

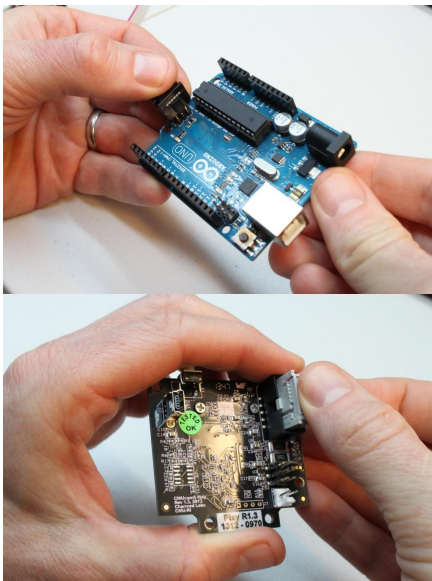
## Hooking up Pixy to a Microcontroller (like an Arduino)

Pixy is meant to talk to a microcontroller.

- If you want to hook your Pixy to a **Raspberry Pi**, go [here](#).
- If you want to hook your Pixy to a **BeagleBone Black**, go [here](#).
- If you want to hook your Pixy to something not listed here, check out our [porting guide](#).
- If you want to hook your Pixy to an **Arduino**, keep reading!

Out of the box, Pixy is ready to talk to an Arduino. It sends block information to Arduino at 1 Mbits/second, which means Pixy can send more than 6000 detected objects per second or 135 detected objects per frame (Pixy can process 50 frames per second.)

OK, to get Pixy and Arduino talking to each other, use the supplied Arduino cable to connect Pixy to your Arduino.

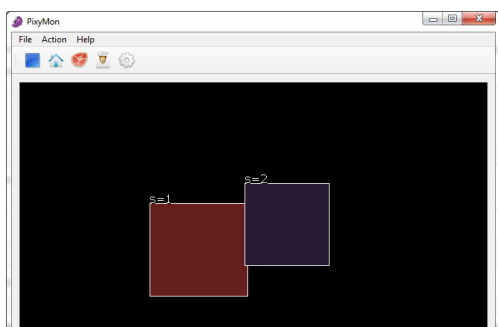


Next, download the latest Arduino library "arduino\_pixy-x.y.z.zip" [here](#). Bring up the Arduino IDE and import the Pixy library by selecting **Sketch→Import Library** in the Arduino IDE, and then browsing to the Arduino zip file that you just downloaded.

Next, load the "hello\_world" example by selecting it in **File→Examples→Pixy**. Upload it and bring up the **Serial Monitor**. You should see messages printed that look similar to this:

```
Detected 1:
  block 0: sig: 1 x: 159 y: 109 width: 61 height: 61
Detected 1:
  block 0: sig: 1 x: 173 y: 114 width: 60 height: 61
Detected 1:
  block 0: sig: 1 x: 146 y: 111 width: 70 height: 65
...
```

Note, this example will only print messages if Pixy is running the "default program" and an object that matches one of its color signatures is visible. This is what PixyMon looks like when Pixy is running the default program and it has detected objects:





## Arduino API

---

Using Pixy with Arduino is really simple. You simply include the SPI and Pixy headers:

```
#include <SPI.h>
#include <Pixy.h>
```

And make a global instance of Pixy by putting this little guy outside your `setup()` and `loop()` functions:

```
Pixy pixy;
```

The most important method in the Arduino library is `getBlocks()`, which returns the number of objects Pixy has detected. You can then look in the `pixy.blocks[]` array for information about each detected object (one array member for each detected object.) Each array member (`i`) contains the following fields:

- `pixy.blocks[i].signature` The signature number of the detected object (1-7 for normal signatures)
- `pixy.blocks[i].x` The x location of the center of the detected object (0 to 319)
- `pixy.blocks[i].y` The y location of the center of the detected object (0 to 199)
- `pixy.blocks[i].width` The width of the detected object (1 to 320)
- `pixy.blocks[i].height` The height of the detected object (1 to 200)
- `pixy.blocks[i].angle` The angle of the object detected object if the detected object is a [color code](#).
- `pixy.blocks[i].print()` A member function that prints the detected object information to the serial port

So it's simple to talk to Pixy with your Arduino! For more information on the Arduino Library and API, go [here](#).

## Updating Pixy Library for Arduino

---

Before installing a new version of the Arduino Library, it's recommended that you delete the existing library. To do this, you can go into your `C:\Users\<yourname>\Documents\Arduino\libraries` (or similar directory, `<yourname>/Documents/Arduino` in OSX and Linux) and remove the Pixy directory. Then re-run the Arduino IDE.