

Code\_generator

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Namespace Documentation</b>	<b>5</b>
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 . . . . .	7
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 . . . . .	8
3.7.1	Detailed Description . . . . .	8
3.8	Package xmlParser . . . . .	8
3.8.1	Detailed Description . . . . .	8

<b>4</b>	<b>Class Documentation</b>	<b>9</b>
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	framework.CodeGenerator Class Reference	10
4.2.1	Detailed Description	10
4.2.2	Constructor & Destructor Documentation	10
4.2.2.1	CodeGenerator()	10
4.2.3	Member Function Documentation	10
4.2.3.1	Generate()	11
4.3	configurator.GPIO.CodeName Enum Reference	11
4.3.1	Detailed Description	11
4.3.2	Member Data Documentation	11
4.3.2.1	STR_NAME	11
4.4	framework.Common Class Reference	12
4.4.1	Detailed Description	12
4.4.2	Member Function Documentation	12
4.4.2.1	getCfgFileCPath()	12
4.4.2.2	getCfgFileHPath()	13
4.4.2.3	getCfgPath()	13
4.4.2.4	getInstallationFwkPath()	14
4.4.2.5	getProjectFwkPath()	14
4.4.2.6	setInstallationFwkPath()	14
4.4.2.7	setProjectFwkPath()	15
4.5	configurator.ConfigurationFile Class Reference	15
4.5.1	Detailed Description	15

4.5.2	Member Data Documentation . . . . .	15
4.5.2.1	STR_PROJ_CONF_FILE . . . . .	15
4.6	xmlCreator.ConfXmlWriter Class Reference . . . . .	16
4.6.1	Detailed Description . . . . .	16
4.6.2	Constructor & Destructor Documentation . . . . .	16
4.6.2.1	ConfXmlWriter() . . . . .	16
4.6.3	Member Function Documentation . . . . .	16
4.6.3.1	addPin() . . . . .	17
4.6.3.2	writeXml() . . . . .	17
4.7	common.ErrorCode Enum Reference . . . . .	17
4.7.1	Detailed Description . . . . .	18
4.7.2	Member Data Documentation . . . . .	18
4.7.2.1	EX_ERROR . . . . .	18
4.7.2.2	FILE_READ_ERROR . . . . .	18
4.7.2.3	FILE_WRITE_ERROR . . . . .	18
4.7.2.4	NO_ERROR . . . . .	18
4.7.2.5	STR_INVALID . . . . .	19
4.8	common.Features Class Reference . . . . .	19
4.8.1	Detailed Description . . . . .	19
4.8.2	Member Function Documentation . . . . .	19
4.8.2.1	debugPrint() . . . . .	19
4.8.2.2	verbosePrint() . . . . .	20
4.8.3	Member Data Documentation . . . . .	20
4.8.3.1	DEBUG . . . . .	20
4.8.3.2	DEBUG_STR . . . . .	20
4.8.3.3	SW_VERSION . . . . .	20
4.8.3.4	VERBOSE . . . . .	21
4.8.3.5	VERBOSE_STR . . . . .	21

4.8.3.6	VERSION_NAME	21
4.8.3.7	VERSION_STATUS	21
4.9	gui.GpioConfWindow Class Reference	21
4.9.1	Detailed Description	22
4.9.2	Constructor & Destructor Documentation	22
4.9.2.1	GpioConfWindow()	22
4.9.3	Member Function Documentation	22
4.9.3.1	main()	22
4.10	gui.MainGui Class Reference	23
4.10.1	Detailed Description	23
4.10.2	Member Function Documentation	23
4.10.2.1	loadProjectFile()	23
4.10.2.2	main()	24
4.10.2.3	saveUc()	24
4.10.2.4	setNewUC()	24
4.10.2.5	showAboutWindow()	24
4.10.2.6	showErrorDialog()	25
4.10.2.7	showGpioConfWindow()	25
4.11	gui.MainWindow Class Reference	25
4.11.1	Detailed Description	26
4.11.2	Constructor & Destructor Documentation	26
4.11.2.1	MainWindow()	26
4.11.3	Member Function Documentation	26
4.11.3.1	main()	26
4.11.3.2	OpenFileChooser()	26
4.11.3.3	setProjectInformation()	27
4.11.3.4	setVisible()	27
4.12	gui.Messages Class Reference	28

4.13 microcontroller.Microcontroller Class Reference . . . . .	28
4.13.1 Detailed Description . . . . .	29
4.13.2 Constructor & Destructor Documentation . . . . .	29
4.13.2.1 Microcontroller() . . . . .	29
4.13.3 Member Function Documentation . . . . .	29
4.13.3.1 getConfiguredPin() . . . . .	29
4.13.3.2 getPin() . . . . .	30
4.13.3.3 getUc_gpioNum() . . . . .	30
4.13.3.4 getUc_manufacturer() . . . . .	30
4.13.3.5 getUc_model() . . . . .	31
4.13.3.6 getUc_pinNum() . . . . .	31
4.13.3.7 getUc_portNum() . . . . .	31
4.13.3.8 isValid() . . . . .	31
4.13.3.9 processDocument() . . . . .	32
4.13.4 Member Data Documentation . . . . .	32
4.13.4.1 GpioCfgPin . . . . .	32
4.13.4.2 Ports . . . . .	32
4.14 configurator.GPIO.Mode Enum Reference . . . . .	32
4.14.1 Detailed Description . . . . .	33
4.14.2 Member Function Documentation . . . . .	33
4.14.2.1 getConfFromString() . . . . .	33
4.14.3 Member Data Documentation . . . . .	33
4.14.3.1 STR_NAME . . . . .	33
4.15 configurator.GPIO.OutLevel Enum Reference . . . . .	34
4.15.1 Detailed Description . . . . .	34
4.15.2 Member Function Documentation . . . . .	34
4.15.2.1 getConfFromString() . . . . .	34
4.15.3 Member Data Documentation . . . . .	35

4.15.3.1	STR_NAME	35
4.16	configurator.GPIO.OutType Enum Reference	35
4.16.1	Detailed Description	36
4.16.2	Member Function Documentation	36
4.16.2.1	getConfFromString()	36
4.16.3	Member Data Documentation	36
4.16.3.1	STR_NAME	36
4.17	microcontroller.Pin Class Reference	37
4.17.1	Detailed Description	38
4.17.2	Constructor & Destructor Documentation	39
4.17.2.1	Pin()	39
4.17.3	Member Function Documentation	39
4.17.3.1	getAdc()	39
4.17.3.2	getClock()	39
4.17.3.3	getFeat_adc()	40
4.17.3.4	getFeat_clock()	40
4.17.3.5	getFeat_i2c()	40
4.17.3.6	getFeat_int()	40
4.17.3.7	getFeat_reset()	41
4.17.3.8	getFeat_spi()	41
4.17.3.9	getFeat_timer()	41
4.17.3.10	getFeat_uart()	41
4.17.3.11	getFunc_gnd()	42
4.17.3.12	getFunc_gpio()	42
4.17.3.13	getFunc_vcc()	42
4.17.3.14	getI2c()	42
4.17.3.15	getInt()	43
4.17.3.16	getName()	43



