# Chapter 0: Introduction

## Objective

After completing chapter 0 you will understand the class objectives for the Wireless Internet Connectivity for Embedded Devices (WICED) WiFi 101. You should be able to explain the learning objectives, agenda, scope of the class, and format of the lab manual.

## Time: 30 Minutes

## Prerequisites

Solid fundamentals in C-Programming (fundamental data types+operators and expressions, control flow, functions and program structure, pointers and arrays, data structures, multi-file module programming)

Some experience with standard MCU concepts and peripherals

## Scope

What this class is?

* A survey of the WICED Ecosystem (chips, modules, software development kit, ecosystem etc.)
* A survey of using the WICED WiFi SDK to create an IOT device by connecting common MCU I/O peripherals to the “Cloud.”
* An introduction to WiFi.
* An introduction to the TCP/IP protocol stack, common cloud application protocols (MQQT, HTTP, COAP) and REST
* An introduction to some cloud providers (Amazon AWS, IBM Bluemix, Microsoft Azure)

What this class is not?

* A C-programming primer.
* A detailed examination of WiFi
* An advanced network programming class.
* An introduction to Bluetooth.
* A detailed examination of fundamental MCU peripherals.
* A detailed tutorial of advanced uses of WICED (Streaming Audio, Bluetooth/Wifi Combos, TCP/IP Bridging/Routing, )

# Chapter 1: Tour of WICED WiFi

## Objective

After completing chapter 1 you should understand from a top level view, all of the components of the WICED ecosystem including the chips, modules, software, documentation, support infrastructure and development kits. You should have WICED installed and working on your computer. In addition you will have completed your first project, successfully programmed and verified its functionality.

## Time: 2 Hours

## Fundamentals

### Tour of WiFi SDK

Eclipse

Directories

App

Doc

Platform

Libraries

Resources

Tools?

Include?

Build?

### Tour of documentation

In the SDK

On the Web

### Forum

### Tour of WiFi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IEEE Standard** | **Bandwidth** | **Frequency** | **Channels** | **Channel Width** | **MIMO/SISO** | **Range** |
| [802.11a](https://en.wikipedia.org/wiki/IEEE_802.11a-1999) | 54 Mbits/s | 5 GHz |  |  |  |  |
| [802.11b](https://en.wikipedia.org/wiki/IEEE_802.11b-1999) | 11 Mbits/s | 2.4 GHz |  |  |  |  |
| [802.11g](https://en.wikipedia.org/wiki/IEEE_802.11g-2003) | 54 Mbits/s | GHz |  |  |  |  |
| [802.11n](https://en.wikipedia.org/wiki/IEEE_802.11n-2009) | 600 Mbits/s | 2.4GHz / 5 GHz |  |  |  |  |
| [802.11ac](https://en.wikipedia.org/wiki/IEEE_802.11ac) |  |  |  |  |  |  |

### Tour of Chips

4343W + 43438

4390x (43907, 43903, 4390)

