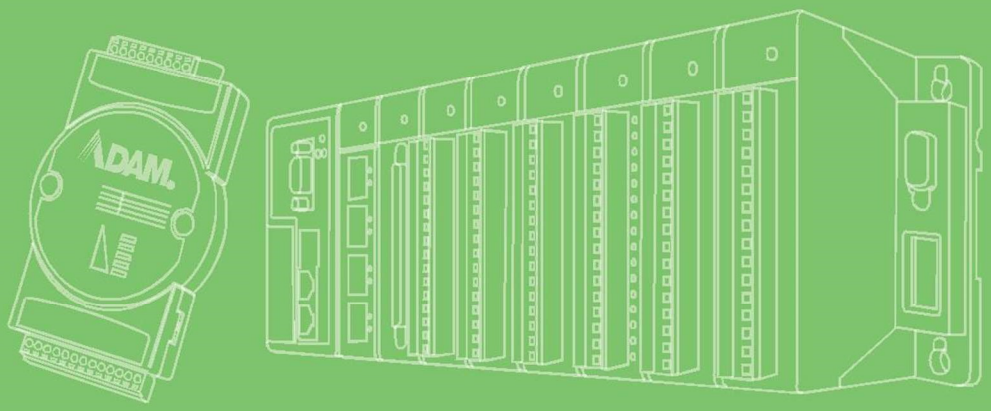


User Manual

ADAM-6700 Series Open RESTful API Specification

V1

CONTENTS

Document History:

Doc Version	Date	Comment
1	2023/04/06	First edition

Documentation Conventions

- List of abbreviations:

R	Read-only
W	Write-only
RW	Read or Write

1. Overview

This document describe the Advantech ADAM-6700 series open RESTful API specification. The ADAM-6700 device acts a REST server, and user use REST request commands to query the system information, configure the I/O parameters, update I/O firmware or image etc.

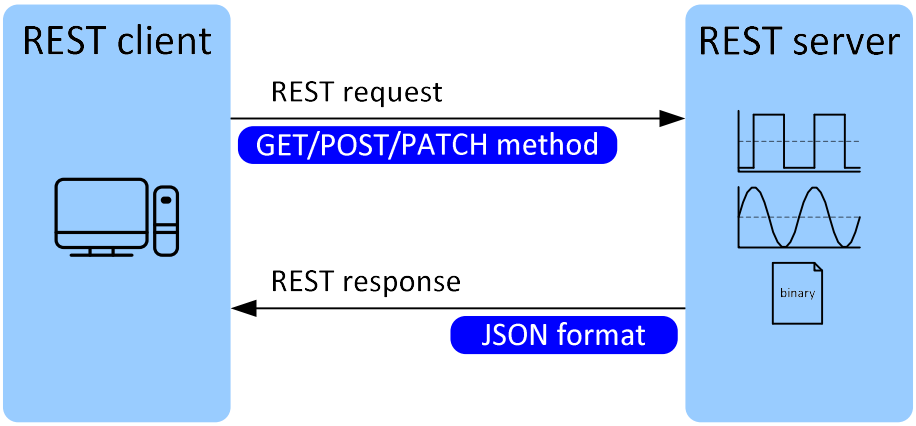


Figure 1

Please pay special attention! If you are updating the image via SD card, please insert the jumper as shown below (see **Figure 2**) and remember to remove the jumper after the update is complete(see **Figure 3**). If you are updating the image via REST command, please make sure that the jumper has been removed (see **Figure 3**). Otherwise, the update will fail.

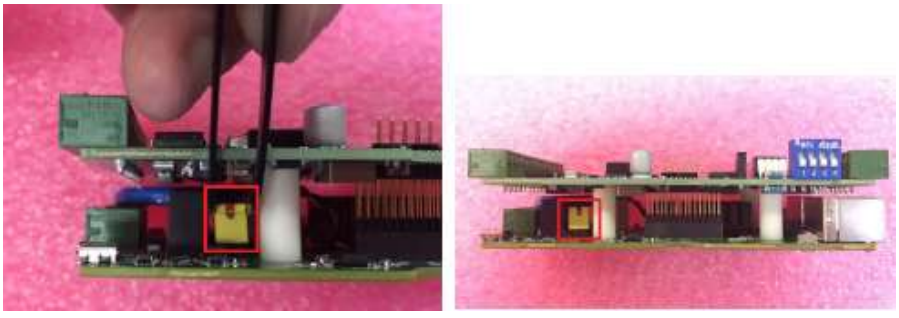


Figure 2

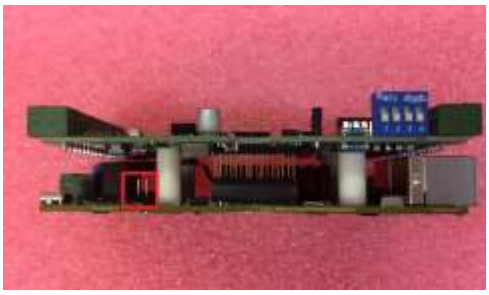


Figure 3

2. Introduction of REST commands

The ADAM-6700 series provides several types of REST commands, including system information, I/O control and reading, and software updates. The following will provide a detailed description of each type.

