SEICHE 2022

Introduction to Arduino

Lesson 4 and 5– IDE Essentials

Instructor: Paul Frommeyer

www.paulfrommeyer.com

Corporate Sponsor: DXC Technology

REMEMBER!

Immediately copy the latest lesson folder (Lesson5 today) from your USB flash drive to your hard drive where you previously copied the other flash drive files (probably Documents folder)

Lesson Plan Overview

Lesson 1 – Intro and Setup
[may require 2 classes]
- Introduction to class format
- Overview of lesson plan
- Presentation format (monitor, camera, screen, whiteboard)
- Inventory of Arduino kits
- Inventory of USB drives
- Installation of ESP32 serial port drivers (Windows only)
- Installation of Arduino software
- History of Computing

Lesson 2 and 3 – Laptop operation review
- Control panel/Settings location
- Home directories and folder hierarchy
- Arduino file locations
- Search functions
- Open questions and issues
- History of Computing redux

Lesson 4 and 5 – IDE essentials
- The boards manager
- Setting board type
- Installing Libraries
- Loading example sketches
- Finding and selecting the Arduino USB port
- Uploading a sketch to the microcontroller [Example Sketch#1]
- Basic Arduino sketch (program) structure
- Uploading your own sketch [Example Sketch #2]
- History of Telecommunications and Networking – Part 1

Lesson 6 – DC Electricity Basics and DMM Intro
- Quick review: DC vs AC electric currents
- DC current operation
- Voltage vs Amperage (current)
- LEDs, constant current, and dropping resistors
- Connecting an LED to a microcontroller [Class Sketch#1]
- Ohm's Law
- Introduction to basic DMM functions and usage
- Series vs parallel circuits
- History of Telecommunications and Networking – Part 2

Lesson 7 – Elementary Components
- Resistors [DMM usage #1]
- Diodes and LED's
- Capacitors: Electrolytic (polarized) vs non-electrolytic (non-polarized)
- Simplified transistor operation
- Connecting a transistor to a microcontroller [Class Sketch #2]

Lesson 8 – ESP32 External power sources [may require 2 classes]
- Microcontroller board power requirements
- Board power input options
- Components with different voltage requirements
- External power circuit design

Lesson 9 – Connecting LED's to the ESP32 controller board
- Microcontroller pinouts and diagrams
- Different types of pins
- Analog input
- Digital Input and digital output

Lesson 10 – Connecting smart LED strips
- Simplified serial communication protocols
- One-wire serial communication protocol
- Hardware vs software serial
- WS2812B and Neopixel connections
- External device power design

Lesson 11 – Connecting LCD and Matrix Displays
- The SPI serial protocol
- SPI connections
- SPI device external power design

Lesson 12 – Connecting Sensors [Time permitting]
- I2C serial protocol
- I2C connections
- I2C sensor board power design and external power

Lesson 13 – Connecting other sensors [Time permitting]
- Connecting buttons
- Connecting potentiometers
- Reading buttons
- Reading potentiometers

Lesson 4 and 5 – IDE Essentials

- The Arduino IDE
- The Boards Manager – Adding a Board
- The Library Manager – Installing Libraries
- Selecting a board
- Accessing example sketches
- Finding and selecting the Arduino USB port
- Uploading a sketch to the microcontroller [Example Sketch#1]
- Basic Arduino sketch (program) structure
- Loading our own sketches
- History of Telecommunications and Networking

Lesson 4 and 5 – Arduino IDE

- IDE stands for “Integrated Development Environment”
- A single application can perform all operations needed to program one or more boards
- Creation of programming code (C++ for Arduino)
- Preparing (compiling) code for upload to a board
- Installing and selecting software Libraries which add operations and functionality created by others
- Provide basic communications with a board
  - Uploading code (sketches)
  - Watching and interacting with the board (Serial Monitor)

The Boards Manager

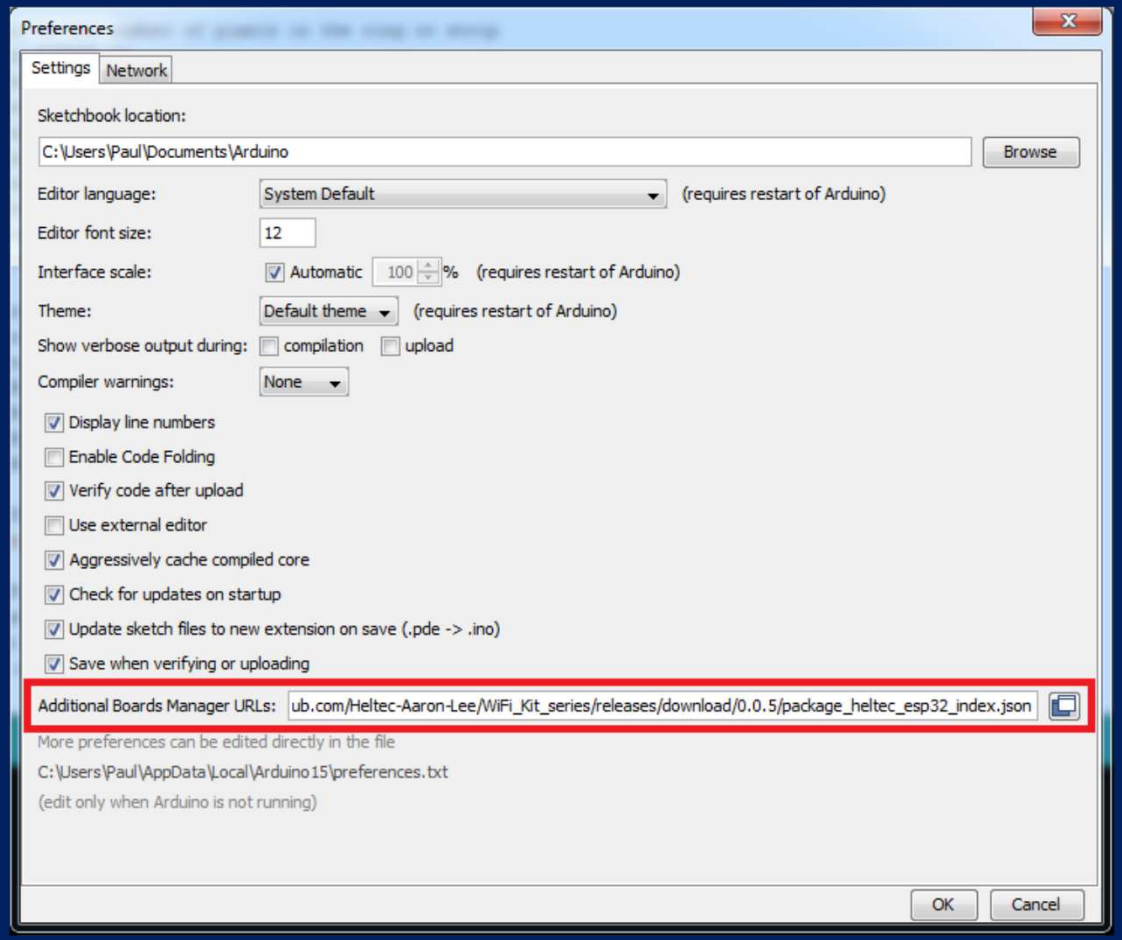
- To know how to prepare (compile) software for and talk with (communicate via USB or RS-232) a board, the Arduino IDE needs a hardware definition for each board to be programmed
- Board definitions are handled by the “Boards Manager”
- The Boards Manager itself uses URLs to locate board definitions on the Internet, usually provided by the manufacturer
- URLs are entered in Preferences
- The Boards Manager itself is accessed from the Tools menu

Let’s make sure everything is set up correctly for our HelTec WiFi Kit 32 board with its ESP32 microprocessor…

Boards Manager – Adding A Board

ADD A BOARD #1

- Find the URL for the new board
- Open Preferences
- Add the board
- Multiple boards with commas, no space
- Filesystem browsers
- Click “OK” to close

