

## Running the Pan/tilt Demo

OK, you've [assembled the pan/tilt mechanism](#) and it looks cool, but it would look even cooler if it moved. One nice thing is that you don't need a battery to move the pan/tilt -- the pan/tilt can use the USB cable for power, **but you must use a 4 ft or shorter cable**. (3 ft is common, and even better.) If you use a longer cable, there can be up to 1 volt of voltage drop along the cable, which is fine for Pixy, but the servos will malfunction. Also, before you run the pan/tilt demo, go ahead and [teach Pixy an object](#) if you haven't already.

And we should also mention that the pan/tilt base needs to be anchored to the table somehow before running the demo. Use a couple pieces of tape, if you like -- it doesn't take much, or use a binder clip to attach the pan/tilt base to a magazine. If you don't anchor the base, the base will tend to rotate under Pixy, leaving Pixy more or less stationary (and we want Pixy to move/rotate!)

There are three ways to run the pan/tilt demo.

### Running the demo by using the button press sequence

When Pixy first powers up, you'll notice that the LED turns on (white) for about 1 second, then flashes, then turns off. What's really happening is Pixy is asking you which program you want to run. Currently, Pixy ships with two programs:

1. the default tracking program that detects objects and streams them out the port you choose
2. the pan/tilt demo

So, power up Pixy, and when the LED turns on (white), hold down the button. The LED will turn red then orange, red then orange, etc. If you release the button when the LED is red, you will select the default tracking program. If you release the button when the LED is orange you will select the pan/tilt demo. So do that -- release the button when the LED is orange. The LED will flash orange a couple times which means "running program 2!"

And then Pixy should track your object. Yay!

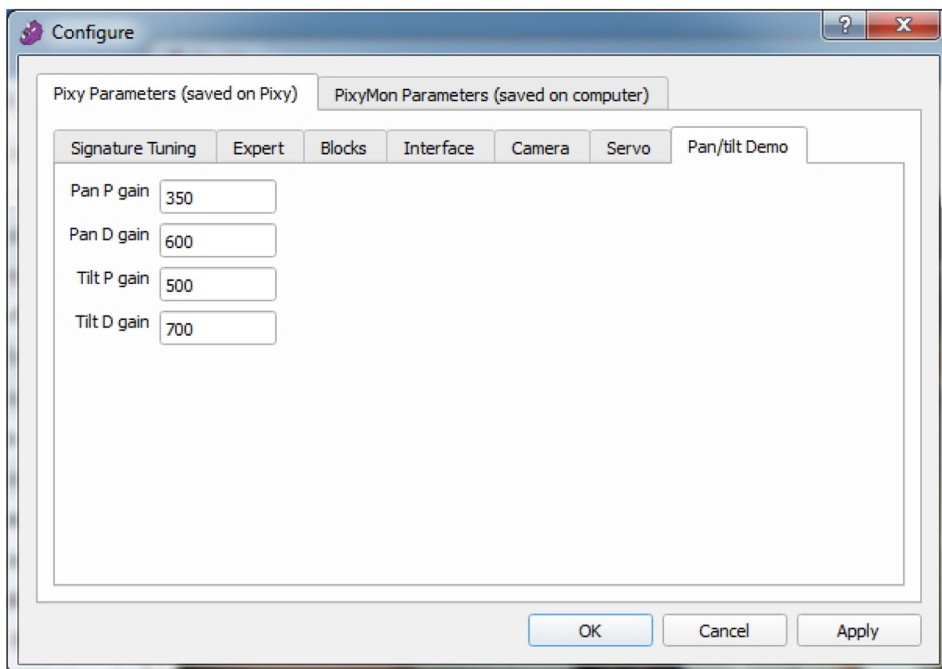
A quick note on program selection -- normally, upon power up, if you don't press the button, Pixy "times-out" and just runs the default program, which is the tracking program, but you can configure which program is the default program from the PixyMon Configure Dialog.

### Running the demo from PixyMon

If you're running PixyMon, you can run the demo (or any other program) at any time by selecting it in the **Action** menu. In particular, select **Action→Run the pan/tilt demo**.

Pixy should then track your object and plot the detected objects in the video window. Yay!

From PixyMon you can also play with the pan/tilt demo parameters.



For example, try changing the Pan P gain (pan axis proportional gain) from 500 to 50, then hit **Apply**. Or increase the gains and see what happens. Many times, increasing the gains results in some erratic behavior, but don't worry -- you won't damage anything. If you want to go back to the default values select **Action→Restore default parameters**.

### Running the demo from an Arduino

---

You can have your Arduino perform pan/tilt tracking too. The Arduino reads in the object blocks and calculates the positions of the pan and tilt servos and sends the calculated positions back to Pixy, and Pixy then updates the servo positions. This is great for getting your Pixy and Arduino better acquainted and getting them both collaborating in some kind of application that you write.

**This requires that you run the "default" Pixy program -- not the pan/tilt demo program.** The default program is the program Pixy runs upon power-up by default without pressing any buttons. You can also run the default program by selecting it in the **Actions** menu in PixyMon.

OK, start by getting the latest Arduino-Pixy code. Load the pantilt demo by selecting it in

**File→Examples→Pixy→pantilt** in the Arduino IDE. Upload it, and as long as Pixy is running the default program, you should start seeing the pan/tilt move to track any object that matches signature 1. (Note, you'll need to teach Pixy an object if you haven't done so already.)

You can also play with the Arduino code, such as changing the proportional and derivative gains.

```
ServoLoop panLoop(500, 800);  
ServoLoop tiltLoop(700, 900);
```

Here, 500 is the proportional gain and 800 is the derivative gain for the pan servo. Likewise, 700 and 900 are the proportional and derivative gains for the tilt axis. Try reducing the proportional gain in half and see what happens. Many times, increasing the gains results in some erratic behavior, but don't worry -- you won't damage anything.

With a little tweaking and modifying, this code can become a "chasing" instead of a "tracking" algorithm. That is, instead of calculating servo positions your Arduino can calculate motor speeds. So now your Pixy-Arduino robot creation can chase a ball, toy, etc.