



ISO/IEC 29341-19-10

Edition 1.0 2011-10

INTERNATIONAL STANDARD



**Information technology – UPnP device architecture –
Part 19-10: Solar Protection Blind Device Control Protocol – Two Way Motion
Motor Service**



THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2011 ISO/IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about ISO/IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00



ISO/IEC 29341-19-10

Edition 1.0 2011-10

INTERNATIONAL STANDARD



**Information technology – UPnP device architecture –
Part 19-10: Solar Protection Blind Device Control Protocol – Two Way Motion
Motor Service**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE

J

ICS 35.200

ISBN 978-2-88912-690-3

CONTENTS

1	Overview and Scope.....	3
2	Service Modeling Definitions.....	3
2.1	Service Type	3
2.2	State Variables.....	4
2.2.1	OperationMode	5
2.2.2	ServiceLocked	5
2.2.3	Position	5
2.2.4	PositionArgType	5
2.2.5	Relationships between State Variables.....	6
2.3	Eventing and Moderation	6
2.4	Actions.....	6
2.4.1	(Void) Open().....	7
2.4.2	(Void) Close()	8
2.4.3	(Void) Stop().....	9
2.4.4	(String) GetOperationMode(RetVal).....	10
2.4.5	(Void) SetOperationMode(String)	10
2.4.6	(Void) IsLocked(Boolean).....	11
2.4.7	(Void) Lock().....	11
2.4.8	(Void) UnLock()	12
2.4.9	(Float) GetPosition(RetVal)	13
2.4.10	(Void) SetPosition(I1).....	13
2.4.11	(String) GetPositionArgType(RetVal)	14
2.4.12	Non-Standard Actions Implemented by a UPnP Vendor	15
2.4.13	Relationships between Actions.....	15
2.4.14	Common Error Codes	15
2.5	Theory of Operation.....	15
3	XML Service Description	17
4	Test	20
Table 1: State Variables		4
Table 1.1: allowedValueList for <u>OperationMode</u>		4
Table 1.2: DefaultValue for <u>OperationMode</u>		4
Table 1.5: allowedValueRange for <u>Position</u>		4
Table 1.6: DefaultValue for <u>Position</u>		4
Table 1.7: allowedValueList for <u>PositionArgType</u>		5
Table 1.8: DefaultValue for <u>PositionArgType</u>		5
Table 2: Event Moderation		6
Table 3: Actions.....		7
Table 4.1: Arguments for <u>Open</u>		7
Table 4.2: Arguments for <u>Close</u>		8
Table 4.3: Arguments for <u>Stop</u>		9

Table 4.4: Arguments for <u>GetOperationMode</u>	10
Table 4.5: Arguments for <u>SetOperationMode</u>	10
Table 4.6: Arguments for <u>IsLocked</u>	11
Table 4.7: Arguments for <u>Lock</u>	11
Table 4.8: Arguments for <u>UnLock</u>	12
Table 4.9: Arguments for <u>GetPosition</u>	13
Table 4.10: Arguments for <u>SetPosition</u>	13
Table 4.11: Arguments for <u>GetPositionArgType</u>	14
Table 5: Common Error Codes	15

INFORMATION TECHNOLOGY – UPNP DEVICE ARCHITECTURE –

Part 19-10: Solar Protection Blind Device Control Protocol – Two Way Motion Motor Service

FOREWORD

- 1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.
- 2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.
- 3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.
- 4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.
- 6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.
- 7) All users should ensure that they have the latest edition of this publication.
- 8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.
- 9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 29341-19-10 was prepared by UPnP Forum Steering committee¹, was adopted, under the fast track procedure, by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 29341 series, under the general title *Information technology – UPnP device architecture*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

¹ UPnP Forum Steering committee, UPnP Forum, 3855 SW 153rd Drive, Beaverton, Oregon 97006 USA. See also "Introduction".

IMPORTANT – The “colour inside” logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this publication using a colour printer.

1 Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0.

