

**Smart tachograph version 2**  
**VU, Remote HMI specification**  
**& Transport protocol for ITS**

## **Introduction**

Regulation (EU) 2021/1228 (published on the 30<sup>th</sup> of July 2021) has modified Regulation (EU) 2016/799 with respect to requirements on construction, testing, installation, operation and repair of smart tachographs and their components. In view of this, the version 2 of the 2<sup>nd</sup> generation tachograph (commonly referred to as the ‘version 2 smart tachograph’) has been introduced in the EU regulatory framework.

The version 2 of the smart tachograph presents a standardized interface with intelligent transport systems (ITS) allowing the data recorded or produced by a tachograph to be used in operational mode, by an external device. Such external devices can obtain data from the tachograph via Bluetooth.

To harmonize certain technical specifications related to the use of Bluetooth, a group of experts composed of representatives from tachograph, card, and vehicle manufacturers agreed to meet in a dedicated and neutral working group hosted by Confederation of Organisations in Road Transport Enforcement (CORTE).

This document presents the outcomes and consensus achieved by the working group. The specifications described in the following two documents must be read in light of Regulation (EU) 2021/1228. The specifications consider only the interface between the tachograph and the external unit; any utilization of the interface by any client – such as mobile or in-vehicle – is optional.

Members of this dedicated Smart Tachograph ITS Interface Working Group, hosted by CORTE, included representatives from the following entities:

- ACEA;
- Continental;
- Intellic;
- Stoneridge;
- EU member states enforcement authorities;
- Representatives of the IRU, ETF, and DG MOVE.

The group also agreed to publish the specifications to make them available to all third parties. The specifications are available on the DG JRC, DG MOVE and CORTE websites.

For any questions, please contact CORTE at [secretariat@corte.be](mailto:secretariat@corte.be).

**Smart tachograph version 2  
VU, Remote HMI specification**

Version 1.0

# 1 Table of Contents

<b>1</b>	<b>Table of Contents .....</b>	<b>2</b>
<b>2</b>	<b>History[1_RHMI_163].....</b>	<b>8</b>
<b>3</b>	<b>Introduction[1_RHMI_164] .....</b>	<b>9</b>
3.1	Overview[1_RHMI_165] .....	9
3.2	Content[1_RHMI_167].....	9
3.3	Scope[1_RHMI_171] .....	9
<b>4</b>	<b>Remote Human Machine Interface Specification[1_RHMI_1] .....</b>	<b>10</b>
4.1	General[1_RHMI_179].....	10
4.2	Remote HMI session handling[1_RHMI_227].....	17
4.2.1	Routine ‘OpenRHMIsession’[1_RHMI_229].....	17
4.2.1.1	Routine ‘OpenRHMIsession’, StartRoutine Request[1_RHMI_251].....	18
4.2.1.2	Routine ‘OpenRHMIsession’, StartRoutine Positive Response[1_RHMI_253] .....	19
4.2.1.3	Routine ‘OpenRHMIsession’, StartRoutine Negative Response[1_RHMI_255] .....	19
4.2.1.4	Routine ‘OpenRHMIsession’, RequestRoutineResults (GetRHMISSessionStatus) Request[1_RHMI_257] .....	19
4.2.1.5	Routine ‘OpenRHMIsession’, RequestRoutineResults (GetRHMISSessionStatus) Positive Response[1_RHMI_259].....	20
4.2.1.5.1	Parameter RHMISSessionStatus[1_RHMI_249a].....	20
4.2.1.6	Routine ‘OpenRHMIsession’, RequestRoutineResults Negative Response[1_RHMI_264] .....	21
4.2.2	Routine CloseRHMISSession[1_RHMI_270].....	21
4.2.2.1	Routine ‘CloseRHMISSession’, StartRoutine Request[1_RHMI_271] .....	21
4.2.2.2	Routine ‘CloseRHMISSession’, StartRoutine Positive Response[1_RHMI_273] .....	21
4.3	Remote HMI Identifier/ checksum/ remote HMI client pairing[1_RHMI_275] .....	22
4.3.1	Remote HMI identifier for in-vehicle RHMI client[1_RHMI_291].....	27
4.3.2	Routine ‘PairRHMIclient’, StartRoutine Request[1_RHMI_302].....	27
4.3.2.1	Parameter RHMIClient [1_RHMI_306] .....	27
4.3.3	Routine ‘PairRHMIclient’, StartRoutine Positive Response[1_RHMI_303] .....	28
4.3.3.1	Parameter TransmittedRHMISSessionIdTransportType [1_RHMI_305] .....	28

4.3.3.2 Parameter TransportChecksum[1_RHMI_308] .....	28
4.3.4 Routine ‘PairRHMIclient’, StartRoutine Negative Response[1_RHMI_310] .....	29
4.3.5 Local HMI screens and user interaction for Remote HMI client pairing[1_RHMI_312] .....	29
4.3.6 Routine ‘VerifyRHMISessionId’, StartRoutine Request[1_RHMI_323].....	29
4.3.7 Routine ‘VerifyRHMISessionId’, StartRoutine Positive Response[1_RHMI_324] .....	30
4.3.8 Routine ‘VerifyRHMISessionId’, StartRoutine Negative Response[1_RHMI_325] .....	30
4.4 Remote set of driver activity[1_RHMI_326] .....	31
4.4.1 Routine ‘SetActivity’, StartRoutine Request[1_RHMI_332] .....	31
4.4.1.1 Parameter ActivityType[1_RHMI_328a] .....	32
4.4.2 Routine ‘SetActivity’, StartRoutine Positive Response[1_RHMI_327] .....	32
4.4.3 Routine ‘SetActivity’ , Negative Response code[1_RHMI_337] .....	32
4.5 Remote manual entries during driver card insertion procedure[1_RHMI_2] .....	33
4.5.1 Routine ‘Manual entry of driver activities and places’.[1_RHMI_12] .....	35
4.5.1.1 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetInsertedCardType, RequestRoutineResults Request[1_RHMI_14] .....	36
4.5.1.1.1 Parameter SmartCardSlot [1_RHMI_16a].....	36
4.5.1.2 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetInsertedCardType, RequestRoutineResults Positive Response[1_RHMI_17] .....	36
4.5.1.2.1 Parameter routineInfo[1_RHMI_20a] .....	37
4.5.1.3 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryPeriod, RequestRoutineResults Request[1_RHMI_187] .....	37
4.5.1.4 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryPeriod, RequestRoutineResults Positive Response[1_RHMI_192] .....	38
4.5.1.4.1 Parameter routineInfo[1_RHMI_189a] .....	38
4.5.1.4.2 Parameters ManualEntryPeriodBegin and ManualEntryPeriodEnd[1_RHMI_21a] .....	39
4.5.1.4.3 Parameter EndCountryAtCardWithdrawal and EndRegionAtCardWithdrawal[1_RHMI_197a] .....	39

4.5.1.5 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryStatus, RequestRoutineResults Request[1_RHMI_188] .....	39
4.5.1.6 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryStatus, RequestRoutineResults Positive Response[1_RHMI_193] .....	40
4.5.1.6.1 Parameter routineInfo[1_RHMI_190a] .....	40
4.5.1.7 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine Request[1_RHMI_193a] .....	41
4.5.1.7.1 Parameter RHMITimestamp[1_RHMI_339a] .....	42
4.5.1.8 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine Positive Response[1_RHMI_23] .....	42
4.5.1.9 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StopRoutine Request [1_RHMI_23a] .....	43
4.5.1.10 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StopRoutine Positive Response [1_RHMI_23b] .....	44
4.5.1.11 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, negative response code [1_RHMI_27] .....	44
4.6 Remote card withdrawal request[1_RHMI_43] .....	45
4.6.1 Routine ‘Request driver/workshop card withdrawal’[1_RHMI_49].....	46
4.6.1.1 Routine ‘DriverWorkshopCardWithdrawal’, Start Routine Request[1_RHMI_51] .....	46
4.6.1.1.1 Parameter SmartCardSlot[1_RHMI_51a].....	47
4.6.1.1.2 Parameters PlaceCountry and PlaceRegion[1_RHMI_355].....	47
4.6.1.1.3 Parameter CurrentShiftPrintoutRequest[1_RHMI_355a] .....	47
4.6.1.1.4 Parameter TechnicalDataPrintoutRequest [1_RHMI_355b].....	48
4.6.1.2 Routine ‘DriverWorkshopCardWithdrawal’, StartRoutine Positive Response[1_RHMI_56] .....	48
4.6.1.3 Routine ‘DriverWorkshopCardWithdrawalDefinition’, negative response code[1_RHMI_60a] .....	48
4.7 Remote start of a VU printout[1_RHMI_66] .....	50
4.7.1 Routine ‘Printout’[1_RHMI_69].....	51
4.7.1.1 Routine ‘Printout’, GetPrintoutPeriod, StartRoutine Request[1_RHMI_72] .....	51
4.7.1.1.1 Parameter PrintoutType and printout parameters overview[1_RHMI_74a] .....	52
4.7.1.2 Routine ‘Printout’, GetPrintoutPeriod, StartRoutine Positive Response[1_RHMI_199] .....	52

4.7.1.2.1 Parameters OldestPrintoutDate and NewestPrintoutDate[1_RHMI_81a] .....	53
4.7.1.3 Routine ‘Printout’, StartPrintout, StartRoutine Request[1_RHMI_202] .....	53
4.7.1.3.1 Parameter VUManufacturer [1_RHMI_202a] .....	55
4.7.1.3.2 VUManufacturerParameterSize [1_RHMI_202b] .....	55
4.7.1.3.3 VUManufacturerParameter [1_RHMI_202c] .....	55
4.7.1.3.4 Parameter PrintoutDate [1_RHMI_202d] .....	55
4.7.1.4 Routine ‘Printout’, StartPrintout, StartRoutine Positive Response[1_RHMI_77] .....	56
4.7.1.4.1 Parameter PrintoutType[1_RHMI_80a] .....	56
4.7.1.5 Routine ‘Printout’, RequestRoutineResults Request[1_RHMI_82] .....	56
4.7.1.6 Routine ‘Printout’, RequestRoutineResults Positive Response[1_RHMI_84] .....	56
4.7.1.7 Routine ‘Printout’, VU manufacturer specific parameters [1_RHMI_202e] .....	57
4.7.1.7.1 Stoneridge [1_RHMI_202f] .....	57
4.7.1.7.2 Continental [1_RHMI_202g] .....	57
4.7.1.7.3 Parameter routineInfo[1_RHMI_86a] .....	58
4.7.1.7.4 Parameter PrintoutType[1_RHMI_87a] .....	58
4.7.1.7.5 Parameters PrintoutEnd[1_RHMI_88a] .....	59
4.7.1.8 Routine ‘Printout’, negative response code [1_RHMI_89a] .....	59
4.8 Request and confirm active warnings and user information[1_RHMI_92] .....	60
4.8.1 Routine ‘AcknowledgeWarningsAndInfo’[1_RHMI_97] .....	62
4.8.1.1 Routine ‘AcknowledgeWarningsAndInfo’, GetActiveVUWarningsOverview, RequestRoutineResults Request[1_RHMI_99] .....	62
4.8.1.2 Routine ‘AcknowledgeWarningsAndInfo’, GetActiveVUWarningsOverview, RequestRoutineResults Positive Response[1_RHMI_101] .....	62
4.8.1.2.1 Parameter NumberOfActiveWarnings [1_RHMI_101a] .....	63
4.8.1.2.2 Parameter FirstActiveWarning [1_RHMI_101b] .....	63
4.8.1.2.3 Parameter WarningIndex [1_RHMI_101c] .....	63
4.8.1.3 Routine ‘AcknowledgeWarningsAndInfo’, GetActiveVUWarning, RequestRoutineResults Request [1_RHMI_101d] .....	63

4.8.1.4 Routine 'AcknowledgeWarningsAndInfo', GetActiveVUWarning, RequestRoutineResults Positive Response [1_RHMI_101e] .....	64
4.8.1.4.1 Parameter WarningVUSpecificCode [1_RHMI_101g].....	65
4.8.1.4.2 Parameter WarningStatus [1_RHMI_101h].....	65
4.8.1.4.3 Parameter NextActiveWarning [1_RHMI_101i].....	65
4.8.1.5 Routine 'AcknowledgeWarningsAndInfo', StartRoutine Request[1_RHMI_105] .....	65
4.8.1.6 Routine 'AcknowledgeWarningsAndInfo', StartRoutine Positive Response[1_RHMI_107].....	66
4.8.1.7 Routine 'AcknowledgeWarningsAndInfo', negative response code [1_RHMI_112a].....	67
4.9 Remote entry of places where daily work periods begin and/or end[1_RHMI_113] .....	67
4.9.1 Routine 'Entry of places where daily work periods begin and/or end'[1_RHMI_119] .....	68
4.9.1.1 Routine 'EntryOfPlace', StartRoutine Request[1_RHMI_121] .....	68
4.9.1.1.1 Parameter EntryType[1_RHMI_123a].....	69
4.9.1.1.2 Parameters PlaceCountry and PlaceRegion[1_RHMI_124a].....	69
4.9.1.2 Routine 'EntryOfPlace', StartRoutine Positive Response[1_RHMI_126] .....	69
4.9.1.3 Routine EntryOfPlace, negative response code[1_RHMI_130a].....	69
4.10 Remote entry of specific condition[1_RHMI_136].....	70
4.10.1 Routine 'Entry of specific condition'[1_RHMI_139] .....	71
4.10.1.1 Routine 'EntryOfSpecificCondition', StartRoutine Request[1_RHMI_141] .....	71
4.10.1.1.1 Parameter EntryType[1_RHMI_352].....	72
4.10.1.2 Routine 'EntryOfSpecificCondition', StartRoutine Positive Response[1_RHMI_144].....	72
4.10.1.3 Routine 'EntryOfSpecificCondition', negative response code [1_RHMI_352a].....	72
4.11 Load / unload entries [1_RHMI_149h] .....	73
4.11.1 Routine 'LoadUnload' [1_RHMI_149i] .....	73
4.11.1.1 Routine 'LoadUnload', StartRoutine Request [1_RHMI_149j].....	73
4.11.1.1.1 Parameter OperationType [1_RHMI_149k].....	74

4.11.1.2	Routine 'LoadUnload', StartRoutine Positive Response [1_RHMI_149l].....	74
4.11.1.3	Routine 'LoadUnload', negative responses [1_RHMI_149m] .....	74
<b>5</b>	<b>Annex [1_PR_1007a]</b> .....	<b>76</b>
5.1	Printout [1_PR_1007b] .....	76
5.2	Data Type Definitions[1_DaDi_15] .....	76
5.2.1	DailyWorkPeriodEntryType[1_DaDi_4057] .....	76
5.2.2	DriverActivityChange[1_DaDi_4124].....	76
5.2.3	DriverCardDriver[1_DaDi_4125] .....	77
5.2.4	DriverCardDriver1[1_DaDi_4126] .....	77
5.2.5	DriverCardDriver2[1_DaDi_4127] .....	77
5.2.6	DriverPlace[1_DaDi_4129].....	77
5.2.7	ManualEntryActivitiesAndPlaces[1_DaDi_4181].....	78
5.3	References to information defined by ISO 16844-7 [1_RHMI_176a].....	79
5.4	References [1_RHMI_176] .....	81

## 2 History [1\_RHMI\_163]

[1\_RHMI\_219]

Revision	Date	Author, Editor	Reason
1.0	2021-28-10	Smart Tachograph ITS Interface Working Group (ACEA, Continental, Intellic, Stoneridge, EU member states enforcement authorities, CORTE, IRU, ETF, DG MOVE).	Original version approved by the group

## 3 Introduction<sub>[1\_RHMI\_164]</sub>

### 3.1 Overview<sub>[1\_RHMI\_165]</sub>

[1\_RHMI\_166]

The purpose of the document is to comply to the [Annex 1C V2], requirements 61 and ITS\_09:

“The recording equipment shall allow drivers and workshops to alternately upload manual entries that need to be entered during the procedure through the ITS interface specified in Appendix 13 and, optionally, through other interfaces”.

This document describes the interfaces between a VU (see [Annex 1C V2]) and an external device (further called ‘remote HMI client’), which provides the VU HMI functionality to the user. A remote HMI client may be a mobile device (e.g. a smart phone) or be part of the vehicle HMI (e.g. part of the instrument cluster).

### 3.2 Content<sub>[1\_RHMI\_167]</sub>

[1\_RHMI\_170] **Note:** The square brackets [] within the requirement description contains references to specific requirements within the document or to other external documents.

### 3.3 Scope<sub>[1\_RHMI\_171]</sub>

[1\_RHMI\_172] Inside of the scope of this document:

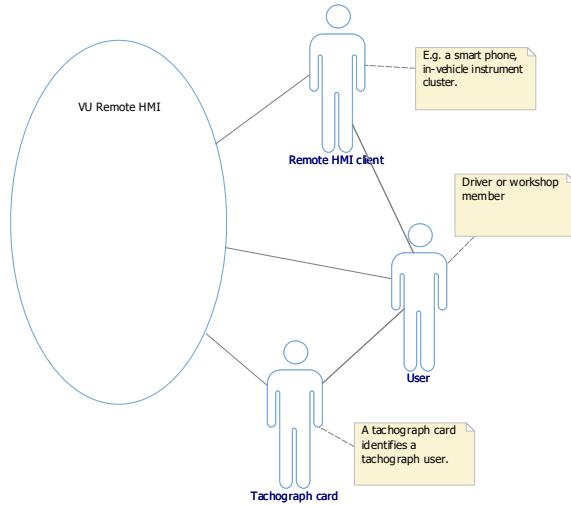
- General requirement on the VU, on the Remote HMI client and for the user;
- Application layer;
- Required data types, additional to [Annex 1C V2], Appendix 1;
- Requirements to store and publish the Remote HMI sessions.

Outside of this scope of this document:

- Physical layer, data link layer, network layer, transport layer and diagnostic application layer;
- Detailed local VU HMI requirements;
- Detailed requirements for the VU HMI on the remote HMI client;
- Data types defined by [Annex 1C V2], Appendix 1.

## 4 Remote Human Machine Interface Specification<sub>[1\_RHMI\_1]</sub>

### 4.1 General<sub>[1\_RHMI\_179]</sub>

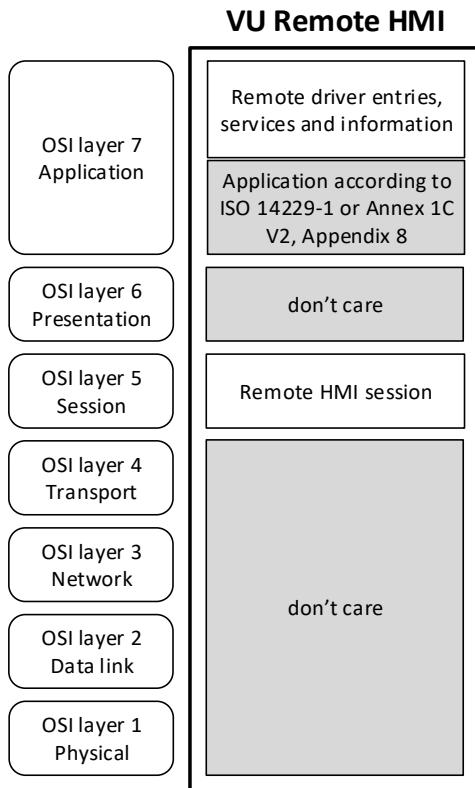


<sub>[1\_RHMI\_221]</sub>**Definition:** In context of the feature ‘Remote HMI’, the user means ‘driver (identified by the driver card)’ / ‘workshop (identified by the workshop card)’.

<sub>[1\_MB\_5889]</sub>The VU shall provide the information, the user entries and other services of the VU local HMI also via calibration/diagnostic interface according to [ISO 14229-1] and [Annex 1C V2], Appendix 8.

The VU shall provide at least the services and information required for the interaction to the users with roles “driver” and “workshop” according to the [Annex 1C V2].

**Note:** right now, in context of the Remote HMI, there are no use cases to support for control and company cards owners. Therefore, the Remote HMI only supports the owners of driver and workshop cards.



**Figure:** Remote HMI OSI layer overview according to [ISO 7498-1].

[1\_MB\_5890] The VU shall provide the Remote HMI via ITS interface according to [Annex 1C V2].

The VU may provide the Remote HMI also via other interfaces.

**Note:** The Remote HMI is not dependent on the ITS interface according to [Annex 1C V2] and is designed to work on each interface which supports the diagnostics according to [Annex 1C V2], Appendix 8 / [ISO 14229-1].

[1\_MB\_5891] The VU shall provide the Remote HMI independent of the VU ignition.

**Note:** the availability of the actual physical interface (e.g. CAN) may be dependent on other constraints.

**Note:** The availability/functionality of the Remote HMI may be restricted due to ADR / ATEX requirements.

[1\_MB\_5892] The VU shall provide at least following VU information, the user entries and other services of local VU HMI via Remote HMI:

User entry / RHMI service	Rationale	Reference	Comment
Remote HMI session handling		[1_RHMI_228]	

User entry / RHMI service	Rationale	Reference	Comment
Remote HMI Identifier/ checksum/ remote HMI client pairing		[1_RHMI_275]	
The procedure 'manual entry of driver activities / places', including all required entries by the user	[Annex 1C V2], requirement 61	[1_RHMI_3]	
The procedure 'withdrawal of the driver and workshop card', including all required entries by the user	[Annex 1C V2], requirement 20	[1_RHMI_44]	
Service to request VU printouts	[Annex 1C V2], requirement 169	[1_RHMI_70]	Mandatory are the [Annex 1C V2] printouts. VU manufacturer specific printouts may be supported.
Change of the activity of driver and co-driver	[Annex 1C V2], requirement 46	[1_RHMI_317]	
User entry of places	[Annex 1C V2], requirement 60	[1_RHMI_114]	
User entry of specific conditions	[Annex 1C V2], requirement 62	[1_RHMI_137]	
User entry of load / unload operations	[Annex 1C V2], requirement 62a	[1_RHMI_149a]	
Warnings about events / faults	[Annex 1C V2], requirements 189 and 190.	[1_RHMI_98]	Mandatory are the [Annex 1C V2] events and faults. VU manufacturer specific events and faults may be supported.

**Table:** mandatory RHMI services.

[1\_MB\_5892a] Optionally, the VU may provide additional VU information, user entries and other services of local VU HMI via Remote HMI.

## Smart Tachograph V2: VU, Remote Human Machine Interface Specification

[1\_MB\_5913] The VU shall provide via diagnostic interface the status of the current and last printout: see [1\_RHMI\_83].

[1\_RHMI\_223] The VU shall support one mobile Remote HMI client (e.g. a smart phone) paired to one of the inserted driver/workshop cards. This tachograph card can be inserted either into driver or co-driver smart card slot.

The VU may support second mobile Remote HMI client. This mobile Remote HMI client shall be paired to another inserted driver/workshop card.

**Note:** right now, it is not possible to pair more than one mobile Remote HMI client to the same driver/workshop card. If same mobile client has to be shared by driver and co-driver, then this client shall be paired to any of those drivers.

For more information, see chapter [1\_RHMI\_275] “Remote HMI client pairing”.

[1\_RHMI\_223a] The VU may support one Remote HMI client installed in the vehicle (e.g. instrument cluster). This Remote HMI client is paired directly to the VU.

**Note:** right now, it is not possible to pair more than one in-vehicle installed client.

For more information, see chapter [1\_RHMI\_275] “Remote HMI client pairing”.

[1\_RHMI\_223b]

**Note:** the VU users can use any paired Remote HMI clients to performed user entries, to request services and to get information for driver, co-driver and for the VU.

**Note:** there is no prioritisation of the RHMI clients.

**Note:** the ‘Remote HMI’ client may be same device as the ‘ITS unit’ according to [Annex 1C V2], Appendix 13.

### Rationale:

- most common use cases are “using own smart phone” and “using integrated vehicle HMI” as a Remote HMI client.
- according to [Annex 1C V2], ITS\_16, at the same time, Bluetooth® connection to one ITS unit is mandatory and to more is optional.

[1\_RHMI\_228a] The VU shall accept the user entries and other services requests via Remote HMI only if

- At least one driver / workshop card is inserted in the VU,
- All owners of the inserted tachograph cards accept the usage of the Remote HMI,
- All other VU specific Remote HMI conditions are fulfilled.

If all these conditions are fulfilled (i.e. Remote HMI can be used), the VU shall allow a Remote HMI client to open the ‘Remote HMI session’.

If the Remote HMI session is opened, only then the VU shall accept the user entries and other service requests via Remote HMI.

