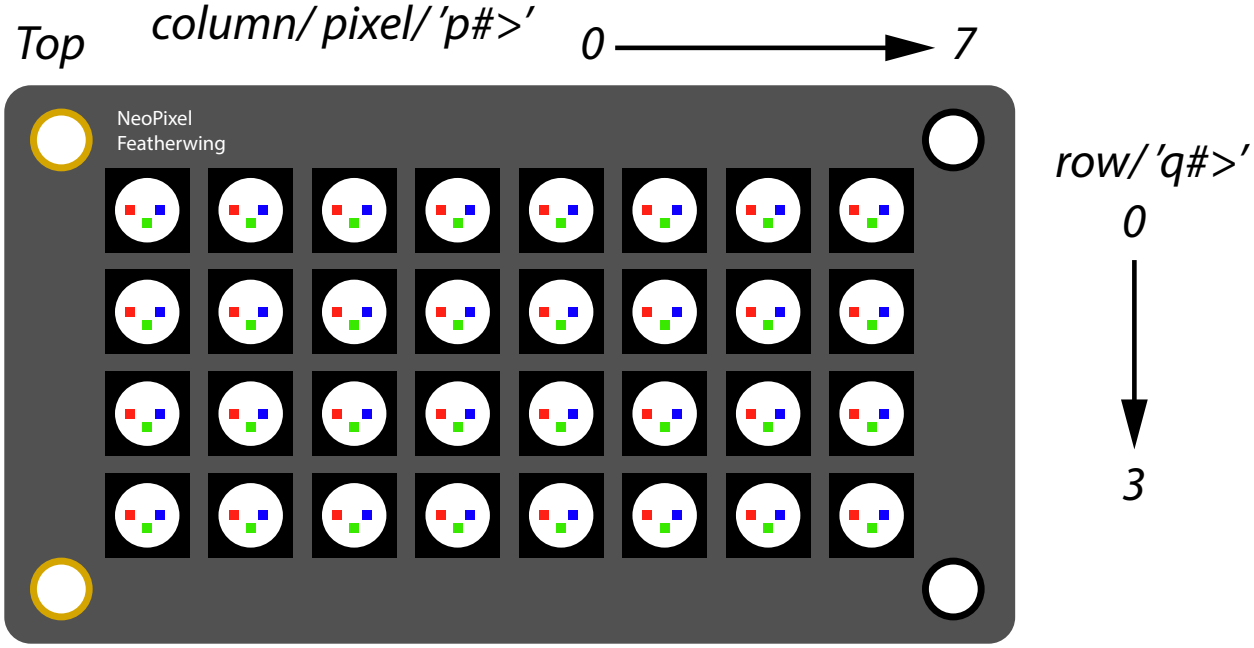
cadSerialPixels Documentation - v1.0
1/15/2018
Send ?s to: cdeister@brown.edu
https://github.com/cdeister/cadNeopixelCode



Above is the pixel map showing how to address the pixels.

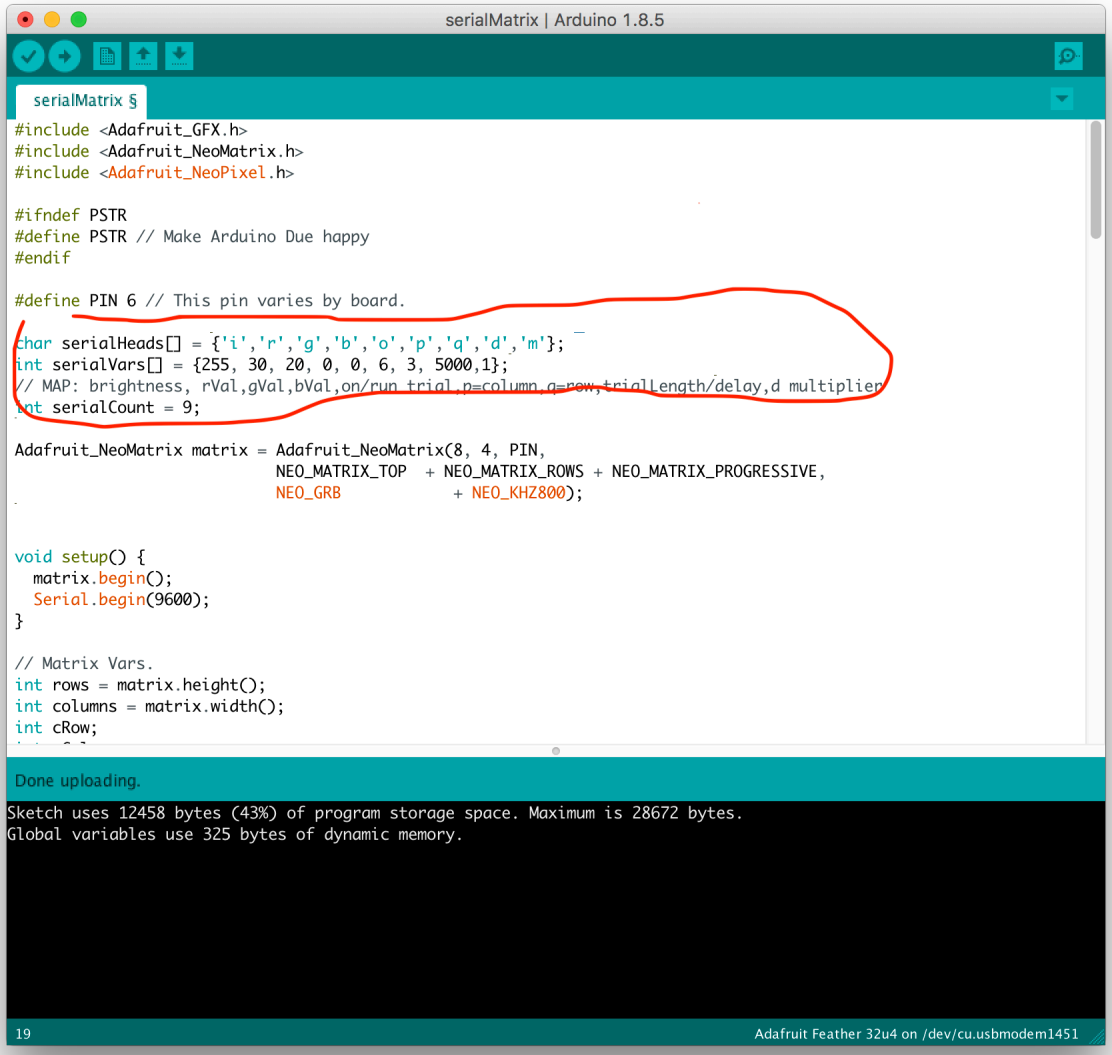
You can pass a serial line that flanks an integer between 0 and 7 with a 'p' and an '>' to set the pixel you want in a trial. You can also pass a serial line that flanks and integer between 0 and 3 with a 'q' and an '>' to set the row/line.

