

DIRECCIÓN DE SERVICIOS DE CERTIFICACIÓN

**LABORATORIO DE ENSAYOS DE
COMPATIBILIDAD ELECTROMAGNÉTICA RADIADA**

**INSTALACIÓN DEL ADAPTADOR USB/GPIB AGILENT 82357B EN
LINUX**

CONSTRUCCIÓN DE LA LIBRERÍA LINUX-GPIB Y CARGA DE FIRMWARE

CÓDIGO: FO-IT-002		N DOC:	
Originado por:	Elaborado por:	Revisado por:	Aprobado por:
Br. Arias B., Jose A.	Br. Arias B., Jose A.	-	-
Fecha: 20/06/2017	Fecha: 20/06/2016	Fecha:	Fecha:

Índice

1	Objetivos	2
2	Alcance	3
3	Documentos de referencia	3
3.1	Código fuente para librería linux-gpib	3
3.2	Actualización de firmware para el adaptador USB/GPIB 82357B	3
3.3	Instalación de los archivos de cabecera para linux	3
3.4	Detección de dispositivos en Linux y carga automática de firmware	4
4	Términos y definiciones	4
5	Personal autorizado	4
6	Personal requerido	4
7	Materiales	4
8	Herramientas y equipos	4
9	Equipos de protección personal	4
10	Precauciones de seguridad	5
11	Descripción de la actividad	5
11.1	Generalidades	5
11.2	Instalación de linux-gpib y módulos de núcleo	6
11.3	Instalación de módulos de núcleo	7
11.4	Instalación de archivos binarios de firmware para el 82357B	8
11.5	Configuración en Linux para que actualice el dispositivo inmediatamente al conectar	10
12	Anexos	13
12.1	Script de Bash para carga automática de firmware	13
12.2	Archivo con regla para detección de dispositivo Agilent 82357	14

1. Objetivos

- Describir el proceso de instalación adaptador USB/GPIB Agilent 82357B en Ubuntu LTS 14.
- Instalar y construir, a partir del código fuente, la librería c de soporte (linux-gpib).

- Obtener los archivos de código fuente y construir la utilidad de `linx fxload`, cargador de firmware para dispositivos USB.
- Obtener los archivos binarios y cargar el firmware para el dispositivo 82357B.

2. Alcance

Describe el proceso de instalación adaptador USB/GPIB Agilent 82357B, la instalación y construcción de la librería `c` de soporte (`linux-gpib`) a partir del código fuente y la obtención y carga del firmware para el adaptador.

3. Documentos de referencia

3.1. Código fuente para librería `linux-gpib`

A la fecha de edición de este documento, se encuentra el código fuente para la librería `linux-gpib` en la versión 4.0.4

<https://sourceforge.net/projects/linux-gpib/files/>
[https://sourceforge.net/projects/linux-gpib/files/latest/download?](https://sourceforge.net/projects/linux-gpib/files/latest/download?source=files)
[source=files](http://linux-gpib.sourceforge.net/)
<http://linux-gpib.sourceforge.net/>
http://linux-gpib.sourceforge.net/doc_html/x263.html#AGILENT-82357A

3.2. Actualización de firmware para el adaptador USB/GPIB 82357B

En el primer enlace se describe paso a paso el proceso de actualización del firmware para el dispositivo 82357B.

<https://gist.github.com/turingbirds/6eb05c9267a6437183a9567700e8581a>
<https://twiki.cern.ch/twiki/bin/view/Main/GpibLinux>

3.3. Instalación de los archivos de cabecera para linux

En los enlaces citados abajo se encuentra información acerca de como instalar los archivos de cabecera para el núcleo (kernel) de linux, de acuerdo al número de versión de la distribución de linux que se utilice.

