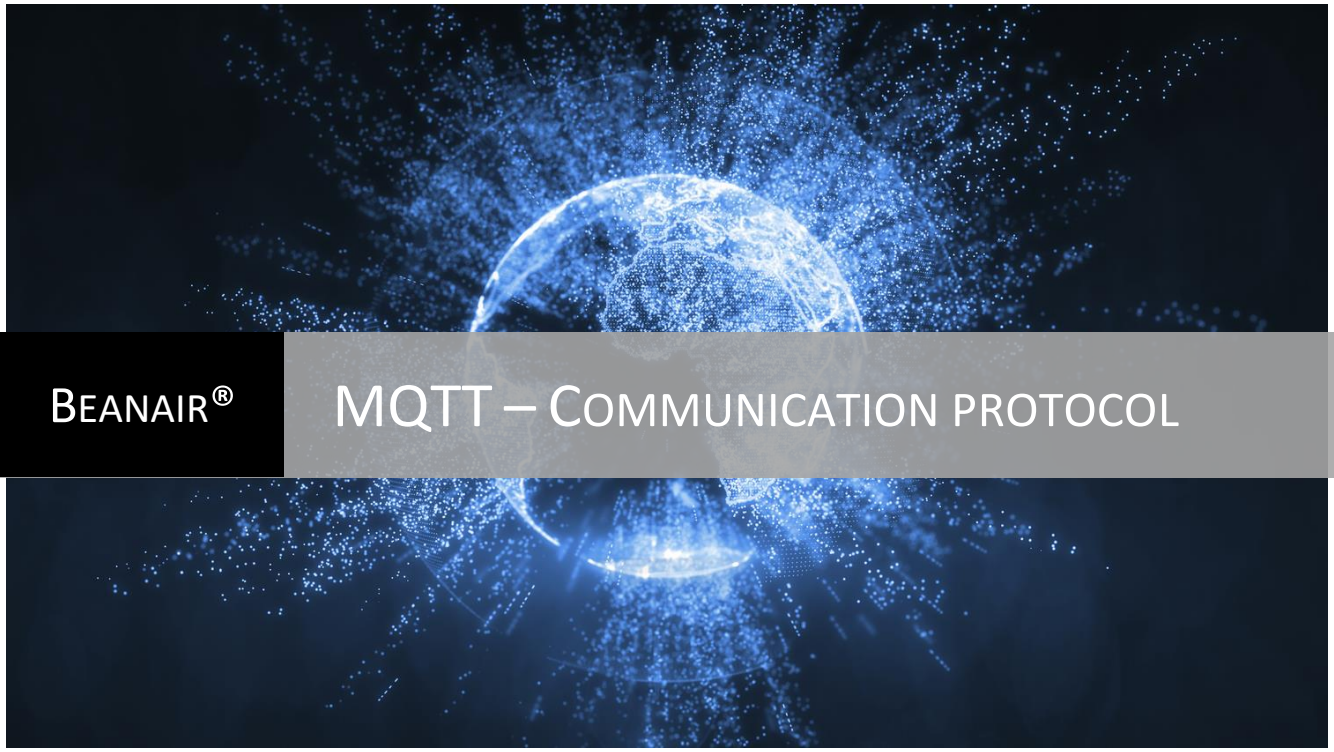
Version 2.5



BEANAIR® MQTT – COMMUNICATION PROTOCOL

BeanAir
Rethinking Sensing Technology

Wilow

DOCUMENT			
Document ID	**RF-TN-004**	**Version**	V2.5
External reference		**Date**	29/08/2022
Author	Ahmed Ben Amara, Embedded Software Engineer		
		Project Code	
Document's name	MQTT communication protocol		

VALIDATION			
Fonction	**Destination**	**For validation**	**For info**
Writer	Ahmed Ben Amara	✓	
Reader	Mohamed-Yosri Jaouadi	✓	
Validation	Antje Jacob		✓

DIFFUSION			
Fonction	**Destination**	**For action**	**For info**
Reader n°1	Antje Jacob, Production Manager	✓	
Reader n°2	Mohamed-Yosri Jaouadi., Embedded software engineer	✓	

UPDATES			
Version	**Date**	**Author**	**Evolution & Status**
V1.2	27/12/2016	Amouri Mootaz	• Added frames seen from data consumer side
V1.3	05/01/2017	Amouri Mootaz	• Added "Otac Topic To Subscribe to" update frame • Added how OTAC_Over_MQTT Topic is changed
V1.4	17/01/2017	Amouri Mootaz	• Delete the "payload length" from OTACs since it is included in Beanscape header • Consider module commands as OTAC • Deleted how OTAC_Over_MQTT Topic is changed • Frames IDs updated

UPDATES			
V1.5	27/01/2017	Amouri Mootaz	• Device's channel topic can be updated on-the-fly without stopping the MQTT module • Device's channel topic Profile is added 5 bytes of offset just before topic name field
V1.6	30/01/2017	Amouri Mootaz	• Format used to publish data to consumer is updated in LDCDA and ALARM Data Acquisition mode. • NetworkId is deleted from WiLo OTAC_Over_MQTT payload.
V1.7	16/02/2017	Amouri Mootaz	• Two Booleans decide if Username and Password fields are used in the OTAC and Profile or not. • Added Streaming acquisition mode • Added 3-bytes published frame descriptor
V1.8	06/03/2017	Amouri Mootaz	• Multicasting option in OTAC_Over_MQTT feature is added to WiLo products
V1.9	10/07/2018	Ahmed Ben Amara	• Format used to publish data to consumer is updated in Streaming burst and streaming one shot. • Format used to publish data to consumer is updated in Shock detection and SET mode. • Otac types • Profiles over Mqtt frame contents
V2.0	18/09/2018	Ahmed Ben Amara	• Update Streaming, SET mode and Sock detection frame: "Use two bytes from Future Use field to store Previous Number of data acquisitions per channel" • Update T_Subpacket equation
V2.1	28/09/2018	Ahmed Ben Amara	• Example of T_Subpacket calculation for streaming mode added
V2.2	08/05/2019	Ahmed Ben Amara	• Add millisecond part to Streaming mode/SET mode /Shock detection • Update Subpacket calculation for Shock detection mode • Update T_Subpacket equation
V2.221	10/05/2019	Mohamed Bechir Besbes	• Weblinks Update

UPDATES			

V2.3	24/03/2020	Habib Jomaa	• Update “Frames related to data acquisition mode”.
			  ○ Update the different beanair devices types ids table
			  ○ Add math result mode to the acquisition type ids table
			  ○ Add synchronization bit to each dynamic mode from “future use” bytes (streaming, S.E.T, SSD)
			  ○ Update the T_subPacket equation
			  ○ Update each dynamic mode (streaming, S.E.T and SSD) content
			  ○ Add LDC Math result specification
			  ○ Add Dynamic Math result to specification
			• Add example for each math result type.

V2.4	29/06/2021	Seddik ATTIG	• Appendices: MQTT configuration using cloud MQTT Broker
V2.5	29/08/2022	Seddik ATTIG	• Update Stop MQTT connection section

Disclaimer

The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Beanair GmbH, is strictly prohibited.

Beanair makes every effort to ensure the quality of the information it makes available. Notwithstanding the foregoing, Beanair does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information.

Beanair disclaims any and all responsibility for the application of the devices characterized in this document, and notes that the application of the device must comply with the safety standards of the applicable country, and where applicable, with the relevant wiring rules.

Beanair reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice.

Such changes will, nevertheless be incorporated into new editions of this document.

Contents

List of Tables

List of Figures

1. TECHNICAL SUPPORT

For general contact, technical support, to report documentation errors and to order manuals, contact ***Beanair Technical Support Center*** (BTSC) at:

tech-support@Beanair.com

For detailed information about where you can buy the Beanair equipment/software or for recommendations on accessories and components visit:

www.Beanair.com

To register for product news and announcements or for product questions contact Beanair's Technical Support Center (BTSC).

Our aim is to make this user manual as helpful as possible. Keep us informed of your comments and suggestions for improvements.
Beanair appreciates feedback from the users of our information.

2. VISUAL SYMBOLS DEFINITION

Symbols	*Definition*
	***Caution or Warning** – Alerts the user with important information about Beanair wireless sensor networks (WSN), if this information is not followed, the equipment /software may fail or malfunction.*
	***Danger** – This information MUST be followed if not you may damage the equipment permanently or bodily injury may occur.*
	***Tip or Information** – Provides advice and suggestions that may be useful when installing Beanair Wireless Sensor Networks.*

3. ACRONYMS AND ABBREVIATIONS

AES	Advanced Encryption Standard
CCA	Clear Channel Assessment
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
GTS	Guaranteed Time-Slot
kSps	Kilo samples per second
LDCDA	Low duty cycle data acquisition
LLC	Logical Link Control
LQI	Link quality indicator
LDCDA	Low duty cycle data acquisition
MAC	Media Access Control
PAN	Personal Area Network
PER	Packet error rate
RF	Radio Frequency
OTAC	Over the air configuration
WSN	Wireless sensor Network

4. OVERVIEW

This document covers the different frames exchanged between BeanDevice ® Wilow® MQTT module and supervision softwares. Messages exchanged are OTAC, SubProfiles reporting/update frames and module commands.
Useful extracted information from the data consumer side is described at the end of the document.

For a better understanding about Wilow® System, it is highly important to review the BEANDEVICE® WILOW® user Manual and Data acquisition modes available on the BeanDevice® Wilow® Technical note.

MQTT module OTAC frames	• The different OTACs frames sent to configure the MQTT module.
MQTT module SubProfiles	• The different subprofiles frames sent by BeanDevice Wilow .
OTAC Over MQTT structure	• Description of the structure of the frame that the user have to build to control the Wilow device.
Device channel's data format	• How device channel's data are published using the MQTT protocol.
Device profiles	• The current configuration of the device

5. MQTT MODULE OTAC (OVER THE AIR CONFIGURATION) SET FRAMES

5.1 DIFFERENT FRAMES ID

The different OTAC frames sent to the MQTT module are identified using the MQTT module ID and the specific OTAC Id, where :
MQTT_MODULE_CONFIG_MESS_ID = **0x90**
The OATC IDs are presented as follow:

Sub-profile	*Value*	*Description*
"Start module" MQTT Otac Id	*0x00*	*The Start command launches the MQTT state machine, data returned from devices are passed to FIFO and are published then to their configured topics.*
"Restart module" MQTT Otac Id	*0x01*	*The connection with the broker is restarted. The Gateway/Access point hosting the broker sends a DISCONNECT frame and then sends a new CONNECTION frame.*
"Stop module" MQTT Otac Id	*0x02*	*Stops the MQTT module. Firstly, the module will try to disconnect from the broker .*
"Client ID & Keep Alive Timer set" MQTT Otac Id	*0x03*	*The settings of the Client Id and the Keep Alive timer value used*
"Broker connection details set" MQTT Otac Id	*0x04*	*The different settings used to configure the Broker connection parameters*
"Password_&_Username set" MQTT Otac Id	*0x05*	*The password and User Name used to CONNECT to the Broker*
"LWT configuration set" MQTT Otac Id	*0x06*	*The Last Will Testament parameters used, main details reported are the Will topic and the Will message*
"Specific device's channel topic set" MQTT Otac Id	*0x07*	*The Topic used by a device's channel to send data over it*
"OTAC_Over_MQTT Topic set" MQTT Otac Id	*0x08*	*The topic subscribed to used for listening to OTAC sent over MQTT network*

Table 1: Different MQTT cartographies IDs

5.2 START MODULE FRAME

This command starts the MQTT module:

Parameter	Description	Default value	Dynamic
MQTT module Id	The Id of the MQTT module	0x90	8-bit
MQTT module specific OTAC Id	The Start command Id	0x00	8-bit

Table 2: Start module frame contents

5.3 RESTART CONNECTION

This command is used to:

- Delete previous non-published MQTT messages
- Restarts the connection with the Broker if connected

Parameter	Description	Default value	Dynamic
MQTT module Id	The Id of the MQTT module	0x90	8-bit
MQTT module specific OTAC Id	The Restart command Id	0x01	8-bit

Table 3: Restarts module frame contents

5.4 STOP CONNECTION

This command stops MQTT module:

Parameter	Description	Default value	Dynamic
Payload length	Length of the message		8-bit
MQTT module Id	The Id of the MQTT module	0x90	8-bit
MQTT module specific OTAC Id	The Stop command Id	0x02	8-bit

Table 4: Stop module frame contents

Example:

0xAA 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0x02, 0x90, 0x02
0xAA 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA: Device MAC ID
0x02: Payload length
0x90, 0x02: Payload (Stop MQTT)

5.5 CLIENT ID AND KEEP-ALIVE TIMER SET

The Client ID is used by the broker to distinguish each connected MQTT client, so it has to be unique to the broker.
If the same Client ID is detected in a CONNECT frame, the broker will assume that the same client is resending a new CONNECT frame and will disconnect the socket.
For this reason, the user is given the choice to supply his own ClientId or to generate it randomly in the BeanDevice® Wilow®.
The randomly generated Client Id is a safer option.
If the user supplies a ClientId with characters outside these "ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz0123456789" with a Length >23, an error should be prompt while typing.
If the user supplies a zero-length ClientId, the Clean Session bit in the CONNECT frame **must** be set to 0, otherwise, the Broker will reject the connection and return a CONNACK return code 0x02 (Identifier rejected).
To avoid such case, Zero-Length Client-Id **must** be avoided.

Parameter	*Description*	*Default value*	*Dynamic*
MQTT module Id	*The Id of the MQTT module*	*0x90*	*8-bit*
MQTT module specific OTAC Id	*The Sub-Id referring to the Client Id and "Keep-Alive timer_&_ClienId" set OTAC command*	*0x03.*	*8-bit*
Keep-Alive Timer	*The time interval in seconds PINGREQ messages should be sent to the broker to keep connection alive if no messages exchanged meanwhile*	*N.A.*	*16-bits LSB first*
Protocol version	*The protocol version used, either 0x03 for version V3.1 or 0x04 for V3.1.1*	*N.A.*	*8-bit*
Auto generated Client-Id flag	*If false the Client Id is given by the user else the client-id will be generated randomly*	*0x00*	*8-Bit*
Client-Id length	*The Client-Id string length*	*N.A.*	*8-bit*
Client-Id	*The Client-Id string*	*N.A.*	*23-Bytes*

Table 5: Client Id and KeepAlive Timer set frame content

5.6 BROKER TO-CONNECT-TO DETAILS SET

The user is free to connect to the broker using a given DNS address or using directly a given IP address. Supplying directly an IP address is useful with "Local Hosted" broker program for testing purpose

Parameter	*Description*	*Default value*	*Dynamic*
MQTT module Id	*The Id of the MQTT module*	*0x90*	*8-bit*

MQTT module specific OTAC Id	*The Sub-Id referring to the "Broker details" set OTAC command*	*0x04*	*8-bit*
Broker port	*The port used to connect to the broker*	*1883*	*16-bit LSB first*
Broker DNS flag	*If true the Broker DNS is valid address else the Broker IP address is valid*	*0x01*	*8-bit*
Broker IP	*Broker IP address*	*N.A.*	*32-Bit*
Broker DNS length	*Broker DNS string length*	*N.A.*	*8-Bits*
Broker DNS	*Broker DNS string*	*N.A.*	*50-Bytes*

Table 6: Broker link set frame contents when DNS flag = true

5.7 USERNAME AND PASSWORD SET

Configuring a password (Password flag == true) without a UserName (UsName flag == false) is prohibited.

Parameter	*Description*	*Default value*	*Dynamic*
MQTT module Id	*The Id of the MQTT module*	*0x90*	*8-bit*
MQTT module specific OTAC Id	*The Sub-Id referring to the "Password & Username" set OTAC command*	*0x05*	*8-bit*
UsName flag	*The username flag embedded in the CONNECT message*	*0x00*	*8-bit*
Password flag	*The password flag embedded in the CONNECT message*	*0x00*	*8-bit*
UsName length	*The Username string length*	*N.A.*	*8-bit*
UsName	*The Username string*	*N.A.*	*50-Bytes*
Password length	*The password string length*	*N.A.*	*8-bit*
Password	*The password string*	*N.A.*	*50-Bytes*

Table 7: Password and UserName set frame contents

5.8 LAST_WILL_TESTAMENT (LWT) PARAMETERS SET

The LWT MQTT feature can be used to inform interested devices (Should be subscribed to Will Topic, mainly data collecting machines) that the WIF Access Point disconnects abnormally or unexpectedly from the Broker.
Network failure causing disconnection is detected by a keep-Alive message absence (T > 1.5*KA) that the BeanGateway commits to send every KeepAlive time period specified at its connect attempt.
The Will_Retain_flag and the Will_QoS describes how the message will be transferred between Broker and interested data consumer devices.
If the **(Will Flag == false)**, the LWT feature is disabled, and **"Will Retain Flag" MUST be forced to 0**.

Parameter	Description	Default value	Dynamic
MQTT module Id	The Id of the MQTT module	0x90	8-bit
MQTT module specific OTAC Id	The Sub-Id referring to the "LWT details" set OTAC command	0x06	8-bit
LWT feature enable flag	LWT feature selection byte	0x00	8-bit
Will Retain flag	The Retain flag embedded in the will message PUBLISHed	N.A.	8-bit
Will QoS level	The Quality of Service embedded in the will message Published	N.A.	8-bit
Will topic length	Will topic string length	N.A.	8-bit
Will topic	Will topic string	N.A.	50-bytes
Will msg length	Will message string length	N.A.	8-bit
Will message	Will message string	N.A.	50-bytes

Table 8: LWT parameters set frame contents

5.9 DEVICE'S CHANNEL TOPIC SET

This frame is used to configure a device's channel topic name. This topic name is packed in the PUBLISH message alongside the data produced from this source.
Different devices channels, even channels from the same device, **can have the same topic name** and their data will be published using the same topic configured.

Parameter	Description	Default value	Dynamic
MQTT module Id	The Id of the MQTT module	0x90	8-bit
MQTT module specific OTAC Id	The Sub-Id referring to the device set OTAC command	0x07	8-bit
Dev-Id	The device's Id displayed in BeanScape	N.A.	16-bit
Chann Nbr	The channel number of the selected device	N.A.	8-bit
Enable Publishing	Enables device's channel publishing	0x00	8-bit
Retain flag	Retain flag embedded later in the PUBLISH message	N.A.	8-bit
Topic name len	The topic name string length	N.A.	8-bit
Topic name	The topic name string	N.A.	50-Bytes

Table 9: Device's channel topic set frame contents

5.10 DEVICE'S STREAMING TOPIC SET

The streaming topic is the one used by the BeanDevice® Wilow® to send all its channels measured data through MQTT.
The MQTT client (data consumer side) must parse the received frame to obtain the requested channels measurements separately.

Parameter	*Description*	*Default value*	*Dynamic*
MQTT module Id	*The Id of the MQTT module*	*0x90*	*8-bit*
MQTT module specific OTAC Id	*The Sub-Id referring to the device set OTAC command*	*0x07*	*8-bit*
Dev-Id	*The device's Id displayed in BeanScape*	*N.A.*	*16-bit*
Chann Nbr	*Constant value*	*250*	*8-bit*
Enable Publishing	*Enables device's streaming publishing*	*0x00*	*8-bit*
Retain flag	*Retain flag embedded later in the PUBLISH message*	*N.A.*	*8-bit*
Topic name len	*The topic name string length*	*N.A.*	*8-bit*
Topic name	*The topic name string*	*N.A.*	*50-Bytes*

Table 10: Device's streaming topic set frame contents

5.11 OTAC_OVER_MQTT TOPIC SET

The OTAC_Over_MQTT feature is helpful when a user wants to send OTAC commands to a remote BeanDevice® Wilow® connected to the same Broker, as if it was sent from BeanScape software over Ethernet.
The OTAC payload should be adapted accordingly to targeted Beanair product.
Of course the user must use a "shared" Topic configured earlier to use to SUBSCRIBE.

