

NJ-NX PLC'lerde MQTT Haberleşmesi

İÇİNDEKİLER

- Giriş
- Uygulama Adımları
 - MQTT Broker Kurulumu
 - PLC'nin MQTT Broker'a Bağlanması
 - Remote Broker ile Local Broker Arasında Köprü (Bridge) Bağlantısı

Giriş

Bu dökümanda MQTT desteği olan cihazlar ile MQTT haberleşmesi adımları anlatılacaktır.

MQTT, makineler arası ve makineden IT sistemlerine veri aktarımını mümkün kılan hafif ve yaygın kullanılan bir haberleşme protokolüdür. Bu uluslararası standart, Endüstri 4.0 ile güçlü bir ilişkiye sahiptir. Yaygın kullanımı ve düşük bant genişliği ihtiyacı sayesinde, IoT ve endüstriyel otomasyon uygulamalarında önemli bir rol oynamaktadır. MQTT, publish/subscribe yapısı ile cihazlar ve IT sistemleri arasında açık, esnek ve güvenli bir haberleşme sağlar. Bu protokol, NX ve NJ Sysmac kontrolörlerde de kullanıcıların hızlı, güvenilir ve etkin veri iletişimi kurmasına imkân vermektedir.

Uygulama Adımları

1. MQTT Broker Kurulumu

- Haberleşmenin merkezinde broker bulunur. Cihazlar arasındaki tüm veri akışı bu broker üzerinden gerçekleşir.
- Popüler broker’lar: **Mosquitto**, **HiveMQ**, **EMQX**.
- Broker, buluta veya yerel bilgisayara kurularak çalıştırılır. Bu uygulamada Mosquitto broker kullanılarak local bir server oluşturulacaktır.



a) Mosquitto Kurulumu

Resmî Eclipse Mosquitto bağlantısından yazılım indirilir ve işletim sistemine uygun şekilde kurulumu yapılır. Aşağıdaki bağlantıdan Mosquitto programı indirilebilir:

<https://Mosquitto.org/download/>

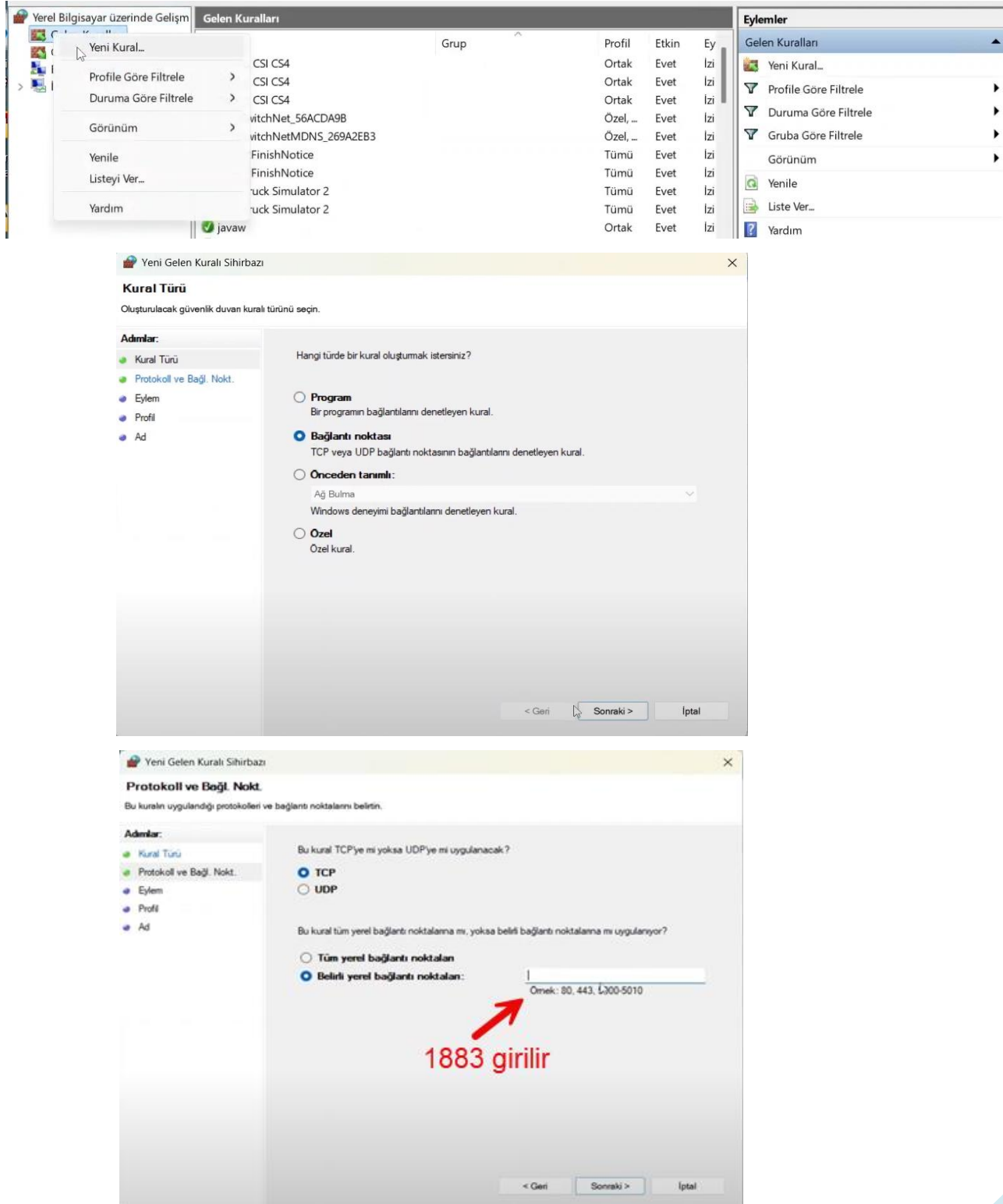
b) Port Ayarlarının Yapılması

- MQTT’nin varsayılan portu 1883 (şifresiz) olarak kullanılacağından, *Mosquitto.conf* dosyasında **listener 1883** satırı ile ayarlanır.
- Bu portun ağ güvenlik duvarında (firewall) ve işletim sistemi üzerinde açık olması gerekir.

İlgili portu açmak için; sırasıyla aşağıda gözüken işlemler yapılmalıdır.

Windows>Başlat(Start) üzerinden “*Gelişmiş Güvenlik Özellikli Windows Defender Güvenlik / Windows Defender Firewall with Advanced Security*” uygulaması çalıştırılır.