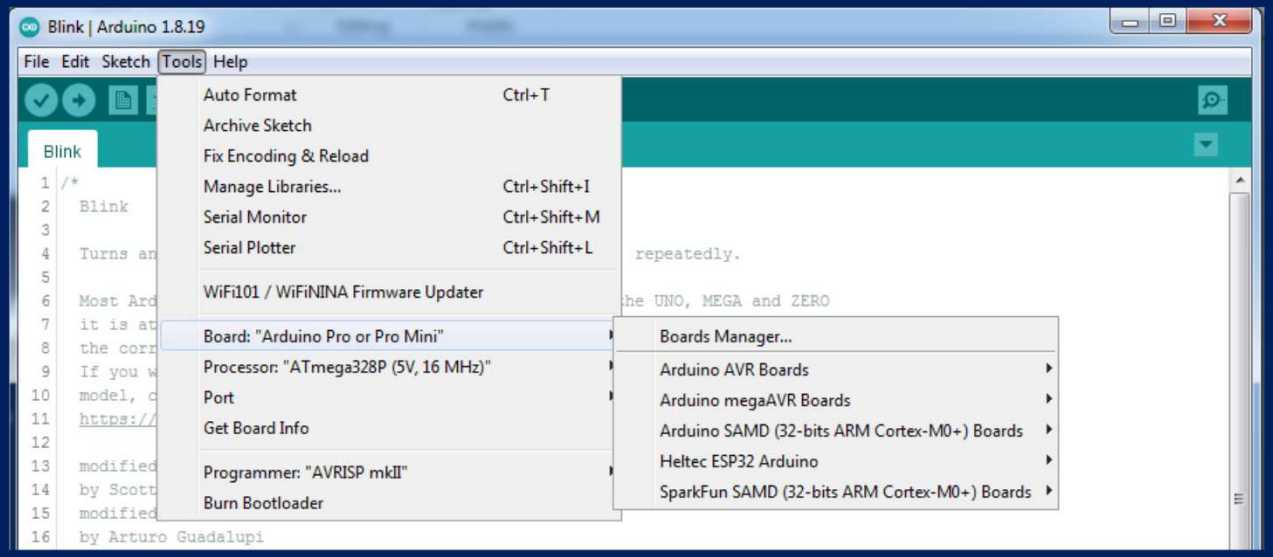


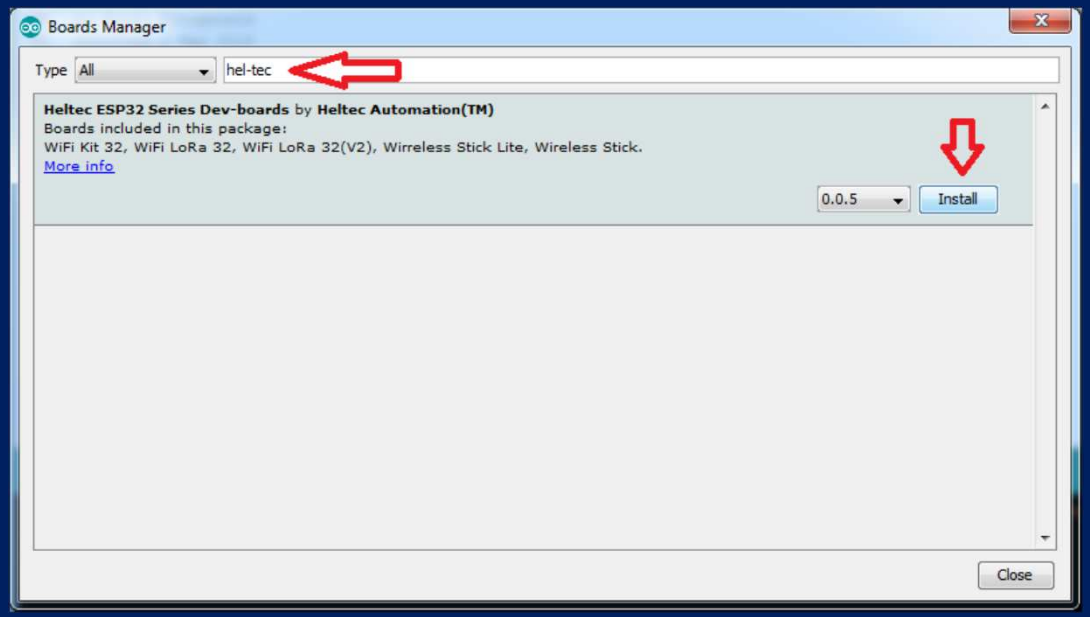
Heltec WiFi Kit 32 board definition URL (from the Heltec website)
https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series/releases/download/0.0.5/package_heltec_esp32_index.json

Boards Manager – WiFi Kit 32

ADD A BOARD #2

- Open Boards Manager under “Tools” menu
- Enter “Heltec” in search menu
- Click “Install” to install the board definitions for all Heltec ESP32 boards, including the WiFi Kit 32





The Library Manager

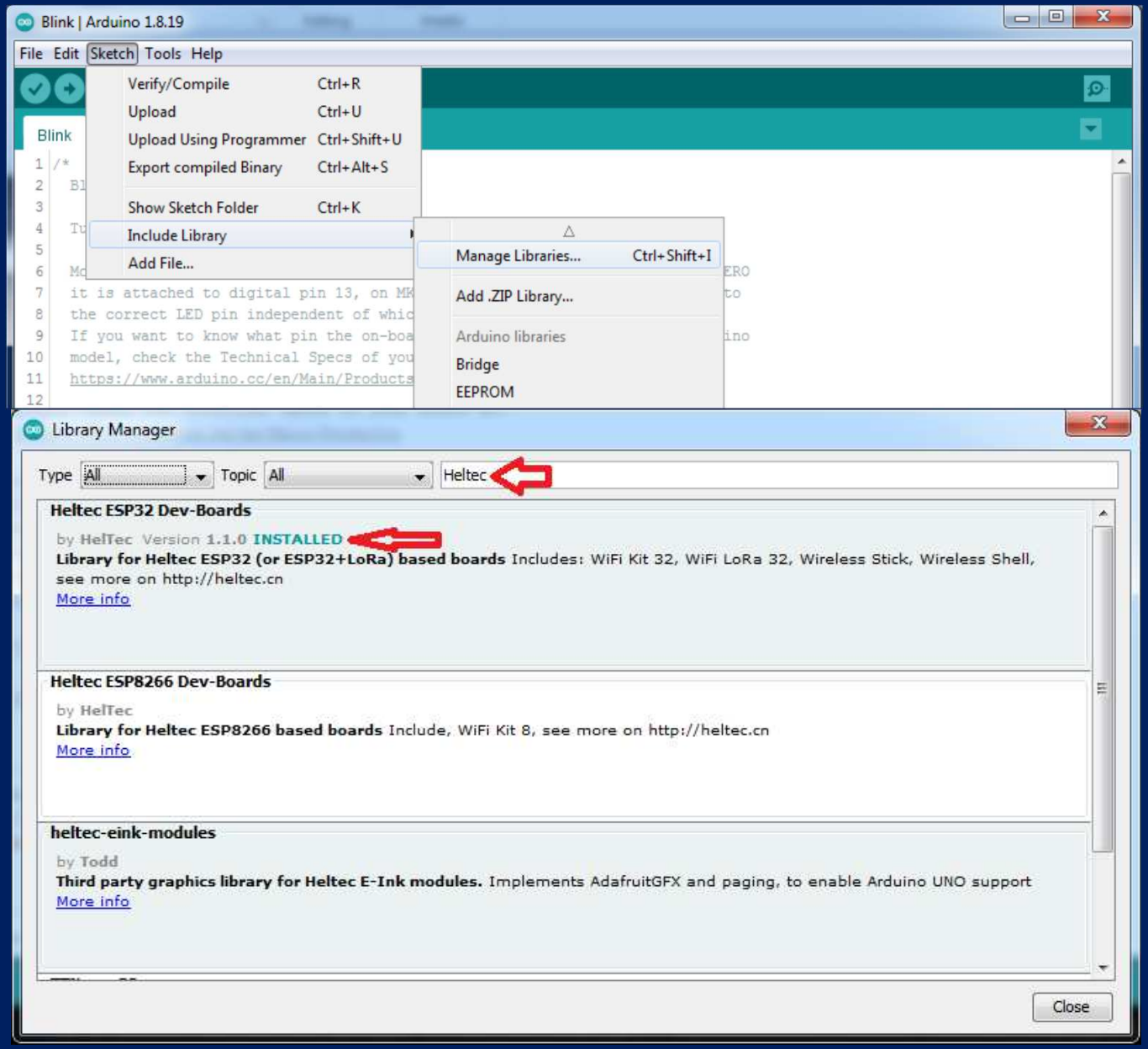
- A library is a collection of software written by others which performs a specific function or related functions
- Libraries allow us to use code written by others in our own projects
- In the Arduino IDE, the Library Manager handles the selection and installation of libraries
- Libraries are board-specific, so you must have the proper board definition(s) installed first, before you can install libraries for that board
- The Library manager is accessed from the Sketch menu

