Code_generator

Generated by Doxygen 1.8.18

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 . . . 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

4 Class Documentation 9

4.1 gui.AboutWindow Class Reference . . . 9

4.1.1 Detailed Description . . . 9

4.1.2 Constructor & Destructor Documentation . . . 9

4.1.2.1 AboutWindow() . . . 9

4.1.3 Member Function Documentation . . . 10

4.1.3.1 main() . . . 10

4.2 microcontroller.Adc Class Reference . . . 10

4.3 configurator.AdcConf Class Reference . . . 10

4.4 gui.AdcConfWindow Class Reference . . . 10

4.4.1 Constructor & Destructor Documentation . . . 11

4.4.1.1 AdcConfWindow() . . . 11

4.4.2 Member Function Documentation . . . 11

4.4.2.1 main() . . . 11

4.5 configurator.GPIO.AltMode Enum Reference . . . 11

4.5.1 Detailed Description . . . 12

4.5.2 Member Function Documentation . . . 12
4.5.2.1 getConfFromString() . . . 12
4.5.3 Member Data Documentation . . . 13
4.5.3.1 ALT_MODE_ANALOG . . . 13
4.5.3.2 ALT_MODE_I2C . . . 13
4.5.3.3 ALT_MODE_MAX_VALUE . . . 13
4.5.3.4 ALT_MODE_NONE . . . 13
4.5.3.5 ALT_MODE_SPI . . . 13
4.5.3.6 ALT_MODE_UART . . . 13
4.5.3.7 STR_NAME . . . 14
4.6 framework.CodeGenerator Class Reference . . . 14
4.6.1 Detailed Description . . . 14
4.6.2 Constructor & Destructor Documentation . . . 14
4.6.2.1 CodeGenerator() . . . 14
4.6.3 Member Function Documentation . . . 15
4.6.3.1 Generate() . . . 15
4.7 configurator.GPIO.CodeName Enum Reference . . . 15
4.7.1 Detailed Description . . . 15
4.7.2 Member Data Documentation . . . 15
4.7.2.1 CODE_NAME . . . 16
4.7.2.2 STR_NAME . . . 16
4.8 framework.Common Class Reference . . . 16
4.8.1 Detailed Description . . . 17
4.8.2 Member Function Documentation . . . 17
4.8.2.1 getCfgFileCPath() . . . 17
4.8.2.2 getCfgFileHPath() . . . 17
4.8.2.3 getCfgPath() . . . 18
4.8.2.4 getCommonCfgDefinitions() . . . 18
4.8.2.5 getCommonIncludes() . . . 19
4.8.2.6 getFrameworkCommonFilePath() . . . 19
4.8.2.7 getFrameworkIncludesFilePath() . . . 19
4.8.2.8 getInstallationFwkPath() . . . 20
4.8.2.9 getProjectFwkPath() . . . 20
4.8.2.10 setInstallationFwkPath() . . . 20
4.8.2.11 setProjectFwkPath() . . . 21
4.8.3 Member Data Documentation . . . 21
4.8.3.1 NL . . . 21
4.8.3.2 STR_DEFINITION . . . 21
4.8.3.3 STR_GEN_CODE_NOTICE_FOOTER . . . 21

4.8.3.4 STR_GEN_CODE_NOTICE_HEADER . . . 22
4.8.3.5 STR_HEADER_EXT . . . 22
4.8.3.6 STR_INCLUDE . . . 22
4.8.3.7 STR_MODULE_GPIO . . . 22
4.9 configurator.ConfigurationFile Class Reference . . . 22
4.9.1 Detailed Description . . . 23
4.9.2 Member Data Documentation . . . 23
4.9.2.1 STR_PROJ_CONF_FILE . . . 23
4.10 xmlCreator.ConfXmlWriter Class Reference . . . 23
4.10.1 Detailed Description . . . 23
4.10.2 Constructor & Destructor Documentation . . . 24
4.10.2.1 ConfXmlWriter() . . . 24
4.10.3 Member Function Documentation . . . 24
4.10.3.1 addPin() . . . 24
4.10.3.2 writeXml() . . . 24
4.11 common.ErrorCode Enum Reference . . . 25
4.11.1 Detailed Description . . . 25
4.11.2 Member Data Documentation . . . 25
4.11.2.1 EX_ERROR . . . 26
4.11.2.2 FILE_CONF_ERROR . . . 26
4.11.2.3 FILE_READ_ERROR . . . 26
4.11.2.4 FILE_WRITE_ERROR . . . 26
4.11.2.5 NO_ERROR . . . 26
4.11.2.6 STR_INVALID . . . 26
4.12 common.Features Class Reference . . . 27
4.12.1 Detailed Description . . . 27
4.12.2 Member Function Documentation . . . 27
4.12.2.1 debugPrint() . . . 27
4.12.2.2 verbosePrint() . . . 28
4.12.3 Member Data Documentation . . . 28
4.12.3.1 DEBUG . . . 28
4.12.3.2 DEBUG_STR . . . 28
4.12.3.3 SW_VERSION . . . 28
4.12.3.4 VERBOSE . . . 29
4.12.3.5 VERBOSE_STR . . . 29
4.12.3.6 VERSION_NAME . . . 29
4.12.3.7 VERSION_STATUS . . . 29
4.13 gui.GpioConfWindow Class Reference . . . 29
4.13.1 Detailed Description . . . 30

4.13.2 Constructor & Destructor Documentation . . . 30
4.13.2.1 GpioConfWindow() . . . 30
4.13.3 Member Function Documentation . . . 30
4.13.3.1 main() . . . 30
4.14 gui.MainGui Class Reference . . . 31
4.14.1 Detailed Description . . . 31
4.14.2 Member Function Documentation . . . 31
4.14.2.1 generateCode() . . . 32
4.14.2.2 loadProjectFile() . . . 32
4.14.2.3 main() . . . 32
4.14.2.4 saveUc() . . . 32
4.14.2.5 setNewUC() . . . 33
4.14.2.6 showAboutWindow() . . . 33
4.14.2.7 showAdcConfWindow() . . . 33
4.14.2.8 showErrorDialog() . . . 33
4.14.2.9 showGpioConfWindow() . . . 33
4.14.3 Member Data Documentation . . . 34
4.14.3.1 ProjectFile . . . 34
4.14.3.2 ProjectPath . . . 34
4.15 gui.MainWindow Class Reference . . . 34
4.15.1 Detailed Description . . . 35
4.15.2 Constructor & Destructor Documentation . . . 35
4.15.2.1 MainWindow() . . . 35
4.15.3 Member Function Documentation . . . 35
4.15.3.1 main() . . . 35
4.15.3.2 OpenFileChooser() . . . 35
4.15.3.3 setProjectInformation() . . . 36
4.15.3.4 setVisible() . . . 36
4.15.4 Member Data Documentation . . . 37
4.15.4.1 FrmCodeGenerator . . . 37
4.16 gui.Messages Class Reference . . . 37
4.16.1 Detailed Description . . . 37
4.16.2 Member Function Documentation . . . 37
4.16.2.1 getString() . . . 37
4.17 microcontroller.Microcontroller Class Reference . . . 38
4.17.1 Detailed Description . . . 39
4.17.2 Constructor & Destructor Documentation . . . 39
4.17.2.1 Microcontroller() . . . 39
4.17.3 Member Function Documentation . . . 39

4.17.3.1 getConfiguredPin() . . . 39
4.17.3.2 getPin() . . . 40
4.17.3.3 getUc_gpioNum() . . . 40
4.17.3.4 getUc_manufacturer() . . . 40
4.17.3.5 getUc_model() . . . 41
4.17.3.6 getUc_pinNum() . . . 41
4.17.3.7 getUc_portNum() . . . 41
4.17.3.8 getUc_selectedPinsNum() . . . 41
4.17.3.9 isValid() . . . 42
4.17.3.10 loadPinsConf() . . . 42
4.17.3.11 processDocument() . . . 42
4.17.4 Member Data Documentation . . . 42
4.17.4.1 AdcCfg . . . 43
4.17.4.2 Definitions_Common . . . 43
4.17.4.3 Definitions_Gpio . . . 43
4.17.4.4 GpioCfgPin . . . 43
4.17.4.5 Includes_Common . . . 43
4.17.4.6 Includes_Gpio . . . 43
4.17.4.7 MAX_NUMBER_OF_PINS_PER_PORT . . . 43
4.17.4.8 Ports . . . 44
4.18 configurator.GPIO.Mode Enum Reference . . . 44
4.18.1 Detailed Description . . . 44
4.18.2 Member Function Documentation . . . 44
4.18.2.1 getConfFromString() . . . 44
4.18.3 Member Data Documentation . . . 45
4.18.3.1 MODE_ALTERNATE_FUNCTION . . . 45
4.18.3.2 MODE_INPUT . . . 45
4.18.3.3 MODE_MAX_VALUE . . . 45
4.18.3.4 MODE_OUTPUT . . . 45
4.18.3.5 STR_NAME . . . 45
4.19 configurator.GPIO.OutLevel Enum Reference . . . 46
4.19.1 Detailed Description . . . 46
4.19.2 Member Function Documentation . . . 46
4.19.2.1 getConfFromString() . . . 46
4.19.3 Member Data Documentation . . . 47
4.19.3.1 HIGH . . . 47
4.19.3.2 LOW . . . 47
4.19.3.3 MAX_VALUE . . . 47
4.19.3.4 STR_NAME . . . 47

