

Using the Arduino Pro Mini 3.3V a [learn.sparkfun.com tutorial](http://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-3-3v)

Available online at: <http://sfe.io/t104>

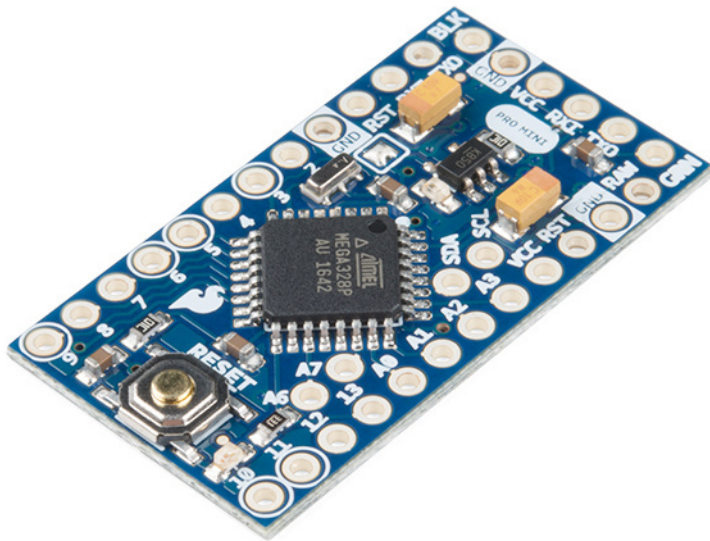
Contents

- [Introduction](#)
- [What It Is \(and Isn't\)](#)
- [Assembly](#)
- [Powering](#)
- [Programming](#)
- [Resources and Going Further](#)

Introduction

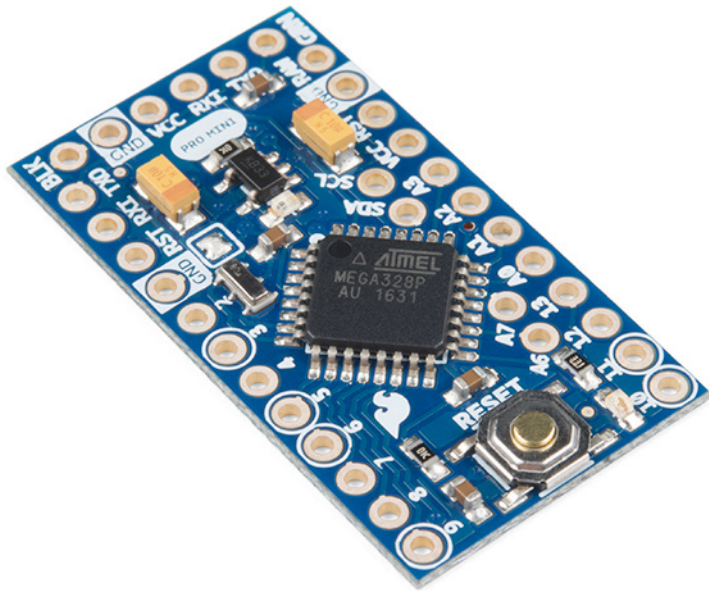
Heads up! This tutorial was written originally written for the Arduino Pro Mini 3.3V/8MHz. However, you can still use this as a guide to upload code for Arduino Pro Mini 5V/16MHz. To upload adjust the Processor in the Arduino IDE when uploading code to the 5V version board.

The original, true-blue [Arduino](#) is open-source hardware, which means anyone is free to download the design files and spin their own version of the popular development board. SparkFun has jumped on this opportunity and created all sorts of Arduino variants, each with their own unique features, dimensions, and applications. Now one of those variants has landed in your hands; congratulations! It's a wild world out there in microcontroller-land, and you're about to take your first step away from the wonderful -- though sometimes stifling -- simplicity of the Arduino Pro Mini. There are two variants, the 5V/16MHz and the 3.3V/8MHz.



[Arduino Pro Mini 328 - 5V/16MHz](#)

DEV-11113
\$11.25



[Arduino Pro Mini 328 - 3.3V/8MHz](#)

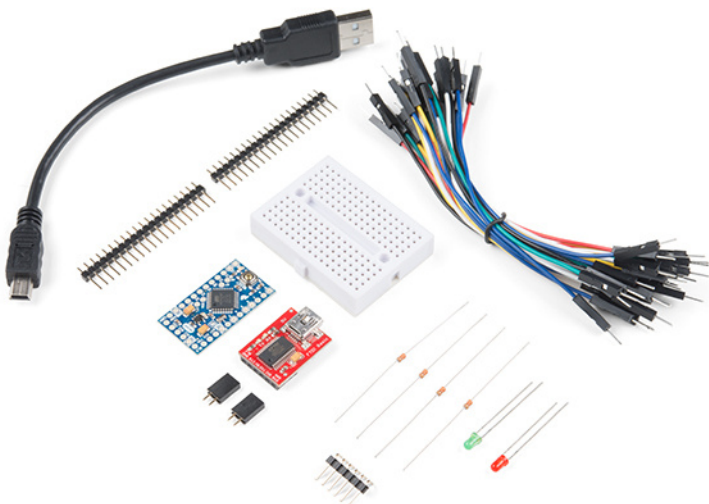
DEV-11114
\$11.25

For the scope of this tutorial, we'll go over how to set up and use the [3.3V Arduino Pro Mini](#), everything from assembling the tiny Arduino to programming it.

