

Homematic IP Legacy API (XML-RPC-Schnittstelle)



Homematic IP Legacy API (XML-RPC-Schnittstelle) Addendum

Spezifikation

Inhaltsverzeichnis

Inhaltsverzeichnis	3
1 Allgemeines	4
2 Methoden der Schnittstellenprozesse	5
2.1 Methodenübersicht.....	5
2.2 Erweiterungen Datentypen	6
2.2.1 <i>64-Bit Integer</i>	6
2.2.2 <i>DeviceDescription</i>	6
2.3 Neue/geänderte Methoden.....	7
2.3.1 <i>updateFirmware</i>	7
2.3.2 <i>installFirmware</i>	7
2.3.3 <i>setInstallModeWithWhitelist</i>	7
2.3.4 <i>deleteDevice</i>	8
2.3.5 <i>setValue</i>	8
2.3.6 <i>putParamset</i>	8
2.3.7 <i>getParamsetDescription</i>	9
2.3.8 <i>getParamset</i>	9

1 Allgemeines

Dieses Dokument beschreibt die Änderungen bzw. Erweiterungen der Legacy API (XML-RPC Schnittstelle) des Cloud Ready RFD (crRFD).

Soweit wie möglich wurde darauf geachtet, dass die Schnittstelle kompatibel zur existierenden HomeMatic (RFD) Implementierung ist.

2 Methoden der Schnittstellenprozesse

In der folgenden Liste mit [optional] gekennzeichneten Methoden müssen von einem konkreten Schnittstellenprozess nicht unbedingt exportiert werden. Bei Verwendung dieser Methoden ist mit einem Fehler zu rechnen, wenn nicht vor dem Aufruf mit `system.methodHelp` oder `system.listMethods` die Existenz geprüft wird.

2.1 Methodenübersicht

<i>Methode</i>	<i>HomeMatic-RF (Port: 2001)</i>	<i>Homematic IP (Port: 2010)</i>
activateLinkParamset	X	-
addDevice	X	-
addLink	X	X
Changekey	X	-
clearConfigCache	X	-
deleteDevice	X	X*
determineParameter	X	-
getDeviceDescription	X	X
getInstallMode	X	X
getKeyMismatchDevice	X	-
getLinkInfo	X	X
getLinkPeers	X	-
getLinks	X	X
getParamset	X	X*
getParamsetDescription	X	X*
getParamsetId	X	X
getValue	X	X*
init	X	X
listDevices	X	X
listTeams	X	_**
logLevel	X	-
putParamset	X	X*
removeLink	X	X
reportValueUsage	X	X
restoreConfigToDevice	X	-
rssInfo	X	-
searchDevices	-	-
setInstallMode	X	X
setLinkInfo	X	X
setTeam	X	_**
setTempKey	X	-
setValue	X	X*
system.listMethods	X	X
system.methodHelp	X	X
system.multicall	X	X
updateFirmware	X	-
listBidcosInterfaces	X	X

Homematic IP Legacy API (XML-RPC-Schnittstelle)

setBidcosInterface	X	-
getServiceMessages	X	-(X***)
getMetadata	X	-
setMetadata	X	-
getAllMetadata	X	-
abortDeleteDevice	X	-
hasVolatileMetadata	X	-
setVolatileMetadata	X	-
getVolatileMetadata	X	-
deleteVolatileMetadata	X	-
getVersion	X	X
setInterfaceClock	X	-
replaceDevice	X	-
listReplaceableDevices	X	-
ping	X	X
refreshDeployedDeviceFirmwareList	X	X
installFirmware	-	X
setInstallModeWithWhitelist	-	X

Legende:

* die Parameterwerte entsprechen nicht komplett der alten Spezifikation (z.B. Flags werden nicht unterstützt oder bestimmte Werteausprägungen sind geändert worden)

** Rauchmeldergruppen in Homematic IP werden durch ein Gruppenkonzept im Backend realisiert

*** Liste der Servicemeldungen muss noch definiert werden

2.2 Erweiterungen Datentypen

2.2.1 64-Bit Integer

Der XML-RPC Standard erlaubt lediglich vorzeichenbehaftete 32-Bit Integer. Der crRFD verwendet jedoch auch vorzeichenbehaftete Integer mit 64 Bit. Diese werden in der "DeviceDescription" mit dem Typ "Long" gekennzeichnet. Um die 64-Bit Integer über XML-RPC verlustfrei übertragen zu können, werden diese grundsätzlich in einen "String" umgewandelt (ausgehend) als solcher über XML-RPC übertragen. Umgekehrt werden eingehende Daten für den Typ "Long" als String erwartet und von der LegacyAPI in einen 64-Bit Integer konvertiert.

