

TP python visa

Automatisation

d'une procédure de mesure

avec génération d'un compte rendu

Les instruments de mesures conformes à la norme VISA (*Virtual Instrument Software Architecture*) possèdent une interface (GPIB¹,USB,Série,Ethernet) qui permet de les contrôler à partir de commandes standard (langage SCPI). Chaque instrument peut ajouter des commandes propres au minimum défini dans le standard SCPI.

Dans ce tp vous allez automatiser une procédure de mesure manuelle en langage python.

La librairie PyVISA offre une interface python (frontend) avec la librairie VISA. Il existe plusieurs librairies VISA (*NI-VISA* de la société National Instruments, *Keysight I/O library* de la société éponyme) et l'implémentation python *pyvisa-py* que nous allons utiliser.

Voir la procédure d'installation et l'ensemble de la documentation ici : <https://pyvisa.readthedocs.io>

Tester votre installation dans l'interpréteur python :

```
>>>import pyvisa
>>>rm = pyvisa.ResourceManager()
>>>rm.list_ressources()
```

La liste des instruments connectés (USB ou Série) doit apparaître.

En plus vous pouvez vérifier la configuration de pyvisa par la commande : **pyvisa-info**

Également disponible la commande **pyvisa-shell** qui permet de communiquer avec un instrument en ligne de commande. Cette commande est très pratique pour tester les commandes SCPI.

Pour communiquer avec un instrument il faut connaître le "device string"² par exemple pour un analyseur de spectre Rigol on peut avoir³ :

```
TCPIP0::192.168.0.47::INSTR ou USB0::0x1AB1::0x04B0::DSA8H213900174::INSTR
```

```
philippe@jupiter:~$ pyvisa-shell
Welcome to the VISA shell. Type help or ? to list commands.
(visa) open TCPIP0::192.168.0.47::INSTR
TCPIP0::192.168.0.47::INSTR has been opened.
You can talk to the device using "write", "read" or "query".
The default end of message is added to each message.
(open) write *RST
(open) query *IDN?
Response: Rigol Technologies, DSA832E, DSA8H213900174, 00.01.04.00.00
```

¹ L'interface GPIB tend à disparaître mais c'est la seule qui permet une synchronisation très précise entre plusieurs instruments. Il existe des adaptateurs GPIB/USB ou GPIB/Ethernet pour les instruments anciens.

² Il faut parfois activer et/ou choisir le type d'interface dans la configuration de l'appareil. Le "device string" correspondant est parfois affichable par l'instrument, parfois fourni par une interface web ou alors dans la documentation.

³ Attention dans le connect string il y a :: (deux fois deux points) alors que dans les commandes SCPI il n'y a que : (une fois deux points)

Les méthodes principalement utilisées sont :

open_resource() : se connecter à l'instrument à partir de la "connect string"

write() : envoyer une commande à l'instrument.

read() : lire une donnée dans l'instrument.

query() : permet d'envoyer une commande et de lire le résultat (combinaison write puis read)

Le programme ci-dessous se connecte à un instrument par une interface ethernet, demande son identifiant puis fait un reset l'appareil :

```
import pyvisa
gbf_connect_string = 'TCPIP0::192.168.0.79::INSTR'
rm = pyvisa.ResourceManager()
gbf = rm.open_resource(gbf_connect_string)
r=gbf.query('*IDN?') #le paramètre de query se termine toujours par ?
print('connect gbf :',r)
gbf.write('*RST')
```

Les commandes qui commencent par * sont communes à tous les instruments. Pour aller plus loin il faut lire attentivement la documentation de l'instrument.

Les commandes ne sont pas sensible à la casse. Il existe parfois une version abrégée pour une commande. Ce peut être utile en ligne de commande mais il n'y a pas vraiment de raison d'utiliser la version abrégée dans un programme car ça peut nuire à la compréhension.

Exemple : Analyseur de spectre Rigol DAS800 :

[:SENSe]:FREQuency:CENTer

Syntax

[:SENSe]:FREQuency:CENTer <freq>

[:SENSe]:FREQuency:CENTer?

Description

Set the center frequency.

Query the center frequency.

Parameter

Name	Type	Range	Default
<freq>	Consecutive Real Number	0 Hz to 7.5 GHz	3.75 GHz

Return Format

The query returns the center frequency in integer and the unit is Hz.

Example

The command below sets the center frequency to 1 MHz.

```
:SENSe:FREQuency:CENTer 1000000
```

The query below returns 1000000.

```
:SENSe:FREQuency:CENTer?
```

[] signifient que le paramètres est optionnel, <> signifie que c'est obligatoire. Il ne faut pas bien sûr inclure [] ou <> dans la commande. La partie écrite en minuscule peut être omise (commande abrégée).

Le programme ci dessous se connecte à l'analyseur RIGOL DSA800 , règle la fréquence centrale à 220 MHz puis vérifie la fréquence centrale.

```
import pyvisa
spectrum_connect_string = 'TCPIP0::192.168.0.35::INSTR'
rm = pyvisa.ResourceManager()
spectrum = rm.open_resource(spectrum_connect_string)
spectrum.write(':SENSE:FREQUENCY:CENTER 220e6')
r=spectrum.query(':SENSE:FREQUENCY:CENTER?')
print('Center frequency :',r)
```

Remarquez les deux :: pour la connect string et seulement une fois : pour les commandes SCPI.

Travail demandé :

- connectez vous à un appareil avec pyvisa-shell et exécutez quelques commandes issues de sa documentation.
- testez les mêmes commandes dans un programme python.
- Dérouler manuellement la procédure de mesure (une procédure différente par binôme) pour afin de bien la maîtriser.
- Recherchez dans la documentation des appareils, comment exécuter les commandes de la procédure. Parfois la méthode est un peu différente de la méthode manuelle.
- Testez les commandes trouvées dans le shell.
- Écrire un programme python qui exécute automatiquement la procédure de mesure.

Votre programme final devra produire un compte rendu de mesure avec les informations suivantes : nom de l'opérateur, date, heure, type de mesure, identification des instruments utilisés, résultats. On pourra utiliser le format CSV (avec un entête) pour pouvoir exploiter facilement les résultats dans un tableur par exemple.

Pour certains instruments, on peut en plus sauver des copies d'écrans.

Rendu : - un dossier compressé avec tous les fichiers nécessaires au fonctionnement du programme.
- un mode d'emploi : programme et configuration initiale des appareils.

Critères de notation :

- fonctionnement du programme, respect de la procédure, validité des résultats.
- clarté du programme : choix des noms de variable, commentaires, organisation du programme (fonctions)
- facilité de paramétrage de la procédure, facilité de changement environnement (adresse IP, changement d'instrument (numéro de série usb).