2014 Taiwan RTC WiFi Training

WF002

Adding WiFi Dalalink to Embedded Systems

June, 2014

Agenda

- What is WiFi?
- Microchip WiFi Products
- Wi-Fi® Networking with RN WiFly Module
- Lab 1 Command mode and Data mode
- Lab 2 UDP and Device Discovery
- Lab 3 TCP
- Lab 4 Soft AP mode
- Lab 5 APP1632 PIC32 MCU board with RN-171
- Real world issues and tricks to put RN WiFi modules in customer's design

Objectives

- **Develops an understanding of Wi-Fi® technology for embedded system**
- **Understands Microchip's Wi-Fi ® solution offerings and their capabilities**
- **Hands on experience with RN-171-PICTAIL board**
  - Configure the RN171 module via Command mode
  - Create a TCP link between a RN171 and a mobile device
  - See TCP link data between a RN171 and a mobile device
  - Link a RN171 to another RN171 to do a wireless cable replacement
- **Writes code to send data from a PIC32 to a PC, iPhone/iPad or Android phone/Pad via RN-171-PICTAIL Wi-Fi module**
- **Knows how to integrate RN131/171 to your host system**

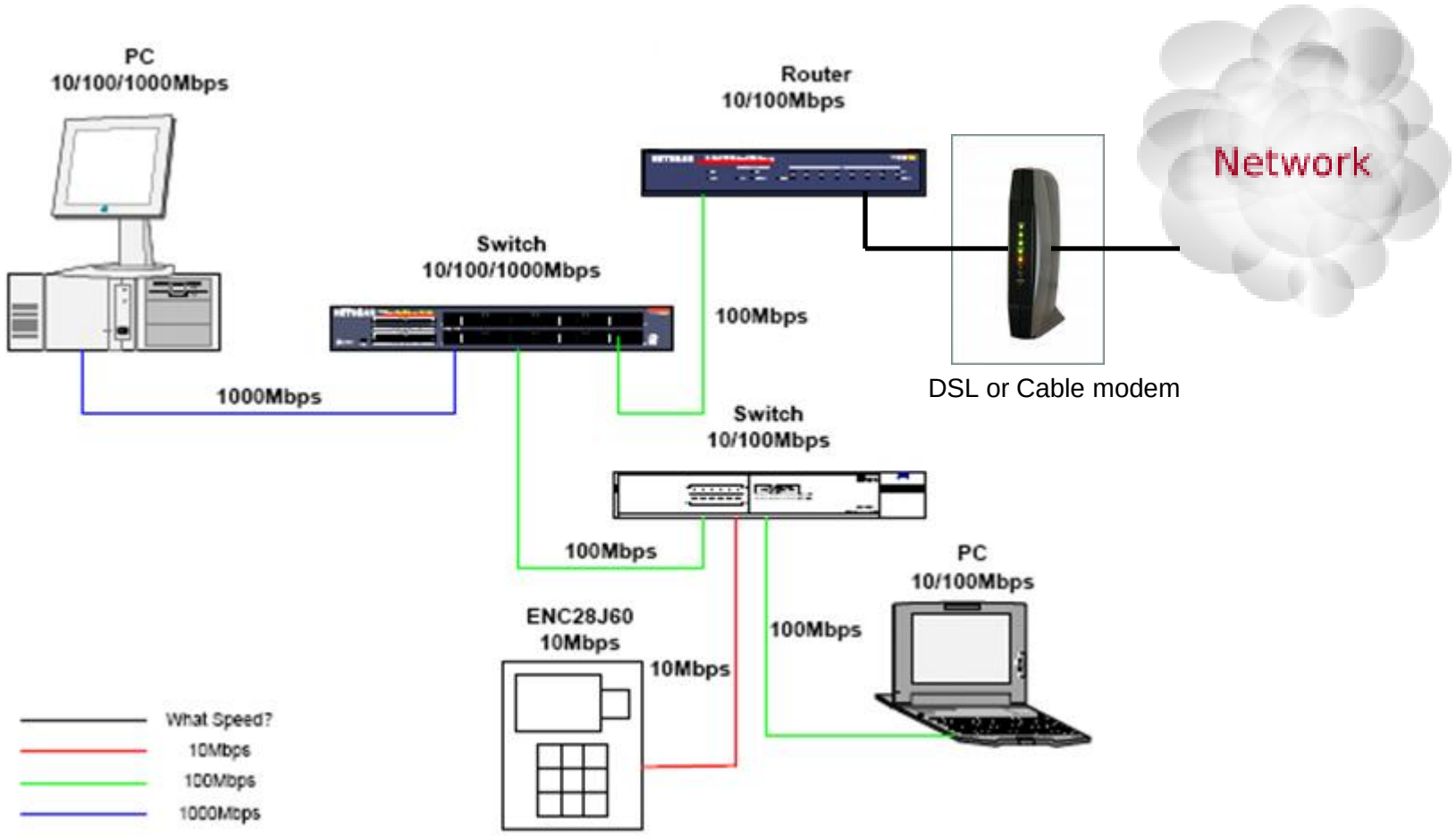
Supporting the Internet of Things
With Embedded Wi-Fi

WiFi™ Overview

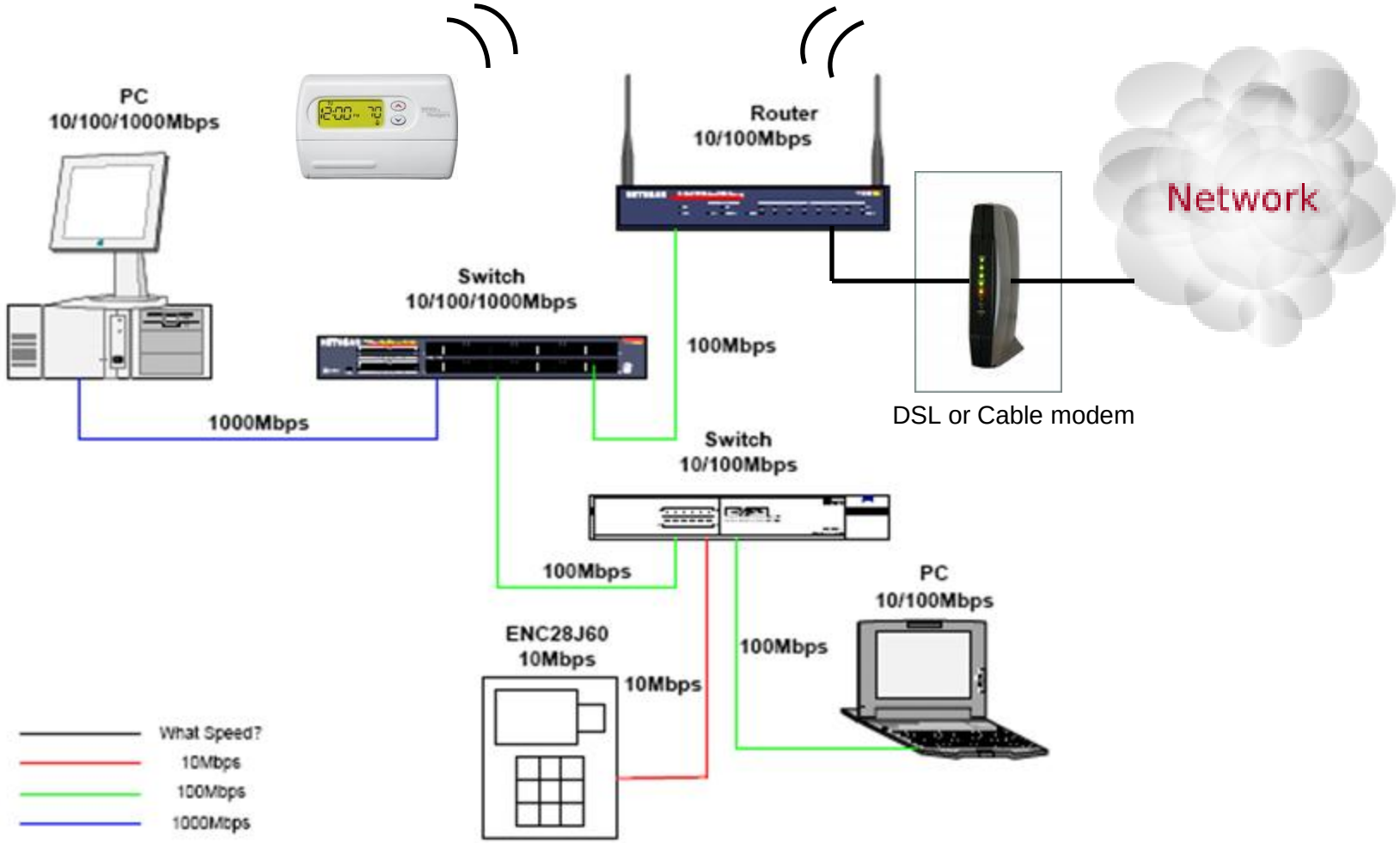
What is Wi-Fi®?

- **802.11/Wi-Fi is wireless Ethernet**
  - Adds mobile internet connectivity
  - Last “100m” connection
  - Removes the wire, but retains the LAN, WAN, WWW connection

Wi-Fi® is an Extension of Ethernet



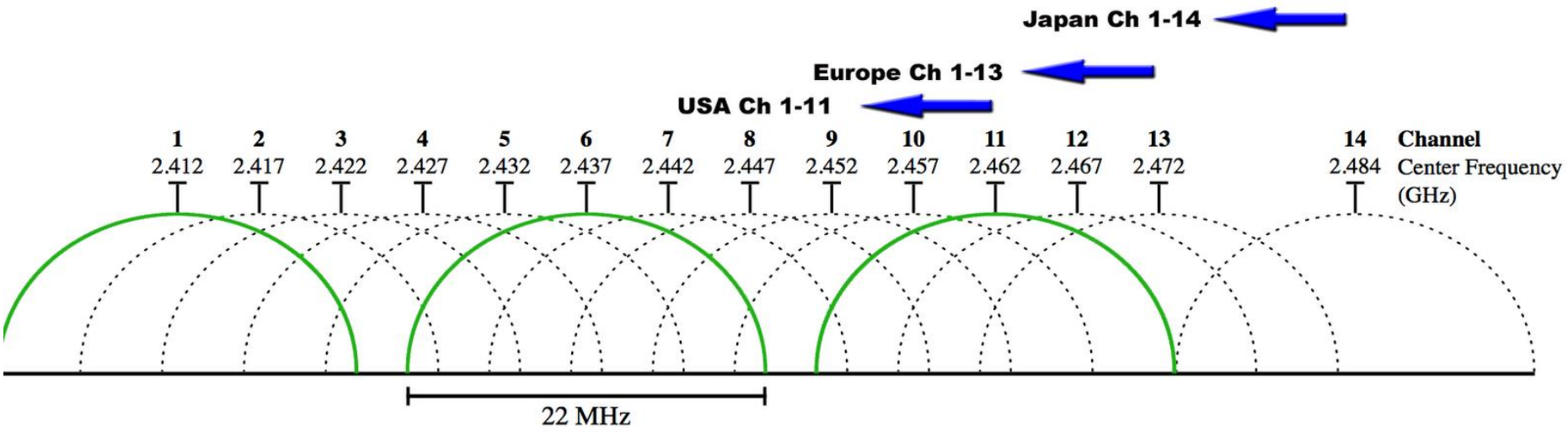
Wi-Fi® is an Extension of Ethernet



802.11b/g/n 2.4GHz Channels

- **802.11b/g/n (ISM band): 2.4 GHz - 2.485 GHz spectrum divided into 14 channels at different frequencies**
- **Geographical location dictates which channels can be used:**
  - US: Channel 1-11
  - Europe: Channel 1-13
  - Japan: Channel 1-14
- **Access Point (AP) administrator chooses channel for AP**
  - Interference possible: channel can be same as that chosen by neighboring AP!
- **Channel 1, 6, 11 typically chosen due to non-overlapping spectrum**

802.11b/g/n 2.4GHz Channel Spacing



- Channels 1, 6 and 11 are non overlapping channels in US
- Channels 3, 8 and 13 are non overlapping channels in Europe

802.11 a/b/g/n Explained

802.11 Protocol	Frequency	Modulation	Bandwidth	Data rates (Mb/s)	# MIMO streams	Comments
a	5GHz	OFDM	20 MHz	6, 9, 12, 18, 24, 36, 48, 54	1	High freq. reduces effective range
b	2.4GHz	DSSS	20 MHz	1, 2, 5.5, 11	1	Many IT departments are turning off “b” access points
g	2.4GHz	OFDM & DSSS	20 MHz	6, 9, 12, 18, 24, 36, 48, 54	1	Only universal module scheme. Access points auto-adjust rate to minimize packet error rate
n	2.4GHz & 5GHz	OFDM	20 MHz & 40MHz	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2 (per stream)	4	Must implement MIMO and 40 MHz bandwidth to get max. data rates (600Mb/s)

802.11a is used to isolate networks and avoid crowded 2.4GHz spectrum

- For example – hospitals and patient records

802.11g are energy efficient radios and fully compatible with **802.11n** networks.

- Same modulation technique

802.11n is useful for high data throughput applications

- High definition video, moving Mbytes of data.
- Must implement **M**ultiple **I**nput **M**ultiple **O**utput (MIMO), to achieve higher rates
- Most embedded device client applications do not need 802.11n

Why IEEE 802.11n?

- **So why are all of our customers asking for 802.11n?**
  - Do they really need a high-speed T-stat?
  - Are they trying to switch their toaster on or off faster?
- **Major driver**
  - Because it's the latest released spec
  - AP compatibility
  - Competition is offering

802.11n Defined

- **802.11n is the current production generation**
- **802.11n uses the same modulation OFDM as 802.11g**
- **802.11n adds additional symbols and optionally, multiple transceivers (MIMO) – high bandwidth**
  - Increased channel bandwidth (40Mhz vs 20Mhz)
  - Shorter guard intervals
  - Result = increase bandwidth from 54mbps to 600mbps.
- **802.11n basic rates of 65 and 72mbps are not affected by 802.11g radios.**

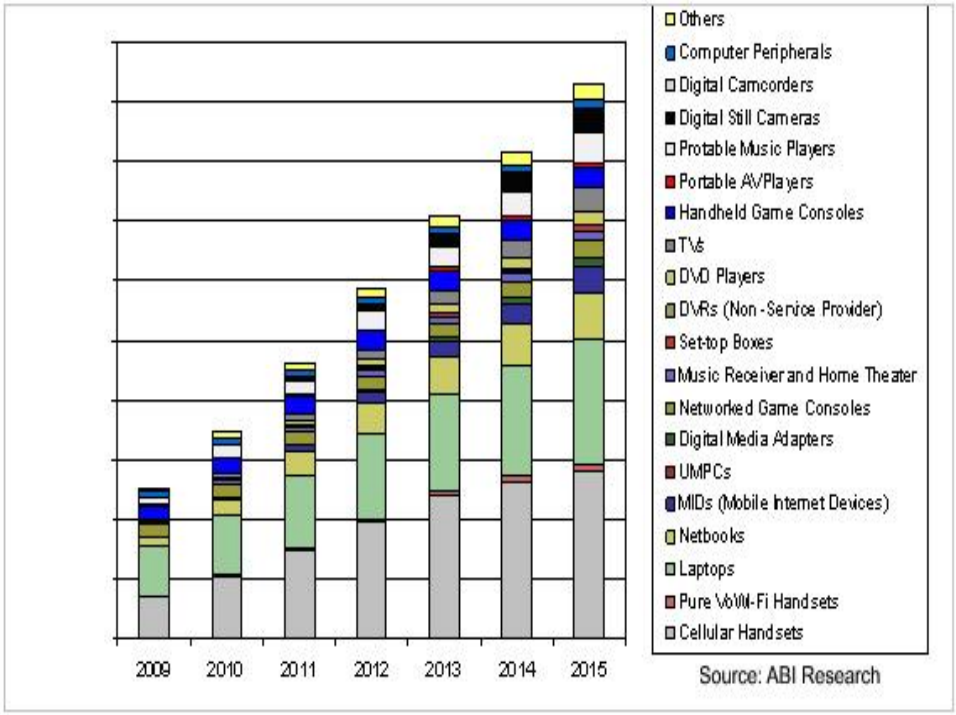
Wi-Fi® a Growing Market

Wi-Fi is a Pervasive Technology

- The Internet is a global standard
  - Largest consumer wireless datacom network
- Billions of connected computers and smartphones
  - All with Wi-Fi / Internet access