Let’s make sure everything is set up correctly for our HelTec WiFi Kit 32 board with its ESP32 microprocessor…

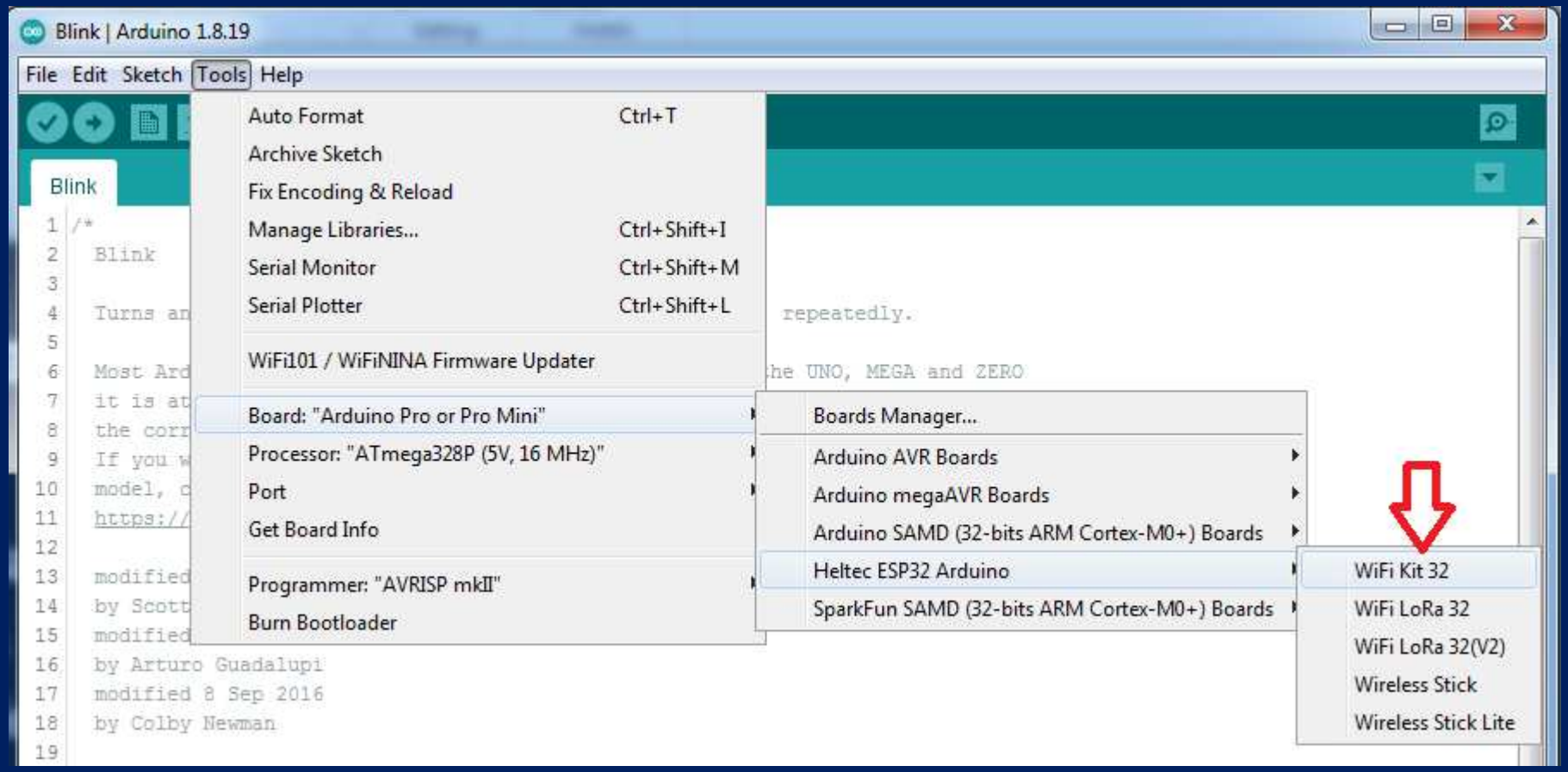
Library Manager – WiFi Kit 32

ADD A LIBRARY

- Open “Manage Libraries” under “Include Library” in Sketch menu
- Enter “Heltec” in seach box
- Install the Heltec ESP32 Dev-Boards library



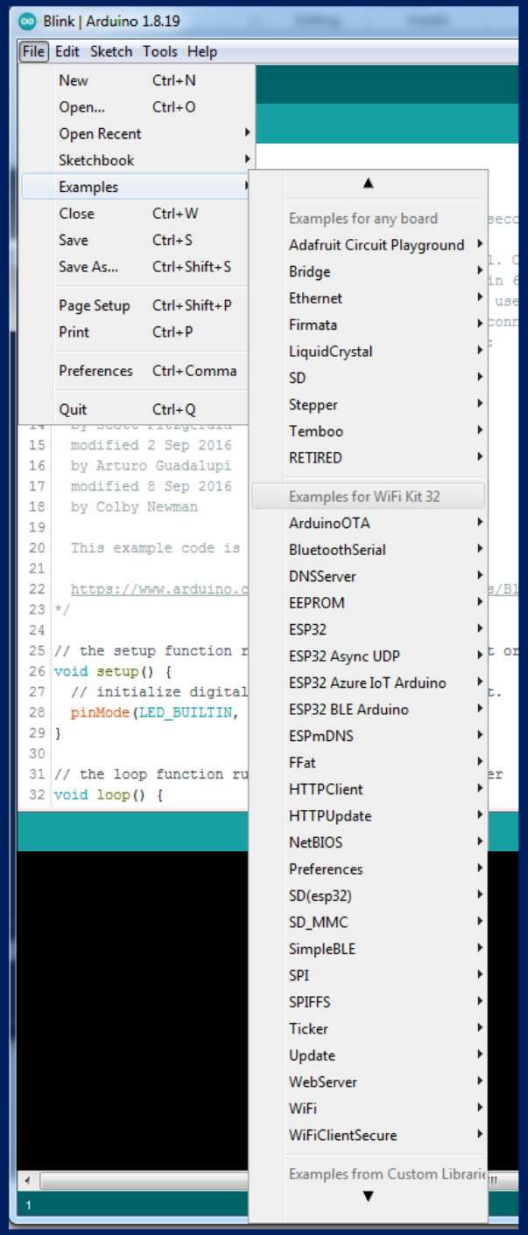
Selecting A Board



- Return to the Board menu under the Tools menu
- Navigate to Heltec ESP32 Arduino
- Select WiFi Kit 32

