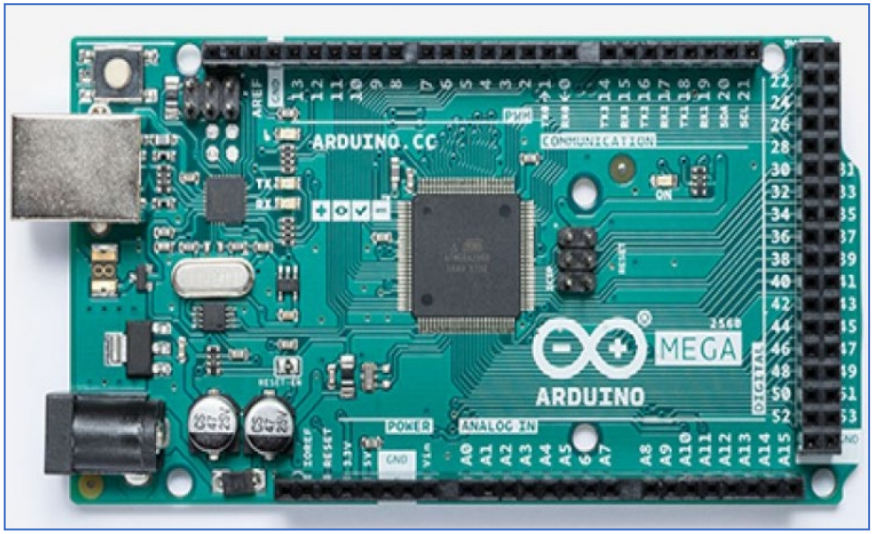
Introduction to the Arduino Hardware and IDE

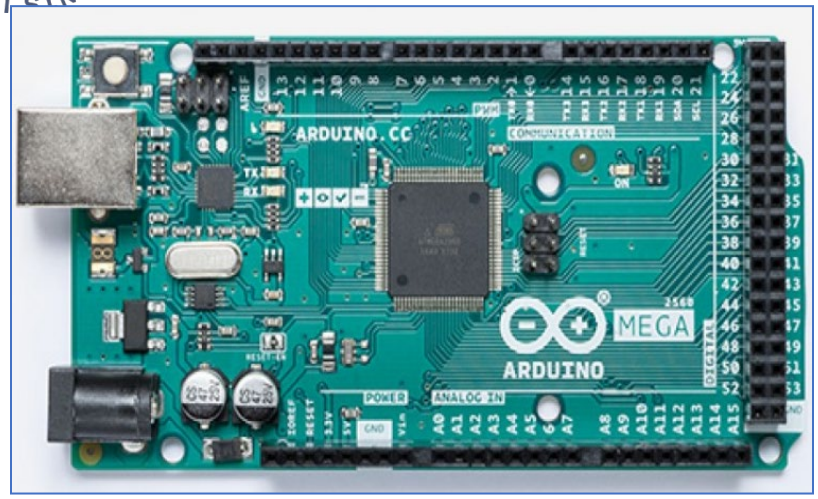
What is an Arduino?

- Arduino is an Italian company that produces a number of microcontroller boards and a common programming interface for those microcontrollers
  - The Microcontrollers on the board are produced by other companies like Atmel, Microchip, Broadcom
  - Boards allow prototyping and easy access to I/O pins
  - A few standard footprints allow 3rd party boards called Shield to be used
  - Arduino IDE allows common code to run on multiple microcontrollers

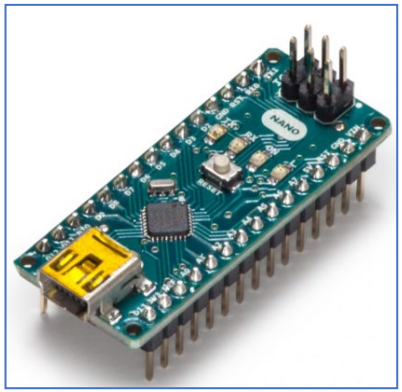


Arduino Mega Development Board which we will use

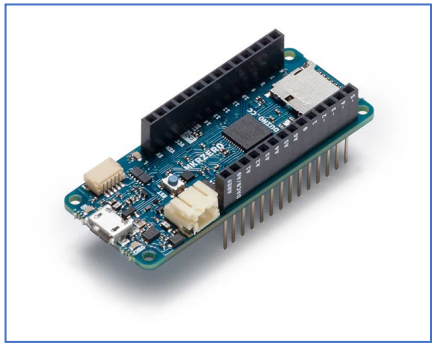
Different Arduino Footprints (Called “Families” as of 2025)



Mega2560, Arduino Mega Family, large number of interfaces (both digital and analog), this is the microcontroller for LaACES



