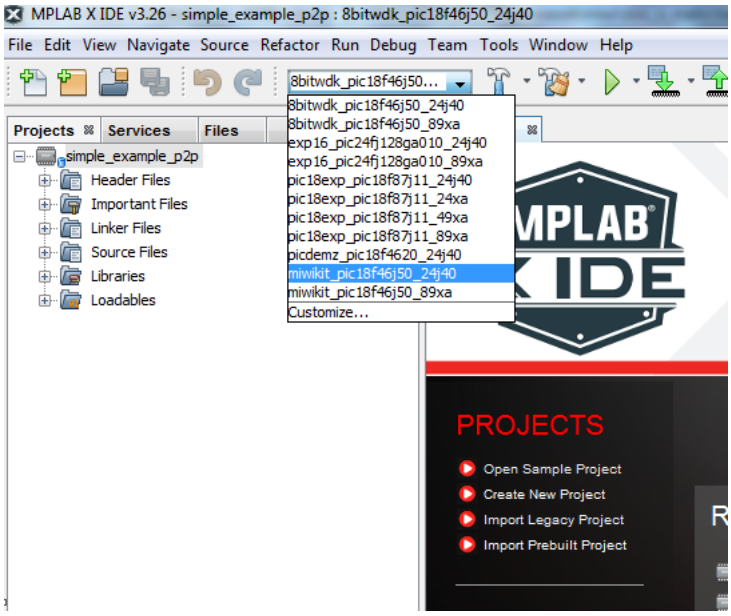
PICDEM Z Demo Board for MiWi™ P2P

PICDEM Z Demo board use PIC18F4620 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F4620" as the device and then click "OK".

Open the simple_example_p2p project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : picdemz_pic18f4620_24j40 from the list of hardware configurations displayed.



Compile and program the MCU on the picdemz board

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi P2P protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_p2p/headers/system_config/picdemz_pic18f4620_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.1.2 PIC18 Explorer Demo Board for MiWi P2P

PIC18 Explorer Demo Board for MiWi™ P2P

By default, PIC18 Explorer Demo board use PIC18F87J11 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F87J11" as the device and then click "OK". If you use a different PIM other than the default PIC18F87J11, please select the corresponding MCU accordingly.

Open the simple_example_p2p project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : pic18exp_pic18f87j11_xxxx from the list of hardware configurations displayed.

xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for PIC 18 Explorer Demo Board are

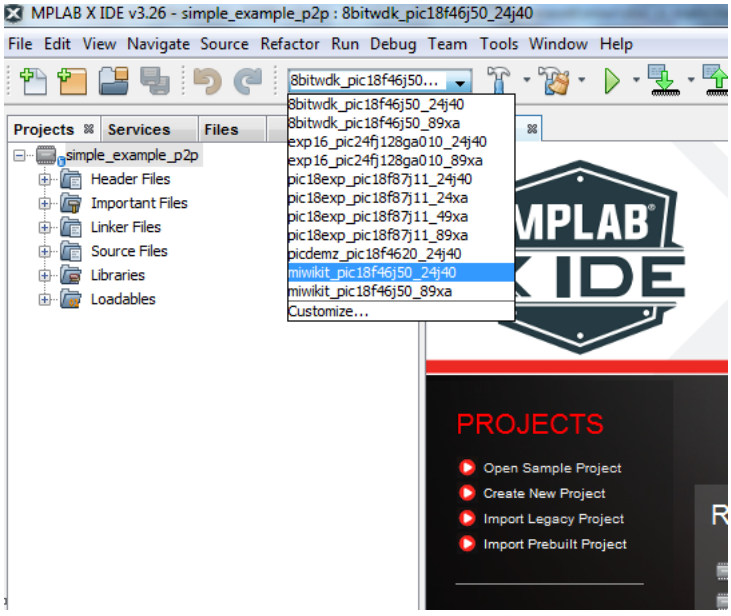
MRF24J40

MRF89XA

MRF49XA

MRF24XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the pic18 explorer board

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi P2P protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_p2p/headers/system_config/pic18exp_pic18f87j11_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.1.3 Explorer 16 Demo Board for MiWi P2P

Explorer 16 Demo Board for MiWi™ P2P

By default, PIC18 Explorer Demo board use PIC24FJ128GA010 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC24FJ128GA010 " as the device and then click "OK". If you use a different PIM other than the default PIC24FJ128GA010 , please select the corresponding MCU accordingly.

Open the simple_example_p2p project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : exp16_ pic24fj128ga010 _xxxx from the list of hardware configurations displayed.

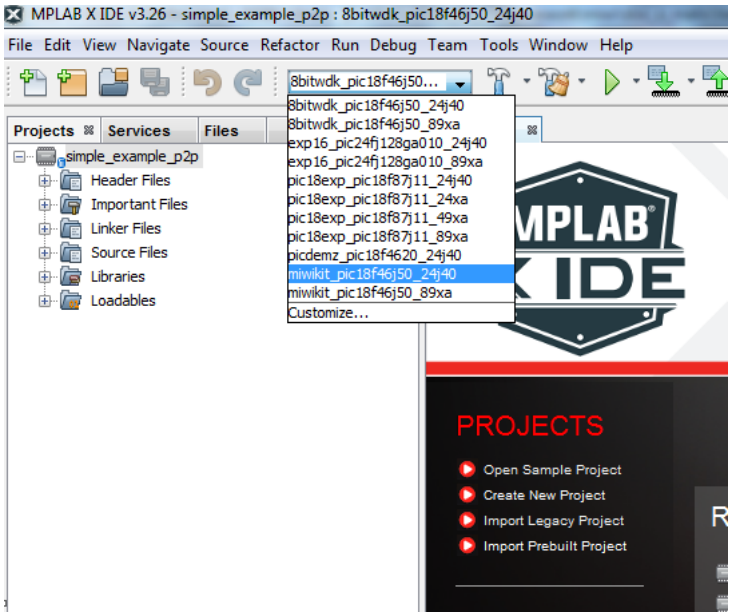
xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for Explorer 16 Demo Board are

MRF24J40

MRF89XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the explorer 16 board

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi P2P protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_p2p/headers/system_config/exp16_ pic24fj128ga010 _24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.1.4 8-bit Wireless Demo Board for MiWi P2P

8-bit Wireless Demo Board for MiWi™ P2P

8-bit Wireless Demo board use PIC18F46J50 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F46J50" as the device and then click "OK".

Open the simple_example_p2p project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : 8bitwdk_pic18f46j50_xxxx from the list of hardware configurations displayed.

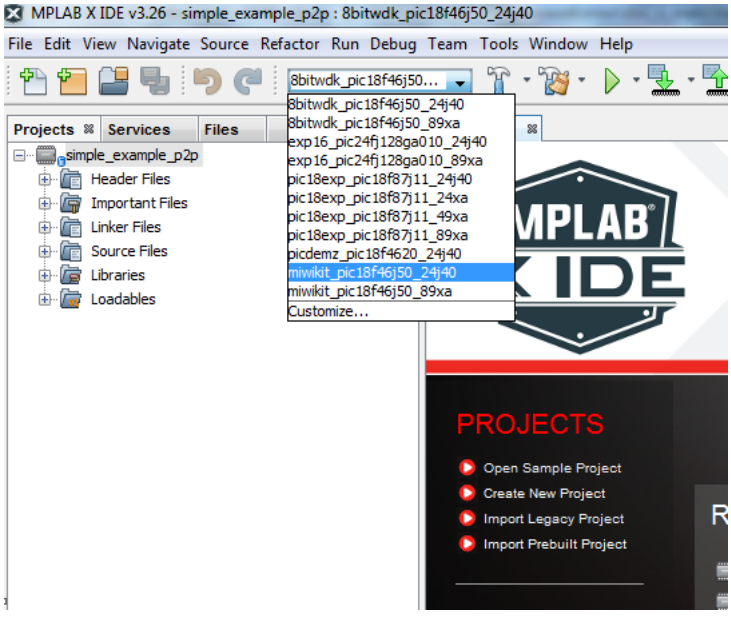
xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for Explorer 16 Demo Board are

MRF24J40

MRF89XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the 8 bit wireless demo board.

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi P2P protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_p2p/headers/system_config/8bitwdk_pic18f46j50_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

Note : Two Demo Projects are supported for MiWi P2P protocol simple_example_p2p and chat_demo

1.6.3.2.1.5 MiWi Demo Kit for MiWi P2P

MiWi Demo Kit for MiWi™ P2P

MiWi Demo kit uses PIC18F46J50 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F46J50" as the device and then click "OK".

Open the simple_example_p2p project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : miwikit_pic18f46j50_xxxx from the list of hardware configurations displayed.

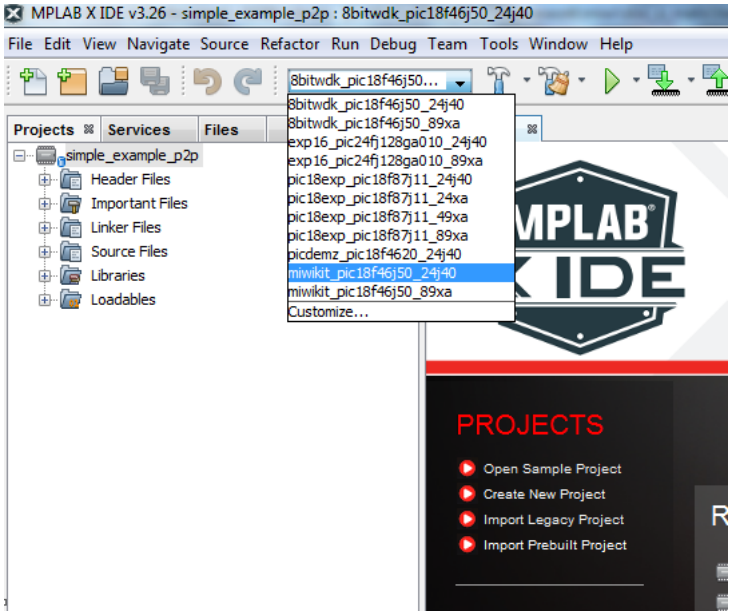
xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for Explorer 16 Demo Board are

MRF24J40

MRF89XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the miwi kit demo board.

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi P2P protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_p2p/headers/system_config/miwikit_pic18f46j50_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.2 MiWi Star

1.6.3.2.2.1 PICDEM Z Demo Board for MiWi Star

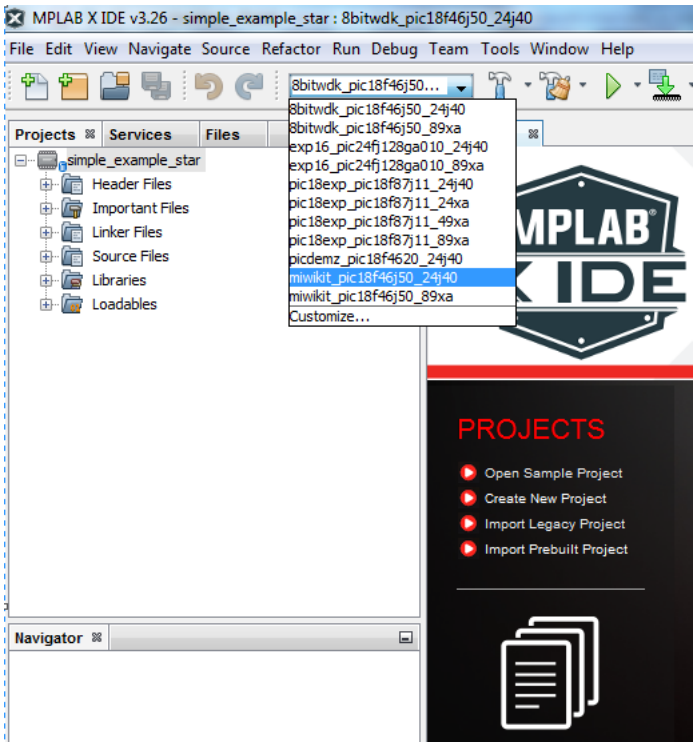
PICDEM Z Demo Board for MiWi™ Star

PICDEM Z Demo board use PIC18F4620 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F4620" as the device and then click "OK".

Open the simple_example_star project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : picdemz_pic18f4620_24j40 from the list of hardware configurations displayed.



Compile and program the MCU on the picdemz board

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi Star protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Note : The config files for MiWi P2P and Star Protocol are the same as MiWi Star is a extension of P2P protocol.

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_star/headers/system_config/picdemz_pic18f4620_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.2.2 PIC18 Explorer Demo Board for MiWi Star

PIC18 Explorer Demo Board for MiWi™ Star

By default, PIC18 Explorer Demo board use PIC18F87J11 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F87J11" as the device and then click "OK". If you use a different PIM other than the default PIC18F87J11, please select the corresponding MCU accordingly.

Open the simple_example_star project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : pic18exp_pic18f87j11_xxxx from the list of hardware configurations displayed.

xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for PIC 18 Explorer Demo Board are

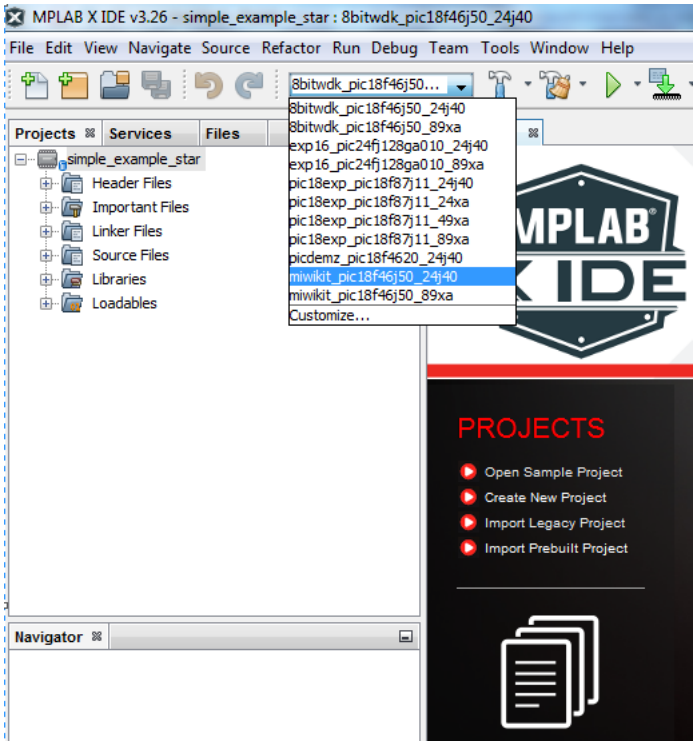
MRF24J40

MRF89XA

MRF49XA

MRF24XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the pic18 explorer board

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi Star protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Note : The config files for MiWi P2P and Star Protocol are the same as MiWi Star is a extension of P2P protocol.

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_star/headers/system_config/pic18exp_pic18f87j11_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.2.3 8-bit Wireless Demo Board for MiWi Star

8-bit Wireless Demo Board for MiWi™ Star

8-bit Wireless Demo board use PIC18F46J50 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F46J50" as the device and then click "OK".

Open the simple_example_star project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : 8bitwdk_pic18f46j50_xxxx from the list of hardware configurations displayed.

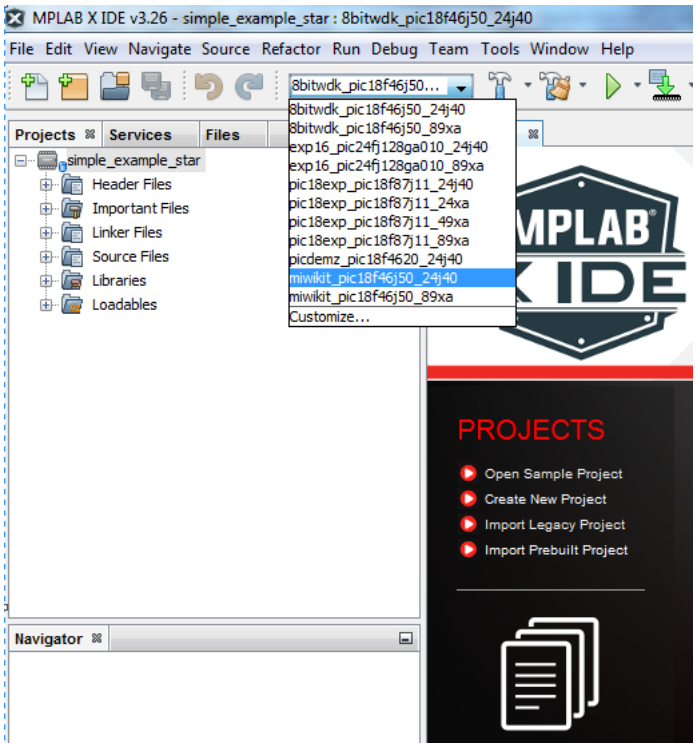
xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for Explorer 16 Demo Board are

MRF24J40

MRF89XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the 8 bit wireless demo board.

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi Star protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Note : The config files for MiWi P2P and Star Protocol are the same as MiWi Star is a extension of P2P protocol.

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_star/headers/system_config/8bitwdk_pic18f46j50_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.2.4 Explorer 16 Demo Board for MiWi Star

Explorer 16 Demo Board for MiWi™ Star

By default, PIC18 Explorer Demo board use PIC24FJ128GA010 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC24FJ128GA010 " as the device and then click "OK". If you use a different PIM other than the default PIC24FJ128GA010 , please select the corresponding MCU accordingly.

Open the simple_example_star project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : exp16_ pic24fj128ga010 _xxxx from the list of hardware configurations displayed.

