

## Using Color Codes

### Read this first!

Color code support is currently beta status. Please follow these steps (in order) and all should be good:

1. Download PixyMon version 1.0.2beta ( [Windows](#), [Mac](#), [Linux](#))
2. Install the new version of PixyMon.
3. Download [firmware version 1.0.2beta](#)
4. Use the new PixyMon to upload the new firmware. Follow the [firmware upload instructions](#).
5. Your Pixy and PixyMon should now be talking happily together.
6. If you're using an Arduino, download the new [Arduino library 0.1.5](#).
7. Delete the old Arduino library before installing the new one. Go to your Arduino documents directory; inside you will find a directory named "libraries". Just delete the Pixy directory inside the libraries directory and restart the Arduino app.
8. Import the new Pixy library

OK, now we're ready!

### Introduction

So, what's a "color code"? A color code (CC) is two or more color tags placed close together. Pixy can detect and decode CCs and present them as detected objects (firmware versions 1.0.0 and greater).



For example, CCs with 2 color tags using 4 different colors (identified by teaching Pixy 4 different color signatures) can detect up to 12 unique objects. CCs with 3, 4 and 5 tags and/or more different colors are possible and can allow for many, many more unique objects. Pixy will detect and report color codes with up to 5 tags arranged in a line. With 5 tags, thousands of unique codes are possible.

### Why Color Codes?

CCs are useful if you have lots of objects you want to detect and identify — more than could be detected with the seven separate color signatures alone. CCs also improve detection accuracy by decreasing false detections. That is, there is a lower probability that specific colors will occur both in a specific order and close together. The drawback is that you need to place a CC on each object you're interested in detecting. Often the object you're interested in (yellow ball, purple toy) has a unique color signature and CCs aren't needed. Objects with CCs and objects without CCs can be used side-by-side with no problems, so you are free to use CCs for some objects and not others.

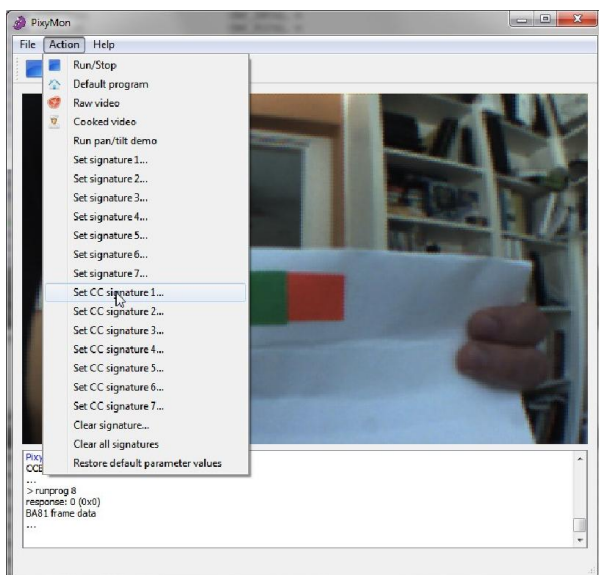
CCs give you an accurate angle estimate of the object (in addition to the position and size). This is a "freebie" that some applications may find useful. The angle estimate, decoded CCs, regular objects and all of their positions and sizes are all provided at 50 frames per second.

CCs might be particularly useful for helping a robot navigate. For example, an indoor environment with CCs uniquely identifying each doorway and hallway would be both low-cost and robust.

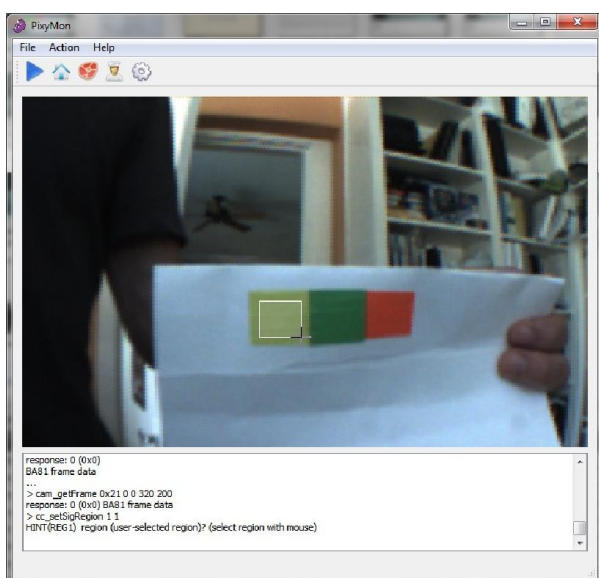
In order to get color codes working on Pixy, you need to do the following:

1. Make sure that the **Color code mode** parameter is not 0 (disabled). (It should be 1 (enabled) by default.)
2. Teach Pixy the color code signatures needed for the color codes you are using.