<https://www.cyberciti.biz/faq/howto-install-kernel-headers-package>
<http://git.net/ml/linux.hardware.gpib.general/2008-02/msg00001.html>

3.4. Detección de dispositivos en Linux y carga automática de firmware

<https://es.wikipedia.org/wiki/Udev>

<https://unix.stackexchange.com/questions/65891/how-to-execute-a-shellscript-when-i-plug-in-a-usb-device>

4. Términos y definiciones

Shell Terminal de linux

GPIO General Purpose Instrumentation Bus

USB Universal Serial Bus

5. Personal autorizado

Personal técnico Cendit con interés en el uso del dispositivo.

6. Personal requerido

Todo aquel personal técnico Cendit con interés en el uso del dispositivo.

7. Materiales

- Adaptador USB/GPIO Agilent 82357B.
- Computador personal con puerto USB y acceso a internet.

8. Herramientas y equipos

Ver sección 7.

9. Equipos de protección personal

No se requieren equipos protección personal.

10. Precauciones de seguridad

Para ejecutar esta actividad no se prevén precauciones de seguridad

11. Descripción de la actividad

11.1. Generalidades

El adaptador USB/GPIB modelo 82357B, de Agilent Technologies como el mostrado en la figura 1 permite conectar un computador a un bus GPIB. Para poder utilizar este dispositivo se requiere que en el computador este instalado el software que le sirve de soporte, como lo son el módulo de kernel y las librerías de usuario.



Figura 1: Adaptador USB GPIB 82357B de Agilent Technologies

La empresa *Keysight Technologies* (antigua Agilent) proporciona un conjunto de herramientas de software, conocido bajo el nombre de *IO Libraries Suite*, de libre descarga en el sitio web de esta empresa en la versión 6.2017, a la fecha de redactar esta nota.

IO Libraries Suite proporciona el software necesario para el intercambio de datos con instrumentos programables a través de buses RS232, GPIB, USB y redes LAN. El software se presenta en forma de utilidades ejecutables que permiten enviar comandos a los instrumentos y recibir sus respuesta, resolución de problemas así como también un librerías para desarrollo de aplicaciones.

Las librerías *VISA (Virtual Instruments Software Architecture)* proporcionan una capa de software uniforme desde el punto de vista del programador, para el acceso a los buses mencionados. La librería VISA es parte integral de *IO Libraries Suite*. Existen librerías VISA producidas por otras empresas del ramo de la instrumentación inteligente, una de las más

reconocidas es la librería *VISA de National Instruments (NI-VISA)* así como también la librería VISA de Tektronix.

El software proporcionado por Keysight Technologies es exclusivo para ambiente Windows. Las librerías VISA de National Instruments están dirigidas Windows y a ciertas distribuciones de Linux, pero no para Ubuntu.

Existe una alternativa, la librería *linux-gpib*

Para utilizar el dispositivo adaptador USB / GPIB se requiere el siguiente soporte de software

- El soporte a nivel de librerías de usuario (user space) en el PC, en Ubuntu se recurre a *linux-gpib*.
- El soporte de software a nivel de núcleo de sistema (kernel) en el PC, los módulos de núcleo.
- El soporte a nivel de firmware en el dispositivo Agilent 82357B, es necesario cargar una versión actualizada del firmware para este dispositivo.

11.2. Instalación de linux-gpib y módulos de núcleo

La librería *linux-gpib* se construirá a partir de su código fuente, el cual puede descargarse de forma libre en el siguiente enlace,

<https://sourceforge.net/projects/linux-gpib/?source=dlp>

Como requisito para construir esta librería, en el sistema deben estar instalados los archivos de cabecera del núcleo, apropiado para la versión de núcleo de linux que se este utilizando. El comando `uname` de shell de linux permite obtener la versión de núcleo que se este utilizando,

```
uname -r
```

Por medio del siguiente comando de shell se instalan los archivos de cabecera, acordes a la versión de linux,

```
sudo apt-get update  
sudo apt-get install linux-headers-$(uname -r)
```

Si se desea tener acceso a esta librería por medio de Python, se debe instalar el paquete *python-setuptools*,

