

Instantly share code, notes, and snippets.



turingbirds / **Agilent 82357B GPIB interface programming on Ubuntu Linux.rst**

Last active 11 months ago

<> Code

Revisions 16

Stars 7

Forks 4

Embed ▾



Download ZIP

Agilent 82357B GPIB interface programming on Ubuntu Linux

**Agilent 82357B GPIB interface programming on Ubuntu Linux.rst**

## GPIB interfacing using Agilent 82357B on Ubuntu Linux

I initially had some problems installing on my laptop, so decided to boot Ubuntu 12.04.5 LTS, 3.13.0-32-generic from USB and work from there.

For an automated installation script, see `gpib_install.sh` (and `gpib.conf`) below. The rest of this document describes the actions of the installation script step-by-step.

First, get the packages that are necessary to support Python bindings:

```
sudo apt-get update
sudo apt-get install python-dev libboost-python-dev python-setuptools
```

Download the linux-gpib package, unpack and build:

```
wget --content-disposition --no-check-certificate https://sourceforge.net/projects/linux-gpib/files/linux-gpib-3.2.20.tar.gz
tar xvfz linux-gpib-3.2.20.tar.gz
cd linux-gpib-3.2.20
./configure
make -j8
```

```
sudo make install
cd ..
```

Also download the firmware binary for the 82357B:

```
wget --content-disposition --no-check-certificate http://linux-gpib.sourceforge.net/gpib_firmware-2008-08-10.tar.gz
```

fxload is used to upload firmware to the GPIB interface.

```
wget --content-disposition --no-check-certificate https://downloads.sourceforge.net/gpib_firmware-2008-10-13.tar.gz
tar xvfz fxload-2008_10_13.tar.gz
cd fxload-2008_10_13
make
sudo make install
cd ..
```

Edit `/etc/gpib.conf` (as superuser) to fill in the correct board type:

```
interface {
    board_type = "agilent_82357a"
    name = "agi"
    ...
}
```

See the attached `gpib.conf` for the full contents of the file (other parameters were left at their default values).

Load kernel module(s):

```
sudo modprobe gpib_common
sudo modprobe agilent_82357a
```

Insert the dongle into the USB port. Only the red "FAIL" LED should be on. Find the bus and device ID:

```
lsusb
```

e.g.

```
...  
Bus 002 Device 005: ID 0957:0518 Agilent Technologies, Inc.  
...
```

Plug the found bus and device ID into the command for fxload:

```
sudo fxload -D /dev/bus/usb/002/005 -t fx2 -I gpib_firmware-2008-08-10
```

Still only the "FAIL" LED is on.

The USB bus and device ID have now changed. Wait a moment and get the new ID:

```
lsusb
```

e.g.

```
...  
Bus 002 Device 006: ID 0957:0518 Agilent Technologies, Inc.  
...
```

Run fxload again with the new bus and device ID:

```
sudo fxload -D /dev/bus/usb/002/006 -t fx2 -I gpib_firmware-2008-08-10
```

All lights should be on.

Change permissions on `/dev/gpib0` (ideally, you would manage this with a "gpib" usergroup instead):

```
sudo chmod 666 /dev/gpib0
```

Now, initialize the dongle. `gpib_config` has some trouble finding the library, so create a symbolic link first:

```
sudo ln -s /usr/local/lib/libgpib.so.0 /lib/libgpib.so.0
sudo gpib_config
```

Only the green "READY" LED should now be on.

Now, make an entry for your device in `/etc/gpib.conf`. Default HP3456B factory address is ASCII "V" (22 dec) for talk and "6" for listen.

```
device {
    name = "hp3456a"
    pad = 22
    ...
}
```

You can use `ibtest` to do some testing.

```
ibtest
```

A simple Python interface can now be made using the linux-gpib Python bindings:

```
import gpib
dev = gpib.find("hp3456a")      # corresponds to device ID in ``/etc/gpib.conf``
print gpib.read(dev, 99)
```

For more advanced applications, consider using the visa (PyVISA) libraries.

## References

[pyvisa]	PyVISA <a href="https://pyvisa.readthedocs.io/en/stable/">https://pyvisa.readthedocs.io/en/stable/</a>
----------	--

[\[linux-gpib\]](#)linux-gpib : <http://linux-gpib.sourceforge.net/>[fxload-2008\\_10\\_13.tar.gz](#)[View raw](#)[gpib.conf](#)

```

1  /*****
2      GPIB.CONF IEEE488 library config file
3      -----
4
5      copyright      : (C) 2002 by Frank Mori Hess
6                      (C) 1994 by C.Schroeter
7      email          : fmhess@users.sourceforge.net
8      *****/
9  /*****
10 *
11 *   Syntax:
12 *
13 *       interface { ... } starts new interface board section
14 *       device {...} device configuration
15 *
16 *****/
17
18 /* This section configures the configurable driver characteristics
19 * for an interface board, such as board address, and interrupt level.
20 * minor = 0 configures /dev/gpib0, minor = 1 configures /dev/gpib1, etc.
21 */
22
23 interface {
24     minor = 0        /* board index, minor = 0 uses /dev/gpib0, minor = 1 uses /dev/gpib1 */
25     board_type = "agilent_82357a" /* type of interface board being used */
26     name = "agi"      /* optional name, allows you to get a board descriptor using */
27     pad = 0 /* primary address of interface */
28     sad = 0 /* secondary address of interface */
29     timeout = T3s    /* timeout for commands */
30

