

Tom's Hardware is supported by its audience. When you purchase through links on our site, we may earn an affiliate commission. [Learn more](#)

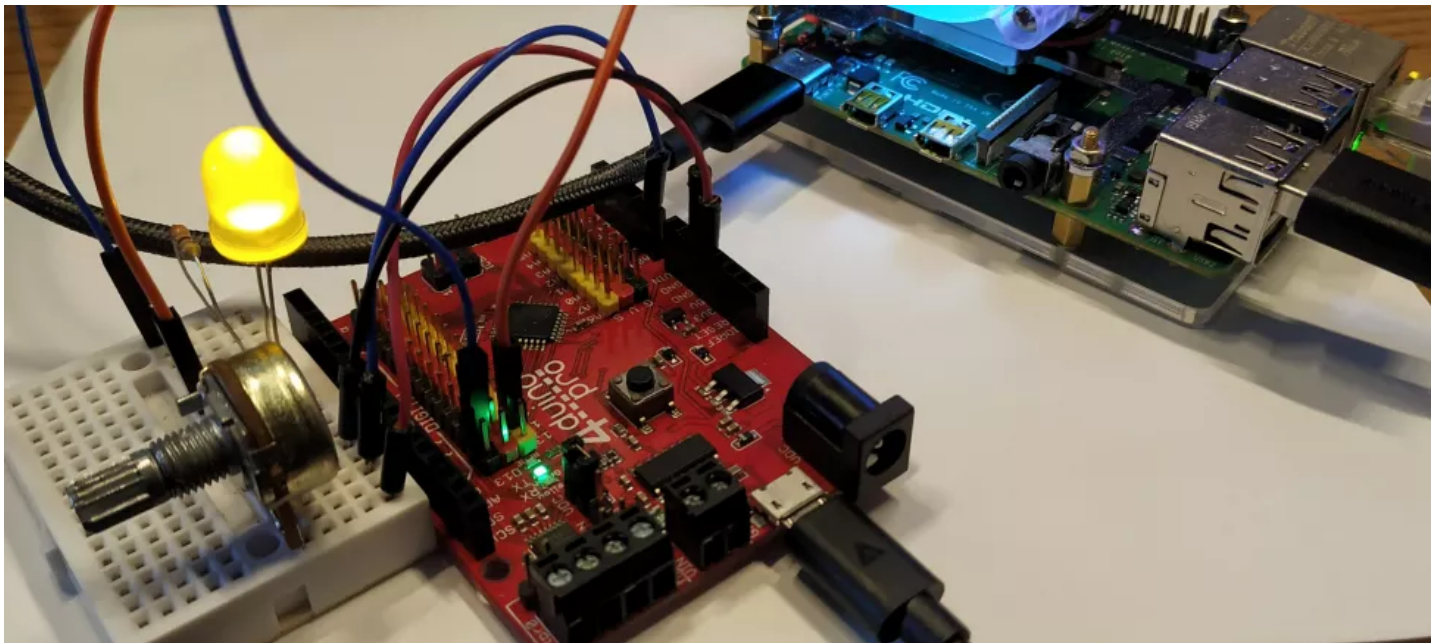
How to Use Raspberry Pi and Arduino Together

By [Les Pounder](#) May 05, 2020

Connect an Arduino board to your Raspberry Pi and take advantage of unique capabilities like a built-in ADC chip.



Comments (2)



(Image credit: Future)

The Arduino and [Raspberry Pi](#) are two very different products, but they both cater to eager hackers and makers. What if we could connect an Arduino to a Raspberry Pi and use it as a slave device: one that reacts to input and sends the output to our Raspberry Pi via Python?

After all, the Arduino, while not a full-fledged computer, has a few things it does better, like converting analog-to-digital with its built-in ADC chip. To connect the Arduino to the Raspberry Pi, we are going to need some special software, and that is where this tutorial starts.

Software Setup to Use Raspberry Pi with Arduino

Before we can write any Python code we need to download and install [Arduino IDE for Linux ARM 32-bit](#). Once it is installed, we need to add the user "pi" to the correct group to send data to the Arduino. So to add pi to the group, open a terminal and type the following:

```
$ sudo usermod -a -G dialout pi
```

Then reboot your Raspberry Pi before continuing. With the Pi rebooted, open **Arduino IDE**, and select **File > Examples > Basic > Blink** and then go to **Tool > Board** and select your board. For our tests, we used an Arduino Uno.

Then go to **Tools > Port** and make sure the port for your board is selected. Now click on **Sketch > Upload** (or click on the arrow in the menu) to upload the code to the Arduino. After a few seconds, the built-in LED of the Arduino should flash on/off slowly. This proves that we have a working unit.



```
$ sudo pip3 install pyfirmata
```

Project Hello World on Raspberry Pi with Arduino

To test that our Arduino will work with Python we will write a quick script to turn on an LED connected to pin 12 of the Arduino. Please see the diagram for the Connections.

