SEICHE 2022
Basic Arduino Programming

Instructor: Paul Frommeyer

www.paulfrommeyer.com

Corporate Sponsor: DXC Technology

DXC
TECHNOLOGY

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)
- Review of microcontrollers and types of boards
- SEICHE LED display architecture
  - ESP8266 pinout
  - High level architecture

Lesson 2 – Laptop operation review – Windows and Linux

- Inventory of USB drives
- Installation of Arduino IDE software
- Installation of CH340/ESP8266 serial port drivers (Windows only)
- Control panel/settings location
- Home directories and folder hierarchy
- Arduino file locations
- Search functions
- (Windows) Device Manager
- (Linux) Konsole
- Copying flash drive contents [critical]
- Open questions and issues

Lesson 3 – IDE essentials

- Starting the Arduino IDE
- Basic Arduino sketch (program) structure
- Loading example sketches
- Loading and configuring new boards
- Connecting boards
- Identifying the microcontroller serial port
  - Linux
  - Windows
- Libraries
- Uploading a sketch
- The serial port monitor
- Printing to the serial port monitor
- A note on brace formatting

Lesson 4/5 – The Binary Number System (may take 2-3 lessons)

- Numerals vs numbers
- Review: the base 10 system and digit place values
- New: the base 2 system and digit place values
- Counting to 1000
- Bits and bytes and nybbles
- Binary addition and subtraction
- Formatting printed output
- Shifting and exponents

Lesson 6/7 – Integers and Arithmetic Operators (may take 2 lessons)

- **The integer data type**
- **Variables and the assignment operator**
- **Autoincrement and autodecrement**
- **Arithmetic Operators**
- **Assigning arithmetic results to a variable**
- **Bytes as data types**

Lesson 8 – Communicating with Sensors

- **Reading a potentiometer**
- **Reading a button**
- **Initializing I2C sensors**
- **Reading I2C sensors**

Lesson 9 – Characters and Arrays

- The hexadecimal number system
- Representing characters as numbers
- The ASCII character code
- The char data type
- Strings and character arrays

Lesson 10 – Comparison Operators

- ==, =>, <=
- strcmp
- Binary comparisons
- Uploading a sketch to the microcontroller

Lesson 11 – Control Structures

- if-then-else
- while
- for-next

Lesson 12 – Programming the LED matrix

- Initializing the SPI interface
- Controlling display brightness
- Lighting and clearing a single pixel
- Displaying text on the LED matrix
- Default fonts
- Nested for-next loops

Lesson 2 – Desktops and IDE

Part A

- Inventory of USB Drives
- Control panel/settings locations
- File manager/Explorer
- Copying USB Drive Contents

Part B

- Starting the Arduino IDE
- Adding a new type of board
- Finding and using example sketches
- Port selection and usage review
- Uploading a sketch to the ESP8266

Part C [Time Permitting]

- Introduction to computer data representation

USB Flash Drive Inventory

DOCUMENTATION – Arduino Reference

- Programming with Arduino.pdf
- Arduino Cookbook-2ndEdition.pdf
- Arduino-For-Beginners-REV2.pdf
- IntroArduinoBook-AlanSmith.pdf
- IntroductionToArduino-Book-AlanGSmith.pdf
- Make_Getting_Started_with_Arduino_3E.pdf

DOCUMENTATION – Electronics Reference

- the-original-guide-to-boards-2021.pdf
- Basic Electronics-Semiconductors.pdf
- Grobs Basic Electronics 2010.pdf
- Instructables-Basic-Electronics.pdf
- Intro to Electronics-Noisemantra.pdf
- Make Electronics 2nd Edition by Charles Platt.pdf
- SPIE-TT107-PracticalElectronicsforOpticalDesignandEngineering-Chapter1.pdf

DOCUMENTATION – ESP8266

- ESP8266 pinout diagram

DOCUMENTATION – Boards Guides

- Original Boards Guides from 2019-2022

Lessons

- SEICHE-BasicArduinoProgramming-Lesson1
  - SEICHE-BasicArduinoProgramming-Lesson1-V1.pdf [PDF document]
  - SEICHE-BasicArduinoProgramming-Lesson1-V1.ppt [PowerPoint document]

Critical Documents

- Make Getting Started With Arduino – Use for homework assignments
- Arduino Cookbook 2nd Edition – Use for homework assignments and critical reference source

RETURN FLASH DRIVES AT END OF EACH LESSON

- I'll load new content each week, that week's lesson and any sketches used in class

Laptop GUI Operation

- Desktop and task bar
- Start menu and control/settings panels
- Launching applications
- Context sensitive menus (right click)
- Filesystem browsers
- File and folder hierarchies
- Searching for applications
- Accessing and safely removing flash drives

Outside class scope: Application usage other than Arduino IDE, operating system customization

Laptop GUI Operation

Windows – Displaying Device Ports

- Device Manager
- Starting Device Manager in Windows 7
  Start Menu -> Enter “device” in Search Box -> Select “Device Manager”
- Starting Device Manager in Windows 10/11
  Windows Key+R -> Enter “devmgmt.msc” in program box

Command Line Interfaces

Windows - CMD

- Starting the DOS command line in Windows 7
  Start Menu - > Search Box
- Starting the DOS command line in Windows 10/11
  Windows Key+R -> Enter ‘cmd’ in program execute box

Linux - Konsole

- Starting Konsole from the start/application menu

Your instructor or class materials will provide what text to enter into the CLI interface for your particular OS

