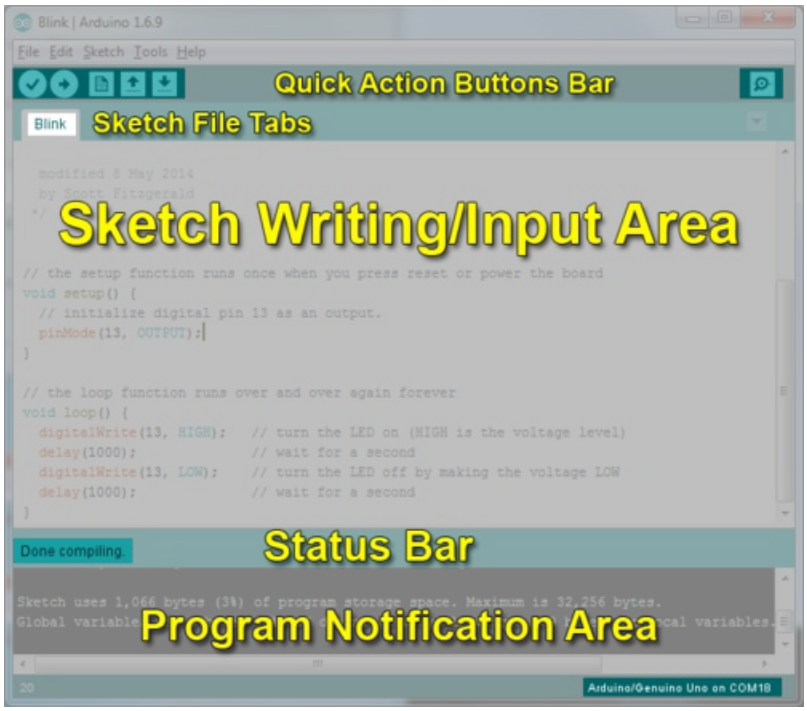
adafruit learning system

Ladyada's Learn Arduino - Lesson #1

Created by lady ada



Last updated on 2018-08-22 03:55:10 PM UTC

Guide Contents

Guide Contents 2
Introduction 3
Lesson Parts 4
Do you have everything you need? 4
Upload Your First Sketch 6
Startup! 6
Getting Ready To Do This Thing! 7
Select Board Type Arduino UNO 7
Select Correct Serial Port 8
~~ Hints! ~~ 9
Open Blink Sketch 9
Verify / Compile 10
Upload 11
Things That Can Go Wrong... 12
Arduino bootloader doesn't respond 12
Can't open serial port device 13
Video of all steps 13
Watch! 13
Experiments 15
How long does Blink run for? 15
What does the Reset button do? 15

Introduction

Ah yes, it is finally time to make your Arduino do something! We're going to start with the classic **hello world!** (https://adafru.it/rc6) of electronics, **a blinking light**. OK it doesn't *sound* too exciting, heck you can just flip your desk lamp on and off without needing a microcontroller.. but I promise you, you'll learn a lot!

This lesson will basically get you up and running using the Arduino software and uploading a sketch to the Arduino board. Once you've completed this step we can continue to the really exciting stuff, which is when we start writing our own sketches!

These instructions mostly show Windows software. Except when indicated, the software is identical on all platforms.

Lesson Parts

Do you have everything you need?

Not much is needed for this lesson, just a USB cable and an Arduino or compatible.

Make sure you've gone through Lesson #0 (https://adafru.it/pcg) first! This lesson assumes you have installed Arduino IDE software and drivers!

Assembled Arduino board, preferrably an Uno or Duemilanove (or whatever the latest version is)

Arduino compatibles will work **but** there's a lot of issues with ultra low cost 'Arduino compatibles' (e.g. eBay, Amazon, etc) where they have shoddy substitutions that can bite you later. It's good to have at least one known-genuine Arduino UNO!

Available at Adafruit (http://adafru.it/50)

OR

You can also use an Adafruit Metro which is a drop-in replacement for the UNO, some components like the LEDs are in different locations.

Available at Adafruit (http://adafru.it/2488)

AND

USB Cable, any length. The cable should match your Arduino's USB connector. Official Arduino UNOs use USB "Printer Cable", a blocky cable. Some compatibles use USB Mini-B or Micro-B.