Parameter	*Description*	*Default value*	*Dynamic*
MQTT module Id	*The Id of the MQTT module*	*0x90*	*8-bit*
MQTT module specific OTAC Id	*The Sub-Id referring to the "OTAC_Over_MQTT Topic" set OTAC command*	*0x08*	*8-bit*
OtacOverMqtt flag	*Enable (if true) or Disable (if false) OTAC_OVER_MQTT feature*	*0x00*	*8-bit*
CleanSession flag	*The MQTT protocol feature is enabled (if true), disabled (if false)*	*0x00*	*8-bit*
New Topic to subscribe to len	*The New Topic string length*	*N.A*	*8-bit*
New Topic to subscribe to	*The New Topic string*	*N.A.*	*50-Bytes*

Table 11: OTAC_Over_MQTT topic set frame contents

6. BEANDEVICE® WILOW® MQTT SUBPROFILES PUBLISHED

6.1 BEANDEVICE® WILOW® MQTT MODULE SUB-PROFILE ID

Below is the different message identifiers used to report SubProfiles to supervision software.
The profile ID to target the MQTT module is fixed to MQTT_MODULE_PROFILE_ID = **0x90.**

Sub-profile	*Value*	*Description*
Sub-profile 90 – MQTT module status	*0x02*	*The status of the MQTT connection to display to the user*
Sub-profile 90 – MQTT Client ID and Keep Alive Timer	*0x03*	*The settings of the Client Id and the Keep Alive timer value used*
Sub-profile 90 – MQTT Broker connection details	*0x04*	*The different settings used to configure the Broker connection parameters*
Sub-profile 90 – MQTT Password & User Name	*0x05*	*The password and User Name used to CONNECT to the Broker*
Sub-profile 90 – MQTT LWT configuration	*0x06*	*The Last Will Testament parameters used, main details reported are the Will topic and the Will message*
Sub-profile 90 – Specific device's channel topic	*0x07*	*The Topic used by a device's channel to send data over it*
Sub-profile 90 – OTAC Over MQTT Topic used	*0x08*	*The topic subscribed-to use for receiving OTACs Over MQTT*

Table 12: SubProfiles IDs

All frames from or to the BeanDevice® Wilow® are preceded by profile header, and **are of a constant length** that depends on its type.

6.2 SUBPROFILE 90: MQTT STATUS

This frame is sent whenever the MQTT status is updated. The status is helpful when troubleshooting connections with the user.

Parameter	*Description*	*Default value*	*Dynamic*
MQTT_PROFILE_ID	*The MQTT module Profile ID*	*0x90*	*8-bit*
MQTT_STATUS_SUBPROFILE_ID	*The Id of the MQTT status*	*0x02*	*8-bit*

MQTT Status	*The Status of the MQTT connection displayed to the user, could be either :*		*0x02*	*8-bit*
	WAIT FOR SOCKET	*The module waits to create a socket*	*0x00*	
	WAIT FOR ETHERNET LINK	*The Ethernet cable is unplugged*	*0x01*	
	STOPPED	*MQTT module is disabled*	*0x02*	
	CONNECTING	*BeanGateway tries to connect to the Broker*	*0x03*	
	CONNECTED	*BeanGateway is MQTT Connected and is ready for sending BeanDevice data*	*0x04*	
	DISCONNECTING	*BeanGateway tries to disconnect from the Broker*	*0x05*	
	STOPPED FOR BAD CONFIG	*The BeanGateway (Wilow®) backup contains erroneous data, user must update his configuration*	*0x06*	
CONNACK message return code	*The CONNACK return code, it informs if the connection is well established with the Broker, and the failure reason*		*0x00*	*8-bit*
	CONNECTION ACCEPTED	*The Broker accepted the client connection*	*0x00*	
	CONNECTION REFUSED	*Unacceptable protocol version*	*0x01*	
		Identifier rejected	*0x02*	
		Server unavailable	*0x03*	
		Bad user name or password	*0x04*	
		Not authorized	*0x05*	
		NA	*0xFF*	

Table 13: MQTT Status report frame contents

6.3 SUBPROFILE 90: CLIENT ID AND KEEP ALIVE TIMER SETTINGS

If the (Forced flag == true) then the “Client-Id length” and the “Client-Id” fields will be updated with the Auto Generated ClientId.

Parameter	*Description*	*Default value*	*Dynamic*
MQTT_PROFILE_ID	*The MQTT module Profile ID*	*0x90*	*8-bit*

MQTT_CLIENT_ID_KA_TIMER_PROFILE_ID	*The Id of the MQTT Client-Id and Keep Alive settings report*	*0x03*	*8-bit*
Keep alive timer	*Keep alive timer value*	*60*	*16-bit LSB first*
Protocol version	*The protocol version used, can be either 0x03 for version V3.1 or 0x04 for V3.1.1*	*0x04*	*8-Bit*
Forced flag	*The flag describing if the Client Id is given by the user (true) or must be generated randomly (false).*	*0x01*	*8-Bit*
Client-Id length	*The Client-Id string length*	*N.A.*	*8-bit*
Client-Id	*The Client-Id string*	*N.A.*	*23-Bytes (constant)*

Table 14: Client Id and KeepAlive timer settings report frame contents

6.4 SUBPROFILE 90: BROKER CONNECTION SETTINGS

Parameter	*Description*	*Default value*	*Dynamic*
MQTT_PROFILE_ID	*The MQTT module Profile ID*	*0x90*	*8-bit*
MQTT_CLIENT_ID_KA_TIMER_PROFILE_ID	*The Id of the MQTT Client-Id and Keep Alive settings report*	*0x04*	*8-bit*
Broker Port	*The broker port used*	*1883*	*16-bits LSB first*
Broker DNS flag	*The Broker DNS address selected*	*N.A.*	*8-bit*
Broker Ip	*The broker IP address*	*N.A.*	*32-Bit*
Broker DNS length	*Broker DNS string length*	*N.A.*	*8-Bit*
Broker DNS	*Broker DNS string*	*N.A.*	*50-bytes (constant)*

Table 15: Broker connection settings report frame contents

6.5 SUBPROFILE 90: PASSWORD AND USER NAME SETTINGS USED

Parameter	Description	Default value	Dynamic
MQTT_PROFILE_ ID	The MQTT module Profile ID	0x90	8-bit
MQTT_PASSWORD_USERN AME_PROFILE_ID	The Id of the MQTT Password and username settings report	0x05	8-bit
UsName flag	The username flag embedded in the CONNECT message	N.A.	8-bit
Password flag	The password flag embedded in the CONNECT message	N.A.	8-bit
UsName length	The User Name string length	N.A.	8-bit
UsName	The User Name string	N.A.	50-bytes (constant)
Password length	The password string length	N.A.	8-bit
Password	The password string	N.A.	50-bytes (constant)

Table 16: Password and User Name frame contents

6.6 SUBPROFILE 90: MQTT LWT (LAST_WILL_TESTAMENT) SETTINGS USED

Parameter	Description	Default value	Dynamic
MQTT_PROFILE_ ID	The MQTT module Profile ID	0x90	8-bit
MQTT_WILL_ CONFIG_ PROFILE_ID	The Id of the MQTT LWT settings report	0x06	8-bit
LWT feature enable flag	LWT feature selection bit	N.A.	8-bit
Will Retain flag	The Retain flag embedded in the will message Published	N.A.	8-bit
Will QoS level	The Quality of Service embedded in the will message Published	N.A.	8-bit
Will topic length	Will topic string length	N.A.	8-bit
Will topic	Will topic string	N.A.	50-bytes
Will msg length	Will message string length	N.A.	8-bit
Will message	Will message string	N.A.	50-bytes

Table 17: LWT settings report frame contents

6.7 SUBPROFILE 90: DEVICE'S CHANNEL SETTINGS USED

If the user wants to "disable" Publishing a device's channel, the "Enable Publishing" byte **must** be set to **0x00**.
The device's channel topic can be updated on-the-fly, meaning the user doesn't need to stop the module to configure new one.
If the (Retain_flag == true), the last device's channel data will be saved in the Broker and transmitted whenever a data consumer device subscribes to that Topic.

Parameter	*Description*	*Default value*	*Dynamic*
MQTT_PROFILE_ID	*The MQTT module Profile ID*	*0x90*	*8-bit*
MQTT_DEVICE_TOPIC_CONFG_PROFILE_ID	*The Id of the MQTT one device's channel topic report*	*0x07*	*8-bit*
Device Nwk Id	*Device Id in the network*	*N.A.*	*16-bit*
Channel Nbr	*The device's channel number*	*N.A.*	*8-bit*
Enable Publishing	*Enables device's channel publishing*	*0x00*	*8-bit*
Retain flag	*The retained flag used when Publishing the device's channel data*	*N.A.*	*8-bit*
Device topic length	*The device's channel topic name length*	*N.A.*	*8-bit*
Offset bytes	*For future usage*	*0x00*	*5-bytes*
Device topic	*The device's channel used topic string*	*N.A.*	*50-bytes*

Table 18: Device's channel publish settings report frame contents

6.8 SUBPROFILE 90: DEVICE'S STREAMING TOPIC USED

Parameter	*Description*	*Default value*	*Dynamic*
MQTT_PROFILE_ID	*The MQTT module Profile ID*	*0x90*	*8-bit*
MQTT_DEVICE_TOPIC_CONFG_PROFILE_ID	*The Id of the MQTT one device's streaming topic report*	*0x07*	*8-bit*
Device Nwk Id	*Device Id in the network*	*N.A.*	*16-bit*
Channel Nbr	*Constant*	*250*	*8-bit*

Enable Publishing	*Device streaming topic used?*	*0x00*	*8-bit*
Retain flag	*The retained flag used when Publishing the device's streaming data*	*N.A.*	*8-bit*
Device topic length	*The device's streaming topic name length*	*N.A.*	*8-bit*
Offset bytes	*For future usage*	*0x00*	*5-bytes*
Device topic	*The device's streaming topic string*	*N.A.*	*50-bytes*

Table 19: Device's streaming topic report frame contents

6.9 SUBPROFILE 90: OTAC_OVER_MQTT SETTINGS USED

Parameter	*Description*	*Default value*	*Dynamic*
MQTT_PROFILE_ID	*The MQTT module Profile ID*	*0x90*	*8-bit*
MQTT_NEW_OTAC_TOPIC_CONFIG_ PROFILE_ID	*Id referring to the "OTAC_Over_MQTT Topic" topic name report*	*0x08*	*8-bit*
OtacOverMqtt flag	*Enables (if true) or Disables (if false) OTAC_OVER_MQTT feature*	*N.A.*	*8-bit*
CleanSession flag	*The MQTT protocol feature is enabled (if true), disabled (if false)*	*0x00*	*8-bit*
New Topic to subscribe to len	*The New Topic length*	*N.A*	*8-bit*
New Topic to subscribe to	*The New Topic*	*N.A.*	*8-bit*

Table 20: OTAC over MQTT settings report frame contents

7. OTAC_OVER_MQTT FRAME CONTENTS

The OTAC_over_MQTT feature is useful when the user wants to configure the BeanDevice® Wilow® using MQTT protocol without using BeanScape®.
The OTAC can target a single desired device or a group of devices as a multicasting option.
The device(s) addressing is implemented using a header added to the OTAC_over_MQTT frame (in the "OTAC_Over_MQTT payload" field) so that the addressed BeanDevice® Wilow® product can use it to filter out unwanted OTACs and know if it is concerned or not.
The RETAIN bit should be set to 0 to not resend the previous OTAC if the BeanDevice® Wilow® reconnects.

7.1 BEANDEVICE® WILOW® FRAME

To address the BeanDevice® Wilow®, the user **must** address it using the details below.

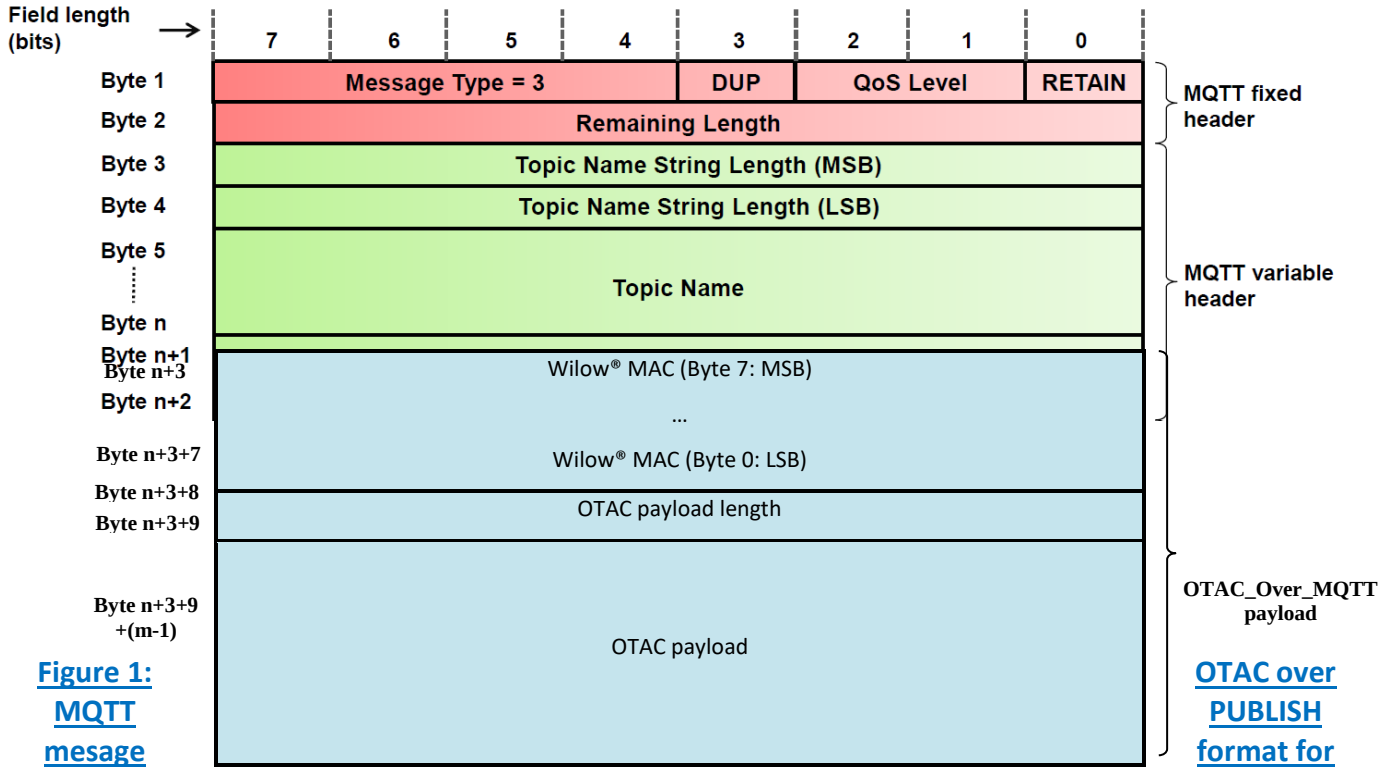


Figure 1: MQTT mesage OTAC over PUBLISH format for Wilow® products

7.2 WILOW® MULTICASTING FRAME

The same OTAC can be submitted to a group of BeanDevice® Wilow® when they are subscribed to the same Topic and is then "Broadcasted" to them by the Broker itself.

To benefit from the multicasting option, the "Wilow® MAC" parameter in the "OTAC_Over_MQTT payload" field **must be set to 0xFFFFFFFF**, this special MAC is not filtered by the Wilow® device and the OTAC is processed.

7.3 OTAC TYPES

7.3.1 Data acquisition configuration (DAQ) OTAC

This OTAC is responsible of configuring the acquisition mode (streaming, SET mode, Alarm, Low duty cycle), it also has the role of configuring the device in TX, log, TX & Log or Stand alone mode. The table below shows in details how the Daq OTAC frame should be organized to be interpreted by the device.

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0		
OTAC Length	1	8	23	
OTAC Id	1	9	0x10	
Daq mode	1	10		See Daq mode table
Daq options	2	11		See Daq options table
Future Use	2	13		
Daq duty cycle(Lsb first)	3	15		
TX Ratio	1	18		
Daq duration(Lsb first)	3	19		
Sampling Rate(Lsb first)	3	22		
Future Use	3	25		
Store and forward Data aging(Lsb first)	2	28		
Future Use	2	30		

Daq mode	value
Commissioning	0x01
low duty cycle	0x02
Streaming	0x03
Alarm	0x04
SET mode	0x05
Shock Detection	0x06
Daq mode Table	

Daq options bit	Signification
0	Datalogger bit: 1 = datalogger enabled 0 = datalogger disabled
1	Store and forward bit: 1 = Store and forward enabled 0 = Store and forward disabled
2	Streaming (bit2,bit3,bit4): Streaming Continuous = (1,0,0) Streaming one shot = (0,1,0) Streaming burst = (1,1,0)
3	
4	
5	Transmission TX bit: 1 = TX enabled 0 = TX disabled
6	Stand Alone bit: 1 = Stand Alone enabled 0 = Stand Alone disabled
7->15	Future use

Daq options table

7.3.1.1 Examples:

The examples of all the OTAC will be shown as follow
The OTAC frame is splitted to bytes with "-" character and each byte is interpreted in decimal.

7.3.1.1.1 Example 1: Streaming Burst log only

The first example shows a streaming burst OTAC with the following configuration:

- Streaming frequency: 25Hz
- Acquisition duration: 10 seconds
- Acquisition cycle: 5 minutes (300seconds)
- Log only

The OTAC frame example
244-184-94-0-166-230-0-0-23-16-3-13-0-0-0-44-1-0-0-10-0-0-25-0-0-0-0-0-0-0-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	8	23	
OTAC Id	1	9	16	
Daq mode	1	10	3	Streaming
Daq options	2	11	13-0	Burst + log only
Future Use	2	13	0-0	
Daq duty cycle(Lsb first)	3	15	44-1-0	
TX Ratio	1	18	0	No TX ratio in streaming
Daq duration(Lsb first)	3	19	10-0-0	10 Seconds
Sampling Rate(Lsb first)	3	22	25-0-0	25 Hz
Future Use	3	25	0-0-0	
Store and forward Data aging(Lsb first)	2	28	0-0	
Future Use	2	30	0-0	

7.3.1.1.2 Example 1: Low duty cycle TX & Log

The second example shows a low duty cycle OTAC with the following configuration:

- Acquisition cycle: 1Hour (3600 seconds)
- TX ratio: 5
- TX and Log

The OTAC frame example

244-184-94-0-166-230-0-0-23-16-2-33-0-0-0-16-14-0-5-0-0-0-0-0-0-0-0-0-0-0-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	8	23	
OTAC Id	1	9	16	
Daq mode	1	10	2	Low duty Cycle
Daq options	2	11	33-0	TX and Log
Future Use	2	13	0-0	
Daq duty cycle(Lsb first)	3	15	16-14-0	3600 seconds

TX Ratio	1	18	5	
Daq duration(Lsb first)	3	19	0-0-0	No duration in LDC
Sampling Rate(Lsb first)	3	22	0-0-0	No sampling rate in LDC
Future Use	3	25	0-0-0	
Store and forward Data aging(Lsb first)	2	28	0-0	
Future Use	2	30	0-0	

7.3.1.1.3 Other Examples

Here are other OTAC frames tested with a device with F4B85E00A6E60000 Mac ID,

- Streaming continuous TX only 500hz store and forward enabled rollover (data aging=65535"255-255"):

244-184-94-0-166-230-0-0-23-16-3-38-0-0-0-0-0-0-0-0-0-0-244-1-0-0-0-0-255-255-0-0

- Set mode sampling rate(100hz)/notification cycle(7200s)/duration(60s) Stand Alone:

244-184-94-0-166-230-0-0-23-16-5-64-0-0-0-32-28-0-0-60-0-0-100-0-0-0-0-0-0-0-0-0

- Shock detection notification cycle(20s)/duration (7 seconds) TX and log 400hz

244-184-94-0-166-230-0-0-23-16-6-33-0-0-0-20-0-0-0-7-0-0-144-1-0-0-0-0-0-0-0-0

7.3.2 System configuration OTAC

This OTAC is responsible of:

- Configuring the power mode (Sleep with network listening, Active)
- Configuring the diagnostic cycle
- Configuring the network listening cycle
- Lock / Unlock OTAC
- Enable/Disable Activity Led

The table below shows in details how the System OTAC frame should be organized to be interpreted by the device.

