

Code\_generator

Generated by Doxygen 1.8.12



# 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 microcontroller . . . . .	6
3.4.1	Detailed Description . . . . .	6
3.5	Package xmlCreator . . . . .	7
3.5.1	Detailed Description . . . . .	7
3.6	Package xmlParser . . . . .	7
3.6.1	Detailed Description . . . . .	7

<b>4</b>	<b>Class Documentation</b>	<b>9</b>
4.1	xmlCreator.ConfXmlWriter Class Reference	9
4.1.1	Detailed Description	9
4.1.2	Constructor & Destructor Documentation	9
4.1.2.1	ConfXmlWriter()	9
4.1.3	Member Function Documentation	10
4.1.3.1	addPin()	10
4.1.3.2	writeXml()	10
4.2	common.ErrorCode Enum Reference	10
4.2.1	Detailed Description	11
4.2.2	Member Data Documentation	11
4.2.2.1	NO_ERROR	11
4.3	common.Features Class Reference	11
4.3.1	Detailed Description	11
4.3.2	Member Function Documentation	12
4.3.2.1	debugPrint()	12
4.3.2.2	verbosePrint()	12
4.3.3	Member Data Documentation	12
4.3.3.1	DEBUG	12
4.3.3.2	DEBUG_STR	12
4.3.3.3	VERBOSE	12
4.3.3.4	VERBOSE_STR	13
4.4	microcontroller.Microcontroller Class Reference	13
4.4.1	Detailed Description	13
4.4.2	Constructor & Destructor Documentation	13
4.4.2.1	Microcontroller()	13
4.4.3	Member Function Documentation	14
4.4.3.1	getPin()	14

4.4.3.2	<a href="#">getUc_gpioNum()</a>	14
4.4.3.3	<a href="#">getUc_manufacturer()</a>	14
4.4.3.4	<a href="#">getUc_model()</a>	15
4.4.3.5	<a href="#">getUc_pinNum()</a>	15
4.4.3.6	<a href="#">processDocument()</a>	15
4.5	<a href="#">configurator.GPIO.Mode Enum Reference</a>	15
4.5.1	<a href="#">Detailed Description</a>	16
4.6	<a href="#">configurator.GPIO.OutType Enum Reference</a>	16
4.6.1	<a href="#">Detailed Description</a>	16
4.7	<a href="#">microcontroller.Pin Class Reference</a>	17
4.7.1	<a href="#">Detailed Description</a>	18
4.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	19
4.7.2.1	<a href="#">Pin()</a>	19
4.7.3	<a href="#">Member Function Documentation</a>	19
4.7.3.1	<a href="#">getAdc()</a>	19
4.7.3.2	<a href="#">getClock()</a>	19
4.7.3.3	<a href="#">getFeat_adc()</a>	19
4.7.3.4	<a href="#">getFeat_clock()</a>	20
4.7.3.5	<a href="#">getFeat_i2c()</a>	20
4.7.3.6	<a href="#">getFeat_int()</a>	20
4.7.3.7	<a href="#">getFeat_reset()</a>	20
4.7.3.8	<a href="#">getFeat_spi()</a>	21
4.7.3.9	<a href="#">getFeat_timer()</a>	21
4.7.3.10	<a href="#">getFeat_uart()</a>	21
4.7.3.11	<a href="#">getFunc_gnd()</a>	21
4.7.3.12	<a href="#">getFunc_gpio()</a>	22
4.7.3.13	<a href="#">getFunc_vcc()</a>	22
4.7.3.14	<a href="#">getI2c()</a>	22