2.2.2 DeviceDescription

Folgende Members sind für Homematic IP hinzugekommen:

- SUBTYPE
Datentype String. Subtyp des Gerätes.
Beispiel: TYPE=SWD, SUBTYPE=SWDO (optischer Tür-/Fensterkontakt)
- firmwareUpdateState
Datentype String. Status des Background OTAU. Mögliche Werte:
 - UP_TO_DATE: Es liegen keine Änderungen vor.
 - NEW_FIRMWARE_AVAILABLE: Eine neue Firmware Version liegt vor.

- DELIVER_FIRMWARE_IMAGE: Das Ausliefern der Firmware Datei läuft.
- READY_FOR_UPDATE: Die Firmware Datei wurde übertragen und mit `installFirmware (String device)` kann der Updatevorgang gestartet werden.
- PERFORMING_UPDATE: Das Gerät führt den Update Vorgang aus, führt einen Werksreset durch und meldet sich dann mit einem Inclusion Request wieder und wird automatisch inkludiert.

2.3 Neue/geänderte Methoden

2.3.1 updateFirmware

```
bool updateFirmware(String device)
```

Starte beim alten HomeMatic Service (rfd) ein Live Update Vorgang (Live-OTAU). Wird von HM/IP nicht unterstützt.

2.3.2 installFirmware

```
bool installFirmware(String device)
```

Mit dieser Methode teilt die Logikschicht dem Schnittstellenprozess mit, dass ein Gerät die über Background OTAU aufgespielte Firmware aktivieren soll.

2.3.3 setInstallModeWithWhitelist

```
void setInstallModeWithWhitelist(Boolean on, Integer time, RpcStruct[] whitelistValues)
```

Mit dieser Methode teilt die Logikschicht dem Schnittstellenprozess mit, dass ein oder mehrere Geräte über die Whitelist für die Inkludierung freigeschaltet werden sollen für den angegebenen Zeitraum.

- `on`, gibt an ob die Install Mode aktiviert oder deaktiviert werden soll. Wenn der Wert „false“ ist haben die weiteren Parameter keine Auswirkung auf die Aktion.
- `time`, gibt die Zeit in Sekunden an, nach der der Install Mode deaktiviert werden soll. Der Defaultwert ist 30 Sekunden.
- `whiteListValues`, gibt die Geräte an, die inkludiert werden sollen, dies ist ein Array aus RPC Structs, welche folgende Daten haben:
 - ADDRESS, gibt die SGTIN bzw. ID des Gerätes an, das inkludiert werden soll.
 - KEY_MODE, gibt den Typen des Keys an, der für die Inklusion verwendet werden soll. Dieser Wert ist ein String mit folgenden unterstützten Werten:
 - LOCAL
 - MASTER und DEFAULT werden zur Zeit vom crRFD noch nicht unterstützt
 - KEY, gibt den zu verwendenden Geräteschlüssel an, der bei der Inklusion

Homematic IP Legacy API (XML-RPC-Schnittstelle)

verwendet werden soll. Dieser Wert ist der Hexstring des 16 Byte langen Schlüssels

Wenn der KEY_MODE oder KEY nicht angegeben ist wird der Master Key vom Key-Server für die Inklusion versucht zu nutzen.

Die Whitelist wird beim Ablauf des Install Mode zurückgesetzt.

2.3.4 deleteDevice

```
void deleteDevice(String address, Integer flags)
```

Diese Methode löscht ein Gerät aus dem Schnittstellenprozess.

Der Parameter `address` ist die Adresse des zu löschenden Gerätes.

