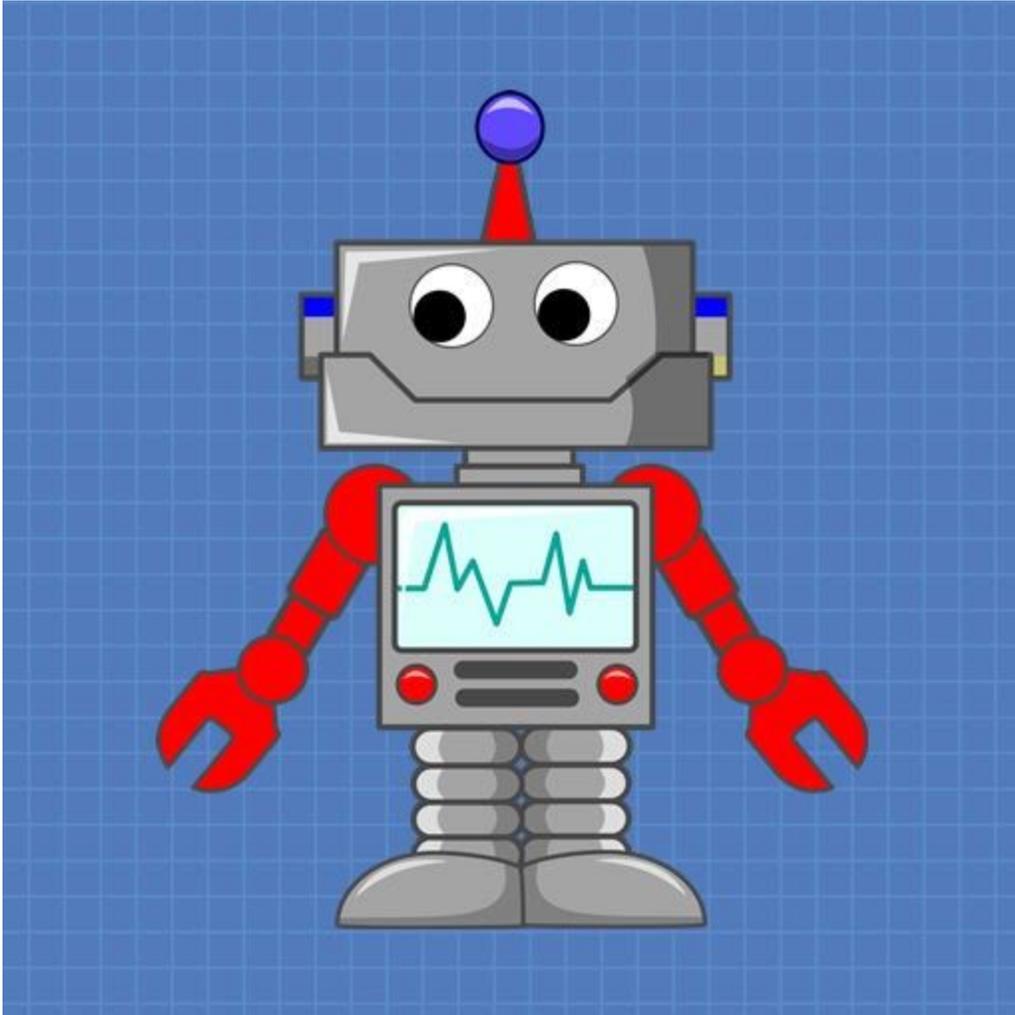


Pixy2 Camera - Object Recognition



DroneBot Workshop Tutorial

<https://dronebotworkshop.com>

Table of Contents

Introduction	2
The Pixy2 Camera	2
Pixy2 History	4
Pixy2 Features	6
Pixy2 Wiki	9
PixyMon	9
Installing PixyMon	9
Using PixyMon	10
Object Recognition	12
Capturing Signatures Manually	13
Capturing Signatures with PixyMon	16
Labels and Fine Tuning	17
Line Tracking	18
Intersections	20
Barcode Detection	20
Connecting Pixy2 to Arduino	21
SPI Connection Cable	22
Arduino Library	24
Arduino Sketch – CCC Hello World	24
Arduino Sketch – Line Hello World	29
Conclusion	32

Introduction

The ability to recognize objects has been a goal of computer scientists and AI researchers for decades. In the past this has required large computers running sophisticated software which has kept the technologies involved confined to labs and research departments with large budgets.

Flash forward to today and object recognition has become mainstream. Google and Facebook can identify faces from photographs and tag the pictures, advertising billboards can (in a somewhat controversial fashion) [identify a person's gender and age to cater ads to them based upon the results](#) and [solve Rubik's Cubes](#).

Object recognition and computer vision technology is now available for experimenters as well, with several kits and cameras with various capabilities available.

Today we will examine one of these offerings, the Pixy2 Camera.

The Pixy2 Camera

The [Pixy2](#) is a small camera designed for object recognition, line tracking and simple barcode reading. The device I used for this article and video was supplied courtesy of [DFRobot](#).



The Pixy2 is capable of recognizing seven distinct objects based upon their shape and color (or hue). Each of these objects is assigned a unique “signature”.

The camera also has algorithms for line following. Unlike traditional line followers the Pixy2 can “look ahead” to determine when the line it is following (called a “vector”) is going to turn or cross another line (i.e. an “intersection”). This is similar to the method you use when you walk, bike or drive down the street – you look ahead to anticipate any turns or stops you’ll need to make.



The Pixy2 can also detect 16 simple barcodes. You can use these as visual indicators for your robot project.

