

Adafruit will not be shipping orders Monday January 2, 2017. Any orders placed after 11am EST Friday December 30 will go out Tuesday January 3.



[0](#)

-
- [SHOP](#)
- [BLOG](#)
- [LEARN](#)
- [FORUMS](#)
- [VIDEOS](#)
- [SIGN IN](#)
- [CLOSE MENU](#)

[0 Items](#)

[Sign In](#)



- [SHOP](#)
- [BLOG](#)
- [LEARN](#)
- [FORUMS](#)
- [VIDEOS](#)



[DotStar Pi Painter](#)

[Ultimate light-painting rig!](#)

- [Overview](#)
- [Raspberry Pi Setup](#)
- [Assembly Part 1](#)
- [Assembly Part 2](#)
- [Assembly Part 3](#)
- [Photograph!](#)
- [Image Prep and Fine-Tuning](#)
-
- [Single Page](#)
- [Download PDF](#)

Contributors

[Phillip Burgess](#)

[Feedback? Corrections?](#)

[RASPBERRY PI LEDS / LED STRIPS](#)

Assembly Part 1

by [Phillip Burgess](#)



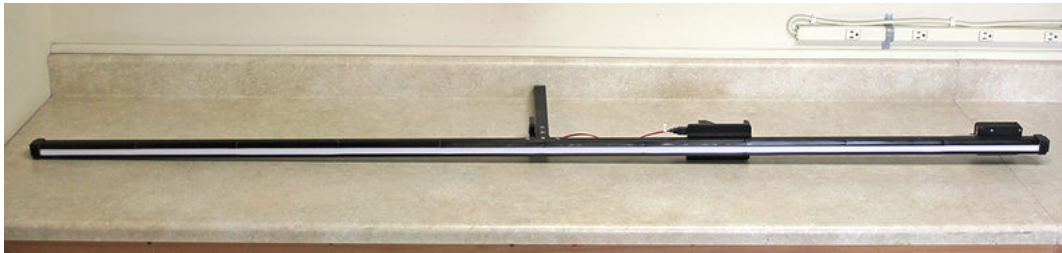


This is the “personal experience” part of the project. Every light painter will be different depending on your skillset and available tools and materials. But they’ll have certain parts in common: a Raspberry Pi, some DotStar LEDs, a few buttons and power of some sort. The rest...improvise!

Rather than rehash a lot of construction steps, much can be gleaned by skimming the [NeoPixel Painter construction](#) page. We’re using a different processor and different LEDs, but the principles behind the physical *thing* are unchanged. Seriously, check it out. I’ll wait here for you!

NeoPixel Painter construction

Due to Arduino RAM constraints and the SD card block size, the NeoPixel Painter had a hard upper limit of 170 pixels. For the DotStar Pi Painter, in order to showcase some of the Raspberry Pi’s benefits, I built a 2-meter, 288-pixel beast that dominated my garage workbench:



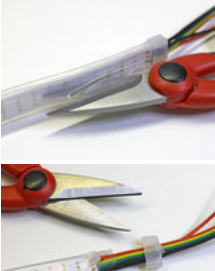
No really. It’s much bigger than it looks there.

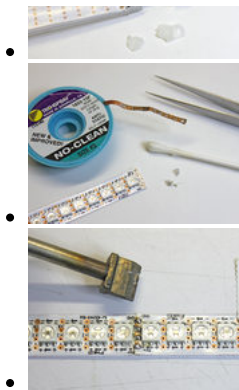
While I wouldn’t actively *discourage* you from building one this big (or even more!), I just need to mention that it’s more complex to make and *really* unwieldy at this scale, bumping into doorframes and hard to move around for anything but straight-across light painting, not to mention the challenge of getting adequate power to all those LEDs. This was more for “showoff value” than anything practical. A one-meter stick (or even 1.5m if you’re tall) still looks and works great and is easier to build and swing around for dynamic photos!



One more sound reason for making a 1-meter painter: **high-density (144 LED/m) DotStar strip comes in 1-meter lengths** and is ready to use **as-is** in its protective sheath.

