

Code_generator

Generated by Doxygen 1.8.14

Contents

1	Namespace Index	1
1.1	Packages	1
2	Class Index	3
2.1	Class List	3
3	Namespace Documentation	5
3.1	Package common	5
3.1.1	Detailed Description	5
3.2	Package configurator	5
3.2.1	Detailed Description	6
3.3	Package framework	6
3.3.1	Detailed Description	6
3.4	Package gui	6
3.4.1	Detailed Description	6
3.5	Package microcontroller	7
3.5.1	Detailed Description	7
3.6	Package projectConfiguration	7
3.6.1	Detailed Description	7
3.7	Package xmlCreator	7
3.7.1	Detailed Description	8
3.8	Package xmlParser	8
3.8.1	Detailed Description	8

4	Class Documentation	9
4.1	gui>AboutWindow Class Reference	9
4.1.1	Constructor & Destructor Documentation	9
4.1.1.1	AboutWindow()	9
4.1.2	Member Function Documentation	9
4.1.2.1	main()	9
4.2	configurator.GPIO.CodeName Enum Reference	10
4.2.1	Detailed Description	10
4.2.2	Member Data Documentation	10
4.2.2.1	STR_NAME	10
4.3	configurator.ConfigurationFile Class Reference	10
4.3.1	Detailed Description	11
4.3.2	Member Data Documentation	11
4.3.2.1	STR_PROJ_CONF_FILE	11
4.4	xmlCreator.ConfXmlWriter Class Reference	11
4.4.1	Detailed Description	11
4.4.2	Constructor & Destructor Documentation	12
4.4.2.1	ConfXmlWriter()	12
4.4.3	Member Function Documentation	12
4.4.3.1	addPin()	12
4.4.3.2	writeXml()	12
4.5	common.ErrorCode Enum Reference	13
4.5.1	Detailed Description	13
4.5.2	Member Data Documentation	13
4.5.2.1	EX_ERROR	14
4.5.2.2	FILE_READ_ERROR	14
4.5.2.3	FILE_WRITE_ERROR	14
4.5.2.4	NO_ERROR	14

4.5.2.5	STR_INVALID	14
4.6	common.Features Class Reference	14
4.6.1	Detailed Description	15
4.6.2	Member Function Documentation	15
4.6.2.1	debugPrint()	15
4.6.2.2	verbosePrint()	15
4.6.3	Member Data Documentation	16
4.6.3.1	DEBUG	16
4.6.3.2	DEBUG_STR	16
4.6.3.3	SW_VERSION	16
4.6.3.4	VERBOSE	16
4.6.3.5	VERBOSE_STR	16
4.6.3.6	VERSION_NAME	17
4.6.3.7	VERSION_STATUS	17
4.7	gui.GpioConfWindow Class Reference	17
4.7.1	Detailed Description	17
4.7.2	Constructor & Destructor Documentation	17
4.7.2.1	GpioConfWindow()	17
4.7.3	Member Function Documentation	18
4.7.3.1	main()	18
4.8	gui.MainGui Class Reference	18
4.8.1	Detailed Description	19
4.8.2	Member Function Documentation	19
4.8.2.1	loadProjectFile()	19
4.8.2.2	main()	19
4.8.2.3	saveUc()	20
4.8.2.4	setNewUC()	20
4.8.2.5	showAboutWindow()	20

4.8.2.6	showErrorDialog()	20
4.8.2.7	showGpioConfWindow()	20
4.9	gui.MainWindow Class Reference	21
4.9.1	Detailed Description	21
4.9.2	Constructor & Destructor Documentation	21
4.9.2.1	MainWindow()	22
4.9.3	Member Function Documentation	22
4.9.3.1	main()	22
4.9.3.2	OpenFileChooser()	22
4.9.3.3	setProjectInformation()	23
4.9.3.4	setVisible()	23
4.10	gui.Messages Class Reference	23
4.11	microcontroller.Microcontroller Class Reference	24
4.11.1	Detailed Description	24
4.11.2	Constructor & Destructor Documentation	24
4.11.2.1	Microcontroller()	24
4.11.3	Member Function Documentation	25
4.11.3.1	getConfiguredPin()	25
4.11.3.2	getPin()	25
4.11.3.3	getUc_gpioNum()	26
4.11.3.4	getUc_manufacturer()	26
4.11.3.5	getUc_model()	26
4.11.3.6	getUc_pinNum()	26
4.11.3.7	getUc_portNum()	27
4.11.3.8	isValid()	27
4.11.3.9	processDocument()	27
4.11.4	Member Data Documentation	27
4.11.4.1	GpioCfgPin	27

4.11.4.2	Ports	28
4.12	configurator.GPIO.Mode Enum Reference	28
4.12.1	Detailed Description	28
4.12.2	Member Function Documentation	28
4.12.2.1	getConfFromString()	28
4.12.3	Member Data Documentation	29
4.12.3.1	STR_NAME	29
4.13	configurator.GPIO.OutLevel Enum Reference	29
4.13.1	Detailed Description	30
4.13.2	Member Function Documentation	30
4.13.2.1	getConfFromString()	30
4.13.3	Member Data Documentation	30
4.13.3.1	STR_NAME	30
4.14	configurator.GPIO.OutType Enum Reference	31
4.14.1	Detailed Description	31
4.14.2	Member Function Documentation	31
4.14.2.1	getConfFromString()	31
4.14.3	Member Data Documentation	32
4.14.3.1	STR_NAME	32
4.15	microcontroller.Pin Class Reference	32
4.15.1	Detailed Description	33
4.15.2	Constructor & Destructor Documentation	34
4.15.2.1	Pin()	34
4.15.3	Member Function Documentation	34
4.15.3.1	getAdc()	34
4.15.3.2	getClock()	35
4.15.3.3	getFeat_adc()	35
4.15.3.4	getFeat_clock()	35

4.15.3.5	getFeat_i2c()	35
4.15.3.6	getFeat_int()	36
4.15.3.7	getFeat_reset()	36
4.15.3.8	getFeat_spi()	36
4.15.3.9	getFeat_timer()	36
4.15.3.10	getFeat_uart()	37
4.15.3.11	getFunc_gnd()	37
4.15.3.12	getFunc_gpio()	37
4.15.3.13	getFunc_vcc()	37
4.15.3.14	getI2c()	38
4.15.3.15	getInt()	38
4.15.3.16	getName()	38
4.15.3.17	getNumber()	38
4.15.3.18	getPort()	39
4.15.3.19	getReset()	39
4.15.3.20	getSpi()	39
4.15.3.21	getTimer()	39
4.15.3.22	getUart()	40
4.15.3.23	isValid()	40
4.15.3.24	setAdc()	40
4.15.3.25	setClock()	40
4.15.3.26	setFeat_adc()	41
4.15.3.27	setFeat_clock()	41
4.15.3.28	setFeat_i2c()	41
4.15.3.29	setFeat_int()	42
4.15.3.30	setFeat_reset()	42
4.15.3.31	setFeat_spi()	42
4.15.3.32	setFeat_timer()	42

4.15.3.33 setFeat_uart()	43
4.15.3.34 setFunc_gnd()	43
4.15.3.35 setFunc_gpio()	43
4.15.3.36 setFunc_vcc()	44
4.15.3.37 setI2c()	44
4.15.3.38 setInt()	44
4.15.3.39 setName()	45
4.15.3.40 setNumber()	45
4.15.3.41 setPort()	45
4.15.3.42 setReset()	45
4.15.3.43 setSpi()	46
4.15.3.44 setTimer()	46
4.15.3.45 setUart()	46
4.15.4 Member Data Documentation	47
4.15.4.1 DEF_FEATURE	47
4.15.4.2 DEF_FEATURE_AV	47
4.15.4.3 DEF_FUNCTION	47
4.15.4.4 DEF_NAME	47
4.15.4.5 DEF_NUMBER	47
4.15.4.6 DEF_PORT	48
4.15.4.7 DISABLE	48
4.15.4.8 ENABLE	48
4.16 configurator.PinConf Class Reference	48
4.16.1 Detailed Description	49
4.16.2 Constructor & Destructor Documentation	49
4.16.2.1 PinConf()	49
4.16.3 Member Function Documentation	49
4.16.3.1 getCodeName()	49

