

# Installing and Navigating the Arduino IDE



#### What is an IDE

- An integrated development environment (IDE) is software designed to increase productivity by integrating useful tools into one application
- These tools vary between different IDEs, but commonly include a source code editor, compiler, and debugger



#### Source Code Versus Machine Code

- Source code is designed for human readability and uses textual syntax that is translated into machine code
- Machine code is low-level binary data written for a computer that does not need additional translation



# Source Code Editor

- A text editor designed specifically for editing code
- Features:
  - Syntax highlighting and brace matching
  - Automatic indentation
  - Auto-complete word prediction that fills in common words or phrases as the programmer is typing



- Brace matching: Clicking beside a brace puts a box around its counterpart (orange arrows)
- Syntax highlighting groups elements by color; void is a data type and loop is a function (red box)



#### Compiler

- Used to convert one language into another language
- Converts the source code into machine code for the computer to read



### Debugger

- Software designed for testing the source code
- Oftentimes, the debugger will offer suggestions based on expectations to help the programmer identify and resolve issues



**Common IDEs** 

- Arduino
- Visual Studio
- Eclipse
- Komodo
- Android Studio
- NetBeans
- Atom
- BlueJ



# The Arduino IDE

- Arduino software is open source and can be downloaded for free at <u>https://www.arduino.cc/en/Main/Software</u>
- Compatible with Windows, Mac OS X and Linux systems





## Choosing the Correct Package

 Select download link for the appropriate operating system

#### Download the Arduino IDE



#### ARDUINO 1.8.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the **Getting Started** page for Installation instructions. Windows Installer, for Windows XP and up Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits Linux 64 bits Linux ARM

Release Notes Source Code Checksums (sha512)



#### Installation

 Follow on-screen instructions to run the installation software (example is for Windows)

Arduino Setup: License Agreement     Please review the license agreement before installing Ardui	uino. If you	Chec	k the components you	want to install and uncheck the compor
accept all terms of the agreement, click I Agree.		you o	lon't want to install. Cli	ick Next to continue.
SNU LESSER GENERAL PUBLIC LICENSE	Â	Select compo	nents to install:	Install Arduino software
Copyright (C) 2007 Free Software Foundation, Inc. < <u>http://fsf.or</u>	r <u>a/</u> >			✓ Install USB driver ✓ Create Start Menu shortcut
Everyone is permitted to copy and distribute verbatim copies of th document, but changing it is not allowed.	nis license	đ		Create Desktop shortcut Associate .ino files
This version of the GNU Lesser General Public License incorporates and conditions of version 3 of the GNU General Public License, sup by the additional permissions listed below.	s the terms plemented	t Space require	ed: 482.3MB	
Cancel Nullsoft Install System v3.0	I Agree	Cancel	Nullsoft Install S	ystem v3.0 < Ba
3.	Arduino Setup: Installation Folde     Setup will install Arduino in the     folder, click Browse and select     installation.     Destination Folder     [C: \Program Files (x86)\Arduino\	r — — X following folder. To install in a different another folder. Click Install to start the Browse		



## Opening the IDE

• Locate the Arduino icon on your desktop and double click to open the program





Should give you this



### Navigation

- The toolbar at the top of the IDE contains functions for file, edit, sketch, tools, and help
- The next line gives shortcuts for verify, upload, new, open, and save
- The tab at the bottom shows the name of the current sketch





# The Serial Monitor

- The icon of the magnifying glass in the top right-hand corner opens the **serial monitor**
- This is a pop-up window that allows the programmer to see interactions as the code runs





# File

- The file folder contains features such as new, open, save, and print
- The sketchbook subfolder contains a collection of code written by the programmer
- The examples subfolder contains fully functional code <u>written for</u> the programmer to assist with common tasks

