

h/p/cosmos  
**coscom<sup>®</sup> v4**  
extended remote service

## Table of contents

<b>[1] Overview .....</b>	<b>3</b>
<b>[2] Feature Matrix .....</b>	<b>4</b>
<b>[3] Actions .....</b>	<b>6</b>
[3A] SetEventMask .....	6
[3B] ShowUserMessage .....	6
[3C] ShowUserImage .....	9
[3D] StartProfile .....	12
[3E] RequestPermissions .....	13
[3F] SetPersonData .....	15
<b>[4] Variables .....</b>	<b>15</b>
[4A] RFID .....	15
[4B] TerminalLanguage .....	16
[4C] Permissions .....	16
[4D] RequestedProfileResult .....	16
Figure 1: Message in menu .....	7
Figure 2: Message during workout .....	8
Figure 3: Image in menu .....	9
Figure 4: Image during workout .....	10
Figure 5: Start ext. profile .....	12
Figure 6: Request send allowance .....	14

[1] **Overview**

The new extended remote service is an extension to the basic remote control service.

It is designed to connect h/p/cosmos coscom v4 devices to external training servers which allow patient/subject and training plan management.

The external server should be able to see if a patient/subject is trying to log in via RFID and it should be possible to send a profile and messages to the device.

As there is a chance of possible misuse sending messages and images to the device **there will be an IP address validation. The external control has to use action “SetServerName” before sending any messages or images to the device terminal. When this action is called the IP address is validated and a user request is shown on the device terminal if the IP address is unknown or untrusted. The user has to accept the request in order to see external messages or images. See variable “IsAllowedToSend” for current state.** This must be done once for every IP address and will be saved for further request. I can be revoked on the terminal any time.

av

[2] **Feature Matrix**

h/p/cosmos coscom v4 is designed to meet different device types. But not all device types support all defined actions and variables. Following matrix will address this issue:

Action/Device type	Treadmill	Ergometer	Ladder	Cross trainer	Stepper
SetEventMask	x	x	x	x	x
ShowUserMessage	x	x	x	x	x
ShowUserImage	x	x	x	x	x
StartProfile	x	x	x	x	x
RequestPermissions	x	x	x	x	x
SetPersonData	x	x	x	x	x

*\* Using this action without external control permission will result in an error response (see chapter "Safety considerations").*

Variable/Device type	Treadmill	Ergometer	Ladder	Cross trainer	Stepper
RFID (Index: 0)	X	X	X	X	X
TerminalLanguage (Index: 1)	X	X	X	X	X
Permissions (Index: 2)	X	X	X	X	X
RequestedProfileResult (Index: 3)	X	X	X	X	X

### [3] Actions

Following you will find a short description with syntax sample of every coscom v4 action of extended remote service.

**Note:** If no other description is given, values are defined as floating point values expected with max. six decimal places and maximal value +/- 1.000.000. Integer values are defined as signed 32-bit values. String values max.

#### [3A] SetEventMask

(Action index: 0) As described in chapter “eventing” in the document “coscom v4 – basic remote service” with this method it is possible to enable or disable automatic value change transmission of the device. After switching on the device all events for this service are disabled. Changing the event mask will lead to new initial events and new event index in the event messages (see chapter “eventing”).

#### Parameters

- *EventMask* (direction in – index 0): A string consisting of 0 and 1 indicating if eventing for specific variables should be turned on. The position in the string determines the variable index. For example, the string 100101 enables eventing for variables with indices 0, 2 and 5. Leading 0 can be left out. The string starts with the highest variable index and ends with index 0. For variable indices see chapter variables. For variable indices which are not supported in the current device type changes to event mask will have no effect.

Syntax sample request: \*A0s1\*I0:1001\*Y0:DE\*Z

Syntax sample response: \*A0s1\*Y0:3F\*Z

#### [3B] ShowUserMessage

(Action index: 1) Sends a message to the device that should be shown to the user. This could e. g. be a welcome message after login or a message at the end which device the user should go to next.

**Note:** Before a message can be sent the IP address of the external connection must be accepted as image or text source on the device terminal. See action “SetServerName” and variable “IsAllowedToSend”.

