Class BluetoothHelper

Version 1.0.1

`public class BluetoothHelper`

This Java Class implements an easy message-based Bluetooth wireless comunication 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.

The actual work of BluetoothHelper is performed opening a `BluetoothSocket` class and using a `InputStream` and a `OutputStream` to pass data. Each message that you write is first put in a `LinkedBlockingQueue` and then asynchronously sent by the write thread. You can choose how to manage incoming messages: you can attach a Listener in order to receive notifications, or you can explicit Read messages on request.

Connection, reading and writing processes are asynchronously made using 3 separated Threads.
This Class is compatible with Android 4.0+

Constructor Summary

	Constructor and Description
	`BluetoothHelper()` Creates an unconnected Bluetooth helper.

Method Summary

Modifier and Type	Method and Description
`void`	**`ClearBuffer()`** Clear all pending incoming and outcoming messages.
`void`	**`Connect(String DeviceName)`** Connects to the remote server with the specified name.
`void`	**`Disconnect()`** Closes the connection.
`void`	**`Disconnect(boolean ClearBuffer)`** Closes the connection, clearing all pending messages if specified.
`boolean`	**`isConnected()`** Returns the connection state.
`String`	**`Read()`** Returns the oldest message received and buffered.
`void`	**`setBluetoothHelperListener(BluetoothHelperListener listener)`** Adds the specified listener to receive events from this BluetoothHelper.
`boolean`	**`Write()`** Send a message to the remote device.

Method Detail

ClearBuffer

```
public 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.

Connect

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

Connects to the remote device (server) with the specified name.
It bootstraps the connection to the paired device "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`. As an alternative you can chech the connection status with the `isConnected()` 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()`

Disconnect

```
public 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.

Disconnect

```
public 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.

isConnected

```
public boolean isConnected()
```

Returns the state of the connection.
The method returns `true` only if all the streams opened for communication 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.

Read

```
public String Read()
```

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 Read method you'll have the next unread message.

Please note that the preferred method to receive messages is attaching a listener, with the `setBluetoothHelperListener(BluetoothHelperListener listener)` method described below.
If the listener is attached, `Read` will ever returns an empty string, because each received message is sent directly to the attached listener.

Returns:

The String containing the message. An empty string otherwise.

See also:

`BluetoothHelper.setBluetoothHelperListener(BluetoothHelperListener listener)`

setBluetoothHelperListener

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

Adds the specified BluetoothHelper 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 isConnected)
```

Parameters:

`listener` – The BluetoothHelperListener.

Write

```
public boolean Write(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.

Returns:

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