

ETSI TS 125 466 V8.1.0 (2008-01)

Technical Specification

**Universal Mobile Telecommunications System (UMTS);
UTRAN Iu-MS interface: Application part
(3GPP TS 25.466 version 8.1.0 Release 8)**



Reference

RTS/TSGR-0325466v810

Keywords

UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2008.
All rights reserved.

DECT™, PLUGTESTS™, UMTS™, TIPHON™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

Contents

| | |
|--|----|
| Intellectual Property Rights | 2 |
| Foreword..... | 2 |
| Foreword..... | 5 |
| 1 Scope | 6 |
| 2 References | 6 |
| 3 Definitions and abbreviations..... | 6 |
| 3.1 Definitions | 6 |
| 3.2 Abbreviations | 8 |
| 4 General | 8 |
| 4.1 Procedure specification principles..... | 8 |
| 4.2 Forwards and backwards compatibility..... | 8 |
| 4.3 Multi-antenna units..... | 8 |
| 4.4 Integer representation | 9 |
| 4.5 TMA Subunits | 9 |
| 5 Services expected from signalling transport..... | 9 |
| 5.1 Elementary procedure format | 9 |
| 5.1.1 Initiating message | 9 |
| 5.1.2 Response message | 9 |
| 6 Control elementary procedures | 10 |
| 6.1 State model..... | 10 |
| 6.2 General procedure handling | 11 |
| 6.2.1 Alarms | 11 |
| 6.2.2 Procedure message interpretation | 11 |
| 6.2.3 Parallel procedure handling | 11 |
| 6.3 Overview of elementary procedures..... | 12 |
| 6.4 Description of elementary procedures..... | 14 |
| 6.5 Common elementary procedures | 15 |
| 6.5.1 Reset Software | 15 |
| 6.5.2 Get Alarm Status..... | 16 |
| 6.5.3 Get Information | 16 |
| 6.5.4 Clear Active Alarms | 17 |
| 6.5.5 Alarm Subscribe | 18 |
| 6.5.6 Self Test..... | 18 |
| 6.5.7 Void | 19 |
| 6.5.8 Void | 19 |
| 6.5.9 Read User Data | 19 |
| 6.5.10 Write User Data | 20 |
| 6.5.11 Download Start | 21 |
| 6.5.12 Download Application..... | 21 |
| 6.5.13 Download End | 22 |
| 6.5.14 Vendor specific procedure | 23 |
| 6.6 Single-antenna elementary procedures..... | 23 |
| 6.6.1 Calibrate..... | 23 |
| 6.6.2 Send Configuration Data | 24 |
| 6.6.3 Set Tilt | 25 |
| 6.6.4 Get Tilt..... | 25 |
| 6.6.5 Alarm Indication..... | 26 |
| 6.6.6 Set Device Data | 27 |
| 6.6.7 Get Device Data..... | 27 |
| 6.7 Multi-antenna elementary procedures | 28 |
| 6.7.1 Antenna Calibrate | 28 |
| 6.7.2 Antenna Set Tilt..... | 29 |

| | | |
|-------------------------------|---|-----------|
| 6.7.3 | Antenna Get Tilt | 30 |
| 6.7.4 | Antenna Set Device Data | 30 |
| 6.7.5 | Antenna Get Device Data | 31 |
| 6.7.6 | Antenna Alarm Indication | 32 |
| 6.7.7 | Antenna Clear Active Alarms | 32 |
| 6.7.8 | Antenna Get Alarm Status | 33 |
| 6.7.9 | Antenna Get Number Of Antennas | 34 |
| 6.7.10 | Antenna Send Configuration Data | 34 |
| 6.8 | TMAAP Elementary procedures for TMA | 35 |
| 6.8.1 | TMA Set Mode | 35 |
| 6.8.2 | TMA Get Mode | 36 |
| 6.8.3 | TMA Get Supported Functions | 37 |
| 6.8.4 | TMA Set Gain | 38 |
| 6.8.5 | TMA Get Gain | 40 |
| 6.8.6 | TMA Set Device Data | 40 |
| 6.8.7 | TMA Get Device Data | 41 |
| 6.8.8 | TMA Alarm Indication | 42 |
| 6.8.8.1 | Further requirements | 43 |
| 6.8.9 | TMA Clear Active Alarms | 43 |
| 6.8.10 | TMA Get Alarm Status | 44 |
| 6.8.11 | TMA Get Number of Subunits | 44 |
| 6.8.12 | 3GPP Clear Active Alarms and Get Alarm Status | 45 |
| 6.8.13 | TMA Get Supported Non-Linear Gain Values | 45 |
| 7 | Unknown elementary procedures | 46 |
| Annex A (normative): | Return codes for secondary devices | 47 |
| Annex B (normative): | Assigned fields for additional data | 48 |
| Annex C (normative): | Procedure sequence for download of software to a secondary device..... | 51 |
| Annex D (informative): | Overview of elementary procedures..... | 52 |
| Annex E: | I-frame and INFO-field format (Informative) | 53 |
| Annex F (informative): | Change history | 54 |
| History | | 55 |

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document specifies the *Remote Electrical Tilting Application Part (RETAP)* between the implementation specific O&M transport function and the RET Antenna Control unit function of the Node B. The document also specifies the *Tower Mounted Amplifier Application Part (TMAAP)* between the implementation specific O&M transport function and the TMA control function of the Node B. It defines the Iuant interface and its associated signaling procedures.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TS 25.460: "UTRAN Iuant Interface: General Aspects and Principles".
- [2] ISO/IEC 13239 (2nd Edition, March 2000): "Information Technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures".
- [3] 3GPP TS 25.462: "UTRAN Iuant Interface: Signalling Transport".
- [4] 3GPP TS 25.461: 'UTRAN Iuant Interface: Layer 1'.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply.

Active alarm: An alarm which has an alarm state that has been raised, but not cleared

Alarm: Persistent indication of a fault

Alarm code: A code that identifies a specific alarm. The alarm code set is a subset of the return code set. The alarm codes are listed in annex A of this TS

Alarm state: A condition or state in the existence of an alarm. Alarm states are raised and cleared

ASCII character: A character forming part of the International Reference Version of the 7-bit character set defined in ISO/IEC 646:1991

Calibrate: Exercise the antenna drive unit over its entire range of travel to ensure fault-free operation and synchronise the measured and actual beam tilt of the antenna

Configuration data: A stored table or function defining the relationship between the physical position of the drive and electrical beam tilt

Data type: A definition determining the value range and interpretation of a series of octets. The following specified data types are used in this TS:

| Name: | Definition: |
|---------------|--|
| AlarmCode | 1 octet unsigned enumerated code All AlarmCode values are listed in annex A of this TS |
| FieldNumber | 1 octet unsigned enumerated code All field number values are listed in annex B of this TS |
| ProcedureCode | 1 octet unsigned enumerated code |
| ReturnCode | 1 octet unsigned enumerated code All ReturnCode values are listed in annex A of this TS |
| TextString | Octets with integer values in the range of 32 to 126 to be interpreted as ASCII characters |

Elementary procedure: The RETAP and TMAAP protocol consists of elementary procedures (EPs). An elementary procedure is a unit of interaction between the primary device (Node B) and the secondary devices (RET or TMAs devices)

An EP consists of an initiating message and possibly a response message.

Two kinds of EPs are used:

- **Class 1:** Elementary procedures with response (success or failure).
- **Class 2:** Elementary procedures without response.

For **Class 1** EPs, the types of responses can be as follows:

Successful

- A signalling message explicitly indicates that the elementary procedure has been successfully completed with the receipt of the response.

Unsuccessful

- A signalling message explicitly indicates that the EP failed.

Class 2 EPs are considered always successful.

Error: Deviation of a system from normal operation

Fault: Lasting error condition

Little endian: The order of transmission in which the least-significant octets of a multi-octet representation of a number are transmitted first. Little endian only applies to binary integer representations

MaxDataReceiveLength: SecondaryPayloadReceiveLength minus 3 octets (see subclause 4.8.1 in [3])

MaxDataTransmitLength: SecondaryPayloadTransmitLength minus 3 octets (see subclause 4.8.1 in [3])

Procedure code: A code identifying an elementary procedure

Reset: A process by which the device is put in the state it reaches after a completed power-up

Return code: A code which defines information about the outcome of an elementary procedure execution

Tilt (also downtilt, tilt angle, beamtilt): The elevation angle between the direction orthogonal to the antenna element axis and the maximum of its main beam in the elevation plane. A positive electrical tilt angle means that the antenna beam is directed below the direction orthogonal to the antenna axis. An antenna has separate values for electrical and mechanical tilt. The mechanical tilt is fixed by the geometry of the installation. In this TS the tilt referred to is always the electrical tilt unless otherwise stated

Tilt value: A signed integer used in elementary procedures to define the electrical tilt setting of the antenna. The tilt value is 10 times the antenna electrical tilt angle in degrees.

TMA: A TMA comprises a low noise amplifier together with its control and monitoring electronics and optional antenna modem.

TMA subunit: A TMA may comprise more than one TMA subunit. *All TMA subunits within one TMA have the same HDLC address and are addressable by an index via the application layer procedures.*

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

| | |
|-------|--|
| EP | Elementary Procedure |
| HDLC | High-Level Data Link Control |
| RET | Remote Electrical Tilting |
| TMA | Tower Mounted Amplifier |
| RETAP | Remote Electrical Tilting Application Part |
| TMAAP | Tower Mounted Amplifier Application Part |
| TCP | Time-Consuming Procedure |

4 General

4.1 Procedure specification principles

The principle for specifying the procedure logic is to specify the functional behaviour of the RET antenna control unit and TMA Subunits exactly and completely. The Node B functional behaviour is left unspecified.

The following specification principles have been applied for the procedure text in clause 6:

- The procedure text discriminates between:

- 1) Functionality which "shall" be executed

The procedure text indicates that the receiving node "shall" perform a certain function Y under a certain condition. If the receiving node supports procedure X but cannot perform functionality Y requested in the REQUEST message of a Class 1 EP, the receiving node shall respond with the message used to report unsuccessful outcome for this procedure, containing an appropriate cause value.

- 2) Functionality which "shall, if supported" be executed

The procedure text indicates that the receiving node "shall, if supported," perform a certain function Y under a certain condition. If the receiving node supports procedure X, but does not support functionality Y, the receiving node shall proceed with the execution of the EP, possibly informing the requesting node about the not supported functionality.

4.2 Forwards and backwards compatibility

The forwards and backwards compatibility of all versions of the protocol shall be assured by a mechanism in which all current and further messages will not be changed in the future. These parts can always be decoded regardless of the standard version.

New functionalities are added into the specification by introducing new procedures and thus the existing messages are not changed in the future.

4.3 Multi-antenna units

The RETAP elementary procedures are split into a single-antenna oriented part, a multi-antenna oriented part and a common part for both device types in order to support RET units controlling single- or multi-antenna devices. The RET unit responds, upon request, the number of antennas it controls. All multi-antenna oriented elementary procedures include a parameter stating which antenna the elementary procedure addresses. Antennas are numbered 1 and upwards.

4.4 Integer representation

Multi-octet integer values are transmitted in little endian order. Signed integers are represented as 2-complement values.

4.5 TMA Subunits

TMA subunits shall be numbered starting with 1 and proceeding upwards. The error message format for TMA procedures follows that of multiple RET devices.

5 Services expected from signalling transport

RETAP and TMAAP requires an assured in-sequence delivery service from the signalling transport and notification if the assured in-sequence delivery service is no longer available.

5.1 Elementary procedure format

Layer 2 provides a full-duplex link for the transmission of RETAP and TMAAP messages.

There are two types of RETAP and TMAAP elementary procedures:

Class 1: Initiating messages are sent either from the primary to a secondary device, or from a secondary to the primary device, in order to initiate some action within the receiving device. The other device sends a response message completing the procedure.