Creating a longer contiguous strip involves removing the covering, desoldering and cleaning up the strip ends, and re-joining strips end-to-end. **This is deceptively challenging,** getting all four solder connections working strip-to-strip while avoiding shorts side-to-side, *working*





working quickly to not overheat & kill the end pixels. You'll need soldering skill, patience to re-do it a few times if needed, and a multimeter for testing the result each time. **All of this is avoided with a 1-meter painter!**

Test DotStar Strip

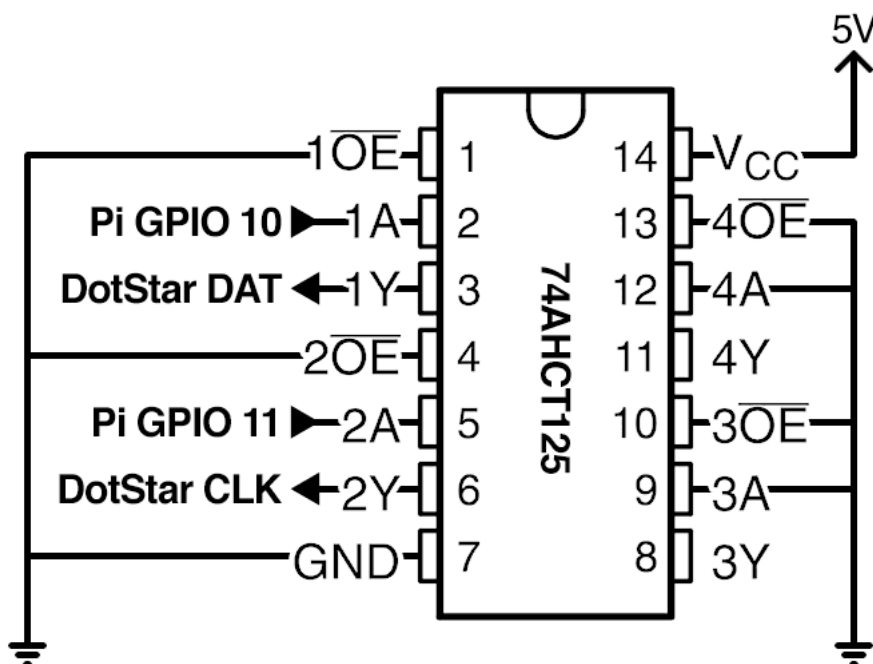
Before embarking on an elaborate construction project, let's test the DotStar strip first. It's easier to do troubleshooting up front (and get replacement parts if needed), before it's built into a *thing*. Ask me how I know this.

A **solderless breadboard** and some jumper wires, perhaps alligator clips, are *really* useful during this test phase. But otherwise, you can solder parts and wires on the Pi HAT board, replacing some wires later for final installation.

The DotStar LED strip needs to be powered from a 5V source (e.g. a "wall wart" or bench supply, or a USB battery bank). **The LED supply ground and Pi GND must be connected, but *do not* join the 5V lines!**

144 LED/m DotStar strips don't have data in/clock in labels! First, examine the strip closely, look for arrows showing the direction of data from "in" to "out"...we're connecting to the "in" end. + and - usually *are* labeled, or if your strip has wires attached, these are red (+) and black (-). The data line is adjacent to ground (usu. green wire if attached), clock is the other (usu. yellow wire). Occasionally our supplier changes these up though...we'll troubleshoot in the next step.

A **74AHCT125** IC is used to interface the **3.3V** Raspberry Pi logic to the **5V** LED strip:



Connect Vcc (pin 14) to 5V from the DotStar strip (not the Pi), and pin 7 to ground.

The 74AHCT125 has four inputs (1A, 2A, 3A, 4A), four outputs (1Y, 2Y, 3Y, 4Y) and four output-enables (1OE, 2OE, 3OE, 4OE). For this project we're using only two of each:

Raspberry Pi **74AHCT125 DotStar Strip**

GPIO 10 (SPI MOSI) 1A (Pin 2)

