



Programming Arduino with Android and Windows Tablets

Created by Mike Barela



Last updated on 2014-01-03 04:00:39 PM EST

Guide Contents

Guide Contents	2
Overview	3
Windows Tablets	5
Cabling	6
Arduino Uno Driver	6
The Arduino IDE	6
Load a Program	7
Trinket and Gemma	7
Bluetooth with Windows Tablets	9
Bluetooth Use	9
Android Tablets	11
Cabling	12
The ArduinoDroid Program	12
Load a Program	12
Bluetooth	13
Going Further	15
But What About iDevices?	15
Arduino Devices Other than Uno	15
Interaction between Arduino Projects and Tablets	15

Overview



Making on the go gets easier every year. With your hacking parts bundled with a computer, you travel to class, the makerspace, or a friend's house and build your favorite circuits. Up to now, this probably meant carrying around a bulky laptop. But the weight and the value of your primary computing device probably makes that option unappealing ("if my Macbook or laptop were to break, I'd be sunk!").

Fortunately you have some great alternatives today at very reasonable price points. Android tablets with modern versions of the Android operating system can be found below \$200. In addition, there is a new crop of modern tablets powered by the Intel Bay Trail CPU and run full versions of Windows 8.1 (not the restricted Windows RT system). You can opt for the full Surface Pro system but that is nearly as bulky as an ultrabook. New 7" and 8" class tablets with high definition screens ("more pixels than retina"), make a much more portable alternative.

With a full Windows system, you have the gamut of developer tools at your fingertips. The Arduino IDE runs well and you can access code from the web, from the cloud, or locally.

With Android, you cannot directly install code from the Arduino project as Android is not a supported operating system. But there are programmers porting the Linux versions to Android.

The main method to communicate from tablet to microcontroller is the USB port. Tablets most often have microUSB ports conforming to the portable OTG (On the Go) specification which is great as they can act as slave ports for peripherals or as masters to control things. [A simple OTG cable, available in the Adafruit shop \(http://adafru.it/1099\)](http://adafru.it/1099), allows the tablet to communicate with outside devices. For an Arduino Uno, you can then use the [regular \(http://adafru.it/62\)](http://adafru.it/62) (or [6 inch \(http://adafru.it/900\)](http://adafru.it/900)) A to B cable. Other boards may require mini or micro USB cables. The OTG cable provides a standard size A socket, making mating cables straightforward.

An appealing alternative to cables is Bluetooth. For Windows, we'll cover programming an Arduino Uno wirelessly via the Adafruit [Bluefruit EZ-Link board \(http://adafru.it/1588\)](http://adafru.it/1588) and [shield \(http://adafru.it/1628\)](http://adafru.it/1628).

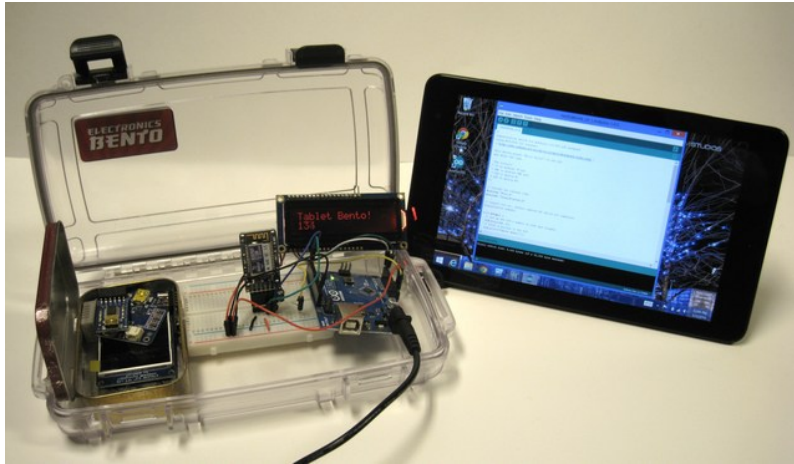
These tablets are photographed with the Arduino Uno and [LadyAda's Bento Box](http://adafru.it/765) (<http://adafru.it/765>), a portable "hack and pack" solution. Bento makes a great carry case for your projects and is about the same size as these new tablets.

If I'm a total newbie to Arduino, how do I understand Arduino basics?

Adafruit's [Learn Arduino series](http://adafru.it/d4s) (<http://adafru.it/d4s>) is the perfect place to start learning Arduino basics.

Adafruit's [Learn Arduino series](http://learn.adafruit.com/lesson-0-getting-started) (<http://adafru.it/d4s>) is the perfect place to start learning Arduino basics.

