ADVANTECH Enabling an Intelligent Planet

WebAccess MQTT

Design Specification for Device

Version：1.3.5

Author：Oz.Chen, Steven.Li
Advantech Co., Ltd

History Record：

Version	Date	Editor	Notes
1.0.0	2016/02/01	Oz.Chen	First
1.0.1	2016/02/17	Oz.Chen	Add default value for Tag Common Part
1.0.2	2016/03/18	Oz.Chen	Add file type for file transfer
1.0.3	2016/03/23	Oz.Chen	Add action fomat
1.0.4	2016/03/25	Oz.Chen	Add heartbeat interval Add primary device ID Modify heartbeat
1.1.0	2016/04/13	Oz.Chen	Add data recovery Add time syncnization
1.1.1	2016/04/15	Oz.Chen	Add get tag list with datalog or RTDB enabled
1.1.2	2016/05/06	Oz.Chen	Add device and tags delete all
1.1.3	2016/05/11	Oz.Chen	Modify data recovery format
1.1.4	2016/05/13	Oz.Chen	Modify time stamp format
1.1.5	2016/05/24	Oz.Chen	Remove data recovery tag list download request
1.1.6	2016/06/07	Oz.Chen	Remove PRI, add BID for configuration
1.1.7	2016/07/04	Oz.Chen	Modify data communication mechanism at start time
1.2.0	2017/09/27	Oz.Chen	Change Document Title
1.2.1	2017/11/17	Steven Li	Modify the wrong format of data recovery example
1.2.2	2018/07/24	Oz.Chen	Modify MQTT connection setting and add "/" for invalid tag name character
1.2.3	2019/03/05	Oz.Chen	Modify broker setting description and remove Data On/Off command
1.2.4	2019/06/03	Oz.Chen	Add text tag for configuration
1.2.5	2019/10/02	Oz.Chen	Add project configuration port indication
1.2.6	2019/10/02	Oz.Chen	Add topic description for chapter Action format
1.2.7	2020/09/25	Oz.Chen	Add alarm feature
1.3.0	2021/03/25	Oz.Chen	Remove unused part and add service status format
1.3.1	2021/06/03	Oz.Chen	Add read device historical value and command ack format
1.3.2	2021/06/15	Oz.Chen	Add block support
1.3.3	2021/06/21	Oz.Chen	Modify block example
1.3.4	2021/07/09	Oz.Chen	Modify Read Historical Value offset from string to number
1.3.5	2021/07/16	Oz.Chen	Modify wrong format of command ack example

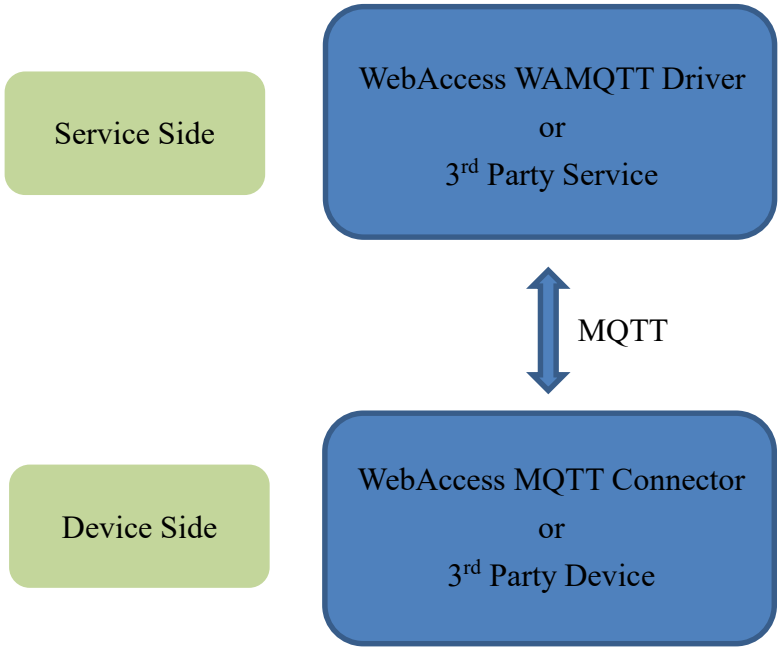
ADVANTECH *Enabling an Intelligent Planet*

Table of Contents

1. Connect to WebAccess with MQTT

Devices which want to connect to WebAccess with MQTT.