4.20 configurator.GPIO.OutType Enum Reference . . . 47
4.20.1 Detailed Description . . . 48
4.20.2 Member Function Documentation . . . 48
4.20.2.1 getConfFromString() . . . 48
4.20.3 Member Data Documentation . . . 49
4.20.3.1 OTYPE_MAX_VALUE . . . 49
4.20.3.2 OTYPE_NOT_AVAILABLE . . . 49
4.20.3.3 OTYPE_OPEN_DRAIN . . . 49
4.20.3.4 OTYPE_PUSH_PULL . . . 49
4.20.3.5 STR_NAME . . . 49
4.21 microcontroller.Pin Class Reference . . . 50
4.21.1 Detailed Description . . . 51
4.21.2 Constructor & Destructor Documentation . . . 52
4.21.2.1 Pin() . . . 52
4.21.3 Member Function Documentation . . . 52
4.21.3.1 getAdc() . . . 52
4.21.3.2 getClock() . . . 52
4.21.3.3 getFeat_adc() . . . 53
4.21.3.4 getFeat_clock() . . . 53
4.21.3.5 getFeat_i2c() . . . 53
4.21.3.6 getFeat_int() . . . 53
4.21.3.7 getFeat_reset() . . . 54
4.21.3.8 getFeat_spi() . . . 54
4.21.3.9 getFeat_timer() . . . 54
4.21.3.10 getFeat_uart() . . . 54
4.21.3.11 getFunc_gnd() . . . 55
4.21.3.12 getFunc_gpio() . . . 55
4.21.3.13 getFunc_misc() . . . 55
4.21.3.14 getFunc_reset() . . . 55
4.21.3.15 getFunc_vcc() . . . 56
4.21.3.16 getI2c() . . . 56
4.21.3.17 getInt() . . . 56
4.21.3.18 getName() . . . 56
4.21.3.19 getNumber() . . . 57
4.21.3.20 getPort() . . . 57
4.21.3.21 getPortPin() . . . 57
4.21.3.22 getReset() . . . 57
4.21.3.23 getSpi() . . . 58
4.21.3.24 getTimer() . . . 58

4.21.3.25 getUart() . . . 58
4.21.3.26 isValid() . . . 58
4.21.3.27 setAdc() . . . 58
4.21.3.28 setClock() . . . 59
4.21.3.29 setFeat_adc() . . . 59
4.21.3.30 setFeat_clock() . . . 59
4.21.3.31 setFeat_i2c() . . . 60
4.21.3.32 setFeat_int() . . . 60
4.21.3.33 setFeat_reset() . . . 60
4.21.3.34 setFeat_spi() . . . 60
4.21.3.35 setFeat_timer() . . . 61
4.21.3.36 setFeat_uart() . . . 61
4.21.3.37 setFunc_gnd() . . . 61
4.21.3.38 setFunc_gpio() . . . 62
4.21.3.39 setFunc_misc() . . . 62
4.21.3.40 setFunc_reset() . . . 62
4.21.3.41 setFunc_vcc() . . . 63
4.21.3.42 setI2c() . . . 63
4.21.3.43 setInt() . . . 63
4.21.3.44 setName() . . . 63
4.21.3.45 setNumber() . . . 64
4.21.3.46 setPort() . . . 64
4.21.3.47 setPortPin() . . . 64
4.21.3.48 setReset() . . . 65
4.21.3.49 setSpi() . . . 65
4.21.3.50 setTimer() . . . 65
4.21.3.51 setUart() . . . 66
4.21.4 Member Data Documentation . . . 66
4.21.4.1 DEF_FEATURE . . . 66
4.21.4.2 DEF_FEATURE_AV . . . 66
4.21.4.3 DEF_FUNCTION . . . 66
4.21.4.4 DEF_NAME . . . 66
4.21.4.5 DEF_NUMBER . . . 67
4.21.4.6 DEF_PORT . . . 67
4.21.4.7 DISABLE . . . 67
4.21.4.8 ENABLE . . . 67
4.22 configurator.PinConf Class Reference . . . 67
4.22.1 Detailed Description . . . 68
4.22.2 Constructor & Destructor Documentation . . . 68

4.22.2.1 PinConf() . . . 68
4.22.3 Member Function Documentation . . . 69
4.22.3.1 getAltMode() . . . 69
4.22.3.2 getCodeName() . . . 69
4.22.3.3 getMode() . . . 69
4.22.3.4 getOutLevel() . . . 70
4.22.3.5 getOutType() . . . 70
4.22.3.6 getPinName() . . . 70
4.22.3.7 getPort() . . . 70
4.22.3.8 getPortPin() . . . 71
4.22.3.9 getPull() . . . 71
4.22.3.10 getSelected() . . . 71
4.22.3.11 getSpeed() . . . 71
4.22.3.12 isAv_Adc() . . . 72
4.22.3.13 isAv_altFunc() . . . 72
4.22.3.14 isAv_I2c() . . . 72
4.22.3.15 isAv_Spi() . . . 72
4.22.3.16 isAv_Uart() . . . 73
4.22.3.17 isValid() . . . 73
4.22.3.18 setAltMode() . . . 73
4.22.3.19 setCodeName() . . . 73
4.22.3.20 setMode() . . . 74
4.22.3.21 setOutLevel() . . . 74
4.22.3.22 setOutType() . . . 74
4.22.3.23 setPull() . . . 75
4.22.3.24 setSelected() . . . 75
4.22.3.25 setSpeed() . . . 75
4.22.4 Member Data Documentation . . . 75
4.22.4.1 DF_ALT_MODE . . . 76
4.22.4.2 DF_CODE_NAME . . . 76
4.22.4.3 DF_MODE . . . 76
4.22.4.4 DF_OUT_LEVEL . . . 76
4.22.4.5 DF_OUTTYPE . . . 76
4.22.4.6 DF_PULL . . . 76
4.22.4.7 DF_SELECTED . . . 76
4.22.4.8 DF_SPEED . . . 77
4.23 projectConfiguration.ProjectSettings Class Reference . . . 77
4.23.1 Detailed Description . . . 77
4.23.2 Constructor & Destructor Documentation . . . 77

4.23.2.1 ProjectSettings() . . . 77
4.23.3 Member Function Documentation . . . 78
4.23.3.1 getConfFile() . . . 78
4.23.3.2 getFrameworkPath() . . . 78
4.23.3.3 getProjectName() . . . 78
4.23.3.4 getUcFile() . . . 78
4.23.3.5 openProjectFile() . . . 78
4.23.3.6 processDocument() . . . 79
4.24 configurator.GPIO.Pull Enum Reference . . . 79
4.24.1 Detailed Description . . . 80
4.24.2 Member Function Documentation . . . 80
4.24.2.1 getConfFromString() . . . 80
4.24.3 Member Data Documentation . . . 80
4.24.3.1 PULL_DOWN . . . 80
4.24.3.2 PULL_MAX_VALUE . . . 81
4.24.3.3 PULL_NOT_AVAILABLE . . . 81
4.24.3.4 PULL_UP . . . 81
4.24.3.5 STR_NAME . . . 81
4.25 configurator.GPIO.Selected Enum Reference . . . 81
4.25.1 Detailed Description . . . 82
4.25.2 Member Function Documentation . . . 82
4.25.2.1 getBoolean() . . . 82
4.25.2.2 getConfFromBoolean() . . . 82
4.25.2.3 getConfFromString() . . . 83
4.25.3 Member Data Documentation . . . 83
4.25.3.1 NOT . . . 83
4.25.3.2 STR_NAME . . . 83
4.25.3.3 YES . . . 83
4.26 configurator.GPIO.Speed Enum Reference . . . 84
4.26.1 Detailed Description . . . 84
4.26.2 Member Function Documentation . . . 84
4.26.2.1 getConfFromString() . . . 84
4.26.3 Member Data Documentation . . . 85
4.26.3.1 SPEED_FAST . . . 85
4.26.3.2 SPEED_HIGH . . . 85
4.26.3.3 SPEED_MAX_VALUE . . . 85
4.26.3.4 SPEED_MEDIUM . . . 85
4.26.3.5 SPEED_NOT_AVAILABLE . . . 85
4.26.3.6 STR_NAME . . . 86

4.27 xmlParser.XmlOpener Class Reference 86
4.27.1 Detailed Description 86
4.27.2 Constructor & Destructor Documentation 86
4.27.2.1 XmlOpener() 86
4.27.3 Member Function Documentation 87
4.27.3.1 getElementInfo() 87
4.27.3.2 getElementInfoFromDoc() 87
4.27.3.3 getParsedDoc() 87
4.27.3.4 OpenFile() 88

Index **89**

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 8
xmlParser 8

Chapter 2

Class Index

2.1 Class List

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

gui.AboutWindow . . . 9
microcontroller.Adc . . . 10
configurator.AdcConf . . . 10
gui.AdcConfWindow . . . 10
configurator.GPIO.AltMode . . . 11
framework.CodeGenerator . . . 14
configurator.GPIO.CodeName . . . 15
framework.Common . . . 16
configurator.ConfigurationFile . . . 22
xmlCreator.ConfXmlWriter . . . 23
common.ErrorCode . . . 25
common.Features . . . 27
gui.GpioConfWindow . . . 29
gui.MainGui . . . 31
gui.MainWindow . . . 34
gui.Messages . . . 37
microcontroller.Microcontroller . . . 38
configurator.GPIO.Mode . . . 44
configurator.GPIO.OutLevel . . . 46
configurator.GPIO.OutType . . . 47
microcontroller.Pin . . . 50
configurator.PinConf . . . 67
projectConfiguration.ProjectSettings . . . 77
configurator.GPIO.Pull . . . 79
configurator.GPIO.Selected . . . 81
configurator.GPIO.Speed . . . 84
xmlParser.XmlOpener . . . 86

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 AdcConf
- 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 AdcConfWindow
- 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 Adc
- 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 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 Detailed Description

About Window, contains version and contact information

Author

ovd

4.1.2 Constructor & Destructor Documentation

4.1.2.1 AboutWindow()

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

Create the application.