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

Ahora se procede a la descarga de los archivos fuente para la librería *linux-gpib*, puede hacerse sin salir del shell por medio del comando `wget`. Se puede crear un directorio nuevo para almacenar los archivos descargados, ejecutando previamente `mkdir nombre_directorio`.

```
wget -content-disposition -no-check-certificate  
https://sourceforge.net/projects/linux-gpib/files/latest/download?source=dlp
```

El paso siguiente es descomprimir el archivo mediante el comando de shell `tar`, como se indica a continuación, en donde se debe introducir el nombre del archivo comprimido el cual indica la versión, en este caso se trata de la versión 3.2.20,

```
tar xvfz tar xvfz linux-gpib-3.2.20.tar.gz
```

Ahora se procede a construir la librería, por medio de la secuencia de comandos,

```
./configure  
make -j8  
make install
```

Para comprobar que la librería ha sido instalada correctamente, el comando de shell `whereis` nos reporta la ubicación del la librería `libgpib.so`.

```
whereis libgpib.so
```

La repuesta del comando debe indicar la ruta en donde se encuentra la librería. En Debian:

```
libgpib: /usr/local/lib/libgpib.so /usr/local/lib/libgpib.la
```

11.3. Instalación de módulos de núcleo

El módulo de núcleo llamado `agilent_82357a.ko` también fue instalado durante el procedimiento de construcción e instalación del código fuente en el paso anterior. Para proceder a su carga, se debe ubicar este módulo dentro del sistema de archivos. Por ejemplo, en Debian, se encuentra en `/lib/modules/3.16.0-4-amd64/gpib/agilent_82357a`. En esta ruta el subdirectorío con nombre `3.16.0-4-amd64` indica la versión de núcleo del PC (3.14) y su arquitectura (`amd64`).

Nos ubicaos dentro de la carpeta que contiene este módulo, para proceder a su carga con el comando de shell `modprobe`, de la siguiente forma,

```
sudo modprobe agilent_82357a
```

Si la carga ocurre de forma correcta, el comando no genera ninguna repuesta. Así que se debe revisar si el módulo ha sido cargado de forma adecuada, por medio de los siguientes comandos de shell,

```
lsmod | grep agilent
```

Este comando emite una repuesta similar a la siguiente,

```
agilent_82357a 22661 0  
gpib_common 35582 1 agilent_82357a  
usbcore 195468 4 uhci_hcd,agilent_82357a,ehci_hcd,ehci_pci
```

La primera línea de la salida del comando `lsmod` indica que el módulo `agilent_82357a` se ha cargado correctamente.

El comando de shell `modinfo` permite obtener información sobre un módulo, se puede utilizar para verificar que el modulo `agilent_82357a` ha sido cargado,

```
modinfo agilent_82357a
```

Al ejecutar el comando anterior, se obtiene un respuesta similar a la siguiente,

```
filename: /lib/modules/3.13.0-110-generic/gpib/agilent_82357a/  
agilent_82357a.ko  
license: GPL  
srcversion: BAB8466E207C02D6E8023B3  
alias: usb:v0957p0718d*dc*dsc*dp*ic*isc*ip*in*  
alias: usb:v0957p0107d*dc*dsc*dp*ic*isc*ip*in*  
depends: gpib_common  
vermagic: 3.13.0-110-generic SMP mod_unload modversions 686
```

La cual indica que el módulo ha sido cargado correctamente.

11.4. Instalación de archivos binarios de firmware para el 82357B

Para que el dispositivo 82357B funcione correctamente debe actualizarse su firmware. Los archivo binarios para el firmware se pueden descargar en el siguiente enlace,

http://linux-gpib.sourceforge.net/firmware/gpib_firmware-2008-08-10.tar.gz

Se descargan los binarios comprimidos, se descomprimen con los acostumbrados comandos `wget` y `tar` respectivamente,

```
wget https://gist.github.com/turingbirds/6eb05c9267a6437183a9567700e8581a  
tar xvfz gpib_firmware-2008-08-10.tar.gz
```