4.7.3.15	<a href="#">getInt()</a>	22
4.7.3.16	<a href="#">getName()</a>	23
4.7.3.17	<a href="#">getNumber()</a>	23
4.7.3.18	<a href="#">getPort()</a>	23
4.7.3.19	<a href="#">getReset()</a>	23
4.7.3.20	<a href="#">getSpi()</a>	24
4.7.3.21	<a href="#">getTimer()</a>	24
4.7.3.22	<a href="#">getUart()</a>	24
4.7.3.23	<a href="#">isValid()</a>	24
4.7.3.24	<a href="#">setAdc()</a>	24
4.7.3.25	<a href="#">setClock()</a>	25
4.7.3.26	<a href="#">setFeat_adc()</a>	25
4.7.3.27	<a href="#">setFeat_clock()</a>	25
4.7.3.28	<a href="#">setFeat_i2c()</a>	25
4.7.3.29	<a href="#">setFeat_int()</a>	27
4.7.3.30	<a href="#">setFeat_reset()</a>	27
4.7.3.31	<a href="#">setFeat_spi()</a>	27
4.7.3.32	<a href="#">setFeat_timer()</a>	27
4.7.3.33	<a href="#">setFeat_uart()</a>	29
4.7.3.34	<a href="#">setFunc_gnd()</a>	29
4.7.3.35	<a href="#">setFunc_gpio()</a>	29
4.7.3.36	<a href="#">setFunc_vcc()</a>	29
4.7.3.37	<a href="#">setI2c()</a>	31
4.7.3.38	<a href="#">setInt()</a>	31
4.7.3.39	<a href="#">setName()</a>	31
4.7.3.40	<a href="#">setNumber()</a>	31
4.7.3.41	<a href="#">setPort()</a>	33
4.7.3.42	<a href="#">setReset()</a>	33

4.7.3.43	setSpi()	33
4.7.3.44	setTimer()	33
4.7.3.45	setUart()	35
4.7.4	Member Data Documentation	35
4.7.4.1	DEF_FEATURE	35
4.7.4.2	DEF_FEATURE_AV	35
4.7.4.3	DEF_FUNCTION	35
4.7.4.4	DEF_NAME	35
4.7.4.5	DEF_NUMBER	36
4.7.4.6	DEF_PORT	36
4.7.4.7	DISABLE	36
4.7.4.8	ENABLE	36
4.8	configurator.PinConf Class Reference	36
4.8.1	Detailed Description	37
4.8.2	Constructor & Destructor Documentation	37
4.8.2.1	PinConf()	37
4.8.3	Member Function Documentation	37
4.8.3.1	getMode()	37
4.8.3.2	getOutType()	38
4.8.3.3	getPin()	38
4.8.3.4	getPort()	38
4.8.3.5	getPull()	38
4.8.3.6	getSpeed()	39
4.8.3.7	isAv_Adc()	39
4.8.3.8	isAv_altFunc()	39
4.8.3.9	isValid()	39
4.8.3.10	setMode()	39
4.8.3.11	setOutType()	40

4.8.3.12	setPull()	40
4.8.3.13	setSpeed()	40
4.8.4	Member Data Documentation	40
4.8.4.1	DF_MODE	40
4.8.4.2	DF_OUTTYPE	41
4.8.4.3	DF_PULL	41
4.8.4.4	DF_SPEED	41
4.9	configurator.GPIO.Pull Enum Reference	41
4.9.1	Detailed Description	41
4.10	configurator.GPIO.Speed Enum Reference	42
4.10.1	Detailed Description	42
4.11	xmlParser.TestMain Class Reference	42
4.11.1	Detailed Description	42
4.11.2	Member Function Documentation	42
4.11.2.1	main()	42
4.12	xmlParser.XmlOpener Class Reference	43
4.12.1	Detailed Description	43
4.12.2	Constructor & Destructor Documentation	43
4.12.2.1	XmlOpener()	43
4.12.3	Member Function Documentation	43
4.12.3.1	getParsedDoc()	43
4.12.3.2	OpenFile()	44



# 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="#">microcontroller</a>	6
<a href="#">xmlCreator</a>	7
<a href="#">xmlParser</a>	7



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">xmlCreator.ConfXmlWriter</a>	9
<a href="#">common.ErrorCode</a>	10
<a href="#">common.Features</a>	11
<a href="#">microcontroller.Microcontroller</a>	13
<a href="#">configurator.GPIO.Mode</a>	15
<a href="#">configurator.GPIO.OutType</a>	16
<a href="#">microcontroller.Pin</a>	17
<a href="#">configurator.PinConf</a>	36
<a href="#">configurator.GPIO.Pull</a>	41
<a href="#">configurator.GPIO.Speed</a>	42
<a href="#">xmlParser.TestMain</a>	42
<a href="#">xmlParser.XmlOpener</a>	43