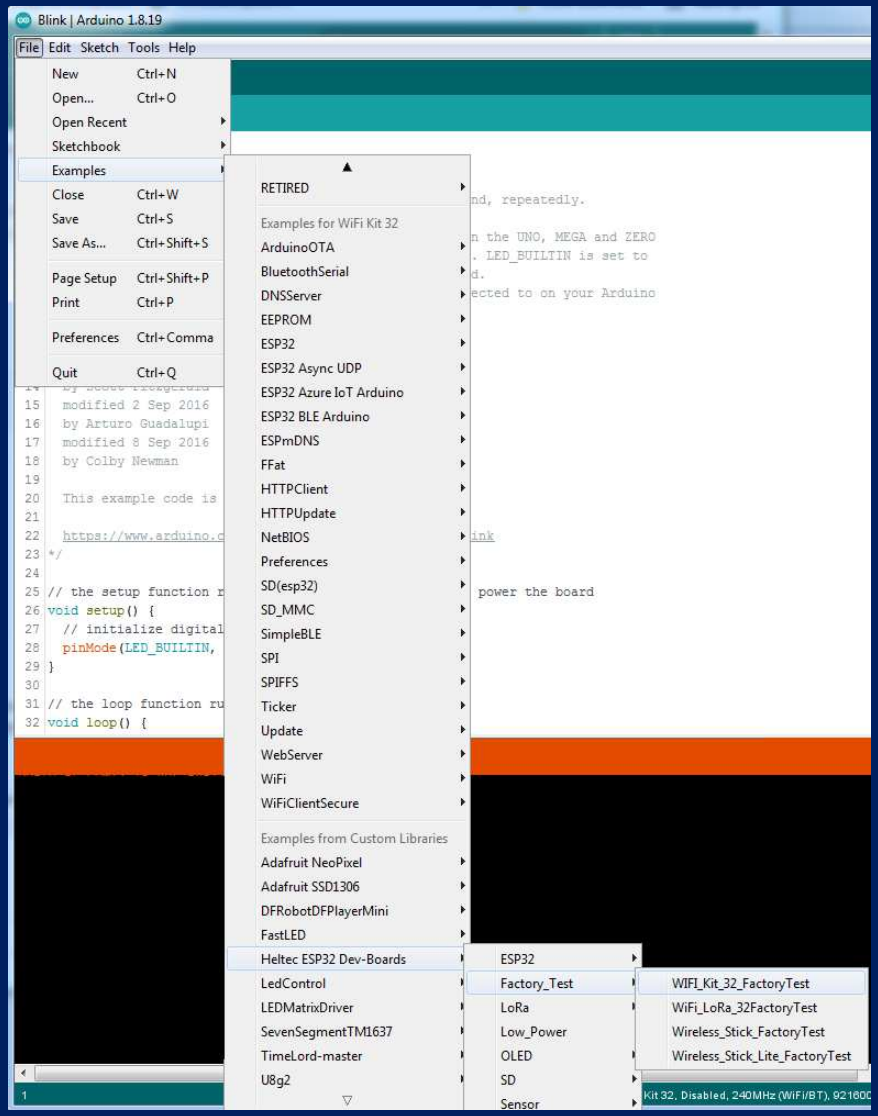
Accessing Example Sketches

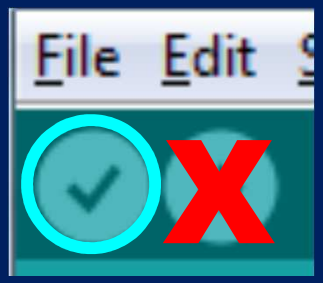
- File->Examples
- At the very bottom of the menu you should see all the relevant examples for the WiFi Kit 32 included with the Heltec ESP32 library just installed
- As you can see, there are dozens upon dozens available
- Typical for all boards
- What makes “plug and play” Arduino use possible
- And thus this class



Preparing a Sketch

- Find the WiFi_Kit_32_FactoryTest example sketch under Heltec ESP32 DevBoards sub-menu
- Click on the “VALIDATE” check mark icon
- This will compile the sketch and check for errors prior to upload (this step is optional)





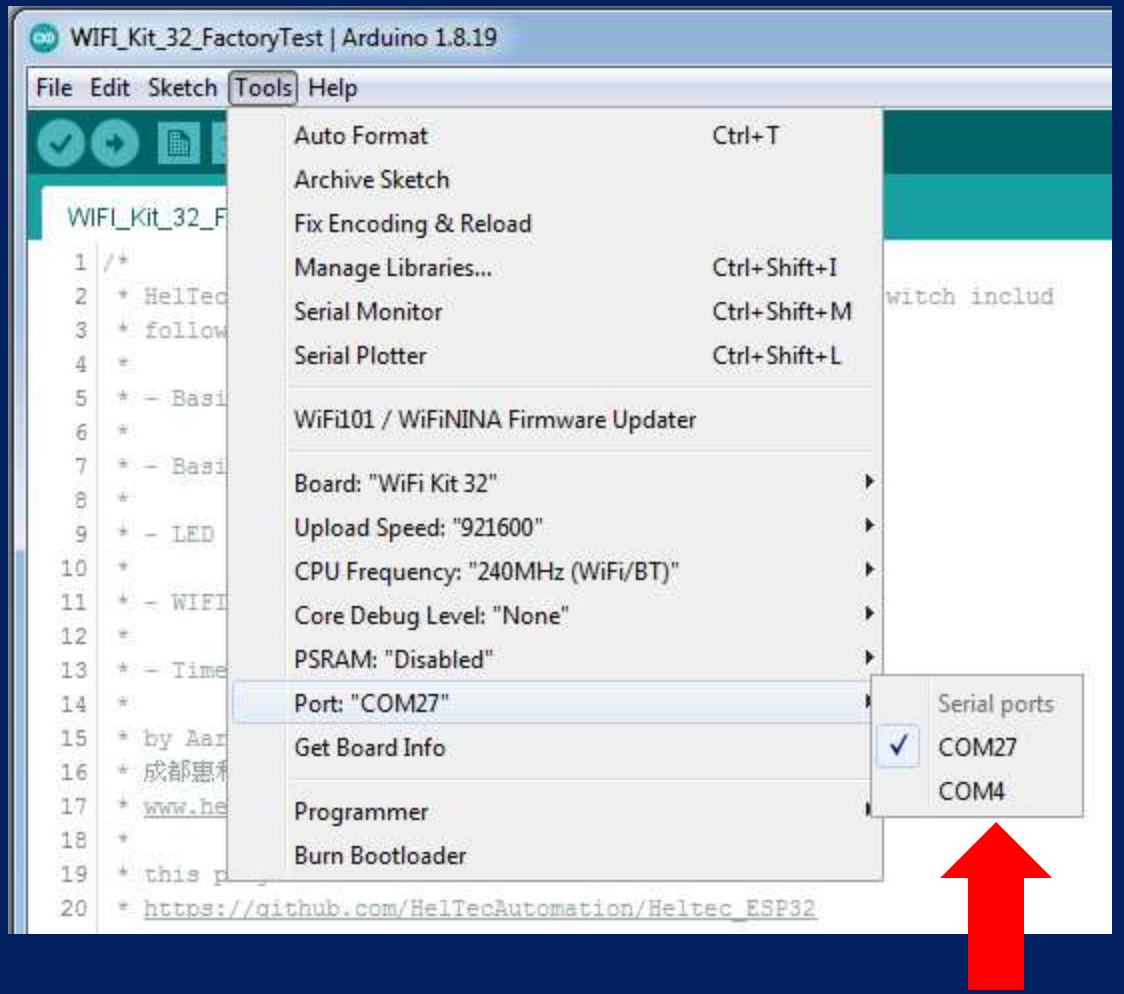
Finding and Setting the USB Port

WINDOWS

- Use Device Manager to identify the USB port
- The port will change with each connect/disconnect of the board

KUBUNTU

- Port should always be /dev/ttyUSB0
- The port will not change with connect/disconnect
- You can use Konsole to verify the port presence `ls /dev/tty*`