USB Cables available at Adafruit (https://adafru.it/zem)

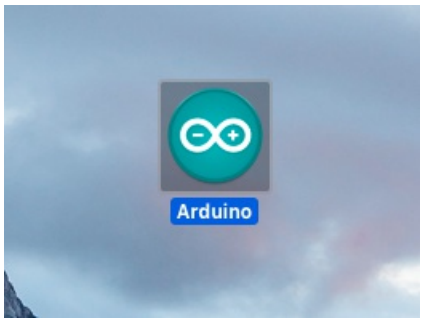
A HUUUUUUGE number of people have problems because they pick a 'charge only' USB cable rather than a "Data/Sync" cable. Make 100% sure you have a good quality syncing cable. Srsly, I can't even express how many times students have nearly given up due to a flakey USB cable!

Upload Your First Sketch

OK don't forget - you have to go though the Lesson #0 steps (https://adafru.it/pcg) to install drivers, software, and verify that your Arduino is powered, has the bootloader installed and shows up as a COM/Serial port... Don't continue unless that has been done!

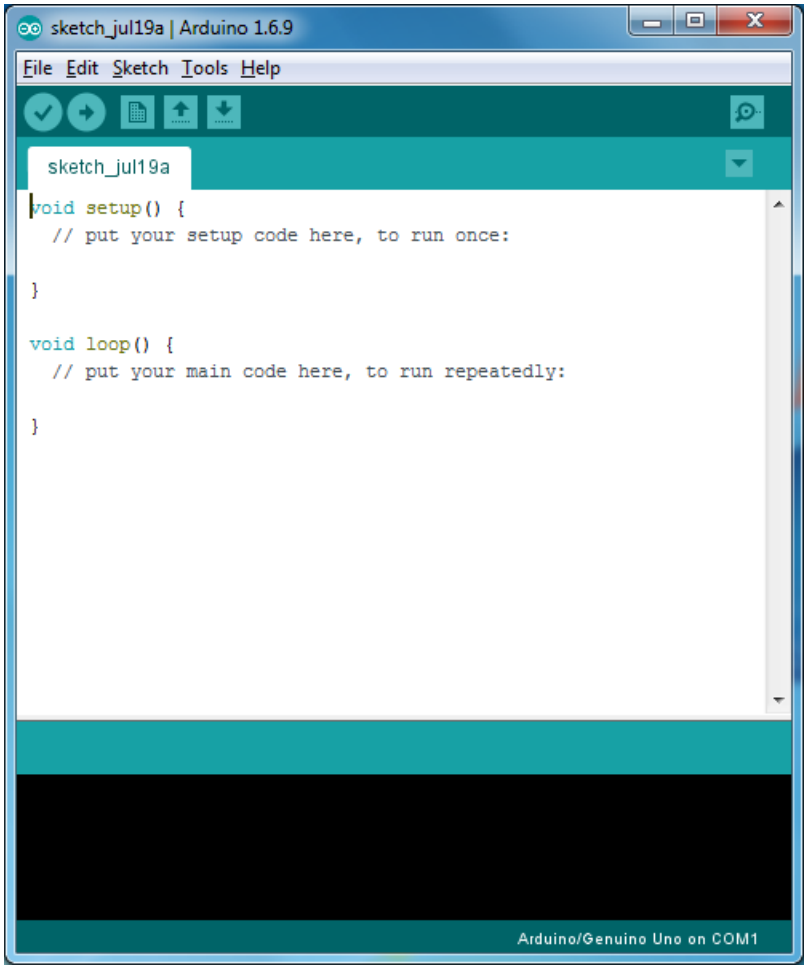
Startup!

