

API for Sending and receiving short messages via AS55X

Version 1.0 (2011-01-25)

0. Introduction

Sending and receiving short messages are controlled by packets with readable Text via a Telnet connection.

Three packet types are defined:

- Request, from application to AS55X
- Response, from AS55X to application
- Indication, from AS55X to application

Every packet consists of a headline and some lines with information elements. Every line begins with a reserved term as line identifier and ends with carriage return and line feed. The whole packet is terminated by an empty line.

The headline begins with the fixed string 'AS55XMessageExchangeV1.0' as line identifier, followed by a blank and the packet identifier. The packet identifier can be either a command type, a indication type or the string 'Response'. Defined packet identifiers are:

- SendMessage
- RequestStatus
- GetMessage
- ReceivedMessageAck
- SetMessageIndication
- ReceivedMessage
- Response

Information elements with an argument get a colon after the identifier, immediately followed by the argument. Some information elements are mandatory, some are optional.

Reserved terms are:

- all line, packet and information element identifiers
- carriage return (<CR>, 0x0d)
- line feed (<LF>, 0x0a)

Reserved terms are case-sensitive.

1. Requests

1.1. Request type: SendMessage

Description: Send a short message

Information elements:

RequestId:<str>

mandatory

<str>: Printable ASCII string of 1 to 16 characters

Channel:<str>

optional, applicable only on Telnet channel 1

<str>: Number of the GSM channel. If omitted, the GSM channel number is set to the Telnet channel number.

To:<str>

mandatory

<str>: Destination number of the message in international order, beginning with '+'.
</str>

ServiceCenter:<str>

optional

<str>: Service center number of the message in international order, beginning with '+'. If omitted, the read service center from the SIM will be used.

Message:<str>

mandatory

<str>: 1 to 160 characters message content. A <CR> is not allowed in the message, but a <LF> may be included.

Possible provisional response:

- Accepted

Possible final responses:

- SyntaxError
- ChannelBusy
- ChannelNotAvailable
- ServiceCenterUnknown
- Unsuccessful
- Successful

1.2. Request type: RequestStatus

Description: Check if short message exchange is possible

Information elements:

RequestId:<str>

mandatory

<str>: Printable ASCII string of 1 to 16 characters

Channel:<str>

optional, applicable only on Telnet channel 1

<str>: Number of the GSM channel. If omitted, the GSM channel number is set to the Telnet channel number.

Possible final responses:

- SyntaxError
- ChannelNotAvailable
- Ready, if the network provider name is present, this will be a description of the response
- ChannelBusy

1.3. Request type: GetMessage

Description: Read a short message, if available

Information elements:

RequestId: <str>

mandatory

<str>: Printable ASCII string of 1 to 16 characters

Channel: <str>

optional, applicable only on Telnet channel 1

<str>: Number of the GSM channel. If omitted, the GSM channel number is set to the Telnet channel number.

AwaitAck

optional, if omitted, the message will be considered as acknowledged with successful sending to the application. If present, the message must be acknowledged explicitly by the request ReceivedMessageAck

Possible provisional response:

- Accepted

Possible final responses:

- SyntaxError

- ReceivedMessage

- NoMessageAvailable

- ChannelNotAvailable

1.4. Request type: ReceivedMessageAck

Description: Acknowledge a received short message

Information elements:

RequestId: <str>

mandatory

<str> must be the ReceivedMessageId of the ReceivedMessageIndication or the RequestId of the GetMessage request.

Possible provisional response:

- Accepted

Possible final responses:

- SyntaxError

- Successful

- NoMessageAvailable

- ChannelNotAvailable

1.5. Request type: SetMessageIndication:

Description: After reception of this request, the AS55X reports all received messages unsolicited to the application. This setting is sticky as long as the TCP connection is open. It will also be cleared with reception of a valid or invalid GetMessage command.

Information elements:

MessageId:<str>

mandatory

<str>: Printable ASCII string of 1 to 16 characters

Channel:<str>

optional, applicable only on Telnet channel 1

<str>: Number of the GSM channel or 'All' for all channels. If omitted, the GSM channel number is set to the Telnet channel number.

AwaitAck

optional, if omitted, received messages will be considered as acknowledged with successful sending to the application. If present, the messages must be acknowledged explicitly by the request ReceivedMessageAck.

Possible final responses:

- SyntaxError
- Successful

2. Responses

Description: Positive or negative, provisional or final response to a former request.

Information elements:

RequestId:<str>

mandatory

<str>: Printable ASCII string of 1 to 16 characters

Cause:<str>

mandatory

<str>: One of these strings

- Accepted
- SyntaxError
- ChannelBusy
- ChannelNotAvailable
- ServiceCenterUnknown
- Unsuccessful
- Successful
- Ready
- ReceivedMessage
- NoMessageAvailable

Description:<str>

optional

<str>: Additional unfomatted information

From:<str>

mandatory if Cause: ReceivedMessage

<str>: Destination number of the message in international order, beginning with '+'.
<str>: 1 to 160 characters message content.