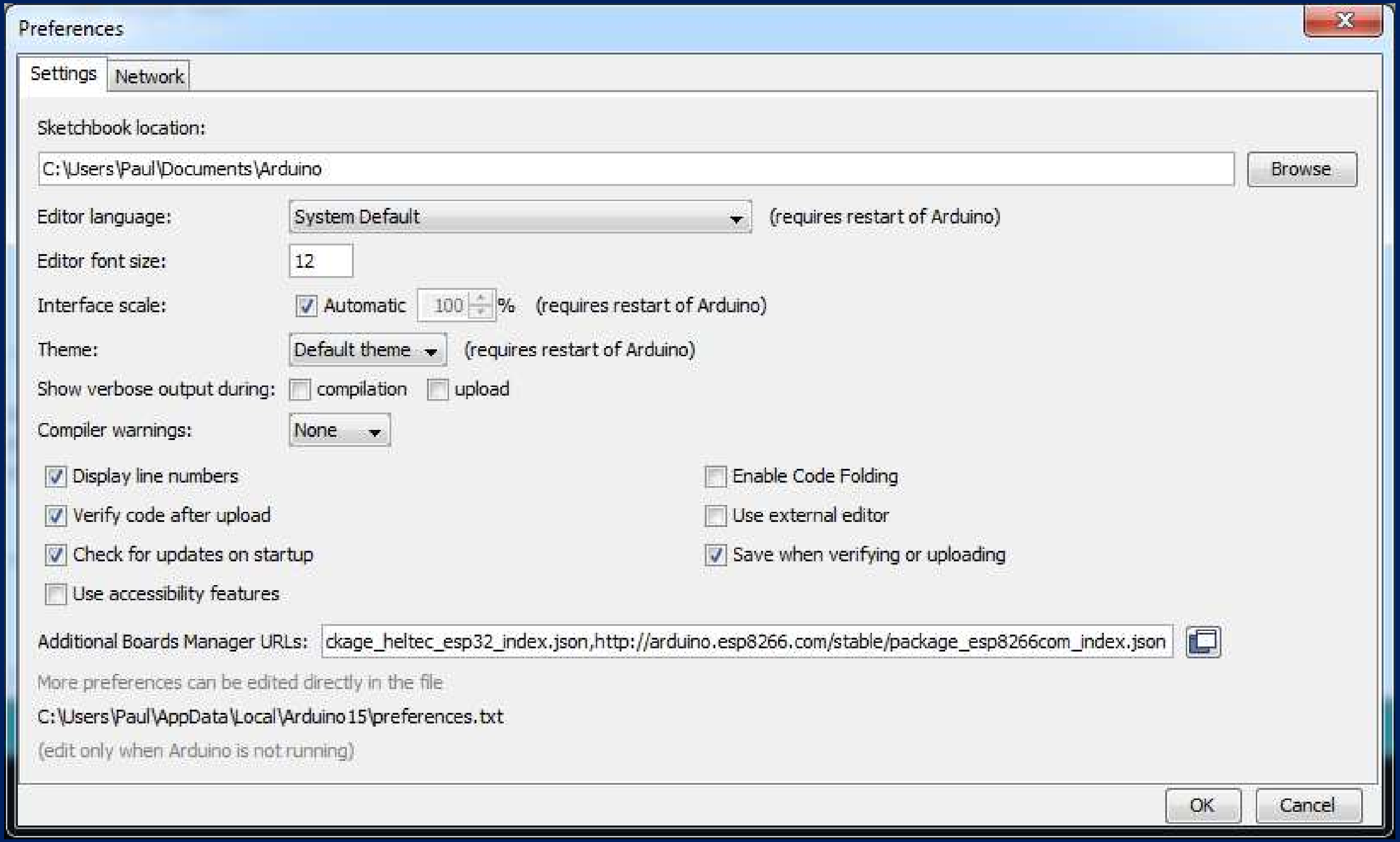
Command Line Interfaces

Linux – Displaying Serial Ports

- Start a Konsole instance
- Enter the following command “ls /dev/tty*”
- Follow all CLI commands with a carriage return/Enter key

This is how you determine what port your board is connected as. 99% of the time, it will be /dev/ttyUSB0, and the Arduino IDE will automagically find it.

Adding the ESP8266 to the IDE



Sketchbook location:

C:\Users\Paul\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

☑ Display line numbers

☑ Verify code after upload

☑ Check for updates on startup

□ Use accessibility features

□ Enable Code Folding

□ Use external editor

☑ Save when verifying or uploading

Additional Boards Manager URLs: ckage_heltec_esp32_index.json,http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