Se necesita la utilidad de linux `fxload` que permite cargar en un dispositivo USB un archivo de firmware con extensión `.hex`. La utilidad `fxload` se construirá a partir de su código fuente, éste se descarga mediante el siguiente enlace,

[https://downloads.sourceforge.net/project/linux-hotplug/fxload/
2008_10_13/fxload-2008_10_13.tar.gz](https://downloads.sourceforge.net/project/linux-hotplug/fxload/2008_10_13/fxload-2008_10_13.tar.gz)

Siguiendo un procedimiento similar al mostrados anteriormente, se descarga el archivo comprimido, se descomprime para luego construir el código fuente de la forma acostumbrada en entornos linux.

Se descarga de este enlace el código fuente con el comando de shell `wget`,

```
wget -content-disposition -no-check-certificate  
https://downloads.sourceforge.net/project/linux-hotplug  
/fxload/2008_10_13/fxload-2008_10_13.tar.gz
```

Se descomprime el archivo con el comando de shell `tar`,

```
tar xvfz fxload-2008_10_13.tar.gz
```


Ahora se ingresa al subdirectorio `fxload-2008_10_13` en donde se encuentra el código fuente,

```
cd fxload-2008_10_13
```

Se construye el mismo por medio de los comandos,

```
make  
sudo make install
```

Se deben editar tres líneas en el archivo `/etc/gpib.conf`. Se emplea el editor de archivos `nano` como super usuario, de la siguiente forma,

```
sudo nano /etc/gpib.conf
```

Dentro del archivo `gpib.conf` se editan las líneas que contienen las entradas para `board_type`, `name` y `pad`. Para la entrada `name` el usuario puede introducir una cadena descriptiva, sin espacios, para identificar el adaptador. La entrada `pad` indica la dirección primaria del adaptador GPIB, el usuario puede establecer un valor numérico entre 0 y 30.

```
board_type = "agilent_82357a"  
name = "AGILENT82357B"  
pad = 22
```

Prosigue la carga de los módulos de núcleo que le dan soporte al adaptador 82357B, `gpib_common` y `agilent_82357a` se cargan por medio del comando shell `modprobe` de la siguiente forma,

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

Ahora se procede a insertar el dispositivo 82357B en el puerto USB del PC, en éste solo debe encontrarse iluminado en rojo el led etiquetado como LED FAIL.

Por medio del comando `lsusb` se encuentran los identificadores (ID) de bus (*bus ID*) y de dispositivo (*device ID*), dentro de la ventana de shell se ejecuta,

```
lsusb
```

El comando arroja varias entradas como respuesta, en una de ellas se debe encontrar la entrada para el adaptador de Agilent Technologies, similar a la que se muestra a continuación,

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

De esta línea se infiere que el ID de bus es el valor 002 y el ID de dispositivo es el valor 005. Los valores de ID de bus y ID de dispositivo pueden resultar diferentes, dependiendo del equipo donde se ejecute el comando. Se requieren estos datos en los argumentos del comando `fxload`.

Dentro de los archivos de firmware descargados se encuentra el archivo llamado `measat_releaseX1.8.hex`, que contiene el firmware para el dispositivo. Para cargar el firmware en el dispositivo 82357B se emplea la utilidad de shell `fxload` ejecutada como superusuario,

```
sudo fxload -D /dev/bus/usb/002/005 -t fx2 -I  
gpi_b_firmware-2008-08-10/agilent_82357a/measat_releaseX1.8.hex
```

En el dispositivo 82357B el LED FAIL debe estar aún iluminado en rojo. Se debe proceder a una nueva carga del firmware. Por medio del comando `lsusb` se obtienen por segunda vez el ID de bus y el ID de dispositivo

```
lsusb
```

El comando debe generar una respuesta similar a la siguiente,

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

El ID de bus debe ser el mismo (Bus 002), el valor que debería haber cambiado es el ID de dispositivo (que ahora es Device 006). Se procede nuevamente a ejecutar la utilidad `fxload`, introduciendo en el argumento de este comando el valores de ID de bus y el nuevo valor de ID de dispositivo,

```
sudo fxload -D /dev/bus/usb/002/006 -t fx2 -I  
gpi_b_firmware-2008-08-10/agilent_82357a/measat_releaseX1.8.hex
```

Como resultado de la ejecución del comando, en el dispositivo 82357B todos los LEDs deben encontrarse encendidos.