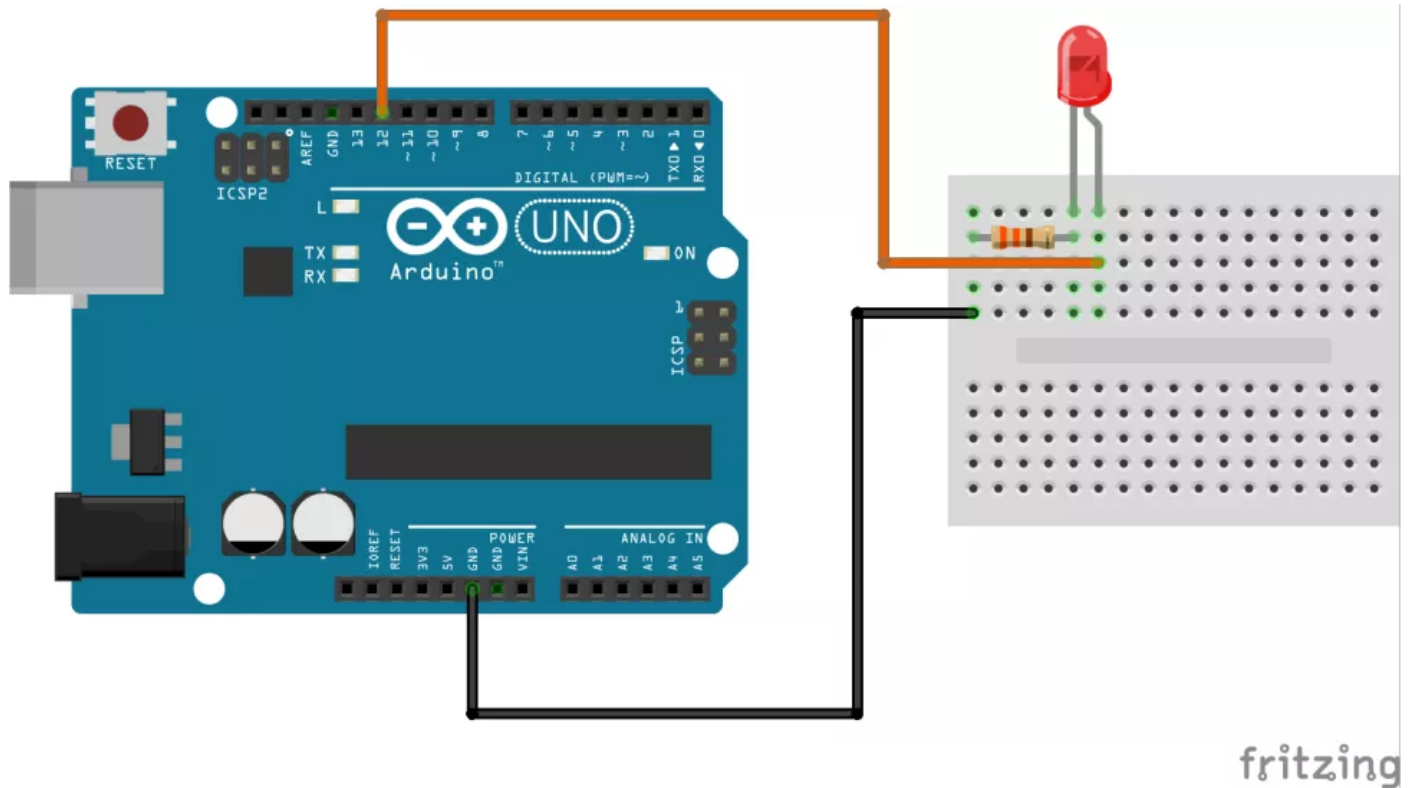


Diagram showing how Arduino Uno connects to LED light. (Image credit: Future)

Plug in the Arduino, and in the terminal type the following code. Look for USB devices such as ttyUSB0 and ttyACM0. Make a note before moving on.

```
$ dmesg
```

Using your favourite Python 3 editor (IDLE, Thonny, nano, Vim), **create a new file and name it LED_test.py**. We shall now write some Python code into this file. Start by importing two classes from the pyFirmata library, which will enable our code to connect to the Arduino. We can then import the sleep function from the time library, by **typing**:

```
from pyfirmata import Arduino, util
from time import sleep
```

The next step is to **create an object called *board*** that will be the connection from our Pi to the Arduino. For this, we shall need to use the USB device information from dmesg . In our case our Arduino was at ttyUSB0.

```
board = Arduino('/dev/ttyUSB0')
```

A variable called *led* is used to store the Arduino pin number. You create it by **adding the line**:

```
led = 12
```

×

```
while True:
    board.digital[led].write(1)
    sleep(0.2)
    board.digital[led].write(0)
    sleep(0.2)
```

Save the code and then run it from your editor (IDLE Run > Run Module/Thonny Run > Run Current Script) and after a few seconds the LED connected to the Arduino will flash, proving that we have a working connection.

