

Code_generator

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Packages	1
2	Class Index	3
2.1	Class List	3
3	Namespace Documentation	5
3.1	Package common	5
3.1.1	Detailed Description	5
3.2	Package configurator	5
3.2.1	Detailed Description	6
3.3	Package framework	6
3.3.1	Detailed Description	6
3.4	Package 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

4	Class Documentation	9
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	12
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	13
4.3.3.2	DEBUG_STR	13
4.3.3.3	VERBOSE	13
4.3.3.4	VERBOSE_STR	13
4.4	microcontroller.Microcontroller Class Reference	13
4.4.1	Detailed Description	14
4.4.2	Constructor & Destructor Documentation	14
4.4.2.1	Microcontroller()	14
4.4.3	Member Function Documentation	14
4.4.3.1	getPin()	14

4.4.3.2	getUc_gpioNum()	15
4.4.3.3	getUc_manufacturer()	15
4.4.3.4	getUc_model()	15
4.4.3.5	getUc_pinNum()	16
4.4.3.6	processDocument()	16
4.5	configurator.GPIO.Mode Enum Reference	16
4.5.1	Detailed Description	17
4.5.2	Member Data Documentation	17
4.5.2.1	STR_NAME	17
4.6	configurator.GPIO.OutType Enum Reference	17
4.6.1	Detailed Description	18
4.6.2	Member Data Documentation	18
4.6.2.1	STR_NAME	18
4.7	microcontroller.Pin Class Reference	18
4.7.1	Detailed Description	20
4.7.2	Constructor & Destructor Documentation	20
4.7.2.1	Pin()	20
4.7.3	Member Function Documentation	21
4.7.3.1	getAdc()	21
4.7.3.2	getClock()	21
4.7.3.3	getFeat_adc()	21
4.7.3.4	getFeat_clock()	22
4.7.3.5	getFeat_i2c()	22
4.7.3.6	getFeat_int()	22
4.7.3.7	getFeat_reset()	22
4.7.3.8	getFeat_spi()	23
4.7.3.9	getFeat_timer()	23
4.7.3.10	getFeat_uart()	23

4.7.3.11	<code>getFunc_gnd()</code>	23
4.7.3.12	<code>getFunc_gpio()</code>	24
4.7.3.13	<code>getFunc_vcc()</code>	24
4.7.3.14	<code>getI2c()</code>	24
4.7.3.15	<code>getInt()</code>	24
4.7.3.16	<code>getName()</code>	25
4.7.3.17	<code>getNumber()</code>	25
4.7.3.18	<code>getPort()</code>	25
4.7.3.19	<code>getReset()</code>	25
4.7.3.20	<code>getSpi()</code>	26
4.7.3.21	<code>getTimer()</code>	26
4.7.3.22	<code>getUart()</code>	26
4.7.3.23	<code>isValid()</code>	26
4.7.3.24	<code>setAdc()</code>	26
4.7.3.25	<code>setClock()</code>	27
4.7.3.26	<code>setFeat_adc()</code>	27
4.7.3.27	<code>setFeat_clock()</code>	27
4.7.3.28	<code>setFeat_i2c()</code>	28
4.7.3.29	<code>setFeat_int()</code>	28
4.7.3.30	<code>setFeat_reset()</code>	28
4.7.3.31	<code>setFeat_spi()</code>	28
4.7.3.32	<code>setFeat_timer()</code>	29
4.7.3.33	<code>setFeat_uart()</code>	29
4.7.3.34	<code>setFunc_gnd()</code>	29
4.7.3.35	<code>setFunc_gpio()</code>	30
4.7.3.36	<code>setFunc_vcc()</code>	30
4.7.3.37	<code>setI2c()</code>	30
4.7.3.38	<code>setInt()</code>	31

4.7.3.39	setName()	31
4.7.3.40	setNumber()	31
4.7.3.41	setPort()	31
4.7.3.42	setReset()	32
4.7.3.43	setSpi()	32
4.7.3.44	setTimer()	32
4.7.3.45	setUart()	33
4.7.4	Member Data Documentation	33
4.7.4.1	DEF_FEATURE	33
4.7.4.2	DEF_FEATURE_AV	33
4.7.4.3	DEF_FUNCTION	33
4.7.4.4	DEF_NAME	34
4.7.4.5	DEF_NUMBER	34
4.7.4.6	DEF_PORT	34
4.7.4.7	DISABLE	34
4.7.4.8	ENABLE	34
4.8	configurator.PinConf Class Reference	34
4.8.1	Detailed Description	35
4.8.2	Constructor & Destructor Documentation	35
4.8.2.1	PinConf()	35
4.8.3	Member Function Documentation	35
4.8.3.1	getMode()	35
4.8.3.2	getOutType()	36
4.8.3.3	getPin()	36
4.8.3.4	getPort()	36
4.8.3.5	getPull()	37
4.8.3.6	getSpeed()	37
4.8.3.7	isAv_Adc()	37