4.1.3 Member Function Documentation

4.1.3.1 main()

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

About window main

Parameters

args	Init parameters

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

- src/gui/AboutWindow.java

4.2 microcontroller.Adc Class Reference

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

- src/microcontroller/Adc.java

4.3 configurator.AdcConf Class Reference

Public Member Functions

- **AdcConf** (Adc adc)

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

- src/configurator/AdcConf.java

4.4 gui.AdcConfWindow Class Reference

Public Member Functions

- AdcConfWindow (Microcontroller uCtrl)

Static Public Member Functions

- static void main (String[ ] args)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 AdcConfWindow()

```
gui.AdcConfWindow.AdcConfWindow (
            Microcontroller uCtrl )
```

Create the application.

Parameters

uCtrl	Microcontroller

4.4.2 Member Function Documentation

4.4.2.1 main()

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

Launch the application.

Parameters

args	General arguments

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

- src/gui/AdcConfWindow.java

4.5 configurator.GPIO.AltMode Enum Reference

Static Public Member Functions

- static AltMode getConfFromString (String conf)

Public Attributes

- ALT_MODE_ANALOG
- ALT_MODE_UART
- ALT_MODE_I2C
- ALT_MODE_SPI
- ALT_MODE_NONE
- ALT_MODE_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "AltMode"

4.5.1 Detailed Description

GPIO modes

Author

Miguel Diaz

Version

0.1

4.5.2 Member Function Documentation

4.5.2.1 getConfFromString()

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

Get the corresponding mode from its name as String

Parameters

conf	Configuration name

Returns

Mode

4.5.3 Member Data Documentation

4.5.3.1 ALT_MODE_ANALOG

`configurator.GPIO.AltMode.ALT_MODE_ANALOG`

Analog

4.5.3.2 ALT_MODE_I2C

`configurator.GPIO.AltMode.ALT_MODE_I2C`

I2C

4.5.3.3 ALT_MODE_MAX_VALUE

`configurator.GPIO.AltMode.ALT_MODE_MAX_VALUE`

Maximum value for Mode enum

4.5.3.4 ALT_MODE_NONE

`configurator.GPIO.AltMode.ALT_MODE_NONE`

No alternate mode

4.5.3.5 ALT_MODE_SPI

`configurator.GPIO.AltMode.ALT_MODE_SPI`

SPI

4.5.3.6 ALT_MODE_UART

`configurator.GPIO.AltMode.ALT_MODE_UART`

UART

4.5.3.7 STR_NAME

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

Name as String

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

- src/configurator/GPIO/AltMode.java

4.6 framework.CodeGenerator Class Reference

Public Member Functions

- CodeGenerator (Microcontroller uC, ProjectSettings projectSettings)
- ErrorCode Generate ()

Static Public Attributes

- static final String **STR_TKN_CFG_DEFS_COMMON** = "FWK_GPIO_COMMON_DEFINITIONS"
- static final String **STR_TKN_CFG_DEFS_GPIO** = "FWK_GPIO_CFG_DEFINITIONS"

4.6.1 Detailed Description

Author

ovd

4.6.2 Constructor & Destructor Documentation

4.6.2.1 CodeGenerator()

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

Constructor

Parameters

uC	Project's microcontroller
projectSettings	Project's settings

4.6.3 Member Function Documentation

4.6.3.1 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

4.7 configurator.GPIO.CodeName Enum Reference

Public Attributes

- CODE_NAME

Static Public Attributes

- static final String STR_NAME = "codeName"

4.7.1 Detailed Description

Author

Miguel Diaz

Version

0.1

4.7.2 Member Data Documentation

4.7.2.1 CODE_NAME

```
configurator.GPIO.CodeName.CODE_NAME
```

Code name for pin

4.7.2.2 STR_NAME

```
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

4.8 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 String getFrameworkCommonFilePath (String fwkPath)
- static String getFrameworkIncludesFilePath (String fwkPath)
- static String getCommonIncludes (Microcontroller uC)
- static String getCommonCfgDefinitions (Microcontroller uC)

Static Public Attributes

- static final String NL = "\r\n"
- static final String STR_GEN_CODE_NOTICE_HEADER
- static final String STR_GEN_CODE_NOTICE_FOOTER
- static final String STR_MODULE_GPIO = "gpio"
- static final String STR_DEFINITION = "#define "
- static final String STR_INCLUDE = "#include "
- static final String STR_HEADER_EXT = ".h"

4.8.1 Detailed Description

Framework common fields and methods

Author

Miguel Diaz

Version

0.1

4.8.2 Member Function Documentation

4.8.2.1 getCfgFileCPath()

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

Get GPIO configuration file path

Parameters

fwkPath	Framework folder path
cfgModule	Configuration module name

Returns

GPIO configuration file path

4.8.2.2 getCfgFileHPath()

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

Get GPIO configuration header file path

Parameters

fwkPath	Framework folder path
cfgModule	Configuration module name

Returns

GPIO configuration header file path

4.8.2.3 getCfgPath()

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

Get configuration module files folder path

Parameters

fwkPath	Framework folder path
cfgModule	Configuration module name

Returns

Configuration files folder path

4.8.2.4 getCommonCfgDefinitions()

```
static String framework.Common.getCommonCfgDefinitions (
            Microcontroller uC )  [static]
```

Get Framework Common definitions

Parameters

uC	Microcontroller used

Returns

Common definitions needed for framework

4.8.2.5 getCommonIncludes()

```
static String framework.Common.getCommonIncludes (
            Microcontroller uC ) [static]
```

Get Framework common headers

Parameters

uC	Microcontroller used

Returns

Common headers needed for framework

4.8.2.6 getFrameworkCommonFilePath()

```
static String framework.Common.getFrameworkCommonFilePath (
            String fwkPath ) [static]
```

Get the framework common header path

Parameters

fwkPath	Framework folder path

Returns

Framework common header path

4.8.2.7 getFrameworkIncludesFilePath()

```
static String framework.Common.getFrameworkIncludesFilePath (
            String fwkPath ) [static]
```

Get the framework includes header path

Parameters

fwkPath	Framework folder path

Returns

Framework includes header path

4.8.2.8 getInstallationFwkPath()

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

Get installation framework path

Returns

installation framework path

4.8.2.9 getProjectFwkPath()

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

Get project's framework path

Returns

project's framework path

4.8.2.10 setInstallationFwkPath()

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

Set installation framework path

Parameters

installationFwkPath	installation framework path

4.8.2.11 setProjectFwkPath()

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

Set project's framework path

Parameters

projectFwkPath	project's framework path

4.8.3 Member Data Documentation

4.8.3.1 NL

```
final String framework.Common.NL = "\r\n"  [static]
```

Common implementation of New Line

4.8.3.2 STR_DEFINITION

```
final String framework.Common.STR_DEFINITION = "#define "  [static]
```

Macro definition String

4.8.3.3 STR_GEN_CODE_NOTICE_FOOTER

```
final String framework.Common.STR_GEN_CODE_NOTICE_FOOTER  [static]
```

Initial value:

```
= "// ################### " + Features.GENERATOR_NAME
            + " generator v" + common.Features.SW_VERSION + ": Generated code! #################" + NL
            + "// ######## Do NOT modify code between this footer and the header above #######"
```

Footer for indicating generated code

4.8.3.4 STR_GEN_CODE_NOTICE_HEADER

```
final String framework.Common.STR_GEN_CODE_NOTICE_HEADER  [static]
```

Initial value:

```
= "// ################## " + Features.GENERATOR_NAME
            + " generator v" + common.Features.SW_VERSION + ": Generated code! ################" + NL
            + "// ######## Do NOT modify code between this header and the footer below #######"
```

Header for indicating generated code

4.8.3.5 STR_HEADER_EXT

```
final String framework.Common.STR_HEADER_EXT = ".h"  [static]
```

Header file extension

4.8.3.6 STR_INCLUDE

```
final String framework.Common.STR_INCLUDE = "#include "  [static]
```

Include header file string

4.8.3.7 STR_MODULE_GPIO

```
final String framework.Common.STR_MODULE_GPIO = "gpio"  [static]
```

GPIO module name

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

- src/framework/Common.java

4.9 configurator.ConfigurationFile Class Reference

Static Public Attributes

- static final String STR_PROJ_CONF_FILE = "cgs"

4.9.1 Detailed Description

Configuration files properties

Author

Miguel Diaz

Version

0.1

4.9.2 Member Data Documentation

4.9.2.1 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

4.10 xmlCreator.ConfXmlWriter Class Reference

Public Member Functions

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

4.10.1 Detailed Description

Write a XML file

Author

Miguel Diaz

Version

0.1

4.10.2 Constructor & Destructor Documentation

4.10.2.1 ConfXmlWriter()

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

Constructor

Parameters

uC	Microcontroller configuration

4.10.3 Member Function Documentation

4.10.3.1 addPin()

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

Add a pin configuration to the file

Parameters

pin	Pin configuration
pinNum	Number of GPIO pin

4.10.3.2 writeXml()

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

Write the XML file

Parameters

fileName	Name of XML configuration file

Returns

Error status

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

- src/xmlCreator/ConfXmlWriter.java

4.11 common.ErrorCode Enum Reference

Public Attributes

- NO_ERROR
- EX_ERROR
- FILE_READ_ERROR
- FILE_WRITE_ERROR
- FILE_CONF_ERROR

Static Public Attributes

- static final String STR_INVALID = "STR_INVALID"

4.11.1 Detailed Description

Error codes enum

Author

Miguel Diaz

Version

0.1

4.11.2 Member Data Documentation

4.11.2.1 EX_ERROR

`common.ErrorCode.EX_ERROR`

Error during execution

4.11.2.2 FILE_CONF_ERROR

`common.ErrorCode.FILE_CONF_ERROR`

File configuration error

4.11.2.3 FILE_READ_ERROR

`common.ErrorCode.FILE_READ_ERROR`

File reading error

4.11.2.4 FILE_WRITE_ERROR

`common.ErrorCode.FILE_WRITE_ERROR`

File writing error

4.11.2.5 NO_ERROR

`common.ErrorCode.NO_ERROR`

No error message

4.11.2.6 STR_INVALID