This service-type enables the following functions:

- Driving a motor between two fixed end limits.
- Reaching any position written in percentage of the full run (optional).
- Returning the actual blind position written in percentage of the full run (optional).

This service template does not address:

- Any motorization whose run is not bounded.
- Any speed control.

2 Service Modeling Definitions

2.1 Service Type

The following service type identifies a service that is compliant with this template:

urn:schemas-UPnP-org:service:*TwoWayMotionMotor:1*.

2.2 State Variables

Table 1: State Variables

Variable Name	Req. or Opt. ^a	Data Type	Allowed Value ^b	Default Value ^b	Eng. Units
<u>OperationMode</u>	<u>R</u>	<u>String</u>	" <u>Manual Unprotected</u> ", " <u>Manual Protected</u> ", " <u>Automatic</u> "	<u>Vendor-defined</u>	
<u>ServiceLocked</u>	<u>O</u>	<u>Boolean</u>	<u>0, 1</u>	<u>1</u>	
<u>Position</u>	<u>O</u>	<u>I1</u>	<u>0 to 100</u>	<u>Vendor-defined</u>	
<u>PositionArgType</u>	<u>O</u>	<u>String</u>	" <u>End Limits</u> ", " <u>Continuous</u> "	<u>Vendor-defined</u>	
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	<i>X</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>
^a R = Required, O = Optional, X = Non-standard ^b Values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.					

Table 1.1: allowedValueList for OperationMode

Value	Req. or Opt.
<u>Manual Unprotected</u>	<u>O</u> a
<u>Manual Protected</u>	<u>O</u> a b
<u>Automatic</u>	<u>O</u> b
<u>Vendor-defined</u>	<u>O</u> b
^a At least one out of "Manual Unprotected" and "Manual Protected" must be implemented. ^b If one of those optional values is implemented "ServiceLocked" must be implemented.	

Table 1.2: DefaultValue for OperationMode

Value	Req. or Opt.
<u>Vendor-defined</u>	<u>R</u>

Table 1.5: allowedValueRange for Position

	Value	Req. or Opt.
Minimum	<u>0</u>	<u>R</u>
Maximum	<u>100</u>	<u>R</u>
Step	<u>Vendor-defined</u>	<u>O</u>

Table 1.6: DefaultValue for Position

Value	Req. or Opt.
<u>Vendor-defined</u> ^a	<u>R</u>
^a This value must reflect the real position of the device.	

Table 1.7: allowedValueList for PositionArgType

Value	Req. or Opt.
<u>End Limits</u>	<u>R</u>
<u>Continuous</u>	<u>R</u>

Table 1.8: DefaultValue for PositionArgType

Value	Req. or Opt.
<u>Vendor-defined</u> ^a	<u>R</u>
^a This value cannot be changed afterward.	

2.2.1 OperationMode

This variable describes with “ServiceLocked” the actual operation mode of the device:

- “Manual Unprotected” allows the user to control the device and disables all the protections available. “Manual Unprotected” is required.
- “Manual Protected” allows the user to control the device and enable all the protections available. “Manual Protected” is optional,
- “Automatic” disables any manual order. “Automatic” is optional.

2.2.2 ServiceLocked

This variable defines if the device is locked or not:

- 1: actions that get information from the service and those that change the operation mode (i.e. “OperationMode” + “ServiceLocked”) are enabled. All other actions are disabled. “ServiceLocked” overcomes automation and protections.
- 0: all actions are enabled.

“ServiceLocked” is optional, but it must be implemented if automation or protections can be turned on.

2.2.3 Position

This variable stores the actual position of the device. It’s written in percentage of the full run:

- 0 is at one end of the full run. For a human being looking at the device, it should be synonymous with closed, down, far right or switch limit of a clockwise move.
- 100 is at the other end of the full run. For a human being looking at the device, it should be synonymous with open, up, far left or switch limit of a counter-clockwise move.
- 50 is the half run. It’s also the value of “Position” if an accurate number cannot be provided which is memorized by “PositionArgType”.