Se deben cambiar los permisos sobre el archivo que representa al dispositivo USB / GPIB, ubicado en la ruta `/dev/gpi_b0`, por medio del comando,

```
sudo chmod 666 /dev/gpi_b0
```

Resta inicializar el dispositivo 82357B por medio del comando de shell `gpi_b_config`. Este comando tiene ciertos problemas en ubicar la librería `libgpi_b.so`, así que se debe crear un *enlace simbólico* hacia ésta, de la siguiente forma,

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

Para ahora ejecutar el comando

```
gpi_b_config
```

Como resultado de procedimiento en el dispositivo 82357B se debe encontrar únicamente el LED READY iluminado en verde.

11.5. Configuración en Linux para que actualice el dispositivo inmediatamente al conectar

<https://unix.stackexchange.com/questions/65891/how-to-execute-a-shellscript-when-i-plug-in-a-usb-device>

Linux brinda al usuario la capacidad de ejecutar alguna acción cuando detecta la conexión o desconexión de un dispositivo. Esto se logra por medio del gestor de dispositivos `udev`, que usa el kernel de Linux en su versión 2.6. Este se encarga de controlar a los *archivos*

de *dispositivo*, ubicados dentro de la ruta `/dev` y de administrar la adicción o remoción de dispositivos, incluyendo la carga de firmware.

Mediante lo que se conoce como un *demonio* (o *daemon*) llamado `udev`, el gestor de dispositivos `udev` cuando se conecta o desconecta un obtiene información del contexto (subsistema de kernel, conector físico usado,...) así como también datos del propio dispositivo (numero de serie, fabricante,...).

El usuario puede especificar que acción se debe ejecutar cuando se identifica un dispositivo. Por ejemplo, puede especificar que se ejecute un script de shell cuando se detecte la inserción de un dispositivo USB con los valores de *Vendor ID (VID)* y *Product ID (PID)* especificados. Los valores hexadecimales de VID y PID identifican de manera unívoca a un dispositivo USB.

La acción a ejecutar se da mediante un *archivo de reglas*. Para ejecutar una acción al conectar un dispositivo USB, en el archivo de reglas se debe indicar en una línea los valores de PID y VID en conjunto con la acción a realizar. Los archivos de reglas se encuentran en `/etc/udev/rules.d`.

Se necesitan los valores hexadecimales de *VID* y *PID* que presenta el dispositivo antes de la carga de firmware. Justo después de conectar el dispositivo se ejecuta el comando de shell,

```
lsusb
```

como resultado en pantalla se observarán varias líneas con información acerca de los dispositivos USB actualmente conectados al PC, debe mostrar una línea similar a la siguiente,

```
Bus 001 Device 038: ID 0957:0518 Agilent Technologies, Inc.  
82357B GPIB Interface
```

donde se han resaltado los valores de VID:PID, se observa que el campo VID es el número 0957 y el campo para PID es el valor 0518.

Hecho esto, nos posicionamos en el subdirectorio `/etc/udev/rules.d` con el comando,

```
cd /etc/udev/rules.d
```

una vez dentro del subdirectorio `rules.d` creamos un archivo con nombre `agilent_82357.rules`. Creamos este archivo como super usuario por medio del editor `nano`,

```
sudo nano agilent_82357.rules
```

para luego introducir en el mismo la siguiente línea,

```
ACTION="add", ATTRS{idVendor}="0957", ATTRS{idProduct}=="0518",  
RUN+="/tmp/firmware_update_agilent_82357.sh"
```

Esta línea le indica al gestor de dispositivos `udev` que debe ejecutar un archivo de comandos de shell llamado `firmware_update_agilent_82357.sh` cuando ocurre la acción de conectar (`ACTION="add"`) un dispositivo USB con los valores de VID y PID indicados, que se obtuvieron con el comando **lsusb**.

