

Wheelphone Library

Generated by Doxygen 1.7.6.1

Fri May 31 2013 16:57:48

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	com.wheelphone.wheelphonelibrary.USBAccessoryManagerMessage.- MessageType Enum Reference	5
3.2	com.wheelphone.wheelphonelibrary.USBAccessoryManager.RETURN- _CODES Enum Reference	5
3.2.1	Detailed Description	5
3.3	com.wheelphone.wheelphonelibrary.USBAccessoryManager Class - Reference	6
3.3.1	Detailed Description	6
3.3.2	Constructor & Destructor Documentation	7
3.3.2.1	USBAccessoryManager	7
3.3.3	Member Function Documentation	7
3.3.3.1	available	7
3.3.3.2	disable	7
3.3.3.3	enable	7
3.3.3.4	ignore	8
3.3.3.5	isConnected	8
3.3.3.6	peek	8
3.3.3.7	read	8
3.3.3.8	write	9

3.4	com.wheelphone.wheelphonelibrary.USBAccessoryManager.USB-AccessoryManagerException Class Reference	9
3.4.1	Detailed Description	9
3.5	com.wheelphone.wheelphonelibrary.USBAccessoryManagerMessage - Class Reference	10
3.5.1	Detailed Description	10
3.5.2	Constructor & Destructor Documentation	10
3.5.2.1	USBAccessoryManagerMessage	10
3.5.2.2	USBAccessoryManagerMessage	11
3.5.2.3	USBAccessoryManagerMessage	11
3.5.2.4	USBAccessoryManagerMessage	11
3.6	com.wheelphone.wheelphonelibrary.WheelphoneRobot Class Reference	11
3.6.1	Constructor & Destructor Documentation	14
3.6.1.1	WheelphoneRobot	14
3.6.2	Member Function Documentation	15
3.6.2.1	calibrateSensors	15
3.6.2.2	disableCliffAvoidance	15
3.6.2.3	disableObstacleAvoidance	15
3.6.2.4	disableSoftAcceleration	15
3.6.2.5	disableSpeedControl	16
3.6.2.6	enableCliffAvoidance	16
3.6.2.7	enableObstacleAvoidance	16
3.6.2.8	enableSoftAcceleration	16
3.6.2.9	enableSpeedControl	16
3.6.2.10	getBatteryRaw	17
3.6.2.11	getBatteryVoltage	17
3.6.2.12	getFlagStatus	17
3.6.2.13	getFrontAmbient	17
3.6.2.14	getFrontAmbients	17
3.6.2.15	getFrontProx	18
3.6.2.16	getFrontProxs	18
3.6.2.17	getGroundAmbient	18
3.6.2.18	getGroundAmbients	18
3.6.2.19	getGroundProx	19

3.6.2.20	getGroundProxs	19
3.6.2.21	getLeftEncoder	19
3.6.2.22	getOdometry	19
3.6.2.23	getOdometryTheta	20
3.6.2.24	getOdometryX	20
3.6.2.25	getOdometryY	20
3.6.2.26	getRightEncoder	20
3.6.2.27	isCalibrating	20
3.6.2.28	isCharged	21
3.6.2.29	isCharging	21
3.6.2.30	isUSBConnected	21
3.6.2.31	pauseUSBCommunication	21
3.6.2.32	resumeUSBCommunication	21
3.6.2.33	setLeftSpeed	22
3.6.2.34	setOdometry	22
3.6.2.35	setOdometryParameters	22
3.6.2.36	setRawLeftSpeed	23
3.6.2.37	setRawRightSpeed	23
3.6.2.38	setRawSpeed	23
3.6.2.39	setRightSpeed	23
3.6.2.40	setSpeed	24
3.6.2.41	setUSBCommunicationTimeout	24
3.6.2.42	startUSBCommunication	24
4	File Documentation	27
4.1	H:/lavoro/local-repo/phonebot-working/_android-side/Wheelphone- Library/src/com/wheelphone/wheelphonelibrary/WheelphoneRobot.java File Reference	27
4.1.1	Detailed Description	27

