proop

User Manual

EMKO

Table Of Contents...

A) Operator Panel

EMKO PROOP provides high speed vector based graphics with powerful Cortex A series CPU. Proop Builder software has user friendly design for rapid and easy development.

A.1. Features

A.1.1. Features Graph And Design

- More than 100 ready to use vector-based elements.
- Vector based image (SVG) support.
- BMP, GIF, JPG, JPEG, PNG, PBM, PGM, PPM, TIFF, XBM, XPM image format support.
- Improved graphics engine; Antialiasing, alphablending support

A.1.2. Support Free type and Windows® Font

Supports TrueType (TTF), PostScript Type1 (PFA/PFB), Bitmap Distribution Format (BDF), CID-keyed Type1, Compact Font Format (CFF), OpenType fonts, SFNT-based bitmap fonts, Portable Compiled Format (PCF), Microsoft Windows Font File Format (Windows FNT), Portable Font Resource (PFR), Type 42 (limited support) font types.

A.1.3. Support Remote Access

Remote control can performed by the internal VNC protocol.

A.1.4. Project Upload/Download Via USB Host

Project upload or download can do in a short time by the high speed data transfer USB 2.0 Port

A.1.5. Regional Formatting Format Support

The time, date, and number formats are sensitive to regional settings.

A.1.6. User Friendly EMKO Macro

Emko Macro is designed to perform custom control functions and calculations with internal I/O and communication devices.

Macro is described under the heading 'Macro'.

A.1.7. Internal Analog/Digital IO Port Support

The user can control the data with the macro and visual elements.

A.1.8. Online/Offline Simulation Mode

Compiled program is simulated in the PC environment without PROOP device.

A.1.9. Industry Standard Multiple Communication Network

- Communication interface: RS232, RS422, RS485, Ethernet
- Communication protocols: MODBUS ASCII, RTU, TCP/IP.
- Siemens S7-200/300/400/1200 PLC protocol support.
- Supported PLC protocols: Siemens PPI, MPI, ISO over TCP

A.2. PROOP Builder Setup

Minimum system requirements for Proop Builder Software install:

- 1GHz or greater CPU
- 1GB RAM
- 2GB Hard Disk (least 500 MB of free memory)
- RJ45 Ethernet Network Cable
- USB 1.1 Port Input
- Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10 operating systems.

Please, follow the steps on the below for installation.

Step 1:

It is strongly recommended that before proceeding, you ensure that no other Windows programs are running.

Step 2:

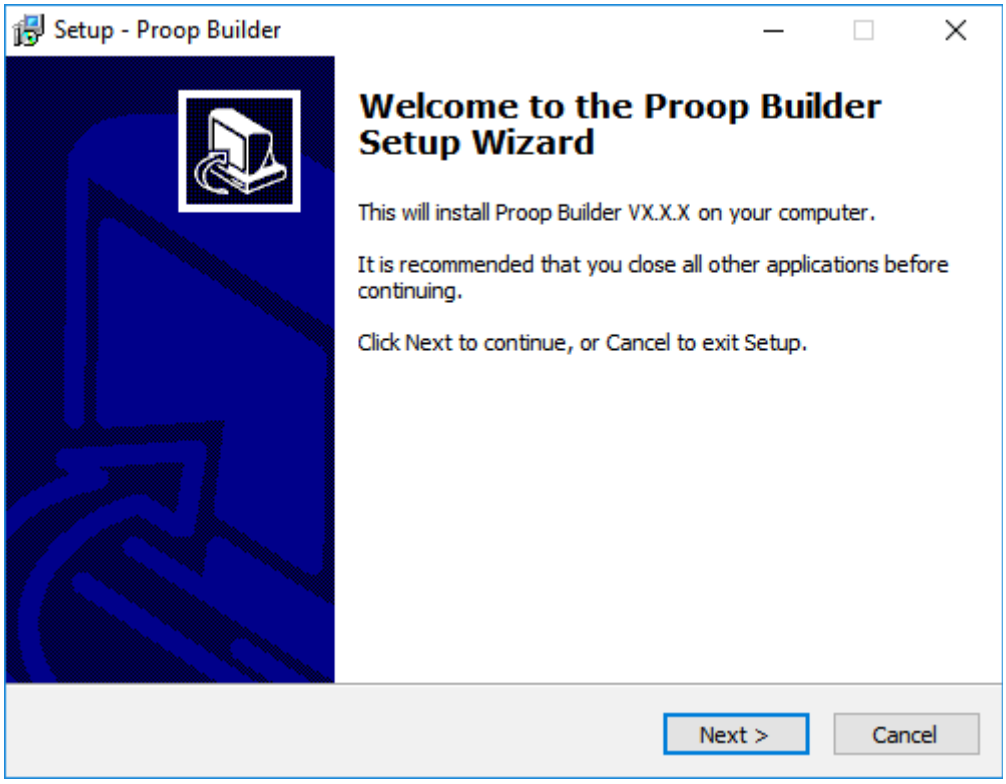
Run the Proop Builder setup file “Proop Builder VX.X.X Setup.exe” to start the installation process.

Step 3:

Continue the installation by following the dialog boxes on the screen and choose where to install.

After selecting the default folder, click '***Next'***. If necessary, you can retrieve individual steps with '***Back'*** option.

Program will automatically be installed in the default folder.



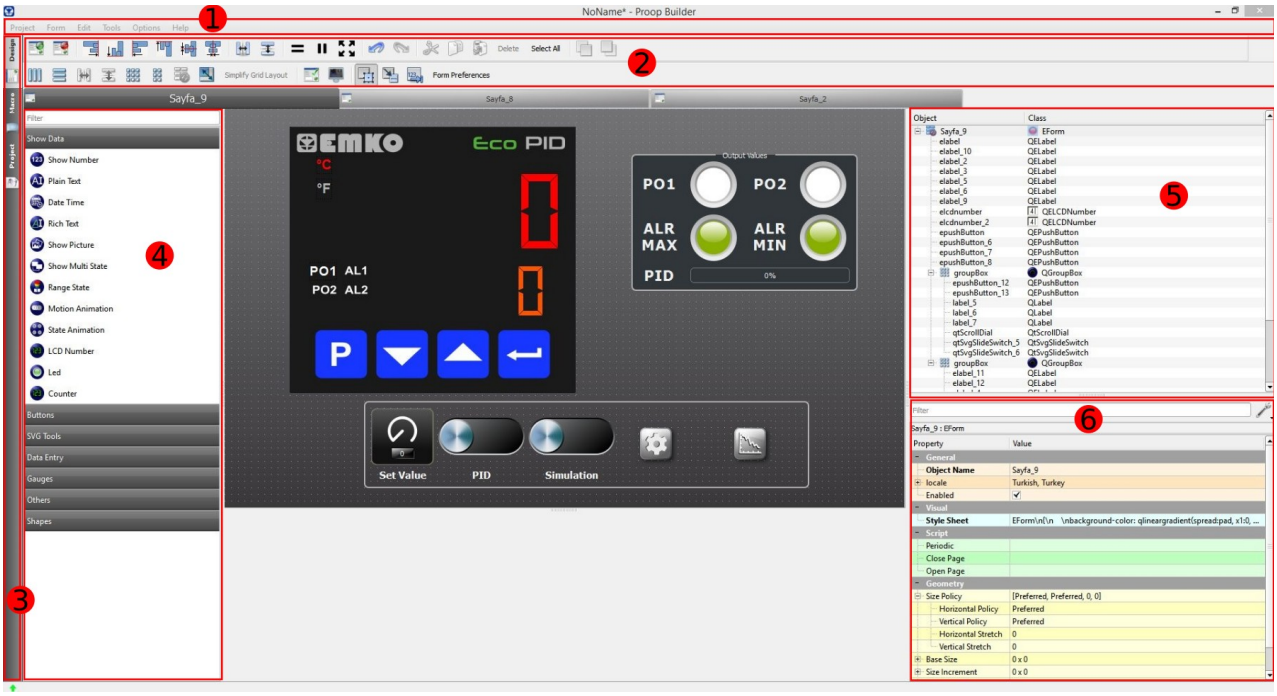
Picture 1: Setup

Step 4:

Please click the "Windows Start>Programs>Proop Builder" shortcut to start the application.

B) Screen Editor

Editor contains six sections; tool and sidebar, elements and property list, element tree.



Picture 2: Screen Editor

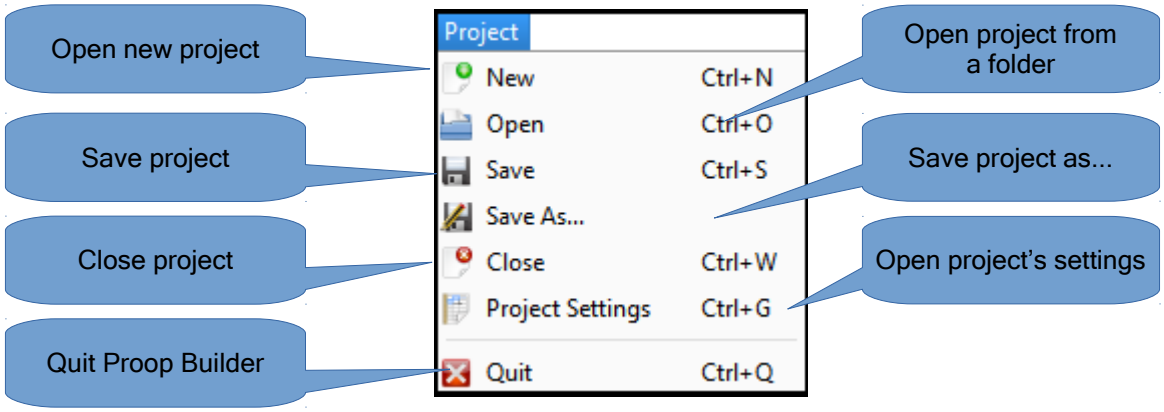
B.1. Menu Bar

The menu bar contains Project, Form, Edit, Tools, Options and Help sections as the picture below.

Picture 3: Menu Bar

B.1.1. Project

Menu with editing options related to the project. There are sub menus as below.

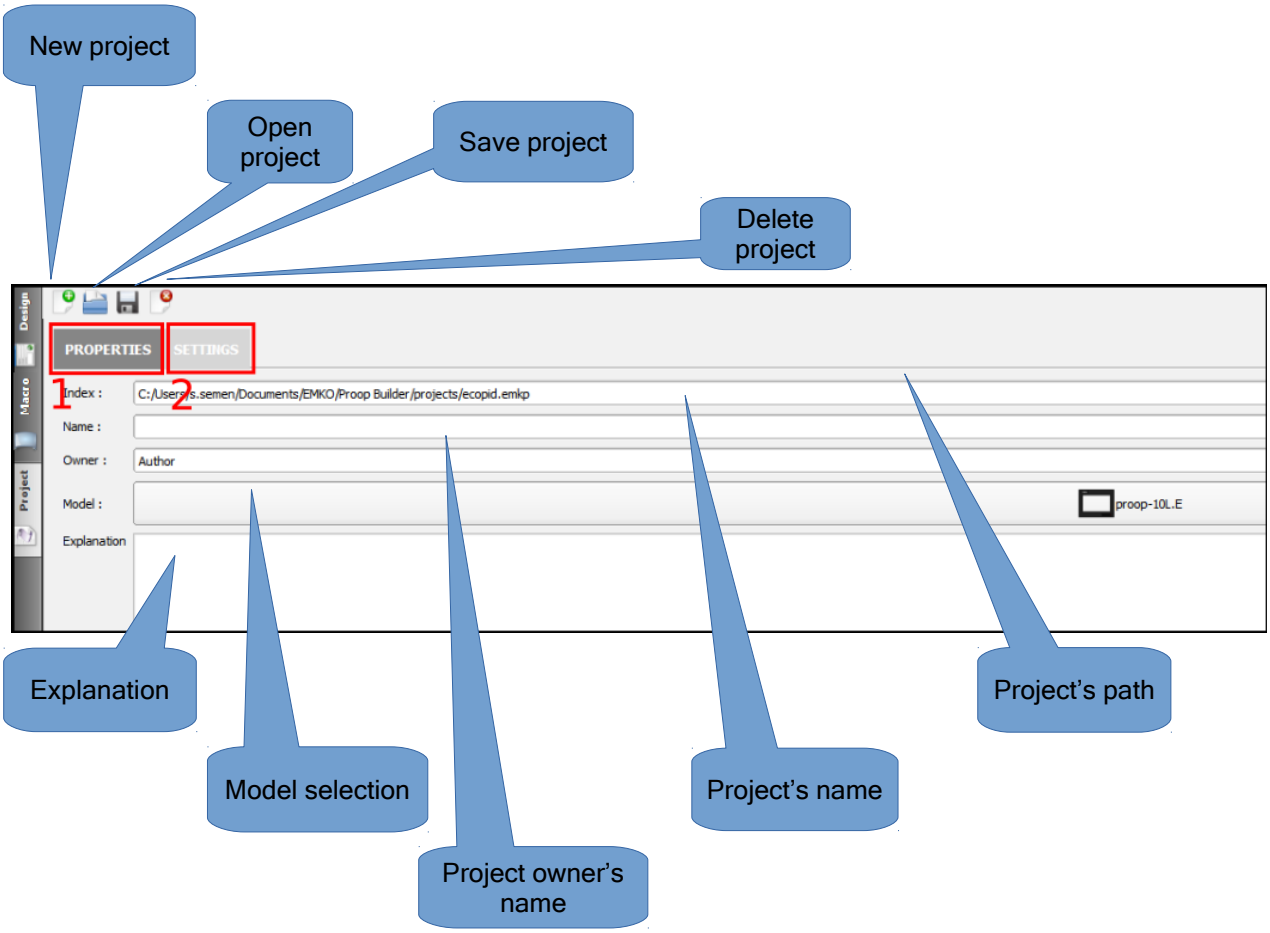


Picture 4: Project

B.1.1.1. Project Settings

Project properties and settings that are active on this page are edited.

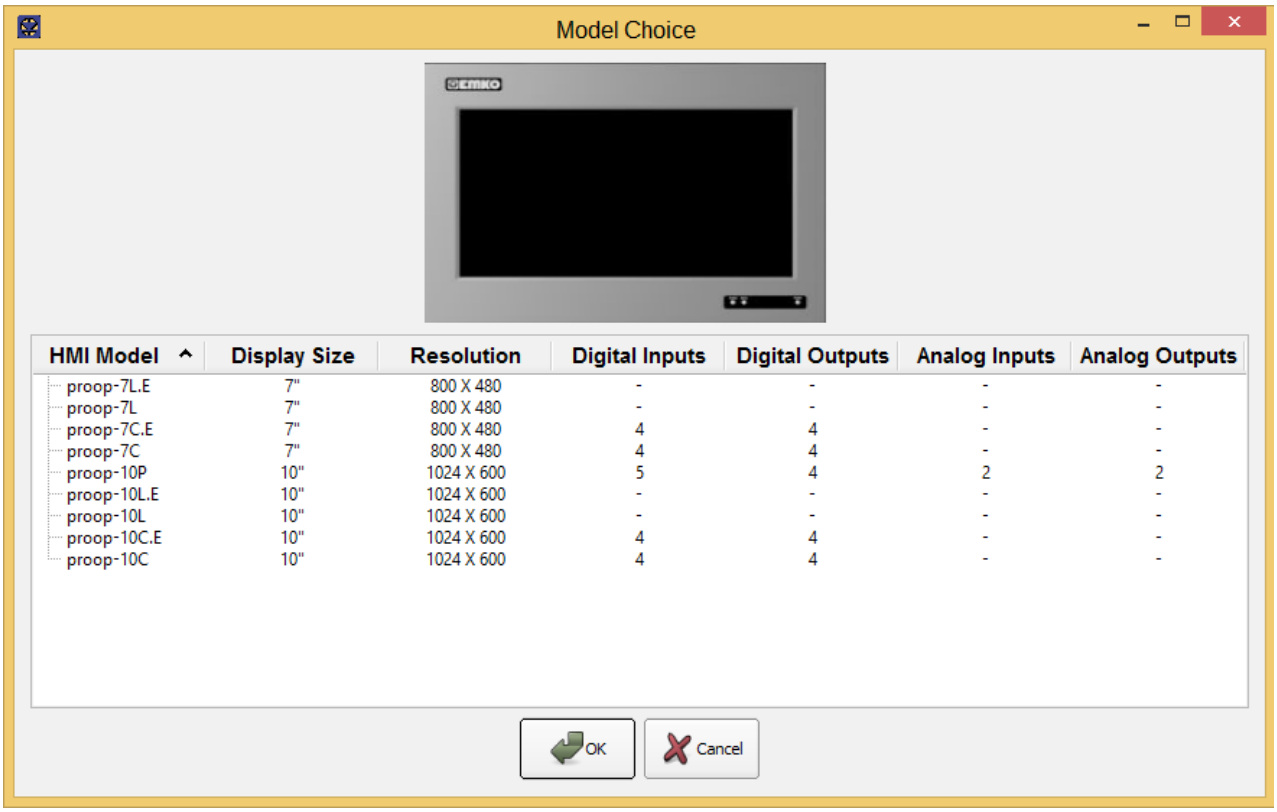
The properties and settings of the active project are edited.



Picture 5: Project Settings

When you click on ***"Properties"*** sections in the field 1, form screen appears. In the picture above contains field descriptions.

Model Selection: Lists all Proop models with specifications and use to select the target model.



Picture 6: Model Selection

Click on the ***"Settings"*** section in the number 2 and the settings form screen appears. Here, frequency of macro work in the project is arranged.

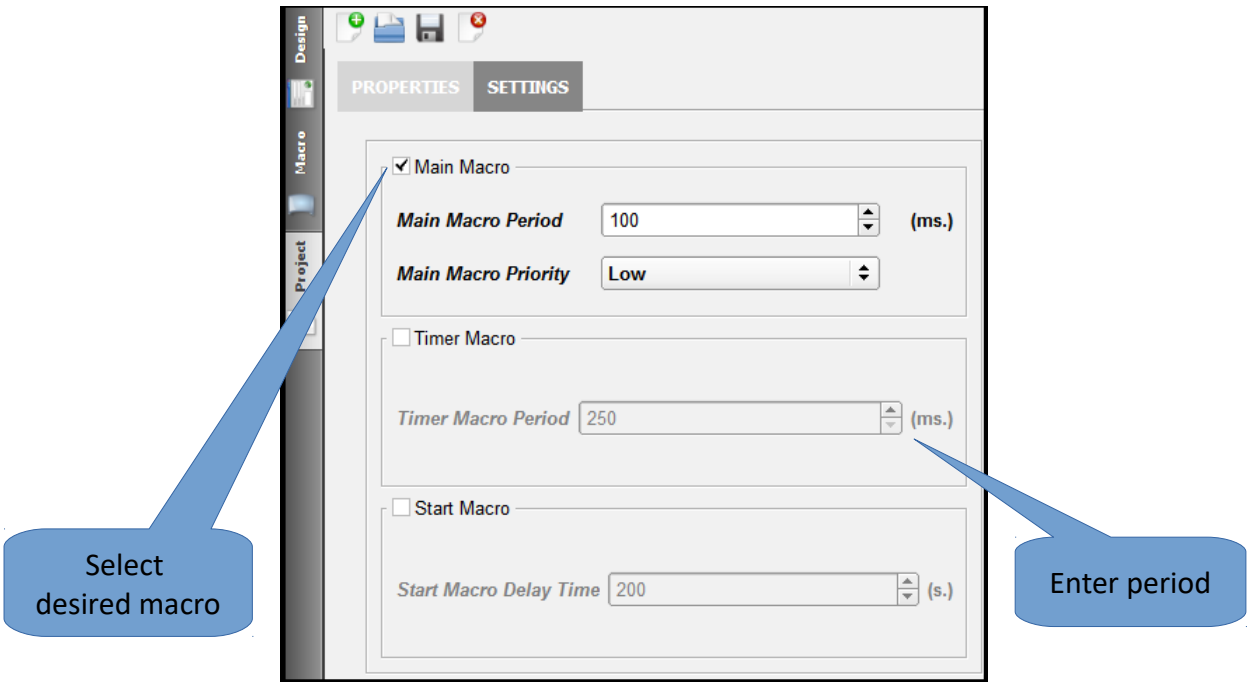
Main Macro: A master macro is written for a project, and this macro is continuous with the specified period.