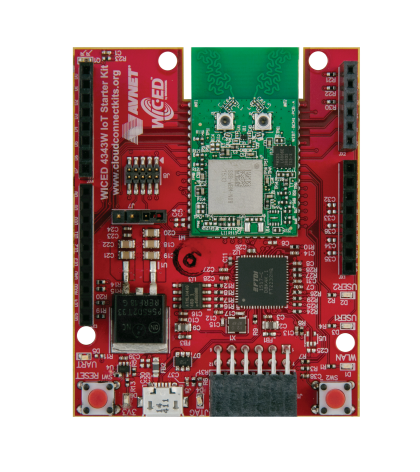
43362 + 43364

### Tour of modules

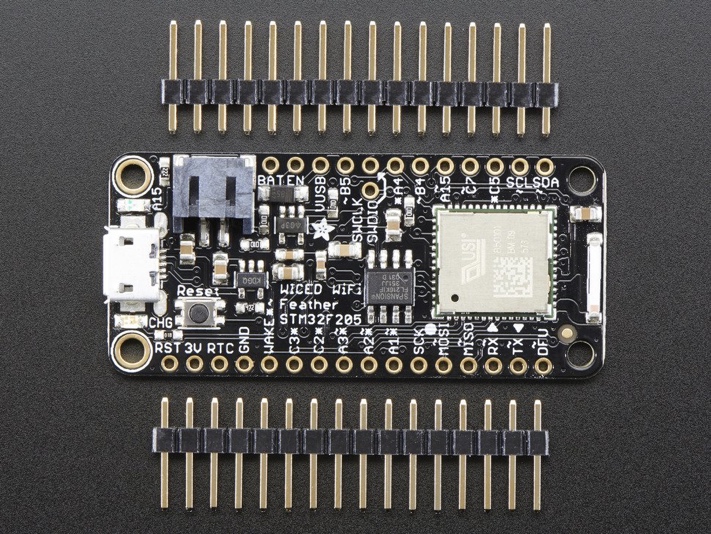
### Tour of development kits

Broadcomm Eval Kits

[Avnet Cloud Connected Kits](http://cloudconnectkits.org/product/avnet-bcm4343w-iot-starter-kit)



[Adafruit Feather](https://www.adafruit.com/products/3056)



[Electric Imp](https://www.electricimp.com/)

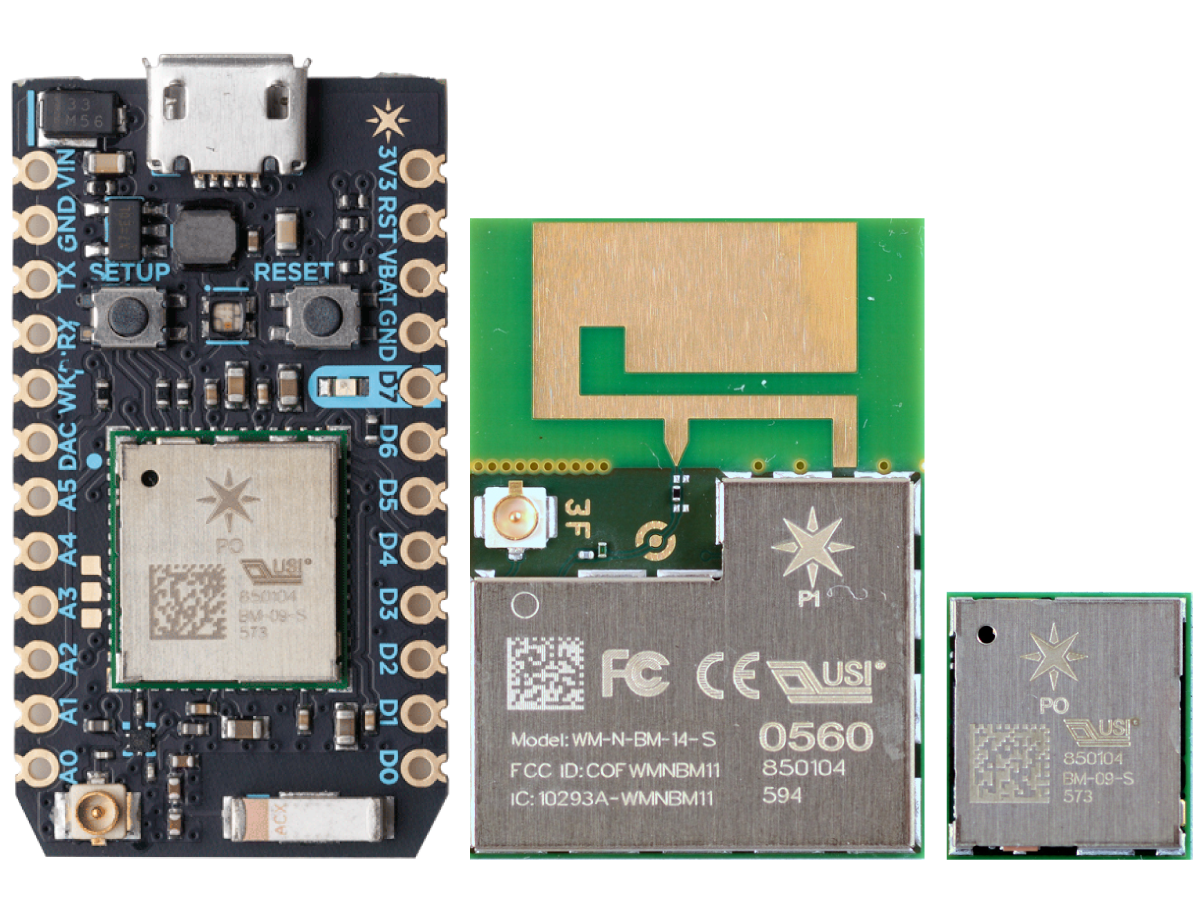
<https://www.electricimp.com/docs/attachments/hardware/product%20briefs/Electric_Imp_imp003_Product_Brief_02May2016.pdf>