Class 2: Initiating messages are sent either from the primary to a secondary device, or from a secondary to the primary device. No response message is expected.

All RETAP and TMAAP messages use the same basic format:

Table 5.1.1: Basic format for all RETAP and TMAAP messages

| Elementary procedure | Number of data octets | Data |
|----------------------|-----------------------|--|
| 1 octet | 2 octets | MaxDataReceiveLength or MaxDataTransmitLength. |

NOTE: Response messages have the same basic format as initiating messages. The elementary procedure code shall be the same in the response message as in the associated initiating message.

5.1.1 Initiating message

The data part of an initiating message may contain parameters as specified in clause 6 of this TS.

5.1.2 Response message

Elementary procedures shall, unless otherwise specified, provide a response message within 1 second. The response time is measured from the time the message frame was received by the transport layer to the time the response message is ready for transfer by the transport layer.

If the class1 elementary procedure requested by the initiating message was successfully executed, the response message data part from a single-antenna device shall contain return code <OK>. Additional information may follow in the data part. The response message data part from a multi-antenna device starts with the antenna number followed by return code <OK> and optional additional information.

If the elementary procedure requested by the initiating message was not successfully executed, the response message data part from a single-antenna device shall contain return code <FAIL>.

The following octet shall contain a second return code which describes why the execution of the requested procedure failed. The response message data part from a multi-antenna device starts with the antenna number followed by return code <FAIL> and a second return code which describes why the execution of the requested procedure failed.

In some situations an initiating message can cause a change of operating conditions, for instance a SetTilt procedure might cause a RET device to discover that an adjuster is jammed or that a previously jammed adjuster works normally again. In these cases an alarm procedure reporting the change of operating conditions shall be used in addition to the regular <OK> or <FAIL> return codes in response message.

A complete annotated table of all return codes with their corresponding hexadecimal numbers is provided in annex A of this TS.

Return codes marked with an X in the Alarm column of annex A in this TS are used to report operating conditions in alarm procedures (see subclauses 6.6.5 and 6.7.6 for details).

6 Control elementary procedures

6.1 State model

The state model describing the secondary device is shown in figure 6.1 with procedures written in *italic*.

The relation to the connection state model for layer 2 can be found in [3].

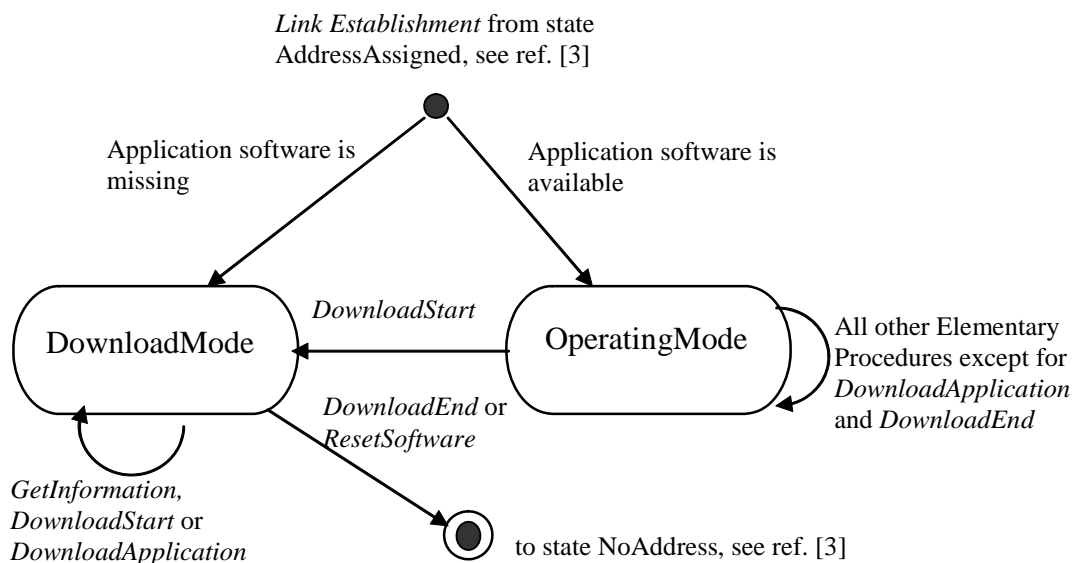


Figure 6.1: State model for the Secondary device

If an application software is not missing the secondary device enters the state **OperatingMode**.

If an application software is missing, the secondary device enters the state **DownloadMode**. In this state only software download functionality is supported in order to restore the application software.

The primary device will be notified that the secondary device has entered the state **DownloadMode** when a procedure which only is supported in the state **OperatingMode** fails with the return code **WorkingSoftwareMissing**.

If no software download functionality is supported, then only the state **OperatingMode** for the secondary device is supported.

6.2 General procedure handling

6.2.1 Alarms

When a fault is detected, the corresponding alarm state shall be changed to state *raised* by the secondary device. When the fault no longer exists, the corresponding alarm state shall be changed to state *cleared* by the secondary device. Alarm changes are reported through the AlarmIndication or AntennaAlarmIndication elementary procedures. Whenever an AlarmIndication or AntennaAlarmIndication elementary procedure message is transmitted, it shall contain all the alarm states changed that have not yet been reported as described in subclauses 6.6.5 and 6.7.6.

All alarm states shall be cleared by any type of reset.

6.2.2 Procedure message interpretation

The following message interpretation rules shall apply to a secondary device in the order mentioned:

- Any message shorter than 3 octets shall be disregarded. In case of Multi-Antenna-Procedures or TMA-Procedures, which uses a subunit field, any messages shorter than 4 octets shall be disregarded;
- If a message has a length inconsistent with its 'Number of data octets' field value it shall be responded with a failure message stating 'FormatError' as the cause of failure. The response message shall be to the initiating message identified by the procedure code;
- If a secondary device in the OperatingMode state receives a procedure message which is undefined for this device type, it shall respond with "Unknown Procedure";
- If a secondary device in the OperatingMode state is receiving a procedure message of an optional procedure not supported, it shall respond with a failure message stating 'UnsupportedProcedure' as the cause of failure;
- If a secondary device receives a procedure message, part of the software download procedure sequence described in Annex C, without having received the previous procedure messages in that sequence it shall respond with a failure message stating 'InvalidProcedureSequence' as the cause of failure;
- If a secondary device in the DownloadMode state is receiving a procedure message not supported in that state it shall respond with a failure message stating 'WorkingSoftwareMissing' as the cause of failure;
- If a message has a length inconsistent with the defined message length in the procedure definition it shall be responded with a failure message stating 'FormatError' as the cause of failure. The response message shall be to the initiating message identified by the procedure code;
- If a secondary device in the OperatingMode state is receiving a procedure message which addressed device subunit does not exist 'FormatError' shall be returned.

6.2.3 Parallel procedure handling

The secondary device shall support parallel execution of in maximum one additional EP only in parallel to one of the Time-Consuming Procedures defined in table 6.2.3.1:

Table 6.2.3.1: Definition of TCPs and the execution of procedures in parallel to a TCP

| Elementary Procedure | TCP | Execution in parallel to a TCP |
|-------------------------------------|-----------------|--------------------------------|
| <u>Common Procedure Set</u> | | |
| (Reserved) | | |
| Reset Software | No | mandatory |
| Get Alarm Status | No | mandatory |
| Get Information | No | mandatory |
| Clear Active Alarms | No | disallowed |
| Read User Data | No | optional |
| Write User Data | No | optional |
| Alarm Subscribe | No | optional |
| Self Test | Yes | disallowed |
| Download Start | No | disallowed |
| Download Application | No | disallowed |
| Download End | No | disallowed |
| Vendor specific procedure | vendor specific | optional |
| <u>Single-Antenna Procedure Set</u> | | |
| Set Device Data | No | optional |
| Get Device Data | No | optional |
| Calibrate | Yes | disallowed |
| Send Configuration Data | No | disallowed |
| Set Tilt | Yes | disallowed |
| Get Tilt | No | optional |
| Alarm Indication | No | optional |
| <u>Multi-Antenna Procedure Set</u> | | |
| Antenna Calibrate | Yes | optional |
| Antenna Send Configuration Data | No | disallowed |
| Antenna Set Tilt | Yes | optional |
| Antenna Get Tilt | No | optional |
| Antenna Set Device Data | No | optional |
| Antenna Get Device Data | No | optional |
| Antenna Alarm Indication | No | optional |
| Antenna Clear Active Alarms | No | disallowed |
| Antenna Get Alarm Status | No | mandatory |
| Antenna Get Number of Antennas | No | mandatory |

'yes' in the "TCP" column indicates that the procedure is a TCP, 'no' in the "TCP" column indicates that the procedure is not a TCP. 'mandatory' in the "Execution in parallel to a TCP" column indicates that the procedure shall be executed in parallel to an ongoing TCP. 'optional' in this column indicates, that the support of the execution of the procedure in parallel to an ongoing TCP is optional and 'disallowed' indicates that the procedure shall not be executed in parallel to a TCP.

If a secondary device receives an initiating message for an EP which cannot be executed due to the ongoing execution of other EPs, the secondary device shall respond with a failure message stating 'Busy' as the cause of failure.

Parallel execution of one TCP marked 'optional' in the "Execution in parallel to a TCP" column in table 6.2.3.1 may be supported for each antenna by the secondary device. The EPs AntennaSetTilt and AntennaCalibrate shall be executed in parallel only for different antenna numbers. If more than one TCP is executed, ResetSoftware shall be executed anyway and never be responded with 'Busy'.

If the EPs Get Tilt and Antenna GetTilt are executed in parallel with a TCP, their response message shall deliver a tilt value sampled during their execution.

TMAAP doesn't define any TCPs. Therefore parallel procedure handling is not supported by TMAAP.

6.3 Overview of elementary procedures

The set of elementary procedures for RET antennas and TMAs control provides procedure-oriented instructions. An overview of the procedures is given in annex D. Table 6.3.1 lists all common elementary procedures described in

subclause 6.5. Table 6.3.2 lists all RETAP elementary procedures specific for single-antenna device types described in subclause 6.6. Table 6.3.3 lists all RETAP elementary procedures specific for multi-antenna device types described in subclause 6.7. Table 6.3.4 lists all TMAAP elementary procedures specific for TMA device types described in subclause 6.8. Subclause 6.4 describes how to interpret the elementary procedure definitions in subclauses 6.5 to 6.8.

Some elementary procedures shall be performed in sequence as described in Annex C for the software download.

