

Operating PLC200 as a MQTT client with Eclipse Mosquitto[™] and IoT WEGnology[™] Brokers

Application Note



MQTT Application Note

Document: 10012928101

Revision: 00

Publication Date: 03/2025

The information below describes the reviews made in this manual.

Version	Revision	Description
V1.04.XX	R00	First Edition.

	IMPORTANT NOTICE	0-5 0-5
1	DESCRIPTION 1.1 REFERENCE DOCUMENTS 1.2 ARCHITECTURE 1.3 PASSIVE NETWORK COMPONENTS	1-1 1-1 1-1 1-1
2	MQTT CLIENT SPECIFICATIONS	2-1 2-1
3	WPS COMMUNICATION WIZARD	3-1
4	 NON-SECURE CONNECTION WITH LOCAL MOSQUITTO BROKER 4.1 IP ADDRESS AND NETWORK CONFIGURATION 4.1.1 PLC200 IP ADDRESS CONFIGURATION 4.1.2 PC IP ADDRESS CONFIGURATION 4.2 BROKER PREREQUISITES 4.3 PLC CONFIGURATION WITH WPS 4.4 SENDING AND RECEIVING MESSAGES 4.4.1 PUBLISH 4.4.2 SUBSCRIBE 	4-1 4-1 4-1 4-1 4-2 4-4 4-4 4-5
5	SECURE CONNECTION WITH WEGNOLOGY BROKER	5-1 5-1 5-1 5-1 5-1 5-3 5-3 5-4 5-5

IMPORTANT NOTICE ABOUT CYBERSECURITY AND COMMUNICATIONS

This product/equipment can connect and exchange information through networks and communication protocols. It has been designed and subjected to tests to ensure correct operation with other automation systems using the protocols mentioned in this manual. Therefore, it is essential that the customer understands the responsibilities in connection with information and cybersecurity when using this equipment.

Consequently, it is the exclusive obligation of the customer to adopt in-depth defense strategies and implement policies and measures to ensure the security of the system as a whole, including with regard to communications sent and received by the equipment. Among such measures, we can point out the installation of firewalls, antivirus and malware protection applications, data encryption, authentication control and physical user access.

WEG and its affiliates take no liability for damages or losses arising from cybersecurity breaches, including, but not limited to, unauthorized access, intrusion, information, or data leak and/or theft, denial-of-service attacks, or any other form of security breach. Using this product under conditions for which it was not specifically designed is not recommended and may result in damage to the product, the network, and the automation system. Thus, it is essential that the customer understand that the external intervention by third-party software applications, such as sniffers or applications with similar actions, has the potential to cause interruptions or restrictions in the functionality of the equipment.

TRADEMARKS

All other trademarks are the property of their respective holders.

1 DESCRIPTION

This application note is intended to provide a description of how to program a PLC200 programmable logic controller to communicate in two MQTT network scenarios. As a client connecting to an Eclipse Mosquitto broker with a non-encrypted connection, and as a client connecting to the WEGnology broker with an encrypted (TLS) connection.

This document is meant for trained personnel working with the described equipment and MQTT network installation, besides a good knowledge of automation and programmable logic controllers.

1.1 REFERENCE DOCUMENTS

This application note was developed based on the following documents and tools:

Document / Tool	Version	Source
MQTT Specification	3.1.1	OASIS MQTT Technical Committee
RFC5246: The Transport Layer Security (TLS) Protocol	1.2	IETF
Mosquitto Broker	2.0.20	Eclipse Foundation
WEGnology Broker	01/2025	WEG
WPS	3.10	WEG
MQTTX Client	1.11.1	EMQ Technologies Inc.

1.2 ARCHITECTURE



Figure 1.1: Network architecture

1.3 PASSIVE NETWORK COMPONENTS

For passive network components - cables, ethernet switch - we recommend using certified components for industrial applications. Please refer to the product documentation for information about the proper network installation.

