



# WAGO Messaging Protocol

© 2019 WAGO Kontakttechnik GmbH & Co. KG  
All rights reserved.

**WAGO Kontakttechnik GmbH & Co. KG**

Hansastraße 27  
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0  
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: [info@wago.com](mailto:info@wago.com)

Web: [www.wago.com](http://www.wago.com)

**Technical Support**

Phone: +49 (0) 571/8 87 – 4 45 55  
Fax: +49 (0) 571/8 87 – 84 45 55

E-Mail: [support@wago.com](mailto:support@wago.com)

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: [documentation@wago.com](mailto:documentation@wago.com)

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

WAGO is a registered trademark of WAGO Verwaltungsgesellschaft mbH.

# Table of Contents

<b>1</b>	<b>Notes about this Documentation .....</b>	<b>5</b>
1.1	Copyright.....	5
1.2	Symbols .....	5
1.3	Number Notation .....	7
1.4	Font Conventions .....	7
1.5	Legal Bases.....	8
1.5.1	Subject to Changes.....	8
1.5.2	Personal Qualifications .....	8
1.5.3	Limitation of Liability.....	8
1.6	Revision History.....	9
<b>2</b>	<b>Introduction .....</b>	<b>10</b>
<b>3</b>	<b>Protocol Handshake .....</b>	<b>10</b>
3.1	Overview messages .....	10
3.2	DeviceHello .....	11
3.3	CloudHello.....	11
3.4	Handshake sequences .....	12
3.4.1	Default case.....	12
3.4.2	Cloud Application legacy version .....	12
3.4.3	DataAgent protocol version is not supported by the Cloud Application .....	13
3.4.4	DataAgent has no connection before handshake was initiated.....	13
<b>4</b>	<b>Protocol Version 1.0 .....</b>	<b>14</b>
4.1	Overview messages .....	14
4.2	Heartbeat .....	15
4.3	DeviceInfo .....	16
4.4	DeviceState .....	17
4.5	TagConfiguration .....	18
4.6	TagValues .....	20
4.7	EventTagValues .....	20
4.8	CommandRegistration.....	21
4.9	CommandResponse.....	22
4.10	MyCommand0 .....	22
4.11	GetDeviceInfo.....	23
4.12	SetSamplingInterval .....	23
4.13	SetPublishingInterval.....	24
4.14	GetTagConfiguration .....	25
4.15	StopTelemetry .....	25
4.16	StartTelemetry.....	26
4.17	SetHeartbeatInterval.....	26
4.18	DeleteController.....	27
<b>5</b>	<b>Protocol Version 1.2 .....</b>	<b>28</b>
5.1	Overview messages .....	28
5.2	WAGO Cloud specific messages.....	29
5.3	DeviceInfo .....	30
5.4	DeviceState .....	31

---

5.5	TagConfiguration .....	33
5.6	TagValues .....	34
5.7	CommandRegistration .....	35
5.8	CommandResponse .....	36
5.9	UpdateStatus .....	36
5.10	UpdateFirmware .....	37
5.11	StartRemoteAccessMediator .....	38
5.12	StopRemoteAccessMediator .....	38
<b>6</b>	<b>Protocol Version 1.4.x .....</b>	<b>39</b>
6.1	Overview messages .....	39
6.2	WAGO Cloud specific messages .....	40
6.3	DeviceInfo .....	41
6.4	TagConfiguration .....	42
6.5	TagValues .....	43
6.6	UpdateFirmware .....	44
6.7	UpdateApplication .....	45
6.8	UpdateDevice .....	46
	<b>List of Figures .....</b>	<b>48</b>
	<b>List of Tables .....</b>	<b>49</b>

# 1 Notes about this Documentation

## 1.1 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

## 1.2 Symbols



### DANGER

**Personal Injury!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

---



### DANGER

**Personal Injury Caused by Electric Current!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

---



### WARNING

**Personal Injury!**

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

---



### CAUTION

**Personal Injury!**

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

---

### NOTICE

**Damage to Property!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

---



## NOTICE

### **Damage to Property Caused by Electrostatic Discharge (ESD)!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.



## Note

### **Important Note!**

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



## Information

### **Additional Information:**

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

## 1.3 Number Notation

Table 1: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

## 1.4 Font Conventions

Table 2: Font Conventions

Font Type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Program Files\WAGO Software</i>
<b>Menu</b>	Menu items are marked in bold letters. e.g.: <b>Save</b>
<b>&gt;</b>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: <b>File &gt; New</b>
<b>Input</b>	Designation of input or optional fields are marked in bold letters, e.g.: <b>Start of measurement range</b>
"Value"	Input or selective values are marked in inverted commas. e.g.: Enter the value "4 mA" under <b>Start of measurement range</b> .
<b>[Button]</b>	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: <b>[Input]</b>
<b>[Key]</b>	Keys are marked with bold letters in square brackets. e.g.: <b>[F5]</b>

## **1.5 Legal Bases**

### **1.5.1 Subject to Changes**

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

### **1.5.2 Personal Qualifications**

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards.

Moreover, the persons cited here must also be familiar with all of the products cited in this document, along with the operating instructions. They must also be capable of correctly predicting any hazards which may not arise until the products are combined.

WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

### **1.5.3 Limitation of Liability**

This documentation describes the use of various hardware and software components in specific example applications. The components may represent products or parts of products from different manufacturers. The respective operating instructions from the manufacturers apply exclusively with regard to intended and safe use of the products. The manufacturers of the respective products are solely responsible for the contents of these instructions.

The sample applications described in this documentation represent concepts, that is, technically feasible application. Whether these concepts can actually be implemented depends on various boundary conditions. For example, different versions of the hardware or software components can require different handling than that described here. Therefore, the descriptions contained in this documentation do not form the basis for assertion of a certain product characteristic.

Responsibility for safe use of a specific software or hardware configuration lies with the party that produces or operates the configuration. This also applies when one of the concepts described in this document was used for implementation of the configuration.

WAGO Kontakttechnik GmbH & Co. KG is not liable for any actual implementation of the concepts.



## 1.6 Revision History

Table 3: Revision History

Version	Author	Description of Change	Date
0.1.0	Stefan Eggert	Initial document	2018-06-19
0.2.0	Daniel Martens	Documented protocol version 1.0	2018-08-07
0.3.0	Anatolij Ristok	Documented protocol version 1.1	2018-10-16
0.3.1	Daniel Martens	Formatting the documentation Documented protocol version 1.2	2019-03-14
0.4.0	Daniel Martens	Documented protocol version 1.4.x	2019-09-04

## 2 Introduction

This document describes the communication between a WAGO PFC running the Cloud-Connectivity software and an application running within a cloud platform. It provides necessary information and documentation for developers on how the protocol handshake must be implemented and gives an overview of the different existing versions of the protocol.

