How Pixy got started

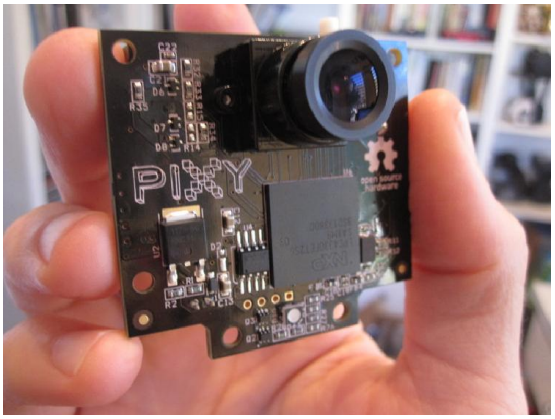
Pixy had a successful Kickstarter campaign in August, 2013. You can watch the original Kickstarter video below -- it's a good introduction!

Background

Image sensors are useful because they are so flexible. With the right algorithm, an image sensor can sense or detect practically anything. But there are two drawbacks with image sensors: 1) they output lots of data, dozens of megabytes per second, and 2) processing this amount of data can overwhelm many processors. And if the processor can keep up with the data, much of its processing power won't be available for other tasks.

Pixy addresses these problems by pairing a powerful dedicated processor with the image sensor. Pixy processes images from the image sensor and only sends the useful information (e.g. purple dinosaur detected at x=54, y=103) to your microcontroller. And it does this at frame rate (50 Hz). The information is available through one of several interfaces: UART serial, SPI, I2C, digital out, or analog out. So your Arduino or other microcontroller can talk easily with Pixy and still have plenty of CPU available for other tasks.

It's possible to hook up multiple Pixys to your microcontroller -- for example, a robot with 4 Pixys and 360 degrees of sensing. Or use Pixy without a microcontroller and use the digital or analog outputs to trigger events, switches, servos, etc.



Purple dinosaurs (and other things)

Pixy uses a hue-based color filtering algorithm to detect objects. Most of us are familiar with RGB (red, green, and blue) to represent colors. Pixy calculates the hue and saturation of each RGB pixel from the image sensor and uses these as the primary filtering parameters. The hue of an object remains largely unchanged with changes in lighting and exposure. Changes in lighting and exposure can have a frustrating effect on color filtering algorithms, causing them to break. Pixy's filtering algorithm is robust when it comes to lighting and exposure changes and significantly better than previous versions of the CMUcam.

Seven color signatures

Pixy remembers up to 7 different color signatures, which means that if you have 7 different objects with unique colors, Pixy's color filtering algorithm will have no problem identifying them. If you need more than seven, you can use color codes, which will be supported soon.

Hundreds of objects

Pixy can find literally hundreds of objects at a time. It uses a connected components algorithm to determine where one object begins and another ends. Pixy then compiles the sizes and locations of each object and reports them through one of its interfaces (e.g. SPI).

50 frames per second

What does “50 frames per second” mean? In short, it means Pixy is fast. Pixy processes an entire 640x400 image frame every 1/50th of a second (20 milliseconds). This means that you get a complete update of all detected objects' positions every 20 ms. At this rate, tracking the path of falling/bouncing ball is possible. (A ball traveling at 30 mph moves less than a foot in 20 ms.)

Teach it the objects you're interested in

Pixy is unique because you can physically teach it what you are interested in sensing. Purple dinosaur? Place the dinosaur in front of Pixy and press the button. Orange ball? Place the ball in front of Pixy and press the button. It’s easy, and it's fast.

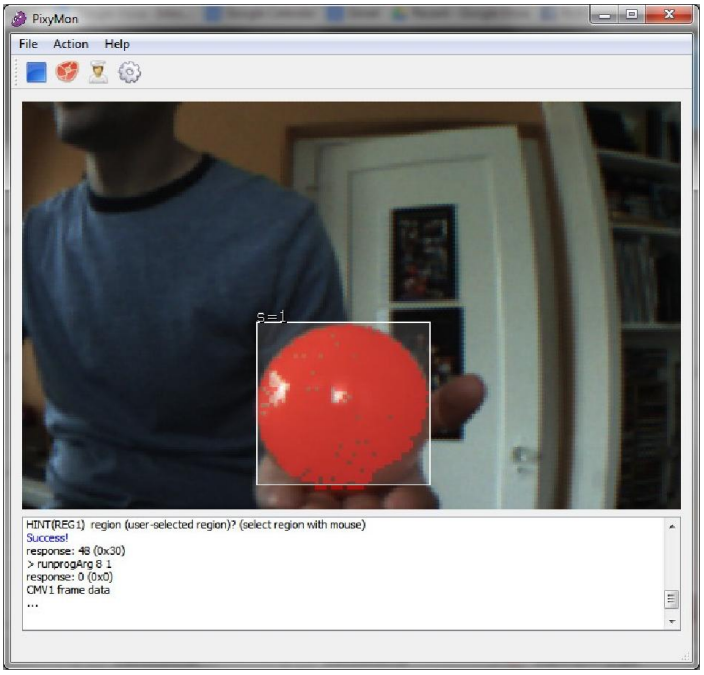
More specifically, you teach Pixy by holding the object in front of its lens while holding down the button located on top. While doing this, the RGB LED under the lens provides feedback regarding which object it is looking at directly. For example, the LED turns orange when an orange ball is placed directly in front of Pixy. Release the button and Pixy generates a statistical model of the colors contained in the object and stores them in flash. It will then use this statistical model to find objects with similar color signatures in its frame from then on.