Required Materials

To follow along, you'll need a few extra items. You may not need everything though depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary. At a minimum you will need, some headers, USB cable, and FTDI. The FTDI Basic will be used to program (and power) the Pro Mini. The headers are optional, but they're our preferred way to interface other devices to the Pro Mini.

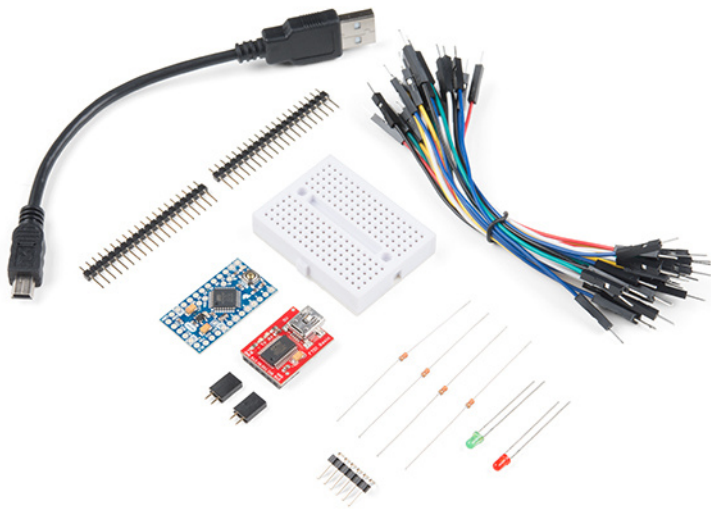
Starter Kits: If you are looking for additional components to prototype, check out some of the related starter kits.



[SparkFun Arduino Pro Mini Starter Kit - 5V/16MHz](#)

KIT-15254

Retired



[SparkFun Arduino Pro Mini Starter Kit - 3.3V/8MHz](#)

KIT-15257

Retired

Tools

Assembly of the Pro Mini also requires [soldering](#). This is a great place to start soldering, if you've never done it before! The joints are all easy, through-hole jobs. So grab a [soldering iron](#), some [solder](#) and [general soldering accessories](#).



[Soldering Iron - 60W \(Adjustable Temperature\)](#)

TOL-14456

\$25.95



[Solder Lead Free - 15-gram Tube](#)

TOL-09163

\$4.95

Suggested Reading

This project tutorial builds on a few more conceptual tutorials. If you're not familiar with the subjects below, consider reading through their respective tutorials first:

[How to Solder: Through-Hole Soldering](#)

This tutorial covers everything you need to know about through-hole soldering.

[Serial Communication](#)

Asynchronous serial communication concepts: packets, signal levels, baud rates, UARTs and more!

[What is an Arduino?](#)

What is this 'Arduino' thing anyway? This tutorial dives into what an Arduino is and along with Arduino projects and widgets.

[Logic Levels](#)

Learn the difference between 3.3V and 5V devices and logic levels.

[How to Install FTDI Drivers](#)

How to install drivers for the FTDI Basic on Windows, Mac OS X, and Linux.

[SparkFun USB to Serial UART Boards Hookup Guide](#)

How to use the SparkFun FTDI based boards to program an Arduino and access another serial device over the hardware serial port, without unplugging anything!

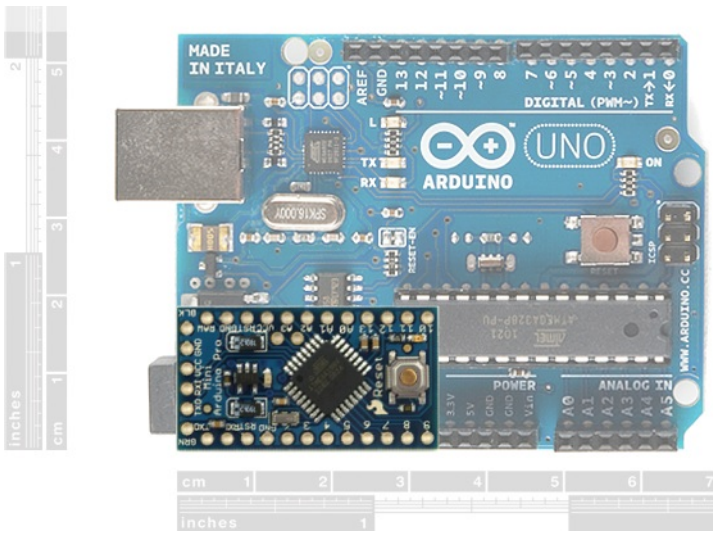
[How to Work with Jumper Pads and PCB Traces](#)

Handling PCB jumper pads and traces is an essential skill. Learn how to cut a PCB trace, add a solder jumper between pads to reroute connections, and repair a trace with the green wire method if a trace is damaged.