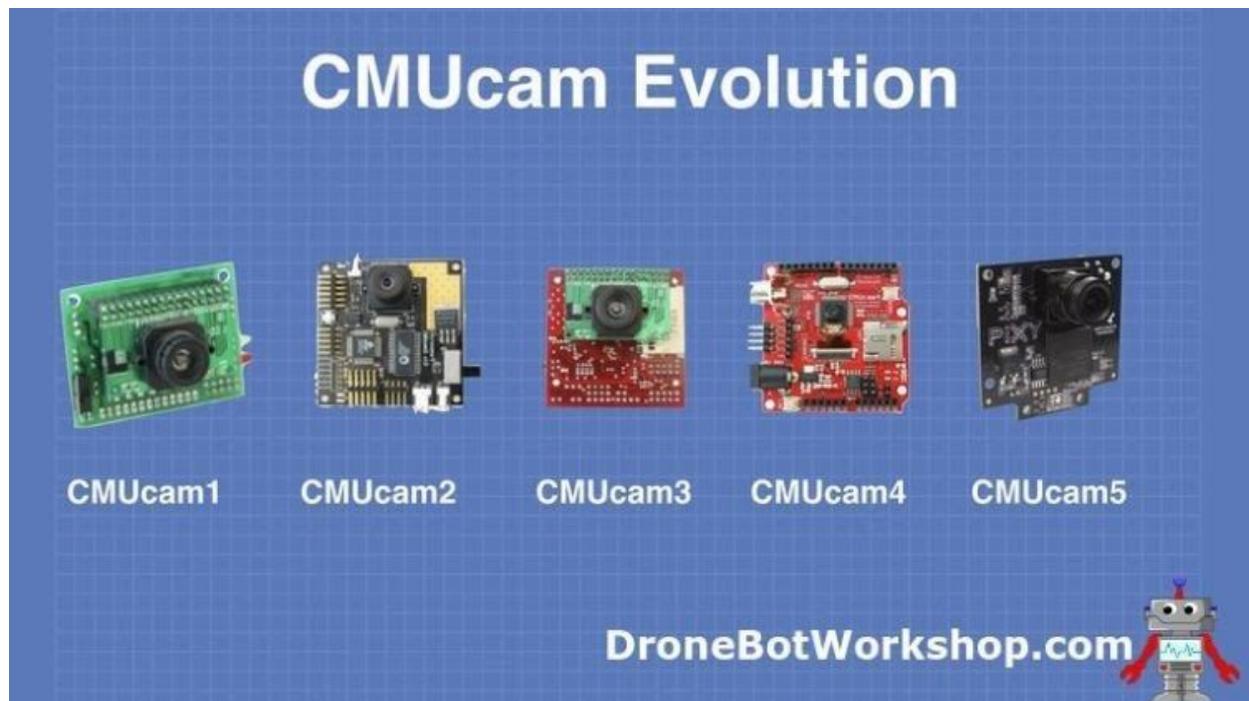
The Pixy2 is a self-contained unit, its onboard processor takes care of all the “heavy lifting” – recognizing specific objects and filtering out extraneous objects. This frees up your microcomputer or microcontroller to perform other operations, using the Pixy2 as an “intelligent sensor”.

Pixy2 History

The origin of the Pixy2 can be traced back to the CMUcam, a device developed at the [Robotics Institute at Carnegie Mellon University](#) in 1999. This device was one of the

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

world's first affordable vision sensors and over the years it has gone through **several** different iterations.



The fifth version of the camera (CMUcam5) was a joint effort of the robotics team at CMU and [Charmed Labs](#). The device was funded in 2014 by a successful Kickstarter campaign under the name Pixy Cam.

The original Pixy Cam was a breakthrough in both capability and price. It could be used with an Arduino, Raspberry Pi or on it's own as a computer peripheral.

The Pixy2 is an improvement on the original. It is smaller, faster and has a fixed focus lens. It is also much more powerful. Unlike the original Pixy Cam the Pixy2 was not a crowdfunded project, Charmed Labs designed and built the Pixy2 using the profits made from sales of the original Pixy Cam.

Pixy2 Features

The Pixy2 is the latest (as of this writing) version of the Pixy Cam.



In addition to having all of the features of the original Pixy Cam the Pixy2 has these additional functions:

- Detects lines, intersections and simple barcodes.
- Has a framerate of 60 frames per second.
- Has an integrated light source consisting of two white LEDs.
- New tracking algorithms that allow you to track objects.

The Pixy2 has a variety of connections that can be used to connect it to a computer, microcomputer or microcontroller.

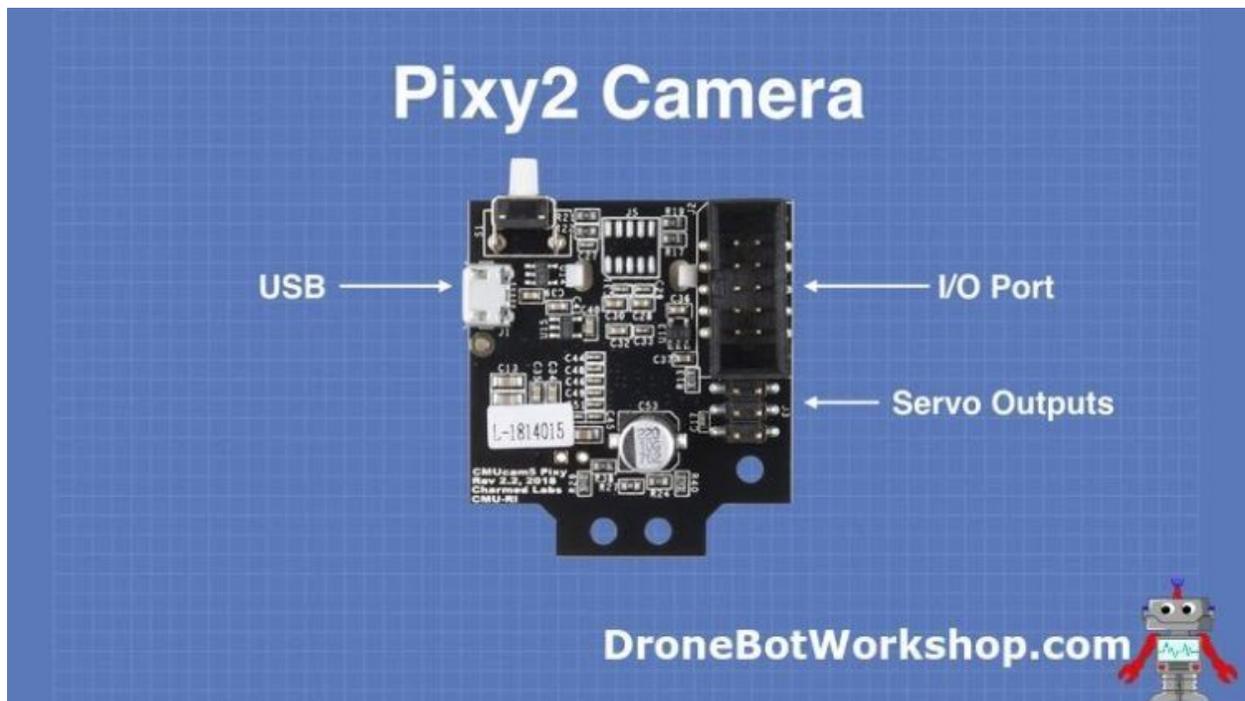
- USB connection.
- Serial UART connection.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

- I2C bus connection.
- SPI Bus connection
- Analog output.
- Digital output.