Table 6.3.1: Common elementary procedure set for all device types

| Elementary procedure | Requirement | Comment |
|---------------------------|-------------|---|
| Reset Software | Mandatory | |
| Get Alarm Status | Mandatory | |
| Get Information | Mandatory | |
| Clear Active Alarms | Mandatory | |
| Alarm Subscribe | Mandatory | |
| Read User Data | Mandatory | |
| Write User Data | Mandatory | |
| Self Test | Mandatory | |
| Download Start | Optional | This procedure is mandatory if the software download feature is supported |
| Download Application | Optional | This procedure is mandatory if the software download feature is supported |
| Download End | Optional | This procedure is mandatory if the software download feature is supported |
| Vendor specific procedure | Optional | |

Table 6.3.2: RETAP Elementary procedure set for single-antenna device type

| Elementary procedure | Requirement | Comment |
|-------------------------|-------------|---------|
| Calibrate | Mandatory | |
| Send Configuration Data | Mandatory | |
| Set Tilt | Mandatory | |
| Get Tilt | Mandatory | |
| Alarm Indication | Mandatory | |
| Set Device Data | Mandatory | |
| Get Device Data | Mandatory | |

Table 6.3.3: RETAP Elementary procedure set for multi-antenna device type

| Elementary procedure | Requirement | Comment |
|---------------------------------|-------------|---------|
| Antenna Calibrate | Mandatory | |
| Antenna Send Configuration Data | Mandatory | |
| Antenna Set Tilt | Mandatory | |
| Antenna Get Tilt | Mandatory | |
| Antenna Set Device Data | Mandatory | |
| Antenna Get Device Data | Mandatory | |
| Antenna Alarm Indication | Mandatory | |
| Antenna Clear Active Alarms | Mandatory | |
| Antenna Get Alarm Status | Mandatory | |
| Antenna Get Number Of Antennas | Mandatory | |

Table 6.3.4: TMAAP elementary procedure set for TMA devices

| Elementary procedure | Requirement | Comment |
|------------------------------------|-------------|---|
| TMASetMode | Optional | Shall only be supported if the TMA subunit supports bypass mode |
| TMAGetMode | Mandatory | |
| TMAGetSupportedFunctions | Mandatory | |
| TMASetGain | Optional | Shall only be supported if the TMA subunit supports variable gain |
| TMAGetGain | Mandatory | |
| TMASetDeviceData | Mandatory | |
| TMAGetDeviceData | Mandatory | |
| TMAAlarmIndication | Mandatory | |
| TMAClearActiveAlarms | Mandatory | |
| TMAGetAlarmStatus | Mandatory | |
| TMAGetNumberOfSubunits | Mandatory | |
| TMAGetSupportedNonLinearGainValues | Optional | Shall only be supported if the TMA subunit supports variable gain with non-linear steps |

6.4 Description of elementary procedures

Table 6.4.1: Description of elementary procedures

| | | | | |
|---|---|---|---|--|
| Name: The name used to refer to the elementary procedure | | | | |
| Code: The code is defined here. All other code references are informative | Issued by: Primary device or secondary device | Procedure class: Class 1 or Class 2 | DownloadMode state: Defines whether the procedure shall be supported in the DownloadMode state. | Power mode: Defines the secondary device power consumption as described in [4] during the execution of the elementary procedure. |

Table 6.4.2: Initiating and response message parameters and format

| Number | Length | Type | Description |
|--|--|---|--------------------------------------|
| The enumerated order in which the parameter occurs in the data field of the message. The first number is 1. | The length of the parameter, in number of octets, if defined. | The data type used in the parameter. | Description of the parameter. |

Table 6.4.3: Response message parameters and format for common class 1 elementary procedures upon error

| Number | Length | Type | Description |
|----------|----------------|-------------------|---------------------------|
| 1 | 1 octet | ReturnCode | Return code FAIL |
| 2 | 1 octet | ReturnCode | Reason for failure |

Table 6.4.4: Response message parameters and format for single-antenna class 1 elementary procedures upon error

| Number | Length | Type | Description |
|----------|----------------|-------------------|---------------------------|
| 1 | 1 octet | ReturnCode | Return code FAIL |
| 2 | 1 octet | ReturnCode | Reason for failure |

Table 6.4.5: Response message parameters and format for multi-antenna class 1 and TMA subunit class 1 elementary procedures upon error

| Number | Length | Type | Description |
|--------|---------|------------------|--------------------------------------|
| 1 | 1 octet | Unsigned integer | Antenna number or TMA subunit number |
| 2 | 1 octet | ReturnCode | Return code FAIL |
| 3 | 1 octet | ReturnCode | Reason for failure |

NOTE 1: The response message in the elementary procedure AntennaGetAntennaNumber, has the format given in table 6.4.4, although it is defined as a multi-antenna class 1 elementary procedure.

NOTE 2: The response message in the elementary procedure TMAGetNumberOfSubunits has the format given in table 6.4.4 as class 1 elementary procedure.

Table 6.4.6: Return codes

| OK | FAIL | Comment |
|---|---|---------------------------------------|
| All return codes applicable in a response message to a successful procedure, except 'OK', are listed here. The return codes are listed by name as defined in annex A. | All return codes applicable in a response message to a failing procedure, except 'FAIL' are listed here. The return codes are listed by name as defined in annex A. | Any comment needed for clarification. |

6.5 Common elementary procedures

6.5.1 Reset Software

Table 6.5.1.1: Elementary procedure Reset Software

| | | | | |
|-------------------------------|-------------------------------------|------------------------------|-----------------------------------|---------------------------|
| Name: ResetSoftware | | | | |
| Code: 0x03 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: Yes | Power mode: Low |

Table 6.5.1.2: Initiating message parameters and format for Reset Software

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.5.1.3: Response message parameters and format for Reset Software

| Number | Length | Type | Description |
|--------|---------|------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On the receipt of the initiating message the secondary device shall reset the application. All alarm states shall be cleared.

If the initiating message is received in the OperatingMode state, the transport layer shall remain unaffected.

If the initiating message is received in the DownloadMode state, the ResetSoftware procedure shall reset the entire device without activating any new application software downloaded since entering the DownloadMode state.

The device shall not execute the reset procedure before transport layer acknowledgement through sequence number update is received for the response.

Table 6.5.1.4: Return codes for Reset Software

| | | |
|----|-------------|--|
| OK | FAIL | Comment |
| | FormatError | In case of format error, the procedure code validity is not secured. |

6.5.2 Get Alarm Status

Table 6.5.2.1: Elementary procedure Get Alarm Status

| | | | | |
|--------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: GetAlarmStatus | | | | |
| Code: 0x04 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.5.2.2: Initiating message parameters and format for Get Alarm Status

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.5.2.3: Response message parameters and format for Get Alarm Status

| Number | Length | Type | Description |
|--------------|----------------|-------------------|------------------------------|
| 1 | 1 octet | ReturnCode | Return code OK |
| i + 1 | 1 octet | AlarmCode | Active alarm number i |

$i = 1 \dots N$

Description:

On receipt of the initiating message the secondary device reports the alarm codes of the active alarms.

Table 6.5.2.4: Return codes for Get Alarm Status

| | | |
|--|---------------------------------------|---------|
| OK | FAIL | Comment |
| All return codes marked as used for alarms in Annex A. | FormatError WorkingSoftwareMissing | |

6.5.3 Get Information

Table 6.5.3.1: Elementary procedure Get Information

| | | | | |
|--------------------------------|-------------------------------------|------------------------------|-----------------------------------|---------------------------|
| Name: GetInformation | | | | |
| Code: 0x05 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: Yes | Power mode: Low |

Table 6.5.3.2: Initiating message parameters and format for Get Information

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.5.3.3: Response message parameters and format for Get Information

| Number | Length | Type | Description |
|--------|---------|------------------|---|
| 1 | 1 octet | ReturnCode | Return code OK |
| 2 | 1 octet | Unsigned integer | Length of parameter 3 in number of octets |
| 3 | | TextString | Product number |
| 4 | 1 octet | Unsigned integer | Length of parameter 5 in number of octets |
| 5 | | TextString | Serial number |
| 6 | 1 octet | Unsigned integer | Length of parameter 7 in number of octets |
| 7 | | TextString | Hardware Version |
| 8 | 1 octet | Unsigned integer | Length of parameter 9 in number of octets |
| 9 | | TextString | Software Version |

Description:

On receipt of the initiating message the secondary device shall return the product number ProdNr and the serial number SerNr of the secondary device. If known, also the hardware version and the software version may be returned. The software version should indicate the version number of the currently executed software.

The parameters HWVersion and SWVersion in the response message refer to the version designators of the hardware and installed software of the secondary device. If the application is missing or no HW or SW version number is found, then an empty string shall be returned as the HW or SW version number. The empty string is represented as a length field equals 0 and no octets in the TextString field.

The response message length shall be less than or equal to the minimum SecondaryPayloadTransmitLength as given in subclause 4.8.1 in [3].

Table 6.5.3.4: Return codes for Get Information

| OK | FAIL | Comment |
|----|-------------|---------|
| | FormatError | |

6.5.4 Clear Active Alarms**Table 6.5.4.1: Elementary procedure Clear Active Alarms**

| | | | | |
|-----------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: ClearActiveAlarms | | | | |
| Code: 0x06 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.5.4.2: Initiating message parameters and format for Clear Active Alarms

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.5.4.3: Response message parameters and format for Clear Active Alarms

| Number | Length | Type | Description |
|--------|---------|------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall first clear all stored alarm information and then return a procedure response message.

Table 6.5.4.4: Return codes for Clear Active Alarms

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy WorkingSoftwareMissing | |

6.5.5 Alarm Subscribe

Table 6.5.5.1: Elementary procedure Alarm Subscribe

| | | | | |
|--------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AlarmSubscribe | | | | |
| Code: 0x12 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.5.5.2: Initiating message parameters and format for Alarm Subscribe

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.5.5.3: Response message parameters and format for Alarm Subscribe

| Number | Length | Type | Description |
|----------|----------------|-------------------|-----------------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall start reporting alarms to the primary device.

Table 6.5.5.4: Return codes for Alarm Subscribe

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy WorkingSoftwareMissing | |

6.5.6 Self Test

Table 6.5.6.1: Elementary procedure Self Test

| | | | | |
|--------------------------|-------------------------------------|------------------------------|----------------------------------|----------------------------|
| Name: SelfTest | | | | |
| Code: 0x0A | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: High |

Table 6.5.6.2: Initiating message parameters and format for Self Test

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.5.6.3: Response message parameters and format for Self Test

| Number | Length | Type | Description |
|--------|---------|------------|---|
| 1 | 1 octet | ReturnCode | Return code OK |
| i + 1 | 1 octet | AlarmCode | Alarm code for alarm i detected during self test. |

i = 1 ... N

Description:

On receipt of the initiating message the secondary device shall execute a test procedure which may include a check of physical and processor functions. The specific tests to be performed are implementation specific, and may include the movement of the adjuster, which shall not exceed +-5% of total available tilting range starting from the current adjuster position.

The response message of the secondary device on the procedure provides information on detected faults or, if no fault is detected, with confidence that the operation of the device is normal in all respects.

During the test the operational parameters of the device shall not change beyond operationally acceptable limits and on completion all parameters shall be returned to their initial values.

In the normal response message, after the self test was executed successfully, the return codes are set to report possible detected faults during the self test. If no faults are detected, this shall be signalled by no return codes following the return code <OK>.

In the case of a failure response message, the self test could not be executed successfully and the reported return code relates to the inability of the device to perform the requested self-test operation.

Table 6.5.6.4: Return codes for Self Test

| OK | FAIL | Comment |
|---|---|---------|
| All return codes marked as alarms in annex A. | FormatError Busy WorkingSoftwareMissing NotCalibrated NotScaled | |

6.5.7 Void

6.5.8 Void

6.5.9 Read User Data

Table 6.5.9.1: Elementary procedure Read User Data

| | | | | |
|------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: ReadUserData | | | | |
| Code: 0x10 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.5.9.2: Initiating message parameters and format for Read User Data

| Number | Length | Type | Description |
|--------|----------|------------------|--------------------------|
| 1 | 2 octets | Unsigned integer | Memory offset |
| 2 | 1 octet | Unsigned integer | Number of octets to read |

NOTE: Number of octets to read shall be less than, or equal to MaxDataTransmit Length minus 1.

Table 6.5.9.3: Response message parameters and format for Read User Data

| Number | Length | Type | Description |
|--------|---|---------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |
| 2 | Number of octets given by parameter 2 of the initiating message | User specific | User data |

Description:

On receipt of the initiating message the secondary device shall send back user specific data stored in a user data area to the primary device.

The user data area is intended for storage of user defined data, e.g. inventory information.

Table 6.5.9.4: Return codes for Read User Data

| OK | FAIL | Comment |
|----|---|---|
| | FormatError WorkingSoftwareMissing OutOfRange | The return code OutOfRange is used, if the given memory address range is outside the valid address space. |

6.5.10 Write User Data**Table 6.5.10.1: Elementary procedure Write User Data**

| | | | | |
|-------------------------------|-------------------------------------|------------------------------|---------------------------|---------------------------|
| Name: WriteUserData | | | | |
| Code: 0x11 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.5.10.2: Initiating message parameters and format for Write User Data

| Number | Length | Type | Description |
|--------|--|------------------|---------------------------|
| 1 | 2 octets | Unsigned integer | Memory offset |
| 2 | 1 octet | Unsigned integer | Number of octets to write |
| 3 | Message specific, given by parameter 2 | User specific | Data to write |

NOTE: Number of octets to write shall be less than, or equal to MaxDataReceiveLength minus 3.