1.1 Architecture



1.2 Connection Parameter

1.2.1 WebAccess MQTT Broker

TCP Host: Domain name or IP address
TCP Port: Default *1883* for TCP, *8883* for TLS
Websocket Host: Domain name or IP address
(Ex: https://192.168.0.100:51329)
Websocket Port: Default *51328* for WS(WebSocket), *51329* for WSS(Secure WebSocket)

1.2.2 Client ID

Client ID must be unique for each device and the format is suguested as the following (optional):
d:group_id:type_id:dev_id

1.2.2.1 d

It means devices.

1.2.2.2 group_id

group_id is composed by WebAccess project name and SCADA name. It is seperated by a under line, for example: MyProject_MySCADA.
The maximum length of **group_id** is 65, and each of project name and SCADA name is 32.

1.2.2.3 type_id

It means device type.

type_id	Device Type
0	Default or Unknow
1	WebAccess/SCADA
2	WebAccess/HMI
3	WISE4000
4	EdgeLink
5	B+B Wzzard
6	WISE6000

1.2.2.4 dev_id

It means device ID. The maximum length is 32. **dev_id** must be unique under same **group_id**.

1.2.3 QoS

The quality of service (QoS) level is an agreement between sender and receiver of a message regarding the guarantees of delivering a message.

1.2.3.1 QoS0

At most once.

1.2.3.2 QoS1

At least once. (QoS1 is default suggestion)

1.2.3.3 QoS2

Exactly once. (not support yet)

1.2.4 Username / Password

User name and password for MQTT broker
Each of maximum length is 32.

1.2.5 SSL/TLS

WebAccess MQTT broker is support TCP and WebSocket with secure.

1.2.6 Conclusion

With those connection parameters for WebAccess MQTT broker, there must be some UI for user to input relative information on the deivces.

1.3 Topic Define

There are several MQTT topic as the following:

1.3.1 Data

iot-2/evt/**wadata**/fmt/**group_id**
This topic is for devices sending their real-time data.

1.3.2 Configuration

iot-2/evt/**wacfg**/fmt/**group_id**
This topic is for devices sending their configuration.

1.3.3 Connection

iot-2/evt/**waconn**/fmt/**group_id**
This topic is for devices sending their connection status.

1.3.4 Alarm

iot-2/evt/**waalm**/fmt/**group_id**
This topic is for devices sending their alarm status.

1.3.5 Command

iot-2/evt/**wacmd**/fmt/**group_id**
iot-2/evt/**wacmd**/fmt/**group_id/dev_id**
This two topics are for devices receiving command from service.

1.3.6 Command Ack

iot-2/evt/**wacmdack**/fmt/**group_id**

This topic is for devices sending their command acks.

1.3.7 Service Status

iot-2/evt/**wast**/fmt/**group_id**

This topic is for service sending its connection status.

1.4 Root Json Format

There are only two property in root json format.

Format:

```
{
    "d": {



    },
    "ts": "2020-04-04T23:26:10+08:00"
}
```

Property:

Name	Limitation	Description
d	Json Format	User define data
ts		UTC Time stamp, support format as the following: 1. 2020-05-13T02:52:51**Z** (seconds) 2. 2020-05-13T02:52:51.742**Z** (milliseconds) 3. 2020-05-13T10:52:51**+**08:00 (seconds with time zone) 4. 2020-05-13T10:52:51.742**+**08:00 (milliseconds with time zone)

2. MQTT Communication Format

In this chapter, it will define json format of data, file, configuration, command and connection topics.

There are several representation styles as the following:

- Red means primary key
- Green means necessary
- Blue means optional but nice to have
- Black means optional
- . . . means user define config space

It is case sensitive for all json property.