Chapter 1

Class Index

1.1 Class List

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

com.wheelphone.wheelphonelibrary.USBAccessoryManagerMessage.- MessageType	5
com.wheelphone.wheelphonelibrary.USBAccessoryManager.RETURN_- CODES	5
com.wheelphone.wheelphonelibrary.USBAccessoryManager	6
com.wheelphone.wheelphonelibrary.USBAccessoryManager.USB- AccessoryManagerException	9
com.wheelphone.wheelphonelibrary.USBAccessoryManagerMessage . .	10
com.wheelphone.wheelphonelibrary.WheelphoneRobot	11

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphonenumberlibrary/-	
WheelphoneRobot.java	
Main Wheelphone class	27

Chapter 3

Class Documentation

3.1 `com.wheelphone.wheelphonelibrary.USBAccessoryManager-Message.MessageType` Enum Reference

Public Attributes

- **READ**
- **ERROR**
- **CONNECTED**
- **DISCONNECTED**

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

- `H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphonelibrary/-USBAccessoryManagerMessage.java`

3.2 `com.wheelphone.wheelphonelibrary.USBAccessoryManager.-RETURN_CODES` Enum Reference

Public Attributes

- **DEVICE_MANAGER_IS_NULL**
- **ACCESSORIES_LIST_IS_EMPTY**
- **FILE_DESCRIPTOR_WOULD_NOT_OPEN**
- **PERMISSION_PENDING**

3.2.1 Detailed Description

Enumeration of possible return values for the enable function

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

- H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphone/USBAccessoryManager.java

3.3 com.wheelphone.wheelphonelibrary.USBAccessoryManager Class Reference

Classes

- class **ReadThread**
- enum **RETURN_CODES**
- class **USBAccessoryManagerException**

Public Member Functions

- **USBAccessoryManager** (Handler handler, int what)
- **RETURN_CODES enable** (Context context, Intent intent)
- void **disable** (Context context)
- boolean **isConnected** ()
- void **write** (byte[] data)
- boolean **isClosed** ()

Package Functions

- void **ignore** (int num)
- int **peek** (byte[] array)
- int **available** ()
- int **read** (byte[] array)

3.3.1 Detailed Description

A class created to assist in making accessing a USB accessory easier for those that are less familiar with programming in Java, working with threads/handlers/synchronization, and those that are not familiar with the Open Accessory framework interface

Author

Microchip Technology Inc.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 `com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager.USBAccessoryManager (Handler handler, int what)`

Public API Creates new USB Accessory Manager

Parameters

<i>handler</i>	The handler where to send USB accessory event messages
<i>what</i>	The "what" value to use for USB accessory event messages

3.3.3 Member Function Documentation

3.3.3.1 `int com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager.available () [package]`

Indicates the number of bytes that are currently in the read buffer. There will be at least this many bytes to read from the buffer (as long as the accessory has not detach or been closed since the call to this function.

Returns

the number of bytes available in the read queue

3.3.3.2 `void com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager.disable (Context context)`

Disables the USB manager and releases all resources

Parameters

<i>context</i>	The context that the manager was enabled with
----------------	-----------------------------------------------

3.3.3.3 `RETURN_CODES com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager.enable (Context context, Intent intent)`

Enables the

Parameters

<i>context</i>	The context that the USB manager should register to
----------------	-----------------------------------------------------

Returns

RETURN_CODES (p. 5) - the status of the enable request

3.3.3.4 void **com.wheelphone.wheelphonelibrary.USBAccessoryManager.ignore** (int *num*) [*package*]

I/O API discards the specified number of bytes from the internal read buffer

Parameters

<i>num</i>	the number of bytes to discard
------------	--------------------------------

3.3.3.5 boolean **com.wheelphone.wheelphonelibrary.USBAccessoryManager.isConnected** ()

Describes if an accessory is attached or not

Returns

boolean - true if one is attached, false otherwise

3.3.3.6 int **com.wheelphone.wheelphonelibrary.USBAccessoryManager.peek** (byte[] *array*) [*package*]

fills the array with data from the read buffer without discarding it.