Table 6.5.10.3: Response message parameters and format for Write User Data

| Number | Length | Type | Description |
|--------|---------|------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall store user data in non-volatile memory. The user data is stored in the user data area using the relative memory address offset given in the initiating message and starting with zero.

The user data area is intended for storage of user defined data, e.g. inventory information.

Table 6.5.10.4: Return codes for Write User Data

| OK | FAIL | Comment |
|----|---|---|
| | FormatError Busy WorkingSoftwareMissing HardwareError OutOfRange | The return code OutOfRange is used if the given memory address range is outside the valid address space. |

6.5.11 Download Start

Table 6.5.11.1: Elementary procedure Download Start

| | | | | |
|-------------------------------|-------------------------------------|------------------------------|-----------------------------------|---------------------------|
| Name: DownloadStart | | | | |
| Code: 0x40 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: Yes | Power mode: Low |

Table 6.5.11.2: Initiating message parameters and format for Download Start

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.5.11.3: Response message parameters and format for Download Start

| Number | Length | Type | Description |
|----------|----------------|-------------------|-----------------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of this initiating message the software download process shall be initiated. Following transition to the DownloadMode state, the secondary device sends return code <OK>. Previous subscription of alarms by use of the AlarmSubscribe procedure is cancelled.

Table 6.5.11.4: Return codes for Download Start

| OK | FAIL | Comment |
|----|--|---------|
| | FormatError Busy UnsupportedProcedure | |

6.5.12 Download Application

Table 6.5.12.1: Elementary procedure Download Application

| | | | | |
|-------------------------------------|-------------------------------------|------------------------------|-----------------------------------|---------------------------|
| Name: DownloadApplication | | | | |
| Code: 0x41 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: Yes | Power mode: Low |

Table 6.5.12.2: Initiating message parameters and format for Download Application

| Number | Length | Type | Description |
|----------|--|------------------------|----------------------|
| 1 | Less than, or equal to MaxDataReceiveLength | Vendor specific | Software data |

Table 6.5.12.3: Response message parameters and format for Download Application

| Number | Length | Type | Description |
|--------|---------|------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

This elementary procedure is used once or several times to transfer software data from the primary device to the secondary device.

Table 6.5.12.4: Return codes for Download Application

| OK | FAIL | Comment |
|----|--|---------|
| | FormatError Busy HardwareError InvalidFileContent InvalidProcedureSequence | |

6.5.13 Download End**Table 6.5.13.1: Elementary procedure Download End**

| | | | | |
|-----------------------------|-------------------------------------|------------------------------|-----------------------------------|---------------------------|
| Name: DownloadEnd | | | | |
| Code: 0x42 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: Yes | Power mode: Low |

Table 6.5.13.2: Initiating message parameters and format for Download End

| Number | Length | Type | Description |
|--------|----------|------|-----------------|
| None | 0 octets | None | No data carried |

Table 6.5.13.3: Response message parameters and format for Download End

| Number | Length | Type | Description |
|--------|---------|------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

This elementary procedure signals the end of a multi-message data transfer to the secondary device. The secondary device shall respond after verifying the received data. The secondary device shall reset autonomously after completion of the layer 2 response and activate the new application software.

Table 6.5.13.4: Return codes for Download End

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy HardwareError ChecksumError InvalidFileContent InvalidProcedureSequence | |

6.5.14 Vendor specific procedure

Table 6.5.14.1: Elementary procedure Vendor Specific Procedure

| | | | | |
|---|--------------------------------------|--|---|---------------------------------------|
| Name: VendorSpecificProcedure | | | | |
| Code: 0x90 | Issued by: Vendor specific | Procedure class: Vendor specific | DownloadMode state: Vendor specific | Power mode: Vendor specific |

Table 6.5.14.2: Initiating message parameters and format for Vendor Specific Procedure

| Number | Length | Type | Description |
|--------------|------------------------|------------------------|--------------------|
| 1 | 2 octets | ASCII | Vendor code |
| 1 + i | Vendor specific | Vendor specific | |

$i = 1 \dots N$

Table 6.5.14.3: Response message parameters and format for Vendor Specific Procedure

| Number | Length | Type | Description |
|----------|------------------------|------------------------|------------------------|
| i | Vendor specific | Vendor specific | Vendor specific |

$i = 1 \dots N$

Description:

The vendor specific procedure is intended for vendor specific purposes like e.g. testing.

Table 6.5.14.4: Return codes for vendor specific procedure

| OK | FAIL | Comment |
|----|---|--|
| | FormatError UnsupportedProcedure | If the Vendor code in the initiating message does not match that of the RET device, UnsupportedProcedure shall be returned. |

6.6 Single-antenna elementary procedures

6.6.1 Calibrate

Table 6.6.1.1: Elementary procedure Calibrate

| | | | | |
|---------------------------|-------------------------------------|------------------------------|----------------------------------|----------------------------|
| Name: Calibrate | | | | |
| Code: 0x31 | Issued by: Primary Device | Procedure class: 1 | DownloadMode state: No | Power mode: High |

Table 6.6.1.2: Initiating message parameters and format for Calibrate

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.6.1.3: Response message parameters and format for Calibrate

| Number | Length | Type | Description |
|----------|----------------|-------------------|-----------------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall perform a calibration of the RET antenna where the actuator is driven through its whole tilt range.

The response time to this Calibrate procedure shall be less than 4 minutes.

Table 6.6.1.4: Return codes for Calibrate

| OK | FAIL | Comment |
|----|--|---------|
| | FormatError Busy HardwareError WorkingSoftwareMissing MotorJam ActuatorJam NotConfigured UnsupportedProcedure | |

6.6.2 Send Configuration Data

Table 6.6.2.1: Elementary procedure Send Configuration Data

| | | | | |
|---------------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: SendConfigurationData | | | | |
| Code: 0x32 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.6.2.2: Initiating message parameters and format for Send Configuration Data

| Number | Length | Type | Description |
|----------|--|------------------------|---------------------------|
| 1 | Less than, or equal to MaxDataReceiveLength | Vendor specific | Configuration data |

Table 6.6.2.3: Response message parameters and format for Send Configuration Data

| Number | Length | Type | Description |
|----------|----------------|-------------------|-----------------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall store the provided vendor and antenna specific configuration data for the relationship between the movement of the drive system and the beam tilt position of the antenna.

If the configuration data exceeds MaxDataReceiveLength, the data shall be split into a number of MaxDataReceiveLength segments and one final segment with whatever is left. The primary device transmits the segments in order. The layer 2 sequence numbers guarantee that no segment will be lost or received out of order.

Table 6.6.2.4: Return codes for Send Configuration Data

| OK | FAIL | Comment |
|----|--|---------|
| | FormatError Busy HardwareError WorkingSoftwareMissing ChecksumError InvalidFileContent UnsupportedProcedure | |

6.6.3 Set Tilt

Table 6.6.3.1: Elementary procedure Set Tilt

| | | | | |
|-------------------------|-------------------------------------|------------------------------|----------------------------------|----------------------------|
| Name: SetTilt | | | | |
| Code: 0x33 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: High |

Table 6.6.3.2: Initiating message parameters and format for Set Tilt

| Number | Length | Type | Description |
|----------|-----------------|----------------|-------------------|
| 1 | 2 octets | Signed integer | Tilt value |

Table 6.6.3.3: Response message parameters and format for Set Tilt

| Number | Length | Type | Description |
|----------|----------------|-------------------|-----------------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall set the electrical tilt in increments of 0.1°.

The secondary device shall respond to the initiating message in less than 2 minutes.

The tilt value corresponding to the actual tilt angle shall not go outside of the range between the tilt value corresponding to the current tilt angle and the tilt value corresponding to the requested tilt angle by more than 5 during this operation.

The format of the value of parameter 1 is given in subclause 3.1.

Table 6.6.3.4: Return codes for Set Tilt

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy HardwareError WorkingSoftwareMissing MotorJam ActuatorJam NotConfigured NotCalibrated OutOfRange UnsupportedProcedure | |

6.6.4 Get Tilt

Table 6.6.4.1: Elementary procedure Get Tilt

| | | | | |
|-------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: GetTilt | | | | |
| Code: 0x34 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.6.4.2: Initiating message parameters and format for Get Tilt

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.6.4.3: Response message parameters and format for Get Tilt

| Number | Length | Type | Description |
|--------|----------|----------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |
| 2 | 2 octets | Signed integer | Tilt value |

Description:

On receipt of the initiating message the secondary device shall return the current tilt value.

The returned tilt value is given in the format specified in subclause 3.1.

Table 6.6.4.4: Return codes for Get Tilt

| OK | FAIL | Comment |
|----|--|---|
| | FormatError Busy HardwareError WorkingSoftwareMissing NotCalibrated NotConfigured UnsupportedProcedure | HardwareError shall only be used, if error is detected in tilt detector. |

6.6.5 Alarm Indication

Table 6.6.5.1: Elementary procedure Alarm Indication

| | | | | |
|---------------------------------|---------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AlarmIndication | | | | |
| Code: 0x07 | Issued by: Secondary device | Procedure class: 2 | DownloadMode state: No | Power mode: Low |

Table 6.6.5.2: Initiating message parameters and format for Alarm Indication

| Number | Length | Type | Description |
|---------|---------|------------------|----------------------------|
| 2 i – 1 | 1 octet | Unsigned integer | Return code i; see annex A |
| 2 i | 1 octet | Unsigned integer | State flag i |

i = 1 ... N

Description:

The secondary device uses this procedure to report alarm state changes to the primary device. This procedure shall only be performed if the secondary has performed an AlarmSubscribe procedure since its latest reset.

For each alarm, the current alarm state and alarm code shall be reported if and only if any change in its state has occurred during the period of time since the last reported state. An AlarmIndication procedure shall be performed if at least one alarm shall be reported. The first AlarmIndication procedure after the AlarmSubscribe procedure shall report the active alarms.

Alarm state changes are considered as reported at the time the message is passed to the transport layer.

State flag = 0 represents alarm state *cleared*.

State flag = 1 represents alarm state *raised*.

6.6.6 Set Device Data

Table 6.6.6.1: Elementary procedure Set Device Data

| | | | | |
|-------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: SetDeviceData | | | | |
| Code: 0x0E | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.6.6.2: Initiating message parameters and format for Set Device Data

| Number | Length | Type | Description |
|----------|--------------------|-------------------------|----------------------------------|
| 1 | 1 octet | Unsigned integer | Field number, see annex B |
| 2 | See annex B | See annex B | Data to write |

Table 6.6.6.3: Response message parameters and format for Set Device Data

| Number | Length | Type | Description |
|----------|----------------|-------------------|-----------------------|
| 1 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall write the data given in the parameters of the initiating message into the fields optionally provided for configuration data and listed in annex B of this TS. If an attempt is made to write to fields which are designated as read only, the return code *ReadOnly* is returned and the data for those fields is ignored. If an attempt is made to write to fields which are not supported by the device the return code *UnknownParameter* is returned and the data for those fields is ignored.

Table 6.6.6.4: Return codes for Set Device Data

| OK | FAIL | Comment |
|----|--|---------|
| | FormatError Busy WorkingSoftwareMissing HardwareError ReadOnly UnknownParameter | |

6.6.7 Get Device Data

Table 6.6.7.1: Elementary procedure Get Device Data

| | | | | |
|-------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: GetDeviceData | | | | |
| Code: 0x0F | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.6.7.2: Initiating message parameters and format for Get Device Data

| Number | Length | Type | Description |
|----------|----------------|-------------------------|----------------------------------|
| 1 | 1 octet | Unsigned integer | Field number; see annex B |

Table 6.6.7.3: Response message parameters and format for Get Device Data

| Number | Length | Type | Description |
|--------|-------------|-------------|----------------|
| 1 | 1 octet | ReturnCode | Return code OK |
| 2 | See annex B | See annex B | Field value |

Description:

In this procedure the secondary device shall return the data stored in the field for configuration data specified by the field number in the procedure and listed in annex B of this TS.