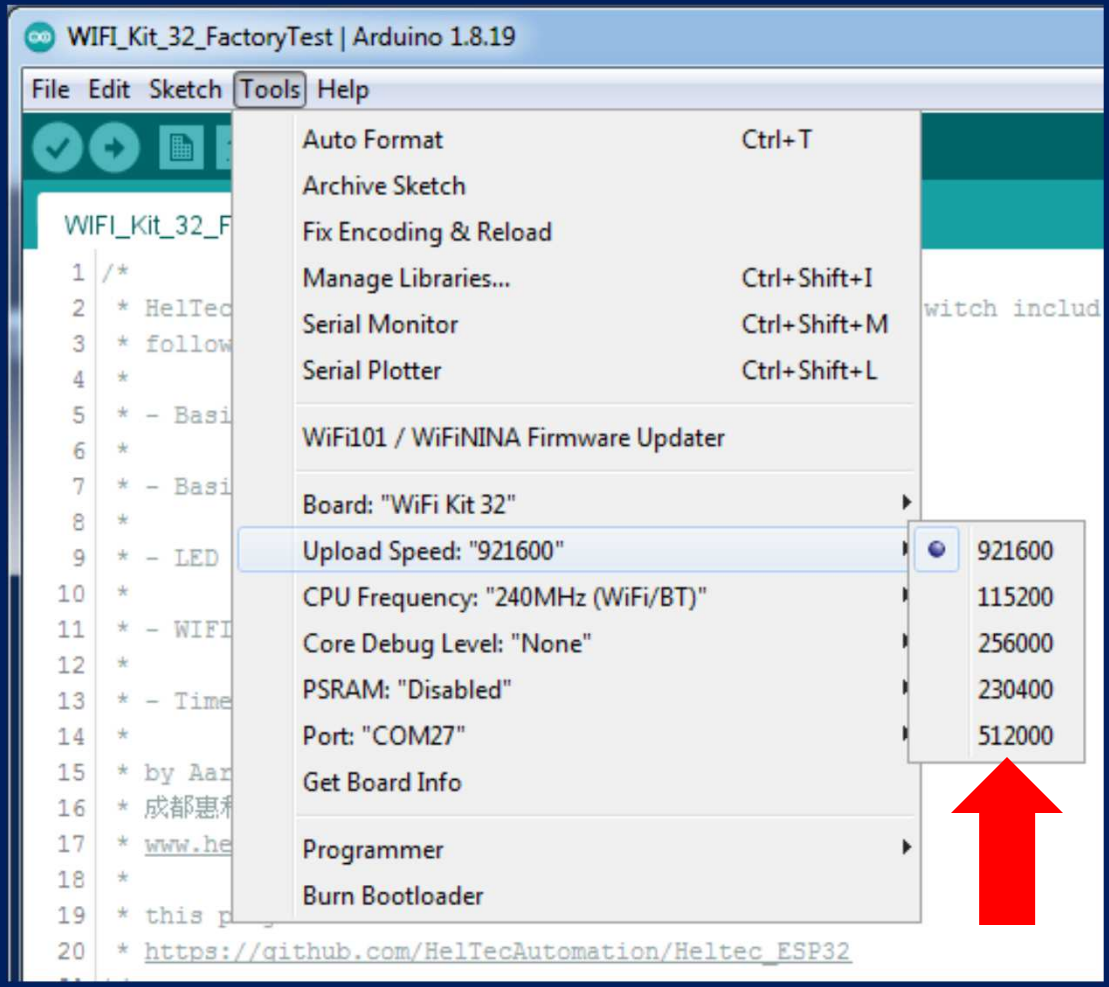


Note: Late breaking information not included in Lesson4

Adjusting Port Speed

CAUTION

- The SiL CP2102 USB-to-Serial chip used on all ESP32 boards has a known bug that can cause the serial port speed to randomize when the board is connected
- Infrequent with Windows
- Often with all Linux distros



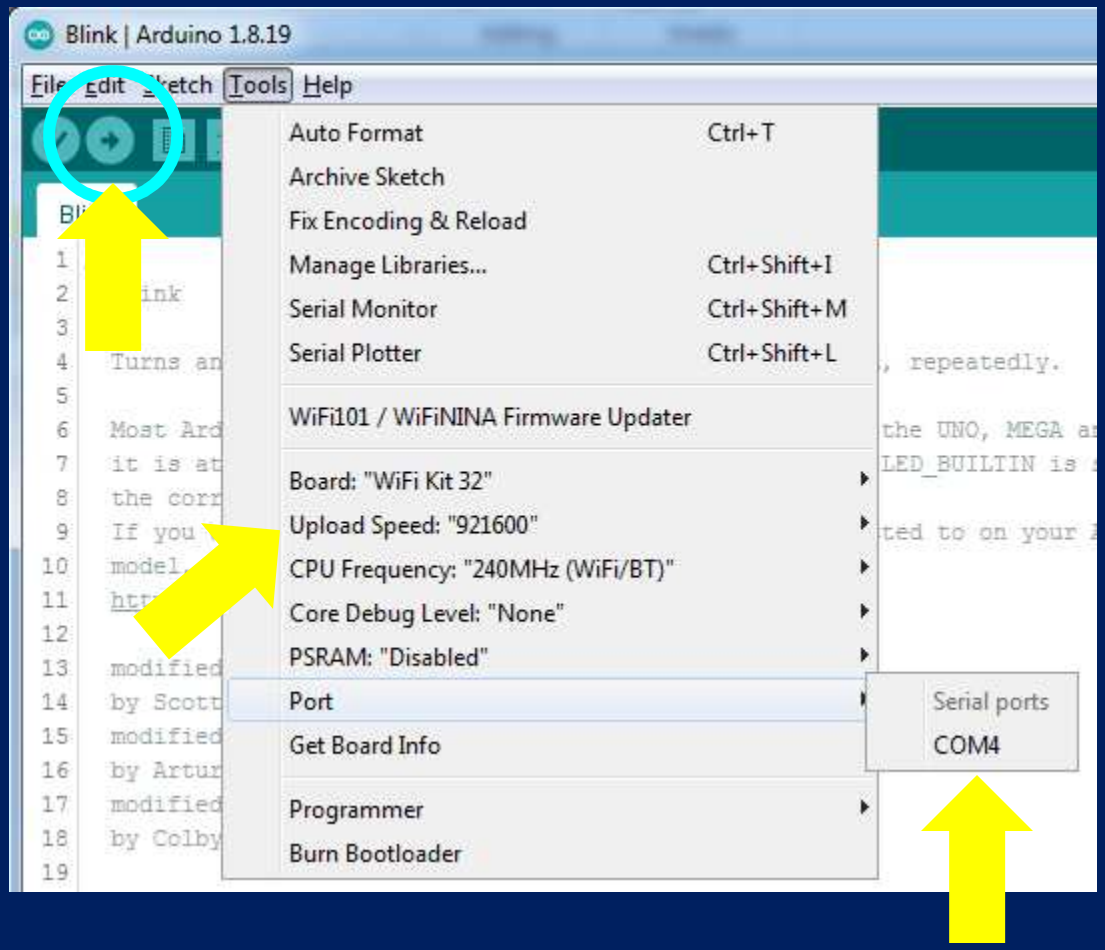
If you get an upload failure

- Recheck port setting
- Retry the upload with the next highest speed
- Lather, rinse, repeat until success
- Suggested initial try order: 921600, 512000, 115200

Note: This slide accidentally omitted from Lesson4

Uploading a Sketch

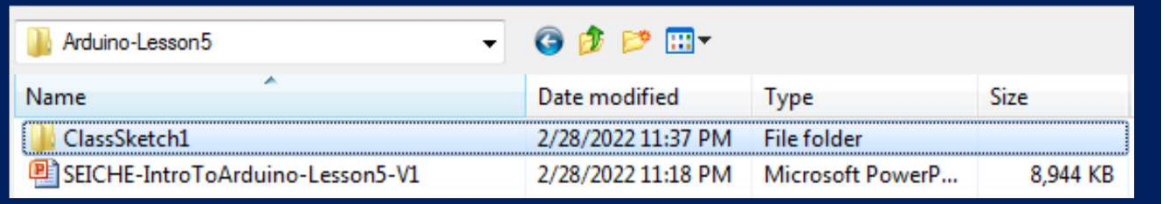
- Ensure you have selected the correct port (and speed!) for your currently plugged in board
- Click on the “UPLOAD” arrow icon
- Lights should flash on your board
- Once upload is finished, the sketch will immediately run
- If you get an error, recheck board, port and speed settings (refer to preceding slides)



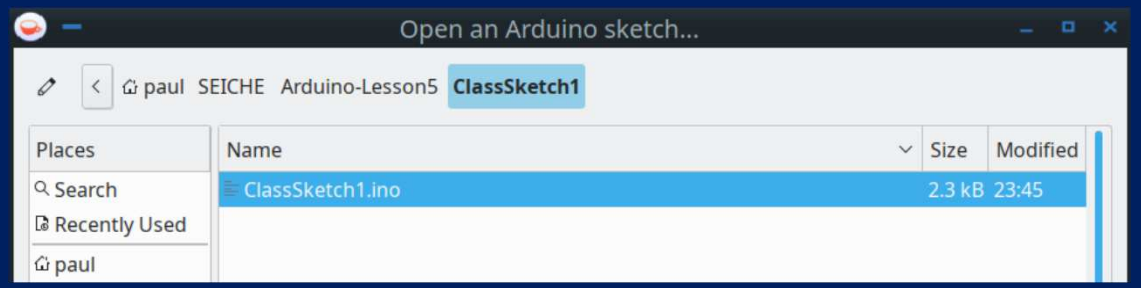
Let's load our own sketch!

- File menu->Open
- Navigate to the"ClassSketch1.ino" file in the **Lesson5->ClassSketch1** folder you copied from your USB drive at the beginning of class
- The sketch will open in the IDE
- Click on the "UPLOAD" arrow icon
- Once compile and upload are finished, the sketch will immediately run

Windows



Kubuntu



Note
Sketches live in folders *with the same name*

Lessons 1-5 Skills Review

At this point you should now be able to:

- Connect your WiFi Kit 32 board to your computer
- Have the computer recognize the board
- Identify the board serial port (Device Manager if Windows)
- Start the Arduino IDE
- Configure and select board type in the Board Manager
- Configure and select matching library in the Library Manager
- Select the serial port and port speed in the IDE
- Load an example sketch or your own sketch
- Validate (compile) a sketch
- Upload a sketch to your board

You can now start experimenting with different sketches from the WiFi Kit 32 library examples!