Double click the Arduino software icon on your desktop

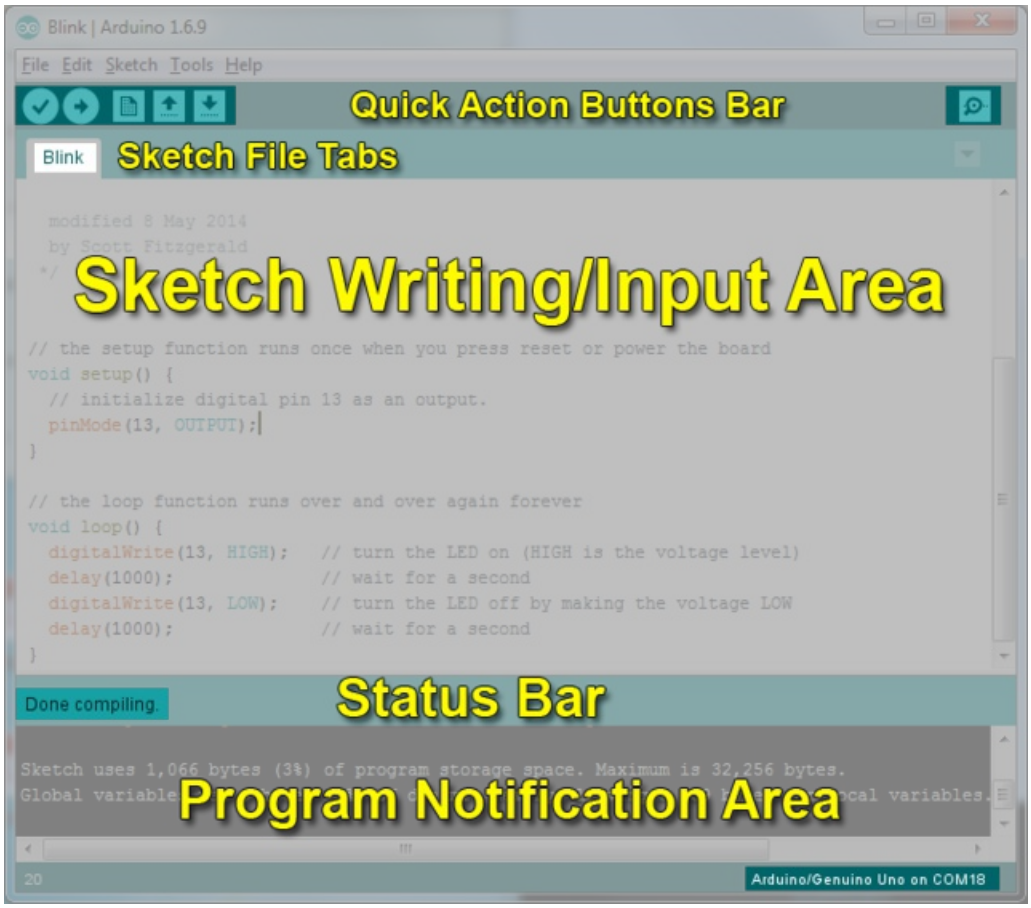


To open up the workspace, also called the IDE. IDE stands for **I**ntegrated **D**evelopment **E**nvironment. Basically, a word processor for writing code. The full name is really a mouthfull, so we say I.D.E. (Eye Dee Eee)

Anyhow, it looks like this:



The main IDE window has multiple tabs and areas, we'll visit all of these shortly, so keep your eye on this diagram!



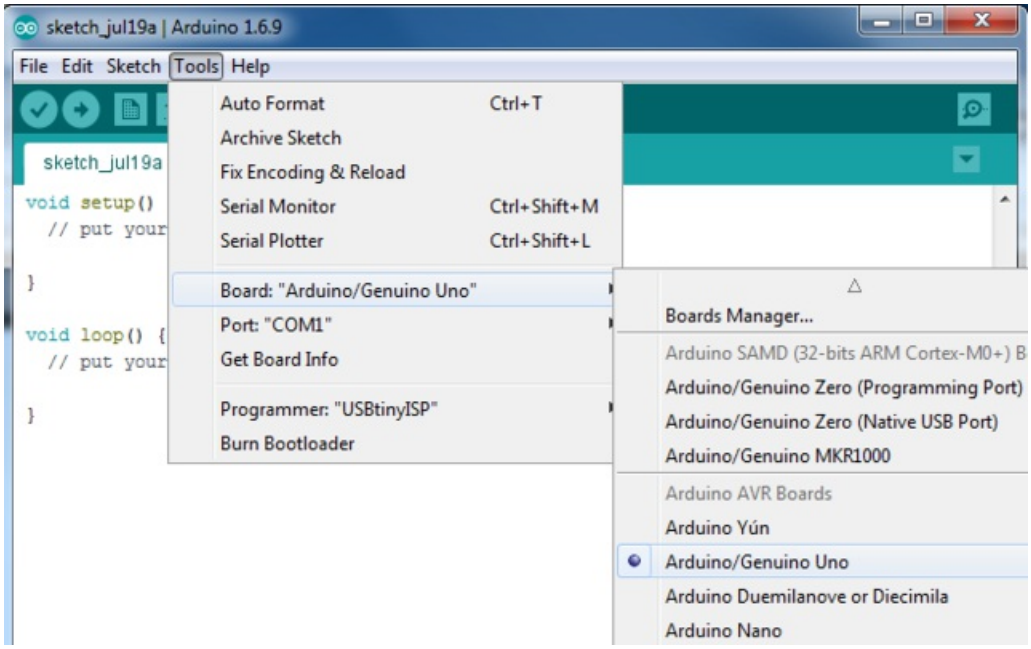
Getting Ready To Do This Thing!