Arduino Nano microcontroller, Nano family, small size

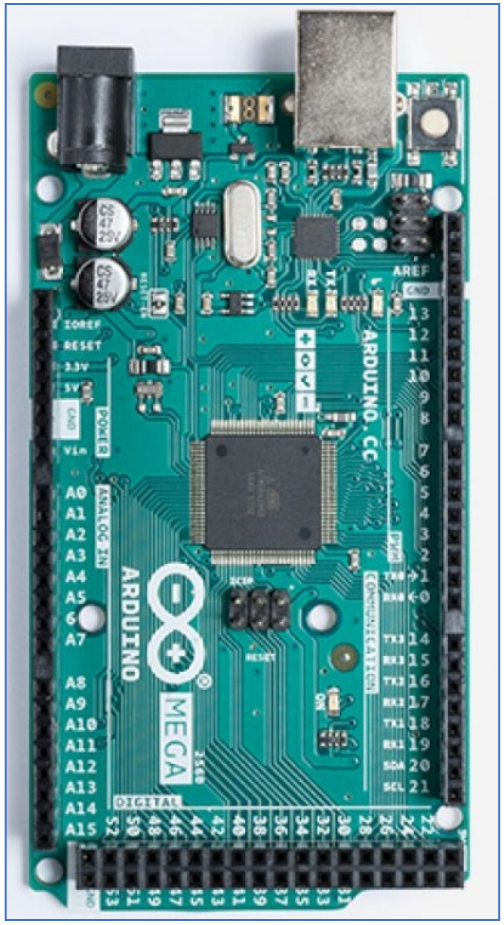


MKR Zero, MKR Family usually has radio-based networking

Arduino Uno, Classic Family, classic shields are compatible with Mega Family

Arduino Key Mega Specifications

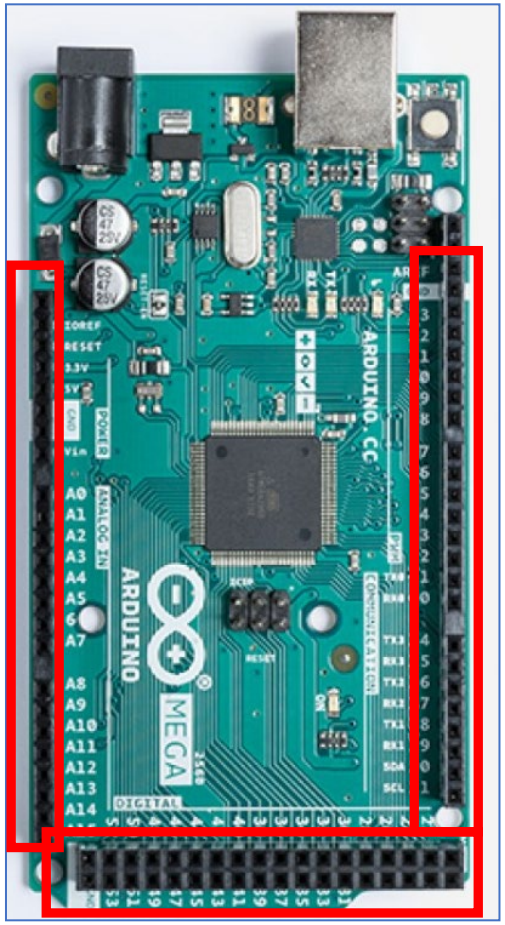
Specification	Value
Supply Voltage	7-12 V
Operating Voltage	5 V
Programming Interface	USB-B (the square one)
Program Storage	256 KiloByte Flash *(Where the uploaded program is saved)*
System Memory	8 KiloByte SRAM *(Where program variables are stored when running)*
Digital IO Pins	54 Pins *(Bottom and Right Headers)*
Digital Interfaces	1xSPI, 1 x I^2C, 4 x Serial UART *(in the 54 digital pins)*
Analog Inputs (ADC)	16 Channel, 0-5V, 10 Bit ADC *(Left Headers)*



Arduino Mega microcontroller

Major Board Components

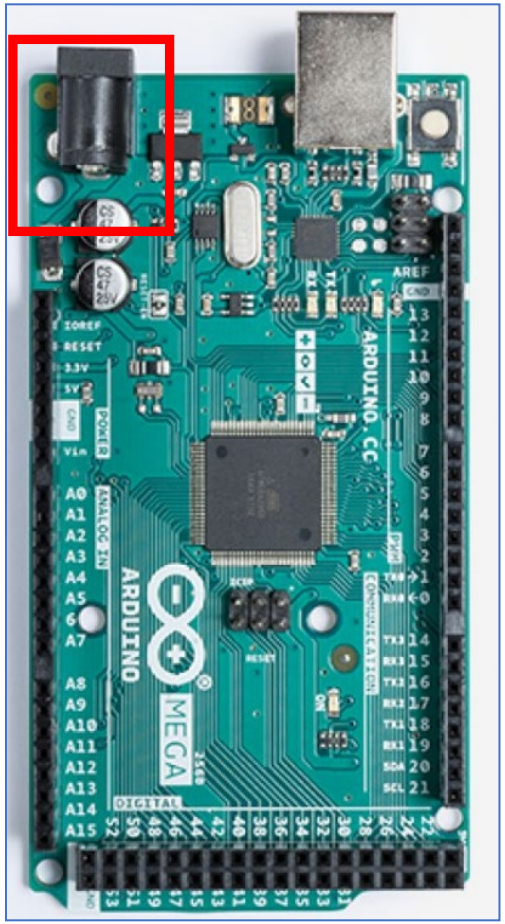
- IO Pin Headers
  - Provide a physical interface to the IO pins and shield stacking
  - We will cover these interfaces in the coming weeks
- 12V DC Power Connection
- USB Connection
- Main Microcontroller
- Reset Button
- USB Interface Microcontroller



Arduino Mega2560 microcontroller board

Major Board Components

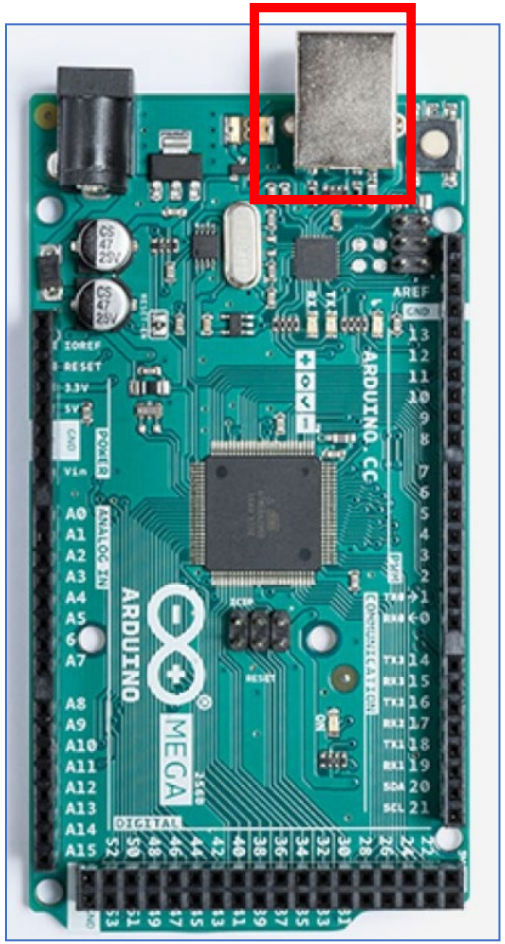
- IO Pin Headers
- 12V DC Power Connection
  - One method power connection for operation without a computer
- USB Connection
- Main Microcontroller
- Reset Button
- USB Interface Microcontroller