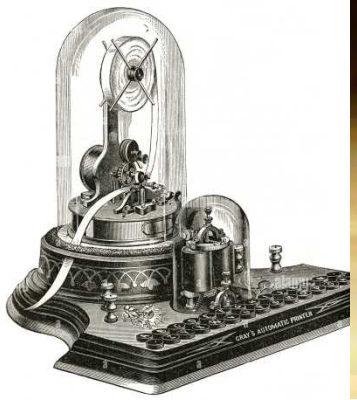
History of Telecommunications Part 1

- The Telegraph
- The Analog Telephone
- Radio (AM then FM)
- Automated dialing
- Digital audio
- Dedicated computer links (DDS)
- The first modem

150 Years of Telecom—From the Telegraph to the Internet; Room 108E

The Telegraph

- The archetype of this device was the Morse system, invented by Samuel Morse in 1838, using a single wire.
- Morse system used Morse Code, a cipher which translated letters into long and short electrical pulses sent with a make/break signalling key
- These pulses could be received as clicks or as audio beeps (see Radio section)
- There were also optical telegraphs
- The first printing telegraphs would punch holes in paper tape
- Later printing telegraphs used letters generated by special codes (Baudot, ITA2)
- Electrical telegraphy can be considered to be the first example of electrical engineering,[5] and was used by the emerging railway companies to develop train control systems that minimised the chances of trains colliding with each other.



LETTERS / FIGURES		A –	B ?	C :	D WHO ARE YOU	E 3	F %	G @	H £	I 8	J BELL	K (	L)	M .	N ,	O 9	P 0	Q 1	R 4	S '	T 5	U 7	V =	W 2	X /	Y 6	Z +	CARRIAGE RETURN	LINE FEED	LETTERS	FIGURES	SPACE	ALL-SPACE NOT IN USE
CODE ELEMENTS	1	●	●		●	●	●				●	●						●		●		●		●	●	●	●			●	●		
	2	●	o	●	o	o	o	●	o	●	●	●	●	o	o	o	●	●	●	o	o	●	●	o	o	o	o		●	●	●	o	o
	3	o	o	●	o	o	●	o	●	●	o	●	o	●	●	o	●	●	o	●	●	●	o	o	●	●	o		o	●	●	●	o	
	4			●	●		●	●		o	●	●		●	●	●		o	●			o	●	●	●					●	●		
	5		●					●	●				●	●		●	●	●			●		●		●		●			●	●		

● INDICATES A MARK ELEMENT (A HOLE PUNCHED IN THE TAPE)
o INDICATES POSITION OF A SPROCKET HOLE IN THE TAPE

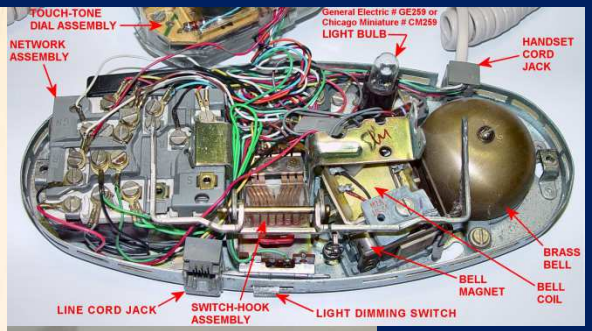
The International Telegraph Alphabet

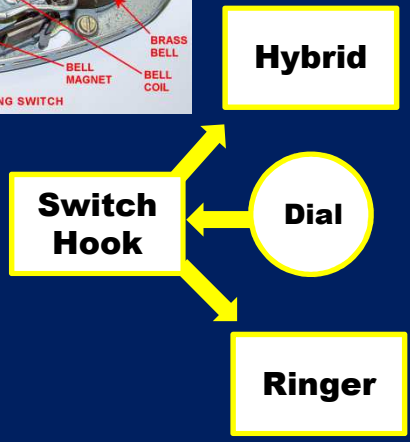
A ●–	J ●–––	S ●●●
B –●●●	K –●–	T –
C –●–●	L ●–●●	U ●●–
D –●●	M ––	V ●●●–
E ●	N –●	W ●––
F ●●–●	O –––	X –●●–
G ––●	P ●––●	Y –●––
H ●●●●	Q ––●–	Z ––●●
I ●●	R ●–●	

Morse Code

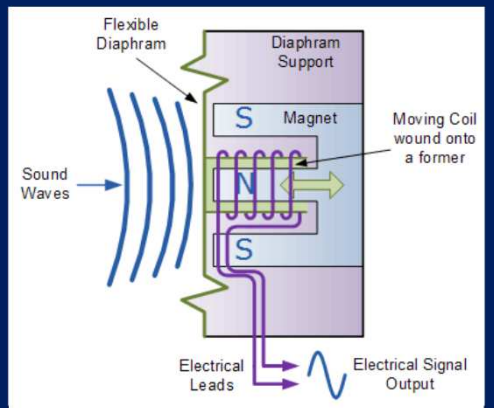
The Analog Telephone

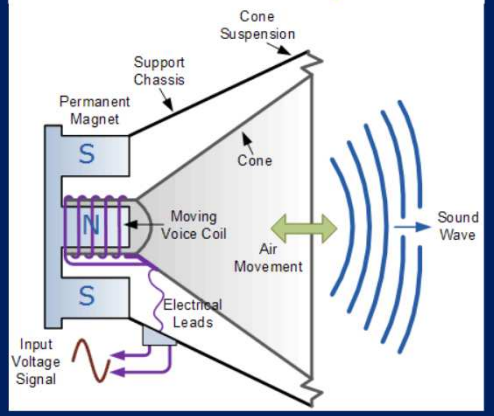
- Invented by Alexander Graham Bell,
- Initially, no station originated dialing – required operators to place and deliver calls
- Phone sets had cranks to signal the operator
- Analog phones use a “hybrid” to allow talk and hearing on 2 wires
- Was a landmark invention in human communication





Parts of an analog telephone



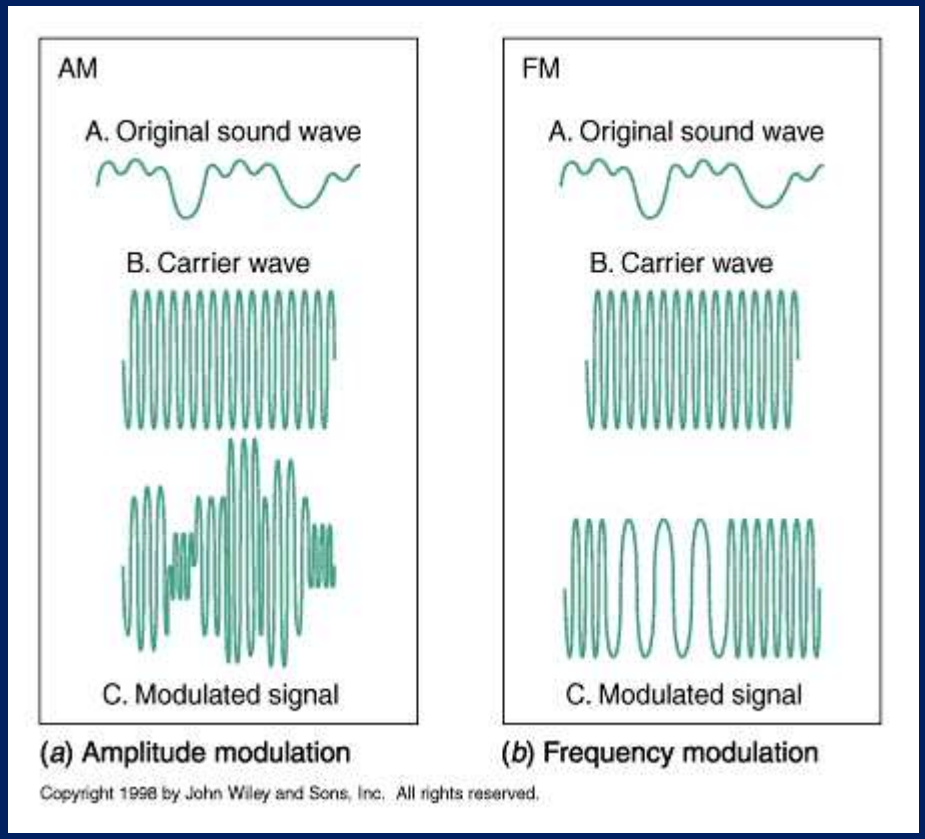


Radio

- Guglielmo Marconi sent and received first radio signal in Italy in 1895
- Initially, radio used Morse code to modulate (interrupt) spark gap transmitters
- The person generally credited as the primary early developer of AM technology is Canadian-born inventor Reginald Fessenden
- AM radio is monophonic (only one audio channel)
- The development of radio was made possible by the invention of vacuum tubes, which permitted amplification of weak electrical signals

