# Advance Serial Communication and Wireless Technologies

# Outline

- Ethernet Communication
  - Ethernet port of LPC1769
  - Direct Connection between LPC1769
  - Connecting LPC1769 to Network
- Wireless Technologies
  - ZigBee
  - Bluetooth
  - Wifi
- Interfacing wireless modules to MCU

# Basic of Ethernet

- Ethernet is a serial protocol which is designed to facilitate network communications.
  - Any device successfully connected to the Ethernet can potentially communicate with any other device connected to the network.
- Networks are often described as being one of two types:
  - Local area network (LAN): usually for devices connected together in close proximity, perhaps in the same building and often without Internet access
  - Wide area network (WAN): describes a network of devices over a greater geographical area, usually connected by the Internet.

# Basic of Ethernet

- Ethernet communications
  - Serial Data communication
  - Defined by the IEEE 802.3 standard
  - Support data rates up to 100 Gigabits per second.
  - Uses differential send (Tx) and receive (Rx) signals, resulting in four wires labeled RX+, RX-, TX+ and TX-

# Basic of Ethernet

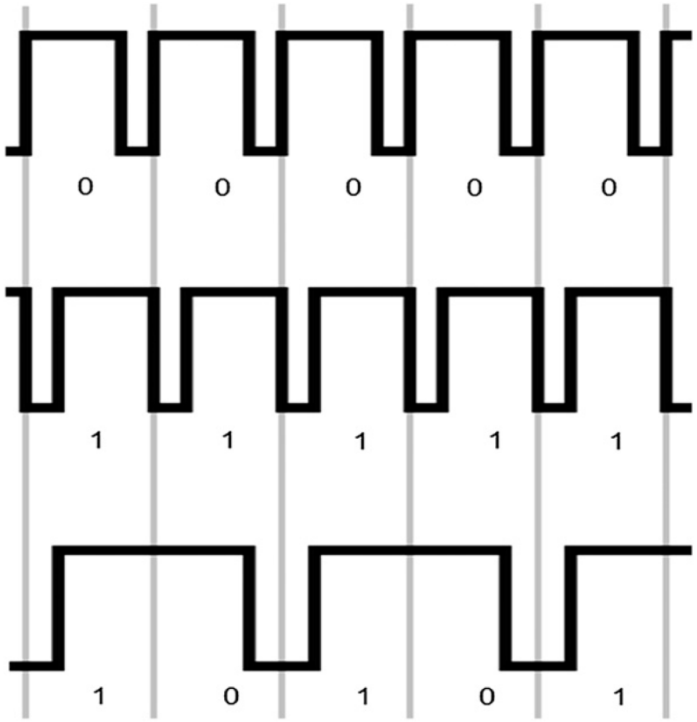
- Ethernet Data package: A serial data packet referred to as ***frame***.
- Frame defines its own size
  - Only the necessary amount of data is communicated, with no wasted or empty data bytes
- Each frame also includes
  - A unique source and destination MAC address.
  - A set of preamble and start of frame (SOF) bytes and
  - A frame check sequence (FCS)
- Package is transmitted least-significant bit first

# Ethernet Frame

| Preamble | Start of frame delimiter | Destinatio MAC address | Source MAC address | Length | Data | Frame check sequence | Interframe gap |
|---|---|---|---|---|---|---|---|
| 7 bytes of 10101010 | 1 byte of 10101011 | 6 bytes | 6 bytes | 2 bytes | 46-1500 bytes | 4 bytes | 12 bytes |
| **synchronous** | | **header** | | | **payload** | **CRC** | **Idle** |

- Minimum package size is 7+1+6+6+2+46+4 = 72 bytes
- If data is less than 46 bytes, 0x00 will be filled in until its payload is equal to 46 bytes
- Using Manchester encoding to simplify clock recovery (it does not have no explicit clock line)

# Manchester Encoding for Ethernet




- ***Low-to-hi*** transition means **1**, ***hi-to-low*** transition means **0**.
- This always creates transition in data stream. As a result, the clock signal is effectively embedded within the data.