2.1 Data Format

Devices publish real-time data to this topic:

iot-2/evt/**wadata**/fmt/**group_id**

2.1.1 Tag Value

It is important for saving bandwidth that devices should publish real-time data which the tag values are changed. It is recommended to have deadband setting for each tag when difference between current and last published value is over some percentage.

Format:

```
{
    "unique string": {
        "Val": {
            "unique string1": number,
            "unique string2": "string",
            "unique string3": {
                "0": number,
                "1": number,
                "2": number,
                . . .
            }
        }
    },
    . . .
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
unique string1~3	21	Tag ID, must be unique under same WebAccess project and can not be "**(**", "**&**", "**,**", "**.**", "**%**", "**=**", "**#**", "**/**" and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address) Tag Value can be number, string or array (index from 0). One single "*****" is for bad value. The value should be bad when devices or I/O can not be reached.

*The maximum length of WebAccess Tag Name is limited to 21 bytes.

Example for normal value:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Val": {
                "W4010-28A7:Fz1Volt": 1213.48,
                "W4010-28A7:Fz1Temp": 63.81,
                "W4010-28A7:Fz1Desc": "Freezer 1",
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

Example for array value:


```
{
    "d": {
        "WISE4010-7F28A7": {
            "Val": {
                "W4010-28A7:Fz1Volt": {
                    "0": 963.56,
                    "3": 1147.38,
                    "4": 1038.84
                }
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```


Example for bad value:


```
{
    "d": {
        "WISE4010-7F28A7": {
            "Val": {
                "W4010-28A7:Fz1Volt": "*",
                "W4010-28A7:Fz1Temp": "*",
                "W4010-28A7:Fz1Desc": "*"
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.1.2 Data Recovery

Service sides can collect data from devices and record those to database for query or analytics usage. However, it gets data lost when devices is offline unexpectedly. Therefore, It will be neccesary to recover the lost data as the following when devices found that they didn't publish some period of data succeccfully.

Format:

```
{
    "unique string": {
        "DRec": {
            "From": number,
            "Tags": {
                "unique string1": {
                    "offset1": number,
                    "offset2": number,
                    ...
                },
                "unique string2": {
                    "offset1": "string",
                    "offset2": "string",
                    ...
                }
            }
        }
    },
    ...
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
DRec		Data Recovery
From		Seconds since 00:00 hours, Jan 1, 1970 UTC
unique string1~N	21	Tag ID, must be unique under same WebAccess project and can not be "**(**", "**&**", "**,**", "**.**", "**%**", "**=**", "**#**", "**/**" and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address) Tag Value can be number, string or array (index from 0). One single "*****" is for bad value. The value should be bad when devices or I/O can not be reached.
offset 1~N		Time Offset with "**From**" in seconds. Value can be number, string or array (index from 0)

Example:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "DRec": {
                "From": 1460129890,
                "Tags":{
                    "W4010-28A7:Fz1Volt": {
                        "3": 147.32,
                        "18": 149.18,
                        "87": 148.93
                    },
                    "W4010-28A7:Fz2Volt ": {
                        "0": 73.83,
                        "118": 84.49,
                        "124": 68.17
                    }
                }
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.2 Configuration Format

Devices publish configuration to this topic:
iot-2/evt/**wacfg**/fmt/**group_id**

2.2.1 Introduction

On WebAccess, it has hierarchy configuration as the following:
+Project
    +SCADA
        +Port
            +Device
                +Tag

It can be configurated automatically on WebAccess if devices publish sufficient configuration information. Devices should keep a last uploaded configuration profile, and sending differential part to service every time by "UTg" or "DTg". In this way, devices can add, modify or delete tag configuration on WebAccess.