The device uses a 5-volt power supply that can be derived from the USB connection. It can also be powered by a regulated voltage source of 5 volts or an unregulated source of 6 to 10 volts (the device has an onboard voltage regulator). It consumes about 140mA at 5 volts.

The Pixy2 also has connections for two servo motors. This is for use with the optional pan and tilt assembly that can be attached to the Pixy2.



One nice thing about the Pixy2 (and there are many “nice things” about it) is that it comes complete with the following accessories:

- High quality micro USB cable.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

- Several mounting brackets, allowing you a number of options for mounting the camera.
- A premade cable for connecting the Pixy2 to an Arduino using the SPI connector.

The camera and the accessories all come packaged in a small box. No documentation is included in the box, instead a link to the Pixy2 website is printed on the package interior.



Pixy2 Wiki

A wealth of information is found on the [Pixy2 Wiki](#). You'll find information regarding hooking up the camera to Arduino, Raspberry Pi and BeagleBone Black, assembling the pan and tilt mount and of course learning to train and use the Pixy2.

The Wiki also has a link to the [Pixy2 Downloads](#) page. This is where you can download sample code and libraries for the camera, as well as firmware updates for the Pixy2 itself. You can also download images of the simple barcodes that you can use with the camera.

Charmed Labs also maintains a [Pixy2 page on GitHub](#) with scripts and binaries.

PixyMon

One software product you will certainly want to download is called PixyMon. In fact it's probably the very first thing you should do after you open up the box with your Pixy2.

PixyMon is software that runs on your computer and allows you to configure the Pixy2, train it to recognize objects and monitor and debug your own programs. It also provides a monitor that allows you to see that the camera is "looking at".

PixyMon is available for Windows, Linux and Mac OS X.

Installing PixyMon

The first step is of course to download the version of PixyMon suitable for your computer, you can get that on the [Pixy2 Download page](#).

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

The wiki provides detailed instructions for installing PixyMon, the following links will take you to the specific instructions for your computer:

- [Windows Vista, 7, 8 or 10](#)
- [Windows XP](#) (believe it or not!)
- [Mac OS X](#)
- [Linux](#) (the procedure shows Linux Mint but it also applies to other distributions like Raspbian and Ubuntu)

Once you have installed PixyMon connect the Pixy2 to the computer via the USB port.

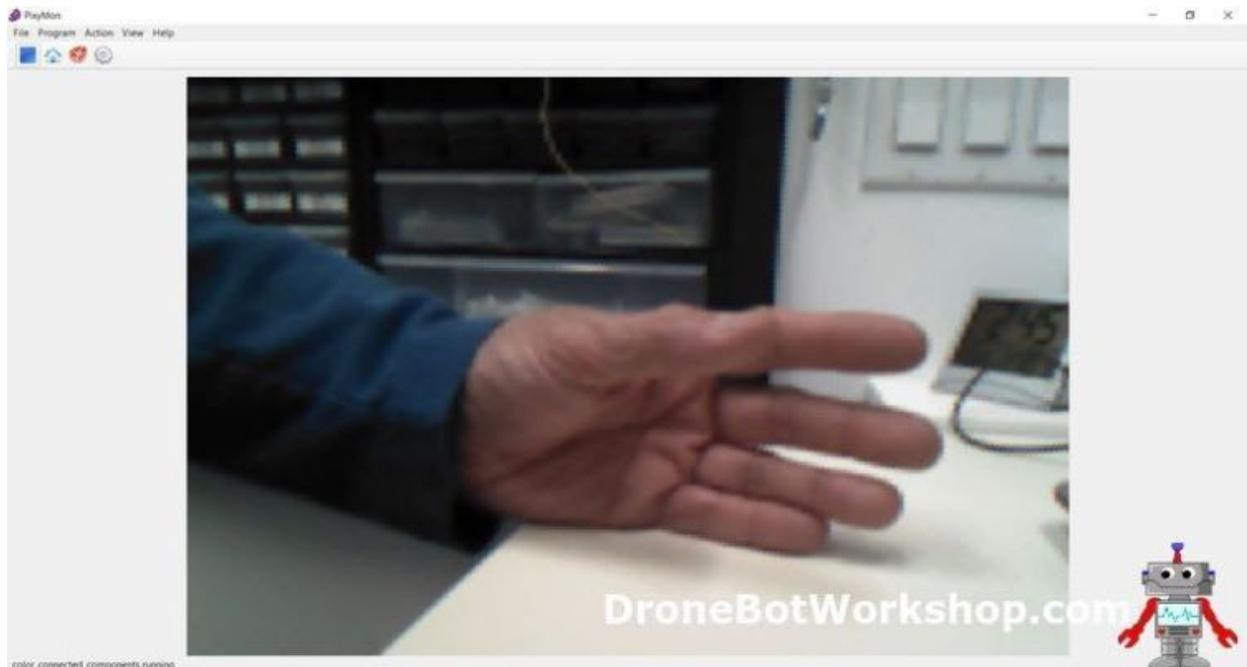
Unless you have a specific reason for doing so you should use the USB cable that was supplied with your camera, cheaper USB cables often use inferior connectors and thinner wire and may not provide the Pixy2 with the proper voltage it required to operate. This is especially important if you are using the optional pan and tilt servos as they require additional current.

Using PixyMon

The first time you connect the Pixy2 to your computer and start PixyMon you may be prompted to download and install a firmware update for your camera. Go ahead and do that so that you have the latest version of the code.

Windows users may also need to wait while some drivers for the Pixy2 USB connection are installed. This will only take a minute or so and only needs to be done once.

After installing any updates and drivers PixyMon will open.



When PixyMon is opened it will default to the Color Connected Components (CCC) mode. In this mode PixyMon will display the video image from the Pixy2 along with outlines and coordinates of any objects it has been trained to detect.

Of course when you first use the program with a new Pixy2 there won't be any objects outlined as it hasn't been trained yet! We will address this shortly.

Aside from the rather obvious display screen PixyMon has a number of features accessible using the menu at the top of the display. These menu selections change depending upon which Program Mode that the Pixy2 is currently running. There are four program modes:

- **Color Connected Components** – In this program mode the Pixy2 is detecting object “signatures” or Color Codes.
- **Line Tracking** – In this program mode the Pixy2 is detecting lines, intersections and barcodes.
- **Pan and Tilt Demo** – A demonstration program for the optional pan and tilt assembly.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

- **Video** – In this mode the Pixy2 functions like a standard video camera.