We're going to jump right in! Before you can run a program on the Arduino you need to tell the IDE where to find it and what kind you're going to be programming (as we mentioned in lesson #0 there are *dozens* of types of Arduinos!)

Select Board Type **Arduino UNO**

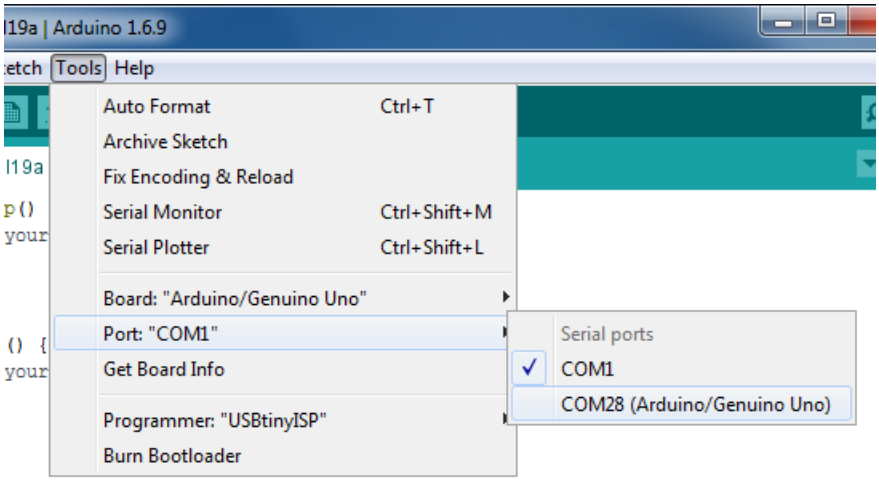
Under the **Tools** menu, find the **Board** submenu and navigate that to select **Arduino (Genuino) UNO**.

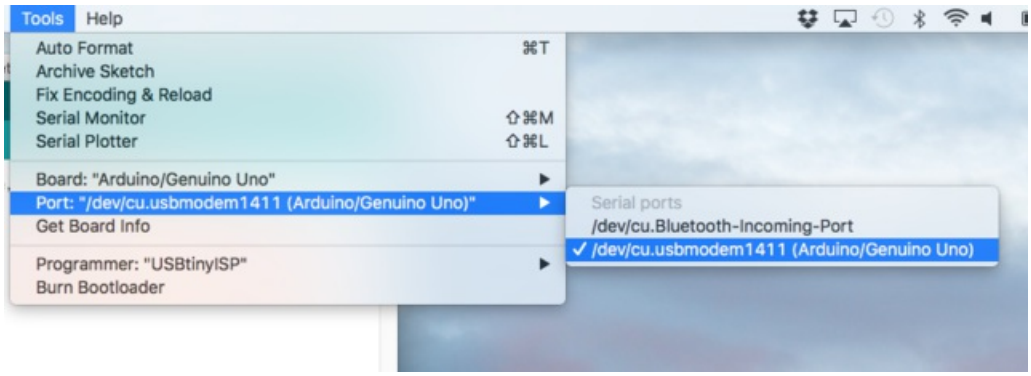
You should use this board selection also if you are using an **Adafruit Metro** or *any other Arduino UNO compatible*