MQTT iletişimi için varsayılan olarak kullanılan **1883** numaralı port üzerinden dinleme yapılmaktadır ve bu nedenle yapılandırmada bu portun belirtilmesi gerekmektedir.

Yeni Gelen Kuralı Sihirbazı

Eylem

Bağlantı kuralda belirtilen koşullarla eşleştiginde yapılacak eylemi belirtin.

Adımlar:

- Kural Türü
- Protokoll ve Bağl. Nokt.
- Eylem**
- Profil
- Ad

Bir bağlantı belirtilen koşulları sağladığında gerçekleştirilmesi gereken eylem nedir?

☒ **Bağlantıya izin ver**
Buna IPsec ile korunanların yanı sıra korunmayan bağlantılar da dahildir.

☐ **Bağlantı güveniyse izin ver**
Buna yalnızca IPsec kullanarak kimliği doğrulanmış bağlantılar dahildir. Bağlantıların güvenliği IPsec özelliklerindeki ayarlar ve Bağlantı Güvenliği Kuralı düğümündeki kurallar kullanılarak sağlanır.
[Özelleştir...](#)

☐ **Bağlantıyı engelle**

< Geri Sonraki > İptal

Yeni Gelen Kuralı Sihirbazı

Profil

Bu kuralın uygulandığı profilleri belirtin.

Adımlar:

- Kural Türü
- Protokoll ve Bağl. Nokt.
- Eylem
- Profil**
- Ad

Bu kural ne zaman uygulanacak?

☒ **Etki alanı**
Bilgisayar kuruluş etki alanına bağlandığında uygulanır.

☒ **Özel**
Bilgisayar ev veya işyeri gibi bir özel ağ konumuna bağlandığında uygulanır.

☒ **Ortak**
Bilgisayar bir ortak ağ konumuna bağlandığında uygulanır.

< Geri Sonraki > İptal

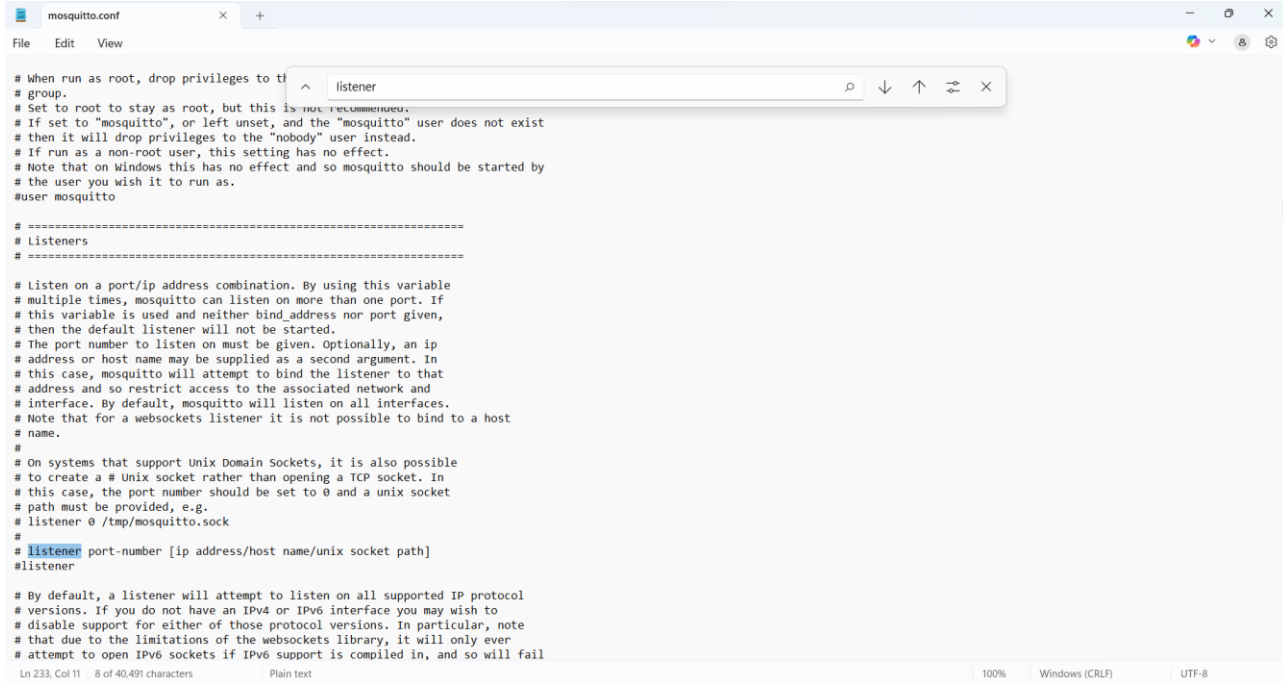
MQTT iletişimi için varsayılan port olan 1883 yapılandırmada etkinleştirilmiş ve dinleme için açılmıştır.

c) Mosquitto.conf Dosyasına Girip Konfigürasyon Ayarlama

- Öncelikle Mosquitto'nun kayıtlı olduğu klasöre gelinir.
- Burada bulunan *Mosquitto.conf* dosyası (notepad ile) açılır.
- Mosquitto.conf dosyasında **listener** satırını kullanarak IP ve port ayarı yapılır.

```
# listener port-number [ip address/host name/unix socket path]
listener 1883 192.168.250.55
```

Burada **host name** local serverda (bilgisayarda) kurulu olan Mosquitto olduğu için bilgisayarın IP adresidir.