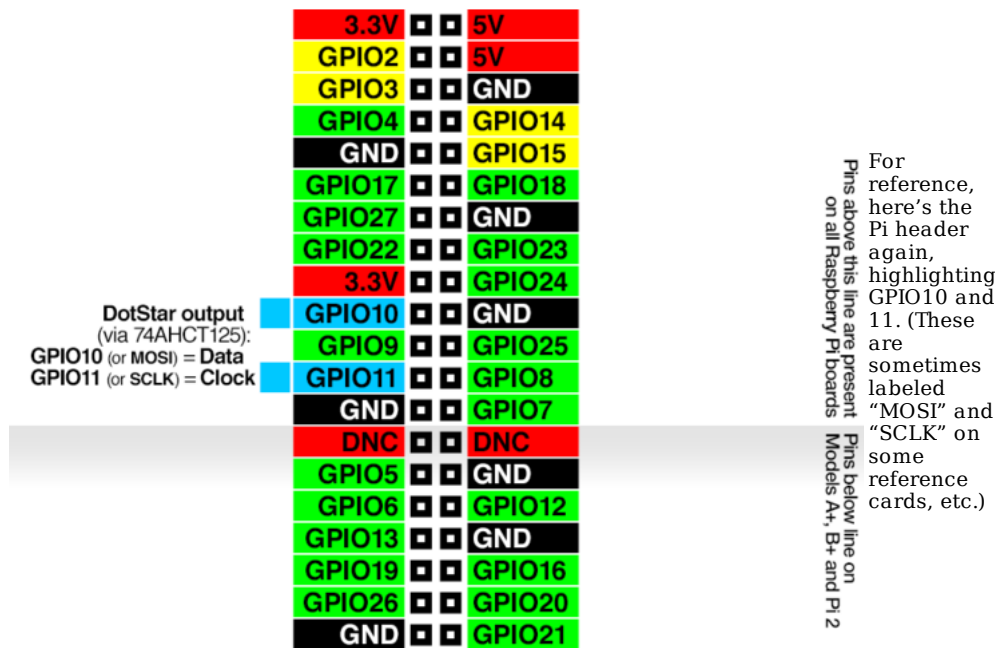
1Y (Pin 3) Data In (DI)

GPIO 11 (SPI CLK) 2A (Pin 5)

2Y (Pin 6) Clock In (CI)

1OE (Pin 1) and 2OE (Pin 4) should both be connected to ground to enable the 1Y and 2Y outputs.

Though we're not using channels 3 and 4, it's good practice not to leave these inputs "floating." Connect 3A, 4A, 3OE and 4OE to ground. 3Y and 4Y can be left unconnected.



Now we'll use a Python program on the Raspberry Pi to test the LED strip. We need to fine-tune some details first though...

[Copy Code](#)

```
1. cd
2. cd Adafruit_DotStar_Pi
3. nano strandtest.py
```

Look for this line early in the code:

[Copy Code](#)

```
1. numpixels = 30 # Number of LEDs in strip
```

Change "30" to the number of LEDs in your DotStar strip; 144 or whatever size you've built there.

A few lines down, look for this:

[Copy Code](#)

```
1. strip = Adafruit_DotStar(numpixels, datapin, clockpin)
```

and change it to:

[Copy Code](#)

```
1. strip = Adafruit_DotStar(numpixels, 12000000)
```

(That's 12 followed by six zeros...12 million.)

Save the changes and run this script using "sudo":

[Copy Code](#)

```
1. sudo python strandtest.py
```

If all goes according to plan, you'll see a few LEDs chase down the length of the strip...first red, then green, then blue...repeating forever (or until you press Control+C). Let it run, watch closely and make sure every LED along the strip lights all three colors.

Success? Skip ahead to the "Assembly" section below.

Troubleshooting

No lights! Nothing at all!

1. Check the connections to the LED strip.

- Confirm DotStar strip is connected to 5V power source.
- Confirm data/clock connection to the INPUT end of the strip (look closely for arrows printed on the strip — these show the direction of data from “in” to “out”).
- Confirm ground is connected between the Raspberry Pi and the DotStar power sources.
- Confirm 74AHCT125 is oriented the right way. The “bite” on one end of the chip indicates the Pin 1 end; pins are numbered counterclockwise from there.
- Double-check wiring between the Raspberry Pi, 74AHCT125 and DotStar strip using diagram and explanation above. Are you connected to GPIO 10 and 11?
- Try switching the “data” and “clock” pins; you might have a different generation of DotStar strip with the wires in a different order.

LEDs are flickering madly!

- Confirm ground is connected between the Raspberry Pi and the DotStar power sources.
- Try switching the “data” and “clock” pins; you might have a different generation of DotStar strip with the wires in a different order.
- Stop the program (Control+C) and try changing the “12000000” to a lower value like “4000000” (4 million), run again. If this improves things, edit the code to find the fastest reliable value using a divide-and-conquer approach (e.g. split the difference — try 8 million, then either 10M or 6M depending on the outcome, repeat).