Timer Macro: When the program starts to run, the timer macro runs.
Macro, runs continuously at the specified period. A timer macro is written for a project.

Beginning Macro: Runs once when the project is opened.

The useage of macro is described under the heading **"Macro"** title.

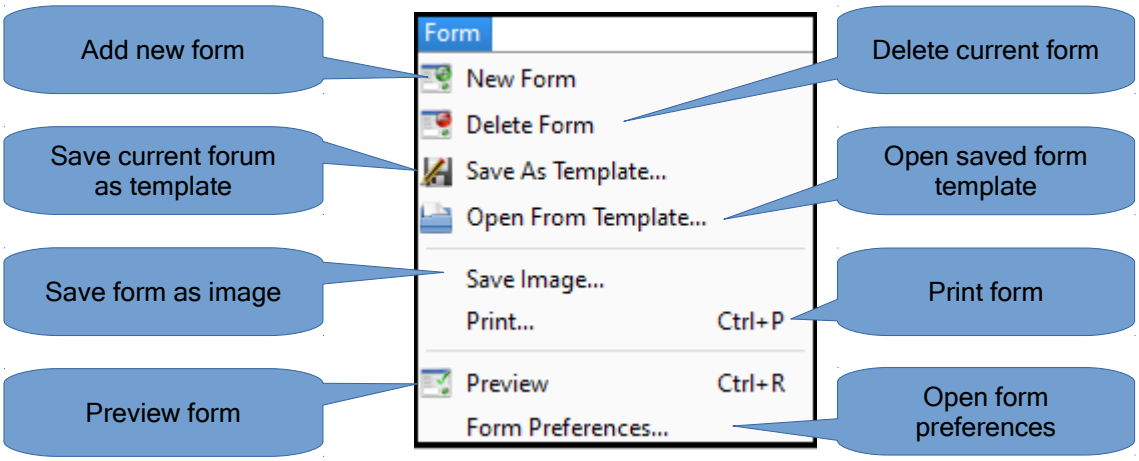
The desired macro is selected and edited as shown below.



Picture 7: Project->Settings

B.1.2. Form

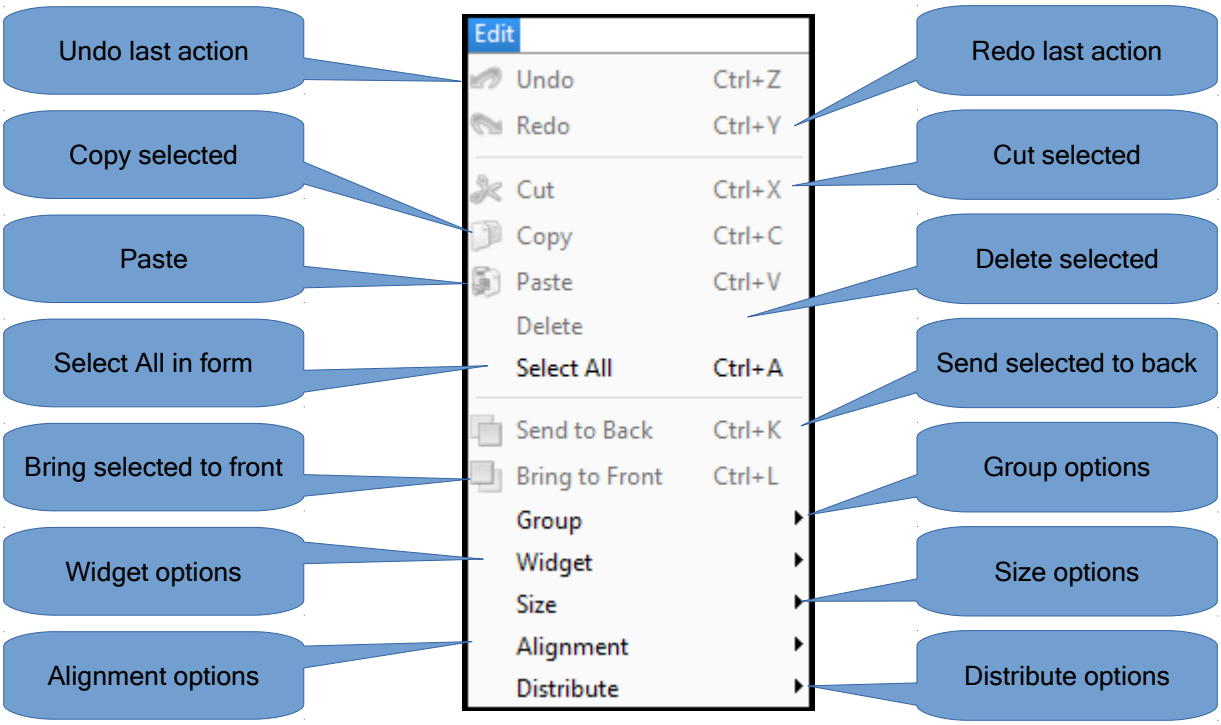
A menu has with options for the form. There are sub menus as below.



Picture 8: Menu Bar->Form

B.1.3. Edit

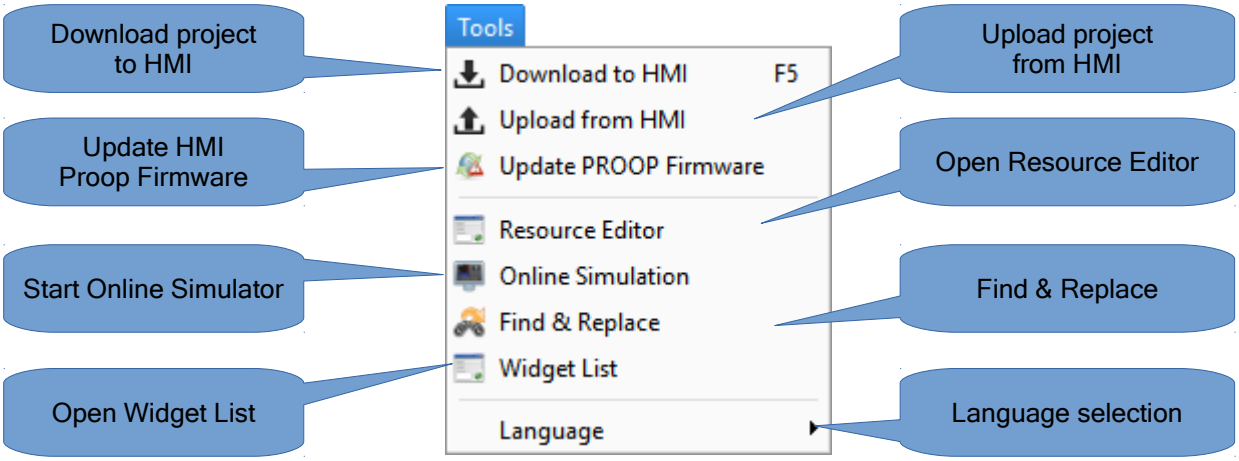
Contains regulations about the elements in the form.



Picture 9: Menu Bar->Edit

B.1.4. Tools

Contains general tools related to the project. There are sub menus as below.



Picture 10: Menu Bar->Tools

B.1.4.1. Uploads

To upload the project files you can use USB Cable or USB disc.

Project Upload Via Port

- To upload the project to the device, plug the USB cable into the device.
- Click to '***Tools Bar>Download'*** or press ***'F5'*** on keyboard.
- Click on the icon in the bottom left of the Proop Builder Program. 

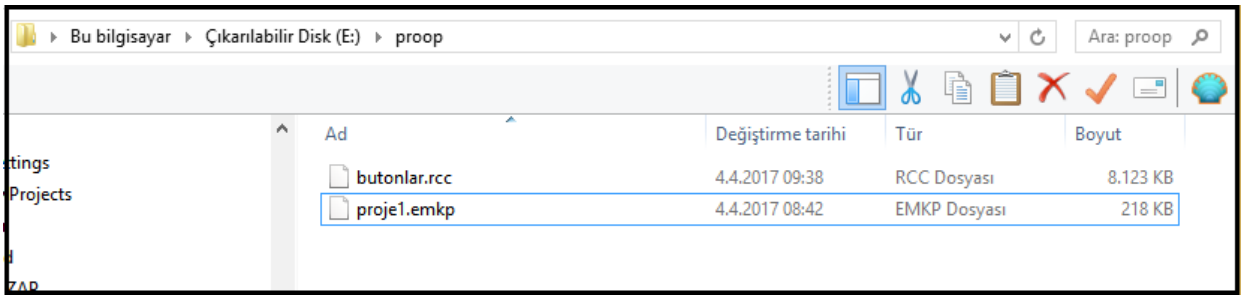
Uploading Project with USB Memory

- To upload the project into the device, create a folder named 'emko' or 'proop' in your USB memory

  Example upload folder: “G:\proop”, “G:\emko”
- Copy the project file (*.emkp) into the upload folder.
- If your project contains resource files please copy the compiled resource files(*.rcc) to the upload folder. You can find the compiled files near your resource library.

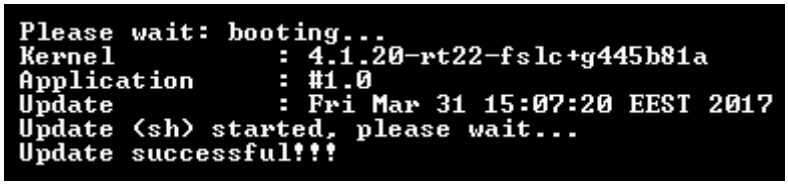
Picture 11: Proop Builder Folder

- The files that should be located in the folder named Proop are as follows .



Picture 12:Proop Folder

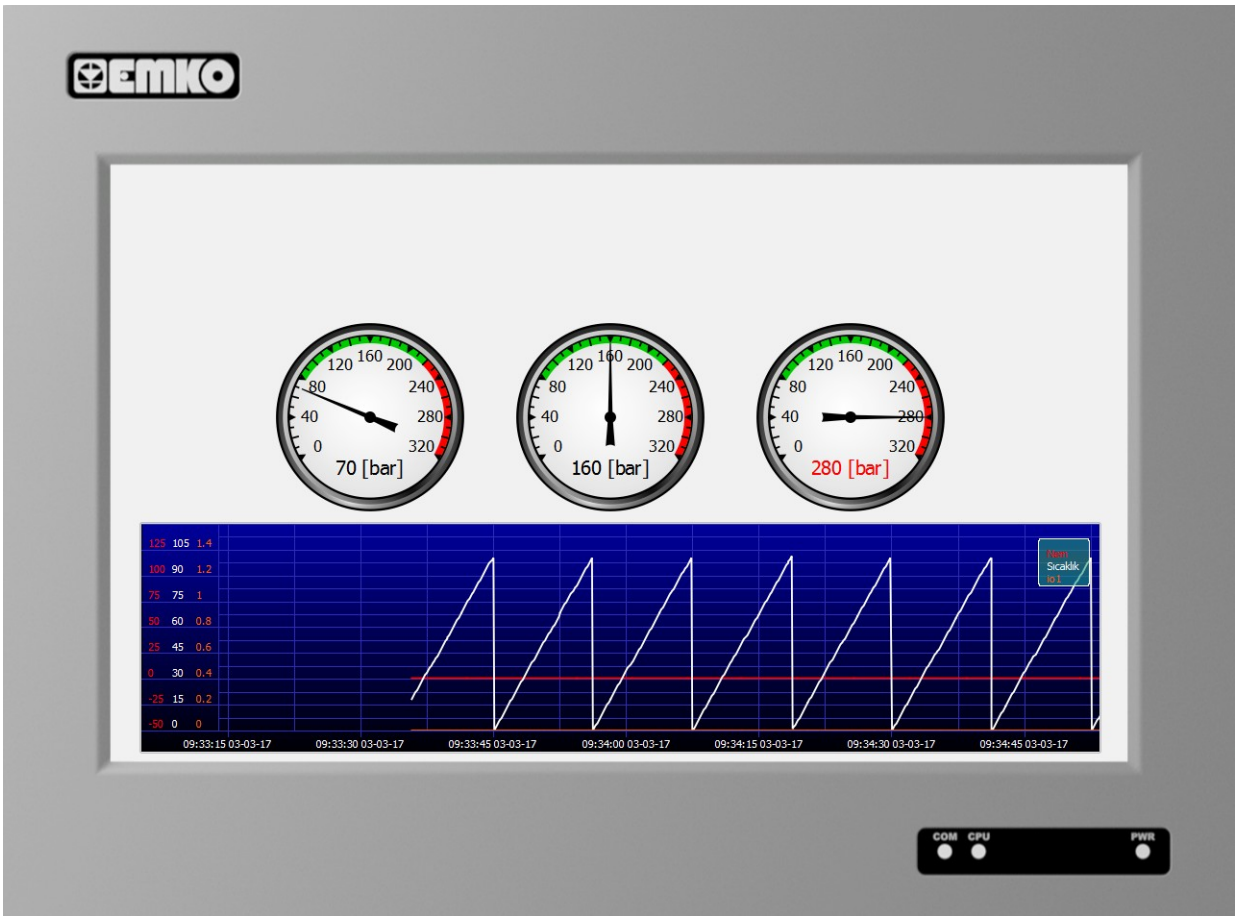
- Unplug the USB after copying to USB memory is finished.
- Plug it into the USB port on the back of the device.
- When you switch off the power and switch again, you can follow the project installation status via the device screen.



Picture 13:Project Upload

B.1.4.2. Online Simulation

Designed pages and macro codes can be simulated in the PC environment.



Picture 14:Menu Bar->Tools->Online Simulation

When you click the right mouse button the pop-up screen appears.

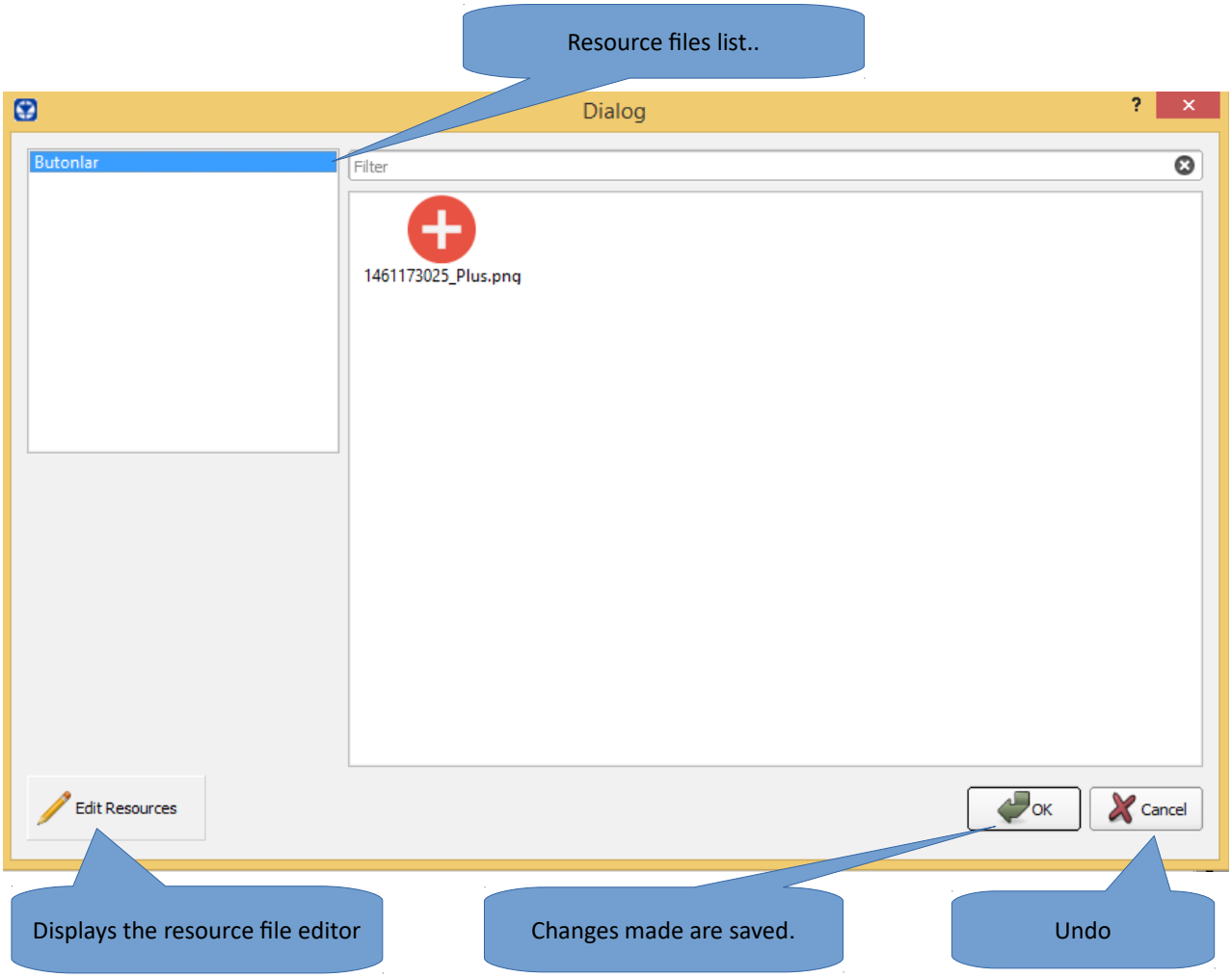
You can navigate between pages and finish the simulation with these shortcuts.

Picture 15:Menu Bar->Tools->Online Simulation->Options

B.1.4.3. Image & Font Library

Used in the program Picture, animations and font types listed here.

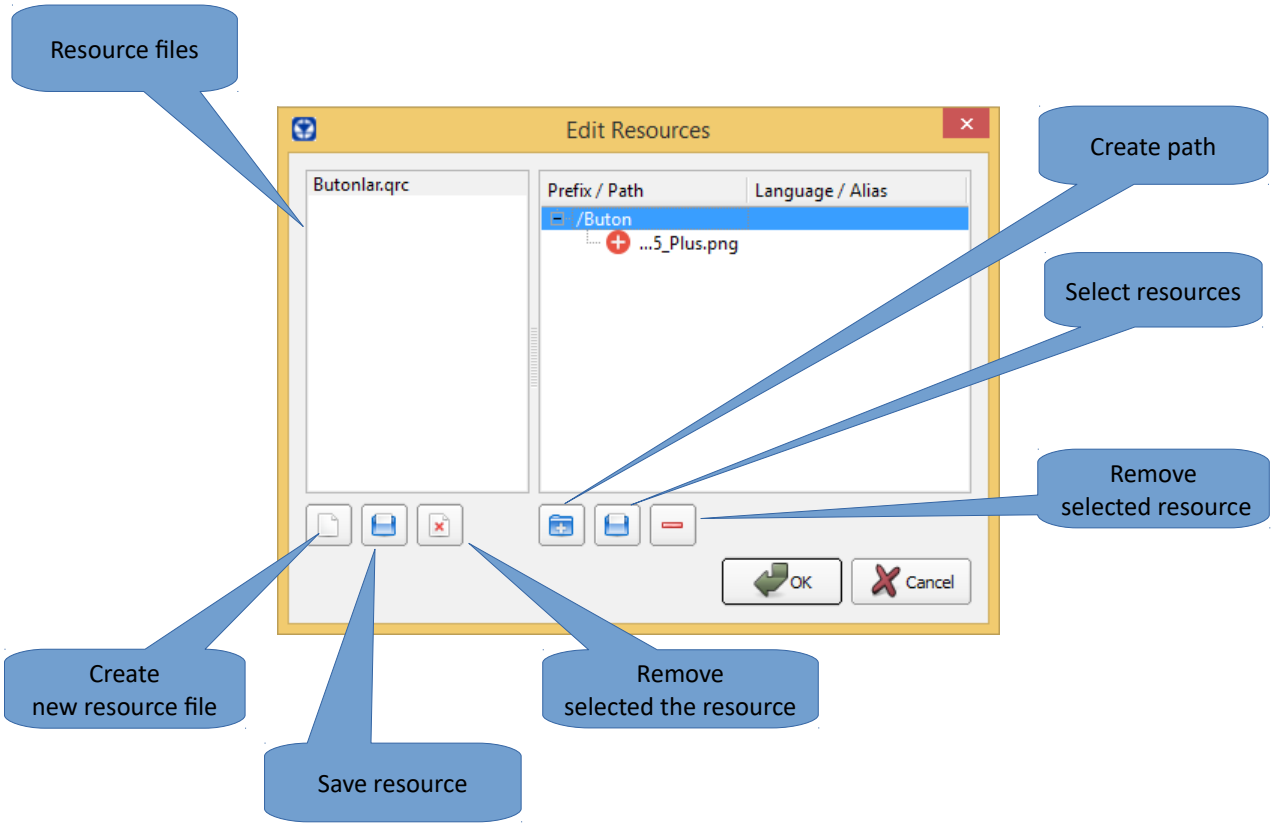
User can be create, edit or delete custom image library and user can load and use the font type that it wants to use in the project in the picture & font library.



