



Connecting and Programming Your Digispark

Bootloader:

The bootloader is the code that is pre-programmed on your Digispark and allows it to act as a USB device so that it can be programmed by the Arduino IDE.

The Digispark runs the “micronucleus tiny85” bootloader version 1.02, an open source project: <https://github.com/micronucleus/micronucleus> [https://github.com/micronucleus/micronucleus] originally written by Bluebie: <https://github.com/Bluebie> [https://github.com/Bluebie].

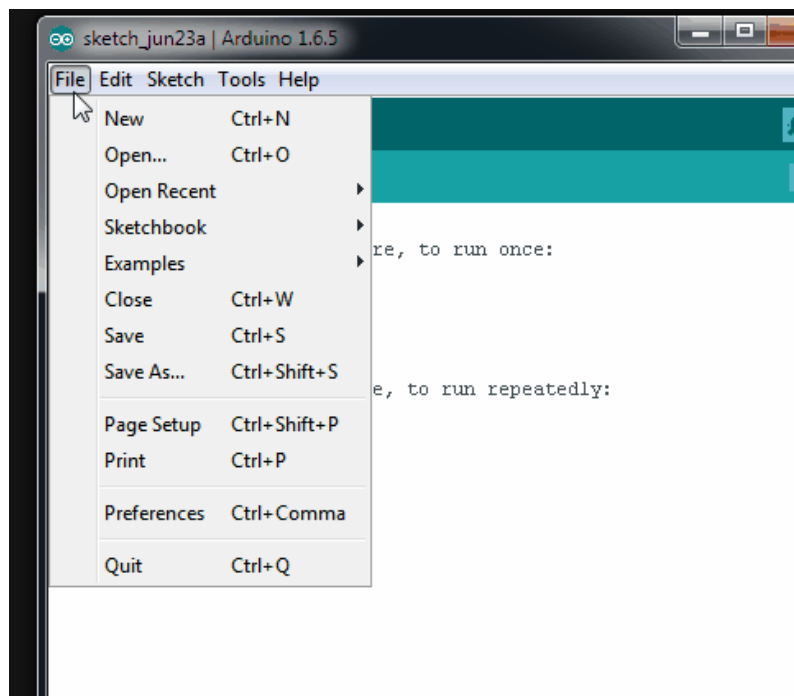
CAUTION We are not at this point supporting upgrading the firmware, so if you decide to, you do so entirely at your own risk.

Software:

The Digispark uses the Arduino IDE 1.6.5+ (Arduino 1.6.5r2 - NOT 1.6.6 or 1.6.7 strongly recommended)

Installation Instructions:

- **First download the appropriate Arduino package at the Arduino.cc website:** <https://www.arduino.cc/en/Main/Software> [https://www.arduino.cc/en/Main/Software]
- If using Arduino 1.6.6 or higher and windows - you will need to download and install the drivers manually. Download, unzip and run “Install Drivers” (on 32bit systems) or “DPIInst64” (on 64bit systems). If you get stuck, try following the steps [shown in this YouTube video](https://www.youtube.com/watch?v=MmDBvgrYGZs) [https://www.youtube.com/watch?v=MmDBvgrYGZs]. The driver files are located here: <https://github.com/digistump/DigistumpArduino/releases/download/1.6.7/Digistump.Drivers.zip> [https://github.com/digistump/DigistumpArduino/releases/download/1.6.7/Digistump.Drivers.zip]
- **Install or Unzip the Arduino application.**
- **Run the Arduino application.**
- **In the Arduino application go to the “File” menu and select “Preferences”**