## 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 [PinConf](#)

### 3.2.1 Detailed Description

Configuration classes

**Author**

Miguel Diaz

**Version**

0.1

## 3.3 Package framework

### 3.3.1 Detailed Description

Framework information

**Author**

H112943

**Version**

0.1

## 3.4 Package microcontroller

**Classes**

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

### 3.4.1 Detailed Description

[Microcontroller](#) related classes

**Author**

Miguel Diaz

**Version**

0.1

## 3.5 Package xmlCreator

### Classes

- class [ConfXmlWriter](#)

#### 3.5.1 Detailed Description

Create configuration XML

#### Author

Miguel Diaz

#### Version

0.1

## 3.6 Package xmlParser

### Classes

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

#### 3.6.1 Detailed Description

XML parser for microcontroller information and project settings

#### Author

Miguel Diaz

#### Version

0.1





## Chapter 4

# Class Documentation

### 4.1 xmlCreator.ConfXmlWriter Class Reference

#### Public Member Functions

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

#### 4.1.1 Detailed Description

Write a XML file

#### Author

Miguel Diaz

#### Version

0.1

#### 4.1.2 Constructor & Destructor Documentation

##### 4.1.2.1 ConfXmlWriter()

```
xmlCreator.ConfXmlWriter.ConfXmlWriter (
    int pins )
```

Create configuration file

**Parameters**

<i>pins</i>	Total number of GPIO pins to save
-------------	-----------------------------------

### 4.1.3 Member Function Documentation

#### 4.1.3.1 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

#### 4.1.3.2 writeXml()

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

Write the XML file

**Parameters**

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

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

- src/xmlCreator/ConfXmlWriter.java

## 4.2 common.ErrorCode Enum Reference

**Public Attributes**

- [NO\\_ERROR](#)

### 4.2.1 Detailed Description

Error codes enum

**Author**

Miguel Diaz

**Version**

0.1

### 4.2.2 Member Data Documentation

#### 4.2.2.1 NO\_ERROR

```
common.ErrorCode.NO_ERROR
```

No error message

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

- src/common/ErrorCode.java

## 4.3 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](#) = false
- static final boolean [VERBOSE](#) = true
- static final String [VERBOSE\\_STR](#) = "# "
- static final String [DEBUG\\_STR](#) = "\$ "

### 4.3.1 Detailed Description

Class that includes all project features

**Author**

Miguel Diaz

**Version**

0.1

## 4.3.2 Member Function Documentation

### 4.3.2.1 debugPrint()

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

Print Debug message to console

#### Parameters

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

### 4.3.2.2 verbosePrint()

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

Print Verbose message to console

#### Parameters

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

## 4.3.3 Member Data Documentation

### 4.3.3.1 DEBUG

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

Enables debug functions

### 4.3.3.2 DEBUG\_STR

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

Debug messages indicator on system console

### 4.3.3.3 VERBOSE

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

Enables console messages

#### 4.3.3.4 VERBOSE\_STR

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

Verbose messages indicator on system console

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

- src/common/Features.java

## 4.4 microcontroller.Microcontroller Class Reference

### Public Member Functions

- [Microcontroller](#) (Document ucDoc)
- [ErrorCode processDocument](#) ()
- [Pin getPin](#) (int pinNum)
- [String getUc\\_model](#) ()
- [String getUc\\_manufacturer](#) ()
- [int getUc\\_pinNum](#) ()
- [int getUc\\_gpioNum](#) ()

### 4.4.1 Detailed Description

[Microcontroller](#) related methods

#### Author

Miguel Diaz

#### Version

0.1

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 Microcontroller()