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0		
OTAC Length	1	8	6	
OTAC Id	1	9	0x21	

System configuration Bitmap	1	10		See Config bitmap Table(Page 35)
Power Mode	1	11		See Power mode Table
Diagnostic Cycle	1	12		Coefficient
Network listening cycle(lsb first)	2	13		in seconds

Config bit	Signification
0	OTAC Status bit:1 = OTAC locked,0 = OTAC unlocked
1	Activity Led bit,1 = Activity Led enabled,0 = Activity disabled
2->7	Future use
Config bitmap table	

Daq mode	value
Active mode	0x01
Sleep with network listening	0x03
Power mode Table	

7.3.2.1 Examples:

The examples of all the OTAC will be shown as follow
The OTAC frame is splitted to bytes with "-" character and each byte is interpreted in decimal.

7.3.2.1.1 Example 1: Sleep mode, Disabled Led OTAC locked and diagnostic cycle set 4

The first example shows a system OTAC with the following configurations:

- Sleep with network listening with listening cycle 25 seconds
- Diagnostic cycle coefficient set to 4
- Activity Led disabled
- OTAC unlocked

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-6-33-0-3-4-25-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000

OTAC Length	1	8	6	
OTAC Id	1	9	33	
System configuration Bitmap	1	10	0	Activity Led Disabled, OTAC unlocked
Power Mode	1	11	3	Sleep with network listening
Diagnostic Cycle	1	12	4	4
Network listening cycle(lsb first)	2	13	25-0	25 seconds

7.3.2.1.2 Example 2: Active mode, Enable Led, Lock OTAC and set diagnostic cycle to 10

The second example shows a system OTAC with the following configurations:

- Active mode
- Diagnostic cycle coefficient set to 10
- Activity Led enabled
- OTAC locked

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-6-33-3-1-10-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	8	6	
OTAC Id	1	9	33	
System configuration Bitmap	1	10	3	Activity Led enabled, OTAC locked
Power Mode	1	11	1	Active mode
Diagnostic Cycle	1	12	10	10
Network listening cycle(lsb first)	2	13	0-0	Not set in active mode

7.3.3 Shock detection configuration OTAC

This OTAC is in charge of:

- Setting the shock acceleration range
- Setting the shock sampling rate

- Setting the shock threshold

The table below shows in details how the Shock detection configuration OTAC frame should be organized to be interpreted by the device.

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0		
OTAC Length	1	8	13	
OTAC Id	1	9	0x42	
Acceleration Range(Lsb first)	2	10		In G
Shock Sampling Rate	2	12		
Shock notification delay	1	14		
Future Use	1	15		
Shock Threshold (Lsb first)	2	16		In mG
Future Use	4	18		

7.3.3.1 Examples:

The examples of all the OTAC will be shown as follow
The OTAC frame is splitted to bytes with "-" character and each byte is interpreted in decimal.

7.3.3.1.1 Example 1: Acceleration range 16g sampling rate 1600Hz Threshold 2000mg

The first example shows Shock detection OTAC with the following configurations:

- Shock detection acceleration range 16g
- Shock detection Sampling rate 1600Hz
- Shock Threshold 2000mg

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-13-66-16-0-64-6-0-0-208-7-0-0-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	8	13	
OTAC Id	1	9	66	0x42
Acceleration Range(Lsb first)	2	10	16-0	16g
Shock Sampling Rate	2	12	64-6	1600Hz

Shock notification delay	1	14	0	
Future Use	1	15	0	
Shock Threshold (Lsb first)	2	16	208-7	2000mg
Future Use	4	18	0-0-0-0	

7.3.3.1.2 Example 2: Acceleration range 4g sampling rate 100Hz Threshold 2850mg

The first example shows Shock detection OTAC with the following configurations:

- Shock detection acceleration range 4g
- Shock detection Sampling rate 100Hz
- Shock Threshold 2850mg

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-13-66-4-0-100-0-0-0-34-11-0-0-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	8	13	
OTAC Id	1	9	66	0x42
Acceleration Range(Lsb first)	2	10	4-0	4g
Shock Sampling Rate	2	12	100-0	100Hz
Shock notification delay	1	14	0	
Future Use	1	15	0	
Shock Threshold (Lsb first)	2	16	34-11	2850mg
Future Use	4	18	0-0-0-0	

7.3.4 Channel Configuration OTAC

This OTAC is responsible of:

- Setting the status of the channel x on/off where (x ∈[0..4])
- Setting alarm threshold of channel x where (x ∈[0..4])
- Setting the calibration of channel x where (x ∈[0..4])

The table below shows in details how the channel configuration OTAC frame should be organized to be interpreted by the device.

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0		
OTAC Length	1	9	43	
OTAC Id	1	10	0x82	
Sensor Id	1	11		see sensor id table
Daq Channel Bitmap	1	12		Bit 0 : Channel Status (1:Enable/0:Disable) Bit 1 : Alarm Threshold Set (1:Threshold updated/ 0: threshold not updated Bit 2 : Sensor Calibration(1:calibration Updated/0:No update on calibration)
Alarm H1(float)(Lsb First)	4	13		
Alarm H2(float)(Lsb First)	4	17		
Alarm L1(float)(Lsb First)	4	21		
Alarm L2(float)(Lsb First)	4	25		
Offset(float)(Lsb First)	4	29		
Ratio(float)(Lsb first)	4	33		
Future use	16	37		

Channel Id	Signification
0	Channel Z
1	Channel X
2	Channel Y
3	INC_X
4	INC_Y

Config bitmap table

7.3.4.1 Examples:

The examples of all the OTAC will be shown as follow
The OTAC frame is splitted to bytes with "-" character and each byte is interpreted in decimal.

7.3.4.1.1 Example 1: Calibrate, Enable of channelY

The first example shows a channelY (id=2) configuration OTAC with the following configurations:

- Calibration of channel with ratio=-2,68 and offset=0,59
- Channel enabled

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-43-130-2-5-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-61-10-23-63-31-133-43-192-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	9	43	
OTAC Id	1	10	130	0x82
Sensor Id	1	11	2	ChannelY
Daq Channel Bitmap	1	12	5	▪ Channel Enabled ▪ Alarm Threshold not updated ▪ calibration Updated
Alarm H1(float)(Lsb First)	4	13	0-0-0-0	Not updated
Alarm H2(float)(Lsb First)	4	17	0-0-0-0	Not updated
Alarm L1(float)(Lsb First)	4	21	0-0-0-0	Not updated
Alarm L2(float)(Lsb First)	4	25	0-0-0-0	Not updated
Offset(float)(Lsb First)	4	29	61-10-23-63	0,59
Ratio(float)(Lsb first)	4	33	31-133-43-192	-2,68
Future use	16	37	0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0	

7.3.4.1.2 Example 2: Set Alarm Threshold levels of Channel INC_Y

The first example shows a channel INC_Y (id=4) Config OTAC with the following configurations:

- Alarm Level are H1=13,3;H2=9,78;L1=-8,65;L2=-26,81
- Channel enabled

The example OTAC frame is the following:

244-184-94-0-166-230-0-0-43-130-4-3-205-204-84-65-225-122-28-65-102-102-10-193-225-122-214-193-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	9	43	
OTAC Id	1	10	130	0x82
Sensor Id	1	11	2	ChannelY
Daq Channel Bitmap	1	12	3	▪ Channel Enabled ▪ Alarm Threshold updated ▪ calibration not Updated
Alarm H1(float)(Lsb First)	4	13	205-204-84-65	H1=13,3
Alarm H2(float)(Lsb First)	4	17	225-122-28-65	H2=9,78
Alarm L1(float)(Lsb First)	4	21	102-102-10-193	L1=-8,65
Alarm L2(float)(Lsb First)	4	25	225-122-214-193	L2=-26,81
Offset(float)(Lsb First)	4	29	0-0-0-0	Not updated
Ratio(float)(Lsb first)	4	33	0-0-0-0	Not updated
Future use	16	37	0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0	

7.3.5 Clock configuration OTAC

This OTAC is responsible of:

- Setting the time zone of the device's clock

- Setting the ntp Configurations (Port, URL, Server name…)

The table below shows in details how the clock configuration OTAC frame should be organized to be interpreted by the device.

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0		
OTAC Length	1	8	46	
OTAC Id	1	10	0x91	
Time zone(signed)(Lsb first)	2			one lsb = 1 minute
Future Use	5			
Ntp Port	2			
Enable DNS	1			
Ntp server IP	4			
Server name length	1			
Server URL	30			

7.3.5.1 Examples:

The examples of all the OTAC will be shown as follow
The OTAC frame is splitted to bytes with "-" character and each byte is interpreted in decimal.

7.3.5.1.1 Example 1: Change Ntp server, change time zone

The first example shows Ntp Configuration OTAC with the following configurations:

- DNS Enabled
- Ntp server"time.google.com"
- Time zone =60 minutes

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-46-145-60-0-0-0-0-0-0-123-0-1-0-0-0-0-15-116-105-109-101-46-103-111-111-103-108-101-46-99-111-109-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0

Name field	Size in byte	Byte index	Value	Additional information
Mac Id(Msb First)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	8	46	
OTAC Id	1	10	145	0x91
Time zone(signed)(Lsb first)	2		60-0	60 minutes
Future Use	5		0-0-0-0-0	

Ntp Port	2		123-0	123
Enable DNS	1		1	DNS enabled
Ntp server IP	4		0-0-0-0	Not set DNS when DNS enabled
Server name length	1		15	15 bytes
Server URL	30		116-105-109-101-46-103-111-111-103-108-101-46-99-111-109-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0	time.google.com

7.3.6 Datalogger Config OTAC

There are two type of Datalogger OTAC

1. General datalogger OTAC
2. Download Response OTAC

7.3.6.1 General Datalogger OTAC

This OTAC is responsible of:

- Setting the end of memory strategy
- Erase ,download cancel download
- Stop logging

The table below shows in details how the General Datalogger OTAC frame should be organized to be interpreted by the device.

General Datalogger OTAC frame				
Field Name	Size in bytes	index in bytes	Values	Additional information
MacId(Msb first)	8	0		
OTAC Length	1	8		
OTAC Id	1	9	0xD0	
End of memory management	1	10		See table end of memory setting values
Download setting	1	11		See table Download Setting values
Index first File to Download(Lsb first)	2	12		

Config	Value
Stop log	0x01
Stop keep Daq	0x02
Stop Go to Commissioning	0x03
Stop auto download erase reset Daq	0x04
Stop auto download switch to commissioning	0x05
Stop auto download erase switch to commissioning	0x06
Table end of memory strategy setting values	

Config	Value
Start Download	0x01
Switch to commissioning then start download	0x02
Start Download then erase	0x03
Switch to commissioning , start download then erase	0x04
Cancel download	0x05
Erase	0x06
Stop logging	0x07
Table Download Setting values	

7.3.6.1.1 Examples:

The examples of all the OTAC will be shown as follow
The OTAC frame is splitted to bytes with "-" character and each byte is interpreted in decimal.

7.3.6.1.1.1 Example 1: Download file 0

The first example shows a Download OTAC with the following configurations:

- Index file = 0
- End of memory strategy is Stop log

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-5-208-1-1-0-0

General Datalogger OTAC frame				
Field Name	Size in bytes	index in bytes	Values	Additional information
MacId(Msb first)	8	0	244-184-94-0-166-230-0-0	F4B85E00A6E60000
OTAC Length	1	8	5	
OTAC Id	1	9	208	0xD0
End of memory management	1	10	1	Stop Log
Download setting	1	11	1	Start Download
Index first File to Download(Lsb first)	2	12	0-0	File index 0

7.3.6.2 Download response OTAC

The download sequence exchange is shown below:


Remote
Device
OTAC Start Download
Datalogger Status (Download_In_Progress)
Send First Frame (Ack requested, File 0, and Progression)
Send AckFirstFrame(File0)
Send Data (Ack not requested, File 0, Seq0, Progression)
Send Data (Ack not requested, File 0), Seq1, Progression)
Send Data (Ack requested, File 0, Seq2, Progression)
Send AckDataFrame (File0)
Send First Frame (Ack requested, File 1, Progression)
Send AckFirstFrame(File1)
Send Data (Ack Not requested.File 1 .Seq0.Progression)
Send Data (Ack Not requested, File 1, Seq1.Progression)
Send Data (Ack Not requested.File 1 .Seq2.Progression)
Send Data (Ack Not requested.File 1 .Seq3.Progression)
Send Data (Ack requested, File 1 .Seq4.Progression)
Send AckDataFrame(File1)
Send First Frame (Ack requested, File n.Progression)
Send AckFirstFrame(File n)
Send Data (Ack Not requested.File n .Seq 0.Progression
Send AckDataFrame(File n)
Datalogger Status(Download_End)

This OTAC is responsible of:

- Responding to download frames sent by the device Ack or NAck

The table below shows in details how the Download response OTAC frame should be organized to be interpreted by the device.

Download Response Otac frame				
Field Name	Size in bytes	index in bytes	Values	Additional information
MacId(Msb first)	8	0		
OTAC Length	1	8		
OTAC Id	1	9	0x92	
Response Id	1	10		See table Download responses values
File index (Lsb First)	2	11		
Frame Type	1	13		See table Download Frame type values

Config	Value
Acknowledgment	0x01
Not acknowledgement	0x02
Table Download Responses values	

Config	Value
First Frame	0x01
Data	0x02
Table Download frame type values	

7.3.6.2.1 Examples:

The examples of all the OTAC will be shown as follow
The OTAC frame is splitted to bytes with "-" character and each byte is interpreted in decimal.

7.3.6.2.1.1 Example 1: Send Acknowledgement of the first frame

The first example shows a Download response OTAC with the following configurations:

- File Index = 0
- Type frame = First frame

The example OTAC frame is the following:
244-184-94-0-166-230-0-0-5-146-1-0-0-1

Download Response OTAC frame				
Field Name	Size in bytes	index in bytes	Values	Additional information

MacId(Msb first)	8	0	244-184-94-0-166-230-0-0	
OTAC Length	1	8	5	
OTAC Id	1	9	146	0x92
Response Id	1	10	1	Acknowledgment
File index (Lsb First)	2	11	0-0	First Frame file 0
Frame Type	1	13	1	First frame

7.3.7 Other OTAC

7.3.7.1 Reset OTAC

This OTAC shall be sent when the remote need to restart the device.

OTAC Reset				
Field Name	Size in bytes	index in bytes	Values	Additional information
MacId(Msb first)	8	0		
OTAC Length	1	8	1	
OTAC Id	1	9	0xAB	

7.3.7.2 Request All profiles OTAC

This OTAC shall be sent when the remote need all profiles from the device.

OTAC Request All Profiles				
Field Name	Size in bytes	index in bytes	Values	Additional information
MacId(Msb first)	8	0		
OTAC Length	1	8	1	
OTAC Id	1	9	0xAD	

7.3.7.3 No More OTAC

This OTAC shall be sent when the remote send all the OTAC pending during a sleep cycle of the device. It informs the device that there is no more OTAC to be sent hence it goes to sleep again. If it is not sent the device goes to sleep again after a timeout.

OTAC No Pending OTAC				
Field Name	Size in bytes	index in bytes	Values	Additional information

MacId(Msb first)	8	0		
OTAC Length	1	8	1	
OTAC Id	1	9	0xAC	

8. FRAMES RELATED TO DATA ACQUISITION MODE

The broker receives data from devices on a set of topics and forwards that to subscribed devices on these topics.
The data consumer Connected to the same broker have to be able to SUBSCRIBE and parse the PUBLISH MQTT frame, the figure below explains the PUBLISH frame received from the broker at TCP level.

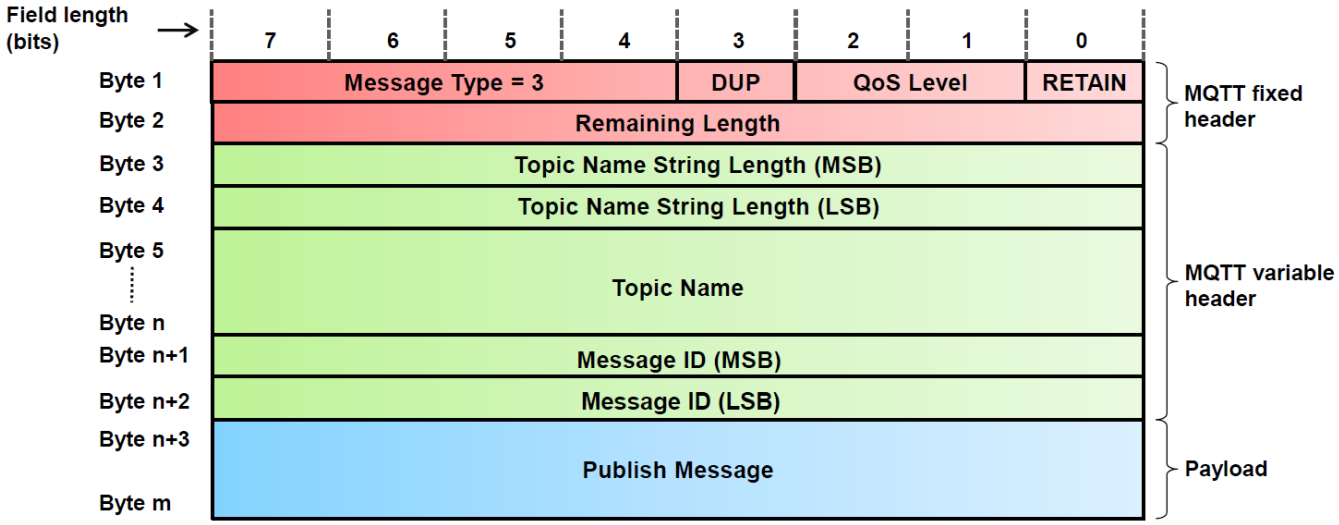


Figure 2: PUBLISH frame contents on TCP level

Message Id is only present in the PUBLISH message (Broker → Data consumer) if the QoS level > 0 (Embedded in the SUBSCRIBE frame sent earlier).
Different fields of this frame (Except Payload contents which are Beanair specified) are well documented in the MQTT official Specifications.
The payload content changes according to the frame nature and data acquisition mode, each frame is preceded with a *Device type* and an *Acquisition type* fields, each mode can be distinguished using the tables below.

Device type	*Value*	*Description*
AX_3D	*0x01*	*AX_3D device Id*
HI_INC_MONO	*0x02*	*HI_INC_MONO Device Id*
HI_INC_BI	*0x03*	*HI_INC_BI Device Id*
X_INC_MONO	*0x04*	*X_INC_MONO Device Id*
X _INC_BI	*0x05*	*X_INC_BI Device Id*
AX_3DS	*0x06*	*AX_3DS device Id*

Table 21: Different Beanair devices types Ids

Data Acquisition type	*Value*	*Description*
LDCDA mode	*0x01*	*The Id of the Low Duty Cycle Data Acquisition mode*
Alarm mode	*0x02*	*The Id of the Alarm Data Acquisition mode*
Streaming mode	*0x03*	*The Id of the Streaming Data Acquisition mode*

Shock Detection mode	*0x04*	*The Id of the Shock Detection mode*
LDC Math Result	*0x05*	*The Id of the Low Duty Cycle Math Result*
SET mode	*0x06*	*The Id of the SET (STREAMING WITH EVENT TRIGGER) mode*
Dynamic Math Result	*0x07*	*The Id of the Dynamic Math Result*

Table 22: Different Acquisition type Ids

8.1 LDCDA MODE

In LDCDA (Low Duty Cycle Data Acquisition) mode, the payload content of the PUBLISH format is as follows.

Data meaning			Size
Device Type			1 byte
Acquisition type (Default 0x01)			1 byte
Channel Id			1 byte
Date in Unix time format (LSB First)			4 bytes
Data sample measured (LSB First)	Byte[0] data bits		1 byte
	Byte[1] data bits		1 byte
	Byte[2]	Sign bit	8th bit
		data bits	7 bits

Table 23: LDCDA frame contents seen from data consumer side

After reading "**Data sample measured**" field, the user must perform the following calculation:

$$Decimal\ value = (-1)^{sign\ bit} * \frac{Remaining\ bits\ in\ decimal\ format}{1000}$$

8.2 ALARM MODE

In ALARM mode, the payload content of the PUBLISH format is as follows.

