FgenInterface

Generated by Doxygen 1.8.9.1

Wed Sep 30 2015 15:43:04

Contents

1 fgen_interface 1

2 Namespace Index 3

2.1 Packages . . . 3

3 Class Index 5

3.1 Class List . . . 5

4 File Index 7

4.1 File List . . . 7

5 Namespace Documentation 9

5.1 fgen_test Namespace Reference . . . 9

5.1.1 Variable Documentation . . . 9

5.1.1.1 __author__ . . . 9

5.1.1.2 fgen . . . 9

5.2 FunctionGenerator Namespace Reference . . . 9

5.3 usbtmc Namespace Reference . . . 9

5.3.1 Variable Documentation . . . 9

5.3.1.1 __all__ . . . 10

5.4 usbtmc.usbtmc Namespace Reference . . . 10

5.5 usbtmc.version Namespace Reference . . . 10

5.5.1 Variable Documentation . . . 10

5.5.1.1 __version__ . . . 10

6 Class Documentation 11

6.1 FunctionGenerator.FunctionGenerator Class Reference . . . 11

6.1.1 Detailed Description . . . 11

6.1.2 Constructor & Destructor Documentation . . . 12

6.1.2.1 __init__ . . . 12

6.1.3 Member Function Documentation . . . 12

6.1.3.1 getError . . . 12

6.1.3.2 getIdn . . . 12

6.1.3.3 getStatus . . . 12
6.1.3.4 loadFromMemory . . . 12
6.1.3.5 pushSin . . . 12
6.1.3.6 setSin . . . 12
6.1.3.7 write . . . 12
6.1.4 Member Data Documentation . . . 13
6.1.4.1 addr . . . 13
6.1.4.2 instr . . . 13
6.1.4.3 selectorMap . . . 13

7 File Documentation **15**

7.1 fgen_test.py File Reference . . . 15
7.2 FunctionGenerator.py File Reference . . . 15
7.3 README.md File Reference . . . 15
7.4 usbtmc/__init__.py File Reference . . . 15
7.5 usbtmc/usbtmc.py File Reference . . . 16
7.6 usbtmc/version.py File Reference . . . 16

Chapter 1

fgen_interface

This library exposes relevant functions on the Agilent 33522A over USB in a straightforward and easy to use way. All documentation is done using the doxygen system (LaTeX and HTML versions can be found in /doc).

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

fgen_test 9
FunctionGenerator 9
usbtmc 9
usbtmc.usbtmc 10
usbtmc.version 10

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

FunctionGenerator.FunctionGenerator
This class wraps much of the functionality required to control an Agilent 33522A function generator over USB 11

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

fgen_test.py 15
FunctionGenerator.py 15
usbtmc/__init__.py 15
usbtmc/usbtmc.py 16
usbtmc/version.py 16

Chapter 5

Namespace Documentation

5.1 fgen_test Namespace Reference

Variables

- string __author__ = "Suyash Kumar (sk317)"
- tuple fgen = FunctionGenerator(1)

5.1.1 Variable Documentation

5.1.1.1 string fgen_test.__author__ = "Suyash Kumar (sk317)"

5.1.1.2 tuple fgen_test.fgen = FunctionGenerator(1)

5.2 FunctionGenerator Namespace Reference

Classes

- class FunctionGenerator

  This class wraps much of the functionality required to control an Agilent 33522A function generator over USB.

5.3 usbtmc Namespace Reference

Namespaces

- usbtmc
- version

Variables

- list __all__ = ["usbtmc"]

  Python USBTMC driver.

5.3.1 Variable Documentation

5.3.1.1 list usbtmc.__all__ = ["usbtmc"]

Python USBTMC driver.

5.4 usbtmc.usbtmc Namespace Reference

5.5 usbtmc.version Namespace Reference

Variables

- string __version__ = '0.6'

5.5.1 Variable Documentation

5.5.1.1 string usbtmc.version.__version__ = '0.6'

Chapter 6

Class Documentation

6.1 FunctionGenerator.FunctionGenerator Class Reference

This class wraps much of the functionality required to control an Agilent 33522A function generator over USB.