You can use PixyMon to put the Pixy2 into any of the above program modes. You can also change the Pixy2 program mode programmatically within your sketch or program.

PixyMon can be used by itself to control the Pixy2. It can also be used in tandem with a microcontroller connected to one of the Pixy2 I/O connections (i.e. I2C, SPI). This allows you to use PixyMon as monitor or as a debug tool, very handy if your Arduino sketch does not perform the way you intended it to.

PixYmon also allows you to configure the parameters on the Pixy2. Settings made using PixyMon are saved in a non-volatile memory on the camera.

You can also use PixyMon to save and recall different configurations. So you can train the Pixy2 to recognize a number of signatures, save them and then wipe the Pixy2 signatures on the device and retrain it on other objects. Later you can load the saved configuration back into your Pixy2. This allows you to work with several groups of object signatures.

Object Recognition

Of course the main feature of the Pixy2 is its ability to recognize and track objects. Let's take a look at how this is accomplished.

The Pixy2 uses a hue or color based filtering algorithm to detect objects. So objects you wish to detect should have a distinct color. You can "fine tune" the settings to some degree but it is difficult to distinguish two objects with the same color.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

In addition to the objects hue the Pixy2 also uses a “region growing algorithm” to distinguish an object.

Training the Pixy2 to detect an object is done by assigning the object a “color signature”. The device can be trained to memorize up to seven color signatures.

You can also train the Pixy2 to remember up to seven additional “color codes”. A color code consists of two colors, a couple of pieces of colored tape can make a good color signature. This can be useful in robotics applications, you can place “color signature signs” at specific locations to allow your robot to take action once it “sees” them.

There are two ways to train your Pixy2 to recognize objects – manually and with PixyMon. All object recognition is performed in Color Connected Components mode.

Capturing Signatures Manually

The Pixy2 is capable of being trained in a stand-alone mode without being connected to any computer or microcontroller. All you need to do is provide a source of power for the camera.

The push button on the top of the Pixy2 is used along with the RGB LED to train the device.

Power up the Pixy2 and observe the status of the RGB LED, located near the bottom on the front of the camera. This indicator is the key to training the device manually.

When the camera is powered up the RGB LED will go through a series of flashes. This is the initialization sequence and you’ll need to wait while it runs (it only takes a few seconds). After initialization the indicator will turn off. You are now ready to train the Pixy2.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

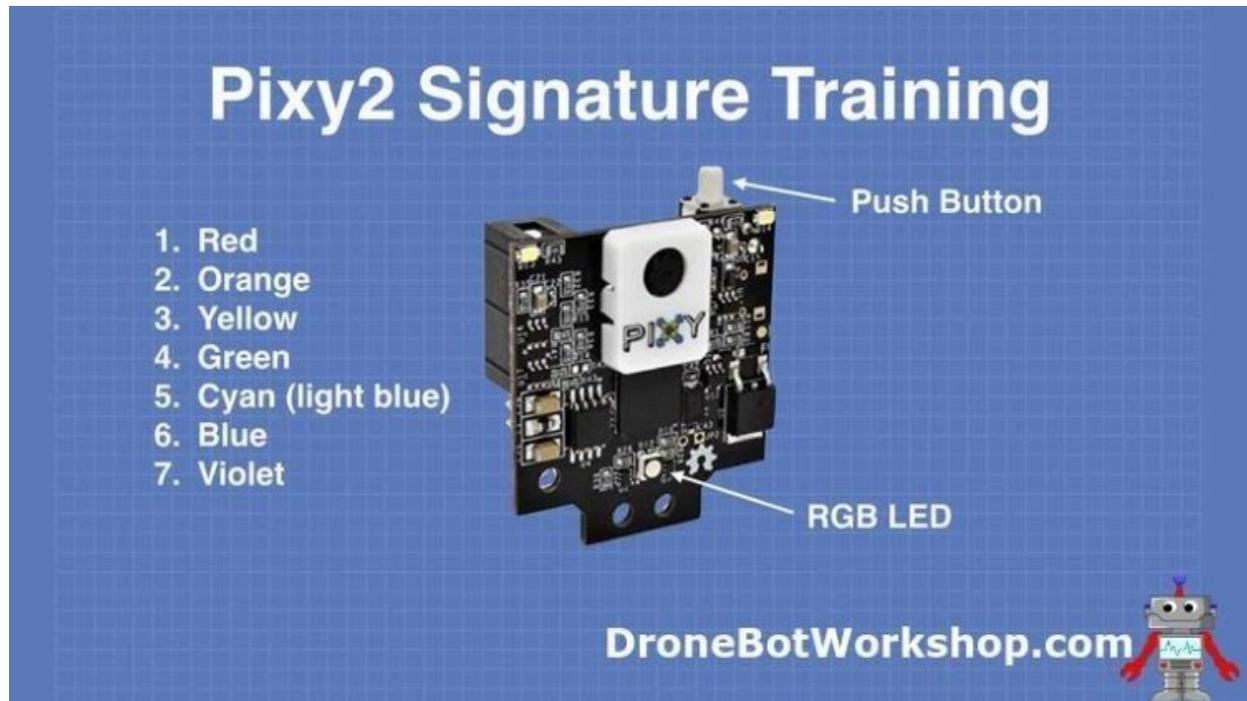
Find a suitable object to train the Pixy2 to detect. A good candidate object will have a distinct color, I used a yellow golf ball as my first object and it worked pretty well. Place the object between 20 – 50 cm (6 – 20 inches) in front of the camera lens.

While you can train the Pixy2 in stand-alone mode it is a good idea to have PixyMon running when you are new to setting color signatures. The monitor will make it easier to be sure you have your camera locking onto the correct object. Once you become more adept at training the device you can dispense with PixyMon and do your training in stand-alone mode.

So with your object in place and PixyMon running to monitor your progress it's time to train the Pixy2 to detect its first object!

Start by pressing down and holding the push button. After about a second the RGB LED should glow white. Continue to hold down the button until it glows red, then immediately release the push button.

The red indicates that you are setting the color signature for signature number 1. If you were to continue holding down the button it will cycle through seven different colors, each one representing one of the color signatures.



Once you release the push button for signature number 1 (or any of the other six signatures) the Pixy2 will be in “light pipe” mode. The RGB LED will now glow in a color that approximates the color of the object you are trying to train the camera to recognize.