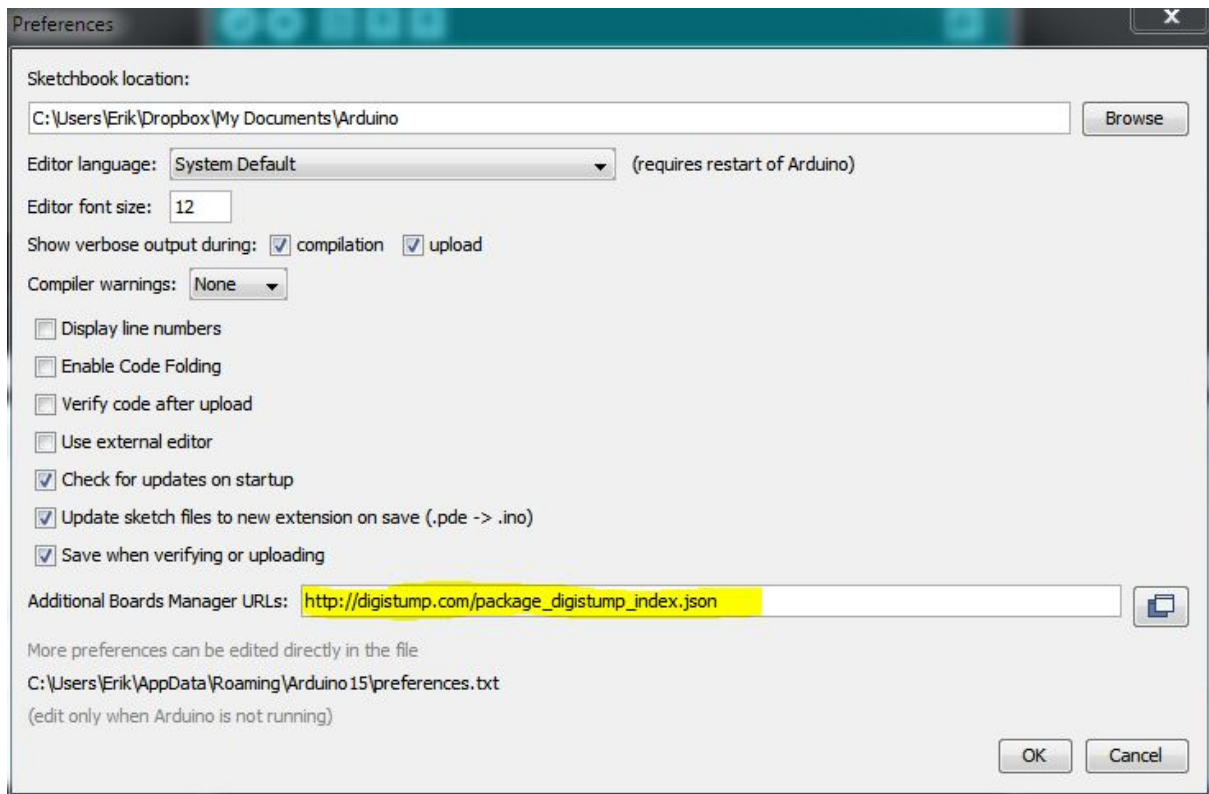


- **In the box labeled “Additional Boards Manager URLs” enter:**

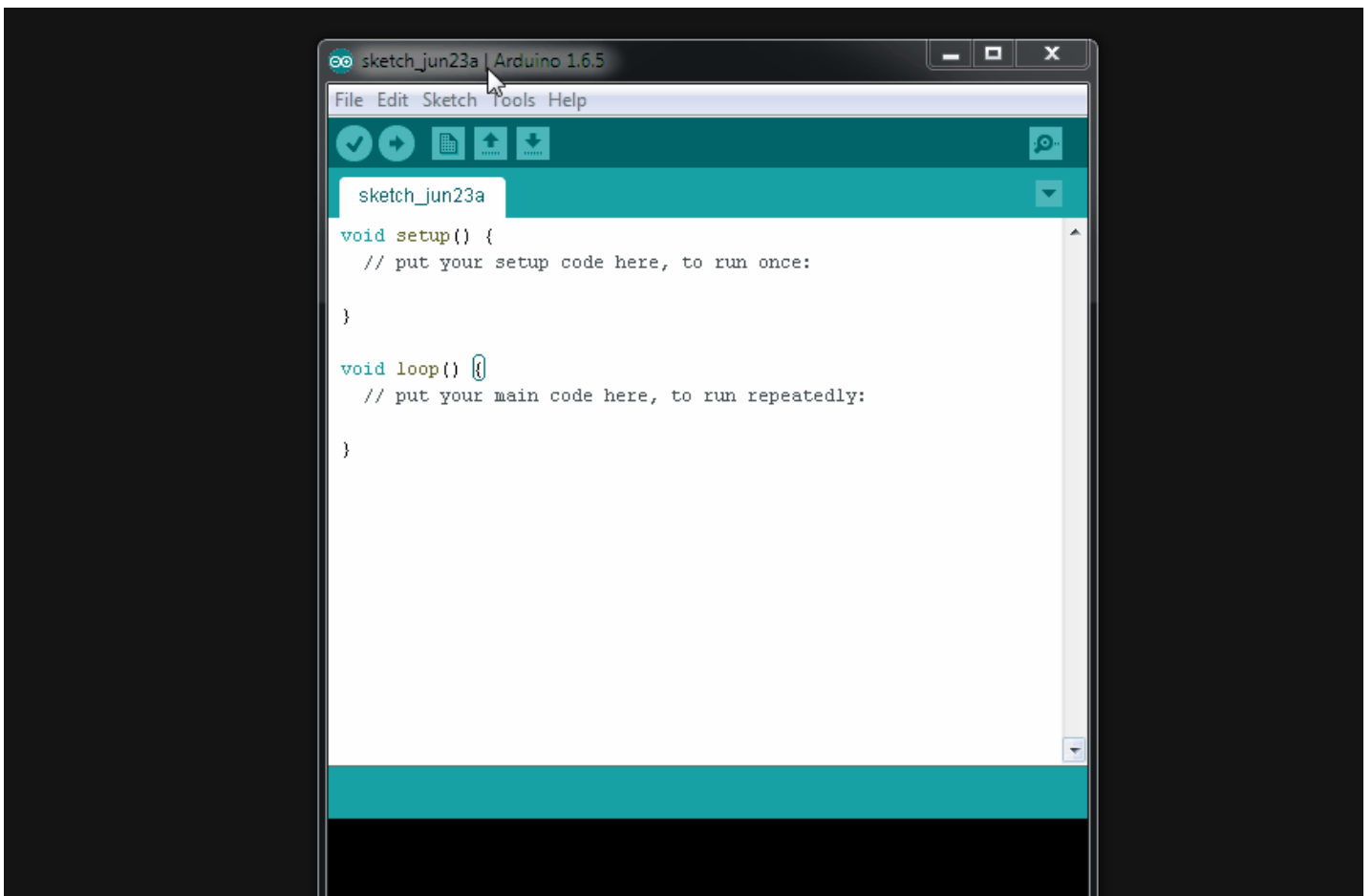
`http://digistump.com/package_digistump_index.json`