2.1. System Information

2.1.1. Device Information

/web/profile

Description	Retrieves the device information including the I/O module.
URL Structure	**https://10.0.0.1/web/profile**
HTTP Method	GET：Returns the representation of all of I/O module information resource.
GET	Request： **GET /web/profile** [Example]: Request : **GET /web/profile** Content-type: application/json Response: 200 OK { "Dev": [{ "SL": 0, "Id": "ADAM-6717", "DIn": 5, "DOn": 4, "RLAn": 0, "AIn": 8,

```
                "UIn": 0,
                "AOn": 0,
                "Cntn": 0,
                "FwVer": "A1.01 B019",
                "OSVer": "ADAM-6717 image 2.0.0 rev 51edbc7 Wed Mar  8 01:38:05 UTC
2023\n"
            }
        ]
}
```


JSON array name definition:

Field	Abbreviation	Data Type
Array of device information	Dev	Array

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Slot Number	SL	Number	R	0, 1 ~: Slot number. For ADAM-6700, this field is 0.
Model ID	Id	String	R	Model ID
Total DI Number	DIn	Number	R	0, 1 ~: Digital input channel number.
Total DO Number	DOn	Number	R	0, 1 ~: Digital output channel number.
Total Relay Number	RLAn	Number	R	0, 1 ~: Relay output channel number.
Total AI Number	AIn	number	R	0, 1 ~: Analog input channel number.
Total AO Number	AOn	Number	R	0, 1 ~: Analog output channel number.
Total Counter Number	Cntn	Number	R	0, 1 ~: Counter channel number.
Firmware Version	FwVer	String	R	Version information of main firmware "" : none
Image Version	OSVer	String	R	Version information of image "" : none

Remarks	

2.2. I/O Control

2.2.1. General Configuration

/web/gen_config

Description	Retrieves the general configuration.
URL Structure	**https://10.0.0.1/web/gen_config**
HTTP Method	GET：Returns the representation of all of general configuration resource. PATCH：Apply partial modifications to general configuration resource.

GET	Request：
	GET /web/gen_config
	
	[Example]:
	Request : **GET /web/gen_config**
	
	Content-type: application/json
	Response: 200 OK
	{
	"EnWDT": 0,
	"EnWDTTime": 0,
	"Acnt": 2,
	"WMd": 0,
	"MBTCPPort": 5020
	}

PATCH	Request：**PATCH /web/gen_config** [Example]: Request: **PATCH /web/gen_config** Content-type: application/json { "EnWDT ":0, "EnWDTTime": “500”

	}

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Enable the Communication WDT to Trigger FSV	EnWDT	Number	RW	1 or 0: Enable or Disable the Communication WDT to Trigger FSV
Watchdog Timer Setting	EnWDTTime	String	RW	1 ~ 9999: Sets communication WDT cycle time from 1 ~ 9999 (unit: 0.1 second.)
MODBUS TCP port number	MBTCPPort	Number	RW	Local port number for web server function Default: 5020

Remarks	

2.2.2. DO Configuration

/web/do_config/slot_0

Description	Retrieves information about the digital output configuration resource.
URL Structure	**https://10.0.0.1/web/do_config/slot_0**
HTTP Method	GET：Return the representation of all of digital output configurations resource. PATCH：Apply partial modifications to digital output configurations resource.
GET	Request： **GET /web/do_config/slot_0** [Example]: Request : **GET /web/do_config/slot_0** Content-type: application/json Response: 200 OK { "DOCfg": [{ "Ch": 0, "Tag": "DO", "Md": 0, "PsLo": 0,

```
            "PsHi": 0,
            "LDT": 0,
            "HDT": 0,
            "FSV": 0,
            "ACh": 0,
            "AMd": 0
        },
        {
            "Ch": 1,
            "Tag": "DO",
            "Md": 0,
            "PsLo": 0,
            "PsHi": 0,
            "LDT": 0,
            "HDT": 0,
            "FSV": 0,
            "ACh": 0,
            "AMd": 0
        },
        {
            "Ch": 2,
            "Tag": "DO",
            "Md": 0,
            "PsLo": 0,
            "PsHi": 0,
            "LDT": 0,
            "HDT": 0,
            "FSV": 0,
            "ACh": 0,
            "AMd": 0
        },
        {
            "Ch": 3,
            "Tag": "DO",
            "Md": 0,
            "PsLo": 0,
            "PsHi": 0,
```

