Application Note

Property Access for Device Activation

Version: 1.0

Date Released: February 08, 2016

Document Number: AY006AQA3-1

Ayla Networks

Referenced Documents

Ayla Networks does not supply all documents that are referenced in this document with the equipment. Ayla Networks reserves the right to decide which documents are supplied with products and services.

Contact Information

Ayla Networks TECHNICAL SUPPORT and SALES

Contact Technical Support: https://support.aylanetworks.com
or via email at support@aylanetworks.com

Contact Sales: https://www.aylanetworks.com/company/contact-us

Ayla Networks REGIONAL OFFICES

GREATER CHINA
Shenzhen
Room 310-311
City University of Hong Kong
Research Institute Building
No. 8 Yuexing 1st Road
High-Tech Industrial Park
Nanshan District
Shenzhen, China
Phone: 0755-86581520

HEADQUARTERS
Silicon Valley
4250 Burton Drive, Suite 100
Santa Clara, CA 95054
United States
Phone: +1 408 830 9844
Fax: +1 408 716 2621

JAPAN
Wise Next Shin
Yokohama, 2-5-14
Shnyokohama, Kohokuku
Yokohama-shi, Kanagawa-ken
Yokohoma, 222-0033 Japan

EUROPE
London
30 Great Guildford St
London SE1 0HS
United Kingdom

TAIWAN
Taipei
5F No. 250 Sec. 1
Neihu Road, Neihu District
Taipei 11493, Taiwan

For a Complete Contact List of Our Offices in the US, China, Europe, Taiwan, and Japan:
https://www.aylanetworks.com/company/contact-us

Table of Contents

Revision History

Revision	Date	Author	Change Description
1	January 29, 2016	P. Hunt/L. Boling	Initial document

1 Introduction

This document provides an overview of the methods available on Ayla-enabled Wi-Fi Modules for accessing MCU properties during device activation.

These methods allow an HTTP client to gain restricted access to the properties implemented by an MCU attached to an Ayla-enabled Wi-Fi module, for device activation and initial configuration. This method is only accessible when the module has disconnected from the Ayla Cloud Service.

This feature is also used for limited field-testing and diagnostics.

1.1 Audience

This document is written for Engineers and Program Managers.

1.2 Related Documentation

Ayla Module and MCU Interface Specification (AY006MSP3)

2 HTTP Access to Properties

When this feature is enabled, an HTTP client or browser can read all current property values supported by the MCU, by performing a GET on `properties.json`, and make changes to properties via a POST to the same URL. Property values are displayed in JSON.

These URLs are only available if the device is in AP mode, and not yet registered to a user. The feature is enabled using a configuration command from the MCU and is only available for up to ten minutes. It is not available in GPIO mode.

The MCU code must support the “get-next” mechanism for reading properties. This mechanism does not request property values by name, but uses a continuation token to iterate over all supported properties. Example code for this mechanism is included in the MCU demo code package provided by Ayla.

IMPORTANT: This feature is not supported on the USI MR-03A, Murata Type ABR-A or Azurewave AW- CU300A modules.

3 Enable the Feature

The MCU enables access to properties via HTTP using configuration tokens, as described in the *Ayla Module and MCU Interface Specification* (AY006MSP3). Access can only be enabled under the following conditions:

- The device must be in AP mode
- The device must not have yet been registered with a user account.

The MCU must specify a timeout, in seconds, specifying the length of time the properties are accessible via the Ayla Wi-Fi module. This timeout must be 600 seconds or less. Setting the timeout to 0 will disable the feature.

The following C code fragment demonstrates how code on the MCU can enable the feature, with a timeout specified in seconds:

```
void prop_access_enable(uint16_t timeout)
{
    uint16_t val;
    enum conf_token prop_access[] = {CT_server, CT_prop, CT_time_limit};

    val = htons(timeout);
    conf_write(1, prop_access, 3, ATLV_UINT, &val, sizeof(val));
}
```

After this call succeeds, the `properties.json` URL is available through the Ayla Wi-Fi module's HTTP server for `timeout` seconds. Using this URL is described in the next section.

