Remote Control Protocol
Connection Handling

AARONIA AG
WWW.AARONIA.DE

Version 1.1

Introduction

The Aaronia Remote Control Protocol (short ARCP) defines the general communication protocol of Aaronia devices connected over longer distances like Ethernet or RS232. Please check the product documentation of your device if it supports the ARCP specification in general, and what version it conforms to in detail.

This documents describes the basic parameters, network discovery and general connection handling that is common to all ARCP compatible devices. Please note that devices may conform to later versions of this specification that aren't necessarily compatible.

Architecture

The ARCP has four basic components:

- the network discovery protocol to identify devices on the local network (does not apply to RS232 devices)
- the basic communication channel
- the device-specific control syntax
- the device-specific configuration protocol

This document will address the first two aspects, while the device specific aspects are specified separately. Please refer to your device documentation for details.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

Network Discovery

ARCP devices obtain automatically an IP address using DHCP, or if no DHCP server is present in the connected network using an IPv4LL address.

The userspace application can detect this by sending a UDP broadcast packet on port 44014 with the payload

```
"Aaronia Discovery ALL"
```

or

```
"Aaronia Discovery $class"
```

where $class is a placeholder for a specific class of devices at the connected subnet and wait for replies on the same port. ARCP devices receiving this will reply with the following data:

```
"Aaronia Device $class;$model;$serial;$macaddress;$label"
```

where `$class`, `$model`, `$serial` and `$macaddress` are placeholders for the respective model, serial and MAC address strings, and `$label` is an user-configurable string. The userspace application can determine the IP address of the device by examining the origin address of the received UDP packet.

Note that the number of fields is not strictly fixed and may differ based on the ARCP version implemented on the device. The order however is guaranteed, e.g. `$label`, if present, will always be the fifth field.

As of this version the following device classes are specified:

- ISOLOG: references IsoLOG 3D Array antennas that were produced before ARCP was implemented and therefore do NOT conform to this specification outside of network discovery.
- ISOLOG_2: references IsoLOG 3D Array antennas that implement ARCP v1.0

Please check the device specific documentation about the class identifier used.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

Communication Parameters / Connection Handling

The actual ARCP control procotol is using regular HTTP/1.1[1] and/or HTTPS[2] requests on port 80 or 443 respectively. Devices using HTTPS can optionally require authentication for some or all requests, this is implemented using HTTP Basic Authentication.

Following the REST concept, in general all requests not changing the device state will use the GET method while requests that actually alter the device state are going to use POST requests and will deliver the changed device state as reply body when possible.

For integrity purposes most devices will only allow a single client to be connected simultaneously at any time. A connection is established by any successful GET or POST request, and is terminated after a specified timeout has passed without requests or a special disconnect request was sent. Clients are identified by a HTTP `Session-Key` header that is assigned by the device in the first reply, and which has to be included in any further requests by the client.

Devices will include a HTTP `Session-Timeout` header with a timeout value in seconds that defines how long the current connection will be kept alive. Client applications that wish to maintain an exclusive connection need to make another request within this time. If the application just wants to keep the connection open without actually changing the device state it should perform a GET request of the „state“ endpoint. Please note that due to network latencies the actual timeframe available might be somewhat shorter than listed in the header.

While a connection is established and another client issues a request the antenna will reply with a HTTP 423 „Locked“ status including a „Retry-After“ header stating when the lock will be lifted (assuming the currently connected client doesn't extend it).

Note: If the antenna is configured to require authentication the connection is only established when valid authentication data is provided, a failed authentication attempt will not prevent other clients from connecting.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

Authentication

ARCP devices *may* be configured to require authentication information to be provided in HTTPS requests. Authentication data is passed using the standard HTTP basic auth mechanism as specified in RFC 7617[3].

Note: Unencrypted HTTP requests are not protected by authentication as the authentication mechanism itself is unencrypted and relies on HTTPS for security.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

HTTP Codes

The following return codes may be sent by ARCP devices and should be handled by the client application:

200 – OK

Default if the request was accepted and processed without problems.

202 – Accepted

The request was accepted but not completely processed yet (firmware update request for example).

The reply body may contain further information.

400 – Bad request

The request was invalid (syntax error, unknown or unsupported values, …)

The reply body may contain further information.

401 – Unauthorized

The device requires authentication and no authentication headers were included in the request.

403 – Forbidden

The device requires authentication and the provided authentication headers were invalid

404 – Not found

The requested endpoint does not exist.

405 – Method not allowed

The specified method is not supported by the requested endpoint.

423 – Locked

Another client is currently connected to the device and only a single connection is allowed.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

The reply may contain a „Retry-After“ header specifying when the current connection timeout expires, and the reply body may contain further information about the connected client.