4.16.3.2	getMode()	50
4.16.3.3	getOutLevel()	50
4.16.3.4	getOutType()	50
4.16.3.5	getPinName()	51
4.16.3.6	getPort()	51
4.16.3.7	getPull()	51
4.16.3.8	getSpeed()	51
4.16.3.9	isAv_Adc()	52
4.16.3.10	isAv_altFunc()	52
4.16.3.11	isValid()	52
4.16.3.12	setCodeName()	52
4.16.3.13	setMode()	53
4.16.3.14	setOutLevel()	53
4.16.3.15	setOutType()	53
4.16.3.16	setPull()	54
4.16.3.17	setSpeed()	54
4.16.4	Member Data Documentation	54
4.16.4.1	DF_MODE	54
4.16.4.2	DF_OUT_LEVEL	54
4.16.4.3	DF_OUTTYPE	55
4.16.4.4	DF_PULL	55
4.16.4.5	DF_SPEED	55
4.17	projectConfiguration.ProjectSettings Class Reference	55
4.17.1	Detailed Description	55
4.17.2	Constructor & Destructor Documentation	56
4.17.2.1	ProjectSettings()	56
4.17.3	Member Function Documentation	56
4.17.3.1	getConfFile()	56

4.17.3.2	getProjectName()	56
4.17.3.3	getUcFile()	56
4.17.3.4	openProjectFile()	56
4.17.3.5	processDocument()	57
4.18	configurator.GPIO.Pull Enum Reference	57
4.18.1	Detailed Description	58
4.18.2	Member Function Documentation	58
4.18.2.1	getConfFromString()	58
4.18.3	Member Data Documentation	58
4.18.3.1	STR_NAME	58
4.19	configurator.GPIO.Speed Enum Reference	59
4.19.1	Detailed Description	59
4.19.2	Member Function Documentation	59
4.19.2.1	getConfFromString()	59
4.19.3	Member Data Documentation	60
4.19.3.1	STR_NAME	60
4.20	xmlParser.TestMain Class Reference	60
4.20.1	Detailed Description	60
4.20.2	Member Function Documentation	60
4.20.2.1	main()	60
4.21	xmlParser.XmlOpener Class Reference	61
4.21.1	Detailed Description	61
4.21.2	Constructor & Destructor Documentation	61
4.21.2.1	XmlOpener()	61
4.21.3	Member Function Documentation	62
4.21.3.1	getElementInfo()	62
4.21.3.2	getElementInfoFromDoc()	62
4.21.3.3	getParsedDoc()	62
4.21.3.4	OpenFile()	63

Chapter 1

Namespace Index

1.1 Packages

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

common	5
configurator	5
framework	6
gui	6
microcontroller	7
projectConfiguration	7
xmlCreator	7
xmlParser	8

Chapter 2

Class Index

2.1 Class List

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

gui>AboutWindow	9
configurator.GPIO.CodeName	10
configurator.ConfigurationFile	10
xmlCreator.ConfXmlWriter	11
common.ErrorCode	13
common.Features	14
gui.GpioConfWindow	17
gui.MainGui	18
gui.MainWindow	21
gui.Messages	23
microcontroller.Microcontroller	24
configurator.GPIO.Mode	28
configurator.GPIO.OutLevel	29
configurator.GPIO.OutType	31
microcontroller.Pin	32
configurator.PinConf	48
projectConfiguration.ProjectSettings	55
configurator.GPIO.Pull	57
configurator.GPIO.Speed	59
xmlParser.TestMain	60
xmlParser.XmlOpener	61

Chapter 3

Namespace Documentation

3.1 Package common

Classes

- enum [ErrorCode](#)
- class [Features](#)

3.1.1 Detailed Description

Common information that needs to be accessed across all the project

Author

Miguel Diaz

Version

0.1

3.2 Package configurator

Classes

- class [ConfigurationFile](#)
- class [PinConf](#)

3.2.1 Detailed Description

Configuration classes

Author

Miguel Diaz

Version

0.1

3.3 Package framework

3.3.1 Detailed Description

Framework information

Author

H112943

Version

0.1

3.4 Package gui

Classes

- class [AboutWindow](#)
- class [GpioConfWindow](#)
- class [MainGui](#)
- class [MainWindow](#)
- class [Messages](#)

3.4.1 Detailed Description

Author

Miguel Diaz

Version

0.1

3.5 Package microcontroller

Classes

- class [Microcontroller](#)
- class [Pin](#)

3.5.1 Detailed Description

[Microcontroller](#) related classes

Author

Miguel Diaz

Version

0.1

3.6 Package projectConfiguration

Classes

- class [ProjectSettings](#)

3.6.1 Detailed Description

Project settings and configuration files

Author

Miguel Diaz

Version

0.1

3.7 Package xmlCreator

Classes

- class [ConfXmlWriter](#)

3.7.1 Detailed Description

Create configuration XML

Author

Miguel Diaz

Version

0.1

3.8 Package xmlParser

Classes

- class [TestMain](#)
- class [XmlOpener](#)

3.8.1 Detailed Description

XML parser for microcontroller information and project settings

Author

Miguel Diaz

Version

0.1

Chapter 4

Class Documentation

4.1 gui.AboutWindow Class Reference

Public Member Functions

- [AboutWindow](#) ()

Static Public Member Functions

- static void [main](#) (String[] args)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 AboutWindow()

```
gui.AboutWindow.AboutWindow ( )
```

Create the application.

4.1.2 Member Function Documentation

4.1.2.1 main()

```
static void gui.AboutWindow.main (
    String [] args ) [static]
```

About window main

Parameters

<i>args</i>	Init parameters
-------------	-----------------

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

- `src/gui/AboutWindow.java`

4.2 configurator.GPIO.CodeName Enum Reference

Static Public Attributes

- static final String `STR_NAME` = "codeName"

4.2.1 Detailed Description

Author

Miguel Diaz

Version

0.1

4.2.2 Member Data Documentation

4.2.2.1 STR_NAME

```
static final String configurator.GPIO.CodeName.STR_NAME = "codeName" [static]
```

Name as String

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

- `src/configurator/GPIO/CodeName.java`

4.3 configurator.ConfigurationFile Class Reference

Static Public Attributes

- static final String `STR_PROJ_CONF_FILE` = "cgs"

4.3.1 Detailed Description

Configuration files properties

Author

Miguel Diaz

Version

0.1

4.3.2 Member Data Documentation

4.3.2.1 STR_PROJ_CONF_FILE

```
final String configurator.ConfigurationFile.STR_PROJ_CONF_FILE = "cgs" [static]
```

Public configuration file extension

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

- src/configurator/ConfigurationFile.java

4.4 xmlCreator.ConfXmlWriter Class Reference

Public Member Functions

- [ConfXmlWriter](#) ([Microcontroller](#) uC)
- void [addPin](#) ([PinConf](#) pin, int pinNum)
- [ErrorCode](#) [writeXml](#) (String fileName)

4.4.1 Detailed Description

Write a XML file

Author

Miguel Diaz

Version

0.1

4.4.2 Constructor & Destructor Documentation

4.4.2.1 ConfXmlWriter()

```
xmlCreator.ConfXmlWriter.ConfXmlWriter (
    Microcontroller uC )
```

Constructor

Parameters

<i>uC</i>	Microcontroller configuration
-----------	-------------------------------

4.4.3 Member Function Documentation

4.4.3.1 addPin()

```
void xmlCreator.ConfXmlWriter.addPin (
    PinConf pin,
    int pinNum )
```

Add a pin configuration to the file

Parameters

<i>pin</i>	Pin configuration
<i>pinNum</i>	Number of GPIO pin

4.4.3.2 writeXml()

```
ErrorCode xmlCreator.ConfXmlWriter.writeXml (
    String fileName )
```

Write the XML file

Parameters

<i>fileName</i>	Name of XML configuration file
-----------------	--------------------------------

Returns

Error status

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

- src/xmlCreator/ConfXmlWriter.java

4.5 common.ErrorCode Enum Reference

Public Attributes

- [NO_ERROR](#)
- [EX_ERROR](#)
- [FILE_READ_ERROR](#)
- [FILE_WRITE_ERROR](#)

Static Public Attributes

- static final String [STR_INVALID](#) = "STR_INVALID"

4.5.1 Detailed Description

Error codes enum

Author

Miguel Diaz

Version

0.1

4.5.2 Member Data Documentation

4.5.2.1 EX_ERROR

```
common.ErrorCode.EX_ERROR
```

Error during execution

4.5.2.2 FILE_READ_ERROR

```
common.ErrorCode.FILE_READ_ERROR
```

File reading error

4.5.2.3 FILE_WRITE_ERROR

```
common.ErrorCode.FILE_WRITE_ERROR
```

File writing error

4.5.2.4 NO_ERROR

```
common.ErrorCode.NO_ERROR
```

No error message

4.5.2.5 STR_INVALID

```
static final String common.ErrorCode.STR_INVALID = "STR_INVALID" [static]
```

Error string

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

- `src/common/ErrorCode.java`

4.6 common.Features Class Reference

Static Public Member Functions

- static void [verbosePrint](#) (String verboseMessage)
- static void [debugPrint](#) (String debugMessage)