Arduino Mega2560 microcontroller board

Major Board Components

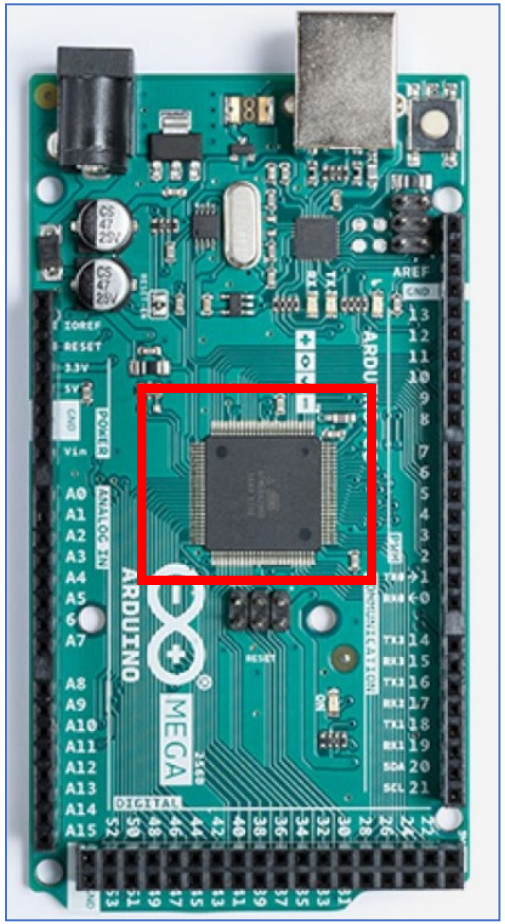
- IO Pin Headers
- 12V DC Power Connection
- USB Connection
  - Provide a data connection to a computer for program uploads and serial communication
  - Can power board at 5V
- Main Microcontroller
- Reset Button
- USB Interface Microcontroller



Arduino Mega2560 microcontroller board

Major Board Components

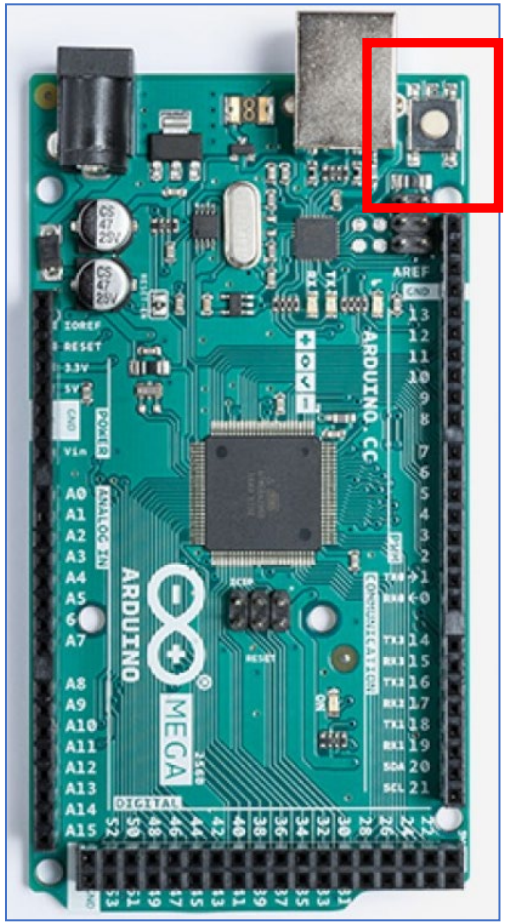
- IO Pin Headers
- 12V DC Power Connection
- USB Connection
- Main Microcontroller
  - AVR Atmega2560
  - Microcontroller that runs our program
- Reset Button
- USB Interface Microcontroller



Arduino Mega2560 microcontroller board

Major Board Components

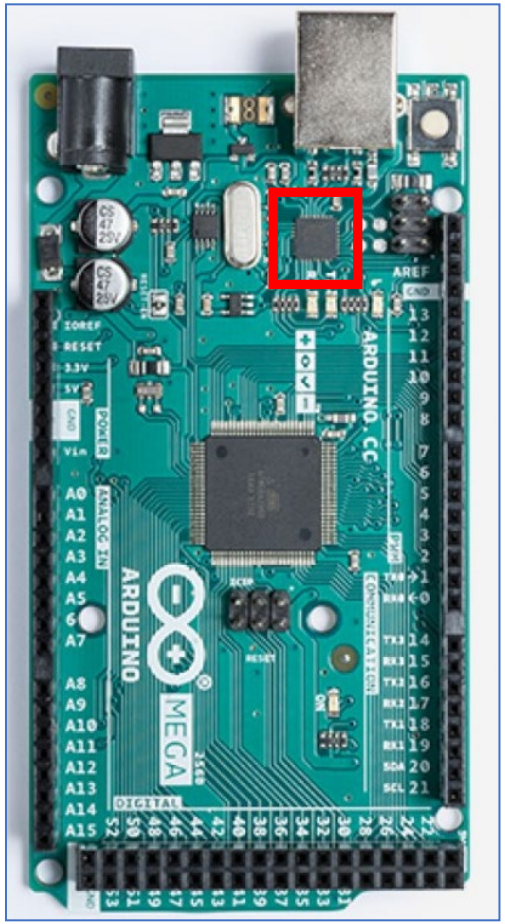
- IO Pin Headers
- 12V DC Power Connection
- USB Connection
- Main Microcontroller
- Reset Button
  - Pressing button restarts the program loaded on the microcontroller
- USB Interface Microcontroller



Arduino Mega2560 microcontroller board

Major Board Components

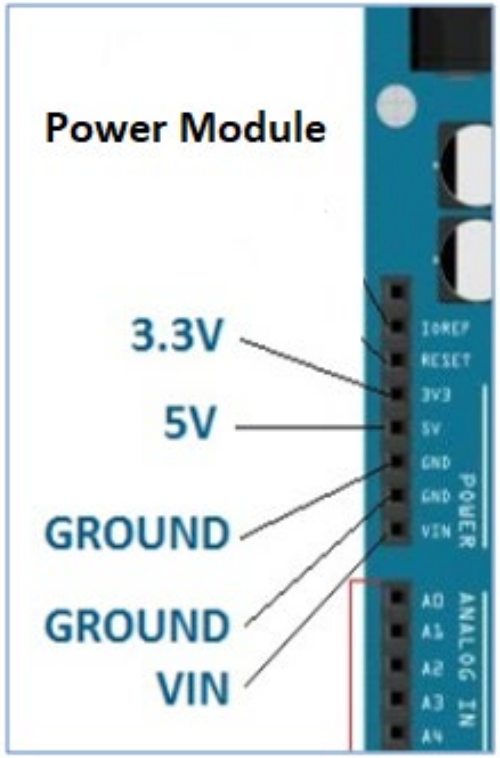
- IO Pin Headers
- 12V DC Power Connection
- USB Connection
- Main Microcontroller
- Reset Button
- USB Interface Microcontroller
  - A second microcontroller (Atmega328)
  - Acts as a translator between the main microcontroller and the USB



