

## O402a - Instrumentation

---

### Laboratoire 3 : Commande d'un générateur Rigol DG1022

---

Le rapport contiendra :

1. Des printscreens de la face avant du projet et du diagramme de l'exercice 2

L'objectif est d'utiliser des commandes SCPI afin de contrôler le générateur du laboratoire DG1022. Une commande est constituée d'une chaîne de caractère spécifique qui cible une commande et des arguments qui permettent de donner la valeur désirée. L'exemple suivant montre une telle instruction SCPI. Le document « DG1000 programming » sur claco reprend toutes les commandes liées au DG1022.

### Commands Introduction

#### Command Syntax

The command systems of DG1000 present a hierarchy structure (tree system) and each command consists of a "Root" keyword and one or multiple sub-keywords. The keywords are separated by ":" and are followed by the parameter settings available, "?" is added at the end of the command string to indicate query and the command and parameter are separated by "space".

For example,

```
FUNCTION:SQUare:DCYcle {<percent>|MINimum|MAXimum}
```

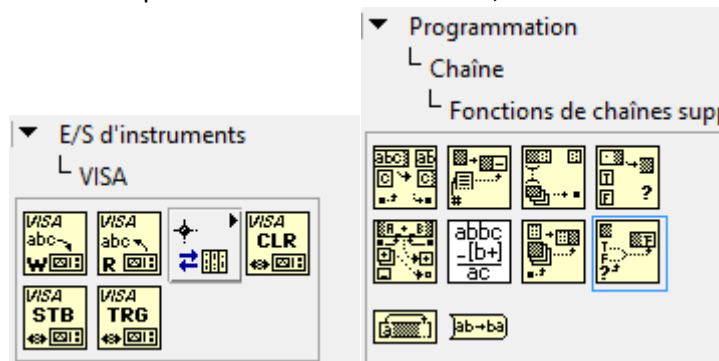
```
FUNCTION:SQUare:DCYcle? [MINimum|MAXimum]
```

**FUNCTION** is the root keyword of the command, **SQUare** and **DCYcle** are the second-level and third-level keywords respectively, all the keywords are separated by ":". <percent> denotes the parameter that users can set; "?" denotes query; the command **FUNCTION:SQUare:DCYcle** and parameter are separated by "space".

"," is usually used to compart multiple parameters existed in one command, for example,

```
DATA VOLATILE,<value>,<value>, . . .
```

Pour réaliser ces commandes, vous devrez manipuler les VI associés aux chaînes de caractères. Ces commandes seront transmises via le protocole VISA. Ces derniers blocs VI sont illustrés sur les images suivantes. Cela commence par une ouverture de la communication. Ensuite, des écritures qui doivent être espacées de 150ms. Finalement, la communication doit être fermée.

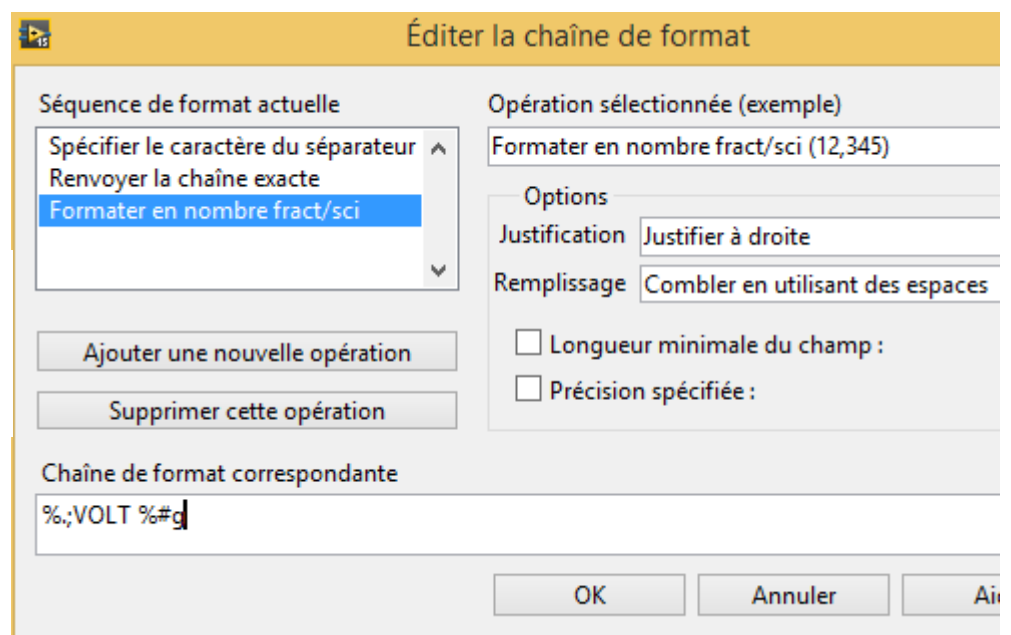
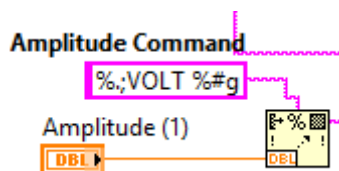


