Arduino Lesson 1

Learning Objectives

- Download Arduino IDE
- Recognize microprocessor components
- Upload sketch to test board
- Learn some coding basics!

The hardest part of making an instrument with an Arduino is often the first step, getting the computer to talk with the Arduino microprocessor. Complete this lesson PRIOR to coming to the Arduino lab day. This might require visiting the professor prior to class, so don't wait until the last minute.

**These directions are meant to be read alongside the video located at* http://lesliehiatt.com/arduino-lessons-for-students/arduino-lesson-1/

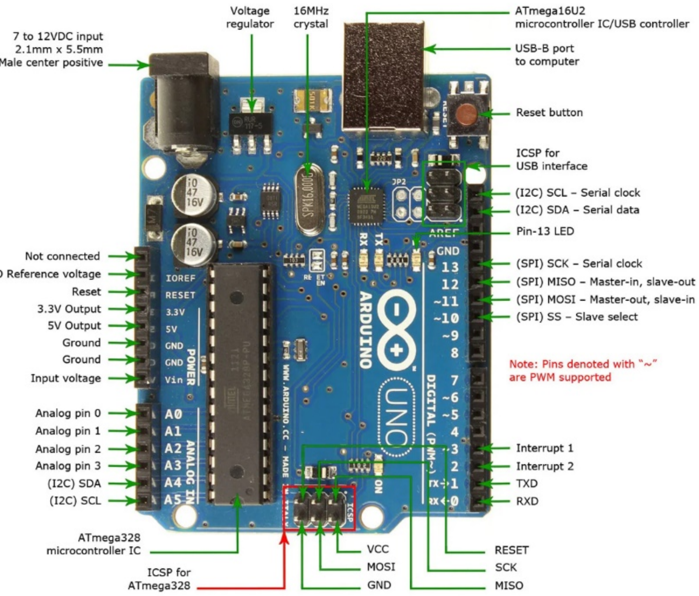
Directions

Installing the IDE (Video times 0:00-1:31)

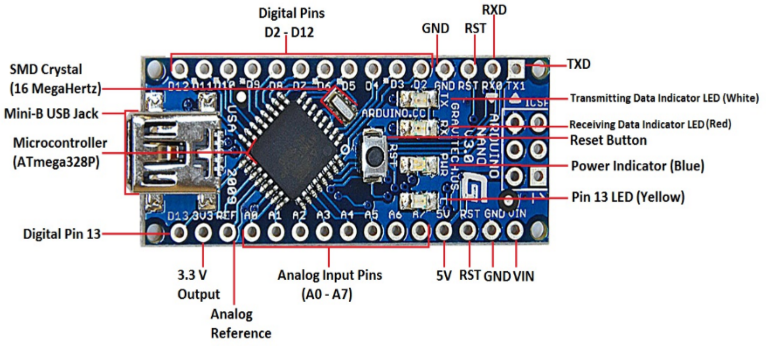
1. Go to https://www.arduino.cc/en/Guide to download the IDE for the Arduino microprocessor.
2. Under the header "Install the Arduino Desktop IDE" click on the appropriate hyperlink for your computer's operating system. This page has all the necessary information to guide you through the installation process.
3. Click on the hyperlink that says "download page" to download the IDE.
4. Under "Download Options" on the right side of the page, click on the appropriate option for your computer's operating system.
5. Click the button labelled "Just Download" to download the software for free.

Introduction to Arduino (Video times 1:32-3:46)

To learn about Arduino, first identify which one you have. The images below show the pinouts for two Arduinos you may potentially have. (There are many off brands such as "Elegoo", but if it has the general pin setups shown below, then it is an Uno or a Nano. There are many other types, but these and the Mega are the most popular.) These pictures are called "pinout diagrams" or "schematics" and can be found by typing the name of the arduino followd by "schematic" (ex. Arduino Nano Schematic). These diagrams will be your best friend when writing/modifying code so keep them close by!