2 MQTT CLIENT SPECIFICATIONS

The PLC200 programmable logic controller MQTT client is meant to be used for reading and writing data to the 20 user parameters (P0800..P0838). This data can then be applied to the user program to monitor or change variables. Some of the client characteristics are:

- The client is configured with a communication wizard available in the WEG Programming Suite (WPS) software.
- The client is able to publish messages at a user-defined interval between 5 and 900 seconds. The message payload has a fixed JSON format, where each object is related to one of the user parameters.
- If the client is configured and enabled, but there isn't a connection available, the PLC200 will store up to 50 messages in its volatile memory that will be published upon reconnection.
- The client is able to subscribe to a user-defined topic. Messages published to the topic, that follow the specified JSON format, can be used to update user parameter values.
- The client is able to establish encrypted connections using TLS v1.2.

2.1 RELEVANT PARAMETERS

The following parameters can be used to verify the status and manage the MQTT client.

P0841: MQTT -	P0841: MQTT - Status					
Adjustable Range:	0 4		Factory Setting:	0		
Properties:	ro, enum					

Description:

It indicates the status of the MQTT client, regarding settings and the sending of data to the server.

Indication	Description		
0 = Inactive	It indicates that the MQTT client function is not set; it is disabled.		
1 = No Connection	It indicates that the MQTT client has been set and is enabled, but there is currently no active link available.		
2 = Connected (Pub)	It indicates that the MQTT client has been set up, is enabled, and has an active connection to the configured Broker for publishing data.		
3 = Connected (Pub/Sub)	It indicates that the MQTT client function has been set up, is enabled, and has an active connection to the configured Broker for publishing and receiving data.		
4 = Connection fail	Indicates a failure with the MQTT client connection.		

P0842: Last Public. MQTT

Adjustable Range:	0 4294967295	Factory Setting:	1704070861
Properties:	ro, date and time epoch		

Description:

It indicates the timestamp of the last successful publication of user parameters.

P0844: MQTT -	Enable/Disable	
Adjustable Range:	0 2	Factory 1 Setting:
Properties:	rw. enum	

Description:

Allow user to enable, disable or enable only the MQTT publish feature (no subscription).

Indication	Description
0 = Disable	Disable the MQTT client.
1 = Enable	Enable the MQTT client for publishing and subscribing.
2 = Enable only publish	Enable the MQTT client for publishing only.



NOTE!

The parameter P0844 (MQTT - Enable/Disable) has priority over the "*enable_sub*" field of the configuration file. This means that if the parameter indicates that subscriptions are disabled, the PLC200 will not subscribe to any topic, regardless of the value of the field in the configuration file. The subscription will only occur if both the parameter and configuration field allow it.

3 WPS COMMUNICATION WIZARD

The PLC200 MQTT client is configured through a communication wizard available at WPS.

mod New File ⊕ _ main Mod Open Cut Copy Paste Ctrl+V Delete Delete Rename	PLC200_MQTT_app PLC200 (PLC200 PLC200 (PLC200 Program Program Program Communica Communica PROGRAMOTT	lication_note v1.3.X) tions		Wel
Delete Set main file	Hod	New File Cut Copy Paste Ctrl+V Delete	↔ Modbus TCP	Open Copy Ctrl+C Delete Delete Rename Set main file

Figure 3.1: Creating MQTT configuration wizard.

The client has a set of basic and advanced configurations:

Basic Configuration Advar	nced			
Broker URL:				
Client ID:				
Username:				
Password:				•
Publication Interval(s):	30			
Publish Topic:				
Security:	O MQTT		(MQTT with TLSv1.2)
Certificate:	Im	port	Remove	
Invalid broker URL.				
		ОК	Cancel	Help

Figure 3.2: MQTT basic configuration.