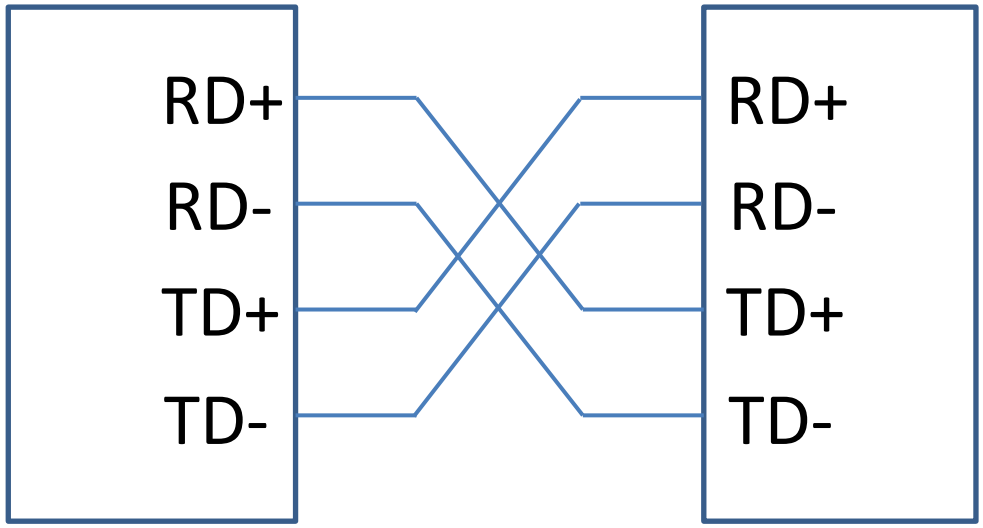
# LPC1769 Ethernet Port

- Features and Configuration
  - 10/100 Mbps, Fully compliant with IEEE 802.3
  - Independent transmit and receive buffers memory mapped to shared SRAM

    (for more see the user manual)
- Configuration
  - Power set PCENET in PCONP
  - Pin set PONSEL, PINMODE
  - Interrupts are enabled in NVIC using ISEN Register

# LPC1769 Ethernet Port

- Ethernet Communicate between two LPC1769s
  - Differential signaling
  - Crossed signal connection
  - Configure its registers or use API such as mbed.h



# LPC1769: Ethernet Port

- Connecting to the network
  - Link embedded system to the internet (and hence the devices under its control)
  - Once the link is established we can do many things such as monitor status, exert the control,  etc.
  - For examples
    - Update firmware of embedded system over the internet
    - Vending machine report its stock to the head office
    - Remotely control  home appliances from the office
    - etc.