Observe the RGB LED and the video screen on PixyMon.

When the object is picked up by the Pixy2 you will see a grid pattern engulfing it. Move the object around until the grid covers it completely, or as much as possible, without picking up on extraneous objects (like your fingers for example). At that point the RGB LED should be illuminated in a color similar to the target object.

Once you are satisfied that you are locked onto the object press and release the push button. The object will now be assigned to the color signature you were training it for.

You have now trained the Pixy2 to recognize its first object!

Capturing Signatures with PixyMon

Another way to train the Pixy2 to recognize objects and assign signatures to them is to use PixyMon.

Once again you'll need to place the object you want to recognize in front of the camera. In PixyMon be sure you are in Color Connected Components mode, which is the default mode when you first start the program.

Once the object is visible in your monitor go to the *Action* menu and choose one of the "Set Signature" selections, i.e. *Set Signature 1*.

Now use your mouse and click and drag to outline the object you are trying to recognize.



Once you have selected the area release the mouse button. PixyMon will now have learned the object. It's as simple as that!

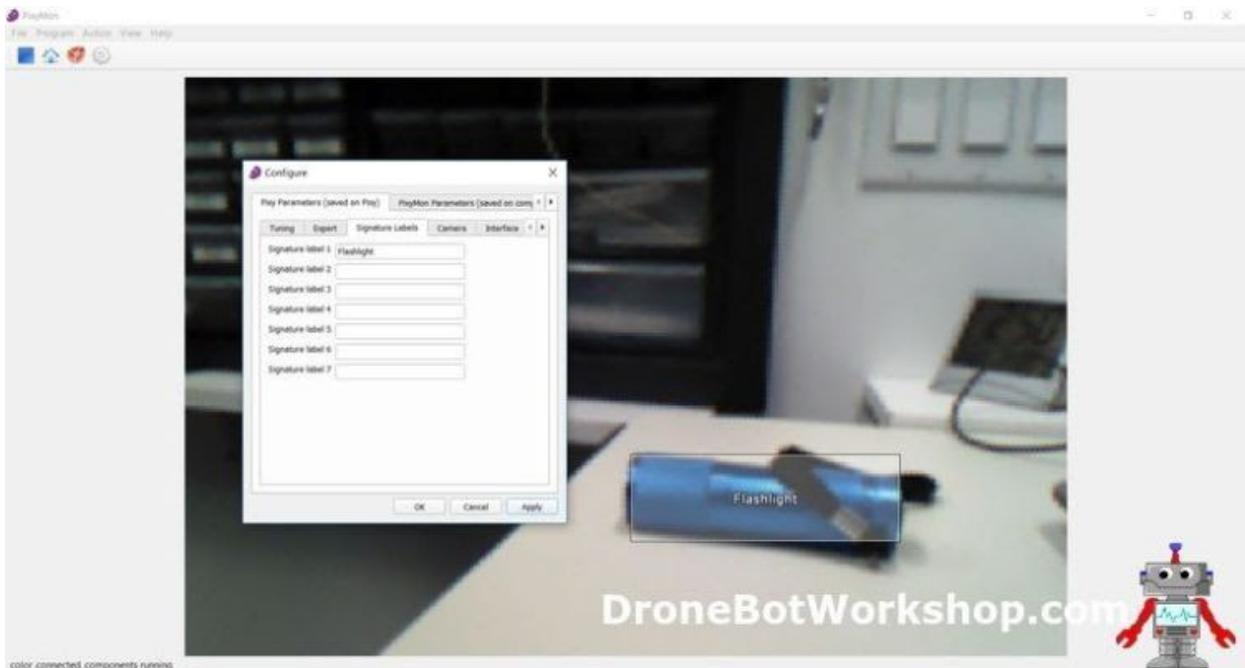
Labels and Fine Tuning

When you train the Pixy2 to recognize an object it will be displayed in PixyMon with a label that indicates its signature number. For example, the object assigned to signature number 1 will have “s=1” printed in its display block.

You can change the label to something that makes more sense. For example, I trained my Pixy2 to recognize a flashlight, which I then labeled “flashlight” (what else?).

You may also want to “fine tune” the camera to recognize your object better or to prevent similar colored objects from falsely registering.

To fine tune the Pixy2 or change the signature labels open the *File* menu and choose *Configure*. This will open the configuration menu.



For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

The configuration menu has two main tabs. One tab (the one that is active by default) allows you to configure settings on the Pixy2 itself, the second tab is for configuring PixyMon.

Within the *Pixy Parameters* tab there are a number of sub-tabs:

- **Tuning** – This tab has a number of sliders that allow you to “fine tune” the signatures. It is very useful to prevent false triggering. The sliders are quite sensitive so a little bit goes a long way!
- **Expert** – As the name implies this tab has a number of advanced parameters.
- **Signature Labels** – This is the tab we are looking for. Here you can assign proper names to the objects you have trained your Pixy2 to detect.
- **Camera** – Parameters specific to the camera on the Pixy2.
- **Interface** – This allows you to set the default interface (i.e. I2C, SPI) for the Pixy2. Note that the micro USB port can still be used for PixyMon regardless of the interface settings.
- **Servos** – Parameters for the two servo motors used in the optional pan and tilt assembly.

The [Pixy2 wiki](#) has a detailed description of using the controls to fine tune the Pixy2.

Line Tracking

As I mentioned earlier the Pixy2 can also be used in line following robot applications. It has several advantages over traditional line followers:

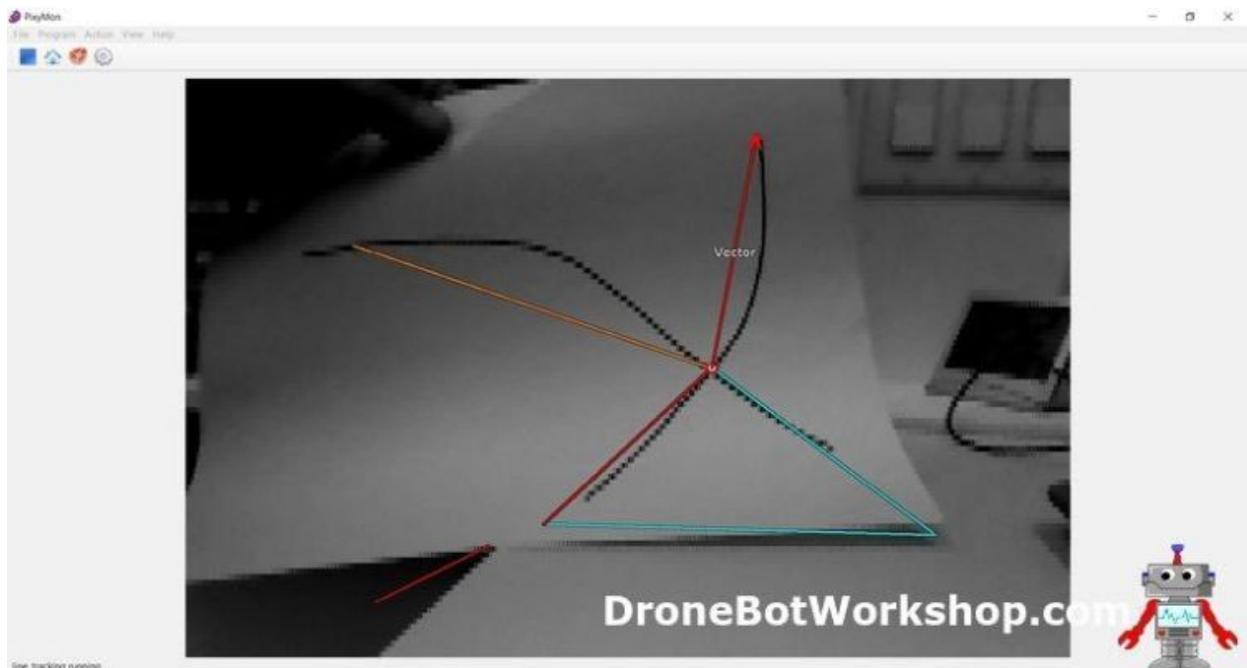
- It can “look ahead” to anticipate upcoming turns.
- It can decide what to do when it encounters an intersection.

It can also be used in combination with its barcode detection capabilities to read “road signs” and guide itself around a course with several intersections and lines.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

Pixy2's algorithms can detect dark lines on a light background or vice-versa. You may also specify the thickness of the line in your program, allowing the Pixy2 to ignore other lines.

The line that you are following is called a "vector". The Pixy2 will determine where the vector starts and stops in each frame and relays that information back to the controller or microcontroller connected to it.



If you are running the Pixy2 from PixyMon you can use the *Program* menu to put the device into line tracking mode. Just open the menu and select *line_tracking*. You can also place the Pixy2 into line tracking mode from your program or sketch.

In some cases you will get better results if you activate the built-in LEDs to illuminate the surface with the line. The Pixy2 has two white LEDs plus the RGB LED that are used to shed a bit of light onto the subject. They can be toggled on and off using the *Action* menu in PixyMon, this can also be done programmatically.

Intersections

In addition to lines the Pixy2 can also detect intersections. The device is capable of handling 3, 4 or 5-way intersections.

Once your robot encounters an intersection it will need to know what to do. This is a dilemma for designers of traditional line followers, especially at a 3-way intersection where a decision needs to be made about how to proceed.

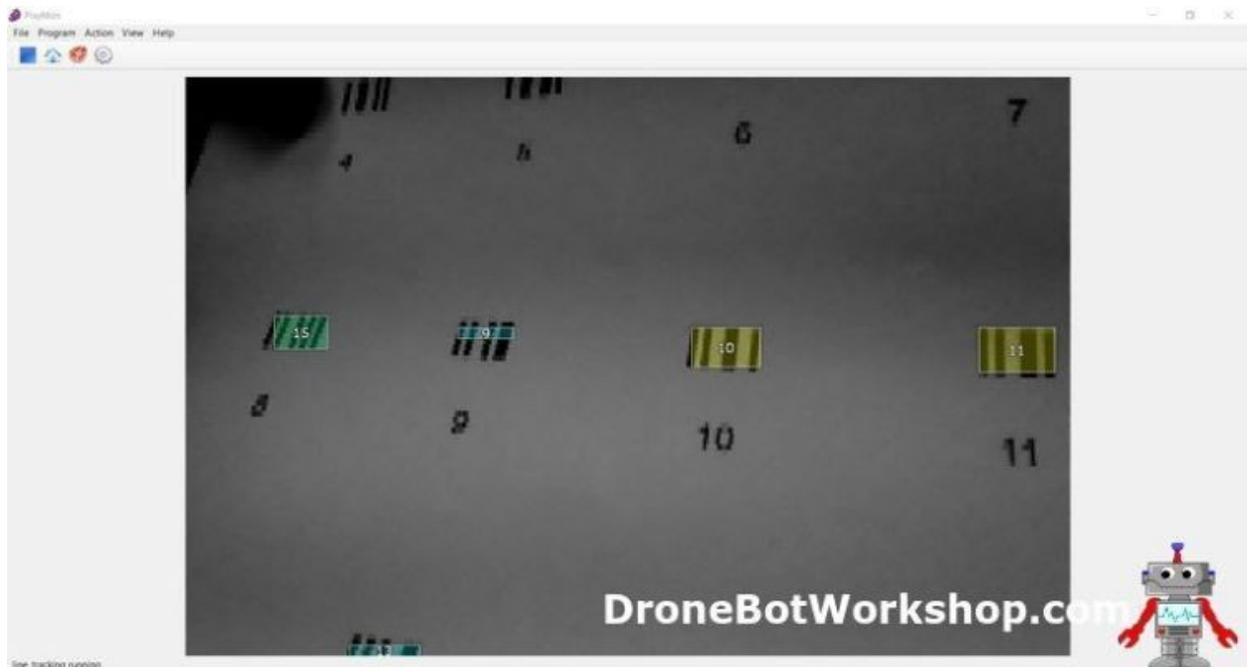
The Pixy2 can resolve this without any assistance from your program (although you can certainly assist it if you prefer). The device will proceed along what it sees as the “default” branch of the intersection.

Out of the box the “default” branch is the “straightest” branch, in other words the branch closest to zero degrees. But if you wish you can change that using the *File-Configure* menu, under the *Expert* tab. So if you set the default to be 90 degrees the device will turn left when it encounters an intersection, to default to right you would set it at -90 degrees.

Barcode Detection

Another capability of the Pixy2 is detecting barcodes.

Now don't get the idea that the device is a full-fledged barcode reader, it isn't. It is capable of reading [16 simple barcodes](#) while it is in line tracking mode.