2.2.2 Device

Format:

```
{
    "unique string": {
        "TID": number,
        "Dsc": "string",
        "Hbt": number,
        "PID": number,
        "BID": "string",
        "UTg": {tag format},
        "DTg": {tag format},
        "Del": 1,
        . . .
    }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
TID	1~5	Device Type 1: WebAccess/SCADA 2: WebAccess/HMI 3: WISE4000 4: ADAM3600 5. B+B Wzzard
Dsc	64	Description
Hbt	1~65535	Interval of Heartbeat (2.5.4), Suggest 5 or 10 Seconds Should be Good
PID	1~60	Inicate Project Port Number of the Device
BID	31	Backup Device ID left backup device ID if the device is backup
UTg		Add or Update Tag
DTg		Delete Tag
Del	1	Delete Device and All Tags on Service

PS: Command order is "**Del**" → "**DTg**" → "**UTg**" if they show at same time

2.2.3 Tag Common Part

Format:

```
"unique string": {
      "TID": number,
      "Dsc": "string",
      "Ary": number,
      "RO": number,
      "Blk": "string"
      . . .
}
```

Property:

Name	Limitation	Description	Default
unique string	21	Tag ID, must be unique under same WebAccess project and can not be "**(**", "**&**", "**,**", "**.**", "**%**", "**=**", "**#**", "**/**" and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address)	
TID	1~3	Tag Type 1: Analog, 2: Digit, 3: Text	
Dsc	64	Description	
Ary	0~32767	Array Size	0
RO	0 or 1	Read Only (0:false, 1:true)	0
Blk	8	Block Type Name	

*The maximum length of WebAccess Tag Name is limited to 21 bytes.

*If "Blk" property is filled, tag id must contain semicolon ":" with format like {block name}:{parameter name}. For example: W4010-28A7:Fz1Volt which "W4010-28A7" will be constructed as block name in WebAccess project and "Fz1Volt" will be parameter. {block name} maximum length is 12 bytes and tag name maximum length is still 21 bytes.

2.2.3.1 Tag ID

It is recommend to use abbreviation because the length is limitation. WebAccess is tag based architecture so it is suggested that using the format as the following will be good:
{Device Unique Abbreviation}_{Tag Abbreviation}

For example, there is a WISE4012E with partial MAC address A924 and one of analog input channel can be show:
W4012-A924_AI0

Notice that devices should keep Tag ID to be unique under the same WebAccess project.

2.2.4 Analog Tag Advance Part

Format:

```
{
    "Log": number,
    "SH": number,
    "SL": number,
    "EU": "string",
    "DSF": "string",
    "Alm": number,
    "HHP": number,
    "HHA": number,
    "HiP": number,
    "HiA: number,
    "LoP": number,
    "LoA": number,
    "LLP": number,
    "LLA": number,
    . . .
}
```

Property:

Name	Limitation	Description	Default
Log	0 or 1	Log Data to RTDB (0:disabled, 1:enabled), RTDB means WebAccess Real-time Database	0
SH	double	Span High	1000
SL	double	Span Low	0
EU	10	Engineer Unit	
DSF	xx.xx	Display Format (Integer.Fraction) (xx: 0~15)	4.2
Alm	0 or 1	Alarm Enabled (0:false, 1:true)	0
HHP	0~99	HH Priority (0:disabled)	0
HHA	double	HH Alarm Limit	
HiP	0~99	High Priority (0:disabled)	0
HiA	double	High Alarm Limit	
LoP	0~99	Low Priority (0:disabled)	0
LoA	double	Low Alarm Limit	
LLP	0~99	LL Priority (0:disabled)	0
LLA	double	LL Alarm Limit	

Note1: HHP >= HiP, LLP >= LoP

Note2: Alm must be taken at same time when some relative alarm setting is updated

2.2.5 Digit Tag Advance Part

Format:

```
{
    "Log": number,
    "S0": "string",
    "S1": "string",
    "S2": "string",
    "S3": "string",
    "S4": "string",
    "S5": "string",
    "S6": "string",
    "S7": "string",
    "Alm": number,
    "S0L": number,
    "S1L": number,
    "S2L": number,
    "S3L": number,
    "S4L": number,
    "S5L": number,
    "S6L": number,
    "S7L": number,
    . . .
}
```

Property:

Name	Limitation	Description	Default
Log	0 or 1	Log Data	0
S0	12	State 0	0
S1	12	State 1	1
S2	12	State 2	NotUsed
S3	12	State 3	NotUsed
S4	12	State 4	NotUsed
S5	12	State 5	NotUsed
S6	12	State 6	NotUsed
S7	12	State 7	NotUsed
Alm	0 or 1	Alarm Enabled (0:false, 1:true)	0
S0P	0~99	State 0 Alarm Priority	0
S1P	0~99	State 1 Alarm Priority	0
S2P	0~99	State 2 Alarm Priority	0
S3P	0~99	State 3 Alarm Priority	0
S4P	0~99	State 4 Alarm Priority	0
S5P	0~99	State 5 Alarm Priority	0
S6P	0~99	State 6 Alarm Priority	0
S7P	0~99	State 7 Alarm Priority	0

2.2.6 Text Tag Advance Part

Format:

```
{
    "Len": number,
    ...
}
```

Property:

Name	Limitation	Description	Default
Len	0~70	Log Data	70

2.2.7 Configuration Download

Devices should keep a up-to-date configuration profile. If devices miss or mess up the profile, it can be restored by this request command. For example: device replacement, the new one need to fill up the device ID of the broken one, then using this request command to get back up-to-date configuration profile from service.

Format:

```
{
    "unique string": {
        "Cdl": 1
    }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
Cdl	1	Config Download

It will return configuration from 2.4.2 after service receive this request commnad.

Example:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Cdl": 1
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.2.8 Add Tag Example (with block configuration)


```
{
    "d": {
        "WISE4010-7F28A7": {
            "TID": 3,
            "Dsc": "Factory A",
            "Hbt": 5,
            "PRI": "",
            "UTg": {
                "W4010-28A7:Fz1Volt": {
                    "TID": 1,
                    "Dsc": "Freezer1 Voltage",
                    "RO": 1,
                    "Log": 1,
                    "SH": 2500,
                    "SL": 0,
                    "EU": "V",
                    "DSF": "4.2",
                    "Alm": 1,
                    "HHP": 0,
                    "HHA": 0,
                    "HiP": 1,
                    "HiA: 1800,
                    "LoP": 1,
                    "LoA": 500,
                    "LLP": 0,
                    "LLA": 0,
                    "Blk": "WISE4010"
                },
                "W4010-28A7:Fz2Volt": {
                    "TID": 1,
                    "Dsc": "Freezer2 Voltage",
                    "RO": 1,
                    "Log": 1,
                    "SH": 2500,
                    "SL": 0,
                    "EU": "V",
```

```
            "Alm": 1,
            "HHP": 0,
            "HHA": 0,
            "HiP": 1,
            "HiA: 1800,
            "LoP": 1,
            "LoA": 500,
            "LLP": 0,
            "LLA": 0,
            "Blk": "WISE4010"
        },
        "W4010-28A7:Fz1Temp": {
            "TID": 1,
            "Dsc": "Freezer1 Temperature",
            "RO": 1,
            "Log": 1,
            "SH": 150,
            "SL": -50,
            "EU": "°C",
            "DSF": "3.2",
            "Alm": 1,
            "HHP": 0,
            "HHA": 0,
            "HiP": 1,
            "HiA: 90,
            "LoP": 1,
            "LoA": -10,
            "LLP": 0,
            "LLA": 0,
            "Blk": "WISE4010"
        },
        "W4010-28A7:Fz2Temp": {
            "TID": 1,
            "Dsc": "Freezer2 Temperature",
            "RO": 1,
            "Log": 1,
            "SH": 150,
```

```
        "DSF": "3.2",
        "Alm": 1,
        "HHP": 0,
        "HHA": 0,
        "HiP": 1,
        "HiA: 90,
        "LoP": 1,
        "LoA": -10,
        "LLP": 0,
        "LLA": 0,
        "Blk": "WISE4010"
    },
    "W4010-28A7:Fz1Switch": {
        "TID": 2,
        "Dsc": "Freezer1 Switch",
        "RO": 0,
        "Log": 1,
        "S0": "On",
        "S1": "Off",
        "S2": "",
        "S3": "",
        "S4": "",
        "S5": "",
        "S6": "",
        "S7": "",
        "Blk": "WISE4010"
    },
    "W4010-28A7:Fz1Mode": {
        "TID": 2,
        "Dsc": "Freezer1 Mode Switch",
        "RO": 0,
        "Log": 1,
        "S0": "Save",
        "S1": "Safe",
        "S2": "Lv1",
        "S3": "Lv2",
        "S4": "Lv3",
```

```
                "S7": "",
                "Blk": "WISE4010"
            },
            "W4010-28A7:Fz2Switch": {
                "TID": 2,
                "Dsc": "Freezer2 Switch",
                "RO": 0,
                "Log": 1,
                "S0": "On",
                "S1": "Off",
                "S2": "",
                "S3": "",
                "S4": "",
                "S5": "",
                "S6": "",
                "S7": "",
                "Blk": "WISE4010"
            },
            "W4010-28A7:Fz2Mode": {
                "TID": 2,
                "Dsc": "Freezer2 Mode Switch",
                "RO": 0,
                "Log": 1,
                "S0": "Save",
                "S1": "Safe",
                "S2": "Lv1",
                "S3": "Lv2",
                "S4": "Lv3",
                "S5": "Manaul",
                "S6": "",
                "S7": "",
                "Blk": "WISE4010"
            },
        }
    }
`
```