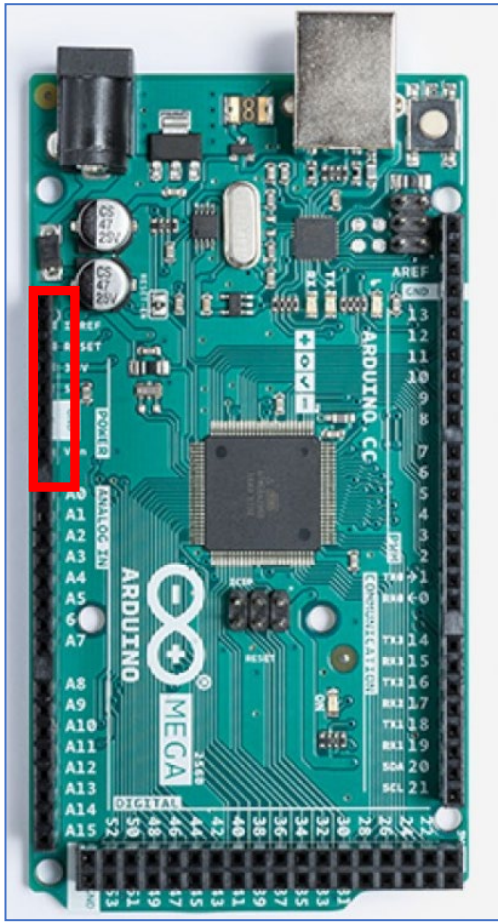
Arduino Mega2560 microcontroller board

Power Header

- Top left header provides common power connections
- **VIN**: Input power from an external supply (7-12V like DC connector)
- **5V**: Output 5V to shields and other external components
- **3V3**: Output 5V to shields and other external components
- **IOREF:** Provide reference voltage for a Digital High or 1, 5V on the Mega
- GND: All grounds (including the USB and Power Connector) are connected together for a common 0V