- Broker URL: The MQTT Broker URL or IP address in which the client should connect. (Max. 128 characters)
- Client ID: Unique identifier for the client registered at the broker. (Max. 25 characters)
- Username: Client username registered at the broker. (Optional, Max. 63 characters)
- Password: Client password registered at the broker. (Optional, Max. 64 characters)
- Publication Interval: The time in seconds between publications of user parameters. (Between 5 and 900 seconds)
- **Publish Topic**: Topic to publish the cyclic payload, if not defined the default value is: "plc200/<*Client ID*>/state". (Max. 63 characters)

- Security: Whether the connection will encrypted (MQTTS) using port 8883 or non-encrypted (MQTT) using port 1883.
- **Certificate**: If the user chose a encrypted connection it must provide a CA certificate file in the PEM format. It will be used during the handshake to verify the broker's certificate chain authenticity.

wes MQTT -> App_note		;
Basic Configuration Adva	nced	
Allow Subscription:	⊖ True	O False
Subscribe Topic:		
Response Topic:		
Broker Port:	1883	
DNS Server:	8.8.8.8	
Keep Alive(s):	30	
Quality of Service (QoS):	0	~
Clean Session:		False
Invalid broker URL.		
		OK Cancel Help

Figure 3.3: MQTT advanced configuration.

- Allow Subscriptions: Whether the client should subscribe to a topic or not.
- **Subscribe Topic**: if "Allow Subscriptions" is true, the user must provide a subscription topic, not used otherwise. If not defined the default values is the same as the "Publish Topic". (Max. 63 characters)
- Response Topic: Topic to publish a reply after a message is recieved with the request status. Default value is set to the same as "Publication Interval". (Optional, Max. 63 characters)
- Broker Port: port in the broker that the client will connect to.
- DNS server: DNS server for name resolving. (Optional)
- Keep Alive: Interval between Keep Alive messages.
- Quality of Service:
 - 0 At most once;
 - 1 At least once;
 - 2 Exactly once;
- Clean Session: Set to false for a persistent session. In a persistent connection the broker will send previously queued messages upon a reconection.

4 NON-SECURE CONNECTION WITH LOCAL MOSQUITTO BROKER

This setup will use a Eclipse Mosquitto broker running in a workstation in a local network with PLC200 programmable logic controller.

4.1 IP ADDRESS AND NETWORK CONFIGURATION

To allow communication among the devices, they need to have a compatible IP address configuration. It means the IP address must be in the same range, according to network mask. For this example, we will use the following IP addresses:

- **Subnet mask**: 255.255.255.0
- IP addresses: each device must have a different IP address.

PLC200: 192.168.0.10

Workstation: 192.168.0.20

4.1.1 PLC200 IP ADDRESS CONFIGURATION

For this application, the following configurations have been done via USB to allow Ethernet communication to WPS:

- P0850 IP Address Config: 0 (Parameters).
- **P0852 IP Address**: 192.168.0.10.
- **P0855 CIDR**: 24 (255.255.255.0).
- **P0856 Gateway**: 0.0.0.0.

4.1.2 PC IP ADDRESS CONFIGURATION

To configure this option on Windows platform, go to "Network Connections" and select "Properties" of the applicable Ethernet interface:

Ethernet0 Properties	X Internet Protocol Version 4 (TCP/IPv4) Properties X
Networking	[General]
Connect using: The (R) 82574L Gigabit Network Connection Configure	You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
This connection uses the following items:	
Client for Microsoft Networks File and Printer Sharing for Microsoft Networks File and Printer Sharing for Microsoft Networks File and Protocol Version 4 (TCP/IPv4) Microsoft Network Adapter Multiplexor Protocol ProFINET IO protocol (DCP/LLDP) Amicrosoft LLDP Protocol Driver Igstall Install Properties	IP address: 192.168.0.20 Sybnet mask: 255.255.0 Default gateway: . Obtain DNS server address automatically Image: Image
Description Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.	Alternate DNS server:
OK Cance	OK Cancel

Figure 4.1: PC IP Address Configuration

4.2 BROKER PREREQUISITES

For the configuration of the Mosquitto broker please refer to the Mosquitto official documentation. The changes made from the default configuration file for the Mosquitto broker are:

- Setting a *listener* on port 1883.
- Setting the *allow_anonymous* flag to false as we want to connect from a external device.
- Creating a username and password for the client with mosquitto_passwd and storing it in a file with the option password_file.