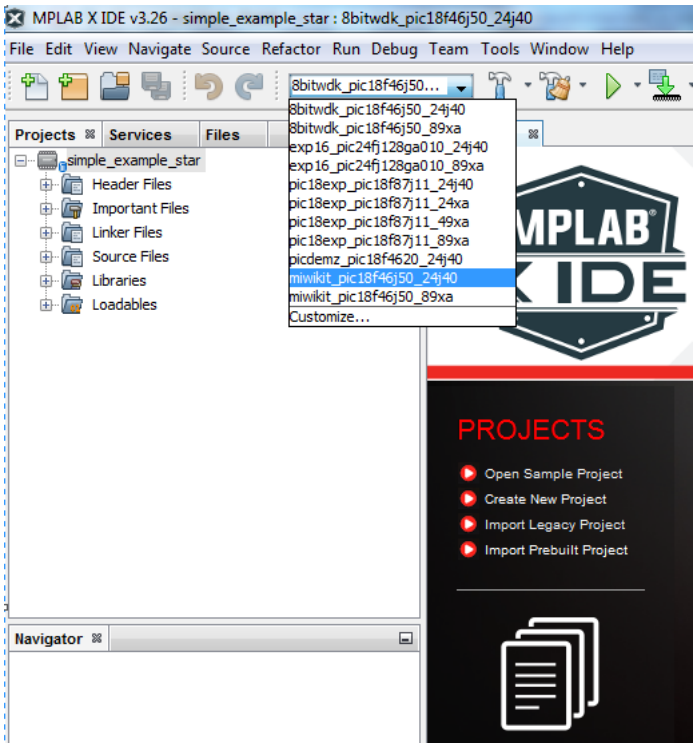
xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for Explorer 16 Demo Board are

MRF24J40

MRF89XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the explorer 16 board

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi Star protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Note : The config files for MiWi P2P and Star Protocol are the same as MiWi Star is a extension of P2P protocol.

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_star/headers/system_config/exp16_ pic24fj128ga010 _24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

miwi_config_p2p.h

console.h

1.6.3.2.2.5 MiWi Demo Kit For MiWi Star

MiWi Demo Kit for MiWi™ Star

MiWi Demo kit uses PIC18F46J50 MCU as the host controller. Select "Configure" from the MPLAB menu and then choose "Select Device...". From the pop up menu, choose "PIC18F46J50" as the device and then click "OK".

Open the simple_example_star project.

Change the Hardware Configuration as shown in the figure below.

Choose the option : miwikit_pic18f46j50_xxxx from the list of hardware configurations displayed.

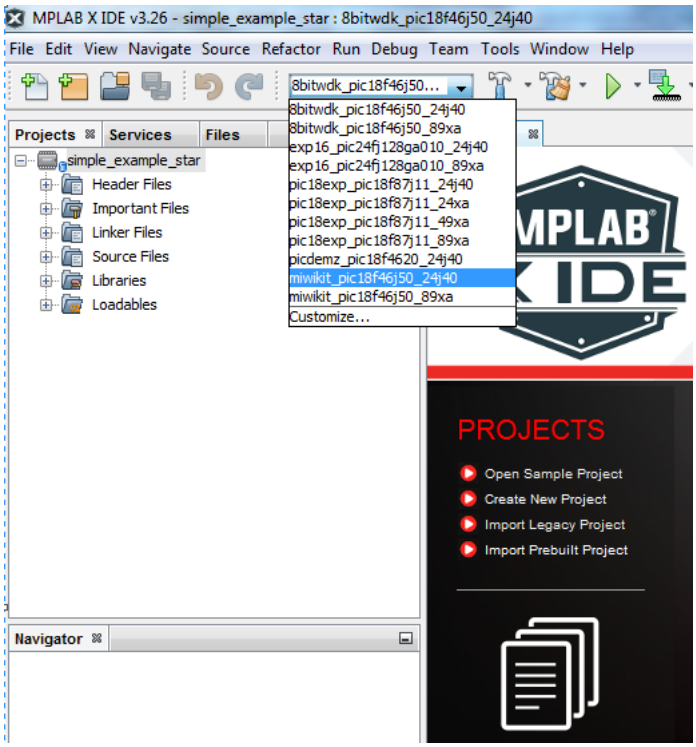
xxxx corresponds to which RF tranceiver , developer wants to choose from.

RF transceivers supported for Explorer 16 Demo Board are

MRF24J40

MRF89XA

Based on RF tranceiver choose the right hardware configuration:



Compile and program the MCU on the miwi kit demo board.

More information on how to use MPLABx IDE is available in MPLABX IDE user guide

Please follow the Running Demos section to understand the working of Demo.

Configuring the MiWi Star protocol to change the different protocol features :

miwi_config.h

miwi_config_p2p.h

config_24j40.h

Note : The config files for MiWi P2P and Star Protocol are the same as MiWi Star is a extension of P2P protocol.

Configuring the Hardware For the Project :

system_config.h

Follow the path to change the above features or to configure hardware (SPI , UART , TIMER , Radio Config etc)

simple_example_p2p/headers/system_config/miwikit_pic18f46j50_24j40/xxxxxxxx.h

xxxxxxxx.h can be any of the below header files

eeprom.h

spi.h

config_24j40.h

lcd.h

delay.h

timer.h

miwi_config.h

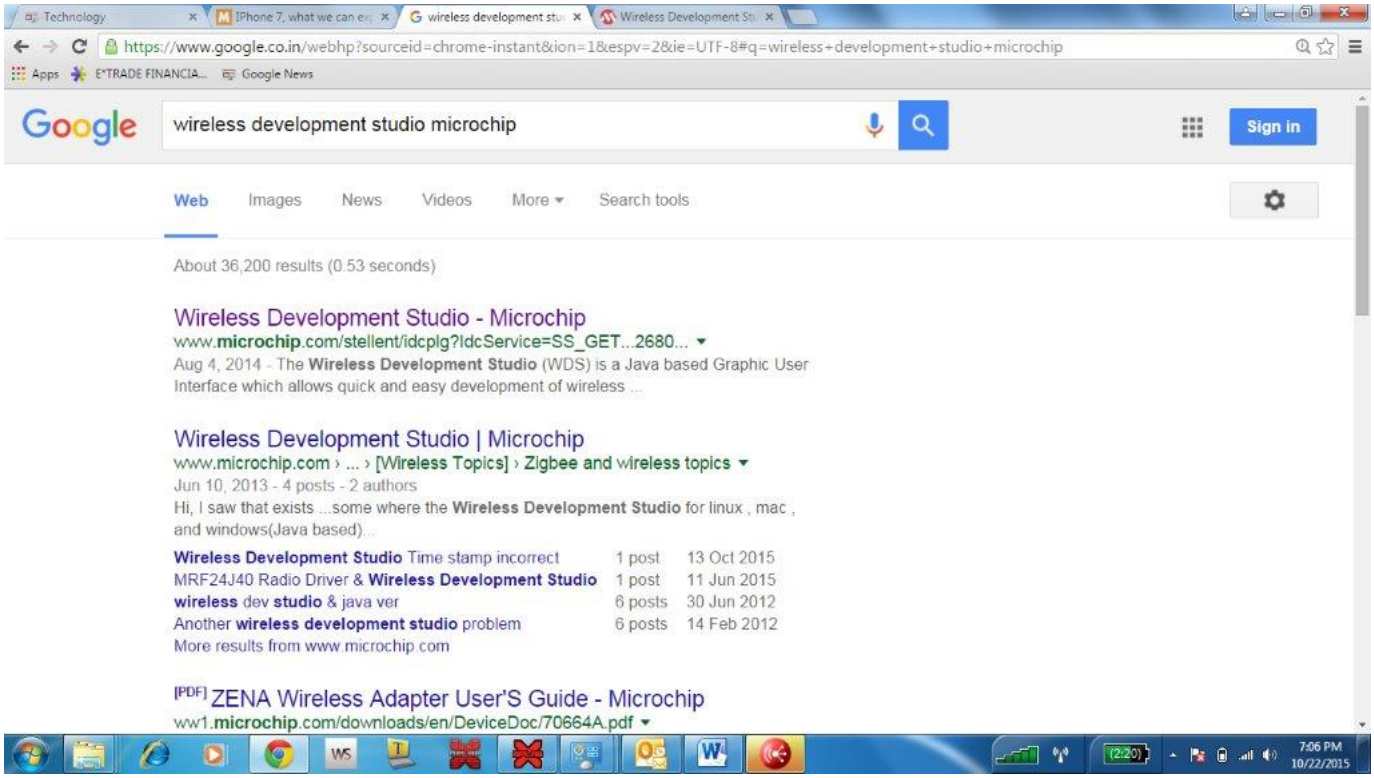
miwi_config_p2p.h

console.h

1.6.3.3 Wireless Development Studio (MiWi Network Sniffer)

Instruction for WDS and ZENA Installation (with Windows 7 on 64 bit machine) & Windows 8.1:

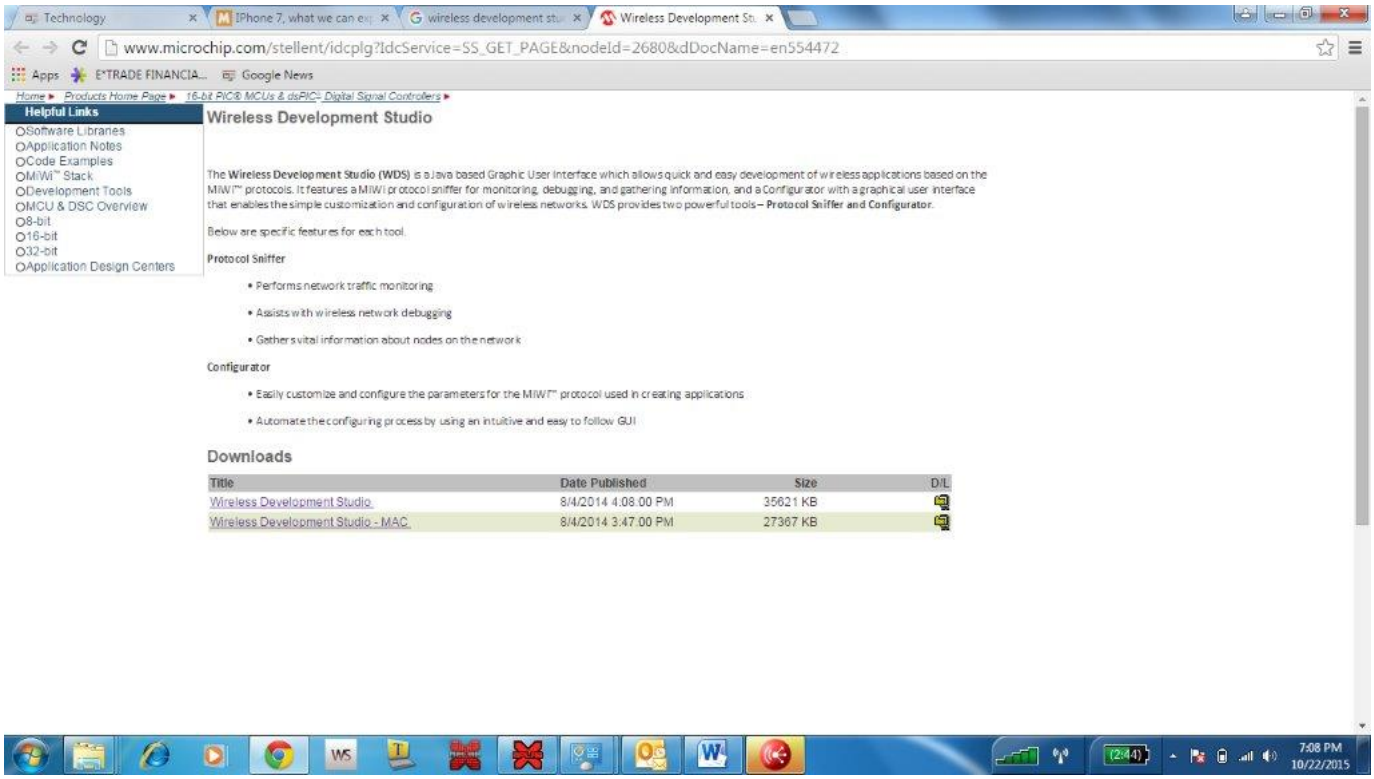
Downloading latest version WDS from Microchip website



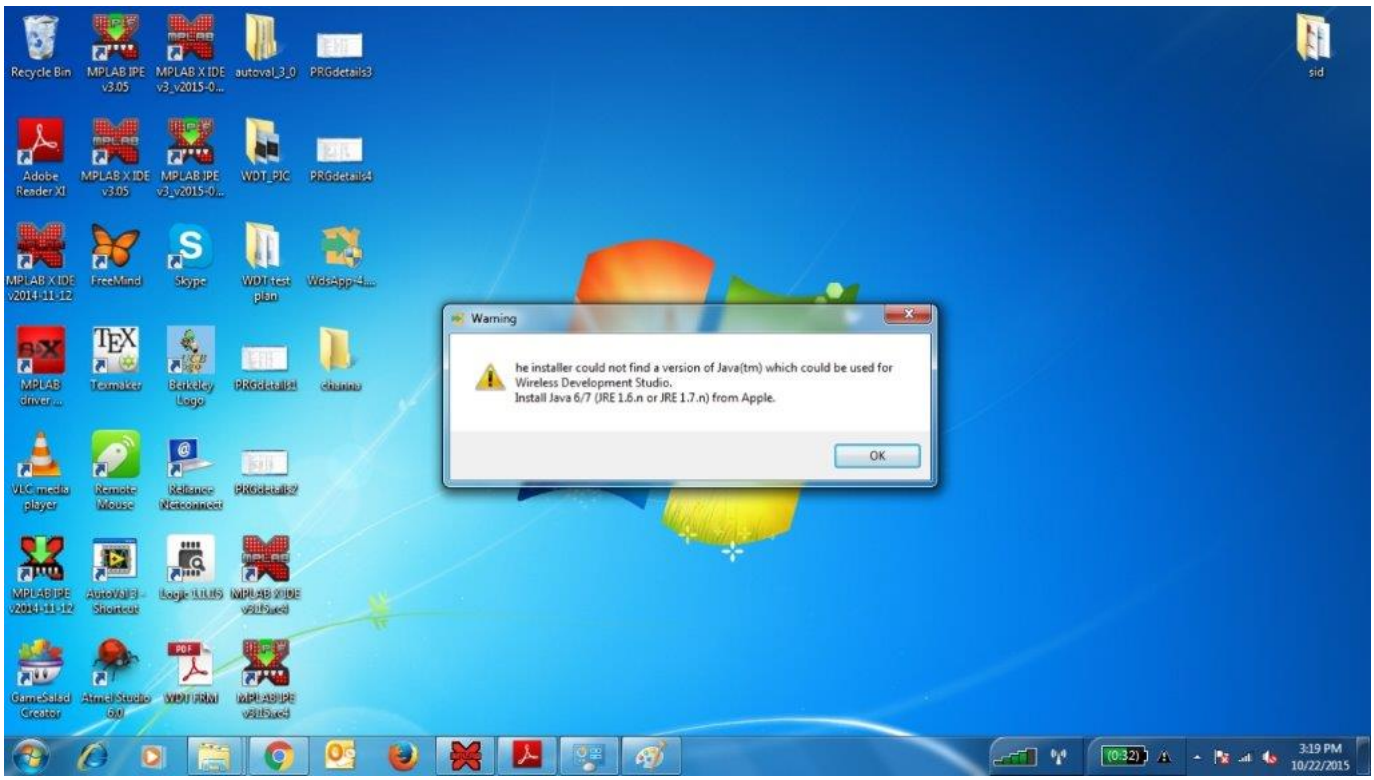
Download ZENA from Microchip website from the link : http://www.microchip.com/design-centers/wireless-connectivity/embedded-wireless/miwi-protocol

Scroll down and click on MiWi Tools Tab , bottom of the page a link to download wireless development studio for windows and mac OS is available.

Go to wireless development studio – Microchip and download the WDS ZIP file.