Parameters

<i>array</i>	the buffer to fill
--------------	--------------------

Returns

the number of bytes copied from the buffer

3.3.3.7 int **com.wheelphone.wheelphonelibrary.USBAccessoryManager.read** (byte[] *array*) [*package*]

Reads bytes from the read buffer, removing them from the buffer once read

Parameters

<i>array</i>	where to copy the data
--------------	------------------------

3.4 com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager.USBAccessoryManagerException Class

Reference

9

Returns

the number of bytes copied (maximum will be the length of the array param)

3.3.3.8 void com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager.write (byte[] data)

Writes data to the accessory

Parameters

<i>data</i>	the data to write
-------------	-------------------

Exceptions

<i>InterruptedException</i>

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

- H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphonenumberlibrary/-USBAccessoryManager.java

3.4 com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager.-USBAccessoryManagerException Class Reference

Public Member Functions

- **USBAccessoryManagerException** (String message)
- String **toString** ()

Package Attributes

- String **errorMessage**

3.4.1 Detailed Description

Exception definition section Exception that can be thrown by the manager - currently not used

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

- H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphonenumberlibrary/-USBAccessoryManager.java

3.5 com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager-Message Class Reference

Classes

- enum **MessageType**

Public Member Functions

- **USBAccessoryManagerMessage** (**MessageType** type)
- **USBAccessoryManagerMessage** (**MessageType** type, byte[] data)
- **USBAccessoryManagerMessage** (**MessageType** type, byte[] data, UsbAccessory accessory)
- **USBAccessoryManagerMessage** (**MessageType** type, UsbAccessory accessory)

Public Attributes

- **MessageType** type
- String **text** = null
- byte[] **data** = null
- UsbAccessory **accessory** = null

3.5.1 Detailed Description

Basic Message class for the **USBAccessoryManager** (p. 6). This is used to send messages from the USB Accessory's read thread to the GUI thread to notify the GUI thread of various USB Accessory events (like data available or device attachment).

Author

Microchip Technologies Inc.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 com.wheelphone.wheelphonenumberlibrary.USBAccessoryManager-Message.USBAccessoryManagerMessage (**MessageType** type)

Creates new message of specified type

Parameters

<i>type</i>	The type of this message
-------------	--------------------------

3.6 com.wheelphone.wheelphonelibrary.WheelphoneRobot Class Reference 11

3.5.2.2 com.wheelphone.wheelphonelibrary.USBAccessoryManagerMessage.- USBAccessoryManagerMessage (MessageType *type*, byte[] *data*)

Creates a new message of specified type with specified data

Parameters

<i>type</i>	The type of this message
<i>data</i>	The data associated with this message

3.5.2.3 com.wheelphone.wheelphonelibrary.USBAccessoryManagerMessage.- USBAccessoryManagerMessage (MessageType *type*, byte[] *data*, UsbAccessory *accessory*)

Creates a new message of specified type with specified data

Parameters

<i>type</i>	The type of this message
<i>data</i>	The data associated with this message
<i>accessory</i>	The accessory associated with this message

3.5.2.4 com.wheelphone.wheelphonelibrary.USBAccessoryManagerMessage.U- SBAccessoryManagerMessage (MessageType *type*, UsbAccessory *accessory*)

Creates a new message of specified type with specified data

Parameters

<i>type</i>	The type of this message
<i>accessory</i>	The accessory associated with this message

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

- H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphonelibrary/-USBAccessoryManagerMessage.java