4.3 PLC CONFIGURATION WITH WPS

After creating a new configuration in WPS 3.10, add a new MQTT file in the "Communications" > "MQTT" folder. In the "Basic Configuration" tag fill out:

- Broker URL: 192.168.0.20
- Client ID: <Any unique identificator>
- Username: <Username as created in the previous step>
- Password: <Password as created in the previous step>
- Publication Interval: 60
- Publish Topic: plc200/<Client ID>/state
- Security: MQTT.
- Certificate: <Not used>

For the advanced tab:

- Allow Subscriptions: True
- Subscribe Topic: plc200/<Client ID>/command
- Response Topic: plc200/<Client ID>/state
- Broker Port: 1883
- DNS server: 8.8.8.8 (not used)
- Keep Alive: 30
- Quality of Service: 0
- Clean Session: True

Wes MQTT -> App_note		×	wes MQTT -> App_note		×
MQTT -> App_note Basic Configuration Adva Broker URL: Client ID: Username: Password: Publication Interval(s): Publish Topic: Security: Certificate:	anced 192.168.0.20 clidappnote123 example_user 60 plc200/clidappnote/state MQTT MQTT MQTT S(MQTT with TLSv1. Import Remove	×	We MQTT -> App_note Basic Configuration Adva Allow Subscription: Subscribe Topic: Response Topic: Broker Port: DNS Server: Keep Alive(s): Quality of Service (QoS): Clean Session:	anced	×
	OK Cancel	Help		OK Cancel I	Help

Figure 4.2: Client configuration for Mosquitto broker in WPS.

After configuring the client, build the resource and download it to the PLC200. After this step, the client should attempt a connection.

Configuração PLC200_MQTT_application_note	
Recurso PLC200	
Dispositivo PLC200	Versão 1.4.X
Opções	
 Para/Inicia o programa automaticamente Inicializa variáveis 	 Download do código fonte Escrita do setup (parâmetros) Recursos de comunicações
Arquivos de memória interna filesystem.bin - 11 mar. 2025 12:54:03 info.bin - 11 mar. 2025 12:54:03 mqtt.bin - 11 mar. 2025 12:54:03 retain.bin - 11 mar. 2025 12:54:03 volatile.bin - 11 mar. 2025 12:54:03 ladder.bin - 11 mar. 2025 12:54:03 ladder.bin - 11 mar. 2025 12:54:03	Download de valores de monitoração Selecionar
	OK Cancelar

Figure 4.3: Download resource files to PLC200.

🔤 Selecionar O	:\Windows\System32\cmd.exe - mosquitto -v -c mosquitto.conf	—		×
C:\Program F 1737368713: 1737368713: 1737368713: 1737368713: 1737368713: 1737368713: 1737368713:	Tiles\mosquitto>mosquitto -v -c mosquitto.conf mosquitto version 2.0.20 starting Config loaded from mosquitto.conf. Opening ipv6 listen socket on port 1883. Opening ipv4 listen socket on port 1883. mosquitto version 2.0.20 running New connection from 192.168.0.10:52435 on port 1883. New client connected from 192.168.0.10:52435 as clidappnote123 (p2, c1, k15, u'example_user').			
1737368713: 1737368713: 1737368713: 1737368713: 1737368713: 1737368733: 1737368733: 1737368733: 1737368753: 1737368753: 1737368753:	No Will message specified. Sending CONNACK to clidappnote123 (0, 0) Received SUBSCRIBE from clidappnote123 plc200/clidappnote123/command (QoS 0) clidappnote123 0 plc200/clidappnote123/ Received PINGREQ from clidappnote123 Sending PINGRESP to clidappnote123 Received PINGREQ from clidappnote123 Sending PINGRESP to clidappnote123 Received PUBLISH from clidappnote123 (d0, q0, r0, m0, 'plc200/clidappnote123/state', (241)	bytes))	

Figure 4.4: Mosquitto broker command line showing PLC200 connection.

