



Clipboard Communication PC ? Pi With Python

by [russ_hensel](#) on February 9, 2015

Table of Contents

Clipboard Communication PC ? Pi With Python	1
Intro: Clipboard Communication PC ? Pi With Python	2
Step 1: Tools and Materials	2
File Downloads	3
Step 2: Try It	3
Step 3: How it Works and Comments	3
Related Instructables	4
Advertisements	4
Comments	4

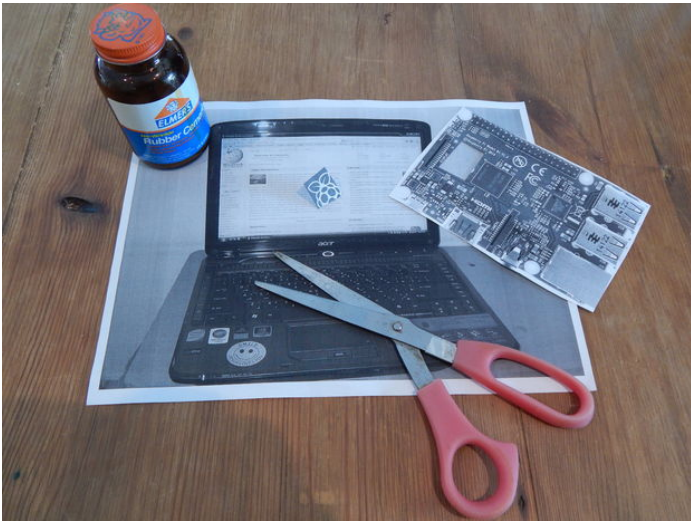


Author: russ_hensel me at one of my favorite web

For now see me at: http://www.opencircuits.com/User:Russ_hensel

Intro: ClipBoard Communication PC ? Pi With Python

I normally use my Raspberry-Pi on a windows shared desktop. I also share a network drive with the Pi. This is great, but I still have the annoyance of the clipboard on the PC not being reflected on the Pi. There may be cross platform apps out there for this problem, but I thought it would be more fun to try to make my own solution using Python. It was fun, useful, and you can learn a bit of Python from it. In the end it took only about 100 lines of code to do the job. Everything here is open source, no purchases required. This instructable does assume that you know your way around the Raspberry-Pi and have some minimal knowledge of Python. If you do not check instructables and the web where you will find lots of information. Some references are included here, but just a tiny, tiny fraction of what is out there.



Step 1: Tools and Materials

This is what I used, adapt the list to your environment, many variations are possible.

On the PC

- Windows 7, but any version should do. On the PC side Ubuntu or Mac OS should also work (not tested).
- I used Python 2.7 any version from about 2.6 up should work, I have not tried Python 3.
- I used an IDE called Spyder which I highly recommend (if you have not used a Python IDE, and do much programming then get it), but you could do it all in just a text editor or with Idle (included in most Python installations). Spyder is installed as part of Anaconda if you want it, get it at <http://docs.continuum.io/anaconda/install.html>
- You probably need to beef up your Python with rpyc and pyperclip. From the command line the following should do it: (run from cmd)

```
pip rpyc
```

```
pip pyperclip
```

- I have a shared drive with the PI, but any method that helps get files between the computers is useful.
- I also share the PI desktop with the PC, this is useful, but not necessary.

On the Pi

- I used Raspbian installed originally from NOOBS but other OS versions should work if they have Python installed.
- Again Python probably needs beefing up. The following commands did this for me: (run from a terminal)

```
sudo apt-get install python-pip
```

```
sudo pip install rpyc
```

```
sudo apt-get install xsel
```

```
sudo pip install pyperclip
```

More info

- Pyperclip at: <http://coffeeghost.net/2010/10/09/pyperclip-a-cro...>
- Rpyc at <http://rpyc.readthedocs.org/en/latest/>



File Downloads



rpc_paste.py (2 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'rpc_paste.py']



rpc_copy.py (3 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'rpc_copy.py']

Step 2: Try It

- Get the files from the instructable and download to both computers. They are **rpc_copy.py** and **rpc_paste.py**.
- Either the Pi or the PC can run either program or both at the same time, but start with one program on each.
- To make the programs work you need to know the tcp/ip address of the computer that is running **rpc_copy**. This computer is where the copying takes place, I will call it the **rpc_copy** computer.