# LPC1769 Ethernet

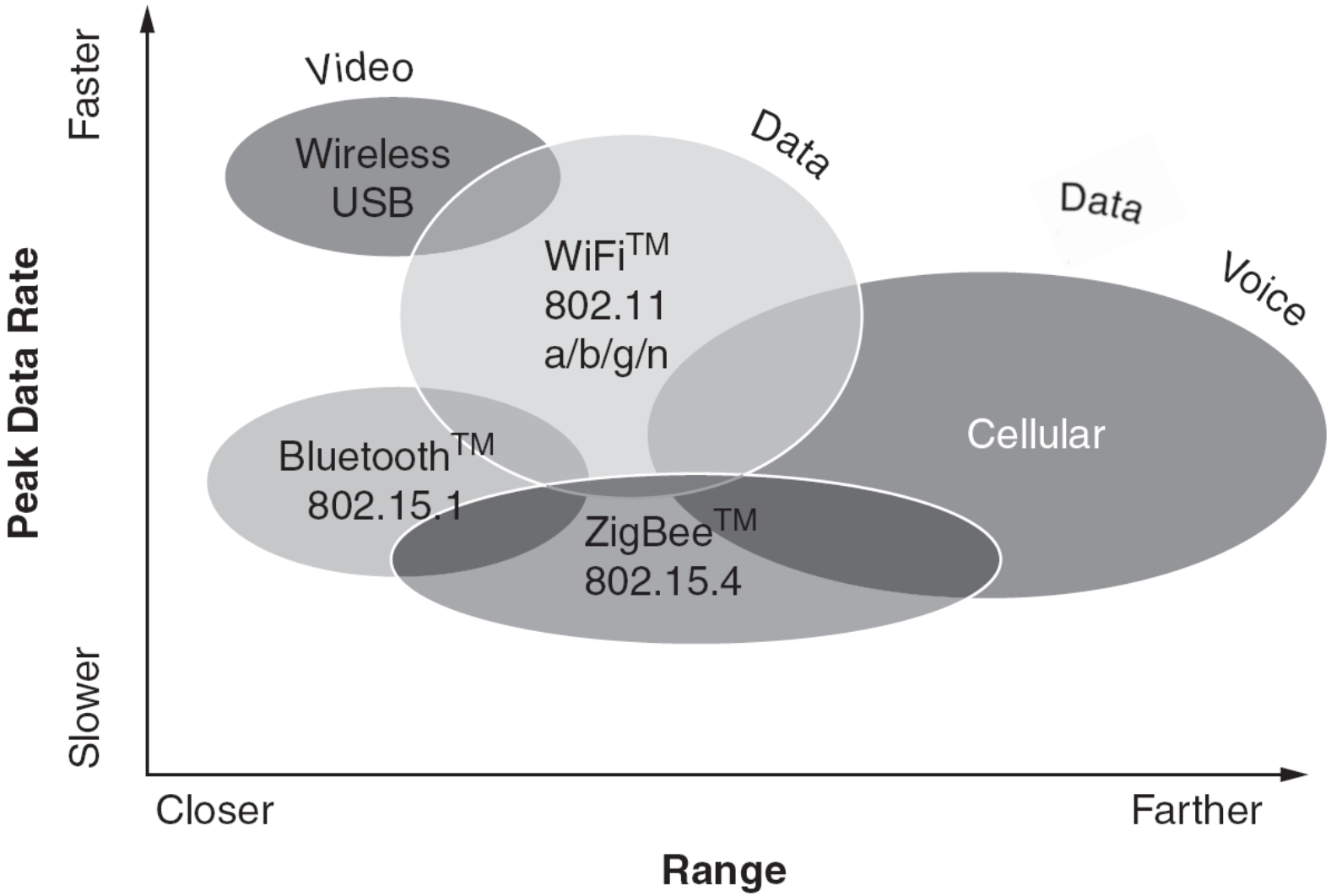
Connecting  Xpresso to the network

- Requires
  - Software: network interface libraries (as a software stack)
  - Hardware: connecting network port to a hub/router (RJ45)
- Using  Xpresso as an HTTP client
  - To access data from the Internet, it requires
- Using  Xpresso as an HTTP File sever
  - Host files to be accessed from a remote PC
- Using Remote Procedure Calls (RPCs) to modify LPC1769 outputs
  - Ex: RPCs  from a network PC to control GPIO port, etc.

