WISE-6610 Payload Engine

Introduction

Payload engine is one of method to parsing LoRaWAN payload, the payload engine is using Json format to descript payload.

Format Content

Basic

name	type	require	
appname	String	Yes	Must equal with node App Arguments.
out_topic	String	Yes	Output topic on MQTT.
devaddr	Boolean	No	Take devaddr on publish topic.
commheader	Boolean	No	If payload has common header.
loop	Boolean	No	If payload is multiple payload.
packet	packet[array]	Yes	

Packet

name	type	require	
fport	Number(1-255)	Yes	Must equal with uplink fport.
conditional	conditional	No	Conditional operator.
value	value[array]	Yes	

Conditional

name	type	require	
offset	Number(1-255)	Yes	Offset of payload.
value	Number(1-255)	Yes	Conditional operator.
and	Number(1-255)	No	Extra condition.
or	Number(1-255)	No	Extra condition.

Value

name	type	require	
name	string	Yes	
format	string	Yes	Format of this value.
name	string	Yes	Output name of this value on MQTT.
length	Number(1-255)	No	If format is string or ignore , this option can define length.
endian	String (big/little)	No	Default is big endian
arithmetic	Arithmetic[array]	No	If value need calculation , add this option.
bit	bit[array]	No	Getting bit value in this data.

Format Table

name	Size(byte)	
uint8	1	
uint16	2	
uint32	4	
uint64	8	
int8	1	
int16	2	
int32	4	
int64	8	
double32	4	
double64	8	
str	1-n	Default length is 1 , if no length option in this value.
boolean	1	
ignore	1-n	Default length is 1 , if no length option in this value.

Arithmetic

name	type	require	

action	string	Yes	See action table
value	Number/float	Yes	

Action Table

Name	
additon	Addition with value
substraction	substraction with value
multiply	multiply with value
division	division with value

bit

name	type	require	
name	string	Yes	
offset	Number(0-7)	Yes	Offset of this value byte.

Example

Basic Example

```
{
    "appname": "IR868LR",
    "out_topic": "IR868LR",
    "devaddr": true,
    "packet":
    [{
        "fport": 1,
        "value": [{
            "format": "uint16",
            "name": "temperature"
        },{
            "format": "uint16",
```

```
            "name": "humidity"
        },{
            "format": "uint8",
            "name": "pm2.5"
        }]
        },{
        "fport": 2,
        "value": [{
            "format": "int32",
            "name": "Voltage"
        },{
            "format": "str",
            "name": "status",
            "length":4
        }]
        }]
}
```


Fport 1

LoRaWAN Payload :

Fport :1

Paylod: 09C419910A

This data using first value format to parsing .

1st format is uint16 : 0x09C4 -> 2500

2nd format is uint16: 0x1991 ->6545

3rd format is uint8 :0x0A -> 10

On MQTT broke , we get 2 information.

This publish from network server , it's RAW data.

{"appargs":"IR868LR","data":"09C419910A","datetime":"2019-04-17T17:21:51Z","de
vaddr":"FF111111","fcnt":1,"port":1,"rssi":-30}

This publish from payload engine.

{"temperature":2500,"humidity":6545,"pm2.5":10}

Fport 2

LoRaWAN Payload :
Fport :2
Paylod: FFFFFFF5474F4F44

This data using 2ND value format to parsing .
1st format is int32 : 0xFFFFFFF5 -> -11
2nd format is str and length is 4: 0x474F4F44 -> “Good”

On MQTT broke , we get 2 information.
This publish from network server , it’s RAW data.
{"appargs":"IR868LR","data":"FFFFFFF5474F4F44","datetime":"2019-04-17T17:21:51Z","devaddr":"FF111111","fcnt":1,"port":2,"rssi":-30}
This publish from payload engine.
{"Voltage":-11,"status":"GOOD"}

Example with conditional option

If payload has multiple data type and all data using same fport, in this case , we can add conditional option in payload engine.

```
{
    "appname": "IR868LR",
    "out_topic": "IR868LR",
    "devaddr": true,
    "packet":
    [{
        "fport": 1,
        "conditional":{"offset":0,"value":1},
        "value": [{
            "format": "uint8",
            "name": "type"
        }, {
            "format": "uint16",
            "name": "temperature"
```

```
        },{
            "format": "uint16",
            "name": "humidity"
        },{
            "format": "uint8",
            "name": "pm2.5"
        }]
        },{
        "fport": 1,
        "conditional":{"offset":0,"value":2},
        "value": [{
            "format": "uint8",
            "name": "type"
        }, {
            "format": "int32",
            "name": "Voltage"
        },{
            "format": "str",
            "name": "status",
            "length":4
        }]
        }]
}
```

Case 1

LoRaWAN Payload :

Fport :1

Paylod: 0109C419910A

This publish from network server , it's RAW data.