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

Constructor

**Parameters**

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

### 4.4.3 Member Function Documentation

#### 4.4.3.1 `getPin()`

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

Get a pin's characteristics

**Parameters**

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

**Returns**

[Pin's](#) characteristics

#### 4.4.3.2 `getUc_gpioNum()`

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

Get the number of GPIOs in the microcontroller

**Returns**

Number of GPIOs

#### 4.4.3.3 `getUc_manufacturer()`

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

Get the microcontroller's manufacturer

**Returns**

[Microcontroller's](#) manufacturer

#### 4.4.3.4 getUc\_model()

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

Get the microcontroller's model

##### Returns

[Microcontroller](#)'s model

#### 4.4.3.5 getUc\_pinNum()

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

Get the microcontroller's pins number

##### Returns

Number of pins

#### 4.4.3.6 processDocument()

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

Process the document obtained from XML file

##### Returns

Error status

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

- src/microcontroller/Microcontroller.java

## 4.5 configurator.GPIO.Mode Enum Reference

### Public Attributes

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

### 4.5.1 Detailed Description

GPIO modes

#### Author

Miguel Diaz

#### Version

0.1

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

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

## 4.6 configurator.GPIO.OutType Enum Reference

### Public Attributes

- **OTYPE\_PUSH\_PULL**
- **OTYPE\_OPEN\_DRAIN**
- **OTYPE\_NOT\_AVAILABLE**

### 4.6.1 Detailed Description

Pin's output type

#### Author

Miguel Diaz

#### Version

0.1

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

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



## 4.7 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](#) ()



## 4.7.2 Constructor & Destructor Documentation

### 4.7.2.1 Pin()

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

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

## 4.7.3 Member Function Documentation

### 4.7.3.1 getAdc()

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

Get the pin's ADC name

#### Returns

Pin's ADC

### 4.7.3.2 getClock()

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

Get the pin's clock name

#### Returns

Pin's clock

### 4.7.3.3 getFeat\_adc()

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

See if the pin has an ADC

#### Returns

Feature availability

#### 4.7.3.4 getFeat\_clock()

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

See if the pin supports a clock

##### Returns

Feature availability

#### 4.7.3.5 getFeat\_i2c()

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

See if the pin has I2C

##### Returns

Feature availability

#### 4.7.3.6 getFeat\_int()

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

See if the pin has an interruption

##### Returns

Feature availability

#### 4.7.3.7 getFeat\_reset()

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

See if the pin has a reset feature

##### Returns

Feature availability

#### 4.7.3.8 getFeat\_spi()

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

See if the pin has SPI

##### Returns

Feature availability

#### 4.7.3.9 getFeat\_timer()

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

See if the pin supports a timer

##### Returns

Feature availability

#### 4.7.3.10 getFeat\_uart()

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

See if the pin has a UART

##### Returns

Feature availability

#### 4.7.3.11 getFunc\_gnd()

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

See if the pin is GND

##### Returns

Function availability

#### 4.7.3.12 `getFunc_gpio()`

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

See if the pin is GPIO

##### Returns

Function availability

#### 4.7.3.13 `getFunc_vcc()`

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

See if the pin is Vcc

##### Returns

Function availability

#### 4.7.3.14 `getI2c()`

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

Get the pin's I2C name

##### Returns

[Pin's I2C](#)

#### 4.7.3.15 `getInt()`

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

Get the pin's interruption name

##### Returns

[Pin's interruption](#)

