***Arduino Library for Python Communications:***

readLine() – Reads 1 line of data from python and returns it as a string so you can put it in a variable or something.

writeLine(String whatYouWantToSend) – Writes 1 line of data to python.

***Python Library for Arduino Communications:***

setPort(String portName) – This tells the library what port to use. As stated in the example script, the arduino mega 2560 is on port “COM11”, while the arduino Uno is on port “COM3”. To find the port name, just go to your arduino IDE, plug in your arduino, and under “Tools”, “Port: ” it will show you.

begin(int baudrate) – Open communications to the arduino. Used the same as arduino’s own “Serial.begin(int baudrate)” command. Just make sure that python and the arduino are using the same baudrate. Set to whatever speed you want (I recommend 115200), just make sure they are the same.

readLine() – Reads 1 line of data from the arduino and returns it as a string so you can put it in a variable or something.

writeLine(String whatYouWantToSend) – Writes 1 line of data to the arduino.

***Other Info:***

The python command “arduino.begin(int baudrate)” also accepts another syntax of “arduino.begin(int baudrate, String portName)” so you don’t have to separate them into 2 separate commands. Use whichever you like more, both syntaxes does the same thing.

The python library for arduino communications is just a simple wrapper around the “serial” library. It just handles a few things for you that the “serial” library doesn’t such as string parsing and you can send/receive strings and whole sentences rather than a single character at a time.

The arduino library for python communications is just a simple wrapper around the “Serial.read()” and “Serial.print()” commands. It takes care of the same things the python library does in the same way.

The arduino and python libraries are designed to work together nicely. Using arduino’s “python.writeLine()” command and python’s “arduino.readLine()” together will work perfectly. Likewise for the other way around.