Our Target Market is <u>Embedded Connected Clients</u>

- Embedded, not computers or tablets
- Client nodes, not access points
- Examples
  - Building, Industrial and home Automation
  - Sensors
  - Location tracking
  - M2M – Machine to Machine control
  - Medical devices
  - Automotive diagnostic



Microchip Wireless Portfolio

Microchip has a corporate-wide commitment to wireless

Technology	Protocol	Markets	Market Driver	Network Stack	Freq.
Embedded Wi-Fi	IEEE 802.11	Widespread Commercial Industrial	Internet	TCP/IP	2.4GHz
Embedded Bluetooth	IEEE 802.15.1	Widespread Commercial Industrial	Smartphone	BT v2.1, BT Audio, BTLE	2.4GHz
Wireless One Way	Proprietary	Vertical Security, Sensors, Remotes	Cost	MiWi + Keeloq	Sub-1GHz
Wireless Two Way	Proprietary or IEEE 802.15.4	Vertical HA, SEP, Sensors	Cost Local Network	MiWi, BT, ZigBee, RF4CE,	Sub-1GHz and 2.4GHz

Bluetooth® Wi Fi™ ZigBee® Alliance MiWi™ Your wireless connectivity made simple. ROVING networks

Bluetooth, WiFi, or MiWi/Zigbee

	WiFi	Bluetooth	MiWi/Zigbee
Embedded Connected Clients	Best fit	Doable but not as straight-forward as WiFi. Needs a Gateway.	Needs a Gateway device to connect to Ethernet or WiFi to get on internet
Smartphone accessories	1. Can connect to both internet and smart phone 2. Can get around iAP	Lowest integration effort to connect to any smart phone	Need a gateway to go to smart phone
Simple cable replacement / Serial port emulation	Doable but not as straight-forward as Bluetooth	This is what Bluetooth is designed for	1. Can easily do cable replacement 2. Can not do serial port emulation on PC/Smart Phone
Wireless sensor	The data are on internet as soon as the WiFi is connected. Perfect for Cloud application.	1. 7 nodes in the network in ideal condition 2. Need middle ware to get data on to internet	1. Needs a gateway to be on internet 2. Can support a lot of nodes
No. of nodes	20-30 nodes for an AP. Up to 253 in a subnet. No limit for general IP network.	Usually 1 to 1. Could be up to 7 if the host device is powerful enough and data rate is low.	8k on MiWi Pro; 64k on Zigbee
Noise immunity	Good	Excellent	Good. Can be improved by user
Cost	Reasonable	Medium	Low
Range	180m for RN-171 and 200m for RN-131	30m for RN-42 and 100m for RN-41	30-50m without PA; 100m+ with PA

Supporting the Internet of Things
With Embedded Wi-Fi

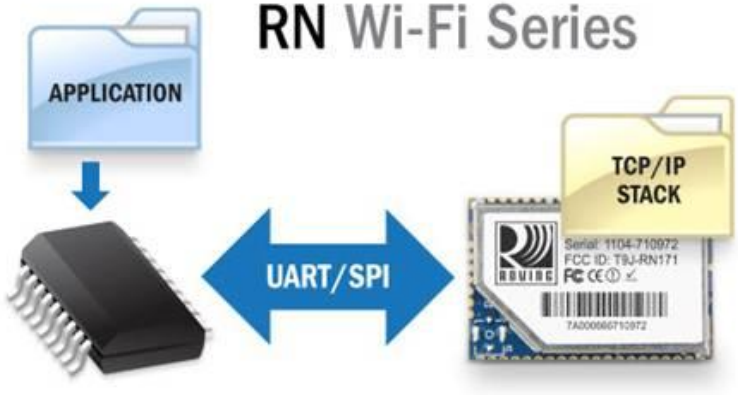
Microchip Wi-Fi™ Products

Microchip Wi-Fi® Modules

	MRF24WB0MB	MRF24WG0MB	RN171	RN131 C/G
802.11 Radio	b	b/g	b/g	b/g
Tx Power	+10dBm	+18dBm	+12dBm	+18dBm
Power Consumption	250uA power save 85mA Rx 150 max Tx	4mA power save 95mA Rx 240 max Tx	4uA sleep 35mA Rx 185 max Tx	4uA sleep 40mA Rx 200 max Tx
Antenna	u.FL / PCB	u.FL/PCB	RF pad	Chip/u.FL
Stack	On PIC MCU	On PIC MCU	Integrated	Integrated
MCU Support	16/32 bit PIC® MCU	16/32 bit PIC MCU	Any 4/8/16/32 bit	Any 4/8/16/32 bit
Certifications	FCC/IC/EN Wi-Fi Alliance	FCC/IC/EN Wi-Fi Alliance	FCC/IC/EN KC/NCC Wi-Fi Alliance	FCC/IC/EN KC/NCC/Telec Wi-Fi Alliance

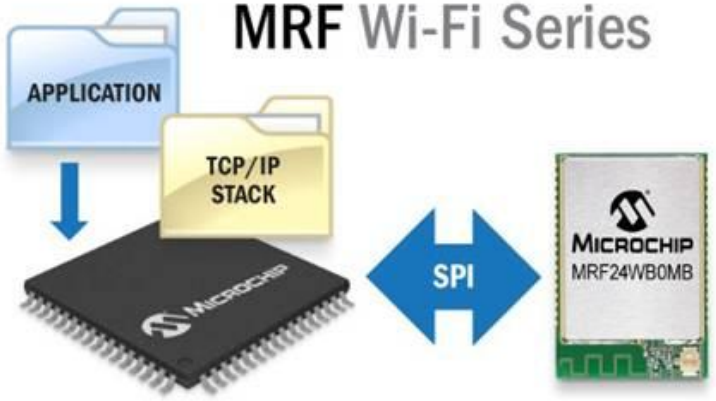
ROVING networks

Wi-Fi® Software Stack



TCP/IP stack on module

- Simple ASCII interface, quick time to market
- Works with any MCU vendor
- Supports ALL PIC® MCU families



TCP/IP stack located in PIC MCU

- Elegant solution for combined Ethernet and Wi-Fi architectures
- Extendable TCP/IP stack for additional services
- For more info attend class: 17049_TCP3

MRF24WG and RN strength

- **MRF24WG strength**
  - 100% customizable since the upper layer stacks are running on PIC
  - Easy integration with PIC
  - High throughput (up to 5mbps UDP)
  - Multiple TCP sockets
  - Supports WiFi Direct, IPv6, SSL, web server and enterprise security
- **RN WiFi strength**
  - Stack on board with easy UART ASCII API
  - Work with any MCU
  - Includes a lot of application layer functions in module (FTP firmware update, HTTP client, UDP beacon, etc…)
  - Super low power consumption, ultra fast connection speed

Get Started with Our Tools

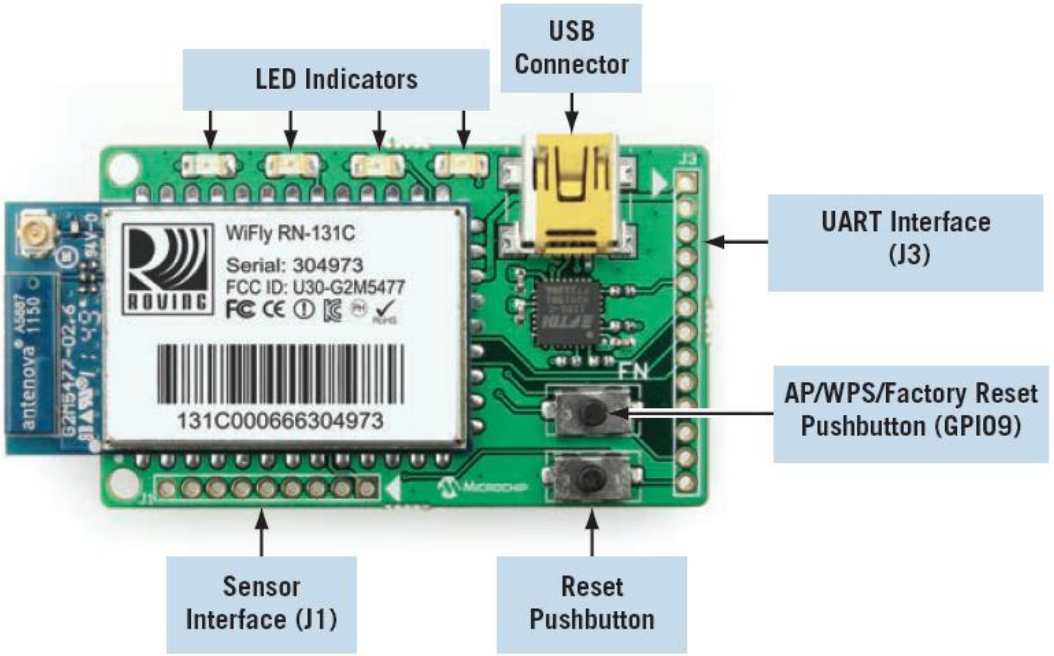
	Explorer Based Development Board				Stand alone Evaluation Kit			
Series	MRF		RN		MRF		RN	
Platform	PICtail/ PICtail+		PICtail/ PICtail+		Wi-Fi Comm Demo		Eval Kit	
Module	MRF24WB	MRF24WG	RN-131	RN-171	MRF24WB	MRF24WG	RN-131	RN-171
Image								
Part #	AC164136-4	AC164149	RN-131-PICTAIL	RN-171-PICTAIL	DV102411	DV102412	RN-131-EK	RN-171-EK
Availability	NOW	NOW	NOW	NOW	NOW	NOW	NOW	NOW

TCP/IP stacks and Application Demos online at: www.microchip.com/mla

RN Series Comparison

Parameter	RN-171	RN-131
Transmit Power	Configurable: 0dBm to +12 dBm	Fixed: +18 dBm
Antenna Options	Chip, PCB trace, Wire, U.Fl connector	Chip, U.Fl connector
Power Consumption	Sleep:4uA Rx: 30mA Tx: 185mA at +10dBm	Sleep: 4uA Rx: 40mA Tx: 210mA
Hardware Interface	UART	UART
Range	Up to 150 meters (LOS)	Up to 200 meters (LOS)
FCC, CE, IC certs	Yes	Yes

Eval Board (EK) Architecture



Condition	Blue LED	Red LED	Yellow LED	Green LED
On solid	Unused	-	-	Connected over TCP
Fast blink	Unused	Not associated	Rx/Tx data transfer	No IP address
Slow blink	Unused	Associated, no Internet	-	IP address OK
Off	Unused	Associated, Internet OK	-	-

RN177 reference design

- Low cost 2 layer RN171 reference design
- Close relative of RN174 but cheaper to build
- Available antenna options :
  - PCB
  - UFL
  - Johanson chip antenna
  - SMA connector

- It is intended to be released to customer to save support effort
- Available only in PCB form.

Supporting the Internet of Things
With Embedded Wi-Fi

Wi-Fi® Networking with WiFly Module For Embedded System

What do you need to know about Wi-Fi® ?

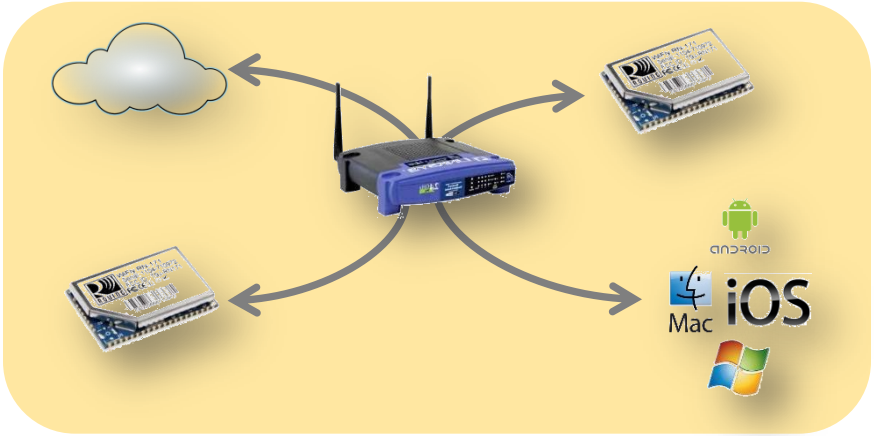
- Types of Network Topologies
- Ways to Provision a device
- Security and Authentication
- Associating to a Wi-Fi Network
- Communicating over Wi-Fi

Wi-Fi® Network Topologies

Network Topology Types

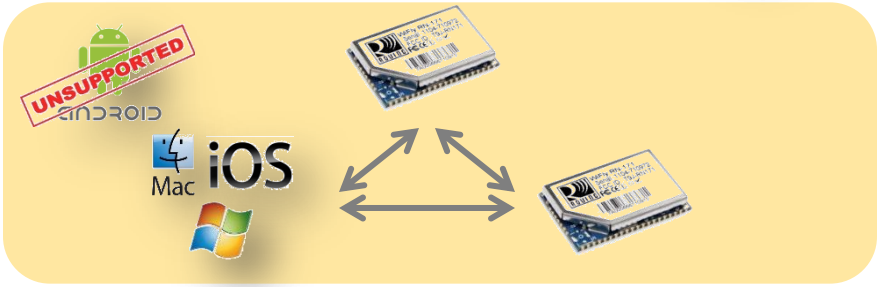
Infrastructure:

- Client nodes communicate via an access point
- Most common, like connecting your PC to a home network



Adhoc:

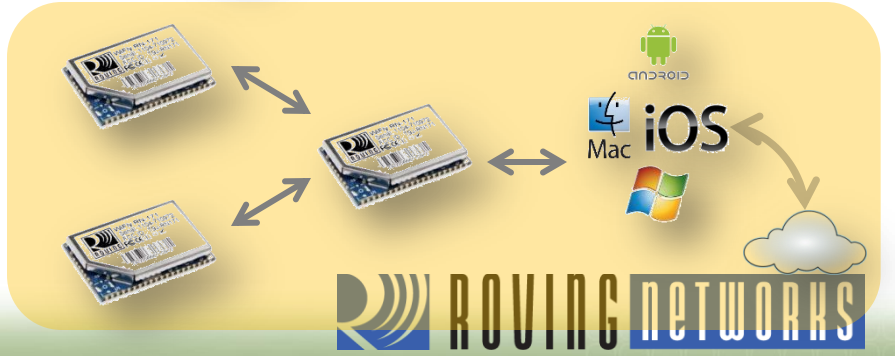
- Point-to-Point connections
- Every node connected to every other node
- Android unsupported



Soft AP:

NEW

- Module looks like an Access Point
- AP module is central coordinator
- Basic network management
- DHCP, routing, gateway redirection

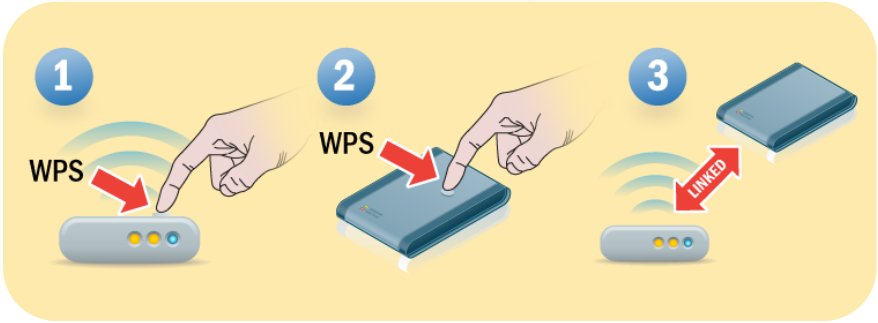


Provisioning a Wi-Fi® Device

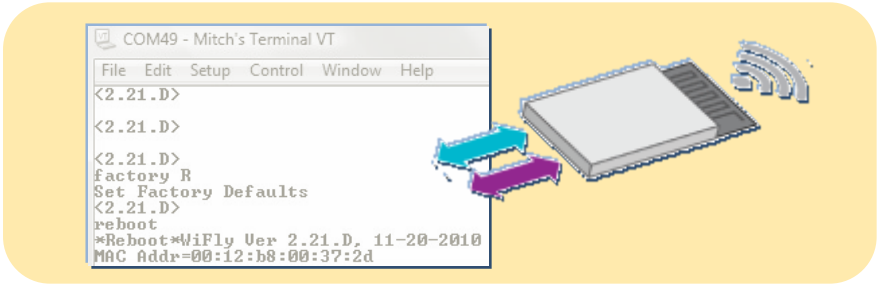
Provisioning Device Clients

Provisioning *(How to connect to a network?)*

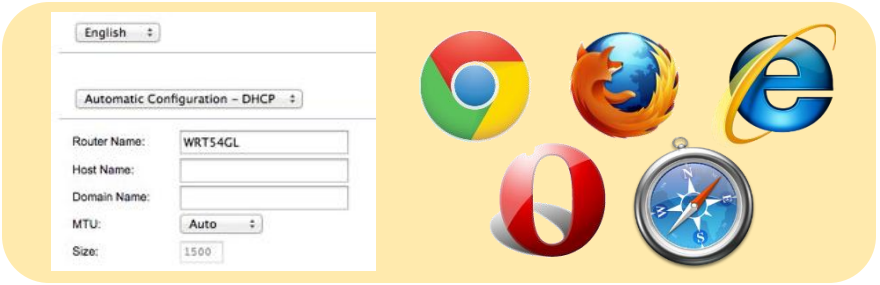
- Wi-Fi® Protected Setup (WPS)
  - No user GUI required
  - No host microcontroller oversight
  - Simple push button user interface