```
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

4.12 common.Features Class Reference

Static Public Member Functions

- static void verbosePrint (String verboseMessage)
- static void debugPrint (String debugMessage)
- static void **initLog** ()

Static Public Attributes

- static final boolean DEBUG = true
- static final boolean VERBOSE = true
- static boolean **LOG_FILE** = 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 = "Dagobah"
- static final String **GENERATOR_NAME** = "Kamino"

4.12.1 Detailed Description

Class that includes all project features

Author

Miguel Diaz

Version

0.1

4.12.2 Member Function Documentation

4.12.2.1 debugPrint()

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

Print Debug message to console

Parameters

debugMessage	Message to display

4.12.2.2 verbosePrint()

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

Print Verbose message to console

Parameters

verboseMessage	Message to display

4.12.3 Member Data Documentation

4.12.3.1 DEBUG

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

Enables debug functions

4.12.3.2 DEBUG_STR

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

Debug messages indicator on system console

4.12.3.3 SW_VERSION

```
final String common.Features.SW_VERSION = VERSION_MAJOR + "." + VERSION_MINOR + "." + VERSION_PA↩
TCH  [static]
```

Complete Software version

4.12.3.4 VERBOSE

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

Enables console messages

4.12.3.5 VERBOSE_STR

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

Verbose messages indicator on system console

4.12.3.6 VERSION_NAME

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

Code name of the software version

4.12.3.7 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

4.13 gui.GpioConfWindow Class Reference

Public Member Functions

- GpioConfWindow (Microcontroller uCtrl)

Static Public Member Functions

- static void main (String[ ] args)

4.13.1 Detailed Description

Window for configuring GPIO pins

Author

Miguel Diaz

Version

0.1

4.13.2 Constructor & Destructor Documentation

4.13.2.1 GpioConfWindow()

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

Create the GPIO configuration window and show it

Parameters

uCtrl	Microcontroller object containing all pin's information

4.13.3 Member Function Documentation

4.13.3.1 main()

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

Gpio configuration window main

Parameters

args	Init parameters

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

- src/gui/GpioConfWindow.java

4.14 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 showAdcConfWindow ()
- static void setNewUC (Microcontroller uC)
- static void saveUc ()
- static ErrorCode generateCode ()

Static Public Attributes

- static File ProjectFile
- static String ProjectPath

4.14.1 Detailed Description

Main GUI state machine

Author

Miguel Diaz

Version

0.1

4.14.2 Member Function Documentation

4.14.2.1 generateCode()

```
static ErrorCode gui.MainGui.generateCode ( )  [static]
```

Generate source code files

Returns

Error code

4.14.2.2 loadProjectFile()

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

Load the project settings file

Parameters

inFile	Settings file

Returns

Error status

4.14.2.3 main()

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

Parameters

args	TBD

4.14.2.4 saveUc()

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

Save the microcontroller's configuration to disk

4.14.2.5 setNewUC()

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

Set the project's microcontroller configuration

Parameters

uC	Microcontroller configuration

4.14.2.6 showAboutWindow()

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

Show about information window

4.14.2.7 showAdcConfWindow()

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

Show the ADCs configuration window

4.14.2.8 showErrorDialog()

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

Show an error dialog

Parameters

message	Message to display

4.14.2.9 showGpioConfWindow()

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

Show the GPIOs configuration window

4.14.3 Member Data Documentation

4.14.3.1 ProjectFile

```
File gui.MainGui.ProjectFile  [static]
```

Project configuration file

4.14.3.2 ProjectPath

```
String gui.MainGui.ProjectPath  [static]
```

Project's location

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

- src/gui/MainGui.java

4.15 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

4.15.1 Detailed Description

Main application window

Author

Miguel Diaz

Version

0.1

4.15.2 Constructor & Destructor Documentation

4.15.2.1 MainWindow()

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

Create the application.

4.15.3 Member Function Documentation

4.15.3.1 main()

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

Open main window

Parameters

args	To be determined

4.15.3.2 OpenFileChooser()

```
File gui.MainWindow.OpenFileChooser (
            String initialPath,
```

```
            String title,
            FileNameExtensionFilter fileFilter )
```

Open file chooser dialog and get the selected file

Parameters

initialPath	Path to search the file in
title	Dialog title
fileFilter	Extension filter

Returns

Selected file

4.15.3.3 setProjectInformation()

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

Set Project's name in its label

Parameters

projectName	Project's name
ucManufacturer	Microcontroller's manufacturer
ucName	Microcontroller's model

Returns

Error status

4.15.3.4 setVisible()

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

Set visibility of About window

Parameters

status	true if visible

4.15.4 Member Data Documentation

4.15.4.1 FrmCodeGenerator

```
JFrame gui.MainWindow.FrmCodeGenerator
```

Frame for the main Window

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

- src/gui/MainWindow.java

4.16 gui.Messages Class Reference

Static Public Member Functions

- static String getString (String key)

4.16.1 Detailed Description

Messages window

Author

ovd

4.16.2 Member Function Documentation

4.16.2.1 getString()

```
static String gui.Messages.getString (
            String key )  [static]
```

Get String

Parameters

key	Key

Returns

String

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

- src/gui/Messages.java

4.17 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 ()
- int getUc_selectedPinsNum ()
- PinConf getConfiguredPin (String gpioName)
- boolean isValid ()

Public Attributes

- String[ ] Ports
- String[ ] Includes_Common
- String[ ] Includes_Gpio
- String[ ] Definitions_Common
- String[ ] Definitions_Gpio
- PinConf[ ] GpioCfgPin
- AdcConf[ ] AdcCfg

Static Public Attributes

- static final int MAX_NUMBER_OF_PINS_PER_PORT = 32

4.17.1 Detailed Description

Microcontroller related methods

Author

Miguel Diaz

Version

0.1

4.17.2 Constructor & Destructor Documentation

4.17.2.1 Microcontroller()

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

Constructor

Parameters

ucDoc	Document obtained from XML file

4.17.3 Member Function Documentation

4.17.3.1 getConfiguredPin()

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

Get the configuration of a pin

Parameters

gpioName	Name of the pin

Returns

Pin configuration

4.17.3.2 getPin()

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

Get a pin's characteristics

Parameters

pinNum	Number of pin

Returns

Pin's characteristics

4.17.3.3 getUc_gpioNum()

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

Get the number of GPIOs in the microcontroller

Returns

Number of GPIOs

4.17.3.4 getUc_manufacturer()

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

Get the microcontroller's manufacturer

Returns

Microcontroller's manufacturer

4.17.3.5 getUc_model()

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

Get the microcontroller's model

Returns

Microcontroller's model

4.17.3.6 getUc_pinNum()

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

Get the microcontroller's pins number

Returns

Number of pins

4.17.3.7 getUc_portNum()

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

Get the number of ports in the microcontroller

Returns

Number of ports

4.17.3.8 getUc_selectedPinsNum()

```
int microcontroller.Microcontroller.getUc_selectedPinsNum ( )
```

Get the total pins selected

Returns

Total of pins selected

4.17.3.9 isValid()

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

Check if the microcontroller configuration is valid

Returns

true if valid

4.17.3.10 loadPinsConf()

```
ErrorCode microcontroller.Microcontroller.loadPinsConf (
            Document confDoc )
```

Load pins' configuration

Parameters

confDoc	Document with pins

Returns

Error Code

4.17.3.11 processDocument()

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

Process the document obtained from XML file

Returns

Error status

4.17.4 Member Data Documentation

4.17.4.1 AdcCfg

```
AdcConf [] microcontroller.Microcontroller.AdcCfg
```

Configured ADCs list

4.17.4.2 Definitions_Common

```
String [] microcontroller.Microcontroller.Definitions_Common
```

List of common definitions that will be available for all framework

4.17.4.3 Definitions_Gpio

```
String [] microcontroller.Microcontroller.Definitions_Gpio
```

List of definitions for GPIO module

4.17.4.4 GpioCfgPin

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

Configured pins list

4.17.4.5 Includes_Common

```
String [] microcontroller.Microcontroller.Includes_Common
```

List of common includes that will be available for all framework

4.17.4.6 Includes_Gpio

```
String [] microcontroller.Microcontroller.Includes_Gpio
```

List of Includes for GPIO module

4.17.4.7 MAX_NUMBER_OF_PINS_PER_PORT

```
final int microcontroller.Microcontroller.MAX_NUMBER_OF_PINS_PER_PORT = 32  [static]
```

Maximum number of pins allowed in a single port

4.17.4.8 Ports

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

Ports name list

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

- src/microcontroller/Microcontroller.java

4.18 configurator.GPIO.Mode Enum Reference

Static Public Member Functions

- static Mode getConfFromString (String conf)

Public Attributes

- MODE_INPUT
- MODE_OUTPUT
- MODE_ALTERNATE_FUNCTION
- MODE_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "Mode"

4.18.1 Detailed Description

GPIO modes

Author

Miguel Diaz

Version

0.1

4.18.2 Member Function Documentation

4.18.2.1 getConfFromString()