**Note:**

- There is only one VU Remote HMI session at the same time. All Remote HMI clients share the same session.
- If Remote HMI session is open, then any of the VU interfaces supporting Remote HMI and any of the paired Remote HMI clients can be used to perform Remote HMI requests.
- For detailed requirements for Remote HMI session, see chapter “Remote HMI session handling” [1\_RHMI\_227].
- Except the OpenRHMIsession, PairRHMIclient and VerifyRHMISessionId services, all other Remote HMI services require the open Remote HMI session.
- There may be further VU specific restrictions on Remote HMI availability (e.g. during vehicle moving; during usage of the local HMI by the user).

**Rationale:**

- “At least one driver / workshop card is inserted in the VU”: it must not be allowed to perform remote user entries in behalf of “unknown user”. Otherwise, the “spamming” of the VU records can be performed without trace to any user.
- “All owners of the inserted tachograph cards accept the usage of the Remote HMI”: the user shall be in control of the Remote HMI usage. Due to “any” paired remote HMI client can be used to perform **all** entries for driver, co-driver and VU”, all VU users (identified by the inserted tachograph cards) shall agree on the Remote HMI usage.

[1\_RHMI\_228b] The VU shall accept the remote user entries only if the user entry request contains the actual timestamp and valid ‘Remote HMI checksum’ (see [1\_RHMI\_279]).

**Note:**

- For more information, see chapter “Remote HMI Identifier/ checksum/ remote HMI client pairing” [1\_RHMI\_275].
- The dependency of a Remote HMI request on ‘Remote HMI checksum’ and the correct timestamp is defined in the request specification.

**Rationale:**

- The checksum ensures that only approved Remote HMI clients can perform remote user entries.
- The actual timestamp prevents the replay of the Remote HMI user entries.

## Smart Tachograph V2: VU, Remote Human Machine Interface Specification

[1\_RHMI\_228c] The VU shall accept only one Remote HMI request at the same time.

If a Remote HMI request is in process, and another Remote HMI request is received, then the VU shall reject another Remote HMI request by negative response 'busyRepeatRequest'.

**Rationale:** To avoid race conditions by concurrent Remote HMI requests.

[1\_RHMI\_224] If the Remote HMI session is open and a received Remote HMI request/entry is correct, then the VU shall accept the Remote HMI request/entry without confirmation on the VU local HMI.

[1\_RHMI\_225] If the Remote HMI session is closed (see [1\_RHMI\_228a]), then the VU shall reject all Remote HMI requests except the requests PairRHMIclient, VerifyRHMI sessionId, OpenRHMI session and GetRHMI session Status.

**Note:** the services PairRHMIclient, VerifyRHMI sessionId, OpenRHMI session and GetRHMI session Status are required to establish the Remote HMI session and to pair the Remote HMI client to the VU.

[1\_RHMI\_226] The VU shall verify the correctness of the Remote HMI request/entry by check the UTC timestamp of the request: If the timestamp of the request/entry differs more than -300 s to +300 s to the current VU UTC system time, then the VU shall reject such Remote HMI request/entry by negative response 'requestOutOfRange'.

**Rationale:**

- The actual timestamp shall prevent the replay of the Remote HMI user entries. See also [1\_RHMI\_228b].
- -300 s to +300 s: see [Annex 1C V2], maximum allowed deviation between internal VU time source and Galileo OSNMA..

[1\_RHMI\_338] **Note:** The Remote HMI supports only driver and workshop cards. See chapter "Remote HMI session handling" [1\_RHMI\_227].

**Rationale:** for company and control card owners, currently there are no use cases to be supported by the VU Remote HMI.

[1\_RHMI\_338a] **Note:** The Remote HMI uses the diagnostic service RoutineControl according to [ISO 14229-1] to provide Remote HMI entries and other services, and uses service ReadDataByIdentifier to provide information available via local VU HMI.

[1\_RHMI\_338b]

<b>Routine</b>	<b>Supported sub-functions</b>			<b>routine Identifier</b>	<b>routineControlOption</b>
	<b>StartRoutine</b>	<b>StopRoutine</b>	<b>RequestRoutine Results</b>		
ManualEntryOfDriverActivitiesAndPlaces	X	X	X	'F200'H	See [1_RHMI_3].
DriverWorkshopCardWithdrawal	X	-	-	'F201'H	See [1_RHMI_44].
AcknowledgeWarningsAndInfo	X	-	X	'F202'H	See [1_RHMI_93].
EntryOfPlace	X	-	-	'F203'H	See [1_RHMI_114].
EntryOfSpecificCondition	X	-	-	'F204'H	See [1_RHMI_139].
SetActivity	X	-	-	'F205'H	See [1_RHMI_326].
LoadUnload	X	-	-	'F206'H	See [1_RHMI_149a].
Reserved for future use (Remote HMI)	-	-	-	'F207'H.. 'F20F'H	
Printout	X	-	X	'F210'H	See [1_RHMI_67].
OpenRHMISESSION	X	-	X	'F211'H	See [1_RHMI_229].
CloseRHMISESSION	X	-	-	'F212'H	See [1_RHMI_270].
PairRHMIclient	X	-	-	'F213'H	See [1_RHMI_302].
VerifyRHMISESSIONId	X	-	-	'F214'H	See [1_RHMI_323].
Reserved for future use (Remote HMI)	-	-	-	'F215'H.. 'F2FF'H	

**Table:** Overview of VU routines used for Remote HMI.**Caption:**

X: sub-function supported for the routine.

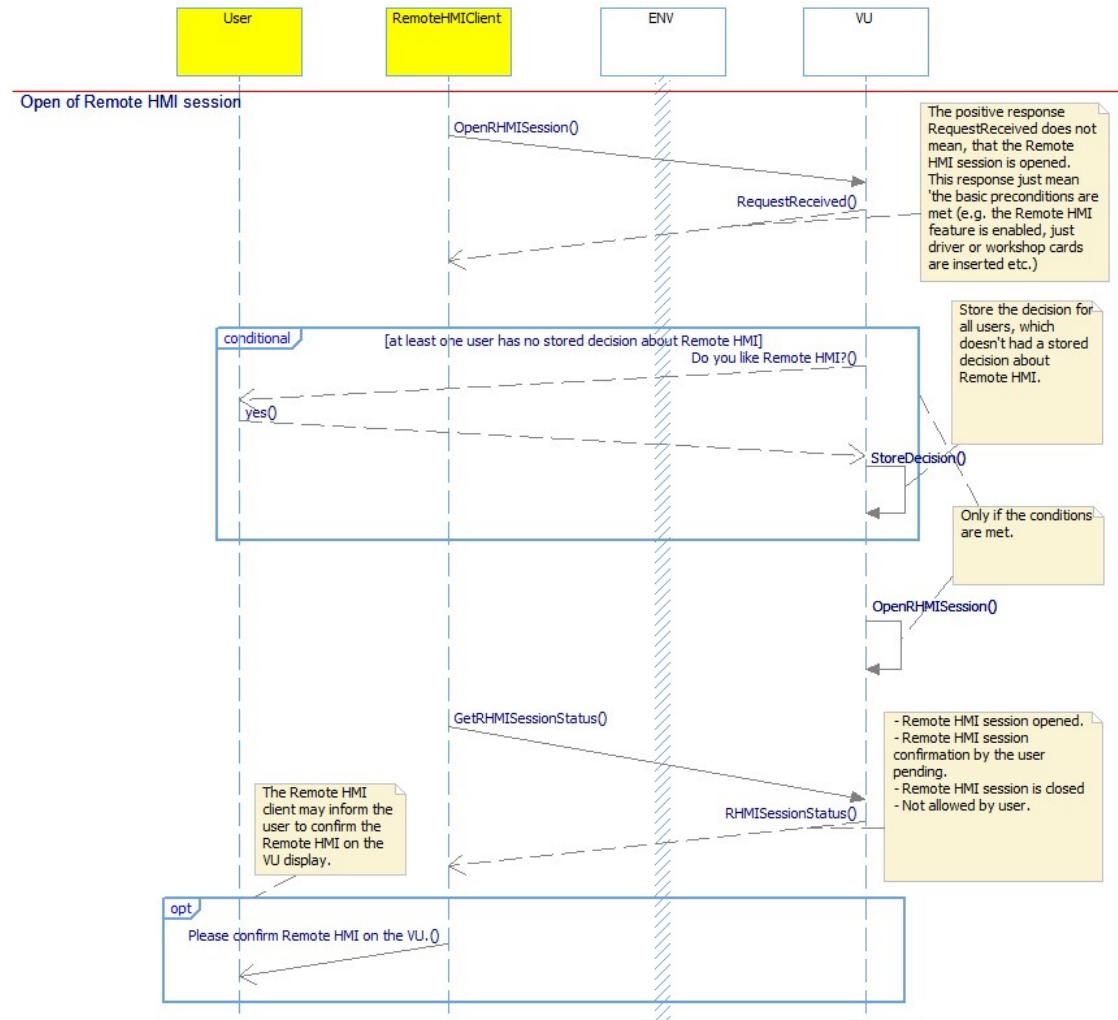
- : sub-function does not supported for the routine.

## 4.2 Remote HMI session handling [1\_RHMI\_227]

[1\_RHMI\_228] The VU shall provide the possibility to request Remote HMI session via RoutineControl service, routine OpenRHMIsession (see [1\_RHMI\_250]).

### 4.2.1 Routine ‘OpenRHMIsession’ [1\_RHMI\_229]

[1\_RHMI\_231]



**Figure:** Sequence ‘Open Remote HMI session’.

[1\_RHMI\_235] If the Remote HMI session is requested (see [1\_RHMI\_228]), then VU shall request the user(s) via local HMI either to confirm or to reject the usage of the Remote HMI for such user(s).

In case both users are requested, then the VU shall request each user separately.

[1\_RHMI\_234] If at least one user rejects the Remote HMI (see [1\_RHMI\_235]), then the VU must not open the Remote HMI session.

[1\_RHMI\_233] If a user confirms or rejects the Remote HMI (see [1\_RHMI\_235]) and [1\_RHMI\_236]), then the VU shall store the user(s) decision for the most recent different tachograph cards as defined by [Annex 1C V2], requirement 133.

[1\_RHMI\_244] If a user entry is accepted via Remote HMI, then the VU shall store the beginning and the end of the Remote HMI session for the inserted driver/workshop cards, separate for driver and co-driver slots.

The VU shall store this for at least 365 days (see [Annex 1C V2], chapter “3.12 Recording and storing in the data memory”), using the same secure mechanisms as for other [Annex 1C V2] data.

Here, a ‘user entry’ means one of following user entries via Remote HMI:

- set of the driver activity [1\_RHMI\_317];
- manual entry of driver activities [1\_RHMI\_3];
- card withdrawal request [1\_RHMI\_44];
- entry of places where daily work periods begin and/or end [1\_RHMI\_114];
- entry of specific condition [1\_RHMI\_137];
- entry of load/unload operation [1\_RHMI\_149a].

**Rationale:** It shall be possible to the user to identify when Remote HMI was used.

[1\_RHMI\_242] The VU shall provide the stored Remote HMI sessions (see [1\_RHMI\_244]) via printout as part of the ‘Driver Activities from VU Daily’ printout. See [1\_PR\_1007] for more information.

[1\_RHMI\_246] If a user rejected the Remote HMI usage (see [1\_RHMI\_235] and [1\_RHMI\_236]), then the VU must not accept the ‘Open Remote HMI session’ request.

[1\_RHMI\_247] During normal operation, the VU shall provide the possibility to the user to change the Remote HMI confirmation/reject decision via local VU HMI, separately for the user corresponding to tachograph cards inserted in the driver and co-driver smart card slots.

#### **4.2.1.1 Routine ‘OpenRHMIsession’, StartRoutine Request<sub>[1\_RHMI\_251]</sub>**

[1\_RHMI\_250] The RoutineControl – Start Routine ‘OpenRHMIsession’- Request message requests the Remote HMI session opening.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = OpenRHMIsession [byte#1(MSB), byte#2]	'F211'H

#### 4.2.1.2 Routine ‘OpenRHMIsession’, StartRoutine Positive Response<sub>[1\_RHMI\_253]</sub>

<sub>[1\_RHMI\_254]</sub> The RoutineControl – Start Routine ‘OpenRHMIsession’ – Positive Response message confirms the Remote HMI session preconditions fulfilment, except the user(s) decision on the Remote HMI.

**Note:** To get the user(s) decision, the request Routine ‘OpenRHMIsession’, RequestRoutineResults (GetRHMISSessionStatus) shall be used.

**Note:** If the Remote HMI session is already opened and the session preconditions are still fulfilled, then the VU will still respond positively and keep the RHMI session open.

A Remote HMI client may check the Remote HMI session status anytime by GetRHMISSessionStatus(see [1\_RHMI\_258]).

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = OpenRHMIsession [byte#1(MSB), byte#2]	'F211'H

#### 4.2.1.3 Routine ‘OpenRHMIsession’, StartRoutine Negative Response<sub>[1\_RHMI\_255]</sub>

<sub>[1\_RHMI\_256]</sub> This definition contains the negative response conditions for routine OpenRHMIsession, sub-function StartRoutine.

Response Name	Value	Description
conditionsNotCorrect	'22'H	The conditions for execution are not met – see [1_RHMI_228a].
busyRepeatRequest	'21'H	No (new) remote entry is possible at the moment: <ul style="list-style-type: none"> <li>• Other ‘Open Remote HMI session’ request is pending – wait for user decision via local HMI.</li> </ul>

#### 4.2.1.4 Routine ‘OpenRHMIsession’, RequestRoutineResults (GetRHMISSessionStatus) Request<sub>[1\_RHMI\_257]</sub>

<sub>[1\_RHMI\_258]</sub> The RoutineControl – RequestRoutineResults ‘OpenRHMIsession’- Request message requests the status of the Remote HMI session.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = OpenRHMIsession [byte#1(MSB), byte#2]	'F211'H

#### 4.2.1.5 Routine ‘OpenRHMIsession’, RequestRoutineResults (GetRHMIgetSessionStatus) Positive Response<sub>[1\_RHMI\_259]</sub>

[1\_RHMI\_260] The RoutineControl – RequestRoutineResults ‘OpenRHMIsession’- Positive Response message provides the status of the Remote HMI session.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = OpenRHMIsession [byte#1(MSB), byte#2]	'F211'H
5	routineStatusRecord[byte#1] = RHMIgetSessionStatus (see [1_RHMI_249])	'XX'H

##### 4.2.1.5.1 Parameter RHMIgetSessionStatus<sub>[1\_RHMI\_249a]</sub>

[1\_RHMI\_249]

Response Name	Value	Definition
RHMIgetSessionClosed-OpenPossible	'00'H	The Remote HMI session is closed but can be opened: all conditions for Remote HMI including user(s) confirmation are met.
RHMIgetSessionClosed-UserDecisionPending	'01'H	The Remote HMI session is (still) closed: the user is requested to confirm or to reject the Remote HMI.
RHMI getSessionIsOpen	'10'H	Remote HMI session is open.
RHMI getSessionClosed-UserRejected	'20'H	Remote HMI session is closed: The Remote HMI is rejected by a user.
RHMI getSessionClosed-LocalHMIisUsed	'21'H	Remote HMI session is closed: The user uses the local VU HMI.
RHMI getSessionClosed-GeneralConditionsAreNotMet	'2F'H	Remote HMI session is closed: General preconditions for Remote HMI session are not met. See [1_RHMI_256], negative response

Response Name	Value	Definition
		'conditionsNotCorrect' for more information.

#### 4.2.1.6 Routine ‘OpenRHMIsession’, RequestRoutineResults Negative Response<sub>[1\_RHMI\_264]</sub>

[1\_RHMI\_265] This definition contains the negative response conditions for routine OpenRHMIsession, sub-function RequestRoutineResults.

Response Name	Value	Description
conditionsNotCorrect	'22'H	The conditions for execution are not met: see [1_RHMI_228a].

#### 4.2.2 Routine CloseRHMI Session<sub>[1\_RHMI\_270]</sub>

[1\_RHMI\_272] The VU shall provide the possibility to request Remote HMI session closing via RoutineControl service, routine ‘CloseRHMI Session’ (see [1\_RHMI\_267]).

**Note:** It is not required to close the RHMI session explicitly. The RHMI session will be automatically closed if the conditions for the session are no longer fulfilled (see requirements in chapter [1\_RHMI\_229]).

##### 4.2.2.1 Routine ‘CloseRHMI Session’, StartRoutine Request<sub>[1\_RHMI\_271]</sub>

[1\_RHMI\_267] The RoutineControl – Start Routine ‘CloseRHMI Session’- Request message requests the Remote HMI session closing.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = CloseRHMI Session [byte#1(MSB), byte#2]	'F212'H

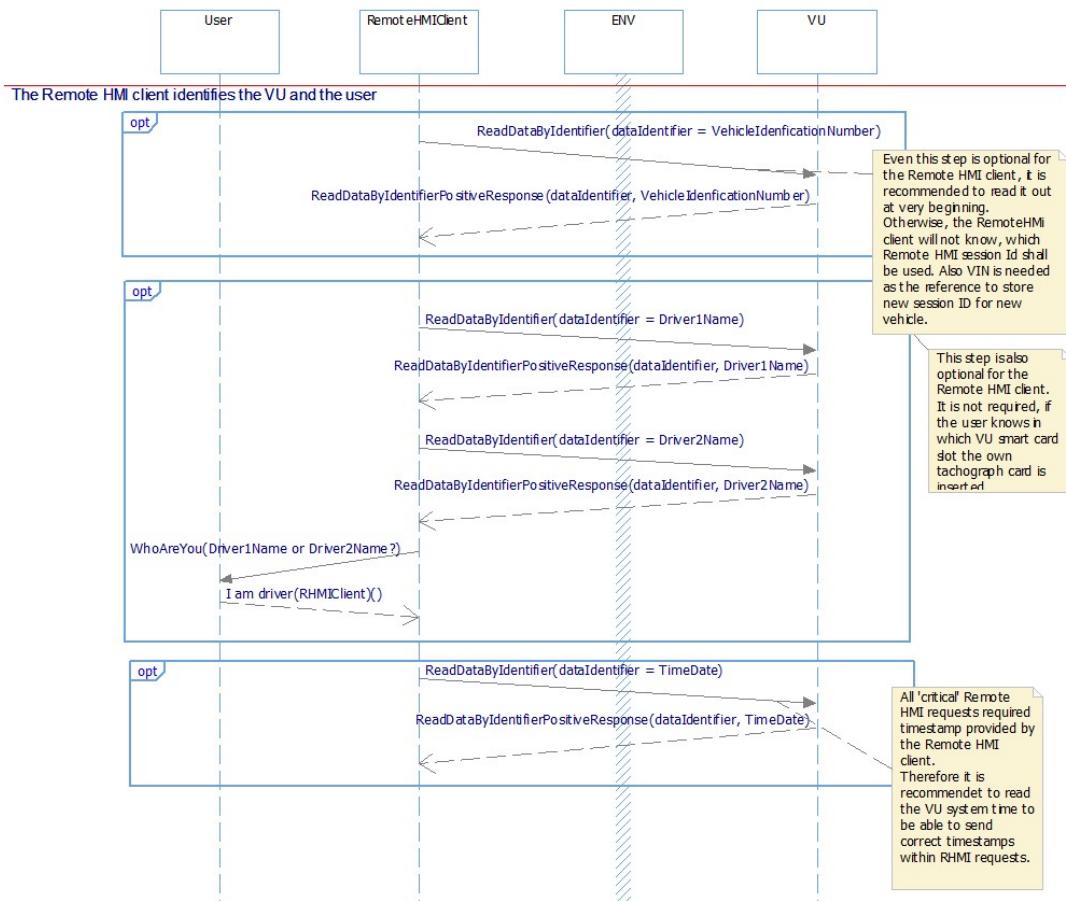
##### 4.2.2.2 Routine ‘CloseRHMI Session’, StartRoutine Positive Response<sub>[1\_RHMI\_273]</sub>

[1\_RHMI\_274] The RoutineControl – Start Routine ‘CloseRHMI Session’- Positive Response message confirms the Remote HMI session closing.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = CloseRHMI Session [byte#1(MSB), byte#2]	'F212'H

## 4.3 Remote HMI Identifier/ checksum/ remote HMI client pairing [1\_RHMI\_275]

[1\_RHMI\_222]



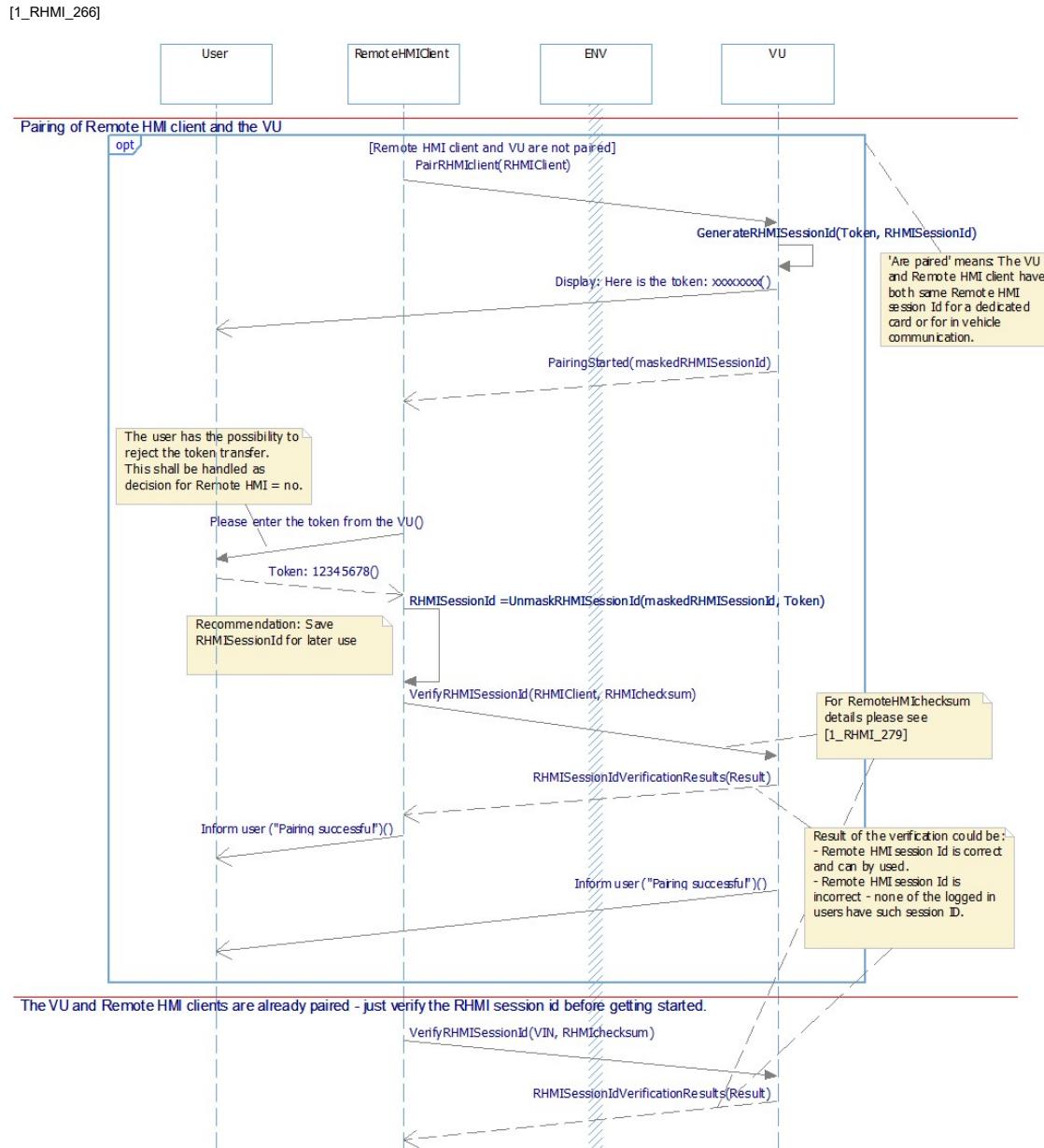
**Figure:** Some recommended initial sequence for the Remote HMI client.

For the Remote HMI handling, it is required that the Remote HMI client requests some general information from the VU:

- VIN: the ‘remote HMI session identifier’ for the Remote HMI is VU specific. I.e. if a Remote HMI client is paired to several VUs, then the client shall be able to identify which ‘remote HMI session identifier’ belongs to the VU.
- The slot of the inserted tachograph card: the ‘remote HMI session identifier’ is tachograph card specific. I.e. the Remote HMI client has to know in which VU smart card slot the tachograph card is inserted.
- VU time/date: The Remote HMI client shall verify if the VU current time/date is circa inline to the Remote HMI client time/date. Otherwise the VU will reject requests with incorrect timestamp.