4.17.3.17	getNumber()	43
4.17.3.18	getPort()	43
4.17.3.19	getReset()	44
4.17.3.20	getSpi()	44
4.17.3.21	getTimer()	44
4.17.3.22	getUart()	44
4.17.3.23	isValid()	45
4.17.3.24	setAdc()	45
4.17.3.25	setClock()	45
4.17.3.26	setFeat_adc()	45
4.17.3.27	setFeat_clock()	46
4.17.3.28	setFeat_i2c()	46
4.17.3.29	setFeat_int()	46
4.17.3.30	setFeat_reset()	47
4.17.3.31	setFeat_spi()	47
4.17.3.32	setFeat_timer()	47
4.17.3.33	setFeat_uart()	47
4.17.3.34	setFunc_gnd()	48
4.17.3.35	setFunc_gpio()	48
4.17.3.36	setFunc_vcc()	48
4.17.3.37	setI2c()	49
4.17.3.38	setInt()	49
4.17.3.39	setName()	49
4.17.3.40	setNumber()	50
4.17.3.41	setPort()	50
4.17.3.42	setReset()	50
4.17.3.43	setSpi()	50
4.17.3.44	setTimer()	51

4.17.3.45 setUart()	51
4.17.4 Member Data Documentation	51
4.17.4.1 DEF_FEATURE	51
4.17.4.2 DEF_FEATURE_AV	52
4.17.4.3 DEF_FUNCTION	52
4.17.4.4 DEF_NAME	52
4.17.4.5 DEF_NUMBER	52
4.17.4.6 DEF_PORT	52
4.17.4.7 DISABLE	52
4.17.4.8 ENABLE	52
4.18 configurator.PinConf Class Reference	53
4.18.1 Detailed Description	53
4.18.2 Constructor & Destructor Documentation	53
4.18.2.1 PinConf()	53
4.18.3 Member Function Documentation	54
4.18.3.1 getCodeName()	54
4.18.3.2 getMode()	54
4.18.3.3 getOutLevel()	54
4.18.3.4 getOutType()	55
4.18.3.5 getPinName()	55
4.18.3.6 getPort()	55
4.18.3.7 getPull()	55
4.18.3.8 getSpeed()	56
4.18.3.9 isAv_Adc()	56
4.18.3.10 isAv_altFunc()	56
4.18.3.11 isValid()	56
4.18.3.12 setCodeName()	56
4.18.3.13 setMode()	57

4.18.3.14	setOutLevel()	57
4.18.3.15	setOutType()	57
4.18.3.16	setPull()	58
4.18.3.17	setSpeed()	58
4.18.4	Member Data Documentation	58
4.18.4.1	DF_CODE_NAME	58
4.18.4.2	DF_MODE	58
4.18.4.3	DF_OUT_LEVEL	59
4.18.4.4	DF_OUTTYPE	59
4.18.4.5	DF_PULL	59
4.18.4.6	DF_SPEED	59
4.19	projectConfiguration.ProjectSettings Class Reference	59
4.19.1	Detailed Description	60
4.19.2	Constructor & Destructor Documentation	60
4.19.2.1	ProjectSettings()	60
4.19.3	Member Function Documentation	60
4.19.3.1	getConfFile()	60
4.19.3.2	getFrameworkPath()	61
4.19.3.3	getProjectName()	61
4.19.3.4	getUcFile()	61
4.19.3.5	openProjectFile()	61
4.19.3.6	processDocument()	62
4.20	configurator.GPIO.Pull Enum Reference	62
4.20.1	Detailed Description	63
4.20.2	Member Function Documentation	63
4.20.2.1	getConfFromString()	63
4.20.3	Member Data Documentation	63
4.20.3.1	STR_NAME	63

4.21 configurator.GPIO.Speed Enum Reference . . . . .	64
4.21.1 Detailed Description . . . . .	64
4.21.2 Member Function Documentation . . . . .	64
4.21.2.1 getConfFromString() . . . . .	64
4.21.3 Member Data Documentation . . . . .	65
4.21.3.1 STR_NAME . . . . .	65
4.22 xmlParser.TestMain Class Reference . . . . .	65
4.22.1 Detailed Description . . . . .	65
4.22.2 Member Function Documentation . . . . .	65
4.22.2.1 main() . . . . .	65
4.23 xmlParser.XmlOpener Class Reference . . . . .	66
4.23.1 Detailed Description . . . . .	66
4.23.2 Constructor & Destructor Documentation . . . . .	66
4.23.2.1 XmlOpener() . . . . .	66
4.23.3 Member Function Documentation . . . . .	67
4.23.3.1 getElementInfo() . . . . .	67
4.23.3.2 getElementInfoFromDoc() . . . . .	67
4.23.3.3 getParsedDoc() . . . . .	67
4.23.3.4 OpenFile() . . . . .	68

# Chapter 1

## Namespace Index

### 1.1 Packages

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

<a href="#">common</a>	5
<a href="#">configurator</a>	5
<a href="#">framework</a>	6
<a href="#">gui</a>	6
<a href="#">microcontroller</a>	7
<a href="#">projectConfiguration</a>	7
<a href="#">xmlCreator</a>	8
<a href="#">xmlParser</a>	8



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">gui&gt;AboutWindow</a>	9
<a href="#">framework.CodeGenerator</a>	10
<a href="#">configurator.GPIO.CodeName</a>	11
<a href="#">framework.Common</a>	12
<a href="#">configurator.ConfigurationFile</a>	15
<a href="#">xmlCreator.ConfXmlWriter</a>	16
<a href="#">common.ErrorCode</a>	17
<a href="#">common.Features</a>	19
<a href="#">gui.GpioConfWindow</a>	21
<a href="#">gui.MainGui</a>	23
<a href="#">gui.MainWindow</a>	25
<a href="#">gui.Messages</a>	28
<a href="#">microcontroller.Microcontroller</a>	28
<a href="#">configurator.GPIO.Mode</a>	32
<a href="#">configurator.GPIO.OutLevel</a>	34
<a href="#">configurator.GPIO.OutType</a>	35
<a href="#">microcontroller.Pin</a>	37
<a href="#">configurator.PinConf</a>	53
<a href="#">projectConfiguration.ProjectSettings</a>	59
<a href="#">configurator.GPIO.Pull</a>	62
<a href="#">configurator.GPIO.Speed</a>	64
<a href="#">xmlParser.TestMain</a>	65
<a href="#">xmlParser.XmlOpener</a>	66