Static Public Attributes

- static final boolean `DEBUG` = true
- static final boolean `VERBOSE` = true
- static final String `VERBOSE_STR` = "# "
- static final String `DEBUG_STR` = "#\$ "
- static final String `SW_VERSION` = VERSION_MAJOR + "." + VERSION_MINOR + "." + VERSION_PATCH
- static final String `VERSION_STATUS` = "Alpha"
- static final String `VERSION_NAME` = "Alderaan"

4.6.1 Detailed Description

Class that includes all project features

Author

Miguel Diaz

Version

0.1

4.6.2 Member Function Documentation

4.6.2.1 `debugPrint()`

```
static void common.Features.debugPrint (  
    String debugMessage ) [static]
```

Print Debug message to console

Parameters

<i>debugMessage</i>	Message to display
---------------------	--------------------

4.6.2.2 `verbosePrint()`

```
static void common.Features.verbosePrint (  
    String verboseMessage ) [static]
```

Print Verbose message to console

Parameters

<i>verboseMessage</i>	Message to display
-----------------------	--------------------

4.6.3 Member Data Documentation

4.6.3.1 DEBUG

```
final boolean common.Features.DEBUG = true [static]
```

Enables debug functions

4.6.3.2 DEBUG_STR

```
final String common.Features.DEBUG_STR = "$ " [static]
```

Debug messages indicator on system console

4.6.3.3 SW_VERSION

```
final String common.Features.SW_VERSION = VERSION_MAJOR + "." + VERSION_MINOR + "." + VERSION_PATCH [static]
```

Complete Software version

4.6.3.4 VERBOSE

```
final boolean common.Features.VERBOSE = true [static]
```

Enables console messages

4.6.3.5 VERBOSE_STR

```
final String common.Features.VERBOSE_STR = "# " [static]
```

Verbose messages indicator on system console

4.6.3.6 VERSION_NAME

```
final String common.Features.VERSION_NAME = "Alderaan" [static]
```

Code name of the software version

4.6.3.7 VERSION_STATUS

```
final String common.Features.VERSION_STATUS = "Alpha" [static]
```

Status of the software version

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

- src/common/Features.java

4.7 gui.GpioConfWindow Class Reference

Public Member Functions

- [GpioConfWindow](#) ([Microcontroller](#) uCtrl)

Static Public Member Functions

- static void [main](#) (String[] args)

4.7.1 Detailed Description

Window for configuring GPIO pins

Author

Miguel Diaz

Version

0.1

4.7.2 Constructor & Destructor Documentation

4.7.2.1 GpioConfWindow()

```
gui.GpioConfWindow.GpioConfWindow (  
    Microcontroller uCtrl )
```

Create the GPIO configuration window and show it

Parameters

<i>uCtrl</i>	Microcontroller object containing all pin's information
--------------	---

4.7.3 Member Function Documentation

4.7.3.1 main()

```
static void gui.GpioConfWindow.main (  
    String [] args ) [static]
```

Gpio configuration window main

Parameters

<i>args</i>	Init parameters
-------------	-----------------

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

- src/gui/GpioConfWindow.java

4.8 gui.MainGui Class Reference

Static Public Member Functions

- static void [main](#) (String[] args)
- static [ErrorCode](#) [loadProjectFile](#) (File inFile)
- static void [showErrorDialog](#) (String message)
- static void [showAboutWindow](#) ()
- static void [showGpioConfWindow](#) ()
- static void [setNewUC](#) ([Microcontroller](#) uC)
- static void [saveUc](#) ()

Static Public Attributes

- static File **ProjectFile**
- static String **ProjectPath**

4.8.1 Detailed Description

Main GUI state machine

Author

Miguel Diaz

Version

0.1

4.8.2 Member Function Documentation

4.8.2.1 loadProjectFile()

```
static ErrorCode gui.MainGui.loadProjectFile (  
    File inFile ) [static]
```

Load the project settings file

Parameters

<i>inFile</i>	Settings file
---------------	---------------

Returns

Error status

4.8.2.2 main()

```
static void gui.MainGui.main (  
    String [] args ) [static]
```

Main function

Parameters

<i>args</i>	TBD
-------------	-----

4.8.2.3 saveUc()

```
static void gui.MainGui.saveUc ( ) [static]
```

Save the microcontroller's configuration to disk

4.8.2.4 setNewUC()

```
static void gui.MainGui.setNewUC (
    Microcontroller uC ) [static]
```

Set the project's microcontroller configuration

Parameters

<i>uC</i>	Microcontroller configuration
-----------	-------------------------------

4.8.2.5 showAboutWindow()

```
static void gui.MainGui.showAboutWindow ( ) [static]
```

Show about information window

4.8.2.6 showErrorDialog()

```
static void gui.MainGui.showErrorDialog (
    String message ) [static]
```

Show an error dialog

Parameters

<i>message</i>	Message to display
----------------	--------------------

4.8.2.7 showGpioConfWindow()

```
static void gui.MainGui.showGpioConfWindow ( ) [static]
```


Show the GPIOs configuration window

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

- src/gui/MainGui.java

4.9 gui.MainWindow Class Reference

Public Member Functions

- [MainWindow](#) ()
- void [setVisible](#) (boolean status)
- File [OpenFileChooser](#) (String initialPath, String title, FileNameExtensionFilter fileFilter)
- [ErrorCode](#) [setProjectInformation](#) (String projectName, String ucManufacturer, String ucName)

Static Public Member Functions

- static void [main](#) (String[] args)

Public Attributes

- JFrame **FrmCodeGenerator**

4.9.1 Detailed Description

Main application window

Author

Miguel Diaz

Version

0.1

4.9.2 Constructor & Destructor Documentation

4.9.2.1 MainWindow()

```
gui.MainWindow.MainWindow ( )
```

Create the application.

4.9.3 Member Function Documentation

4.9.3.1 main()

```
static void gui.MainWindow.main (
    String [] args ) [static]
```

Open main window

Parameters

<i>args</i>	To be determined
-------------	------------------

4.9.3.2 OpenFileDialog()

```
File gui.MainWindow.OpenFileChooser (
    String initialPath,
    String title,
    FileNameExtensionFilter fileFilter )
```

Open file chooser dialog and get the selected file

Parameters

<i>initialPath</i>	Path to search the file in
<i>title</i>	Dialog title
<i>fileFilter</i>	Extension filter

Returns

Selected file

4.9.3.3 setProjectInformation()

```
ErrorCode gui.MainWindow.setProjectInformation (
    String projectName,
    String ucManufacturer,
    String ucName )
```

Set Project's name in its label

Parameters

<i>projectName</i>	Project's name
<i>ucManufacturer</i>	Microcontroller's manufacturer
<i>ucName</i>	Microcontroller's model

Returns

Error status

4.9.3.4 setVisible()

```
void gui.MainWindow.setVisible (
    boolean status )
```

Set visibility of About window

Parameters

<i>status</i>	true if visible
---------------	-----------------

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

- src/gui/MainWindow.java

4.10 gui.Messages Class Reference

Static Public Member Functions

- static String **getString** (String key)

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

- src/gui/Messages.java

4.11 microcontroller.Microcontroller Class Reference

Public Member Functions

- [Microcontroller](#) (Document ucDoc)
- [ErrorCode processDocument](#) ()
- [ErrorCode loadPinsConf](#) (Document confDoc)
- [Pin getPin](#) (int pinNum)
- [String getUc_model](#) ()
- [String getUc_manufacturer](#) ()
- [int getUc_pinNum](#) ()
- [int getUc_gpioNum](#) ()
- [int getUc_portNum](#) ()
- [PinConf getConfiguredPin](#) (String gpioName)
- [boolean isValid](#) ()

Public Attributes

- [String \[\] Ports](#)
- [PinConf \[\] GpioCfgPin](#)

Static Public Attributes

- static final int **MAX_NUMBER_OF_PINS_PER_PORT** = 16

4.11.1 Detailed Description

[Microcontroller](#) related methods

Author

Miguel Diaz

Version

0.1

4.11.2 Constructor & Destructor Documentation

4.11.2.1 Microcontroller()

```
microcontroller.Microcontroller.Microcontroller (
    Document ucDoc )
```

Constructor

Parameters

<i>ucDoc</i>	Document obtained from XML file
--------------	---------------------------------

4.11.3 Member Function Documentation

4.11.3.1 getConfiguredPin()

```
PinConf microcontroller.Microcontroller.getConfiguredPin (
    String gpioName )
```