On windows you can find the tcp/ip address from the command line (run **cmd**) with **ipconfig**.

On the PI you can find tcp/ip address by running **ifconfig** in a terminal window.

- When you have the address, code it in the program **rpc_paste.py** using what ever Python editor you use. Idle which comes with Python, is fine for this and has a menu option to launch the program when you are ready.
- On the computer where you are running **rpc_paste**, no edits are necessary.
- First start **rpc_paste** on the **rpc_paste** computer then **rpc_copy** on the other. Leave them running both places. You can start **rpc_paste** first but if it cannot connect in the first minute or so (adjustable in the code) it will give up.
- You should now be able to copy from the **rpc_copy** computer and paste to the **rpc_paste** computer.
- If you run both programs on both machines then you can copy and paste both ways.

```
112 if __name__ == '__main__':
113
114     # client is where cliips
115     print "rpc_copy: enable rpc copying from this computer"
116
117     # modify this to match the tcpip address of your remote computer which will paste
118     ip_adr      = "192.168.0.103"
119     ip_adr_smithers = "192.168.0.108"
120     ip_adr_xp      = "192.168.0.109" # xp
121     ip_adr_pi      = "192.168.0.175"
122
123
124     rpc_copy = RpcCopy( ) #
125
126
127     rval = rpc_copy.tryConnect( ip_adr_pi, times=TRY_TIMES, sleep = TRY_SLEEP )
128     if rval:
129         rpc_copy.runn()
130
131     print "rpc_copy all done"
132
133
134
135
136
```

Step 3: How it Works and Comments

One of the best way to understand how a computer program works is to read and investigate its source code. I have included fairly well commented programs you should do that. However, a brief high level overview may help. The basic technique used here is called Remote Procedure Call or RPC. It uses the Ethernet connection between the two computers to communicate. This should work between any two computers at any distance if they are on the same network and have the proper ports open. The OS should not matter as long as they support Python and the modules that extend them.

We use two modules that are often not in the distribution of Python:

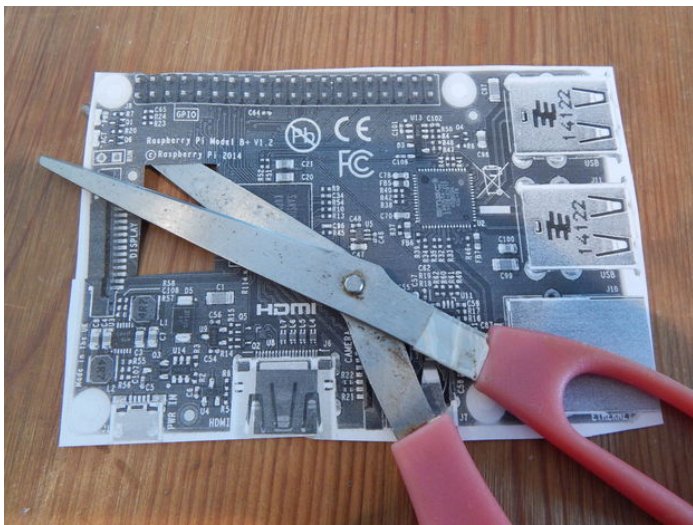
- rpyc: module supports a fairly simple RPC connection between computers.
- pyperclip: module supports access to system clipboard.

RPC from your PC to the Pi has many more uses than just clip board communication. You can fairly easily control any Python program on the Pi from your PC (and vice versa).

If you do not already use a remote connection for you Pi take a look at this. Configuring the Raspberry Pi ethernet port for remote control

<http://www.instructables.com/id/Configuring-the-Raspberry-Pi-ethernet-port-for-rem/>

<http://www.instructables.com/id/Clipboard-Communication-PC-Pi-With-Python/>



Related Instructables



Building robots with Raspberry Pi and Python
by ZRob314



iPhone remote controlled lamp
by Samuel LIU



0-5V Analog input from Raspberry Pi graphed on Web
by electronicshouse



Raspberry Pi Temperature & Humidity Network Monitor
by wpnsmith



Tweeting Bird Feeder by Reimey



Raspberry Web server sending GET data to Arduino Nano driving 6 servos
by srhnz

Comments