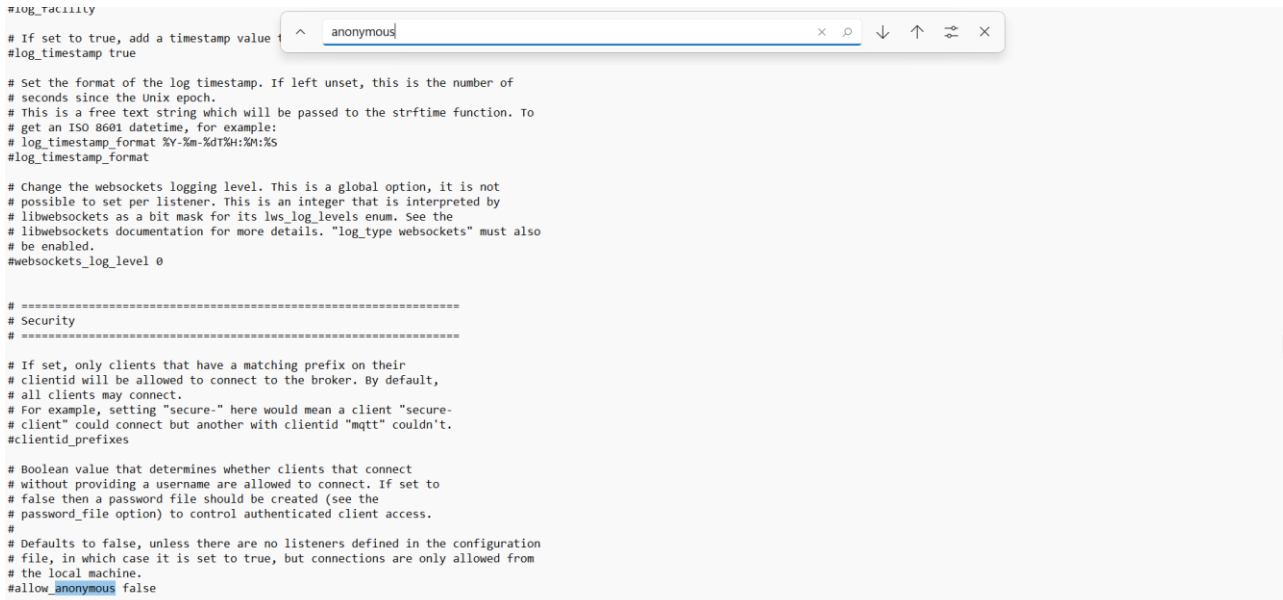
```
# When run as root, drop privileges to the
# group.
# Set to root to stay as root, but this is not recommended.
# If set to "mosquitto", or left unset, and the "mosquitto" user does not exist
# then it will drop privileges to the "nobody" user instead.
# If run as a non-root user, this setting has no effect.
# Note that on Windows this has no effect and so mosquitto should be started by
# the user you wish it to run as.
#user mosquitto

# =====
# Listeners
# =====

# Listen on a port/ip address combination. By using this variable
# multiple times, mosquitto can listen on more than one port. If
# this variable is used and neither bind_address nor port given,
# then the default listener will not be started.
# The port number to listen on must be given. Optionally, an ip
# address or host name may be supplied as a second argument. In
# this case, mosquitto will attempt to bind the listener to that
# address and so restrict access to the associated network and
# interface. By default, mosquitto will listen on all interfaces.
# Note that for a websockets listener it is not possible to bind to a host
# name.
#
# On systems that support Unix Domain Sockets, it is also possible
# to create a # Unix socket rather than opening a TCP socket. In
# this case, the port number should be set to 0 and a unix socket
# path must be provided, e.g.
# listener 0 /tmp/mosquitto.sock
#
# listener port-number [ip address/host name/unix socket path]
#listener

# By default, a listener will attempt to listen on all supported IP protocol
# versions. If you do not have an IPv4 or IPv6 interface you may wish to
# disable support for either of those protocol versions. In particular, note
# that due to the limitations of the websockets library, it will only ever
# attempt to open IPv6 sockets if IPv6 support is compiled in, and so will fail
```

Aynı yerel ağdaki tüm kullanıcıların bağlanabilmesi amacıyla kimliksiz (şifresiz) girişleri (**allow_anonymous true**) aktif edilir.



```
#log_timestamp
# If set to true, add a timestamp value to the log
#log_timestamp true

# Set the format of the log timestamp. If left unset, this is the number of
# seconds since the Unix epoch.
# This is a free text string which will be passed to the strftime function. To
# get an ISO 8601 datetime, for example:
# log_timestamp_format %Y-%m-%dT%H:%M:%S
#log_timestamp_format

# Change the websockets logging level. This is a global option, it is not
# possible to set per listener. This is an integer that is interpreted by
# libwebsockets as a bit mask for its lws_log_levels enum. See the
# libwebsockets documentation for more details. "log_type websockets" must also
# be enabled.
#websockets_log_level 0

# =====
# Security
# =====

# If set, only clients that have a matching prefix on their
# clientid will be allowed to connect to the broker. By default,
# all clients may connect.
# For example, setting "secure-" here would mean a client "secure-
# client" could connect but another with clientid "mqtt" couldn't.
#clientid_prefixes

# Boolean value that determines whether clients that connect
# without providing a username are allowed to connect. If set to
# false then a password file should be created (see the
# password_file option) to control authenticated client access.
#
# Defaults to false, unless there are no listeners defined in the configuration
# file, in which case it is set to true, but connections are only allowed from
# the local machine.
#allow_anonymous false
```

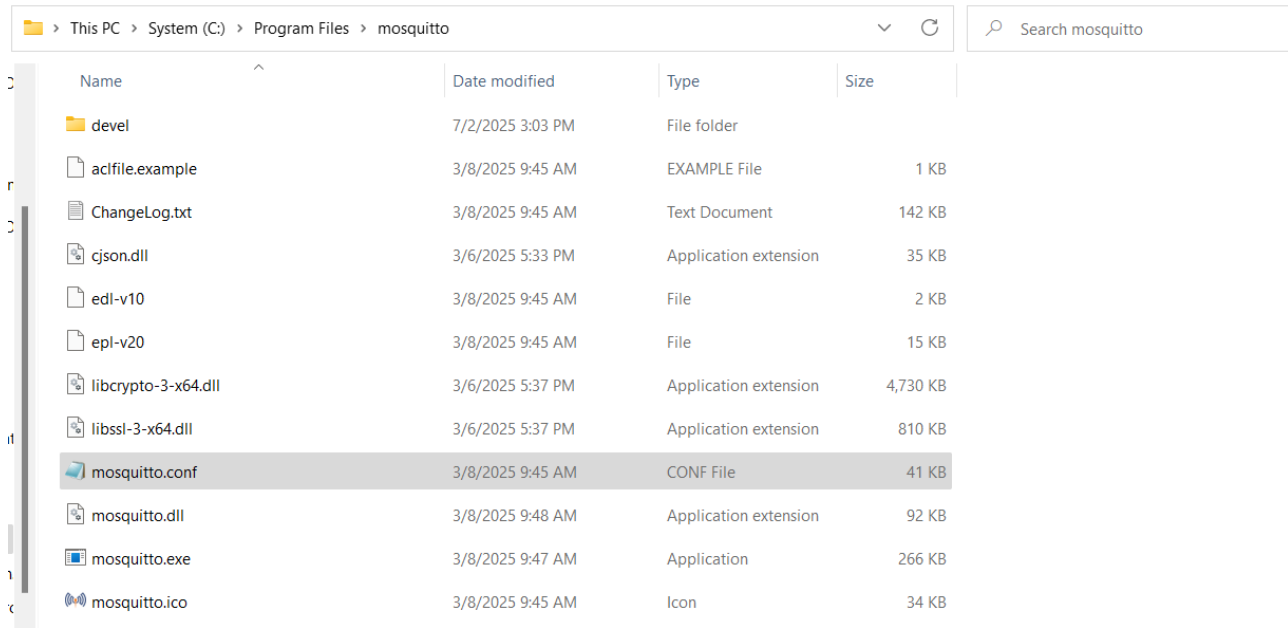
allow_anonymous true

Son olarak, varsayılan ayarların korunması ve yapılan özel konfigürasyonların ayrı bir dosya üzerinden yönetilebilmesi amacıyla, dosya (Mosquitto.conf) bulunduğu klasörden farklı bir konuma (Save As/Farklı Kaydet ile) kaydedilmelidir.