**Note:** The VU may suppress the users’ private data like driver name if the private data is requested via ITS interface. For more information, see [Annex 1C V2], requirement 13.

## Smart Tachograph V2: VU, Remote Human Machine Interface Specification



**Figure:** Sequences ‘Remote HMI client paring’ and ‘RHMI session identifier verification’.

[1\_RHMI\_276] The VU shall verify the correctness of a Remote HMI request/entry (see [1\_RHMI\_224]) by verifying the Remote HMI checksum attached to such request/entry:

If the checksum attached to such request matches to entire request/entry (data + Remote HMI session identifier assigned to the driver, co-driver or to the in-vehicle RHMI client, see [1\_RHMI\_279]), then the VU shall accept such entry/request.

**Rationale:** see [1\_RHMI\_223], [1\_RHMI\_223a], [1\_RHMI\_223b] and [1\_RHMI\_228a].

Any of the RHMI clients can be shared by driver and co-driver to perform entries. The Remote HMI request does not contain any reference to a specific Remote HMI session identifier. Therefore, the VU has to verify the Remote HMI checksum correctness using all Remote HMI session identifiers (up to 3, see [1\_RHMI\_223], [1\_RHMI\_223a]).

[1\_RHMI\_279] **Definition:** The ‘Remote HMI checksum’ (**RHMICS**)

- is used to verify correctness of a Remote HMI requests. I.e. to verify
  - If the request is received from the paired Remote HMI client;
  - If the user of such Remote HMI client has his/her tachograph card inserted into the VU;
  - If request/entry is not modified.
- is calculated as CRC32(applied to entire RHMI request on application layer but without Remote HMI checksum itself ||\* RHMI sessionId);  
\*: ‘||’ means ‘concatenated’.
- the CRC32 is calculated with the polynomial as  
 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$  ('04C11DB7'H);
- input is reflected;
- output is reflected;
- the initial value shall be '00000000'H;
- the final result of the CRC32 shall not be combined/obfuscated additionally, i.e. XOR with '00000000'H.

**Example:**

Result RHMIchecksum = CRC32('31 01 F2 05 00 01 61 0A 85 E4 (set activity request) B3 3B 56 11 53 5A 8D B1 (RHMIsessionId)'H) = ‘3807F449’H;

[1\_RHMI\_278] **Definition:** A ‘Remote HMI session identifier’ (**RHMIsessionId**)

- is 8 bytes random number;
- is generated by the VU on request by a Remote HMI client;
- is specific to the Remote HMI user(card) which requests it;
- is used to calculate the Remote HMI checksum.

[1\_RHMI\_277] The VU shall store the ‘Remote HMI session identifier’ for each supported RHMI clients (see [1\_RHMI\_223]) and [1\_RHMI\_233]).

**Note:** Today, each of the stored driver decisions as defined by [Annex 1C V2], requirement 133, may have stored ‘Remote HMI session identifier’ + 1 separate for the in-vehicle client.

[1\_RHMI\_282] The Remote HMI client shall not publish the Remote HMI session identifier(s) outside of the Remote HMI client.

[1\_RHMI\_281] **Definition:** A ‘Remote HMI token’ (**RHMItoken**)

- is 4 bytes (8 digits) BCD random number;
- is generated together with the Remote HMI session identifier’ (see [1\_RHMI\_278]) by the VU;
- is used to obfuscate the Remote HMI session identifier’ during transport to the Remote HMI client;
- is displayed during pairing process at local HMI
- is entered by the user in the Remote HMI client.

[1\_RHMI\_286] The VU shall provide the possibility to a Remote HMI client to request new Remote HMI session identifier for such Remote HMI client (see [1\_RHMI\_278]).

[1\_RHMI\_285] **Definition:**

- Before the activation or during calibration mode: The ‘transmitted Remote HMI session Id’ is the plain Remote HMI session identifier, i.e.  
**TransmittedRHMIsessionId** = RHMIsessionId.
- Else: The ‘transmitted Remote HMI session Id’ is the Remote HMI session identifier which is obfuscated by the Remote HMI token as

**TransmittedRHMIsessionId** = XOR(RHMIsessionId[7-4], RHMItoken) ||\*  
XOR(RHMIsessionId[3-0], RHMItoken)

\*: ‘||’ means ‘concatenated’.

**Rationale:** to allow automatic, tool supported pairing between the VU and the Remote HMI client in a “trusted environment” like vehicle EOL or tachograph authorised workshops.

[1\_RHMI\_284] If the Remote HMI session identifier is requested, then the VU shall show the correspondent Remote HMI token only on the VU display.

**Exception:** See [1\_RHMI\_300].

## Smart Tachograph V2: VU, Remote Human Machine Interface Specification

[1\_RHMI\_283] The Remote HMI client shall restore the original Remote HMI session identifier from the obfuscated Remote HMI session identifier using the Remote HMI token entered by the user by the method as it was obfuscated in VU:

```
RHMIsessionId = XOR(TransmittedRHMIsessionId[7-4], RHMItoken) ||  
XOR(TransmittedRHMIsessionId[3-0], (RHMItoken))
```

Then the Remote HMI client should store such Remote HMI session identifier assigned to the VU (use Vehicle Identification Number as reference) and use it at the next time for RHMI interface on this VU.

[1\_RHMI\_290] After the activation and if not in calibration mode, the VU must not allow to read the Remote HMI session identifiers.

**Note:** This is valid only for driver cards. If a workshop card is inserted then the remote client can read out the plain, newly generated Remote HMI session identifier for this workshop session.

Before activation, see [1\_RHMI\_300].

[1\_RHMI\_289] If the Remote HMI checksum is incorrect, then the VU shall reject such Remote HMI request/entry for by negative response ‘invalidKey’.

[1\_RHMI\_288] In case of invalid Remote HMI checksum (see [1\_RHMI\_289]), the VU shall reject any further Remote HMI request for 10 s by negative response ‘requiredTimeDelayNotExpired’.

**Note:** In such case, it is recommended to pair the VU and the Remote HMI client again.

**Rationale:** to prevent ‘brute force’ attacks on Remote HMI session identifiers.

[1\_RHMI\_287] The VU shall provide the possibility to the user to reject the Remote HMI session identifier transport to the Remote HMI client.

The VU shall handle it as if such user decides not to have the Remote HMI available.

The VU shall store such user decision (see [1\_RHMI\_233]).

**Exception:** If the user rejects the Remote HMI session identifier transport for the in-vehicle Remote HMI client, then the VU must not change the stored user(s) decision about Remote HMI availability.

[1\_RHMI\_294] If the user decides not to have the Remote HMI available, then the VU shall delete the Remote HMI session identifier stored for such user.

[1\_RHMI\_293] The VU shall provide the possibility to verify a Remote HMI session identifier for an inserted tachograph card or for the in-vehicle Remote HMI client.

### 4.3.1 Remote HMI identifier for in-vehicle RHMI client<sub>[1\_RHMI\_291]</sub>

[1\_RHMI\_301] For in-vehicle Remote HMI client (e.g. secondary display or instrument cluster), it is recommended to provide the possibility to the user to request new Remote HMI session identifier.

[1\_RHMI\_300]

- Before activation or during calibration mode, the VU shall provide the possibility to read the Remote HMI session identifier as 'plain text'.
- Else, the VU shall provide the possibility to read the Remote HMI session identifier only obfuscated.

**Note:** By starting the 'normal' Remote HMI pairing procedure and read out of the Remote HMI Session identifier, the in-vehicle RHMI client is able to automatically pair with the VU without user interaction. Same also for the workshop members.

But this possibility is only available before activation.

**Rationale:** See the rationale for [1\_RHMI\_285].

[1\_RHMI\_299a] After activation, the VU shall accept the request for a new Remote HMI session identifier for the in-vehicle RHMI client only in calibration mode of operation.

**Rationale:** to reduce the risk that an unknown user can pair a Remote HMI client and pretend to be the in-vehicle client.

[1\_RHMI\_299b] If the VU is activated and is not in calibration mode of operation, and the VU receives a request for a new Remote HMI session identifier for the in-vehicle client (see [1\_RHMI\_299a]), then the VU shall reject the request by a negative response 'conditionsNotCorrect'.

### 4.3.2 Routine 'PairRHMIclient', StartRoutine Request<sub>[1\_RHMI\_302]</sub>

[1\_RHMI\_297] The RoutineControl – StartRoutine 'PairRHMIclient'- Request message requests the Remote HMI session identifier for specific tachograph card.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = PairRHMIclient [byte#1(MSB), byte#2]	'F213'H
5	routineControlOption[byte#1] = RHMIClient (see [1_RHMI_307])	'XX'H

#### 4.3.2.1 Parameter RHMIClient<sub>[1\_RHMI\_306]</sub>

[1\_RHMI\_307] The RHMIClient provides the information, for which Remote HMI client the Remote HMI session identifier is requested / is provided (i.e. shall be assigned and stored).

RHMIClient ::= INTEGER(0..255)

**Value assignment:**

- 0: in-vehicle RHMI client;
- 1: RHMI client assigned to the tachograph card inserted into VU smart card slot 1;
- 2: RHMI client assigned to the tachograph card inserted into VU smart card slot 2;
- 3..255: reserved for future use.

### 4.3.3 Routine ‘PairRHMIclient’, StartRoutine Positive Response<sub>[1\_RHMI\_303]</sub>

[1\_RHMI\_304] The RoutineControl – StartRoutine ‘PairRHMIclient’ Positive Response message provides the Remote HMI session identifier for the specific RHMI client.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = PairRHMIclient [byte#1(MSB), byte#2]	'F213'H
5	routineStatusRecord[byte#1] = RHMIClient (see [1_RHMI_307], echo from the request)	'XX'H
6	routineStatusRecord[byte#2] = TransmittedRHMIsessionIdTransportType (see [1_RHMI_296])	'XX'
7..14	routineStatusRecord[byte#3..byte#10] = TransmittedRHMIsessionId (see [1_RHMI_285])	'XX...XX' H
15..18	routineStatusRecord[byte#11..byte#14] = TransportChecksum (see [1_RHMI_309])	'XX...XX' H

#### 4.3.3.1 Parameter TransmittedRHMIsessionIdTransportType<sub>[1\_RHMI\_305]</sub>

[1\_RHMI\_296] The TransmittedRHMIsessionIdTransportType provides the information, if the Remote HMI session Id is obfuscated by Remote HMI token or as ‘plain text’:

- ‘00’H: The TransmittedRHMIsessionId contains ‘plain text’ Remote HMI session Id.
- ‘01’H: The TransmittedRHMIsessionId contains Remote HMI session Id obfuscated by Remote HMI token.

#### 4.3.3.2 Parameter TransportChecksum<sub>[1\_RHMI\_308]</sub>

[1\_RHMI\_309] The TransportChecksum is calculated as CRC32 (RemoteHMIsessionId ||\* RemoteHMItoken).

CRC32 shall use the same polynomial as for the Remote HMI checksum.

\*: ‘||’ means ‘concatenated’.

The send order of TransportChecksum is ‘MSB first to LSB last’.

**Note:** If the Remote HMI session identifier is obfuscated, then it is recommended that the Remote HMI client should verify the TransportChecksum for validity before using the Remote HMI session identifier.

#### 4.3.4 Routine ‘PairRHMIclient’, StartRoutine Negative Response<sub>[1\_RHMI\_310]</sub>

[1\_RHMI\_311] This definition contains the negative response conditions for routine PairRHMIclient, sub-function StartRoutine, which are additional to the standard negative responses for the RoutineControl service.

Response Name	Value	Description
conditionsNotCorrect	'22'H	<p>The conditions for execution are not met:</p> <ul style="list-style-type: none"> <li>• There is no tachograph card inserted into the VU as requested in RHMIClient.</li> <li>• A remote HMI pairing for the in-vehicle client is requested but the VU is not in calibration mode of operation (see [1_RHMI_299a]).</li> </ul>
busyRepeatRequest	'21'H	<p>No (new) remote entry is possible at the moment:</p> <ul style="list-style-type: none"> <li>• Other ‘Pair Remote HMI client’ request is pending – wait for transfer of the Remote HMI token by the user.</li> </ul>

#### 4.3.5 Local HMI screens and user interaction for Remote HMI client pairing<sub>[1\_RHMI\_312]</sub>

[1\_RHMI\_316] The VU shall show on VU display,

- the Remote HMI token,
- the affected smart card slot (according to the received RHMIClient (see [1\_RHMI\_304])).

In case of the Remote HMI in-vehicle client, the card slot number shall be omitted.

See [1\_RHMI\_284] for more information.

#### 4.3.6 Routine ‘VerifyRHMISessionId’, StartRoutine Request<sub>[1\_RHMI\_323]</sub>

[1\_RHMI\_320] The RoutineControl – StartRoutine ‘VerifyRHMISessionId’ – Request message requests the verification of the Remote HMI session identifier for a specific Remote HMI client.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = VerifyRHMI sessionId [byte#1(MSB), byte#2]	'F214'H
5	routineControlOption[byte#1] = RHMIClient (see [1_RHMI_307])	'XX'H
6..9	routineControlOption[byte#2..byte#5] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
10..13	routineControlOption[byte#6..byte#9] = RHMICS (see [1_RHMI_279]).	XX...XX' H

#### 4.3.7 Routine ‘VerifyRHMI sessionId’, StartRoutine Positive Response<sub>[1\_RHMI\_324]</sub>

[1\_RHMI\_319] If the Remote HMI checksum within the request is correct (and therefore very likely the Remote HMI session identifier) then the VU shall inform the RHMI client by RoutineControl – StartRoutine ‘VerifyRHMI sessionId’ Positive Response message.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = VerifyRHMI sessionId [byte#1(MSB), byte#2]	'F214'H
5	routineStatusRecord[byte#1] = RHMIClient (see [1_RHMI_307], echo from the request)	'XX'H

#### 4.3.8 Routine ‘VerifyRHMI sessionId’, StartRoutine Negative Response<sub>[1\_RHMI\_325]</sub>

[1\_RHMI\_318] This definition contains the negative response conditions for routine VerifyRHMI sessionId, sub-function StartRoutine, which are additional to the standard negative responses for the service RoutineControl.

Response Name	Value	Description
conditionsNotCorrect	'22'H	The conditions for execution are not met: <ul style="list-style-type: none"> <li>The VU local HMI is currently used.</li> <li>There is no tachograph card inserted into the VU as requested in RHMIClient.</li> </ul>
requestOutOfRange	'31'H	The request timestamp is outside of expected range. See [1_RHMI_226].

invalidKey	35	The Remote HMI checksum is incorrect (see [1_RHMI_289]). <b>Note:</b> The security as defined by VU PP is not involved here. •
requiredTimeDelayNotExpired	37	The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288]).

## 4.4 Remote set of driver activity<sub>[1\_RHMI\_326]</sub>

[1\_RHMI\_317] The VU shall provide the possibility to set a driver activity via RoutineControl service, routine SetActivity.

[1\_RHMI\_313] If the VU is not activated, then the VU shall reject the ‘RoutineControl – Request Routine ‘SetActivity’ requests by negative response ‘conditionsNotCorrect’.

[1\_RHMI\_331] If the Remote HMI session is closed, then the VU shall reject the ‘RoutineControl – Request Routine ‘SetActivity’ requests by negative response ‘conditionsNotCorrect’.

### 4.4.1 Routine ‘SetActivity’, StartRoutine Request<sub>[1\_RHMI\_332]</sub>

[1\_RHMI\_329] The RoutineControl – Start Routine ‘SetActivity’ – Request Message provides the possibility to set driver activity for the driver corresponding to a specific smart card slot.

Byte#	Parameter name	Value
1	serviced = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = ‘SetActivity’ [byte#1(MSB), byte#2]	'F205'H
5	routineControlOption[byte#1] = SmartCardSlot (see [1_RHMI_53])	'XX'H
6	routineControlOption[byte#2] = ActivityType (see [1_RHMI_328])	'XX'H
7..10	routineControlOption[byte#3..byte#6] = RHMItimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
11..14	routineControlOption[byte#7..byte#10] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

Table: Routine ‘SetActivity’, StartRoutine Request.

#### 4.4.1.1 Parameter ActivityType<sub>[1\_RHMI\_328a]</sub>

<sub>[1\_RHMI\_328]</sub>The ActivityType defines the drivers activity to set:

- '00'H: BREAK/REST;
- '01'H: AVAILABILITY;
- '02'H: WORK.

#### 4.4.2 Routine ‘SetActivity’, StartRoutine Positive Response<sub>[1\_RHMI\_327]</sub>

<sub>[1\_RHMI\_333]</sub>The RoutineControl – Start Routine ‘SetActivity’ – Positive Response Message provides confirmation, if the set of the driver activity is accepted.

**Note:** The set of the driver activity happens in real-time, i.e. it will be handled as if the activity is changed at the same time via local HMI. The RHMItimestamp has not impact on the time, when the activity change is performed.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = SetActivity [byte#1(MSB), byte#2]	'F205'H

Table: Routine ‘SetActivity’, StartRoutine Positive Response.

#### 4.4.3 Routine ‘SetActivity’, Negative Response code<sub>[1\_RHMI\_337]</sub>

<sub>[1\_RHMI\_334]</sub>This definition contains the negative response conditions for routine SetActivity, which are additional to the standard negative responses for the service RoutineControl.

Response Name	Value	Description
conditionsNotCorrect	'22'H	<p>The conditions for execution are not met:</p> <ul style="list-style-type: none"> <li>• The Remote HMI session is closed.</li> <li>• The VU is not activated;</li> <li>• A card insertion or withdrawal procedure is ongoing;</li> <li>• Any other reason, when a activity set is not possible or not allowed.</li> </ul>
requestOutOfRange	'31'H	<ul style="list-style-type: none"> <li>• The request timestamp is outside of expected range. See[1_RHMI_226].</li> <li>• At least one parameter for the routine ‘SetActivity’ is out of range.</li> </ul>

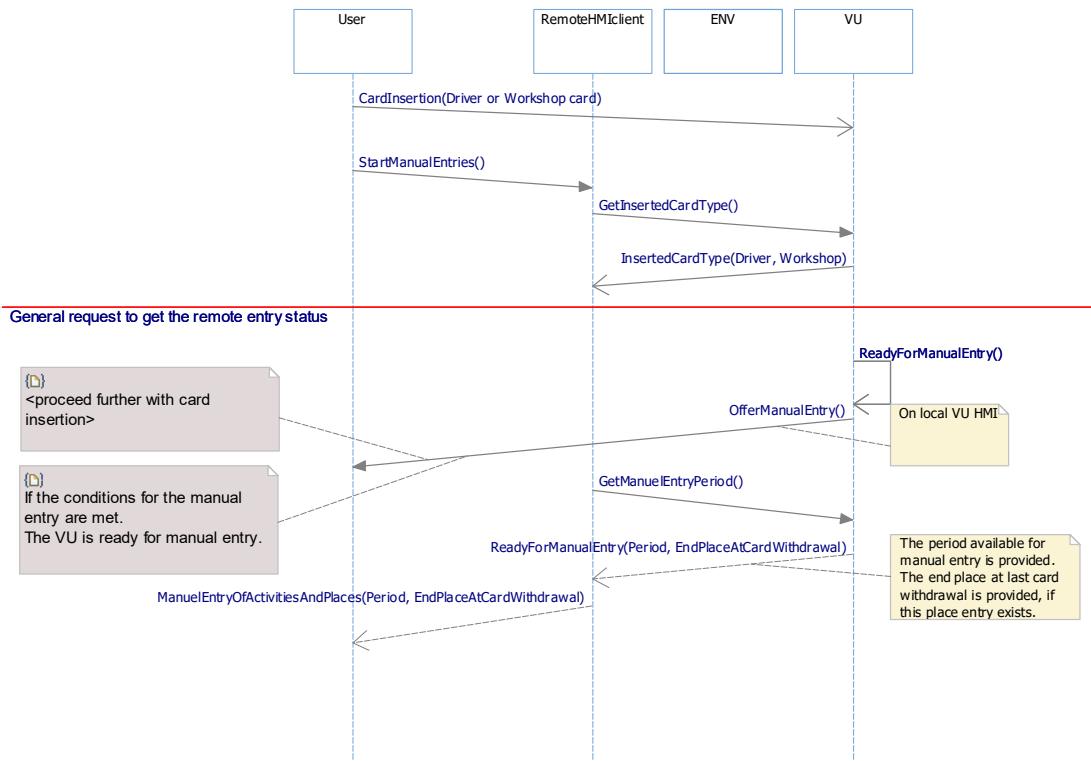
Response Name	Value	Description
invalidKey	'35'H	The Remote HMI checksum is incorrect (see [1_RHMI_289]). <b>Note:</b> The security as defined by VU PP is not involved here.
requiredTimeDelayNotExpired	'37'H	The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288])

## 4.5 Remote manual entries during driver card insertion procedure<sub>[1\_RHMI\_2]</sub>

[1\_RHMI\_3] The VU shall provide the possibility during driver / co-driver card insertion procedure to enter driver manual entries via RoutineControl service (see [Annex 1C V2], Appendix 8).

**Note:** For more information, see the requirements in [Annex 1C V2], in the chapter 3.6.2 'Manual entry of driver activities and driver consent for ITS interface'.

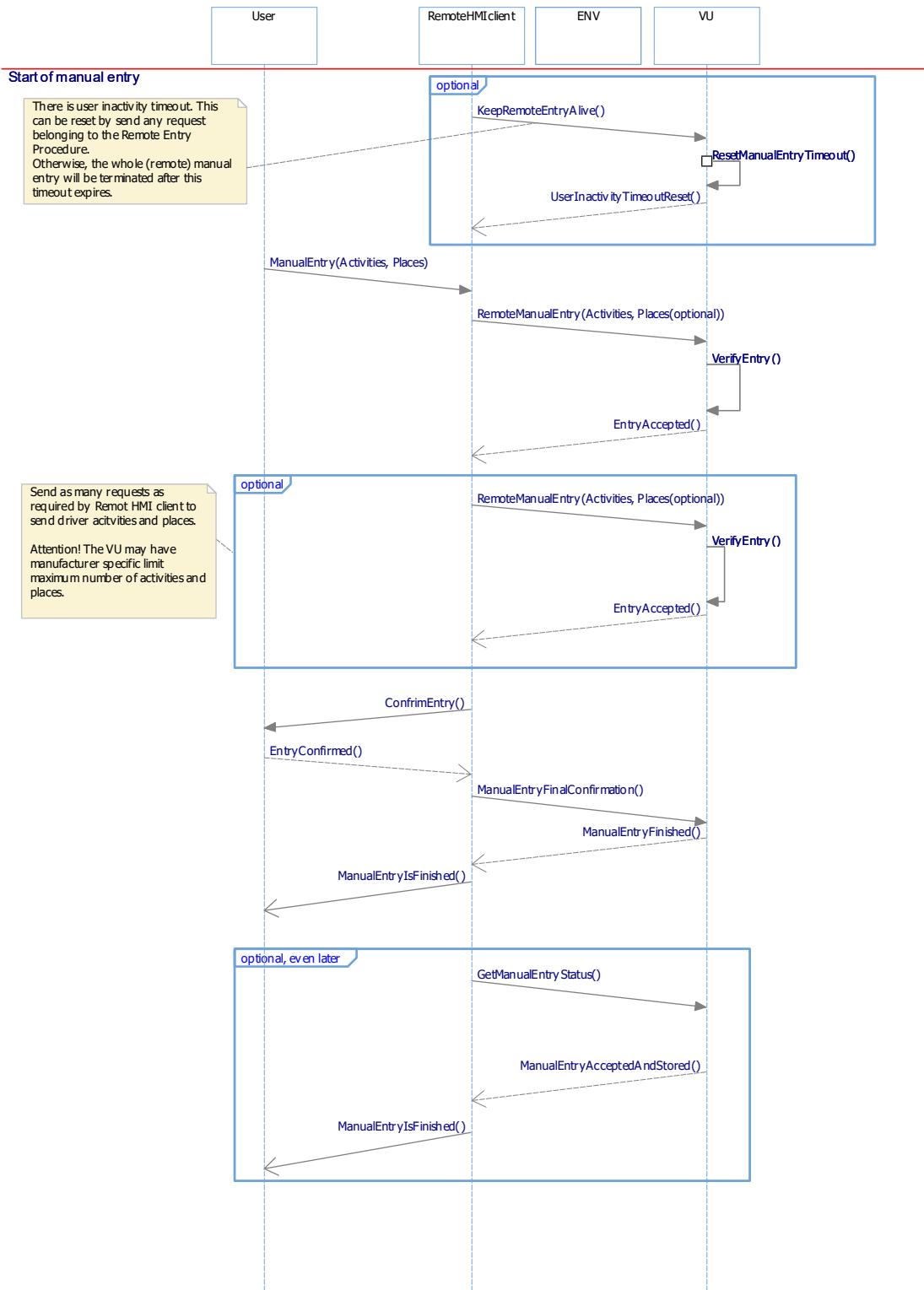
[1\_RHMI\_204]



**Figure:** Remote manual entry of driver activities and places – ‘happy day’ – driver/workshop card insertion.

[1\_RHMI\_4]

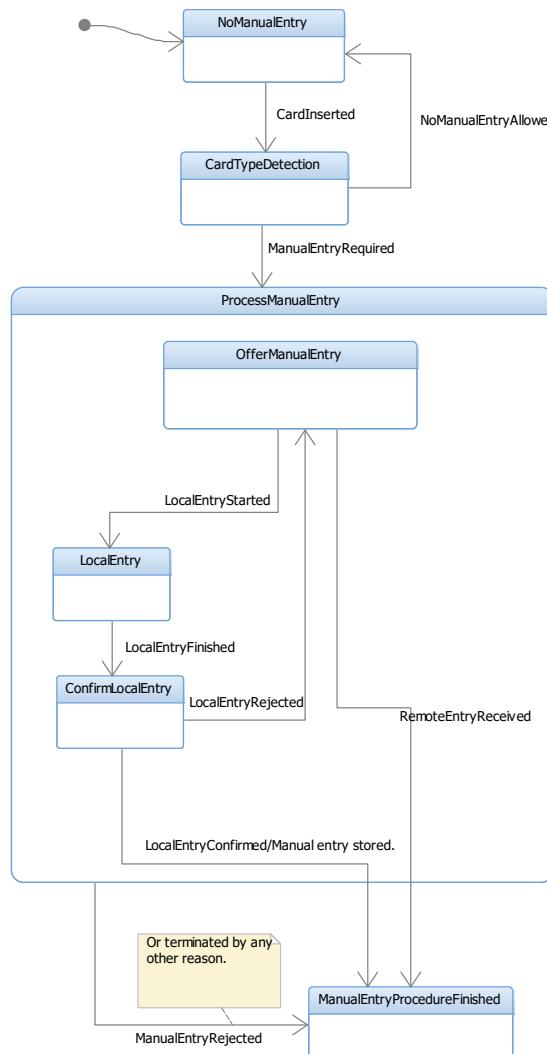
# Smart Tachograph V2: VU, Remote Human Machine Interface Specification



**Figure:** Remote manual entry of driver activities and places – ‘happy day’ – remote manual entry.

## Smart Tachograph V2: VU, Remote Human Machine Interface Specification

[1\_RHMI\_206]



**Figure:** Remote manual entry states.

### 4.5.1 Routine ‘Manual entry of driver activities and places’.[1\_RHMI\_12]

[1\_RHMI\_13] If the VU is not activated, then the VU shall reject the ‘RoutineControl – Request Routine ‘ManualEntryOfDriverActivitiesAndPlaces’’ requests by appropriate negative response.