## 3 Protocol Handshake

The protocol handshake is the first set of messages that are exchanged between DataAgent (major part of the Cloud-Connectivity software) and the Cloud Application. The result of the handshake is a specific protocol version that is supported by both, the DataAgent and the Cloud Application.

### 3.1 Overview messages

Two kinds of messages are involved in the protocol handshake: DeviceHello and CloudHello (see Figure 3-1).



Figure 3-1: Protocol handshake messages

Table 4: Overview messages

Message	Origin	Application properties (WAGO Cloud, Azure)	MQTT topic (AnyMQTT, AWS, IMB Bluemix)
<i>DeviceHello</i>	DataAgent	MessageType=DeviceHello	{CID}/DeviceHello
<i>CloudHello</i>	Cloud App	MessageType=CloudHello	{CID}/CloudHello

{CID} represents the "Client ID" entered in WBM.

## 3.2 DeviceHello

```
{
  "ProtocolVersions": [
    "1.0",
    "1.2",
    "1.4.0",
    "1.4.1" // Available if configured so
  ]
}
```

The message body is formatted as shown in the example above. The message will be sent by the DataAgent and initiates a protocol handshake. It contains a list of protocol versions that are supported by DataAgent.

From protocol version 1.4 it is also possible to send the data as compressed messages. The last digit of the protocol version indicates the compression.

- 1.x.0: without compression
- 1.x.1: compression with GZIP (contains the same JSON format)

## 3.3 CloudHello

```
{
  "ProtocolVersion": "1.4.0"
}
```

The message body is formatted as shown in the example above. This message is the answer of the Cloud Application to the DeviceHello message. It contains the protocol version that DataAgent should use for further communication with the Cloud Application.