d) Mosquitto Server’ın Çalıştırılması

Mosquitto broker’ın çalıştırılabilmesi için öncelikle Windows arama çubuğundan **“cmd” (Command Prompt)** açılır. Ardından **“cd”** komutu ile Mosquitto’nun kurulu olduğu (orjinal) dizine geçiş yapılır:

```
C:\WINDOWS\System32>cd "C:\Program Files\mosquitto"
C:\Program Files\mosquitto>
```



Name	Date modified	Type	Size
devel	7/2/2025 3:03 PM	File folder	
acfile.example	3/8/2025 9:45 AM	EXAMPLE File	1 KB
ChangeLog.txt	3/8/2025 9:45 AM	Text Document	142 KB
cjson.dll	3/6/2025 5:33 PM	Application extension	35 KB
edl-v10	3/8/2025 9:45 AM	File	2 KB
epl-v20	3/8/2025 9:45 AM	File	15 KB
libcrypto-3-x64.dll	3/6/2025 5:37 PM	Application extension	4,730 KB
libssl-3-x64.dll	3/6/2025 5:37 PM	Application extension	810 KB
mosquitto.conf	3/8/2025 9:45 AM	CONF File	41 KB
mosquitto.dll	3/8/2025 9:48 AM	Application extension	92 KB
mosquitto.exe	3/8/2025 9:47 AM	Application	266 KB
mosquitto.ico	3/8/2025 9:45 AM	Icon	34 KB

Bu adım, broker’ın (Mosquitto) çalıştırılabilir dosyalarının bulunduğu ana dizinine ulaşmak için gereklidir. Daha sonra, hazırlanan özel konfigürasyon dosyası kullanılarak server başlatılır.

Özel konfigürasyona giriş yapmak için, terminalde;

*Mosquitto -c “**konfigüre edilmiş Mosquitto dizini**\Mosquitto.conf” -v*

komutu ile konfigürasyon dosyası (*Mosquitto.conf*) çalıştırılır.

-c komutu (config file)

Mosquitto’ya, hangi konfigürasyon dosyasının kullanılacağını belirtilir. Varsayılan ayarlarla değil, hazırlanan **özel konfigürasyon dosyası** ile broker’ın açılması sağlanmış olur.

-v komutu (verbose mode)

Çalışma sırasında broker’ın yaptığı her şeyin ekranda görülmesini sağlar (bağlantı, publish, subscribe vs.) Bu şekilde Broker’ın gerçekten çalışıp çalışmadığını ve istemcilerin bağlanıp mesaj gönderdiği anlık izlenmiş olur.

```

C:\WINDOWS\System32>cd "C:\Program Files\mosquitto"
C:\Program Files\mosquitto>mosquitto -c "C:\Users\ersahm\OneDrive - OMRON\Desktop\MQTT\mosquitto.conf" -v
1756803103: mosquitto version 2.0.21 starting
1756803103: Config loaded from C:\Users\ersahm\OneDrive - OMRON\Desktop\MQTT\mosquitto.conf.
1756803103: Opening ipv4 listen socket on port 1883.
1756803103: mosquitto version 2.0.21 running
  
```

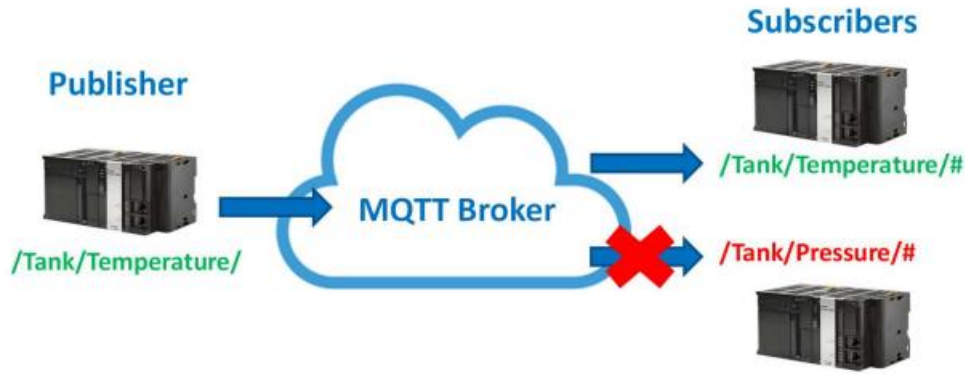
Yapılan bu konfigürasyon ve komut satırı işlemlerinin ardından Mosquitto Server başarılı bir şekilde çalıştırılmış olur. Server, belirtilen port üzerinden istemcilerden gelecek bağlantı taleplerini kabul etmeye hazır hale gelir. Bu aşamadan sonra **publish** ve **subscribe** işlemleri gerçekleştirilerek MQTT haberleşmesi test edilebilir.

2. PLC’nin MQTT Broker’a (Mosquitto Server) Bağlanması

Mosquitto Server çalışır duruma geldikten sonra istemcinin broker’a bağlanması gerekir. Bu uygulamada istemci PLC1 dir.

a) Topic Belirleme

MQTT haberleşmesinde veri alışverişi topic başlıkları üzerinden yapılır. Her istemci sadece ilgili olduğu topic’e bağlanır. Bu topic üzerinden “PUBLISH, SUBSCRIBE” işlemleri yapılabilir.