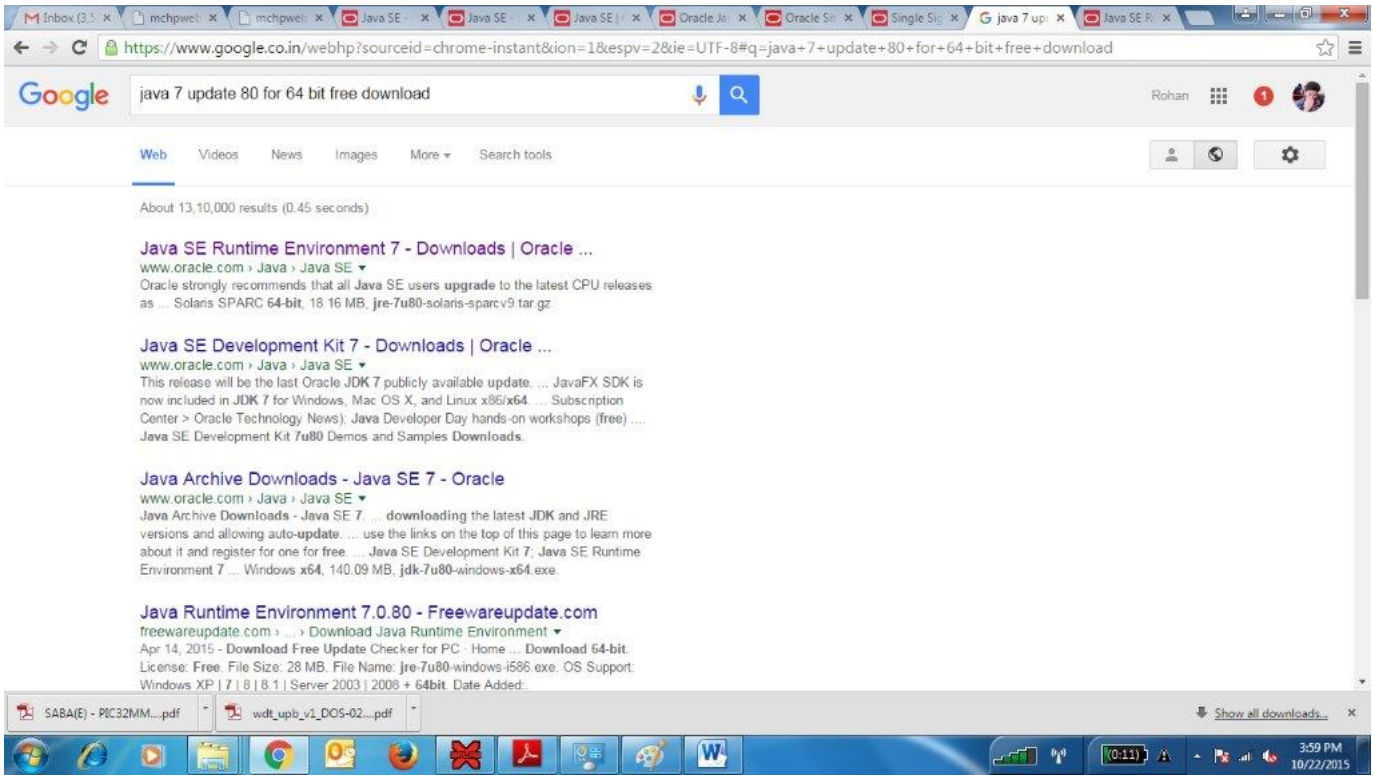
After downloading, try installing the WDS software, you will get a prompt as below



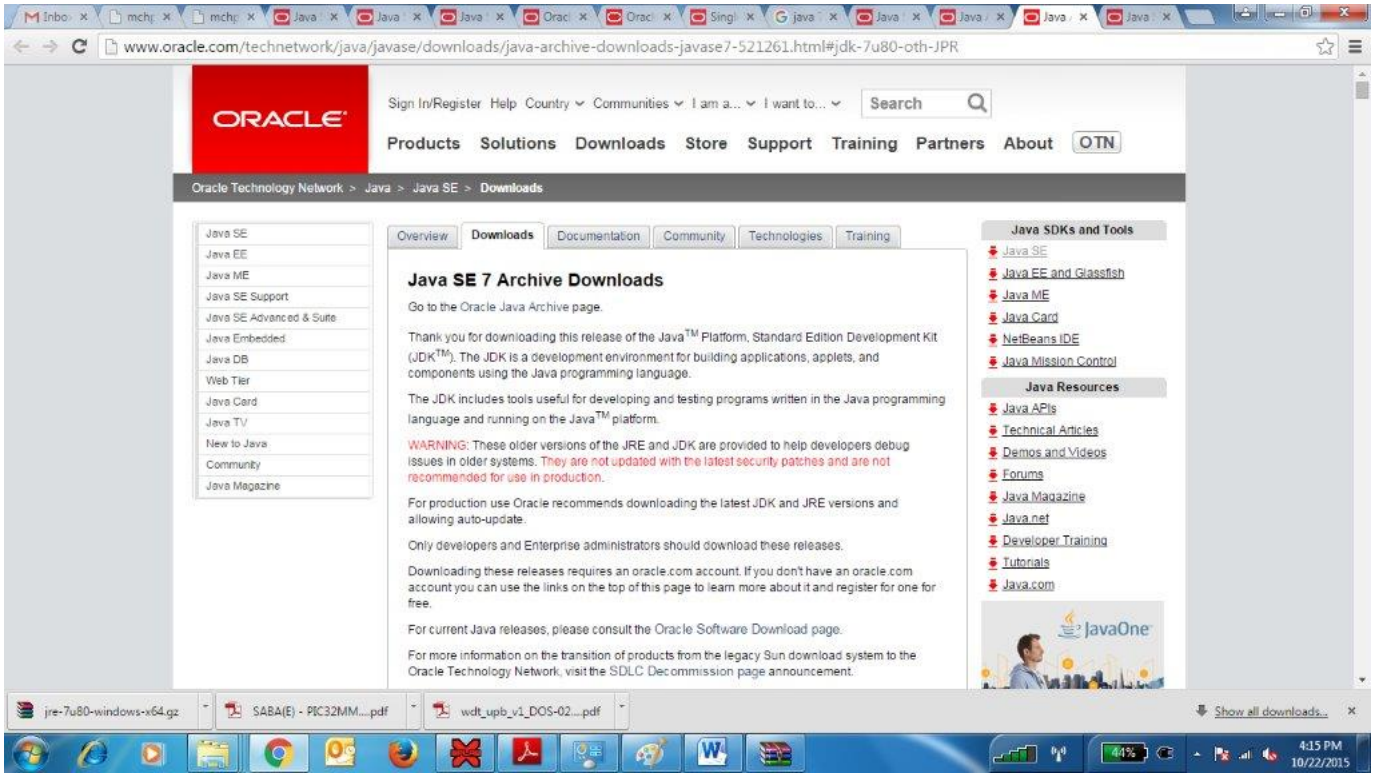
By clicking on OK you will go to web page of ORACLE to download Home page.

Go to Archive and download Java 7 update 80 (64 bit, version 7.0.800).

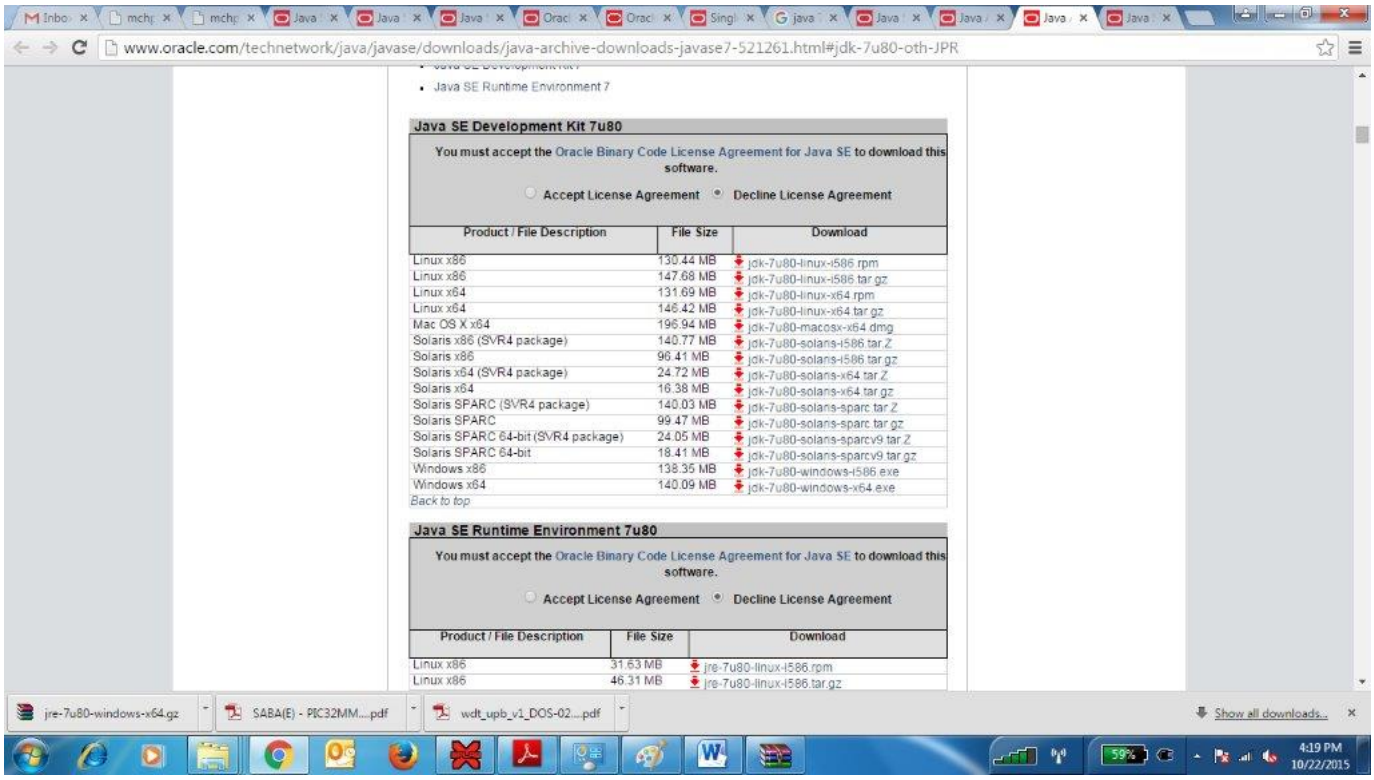
Install JRE1.7 or Java SE Development Kit 7 update 80 (64 bit, version 1.7.0.800)

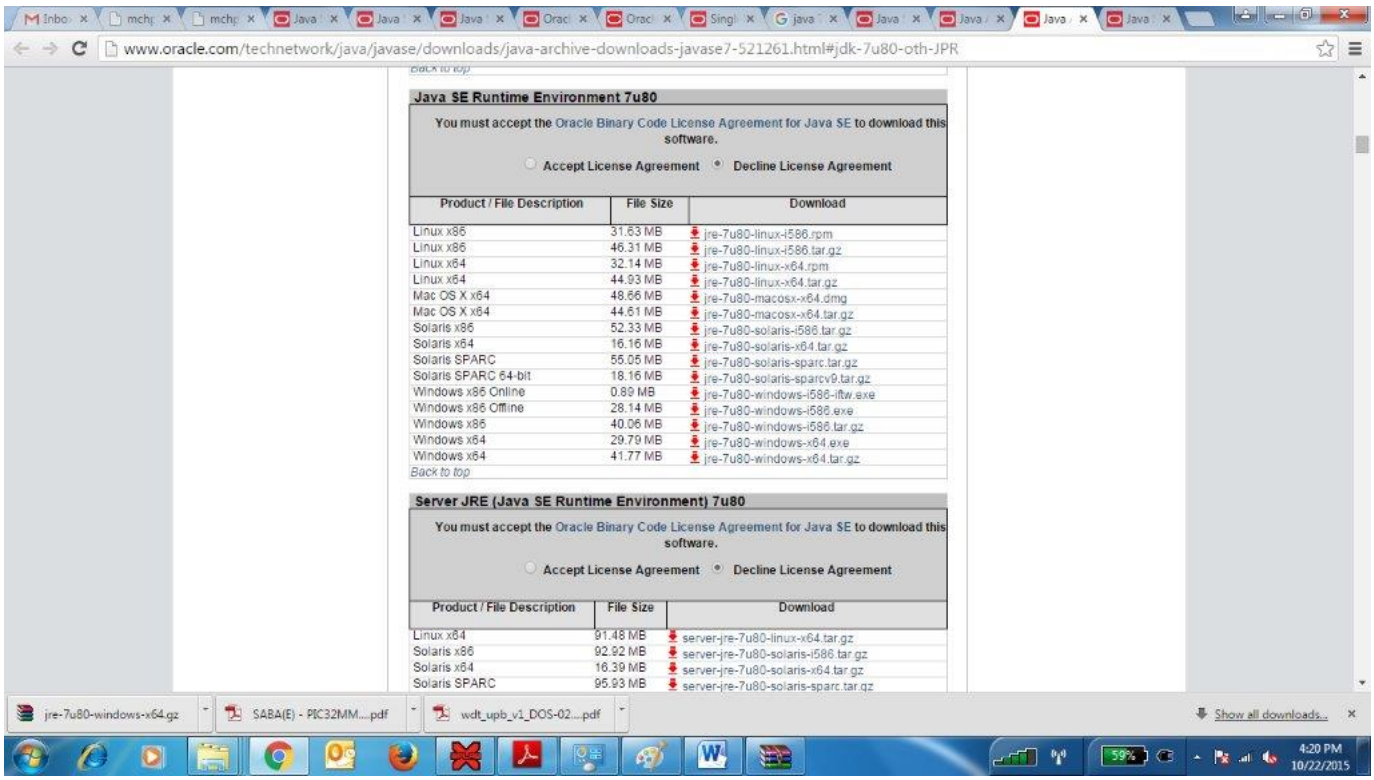


Click on Java Archive Download, it will open Oracle webpage.

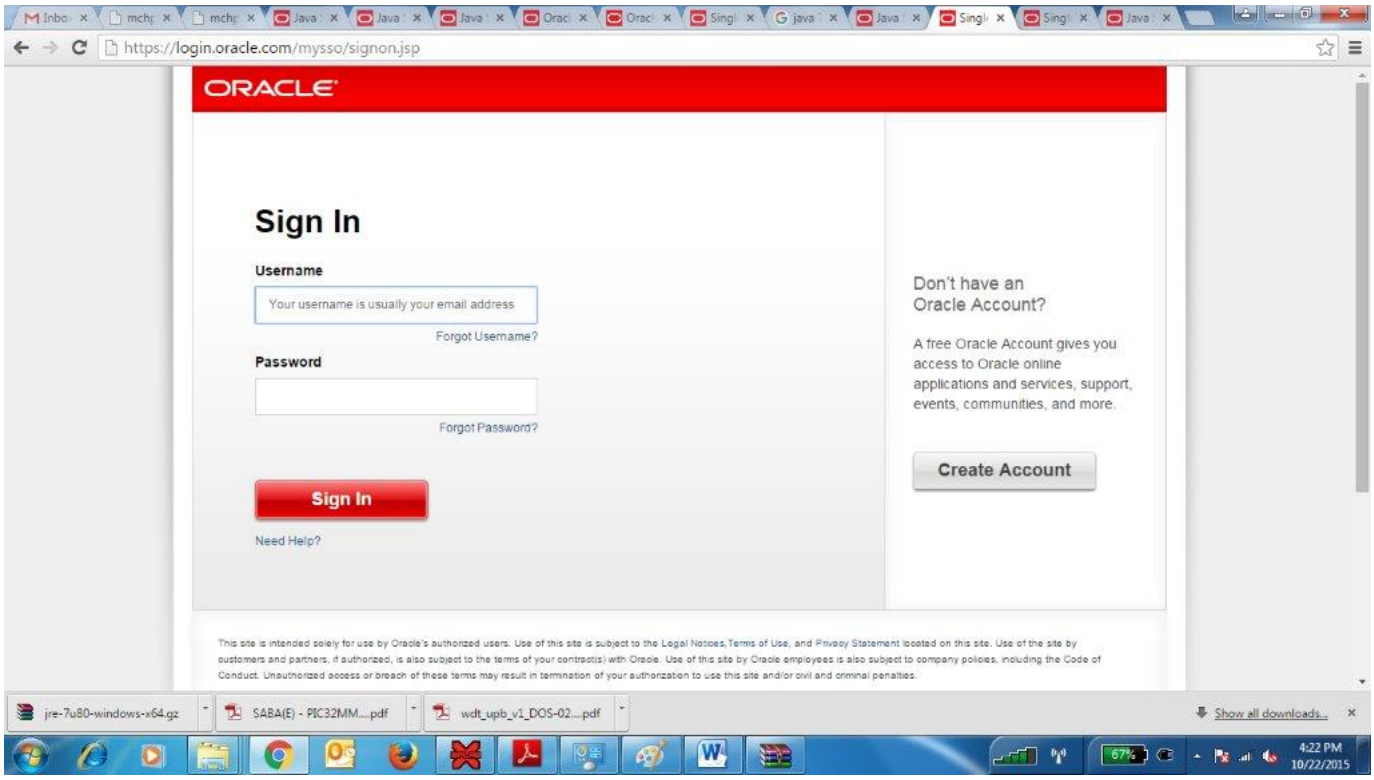


Scroll down and accept the license agreement, Download the Windows x64 JRE and JDK of 7u80.



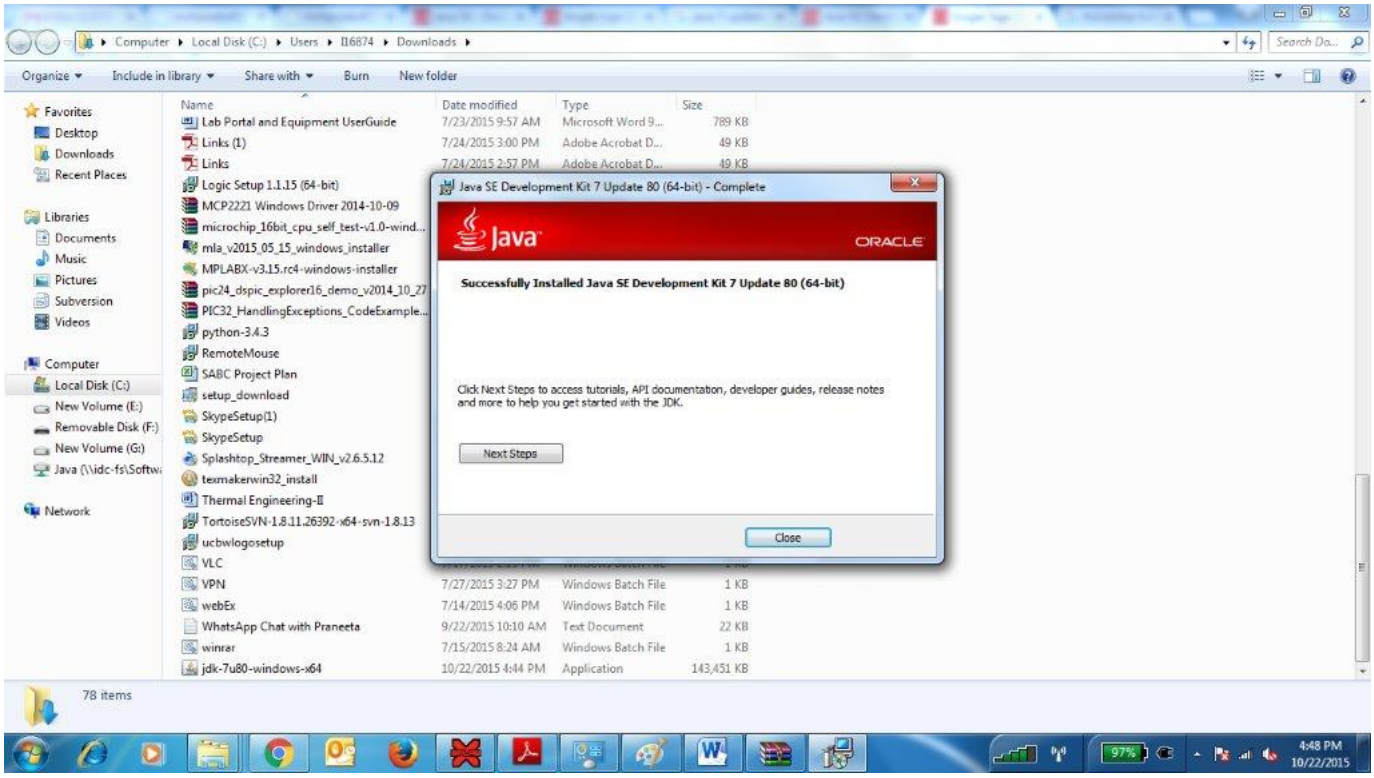


Next, you need to sign in to download. Sign in or create new account if don't have account.