4.4 SENDING AND RECEIVING MESSAGES

4.4.1 PUBLISH

After configured and connected, the PLC200 client should start publishing data at the interval defined in the configuration step. The payload published contains one object with the value for each of the twenty user parameters available (P0800 to P0838) and also an extra object with the sampling time in epoch format. The following JSON is an example of a payload published by the PLC200.

```
{
        "data": {
                "user1": 5,
                "user2": 0,
                "user3": 10,
                "user4": 0,
                "user5": 0,
                "user6": 999,
                "user7": -10,
                "user8": 0,
                 "user9": 0,
                 "user10": 2147483647,
                 "user11": 0,
                 "user12": 0,
                "user13": 0,
                 "user14": 0,
                "user15": 71254,
                "user16": 0,
                "user17": 0,
                "user18": 0,
                "user19": 0,
                "user20": 0
        },
        "time": 1737029549
```

```
}
```

NOTE!

User parameters are sampled at the publishing interval, but if for some reason the client is disconnected at the time, The message will be store in the PLC200 volatile memory and published upon reconnection. The client can store up to 50 samples.

To interact with the PLC200 MQTT client is recommended to use another client to read the published data. In this example the MQTTX client for Windows is used but any client with support to version 3.1.1 is acceptable. The client must be properly configured in the broker and subscribe to the PLC200 publishing topic.

S MQTTX \times _ File Edit View Window Help PLC200_APP_NOTE@1 🗸 😽 🚺 () 「□ 2 … × Plaintext ~ + New Subscription All Received Published plc200/clidappno... QoS 0 Topic: plc200/clidappnote123/state QoS: 0 ዋ {"data": {"user1":0, "user2":0, "user3":0, "user4":0, "user5":0, "user6":0, "use r7":0, "user8":0, "user9":0, "user10":0, "user11":0, "user12":0, "user1 3":0,"user14":0,"user15":0,"user16":0,"user17":0,"user18":0,"user 19":0,"user20":0},"time": 1737370039 } 2025-01-20 07:48:15:820 ሐ 3

NON-SECURE CONNECTION WITH LOCAL MOSQUITTO BROKER

Figure 4.5: Visualization of message published by PLC200 in MQTTX.

4.4.2 SUBSCRIBE

After connecting to the broker, the PLC200 client subscribes to the topic configured and starts receiving messages from other clients, like the MQTTX client for example. The PLC200 allows for the other clients to publish messages that update the user parameters (P0800 to P0838) values. The PLC200 client expects a payload in the following format:

```
{
        "payload": {
                 "id": "addf120b430021c36c232c99ef8d926aea2acd6b",
                 "args": {
                          "1": 0,
                         "2": 0,
                          "3": 0,
                          "4": 0,
                          "5": 0,
                          "6": 0,
                          "7": 0,
                          "8": 0,
                          "9": 0,
                          "10": 0,
                          "11": 0,
                          "12": 0,
                          "13": 0,
                          "14": 0,
                          "15": 0,
                          "16": 0,
                          "17": 0,
                          "18": 0,
                          "19": 0,
                          "20": 0
```

}

Where each key inside the "payload.args" object refers to user parameter number (1 to 20) and the corresponding value to be written. Each message doesn't need to contain all user parameters, only the ones that should be updated. There is also an optional "payload.id" object that can store a string of up to 40 characters, this field value will be send with the response payload and is meant to be used as a message identification mechanism, if the object is not present in the payload the response payload will have an ID object with value 0.

Upon receiving a publication, the client will update the user parameters and reply in the configured response topic with a message containing the request status and ID. The status field correspond to the HTTP response status codes and indicate whether the requested update was successful (200), an internal error occurred (500) or an invalid request was identified (400).

```
{
    "data": {
        "id": "addf120b430021c36c232c99ef8d926aea2acd6b",
            "status": "200",
            "time": 1741706726
    }
}
```

Using the another MQTT client is possible to update the user parameters.



Figure 4.6: User paramters update example using the MQTTX client.