and click OK

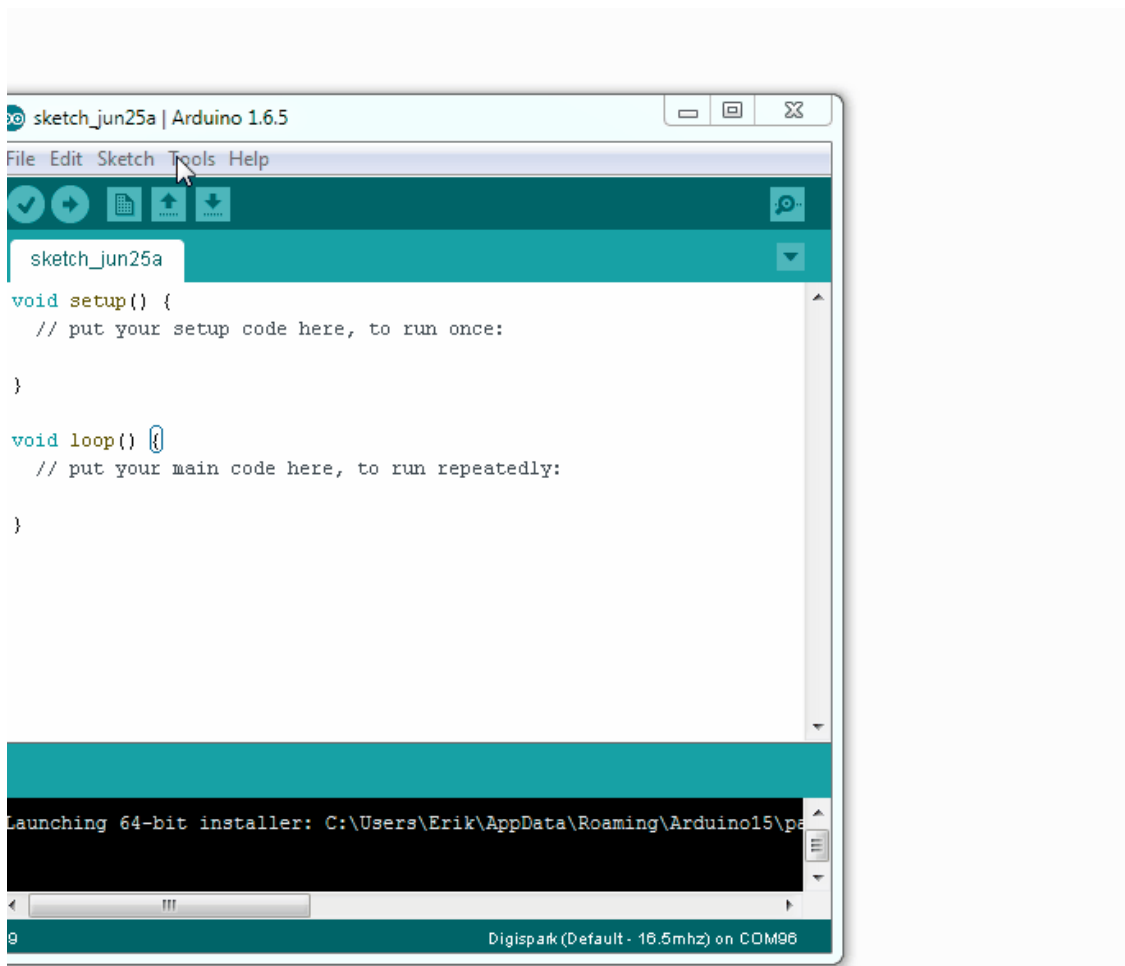
Note: If you already have additional URLs entered in that box, then click the button on the right of the box and enter this URL on a new line.



