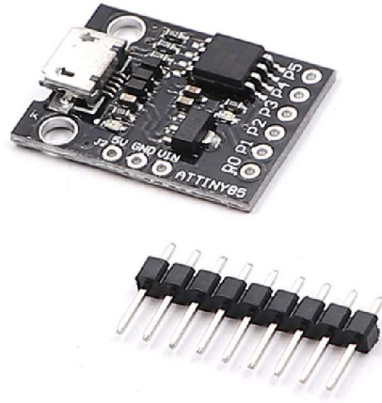


Development Board ATTINY85 for Arduino USB use Micro Model:Digispark Kickstarter User Manual



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 tiny[^]” bootloader version 1.0.2, an open source project:<https://github.com/micronucleus/micronucleus> originally written by Bluebie: <https://github.com/Bluebie>.

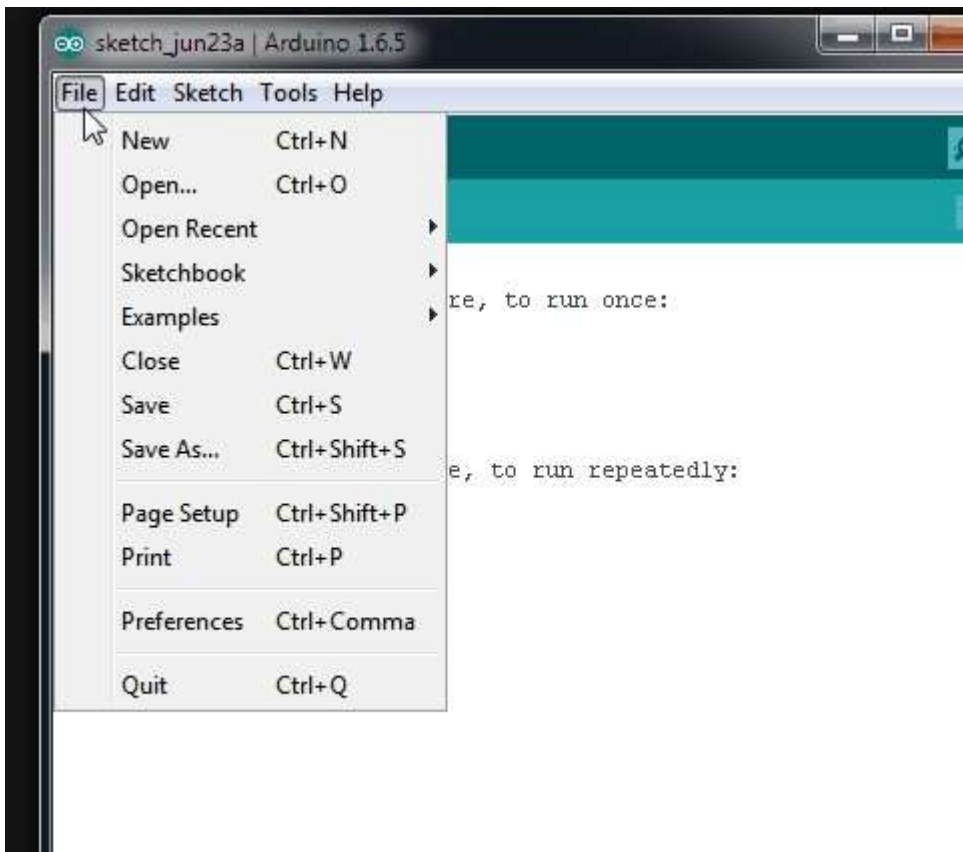
CAUTION We are not at this point supporting upgrading the firmware, so if you decide to, you do so entirely at your own risk.

Software:

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

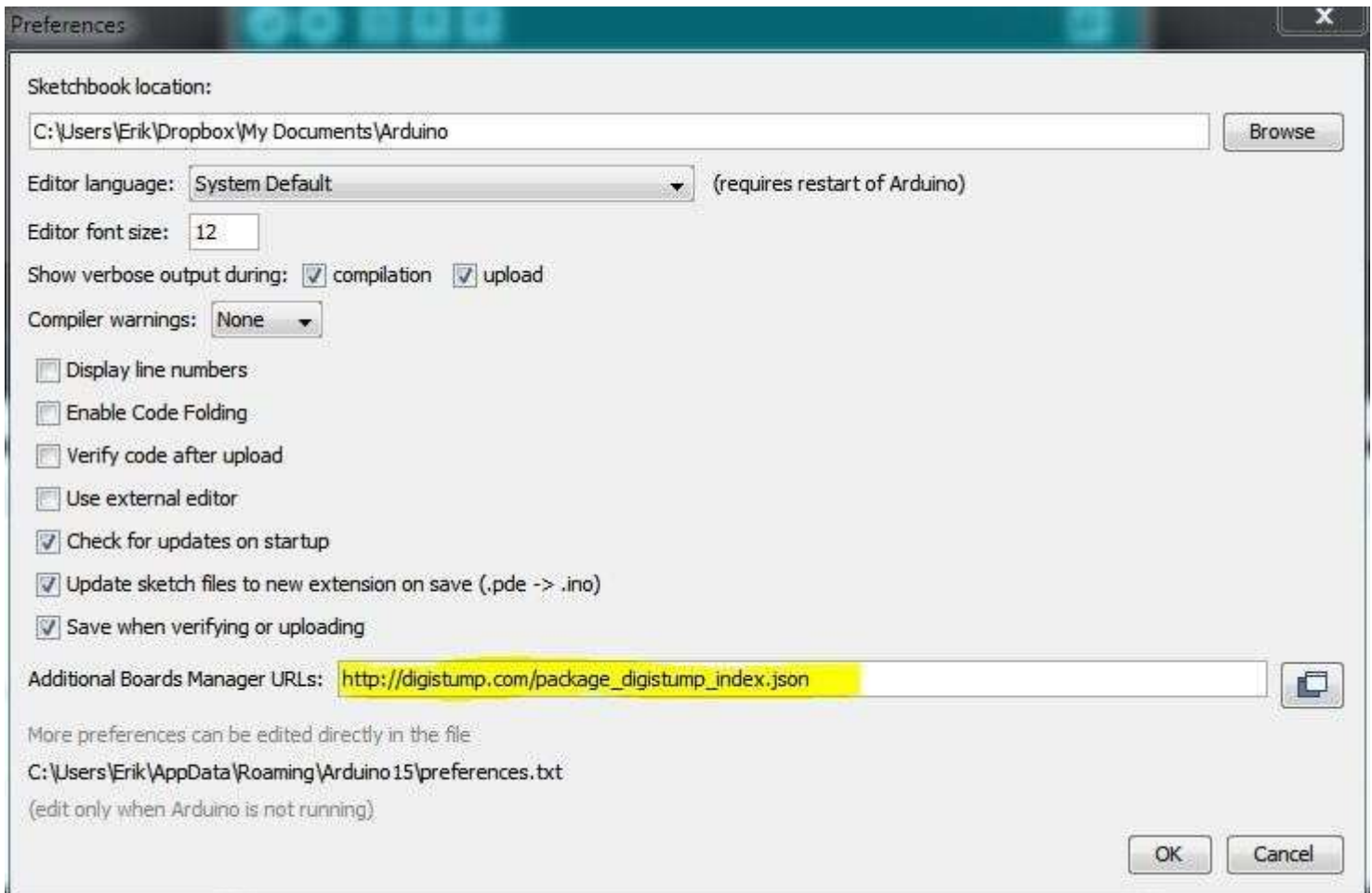
Installation Instructions:

- **First download the appropriate Arduino package at the Arduino.cc website:**
<https://www.arduino.cc/en/Main/Software>
- If using Arduino 1.6.6 or higher and windows - download, unzip, and run “Install Drivers” from here:<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”**