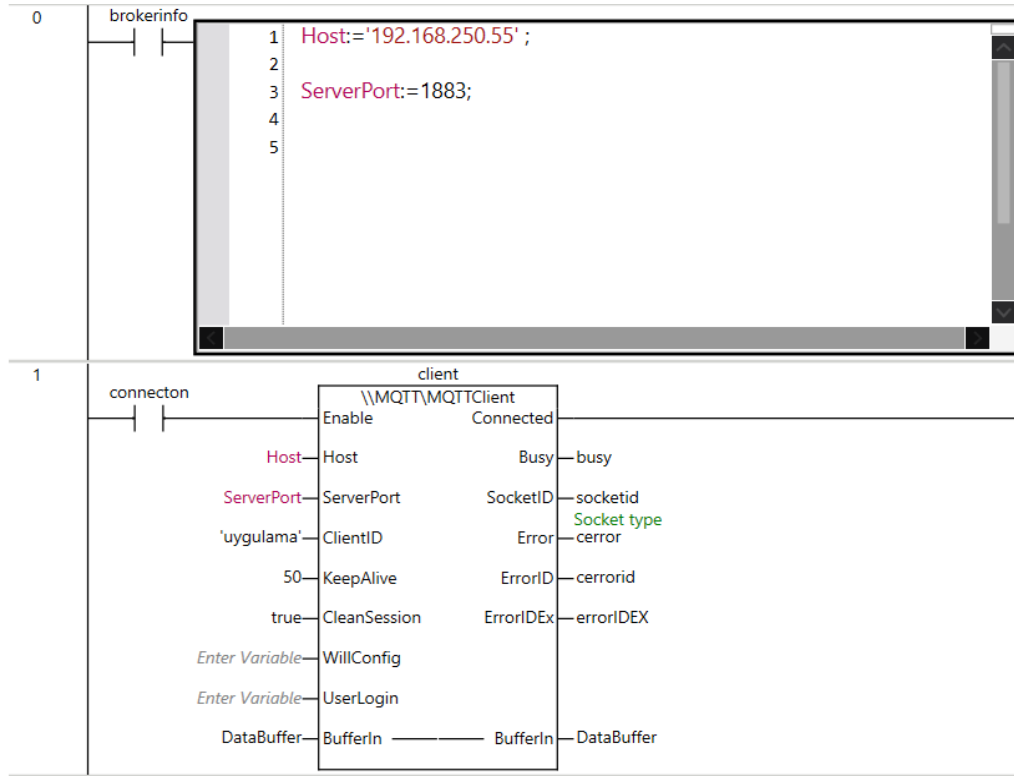
b) MQTTClient ile Mosquitto Server’a Bağlantı

PLC (NJ/NX), **MQTTClient** fonksiyon bloğu kullanılarak broker’a bağlanır. Bağlantı parametreleri (BrokerIP (Host), ServerPort, ClientID, KullanıcıAdı/Şifre) bu fonksiyon bloğu içerisinde ayarlanır. Bağlantı başarılı olduğunda PLC, broker üzerinde **publish** ve **subscribe** işlemlerini gerçekleştirebilir.

Bağlantı Parametreleri:

- **BrokerIP (Host):** Mosquitto’nun kurulu olduğu bilgisayarın IP adresi (ör. 192.168.250.55)
- **ServerPort:** 1883
- **ClientID:** Her PLC için farklı olmalıdır (ör. PLC1_Client, PLC2_Client).
- **KullanıcıAdı/Şifre:** Eğer *allow_anonymous true* parametresi aktif ise, broker’a bağlanmak için kullanıcı adı ve şifre girilmesine gerek yoktur. İstemciler herhangi bir kimlik doğrulama yapmadan broker’a bağlanabilir ancak **konfigürasyon**

dosyasında kullanıcı adı ve şifre tanımlanmışsa ,bu durumda istemcinin broker'a bağlanırken ilgili kullanıcı adı ve şifreyi göndermesi (UserLogin.userid ve UserLogin.userpassword) zorunlu olur. Aksi halde bağlantı reddedilir.



Broker bağlantısı sağlandıktan sonra, (MQTTClient) **client.Connected** çıkışı aktif hale gelir ve PLC'nin MQTT üzerinden iletişime geçtiği doğrulanır.

c) MQTTPublish ile Mosquitto Server'a Veri Paylaşımı

MQTTPublish fonksiyon bloğu kullanılarak veriler belirlenen topic'e gönderilmektedir.

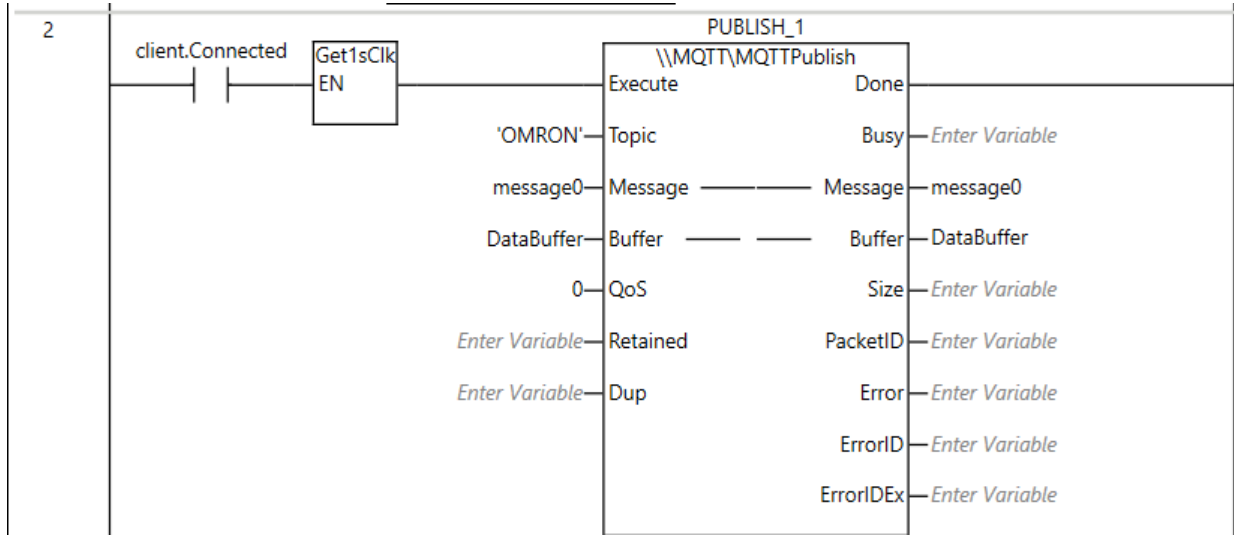
Bağlantı Parametreleri:

Topic: Mesajın gönderileceği konu başlığı ('OMRON')

Message: Gönderilecek veri (ör. *message0* (String veri tipinde))

Buffer: Veri paketinin saklandığı buffer alanı (*DataBuffer*)

QoS: Hizmet kalitesi seviyesi (0 = en hızlı, onaysız gönderim)



PLC broker'a bağlandıktan sonra her saniye bir tetikleme sinyali (Get1sClk) ile *MQTTPublish* bloğu çalıştırılır. Bu sayede 'OMRON' Topic'i üzerinden düzenli aralıklarla message0 broker'a gönderilir. Broker bu mesajları ilgili topic'e abone olmuş tüm istemcilere yönlendirir.

d) MQTTSubscribe ile Mosquitto Server'dan Veri Alımı

Mosquitto Server'a başarıyla bağlanan PLC, belirlenen topic üzerinden veri alabilmek için *MQTTSubscribe* fonksiyon bloğunu kullanır.

