

# Arduino in IPython

## Import the Libraries

```
In [1]: from IPython.display import clear_output
        from time import sleep
        import serial
        import os
        import sys
```

## Run this prior to plugging in your Arduino to set a baseline.

```
In [2]: os.system("ls /dev/tty* > prelist.txt") # Build a list of all the currently
        connected tty devices (For linux)
```

```
Out[2]: 0
```

## Plug in the Arduino now.

```
In [3]: os.system("ls /dev/tty* > postlist.txt") # Build a list of all connected
        tty devices that includes the Arduino
        pre_list=[]
        with open('prelist.txt', 'r') as f:
            for line in f:
                pre_list.append(line)
        with open('postlist.txt', 'r') as f:
            for line in f:
                if line not in pre_list:
                    print("Your device id is {}".format(line))
                    dev = line.strip('\n') # Remove the end of line character
```

```
Your device id is /dev/ttyUSB0
```

**Copied code from [instructables.com](http://www.instructables.com/id/Interface-Python-and-Arduino-with-pySerial/step3/Program-Idle/)  
(<http://www.instructables.com/id/Interface-Python-and-Arduino-with-pySerial/step3/Program-Idle/>) works with Python 2.7**

```

from time import sleep
import serial
ser = serial.Serial('/dev/tty.usbmodem1d11', 9600) # Establish the connection on a specific port
counter = 32 # Below 32 everything in ASCII is gibberish
while True:
    counter +=1
    ser.write(str(chr(counter))) # Convert the decimal number to ASCII then send it to the Arduino
    print ser.readline() # Read the newest output from the Arduino
    sleep(.1) # Delay for one tenth of a second
    if counter == 255:
        counter = 32

```

## IPython3 Version (Slightly embellished)

- The code was slightly embellished with the use of `clear_output()`, and `stdout.flush()` found in the animation example in the official IPython Notebook

Example Collection

(<https://github.com/ipython/ipython/tree/master/examples/notebooks>).

In [4]:

```

ser = serial.Serial(dev, 9600, timeout = 0.25) # Establish the connection on a specific port
counter = 32 # Below 32 everything in ASCII is gibberish
sleep(5) # The Arduino needs 5 seconds to wake up unless the code is uploaded as hex.
back = ser.readline().decode('ascii') # Python3.* requires explicit encoding and decoding of bytes
print(back) # Should say 'Ready'
sys.stdout.flush() # This forces the output to the screen (avoid delays)
sleep(0.5) # 0.5 second delay
ser.write(b'\x00') # Send a "flush serial" signal to the arduino (see Arduino Code)
ser.flush()
while counter<65: # Send and display the ASCII characters from #32 to #65
    counter +=1
    out = str(chr(counter)).encode('ascii') # The output must be explicitly encoded in Python3.*
    clear_output() # Erase what is currently displayed (avoids a long line of output)

    # The clear_output() does cause a flicker, (the lower cells temporarily jump up).

    print("{} was sent as {} (ASCII - {})".format(out.decode('ascii'), out, counter))
    ser.write(out)
    sleep(0.1) # Give the arduino a chance to talk back
    back = ser.readline().decode('ascii')
    print(back) # Read the newest output from the Arduino

```

```
sys.stdout.flush() # Reduces the flicker
```

```
sleep(0.5) # Delay for 0.5 seconds
```

A was sent as b'A' (ASCII - 65)  
The arduino recieved 'A'

## Arduino Code

### Previously Uploaded Arduino Code

This code needs to be uploaded to the Arduino prior to running this Notebook.

```
void setup() {  
    Serial.begin(9600); // set the baud rate  
    Serial.println("Ready"); // print "Ready" once  
}  
void loop() {  
    char inByte = ' ';  
    if(Serial.available()){ // only send data back if data has been  
n sent  
        char inByte = Serial.read(); // read the incoming data  
        if (inByte != 0){  
            char echo[] = "The arduino recieved '#";  
            echo[22]=inByte; // replace the '#' with the character rec  
ieved  
            Serial.println(echo); // send the data back in a new line  
so that it is not all one long line  
        }  
        else{  
            Serial.flush(); // a call to flush the serial buffer was m  
ade  
        }  
    }  
    delay(100); // delay for 1/10 of a second  
}
```

This is slightly modified from what can be found at [instructables.com](http://www.instructables.com/id/Interface-Python-and-Arduino-with-pySerial/step2/Program-the-Arduino/)  
(<http://www.instructables.com/id/Interface-Python-and-Arduino-with-pySerial/step2/Program-the-Arduino/>).