4.8.3.8	isAv_altFunc()	37
4.8.3.9	isValid()	38
4.8.3.10	setMode()	38
4.8.3.11	setOutType()	38
4.8.3.12	setPull()	38
4.8.3.13	setSpeed()	39
4.8.4	Member Data Documentation	39
4.8.4.1	DF_MODE	39
4.8.4.2	DF_OUTTYPE	39
4.8.4.3	DF_PULL	39
4.8.4.4	DF_SPEED	40
4.9	configurator.GPIO.Pull Enum Reference	40
4.9.1	Detailed Description	40
4.9.2	Member Data Documentation	40
4.9.2.1	STR_NAME	41
4.10	configurator.GPIO.Speed Enum Reference	41
4.10.1	Detailed Description	41
4.10.2	Member Data Documentation	41
4.10.2.1	STR_NAME	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()	43

Chapter 1

Namespace Index

1.1 Packages

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

common	5
configurator	5
framework	6
microcontroller	6
xmlCreator	7
xmlParser	7

Chapter 2

Class Index

2.1 Class List

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

xmlCreator.ConfXmlWriter	9
common.ErrorCode	10
common.Features	11
microcontroller.Microcontroller	13
configurator.GPIO.Mode	16
configurator.GPIO.OutType	17
microcontroller.Pin	18
configurator.PinConf	34
configurator.GPIO.Pull	40
configurator.GPIO.Speed	41
xmlParser.TestMain	42
xmlParser.XmlOpener	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 gpioPins)
- 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 gpioPins )
```

Constructor

Parameters

<i>gpioPins</i>	NUmber of GPIO pins
-----------------	---------------------

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`

Static Public Attributes

- static final String `STR_NAME` = "Mode"

4.5.1 Detailed Description

GPIO modes

Author

Miguel Diaz

Version

0.1

4.5.2 Member Data Documentation

4.5.2.1 STR_NAME

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

Name as String

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

- src/configurator/GPIO/Mode.java

4.6 configurator.GPIO.OutType Enum Reference

Public Attributes

- **OTYPE_PUSH_PULL**
- **OTYPE_OPEN_DRAIN**
- **OTYPE_NOT_AVAILABLE**

Static Public Attributes

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

4.6.1 Detailed Description

Pin's output type

Author

Miguel Diaz

Version

0.1

4.6.2 Member Data Documentation

4.6.2.1 STR_NAME

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

Name as String

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

- src/configurator/GPIO/OutType.java

4.7 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` ()

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

4.7.1 Detailed Description

Basic pin object.

- [Pin](#) necessary characteristics:
 - Name
 - Number
- [Pin](#) optional characteristics:
 - Port
- [Pin](#) main functions:
 - VCC
 - GND
 - GPIO
- [Pin](#) features:
 - Interruption
 - ADC
 - UART
 - I2C
 - SPI
 - Clock
 - Reset

Author

Miguel Diaz

Version

0.1

4.7.2 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>	Pin's I2C
----------------	-----------

4.7.3.38 setInt()

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

Set the pin's interruption

Parameters

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

4.7.3.39 setName()

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

Set the pin's name

Parameters

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

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**

Static Public Attributes

- static final String **STR_NAME** = "Mode"

4.9.1 Detailed Description

Pin's pull resistor

Author

Miguel Diaz

Version

0.1

4.9.2 Member Data Documentation

4.9.2.1 STR_NAME

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