“Position” can be implemented if and only if the service can inform when a limit switch is reached and which one is it. This variable is optional, but if it’s implemented, “PositionArgType” must also be implemented.

2.2.4 PositionArgType

This variable describes how to interpret “Position”:

- “End Limits”: the service can only determine if the device has reached a limit switch and if so, which one.
- “Continuous”: the service can always give an accurate value.

“PositionArgType” is implemented if and only if “Position” is implemented. This state variable must be defined by the vendor and cannot be changes afterward.

2.2.5 Relationships between State Variables

“ServiceLocked” must be implemented if “OperationMode” can value Manual Protected” or “Automatic”.

“PositionArgType” and “Position” cannot be implemented one without the other.

2.3 Eventing and Moderation

Table 2: Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate ^a	Logical Combination	Min Delta per Event ^b
<u>OperationMode</u>	<u>Yes</u>	<u>No</u>			
<u>ServiceLocked</u>	<u>Yes</u>	<u>No</u>			
<u>Position</u>	<u>Yes</u>	<u>Yes</u>			<u>5</u>
<u>PositionArgType</u>	<u>No</u>	<u>n/a</u>			
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>
^a Determined by N, where Rate = (Event)/(N secs). ^b (N) * (allowedValueRange Step).					

2.4 Actions

The service provides three kinds of action:

- to control the device (“Open”, “Close”; “Stop”; “SetPosition”). “SetPosition” is implemented if and only if the device can perform it,
- to know and to set the operation mode (“GetOperationMode”, “SetOperationMode”, “IsLocked”, “Lock” and “UnLock”),
- to have information about the device position (“GetPosition”, “GetPositionArgType”). These functions are implemented if and only if the variables “Position” and “PositionArgType” are implemented.

Immediately following this table is detailed information about these actions, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 3: Actions

Name	Req. or Opt. ^a
<u>Open</u>	<u>R</u>
<u>Close</u>	<u>R</u>
<u>Stop</u>	<u>R</u>
<u>GetOperationMode</u>	<u>R</u>
<u>SetOperationMode</u>	<u>R</u>
<u>IsLocked</u>	<u>O</u>
<u>Lock</u>	<u>O</u>
<u>UnLock</u>	<u>O</u>
<u>GetPosition</u>	<u>O</u>
<u>SetPosition</u>	<u>O</u>
<u>GetPositionArgType</u>	<u>O</u>
<i>Non-standard actions implemented by an UPnP vendor go here.</i>	X
^a R = Required, O = Optional, X = Non-standard	

2.4.1 (Void) Open()**2.4.1.1 Arguments****Table 4.1: Arguments for Open**

Argument	Direction	relatedStateVariable

2.4.1.2 Dependency on State

This action is disabled if “ServiceLocked” is True (see description 2.4.1.4)

“OperationMode” changes how this action is executed (see description 2.4.1.4).

2.4.1.3 Effect on State

If the action is done:

If “Position” is implemented then it’s incremented according to the sensor (expressed in percentage of the full run).

If the protection must stop the move then “ServiceLocked” switches True.

2.4.1.4 Description

If the device is locked then an error 700 (“Forbidden”) is returned.

Otherwise, depending on the operation mode, the system acts differently:

- “OperationMode” = “Manual Unprotected”

Drives the motor in the way that increments “Position” (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise) until one of the following conditions is verified:

“Stop”, “Close” or “SetPosition” with a position requested smaller than the actual one, is received,

The device is fully opened.

- “OperationMode” = “Manual Protected”

If the protection doesn't allow the device to open, an error 701 (“Not Allowed”) is returned. Then the protection decides if the device must be locked or not.

If the protection allows the device to open, the system drives the motor in the way that increments “Position” (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise) until one of the following conditions is verified:

“Stop”, “Close” or “SetPosition” with a position requested smaller than the actual one, is received,

The device is fully opened.

The protection stops the opening. If it happens, “ServiceLocked” must immediately switch True. Note: the protection can still drive the motor to put the device in a safe configuration.

- “OperationMode” = “Automatic”

An error 700 (“Forbidden”) is returned.