2.2.9 Update Tag Example


```
{
    "d": {
        "WISE4010-7F28A7": {
            "TID": 3,
            "Dsc": "Factory A",
            "Hbt": 5,
            "PRI": "",
            "UTg": {
                "W4010-28A7:Fz1Volt": {
                    "SH": 5000,
                    "SL": 500,
                },
                "W4010-28A7:Fz2Volt": {
                    "SH": 7500,
                    "SL": 2000,
                }
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.2.10 Delete Tag Example


```
{
    "d": {
        "WISE4010-7F28A7": {
            "TID": 3,
            "Dsc": "Factory A",
            "Hbt": 5,
            "PRI": "",
            "DTg": {
                "W4010-28A7:Fz2Temp": 1,
                "W4010-28A7:Fz2Switch": 1
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.2.11 Devices Plug and Play Mechanism

It still need to setup project, SCADA, ports , devices and tags because there is also a WebAccess software on service side. To reduce this configuration effort, auto-configuration from each device is a must feature, as we call "Plug and Play". Devices have control right to add, modify and delete configuration of WebAccess on service side.

- Devices must supoort these two way for Plug and Play:
  1. Add full configuration: through the way like 2.2.8. Usually, it is used in scenario as the following:
     - connect to the group_id at first time
     - device configuration was deleted by service side
  2. Partial update or delete configuration: through the way like 2.2.9 and 2.2.10. Devices should keep a up-to-date configuration profile, and update or delete configuration based on this profile.

- Device replacement scenario:
  Assume A is broken device, and B is new one.
  1. Find A and ready to replace with B.
  2. Set device ID of A to B
  3. Let B connect to service and get back up-to-date configuration profile through the way at 2.2.6. B should setup well according to this profile.
  4. Start to publish real-time data through way at 2.1.1 when B all gets ready.

2.3 Connection Format

Devices publish connection status to this topic:
iot-2/evt/**waconn**/fmt/**group_id**

2.3.1 Connect

Notify service side after connecting successfully.

Format:

```
{
      "unique string": {
            "Con": 1
      }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
Con	1	Connect

Example:

```
{
      "d": {
            "WISE4010-7F28A7": {
                  "Con": 1
            }
      },
      "ts": "2020-04-17T10:19:51+08:00"
}
```

2.3.2 Disconnect

Notify service side after disconnecting successfully.

Format:

```
{
    "unique string": {
        "DsC": 1
    }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
DsC	1	Disconnect

Example:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "DsC": 1
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.3.3 Unexpected Disconnect

Implement with MQTT Last Will mechanism to nodify service side after unexpected disconnection happening.