En el archivo `firmware_update_agilent_82357.sh` se encuentran las instrucciones de bash para la carga de firmware, de forma automática sin intervención de usuario. Se muestra su contenido en el anexo 12.1.

Es importante que en el archivo de comandos a ejecutar cuando ocurre la conexión se establezca el permiso de ejecución y su como propietario del mismo al usuario administrador (`root`), de lo contrario no será ejecutado.

Se establece el *bit de ejecución* en el archivo por medio del comando,

```
sudo chmod 744 firmware_update_agilent_82357.sh
```

Se asigna como propietario del archvo al usurio raiz (`root`) por medio del comando,

```
sudo chown root:root firmware_update_agilent_82357.sh
```

Por último resta indicarle a `udev` que recargue los archivos de reglas, se logra esto con los siguientes comandos,

```
udevadm control -reload-rules  
udevadm trigger
```

Se desconecta y vuelve a conectar el dispositivo 82357B. Después de pocos segundos de conectado, las luces del mismo comenzaran a cambiar hasta que aparezca únicamente encendido el led verde `READY`, lo cual indica que la carga automática de firmware es exitosa.

12. Anexos

12.1. Script de Bash para carga automática de firmware

Listing 1: firmware_update_agilent_82357.sh

```
#!/bin/bash

# Script de Bash para actualización automática de firmware para
# el adaptador USB a GPIB Agilent 82357B.
# Se encarga de buscar el ID de BUS y el ID de DEVICE para
# preparar el comando de carga de firmware, por medio de <fxload>.
# El comando de carga debe ejecutarse dos veces, debido a un bug
# en el archivo .hex de código de firmware.

# Autor:          Jose A. Arias B.
# Correo:         correo@josearias.com.ve
# Version:        1.0

re="Bus ([0-9]+) Device ([0-9]+): ID 0957:0518"
#input="Bus 001 Device 006: ID 0957:0518 Agilent Technologies, Inc."

input=$(lsusb)

if [[ $input =~ $re ]]
then

    echo "Encontrado adaptador Agilent 82357B"
    echo "Primer intento para carga de firmware"

    bus=${BASH_REMATCH[1]}
    dev=${BASH_REMATCH[2]}
    echo ${BASH_REMATCH[0]}
    echo "1: Bus = $bus Device = $dev"
    sudo fxload -D /dev/bus/usb/$bus/$dev -t fx2 -I
    /etc/hotplug/usb/agilent_82357a/measat_releaseX1.8.hex

else

    echo "No se ha encontrado al adaptador Agilent 82357B"
    exit

fi

# Espera a que proceda la carga de firmware
sleep 3
```

```
input=$(lsusb)
echo $input

if [[ $input =~ $re ]]
then

echo "Segundo intento para carga de firmware"
bus=${BASH_REMATCH[1]}
dev=${BASH_REMATCH[2]}
echo ${BASH_REMATCH[0]}
echo "2: Bus = $bus Device = $dev"
sudo fxload -D /dev/bus/usb/$bus/$dev -t fx2 -I
/etc/hotplug/usb/agilent_82357a/measat_releaseX1.8.hex

# Espera de nuevo a que proceda la carga de firmware
sleep 3

sudo gpib_config

fi

echo "El adaptador Agilent 82357B esta listo para usarse"
```

12.2. Archivo con regla para detección de dispositivo Agilent 82357

Listing 2: agilent_82357.rules

```
ACTION='add', ATTRS\{idVendor\}='0957', ATTRS\{idProduct\}=='0518',
RUN+='/tmp/firmware\_update\_agilent\_82357.sh'
```