- Go to the “Tools” menu and then the “Board” submenu - select “Boards Manager” and then from the type drop down select “Contributed”:
- Select the “Digistump AVR Boards” package and click the “Install” button.



- You'll see the download progress on the bottom bar of the “Boards Manager” window, when complete it will show “Installed” next to that item on the list.
- WINDOWS USERS: When complete the install with pop up a Driver Install Wizard window, please click “Next” on this Window to install the drivers for Digistump Boards (If you already have them installed, this installer will update them and install any that are missing)
- With the install complete, close the “Boards Manager” window and select the Digispark from the Tools – Boards menu. “Digispark (Default - 16.5mhz)” is the board that should be selected by all new users.



- **The install is now complete!** (Linux users see note below)

Linux Install

- If you haven't before - Install the udev rules found here: [Troubleshooting section \[http://digistump.com/wiki/digispark/tutorials/linuxtroubleshooting\]](http://digistump.com/wiki/digispark/tutorials/linuxtroubleshooting)
- **ATTENTION LINUX USERS: PLEASE CHECKOUT THE LINUX TROUBLESHOOTING PAGE****

Sources

Digistump Package Sources: <https://github.com/digistump/DigistumpArduino> [<https://github.com/digistump/DigistumpArduino>]

Using the Digispark with the Arduino IDE:

The Digispark works a bit differently than some Arduino compatible products. The Digispark programs with a different procedure.

From the Tools menu select Board → Digispark (Default - 16.5Mhz)

(The Tools → Programmer selection does not matter)

Write some code, open your code, or open a Digispark example.

You do not need to plug in your Digispark before invoking upload

Hit the upload button. The bottom status box will now ask you to plug in your Digispark - at this point you need to plug it in - or unplug and replug it.

You'll see the upload progress and then it will immediately run your code on the Digispark.

If you unplug the Digispark and plug it back in or attach it to another power source there will be a delay of 5 seconds before the code you programmed will run. This 5 second delay is the Digispark Pro checking to see if you are trying to program it.

Your first upload:

If you're just getting started try the above procedure with the example you can find by going to File → Examples → Digispark → Start

Here is the code it will load:

```
// the setup routine runs once when you press reset:  
void setup() {
```

```

// initialize the digital pin as an output.
pinMode(0, OUTPUT); //LED on Model B
pinMode(1, OUTPUT); //LED on Model A
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(0, HIGH); // turn the LED on (HIGH is the voltage level)
  digitalWrite(1, HIGH);
  delay(1000);           // wait for a second
  digitalWrite(0, LOW); // turn the LED off by making the voltage LOW
  digitalWrite(1, LOW);
  delay(1000);           // wait for a second
}

```

This code will blink the BUILTIN LED (on either pin 0 or pin 1 depending on the Digispark model see:[Model Identification](#)).

As soon as it has uploaded you should see the LED start to blink!

For some sample code for the basic I/O function see here: [Digispark Basics](#)

Digispark Differences and Using Arduino/Processing with the Digispark:

The Arduino Environment introduction is a great start if you are new to the Arduino IDE: <http://arduino.cc/en/Guide/Environment> [<http://arduino.cc/en/Guide/Environment>]

The Digispark supports all features found in the IDE with the exception of the serial monitor and the burn bootloader functionality.

Many existing libraries will not work with the Digispark: For I2C devices check out the TinyWireM library, which makes it super simple to port an I2C based device library over to use with the Digispark.