Format:

```
{
      "unique string": {
            "UeD": 1
      }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
UeD	1	Unexpected Disconnect

Example:

```
{
      "d": {
            "WISE4010-7F28A7": {
                  "UeD": 1
            }
      },
      "ts": "2020-04-17T10:19:51+08:00"
}
```

2.3.4 Heartbeat

Notify service side that the device is alive.

Format:

```
{
    "unique string": {
        "Hbt": 1
    }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
Hbt	1	Heartbeat for Keep Alive

Example:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Hbt": 1
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.3.5 Time Synchronization

Synchronize time from service side

Format:

```
{
    "unique string": {
        "TSyn": 1
    }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
TSyn	1	Time Synchronization

Example:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "TSyn": 1
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

(reference response command "UTC" at 2.5.3)

2.4 Alarm Format

Devices publish alarm to this topic:
iot-2/evt/**waalm**/fmt/**group_id**

2.4.1 Tag Alarm

Format:

```
{
    "unique string": {
        "Alm": {
            "unique string1": number,
            . . .
        }
    },
    . . .
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
unique string1	21	Tag ID, must be unique under same WebAccess project and can not be "**(**", "**&**", "**,**", "**.**", "**%**", "**=**", "#", "/" and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address) Tag Value can be number, string or array (index from 0). One single "*****" is for bad value. The

		value should be bad when devices or I/O can not be reached.
number	0~2	Alarm Status: 0: Normal 1: Alarm 2: Acked

*The maximum length of WebAccess Tag Name is limited to 21 bytes.

Example for normal value:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Alm": {
                "W4010-28A7:Fz1Volt": 0,
                "W4010-28A7:Fz1Temp": 1,
                "W4010-28A7:Fz1Desc": 2
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.5 Command Format

Devices should subscribe these two topics to receive commands from service:
iot-2/evt/**wacmd**/fmt/**group_id**
iot-2/evt/**wacmd**/fmt/**group_id/dev_id**

2.5.1 Write Value

Write values from service to the device
iot-2/evt/**wacmd**/fmt/**group_id/dev_id**

Format:

```
{
    "Cmd": "WV",
    "Val": {
        "unique string1": number,
        "unique string2": "string"
        "unique string3": {
            "0": number,
            "1": number,
            "2": number,
            . . .
        }
    }
}
```

Property:

Name	Limitation	Description
unique string1~3	21	Tag ID, must be unique under same WebAccess project and can not be "**(**", "**&**", "**,**", "**.**", "**%**", "**=**", "**#**", "/" and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address)

Example:


```
{
    "d": {
        "Cmd": "WV",
        "Val": {
            "W4010-28A7:Fz1Volt": 1213.48,
            "W4010-28A7:Fz1Temp": 63.81
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.5.2 Write Config

Write confgiuration from service to the device
iot-2/evt/**wacmd**/fmt/**group_id/dev_id**

Format:

```
{
    "Cmd": "WC",
    "UTg": {

    },
    "DTg": {

    },
    "Del": 1
}
```

Property:

Name	Limitation	Description
UTg		Add or Update Tags
DTg		Delete Tags
Del	1	Delete All Tags

Example: (modify tag span high and span low, and delete tag)


```
{
    "d": {
        "Cmd": "WC",
        "UTg": {
            "W4010-28A7:Fz2Volt": {
                "SH": 2000,
                "SL": 200
            },
            "W4010-28A7:Fz2Temp": {
                "SH": 150,
                "SL": -20
            }
        },
        "DTg": {
            "W4010-28A7:Fz1Volt": 1,
            "W4010-28A7:Fz1Temp": 1
        },
    },
    "ts": "2020-04-17T11:58:07+08:00"
}
```

2.5.3 Time Synchronization

Receive time synchronization from service
iot-2/evt/**wacmd**/fmt/**group_id/dev_id**

Format:

```
{
    "Cmd": "TSyn",
    "UTC": number
}
```

Property:

Name	Limitation	Description
TSyn		Time Synchronization
UTC		UTC Time from Service

Example:

```
{
    "d": {
        "Cmd": "TSyn",
        "UTC": 1460129890
    },
    "ts": "2020-04-17T13:33:29+08:00"
}
```

(reference request command "TSyn" at 2.3.5)

2.5.4 ACK

Receive tags ack from service

iot-2/evt/**wacmd**/fmt/**group_id/dev_id**

Format:

```
{
    "Cmd": "ACK"
    "Ack": {
        "unique string": 1
    }
}
```

Property:

Cmd Name	Limitation	Description
ACK		Ack alarm.
unique string	128	Tag ID

Example:

```
{
    "d": {
        "Cmd": "ACK",
        "Ack": {
            "W4010-28A7:Fz1Temp": 1
        }
    },
    "ts": "2020-09-17T13:33:29+08:00"
}
```

2.5.5 ACK All

Receive all tags ack from service

iot-2/evt/**wacmd**/fmt/**group_id/dev_id**

Format:

```
{
    "Cmd": "ACA"
}
```

Property:

Cmd Name	Limitation	Description
ACA		Ack all alarm under this device.

Example:

```
{
    "d": {
        "Cmd": "ACA"
    },
    "ts": "2020-09-17T13:33:29+08:00"
}
```

2.5.6 Read Historical Value

Read historical value from service to the device
iot-2/evt/**wacmd**/fmt/**group_id/dev_id**
Devices will response from format at 2.1.2

Format:

```
{
    "Cmd": "RH",
    "DHis": {
        "From": number,
        "Tags": {
            "unique string1": [
                offset1,
                offset2,
                . . .
            ],
            "unique string2": [
                offset1,
                offset2,
                . . .
            ]
        }
    }
}
```

Property:

Name	Limitation	Description
DHis		Data History
From		Seconds since 00:00 hours, Jan 1, 1970 UTC
unique string1~N	21	Tag ID, must be unique under same WebAccess project and can not be "**(**", "**&**", "**,**", "**.**", "**%**", "**=**", "**#**", "/" and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address)

	Tag Value can be number, string or array (index from 0). One single "*" is for bad value. The value should be bad when devices or I/O can not be reached.
offset 1~N	Time Offset with "**From**" in seconds.

Example:

```
{
    "d": {
        "Cmd": "RH",
        "DHis": {
            "From": 1460129890,
            "Tags":{
                "W2200-28A7:Fz1Volt": [
                    0,
                    900,
                    1800
                ],
                "W2200-28A7:Fz2Volt ": [
                    0,
                    1800,
                    3600
                ]
            }
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.5.7 Command List

Cmd	Property	Description
WV	Val	Write Value
WC	Cfg	Write Config
TSyn	UTC	Time Synchronization
ACK	Ack	Ack Tags
ACA		Ack All Tags
RH	DHis	Read Historical Value

2.6 Command Ack Format

Devices should publish this topic to acknowledge service which send command:
iot-2/evt/**wacmdack**/fmt/**group_id**

2.6.1 Ack Any Command

Format:

```
{
    "unique string": {
        "Cmd": "string",
        "Ts": "string"
    }
}
```

Property:

Name	Limitation	Description
unique string	31	Device ID (unique under same group_id)
Cmd		Support command at 2.5
Ts		UTC time which receive from command

Example for ack write value:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Cmd": "WV",
            "Ts": "2020-04-17T10:19:49+08:00"
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

Example for ack read device historical value:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Cmd": "RH",
            "Ts": "2020-04-17T10:19:50+08:00"
        }
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.7 Service Status Format

Device subscribe this topic to get connection status from service:
iot-2/evt/**wast**/fmt/**group_id**

2.7.1 Connect

Notify device side after connecting successfully.

Format:

```
{
    "Con": 1
}
```

Property:

Name	Limitation	Description
Con	1	Connect

Example:

```
{
    "d": {
        "Con": 1
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.7.2 Disconnect

Notify device side after disconnecting successfully.

Format:

```
{
    "DsC": 1
}
```

Property:

Name	Limitation	Description
DsC	1	Disconnect

Example:

```
{
    "d": {
        "DsC": 1
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```

2.7.3 Unexpected Disconnect

Implement with MQTT Last Will mechanism to nodify device side after unexpected disconnection happening.

Format:

```
{
    "UeD": 1
}
```

Property:

Name	Limitation	Description
UeD	1	Unexpected Disconnect

Example:

```
{
    "d": {
        "UeD": 1
    },
    "ts": "2020-04-17T10:19:51+08:00"
}
```