## 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

### Classes

- class [CodeGenerator](#)
- class [Common](#)
- class **GpioGenerator**

### 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 framework.CodeGenerator Class Reference

**Public Member Functions**

- [CodeGenerator](#) ([Microcontroller](#) uC, [ProjectSettings](#) projectSettings)
- [ErrorCode Generate](#) ()

### 4.2.1 Detailed Description

**Author**

ovd

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 CodeGenerator()

```
framework.CodeGenerator.CodeGenerator (
    Microcontroller uC,
    ProjectSettings projectSettings )
```

**Constructor****Parameters**

<i>uC</i>	Project's microcontroller
<i>projectSettings</i>	Project's settings

### 4.2.3 Member Function Documentation

#### 4.2.3.1 Generate()

```
ErrorCode framework.CodeGenerator.Generate ( )
```

Generate project's configuration files

##### Returns

Error code

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

- src/framework/CodeGenerator.java

### 4.3 configurator.GPIO.CodeName Enum Reference

#### Static Public Attributes

- static final String `STR_NAME` = "codeName"

#### 4.3.1 Detailed Description

##### Author

Miguel Diaz

##### Version

0.1

#### 4.3.2 Member Data Documentation

##### 4.3.2.1 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

## 4.4 framework.Common Class Reference

### Static Public Member Functions

- static String [getInstallationFwkPath](#) ()
- static void [setInstallationFwkPath](#) (String installationFwkPath)
- static String [getProjectFwkPath](#) ()
- static void [setProjectFwkPath](#) (String projectFwkPath)
- static String [getCfgPath](#) (String fwkPath, String cfgModule)
- static String [getCfgFileCPath](#) (String fwkPath, String cfgModule)
- static String [getCfgFileHPath](#) (String fwkPath, String cfgModule)

### Static Public Attributes

- static final String **NL** = "\r\n"
- static final String **STR\_MODULE\_GPIO** = "gpio"
- static final String **STR\_DEFINITION** = "#define "

#### 4.4.1 Detailed Description

Framework common fields and methods

##### Author

Miguel Diaz

##### Version

0.1

#### 4.4.2 Member Function Documentation

##### 4.4.2.1 [getCfgFileCPath\(\)](#)

```
static String framework.Common.getCfgFileCPath (  
    String fwkPath,  
    String cfgModule ) [static]
```

Get GPIO configuration file path



## Parameters

<i>fwkPath</i>	Framework folder path
<i>cfgModule</i>	Configuration module name

## Returns

GPIO configuration file path

## 4.4.2.2 getCfgFileHPath()

```
static String framework.Common.getCfgFileHPath (  
    String fwkPath,  
    String cfgModule ) [static]
```

Get GPIO configuration header file path

## Parameters

<i>fwkPath</i>	Framework folder path
<i>cfgModule</i>	Configuration module name

## Returns

GPIO configuration header file path

## 4.4.2.3 getCfgPath()

```
static String framework.Common.getCfgPath (  
    String fwkPath,  
    String cfgModule ) [static]
```

Get configuration module files folder path

## Parameters

<i>fwkPath</i>	Framework folder path
<i>cfgModule</i>	Configuration module name

**Returns**

Configuration files folder path

**4.4.2.4 getInstallationFwkPath()**

```
static String framework.Common.getInstallationFwkPath ( ) [static]
```

Get installation framework path

**Returns**

installation framework path

**4.4.2.5 getProjectFwkPath()**

```
static String framework.Common.getProjectFwkPath ( ) [static]
```

Get project's framework path

**Returns**

project's framework path

**4.4.2.6 setInstallationFwkPath()**

```
static void framework.Common.setInstallationFwkPath (
    String installationFwkPath ) [static]
```

Set installation framework path

**Parameters**

<i>installationFwkPath</i>	installation framework path
----------------------------	-----------------------------

#### 4.4.2.7 setProjectFwkPath()

```
static void framework.Common.setProjectFwkPath (  
    String projectFwkPath )    [static]
```

Set project's framework path

##### Parameters

<i>projectFwkPath</i>	project's framework path
-----------------------	--------------------------

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

- src/framework/Common.java

## 4.5 configurator.ConfigurationFile Class Reference

### Static Public Attributes

- static final String [STR\\_PROJ\\_CONF\\_FILE](#) = "cgs"

#### 4.5.1 Detailed Description

Configuration files properties

##### Author

Miguel Diaz

##### Version

0.1

#### 4.5.2 Member Data Documentation

##### 4.5.2.1 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

## 4.6 xmlCreator.ConfXmlWriter Class Reference

### Public Member Functions

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

### 4.6.1 Detailed Description

Write a XML file

#### Author

Miguel Diaz

#### Version

0.1

### 4.6.2 Constructor & Destructor Documentation

#### 4.6.2.1 ConfXmlWriter()

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

#### Constructor

##### Parameters

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

### 4.6.3 Member Function Documentation

#### 4.6.3.1 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

#### 4.6.3.2 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

## 4.7 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"

### 4.7.1 Detailed Description

Error codes enum

#### Author

Miguel Diaz

#### Version

0.1

### 4.7.2 Member Data Documentation

#### 4.7.2.1 EX\_ERROR

`common.ErrorCode.EX_ERROR`

Error during execution

#### 4.7.2.2 FILE\_READ\_ERROR

`common.ErrorCode.FILE_READ_ERROR`

File reading error

#### 4.7.2.3 FILE\_WRITE\_ERROR

`common.ErrorCode.FILE_WRITE_ERROR`

File writing error

#### 4.7.2.4 NO\_ERROR

`common.ErrorCode.NO_ERROR`

No error message

#### 4.7.2.5 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

## 4.8 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](#) = "Bespin"

### 4.8.1 Detailed Description

Class that includes all project features

Author

Miguel Diaz

Version

0.1

### 4.8.2 Member Function Documentation

#### 4.8.2.1 debugPrint()

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

Print Debug message to console

**Parameters**

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

**4.8.2.2 verbosePrint()**

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

Print Verbose message to console

**Parameters**

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

**4.8.3 Member Data Documentation****4.8.3.1 DEBUG**

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

Enables debug functions

**4.8.3.2 DEBUG\_STR**

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

Debug messages indicator on system console

**4.8.3.3 SW\_VERSION**

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

Complete Software version



#### 4.8.3.4 VERBOSE

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

Enables console messages

#### 4.8.3.5 VERBOSE\_STR

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

Verbose messages indicator on system console

#### 4.8.3.6 VERSION\_NAME

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

Code name of the software version

#### 4.8.3.7 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`

## 4.9 gui.GpioConfWindow Class Reference

### Public Member Functions

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

### Static Public Member Functions

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

### 4.9.1 Detailed Description

Window for configuring GPIO pins

#### Author

Miguel Diaz

#### Version

0.1

### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 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
--------------	---