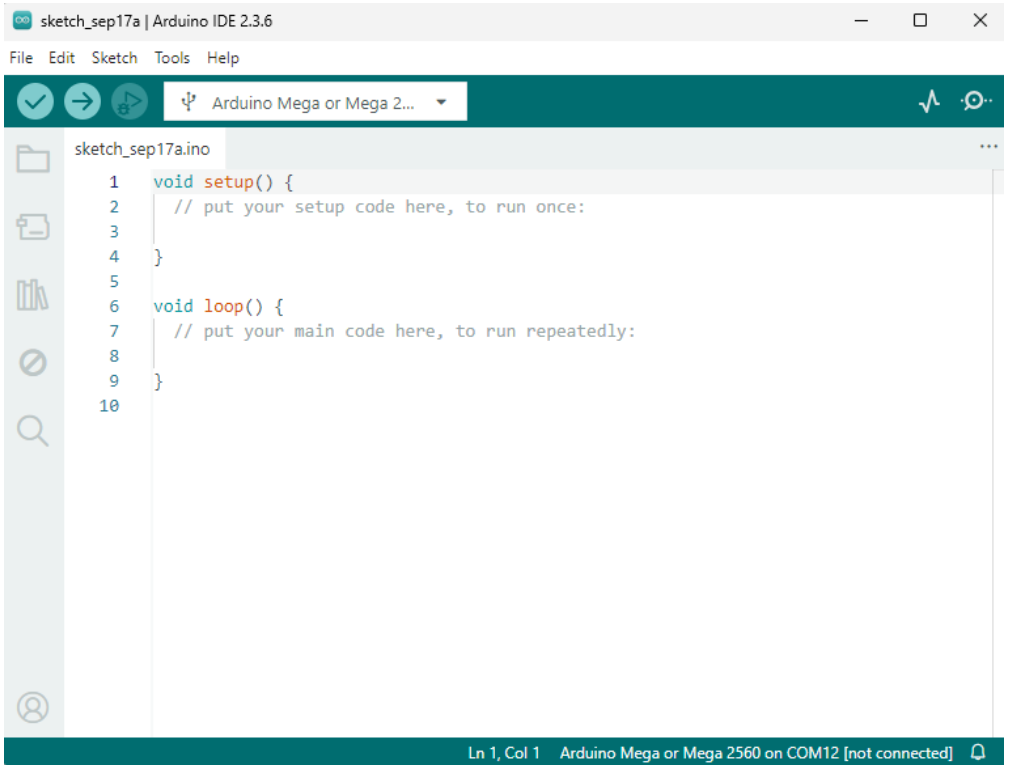


Arduino Mega Power Header



Arduino IDE

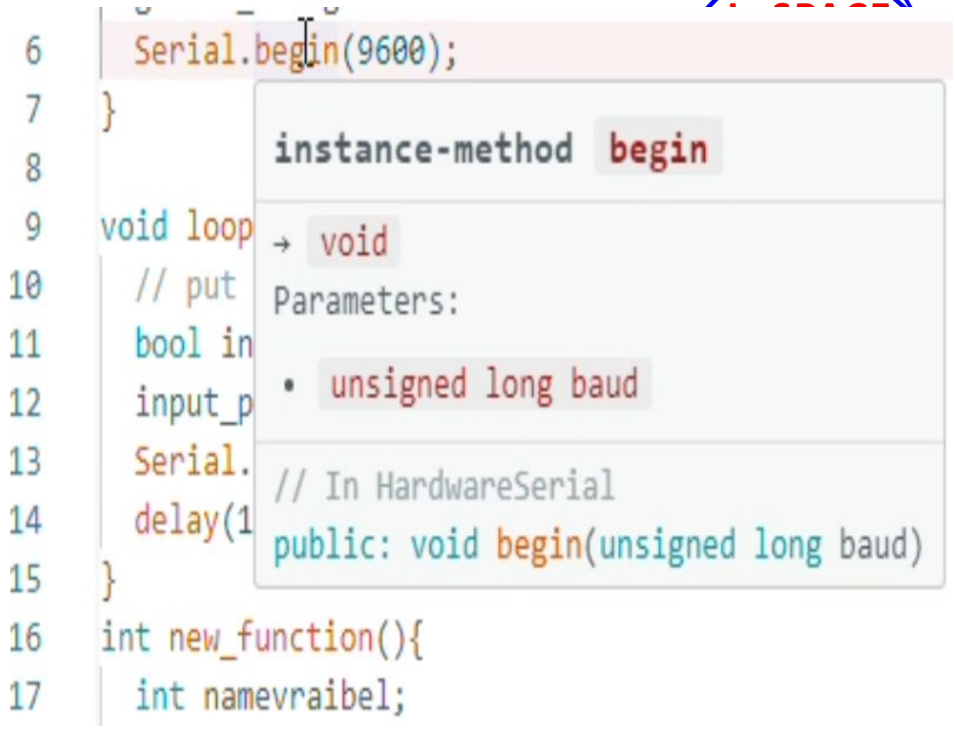
- An IDE is an integrated development environment (IDE)
- Combines several tools for writing code into a single program
  - Source Code Editor
  - Library Manager
  - Compiler
  - Debugger and troubleshooting tools



The Arduino IDE screen as seen when first opening a blank sketch

Source Code Editor

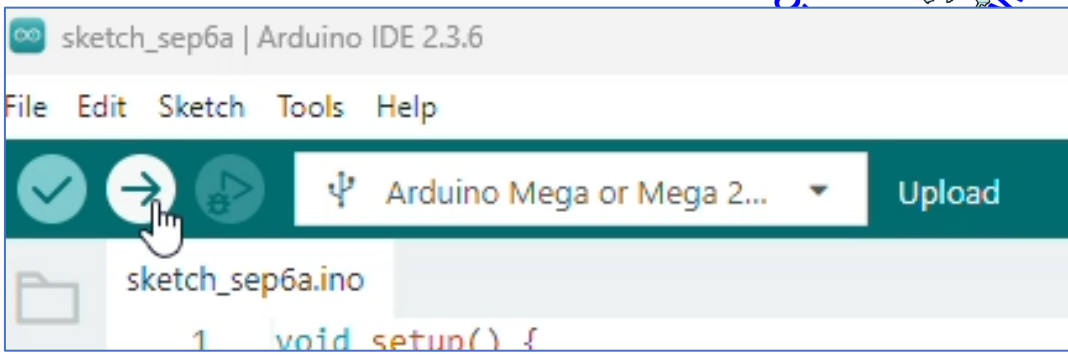
- The main portion of the IDE window is the text editor used to write the code
- The numbers on the left are just for reference (can be turned on and off)
- Quality of Life Features:
  - Text color-coded to differentiate functions, types, comments
  - Closing parentheses and braces are automatically added
  - Automatic indentation on new lines inside braces
  - You can mouseover or right-click variables and functions to look at their definition (very useful for library functions)



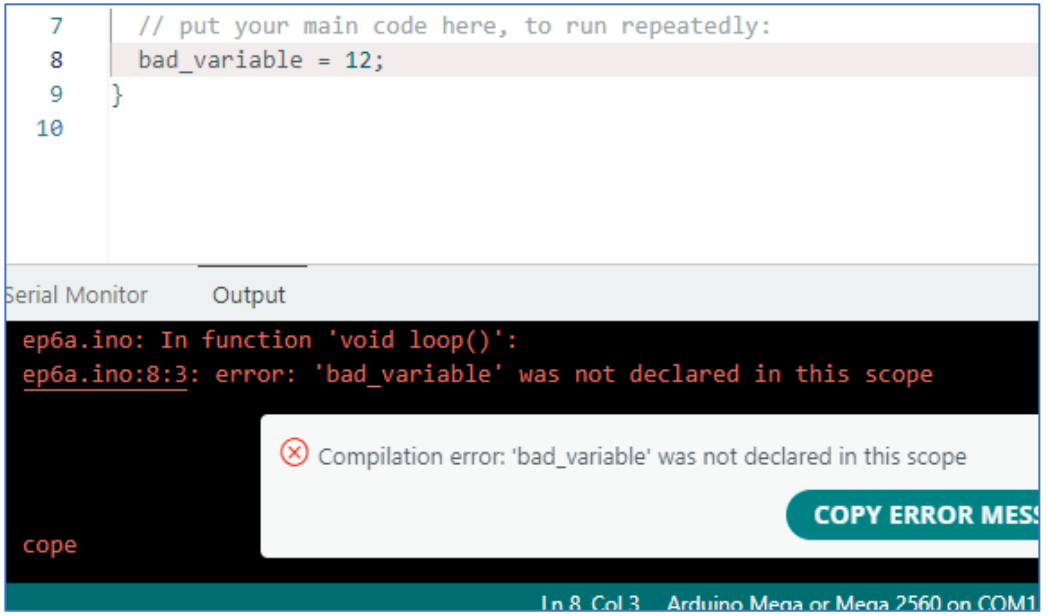
The abbreviated definition for the begin() function, you could also jump to the definition (inside a library in this case) by right-clicking

Compiler

- Before code can run, it must be translated from the human-readable text into machine code (compiled)
- Then the code must be uploaded to the microcontroller
  - The verify button tries to compile the code to check for errors
  - The verify compiles and then uploads the code to the connected microcontroller
  - If the compiler encounters an error, a message will show up in the bottom of the window, with a line number and function reference
  - The bottom window can be made larger or scrolled up if there is more text
  - The compiler will stop at the first error encountered
- Be aware the IDE autosaves (possibly overwriting an old version) when you verify or upload (this can be turned off)