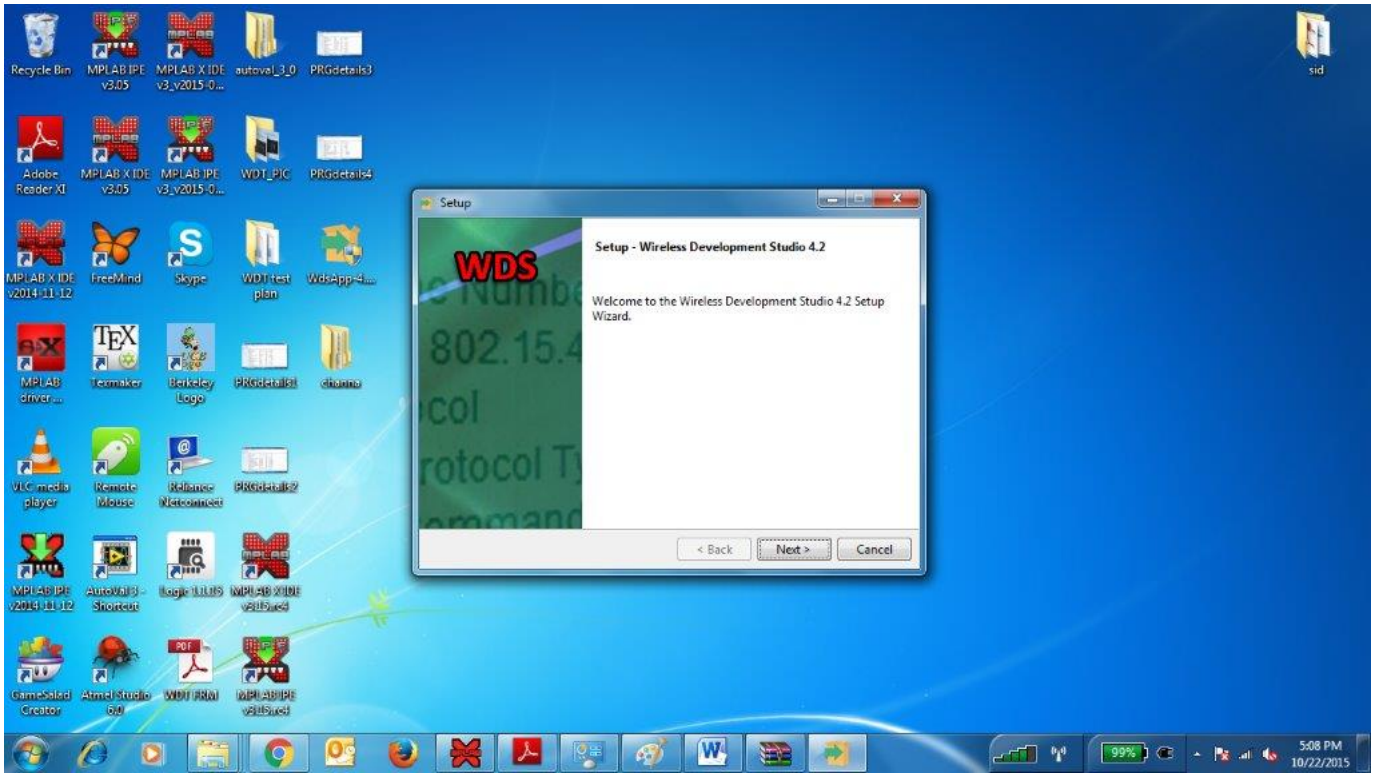
And it starts to download once you enter in the correct details and click on sign in.

After download open it and install.

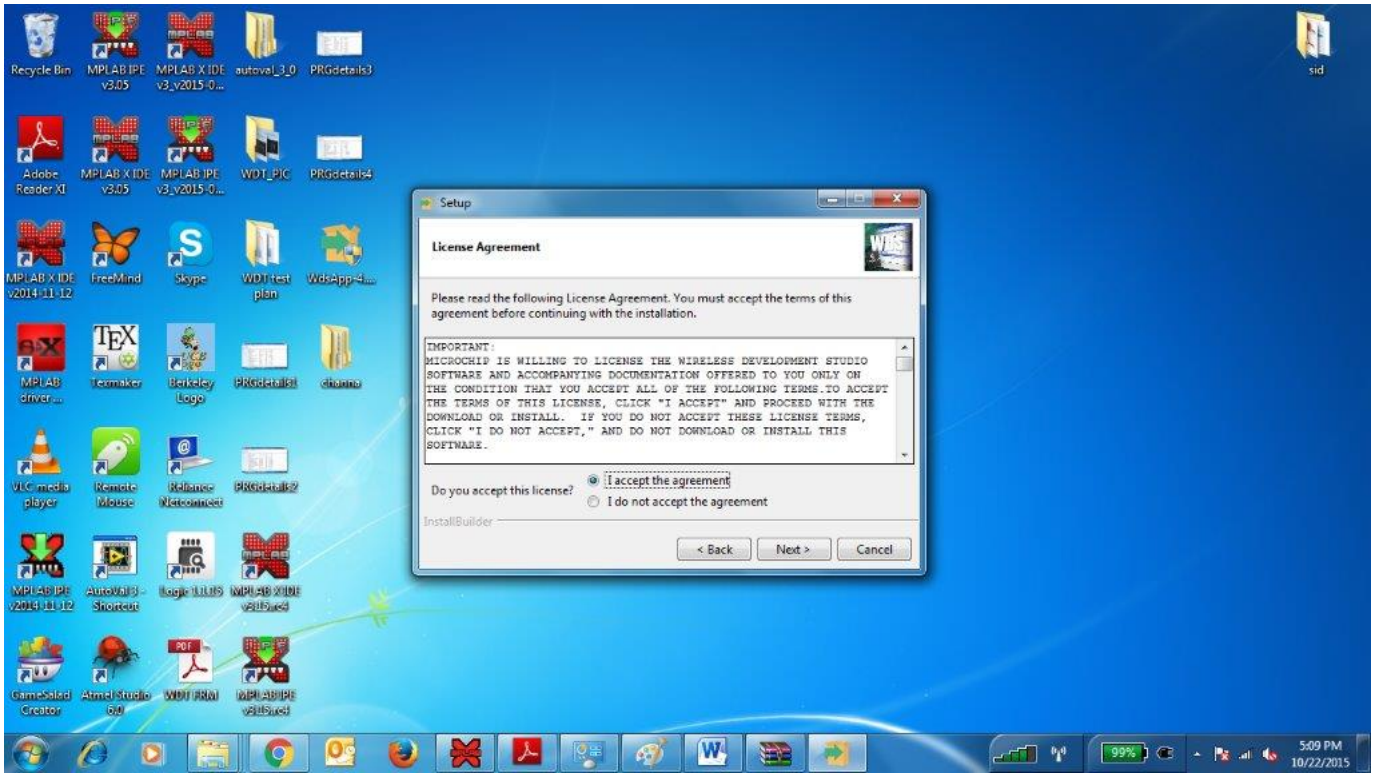


Close and install JRE now.

Now double click on .exe file of WDS and install the software.



Accept the license.

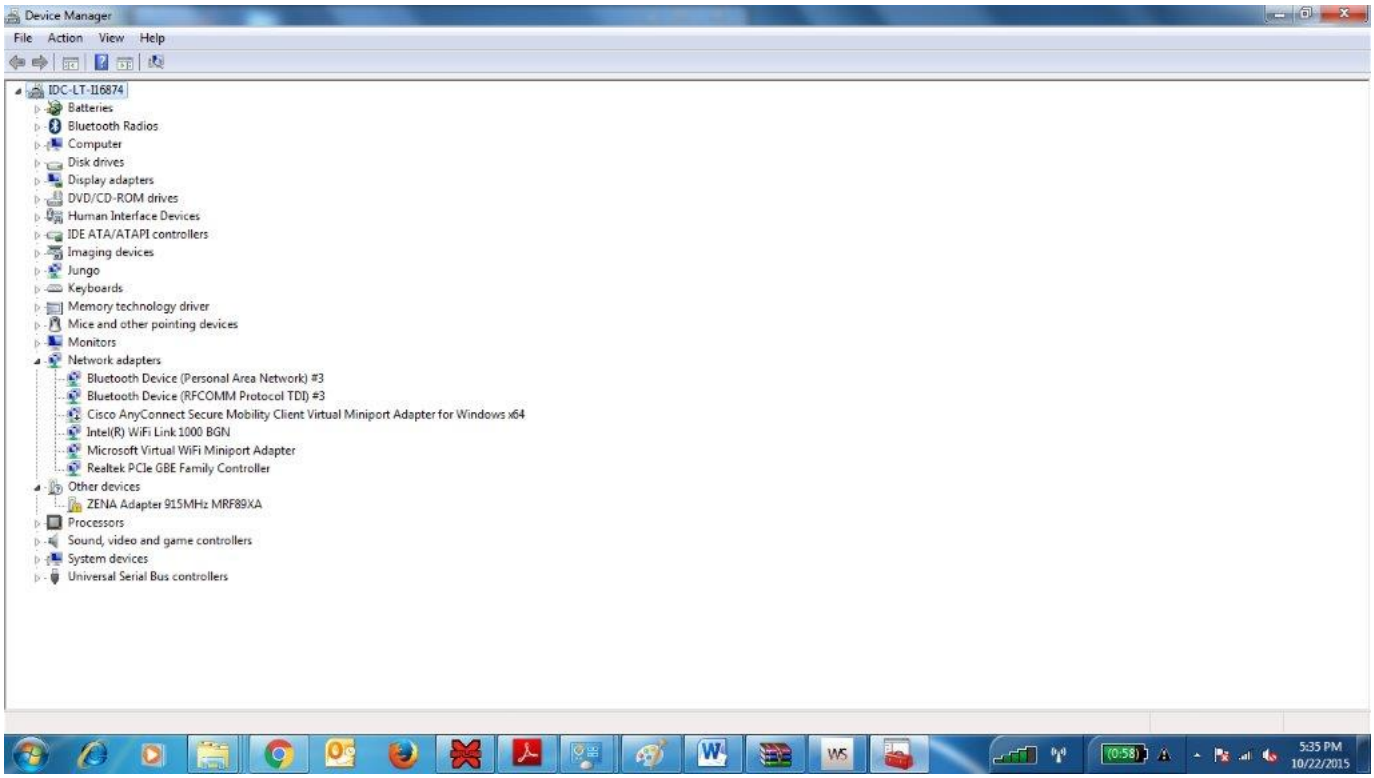


Go on clicking next. Finally WDS will be installed.

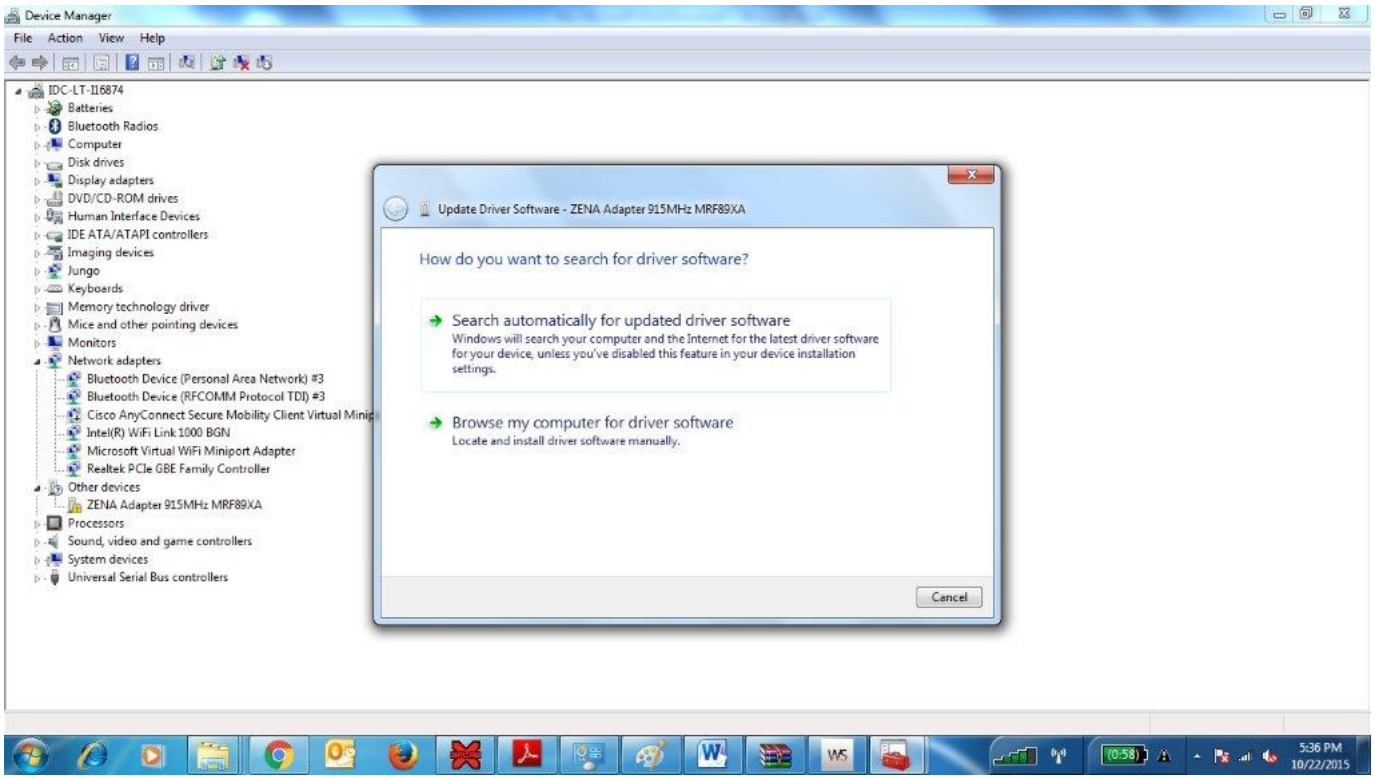
For ZENA detection and installation of drivers

First we should put the ZENA module into one of the USB port and should be able see the device in Device Manager.

For updating the Driver and Firmware go to device manager and right click on ZENA and give the path of the driver location. As shown below.



Right click and give update driver.

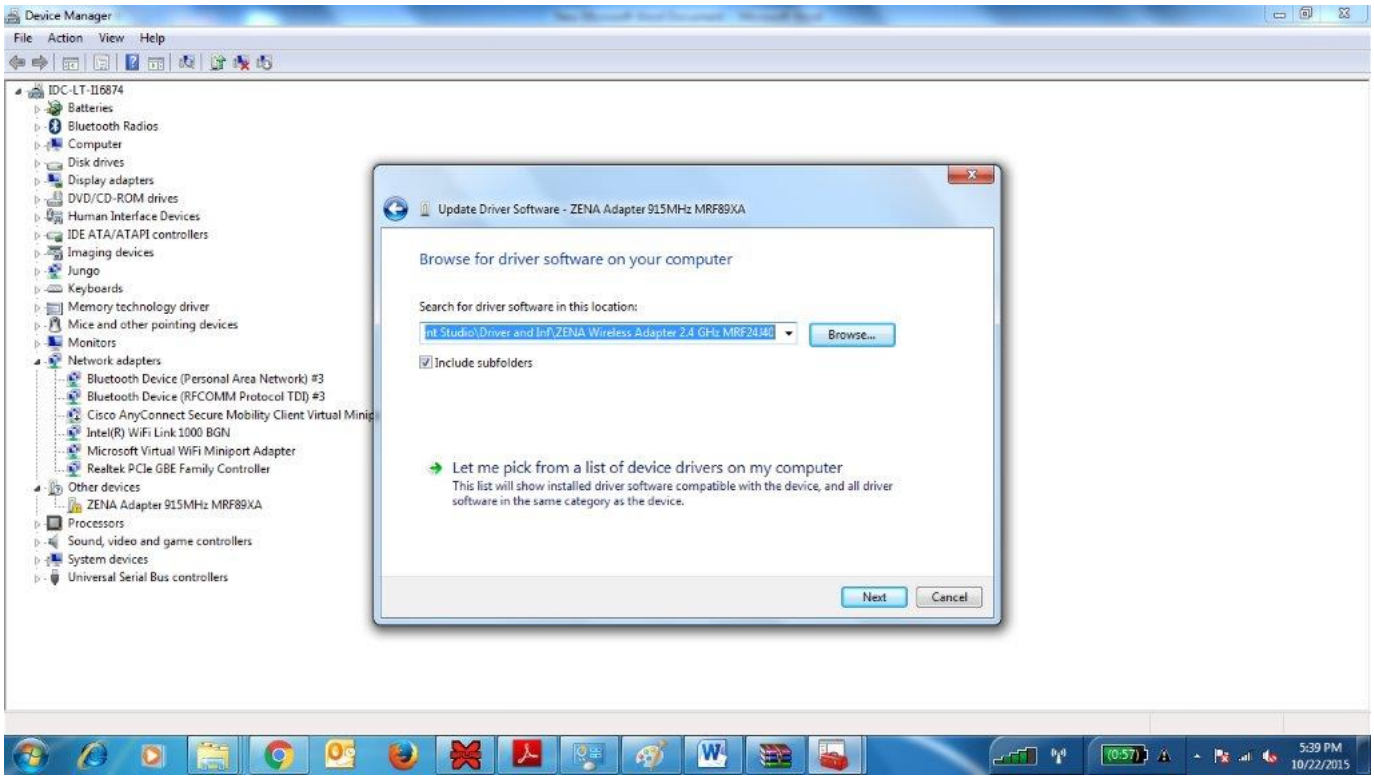


Choose browse my computer.