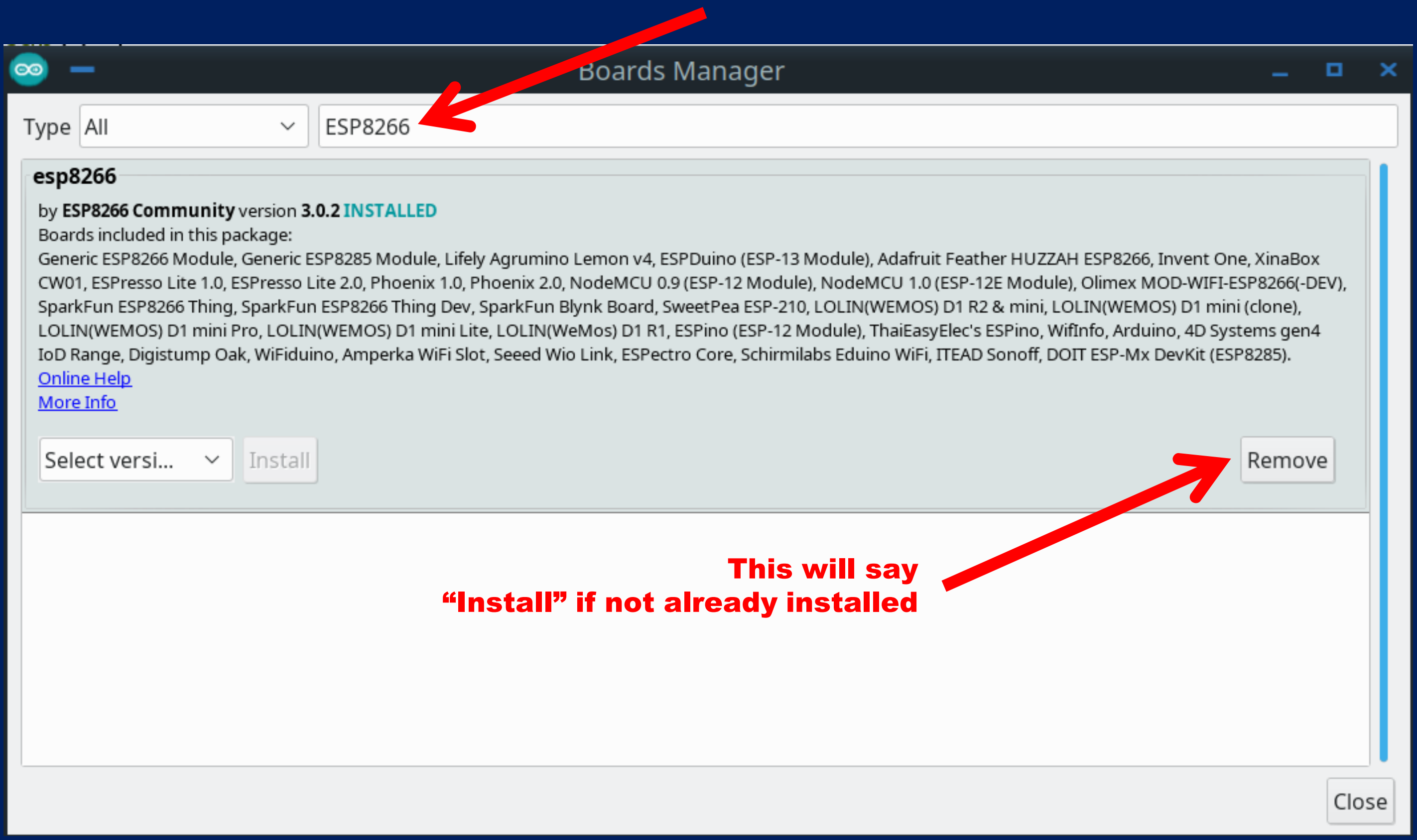
C:\Users\Paul\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

Open Preferences (File or Edit menu), and add the following URL to the Additional Boards Manager URLs field:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Adding the ESP8266 to the IDE