Bağlantı Parametreleri:

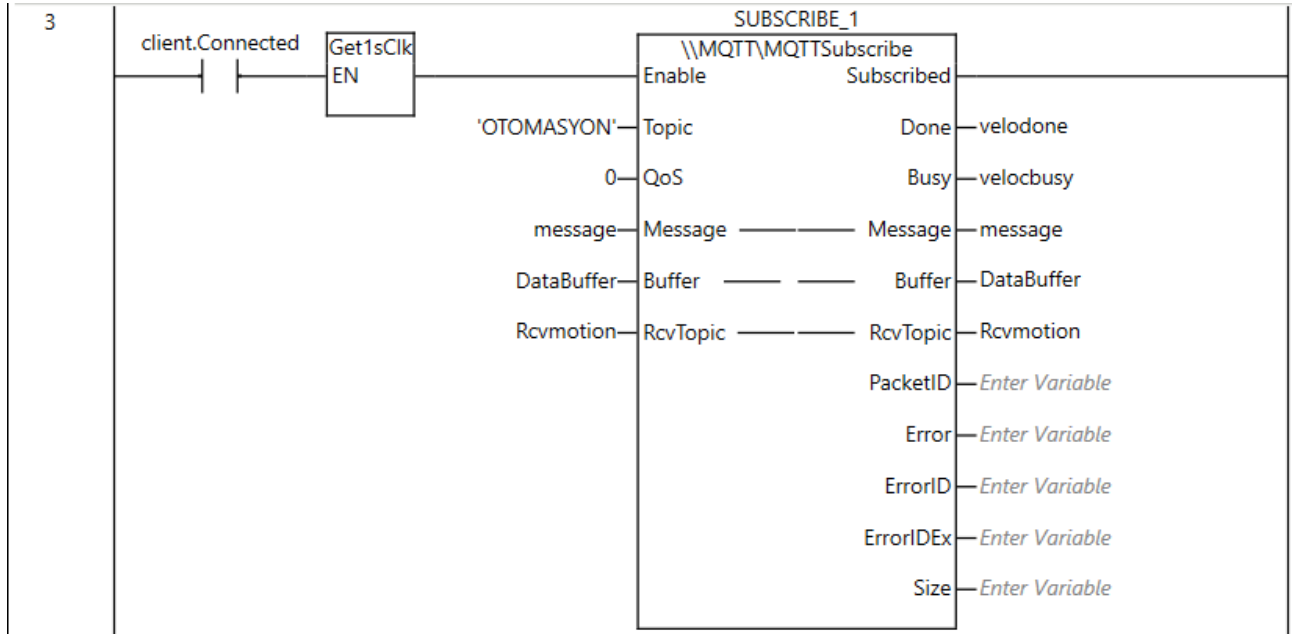
Topic: Abone olunacak başlık (ör. 'OTOMASYON')

QoS: Hizmet kalitesi seviyesi (0 = hızlı, onaysız alım)

Message: Broker'dan gelen mesajın PLC içerisinde tutulduğu değişken

Buffer: Mesajın geçici olarak saklandığı alan

RcvTopic: Mesajın hangi topic üzerinden geldiğini gösteren çıkış



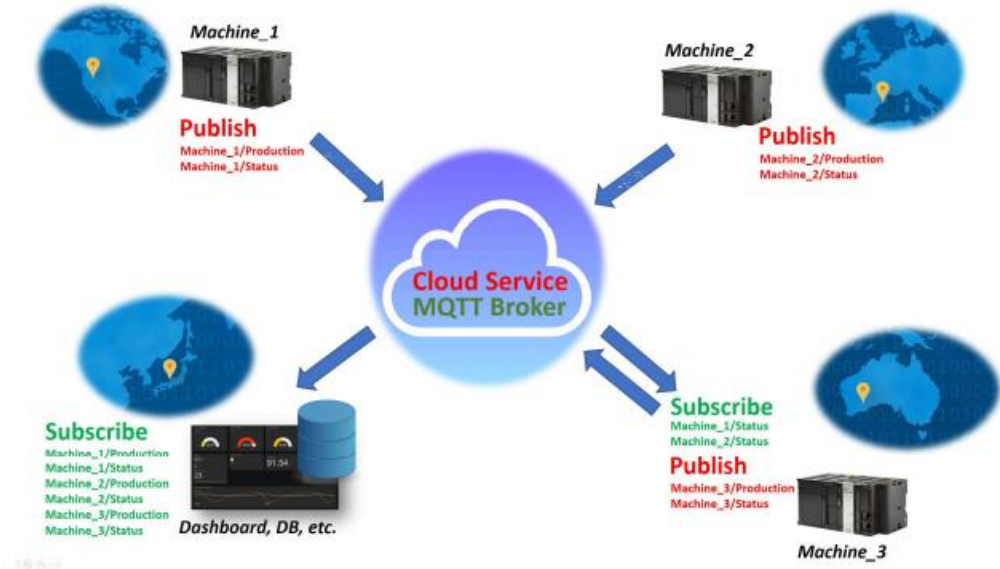
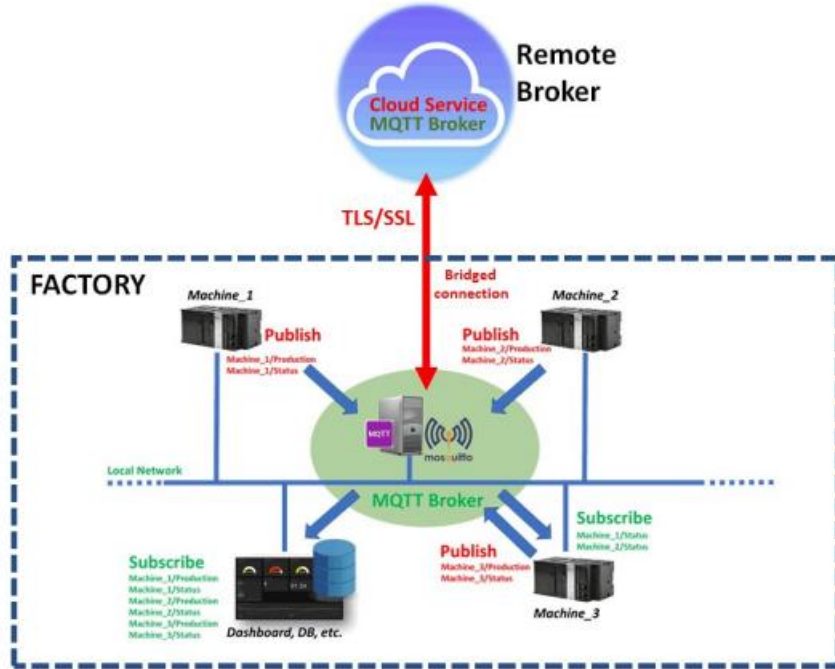
PLC broker'a bağlandıktan sonra her saniye bir tetikleme sinyali (*Get1sClk*) ile **MQTTSubscribe** bloğu çalıştırılır. Bu sayede PLC, '**OTOMASYON**' topic'i üzerinden düzenli aralıklarla broker'dan gelen verileri dinler. Broker'a bu topic altında publish edilen her mesaj, **MQTTSubscribe** bloğu aracılığıyla alınarak PLC içerisindeki **message** değişkenine aktarılır ve kontrol sürecinde kullanılabilir.