Data meaning			Size
Device Type			1 byte
Acquisition type (Default 0x02)			1 byte
Channel Id			1 byte
Date In Unix time format (LSB First)			4 bytes
Alarm status	0x00	No Alarm	1 byte
	0x01	Alarm Start	
	0x02	Alarm in progress	
	0x03	Alarm End	
Data sample measured (LSB First)	Byte[0] data bits		1 byte
	Byte[1] data bits		1 byte
	Byte[2]	Sign bit	8th bit

		data bits	7 bits

Table 24: ALARM frame contents seen from data consumer side

After reading "***Data sample measured***" field, the user must perform the following calculation:

$$Decimal\ value = (-1)^{sign\ bit} * \frac{Remaining\ bits\ in\ decimal\ format}{1000}$$

8.3 STREAMING MODE

In STREAMING mode, the payload content of the PUBLISH format is as follows, further description on how to use the frame contents are explained below.

	Data meaning				
Part 1: used to compute each data acquisition time	**Device Type**		1 byte		
	Acquisition type (Default 0x03)		1 byte		
	Reference time In Unix time format (LSB First)		4 bytes		
	Reference millisecond (LSB First)		2 bytes		
	Sampling frequency (LSB First)		2 bytes		
	Channels bitmap (LSB First)	Is channel 1 activated?	0th Bit	1st Byte	4 bytes
		Is channel 2 activated?	1st Bit		
		Is channel 3 activated?	2nd Bit		
		:	:		
		:	:		
		:	:	2nd Byte	
		:	:	3rd Byte	
		:	:	4th Byte	
		Is channel 32 activated ?	31th Bit		
	Frame Sequence Id (LSB First):(Begins from 0)		3 bytes		
	Number of data acquisitions per channel		2 bytes		
	Data Acquisition cycle		3 bytes		
	Data acquisition duration		3 bytes		
	Previous Number of data acquisitions per channel		2 bytes		
	Flags	Synchronization	1 bit		1 byte
		Future Use	7 bits		
	Network Quality (LQI)		1 byte		

Part 2: Data samples

Data Sample 1 of channel 1 (LSB First)	Byte[0] data bits		1st Sub Packet	1 byte	3 bytes
	Byte[1]			1 byte	
	Byte[2]	Sign bit		8th bit	1 byte
		data bits		7 bits	
:				:	
Data Sample 1 of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes	
Data Sample 2 of channel 1 (LSB First)			2nd Sub	3 bytes	
Data Sample 2 of next channel (LSB First)				3 bytes	
:				:	
Data Sample 2 of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes	
:				:	
Data Sample M of channel 1 (LSB First)			Mth Sub Packet	3 bytes	
Data Sample M of next channel (LSB First)				3 bytes	
:				:	
Data Sample M of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes	

Table 25: STREAMING frame contents seen from data consumer side

- To meet the streaming mode high frequency publishing, the data is compacted in a single packet and sent to the broker. The data consumer has to parse the frame from Part 2 and compute its occurrence time using Part 1.
- To compute current SubPacket time use the following formula:

$$T_{SubPacket} = Reference\ Time\ Second + \frac{\text{Reference Millisecond}}{1000} + \left(\frac{1}{Sampling\ frequency}\right) * SubPacket\ Index$$

Where

$$SubPacket\ Index = (Frame\ Sequence\ Id * Previous\ Number\ of\ data\ acquisitions\ per\ channel) + Current\ SubPacket\ row$$

- The channels bitmap is important during parsing to know to what channel the data belongs to.
- SubPacket Index is the index of the current SubPacket in the whole streaming event (from frame Id 0 when the stream started).
- During parsing, the *Current SubPacket row* is the index of the SubPacket in the current frame.
- To obtain a meaningful decimal value, the "Data *Sample i of channel j"* field must be used as follows:

$$Decimal\ value = (-1)^{sign\ bit} * \frac{Remaining\ bits\ in\ decimal\ format}{1000}$$

8.4 S.E.T MODE

In S.E.T mode, the payload content of the PUBLISH format is as follows, further description on how to use the frame contents are explained below.

Part 1: used to compute each data acquisition

Data meaning		Size		
Device Type		1 byte		
Acquisition type (Default 0x06)		1 byte		
Reference time In Unix time format (LSB First)		4 bytes		
Reference millisecond (LSB First)		2 bytes		
Sampling frequency (LSB First)		2 bytes		
Channels bitmap (LSB First)	Is channel 1 activated?	0th Bit	1st Byte	4 bytes
	Is channel 2 activated?	1st Bit		
	Is channel 3 activated?	2nd Bit		
	.....	.....		
	.....	.....		
	.....	:	2nd Byte	
	.....	:	3rd Byte	
	.....		4th Byte	
	Is channel 32 activated?	31th Bit		
Frame Sequence Id (LSB First): (Begins from 0)		3 bytes		
Number of data acquisitions per channel		2 bytes		
Data Notification cycle		3 bytes		
Data acquisition duration		3 bytes		
Previous Number of data acquisitions per channel		2 bytes		
Flags	Synchronization	1 bit		1 byte
	Future Use	7 bits		
Network Quality (LQI)		1 byte		
Alarm Status		1 byte		

Part 2: Data samples

Data Sample 1 of channel 1 (LSB First)	Byte[0] data bits		1st Sub Packet	1 byte		3 bytes
	Byte[1]			1 byte		
	Byte[2]	Sign bit		8th bit	1 byte	
		data bits		7 bits		
Data Sample 1 of next channel (LSB First)				3 bytes		
:				:		
Data Sample 1 of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes		
Data Sample 2 of channel 1 (LSB First)			2nd Sub Packet	3 bytes		
Data Sample 2 of next channel (LSB First)				3 bytes		
:				:		
Data Sample 2 of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes		
:				:		
Data Sample M of channel 1 (LSB First)			Mth Sub Packet	3 bytes		
Data Sample M of next channel (LSB First)				3 bytes		
:				:		
Data Sample M of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes		

Table 26: S.E.T frame contents seen from data consumer side

- To meet the SET mode high frequency publishing we used the same data compaction technique as the Streaming mode and kept the same formulas.
- To compute current SubPacket time use the following formula:

$$T_{SubPacket} = Reference\ Time\ Second + \frac{\text{Reference Millisecond}}{1000} + \left(\frac{1}{Sampling\ frequency}\right) * SubPacket\ Index$$

Where

$$SubPacket\ Index = (Frame\ Sequence\ Id * Previous\ Number\ of\ data\ acquisitions\ per\ channel) + Current\ SubPacket\ row$$

- To obtain a meaningful decimal value, the "Data *Sample i of channel j"* field must be used as follows:

$$Decimal\ value = (-1)^{sign\ bit} * \frac{Remaining\ bits\ in\ decimal\ format}{1000}$$

8.5 SHOCK DETECTION

In Shock Detection mode, the payload content of the PUBLISH format is as follows, further description on how to use the frame contents are explained below.

Part 1: used to compute each data acquisition time

Data meaning		Size		
Device Type		1 byte		
Acquisition type (Default 0x04)		1 byte		
Reference time In Unix time format (LSB First)		4 bytes		
Reference millisecond (LSB First)		2 bytes		
Sampling frequency (LSB First)		2 bytes		
Channels bitmap (LSB First)	Is channel 1 activated?	0th Bit	1st Byte	4 bytes
	Is channel 2 activated?	1st Bit		
	Is channel 3 activated?	2nd Bit		
	.....	.....		
	.....	.....		
	.....	:	2nd Byte	
	.....	:	3rd Byte	
	.....		4th Byte	
	Is channel 32 activated?	31th Bit		
Frame Sequence Id (LSB First): (Begins from 0)		3 bytes		
Number of data acquisitions per channel		2 bytes		
Data Monitoring cycle		3 bytes		
Data acquisition duration		3 bytes		
Previous Number of data acquisitions per channel		2 bytes		
Flags	**Synchronization**	1 bit		1 byte
	Future Use	7 bits		
LQI (Network Quality)		1 bytes		
Shock source		1 byte		
X Axis First data (2's complement)		2 bytes		
Y Axis First data (2's complement)		2 bytes		
Z Axis First data (2's complement)		2 bytes		

Part 2: Data samples

Data Sample 1 of channel 1 (LSB First)	Byte[0] (data byte)		1st Sub Packet	1 byte	
	Byte[1] (data byte)			1 byte	
	Byte[2]	Sign bit		8th bit	1 byte
		data bits		7 bits	
Data Sample 1 of next channel (LSB First)				3 bytes	
:				:	
Data Sample 1 of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes	
Data Sample 2 of channel 1 (LSB First)			2nd Sub Packet	3 bytes	
Data Sample 2 of next channel (LSB First)				3 bytes	
:				:	
Data Sample 2 of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes	
:			:	:	
Data Sample M of channel 1 (LSB First)			Mth Sub Packet	3 bytes	
Data Sample M of next channel (LSB First)				3 bytes	
:				:	
Data Sample M of channel n (last one present in the "channels bitmap" field) (LSB First)				3 bytes	

Table 27: Shock detection frame contents seen from data consumer side

- To meet the Sock detection mode high frequency publishing we used almost the same data compaction technique as the Streaming mode and kept the same formula.
- To compute current SubPacket time use the following formula:

$$T_{SubPacket} = Reference\ Time\ Second + \frac{\text{Reference Millisecond}}{1000} + \left(\frac{1}{Sampling\ frequency}\right) * SubPacket\ Index$$

Where,

$$SubPacket\ Index = (Frame\ Sequence\ Id * Previous\ number\ of\ data\ acquisitions\ per\ channel) + Current\ SubPacket\ row$$

- To obtain a meaningful decimal value, the "Data *Sample i of channel j"* field must be used as follows:

$$Decimal\ value = (-1)^{sign\ bit} * \frac{Remaining\ bits\ in\ decimal\ format}{1000}$$

Channel name	Channel Id
channel Z	0x00
channel X	0x01

channel Y	0xx2

Note;

- Shock source will hold the channel id that caused the shock:
- For each axis first data field (X, Y and Z) it holds the data sample at the moment of the shock

8.6 LDC MATH RESULT

In LDC Math Result, which is available only in the LDC mode, the payload content of the PUBLISH format is as follows.

Data meaning			Size
Device Type			1 byte
Acquisition type (Default 0x05)			1 byte
Channel Id			1 byte
Math Result Id			1 byte
Event In Unix time format (LSB First)			4 bytes
Start In Unix time format (LSB First)			4 bytes
End In Unix time format (LSB First)			4 bytes
Data sample (LSB First)	Byte[0] data bits		1 byte
	Byte[1] data bits		1 byte
	Byte[2]	Sign bit	8th bit
		data bits	7 bits

Table 28: LDC Math Result frame contents seen from data consumer side

- After reading "**Data sample measured**" field, the user must perform the following calculation:

$$Decimal\ value = (-1)^{sign\ bit} * \frac{Remaining\ bits\ in\ decimal\ format}{1000}$$

- The different Math result Ids:

Math result type	Math result Id	Description
Maximum	0x01	Represent the maximum measurement in a specific period of time X.
Minimum	0x02	represent the minimum measurement in a specific period of time X.
Average	0x03	It's the average value calculated from measurements collected in a specific period of time X.

(Note: The period of time X is the Math Notif cycle.)

- If the math result represent a maximum or minimum then both of "Start" and "End" fields are empty.

- If the math result represent average then only the “Event” field is empty.

8.7 DYNAMIC MATH RESULT

The Dynamic Math Result is for the Streaming (burst and one shot), SET and shock detection modes, the payload content of the PUBLISH format is as follows. Further description on how to use the frame contents are explained below

Data meaning		Size		
Device Type		1 byte		
Acquisition type (Default 0x07)		1 byte		
Acquisition mode		1 byte		
Acquisition frequency		2 bytes		
Acquisition cycle		3 bytes		
Acquisition duration		3 bytes		
Future use (Default 0x55)		1 byte		
Channels bitmap (LSB First)	Is channel 1 activated?	0th Bit	1st Byte	4 bytes
	Is channel 2 activated?	1st Bit		
	:	:		
	….	….	2nd Byte	
	….	….	3rd Byte	
	:	:	4th Byte	
	Is channel 31 activated?	30th Bit		
	Is channel 32 activated?	31th Bit		
Start date of math result	Years	First 2 bytes (LSB first)		9 bytes
	Months	1 byte		
	Days	1 byte		
	Hours	1 byte		
	Minutes	1 byte		
	Seconds	1 byte		
	Milliseconds	2 bytes (LSB first)		
End date of math result		9 bytes		
Number of data per channel		2 bytes		

Frame Sequence Id of the minimum value of channel 1 (LSB First)	Channel 1	4 bytes
Index of the minimum value of channel 1 (LSB First)		2 bytes
Minimum value of channel 1 (last bit for sign)		3 bytes
Frame Sequence Id of the maximum value of channel 1 (LSB First)		4 bytes
Index of the maximum value of channel 1 (LSB First)		2 bytes
Maximum value of channel 1 (last bit for sign)		3 bytes
Average value of channel 1 (last bit for sign)		3 bytes
:	...	:
Frame Sequence Id of the minimum value of channel n (LSB First)	Channel n	4 bytes
Index of the minimum value of channel n (LSB First)		2 bytes
Minimum value of channel n (last bit for sign)		3 bytes
Frame Sequence Id of the maximum value of channel n (LSB First)		4 bytes
Index of the maximum value of channel n (LSB First)		2 bytes
Maximum value of channel n (last bit for sign)		3 bytes
Average value of channel n (last bit for sign)		3 bytes

Table 29: Dynamic Math Result frame contents seen from data consumer side

- Acquisition mode is Streaming without the option(streaming continue ,burst , one shot) or SSD or S.E.T.
- To compute the Minimum or the Maximum event time use the following formula:

$$Event\ Time = reference\ start\ time\ + \left(\frac{1}{\text{Acquisition } frequency}\right) * full\ Index$$

Where

  - $full\ Index = (Frame\ Sequence\ Id * Number\ of\ data\ acquisitions\ per\ channel\ ) + Index$
  - Reference start time is the timestamp of the start date of math result in millisecond

- Perform the same calculation as the LDC Math result for the minimum, maximum and average values:

$$Decimal\ value = (-1)^{sign\ bit} * \frac{Remaining\ bits\ in\ decimal\ format}{1000}$$

8.8 DIAGNOSTIC

The payload content of the PUBLISH Diagnostic message is as follows, further description on how to use the frame contents are explained below. The topic name is MACID/UPDATE

Device diagnostic header

Data meaning			Size
Reserved			17 bytes
Date	Year	2 bytes	7 bytes
	Month	1 byte	
	Day	1 byte	
	Hour	1 byte	
	Minute	1 byte	
	Second	1 byte	
Diagnostic type			1 byte
reserved			1 byte
PER (Packet Error Rate)			2 bytes
LQI (Network Quality)			1 bytes
Reserved			2 bytes
Diagnostic Options			2 bytes
Internal Temperature			2 bytes
Reserved			2 bytes
Datalogger Free Memory			1 byte
Energy harvester Status			1 byte
Reserved			3 bytes
Battery voltage			2 bytes
Number of available sensor channel			1 byte

Sensor diagnostic

Sensor Status Bitmap	First Sensor Status	Bit 0	1: SC 0 : SDC	1 byte	(Number of sensor channel) bytes
		Bit 1	1 : SE 0 :SDS		
		Bit 2	1 : SF 0 :SWW		
		Bit 3 to Bit 7	Not used		
	.....			1 byte	
	Last Sensor Status	Bit 0	1: SC 0 : SDC		
		Bit 1	1 : SE 0 :SDS		
		Bit 2	1 : SF 0 :SWW		
		Bit 3 to Bit 7	Not used		

Table 30: Diagnostic frame contents seen from data consumer side

- **SC** : *Sensor connected*
- **SDC** : *sensor disconnected*
- **SE** : *Sensor Enabled*
- **SDS** : *Sensor Disabled*
- **SF** : *Sensor Fail*
- **SWW** : *Sensor Working Well*

9. PROFILES OVER_MQTT FRAME CONTENTS

All profiles are published on the MACID/CREATE topic.

9.1 GENERAL PROFILE

This profile contains the following information:

- BeanDevice® MAC ID
- IP Address and DHCP client Status (Enabled , Disabled)
- BeanDevice® Hardware Version
- BeanDevice® Software Version
- Data acquisition capability
- Number profile layers to be transmitted after this profile including the general profile
- Profile ID of the Profile to be sent in order (LSB = profile id of the first profile)

The profile data frame comes as follow:

GENERAL PROFILE				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape® Header	17	0		
Profile Id	1	17	0x02	
Future Use	2	18		
MAC Id (Msb First)	8	20		
Future use	1	28		
IP Config Mode	1	29		see IP Config Mode table
IP Address	6	30		
Hardware version	1	36		
Software version	1	37		
WSN Stack version	1	38	0x10	
Data acquisition capability	1	39		See data acquisition capability table
Profiles number	1	40		Contains number of profile will be sent
Profiles Id	26	41		

Ip Config Mode table	
IP config value	**Description**
0	Static IP
1	Dynamic IP

Data acquisition capability table	
Bit number	**Description**
0	1: Alarm & low duty cycle are supported 0: Alarm & low duty cycle are not supported
1	1: Streaming and Set mode and commissioning are supported 0: Streaming and Set mode and commissioning are not supported
2	1: Shock detection is supported 0: Shock detection is not supported

9.2 DAQ PROFILE

This Profile contains the following information:

- Data acquisition mode (streaming, Low duty cycle, SET mode…)
- DAQ options (TX for data transmission, TX & Log for data transmission and data logging, Standalone, Streaming Options…)
- Sampling rate and max sampling rate
- Data acquisition cycle
- Transmission ratio and Max Transmission ratio
- Data aging of the store and forward
- Data acquisition duration

The profile data frame is shown in the table below:

DATA ACQUISITION PROFILE				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape® Header	17	0		
Profile Id	1	17	0x10	
Daq Mode	1	18		see Daq mode table
Daq options	2	19		see Daq option table
Future Use	2	21		
Daq Cycle(Lsb first)	3	23		
Max TX Ratio	1	26		

TX Ratio	1	27		
Daq Duration(Lsb first)	3	28		
Max sampling rate(Lsb first)	3	31		
Sampling rate(Lsb first)	3	34		
Future Use	3	37		
Store and forward data aging(Lsb first)	2	40		
Future Use	2	42		

Daq mode	value
Commissioning	0x01
low duty cycle	0x02
Streaming	0x03
Alarm	0x04
SET mode	0x05
Shock Detection	0x06
Daq mode Table	

Daq options bit	Signification
0	Datalogger bit: 1 : datalogger enabled 0 : datalogger disabled
1	Store and forward bit 1 : Store and forward enabled, 0 :Store and forward disabled
2	Streaming (bit2,bit3,bit4): Streaming Continuous:(1,0,0) Streaming one shot:(0,1,0) Streaming burst:(1,1,0)
3	
4	
5	Transmission TX bit: 1 : TX enabled,0 : TX disabled
6	Stand Alone bit: 1 : Stand Alone enabled 0 : Stand Alone disabled
7->15	Future use
Daq options table	

9.3 SYSTEM STATUS PROFILE

This Profile contains the following information:

- Activity led status (Enabled/Disabled)
- OTAC status(locked/unlocked)
- Power source
- Power mode (Active/Sleep)
- Diagnostic cycle
- Listening cycle

The profile data frame is shown in the table below:

System Status Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0		
Profile Id	1	17	0x21	
System Status	1	18		see system status
Power status	1	19		see power status table
Diagnostic cycle	1	20		
Network Listening cycle(lsb first)	2	21		

Config bit	Signification
0	OTAC Status bit: 1 = OTAC locked, 0 = OTAC unlocked
1	Activity Led bit, 1 = Activity Led enabled, 0 = Activity Led disabled
2->7	Future use
System status	