- Command Line Interface (CLI)
  - UART / SPI – Simple ASCII interface
  - Host microcontroller control



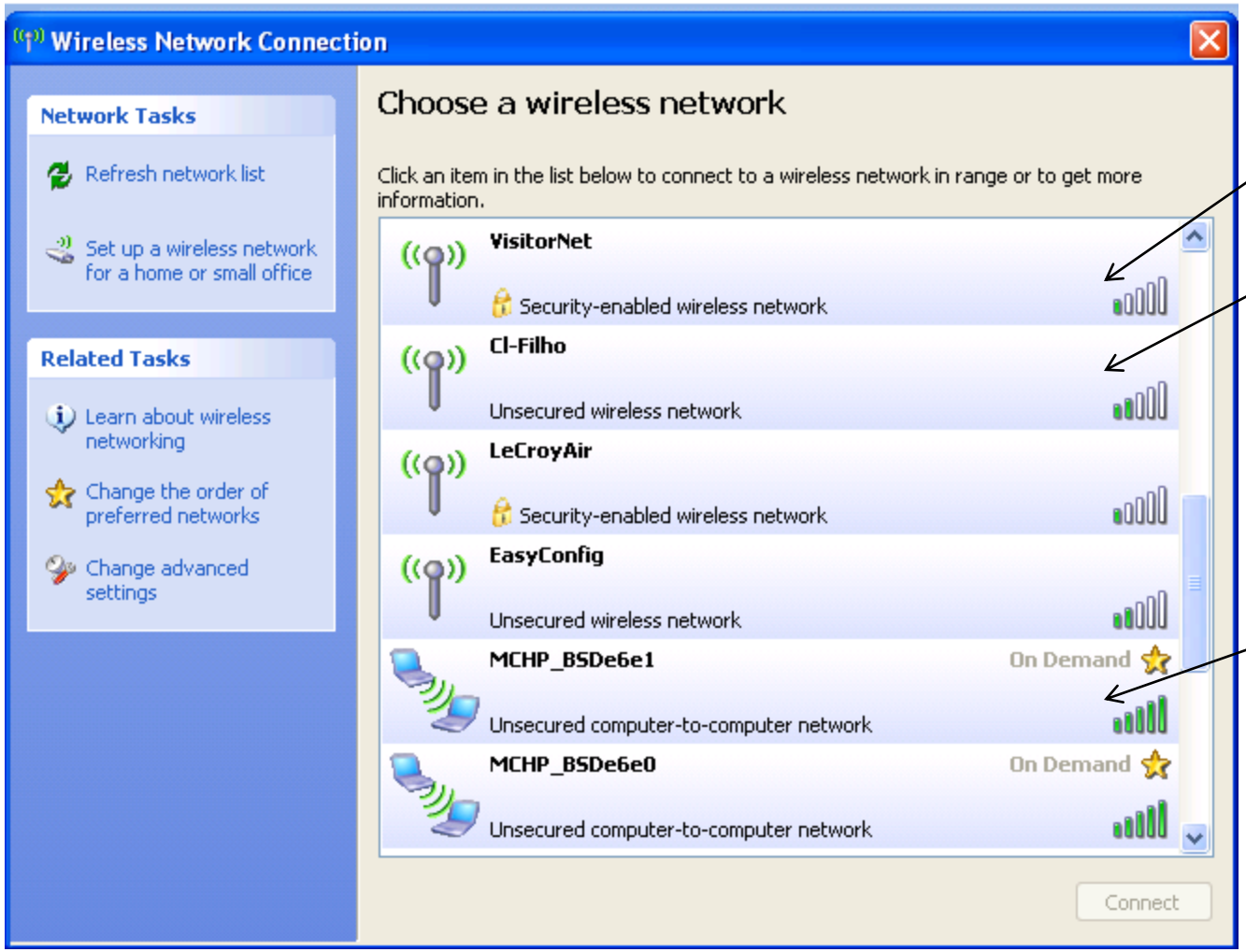
NEW Webserver (http:// port 80)
  - Browser Interface



Security and Authentication

Security and Authentication

SSID or BSSID (Ad hoc) is the name of the network.



Secured infrastructure

Open infrastructure

Open Ad hoc

Security and Authentication

WEP: **Wired Equivalent Privacy**

- 1999-2003, considered obsolete
- Prohibited by '*Payment Card Industry Security Standards Council*' since 2008

WPAv1: **Wi-Fi® Protected Access (v1)**

- A trimmed down 802.11i
- Same hardware as WEP
- Similar to WEP but uses a TKIP end-to-end encryption
- 8-64 Hexadecimal key, longer keys increase complexity
- Not recommended – but reasonable security

WPAv2: **Wi-Fi Protected Access (v2)**

- Requires upgraded hardware
- AES-CCMP algorithm is mandatory 256bit key
- Considered very secure

WPA/WPA2 Enterprise:

- Corporate level security additions to WPA/WPA2
- Complex implementation
- Users are qualified for network infrastructure and domain use
- Considered very secure

WiFly Data Transmission

Data Transmission Options

TCP Service:

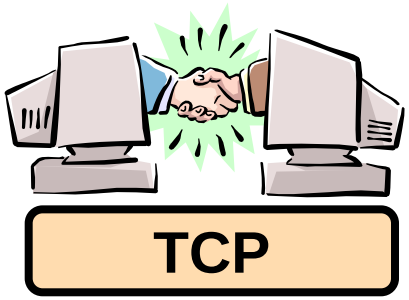
- *Connection-oriented:* setup required between client and server processes
- *Reliable transport best effort data delivery*
- *Flow control:* sender won't overwhelm receiver
- *Congestion control:* throttle sender when network overloaded
- *Does not provide:* timing, minimum throughput guarantees, security

UDP Service:

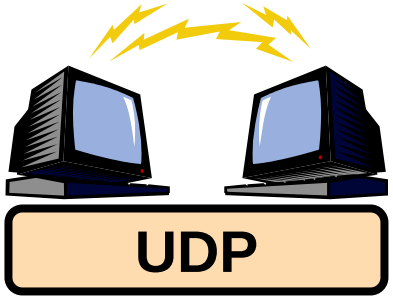
- Unacknowledged transfer service between sending and receiving process
- *Does not provide:* connection setup, guaranteed packet delivery, flow control, congestion control, timing, throughput guarantee, or security

Q: Why bother? Why is there a UDP?

TCP vs. UDP



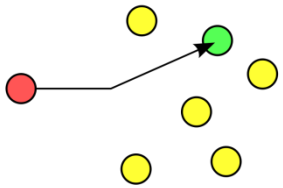
TCP



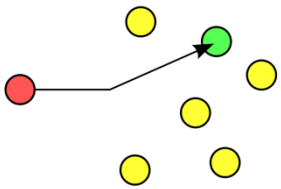
UDP

- Slower but reliable transfers
- Typical applications:
  - Email
  - Web browsing

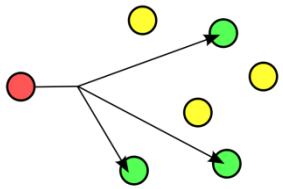
- Fast but non-guaranteed transfers (“best effort”)
- Typical applications:
  - VoIP
  - Music streaming



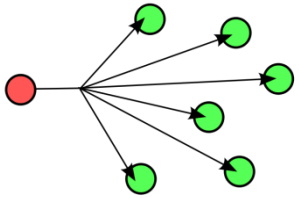
unicast



unicast



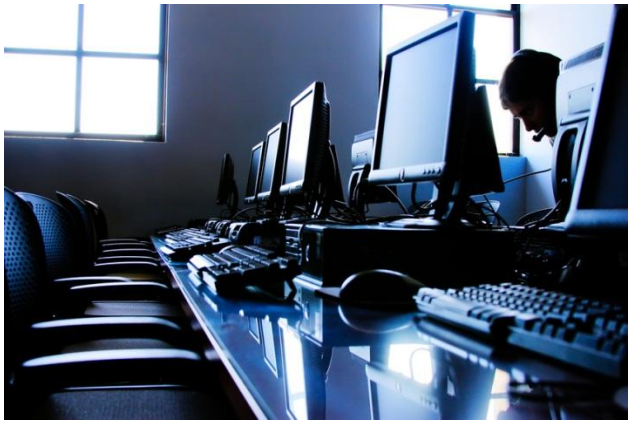
multicast



broadcast

Hands-On Wi-Fi Lab

- **Lab 1: Command Mode vs. Data Mode**
- **Lab 2: Association & UDP**
- **Lab 3: TCP**
- **Lab 4: Access Point Mode**



Lab Prerequisites

- Hardware Resources
  - RN-131-PICTAIL/RN171-PICTAIL evaluation kit
  - Prolific USB to UART cable
  - Configured access point (AP)
    - Security: WPA2-AES
- Software Resources
  - PortPeeker
  - Terminal Emulator
    - TeraTrem
  - Prolific chipset drivers

Access Point Setup

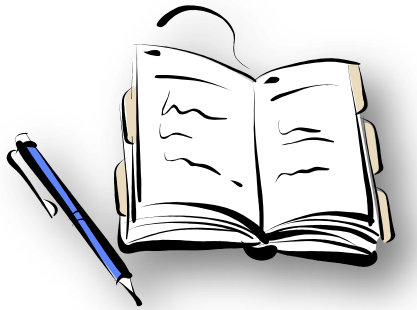
SSID: rtc-class-1
Pass: rubygirl

SSID: rtc-class-2
Pass: rubygirl

SSID: rtc-class-3
Pass: rubygirl

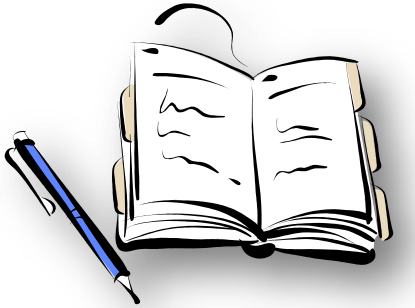
SSID: rtc-class-4
Pass: rubygirl

Lab 1: Command mode and Data mode



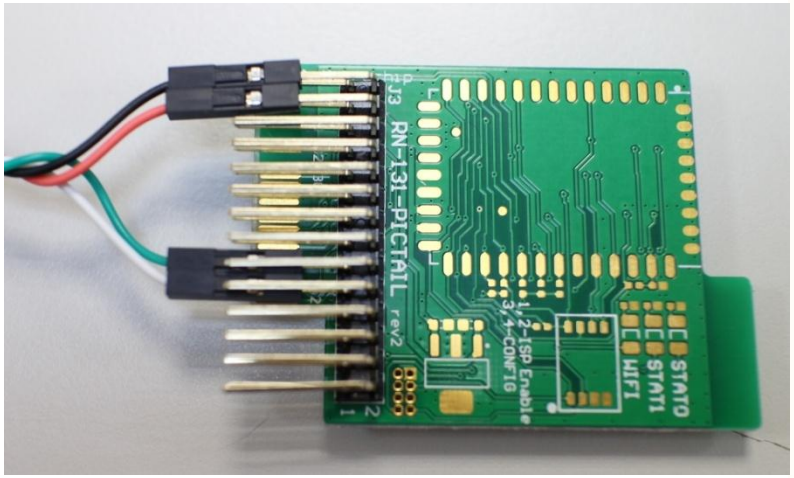
Lab 1: Learning objectives

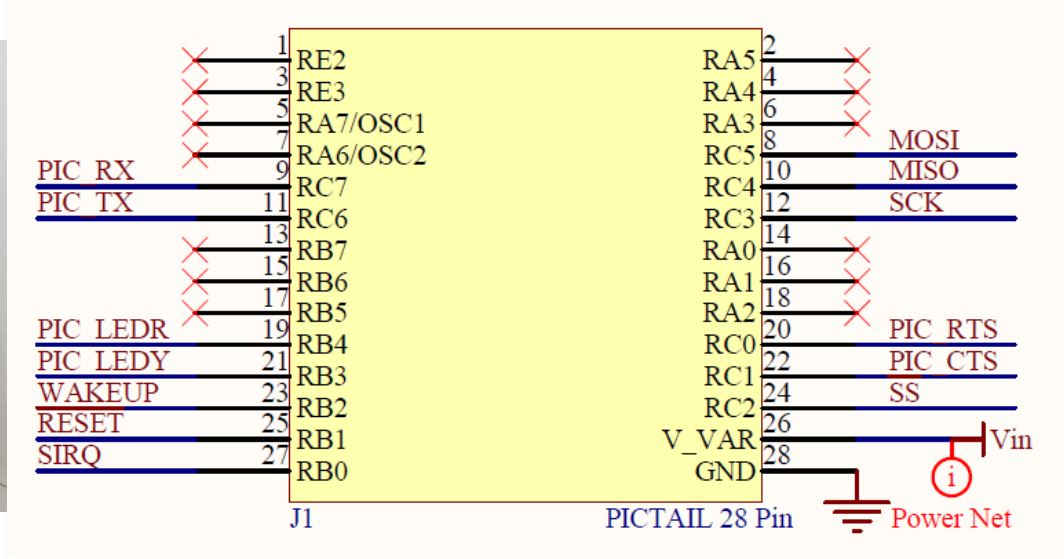
- **By the end of this lab you will be able to:**
  - Connect the evaluation kit hardware to you PC
  - Switch between command mode and data mode
  - View the current settings on the module
  - Check firmware version running on the module
  - Associate the module to an Access Point



Lab1: WiFly Setup

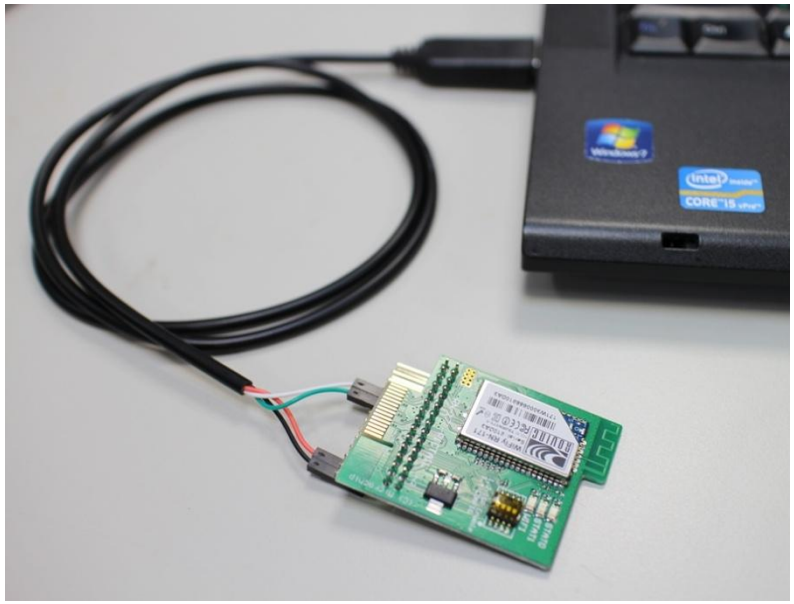
- Connect RN-171-PICTAIL to Prolific USB to UART cable
  - +5V, GND, UART_TX, UART_RX on the Prolific USB to UART cable
  - +5V (red) to pin 26 on PICTAIL
  - GND (black) to pin 28 on PICTAIL
  - UART_TX (green) to pin 11 on PICTAIL
  - UART_RX (white) to pin 9 on PICTAIL