Name as String

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**

Static Public Attributes

- static final String **STR_NAME** = "Mode"

4.10.1 Detailed Description

Pin's speed

Author

Miguel Diaz

Version

0.1

4.10.2 Member Data Documentation

4.10.2.1 STR_NAME

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

Name as String

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, 13
 - DEBUG, 12
 - debugPrint, 12
 - VERBOSE_STR, 13
 - VERBOSE, 13
 - verbosePrint, 12
- ConfXmlWriter
 - xmlCreator::ConfXmlWriter, 9
- configurator, 5
- configurator.GPIO.Mode, 16
- configurator.GPIO.OutType, 17
- configurator.GPIO.Pull, 40
- configurator.GPIO.Speed, 41
- configurator.PinConf, 34
- configurator::GPIO::Mode
 - STR_NAME, 17
- configurator::GPIO::OutType
 - STR_NAME, 18
- configurator::GPIO::Pull
 - STR_NAME, 40
- configurator::GPIO::Speed
 - STR_NAME, 41
- configurator::PinConf
 - DF_MODE, 39
 - DF_OUTTYPE, 39
 - DF_PULL, 39
 - DF_SPEED, 39
 - getMode, 35
 - getOutType, 36
 - getPin, 36
 - getPort, 36
 - getPull, 36
 - getSpeed, 37
 - isAv_Adc, 37
 - isAv_altFunc, 37
 - isValid, 37
 - PinConf, 35

- setMode, 38
- setOutType, 38
- setPull, 38
- setSpeed, 39

- DEBUG_STR
 - common::Features, 13
- DEBUG
 - common::Features, 12
- DEF_FEATURE_AV
 - microcontroller::Pin, 33
- DEF_FEATURE
 - microcontroller::Pin, 33
- DEF_FUNCTION
 - microcontroller::Pin, 33
- DEF_NAME
 - microcontroller::Pin, 33
- DEF_NUMBER
 - microcontroller::Pin, 34
- DEF_PORT
 - microcontroller::Pin, 34
- DF_MODE
 - configurator::PinConf, 39
- DF_OUTTYPE
 - configurator::PinConf, 39
- DF_PULL
 - configurator::PinConf, 39
- DF_SPEED
 - configurator::PinConf, 39
- DISABLE
 - microcontroller::Pin, 34
- debugPrint
 - common::Features, 12
- ENABLE
 - microcontroller::Pin, 34

- framework, 6

- getAdc
 - microcontroller::Pin, 21
- getClock
 - microcontroller::Pin, 21
- getFeat_adc
 - microcontroller::Pin, 21
- getFeat_clock

- microcontroller::Pin, [21](#)
- getFeat_i2c
 - microcontroller::Pin, [22](#)
- getFeat_int
 - microcontroller::Pin, [22](#)
- getFeat_reset
 - microcontroller::Pin, [22](#)
- getFeat_spi
 - microcontroller::Pin, [22](#)
- getFeat_timer
 - microcontroller::Pin, [23](#)
- getFeat_uart
 - microcontroller::Pin, [23](#)
- getFunc_gnd
 - microcontroller::Pin, [23](#)
- getFunc_gpio
 - microcontroller::Pin, [23](#)
- getFunc_vcc
 - microcontroller::Pin, [24](#)
- getI2c
 - microcontroller::Pin, [24](#)
- getInt
 - microcontroller::Pin, [24](#)
- getMode
 - configurator::PinConf, [35](#)
- getName
 - microcontroller::Pin, [24](#)
- getNumber
 - microcontroller::Pin, [25](#)
- getOutType
 - configurator::PinConf, [36](#)
- getParsedDoc
 - xmlParser::XmlOpener, [43](#)
- getPin
 - configurator::PinConf, [36](#)
 - microcontroller::Microcontroller, [14](#)
- getPort
 - configurator::PinConf, [36](#)
 - microcontroller::Pin, [25](#)
- getPull
 - configurator::PinConf, [36](#)
- getReset
 - microcontroller::Pin, [25](#)
- getSpeed
 - configurator::PinConf, [37](#)
- getSpi
 - microcontroller::Pin, [25](#)
- getTimer
 - microcontroller::Pin, [26](#)
- getUart
 - microcontroller::Pin, [26](#)
- getUc_gpioNum
 - microcontroller::Microcontroller, [15](#)
- getUc_manufacturer
- microcontroller::Microcontroller, [15](#)
- getUc_model
 - microcontroller::Microcontroller, [15](#)
- getUc_pinNum
 - microcontroller::Microcontroller, [15](#)
- isAv_Adc
 - configurator::PinConf, [37](#)
- isAv_altFunc
 - configurator::PinConf, [37](#)
- isValid