### 4.9.3 Member Function Documentation

#### 4.9.3.1 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

## 4.10 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 void [generateCode](#) ()

### Static Public Attributes

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

### 4.10.1 Detailed Description

Main GUI state machine

Author

Miguel Diaz

Version

0.1

### 4.10.2 Member Function Documentation

#### 4.10.2.1 loadProjectFile()

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

Load the project settings file

**Parameters**

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

**Returns**

Error status

**4.10.2.2 main()**

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

**Parameters**

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

**4.10.2.3 saveUc()**

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

Save the microcontroller's configuration to disk

**4.10.2.4 setNewUC()**

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

Set the project's microcontroller configuration

**Parameters**

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

**4.10.2.5 showAboutWindow()**

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

Show about information window

#### 4.10.2.6 showErrorDialog()

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

Show an error dialog

##### Parameters

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

#### 4.10.2.7 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

## 4.11 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**

### 4.11.1 Detailed Description

Main application window

Author

Miguel Diaz

Version

0.1

### 4.11.2 Constructor & Destructor Documentation

#### 4.11.2.1 MainWindow()

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

Create the application.

### 4.11.3 Member Function Documentation

#### 4.11.3.1 main()

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

Open main window

Parameters

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

#### 4.11.3.2 OpenFileChooser()

```
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

#### 4.11.3.3 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

#### 4.11.3.4 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`

## 4.12 `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`

## 4.13 `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



### 4.13.1 Detailed Description

[Microcontroller](#) related methods

Author

Miguel Diaz

Version

0.1

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 Microcontroller()

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

Constructor

Parameters

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

### 4.13.3 Member Function Documentation

#### 4.13.3.1 getConfiguredPin()

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

Get the configuration of a pin

Parameters

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

**Returns**

[Pin](#) configuration

**4.13.3.2 getPin()**

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

Get a pin's characteristics

**Parameters**

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

**Returns**

[Pin's](#) characteristics

**4.13.3.3 getUc\_gpioNum()**

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

Get the number of GPIOs in the microcontroller

**Returns**

Number of GPIOs

**4.13.3.4 getUc\_manufacturer()**

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

Get the microcontroller's manufacturer

**Returns**

[Microcontroller's](#) manufacturer

#### 4.13.3.5 `getUc_model()`

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

Get the microcontroller's model

##### Returns

Microcontroller's model

#### 4.13.3.6 `getUc_pinNum()`

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

Get the microcontroller's pins number

##### Returns

Number of pins

#### 4.13.3.7 `getUc_portNum()`

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

Get the number of ports in the microcontroller

##### Returns

Number of ports

#### 4.13.3.8 `isValid()`

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

Check if the microcontroller configuration is valid

##### Returns

true if valid

#### 4.13.3.9 processDocument()

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

Process the document obtained from XML file

##### Returns

Error status

### 4.13.4 Member Data Documentation

#### 4.13.4.1 GpioCfgPin

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

Configured pins list

#### 4.13.4.2 Ports

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

Ports name list

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

- src/microcontroller/Microcontroller.java

## 4.14 configurator.GPIO.Mode Enum Reference

### Static Public Member Functions

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

### Public Attributes

- **MODE\_INPUT**
- **MODE\_OUTPUT**
- **MODE\_ALTERNATE\_FUNCTION**
- **MODE\_ANALOG**

## Static Public Attributes

- static final String [STR\\_NAME](#) = "Mode"

### 4.14.1 Detailed Description

GPIO modes

Author

Miguel Diaz

Version

0.1

### 4.14.2 Member Function Documentation

#### 4.14.2.1 `getConfFromString()`

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

Get the corresponding mode from its name as String

Parameters

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

Returns

[Mode](#)

### 4.14.3 Member Data Documentation

#### 4.14.3.1 `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`

## 4.15 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"

### 4.15.1 Detailed Description

Pin's output/input level

Author

Miguel Diaz

Version

0.1

### 4.15.2 Member Function Documentation

#### 4.15.2.1 `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

### 4.15.3 Member Data Documentation

#### 4.15.3.1 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

## 4.16 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"

### 4.16.1 Detailed Description

Pin's output type

Author

Miguel Diaz

Version

0.1

### 4.16.2 Member Function Documentation

#### 4.16.2.1 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

### 4.16.3 Member Data Documentation

#### 4.16.3.1 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



## 4.17 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

### 4.17.1 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

## 4.17.2 Constructor & Destructor Documentation

### 4.17.2.1 Pin()

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

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

## 4.17.3 Member Function Documentation

### 4.17.3.1 getAdc()

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

Get the pin's ADC name

Returns

Pin's ADC

### 4.17.3.2 getClock()

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

Get the pin's clock name

Returns

Pin's clock

#### 4.17.3.3 `getFeat_adc()`

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

See if the pin has an ADC

##### Returns

Feature availability

#### 4.17.3.4 `getFeat_clock()`

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

See if the pin supports a clock

##### Returns

Feature availability

#### 4.17.3.5 `getFeat_i2c()`

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

See if the pin has I2C

##### Returns

Feature availability

#### 4.17.3.6 `getFeat_int()`

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

See if the pin has an interruption

##### Returns

Feature availability

#### 4.17.3.7 getFeat\_reset()

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

See if the pin has a reset feature

##### Returns

Feature availability

#### 4.17.3.8 getFeat\_spi()

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

See if the pin has SPI

##### Returns

Feature availability

#### 4.17.3.9 getFeat\_timer()

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

See if the pin supports a timer

##### Returns

Feature availability

#### 4.17.3.10 getFeat\_uart()

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

See if the pin has a UART

##### Returns

Feature availability

#### 4.17.3.11 `getFunc_gnd()`

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

See if the pin is GND

##### Returns

Function availability

#### 4.17.3.12 `getFunc_gpio()`

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

See if the pin is GPIO

##### Returns

Function availability

#### 4.17.3.13 `getFunc_vcc()`

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

See if the pin is Vcc

##### Returns

Function availability

#### 4.17.3.14 `getI2c()`

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

Get the pin's I2C name

##### Returns

[Pin's I2C](#)

#### 4.17.3.15 getInt()

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

Get the pin's interruption name

Returns

Pin's interruption

#### 4.17.3.16 getName()

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

Get the pin's name

Returns

Pin's name

#### 4.17.3.17 getNumber()

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

Get the pin's number

Returns

Pin's number

#### 4.17.3.18 getPort()

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

Get the pin's port

Returns

Pin's port

#### 4.17.3.19 `getReset()`

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

Get the pin's reset name

##### Returns

[Pin's reset](#)

#### 4.17.3.20 `getSpi()`

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

Get the pin's SPI name

##### Returns

[Pin's SPI](#)

#### 4.17.3.21 `getTimer()`

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

Get the pin's timer name

##### Returns

[Pin's timer](#)

#### 4.17.3.22 `getUart()`

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

Get the pin's UART name

##### Returns

[Pin's UART](#)



#### 4.17.3.23 isValid()

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

Check if the pin is correctly initialized

##### Returns

True if the pin is correctly initialized

#### 4.17.3.24 setAdc()

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

Set the pin's ADC

##### Parameters

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

#### 4.17.3.25 setClock()

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

Set the pin's clock

##### Parameters

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

#### 4.17.3.26 setFeat\_adc()

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

Set the pin's ADC feature

**Parameters**

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

**4.17.3.27 setFeat\_clock()**

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

Set the pin's Clock feature

**Parameters**

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

**4.17.3.28 setFeat\_i2c()**

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

Set the pin's I2C feature

**Parameters**

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

**4.17.3.29 setFeat\_int()**

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

Set the pin's interruption feature

**Parameters**

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

#### 4.17.3.30 setFeat\_reset()

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

Set the pin's reset feature

##### Parameters

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

#### 4.17.3.31 setFeat\_spi()

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

Set the pin's SPI feature

##### Parameters

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

#### 4.17.3.32 setFeat\_timer()

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

Set the pin's timer feature

##### Parameters

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

#### 4.17.3.33 setFeat\_uart()