## 3.4 Handshake sequences

### 3.4.1 Default case

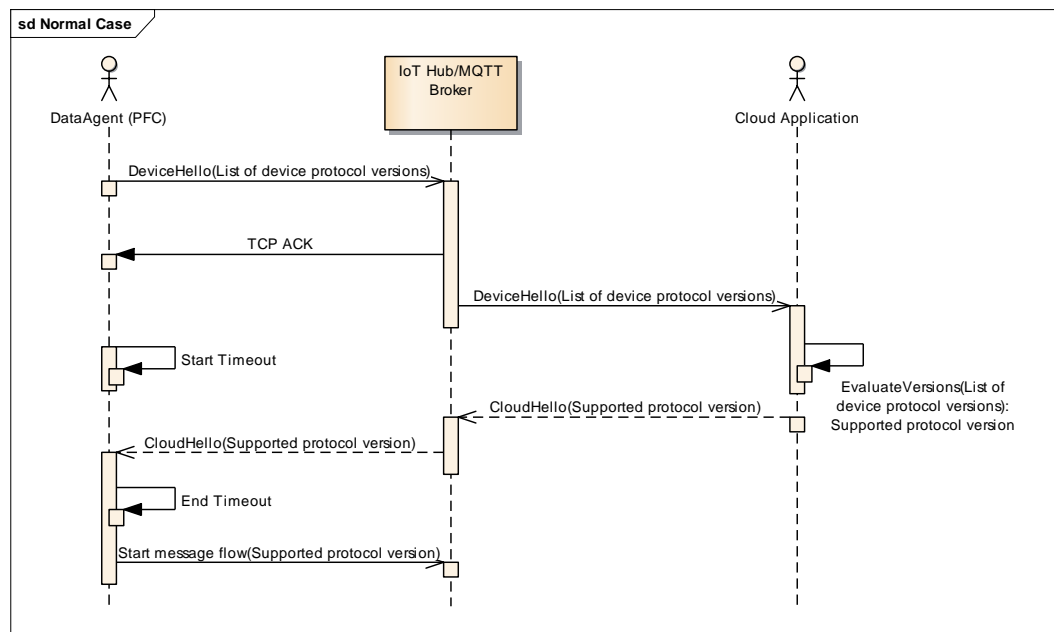


Figure 3-2: Handshake sequence - Default case

### 3.4.2 Cloud Application legacy version

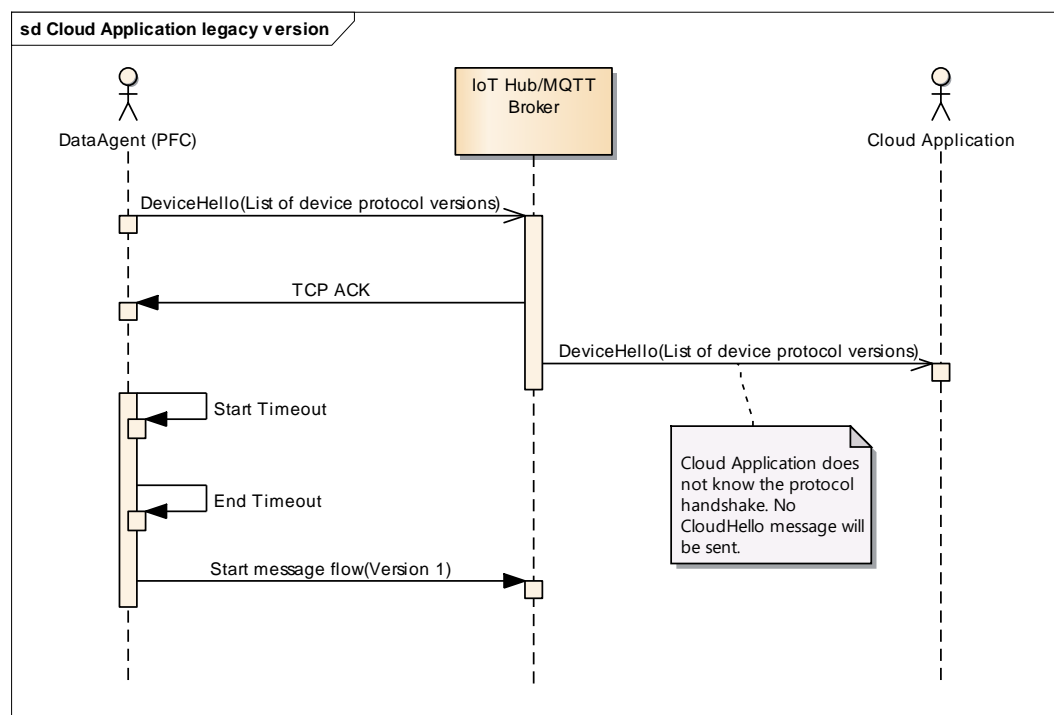


Figure 3-3: Handshake sequence - Cloud Application legacy version

### 3.4.3 DataAgent protocol version is not supported by the Cloud Application

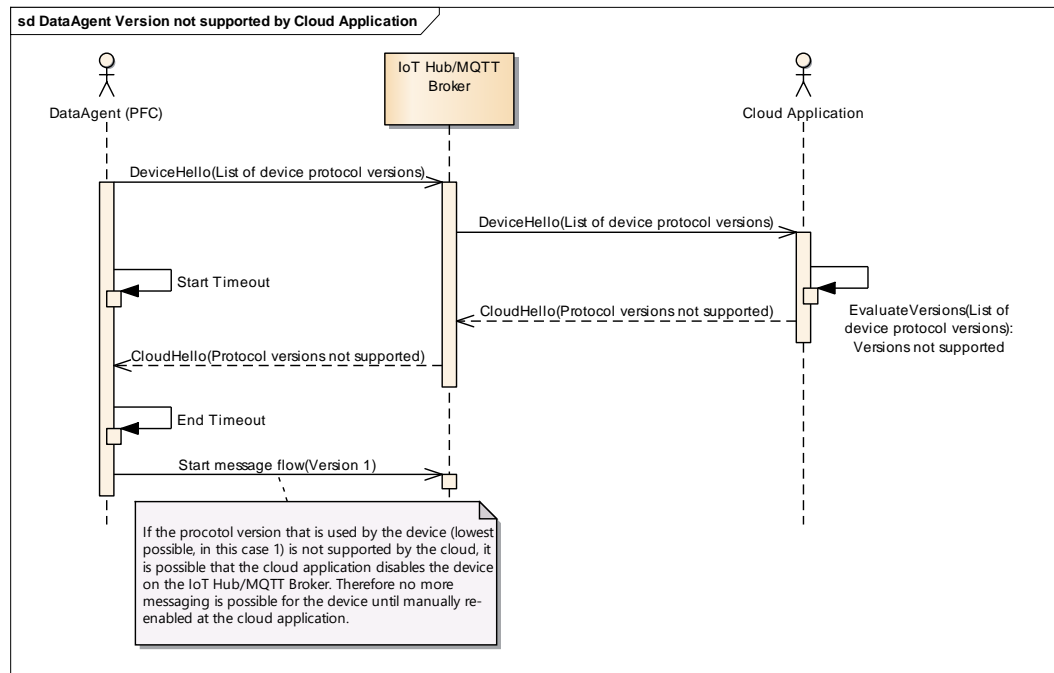


Figure 3-4: Handshake sequence - DataAgent protocol version is not supported by the Cloud Application

### 3.4.4 DataAgent has no connection before handshake was initiated

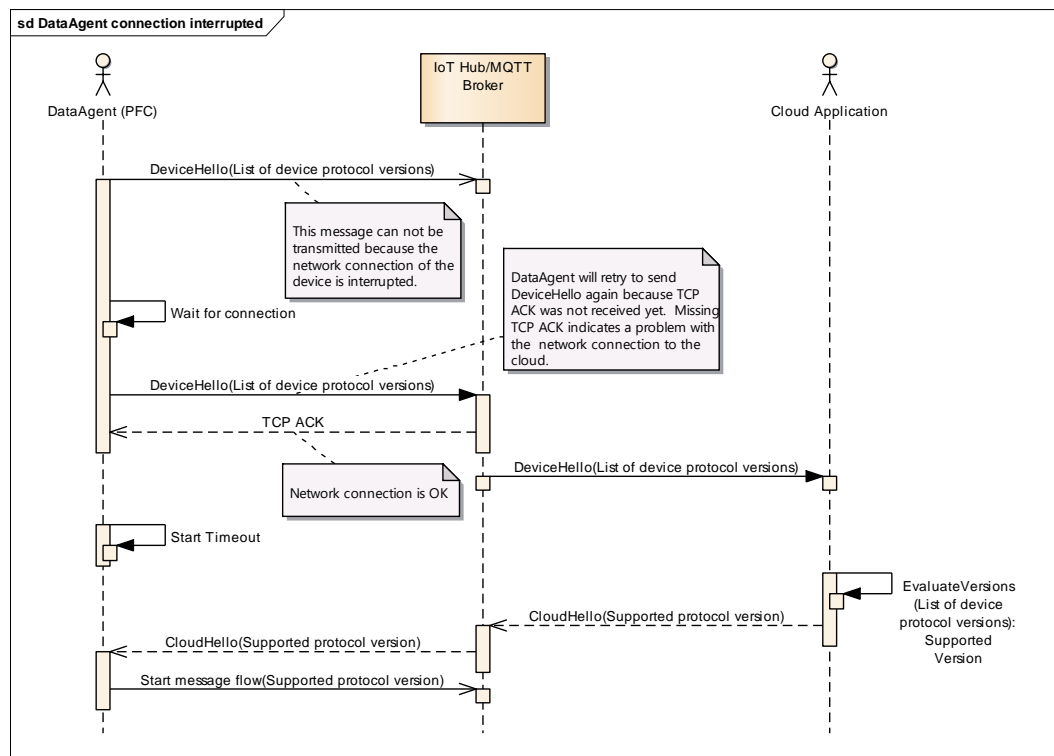


Figure 3-5: Handshake sequence - DataAgent connection is interrupted while handshake was initiated

## 4 Protocol Version 1.0

DataAgent, which was released as part of PFC Firmware 11 will by default communicate via this protocol without initiating the protocol handshake. As opposite an “more up to date” DataAgent will always initiate the protocol handshake but use this protocol as fallback in case no other version could be arranged with the Cloud Application.

### 4.1 Overview messages

Table 5: Overview messages

Message	Origin	Application properties (WAGO Cloud, Azure)	MQTT topic (AnyMQTT, AWS)	MQTT topic (IBM Cloud)
<i>Heartbeat</i>	DataAgent	MessageType=Heartbeat	-	-
<i>DeviceInfo</i>	DataAgent	MessageType=DeviceInfo	{CID}/DeviceInfo	iot-2/evt/ DeviceInfo/fmt/json
<i>DeviceState</i>	DataAgent	MessageType=DeviceState	{CID}/DeviceState	iot-2/evt/ DeviceState/fmt/json
<i>TagConfiguration</i>	PLC App	MessageType=TagConfiguration	{CID}/TagConfiguration	iot-2/evt/ TagConfiguration/fmt/json
<i>TagValues</i>	PLC App	MessageType=TagValues	{CID}/TagValues	iot-2/evt/ TagValues/fmt/json
<i>EventTagValues</i>	PLC App	MessageType=EventTagValues	{CID}/EventTagValues	iot-2/evt/ EventTagValues/fmt/json
<i>CommandRegistration</i>	PLC App	MessageType=CommandRegistration	{CID}/CommandRegistration	-
<i>CommandResponse</i>	PLC App / DataAgent	MessageType=CommandResponse	{CID}/CommandResponse	-
<i>MyCommand0</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-
<i>GetDeviceInfo</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-
<i>SetSamplingInterval</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-
<i>SetPublishingInterval</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-
<i>GetTagConfiguration</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-
<i>StopTelemetry</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-

## WAGO Messaging Protocol

<i>StartTelemetry</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-
<i>SetHeartbeatInterval</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-
<i>DeleteController</i>	Cloud App	CommandRequestId={I}	{CID}/Commands	-

{I} is a placeholder for identifier given to the command request  
 {CID} represents the "Client ID" entered in WBM.

## 4.2 Heartbeat

```
{
  "MessageType": "Heartbeat"
}
```

The message body is formatted as shown in the example above. The message will be sent by the DataAgent on a constant time cycle.

## 4.3 DeviceInfo

```
{
  "FormatId": "DeviceInfo",
  "ApiVersion": 1,
  "CurrentRuntimeVersion": "3",
  "DataAgentVersion": "1.0.1.2312",
  "FirmwareVersion": "02.08.25(11)",
  "LicenseInfo": "Codesys-Runtime-License",
  "NTP": "disabled",
  "OrderNumber": "750-8207",
  "ProductDescription": "WAGO 750-8207 PFC200 CS 2ETH RS 3G",
  "WebserverDefault": "Webserver",
  "WebserverStateCodesys": "",
  "WebserverStateECockpit": "disabled",
  "NetworkInterfaces": [
    {
      "Name": "X1",
      "Mac": "00-30-de-42-15-6d",
      "Ip": "192.168.2.132",
      "Netmask": "255.255.255.0"
    }
  ],
  "IoModules": [
    {
      "Position": 1,
      "Code": 34817
    }
  ]
}
```

This message will be sent at the beginning of the communication with the cloud or explicitly when requested by cloud application.



## 4.4 DeviceState

```
{
  "FormatId": "DeviceState",
  "ApiVersion": 1,
  "CurrentTime": "2018-10-16T16:15:57.246+02:00",
  "UserSwitch": "State:Run",
  "Leds": [
    {
      "Name": "IO",
      "State": "On",
      "Color": "Green"
    },
    {
      "Name": "MS",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "NET",
      "State": "Blinking",
      "Color": "Red"
    },
    {
      "Name": "NS",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "RUN",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S1",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S2",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S3",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S4",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S5",
      "State": "On",
      "Color": "Off"
    }
  ]
}
```

```

    },
    {
        "Name": "SYS",
        "State": "On",
        "Color": "Green"
    },
    {
        "Name": "USR",
        "State": "On",
        "Color": "Off"
    }
]
}

```

This message will be sent when status of the LED's or the START/STOP-switch has changed. DataAgent will detect such changes by polling the status continuously.

## 4.5 TagConfiguration

```

{
    "FormatId": "TagConfiguration",
    "ApiVersion": 1,
    "Complete": true,
    "Collections": [
        {
            "Id": 0,
            "CollectionName": "MyCollection0",
            "SampleRate": 10000,
            "PublishInterval": 10000,
            "TagData": [
                {
                    "Tag": "MyVariable1",
                    "Unit": "kg",
                    "TypeId": 0,
                    "ValueType": "real" // {bool, byte, uint, int, uint,
word,                                     dint, uint, dword, real,
lreal}
                },
                {
                    "Tag": "MyVariable2",
                    "Unit": "litre",
                    "TypeId": 0,
                    "ValueType": "word"
                }
            ]
        },
        {
            "Id": 1,
            "CollectionName": "MyCollection1",
            "SampleRate": 10000,
            "PublishInterval": 10000,
            "TagData": [
                {
                    "Tag": "MyVariable3",
                    "Unit": "kg",
                    "TypeId": 0,
                    "ValueType": "real" // {bool, byte, uint, int, uint,
word,                                     dint, uint, dword, real,
lreal}
                },
                {
                    "Tag": "MyVariable4",
                    "Unit": "litre",
                    "TypeId": 0,
                    "ValueType": "word"
                }
            ]
        }
    ]
}

```

```
"Id": 1,
"CollectionName": "MyCollection1",
"SampleRate": 10000,
"PublishInterval": 10000,
"TagData": [
  {
    "Tag": "MyVariable3",
    "Unit": "u3",
    "TypeId": 0,
    "ValueType": "byte"
  },
  {
    "Tag": "MyVariable4",
    "Unit": "u4",
    "TypeId": 0,
    "ValueType": "dword"
  }
]
}
```

The description of the collections (lists of variables) will be sent once to the cloud. The description consists of variable lists. The variable list consists of a list of variables which are specified within the PLC program. The configuration of the cyclical and the eventbased collections are sending with TagConfiguration to the cloud.

The PLC application developer must configure each collection if the data will be sent cyclical or on events.

## 4.6 TagValues

```
{
  "FormatId": "TagValues",
  "ApiVersion": 1,
  "CollectionId": 0,
  "TagData": [
    {
      "Time": "2018-10-16T16:16:18.331+02:00",
      "Values": {
        "MyVariable1": 5.530000,
        "MyVariable2": 10
      }
    }
  ]
}
```

This message will be sent cyclically by means of the configured PublishInterval within the PLC application.

## 4.7 EventTagValues

```
{
  "FormatId": "EventTagValues",
  "ApiVersion": 1,
  "CollectionId": 1,
  "TagData": [
    {
      "Time": "2018-10-16T16:16:20.331+02:00",
      "Values": {
        "MyVariable4": 10
      }
    }
  ]
}
```

The data transmission is triggered by an event within the PLC application. With the *collectionTrigger* all data of the collections and with the *variableTrigger* only the values of related variable are included within the message body.

## 4.8 CommandRegistration

```
{
  "FormatId": "CommandRegistration",
  "ApiVersion": 1,
  "CommandList": [
    {
      "Name": "MyCommand0",
      "Id": 0,
      "RequestParameters": [
        {
          "Name": "Parameter 0 from Cloud to PFC",
          "Type": "real"           // {bool, dint, string,
real, uint}
        },
        {
          "Name": "Parameter 1 from Cloud to PFC",
          "Type": "string"
        }
      ],
      "ResponseParameters": [
        {
          "Name": "Response Param 0 PFC to Cloud",
          "Type": "real"
        },
        {
          "Name": "Response Param: Command received",
          "Type": "bool"
        }
      ]
    }
  ]
}
```

The PLC application developer can define specific commands which can be triggered from the cloud application.

This message contains list of the custom commands, which was created within the PLC application, and will be sent to the cloud once. Every command lists its own request- and response parameters. Later on the Cloud Application can request the execution of such a command by sending corresponding command request. After the PLC application have executed the commands it should also send corresponding command response informing about the results.

## 4.9 CommandResponse

```
{
  "FormatId": "CommandResponse",
  "ApiVersion": 1,
  "CommandId": 256,
  "CommandRequestId": 42,
  "CommandResponseParameters": [
    {
      "Name": "myRespP5",
      "Value": "myValue"
    },
    {
      "Name": "myRespP6",
      "Value": "1.713"
    }
  ]
}
```

After receiving a command request from the cloud the DataAgent / PLC application will execute it and optionally confirm the results by sending similar message as shown above.

## 4.10 MyCommand0

```
{
  "CommandId": 0,
  "CommandParameters": [
    {
      "Name": "Parameter 0 from Cloud to PFC",
      "Value": "23.42"
    },
    {
      "Name": "Parameter 1 from Cloud to PFC",
      "Value": "myStringValue"
    }
  ]
}
```

The example above shows body of a custom command. Custom commands are defined by PLC application and forwarded to the cloud within CommandRegistration message.

## 4.11 GetDeviceInfo

```
{
  "CommandId": 504,
  "CommandParameters": [
  ]
}
```

This standard command enables Cloud application to request DeviceInfo message.

The CommandResponse of this standard command includes the following parameter:

- "Name": "TransferTriggered"  
"Value": "bool"

## 4.12 SetSamplingInterval

```
{
  "CommandId": 256,
  "CommandParameters": [
    {
      "Name": "CollectionId",
      "Value": "1"
    },
    {
      "Name": "SampleIntervalMilliseconds",
      "Value": "1000"
    }
  ]
}
```

This standard command enables Cloud application to set sampling interval for specified collection.

The CommandResponse of this standard command includes the following parameters:

- "Name": "Result"  
"Value": "bool"
- "Name": "CollectionId"  
"Value": "uint"
- "Name": "SampleIntervalMilliseconds"  
"Value": "uint"

## 4.13 SetPublishingInterval

```
{
  "CommandId": 257,
  "CommandParameters": [
    {
      "Name": "CollectionId",
      "Value": "1"
    },
    {
      "Name": "PublishIntervalMilliseconds",
      "Value": "5000"
    }
  ]
}
```

This standard command enables Cloud application to set publishing interval for specified collection.

The CommandResponse of this standard command includes the following parameters:

- "Name": "Result"  
"Value": "bool"
- "Name": "CollectionId"  
"Value": "uint"
- "Name": "PublishIntervalMilliseconds"  
"Value": "uint"



## 4.14 GetTagConfiguration

```
{
  "CommandId": 258,
  "CommandParameters": [
    {
      "Name": "SendTagConfiguration",
      "Value": "true"
    }
  ]
}
```

This standard command enables Cloud application to request TagConfiguration message.

The CommandResponse of this standard command includes the following parameter:

- "Name": "Result"  
"Value": "bool"

## 4.15 StopTelemetry

```
{
  "CommandId": 501,
  "CommandParameters": [

  ]
}
```

This standard command enables Cloud application to stop transmission of telemetry data.

The CommandResponse of this standard command includes the following parameter:

- "Name": "Stopped"  
"Value": "bool"

## 4.16 StartTelemetry

```
{
  "CommandId": 502,
  "CommandParameters": [
  ]
}
```

This standard command enables Cloud application to start transmission of telemetry data.

The CommandResponse of this standard command includes the following parameter:

- "Name": "Started"  
"Value": "bool"

## 4.17 SetHeartbeatInterval

```
{
  "CommandId": 503,
  "CommandParameters": [
    {
      "Name": "IntervalInSeconds",
      "Value": "600"
    }
  ]
}
```

This standard command enables Cloud application to set heartbeat interval.

The CommandResponse of this standard command includes the following parameter:

- "Name": "IntervalInSeconds"  
"Value": "dint"

## 4.18 DeleteController

```
{
  "CommandId": 500,
  "CommandParameters": [

  ]
}
```

This standard command enables Cloud application to reset configuration of DataAgent and stop it. Afterwards the PFC is disconnected from the cloud platform.

The CommandResponse of this standard command includes the following parameter:

- "Name": "Succeeded"  
"Value": "bool"

## 5 Protocol Version 1.2

### 5.1 Overview messages

Table 6: Overview messages

Message	Origin	Application properties (WAGO Cloud, Azure)	MQTT topic (AnyMQTT, AWS)	MQTT topic IBM Cloud
<i>Heartbeat</i>	DataAgent	MessageType=Heartbeat ProtocolVersion=1.2	-	-
<i>DeviceInfo</i>	DataAgent	MessageType=DeviceInfo ProtocolVersion=1.2	1.2/{CID}/ DeviceInfo	iot-2/evt/ DeviceInfo/fmt/ json
<i>DeviceState</i>	DataAgent	MessageType=DeviceState ProtocolVersion=1.2	1.2/{CID}/ DeviceState	iot-2/evt/ DeviceState/f mt/json
<i>TagConfiguration</i>	PLC App	MessageType=TagConfiguration ProtocolVersion=1.2	1.2/{CID}/ TagConfiguration	iot-2/evt/ TagConfigurat ion/ fmt/json
<i>TagValues</i>	PLC App	MessageType=TagValues ProtocolVersion=1.2	1.2/{CID}/ TagValues	iot-2/evt/ TagValues/fmt /json
<i>CommandRegistration</i>	PLC App	MessageType=CommandRegistration ProtocolVersion=1.2	1.2/{CID}/ CommandRegistration	-
<i>CommandResponse</i>	PLC App / DataAgent	MessageType=CommandResponse ProtocolVersion=1.2	1.2/{CID}/ CommandResponse	-
<i>GetDeviceInfo</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-
<i>SetSamplingInterval</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-
<i>SetPublishingInterval</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-

## WAGO Messaging Protocol

<i>GetTagConfiguration</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-
<i>StopTelemetry</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-
<i>StartTelemetry</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-
<i>SetHeartbeatInterval</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-
<i>DeleteController</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-
<i>MyCommand0</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2	1.2/{CID}/Commands	-

{I} is a placeholder for identifier given to the command request  
 {CID} represents the "Client ID" entered in WBM.

## 5.2 WAGO Cloud specific messages

Message	Origin	Application properties (WAGO Cloud)
<i>UpdateStatus</i>	DataAgent	MessageType=UpdateStatus ProtocolVersion=1.2
<i>StartRemoteAccessMediator</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2
<i>StopRemoteAccessMediator</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2
<i>UpdateFirmware</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.2

## 5.3 DeviceInfo

```
{
  "MessageType": "DeviceInfo",
  "WagoProtocol": "1.2",
  "CurrentRuntimeVersion": "string",
  "DataAgentVersion": "string",
  "FirmwareVersion": "string",
  "HardwareReleaseIndex": "string",
  "NTP": "string",
  "LicenseInfo": "string",
  "OrderNumber": "string",
  "ProductDescription": "string",
  "WebserverDefault": "string",
  "WebserverStateCodesys": "string",
  "WebserverStateECockpit": "string",
  "NetworkInterfaces": [
    {
      "Name": "string",
      "Mac": "string",
      "Ip": "string",
      "Netmask": "string"
    }
  ],
  "IoModules": [
    {
      "Position": 0,
      "Code": 0
    }
  ],
  "ConfiguredFeatures": [
    "DeviceStatusOnChange",
    "StandardCommands"
    "FirmwareUpdate", // optional
    "RemoteAccess", // optional
  ]
}
```

PFC will send this message automatically on start if device information was configured within WBM. DataAgent will determine the list of configured features at runtime, so that features can restrict the described version of WAGO Protocol to a subset actually. Example: Standard commands are not available if the feature StandardCommands is not listed within the message body shown above.

## 5.4 DeviceState

```
{
  "MessageType": "DeviceState",
  "WagoProtocol": "1.2",
  "CurrentTime": "2018-10-16T16:15:57.246+02:00",
  "UserSwitch": "State:Run",
  "Leds": [
    {
      "Name": "IO",
      "State": "On",
      "Color": "Green"
    },
    {
      "Name": "MS",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "NET",
      "State": "Blinking",
      "Color": "Red"
    },
    {
      "Name": "NS",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "RUN",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S1",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S2",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S3",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S4",
      "State": "On",
      "Color": "Off"
    },
    {
      "Name": "S5",
      "State": "On",
      "Color": "Off"
    }
  ]
}
```

```
    },  
    {  
      "Name": "SYS",  
      "State": "On",  
      "Color": "Green"  
    },  
    {  
      "Name": "USR",  
      "State": "On",  
      "Color": "Off"  
    }  
  ]  
}
```

PFC will send this message automatically if device status was configured within WBM (feature DeviceStatusOnChange) and status of the LED's or the START/STOP-switch has changed. DataAgent will detect such changes by polling the status continuously.



## 5.5 TagConfiguration

```
{
  "MessageType": "TagConfiguration",
  "WagoProtocol": "1.2",
  "Complete": true,
  "Collections": [
    {
      "Id": 0,
      "CollectionName": "MyCollection0",
      "SamplingControl": "cloud",           // {cloud, iec_only}
      "SamplingIntervalType": "equal",     // {equal, unequal}
      "SampleContent": "all",              // {all, custom}
      "SampleRate": 10000,
      "PublishInterval": 10000,
      "TagData": [
        {
          "Tag": "MyVariable1",
          "Unit": "kg",
          "TypeId": 0,
          "ValueType": "real" // {bool, byte, uint, int, uint,
word,
                                dint, uint, dword, real,
lreal}
        },
        {
          "Tag": "MyVariable2",
          "Unit": "litre",
          "TypeId": 0,
          "ValueType": "word"
        }
      ]
    },
    {
      "Id": 1,
      "CollectionName": "MyCollection1",
      "SampleRate": 10000,
      "PublishInterval": 10000,
      "TagData": [
        {
          "Tag": "MyVariable3",
          "Unit": "u3",
          "TypeId": 0,
          "ValueType": "byte"
        },
        {
          "Tag": "MyVariable4",
          "Unit": "u4",
          "TypeId": 0,
```

```
        "ValueType": "dword"
      }
    ]
  }
]
```

The description of the collections will be sent once to the cloud. The description consists of variable lists. The variable lists are coded within the PLC program. The description of the method how the variable values are sampled within the PLC program is also included within this message. Standard command `SetSamplingInterval` is not available for cloud applications in case `SamplingControl` is `iec_only`.

## 5.6 TagValues

```
{
  "MessageType": "TagValues",
  "WagoProtocol": "1.2",
  "CollectionId": 0,
  "TagData": [
    {
      "Time": "2018-10-16T16:16:18.331+02:00",
      "Values": {
        "MyVariable1": 5.530000,
        "MyVariable2": 10
      }
    }
  ]
}
```

PFC can send this message cyclically by means of the configured `PublishInterval` within the PLC application (see `TagConfiguration` message).

Another use case is when the PLC application samples its data based on some events. This information will also be reflected within the `TagConfiguration` message. In such case the message body of `TagValues` might contain only subset of collection variables.

## 5.7 CommandRegistration

```
{
  "MessageType": "CommandRegistration",
  "WagoProtocol": "1.2",
  "CommandList": [
    {
      "Name": "MyCommand0",
      "Id": 0,
      "RequestParameters": [
        {
          "Name": "Parameter 0 from Cloud to PFC",
          "Type": "real"           // {bool, dint, string, real,
                                udint}
        },
        {
          "Name": "Parameter 1 from Cloud to PFC",
          "Type": "string"
        }
      ],
      "ResponseParameters": [
        {
          "Name": "Response Param 0 PFC to Cloud",
          "Type": "real"
        },
        {
          "Name": "Response Param: Command received",
          "Type": "bool"
        }
      ]
    }
  ]
}
```

The PLC application developer can define specific commands which can be triggered from the cloud application.

This message contains list of the custom commands, which was created within the PLC application, and will be sent to the cloud once. Every command lists its own request- and response parameters. Later on the Cloud Application can request the execution of such a command by sending corresponding command request. After the PLC application have executed the commands it should also send corresponding command response informing about the results.

## 5.8 CommandResponse

```
{
  "MessageType": "CommandResponse",
  "WagoProtocol": "1.2",
  "CommandId": 256,
  "CommandRequestId": 42,
  "CommandResponseParameters": [
    {
      "Name": "myRespP5",
      "Value": "myValue"
    },
    {
      "Name": "myRespP6",
      "Value": "1.713"
    }
  ]
}
```

After receiving a command request from the cloud the DataAgent / PLC application will execute it and optionally confirm the results by sending similar message as shown above.

## 5.9 UpdateStatus

```
{
  "MessageType": "UpdateStatus",
  "WagoProtocol": "1.2",
  "ErrorCode": "int",
  "ErrorMessage": "string",
  "InternalState": "string", // internal state from the fwupdate
                             tool
  "Progress": "int",         // in percent
  "State": "string",
  //FirmwareUpdateInProgress,FirmwareUpdateFinished,FirmwareUpdate
  Failed
  "UpdateType": "string"    // FirmwareUpdate
}
```

This message belongs to the feature “FirmwareUpdate”. It reports the current status of the update process which was triggered by cloud application before by corresponding command request.

Please consider that this message might never arrive your application in case the connection between PFC and your application broke down suddenly.

## 5.10 UpdateFirmware

```
{
  "CommandId": 506,
  "CommandParameters": [
    {
      "Name": "FirmwareStorageAccount",
      "Value": "https://myhost/"
    },
    {
      "Name": "FirmwareStoragePath",
      "Value": "mydir/"
    },
    {
      "Name": "FirmwareControlFile",
      "Value": "package-info.xml"
    },
    {
      "Name": "FirmwareStorageSAS",
      "Value": "?st=affe"
    },
    {
      "Name": "TimeoutPrepared",
      "Value": "120"
    },
    {
      "Name": "TimeoutUnconfirmed",
      "Value": "10"
    }
  ]
}
```

This command request belongs to the feature “FirmwareUpdate” and will trigger the update of the firmware. The cloud application shall pass string parameter values as properly URL-encoded because DataAgent will build the URL to the control file by string concatenation. The resulted URL shall be valid in order to download the control file successfully.

For PFCs, which are connected to the cloud via mobile network, it is recommended to use a quite high value for “TimeoutPrepared” within the command because PFC will usually need that amount of time to download the new firmware files. The unit of timeout parameter is minutes.

The CommandResponse of this standard command includes the following parameter:

- “Name”: “Received”  
“Value”: “bool”

## 5.11 StartRemoteAccessMediator

```
{
  "CommandId": 509,
  "CommandParameters": [
    {
      "Name": "URL",
      "Value": "<Proxy server URL>"
    }
  ]
}
```

This command belongs to the feature “RemoteAccess” and enables the Cloud application to start the remote access mediator. With the help of this functionality the user is able to call the WBM and the WebVisu remotely.

The CommandResponse of this standard command includes the following parameter:

- “Name”: “Started”  
“Value”: “bool”

## 5.12 StopRemoteAccessMediator

```
{
  "CommandId": 510,
  "CommandParameters": [
  ]
}
```

This command belongs to the feature “RemoteAccess” and enables the Cloud application to stop the remote access mediator.

The CommandResponse of this standard command includes the following parameter:

- “Name”: “Stopped”  
“Value”: “bool”

## 6 Protocol Version 1.4.x

### 6.1 Overview messages

In general some messages are as in previous protocol. New messages are listed below.

Table 7: Overview messages

Message	Origin	Application properties (WAGO Cloud, Azure)	MQTT topic (AnyMQTT, AWS)	MQTT topic IBM Cloud
<i>Heartbeat</i>	DataAgent	MessageType=Heartbeat ProtocolVersion=1.4.{x}	-	-
<i>DeviceInfo</i>	DataAgent	MessageType=DeviceInfo ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ DeviceInfo	iot-2/evt/ DeviceInfo/f mt/json
<i>DeviceState</i>	DataAgent	MessageType=DeviceState ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ DeviceState	iot-2/evt/ DeviceState/ fmt/json
<i>TagConfiguration</i>	PLC App	MessageType=TagConfiguration ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ TagConfiguration	iot-2/evt/ TagConfigur ation/ fmt/json
<i>TagValues</i>	PLC App	MessageType=TagValues ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ TagValues	iot-2/evt/ TagValues/f mt/json
<i>CommandRegistration</i>	PLC App	MessageType=CommandRegistration ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ CommandRegistration	-
<i>CommandResponse</i>	PLC App / DataAgent	MessageType=CommandResponse ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ CommandResponse	-
<i>GetDeviceInfo</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-
<i>SetSamplingInterval</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-
<i>SetPublishingInterval</i>	Cloud App	CommandRequestId={I} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-

<i>GetTagConfiguration</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-
<i>StopTelemetry</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-
<i>StartTelemetry</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-
<i>SetHeartbeatInterval</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-
<i>DeleteController</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-
<i>MyCommand0</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}	1.4.{x}/{CID}/ Commands	-

{l} is a placeholder for identifier given to the command request  
 {CID} represents the "Client ID" entered in WBM.

{x} indicates whether the data is sent uncompressed or compressed. There are two possibilities:

- 1.4.0: uncompressed
- 1.4.1: compressed with GZIP (contains the same JSON format)

## 6.2 WAGO Cloud specific messages

Message	Origin	Application properties (WAGO Cloud)
<i>UpdateStatus</i>	DataAgent	MessageType=UpdateStatus ProtocolVersion=1.4.{x}
<i>StartRemoteAccessMediator</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}
<i>StopRemoteAccessMediator</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}
<i>UpdateFirmware</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}
<i>UpdateApplication</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}
<i>UpdateDevice</i>	Cloud App	CommandRequestId={l} ProtocolVersion=1.4.{x}



## 6.3 DeviceInfo

```
{
  "MessageType": "DeviceInfo",
  "WagoProtocol": "1.4",
  "CurrentRuntimeVersion": "string",
  "DataAgentVersion": "string",
  "FirmwareVersion": "string",
  "HardwareReleaseIndex": "string",
  "NTP": "string",
  "LicenseInfo": "string",
  "OrderNumber": "string",
  "ProductDescription": "string",
  "WebserverDefault": "string",
  "WebserverStateCodesys": "string",
  "WebserverStateECockpit": "string",
  "NetworkInterfaces": [
    {
      "Name": "string",
      "Mac": "string",
      "Ip": "string",
      "Netmask": "string"
    }
  ],
  "IoModules": [
    {
      "Position": 0,
      "Code": 0
    }
  ],
  "ConfiguredFeatures": [
    "DeviceStatusOnChange",
    "StandardCommands",
    "FirmwareUpdate",          // optional
    "RemoteAccess",           // optional
    "ApplicationUpdate",       // optional
    "DeviceUpdate"             // optional
  ]
}
```

PFC will send this message automatically on start if device information was configured within WBM. DataAgent will determine the list of configured features at runtime, so that features can restrict the described version of WAGO Protocol to a subset actually. Example: Standard commands are not available if the feature "StandardCommands" is not listed within the message body shown above.

## 6.4 TagConfiguration

```
{
  "MessageType": "TagConfiguration",
  "WagoProtocol": "1.4",
  "Complete": true,
  "Collections": [
    {
      "Id": 0,
      "CollectionName": "MyCollection0",
      "SamplingControl": "cloud",           // {cloud, iec_only}
      "SamplingIntervalType": "equal",     // {equal, unequal}
      "SampleContent": "all",              // {all, custom}
      "SampleRate": 10000,
      "PublishInterval": 10000,
      "TagData": [
        {
          "Tag": "MyVariable1",
          "Unit": "kg",
          "TypeId": 0,
          "ValueType": "real" // {bool, byte, uint, int, uint,
word,
                                dint, uint, dword, real,
lreal, string}
        },
        {
          "Tag": "MyVariable2",
          "Unit": "litre",
          "TypeId": 0,
          "ValueType": "word"
        },
        {
          "Tag": "MyVariable5",
          "Unit": "u5",
          "TypeId": 0,
          "ValueType": "string"
        }
      ]
    },
    {
      "Id": 1,
      "CollectionName": "MyCollection1",
      "SampleRate": 10000,
      "PublishInterval": 10000,
      "TagData": [
        {
          "Tag": "MyVariable3",
          "Unit": "u3",
          "TypeId": 0,
```

```

        "ValueType": "byte"
      },
      {
        "Tag": "MyVariable4",
        "Unit": "u4",
        "TypeId": 0,
        "ValueType": "dword"
      }
    ]
  }
]
}

```

The description of the collections will be sent once to the cloud. The description consists of variable lists. The variable lists are coded within the PLC program. The description of the method how the variable values are sampled within the PLC program is also included within this message. Standard command `SetSamplingInterval` is not available for cloud applications in case `SamplingControl` is `iec_only`.

## 6.5 TagValues

```

{
  "MessageType": "TagValues",
  "WagoProtocol": "1.4",
  "CollectionId": 0,
  "TagData": [
    {
      "Time": "2018-10-16T16:16:18.331+02:00",
      "Values": {
        "MyVariable1": 5.530000,
        "MyVariable2": 10,
        "MyVariable5": "MyString",
      }
    }
  ]
}

```

PFC can send this message cyclically by means of the configured `PublishInterval` within the PLC application (see `TagConfiguration` message).

Another use case is when the PLC application samples its data based on some events. This information will also be reflected within the `TagConfiguration` message. In such case the message body of `TagValues` might contain only subset of collection variables.

## 6.6 UpdateFirmware

```
{
  "CommandId": 506,
  "CommandParameters": [
    {
      "Name": "FirmwareStorageAccount",
      "Value": "https://myhost/"
    },
    {
      "Name": "FirmwareStoragePath",
      "Value": "mydir/"
    },
    {
      "Name": "FirmwareControlFile",
      "Value": "package-info.xml"
    },
    {
      "Name": "FirmwareStorageSAS",
      "Value": "?st=affe"
    },
    {
      "Name": "TimeoutPrepared",
      "Value": "120"
    },
    {
      "Name": "TimeoutUnconfirmed",
      "Value": "10"
    }
  ]
}
```

This command request belongs to the feature “FirmwareUpdate” and will trigger the update of the firmware.

For PFCs, which are connected to the cloud via mobile network, it is recommended to use a quite high value for “TimeoutPrepared” within the command because PFC will usually need that amount of time to download the new firmware files. The unit of timeout parameter is minutes.

The CommandResponse of this standard command includes the following parameter:

- “Name”: “UpdateStatus”  
“Value”: {“Started” | “Rejected”}

## 6.7 UpdateApplication

```
{
  "CommandId": 507,
  "CommandParameters": [
    {
      "Name": "FirmwareStorageAccount",
      "Value": "https://myhost/"
    },
    {
      "Name": "FirmwareStoragePath",
      "Value": "mydir/"
    },
    {
      "Name": "FirmwareControlFile",
      "Value": "package-info.xml"
    },
    {
      "Name": "FirmwareStorageSAS",
      "Value": "?st=affe"
    },
    {
      "Name": "Timeout",
      "Value": "60"
    }
  ]
}
```

This command request belongs to feature “StandardCommands” and will trigger the update of the IEC application. The cloud application shall pass string parameter values as properly URL-encoded because DataAgent will build the URL to the control file by string concatenation. The resulted URL shall be valid in order to download the control file successfully.

For PFCs, which are connected to the cloud via mobile network, it is recommended to use a quite high value for “Timeout” within the command because PFC will usually need that amount of time to download the files. The unit of timeout parameter is minutes.

The CommandResponse of this standard command includes the following parameter:

- “Name”: “UpdateStatus”  
“Value”: {“Started” | “Rejected”}

## 6.8 UpdateDevice

```
{
  "CommandId": 508,
  "CommandParameters": [
    {
      "Name": "FirmwareStorageAccount",
      "Value": "https://myhost/"
    },
    {
      "Name": "FirmwareStoragePath",
      "Value": "mydir/"
    },
    {
      "Name": "FirmwareControlFile",
      "Value": "package-info.xml"
    },
    {
      "Name": "FirmwareStorageSAS",
      "Value": "?st=affe"
    },
    {
      "Name": "TimeoutPrepared",
      "Value": "120"
    },
    {
      "Name": "TimeoutUnconfirmed",
      "Value": "60"
    }
  ]
}
```

This command request belongs to feature “StandardCommands” and will trigger the update of the firmware and IEC application as atomic change. That means either both are successful or no one. The cloud application shall pass string parameter values as properly URL-encoded because DataAgent will build the URL to the files by string concatenation. The resulted URL shall be valid in order to download the control file successfully.

For PFCs, which are connected to the cloud via mobile network, it is recommended to use a quite high values for “TimeoutPrepared” and “TimeoutUnconfirmed” within the command because PFC will usually need that amount of time to download the files. The unit of timeout parameters is minutes.

The CommandResponse of this standard command includes the following parameter:

- “Name”: “UpdateStatus”  
“Value”: {“Started” | “Rejected”}



---

## List of Figures

Figure 3-1: Protocol handshake messages .....	10
Figure 3-2: Handshake sequence - Default case.....	12
Figure 3-3: Handshake sequence - Cloud Application legacy version .....	12
Figure 3-4: Handshake sequence - DataAgent protocol version is not supported by the Cloud Application .....	13
Figure 3-5: Handshake sequence - DataAgent connection is interrupted while handshake was initiated.....	13



---

# List of Tables

Table 1: Number Notation ..... 7

Table 2: Font Conventions ..... 7

Table 3: Revision History ..... 9

Table 4: Overview messages .....10

Table 5: Overview messages .....14

Table 6: Overview messages .....28

Table 7: Overview messages .....39





WAGO Kontakttechnik GmbH & Co. KG  
Postfach 2880 • 32385 Minden  
Hansastraße 27 • 32423 Minden  
Phone: +49 571 887 – 0  
Fax: +49 571 887 – 844169  
E-Mail: [info@wago.com](mailto:info@wago.com)  
Internet: [www.wago.com](http://www.wago.com)