 - configurator::PinConf, [37](#)
 - microcontroller::Pin, [26](#)
- main
 - xmlParser::TestMain, [42](#)
- Microcontroller
 - microcontroller::Microcontroller, [14](#)
- microcontroller, [6](#)
- microcontroller.Microcontroller, [13](#)
- microcontroller.Pin, [18](#)
- microcontroller::Microcontroller
 - getPin, [14](#)
 - getUc_gpioNum, [15](#)
 - getUc_manufacturer, [15](#)
 - getUc_model, [15](#)
 - getUc_pinNum, [15](#)
 - Microcontroller, [14](#)
 - processDocument, [16](#)
- microcontroller::Pin
 - DEF_FEATURE_AV, [33](#)
 - DEF_FEATURE, [33](#)
 - DEF_FUNCTION, [33](#)
 - DEF_NAME, [33](#)
 - DEF_NUMBER, [34](#)
 - DEF_PORT, [34](#)
 - DISABLE, [34](#)
 - ENABLE, [34](#)
 - getAdc, [21](#)
 - getClock, [21](#)
 - getFeat_adc, [21](#)
 - getFeat_clock, [21](#)
 - getFeat_i2c, [22](#)
 - getFeat_int, [22](#)
 - getFeat_reset, [22](#)
 - getFeat_spi, [22](#)
 - getFeat_timer, [23](#)
 - getFeat_uart, [23](#)
 - getFunc_gnd, [23](#)
 - getFunc_gpio, [23](#)
 - getFunc_vcc, [24](#)
 - getI2c, [24](#)
 - getInt, [24](#)
 - getName, [24](#)
 - getNumber, [25](#)

- getPort, [25](#)
- getReset, [25](#)
- getSpi, [25](#)
- getTimer, [26](#)
- getUart, [26](#)
- isValid, [26](#)
- Pin, [20](#)
- setAdc, [26](#)
- setClock, [27](#)
- setFeat_adc, [27](#)
- setFeat_clock, [27](#)
- setFeat_i2c, [28](#)
- setFeat_int, [28](#)
- setFeat_reset, [28](#)
- setFeat_spi, [28](#)
- setFeat_timer, [29](#)
- setFeat_uart, [29](#)
- setFunc_gnd, [29](#)
- setFunc_gpio, [30](#)
- setFunc_vcc, [30](#)
- setI2c, [30](#)
- setInt, [31](#)
- setName, [31](#)
- setNumber, [31](#)
- setPort, [31](#)
- setReset, [32](#)
- setSpi, [32](#)
- setTimer, [32](#)
- setUart, [33](#)
- microcontroller::Pin, [27](#)
- setFeat_i2c
 - microcontroller::Pin, [28](#)
- setFeat_int
 - microcontroller::Pin, [28](#)
- setFeat_reset
 - microcontroller::Pin, [28](#)
- setFeat_spi
 - microcontroller::Pin, [28](#)
- setFeat_timer
 - microcontroller::Pin, [29](#)
- setFeat_uart
 - microcontroller::Pin, [29](#)
- setFunc_gnd
 - microcontroller::Pin, [29](#)
- setFunc_gpio
 - microcontroller::Pin, [30](#)
- setFunc_vcc
 - microcontroller::Pin, [30](#)
- setI2c
 - microcontroller::Pin, [30](#)
- setInt
 - microcontroller::Pin, [31](#)
- setMode
 - configurator::PinConf, [38](#)
- setName
 - microcontroller::Pin, [31](#)
- setNumber
 - microcontroller::Pin, [31](#)
- setOutType
 - configurator::PinConf, [38](#)
- setPort
 - microcontroller::Pin, [31](#)
- setPull
 - configurator::PinConf, [38](#)
- setReset
 - microcontroller::Pin, [32](#)
- setSpeed
 - configurator::PinConf, [39](#)
- setSpi
 - microcontroller::Pin, [32](#)
- setTimer
 - microcontroller::Pin, [32](#)
- setUart
 - microcontroller::Pin, [33](#)
- VERBOSE_STR
 - common::Features, [13](#)
- VERBOSE
 - common::Features, [13](#)
- verbosePrint
 - common::Features, [12](#)
- writeXml
 - xmlCreator::ConfXmlWriter, [10](#)
- NO_ERROR
 - common::ErrorCode, [11](#)
- OpenFile
 - xmlParser::XmlOpener, [43](#)
- Pin
 - microcontroller::Pin, [20](#)
- PinConf
 - configurator::PinConf, [35](#)
- processDocument
 - microcontroller::Microcontroller, [16](#)
- STR_NAME
 - configurator::GPIO::Mode, [17](#)
 - configurator::GPIO::OutType, [18](#)
 - configurator::GPIO::Pull, [40](#)
 - configurator::GPIO::Speed, [41](#)
- setAdc
 - microcontroller::Pin, [26](#)
- setClock
 - microcontroller::Pin, [27](#)
- setFeat_adc
 - microcontroller::Pin, [27](#)
- setFeat_clock

- 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](#)