Frequency Modulation, with amazing clarity and no static, was invented in 1933 by American engineer Edwin Armstrong. However, it's adoption was slowed by RCA for over 30 years and it did not come into widespread use until the 1960's with stereo broadcasting

Guglielmo Marconi

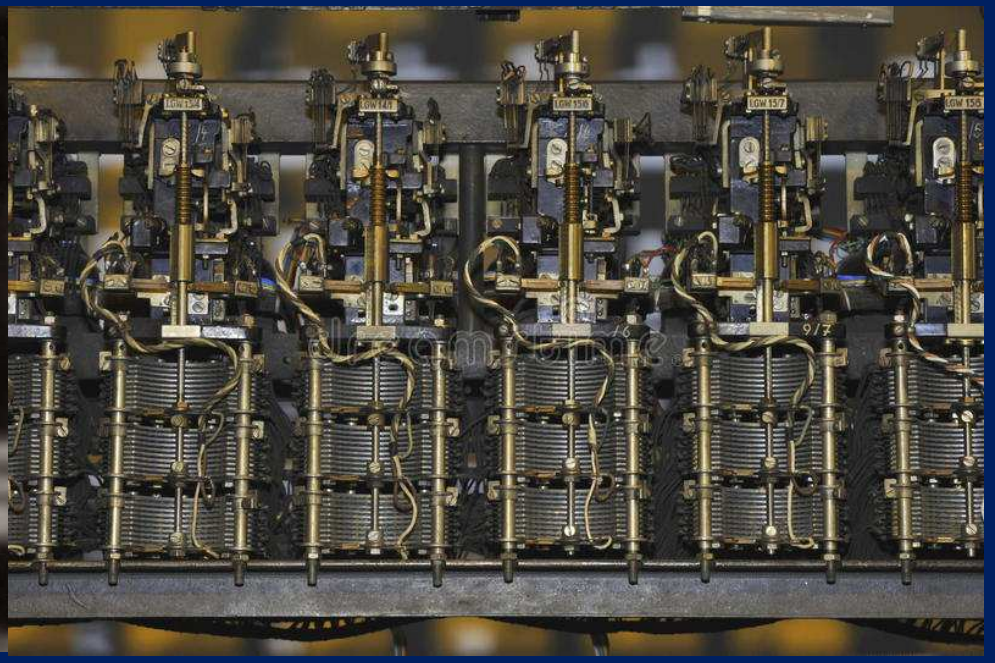


AM vs FM

Automated Telephone Dialing

- Originally, all calls were manually switched through operator's patch cord consoles
- The Strowger switch is the first commercially successful electromechanical stepping switch telephone exchange system, developed by the Strowger Automatic Telephone Exchange Company in 1891
- Automatic switches allowed end-station dialing, initially using a rotary dial
- The transistor age allowed the development of touch-tone dials, which used Dual-Tone Multi-Frequency (DTMF) in-band signalling for dialing

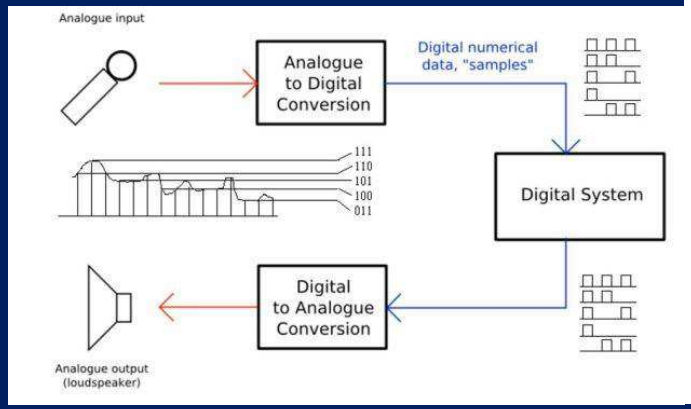
Traditional telephone operator's console



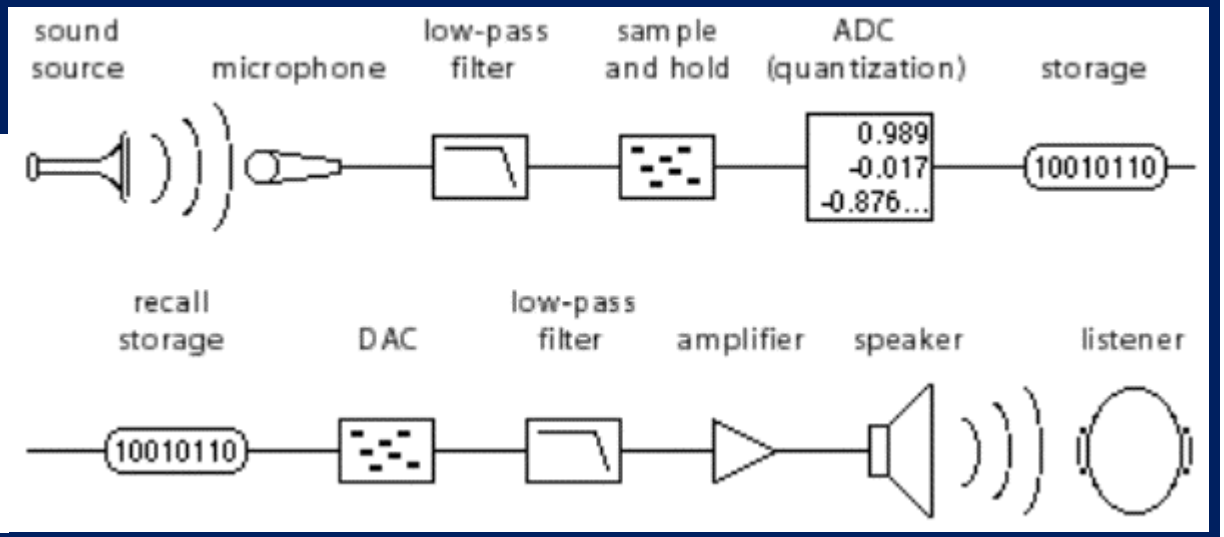
Telephone rotary stepper switches

Digital Audio

- Digital audio is a representation of sound recorded in, or converted into, digital form. In digital audio, the sound wave of the audio signal is typically encoded as numerical samples in a continuous sequence.
- Primarily invented by Claude Shannon
- Pulse-code modulation (PCM) is a method used to digitally represent sampled analog signals. It is the standard form of digital audio in computers, compact discs, digital telephony and other digital audio applications.
- US telephone system (AT&T) began using PCM in 1961



Quantization

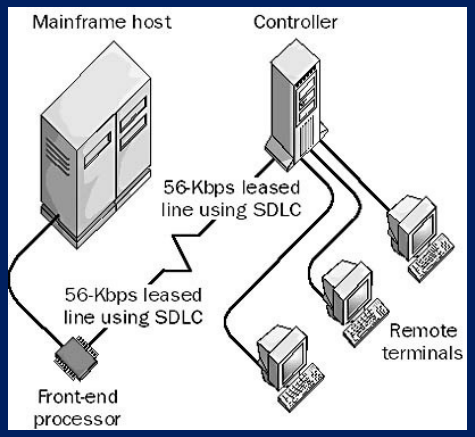


Complete Signal Flow

Dedicated Computer Links

- AT&T offered dedicated point-to-point digital circuits for use by mainframe computer systems
- Commonly called DDS for Digital Data Service
- These were all low-speed circuits of 56Kbps or less
- They were used primarily with IBM systems which used a method called SDLC to place remote terminals
- Other systems required a dedicated link *per terminal*