Table 6.6.7.4: Return codes for Get Device Data

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy WorkingSoftwareMissing UnknownParameter | |

6.7 Multi-antenna elementary procedures

6.7.1 Antenna Calibrate

Table 6.7.1.1: Elementary procedure Antenna Calibrate

| | | | | |
|----------------------------------|-------------------------------------|------------------------------|----------------------------------|----------------------------|
| Name: AntennaCalibrate | | | | |
| Code: 0x80 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: High |

Table 6.7.1.2: Initiating message parameters and format for Antenna Calibrate

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Antenna number |

Table 6.7.1.3: Response message parameters and format for Antenna Calibrate

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall perform a calibration of the antenna addressed by the antenna number. During calibration the actuator is driven through the whole tilt range of the antenna.

The response time to this Antenna Calibrate procedure shall be less than 4 minutes.

Table 6.7.1.4: Return codes for Antenna Calibrate

| OK | FAIL | Comment |
|----|--|---|
| | FormatError Busy HardwareError WorkingSoftwareMissing MotorJam ActuatorJam NotConfigured UnsupportedProcedure | If the addressed antenna is not existing, FormatError is returned. |

6.7.2 Antenna Set Tilt

Table 6.7.2.1: Elementary procedure Antenna Set Tilt

| | | | | |
|--------------------------------|-------------------------------------|------------------------------|----------------------------------|----------------------------|
| Name: AntennaSetTilt | | | | |
| Code: 0x81 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: High |

Table 6.7.2.2: Initiating message parameters and format for Antenna Set Tilt

| Number | Length | Type | Description |
|----------|-----------------|-------------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 2 octets | Signed integer | Tilt value |

Table 6.7.2.3: Response message parameters and format for Antenna Set Tilt

| Number | Length | Type | Description |
|----------|----------------|-------------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall set the electrical tilt of the antenna addressed by the antenna number in increments of 0.1°.

The secondary device shall respond to the initiating message in less than 2 minutes.

The tilt value corresponding to the actual tilt angle shall not go outside of the range between the tilt value corresponding to the current tilt angle and the tilt value corresponding to the requested tilt angle by more than 5 during this operation.

The format of the value of parameter 2 is given in subclause 3.1.

Table 6.7.2.4: Return codes for Antenna Set Tilt

| OK | FAIL | Comment |
|----|---|---|
| | FormatError Busy HardwareError WorkingSoftwareMissing MotorJam ActuatorJam NotConfigured NotCalibrated OutOfRange UnsupportedProcedure | If the addressed antenna is not existing, FormatError is returned. |

6.7.3 Antenna Get Tilt

Table 6.7.3.1: Elementary procedure Antenna Get Tilt

| | | | | |
|--------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaGetTilt | | | | |
| Code: 0x82 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.7.3.2: Initiating message parameters and format for Antenna Get Tilt

| Number | Length | Type | Description |
|----------|----------------|--------------------------|-----------------------|
| 1 | 1 octet | Unsigned interger | Antenna number |

Table 6.7.3.3: Response message parameters and format for Antenna Get Tilt

| Number | Length | Type | Description |
|----------|-----------------|-------------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |
| 3 | 2 octets | Signed integer | Tilt value |

Description:

On receipt of the initiating message the secondary device shall return the current tilt value of the antenna addressed by the antenna number.

The returned tilt value is in the format specified in subclause 3.1.

Table 6.7.3.4: Return codes for Antenna Get Tilt

| OK | FAIL | Comment |
|----|---|---|
| | FormatError Busy HardwareError WorkingSoftwareMissing NotConfigured NotCalibrated UnsupportedProcedure | If the addressed antenna is not existing, FormatError is returned. HardwareError shall only be used, if an error is detected in tilt detector. |

6.7.4 Antenna Set Device Data

Table 6.7.4.1: Elementary procedure Antenna Set Device Data

| | | | | |
|--------------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaSetDeviceData | | | | |
| Code: 0x83 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.7.4.2: Initiating message parameters and format for Antenna Set Device Data

| Number | Length | Type | Description |
|----------|--------------------|-------------------------|----------------------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | Unsigned integer | Field number; see annex B |
| 3 | See annex B | See annex B | Data to write |

Table 6.7.4.3: Response message parameters and format for Antenna Set Device Data

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall write the provided data for the antenna addressed by the antenna number into the fields optionally provided for configuration data and listed in annex B of this TS. If an attempt is made to write to fields which are designated as read only for the addressed antenna the return code ReadOnly is returned and the data for those fields is ignored. If an attempt is made to write to fields which are not supported for the addressed antenna the return code UnknownParameter is returned and the data for those fields is ignored.

Table 6.7.4.4: Return codes for Antenna Set Device Data

| OK | FAIL | Comment |
|----|--|--|
| | FormatError Busy HardwareError WorkingSoftwareMissing ReadOnly UnknownParameter UnsupportedProcedure | If the addressed antenna is not existing, FormatError is returned. |

6.7.5 Antenna Get Device Data**Table 6.7.5.1: Elementary procedure Antenna Get Device Data**

| | | | | |
|--------------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaGetDeviceData | | | | |
| Code: 0x84 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.7.5.2: Initiating message parameters and format for Antenna Get Device Data

| Number | Length | Type | Description |
|--------|---------|------------------|-----------------------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | Unsigned integer | Field number to read; see annex B |

Table 6.7.5.3: Response message parameters and format for Antenna Get Device Data

| Number | Length | Type | Description |
|--------|-------------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |
| 3 | See annex B | See annex B | Field value |

Description:

On receipt of the initiating message the secondary device shall return the data stored for the addressed antenna in the field for configuration data specified by the field number in the initiating message and listed in annex B of this TS.

Table 6.7.5.4: Return codes for Antenna Get Device Data

| OK | FAIL | Comment |
|----|--|---|
| | FormatError Busy WorkingSoftwareMissing UnsupportedProcedure UnknownParameter | If the addressed antenna is not existing, FormatError is returned. |

6.7.6 Antenna Alarm Indication

Table 6.7.6.1: Elementary procedure Antenna Alarm Indication

| | | | | |
|--|---------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaAlarmIndication | | | | |
| Code: 0x85 | Issued by: Secondary device | Procedure class: 2 | DownloadMode state: No | Power mode: Low |

Table 6.7.6.2: Initiating message parameters and format for Antenna Alarm Indication

| Number | Length | Type | Description |
|---------------|----------------|-------------------------|-----------------------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 i | 1 octet | Unsigned integer | Return code i; see annex A |
| 2 i +1 | 1 octet | Unsigned integer | State flag i |

$i = 1 \dots N$

Description:

The multi-antenna secondary device uses this procedure to report antenna alarm state changes to the primary device. This procedure shall only be performed if the secondary has performed an AlarmSubscribe procedure since its latest reset. Multi-antenna devices shall use this AntennaAlarmIndication procedure only for multi-antenna specific alarms and the AlarmIndication procedure in subclause 6.6.5 for the other alarms.

For each alarm, the current alarm state and alarm code shall be reported if and only if any change in its state has occurred during the period of time since the last reported state. An AntennaAlarmIndication procedure shall be performed if at least one multi-antenna specific alarm shall be reported. The first AntennaAlarmIndication procedure after the AlarmSubscribe procedure shall report the active alarms.

Alarm state changes are considered as reported at the time the message is passed to the transport layer.

State flag = 0 represents alarm state *cleared*.

State flag = 1 represents alarm state *raised*.

6.7.7 Antenna Clear Active Alarms

Table 6.7.7.1: Elementary procedure Antenna Clear Active Alarms

| | | | | |
|--|---------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaClearActiveAlarms | | | | |
| Code: 0x86 | Issued by: Secondary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.7.7.2: Initiating message parameters and format for Antenna Clear Active Alarms

| Number | Length | Type | Description |
|----------|----------------|-------------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |

Table 6.7.7.3: Response message parameters and format for Antenna Clear Active Alarms

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall first clear all stored alarm information for the addressed antenna and then return a procedure response message.

Table 6.7.7.4: Return codes for Antenna Clear Active Alarms

| OK | FAIL | Comment |
|----|---|--|
| | FormatError Busy WorkingSoftwareMissing UnsupportedProcedure | If the addressed antenna is not existing, FormatError is returned. |

6.7.8 Antenna Get Alarm Status**Table 6.7.8.1: Elementary procedure Antenna Get Alarm Status**

| | | | | |
|---------------------------------------|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaGetAlarmStatus | | | | |
| Code: 0x87 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.7.8.2: Initiating message parameters and format for Antenna Get Alarm Status

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Antenna number |

Table 6.7.8.3: Response message parameters and format for Antenna Get Alarm Status

| Number | Length | Type | Description |
|--------|---------|------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |
| i + 2 | 1 octet | AlarmCode | Active alarm number i |

i = 1 ... N

Description:

On receipt of the initiating message the secondary device shall report the alarm codes of the active alarms for the addressed antenna.

Table 6.7.8.4: Return codes for Antenna Get Alarm Status

| OK | FAIL | Comment |
|---|---|--|
| All return codes marked as used for alarms in Annex A | FormatError WorkingSoftwareMissing UnsupportedProcedure | If the addressed antenna is not existing, FormatError is returned. |

6.7.9 Antenna Get Number Of Antennas

Table 6.7.9.1: Elementary procedure Antenna Get Number Of Antennas

| | | | | |
|--|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaGetNumberOfAntennas | | | | |
| Code: 0x88 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.7.9.2: Initiating message parameters and format for Antenna Get Number Of Antennas

| Number | Length | Type | Description |
|-------------|-----------------|-------------|------------------------|
| None | 0 octets | None | No data carried |

Table 6.7.9.3: Response message parameters and format for Antenna Get Number Of Antennas

| Number | Length | Type | Description |
|----------|----------------|-------------------------|---------------------------|
| 1 | 1 octet | ReturnCode | Return code OK |
| 2 | 1 octet | Unsigned integer | Number of antennas |

Description:

On receipt of the initiating message the secondary device shall return the number of antennas it controls.

Table 6.7.9.4: Return codes for Antenna Get Number Of Antennas

| OK | FAIL | Comment |
|----|--|---------|
| | FormatError WorkingSoftwareMissing UnsupportedProcedure | |

6.7.10 Antenna Send Configuration Data

Table 6.7.10.1: Elementary procedure Antenna Send Configuration Data

| | | | | |
|--|-------------------------------------|------------------------------|----------------------------------|---------------------------|
| Name: AntennaSendConfigurationData | | | | |
| Code: 0x89 | Issued by: Primary device | Procedure class: 1 | DownloadMode state: No | Power mode: Low |

Table 6.7.10.2: Initiating message parameters and format for Antenna Send Configuration Data

| Number | Length | Type | Description |
|----------|---|-------------------------|---------------------------|
| 1 | 1 octet | Unsigned Integer | Antenna number |
| 2 | Less than, or equal to MaxDataReceiveLength minus 1 | Vendor specific | Configuration data |

Table 6.7.10.3: Response message parameters and format for Antenna Send Configuration Data

| Number | Length | Type | Description |
|----------|----------------|-------------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Antenna number |
| 2 | 1 octet | ReturnCode | Return code OK |

Description:

On receipt of the initiating message the secondary device shall store the provided vendor and antenna specific configuration data for the relationship between the movement of the drive system and the beam tilt position of the addressed antenna.

If the configuration data exceeds MaxDataReceiveLength minus 1, the data shall be split into a number of MaxDataReceiveLength minus 1 segments and one final segment with whatever is left. The primary device transmits the segments in order. The layer 2 sequence numbers guarantee that no segment will be lost or received out of order.