5 SECURE CONNECTION WITH WEGNOLOGY BROKER

This setup will use a PLC200 programmable logic controller connected to the WEGnology broker through the Internet.

Although the PLC200 is capable of connecting to the Internet, the network managers must take measures to ensure the system security as already disclaimed at IMPORTANT NOTICE ABOUT CYBERSECURITY AND COMMUNICATIONS.

5.1 IP ADDRESS AND NETWORK CONFIGURATION

5.1.1 PLC200 IP ADDRESS CONFIGURATION

The PLC200 IP address should be assigned by a DHCP server:

P0850 IP Address Config: 1 (DHCP).

5.1.2 PLC200 DATE CONFIGURATION

The TLS protocol, used for securing the connection, checks the date in each certificate sent by the server during the handshake process. So is necessary to have a correct date configured at the PLC200. A way to guarantee the time synchronization is using the Simple Network Time Protocol (SNTP). To activate the SNTP, the user must configure the server IP and update interval:

- P0770 SNTP Server 1: 200.160.7.186 (a.st1.ntp.br)
- P0774 SNTP Server 2: 201.49.148.135 (b.st1.ntp.br)
- **P0779 SNTP Update Interval**: 60

5.2 BROKER PREREQUISITES

For information on how to create a device in the WEGnology broker, please refer to the official documentation. In order to follow this application the user must create a **Standalone Device** and must add the following "Number" attributes for the 20 user parameters:

SECURE CONNECTION WITH WEGNOLOGY BROKER

Attribute Name	Data Type		user14	# Number v] -
user1	# Number V	-	Attribute Name	Data Type	
Attribute Name	Data Type		user15	# Number v] 🗕
user2	# Number V	-	Attribute Name	Data Type	
			user16	# Number V] -
Attribute Name	Data Type				
user3	# Number V	-	Attribute Name	Data Type	
			user17	# Number V	-
Attribute Name	Data Type		Attribute Name	Data Tuno	
user4	# Number V	-			
			userio	# Number	
Attribute Name	Data Type		Attribute Name	Data Type	
user5	# Number V	-	user19	# Number V] -
Attribute Name	Data Type		Attribute Name	Data Type	
user6	# Number V	-	user20	# Number V] -
			Attribute Name	Data Type	
Attribute Name	Data Type			Select	
user7	# Number V	-			
Attribute Name	Data Type		Add Attribute		
user8	# Number V	-	Include optional description fields		
Attribute Name	Data Type				
user9	# Number v	-	Update Attributes Cancel		

Figure 5.1: WEGnology device attribute creation.

The device will need credentials to connect to the broker, the user must generate and associate them with the created device, then it must also copy the Client ID, Access Key and Access Secret that will be used later when configuring the PLC200 client.

The user must supply the PEM formatted CA certificate for the broker at broker.app.wnology.io:

broker.app.wnology.io		
broker.app.wnology.io	×	← Security ×
Connection is secure	>	broker.app.wnology.io
Ocokies and site data	>	Connection is secure
Site settings	ß	rour information (for example, passwords or credit card numbers) is private when it is sent to this site. <u>Learn more</u>
(i* About this page Learn about its source and topic	Ľ	Certificate is valid

Figure 5.2: Accessing WEGnology certificate.

Certificate Viewer: *.app.wnology.io	×
General Details	
Certificate Hierarchy	
* USERTrust RSA Certification Authority	
*.app.wnology.io	
Certificate Fields	
♥ USERTrust RSA Certification Authority	
Version	
Serial Number	
Certificate Signature Algorithm	
Issuer	
▼ Validity	
Not Before	-
Field Value	
Expo	rt

Figure 5.3: Exporting the certificate.

Depending on the extension the browser saves the certificate, it may be necessary to change the file type to ".pem" or ".cer".