En ce qui concerne les chaînes de caractères, deux blocs utiles sont les « chaînes multiligne » pour sélectionner un choix entre différentes propositions via l'interface utilisateur.



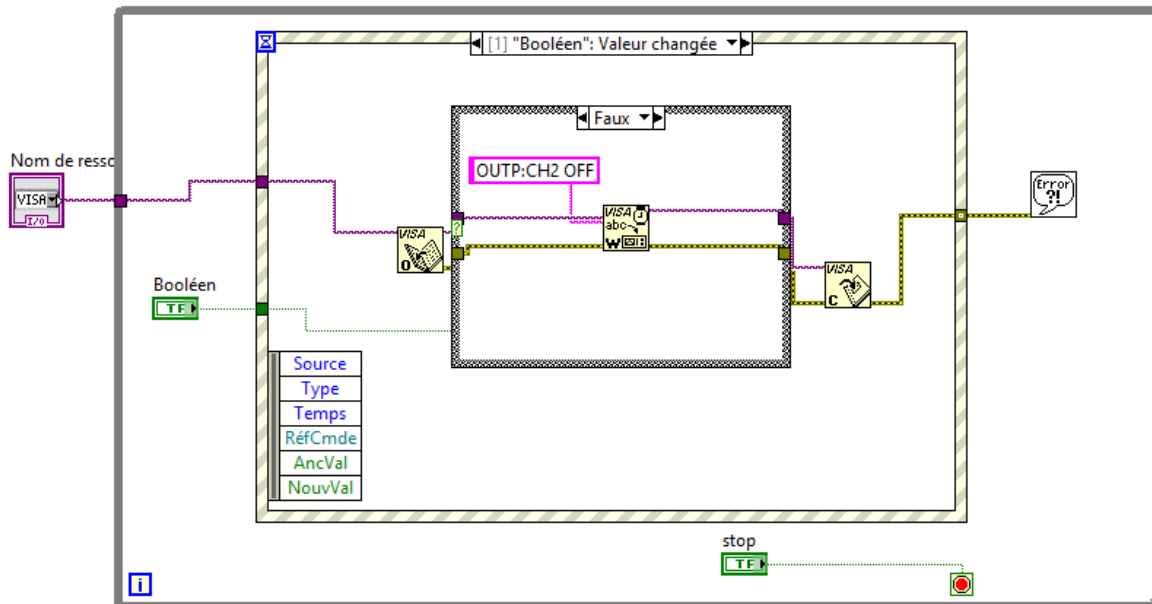
Un autre bloc permet de traduire des données numériques fournies pour l'interface utilisateur en chaîne de caractère, il faut alors transformer les données selon la convention suivante :

- Spécifier un caractère du séparateur décimal (.)
- Renvoyer une chaîne liée à la commande
- Un espace
- Formater le nombre en nombre fract/sci (justifié à droite et combler par des espaces)



1. Réaliser une interface d'activation pour le générateur Rigol

Installez le driver du matériel Rigol comme spécifié dans l'annexe. Réalisez une interface basique qui active ou non les channels du générateur. Un conseil est d'utiliser les structures d'évènements et booléennes comme illustré sur la figure suivante. Ajoutez le deuxième channel.



## 2. Reproduire l'interface du Rigol DG1022

Faites une interface reprenant les principales fonctionnalités du Rigol. Un exemple est illustré à la figure suivante. Testez une étape à la fois. Vous pouvez hiérarchiser vos Vis.