Note: Besides the FTDI that is used in this tutorial, there are other USB-to-serial converters (e.g. [CH340](#), CP210X, CY7C65213, etc.) that can be used to upload code to the Arduino Pro Mini. Make sure to check out the board and install the respective driver for your USB-to-serial converter.

What It Is (and Isn't)

So what differentiates the Arduino Pro Mini from the Arduino Uno? Well, the most obvious difference is the form factor. The Pro Mini's pretty...mini, measuring in at just 1.3x0.70". It's about 1/6th the size of the Arduino Uno. The compact size is great for projects where you may need to fit the Arduino into a tiny enclosure, but it also means that the Pro Mini is **not physically compatible with Arduino shields** (you could still hard-wire the Mini up to any Arduino shield).



Comparing the size of a standard Arduino Uno with the (aptly named) Pro Mini.

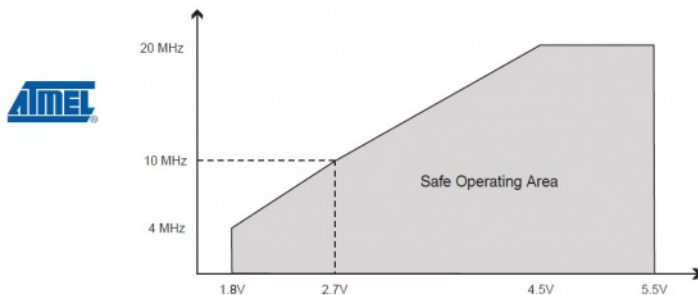
The Mini packs almost as much microprocessor-punch as the regular Arduino, but there are a few major hardware changes you should be aware of before you start adapting your project to the Mini. The first glaring hardware difference is the voltage that the Mini operates at: **3.3V**. Unlike the Arduino Uno, which has both a 5V and 3.3V regulator on board, the Mini only has one regulator. This means that if you've got peripherals that only work at 5V, you might have to do some [level shifting](#) before you hook it up to the Pro Mini (or you could go for the [5V variant](#) of the Pro Mini).

Another major variation from the standard Arduino lies in the speed at which the ATmega328 runs. The Pro Mini 3.3V runs at **8MHz**, half the speed of an Arduino Uno. We put a slower resonator on the Mini to guarantee safe operation of the ATmega. That said, don't let the slower speed scare you away from using the Mini; 8MHz is still plenty fast, and the Mini will still be capable of controlling almost any project the Arduino Uno can.

Speed Grades

Maximum frequency is dependent on V_{CC} . As shown in Figure 28-1, the Maximum Frequency vs. V_{CC} curve is linear between $1.8V < V_{CC} < 2.7V$ and between $2.7V < V_{CC} < 4.5V$.

Figure 28-1. Maximum Frequency vs. V_{CC}



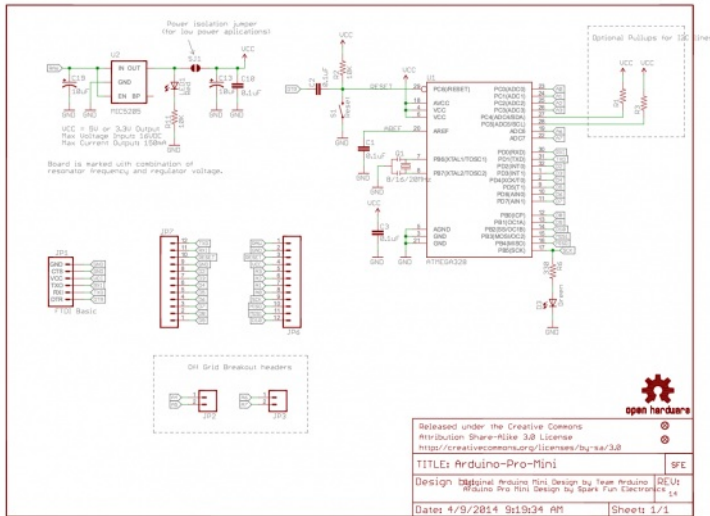
One last missing piece of hardware is the Atmega16U2-based USB-to-Serial converter, and the USB connector that goes with it. All of the USB circuitry had to be eliminated for us to make the Pro Mini as small as possible. The absence of this circuit means an external component, the [FTDI Basic Breakout](#) or any USB-to-serial converter, is required to upload code to the Arduino Pro Mini.