<table>
<tr><td></td><td>"LDT": 0,
"HDT": 0,
"FSV": 0,
"ACh": 0,
"AMd": 0
}
]
}</td></tr>
<tr><td>PATCH</td><td>Request：
PATCH /web/do_config/slot_0

[Example]:
Request: PATCH /web/do_config/slot_0
Content-type: application/json
{
"DOCfg": [
{
"Ch":0,
"Md":1
},
{
"Ch":2,
"Md ": 0
}
]
}</td></tr>
</table>

JSON array name definition:

Field	Abbreviation	Data Type
Array of Digital output configurations	DOCfg	Array

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Channel Number	Ch	Number	R	0, 1 ~ : Digital output channel number.
Mode	Md	Number	RW	Digital output mode.

Value	Mode	Note

				0	DO
				1	Pulse Output
				2	LowToHighDelay
				3	HighToLowDelay
Low Signal Width	PsLo	Number	RW	Low signal width of pulse 1 - 4294967295 (0.1 ms)	
High Signal Width	PsHi	Number	RW	High signal width of pulse 1 - 4294967295 (0.1 ms).	
High To Low Delay Time	HDT	Number	RW	Time for High To Low Delay 1 - 4294967295 (0.1 ms)	
Low To High Delay Time	LDT	Number	RW	Time for Low To High Delay 1 - 4294967295 (0.1 ms).	
Tag Name	Tag	String	R	The description tag for this channel. Max. 21 characters	
Remarks					

2.2.3. DO Data Acquisition

/web/do_value/slot_0

Description	Retrieves information about the digital input value resource.
URL Structure	**https://10.0.0.1/web/do_value/slot_0**
HTTP Method	GET：Returns the representation of all of digital output value resource. PATCH：Apply partial modifications to digital output value resource.
GET	Request： **GET /web/do_value/slot_0** [Example]: Request : **GET /web/do_value/slot_0** Content-type: application/json Response: 200 OK { "DOVal": [

```
{
    "Ch": 0,
    "Tag": "DO",
    "Stat": 0,
    "PsStop": 0,
    "PsIV": 0,
    "Md": 0,
    "Val": 0,
    "PsCtn": 0
},
{
    "Ch": 1,
    "Tag": "DO",
    "Stat": 0,
    "PsStop": 0,
    "PsIV": 0,
    "Md": 1,
    "Val": 0,
    "PsCtn": 0
},
{
    "Ch": 2,
    "Tag": "DO",
    "Stat": 0,
    "PsStop": 0,
    "PsIV": 0,
    "Md": 2,
    "Val": 0,
    "PsCtn": 0
},
{
    "Ch": 3,
    "Tag": "DO",
    "Stat": 0,
    "PsStop": 0,
    "PsIV": 0,
    "Md": 3,
```

<table>
<tr><td></td><td>"Val": 0,
"PsCtn": 0
}
]
}</td></tr>
<tr><td>PATCH</td><td>Request：
PATCH /web/do_value/slot_0

[Example]:
Request: PATCH /web/do_value/slot_0
Content-type: application/json
{
"DOVal": [
{
"Ch":2,
" Val ": 1
},
{
"Ch":3,
"PsStop":1
}
]
}</td></tr>
</table>

JSON array name definition:

Field	Abbreviation	Data Type
Array of Digital input configurations	DOVal	Array

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Channel Number	Ch	Number	R	0, 1 ~ : Digital output channel number.
Mode	Md	Number	R	Digital output mode. 0 DO 1 Pulse Output 2 LowToHighDelay

<table>
<tr><td colspan="3"></td><td></td><td>3 | HighToLowDelay</td><td></td></tr>
<tr><td>Channel Value</td><td>Val</td><td>Number</td><td>RW</td><td>DO measurement data
<table><tr><th>Output Mode</th><th>Value Description</th></tr><tr><td>DO</td><td>Get the current signal status or set its status</td></tr><tr><td>Pulse Output</td><td>Get or set the absolute pulse count value</td></tr><tr><td>LowToHighDelay</td><td>Get the current signal status or set its status</td></tr><tr><td>HighToLowDelay</td><td>Get the current signal status or set its status</td></tr></table></td><td></td></tr>
<tr><td>Pulse Output Continue State</td><td>PsCtn</td><td>Number</td><td>RW</td><td>1 / 0: Pulse outputting is continuous or not.</td><td></td></tr>
<tr><td>Stop Pulse Output</td><td>PsStop</td><td>Number</td><td>W</td><td>1: Stop the pulse outputting.
(Continue is disabled, Absolute and incremental values are reset to zero. DO signal status is set to logic low.)</td><td></td></tr>
<tr><td>Remarks</td><td colspan="5"></td></tr>
</table>