Type All

ESP8266

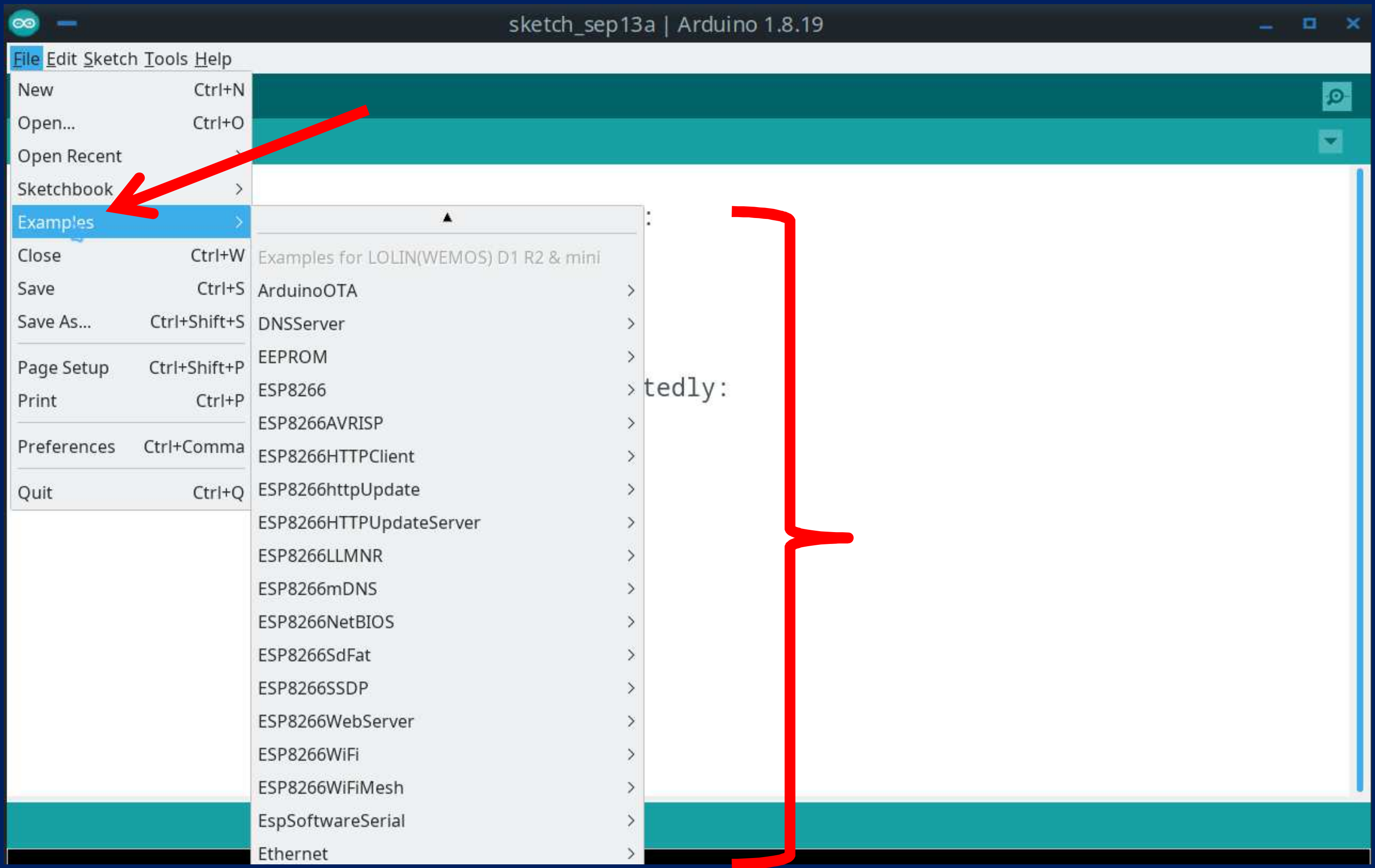
esp8266

by **ESP8266 Community** version **3.0.2** **INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, Lifely Agrumino Lemon v4, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SparkFun Blynk Board, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini (clone), LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, LOLIN(WeMos) D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core, Schirmilabs Eduino WiFi, ITEAD Sonoff, DOIT ESP-Mx DevKit (ESP8285).
Online Help
More Info

This will say
“Install” if not already installed

- Open the Boards Manager (Tools menu), enter “ESP8266” in the search box at the top
- Click on INSTALL (if it says “REMOVE”, the board is good to go)

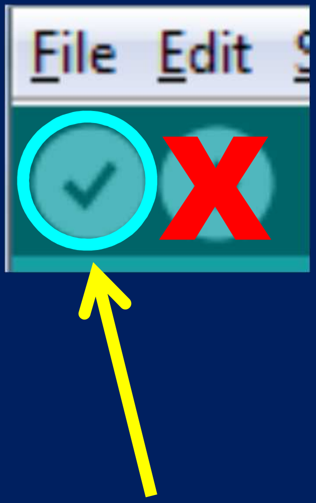
IDE: Using Examples – ESP8266



- Open the Examples menu
- Select the example you want
- For today, select “BLINK” from “Built In Examples”

Preparing a Sketch

- As mentioned, find the BLINK example sketch under Built In Examples”
- Click on the “VALIDATE” check mark icon
- DO NOT click on the “right arrow” (UPLOAD) icon at this time
- This will *compile* the sketch and check for errors prior to upload
- This is an optional step, since selecting upload will always compile the code first.
- Whenever using a new board (or library) for the first time, it’s a good idea to validate just to assure the software is happy before attempting and upload
- VALIDATE is extremely useful when troubleshooting code that won’t compile. Once it compiles without errors, it’s ready for upload.