3.6 com.wheelphone.wheelphonelibrary.WheelphoneRobot Class - Reference

Classes

- class **communicationTask**

Public Member Functions

- **WheelphoneRobot** (Activity a, Context c)
Class constructor.
- void **startUSBCommunication** ()
To be inserted into the "onStart" function of the main activity class.
- void **resumeUSBCommunication** ()
To be inserted into the "onResume" function of the main activity class.
- void **pauseUSBCommunication** ()
To be inserted into the "onPause" function of the main activity class.
- void **setSpeed** (int l, int r)
Set the new left and right speeds for the robot. The new data will be actually sent to the robot when "sendCommandsToRobot" is called the next time within the timer communication task (50 ms cadence). This means that the robot speed will be updated after at most 50 ms (if the task isn't delayed by the system).
- void **setLeftSpeed** (int l)
Set the new left speed for the robot. For more details refer to "setSpeed".
- void **setRightSpeed** (int r)
Set the new right speed for the robot. For more details refer to "setSpeed".
- void **setRawSpeed** (int l, int r)
Set the new left and right speeds for the robot. For more details refer to "setSpeed".
- void **setRawLeftSpeed** (int l)
Set the new left speed for the robot. For more details refer to "setSpeed".
- void **setRawRightSpeed** (int r)
Set the new right speed for the robot. For more details refer to "setSpeed".
- void **enableSpeedControl** ()
Enable speed control on the robot (controller based on speed measure with back-emf).
- void **disableSpeedControl** ()
Disable speed control on the robot.
- void **enableSoftAcceleration** ()
Enable soft acceleration on the robot; this is useful when the robot is started fast (from standstill) because it avoid the robot to wheelie.
- void **disableSoftAcceleration** ()
Disable soft acceleration on the robot.
- void **enableObstacleAvoidance** ()
Enable obstacle avoidance onboard.
- void **disableObstacleAvoidance** ()
Disable obstacle avoidance onboard.
- void **enableCliffAvoidance** ()
Enable cliff avoidance onboard; when a cliff is detected the robot is stopped until this flag is reset.
- void **disableCliffAvoidance** ()
Disable cliff avoidance onboard.
- void **calibrateSensors** ()

3.6 com.wheelphone.wheelphonenumber.WheelphoneRobot Class Reference 13

Start the calibration of all the sensors. Use "isCalibrating" to know when the calibration is done.

- int **getBatteryRaw** ()
Returns the sampled value of the battery.
- float **getBatteryVoltage** ()
Returns the current battery voltage.
- byte **getFlagStatus** ()
Returns the flag byte that the robot set/clear itself.
- boolean **isCharging** ()
Returns the charging status.
- boolean **isCharged** ()
Returns the battery charged status.
- int **getLeftEncoder** ()
The value of the left encoder returned from the robot. The encoders values are based on the measured speed not on a real encoder device. The values given by the robot are the current encoders values, not the absolute value.
- int **getRightEncoder** ()
The value of the right encoder returned from the robot. For more details refer to "getLeftEncoder".
- int[] **getFrontProxs** ()
The robot has 4 front proximity sensors positioned as follows:
- int **getFrontProx** (int ind)
Return the corresponding front proximity sensor value. For more details refer to "getFrontProxs".
- int[] **getFrontAmbients** ()
The robot has 4 front ambient sensors, actually they are the front proximity sensors that can measure also the ambient light. The higher the value the lighter the environment.
- int **getFrontAmbient** (int ind)
Return the corresponding front ambient sensor value. For more details refer to "getFrontAmbients".
- int[] **getGroundProxs** ()
The robot has 4 ground proximity sensors positioned as follows:
- int **getGroundProx** (int ind)
Return the corresponding ground proximity sensor value. For more details refer to "getGroundProxs".
- int[] **getGroundAmbients** ()
The robot has 4 ground ambient sensors, actually they are the ground proximity sensors that can measure also the ambient light. The higher the value the lighter the environment.
- int **getGroundAmbient** (int ind)
Return the corresponding ground ambient sensor value. For more details refer to "getGroundAmbients".
- boolean **isUSBConnected** ()
Indicate whether the robot is connected (and exchanging packets) with the phone or not.
- void **setUSBCommunicationTimeout** (int ms)

This timeout sets how much to wait for a response from the robot before changing to a disconnected state.

- boolean **isCalibrating** ()
Tell whether the calibration is still in progress or not.
- double[] **getOdometry** ()
Return the odometry information resulting from the encoders values received by the robot. The positive X axis is pointing forward and the positive Y axis is pointing to the left side of the robot.
- double **getOdometryX** ()
Return the x absolute position in mm. For more information refer to "getOdometry".
- double **getOdometryY** ()
Return the y absolute position in mm. For more information refer to "getOdometry".
- double **getOdometryTheta** ()
Return the theta absolute angle in radians. For more information refer to "getOdometry".
- void **setOdometry** (double x, double y, double t)
Set/reset odometry components. For more information refer to "getOdometry".
- void **setOdometryParameters** (double dl, double dr, double wb)
Set/reset odometry parameters.
- void **appendLog** (String text)

Package Attributes

- int **rightEncoder** = 0
- int **rSpeed** = 0
- double **rightEncSum** = 0.0
- double **rightEncSumPrev** = 0.0
- double **finalTime** = 0.0
- double **totalTime** = 0.0

3.6.1 Constructor & Destructor Documentation

3.6.1.1 com.wheelphone.wheelphonelibrary.WheelphoneRobot.WheelphoneRobot (Activity a, Context c)

Class constructor.

