BluetoothHelper

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 BluetoothHelper Class Reference

A Bluetooth Java helper Class for Android.

Classes

- interface **BluetoothHelperListener**

Public Member Functions

- void **setBluetoothHelperListener** (**BluetoothHelperListener** listener)

  *Adds the specified **BluetoothHelper** (*p. *3) listener to receive events from this class.*
- boolean **isConnected** ()

  Returns the state of the connection.
- void **Connect** (String DeviceName)

  Connects to the remote device (server) with the specified name.
- void **Connect** (BluetoothDevice bluetoothDevice)

  Connects to the remote BluetoothDevice (server).
- void **ClearBuffer** ()

  Clear all pending incoming and outcoming messages.
- void **Disconnect** ()

  Disconnects from the connected remote device (server).
- void **Disconnect** (boolean **ClearBuffer**)

  Disconnects from the connected remote device (server).
- String **ReceiveMessage** ()

  Returns the oldest message received and buffered.
- boolean **SendMessage** (String msg)

  Send a message to the remote device.

Public Attributes

- char **Delimiter** = '\n'

  The messages delimiter.

2.1.1 Detailed Description

A Bluetooth Java helper Class for Android.

This Java Class implements an easy message-based Bluetooth wireless communication layer between an **Android device** (the client) and a **Microcontroller** (the server).

Using this class you can Connect, Disconnect, Send String messages, Receive String messages via Listener (best way) or with explicit polling, automatically reconnect and check the status of your Bluetooth connection in a simple and thread-safe way.
You can read the incoming messages attaching a Listener or using explicit polling.
Connection, reading and writing processes are asynchronously made using 3 separated Threads.
This Class is compatible with Android 4.0+

Version

1.0.5

Author

BasicAirData

2.1.2 Member Function Documentation

2.1.2.1 ClearBuffer()

```
void ClearBuffer ( )
```

Clear all pending incoming and outcoming messages.

It clears the inputMessageQueue and then the outputMessageQueue, used by separate threads to perform communication.
Normally this method is called by class itself during the Disconnection process, and should not be called.
The method is public in case of particular user needs.

2.1.2.2 Connect() [1/2]

```
void Connect (
             String DeviceName )
```

Connects to the remote device (server) with the specified name.

It bootstraps the connection to the paired (bonded) device named “DeviceName” if exists.
The function does return immediately, when the connection process is yet in progress, with no result.
You can receive a notification when the connection process is completed attaching a **BluetoothHelperListener** (p. 8). As an alternative you can check the connection status with the **isConnected()** (p. 6) method described below.
An onBluetoothHelperConnectionStateChanged event occurs (if listener is attached) when the connection process terminates, returning the new status of the connection. In case of success, the class will be ready to communicate with the remote device.

Parameters

DeviceName	The DeviceName of the remote Device

See also

BluetoothAdapter
BluetoothAdapter.getBondedDevices()
BluetoothDevice.getName()

2.1.2.3 Connect() [2/2]

```
void Connect (
             BluetoothDevice bluetoothDevice )
```

Connects to the remote BluetoothDevice (server).

It bootstraps the connection to the paired (bonded) device BluetoothDevice if exists.
The function does return immediately, when the connection process is yet in progress, with no result.
You can receive a notification when the connection process is completed attaching a **BluetoothHelperListener** (p. 8). As an alternative you can check the connection status with the **isConnected()** (p. 6) method described below.
An onBluetoothHelperConnectionStateChanged event occurs (if listener is attached) when the connection process terminates, returning the new status of the connection. In case of success, the class will be ready to communicate with the remote device.

Parameters

bluetoothDevice	The remote BluetoothDevice

See also

BluetoothAdapter
BluetoothAdapter.getBondedDevices()
BluetoothDevice

2.1.2.4 Disconnect() [1/2]

```
void Disconnect ( )
```

Disconnects from the connected remote device (server).

The method closes the Streams, the Socket and terminates all the threads that manage the connection and the communication.
An onBluetoothHelperConnectionStateChanged event occurs (if listener is attached) when the disconnection process terminates, returning the new status of the connection.