2.2.4. DI Configuration

/web/di_config/slot_0

Description	Retrieves information about the digital input configuration resource.
URL Structure	**https://10.0.0.1/web/di_config/slot_0**
HTTP Method	GET：Return the representation of all of digital input configurations resource. PATCH：Apply partial modifications to digital input configurations resource.
GET	Request： **GET /web/di_config/slot_0** [Example]: Request : **GET /web/di_config/slot_0** Content-type: application/json

Response: 200 OK


```
{
    "DICfg": [
        {
            "Ch": 0,
            "Tag": "DI",
            "Md": 0,
            "CntKp": 0,
            "Fltr": 0,
            "Inv": 0,
            "FtLo": 1,
            "FtHi": 1,
            "CntIV": 0,
            "FqP": 0,
            "FqT": 0
        },
        {
            "Ch": 1,
            "Tag": "DI",
            "Md": 1,
            "CntKp": 0,
            "Fltr": 0,
            "Inv": 0,
            "FtLo": 1,
            "FtHi": 1,
            "CntIV": 0,
            "FqP": 0,
            "FqT": 0
        },
        {
            "Ch": 2,
            "Tag": "DI",
            "Md": 2,
            "CntKp": 0,
            "Fltr": 0,
            "Inv": 0,
            "FtLo": 5,
```

	"FtHi": 5,
	"CntIV": 0,
	"FqP": 0,
	"FqT": 0
	},
	{
	"Ch": 3,
	"Tag": "DI",
	"Md": 3,
	"CntKp": 0,
	"Fltr": 0,
	"Inv": 0,
	"FtLo": 5,
	"FtHi": 5,
	"CntIV": 0,
	"FqP": 0,
	"FqT": 0
	},
	{
	"Ch": 4,
	"Tag": "DI",
	"Md": 4,
	"CntKp": 0,
	"Fltr": 0,
	"Inv": 0,
	"FtLo": 5,
	"FtHi": 5,
	"CntIV": 0,
	"FqP": 0,
	"FqT": 0
	}
	]
	}

PATCH	Request： **PATCH /web/di_config/slot_0** [Example]:

Request: **PATCH /web/di_config/slot_0**

Content-type: application/json

```
{
"DICfg": [
      {
         "Ch":0,
         "Inv":0
      },
      {
         "Ch":2,
         "CntKp": 0
      },
      {
         "Ch":3,
         "Md":1,
         "Fltr": 1,
         "FtLo": 500,
         "FtHi": 500
      }
   ]
}
```

JSON array name definition:

Field	Abbreviation	Data Type
Array of Digital input configurations	DICfg	Array

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Channel Number	Ch	Number	R	0, 1 ~ : Digital input channel number.
Mode	Md	Number	RW	Digital input mode. 0 DI 1 Counter 2 LowToHighLatch 3 HighToLowLatch 4 Frequency
Invert Signal	Inv	Number	RW	1 or 0: Enable or Disable invert signal function.
Digital Filter	Fltr	Number	RW	1 or 0: Enable or Disable digital filter function

Min. Low Signal Width	FtLo	Number	RW	Minimum low signal width of digit filter 1 - 4294967295 (0.1 ms)
Min. High Signal Width	FtHi	Number	RW	Minimum high signal width of digit filter 1 - 4294967295 (0.1 ms).
Keep Counter Value When Poweroff	CntKp	Number	RW	1 or 0: Enable / Disable keep last value when power off.
Tag Name	Tag	String	R	The description tag for this channel. Max. 21 characters
Remarks				

2.2.5. DI Data Acquisition

/web/di_value/slot_0

Description	Retrieves information about the digital input value resource.
URL Structure	**https://10.0.0.1/web/di_value/slot_0**
HTTP Method	GET：Returns the representation of all of digital input value resource. PATCH：Apply partial modifications to digital input value resource.
GET	Request： **GET /web/di_value/slot_0** [Example]: Request : **GET /web/di_value/slot_0** Content-type: application/json Response: 200 OK { "DIVal": [{ "Ch": 0, "Tag": "DI", "Stat": 0, "Md": 0, "Cnting": 1, "Val": 0

	},
	{
	"Ch": 1,
	"Tag": "DI",
	"Stat": 0,
	"Md": 1,
	"Cnting": 1,
	"Val": 0
	},
	{
	"Ch": 2,
	"Tag": "DI",
	"Stat": 0,
	"Md": 2,
	"Cnting": 1,
	"OvLch": 0
	},
	{
	"Ch": 3,
	"Tag": "DI",
	"Stat": 0,
	"Md": 3,
	"Cnting": 1,
	"OvLch": 0
	},
	{
	"Ch": 4,
	"Tag": "DI",
	"Stat": 0,
	"Md": 4,
	"Cnting": 1,
	"Val": 0
	}
	]
	}