429 – Too many requests

The client sends requests faster than the device can handle them.

500 – Internal server error

An unspecified error occured. The reply body may contain further information.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

Control Protocol

The actual control protocol uses simple HTTP based plaintext commands. While the details are device specific, they share a common basis that will be listed here and is extended by the individual device specification documents.

Basic workflow of client applications should be to request the „info“ endpoint at startup, cache the configuration information that is returned and use it as reference for further operations. Regular operations should avoid the „info“ endpoint however due to the XML processing overhead and the amount of static data contained in replies. Instead the „state“ endpoint should be used for get/set requests of device state.

Endpoints

All ARCP devices will have at least two endpoints „info“ and „state“. The first will return a complete description of the device configuration and state as XML document, the second is changing the device state and accessing the regulary used state information in an easy to parse format.

Info

Supported methods: GET

Description:

Returns a XML description of the device configuration and its current status. Basic structure of the XML document is as follows (individual device specifications will define additional fields):

<device>

parent: root element
attributes: none
text content: none
child elements: info (1), versions (1), configuration (1), status (1)

<info>

parent: device
attributes: none
text content: none
child elements: model (1), serial (1)

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

<versions>

parent: device
attributes: none
text content: none
child elements: firmware (1)

<configuration>

parent: device
attributes: none
text content: none
child elements: settings (1)

<status>

parent: device
attributes: none
text content: none
child elements: value (0-N)

<model>

parent: info
attributes: none
text content: model description string
child elements: none

<serial>

parent: info
attributes: none
text content: serial number string (usually 5 decimal digits)
child elements: none

<firmware>

parent: versions
attributes: none
text content: version string (typically YYYYMMDD)
child elements: none

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

<settings>

parent: configuration
attributes: none
text content: none
child elements: value (0-N)

<value>

parent: settings
attributes:

- id: numeric value that is unique inside the parent element to reference this setting. If this attribute is missing the element should be ignored by applications.
- var: name of the variable referenced by this value, this determines the semantics of this setting (defined in device specification).
- hide (optional): „1“ if the variable will not be included in replies to GET requests of the „state“ endpoint (it is still available at the „info“ endpoint or in a reply when the value was changed by a „state“ POST request)
- type: one of „int“, „bool“, „text“ or „select“
- min (optional): only for type=int, specifies the minium possible value
- max (optional): only for type=int, specifies the maximum possible value
- step (optional): only for type=int, specifies the stepping interval of the value
- values: only for type=select, semicolon-separated list of available values
- readonly (optional): „1“ for settings that cannot be changed directly

text content: user-visible description what this setting controls
child elements: none

<value>

parent: status
attributes:

- id: value of the id attribute of the corresponding configuration <value>

text content: numeric value
child elements: none

state

Supported methods: GET, POST

Description: In POST mode send a change request for one or more variables. In both GET and POST modes returns a list of the current values for all default variables.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

The format is a simple newline-separated list of key=value assignments, where key is built from the corresponding tag-name (e.g. „value“) and id attribute value (as listed in the xml configuration description) separated by underscore. For example <value id=“2“>1</value> becomes value_2=1

In POST mode, the Content-Type of the request body should be set to „text/plain; charset=US-ASCII“.

control

Supported methods: POST

Description: This endpoint is for special commands that can't be handled by variables. The following commands (contained in the request body) are currently specified:

- reboot: triggers a reboot of the device
- logout: releases the connection so other clients can connect without waiting for the timeout

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

Serial Connection Handling

When a ARCP device is using a serial connection directly instead of a TCP/IP connection obviously there are some differences.

Connection Parameters

ARCP serial connections will always use the following parameters:

- Baudrate: 115200
- Flow-Control: None
- Parity: None
- Data-Bits: 8
- Stop-Bits: 1

Authentication

Serial connections are only supported as point-to-point connections in a highly controlled environment, so authentication is omitted for them.

Request Mapping

A request uses the following format (without quotes):

```
$verb $endpoint '\n'
[$content '\n']
'\n'
```

where `$verb` is the access method ('GET' or 'POST'), `$endpoint` the endpoint address as specified and `$content` the actual request content in case of POST requests. Every request must end with two newlines. The device will reply in the same format as for regular HTTP requests as specified above.

Gewerbegebiet Aaronia AG, D-54597 Strickscheid
Telefon: +49(0)6556-93033 Fax: +49(0)6556-93034
www.aaronia.de E-Mail: mail@aaronia.de

1 https://www.ietf.org/rfc/rfc2616.txt
2 https://www.ietf.org/rfc/rfc2818.txt
3 https://www.ietf.org/rfc/rfc7617.txt