Pixy can learn seven color signatures, numbered 1-7. Color signature 1 is the default signature. To teach Pixy the other signatures (2-7) requires a simple button pressing sequence.

PixyMon lets you see what Pixy sees

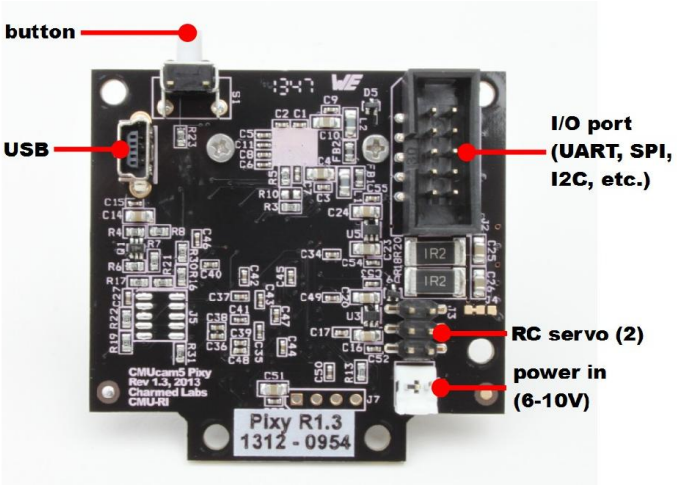
PixyMon is an application that runs on your PC or Mac. It allows you to see what Pixy sees, either as raw or processed video. It also allows you to configure your Pixy, set the output port and manage color signatures. PixyMon communicates with Pixy over a standard mini USB cable.

PixyMon is great for debugging your application. You can plug a USB cable into the back of Pixy and run PixyMon and then see what Pixy sees while it is hooked to your Arduino or other microcontroller – no need to unplug anything. PixyMon is open source, like everything else. It's written using the Qt framework.



Technical specs

- Processor: NXP LPC4330, 204 MHz, dual core
- Image sensor: Omnivision OV9715, 1/4", 1280x800
- Lens field-of-view: 75 degrees horizontal, 47 degrees vertical
- Lens type: standard M12 (several different types available)
- Power consumption: 140 mA typical
- Power input: USB input (5V) or unregulated input (6V to 10V)
- RAM: 264K bytes
- Flash: 1M bytes
- Available data outputs: UART serial, SPI, I2C, USB, digital, analog
- Dimensions: 2.1" x 2.0" x 1.4
- Weight: 27 grams



Design longevity

Image sensors are consumer devices – new sensors are released every year that take advantage of new process technologies, and older sensors are discontinued. It's not unusual for an image sensor to have a lifetime of only a couple years. Any device that uses an image sensor will have to contend with this issue -- there will come a day when the specific image sensor in the design will no longer be available. Previous CMUcams had this problem and had to be discontinued.

Pixy addresses this issue on several fronts:

- We chose an image sensor that is targeted for the automotive industry. Automotive products (silicon devices) have longer lifetimes.
- We chose a basic sensor with raw Bayer output. This is the most common type of sensor, and when/if we need to switch sensors, we will have a large selection to choose from, and the code changes will be minimal.
- We did not choose a camera module (sensor/lens combination). We mount the image sensor ourselves and use a standard M12 lens holder and lens. Camera modules tend to have the shortest product lifetimes because they target the fast-paced smart phone, tablet, and laptop industry. An M12 lens is also

replaceable – you can choose a different lens with a field-of-view and f-stop that works best for your application. (We chose a lens with a 75-degree horizontal field-of-view and f1.2, which is great for robotics applications, but you can easily replace.)

Color codes

Color codes are going to be in our next firmware release, estimated end of April, early May, 2014.

Face detection

Face detection will be in the release after color codes. Stay tuned! But in the meantime, Pixy is a great sensor, primarily intended to recognize objects using color cues. If you already own Pixy or find yourself in possession of one soon, all firmware updates and new functionality will be available to you.