PATCH	Request： **PATCH /web/di_value/slot_0**

[Example]:

Request: **PATCH /web/di_value/slot_0**

Content-type: application/json

```
{
"DIVal": [
      {
         "Ch":2,
         "Cnting": 1
      },
      {
         "Ch":3,
         "OvLch":0
      }
   ]
}
```

JSON array name definition:

Field	Abbreviation	Data Type
Array of Digital input configurations	DIVal	Array

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Channel Number	Ch	Number	R	0, 1 ~ : Digital input channel number.
Mode	Md	Number	R	Digital input mode. 0 DI 1 Counter 2 LowToHighLatch 3 HighToLowLatch 4 Frequency
Channel Value	Val	Number	R	DI measurement data Input Mode: Value Description DI: Logic status of DI Counter: Counter value LowToHighLatch: Logic status of DI HighToLowLatch: Logic status of DI

				Frequency	Frequency(unit. 1 Hz)
Start Counter	Cnting	Number	RW	Start/Stop counter counting Read 1 : counter is counting 0 : not counting Write 1 : start counting 0 : stop counting	
Clear Counter	ClrCnt	Number	W	1 : Clear the counter value	
Get/Clear Counter Overflow or Latch Status	OvLch	Number	RW	counter overflow or latch status Read 1 : overflow/latch occurred. 0 : no overflow or latch Write 0 : clear the overflow or latch status	
Remarks					

2.2.6. AI Configuration

/web/ai_config/slot_0

Description	Retrieves information about the digital input configuration resource.
URL Structure	**https://10.0.0.1/web/ai_config/slot_0**
HTTP Method	GET：Return the representation of all of analog input configurations resource. PATCH：Apply partial modifications to analog input configurations resource.
GET	Request： **GET /web/ai_config/slot_0** [Example]: Request : **GET /web/ai_config/slot_0** Content-type: application/json Response: 200 OK { "AICfg": [{ "Ch": 0, "En": 1,

```
            "Rng": 7
        },
        {
            "Ch": 1,
            "En": 1,
            "Rng": 7
        },
        {
            "Ch": 2,
            "En": 1,
            "Rng": 7
        },
        {
            "Ch": 3,
            "En": 1,
            "Rng": 7
        },
        {
            "Ch": 4,
            "En": 1,
            "Rng": 7
        },
        {
            "Ch": 5,
            "En": 1,
            "Rng": 7
        },
        {
            "Ch": 6,
            "En": 1,
            "Rng": 7
        },
        {
            "Ch": 7,
            "En": 1,
            "Rng": 7
        }
```

<table>
<tr><td></td><td>]
}</td></tr>
<tr><td>PATCH</td><td>Request：
PATCH /web/ai_config/slot_0

[Example]:
Request: PATCH /web/ai_config/slot_0
Content-type: application/json
{
"AICfg": [
{
"Ch":0,
"Rng": "13"
},
{
"Ch":2,
" Rng ": "7"
}
]
}</td></tr>
</table>

JSON array name definition:

Field	Abbreviation	Data Type
Array of Analog input configurations	AICfg	Array

Resource value definitions by **Each Channel (Total channels = AI channel number + 1 average channel)** :

Field	Abbreviation	Data Type	Property	Description
Channel Number	Ch	Number	R	0, 1 ~ : Analog input channel number.
Channel Enable	En	Number	RW	1 / 0: Enable / Disable AI conversion
Input Range	Rng	Number	RW	Analog input range.

7	4~20 mA
8	+/- 10 V
9	+/- 5 V
10	+/- 1 V
11	+/- 500 mV
12	+/- 150 mV
13	+/- 20 mA

	72	0~10 V	
	73	0~5 V	
	74	0~1 V	
	75	0~500 mV	
	76	0~150 mV	
	77	0~20 mA	
Remarks			

2.2.7. AI Data Acquisition

/web/ai_value/slot_0

Description	Retrieves information about the analog input value resource.
URL Structure	**https://10.0.0.1/web/ai_value/slot_0**
HTTP Method	GET：Returns the representation of all of analog input value resource.
GET	Request： **GET /web/ai_value/slot_0** [Example]: Request : **GET /web/ai_value/slot_0** Content-type: application/json Response: 200 OK { "AIVal": [{ "Ch": 0, "En": 1, "Rng": 13, "EgF": 0.012000, "Evt": 0 }, { "Ch": 1,

```
            "En": 1,
            "Rng": 7,
            "EgF": 0,
            "Evt": 8
        },
        {
            "Ch": 2,
            "En": 1,
            "Rng": 7,
            "EgF": 0,
            "Evt": 8
        },
        {
            "Ch": 3,
            "En": 1,
            "Rng": 7,
            "EgF": 0,
            "Evt": 8
        },
        {
            "Ch": 4,
            "En": 1,
            "Rng": 7,
            "EgF": 0,
            "Evt": 8
        },
        {
            "Ch": 5,
            "En": 1,
            "Rng": 7,
            "EgF": 0,
            "Evt": 8
        },
        {
            "Ch": 6,
            "En": 1,
            "Rng": 7,
```

```
                "EgF": 0,
                "Evt": 8
            },
            {
                "Ch": 7,
                "En": 1,
                "Rng": 7,
                "EgF": 0,
                "Evt": 8
            }
        ]
}
```