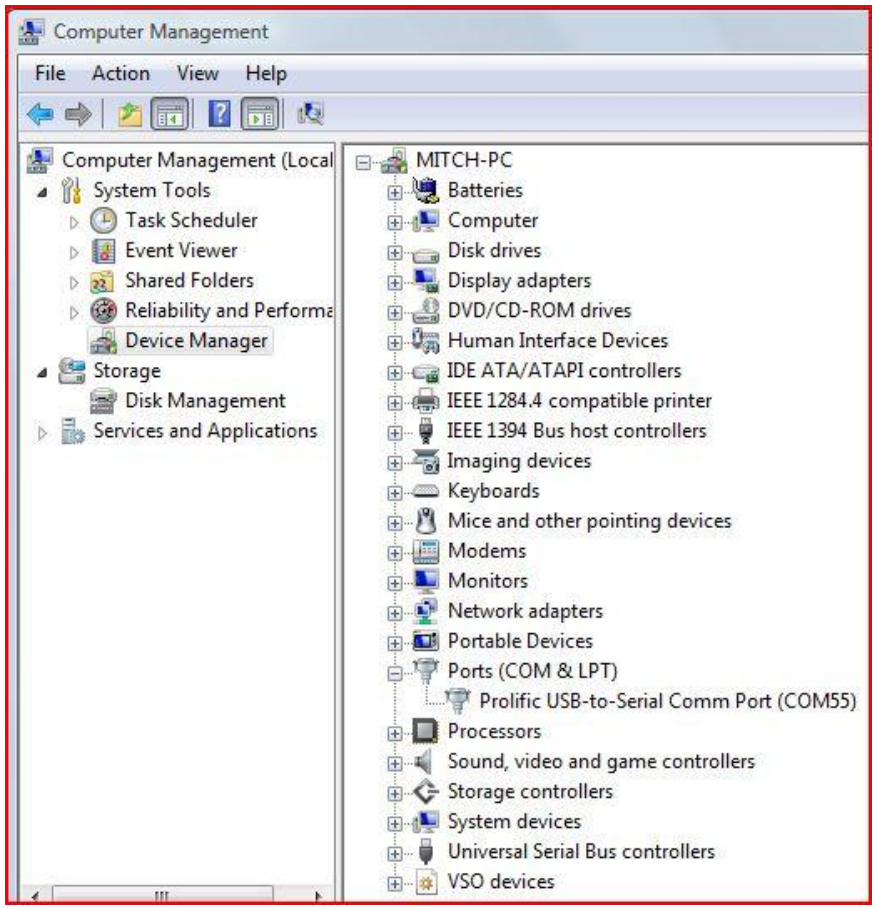




Lab1: WiFly Setup

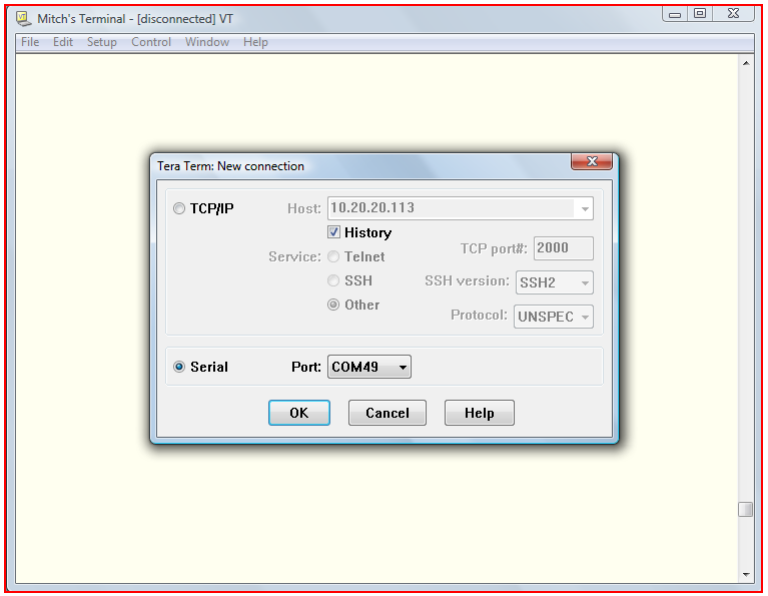
- Connect Evaluation Board
  - Connect USB to UART cable to your computer
  - Use device manager to find the COM port



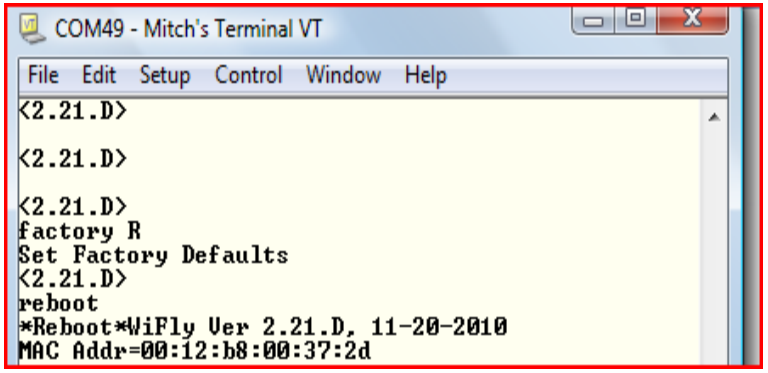


Lab 1: Configure Module via UART

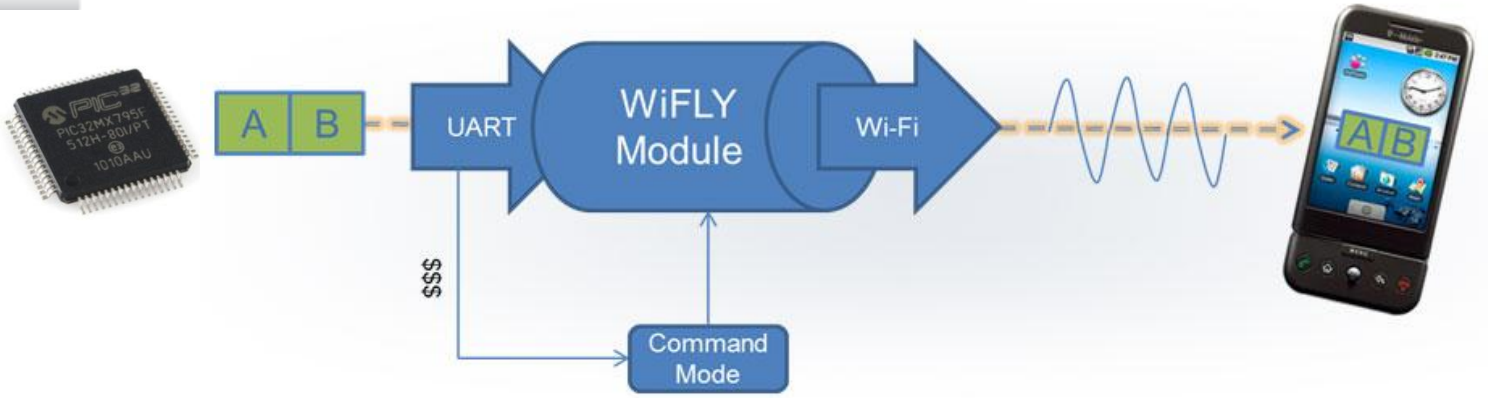
- **Launch Command Mode**
  - Run Tera Term
  - Open assigned COM port
    - Serial port settings: 9600 baud,
      8 bits, No Parity, 1 stop bit, no flow control
  - Type **$$$**
  - Module responds with <CMD>



- **Review & Reset Configuration**
  - Check configuration & firmware version
    - **get e**
    - **ver**
  - Perform factory reset
    (starts module in known state)
    - **factory R**
    - **reboot**



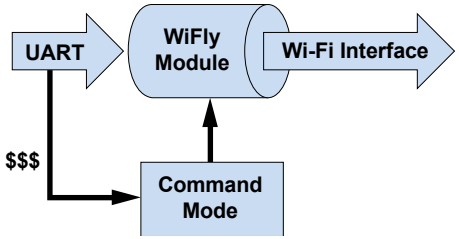
WiFly: Data & Command Modes



- **Data Mode (Default State)**
  - WiFly module like data pipe
  - TCP/UDP header stripped or added, transparent to UART
  - Data written to UART is sent out over Wi-Fi®
  - Data received over Wi-Fi is read from UART
- **Command Mode ($$$)**
  - Special configuration mode entered using $$$
  - Used to set module parameters e.g., SSID, pass phrase, etc.

WiFly: Data & Command Modes

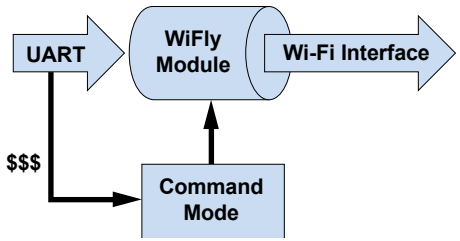
- Performed Using ASCII Commands
  - User Guide
- Configuration Settings Stored in Module's Flash
  - Changes are persistent & reloaded after power cycling
  - Changes kept using **save** command
  - Many settings require reboot to take effect



Command Mode: Syntax

- Commands use keyword followed by additional information
- Command rules
  - case sensitive
  - spaces not allowed, substitute $
    Can use shorthand
    - set wlan ssid my$network

  - **set uart baudrate 115200** valid
  - **set uart b 115200** valid
  - **set u b 115200** valid
  - s uart baudrate 115200 Invalid



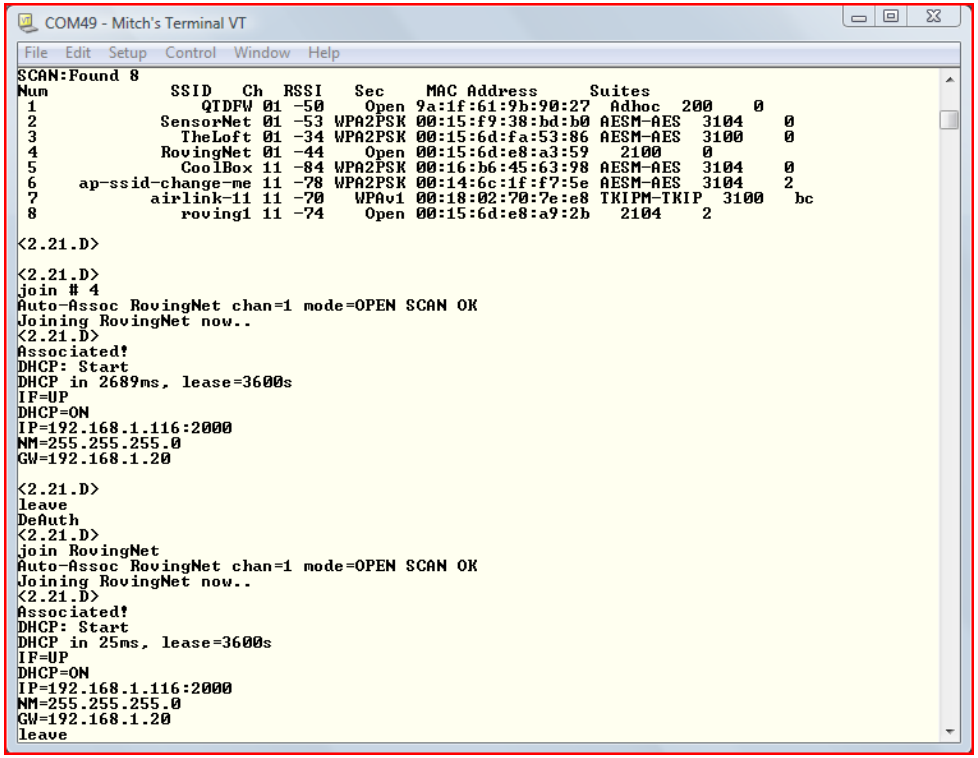
Command Mode: Keywords

Set Command	Function
AdHoc	Controls the ad hoc parameters
Broadcast	Controls the broadcast hello/heartbeat UDP message
COMM	Communication & data transfer, matching characters
DNS	DNS host & domain
FTP	FTP host address & login information
IP	IP settings
Option	Optional & infrequently used parameters
Sys	System settings such as sleep & wake timers
Time	Real-time clock settings
UART	Serial port settings such as baud rate & parity
WLAN	Wireless interface, such as SSID, channel & security options

- Set: Immediate Effect, Permanent If Saved to Configuration File
- Get: Retrieve & Display Permanently Stored Information
- Status: Current Interface Status, IP Address, etc.
- Action: Perform Actions Such As Scan, Connect & Disconnect
- File: Upgrade, Load & Save Configuration, Delete Files, etc.

Lab 1: Joining an AP

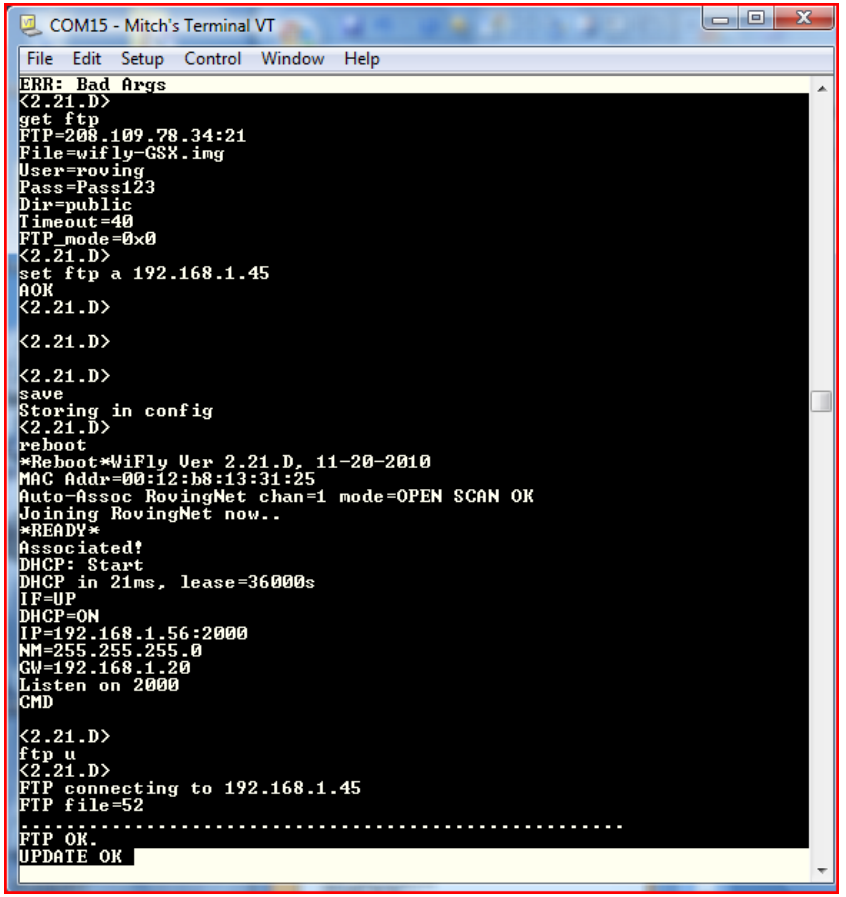
- **Search for Networks**
  - **$$$** (enter command mode)
  - **scan**
- **Join a network**
  - **join** <*string*>
    (e.g., **join rtc-class-1**)
  - **leave**
- **Auto-Join Network with Persistent Configuration**
  - **set wlan join 1**
  - **set wlan ssid** <*string*>
  - **set wlan pass** <*string*>
  - **save**
  - **reboot**



TIP: If Network Is Secure, Set Pass Phrase with set wlan pass <*string*> before Joining Network

Lab 1: FTP Update
(REQUIRES Internet Access)