Message:<str>

mandatory if Cause: ReceivedMessage

<str>: 1 to 160 characters message content.

3. Indications

3.1. Indication type: ReceivedMessageIndication

Description: Report a received message to the application.

Information elements:

ReceivedMessageId:<str>

mandatory

<str>: Printable ASCII string of 1 to 16 characters

From:<str>

mandatory

<str>: Destination number of the message in international order, beginning with '+'.
</p></div>

Message:<str>

mandatory

<str>: 1 to 160 characters message content.

AckRequired

optional, if present, this indication has to be acknowledged with the request ReceivedMessageAck

4. Timing considerations

Requests will be responded promptly by the AS55X, either with a provisional or a final response. In case of a provisional response, the delay of the final response can take up to 30 seconds.

If a ReceivedMessageIndication with AckRequired is not acknowledged within five seconds, the ReceivedMessageIndication will be repeated until either an acknowledge has been received, a GetMessage request has been received or the TCP connection has been closed.

5. Configuration

There are two operation modes of this short message server, single channel mode and multiple channels mode.

In the single channel mode, up to six short message server channels are directly linked to the corresponding gsm channels. Short message exchange can be done in parallel to voice access and the reaction time is faster than in the multiple channels mode.

By use of the multiple channels mode, all available gsm channels (up to 30) of the AS55X can be controlled by one short message server channel. The Channel: information element is used to address a specific gsm channel. In this mode, one interface to the gsm channel is shared by the short message server and the voice application. Hence, if a voice connection is running, no short message access can be made at the same time and vice versa. If a short message access is tried while a voice connection is active or in progress, the request will be responded with ChannelBusy and the command should be repeated later on.

In the multiple channels mode, incoming message on all gsm channels can be indicated via this server channel.

Both modes can be mixed. If multiple channels mode is configured on Telnet channel 1 and single channel mode on another one, no access can be made from the multiple message controller to this specific GSM channel.

6. Examples:

Example 1. Send a message to the German number 01711234567 via the GSM channel 3 with T-Mobile SIM.

Request to AS55X:
AS55XMessageExchangeV1.0 SendMessage<CR><LF>
RequestId:jef455A<CR><LF>
Channel:3<CR><LF>
To:+491711234567<CR><LF>
ServiceCenter:+491710760000<CR><LF>
Message:To be or not to be<CR><LF>
<CR><LF>

Provisional response from AS55X:
AS55XMessageExchangeV1.0 Response<CR><LF>
RequestId:jef455A<CR><LF>
Cause:Accepted<CR><LF>
<CR><LF>

Final response from AS55X:
AS55XMessageExchangeV1.0 Response<CR><LF>
RequestId:jef455A<CR><LF>
Cause:Successful<CR><LF>
<CR><LF>

Example 2. Send request with syntax error

Request to AS55X:
AS55XMessageExchangeV1.0 SendMessage<CR><LF>
RequestId:jef455A<CR><LF>
Channel:3<CR><LF>
to:+491711234567<CR><LF>
ServiceCenter:+491710760000<CR><LF>
Message:To be or not to be<CR><LF>
<CR><LF>

Final response from AS55X:
AS55XMessageExchangeV1.0 Response<CR><LF>
RequestId:jef455A<CR><LF>
Cause:SyntaxError<CR><LF>
CauseDescription:ToMissingOrInvalid<CR><LF>
<CR><LF>

Example 3. Poll for received message without acknowledge

Request to AS55X:
AS55XMessageExchangeV1.0 GetMessage<CR><LF>
RequestId:55554444<CR><LF>
<CR><LF>

Final response from AS55X:
AS55XMessageExchangeV1.0 Response<CR><LF>
RequestId:55554444<CR><LF>
Cause:ReceivedMessage<CR><LF>
From:+491717654321<CR><LF>
Message:That is the question<CR><LF>
<CR><LF>

Example 4. Set received message indication, receive and acknowledge a message.

Request to AS55X:
AS55XMessageExchangeV1.0 SetMessageIndication<CR><LF>
RequestId:2955455<CR><LF>
Channel:2<CR><LF>
AwaitAck<CR><LF>
<CR><LF>

Final response from AS55X:
AS55XMessageExchangeV1.0 Response<CR><LF>
RequestId:2955455<CR><LF>
Cause:Successful<CR><LF>
<CR><LF>

Indication from AS55X:
AS55XMessageExchangeV1.0 ReceivedMessageIndication<CR><LF>
ReceivedMessageId:52866<CR><LF>
From:+491717654321<CR><LF>
Message:That is the question<CR><LF>
AckRequired<CR><LF>
<CR><LF>

Request to AS55X:
AS55XMessageExchangeV1.0 ReceivedMessageAck<CR><LF>
RequestId:52866<CR><LF>
<CR><LF>

Final response from AS55X:
AS55XMessageExchangeV1.0 Response<CR><LF>
RequestId:sidurhiu<CR><LF>
Cause:Successful<CR><LF>
<CR><LF>