JSON array name definition:

Field	Abbreviation	Data Type
Array of Analog input configurations	AIVal	Array

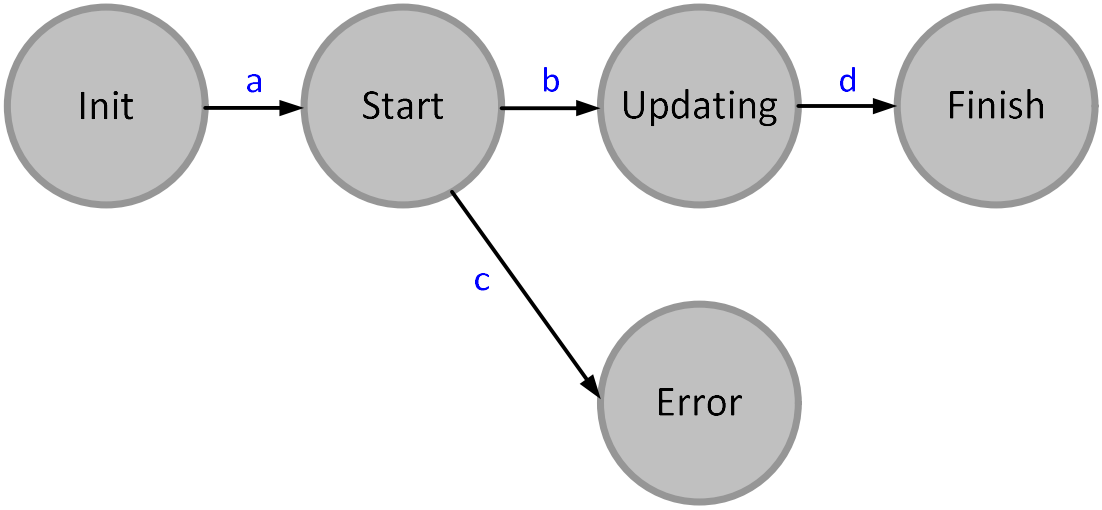
Resource value definitions (Total channels = AI channel number + 1 average channel):

Field	Abbreviation	Data Type	Property	Description
Channel Number	Ch	Number	R	0, 1 ~ : Analog input channel number.
Input Range	Rng	Number	R	Analog input range. 7: 4~20 mA 8: +/- 10 V 9: +/- 5 V 10: +/- 1 V 11: +/- 500 mV 12: +/- 150 mV 13: +/- 20 mA 72: 0~10 V 73: 0~5 V 74: 0~1 V 75: 0~500 mV 76: 0~150 mV 77: 0~20 mA
Channel Enable	En	Number	R	1 / 0: Enable / Disable AI conversion Notice: Average channel is read only. When channel mask of average is not 0, the value is 1.

Channel Status	Evt	Number	R	AI statuses 0 Normal 2 Over Range 4 Under Range 8 Burnout
Engineering data	EgF	Number	R	AI engineering data.
Remarks				

2.3. Software Update

ADAM-6700 provides remote software updates. Due to the consideration that transmission and processing times may be longer for larger files, we adopt an asynchronous approach. Please refer to the following state machine for details.



Update flow State	
State	Description
Init	The initial state of the device after power-on.
Start	In this state, it indicates that a POST request has been received and a

	response will be sent indicating that the command has been received. At the same time, the file will be asynchronously checked for correctness.
Updating	Update is in progress. It is necessary to reboot (except for custom files).
Finish	Update complete (If the device is rebooting, it may skip this state.)

Update flow translation	
Translation	Description
a	Received a request to update the software (POST content). The duration of this process depends on the size of the file.
b	File check completed, software update process starting.
c	The file check has failed. This may be due to a mismatch in the binary file module ID, or an incorrect compressed file.
d	The update action has been completed.

