Search the Arduino Playground

Manuals and Curriculum
(http://playground.arduino.cc/Main/ManualsAndCurriculum)
Arduino StackExchange
(http://arduino.stackexchange.com)
Board Setup and Configuration
(http://playground.arduino.cc/Main/ArduinoCoreHardware)
Development Tools
(http://playground.arduino.cc/Main/DevelopmentTools)
Arduino on other Atmel Chips
(http://playground.arduino.cc/Main/ArduinoOnOtherAtmelChips)
Interfacing With Hardware
(http://playground.arduino.cc/Main/InterfacingWithHardware)

- Output
  (http://playground.arduino.cc/Main/InterfacingWithHardware#Output)
- Input
  (http://playground.arduino.cc/Main/InterfacingWithHardware#InputTOC)
- User Interface
  (http://playground.arduino.cc/Main/InterfacingWithHardware#ui)
- Storage
  (http://playground.arduino.cc/Main/InterfacingWithHardware#Storage)
- Communication
  (http://playground.arduino.cc/Main/InterfacingWithHardware#Communication)
- Power supplies
  (http://playground.arduino.cc/Main/IntWithHW-
  PwrSup)
- General
  (http://playground.arduino.cc/Main/InterfacingWithHardware#General)

Interfacing with Software
(http://playground.arduino.cc/Main/InterfacingWithSoftware)
User Code Library
(http://playground.arduino.cc/Main/GeneralCodeLibrary)

- Snippets and Sketches
  (http://playground.arduino.cc/Main/SketchList)
- Libraries
  (http://playground.arduino.cc/Main/LibraryList)
- Tutorials
  (http://playground.arduino.cc/Main/TutorialList)

Suggestions & Bugs
(http://code.google.com/p/arduino/issues/list)
Electronics Technique
(http://playground.arduino.cc/Main/ElectroInfoResources)
Sources for Electronic Parts
(http://playground.arduino.cc/Main/Resources)
Related Hardware and Initiatives
(http://playground.arduino.cc/Main/SimilarBoards)
Arduino People/Groups & Sites
(http://playground.arduino.cc/Main/People)
Exhibition
(http://playground.arduino.cc/Projects/ArduinoUsers)
Project Ideas
(http://playground.arduino.cc/Projects/Ideas)
Languages
(http://playground.arduino.cc/Main/Languages)

PARTICIPATE
(http://playground.arduino.cc/Main/Participate)

- Suggestions
  (http://code.google.com/p/arduino/issues/list)
- Formatting guidelines
  (http://playground.arduino.cc/Main/Participate#contribrules)
- All recent changes
  (http://playground.arduino.cc/Site/AllRecentChanges)
- PmWiki
  (http://playground.arduino.cc/PmWiki/PmWiki)
- WikiSandBox training
  (http://playground.arduino.cc/Main/WikiSandbox)
- Basic Editing
  (http://playground.arduino.cc/PmWiki/BasicEditing)
- Cookbook (addons)
  (http://www.pmwiki.org/wiki/Cookbook/CookbookBasics)
- Documentation index
  (http://www.pmwiki.org/wiki/PmWiki/DocumentationIndex)
- Drone with Arduino
  (http://www.bartoloilliano.com/arduino-tutorial-costruire-un-drone-con-webcam-telecomandato-da-pc-tramite-csharp.html)
- Thermostat with Arduino
  (http://arduinothermostat.blogspot.co.uk)

Arduino and Python

Talking to Arduino over a serial interface is pretty trivial in Python. On Unix-like systems you can read and write to the serial device as if it were a file, but there is also a wrapper library called pySerial (http://pyserial.sourceforge.net/) that works well across all operating systems.

After installing pySerial (http://pyserial.sourceforge.net/), reading data from Arduino is straightforward:

```
>>> import serial
>>> ser = serial.Serial('/dev/tty.usbserial', 9600)
>>> while True:
...     print ser.readline()
'1 Hello world!\r\n'
'2 Hello world!\r\n'
'3 Hello world!\r\n'
```

Writing data to Arduino is easy too (the following applies to Python 2.x):

```
>>> import serial # if you have not already done so
>>> ser = serial.Serial('/dev/tty.usbserial', 9600)
>>> ser.write('5')
```

In Python 3.x the strings are unicode by default. When sending data to Arduino, they have to be converted to bytes. This can be done by prefixing the string with b:

```
>>> ser.write(b'5') # prefix b is required for Python 3.x, optional for Python 2.x
```

Note that you will need to connect to the same device that you connect to from within the Arduino development environment. I created a symlink between the longer-winded device name and `/dev/tty.usbserial` to cut down on keystrokes.

It is worth noting that the example above will not work on a windows machine; the arduino serial device takes some time to load, and when a serial connection is established it resets the arduino.

Any write() commands issued before the device initialised will be lost. A robust server side script will read from the serial port until the arduino declares itself ready, and then issue write commands. Alternatively It is possible to work around this issue by simply placing a 'time.sleep(2)' call between the serial connection and the write call.