[1\_RHMI\_19] If the VU respond with RoutineControl – Request Routine ‘ManualEntryOfDriverActivitiesAndPlaces’ Results – Positive Response Message, then the VU shall reset the user inactivity timeout.

#### 4.5.1.1 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetInsertedCardType, RequestRoutineResults Request<sub>[1\_RHMI\_14]</sub>

[1\_RHMI\_18] The request for the Routine ‘ManualEntryOfDriverActivitiesAndPlaces’ results, with the option GetInsertedCardType, requests the type of the inserted tachograph card within appropriate smart card slot.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineControlOption[byte#1] = GetInsertedCardType	'01'H
6	routineControlOption[byte#2] = SmartCardSlot (see [1_RHMI_16]).	'XX'H
7..10	routineControlOption[byte#3..byte#6] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
11..14	routineControlOption[byte#7..byte#10] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, RequestRoutineResults, GetInsertedCardType Request.

##### 4.5.1.1.1 Parameter SmartCardSlot <sub>[1\_RHMI\_16a]</sub>

[1\_RHMI\_16] The SmartCardSlot defines the affected VU smart card slot – see data type CardSlotNumber, [Annex 1C V2], Appendix 1.

#### 4.5.1.2 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetInsertedCardType, RequestRoutineResults Positive Response<sub>[1\_RHMI\_17]</sub>

[1\_RHMI\_18] The Positive Response on the RoutineControl, Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, RequestRoutineResults with option GetInsertedCardType provides the status for the driver/workshop card insertion.

Byte#	Parameter name	Value
1	serviced = RoutineControlPositiveResponse	'71'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineStatusRecord[byte#1] = GetInsertedCardType	'01'H

Byte#	Parameter name	Value
6	routineControlOption[byte#2] = SmartCardSlot (see [1_RHMI_16]).	'XX'H
7	routineStatusRecord[byte#2] = routineInfo (see [1_RHMI_20])	'XX'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetInsertedCardType, RequestRoutineResults Positive Response

#### 4.5.1.2.1 Parameter routineInfo<sub>[1\_RHMI\_20a]</sub>

[1\_RHMI\_20]

Response Name	Value	Definition
DriverCardIsInserted	'01'H	A driver card is inserted into the appropriate smart card slot.
WorkshopCardIsInserted	'02'H	A workshop card is inserted into the appropriate smart card slot. <b>Note:</b> A workshop card requires a PIN entry. The workshop card PIN can be entered via local VU HMI or sent to the VU via diagnostic interface.

**Table:** Parameter routineInfo for the routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetInsertedCardType.

#### 4.5.1.3 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryPeriod, RequestRoutineResults Request<sub>[1\_RHMI\_187]</sub>

[1\_RHMI\_182] The request for the Routine ‘ManualEntryOfDriverActivitiesAndPlaces’ results, with the option GetManualEntryPeriod, requests the period available for manual entry for the driver/workshop card inserted into appropriate smart card slot.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineControlOption[byte#1] = GetManualEntryPeriod	'02'H
6	routineControlOption[byte#2] = SmartCardSlot (see [1_RHMI_16]).	'XX'H
7..10	routineControlOption[byte#3..byte#6] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
11..14	routineControlOption[byte#7..byte#10] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryPeriod, RequestRoutineResults Request.

[1\_RHMI\_183] If the VU is not ready to provide the period available for manual entry for the driver/workshop card inserted into appropriate smart card slot, then the VU shall reject the request for the request for the period available for manual entry by appropriate negative response.

**4.5.1.4 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’,  
GetManualEntryPeriod, RequestRoutineResults Positive  
Response** [1\_RHMI\_192]

[1\_RHMI\_191] The Positive Response on the RoutineControl, Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, RequestRoutineResults with option GetManualEntryPeriod provides the period available for the manual entry of drivers acitivities and places.

Byte#	Parameter name	Value
1	serviced = RoutineControlPositiveResponse	'71'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineStatusRecord[byte#1] = GetManualEntryPeriod	'02'H
6	routineStatusRecord[byte#2] = routineInfo (see [1_RHMI_189])	'XX'H
7..10	routineStatusRecord[byte#3.. byte#6] = ManualEntryPeriodBegin[byte#1(MSB)..byte#4] (see [1_RHMI_21])	'XX'H ... 'XX'H
11..14	routineStatusRecord[byte#7.. byte#10] = ManualEntryPeriodEnd[byte#1(MSB)..byte#4] (see [1_RHMI_21])	'XX'H ... 'XX'H
15	routineStatusRecord[byte#11] = EndCountryAtCardWithdrawal (see [1_RHMI_197])	'XX'H
16	routineStatusRecord[byte#12] = EndRegionAtCardWithdrawal (see [1_RHMI_197])	'XX'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryPeriod, RequestRoutineResults Positive Response.

**4.5.1.4.1 Parameter routineInfo** [1\_RHMI\_189a]

[1\_RHMI\_189]

Response Name	Value	Definition
ReadyForManualEntryOfActivitiesAndPlaces	'10'H	The VU provides the possibility to enter manually the activities and places for the appropriate smart card slot.

**Table:** Parameter routineInfo for the routine  
'ManualEntryOfDriverActivitiesAndPlaces', GetManualEntryPeriod.

#### 4.5.1.4.2 Parameters ManualEntryPeriodBegin and ManualEntryPeriodEnd<sub>[1\_RHMI\_21a]</sub>

[1\_RHMI\_21] The parameters ManualEntryPeriodBegin and ManualEntryPeriodEnd provide the 'period of the last card withdrawal – current insertion' according to [Annex 1C V2] requirement 61, both as data type TimeReal according to [Annex 1C V2], Appendix 1, UTC time.

**Note:** It is recommended that Remote HMI client requests this period to ensure that the driver activities and places within ManualEntryActivitiesAndPlaces (see [1\_RHMI\_22]) are within the allowed period according to [Annex 1C V2].

#### 4.5.1.4.3 Parameter EndCountryAtCardWithdrawal and EndRegionAtCardWithdrawal<sub>[1\_RHMI\_197a]</sub>

[1\_RHMI\_197] The parameters EndCountryAtCardWithdrawal and EndRegionAtCardWithdrawal provide the end place of the driver entered at the last card withdrawal.

If not such end place entered at the last card withdrawal, then EndCountryAtCardWithdrawal and EndRegionAtCardWithdrawal shall be set to 'no information available'.

The EndCountryAtCardWithdrawal has the data type NationNumeric (see [Annex 1C V2], Appendix 1).

The EndRegionAtCardWithdrawal has the data type RegionNumeric (see [Annex 1C V2], Appendix 1).

#### 4.5.1.5 Routine 'ManualEntryOfDriverActivitiesAndPlaces', GetManualEntryStatus, RequestRoutineResults Request<sub>[1\_RHMI\_188]</sub>

[1\_RHMI\_184] The request for the Routine 'ManualEntryOfDriverActivitiesAndPlaces' results, with the option GetManualEntryStatus, requests the status of the remote manual entries/places by the user for the driver/workshop card inserted into appropriate smart card slot.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = RequestRoutineResults	'03'H

Byte#	Parameter name	Value
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineControlOption[byte#1] = GetManualEntryStatus	'03'H
6	routineControlOption[byte#2] = SmartCardSlot (see [1_RHMI_16]).	'XX'H
7..10	routineControlOption[byte#3..byte#6] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
11..14	routineControlOption[byte#7..byte#10] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, RequestRoutineResults, GetManualEntryStatus Request.

[1\_RHMI\_185] If the VU has not received consistent remote activities and places before, then the VU shall reject the request for status of the remote manual entries/places by appropriate negative response.

#### 4.5.1.6 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryStatus, RequestRoutineResults Positive Response [1\_RHMI\_193]

[1\_RHMI\_194] The Positive Response on the RoutineControl, Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, RequestRoutineResults with option GetManualEntryStatus provides the status of the remote manual entries/places by the user for the driver/workshop card inserted into appropriate smart card slot.

Byte#	Parameter name	Value
1	serviced = RoutineControlPositiveResponse	'71'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineStatusRecord[byte#1] = GetManualEntryStatus	'03'H
6	routineStatusRecord[byte#2] = routineInfo (see [1_RHMI_190])	'XX'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, GetManualEntryStatus, RequestRoutineResults Positive Response.

##### 4.5.1.6.1 Parameter routineInfo [1\_RHMI\_190a]

[1\_RHMI\_190]

Response Name	Value	Definition
ManualEntryAcceptedAndStored	'23'H	The manual entry is stored. The manual entry procedure ends successfully.

**Table:** Parameter routineInfo for the routine  
'ManualEntryOfDriverActivitiesAndPlaces', GetManualEntryStatus.

#### 4.5.1.7 Routine 'ManualEntryOfDriverActivitiesAndPlaces', StartRoutine Request<sub>[1\_RHMI\_193a]</sub>

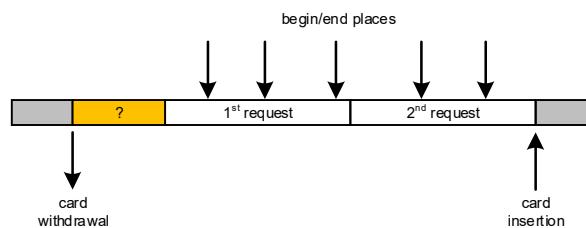
[1\_RHMI\_22] The RoutineControl – Start Routine 'ManualEntryOfDriverActivitiesAndPlaces' – Request Message provides the driver manual entry of the activities and places.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineControlOption[byte#1] = SmartCardSlot (see [1_RHMI_16])	'XX'H
6..n	routineControlOption[byte#2..byte#m] = ManualEntryActivitiesAndPlaces (see [1_DaDi_3209]).	'XX'H ... 'XX'H
n+1 .. n+4	routineControlOption[byte#m+1..byte#m+4] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
n+5 .. n+10	routineControlOption[byte#m+5..byte#m+8] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine 'ManualEntryOfDriverActivitiesAndPlaces', StartRoutine Request.

[1\_RHMI\_22a] The VU shall provide the possibility to receive and process several 'ManualEntryOfDriverActivitiesAndPlaces', StartRoutine requests.

In this case, the VU shall continue the newest activity change from already accepted most recent ManualEntryOfDriverActivitiesAndPlaces StartRoutine request until the oldest activity change from the current request.



**Figure:** example of the transmission of the ‘driver manual entry of the activities and places’ distributed over several ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine requests.

[1\_RHMI\_22b] If driver activities and places don’t fit into a single ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine request, then the Remote HMI client shall send the remaining activities and places via additional ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine request(s).

By doing this, the Remote HMI client shall transmit the activities and places, started from older to the newest, without overlapping of activities between the requests.

**Note:** The VU may have manufacturer specific limit for maximum number of activities and places.

[1\_RHMI\_22c] **Note:** all times in ManualEntryActivitiesAndPlaces are in UTC time.

#### 4.5.1.7.1 Parameter RHMITimestamp [1\_RHMI\_339a]

[1\_RHMI\_339] The RHMITimestamp provides the UTC timestamp of the request as data type TimeReal according to [Annex 1C V2].

#### 4.5.1.8 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine Positive Response [1\_RHMI\_23]

[1\_RHMI\_24] The RoutineControl – Start Routine ‘ManualEntryOfDriverActivitiesAndPlaces’ – Positive Response Message provides the confirmation, that the transmitted drivers’ activities and places are successfully received, passed the consistency check and stored by the VU.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine Positive Response.

[1\_RHMI\_186] If the VU is not ready to receive the driver manual entry of the activities and places for the appropriate smart card slot, then the VU shall reject the reception of the driver manual entry of the activities and places by appropriate negative response.

[1\_RHMI\_25] If the VU respond with RoutineControl – Start Routine ‘ManualEntryOfDriverActivitiesAndPlaces’ – Positive Response Message, then the VU shall reset the local VU HMI user inactivity timeout.

[1\_RHMI\_26] If the order of the driver activity changes within ManualEntryActivitiesAndPlaces is incorrect, then the VU shall reject such request by appropriate negative response.

[1\_RHMI\_196] If the order of the driver place entries within ManualEntryActivitiesAndPlaces is incorrect, or more than one begin place or more than one end place within same calendar minute is within ManualEntryActivitiesAndPlaces, then the VU shall reject such request by appropriate negative response.

[1\_RHMI\_209] If a driver place entry within ManualEntryActivitiesAndPlaces is older or newer as the period ‘ManualEntryPeriodBegin to ManualEntryPeriodEnd’ (see [1\_RHMI\_191]), then the VU shall reject such request by appropriate negative response.

[1\_RHMI\_210] If a driver activity change within ManualEntryActivitiesAndPlaces is older as ManualEntryPeriodBegin, or it is equal or newer as ManualEntryPeriodEnd, then the VU shall reject such request by appropriate negative response.

[1\_RHMI\_218] **Exception** to [1\_RHMI\_196] and [1\_RHMI\_210]: The VU must not accept the ‘end place’ for the time of the current card insertion.

[1\_RHMI\_210a] If a driver activity change or a place within received ManualEntryActivitiesAndPlaces is older or same as the most recent driver activity change / place with the previous ‘ManualEntryOfDriverActivitiesAndPlaces’, StartRoutine request, then the VU shall reject such request by appropriate negative response.

**Rationale:** avoid overlapping of the activity and places across the several requests.

#### 4.5.1.9 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StopRoutine Request [1\_RHMI\_23a]

[1\_RHMI\_218a] The RoutineControl – Stop Routine ‘ManualEntryOfDriverActivitiesAndPlaces’ – Request Message provides the final confirmation of already transmitted (see [1\_RHMI\_24]) driver activities and places.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StopRoutine	'02'H

Byte#	Parameter name	Value
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H
5	routineControlOption[byte#1] = SmartCardSlot (see [1_RHMI_16])	'XX'H
6..9	routineControlOption[byte#2..byte#5] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
10..13	routineControlOption[byte#6..byte#9] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StopRoutine Request.

#### 4.5.1.10 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StopRoutine Positive Response [1\_RHMI\_23b]

[1\_RHMI\_218b] The RoutineControl – Stop Routine

‘ManualEntryOfDriverActivitiesAndPlaces’ – Positive Response Message provides the confirmation, that the drivers’ manual entry of activities and places is finished.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StopRoutine	'02'H
3, 4	routineIdentifier[] = ManualEntryOfDriverActivitiesAndPlaces [byte#1(MSB), byte#2]	'F200'H

**Table:** Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, StopRoutine Positive Response.

[1\_RHMI\_218c] After the drivers’ manual entry of activities and places is finished for a driver/workshop card, then the VU shall reject further RoutineControl, Start Routine ‘ManualEntryOfDriverActivitiesAndPlaces’ requests for the card by appropriate negative response.

#### 4.5.1.11 Routine ‘ManualEntryOfDriverActivitiesAndPlaces’, negative response code [1\_RHMI\_27]

[1\_RHMI\_28] This definition contains the negative response conditions for routine ‘ManualEntryOfDriverActivitiesAndPlaces’, which are additional to the standard negative responses defined for the service RoutineControl.

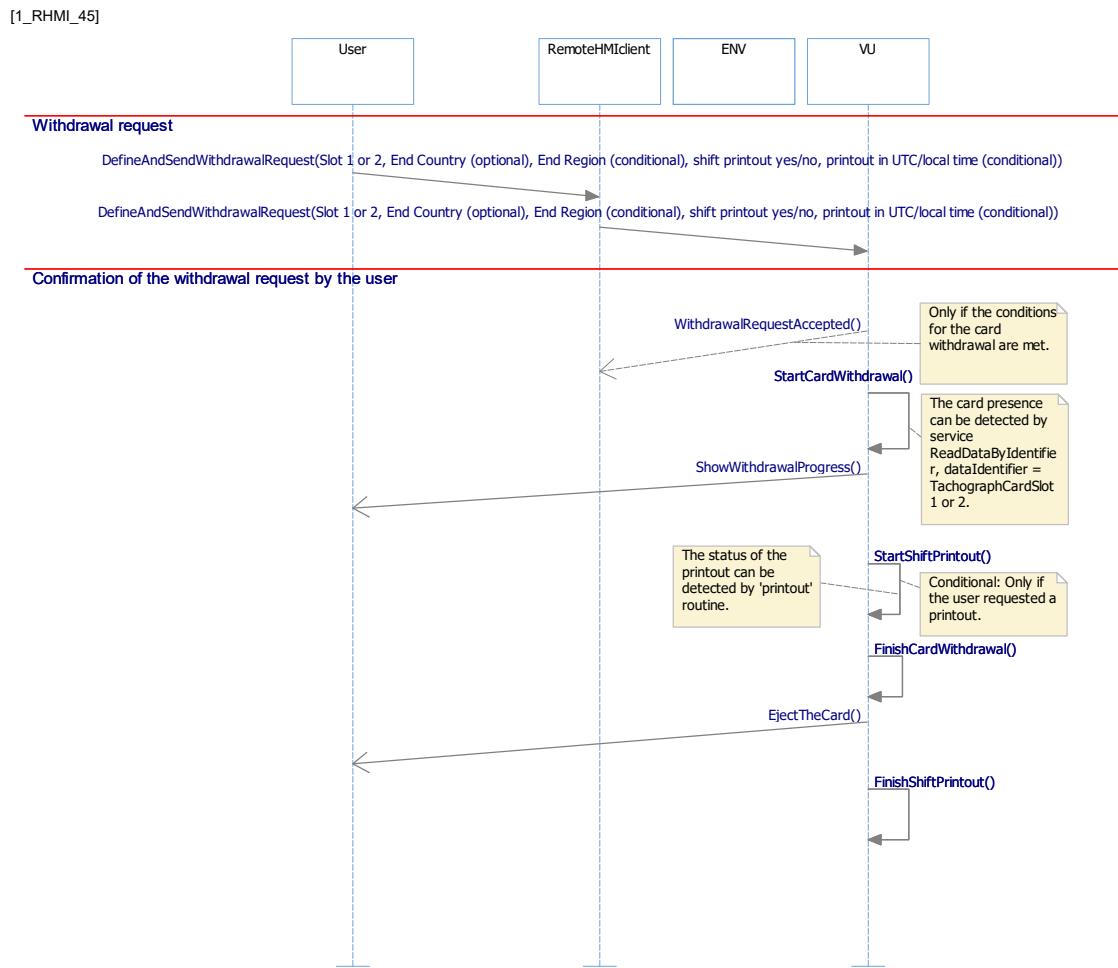
Response Name	Value	Description
conditionsNotCorrect	'22'H	The conditions for execution remote manual entry are not met:

		<ul style="list-style-type: none"> <li>• The Remote HMI session is closed;</li> <li>• The VU is not activated;</li> <li>• No driver/workshop card is inserted into requested smart card slot;</li> <li>• The manual entry is not possible;</li> <li>• The manual entry is started or already performed via local HMI;</li> <li>• Any other reason, when a remote manual entry is not possible or not allowed.</li> </ul>
busyRepeatRequest	'21'H	<p>No (new) remote entry is possible at the moment:</p> <ul style="list-style-type: none"> <li>• The VU is still busy processing the inserted smart card.</li> </ul>
requestOutOfRange	'31'H	<ul style="list-style-type: none"> <li>• At least one parameter for the routine 'ManualEntryOfDriverActivitiesAndPlaces' is out of range.</li> <li>• At least one of activities or places within remote manual entry is outside of the allowed period (see [1_RHMI_209] and [1_RHMI_210]).</li> <li>• At least one of activities or places within remote manual entry is inconsistent to other entries; see [1_RHMI_26], [1_RHMI_196] and [1_RHMI_210a].</li> <li>• The request timestamp is outside of expected range. See [1_RHMI_339]</li> </ul>
invalidKey	'35'H	<p>The Remote HMI checksum is incorrect (see [1_RHMI_289]).</p> <p><b>Note:</b> The security as defined by VU PP is not involved here.</p>
requiredTimeDelayNotExpired	'37'H	The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288]).
requestCorrectlyReceived-ResponsePending	'78'H	The VU verifies the received activities and places for consistency, please wait.
		<b>Note:</b> last row.

**Table:** Routine 'ManualEntryOfDriverActivitiesAndPlaces', negative response code.

## 4.6 Remote card withdrawal request<sub>[1\_RHMI\_43]</sub>

<sub>[1\_RHMI\_44]</sub>The VU shall provide the possibility to request the withdrawal of a driver or workshop card via RoutineControl service.



**Figure:** Sequence of the remote driver/workshop card withdrawal request, happy day scenario.

#### 4.6.1 Routine ‘Request driver/workshop card withdrawal’ [1\_RHMI\_49]

[1\_RHMI\_50] If the VU is not activated, then the VU shall reject the ‘RoutineControl – Request Routine DriverWorkshopCardWithdrawal requests by appropriate negative response.

##### 4.6.1.1 Routine ‘DriverWorkshopCardWithdrawal’, Start Routine Request [1\_RHMI\_51]

[1\_RHMI\_52] The RoutineControl – Start Routine ‘DriverWorkshopCardWithdrawal’ – Request Message requests the driver/workshop card withdrawal for a specific smart card slot and provides for such withdrawal procedure required parameters.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
3, 4	routineIdentifier[] = DriverWorkshopCardWithdrawal [byte#1(MSB), byte#2]	'F201'H
5	routineControlOption[byte#1] = SmartCardSlot (see [1_RHMI_53])	'XX'H
6	routineControlOption[byte#2] = PlaceCountry (see [1_RHMI_54])	'XX'H
7	routineControlOption[byte#3] = PlaceRegion (see [1_RHMI_54])	'XX'H
8	routineControlOption[byte#4] = CurrentShiftPrintoutRequest (see [1_RHMI_55])	'XX'H
9	routineControlOption[byte#5] = TechnicalDataPrintoutRequest (see [1_RHMI_55a])	'XX'H
10..13	routineControlOption[byte#6..byte#9] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
14..17	routineControlOption[byte#10..byte#13] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘DriverWorkshopCardWithdrawal’, Start Routine Request.