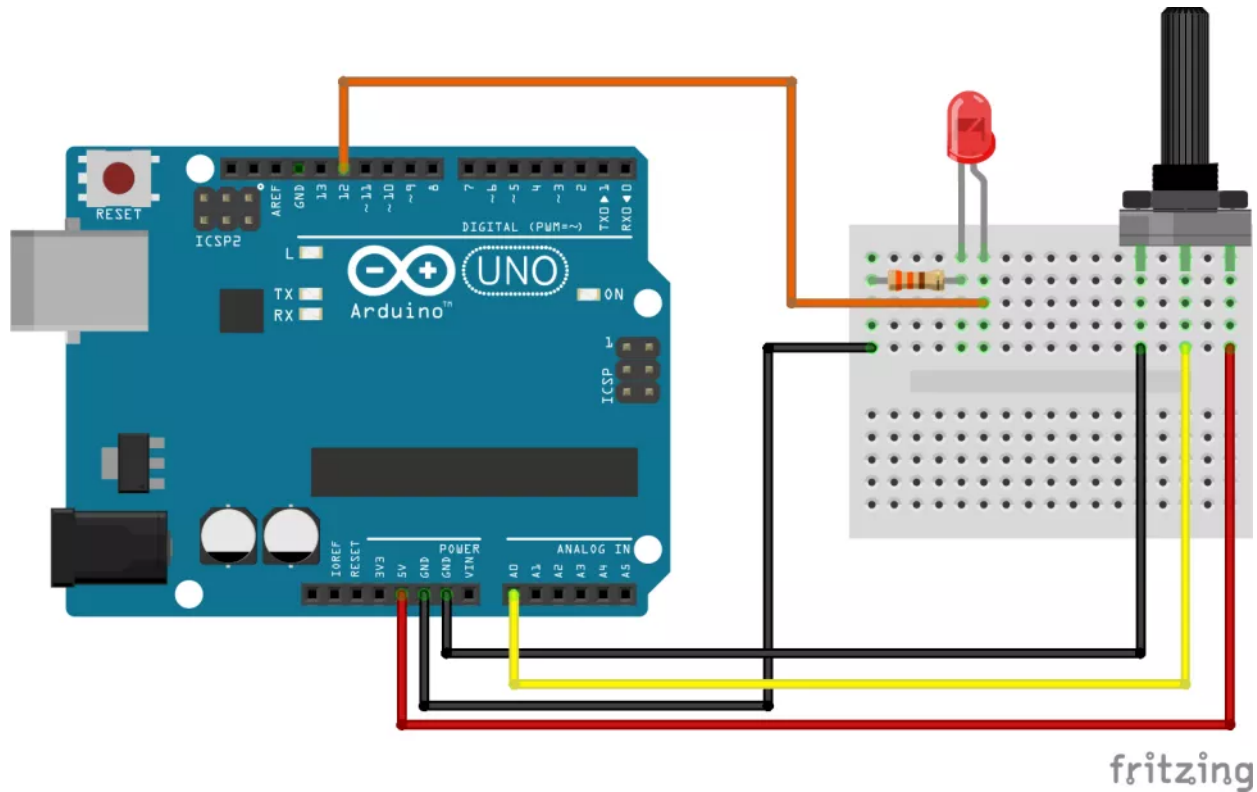
The final python script should look like this:

```
from pyfirmata import Arduino, util
from time import sleep
board = Arduino('/dev/ttyUSB0')
led = 12
while True:
    board.digital[led].write(1)
    sleep(0.2)
    board.digital[led].write(0)
    sleep(0.2)
```

Flashing LED Lights with Raspberry Pi and Arduino

Now let's make a new project. This will be an LED that flashes, but the interval between each flash is controlled via a potentiometer, an analogue electronic component -- something that the Raspberry Pi cannot ordinarily use without extra ADC (analog-to-digital conversion) boards.

We can use the value returned from the Arduino to control the speed at which the LED flashes. We will add the potentiometer to the existing LED test circuit that we have just built and tested. Please see the diagram below for more information on this.