Python Arduino Prototyping API v2 (https://github.com/vascop/Python-Arduino-Proto-API-v2)

An updated version of the Arduino Prototyping API library used to quickly prototype an Arduino based application in Python.

Python Firmata (http://github.com/lupeke/python-firmata/)

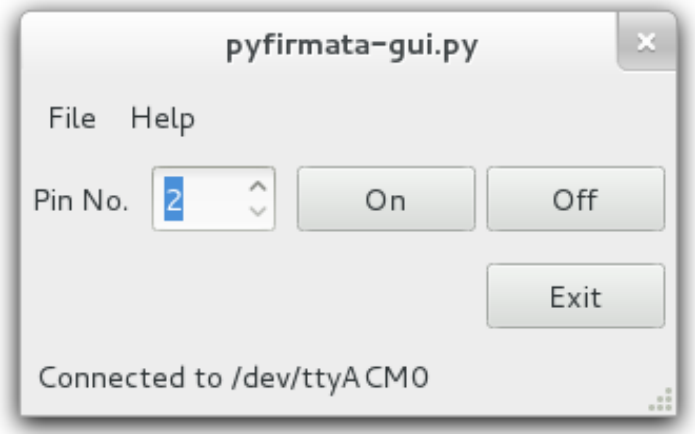
A Python API for the Firmata protocol. It's based on the Processing Library for Arduino and supports firmata v2.0

---

pyFirmata (https://github.com/tino/pyFirmata)

Python interface for the Firmata protocol. It is compliant with Firmata 2.1.

- Documentation (https://github.com/tino/pyFirmata/blob/master/README.rst)
- Examples (by Fabian) (https://bitbucket.org/fab/pyfirmata/src/96116e877527/examples)



(gui can be build with gtk, not included)

Py2B

Implements a simple 2 ASCII Chars protocol.
To start playing with Python and Arduino go to Py2B (http://playground.arduino.cc/Code/2B)

---

arduino_serial.py (https://github.com/wiseman/arduino-serial) is a library and command line that allows you to communicate with an Arduino board without installing any additional Python modules.

Turn your arduino into an IRC bot with python (http://blog.datasingularity.com/?p=148).

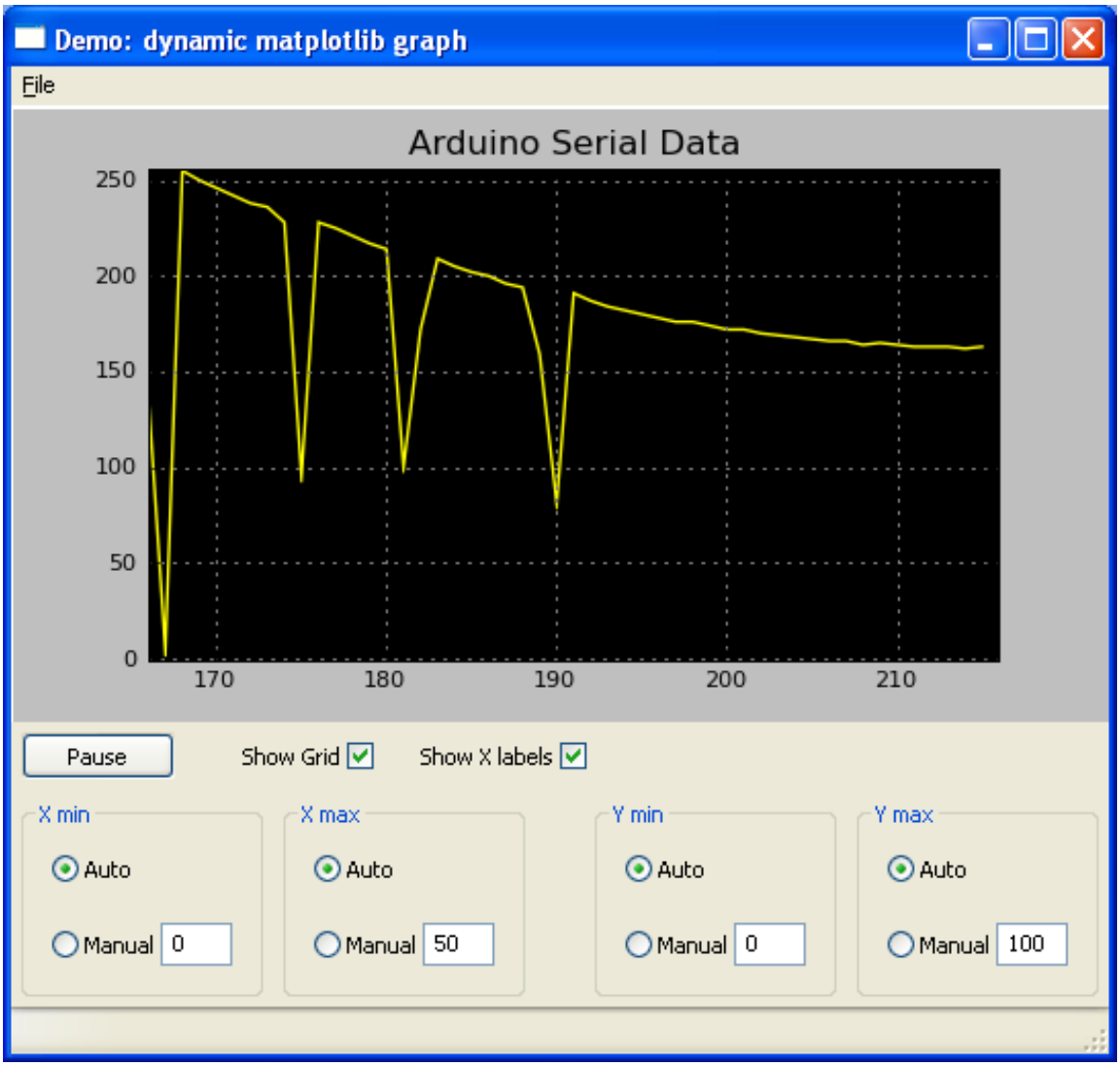
Control Arduino using HTML forms (http://blog.datasingularity.com/?p=50) via Python.