Bit	Daq mode	value
4 Low bits	Active mode	0xY1(Y any number)
	Sleep with network listening	0xY3(Y any number)
	Standby low battery	0xY5(Y any number)
	Standby	0xY6(Y any number)
	next 4 bits are for power source type	
4 High bits	External power supply	0x1Y(Y any number)
	Internal Battery	0x2Y(Y any number)
	Energy harvesting	0x3Y(Y any number)
	Power mode Table	

9.4 WIRELESS LINK PROFILE

This Profile contains the following information:

- SSID
- Wi-Fi authentication mode

The profile data frame is shown in the table below:

Wireless link Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0		
Profile Id	1	17	0x31	
Future Use	1	18		
Wi-Fi authentication mode	1	19		see authentification table

SSID length	1	20		
SSID	30	21		
Future Use	2	51		

Authentication type table	
Authentication type	Value
Open	0
WEP	1
WPA	2
WPA2	3

9.5 MAIN SENSOR PROFILE

This Profile contains the following information:

- Internal sensor profile
- Number of channels
- Shock available acceleration range
- Current Shock acceleration range
- Shock sampling rate
- Shock notification delay
- Shock threshold

The profile data frame is shown in the table below:

Main Sensor PROFILE				
Name field	Size in byte	Byte index	Value	Additional information
BeanScape Header	17	0		
Profile Id	1	17	0x42	
Number of channels	1	18		
Internal sensor profile	1	19		see internal sensor profile table
Shock Available acceleration range	1	20		see available acceleration range table
Shock acceleration range(Lsb First)	2	21		
Shock Available Sampling rate	1	23		
Shock Sampling rate((Lsb First)	2	24		
Shock notification delay	2	26		

Future Use	1	28		
Shock threshold(Lsb First)	2	29		
Future Use	4	31		

Sensor Type	Value
AX3D	0x01
Hi Inc mono Axial	0x02
Hi Inc Bi Axial	0x03
Xinc Mono	0x04
Xinc Bi	0x05
AX3DS	0x06
Internal sensor profile table	

Range Type	value
2G-4G-6G-8G-16G	0x01
6G-12G-24G	0x02
2G-4G-8G	0x03
100G-200G-400G	0x04
Available acceleration range table	

9.6 CHANNEL PROFILE

This Profile contains the following information:

- Sensor range
- Channel Id
- Alarm threshold levels(H1,H2,L1,L2)
- Channel Status
- Calibration date
- Calibration values
- Unit

The profile data frame is shown in the table below:

Channel PROFILE				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0		
Profile Id	1	17	0x82	
Channel Id	1	18		0:ChannelZ 1:ChannelX 2:ChannelY 3:INCX 4:INCY
Sensor range	1	19		in G
Alarm threshold H1(float LSB first)	4	20		
Alarm threshold H2(float LSB first)	4	24		

Alarm threshold L1(float LSB first)	4	28		
Alarm threshold L2(float LSB first)	4	32		
Channel status	1	36		Bit 0 : Channel Status (1:Enabled/0:Disabled) Bit 1 : Sensor Calibration(1:calibrated/0:Not calibrated)
Calibration date(Year)	2	37		
Calibration date(Month)	1	39		
Calibration date(Day)	1	40		
Calibration date(Hour)	1	41		
Calibration date(Minute)	1	42		
Calibration date(Second)	1	43		
Offset(float LSB first)	4	44		
Ratio(float LSB first)	4	48		
Future Use	16	52		
Unit	1	68		7:mg,8:Deg

9.7 DATALOGGER STATUS PROFILE

This Profile contains the following information:

- Datalogger status
- End memory strategy
- Datalogger current action
- Free memory space

The profile data frame is shown in the table below:

Datalogger status				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0		
Profile Id	1	17	0xD0	
Datalogger status	1	18		see datalogger status table
End Of memory strategy	1	19		see end of memory strategy table

Datalogger status	Value
Not Init	0x01
Initializing	0x02
Ready	0x03
Ready download only	0x04
Logging	0x05
Stopped	0x06
Failure	0x07
Erase in progress	0x08
Memory Empty	0x09
Memory full	0x10
Download in progress	0x11
canceled	0x12
Download End	0x13
Stand Alone	0x14
Datalogger status table	

Config	Value
Stop log	0x01
Stop keep Daq	0x02
Stop Go to Commissioning	0x03
Stop auto download erase reset Daq	0x04
Stop auto download switch to commissioning	0x05
Stop auto download erase switch to commissioning	0x06
Table end of memory strategy setting values	

Datalogger current action	1	20		See table Datalogger current action values
Future Use	4	21		
Free memory space	1	25		0→200 /0:full , 200 :empty

9.8 CLOCK PROFILE

This Profile contains the following information:

- Time zone
- NTP server
- NTP port

The profile data frame is shown in the table below:

Config	Value
Start Download	0x01
Switch to commissioning then start download	0x02
Start Download then erase	0x03
Switch to commissioning , start download then erase	0x04
Cancel download	0x05
Erase	0x06
Stop logging	0x07

Table Datalogger current action values

Ntp Config Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape® Header	17	0		
Profile Id	1	17	0x91	
Time zone	2	18		in minutes
Future Use	5	20		
NTP Port	2	25		
DNS Enabled/Disabled	1	27		0: DNS disabled 1: DNS enabled
NTP server IP	4	28		in case DNS disabled
NTP server Name length	1	32		in case DNS enabled
NTP server URL	31	33		in case DNS enabled

9.9 MQTT PROFILES

MQTT profiles published via MQTT come with the following structure:

Mqtt module status				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0		
Mqtt Profile Payload	1	17		

The Mqtt Profile Payload is listed in section 7.

10. APPENDIX 1: EXAMPLES

10.1 BEANDEVICE® WILOW VERSION PROFILE EXAMPLE

Device Version PROFILE Example				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	**0x42-0x4f-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xff-0xFE-0x19-0x32**	
Profile Id	1	17	**0x02**	
Future Use	2	18	**0x01-0x00**	
MAC Id (Msb First)	8	20	**0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00**	MAC ID = F4B85E00A4760000
Future use	1	28	**0x00**	
IP Config Mode	1	29	**0x01**	dynamic IP set
IP Address	6	30	**0xC0-0xA8-0x01-0x02-0x00-0x00**	0xC0-0xA8-0x01-0x02 = 192.168.1.2 0x00-0x00 for future use
Hardware version	1	36	**0x20**	V2R0
Software version	1	37	**0x29**	V2R9
WSN Stack version	1	38	***0x10***	
Data acquisition capability	1	39	**0x07**	LDC | STR |STSD (all modes are supported by this device)
Profile number	1	40	**0x15**	this device contains 21 other profiles (other than this one)

Whole frame
Buffer[67]> 0x42-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x32-0x02-0x01-0x00-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x00-0x01-0xC0-0xA8-0x01-0x02-0x00-0x00-0x20-0x29-0x10-0x07-0x15-0x02-0x10-0x21-0x31-0x42-0x82-0x82-0x82-0xC0-0xD0-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x91-0x00-0x00-0x00-0x00-0x00-

Profiles Id	26	41	0x02-0x10-0x21-0x31-0x42-0x82-0x82-0x82-0xC0-0xD0-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x90-0x91-0x00-0x00-0x00-0x00-0x00-	List of the id of each profile that will be sent in order 0x00 means

10.2 BEANDEVICE® WILOW WIRELESS LINK PROFILE EXAMPLE

Wireless link Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x34-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x24	
Profile Id	1	17	0x31	
Future Use	1	18	0x01	
Wi-Fi authentication mode	1	19	0x03	WPA2
SSID length	1	20	0x09	
SSID	30	21	0x5F-0x6C-0x6F-0x62-0x61-0x6C-0x6E-0x65-0x74-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00	_lobalnet
Future Use	2	51	0x00-0x00	

Whole frame
Buffer[53]> 0x34-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x24-0x31-0x01-0x03-0x09-0x5F-0x6C-0x6F-0x62-0x61-0x6C-0x6E-0x65-0x74-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-

10.3 BEANDEVICE® WILOW SYSTEM STATUS PROFILE EXAMPLE

System Status Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	**0x16-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x06**	
Profile Id	1	17	**0x21**	
System Status	1	18	**0x02**	Led activated ,OTAC unlocked
Power status	1	19	**0x11**	Power mode = Active, Power source = External power supply
Diagnostic cycle	1	20	**0x01**	Diagnostic coefficient = 1
Network Listening cycle(lsb first)	2	21	**0x3C-0x00**	Network listening cycle = 60s (not used here because it is in Active mode)

Whole frame
Buffer[23]> **0x16-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x06-0x21-0x02-0x11-0x01-0x3C-0x00-**

10.4 BEANDEVICE® WILOW DATA ACQUISITION PROFILE EXAMPLE

DATA ACQUISITION PROFILE				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x2B-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x1B	
Profile Id	1	17	0x10	
Daq Mode	1	18	0x02	Low duty cycle
Daq options(Lsb First)	2	19	0x21-0x00	TX & Log
Future Use	2	21	0x00-0x00	
Daq Cycle(Lsb first)	3	23	0x01-0x00-0x00	Daq cycle 1second
Max TX Ratio	1	26	0x09	Max TX ratio =9
TX Ratio	1	27	0x01	Current TX ratio =1
Daq Duration(Lsb first)	3	28	0x00-0x00-0x00	Not used in low duty cycle
Max sampling rate(Lsb first)	3	31	0xF4-0x01-0x00	500Hz
Sampling rate(Lsb first)	3	34	0xE8-0x03-0x00	Not used in low duty cycle
Future Use	3	37	0x00-0x00-0x00	
Store and forward data aging(Lsb first)	2	40	0x00-0x00	data aging = 0ms
Future Use	2	42	0x10-0x0E	

Whole frame
Buffer[44]> 0x2B-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x1B-0x10-0x02-0x21-0x00-0x00-0x00-0x01-0x00-0x00-0x09-0x01-0x00-0x00-0x00-0xF4-0x01-0x00-0xE8-0x03-0x00-0x00-0x00-0x00-0x00-0x00-0x10-0x0E-

10.5 BEANDEVICE® WILOW MAIN SENSOR PROFILE EXAMPLE

Main Sensor PROFILE				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x22-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x12	
Profile Id	1	17	0x42	
Number of channels	1	18	0x03	The device contains 3 channels
Internal sensor profile	1	19	0x01	the device is Ax3D
Shock Available acceleration range	1	20	0x01	acceleration range type 2G-4G-6G-8G-16G
Shock acceleration range(Lsb first)	2	21	0x10-0x00	16G
Shock Available Sampling rate	1	23	0x01	25Hz,50Hz,100Hz,400Hz,800Hz,1600Hz
Shock Sampling rate(lsb first)	2	24	0x20-0x03	800Hz
Shock notification delay	2	26	0x00-0x00	
Future Use	1	28	0x00	
Shock threshold	2	29	0xD0-0x07	2000mg
Future Use	4	31	0xD0-0x07-0xD0-0x0	

Whole frame
Buffer[35]> 0x22-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x12-0x42-0x03-0x01-0x01-0x10-0x00-0x01-0x20-0x03-0x00-0x00-0x00-0xD0-0x07-0xD0-0x07-0xD0-0x07-

10.6 BEANDEVICE® WILOW CHANNEL PROFILE EXAMPLE

Channel PROFILE				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x44-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x34	
Profile Id	1	17	0x82	
Channel Id	1	18	0x00	0:ChannelZ
Sensor range	1	19	0x02	-2G/+2G
Alarm threshold H1(float LSB first)	4	20	0x1F-0x85-0xAB-0x3F	1.34
Alarm threshold H2(float LSB first)	4	24	0x0A-0xD7-0x63-0x3F	0.89
Alarm threshold L1(float LSB first)	4	28	0x29-0x5C-0x0F-0xBF	-0.56
Alarm threshold L2(float LSB first)	4	32	0xF6-0x28-0xBC-0xBF	-1.47
Channel status	1	36	0x03	Channel enabled and calibrated
Calibration date(Year)	2	37	0xE2-0x07	calibration date: 06/07/2018 10h8min15sec
Calibration date(Month)	1	39	0x07	
Calibration date(day)	1	40	0x06	
Calibration date(Hour)	1	41	0x0A	
Calibration date(Minute)	1	42	0x08	
Calibration date(Second)	1	43	0x0E	
Offset(float LSB first)	4	44	0xAE-0x47-0x21-0x3F	0.63
Ratio(float LSB first)	4	48	0x00-0x00-0x20-0xC0	-2.5

Future Use	16	52	0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00	
Unit	1	68	0x07	g

Whole frame
Buffer[69]> 0x44-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x34-0x82-0x00-0x02-0x1F-0x85-0xAB-0x3F-0x0A-0xD7-0x63-0x3F-0x29-0x5C-0x0F-0xBF-0xF6-0x28-0xBC-0xBF-0x03-0xE2-0x07-0x07-0x06-0x0A-0x08-0x0E-0xAE-0x47-0x21-0x3F-0x00-0x00-0x20-0xC0-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x07

10.7 BEANDEVICE® WILOW DATALOGGER STATUS EXAMPLE

Datalogger status				
Name field	Size in byte	Byte index	Value	Additional information
BeanScape Header	17	0	0x19-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x09	
Profile Id	1	17	0xD0	
Datalogger status	1	18	0x02	Initializing
End Of memory management	1	19	0x03	Stop Go to Commissioning
Memory download setting	1	20	0xFF	None
Future Use	4	21	0x00-0x00-0x00-0x00	
Free memory space	1	25	0xC3	195(96%)

Whole frame
Buffer[26]> 0x19-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x09-0xD0-0x02-0x03-0xFF-0x00-0x00-0x00-0x00-0xC3-

10.8 BEANDEVICE® WILOW MQTT MODULE STATUS EXAMPLE

Mqtt module status				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x14-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x04	
Profile Id	1	17	0x90	
Mqtt sub Id	1	18	0x02	Module Status
Mqtt status	1	19	0x08	Connected
fixed in Mqtt	1	20	0x00	

Whole frame
Buffer[21]> 0x14-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x04-0x90-0x02-0x08-0x00-

10.9 BEANDEVICE® WILOW MQTT CLIENT ID & KEEP ALIVE PROFILE EXAMPLE

Mqtt Client Id & Keep Alive profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x2E-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x1E	
Profile Id	1	17	0x90	
Mqtt Sub Id	1	18	0x03	Client Id and Keep Alive sub profile
Keep Alive interval in seconds	2	19	0x3C-0x00	60 seconds
Mqtt Protocol Version	1	21	0x04	Mqtt V3.1.1
Auto generation Client	1	22	0x01	Auto generation enabled
Client Id length	1	23	0x17	17
Client Id	23	24	0x57-0x49-0x4C-0x4F-0x34-0x35-0x39-0x34-0x38-0x36-0x31-0x35-0x33-0x30-0x36-0x39-0x36-0x39-0x37-0x37-0x38-0x31-0x39	WILO4594861530696977819

Whole frame
Buffer[47]> 0x2E-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x1E-0x90-0x03-0x3C-0x00-0x04-0x01-0x17-0x57-0x49-0x4C-0x4F-0x34-0x35-0x39-0x34-0x38-0x36-0x31-0x35-0x33-0x30-0x36-0x39-0x36-0x39-0x37-0x37-0x38-0x31-0x39-

10.10 BEANDEVICE® WILOW MQTT BROKER DETAILS PROFILE EXAMPLE

Mqtt Broker Details Profile				
Name field	Size in byte	Byte index	Value	Additional information
BeanScape Header	17	0	0x4C-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x3C	
Profile Id	1	17	0x90	
Mqtt Sub Id	1	18	0x04	Mqtt Broker Details sub profile
Broker Port	2	19	0x5B-0x07	1883
Broker DNS	1	21	0x00	DNS is disabled
Broker IP	4	22	0xC0-0xA8-0x01-0xFC	192.168.1.252
Broker DNS Length	1	26	0x00	0
Broker DNS	50	27	0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00	all bytes are null because DNS is disabled

Whole frame
Buffer[77]> 0x4C-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x3C-0x90-0x04-0x5B-0x07-0x00-0xC0-0xA8-0x01-0xFC-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-

10.11 BEANDEVICE® WILOW MQTT USER NAME & PASSWORD PROFILE EXAMPLE

Mqtt User Name & Password Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x7A-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x6A	
Profile Id	1	17	0x90	
Mqtt Sub Id	1	18	0x05	
User Name Enabled/Disabled	1	19	0x01	Enabled
Password Enabled/Disabled	1	20	0x01	Enabled
User Name Length	1	21	0x07	
User Name	50	22	0x42-0x65-0x61-0x6E-0x41-0x69-0x72-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00	BeanAir
Password Length	1	72	0x0B	11
Password	50	73	0x6D-0x71-0x74-0x74-0x62-0x65-0x61-0x6E-0x61-0x69-0x72-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-	mqttbeanair

Whole frame
Buffer[123]> 0x7A-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x6A-0x90-0x05-0x01-0x01-0x07-0x42-0x65-0x61-0x6E-0x41-0x69-0x72-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x0B-0x6D-0x71-0x74-0x74-0x62-0x65-0x61-0x6E-0x61-0x69-0x72-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-

10.12 BEANDEVICE® WILOW MQTT WILL TOPIC PROFILE EXAMPLE

Mqtt Will Topic Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x7B-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x6B	
Profile Id	1	17	0x90	
Mqtt Sub Id	1	18	0x06	
Will Enabled/Disabled	1	19	0x01	
Will retain flag Enabled/Disabled	1	20	0x00	
Will QoS	1	21	0x00	
Will topic Length	1	22	0x14	
Will topic	50	23	0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2F-0x4C-0x57-0x54-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00	F4B85E00A4760000/LWT
Will message Length	1	73	0x04	
Will message	50	74	0x44-0x45-0x41-0x44-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-	DEAD

Whole frame

```
Buffer[124]> 0x7B-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x6B-0x90-0x06-0x01-0x00-0x00-0x14-0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2F-0x4C-0x57-0x54-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x04-0x44-0x45-0x41-0x44-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-
```

10.13 BEANDEVICE® WILOW MQTT STREAMING TOPIC PROFILE EXAMPLE

Mqtt Streaming Topic Profile				
Name field	Size in byte	Byte index	Value	Additional information
BeanScape Header	17	0	0x4F-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x3F	
Profile Id	1	17	0x90	
Mqtt Sub Id	1	18	0x07	Publish Topic
Network Id	2	19	0x01-0x02	
Channel number	1	21	0xFA	
Publish status	1	22	0x01	Publish enabled
Future Use	1	23	0x00	
Topic Name length	1	24	0x1A	26
Mqtt publish fix header	3	25	0x30-0x91-0x01	
Reserved	2	28	0x00-0x1A	
Topic Name	50	30	0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2F-0x53-0x54-0x52-0x45-0x41-0x4D-0x49-0x4E-0x47-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-	F4B85E00A4760000/STREAMING

Whole frame

Buffer[80]> 0x4F-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x3F-0x90-0x07-0x01-0x02-0xFA-0x01-0x00-0x1A-0x30-0x91-0x01-0x00-0x1A-0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2F-0x53-0x54-0x52-0x45-0x41-0x4D-0x49-0x4E-0x47-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-

10.14 BEANDEVICE® WILOW MQTT LDC PROFILE EXAMPLE

Mqtt ldc Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x4F-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x3F	
Profile Id	1	17	0x90	
Mqtt Sub Id	1	18	0x07	
Network Id	2	19	0x01-0x02	
Channel number	1	21	0x00	Channel Z
Publish status	1	22	0x01	Publish Enabled
Future use	1	23	0x00	
Topic Name length	1	24	0x19	25
Mqtt publish fix header	3	25	0x00-0x30-0x25	
Reserved	2	28	0x00-0x19	
Topic Name	50	30	0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2F-0x53-0x45-0x4E-0x53-0x4F-0x52-0x2F-0x30-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-	F4B85E00A4760000/SENSOR/0

Whole frame

Buffer[80]> 0x4F-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x3F-0x90-0x07-0x01-0x02-0x00-0x01-0x00-0x19-0x00-0x30-0x25-0x00-0x19-0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2F-0x53-0x45-0x4E-0x53-0x4F-0x52-0x2F-0x30-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-