File	Edit Sketch 1	Fools Help	
	New	Ctrl+N	
	Open	Ctrl+O	
	Open Recent		>
	Sketchbook		>
	Examples		>
	Close	Ctrl+W	
	Save	Ctrl+S	
	Save As	Ctrl+Shift+S	
	Page Setup	Ctrl+Shift+P	
	Print	Ctrl+P	
	Preferences	Ctrl+Comma	
	Quit	Ctrl+Q	



#### File: Examples

• The **examples** subfolder covers a wide range of processes

le Edit Sketch	Tools Help	Built-in Examples	
Onen	Ctrl+N	01.Basics	PlinkWithoutDelay
Open Recent	2	03.Analog	Button
Sketchbook	3	04.Communication	Debounce
Examples	;	05.Control	DigitalInputPullup
Close	Ctrl+W	06.Sensors	StateChangeDetection
Save	Ctrl+S	07.Display	toneKeyboard
Save As	Ctrl+Shift+S	08.Strings	toneMelody
Page Setup Print	Ctrl+Shift+P Ctrl+P	09.USB 10.StarterKit_BasicKit 11.ArduinoISP	toneMultiple tonePitchFollower
Preferences	Ctrl+Comma	Examples for any board	
Quit	Ctrl+Q	Adafruit Circuit Playground	>

LSU rev20AUG2020



# File: Preferences

- The preferences subfolder allows you to customize the IDE
- Display line numbers
   is a popular feature
   that adds or removes
   line numbers from
   the source code





# Edit

- Edit provides shortcuts for useful features such as undo, redo, cut, copy, and paste
- You can increase or decrease indentions as well as font size
- A simple shortcut lets you comment or uncomment an entire section of highlighted code

File	Edit	Sketch Tools Help		
		Undo	Ctrl+Z	
		Redo	Ctrl+Y	
S		Cut	Ctrl+X	H
1		Сору	Ctrl+C	
2		Copy for Forum	Ctrl+Shift+C	×
3		Copy as HTML	Ctrl+Alt+C	
4	5	Paste	Ctrl+V	
5		Select All	Ctrl+A	
6		Go to line	Ctrl+L	
7		Comment/Uncomment	Ctrl+Slash	ł
8		Increase Indent	Tab	
		Decrease Indent	Shift+Tab	
		Increase Font Size	Ctrl+Plus	
		Decrease Font Size	Ctrl+Minus	
		Find	Ctrl+F	
		Find Next	Ctrl+G	
		Find Previous	Ctrl+Shift+G	



#### Sketch

- **Sketch** provides shortcuts for verifying, compiling, and uploading code
- It allows you to include libraries or incorporate files into the code as needed

File Edit	Sketch	Tools Help	
00	V	erify/Compile	Ctrl+R
	U	pload	Ctrl+U
sketch	U	pload Using Programmer	Ctrl+Shift+U
1 <b>v</b> o	Б	(port compiled Binary	Ctrl+Alt+S
2 3 4 }	SI In A	now Sketch Folder Iclude Library dd File	Ctrl+K

LSU rev20AUG2020



# Verifying and Uploading Code

- Verifying checks for problems
- Uploading verifies and sends the code to a microcontroller
- It is recommended to verify often as it is easier to locate mistakes from a short line of code as opposed to an entire script



### Libraries

- Libraries are a collection of precompiled modules that use keywords to activate functions
- The example below uses a library named pitches.h for its preset melody, duration, and notes without the need for additional code