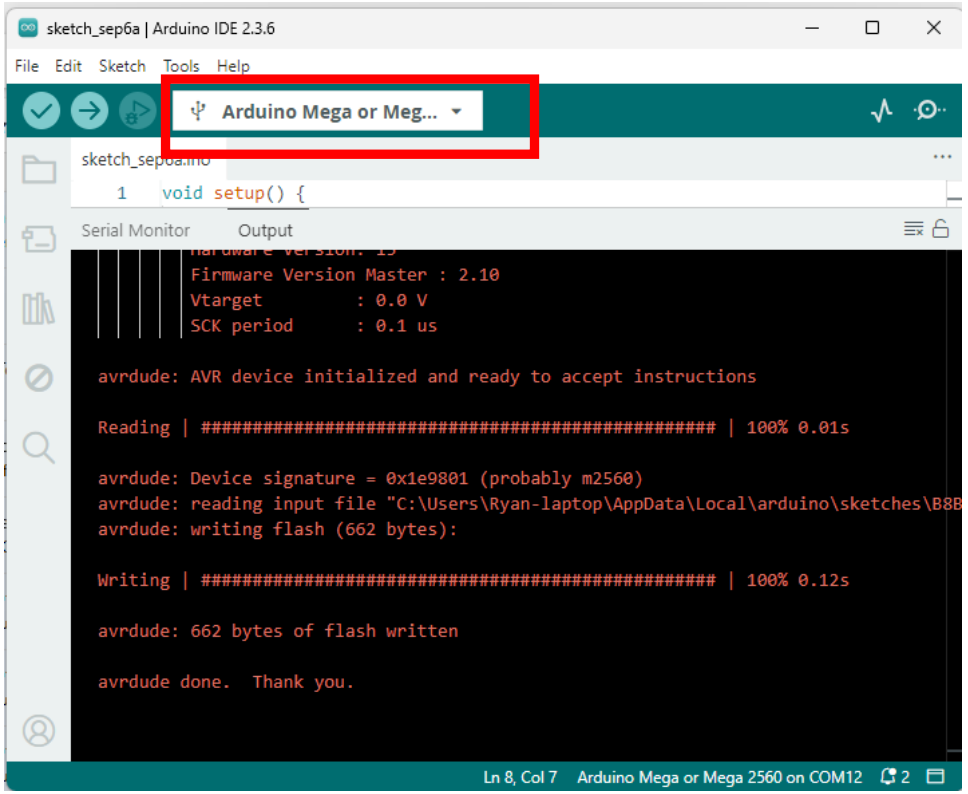
The verify (check) and upload (arrow) buttons are found at the top the IDE window



Sample compiler error, here we can see the function void(), on line 8

Uploading the code

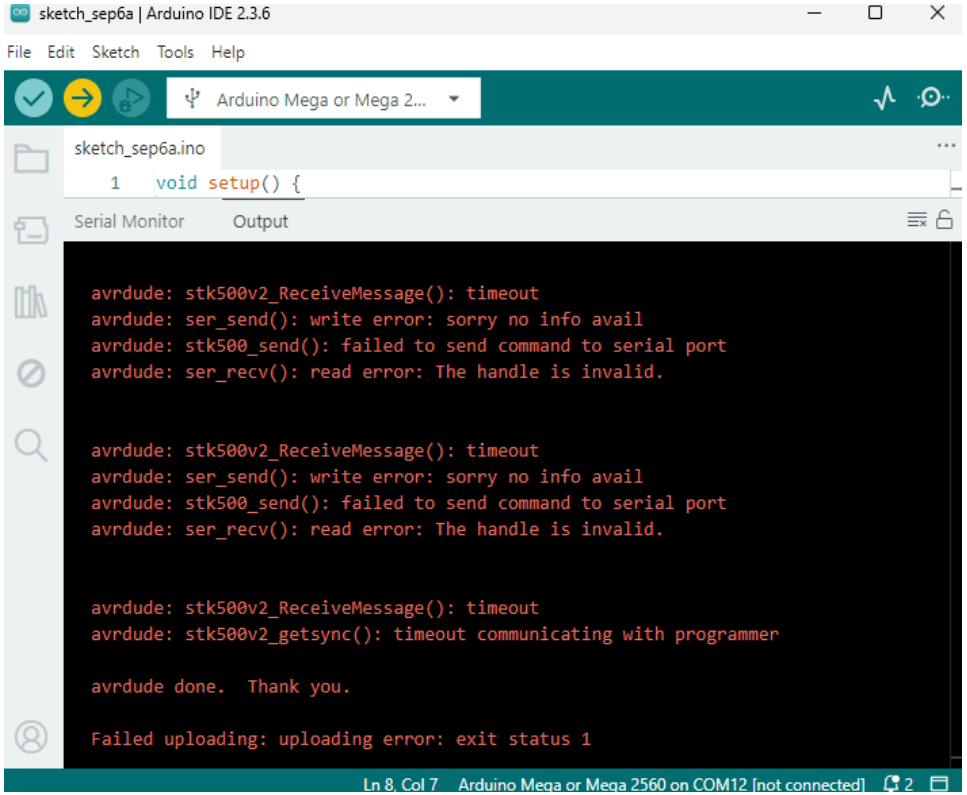
- To upload the code, the correct board and port must be selected in the dropdown menu to the right of the upload button
  - The Arduino must be connected and powered
  - Usually but not always the correct target be automatically selected when the USB is plugged in
  - If not listed, you may need to pick the board and port under the tools menu
- A successful upload will give you a success message, with the size of your program written to the flash memory
- A failed upload will give an error message (usually a timeout)