<https://www.electricimp.com/docs/attachments/hardware/product%20briefs/Electric_Imp_imp005_Product_Brief_02May2016.pdf>

[Inventek](http://www.inventeksys.com/)

[Particle](https://www.particle.io/)

[Photon](https://www.particle.io/products/hardware/photon-wifi-dev-kit)



Tour of website

Example first project

Make a new project

Copy the folder snip/gpio to be snip/arhtest

Rename gpio.c to arhtest.c

Rename gpio.mk to arhtest.mk

Edit gpio.mk

Add a make target - "make config"

Add a make target "snip.arhtest-BCM94343WWCD1 download run"

## Exercise(s)

Redo example first project

Create a forum account

Open the documentation

# Chapter 2: Using the WICED SDK to connect inputs and outputs

## Objective

At the end of this chapter you should be able to write firmware interface will the MCU peripherals available in the system including the ADC, UART, SPI, GPIO’s, I2C and PWMs. In addition you should understand the role of the critical files platform.h/c.

## Time: 2 Hours

## Fundamentals

### platform.h + platform.c

How does the WICED SDK deal with pin names

How does the WICED SDK deal with peripherals

## Exercise(s)

(platforms) Make a new one for the shield

(uart)Print to screen

(gpio) Blink led you need to init the pin

(gpio) read switch How to figure out what the pins are (platform.h / platform.c)

(Uart) don’t use built in

(Pwm) led brightness

(I2c) change duty cycle on psoc

(i2c) read retries != 0

(Adc) read thermistor

(Spi) save data in the spi flash

# Chapter 3: Using threads in the WICED SDK

## Objective

## Time: 2 Hours

## Fundamentals

### Co-operative multitasking

### Thread

### Mutex

### Semaphore

### Queue

### Timer

## Exercise(s)

(thread) Create a blinking led thread

(semaphore) Button press in main thread -> semaphore to lock led thread

(mutex) make a function to lock printing

(queues) send a message to say # of times to blink

(timers) make a blinking led based on timer

Debugging threads

(worker threads)

(events)

(event flags)

# Chapter 4: Using the WICED-SDK Library

## Objective

## Time: 0 Hours

## Fundamentals

## Exercise(s)

(File system)

(Graphics)