#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);
```



#### LaACES Student Ballooning Course How to Add a Library

• Select Manage Libraries subfolder

Ð	Verify/Compile	Ctrl+R Ctrl+U	Manage Libraries	Ctrl+Shift+1
etch	Upload Using Programmer	Ctrl+Shift+U	Add .ZIP Library	
vo	Export compiled Binary	Ctrl+Alt+S	Arduino libraries	
	Show Sketch Folder	Ctrl+K	Bridge	
	Include Library	3	EEPROM	
}	Add File		Esplora	

• Type keyword to locate a library and select install to download



• Locate and select new library in include library subfolder





#### Tools

- Tools is where the board and processor are selected
  - Code will not run properly if this does not match your equipment
- Use 'Get Board Info' if you are unsure about the microcontroller being used







#### Tools: Port

- The serial port number is determined by the microprocessor
- The following link contains instructions for determining the correct port for Windows, Mac, and Linux:

http://www.me.umn.edu/courses/me2011/arduino/techn otes/debug/arduinodebug.html

Board: "Arduino/Genuino Mega or Mega 2560"	>	
Processor: "ATmega2560 (Mega 2560)"	>	
Port: "COM10 (Arduino/Genuino Mega or Mega 2560)"	>	Serial ports
Get Board Info	~	COM10 (Arduino/Genuino Mega or Mega 2560)



# Help

If ever stuck,
 help offers links

 for getting
 started,
 frequently asked
 questions, and
 troubleshooting

File	Edit	Sket	ch	Tools	Help	
Sk	() (etch_	jul0	8a §			Getting Started Environment Troubleshooting
1	vo	id	se	tup		Reference
2 3 4 5 6 7	} vo.	// id //	pu lc pu	oop( it y		Galileo Help Getting Started Troubleshooting Edison Help Getting Started Troubleshooting
8	}					Find in Reference Ctrl+Shift+F Frequently Asked Questions Visit Arduino.cc About Arduino



#### **Baud Rates**

- The **baud rate** is the rate by which information is transferred
- The serial monitor will display unintended caricature if the baud rate on the monitor is not set to match what is dictated in the code





# Debugging Code

- The debug window provides information based on normal expectations
- The example shows two conflicting data types assigned to one variable, int and word
- The debugger highlights the potential error in the text window (top) and alerts the programmer in the debug window (bottom)

V	roid loop() {
	// read the sensor:
	<pre>int word sensorReading = analogRead(A0);</pre>
	// print the sensor reading so you know its range
e	expected initializer before 'sensorReading'
	^
e	exit status 1
е	expected initializer before 'sensorReading'
LSU rev20AU	2020 L07.02



### **Getting Started**

- Arduino breaks the sketch into two parts, void setup () and void loop ()
- Setup runs once whereas loop runs repeatedly





#### LaACES Student Ballooning Course Global Declarations

- Libraries and global variables go outside of the two main functions (setup and loop) and are visible to every line in the code
- The following example defines the libraries needed to initiate an SD card reader on lines 2 and 3
- It defines global variables on lines 5, 6, and 7

1
2
3#include< <mark>SD.</mark> h>
4
5 const int chipSelect = 10;
6 int error = 0;
7 long timeStamp;
8
<pre>9 void setup() {</pre>
0 Serial.begin(9600);



# Void Setup

- Functions that only need to run once go under **void setup**
- Can declare **baud rate** and **initialize system checks**
- In the example, the SD card writes via chipSelect (pin 10 defined on slide 28). If it fails to write, it prints an error message in the serial monitor (at a rate of 9600) and stops running the code





# Void Loop

- Location of the main code that will loop repeatedly
- In the example, the variable dataString will store the reading from a sensor (sensorVal) with 4 measurements taken and it will create a timeStamp that repeats once every millisecond (millis); this will repeat indefinitely

```
22 void loop() {
23 String dataString = "";
24 timeStamp = millis();
25
26 for (int analogPin = 0; analogPin < 4; analogPin ++)
27 {
28 int sensorVal = analogRead(analogPin);
29 dataString += String(sensorVal);
</pre>
```



# Troubleshooting

- Check syntax
  - Missing semicolons, brackets, etc.
- Libraries
  - Proper syntax and keywords; some libraries may conflict with one another
- Correct baud rate
- Case sensitive context
- Correct data types



#### References

- <u>https://www.freeiconspng.com/img/12780</u>
- <u>https://www.iconfinder.com/icons/37037/apple\_face\_finder\_mac\_os\_x\_mettalic\_icon</u>
- <u>https://www.flaticon.com/free-icon/linux 518713</u>