Version 2.0

ECE Department

Portland State university

ECE 211 Intro to design processes

Lab-A0

**Lab-A0: Installation & Basics**

# Required Hardware & Software

* Arduino UNO Rev 3 microcontroller board (-*or*- compatible clone)
* Your own laptop or desktop computer
* Arduino hardware drivers and Arduino Desktop IDE software
* Access to D2L

# Introduction

Welcome to the first Arduino lab! Here is your chance learn more about hardware interfacing using an Arduino-based microcontroller board. Most of the labs are straightforward but please try to understand the underlying principles of both the HW and SW operation. In other words, do not stop at just copy-pasting of the existing code.

First things first – you need to install the required software on your computer (laptop or desktop). Instructions on how to do this are posted on D2L in the “Lab 0” folder but you can also download them from the Elegoo site here <https://www.elegoo.com/download/> (it is part of the software associated with the “Elegoo UNO R3 Super Starter Kit”). Your Scrum Master will ask you to show that you have installed this software and that it runs properly. Obviously, you cannot do the actual labs without this first step!

Your tasks are:

* Follow the instructions for the given lesson.
  + You will download lesson guides (like this one) and code from D2L, construct a small circuit, and run the sample code. You will have the chance to enhance the programs later.
* Answer a few questions on a worksheet and upload it on D2L.
* Demonstrate your work to your Scrum Master (helper).

For Lab-A0, the lessons are simple and are designed just to make you familiar with the Arduino. First, we need to make sure you have installed all the necessary software and can run some basic examples, such as LEDs. In Lab-A1 we will expand into switches, a buzzer, and a servo. Lab-A2 will be more interesting and introduce temperature sensors, analog joysticks, infrared remote control, and LCD displays. Lab-A3 is optional and will deal with some advanced features.

# Arduino Hardware Overview

The kit contains an Elegoo Uno R3, which is an open-hardware clone of the popular Arduino Uno R3 microcontroller board. For the rest of this document, “Arduino” will refer to both the official Arduino Uno and the Elegoo Uno, unless otherwise noted. Official Arduino website is <https://www.arduino.cc> . Here are some of the hardware specifications of the Arduino Uno R3:

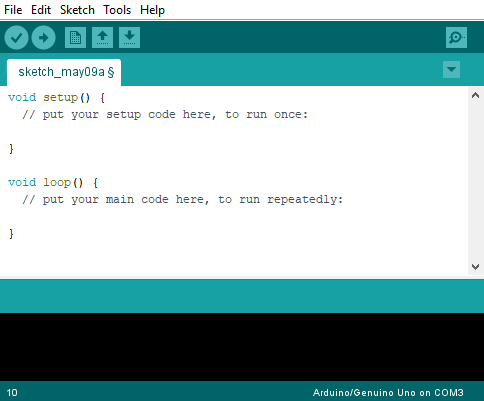
|  |  |  |  |
| --- | --- | --- | --- |
| Microcontroller: | ATmega328P | DC Current per I/O Pin: | 20 mA |
| Operating Voltage: | 5V | DC Current for 3.3V Pin: | 50 mA |
| Input Voltage: | 7-12V | Flash Memory: | 32 KB |
| Input Voltage (limit): | 6-20V | SRAM: | 2 KB |
| Digital I/O Pins: | 14 (6 provide PWM output) | EEPROM: | 1 KB |
| PWM Digital I/O Pins: | 6 | Clock Speed: | 16 MHz |
| Analog Input Pins: | 6 |  |  |

For a very brief overview of the Arduino Uno hardware, go here:

<http://www.instructables.com/id/INTRO-TO-ARDUINO-UNO-R3>

# Arduino Desktop IDE

The development environment application runs on a separate host computer and looks like the figure below. The central area is the editing window. Status messages are shown below it.



Toolbar icons:

* *check mark* : Compiles and verifies your program code.
* *right arrow* : Compiles your program code and then uploads the binary image to the Arduino.
* *up & down arrows* : Open file and save file, respectively.

Tip:

There are settings under *File → Preferences* that can add line numbers to the editing window, adjust the font size, and turn on verbose (more detailed) compiler messages.

The Arduino board must be connected to the host computer using a USB cable before you can upload compiled code. The Arduino presents itself as a serial COM port to the host. You must select the correct COM port number in the IDE via the Tools→Port menu before the Arduino and host can communicate with each other.

