

About this Course

The Arduino is an open-source computer hardware/software platform for building digital devices and interactive objects that can sense and control the physical world around them. In this class you will learn how the Arduino platform works in terms of the physical board and libraries and the IDE (integrated development environment). You will also learn about shields, which are smaller boards that plug into the main Arduino board to perform other functions such as sensing light, heat, GPS tracking, or providing a user interface display. The course will also cover programming the Arduino using C code and accessing the pins on the board via the software to control external devices.

The course is broken down for beginners with no electronics or programming experience. It starts out with hands on, how to load the code and get the on board led to blink. Once the experiment is completed the hardware is analyzed, schematics are reviewed and code is analyzed. Students can then build on their experience and modify the blinking rate of the LED light emitting diode, and get multiple LED's to blink in sequence.

By the end of the lesson the student will be exposed to reading schematics, ohms law, electronic components, resistors, led, microprocessor, how to wire circuits, and programming.

Students work in teams and learn the value of collaboration and project management.

Basics

The course starts out with the basics, just the absolute information to get started and complete the experiments. After the experiment is successfully completed you can read up and learn about the details of the components and software needed to complete the experiment. For this reason we suggest you follow the course in the sequence intended. We have provided hot links so you can go back to more basic experiments to find out what you need to know to complete the more complex experiments.

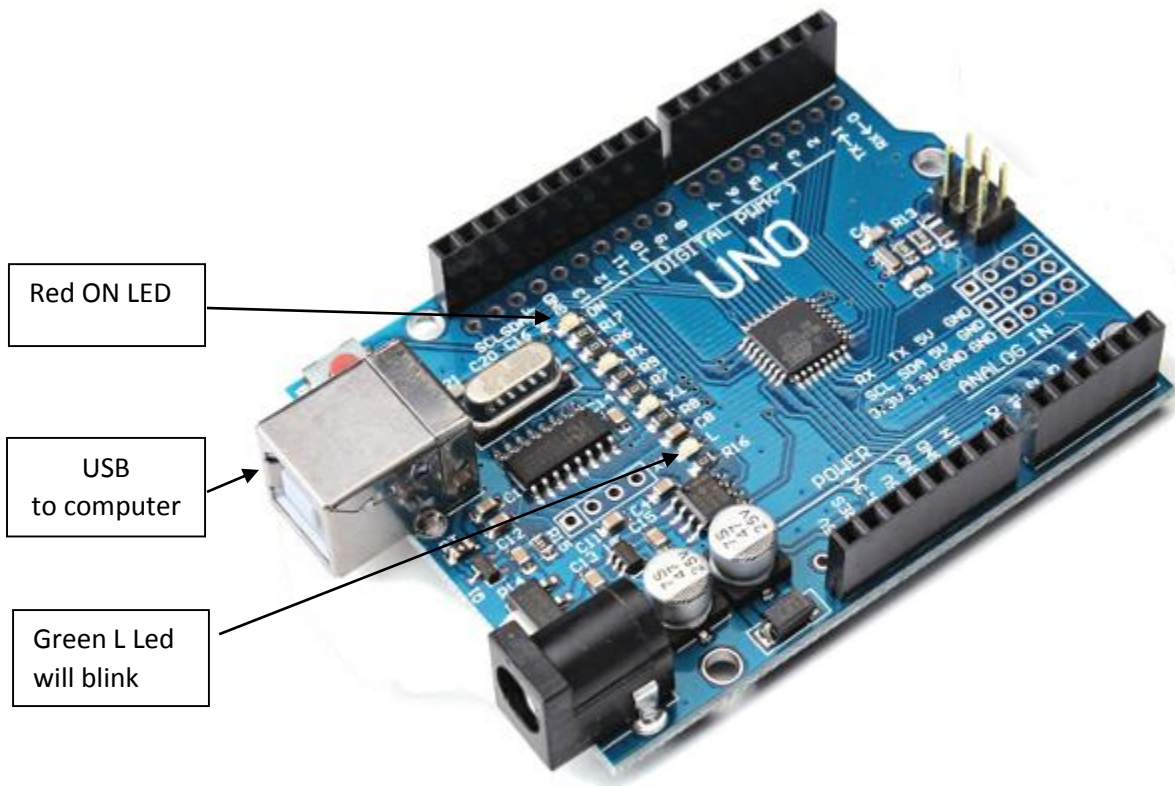
- 1) Identifying Parts in the Kit
- 2) How to use the bread board
- 3) How to find and load code to Arduino
- 4) Onboard LED Blink
- 5) External LED Blink
- 6) Four LED Sequencing
- 7) Counting binary
- 8) Morse Code LED SOS Save our Ship
- 9) Dimmable LED with photo resistor
- 10) RGB Traffic Light Red Yellow Green
- 11) Color Spectrum display RGB Led
- 12) 7 segment Display Counting 0 – 99
- 13) Speaker Sound Music
- 14) Speaker Frequency Dependent on Photo resistor
- 15) Temperature and Humidity to Terminal in Celsius
- 16) Temperature in Fahrenheit to Terminal