`flags` ist ein bitweises oder folgender Werte:

- `0x02=DELETE_FLAG_FORCE`
Das Gerät wird auch gelöscht, wenn es nicht erreichbar ist

Bei Homematic IP sind die Flags `0x01` und `0x04` nicht mehr vorhanden und das Löschen von Geräten ist per Default so, als wenn diese Flags gesetzt worden wären.

2.3.5 setValue

```
void setValue(String address, String value_key, ValueType value)
```

Mit dieser Methode wird ein einzelner Wert aus dem Parameter-Set „VALUES“ geschrieben. Der Parameter `address` ist die Adresse eines logischen Gerätes. Der Parameter `value_key` ist der Name des zu schreibenden Wertes. Die möglichen Werte für `value_key` ergeben sich aus der `ParamsetDescription` des entsprechenden Parameter-Sets „VALUES“. Der Parameter `value` ist der zu schreibende Wert.

Bei HomeMatic IP kann `address` auch die spezielle Adresse „ALL_SMOKE_DETECTORS“ sein, welche für die Übertragung über Funk nicht eine Geräteadresse sondern die Multicast Adresse (`0xF00005`) für alle Rauchmelder verwendet und somit auch alle in Funkreichweite befindlichen Rauchmelder des Funknetzwerkes diesen ausführen sollten.

2.3.6 putParamset

```
void putParamset(String address, String paramset_key, Paramset set)
```

Mit dieser Methode wird ein komplettes Parameter-Set für ein logisches Gerät geschrieben. Der Parameter `address` ist die Adresses eines logischen Gerätes. Der Parameter `paramset_key` ist „MASTER“, „VALUES“ oder die Adresse eines Kommunikationspartners für das entsprechende Link-Parameter-Set (siehe `getLinkPeers`).

Der Parameter `set` ist das zu schreibende Parameter-Set. In `set` nicht vorhandene Member werden einfach nicht geschrieben und behalten ihren alten Wert.

Bei HomeMatic IP kann `address` für das Parameter Set „MASTER“ auch die spezielle Adresse „ALL_SMOKE_DETECTORS“ sein, welche für die Übertragung über Funk nicht

eine Geräteadresse sondern die Multicast Adresse (0xF00005) für alle Rauchmelder verwendet und somit auch alle in Funkreichweite befindlichen Rauchmelder des Funknetzwerkes diesen ausführen sollten.

2.3.7 getParamsetDescription

```
ParamsetDescription getParamsetDescription(String address, String paramset_type)
```

Mit dieser Methode wird die Beschreibung eines Parameter-Sets ermittelt. Der Parameter `address` ist die Adresse eines logischen Gerätes (z.B. von `listDevices` zurückgegeben). Der Parameter `paramset_type` ist „MASTER“, „VALUES“ oder „LINK“.

Bei HomeMatic IP gibt es für bestimmte Geräte zusätzlich den `paramset_type` „SERVICE“, welche Geräte dies sind, ist der jeweiligen `DeviceDescription` zu entnehmen. Ebenso ist dieses Parameter Set kanalübergreifend, so dass es über die XML-RPC-Schnittstelle für alle Kanäle abgefragt werden kann, aber immer dieselben Parameter zurückliefert.

2.3.8 getParamset

```
Paramset getParamset(String address, String paramset_key)
```

Mit dieser Methode wird ein komplettes Parameter-Set für ein logisches Gerät gelesen. Der Parameter `address` ist die Adresse eines logischen Gerätes. Der Parameter `paramset_key` ist „MASTER“, „VALUES“ oder die Adresse eines Kommunikationspartners für das entsprechende Link-Parameter-Set (siehe `getLinkPeers`).

Bei HomeMatic IP gibt es für bestimmte Geräte zusätzlich den `paramset_type` „SERVICE“, welche Geräte dies sind, ist der jeweiligen `DeviceDescription` zu entnehmen. Ebenso ist dieses Parameter Set kanalübergreifend, so dass es über die XML-RPC-Schnittstelle für alle Kanäle abgefragt werden kann, aber immer dieselben Parameter zurückliefert.