***Picture 16:*Resource Editor**

Click the ***'Edit Resource'*** button to edit the image files.

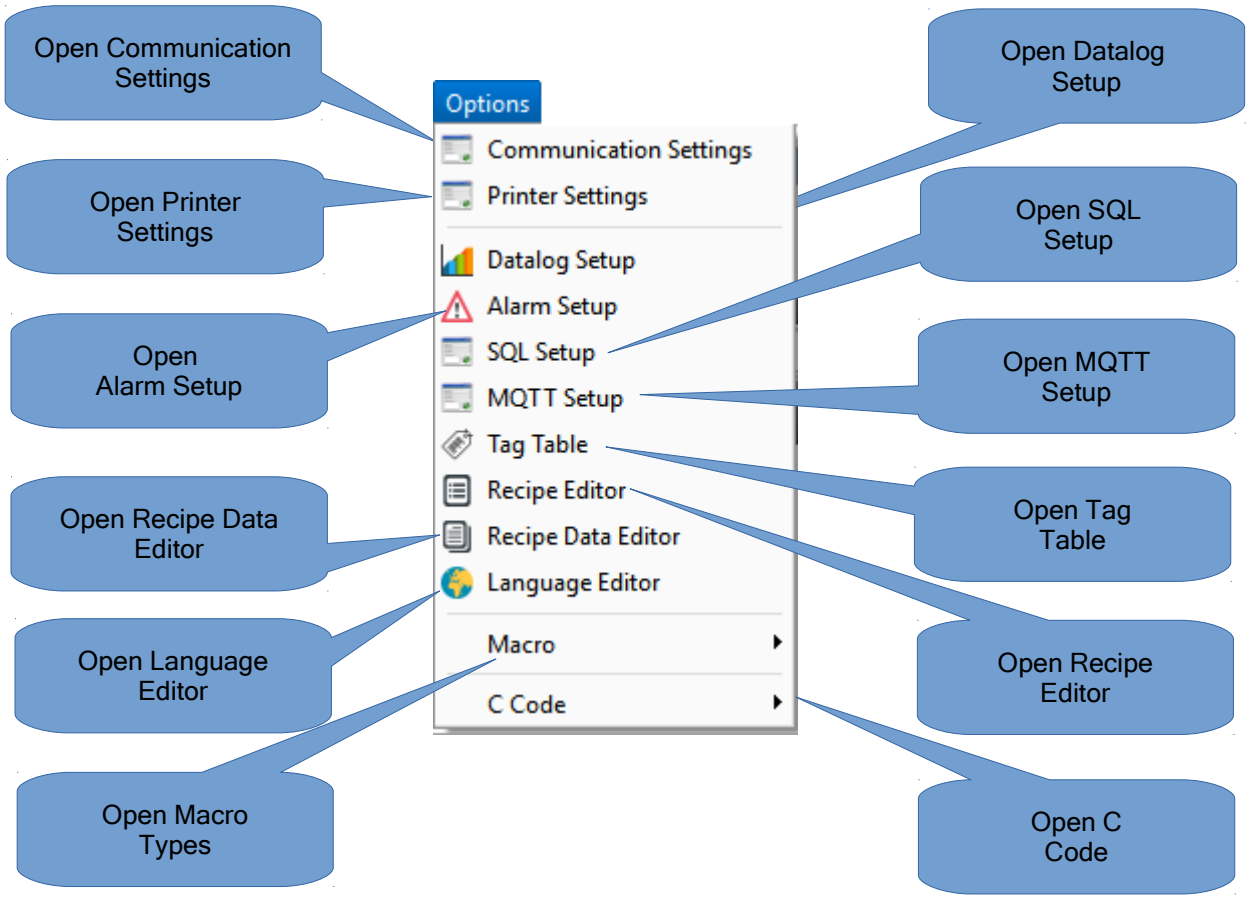
The created or existing image files are managed from here.



Picture 17: Menu Bar->Edit->Resource Editor

B.1.5. Options

Contains a project options. There are sub menus as below



Picture 18: Options

B.1.5.1. Communication Settings

Window contains communication settings of HMI connected device.

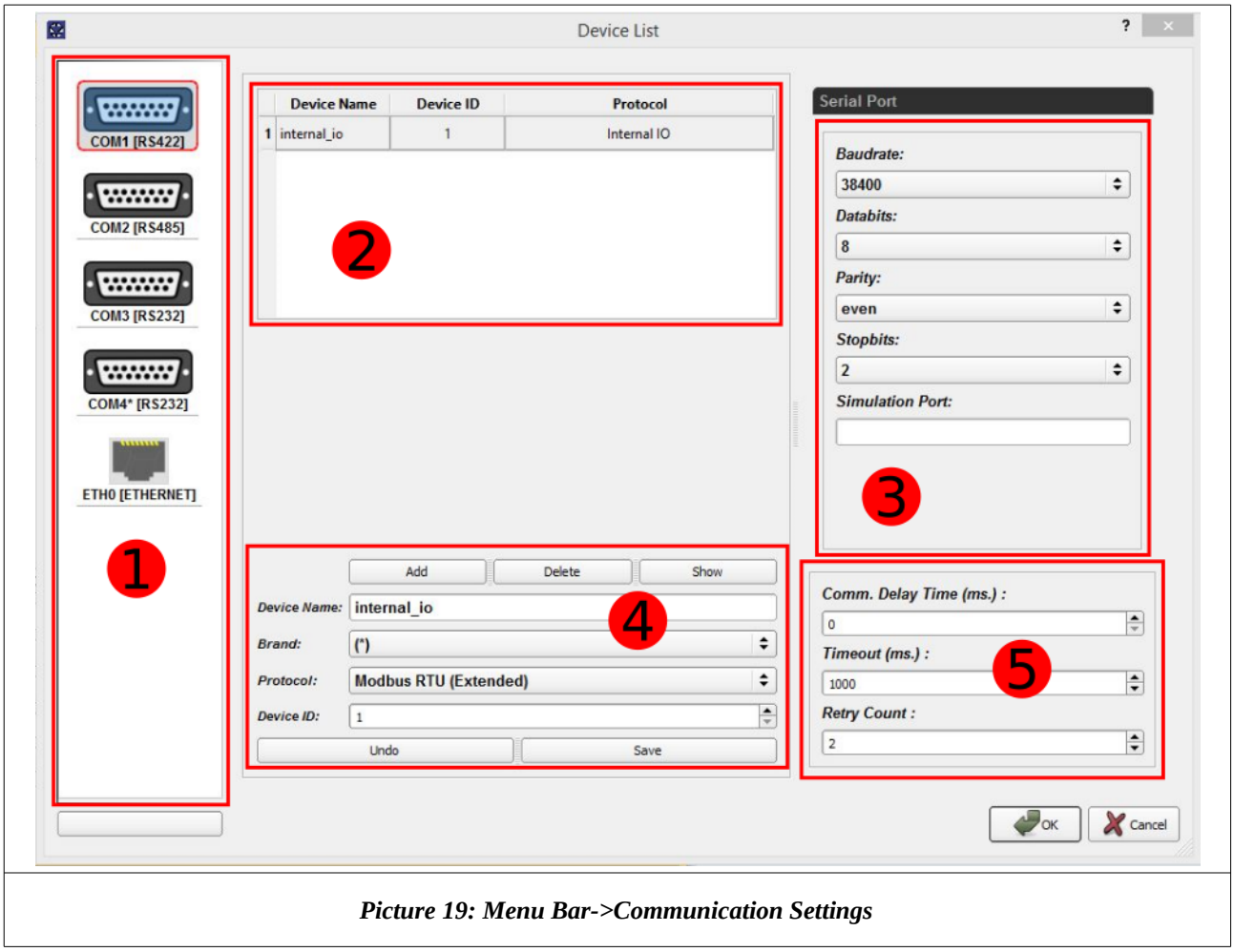
The field number 1; Selected COM port.

The field number 2; Lists added devices into the selected COM port.

The field number 3, Selected COM port communication settings. The simulation port field in the serial settings specifies the PC comport to be used during online simulation.

The field number 4, contains device information fields with modification.

The field number 5, contains additional options for connection.



Picture 19: Menu Bar->Communication Settings

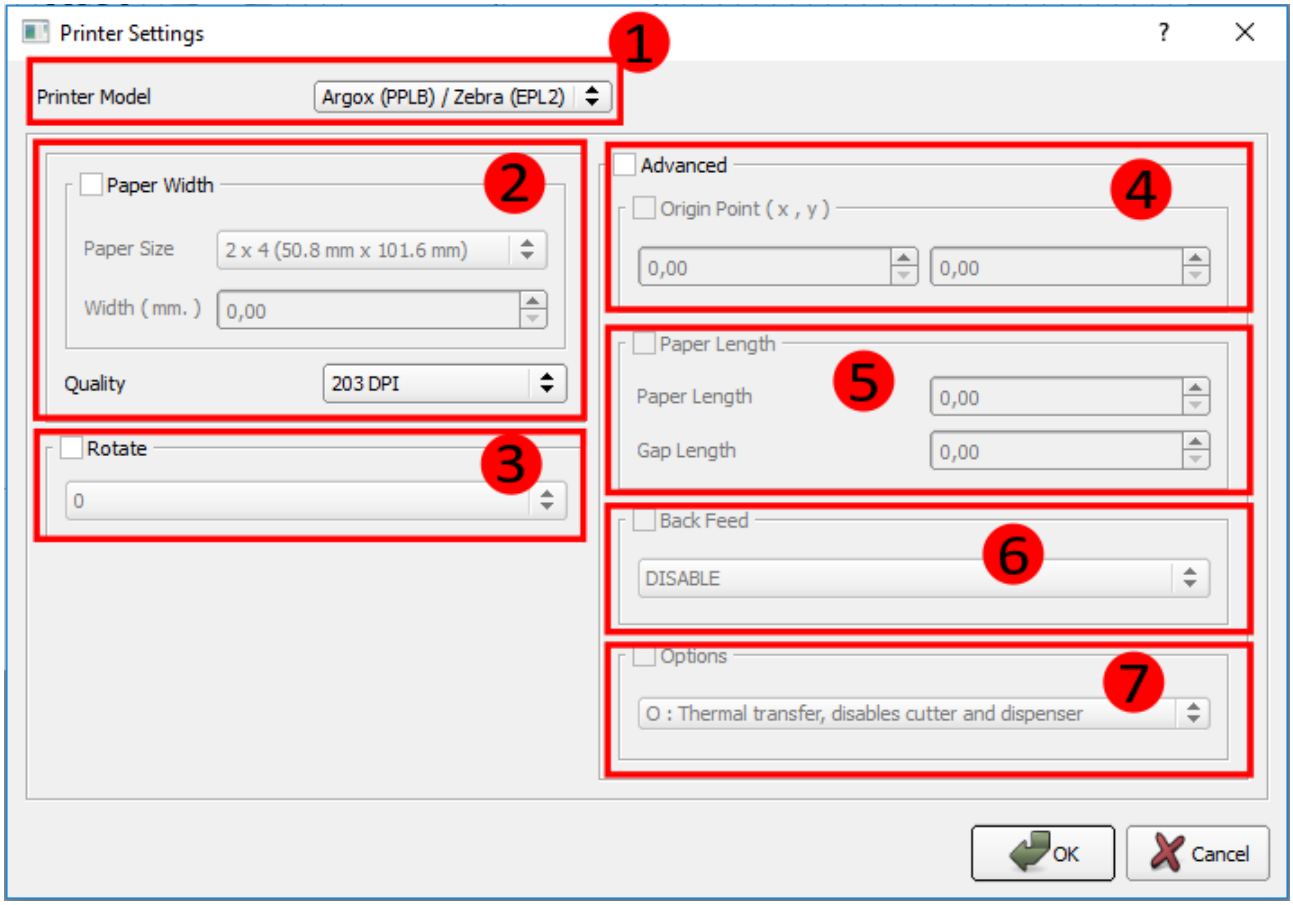
To Add a Device;

- Select the connected point of the device from area 1
- Enter the device information field 4 and click the add button.
- Lists the added device in the second area device list and select the device.
- Arrange the serial port settings in area 3.
- Finally, Enter the communication delay time from the 5th area.
- Click the save button, after making changes to the devices in the device list.

B.1.5.2. Barcode Printer / Reader

B.1.5.2.1. Printer Settings

Barcode printers are working with all brands of PPLB and EPL2 languages are supported. (Argox and Zebra).



Picture 20: Menu Bar → Options → Printer Settings

To Add a Printer;

Select printer model from area 1.

Select page width settings from area 2.

Select angle rotation from area 3. Used as the rotation angle of the text to be printed.

Starting point coordinates are entered from area 4. Starting point of X and Y coordinates.

Select label and space length from area 5.

Select feedback property from area 6.

- *Enable -> Feedback property is active.*
- *Disable -> Feedback property is passive.*

If you have a printer type with feedback, you should select enable for to activate this feature.

Select printing options from area 7.

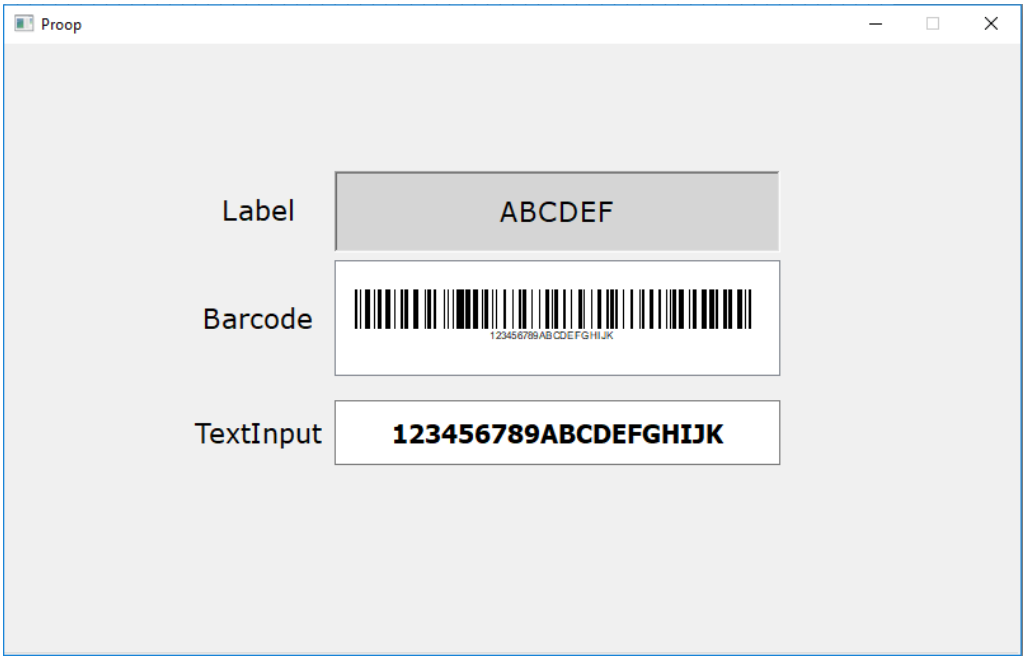
B.1.5.2.2. Reading Barcode

All barcode readers connected via USB are supported.

Example -1: Barcode Trigger

This example describes how to convert the value entered via the keyboard to the barcode text.

- **Step 1:** Elements used: Label, Barcode, TextInput



Picture 21: Barcode Reading Example

- **Step 2 :** Read address, write address and character length are entered.

Element	Character Length	Read Address	Write Address
	20	-	internal_memory@$S4
	20	-	internal_memory@$M0
	20	internal_memory@$M0	-

All elements must be entered in '**Character Length**'. If the character length is not entered, the barcode text will not appear on the screen. In this example, the character length is set to '20'. The text on the Barcode is kept to be 2 characters in the address; therefore, for example, a text of 20 characters is kept at 10 addresses from the lastbarcode address, the following table describes the sample.

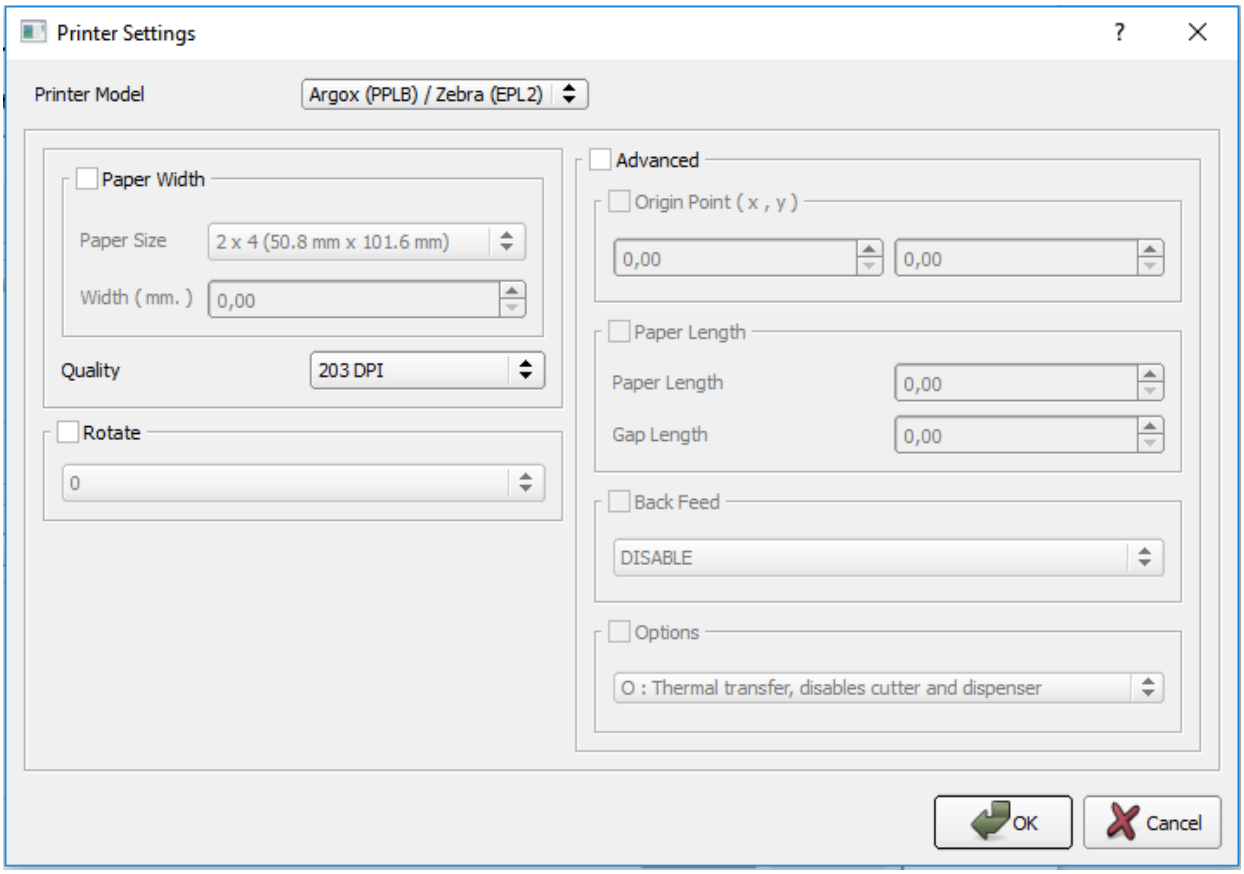
Address	TextInput Value Entered	Barcode Text
$M0 = $S4	"123456789ABCDEFGHIJK"	"12"
$M1		"34"
$M2		"56"
$M3		"78"
. . .		. . .
$M9		"JK"

- **Step 3:** The $S4 address shows the last barcode value. The last incoming barcode value is compared with the internal temporary memory address and, if it is different, is synchronized and assigned to the non-volatile memory. The following table shows the macro codes.