#### 4.7.3.16 getName()

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

Get the pin's name

##### Returns

Pin's name

#### 4.7.3.17 getNumber()

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

Get the pin's number

##### Returns

Pin's number

#### 4.7.3.18 getPort()

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

Get the pin's port

##### Returns

Pin's port

#### 4.7.3.19 getReset()

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

Get the pin's reset name

##### Returns

Pin's reset

**4.7.3.20 getSpi()**

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

Get the pin's SPI name

**Returns**

Pin's SPI

**4.7.3.21 getTimer()**

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

Get the pin's timer name

**Returns**

Pin's timer

**4.7.3.22 getUart()**

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

Get the pin's UART name

**Returns**

Pin's UART

**4.7.3.23 isValid()**

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

Check if the pin is correctly initialized

**Returns**

True if the pin is correctly initialized

**4.7.3.24 setAdc()**

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

Set the pin's ADC



## Parameters

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

## 4.7.3.25 setClock()

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

Set the pin's clock

## Parameters

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

## 4.7.3.26 setFeat\_adc()

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

Set the pin's ADC feature

## Parameters

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

## 4.7.3.27 setFeat\_clock()

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

Set the pin's Clock feature

## Parameters

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

## 4.7.3.28 setFeat\_i2c()

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

Set the pin's I2C feature

## Parameters

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

## 4.7.3.29 setFeat\_int()

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

Set the pin's interruption feature

## Parameters

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

## 4.7.3.30 setFeat\_reset()

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

Set the pin's reset feature

## Parameters

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

## 4.7.3.31 setFeat\_spi()

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

Set the pin's SPI feature

## Parameters

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

## 4.7.3.32 setFeat\_timer()

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

Set the pin's timer feature

## Parameters

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

## 4.7.3.33 setFeat\_uart()

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

Set the pin's UART feature

## Parameters

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

## 4.7.3.34 setFunc\_gnd()

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

Set the pin to GND status

## Parameters

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

## 4.7.3.35 setFunc\_gpio()

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

Set the pin to GPIO status

## Parameters

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

## 4.7.3.36 setFunc\_vcc()

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

Set the pin to Vcc status

## Parameters

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

## 4.7.3.37 setI2c()

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

Set the pin's I2C

## Parameters

<i>feature</i>	<a href="#">Pin's I2C</a>
----------------	---------------------------

## 4.7.3.38 setInt()

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

Set the pin's interruption

## Parameters

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

## 4.7.3.39 setName()

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

Set the pin's name

## Parameters

<i>pinName</i>	<a href="#">Pin's name</a>
----------------	----------------------------

## 4.7.3.40 setNumber()

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

Set the pin's number



## Parameters

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

## 4.7.3.41 setPort()

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

Set the pin's port

## Parameters

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

## 4.7.3.42 setReset()

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

Set the pin's reset

## Parameters

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

## 4.7.3.43 setSpi()

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

Set the pin's SPI

## Parameters

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

## 4.7.3.44 setTimer()

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

Set the pin's timer

**Parameters**

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

**4.7.3.45 setUart()**

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

Set the pin's UART

**Parameters**

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

**4.7.4 Member Data Documentation****4.7.4.1 DEF\_FEATURE**

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

Default value for pin's feature as not available

**4.7.4.2 DEF\_FEATURE\_AV**

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

Default value for pin's feature availability as not available

**4.7.4.3 DEF\_FUNCTION**

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

Default value for pin's function as not enabled

**4.7.4.4 DEF\_NAME**

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

Default value for pin's name

#### 4.7.4.5 DEF\_NUMBER

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

Default value for pin's number

#### 4.7.4.6 DEF\_PORT

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

Default value for pin's port

#### 4.7.4.7 DISABLE

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

Disable value for features and functions

#### 4.7.4.8 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`

## 4.8 configurator.PinConf Class Reference

### Public Member Functions

- [PinConf](#) ([Pin](#) gpioPin)
- boolean [isValid](#) ()
- String [getPort](#) ()
- String [getPin](#) ()
- [Mode](#) [getMode](#) ()
- void [setMode](#) ([Mode](#) mode)
- [OutType](#) [getOutType](#) ()
- void [setOutType](#) ([OutType](#) outType)
- [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 [Pull DF\\_PULL](#) = Pull.PULL\_NOT\_AVAILABLE

### 4.8.1 Detailed Description

GPIO pin configuration

#### Author

Miguel Diaz

#### Version

0.1

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 PinConf()

