h/p/cosmos coscom v3 protocol

implementation notes

Created for:
h/p/cosmos sports & medical gmbh
Am Sportplatz 8
DE 83365 Nussdorf-Traunstein
Germany
phone + 49 86 69 86 42 0
fax + 49 86 69 86 42 49
email@h-p-cosmos.com
www.h-p-cosmos.com

Author:
M. Sc. Andreas Feil, Altotec GmbH, Altofing 7, 83367 Petting / Germany

Date: 2016-03-09

[1.]Content

[2.]Introduction

This document gives a brief overview of the h/p/cosmos coscom v3 protocol for MCU5 running machine devices. cosom v3 is based on the XDOP (Inde**x**ed **D**evice **O**bject **P**rotocol) protocol. A detailed description including an EBNF grammar notation of this protocol is found in the PDF document “XDOP DA 103”.
This document adds some specific coscom v3 implementation notes to the general protocol description. In order to understand the following notes you should take a brief look at the protocol object model and message types which are described in the XDOP DA 103 document.
A complete overview of all possible coscom v3 control elements are described in the document “MCU5 device description with examples.pdf”. Further information about basic device control can be found in the h/p/cosmos coscom v3 services notes document.

[3.]The coscom protocol

The h/p/cosmos coscom interface protocol has it's origin in the year 1992 and has been developed for safe, reliable and advanced communication, control and links between different ergometers like running machines, treadmills, bicycle ergometers, ladder ergometers etc., as well as control and monitoring equipment like PC, EMG, ECG, EKG, ergospirometry, VO2max systems, metabolic carts, cardiopulmonary stress test systems, biomechanics and motion analysis systems, fitness and sports medical as well as lactate evaluation and analyzing software, etc. Late 2007 a new version of the protocol - coscom v3 - was introduced which expands the device control possibilities.

[4.]Important Notes, Safety Precautions, Warnings

The coscom v3 protocol can be used with h/p/cosmos running machines and OEM running machines which are equipped with MCU4 and MCU5 control board (UserTerminal).

			
h/p/cosmos mercury med	h/p/cosmos mercury lt med	h/p/cosmos para control® 4.0	MCU 5 control board

Do not use coscom v3 with a MCU4 device which has a firmware version between version MCU4 EPROM Firmware version 4.04.1 and 4.04.4. These versions contain implementation errors for coscom v3 features and must not be used with coscom v3. From MCU4 Firmware v 4.04.5.0025 the coscom v3 protocol can also be used, so please make sure the Firmware was updated this 4.04.5 or higher.

Pay attention to all safety instructions and warnings as well as the chapters “intended use” and “forbidden use” of the operation and service manual of the h/p/cosmos running machines and also the h/p/cosmos para control® 4.0 software!

Always activate “failsafe timeout mode”, so the running machines stops automatically in case of termination of the interface communication.

Always communicate the “status” mode of the running machine with any host programs.
In case the running machine was stopped via STOP button on the running machine or via any other reason (for example power failure for the running machine supply), the host program must realize this “stop status” and must terminate automatically any control functions of speed any elevation control of the running machine. If not, a user may be caught by surprise and unexpected command from host program, although the user pressed STOP on the running machine before. This may lead to serious accident.

[5th] coscom v3 specific implementation notes

[5.A] Descriptions

XDOP is an universal protocol which is designed for self description of devices and services. Therefore three different types of descriptions exit:

- Root description: Contains the xdop description version (currently only 1.0 exists), a small device overview including the device list.
- Device description: Contains information about the actual device and manufacturer and a list of all available services.
- Service description: Contains information about the service capabilities: Actions and variables.

coscom v3 is an actual schema of XDOP for ergometers.

In case of the h/p/cosmos MCU4Treadmill and MCU5Treadmill the root description is some kind of fix and will not change anymore. You need not query the root description at all.

You should query the device description with "*D1*Z (short for "*D1d0*Z). In the description you should check the "deviceType" for "urn:schemas-coscom-org:device:MCU5Treadmill:1" in order to verify that a MCU5Treadmill is connected. An application in general should save the service indices of this description and use them in all coscom v3 commands for a specific service. In case of h/p/cosmos MCU5Treadmill this indices are also some kind of fix and will not change anymore. In the current firmware for example the service "urn:schemas-coscom-org:service:TreadmillMeasures:1" has service index 2.