Makro Name	Executed Macro Code	
Startup Macro	`global g_var1;`	
	`func main()`	// main function
	`local loc1;`	
	`$0 = "";`	//$0 address initial value to null.
	`endf`	//end function
	`endp`	//end program
Main Macro	`global g_var1;`	
	`func main()`	// main function
	`local loc1;`	
	`if $0 != $S4`	// LastBarcode ($S4) $0 'a if not equal // $0 is assigned in the starting macro.
	`  $0 = $S4;`	
	`  $M0 = $S4;`	//The $ M0 address is assigned to the permanent memory of the entered LastBarcode text.
	endif;	// end if condition
	endf	//end function
	endp	// end program

Example -2: Writing Barcode

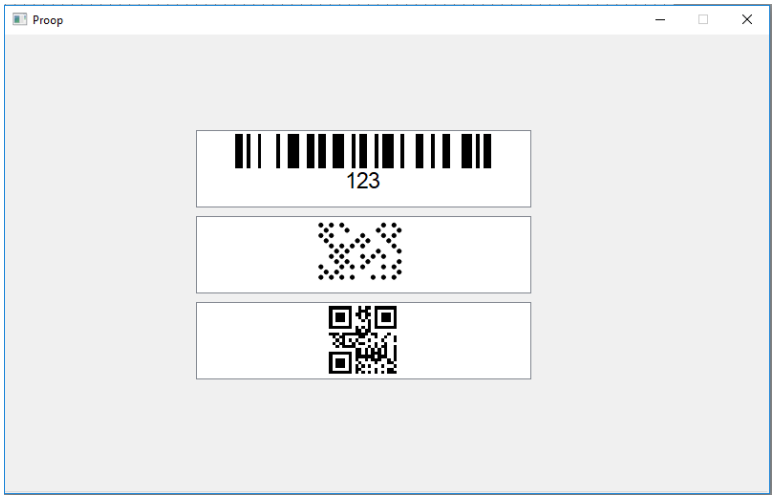
- **Step 1: Setting the Barcode Printer**



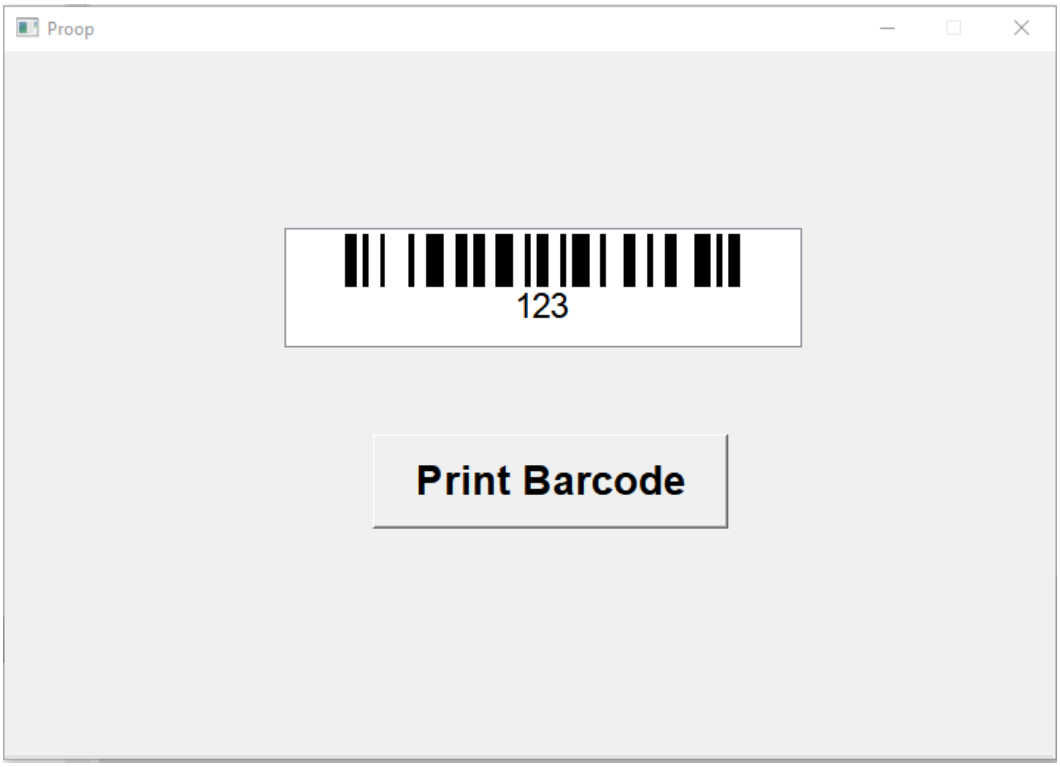
Select printer model, after the required properties are set, “OK” is pressed.

- **Step 2: Use of the Barcode Element**

With the Barcode element, you can create different types of barcodes.(Ex: CODE128,EXCODE,QRCODE.. vb) . The following illustration shows example barcode types.



- **Step 3: Writing Barcode**



Picture 22: Writing Barcode

The macro code to print Barcode is seen below.

printobject() writing barcode function.

printobject(“form_name”,”barcode_object_name”) it is defined as.

Button Used	Execute Macro Code
Print Barcode released button	1 func main() 2 local loc1; 3 printobject("Form_1","ebarcode"); 4 endf 5 endp

B.1.5.3. Datalog Setup

Data is read from given address and saved to csv file. Setup about this operation can be configured in Datalog Setup.

“Channel Name” is given name to log.

“Storage Type” shows where logs are saved.

“Group Name” is given name to log in file.

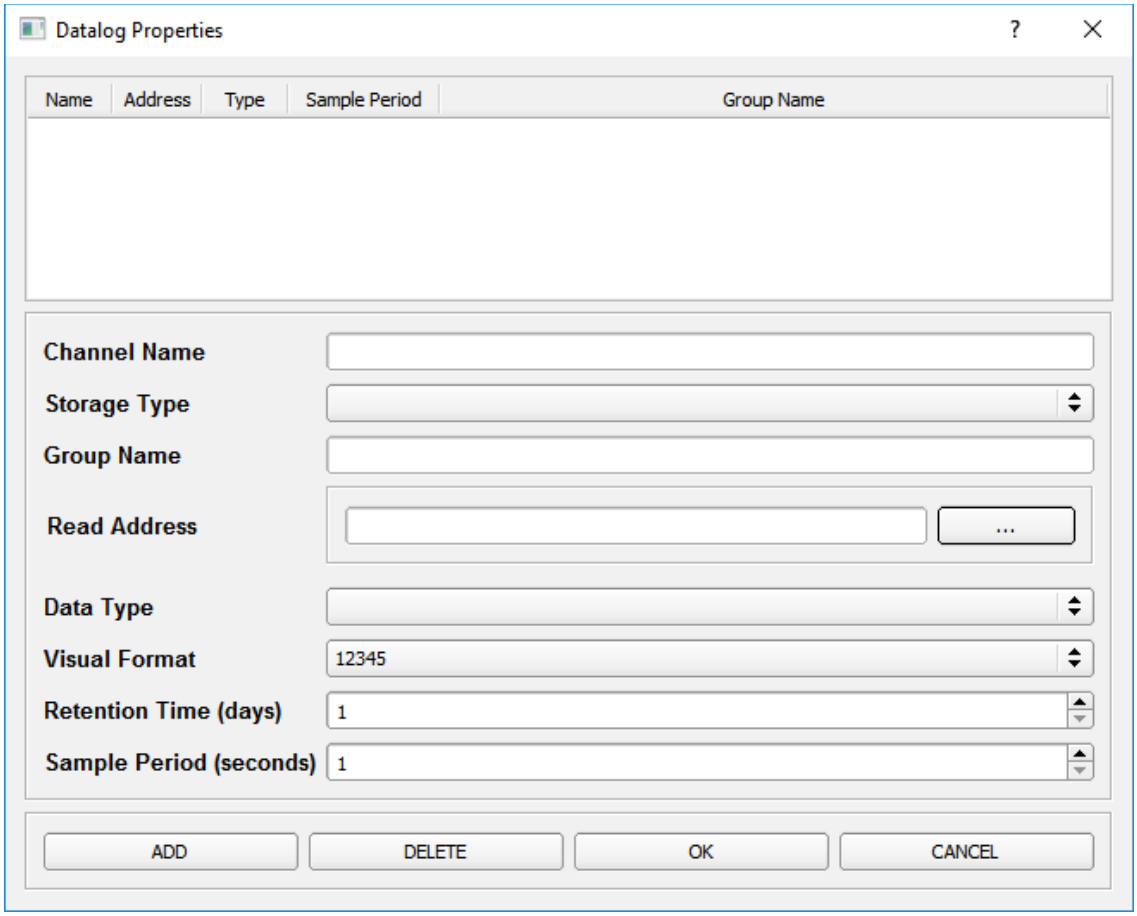
“Read Address” is data’s address.

“Data Type” is type of read data.

“Visual Format” shows data’s decimal type.

“Retention Time” shows data’s retention period.

“Sample Period” shows data’s read time.



Picture 23: Menu Bar->Options->Datalog Setup

B.1.5.4. Alarm Setup

Data is read from given address and compared according to comparison and as result alarm may rise. Setup about alarms can be configured in Alarm Setup.

“Max. Internal Records” is number of record as internal.

“Read Address” is value’s address and comparison condition.

“Alarm Text” is shown alarm text.

“Video” can be played when alarm rises.

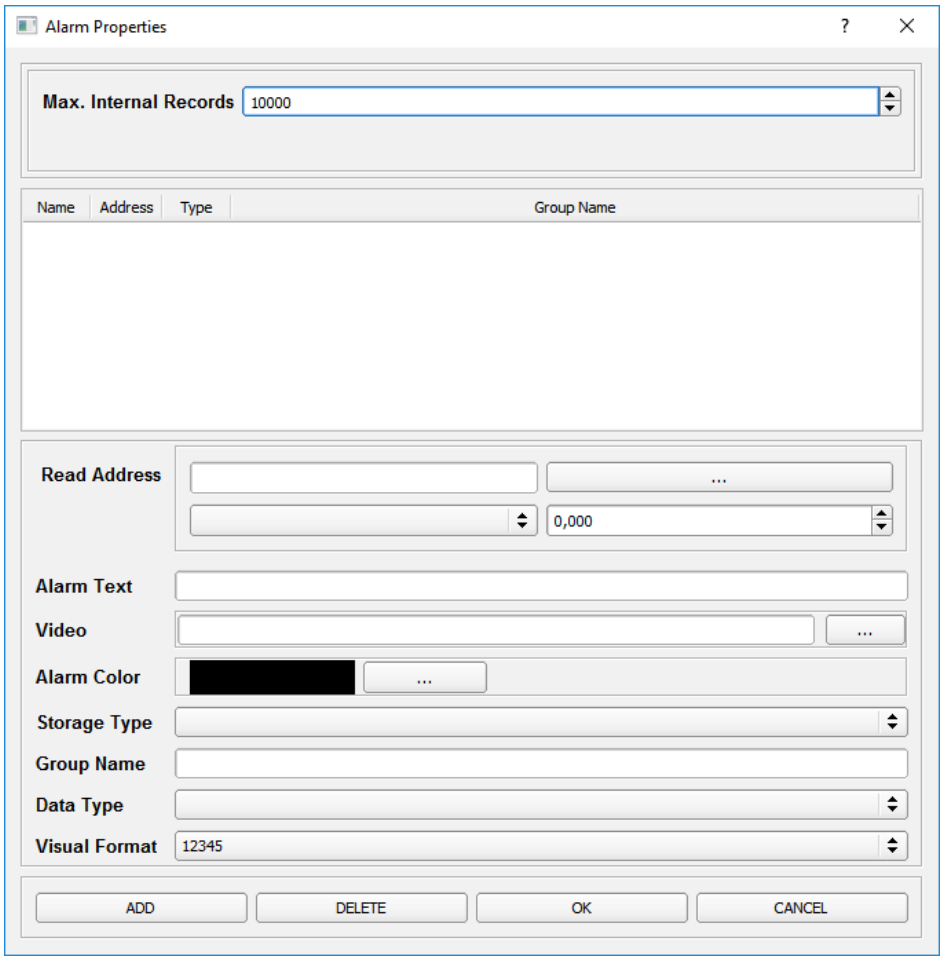
“Alarm Color” is background color of alarm.

“Storage Type” is selection of store type. CVS file can be saved to usb.

“Group Name” is name in file.

“Data Type” is type of read value.

“Visual Format” shows decimal type of value.



Picture 24: Menu Bar->Options->Alarm Setup

B.1.5.5. SQL Setup

SQLite and MS SQL database servers to connect to the window.

DB Label: Enter the name of the database to use in the macro code according to the database type (MSSQL,SQLite). (MSSQL_Demo1,SQLite_Test1, SQLite_Demo2,.. etc.)

Driver: The database type selection list. (MSSQL,SQLite)

Database: Database file name for Sqlite and database name used for MSSQL on the server must be entered.

IP: Enter the IP address of the database.

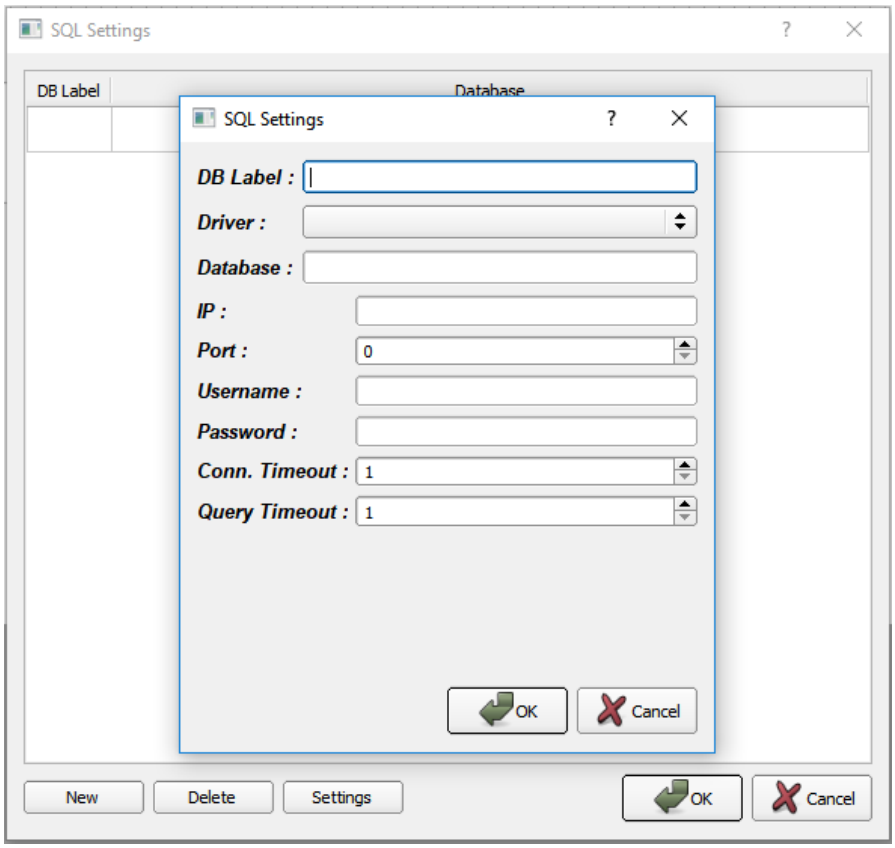
Port: Enter the port number of the database.

Username: Enter the username for connecting the database.

Password: Enter the password for connecting the database.

Conn. Timeout: Enter the connection time-out period. (In seconds for MSSQL)

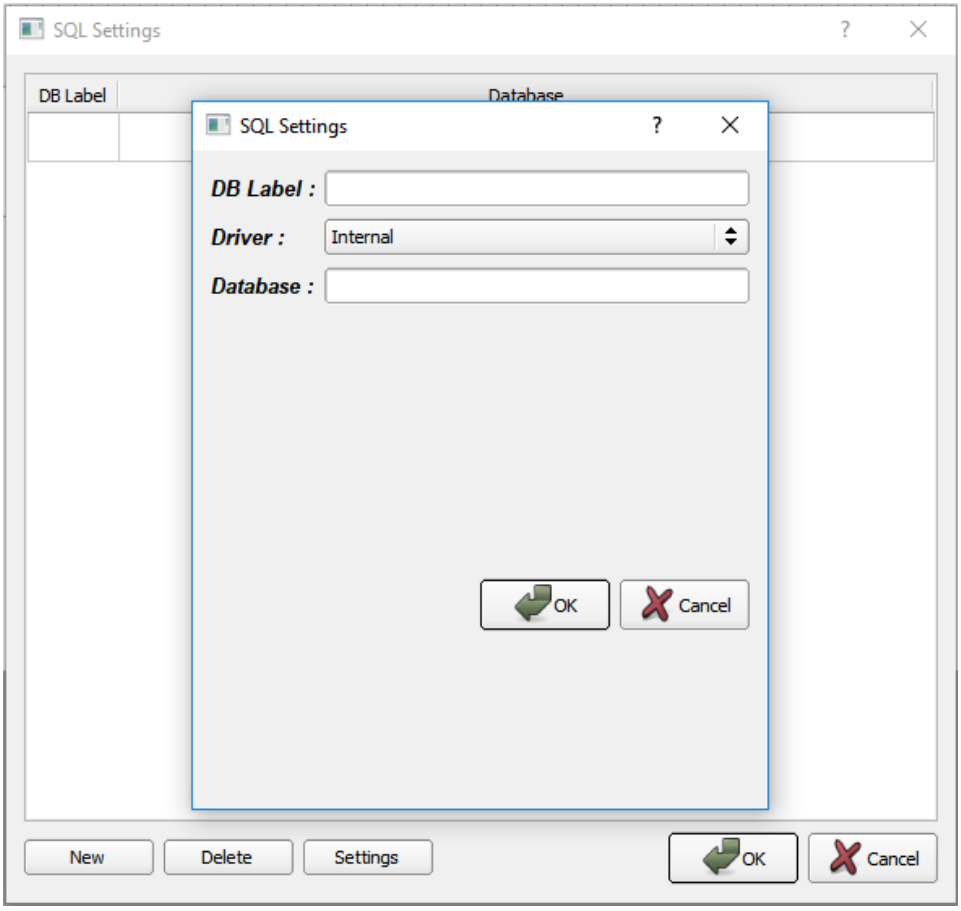
Query Timeout: Enter the query processing timeout. (In seconds for MSSQL)



Picture 25: Menu Bar->Options->SQL Setup

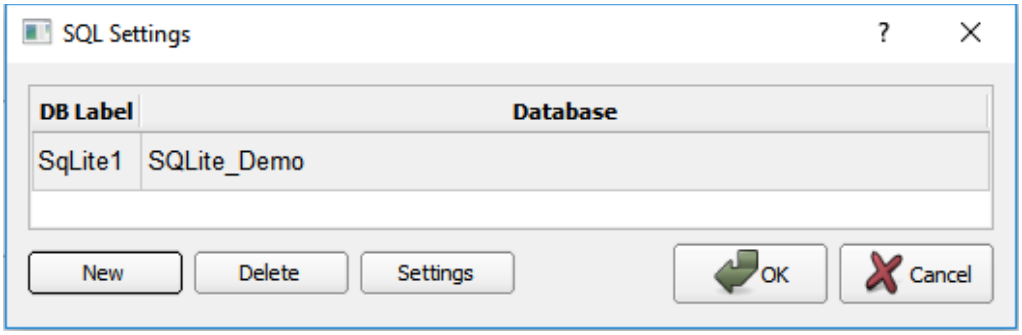
B.1.5.5.1. SQLite Settings

SQLite database settings are shown in the picture below.