```
void microcontroller.Pin.setFeat_uart (
```

```
boolean featState )
```

Set the pin's UART feature

**Parameters**

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

#### 4.17.3.34 setFunc\_gnd()

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

Set the pin to GND status

**Parameters**

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

#### 4.17.3.35 setFunc\_gpio()

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

Set the pin to GPIO status

**Parameters**

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

#### 4.17.3.36 setFunc\_vcc()

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

Set the pin to Vcc status

## Parameters

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

## 4.17.3.37 setI2c()

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

Set the pin's I2C

## Parameters

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

## 4.17.3.38 setInt()

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

Set the pin's interruption

## Parameters

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

## 4.17.3.39 setName()

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

Set the pin's name

## Parameters

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

**4.17.3.40 setNumber()**

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

Set the pin's number

**Parameters**

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

**4.17.3.41 setPort()**

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

Set the pin's port

**Parameters**

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

**4.17.3.42 setReset()**

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

Set the pin's reset

**Parameters**

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

**4.17.3.43 setSpi()**

```
void microcontroller.Pin.setSpi (
```

```
String feature )
```

Set the pin's SPI

#### Parameters

<i>feature</i>	<a href="#">Pin's SPI</a>
----------------	---------------------------

#### 4.17.3.44 setTimer()

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

Set the pin's timer

#### Parameters

<i>feature</i>	<a href="#">Pin's timer</a>
----------------	-----------------------------

#### 4.17.3.45 setUart()

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

Set the pin's UART

#### Parameters

<i>feature</i>	<a href="#">Pin's UART</a>
----------------	----------------------------

### 4.17.4 Member Data Documentation

#### 4.17.4.1 DEF\_FEATURE

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

Default value for pin's feature as not available

#### 4.17.4.2 DEF\_FEATURE\_AV

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

Default value for pin's feature availability as not available

#### 4.17.4.3 DEF\_FUNCTION

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

Default value for pin's function as not enabled

#### 4.17.4.4 DEF\_NAME

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

Default value for pin's name

#### 4.17.4.5 DEF\_NUMBER

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

Default value for pin's number

#### 4.17.4.6 DEF\_PORT

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

Default value for pin's port

#### 4.17.4.7 DISABLE

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

Disable value for features and functions

#### 4.17.4.8 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



## 4.18 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](#) = ""

### 4.18.1 Detailed Description

GPIO pin configuration

Author

Miguel Diaz

Version

0.1

### 4.18.2 Constructor & Destructor Documentation

#### 4.18.2.1 PinConf()

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

Constructor

**Parameters**

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

### 4.18.3 Member Function Documentation

#### 4.18.3.1 getCodeName()

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

Get the pin's user selected name

**Returns**

pin's name

#### 4.18.3.2 getMode()

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

Get the pin's mode configuration

**Returns**

Mode

#### 4.18.3.3 getOutLevel()

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

Get the pin's output level

**Returns**

Pin's output level

#### 4.18.3.4 getOutType()

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

Get the pin's output configuration

##### Returns

Output configuration

#### 4.18.3.5 getPinName()

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

Get the pin's number

##### Returns

Pin's number

#### 4.18.3.6 getPort()

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

Get the pin's port

##### Returns

Port

#### 4.18.3.7 getPull()

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

Get the pin's pull resistor configuration

##### Returns

Pull Resistor configuration

#### 4.18.3.8 `getSpeed()`

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

Get the pin's speed

##### Returns

Speed

#### 4.18.3.9 `isAv_Adc()`

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

Check availability of ADC

##### Returns

True if ADC is available

#### 4.18.3.10 `isAv_altFunc()`

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

Check the availability of alternate function

##### Returns

True if alternate function is available

#### 4.18.3.11 `isValid()`

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

Check if the GPIO pin is valid

##### Returns

True if valid

#### 4.18.3.12 `setCodeName()`

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

Set the pin's user selected name

## Parameters

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

## 4.18.3.13 setMode()

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

Set the pin's mode configuration

## Parameters

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

## 4.18.3.14 setOutLevel()

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

Set the pin's output level

## Parameters

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

## 4.18.3.15 setOutType()

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

Set the pin's output configuration

## Parameters

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

#### 4.18.3.16 setPull()

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

Set the pull resistor configuration

##### Parameters

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

#### 4.18.3.17 setSpeed()

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

Set the pin's speed

##### Parameters

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

### 4.18.4 Member Data Documentation

#### 4.18.4.1 DF\_CODE\_NAME