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

Get the corresponding mode from its name as String

Parameters

conf	Configuration name

Returns

Mode

4.18.3 Member Data Documentation

4.18.3.1 MODE_ALTERNATE_FUNCTION

```
configurator.GPIO.Mode.MODE_ALTERNATE_FUNCTION
```

Alternate function

4.18.3.2 MODE_INPUT

```
configurator.GPIO.Mode.MODE_INPUT
```

Input

4.18.3.3 MODE_MAX_VALUE

```
configurator.GPIO.Mode.MODE_MAX_VALUE
```

Maximum value for Mode enum

4.18.3.4 MODE_OUTPUT

```
configurator.GPIO.Mode.MODE_OUTPUT
```

Output

4.18.3.5 STR_NAME

```
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

4.19 configurator.GPIO.OutLevel Enum Reference

Static Public Member Functions

- static OutLevel getConfFromString (String conf)

Public Attributes

- LOW
- HIGH
- MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "OutLevel"

4.19.1 Detailed Description

Pin's output/input level

Author

Miguel Diaz

Version

0.1

4.19.2 Member Function Documentation

4.19.2.1 getConfFromString()

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

Get the corresponding mode from its name as String

Parameters

conf	Configuration name

Returns

level

4.19.3 Member Data Documentation

4.19.3.1 HIGH

`configurator.GPIO.OutLevel.HIGH`

High, logical 1, Vcc

4.19.3.2 LOW

`configurator.GPIO.OutLevel.LOW`

Low, logical 0, Ground

4.19.3.3 MAX_VALUE

`configurator.GPIO.OutLevel.MAX_VALUE`

Maximum value for OutLevel enum

4.19.3.4 STR_NAME

`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

4.20 configurator.GPIO.OutType Enum Reference

Static Public Member Functions

- static OutType getConfFromString (String conf)

Public Attributes

- OTYPE_PUSH_PULL
- OTYPE_OPEN_DRAIN
- OTYPE_NOT_AVAILABLE
- OTYPE_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "OutType"

4.20.1 Detailed Description

Pin's output type

Author

Miguel Diaz

Version

0.1

4.20.2 Member Function Documentation

4.20.2.1 getConfFromString()

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

Get the corresponding output type from its name as String

Parameters

conf	Configuration name

Returns

Output type

4.20.3 Member Data Documentation

4.20.3.1 OTYPE_MAX_VALUE

`configurator.GPIO.OutType.OTYPE_MAX_VALUE`

Maximum value for OutType enum

4.20.3.2 OTYPE_NOT_AVAILABLE

`configurator.GPIO.OutType.OTYPE_NOT_AVAILABLE`

If the pin is configured as input

4.20.3.3 OTYPE_OPEN_DRAIN

`configurator.GPIO.OutType.OTYPE_OPEN_DRAIN`

Open Drain

4.20.3.4 OTYPE_PUSH_PULL

`configurator.GPIO.OutType.OTYPE_PUSH_PULL`

Push Pull, totem

4.20.3.5 STR_NAME

```
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

4.21 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 setFunc_reset (boolean funcState)
- boolean getFunc_reset ()
- void setFunc_misc (boolean funcState)
- boolean getFunc_misc ()
- 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 ()
- String getPortPin ()
- void setPortPin (String portPin)
- 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

4.21.1 Detailed Description

Basic pin object.

- Pin necessary characteristics:
  - Name
  - Number
- Pin optional characteristics:
  - Port
- Pin main functions:
  - VCC
  - GND
  - GPIO
  - RESET
  - MISC
- Pin features:
  - Interruption
  - ADC
  - UART
  - I2C
  - SPI
  - Clock
  - Reset

Author

Miguel Diaz

Version

0.1

4.21.2 Constructor & Destructor Documentation

4.21.2.1 Pin()

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

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

4.21.3 Member Function Documentation

4.21.3.1 getAdc()

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

Get the pin's ADC name

Returns

Pin's ADC

4.21.3.2 getClock()

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

Get the pin's clock name

Returns

Pin's clock

4.21.3.3 getFeat_adc()

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

See if the pin has an ADC

Returns

Feature availability

4.21.3.4 getFeat_clock()

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

See if the pin supports a clock

Returns

Feature availability

4.21.3.5 getFeat_i2c()

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

See if the pin has I2C

Returns

Feature availability

4.21.3.6 getFeat_int()

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

See if the pin has an interruption

Returns

Feature availability

4.21.3.7 getFeat_reset()

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

See if the pin has a reset feature

Returns

Feature availability

4.21.3.8 getFeat_spi()

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

See if the pin has SPI

Returns

Feature availability

4.21.3.9 getFeat_timer()

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

See if the pin supports a timer

Returns

Feature availability

4.21.3.10 getFeat_uart()

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

See if the pin has a UART

Returns

Feature availability

4.21.3.11 getFunc_gnd()

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

See if the pin is GND

Returns

Function availability

4.21.3.12 getFunc_gpio()

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

See if the pin is GPIO

Returns

Function availability

4.21.3.13 getFunc_misc()

```
boolean microcontroller.Pin.getFunc_misc ( )
```

See if the pin is MISC

Returns

Function availability

4.21.3.14 getFunc_reset()

```
boolean microcontroller.Pin.getFunc_reset ( )
```

See if the pin is RESET

Returns

Function availability

4.21.3.15 getFunc_vcc()

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

See if the pin is Vcc

Returns

Function availability

4.21.3.16 getI2c()

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

Get the pin's I2C name

Returns

Pin's I2C

4.21.3.17 getInt()

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

Get the pin's interruption name

Returns

Pin's interruption

4.21.3.18 getName()

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

Get the pin's name

Returns

Pin's name

4.21.3.19 getNumber()

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

Get the pin's number

Returns

Pin's number

4.21.3.20 getPort()

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

Get the pin's port

Returns

Pin's port

4.21.3.21 getPortPin()

```
String microcontroller.Pin.getPortPin ( )
```

Get port pin number

Returns

port pin number

4.21.3.22 getReset()

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

Get the pin's reset name

Returns

Pin's reset

4.21.3.23 getSpi()

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

Get the pin's SPI name

Returns

Pin's SPI

4.21.3.24 getTimer()

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

Get the pin's timer name

Returns

Pin's timer

4.21.3.25 getUart()

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

Get the pin's UART name

Returns

Pin's UART

4.21.3.26 isValid()

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

Check if the pin is correctly initialized

Returns

True if the pin is correctly initialized

4.21.3.27 setAdc()

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

Set the pin's ADC

Parameters

feature	Pin's ADC

4.21.3.28 setClock()

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

Set the pin's clock

Parameters

feature	Pin's clock

4.21.3.29 setFeat_adc()

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

Set the pin's ADC feature

Parameters

featState	Feature availability

4.21.3.30 setFeat_clock()

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

Set the pin's Clock feature

Parameters

featState	Feature availability

4.21.3.31 setFeat_i2c()

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

Set the pin's I2C feature

Parameters

featState	Feature availability

4.21.3.32 setFeat_int()

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

Set the pin's interruption feature

Parameters

featState	Feature availability

4.21.3.33 setFeat_reset()

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

Set the pin's reset feature

Parameters

featState	Feature availability

4.21.3.34 setFeat_spi()

```
void microcontroller.Pin.setFeat_spi (
```

```
            boolean featState )
```

Set the pin's SPI feature

Parameters

featState	Feature availability

4.21.3.35 setFeat_timer()

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

Set the pin's timer feature

Parameters

featState	Feature availability

4.21.3.36 setFeat_uart()

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

Set the pin's UART feature

Parameters

featState	Feature availability

4.21.3.37 setFunc_gnd()

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

Set the pin to GND status

Parameters

funcState	Function availability

4.21.3.38 setFunc_gpio()

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

Set the pin to GPIO status

Parameters

funcState	Function availability

4.21.3.39 setFunc_misc()

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

Set the pin to MISC status

Parameters

funcState	Function availability

4.21.3.40 setFunc_reset()

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

Set the pin to RESET status

Parameters

funcState	Function availability

4.21.3.41 setFunc_vcc()

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

Set the pin to Vcc status

Parameters

funcState	Function availability

4.21.3.42 setI2c()

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

Set the pin's I2C

Parameters

feature	Pin's I2C

4.21.3.43 setInt()

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

Set the pin's interruption

Parameters

feature	Pin's interruption

4.21.3.44 setName()

```
void microcontroller.Pin.setName (
```

```
            String pinName )
```

Set the pin's name

Parameters

pinName	Pin's name

4.21.3.45 setNumber()

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

Set the pin's number

Parameters

pinNum	Pin's number

4.21.3.46 setPort()

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

Set the pin's port

Parameters

pinPort	Pin's port

4.21.3.47 setPortPin()

```
void microcontroller.Pin.setPortPin (
            String portPin )
```

Set port pin number

Parameters

portPin	Port pin number

4.21.3.48 setReset()

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

Set the pin's reset

Parameters

feature	Pin's reset

4.21.3.49 setSpi()

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

Set the pin's SPI

Parameters

feature	Pin's SPI

4.21.3.50 setTimer()

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

Set the pin's timer

Parameters

feature	Pin's timer

4.21.3.51 setUart()

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

Set the pin's UART

Parameters

feature	Pin's UART

4.21.4 Member Data Documentation

4.21.4.1 DEF_FEATURE

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

Default value for pin's feature as not available

4.21.4.2 DEF_FEATURE_AV

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

Default value for pin's feature availability as not available

4.21.4.3 DEF_FUNCTION

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

Default value for pin's function as not enabled

4.21.4.4 DEF_NAME

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

Default value for pin's name

4.21.4.5 DEF_NUMBER

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

Default value for pin's number

4.21.4.6 DEF_PORT

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

Default value for pin's port

4.21.4.7 DISABLE

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

Disable value for features and functions

4.21.4.8 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

4.22 configurator.PinConf Class Reference

Public Member Functions

- PinConf (Pin gpioPin)
- boolean isValid ()
- String getPort ()
- String getPortPin ()
- String getPinName ()
- String getCodeName ()
- void setCodeName (String name)
- Selected getSelected ()
- void setSelected (Selected selection)
- Mode getMode ()
- void setMode (Mode mode)
- AltMode getAltMode ()