2.4.1.5 Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.
<u>700</u>	<u>Forbidden</u>	<u>The device is locked or in the automatic mode</u>
<u>701</u>	<u>Not Allowed</u>	<u>The protection doesn't allow the action.</u>

2.4.2 (Void) Close()

2.4.2.1 Arguments

Table 4.2: Arguments for Close

Argument	Direction	relatedStateVariable

2.4.2.2 Dependency on State

This action is disabled if “ServiceLocked” is True (see description 2.4.2.4)

“OperationMode” changes how this action is executed (see description 2.4.2.4).

2.4.2.3 Effect on State

If the action is done:

If “Position” is implemented then it's decremented according to the sensor (expressed in percentage of the full run),

If the protection must stop the move then “ServiceLocked” switches True.

2.4.2.4 Description

If the device is locked then an error 700 (“Forbidden”) is returned.

Otherwise, depending on the operation mode, the system acts differently:

- “OperationMode” = “Manual Unprotected”

Drives the motor in the way that decrements “Position” (i.e. the device is closing, moving down, moving right or the motor is turning clockwise) until one of the following conditions is verified:

“Stop”, “Open” or “SetPosition” with a position requested greater than the actual one, is received,
the device is fully closed.

- “OperationMode” = “Manual Protected”

If the protection doesn’t allow the device to close, an error 701 (“Not Allowed”) is returned. Then the protection decides if the device must be locked or not.

If the protection allows the device to close, the system drives the motor in the way that decrements “Position” (i.e. the device is closing, moving down, moving right or the motor is turning clockwise) until one of the following conditions is verified:

“Stop”, “Open” or “SetPosition” with a position requested greater than the actual one, is received,

The device is fully closed.

The protection stops the move. If it happens, “ServiceLocked” must immediately switch True. Note: the protection can still drive the motor to put the device in a safe configuration.

- “OperationMode” = “Automatic”

An error 700 (“Forbidden”) is returned.

2.4.2.5 Errors

ErrorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.
<u>700</u>	<u>Forbidden</u>	<u>The device is locked or in the automatic mode.</u>
<u>701</u>	<u>Not Allowed</u>	<u>The protection doesn’t allow the action.</u>

2.4.3 (Void) Stop()

2.4.3.1 Arguments

Table 4.3: Arguments for **Stop**

Argument	Direction	relatedStateVariable

2.4.3.2 Dependency on State

This action is disabled if “ServiceLocked” is True (see description 2.4.3.4)

“OperationMode” changes how this action is executed (see description 2.4.3.4).

2.4.3.3 Effect on State

If the action tries to overcome the protection or the automation then “ServiceLocked” switches True.

2.4.3.4 Description

If the device is locked then an error 700 (“Forbidden”) is returned.

Otherwise, depending on the operation mode, the system acts differently:

- “OperationMode” = “Manual Unprotected”
Stops the device at the current position until a control action (“Open”, “Close”, or “SetPosition”) is received.
- “OperationMode” = “Manual Protected”
If the protection doesn’t allow the device to stop then “ServiceLocked” switches True. Note: the protection can still drive the motor to put the device in a safe configuration.
If the protection allows the device to stop, the system is stop at the current “Position”. It remains motionless until one of the following conditions is verified:
A control action (“Open”, “Close”, or “SetPosition” with a position requested not equal to the actual one) is received,
The protection decides to move the device.
- “OperationMode” = “Automatic”
If the device is moving then “ServiceLocked” switches True. Note: the automation can still drive the motor to put the device in a safe configuration.

2.4.3.5 Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.
<u>700</u>	<u>Forbidden</u>	<u>The device is locked or in the automatic mode</u>
<u>701</u>	<u>Not Allowed</u>	<u>The protection doesn’t allow the action.</u>

2.4.4 (String) GetOperationMode(RetVal)

2.4.4.1 Arguments

Table 4.4: Arguments for GetOperationMode

Argument	Direction	relatedStateVariable
<u>RetOperationMode</u>	<u>Out – RetVal</u>	<u>OperationMode</u>

2.4.4.2 Description

Returns the current value of “OperationMode” in “RetOperationMode”.