We'll start the code for this project in a new blank file, using the same lines to import and configure the pin being used on the Arduino.

```
from pyfirmata import Arduino, util
from time import sleep
board = Arduino('/dev/ttyUSB0')
led = 12
```

To read the analogue values from the Arduino we need to **create a thread** that will run and not interrupt the main code. We should **create an object called *it*** and then connect this to the Arduino, before then starting the thread.

```
it = util.Iterator(board)
it.start()
board.analog[0].enable_reporting()
```

The main body of code is a *while True* loop, which will read the current value of analogue pin 0, which is connected to the potentiometer, and store the value in a variable called *value*. This value will then be printed to the Python shell.

```
while True:
    value = board.analog[0].read()
    print(value)
```

A conditional test is now applied to the *value* variable. If the value has no data, then it will return none, and this will crash the code. Therefore an *if* condition checks the value, and if it is none, it changes it to 0.

```
if value == None:
    value = 0
```

Another condition to test this is if the value is greater than 0.05, the analogue values returned are between 0.0 and 1.0. If the value's greater than 0.05 the LED is turned on, and the sleep interval, used to keep the LED on/off by pausing the code, is controlled by the *value*.

```
elif value > 0.05:
    board.digital[led].write(1)
    sleep(value)
    board.digital[led].write(0)
    sleep(value)
```

The final lines of code are an else condition, which turns the LED off if the value is less than 0.05. Save the code and run. Now turn the potentiometer and watch the LED come to life. Your final python code should look like this:

```
from pyfirmata import Arduino, util
from time import sleep
board = Arduino('/dev/ttyUSB0')
led = 12
it = util.Iterator(board)
it.start()
board.analog[0].enable_reporting()
while True:
    value = board.analog[0].read()
    print(value)
    if value == None:
        value = 0
    elif value > 0.05:
        board.digital[led].write(1)
        sleep(value)
        board.digital[led].write(0)
        sleep(value)
```



MORE ABOUT...

LATEST

How to Mine Ethereum: NiceHash, Mining Pools, Optimal Settings ▶

Best PSUs For Crypto Mining: Reliable Power Supplies for Ethereum ▶

SEE MORE LATEST ▶

TOPICS

RASPBERRY PI

SEE ALL COMMENTS (2)





2 COMMENTS

COMMENT FROM THE FORUMS ►

glenn_n 05 May 2020 18:34

Why not just use an A/D converter? There's less hardware, much less complicated, you only need to program the R-Pi, don't need to also write code for Arduino. This article shows you the way (except the software bits are outdated): <https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters>

REPLY ►

Dr.J 09 July 2020 22:13*glenn_n said:*

Why not just use an A/D converter? There's less hardware, much less complicated, you only need to program the R-Pi, don't need to also write code for Arduino. This article shows you the way (except the software bits are outdated): <https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters>

Because its fun and after learning that way knowing will be fun as well

REPLY ►

VIEW ALL 2 COMMENTS ►

MOST POPULAR

Block Based Coding Comes To the Raspberry Pi Pico

By [Les Pounder](#) March 04, 2021

READ MORE ►

GPU Pricing Index: Tracking eBay Sadness

By [Jarred Walton](#) March 04, 2021

READ MORE ►

The Samsung 980 SSD Spotted: A DRAM-Less PCIe 3.0 x4 SSD

By [Anton Shilov](#) March 04, 2021

READ MORE ►

Rocket Lake-S Equipped Laptops Found in China

By [Aaron Klotz](#) March 04, 2021



READ MORE ►

Raspberry Pi Cat Detector Gun Scans for Felines

By [Ash Puckett](#) March 04, 2021

READ MORE ►

Nvidia's RTX 3080 Ti Could Have a Mining Limiter

By [Aaron Klotz](#) March 04, 2021



READ MORE ►

Webcam Deal: Grab Razer Kiyo for Less Than \$90

By [Jason England](#) March 04, 2021

READ MORE ►

Geniatech Launches XPI-3288 Raspberry Pi Alternative

By [Les Pounder](#) March 04, 2021



[READ MORE ►](#)

BE IN THE KNOW

Get instant access to breaking news, in-depth reviews and helpful tips.

Your Email Address

- ☐ Contact me with news and offers from other Future brands
- ☐ Receive email from us on behalf of our trusted partners or sponsors

SIGN ME UP

No spam, we promise. You can unsubscribe at any time and we'll never share your details without your permission.

MOST POPULAR

MOST SHARED



1 Intel Ordered to Pay \$2.2 Billion in Damages for Patent Infringement

×

- 4 CPU Deal: Intel Core i9-9900K now \$100 off at Best Buy
- 5 Raspberry Pi Uses Siri to Operate a Garage Door

Tom's Hardware is part of Future US Inc, an international media group and leading digital publisher. **Visit our corporate site.**

Terms and conditions

Privacy policy

Cookies policy

Accessibility Statement

Advertise

About us

Contact us

© Future US, Inc. 11 West 42nd Street, 15th Floor, New York, NY 10036.