With this service index you can query the service description, which contains all necessary information in order to control the running machine. XDOP defines actions, variables and events for controlling the device. These elements all have specific indices which are described by the service description.
A detailed description of all this descriptions is also found in the “XDOP DA 103.pdf” document.
In general your application should query all service descriptions when establishing a connection and use the action and variable indices for further communication with the running machine device. **In case of coscom v3 these indices are all constant and won’t change anymore. But you should check the servicetype and its version in order to confirm that the actual service implements the interfaces you need. See the next chapter version policy for futher information.**

[5.B] Version policy

In the device description you should verify that the service you need is in the actual service list of the device. The device description string and the device type version describe a constant interface of a service. If a service is expaned (e. g. new actions will be added) the service type will stay the same but the version will be raised. The changes in that case are downwards compatible. If there is a need to change the service in a way that is not compatible with the former versions the service type string will be changed.

At present all service types have version 1. They are not likely to change in future but it's possible that future situations will enforce use to make some changes.

If you can't find the service type you need in your application or the service type version is to low you must inform your user that the device can't be controlled with your software. If that is ever the case, contact h/p/cosmos for information on newer coscom v3 versions or visit www.coscom.org.

[5thC] Syntax examples

[5thC1] Actions

Attached you will find the document "MCU5 device description with examples". In this document all services including their actions and variables are described. This version additionally contains coscom v3 protocol examples for all actions. For information about basic device control refer the document h/p/cosmos coscom v3 services notes.

[5.C2] Variables

Following a short example for a querying a variable:

Service: TreadmillMeasures, variable: HeartRate

coscom request syntax: *Q4s2d0*Z
coscom response syntax: *Q4s2d0:140*Z

This example queries the current heart rate value. In case of the MCU5Treadmill the device index is always 0. The service index for the TreadmillMeasures service is 2 and the variable index is 4. The returned heart rate value in this example is 140. For detailed message syntax refer the XDOP DA 103 document.

[5.C3] Eventing

coscom v3 defines so called events. With these events a device can inform a client that its state (variable value) has changed.
Events are service specific and in order to receive them you have to subscribe to them. Following example subscribes to the events of the TreadmillMeasures (service index 2) service:

coscom request syntax: *S1s2d0*Z
coscom response syntax: *S1s2d0*Z

In general you would receive all variable changes of that service now. The description of the event message format is located in the "XDOP DA 103.pdf".

In case of the MCU5 the measures service is a kind of exception. Because sending all possible changes of acceleration, speed, distance, time, elevation, etc. could cause a “flood” of events there is an additional action called “SetEventMask” in this service. It controls the events which are sent. If you want to receive all possible events you have to set its value to 0xff. For further information of the action see the “MCU5 device description with examples.pdf” or the h/p/cosmos coscom v3 services notes document.

If you have subscribed to service events you can cancel the subscribtion everytime with an unsubscribe message to the device. It’s basicly the same message as the subscription but with a different index for the main index “S”.

coscom request syntax: *S**0**s2d**0***Z
coscom response syntax: *S**O**s2d0*Z

[5.C4]Error responses

Coscom v3 has the ability to define errors. Every command to a device can respond in an error message from the device if the actual command couldn’t be processed. You should always be able to process error responses from the device. The formal specification of the message format is shown in the XDOP device architecture document. The message defines two elements: the error number (F0) and the optional error description (F1). Beside general errors every action can define custom errors. A list of all defined action errors for h/p/cosmos coscom v3 is shown next to every action in the device description documents. Here is a small sample message which could be the result of an invalid speed setting in the SetDriveSpeed action of the TreadmillDirectControlService:

coscom request syntax: *A2s4*I0:7*I1:3*I2:0*Z
coscom response syntax: *A2s4*F0:831*F1:Speed out of range*Z

The application tries to set a new speed value of 7 m/s with an acceleration index 3 (value for exact acceleration is 0). The device response contains the error code 831 and the optional errer message “Speed out of range”. In this case the running machines maximum speed is less than 7 m/s or the options settings for the maximum speed limit the maximum speed.