# Wireless Technologies

- Embedded System + Wireless Technology

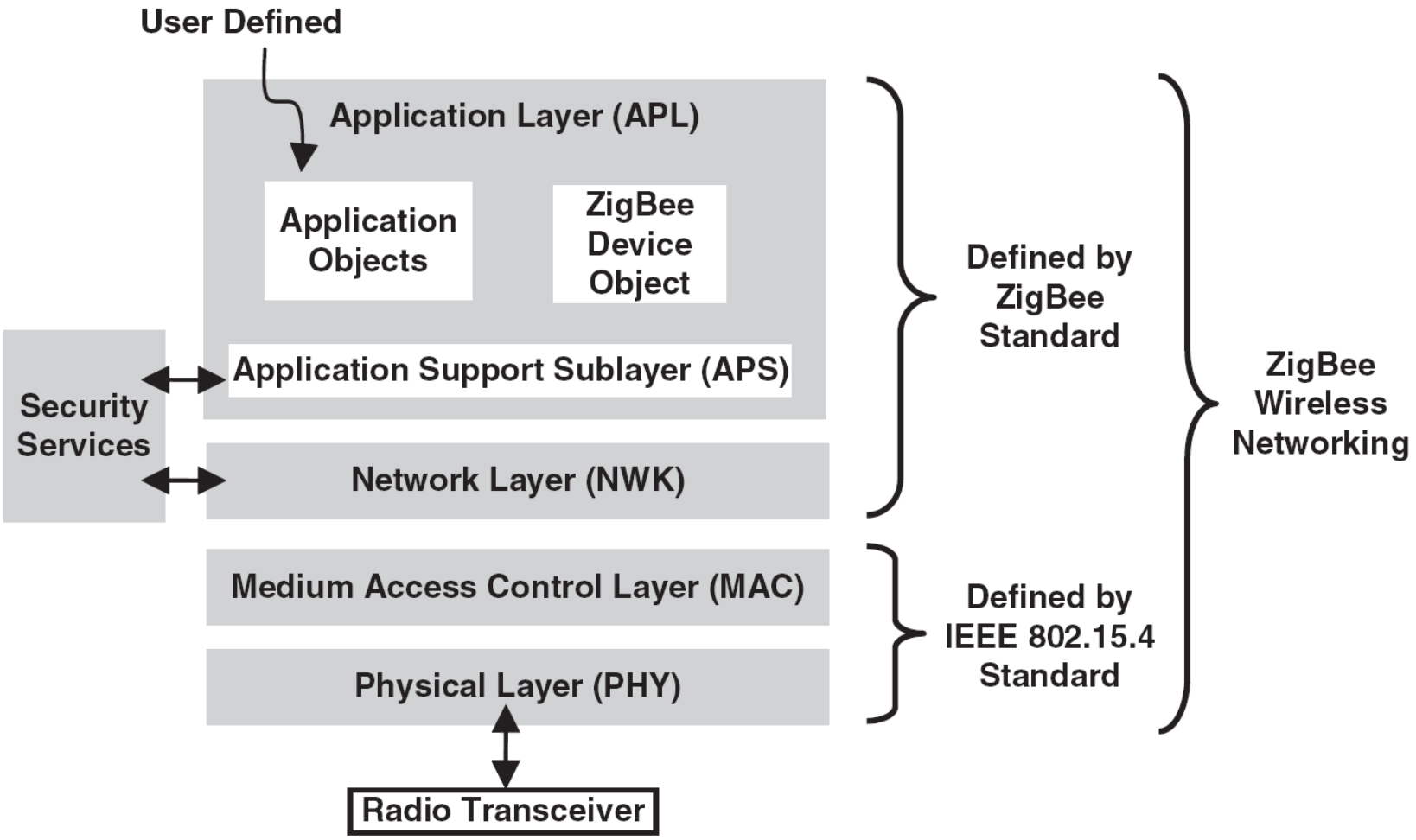
# Wireless Technologies



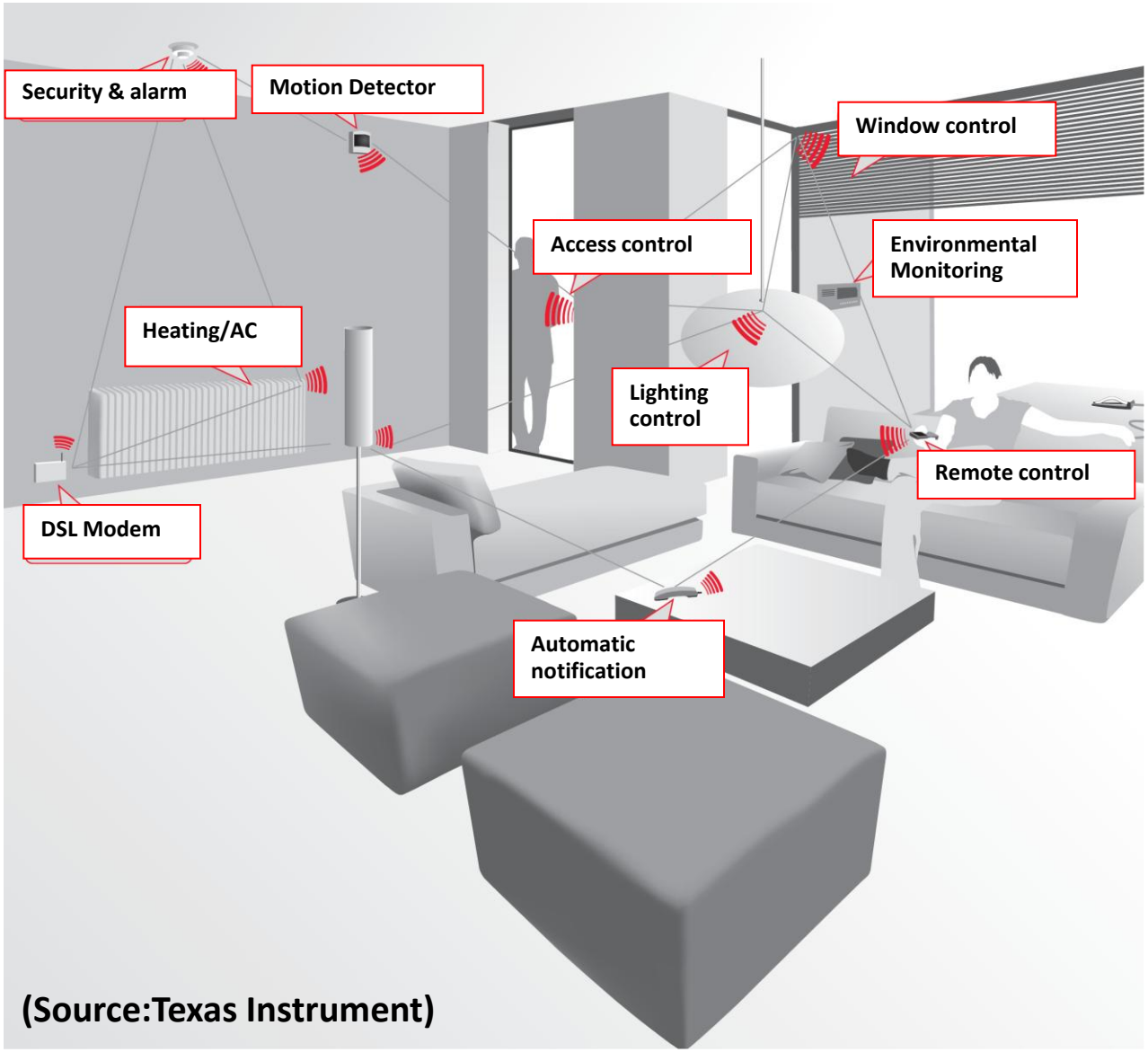
# ZigBee: Basic

- Protocol for  low speed, low power wireless network
- Adapted from IEEE 802.15.4
- Applications
  - Home Automation
  - Building Automation
  - Healthcare system
  - Wireless sensor network  (WSN)