```
final String configurator.PinConf.DF_CODE_NAME = "" [static]
```

Default pin's code name

#### 4.18.4.2 DF\_MODE

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

Default Pin mode

#### 4.18.4.3 DF\_OUT\_LEVEL

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

Default pin's output level

#### 4.18.4.4 DF\_OUTTYPE

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

Default pin's output type

#### 4.18.4.5 DF\_PULL

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

Default pin's pull resistor

#### 4.18.4.6 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

## 4.19 projectConfiguration.ProjectSettings Class Reference

### Public Member Functions

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

### 4.19.1 Detailed Description

Project settings class

Author

Miguel Diaz

Version

0.2

### 4.19.2 Constructor & Destructor Documentation

#### 4.19.2.1 ProjectSettings()

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

Constructor

### 4.19.3 Member Function Documentation

#### 4.19.3.1 getConfFile()

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

Get the project configuration file

Returns

Project configuration file



#### 4.19.3.2 getFrameworkPath()

```
String projectConfiguration.ProjectSettings.getFrameworkPath ( )
```

Get the framework folder

##### Returns

framework folder

#### 4.19.3.3 getProjectName()

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

Get the project's name

##### Returns

Project's name

#### 4.19.3.4 getUcFile()

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

Get the project microcontroller file

##### Returns

Project microcontroller file

#### 4.19.3.5 openProjectFile()

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

Open the project settings file

**Parameters**

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

**Returns**

Error Status

**4.19.3.6 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

## 4.20 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"

### 4.20.1 Detailed Description

Pin's pull resistor

Author

Miguel Diaz

Version

0.1

### 4.20.2 Member Function Documentation

#### 4.20.2.1 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

### 4.20.3 Member Data Documentation

#### 4.20.3.1 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

## 4.21 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"

### 4.21.1 Detailed Description

Pin's speed

#### Author

Miguel Diaz

#### Version

0.1

### 4.21.2 Member Function Documentation

#### 4.21.2.1 [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](#)

### 4.21.3 Member Data Documentation

#### 4.21.3.1 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

## 4.22 xmlParser.TestMain Class Reference

### Static Public Member Functions

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

#### 4.22.1 Detailed Description

Dummy main class for testing the other classes

Author

Miguel Diaz

### 4.22.2 Member Function Documentation

#### 4.22.2.1 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`

## 4.23 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)

#### 4.23.1 Detailed Description

Open and process XML files

#### Author

H112943

#### Version

0.1

#### 4.23.2 Constructor & Destructor Documentation

##### 4.23.2.1 XmlOpener()

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

#### Constructor

### 4.23.3 Member Function Documentation

#### 4.23.3.1 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

#### 4.23.3.2 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

#### 4.23.3.3 getParsedDoc()

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

Get the parsed document AFTER opening the file

**Returns**

Parsed document

**4.23.3.4   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, 16  
  
CodeGenerator  
    framework::CodeGenerator, 10  
common, 5  
common.ErrorCode, 17  
common.Features, 19  
common::ErrorCode  
    EX\_ERROR, 18  
    FILE\_READ\_ERROR, 18  
    FILE\_WRITE\_ERROR, 18  
    NO\_ERROR, 18  
    STR\_INVALID, 18  
common::Features  
    DEBUG\_STR, 20  
    DEBUG, 20  
    debugPrint, 19  
    SW\_VERSION, 20  
    VERBOSE\_STR, 21  
    VERBOSE, 20  
    VERSION\_NAME, 21  
    VERSION\_STATUS, 21  
    verbosePrint, 20  
ConfXmlWriter  
    xmlCreator::ConfXmlWriter, 16  
configurator, 5  
configurator.ConfigurationFile, 15  
configurator.GPIO.CodeName, 11  
configurator.GPIO.Mode, 32  
configurator.GPIO.OutLevel, 34  
configurator.GPIO.OutType, 35  
configurator.GPIO.Pull, 62  
configurator.GPIO.Speed, 64  
configurator.PinConf, 53  
configurator::ConfigurationFile  
    STR\_PROJ\_CONF\_FILE, 15  
configurator::GPIO::CodeName  
    STR\_NAME, 11  
configurator::GPIO::Mode  
    getConfFromString, 33  
    STR\_NAME, 33  
configurator::GPIO::OutLevel  
    getConfFromString, 34  
    STR\_NAME, 35  
configurator::GPIO::OutType  
    getConfFromString, 36  
    STR\_NAME, 36  
configurator::GPIO::Pull  
    getConfFromString, 63  
    STR\_NAME, 63  
configurator::GPIO::Speed  
    getConfFromString, 64  
    STR\_NAME, 65  
configurator::PinConf  
    DF\_CODE\_NAME, 58  
    DF\_MODE, 58  
    DF\_OUT\_LEVEL, 58  
    DF\_OUTTYPE, 59  
    DF\_PULL, 59  
    DF\_SPEED, 59  
    getCodeName, 54  
    getMode, 54  
    getOutLevel, 54  
    getOutType, 54  
    getPinName, 55  
    getPort, 55  
    getPull, 55  
    getSpeed, 55  