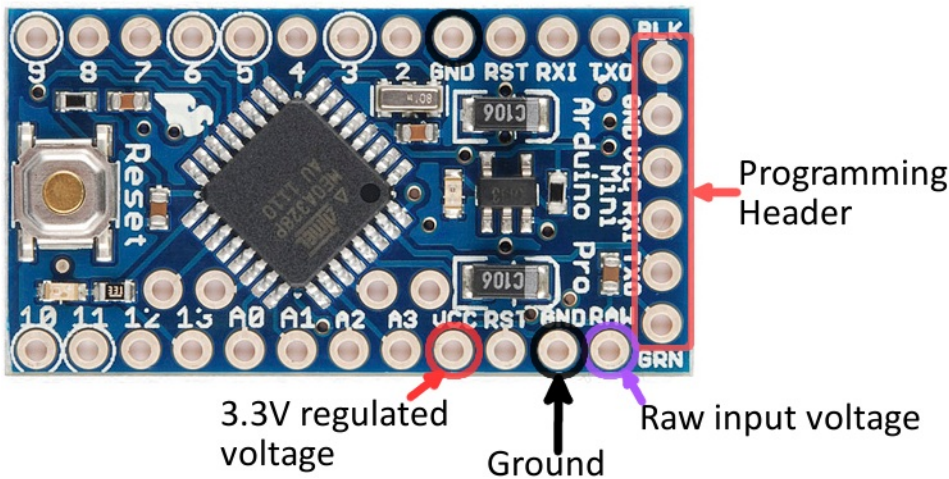
Note: Besides the FTDI that is used in this tutorial, there are other USB-to-serial converters (e.g. [CH340](#), CP210X, CY7C65213, etc.) that can be used to upload code to the Arduino Pro Mini. Make sure to check out the board and install the respective driver for your USB-to-serial converter.

Schematic and Pin-out

The schematic of the Pro Mini can be broken down into three blocks: the voltage regulator, the ATmega328 and supporting circuitry, and the headers.

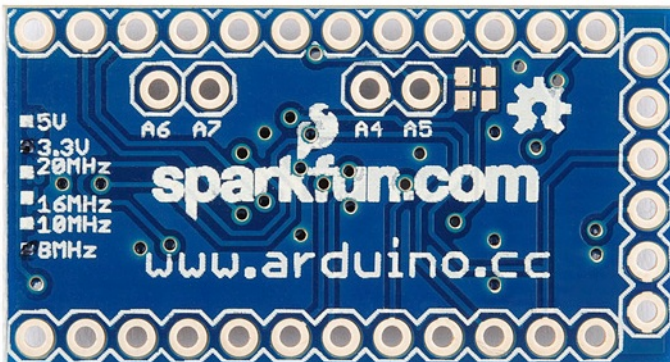


The Pro Mini's pins surround three of the four sides. The pins on the short side are used for programming, they match up to the FTDI Basic Breakout. The pins on the other two sides are an assortment of power and GPIO pins (just like the standard Arduino).



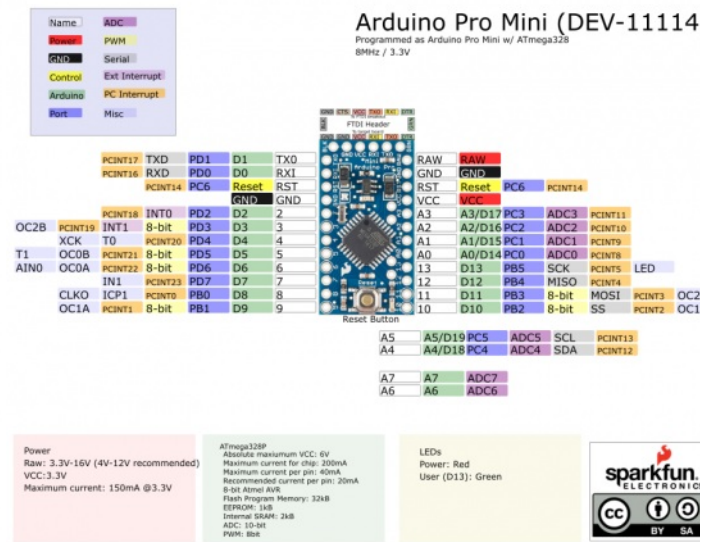
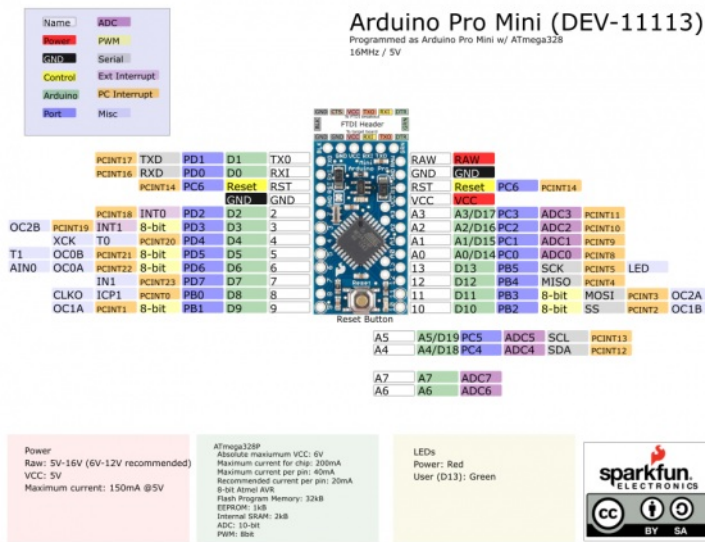
There are three different power-related pins: GND, VCC, and RAW. GND, obviously, is the common/ground/0V reference. RAW is the input voltage that runs into the regulator. The voltage at this input can be anywhere from 3.4 to 12V. The voltage at VCC is supplied directly to the Pro Mini, so any voltage applied to that pin should already be regulated to 3.3V.