Picture 26: Menu Bar->Options->SQL Setup->Internal

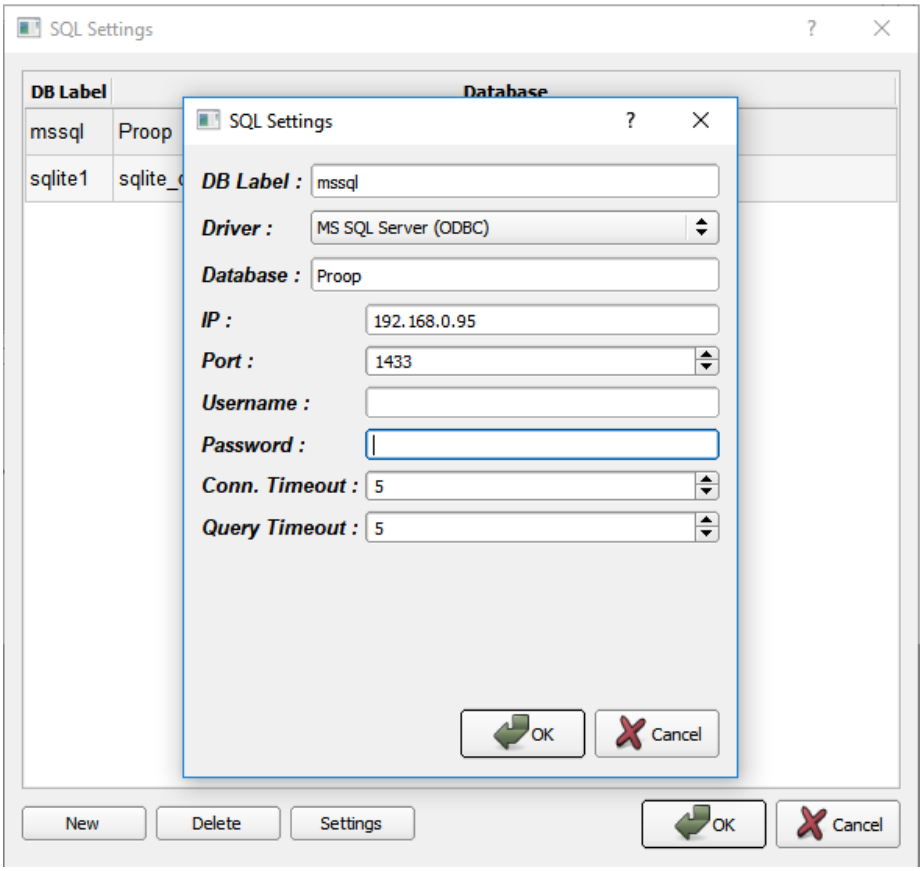
Click on the '**New**' button in the SQL setup window from on the Options Menu and select '**Internal**' under the driver heading. DB Label and Database names are specified. Press '**OK**' to save the settings.



Picture 27: SQLite Settings

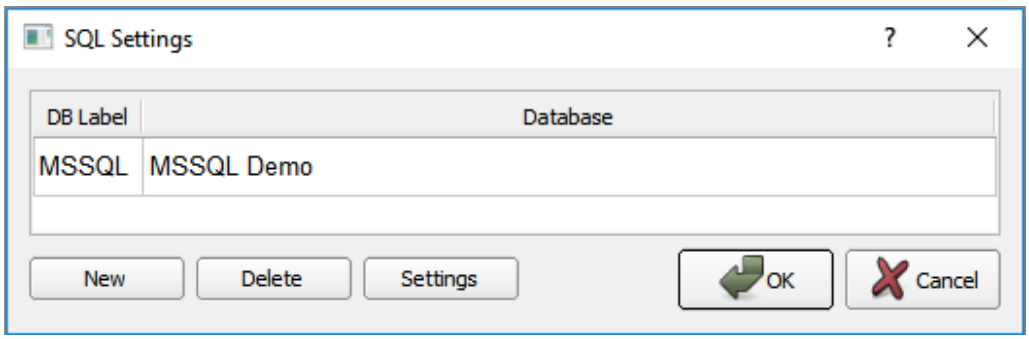
B.1.5.5.2. MSSQL Settings

MSSQL database settings are shown in the picture below.



Picture 28: Menu Bar->Options->SQL Setup->MS SQL Server (ODBC)

Click on the '**New**' button in the SQL setup window from on the Options Menu and select '***MS SQL Server (ODBC)***' under the driver heading. DB Label and Database names are specified. IP, port, user name, password, connection and query timeouts are entered and press '**OK**' to save the settings.

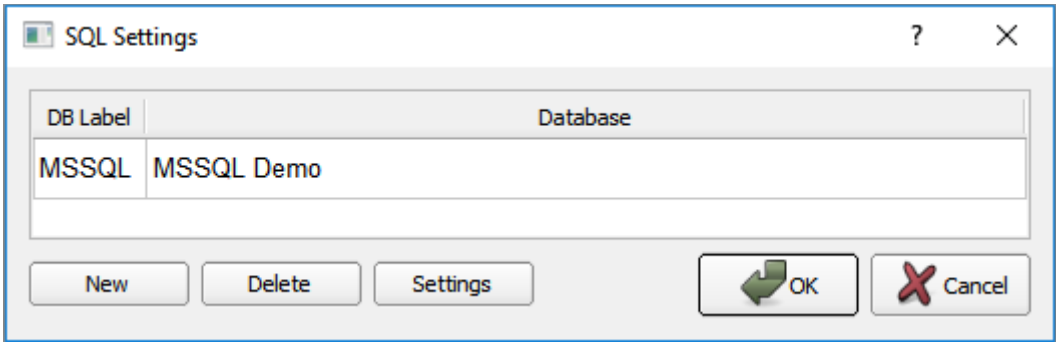


Picture 29: MSSQL Settings

Example -1: Using MS SQL

- **Step 1: Create Database**

  Menu Bar -> Options->Follow the SQL Setup steps to create the database.



- **Step 2: Create new table**

  Add a table to the database use “CREATE TABLE” command.

Using Button	Execute Macro Code
Create Table released button	1 func main() 2 local loc1; loc1 =1; 3 loc1 = execsql("mssql","create table table_demo (id int IDENTITY(1,1) PRIMARY KEY, datetime text, value text);"); 4 endf 5 endp

`execsql :` SQL queries to be executed.

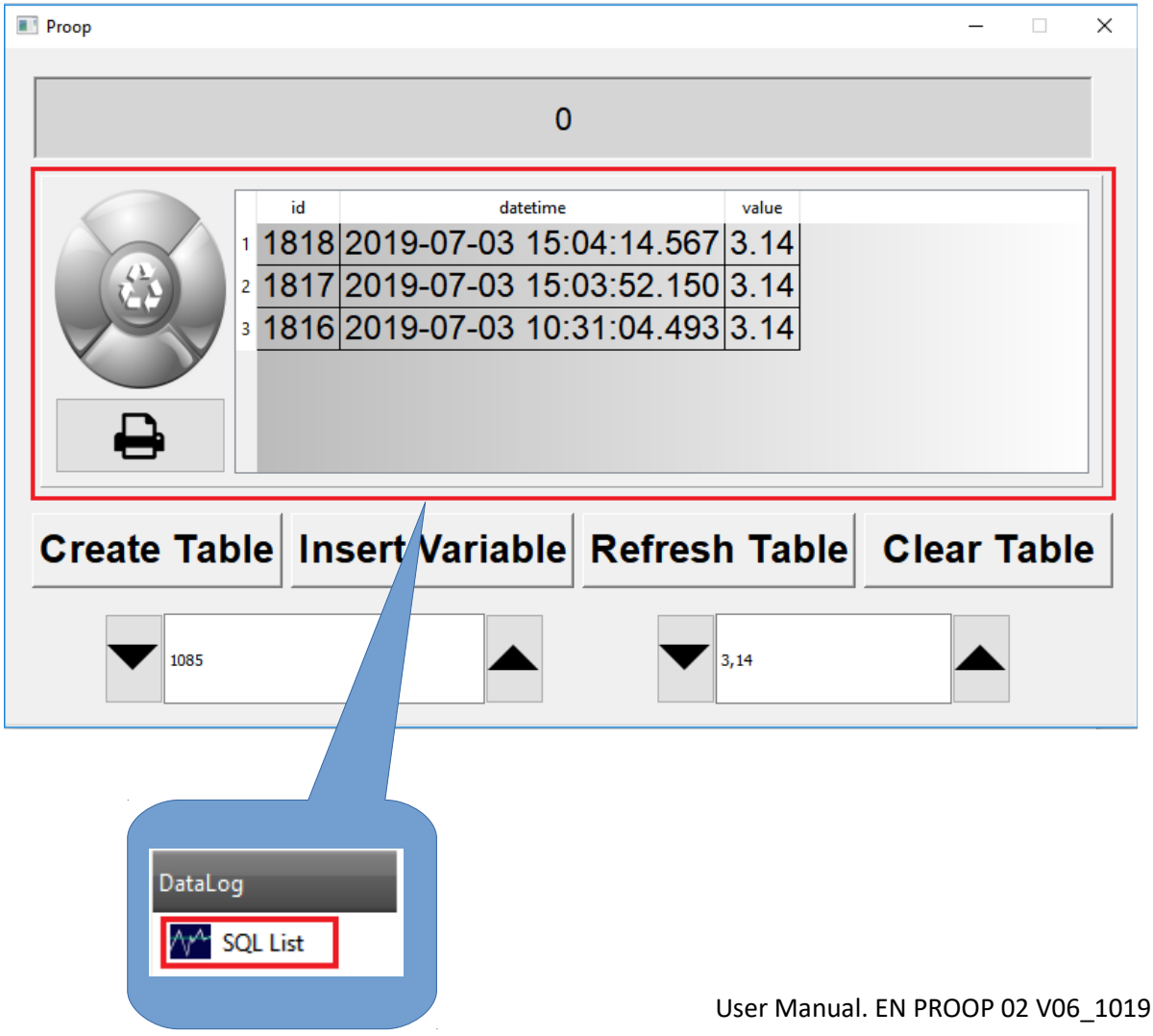
- **Step 3: Add data in table**

Add a data to the table use “Insert” command.

Insert Variable released button	1	`func main()`
	2	local loc1,str1;
	3	loc1 = execsql("mssql","insert into table_demo(datetime,value) values(CONVERT(varchar, GETDATE() ,121) , CONVERT(varchar,:value));", $0);
	4	endf
	5	endp

`execsql` : SQL queries to be executed.

Using the SQL List element, you can list the data in the table on the screen. *Element List -> Datalog -> SQL List* drag and drop the element to the workspace



- **Step 4: Update and delete data in a table**

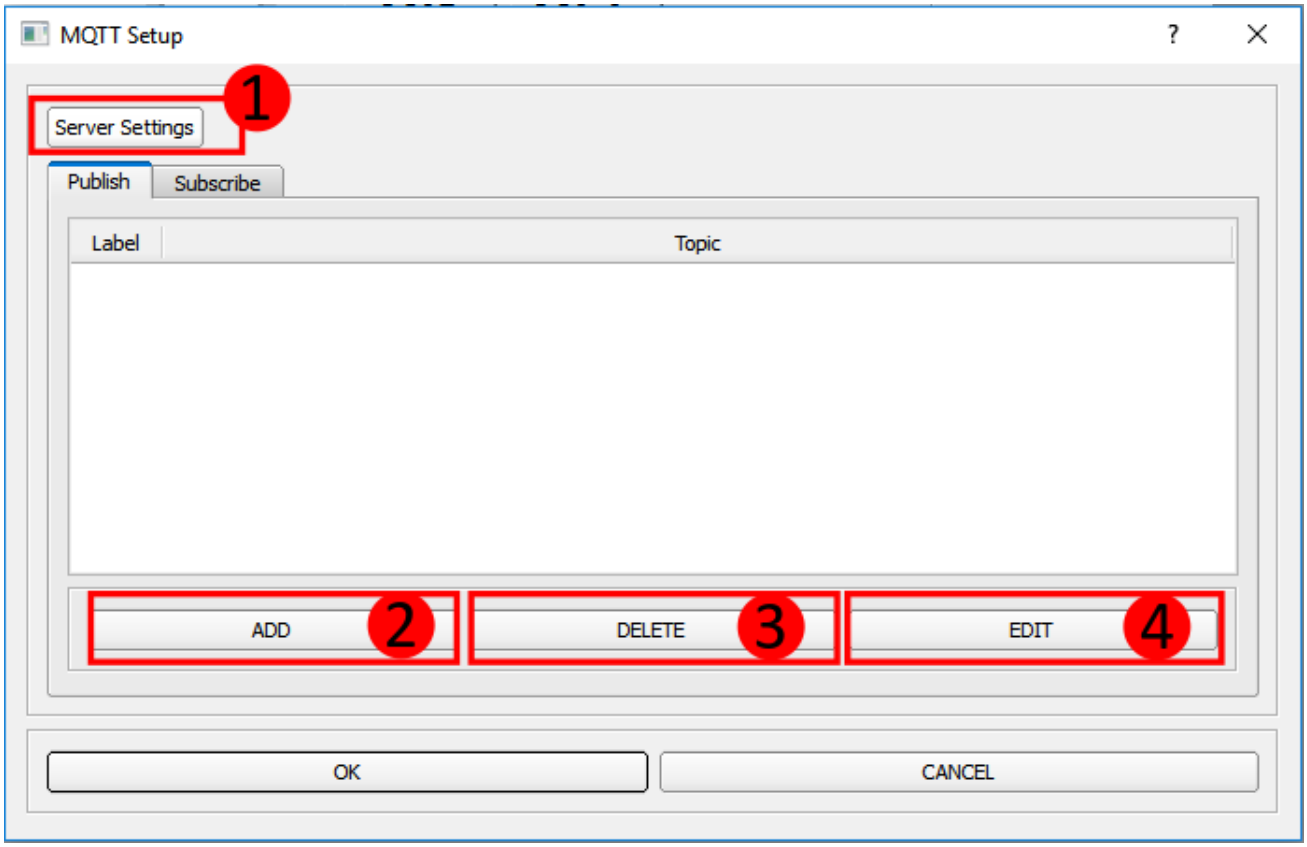
  Remove a table to the database use “DELETE” command.

<table>
<tr><th>Using Button</th><th colspan="2">Execute Macro Code</th></tr>
<tr><td>  </td><td colspan="2">Refreshes the table. This is done by using signal / slot.
clicked()->updateList()</td></tr>
<tr><td rowspan="6"> 
released button</td><td>1</td><td>func main()</td></tr>
<tr><td>2</td><td>local loc1,str1;</td></tr>
<tr><td>3</td><td>loc1 = execsql("mssql","delete from table_demo;");</td></tr>
<tr><td>4</td><td>endf</td></tr>
<tr><td>5</td><td>endp</td></tr>
</table>

`execsql :` SQL queries to be executed.

B.1.5.6. MQTT Settings

MQTT (Message Queuing Telemetry Transport), object can send message to a remote server, or subscribe to topics on a remote server.



Picture 30: Menu Bar->Options->MQTT Settings

Select server settings from area 1.

Add published message from area 3.

Delete published message from area 3.

Edit published message from area 3.

Press '**OK**' to save the settings.

B.1.5.6.1. Server Settings

General

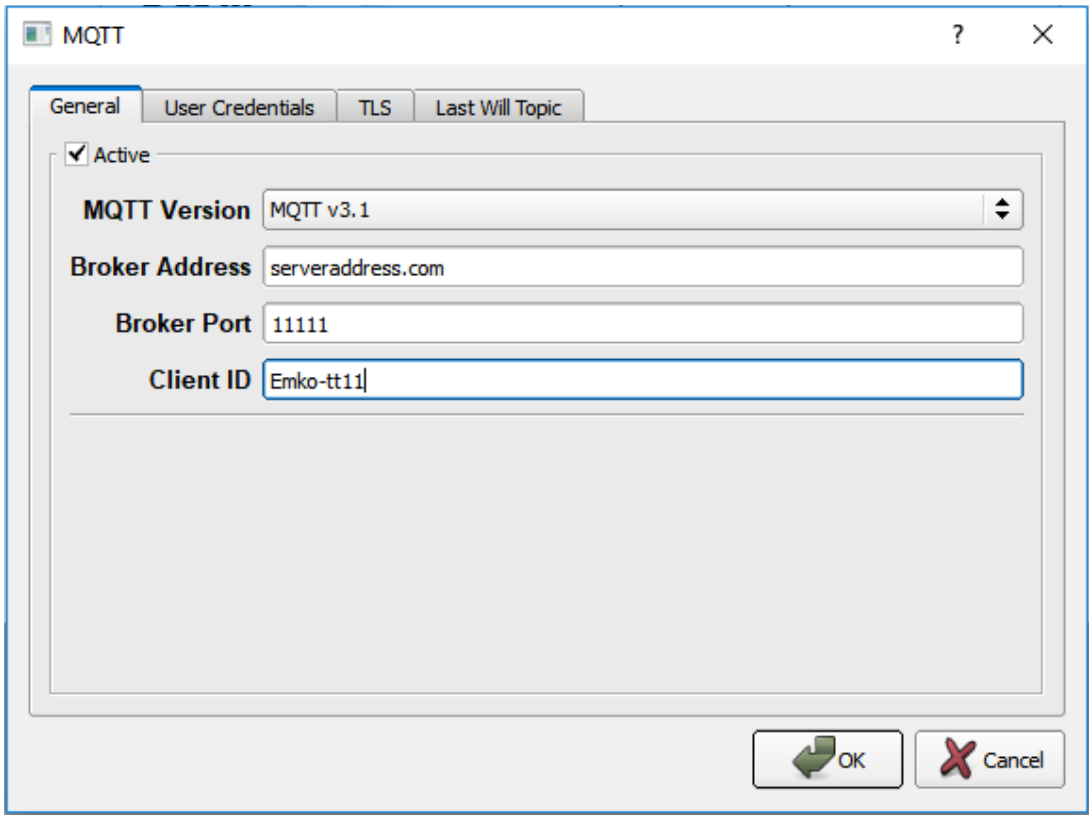
Under the General tab, there are fields where information about MQTT is entered.

MQTT Version: Supports MQTT versions.

Broker Addresss: Enter MQTT broker address.

Broker Port: Enter MQTT broker port number.

Client ID: Enter connection client ID.



Picture 31: Options->MQTT Settigs→Server Settings->General

Note: "Active" selection, server settings can be active or passive.

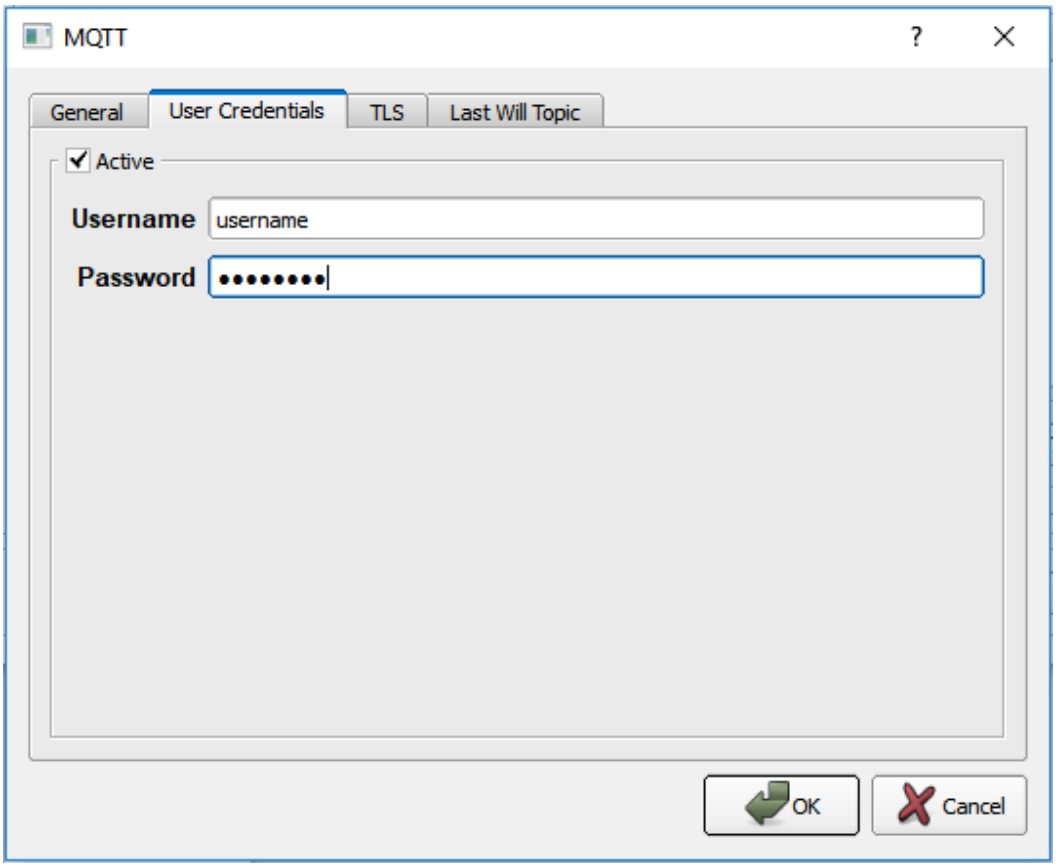
User Credentials

User credential is entered under this tab.