Depending on current profile state the message will be shown different:

If no profile is running the message will be shown as a popup over mostly the whole screen regardless of where the user is in the menu:

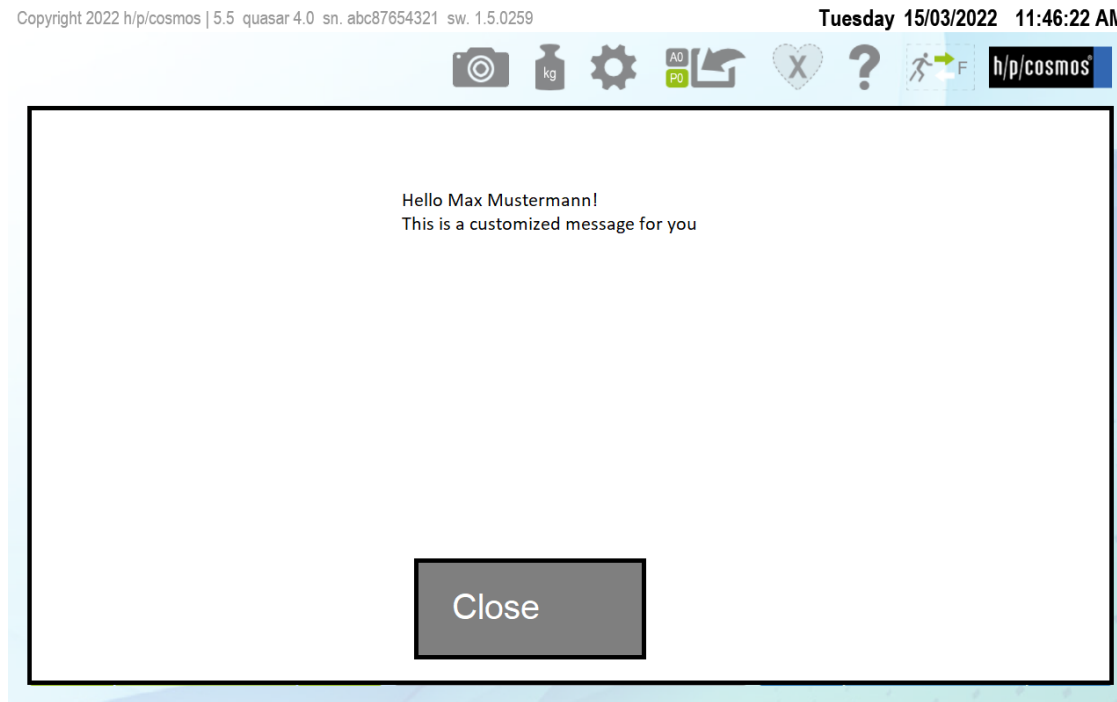


Figure 1: Message in menu

If a profile is running the message will be shown over the workout graph:

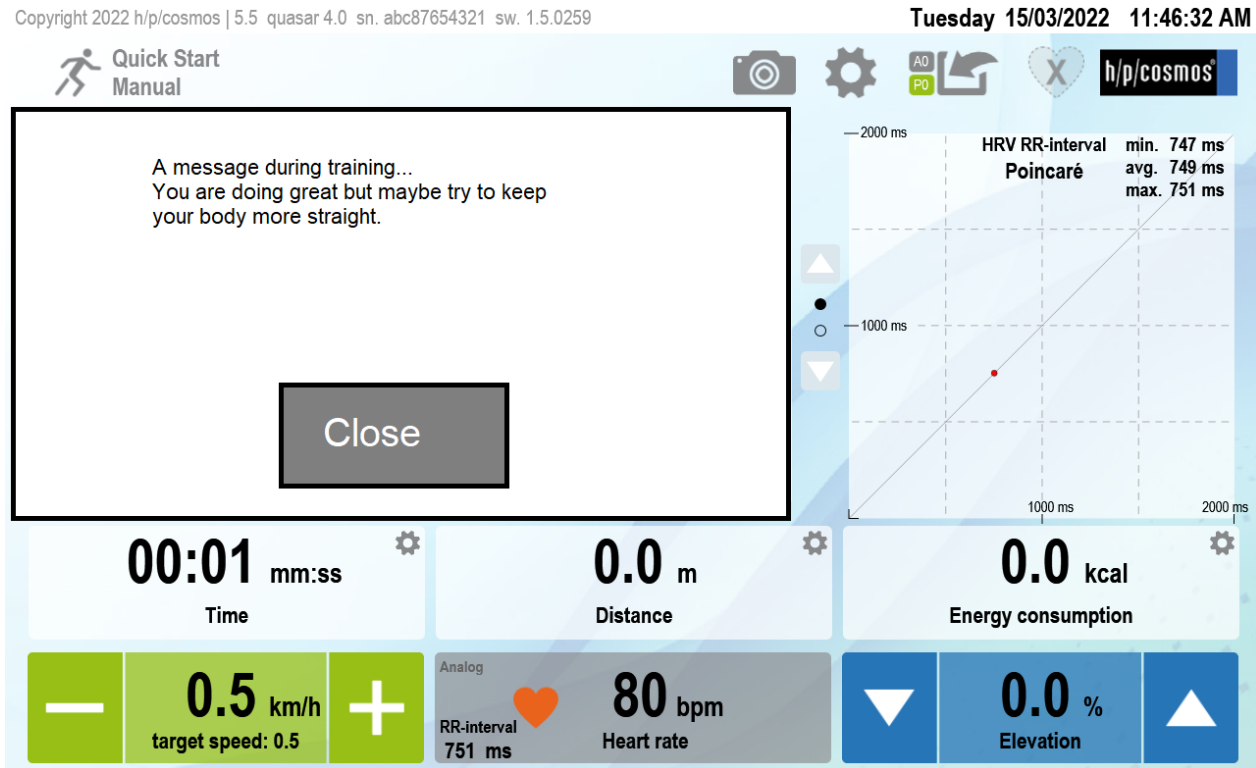


Figure 2: Message during workout

Parameters:

- *message* (direction in - index 0, max. length: x characters):  
The message which should be shown to the user
- *displayTime* (direction in - index 1, integer): The Specifies the length of time in seconds the message should be displayed. Possible value range: 5-120 seconds. special value: 0 = message must be acknowledged and remains permanently visible.

Syntax sample request: \*A1s1\*I0:Hallo Max!\*I1:5\*Y0:87\*Z

Syntax sample response: \*A1s1\*Y0:40\*Z



[3C] ShowUserImage

(Action index: 2) Sends an image to the device that should be shown to the user. If text message is not enough this action could be used to show any image to the user. The Image should be aspect ratio x:x.

**Note:** Before a image can be sent the IP address of the external connection must be accepted as image or text source on the device terminal. See action "SetServerName" and variable "IsAllowedToSend".

Depending on current profile state the message will be shown different:

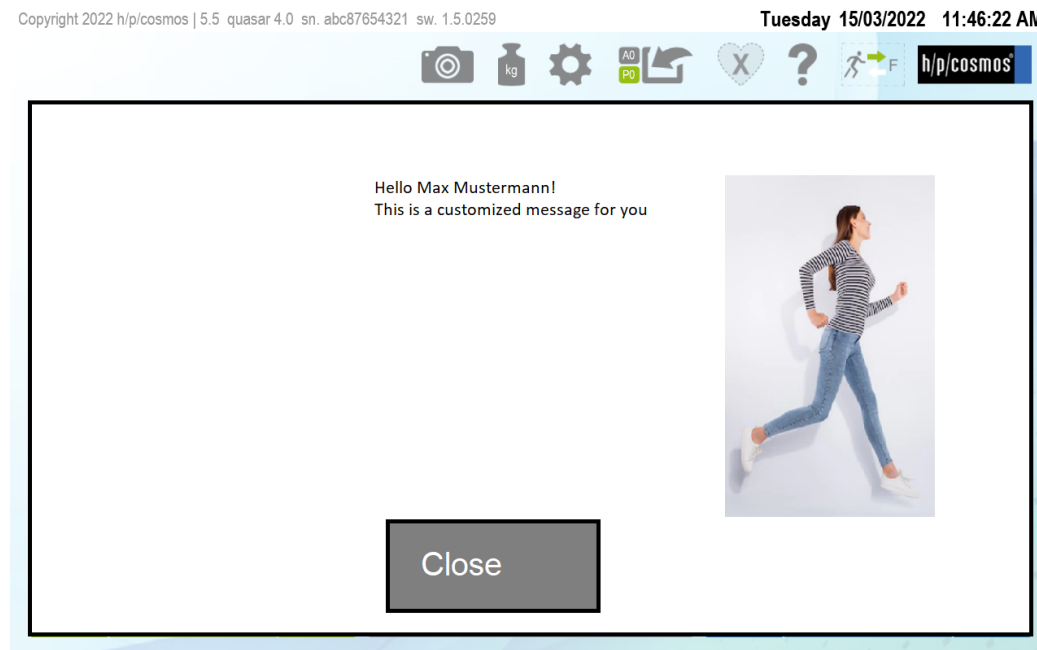


Figure 3: Image in menu

If a profile is running the message will be shown over the workout graph:

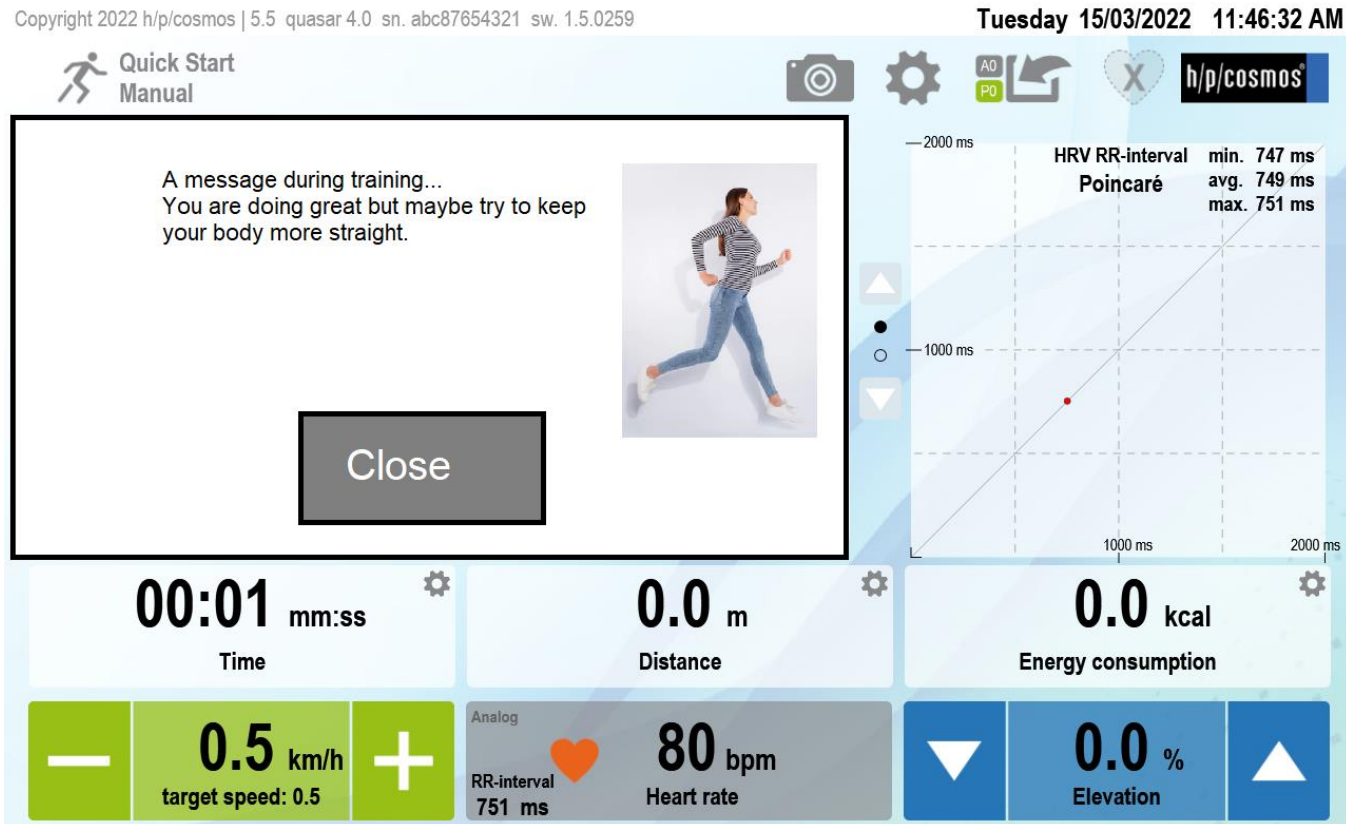


Figure 4: Image during workout

Parameters:

- *image* (direction in - index 0, bin.base64 string):  
The image which should be shown to the user. Must a valid image file (png, bmp, jpg)
- *displayTime* (direction in - index 1, integer): The Specifies the length of time in seconds the image should be displayed. Possible value range: 5-120 seconds. special value: 0 = image must be acknowledged and remains permanently visible.