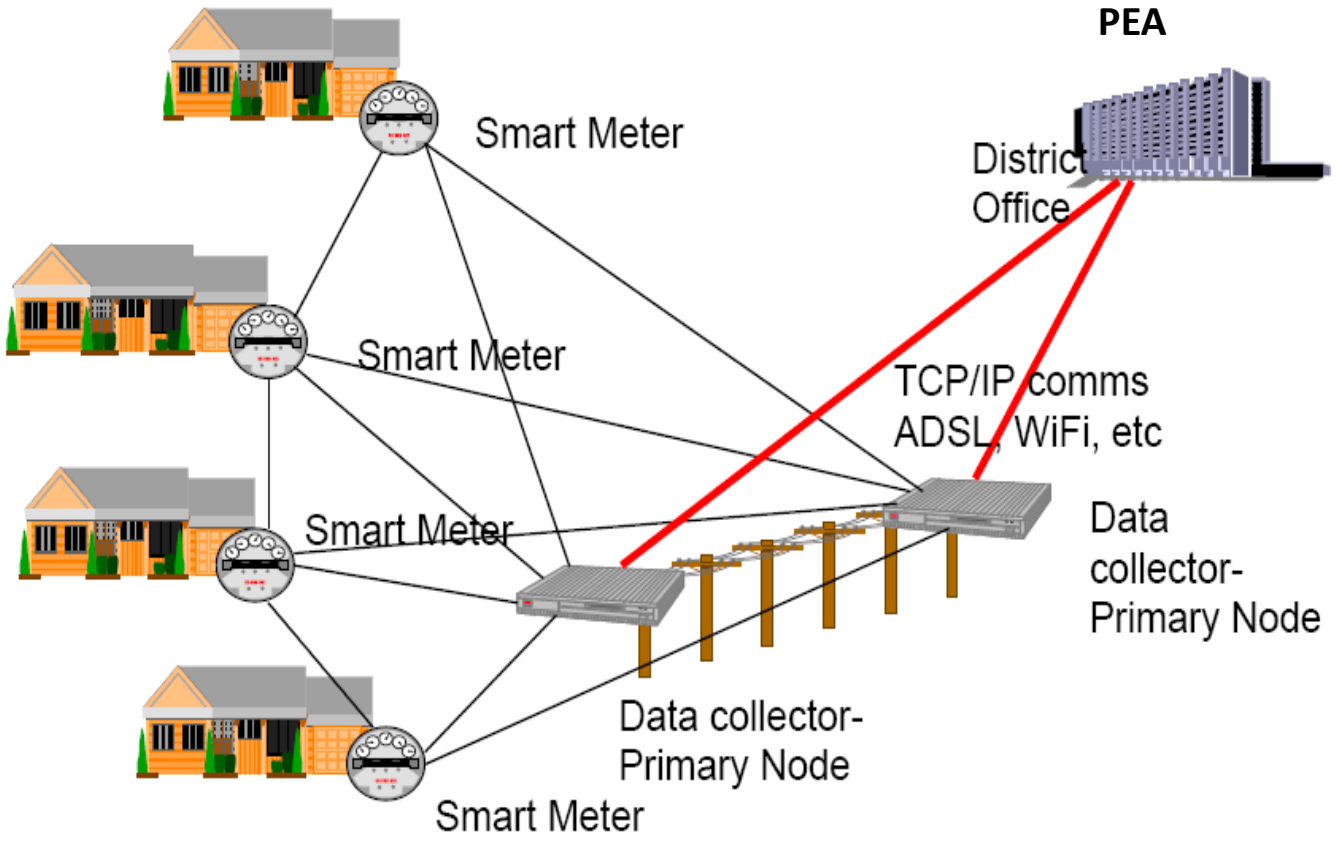
# ZigBee Protocol Stack



# ZigBee in Home Automation

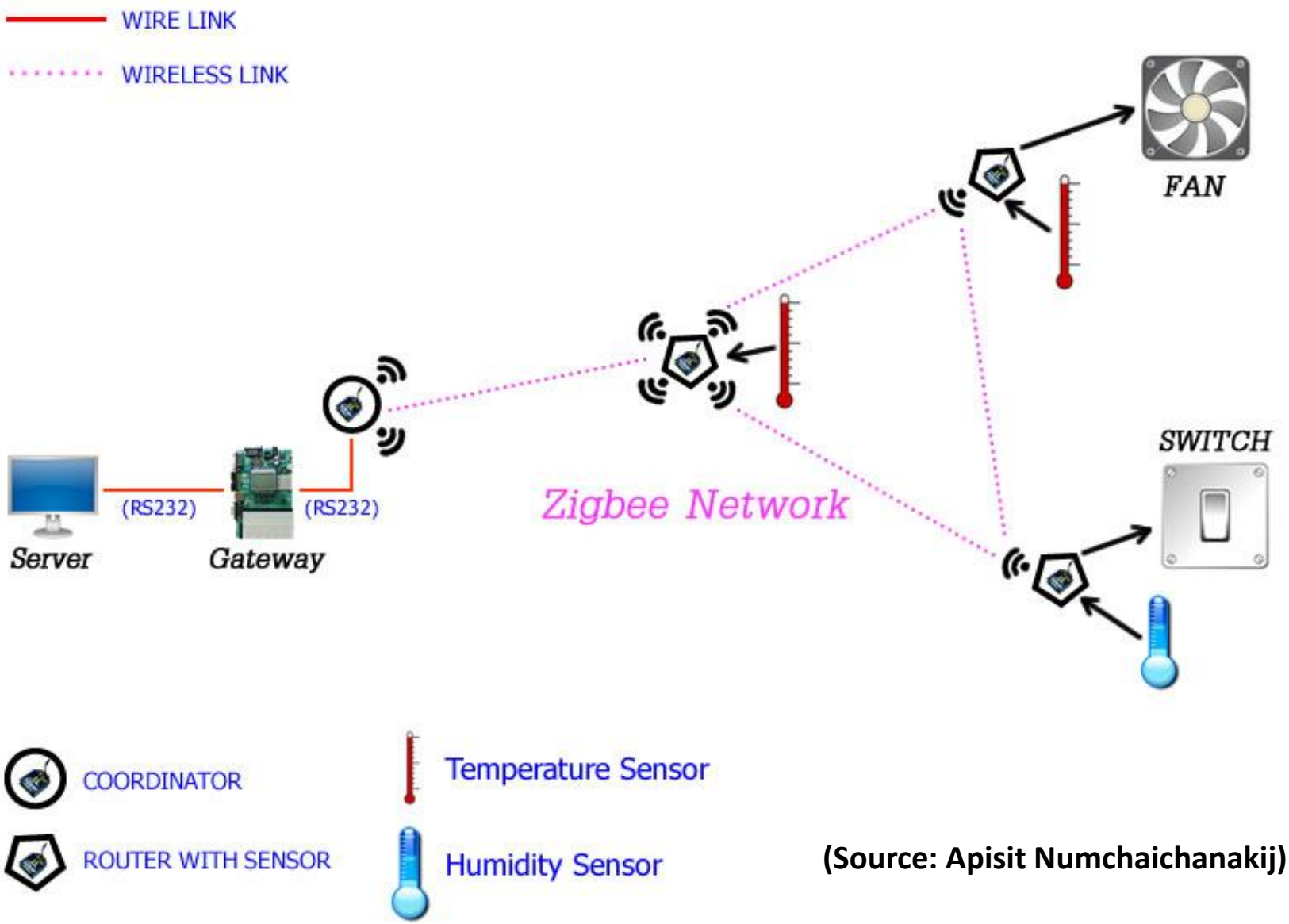


# ZigBee in Smart Power Management




**(Source: Dr.Visit Hirankitti and PEA)**

# Wireless Sensor Network using ZigBee




(Source: Apisit Numchaichanakij)

# ZigBee: Basic

- How to use Zigbee with MCU ?
  - Using ZigBee module such as the Xbee
    - Radio Transceiver
    - Simple digital I/O
    - Interface with MCU using UART

# Bluetooth

- What is Bluetooth?
  - A form of digital radio communication, operating in the 2.402-2.480 GHz radio band
  - Provides wireless data links between devices such as mobile phones, computers, wireless audio headsets and systems requiring the use of remote sensors
  - Accepted as a standard ( IEEE 802.15)
- Main characteristics :
  - Class
  - Range
  - Power
  - Data rate
  - Number of simultaneous links