Username: The username that provides access to the server is entered in this field.

Password: The password that provides access to the server is entered in this field.

The settings are saved by pressing the "**OK**" button.



Picture 32: Options->MQTT Settings->Server Settings->User Credentials

Note: "Active" selection, user credential can be active or passive.

TLS

Enable TLS authentication.

Enable TLS : TLS authentication enable or disable settings.

Versiyon : TLS version can be selected from this area.

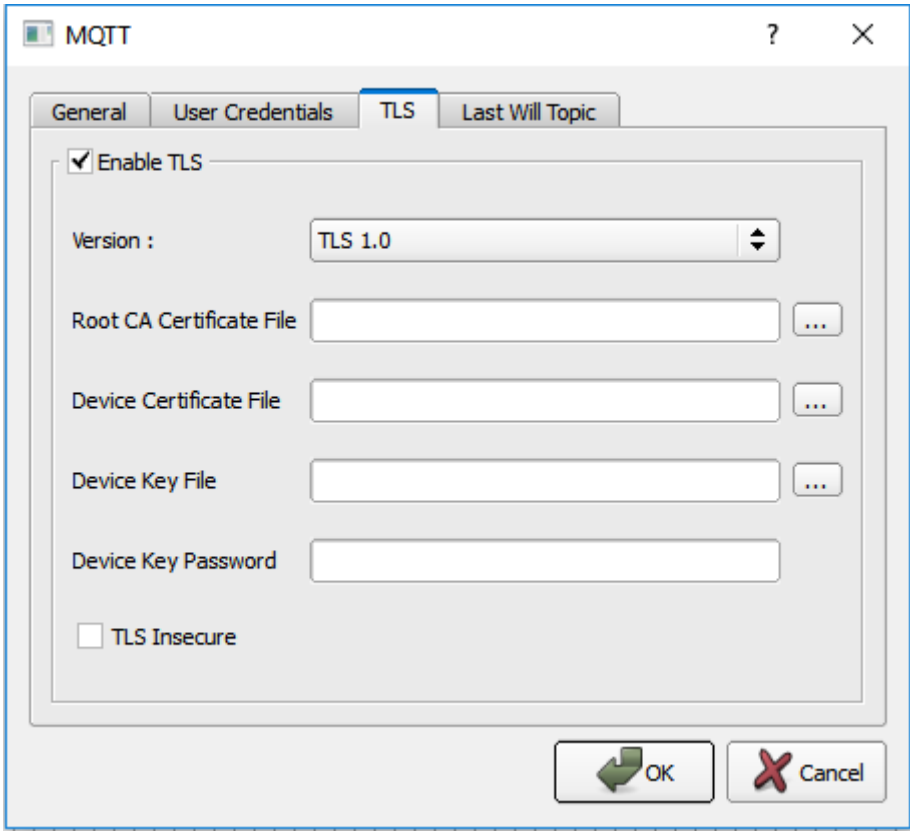
Root CA Certificate File : Select root CA certificate file from area.

Device Certificate File: Select device certificate file from area.

Device Key File: Select device key file from area.

Device Key Password: Enter device key password from this area.

TLS Insecure: This option is used to enable or disable the TLS insecure option.



Picture 33: Options->MQTT Settings->Server Settings->TLS

Last Will Topic

In MQTT, you use the Last Will and Testament (LWT) feature to notify other clients about an ungracefully disconnected client. Knowing whether a client disconnected gracefully or ungracefully (without a disconnect message), helps you respond correctly.

Enable Last Will Topic: Indicates whether the LWT feature is active or passive.

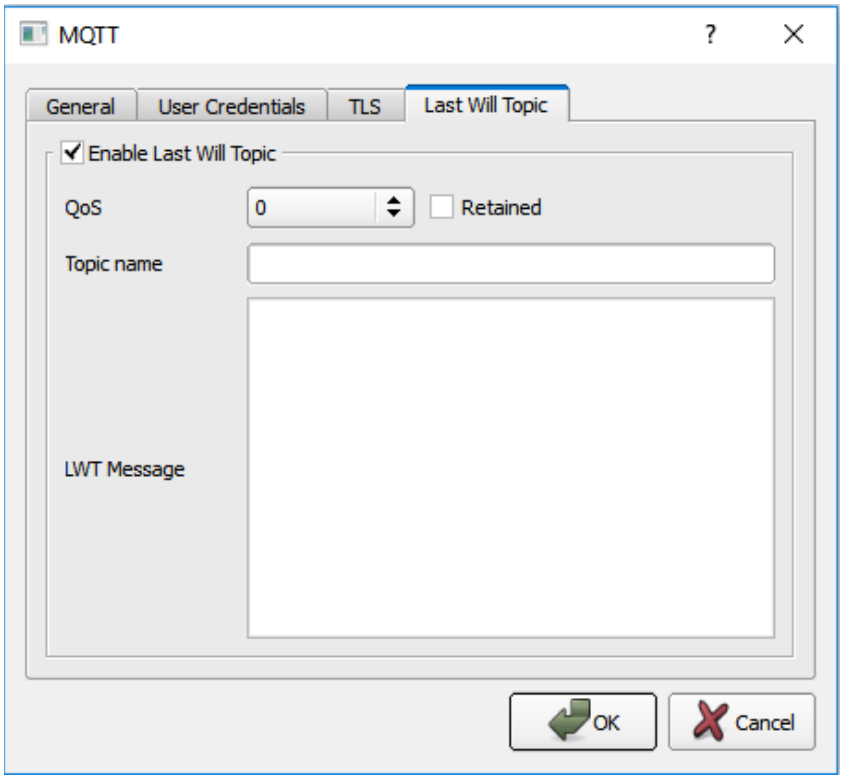
QoS : MQTT provides three levels of reliability, which are known as qualities of service (QoS). The reliability of the message determines the persistence of the message.

- 0: At most once, messages are not persistent.
- 1: At least once.
- 2: Exactly once.

Retained: The option that permanently or permanently activates the message.

Topic Name: Enter topic name from area.

LWT Message: Enter LWT message from area.



Picture 34: Options->MQTT Settings->Server Settings->Last Will Topic

B.1.5.6.2. MQTT Topic Publisher

MQTT Settings window opens from the options menu. Press the 'Add' button under the publish field. Window tabs are described under subtitles.

Topic Properties

Label: The alias for the MQTT topic. (Ex: Alarm1,Temp1,Value1)

Topic: This is the field where a topic is specified on the MQTT server. The title can be dynamic.

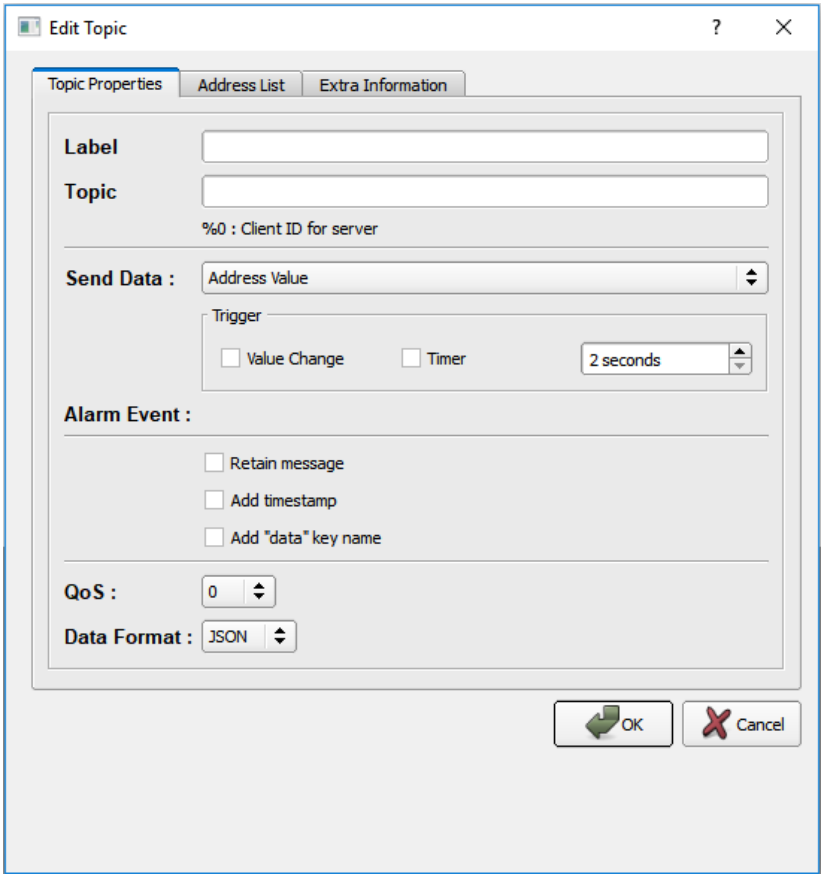
Send Data: Specifies the sending trigger status of the message.

- Value Change : Send the message according to the values in the address list.
- Timer : Send the message according to the specified time.

Alarm Event: Select alarm status from area.

QoS: The service that determines the reliability of the message.

Data Format: Select data format from area. (Ex : JSON)



Picture 35: Options->MQTT Settings->Add->Topic Properties

Address List

If the 'Value Change' box of the 'Trigger' title under the topic properties section is checked, the message is broadcast according to the addresses defined in the Address List.

Value Name: The name of the value to be read is entered from this field.

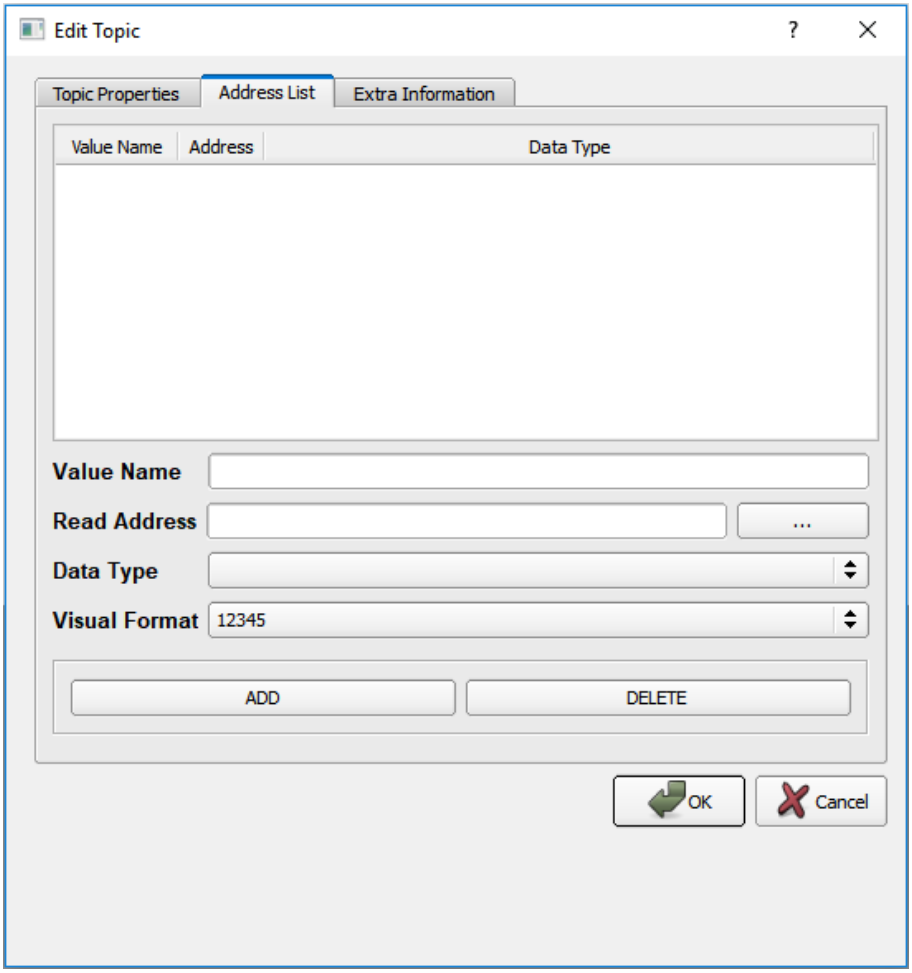
Read Adsress: Specifies the address to read.

Data Type: Specifies the data type to read.(Ex: bit,float,double,UnsignedInt8,...)

Visual Format: If the data to be read is displayed as a dotted value, it is selected from this field.

If the data is a non-point value, it contains "." must select the option without an expression.

Press the '***Add***' button, the values created in the Address List are added. Press '**OK**' button and save address list.

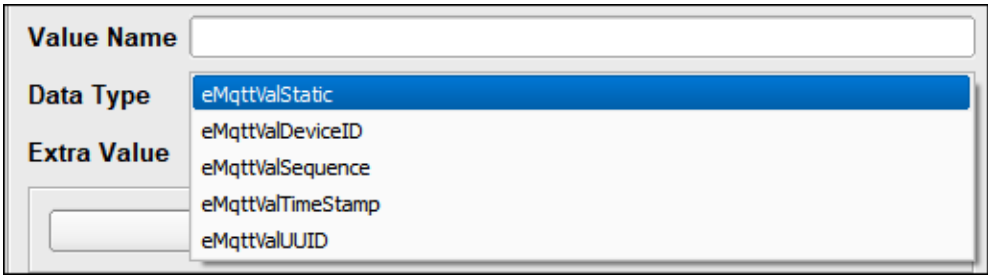


Picture 36: Options->MQTT Settings->Add->Address List

Extra Information

Value Name: The name of the value to be used in the system and to be saved is entered from area.

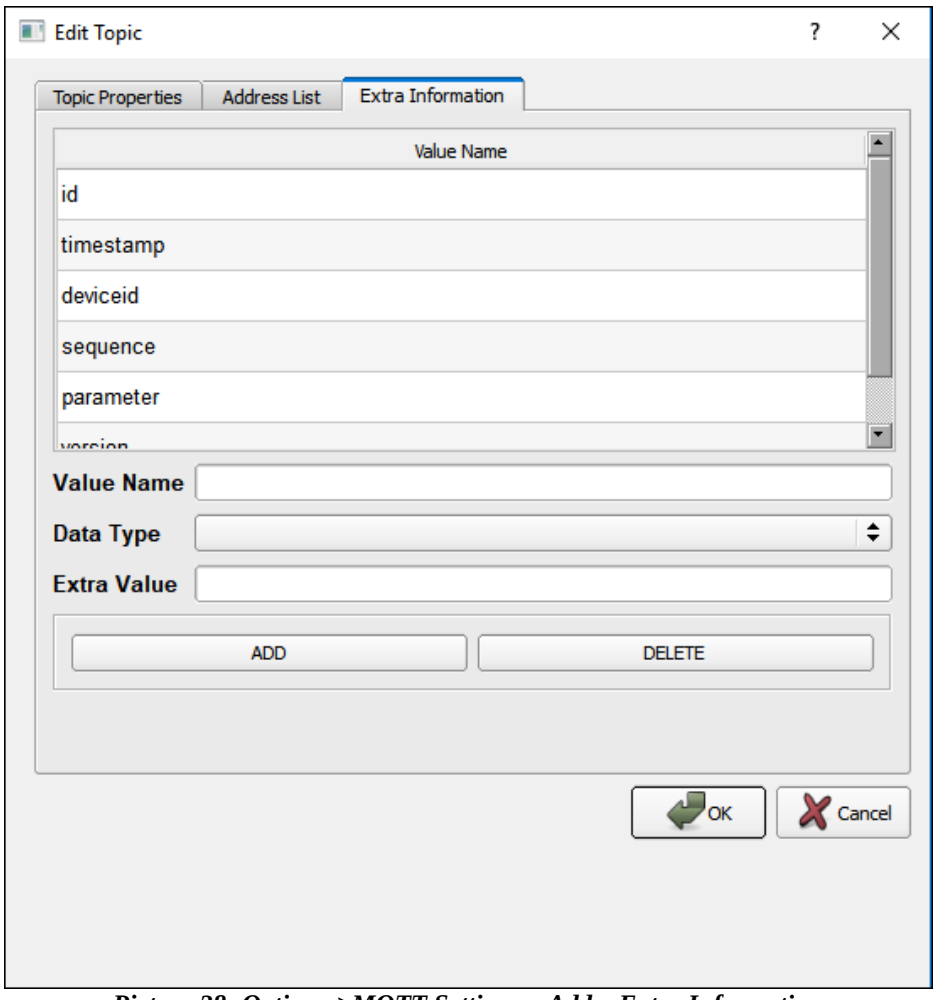
Data Type: Input value data type.



Picture 37: Data Type

Extra Value: The text is an explanatory text of the value name. The following example is described.

Example ; Value Name: parameter ,Data Type: Static Text, Extra Value: Alarm1



Picture 38: Options->MQTT Settings→Add→Extra Information

B.1.5.6.3. MQTT Functions

Function	getmqttid()
Comment	Returns the Client ID value specified in the Mqtt server settings.
Example	`$0 = getmqttid();`

Function	sendmqtt()
Comment	Message sending function.
Example	`sendmqtt(("devices/" + getmqttid() + "/messages/events/"), jsonstring);`

B.1.5.6.4. IOT Functions

Function	getjsonmap()
Comment	Creates a map in Json data format. Its structure is as follows. { .... : .... : .... : ..... }
Usage	`local json1;` `json1 = getjsonmap();`
Example	{ "backgroundcolor":"#656667", "height" : 4, "width" : 4 }

Function	putjsonmap()	
Comment	A function that adds the data of the map structure created by getjsonmap.	
Usage	putjsonmap(local val,"key" ,data,);	
Example	local json1; json1 = getjsonmap(); putjsonmap(json1,"Id", getuuid());	local json1,json2; json1 = getjsonmap(); json2 = getjsonmap(); putjsonmap(json1, "Id", getuuid()); putjsonmap(json2, "Id", getuuid()); putjsonmap(json1, json2); //Nested map created

Function	getuuid()
Comment	Function that holds a unique id.
Usage	$0 = getuuid();
Example	local json1; json1 = getjsonmap(); putjsonmap(json1, "Id", getuuid());

<table>
<tr><th>Function</th><th>getdatevalue()</th></tr>
<tr><td>Comment</td><td>Returns the value of the type (int type) to the shape of the given format.</td></tr>
<tr><td>Usage</td><td>
<pre>getdatevalue(int timevalue, "timeformat"));</pre>
Time format descriptions :
<table>
<tr><th>expression</th><th>comment</th></tr>
<tr><td>d</td><td>Specifies the day number. Without 0 at the beginning.(1-31)</td></tr>
<tr><td>dd</td><td>Specifies a day number per 0.(01-31)</td></tr>
<tr><td>ddd</td><td>Specifies the abbreviated day name.(Mon,Wed vb.)</td></tr>
<tr><td>dddd</td><td>Specifies the full name of the day. (Monday,Tuesday vb.)</td></tr>
<tr><td>M</td><td>Specifies the month number. Without 0 at the beginning.(1-12)</td></tr>
<tr><td>MM</td><td>Specifies a month number per 0.(01-12)</td></tr>
<tr><td>MMM</td><td>Specifies the abbreviated month name. (Feb,May vb.)</td></tr>
<tr><td>MMMM</td><td>Specifies the full name of the month.(February,May vb.)</td></tr>
<tr><td>yy</td><td>Specifies the year as a two-digit number.(00-99)</td></tr>
<tr><td>yyyy</td><td>Specifies the full-digit version of the year.(2000,1992 vb.)</td></tr>
<tr><td>h</td><td>Holds the hour number. Without 0 at the beginning.(1-23 AM,1-12 PM)</td></tr>
<tr><td>hh</td><td>Specifies by taking the number 0 per time.(01-23 AM,01-12 PM)</td></tr>
<tr><td>H</td><td>No matter whether AM-PM shows the time number between 0-23</td></tr>
<tr><td>HH</td><td>No matter whether AM-PM shows the number between 00-23, taking 0 per hour number.</td></tr>
<tr><td>m</td><td>Displays the minute value. Without 0 at the beginning.(0-59)</td></tr>
<tr><td>mm</td><td>Indicates by taking the value 0 per minute.(00-59)</td></tr>
<tr><td>s</td><td>Displays the seconds value. Without 0 at the beginning.(0-59)</td></tr>
<tr><td>ss</td><td>Indicates a value of 0 per second.(00-59)</td></tr>
<tr><td>z</td><td>Holds the value in milliseconds. Without 0 at the beginning.(0-999)</td></tr>
<tr><td>zzz</td><td>Keeps the value in milliseconds at the beginning of 0.(000-999)</td></tr>
<tr><td>AP veya A</td><td>AM or PM definition</td></tr>
<tr><td>ap veya a</td><td>am or pm definition</td></tr>
<tr><td>t</td><td>Returns the time zone.</td></tr>
</table>
</td></tr>
<tr><td>Example</td><td><pre>$0 = getdatevalue(getsystime(), "yyyy-MM-ddThh:mm:ss.zzzZ"));</pre></td></tr>
</table>