Table 6.7.10.4: Return codes for Antenna Send Configuration Data

| OK | FAIL | Comment |
|----|--|---|
| | FormatError Busy HardwareError WorkingSoftwareMissing ChecksumError InvalidFileContent UnsupportedProcedure | If the addressed antenna is not existing, FormatError is returned. |

6.8 TMAAP Elementary procedures for TMA

6.8.1 TMA Set Mode

The TMA Set Mode procedure shall only be supported if the TMA subunit can be set in bypass mode. On receipt of the initiating message, the secondary device shall first set the TMA subunit in the appropriate mode as indicated by the mode state flag, and then return a response message. If a TMA subunit in bypass mode receives the elementary procedure TMASetMode to Bypass, the TMA subunit shall remain in the Bypass mode and the response OK shall be returned.

Mode state flag = 0 represents *Normal mode*.

Mode state flag = 1 represents *Bypass mode*.

Table 6.8.1.1: Elementary procedure TMA Set Mode

| Name: TMA Set Mode | | | | |
|---------------------------|-----------------------|------------------|---------------------|-------------|
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x70 | Primary device | 1 | No | n/a |

Table 6.8.1.2: Initiating message parameters and format TMA Set Mode

| Number | Length | Type | Description |
|--------|---------|------------------|-----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | Unsigned integer | Mode state flag |

Table 6.8.1.3: Response message parameters and format for TMA Set Mode

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |

Table 6.8.1.4: Return codes for TMA Set Mode

| OK | FAIL | Comment |
|----|------------------------|---|
| | FormatError | |
| | Busy | |
| | HardwareError | HardwareError shall refer to a detected inability to switch mode. |
| | WorkingSoftwareMissing | |
| | UnsupportedProcedure | UnsupportedProcedure shall be returned if set mode is not supported by the TMA subunit. |
| | OutOfRange | OutOfRange shall be returned if the mode state flag has another value than those listed in the procedure description. |
| | MajorTMAFault | MajorTMAFault shall be returned if the TMA subunit is in bypass mode due to a major TMA fault and TMASetMode to Normal is received and not possible to set. |
| | MinorTMAFault | MinorTMAFault shall be returned if the TMA subunit is in bypass mode due to a minor TMA fault and TMASetMode to Normal is received and not possible to set. |

6.8.2 TMA Get Mode

On receipt of the initiating message, the secondary device shall respond with the mode state flag indicating whether the TMA subunit is in normal mode or in bypass mode. TMA subunits which do not support bypass mode shall return Normal mode.

Mode state flag = 0x00 represents *Normal mode*.

Mode state flag = 0x01 represents *Bypass mode*.

Table 6.8.2.1: Elementary procedure TMA Get Mode

| Name: TMA Get Mode | | | | |
|---------------------------|------------|------------------|---------------------|-------------|
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |

| | | | | |
|-------------|-----------------------|----------|-----------|------------|
| 0x71 | Primary device | 1 | No | n/a |
|-------------|-----------------------|----------|-----------|------------|

Table 6.8.2.2: Initiating message parameters and format for TMA Get Mode

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |

Table 6.8.2.3: Response message parameters and format for TMA Get Mode

| Number | Length | Type | Description |
|--------|---------|------------------|-----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |
| 3 | 1 octet | Unsigned integer | Mode state flag |

Table 6.8.2.4: Return codes TMA Get Mode

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy WorkingSoftwareMissing | |

6.8.3 TMA Get Supported Functions

On receipt of the initiating message, the secondary device shall respond with the function flags and parameters indicating the supported functionality of the addressed TMA subunit.

Table 6.8.3.1: Elementary procedure TMAGetSupportedFunctions

| | | | | |
|--|-----------------------|------------------|---------------------|-------------|
| Name: TMA Get Supported Functions | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x7A | Primary device | 1 | No | n/a |

Table 6.8.3.2: Initiating message parameters and format for TMAGetSupportedFunctions

| Number | Length | Type | Description |
|--------|---------|-------------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |

Table 6.8.3.3: Response message parameters and format for TMAGetSupportedFunctions

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |

| | | | |
|---|---------|-------------------------|---|
| 3 | 1 octet | Unsigned integer | Function flags |
| 4 | 1 octet | Unsigned integer | Min Gain capability (expressed in dB/4) |
| 5 | 1 octet | Unsigned integer | Max Gain capability (expressed in dB/4) |
| 6 | 1 octet | Unsigned integer | Resolution capability (expressed in dB/4) |

Description:

1. A fixed gain TMA subunit shall have min and max gain as the same value.
2. If the resolution is zero, then non linear gain steps are supported (e.g. 3dB and 6dB and 12dB).

NOTE: These parameters represent absolute fixed physical data. Any change of the corresponding parameter in the additional data will not have any operational impact on the TMA.

Table 6.8.3.4: Return codes TMAGetSupportedFunctions

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy WorkingSoftwareMissing | |

Function flags:

| Bit | 7 to 1 | 0 |
|----------|--------|-------------|
| Function | Spare | Bypass Mode |

- Bits are numbered from 0...7, bit number 0 set to 1 represents the value 0x01
- Bit value 0 represents function is not supported
- Bit value 1 represents function is supported
- Spare bits shall be set to 0

6.8.4 TMA Set Gain

The procedure TMASetGain shall only be supported if the TMA subunit gain can be adjusted. On receipt of the initiating message, the secondary device shall first set the addressed TMA subunit to the gain determined by the TMA gain figure parameter, and then return the response message. The TMA gain figure parameter is calculated as 4 times the required gain expressed in dB. (This method of specification allows the gain to be set with a resolution of 0.25 dB while using an integer parameter.)

If the TMA subunit is set in bypass mode by TMASetMode, and TMASetGain is received, then the procedure shall be performed and bypass mode shall be retained.

If any alarm state flag is set than the status has to be validated and the result has to be responded.

Gain shall be accepted if $G_{min} \leq G_{demanded} \leq G_{max}$

For linear steps: $G_{demanded} = (G_{min} + n * G_{resolution})$ where n is a non-negative integer

For non-linear steps: $G_{demanded}$ must be equal to a supported value.

Gmin, Gmax and Gresolution are reported by TMAGetSupportedFunctions. The supported values for non-linear steps are reported by TMAGetSupportedNonLinearGainValues.

For all other values of Gdemanded, the TMA subunit shall respond UnsupportedValue.

Table 6.8.4.1: Elementary procedure TMA Set Gain

| | | | | |
|---------------------------|-----------------------|------------------|---------------------|-------------|
| Name: TMA Set Gain | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x72 | Primary device | 1 | No | n/a |

Table 6.8.4.2: Initiating message parameters and format TMA Set Gain

| Number | Length | Type | Description |
|--------|---------|-------------------------|-----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | Unsigned integer | TMA gain figure |

Table 6.8.4.3: Response message parameters and format for TMA Set Gain

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |

Table 6.8.4.4: Return codes for TMA Set Gain

| OK | FAIL | Comment |
|----|------------------------|--|
| | FormatError | |
| | Busy | |
| | HardwareError | HardwareError shall refer to the detected inability to set the gain according to the instruction, although it is within the TMA gain adjustment range. |
| | WorkingSoftwareMissing | |
| | UnsupportedProcedure | UnsupportedProcedure shall be returned if gain adjustment is not supported by the TMA subunit. |
| | UnsupportedValue | The requested value is not supported. |
| | MajorTMAFault | |
| | MinorTMAFault | |

6.8.5 TMA Get Gain

On receipt of the initiating message, the secondary device shall return the set gain of the TMA subunit. Fixed gain TMA subunits shall return their fixed gain value. The TMA gain figure is calculated as 4 times the set gain expressed in dB.

Table 6.8.5.1: Elementary procedure TMA Get Gain

| | | | | |
|---------------------------|-----------------------|------------------|---------------------|-------------|
| Name: TMA Get Gain | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x73 | Primary device | 1 | No | n/a |

Table 6.8.5.2: Initiating message parameters and format TMA Get Gain

| Number | Length | Type | Description |
|--------|---------|-------------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |

Table 6.8.5.3: Response message parameters and format for TMA Get Gain

| Number | Length | Type | Description |
|--------|---------|------------------|-----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | Return Code | Return code OK |
| 3 | 1 octet | Unsigned integer | TMA gain figure |

Table 6.8.5.4: Return codes for TMA Get Gain

| OK | FAIL | Comment |
|----|------------------------|--|
| | FormatError | |
| | Busy | |
| | WorkingSoftwareMissing | |
| | MajorTMAFault | |
| | MinorTMAFault | |
| | BypassMode | BypassMode shall be returned if the TMA subunit is in bypass mode due to a TMASetMode procedure. |

6.8.6 TMA Set Device Data

On receipt of the initiating message the secondary device shall first write the provided data for the TMA subunit addressed by the subunit number into the fields provided for device data (and listed in Annex C of this document) and then return the response message. If an attempt is made to write to a field which is implemented as read only for the addressed TMA subunit, the return code *ReadOnly* is returned and the data for that field is ignored. If an attempt is

made to write to a field which is not supported for the addressed TMA subunit the return code UnknownParameter is returned and the data for that field is ignored.

Table 6.8.6.1: Elementary procedure TMA Set Device Data

| Name: TMASetDeviceData | | | | |
|-------------------------------|-------------------|-------------------------|----------------------------|--------------------|
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x74 | Primary device | 1 | No | n/a |

Table 6.8.6.2: Initiating message parameters and format for TMA Set Device Data

| Number | Length | Type | Description |
|----------|--------------------|-------------------------|----------------------------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | Unsigned integer | Field number; see annex C |
| 3 | See annex C | See annex C | Data to write |

Table 6.8.6.3: Response message parameters and format for TMA Set Device Data

| Number | Length | Type | Description |
|----------|----------------|-------------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |

Table 6.8.6.4: Return codes for TMA Set Device Data

| OK | FAIL | Comment |
|----|--|---------|
| | FormatError Busy HardwareError WorkingSoftwareMissing ReadOnly UnknownParameter | |

6.8.7 TMA Get Device Data

On receipt of the initiating message the secondary device shall return the data stored for the addressed TMA subunit in the field for additional device data specified by the field number in the initiating message and listed in Annex C of this document.

Table 6.8.7.1: Elementary procedure TMA Get Device Data

| | | | | |
|-------------------------------|-------------------|-------------------------|----------------------------|--------------------|
| Name: TMAGetDeviceData | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x75 | Primary device | 1 | No | n/a |

Table 6.8.7.2: Initiating message parameters and format for TMA Get Device Data

| Number | Length | Type | Description |
|----------|----------------|-------------------------|---|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | Unsigned integer | Field number to be read; see Annex C |

Table 6.8.7.3: Response message parameters and format for TMA Get Device Data

| Number | Length | Type | Description |
|----------|--------------------|-------------------------|-----------------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | Return Code | Return code OK |
| 3 | See Annex C | See Annex C | Field value |

Table 6.8.7.4: Return codes for TMA Get Device Data

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy WorkingSoftwareMissing UnknownParameter | |

6.8.8 TMA Alarm Indication

TMA's use this procedure to report TMA alarm state changes to the primary device. This procedure shall only be performed if the TMA has performed an AlarmSubscribe procedure since its latest reset.

Table 6.8.8.1: Elementary procedure TMA Alarm Indication

| | | | | |
|---------------------------------|-------------------------|-------------------------|----------------------------|--------------------|
| Name: TMAAlarmIndication | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x76 | Secondary device | 2 | No | n/a |

Table 6.8.8.2: Initiating message parameters and format for TMA Alarm Indication

| Number | Length | Type | Description |
|--------|--------|------|-------------|
|--------|--------|------|-------------|

| | | | |
|--------|---------|------------------|----------------------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2* i | 1 octet | Unsigned integer | Return code i; see Annex B |
| 2*i +1 | 1 octet | Unsigned integer | State flag i |

$i = 1 \dots N$

6.8.8.1 Further requirements

For each alarm, the current alarm state and alarm code shall be reported if and only if any change in its state has occurred since the last reported state.