Four pins are actually not located on the edge of the board: A4, A5, A6 and A7. Each of these analog pins is labeled on the back side of the board.



A4 and A5's location may be very important if you plan on using I²C with the Pro Mini -- those are the hardware SDA and SCL pins.

More information about pins can be found in our graphical datasheets. The pinout listed on both datasheets are the same. The only difference is the voltage and frequency that the board operates at.



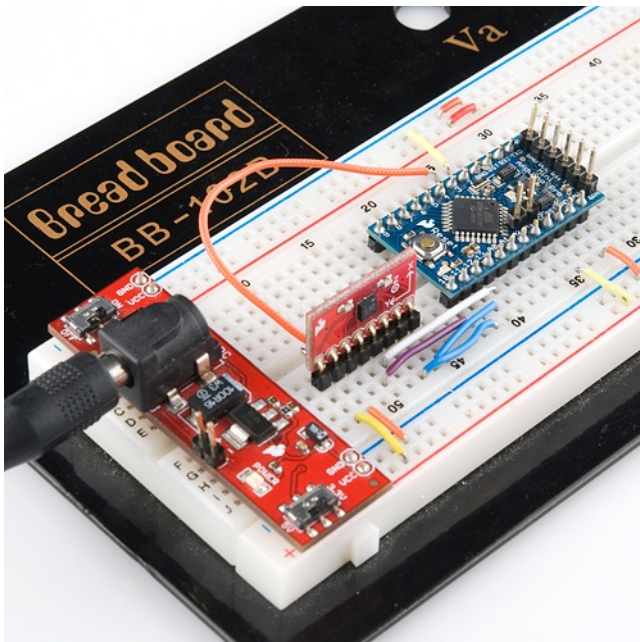
Graphical Datasheet for the 5V/16Mhz Version

Graphical Datasheet for the 3.3V/8MHz Version

Assembly

The Arduino Pro Mini doesn't look like much when you first get it; it's as bare-bones as can be. We've left it up to you to solder headers or wires into the open through-holes. There are a few things to make you aware of though.

First, decide how you want to connect the FTDI Basic Breakout to the Pro Mini's **programming header**. The programming header is a row of six pins on the side of the board, labeled "BLK", "GND", "VCC", "RXI", "TXO", and "GRN". Since the FTDI Basic board is equipped with a female header, it's usually best to equip your Mini's programming header with mating male headers, either [straight](#) or [right-angle](#).

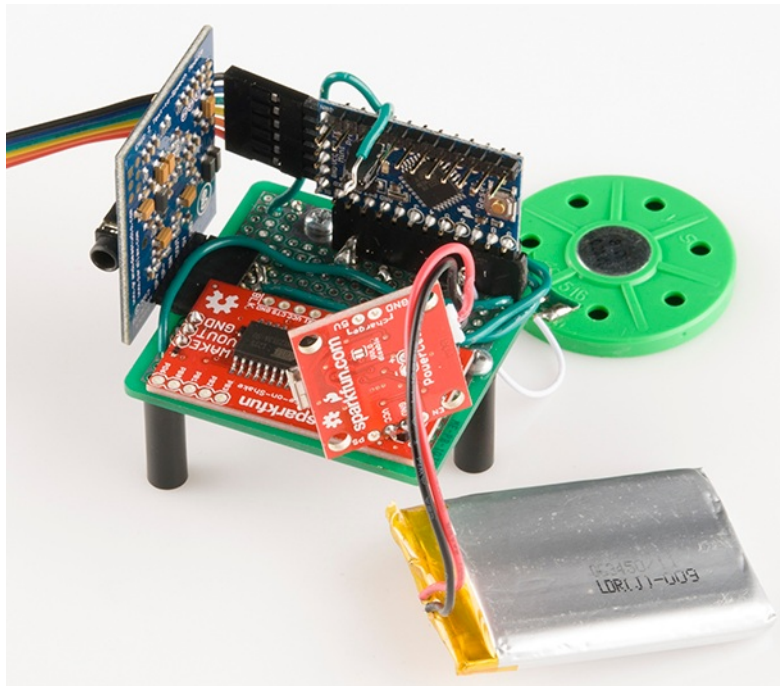


This Pro Mini had male headers soldered into all pins, so it could slot directly into a breadboard. Notice the programming header pins are soldered "upside-

down", to keep them accessible.