10.15 BEANDEVICE® WILOW MQTT OTAC TOPIC PROFILE EXAMPLE

Mqtt OTAC topic Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0		
Profile Id	1	17	0x90	
Mqtt Sub Id	1	18	8	
OTAC Subscribe status	1	19	0x01	
Reserved	1	20	0x00	
Topic Name length	1	21	0x15	21
Topic Name	50	22	0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2f-0x4f-0x54-0x41-0x43-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00	F4B85E00A4760000/OTAC

Whole frame

Buffer[72]> 0x47-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x37-0x90-0x08-0x01-0x00-0x15-0x46-0x34-0x42-0x38-0x35-0x45-0x30-0x30-0x41-0x34-0x37-0x36-0x30-0x30-0x30-0x30-0x2f-0x4f-0x54-0x41-0x43-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00

10.16 BEANDEVICE® WILOW MQTT NTP CONFIG PROFILE EXAMPLE

Ntp Config Profile				
Name field	**Size in byte**	**Byte index**	**Value**	**Additional information**
BeanScape Header	17	0	0x3F-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x2F	
Profile Id	1	17	0x91	
Time zone	2	18	0x3C-0x00	60 minutes
Future Use	5	20	0x00-0x00-0x00-0x00-0x00	
Ntp Port	2	25	0x7B-0x00	123
DNS Enabled/Disabled	1	27	0x01	DNS enabled
Ntp server IP	4	28	0x00-0x00-0x00-0x00	DNS is enabled IP is not filled
Ntp server Name length	1	32	0x0F	15
Ntp server URL	31	33	0x74-0x69-0x6D-0x65-0x2E-0x67-0x6F-0x6F-0x67-0x6C-0x65-0x2E-0x63-0x6F-0x6D-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-	time.google.com

Whole frame
Buffer[64]> 0x3F-0x4F-0x01-0xF4-0xB8-0x5E-0x00-0xA4-0x76-0x00-0x00-0x01-0x02-0xFF-0xFE-0x19-0x2F-0x91-0x3C-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x7B-0x00-0x01-0x00-0x00-0x00-0x00-0x00-0x0F-0x74-0x69-0x6D-0x65-0x2E-0x67-0x6F-0x6F-0x67-0x6C-0x65-0x2E-0x63-0x6F-0x6D-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-0x00-

10.17 EXAMPLE OF DOWNLOAD FRAMES

1. *The first frame*

1.1. The whole frame before decomposition

The whole size of the first frame is 576 bytes

ff-3f-2-4f-1-f4-b8-5e-0-a1-4b-0-0-1-b4-ff-fe-19-2c-2-f1-0-0-1-0-0-0-11-ac-1-0-28-20-2-f4-b8-5e-0-a1-4b-0-0-1-b4-ff-fe-7-0-0-0-42-3-1-1-2-0-1-64-0-1-0-0-ae-7-ae-7-ae-7-82-0-2-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-3-e2-7-2-1-f-1f-1b-9a-99-99-bd-3-42-c9-40-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-7-0-37-0-0-af-6e-1-20-26-f-4-5b-0-37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-82-1-2-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-3-e2-7-2-1-f-20-37-96-43-8b-bd-0-0-c8-40-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-7-0-37-0-0-af-6e-1-20-26-f-4-5b-0-37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-82-2-2-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-3-e2-7-2-1-f-21-37-4e-62-d0-be-7f-a0-c8-40-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-7-0-37-0-0-af-6e-1-20-26-f-4-5b-0-37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-0-37-0-0-af-6e-1-20-26-f-4-5b-0-37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-4-5b-69-0-e2-7-5-16-c-25-3a-0-0-0-0-0-0-0-0-0-0-a5-a5-a5-a5-c1-e-3-20-1-fe-1-20-a5-a5-a5-a5-45-90-0-20-3-af-1-0-e8-c1-2-20-17-1-4-5b-2-0-1-20-d8-af-1-0-25-8b-0-20-0-0-0-0-c1-c5-1-20-a5-a5-a5-a5-a5-a5-a5-a-0-c0-46-55-0-0-0-80-8e-3-20-c9-a6-1-0-24-64-3-20-c4-5f-3-20-30-7f-3-20-80-44-3-20-5-0-0-0-94-51-3-20-94-51-3-20-30-7f-3-20-8c-51-3-20-7-0-0-0-88-7f-3-20-44-41-51-5f-54-41-53-4b-3-0-0-0-0-a-0-0-1-0-0-e2-7-5-16-c-25-3a-

1.2. Decomposition of the frame

1.2.1. BeanDevice® Wilow® Frame Header

ff-3f-2-4f-1-f4-b8-5e-0-a1-4b-0-0-1-b4-ff-fe-19-2c-2

The total length = the whole size of the frame – size of the first field

Total length = 576 – 1(size of the frame length version) =575bytes=0x23F

Field name	Size in bytes	Index	Value
Frame length version	1	0	0xFF
Total Length (LSB)	2	1	0x3F
		2	0x02
Fixed(LSB)	2	3	0x4F
		4	0x01
MAC ID (MSB)	8	5	0xF4
		6	0xB8
		7	0x5E
		8	0x00
		9	0xA1
		10	0x4B
		11	0x00

		12	0x00
Fixed(LSB)	5	13	0x01
		14	0xB4
		15	0xFF
		16	0xFE
		17	0x19
Remaining Bytes(LSB)	2	18	0x2c
		19	0x02

1.2.2. Datalogger frame header

f1-0-0-1-0-0-0-11-ac

Field Name		Size in bytes	Value		Additional information
Frame type		1	0xf1		
File Index(LSB)		2	0x00		
			0x00		
Current Sequence Index(LSB)		4	0x01		
			0x00		
			0x00		
			0x00		
Datalogger frame flags		1	0x21		
4 Highest bits	4 lowest bits		4H bits	4L bits	
Ack requested/Not requested	Frame ID		0x2	0x1	
Download process		1	0xAC		1Lsb =0.5% =>0xAC=86%

1.2.3. Payload First frame

1-0-28-20-2-f4-b8-5e-0-a1-4b-0-0-1-b4-ff-fe-7-0-0-0-42-3-1-1-2-0-1-64-0-1-0-0-ae-7-ae-7-ae-7-82-0-2-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-3-e2-7-2-1-f-1f-1b-9a-99-99-bd-3-42-c9-40-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-7-0-37-0-0-af-6e-1-20-26-f-4-5b-0-37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-82-1-2-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-3-e2-7-2-1-f-20-37-96-43-8b-bd-0-0-c8-40-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-7-0-37-0-0-af-6e-1-20-26-f-4-5b-0-37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-82-2-2-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-3-e2-7-2-1-f-21-37-4e-62-d0-be-7f-a0-c8-40-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-7-0-37-0-0-af-6e-1-20-26-f-4-5b-0-37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-0-37-0-0-af-6e-1-20-26-f-4-5b-0-

37-0-0-2-b1-0-0-cf-d5-0-20-e8-c1-2-20-1e-0-2-20-94-e7-6f-d-0-0-0-0-c6-80-1-0-26-f-4-5b-69-0-e2-7-5-16-c-25-3a-0-0-0-0-0-0-0-0-0-0-a5-a5-a5-a5-c1-e-3-20-1-fe-1-20-a5-a5-a5-a5-45-90-0-20-3-af-1-0-e8-c1-2-20-17-1-4-5b-2-0-1-20-d8-af-1-0-25-8b-0-20-0-0-0-0-c1-c5-1-20-a5-a5-a5-a5-a5-a5-a5-a5-a-0-c0-46-55-0-0-0-80-8e-3-20-c9-a6-1-0-24-64-3-20-c4-5f-3-20-30-7f-3-20-80-44-3-20-5-0-0-0-94-51-3-20-94-51-3-20-30-7f-3-20-8c-51-3-20-7-0-0-0-88-7f-3-20-44-41-51-5f-54-41-53-4b-3-0-0-0-0-a-0-0-1-0-0-e2-7-5-16-c-25-3a

Field Name	Size in bytes	Byte index	Value	Additional information
Fixed	2	0	0x01	0x0001
		1	0x00	
Software Version	1	2	0x28	V2R8
Hardware Version	1	3	0x20	V2R0
Profile Version	1	4	0x02	Device Wilow
MAC ID(MSB)	8	5	0xf4	0xf4b85e00a14b0000
		6	0xb8	
		7	0x5e	
		8	0x00	
		9	0xa1	
		10	0x4b	
		11	0x00	
		12	0x00	
Fixed	4	13	0x01	
		14	0xb4	
		15	0xff	
		16	0xfe	
Channel bitmap(LSB)	4	17	0x07	Bit 0 set(channel Z activated)
				Bit 1 set(channel X activated)
				Bit 2 set(channel Y activated)
		18	0x00	
		19	0x00	
		20	0x00	

Main channel payload Id	**1**	**21**	**0x42**	
Number of channels	**1**	**22**	**0x03**	**Channel Z,X,Y**
Fixed	**1**	**23**	**0x01**	
Shock Detection Available Acceleration Range	**1**	**24**		
Shock Detection Acceleration Range	**2**	**25**		
		26		
Shock Detection Available sampling rate	**1**	**27**		
Shock Detection Sampling rate	**2**	**28**		
		29		
Shock notification delay	**2**	**30**		
		31		
Fixed	**1**	**32**	**0x00**	
Threshold X (signed short)	**2**	**33**		
		34		
Threshold Y (signed short)	**2**	**35**		
		36		
Threshold Z (signed short)	**2**	**37**		
		38		
Specific channel Payload Id	**1**	**39**	**0x82**	
Channel Id	**1**	**40**	**0x00**	**Channel Z**
Sensor Range	**1**	**41**	**0x02**	**Range -2/2g**

Threshold alarm(float)	4		42	0x00		
			43	0x00		
			44	0x00		
			45	0x00		
Threshold alarm(float)	4		46	0x00		
			47	0x00		
			48	0x00		
			49	0x00		
Threshold alarm(float)	4		50	0x00		
			51	0x00		
			52	0x00		
			53	0x00		
Threshold alarm(float)	4		54	0x00		
			55	0x00		
			56	0x00		
			57	0x00		
Channel Status	1	Bit0 Enable/Disable	58	1	0x03	
		Bit 1 Sensor Calibrated/not calibrated		1		
		Bit 2 (future use)		0		
		Bit 3(future use)		0		
		Bit 4(future use)		0		
		Bit 5(future use)		0		
		Bit 6(future use)		0		
		Bit 7(future use)		0		
Calibration date	7	Year(2 bytes)(Lsb)	59	0xe7	(0x7e7)2018	Date: 01/02/2018 at 15h:31m:27s
			60	0x07		

		Month(1 byte)	61	0x02	2	
		Day(1 byte)	62	0x01	1	
		Hour(1 byte)	63	0x0f	15	
		Minute(1 byte)	64	0x1f	31	
		Second(1 byte)	65	0x1b	27	
Calibration parameters	24	Offset(float)(LSB)	66	0x9a	0xbd99999a	-0,075
			67	0x99		
			68	0x99		
			69	0xbd		
		Ratio(Float)(LSB)	70	0x03	0x40c94203	6,28931
			71	0x42		
			72	0xc9		
			73	0x40		
		Future use	74			
			75			
			76			
			77			
			78			
			79			
			80			
			81			
			82			
			83			
			84			
			85			
			86			
			87			
			88			
			89			
Measurement unit	1		90	0x07		mg
Padding bytes	46		91->136	x		
Specific channel Payload Id	1		137	0x82		
Channel Id	1		138	0x01		Channel X

Sensor Range	1		139	0x02		Range -2/2g
Threshold alarm(float)	4		140	0x00		
			141	0x00		
			142	0x00		
			143	0x00		
Threshold alarm(float)	4		144	0x00		
			145	0x00		
			146	0x00		
			147	0x00		
Threshold alarm(float)	4		148	0x00		
			149	0x00		
			150	0x00		
			151	0x00		
Threshold alarm(float)	4		152	0x00		
			153	0x00		
			154	0x00		
			155	0x00		
Channel Status	1	Bit0 Enable/Disable	156	1	0x03	
		Bit 1 Sensor Calibrated/not calibrated		1		
		Bit 2 (future use)		0		
		Bit 3(future use)		0		
		Bit 4(future use)		0		
		Bit 5(future use)		0		
		Bit 6(future use)		0		
		Bit 7(future use)		0		
	7		157	0xe7	(0x7e7)2018	Date:

Calibration date		Year(2 bytes)(Lsb)	158	0x07		01/02/2018 at 15h:32m:55s
		Month(1 byte)	159	0x02	2	
		Day(1 byte)	160	0x01	1	
		Hour(1 byte)	161	0x0f	15	
		Minute(1 byte)	162	0x20	32	
		Second(1 byte)	163	0x37	55	
Calibration parameters	24	Offset(float)(LSB)	164	0x96	0xbd8b4396	-0,068
			165	0x43		
			166	0x8b		
			167	0xbd		
		Ratio(Float)(LSB)	168	0x00	0x40c80000	6,25
			169	0x00		
			170	0xc8		
			171	0x40		
		Future use	172			
			173			
			174			
			175			
			176			
			177			
			178			
			179			
			180			
			181			
			182			
			183			
			184			
			185			
			186			
			187			
Measurement unit	1		188	0x07		mg
Padding bytes	46		189->234	x		

Specific channel Payload Id	1		235	0x82		
Channel Id	1		236	0x02		Channel Y
Sensor Range	1		237	0x02		Range -2/2g
Threshold alarm(float)	4		238	0x00		
			239	0x00		
			240	0x00		
			241	0x00		
Threshold alarm(float)	4		242	0x00		
			243	0x00		
			244	0x00		
			245	0x00		
Threshold alarm(float)	4		246	0x00		
			247	0x00		
			248	0x00		
			249	0x00		
Threshold alarm(float)	4		250	0x00		
			251	0x00		
			252	0x00		
			253	0x00		
Channel Status	1	Bit0 Enable/Disable	254	1	0x03	
		Bit 1 Sensor Calibrated/not calibrated		1		
		Bit 2 (future use)		0		
		Bit 3(future use)		0		
		Bit 4(future use)		0		
		Bit 5(future use)		0		
		Bit 6(future use)		0		
		Bit 7(future use)		0		

Calibration date	7	Year(2 bytes)(Lsb)	255	0xe7	(0x7e7)2018	Date: 01/02/2018 at 15h:33m:55s
			256	0x07		
		Month(1 byte)	257	0x02	2	
		Day(1 byte)	258	0x01	1	
		Hour(1 byte)	259	0x0f	15	
		Minute(1 byte)	260	0x21	33	
		Second(1 byte)	261	0x37	55	
Calibration parameters	24	Offset(float)(LSB)	262	0x4e	0xbed0624e	-0,407
			263	0x62		
			264	0xd0		
			265	0xbe		
		Ratio(Float)(LSB)	266	0x7f	0x40c8a07f	6,26959
			267	0xa0		
			268	0xc8		
			269	0x40		
		Future use	270			
			271			
			272			
			273			
			274			
			275			
			276			
			277			
			278			
			279			
			280			
			281			
			282			
			283			
			284			
			285			
Measurement unit	1		286	0x07		mg

Padding bytes	**46**		**287->332**	**x**	
Specific channel Payload Id	**1**		**333**	**x**	
Channel Id	**1**		**334**	**x**	**Channel not used here**
Sensor Range	**1**		**335**	**x**	
Threshold alarm(float)	**4**		**336**	**x**	
			337	**x**	
			338	**x**	
			339	**x**	
Threshold alarm(float)	**4**		**340**	**x**	
			341	**x**	
			342	**x**	
			343	**x**	
Threshold alarm(float)	**4**		**344**	**x**	
			345	**x**	
			346	**x**	
			347	**x**	
Threshold alarm(float)	**4**		**348**	**x**	
			349	**x**	
			350	**x**	
			351	**x**	
Channel Status	**1**	**Bit0 Enable/Disable**	**352**	**x**	**x**
		Bit 1 Sensor Calibrated/not calibrated		**x**	
		Bit 2 (future use)		**x**	
		Bit 3(future use)		**x**	
		Bit 4(future use)		**x**	
		Bit 5(future use)		**x**	

		Bit 6(future use)		x		
		Bit 7(future use)		x		
Calibration date	7	Year(2 bytes)(Lsb)	353	x	x	
			354	x		
		Month(1 byte)	355	x	x	
		Day(1 byte)	356	x	x	
		Hour(1 byte)	357	x	x	
		Minute(1 byte)	358	x	x	
		Second(1 byte)	359	x	x	
Calibration parameters	24	Offset(float)(LSB)	360	x	x	
			361	x		
			362	x		
			363	x		
		Ratio(Float)(LSB)	364	x	x	
			365	x		
			366	x		
			367	x		
		Future use	368	x		
			369	x		
			370	x		
			371	x		
			372	x		
			373	x		
			374	x		
			375	x		
			376	x		
			377	x		
			378	x		
			379	x		
			380	x		
			381	x		
			382	x		
			383	x		
Measurement unit	1		384	x		

Padding bytes	**46**		**385->430**	**x**	
Specific channel Payload Id	**1**		**431**	**x**	
Channel Id	**1**		**432**	**x**	
Sensor Range	**1**		**433**	**x**	
Threshold alarm(float)	**4**		**434**	**x**	
			435	**x**	
			436	**x**	
			437	**x**	
Threshold alarm(float)	**4**		**438**	**x**	
			439	**x**	
			440	**x**	
			441	**x**	
Threshold alarm(float)	**4**		**442**	**x**	
			443	**x**	
			444	**x**	
			445	**x**	
Threshold alarm(float)	**4**		**446**	**x**	
			447	**x**	
			448	**x**	
			449	**x**	
Channel Status	**1**	**Bit0 Enable/Disable**	**450**	**x**	**x**
		Bit 1 Sensor Calibrated/not calibrated		**x**	
		Bit 2 (future use)		**x**	
		Bit 3(future use)		**x**	
		Bit 4(future use)		**x**	
		Bit 5(future use)		**x**	
		Bit 6(future use)		**x**	

		Bit 7(future use)		x		
Calibration date	7	Year(2 bytes)(Lsb)	451	x	x	
			452	x		
		Month(1 byte)	453	x	x	
		Day(1 byte)	454	x	x	
		Hour(1 byte)	455	x	x	
		Minute(1 byte)	456	x	x	
		Second(1 byte)	457	x	x	
Calibration parameters	24	Offset(float)(LSB)	458	x	x	
			459	x		
			460	x		
			461	x		
		Ratio(Float)(LSB)	462	x	x	
			463	x		
			464	x		
			465	x		
		Future use	466	x		
			467	x		
			468	x		
			469	x		
			470	x		
			471	x		
			472	x		
			473	x		
			474	x		
			475	x		
			476	x		
			477	x		
			478	x		
			479	x		
			480	x		
			481	x		
Measurement unit	1		482	x		
Padding bytes	46		483->528	x		

Measurement mode	1		529	0x03		Streaming
Low duty Cycle(LSB) in seconds	3		530	0x00		
			531	0x00		
			532	0x00		
Tx Ratio	1		533	0x00		
Streaming Frequency(LSB)	3		534	0x0a		10Hz
			535	0x00		
			536	0x00		
Duration in seconds	1		537	0x01		1Second
			538	0x00		
			539	0x00		
Start logging date	Year	2 byte(LSB)	540	0xe2	0x7e2 =2018	22/05/2018 at 12
			541	0x07		
	Month	1	542	0x05	May	
	Day	1	543	0x16	22	
	Hour	1	544	0x0c	12	
	Minute	1	545	0x25	37	
	Second	1	546	0x3a	58	

2. *Data frame*

2.1. The whole frame before decomposition

73-4f-1-f4-b8-5e-0-a1-4b-0-0-1-b4-ff-fe-19-63-f2-0-0-1-0-0-0-12-c8-d6-3-0-b-0-0-18-1-80-d8-3-0-8-0-0-18-1-80-d9-3-0-9-0-0-18-1-80-d6-3-0-b-0-0-18-1-80-d7-3-0-9-0-0-19-1-80-d2-3-0-9-0-0-19-1-80-d4-3-0-d-0-0-18-1-80-d2-3-0-7-0-0-1a-1-80-d3-3-0-9-0-0-1b-1-80-d7-3-0-d-0-0-1a-1-80-

2.2. Decomposition of the frame

2.2.1. BeanDevice® Wilow® frame header

***NB:** If the first byte of the Wilow® frame header equals **0xff** we have a long frame and the <u>total length</u> is contained on the two next bytes otherwise we have a short frame and the <u>total length</u> is contained on the first byte.*

In our example we have the frame starting with a 0x73 =/= 0xff hence we have a short frame and the first byte refers to Total length.