- void setAltMode (AltMode altMode)
- 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_Uart ()
- boolean isAv_I2c ()
- boolean isAv_Spi ()
- boolean isAv_altFunc ()

Static Public Attributes

- static final Selected DF_SELECTED = Selected.NOT
- static final Mode DF_MODE = Mode.MODE_INPUT
- static final AltMode DF_ALT_MODE = AltMode.ALT_MODE_NONE
- 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 = ""

4.22.1 Detailed Description

GPIO pin configuration

Author

Miguel Diaz

Version

0.1

4.22.2 Constructor & Destructor Documentation

4.22.2.1 PinConf()

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

Constructor

Parameters

gpioPin	Pin information

4.22.3 Member Function Documentation

4.22.3.1 getAltMode()

```
AltMode configurator.PinConf.getAltMode ( )
```

Get pin's alternative mode

Returns

Alternative mode

4.22.3.2 getCodeName()

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

Get the pin's user selected name

Returns

pin's name

4.22.3.3 getMode()

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

Get the pin's mode configuration

Returns

Mode

4.22.3.4 getOutLevel()

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

Get the pin's output level

Returns

Pin's output level

4.22.3.5 getOutType()

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

Get the pin's output configuration

Returns

Output configuration

4.22.3.6 getPinName()

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

Get the pin's number

Returns

Pin's number

4.22.3.7 getPort()

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

Get the pin's port

Returns

Port

4.22.3.8 getPortPin()

```
String configurator.PinConf.getPortPin ( )
```

Get the port pin number

Returns

Port pin number

4.22.3.9 getPull()

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

Get the pin's pull resistor configuration

Returns

Pull Resistor configuration

4.22.3.10 getSelected()

```
Selected configurator.PinConf.getSelected ( )
```

Get the pin's selection

Returns

Selection

4.22.3.11 getSpeed()

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

Get the pin's speed

Returns

Speed

4.22.3.12 isAv_Adc()

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

Check availability of ADC

Returns

True if ADC is available

4.22.3.13 isAv_altFunc()

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

Check the availability of alternate function

Returns

True if alternate function is available

4.22.3.14 isAv_I2c()

```
boolean configurator.PinConf.isAv_I2c ( )
```

Check availability of I2C

Returns

True if I2C is available

4.22.3.15 isAv_Spi()

```
boolean configurator.PinConf.isAv_Spi ( )
```

Check availability of SPI

Returns

True if SPI is available

4.22.3.16 isAv_Uart()

```
boolean configurator.PinConf.isAv_Uart ( )
```

Check availability of UART

Returns

True id UART is available

4.22.3.17 isValid()

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

Check if the GPIO pin is valid

Returns

True if valid

4.22.3.18 setAltMode()

```
void configurator.PinConf.setAltMode (
            AltMode altMode )
```

Set pin's alternative mode

Parameters

altMode	Alternative mode

4.22.3.19 setCodeName()

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

Set the pin's user selected name

Parameters

name	Pin's name

4.22.3.20 setMode()

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

Set the pin's mode configuration

Parameters

mode	Mode

4.22.3.21 setOutLevel()

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

Set the pin's output level

Parameters

level	Pin's output level

4.22.3.22 setOutType()

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

Set the pin's output configuration

Parameters

outType	Output configuration

4.22.3.23 setPull()

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

Set the pull resistor configuration

Parameters

pull	Resistor configuration

4.22.3.24 setSelected()

```
void configurator.PinConf.setSelected (
            Selected selection )
```

Set the pin's selection

Parameters

selection	Selection

4.22.3.25 setSpeed()

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

Set the pin's speed

Parameters

speed	Speed

4.22.4 Member Data Documentation

4.22.4.1 DF_ALT_MODE

```
final AltMode configurator.PinConf.DF_ALT_MODE = AltMode.ALT_MODE_NONE  [static]
```

Default Pin alternative mode

4.22.4.2 DF_CODE_NAME

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

Default pin's code name

4.22.4.3 DF_MODE

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

Default Pin mode

4.22.4.4 DF_OUT_LEVEL

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

Default pin's output level

4.22.4.5 DF_OUTTYPE

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

Default pin's output type

4.22.4.6 DF_PULL

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

Default pin's pull resistor

4.22.4.7 DF_SELECTED

```
final Selected configurator.PinConf.DF_SELECTED = Selected.NOT  [static]
```

Default Pin's selection

4.22.4.8 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

4.23 projectConfiguration.ProjectSettings Class Reference

Public Member Functions

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

4.23.1 Detailed Description

Project settings class

Author

Miguel Diaz

Version

0.2

4.23.2 Constructor & Destructor Documentation

4.23.2.1 ProjectSettings()

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

Constructor

4.23.3 Member Function Documentation

4.23.3.1 getConfFile()

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

Get the project configuration file

Returns

Project configuration file

4.23.3.2 getFrameworkPath()

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

Get the framework folder

Returns

framework folder

4.23.3.3 getProjectName()

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

Get the project's name

Returns

Project's name

4.23.3.4 getUcFile()

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

Get the project microcontroller file

Returns

Project microcontroller file

4.23.3.5 openProjectFile()

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

Open the project settings file

Parameters

inFile	Project file

Returns

Error Status

4.23.3.6 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

4.24 configurator.GPIO.Pull Enum Reference

Static Public Member Functions

- static Pull getConfFromString (String conf)

Public Attributes

- PULL_UP
- PULL_DOWN
- PULL_NOT_AVAILABLE
- PULL_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "Pull"

4.24.1 Detailed Description

Pin's pull resistor

Author

Miguel Diaz

Version

0.1

4.24.2 Member Function Documentation

4.24.2.1 getConfFromString()

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

Get the corresponding Pull configuration from its name as String

Parameters

conf	Configuration name

Returns

Pull configuration

4.24.3 Member Data Documentation

4.24.3.1 PULL_DOWN

```
configurator.GPIO.Pull.PULL_DOWN
```

Pull Down

4.24.3.2 PULL_MAX_VALUE

`configurator.GPIO.Pull.PULL_MAX_VALUE`

Maximum value for Pull enum

4.24.3.3 PULL_NOT_AVAILABLE

`configurator.GPIO.Pull.PULL_NOT_AVAILABLE`

If the pin is configured as output, or there is no resistor available

4.24.3.4 PULL_UP

`configurator.GPIO.Pull.PULL_UP`

Pull Up

4.24.3.5 STR_NAME

`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

4.25 configurator.GPIO.Selected Enum Reference

Public Member Functions

- boolean getBoolean ()

Static Public Member Functions

- static Selected getConfFromString (String conf)
- static Selected getConfFromBoolean (Boolean conf)

Public Attributes

- NOT
- YES

Static Public Attributes

- static final String STR_NAME = "selected"

4.25.1 Detailed Description

Pin's selection

Author

Miguel Díaz

Version

0.1

4.25.2 Member Function Documentation

4.25.2.1 getBoolean()

```
boolean configurator.GPIO.Selected.getBoolean ( )
```

Get the corresponding boolean from its selection

Returns

Selected pin state

4.25.2.2 getConfFromBoolean()

```
static Selected configurator.GPIO.Selected.getConfFromBoolean (
            Boolean conf )  [static]
```

Get the corresponding mode from a boolean

Parameters

conf	Configuration value

Returns

Selected

4.25.2.3 getConfFromString()

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

Get the corresponding mode from its name as String

Parameters

conf	Configuration name

Returns

Selected

4.25.3 Member Data Documentation

4.25.3.1 NOT

```
configurator.GPIO.Selected.NOT
```

Pin not selected

4.25.3.2 STR_NAME

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

Name as String

4.25.3.3 YES

```
configurator.GPIO.Selected.YES
```

Pin selected

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

- src/configurator/GPIO/Selected.java

4.26 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
- SPEED_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "Speed"

4.26.1 Detailed Description

Pin's speed

Author

Miguel Diaz

Version

0.1

4.26.2 Member Function Documentation

4.26.2.1 getConfFromString()