Parameters

a	pass the main activity instance (this)
c	pass the main activity instance (this)

3.6 com.wheelphone.wheelphonelibrary.WheelphoneRobot Class Reference 15

Returns

WheelphoneRobot (p. 11) instance

3.6.2 Member Function Documentation

3.6.2.1 void com.wheelphone.wheelphonelibrary.WheelphoneRobot.calibrateSensors ()

Start the calibration of all the sensors. Use "isCalibrating" to know when the calibration is done.

Returns

none

3.6.2.2 void com.wheelphone.wheelphonelibrary.WheelphoneRobot.disableCliffAvoidance ()

Disable cliff avoidance onboard.

Returns

none

3.6.2.3 void com.wheelphone.wheelphonelibrary.WheelphoneRobot.disableObstacleAvoidance ()

Disable obstacle avoidance onboard.

Returns

none

3.6.2.4 void com.wheelphone.wheelphonelibrary.WheelphoneRobot.disableSoftAcceleration ()

Disable soft acceleration on the robot.

Returns

none

3.6.2.5 `void com.wheelphone.wheelphonelibrary.WheelphoneRobot.disableSpeedControl ()`

Disable speed control on the robot.

Returns

none

3.6.2.6 `void com.wheelphone.wheelphonelibrary.WheelphoneRobot.enableCliffAvoidance ()`

Enable cliff avoidance onboard; when a cliff is detected the robot is stopped until this flag is reset.

Returns

none

3.6.2.7 `void com.wheelphone.wheelphonelibrary.WheelphoneRobot.enableObstacleAvoidance ()`

Enable obstacle avoidance onboard.

Returns

none

3.6.2.8 `void com.wheelphone.wheelphonelibrary.WheelphoneRobot.enableSoftAcceleration ()`

Enable soft acceleration on the robot; this is useful when the robot is started fast (from standstill) because it avoid the robot to wheelie.

Returns

none

3.6.2.9 `void com.wheelphone.wheelphonelibrary.WheelphoneRobot.enableSpeedControl ()`

Enable speed control on the robot (controller based on speed measure with back-emf).

Returns

none

3.6 com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot Class Reference 17

3.6.2.10 int com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.getBatteryRaw ()

Returns the sampled value of the battery.

Returns

battery level (from 0 to 100)

3.6.2.11 float com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.getBatteryVoltage ()

Returns the current battery voltage.

Returns

battery voltage (from 3.7 to 4.2 volts)

3.6.2.12 byte com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.getFlagStatus ()

Returns the flag byte that the robot set/clear itself.

Returns

flag byte:

- bit 5: 1 => robot is charging, 0 => robot not charging
- bit 6: 1 => robot completely charged, 0 => robot not completely charged
- others bits not used

3.6.2.13 int com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.getFrontAmbient (int *ind*)

Return the corresponding front ambient sensor value. For more details refer to "getFrontAmbients".

Returns

sensor value

3.6.2.14 int [] com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.getFrontAmbients ()

The robot has 4 front ambient sensors, actually they are the front proximity sensors that can measure also the ambient light. The higher the value the lighter the environment.

Returns

array of size 4 containing the sensors values

3.6.2.15 `int com.wheelphone.wheelphonelibrary.WheelphoneRobot.getFrontProx (int ind)`

Return the corresponding front proximity sensor value. For more details refer to "get-FrontProxs".

Returns

sensor value

3.6.2.16 `int [] com.wheelphone.wheelphonelibrary.WheelphoneRobot.getFrontProxs ()`

The robot has 4 front proximity sensors positioned as follows:

```
1 2
0 3
```

The higher the value the nearer the object in front of the sensor.