LEDs work, but the sequence isn’t red-green-blue.

You might have a different revision of DotStar strip; the native color order was changed between the initial and current product. Change the “strip” line in the script to read:

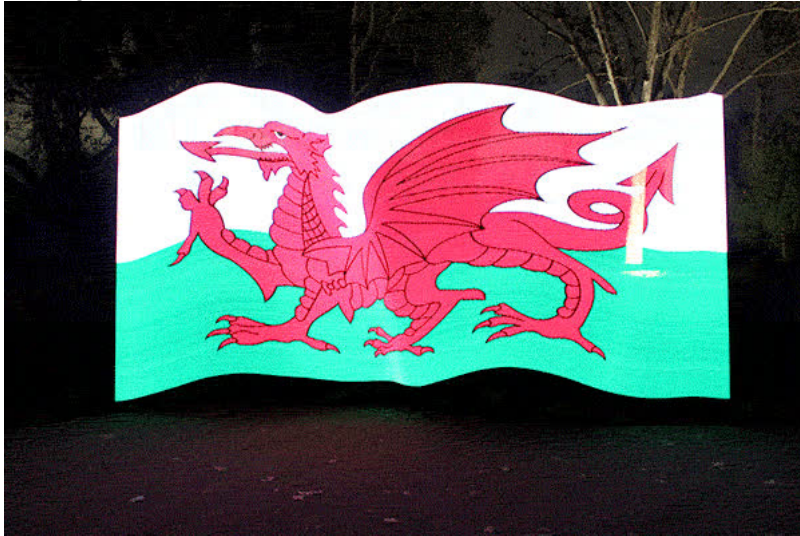
[Copy Code](#)

```
1. strip = Adafruit_DotStar(numpixels, 12000000, order='gbr')
```

Other problems?

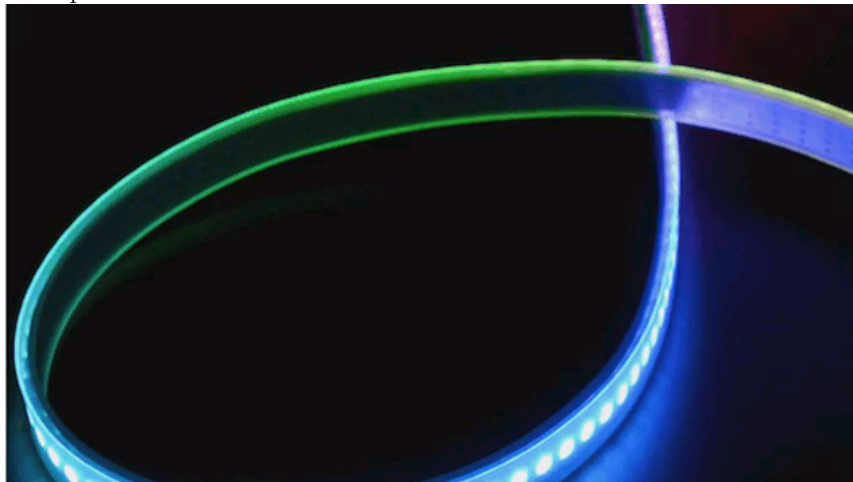
If the above steps aren’t working or there’s some other issue that’s not addressed here, start a new thread in the [Adafruit Forums](#). It’s *extremely* helpful if you can provide quality photos showing any wiring and/or soldering between the Raspberry Pi, level-shifter chip, DotStar LED strip and power source.

Do not continue until you have a working DotStar LED strip with the Raspberry Pi “talking” to the correct end.