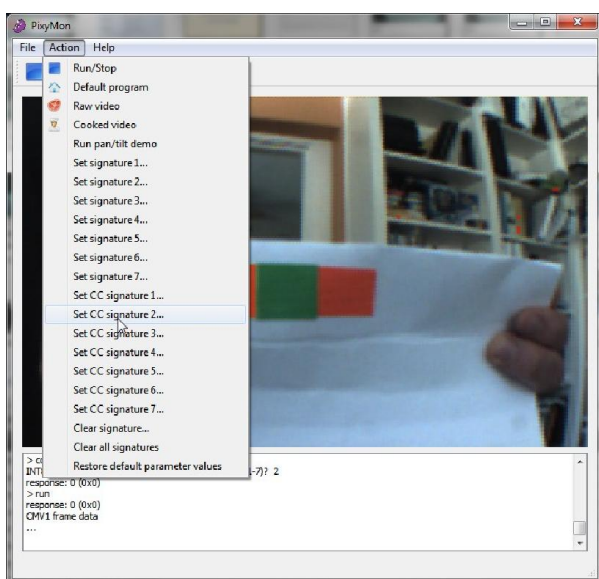
To teach Pixy the color code signatures, bring up PixyMon and choose Raw mode. Put a color code in front of Pixy and select **Action→Set CC signature 1** and select the first color in the color code.

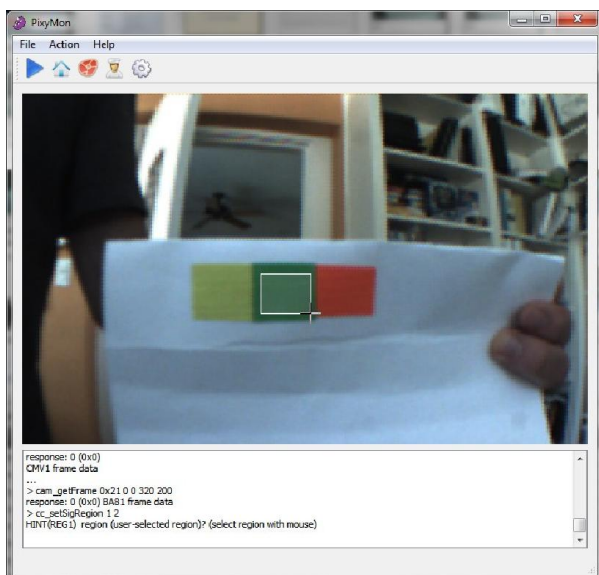


Then outline the first tag on the color code.



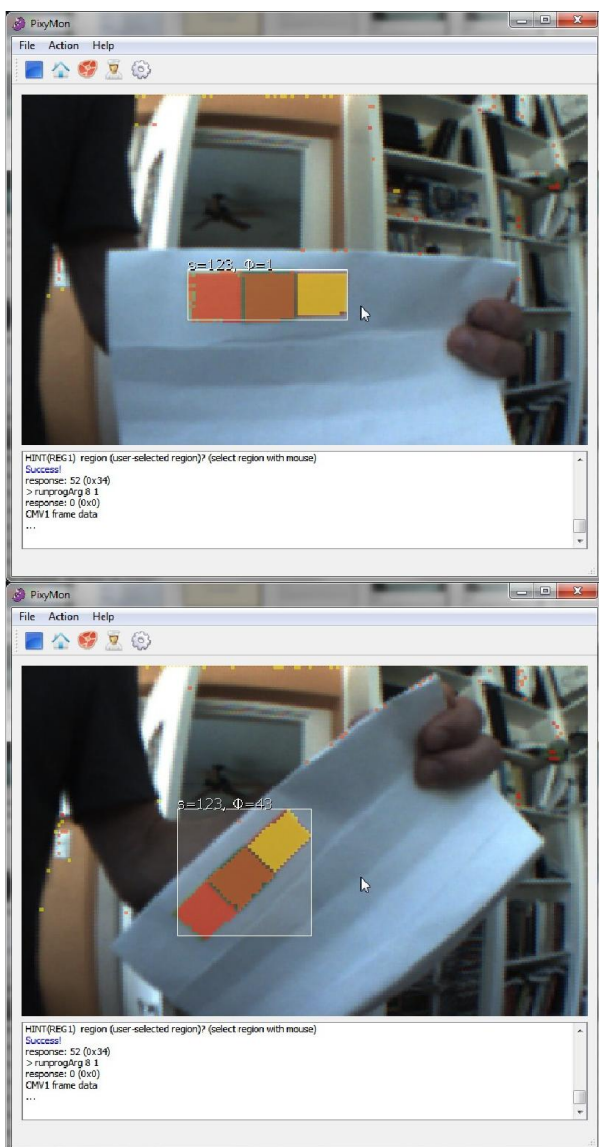
Pixy will then generate the first color code signature and store it. Next, select **Action** → **Set CC signature 2** for the second color in the color code and select the next tag.