Windows Tablets



Portable microcontroller programming dates back to 1984 and the introduction of the Compaq luggable PC. Lately, it typically involves carrying a laptop to program via USB. But if you believe your laptop is too heavy (or too expensive to risk carrying around willy-nilly) there are new alternatives coming out on the market.

The Dell Venue 8 Pro is typical of a [new class of tablets](http://adafru.it/d3x) (<http://adafru.it/d3x>) from Dell, Toshiba, Lenovo, Acer and others offering a full Windows 8.1 environment in an 8 inch form factor, available in the \$200 range. This is a perfect size for portable computing and hacking. These devices include Bluetooth 4.0, 802.11n wifi, microSDXC card slot and microUSB OTG connection.

Note: This tutorial does not explain how to use the Arduino on the Windows RT operating system which is not supported software-wise by the Arduino Team. This tutorial discusses using a tablet with the full Intel-based Windows operating system.



Cabling

You will need a microUSB male OTG to A female cable [available from Adafruit](http://adafru.it/1099) (<http://adafru.it/1099>) and many mobile shops at a reasonable price. This will provide you a full A male connection to plug in a cable to fit your microcontroller. For our Uno example, you would want a standard USB A to B cable (Adafruit sells [long](http://adafru.it/62) (<http://adafru.it/62>) and easy to pack [short](http://adafru.it/900) (<http://adafru.it/900>) versions).

Arduino Uno Driver

The good news: Drivers for nearly any Arduino are available for Windows. The bad news: Windows 8.1 really, really hates installing unsigned drivers like those supplied for the Uno by the Arduino team. Some tablets may provide provision to go into recovery and allow unsigned drivers. The procedure to do that is [described here](http://adafru.it/d2W) (<http://adafru.it/d2W>). Note this may be nearly impossible with the Dell Venue 8 Pro and other tablets enabling Bitlocker to protect the main drive.

If you cannot use the method above to load the unsigned driver included with the Arduino IDE and until there are signed drivers (which should install "normally"), there is a workaround. [This blog post](http://adafru.it/d2X) (<http://adafru.it/d2X>) and the accompanying [Youtube video](http://adafru.it/d2Y) (<http://adafru.it/d2Y>) go through the steps of loading a signed driver. It installs as a Richochet Wireless USB Modem which is compatible with the Uno.

If you use an Arduino device with FTDI such as Duemilanove or compatibles (NOT UNOs!), there are signed drivers for Windows 8.1 at <http://www.ftdichip.com/Drivers/VCP.htm> (<http://adafru.it/ajv>) (check your version of Windows 32 or 64 bit in the Control Panel, System option). Yes, these are hacks but easy ones. If the Arduino team publishes a signed driver, the modem hack will no longer be needed. It is discussed in their forums at <http://forum.arduino.cc/index.php?topic=205837.0> (<http://adafru.it/d2Z>).

The Arduino IDE

You may download the Arduino software (integrated development environment or IDE) from the

arduino.cc (<http://adafru.it/aHs>) website. You can alternatively load [version 1.05 from Adafruit](#) (<http://adafru.it/aVE>) with built-in support for Flora, Gemma, and Trinket. If you have SD card storage, you may want to install the software to the card to save flash on 32GB tablets.

Once the IDE is installed, click the desktop icon to get to the Windows desktop. Double-click the Arduino icon and you will be in the development environment. Load the included Blink sketch by clicking File then Examples then 01. Basics then Blink. A new window opens with the Blink software sketch.

Ensure your OTG to Uno cable from the previous section is connected from the tablet to the Uno. The Uno should have the power LED on, powered from the tablet. Use an external power supply for the Uno to save tablet battery life. Set the environment for download by clicking Tools, then Board, then select "Arduino Uno" from the list. Go back to the Tools Menu, then Serial Port, then you usually select the last port (COM7 in my case). You can look up which COM port the Arduino uses by opening Control Panel, then "Devices and Printers" then look for the Arduino Uno. If the communications port is not listed, you can right click the icon, select Properties then click the Hardware tab to see which COM port is listed.

Load a Program

Ensure your tablet is connected to the Arduino Uno with the cables noted above. The power light should be lit on the Uno.

To upload the Blink sketch to the Uno, select the right arrow icon. The yellow LED neat the Uno's Pin 13 should now be blinking. The process may take a few seconds longer near the end as the compiled code loads to the Uno.