You can use these barcodes as “street signs” to direct your robot to make a turn, slow down or stop. You can add human readable signs to these if you wish as long as you don’t obscure the barcode. Of course you can also use these barcodes for custom applications.

The Pixy2 will return the barcode number (1-16) and position in the frame to your code.

Connecting Pixy2 to Arduino

Using the Pixy2 with PixyMon is fun and educational, however I’m sure you’ll want to go further and hook the device up to an Arduino or other microcontroller..

There are a number of ways that you can connect a Pixy2 to an Arduino thanks to the multitude of interface options the camera has.

If you want to connect multiple Pixy2’s to your Arduino then you’ll probably want to use the I2C bus. You will need to use PixyMon to configure the I2C address of the second

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

(and third etc) Pixy2 as each device on the bus needs a distinct address. You could use the same technique with other microcontrollers and microcomputers as well.

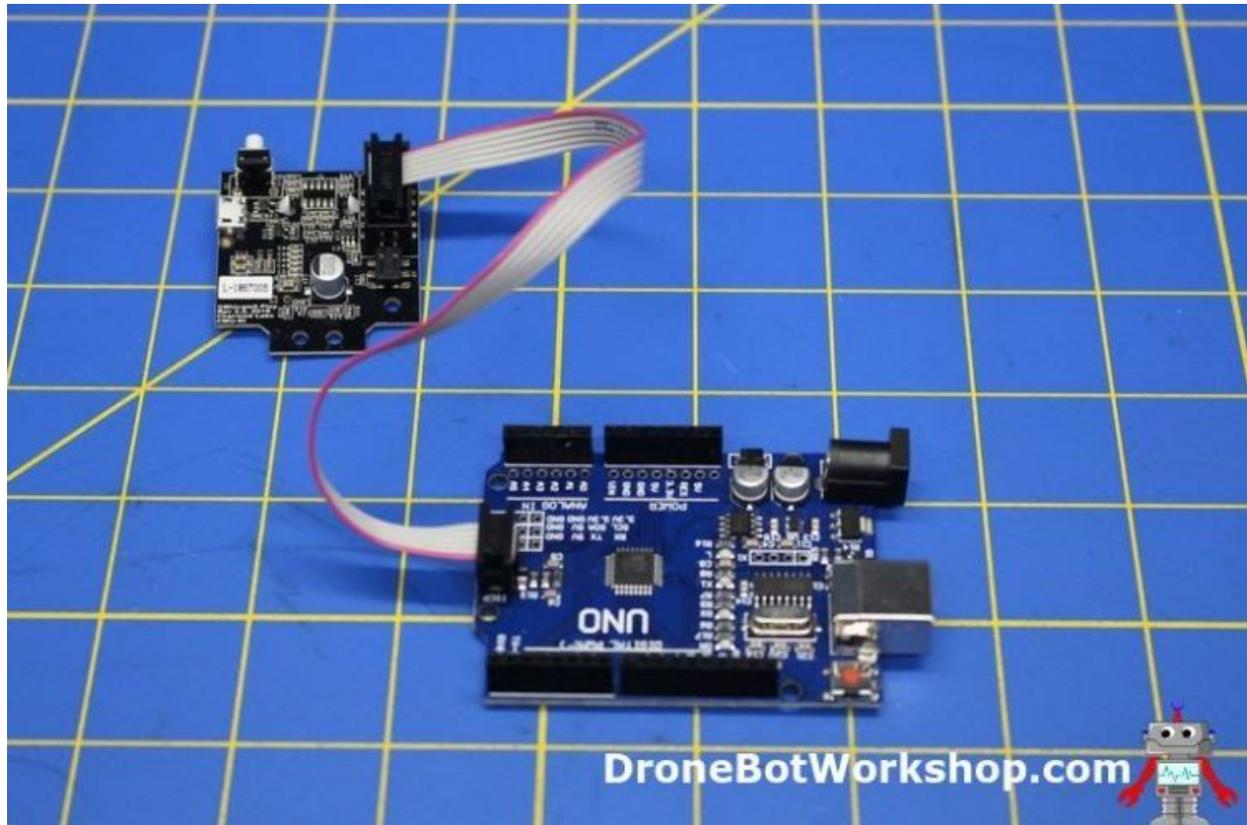
The USB port is a great option for connecting the Pixy2 to a Raspberry Pi or other Linux-based microcomputer. It is the fastest interface supported by the camera. But it isn't really an good option for the Arduino, USB requires too much memory overhead to be practical and using the USB port would prevent you from simultaneously using PixyMon as a debug tool.

SPI Connection Cable

If you just want to connect a single Pixy2 to your Arduino the easiest method is to use the SPI bus. It's actually faster than the I2C bus and when you have a device capable of processing 60 frames of video per second speed is an important factor.

The Arduino SPI bus has a clock rate of 2 MHz, allowing you to transfer data at 2 Mbits/second. Using this connection you can get back data from the camera and can also control the two servo motors used in the optional pan and tilt mount.

The Pixy2 is packaged with a premade ribbon cable designed specifically for connecting it to an Arduino SPI bus connector. This is the 6-pin (2 x 3) ICSP connector mounted on the Arduino Uno, Nano and Mega boards.



Note the orientation of the cable. On the Uno and Mega the cable protrudes outwards, in other words the wires point away from the circuit board. On the Nano the connector is mounted in the opposite direction.

The Pixy2 needs to be configured to use the SPI port with the Arduino. By default this is the setting it is shipped with so no configuration on your part is required, however if you've changed the settings you'll need to reset them.

To setup the interface go to PixyMon and select the *Edit – Configure* menu. Under the *Pixy Parameters* tab go to the *Interface* subtab.

The first selection in the *Interface* tab is a drop-down to select Data out port. To use your Pixy2 with an Arduino this needs to be set to *Arduino ICSP SPI*. In this

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

configuration the Pixy2 will use the SPI port without a slave select which is what is the Arduino requires.

Arduino Library

Using the Pixy2 with the Arduino is very easy thanks to the availability of a custom library. You can get the latest version of this library from the [Pixy2 Downloads](#) page.

The library is in a ZIP file that you will need to install into your [Arduino IDE](#). You can do this as follows:

1. Open the Arduino IDE.
2. Open the *Sketch* menu.
3. Click on *Include Library* to open the library submenu.
4. Select *Add ZIP Library*.
5. Use the dialog box to select the Pixy2 library ZIP file.
6. The library will be installed in your Arduino IDE.