VISA resource name

**Waveform Function** (1: Sine)  
 1

**Amplitude** (1)      **Unit** (0: Vpp)      **DC Offset** (0.00 V)  
           

**Frequency** (1000 Hz)      **phase**  
     

**Configure**      **Output 1**      **Stop Button**  
           

**Output 2**

**output control error**  

status	code
	<input style="width: 90%;" type="text" value="0"/>
<b>source</b>	
<input style="width: 100%;" type="text"/>	

### 3. Extra : réaliser un test de vérification

Valider la configuration du générateur en lisant dans les registres du Rigol, les paramètres que vous avez sélectionné sur votre face-avant. . Afficher le résultat de cette vérification.

### Annexe : Installation du driver Rigol

Depuis l'interface LabVIEW, parcourez le chemin :

- Tools»Instrumentation»Find Instrument Drivers

Référence du driver :

- **Fabricant** : Rigol Technologies

Faites rechercher et sélectionnez. :

- **Driver** : rgdg1xxx Instrument driver qui est compatible avec le DG1022

## Types of Instrument Drivers

[»Table of Contents](#)

You can use Plug and Play and Interchangeable Virtual Instrument (IVI) driver types to control instruments in LabVIEW. National Instruments considers drivers Certified or Noncertified depending on if they meet specific requirements.

### Plug and Play Instrument Drivers

A LabVIEW Plug and Play instrument driver is a set of VIs that controls and communicates with a programmable instrument. Each VI corresponds to a programmatic operation, such as configuring, reading from, writing to, or triggering an instrument. LabVIEW Plug and Play instrument drivers include error handling, front panels, block diagrams, icons, and help. Because LabVIEW Plug and Play drivers maintain a common architecture and interface, you can quickly connect to and communicate with an instrument with very little or no code development. Project-style drivers take advantage of the LabVIEW project and library. You can use project-style drivers in the same way as previous LabVIEW Plug and Play drivers.

Use the [NI Instrument Driver Finder](#) to search for and install LabVIEW Plug and Play instrument drivers without leaving the LabVIEW development environment. Select **Tools»Instrumentation»Find Instrument Drivers** or **Help»Find Instrument Drivers** to launch the Instrument Driver Finder. If you want to download IVI and contributed drivers, use the [Instrument Driver Network](#). You also can search for National Instruments drivers at [NI Drivers and Updates](#) on the National Instruments Web site.

### IVI Instrument Drivers

National Instruments IVI drivers are sophisticated, DLL-based drivers developed in LabWindows™/CVI™ that allow for simulation and instrument interchangeability. You do not have to rewrite an application to use it with a similar instrument type. For example, you can write a VI that works with several different brands of oscilloscopes, even if those oscilloscopes use different bus connections. To achieve interchangeability, the IVI Foundation, which develops IVI standards, defines specifications for the following instrument classes: DMM, oscilloscope, arbitrary waveform/function generator, DC power supply, switch, power meter, spectrum analyzer, and RF signal generator. Use IVI drivers for the following additional benefits: instrument state caching for improved performance, multithread safety, and instrument attribute access. Refer to [IVI](#) on the National Instruments Web site for more information about IVI instrument drivers.

To convert an instrument driver written in LabWindows/CVI, download the LabVIEW Instrument Driver Import Wizard from the [National Instruments Web site](#).

### Instrument Driver Certification

Certified instrument drivers comply with instrument driver standards including programming style, error handling, documentation, and functional testing.

Noncertified instrument drivers typically solve a specific application need and are not usually fully functional. They are provided as is and are not supported by National Instruments.

Refer to the [NI Developer Zone](#) for more information on Instrument Driver Certification.

### Using VISA

VISA is a standard I/O API for instrumentation programming that provides the building blocks for instrument drivers. Refer to [Using VISA in LabVIEW](#) for more information on using VISA.

### Installing Instrument Drivers

You can [search for and install](#) existing LabVIEW instrument drivers.

#### LabVIEW 2015 Help

**Edition Date:** June 2015

**Part Number:** 371361M-01

[»View Product Info](#)

[Download Help \(Windows Only\)](#)