2.4.4.3 Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.

2.4.5 (Void) SetOperationMode(String)

2.4.5.1 Arguments

Table 4.5: Arguments for SetOperationMode

Argument	Direction	relatedStateVariable
<u>NewOperationMode</u>	<u>In</u>	<u>OperationMode</u>

2.4.5.2 Dependency on State

Each not implemented value of “OperationMode” is an invalid argument for “SetOperationMode”.

2.4.5.3 Effect on State

Sets the new value of “OperationMode”.

2.4.5.4 Description

If “NewOperationMode” is an operation mode implemented then “OperationMode” gets this new value.

Otherwise an error 702 (“Disabled”) is returned.

2.4.5.5 Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.
<u>702</u>	<u>Disabled</u>	<u>The operation mode request is disabled.</u>

2.4.6 (Void) IsLocked(Boolean)

2.4.6.1 Arguments

Table 4.6: Arguments for IsLocked

Argument	Direction	relatedStateVariable
<u>RetLocking</u>	<u>Out – RetVal</u>	<u>ServiceLocked</u>

2.4.6.2 Dependency on State

This action is implemented if and only if “ServiceLocked” is implemented.

2.4.6.3 Description

Returns the current value of “ServiceLocked” in “RetLocking”.

2.4.6.4 Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture clause on Control.
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.

2.4.7 (Void) Lock()

2.4.7.1 Arguments

Table 4.7: Arguments for Lock

Argument	Direction	relatedStateVariable

2.4.7.2 Dependency on State

This action is implemented if and only if “ServiceLocked” is implemented.

2.4.7.3 Effect on State

If “ServiceLocked” is implemented then switches it True.

2.4.7.4 Description

Switches “ServiceLocked” True and stops immediately any move.

2.4.7.5 Errors

ErrorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture clause on Control.
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.

2.4.8 (Void) UnLock()

2.4.8.1 Arguments

Table 4.8: Arguments for UnLock

Argument	Direction	relatedStateVariable

2.4.8.2 Dependency on State

This action is implemented if and only if “ServiceLocked” is implemented.

2.4.8.3 Effect on State (if any)

If “ServiceLocked” is implemented then it switches False

2.4.8.4 Description

If the device is performing a safe move then an error 701 (“Not Allowed”) is returned. Otherwise, “ServiceLocked” switches False and any move is immediately stopped.

2.4.8.5 Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture clause on Control.
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.
<u>701</u>	<u>Not Allowed</u>	<u>The protection doesn't allow the action.</u>

2.4.9 (Float) GetPosition(RetVal)

2.4.9.1 Arguments

Table 4.9: Arguments for GetPosition

Argument	Direction	relatedStateVariable
<u>RetPosition</u>	<u>Out – RetVal</u>	<u>Position</u>

2.4.9.2 Dependency on State

This action is implemented if and only if “Position” is implemented.

2.4.9.3 Description

Returns the current value of “Position” in “RetPosition”.

2.4.9.4 Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture clause on Control.
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.

2.4.10 (Void) SetPosition(I1)

2.4.10.1 Arguments

Table 4.10: Arguments for SetPosition

Argument	Direction	relatedStateVariable
<u>NewPosition</u>	<u>In</u>	<u>Position</u>

2.4.10.2 Dependency on State

This action is implemented if and only if “PositionArgType” equals “Continuous”.

This action is disabled if “ServiceLocked” is True (see description 2.4.10.4)

“OperationMode” changes how this action is executed (see description 2.4.10.4).

2.4.10.3 Effect on State

If the action is performed, “Position” will increase or decrease until it reaches the value of “NewPosition”.

If the protection stops the move, “ServiceLocked” switches True.

2.4.10.4 Description

If “NewPosition” is not between 0 and 100 (included) then an error 601 (“Out of Range”) is returned.

If the device is locked then an error 700 (“Forbidden”) is returned.