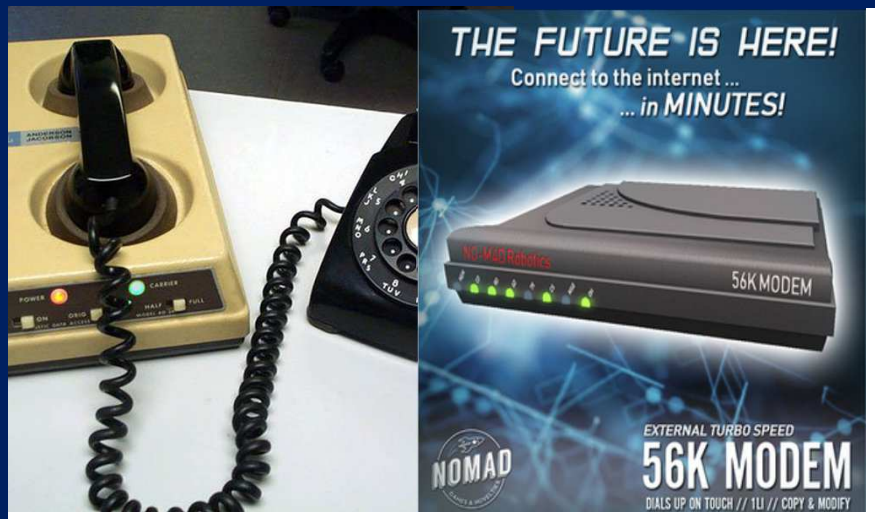
S/360 CPU and Peripherals

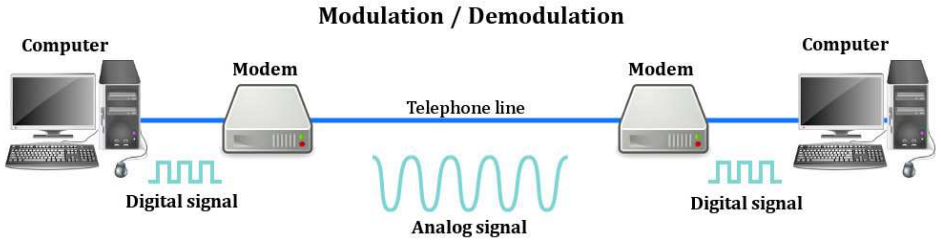


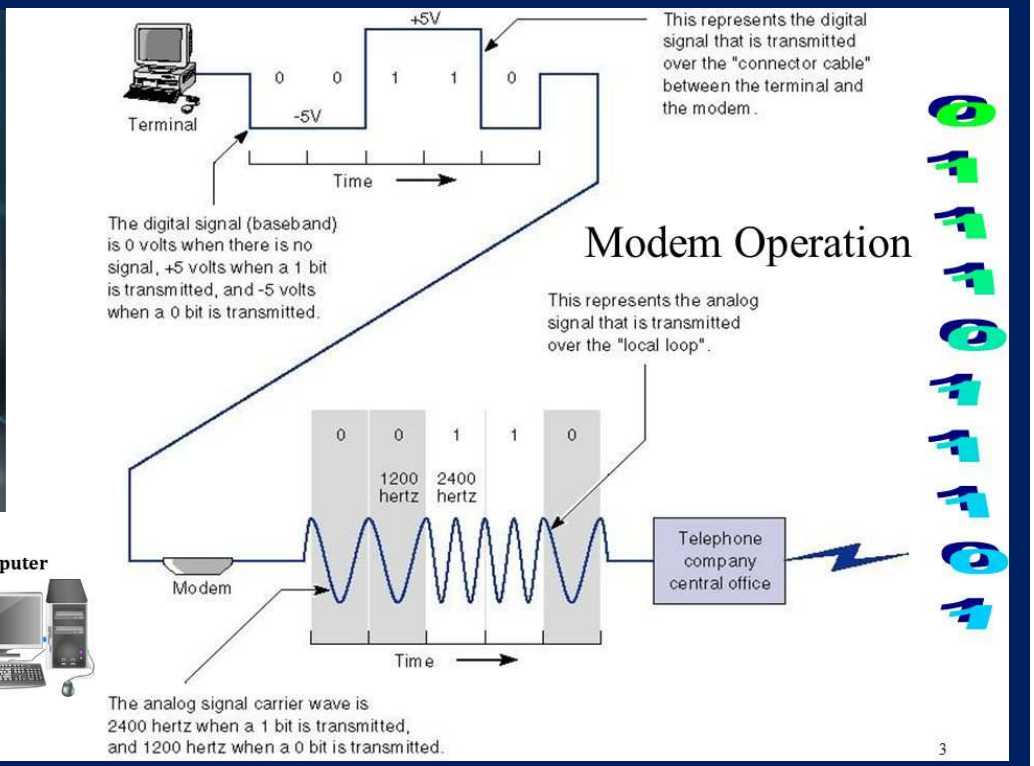
Remote terminals with SDLC

The First Modem

- A modem is a MODulator dEModulator, which changes digital signals into audio signals or vice-versa
- Permits two-way digital communication between two computers over ordinary phone lines
- The original modems were owned by AT&T and used the Bell 212A standard, which was 110bps and required the use of an “acoustic coupler” that held the handset
- With the AT&T breakup in the 1980’s, faster modems were developed up to 56Kbps, close to the theoretical Shannon limit for speed over a 4K bandwidth phone line
- Modem technology evolved to support Digital Subscriber Line connections (DSL) at speeds up to 50Mbps



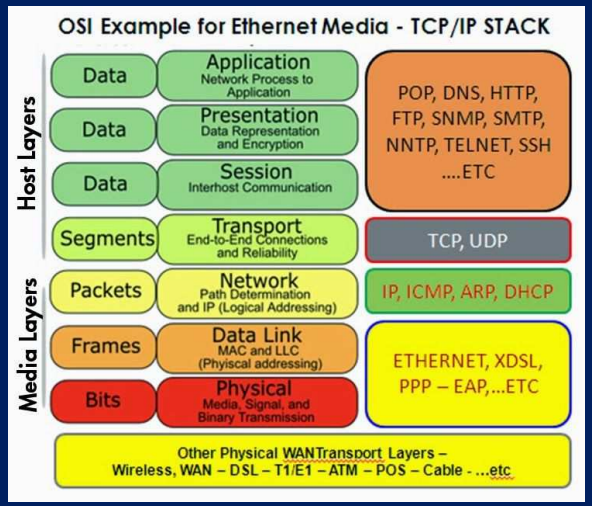


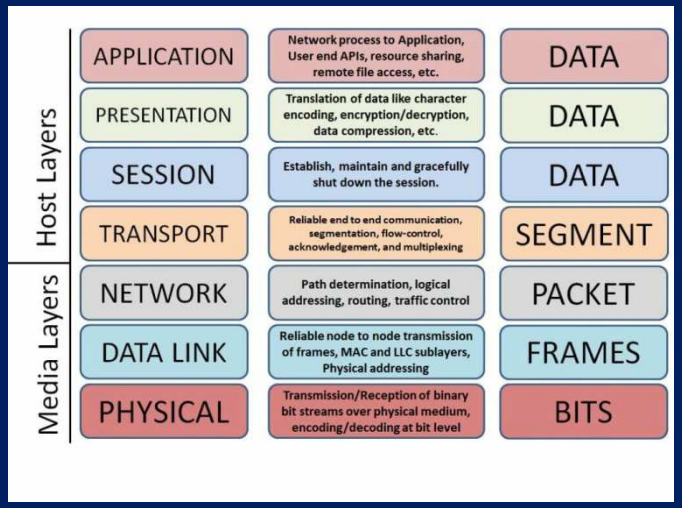


The OSI/ISO Layered Network Model

- In a Quora post asking about the purpose of the OSI model, Vikram Kumar answered this way:
- "The purpose of the OSI reference model is to guide vendors and developers so the digital communication products and software programs they create will interoperate, and to facilitate clear comparisons among communications tools."
- While some people may argue that the OSI model is obsolete (due to its conceptual nature) and less important than the four layers of the TCP/IP model, Kumar says that "it is difficult to read about networking technology today without seeing references to the OSI model and its layers, because the model's structure helps to frame discussions of protocols and contrast various technologies."
- If you can understand the OSI model and its layers, you can also then understand which protocols and devices can interoperate with each other when new technologies are developed and explained.

Source: Network World





Layer			Protocol data unit (PDU)	Function[21]
Host layers	7	Application	Data	High-level APIs, including resource sharing, remote file access
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

Formal End of Lesson 4 and 5

In next week's exciting episode

- Quick review: DC vs AC electric currents
- DC current operation
- Voltage vs Amperage (current)
- LEDs, constant current, and dropping resistors
- Connecting an LED to a microcontroller [Class Sketch#2]
- Ohm's Law
- Introduction to basic DMM functions and usage
- Series vs parallel circuits
- History of Telecommunications – Part 2
  - The OSI/ISO Networking Model Redux
  - Local Area Networks (LANs)
  - Wide Area Networks (WANs)
  - Wireless LAN (WiFi)
  - Cellular data networks
  - TCP/IP and the OSI model
  - The Rise and Fall of the Internet

Don't forget to return your USB drives!