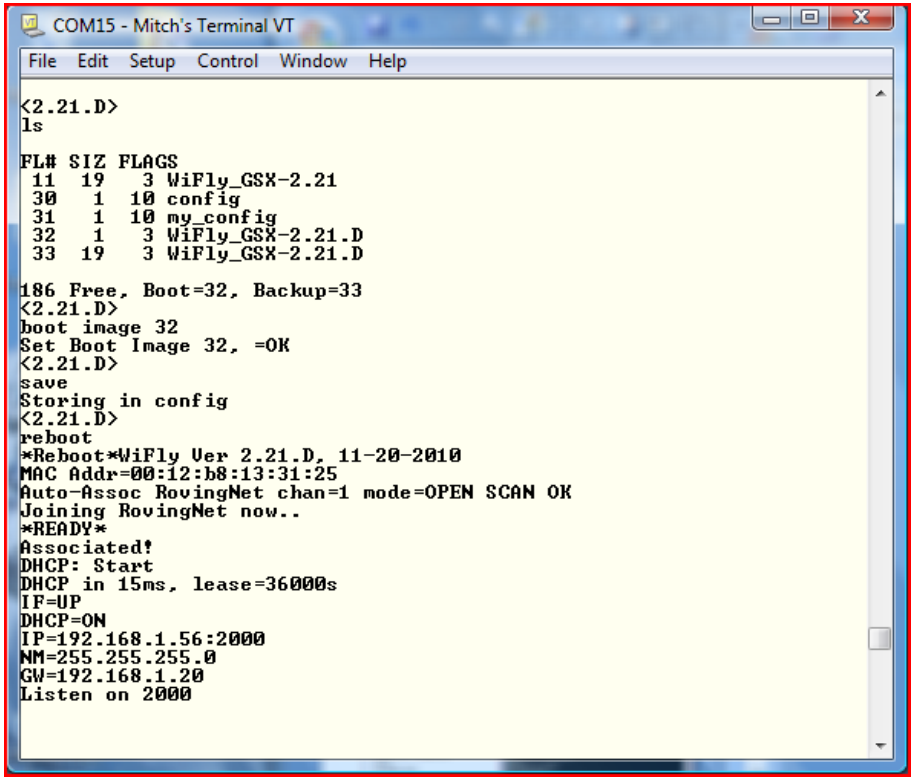
- **Module defaults to connect to microchip's FTP server**
- **Module must be associated to a network with internet access**
- **Use Local FTP Server**
  - Enter command mode
  - **factory R**
  - Associate module with AP
  - **save** & **reboot**
- **Update Firmware**
  - Enter command mode
  - **set ftp address** <*local_FTP_server*>
  - **ftp update**
  - **ver**
  - **reboot**
  - Enter command mode
  - **ver**



NOTE: After Downloading New Firmware, Restore Module to Factory Defaults Before Using It

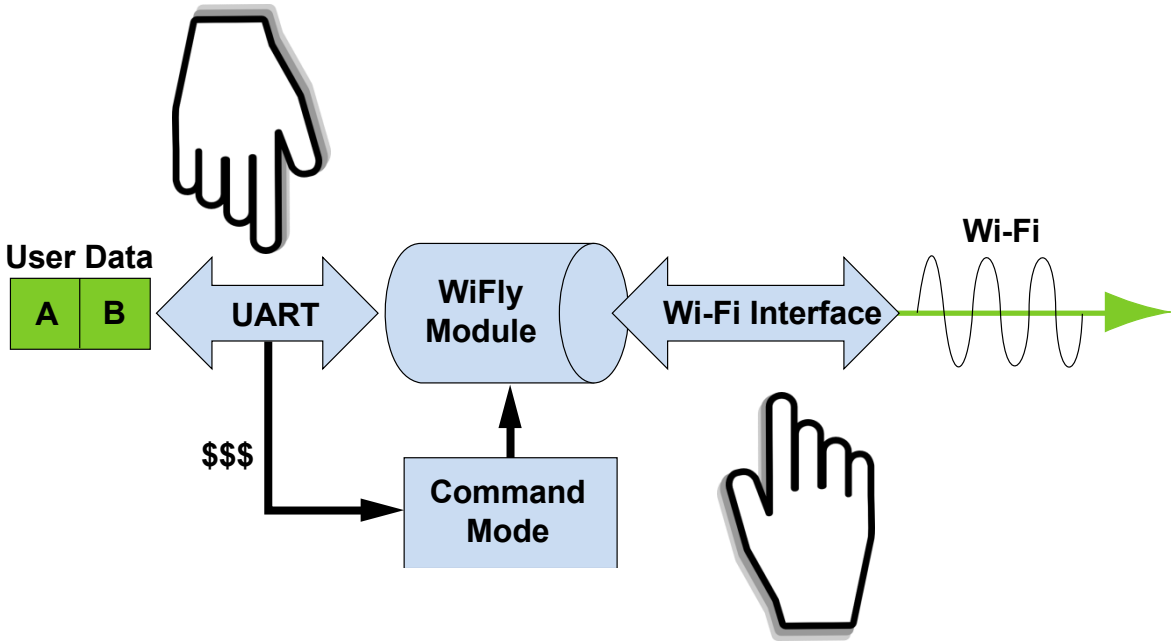
Lab 1: Setting Firmware Boot Image

- **Firmware Stored in Embedded Flash Memory**
- **Boot Image is Firmware Version Module Is Currently Running**
- **After Successful Update, Boot Image Changes to New Firmware File**
- **Change Boot Image**
  - Enter command mode
- **View Files in Flash**
  - Enter command mode
  - **ls**
- **Change Boot Image**
  - Enter command mode
  - **boot image** <*file_name*>
  - **reboot**



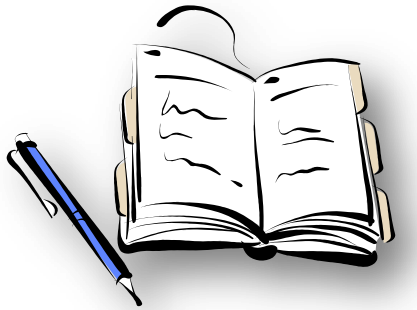
Lab 1: Summary

- Knows how to connect RN-171-PICTAIL
- Configured Module via the UART in Command Mode



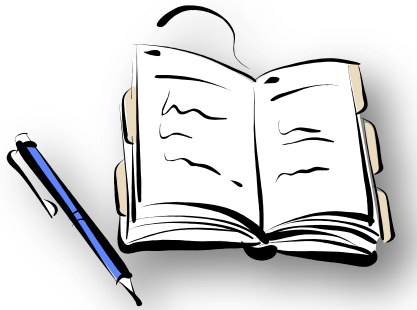
- Scan, join Wi-Fi networks
- FTP firmware update and set boot image

Lab 2: UDP and Device Discovery



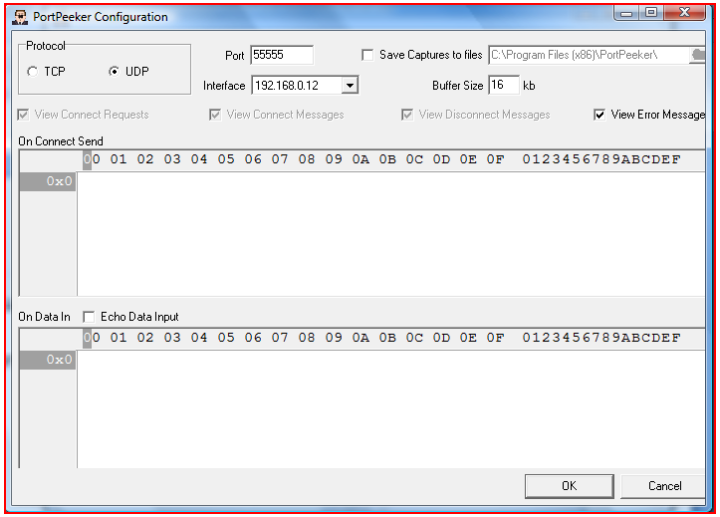
Lab 2: Learning objectives

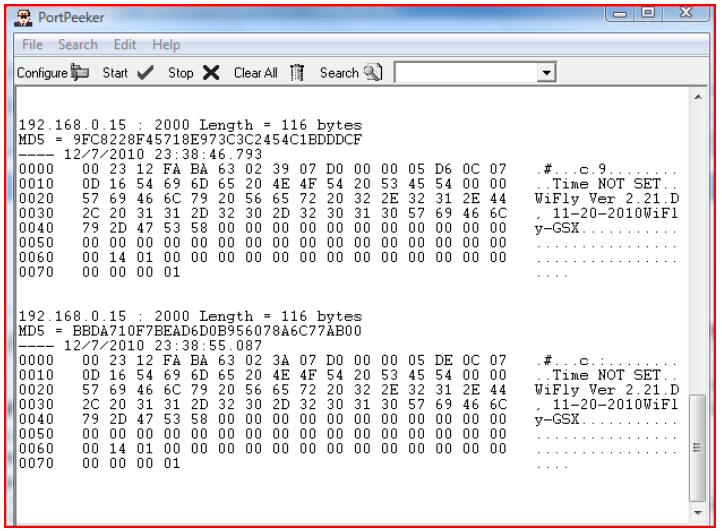
- **By the end of this lab you will be able to:**
  - Discover the WiFly module on your wireless network via UDP
  - Change Device ID and see it on UDP broadcast data
  - Send data via UDP



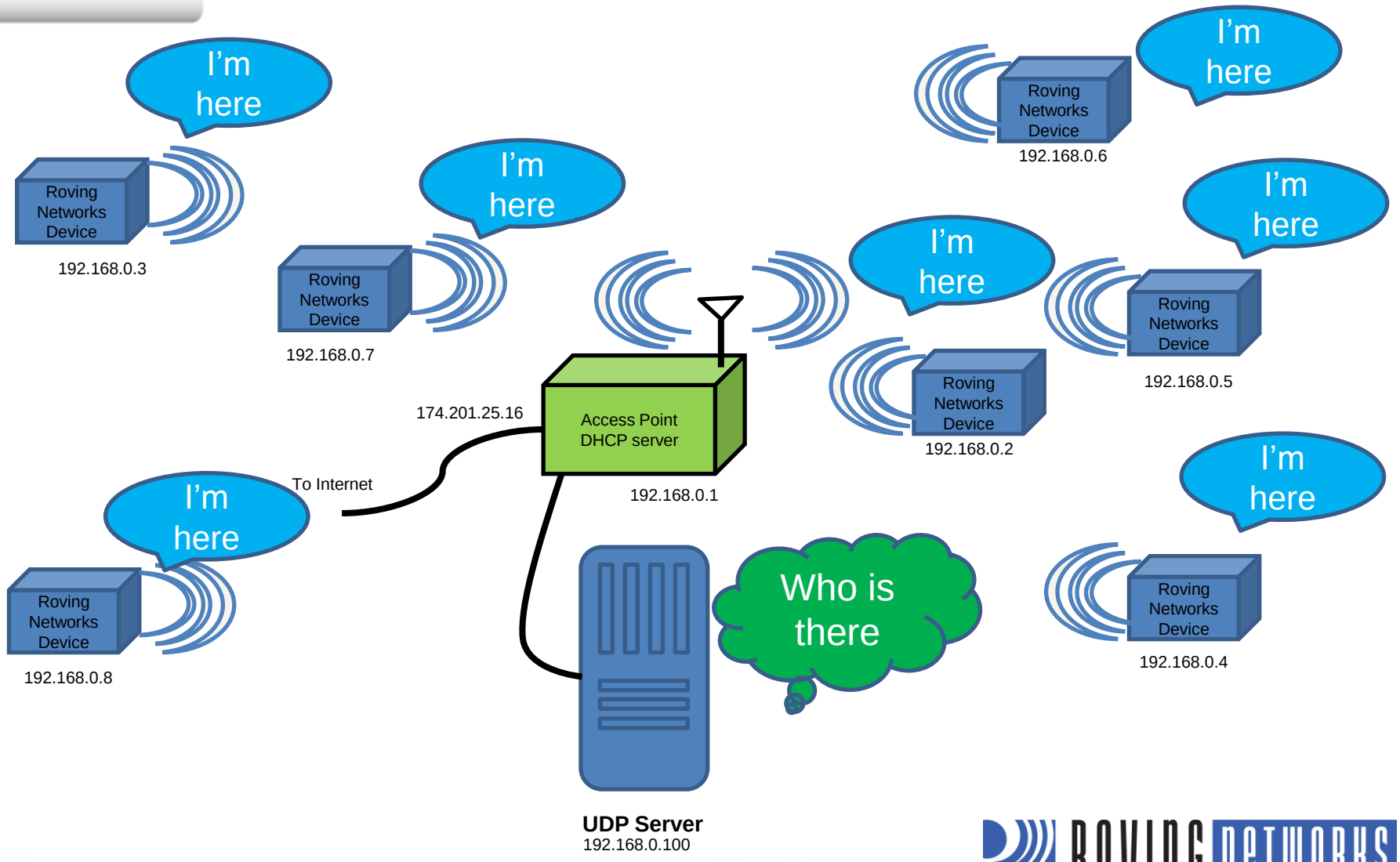
Lab 2: Configure & Capture UDP with PortPeeker

- **Associate PC with Same AP as Module**
  - Enter command mode & retrieve module's IP address
  - **Ensure PC is on same subnet**
- **Launch & Configure PortPeeker**
  - Click **configure**
  - Set port number to **55555** (default)
  - Set protocol to **UDP**
  - Click **OK**
- **Start UDP Packet Capture**
  - Click **Start**
  - If PC & module are on same subnet, broadcast packets shown
  - If multiple nodes on network, look for YOUR IP address



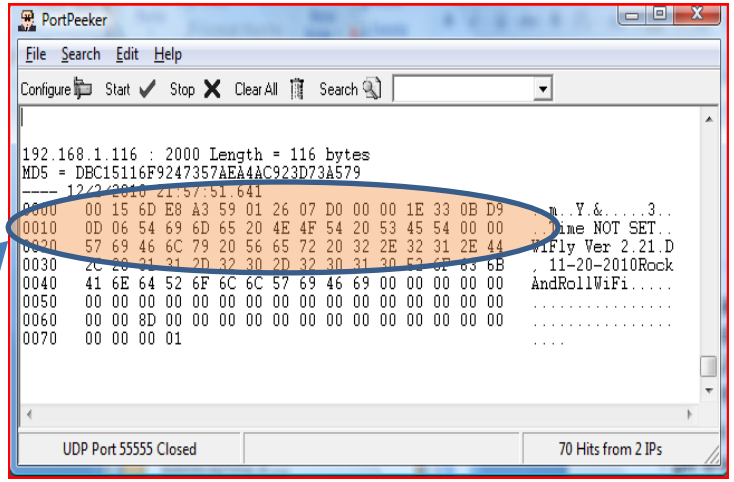


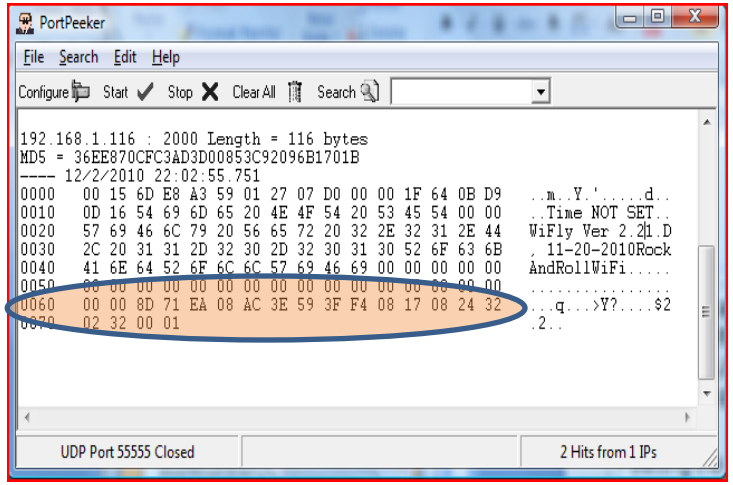
Lab 2: Device discovery via UDP broadcast



Lab 2: UDP Broadcast

- **Module sends UDP broadcast at programmable intervals to make itself discoverable**
- **UDP broadcast contains information that identifies module on network**
- **Set UDP Broadcast Interval**
  - Enter command mode
    - **get broadcast** (observe current interval)
    - **set b i 3** (b=broadcast, i=interval)
    - **save & reboot**
  - Review UDP messages in PortPeeker
- **Enable Sensor Data in UDP Broadcast**
  - Enter command mode
    - **set q s 0xff** (set sensor mask)
    - **save** to make persistent
  - Reboot not required
  - Review UDP messages in PortPeeker
    - Sensor data highlighted