[RASPBERRY PI SETUP ASSEMBLY PART 2](#)

Last updated on 2015-12-16 at 03.24.46 PM Published on 2015-12-11 at 01.35.28 PM





Adafruit DotStar Digital LED Strip - White 144 LED/m - One Meter
\$74.95 [Add To Cart](#)



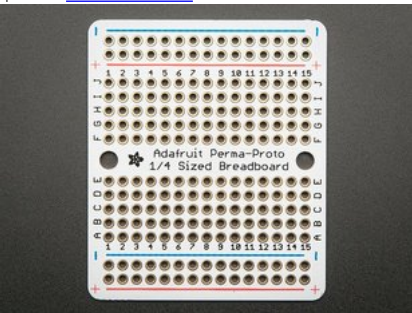
Raspberry Pi 2 - Model B - ARMv7 with 1G RAM
\$39.95 [Add To Cart](#)



Raspberry Pi Model A+ 512MB RAM
\$24.95 [Add To Cart](#)



Adafruit Perma-Proto HAT for Pi Mini Kit - No EEPROM
\$4.95 [Add To Cart](#)



Adafruit Perma-Proto Quarter-sized Breadboard PCB - Single
\$2.95 [Add To Cart](#)



74AHCT125 - Quad Level-Shifter (3V to 5V)
\$1.50 [Add To Cart](#)





Colorful Square Tactile Button Switch Assortment - 15 pack
\$5.95 [Add To Cart](#)



USB Battery Pack for Raspberry Pi - 10000mAh - 2 x 5V outputs
\$39.95 [Add To Cart](#)



USB DIY Connector Shell - Type A Male Plug
\$0.95 [Add To Cart](#)
[ADD ALL TO CART](#)

RELATED GUIDES

[Raspberry Pi Wearable Time Lapse Camera](#)

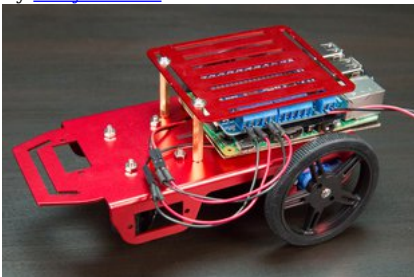
[Tiny Computer. Tiny Camera](#)
by [Phillip Burgess](#)



[Worn on a lanyard or clipped to a pocket or pack, this adorable camera snaps a photo every few seconds. Slide the SD card into your computer to review the day's activities or merge all the images into a timelapse animation. Powered by the diminutive and affordable Raspberry Pi Zero, this DIY project is eminently configurable and customizable!](#)

[Simple Raspberry Pi Robot](#)

[Build a simple moving robot with a Raspberry Pi and Adafruit's excellent robot platform!](#)
by [Tony DiCola](#)

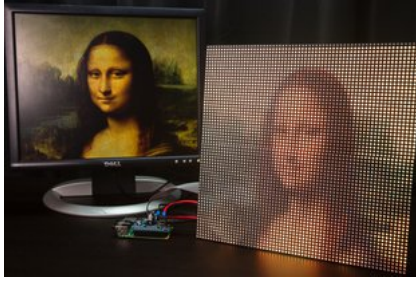


[Use a Raspberry Pi and Adafruit Motor HAT to control two DC motors that move a simple robot around using Python code. This is a great base for starting with a Pi robot project!](#)

[Raspberry Pi LED Matrix Display](#)

[Show the Pi's video output on a large RGB LED matrix display!](#)

by [Tony DiCola](#)



[Learn how to make a chain of RGB LED matrices into a large display that shows the video output of a Raspberry Pi. Great for playing games, movies, and more on a big, bright, beautiful display!](#)

[FEATURED](#)

[Introducing the Raspberry Pi 2 - Model B](#)

[We can rebuild the Raspberry Pi, we have the technology!](#)

by [lady ada](#)



[Didn't think the Raspberry Pi could get any better? You're in for a big surprise! The Raspberry Pi 2 Model B is out and it's amazing! With an upgraded ARMv7 multicore processor, and a full Gigabyte of RAM, this pocket computer has moved from being a 'toy' to a real desktop PC!](#)

[×](#)

OUT OF STOCK NOTIFICATION

YOUR NAME

YOUR EMAIL

[NOTIFY ME](#)

- [CONTACT](#)
- [SUPPORT](#)
- [DISTRIBUTORS](#)
- [EDUCATORS](#)
- [JOBS](#)
- [FAQ](#)
- [SHIPPING & RETURNS](#)
- [TERMS OF SERVICE](#)
- [PRIVACY & LEGAL](#)
- [ABOUT US](#)

[ENGINEERED IN NYC](#) Adafruit ®

"Achieving a goal provides immediate satisfaction; the process of achieving a goal is a lasting pleasure" - [Evelyn Berezin](#)