This check mark icon is for VALIDATE

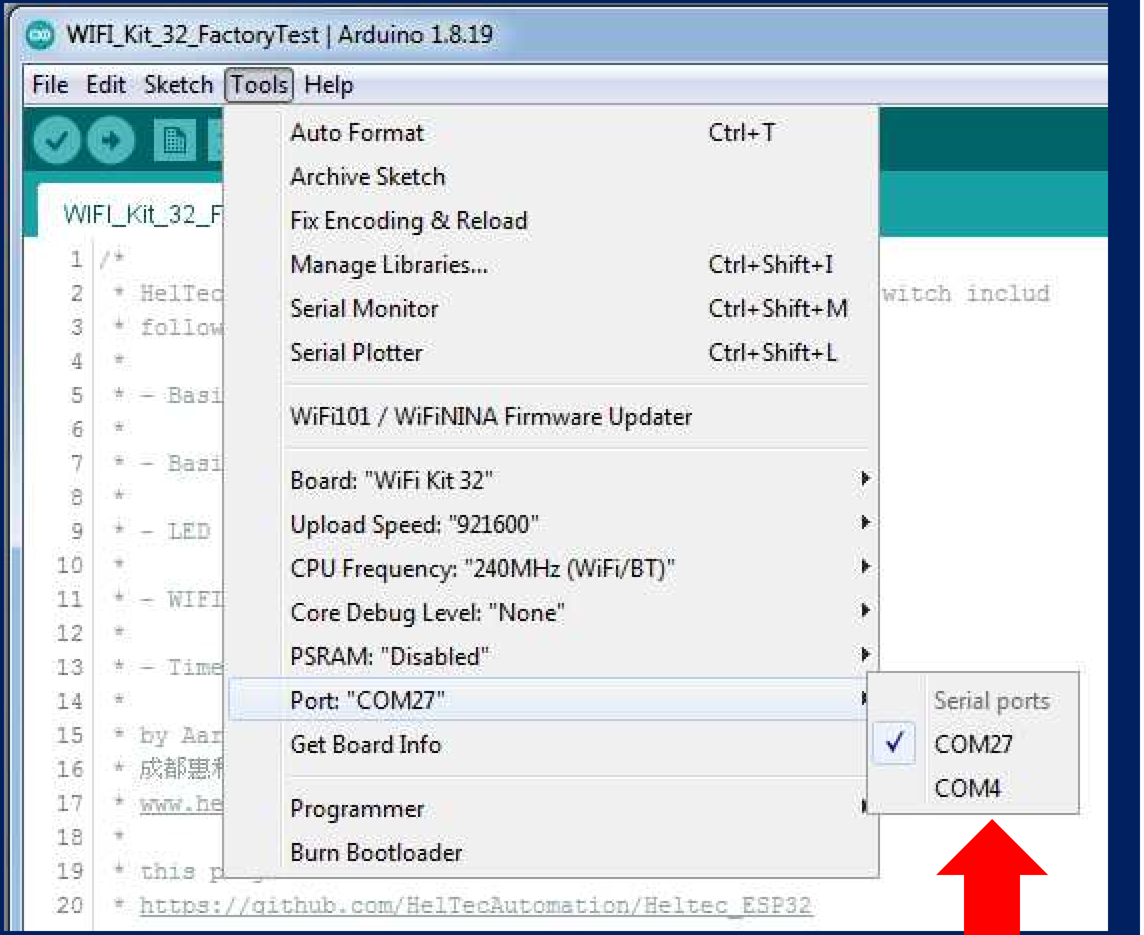
Finding and Setting the USB Port

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*`


WIFI_Kit_32_FactoryTest | Arduino 1.8.19
File Edit Sketch Tools Help
WIFI_Kit_32_F
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Programmer
Burn Bootloader
Serial ports
✓ COM27
COM4

Board: "WiFi Kit 32"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Core Debug Level: "None"
PSRAM: "Disabled"
Port: "COM27"

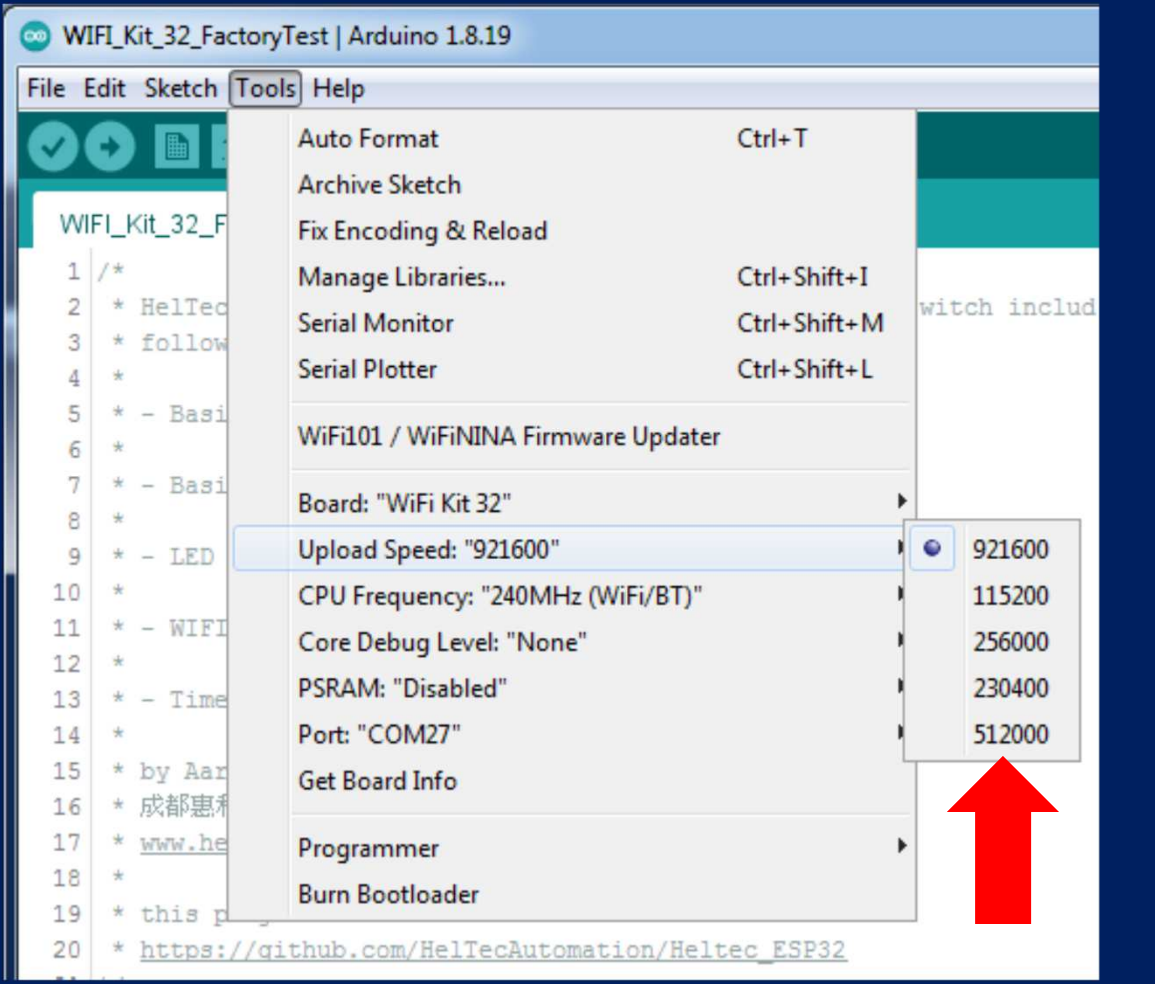
```
/*
 * HelTec                                              witch includ
 * follow
 *
 * - Basi
 *
 * - Basi
 *
 * - LED
 *
 * - WIFI
 *
 * - Time
 *
 * by Aar
 * 成都惠利
 * www.he
 *
 * this p
 * https://github.com/HelTecAutomation/Heltec_ESP32
```

Adjusting Port Speed

Always start by using 921600, this works with nearly all USB chipsets (there are exceptions for the ESP32, which uses the Sil2102 driver– but that's not us!)