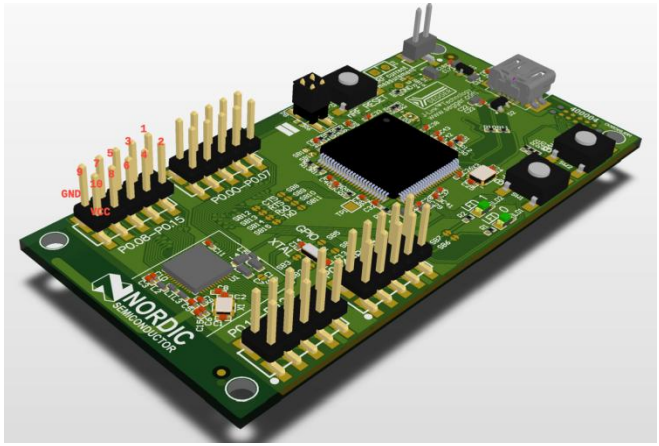
# Bluetooth: Basic

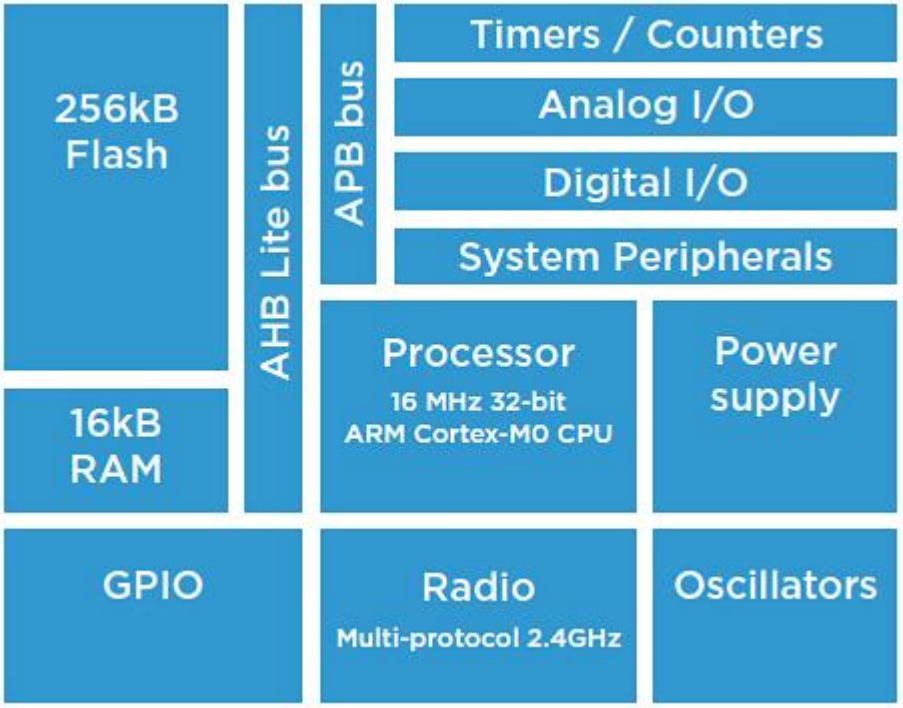
| Class | Max. permitted power | | Typ. range (m) |
|---|---|---|---|
| | (mW) | (dBm) | |
| 1 | 100 | 20 | 100 |
| 2 | 2.5 | 4 | 10 |
| 3 | 1 | 0 | 1 |

| **Version** | **Data rate** | **Max. application throughput** |
|---|---|---|
| 1.2 | 1 Mbit/s | 80 kbit/s |
| 2.0 + EDR | 3 Mbit/s | 80 kbit/s |
| 3.0 + HS | 24 Mbit/s | N/A |
| 4.0 | 24 Mbit/s | N/A |

Current version is V 4.1 ( low Energy Blue Tooth Smart)

# Bluetooth System-on-chip






nRF51822
MCU + BT transceiver

# Bluetooth System-on-chip

## Button and LED connection

| Part | GPIO | Short |
|---|---|---|
| Button 1 | P0.16 | |
| Button 2 | P0.17 | |
| LED 1 | P0.18 | SB6 |
| LED 2 | P0.19 | SB7 |