Get the configuration of a pin

Parameters

<i>gpioName</i>	Name of the pin
-----------------	-----------------

Returns

Pin configuration

4.11.3.2 getPin()

```
Pin microcontroller.Microcontroller.getPin (
    int pinNum )
```

Get a pin's characteristics

Parameters

<i>pinNum</i>	Number of pin
---------------	---------------

Returns

Pin's characteristics

4.11.3.3 `getUc_gpioNum()`

```
int microcontroller.Microcontroller.getUc_gpioNum ( )
```

Get the number of GPIOs in the microcontroller

Returns

Number of GPIOs

4.11.3.4 `getUc_manufacturer()`

```
String microcontroller.Microcontroller.getUc_manufacturer ( )
```

Get the microcontroller's manufacturer

Returns

[Microcontroller](#)'s manufacturer

4.11.3.5 `getUc_model()`

```
String microcontroller.Microcontroller.getUc_model ( )
```

Get the microcontroller's model

Returns

[Microcontroller](#)'s model

4.11.3.6 `getUc_pinNum()`

```
int microcontroller.Microcontroller.getUc_pinNum ( )
```

Get the microcontroller's pins number

Returns

Number of pins

4.11.3.7 getUc_portNum()

```
int microcontroller.Microcontroller.getUc_portNum ( )
```

Get the number of ports in the microcontroller

Returns

Number of ports

4.11.3.8 isValid()

```
boolean microcontroller.Microcontroller.isValid ( )
```

Check if the microcontroller configuration is valid

Returns

true if valid

4.11.3.9 processDocument()

```
ErrorCode microcontroller.Microcontroller.processDocument ( )
```

Process the document obtained from XML file

Returns

Error status

4.11.4 Member Data Documentation

4.11.4.1 GpioCfgPin

```
PinConf [ ] microcontroller.Microcontroller.GpioCfgPin
```

Configured pins list

4.11.4.2 Ports

```
String [] microcontroller.Microcontroller.Ports
```

Ports name list

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

- src/microcontroller/Microcontroller.java

4.12 configurator.GPIO.Mode Enum Reference

Static Public Member Functions

- static [Mode getConfigFromString](#) (String conf)

Public Attributes

- **MODE_INPUT**
- **MODE_OUTPUT**
- **MODE_ALTERNATE_FUNCTION**
- **MODE_ANALOG**

Static Public Attributes

- static final String [STR_NAME](#) = "Mode"

4.12.1 Detailed Description

GPIO modes

Author

Miguel Diaz

Version

0.1

4.12.2 Member Function Documentation

4.12.2.1 getConfigFromString()

```
static Mode configurator.GPIO.Mode.getConfigFromString (  
    String conf ) [static]
```

Get the corresponding mode from its name as String

Parameters

<i>conf</i>	Configuration name
-------------	--------------------

Returns

[Mode](#)

4.12.3 Member Data Documentation

4.12.3.1 STR_NAME

```
static final String configurator.GPIO.Mode.STR_NAME = "Mode" [static]
```

Name as String

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

- `src/configurator/GPIO/Mode.java`

4.13 configurator.GPIO.OutLevel Enum Reference

Static Public Member Functions

- static [OutLevel getConfigFromString](#) (String conf)

Public Attributes

- **LOW**
- **HIGH**

Static Public Attributes

- static final String [STR_NAME](#) = "OutLevel"

4.13.1 Detailed Description

Pin's output/input level

Author

Miguel Diaz

Version

0.1

4.13.2 Member Function Documentation

4.13.2.1 getConfigFromString()

```
static OutLevel configurator.GPIO.OutLevel.getConfigFromString (  
    String conf ) [static]
```

Get the corresponding mode from its name as String

Parameters

<i>conf</i>	Configuration name
-------------	--------------------

Returns

level

4.13.3 Member Data Documentation

4.13.3.1 STR_NAME

```
static final String configurator.GPIO.OutLevel.STR_NAME = "OutLevel" [static]
```

Name as String

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

- src/configurator/GPIO/OutLevel.java

4.14 configurator.GPIO.OutType Enum Reference

Static Public Member Functions

- static [OutType](#) `getConfigFromString` (String *conf*)

Public Attributes

- `OTYPE_PUSH_PULL`
- `OTYPE_OPEN_DRAIN`
- `OTYPE_NOT_AVAILABLE`

Static Public Attributes

- static final String `STR_NAME` = "OutType"

4.14.1 Detailed Description

Pin's output type

Author

Miguel Diaz

Version

0.1

4.14.2 Member Function Documentation

4.14.2.1 `getConfigFromString()`

```
static OutType configurator.GPIO.OutType.getConfigFromString (  
    String conf ) [static]
```

Get the corresponding output type from its name as String

Parameters

<i>conf</i>	Configuration name
-------------	--------------------

Returns

Output type

4.14.3 Member Data Documentation

4.14.3.1 STR_NAME

```
static final String configurator.GPIO.OutType.STR_NAME = "OutType" [static]
```

Name as String

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

- src/configurator/GPIO/OutType.java

4.15 microcontroller.Pin Class Reference

Public Member Functions

- [Pin](#) ()
- void [setFunc_vcc](#) (boolean funcState)
- boolean [getFunc_vcc](#) ()
- void [setFunc_gnd](#) (boolean funcState)
- boolean [getFunc_gnd](#) ()
- void [setFunc_gpio](#) (boolean funcState)
- boolean [getFunc_gpio](#) ()
- void [setFeat_int](#) (boolean featState)
- boolean [getFeat_int](#) ()
- void [setFeat_adc](#) (boolean featState)
- boolean [getFeat_adc](#) ()
- void [setFeat_uart](#) (boolean featState)
- boolean [getFeat_uart](#) ()
- void [setFeat_i2c](#) (boolean featState)
- boolean [getFeat_i2c](#) ()
- void [setFeat_spi](#) (boolean featState)
- boolean [getFeat_spi](#) ()
- void [setFeat_clock](#) (boolean featState)
- boolean [getFeat_clock](#) ()
- void [setFeat_timer](#) (boolean featState)
- boolean [getFeat_timer](#) ()
- void [setFeat_reset](#) (boolean featState)
- boolean [getFeat_reset](#) ()
- void [setInt](#) (String feature)

- String `getInt ()`
- void `setAdc (String feature)`
- String `getAdc ()`
- void `setUart (String feature)`
- String `getUart ()`
- void `setI2c (String feature)`
- String `getI2c ()`
- void `setSpi (String feature)`
- String `getSpi ()`
- void `setClock (String feature)`
- String `getClock ()`
- void `setReset (String feature)`
- String `getReset ()`
- void `setTimer (String feature)`
- String `getTimer ()`
- void `setName (String pinName)`
- String `getName ()`
- void `setNumber (int pinNum)`
- int `getNumber ()`
- void `setPort (String pinPort)`
- String `getPort ()`
- boolean `isValid ()`

Static Public Attributes

- static final boolean `ENABLE` = true
- static final boolean `DISABLE` = false
- static final boolean `DEF_FUNCTION` = DEF_BOOLEAN
- static final boolean `DEF_FEATURE_AV` = DEF_BOOLEAN
- static final String `DEF_FEATURE` = DEF_STRING
- static final String `DEF_NAME` = DEF_STRING
- static final int `DEF_NUMBER` = DEF_INT
- static final String `DEF_PORT` = DEF_STRING

4.15.1 Detailed Description

Basic pin object.

- `Pin` necessary characteristics:
 - Name
 - Number
- `Pin` optional characteristics:
 - Port
- `Pin` main functions:
 - VCC

- GND
 - GPIO
- [Pin](#) features:
 - Interruption
 - ADC
 - UART
 - I2C
 - SPI
 - Clock
 - Reset

Author

Miguel Diaz

Version

0.1

4.15.2 Constructor & Destructor Documentation

4.15.2.1 `Pin()`

```
microcontroller.Pin.Pin ( )
```

Initialize all pin's characteristics and features to their default values

4.15.3 Member Function Documentation

4.15.3.1 `getAdc()`

```
String microcontroller.Pin.getAdc ( )
```

Get the pin's ADC name

Returns

[Pin's](#) ADC

4.15.3.2 `getClock()`

```
String microcontroller.Pin.getClock ( )
```

Get the pin's clock name

Returns

Pin's clock

4.15.3.3 `getFeat_adc()`

```
boolean microcontroller.Pin.getFeat_adc ( )
```

See if the pin has an ADC

Returns

Feature availability

4.15.3.4 `getFeat_clock()`

```
boolean microcontroller.Pin.getFeat_clock ( )
```