Public Member Functions

- def __init__ (self, instrumentSelector)

  The constructor for the function generator object needs to know the USBTMC address of the device being used.
- def getIdn (self)

  Asks the function generator to identify itself and retuns a unicode string of the response.
- def write (self, command)

  Writes the given custom SCPI command to the instrument over usbtmc.
- def getStatus ()

  Gets function generator's status and what output/settings are currently set to.
- def pushSin

  Pushes sin wave of the given parameters to function generator AND turns on output.
- def setSin
- def getError (self)
- def loadFromMemory (self, stateName)

Public Attributes

- addr
- instr

Static Public Attributes

- dictionary selectorMap = {1:"USB0::2391::8967::MY50000586::INSTR"}

6.1.1 Detailed Description

This class wraps much of the functionality required to control an Agilent 33522A function generator over USB.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 def FunctionGenerator.FunctionGenerator.__init__ (*self, instrumentSelector*)

The constructor for the function generator object needs to know the USBTMC address of the device being used.

The address can be directly supplied as a string, or an integer representing the function generator in Dr. Nightingale's lab can be passed.

Parameters

instrument↩ Selector	Either a string representing the USBTMC address of the function generator or a int identifier representing one of the function generators in Kathy Nightingale's lab.

6.1.3 Member Function Documentation

6.1.3.1 def FunctionGenerator.FunctionGenerator.getError (*self*)

6.1.3.2 def FunctionGenerator.FunctionGenerator.getIdn (*self*)

Asks the function generator to identify itself and retuns a unicode string of the response.

Returns

identity a unicode string the attached function generator uses to identify itself.

6.1.3.3 def FunctionGenerator.FunctionGenerator.getStatus ( )

Gets function generator's status and what output/settings are currently set to.

Returns

status string of current status

6.1.3.4 def FunctionGenerator.FunctionGenerator.loadFromMemory (*self, stateName*)

6.1.3.5 def FunctionGenerator.FunctionGenerator.pushSin (*self, frequency, amplitude* = `1`, *offset* = `0`)

Pushes sin wave of the given parameters to function generator AND turns on output.

Parameters

frequency	the frequency in Hz of the sin wave
amplitude	(optional, default set to 1V), amplitude of the sin wave in volts
offset	(optional, default set to 0V), dc offset of sin wave in volts.

6.1.3.6 def FunctionGenerator.FunctionGenerator.setSin (*self, frequencey, amplitude* = `1`, *offset* = `0`)

6.1.3.7 def FunctionGenerator.FunctionGenerator.write (*self, command*)

Writes the given custom SCPI command to the instrument over usbtmc.

Parameters

command	A string representing the SCPI command

6.1.4 Member Data Documentation

6.1.4.1 FunctionGenerator.FunctionGenerator.addr

6.1.4.2 FunctionGenerator.FunctionGenerator.instr

6.1.4.3 dictionary FunctionGenerator.FunctionGenerator.selectorMap = {1:"USB0::2391::8967::MY50000586::INSTR"} [static]

The documentation for this class was generated from the following file:

- FunctionGenerator.py

Chapter 7

File Documentation

7.1 fgen_test.py File Reference

Namespaces

- fgen_test

Variables

- string fgen_test.__author__ = "Suyash Kumar (sk317)"
- tuple fgen_test.fgen = FunctionGenerator(1)

7.2 FunctionGenerator.py File Reference

Classes

- class FunctionGenerator.FunctionGenerator

  This class wraps much of the functionality required to control an Agilent 33522A function generator over USB.

Namespaces

- FunctionGenerator

7.3 README.md File Reference

7.4 usbtmc/__init__.py File Reference

Namespaces

- usbtmc

Variables

- list usbtmc.__all__ = ["usbtmc"]

  Python USBTMC driver.

7.5 usbtmc/usbtmc.py File Reference

Namespaces

- usbtmc.usbtmc

7.6 usbtmc/version.py File Reference

Namespaces

- usbtmc.version

Variables

- string usbtmc.version.__version__ = '0.6'