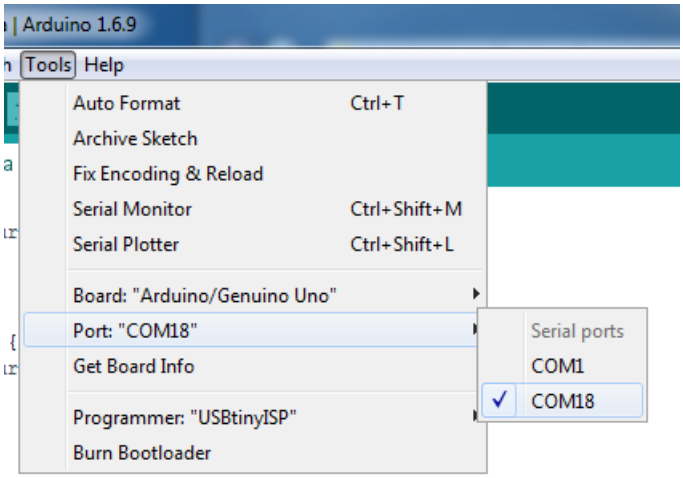
Select Correct Serial Port

Most important you will *also* need to select the correct Serial port. **This is where people can get tripped up**. The good news is if you have a genuine Arduino UNO, the name will appear next to the serial port menu item to make it easy to pick out!





If you have an Arduino compatible like Adafruit Metro, you may not see anything after the Serial port name, but it will still be the only option available that is *not* COM1 or something with Bluetooth in the name



~~ Hints! ~~

Windows: It will never be `COM1` don't pick that one. You should only have one other option.

Mac OS X and **Linux**: It will never be an option with 'bluetooth' in the name, look for `/dev/cu.usbmodem` or `/dev/cu.usbserial` or `/dev/ttyUSB` or similar!

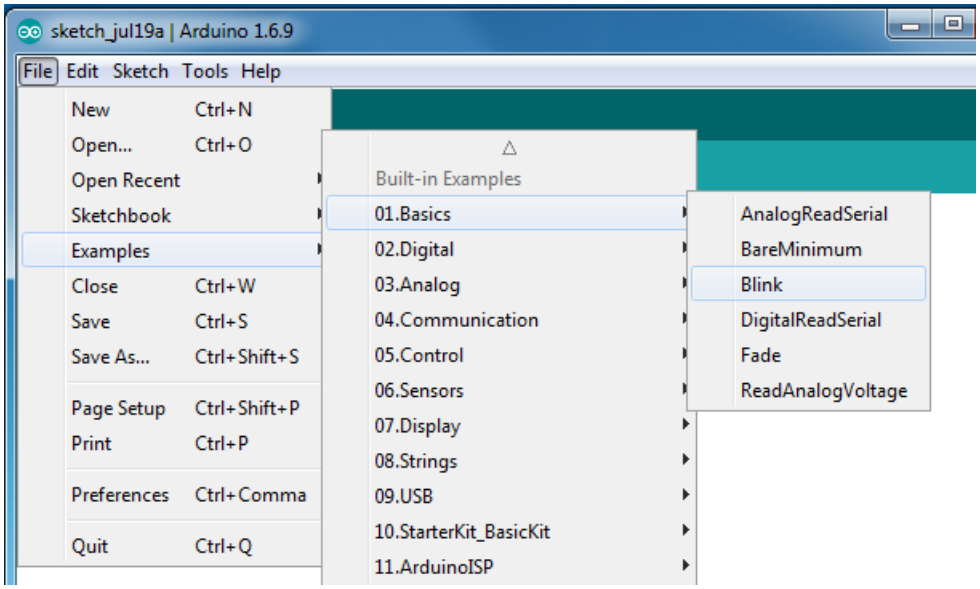
> Have lots of Serial ports? Try unplugging your Arduino and seeing which port disappears next time you look at the Ports list!

The board and serial port preferences are saved so you only have to set it once, the program will remember next time it's run.