Update Error Code		
Code	Name	Description
0	SUCCESS	Action successful.
1	FILE_ERROR	Failed to access file. Usually occurs when reading and writing files fails.
2	INCORRECT_BINARY	Incorrect binary file. - Firmware download: Binary module ID mismatch - Image download: Compressed file is incomplete
3	INCORRECT_FILE_NAME	Incorrect binary file name. - Custom file: The naming rule is custom**N**.tar.gz, N is number , and the number starts from 1. For example: **custom1.tar.gz**

Due to the asynchronous nature of the update process, we provide a REST API for querying the current status and results. For details, please refer to the following sections.

2.3.1. Image Update

Please compress the image file into a **zip** format. If there are custom files that need to be updated at the same time, please compress them into the same zip file (you can also choose to update them separately, please refer to chapter 2.3.3). Please refer to Appendix 1 for the format and naming conventions of custom files.

/web/image_download

Description	Upload image zip file in a web application.
URL Structure	**https://10.0.0.1/web/image_download**
HTTP Method	POST：Transport an image zip file to the module.

GET	Request：
	GET /web/image_download
	
	[Example]:
	Request : **GET /web/image_download**
	
	Content-type: application/json
	Response: 200 OK
	{
	"StatusCode": 0,
	"LastErrorCode": 0,
	"Status": "Init",
	"Message": "The update has not started yet. No error."
	}
	
	Or
	
	Content-type: application/json
	Response: 200 OK
	{
	"StatusCode": 4,
	"LastErrorCode": 2,

<table>
<tr><td></td><td>"Status": "Error",
"Message": "Update fail. Incorrect binary."
}</td></tr>
<tr><td>POST</td><td>Request：
POST /web/image_download

[Example]:
Request: POST /web/image_download, upload the image zip file to the module.

Content-Type：multipart/form-data;
boundary=----WebKitFormBoundaryIcafShJTMskmwleL (generated by software randomization)
…
----WebKitFormBoundaryIcafShJTMskmwleL\r\n
Content-Disposition: form-data; name="filename"; filename=" adam6717_230331.zip"\r\n
Content-Type：application/zip \r\n\r\n
<Image data>\r\n
------WebKitFormBoundaryIcafShJTMskmwleL--\r\n

Response: 200 OK</td></tr>
</table>

[GET]

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Update status code	StatusCode	Number	R	Update status
Last error code	LastErrorCode	Number	R	Update flow last error code.

Update status

Status Code	Status Description
0	Init
1	Start
2	Finish
3	Updating
4	Error
99	Unknown

Update flow last error code.

Status Code	Status Description
0	SUCCESS
1	FILE_ERROR

				2 INCORRECT_BINARY 3 INCORRECT_FILE_NAME
Update status	Status	String	R	Update status Status Code: 0 Init; 1 Start; 2 Finish; 3 Updating; 4 Error; 99 Unknown
Update flow message	Message	String	R	Update flow message.

[POST]

Resource value definitions :

Parameters for ***Name = " xxx "***	Description	Example
filename	The file contents	------WebKitFormBoundaryIcafShJTMskmwleL Content-Disposition:form-data;name="filename"； Content-Type:application/octet-stream (text/html …) <…file data…>

2.3.2. Firmware Update

/web/firmware_download

Description	Upload firmware binary file in a web application.
URL Structure	**https://10.0.0.1/web/firmware_download**
HTTP Method	POST：Transport a firmware binary file to the module.
GET	Request： **GET /web/firmware_download**

	[Example]:
	Request : **GET /web/firmware_download**
	
	Content-type: application/json
	Response: 200 OK
	{
	"StatusCode": 0,
	"LastErrorCode": 0,
	"Status": "Init",
	"Message": "The update has not started yet. No error."
	}
	
	Or
	
	Content-type: application/json
	Response: 200 OK
	{
	"StatusCode": 4,
	"LastErrorCode": 2,
	"Status": "Error",
	"Message": "Update fail. Incorrect binary."
	}

POST	Request： **POST /web/firmware_download** [Example]: Request: **POST /web/firmware_download**, upload the firmware binary to the module. Content-Type：multipart/form-data; boundary=----WebKitFormBoundaryIcafShJTMskmwleL (generated by software randomization) … ----WebKitFormBoundaryIcafShJTMskmwleL\r\n Content-Disposition: form-data; name="filename"; filename="ADAM-6717_A101B09_UT.bin"\r\n Content-Type：application/octet-stream \r\n\r\n <firmware binary data>\r\n

	------WebKitFormBoundaryIcafShJTMskmwleL--\r\n Response: 200 OK

[GET]

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Update status code	StatusCode	Number	R	Update status Status Code: 0 Init; 1 Start; 2 Finish; 3 Updating; 4 Error; 99 Unknown
Last error code	LastErrorCode	Number	R	Update flow last error code. Status Code: 0 SUCCESS; 1 FILE_ERROR; 2 INCORRECT_BINARY; 3 INCORRECT_FILE_NAME
Update status	Status	String	R	Update status Status Code: 0 Init; 1 Start; 2 Finish; 3 Updating; 4 Error; 99 Unknown
Update flow message	Message	String	R	Update flow message.