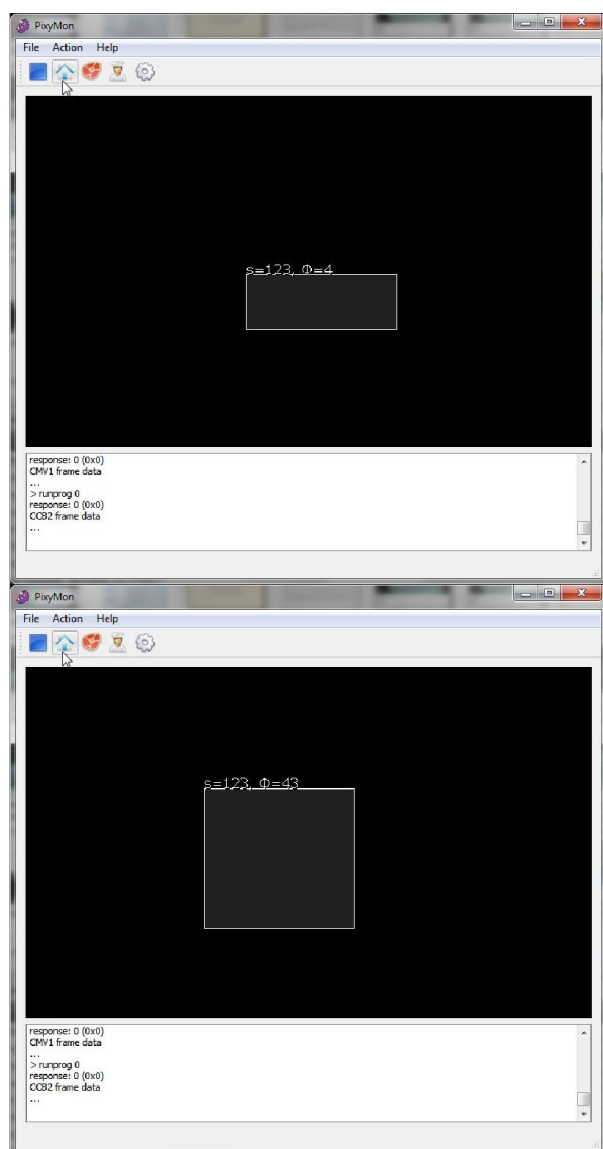


Repeat this process until all the unique colors are taught to Pixy as color codes. When all color code signatures have been taught, Pixy should be identifying color codes that are present, using the color code signatures that you just taught it.

You'll see something similar (below) in Cooked mode.



And here's what you'll see in Default mode.



And that's the gist of color codes -- fairly simple, really.

### What's this "color code signature" business? How are color code signatures different from normal signatures?

As you know, Pixy has seven color signatures that it can learn/remember. Each signature can either be a **normal signature** or a **color code signature**. For now, all signatures that you teach Pixy are considered **normal** unless you choose a **Set CC signature** option from the **Action** menu. Color code signatures, as you can probably guess, are only used for color codes. In other words, these signatures need to form a color code (two or more color code signatures) for Pixy to recognize them as an object and report them. This is the default color code behavior, but the **Color code mode** can change this (see below).

Color code signatures also have their own set of signature creation parameters: CC min saturation, CC hue spread, and CC saturation spread. You can set these parameters such that they work best with the color code tags you are using.

### How color codes are reported

Color code objects are returned as special object blocks, which are very similar to normal object blocks (see below).

Color code object block

Bytes	16-bit words	Description
0, 1	0	sync (0xaa56)
2, 3	1	checksum (sum of all 16-bit words 2-6)
4, 5	2	color code number
6, 7	3	x center of object
8, 9	4	y center of object
10, 11	5	width of object
12, 13	6	height of object
14, 15	7	angle of object (only applies to color code objects)

But the sync code is different (0xaa56 instead of 0xaa55) and instead of the signature number (word 1) returning a single number between 1 and 7, color code object blocks return an octal (base 8) number with each digit being the signature number of the detected tag (16 bits can represent up to 5 octal digits). For example, the color code above, which consists of 3 tags, signatures 1, 2, and 3, in order, is 123 in octal (base 8), or 83 in decimal.

The way color codes are reported raises a question – since color codes can be in any orientation, aren't there two possible ways to report a color code? For example, a color code that has three tags consisting of: color code signature 1, next to color code signature 2, next to color code signature 3, could have an octal representation of 123, or possibly 321. Which representation is used? The smaller representation (123) is always reported for consistency.

### Color code modes

Pixy supports four different color code modes through the **Color code mode** parameter.

1. **0=disabled** No color codes are detected or reported. Objects that match color code signatures are reported as normal objects.
1. **1=enabled** This is the default mode. Detected color code objects are reported along with normal objects. However, objects that match color code signatures, but are not part of a color code (two or more color code signatures) are not reported.
1. **2=color codes only** Objects that match color signatures whether they are normal signatures or color code signatures are reported only if they are part of color codes (two or more signatures).
1. **3=mixed** In this mode, color code objects are reported along with normal objects. Additionally, objects that match color code signatures, but are not part of a color code are also reported.

Now, that may have been confusing... We expect most people will find that mode 1 is the most useful, so use mode 1 if in doubt.