See if the pin supports a clock

Returns

Feature availability

4.15.3.5 `getFeat_i2c()`

```
boolean microcontroller.Pin.getFeat_i2c ( )
```

See if the pin has I2C

Returns

Feature availability

4.15.3.6 `getFeat_int()`

```
boolean microcontroller.Pin.getFeat_int ( )
```

See if the pin has an interruption

Returns

Feature availability

4.15.3.7 `getFeat_reset()`

```
boolean microcontroller.Pin.getFeat_reset ( )
```

See if the pin has a reset feature

Returns

Feature availability

4.15.3.8 `getFeat_spi()`

```
boolean microcontroller.Pin.getFeat_spi ( )
```

See if the pin has SPI

Returns

Feature availability

4.15.3.9 `getFeat_timer()`

```
boolean microcontroller.Pin.getFeat_timer ( )
```

See if the pin supports a timer

Returns

Feature availability

4.15.3.10 getFeat_uart()

```
boolean microcontroller.Pin.getFeat_uart ( )
```

See if the pin has a UART

Returns

Feature availability

4.15.3.11 getFunc_gnd()

```
boolean microcontroller.Pin.getFunc_gnd ( )
```

See if the pin is GND

Returns

Function availability

4.15.3.12 getFunc_gpio()

```
boolean microcontroller.Pin.getFunc_gpio ( )
```

See if the pin is GPIO

Returns

Function availability

4.15.3.13 getFunc_vcc()

```
boolean microcontroller.Pin.getFunc_vcc ( )
```

See if the pin is Vcc

Returns

Function availability

4.15.3.14 `getI2c()`

```
String microcontroller.Pin.getI2c ( )
```

Get the pin's I2C name

Returns

[Pin's I2C](#)

4.15.3.15 `getInt()`

```
String microcontroller.Pin.getInt ( )
```

Get the pin's interruption name

Returns

[Pin's interruption](#)

4.15.3.16 `getName()`

```
String microcontroller.Pin.getName ( )
```

Get the pin's name

Returns

[Pin's name](#)

4.15.3.17 `getNumber()`

```
int microcontroller.Pin.getNumber ( )
```

Get the pin's number

Returns

[Pin's number](#)

4.15.3.18 getPort()

```
String microcontroller.Pin.getPort ( )
```

Get the pin's port

Returns

Pin's port

4.15.3.19 getReset()

```
String microcontroller.Pin.getReset ( )
```

Get the pin's reset name

Returns

Pin's reset

4.15.3.20 getSpi()

```
String microcontroller.Pin.getSpi ( )
```

Get the pin's SPI name

Returns

Pin's SPI

4.15.3.21 getTimer()

```
String microcontroller.Pin.getTimer ( )
```

Get the pin's timer name

Returns

Pin's timer

4.15.3.22 `getUart()`

```
String microcontroller.Pin.getUart ( )
```

Get the pin's UART name

Returns

Pin's UART

4.15.3.23 `isValid()`

```
boolean microcontroller.Pin.isValid ( )
```

Check if the pin is correctly initialized

Returns

True if the pin is correctly initialized

4.15.3.24 `setAdc()`

```
void microcontroller.Pin.setAdc (
    String feature )
```

Set the pin's ADC

Parameters

<i>feature</i>	Pin's ADC
----------------	-----------

4.15.3.25 `setClock()`

```
void microcontroller.Pin.setClock (
    String feature )
```

Set the pin's clock

Parameters

<i>feature</i>	Pin's clock
----------------	-------------

4.15.3.26 setFeat_adc()

```
void microcontroller.Pin.setFeat_adc (
    boolean featState )
```

Set the pin's ADC feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.27 setFeat_clock()

```
void microcontroller.Pin.setFeat_clock (
    boolean featState )
```

Set the pin's Clock feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.28 setFeat_i2c()

```
void microcontroller.Pin.setFeat_i2c (
    boolean featState )
```

Set the pin's I2C feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.29 setFeat_int()

```
void microcontroller.Pin.setFeat_int (
    boolean featState )
```

Set the pin's interruption feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.30 setFeat_reset()

```
void microcontroller.Pin.setFeat_reset (
    boolean featState )
```

Set the pin's reset feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.31 setFeat_spi()

```
void microcontroller.Pin.setFeat_spi (
    boolean featState )
```

Set the pin's SPI feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.32 setFeat_timer()

```
void microcontroller.Pin.setFeat_timer (
```

```
boolean featState )
```

Set the pin's timer feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.33 setFeat_uart()

```
void microcontroller.Pin.setFeat_uart (
    boolean featState )
```

Set the pin's UART feature

Parameters

<i>featState</i>	Feature availability
------------------	----------------------

4.15.3.34 setFunc_gnd()

```
void microcontroller.Pin.setFunc_gnd (
    boolean funcState )
```

Set the pin to GND status

Parameters

<i>funcState</i>	Function availability
------------------	-----------------------

4.15.3.35 setFunc_gpio()

```
void microcontroller.Pin.setFunc_gpio (
    boolean funcState )
```

Set the pin to GPIO status

Parameters

<i>funcState</i>	Function availability
------------------	-----------------------

4.15.3.36 setFunc_vcc()

```
void microcontroller.Pin.setFunc_vcc (  
    boolean funcState )
```

Set the pin to Vcc status

Parameters

<i>funcState</i>	Function availability
------------------	-----------------------

4.15.3.37 setI2c()

```
void microcontroller.Pin.setI2c (  
    String feature )
```

Set the pin's I2C

Parameters

<i>feature</i>	Pin's I2C
----------------	---------------------------

4.15.3.38 setInt()

```
void microcontroller.Pin.setInt (  
    String feature )
```

Set the pin's interruption

Parameters

<i>feature</i>	Pin's interruption
----------------	------------------------------------

4.15.3.39 setName()

```
void microcontroller.Pin.setName (
    String pinName )
```

Set the pin's name

Parameters

<i>pinName</i>	Pin's name
----------------	------------

4.15.3.40 setNumber()

```
void microcontroller.Pin.setNumber (
    int pinNum )
```

Set the pin's number

Parameters

<i>pinNum</i>	Pin's number
---------------	--------------

4.15.3.41 setPort()

```
void microcontroller.Pin.setPort (
    String pinPort )
```

Set the pin's port

Parameters

<i>pinPort</i>	Pin's port
----------------	------------

4.15.3.42 setReset()

```
void microcontroller.Pin.setReset (
```

```
String feature )
```

Set the pin's reset

Parameters

<i>feature</i>	Pin's reset
----------------	-------------

4.15.3.43 setSpi()

```
void microcontroller.Pin.setSpi (
    String feature )
```

Set the pin's SPI

Parameters

<i>feature</i>	Pin's SPI
----------------	-----------

4.15.3.44 setTimer()

```
void microcontroller.Pin.setTimer (
    String feature )
```

Set the pin's timer

Parameters

<i>feature</i>	Pin's timer
----------------	-------------

4.15.3.45 setUart()

```
void microcontroller.Pin.setUart (
    String feature )
```

Set the pin's UART

Parameters

<i>feature</i>	Pin's UART
----------------	------------

4.15.4 Member Data Documentation

4.15.4.1 DEF_FEATURE

```
final String microcontroller.Pin.DEF_FEATURE = DEF_STRING [static]
```

Default value for pin's feature as not available

4.15.4.2 DEF_FEATURE_AV

```
final boolean microcontroller.Pin.DEF_FEATURE_AV = DEF_BOOLEAN [static]
```

Default value for pin's feature availability as not available

4.15.4.3 DEF_FUNCTION

```
final boolean microcontroller.Pin.DEF_FUNCTION = DEF_BOOLEAN [static]
```

Default value for pin's function as not enabled

4.15.4.4 DEF_NAME

```
final String microcontroller.Pin.DEF_NAME = DEF_STRING [static]
```

Default value for pin's name

4.15.4.5 DEF_NUMBER

```
final int microcontroller.Pin.DEF_NUMBER = DEF_INT [static]
```

Default value for pin's number

4.15.4.6 DEF_PORT

```
final String microcontroller.Pin.DEF_PORT = DEF_STRING [static]
```

Default value for pin's port

4.15.4.7 DISABLE

```
final boolean microcontroller.Pin.DISABLE = false [static]
```

Disable value for features and functions

4.15.4.8 ENABLE

```
final boolean microcontroller.Pin.ENABLE = true [static]
```

Enable value for features and functions

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

- src/microcontroller/Pin.java

4.16 configurator.PinConf Class Reference

Public Member Functions

- [PinConf](#) ([Pin](#) gpioPin)
- boolean [isValid](#) ()
- String [getPort](#) ()
- String [getPinName](#) ()
- String [getCodeName](#) ()
- void [setCodeName](#) (String name)
- [Mode](#) [getMode](#) ()
- void [setMode](#) ([Mode](#) mode)
- [OutType](#) [getOutType](#) ()
- void [setOutType](#) ([OutType](#) outType)
- [OutLevel](#) [getOutLevel](#) ()
- void [setOutLevel](#) ([OutLevel](#) level)
- [Speed](#) [getSpeed](#) ()
- void [setSpeed](#) ([Speed](#) speed)
- [Pull](#) [getPull](#) ()
- void [setPull](#) ([Pull](#) pull)
- boolean [isAv_Adc](#) ()
- boolean [isAv_altFunc](#) ()