The serial commands possible are:
'i#>' This sets brightness. # can be 0 or 255.
'r#>' This sets the red value. # can be 0 or 255.
'g#>' This sets the green value. # can be 0 or 255.
'b#>' This sets the blue value. # can be 0 or 255.

'p#>' This sets the pixel in column p. # can be 0 or 7.
'q#>' This sets the matrix row. # can be 0 or 3.
Note: see figure above.

'd#>' This sets the duration of the trial in milliseconds. # can be 0 to 65535.
'm#>' This sets a duration multiplier. # can be 0 to 65535.
Note: total duration is d*m
'o#>' This sets a trial to be on or off. # can be 0 (on) or 1 (off).
Note: always set *o* last.

You can just run trials by changing these variables by hand in the Arduino IDE and pushing to the micro-controller.

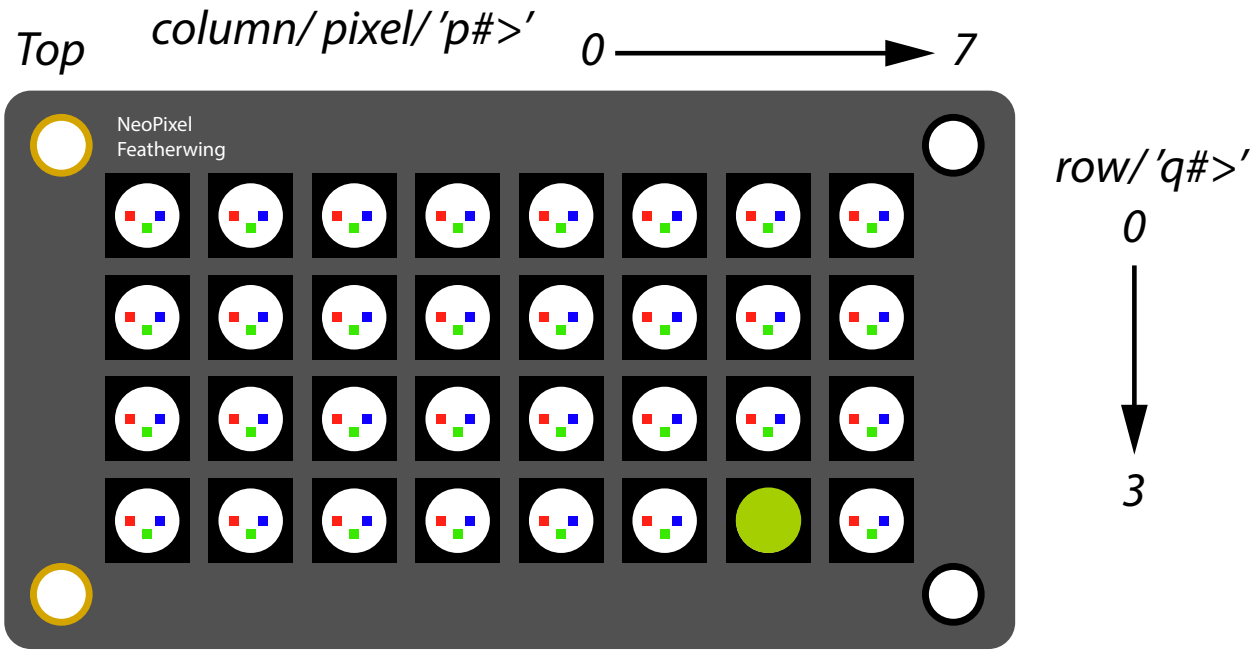


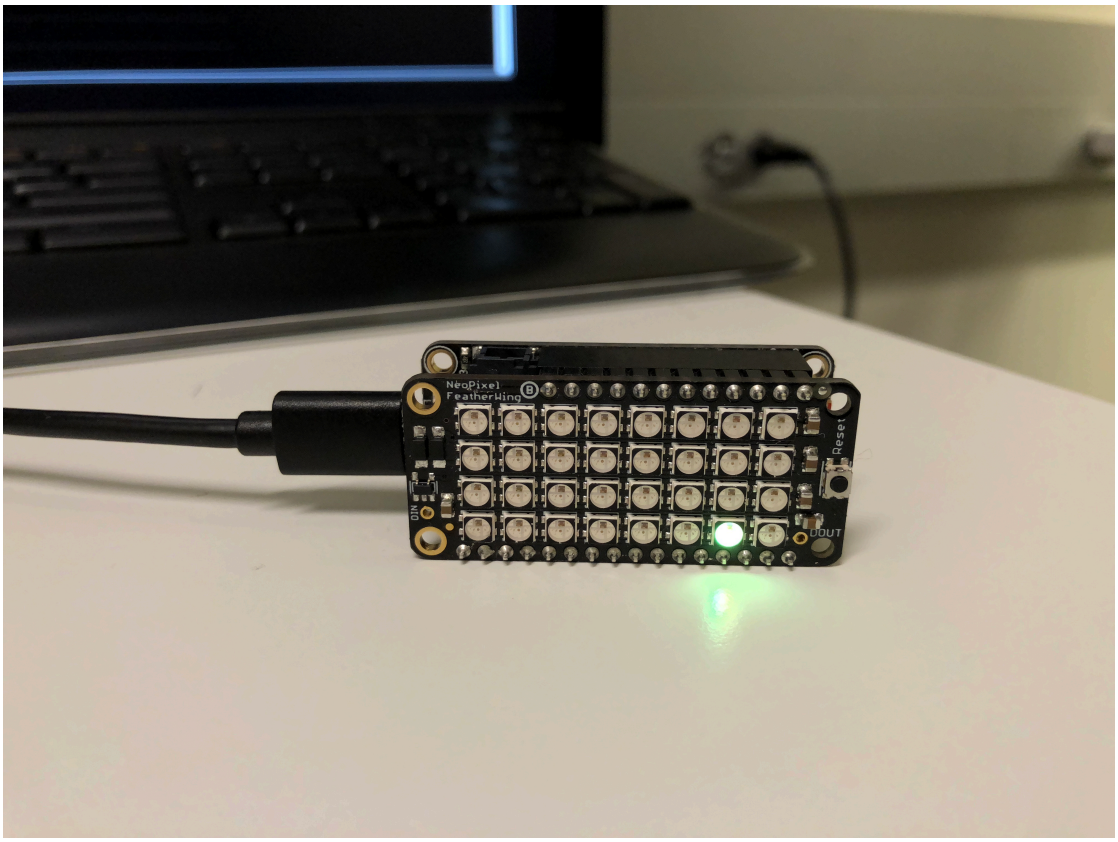
The variables described on page 1 are kept in two structs (curly brackets) that can be seen in the figure above circled in red.

For manually setting the values, the only one that matters is the “serialVars[]” struct. In the posted code I set the values to provide an example, but normally they’d be initialized to 0.

As set here, the order of the Vars follows the “serialHeads” structs order.