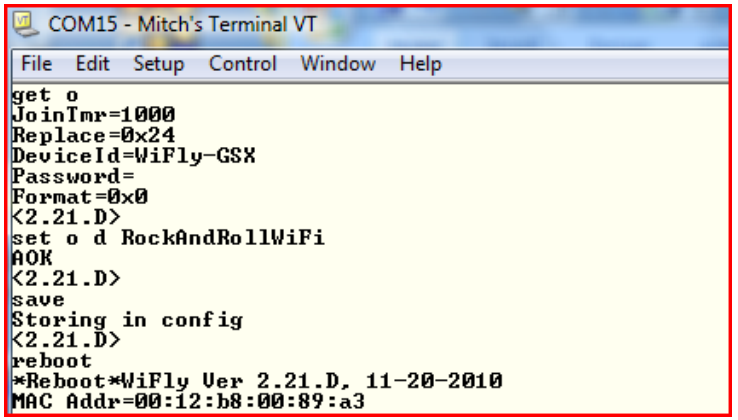


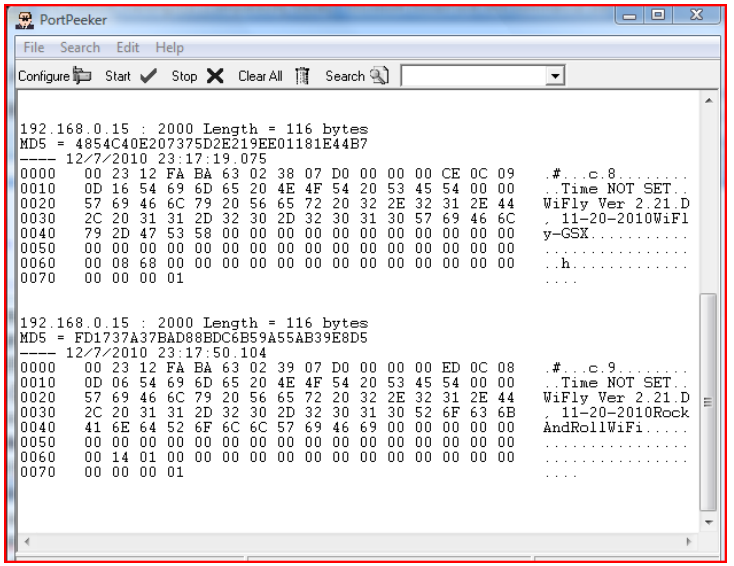


Lab 2: Setting Device Name

- **Device Names Can Identify Products on Network**
- **Can Append Device ID to UDP Broadcast**
- **Set Device ID**
  - Enter command mode
    - **get option**
    - **set o d RockAndRollWiFi** (o=optional, d=deviceID)
    - **save & reboot**
  - Review UDP messages in PortPeeker
- **Set Broadcast UDP Port**
  - Enter command mode
    - **get broadcast**
    - **set b p 50000** (b=broadcast, p=port)
    - Save & reboot not required
  - Reconfigure PortPeeker to listen for UDP packets on port 50000
  - Review UDP messages in PortPeeker

TIP: UDP Broadcast on by Default
Set Interval to 0 to Turn It Off





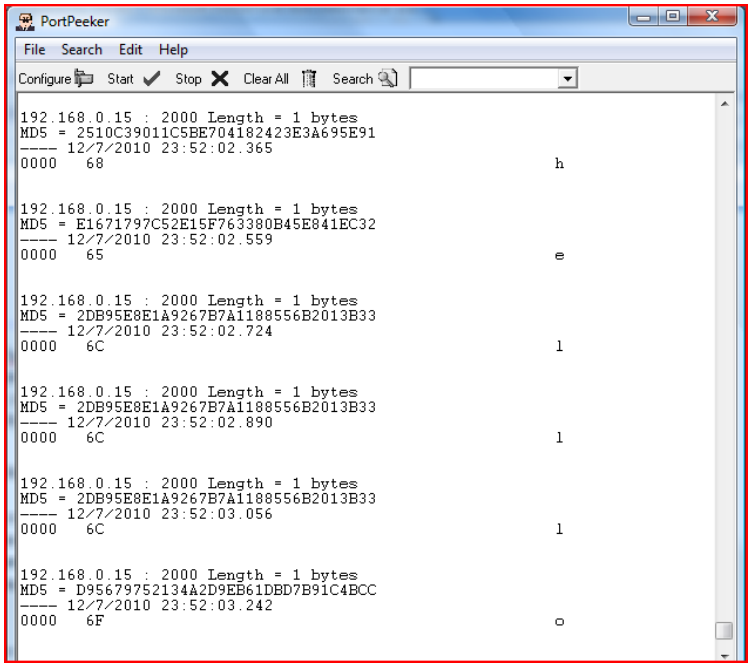
Lab 2: Sending data via UDP

- **UDP Mode Not Enabled by Default**
- **Enable UDP by Setting Remote Host, Port & Protocol**
  - Enter command mode
    - **factory R**
    - Associate with AP
    - **set ip host** <*address*>
    - **set ip remote 50000**
    - **set ip proto 1** (IP protocol bitmask; 1 = UDP)
    - **set comm timer 1000** (try 10, see the change)
    - **get ip**
    - **save & reboot**
  - Type characters; they appear in PortPeeker

Bit Position	Protocol
0	UDP
1	TCP Server & Client (Default)
2	Secure (only receive packets with IP address matches the store host IP)
3	TCP Client only
4	HTTP client mode

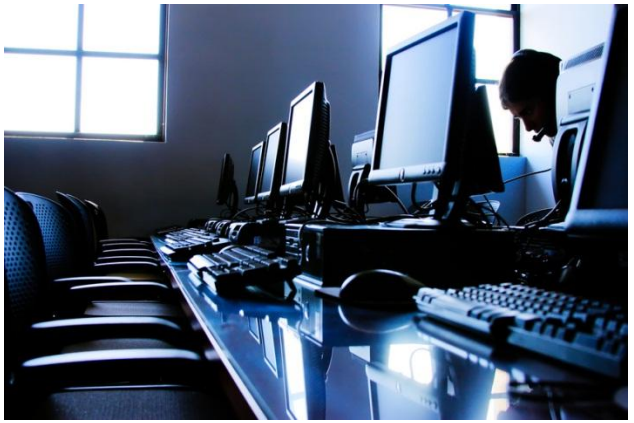
TIP: IP Protocol Value Is a Bit Mask
You Can Enable Both TCP & UDP Messages

```
<2.21.D>
get ip
IF=UP
DHCP=ON
IP=192.168.0.15:2000
NM=255.255.255.0
GW=192.168.0.1
HOST=192.168.0.12:55555
PROTO=UDP,
MTU=1524
FLAGS=0x7
BACKUP=0.0.0.0
```



Lab 2: Conclusion

- **In this lab, we use PortPeeker to discover WiFly devices via UDP broadcast**
- **Module sends UART data as UDP packets when associated with network in UDP Mode**

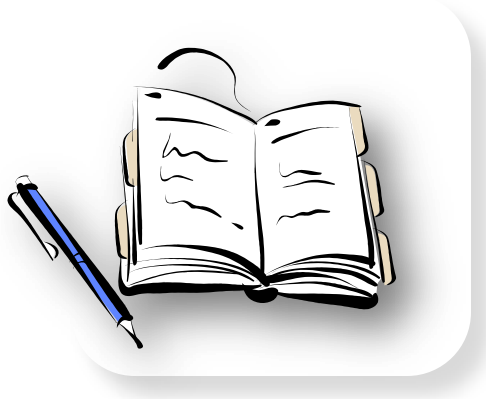


Lab 3: TCP

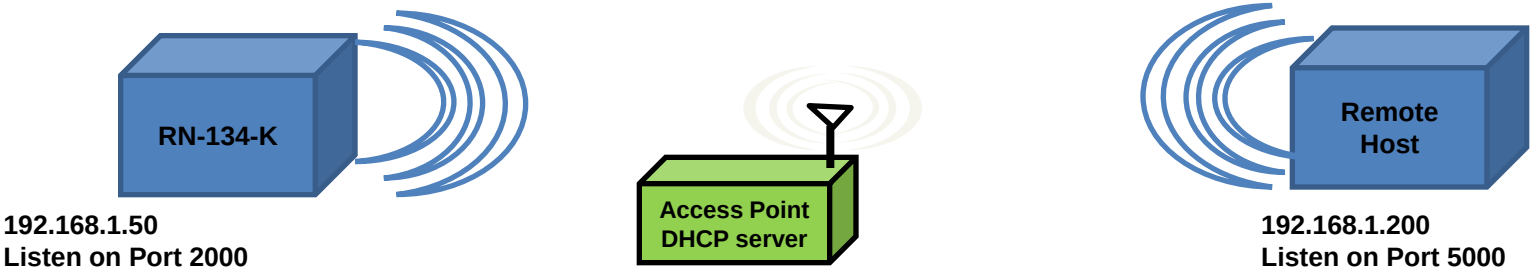
ROVING networks

Lab 3: TCP

- **After the completion of this lab, you will be able to:**
  - Connect **from** module **to** remote host using TCP (client)
  - Connect **to** module **from** remote host using TCP (server)
  - Distinguish between TCP modes
  - Automatically open TCP connections via timer
  - Control TCP connections via micro controller
  - Trigger TCP flush based on different events



Lab 3: TCP Connections



TCP Connections Are Point to Point
Provide Reliable, Guaranteed, In Order Data Delivery
Also Known As Sockets

open 192.168.1.200 5000 ⟶

WiFly Module Opens TCP Connection

- Sensing applications
- Sending data to web server
- Data acquisition systems
- Fleet management

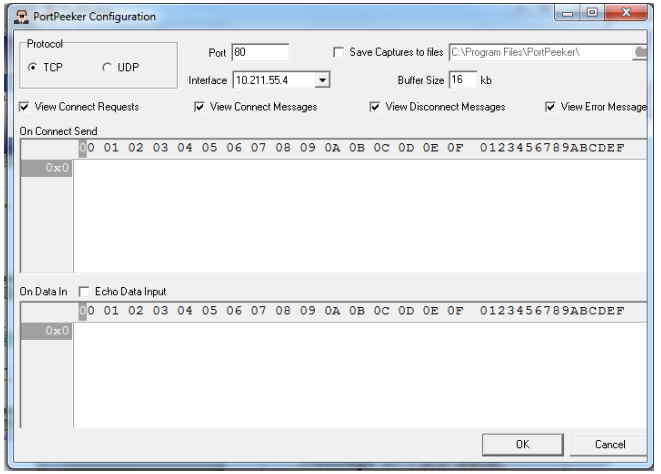
⟵ **open 192.168.1.50 2000**

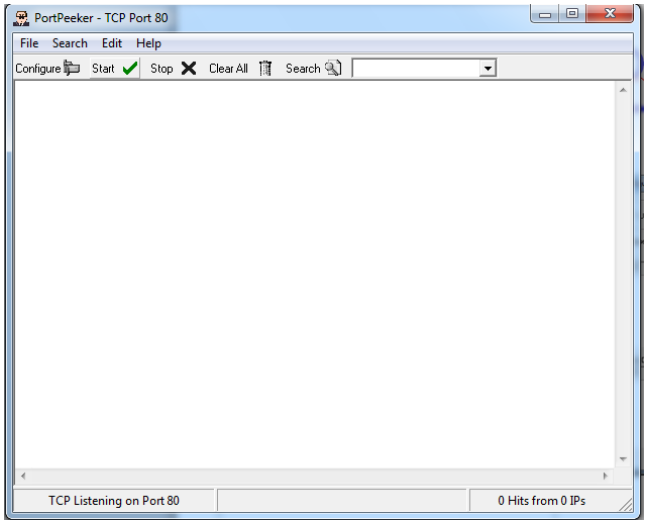
Remote Host Opens TCP Connection

- Industrial controls
- Home automation
- Universal remotes

Lab 3: Setup PortPeeker

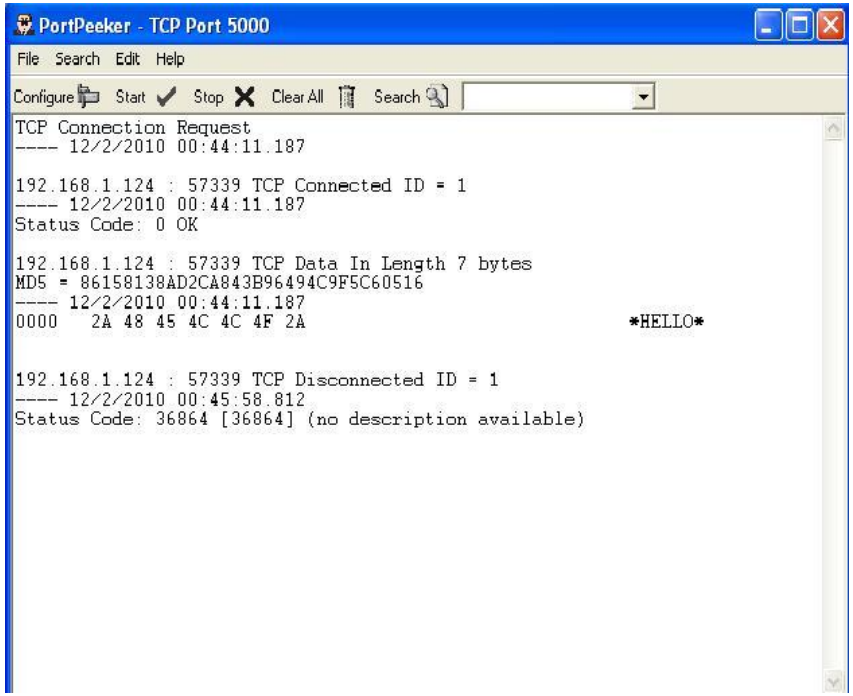
- **Associate Computer with AP**
- **Launch PortPeeker**
- **Configure PortPeeker**
  - Click **Configure**
    (Note PC's IP address in Interface box)
  - Set port to 5000
    (port number matches remote port of WiFly module)
  - Set protocol to TCP
- **Click Start to Capture TCP Packets**





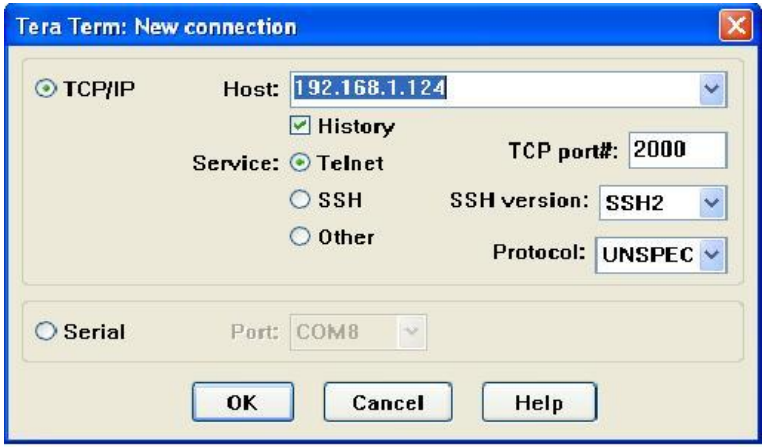
Lab 3: Open TCP Connection from Module

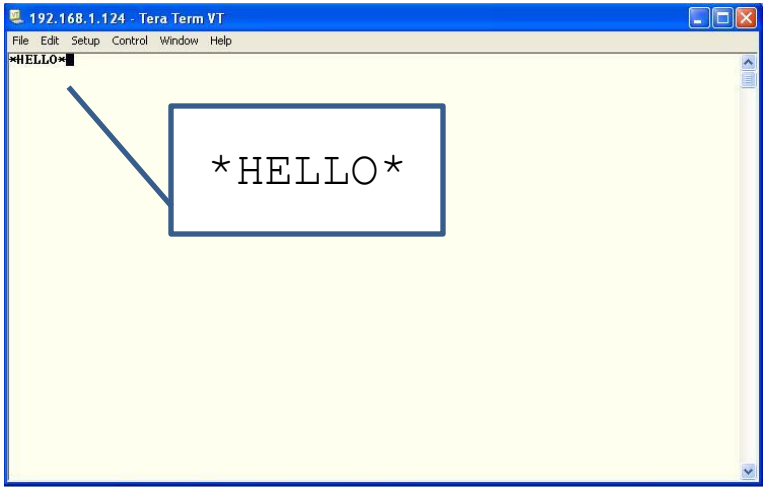
- **With Module Connected to PC over USB-Serial Cable, Open Tera Term on Serial COM Port**
- **Restore Module to Factory Defaults**
  - Enter command mode
  - **factory R**
  - Associate with AP
  - **save** & **reboot**
- **Open TCP connection**
  - **open** <*IP_address*> **5000**
  - *OPEN* shown on serial port (Tera Term window) & packet with *HELLO* on PortPeeker
- **Close TCP Connection**
  - Enter command mode
  - **close**
  - Close string *CLOS* displayed in Tera Term
- **Extra Credit**
  - Change COM timer to 2000 and observe difference on Port Peeker