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

Constructor

#### Parameters

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

### 4.8.3 Member Function Documentation

#### 4.8.3.1 getMode()

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

Get the pin's mode configuration

#### Returns

Mode

#### 4.8.3.2 getOutType()

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

Get the pin's output configuration

##### Returns

Output configuration

#### 4.8.3.3 getPin()

```
String configurator.PinConf.getPin ( )
```

Get the pin's number

##### Returns

Pin's number

#### 4.8.3.4 getPort()

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

Get the pin's port

##### Returns

Port

#### 4.8.3.5 getPull()

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

Get the pin's pull resistor configuration

##### Returns

Pull Resistor configuration

#### 4.8.3.6 getSpeed()

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

Get the pin's speed

##### Returns

Speed

#### 4.8.3.7 isAv\_Adc()

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

Check availability of ADC

##### Returns

True if ADC is available

#### 4.8.3.8 isAv\_altFunc()

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

Check the availability of alternate function

##### Returns

True if alternate function is available

#### 4.8.3.9 isValid()

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

Check if the GPIO pin is valid

##### Returns

True if valid

#### 4.8.3.10 setMode()

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

Set the pin's mode configuration

**Parameters**

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

**4.8.3.11 setOutType()**

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

Set the pin's output configuration

**Parameters**

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

**4.8.3.12 setPull()**

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

Set the pull resistor configuration

**Parameters**

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

**4.8.3.13 setSpeed()**

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

Set the pin's speed

**Parameters**

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

**4.8.4 Member Data Documentation****4.8.4.1 DF\_MODE**

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



Default Pin mode

#### 4.8.4.2 DF\_OUTTYPE

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

Default pin's output type

#### 4.8.4.3 DF\_PULL

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

Default pin's pull resistor

#### 4.8.4.4 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

## 4.9 configurator.GPIO.Pull Enum Reference

### Public Attributes

- **PULL\_UP**
- **PULL\_DOWN**
- **PULL\_NOT\_AVAILABLE**

### 4.9.1 Detailed Description

Pin's pull resistor

#### Author

Miguel Diaz

#### Version

0.1

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

- src/configurator/GPIO/Pull.java

## 4.10 configurator.GPIO.Speed Enum Reference

### Public Attributes

- **SPEED\_FAST**
- **SPEED\_MEDIUM**
- **SPEED\_HIGH**
- **SPEED\_NOT\_AVAILABLE**

### 4.10.1 Detailed Description

Pin's speed

#### Author

Miguel Diaz

#### Version

0.1

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

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

## 4.11 xmlParser.TestMain Class Reference

### Static Public Member Functions

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

### 4.11.1 Detailed Description

Dummy main class for testing the other classes

#### Author

Miguel Diaz

### 4.11.2 Member Function Documentation

#### 4.11.2.1 `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

## 4.12 xmlParser.XmlOpener Class Reference

### Public Member Functions

- [XmlOpener](#) ()
- [ErrorCode OpenFile](#) (String fileName)
- Document [getParsedDoc](#) ()

### 4.12.1 Detailed Description

Open and process XML files

#### Author

H112943

#### Version

0.1

### 4.12.2 Constructor & Destructor Documentation

#### 4.12.2.1 XmlOpener()

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

Constructor

### 4.12.3 Member Function Documentation

#### 4.12.3.1 getParsedDoc()

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

Get the parsed document AFTER opening the file

#### Returns

Parsed document

#### 4.12.3.2 OpenFile()

```
ErrorCode xmlParser.XmlOpener.OpenFile (  
    String fileName )
```

Open the XML file

**Parameters**

<i>fileName</i>	Complete path and name of XML file
-----------------	------------------------------------