Arduino Uno

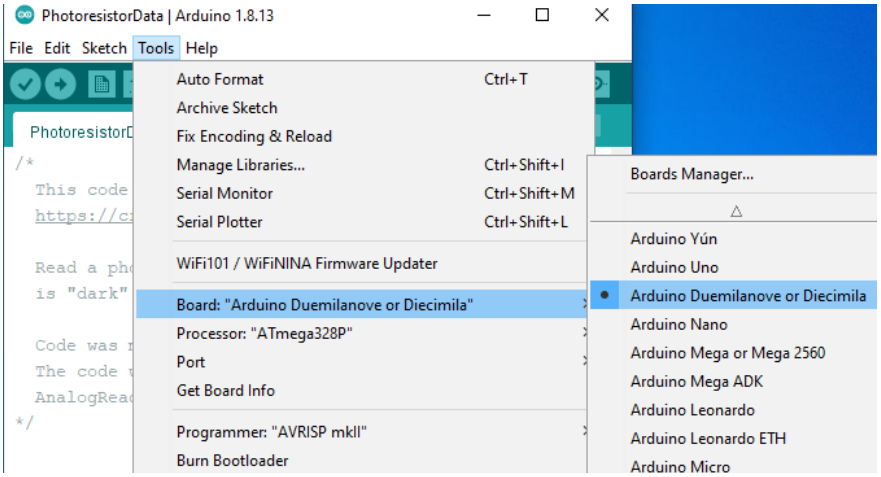


Digital Pins D2 - D12
GND
RST
RXD
TXD
SMD Crystal (16 MegaHertz)
Mini-B USB Jack
Microcontroller (ATmega328P)
Transmitting Data Indicator LED (White)
Receiving Data Indicator LED (Red)
Reset Button
Power Indicator (Blue)
Pin 13 LED (Yellow)
Digital Pin 13
3.3 V Output
Analog Reference
Analog Input Pins (A0 - A7)
5V
RST
GND
VIN
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN
ARDUINO.CC
ARDUINO NANO V3.0

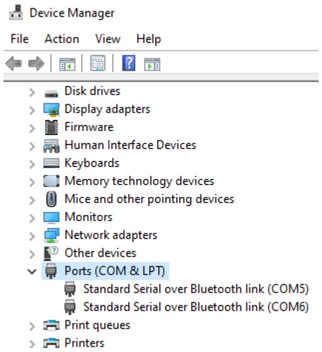
Arduino Nano

Interfacing with Your Computer (Video times 3:47-7:32)

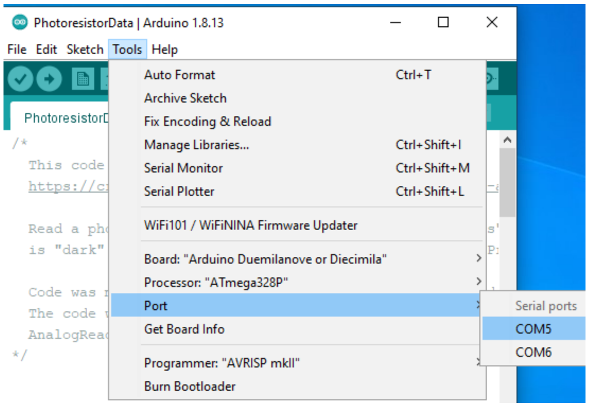
1. Begin by plugging the Arduino into your computer with the USB cable. Your computer will most likely make a noise, and a power indicator light will activate on your Arduino.
2. Open the IDE installed earlier and navigate to the “Tools” button on the top left ribbon.
3. Move the mouse cursor over the “Board: “Arduino Nano” and it will open a window with a bunch of options for different types of Arduinos. Click on the one which correlates with the microprocessor being used. The nano often shows up under “Arduino Nano” or as “Arduino Duemilanove or Diecimila”.


PhotoresistorData | Arduino 1.8.13
File Edit Sketch Tools Help
PhotoresistorD
/*
This code
https://c
Read a ph
is "dark"
Code was
The code
AnalogRea
*/
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
Board: "Arduino Duemilanove or Diecimila"
Processor: "ATmega328P"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino Yún
Arduino Uno
● Arduino Duemilanove or Diecimila
Arduino Nano
Arduino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino Micro

4. Now find which COM port your Arduino is on. Firstly, type “device manager” into the search bar at the bottom left of the computer (If using Windows 10). Open device manager and click on the dropdown option that says “Ports (COM & LPT)”. Plug and unplug the Arduino from the computer while looking at device manager to see which port goes away when you unplug the Arduino and then comes back when plugged back in: this is the port the Arduino is using. (For Mac users, the program being searched for is “System Information” and it should function similarly) Some computers have difficulties identifying the Arduino Nano devices as ports (Microsoft Surface). If this is the case, ask your professor for assistance.