Lab 3: Connecting from Remote Host to WiFly module

- **In Command Mode, Obtain Module's IP Address**
  - **get ip**
- **Open Telnet Connection from PC Using Tera Term (Use Existing Instance)**
  - Click **File > New connection**
  - Select **TCP/IP**
  - Select **Telnet**
  - In **Host** field, type module's IP address
  - **TCP port#** is 2000 (default listening port)
  - Click **OK**
- ***HELLO* Message Shown in Telnet Window Indicating Successful TCP Connection**
- **Type in Telnet Window; Data Appears on Serial Port Window & Vice Versa**
- **Can Configure Module Remotely over Telnet by Entering Command Mode**





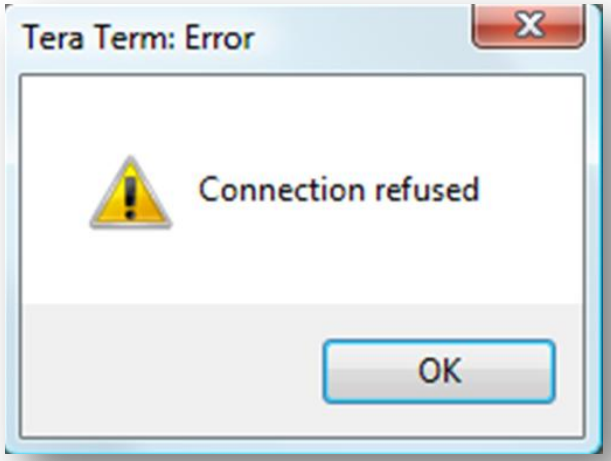
Lab 3: TCP modes

- **Module Supports Three TCP Modes**
  - TCP client & server mode
    - Default mode initiates & accepts TCP connections
    - Currently supports one active connection at a time
    - Concurrent TCP connections supported in future
  - TCP client ONLY mode
    - ONLY initiates TCP connections; cannot accept incoming connections

TIP: Refer to User Manual for More Details on TCP Modes

Lab 3: TCP Client Mode

- **Set Up Module in TCP Client Mode**
  - **set ip proto 8**
  - **save & reboot**
- **Open New Telnet Connection to Module from Tera Term**
- **Second Connection Is Refused Indicating TCP_Client Mode Works Correctly**

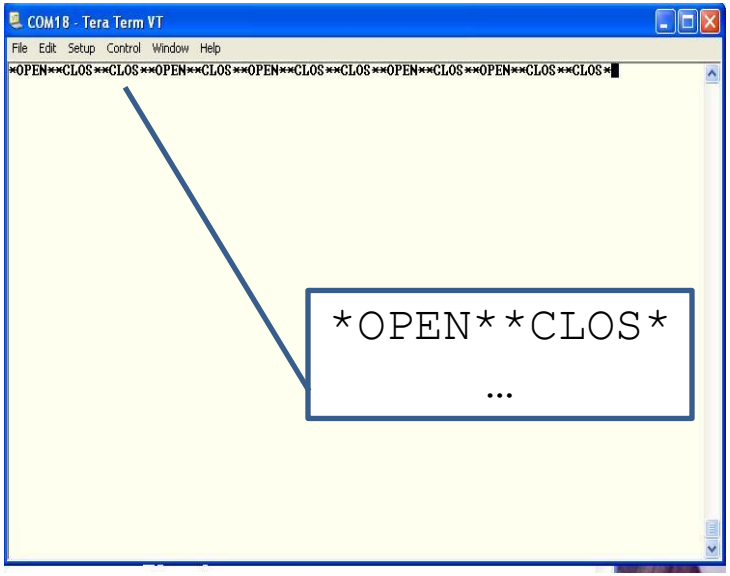


Lab 3: Auto-Connect Feature

- **Module Can Automatically Open TCP Connection to Remote Host on Power Up or Waking from Sleep**
- **Auto-Connect Controlled by autoconn Setting**
  - **set sys auto 1** // Attempts to open TCP connection immediately once only
  - **set sys auto** <*value*> // Attempts to open TCP connection every <*value*> seconds
  - **set sys auto 255** // Attempts to open TCP connection once & go back to sleep immediately // when connection is closed
- **Auto-Connect Requires Module to Store Remote Host's IP Address & Port #**
  - **set ip host** <*host IP address*>
  - **set ip remote** <*port*>
- **Once TCP Connection Is Opened, It Can Be Closed in Several Ways**
  - **close** command
  - Idle timer
  - Remote host
- **Idle Timer Closes TCP Connection after Preset # of Seconds of No Activity (No Tx or Rx) on the TCP Link**
  - **set com idle** <*value*> //Closes the TCP connection after <*value*> seconds of inactivity

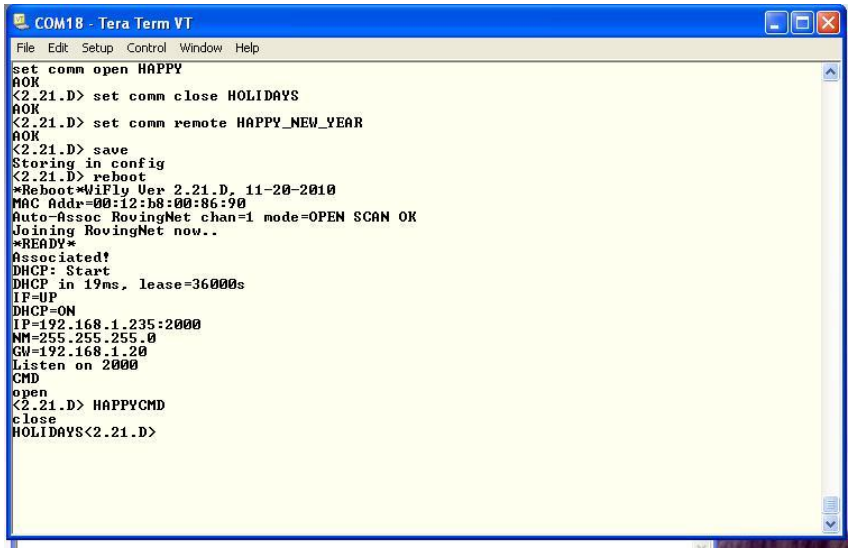
Lab 3: Auto Connect Feature (Timers)

- **Configure the module open a TCP Connection Every 10 seconds, Drops Connection after 3 seconds Inactivity**
  - **set ip host** *<address>*
  - **set ip remote 5000**
  - **set sys auto 10**
  - **set comm idle 3**
  - **save**
  - **reboot**
- **PortPeeker: Connection Opens & Closes**
- **Tera Term: Open & Close Strings Shown when Each Connection Opens & Closes**



Lab 3: Setting Comm Strings

- **Microcontroller Can Look for UART comm Strings as Indication of TCP Connection Status**
  - Factory reset
  - **reboot**
  - Associate with AP
  - **set ip host** <*address*>
  - **set ip remote 5000**
  - **set comm open HAPPY**
  - **set comm close HOLIDAYS**
  - **set comm remote HAPPY_NEW_YEAR**
  - **save & reboot**
  - Enter command mode
  - **open**
  - See open string in Tera Term
  - See remote string in PortPeeker
  - Enter command mode
  - **close**
  - See close string in Tera Term



Tip: Microcontroller Can Read UART Open & Close Strings to Determine TCP Connection Status

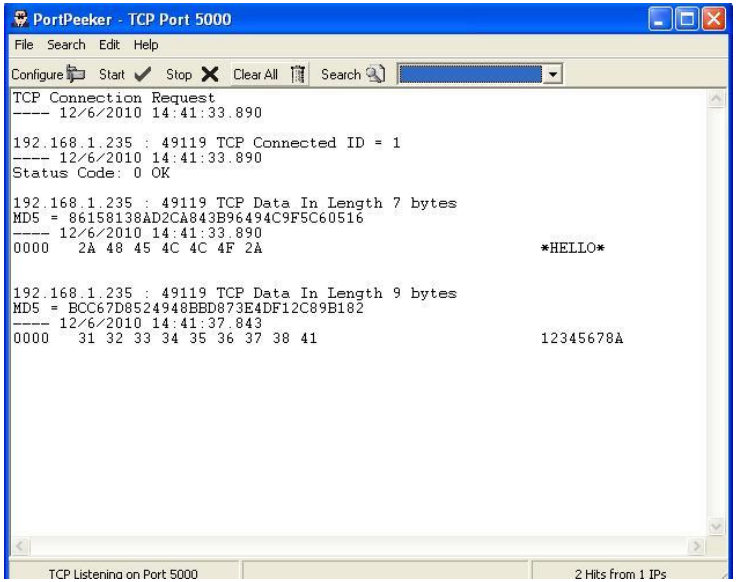
Lab 3: Forwarding TCP packets

- **When Data Is Written to Module's UART, TCP Packets Forwarded Based On**
  - Flush timer
  - Flush size
  - Match character
- **TCP Packet Sent When Any of These Events Occur**
- **Parameters Logically ORed to Determine when TCP Packet Is Sent**
- **When Configured Correctly, Module Can Be Optimized for Low Latency or High Throughput**
  - Low latency: use lower flush timer value & flush size
  - High throughput: use higher flush timer value & flush size

Tip: Module Tries to Optimize Automatically for Bandwidth by Increasing Default Flush Size with Higher Baud Rates

Lab 3: Forwarding TCP packets

- **Forwarding Packets Based on Flush Timer**
  - **set comm timer 1000**
  - **save**
  - **open**
  - Type text after TCP connection opens
  - After you stop typing, TCP packet is sent 1 second later
- **Forward Packets Based on Match Character**
  - **set c t 0** *(why do we send this command ?)*
  - **set comm match 65**
    - This parameter expects ASCII decimal character or HEX value of the match character (e.g., 65= Capital A)
  - **save**
  - **open**
  - Type 12345678A
  - TCP packet sent out after you type A character
  - Observe packet in PortPeeker
- **What Do You Learn from Using 'get c' Command?**



Lab 3: Open TCP Connection from Module

- With Module Connected to PC over USB-Serial Cable, Open Tera Term on Serial COM Port
- Restore Module to Factory Defaults
  - Enter command mode
  - Factory reset and reboot
  - Associate with AP
- Open TCP connection
  - **open** <*IP_address*> < port>
  - *OPEN* shown on serial port (Tera Term window) & packet with *HELLO* on other side
- TIP: Find your IP address via *get ip* command

Lab 3 Open TCP Connection from Module

- Send data to your lab partner
- This is texting via TCP!

Lab 3: Conclusion

- **Module Can Open TCP Connection to Remote Host & Accept Incoming Connections from Remote Host**
- **Auto-Connect Automatically Opens TCP Connection**
- **Idle Timer Can Automatically Close TCP Connection**
- **Alternative GPIO Functions Allow Microcontroller to Control & Monitor TCP Connections**
- **comm open, close & remote Strings Can Indicate TCP Connection Status**
- **TCP Packets Forwarded Based On**
  - Packet size
  - Match character
  - Flush timer

Lab 4: Access Point mode

Lab 4: Access Point Mode

Microchip RN Wi-Fi modules now have the soft Access Point as the standard feature

- **Advantages**
  - Enables Android devices to talk to modules without need for infrastructure
  - Runs DHCP server
  - Supports up to 7 clients
  - Supports routing between clients
  - Support WPA2-AES personal security in 4.41 firmware
- **How about AdHoc mode?**
  - Use 2.383 firmware

Lab 4: Access Point mode

UDP Server
192.168.0.100

Lab 4: Creating a Default AP Network

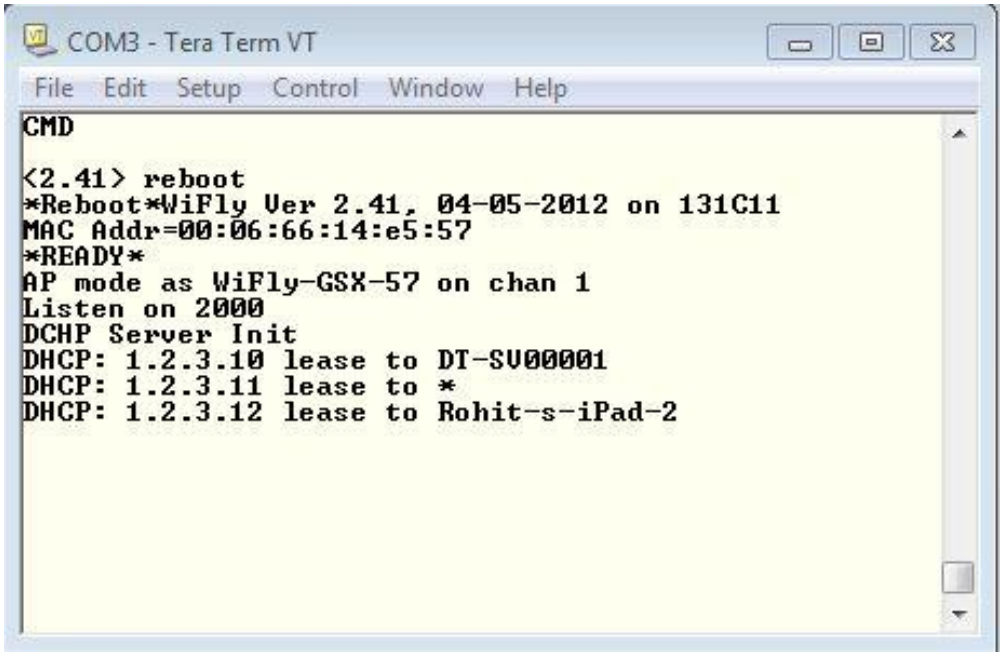
- **Two ways to Creating a Default AP Network**
  - **Pull GPIO9 up to 3.3VDC before reset**
  - **Issue apmode command in command mode**
    - **apmode <*bssid*> <*channel*>**
- **Default AP Network Setting**
  - SSID: WiFly-EZX-*XX* (WiFly-GSX-*XX* on RN-131), where *XX* is last two bytes of MAC address
  - Channel: 1
  - DHCP server: Enabled
  - IP address: 192.168.1.1
  - Netmask: 255.25.5255.0
  - Gateway: 192.168.1.1

Lab 4: Create permanent Custom AP Network in Software

- **Create Custom AP Network with User-Defined Settings**
  - **set wlan join 7** // Create AP mode network
  - **set wlan channel** <*value*> // Specify channel to create // network
  - **set apmode ssid** <*string*> // Set up network SSID
  - **set apmode passphrase** <*string*> // Set up network security
  - **set ip dhcp 4** // Enable DHCP server
  - **set ip address** <*address*> // Specify IP address
  - **set ip net** <*address*> // Specify subnetmask
  - **set ip gateway** <*address*> // Spcify gateway
  - **save** // Store settings
  - **reboot** // Reboot module in AP mode

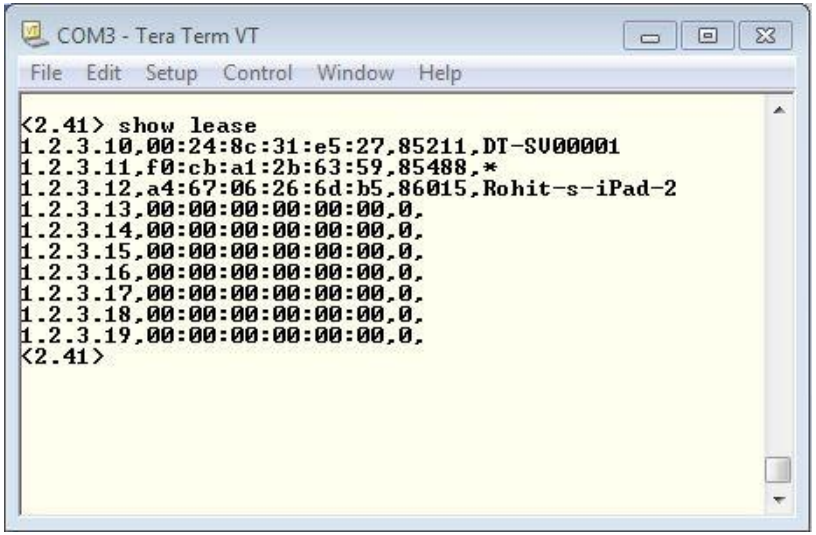
Lab 4: Connect to AP Network Created by Module