73-4f-1-f4-b8-5e-0-a1-4b-0-0-1-b4-ff-fe-19-63

Field name	Size in bytes	Byte Index	Value
Total Length	1	0	0x73

Fixed(LSB)	2	1	0x4F
			0x01
MAC ID (MSB)	8	3	0xF4
			0xB8
			0x5E
			0x00
			0xA1
			0x4B
			0x00
			0x00
Fixed(LSB)	5	11	0x01
			0xB4
			0xFF
			0xFE
			0x19
Remaining Bytes	1	16	0x63

2.2.2. Datalogger frame header

f2-0-0-1-0-0-0-12-c8

Field Name		Size in bytes	Value		Additional informations
Frame type		1	0xf2		
File Index(LSB)		2	0x00		
			0x00		
Current Sequence Index(LSB)		4	0x01		
			0x00		
			0x00		
			0x00		
Datalogger frame flags		1	0x12		
4 Highest bits	4 lowest bits		4H bits	4L bits	
Ack requested/Not requested	Frame ID		0x1	0x2	
Download process		1	0xc8		1Lsb =0.5% =>0xc8=100%

2.2.2.1. Frame types

Frame type	*value*
First frame	*0xf1*
Data frame	*0xf2*

2.2.2.2. Datalogger flags

flags	*values*
Acknowledgment requested	*0x10*
Acknowledgment not requested	*0x20*
First frame id	*0x01*
Data frame id	*0x02*

2.2.3. Payload

The payload is a set of data acquired each data is signed using sign-magnitude and 3 bytes sized generally the data is organized as follow:

Index in payload	*0*	*3*	*6*	*9*	*12*	*15*	*18*	*21*
Corresponding data	*channelZ*	*ChannelX*	*ChannelY*	*INCX*	*INCY*	*channelZ*	*ChannelX*	*…*

This frame depends on channel status Enabled/Disabled found in Channel bitmap field in the First frame payload index 17, for example:

- **If channel bitmap = 0x01 → only channel Z is activated and the frame will be like**

Index in payload	*0*	*3*	*6*	*9*	*12*	*15*	*18*	*21*
Corresponding data	*channelZ*	*ChannelZ*	*ChannelZ*	*ChannelZ*	*ChannelZ*	*channelZ*	*ChannelZ*	*…*

- **If the channel bitmap = 0x05 → channel Z and Channel Y are activated and the frame will be like**

Index in payload	*0*	*3*	*6*	*9*	*12*	*15*	*18*	*21*

Corresponding data	*channelZ*	*ChannelY*	*ChannelZ*	*ChannelY*	*ChannelZ*	*channelY*	*ChannelZ*	…

- **If the channel bitmap = 0x18 → channel IncX and Channel IncY are activated and the frame will be like (wich not the case in AX3D we do not have inclinometer sensors)**

Index in payload	*0*	*3*	*6*	*9*	*12*	*15*	*18*	*21*
Corresponding data	*INCX*	*INCY*	*INCX*	*INCY*	*INCX*	*INCY*	*INCX*	…

Back to our example where we have the following payload:
d6-3-0-b-0-0-18-1-80-d8-3-0-8-0-0-18-1-80-d9-3-0-9-0-0-18-1-80-d6-3-0-b-0-0-18-1-80-d7-3-0-9-0-0-19-1-80-d2-3-0-9-0-0-19-1-80-d4-3-0-d-0-0-18-1-80-d2-3-0-7-0-0-1a-1-80-d3-3-0-9-0-0-1b-1-80-d7-3-0-d-0-0-1a-1-80
The number of data depends on channel activated and sampling rate and duration of acquisition:
Here we have streaming (as mentioned in the first frame Measurement mode index number 529)10hz (as mentioned in the first frame Streaming Frequency index number 534,535,536) one shot with 1s duration (as mentioned in the first frame Duration index number 537,538,539)
Channel Z : d6-3-0 =0x3d6 =982mg
Channel X: b-0-0=0xb=11mg
Channel Y: 18-1-80 = 0x800118(negative value) =0b 1000 0000 0000 0001 0001 1000 = -280mg

10.18 BEANDEVICE® WILOW MQTT LDC MATH RESULT EXAMPLE

3 frames from sensor0 of the X-in device for Minimum, Maximum and Average results.

Frame 1: 0505 0001 197F 735E 167F 735E 197F 735E EC03 00
Frame 2: 0505 0002 297F 735E 167F 735E 297F 735E E303 00
Frame 3: 0505 0003 0000 0000 167F 735E 297F 735E E703 00

Parsing frame 1:

Field name	Size	Value in Hex (LSB first)	Data meaning	Final data
Device type	1 byte	05	X Inc	X Inc
DAC type	1 byte	05	LDC math result	LDC math result
Channel Id	1 byte	00	Channel Z	Channel Z
Math result type	1 byte	01	Maximum	Maximum
Event time (unix format)	4 bytes	19 7F 73 5E	2020/03/19 15:18:01	2020/03/19 15:18:01
Start time (unix format)	4 bytes	16 7F 73 5E	2020/03/19 15:17:58	NA

End time (unix format)	4 bytes	29 7F 73 5E	2020/03/19 15:18:17	NA
Data measurement	3 bytes	EC 03 00	1004	0.1004

Parsing frame 2:

Field name	Size	Value in Hex (LSB first)	Data meaning	Final data
Device type	1 byte	05	X Inc	X Inc
DAC type	1 byte	05	LDC math result	LDC math result
Channel Id	1 byte	00	Channel Z	Channel Z
Math result type	1 byte	02	Minimum	Minimum
Event time (unix format)	4 bytes	29 7F 73 5E	2020/03/19 15:18:17	2020/03/19 15:18:17
Start time (unix format)	4 bytes	16 7F 73 5E	2020/03/19 15:17:58	NA
End time (unix format)	4 bytes	29 7F 73 5E	2020/03/19 15:18:17	NA
Data measurement	3 bytes	E3 03 00	995	0.995

Parsing frame 2:

Field name	Size	Value in Hex (LSB first)	Data meaning	Final data
Device type	1 byte	05	X Inc	X Inc
DAC type	1 byte	05	LDC math result	LDC math result
Channel Id	1 byte	00	Channel Z	Channel Z
Math result type	1 byte	03	Average	Average
Event time (unix format)	4 bytes	0000 0000	NA	NA
Start time (unix format)	4 bytes	16 7F 73 5E	2020/03/19 15:17:58	2020/03/19 15:17:58
End time (unix format)	4 bytes	29 7F 73 5E	2020/03/19 15:18:17	2020/03/19 15:18:17
Data measurement	3 bytes	E7 0300	999	0.999

- ✓ From each channel we will receive 3 frames for math result (maximum, minimum and average).
- ✓ For Minimum and Maximum result we need only the event time.
- ✓ For Average result we need the start and the end time
- ✓ The time is in unix format and it's LSB first which mean the first byte is the less significant byte.

- ✓ Data measurement is LSB first (first byte is the less significant byte) and the last bit is the sign bit which mean if it's 0 the value is positive if it's 1 the value is negative. After that you need to divide it by 1000 to get the right measurement.

10.19 BEANDEVICE® WILOW MQTT DYNAMIC MATH RESULT EXAMPLE

Example of the dynamic math result frame captured from X-Inx device in streaming one shot mode with all 5 channels enabled.

0507 03D0 0700 0000 0500 0055 1F00 0000 E407 0313 0F0E 3500 00E4 0703 130F 0E3A 0000 4200 9100 0000 3D00 AD03 0063 0000 0012 0016 0400 E403 006E 0000 003A 0000 0000 2E00 0000 2200 1900 000D 0000 2A00 0000 0200 1A00 8019 0000 0025 0001 0080 0E00 8093 0000 002C 003A 0480 2C00 0000 2600 1F04 802B 0480 2A00 0000 0400 6303 0077 0000 0041 007F 0300 7103 00

Field name				Size		Value in Hex (LSB first)	Data
Device type				1 byte		05	X-Inc
DAC type				1 byte		07	Dynamic math result
DAC mode				1 byte		03	Streaming
Acquisition frequency				2 byte		D0 07	2000 Hz
Acquisition cycle				3 bytes		00 00 00	NA
Acquisition duration				3 bytes		05 00 00	5 sec
Future use				1 bytes		55	NA
Channels bitmap				4 bytes		1F 00 00 00	Channels 1, 2, 3, 4, 5 enabled
Start time	Years			2 bytes	9 bytes	E4 07	2020
	Months			1 byte		03	03
	Days			1 byte		13	19
	Hours			1 byte		0F	15
	Minutes			1 byte		0E	14
	Seconds			1 byte		35	53
	Milliseconds			2 bytes		00 00	0
End time	Years			2 bytes	9 bytes	E4 07	2020
	Months			1 byte		03	03
	Days			1 byte		13	19
	Hours			1 byte		0F	15
	Minutes			1 byte		0E	14
	Seconds			1 byte		3A	58
	Milliseconds			2 bytes		00 00	0
Number data per channel				2 bytes		42 00	66
Channel 1		Minimum	Frame Id	4 bytes		91 00 00 00	145

			Index	2 bytes	3D 00	61
			Value	3 bytes	AD 03 00	0.941
		Maximum	Frame Id	4 bytes	63 00 00 00	99
			Index	2 bytes	12 00	18
			Value	3 bytes	16 04 00	1.046
		Average	Value	3 bytes	E4 03 00	0.996
	Channel 2	Minimum	Frame Id	4 bytes	6E 00 00 00	110
			Index	2 bytes	3A 00	58
			Value	3 bytes	00 00 00	0
		Maximum	Frame Id	4 bytes	2E 00 00 00	46
			Index	2 bytes	22 00	34
			Value	3 bytes	19 00 00	0.025
		Average	Value	3 bytes	0D 00 00	0.013
	Channel 3	Minimum	Frame Id	4 bytes	2A 00 00 00	42
			Index	2 bytes	02 00	2
			Value	3 bytes	1A 00 80	-0.026
		Maximum	Frame Id	4 bytes	19 00 00 00	25
			Index	2 bytes	25 00	37
			Value	3 bytes	01 00 80	-0.001
		Average	Value	3 bytes	0E 00 80	-0.014
	Channel 4	Minimum	Frame Id	4 bytes	93 00 00 00	147
			Index	2 bytes	2C 00	44
			Value	3 bytes	3A 04 80	-1.082
		Maximum	Frame Id	4 bytes	2C 00 00 00	44
			Index	2 bytes	26 00	38
			Value	3 bytes	1F 04 80	-1.055
		Average	Value	3 bytes	2B 04 80	-1.067
	Channel 5	Minimum	Frame Id	4 bytes	2A 00 00 00	42
			Index	2 bytes	04 00	4
			Value	3 bytes	63 03 00	0.867
		Maximum	Frame Id	4 bytes	77 00 00 00	119
			Index	2 bytes	41 00	65
			Value	3 bytes	7F 03 00	0.895
		Average	Value	3 bytes	71 03 00	0.881

Example to calculate the event time of the maximum result for the channel 1:

- ✓ First calculate the full index which represent the index of the maximum value from the whole stream event (since frame id 0)

FullIndex = Frame Id * number of data per channel + Index

So FullIndex = 99 * 66 + 18 = 6552

- ✓ Convert the start time to timestamp in millisecond.

2020/03/19 15:14:53.0 => 1584630893

- ✓ Now performe this formula to get the event time of the maximum value:

EventTime = startTime + 1/acquisition frequency * FullIndex

So EventTime = 1584630893 + 1/2000 * 6552 = 1584630896,276

- ✓ Last step is to convert the timestam to a human date:

1584630896,276 =>2020/03/19 15:14:56.276

11. APPENDICE 2: HOW TO CALCULATE A DATE WITH FRACTION OF SECONDS FOR STREAMING MODE

On this example we will show how to estimate the Timestamp frame in Streaming Mode.
The Date is obtained from the Start Date and SubPacket (which provides the tiemstamp information).
The following formulation is used to get thee Subpacket value:

$$T_{SubPacket} = Reference\ Time + \left(\frac{1}{Sampling\ frequency}\right) * SubPacket\ Index$$

Where

$$SubPacket\ Index = (Frame\ Sequence\ Id * Previous\ Number\ of\ data\ acquisitions\ per\ channel) + Current\ SubPacket\ row$$

Acquisition mode: **Streaming one shot**
Sampling frequency: **5 Hz**
Duration: **30seconds**

Frame Sequence Id = 0

MqttStreamingFrame:

The frame could be interpreted as two parts:

1. Header (colored Font)
2. Payload Data (highlighted in Yellow"for first channel",Green"second channel" and Blue"third channel")

|0x05|0x03|0x24|0x21|0xA2|0x5B|0x05|0x00|0x07|0x00|0x00|0x00|0x00|0x00|0x00|0x6E|0x00|
0x00|0x00|0x00|0x1E|0x00|0x00|0x00|0x00|0x00|0x8E|0x97|0x00|0x00|0x01|0x00|0x00|0x04|0
x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x
04|0x00|0x80|0x9A|0x00|0x00|0x02|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x0
0|0x05|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00
|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|
0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0
x01|0x00|0x00|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x
00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x0
0|0x00|0x01|0x00|0x00|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x02|0x00|0x80|0x99
|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|
0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0
x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x
00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x0
4|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x02|0x00|0x00
|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|
0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x00|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0
x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x03|0x00|0x80|0x9A|0x00|0x00|0x
01|0x00|0x00|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x0
0|0x02|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00
|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|

0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x02|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x00|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x00|0x00|0x00|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x03|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x00|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|

Data is organized in SubPacket:

Referring to channels bitmap we have 3 channels activated:
Channel0 =Z,
Channel1=X
Channel2=Y
hence each SubPacket will contain 9bytes (3bytes for each channel) below the data organized in SubPackets:

SubPacketRow	Channel Z	Channel X	Channel Y
0	|0x97|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
1	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
2	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
3	|0x9A|0x00|0x00|	|0x02|0x00|0x00|	|0x04|0x00|0x80|
4	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x05|0x00|0x80|
5	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
6	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
7	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
8	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
9	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
10	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
11	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
12	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
13	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
14	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
15	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x02|0x00|0x80|
16	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
17	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
18	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
19	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
20	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
21	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
22	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
23	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
24	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
25	|0x9A|0x00|0x00|	|0x02|0x00|0x00|	|0x03|0x00|0x80|
26	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
27	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
28	|0x98|0x00|0x00|	|0x00|0x00|0x00|	|0x04|0x00|0x80|
29	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
30	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
31	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
32	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
33	|0x9A|0x00|0x00|	|0x02|0x00|0x00|	|0x04|0x00|0x80|
34	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
35	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
36	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
37	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
38	|0x99|0x00|0x00|	|0x02|0x00|0x00|	|0x04|0x00|0x80|
39	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
40	|0x99|0x00|0x00|	|0x00|0x00|0x00|	|0x04|0x00|0x80|

41	|0x98|0x00|0x00|	|0x00|0x00|0x00|	|0x03|0x00|0x80|
42	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
43	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
44	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
45	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
46	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
47	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
48	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
49	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
50	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
51	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
52	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
53	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
54	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
55	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
56	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
57	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
58	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
59	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
60	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
61	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
62	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
63	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
64	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
65	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
66	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
67	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
68	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
69	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
70	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
71	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
72	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
73	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
74	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
75	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
76	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
77	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
78	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
79	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
80	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
81*	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
82	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
83	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|

84	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
85	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
86	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
87	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
88	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
89	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
90	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
91	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
92	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
93	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
94	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
95	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
96	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
97	|0x98|0x00|0x00|	|0x00|0x00|0x00|	|0x04|0x00|0x80|
98	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
99	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
100	|0x99|0x00|0x00|	|0x00|0x00|0x00|	|0x04|0x00|0x80|
101	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
102	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
103	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
104	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
105	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
106	|0x99|0x00|0x00|	|0x00|0x00|0x00|	|0x04|0x00|0x80|
107	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
108	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
109	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|

<table>
<tr><th colspan="3">Data meaning</th><th colspan="2">Size</th><th>Example value</th></tr>
<tr><td colspan="3">Device Type</td><td colspan="2">1 byte</td><td>0x05</td></tr>
<tr><td colspan="3">Acquisition type (Default 0x03)</td><td colspan="2">1 byte</td><td>0x03</td></tr>
<tr><td colspan="3">Reference Start time
In Unix time format (LSB First)</td><td colspan="2">4 bytes</td><td>0x24|0x21|0xA2|0x5B</td></tr>
<tr><td colspan="3">Sampling frequency (LSB First)</td><td colspan="2">2 bytes</td><td>0x05|0x00</td></tr>
<tr><td rowspan="8">Channels bitmap
(LSB First)</td><td>Is channel 1 activated?</td><td>0th Bit</td><td rowspan="4">1st Byte</td><td rowspan="8">4 bytes</td><td>1</td></tr>
<tr><td>Is channel 2 activated?</td><td>1st Bit</td><td>1</td></tr>
<tr><td>Is channel 3 activated?</td><td>2nd Bit</td><td>1</td></tr>
<tr><td>……</td><td>……</td><td>0</td></tr>
<tr><td>……</td><td>……</td><td>:</td></tr>
<tr><td>……</td><td>:</td><td>2nd Byte</td><td>:</td></tr>
<tr><td>……</td><td>:</td><td>3rd Byte</td><td>:</td></tr>
<tr><td>……</td><td></td><td>4th Byte</td><td>:</td></tr>
</table>

	Is channel 32 activated ?		31th Bit		0
Frame Sequence Id (LSB First):(Begins from 0)			3 bytes		0x00|0x00|0x00
Number of data acquisition per channel			2 bytes		0x6E|0x00
Data Acquisition cycle			3 bytes		0x00|0x00|0x00
Data acquisition duration			3 bytes		0x1E|0x00|0x00
Previous Number of data acquisition per channel(LSB first)			2 bytes		0x00|0x00
Synchronization			1 bit		0
Future Use			7 bits		0000000
Network Quality (LQI)			1 byte		0x8E
Data Sample 1 of channel 1 (LSB First)		1st Sub Packet	1 byte		0x97
	Byte[1]		1 byte		0x00
	Byte[2] Sign bit		8th bit	1 byte	0x00
	Byte[2] data bits		7 bits		
Data Sample 1 of channel 2 (LSB First)			3 bytes		0x01|0x00|0x00
Data Sample 1 of channel 3 (LSB First)			3 bytes		0x04|0x00|0x80
Data Sample 2 of channel 1 (LSB First)		2nd Sub Packet	3 bytes		
Data Sample 2 of channel 2 (LSB First)			3 bytes		
Data Sample 2 of channel 3 (LSB First)			3 bytes		
....		...			
Data Sample 110 of channel 1 (LSB First)		110th Sub Packet	3 bytes		0x98|0x00|0x00
Data Sample 110 of channel 2 (LSB First)			3 bytes		0x01|0x00|0x00
Data Sample 110 of channel 3 (LSB First)			3 bytes		0x04|0x00|0x80

Let's calculate the timestamp of the following Subpacket :

81	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|

$$T_{SubPacket} = Reference\ Time + \left(\frac{1}{Sampling\ frequency}\right) * SubPacket\ Index$$

$$SubPacket\ Index = (Frame\ Sequence\ Id * Previous\ Number\ of\ data\ acquisitions\ per\ channel) + Current\ SubPacket\ row$$

Calculate SubPacket index:

Frame sequence Id = 0

Previous number of data acquisition per channel = 0

Current SubPacket row = 81

$$SubPacket\ Index = 81$$

Calculate T SubPacket:

Reference Start Time = 19/09/2018 10:12:52

Sampling rate = 5 Hz

$$T_{SubPacket} = 19/09/2018\ 10:13:08:200ms$$

Frame Sequence Id = 1

MqttStreamingFrame:

The frame could be interpreted as two parts:

1. Header (colored Font)
2. Payload Data (highlighted in Yellow"for first channel",Green"second channel" and Blue"third channel")

|0x05|0x03|0x24|0x21|0xA2|0x5B|0x05|0x00|0x07|0x00|0x00|0x00|0x01|0x00|0x00|0x28|0x00|0x00|0x00|0x00|0x1E|0x00|0x00|0x6E|0x00|0x00|0x8E|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x03|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x00|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x00|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x9A|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x00|0x00|0x00|0x04|0x00|0x80|0x99|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|0x98|0x00|0x00|0x01|0x00|0x00|0x04|0x00|0x80|

SubPacketRow	Channel Z	Channel X	Channel Y
0	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
1	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|

2	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
3	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
4	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
5	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
6	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
7	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
8	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
9	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
10	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
11	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
12	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
13	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
14	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
15	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
16	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
17	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
18	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x03|0x00|0x80|
19	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
20	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
21	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
22	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
23	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
24	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
25	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
26	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
27	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
28	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
29	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
30	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
31	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
32	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
33	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
34	|0x9A|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
35*	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
36	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
37	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
38	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|
39	|0x98|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|

Data meaning	Size	Example value
Device Type	1 byte	**0x05**
Acquisition type (Default 0x03)	1 byte	**0x03**

Reference time In Unix time format (LSB First)		4 bytes			**0x24|0x21|0xA2|0x5B**
Sampling frequency (LSB First)		2 bytes			**0x05|0x00**
Channels bitmap (LSB First)	Is channel 1 activated?	0^{th} Bit	1^{st} Byte	4 bytes	1
	Is channel 2 activated?	1^{st} Bit			1
	Is channel 3 activated?	2^{nd} Bit			1
	……	……			0
	……	……			:
	……	:	2^{nd} Byte		:
	……	:	3^{rd} Byte		:
	……		4^{th} Byte		:
	Is channel 32 activated ?	31^{th} Bit			0
Frame Sequence Id (LSB First):(Begins from 0)		3 bytes			**0x01|0x00|0x00**
Number of data acquisitions per channel		2 bytes			**0x28|0x00**
Data Acquisition cycle		3 bytes			**0x00|0x00|0x00**
Data acquisition duration		3 bytes			**0x1E|0x00|0x00**
Previous Number of data acquisition per channel(LSB first)		2 bytes			**0x6E|0x00**
Synchronization		1 bit			**0**
Future Use		7 bits			**0000000**
Network Quality (LQI)		1 byte			**0x8E**

Data Sample 1 of channel 1 (LSB First)			1st Sub Packet	1 byte	**0x98**
	Byte[2]			1 byte	**0x00**
	Byte[2]	Sign bit		8th bit	**0x00**
		data bits		7 bits	
Data Sample 1 of channel 2 (LSB First)				3 bytes	**0x01|0x00|0x00**
Data Sample 1 of channel 3 (LSB First)				3 bytes	**0x04|0x00|0x80**
Data Sample 2 of channel 1 (LSB First)			2nd Sub Packet	3 bytes	
Data Sample 2 of channel 2 (LSB First)				3 bytes	
Data Sample 2 of channel 3 (LSB First)				3 bytes	
....			...		
Data Sample 40 of channel 1 (LSB First)			110th Sub Packet	3 bytes	**0x98|0x00|0x00**
Data Sample 40 of channel 2 (LSB First)				3 bytes	**0x01|0x00|0x00**
Data Sample 40 of channel 3 (LSB First)				3 bytes	**0x04|0x00|0x80**

Let's calculate the timestamp of the following Subpacket :

35*	|0x99|0x00|0x00|	|0x01|0x00|0x00|	|0x04|0x00|0x80|

$$T_{SubPacket} = Reference\ Time + \left(\frac{1}{Sampling\ frequency}\right) * SubPacket\ Index$$

$$SubPacket\ Index = (Frame\ Sequence\ Id * Previous\ Number\ of\ data\ acquisitions\ per\ channel) + Current\ SubPacket\ row$$

Calculate SubPacket index:

Frame sequence Id = 1
Previous number of data acquisition per channel = 110
Current SubPacket row = 35

$$SubPacket\ Index = 145$$

Calculate T SubPacket:

Reference Time = 19/09/2018 10:12:52
Sampling rate = 5 Hz

$$T_{SubPacket} = 19/09/2018\ 10:13:21$$

12. MQTT CONFIGURATION USING CLOUD MQTT BROKER

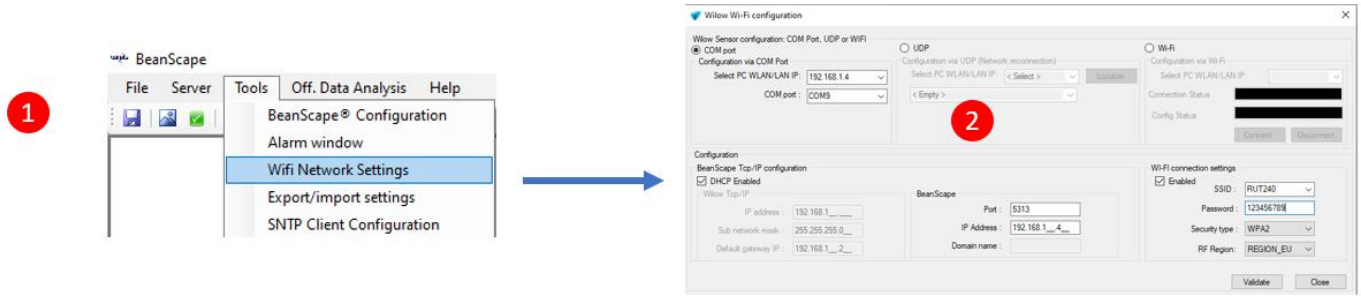
when using a cloud MQTT Broker make sure that your Broker is linked to internet service.

12.1 USING HIVEMQ MQTT BROKER

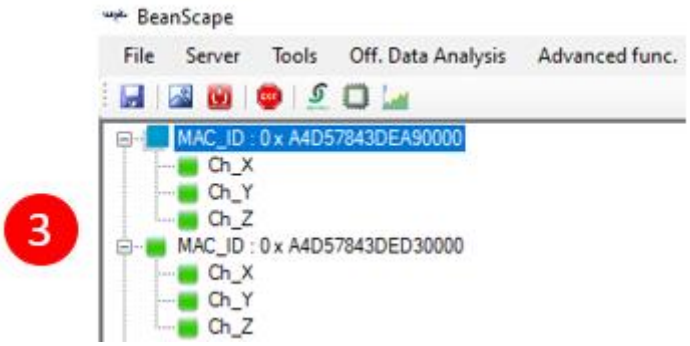
Broker DNS: **broker.hivemq.com**

12.1.1 BeanDevice® MQTT Configuration

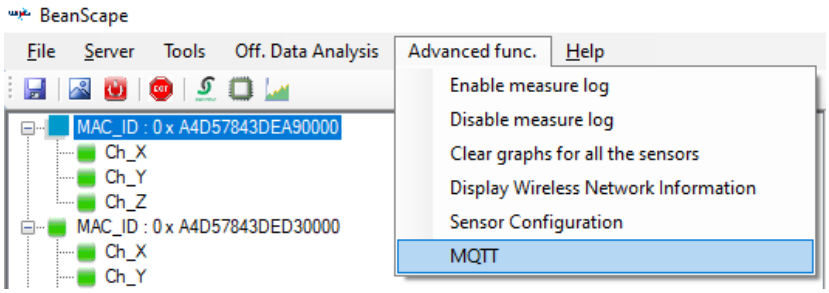
1.Go to tools > WIFI network settings, and enter the right access point settings to connect your BeanDevice® to BeanScape® software.



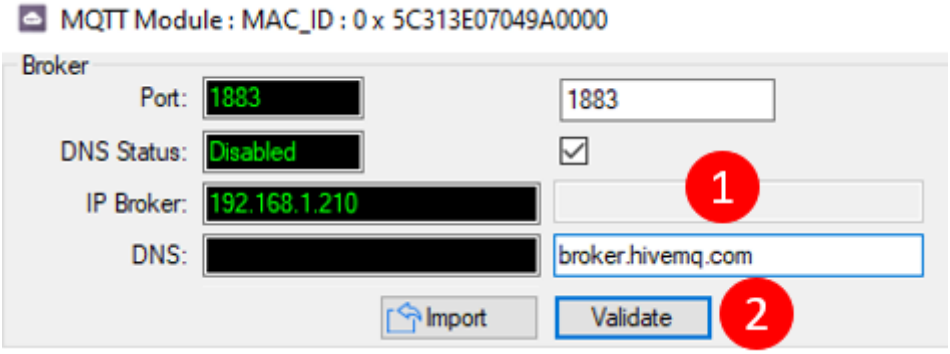
The BeanDevice® profile will be displayed on the left panel.



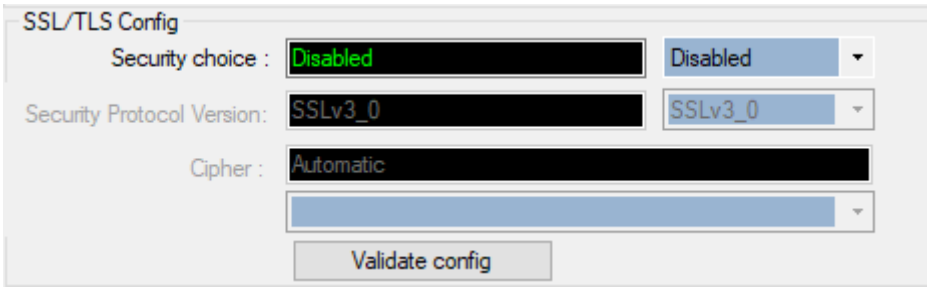
2.Click on the BeanDevice® profile then navigate to Advanced functions > MQTT



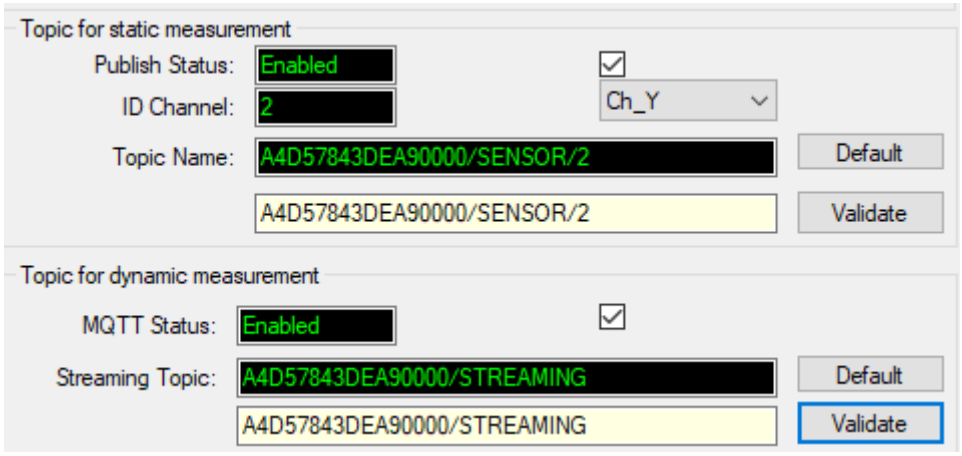
3.Configure the broker by entering the right MQTT port, check the DNS check box to enable the DNS option, then tap the broker DNS.



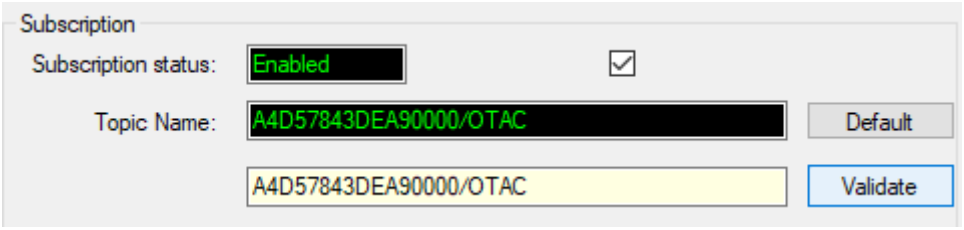
Make sure that the SSL/TLS option is disable in case if not used.



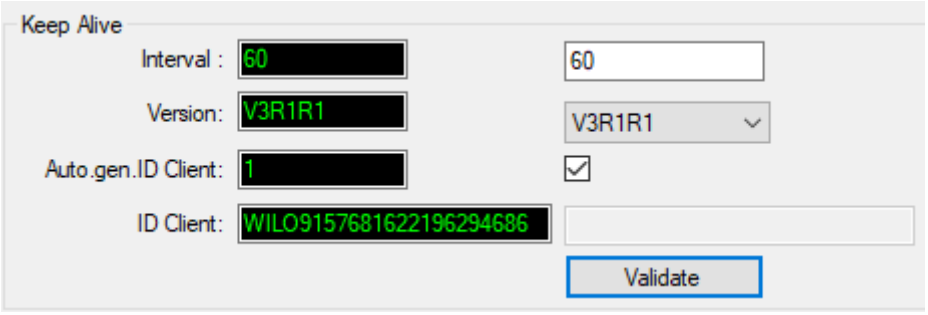
4.Configure the topics for dynamic and static measurement mode for each sensor.



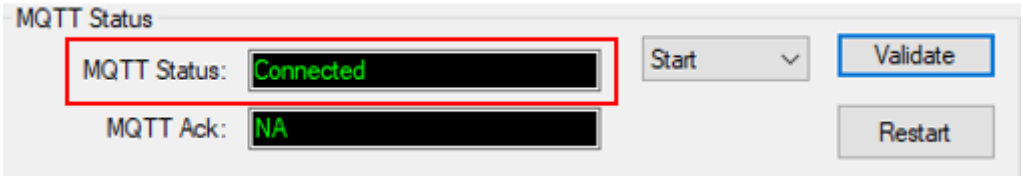
5.Configure the Topic Name on which the device and the software will subscribe.



6.Configure the Keep alive option and generate and automatic client ID, then click validate.



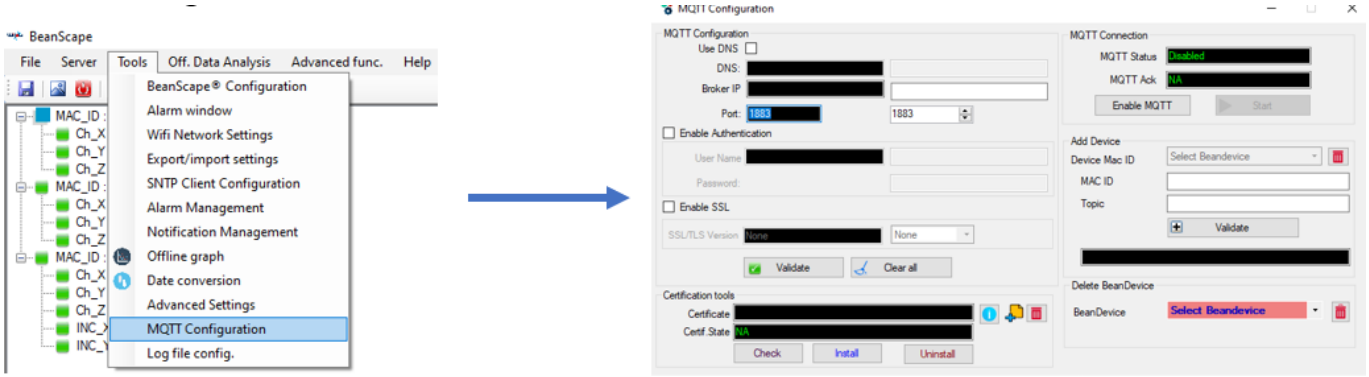
7.Now, launch the MQTT communication.



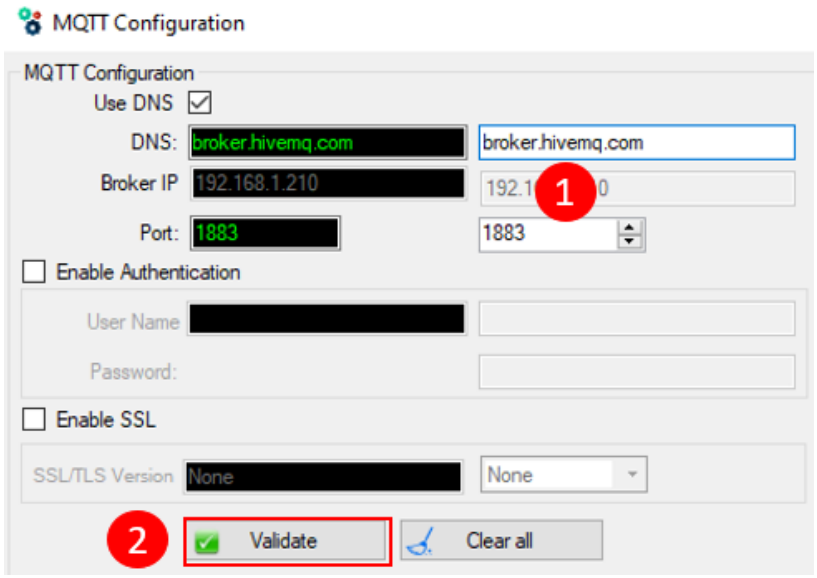
The MQTT status should display connected, otherwise check your configuration settings again or check the internet status on your router.

12.1.2 BeanScape® MQTT configuration (Only RA version)

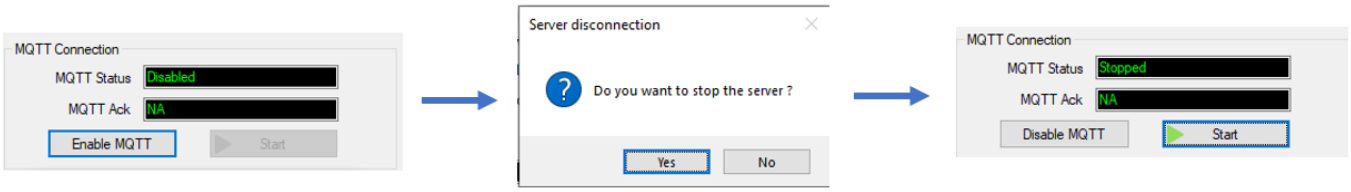
Go to Tools > MQTT Configuration



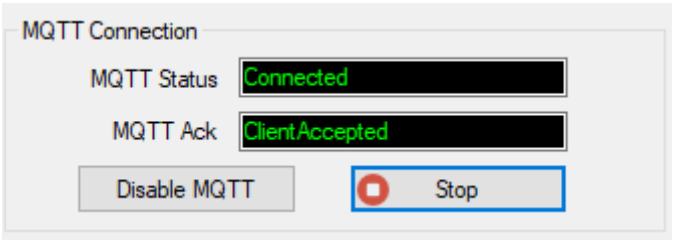
1. Set the MQTT port and Your router IP address, make sure that is the same configuration used on the BeanDevice. Don't forget to click on validate



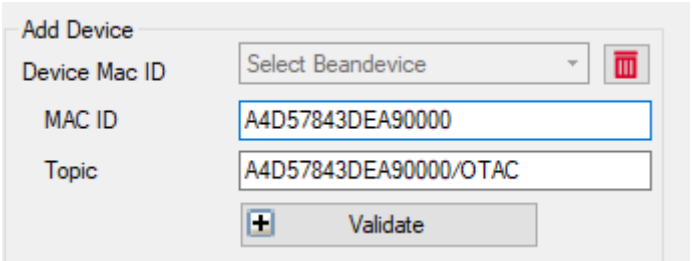
2. Now enable the MQTT, then click on Yes. After that click on Start MQTT



3. Once it done, the MQTT status should be connected and the client is accepted



4. One last step, is to enter the BeanDevice MAC ID and subscribe to the subscription topic name, click on validate



5. Now your BeanDevice is working on the MQTT protocol.