- 17) Time Date and Temperature to Terminal
- 18) Time 24 hour Day name Temperature Fahrenheit to Terminal
- 19) LCD Hello World
- 20) LCD Hello World Count
- 21) LCD Thermostat
- 22) LCD Clock
- 23) LCD Clock Calendar
- 24) LCD Clock Calendar Temperature
- 25) LCD Clock Calendar Temperature Fahrenheit

So let's get started Experiment 4 (On board) Led Blink

4.1 First step: let's look at the parts you will need for the first experiment.

Uno Board



USB Cable



Green Text Step you need to preform

Red Text to verify your results

STEP 1

1. Connect the USB cable to the USB port on the UNO.
2. Connect the USB cable to the USB port on your Computer.

3. The Red LED (ON) on the Arduino UNO board will illuminate.

4. The Green LED (L) on the Arduino UNO board will flash.

The blink code may already be loaded on the UNO if so the Green LED (L) will blink on an off at a one second interval if so go and load the blink code anyway.

3. Downloading the Uno code to your PC (IDE)

Integrated Development Environment: is a software application that provides comprehensive facilities to computer programmers for software development. An **IDE** normally consists of a source code editor, build automation tools and a debugger.

<https://www.arduino.cc/en/Main/Software>

Windows Installer

Windows ZIP file for non admin install

You need to load the USB drivers for the UNO

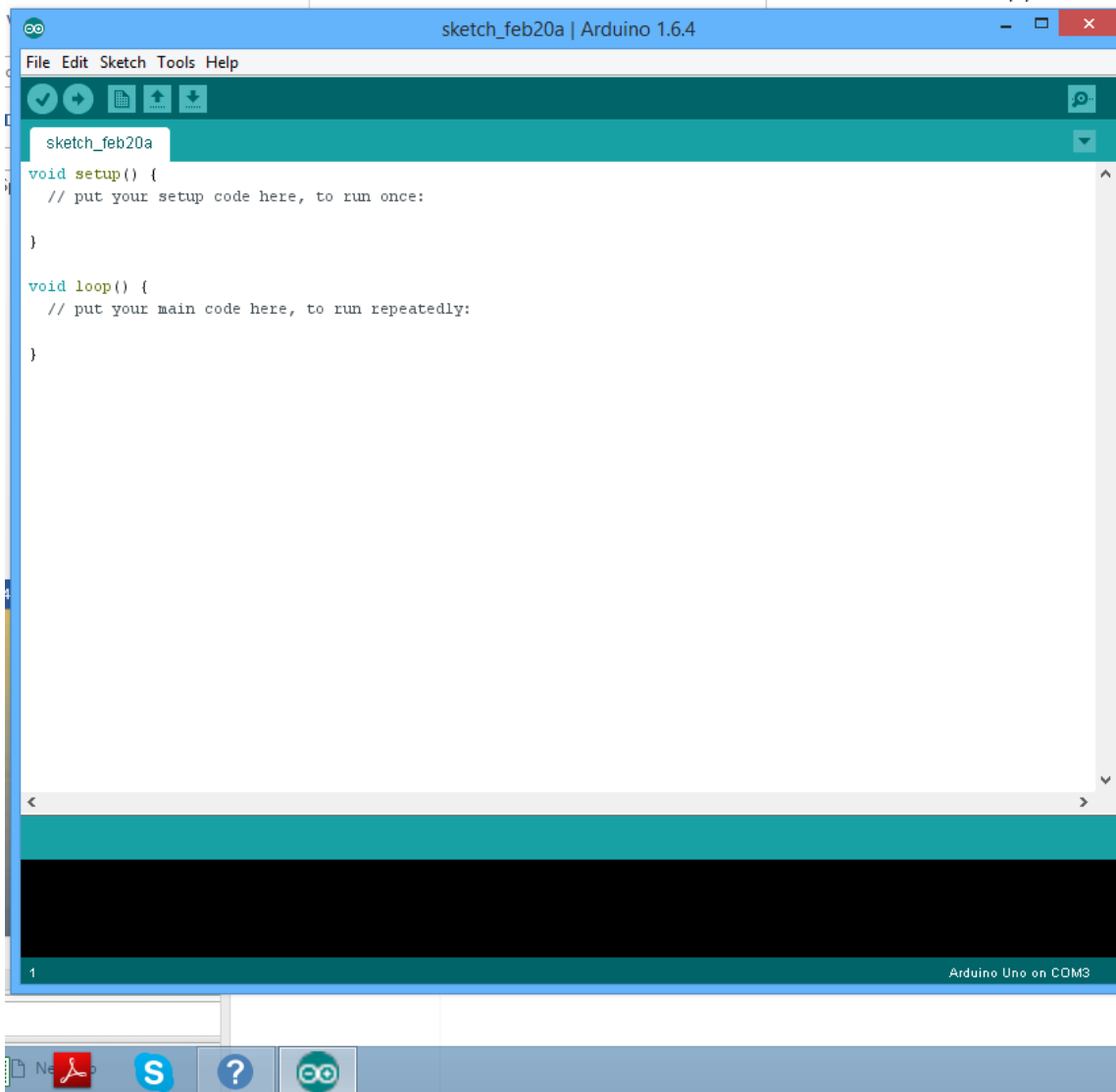
[USB Drivers CH340 / CH341](#)

