

## FgenInterface

Generated by Doxygen 1.8.9.1

Wed Sep 30 2015 15:37:01



# Contents

<b>1</b>	<b>fgen_interface</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Packages	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
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
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
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	<a href="#">getStatus</a>	12
6.1.3.4	<a href="#">loadFromMemory</a>	12
6.1.3.5	<a href="#">pushSin</a>	12
6.1.3.6	<a href="#">setSin</a>	12
6.1.3.7	<a href="#">write</a>	12
6.1.4	<a href="#">Member Data Documentation</a>	13
6.1.4.1	<a href="#">addr</a>	13
6.1.4.2	<a href="#">instr</a>	13
6.1.4.3	<a href="#">selectorMap</a>	13
<b>7</b>	<b><a href="#">File Documentation</a></b>	<b>15</b>
7.1	<a href="#">fgen_test.py File Reference</a>	15
7.2	<a href="#">FunctionGenerator.py File Reference</a>	15
7.3	<a href="#">README.md File Reference</a>	15
7.4	<a href="#">usbtmc/__init__.py File Reference</a>	15
7.5	<a href="#">usbtmc/usbtmc.py File Reference</a>	16
7.6	<a href="#">usbtmc/version.py File Reference</a>	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):

<a href="#">fgen_test</a>	9
<a href="#">FunctionGenerator</a>	9
<a href="#">usbtmc</a>	9
<a href="#">usbtmc.usbtmc</a>	10
<a href="#">usbtmc.version</a>	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:

<a href="#">fgen_test.py</a> . . . . .	15
<a href="#">FunctionGenerator.py</a> . . . . .	15
<a href="#">usbtmc/__init__.py</a> . . . . .	15
<a href="#">usbtmc/usbtmc.py</a> . . . . .	16
<a href="#">usbtmc/version.py</a> . . . . .	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.

Copyright (c) 2012 Alex Forencich

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 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 returns 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

<i>instrumentSelector</i>	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 returns 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

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

#### 6.1.3.6 `def FunctionGenerator.FunctionGenerator.setSin( self, frequency, 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

<i>command</i>	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 = {"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'