3. Remote Broker ile Local Broker Arasında Köprü (Bridge) Bağlantısı

Yerel ağda çalışan MQTT sistemlerinde mesajlar yalnızca aynı ağ içerisindeki istemci ve broker arasında iletilir. Ancak, sistemin yalnızca lokal ile sınırlı kalmaması ve tüm verilerin merkezi bir bulut altyapısına da aktarılması gerektiğinde **bridge (köprü) bağlantısı** kurulmaktadır. Bridge bağlantısı sayesinde, yerel broker'a gelen tüm mesajlar belirlenen kurallar doğrultusunda otomatik olarak bulut broker'a yönlendirilir. Böylece veriler hem lokal istemciler hem de uzak (remote) istemciler tarafından erişilebilir hale gelir.

Bu senaryoda, **bulut broker** ve yerel broker sürekli senkronize çalışır. Güvenli veri aktarımı için broker'lar arasındaki iletişim, bulut tarafının sağladığı **TLS (Transport Layer Security)** mekanizması ile korunmak zorundadır. Bu yapı, hem verilerin şifrelenmiş bir şekilde iletilmesini sağlar hem de kimlik doğrulama ile güvenliği artırır. Sonuç olarak, bridge bağlantısı lokal verilerin yalnızca yerel istemcilerle sınırlı

kalmadan, güvenli bir şekilde bulut ortamına taşınmasına ve farklı coğrafi bölgelerdeki istemcilerin de aynı verilere erişmesine imkân tanır.



Bu uygulamada bulut broker olarak **HiveMQ** tercih edilmiştir. Bridge bağlantısına ait gerekli tüm ayarlar ise daha önce oluşturulmuş olan özel **mosquitto.conf** yapılandırma dosyası içerisinde tanımlanacaktır.



a) HiveMQ Bulut Broker Kurulumu

Öncelikle, <https://www.hivemq.com/mqtt/public-mqtt-broker/> platformu üzerinden bir kullanıcı hesabı oluşturulur. Hesap oluşturma işlemi tamamlandıktan sonra, ücretsiz olarak sunulan **Free Cluster** seçeneklerinden biri tercih edilerek bulut broker yapılandırması başlatılır.

Ardından sistem tarafından atanan Cluster URL ve ilgili bağlantı portları (MQTT için 8883, WebSocket için 8884) kullanıcıya sunulur. Bu bilgiler, istemci veya yerel broker ile bulut broker arasındaki iletişimin sağlanabilmesi için gerekli temel bağlantı parametrelerini oluşturmaktadır.

Kimlik doğrulama aşamasında, kullanıcılar “*Create Connection Credentials*” alanı üzerinden bir kullanıcı adı ve şifre belirleyerek kalıcı kimlik bilgilerini oluşturur. Bu kimlik bilgileri, istemcilerin güvenli bir şekilde HiveMQ bulut broker’a bağlanmasını sağlamak ve ilerleyen aşamalarda köprü (bridge) bağlantısı yapılırken **mosquitto.conf** dosyası içerisinde kullanılmaktadır.

The screenshot shows the HiveMQ Cloud console interface. On the left, there's a sidebar with 'ORGANIZATIONS' (showing 'asdas') and 'CLUSTERS OVERVIEW' (listing 'Free #1' and 'Free #2'). The main area is titled 'Free #1' and has tabs for 'Overview', 'Access Management', 'Integrations', 'Web Client', and 'Getting Started'. The 'Getting Started' tab is active, showing 'Your Connection Settings' and 'Create connection credentials'.

Your Connection Settings

Your cluster is already up and running! Use these connection setting to connect to your cluster.

Cluster URL:

Port:

Websocket Port:

Create connection credentials

Define the credentials that your MQTT clients can use to connect to your HiveMQ Cloud cluster. Please visit the [HiveMQ documentation](#) for examples on how to use the credentials to connect an MQTT client.

Username:

Password:

Confirm Password:

b) Bulut Broker için Konfigürasyon Dosyası Hazırlama

HiveMQ üzerinde oluşturulan kullanıcı adı, şifre ve bağlantı parametrelerinin yerel broker tarafından kullanılabilmesi için **mosquitto.conf** dosyasında gerekli düzenlemeler yapılmalıdır. Bu adım, yerel broker ile bulut broker arasında **bridge (köprü) bağlantısının** kurulmasını sağlar. Konfigürasyon dosyasında öncelikle bağlantı ismi, adres bilgisi, kullanıcı adı ve şifre tanımlanır. Ardından, güvenli iletişim için gerekli olan TLS sertifikası indirilir ve konfigürasyona eklenir.

Sertifika, HiveMQ'nun resmi topluluk sayfası üzerinden elde edilebilir.

<https://community.hivemq.com/t/tls-certificates/1068/11>

Konfigürasyonda bu sertifikanın dosya yolu **bridge_cafile** parametresi ile gösterilir. Böylece yerel broker, bulut broker'ın kimliğini doğrulayarak güvenli iletişim sağlar. Son aşamada ise hangi **topic**'lerin yönlendirileceği belirtilir. Eğer tüm topic'lerin senkronize edilmesi isteniyorsa konfigürasyona şu ifade eklenir:

topic # both 0

Konfigürasyonu yapmak için daha önce yerel broker için hazırlanmış olan **mosquitto.conf** dosyasına girilir ve gerekli parametreler dosyanın en sonuna eklenir.

