sd accessorycommunication impl::AccessoryCommunication command::AbstractCommand command::GetBatteryStateCommand Looper Handler requestBatteryState() postCommand(command: (Ausder Logik) AbstractCommand) post(new GetBatteryStateCommand(...)) obtainMessage() Die Verarbeitung findet im Thread statt. Per run() post(...) wurde die Message einer Wartschlange hinzugefügt. Diese wird in einer sendCommand() Schleife abgearbeitet. Sobald der Command getRequestCommandType():RequestCommand an der Reihe ist wird er verarbeitet. [payloadLength > 0] fillPayload(payload:ByteBuffer) receiveAndProcessResponse() receiveResponse():byte Die Aufrufe an AccessoryCommunication validateMessage(responseBuffer:byte[]) werden hier wieder in den ursprünglichen Thread mapResponseMessage(responseBuffer:byte[]): zurückgepostet und dann ResponseMessage an den Listener getPayload(responseMessage: weitergegeben. ResponseMessage, responseBuffer:byte[]): ByteBuffer processResponse(responseMessage ResponseMessage, payload: ByteBuffer) alt ResponseMessage processError(payload:ByteBuffer) [ERROR] errorReceived(message:String) [BATTERY NEAR EMPTY] processBatteryNearEmpty() batteryNearEmpty() [getExpectedResponeType() (hier BATTERY_STATE)] processExpectedResponse(payload:ByteBuffer) batteryStateReceived(chargingLevel:float) [Andere] Fehler