Syntax sample request: \*A2s1\*I0:aabbcc\*I1:10\*Y0:A9\*Z

Syntax sample response: \*A2s1\*Y0:41\*Z



[3D] StartProfile

(Action index: 3) With this action a profile can be send to the device which is there shown to the user and can be started. This can be done any time. If a profile is already running an online change to the new profile can be done. Note: For security reasons a running countdown e. g. speed change within next 5 seconds will be shown above this message.

External software will be notified with variable "RequestProfileResult" when user presses "No" or "Start".

The profile will be shown as following to the user:

Copyright 2022 h/p/cosmos | 5.5 quasar 4.0 sn. abc87654321 sw. 1.5.0260 Dienstag 15/03/2022 13:31:57

Die externe Steuerung (ServerName) hat einen Vorschlag gesendet:  
 Profil "Profilname" soll gestartet werden:

Schritt	Geschwindigkeit [km/h]	Zeit [mm:ss]	Steigung [%]
1	7,2	05:00	0,0
2	9,0	03:00	0,0
3	7,2	02:00	0,0
4	9,0	03:00	0,0
5	7,2	02:00	0,0
6	9,0	03:00	0,0
7	7,2	02:00	0,0
8	9,0	03:00	0,0
9	7,2	02:00	0,0

Optionale Profilbeschreibung laut Profil.xml

**Nein** **Start**

— **0,0 km/h** +  
Startgeschwindigkeit

Analog **80 bpm**  
RR-Intervall 740 ms Herzfrequenz

▼ **0,0 %** ▲  
Steigung

Figure 5: Start ext. profile

Parameters:

- *ProfileXML* (direction in - index 0, string):  
The profile xml which should be startet.

**Syntax sample request:** \*A3s1\*I0:Profile><Header><CreationDate>2022-02-21 07:58:32</CreationDate><Creator>vendor X</Creator><DeviceType>Treadmill</DeviceType><Name Language="en">Profile XYZ for Workoutplan 1</Name><Name Language="de">Profil XYZ für Trainingsplan 1</Name><ShortDescription Language="de">Einsteigerprofil Trainingswoche 2</ShortDescription><ShortDescription Language="en">Profile beginners training week 2</ShortDescription></Header>\r\n<ProfileSteps><Step Phase="Main" StepType="Standard" StartManual="False"><Interval Type="Time">10</Interval><SpeedValue>2.778</SpeedValue><AccelerationValue Type="Exact">0.611</AccelerationValue><ElevationValue>0.000</ElevationValue></Step><Step Phase="Main" StepType="Standard" StartManual="False"><Interval Type="Time">600</Interval></Step></ProfileSteps></Profile>\*Y0:BF\*Z

**Syntax sample response:** \*A3s1\*Y0:42\*Z

**[3E] RequestPermissions**

(Action index: 4) With this action an external control can request permissions on the device terminal. Permissions for e. g. sending text and images must be granted before any image or text can be displayed on the screen. With this action an IP validation will be done on the device terminal. If the IP address is already trusted and requested permission granted nothing is shown to the user. If IP address is not trusted yet or a new permission is requested, the user will see a request if external software with given name and IP address should be allowed to requested permissions for this device.

External software will be notified with variable "Permissions" when user presses "Decline" or "Allow" on the screen.