Örnek uygulama için;

```

connection hivemq
address c7a38321f1394d85a3c43096ed45c460.s1.eu.hivemq.cloud:8883
remote_username ahmet
remote_password Ahmetmete61
bridge_protocol_version mqttv311
try_private false
bridge_cafile C:\Users\ersahm\OneDrive
OMRON\Desktop\MQTT\isrgrootx1.pembridge_insecure false
topic # both 0

```

Ardından, yapılan değişikliklerin etkin olabilmesi için **komut satırı (CMD)** üzerinden Mosquitto server yeniden çalıştırılır. Bu işlem sonucunda, yapılan konfigürasyon sayesinde yerel broker ile bulut broker arasında bridge bağlantısı kurulmuş olur ve senkronizasyon sağlanır.

```

Command Prompt - mosquitto -c "C:\Users\ersahm\OneDrive - OMRON\Desktop\MQTT\mosquitto.conf" -v
Microsoft Windows [Version 10.0.22000.3260]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\System32>cd "C:\Program Files\mosquitto"

C:\Program Files\mosquitto>mosquitto -c "C:\Users\ersahm\OneDrive - OMRON\Desktop\MQTT\mosquitto.conf" -v
1756887187: mosquitto version 2.0.21 starting
1756887187: Config loaded from C:\Users\ersahm\OneDrive - OMRON\Desktop\MQTT\mosquitto.conf.
1756887187: Opening ipv4 listen socket on port 1883.
1756887187: Bridge local.IABTR00233.hivemq doing local SUBSCRIBE on topic #
1756887187: Connecting bridge hivemq (c7a38321f1394d85a3c43096ed45c460.s1.eu.hivemq.cloud:8883)
1756887187: Bridge IABTR00233.hivemq sending CONNECT
1756887187: mosquitto version 2.0.21 running
1756887188: Received CONNACK on connection local.IABTR00233.hivemq.
1756887188: Bridge local.IABTR00233.hivemq sending SUBSCRIBE (Mid: 2, Topic: #, QoS: 0, Options: 0x00)
1756887188: Received PUBACK from local.IABTR00233.hivemq (Mid: 1, RC:0)
1756887188: Received SUBACK from local.IABTR00233.hivemq
1756887188: Received PUBLISH from local.IABTR00233.hivemq (d0, q0, r1, m0, '$SYS/broker/connection/IABTR00233.hivemq/state', ... (1 bytes))
1756887188: Received PUBLISH from local.IABTR00233.hivemq (d0, q0, r1, m0, '$SYS/broker/connection/DESKTOP-M661913.hivemq/state', ... (1 bytes))

```

c) Bulut Broker’den Publish veya Subscribe Olma

<https://www.hivemq.com/demos/websocket-client/> bağlantısına giriş yapılır.

The screenshot shows the HiveMQ Websockets Client Showcase interface. At the top, there is a header with the HiveMQ logo and the text 'Websockets Client Showcase'. Below this, there is a banner for 'Need a fully managed MQTT broker?' with a 'Get your free account' button. The main section is titled 'Connection' and shows a 'connected' status with a green dot. The connection details are as follows:

Field	Value
Host	c7a38321f1394d85a3c43096ed45c460.s1.eu.hivemq.c
Port	8884
ClientID	uygulama
Username	ahmet
Password	*****
Keep Alive	60
SSL	☒
Clean Session	☒
Last-Will Topic	
Last-Will QoS	0
Last-Will Retain	☐
Last-Will Message	

Buradaki bilgiler, oluşturulan bulut server konfigürasyonuna göre girilir. Sağ üstte **Connected** ifadesinin görülmesi, broker’a başarılı bir şekilde bağlantı sağlandığını gösterir. Burada dikkat edilmesi gereken nokta, kullanılan port numarasıdır. Görselde de görüldüğü üzere, **8883** portu standart TLS bağlantısı için, **8884** portu ise WebSocket üzerinden bağlantılar için kullanılmaktadır. Bu nedenle, yapılan konfigürasyonda port numarasının doğru seçilmesi kritik önem taşır. Bağlantı kurulduktan sonra, belirlenen topic’lere **subscribe** olunabilir veya istenilen veriler ilgili topic’lere **publish** edilerek gönderilebilir.

Publish

Topic

OTOMASYON

QoS

0

Retain

☐

Publish

Message

SELAMLAR

Subscriptions

Add New Topic Subscription

Qos: 0

OMRON

X

Messages

2025-09-03 14:00:10	Topic: OMRON	Qos: 0	OMRON PLC ESENLİKLER
2025-09-03 14:00:09	Topic: OMRON	Qos: 0	OMRON PLC ESENLİKLER
2025-09-03 14:00:08	Topic: OMRON	Qos: 0	OMRON PLC ESENLİKLER

Bu uygulamada PLC, MQTT üzerinden bulut broker ile çift yönlü haberleşme sağlamaktadır. Programda kullanılan **MQTTPublish** bloğu ile PLC’den belirlenen ‘OMRON’ TOPİC’ine mesaj gönderilirken, **MQTTSubscribe** bloğu ile ‘OTOMASYON’ bloğu ile veri paylaşımı gerçekleşir. Bağlantı durumu sürekli kontrol edilmekte ve her saniyede bir tetikleme sinyali sayesinde publish ve subscribe işlemleri düzenli aralıklarla gerçekleştirilmiştir. Böylece PLC ile bulut broker arasında güvenilir ve senkronize bir veri alışverişi sağlanmıştır.

Section0 - Program0 X

Variables

client.Connected

Get1sCik

EN

PUBLISH_1

\\MQTT\MQTTPublish

Execute

Done

Busy

Enter Variable

'OMRON' Topic

(OMR.ESEN)message0 Message

Message

message0 (OMR.ESEN)

DataBuffer Buffer

Buffer

DataBuffer

QoS

Size

Enter Variable

Retained

PacketID

Enter Variable

Dup

Error

Enter Variable

ErrorID

Enter Variable

ErrorIDex

Enter Variable

SUBSCRIBE_1

\\MQTT\MQTTSubscribe

Enable

Subscribed

Done

velodone (False)

Busy

velocbusy (True)

(SEL.ELAML)message Message

Message

message (SEL.ELAML)

DataBuffer Buffer

Buffer

DataBuffer

(OTOMAS..)Rcmotion RcvTopic

RcvTopic

Rcmotion (OTOMAS..)

PacketID

Enter Variable

Watch (Project)1

Device name	Name	Online value	Modify	Comment	Data type	AT
new_Controller_0	Program0.message	SELAMLAR			STRING[1986]	
new_Controller_0	Program0.message0	OMRON PLC ESENLIKLER			STRING[1986]	
new_Controller_0	Input Name...					

Output Build Watch (Project)1