    isAv\_Adc, 56  
    isAv\_altFunc, 56  
    isValid, 56  
    PinConf, 53  
    setCodeName, 56  
    setMode, 57  
    setOutLevel, 57  
    setOutType, 57  
    setPull, 58  
    setSpeed, 58  
  
DEBUG\_STR  
    common::Features, 20  
DEBUG  
    common::Features, 20  
DEF\_FEATURE\_AV  
    microcontroller::Pin, 51  
DEF\_FEATURE  
    microcontroller::Pin, 51  
DEF\_FUNCTION  
    microcontroller::Pin, 52

- DEF\_NAME
  - microcontroller::Pin, 52
- DEF\_NUMBER
  - microcontroller::Pin, 52
- DEF\_PORT
  - microcontroller::Pin, 52
- DF\_CODE\_NAME
  - configurator::PinConf, 58
- DF\_MODE
  - configurator::PinConf, 58
- DF\_OUT\_LEVEL
  - configurator::PinConf, 58
- DF\_OUTTYPE
  - configurator::PinConf, 59
- DF\_PULL
  - configurator::PinConf, 59
- DF\_SPEED
  - configurator::PinConf, 59
- DISABLE
  - microcontroller::Pin, 52
- debugPrint
  - common::Features, 19
- ENABLE
  - microcontroller::Pin, 52
- EX\_ERROR
  - common::ErrorCode, 18
- FILE\_READ\_ERROR
  - common::ErrorCode, 18
- FILE\_WRITE\_ERROR
  - common::ErrorCode, 18
- framework, 6
- framework.CodeGenerator, 10
- framework.Common, 12
- framework::CodeGenerator
  - CodeGenerator, 10
  - Generate, 10
- framework::Common
  - getCfgFileCPath, 12
  - getCfgFileHPath, 13
  - getCfgPath, 13
  - getInstallationFwkPath, 14
  - getProjectFwkPath, 14
  - setInstallationFwkPath, 14
  - setProjectFwkPath, 14
- Generate
  - framework::CodeGenerator, 10
- getAdc
  - microcontroller::Pin, 39
- getCfgFileCPath
  - framework::Common, 12
- getCfgFileHPath
  - framework::Common, 13
- getCfgPath
  - framework::Common, 13
- getClock
  - microcontroller::Pin, 39
- getCodeName
  - configurator::PinConf, 54
- getConfFile
  - projectConfiguration::ProjectSettings, 60
- getConfFromString
  - configurator::GPIO::Mode, 33
  - configurator::GPIO::OutLevel, 34
  - configurator::GPIO::OutType, 36
  - configurator::GPIO::Pull, 63
  - configurator::GPIO::Speed, 64
- getConfiguredPin
  - microcontroller::Microcontroller, 29
- getElementInfo
  - xmlParser::XmlOpener, 67
- getElementInfoFromDoc
  - xmlParser::XmlOpener, 67
- getFeat\_adc
  - microcontroller::Pin, 39
- getFeat\_clock
  - microcontroller::Pin, 40
- getFeat\_i2c
  - microcontroller::Pin, 40
- getFeat\_int
  - microcontroller::Pin, 40
- getFeat\_reset
  - microcontroller::Pin, 40
- getFeat\_spi
  - microcontroller::Pin, 41
- getFeat\_timer
  - microcontroller::Pin, 41
- getFeat\_uart
  - microcontroller::Pin, 41
- getFrameworkPath
  - projectConfiguration::ProjectSettings, 60
- getFunc\_gnd
  - microcontroller::Pin, 41
- getFunc\_gpio
  - microcontroller::Pin, 42
- getFunc\_vcc
  - microcontroller::Pin, 42
- getI2c
  - microcontroller::Pin, 42
- getInstallationFwkPath
  - framework::Common, 14
- getInt
  - microcontroller::Pin, 42
- getMode
  - configurator::PinConf, 54
- getName
  - microcontroller::Pin, 43

- getNumber
  - microcontroller::Pin, [43](#)
- getOutLevel
  - configurator::PinConf, [54](#)
- getOutType
  - configurator::PinConf, [54](#)
- getParsedDoc
  - xmlParser::XmlOpener, [67](#)
- getPin
  - microcontroller::Microcontroller, [30](#)
- getPinName
  - configurator::PinConf, [55](#)
- getPort
  - configurator::PinConf, [55](#)
  - microcontroller::Pin, [43](#)
- getProjectFwkPath
  - framework::Common, [14](#)
- getProjectName
  - projectConfiguration::ProjectSettings, [61](#)
- getPull
  - configurator::PinConf, [55](#)
- getReset
  - microcontroller::Pin, [43](#)
- getSpeed
  - configurator::PinConf, [55](#)
- getSpi
  - microcontroller::Pin, [44](#)
- getTimer
  - microcontroller::Pin, [44](#)
- getUart
  - microcontroller::Pin, [44](#)
- getUc\_gpioNum
  - microcontroller::Microcontroller, [30](#)
- getUc\_manufacturer
  - microcontroller::Microcontroller, [30](#)
- getUc\_model
  - microcontroller::Microcontroller, [30](#)
- getUc\_pinNum
  - microcontroller::Microcontroller, [31](#)
- getUc\_portNum
  - microcontroller::Microcontroller, [31](#)
- getUcFile
  - projectConfiguration::ProjectSettings, [61](#)
- GpioCfgPin
  - microcontroller::Microcontroller, [32](#)
- GpioConfWindow
  - gui::GpioConfWindow, [22](#)
- gui, [6](#)
  - gui.AboutWindow, [9](#)
  - gui.GpioConfWindow, [21](#)
  - gui.MainGui, [23](#)
  - gui.MainWindow, [25](#)
  - gui.Messages, [28](#)
  - gui::AboutWindow
  - AboutWindow, [9](#)
  - main, [9](#)
- gui::GpioConfWindow
  - GpioConfWindow, [22](#)
  - main, [22](#)
- gui::MainGui
  - loadProjectFile, [23](#)
  - main, [24](#)
  - saveUc, [24](#)
  - setNewUC, [24](#)
  - showAboutWindow, [24](#)
  - showErrorDialog, [25](#)
  - showGpioConfWindow, [25](#)
- gui::MainWindow
  - main, [26](#)
  - MainWindow, [26](#)
  - OpenFileChooser, [26](#)
  - setProjectInformation, [27](#)
  - setVisible, [27](#)
- isAv\_Adc
  - configurator::PinConf, [56](#)
- isAv\_altFunc
  - configurator::PinConf, [56](#)
- isValid
  - configurator::PinConf, [56](#)
  - microcontroller::Microcontroller, [31](#)
  - microcontroller::Pin, [44](#)
- loadProjectFile
  - gui::MainGui, [23](#)
- main
  - gui::AboutWindow, [9](#)
  - gui::GpioConfWindow, [22](#)
  - gui::MainGui, [24](#)
  - gui::MainWindow, [26](#)
  - xmlParser::TestMain, [65](#)
- MainWindow
  - gui::MainWindow, [26](#)
- Microcontroller
  - microcontroller::Microcontroller, [29](#)
- microcontroller, [7](#)
  - microcontroller.Microcontroller, [28](#)
  - microcontroller.Pin, [37](#)