# Chapter 5: Connecting to Access Points (AP)

## Objective

## Time: 1 Hr

## Fundamentals

### Networking Stack

### WiFi

#### (physical layer) Channels

#### (data link layer) SSID

#### (data link layer) Encryption

##### OPEN

##### WEP

##### WPA2 PSK or Personal

##### WPA

### DHCP

### Netmask

## Exercise(s)

WICED\_DCT

Find access points

Security (wep, wpa

Dhcp, dns,

connect/reconnect

# Chapter 6: Putting/Getting data to/from the Cloud

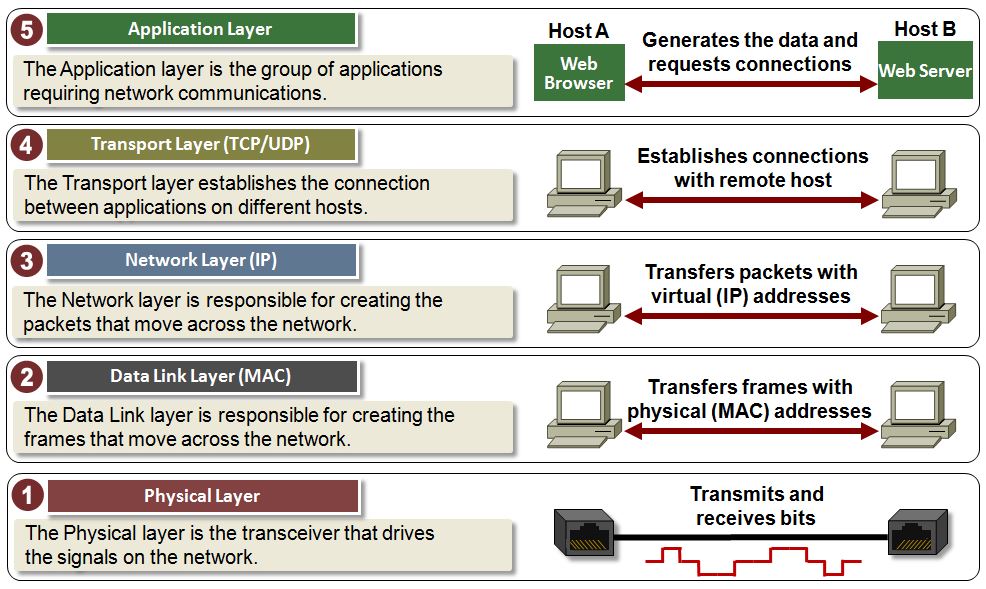
## Objective

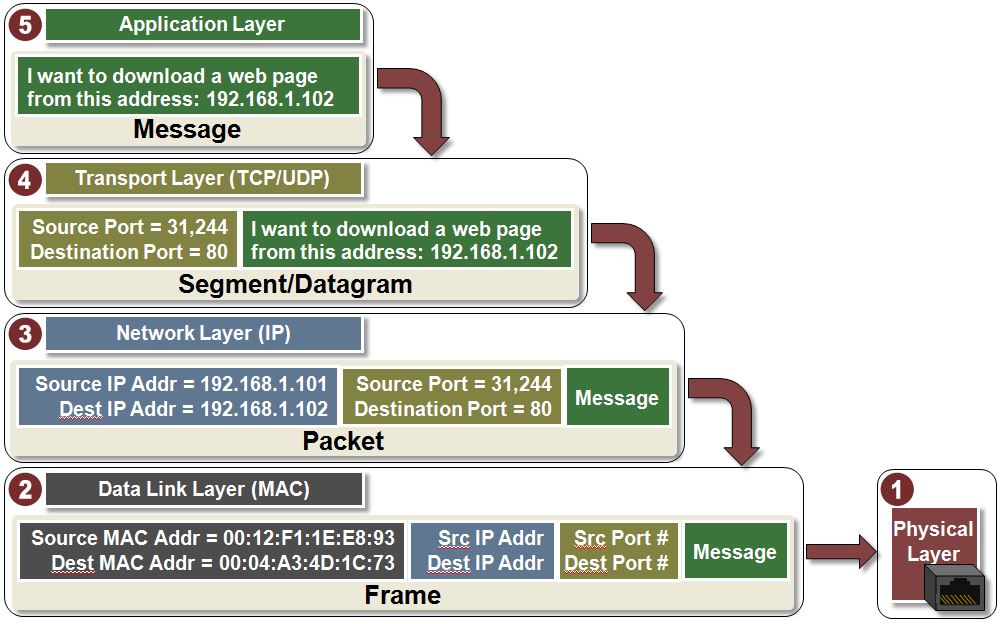
## Time: 4 Hours

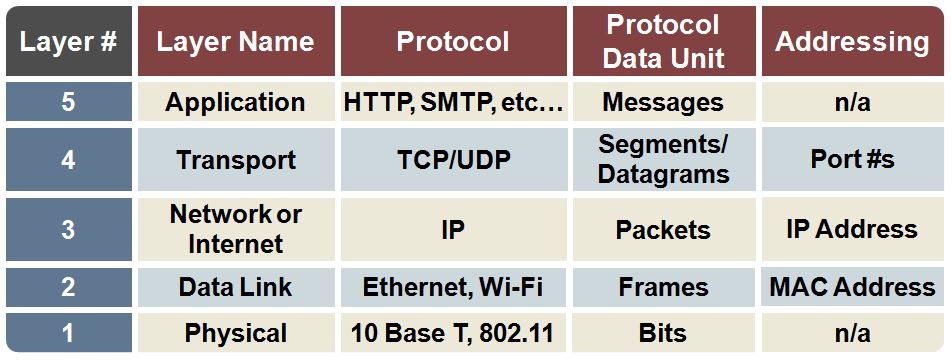
## Fundamentals

### Networking Stack the Application Layer

http://microchip.wikidot.com/tcpip:tcp-ip-five-layer-model







#### HTTP

#### MQQT

#### COAP

#### Sockets

#### AMQP

### Clouds

#### AWS (MQTT)

##### <http://www.slideshare.net/PeterREgli/mq-telemetry-transport>

##### Introduction + Setup

##### Publisher

##### Subscriber

##### Shadow

#### IBM Bluemix

#### Ali Cloud

#### Microsoft Azure

#### Cypress (sockets)

## Exercise(s)

### Aws shadow

### Aws pub/sub

# Chapter 7: Class Project

## Objective

## Time: 4 Hours

## Fundamentals

## Exercise(s)

### Get weather data and display

### Set the city(s) to display

### Rotate/Scroll the display

### Weather alerts

### Become an AP

### Let people login

# Chapter 8: WICED Academy Shield

## Description

## Features

### Psoc4 (Analog coprocessor)

### Leds

### Capsense

### PIR

### ALS

### I2C->Ardunio pins

### Buzzer -> Ardunio pin

### Arudino pins

### Buzzer (on the PSoC and On the Dx pin)

### Thermister (on the PSoC as Differential + on an Ax pin)

### I2C --> WICED

### UART -> WICED

### WICED device

### LEDs

### I2C --> PSOC

### UART --> WICED

### Cypress Spi Nor flash (WICED SPI)

## Schematic

# Chapter 9: Glossary (name, taxonomy, wikpedia link, website link)

802.11a

802.11b

802.11n

802.11ac

Ali Cloud

[AMQP](https://en.wikipedia.org/wiki/Advanced_Message_Queuing_Protocol) – [www.amqp.org](http://www.amqp.org) Advanced Message Queueing Protocol.

Azure – see Microsoft Azure

Bluemix – see IBM Bluemix