A TMA Alarm Indication procedure shall be performed if at least one TMA alarm shall be reported for the TMA subunit. The first TMA Alarm Indication procedure after the Alarm Subscribe procedure shall report the active alarms.

Alarm state changes are considered as reported at the time the message is passed to the transport layer.

State flag = 0x00 represents alarm state *cleared*.

State flag = 0x01 represents alarm state *raised*.

6.8.9 TMA Clear Active Alarms

On receipt of the initiating message the secondary device shall first clear all stored alarm information for the addressed TMA subunit and then return a procedure response message. In the event that the cause of the alarm persists the alarm shall be re-raised and a new TMA Alarm Indication procedure shall be performed.

Table 6.8.9.1: Elementary procedure TMA Clear Active Alarms

| | | | | |
|-----------------------------------|-----------------------|------------------|---------------------|-------------|
| Name: TMAClearActiveAlarms | | | | |
| Code: | Issued by | Procedure class: | DownloadMode state: | Power mode: |
| 0x77 | Primary device | 1 | No | n/a |

Table 6.8.9.2: Initiating message parameters and format for TMA Clear Active Alarms

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |

Table 6.8.9.3: Response message parameters and format for TMA Clear Active Alarms

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |

Table 6.8.9.4: Return codes for TMA Clear Active Alarms

| OK | FAIL | Comment |
|----|-------------|---------|
| | FormatError | |

| | | |
|--|--------------------------------|--|
| | Busy WorkingSoftwareMissing | |
|--|--------------------------------|--|

6.8.10 TMA Get Alarm Status

On receipt of the initiating message the secondary device shall report the alarm codes of the active alarms for the addressed TMA subunit.

Table 6.8.10.1: Elementary procedure TMA Get Alarm Status

| | | | | |
|--------------------------------|-----------------------|------------------|---------------------|-------------|
| Name: TMAGetAlarmStatus | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x78 | Primary device | 1 | No | n/a |

Table 6.8.10.2: Initiating message parameters and format for TMA Get Alarm Status

| Number | Length | Type | Description |
|--------|---------|------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |

Table 6.8.10.3: Response message parameters and format for TMA Get Alarm Status

| Number | Length | Type | Description |
|--------|---------|------------------|-------------------------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |
| 2 + i | 1 octet | AlarmCode | Alarm code for alarm number i |

i = 1 ... N

Table 6.8.10.4: Return codes for TMA Get Alarm Status

| OK | FAIL | Comment |
|---|---|---------|
| All return codes marked as used for alarms in Annex B | FormatError Busy WorkingSoftwareMissing | |

6.8.11 TMA Get Number of Subunits

On receipt of the initiating message the secondary device shall return the number of subunits it controls.

Table 6.8.11.1: Elementary procedure TMAGetNumberOfSubunits

| | | | | |
|-------------------------------------|------------|------------------|---------------------|-------------|
| Name: TMAGetNumberOfSubunits | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |

| | | | | |
|-------------|-----------------------|----------|-----------|------------|
| 0x79 | Primary device | 1 | No | n/a |
|-------------|-----------------------|----------|-----------|------------|

Table 6.8.11.2: Initiating message parameters and format for TMAGetNumberOfSubunits

| Number | Length | Type | Description |
|--------|----------|------|-----------------|
| None | 0 octets | None | No data carried |

Table 6.8.11.3: Response message parameters and format for TMAGetNumberOfSubunits

| Number | Length | Type | Description |
|--------|---------|------------------|--------------------|
| 1 | 1 octet | ReturnCode | Return code OK |
| 2 | 1 octet | Unsigned integer | Number of subunits |

Table 6.8.11.4: Return codes for TMAGetNumberOfSubunits

| OK | FAIL | Comment |
|----|---|---------|
| | FormatError Busy WorkingSoftwareMissing | |

6.8.12 3GPP Clear Active Alarms and Get Alarm Status

When an TMA receives ClearActiveAlarms, it shall perform the procedure for all subunits of the TMA and then return a single procedure response message.

When an TMA receives GetAlarmStatus, it shall in a single procedure response message report the union of the alarm codes of all active alarms for all subunits, i.e. the same alarm codes shall be reported only once.

6.8.13 TMA Get Supported Non-Linear Gain Values

On receipt of the initiating message, the secondary device shall respond with a message containing a list of supported values in numerical order, preceded by the number (N) of such values contained in the list.

Table 6.8.13.1: Elementary procedure TMAGetSupportedNonLinearGainValues

| | | | | |
|---|-----------------------|------------------|---------------------|-------------|
| Name: TMAGetSupportedNonLinearGainValues | | | | |
| Code: | Issued by: | Procedure class: | DownloadMode state: | Power mode: |
| 0x7B | Primary device | 1 | No | n/a |

Table 6.8.13.2: Initiating message parameters and format for TMAGetSupportedNonLinearGainValues

| Number | Length | Type | Description |
|--------|--------|------|-------------|
|--------|--------|------|-------------|

| | | | |
|---|---------|-------------------------|----------------|
| 1 | 1 octet | Unsigned integer | Subunit number |
|---|---------|-------------------------|----------------|

Table 6.8.13.3: Response message parameters and format for TMAGetSupportedNonLinearGainValues

| Number | Length | Type | Description |
|--------|---------|-------------------------|--|
| 1 | 1 octet | Unsigned integer | Subunit number |
| 2 | 1 octet | ReturnCode | Return code OK |
| 3 | 1 octet | Unsigned integer | Number of non linear gain values supported (N) |
| 3 + i | 1 octet | Unsigned integer | Non linear gain supported value number i (expressed in dB/4) |

i=1 .. N

Table 6.8.13.4: Return codes TMAGetSupportedNonLinearGainValues

| OK | FAIL | Comment |
|----|------------------------|--|
| | FormatError | |
| | Busy | |
| | WorkingSoftwareMissing | |
| | UnsupportedProcedure | UnsupportedProcedure shall be returned if the TMA does not support non-linear gain steps |

7 Unknown elementary procedures

Void.

Annex A (normative): Return codes for secondary devices

Table A.1: Return Codes for Secondary Devices

| Code | Name | Comment | Alarm | DownloadMode state |
|------|--------------------------|---|-------|--------------------|
| 0x00 | OK | Normal response | | X |
| 0x02 | Motor Jam | Motor cannot move | X | |
| 0x03 | ActuatorJam | Actuator jam has been detected. No movement of the actuator, but movement of the motor was detected | X | |
| 0x05 | Busy | The device is busy and cannot execute the procedure until an ongoing activity is completed | | |
| 0x06 | ChecksumError | Checksum incorrect for otherwise valid data.. | | |
| 0x0B | FAIL | Abnormal response. Indicates that a procedure has not been executed successfully | | X |
| 0x0E | NotCalibrated | The device has not completed a calibration operation, or calibration has been lost | X | |
| 0x0F | NotConfigured | Actuator configuration data is missing | X | |
| 0x11 | HardwareError | Any hardware error which cannot be classified. May not be reported as an alarm until the fault is likely to be persistent | X | X |
| 0x13 | OutOfRange | A parameter given by an operator (e.g. tilt value or memory offset) is out of range | | |
| 0x19 | UnknownProcedure | Received procedure code is not defined | | X |
| 0x1D | ReadOnly | Invalid device data parameter usage | | X |
| 0x1E | UnknownParameter | Specified parameter is not supported for the used procedure | | X |
| 0x21 | WorkingSoftwareMissing | The unit is inDownloadMode state. Returned upon unsupported procedure when in DownloadMode state | | X |
| 0x22 | InvalidFileContent | The data being downloaded is detected to be of wrong format or size | | X |
| 0x24 | FormatError | Procedure message is inconsistent or if an addressed field or antenna is invalid or the data parameter field length is inconsistent with the corresponding field length parameter | | X |
| 0x25 | UnsupportedProcedure | The procedure is optional and not supported or the procedure does not apply to this device type | | |
| 0x26 | InvalidProcedureSequence | Procedure sequence as described in annex C is expected but not experienced by the secondary device | | |
| 0x27 | ActuatorInterference | An actuator movement outside the control of the RET unit has been detected. Probable cause is manual interference | X | |
| 0x1A | MinorTMAFault | A fault in the TMA subunit is detected which reduces the gain performance but maintains its function. | X | |
| 0x1B | MajorTMAFault | A fault in the TMA subunit is detected. The fault prevents the function of the TMA subunit. | X | |
| 0x1C | UnsupportedValue | The requested value is not supported. | | |
| 0x1F | BypassMode | The TMA subunit is in bypass mode and cannot report a correct gain value. | X | |

NOTE: A TMA subunit with a fault which is causing it to switch to bypass mode will report both the fault and the BypassMode alarm.

Annex B (normative): Assigned fields for additional data

The following standard fields have no operational impact and are used by the procedures SetDeviceData, GetDeviceData, AntennaSetDeviceData and AntennaGetDeviceData. Little endian order is used for storage of multiple-octet numbers. Where ASCII variables are shorter than the assigned field lengths the characters are right aligned and leading blanks are filled with null characters (0x00). Unused or not initialized parameter shall return the value 0x00 for the GetDeviceData, AntennaGetDeviceData and TMAGetDeviceData procedure.

Table B.1: Assigned fields for additional data of RETAP

| Field No. | Length (octets) | Format | Description |
|-----------|-----------------|---------------------|--|
| 0x01 | 15 | ASCII | Antenna model number |
| 0x02 | 17 | ASCII | Antenna serial number |
| 0x03 | 2 | 16-bit unsigned | Antenna operating band(s): see below |
| 0x04 | 8 | 4 x 16-bit unsigned | Beamwidth for each operating band in band order (deg), beginning with lowest band. The lowest band is transmitted within the first 16-bit value. (example: width for band I, width for band III) |
| 0x05 | 4 | 4 x 8-bit unsigned | Gain [dBi] for each operating band in band order, expressed in gain value times 10, beginning with the lowest band. The lowest band is transmitted within the first 8-bit value (example: gain for band I, gain for band III) |
| 0x06 | 2 | 16-bit signed | Maximum supported electrical tilt [degree], expressed in tilt value times 10, format as in subclause 3.1 |
| 0x07 | 2 | 16-bit signed | Minimum supported electrical tilt [degree], expressed in tilt value times 10, format as in subclause 3.1 |
| 0x21 | 6 | ASCII | Installation date |
| 0x22 | 5 | ASCII | Installer's ID |
| 0x23 | 32 | ASCII | Base station ID |
| 0x24 | 32 | ASCII | Sector ID |
| 0x25 | 2 | 16-bit unsigned | Antenna bearing [degree], in the range of 0 – 359,9 degree, expressed as bearing value times 10 |
| 0x26 | 2 | 16-bit signed | Installed mechanical tilt [degree], expressed in tilt value times 10, format as in subclause 3.1 |

Table B.2: Coding for operating bands in field 0x03

| Bit no | 15...11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|---------|----|---|----|------|-----|---|----|-----|----|---|----|
| Operating band | Spare | XI | X | IX | VIII | VII | I | II | III | IV | V | VI |

The operating bands are defined in subclause 4.3.7 in [4].

Bits are numbered from 0 to 15, bit no 0 set=1 represents the value 0x0001.

Bit set=1 represents operating band is supported.

Bit set=0 represents operating band is not supported.

Spare bits shall be set=0.

Unused Beamwidth and Gain octets shall be set to 0x0000.