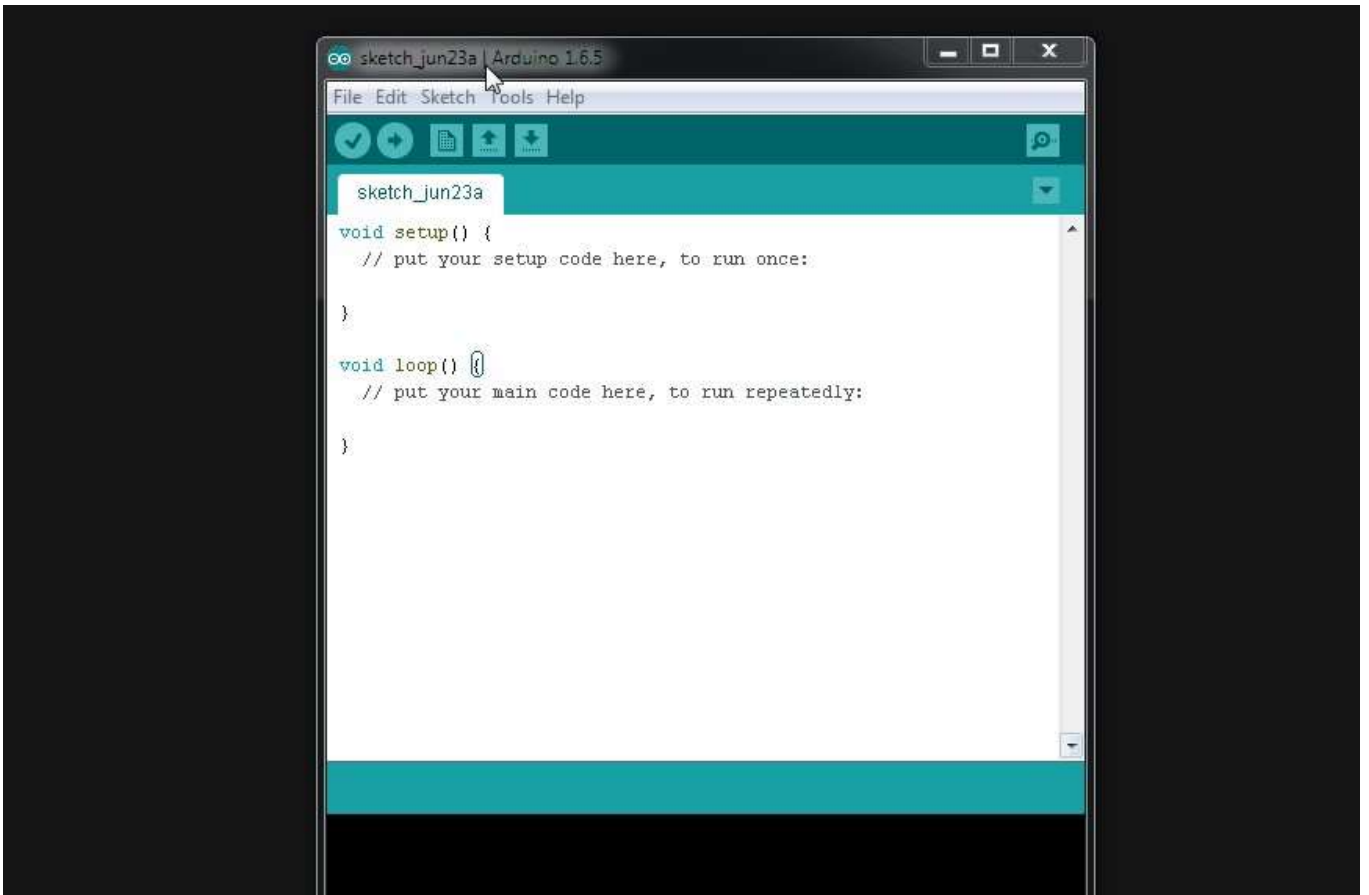


- In the box labeled “Additional Boards Manager URLs” enter:

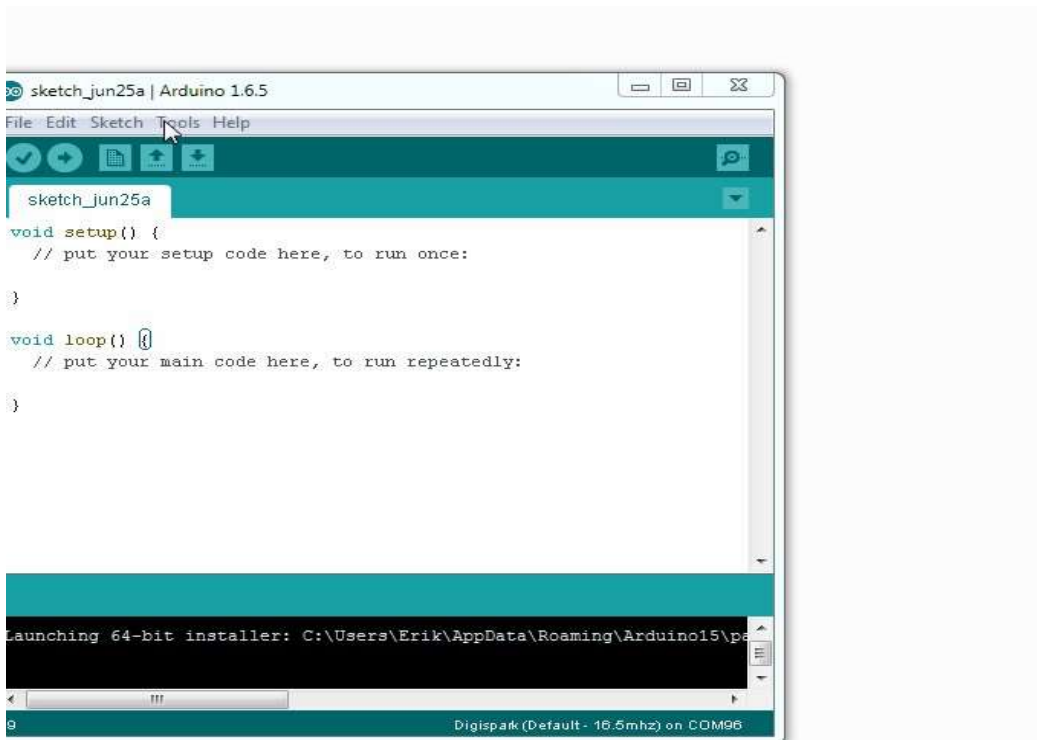
http://digistump.com/package_digistump_index.json



- 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 will 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](#)

- **ATTENTION LINUX USERS:** [PLEASE CHECKOUT THE LINUX TROUBLESHOOTING PAGE**](#) Sources

Digistump Package Sources: <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 - 16,0Mhz)

(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 0 seconds before the code you programmed will run. This 0 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

}
```