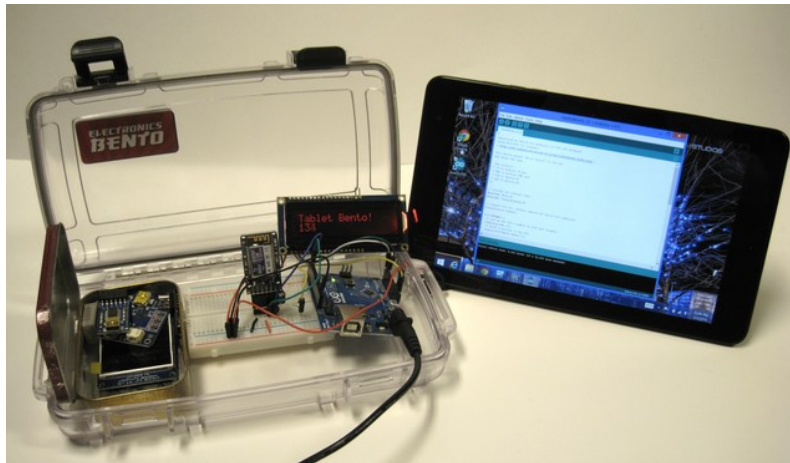
You can load your own code via a network connection, downloaded from the local flash drive or from a microSD card. When you save programs, I suggest a microSD card as the 32GB flash on the Venue 8 Pro is not a huge amount for a Windows installation (Dell includes Office which bloats the device).



Trinket and Gemma

A Windows tablet will recognize Trinket and Gemma if you have the correct OTG MicroUSB to USB and USB to mini USB cables. You must install the [USBtinyISP driver \(http://adafruit.it/cf7\)](http://adafruit.it/cf7). Just like the Uno driver, it is unsigned so Windows will probably complain and not install the driver. For Windows 8.1, that means again [rebooting into recovery \(http://adafruit.it/d2W\)](http://adafruit.it/d2W) to allow unsigned drivers. On my Dell Venue 8 Pro this is very problematic due to Dell enabling Bitlocker and no physical keyboard attached to type in prompts.

Bluetooth with Windows Tablets



Bluetooth Use

With the newly released [Bluefruit EZ-link board](http://adafru.it/1588) (<http://adafru.it/1588>) and [shield](http://adafru.it/1628) (<http://adafru.it/1628>), you can download your programs without the cable (freeing the tablet for charging).

For the Bluefruit EZ-link board (above), wire to Uno [per the tutorial here](http://adafru.it/d30) (<http://adafru.it/d30>). The shield (below) is a little easier to use since it just plugs on top, but I had a breakout handy and it only takes a few wires to connect it

The benefits of using Bluetooth is that there's no wonky driver nonsense. Bluetooth is built into every single tablet and replaces the "FTDI" or "Uno" driver requirements!



This technique can be used for Arduino Uno, Duemilanove, Mega and any other ATmega328/168/1280/2560 based Arduinos that use a USB->Serial converter. It won't work with "USB" Arduinos such as the Leonardo/Micro/Flora since they need a direct USB connection...

Pairing to EZ-Link on a tablet is just like doing so on a computer: Power up the circuit. Pair the EZ-link via [this tutorial \(http://adafru.it/d31\)](http://adafru.it/d31).

If you go to the Windows 8 desktop, the Bluetooth icon is in the notification area, lower right (you might have to tap the little up arrow to see all the icons, the Bluetooth icon is a blue oval with a B on it). Select **Add a Bluetooth Device** from the menu. You may need to press the Bluefruit "Pair" button on the circuit board. Your tablet should detect something is pairing and load the driver for you (warning - this can take a minute). If your Bluefruit is listed but will not connect, delete the current entry, press the pair button, and it should rediscover the device and allow pairing.

To make sure you have the right COM port, right-click (or press and hold with touch) the Windows icon in Windows 8.1 lower left corner of the screen. Select **Control Panel**. In **Control Panel** select **Devices and Printers**. You should see an Adafruit Bluefruit icon near the bottom of the icon list per the [Bluefruit tutorial \(http://adafru.it/d32\)](http://adafru.it/d32). Double click the Adafruit EZ-Link icon then click the Hardware tab. It should say: **Standard Serial over Bluetooth link (COM3)** - it might have a different COM number, no problem, just note it.

In the Arduino IDE, go to the Tools Menu, then Serial Port, then select the COM port you just discovered which belongs to Bluefruit.