[6.]Safety Features and Implementation notes

At this point some important safety features of the running machine devices will be discussed.

[6.A] Fail safe

The failsafe action is a basic security feature of the running machine. It sets a safety timeout in case the interface communication is interrupted (e. g. someone pulls the RS232 interface plug). If that timeout elapses the running machine will stop automatically.

You should constantly set this timeout in your application in order to check if a failsafe occurred.
Note: The failsafe will only occur when the running machine is actually in run mode. In selection mode you can set the timeout but it will not elapse till the running machine is started. If a failsafe elapses you have to set it again in order to restart the feature. It will not restart automaticly.

Action parameters:

newTimeout (byte): Sets the new timeout in 1/10 seconds (from 0 to 25.5 seconds). You should use a small timeout interval (<= 1 sec.) in order to quickly dectect an occurring failsafe.

oldTimeout (byte): Returning parameter that indicates the last set value of the timeout. This returning parameter is very useful because it indicates the state of the failsafe. If the fail safe wasn't set before it will be 0. If the fail safe was set before it should contain the previously set value, if it doesn't a failsafe occurred since you last set the time out.

A small protocol example for setting a new failsafe timeout to 1 second:

coscom request syntax: *A5s0d0*I0:10*Z
cosocm response syntax a: *A5s0d0*O1:10*Z
cosocm response syntax b: *A5s0d0*O1:0*Z

Sample response 'a' contains 10 as returning parameter for the old failsafe value. This means a failsafe was set before and has not elapsed. Sample response 'b' contains 0 as returning parameter. This means either you haven't set the timeout before or the timeout elapsed since you last set it.

[6.B] **Check sum**

coscom defines an optional message checksum. The checksum is the last part of a coscom message and is indicated by the index Y0.

You enable the checksum by invoking the action SetProtocolParams in the service TreadmillGeneralConfig. This action has the following parameters:

- processChecksum (boolean): Enables the processing of an 2 hex digit (modulo 256) checksum with element Y0.
- appendCRLF (boolean): Appends the characters CR and LF at the end of a message (after *Z).
- outputDescNames (boolean): If set, the XDOP descriptions contain the optional element descriptions.

If you would like to set all three parameters to true a message will be look like this:

coscom request syntax: *A0*I0:1*I1:1*I2:1*Z
coscom response syntax: *A0*Z

If "processChecksum" is set, a "SetDriveSpeed" Action command could look like this:

coscom request syntax: *A2s4*I0:2*I1:3*I2:0***Y0:73***Z
coscom response syntax: *A2s4***Y0:44***Z

The calculation of the checksum is done by getting all UTF-8 bytes for the characters starting by the first '*' character of the message and ending before the message end "*Z". The bytes have to be added and computed modulo 256. The hexadecimal representation of the result is the actual checksum.

[6.C] Stop status

As mentioned above always communicate the “status” mode of the running machine with any host programs.
In case the running machine was stopped via the STOP button on the running machine or for any other reason (for example power failure for the running machine supply), the host program must realize this “stop status” and must terminate automatically any control functions of speed any elevation control of the running machine.

The best way to fulfill this demand is to monitor the variable ControlStatus in service TreadmillUserControl. This is best done by enabling eventing for this service and checking the incoming event messages. The status can hold the following values:

- 0 = Stop
- 1 = Run
- 2 = Pause
- 8 = Graded test pause freezed
- 9 = Emergency stop

If you receive a status other than “Run”, cancel all automatic changes of speed, acceleration and elevation. If you receive “Pause” or “Graded test pause freezed”, the user has paused the current workout therefore also no automatic changes are allowed during this state. You can either cancel or only pause your automatic control in this case. If the user continues a paused workout (state will change to “Run” again), a paused automatic control could also continue.

An exception of the above rules are situations where status “Pause” is part of the external control. In this case a new speed command is allowed in order to continue the current control. This should only be used when really necesarry. For maximum safety considerations external control should be stopped when “Pause” status is received.

A sample event message (status “stop”) could look like this:

coscom event syntax sample: *E8s3*V1:**0***Z

For further information on event messages review the XDOP device architecture document.