You will also need to make sure you have a copy of the following include files.

Wire.h

ds3231.h

LiquidCrystal.h



4. Open the program on your PC
arduino Application
5. On the Sketch window above click on File then Open
6. Click on GoldSTEM_Lesson_4_Onboard_LED_Blink

You should see the screen below.

7) Click on the arrow under the word Edit.

You will see the word Upload appear to the right.

The program will be compiled and uploaded to the Uno.

You will see the Green LED (L) and the TX and RX LRD's flash. This will happen very quickly.

Once the program is running the green "on board" (L) LED will flash one second on one second off.

You have successfully completed your first UNO experiment

```
File Edit Sketch Tools Help
GoldSTEM_Lesson_4_Onboard_LED_Blink
/* The slash asterisk denotes the beginning of a multi-line comment field the asterisk
slash denotes the end of a multi-line comment field. The comments are ignored by the compiler
and does not add to the run time or overhead of the program.
Lesson_4_Blink GoldSTEM tm 2-21-2016
Turns on board LED (L) at pin 13 on for one second, then off for one second, repeatedly.
*/

// is also a comment field but it ends at the end of the line.
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
|
Done Saving
22 Arduino Uno on COM5
```

About the Sketch, The program run by the UNO is called a Sketch

/ The slash asterisk denotes the beginning of a multi-line comment field the asterisk slash denotes the end of a multi-line comment field. The comments are ignored by the compiler and does not add to the run time or overhead of the program.*

Blink GoldSTEM tm 2-21-2016

Turns on board LED (L) at pin 13 on for one second, then off for one second, repeatedly.

**/*

// is also a comment field but it ends at the end of the line.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

If you just wanted the actual code without the comments this is what you would have

```
void setup() {
pinMode(13, OUTPUT);
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

STEP 2

**If you notice in the code there are two lines with delays.
One is for the LED on time and the second is for the LED off time.
The number is (1000) and the comment says wait for a second.
The program uses milliseconds and 1000 ms = 1 second.**

```
delay(1000);           // wait for a second
```

Change both times to half a second.

Yes that would be (500)

You can just go and edit the numbers.

Compile and load the Sketch again by clicking on the arrow.

If your modification is successful you will see the LED blinking on and off each half a second.

Step 3

Now change both times to (10) ms and rerun the sketch.

Question 1.

The LED looks like it is on constantly?
Any Ideas why?