Warning: Make sure to match the [logic levels](#) for the USB-to-Serial Converter and the Arduino Pro Mini that you are using. For FTDI's, there can be an option to adjust the logic levels to either 3.3V or 5V by [cutting a trace and adding a solder jumper](#).

The remaining assembly choices are up to you. There are many options; you could solder in male headers to make it breadboard-compatible [female headers](#) to make it compatible with [jumper wires](#), or just solder [stranded-wire](#) straight into the pins.



The Arduino Pro Mini in the [Uncertain Z-Cube Project](#) used a combination of right and straight male headers.

Versatility is what makes this board so great, and you can assemble it in whatever way makes the most sense for your project.

Powering

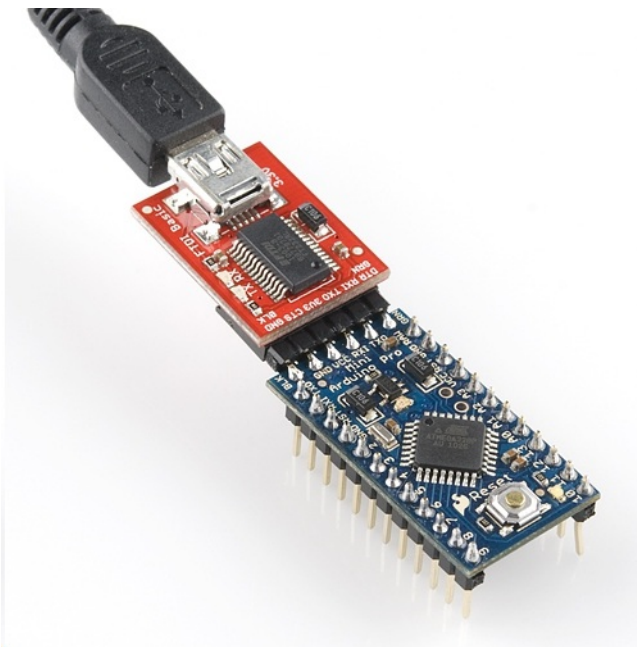
The most important factor in any project is [what's going to power it](#). The Pro Mini doesn't have a barrel jack, or any other obvious way to connect a power supply, so how do you power the thing?

Pick a power source that suits your project. If you want something that matches the compactness of the Pro Mini, a battery - [LiPo](#), [alkaline](#), [coin cell](#), etc. -- may be a good choice. Or you could use a [wall power supply](#) along with a [barrel jack adapter](#).

If you have a supply that's **greater than 3.3V** (but less than 12V), you'll want to connect that to the **RAW** pin on the Mini. This pin is akin to the VIN pin, or even the barrel jack, on the Arduino Uno. The voltage applied here is regulated to 3.3V before it gets to the processor.

If you already have a **regulated 3.3V source** from somewhere else in your project, you can connect that directly to the VCC pin. This will bypass the regulator and directly power the ATmega328. Don't forget to connect the grounds (GND) too!

There is a third power option that's only usually available while you're programming the Pro Mini. The FTDI Basic Breakout can be used to power the Mini via your computer's USB port. Keep in mind that this option may not be available when your project has entered the wild, absent from any computers or USB supplies.



That leads us to the next section...programming the Arduino Pro Mini.

Heads up! This tutorial was written originally for the Arduino Pro Mini 3.3V/8MHz. If you are using an Arduino Pro Mini 5V/16MHz, you will be using a 5V FTDI. To upload adjust the Processor in the Arduino IDE when uploading code to the 5V version board.

Warning: Make sure to match the [logic levels](#) for the USB-to-Serial Converter and the Arduino Pro Mini that you are using. For FTDI's, there can be an option to adjust the logic levels to either 3.3V or 5V by [cutting a trace and adding a solder jumper](#).

Programming

Arduino IDE

If you've never used an Arduino before (how bold of you to go straight for the Mini!), you'll need to download the [Arduino IDE](#). Check out our tutorial on [installing Arduino](#) for help on that subject.

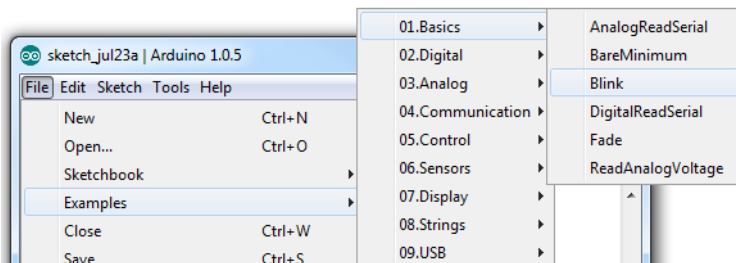
Drivers

If this is the first time that you have plugged the FTDI Basic Breakout into your computer, you may need to install drivers for it. Check out our [installing FTDI Drivers tutorial](#) for help there as suggested earlier in the tutorial.

