

Intro to Arduino

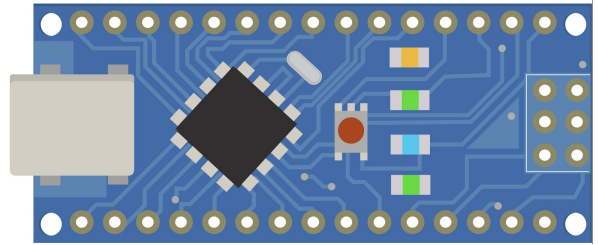
PDQ 2 - A Peek Under the IDE Hood

Grades: 5-7

Time: 5 -15 minutes

Subject: Engineering, technology and application of science

Topics: Computer Science, Electronics



Overview

This PDQ provides a quick and easy peek at some sample Arduino [Code](#). Have no fear – it’s simple and clear!

Background

Arduino is both an [Open Source hardware](#) AND [software](#) platform that enables creators, inventors, students and just about anyone to learn basic electronics and coding to make projects. The [FREE software](#) for programming your Arduino [hardware](#) is called [Arduino IDE](#). IDE stands for Integrated Development Environment. The IDE is available for computers running Windows, MacOS, and LINUX. Let’s take a look!

Objectives

Understand & Recognize:

- “Arduino” as a [hardware](#) and [software](#) platform for making projects.
- “Community” in the sense of people connected through a common interest such as making cool projects with Arduino.
- “[Input](#)” and “[output](#)” in both [hardware](#) and [software](#) based on looking at Arduino projects

What You'll Need

- All you need for this project are an Internet connected device and an Internet browser, <https://databot.us.com/introarduino/> to view the [code](#) online, or
- Make available copies of [Arduino IDE Sketch - Intro to Arduino Sketch.pdf](#).

Important Terms

Open Source: A model of sharing inventions and information for others to use, improve and share again.

Hardware: The “physical” part of a computer or device. If you can thump it on a table, it’s probably hardware.

Software: The computer program or “instructions you write” for the hardware. It’s not something you can thump on the table, but the hardware is worthless without it.

Sketch: A “sketch” in Arduino lingo refers to a computer program you’ve written in the IDE to run your Arduino hardware.

Code: The actual lines of instruction in your computer program (sketch) are code. Code is written in different languages such as Arduino. Code is also a verb – “I’m going to code for a few hours – see you later!”

Arduino IDE: IDE stands for Integrated Development Environment which is a mouthful, but it just means you have all the tools you need to write fun instructions for your Arduino microcontroller to follow.

Input: Things that go “into” a computing system are input. When you type on your phone you are “inputting” information.

Output: Something that come out of. your computing system, like the display on a smartphone, is “output.”

Prep (5 mins)

- Make available copies of [Arduino IDE Sketch - Intro to Arduino Sketch.pdf](#), or <https://datobot.us.com/introarduino/> to view the code online.

PDQ 2 (10 mins)

This image is an actual screen capture from the [Arduino IDE](#). Look for the following to help you understand the [Arduino IDE! \(Intro to Arduino Sketch.pdf\)](#)

- *What you are looking at is an Arduino sketch. It is an example of software that provides your Arduino hardware with a set of instructions to follow. The instructions are written in code. This code is written in the Arduino programming language.*
- *Notice the line numbers. These help you easily reference the line of code you are working on. For example, “Hey Jake! Take a look at line 34 of this code for me!”*

```
33 digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34 delay(1000); // wait for a second
35 digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36 delay(1000); // wait for a second
37 }
```

- Look now at line numbers 1 and 23.
You will see the symbols `/*` and `*/`.
- The text between them explains the program.

```
Blink
1 /*
2  Blink
3
```

```
22 http://www.arduino.cc/en/Tutorial/Blink
23 */
24
```

- Also look at lines 25 and 27 for `//`.

These are incredibly important and are used throughout the `code` as “comments” – this is where you explain things for yourself and others to understand. Remember, when you are working on a piece of `code`, it makes perfect sense. But a year later it can be hard to remember. A good `coder` comments well to help those who come later – be a good coding citizen!

- Note in the comment line 27 the reference to “`output`” – what is the `output` in this project?

```
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
```

What is the `input`?

- Take a special note of line 20 and “Public Domain” – this means this `code` is FREE to share and use again.

```
19
20 This example code is in the public domain.
21
```

The concept of “[open source](#)” is magical – you share what you build and other’s can build on it and share it back. Plus you get to benefit from the millions of other [coders](#) sharing solutions that might save your bacon one day!

```
36  delay(1000);           // wait for a second
37 }
```

- Here’s a simple coding challenge. Look at line 36.

This program causes an LED to “blink” on the Arduino board. Look at the comment – how would you change the command `delay(1000)` to make the LED blink faster?

Wow, you came off the blocks like a rocket!

Since you came off the blocks like a rocket in PDQ 1 and 2, are you ready for a bigger bite?

Next Step, Time to Experiment!

Educator Resources

Prep

- Read through the sample [code](#) and complete the highlighted checklist so that you are familiar with the concepts presented.
- Either print copies of the [Arduino IDE code](#), or go to: <https://databot.us.com/introarduino/> to view the [code](#)!

Objectives

Understand and Recognize:

- “Arduino” as a [hardware](#) and [software](#) platform for making projects.
- “Community” in the sense of people connected through a common interest such as making cool projects with Arduino.
- “[Input](#)” and “[output](#)” in both [hardware](#) and [software](#) based on looking at Arduino projects

Misconceptions

There are a number of misconceptions about programming and coding that you can begin to address with this simple activity.

Here are three:

- Coding is a “boring” repetitive task. Not true! Coding is like a puzzle that you need to solve. Finding the best and most efficient way to accomplish something in your [code](#) can be fun and challenging.
- Coding is only for those who are super smart at math. Not true! The most important skills required are to be methodical, logical, and persistent.
- Coding is not creative. Not true! Solving problems efficiently and elegantly requires great creativity. Think about the [software](#) you use on a regular basis for work or play. Someone had to come up with all of those ideas and features. Someone had to create the idea behind Angry Birds long before it ever became polished [software](#).

Guiding Questions

- “Looking at this example of “[code](#)” – is this what you expected a computer program to look like?”

- “Which line number marks the most interesting line of [code](#) to you?” and “What lines of [code](#) don’t you understand?”
- “It’s pretty cool to see the names of the people who wrote this program – imagine writing something useful that you could release as [Open Source](#) that millions would see and use!

Additional Resources:

9 Misconceptions People Have About Programming

<https://www.oxford-royale.co.uk/articles/9-misconceptions-programming.html>

The Official Arduino Documentation on the IDE

<https://www.arduino.cc/en/guide/environment>

Everything You Need to Know About Arduino [Code](#)

<https://www.circuito.io/blog/arduino-code/>

References:

Wikimedia Commons

Intro to Arduino by Robert O. Grover & Team databot™ is licensed under a Creative Commons Attribution 4.0 International License. Permissions beyond the scope of this license may be available at databot.us.com/contact.



One little cube. Science on the move.

+1 208-451-2281
contact@databot.us.com
databot.us.com