#### 4.6.1.1.1 Parameter SmartCardSlot<sub>[1\_RHMI\_51a]</sub>

<sub>[1\_RHMI\_53]</sub>The SmartCardSlot defines the affected VU smart card slot – see data type CardSlotNumber, [Annex 1C V2], Appendix 1.

#### 4.6.1.1.2 Parameters PlaceCountry and PlaceRegion<sub>[1\_RHMI\_355]</sub>

<sub>[1\_RHMI\_54]</sub>The PlaceCountry and PlaceRegion defines the ‘place where the daily work period ends’ and have the format NationNumeric for PlaceCountry and RegionNumeric for PlaceRegion as specified in [Annex 1C V2].

<sub>[1\_RHMI\_356]</sub>If PlaceCountry = ‘no information available’, then the VU shall reject this ‘DriverWorkshopCardWithdrawal’ request by the negative response ‘requestOutOfRange’.

#### 4.6.1.1.3 Parameter CurrentShiftPrintoutRequest<sub>[1\_RHMI\_355a]</sub>

<sub>[1\_RHMI\_55]</sub>The CurrentShiftPrintoutRequest defines, if the ‘driver activities from the card daily’ printout for the current day (see [Annex 1C V2], Appendix 4, PRT\_008) shall be performed at the end of this card withdrawal procedure:

Value	Definition
'00'H	No printout requested.
'01'H	The VU shall perform the 'driver activities from the card daily' printout print using UTC time base.
'02'H	The VU shall perform the 'driver activities from the card daily' printout using local time base.

**Table:** Parameter CurrentShiftPrintoutRequest.**4.6.1.1.4 Parameter TechnicalDataPrintoutRequest** [1\_RHMI\_355b]

[1\_RHMI\_55a] The TechnicalDataPrintoutRequest defines, if the 'technical data printout (see [Annex 1C V2], Appendix 4, PRT\_012) shall be performed at the end of this card withdrawal procedure:

Value	Definition
'00'H	No printout requested.
'01'H	The VU shall perform the 'technical data' printout.

**Table:** Parameter TechnicalDataPrintoutRequest.**4.6.1.2 Routine 'DriverWorkshopCardWithdrawal', StartRoutine Positive Response** [1\_RHMI\_56]

[1\_RHMI\_57] The RoutineControl – Start Routine 'DriverWorkshopCardWithdrawal' – Positive Response Message provides confirmation, if the withdrawal request is correct.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = DriverWorkshopCardWithdrawal [byte#1(MSB), byte#2]	'F201'H

**Table:** Routine 'DriverWorkshopCardWithdrawal', StartRoutine Positive Response.**4.6.1.3 Routine 'DriverWorkshopCardWithdrawalDefinition', negative response code** [1\_RHMI\_60a]

[1\_RHMI\_60] This definition contains the negative response conditions for routine DriverWorkshopCardWithdrawal, which are additional to the standard negative responses for the service RoutineControl.

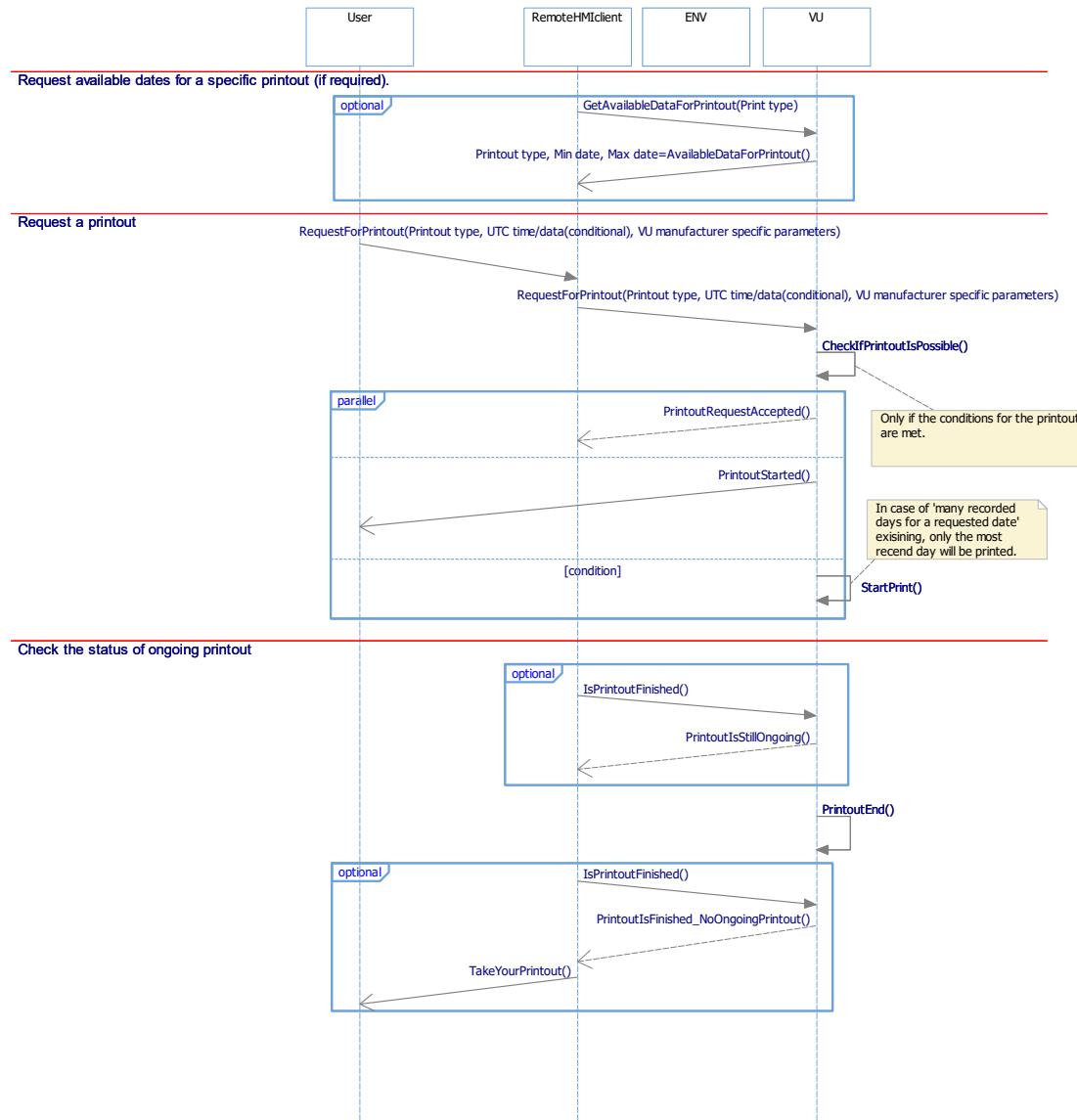
Response name	Value	Description
conditionsNotCorrect	'22'H	<p>The conditions for execution of card withdrawal are not met:</p> <ul style="list-style-type: none"> <li>• The Remote HMI session is closed;</li> <li>• The VU is not activated;</li> <li>• No driver/workshop card is inserted into requested smart card slot;</li> <li>• The inserted driver/workshop card is not ready to be withdrawn;</li> <li>• The VU is a ADR variant and the ignition is off;</li> <li>• The card withdrawal procedure is already ongoing for the requested smart card slot;</li> <li>• Any other reason, when a card withdrawal is not possible or not allowed.</li> </ul>
busyRepeatRequest	21'H	<p>The withdrawal of the appropriate card is not possible at the moment:</p> <ul style="list-style-type: none"> <li>• A smart card in another smart card slot is currently processed.</li> </ul>
requestOutOfRange	'31'H	<ul style="list-style-type: none"> <li>• At least one parameter for the routine 'DriverWorkshopCardWithdrawal' is out of range;</li> <li>• The PlaceCountry and PlaceRegion are not consistent with each other;</li> <li>• The request timestamp is outside of expected range. See [1_RHMI_226];</li> <li>• PlaceCountry = 'no information available'.</li> </ul>
invalidKey	'35'H	<p>The Remote HMI checksum is incorrect (see [1_RHMI_289]).</p> <p><b>Note:</b> The security as defined by VU PP is not involved here.</p>
requiredTimeDelayNotExpired	'37'H	<p>The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288]).</p>
requestCorrectlyReceived-ResponsePending	'78'H	<p>The request was received correctly, and all parameters within the request were valid, but the request procedure is not yet completed and the VU is not yet ready to receive another request.</p>

**Table:** Routine 'DriverWorkshopCardWithdrawalDefinition', negative response code .

## 4.7 Remote start of a VU printout [1\_RHMI\_66]

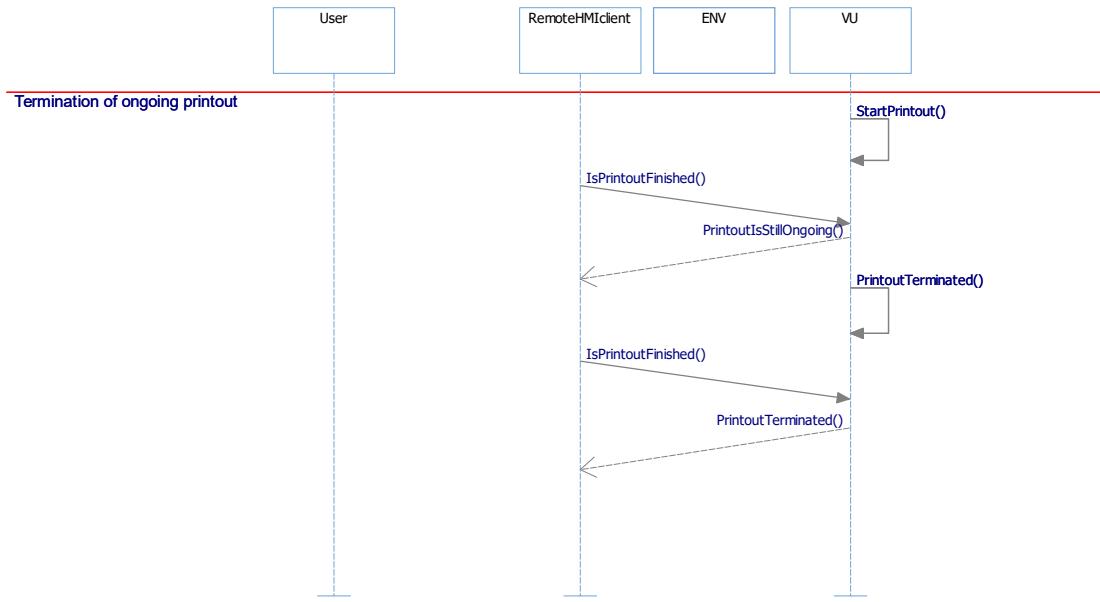
[1\_RHMI\_67] The VU shall provide the possibility to start some of the VU printouts via RoutineControl service.

[1\_RHMI\_68]



**Figure:** Remote printout request sequence – happy day.

[1\_RHMI\_205]



**Figure:** Remote printout request sequence – exceptions.

#### 4.7.1 Routine ‘Printout’<sup>[1\_RHMI\_69]</sup>

<sup>[1\_RHMI\_70]</sup>The routine ‘Printout’ provides the possibility to start a VU printout, to get the status of (ongoing) printout and, if applicable, to request for the printout available dates.

##### 4.7.1.1 Routine ‘Printout’, GetPrintoutPeriod, StartRoutine Request<sup>[1\_RHMI\_72]</sup>

<sup>[1\_RHMI\_200]</sup>The StartRoutine ‘Printout’ with the option GetPrintoutPeriod requests the available data period for a specific VU printout.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	‘31’H
2	routineControlType = StartRoutine	‘01’H
3, 4	routineIdentifier[] = Printout [byte#1(MSB), byte#2]	‘F210’H
5	routineControlOption[byte#1] = GetPrintoutPeriod	‘01’H
6	routineControlOption[byte#2] = PrintoutType (see [1_RHMI_74]).	‘XX’H
7..10	routineControlOption[byte#3..byte#6] = RHMITimestamp (see [1_RHMI_339])	‘XX’H ... ‘XX’H
11..14	routineControlOption[byte#7..byte#10] = RHMICS (see [1_RHMI_279])	‘XX’H ... ‘XX’H

**Table:** Routine ‘Printout’, GetPrintoutPeriod, StartRoutine Request.

#### 4.7.1.1.1 Parameter PrintoutType and printout parameters overview<sub>[1\_RHMI\_74a]</sub>

[1\_RHMI\_74]

Printout type	Printout identifier	Required PrintoutDate
RFU	‘00’H.	
The driver activities from card daily printout	‘01’H	required
The co-driver activities from card daily printout	‘02’H	required
The driver events and faults from card printout	‘03’H	-
The co-driver events and faults from card printout	‘04’H	-
The drivers activities from VU daily printout	‘05’H	required
The events and faults from VU printout	‘06’H	-
The technical data printout	‘07’H	-
The over speeding printout	‘08’H	-
The historic of inserted cards printout	‘09’H	-
RFU	‘0A’H..‘7F’H	
VU manufacturer specific printouts	‘80’H..‘FF’H	VU manufacturer specific.

**Table:** Parameter PrintoutType and printout parameters overview.

#### 4.7.1.2 Routine ‘Printout’, GetPrintoutPeriod, StartRoutine Positive Response<sub>[1\_RHMI\_199]</sub>

[1\_RHMI\_201] The Positive Response on the StartRoutine ‘Printout’ with the option GetPrintoutPeriod provides the available data period for a specific VU printout.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	‘71’H
2	routineControlType = StartRoutine	‘01’H
3, 4	routineIdentifier[] = Printout [byte#1(MSB), byte#2]	‘F210’H
5	routineStatusRecord[byte#1] = GetPrintoutPeriod	‘01’H

Byte#	Parameter name	Value
6	routineStatusRecord[byte#2] = PrintoutType (see [1_RHMI_80])	'XX'H
7..10	routineStatusRecord [routineStatus#3.. routineStatus#6] = OldestPrintoutDate[byte#1(MSB)..byte#4] (see [1_RHMI_81])	'XX'H ... 'XX'H
11..14	routineStatusRecord [routineStatus#7.. routineStatus#10] = NewestPrintoutDate[byte#1(MSB)..byte#4] (see [1_RHMI_81])	'XX'H ... 'XX'H

**Table:** Routine ‘Printout’, GetPrintoutPeriod, StartRoutine Positive Response.

#### 4.7.1.2.1 Parameters OldestPrintoutDate and NewestPrintoutDate<sub>[1\_RHMI\_81a]</sub>

<sub>[1\_RHMI\_81]</sub>If a printout requires a date as mandatory printout parameter (see [1\_RHMI\_74]), then:

- The OldestPrintoutDate is the oldest available date for such printout;
- The NewestPrintoutDate is the newest available date for such printout.

If a printout does not require a date as mandatory parameter, then the VU shall provide the oldest and newest available dates as ‘not available’.

The OldestPrintoutDate and NewestPrintoutDate have the data type TimeReal (see [Annex 1C V2], Appendix 1).

#### 4.7.1.3 Routine ‘Printout’, StartPrintout, StartRoutine Request<sub>[1\_RHMI\_202]</sub>

<sub>[1\_RHMI\_73]</sub>The StartRoutine ‘Printout’ with the option StartPrintout requests a specific VU printout, with required printout date (see [1\_RHMI\_74]).

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = Printout [byte#1(MSB), byte#2]	'F210'H
5	routineControlOption[byte#1] = StartPrintout	'02'H
6	routineControlOption[byte#2] = PrintoutType (see [1_RHMI_74]).	'XX'H
7..10	routineControlOption[byte#3..byte#6] = PrintoutDate [byte#1 (MSB)..byte#4] (see [1_RHMI_76])	'XX'H ... 'XX'H
11	routineControlOption[byte#7] = VUManufacturer (see [1_RHMI_73b])	'XX'H
12	routineControlOption[byte#8] = VUManufacturerParameterSize (see [1_RHMI_73c])	'XX'H

Byte#	Parameter name	Value
13 .. 13+(m-9)	routineControlOption[byte#9..m] = VUManufacturerParameter (C <sub>1</sub> ) (see [1_RHMI_73d])	'XX'H ... 'XX'H
13+(m-9)+1 .. 13+(m-9)+4	routineControlOption[byte#m+1..byte#m+4] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
13+(m-9)+5 .. 13+(m-9)+8	routineControlOption[byte#m+5..byte#m+8] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘Printout’, StartPrintout, StartRoutine Request, with required printout date.

**Caption:**

C <sub>1</sub>	The number of bytes of this parameter is defined by the VUManufacturerParameterSize Option parameter.
----------------	---

[1\_RHMI\_73a] The StartRoutine ‘Printout’ with the option StartPrintout requests a specific VU printout, without required printout date (see [1\_RHMI\_74]).

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = Printout [byte#1(MSB), byte#2]	'F210'H
5	routineControlOption[byte#1] = StartPrintout	'02'H
6	routineControlOption[byte#2] = PrintoutType (see [1_RHMI_74]).	'XX'H
7	routineControlOption[byte#3] = VUManufacturer (see [1_RHMI_73b])	'XX'H
8	routineControlOption[byte#4] = VUManufacturerParameterSize (see [1_RHMI_73c])	'XX'H
9 .. 9+(m-5)	routineControlOption[byte#5..m] = VUManufacturerParameter (C <sub>1</sub> ) (see [1_RHMI_73d])	'XX'H ... 'XX'H
9+(m-5)+1 .. 9+(m-5)+4	routineControlOption[byte#m+1..byte#m+4] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H

Byte#	Parameter name	Value
9+(m-5)+5 .. 9+(m-5)+8	routineControlOption[byte#m+5..byte#m+8] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘Printout’, StartPrintout, StartRoutine Request, without required printout date.

#### 4.7.1.3.1 Parameter VUManufacturer [1\_RHMI\_202a]

[1\_RHMI\_73b] The parameter VUManufacturer provides the VU manufacturer, when the RHMI client uses VU specific parameters.

The VUManufacturer shall be according to [Annex 1C V2], Appendix 1, ManufacturerCode.

In case the VU manufacture specific parameter is not used, then the RHMI client shall set the ManufacturerCode to ‘FF’H.

#### 4.7.1.3.2 VUManufacturerParameterSize [1\_RHMI\_202b]

[1\_RHMI\_73c] The parameter VUManufacturerParameterSize shall be INTEGER(0..255) and provides the size of the following VU manufacturer parameter (see [1\_RHMI\_73d]) in bytes.

In case the VU manufacturer specific parameter is not used, then the RHMI client shall set the VUManufacturerParameterSize to 0.

[1\_RHMI\_73e] In the StartRoutine ‘Printout’ with the option StartPrintout request, and if the VU manufacturer specific parameter is not used and the VUManufacturerParameterSize is not 0, then the VU shall reject the request by negative response ‘requestOutOfRange’.

#### 4.7.1.3.3 VUManufacturerParameter [1\_RHMI\_202c]

[1\_RHMI\_73d] The parameter VUManufacturerParameter provides additional VU manufacturer specific parameter for the specific Remote HMI request.

#### 4.7.1.3.4 Parameter PrintoutDate [1\_RHMI\_202d]

[1\_RHMI\_76] If applicable (see [1\_RHMI\_74]), the parameter PrintoutDate provides the date for requested printout, in UTC and as data type TimeReal (see [Annex 1C V2], Appendix 1, data type TimeReal).

**4.7.1.4 Routine ‘Printout’, StartPrintout, StartRoutine Positive Response**<sub>[1\_RHMI\_77]</sub>

[1\_RHMI\_78] The Positive Response on the StartRoutine ‘Printout’ request with the option StartPrintout provides confirmation of the requested printout start.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = Printout [byte#1(MSB), byte#2]	'F210'H
5	routineStatusRecord[byte#1] = StartPrintout	'02'H
6	routineStatusRecord[byte#2] = PrintoutType (see [1_RHMI_80])	'XX'H

**Table:** Routine ‘Printout’, StartPrintout, StartRoutine Positive Response.

**4.7.1.4.1 Parameter PrintoutType**<sub>[1\_RHMI\_80a]</sub>

[1\_RHMI\_80] The PrintoutType shall be the echo of the PrintoutType within the request.

**4.7.1.5 Routine ‘Printout’, RequestRoutineResults Request**<sub>[1\_RHMI\_82]</sub>

[1\_RHMI\_83] The RoutineControl – Request Routine ‘Printout’ Results – Request Message requests the status of ongoing or last printout.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = Printout [byte#1(MSB), byte#2]	'F210'H
5..8	routineControlOption[byte#1..byte#4] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
9..12	routineControlOption[byte#5..byte#8] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘Printout’, RequestRoutineResults Request.

**4.7.1.6 Routine ‘Printout’, RequestRoutineResults Positive Response**<sub>[1\_RHMI\_84]</sub>

[1\_RHMI\_85] The RoutineControl – Request Routine ‘Printout’ Results – Positive Response Message provides the status of ongoing or last printout.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = Printout [byte#1(MSB), byte#2]	'F210'H
5	routineStatusRecord[byte#1] = routineInfo (see [1_RHMI_86])	'XX'H
6	routineStatusRecord[byte#2] = PrintoutType (see [1_RHMI_87])	'XX'H
7..10	routineStatusRecord[byte#3..byte#6] = PrintoutEnd[byte#1(MSB)..byte#4] (see [1_RHMI_88])	'XX'H ... 'XX'H

**Table:** Routine ‘Printout’, RequestRoutineResults Positive Response.**4.7.1.7 Routine ‘Printout’, VU manufacturer specific parameters** [1\_RHMI\_202e]**4.7.1.7.1 Stoneridge** [1\_RHMI\_202f]

[1\_RHMI\_85a]

PrintoutType	VUManufacturerParameterSize	Format of VUManufacturerParameter
the driver activities from card daily printout	1	INTEGER(0..255): 1: printout of data from the generation 1 card application;
The co-driver activities from card daily printout	1	2: printout of data from the generation 2 card application.

**Table:** Routine ‘Printout’, parameter VUManufacturerParameter, Stoneridge specific.**4.7.1.7.2 Continental** [1\_RHMI\_202g]

[1\_RHMI\_75]

PrintoutType	VUManufacturerParameterSize	Format of VUManufacturerParameter
the driver activities from card daily printout	1	INTEGER(0..255): 1: the VU shall print using UTC time base;
The co-driver activities from card daily printout	1	2: the VU shall print using local time base.

<b>PrintoutType</b>	<b>VUManufacturerParameterSize</b>	<b>Format of VUManufacturerParameter</b>
the driver events and faults from card printout		
the co-driver events and faults from card printout		
the drivers activities from vu daily printout		
the events and faults from vu printout		
the technical data printout		

**Table:** Routine ‘Printout’, parameter VUManufacturerParameter, Continental specific.

#### 4.7.1.7.3 Parameter routineInfo<sub>[1\_RHMI\_86a]</sub>

[1\_RHMI\_86]

<b>Response Name</b>	<b>Value</b>	<b>Definition</b>
NormalExitWithResultAvailable	'61'H	The ‘Printout’ routine was executed completely.
AbnormalExitWithResultsAvailable	'63'H	The ‘Printout’ routine was terminate before completion (e.g by the user, by paper end, etc.).
RoutineExecutionOngoing	'78'H	The ‘Printout’ routine is still executed.
NoPrintoutStatusAvailable	'FF'H	No last printout known and not current printout ongoing.

**Table:** Parameter routineInfo.

#### 4.7.1.7.4 Parameter PrintoutType<sub>[1\_RHMI\_87a]</sub>

[1\_RHMI\_87] Within the RoutineControl – Request Routine ‘Printout’ Results – Positive Response Message, the PrintoutType provides the current or last executed printout, as defined by [1\_RHMI\_74].

If the current or last executed printout is not defined within [1\_RHMI\_74] (i.e. such printout is not supported to be requested via Remote HMI interface), then the VU shall set the PrintoutType to 'FA'H.

#### 4.7.1.7.5 Parameters PrintoutEnd<sub>[1\_RHMI\_88a]</sub>

<sub>[1\_RHMI\_88]</sub>If a printout is ongoing, then the PrintoutEnd is set to 'not available'.

If no printout is ongoing, then the PrintoutEnd is the timestamp of the end of last executed or abnormally terminated printout.

The data type of PrintoutEnd is TimeReal.

#### 4.7.1.8 Routine 'Printout', negative response code <sub>[1\_RHMI\_89a]</sub>

<sub>[1\_RHMI\_89]</sub>This definition contains the negative response conditions for routine 'Printout'.