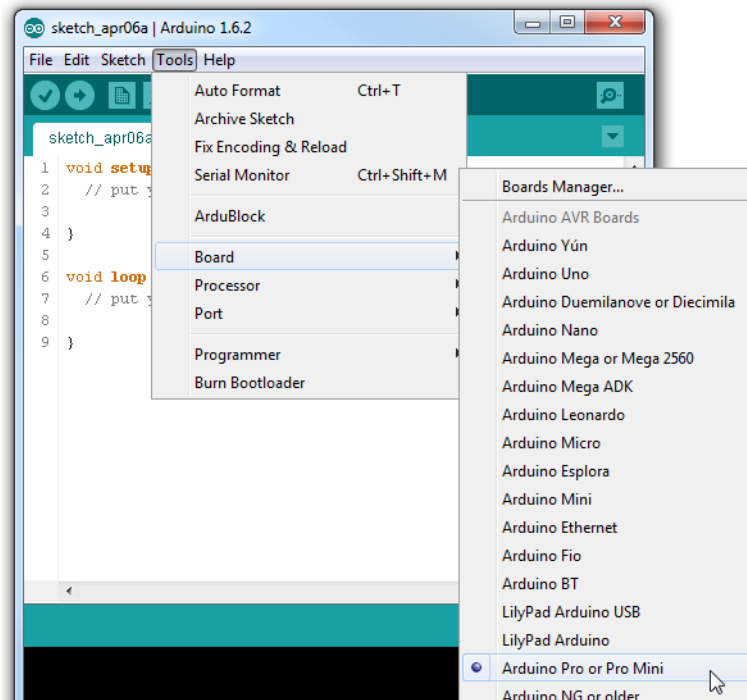
Note: Besides the FTDI that is used in this tutorial, there are other USB-to-serial converters (e.g. [CH340](#), CP210X, CY7C65213, etc.) that can be used to upload code to the Arduino Pro Mini. Make sure to check out the board and install the respective driver for your USB-to-serial converter.

Uploading Code

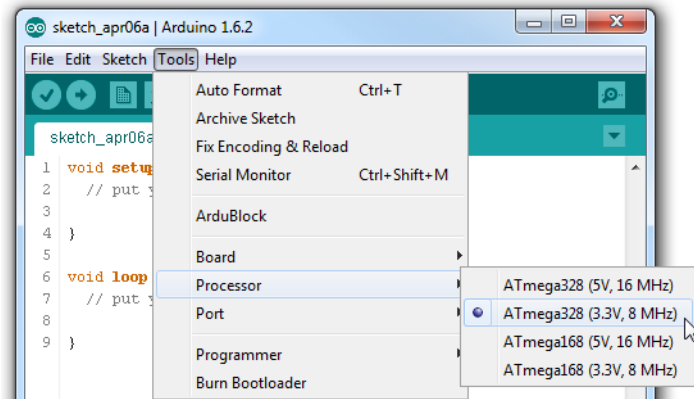
Once both Arduino and the FTDI drivers are installed, it's time to get programming. We'll start by uploading everyone's favorite sketch **Blink**. Open up Arduino, then open the Blink sketch by going to *File > Examples > 01.Basics > Blink*



Before we can upload the sketch to the Mini, you'll need to tell Arduino what **board** you're using. Go to *Tools > Board* and select **Arduino Pro or Pro Mini**.

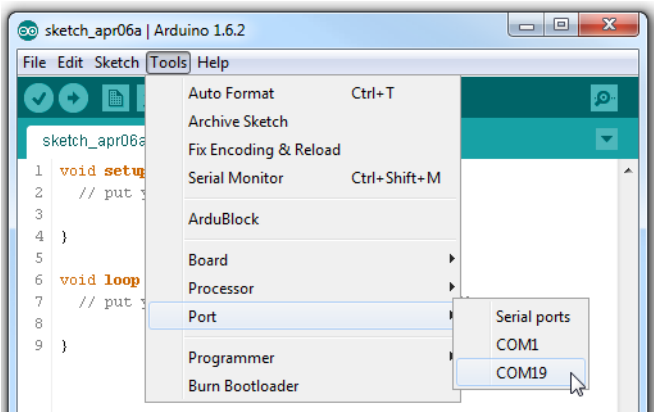


Then, go back up to **Tools > Processor** and select **ATmega328 (3.3V, 8MHz)**. This tells Arduino to compile the code with an 8MHz clock speed in mind, that way the delay(1000); calls will actually delay one second.



Heads up! This tutorial was originally written for the Arduino Pro Mini 3.3V/8MHz. If you are using an Arduino Pro Mini 5V/16MHz, you would select **ATmega328 5V/16MHz** under the Processor in the Arduino IDE when uploading code to the 5V version board.

You'll next need to tell Arduino which **serial port** your FTDI Basic Breakout has been assigned to. On Windows this will be something like COM2, COM3, etc. On Mac it'll look something like `/dev/tty.usbserial-A6006hSc`.