So, i/brightness is set to max: 255; *red value* is30/255; *blue value* is 20/255; *green value* is 0/255. Further, ‘on’ is set to 0; p/column is 6 (next to last); q/row is 3 (the last one); duration of a trial is 5000 (5 seconds); the multiplier is 1.

Thus, what I expect is the next to last pixel (reference from the top/left) on the last row to pulse a greenish yellow color for 5 seconds.





If you want to set all pixels a particular color, set p to -1.

```
char serialHeads[] = {'i','r','g','b','o','p','q','d','m'};
int serialVars[] = {255, 126, 0, 0, 0, -1, 0, 5000,1};
// MAP: brightness, rVal,gVal,bVal,on/run trial,p=column,q=row,trialLength/delay,d multiplier
int serialCount = 9;
```

This will give half full red for 5 seconds on all pixels.

Next I will show you how to control it from Matlab:
Open “matlabInteraction.m” in Matlab and it is documented there.

Extras (installing feather support in your Arduino IDE):

TLDR:
Add to board managers in preferences
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

Then search adafruit and install SAMD and AVR support.

For adafruit full instructions:

M0: https://learn.adafruit.com/adafruit-feather-m0-basic-proto/setup
32u4 (ATMEGA): https://learn.adafruit.com/adafruit-feather-32u4-bluefruit-le?view=all#setup

Also, instructions for the matrix:
https://learn.adafruit.com/adafruit-neopixel-featherwing?view=all