{"appargs":"IR868LR","data":"0109C419910A","datetime":"2019-04-17T17:21:51Z","devaddr":"FF111111","fcnt":1,"port":1,"rssi":-30}

This publish from payload engine.

{"type":1,"temperature":2500,"humidity":6545,"pm2.5":10}

Case 2

LoRaWAN Payload :
Fport :1
Paylod: 02FFFFFFF5474F4F44

This publish from network server , it's RAW data.
{"appargs":"IR868LR","data":"02FFFFFFF5474F4F44","datetime":"2019-04-17T17:21:51Z","devaddr":"FF111111","fcnt":1,"port":1,"rssi":-30}
This publish from payload engine.
{"type":2,"Voltage":-11,"status":"GOOD"}

Example with bit option

If value is bit in a byte, in this case , we can add bit option in payload engine

```
{
    "appname": "IR868LR",
    "out_topic": "IR868LR",
    "devaddr": true,
    "packet":
    [{
        "fport": 1,
        "value": [{
            "format": "uint8",
            "name": "bit",
            "bit":[{
                "name":"TempLowAlarm",
                "offset":1
            },{
                "name":"TempHighAlarm",
                "offset":0

            }]
```

```
        }, {
            "format": "uint16",
            "name": "temperature"
        },{
            "format": "uint16",
            "name": "humidity"
        },{
            "format": "uint8",
            "name": "pm2.5"
        }]
        }]
}
```

Case 1

LoRaWAN Payload :
Fport :1
Paylod: 0109C419910A

This publish from network server , it's RAW data.

{"appargs":"IR868LR","data":"0109C419910A","datetime":"2019-04-17T17:21:51Z","devaddr":"FF111111","fcnt":1,"port":1,"rssi":-30}

This publish from payload engine.

{"TempLowAlarm":0"TempHighAlarm":1,"temperature":2500,"humidity":6545,"pm2.5":10}

Example with arithmetic

If value need covert to real value , we can and arithmetic in payload engine

```
{
    "appname": "IR868LR",
    "out_topic": "IR868LR",
    "devaddr": true,
    "packet":
    [{
```

```
"fport": 1,
"value": [{
        "format": "uint8",
        "name": "bit",
        "bit":[{
                "name":"TempLowAlarm",
                "offset":1
        },{
                "name":"TempHighAlarm",
                "offset":0

        }]
}, {
        "format": "uint16",
        "name": "temperature",
        "arithmetic":[{
                "action":"multiply",
                "value":0.01
        }]
},{
        "format": "uint16",
        "name": "humidity",
        "arithmetic":[{
                "action":"division",
                "value":100
        }]
},{
        "format": "uint8",
        "name": "pm2.5",
        "arithmetic":[{
                "action":"additon",
                "value":100
        },{
                "action":"substraction",
                "value":50
        }]
}]
}]
```