Returns

array of size 4 containing the sensors values

3.6.2.17 `int com.wheelphone.wheelphonelibrary.WheelphoneRobot.getGroundAmbient (int ind)`

Return the corresponding ground ambient sensor value. For more details refer to "get-GroundAmbients".

Returns

sensor value

3.6.2.18 `int [] com.wheelphone.wheelphonelibrary.WheelphoneRobot.getGroundAmbients ()`

The robot has 4 ground ambient sensors, actually they are the ground proximity sensors that can measure also the ambient light. The higher the value the lighter the environment.

Returns

array of size 4 containing the sensors values

3.6 `com.wheelphone.wheelphonelibrary.WheelphoneRobot` Class Reference 19

3.6.2.19 `int com.wheelphone.wheelphonelibrary.WheelphoneRobot.getGroundProx (int ind)`

Return the corresponding ground proximity sensor value. For more details refer to "getGroundProxs".

Returns

sensor value

3.6.2.20 `int [] com.wheelphone.wheelphonelibrary.WheelphoneRobot.getGroundProxs ()`

The robot has 4 ground proximity sensors positioned as follows:

```
1 2
0 3
```

The higher the value the darker the object in front of the sensor.

Returns

array of size 4 containing the sensors values

3.6.2.21 `int com.wheelphone.wheelphonelibrary.WheelphoneRobot.getLeftEncoder ()`

The value of the left encoder returned from the robot. The encoders values are based on the measured speed not on a real encoder device. The values given by the robot are the current encoders values, not the absolute value.

Returns

left encoder (positive or negative)

3.6.2.22 `double [] com.wheelphone.wheelphonelibrary.WheelphoneRobot.getOdometry ()`

Return the odometry information resulting from the encoders values received by the robot. The positive X axis is pointing forward and the positive Y axis is pointing to the left side of the robot.

Returns

array of length 3 containing sequentially x position (mm), y position (mm), theta (radians).

3.6.2.23 `double com.wheelphone.wheelphonelibrary.WheelphoneRobot.getOdometryTheta ()`

Return the theta absolute angle in radians. For more information refer to "getOdometry".

Returns

theta (radians)

3.6.2.24 `double com.wheelphone.wheelphonelibrary.WheelphoneRobot.getOdometryX ()`

Return the x absolute position in mm. For more information refer to "getOdometry".

Returns

x position (mm)

3.6.2.25 `double com.wheelphone.wheelphonelibrary.WheelphoneRobot.getOdometryY ()`

Return the y absolute position in mm. For more information refer to "getOdometry".

Returns

y position (mm)

3.6.2.26 `int com.wheelphone.wheelphonelibrary.WheelphoneRobot.getRightEncoder ()`

The value of the right encoder returned from the robot. For more details refer to "getLeftEncoder".

Returns

right encoder (positive or negative)

3.6.2.27 `boolean com.wheelphone.wheelphonelibrary.WheelphoneRobot.isCalibrating ()`

Tell whether the calibration is still in progress or not.

Returns

true (calibration in progress), false otherwise

3.6 com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot Class Reference 21

3.6.2.28 boolean com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.isCharged ()

Returns the battery charged status.

Returns

true if battery charged, false otherwise

3.6.2.29 boolean com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.isCharging ()

Returns the charging status.

Returns

true if charging, false otherwise

3.6.2.30 boolean com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.isUSBConnected ()

Indicate whether the robot is connected (and exchanging packets) with the phone or not.

Returns

true (if robot connected), false otherwise

3.6.2.31 void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.pauseUSBCommunication ()

To be inserted into the "onPause" function of the main activity class.

Returns

none

3.6.2.32 void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.resumeUSBCommunication ()

To be inserted into the "onResume" function of the main activity class.

Returns

none

3.6.2.33 void `com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setLeftSpeed (int l)`

Set the new left speed for the robot. For more details refer to "setSpeed".

Parameters

	<i>l</i>	left speed given in mm/s
--	----------	--------------------------

Returns

none

3.6.2.34 void `com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setOdometry (double x, double y, double t)`

Set/reset odometry components. For more information refer to "getOdometry".

Parameters

	<i>x</i>	x position (mm)
	<i>y</i>	y position (mm)
	<i>t</i>	theta angle (radians)

Returns

none

3.6.2.35 void `com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setOdometryParameters (double dl, double dr, double wb)`

Set/reset odometry parameters.