Static Public Attributes

- static final [Mode](#) [DF_MODE](#) = Mode.MODE_INPUT
- static final [Speed](#) [DF_SPEED](#) = Speed.SPEED_FAST
- static final [OutType](#) [DF_OUTTYPE](#) = OutType.OTYPE_PUSH_PULL
- static final [OutLevel](#) [DF_OUT_LEVEL](#) = OutLevel.LOW
- static final [Pull](#) [DF_PULL](#) = Pull.PULL_NOT_AVAILABLE
- static final String [DF_CODE_NAME](#) = ""

4.16.1 Detailed Description

GPIO pin configuration

Author

Miguel Diaz

Version

0.1

4.16.2 Constructor & Destructor Documentation

4.16.2.1 PinConf()

```
configurator.PinConf.PinConf (
    Pin gpioPin )
```

Constructor

Parameters

<i>gpioPin</i>	Pin information
----------------	-----------------

4.16.3 Member Function Documentation

4.16.3.1 getCodeName()

```
String configurator.PinConf.getCodeName ( )
```

Get the pin's user selected name

Returns

pin's name

4.16.3.2 `getMode()`

```
Mode configurator.PinConf.getMode ( )
```

Get the pin's mode configuration

Returns

Mode

4.16.3.3 `getOutLevel()`

```
OutLevel configurator.PinConf.getOutLevel ( )
```

Get the pin's output level

Returns

Pin's output level

4.16.3.4 `getOutType()`

```
OutType configurator.PinConf.getOutType ( )
```

Get the pin's output configuration

Returns

Output configuration

4.16.3.5 getPinName()

```
String configurator.PinConf.getPinName ( )
```

Get the pin's number

Returns

Pin's number

4.16.3.6 getPort()

```
String configurator.PinConf.getPort ( )
```

Get the pin's port

Returns

Port

4.16.3.7 getPull()

```
Pull configurator.PinConf.getPull ( )
```

Get the pin's pull resistor configuration

Returns

Pull Resistor configuration

4.16.3.8 getSpeed()

```
Speed configurator.PinConf.getSpeed ( )
```

Get the pin's speed

Returns

Speed

4.16.3.9 isAv_Adc()

```
boolean configurator.PinConf.isAv_Adc ( )
```

Check availability of ADC

Returns

True if ADC is available

4.16.3.10 isAv_altFunc()

```
boolean configurator.PinConf.isAv_altFunc ( )
```

Check the availability of alternate function

Returns

True if alternate function is available

4.16.3.11 isValid()

```
boolean configurator.PinConf.isValid ( )
```

Check if the GPIO pin is valid

Returns

True if valid

4.16.3.12 setCodeName()

```
void configurator.PinConf.setCodeName (
    String name )
```

Set the pin's user selected name

Parameters

<i>name</i>	Pin's name
-------------	------------

4.16.3.13 setMode()

```
void configurator.PinConf.setMode (
    Mode mode )
```

Set the pin's mode configuration

Parameters

<i>mode</i>	Mode
-------------	------

4.16.3.14 setOutLevel()

```
void configurator.PinConf.setOutLevel (
    OutLevel level )
```

Set the pin's output level

Parameters

<i>level</i>	Pin's output level
--------------	--------------------

4.16.3.15 setOutType()

```
void configurator.PinConf.setOutType (
    OutType outType )
```

Set the pin's output configuration

Parameters

<i>outType</i>	Output configuration
----------------	----------------------

4.16.3.16 setPull()

```
void configurator.PinConf.setPull (
    Pull pull )
```

Set the pull resistor configuration

Parameters

<i>pull</i>	Resistor configuration
-------------	------------------------

4.16.3.17 setSpeed()

```
void configurator.PinConf.setSpeed (
    Speed speed )
```

Set the pin's speed

Parameters

<i>speed</i>	Speed
--------------	-------

4.16.4 Member Data Documentation

4.16.4.1 DF_MODE

```
final Mode configurator.PinConf.DF_MODE = Mode.MODE_INPUT [static]
```

Default Pin mode

4.16.4.2 DF_OUT_LEVEL

```
final OutLevel configurator.PinConf.DF_OUT_LEVEL = OutLevel.LOW [static]
```

Default pin's output level

4.16.4.3 DF_OUTTYPE

```
final OutType configurator.PinConf.DF_OUTTYPE = OutType.OTYPE_PUSH_PULL [static]
```

Default pin's output type

4.16.4.4 DF_PULL

```
final Pull configurator.PinConf.DF_PULL = Pull.PULL_NOT_AVAILABLE [static]
```

Default pin's pull resistor

4.16.4.5 DF_SPEED

```
final Speed configurator.PinConf.DF_SPEED = Speed.SPEED_FAST [static]
```

Default pin's speed

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

- `src/configurator/PinConf.java`

4.17 projectConfiguration.ProjectSettings Class Reference

Public Member Functions

- [ProjectSettings](#) ()
- [ErrorCode](#) [processDocument](#) ()
- [ErrorCode](#) [openProjectFile](#) (File inFile)
- File [getConfFile](#) ()
- File [getUcFile](#) ()
- String [getProjectName](#) ()

4.17.1 Detailed Description

Project settings class

Author

Miguel Diaz

Version

0.1

4.17.2 Constructor & Destructor Documentation

4.17.2.1 ProjectSettings()

```
projectConfiguration.ProjectSettings.ProjectSettings ( )
```

Constructor

4.17.3 Member Function Documentation

4.17.3.1 getConfFile()

```
File projectConfiguration.ProjectSettings.getConfFile ( )
```

Get the project configuration file

Returns

Project configuration file

4.17.3.2 getProjectName()

```
String projectConfiguration.ProjectSettings.getProjectName ( )
```

Get the project's name

Returns

Project's name

4.17.3.3 getUcFile()

```
File projectConfiguration.ProjectSettings.getUcFile ( )
```

Get the project microcontroller file

Returns

Project microcontroller file

4.17.3.4 openProjectFile()

```
ErrorCode projectConfiguration.ProjectSettings.openProjectFile (
    File inFile )
```

Open the project settings file

Parameters

<i>inFile</i>	Project file
---------------	--------------

Returns

Error Status

4.17.3.5 processDocument()

```
ErrorCode projectConfiguration.ProjectSettings.processDocument ( )
```

Process the document obtained from the XML file

Returns

Error Status

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

- src/projectConfiguration/ProjectSettings.java

4.18 configurator.GPIO.Pull Enum Reference

Static Public Member Functions

- static [Pull getConfigFromString](#) (String conf)

Public Attributes

- **PULL_UP**
- **PULL_DOWN**
- **PULL_NOT_AVAILABLE**

Static Public Attributes

- static final String [STR_NAME](#) = "Pull"

4.18.1 Detailed Description

Pin's pull resistor

Author

Miguel Diaz

Version

0.1

4.18.2 Member Function Documentation

4.18.2.1 getConfigFromString()

```
static Pull configurator.GPIO.Pull.getConfigFromString (  
    String conf ) [static]
```

Get the corresponding [Pull](#) configuration from its name as String

Parameters

<i>conf</i>	Configuration name
-------------	--------------------

Returns

[Pull](#) configuration

4.18.3 Member Data Documentation

4.18.3.1 STR_NAME

```
static final String configurator.GPIO.Pull.STR_NAME = "Pull" [static]
```

Name as String

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

- src/configurator/GPIO/Pull.java

4.19 configurator.GPIO.Speed Enum Reference

Static Public Member Functions

- static [Speed](#) [getConfFromString](#) (String conf)

Public Attributes

- **SPEED_FAST**
- **SPEED_MEDIUM**
- **SPEED_HIGH**
- **SPEED_NOT_AVAILABLE**

Static Public Attributes

- static final String [STR_NAME](#) = "Speed"

4.19.1 Detailed Description

Pin's speed

Author

Miguel Diaz

Version

0.1

4.19.2 Member Function Documentation

4.19.2.1 [getConfFromString\(\)](#)

```
static Speed configurator.GPIO.Speed.getConfFromString (  
    String conf ) [static]
```

Get the corresponding [Speed](#) configuration from its name as String

Parameters

<i>conf</i>	Configuration name
-------------	--------------------

Returns

[Speed](#)