[POST]

Resource value definitions :

Parameters for	Description	Example

Name = " xxx "		
filename	The file contents	------WebKitFormBoundaryIcafShJTMskmwleL Content-Disposition:form-data;name="filename" ; Content-Type:application/octet-stream (text/html …) <…file data…>

2.3.3. Custom Software Update

This section refers to updating a custom file separately. If you want to install the custom file during the Image installation process, please refer to section 2.3.1.

The custom file naming rule is custom**N**.tar.gz, N is number , and the number starts from 1.
For example: **custom1.tar.gz**.

Please refer to Appendix 1 for the format and naming conventions of custom files.

/web/custom_download

Description	Upload custom file in a web application.
URL Structure	**https://10.0.0.1/web/custom_download**
HTTP Method	POST：Transport a custom file to the module.
GET	Request： **GET /web/custom_download** [Example]: Request : **GET /web/custom_download** Content-type: application/json Response: 200 OK {

	"StatusCode": 0,
	"LastErrorCode": 0,
	"Status": "Init",
	"Message": "The update has not started yet. No error."
	}
	
	Or
	
	Content-type: application/json
	Response: 200 OK
	{
	"StatusCode": 4,
	"LastErrorCode": 2,
	"Status": "Error",
	"Message": "Update fail. Incorrect binary."
	}

POST	Request： **POST /web/custom_download** [Example]: Request: **POST /web/custom_download**, upload the custom file to the module. Content-Type：multipart/form-data; boundary=----WebKitFormBoundaryIcafShJTMskmwleL (generated by software randomization) … ----WebKitFormBoundaryIcafShJTMskmwleL\r\n Content-Disposition: form-data; name="filename"; filename="custom1.tar.gz"\r\n Content-Type：application/gzip \r\n\r\n <custom file data>\r\n ------WebKitFormBoundaryIcafShJTMskmwleL--\r\n Response: 200 OK

[GET]

Resource value definitions :

Field	Abbreviation	Data Type	Property	Description

<table>
<tr><td>Update status code</td><td>StatusCode</td><td>Number</td><td>R</td><td>Update status<table><tr><th>Status Code</th><th>Status Description</th></tr><tr><td>0</td><td>Init</td></tr><tr><td>1</td><td>Start</td></tr><tr><td>2</td><td>Finish</td></tr><tr><td>3</td><td>Updating</td></tr><tr><td>4</td><td>Error</td></tr><tr><td>99</td><td>Unknown</td></tr></table></td></tr>
<tr><td>Last error code</td><td>LastErrorCode</td><td>Number</td><td>R</td><td>Update flow last error code.<table><tr><th>Status Code</th><th>Status Description</th></tr><tr><td>0</td><td>SUCCESS</td></tr><tr><td>1</td><td>FILE_ERROR</td></tr><tr><td>2</td><td>INCORRECT_BINARY</td></tr><tr><td>3</td><td>INCORRECT_FILE_NAME</td></tr></table></td></tr>
<tr><td>Update status</td><td>Status</td><td>String</td><td>R</td><td>Update status<table><tr><th>Status Code</th><th>Status Description</th></tr><tr><td>0</td><td>Init</td></tr><tr><td>1</td><td>Start</td></tr><tr><td>2</td><td>Finish</td></tr><tr><td>3</td><td>Updating</td></tr><tr><td>4</td><td>Error</td></tr><tr><td>99</td><td>Unknown</td></tr></table></td></tr>
<tr><td>Update flow message</td><td>Message</td><td>String</td><td>R</td><td>Update flow message.</td></tr>
</table>

[POST]

Resource value definitions :

Parameters for ***Name = " xxx "***	Description	Example
filename	The file contents	------WebKitFormBoundaryIcafShJTMskmwleL Content-Disposition:form-data;name="filename" ; Content-Type:application/octet-stream (text/html …) <…file data…>

Appendix

1. How to create custom file?

A custom file is a compressed package provided to meet specific customer needs. Customers can install multiple files at the relative positions of the system directory as needed. The installation program will execute the " tar xf custom1.tar.gz -C / " command to decompress the files to the root directory. Therefore, when creating the compressed package, it is necessary to fill in the complete directory.

The naming rule is custom**N**.tar.gz, N is number , and the number starts from 1. For example: **custom1.tar.gz**. The following is an example of a compressed package:

```
root@adam6717:~# tar zvcf custom1.tar.gz /home/root/example/ /etc/init.d/advantech.sh
/home/root/example/
/home/root/example/b
/home/root/example/folder_1/
/home/root/example/folder_1/c
/home/root/example/a
/etc/init.d/advantech.sh
```