[6.D] SetDriveSpeed

With this action a new speed and acceleration respectively deceleration can be set. You should query the actual device speed and acceleration ranges before, in order to verify that the desired values lie within the permitted range. Before making automatic speed changes you should display the user a note. Unexptected changes may catch the user by surprise which can lead to serious accidents.

Beside of settting a new speed value you must declare an acceleration respectively deceleration value. You can either use a predefined index value **or** an exact acceleration value in m/s². If you set an index value other than “0” the exact accerlation value will be ignored. On the other side if you want to use an exact value you must specify “0” for the index value.

Action parameters:

speed (r4): The new final Speed in m/s
accelIndex (ui1): The acceleration level index. Following values are possible:

0 = Use "acceleration" argument
1 = (Max. FU speed) / 131 s
2 = (Max. FU speed) / 65.5 s
3 = (Max. FU speed) / 32.8 s
4 = (Max. FU speed) / 16.0 s
5 = (Max. FU speed) / 8.0 s
6 = (Max. FU speed) / 5.0 s
7 = (Max. FU speed) / 3.0 s

acceleration (r4): The exact acceleration value in m/s². You can always specify positive values regardless of acceleration or deceleration.

Protocol sample:

coscom request syntax: *A2s4*I0:2*I1:3*I2:0*Z
coscom response syntax: *A2s4*Z

Invalid speed value (error response from device):

coscom request syntax: *A2s4*I0:7*I1:3*I2:0*Z
coscom response syntax: *A2s4*F0:831*F1:Speed out of range*Z

[6.E] Warn user before changes

You should warn an user before automatic changes of speed, acceleration and elevation. You can do this by visual effects in your application or/and by using the buzzer of the running machine device. The action “Beep” in service Terminal is designed to activate the buzzer for a certain time.

The action has one parameter:

duration (ui1): The beep time in 1/100 seconds (0 to 2.55 s)

A small coscom syntax sample which will activate the buzzer for 1 second:

coscom request syntax: *A4s1*I0:100*Z
coscom response syntax: *A4s1*Z

[7th] h/p/cosmos coscom v3 .NET objects

You can either implement the protocol yourself or if you’re using the Microsoft .NET Framework you can use the h/p/cosmos coscom v3 .NET Objects. These class libraries implement the coscom v3 protocol and define a high level interface for the coscom v3 device and service model. You will find a method for every action that the h/p/cosmos running devices define. For further information see www.coscom.org.

[8th]h/pcosmos coscom v3 .NET Controls

If you want to rebuild the build-in control and you're using the Microsoft .NET Framework you can use the h/p/cosmos coscom v3 .NET Controls. This class library contains controls for the 6 displays and user keys. For further information and a small sample project for visual studio visit www.coscom.org.

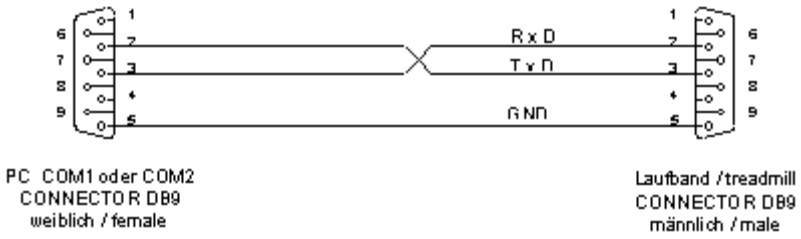
[9.]Troubleshooting

[9.A] Running machine protocol settings

The MCU5Treadmill defines 2 possible baud rates (9600 and 115200) for coscom v3. Depending of the serial port on which the application is connected and the port settings in the user options:
COM1: User Option 20 ("1 coscom" is baud rate 9600 and "20 coscom 3" is baud rate 115200)
COM2: User option 21 ("1 coscom" is baud rate 9600)
COM3: always coscom v3 (baud rate 115200)
COM4: User option 24 (if available) ("20 cosocm 3" is baudrate 115200)

[9.B] Interface cable

In order to connect your pc to the running machine you must use an interface cable which is wired like in following picture:



For further information visit http://coscom.org/#Interface_Cable