4.19.3 Member Data Documentation

4.19.3.1 STR_NAME

```
static final String configurator.GPIO.Speed.STR_NAME = "Speed" [static]
```

Name as String

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

- `src/configurator/GPIO/Speed.java`

4.20 xmlParser.TestMain Class Reference

Static Public Member Functions

- static void [main](#) (String[] openOption)

4.20.1 Detailed Description

Dummy main class for testing the other classes

Author

Miguel Diaz

4.20.2 Member Function Documentation

4.20.2.1 main()

```
static void xmlParser.TestMain.main (  
    String [] openOption ) [static]
```

Main without GUI

Parameters

<i>openOption</i>	Options include:
-------------------	---------------------

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

- src/xmlParser/TestMain.java

4.21 xmlParser.XmlOpener Class Reference

Public Member Functions

- [XmlOpener](#) ()
- [ErrorCode OpenFile](#) (File inFile)
- Document [getParsedDoc](#) ()

Static Public Member Functions

- static String [getElementInfoFromDoc](#) (Document doc, String elementName)
- static String [getElementInfo](#) (Element element, String elementName)

4.21.1 Detailed Description

Open and process XML files

Author

H112943

Version

0.1

4.21.2 Constructor & Destructor Documentation

4.21.2.1 XmlOpener()

```
xmlParser.XmlOpener.XmlOpener ( )
```

Constructor

4.21.3 Member Function Documentation

4.21.3.1 getElementInfo()

```
static String xmlParser.XmlOpener.getElementInfo (
    Element element,
    String elementName ) [static]
```

Get an XML sub element information

Parameters

<i>element</i>	XML main element
<i>elementName</i>	Sub element's name

Returns

Sub element's information

4.21.3.2 getElementInfoFromDoc()

```
static String xmlParser.XmlOpener.getElementInfoFromDoc (
    Document doc,
    String elementName ) [static]
```

Get an XML element information

Parameters

<i>doc</i>	Document from XML file
<i>elementName</i>	Element's name

Returns

Element's information

4.21.3.3 getParsedDoc()

```
Document xmlParser.XmlOpener.getParsedDoc ( )
```

Get the parsed document AFTER opening the file

Returns

Parsed document

4.21.3.4 OpenFile()

```
ErrorCode xmlParser.XmlOpener.OpenFile (
    File inFile )
```

Open the XML file

Parameters

<i>inFile</i>	XML file
---------------	----------

Returns

Error code

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

- src/xmlParser/XmlOpener.java

Index

- AboutWindow
 - gui::AboutWindow, [9](#)
- addPin
 - xmlCreator::ConfXmlWriter, [12](#)
- common, [5](#)
- common.ErrorCode, [13](#)
- common.Features, [14](#)
- common::ErrorCode
 - EX_ERROR, [13](#)
 - FILE_READ_ERROR, [14](#)
 - FILE_WRITE_ERROR, [14](#)
 - NO_ERROR, [14](#)
 - STR_INVALID, [14](#)
- common::Features
 - DEBUG_STR, [16](#)
 - DEBUG, [16](#)
 - debugPrint, [15](#)
 - SW_VERSION, [16](#)
 - VERBOSE_STR, [16](#)
 - VERBOSE, [16](#)
 - VERSION_NAME, [16](#)
 - VERSION_STATUS, [17](#)
 - verbosePrint, [15](#)
- ConfXmlWriter
 - xmlCreator::ConfXmlWriter, [12](#)
- configurator, [5](#)
- configurator.ConfigurationFile, [10](#)
- configurator.GPIO.CodeName, [10](#)
- configurator.GPIO.Mode, [28](#)
- configurator.GPIO.OutLevel, [29](#)
- configurator.GPIO.OutType, [31](#)
- configurator.GPIO.Pull, [57](#)
- configurator.GPIO.Speed, [59](#)
- configurator.PinConf, [48](#)
- configurator::ConfigurationFile
 - STR_PROJ_CONF_FILE, [11](#)
- configurator::GPIO::CodeName
 - STR_NAME, [10](#)
- configurator::GPIO::Mode
 - getConfFromString, [28](#)
 - STR_NAME, [29](#)
- configurator::GPIO::OutLevel
 - getConfFromString, [30](#)
 - STR_NAME, [30](#)
- configurator::GPIO::OutType
 - getConfFromString, [31](#)
 - STR_NAME, [32](#)
- configurator::GPIO::Pull
 - getConfFromString, [58](#)
 - STR_NAME, [58](#)
- configurator::GPIO::Speed
 - getConfFromString, [59](#)
 - STR_NAME, [60](#)
- configurator::PinConf
 - DF_MODE, [54](#)
 - DF_OUT_LEVEL, [54](#)
 - DF_OUTTYPE, [54](#)
 - DF_PULL, [55](#)
 - DF_SPEED, [55](#)
 - getCodeName, [49](#)
 - getMode, [50](#)
 - getOutLevel, [50](#)
 - getOutType, [50](#)
 - getPinName, [50](#)
 - getPort, [51](#)
 - getPull, [51](#)
 - getSpeed, [51](#)
 - isAv_Adc, [51](#)
 - isAv_altFunc, [52](#)
 - isValid, [52](#)
 - PinConf, [49](#)
 - setCodeName, [52](#)
 - setMode, [53](#)
 - setOutLevel, [53](#)
 - setOutType, [53](#)
 - setPull, [54](#)
 - setSpeed, [54](#)
- DEBUG_STR
 - common::Features, [16](#)
- DEBUG
 - common::Features, [16](#)
- DEF_FEATURE_AV
 - microcontroller::Pin, [47](#)
- DEF_FEATURE
 - microcontroller::Pin, [47](#)
- DEF_FUNCTION
 - microcontroller::Pin, [47](#)
- DEF_NAME
 - microcontroller::Pin, [47](#)
- DEF_NUMBER

- microcontroller::Pin, 47
- DEF_PORT
 - microcontroller::Pin, 47
- DF_MODE
 - configurator::PinConf, 54
- DF_OUT_LEVEL
 - configurator::PinConf, 54
- DF_OUTTYPE
 - configurator::PinConf, 54
- DF_PULL
 - configurator::PinConf, 55
- DF_SPEED
 - configurator::PinConf, 55
- DISABLE
 - microcontroller::Pin, 48
- debugPrint
 - common::Features, 15
- ENABLE
 - microcontroller::Pin, 48
- EX_ERROR
 - common::ErrorCode, 13
- FILE_READ_ERROR
 - common::ErrorCode, 14
- FILE_WRITE_ERROR
 - common::ErrorCode, 14
- framework, 6
- getAdc
 - microcontroller::Pin, 34
- getClock
 - microcontroller::Pin, 34
- getCodeName
 - configurator::PinConf, 49
- getConfFile
 - projectConfiguration::ProjectSettings, 56
- getConfFromString
 - configurator::GPIO::Mode, 28
 - configurator::GPIO::OutLevel, 30
 - configurator::GPIO::OutType, 31
 - configurator::GPIO::Pull, 58
 - configurator::GPIO::Speed, 59
- getConfiguredPin
 - microcontroller::Microcontroller, 25
- getElementInfo
 - xmlParser::XmlOpener, 62
- getElementInfoFromDoc
 - xmlParser::XmlOpener, 62
- getFeat_adc
 - microcontroller::Pin, 35
- getFeat_clock
 - microcontroller::Pin, 35
- getFeat_i2c
 - microcontroller::Pin, 35
- getFeat_int
 - microcontroller::Pin, 35
- getFeat_reset
 - microcontroller::Pin, 36
- getFeat_spi
 - microcontroller::Pin, 36
- getFeat_timer
 - microcontroller::Pin, 36
- getFeat_uart
 - microcontroller::Pin, 36
- getFunc_gnd
 - microcontroller::Pin, 37
- getFunc_gpio
 - microcontroller::Pin, 37
- getFunc_vcc
 - microcontroller::Pin, 37
- getI2c
 - microcontroller::Pin, 37
- getInt
 - microcontroller::Pin, 38
- getMode
 - configurator::PinConf, 50
- getName
 - microcontroller::Pin, 38
- getNumber
 - microcontroller::Pin, 38
- getOutLevel
 - configurator::PinConf, 50
- getOutType
 - configurator::PinConf, 50
- getParsedDoc
 - xmlParser::XmlOpener, 62
- getPin
 - microcontroller::Microcontroller, 25
- getPinName
 - configurator::PinConf, 50
- getPort
 - configurator::PinConf, 51
 - microcontroller::Pin, 38
- getProjectName
 - projectConfiguration::ProjectSettings, 56
- getPull
 - configurator::PinConf, 51
- getReset
 - microcontroller::Pin, 39
- getSpeed
 - configurator::PinConf, 51
- getSpi
 - microcontroller::Pin, 39
- getTimer
 - microcontroller::Pin, 39
- getUart
 - microcontroller::Pin, 39
- getUc_gpioNum