Examples of operating bands: 0000 0000 0001 0000 = Operating band II

 0000 0000 0011 1000 = Operating band I, II and III

Table B.3: Assigned fields for additional data of TMAAP

| Field No. | Length (octets) | Format | Description |
|-----------|-----------------|---------------------------|---|
| 0x01 | 15 | ASCII | Antenna model number |
| 0x02 | 17 | ASCII | Antenna serial number |
| 0x03 | 2 | 16-bit unsigned | Antenna operating band(s): see below |
| 0x04 | 8 | 4 x 16-bit unsigned | Beamwidth for each operating band in band order (deg) beginning with lowest band. The lowest band is transmitted within the first 16-bit value. |
| 0x05 | 4 | 4 x 8-bit unsigned | Gain [dBi] for each operating band in band order, expressed in gain value times 10, beginning with the lowest band. The lowest band is transmitted within the first 8-bit value (example: gain for band I, gain for band III) |
| 0x21 | 6 | ASCII | Installation date |
| 0x22 | 5 | ASCII | Installer's ID |
| 0x23 | 32 | ASCII | Base station ID |
| 0x24 | 32 | ASCII | Sector ID |
| 0x25 | 2 | 16-bit unsigned | Antenna bearing |
| 0x26 | 2 | 16-bit signed | Installed mechanical tilt (degrees * 10) |
| 0x13 | 1 | 8-bit unsigned | TMA subunit type (see table B.4) |
| 0x14 | 4 | 2x16-bit unsigned integer | TMA subunit receive frequency band (see Table B.5) |
| 0x15 | 4 | 2x16-bit unsigned integer | TMA subunit transmit frequency band (see Table B.5) |
| 0x16 | 1 | Unsigned integer | Maximum supported gain given as gain figure expressed in dB/4 |
| 0x17 | 1 | Unsigned integer | Minimum supported gain given as gain figure expressed in dB/4 |
| 0x18 | 1 | Unsigned integer | Gain resolution given as the gain resolution figure expressed in dB/4 |

Table B.4: Field 0x13 interpretation

| Bit number | 7... 2 | 1 | 0 |
|------------------|--------|-----------------------|--------|
| TMA subunit type | Spare | Reserved for AISG use | Bypass |

Bits are numbered from 0...7, bit numbered 0 set to 1 represents the value 0x01.

Bit value 0 represents TMA subunit type is not supported.

Bit value 1 represents TMA subunit type is supported.

Spare bits and Bit 1 shall be set to 0.

Example: 0000 0001 = Bypass

Table B.5: Field 0x14 and 0x15 interpretation

| Octet number | Description |
|---------------------|--------------------|
| 0 | fmin low octet |
| 1 | fmin high octet |
| 2 | fmax low octet |
| 3 | fmax high octet |

fmin and fmax are expressed in 100kHz steps from 0kHz.

Example: For 850MHz, low octet = 0x34, high octet = 0x21.

Annex C (normative): Procedure sequence for download of software to a secondary device

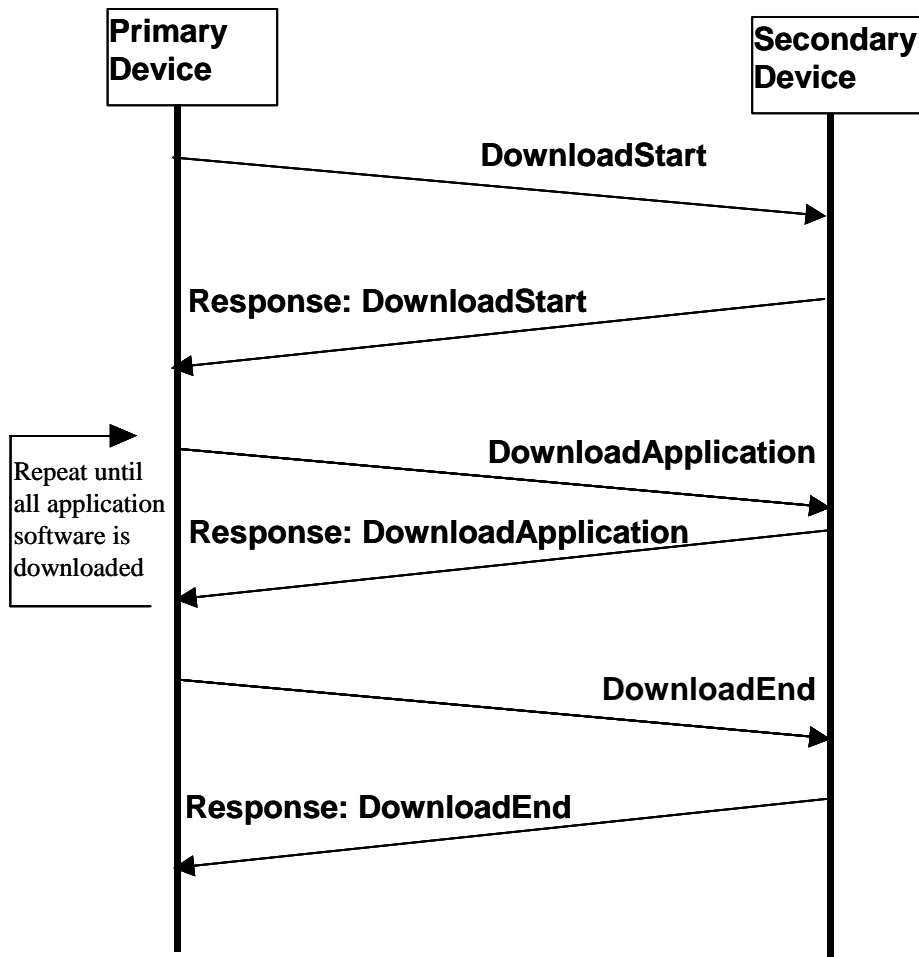


Figure C.1: Procedure sequence for Software Download

The erasure of the secondary device application software shall not be done before the reception of the Download Application message. The data content of the Download Application message is implementation specific but it is recommended to support an application software validity feature that shall minimise the risk of downloading faulty or invalid application software.

Annex D (informative): Overview of elementary procedures

Table D.1: Elementary Procedures and Procedure Codes

| Elementary Procedure | Procedure Code | Issued by | DownloadMode state |
|---|----------------|------------------|--------------------|
| Common Procedure Set | | | |
| (Reserved) | 0x01 | | |
| Reset Software | 0x03 | primary device | Yes |
| Get Alarm Status | 0x04 | primary device | No |
| Get Information | 0x05 | primary device | Yes |
| Clear Active Alarms | 0x06 | primary device | No |
| Read User Data | 0x10 | primary device | No |
| Write User Data | 0x11 | primary device | No |
| Alarm Subscribe | 0x12 | primary device | No |
| Self Test | 0x0A | primary device | No |
| Download Start | 0x40 | primary device | Yes |
| Download Application | 0x41 | primary device | Yes |
| Download End | 0x42 | primary device | Yes |
| Vendor Specific Procedure | 0x90 | primary device | Vendor specific |
| RETAP Single-Antenna Procedure Set | | | |
| Set Device Data | 0x0E | primary device | No |
| Get Device Data | 0x0F | primary device | No |
| Calibrate | 0x31 | primary device | No |
| Send Configuration Data | 0x32 | primary device | No |
| Set Tilt | 0x33 | primary device | No |
| Get Tilt | 0x34 | primary device | No |
| Alarm Indication | 0x07 | secondary device | No |
| RETAP Multi-Antenna Procedure Set | | | |
| Antenna Calibrate | 0x80 | primary device | No |
| Antenna Send Configuration Data | 0x89 | primary device | No |
| Antenna Set Tilt | 0x81 | primary device | No |
| Antenna Get Tilt | 0x82 | primary device | No |
| Antenna Set Device Data | 0x83 | primary device | No |
| Antenna Get Device Data | 0x84 | primary device | No |
| Antenna Alarm Indication | 0x85 | secondary device | No |
| Antenna Clear Active Alarms | 0x86 | primary device | No |
| Antenna Get Alarm Status | 0x87 | primary device | No |
| Antenna Get Number of Antennas | 0x88 | primary device | No |
| TMAAP Procedure Set for multi and single TMA | | | |
| TMASetMode | 0x70 | primary device | No |
| TMAGetMode | 0x71 | primary device | No |
| TMAGetSupportedFunctions | 0x7A | primary device | No |
| TMASetGain | 0x72 | primary device | No |
| TMAGetGain | 0x73 | primary device | No |
| TMASetDeviceData | 0x74 | primary device | No |
| TMAGetDeviceData | 0x75 | primary device | No |
| TMAAlarmIndication | 0x76 | secondary device | No |
| TMAClearActiveAlarms | 0x77 | primary device | No |
| TMAGetAlarmStatus | 0x78 | primary device | No |
| TMAGetNumberOfSubunits | 0x79 | primary device | No |
| TMAGetSupportedNonLinearGainValues | 0x7B | primary device | No |

NOTE: The notion "yes" in the DownloadMode state column indicates that the listed procedures are mandatory if the DownloadMode state can be entered by the secondary device.

Annex E: I-frame and INFO-field format (Informative)

The I-frame and INFO-field formats for both primary and secondary stations are illustrated. To transfer elementary procedures, the INFO-field of the I-frame is used.

Table E.1: HDLC-Frame:

| Flag 1 octet | ADR 1 octet | Control 1 octet | INFO N octets | CRC 2 octets | | Flag 1 octet |
|-----------------|----------------|--------------------|---|-----------------|-----------------|-----------------|
| 0x7E | Device Address | Control bits | Variable length (must support a maximum length of at least 74 octets) | CRC1 Low Octet | CRC2 high octet | 0x7E |

| Procedure ID | Number of data octets | | Data octets |
|--------------|-----------------------|------------|---|
| 1 octet | low octet | high octet | Variable length (must support a maximum length of at least 71 octets) |

Format of the I-Frame and INFO Field

Devices shall support the following data length:

Mandatory: $0 \leq \text{data octets} \leq 71$

Optional: $0 \leq \text{data octets} < 65,536$ octets

Annex F (informative): Change history

| Change history | | | | | | | |
|----------------|-------|-----------|----|-----|--|-------|-------|
| Date | TSG # | TSG Doc. | CR | Rev | Subject/Comment | Old | New |
| 03/2007 | 35 | RP-070059 | | | Submitted to TSG-RAN #35 for information and approval | - | 1.2.0 |
| 03/2007 | 35 | RP-070059 | | | approved at TSG-RAN#35 and placed under change control | 1.2.0 | 7.0.0 |
| 06/2007 | 36 | RP-070325 | 1 | | Adaptation of TMA Set Mode return code description | 7.0.0 | 7.1.0 |
| 06/2007 | 36 | RP-070325 | 2 | 2 | Clarification of TMA Get Mode description | 7.0.0 | 7.1.0 |
| 06/2007 | 36 | RP-070325 | 4 | | Missing return code for TMA Set Mode | 7.0.0 | 7.1.0 |
| 06/2007 | 36 | RP-070325 | 5 | 1 | Return code description for TMA Set Gain | 7.0.0 | 7.1.0 |
| 06/2007 | 36 | RP-070325 | 8 | 1 | Adaptation of procedure message interpretation | 7.0.0 | 7.1.0 |
| 06/2007 | 36 | RP-070321 | 9 | 2 | GetDeviceData new cause value | 7.0.0 | 7.1.0 |
| 06/2007 | 36 | RP-070325 | 10 | | Correction of return code table description in annex A | 7.0.0 | 7.1.0 |
| 09/2007 | 37 | RP-070570 | 11 | 1 | Correction of procedure TMAGetSupportedNonLinearGainValues | 7.1.0 | 7.2.0 |
| 09/2007 | 37 | RP-070570 | 12 | 1 | Correction of return code UnsupportedValue | 7.1.0 | 7.2.0 |
| 09/2007 | 37 | RP-070576 | 13 | | Introduction of UMTS1500 requirement | 7.2.0 | 8.0.0 |
| 12/2007 | 38 | RP-070838 | 15 | | Missing Parameter for "TMA Get Supported Non-Linear Gain Values" | 8.0.0 | 8.1.0 |

History

| Document history | | |
|-------------------------|--------------|-------------|
| V8.0.0 | January 2008 | Publication |
| V8.1.0 | January 2008 | Publication |
| | | |
| | | |
| | | |