5. To tell the IDE which port correlates with the Arduino, click on “Tools” and this time mouse over “Port”. Click on the port that the Arduino is using in the window that appears.


PhotoresistorData | Arduino 1.8.13
File Edit Sketch Tools Help
PhotoresistorD
/*
This code
https://c
Read a ph
is "dark"
Code was
The code
AnalogRea
*/
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
Board: "Arduino Duemilanove or Diecimila"
Processor: "ATmega328P"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM5
COM6

6. Success in the above steps is known if when the upload button (right arrow) is clicked an error message does not appear at the bottom of the program.

Running a Sketch (Video times 7:33-End)

1. Click on “File” in the ribbon at the top left, then mouse over “Examples”, then “01 Basics”, then click on “Blink”. A new window with the blink program in it will open. Close the other Arduino window; you won’t need it anymore.

The text seen in the new window is the code for the program (this is called a sketch), and there are some basics you can pick up from it. On the first line you will see "/*" this denotes the beginning of a comment and the "*/" at the bottom of the greyed-out text denotes the end. A comment is text written into code that the computer will not run as code. Comments are used to tell the person reading it information about the program. Comments can also be written on a single line and are denoted with "//". All other text in the window is actual code.

Functions are a core component of coding and can be thought of as similar to mathematical functions: they do something or perform a task. There is a reference guide on Arduino's website (https://www.arduino.cc/reference/en/) that you can use to figure out what a function does. After a given function there will be brackets, what is in these brackets is called the input values for the function.

The code may seem like complicated gibberish right now, but as we progress though the semester you will become much more familiar with code and how it works. Something you can do to improve your understanding right now is to read all of the code and its comments to see what each part of the code does, while using the references guide on Arduino's website to tell you what each function is doing. Useful urls: https://www.arduino.cc/en/Reference/FunctionDeclaration and the reference guide mentioned above.

2. Click the check mark button at the top left of the window. ✓ This will verify or compile the code (check for errors such as forgetting a closing bracket) which prepares it to be run.
3. Then click the right arrow next to the verify button, → this is the upload button. This will send the sketch to the Arduino and you should see a few lights blink rapidly. (This indicates that the microprocessor is accepting the code from the computer and saving it on the board.) Then after the initial rapid blinks it will start to run the code.

When the code starts to run, it will start at the first line and read downwards. It will skip over the comment, as comments are not run, and will then arrive at the "void setup" followed by the "pinMode" function that is inside the setup function. In the "void setup" function there are two parts: the "setup" part which tells the computer that what is inside of the brackets is the program setup, and "void" which tells you that it will not return a value. Inside of the setup function lies the "pinMode" function which sets up the led lights on the device as an output. After the setup it will continue reading downwards to the "loop" function. This function will cause whatever is inside the loop function to loop forever unless told to stop. Inside the loop function are 4 instructions: the first turns the LED light on (setting the voltage to "HIGH"), the second sets a delay before the next function can be read (in 1000ths of second so 1000 = 1 sec), the third turns the led off (setting the voltage to "LOW"), and the fourth sets another delay. This loop function makes the light turn on, stay on for 1 second, turn off, stay off for 1 second, and then repeat.

4. Attempt to alter the blinking pattern by changing the delays in the loop function. Do this with a few different time values to get a feel for how it affects the blinking. Change one or more of the values to 100, then run the program. Then change to 5000 and run the program again. Note what happens with each different value.

Questions to Consider

1. What do the functions "void setup", "pinMode", and "loop" do?
2. How would you make the light turn on for 1.5 sec and then turn off for 0.5 sec in a loop?
3. What do you have to do to a sketch before you can run it on the Arduino to make sure there are no errors?
4. How do you write a comment? How would you do so?