**Returns**

Error code

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

- src/xmlParser/XmlOpener.java



# Index

- addPin
  - xmlCreator::ConfXmlWriter, [10](#)
- common, [5](#)
- common.ErrorCode, [10](#)
- common.Features, [11](#)
- common::ErrorCode
  - NO\_ERROR, [11](#)
- common::Features
  - DEBUG\_STR, [12](#)
  - DEBUG, [12](#)
  - debugPrint, [12](#)
  - VERBOSE\_STR, [12](#)
  - VERBOSE, [12](#)
  - verbosePrint, [12](#)
- ConfXmlWriter
  - xmlCreator::ConfXmlWriter, [9](#)
- configurator, [5](#)
- configurator.GPIO.Mode, [15](#)
- configurator.GPIO.OutType, [16](#)
- configurator.GPIO.Pull, [41](#)
- configurator.GPIO.Speed, [42](#)
- configurator.PinConf, [36](#)
- configurator::PinConf
  - DF\_MODE, [40](#)
  - DF\_OUTTYPE, [41](#)
  - DF\_PULL, [41](#)
  - DF\_SPEED, [41](#)
  - getMode, [37](#)
  - getOutType, [37](#)
  - getPin, [38](#)
  - getPort, [38](#)
  - getPull, [38](#)
  - getSpeed, [38](#)
  - isAv\_Adc, [39](#)
  - isAv\_altFunc, [39](#)
  - isValid, [39](#)
  - PinConf, [37](#)
  - setMode, [39](#)
  - setOutType, [40](#)
  - setPull, [40](#)
  - setSpeed, [40](#)
- DEBUG\_STR
  - common::Features, [12](#)
- DEBUG
  - common::Features, [12](#)
- DEF\_FEATURE\_AV
  - microcontroller::Pin, [35](#)
- DEF\_FEATURE
  - microcontroller::Pin, [35](#)
- DEF\_FUNCTION
  - microcontroller::Pin, [35](#)
- DEF\_NAME
  - microcontroller::Pin, [35](#)
- DEF\_NUMBER
  - microcontroller::Pin, [35](#)
- DEF\_PORT
  - microcontroller::Pin, [36](#)
- DF\_MODE
  - configurator::PinConf, [40](#)
- DF\_OUTTYPE
  - configurator::PinConf, [41](#)
- DF\_PULL
  - configurator::PinConf, [41](#)
- DF\_SPEED
  - configurator::PinConf, [41](#)
- DISABLE
  - microcontroller::Pin, [36](#)
- debugPrint
  - common::Features, [12](#)
- ENABLE
  - microcontroller::Pin, [36](#)
- framework, [6](#)
- getAdc
  - microcontroller::Pin, [19](#)
- getClock
  - microcontroller::Pin, [19](#)
- getFeat\_adc
  - microcontroller::Pin, [19](#)
- getFeat\_clock
  - microcontroller::Pin, [19](#)
- getFeat\_i2c
  - microcontroller::Pin, [20](#)
- getFeat\_int
  - microcontroller::Pin, [20](#)
- getFeat\_reset
  - microcontroller::Pin, [20](#)
- getFeat\_spi

- microcontroller::Pin, 20
- getFeat\_timer
  - microcontroller::Pin, 21
- getFeat\_uart
  - microcontroller::Pin, 21
- getFunc\_gnd
  - microcontroller::Pin, 21
- getFunc\_gpio
  - microcontroller::Pin, 21
- getFunc\_vcc
  - microcontroller::Pin, 22
- getI2c
  - microcontroller::Pin, 22
- getInt
  - microcontroller::Pin, 22
- getMode
  - configurator::PinConf, 37
- getName
  - microcontroller::Pin, 22
- getNumber
  - microcontroller::Pin, 23
- getOutType
  - configurator::PinConf, 37
- getParsedDoc
  - xmlParser::XmlOpener, 43
- getPin
  - configurator::PinConf, 38
  - microcontroller::Microcontroller, 14
- getPort
  - configurator::PinConf, 38
  - microcontroller::Pin, 23
- getPull
  - configurator::PinConf, 38
- getReset
  - microcontroller::Pin, 23
- getSpeed
  - configurator::PinConf, 38
- getSpi
  - microcontroller::Pin, 23
- getTimer
  - microcontroller::Pin, 24
- getUart
  - microcontroller::Pin, 24
- getUc\_gpioNum
  - microcontroller::Microcontroller, 14
- getUc\_manufacturer
  - microcontroller::Microcontroller, 14
- getUc\_model
  - microcontroller::Microcontroller, 14
- getUc\_pinNum
  - microcontroller::Microcontroller, 15
- isAv\_Adc
  - configurator::PinConf, 39
- isAv\_altFunc
  - configurator::PinConf, 39
- isValid
  - configurator::PinConf, 39
  - microcontroller::Pin, 24
- main
  - xmlParser::TestMain, 42
- Microcontroller
  - microcontroller::Microcontroller, 13
- microcontroller, 6
- microcontroller.Microcontroller, 13
- microcontroller.Pin, 17
- microcontroller::Microcontroller
  - getPin, 14
  - getUc\_gpioNum, 14
  - getUc\_manufacturer, 14
  - getUc\_model, 14
  - getUc\_pinNum, 15
  - Microcontroller, 13
  - processDocument, 15
- microcontroller::Pin
  - DEF\_FEATURE\_AV, 35
  - DEF\_FEATURE, 35
  - DEF\_FUNCTION, 35
  - DEF\_NAME, 35
  - DEF\_NUMBER, 35
  - DEF\_PORT, 36
  - DISABLE, 36
  - ENABLE, 36
  - getAdc, 19
  - getClock, 19
  - getFeat\_adc, 19
  - getFeat\_clock, 19
  - getFeat\_i2c, 20
  - getFeat\_int, 20
  - getFeat\_reset, 20
  - getFeat\_spi, 20
  - getFeat\_timer, 21
  - getFeat\_uart, 21
  - getFunc\_gnd, 21
  - getFunc\_gpio, 21
  - getFunc\_vcc, 22
  - getI2c, 22
  - getInt, 22
  - getName, 22
  - getNumber, 23
  - getPort, 23
  - getReset, 23
  - getSpi, 23
  - getTimer, 24
  - getUart, 24
  - isValid, 24
  - Pin, 19