If you get an upload failure

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


WIFI_Kit_32_FactoryTest | Arduino 1.8.19
File Edit Sketch Tools Help
WIFI_Kit_32_F
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Programmer
Burn Bootloader
Get Board Info
921600
115200
256000
230400
512000

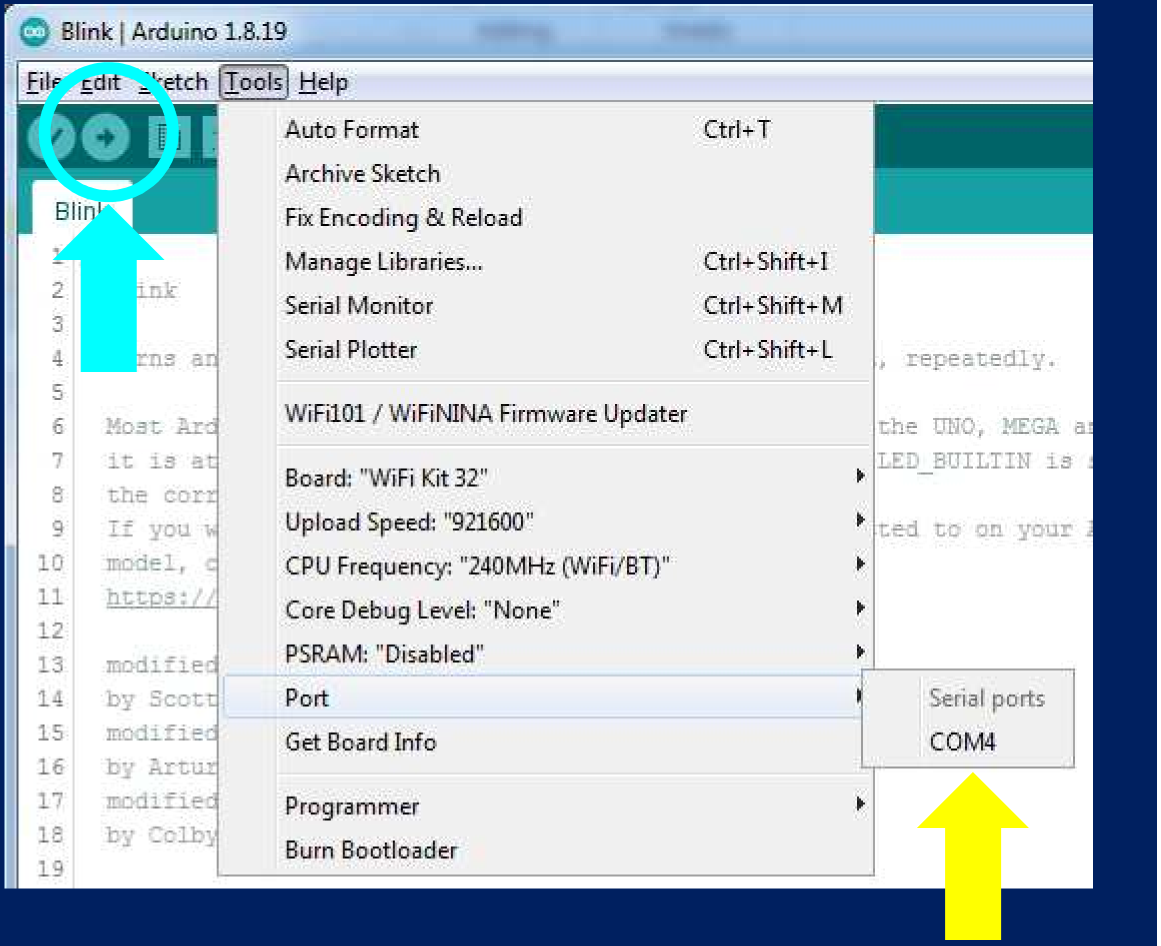
Board: "WiFi Kit 32"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Core Debug Level: "None"
PSRAM: "Disabled"
Port: "COM27"

● 921600

```
/*
 * HelTec
 * follow                                    witch includ
 *
 * - Basi
 *
 * - Basi
 *
 * - LED
 *
 * - WIFI
 *
 * - Time
 *
 * by Aar
 * 成都惠利
 * www.he
 *
 * this p
 * https://github.com/HelTecAutomation/Heltec_ESP32
```

Uploading a Sketch

- **Ensure you have selected the correct port for your currently plugged in board**
- NOW click on the “UPLOAD” arrow icon
- A light or lights should flash on your board
- Once upload is finished, the sketch will immediately run
- You should see the onboard LED on your board start blinking
- Note: On the ESP8266, the LED is next to the silver CPU module



Board: "WiFi Kit 32"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Core Debug Level: "None"
PSRAM: "Disabled"

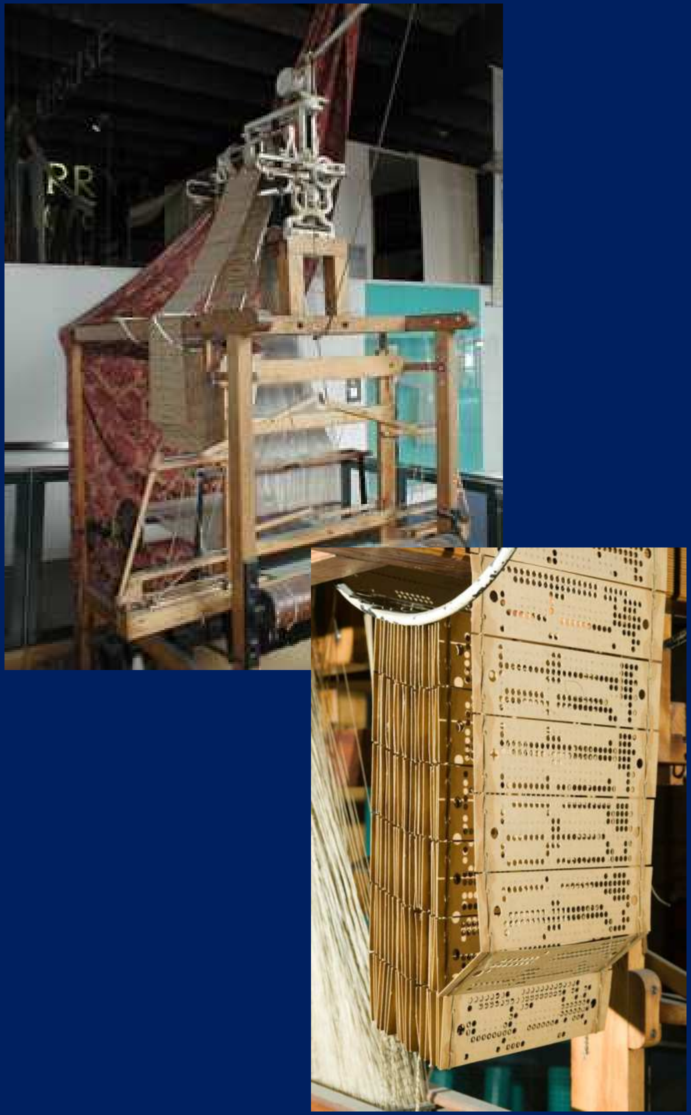
Introduction to Computer Data Representation

Chalk Talk – No formal class notes, sorry!

Lots of details on this next week, when we start learning about numbers and variables in C++.

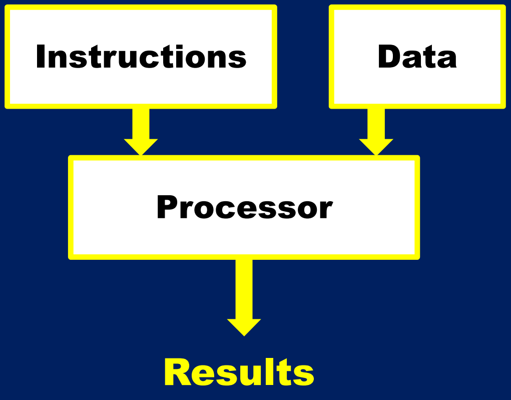
Jacquard Loom

- Invented by Joseph Marie Jacquard in 1804
- Technically, a Jacquard *Machine* which was attached to the heddles of an existing pattern loom
- Provided automatic control of the heddles managing the warp threads
- Used large punched cards to define the woven pattern
- A hole in the card would cause that thread of the warp to be raised, thus excluded from the weft during a shuttle pass
- Allowed extremely complex patterns to be woven
- Once machine driven, gave rise to modern weaving industry
- Mechanism still in use today, but electromechanical relay driven and controlled by digital computer

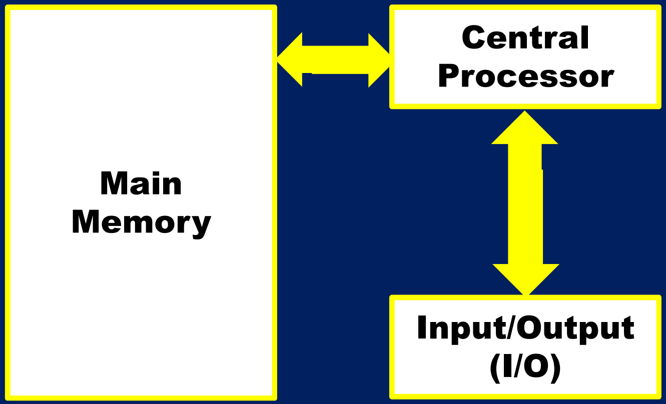


Von Neumann Architecture

- Earliest computers had fixed (hardwired) programs
- Control storage (program) was originally separate from data storage (information to be operated upon)
- John Von Neumann invented the idea of using a single storage medium for both instructions and data
- This was a revolutionary design, reducing costs and increasing flexibility, and is used by all computers since



Historic Architecture



Von Neumann Architecture

Formal End of Lesson 2

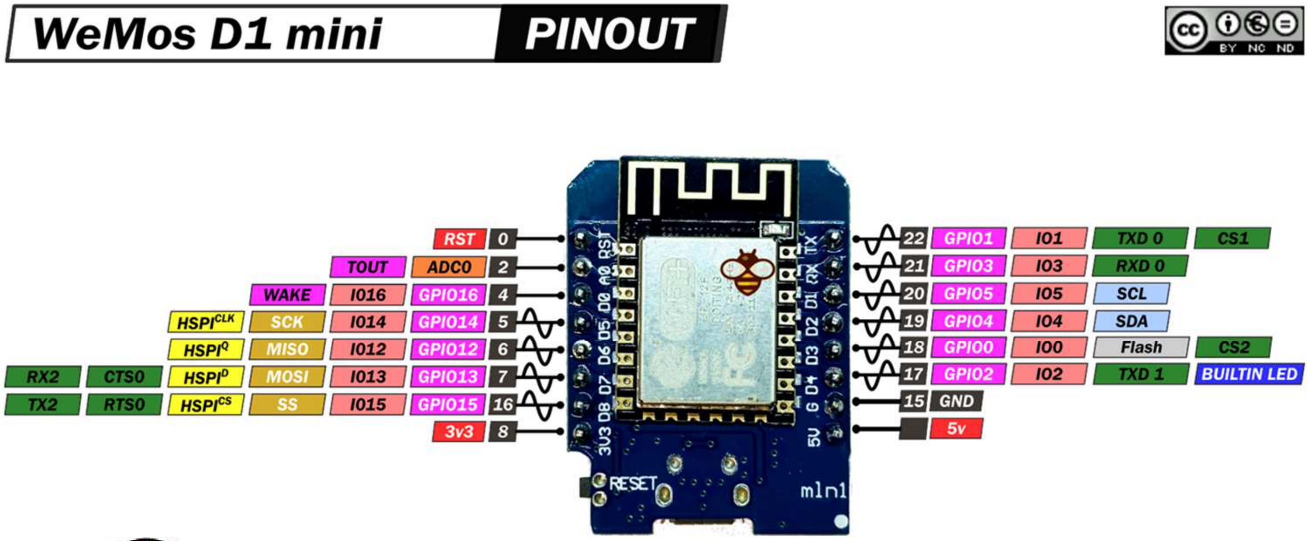
HOMEWORK!

- **Getting Started With Arduino – Chapter 3**
- **Arduino Cookbook – Chapter 1**

In next week’s exciting episode

- Statements, values, and variables in C++
- The serial port monitor
- Computer data representation, Part 1

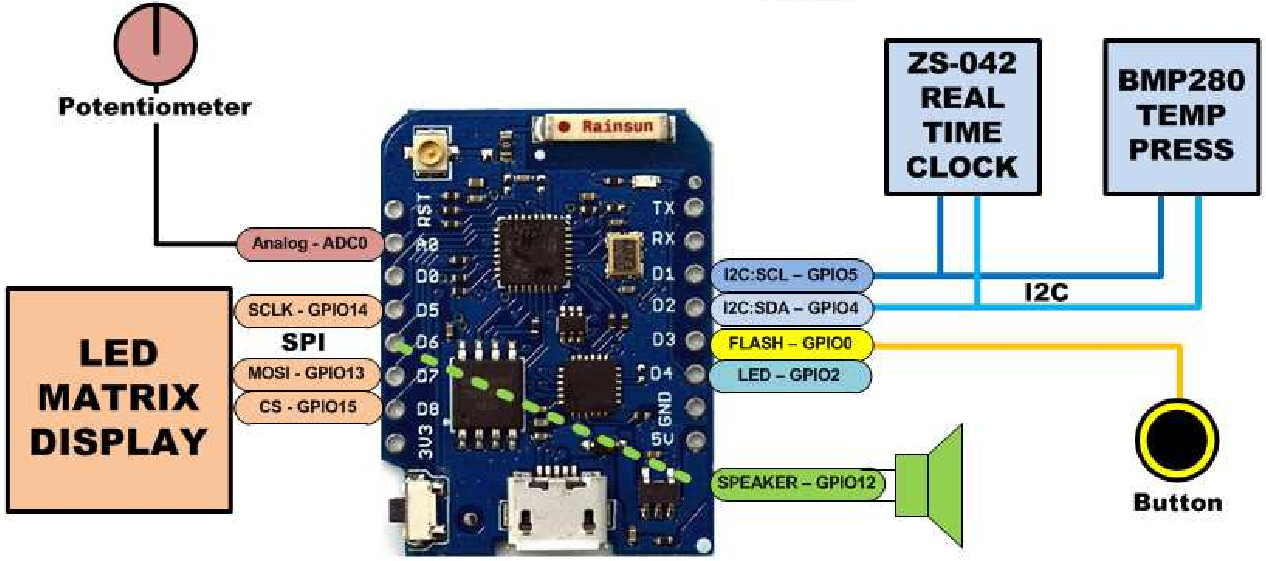
LESSON REFERENCE



WeMos D1 mini PINOUT

						RST	0
					TOUT	ADC0	2
				WAKE	IO16	GPIO16	4
		HSPI^{CLK}	SCK		IO14	GPIO14	5
		HSPI^{Q}	MISO		IO12	GPIO12	6
RX2	CTS0	HSPI^{D}	MOSI		IO13	GPIO13	7
TX2	RTS0	HSPI^{CS}	SS		IO15	GPIO15	16
						3v3	8

22	GPIO1	IO1	TXD 0	CS1
21	GPIO3	IO3	RXD 0	
20	GPIO5	IO5	SCL	
19	GPIO4	IO4	SDA	
18	GPIO0	IO0	Flash	CS2
17	GPIO2	IO2	TXD 1	BUILTIN LED
15	GND			
	5v			



SEICHE LED DISPLAY ARCHITECTURE