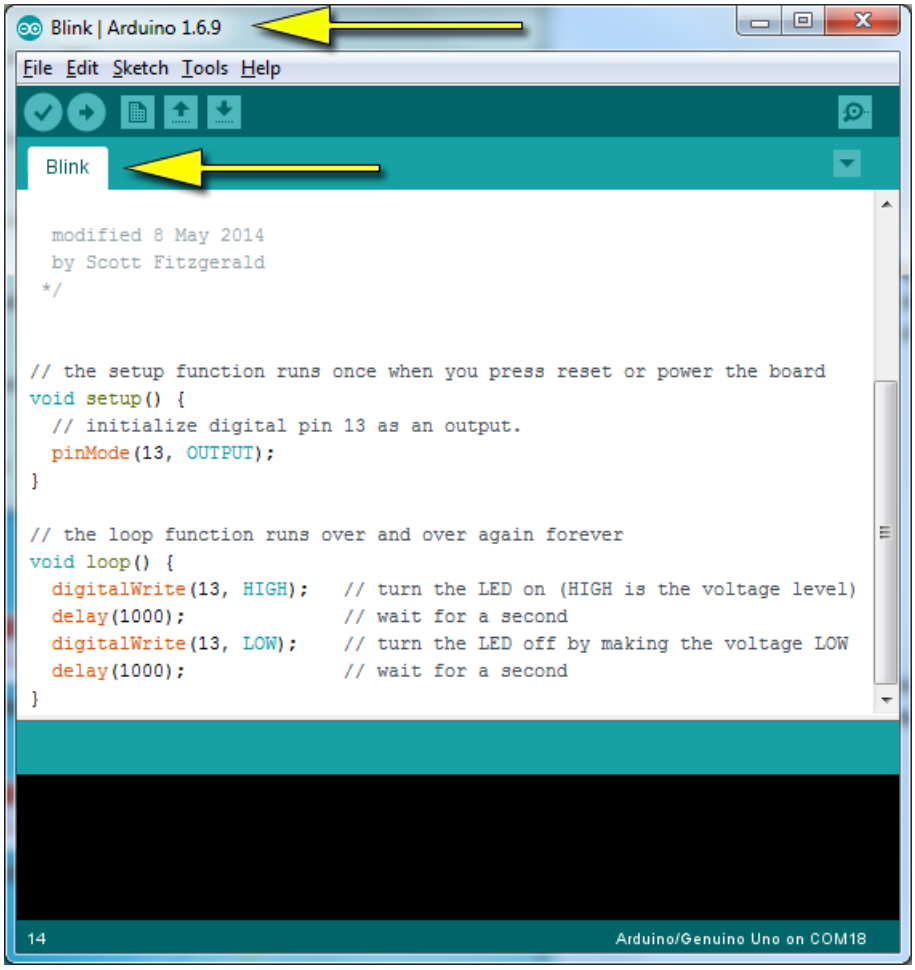
However, **if you have multiple Arduino's,** they may be assigned difference COM ports. So every time you plug in a new Arduino, double check that the correct port and type is selected!

Open Blink Sketch

Sketches are little scripts that you can send to the Arduino to tell it how to act. Let's open up an **Example Sketch**. Go to the **File menu -> Examples -> 01.Basics -> Blink**



This should open up a new window that should now look like this, with a bunch of text in the center white space. Above the text is a tab labeled **Blink**



Verify / Compile

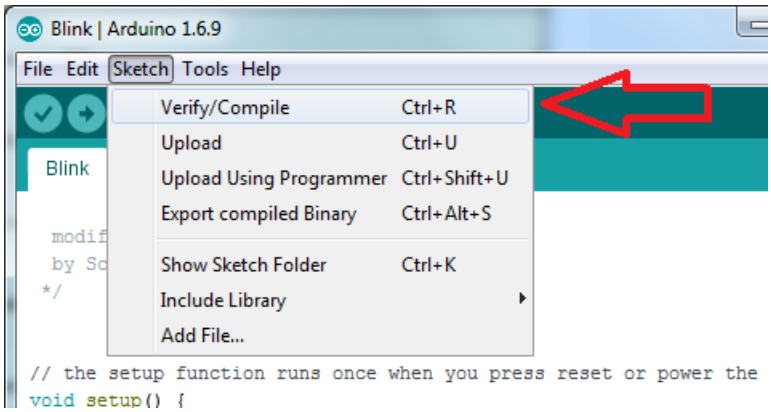
Let's keep going! The first step to getting a **Sketch** ready for transfer over to the arduino is to **Verify/Compile** it. That

means check it over for mistakes (sort of like spell-checking or grammatical editing) and then translate it into an app that is compatible with the Arduino hardware.

> **Verify:** Like having a friend look over your homework essay before handing it in, *verifying* means the Arduino software will check over and look for typos, common errors - it can't catch *all* errors, just like a spellcheck wont be able to tell that you spelled "bear" like "bare" by accident since both are proper words
>
> **Compile:** check your music player and you'll probably find you own at least one *compilation,* which means a collection that someone put together in a specific order. When Arduino *compiles* your sketch, it is putting/arranging it together into the right order for your Arduino hardware board to be able to run