Response name	Value	Description
conditionsNotCorrect	'22'H	<p>The conditions for execution of the printout are not met:</p> <ul style="list-style-type: none"> <li>• Due to ADR intended use (see [1_MB_5751]);</li> <li><b>Exception:</b> This is not valid for the RequestRoutineResults;</li> <li>• Printer head over-temperature;</li> <li>• Any other reason, why a printout is not possible or not allowed.</li> </ul>
busyRepeatRequest	'21'H	<p>The VU is temporarily too busy to perform the request:</p> <ul style="list-style-type: none"> <li>• A printout is already in the printout queue or a printout is ongoing – no additional printout could be accepted;</li> <li><b>Only for RequestRoutineResults request:</b> A remote HMI user confirmation request screen is currently shown;</li> <li>• Other routine 'Printout' request is currently processed.</li> </ul>
requestOutOfRange	'31'H	<ul style="list-style-type: none"> <li>• At least one parameter for the routine 'Printout' is out of range.</li> <li>• A VU manufacturer specific parameter is not supported.</li> <li>• the VU manufacturer specific parameter is not used but VUManufacturerParameterSize is &gt; 0.</li> <li>• The requested date for the requested printout is not available.</li> </ul>

Response name	Value	Description
		<ul style="list-style-type: none"> <li>The request timestamp is outside of expected range. See [1_RHMI_226]]</li> </ul>
invalidKey	'35'H	<p>The Remote HMI checksum is incorrect (see [1_RHMI_289]).</p> <p><b>Note:</b> The security as defined by VU PP is not involved here.</p>
requiredTimeDelayNotExpired	'37'H	The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288]).
requestCorrectlyReceived-ResponsePending	'78'H	The VU needs more time to process this request.
		<b>Note:</b> last row.

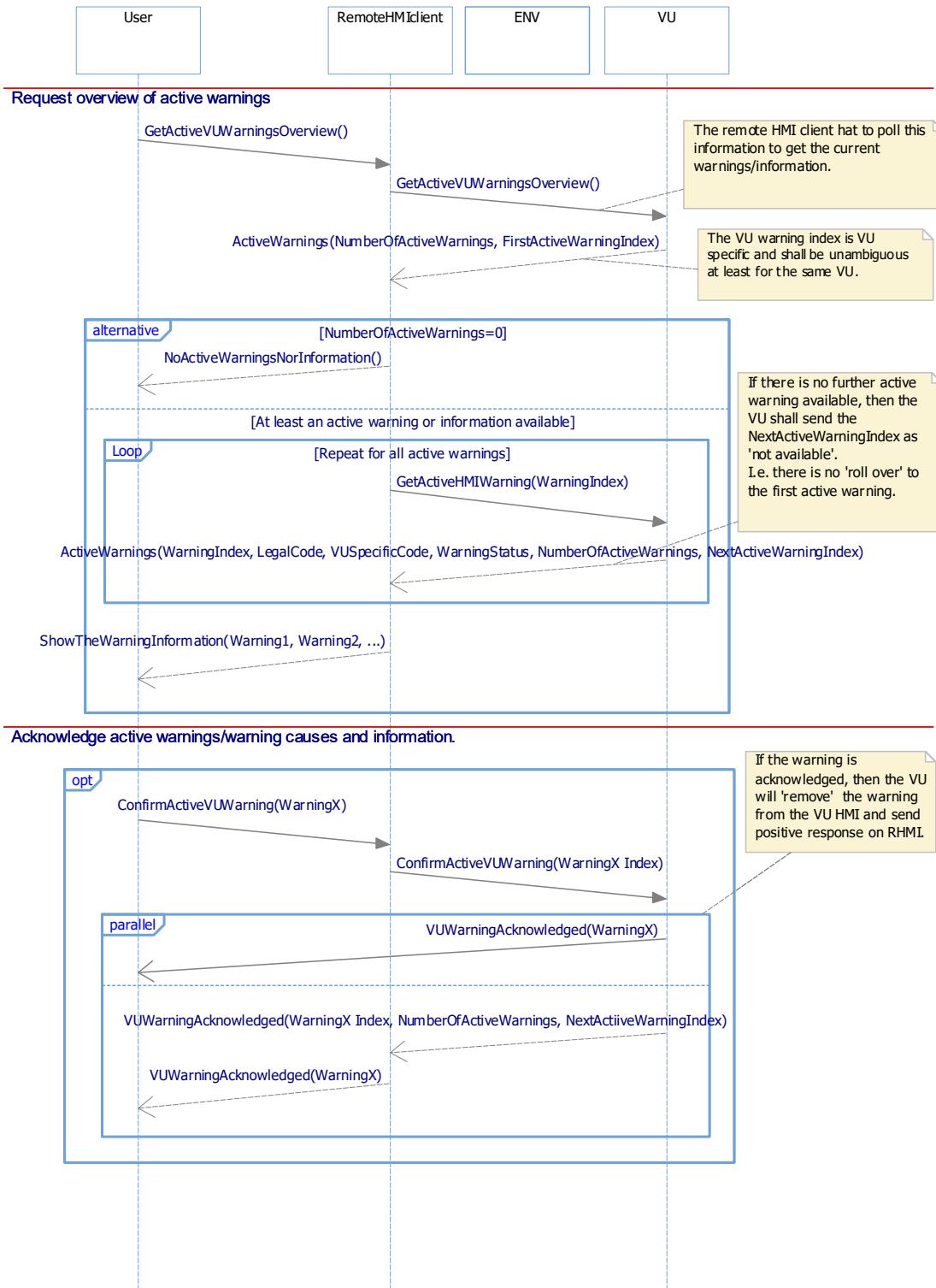
**Table:** Routine ‘Printout’, negative response code.

## 4.8 Request and confirm active warnings and user information<sub>[1\_RHMI\_92]</sub>

<sub>[1\_RHMI\_93]</sub>The VU shall provide the possibility to request and to acknowledge active warnings causes and other user information.

<sub>[1\_RHMI\_94]</sub>

# Smart Tachograph V2: VU, Remote Human Machine Interface Specification



**Figure:** Request and acknowledge of active warning causes and information sequences – happy day.

### 4.8.1 Routine ‘AcknowledgeWarningsAndInfo’<sub>[1\_RHMI\_97]</sub>

<sub>[1\_RHMI\_98]</sub>The routine ‘AcknowledgeWarningsAndInfo’ provides the possibility to request all active warnings/warning causes and information, and to acknowledge them.

#### 4.8.1.1 Routine ‘AcknowledgeWarningsAndInfo’, GetActiveVUWarningsOverview, RequestRoutineResults Request<sub>[1\_RHMI\_99]</sub>

<sub>[1\_RHMI\_100]</sub>The RoutineControl – Request Routine ‘AcknowledgeWarningsAndInfo’ Results – Request Message, with the option GetActiveVUWarningsOverview, requests number of all active VU warnings/warning causes and information.

Byte#	Parameter name	Value
1	serviced = RoutineControl	'31'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = AcknowledgeWarningsAndInfo [byte#1(MSB), byte#2]	'F202'H
5	routineControlOption[byte#1] = GetActiveVUWarningsOverview	'01'H
6..9	routineControlOption[byte#2..byte#5] = RHMITimestamp (see – [1_RHMI_339])	'XX'H ... 'XX'H
10..13	routineControlOption[byte#6..byte#9] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘AcknowledgeWarningsAndInfo’, RequestRoutineResults Request, with the option GetActiveVUWarningsOverview.

#### 4.8.1.2 Routine ‘AcknowledgeWarningsAndInfo’, GetActiveVUWarningsOverview, RequestRoutineResults Positive Response<sub>[1\_RHMI\_101]</sub>

<sub>[1\_RHMI\_102]</sub>The RoutineControl – Request Routine ‘AcknowledgeWarningsAndInfo’ Results for the option GetActiveVUWarningsOverview – Positive Response Message provides the overview of active recording equipment warnings/warning causes and information.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = AcknowledgeWarningsAndInfo[byte#1(MSB), byte#2]	'F202'H

Byte#	Parameter name	Value
5	routineControlOption[byte#1] = GetActiveVUWarningsOverview	'01'H
6, 7	routineControlOption[byte#2(MSB), byte#3] = NumberOfActiveWarnings (see [1_RHMI_102a])	'XXXX'H
8, 9	routineControlOption[byte#4(MSB), byte#5] = FirstActiveWarning (see [1_RHMI_102b])	'XXXX'H

**Table:** Routine 'AcknowledgeWarningsAndInfo', RequestRoutineResults, for the option GetActiveVUWarningsOverview, Positive Response.

#### 4.8.1.2.1 Parameter NumberOfActiveWarnings [1\_RHMI\_101a]

[1\_RHMI\_102a] The NumberOfActiveWarnings shall be INTEGER(0..2<sup>16</sup>-1) and shall provide the number of active VU warnings, 0 if currently no warnings are active.

#### 4.8.1.2.2 Parameter FirstActiveWarning [1\_RHMI\_101b]

[1\_RHMI\_102b] The FirstActiveWarning shall have format WarningIndex (see [1\_RHMI\_102c]) and shall provide the index of first active VU warning.

**Note:** the definition of 'first active VU warning' is VU specific. It may be e.g. the current shown VU warning or first active warning according to VU specific implementation.

#### 4.8.1.2.3 Parameter WarningIndex [1\_RHMI\_101c]

[1\_RHMI\_102c] The WarningIndex shall be INTEGER(0..2<sup>16</sup>-1) and is the index of a specific VU warnings.

The WarningIndex for a VU warning shall be unambiguous within same VU and can be VU specific.

The value 'FFFF'H is specified as 'not available'.

#### 4.8.1.3 Routine 'AcknowledgeWarningsAndInfo', GetActiveVUWarning, RequestRoutineResults Request [1\_RHMI\_101d]

[1\_RHMI\_100a] The RoutineControl – Request Routine 'AcknowledgeWarningsAndInfo' Results – Request Message, with the option GetActiveVUWarning, requests data about specific VU warnings/warning causes and information.

Byte#	Parameter name	Value
1	serviced = RoutineControl	'31'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = AcknowledgeWarningsAndInfo [byte#1(MSB), byte#2]	'F202'H

5	routineControlOption[byte#1] = GetActiveVUWarning	'02'H
6, 7	routineControlOption[byte#2(MSB), byte#3] = WarningIndex (see [1_RHMI_102c])	'XXXX'H
8..11	routineControlOption[byte#4..byte#7] = RHMITimestamp (see – [1_RHMI_339])	'XX'H ... 'XX'H
12..15	routineControlOption[byte#8..byte#11] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine ‘AcknowledgeWarningsAndInfo’, RequestRoutineResults Request, with the option GetActiveVUWarning.

#### 4.8.1.4 Routine ‘AcknowledgeWarningsAndInfo’, GetActiveVUWarning, RequestRoutineResults Positive Response [1\_RHMI\_101e]

[1\_RHMI\_100b] The RoutineControl – Request Routine ‘AcknowledgeWarningsAndInfo’ Results for the option GetActiveVUWarning – Positive Response Message provides the overview of active recording equipment warnings/warning causes and information.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = RequestRoutineResults	'03'H
3, 4	routineIdentifier[] = AcknowledgeWarningsAndInfo[byte#1(MSB), byte#2]	'F202'H
5	routineControlOption[byte#1] = GetActiveVUWarning	'02'H
6, 7	routineControlOption[byte#2(MSB), byte#3] = WarningIndex (see [1_RHMI_102c])	'XXXX'H
8	routineControlOption[byte#4] = WarningLegalCode (see [1_RHMI_100c])	'XX'H
9..12	routineControlOption[byte#5(MSB)..byte#8] = WarningVUSpecificCode (see [1_RHMI_100d])	'XX'H ... 'XX'H
13	routineControlOption[byte#9] = WarningStatus (see [1_RHMI_100e])	'XX'H
14, 15	routineControlOption[byte#10(MSB), byte#11] = NumberOfActiveWarnings (see [1_RHMI_102a])	'XXXX'H
16, 17	routineControlOption[byte#12(MSB), byte#13] = NextActiveWarning (see [1_RHMI_100f])	'XXXX'H

**Table:** Routine 'AcknowledgeWarningsAndInfo', RequestRoutineResults, for the option GetActiveVUWarning, Positive Response.

#### Parameter WarningLegalCode [1\_RHMI\_101f]

[1\_RHMI\_100c] The WarningLegalCode contains the legal VU event/fault code of the warning and has format EventFaultType according to [Annex 1C V2], Appendix 1.

#### 4.8.1.4.1 Parameter WarningVUSpecificCode [1\_RHMI\_101g]

[1\_RHMI\_100d] The parameter WarningVUSpecificCode contains the VU specific warning code and has format ManufacturerSpecificEventFaultData according to [Annex 1C V2], Appendix 1.

#### 4.8.1.4.2 Parameter WarningStatus [1\_RHMI\_101h]

[1\_RHMI\_100e] The WarningStatus contains the current status of the VU warning and has format INTEGER(0..255):

0: warning is inactive.

1: warning is active.

2..255: RFU.

#### 4.8.1.4.3 Parameter NextActiveWarning [1\_RHMI\_101j]

[1\_RHMI\_100f] The NextActiveWarning shall have format WarningIndex (see [1\_RHMI\_102c]) and provides the index of next active VU warning (if available).

#### 4.8.1.5 Routine 'AcknowledgeWarningsAndInfo', StartRoutine Request [1\_RHMI\_105]

[1\_RHMI\_106] The RoutineControl - Start Routine 'AcknowledgeWarningsAndInfo' - Request Message acknowledges a specific active warnings/warning causes and information.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = AcknowledgeWarningsAndInfo [byte#1(MSB), byte#2]	'F202'H
5, 6	routineControlOption[byte1(MSB), byte#2] = WarningIndex (see [1_RHMI_102c])	'XXXX'H

Byte#	Parameter name	Value
7..10	routineControlOption[byte#3(MSB)..byte#6] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
11..14	routineControlOption[byte#7..byte#10] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine 'RequestConfirmActiveWarningsAndInformation', StartRoutine Request.

#### 4.8.1.6 Routine 'AcknowledgeWarningsAndInfo', StartRoutine Positive Response<sub>[1\_RHMI\_107]</sub>

<sub>[1\_RHMI\_108]</sub>The RoutineControl - Start Routine 'AcknowledgeWarningsAndInfo' – Positive Response Message confirms the remote acknowledge of a specific active warnings / warnings causes and user information.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = AcknowledgeWarningsAndInfo [byte#1(MSB), byte#2]	'F202'H
5, 6	routineControlOption[byte1(MSB), byte#2] = WarningIndex (see [1_RHMI_102c])	'XXXX'H
7, 8	routineControlOption[byte#3(MSB), byte#4] = NumberOfActiveWarnings (see [1_RHMI_102a])	'XXXX'H
9, 10	routineControlOption[byte#5(MSB), byte#6] = NextActiveWarning (see [1_RHMI_100f])	'XXXX'H

**Table:** Routine 'AcknowledgeWarningsAndInfo', StartRoutine Positive Response.

<sub>[1\_RHMI\_109]</sub>If the VU accepts the remote acknowledge request for a specific active warning / warning cause and information, then the VU shall acknowledge this warning/warning cause and information, even the warning/warning cause currently not shown on the VU display.

<sub>[1\_RHMI\_110a]</sub> The Remote HMI client shall show to the user the active recording equipment event/fault and other information, before it allows the user to acknowledge the event/fault/information.

#### 4.8.1.7 Routine 'AcknowledgeWarningsAndInfo', negative response code

[1\_RHMI\_112a]

[1\_RHMI\_112] This definition contains the negative response conditions for routine 'AcknowledgeWarningsAndInfo', which are additional to the standard negative responses for the service RoutineControl.

Response name	Value	Description
conditionsNotCorrect	'22'H	The Remote HMI session is closed.
requestOutOfRange	'31'H	<ul style="list-style-type: none"> <li>The request timestamp is outside of expected range. See [1_RHMI_226].</li> <li>The warning / warning cause to be acknowledged is not supported by the VU.</li> <li>At least one of the routine options is outside of supported values range.</li> </ul>
invalidKey	'35'H	<p>The Remote HMI checksum is incorrect (see [1_RHMI_289]).</p> <p><b>Note:</b> The security as defined by VU PP is not involved here.</p>
requiredTimeDelayNotExpired	'37'H	The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288]).
requestCorrectlyReceived-ResponsePending	'78'H	The request was received correctly, and all parameters within the request were valid, but the request procedure is not yet completed and the VU is not yet ready to receive another request.

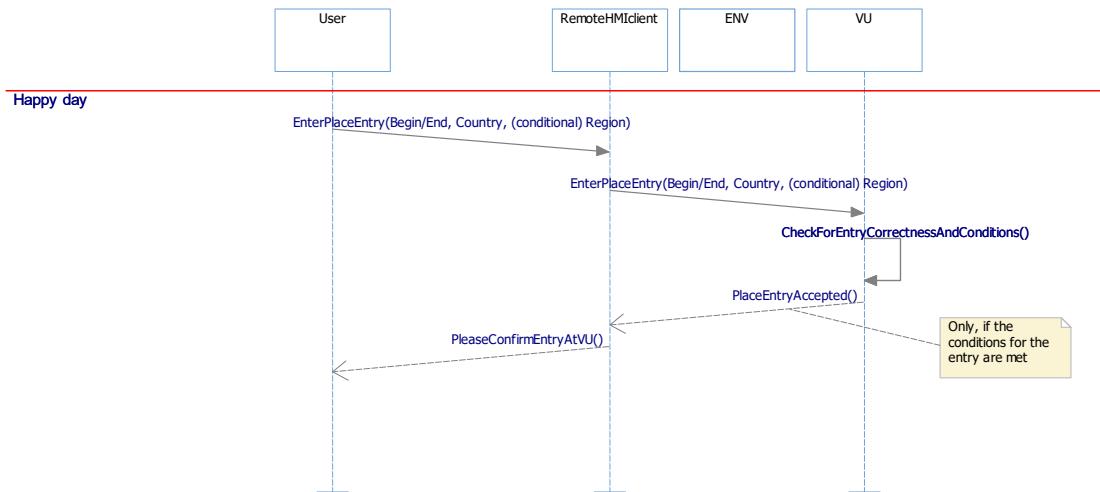
Table: Routine 'AcknowledgeWarningsAndInfo', negative response code.

#### 4.9 Remote entry of places where daily work periods begin and/or end

[1\_RHMI\_113]

[1\_RHMI\_114] The VU shall provide the possibility to enter a place where daily work periods begin and/or end via RoutineControl service.

[1\_RHMI\_115]



**Figure:** remote place entry sequence – happy day.

#### 4.9.1 Routine 'Entry of places where daily work periods begin and/or end'<sup>[1\_RHMI\_119]</sup>

<sup>[1\_RHMI\_120]</sup>If the VU is not activated, then the VU shall reject the 'RoutineControl - Request Routine 'EntryOfPlace' requests by appropriate negative response.

##### 4.9.1.1 Routine 'EntryOfPlace', StartRoutine Request<sup>[1\_RHMI\_121]</sup>

<sup>[1\_RHMI\_122]</sup>The RoutineControl - Start Routine 'EntryOfPlace' - Request Message provides the possibility to request an entry of places where daily work periods begin and/or end.

Byte#	Parameter name	Value
1	serviced = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3..4	routineIdentifier[] = EntryOfPlace [byte#1(MSB), byte#2]	'F203'H
5	routineControlOption[byte#1] = SmartCardSlot (see [1_RHMI_53])	'XX'H
6	routineControlOption[byte#2] = EntryType (see [1_RHMI_123])	'XX'H
7	routineControlOption[byte#3] = PlaceCountry (see [1_RHMI_124])	'XX'H
8	routineControlOption[byte#4] = PlaceRegion (see [1_RHMI_124])	'XX'H
9..12	routineControlOption[byte#5..byte#8] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H

Byte#	Parameter name	Value
13..16	routineControlOption[byte#9..byte#12] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine 'EntryOfPlace', StartRoutine Request.**4.9.1.1.1 Parameter EntryType**<sub>[1\_RHMI\_123a]</sub>

[1\_RHMI\_123] The EntryType defines the type of entry: The place where the daily work period begins or where it ends as data type DailyWorkPeriodEntryType (see [1\_DaDi\_3212]).

**4.9.1.1.2 Parameters PlaceCountry and PlaceRegion**<sub>[1\_RHMI\_124a]</sub>

[1\_RHMI\_124] The PlaceCountry and PlaceRegion defines the 'place where the daily work period begin and/or end' and have the format NationNumeric for PlaceCountry and RegionNumeric for PlaceRegion (see [Annex 1C V2], Appendix 1).

[1\_RHMI\_125] If PlaceCountry = 'no information available' within remote place entry request, then the VU shall reject the remote place entry request by appropriate negative response.

**4.9.1.2 Routine 'EntryOfPlace', StartRoutine Positive Response**<sub>[1\_RHMI\_126]</sub>

[1\_RHMI\_127] The RoutineControl - Start Routine 'EntryOfPlace' - Positive Response Message provides confirmation, if the remote place entry request is correct.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = EntryOfPlace [byte#1(MSB), byte#2]	'F203'H

**Table:** Routine 'EntryOfPlace', StartRoutine Positive Response.**4.9.1.3 Routine EntryOfPlace, negative response code**<sub>[1\_RHMI\_130a]</sub>

[1\_RHMI\_130] This definition contains the negative response conditions for routine EntryOfPlace.

Response Name	Value	Description
conditionsNotCorrect	'22'H	The conditions for execution of remote place entry are not met: <ul style="list-style-type: none"><li>• The Remote HMI session is closed</li><li>• The VU is not activated;</li></ul>

Response Name	Value	Description
		<ul style="list-style-type: none"> <li>• A card insertion or withdrawal procedure is ongoing;</li> <li>• A card operation is ongoing;</li> <li>• Same type of entry already exists in this calendar minute - only one entry per calendar minute is allowed (see [Annex 1C V2] requirement 62);</li> <li>• Any other reason, when a place entry is not possible or not allowed.</li> </ul>
requestOutOfRange	'31'H	<ul style="list-style-type: none"> <li>• At least one parameter for the routine 'EntryOfPlace' is out of range.</li> <li>• The PlaceCountry and PlaceRegion are not consistent to each other.</li> <li>• The request timestamp is outside of expected range. See [1_RHMI_226]</li> </ul>
invalidKey	'35'H	<p>The Remote HMI checksum is incorrect (see [1_RHMI_289]).</p> <p><b>Note:</b> The security as defined by VU PP is not involved here.</p>
requiredTimeDelayNotExpired	'37'H	<p>The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288]).</p>
requestCorrectlyReceived-ResponsePending	'78'H	<p>The request was received correctly, and all parameters within the request were valid, but the request procedure is not yet completed and the VU is not yet ready to receive another request.</p>

**Table:** Routine EntryOfPlace, negative response code.

## 4.10 Remote entry of specific condition<sub>[1\_RHMI\_136]</sub>

<sub>[1\_RHMI\_137]</sub>The VU shall provide the possibility to enter the specific conditions via RoutineControl service.

<sub>[1\_RHMI\_138]</sub>



**Figure:** remote specific condition entry sequence – happy day.

#### 4.10.1 Routine 'Entry of specific condition'<sub>[1\_RHMI\_139]</sub>

[1\_RHMI\_140] If the VU is not activated, then the VU shall reject the 'RoutineControl - Request Routine 'EntryOfSpecificCondition' requests by appropriate negative response.

##### 4.10.1.1 Routine 'EntryOfSpecificCondition', StartRoutine Request<sub>[1\_RHMI\_141]</sub>

[1\_RHMI\_142] The RoutineControl - Start Routine 'EntryOfSpecificCondition' - Request Message provides the possibility to request an entry of specific condition.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = EntryOfSpecificCondition [byte#1(MSB), byte#2]	'F204'H
5	routineControlOption[byte#1] = EntryType (see [1_RHMI_143])	'XX'H
6..9	routineControlOption[byte#2..byte#5] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
10..13	routineControlOption[byte#6..byte#9] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine 'EntryOfSpecificCondition', StartRoutine Request.

#### 4.10.1.1.1 Parameter EntryType<sub>[1\_RHMI\_352]</sub>

<sub>[1\_RHMI\_143]</sub>The EntryType defines the type of the specific condition entry as the data type SpecificConditionType for the VU (see [Annex 1C V2], Appendix 1).

#### 4.10.1.2 Routine 'EntryOfSpecificCondition', StartRoutine Positive Response<sub>[1\_RHMI\_144]</sub>

<sub>[1\_RHMI\_145]</sub>The RoutineControl - Start Routine 'EntryOfSpecificCondition' - Positive Response Message provides confirmation, if the remote specific condition entry request is correct.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = EntryOfSpecificCondition [byte#1(MSB), byte#2]	'F204'H

**Table:** Routine 'EntryOfSpecificCondition', StartRoutine Positive Response.

<sub>[1\_RHMI\_148]</sub>If 'OUT OF SCOPE' condition is open and EntryType = 'Ferry / Train crossing – Begin' or 'Ferry / Train crossing – End', then the VU shall reject the Routine 'EntryOfSpecificCondition', StartRoutine request via negative response 'conditionsNotCorrect'.