If your circuit is built correctly, you can now remotely send a program over Bluetooth to your Uno. The capacitor from Bluefruit DTR pin to the Uno Reset pin is required to have the sketch auto-reboot when loaded. If you do not have one, just push the Uno RESET button after the sketch loads.

Note the load process takes longer wirelessly than with USB. It will work, it just takes 60-90 seconds.

Android Tablets



The Arduino team does not publish native Android versions of the Arduino IDE. Fortunately Anton Smirnov has come to the rescue with his program ArduinoDroid. It is available from the Google Play app store (free ad-supported, ad-free is \$2.99). Load the program and you will have an icon in your all apps listing.

For this part of the tutorial, I will be using a Google/Asus Nexus 7 (2013 version) which has a 7 inch screen and weighs 0.64 pounds. It has dual band wifi 802.11 a/b/g/n, Bluetooth 4.0, and near field (NFC). The base model has 16GB flash and 2GB RAM. It has a microUSB port with OTG (on the go) master and slave capability. There are a great variety of Android devices using various versions of the Android operating system. Other devices might not be compatible with OTG USB ports or the ArduinoDroid program.

Note this was tested with a Nexus 7 2013 tablet. There are many Android devices on the market so other devices may lack the capability to connect via USB or run ArduinoDroid.



Cabling

You will need a microUSB male OTG to A female cable [available from Adafruit \(http://adafru.it/1099\)](http://adafru.it/1099) and mobile shops at a reasonable price. This will provide you a full A male connection to plug in a cable to fit your microcontroller. For our Uno example, a standard USB A to B cable (Adafruit sells [long \(http://adafru.it/62\)](http://adafru.it/62) and easy to pack [short \(http://adafru.it/900\)](http://adafru.it/900) versions).

You may want to power the Uno from a separate power supply - if you use USB power, it will shorten your tablet battery life depending on the current draw and could exceed the power that can be drawn if you connect shields or other devices.

The ArduinoDroid Program

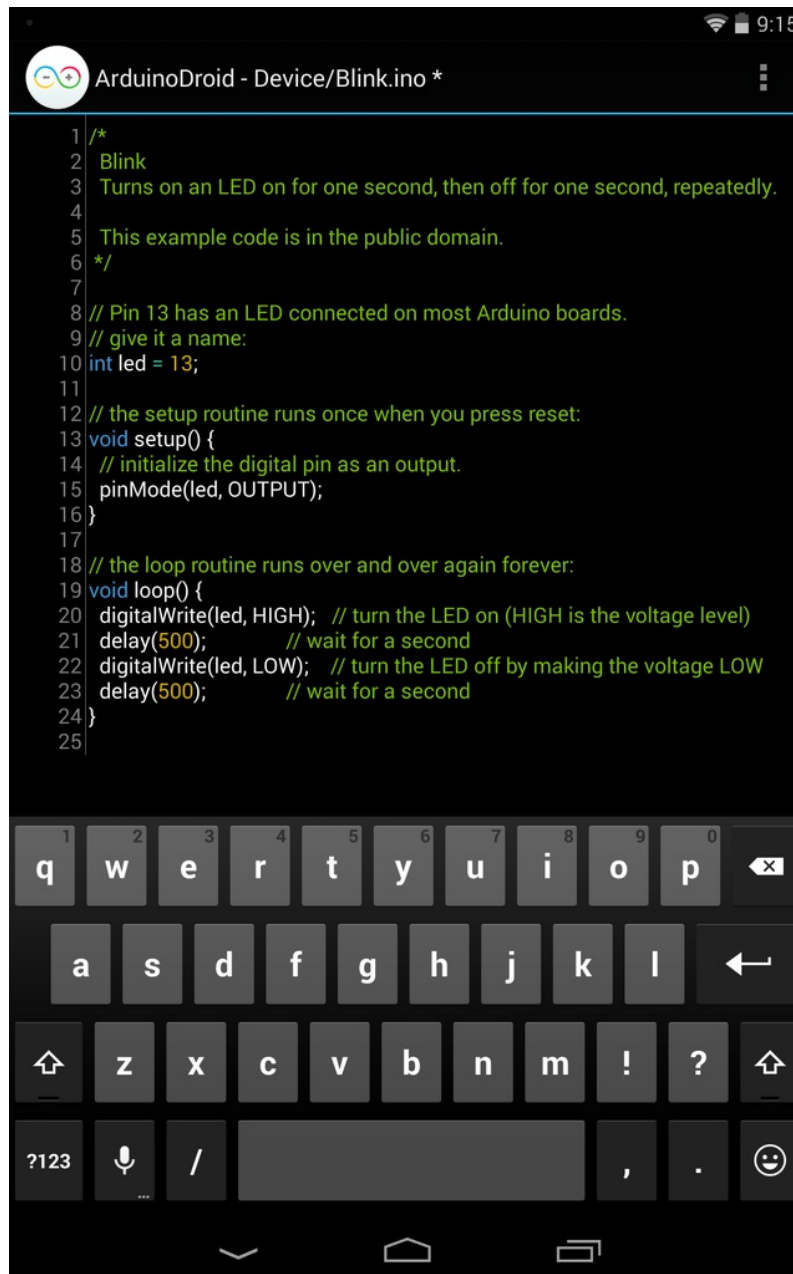
The program does not look like the typical Arduino IDE. It is a blank canvas with line numbering. If you turn the tablet in portrait mode, you can have the IDE and keyboard visible at the same time. You use the ... icon in the upper right to select what to do: load a file, or other actions you would do with the Arduino IDE menu bar. If you navigate to Sketch then Examples, you can go to 01. Basics and select the Blink example sketch.