Parameters

	<i>dl</i>	left wheel diameter (m)
	<i>dr</i>	right wheel diameter (m)
	<i>wb</i>	wheels distance (m)

Returns

none

3.6 `com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot` Class Reference 23

3.6.2.36 `void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setRawLeftSpeed (int l)`

Set the new left speed for the robot. For more details refer to "setSpeed".

Parameters

<i>l</i>	left speed (range is from -127 to 127)
----------	----------------------------------------

Returns

none

3.6.2.37 `void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setRawRightSpeed (int r)`

Set the new right speed for the robot. For more details refer to "setSpeed".

Parameters

<i>r</i>	right speed (range is from -127 to 127)
----------	-----------------------------------------

Returns

none

3.6.2.38 `void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setRawSpeed (int l, int r)`

Set the new left and right speeds for the robot. For more details refer to "setSpeed".

Parameters

<i>l</i>	left speed (range is from -127 to 127)
<i>r</i>	right speed (range is from -127 to 127)

Returns

none

3.6.2.39 `void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setRightSpeed (int r)`

Set the new right speed for the robot. For more details refer to "setSpeed".

Parameters

<i>r</i>	right speed given in mm/s
----------	---------------------------

Returns

none

3.6.2.40 void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setSpeed (int *l*, int *r*)

Set the new left and right speeds for the robot. The new data will be actually sent to the robot when "sendCommandsToRobot" is called the next time within the timer communication task (50 ms cadence). This means that the robot speed will be updated after at most 50 ms (if the task isn't delayed by the system).

Parameters

<i>l</i>	left speed given in mm/s
<i>r</i>	right speed given in mm/s

Returns

none

3.6.2.41 void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.setUSBCommunicationTimeout (int *ms*)

This timeout sets how much to wait for a response from the robot before changing to a disconnected state.

Parameters

<i>timeout</i>	in milliseconds
----------------	-----------------

Returns

none

3.6.2.42 void com.wheelphone.wheelphonenumberlibrary.WheelphoneRobot.startUSBCommunication ()

To be inserted into the "onStart" function of the main activity class.

3.6 com.wheelphone.wheelphonelibrary.WheelphoneRobot Class Reference 25

Returns

none

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

- **H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphonelibrary/WheelphoneRobot.java**

Chapter 4

File Documentation

4.1 H:/lavoro/local-repo/phonebot-working/_android-side/Wheelphone-Library/src/com/wheelphone/wheelphonelibrary/Wheelphone-Robot.java File Reference

Main Wheelphone class.

Classes

- class `com.wheelphone.wheelphonelibrary.WheelphoneRobot`
- class `com.wheelphone.wheelphonelibrary.WheelphoneRobot.communication-Task`

4.1.1 Detailed Description

Main Wheelphone class.

Author

Stefano Morgani <stefano@gctronic.com>

Version

1.0

Date

29.05.13

Copyright

GNU GPL v3

The WheelphoneRobot class is the main class that need to be instantiated in the application in order to communicate with the robot (receive sensors data and send commands). The low-level communication with the robot (packets exchange) is handled internally by this class.

Index

H:/lavoro/local-repo/phonebot-working/_android-side/WheelphoneLibrary/src/com/wheelphone/wheelphone/WheelphoneRobot.java, 27

USBAccessoryManager
com::wheelphone::wheelphonerlibrary::USBAccessoryManager, 7

USBAccessoryManagerMessage
com::wheelphone::wheelphonerlibrary::USBAccessoryManagerMessage, 10, 11

