



Co-funded by the  
Erasmus+ Programme  
of the European Union



# Cyber-Physical Industrial Systems

---

Module 3 Session 3

**IoT communication protocols basics**

Lecture





# MQTT – the IoT protocol



## MQTT Version 5.0

OASIS Standard

07 March 2019

*MQTT Version 5.0*. Edited by Andrew Banks, Ed Briggs, Ken Borgendale, and Rahul Gupta. 07 March 2019. OASIS Standard. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>. Latest version: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.

Co-funded by the  
Erasmus+ Programme  
of the European Union





# Data representation

## 1.5.1 Bits

Bits in a byte are labelled 7 to 0. Bit number 7 is the most significant bit, the least significant bit is assigned bit number 0.

## 1.5.2 Two Byte Integer

Two Byte Integer data values are 16-bit unsigned integers in big-endian order: the high order byte precedes the lower order byte. This means that a 16-bit word is presented on the network as Most Significant Byte (MSB), followed by Least Significant Byte (LSB).

## 1.5.3 Four Byte Integer

Four Byte Integer data values are 32-bit unsigned integers in big-endian order: the high order byte precedes the successively lower order bytes. This means that a 32-bit word is presented on the network as Most Significant Byte (MSB), followed by the next most Significant Byte (MSB), followed by the next most Significant Byte (MSB), followed by Least Significant Byte (LSB).

## 1.5.4 UTF-8 Encoded String

Text fields within the MQTT Control Packets described later are encoded as UTF-8 strings. UTF-8 [\[RFC3629\]](#) is an efficient encoding of Unicode [\[Unicode\]](#) characters that optimizes the encoding of ASCII characters in support of text-based communications.

Co-funded by the  
Erasmus+ Programme  
of the European Union





# Data representation

## 1.5.5 Variable Byte Integer

The Variable Byte Integer is encoded using an encoding scheme which uses a single byte for values up to 127. Larger values are handled as follows. The least significant seven bits of each byte encode the data, and the most significant bit is used to indicate whether there are bytes following in the representation. Thus, each byte encodes 128 values and a "continuation bit". The maximum number of bytes in the Variable Byte Integer field is four. **The encoded value MUST use the minimum number of bytes necessary to represent the value [MQTT-1.5.5-1].** This is shown in Table 1-1 Size of Variable Byte Integer.

## 1.5.6 Binary Data

Binary Data is represented by a Two Byte Integer length which indicates the number of data bytes, followed by that number of bytes. Thus, the length of Binary Data is limited to the range of 0 to 65,535 Bytes.

## 1.5.7 UTF-8 String Pair

A UTF-8 String Pair consists of two UTF-8 Encoded Strings. This data type is used to hold name-value pairs. The first string serves as the name, and the second string contains the value.





## 2.1. MQTT Control Packet Structure

---

---

Fixed Header, present in all MQTT Control Packets

---

Variable Header, present in some MQTT Control Packets

---

Payload, present in some MQTT Control Packets

---





## 2.1.1. MQTT Fixed Header Structure

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type				Flags specific to each MQTT Control Packet type			
byte 2...	Remaining Length							





## 2.1.2. MQTT Control Packet Types

Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client to Server	Connection request
CONNACK	2	Server to Client	Connect acknowledgment
PUBLISH	3	Client to Server or Server to Client	Publish message
PUBACK	4	Client to Server or Server to Client	Publish acknowledgment (QoS 1)

SUBSCRIBE	8	Client to Server	Subscribe request
SUBACK	9	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
UNSUBACK	11	Server to Client	Unsubscribe acknowledgment
PINGREQ	12	Client to Server	PING request
PINGRESP	13	Server to Client	PING response

Co-funded by the  
Erasmus+ Programme  
of the European Union





## 2.1.3. MQTT Flags

MQTT Control Packet	Fixed Header flags	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT v5.0	DUP	QoS		RETAIN
PUBACK	Reserved	0	0	0	0
PUBREC	Reserved	0	0	0	0
PUBREL	Reserved	0	0	1	0
PUBCOMP	Reserved	0	0	0	0
SUBSCRIBE	Reserved	0	0	1	0
SUBACK	Reserved	0	0	0	0
UNSUBSCRIBE	Reserved	0	0	1	0

UNSUBACK	Reserved	0	0	0	0
PINGREQ	Reserved	0	0	0	0
PINGRESP	Reserved	0	0	0	0
DISCONNECT	Reserved	0	0	0	0
AUTH	Reserved	0	0	0	0







# CONNECT Variable Header

	Description	7	6	5	4	3	2	1	0
Protocol Name									
byte 1	Length MSB (0)	0	0	0	0	0	0	0	0
byte 2	Length LSB (4)	0	0	0	0	0	1	0	0
byte 3	'M'	0	1	0	0	1	1	0	1
byte 4	'Q'	0	1	0	1	0	0	0	1
byte 5	'T'	0	1	0	1	0	1	0	0
byte 6	'T'	0	1	0	1	0	1	0	0

	Description	7	6	5	4	3	2	1	0
Protocol Level									
byte 7	Version(5)	0	0	0	0	0	1	0	1

Co-funded by the  
Erasmus+ Programme  
of the European Union





# CONNECT Variable Header

## Connect Flags

Bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Start	Reserved
byte 8	X	X	X	X	X	X	X	0

## Properties

Property Length  
Session Expiry Interval  
Receive Maximum  
Maximum Packet Size  
Topic Alias Maximum  
Request Response Information  
Request Problem Information

.....

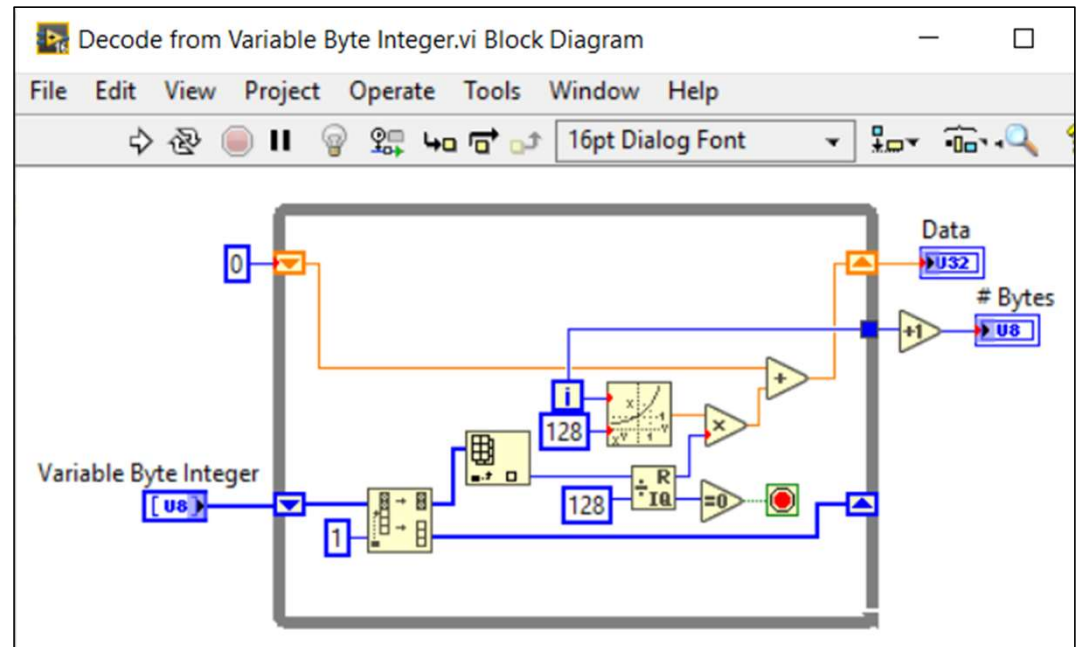
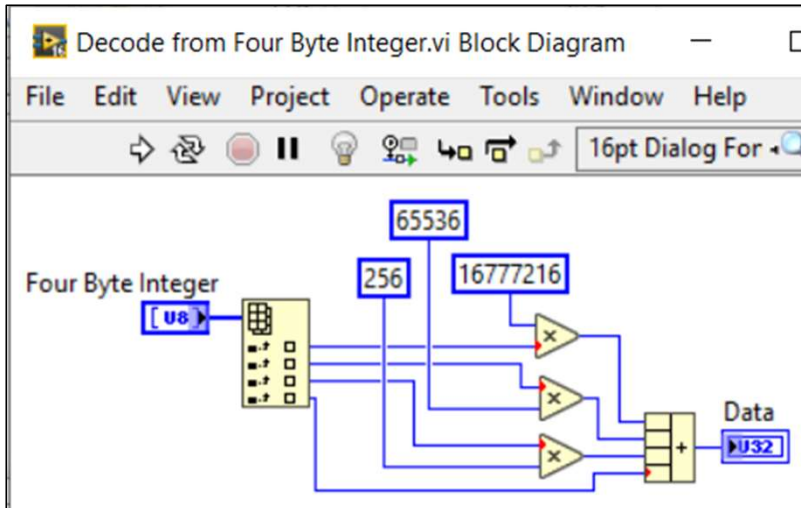
## Keep Alive

Bit	7	6	5	4	3	2	1	0
byte 9	Keep Alive MSB							
byte 10	Keep Alive LSB							

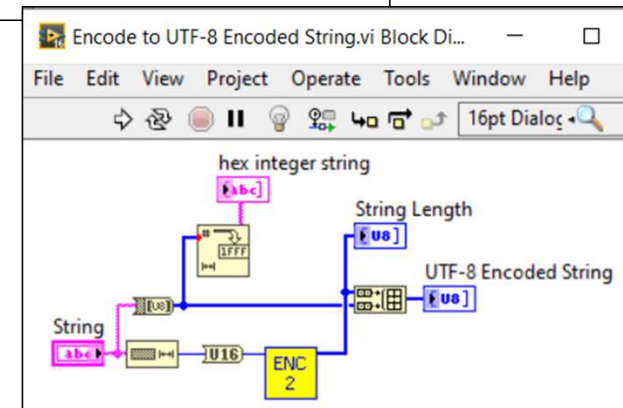
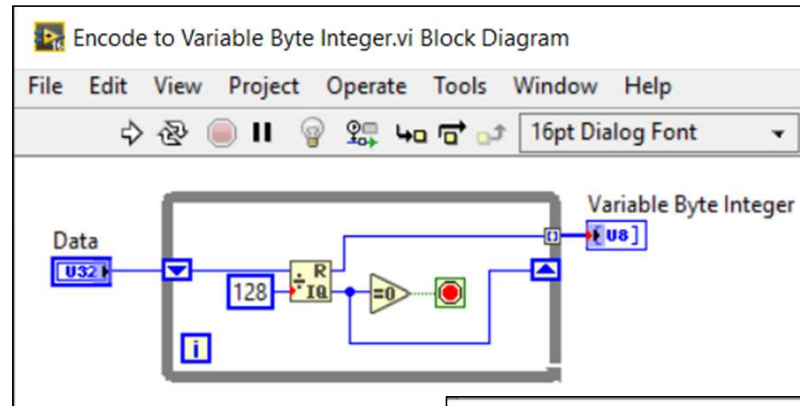
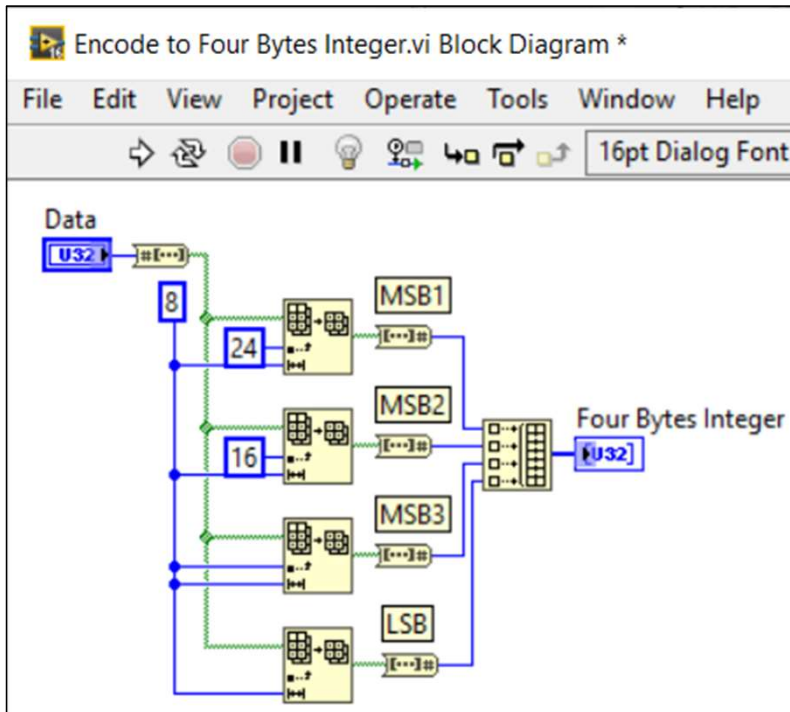
Co-funded by the  
Erasmus+ Programme  
of the European Union



# LabVIEW Implementation

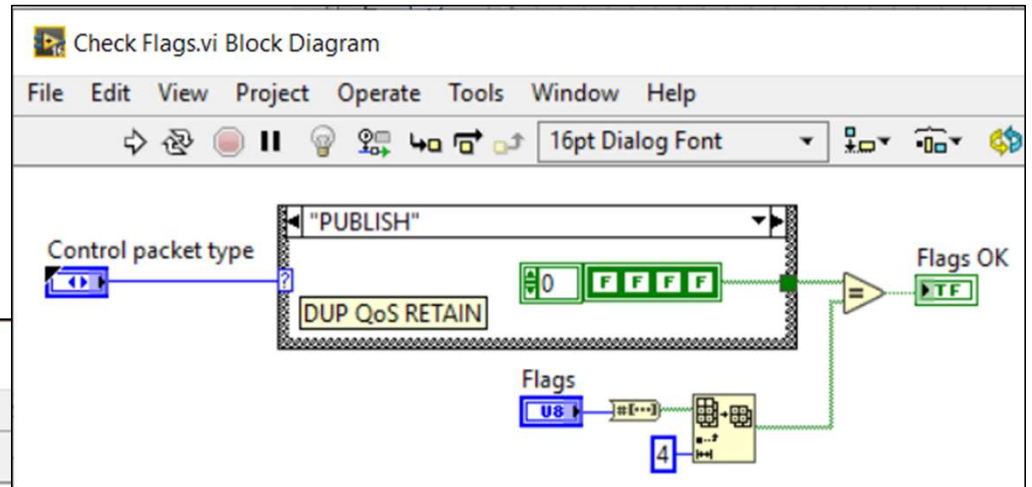
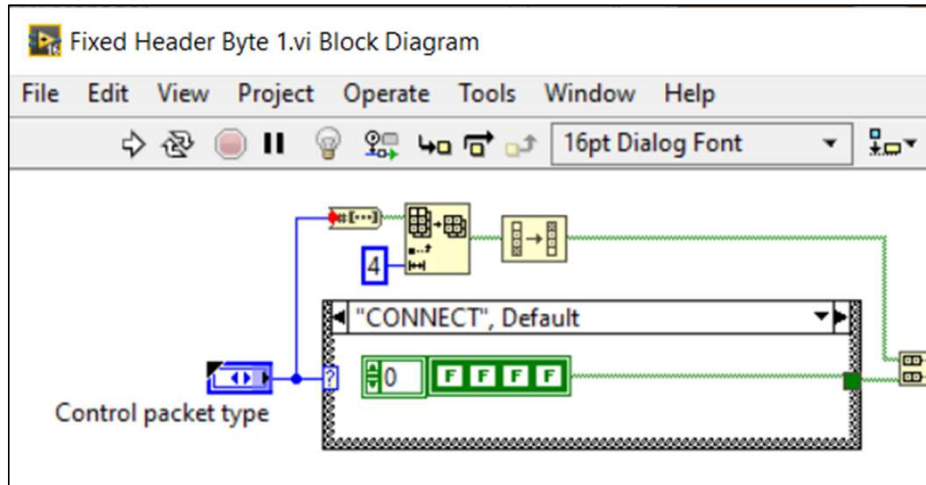


# LabVIEW Implementation





# LabVIEW Implementation



Co-funded by the  
Erasmus+ Programme  
of the European Union