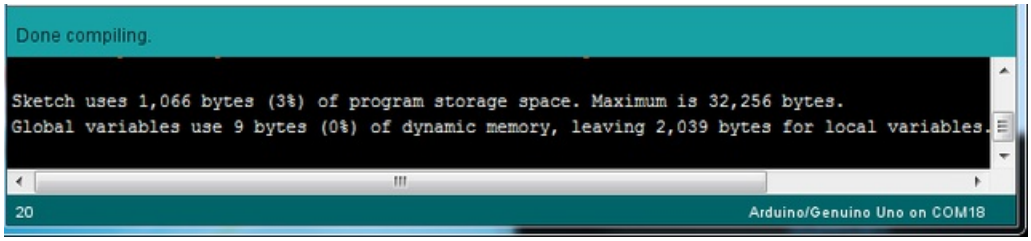
While in theory you could have Arduino do each one separately, it's faster to just have it do both at the same time

You can start the action via the **Sketch** menu



After a few seconds, you should see the message `Done compiling.` in the **Status Bar** and `Sketch uses ... bytes (x%) of program storage space` (or something similar) in the **Program Notification Area.**

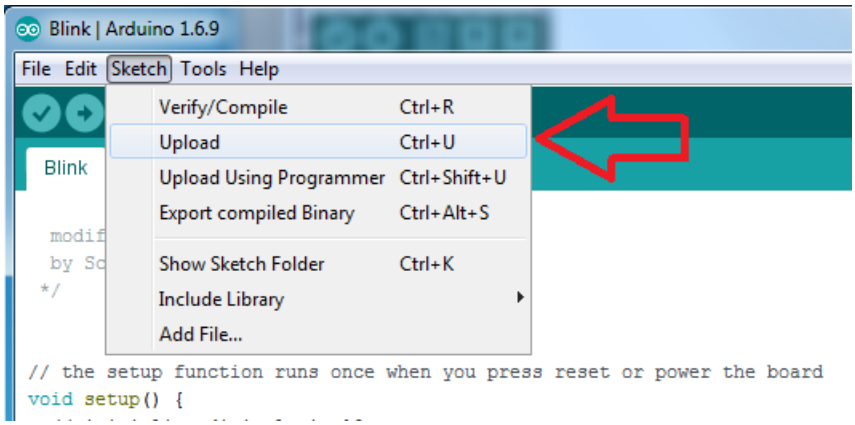
This means the sketch was well-written and is ready for uploading to the Arduino board!



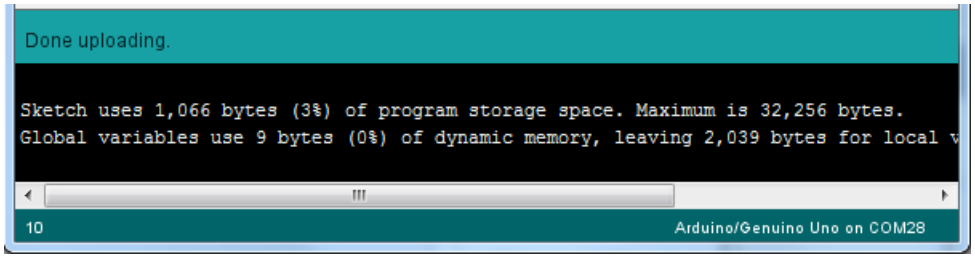
Upload

Ready for the moment of truth? Now it's time to upload your very first sketch. Make sure the Arduino is plugged in, the green light is on and the correct board and Serial Port is selected.

Select **Upload** from the **Sketch** menu



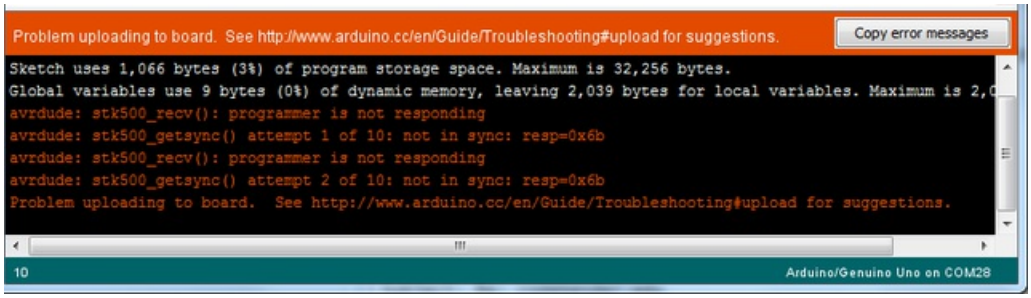
After a few seconds you should get this screen, with the message **Done uploading.** in the status bar.