Load a Program

You can download your own code via USB (like a flash drive), via Google Drive to a local folder or by cut and paste from a browser. The program is also designed to work with Dropbox (an additional fee may apply). You can select Sketch, then Open and you can select a recent sketch from the device or from Dropbox. Likewise Sketch -> Save as will save to the tablet local folders or to Dropbox.

The text is color-coded and is edited like any text file.

You can upload your sketch to the Uno by getting the menu using the ... icon, then Actions then you can Verify/Compile. Once it compiles cleanly use the same menu to select upload. I initially had an upload error but the code loaded. Subsequent uploads worked fine.

A screenshot of the ArduinoDroid application on an Android device. The top status bar shows a Wi-Fi icon and the time 9:15. The app's title bar reads "ArduinoDroid - Device/Blink.ino *". The main area displays a C++ sketch for a blinking LED. The code is as follows:

```
1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  This example code is in the public domain.
6  */
7
8  // Pin 13 has an LED connected on most Arduino boards.
9  // give it a name:
10 int led = 13;
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14   // initialize the digital pin as an output.
15   pinMode(led, OUTPUT);
16 }
17
18 // the loop routine runs over and over again forever:
19 void loop() {
20   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
21   delay(500);              // wait for a second
22   digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
23   delay(500);              // wait for a second
24 }
25
```

At the bottom of the screen is a virtual QWERTY keyboard with a microphone icon and a smiley face icon. The Android navigation bar is visible at the very bottom.

Bluetooth

As of this tutorial's release, ArduinoDroid lacks a method of sending the program via Bluetooth although the author is considering it.

However, once you have your Arduino programmed, there are a wide variety of programs that can initiate Bluetooth communication with an Arduino connected to an Adafruit Bluefruit EZ-Link or Shield. Look in the Google Play app store, search for the term "arduino".

Going Further

It is anticipated that portable computing will expand in coming years with new technologies from Intel and manufacturers leveraging low cost, powerful ARM devices.

Sub-\$200 portable computing is an exciting development. Certainly these methods will work on many larger devices also.

But What About iDevices?

Apple devices are harder to interface due to the lack of standard USB ports. MakerShed markets the [RedPark TTL iOS Cable \(http://adafru.it/d34\)](http://adafru.it/d34) to interface Apple products with the older dock connector. Apple Bluetooth lacks the SPP protocol required to interface with serial devices like Adafruit Bluefruit.

Arduino Devices Other than Uno

Communicating between Android and Arduino Mega, Leonardo, Micro, and Duemilanove should be straightforward and are listed in the ArduinoDroid board type list. The developer is active so more devices may be added but it would appear to be a 1.05 IDE fork so support for Arduino Due, Yun, and Galileo are probably not expected until a major revision

For Windows tablets, most boards should work but it could take some experimentation.

Programming Adafruit's Trinket and Gemma are currently beyond Android. For Windows tablets, installation of the Adafruit tweaked 1.05 IDE works well. You must also be able to install the USBtinyISP driver. On Windows 8.1, the unsigned driver will probably be very problematic.

Interaction between Arduino Projects and Tablets

There are programs for many operating systems that allow interaction between an Arduino project and a computer via Wifi, Bluetooth, or USB. That discussion would be lengthy enough for a separate tutorial.

To get you started, there are Android programs in the Google Play store if you search for "Arduino".

On a simpler level, nearly all operating systems can open a terminal or command prompt. If the device can open a serial stream over the communications channel, you may be interacting with your Arduino project.