Otherwise, depending on the operation mode, the system acts differently:

- “OperationMode” = “Manual Unprotected”

If “NewPosition” equals “Position” then the device is stopped.

If “NewPosition” is greater than “Position” then the motor is driven in the way that increments “Position” (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise).

If “NewPosition” is smaller than “Position” then the motor is driven in the way that decrements “Position” (i.e. the device is closing, moving down, moving right or the motor is turning clockwise).

In these two last cases, the motion is performed until one of the following conditions is verified:

A control action (“Stop”, “Open”, “Close” or “SetPosition”) is received,

“Position” equals “NewPosition”.

- “OperationMode” = “Manual Protected”

If “NewPosition” equals “Position”, the protection must allow the stop. Otherwise, an error 701 (“Not Allowed”) is returned and “ServiceLocked” switches True. Note: in that case, the protection can still drive the motor to put the device in a safe configuration.

If “NewPosition” is greater than “Position”, the protection must allow the motor to be driven in the way that increments “Position” (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise). If the protection doesn’t allow it, an error 701 (“Not Allowed”) is returned and “ServiceLocked” switches True.

If “NewPosition” is smaller than “Position”, the protection must allow the motor to be driven in the way that decrements “Position” (i.e. the device is closing, moving down, moving right or the motor is turning clockwise). If the protection doesn’t allow it, an error 701 (“Not Allowed”) is returned and “ServiceLocked” switches True.

In these two last cases, any move is performed until one of the following conditions is verified:

A control action (“Stop”, “Open”, “Close” or “SetPosition”) is received,

“Position” equals “NewPosition”.

The protection stops the move. If it happens, “ServiceLocked” must immediately switch True. Note: the protection can still drive the motor to put the device in a safe configuration.

- “OperationMode” = “Automatic”

An error 700 (“Forbidden”) is returned.

2.4.10.5 Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture clause on Control.
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.
601	Out of Range	The argument value is not between 0 and 100 included.
<u>700</u>	<u>Forbidden</u>	<u>The device is locked or in the automatic mode.</u>
<u>701</u>	<u>Not Allowed</u>	<u>The protection doesn’t allow the action.</u>

2.4.11 (String) GetPositionArgType(RetVal)

2.4.11.1 Arguments

Table 4.11: Arguments for GetPositionArgType

Argument	Direction	relatedStateVariable
----------	-----------	----------------------

Argument	Direction	relatedStateVariable
<u>RetArgType</u>	<u>Out – RetVal</u>	<u>PositionArgType</u>

2.4.11.2 Dependency on State

This action is implemented if and only if “PositionArgType” is implemented.

2.4.11.3 Description

Returns the current value of “PositionArgType” in “RetArgType”.

2.4.11.4 Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture clause on Control.
402	Invalid Args	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.

2.4.12 Non-Standard Actions Implemented by a UPnP Vendor

To facilitate certification, non-standard actions implemented by UPnP vendors should be included in this service template. The UPnP Device Architecture lists naming requirements for non-standard actions (see the clause on Description).

2.4.13 Relationships between Actions

“SetPosition” with a position requested greater than the actual position acts as “Open”. If the position requested is smaller than the actual position then “SetPosition” acts as “Close”.

“Lock” and “UnLock” are opposite actions.