}

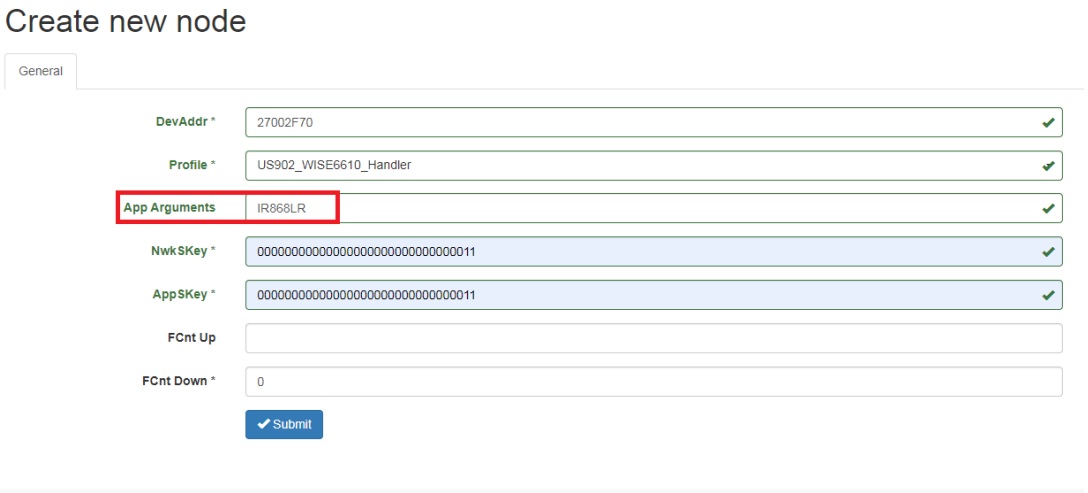
Case 1

LoRaWAN Payload :
Fport :1
Paylod: 0109C419910A

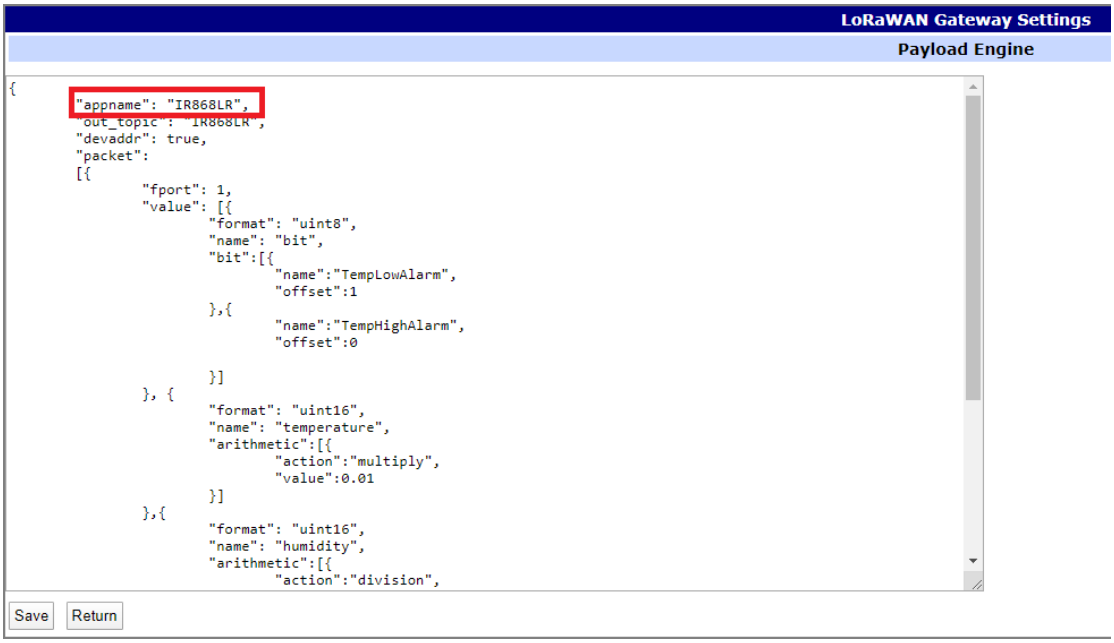
This publish from network server , it's RAW data.
{"appargs":"IR868LR","data":"0109C419910A","datetime":"2019-04-17T17:21:51Z","devaddr":"FF111111","fcnt":1,"port":1,"rssi":-30}
This publish from payload engine.
{"TempLowAlarm":0"TempHighAlarm":1,"temperature":25.000000,"humidity":65.450000,"pm2.5":60.000000}

SOP

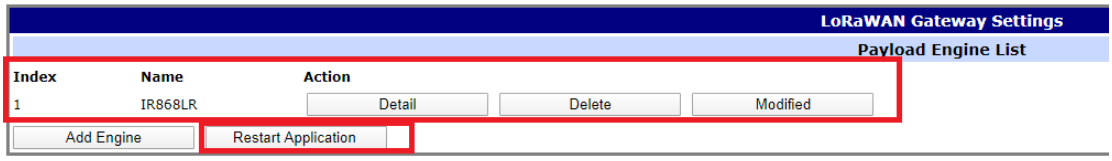
1. Create node on network server , and App Argument must equal appname in payload engine Json format.



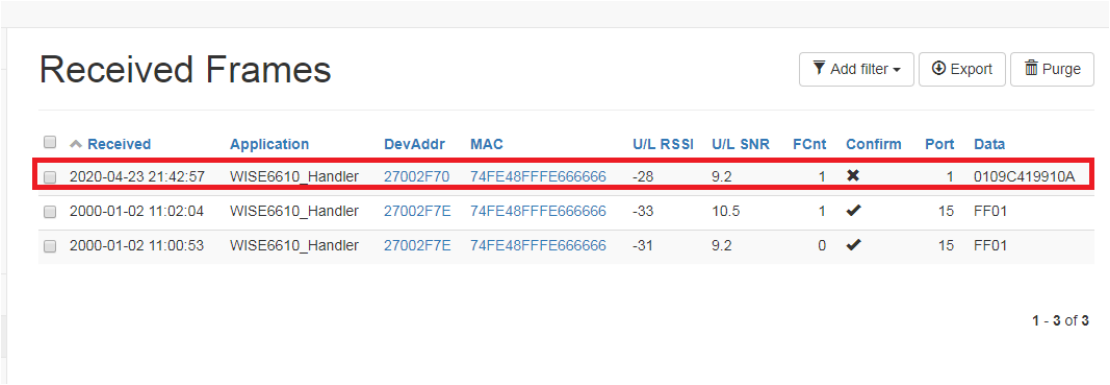
2. Add payload engine Json format on WEB.



3. Check payload engine is successful and restart application server



4. Check LoRaWAN network server has receive data .



5. Get MQTT data