[COAP](https://en.wikipedia.org/wiki/Constrained_Application_Protocol) – Constrained Application Protocol. Constrained Application Protocol (CoAP) is a software protocol intended to be used in very simple electronics devices, allowing them to communicate interactively over the Internet. It is particularly targeted for small, low-power sensors, switches, valves and similar components that need to be controlled or supervised remotely, through standard Internet networks. CoAP is an application layer protocol that is intended for use in resource-constrained internet devices, such as WSN nodes. CoAP is designed to easily translate to HTTP for simplified integration with the web, while also meeting specialized requirements such as multicast support, very low overhead, and simplicity.[1][2] Multicast, low overhead, and simplicity are extremely important for Internet of Things (IoT) and Machine-to-Machine (M2M) devices, which tend to be deeply embedded and have much less memory and power supply than traditional internet devices have. Therefore, efficiency is very important. CoAP can run on most devices that support UDP or a UDP analogue.

[DHCP](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol) – Dynamic Host Configuration Protocol

[DNS](https://en.wikipedia.org/wiki/Domain_Name_System) – Domain Name System

Gedday -

[HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) – Hyper Text Transfer Protocol

IE

IP – Internet Protocol

IBM Bluemix

Microsoft Azure

MIMO – Multiple In/Multiple out. In 802.11n/ac you can increase the bandwidth by bonding multiple channel together (e.g. 2x channels will double the bandwith)

[MQTT](https://en.wikipedia.org/wiki/MQTT) – [WWW.MQTT.ORG](http://WWW.MQTT.ORG) MQTT[1] (formerly Message Queueing Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922)[2] publish-subscribe-based "lightweight" messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message.

Mutex –

OASIS –

OSI Model -

OTA – Over the Air

Queue –

REST –

Semaphore –

SISO -

Sockets -

SSDP – Simple Service Discovery Protocol

TCP/IP

Timer -

[TFTP](https://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol) – Trivial File Transfer Protocol

Thread –

WPS -

# Chapter 10: Issues, Questions, Errata

MFI/Homekit?

Security

Cryptogtraphy

# Chapter 12: Become an Access Point (another class)

## Objective

## Time: 0 Hours

## Fundamentals

## Exercise(s)

### Addresses

### Web server