2.1.2.5 Disconnect() [2/2]

```
void Disconnect (
            boolean ClearBuffer )
```

Disconnects from the connected remote device (server).

The method closes the Streams, the Socket and terminates all the threads that manage the connection.
Then it clears all pending incoming and outcoming messages, if requested.
An onBluetoothHelperConnectionStateChanged event occurs (if listener is attached) when the disconnection process terminates, returning the new status of the connection.

Parameters

ClearBuffer	If true, disconnects from the remote device clearing all pending incoming outcoming messages; otherwise, disconnects without touching the queues.

2.1.2.6 isConnected()

```
boolean isConnected ( )
```

Returns the state of the connection.

The method returns true only if all the communication streams are opened.
It return false also in case of connection in progress.

Returns

The the connection state: true if the connection is opened, false otherwhise.

2.1.2.7 ReceiveMessage()

```
String ReceiveMessage ( )
```

Returns the oldest message received and buffered.

The incoming messages are asynchronously stored in a LinkedBlockingQueue. With this method you can get the oldest received message that you have not yet read. That message is deleted from the queue.
Each time you call ReceiveMessage method you'll obtain the next unread message.

Please note that the preferred method to receive messages is attaching a listener, with the setBluetoothHelper↩ Listener method described below.
If the listener is attached, ReceiveMessage will ever returns an empty string, because each received message is sent directly to the attached listener.

See also

setBluetoothHelperListener() (p. 7)

Returns

The String containing the message. An empty string otherwise

2.1.2.8 SendMessage()

```
boolean SendMessage (
             String msg )
```

Send a message to the remote device.

The message is stored in a LinkedBlockingQueue and asynchronously sent to the remote device by the dedicate thread.
The function returns true if the message is stored in the sending queue.

Parameters

msg	The message to send

Returns

true if the message is stored in the sending queue. false if a problem occurs

2.1.2.9 setBluetoothHelperListener()

```
void setBluetoothHelperListener (
              BluetoothHelperListener listener )
```

Adds the specified **BluetoothHelper** (p. 3) listener to receive events from this class.

Events occur when a message is received, or the connection status is changed. If it is null, no exception is thrown and no action is performed.
This is the preferred method to receive messages.

Each message you'll receive will be notified with a:

```
public void onBluetoothHelperMessageReceived(BluetoothHelper bluetoothhelper, String message)
```

Each time the status of the connection changes will be notified with a:

```
public void onBluetoothHelperConnectionStateChanged(BluetoothHelper bluetoothhelper, boolean is
```

Parameters

listener	The **BluetoothHelperListener** (p. 8)

2.1.3 Member Data Documentation

2.1.3.1 Delimiter

```
char Delimiter = '\n'
```

The messages delimiter.

That character is the separator between every incoming (and sent) messages.
It is automatically appended to every string passed to SendMessage, and removed to every message received.

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

- BluetoothHelper.java

2.2 BluetoothHelper.BluetoothHelperListener Interface Reference

Public Member Functions

- void **onBluetoothHelperMessageReceived** (**BluetoothHelper** bluetoothhelper, String message)

  Event fired each time a message is received from the remote device.
- void **onBluetoothHelperConnectionStateChanged** (**BluetoothHelper** bluetoothhelper, boolean **isConnected**)

  Event fired when the connection status changes.

2.2.1 Member Function Documentation

2.2.1.1 onBluetoothHelperConnectionStateChanged()

```
void onBluetoothHelperConnectionStateChanged (
              BluetoothHelper bluetoothhelper,
             boolean isConnected )
```

Event fired when the connection status changes.

The event is also fired when the **Connect()** (p. 4) method ends, returning the result of the **Connect()** (p. 4) request.

Parameters

bluetoothhelper	The **BluetoothHelper** (p. 3) class
isConnected	The status of the connection

2.2.1.2 onBluetoothHelperMessageReceived()

```
void onBluetoothHelperMessageReceived (
 BluetoothHelper bluetoothhelper,
String message )
```

Event fired each time a message is received from the remote device.

The event is received by your listeners in a worker thread;

Parameters

bluetoothhelper	The **BluetoothHelper** (p. 3) class
message	The message received

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

- BluetoothHelper.java