- microcontroller::Microcontroller, 25
- getUc_manufacturer
 - microcontroller::Microcontroller, 26
- getUc_model
 - microcontroller::Microcontroller, 26
- getUc_pinNum
 - microcontroller::Microcontroller, 26
- getUc_portNum
 - microcontroller::Microcontroller, 26
- getUcFile
 - projectConfiguration::ProjectSettings, 56
- GpioCfgPin
 - microcontroller::Microcontroller, 27
- GpioConfWindow
 - gui::GpioConfWindow, 17
- gui, 6
- gui.AboutWindow, 9
- gui.GpioConfWindow, 17
- gui.MainGui, 18
- gui.MainWindow, 21
- gui.Messages, 23
- gui::AboutWindow
 - AboutWindow, 9
 - main, 9
- gui::GpioConfWindow
 - GpioConfWindow, 17
 - main, 18
- gui::MainGui
 - loadProjectFile, 19
 - main, 19
 - saveUc, 20
 - setNewUC, 20
 - showAboutWindow, 20
 - showErrorDialog, 20
 - showGpioConfWindow, 20
- gui::MainWindow
 - main, 22
 - MainWindow, 21
 - OpenFileChooser, 22
 - setProjectInformation, 22
 - setVisible, 23
- isAv_Adc
 - configurator::PinConf, 51
- isAv_altFunc
 - configurator::PinConf, 52
- isValid
 - configurator::PinConf, 52
 - microcontroller::Microcontroller, 27
 - microcontroller::Pin, 40
- loadProjectFile
 - gui::MainGui, 19
- main
 - gui::AboutWindow, 9
 - gui::GpioConfWindow, 18
 - gui::MainGui, 19
 - gui::MainWindow, 22
 - xmlParser::TestMain, 60
- MainWindow
 - gui::MainWindow, 21
- Microcontroller
 - microcontroller::Microcontroller, 24
- microcontroller, 7
- microcontroller.Microcontroller, 24
- microcontroller.Pin, 32
- microcontroller::Microcontroller
 - getConfiguredPin, 25
 - getPin, 25
 - getUc_gpioNum, 25
 - getUc_manufacturer, 26
 - getUc_model, 26
 - getUc_pinNum, 26
 - getUc_portNum, 26
 - GpioCfgPin, 27
 - isValid, 27
 - Microcontroller, 24
 - Ports, 27
 - processDocument, 27
- microcontroller::Pin
 - DEF_FEATURE_AV, 47
 - DEF_FEATURE, 47
 - DEF_FUNCTION, 47
 - DEF_NAME, 47
 - DEF_NUMBER, 47
 - DEF_PORT, 47
 - DISABLE, 48
 - ENABLE, 48
 - getAdc, 34
 - getClock, 34
 - getFeat_adc, 35
 - getFeat_clock, 35
 - getFeat_i2c, 35
 - getFeat_int, 35
 - getFeat_reset, 36
 - getFeat_spi, 36
 - getFeat_timer, 36
 - getFeat_uart, 36
 - getFunc_gnd, 37
 - getFunc_gpio, 37
 - getFunc_vcc, 37
 - getI2c, 37
 - getInt, 38
 - getName, 38
 - getNumber, 38
 - getPort, 38
 - getReset, 39
 - getSpi, 39

- getTimer, 39
- getUart, 39
- isValid, 40
- Pin, 34
- setAdc, 40
- setClock, 40
- setFeat_adc, 41
- setFeat_clock, 41
- setFeat_i2c, 41
- setFeat_int, 42
- setFeat_reset, 42
- setFeat_spi, 42
- setFeat_timer, 42
- setFeat_uart, 43
- setFunc_gnd, 43
- setFunc_gpio, 43
- setFunc_vcc, 44
- setI2c, 44
- setInt, 44
- setName, 45
- setNumber, 45
- setPort, 45
- setReset, 45
- setSpi, 46
- setTimer, 46
- setUart, 46
- NO_ERROR
 - common::ErrorCode, 14
- OpenFile
 - xmlParser::XmlOpener, 63
- OpenFileChooser
 - gui::MainWindow, 22
- openProjectFile
 - projectConfiguration::ProjectSettings, 56
- Pin
 - microcontroller::Pin, 34
- PinConf
 - configurator::PinConf, 49
- Ports
 - microcontroller::Microcontroller, 27
- processDocument
 - microcontroller::Microcontroller, 27
 - projectConfiguration::ProjectSettings, 57
- projectConfiguration, 7
- projectConfiguration.ProjectSettings, 55
- projectConfiguration::ProjectSettings
 - getConfigFile, 56
 - getProjectName, 56
 - getUcFile, 56
 - openProjectFile, 56
 - processDocument, 57
 - ProjectSettings, 56
- ProjectSettings
 - projectConfiguration::ProjectSettings, 56
- STR_INVALID
 - common::ErrorCode, 14
- STR_NAME
 - configurator::GPIO::CodeName, 10
 - configurator::GPIO::Mode, 29
 - configurator::GPIO::OutLevel, 30
 - configurator::GPIO::OutType, 32
 - configurator::GPIO::Pull, 58
 - configurator::GPIO::Speed, 60
- STR_PROJ_CONF_FILE
 - configurator::ConfigurationFile, 11
- SW_VERSION
 - common::Features, 16
- saveUc
 - gui::MainGui, 20
- setAdc
 - microcontroller::Pin, 40
- setClock
 - microcontroller::Pin, 40
- setCodeName
 - configurator::PinConf, 52
- setFeat_adc
 - microcontroller::Pin, 41
- setFeat_clock
 - microcontroller::Pin, 41
- setFeat_i2c
 - microcontroller::Pin, 41
- setFeat_int
 - microcontroller::Pin, 42
- setFeat_reset
 - microcontroller::Pin, 42
- setFeat_spi
 - microcontroller::Pin, 42
- setFeat_timer
 - microcontroller::Pin, 42
- setFeat_uart
 - microcontroller::Pin, 43
- setFunc_gnd
 - microcontroller::Pin, 43
- setFunc_gpio
 - microcontroller::Pin, 43
- setFunc_vcc
 - microcontroller::Pin, 44
- setI2c
 - microcontroller::Pin, 44
- setInt
 - microcontroller::Pin, 44
- setMode
 - configurator::PinConf, 53
- setName
 - microcontroller::Pin, 45

- setNewUC
 - gui::MainGui, [20](#)
- setNumber
 - microcontroller::Pin, [45](#)
- setOutLevel
 - configurator::PinConf, [53](#)
- setOutType
 - configurator::PinConf, [53](#)
- setPort
 - microcontroller::Pin, [45](#)
- setProjectInformation
 - gui::MainWindow, [22](#)
- setPull
 - configurator::PinConf, [54](#)
- setReset
 - microcontroller::Pin, [45](#)
- setSpeed
 - configurator::PinConf, [54](#)
- setSpi
 - microcontroller::Pin, [46](#)
- setTimer
 - microcontroller::Pin, [46](#)
- setUart
 - microcontroller::Pin, [46](#)
- setVisible
 - gui::MainWindow, [23](#)
- showAboutWindow
 - gui::MainGui, [20](#)
- showErrorDialog
 - gui::MainGui, [20](#)
- showGpioConfWindow
 - gui::MainGui, [20](#)
- VERBOSE_STR
 - common::Features, [16](#)
- VERBOSE
 - common::Features, [16](#)
- VERSION_NAME
 - common::Features, [16](#)
- VERSION_STATUS
 - common::Features, [17](#)
- verbosePrint
 - common::Features, [15](#)
- writeXml
 - xmlCreator::ConfXmlWriter, [12](#)
- xmlCreator, [7](#)
- xmlCreator.ConfXmlWriter, [11](#)
- xmlCreator::ConfXmlWriter
 - addPin, [12](#)
 - ConfXmlWriter, [12](#)
 - writeXml, [12](#)
- XmlOpener
 - xmlParser::XmlOpener, [61](#)
- xmlParser, [8](#)
- xmlParser.TestMain, [60](#)
- xmlParser.XmlOpener, [61](#)
- xmlParser::TestMain
 - main, [60](#)
- xmlParser::XmlOpener
 - getElementInfo, [62](#)
 - getElementInfoFromDoc, [62](#)
 - getParsedDoc, [62](#)
 - OpenFile, [63](#)
 - XmlOpener, [61](#)