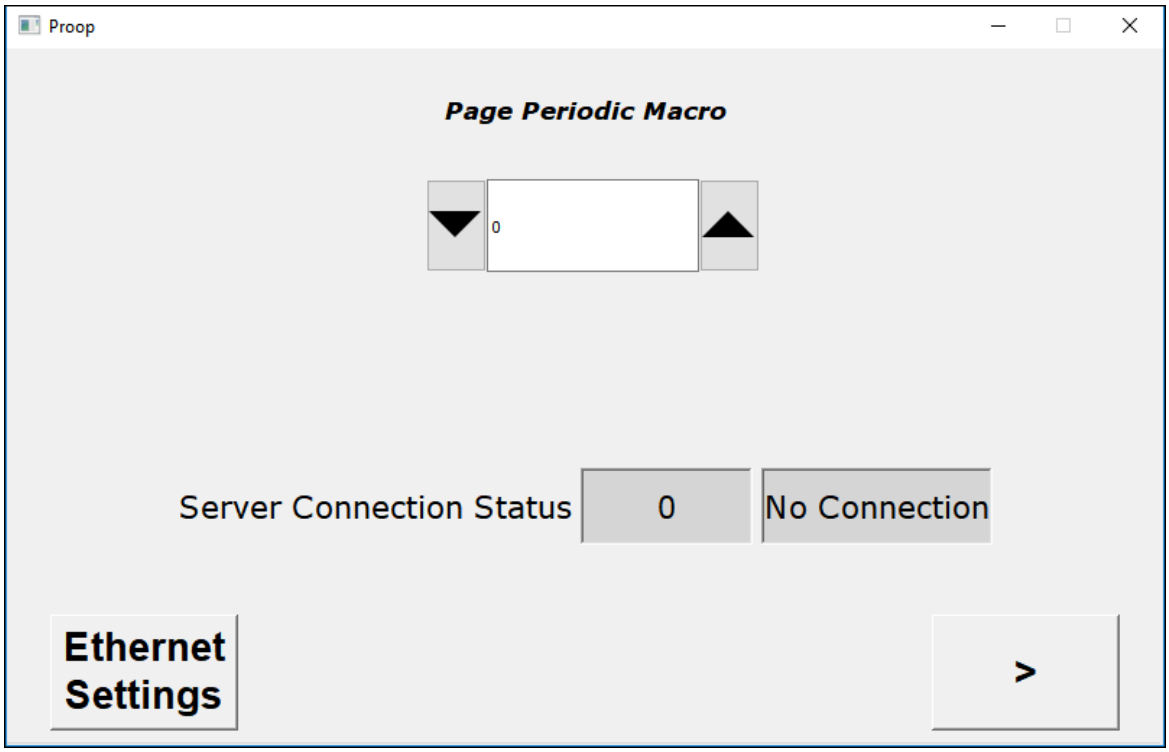
Function	getjsonlist()
Comment	This structure format is used for holding json map value. [{ .... : .... : .... : ..... }, { .... : .... : .... : ..... }]
Usage	global jsonlist; jsonlist = getjsonlist();
Example	global jsonlist; func sendData(param, val) .... endf func main() local ret1; if $M1 != $M0 $M1 = $M0; jsonlist = getjsonlist(); call sendData("Temperature", $M0); ret1 = getjsonstr(jsonlist); endif; endf endp

Fonksiyon	getjsonstr()
Comment	Prints the data in Json format as string format.
Usage	`$0 = getjsonstr(jsonlist);`
Example	`global jsonlist;` `func main()` `  jsonlist = getjsonlist();` `  $0 = getjsonstr(jsonlist);` `endf`

B.1.5.6.5. MQTT Application Examples

Example -1: MQTT server connection

This example describes the connection settings to the MQTT server. The screenshot is as follows.



Picture 39: Example -1 Screenshot

- The server link state element read address is read from **MqttStatus** '**$S5**' under **internal settings**.
- The element addresses used are indicated in the table below.

Element	Data Type	Write Address	Read Address
Counter	Double	internal_memory@$M0	-
Server Connection Status (Show Number)	SignedInt32	-	internal_memory@$S5
Show Multi State Label *(No Connection, Connected, Error)*	Double	-	internal_memory@$5

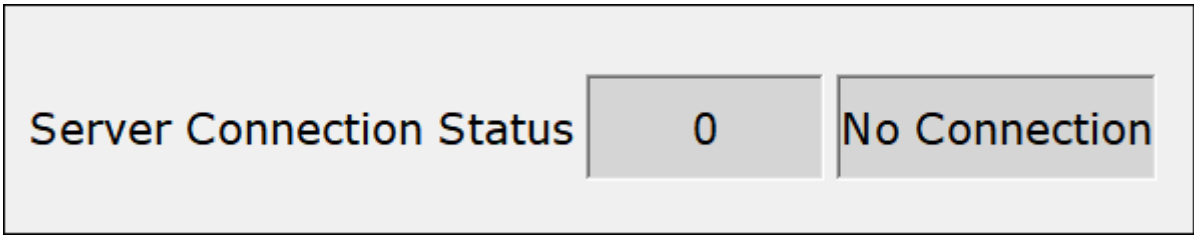
- With the Timer macro, the MQTT connection status is checked at the specified timer period.

Timer Macro Code

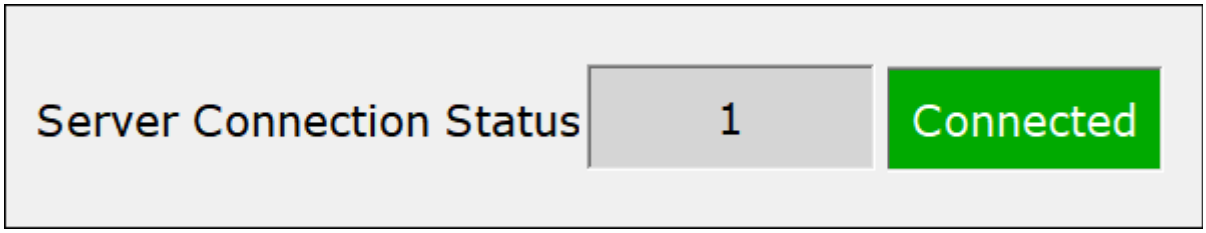
```
global g_var1, jsonlist;
func main()
local ret1, loc1, loc2, locint;

if $S5 == -1      // if the error is returning from $S5;
  $5 = 2;         // write 2 status to $5 address.
else
  $5 = $S5;       // If there is no error, write $S5 to $5.
endif;

endf
endp
```

- The connection status is indicated by 3 states using the multi-status label.

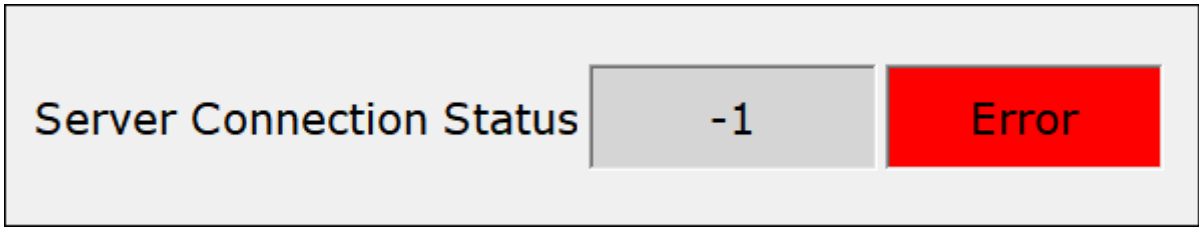
0. Status: ***No Connection*** *, MqttStatus : 0*



1. Status: ***Conectted*** *, MqttStatus : 1*



- *2. Status:* ***Error*** *, MqttStatus : -1*



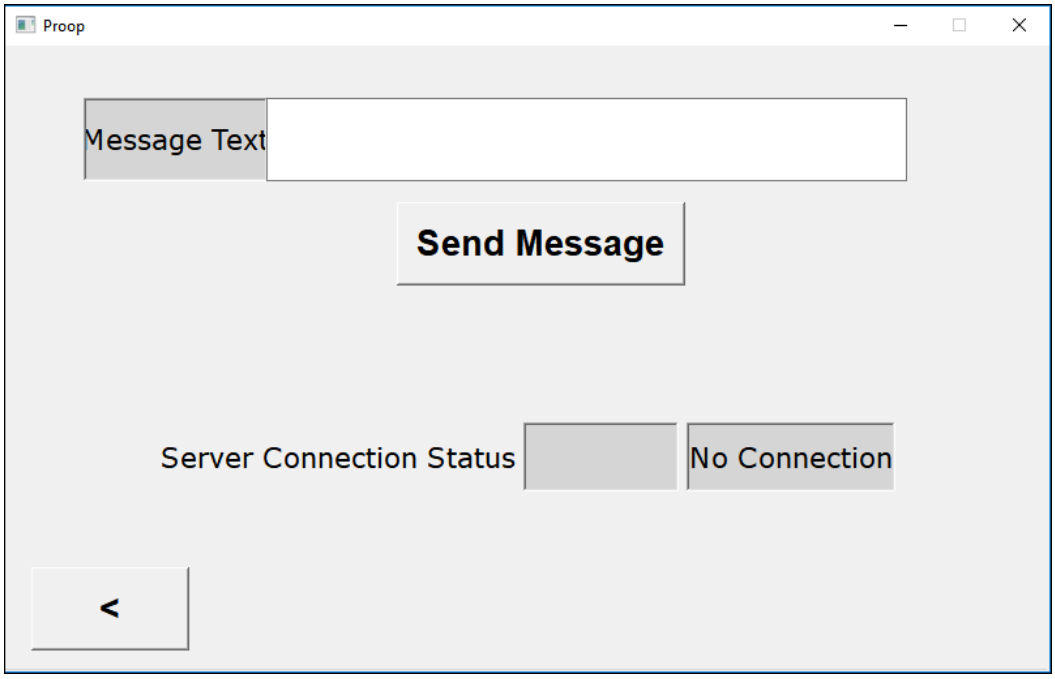
- The Message that is created using page periodic macro code in json format is send to mqtt server.

Page Periodic Macro Code

```
global jsonlist;

func sendData(param, val)
local ret1, json1;
json1 = getjsonmap();
putjsonmap(json1, "Id", getuuid());
putjsonmap(json1, "timestamp", getdatevalue(getsystime(), "yyyy-MM-
ddThh:mm:ss.zzzZ"));
putjsonmap(json1, "deviceid", getvalue(getmqttid(), "string"));
putjsonmap(json1, "parameter", param);
putjsonmap(json1, "value", val);
putjsonmap(json1, "version", "1.0.0");
putjsonmap(json1, "sequence", (int)$0);
putjsonmap(jsonlist, json1);
$0 = $0 + 1;
endf
func main()
local ret1, loc1, loc2, locint;
if  $M1 != $M0
     $M1 = $M0;
     jsonlist = getjsonlist();
     call sendData("Temperature", $M0);
     call sendData("Humidity", $M0 * 2);

     ret1 = getjsonstr(jsonlist);
     loc2 = sendmqtt(("devices/" + getmqttid() + "/messages/events/"), ret1);
endif;
endf
endp
```

Example -2: Send message MQTT server

This example describes how to send messages to the MQTT server. The screenshot is as follows.



Picture 40: Example -2 Screenshot

- The addresses of the elements used are as follows.

Element	Data Type	Write Address	Read Address
TextInput (Message Text)	Double	internal_memory@$M110	-
Server Connection Status	SignedInt32	-	internal_memory@$S5
Show Multi State Label *(No Connection, Connected, Error)*	Double	-	internal_memory@$5

- Enter message text. Press “**Send Message**” button then release the button. The macro code will be executed when the button is released. “**Released**” macro code is given below.

Released Macro Code

```
global jsonlist;

func sendData(param, val)
local ret1, json1;

json1 = getjsonmap();

putjsonmap(json1, "Id", getuuid());
putjsonmap(json1, "timestamp", getdatevalue(getsystime(), "yyyy-MM-
ddThh:mm:ss.zzzZ"));
putjsonmap(json1, "deviceid", getvalue(getmqttid(), "string"));
putjsonmap(json1, "parameter", param);
putjsonmap(json1, "value", val);
putjsonmap(json1, "version", "1.0.0");
putjsonmap(json1, "sequence", (int)$0);
putjsonmap(jsonlist, json1);
$0 = $0 + 1;
endf

func main()
local ret1, loc1, loc2, locint;
jsonlist = getjsonlist();
call sendData("Message", (string)$M110);
ret1 = getjsonstr(jsonlist);
loc2 = sendmqtt(("devices/" + getmqttid() + "/messages/events/"), ret1);
endf
endp
```

B.1.5.7. Recipe Editor

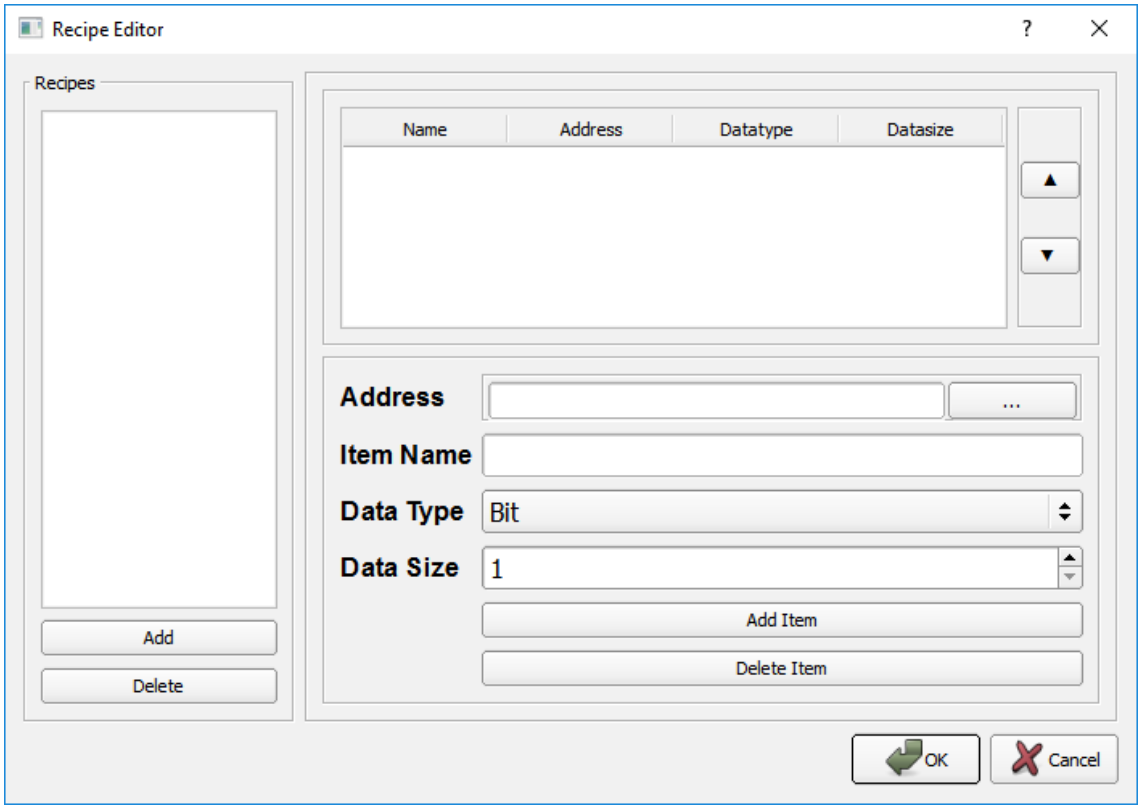
In this editor, new recipe and it's items can be added and configured.

"Address" is item's address.

"Item Name" is given name to item.

"Data Type" is item's data type.

"Data Size" is item's data size.

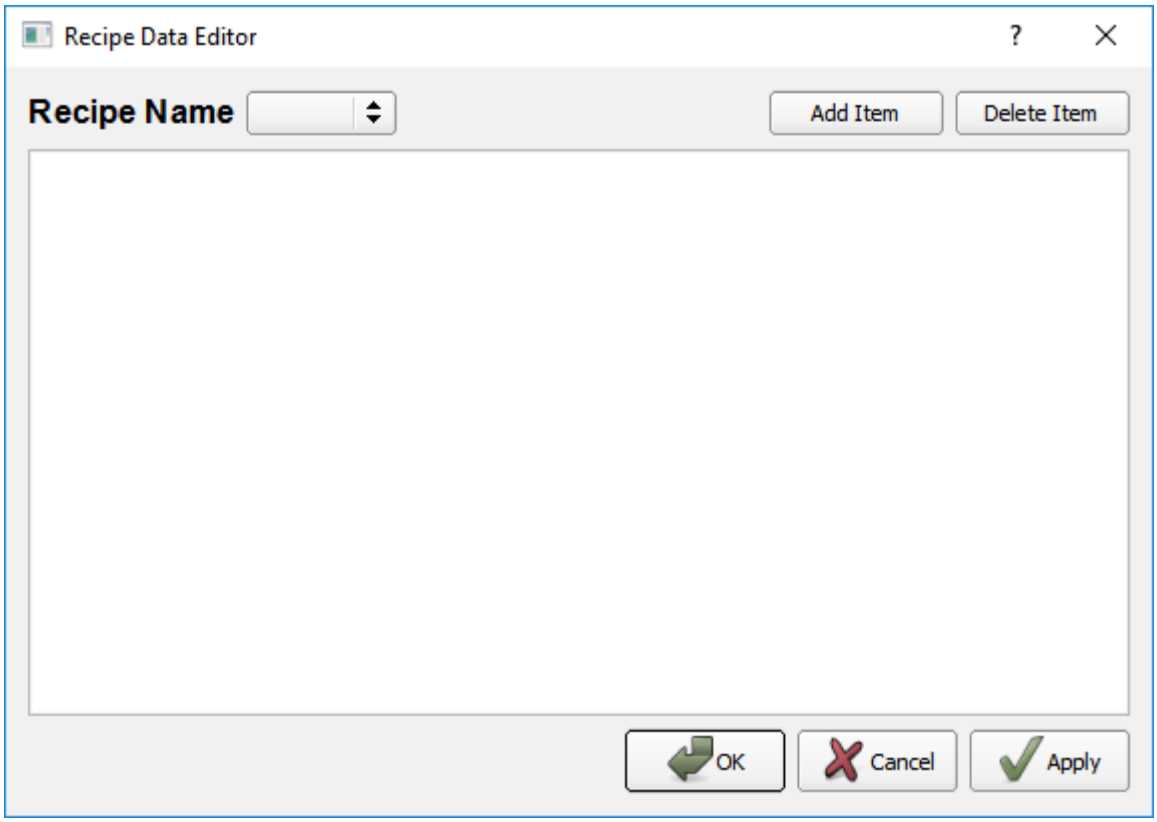


*Picture **41: Menu Bar → Options → Recipe Editor***

B.1.5.8. Recipe Data Editor

In this editor, programs can be derived with prepared recipe.

“Add Item” is used to add a new item and under “title”, a name can be entered and values can be assigned to recipe’s items.