Arduino-Python 4-Axis Servo Control (http://principialabs.com/arduino-python-4-axis-servo-control/): with a good general explanation of how to coordinate Arduino and Python code.

Here is an arduino serial utility for BSDish systems (http://blog.datasingularity.com/?p=64) that is written in python.

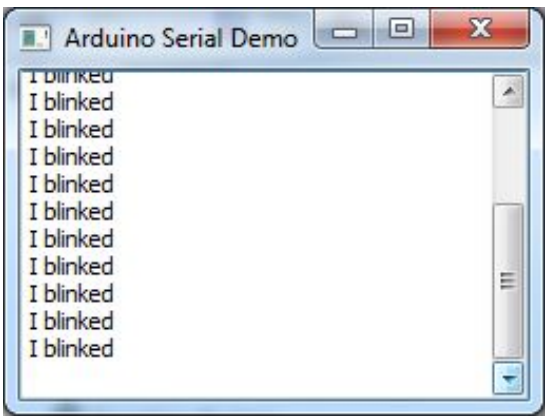
Control Arduino using this Python-SimpleMessageSystem library. (http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1199237936/0)

Realtime graphing of data over serial



Graphing realtime data from an Arduino using Python and Matplotlib:
http://www.blendedtechnologies.com/realtime-plot-of-arduino-serial-data-using-python
(http://www.blendedtechnologies.com/realtime-plot-of-arduino-serial-data-using-python)

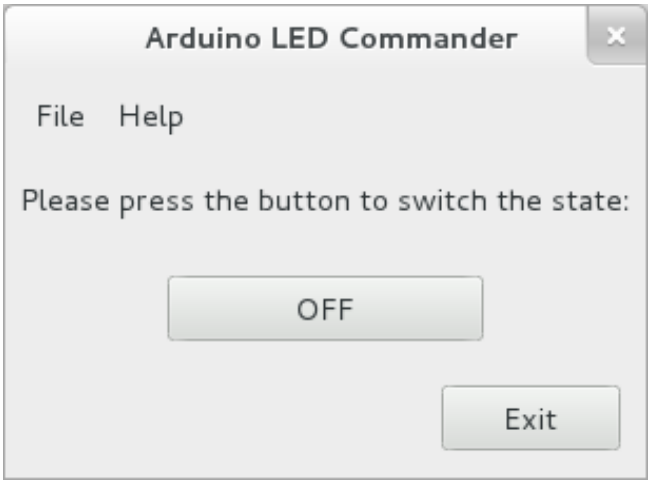
A threaded GUI example with Python and Qt/PyQt



A simple example/tutorial of a threaded GUI displaying information from the Arduino.
http://blog.wickeddevice.com/?p=191 (http://blog.wickeddevice.com/?p=191)

A simple UI with Python and GTK

Arduino LED commander is a simple GUI for controlling an output pin of the Arduino. This project was made as a little tutorial/example to show the possibility of controlling an Arduino board with Python and a GTK interface.



- Details: http://fabian-affolter.ch/blog/index.php/arduino-controlled-by-a-gtk (http://fabian-affolter.ch/blog/index.php/arduino-controlled-by-a-gtk)
- Source: http://www.gitorious.org/arduino-led (http://www.gitorious.org/arduino-led)
- Video: http://youtu.be/WZ9FKHfVKRY (http://youtu.be/WZ9FKHfVKRY)

Share

(https://plus.google.com/+Arduino) (http://www.flickr.com/photos/arduino_cc)

(http://youtube.com/arduinoteam)