

gattstatus (STATUS)

rssivalue (VALUE)

enablePolling /
disablePolling

readrssi

readrssireply (RES)

updategatt

setpollingdelay (VAL)

GattActor

GattDescriptor

<<data class>>
GattDescriptor

+ gatt: BluetoothGatt
+ gattCallback : LambdaGattCallback

LambdaGattCallback

- scope : CoroutineScope {val}
- onRssiReaded : MutableList<(BluetoothGatt,
Int, Int) -> Unit> = mutableListOf()
- onConnectionStateChange : MutableList<(BluetoothGatt?,
Int, Int) -> Unit> = mutableListOf()
- updateUiOnRssiRead : MutableList<(BluetoothGatt?,
Int, Int) -> Unit> = mutableListOf()
- updateUiOnConnectionStateChange : MutableList<(BluetoothGatt?, Int, Int) -> Unit> =
mutableListOf()

+ addOnRssiReaded(Boolean,
(BluetoothGatt, Int, Int) -> Unit) : Boolean
+ removeOnRssiReaded(
(BluetoothGatt, Int, Int) -> Unit) : Boolean
+ addOnConnectionStateChanged(Boolean,
(BluetoothGatt?, Int, Int) -> Unit) : Boolean
+ removeOnConnectionStateChanged(
(BluetoothGatt?, Int, Int) -> Unit) : Boolean
+ onReadRemoteRssi(BluetoothGatt, Int, Int)
+ onConnectionStateChange(BluetoothGatt?, Int, Int)

begin

idle

updateGatt

[else]

setpollingdelay

setPollingDelay

[else]

readrssi

[else]

handleRssiRequest

enablepolling
[ifIsPolling()]

enablePolling

disablePolling

$\epsilon - m$

disablepolling

[ifIsPolling()]

handleRssiRequest

setpollingdelay

polling

[time: pollingDelay]

[ifIsPolling()]

rssivalue (VALUE)

BluetoothGattCallback