2.4.14 Common Error Codes

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 5: Common Error Codes

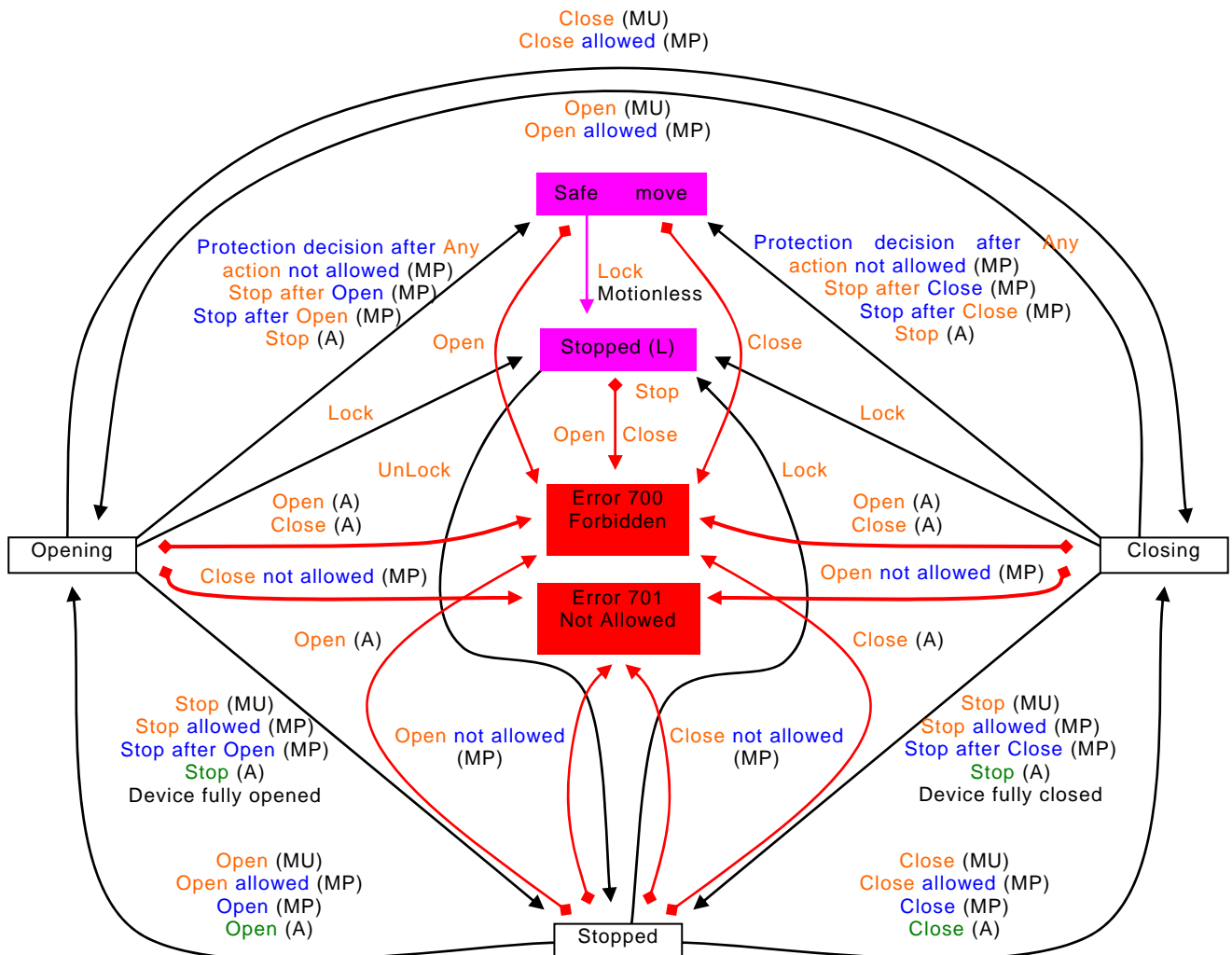
errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture clause on Control.
402	Invalid Args	See UPnP Device Architecture clause on Control.
404	Invalid Var	See UPnP Device Architecture clause on Control.
501	Action Failed	See UPnP Device Architecture clause on Control.
601	Out of Range	The argument value is not between 0 and 100 included.
<u>700</u>	<u>Forbidden</u>	<u>The device is locked or in the automatic mode.</u>
<u>701</u>	<u>Not Allowed</u>	<u>The protection doesn't allow the action.</u>
<u>702</u>	<u>Disabled</u>	<u>The operation mode request is disabled</u>
704-799		Common action errors defined by the UPnP Forum working committees.
<u>800-899</u>	<u>TBD</u>	<u>(Specified by UPnP vendor.)</u>

2.5 Theory of Operation

The device cannot be moved if the service is locked (L). When it's unlocked, it can be controlled manually in the manual unprotected operation mode (MU) or under the control of

the protection in the manual protected mode (MP). The last solution is to let the automation controls the device in the automatic operation mode (A). The following state machine summarizes how to control the device. To simplify the notation, only “Open”, “Close”, “Stop”, “Stopped”, “Opening” and “Closing” are shown. For full description refer to the corresponding action.