Once the library is installed there will be some sample code loaded into the *Examples* menu. We will look at two of these sketches now.

Arduino Sketch – CCC Hello World

The first example sketch we will examine is the *ccc_hello_world* sketch. You can load this by opening the *File* menu in the Arduino IDE, selecting the *Examples* submenu and scrolling down until you get the the *Pixy2* section.


```
2 {
3   Serial.begin(115200);
2   Serial.print("Starting...\n");
4
2
5   pixy.init();
2
6 }
2
7 void loop()
2
8 {
2   int i;
9   // grab blocks!
3   pixy.ccc.getBlocks();
0
3
1   // If there are detect blocks, print them!
3   if (pixy.ccc.numBlocks)
2   {
3     Serial.print("Detected ");
3
3     Serial.println(pixy.ccc.numBlocks);
3
4     for (i=0; i<pixy.ccc.numBlocks; i++)
3     {
5       Serial.print("  block ");
3
6       Serial.print(i);
3
3       Serial.print(": ");
7
3       pixy.ccc.blocks[i].print();
3
8     }
3
9   }
4
0 }
```

```
4
1
4
2
4
3
4
4
4
5
4
6
4
7
4
8
4
9
5
0
5
1
5
2
5
3
5
4
5
5
```

This sketch demonstrates how to read data from the Pixy2 when it is in Color Connected Components mode, which is the default mode. You will need to have your Pixy2 trained to detect a few objects before you run the sketch.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

The sketch starts by including the Pixy2 library. Next an object called “pixy” is defined using the library.

In the setup section the serial monitor is initialized. Note that it is set to run at 115200 Bps instead of the default 9600 Bps. You will need to be sure to set your serial monitor for this speed, the higher speed is used to accommodate the high data transfer rate from the Pixy2 to the Arduino.

The Pixy2 is also initialized in the setup section.

In the loop an integer called “i” is defined, this will be used as a counter.

The [getBlock](#) function is used to get all of the blocks detected by the Pixy2. A “block” is another term for an object that has been assigned a color signature.

The *numBlocks* property is used to determine if there are indeed any blocks in the current frame. If there are detected blocks they are printed to the serial monitor using the `blocks[]` array.

The loop then ends and the sequence repeats itself.

Run the sketch and observe the output on the serial monitor. Assuming your Pixy2 is detecting objects (blocks) you should see the contents of the `blocks[]` array. The array has several elements:

- **Block** – this is the block position in the array, starting at zero.
- **Sig** – The signature number, from 1 to 7. Note that a number will be returned even if you labeled the individual signatures using PixyMon.
- **X and Y** – The coordinates of the block within the visible frame.
- **Width** – The block width.
- **Height** – The block height.
- **Index** – A unique number assigned to an object on the display. This allows you to track objects even when there are a multitude of identical objects.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

- **Age** – A number from 1 to 255. The “age” is the number of frames this particular block has been seen in, this allows you to form a “history” of the object.

While this is a simple example it does illustrate how the Pixy2 library makes processing data very easy. You can use it as a sample for your own code.

Arduino Sketch – Line Hello World

A second example is the `line_hello_world` sketch which is also found within the Pixy2 sample sketches.

```
1 //
2 // begin license header
3 //
4 // This file is part of Pixy CMUcam5 or "Pixy" for short
5 //
6 // All Pixy source code is provided under the terms of the
7 // GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).
8 // Those wishing to use Pixy source code, software and/or
9 // technologies under different licensing terms should contact us at
10 // cmucam@cs.cmu.edu. Such licensing terms are available for
11 // all portions of the Pixy codebase presented here.
12 //
13 // end license header
14 //
15
16 #include <Pixy2.h>
17
18 Pixy2 pixy;
19
20 void setup()
21 {
22     Serial.begin(115200);
23     Serial.print("Starting...\n");
24
25     pixy.init();
26
27     // change to the line_tracking program. Note, changeProg can use partial
28     // strings, so for example,
29     // you can change to the line_tracking program by calling changeProg("line")
30     // instead of the whole
31     // string changeProg("line_tracking")
```

```
30   Serial.println(pixy.changeProg("line"));
31 }
32
33 void loop()
34 {
35   int8_t i;
36   char buf[128];
37   pixy.line.getMainFeatures();
38
39   if (pixy.line.numVectors)
40     pixy.line.vectors->print();
41   if (pixy.line.numIntersections)
42     pixy.line.intersections->print();
43
44   if (pixy.line.barcodes)
45     pixy.line.barcodes->print();
46
47 }
48
```

This sketch demonstrates the line tracking, intersection detection and barcode reading capabilities of the Pixy2. You don't need to train the Pixy2 before running it but it is a good idea to draw out a few lines and intersections and print out the [barcode samples document](#) to give it a proper test.

The sketch starts out exactly the same as the previous one, until you get to the setup function.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

The setup function contains an additional command after initializing the Pixy2. The *changeProg* function changes the Pixy2 into line tracking mode. This illustrates how you can do this programmatically, as you'll recall you can also change mode using PixyMon.

The loop is pretty simple. The *getMainFeatures* function is used to extract any vectors, intersections or barcodes that the Pixy2 detects. The results are printed to the serial monitor.

The output on the serial monitor is pretty simple to interpret, with the possible exception of one term – “flags”. Flags are additional pieces of information that can be extracted in advanced mode. Their display on the output simply indicates that four flags exist, but you are not told what they actually are. In most cases you can just ignore this.

Otherwise the output is read as follows:

- **Vectors** – The start and end coordinates of the vector along with its unique index number (which is useful to be sure you are following the correct line).
- **Intersections** – The coordinate for the intersection and a unique index number and angle for every line forming the intersection.
- **Barcodes** – The coordinates of the barcode plus its value.

Once again the Pixy2 library makes it very easy to extract information from the camera for use in your own sketches.

Conclusion

As you can see the Pixy2 is a very capable little camera. And I've only scratched the surface describing its capabilities in this article and video, there is a lot more that you can do with it.

For more projects and tutorials visit the DroneBot Workshop - <https://dronebotworkshop.com>

The onboard processor and the libraries of code available for the Arduino, Raspberry Pi and Beaglebone Black have simplified advanced object recognition technologies to the point where you can very easily incorporate vision into your next robotics or automation project.

Again I'd like to thank the folks at DFRobot for providing the Pixy2, I'm looking forward to using it in a few projects. I think an excellent project to add it to would be the [5 DOF Robot Arm](#) I built earlier.

I'll keep you posted!