WheelphoneRobot
com::wheelphone::wheelphonerlibrary::WheelphoneRobot, 14

available
com::wheelphone::wheelphonerlibrary::USBAccessoryManager, 7

calibrateSensors
com::wheelphone::wheelphonerlibrary::WheelphoneRobot, 15

com.wheelphone.wheelphonerlibrary.USBAccessoryManager, 6

com.wheelphone.wheelphonerlibrary.USBAccessoryManager.RETURN_CODES, 5

com.wheelphone.wheelphonerlibrary.USBAccessoryManager.USBAccessoryManagerException, 9

com.wheelphone.wheelphonerlibrary.USBAccessoryManagerMessage, 10

com.wheelphone.wheelphonerlibrary.USBAccessoryManagerMessage.MessageType, 5

com.wheelphone.wheelphonerlibrary.WheelphoneRobot, 11

com::wheelphone::wheelphonerlibrary::USBAccessoryManager, 7

available, 7

disable, 7

enable, 7

ignore, 8

isConnected, 8

peek, 8

read, 8

write, 9

com::wheelphone::wheelphonerlibrary::USBAccessoryManagerMessage, 10, 11

USBAccessoryManagerMessage, 10, 11

com::wheelphone::wheelphonerlibrary::WheelphoneRobot, 14

calibrateSensors, 15

disableCliffAvoidance, 15

disableObstacleAvoidance, 15

disableSoftAcceleration, 15

disableSpeedControl, 15

enableCliffAvoidance, 16

enableObstacleAvoidance, 16

enableSoftAcceleration, 16

enableSpeedControl, 16

getBatteryRaw, 16

getBatteryVoltage, 17

getFlagStatus, 17

getFrontAmbient, 17

getFrontAmbients, 17

getFrontProx, 18

getFrontProxs, 18

getGroundAmbient, 18

getGroundAmbients, 18

getGroundProx, 18

getGroundProxs, 19

getLeftEncoder, 19

getOdometry, 19

getOdometryTheta, 19

getOdometryX, 20

getOdometryY, 20

getRightEncoder, 20
 isCalibrating, 20
 isCharged, 20
 isCharging, 21
 isUSBConnected, 21
 pauseUSBCommunication, 21
 resumeUSBCommunication, 21
 setLeftSpeed, 21
 setOdometry, 22
 setOdometryParameters, 22
 setRawLeftSpeed, 22
 setRawRightSpeed, 23
 setRawSpeed, 23
 setRightSpeed, 23
 setSpeed, 24
 setUSBCommunicationTimeout, 24
 startUSBCommunication, 24

 disable
 com::wheelphone::wheelphonedlibrary-
 ::USBAccessoryManager, 7
 disableCliffAvoidance
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 15
 disableObstacleAvoidance
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 15
 disableSoftAcceleration
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 15
 disableSpeedControl
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 15

 enable
 com::wheelphone::wheelphonedlibrary-
 ::USBAccessoryManager, 7
 enableCliffAvoidance
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 16
 enableObstacleAvoidance
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 16
 enableSoftAcceleration
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 16
 enableSpeedControl
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 16

 getBatteryRaw
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 16
 getBatteryVoltage
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 17
 getFlagStatus
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 17
 getFrontAmbient
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 17
 getFrontAmbients
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 17
 getFrontProx
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 18
 getFrontProxs
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 18
 getGroundAmbient
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 18
 getGroundAmbients
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 18
 getGroundProx
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 18
 getGroundProxs
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 19
 getLeftEncoder
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 19
 getOdometry
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 19
 getOdometryTheta
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 19
 getOdometryX
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 20
 getOdometryY
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 20
 getRightEncoder
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 20

ignore
 com::wheelphone::wheelphonedlibrary-
 ::USBAccessoryManager, 8

isCalibrating
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 20

isCharged
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 20

isCharging
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 21

isConnected
 com::wheelphone::wheelphonedlibrary-
 ::USBAccessoryManager, 8

isUSBConnected
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 21

pauseUSBCommunication
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 21

peek
 com::wheelphone::wheelphonedlibrary-
 ::USBAccessoryManager, 8

read
 com::wheelphone::wheelphonedlibrary-
 ::USBAccessoryManager, 8

resumeUSBCommunication
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 21

setLeftSpeed
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 21

setOdometry
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 22

setOdometryParameters
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 22

setRawLeftSpeed
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 22

setRawRightSpeed
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 23

setRawSpeed
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 23

setRightSpeed
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 23

setSpeed
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 24

setUSBCommunicationTimeout
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 24

startUSBCommunication
 com::wheelphone::wheelphonedlibrary-
 ::WheelphoneRobot, 24

write
 com::wheelphone::wheelphonedlibrary-
 ::USBAccessoryManager, 9