<sub>[1\_RHMI\_353]</sub>If the specific condition entry of the same type already stored for the current calendar minute as in EntryType within the 'RoutineControl - Start Routine 'EntryOfSpecificCondition' – Request' message then the VU shall reject this 'EntryOfSpecificCondition' request via negative response 'conditionsNotCorrect'.

#### 4.10.1.3 Routine 'EntryOfSpecificCondition', negative response code<sub>[1\_RHMI\_352a]</sub>

<sub>[1\_RHMI\_149]</sub>This definition contains the negative response conditions for routine 'EntryOfSpecificCondition', which are additional to the standard negative responses for the service RoutineControl.

Response Name	Value	Description
conditionsNotCorrect	'22'H	<p>The conditions for execution of remote specific condition entry are not met:</p> <ul style="list-style-type: none"> <li>• The Remote HMI session is closed</li> <li>• The VU is not activated;</li> <li>• A card insertion or withdrawal procedure is ongoing;</li> <li>• A card operation is ongoing;</li> </ul>

		<ul style="list-style-type: none"> <li>The entered EntryType is not allowed at the moment. See [1_RHMI_148] and [1_RHMI_353] for more information.</li> <li>Any other reason, when a specific condition entry is not possible or not allowed.</li> </ul>
requestOutOfRange	'31'H	<ul style="list-style-type: none"> <li>At least one parameter for the routine 'EntryOfSpecificCondition' is out of range.</li> <li>The request timestamp is outside of expected range. See [1_RHMI_226].</li> </ul>
invalidKey	'35'H	<p>The Remote HMI checksum is incorrect (see [1_RHMI_289]).</p> <p><b>Note:</b> The security as defined by VU PP is not involved here.</p>
requiredTimeDelayNotExpired	'37'H	The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288])..
requestCorrectlyReceived-ResponsePending	'78'H	The request was received correctly, and all parameters within the request were valid, but the request procedure is not yet completed and the VU is not yet ready to receive another request.

**Table:** Routine 'EntryOfSpecificCondition', negative response code .

## 4.11 Load / unload entries [1\_RHMI\_149h]

[1\_RHMI\_149a] The VU shall provide the possibility to enter a load/unload operation via RoutineControl service, routine 'LoadUnload'.

### 4.11.1 Routine 'LoadUnload' [1\_RHMI\_149i]

#### 4.11.1.1 Routine 'LoadUnload', StartRoutine Request [1\_RHMI\_149j]

[1\_RHMI\_149c] The RoutineControl - Start Routine 'LoadUnload' - Request Message provides the possibility to request an entry of load/unload operation.

Byte#	Parameter name	Value
1	serviceId = RoutineControl	'31'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = LoadUnload [byte#1(MSB), byte#2]	'F206'H
5	routineControlOption[byte#1] = OperationType (see [1_RHMI_149d])	'XX'H

Byte#	Parameter name	Value
6..9	routineControlOption[byte#2..byte#5] = RHMITimestamp (see [1_RHMI_339])	'XX'H ... 'XX'H
10..13	routineControlOption[byte#6..byte#9] = RHMICS (see [1_RHMI_279])	'XX'H ... 'XX'H

**Table:** Routine 'LoadUnload', StartRoutine Request.**4.11.1.1 Parameter OperationType** [1\_RHMI\_149k]

[1\_RHMI\_149d] The OperationType defines the type of the load/unload operation as data type OperationType for the VU (see [Annex 1C V2], Appendix 1).

**4.11.1.2 Routine 'LoadUnload', StartRoutine Positive Response** [1\_RHMI\_149j]

[1\_RHMI\_149e] The RoutineControl - Start Routine 'LoadUnload' - Positive Response Message provides confirmation, if the remote load/unload operation entry request is correct.

Byte#	Parameter name	Value
1	serviceId = RoutineControlPositiveResponse	'71'H
2	routineControlType = StartRoutine	'01'H
3, 4	routineIdentifier[] = LoadUnload [byte#1(MSB), byte#2]	'F206'H

**Table:** Routine 'LoadUnload', StartRoutine Positive Response.

[1\_RHMI\_149f] If a load/unload operation entry of the same type is already stored for the current calendar minute as in OperationType within the 'RoutineControl - Start Routine 'LoadUnload' – Request' message, then the VU shall reject this 'LoadUnload' request via negative response 'conditionsNotCorrect'.

[1\_RHMI\_149b] If the VU is not activated, then the VU shall reject the 'RoutineControl - Request Routine 'LoadUnload' requests by appropriate negative response.

**4.11.1.3 Routine 'LoadUnload', negative responses** [1\_RHMI\_149m]

[1\_RHMI\_149g] This definition contains the negative response conditions for routine 'LoadUnload', which are additional to the standard negative responses for the service RoutineControl.

Response Name	Value	Description
---------------	-------	-------------

conditionsNotCorrect	'22'H	The conditions for execution of remote load/unload operation entry are not met: <ul style="list-style-type: none"><li>• The Remote HMI session is closed;</li><li>• The VU is not activated;</li><li>• A card insertion or withdrawal procedure is ongoing;</li><li>• A card operation is ongoing;</li><li>• The entered load/unload operation is not allowed at the moment. See [1_RHMI_149f] for more information.</li><li>• Any other reason, when a load/unload operation entry is not possible or not allowed.</li></ul>
requestOutOfRange	'31'H	<ul style="list-style-type: none"><li>• At least one parameter for the routine 'LoadUnload' is out of range.</li><li>• The request timestamp is outside of expected range. See [1_RHMI_226].</li></ul>
invalidKey	'35'H	The Remote HMI checksum is incorrect (see [1_RHMI_289]). <b>Note:</b> The security as defined by VU PP is not involved here.
requiredTimeDelayNotExpired	'37'H	The previous request had invalid Remote HMI checksum and the timeout since then is not expired (see [1_RHMI_288])..
requestCorrectlyReceived-ResponsePending	'78'H	The request was received correctly, and all parameters within the request were valid, but the request procedure is not yet completed and the VU is not yet ready to receive another request.

**Table:** Routine 'LoadUnload', negative response code.

## 5 Annex [1\_PR\_1007a]

### 5.1 Printout [1\_PR\_1007b]

[1\_PR\_1007]

#### Printout block "Remote HMI sessions with remote entries":

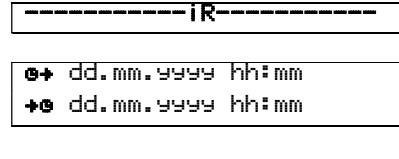
##### *Remote HMI sessions with remote entries*

Block identifier 'Stored Remote HMI sessions with remote entries'.

The begin and end timestamp periods, where Remote HMI session was active including remote user entries.

If a Remote HMI session is still ongoing, then the end timestamp shall be omitted.

If the Remote HMI session is recorded for driver and co-driver separately, then the VU may output both records even they are equal.



## 5.2 Data Type Definitions [1\_DaDi\_15]

### 5.2.1 DailyWorkPeriodEntryType [1\_DaDi\_4057]

[1\_DaDi\_3212]The DailyWorkPeriodEntryType defines the type of place entry where daily works begin/end.

DailyWorkPeriodEntryType ::= INTEGER(0..255)

#### Value assignment:

0: the place where a daily work period begins;

1: the place where a daily work ends.

### 5.2.2 DriverActivityChange [1\_DaDi\_4124]

[1\_DaDi\_3210]The DriverActivityChange is a change of the driver activity.

```
DriverActivityChange ::= SEQUENCE (
    activityStart    TimeReal,
    activity      INTEGER(0..255)
)
```

**activityStart** is the start of an activity as UTC time.

**Note:** Activity ends at the start of next following activity.

**activity** is a driver activity.

**Value assignment:**

**activityStart:** The VU shall truncate the activityStart to the start of the calendar minute.

E.g. activityStart at 5 min 59 s shall be truncated to exact 5 min.

**activity:**

- 0: BREAK/REST;
- 1: AVAILABILITY;
- 2: WORK;
- 4: UNKNOWN.

### **5.2.3 DriverCardDriver**<sub>[1\_DaDi\_4125]</sub>

<sub>[1\_DaDi\_2296]</sub>Status of a driver card within the tachograph.

DriverCardDriver ::= INTEGER(0..255)

**Value assignment:**

'00'H: no driver card is in the appropriate smart card slot;

'01'H: a driver card is in the appropriate smart card slot.

### **5.2.4 DriverCardDriver1**<sub>[1\_DaDi\_4126]</sub>

<sub>[1\_DaDi\_2300]</sub>DriverCardDriver1 provides the read out status of the driver 1 card.

DriverCardDriver1 ::= DriverCardDriver

### **5.2.5 DriverCardDriver2**<sub>[1\_DaDi\_4127]</sub>

<sub>[1\_DaDi\_2304]</sub>DriverCardDriver2 provides the read out status of the driver 2 (co-driver) card.

DriverCardDriver2 ::= DriverCardDriver

### **5.2.6 DriverPlace**<sub>[1\_DaDi\_4129]</sub>

<sub>[1\_DaDi\_3211]</sub>The DriverPlace is the begin/end place entry of the driver.

```
DriverPlace ::= SEQUENCE (
    entryTime          TimeReal,
    entryType          DailyWorkPeriodEntryType,
    dailyWorkPeriodCountry NationNumeric,
    dailyWorkPeriodRegion   RegionNumeric
)
```

**entryTime** is a date and time related to the place entry as UTC time.

**entryType** is the type of entry.

**dailyWorkPeriodCountry** is the country entered.

**dailyWorkPeriodRegion** is the region entered.

#### **Value assignment:**

**entryTime:** The VU shall truncate the place entry to the start of the calendar minute.

E.g. entryTime 5 min 59 s shall be truncated to exact 5 min.

**dailyWorkPeriodCountry:** The VU must not accept the dailyWorkPeriodCountry = 'no information available'.

**dailyWorkPeriodRegion:** The region shall be set according to the country in dailyWorkPeriodCountry.

### **5.2.7 ManualEntryActivitiesAndPlaces** [1\_DaDi\_4181]

[1\_DaDi\_3209] The ManualEntryActivitiesAndPlaces is the manual entry of the drivers' activities and places.

```
ManualEntryActivitiesAndPlaces ::= SEQUENCE (
    noOfActivities   INTEGER(0..255),
    driverActivities SET SIZE(noOfActivities) OF DriverActivityChange,
    noOfPlaces       INTEGER(0..255),
    driverPlaces     SET SIZE(noOfPlaces) OF DriverPlace
)
```

**noOfActivities** is the number of driver activity entries.

**driverActivities** is driver activity entries.

**noOfPlaces** is the number of driver place entries.

**driverPlaces** is driver place entries.

**Value assignment:**

**noOfActivities:**

Gain: 1;  
Offset: 0;  
Range: 1..40.

**noOfPlaces:**

Gain: 1;  
Offset: 0;  
Range: 0..5.

**driverActivities:**

- The DriverActivityChange within driverActivities shall be in ascending order, beginning with oldest activity change to the newest activity change.
- The VU must not accept more than one DriverActivityChange within same calendar minute.

**driverPlaces:**

- The DriverPlace wihtin driverPlaces shall be in ascending order, beginning with the oldest place entry to the newest place entry.
- The VU shall accept one begin place and one end place within same calendar minute.
- The VU must not accept more than one begin place or more than one end place within same calendar minute.

## 5.3 References to information defined by ISO 16844-7

[1\_RHMI\_176a]

[1\_DaDi\_3209a] Following information are requested to be available via local HMI by [Annex 1C V2], including the references of the same information available via calibration / diagnostic services.

<b>Local HMI Information (see [Annex 1C V2], chapter 3.15 'Displaying',</b>	<b>ISO 16844-7 reference</b>
UTC time and local time offset	TimeDate
odometer	HighResolutionTotalVehicleDistance
speed	TachographVehicleSpeed

<b>Local HMI Information (see [Annex 1C V2], chapter 3.15 'Displaying',</b>	<b>ISO 16844-7 reference</b>
the mode of operation	ModeOfOperation
the current activity of the driver	Driver1WorkingState
the current activity of the co-driver	Driver2WorkingState
driver continuous driving time	Driver1ContinuousDrivingTime
driver cumulative break time	Driver1CumulativeBreakTime
co-driver continuous driving time	Driver2ContinuousDrivingTime
co-driver cumulative break time	Driver2CumulativeBreakTime
the accumulated driving time of the driver for the previous and the current week	Driver1CumulatedDrivingTimePreviousAndCurrentWeek
the accumulated driving time of the co-driver for the previous and the current week	Driver2CumulatedDrivingTimePreviousAndCurrentWeek
the accumulated driving time of the driver for the current week	Driver1CurrentWeeklyDrivingTime
the accumulated driving time of the driver for the current daily work period	Driver1CurrentDailyDrivingTime
the accumulated driving time of the co-driver for the current daily work period	Driver2CurrentDailyDrivingTime
driver, the current duration of this activity (since it was selected)	Driver1CurrentDurationOfSelectedActivity
co-driver, the current duration of this activity (since it was selected)	Driver2CurrentDurationOfSelectedActivity
Ferry/Train condition	Not supported

<b>Local HMI Information (see [Annex 1C V2], chapter 3.15 'Displaying',</b>	<b>ISO 16844-7 reference</b>
Out of scope condition	OutOfScopeCondition
the content of any of the printouts listed in requirement 169 under the same formats as the printouts themselves	Not supported

**Note:** not all information is mandatory and therefore may be not available in the VU.

## 5.4 References [1\_RHMI\_176]

- ISO/DIS 16844-7, Road vehicles – Tachograph systems - Part 7: Parameters, 2021-03 (Draft International Standard, not yet released).
- ISO 14229-1: Road vehicles - Unified diagnostic services (UDS) - Part 1: Specification and requirements, 2013-03.
- ISO 7498-1: Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model, 1994
- Annex 1C V2 (not yet released): Smart Tachograph v2, ANNEX to the COMMISSION IMPLEMENTING REGULATION (EU) amending Commission Implementing Regulation (EU) 2016/799 of 18 March 2016 laying down the requirements for the construction, testing, installation, operation and repair of tachographs and their components.
- VU PP (not yet released).

## Transport protocol for ITS

Version 1.0

# 1 Table of Contents

<b>1</b>	<b>Table of Contents</b>	<b>2</b>
<b>2</b>	<b>History</b>	<b>4</b>
<b>3</b>	<b>Introduction</b>	<b>5</b>
3.1	Overview	5
3.2	Scope	5
<b>4</b>	<b>Serial Port Services Specification</b>	<b>6</b>
4.1	General	6
4.2	Serial Port Connection Setup	7
4.2.1	GAP Connection	7
4.2.2	GATT Service Discovery	9
4.2.3	Serial Port Connection	9
4.3	Serial Port Services	9
4.3.1	Service – Download	10
4.3.1.1	Characteristic – FIFO	10
4.3.1.1.1	Value	10
4.3.1.1.2	Client Configuration	10
4.3.1.2	Characteristic – Credits (Mandatory)	10
4.3.1.2.1	Value	10
4.3.1.2.2	Client Configuration	10
4.3.2	Service – Diagnostics	10
4.3.2.1	Characteristic – FIFO	11
4.3.2.1.1	Value	11
4.3.2.1.2	Client Configuration	11
4.3.2.2	Characteristic – Credits (Mandatory)	11
4.3.2.2.1	Value	11
4.3.2.2.2	Client Configuration	11
4.4	Bonding	11
4.5	Security	12
4.6	Connection Setup Diagrams	13
4.6.1	Connection with Flow Control (non-bonded)	13
4.6.2	Connection with Flow Control (bonded)	14
4.6.3	MTU Exchange	14

4.6.4 Data Exchange .....	15
<b>5 References .....</b>	<b>18</b>

## 2 History

Revision	Date	Author, Editor	Reason
1.0	2021-28-10	Smart Tachograph ITS Interface Working Group (ACEA, Continental, Intellic, Stoneridge, EU member states enforcement authorities, CORTE, IRU, ETF, DG MOVE).	Original version approved by the group

## 3 Introduction

### 3.1 Overview

In Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) systems, also known as Bluetooth Classic, there is a Serial Port Profile (SPP) which is defined by the Bluetooth Special Interest Group (SIG) and this profile is used to emulate a serial port connection over a Bluetooth wireless connection.

But in Bluetooth Low Energy, abbreviated as Bluetooth LE, a profile such as SPP is not defined. As a result, CORTE working group has to implement a custom serial port profile.

This document describes the serial port services (download and diagnostics) that will be implemented on top of the Generic Attribute Profile, abbreviated as GATT, to facilitate data transmission.

This document refers to a smartphone as an ITS unit (external device) that connects to the smart tachograph version 2 via Bluetooth Low Energy.

### 3.2 Scope

Inside the scope of this document:

- Serial Port Services (Download and Diagnostics)

Outside the scope of this document:

- Underlying protocol used for internally transferring the download and diagnostics data.
- Establishing Asynchronous Connection-Less Link (ACL)

## 4 Serial Port Services Specification

### 4.1 General

The serial port services are implemented on top of GATT. It consists of the following characteristics

- FIFO for reading and writing data.
- Tokens/Credits to simulate the Bluetooth Classic credit-based flow control.

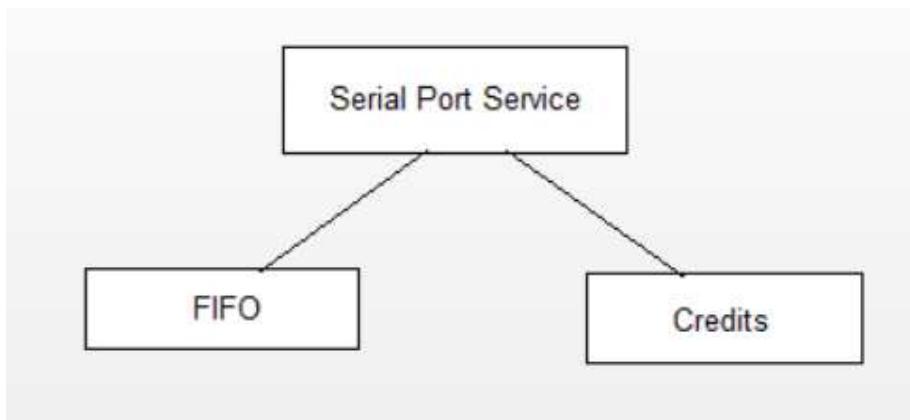


Figure 1: Serial Port Service

To establish a connection, the central/client (ex. Smartphone) has to set up the Asynchronous Connection-Less Link (ACL) and enable indications for both the FIFO and credits characteristics. The peripheral/server (tachograph) sends the number of credits to the client using the credit indication, and the client writes the number of credits to the server. The credits correspond to the number of packets the remote side may write before receiving new credits. Received credits are added to the remaining ones (if any).

Once connected and credits have been received, data transmission may start. The client writes to the FIFO and the server sends indications (as defined in Bluetooth specification) with the FIFO data.

It is recommended that the connection setup is done with a short connection interval. There are a few packets that must be transmitted for a complete serial port connection setup and a shorter connection interval decreases the total connect time. Once complete, the connection interval can be changed to a more appropriate one.

It is recommended to use a higher value of ATT Maximum Transmission Unit (MTU), which is the maximum length of an ATT packet, to increase data transmission throughput while also decreasing power consumption. The ATT MTU is defined by the L2CAP with a minimum value of 23 octets/bytes (including 3 bytes of ATT header). In Bluetooth 5.0 or higher, this value can be increased up to 512 bytes (as per Bluetooth Core Specification 5.0, Vol3, Part F, Section 3.2.9) using the feature Data Length Extension. During the initial connection setup, the central and peripheral may

exchange the maximum size of a packet that can be received using the Exchange MTU Request and Response PDUs. If the application data to be sent over Bluetooth LE is more than the size of ATT\_MTU, then such data will be sent out using multiple packets. Such multiple packet transfer is done at the Bluetooth LE transport layer level and, for the application layer this is seen as a single message transfer.

## 4.2 Serial Port Connection Setup

### 4.2.1 GAP Connection

GAP Connection is set up when

- the peripheral makes an advertisement and
- the central detects the advertisement and identifies the peripheral as the device it wants to connect to.

Note: Directed advertisement (advertisement only to a particular external device) shall not be supported. Frequency of advertisement by the tachograph and the connection parameters are VU manufacturer specific.

The smart tachograph version 2 performs the role of a peripheral and the external Bluetooth LE device such as a smartphone performs the role of a central.

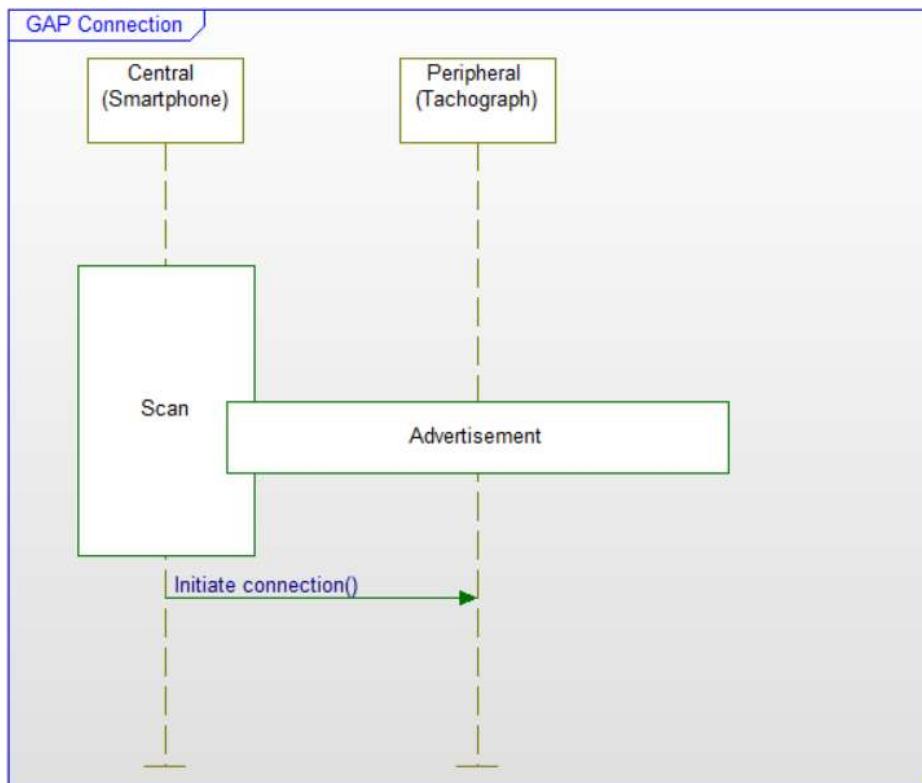


Figure 2: GAP connection setup

	Peripheral (Tachograph)	Central (Smartphone)
--	-------------------------	----------------------

<b>Physical layer functionality:</b>	<ul style="list-style-type: none"> <li>• Transmitter Characteristics</li> <li>• Receiver Characteristics</li> </ul>	<ul style="list-style-type: none"> <li>• Mandatory</li> <li>• Mandatory</li> </ul>	<ul style="list-style-type: none"> <li>• Mandatory</li> <li>• Mandatory</li> </ul>
<b>Link layer functionality:</b> <u>States</u>	<ul style="list-style-type: none"> <li>• Standby State</li> <li>• Advertising State</li> <li>• Scanning State</li> <li>• Initiating State</li> <li>• Connection State <ul style="list-style-type: none"> <li>- Slave role</li> <li>- Master role</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Mandatory</li> <li>• Mandatory</li> <li>• Excluded</li> <li>• Excluded</li> <li>• Mandatory</li> <li>• Excluded</li> <li>• Excluded</li> </ul>	<ul style="list-style-type: none"> <li>• Mandatory</li> <li>• Excluded</li> <li>• Mandatory</li> <li>• Mandatory</li> <li>• Excluded</li> <li>• Mandatory</li> </ul>
<b>Advertisement event types:</b>	<ul style="list-style-type: none"> <li>• Connectable and scannable undirected event</li> </ul>	<ul style="list-style-type: none"> <li>• Mandatory</li> </ul>	<ul style="list-style-type: none"> <li>• Excluded</li> </ul>

Note: Excluded is used for capabilities that may be supported by the unit but shall never be used

The peripheral (smart tachograph version 2) shall be in Discoverable mode unless specified.

When the ignition is turned off, the availability of Bluetooth LE in the smart tachograph version shall be made configurable.

- Configuration\_BluetoothOff -> Bluetooth is unavailable when ignition is off. No advertising as well.
- Configuration\_BluetoothOn -> Bluetooth is available for Configuration\_BluetoothOnH hours after ignition is off. After Configuration\_BluetoothOnH hours, Bluetooth will be turned off until next ignition on.