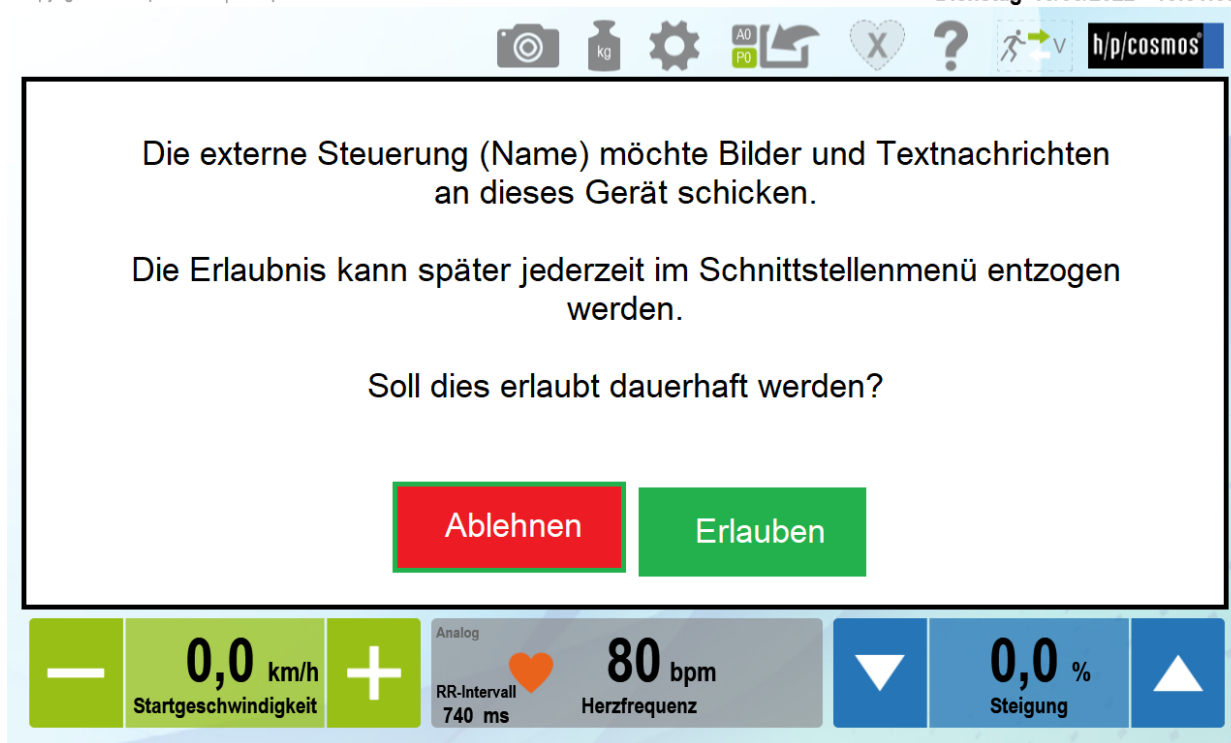


Figure 6: Request send allowance

Parameters:

- *Ext. software name* (direction in - index 0, string):  
The name of the ext. software requesting the permission. This will be shown to the user for identifying ext. software. It can be changed on the device if the user wants to use another name.
- *Requested permissions* (direction in - index 1, bit mask):  
Permissions which should be granted to the ext. software. See variable "Permissions" for details.

Syntax sample request: \*A4s1\*I0:AktivSystem\*I0:1\*Y0:B2\*Z

Syntax sample response: \*A4s1\*Y0:43\*Z

[3F] **SetPersonData**

(Action index: 4) With this action an external control can optionally set the display name and ID of the person using the device. The provided information will be permanently shown on the user terminal in the upper space next to the date and time. External control is responsible for clearing the data sending an empty name and ID.

Parameters:

- *Name* (direction in - index 0, string):  
The name of the person using the device.
- *PatientID* (direction in - index 1, string):  
The patient ID which should be shown on the device terminal.

Syntax sample request: \*A5s1\*I0:Max Mustermann\*I1:1234546\*Y0:D8\*Z

Syntax sample response: \*A5s1\*Y0:44\*Z

[4] **Variables**

h/p/cosmos extended remote service defines with following variables:

**Note:** If no other description is given, values are defined as floating point values expected with max. two decimal places and maximal value +/- 1.000.000. Integer values are defined as signed 32-bit values. String values max.

[4A] **RFID**

(Variable index: 0, string) The current RFID detected on the Terminal. If no RFID is in reach an empty string is send.

Syntax sample query request: \*Q0s1\*Y0:49\*Z

Syntax sample query response: \*Q0s1:11223344556677\*Y0:61\*Z

[4B] **TerminalLanguage**

(Variable index: 1, string) The current selected language on the terminal. If external software wants to send message in the current selected language on the terminal it can use this variable to check the current language.

Syntax sample query request: \*Q1s1\*Y0:50\*Z

Syntax sample query response: \*Q1s1:de\*Y0:53\*Z

[4C] **Permissions**

(Variable index: 2, bit mask) Indicates the current permission of the external device. External control can request permissions with action RequestPermissions. Permissions will be stored permanently on the device until it is removed by the user. Following permissions are available right now:

Bit 0 = Send images and texts

Syntax sample query request: \*Q2s1\*Y0:51\*Z

Syntax sample query response: \*Q2s1:1\*Y0:BC\*Z

[4D] **RequestedProfileResult**

(Variable index: 3, integer) Indicates if last sent profile was accepted on the terminal.

Possible values:

0 = not accepted,

1 = pending request,

2 = accepted

Syntax sample query request: \*Q2s1\*Y0:51\*Z

Syntax sample query response: \*Q2s1:1\*Y0:BC\*Z