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

Get the corresponding Speed configuration from its name as String

Parameters

conf	Configuration name

Returns

Speed

4.26.3 Member Data Documentation

4.26.3.1 SPEED_FAST

`configurator.GPIO.Speed.SPEED_FAST`

Fast

4.26.3.2 SPEED_HIGH

`configurator.GPIO.Speed.SPEED_HIGH`

High

4.26.3.3 SPEED_MAX_VALUE

`configurator.GPIO.Speed.SPEED_MAX_VALUE`

Maximum value for Speed enum

4.26.3.4 SPEED_MEDIUM

`configurator.GPIO.Speed.SPEED_MEDIUM`

Medium

4.26.3.5 SPEED_NOT_AVAILABLE

`configurator.GPIO.Speed.SPEED_NOT_AVAILABLE`

Not all MCUs will have this setting

4.26.3.6 STR_NAME

```
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

4.27 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)

4.27.1 Detailed Description

Open and process XML files

Author

H112943

Version

0.1

4.27.2 Constructor & Destructor Documentation

4.27.2.1 XmlOpener()

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

Constructor

4.27.3 Member Function Documentation

4.27.3.1 getElementInfo()

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

Get an XML sub element information

Parameters

element	XML main element
elementName	Sub element's name

Returns

Sub elemen't information

4.27.3.2 getElementInfoFromDoc()

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

Get an XML element information

Parameters

doc	Document from XML file
elementName	Element's name

Returns

Element's information

4.27.3.3 getParsedDoc()

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

Get the parsed document AFTER opening the file

Returns

Parsed document

4.27.3.4 OpenFile()

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

Open the XML file

Parameters

inFile	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
AdcCfg
    microcontroller.Microcontroller, 42
AdcConfWindow
    gui.AdcConfWindow, 11
addPin
    xmlCreator.ConfXmlWriter, 24
ALT_MODE_ANALOG
    configurator.GPIO.AltMode, 13
ALT_MODE_I2C
    configurator.GPIO.AltMode, 13
ALT_MODE_MAX_VALUE
    configurator.GPIO.AltMode, 13
ALT_MODE_NONE
    configurator.GPIO.AltMode, 13
ALT_MODE_SPI
    configurator.GPIO.AltMode, 13
ALT_MODE_UART
    configurator.GPIO.AltMode, 13

CODE_NAME
    configurator.GPIO.CodeName, 15
CodeGenerator
    framework.CodeGenerator, 14
common, 5
common.ErrorCode, 25
    EX_ERROR, 25
    FILE_CONF_ERROR, 26
    FILE_READ_ERROR, 26
    FILE_WRITE_ERROR, 26
    NO_ERROR, 26
    STR_INVALID, 26
common.Features, 27
    DEBUG, 28
    DEBUG_STR, 28
    debugPrint, 27
    SW_VERSION, 28
    VERBOSE, 28
    VERBOSE_STR, 29
    verbosePrint, 28
    VERSION_NAME, 29
    VERSION_STATUS, 29
configurator, 5
configurator.AdcConf, 10
configurator.ConfigurationFile, 22
    STR_PROJ_CONF_FILE, 23
configurator.GPIO.AltMode, 11
    ALT_MODE_ANALOG, 13
    ALT_MODE_I2C, 13
    ALT_MODE_MAX_VALUE, 13
    ALT_MODE_NONE, 13
    ALT_MODE_SPI, 13
    ALT_MODE_UART, 13
    getConfFromString, 12
    STR_NAME, 13
configurator.GPIO.CodeName, 15
    CODE_NAME, 15
    STR_NAME, 16
configurator.GPIO.Mode, 44
    getConfFromString, 44
    MODE_ALTERNATE_FUNCTION, 45
    MODE_INPUT, 45
    MODE_MAX_VALUE, 45
    MODE_OUTPUT, 45
    STR_NAME, 45
configurator.GPIO.OutLevel, 46
    getConfFromString, 46
    HIGH, 47
    LOW, 47
    MAX_VALUE, 47
    STR_NAME, 47
configurator.GPIO.OutType, 47
    getConfFromString, 48
    OTYPE_MAX_VALUE, 49
    OTYPE_NOT_AVAILABLE, 49
    OTYPE_OPEN_DRAIN, 49
    OTYPE_PUSH_PULL, 49
    STR_NAME, 49
configurator.GPIO.Pull, 79
    getConfFromString, 80
    PULL_DOWN, 80
    PULL_MAX_VALUE, 80
    PULL_NOT_AVAILABLE, 81
    PULL_UP, 81
    STR_NAME, 81
configurator.GPIO.Selected, 81
    getBoolean, 82
    getConfFromBoolean, 82
    getConfFromString, 83
    NOT, 83

STR_NAME, 83
YES, 83
configurator.GPIO.Speed, 84
getConfFromString, 84
SPEED_FAST, 85
SPEED_HIGH, 85
SPEED_MAX_VALUE, 85
SPEED_MEDIUM, 85
SPEED_NOT_AVAILABLE, 85
STR_NAME, 85
configurator.PinConf, 67
DF_ALT_MODE, 75
DF_CODE_NAME, 76
DF_MODE, 76
DF_OUT_LEVEL, 76
DF_OUTTYPE, 76
DF_PULL, 76
DF_SELECTED, 76
DF_SPEED, 76
getAltMode, 69
getCodeName, 69
getMode, 69
getOutLevel, 69
getOutType, 70
getPinName, 70
getPort, 70
getPortPin, 70
getPull, 71
getSelected, 71
getSpeed, 71
isAv_Adc, 71
isAv_altFunc, 72
isAv_I2c, 72
isAv_Spi, 72
isAv_Uart, 72
isValid, 73
PinConf, 68
setAltMode, 73
setCodeName, 73
setMode, 74
setOutLevel, 74
setOutType, 74
setPull, 75
setSelected, 75
setSpeed, 75
ConfXmlWriter
xmlCreator.ConfXmlWriter, 24

DEBUG
common.Features, 28
DEBUG_STR
common.Features, 28
debugPrint
common.Features, 27
DEF_FEATURE
microcontroller.Pin, 66
DEF_FEATURE_AV
microcontroller.Pin, 66
DEF_FUNCTION
microcontroller.Pin, 66
DEF_NAME
microcontroller.Pin, 66
DEF_NUMBER
microcontroller.Pin, 66
DEF_PORT
microcontroller.Pin, 67
Definitions_Common
microcontroller.Microcontroller, 43
Definitions_Gpio
microcontroller.Microcontroller, 43
DF_ALT_MODE
configurator.PinConf, 75
DF_CODE_NAME
configurator.PinConf, 76
DF_MODE
configurator.PinConf, 76
DF_OUT_LEVEL
configurator.PinConf, 76
DF_OUTTYPE
configurator.PinConf, 76
DF_PULL
configurator.PinConf, 76
DF_SELECTED
configurator.PinConf, 76
DF_SPEED
configurator.PinConf, 76
DISABLE
microcontroller.Pin, 67

ENABLE
microcontroller.Pin, 67
EX_ERROR
common.ErrorCode, 25

FILE_CONF_ERROR
common.ErrorCode, 26
FILE_READ_ERROR
common.ErrorCode, 26
FILE_WRITE_ERROR
common.ErrorCode, 26
framework, 6
framework.CodeGenerator, 14
CodeGenerator, 14
Generate, 15
framework.Common, 16
getCfgFileCPath, 17
getCfgFileHPath, 17
getCfgPath, 18
getCommonCfgDefinitions, 18

getCommonIncludes, 19
getFrameworkCommonFilePath, 19
getFrameworkIncludesFilePath, 19
getInstallationFwkPath, 20
getProjectFwkPath, 20
NL, 21
setInstallationFwkPath, 20
setProjectFwkPath, 21
STR_DEFINITION, 21
STR_GEN_CODE_NOTICE_FOOTER, 21
STR_GEN_CODE_NOTICE_HEADER, 21
STR_HEADER_EXT, 22
STR_INCLUDE, 22
STR_MODULE_GPIO, 22
FrmCodeGenerator
gui.MainWindow, 37

Generate
framework.CodeGenerator, 15
generateCode
gui.MainGui, 31
getAdc
microcontroller.Pin, 52
getAltMode
configurator.PinConf, 69
getBoolean
configurator.GPIO.Selected, 82
getCfgFileCPath
framework.Common, 17
getCfgFileHPath
framework.Common, 17
getCfgPath
framework.Common, 18
getClock
microcontroller.Pin, 52
getCodeName
configurator.PinConf, 69
getCommonCfgDefinitions
framework.Common, 18
getCommonIncludes
framework.Common, 19
getConfFile
projectConfiguration.ProjectSettings, 78
getConfFromBoolean
configurator.GPIO.Selected, 82
getConfFromString
configurator.GPIO.AltMode, 12
configurator.GPIO.Mode, 44
configurator.GPIO.OutLevel, 46
configurator.GPIO.OutType, 48
configurator.GPIO.Pull, 80
configurator.GPIO.Selected, 83
configurator.GPIO.Speed, 84
getConfiguredPin
microcontroller.Microcontroller, 39
getElementInfo
xmlParser.XmlOpener, 87
getElementInfoFromDoc
xmlParser.XmlOpener, 87
getFeat_adc
microcontroller.Pin, 52
getFeat_clock
microcontroller.Pin, 53
getFeat_i2c
microcontroller.Pin, 53
getFeat_int
microcontroller.Pin, 53
getFeat_reset
microcontroller.Pin, 53
getFeat_spi
microcontroller.Pin, 54
getFeat_timer
microcontroller.Pin, 54
getFeat_uart
microcontroller.Pin, 54
getFrameworkCommonFilePath
framework.Common, 19
getFrameworkIncludesFilePath
framework.Common, 19
getFrameworkPath
projectConfiguration.ProjectSettings, 78
getFunc_gnd
microcontroller.Pin, 54
getFunc_gpio
microcontroller.Pin, 55
getFunc_misc
microcontroller.Pin, 55
getFunc_reset
microcontroller.Pin, 55
getFunc_vcc
microcontroller.Pin, 55
getI2c
microcontroller.Pin, 56
getInstallationFwkPath
framework.Common, 20
getInt
microcontroller.Pin, 56
getMode
configurator.PinConf, 69
getName
microcontroller.Pin, 56
getNumber
microcontroller.Pin, 56
getOutLevel
configurator.PinConf, 69
getOutType
configurator.PinConf, 70