5.3 PLC CONFIGURATION WITH WPS

After creating a new configuration in WPS 3.10, add a new MQTT file in the "Communications" > "MQTT" folder. In the "Basic Configuration" tag fill out:

- Broker URL: broker.app.wnology.io
- Client ID: <Client ID created at the WEGnology platform>
- Username: <Access Key created at the WEGnology platform>
- Password: <Access Secret created at the WEGnology platform>
- Publication Interval: 60
- Publish Topic: wnology/<Client ID>/state
- Security: MQTTS (MQTT with TLS v1.2).
- Certificate: <Import the certificate extracted in the prerequisites step>

For the advanced tab:

- Allow Subscriptions: True
- Subscribe Topic: wnology/<Client ID>/command
- Response Topic: wnology/<Client ID>/state
- Broker Port: 8883
- **DNS server**: 8.8.8.8

- Keep Alive: 30
- Quality of Service: 0
- Clean Session: True

Adva	anceu		basic configuration Adva	inceu	_
Broker URL: Client ID: Jsername: Password: Publication Interval(s): Publich Topic:	broker.app.wnology.io XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		Allow Subscription: Subscribe Topic: Response Topic: Broker Port: DNS Server: Keep Alive(s): Quality of Service (QoS):		
Security. Certificate: Sf7dNXGiFSeUHM9h4ya7 XHIKYC6SQK5MNyosycdiy qS3fuQL39ZeatTXaw2ewl VXyNWQKV3WKdwrnuW L6KCq9NjRHDEjf8tM7qtj: jjxDah2nGN59PRbxYvnKk END CERTIFICATE	Import Ib6NnJSFd5t0dCy5oGzuCr+yDZ- A5d9zZbyuAIJQG03RoHnHcAP9 h0qpKJ4jjv9cJ2vhsE/zB+4ALtRZh ih0hKWbt5DHDAff9Yk2dDLWKK 3u1cliuPhnPQCjY/MiQu12ZlvV55 Kj9	Remove 4XUmFF0sbmZgIn/f Dc1ew91Pq7P8yF1r 18tSQZXq9EfX7mR8 AGwsAvgnEzDHNbi IjFH4gxQ+6IHdfG	Clean Session:	• True	○ False

Figure 5.4: WEGnology config in WPS.

After configuring the client, build the resource and download it to the PLC200. After this step the client should attempt a connection.

Recurso PLC200	
Dispositivo PLC200	Versão 1.4.X
Opções	
🗌 Para/Inicia o programa automaticamente	Download do código fonte
Inicializa variáveis	 Escrita do setup (parâmetros) Recursos de comunicações
Arquivos de memória interna	Download de valores de monitoração
ilesystem.bin - 11 mar. 2025 12:54:03	Selecionar
nfo.bin - 11 mar. 2025 12:54:03	
nqtt.bin - 11 mar. 2025 12:54:03	
etain.bin - 11 mar. 2025 12:54:03	
armap.bin - 11 mar. 2025 12:54:05	
olacile.biti - 11 iliai, 2023 12,34,03	
adder bin - 11 mar 2025 12:54:03	

Figure 5.5: Download resource files to PLC200.

5.4 SENDING AND RECEIVING MESSAGES

5.4.1 PUBLISH

After configured and connected, the PLC200 client should start publishing data at the interval defined in the configuration step. The payload published contains one object with the value for each of the twenty user parameters available (P0800 to P0838) and also an extra object with the sampling time in epoch format. The following JSON is an example of a payload published by the PLC200.

```
ł
        "data": {
                 "user1": 5,
                 "user2": 0,
                 "user3": 10,
                 "user4": 0,
                 "user5": 0,
                 "user6": 999,
                 "user7": -10,
                 "user8": 0,
                 "user9": 0,
                 "user10": 2147483647,
                 "user11": 0,
                 "user12": 0,
                 "user13": 0,
                 "user14": 0,
                 "user15": 71254,
                 "user16": 0,
                 "user17": 0,
                 "user18": 0,
                 "user19": 0,
                 "user20": 0
        },
        "time": 1737029549
}
```



NOTE!

User parameters are sampled at the publishing interval, but if for some reason the client is disconnected at the time, The message will be store in the PLC200 volatile memory and published upon reconnection. The client can store up to 50 samples.

In the WEGnology console is possible to check for the PLC200 MQTT client status and see the user parameters publications.

SECURE CONNECTION WITH WEGNOLOGY BROKER



Figure 5.6: PLC200 connection to WEGnology and data publication.

.

5.4.2 SUBSCRIBE

After connecting to the broker, the PLC200 client subscribes to the topic configured and starts receiving messages from other clients, like the MQTTX client for example. The PLC200 allows for the other clients to publish messages that update the user parameters (P0800 to P0838) values. The PLC200 client expects a payload in the following format:

```
{
        "payload": {
                "id": "addf120b430021c36c232c99ef8d926aea2acd6b",
                "args": {
                         "1": 0.
                         "2": 0,
                         "3": 0,
                         "4": 0,
                         "5": 0,
                         "6": 0,
                         "7": 0,
                         "8": 0,
                         "9": 0,
                         "10": 0,
                         "11": 0,
                         "12": 0,
                         "13": 0,
                         "14": 0,
                         "15": 0,
                         "16": 0,
                         "17": 0,
                         "18": 0,
                         "19": 0,
```

```
"20": 0
}
}
```

Where each key inside the "payload.args" object refers to user parameter number (1 to 20) and the corresponding value to be written. Each message doesn't need to contain all user parameters, only the ones that should be updated. There is also an optional "payload.id" object that can store a string of up to 40 characters, this field value will be send with the response payload and is meant to be used as a message identification mechanism, if the object is not present in the payload the response payload will have an ID object with value 0.

Upon receiving a publication, the client will update the user parameters and reply in the configured response topic with a message containing the request status and ID. The status field correspond to the HTTP response status codes and indicate whether the requested update was successful (200), an internal error occurred (500) or an invalid request was identified (400).

```
{
    "data": {
        "id": "addf120b430021c36c232c99ef8d926aea2acd6b",
            "status": "200",
            "time": 1741706726
    }
}
```

A message can be published to the PLC200 client with the "Send Device Command" device action in the WEGnology broker interface:

Send Device Cor	nmand		\times
Device			
PLC200_APPLICAT	ION_NOTE		
Commands instruct yo	COMMAND our devices to perfo	INFO	e, start , stypically
up to your device's firr JSON payload if the c example, the start c "resolution": 1080,	nware. Commands ommand requires ommand could inc "fps": 60 }.	3 also include an optiona additional parameters. F Ilude parameters like {	al string or or
Command Name			
test			
Command Payload ("id": "19853ec "args": { "2": 2, "9": 10, "5": -55 } }) (View Example) 79ad7708011b4f9	540360069715c40cb3"	,
Treat payload as J	SON		
Send Command	Cancel	C Keep n	iodal open

Figure 5.7: Send update to PLC200 via WEGnology console.

If the "id" string and the "status" number attributes were not created, the WEGnology platform will ignore the

response payload but the user parameters will be updated successfully.

```
Device PLC200_APPLICATION_NOTE reported its
state
MQTT wnology/67866799d718771c7b07c23a/state Mon Jan 20, 2025 12:10:56.374 GMT-03:00
Received Payload
▼ (root) {} 2 keys
    ▼ "data": {} 20 keys
       "user1": 1
       "user10": -90
       "user11": 0
        "user12": 0
        "user13": 0
       "user14": 0
       "user15": 0
        "user16": 0
        "user17": 0
       "user18": 9263
       "user19": 0
        "user2": 2
        "user20": 20
       "user3": 0
        "user4": 0
        "user5": -55
        "user6": 0
        "user7": 87654321
        "user8": 0
        "user9": 10
    "time": 1737385855
```

Figure 5.8: Cyclic paylod after update request.



BRAZIL WEG DRIVES & CONTROLS - AUTOMAÇÃO LTDA. Av. Prefeito Waldemar Grubba, 3000 89256-900 - Jaraguá do Sul - SC Phone: 55 (47) 3276-4000 Fax: 55 (47) 3276-4060 www.weg.net/br