Pin outs:

- All pins can be used as Digital I/O
- Pin 0 → I2C SDA, PWM (LED on Model B)
- Pin 1 → PWM (LED on Model A)
- Pin 2 → I2C SCK, Analog In
- Pin 3 → Analog In (also used for USB+ when USB is in use)
- Pin 4 → PWM, Analog (also used for USB- when USB is in use)
- Pin 5 → Analog In

For a handy pin reference flip over the Digispark - pin capabilities are listed on the back

For some sample code for the basic i/o function see here: [Digispark Basics](#)

Precautions:

The Digispark, due to its small size and low cost is not as robust as a full blown Arduino.

When testing a new circuit we recommend that you test it with an external power supply first. Connecting a shorted circuit to the Digispark and connecting it to your computer could damage your computer and/or its USB ports. We take no responsibility for damage to your machine as a result of the use of a Digispark.

We strongly recommend connecting your Digispark through a USB hub which will often limit the damage caused by a short circuit to the usb hub. For the record, we've found many computers have usb fuses built in, and when we blew them on our 27" Mac monitor, thankfully they reset and everything worked after a power down.

The Digispark does not have short circuit or reverse polarity protection. Connecting power to the Digispark power pins backwards will almost certainly destroy it.

The Digispark is small enough to present a choking hazard and small enough to be inserted into some sockets. We take no responsibility for misuse of the product. Please treat electricity and electronics with respect and common sense.

Troubleshooting:

The Digispark pushes affordable, small components to their limits by using software to emulate USB functions generally served by a dedicated chip. For this reason much of the Digispark's USB operation is considered borderline out of spec. Certain laptops, older machines, and just some cheaply made USB hubs/controllers do not like this. We tested and designed the Digispark for maximum compatibility but we can not guarantee it will work with everything.

In addition remember the Digispark only shows up as a programmable device for 5 seconds, after that it will start running its code (when it is new and un-programmed this means it will blink) and disappear or act like the USB device you programmed it to act like.

Always remember to unplug the Digispark before hitting upload and plug it in when the Arduino IDE requests you to. If you get an error that assertion failed or micronucleus crashed during upload then you probably did not unplug your Digispark before uploading.

If the computer will not recognize the Digispark try the following:

- Try connecting it to another USB port or system.
- Try connecting it to the rear ports (if a desktop)
- Try connecting it to a USB hub.
- Try a powered USB hub.
- Try a different cable.

If it won't upload:

Did you set your board AND programmer to Digispark in the Arduino IDE?

WINDOWS:

- **Did you install the right driver?**
- Try following the steps shown in this video [<https://www.youtube.com/watch?v=MmDBvgrYGZs>], as it addresses most of the problems encountered during the driver installation on Windows, including needing to run a different installer on 64 bit versions of windows, and how to finish installing the drivers without having to plug and unplug your digispark multiple times.
- For the example programs if you are getting "Device not found" even after installing the DigiUSB driver in the examples folder then try this: <http://digistump.com/board/index.php/topic,257.0.html> [<http://digistump.com/board/index.php/topic,257.0.html>]
- If Windows does not recognize a "DigiUSB" but a generic "HID-compliant device"/"USB Input Device" pair, go to the Device Manager, find in the "Human Interface Devices" category the USB Input Device with a vendor ID of 0x16C0 and select "update driver" / "browse my computer" / "let me pick from a list" and the DigiUSB should show up there.

MAC

- If the Arduino application works fine but the computer does not recognize the Digispark after you are asked to plug it in - try a usb extension cable or hub, some MacBook's ports are recessed too far for the data connectors to make contact.
- **If the Arduino application won't start (especially an error like: "Arduino" is damaged and can't be opened.) or the upload fails try allowing all unsigned code to be run: See the "How to change Gatekeeper settings to allow or disallow unidentified apps" section on this page:** <http://www.imore.com/how-open-apps-unidentified-developer-os-x-mountain-lion> [<http://www.imore.com/how-open-apps-unidentified-developer-os-x-mountain-lion>]
- If you're getting the error above and allowing unidentified apps doesn't fix it try this: <http://digistump.com/board/index.php/topic,119.0.html> [<http://digistump.com/board/index.php/topic,119.0.html>] - we are actively working on a fix for this issue.
- **2012 MBP** The USB ports on the 2012 MBPs (Retina and non) cause issues due to their USB3 controllers, currently the best work around is to use a cheap USB hub (non USB3) - we are working on future solutions. The hub on a Cinema display will work as well.

LINUX: See the [Linux troubleshooting page](#)

If your upload fails in the middle of it, you may also be using a bad USB cord and or a bad hub.

Support requests are best posted to the forum, so everyone can benefit and we can help each other through any issues - <http://digistump.com/board> [<http://digistump.com/board>]