getParsedDoc

xmlParser.XmlOpener, 87
getPin
microcontroller.Microcontroller, 40
getPinName
configurator.PinConf, 70
getPort
configurator.PinConf, 70
microcontroller.Pin, 57
getPortPin
configurator.PinConf, 70
microcontroller.Pin, 57
getProjectFwkPath
framework.Common, 20
getProjectName
projectConfiguration.ProjectSettings, 78
getPull
configurator.PinConf, 71
getReset
microcontroller.Pin, 57
getSelected
configurator.PinConf, 71
getSpeed
configurator.PinConf, 71
getSpi
microcontroller.Pin, 57
getString
gui.Messages, 37
getTimer
microcontroller.Pin, 58
getUart
microcontroller.Pin, 58
getUc_gpioNum
microcontroller.Microcontroller, 40
getUc_manufacturer
microcontroller.Microcontroller, 40
getUc_model
microcontroller.Microcontroller, 40
getUc_pinNum
microcontroller.Microcontroller, 41
getUc_portNum
microcontroller.Microcontroller, 41
getUc_selectedPinsNum
microcontroller.Microcontroller, 41
getUcFile
projectConfiguration.ProjectSettings, 78
GpioCfgPin
microcontroller.Microcontroller, 43
GpioConfWindow
gui.GpioConfWindow, 30
gui, 6
gui.AboutWindow, 9
AboutWindow, 9
main, 10
gui.AdcConfWindow, 10
AdcConfWindow, 11
main, 11
gui.GpioConfWindow, 29
GpioConfWindow, 30
main, 30
gui.MainGui, 31
generateCode, 31
loadProjectFile, 32
main, 32
ProjectFile, 34
ProjectPath, 34
saveUc, 32
setNewUC, 32
showAboutWindow, 33
showAdcConfWindow, 33
showErrorDialog, 33
showGpioConfWindow, 33
gui.MainWindow, 34
FrmCodeGenerator, 37
main, 35
MainWindow, 35
OpenFileChooser, 35
setProjectInformation, 36
setVisible, 36
gui.Messages, 37
getString, 37

HIGH
configurator.GPIO.OutLevel, 47

Includes_Common
microcontroller.Microcontroller, 43
Includes_Gpio
microcontroller.Microcontroller, 43
isAv_Adc
configurator.PinConf, 71
isAv_altFunc
configurator.PinConf, 72
isAv_I2c
configurator.PinConf, 72
isAv_Spi
configurator.PinConf, 72
isAv_Uart
configurator.PinConf, 72
isValid
configurator.PinConf, 73
microcontroller.Microcontroller, 41
microcontroller.Pin, 58

loadPinsConf
microcontroller.Microcontroller, 42
loadProjectFile
gui.MainGui, 32
LOW
configurator.GPIO.OutLevel, 47

main
gui.AboutWindow, 10
gui.AdcConfWindow, 11
gui.GpioConfWindow, 30
gui.MainGui, 32
gui.MainWindow, 35
MainWindow
gui.MainWindow, 35
MAX_NUMBER_OF_PINS_PER_PORT
microcontroller.Microcontroller, 43
MAX_VALUE
configurator.GPIO.OutLevel, 47
Microcontroller
microcontroller.Microcontroller, 39
microcontroller, 7
microcontroller.Adc, 10
microcontroller.Microcontroller, 38
AdcCfg, 42
Definitions_Common, 43
Definitions_Gpio, 43
getConfiguredPin, 39
getPin, 40
getUc_gpioNum, 40
getUc_manufacturer, 40
getUc_model, 40
getUc_pinNum, 41
getUc_portNum, 41
getUc_selectedPinsNum, 41
GpioCfgPin, 43
Includes_Common, 43
Includes_Gpio, 43
isValid, 41
loadPinsConf, 42
MAX_NUMBER_OF_PINS_PER_PORT, 43
Microcontroller, 39
Ports, 43
processDocument, 42
microcontroller.Pin, 50
DEF_FEATURE, 66
DEF_FEATURE_AV, 66
DEF_FUNCTION, 66
DEF_NAME, 66
DEF_NUMBER, 66
DEF_PORT, 67
DISABLE, 67
ENABLE, 67
getAdc, 52
getClock, 52
getFeat_adc, 52
getFeat_clock, 53
getFeat_i2c, 53
getFeat_int, 53
getFeat_reset, 53
getFeat_spi, 54
getFeat_timer, 54
getFeat_uart, 54
getFunc_gnd, 54
getFunc_gpio, 55
getFunc_misc, 55
getFunc_reset, 55
getFunc_vcc, 55
getI2c, 56
getInt, 56
getName, 56
getNumber, 56
getPort, 57
getPortPin, 57
getReset, 57
getSpi, 57
getTimer, 58
getUart, 58
isValid, 58
Pin, 52
setAdc, 58
setClock, 59
setFeat_adc, 59
setFeat_clock, 59
setFeat_i2c, 60
setFeat_int, 60
setFeat_reset, 60
setFeat_spi, 60
setFeat_timer, 61
setFeat_uart, 61
setFunc_gnd, 61
setFunc_gpio, 62
setFunc_misc, 62
setFunc_reset, 62
setFunc_vcc, 63
setI2c, 63
setInt, 63
setName, 63
setNumber, 64
setPort, 64
setPortPin, 64
setReset, 65
setSpi, 65
setTimer, 65
setUart, 66
MODE_ALTERNATE_FUNCTION
configurator.GPIO.Mode, 45
MODE_INPUT
configurator.GPIO.Mode, 45
MODE_MAX_VALUE
configurator.GPIO.Mode, 45
MODE_OUTPUT
configurator.GPIO.Mode, 45

NL

framework.Common, 21
NO_ERROR
common.ErrorCode, 26
NOT
configurator.GPIO.Selected, 83

OpenFile
xmlParser.XmlOpener, 88
OpenFileChooser
gui.MainWindow, 35
openProjectFile
projectConfiguration.ProjectSettings, 78
OTYPE_MAX_VALUE
configurator.GPIO.OutType, 49
OTYPE_NOT_AVAILABLE
configurator.GPIO.OutType, 49
OTYPE_OPEN_DRAIN
configurator.GPIO.OutType, 49
OTYPE_PUSH_PULL
configurator.GPIO.OutType, 49

Pin
microcontroller.Pin, 52
PinConf
configurator.PinConf, 68
Ports
microcontroller.Microcontroller, 43
processDocument
microcontroller.Microcontroller, 42
projectConfiguration.ProjectSettings, 79
projectConfiguration, 7
projectConfiguration.ProjectSettings, 77
getConfFile, 78
getFrameworkPath, 78
getProjectName, 78
getUcFile, 78
openProjectFile, 78
processDocument, 79
ProjectSettings, 77
ProjectFile
gui.MainGui, 34
ProjectPath
gui.MainGui, 34
ProjectSettings
projectConfiguration.ProjectSettings, 77
PULL_DOWN
configurator.GPIO.Pull, 80
PULL_MAX_VALUE
configurator.GPIO.Pull, 80
PULL_NOT_AVAILABLE
configurator.GPIO.Pull, 81
PULL_UP
configurator.GPIO.Pull, 81

saveUc
gui.MainGui, 32
setAdc
microcontroller.Pin, 58
setAltMode
configurator.PinConf, 73
setClock
microcontroller.Pin, 59
setCodeName
configurator.PinConf, 73
setFeat_adc
microcontroller.Pin, 59
setFeat_clock
microcontroller.Pin, 59
setFeat_i2c
microcontroller.Pin, 60
setFeat_int
microcontroller.Pin, 60
setFeat_reset
microcontroller.Pin, 60
setFeat_spi
microcontroller.Pin, 60
setFeat_timer
microcontroller.Pin, 61
setFeat_uart
microcontroller.Pin, 61
setFunc_gnd
microcontroller.Pin, 61
setFunc_gpio
microcontroller.Pin, 62
setFunc_misc
microcontroller.Pin, 62
setFunc_reset
microcontroller.Pin, 62
setFunc_vcc
microcontroller.Pin, 63
setI2c
microcontroller.Pin, 63
setInstallationFwkPath
framework.Common, 20
setInt
microcontroller.Pin, 63
setMode
configurator.PinConf, 74
setName
microcontroller.Pin, 63
setNewUC
gui.MainGui, 32
setNumber
microcontroller.Pin, 64
setOutLevel
configurator.PinConf, 74
setOutType
configurator.PinConf, 74
setPort

microcontroller.Pin, 64
setPortPin
microcontroller.Pin, 64
setProjectFwkPath
framework.Common, 21
setProjectInformation
gui.MainWindow, 36
setPull
configurator.PinConf, 75
setReset
microcontroller.Pin, 65
setSelected
configurator.PinConf, 75
setSpeed
configurator.PinConf, 75
setSpi
microcontroller.Pin, 65
setTimer
microcontroller.Pin, 65
setUart
microcontroller.Pin, 66
setVisible
gui.MainWindow, 36
showAboutWindow
gui.MainGui, 33
showAdcConfWindow
gui.MainGui, 33
showErrorDialog
gui.MainGui, 33
showGpioConfWindow
gui.MainGui, 33
SPEED_FAST
configurator.GPIO.Speed, 85
SPEED_HIGH
configurator.GPIO.Speed, 85
SPEED_MAX_VALUE
configurator.GPIO.Speed, 85
SPEED_MEDIUM
configurator.GPIO.Speed, 85
SPEED_NOT_AVAILABLE
configurator.GPIO.Speed, 85
STR_DEFINITION
framework.Common, 21
STR_GEN_CODE_NOTICE_FOOTER
framework.Common, 21
STR_GEN_CODE_NOTICE_HEADER
framework.Common, 21
STR_HEADER_EXT
framework.Common, 22
STR_INCLUDE
framework.Common, 22
STR_INVALID
common.ErrorCode, 26
STR_MODULE_GPIO
framework.Common, 22
STR_NAME
configurator.GPIO.AltMode, 13
configurator.GPIO.CodeName, 16
configurator.GPIO.Mode, 45
configurator.GPIO.OutLevel, 47
configurator.GPIO.OutType, 49
configurator.GPIO.Pull, 81
configurator.GPIO.Selected, 83
configurator.GPIO.Speed, 85
STR_PROJ_CONF_FILE
configurator.ConfigurationFile, 23
SW_VERSION
common.Features, 28

VERBOSE
common.Features, 28
VERBOSE_STR
common.Features, 29
verbosePrint
common.Features, 28
VERSION_NAME
common.Features, 29
VERSION_STATUS
common.Features, 29

writeXml
xmlCreator.ConfXmlWriter, 24

xmlCreator, 8
xmlCreator.ConfXmlWriter, 23
addPin, 24
ConfXmlWriter, 24
writeXml, 24
XmlOpener
xmlParser.XmlOpener, 86
xmlParser, 8
xmlParser.XmlOpener, 86
getElementInfo, 87
getElementInfoFromDoc, 87
getParsedDoc, 87
OpenFile, 88
XmlOpener, 86

YES
configurator.GPIO.Selected, 83