# Programming the Arduino

The Arduino programming language is C-based with some C++ elements. Not all standard library functions are available, since it would not make sense in the context of a microcontroller. However, it does includes a library of functions for accessing the Arduino’s hardware I/O ports. Third-party libraries can be added to make controlling certain external components easier.

Arduino projects are called “sketches” and are folders in the Documents/Arduino directory within your user account. Your program code file is stored within the sketch folder. It is a standard text file with a “.ino” suffix.

An Arduino program always contains two functions named setup() and loop(). Code to initialize the Arduino, such as setting pin modes, is placed in setup(). It is executed just once when the program runs. Your primary code goes into loop(), which is executed in an endless loop.

Note: For the lab lessons, you can create a new sketch and simply copy and paste your downloaded code to the sketch’s program file.

# Running Your Program

Once you have uploaded a program to the Arduino, it will being executing immediately. Pressing the reset button on the unit puts the Arduino back to its initial state and then begins executing again. The only way to halt it is to shut down power to the Arduino board.

* For a more detailed introduction to the Arduino Desktop IDE, visit here:
  + <https://www.arduino.cc/en/Guide/ArduinoUno#toc2>
* For a very extensive (though poorly formatted) Arduino C/C++ language reference, go here:
  + <http://www.ele.uri.edu/courses/ele205/ELE205Lab/ELE205_Lab_files/Arduino%20-%20Reference.pdf>

# Part 1: Introductory Lessons

* Download these files from D2L:

**Guide**: 1.5 Blink and Add Libraries.pdf **Code**: Blink.ino

**Guide**: 2.1 LED.pdf **Code**: N/A

**Guide**: 2.2 RGB LED.pdf **Code**: RGB\_LED.ino

Read the lessons[[1]](#footnote-1) carefully (don’t worry, they are short) and build the requested circuit (if any) on the protoboard. For all these lessons, examine the program code closely to learn how to access the various hardware features of the Arduino. For the lesson “1.5 Blink …” you do not have to do the section on libraries – we will revisit that later.

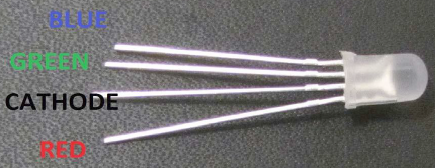
Notes:

* Lessons may give specific directory where required \*.ino or \*.pdf files can be found, e.g. “\Elegoo The Most Complete Starter Kit for UNO\English\Part 2 Module Learning\2.2 RGB LED”. This will likely not be the same as the directory where you stored your files – just load them as needed from wherever you have them stored and do not try to locate this directory.
* Have the corresponding code open in IDE while you are reading the lesson. This will, hopefully, make it easier for you to follow the explanations and you can test some ideas right away.
* You can also download the full set of lessons directly from Elegoo site

<https://www.elegoo.com/tutorial/Elegoo%20Super%20Starter%20Kit%20for%20UNO%20V2.0.2020.5.13.zip>

Be careful where you extract files from this compressed file. Files stored in it have very long names and you can run into errors complaining about that. Extract it into a root directory, e.g. D:\ and then rename the top one or two folders to have shorter names so that you can move them, or just leave it there.

* The shorter pin of a 2-pin LED is the cathode (-).
* This is the layout of the pins for the 4-pin RGB LED. The longest pin is the cathode (-):



Remember that you **must** put a 220 Ω resistor in series with each of the color (blue, green, red) leads.

This section will be posted on D2L separately as Word document. Use that file to write your report. It is included here for completeness so that you have everything in one place.

**► Exercise 1.**

Follow the instructions in the *Lesson 2 Blink*, *Lesson 3 LE*D, and *Lesson 4 RGB LED* guide and run the code to verify that the hardware and program are working.

After you are done, reflect on what you have discovered by doing this exercise. In the box below, list at least three technical things about the Arduino programming process that you have learned (or refreshed).

1. **Submit the completed worksheet (this file) on D2L and show your Scrum Master some part of what you have done.**

1. **COPYRIGHT NOTICE:** All of the lesson guides are extracted from the original Elegoo V2.0.2020.5.13 tutorial document. They were saved as individual guides to keep each lesson separate and focused. Elegoo Inc. and Simon Monk retain full copyrights on all of the lesson content. [↑](#footnote-ref-1)