The other actions provide the tools needed to handle the state variables (see Description of each action for full information).



3 XML Service Description

```

<?xml version="1.0"?>
<scpd xmlns="urn:schemas-UPnP-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>Open</name>
      <argumentList>
        <argument>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>Close</name>
      <argumentList>
        <argument>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>Stop</name>
      <argumentList>
        <argument>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetOperationMode</name>
      <argumentList>
        <argument>
          <name>RetOperationMode</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>OperationMode</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetOperationMode</name>
      <argumentList>
        <argument>
          <name>NewOperationMode</name>
          <direction>in</direction>
          <relatedStateVariable>OperationMode</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>IsLocked</name>
      <argumentList>
        <argument>
          <name>RetLocking</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>ServiceLocked</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>Lock</name>
      <argumentList>
        <argument>
        </argument>
      </argumentList>
    </action>
  </actionList>

```

```

</action>
<action>
<name>UnLock</name>
  <argumentList>
    <argument>
      </argument>
    </argumentList>
  </action>
<action>
<name>GetPosition</name>
  <argumentList>
    <argument>
      <name>RetPosition</name>
      <direction>out</direction>
      <retval />
      <relatedStateVariable>Position</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
<name>SetPosition</name>
  <argumentList>
    <argument>
      <name>NewPosition</name>
      <direction>in</direction>
      <relatedStateVariable>Position</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
<name>GetPositionArgType</name>
  <argumentList>
    <argument>
      <name>RetArgType</name>
      <direction>out</direction>
      <retval />
      <relatedStateVariable>PositionArgType</relatedStateVariable>
    </argument>
  </argumentList>
</action>
  Declarations for other actions added by UPnP vendor (if any) go here
</actionList>
<serviceStateTable>
  <stateVariable sendEvents="yes">
    <name>OperationMode</name>
    <dataType>String</dataType>
    <defaultValue>Vendor defined</defaultValue>
    <allowedValueList>
      <allowedValue>Manual Unprotected</allowedValue>
      <allowedValue>Manual Protected</allowedValue>
      <allowedValue>Automatic</allowedValue>
    </allowedValueList>
  </stateVariable>
  <stateVariable sendEvents="yes">
    <name>ServiceLocked</name>
    <dataType>Boolean</dataType>
    <defaultValue>1</defaultValue>
    <allowedValueList>
      <allowedValue>1</allowedValue>
      <allowedValue>0</allowedValue>
    </allowedValueList>
  </stateVariable>
  <stateVariable sendEvents="yes">
    <name>Position</name>
    <dataType>I1</dataType>
    <defaultValue>Vendor defined</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>100</maximum>
    </allowedValueRange>
  </stateVariable>
</serviceStateTable>

```

```

    <step></step>
  </allowedValueRange>
</stateVariable>
<stateVariable sendEvents="no">
  <name>PositionArgType</name>
  <dataType>String</dataType>
  <defaultValue>Vendor defined</defaultValue>
  <allowedValueList>
    <allowedValue>End Limits</allowedValue>
    <allowedValue>Continuous</allowedValue>
  </allowedValueList>
</stateVariable>
Declarations for other state variables added by UPnP vendor (if any)
go here
</serviceStateTable>
</scpd>

```


4 Test

Syntactical testing is performed by the UPnP test tool based on the XML description as provided in Clause 3.

The working committee and the implementers have come to the conclusion that further test descriptions e.g. for semantical testing do not provide a higher level of interoperability.

Thus the XML description is deemed to be sufficient for testing of devices that implement this template and further test descriptions are not provided by this template.

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch