

Figure 2.1 – Structure of an MQTT Control Packet

Fixed header, present in all MQTT Control Packets
Variable header, present in some MQTT Control Packets
Payload, present in some MQTT Control Packets

3 MQTT Control Packets

3.1 CONNECT – Client requests a connection to a Server

3.1.1 Fixed header

Figure 3.1 – CONNECT Packet fixed header

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type (1)				Reserved			
	0	0	0	1	0	0	0	0
byte 2...	Remaining Length							

Remaining Length field

Remaining Length is the length of the variable header (10 bytes) plus the length of the Payload. It is encoded in the manner described in section 2.2.3.

3.1.2 Variable header

The variable header for the CONNECT Packet consists of four fields in the following order: Protocol Name, Protocol Level, Connect Flags, and Keep Alive.

3.1.2.1 Protocol Name

Figure 3.2 - Protocol Name bytes

	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

3.1.2.2 Protocol Level

Figure 3.3 - Protocol Level byte

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

3.1.2.3 Connect Flags

The Connect Flags byte contains a number of parameters specifying the behavior of the MQTT connection. It also indicates the presence or absence of fields in the payload.

Figure 3.4 - Connect Flag bits

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

3.1.2.10 Keep Alive

Figure 3.5 Keep Alive bytes

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

3.1.3 Payload

The payload of the CONNECT Packet contains one or more length-prefixed fields, whose presence is determined by the flags in the variable header. These fields, if present, MUST appear in the order Client Identifier, Will Topic, Will Message, User Name, Password [MQTT-3.1.3-1].

3.1.3.1 Client Identifier

The Client Identifier (ClientId) identifies the Client to the Server. Each Client connecting to the Server has a unique ClientId. The ClientId MUST be used by Clients and by Servers to identify state that they hold relating to this MQTT Session between the Client and the Server [MQTT-3.1.3-2].

The Client Identifier (ClientId) MUST be present and MUST be the first field in the CONNECT packet payload [MQTT-3.1.3-3].

3.1.3.4 User Name

If the User Name Flag is set to 1, this is the next field in the payload. The User Name MUST be a UTF-8 encoded string as defined in Section 1.5.3 [MQTT-3.1.3-11]. It can be used by the Server for authentication and authorization.

3.1.3.5 Password

If the Password Flag is set to 1, this is the next field in the payload. The Password field contains 0 to 65535 bytes of binary data prefixed with a two byte length field which indicates the number of bytes used by the binary data (it does not include the two bytes taken up by the length field itself).

Figure 3.7 - Password bytes

Bit	7	6	5	4	3	2	1	0
byte 1	Data length MSB							
byte 2	Data length LSB							
byte 3	Data, if length > 0.							

Example without Username and Password–

Fixed Header

0x10 # CONNECT
0x17 # Remaining length = 23 bytes (below)

Variable Header

0x00 0x04 # Length of "MQTT"
0x4D 0x51 0x54 0x54 # "MQTT"
0x04 # Protocol Level = 4
0x02 # Flags = Clean Session
0x00 0x3C # Keep Alive = 60 (0x003C)

Payload (Client ID, No Username or Password)

0x00 0x0B # Length of ClientID (11)
0x53 0x54 0x4D 0x33 0x32 0x43 0x6C 0x69 0x65 0x6E 0x74 # "STM32Client"

Full CONNECT Packet in String Format

\x10\x17\x00\x04MQTT\x04\x02\x00\x3C\x00\x0BSTM32Client

Example with Username and Password–

Fixed Header

0x10 # CONNECT
0x26 # Remaining length = 38 bytes (below)

Variable Header

0x00 0x04 # Length of "MQTT"
0x4D 0x51 0x54 0x54 # "MQTT"
0x04 # Protocol Level = 4
0xC2 # Connect Flags (username+password+clean session)
0x00 0x3C # Keep Alive = 60 (0x003C)

Payload (Client ID, No Username or Password)

0x00 0x0B	# Length of ClientID (11)
0x53 0x54 0x4D 0x33 0x32 0x43 0x6C 0x69 0x65 0x6E 0x74	# "STM32Client"
0x00 0x05	# Length of Username(5)
0x75 0x73 0x65 0x72 0x31	# "user1"
0x00 0x05	# Length of Password(5)
0x70 0x61 0x73 0x73 0x31	# "pass1"

Full CONNECT Packet in String Format

\x10\x26\x00\x04MQTT\x04\xC2\x00\x3C\x00\x0BSTM32Client\x00\x05user1\x00\x05pass1

3.2 CONNACK – Acknowledge connection request

The CONNACK Packet is the packet sent by the Server in response to a CONNECT Packet received from a Client. **The first packet sent from the Server to the Client MUST be a CONNACK Packet [MQTT-3.2.0-1].**

If the Client does not receive a CONNACK Packet from the Server within a reasonable amount of time, the Client SHOULD close the Network Connection. A "reasonable" amount of time depends on the type of application and the communications infrastructure.

3.2.1 Fixed header

The fixed header format is illustrated in [Figure 3.8 – CONNACK Packet fixed header](#).

Figure 3.8 – CONNACK Packet fixed header

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet Type (2)				Reserved			
	0	0	1	0	0	0	0	0
byte 2	Remaining Length (2)							
	0	0	0	0	0	0	1	0

3.2.2 Variable header

The variable header format is illustrated in [Figure 3.9 – CONNACK Packet variable header](#).

Figure 3.9 – CONNACK Packet variable header

	Description	7	6	5	4	3	2	1	0	
Connect Acknowledge Flags		Reserved								SP ¹
byte 1		0	0	0	0	0	0	0	X	
Connect Return code										
byte 2		X	X	X	X	X	X	X	X	

3.2.2.1 Connect Acknowledge Flags

Byte 1 is the "Connect Acknowledge Flags". Bits 7-1 are reserved and MUST be set to 0.

Bit 0 (SP¹) is the Session Present Flag.

Table 3.1 – Connect Return code values

Value	Return Code Response	Description
0	0x00 Connection Accepted	Connection accepted
1	0x01 Connection Refused, unacceptable protocol version	The Server does not support the level of the MQTT protocol requested by the Client
2	0x02 Connection Refused, identifier rejected	The Client identifier is correct UTF-8 but not

Broker Response

If success, you should see broker reply with **CONNACK** (0x20 0x02 0x00 0x00)

3.3 PUBLISH – Publish message

A PUBLISH Control Packet is sent from a Client to a Server or from Server to a Client to transport an Application Message.

3.3.1 Fixed header

Figure 3.10 – PUBLISH Packet fixed header illustrates the fixed header format:

Figure 3.10 – PUBLISH Packet fixed header

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type (3)				DUP flag	QoS level		RETAIN
	0	0	1	1	X	X	X	X
byte 2	Remaining Length							

3.3.2 Variable header

The variable header contains the following fields in the order: Topic Name, Packet Identifier.

3.3.2.1 Topic Name

The Topic Name identifies the information channel to which payload data is published.

3.3.3 Payload

The Payload contains the Application Message that is being published. The content and format of the data is application specific. The length of the payload can be calculated by subtracting the length of the variable header from the Remaining Length field that is in the Fixed Header. It is valid for a PUBLISH Packet to contain a zero length payload.

Example-

Fixed Header

\x30 # PUBLISH (0x3 << 4) | QoS0 | retain=0
\x20 # Remaining length = 32 bytes

Variable Header

\x00\x13 # Topic length = 0x0013 (19)
controllerstech/test # Topic Name

Payload/Message

Hello STM32 #Message to be sent

Full Publish Message

\x30\x20\x00\x13controllerstech/testHello STM32

NO PUBACK for QoS = 0.

3.12 PINGREQ – PING request

The PINGREQ Packet is sent from a Client to the Server. It can be used to:

1. Indicate to the Server that the Client is alive in the absence of any other Control Packets being sent from the Client to the Server.
2. Request that the Server responds to confirm that it is alive.
3. Exercise the network to indicate that the Network Connection is active.

This Packet is used in Keep Alive processing, see Section 3.1.2.10 for more details.

3.12.1 Fixed header

Figure 3.33 – PINGREQ Packet fixed header

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type (12)				Reserved			
	1	1	0	0	0	0	0	0
byte 2	Remaining Length (0)							
	0	0	0	0	0	0	0	0

3.12.2 Variable header

The PINGREQ Packet has no variable header.

3.12.3 Payload

The PINGREQ Packet has no payload.

Example-

Command- \xC0\x00

Response- \xD0\x00

3.13 PINGRESP – PING response

A PINGRESP Packet is sent by the Server to the Client in response to a PINGREQ Packet. It indicates that the Server is alive.

This Packet is used in Keep Alive processing, see Section 3.1.2.10 for more details.

3.13.1 Fixed header

Figure 3.34 – PINGRESP Packet fixed header

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type (13)				Reserved			
	1	1	0	1	0	0	0	0
byte 2	Remaining Length (0)							
	0	0	0	0	0	0	0	0