  - microcontroller::Microcontroller
    - getConfiguredPin, [29](#)
    - getPin, [30](#)
    - getUc\_gpioNum, [30](#)
    - getUc\_manufacturer, [30](#)
    - getUc\_model, [30](#)
    - getUc\_pinNum, [31](#)
    - getUc\_portNum, [31](#)
    - GpioCfgPin, [32](#)
    - isValid, [31](#)

- Microcontroller, [29](#)
- Ports, [32](#)
- processDocument, [31](#)
- microcontroller::Pin
  - DEF\_FEATURE\_AV, [51](#)
  - DEF\_FEATURE, [51](#)
  - DEF\_FUNCTION, [52](#)
  - DEF\_NAME, [52](#)
  - DEF\_NUMBER, [52](#)
  - DEF\_PORT, [52](#)
  - DISABLE, [52](#)
  - ENABLE, [52](#)
  - getAdc, [39](#)
  - getClock, [39](#)
  - getFeat\_adc, [39](#)
  - getFeat\_clock, [40](#)
  - getFeat\_i2c, [40](#)
  - getFeat\_int, [40](#)
  - getFeat\_reset, [40](#)
  - getFeat\_spi, [41](#)
  - getFeat\_timer, [41](#)
  - getFeat\_uart, [41](#)
  - getFunc\_gnd, [41](#)
  - getFunc\_gpio, [42](#)
  - getFunc\_vcc, [42](#)
  - getI2c, [42](#)
  - getInt, [42](#)
  - getName, [43](#)
  - getNumber, [43](#)
  - getPort, [43](#)
  - getReset, [43](#)
  - getSpi, [44](#)
  - getTimer, [44](#)
  - getUart, [44](#)
  - isValid, [44](#)
  - Pin, [39](#)
  - setAdc, [45](#)
  - setClock, [45](#)
  - setFeat\_adc, [45](#)
  - setFeat\_clock, [46](#)
  - setFeat\_i2c, [46](#)
  - setFeat\_int, [46](#)
  - setFeat\_reset, [47](#)
  - setFeat\_spi, [47](#)
  - setFeat\_timer, [47](#)
  - setFeat\_uart, [47](#)
  - setFunc\_gnd, [48](#)
  - setFunc\_gpio, [48](#)
  - setFunc\_vcc, [48](#)
  - setI2c, [49](#)
  - setInt, [49](#)
  - setName, [49](#)
  - setNumber, [50](#)
  - setPort, [50](#)
  - setReset, [50](#)
  - setSpi, [50](#)
  - setTimer, [51](#)
  - setUart, [51](#)
- NO\_ERROR
  - common::ErrorCode, [18](#)
- OpenFile
  - xmlParser::XmlOpener, [68](#)
- OpenFileChooser
  - gui::MainWindow, [26](#)
- openProjectFile
  - projectConfiguration::ProjectSettings, [61](#)
- Pin
  - microcontroller::Pin, [39](#)
- PinConf
  - configurator::PinConf, [53](#)
- Ports
  - microcontroller::Microcontroller, [32](#)
- processDocument
  - microcontroller::Microcontroller, [31](#)
  - projectConfiguration::ProjectSettings, [62](#)
- projectConfiguration, [7](#)
- projectConfiguration.ProjectSettings, [59](#)
- projectConfiguration::ProjectSettings
  - getConfFile, [60](#)
  - getFrameworkPath, [60](#)
  - getProjectName, [61](#)
  - getUcFile, [61](#)
  - openProjectFile, [61](#)
  - processDocument, [62](#)
  - ProjectSettings, [60](#)
- ProjectSettings
  - projectConfiguration::ProjectSettings, [60](#)
- STR\_INVALID
  - common::ErrorCode, [18](#)
- STR\_NAME
  - configurator::GPIO::CodeName, [11](#)
  - configurator::GPIO::Mode, [33](#)
  - configurator::GPIO::OutLevel, [35](#)
  - configurator::GPIO::OutType, [36](#)
  - configurator::GPIO::Pull, [63](#)
  - configurator::GPIO::Speed, [65](#)
- STR\_PROJ\_CONF\_FILE
  - configurator::ConfigurationFile, [15](#)
- SW\_VERSION
  - common::Features, [20](#)
- saveUc
  - gui::MainGui, [24](#)
- setAdc
  - microcontroller::Pin, [45](#)
- setClock

- microcontroller::Pin, [45](#)
- setCodeName
  - configurator::PinConf, [56](#)
- setFeat\_adc
  - microcontroller::Pin, [45](#)
- setFeat\_clock
  - microcontroller::Pin, [46](#)
- setFeat\_i2c
  - microcontroller::Pin, [46](#)
- setFeat\_int
  - microcontroller::Pin, [46](#)
- setFeat\_reset
  - microcontroller::Pin, [47](#)
- setFeat\_spi
  - microcontroller::Pin, [47](#)
- setFeat\_timer
  - microcontroller::Pin, [47](#)
- setFeat\_uart
  - microcontroller::Pin, [47](#)
- setFunc\_gnd
  - microcontroller::Pin, [48](#)
- setFunc\_gpio
  - microcontroller::Pin, [48](#)
- setFunc\_vcc
  - microcontroller::Pin, [48](#)
- setI2c
  - microcontroller::Pin, [49](#)
- setInstallationFwkPath
  - framework::Common, [14](#)
- setInt
  - microcontroller::Pin, [49](#)
- setMode
  - configurator::PinConf, [57](#)
- setName
  - microcontroller::Pin, [49](#)
- setNewUC
  - gui::MainGui, [24](#)
- setNumber
  - microcontroller::Pin, [50](#)
- setOutLevel
  - configurator::PinConf, [57](#)
- setOutType
  - configurator::PinConf, [57](#)
- setPort
  - microcontroller::Pin, [50](#)
- setProjectFwkPath
  - framework::Common, [14](#)
- setProjectInformation
  - gui::MainWindow, [27](#)
- setPull
  - configurator::PinConf, [58](#)
- setReset
  - microcontroller::Pin, [50](#)
- setSpeed
  - configurator::PinConf, [58](#)
- setSpi
  - microcontroller::Pin, [50](#)
- setTimer
  - microcontroller::Pin, [51](#)
- setUart
  - microcontroller::Pin, [51](#)
- setVisible
  - gui::MainWindow, [27](#)
- showAboutWindow
  - gui::MainGui, [24](#)
- showErrorDialog
  - gui::MainGui, [25](#)
- showGpioConfWindow
  - gui::MainGui, [25](#)
- VERBOSE\_STR
  - common::Features, [21](#)
- VERBOSE
  - common::Features, [20](#)
- VERSION\_NAME
  - common::Features, [21](#)
- VERSION\_STATUS
  - common::Features, [21](#)
- verbosePrint
  - common::Features, [20](#)
- writeXml
  - xmlCreator::ConfXmlWriter, [17](#)
- xmlCreator, [8](#)
- xmlCreator.ConfXmlWriter, [16](#)
- xmlCreator::ConfXmlWriter
  - addPin, [16](#)
  - ConfXmlWriter, [16](#)
  - writeXml, [17](#)
- XmlOpener
  - xmlParser::XmlOpener, [66](#)
- xmlParser, [8](#)
- xmlParser.TestMain, [65](#)
- xmlParser.XmlOpener, [66](#)
- xmlParser::TestMain
  - main, [65](#)
- xmlParser::XmlOpener
  - getElementInfo, [67](#)
  - getElementInfoFromDoc, [67](#)
  - getParsedDoc, [67](#)
  - OpenFile, [68](#)
  - XmlOpener, [66](#)