Give path as below for different ZENA modules.

1. For 2.4GHz ZENA sniffer.

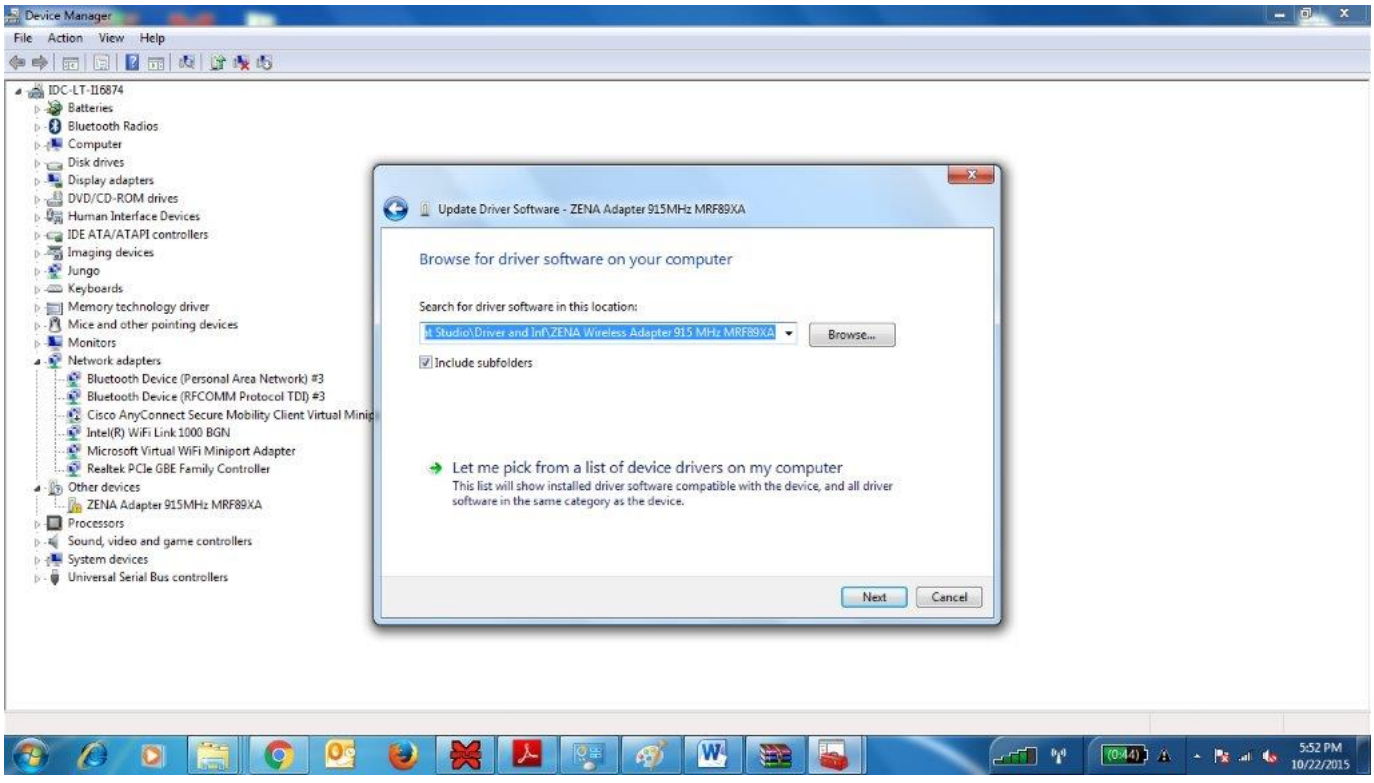
C:\Program Files (x86)\Microchip\Wireless Development Studio\Driver and Inf\ZENA Wireless Adapter 2.4 GHz MRF24J40



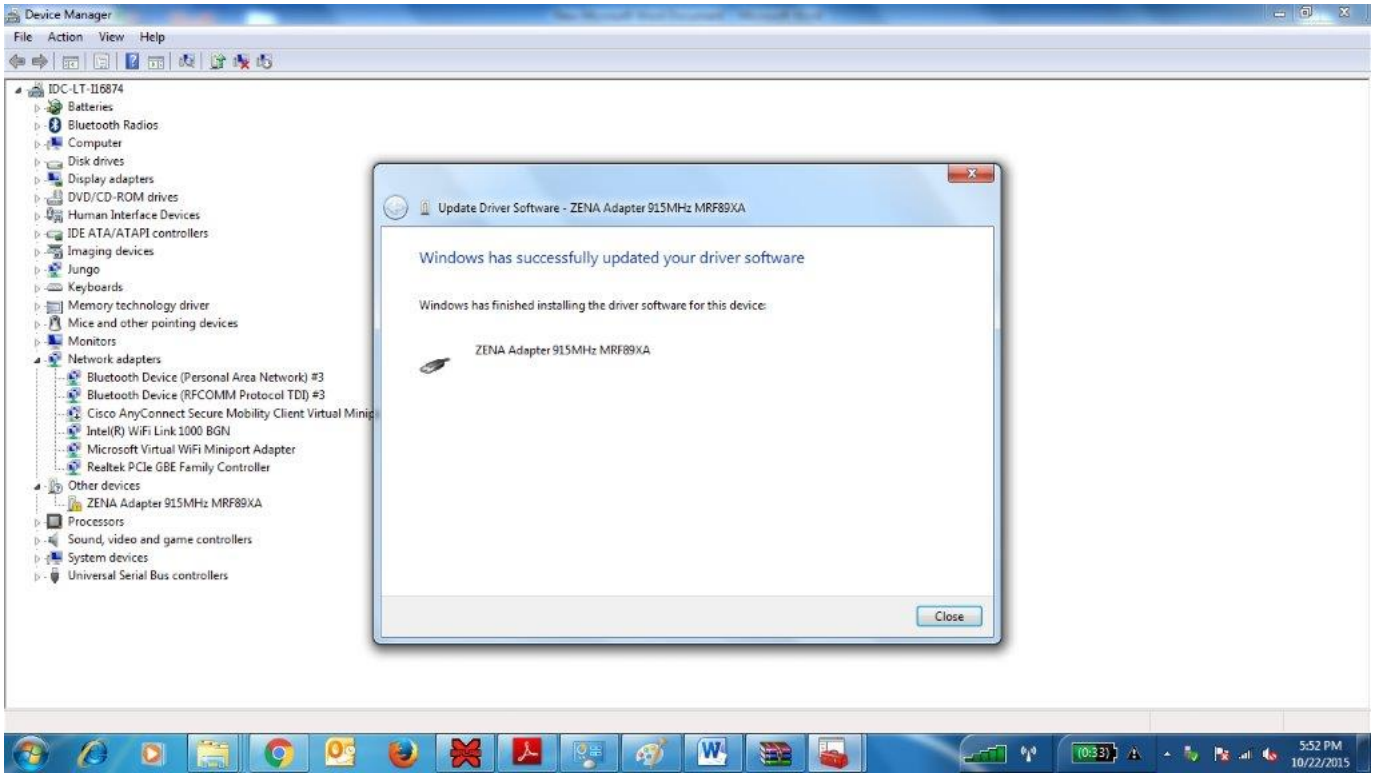
2. For 915MHz ZENA sniffer

give path as below:

C:\Program Files (x86)\Microchip\Wireless Development Studio\Driver and Inf\ZENA Wireless Adapter 915 MHz MRF89XA



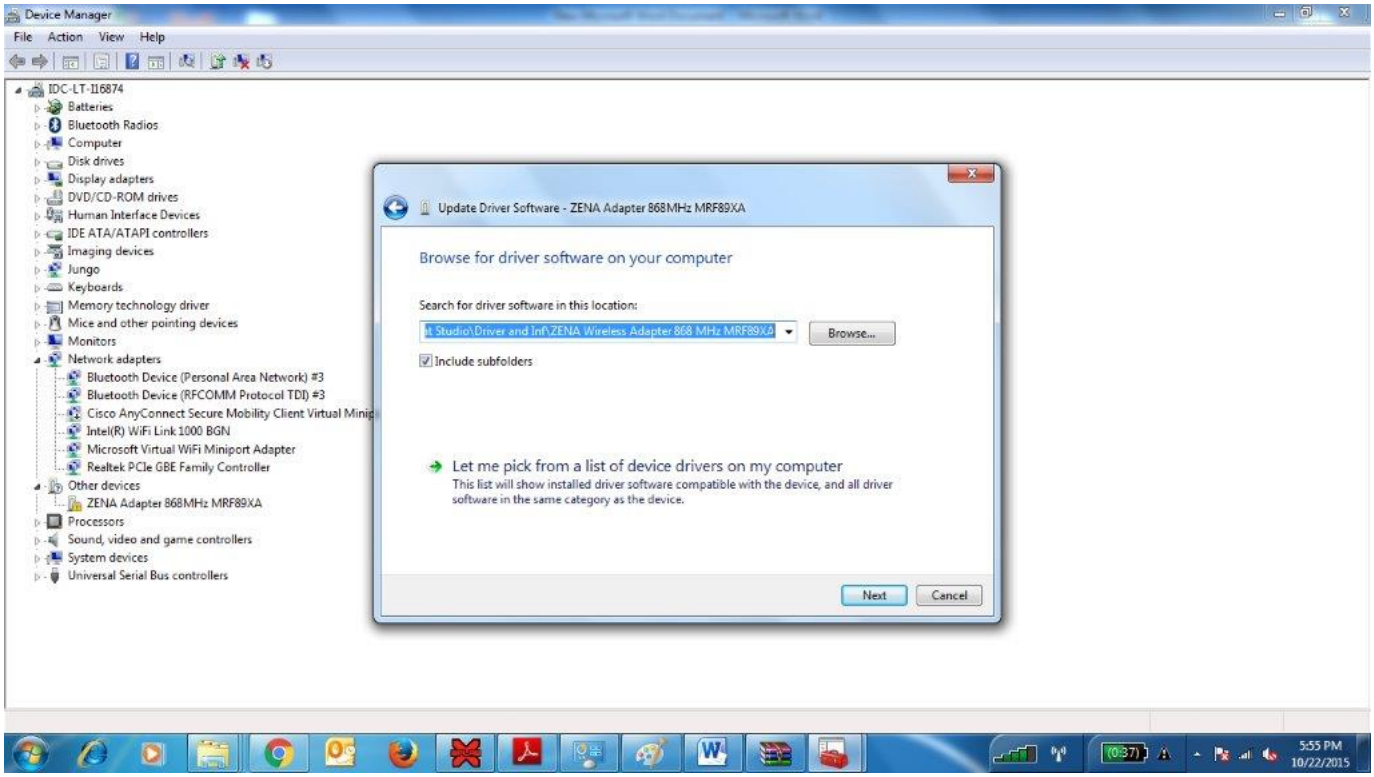
Click on next. And shows



3. For 868MHz ZENA sniffer

Path:

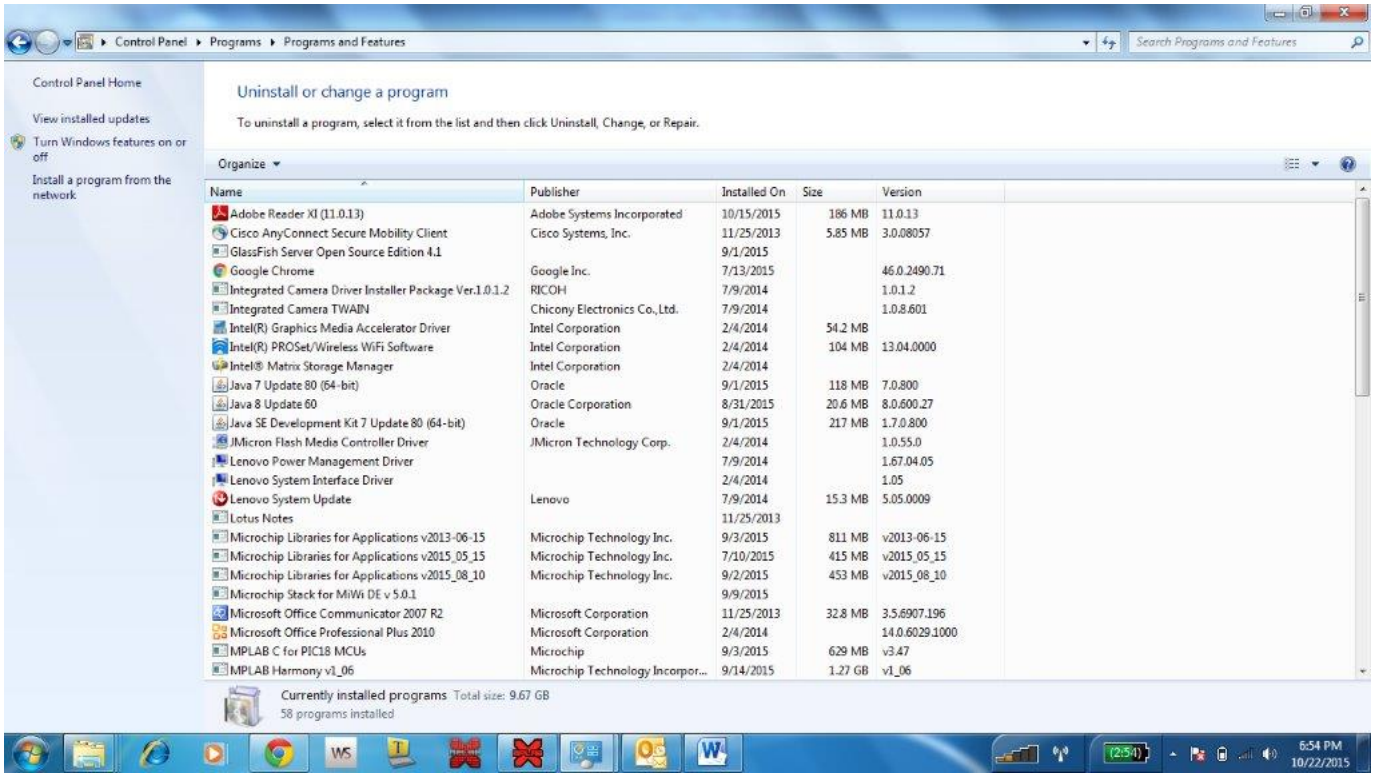
C:\Program Files (x86)\Microchip\Wireless Development Studio\Driver and Inf\ZENA Wireless Adapter 868 MHz MRF89XA



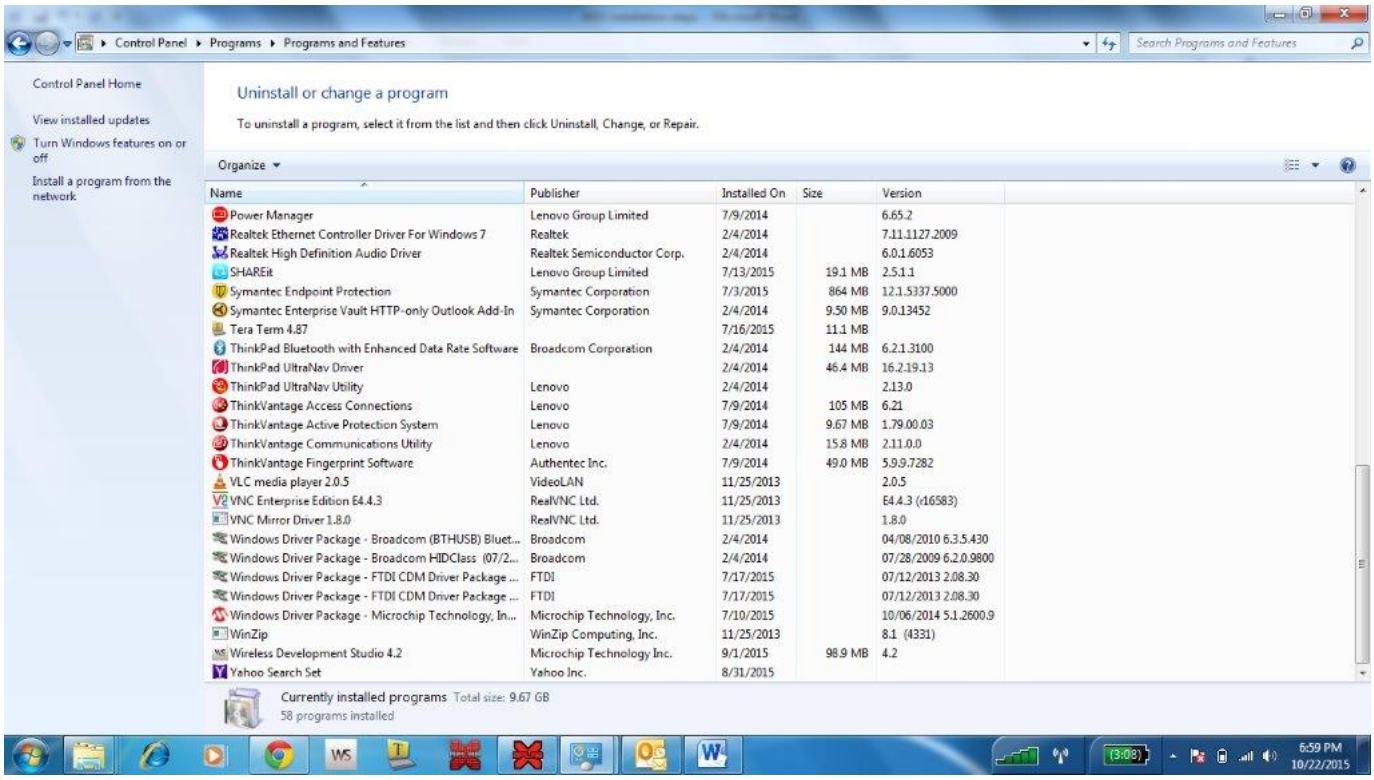
Click on next. It will be installed.

Checking installed version

After installing the software go to Windows CONTROL PANEL ? Programs and Features ? Uninstall or change a program screen, and you should be able to see the installed JDK and JAVA version 7.0.800 in it as shown below.

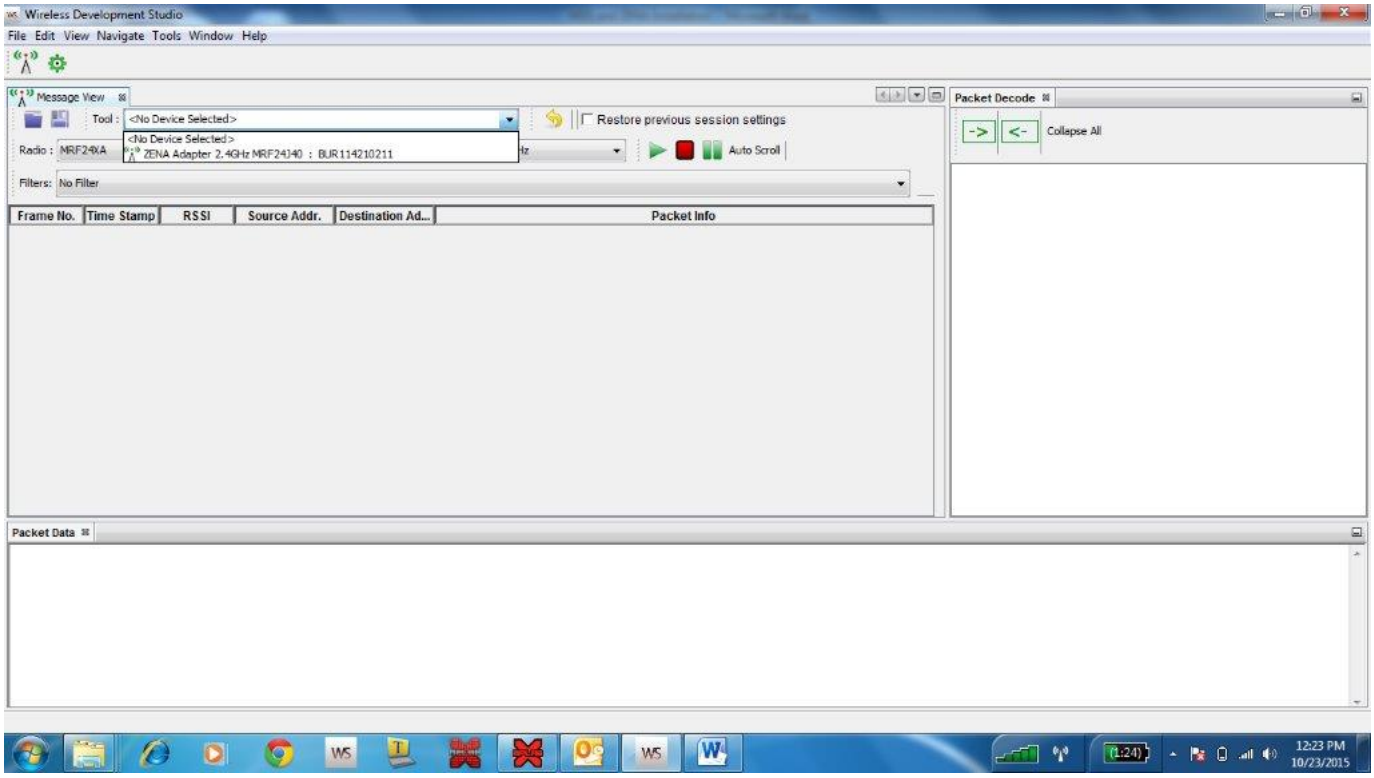


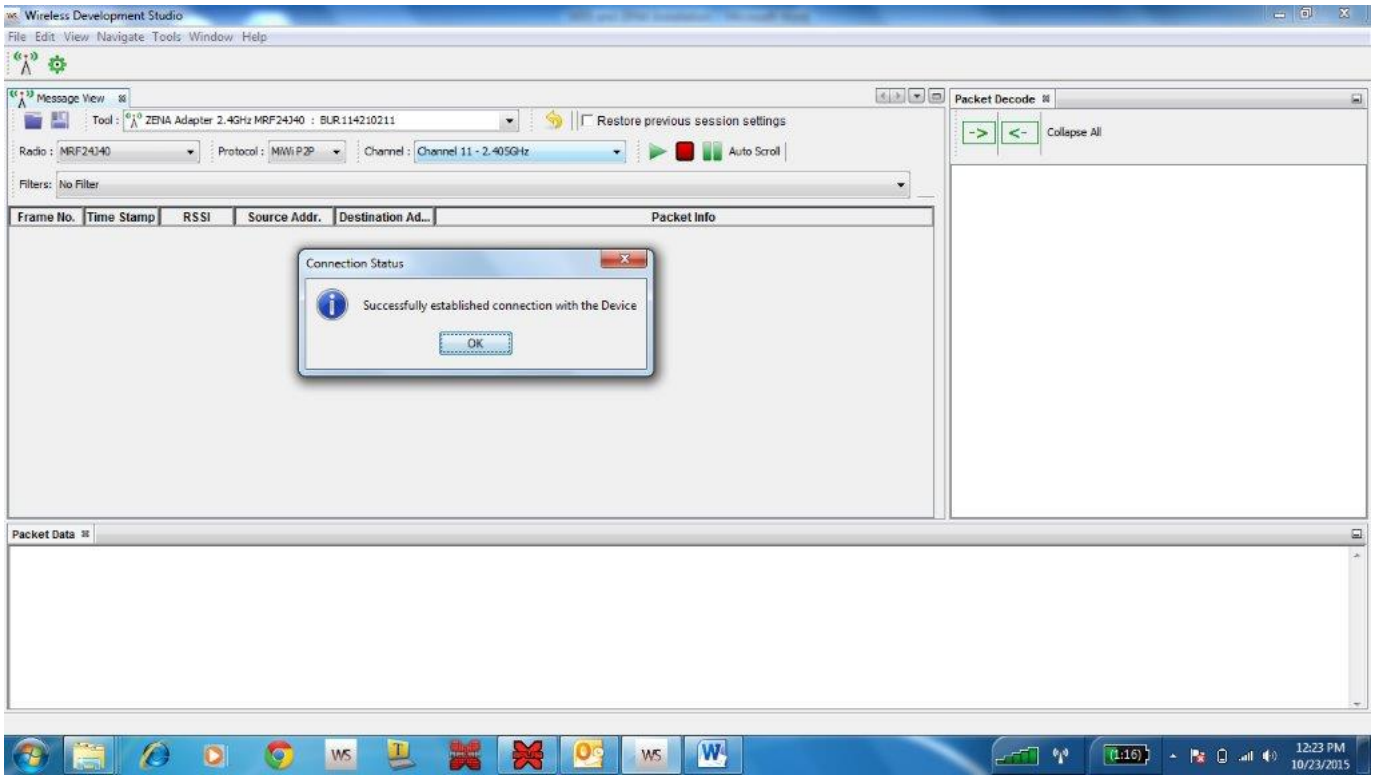
And also WDS version 4.2 as shown below.



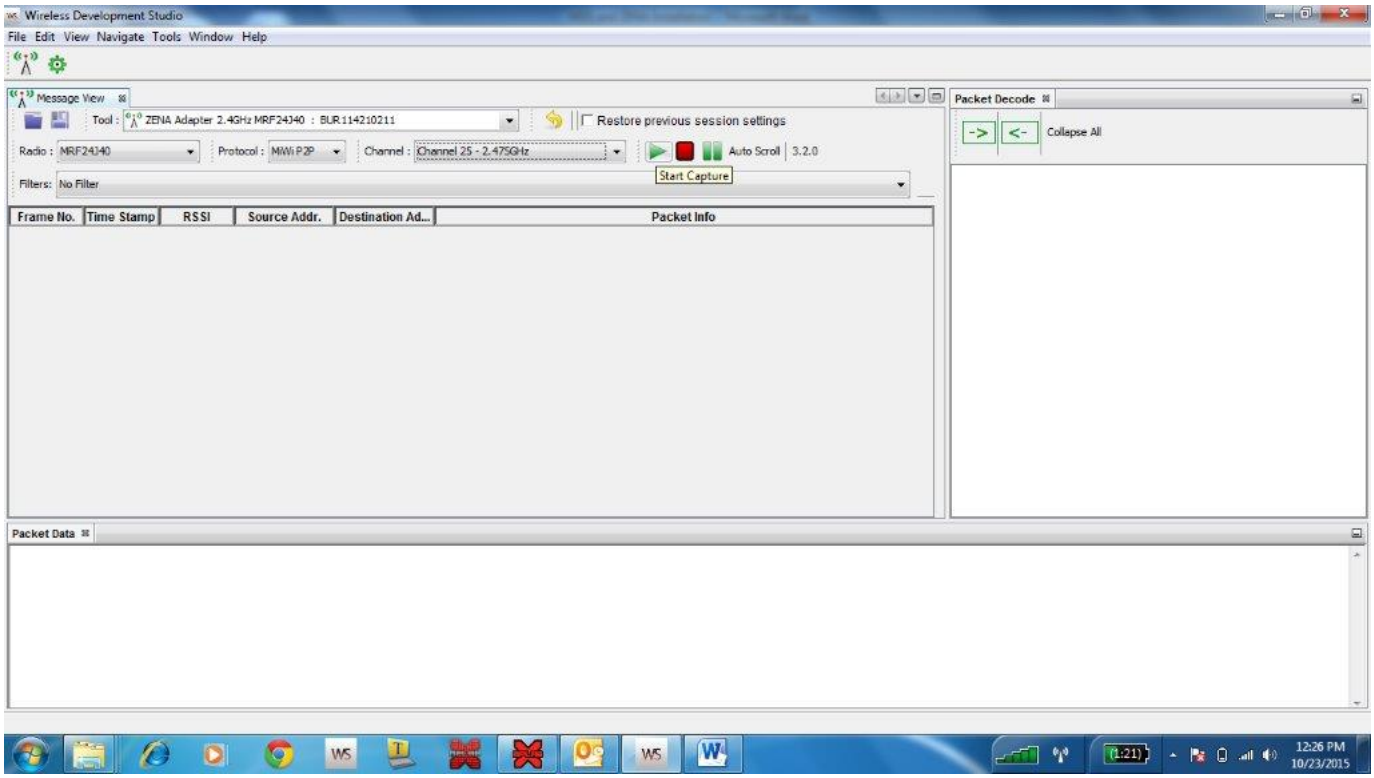
Procedure to use ZENA sniffer in WDS:

Put the ZENA sniffer in USB port and open the WDS after that. You can see the detection of ZENA as shown below and select that.

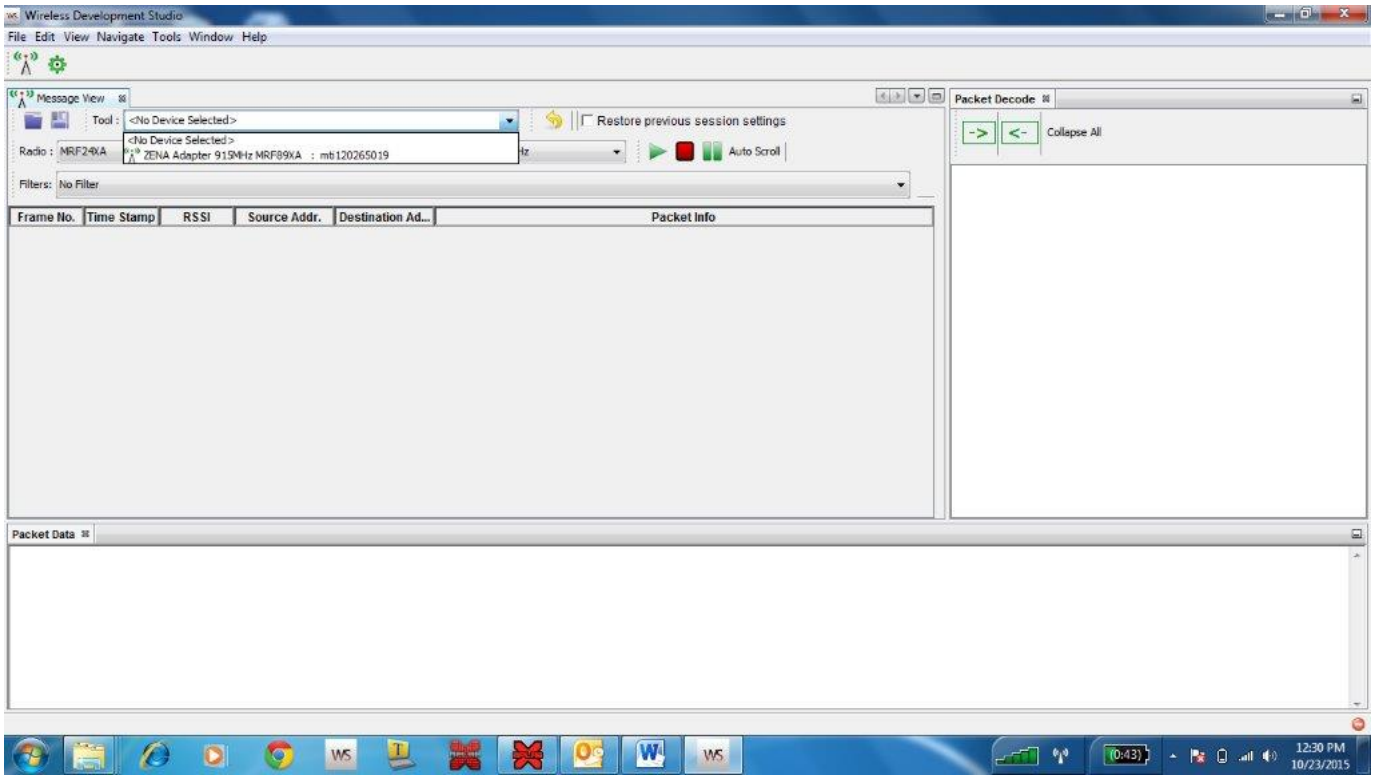


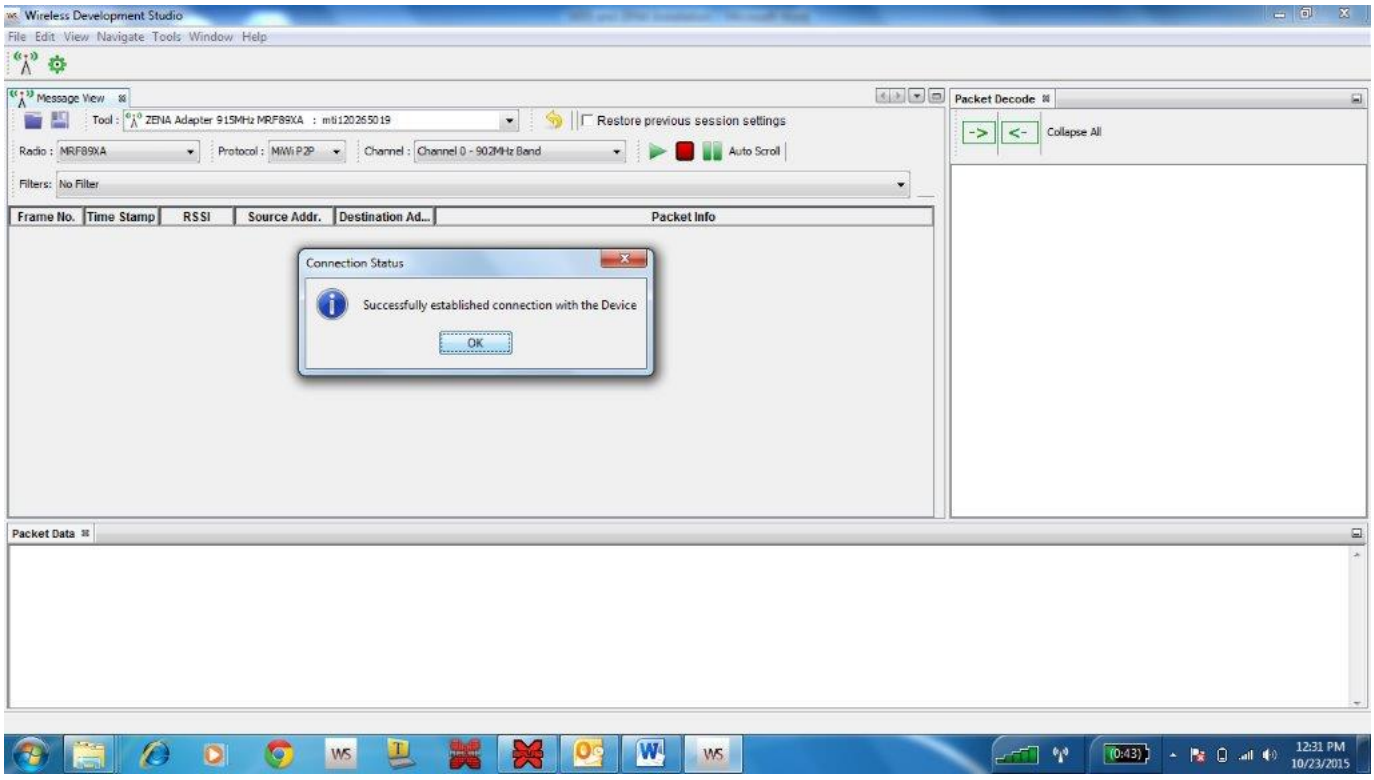


Click on ok. And select the radio which you are using, and also the protocol used in the stack of MiWi and the channel in which the MiWi is operating. After setting as told, click on start capturing to see the packets.

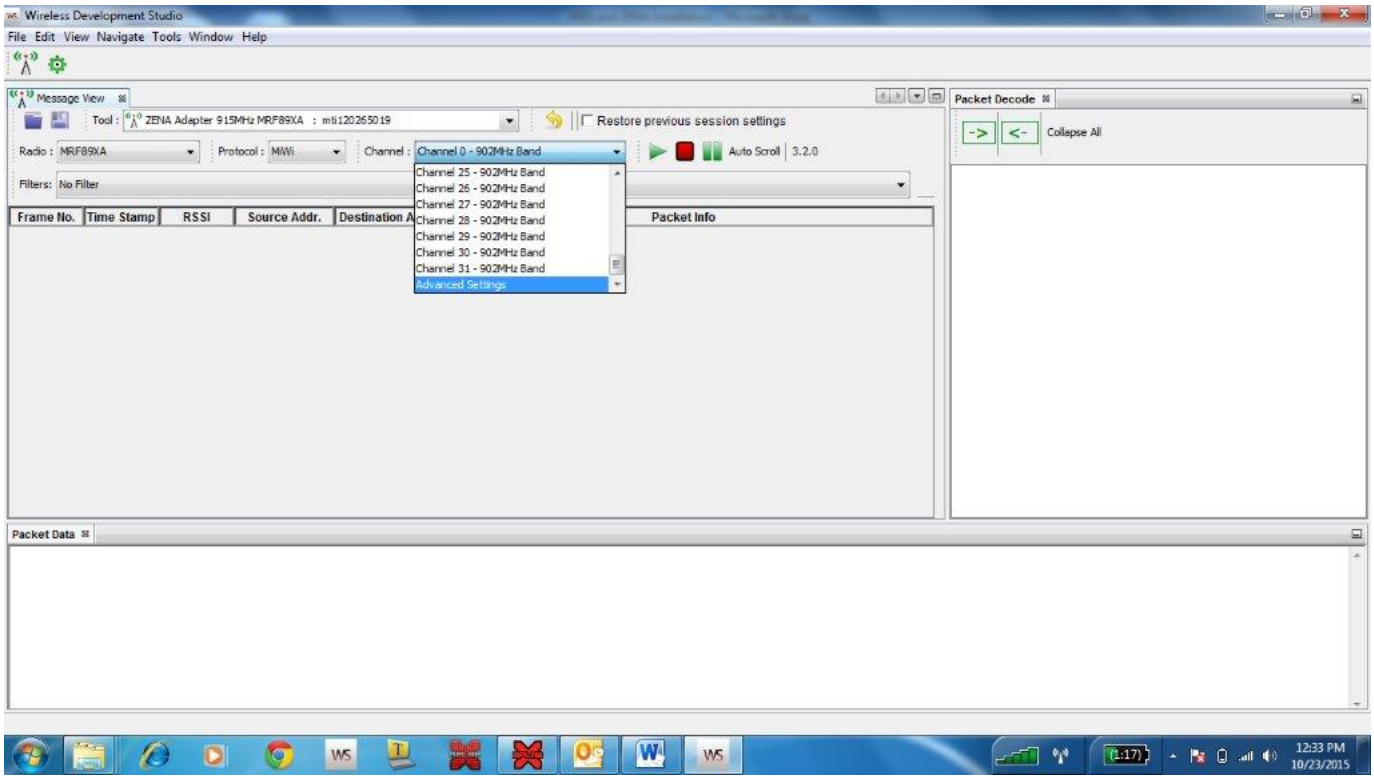


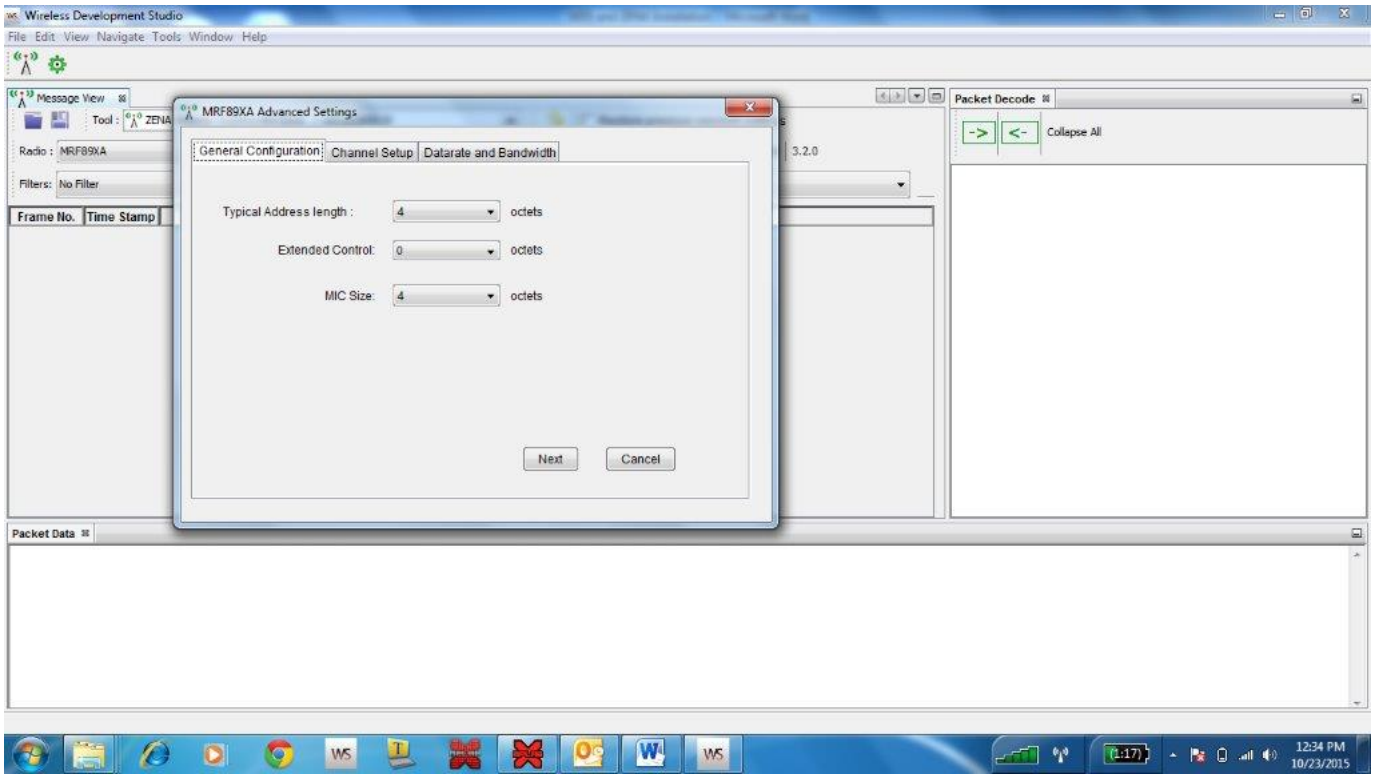
Advanced settings only for MRF89XA modules:



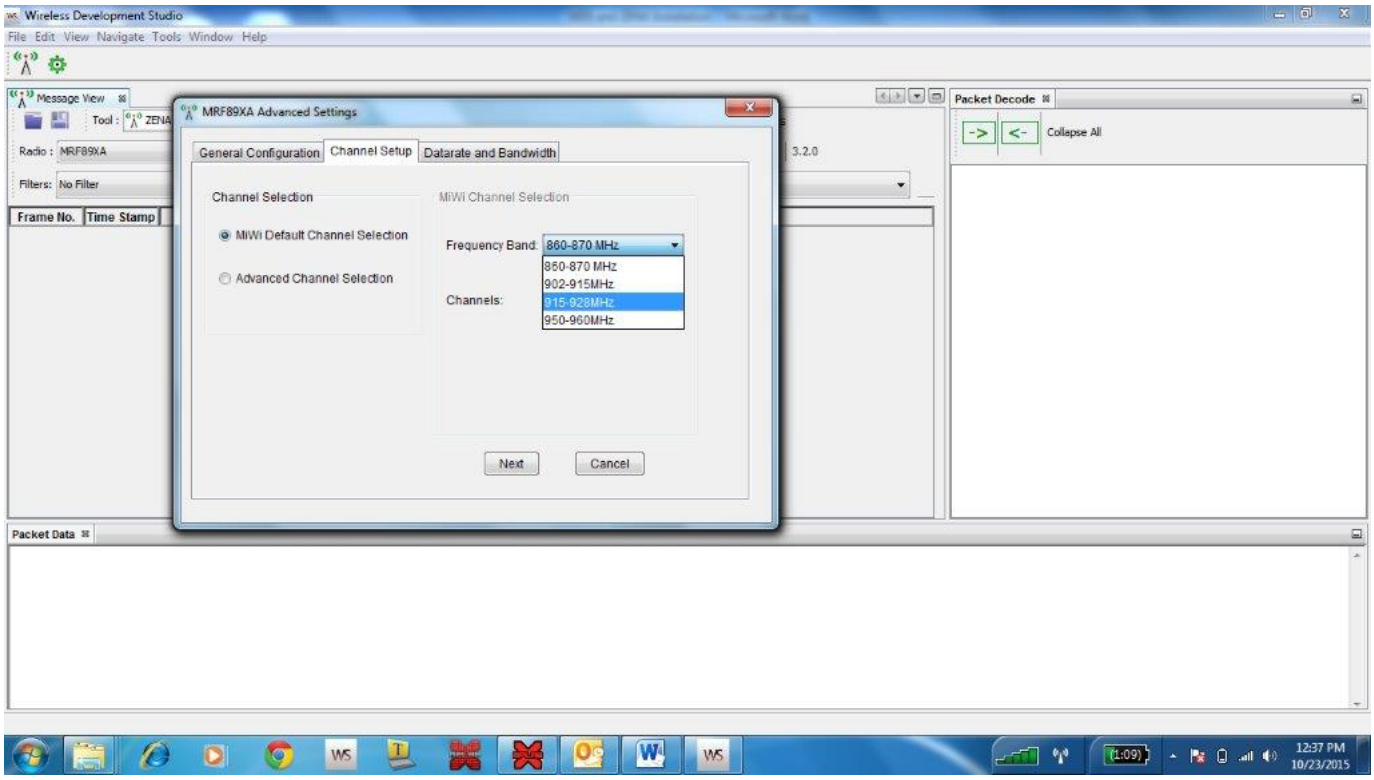


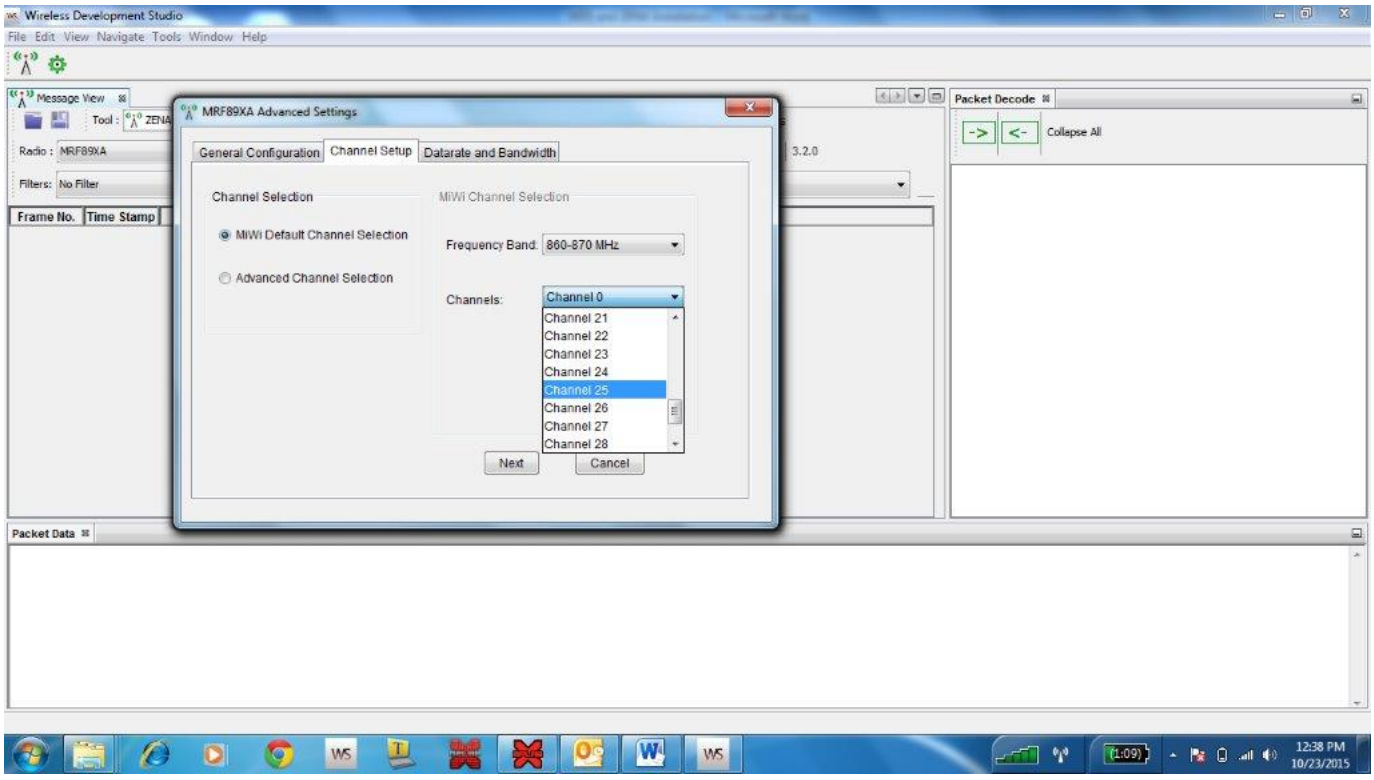
After selection of radio, protocol go to channel selection, and scroll down to end and click on the advanced settings. You will get new dialog window of advanced settings as shown below.

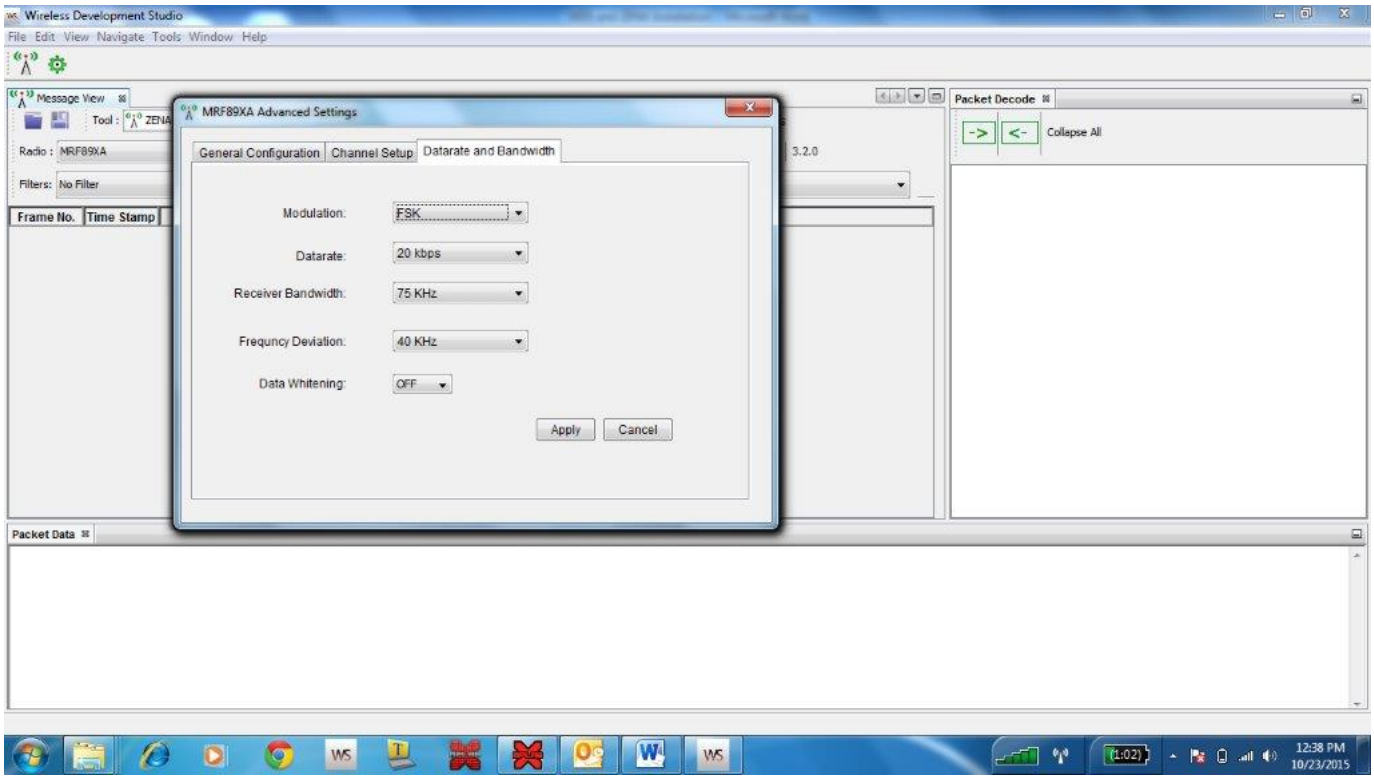




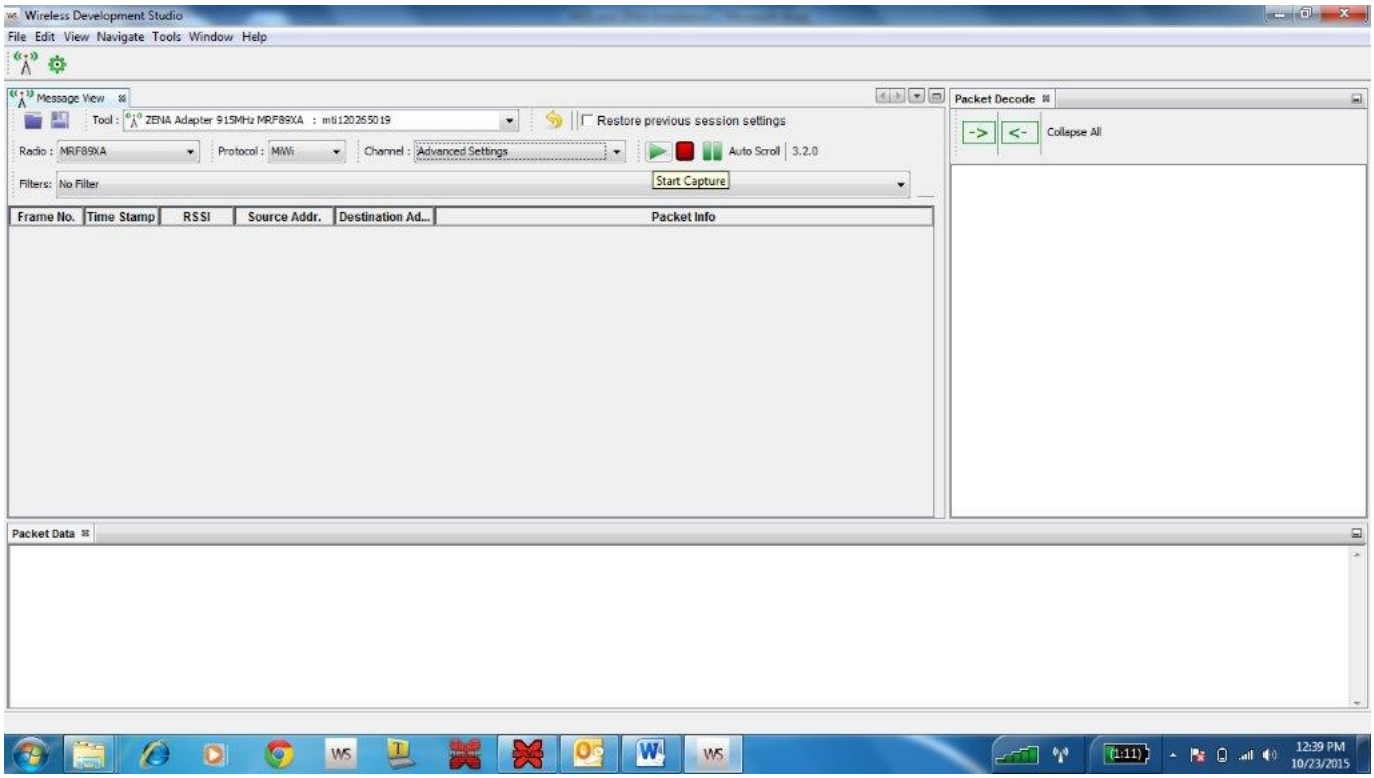
After that click on next or go to channel setup, and there select the whichever operating frequency range you want and whichever channel you want, select that and click on next and click on apply.


Wireless Development Studio
File Edit View Navigate Tools Window Help
Message View
Tool : ZENA
Radio : MRF89XA
Filters: No Filter
Frame No.
Time Stamp
MRF89XA Advanced Settings
General Configuration
Channel Setup
Datarate and Bandwidth
Channel Selection
MiWi Default Channel Selection
Advanced Channel Selection
MiWi Channel Selection
Frequency Band:
860-870 MHz
902-915MHz
915-928MHz
950-960MHz
Channels:
Next
Cancel
Packet Decode
Collapse All
Packet Data
12:37 PM
10/23/2015


Wireless Development Studio
File Edit View Navigate Tools Window Help
Message View
Tool : ZENA
Radio : MRF89XA
Filters: No Filter
Frame No.
Time Stamp
MRF89XA Advanced Settings
General Configuration
Channel Setup
Datarate and Bandwidth
Channel Selection
MiWi Default Channel Selection
Advanced Channel Selection
MiWi Channel Selection
Frequency Band:
860-870 MHz
Channels:
Channel 0
Channel 21
Channel 22
Channel 23
Channel 24
Channel 25
Channel 26
Channel 27
Channel 28
Next
Cancel
Packet Decode
Collapse All
Packet Data
12:38 PM
10/23/2015



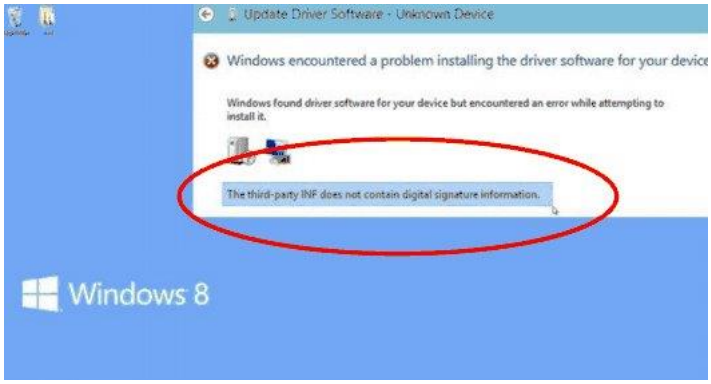
And after this click on start capture to see the packets



Installation of ZENA Wireless Adapter process differs in Windows 8.1:

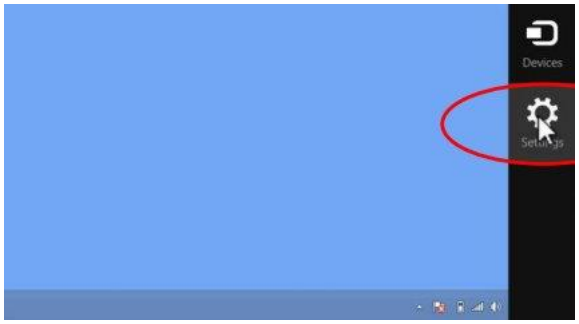
Plug in the ZENA Wireless Adapter in one of the USB port and the device should be seen in device manager. When you try to update the driver, it throws an error saying that

“The third party INF does not contain digital signature information”.

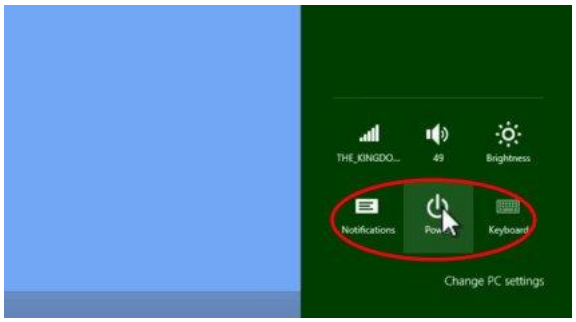


This is because the driver for the device is not “Digitally signed” for Windows 8. In order to overcome this error you have to disable the Windows 8 driver signature enforcement and install the driver that your device needs. Following are the steps to disable that feature:

Step 1: Move your cursor to the right of the screen to access the Charm Bar and choose: Settings

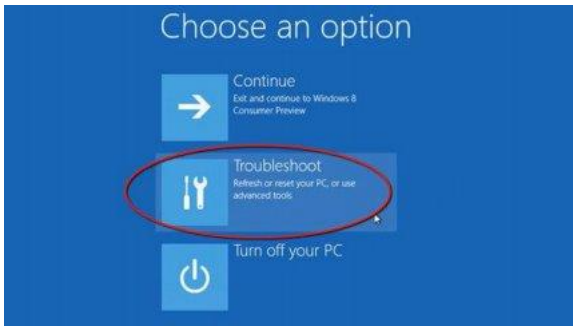


Step 2: Choose Power



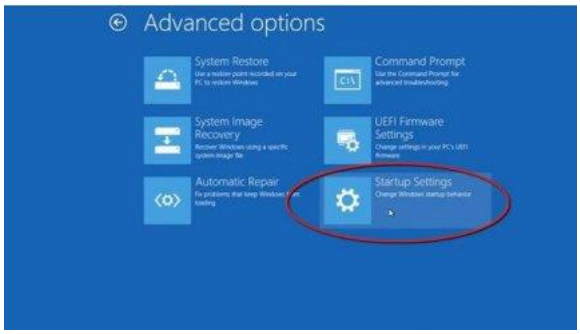
Step 3: while holding the shift key down on your keyboard click: update and restart.

Step 4: Select Troubleshoot.

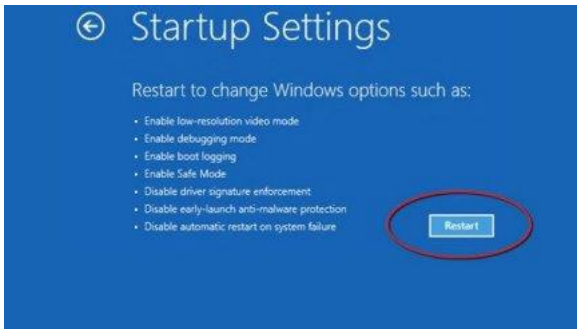


Step 5: Choose Advance Options

Step 6: Select Startup Setting



Step 7: Choose Restart



Step 8: In the next screen; using your keyboard choose number 7) Disable driver signature enforcement and the computer will begin to restart.

Step 9: Once the computer finishes rebooting, go to the device manager where you see “ZENA Wireless adapter” right click on it and update the driver. For this, follow the same steps shown above in windows 7 installation. Now the device will be installed successfully.

The further steps involved in working with WDS will be same as in Windows 7 which was shown above.

1.6.4 Running Demos

Running Demos

Two demos are provided to demonstrate the simplicity and functionalities of MiWi™ Development Environment.

1.6.4.1 Simple_Example_p2p

Simple Example P2P

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 have been demonstrated:

• Establish connection automatically between two devices

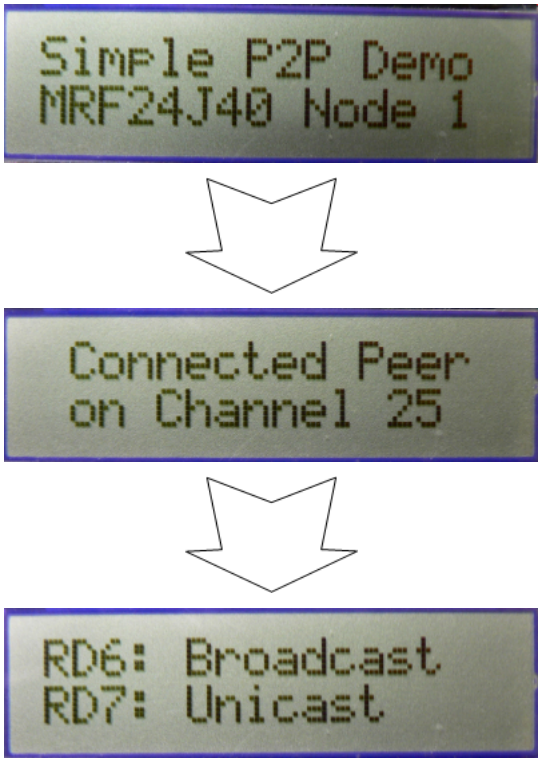
• Broadcast a packet

• Unicast a packet

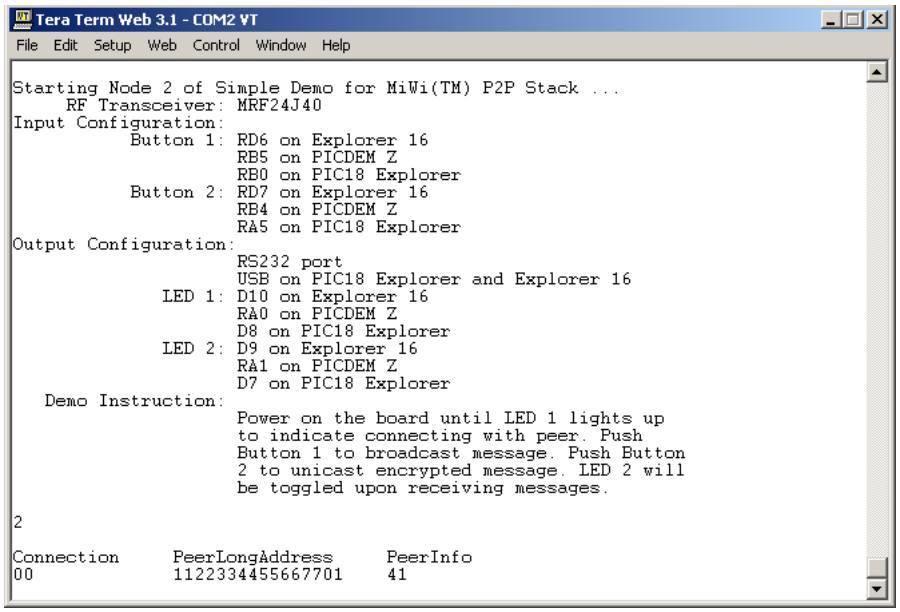
• Apply security to the transmitted packet

To run the simple example application, following is the instruction:

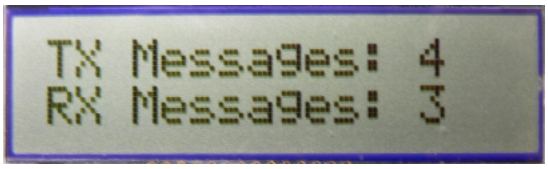
1. Program node 1 and node 2 with proper firmware. We assume that the users are familiar with Microchip tool chain and have no problem compile and program the firmware to the demo boards.
2. Power on node 1 and node 2 respectively
3. Wait a few seconds, until the first LED (RA0 on PICDEM Z, D8 on PIC18 Explorer or D10 on Explorer 16) on both nodes light up. These are the steps to establish connections between two devices.
   - 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 PIC18 Explorer, 8-bit wireless demo board or Explorer 16 demo boards, 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: button 1 for broadcast and button 2 for 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.

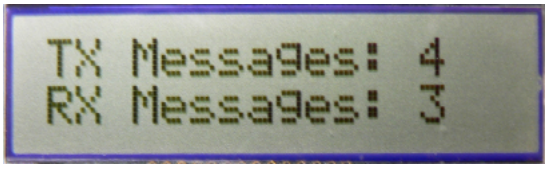


4. Press button 1 (RB5 on PICDEM Z, RB0 on PIC18 Explorer or RD6 on Explorer 16) on one node will toggle the second LED (RA1 on PICDEM Z, D7 on PIC18 Explorer or D9 on Explorer 16) on the other node
   - This shows how a broadcast packet has been transmitted.
   - If the demo is running on PIC18 Explorer or Explorer 16 demo board, the total number of transmitted and received messages will be shown on the LCD.



- 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 button 1 continuously on one end will display the complete bit map of "MiWi".

5. Press button 2 (RB4 on PICDEM Z, RA5 on PIC18 Explorer or RD7 on Explorer 16) on one node will toggle the second LED (RA1 on PICDEM Z, D7 on PIC18 Explorer or D9 on Explorer 16) on the other node.
   - This shows how an encrypted unicast packet has been transmitted and decrypted by the radio after it is received. For the details of how MiWi™ P2P handles encryption, please refer to section “Security Features” in application note AN1204 “Microchip MiWi™ P2P Wireless Protocol”.
   - If the demo is running on PIC18 Explorer or Explorer 16 demo board, the total number of transmitted and received messages will be shown on the LCD.



- 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 secured unicast packet source address, signal strength and the packet payload. The packet payload should have been decrypted by the receiving device. The packet payload is the one line of bit map of “DE”. Press the button 2 continuously on one end will display the complete bit map of "DE".

1.6.4.2 Simple_Example_Star

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 have been demonstrated:

• Establish connection automatically between two devices

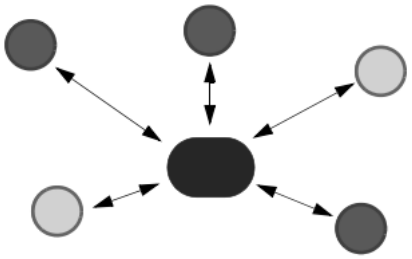
• Broadcast a packet

• Unicast a packet from one node to another through Pan Coordinator.

• Apply security to the transmitted packet.

Star Network is supported on 5 different Hardware Development Boards and works the same way as discussed below for all the configurations.

Star Network Configuration



DEMO OPERATION

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

Simple STAR

MRF24J40 Node

2.The LCD screen displays the operating channel

Connecting Peer

On Channel XX

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

PC :RB0 or RB2

To Broadcast

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

Connected Peer

On Channel XX

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.

RB0: Unicast

RB2: Next Node

Pressing RB0 sends the unicast message.

6. Pressing RB2 push button on the Node, 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:

RB0: 00-XXXXXX-me

RB2: Next Node

or

RB0: 00-XXXXXX

RB2: Next Node

7. Now if RB0 push button is pressed at End Node, a unicast message is sent to PAN Coordinator in the case of RB0:xx-xxxxxx-me or to the destination node as indicated by RB0: 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 is displayed.

At the source End Node the LCD will display:

TX: xx , RX: yy

Message Count

At the Destination Node:

TX: xx , RX: yy

Message Count

After One second the display changes to

Data Packet from

Address: XXXXXX

8. If RB2 push button is pressed at End Node, the LCD displays the next node present in the connection table.

RB0: 01-XXXXXX

RB2: Next Node

Note: At End Nodes, RB0 push button is used to unicast message to the selected node. RB2 push button is used to change/select the other destination node of the unicast message.

9. Press RB0 or RB2 push button at PAN CO to broadcast message to all the End nodes in the network. The LCD displays the incremented TX value.

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 back to following messages on the LCD Display

At PAN CO:

PC :RB0 or RB2

To Broadcast

At Source or Destination End Nodes:

RB0: Unicast

RB2: Next Node

If power is recycled on an End Node, the description given below may be used to restore the node onto the network using the “Freezer” option.

11. On power recycle the End Node. The LCD will display

Simple STAR

MRF24J40 Node

Followed by this:

RB0:New Network

RB2:Freezer

Now press the RB2 push button to retrieve the previous network information. Next the LCD display will be update to:

On the PAN Coordinator:

PC :RB0 or RB2

To Broadcast

On the End Node:

RB0: Unicast

RB2: Next Node

The next steps for sending messages will be similar to the steps discussed above.

Index

T

U

W