```

```
31         eos = 0x0a      /* EOS Byte, 0xa is newline and 0xd is carriage return */
32         set-reos = yes   /* Terminate read if EOS */
33         set-bin = no     /* Compare EOS 8-bit */
34         set-xeos = no    /* Assert EOI whenever EOS byte is sent */
35         set-eot = yes    /* Assert EOI with last byte on writes */
36
37     /* settings for boards that lack plug-n-play capability */
38         base = 0          /* Base io ADDRESS */
39         irq  = 0          /* Interrupt request level */
40         dma  = 0          /* DMA channel (zero disables) */
41
42     /* pci_bus and pci_slot can be used to distinguish two pci boards supported by the s
43     /*     pci_bus = 0 */
44     /*     pci_slot = 7 */
45
46         master = yes     /* interface board is system controller */
47     }
48
49     /* This is how you might set up a pcIIa board on /dev/gpib1, uncomment to use. */
50     /*****
51     interface {
52         minor = 1
53         board_type = "pcIIa"
54         pad = 0
55         sad = 0
56         timeout = T3s
57
58         eos = 0x0a
59         set-reos = yes
60         set-bin = no
61
62         base = 0x2e1
63         irq  = 7
64         dma  = 1
65
66         master = yes
67     }
68     *****/
69
70     /* Now the device sections define the device characteristics for each device.
71     * These are only used if you want to open the device using ibfind() (instead
72     * of ibdev() )
```

```
73 */
74
75 device {
76     minor = 0
77     name = "hp3456a"
78     pad = 22
79 }
80
81
82
```

 [gpib\\_firmware-2008-08-10.tar.gz](#)

[View raw](#)

 [gpib\\_install.sh](#)

```
1  #!/usr/bin/env bash
2  sudo apt-get update
3  sudo apt-get install python-dev libboost-python-dev python-setuptools --yes
4  wget --content-disposition --no-check-certificate https://sourceforge.net/projects/linux-gpib/files/linux-gpib-3.2.20.tar.gz
5  tar xvfz linux-gpib-3.2.20.tar.gz
6  cd linux-gpib-3.2.20
7  ./configure
8  make -j8
9  sudo make install
10 cd ..
11 wget --content-disposition --no-check-certificate http://linux-gpib.sourceforge.net/files/gpib_firmware-2008-08-10.tar.gz
12 tar xvfz gpib_firmware-2008-08-10.tar.gz
13 wget --content-disposition --no-check-certificate https://downloads.sourceforge.net/linux-gpib/fxload-2008_10_13.tar.gz
14 tar xvfz fxload-2008_10_13.tar.gz
15 cd fxload-2008_10_13
16 make
17 sudo make install
18 cd ..
19 sudo ln -s /usr/local/lib/libgpib.so.0 /lib/libgpib.so.0
20 sudo cp -v gpib.conf /etc/gpib.conf
21 sudo modprobe gpib_common
22 sudo modprobe agilent_82357a
23 sudo fxload -D /dev/bus/usb/`lsusb | grep Agilent | cut -f 2 -d " " |`lsusb | grep Agilent
24 sleep 10
25 sudo fxload -D /dev/bus/usb/`lsusb | grep Agilent | cut -f 2 -d " " |`lsusb | grep Agilent
```

```
26 sleep 10
27 sudo gpib_config
28 echo "See https://gist.github.com/turingbirds/6eb05c9267a6437183a9567700e8581a for t
29
```

 [linux-gpib-3.2.20.tar.gz](#)

This file has been truncated, but you can [view the full file](#).

[View raw](#)



**zamora18** commented on Jan 16, 2018 • edited ▼

Hello. This was very useful. However, after following these instructions when I run the simple script:

```
import gpib
dev = gpib.find("hp3456a")
print gpib.read(dev, 99)
```

I get this error

```
InternalReceiveSetup: command failed
Traceback (most recent call last):
  File "/home/ubuntu/vna_comm_test/hello_world.py", line 3, in <module>
    print gpib.read(dev, 99)
gpib.GpibError: read() failed: An attempt to write command bytes to the bus
has timed out.
```

Could you give me some insight as to why this could be happening? Thank you.



**vishnubpatel** commented on Feb 18, 2018 • edited ▼



```
I am getting error with Agilent 82357b
$ sudo fxload -D /dev/bus/usb/003/010 -t fx2 -l gpib_firmware-2008-08-
10/agilent_82357a/measat_releaseX1.8.hex
$ sudo chmod 666 /dev/gpib0
$ sudo ln -s /usr/local/lib/libgpib.so.0 /lib/libgpib.so.0

$ sudo gpib_config
syntax error, unexpected $undefined
parameter error on line 1 of /etc/gpib.conf
libgpib: failed to parse configuration file
failed to parse config file /etc/gpib.conf
$
```



**hemanti0503** commented on Jul 5, 2018 • edited ▼

I get the same error as [@vishnubpatel](#).

```
$ sudo gpib_config
syntax error, unexpected $undefined
parameter error on line 1 of /etc/gpib.conf
libgpib: failed to parse configuration file
failed to parse config file /etc/gpib.conf
```

ibtest and the Python "Gbp" module (from the linux-gpib tree) fail with the same error message. Can anyone help on this, please?



**zamora18** commented on Sep 13, 2018

I am going back through this on a new machine (Ubuntu 16.04, kernel 4.15.0-34-generic) and I get this error when attempting `sudo modprobe gpib_common` :

```
modprobe: FATAL: Module gpib_common not found in directory
/lib/modules/4.15.0-34-generic
```

I get a similar error when running `sudo modprobe agilent_82357a` :

```
modprobe: FATAL: Module agilent_82357a not found in directory
/lib/modules/4.15.0-34-generic
```

Is there any way we can get help with this? It would be much appreciated.