Serratron IoT Gateway

SGateway

Version 1.10 - 2023/36

IMPORTANT

This user manual is intended for technicians and engineers with basic knowledge of both Serratron welding controllers and information technology (IT). The manual must be read and understood before attempting any operation with the welding controllers.

Serra Soldadura, S.A.U.
Polígono Industrial Zona Franca
Calle D, nº 29
08040 BARCELONA (Spain)

Telephone: +34 93 261 71 00
Technical assistance: +34 93 261 71 00 Ext.: 17200

Internet: http://www.serrasold.com
Downloads: http://serratron.serrasold.com

CONTENTS

Capítulo 1 1-1

INTRODUCTION 1-1
- Industry 4.0 1-1
- Cloud computing 1-1

Capítulo 2 2-1

SGateway 2-1
- Hardware 2-1
- Software 2-2
- Personalization 2-3
  - IP Address 2-3
  - Advanced features 2-5
- Debugging 2-5
- Node-RED 2-6

Capítulo 3 3-1

Programming interface 3-1
- Network 3-1
- MQTT 3-3
- MySQL 3-3
- OPC UA 3-3
- License 3-4
- Status 3-5

Capítulo 4 4-1

MQTT 4-1
- General 4-1
- Configuration 4-2
- Messages 4-4
  - Failures 4-4
  - Weldings 4-5
  - Programs 4-6
  - Maneuvers 4-8
- Broker 4-9

Capítulo 5 5-1

MySQL 5-1
- General 5-1
- Configuration 5-2
- Tables 5-3
  - Errors 5-3
  - Monitoring 5-4
  - Maneuvers 5-5
- Database manager 5-5

Capítulo 6 6-1

OPC UA 6-1
- General 6-1
- Configuration 6-2
- Data structures 6-3
  - Object Types 6-3
  - Methods 6-6
  - Objects 6-7
- UA Expert 6-8

●●●

Capítulo 1
INTRODUCTION

Industry 4.0

Industry 4.0 and its synonym Fourth Industrial Revolution are terms used to describe a fourth stage of technical evolution.

This is a new way of organising the means of production. The objective is to implement smart factories that are able to adapt to the production needs and processes better as well as to allocate resources more efficiently.

One of the technological foundations on which this orientation relies is the Internet of Things (IoT). This is a concept that refers to the digital interconnection of equipment to allow an improvement in the response to incidents, an improvement in the quality of manufactured products, predictive maintenance and, in short, a reduction in operating costs.

Industry 4.0 involves the complete digitisation of production chains through the integration of data processing technologies, intelligent software and sensors.

This entails accumulating large volumes of data that need to be transmitted and stored.

Cloud computing

Cloud computing is a model for offering computing services over a network, usually the Internet.

Cloud computing provides:

- Agility: The ability to offer improvements in the technological resources consumed by users depends on their providers.
- Scalability: The resources needed for the service can virtually change in real time depending on the needs at any given time. Therefore, costs are reduced to the bare minimum.
- Independence of location: Users can access the services regardless of their location.
- Shared storage devices. Applications can easily be moved from one physical server to another.
- Centralisation of data. Backup copies: Full recovery in case of loss. Reduction of downtime to a minimum.

•••

Capítulo 2
SGateway

Welding controllers, as a source of data, have to be incorporated into the transmission chain of Industry 4.0. Because of their technology or convenience, Serratron welding controllers do not have the ability to send those data directly to the cloud. This is why an intermediate stage or adapter is needed. This collects the data from the Serratron and sends them to the cloud using one of the established protocols.

The SGateway (Serratron IoT Gateway) is a gateway that interfaces between Serratron welding controllers and cloud computing.

It can be used with any welding controller that has an Ethernet interface. It currently supports the following SERRA devices:
- Serratron 300dp
- Serratron 100
- Serratron 100C
- MFC-3000 Ver. 2.x
- MFC-3000 Ver. 3.x
- MFC-4000 Ver 1.x
- PES-10
- PES-20B

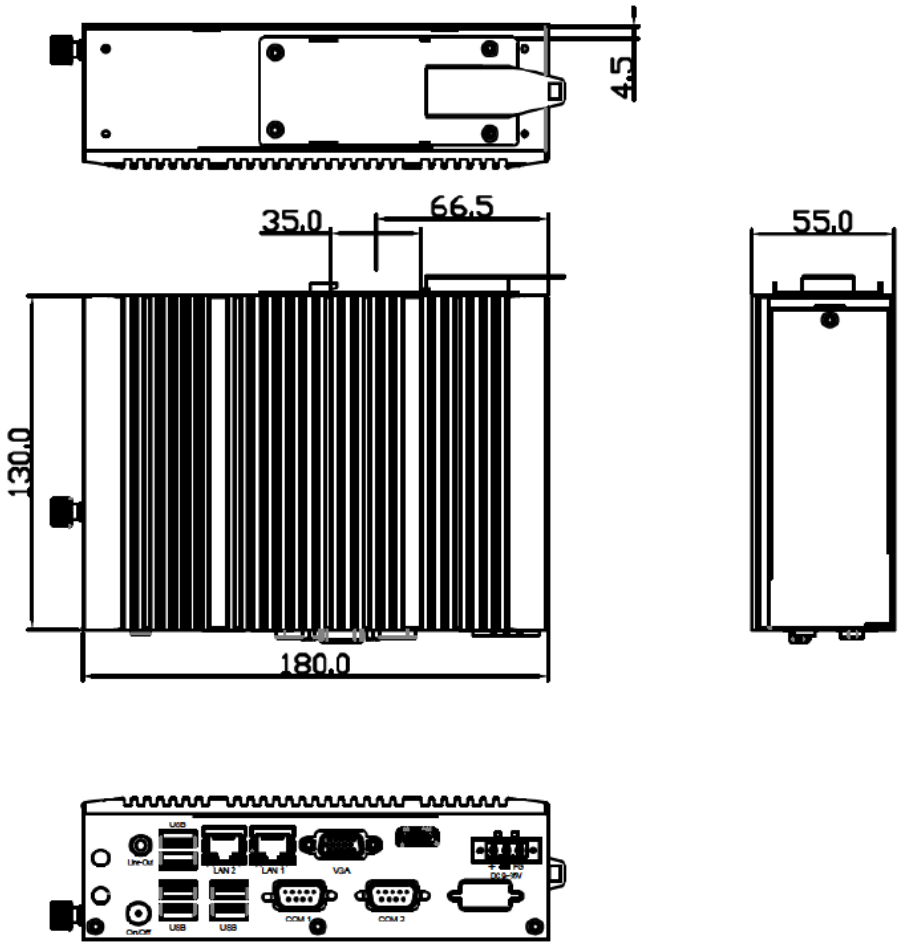
Up to 32 different devices can be connected.

It can be adapted to any of the protocols required by the industry. It currently supports MQTT, MySQL and OPC UA protocols.

The SGateway is offered as a platform that works autonomously without the need for a dedicated computer. It only needs two connections:
- a 24 V power supply and
- either a Copper Ethernet or a Wi-Fi connection.

Hardware

It is supplied in a small box for DIN-rail mounting. It requires a 24 V 1 A power supply.



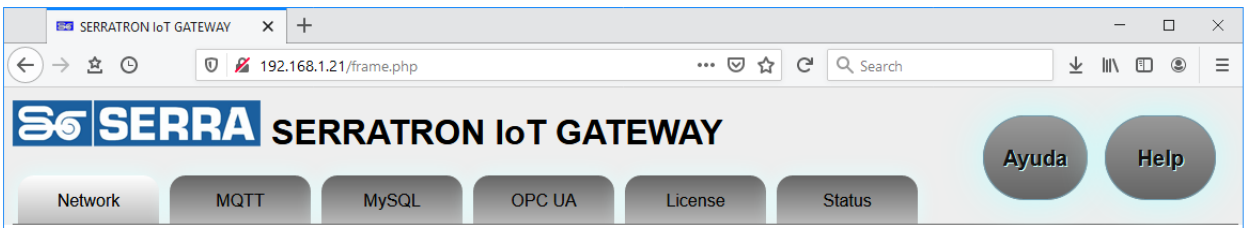
Software

The SGateway is configured via a web browser. No special program is therefore needed. Simply connect a computer to the same network as the SGateway and use Internet Explorer, Firefox or other similar browser.

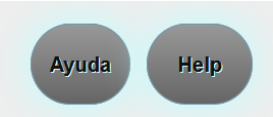
All texts are in English. This avoids any ambiguity in the translation into other languages.

To discover the IP address of the SGateway, see the **Dispector** chapter.

After typing the SGateway's IP address in the browser's address bar, the configuration web page will be shown.



These buttons appear on all screens



Download the application manual in Spanish or English, respectively.

See the meaning of the different tabs in the following chapters.

Personalization

IP Address

The SGateway is factory-set with the IP address 192.168.1.100, which may not be suitable for a particular user.

There are two possible methods to change the IP address: Direct connection and Dispector. The Dispector method takes precedence over Direct connection. Once the IP has been modified through the Dispector, modifications through Direct connection have no effect.

Direct connection

To change the IP Address, Mask and Gateway, you must employ a monitor with an HDMI interface, an HDMI cable and a USB mouse, and connect them to the Sgateway.

After connecting the power, a graphical environment will appear on the screen that will allow you to make the necessary changes.

Choose the menu "**Settings – Network – Ethernet (enpxs0)**"

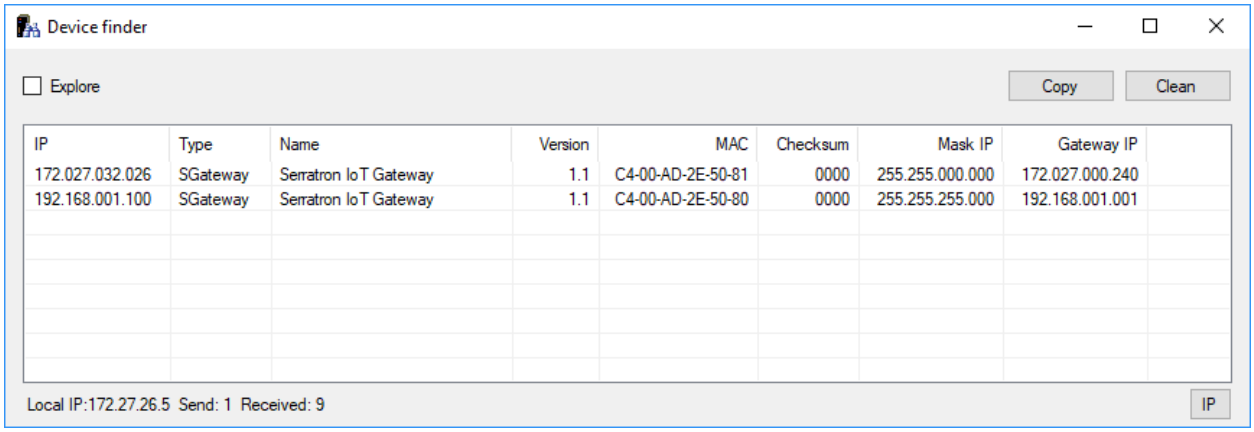
Next, choose the Ethernet port to be programmed, **enp3s0 (LAN1) or enp2s0 (LAN2)**, and finally, in the **IPv4 Settings** tab, choose **Method: Manual** and write the appropriate addresses.

Dispector

Dispector is a SERRA devices discoverer that is supplied together with the CPC-Connect control programming software.

Useful for both changing and discovering the device's IP addresses.

After starting up (by selecting Network - Dispector menu), press button **Explore**, and the following information will appear on the screen:



The device has two Copper Ethernet interfaces, each with its own particular IP configuration.

Only those interfaces that have a cable connected will appear in the Dispector.

In the figure example, there are two interfaces connected.

One corresponds to the IP that is supplied by default, 192.168.1.100, and the other has been configured by DHCP to the address 172.27.32.26.

The meaning of the different columns is as follows:

Type
Type of device. In the case of Serratron IoT Gateway, SGateway will appear (in older versions of CPC-Connect it will appear 0x00B0). If the device is a welding control, it will show MFC-3000, for example.

Name
Device Name. It is recommended to assign a name that helps to distinguish between different SGateway devices.

Version
Software version that is running on the device.

IP, Mask IP, Gateway IP
IP address, Mask and Gateway of the interface. These parameters will be those that have been configured manually, or those assigned by the address server in the case of DHCP mode.

To configure the interface in DHCP mode, program the IP address = 127.0.0.0

Verify that the SGateway mask is the same as that of the Serratron to be connected.

MAC
Physical address, unique for each interface. The one ending with an even number corresponds to the LAN A connector, and the one ending with an odd number corresponds to LAN B, although this depends on the manufacturer and could change.

Checksum
Always 0000. Present for compatibility with welding controls.

To modify the configuration of an interface, double click on its line. In the dialog box that appears, fill in the data you want to change.

Advanced features

SGateway works under a Linux environment.

A user with the appropriate knowledge will be able to make different modifications to the system:

- Add users to the included MQTT Broker
- Add users to the included MySQL server
- Change IP address or Gateway
- Change the root password

The O.S. could be accessed through the SSH protocol using the following parameters:
IP Address: **192.168.1.100**
User: **serra**
Password: **serra**

Debugging

The SGateway generates a dump file where it saves the content of the Status tab, as well as other important warnings and internal actions. You can view the content of this file by typing in the browser's address bar:

http://172.27.31.161/SGateway.log (being 172.27.31.162 the IP address of the SGateway)

The screen will show something like:

```
[2023-08-29 17:58:12]    INFO  : SERRATRON IoT GATEWAY Version 1.10
[2023-08-29 17:58:12]    INFO  : Program Start
[2023-08-29 17:58:12]    INFO  : Shared memory successfully created
[2023-08-29 17:58:12]    INFO  : License active
[2023-08-29 17:57:15]    INFO  : OPC UA Server started
[2023-08-29 17:57:15]    TRACE : StackTrace - Most recent calls appear first:
                                 ?? ??:0
                                 UA_Server_run_startup at ??:?
                                 UA_Server_run at ??:?
                                 OPCUA_thread(void*) at sopcua.cpp:167
                                 ?? ??:0
                                 ?? ??:0
[2023-08-29 17:57:15]    DEBUG : Application_Exit
```

Node-RED

Node-RED is a flow-based development tool for visual programming, for wiring together hardware devices, APIs and online services as part of the Internet of Things.

Serratron IoT Gateway offers support to integrate Node-RED into its operation and allow the exchange of data between protocols not initially foreseen.

It is offered as a custom plug-in prepared according to the client's needs.

When active, a button appears on the Status screen giving access to the flow editor for evaluation or modification.

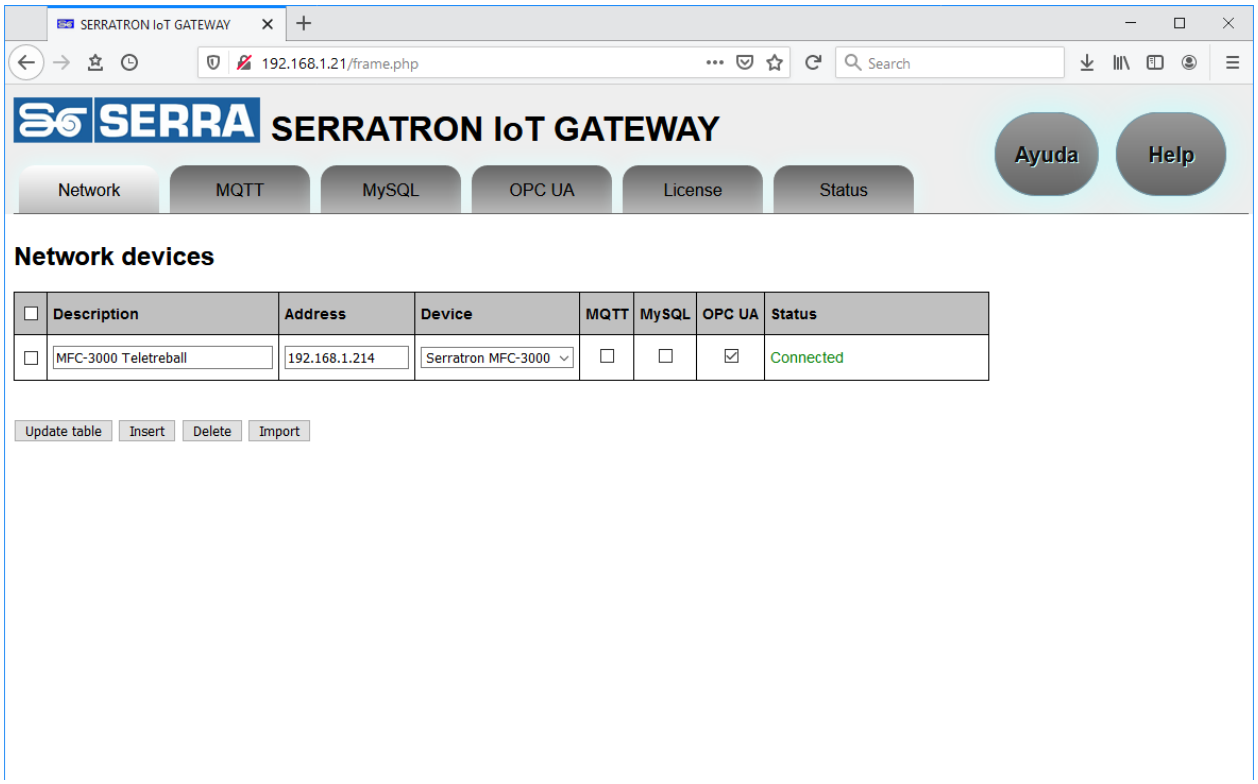
Capítulo 3
Programming interface

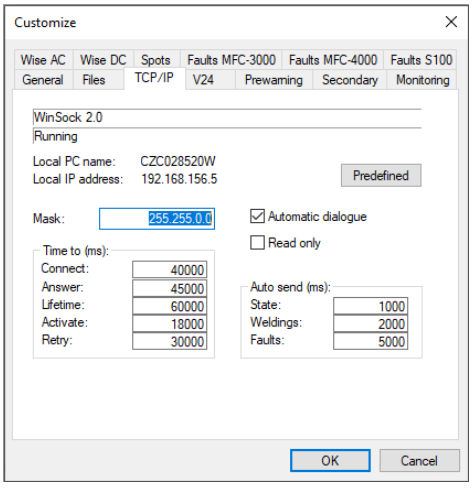
It features several tabs.

Network

List of welding timers to be connected to the SGateway.

Up to 32 different timers can be connected





Welding timers can be connected simultaneously to the CPC-Connect and the SGateway. The only precaution to take into account is that the **Automatic Dialogue** mode is activated in the TCP/IP tab of the **Options – Customize** dialog of CPC-Connect (this is the default option).

The information that is shown in the table for each of them reads as follows:

Description Name of welding controller.
This will be the text that identifies it within the communications protocol. It cannot be repeated.

Address The welding controller IP address.
It identifies the controller within the communications network. It cannot be repeated.

Device Serratron type.
This specifies the actual type of welding controller we will be communicating with.
The types shown in the drop-down window are the following:

- **Serratron MFC-3000 V2**
  This includes the entire family of Serratron MFC-3000 and MFC-3500 with firmware version 2.x.
- **Serratron MFC-3000 V3**
  This includes the entire family of Serratron MFC-3000 and MFC-3500 with firmware version 3.x.
- **Serratron MFC-4000 V1**
  This includes the entire family of Serratron MFC-4000 and MFC-4500 with firmware version 1.x.
- **Serratron 100**
  For both Serratron 100 and Serratron 100C.
- **PES-10/20B**
  This includes the PES-10 and PES-20B electric welding gun controllers.
- **Serratron 300dp**
  AC welding controller for three groups of thyristors.

MQTT This activates the MQTT protocol.
Please refer to the MQTT section for the configuration options.

MySQL This activates the MySQL protocol.

OPC UA This activates the OPC UA protocol.
Please refer to the OPC UA section for the configuration options.

Status Communications status.
It states whether the SGateway is communicating with the corresponding welding controller and whether there is a problem.

A number of buttons are also displayed to issue several commands:

All changes made to the table are updated in the SGateway.
Do not forget to press this button when all changes have been made.

This inserts a new device at the end of the table.
The new device is inserted in a similar way to the first one selected. If none is selected, the last one in the table will be taken as a reference. This means that the device and protocols will be copied from the selected device and the IP address will be that of the selected device + 1.

This deletes all selected devices from the table.

Import

This is used to easily add multiple devices to the table.
When this button is pressed, a window where you can paste a list of devices in comma-separated, tab-separated or semicolon-separated values.
This list can easily be generated from a CPC-Connect network file saved in Access format by proceeding as follows:

- Open a new blank Excel spreadsheet.
- Under the **File** menu, select the **Access Databases** format
- Open the **CPCX_Network (Default).mdb** file in the CPC-Connect directory or any network file that has been previously saved. From the list of tables displayed, select **NetworkConfig**.
- Copy the lines of the devices you want to import.
- Paste them on the window.

MQTT

MQTT protocol configuration parameters

Please refer to the relevant chapter for further details.

MySQL

MySQL protocol configuration parameters

Please refer to the relevant chapter for further details.

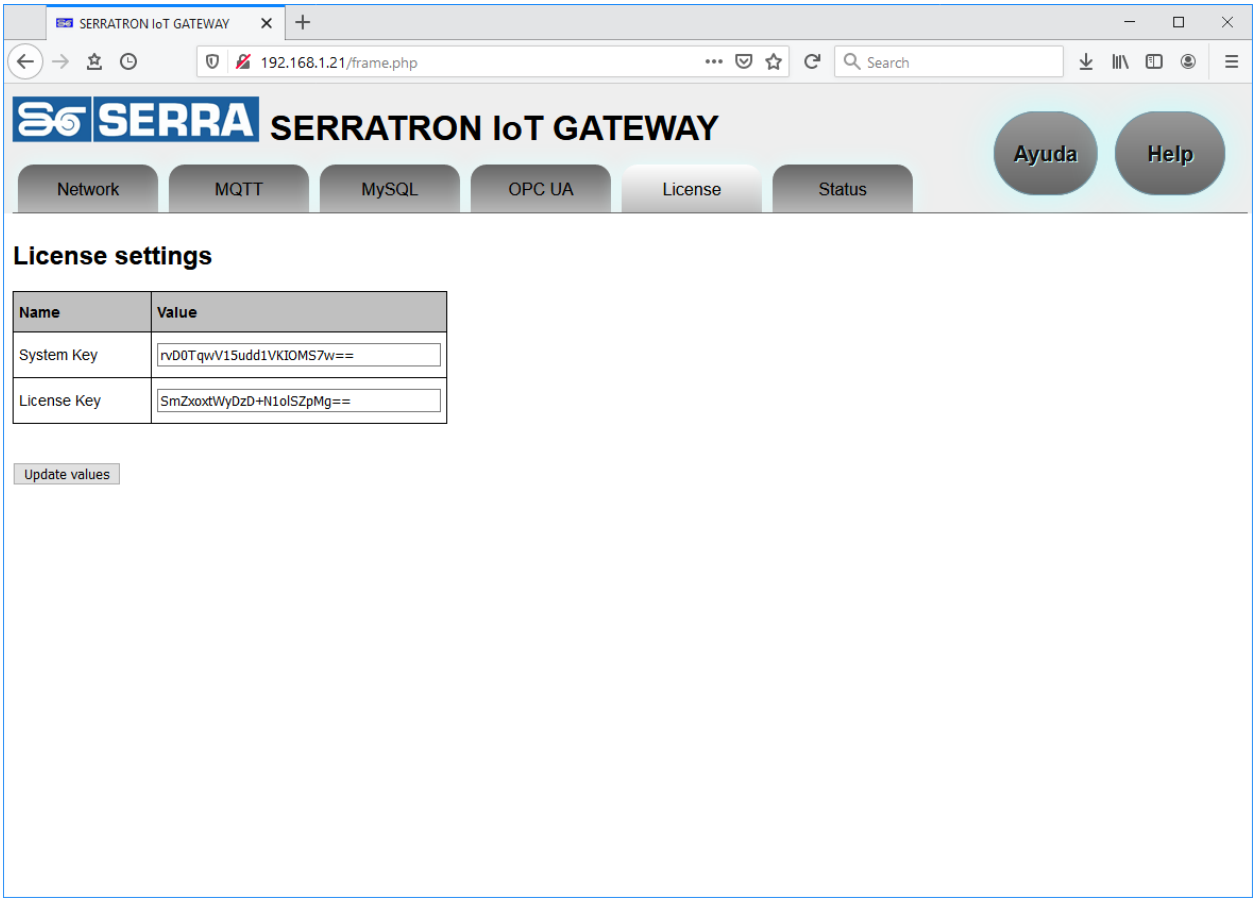
OPC UA

OPC UA protocol configuration parameters

Please refer to the relevant chapter for further details.

License

Activating the program license



The license activation procedure is the following:

When you first start up the program, a five-day temporary license is activated. That allows you to evaluate the program without the need to purchase a permanent license.

To request a permanent license, send the contents of the **System Key** field to Serra Soldadura's technical service. This can be copied and pasted onto an email message.

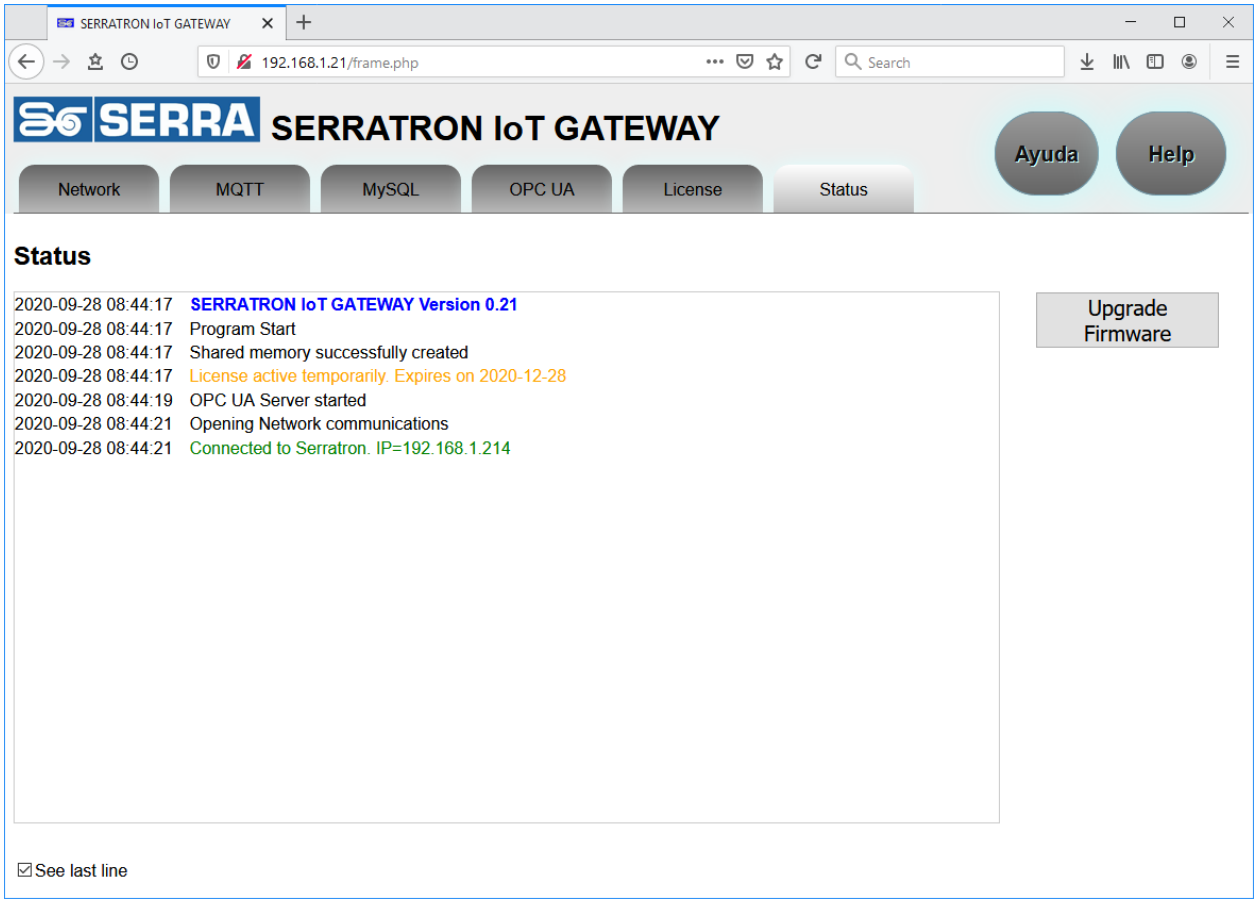
Then, you will receive a string of characters to paste into the **License Key** field. Paste it and press the **Update values** button.

When the license expires, no communication is possible with the Serratrons or the selected protocols.

The license status is shown on the **Status** tab you will see when you start up the program.

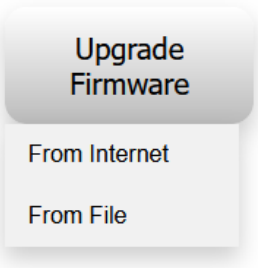
Status

This is an information screen for the program events.



The event colour denotes the severity of the event. Green for successful communications. Red signals a problem, usually in communications. In this case, there is nothing to be done in principle. If the SGateway has to communicate with a Serratron or a protocol client or server and is unable to do so, it will keep trying until it succeeds.

There are also some buttons to carry out different actions:



Allows to update the SGateway version.
After pressing the button, two options appear:

From Internet — **T**he latest version available on the Serra server will be displayed, and the option to install it will be offered.

For this option to work, the SGateway must be connected to a network with internet access.

From File — It offers the possibility of choosing a file with the new version, stored on the local computer.

There is a checkbox at the bottom of the screen:

Delete list — Clears the list on the screen. Displays a notice to indicate that the list has been deleted.

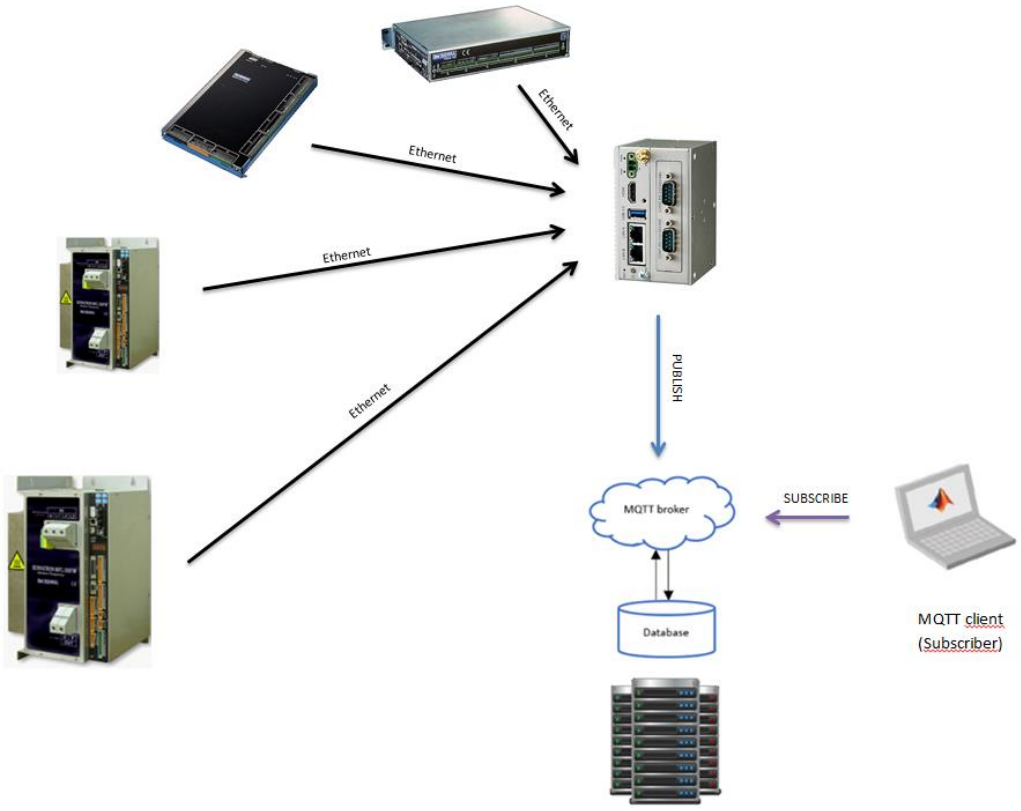
☑ **See last line** — If checkbox is active, forces the last line added to the list to always be displayed. Otherwise, the display does not change, allowing you to browse the different lines by using the scroll bar.

Capítulo 4
MQTT

General

Message Queue Telemetry Transport (MQTT) is a protocol used for machine-to-machine (M2M) communication in the Internet of Things. This protocol is used for sensor communications because it consumes very little bandwidth and can be used in most embedded devices with few resources. MQTT's architecture follows a star design with a central node that acts as a server or broker with a capacity of up to 10,000 clients. The broker manages the network and transmits the messages. To keep the channel active, clients periodically send a package (PINGREQ) and wait for the broker's response (PINGRESP).

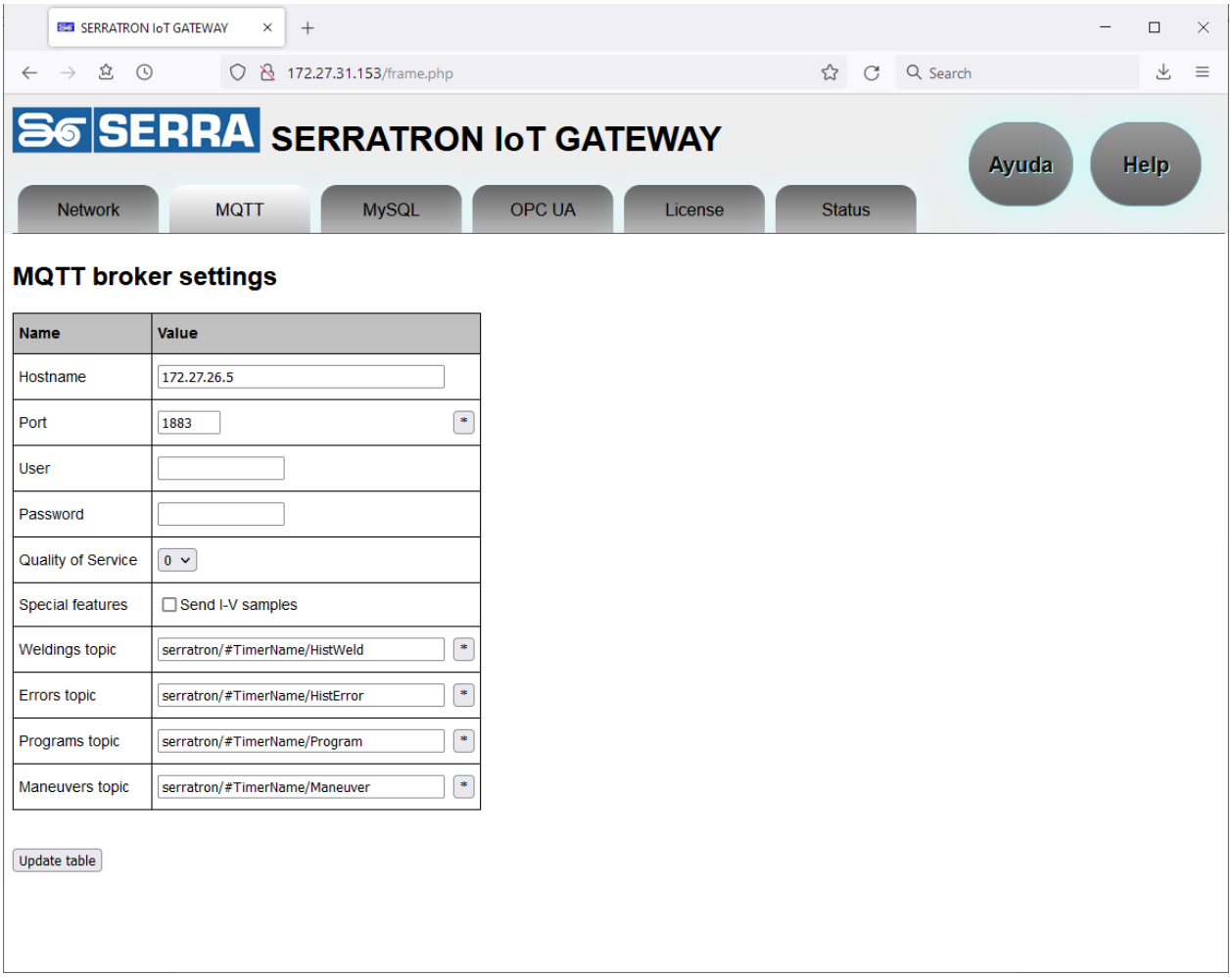
The communication is based on topics that clients who publish messages create and nodes that wish to receive them must subscribe to them. Communication can be one-to-one, or one-to-many. Topics are represented by threads and have hierarchical structures. Each hierarchy is separated by a slash (/). For example, “building1/plant5/room1/temperature” or “building3/plant0/room3/noise”. In this way, hierarchies of clients that publish and receive data can be created. This allows a node to subscribe to one specific topic ("building1/plant5/room0/temperature") or several ("building1/plant5/#").



Configuration

The protocol version used is 3.1

If you click on the MQTT tab, you can configure the following parameters:



Their meaning is as follows:

Hostname This is the broker's address.
Either the broker's URL or its IP address.

Port Communications port.
The MQTT communications protocol is assigned to port 1883 by default, although it can be changed if, for example, the broker has been configured to use a different one.

If you press the * button, the checkbox will be updated by default.

User Username.
If the broker has activated the password system, the name of an authorised user must be entered.
If the password system is not activated, this field should remain blank.

Password Password.
Password corresponding to the above user. If the password system is not activated, the content of this field is irrelevant.

QoS Quality of Service.
This shows the degree of agreement required in communications between clients and brokers. Any of the following standardised values can be chosen:
Value 0: The message is delivered only once at most. If the client is not available at that time, the message will be lost.
Value 1: The message must be delivered at least once.
Value 2: The message must be delivered exactly once.

Special features Allows to enable special functions, only of interest to some users.

Send I-V samples It sends the current and voltage samples for each weld, too. A sample is available every 2 milliseconds.

Weldings topic Topic of weldings.
Name of the topic corresponding to the welding messages. The topic can be customised for each device by including the **#TimerName** keyword, which will be replaced in each topic with by the name of the device, as it appears in the **Network** tab.
PES-10/20B devices will not publish this topic.

Errors topic Topic of errors.
Name of the topic corresponding to the error messages. The topic can be customised for each device by including the **#TimerName** keyword, which will be replaced in each topic with by the name of the device, as it appears in the **Network** tab.

Programs topic Topic of programs.
Name of the topic corresponding to the program parameter messages. The topic can be customized for each device by including the **#TimerName** keyword, which will be replaced in each topic with by the name of the device, as it appears in the **Network** tab.
Only MFC-3000 (version 2.68 or later, or version 3.6 or later) or MFC-4000 devices will publish this topic.

Maneuvers topic This is the topic of maneuvers.
Name of the topic corresponding to the maneuvers messages. The topic can be customised for each device by including the **#TimerName** keyword, which will be replaced in each topic with by the name of the device, as it appears in the **Network** tab.
Only PES-10/20B devices will publish this topic.

A button is shown at the bottom of the screen:

All changes made to the table are updated in the SGateway when this button is pressed.
Do not forget to press this button when all changes have been made.

Messages

The content of MQTT messages are transmitted in text format so that it is easily understandable by both a person and a machine. We have chosen a text format so that it is easily understood by both humans and machines. We have chosen JSON for the text formats, as it is widely used and has many tools for computer processing. It also allows you to easily add more data if needed, or ignore data that are not needed.

All messages, including parameter names and content, are sent in English. This prevents errors of interpretation and makes the messages properly understood anywhere in the world.

Each message, in addition to the specific fields corresponding to its type, provides details of the sender and the date of the event.

The date is sent in two formats:

Date: Local time in text format. The criterion is to specify the longest time periods first and then the shortest (year, month, day). This means that the alphabetical order matches the chronological order.

Epoch: Also known as the Unix Time Stamp, this specifies the number of seconds since 1 January 1970 in GMT time.

The possible messages are described below:

Failures

Both the welding controllers and the electric welding gun controllers send these messages.

They contain the relevant details of a fault:

```
{
  "Name": "R08L110",                    Device name on the Network tab
  "Type": "Serratron MFC-3007CN",       Serratron sub-type
  "Program": {                          Details of the program that failed (if applicable)
    "Number": 0,                        Number
    "Code": 0                           Welding point code
  },
  "Error": {                            Specific error data
    "Text": "Weld NO",                  Text
    "Number": 20,                       "Number"
    "Index": 0,                         "Index"
    "Duration": 0.3                     "Duration"
  },
  "Date": "2019-05-24 08:38:25",        Error date and time
  "Epoch": 1558687105                   Error epoch
}
```

Weldings

Only welding controllers send these messages.

They contain the relevant details of a welding:

```
{
  "Name": "R08L210",                    Device name on the Network tab
  "Type": "Serratron MFC-3007CN",       Serratron sub-type
  "Program": {                          Details of the program used for the welding
    "Number": 1,                        Program number
    "Code": 144332                      Welding point code
  },
  "Welding": {
    "KSR": 2,                           Current regulation mode
    "Weld1": {                          Parameters of welding time 1
      "Phau": 0,                        Used phase
      "Iu": 0.00,                       Used current
      "Pham": 0,                        Measured phase
      "Im": 0.00                        Measured current
    },
    "Weld2": {                          Parameters of welding time 2
      "Phau": 8,                        Used phase
      "Iu": 7.00,                       Used current
      "Pham": 16,                       Measured phase
      "Im": 7.03                        Measured curren
      "Wtm": 200                        Measured welding time
    },
    "Weld3": {                          Parameters of welding time 3
      "Phau": 0,                        Used phase
      "Iu": 0.00,                       Used current
      "Pham": 0,                        Measured phase
      "Im": 0.00                        Measured current
    },
    "Length": 200,                      Welding duration, in ms
    "Energy": 6749,                     Welding energy
    "Resis": 162,                       Welding point resistance
    "FPVu": 3.2,                        Force used by the proportional valve
    "AnI1": 43,                         Measured value at the analogue input
    "SpotCnt": 39                       Weld spot counter
    "TDCnt": 4,                         Tip dressing counter
    "Current": [                        Tabla con las muestras de corriente
      8.80,
      8.80,
      8.66,
      8.66
    ],
    "Voltage": [                        Tabla con las muestras de tensión
      1431,
      1419,
      1465,
      1465
    ]                                   Datos opcionales
  },
```

```
  "Error": {                              Details of the welding error (if applicable)
    "Number": 0,                          Number
    "Index": 0                            Index
  },
  "Date": "2019-05-24 08:37:57",          Date and time of welding.
  "Epoch": 1558687077                     Welding epoch
}
```

Programs

Only Serratron 100 and MFC-3000 welding timers send this message, with the following particularities:

- The data of all non-blocked programs are sent after establishing a connection between the SGateway and the Serratron.
- The data of all non-blocked programs is sent at midnight.
- The data of a program is sent after it has been modified by means of the TP-10.
- MFC-3000 Ver 2.68 (and later), MFC-3000 Ver 3.x and MFC-4000 send the data of a program after it has been changed by means of CPC-Connect.

They contain the relevant parameters of a program:

```
{
  "Name": "R08L210",                      Device name on the Network tab
  "Type": "Serratron MFC-3007CN",         Serratron sub-type
  "Program": {
    "Number": 5,                          Program number
    "Code": 1005                          Welding point code
  },
  "Parameters": {                         Parámetros correspondientes al programa "Number"
    "Electrode": 1,                       Electrode number
    "ElLife": 20000,                      Electrode life
    "KSR": 2,                             Current operation mode
    "Agress": 0,                          Agressiveness
    "RFault": 0,                          Fault resistance value
    "RWarn": 0,                           Warning resistance value
    "Fsq": 200,                           First squeeze time
    "Sqz": 100,                           Squeeze time
    "Ct1": 10,                            Cool time 1
    "Weld1": {                            Values corresponding to weld time 1
      "Phai": 1,                          Initial phase
      "Phaf": 1,                          Final phase
      "Ii": 0.01,                         Initial current
      "If": 0.01,                         Final current
      "Wt": 0                             Welding time
    },
```

```
    "Weld2": {                          Values corresponding to weld time 2
      "Phai": 1,                        Initial phase
      "Phaf": 1,                        Final phase
      "Ii": 8.00,                       Initial current
      "If": 0.01,                       Final current
      "UpSl": 5,                        Up slop time
      "Wt": 100,                        Welding time
      "DwSl": 0                         Down slop time
    },
    "Ct2": 10,                          Cool time 2
    "Ct3": 10,                          Cool time 3
    "Weld3": {                          Values corresponding to weld time 3
      "Phai": 1,                        Initial phase
      "Phaf": 1,                        Final phase
      "Ii": 0.01,                       Initial current
      "If": 0.01,                       Final current
      "Wt": 0                           Welding time
    },
    "Hold": 400,                        Hold time
    "Off": 10,                          Off time
    "FPVi": 2.0,                        Initial PV force
    "FPVf": 0.0,                        Final PV force
    "Imp": 1,                           Number of impulses
    "Tol+": 10,                         Up tolerance
    "Tol-": 10,                         Down tolerance
    "Prealarm": 0,                      Prealarm
    "Step": 0,                          Compensation of electrode wear
    "TDNum": 0,                         Tip-dressings allowed
    "TDFreq": 0                         Tip-dressing frequency
  },
  "Date": "2022-03-04 15:45:43",        Date and time of the message
  "Epoch": 1646401543                   Epoch of the message
}
```

Maneuvers

Only electric welding gun controllers send these messages.

They contain the relevant details of a maneuver:

```
{
  "Name": "Larguero AD",                 Device name on the Network tab
  "Type": "PES-20B",                     Sub-type of electric welding gun controller
  "Gun": 2,                              Number of current gun
  "Program": {                           Details of the program that has acted
    "Number": 1,                         Number
    "Code": 876363                       Point code
  },
  "Maneuver": {
    "Type": 1,                           Maneuver type
    "ThPl": 2.2,                         Measured sheet metal thickness
    "Fric": 21,                          Measured friction
    "ElHWe": 0.4,                        Electrode holder wear
    "ElWe": 2.4,                         Electrode wear
    "ElDisp": 0.4,                       Displacement of the fixed electrode
    "MMTemp": 78,                        Main motor temperature
    "CMTemp": 32,                        Compensation motor temperature
    "Counter": 78654,                    Maneuver counter
  },
  "Error": {                             Specific error data (if applicable)
      "Number": 21,                      Number
      "IdxError": 0                      Index
  },
  "Date": "1918-9-12 01:11:45",          Date and time of maneuver
  "Epoch": 1551085258                    Maneuver epoch
}
```

Broker

To facilitate start-up, or for installations with a small number of controls, the SGateway incorporates a Broker.

Among the various options, we have chosen **mosquitto**, because it is Open Source (distributed under the EPL / EDL license), lightweight and multiplatform, and because it is suitable for use on low-power servers.

To access the Broker, these parameters must be programmed:
Hostname: **localhost**
Port: **1883**
User: **serratron**
Password: **serra**

When installing **mosquitto** on another server, keep in mind that in most cases it is necessary to edit the configuration file **mosquitto.conf**:

- Change **#listener** to **listener 1883** to allow connections to outside the computer
- Change **#allow_anonymous false** to **allow_anonymous true** to allow access without username or password, during the first tests..

Start the program taking into account the configuration file:

```
user:/home$ mosquitto -v -c mosquitto.conf
```

•••

Capítulo 5
MySQL

General

MySQL is an open source relational database management system (RDBMS) based on structured query language (SQL).

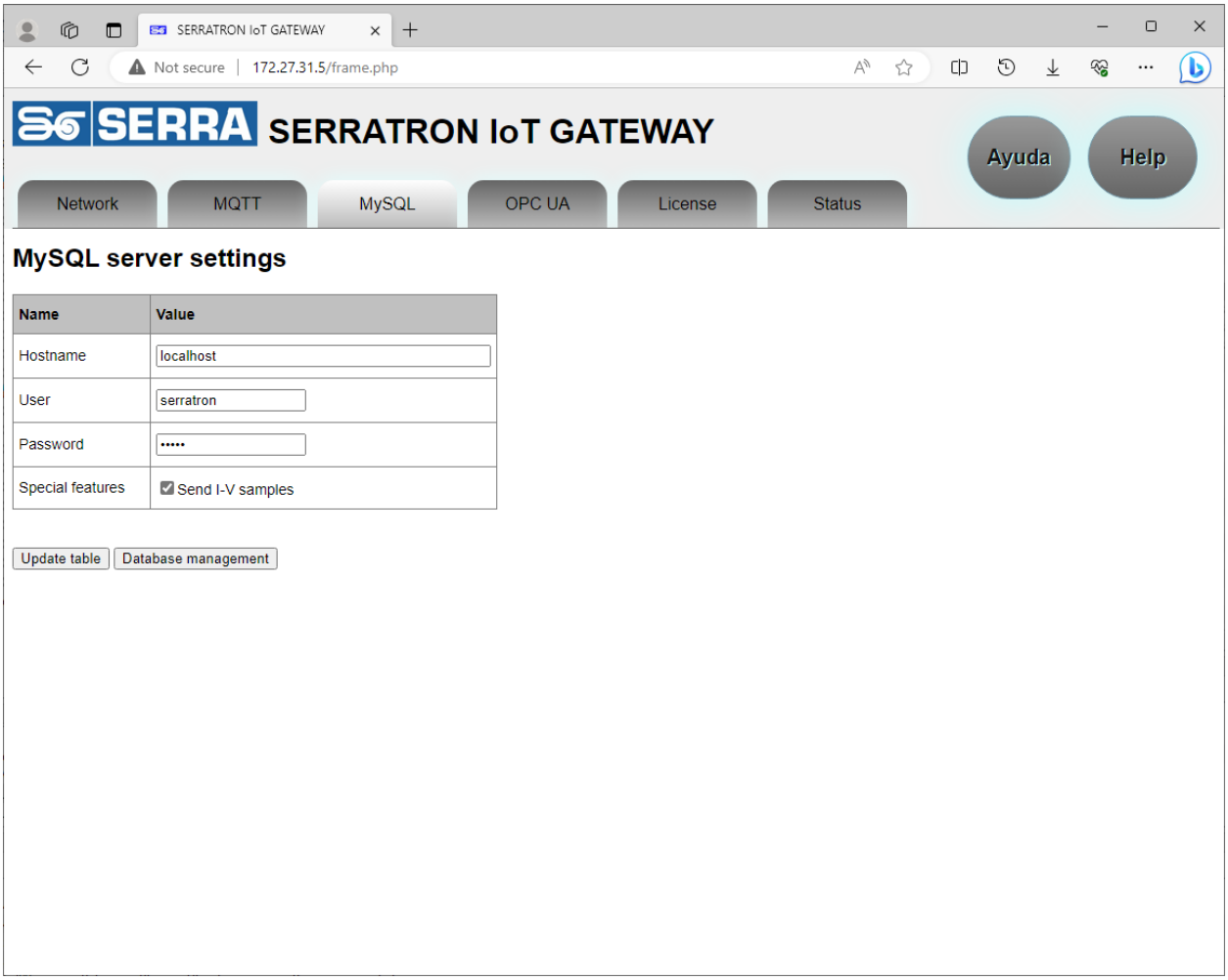
The relational database model is based on first-order logic and set theory. Its fundamental idea is the use of relationships. These relationships could logically be considered as data sets called tuples. Although this is the theory of the relational databases created by Codd, most of the time it is conceptualized in a way that is easier to imagine, thinking of each relationship as if it were a table that is composed of records (each row in the table would be a record or tuple) and columns (also called fields).

Structured query language (SQL) is a domain-specific language used in programming designed to manage and retrieve information from relational database management systems.

The main advantages of MySQL are: Its flexibility and scalability regarding the features of the platform on which it is implemented, its capacity to offer high speed in data manipulation and, at the same time, safeguard the security and referential integrity of the information through a strong control of transactions, its simplicity and ease of learning.

Configuration

If you click on the MySQL tab, you can configure the following parameters:



Their meaning is as follows:

Hostname This is the MYSQL server's address.
Either the server's URL or its IP address.

User Username.
By default, the password system is always active in MySQL. Type an authorised username.

Password Password.
Password corresponding to the above user.

Special features Allows to enable special functions, only of interest to some users.

Send I-V samples Fill in two columns in the **Monitoring** table with the current and voltage samples for each weld. A sample is available every 2 milliseconds.

Several buttons are shown at the bottom of the screen:

Update table	All changes made to the table are updated in the SGateway when this button is pressed. Do not forget to press this button when all changes have been made.
Database management	This button is only shown if the system has a tool to manage the MySQL database. Please refer to section Database manager.

Tables

There is no need to worry about the tables where the data will be stored. SGateway itself will create them the first time it connects to the MySQL server.

All fields in the tables are in English. This prevents errors of interpretation and makes data properly identified anywhere in the world.

The tables created are described below:

Errors

This contains the relevant data of the failures of the welding controllers and the electric welding gun controllers.

Id	*Auto-incremental field to generate a unique identifier*
Timer	*Device name on the **Network** tab*
TError	*Error text*
Error	*Error number*
ErIndex	*Index of the error*
Program	*Program that failed (if applicable)*
Code	*Welding point code*
Duration	*Error duration*
Date	*Error date and time*

Monitoring

This contains the welding relevant data of the welding controllers.

Id	*Auto-incremental field to generate a unique identifier*
Timer	*Device name on the **Network** tab*
Program	*Program used for welding*
KSR	*Current regulation mode*
Pha1u	*Phase used in Weld time 1*
I1u	*Current used in Weld time 1*
Pha1m	*Phase measured in Weld time 1*
I1m	*Current measured in Weld time 1*
Pha2u	*Phase used in Weld time 2*
I2u	*Current used in Weld time 2*
Pha2m	*Phase measured in Weld time 2*
I2m	*Current measured in Weld time 2*
Pha3u	*Phase used in Weld time 3*
I3u	*Current used in Weld time 3*
Pha3m	*Phase measured in Weld time 3*
I3m	*Current measured in Weld time 3*
FPVu	*Force used by the proportional valve*
SpotCnt	*Weld spot counter*
TSpotCnt	*Total weld spot counter*
Error	*Welding error (if applicable)*
AnI1	*Measured value at the analogue input*
Date	*Date and time of welding.*
Current	*Samples with instantaneous currents along the welding*
Voltage	*Samples with instantaneous voltages along the welding*

Maneuvers

This contains the relevant data of the maneuvers of the electric welding gun controllers.

Id	*Auto-incremental field to generate a unique identifier*
Timer	*Device name on the **Network** tab*
Gun	*Number of current gun*
Program	*Activated program*
Code	*Point code*
Error	*Error during the maneuver (if applicable)*
ErIndex	*Index of the error*
Type	*Maneuver type*
ThPl	*Sheet metal thickness*
Fric	*Measured friction*
ElHWe	*Electrode holder wear*
ElWe	*Electrode wear*
ElRWe	*Electrode relative wear*
Counter	*Maneuver counter*
Date	*Date and time of maneuver*

Database manager

The Database manager allows you to review or modify the structures generated by the SGateway.

To facilitate start-up, or for installations with a small number of controls, the SGateway incorporates a MySQL database server.

From the various options, we have chosen **Adminer** because it is open source and meets our needs. As it also allows for SQL queries, it is possible to manage the database at our convenience.

To access the server, these parameters must be programmed:
Hostname: **localhost**
User: **serratron**
Password: **serra**

To register a username and password (e.g. serratron2 and serra2), enter these MySQL commands:

```
CREATE USER 'serratron2'@'%' IDENTIFIED BY 'serrasold2';
GRANT ALL PRIVILEGES ON *.* TO 'serratron2'@'%' WITH GRANT OPTION;
```

•••

Capítulo 6
OPC UA

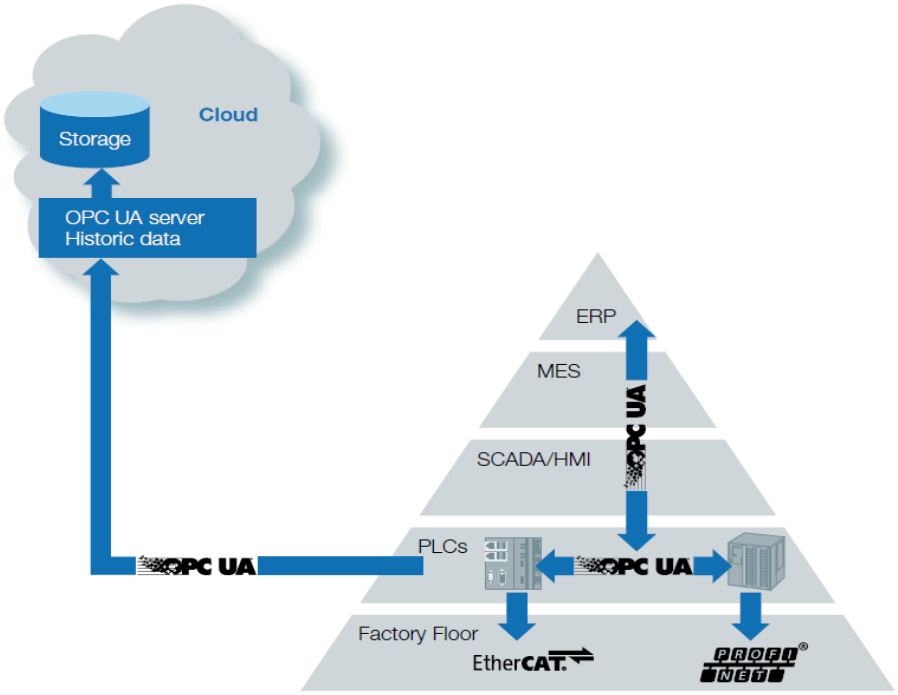
General

OPC Unified Architecture (OPC UA) is a **vendor-independent communication protocol** for industrial automation applications. It is based on the client-server model and enables continuous communication from individual sensors and actuators to the cloud. The protocol is **platform-independent** and has embedded security mechanisms. OPC UA is flexible and totally independent, so it is considered the ideal communication protocol for the implementation of **Industry 4.0**.

OPC UA bridges the gap between the IP-based computing world and the production plant. Interfaces, gateways and the resulting loss of information are a thing of the past because all production process data is transferred via a single protocol, whether within a single machine, from one machine to another, or from one machine to a database in the cloud. OPC UA takes away the need for traditional factory-wide fieldbus systems.

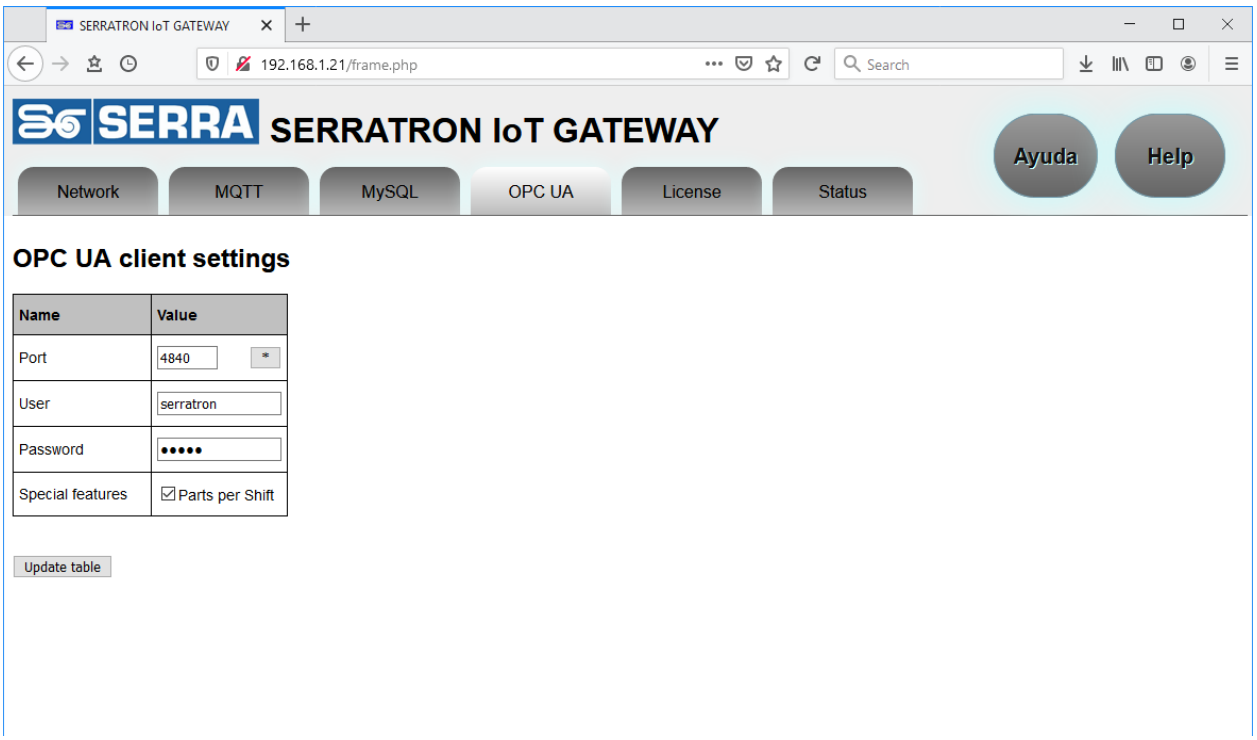
Here are the main features and benefits of OPC UA:

- Neutral platform that works on any operating system
- Ready for the future and to communicate with legacy systems
- Easy setup and maintenance
- Service-oriented technology
- Enhanced visibility
- Increased range of connectivity
- High performance
- Unified access
- Access via firewalls and the Internet



Configuration

If you click on the OPC UA tab, you can configure the following parameters:



Port Communications port.
The OPC UA communications protocol is assigned to port 4840 by default, although it can be changed if, for example, the client has been configured to use a different one.
If you press the * button, the checkbox will be updated by default.

User Username.
If the client is identified with a password, the name of an authorised user must be entered.
If the password system is not used, this field should remain blank.

Password Password.
Password corresponding to the above user. If the password system is not activated, the content of this field is irrelevant.

Special features Allows to enable special functions, only of interest to some users:

- **Parts per Shift** Saves total parts manufactured per shift. See section **SerratronPPSType**.
- **Welding parameters** Muestra en un objeto separado, dentro del objeto Soldadura, los parámetros más relevantes del programa que ha soldado.

This button is shown at the bottom of the screen:

Update table	All changes made to the table are updated in the SGateway when this button is pressed. Do not forget to press this button when all changes have been made.

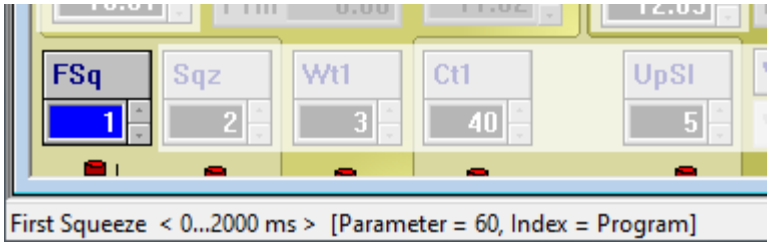
Data structures

The OPC UA Server generates a series of SERRA_TIMER objects based on the devices selected in the Network tab. To view them comfortably you can use the **UA Expert** client. See the corresponding section below.

Each **SERRA_TIMER** object is made up of a series of nodes. The nodes can be of different classes:

- Method
- Objet
- Variable

Variable type objects correspond to parameters of the corresponding device. They present an abbreviation, which is the same one that appears in the CPC-Connect in English. The application CPC-Connect, in the status bar, displays the description and parameter number of the selected box.



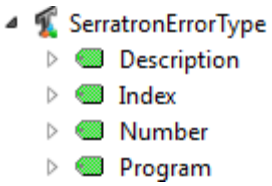
More information about the parameters can be obtained in the device manual, chapter **PARAMETERS**.

Object Types

The generated object types are the following:

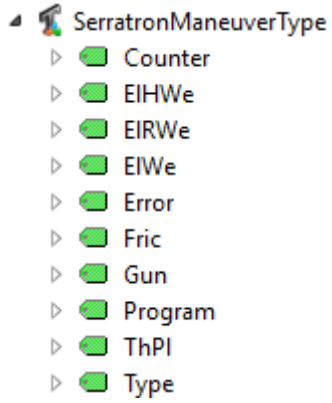
SerratronErrorType

Error type, common to all devices.



SerratronManeuverType

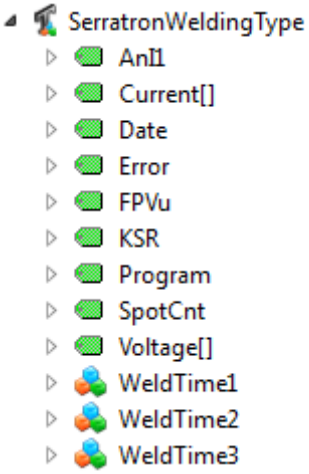
Maneuver type, applicable only to PES-10/20B electric welding gun controllers.



SerratronWeldingType

Welding type, applicable only to welding controllers.

Includes the parameters measured during a welding cycle.

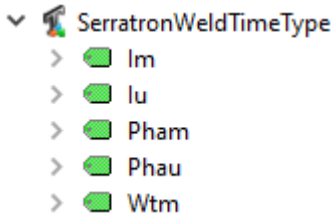


SerratronWeldTimeType

Welding time type.

Includes the parameters measured in a welding time.

Welding type can include up to three welding times.



SerratronWeldParamType

Welding parameters type, not applicable to Serratron 300dp welding timers.

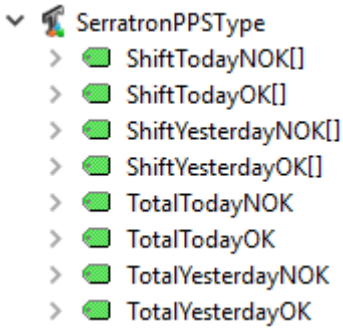
Includes the most relevant parameters of the program that has just welded. The parameter number can be different for different types of welding timers, so it appears in the **Description** attribute of each variable.

This object is optional. For it to appear, the corresponding mark must be selected in the OPC UA tab of the programming interface.

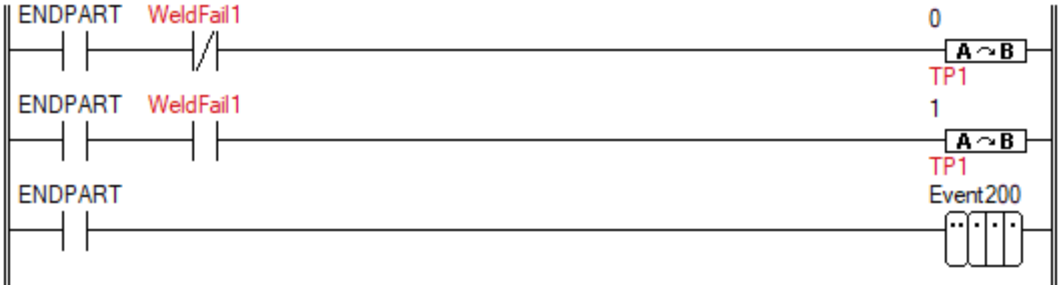
SerratronPPSType

Parts per Shift type, not applicable to Serratron 300dp welding timers.

This object is optional. For it to appear, the corresponding mark must be selected in the OPC UA tab of the programming interface.



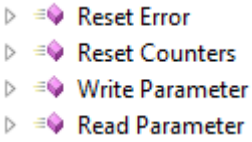
To increase the Piece Per Shift counters, the Serratron PLC program must be modified by adding the following lines:



Every time a piece is finished, the Event200 coil is activated. Memory TP1 indicates whether the part has been good (Value=0) or bad (Value=1).

Methods

Different functions or methods have been incorporated into the basic types. Each type incorporates the functions that are relevant to it.



Reset Error: Reset of device failures.

Reset Counters: Reset all counters

Write Parameter: Writes a parameter value to the device. As input parameters, you need to specify the Program number, the Function number and the Value to be written.

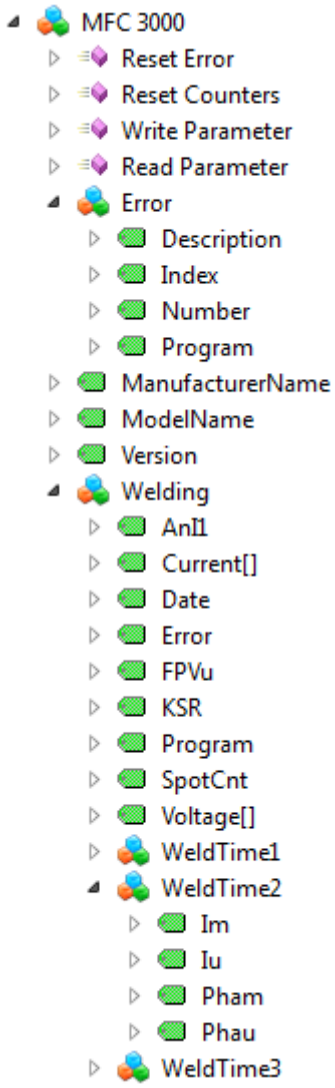
Read parameter: Returns a parameter value from the device. As input parameters you need to specify the Program number and the Function number. As output parameter, it returns the requested value.

Objects

Based on these types, the following basic objects are generated:

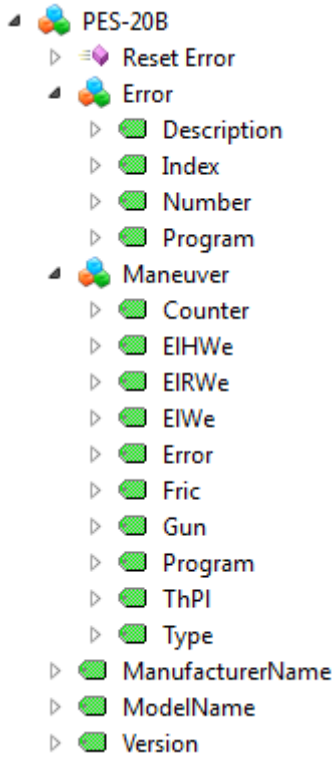
SerraBaseTimerType

Serratron welding control object



SerraBaseGunType

PES-20B electric welding gun controller object.



UA Expert

UaExpert® is a program designed as a general purpose test client that supports OPC UA features such as Data Access, Alarms and Method calls.

It is very useful to verify the correct operation of the server in the SGateway.

UaExpert is available for Windows and Linux and there is a free access version.

The main screen is composed of several frames with a variety of information. The most commonly used of these are:

- **Project**
  To connect to a specific server. In our case, we'll connect to:
  Endpoint URL: **opc.tcp://172.27.32.15:4840**
- **Address Space**
  This shows the list of nodes or devices offered by SGateway, as well as the objects, variables and functions available for each of them.
- **Data Access View**
  This allows us to see the value of one or more variables in real time.
- **Attributes**
  This shows the attributes of the selected node.