- setAdc, [24](#)
- setClock, [25](#)
- setFeat\_adc, [25](#)
- setFeat\_clock, [25](#)
- setFeat\_i2c, [25](#)
- setFeat\_int, [27](#)
- setFeat\_reset, [27](#)
- setFeat\_spi, [27](#)
- setFeat\_timer, [27](#)
- setFeat\_uart, [29](#)
- setFunc\_gnd, [29](#)
- setFunc\_gpio, [29](#)
- setFunc\_vcc, [29](#)
- setI2c, [31](#)
- setInt, [31](#)
- setName, [31](#)
- setNumber, [31](#)
- setPort, [33](#)
- setReset, [33](#)
- setSpi, [33](#)
- setTimer, [33](#)
- setUart, [35](#)
- NO\_ERROR
  - common::ErrorCode, [11](#)
- OpenFile
  - xmlParser::XmlOpener, [43](#)
- Pin
  - microcontroller::Pin, [19](#)
- PinConf
  - configurator::PinConf, [37](#)
- processDocument
  - microcontroller::Microcontroller, [15](#)
- setAdc
  - microcontroller::Pin, [24](#)
- setClock
  - microcontroller::Pin, [25](#)
- setFeat\_adc
  - microcontroller::Pin, [25](#)
- setFeat\_clock
  - microcontroller::Pin, [25](#)
- setFeat\_i2c
  - microcontroller::Pin, [25](#)
- setFeat\_int
  - microcontroller::Pin, [27](#)
- setFeat\_reset
  - microcontroller::Pin, [27](#)
- setFeat\_spi
  - microcontroller::Pin, [27](#)
- setFeat\_timer
  - microcontroller::Pin, [27](#)
- setFeat\_uart
  - microcontroller::Pin, [29](#)
- setFunc\_gnd
  - microcontroller::Pin, [29](#)
- setFunc\_gpio
  - microcontroller::Pin, [29](#)
- setFunc\_vcc
  - microcontroller::Pin, [29](#)
- setI2c
  - microcontroller::Pin, [31](#)
- setInt
  - microcontroller::Pin, [31](#)
- setMode
  - configurator::PinConf, [39](#)
- setName
  - microcontroller::Pin, [31](#)
- setNumber
  - microcontroller::Pin, [31](#)
- setOutType
  - configurator::PinConf, [40](#)
- setPort
  - microcontroller::Pin, [33](#)
- setPull
  - configurator::PinConf, [40](#)
- setReset
  - microcontroller::Pin, [33](#)
- setSpeed
  - configurator::PinConf, [40](#)
- setSpi
  - microcontroller::Pin, [33](#)
- setTimer
  - microcontroller::Pin, [33](#)
- setUart
  - microcontroller::Pin, [35](#)
- VERBOSE\_STR
  - common::Features, [12](#)
- VERBOSE
  - common::Features, [12](#)
- verbosePrint
  - common::Features, [12](#)
- writeXml
  - xmlCreator::ConfXmlWriter, [10](#)
- xmlCreator, [7](#)
- xmlCreator.ConfXmlWriter, [9](#)
- xmlCreator::ConfXmlWriter
  - addPin, [10](#)
  - ConfXmlWriter, [9](#)
  - writeXml, [10](#)
- XmlOpener
  - xmlParser::XmlOpener, [43](#)
- xmlParser, [7](#)
- xmlParser.TestMain, [42](#)
- xmlParser.XmlOpener, [43](#)

xmlParser::TestMain  
  main, [42](#)  
xmlParser::XmlOpener  
  getParsedDoc, [43](#)  
  OpenFile, [43](#)  
  XmlOpener, [43](#)