Hinweis: Diese Parameter werden bei jeder Anfrage direkt vom Gerät abgefragt, deshalb sollten sie wegen der DutyCycle Belastung nicht sehr häufig abgefragt werden.

Ab der Version 1.5 gibt es zusätzlich für die Geräte mit Router Funktionalität, wenn diese aktiviert ist (Konfigurations-Parameter „ROUTER_MODULE_ENABLED“ mit Wert „True“ bzw. „true“), den `paramset_type` „ROUTING_TABLE“, welche Geräte dies sind, ist der jeweiligen `DeviceDescription` zu entnehmen. Ebenso ist dieses Parameter Set kanalübergreifend, so dass es über die XML-RPC-Schnittstelle für alle Kanäle abgefragt werden kann, aber immer dieselben Parameter zurückliefert.

Hinweis: Diese Parameter werden bei jeder Anfrage direkt vom Gerät abgefragt, deshalb sollten sie wegen der DutyCycle Belastung nicht sehr häufig abgefragt werden. Da diese Routing Tabelle in der Anzahl der Werte dynamisch ist gibt es hierbei keine festgelegte Anzahl an Parametern, welches auch zu unterschiedlich langen Zeiten bei der Abfrage führen kann.

Die Antwort enthält folgende Parameter, welche jeweils bis zu 400mal auftreten können, aber

Homematic IP Legacy API (XML-RPC-Schnittstelle)

pro fortlaufender Nummer zusammen gehören.

Homematic IP Legacy API (XML-RPC-Schnittstelle)

Key	Type	Beschreibung
DYNAMIC_ROUTE_IP_DESTINATION_ADDRESS_	HEX-String	Adresse des Gerätes zu dem geroutet werden soll. Die Adresse hat entweder 24bit, 32bit oder 128bit je nach Adressfamilie.
DYNAMIC_ROUTE_NEXT_HOP_ADDRESS_	HEX-String	Die Adresse des nächsten Routers, welcher für die Erreichung der Zieladresse angesprochen werden muss. Die Adresse hat entweder 24bit, 32bit oder 128bit je nach Adressfamilie.
DYNAMIC_ROUTE_OM_ACCESS_CONTROLLER_	Boolean	Gibt an ob das Zielgerät ein Access Controller ist.
DYNAMIC_ROUTE_OM_ROUTER_	Boolean	Gibt an ob das Zielgerät ein Router ist.
DYNAMIC_ROUTE_OM_PORTABLE_DEVICE_	Boolean	Gibt an ob das Zielgerät ein tragbares Gerät ist, welches dadurch keinen festen Standort haben muss.
DYNAMIC_ROUTE_OM_LISTENER_MODE_	Enumeration	Gibt an über welchen Empfangsmodus das Zielgerät zu erreichen ist.
DYNAMIC_ROUTE_OM_VALID_	Boolean	Gibt an ob die Informationen, deren Key mit „DYNAMIC_ROUTE_OM_“ startet, verifiziert und gültig sind.
DYNAMIC_ROUTE_DISTANCE_	Integer	Gibt die Anzahl der „Hops“ an, welche zum Erreichen des Zielgerätes notwendig sind. Der Wertebereich ist 0 bis 7.
DYNAMIC_ROUTE_IS_STATIC_ROUTE_	Boolean	Gibt an ob die Routing Informationen aus seiner statischen Konfiguration stammt oder nicht.
DYNAMIC_ROUTE_IS_NEIGHBOUR_	Boolean	Gibt an ob das Zielgerät von dem Router direkt erreichbar ist.
DYNAMIC_ROUTE_IS_MAC_SEQUENCE_NUMBER_VALID_	Boolean	Gibt an ob der Router eine gültige „MAC Sequence Number“ des Zielgerätes hat.
DYNAMIC_ROUTE_RSSI_	Integer	Optional, Leistungspegel der Funkstrecke zu dem Zielgerät in dBm. Wird nur übergeben, wenn bekannt.

Tabelle 1: Parameter der Routing Tabellen Abfrage

Homematic IP Legacy API (XML-RPC-Schnittstelle)

Neue Fehlercodes

Fehlercode	Bedeutung
-10	Transmission (Übertragung an ein Gerät) ausstehend