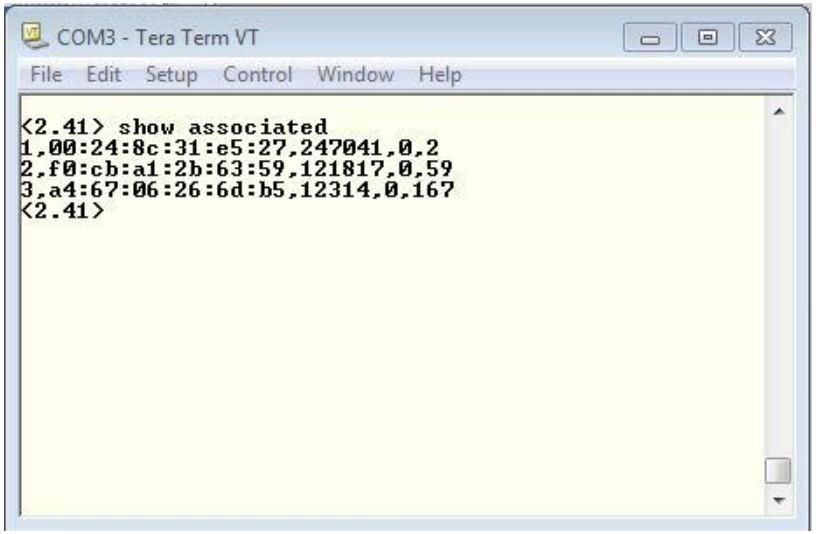
- From PC/Mobile Phone/Tablet, Connect to Module-Created Network
- Module Displays Client's Device Name



Lab 4: View Associated Devices & Lease Times

- **View Device Lease Times**
  - **show lease**
- **View List of Connected Devices**
  - **show associated**



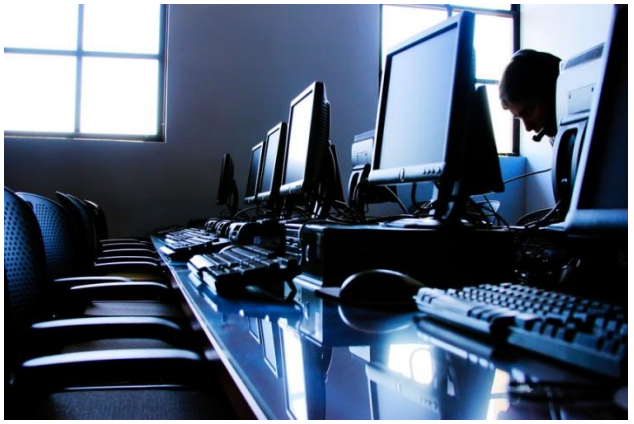


Lab 4: Exercise

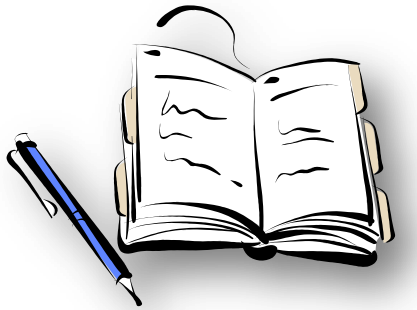
- **Connect your PC or mobile device to RN-171 in AP mode**
  - **Put RN-171 into AP mode**
  - **Associate your PC or mobile device to the AP network created by RN-171**
  - **Connect your PC or mobile device to RN-171 via TCP**
  - **Send some data over TCP socket**
- **Connect 2 modules together and send data via TCP**
  - **Work with your neighbor**
  - **One as AP and one as infrastructure client**
  - **…**

Lab 4: Conclusion

- In this lab, we learn different ways to enable soft AP mode on RN171
- You can use a PC or a mobile device to associate the soft AP created by RN171 and send data to RN171 via TCP
- Connect 2 modules together to do cable replacement by using soft AP mode



Lab 5: Connect RN171 to PIC32



Lab 5 Hardware Setup

- Connect to RN-171-PICTAIL on APP1632 with PIC32MX795 module
- Connect debugger (ICD3 or PICKIT3)
- Connect a RS232 cable to APP1632
- Power up the APP1632

Lab 5 Software Setup

- **Install MPLAB C32 and MPLAB IDE**
- **Get WF002 code from RTC 教育訓練光碟**
- **Open TereTerm to monitor APP1632 RS232 console**
- **Open WF002 APP1632 Lab project and build it**
- **Run the project. You should see boot up string and RN171 being reset in the APP1632 RS232 console port**
  - Type ‘c’ to APP1632 RS232 console port get into command
  - Type ‘a’ to create a default AP
  - Type ‘r’ to reboot

Lab 5 Exercise

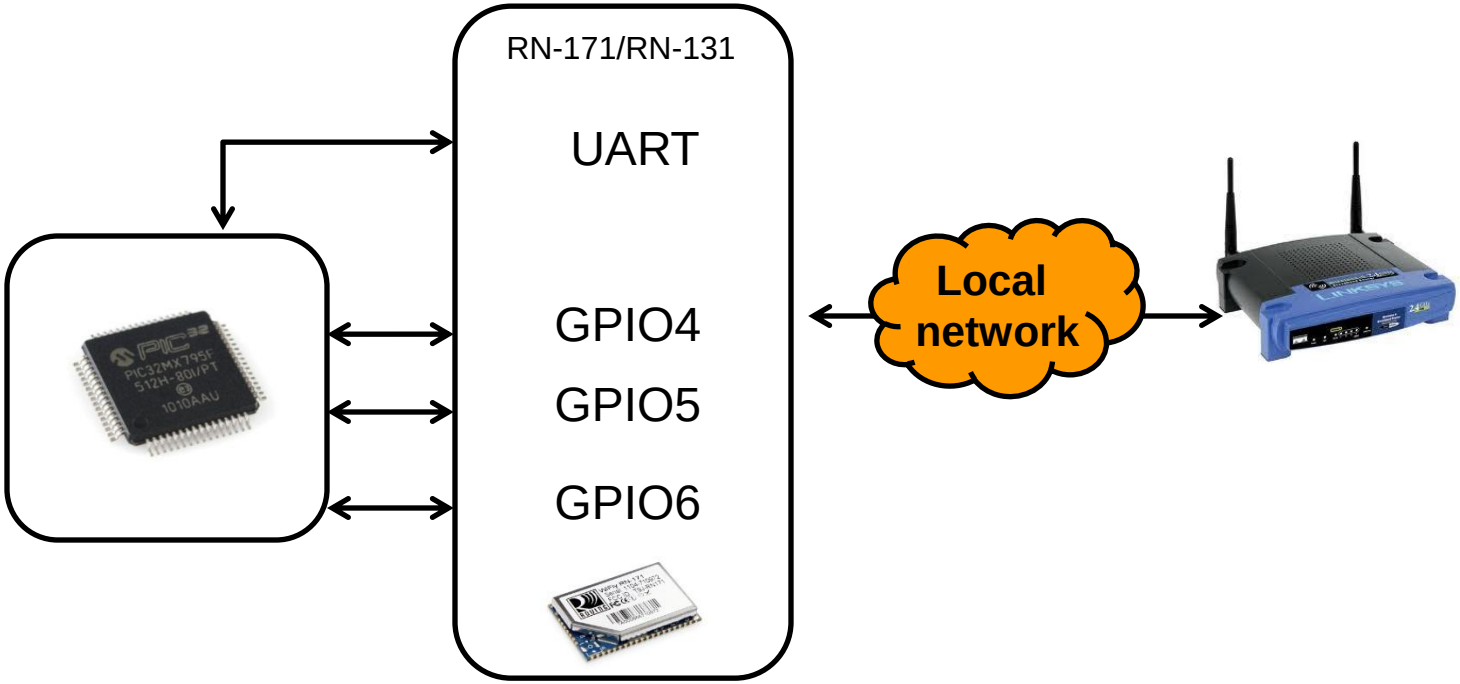
- **Modify the code to do the following**
  - When you type ‘a’ in the APP1632 RS232 console, create an AP network of a SSID you want.
  - When you type ‘j’ in the APP1632 RS232 console, join a SSID you want.

Supporting the Internet of Things
With Embedded Wi-Fi

Real world issues and tricks to put RN Wi-Fi modules in your design

Interfacing Module to Micro

- 3 GPIOs for Control
- UART for data transmission and receipt



GPIO Pin Function And Usage

Embedded Microcontroller

RN Wi-Fi® Module

Notifies the MCU that it now has Wireless connectivity

GPIO 4 = HIGH by RN Module

Event: Module is associated with AP, has a valid IP address

TCP open

TCP close

GPIO 5 = HIGH by MCU

RN Module attempts to open a TCP connection to the stored IP address

Notifies the MCU that now a TCP connections is established and it can initiate data transfer over UART

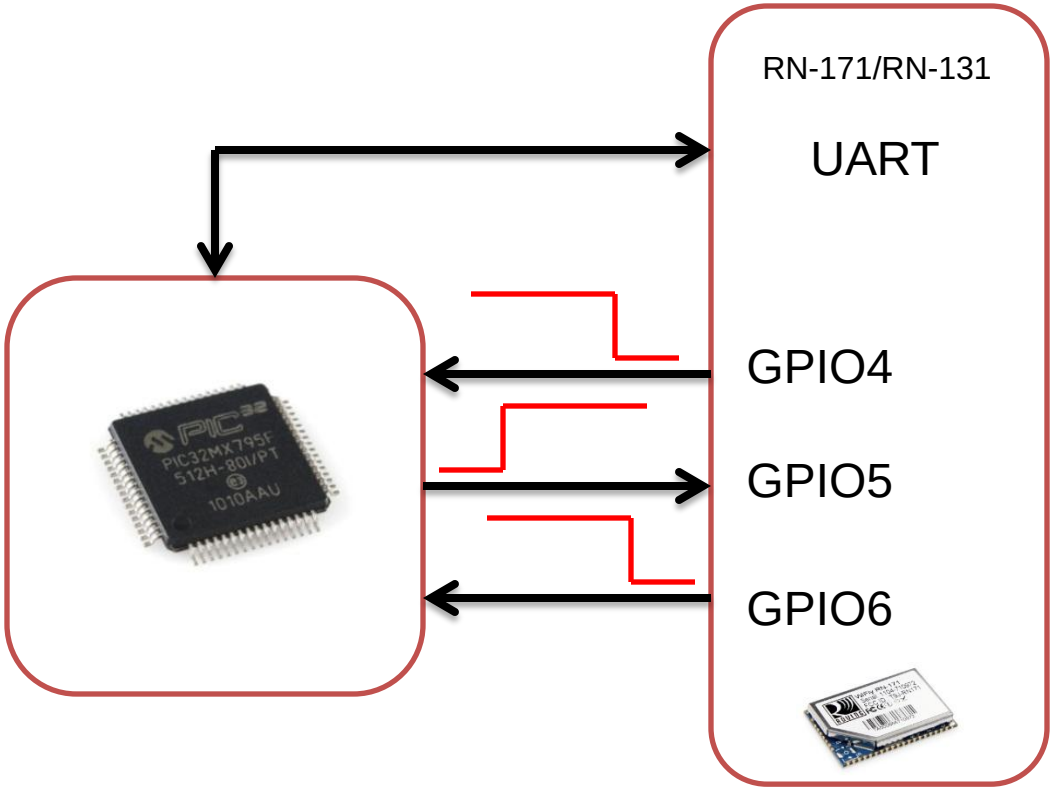
GPIO 6 = HIGH by RN Module

Event: Module has successfully opened a TCP connection

Programming the GPIO Function Pins

Enable Alternative Functions

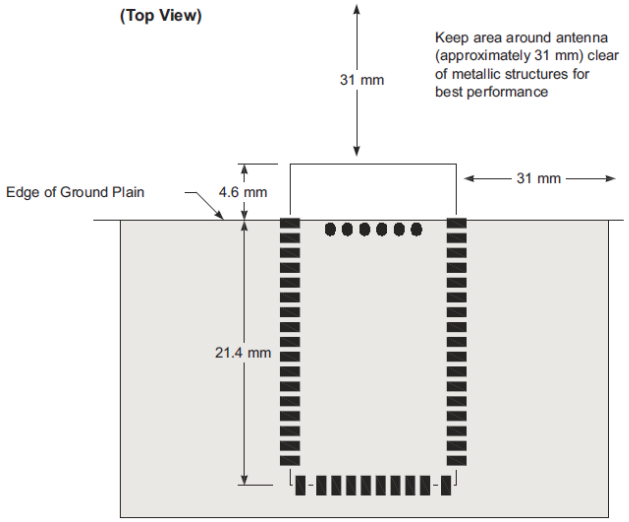
– **set sys iofunc 0x70**



Poor Performance???

- Common problem encountered by customers: expertise in end product but lack experience in wireless
  - Disconnection or loss of data packet
  - Cannot cover normal distance between device and router
  - Unstable firmware
- RF module still need to follow all design rules that are applicable to RF design
- Microchip helps customers review the designs

Integrate RN131/RN171 with your MCU



- **Good Antenna exposure**
  - Have it exposed
  - Keep Antenna away from other
  - Do not use Metal casing
- **Power Supply**
  - Reserve 300mA for RN171 and 500mA for RN131
  - Condition your power with a 0.1uF and 47uF cap right before feeding into the module
  - Protect the power trace from other noise
- **Digital / RF Isolation**
  - Isolate Power and ground from the MCU as much as possible if you need to put them on the same PCB.

Integrate RN131/RN171 with your MCU

- Put 3.3k ohm resister between MCU UART TX, RX, CTS and RTS
- If you use default LED blinking for connection status, connect GPIO4, 5, 6 to LED with a 220 ohm resister.
- If you use GPIO for connection control from MCU, set alternative GPIO function register and use GPIO4, 5, 6 for connection control.
- Reserve test points or pin header for ISP UART
  - For local UART firmware update
  - ISP_TX, ISP_RX, 3.3VDC and GND

Data communication limit of RN171/131

- RN131/171 WiFi modules are designed for medium to low data rate data only
- WiFi is “packet base” network protocol
- You cannot put too many packets in the air. The realistic packet rate is 70-100ms per second
- The UART baud rate can go up to 921600bps
- The realistic throughput is about 100kbps in combination of 2 directions (TX, RX) for none-Apple application

WiFly 4.41 firmware downgrade issue

- WiFly 4.41 firmware is not friendly to firmware download
- If you do ftp u <older firmware> in 4.41, you will brick the module
- You need to delete the module config file before you downgrade to other firmware
- The easiest way to do it is: ftp cu <firmware>
- If the module is bricked due to 4.41 firmware downgrade. Contact Martin Shay.

Taiwan NCC

- My wireless parts are stuck at custom!!!
- Taiwan government requires wireless device to pass NCC to be imported to Taiwan
- Most MCHP WPD modules are NCC certified
- That is not enough!?
  - Microchip needs to authorize you to use the specific NCC approval for each wireless module. This only needs to be done once for each module.
  - Your company needs 2 things to import RF parts:
    - 無線產品營業項目登記（商業司）
    - 低功率射頻執照（**NCC**）

Firmware release to use

- If you order the modules without specifying the part number, what you will get it the latest and greatest
- If your customer approved certain firmware release and do not want to use the latest release. You need to specify the custom f/w number at ordering time
- Latest realeses
  - AP mode: 4.41
  - Adhoc mode: 2.383
- The part number convention is <default part number>+<f/w release>
- For example
  - RN131G-I/RM238 for WiFly 2.38 RN131G
  - RN171-I/RM228 for WiFly 2.28 RN171

Resources

- **Visit Roving's support site for all documentation**
  - http://www.microchip.com/wireless

- **User Manuals**
- **Data Sheets**
- **App Notes**
- **CAD Tools**
- **Schematics**
- **Certifications**
- **Utilities**
  - (e.g.. TeraTerm)

Trademarks

Questions???

謝其煜

Martin.Shay@microchip.com

+886.939125122
+886.2.2508.8642

Thank You