

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. **The USB cable that came with your Pixy2 is recommended** for powering your Pixy and pan-tilt. But you can use your own, **as long as the cable is 4 ft or shorter** (3 ft is common.) 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.

We should also mention that the pan-tilt base needs to be anchored to the table somehow before running the demo. Attaching your pan-tilt to an Arduino is sufficient. Otherwise, you might try placing the pan-tilt base between two books. 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!)

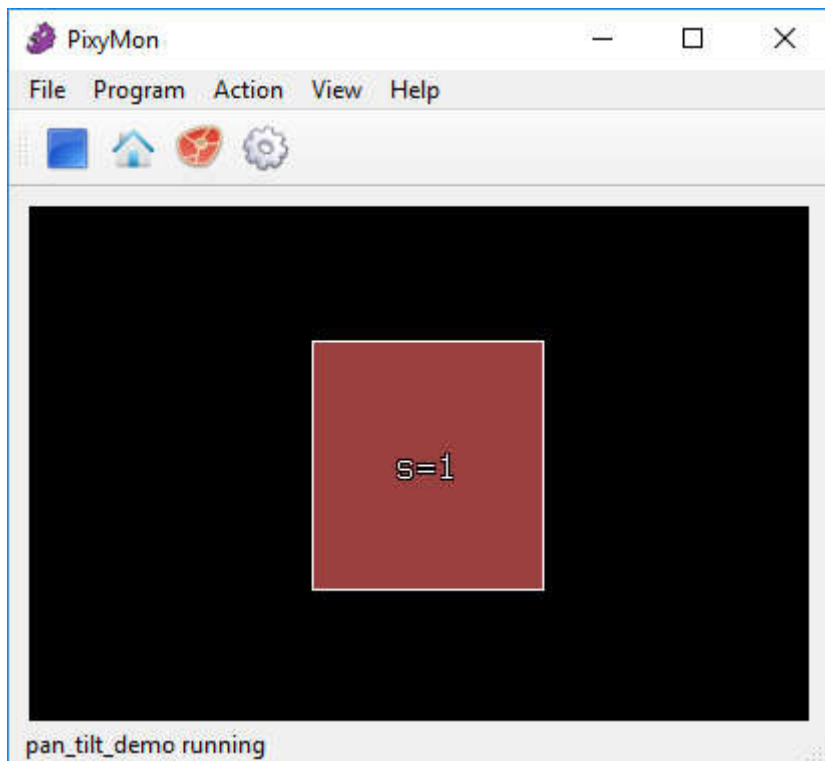
Note: if you're having trouble making the pan-tilt run properly, [here's a handy troubleshooting page](#).

There are a couple ways to run the pan/tilt demo.

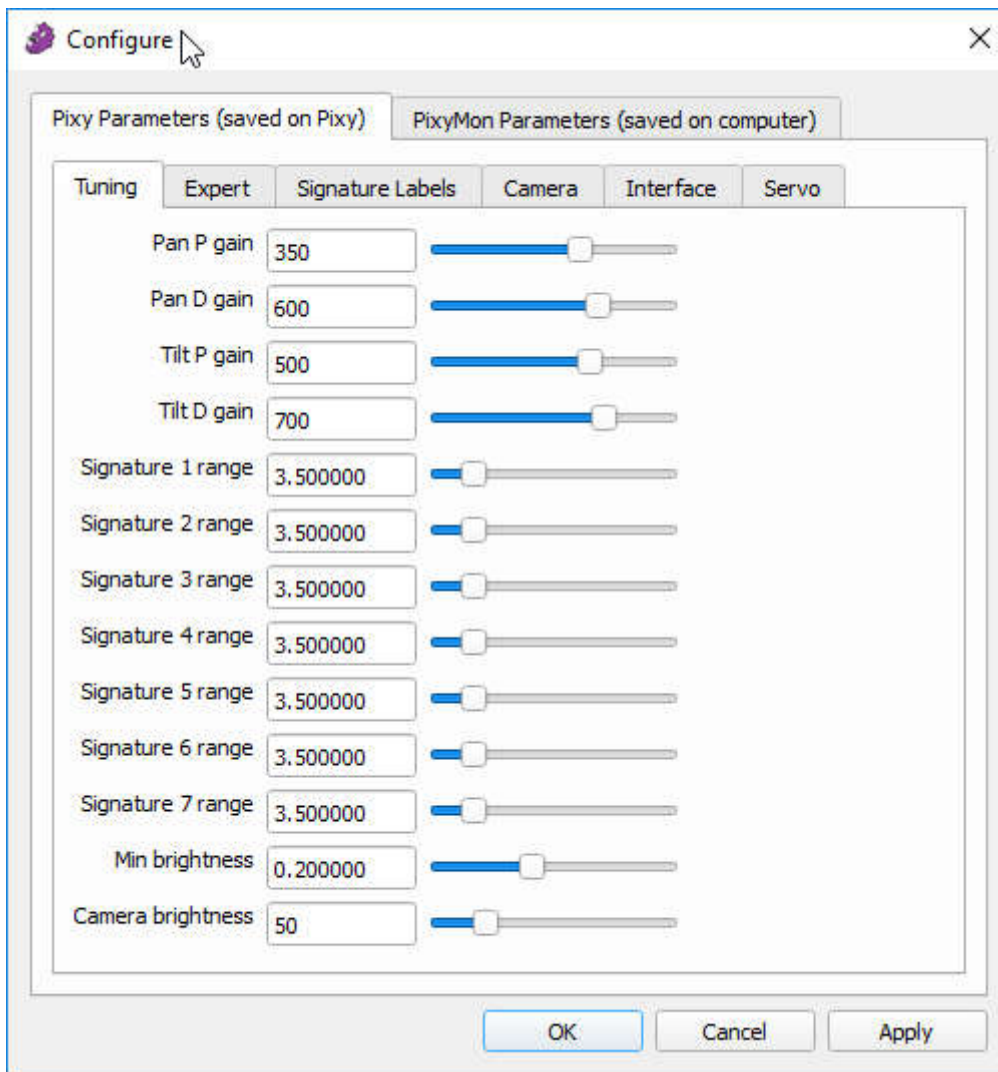
Running the demo from PixyMon

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

Pixy should then track your object and plot the detected objects in the video window. Yay! Note, PixyMon will switch to a block-only view, so the video window will turn black except for what Pixy is detecting (shown as rectangular blocks). The pan-tilt demo switches to this view because it allows Pixy to process at maximum framerate (doesn't slow Pixy down.)



From PixyMon, you can also play with the pan-tilt demo parameters:



For example, try reducing the Pan P gain (pan axis proportional gain) to 100 and see how the pan axis slows down. Or increase the P gain 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 **File→Restore default parameters**. There is more information [here](#).

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 **color_connected_components** Pixy program – not the **pan_tilt_demo** program.

OK, start by getting the latest Arduino-Pixy2 code, described [here](#). Load the pan-tilt demo by selecting it in **File→Examples→Pixy2→ccc_pantilt** in the Arduino IDE. Upload it, and as long as Pixy is running the `color_connected_components` program, you should start seeing the pan-tilt move to track the first object Pixy acquires. (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.

```
PIDLoop panLoop(500, 0, 400, true);
PIDLoop tiltLoop(600, 0, 500, true);
```

Here, 500 is the proportional gain and 400 is the derivative gain for the pan servo. Likewise, 600 and 500 are the proportional and derivative gains for the tilt axis. Try reducing the proportional gain by 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. The Arduino example **ccc_zumo_chase** does exactly this. It controls 4 axes: pan, tilt, left and right motors to chase the object it acquires.