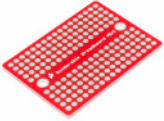


Finally, you're all set to upload the sketch to your Mini. Click on the **Upload** button (the right-pointing arrow). After a few moments you should see the red and green RX/TX LEDs on your FTDI board flash, followed by a "Done Uploading" message in Arduino's status bar. Voilà, Blinky! The Mini may be missing a few components, but it's got the most important component: LEDs!

Resources and Going Further

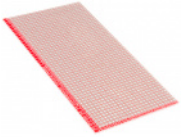
Arduino Pro Mini Prototyping

If you are looking at making a permanent connection, try soldering the Arduino Pro Mini on the following [prototyping boards](#) or stacking a shield to the boards.



[SparkFun Solder-able Breadboard - Mini](#)

PRT-12702
\$4.25



[SparkFun Snappable Protoboard](#)

PRT-13268
\$13.50



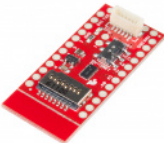
[ProtoShield for Arduino Pro Mini](#)

DEV-09709
Retired



[SparkFun MiniGen - Pro Mini Signal Generator Shield](#)

BOB-11420
Retired



[SparkFun Mini GPS Shield](#)

GPS-14030
Retired

Like its predecessor, the Arduino Pro Mini is open-source hardware, which means all of the design files are free to view and modify.

- [Schematic \(PDF\)](#)
- [Eagle Files \(ZIP\)](#)
- Graphical Datasheet
 - [5V/16MHz](#)

- [3.3V/8MHz](#)
- [GitHub](#)

For more information on the Arduino Pro Mini, first check out our [product page](#) as well as Arduino's [Pro Mini homepage](#).

Now that you know the basics of hooking up the Arduino Pro Mini, what project are you going to stick it into? Need some inspiration? Check out some of these tutorials and videos.

- [Wake on Shake](#) -- Add the Wake on Shake to provide power briefly to a Pro Mini only when there is a shake or bump.
- [The Uncertain 7 Cube](#) -- The Pro Mini is the brains of this fun riff on the magic 8-ball.
- [MP3 Player Shield Music Box](#) -- This project uses an Arduino to control the MP3s and motors, but it'd be interesting to swap that out for a Pro Mini.
- [Mario the Magician's Magical Lapel Flower](#) -- This project uses a [Pro Micro](#) to control the magical flower, but that could be swapped out for a Pro Mini too.
- [Interactive Hanging LED Array](#) -- Create a giant LED array driven by the Arduino Pro Mini.
- [Reducing Arduino Power Consumption](#) - Reduce current draw on your Arduino project.
- [LED Cloud-Connected Cloud](#) - Add the Pro Mini with some WS2812-based Addressable LEDs to light up a cloud for an installation. Better yet, have the Pro Mini talk with the ESP8266 to connect online and have the LEDs react to the weather in your region!
- [Raspberry Pi Zero W](#) - If the Pro Mini doesn't have the horsepower you're looking for, consider moving up to a Raspberry Pi platform. The Raspberry Pi Zero provides a smaller form factor.

[MP3 Player Shield Music Box](#)

Music Box Project based on the Dr. Who TARDIS.

[The Uncertain 7-Cube](#)

The Uncertain 7-Cube is a non-committal, less-than-helpful, but also entirely honest fortune teller. Simply ask it a yes or no question, give it a nudge, and the 7-Cube will dutifully inform you that it doesn't have all the facts and doesn't feel comfortable making a guess.

[Mario the Magician's Magical Lapel Flower](#)

A guest tutorial from the astonishingly talented Mario the Magician showing how to put together your own servo-controlled lapel flower.

[Interactive Hanging LED Array](#)

Learn how we converted 72 lightbulbs into an interactive LED array for our conference room.

[Wake-on-Shake Hookup Guide](#)

A basic hookup guide for getting started with the SparkFun Wake-on-Shake. The board gives you the ability to put your project into hibernation until bumped or shaken awake using the ADXL362 accelerometer. This means you can design projects meant to stay inert for long periods of time, possibly even several years, depending on the battery type used to power the project.

[LED Cloud-Connected Cloud](#)

Make an RGB colored cloud light! You can also control it from your phone, or hook up to the weather!

[Reducing Arduino Power Consumption](#)

A tutorial about different ways to reduce the current draw for your next Arduino project the easy way.

Or check out this blog post for ideas:

[T³: Programming an ATtiny84A from Arduino](#)

[November 22, 2016](#)

[Read Post](#)

[Hardware Hump Day: Installing an ATtiny Bootloader with V-USB](#)

[February 15, 2017](#)

[Read Post](#)

[Enginursday: Motion-Controlled, Wearable LED Dance Harness](#)

[January 31, 2019](#)

[Read Post](#)

[Expecto Patronum!](#)

[February 12, 2019](#)

[Read Post](#)

learn.sparkfun.com | [CC BY-SA 3.0](#) | SparkFun Electronics | Niwot, Colorado