The following C code fragment demonstrates how code on the MCU can determine whether the feature is enabled:

```
void prop_access_read_cb(void *buf, size_t len)
{
    struct ayla_tlv *tlv;
    u8 status;

    tlv = (struct ayla_tlv *)buf;
    if (tlv->len != sizeof(status)) {
        return;
    }
    memcpy(&status, tlv + 1, sizeof(status)); /* 0=disabled; 1=enabled */
}

void prop_access_read(void)
{
```

```
        enum conf_token prop_access[] = {CT_server, CT_prop};

        conf_read(1, prop_access, 2, prop_access_read_cb);
    }
```

In the `prop_access_read_cb()` routine, above, the contents of `status` will be 1 if the feature is enabled, or 0 is it is not.

4 Read Property Values

When this feature is enabled, property values are available to an HTTP client connected to the Ayla Wi-Fi module when the module is in AP mode. This access method is not available in LAN mode, or via the local Wi-Fi network when the module is connected to an AP.

Property values are returned in a JSON record when a GET is performed on the `properties.json` URL, as follows:

```
# curl -G http://192.168.0.1/properties.json
{"properties":[{"name":"Blue_LED","value":"1"},{"name":"Green_LED","value":
"0"}]}
```

All property values supported by the MCU are returned in response; there is no mechanism provided to get a property by name.

4.1 MCU Code Requirements

The MCU code must support the get-next method of reading property values for this call to work. This method is described in the Ayla Module and MCU Interface Specification (AY006MSP3), and example code implementing this mechanism is provided in the MCU demo code package provided by Ayla.

When using the get-next method, the module does not request property values from the MCU by name, but by using a continuation token.

A get-next request is a data operation initiated by the module, using opcode 0x08 (Request Next Property), and containing a continuation token passed in an ATLV_CONT record. On the first request from the module, the ATLV_CONT token is 0. The MCU should return the first property name and value in response. If there are more properties supported by the module, the MCU should also include an ATLV_CONT record containing a continuation token with a value the MCU can use to identify the next property that should be requested.

The module may then issue a subsequent request using the same continuation token; the MCU should return the name and value for the property corresponding to that token, and a continuation token for the next property. If no more properties exist, no continuation token should be included in the reply.

The `spi_rx_send_next_tlv()` and `serial_rx_send_next_tlv()` functions in the MCU demo code demonstrate how to handle a get-next request and continuation token. In this example, the MCU code maintains a table of property entries, and uses the table index as the continuation token.

5 Set Property Values

When this feature is enabled, property values are available to an HTTP client connected to the Ayla Wi-Fi module when the module is in AP mode. This access method is not available in LAN mode, or via the local Wi-Fi network when the module is connected to an AP.

An HTTP client sets property values by passing a JSON record in a POST operation to the properties.json URL, as follows:

```
# curl --data
'{"properties":[{"property":{"name":"Blue_LED","base_type":"boolean","value
":1}}, {"property":{"name":"Green_LED","base_type":"boolean","value":0}}]}'
http://192.168.0.1/properties.json
```

NOTE Only up to three properties may be set in a single command.

The command contains an array of property records, with each record containing the name, type and value of the property to set. The module will issue an individual set operation on each property in turn.

5.1 MCU Code Requirements

There are no special requirements on the MCU to support this feature. The set operation is same as an operation received from the Ayla service or from a LAN client during normal operation.

6 Security Considerations

This feature grants unauthenticated access to device properties, and is only intended for device testing during manufacturing, for limited access to properties during device activation, and for limited field service testing. The feature can only be enabled when the device is in AP mode, and when the device has not yet been registered with a user.

In addition to the security measures mentioned above, Ayla strongly recommends the code on the MCU be very restrictive in enabling the feature. It should not enable the feature automatically on device power-up, for example, as there may be situations in which the device reverts to AP mode if Wi-Fi access is temporarily unavailable.

Ayla recommends that some physical intervention – a push-button, for example – be required before the MCU code enables the feature.

4250 Burton Drive, Santa Clara, CA 95054
Phone: +1 408 830 9844
Fax: +1 408 716 2621