Successful upload, the # symbols will actually fill in acting as progress bars

Uploading the code

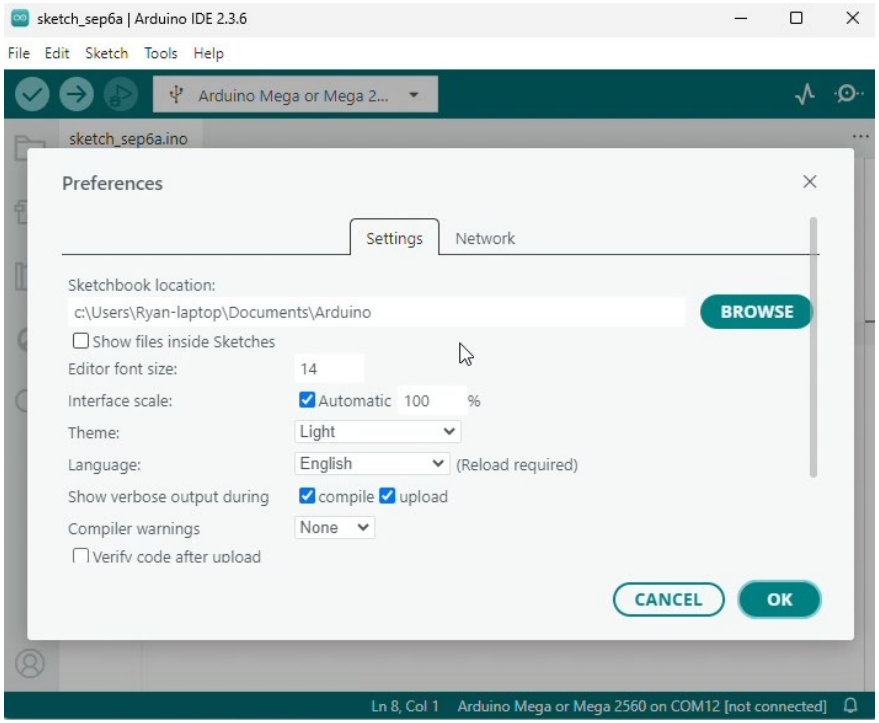
- To upload the code, the correct board and port must be selected in the dropdown menu to the right of the upload button
  - The Arduino must be connected and powered
  - Usually but not always the correct target be automatically selected when the USB is plugged in
  - If not listed, you may need to pick the board and port under the tools menu
- A successful upload will give you a success message, with the size of your program written to the flash memory
- A failed upload will give an error message (usually a timeout)



Failed upload message (no Arduino was connected), notice how the dropdown menu still says Arduino Mega

File Menu

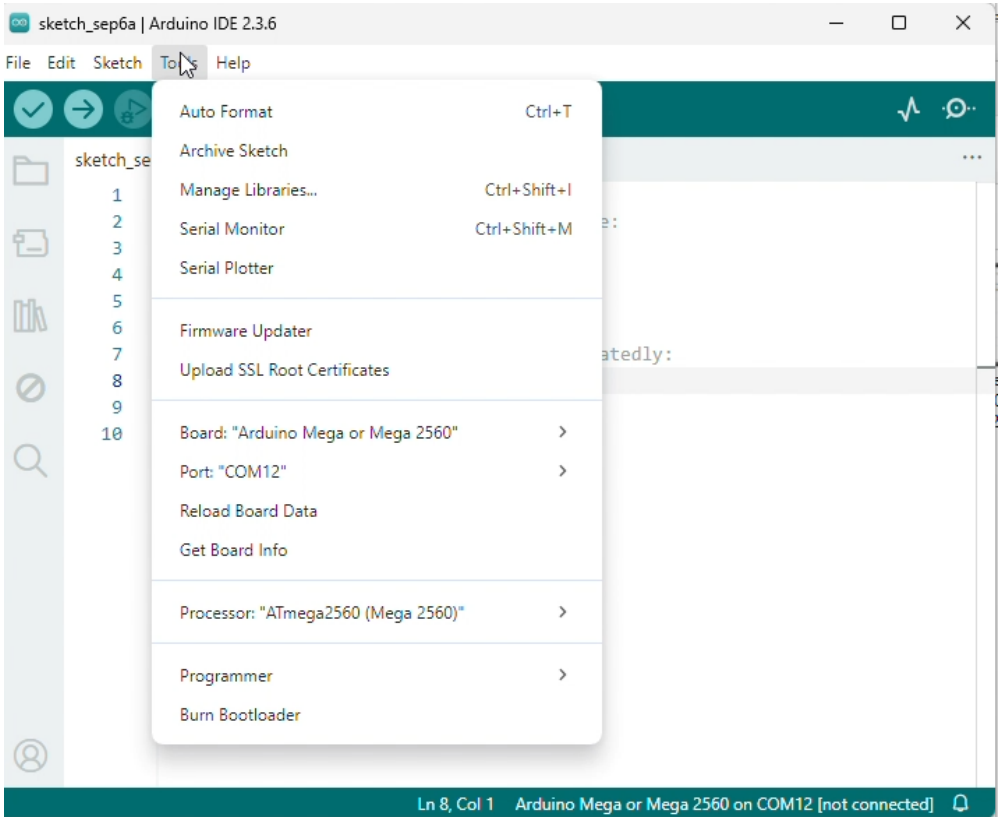
- The file folder expected options to save, open, close, etc.
- The preferences window includes a number of important options
  - The sketchbook location is the default location for your sketches (program files) and **where library files** will be installed
  - Hitting browse will open that folder
  - Editor Quick Suggestions will allow the IDE to suggest functions when you are writing code
  - Autosave controls the autosave on Verify/Upload



Preferences window, opened via the file menu

Tools Menu

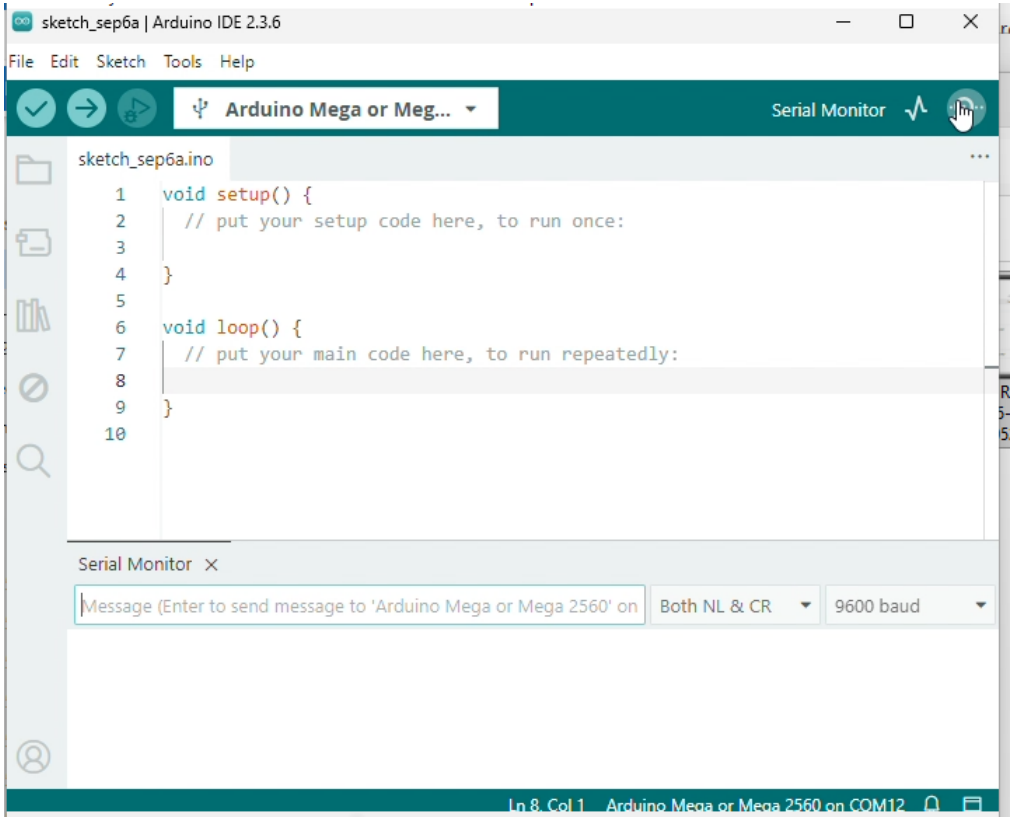
- This menu allows you to select the target board and port
  - They should autoselect when the USB is plugged in, but this sometimes fails
  - The Mega is listed under Arduino AVR Boards
  - In Windows com ports will be listed as COM#
  - In Mac/Linux, they will be listed as /dev/sTTY# (or similar)
  - The number may change when unplugging and plugging in the Arduino
  - Your computer may have COM ports unrelated to the Arduino
- You can use the “Get Board Info” to check if you have the correct port, since that should give you the name of the microcontroller



The tools menu is primarily for settings related to communications with the Arduino

Serial Monitor

- The Magnifying Glass on the top-right and the “Serial Monitor” option both will attempt to open a Serial Monitor for sending data to and from the Arduino
- This uses the same port setting from the tools menu
- Any text output to the Arduino Serial (AKA Serial0) port will be printed here
- Any text typed into the message box will be sent to the Arduino when you hit enter
  - The NL, CR drop-down allows you to add a Newline or Carriage Return (move the cursor to the left) when sending text to the Arduino
- The baud setting is the rate of data transmission and must be the same in the serial monitor and Arduino (set by the program uploaded)



A successful connected Serial monitor, notice how the very bottom of the IDE shows the Arduino connected on COM12

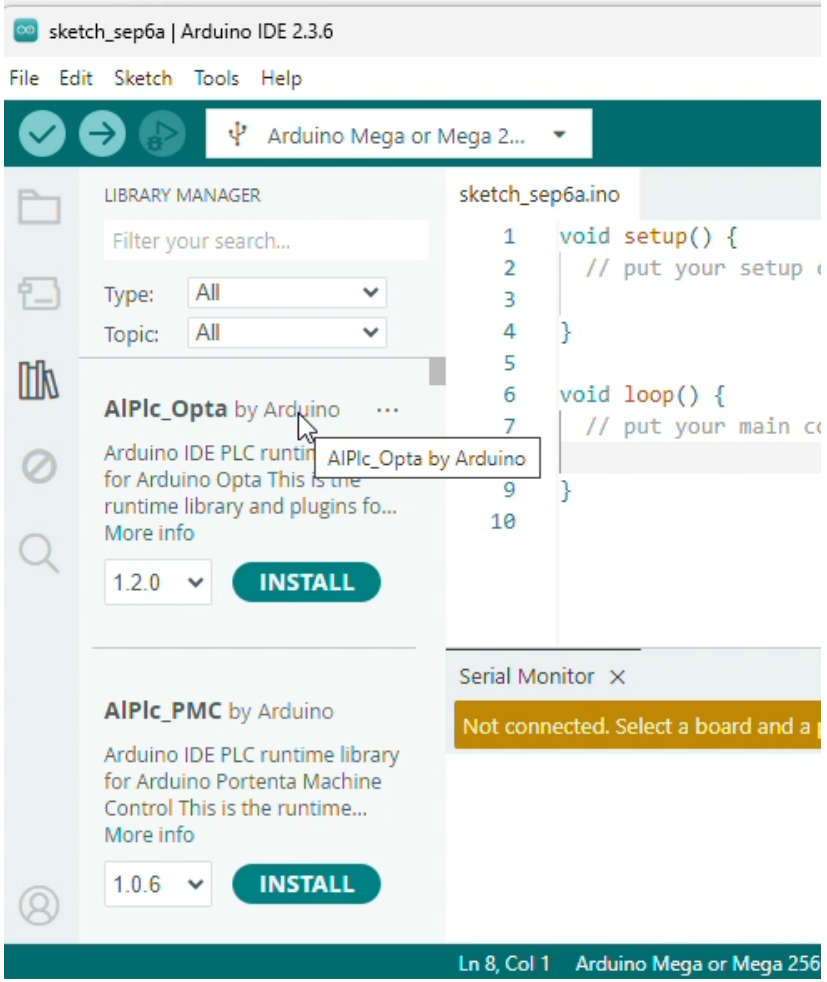
Libraries

- Libraries are a collection of prewritten modules that have their own functions and variables for use in your program
- The example below uses a library named pitches.h
  - The #include line tells the compiler to use the library (notice the lack of semicolon)
- The Notes are all variables defined in the library
- The function tone() is a function that plays a note

```
#include "pitches.h"

int melody[] = {NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4};

int noteDurations[] = {4, 8, 8, 4, 4, 4, 4, 4};

tone(8, melody[thisNote], noteDuration);
```

Library Manager

- Libraries must be installed on your computer before they can be added to a program
- The IDE includes a library manager accessed by the book icon on the left
  - By typing in the search bar, you can look for a library in the online Arduino libraries
  - The number dropdown allows you to pick a specific version
  - Clicking install will download it and copy it to your sketchbook folder
  - Clicking more info will take you to a documentation page for that library
- Once installed, the library can be added by using the “Include Library” option under the sketch menu
  - Many libraries also come with example sketches, which can be accessed by the “Examples” option under the File Menu



Board Manager

- For each of the different Arduino Microcontrollers, there is a set of built-in libraries
- These deal with the capabilities of the Microcontroller, like Serial ports, or IO pins
- These also tell the IDE how to compile and upload the code
- These libraries are managed by the “Board Manager” just above the Library Manager icon
- Once installed, the board will be available to select under “Board Option” in the tools menu
  - An include statement is not needed for the board libraries; they are automatically included by selecting a board