## UART connections on nRF51822 and the Interface MCU

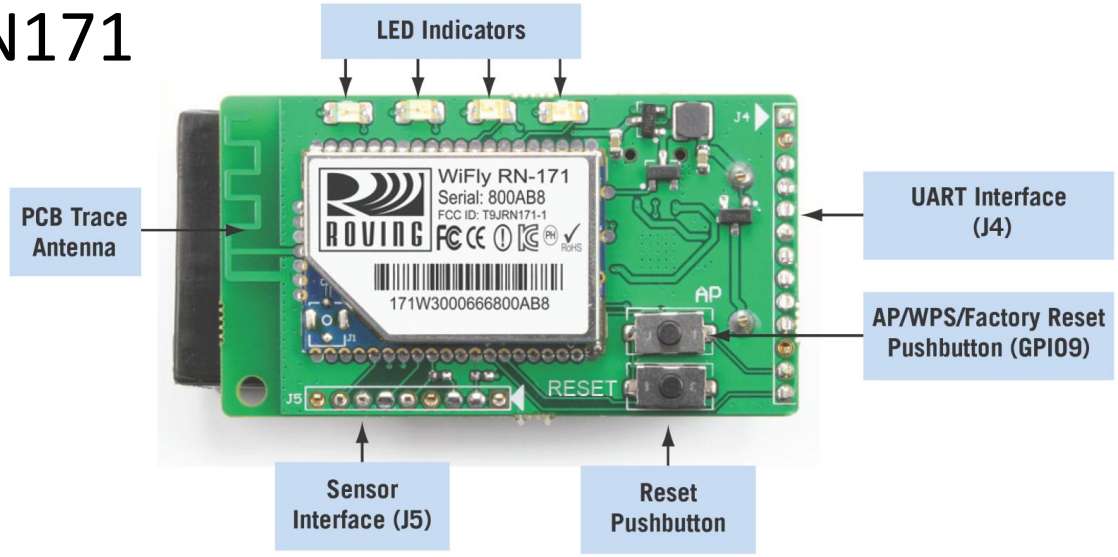
| nRF51822 | | Interface MCU |
|---|---|---|
| Default GPIO | UART | UART |
| P0.09 | TXD | RXD |
| P0.11 | RXD | TXD |

# Wi-Fi Overview

- Protocol
  - Built on the IEEE 802.11 standard (WLAN)
  - Conformance testing performed by the non-profit Wi-Fi Alliance (formed in 1999)
- Why Wi-Fi Network**?**
  - Widely used to connect a variety of devices
  - Expanding network in the future
  - Flexibility
  - Internet of Things
  - Recently TI has introduced ARM Cortex-M4 with WiFi

# Wi-Fi Module

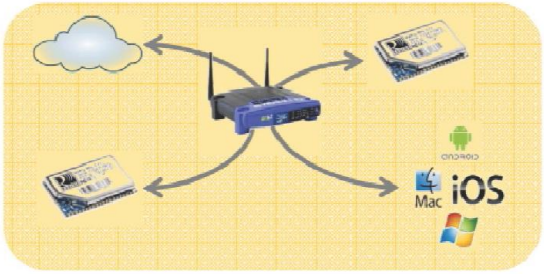
Example : RN171



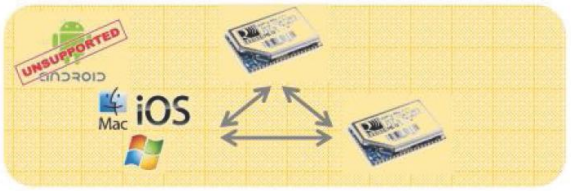

- 10 general-purpose digital I/O pins
- 8 analog sensor interfaces;
- Ultra-low power: 4 µA sleep, 40 mA Rx, 180 mA Tx at +10 dBm
- Configurable transmit power: 0 to +12 dBm
- UART hardware interface
- Up to 1 Mbps data rate over UART

# RN-171 Network Support

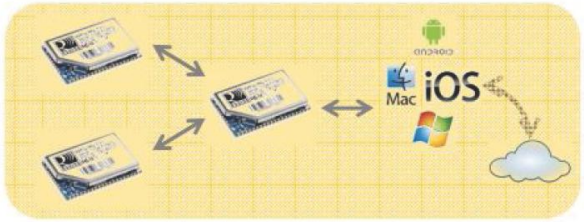
- Infrastructure
  - Client nodes communicate via an access point
  - Most common, like connecting your PC to a
  - home network




- Adhoc
  - Point-to-Point connection
  - Every node connected to every other node
  - Android unsupported




- Soft AP
  - Module looks like an Access point
  - AP module is central coordinator
  - Basic network management
  - DHCP, Routing, Gateway redirection



# Interfacing to MCU

- Usually interfaces to MCU through serial communication such as UART, USB, SPI, I2C
- Modules for standard communication usually provide either partial or complete software stack
  - Partial: User interfaces with drivers and implements stack and services
  - Complete: User interfaces to simple API
- Configuration/set up
  - Can be done in text mode  such as using  text command with terminal emulator configure before connecting to a MCU
  - Can be done by connecting to MCU and using its serial port to configure the device registers

# The End