Things That Can Go Wrong...

Arduino bootloader doesn't respond

If you get the following error message `avrdude: stk500_getsync(): not in sync` or `avrdude: stk500_recv(): programmer is not responding` that means that the Arduino bootloader is not responding



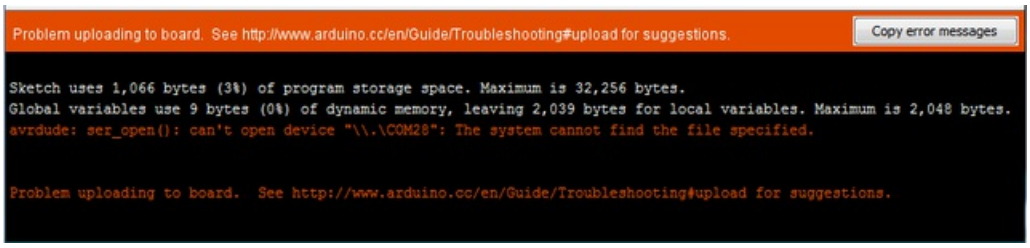
This error can be caused by a lot of issues. Check the following:

- Is the correct Arduino Board selected?
- Is the correct Serial Port selected?
- Is the correct driver installed?
- Is something connected to the Reset pin or pressing the reset button?
- Try disconnecting *all* connected shields, jumper wires and components
- Is the chip inserted into the Arduino properly? (If you built your own arduino or have burned the bootloader on yourself)
- Does the chip have the correct bootloader on it? (If you built your own arduino or have burned the bootloader on yourself)

You can also check Lesson #0 on how to verify if the bootloader is installed (https://adafru.it/pch)

Can't open serial port device

If you get an error like `avrdude: ser_open(): can't open device` it likely means your Arduino got disconnected from USB somehow



This error can be caused by a few possible issues. Check the following:

- Is the correct Serial Port selected?
- Is the correct driver installed?
- Try unplugging/replugging the USB cable
- Try another USB cable or USB port
- Your computer's USB system may have crashed. Shutdown your computer, disconnect power, wait 3 minutes then restart it

Video of all steps

Here is a video showing the timing of the steps described so far, opening the Blink sketch and setting the board and serial port.

Watch!

If you have a UNO Arduino, the upload process is quite fast: Once you click Upload from the software the IDE will ask the Arduino to *reset* itself into the bootloader and immediately start uploading the **blink** sketch. After uploading, the sketch is quickly verified - the IDE asks the Arduino 'please repeat back what I just asked to you to do' - and reset again to start running the sketch.

The little translucent **L** LED will start blinking on and off, a second on and a second off.

Here's a looping animation showing upload and then 3 blinks: **notice the RX and TX LEDs blink during upload, but it happens very fast!**


AREF
GND
DIGITAL (PWM~)
TX
RX
ARDUINO
UNO
ON
ICSP
MADE IN ITALY
ARDUINO.CC
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
Vin
A0
A1
A2
A3
A4
A5

Experiments

How long does Blink run for?

Once you have Blink running. Try this experiment:

Unplug your Arduino, wait a few moments and plug it back in. Is **Blink** still running on your Arduino (LED blinking on and off once every second)?

Walk away from your computer and go have a cup of tea. Then come back, is **Blink** still running?

Now go to sleep for 8 hours, wake up and eat breakfast. Then check back on your Arduino, is **Blink** still running?

Aspire to become a monk. Leave your home with only the clothes on your back and an Arduino in your pocket. Travel the world for 10 years, learning about yourself and the depth of your own existence. In a distant land, find a power plug or USB cable and plug in your Arduino, is **Blink** still blinking?

The answer is (and would be if you had 10 years): **yes!** Blink will run essentially *forever* on your Arduino. It isn't like an app you have to launch or upgrade. The Blink program is saved into the FLASH storage of the chip, which is like a very very small USB key. Even if you lose power and forget about your Arduino for years, you can plug it back in and it will run the same sketch it did last time.

What does the Reset button do?

While looking at the LED, hold down the Reset button? What happens to the LED? It should have stopped blinking. Holding down the reset button will halt the Arduino temporarily.

While looking at the LED, release the Reset button. You'll see a quick triple-flash from the Arduino and then it will return to turning on and off the LED once every second