Note: Configuration\_BluetoothOnH is manufacturer specific.

The VU shall support the ITS connection to at least one ITS unit and may support connections to multiple ITS units at the same time (Appendix 13, ITS\_16). Supporting connection to multiple ITS units is optional and VU manufacturer specific.

The VU shall provide the possibility to support two services (diagnostics and download) per ITS unit, taking into account the respective access rights of the inserted card.

#### 4.2.2 GATT Service Discovery

Before a central (smartphone) makes a connection setup, it must find all relevant attribute handles. This can be done in several ways. Caching is recommended to keep the total connection setup time as low as possible for future connections.

The tachograph performs the role of a Server and the smartphone performs the role of a Client.

#### 4.2.3 Serial Port Connection

The peripheral (tachograph) contains services with FIFO and credits characteristics.

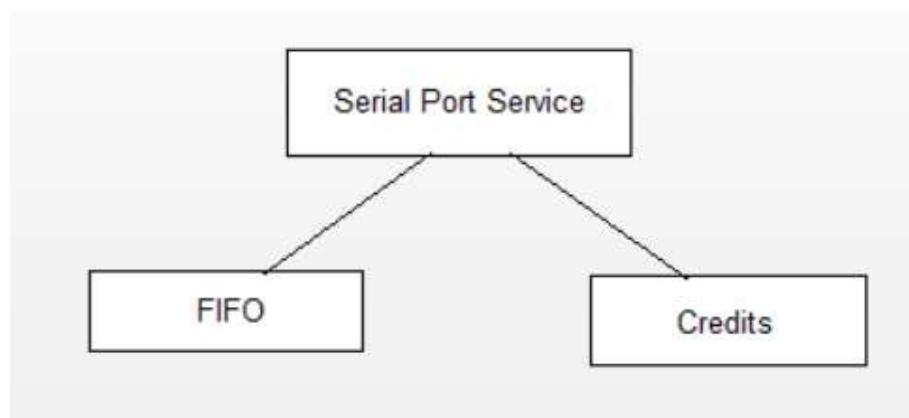
The FIFO characteristics are used for writing data from the central (smartphone) to the peripheral (tachograph) and to indicate data from the peripheral to the central.

The credits characteristics simulate the Bluetooth Classic credit-based flow control. It is allowed only to transmit the number of packets over air that matches the received number of credits. Received credits are added to the remaining ones. The receiving side must always keep available buffers that matches with the number of outgoing credits. The benefit of the credit-based flow control is that data will not be lost.

Note: Connection without flow control shall not be supported.

A well-defined serial port connection setup involves enabling indications and exchanging credits.

### 4.3 Serial Port Services



The serial port service consists of the following characteristics

1. FIFO – It represents the buffer that contains the data to be send to another Bluetooth LE device or the data that is received from another Bluetooth LE device that has to be transmitted to the application layer.
2. Credits – It is used to send and receive credits. For a credit-based connection, credits must have been received before sending data on the FIFO. Both client (via writing) and server (via indication) must send it.

A connection with flow control is established by the central sending credits to the peripheral. The peripheral accepts the connection by sending credits to the central. It may reject the connection by responding with credits data with value -1 (0xFF) to the central. A connection with flow control is disconnected by writing credits data with value -1 (0xFF). The disconnect is not confirmed.

Each characteristic contains two attributes – value and client configuration (according to Bluetooth specification)

1. Value
2. Client configuration

### **4.3.1 Service – Download**

UUID: eef90782-55dd-4388-b80b-695aba7a69b5

#### **4.3.1.1 Characteristic – FIFO**

##### **4.3.1.1.1 Value**

UUID: 29d3a479-1592-47df-80a4-afa742d369bb

Type: OCTET STRING (SIZE(ATT\_MTU))

Properties (Mandatory):

For the central (smartphone) – Write

For the peripheral (tachograph) – Indication

##### **4.3.1.1.2 Client Configuration**

Properties (Mandatory): Write (information to the client via indication)

#### **4.3.1.2 Characteristic – Credits (Mandatory)**

##### **4.3.1.2.1 Value**

UUID: db9c4128-bff3-41fe-a306-fb6f9a8aeb2d

Type: INTEGER (0..255)

Properties (Mandatory):

For the central (smartphone) – Write

For the peripheral (tachograph) – Indication

##### **4.3.1.2.2 Client Configuration**

Properties (Mandatory): Write (information to the client via indication)

### **4.3.2 Service – Diagnostics**

UUID: fa213def-aef4-475c-bcea-0a8d69073efc

#### **4.3.2.1 Characteristic – FIFO**

##### **4.3.2.1.1 Value**

UUID: e413960c-75ba-4ca9-8a67-99bc052a1b13

Type: OCTET STRING (SIZE(ATT\_MTU))

Properties (Mandatory):

For the central (smartphone) – Write

For the peripheral (tachograph) – Indication

##### **4.3.2.1.2 Client Configuration**

Properties (Mandatory): Write (information to the client via indication)

#### **4.3.2.2 Characteristic – Credits (Mandatory)**

##### **4.3.2.2.1 Value**

UUID: e168d1a6-304f-42b4-ab96-4cd1d4efebd9

Type: INTEGER (0..255)

Properties (Mandatory):

For the central (smartphone) – Write

For the peripheral (tachograph) – Indication

##### **4.3.2.2.2 Client Configuration**

Properties (Mandatory): Write (information to the client via indication)

## **4.4 Bonding**

After setting up the ACL, the client configuration characteristics have to be written before transferring credits or data. This leads to device bonding.

The purpose of bonding is to create a relation between two Bluetooth devices based on a common link key (a bond). The link key is created and exchanged (pairing) during the bonding procedure and is expected to be stored by both Bluetooth devices, to be used for future authentication. In addition to pairing, the bonding procedure can involve higher-layer initialization procedures.

Device bonding shall be made mandatory.

For bonded devices, the state of the client configuration characteristic shall be stored. The procedure to set up a connection is slightly different for bonded and non-bonded devices which is shown in the chapter 4.6.

## 4.5 Security

LE Secure Connection with Numeric Comparison shall be used to simplify the pairing procedure for the user and also to maintain/improve the security. This procedure protects from active or passive eavesdropping.

In this security model, the user is shown a six-digit number (from "000000" to "999999") on both displays (tachograph and smartphone) and then asked whether the numbers are the same on both devices. If the user enters "yes" on both devices, the pairing is successful.

## 4.6 Connection Setup Diagrams

### 4.6.1 Connection with Flow Control (non-bonded)

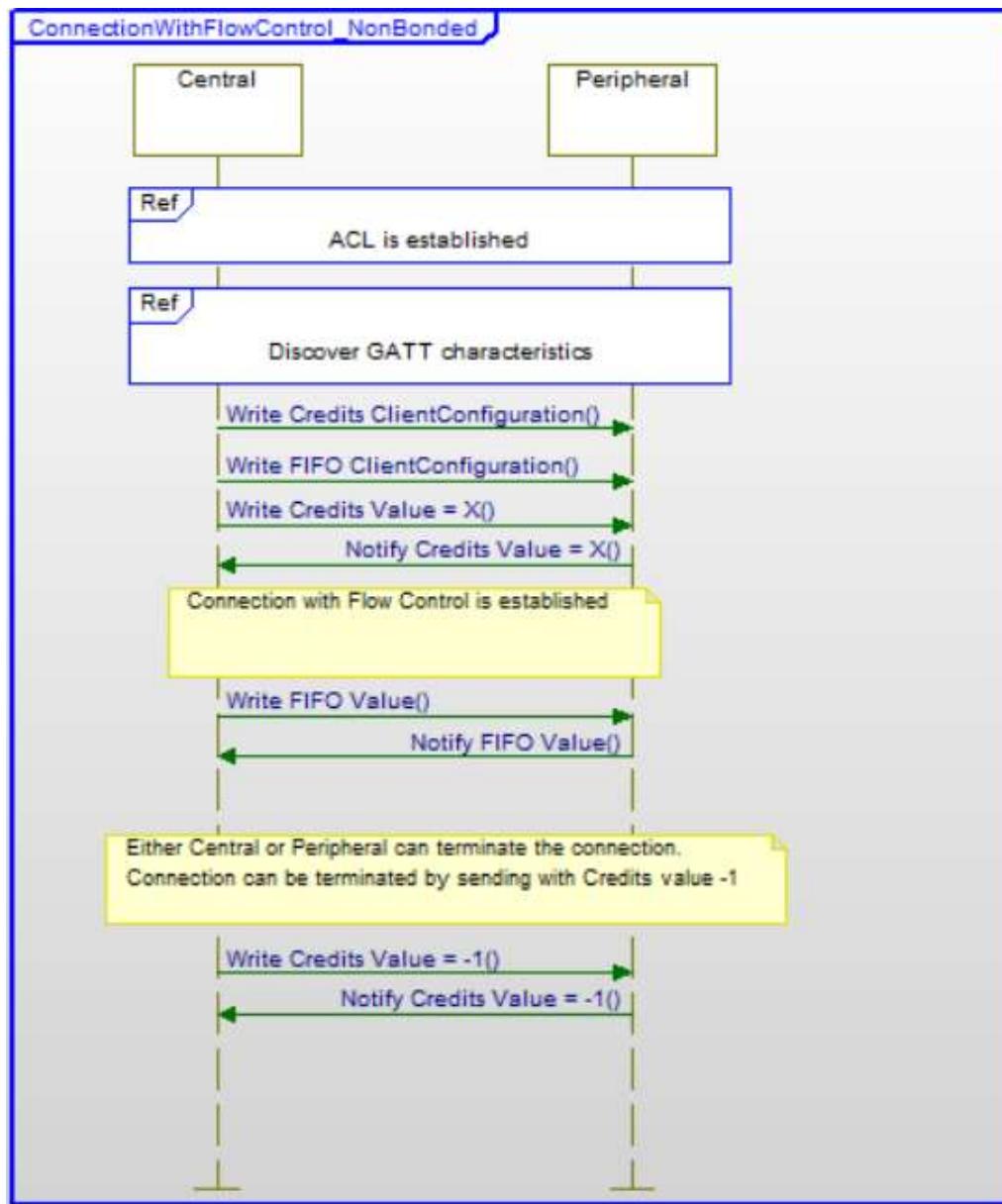


Figure 3: Connection with Flow Control (Non-Bonded Devices)

#### 4.6.2 Connection with Flow Control (bonded)

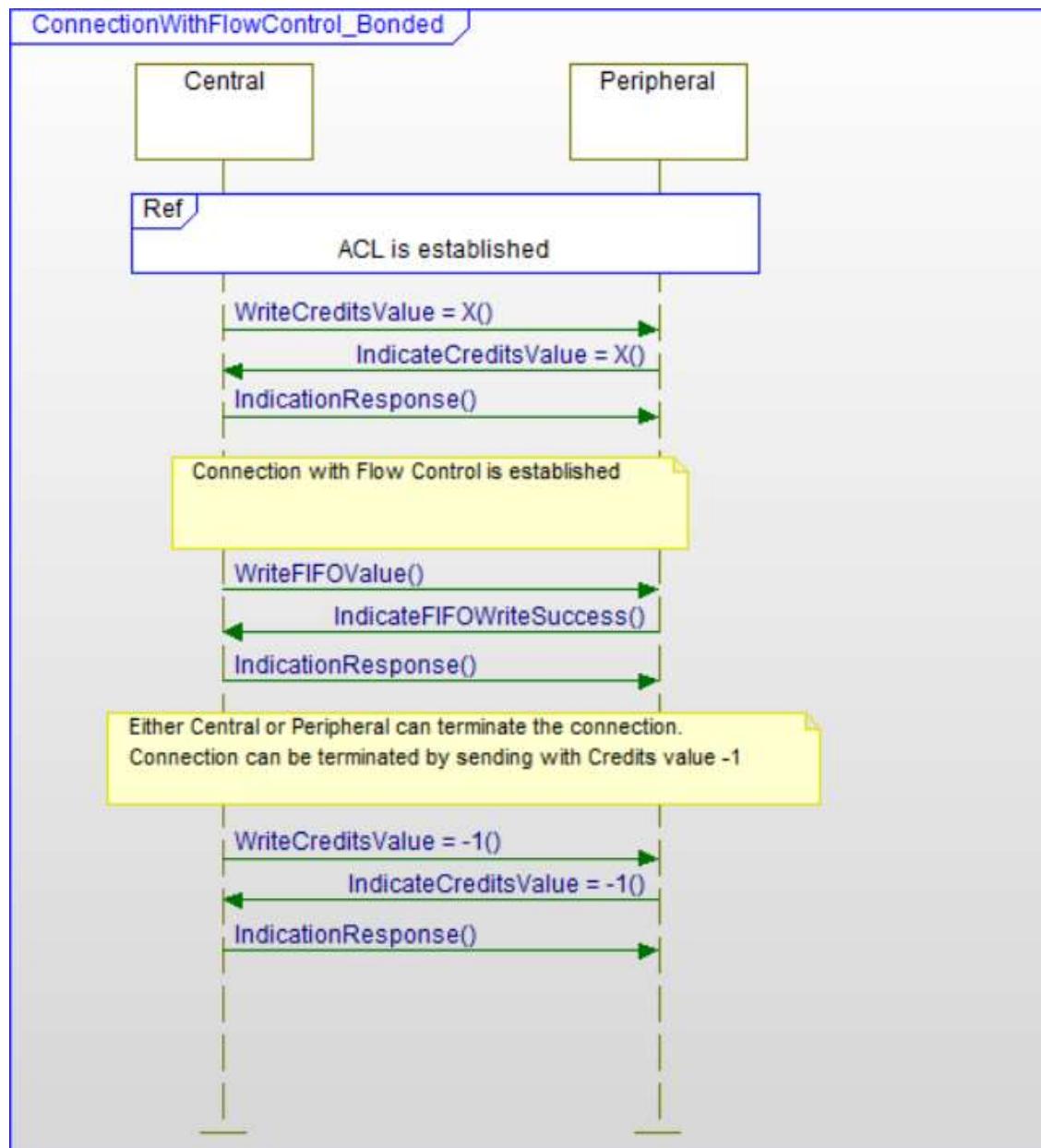


Figure 4: Connection with Flow Control (Bonded Devices)

#### 4.6.3 MTU Exchange

The central shall start the MTU exchange request-response during the connection setup. This is needed to decide the size of MTU used for the data packet transfer. Both, central and peripheral, shall share the maximum MTU size that they support. The minimum MTU size (not less than the default ATT\_MTU defined by the Bluetooth Specification) is used for subsequent data packet transfer.

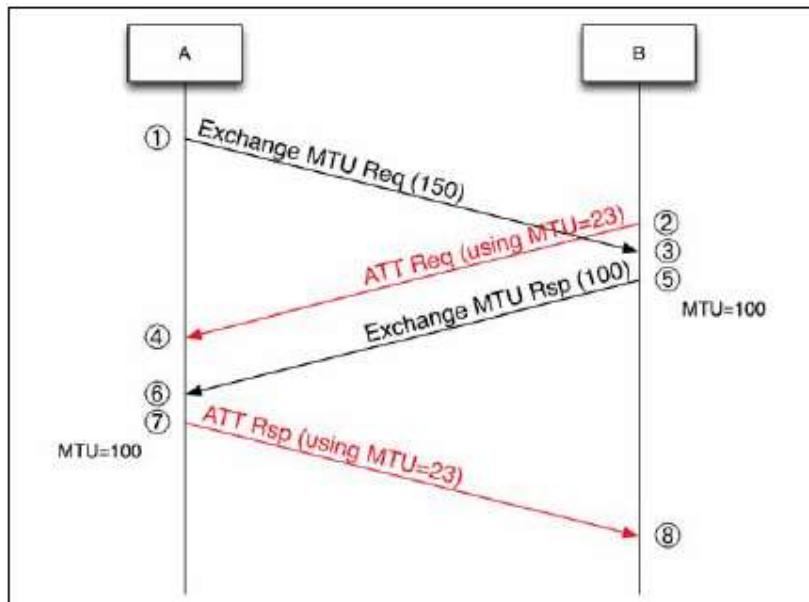


Figure 5: MTU Request and Response Exchange, Source: Bluetooth Core Specification 5.0, Vol 3, Part F, Section 3.4.2

#### 4.6.4 Data Exchange

If the application data to be sent over Bluetooth LE is more than the size of ATT\_MTU, then such data will be sent out using multiple packets.

First packet of every new message indicates the number of packets that will be sent along with the first packet.

ATT Header	Packet Information	Application Data
xx xx xx	AA 01	Data

AA = Total number of packets to be sent for the current data received from application layer.

Subsequent packets will have the packet number and application data.

ATT Header	Packet Information	Application Data
xx xx xx	00 02	Data
xx xx xx	00 03	Data
xx xx xx	00 NN	Data

For a default size of ATT\_MTU which is 23 bytes, NN can have a maximum value of 15.

## Smart Tachograph V2: Transport protocol for ITS

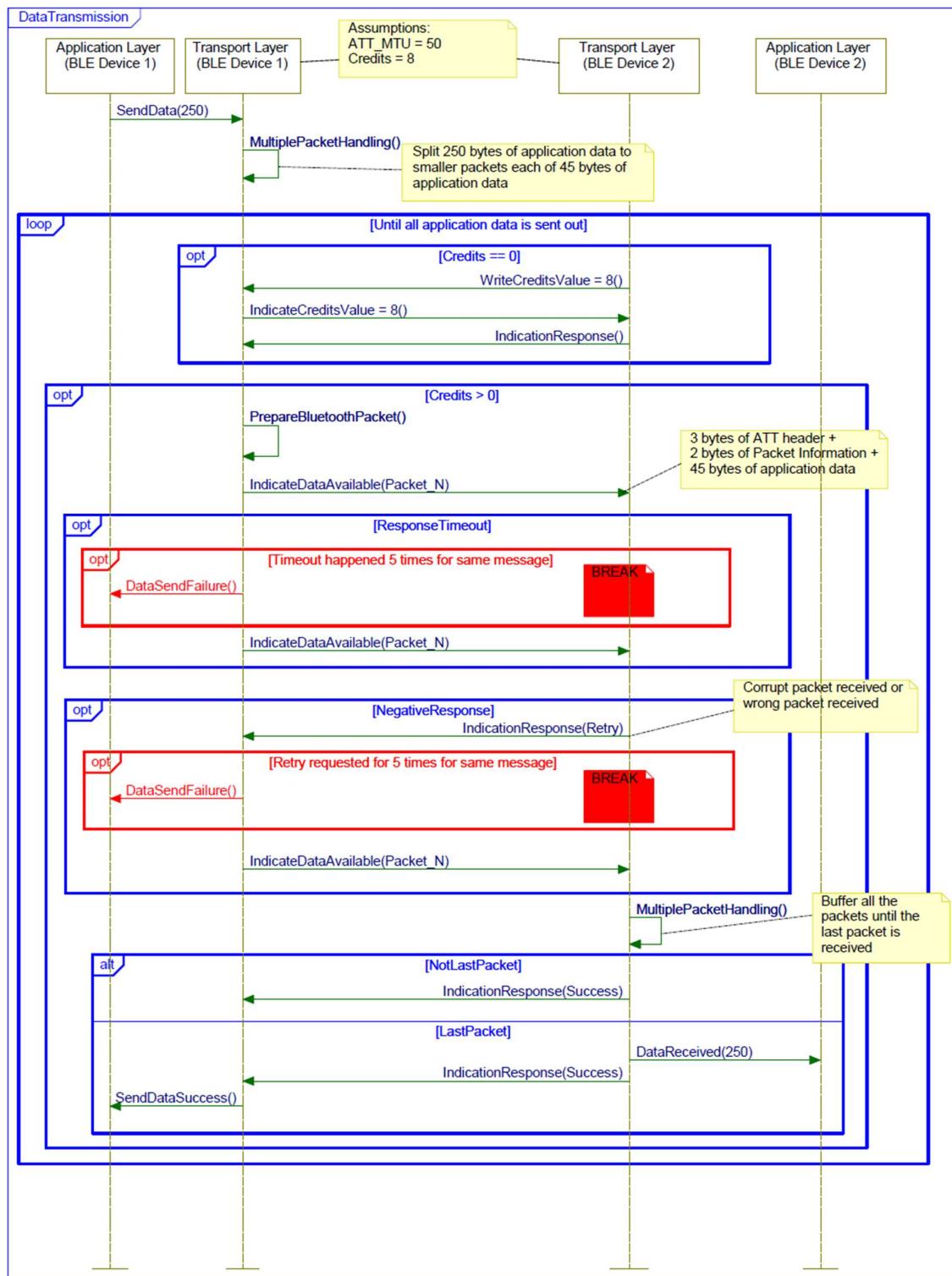


Figure 6: Data transmission example

## Smart Tachograph V2: Transport protocol for ITS

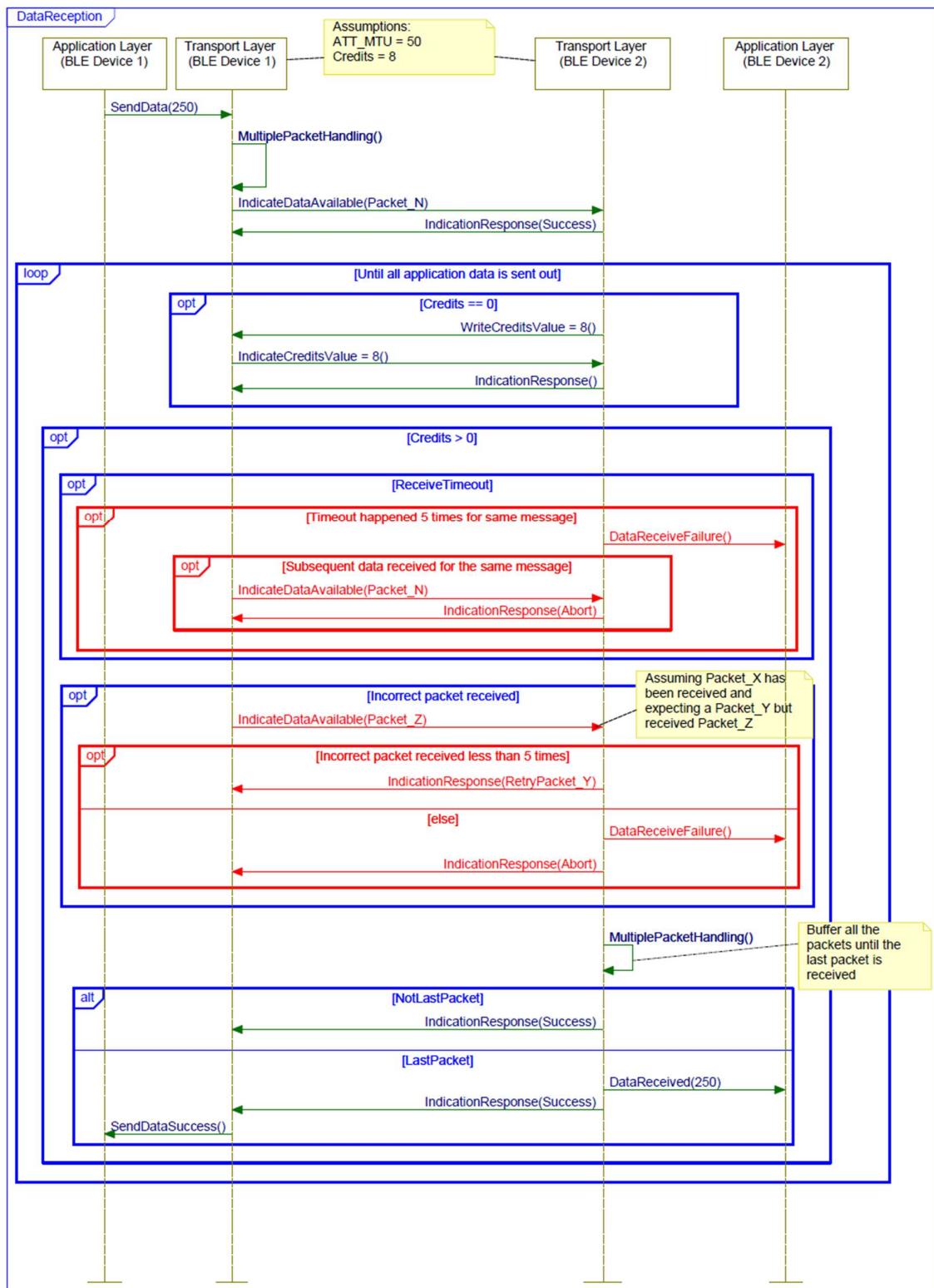


Figure 7: Data reception example

## 5 References

- u-blox Low Energy Serial Port Service Protocol Specification by u-blox
- Bluetooth® Core Specification v5.0