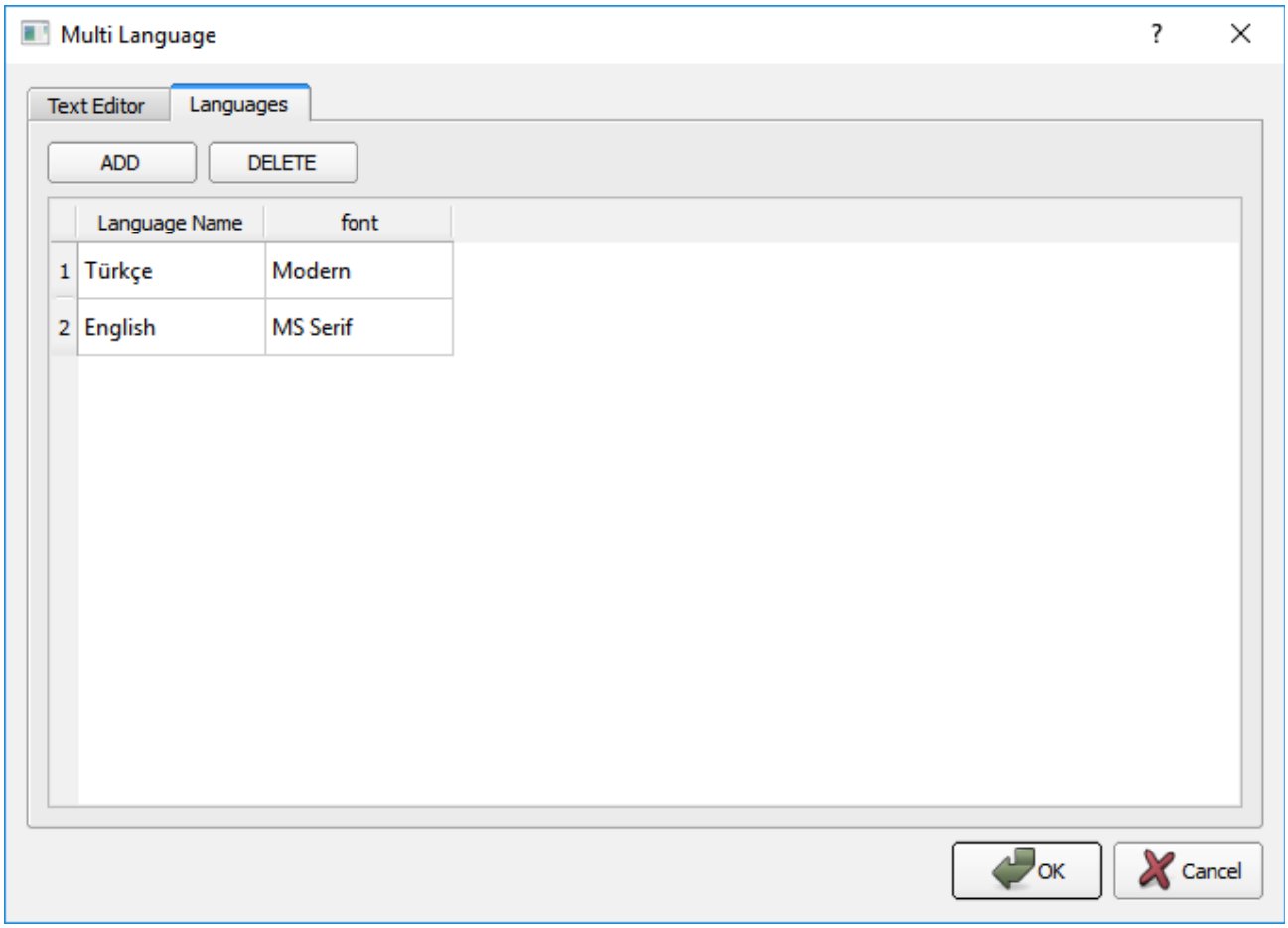
*Picture **42: Menu Bar → Options → Recipe Data Editor***

B.1.5.9. Language Editor

This editor helps user for language types and translations. Labels can be assigned by language to elements in forms and labels' fonts can be changed. Translations can be exported or imported as Excel file.

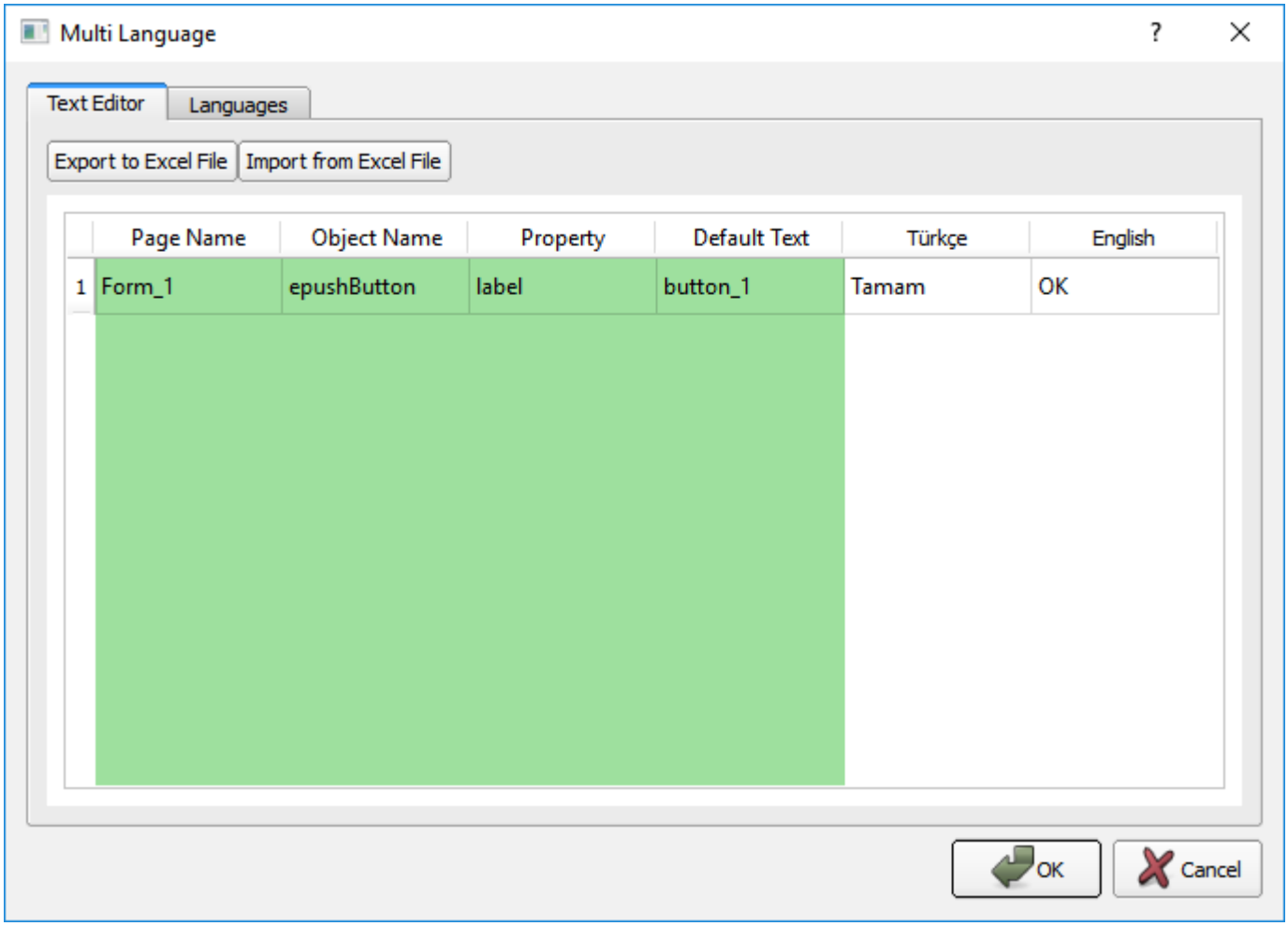
Languages can be added and its' fonts can be changed in Languages tab.

After adding language, Text Editor tab should be configured.



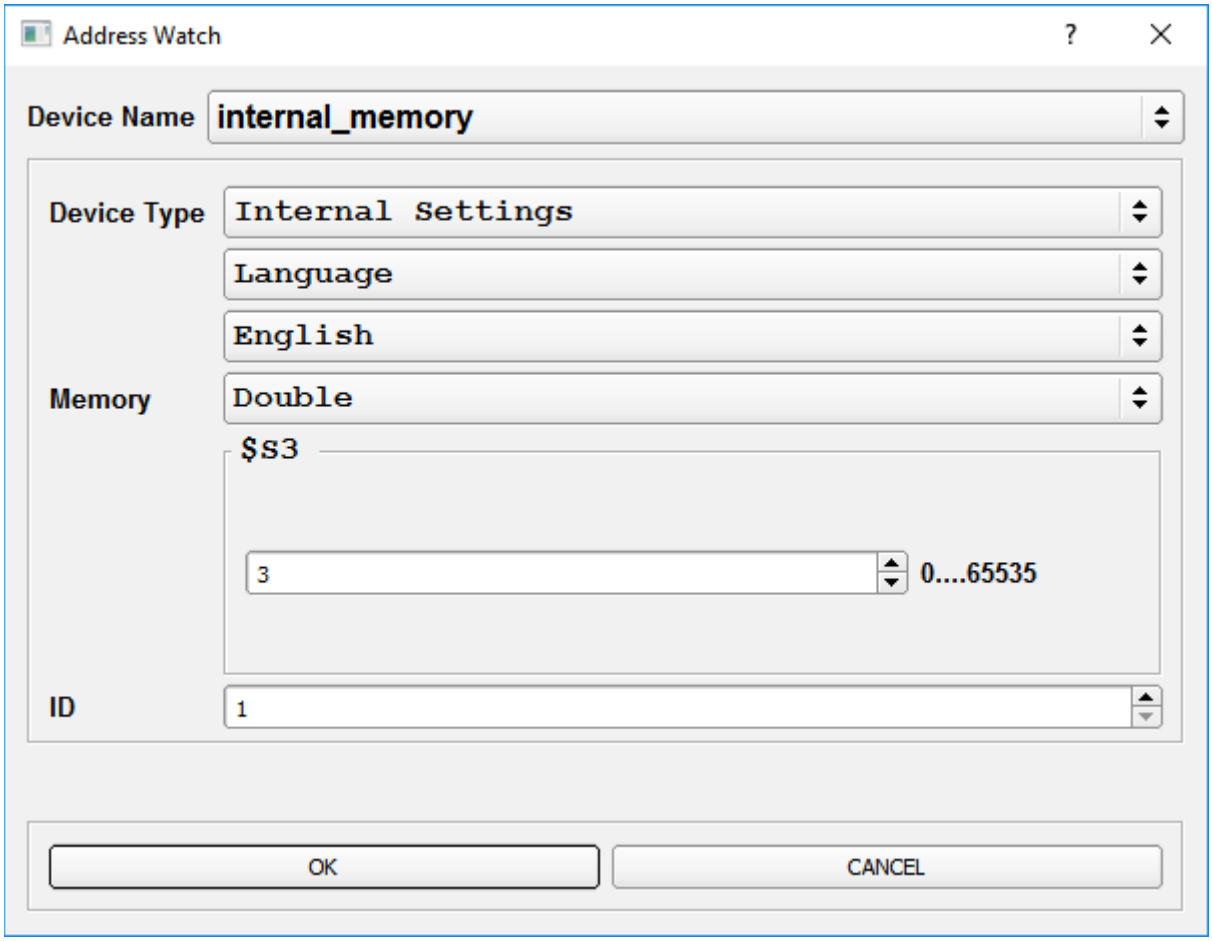
*Picture **43: Menu Bar → Options → Language Editor->Languages***

In Text Editor, elements are listed and labels are translated by added languages.



*Picture **44: Menu Bar → Options → Language Editor->Languages***

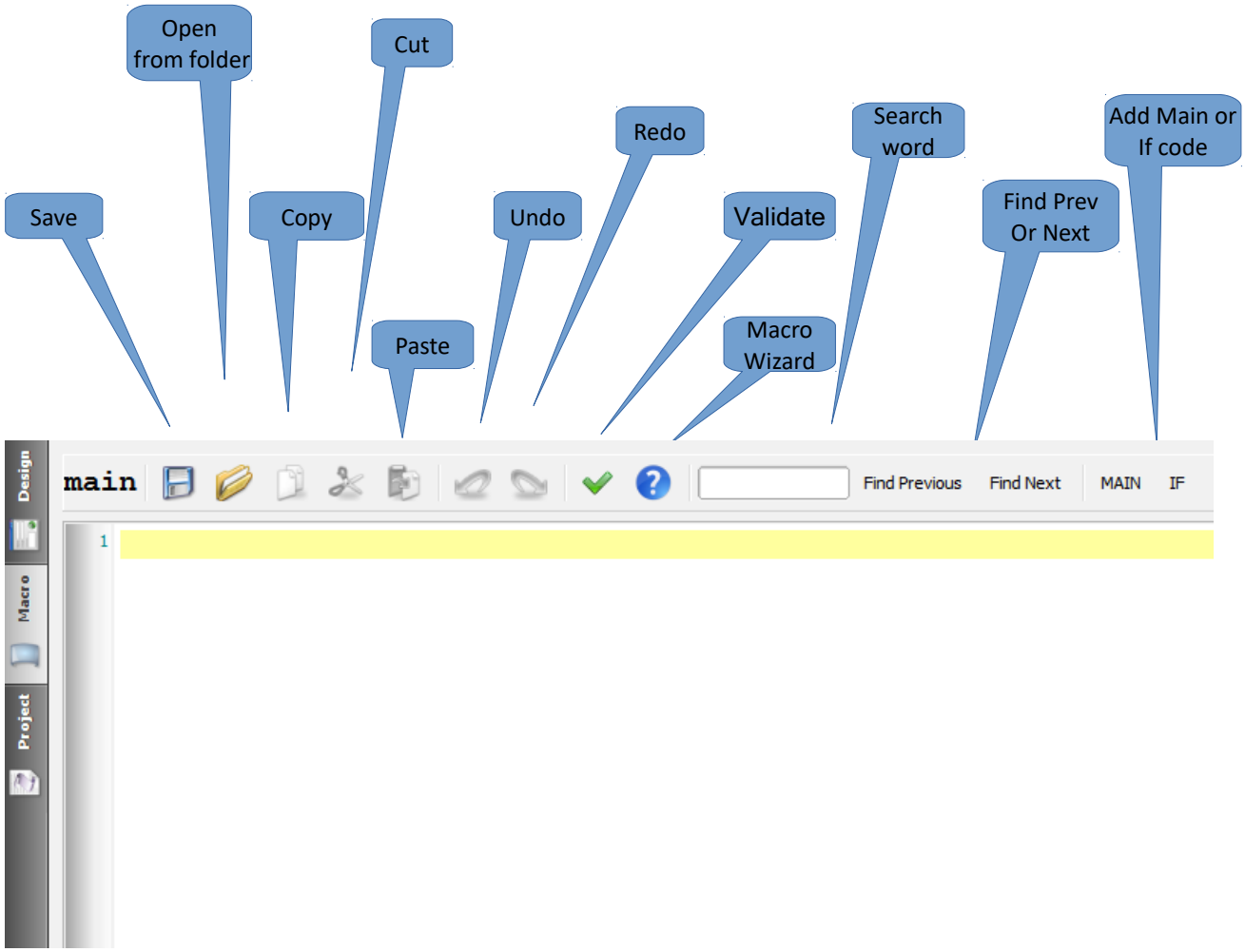
After adding languages and translations, those languages are listed under internal settings and language menu. For example, a button configurations is shown here and with this button, items' labels are going to be English.



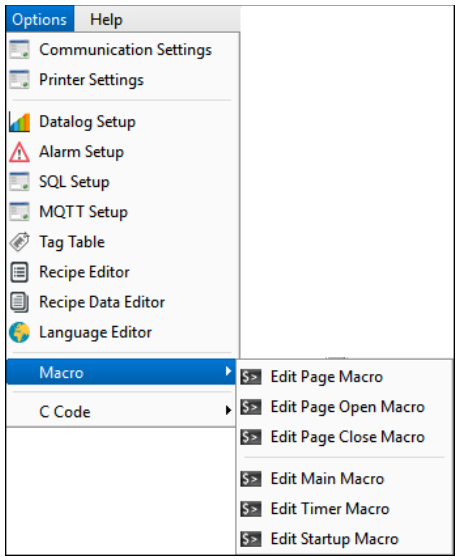
*Picture **45: Menu Bar → Options → Language Editor (Language Assignments)***

B.1.5.10. Macro Editing

Macro language is added for user convenience. The generated macros can be exported or macros can be transferred from the outside.



Picture 46: Menu Bar->Options->Edit Page Macro



*Picture **47: Menu Bar → Options (Macro)***

Macro shortcuts in Options menu

Edit Page Macro: Current page's macro codes can be edited.

Edit Page Open Macro: Current page's macro codes while opening can be edited.

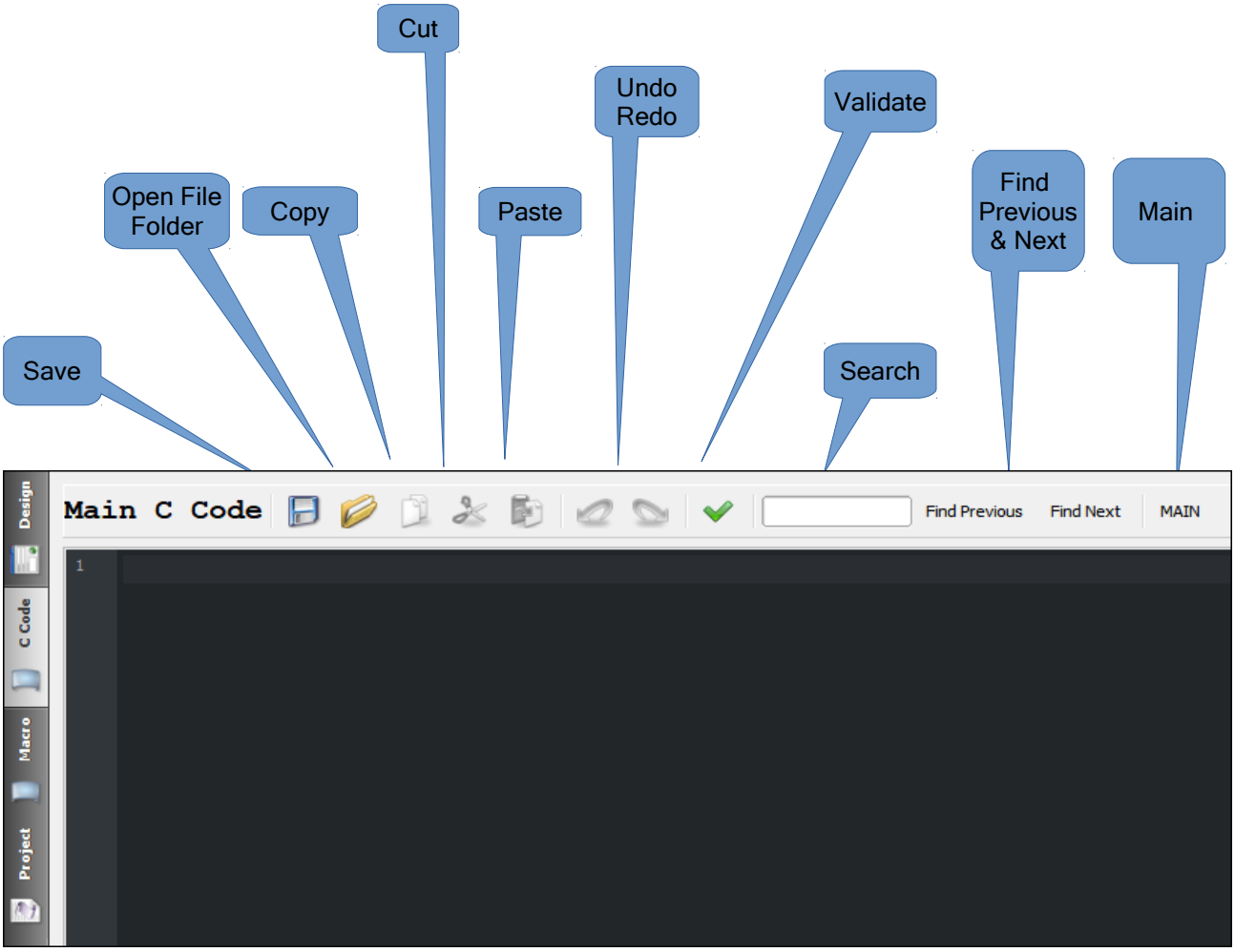
Edit Page Close Macro: Current page's macro codes while closing can be edited.

Edit Main Macro: Project's main macro codes can be edited.

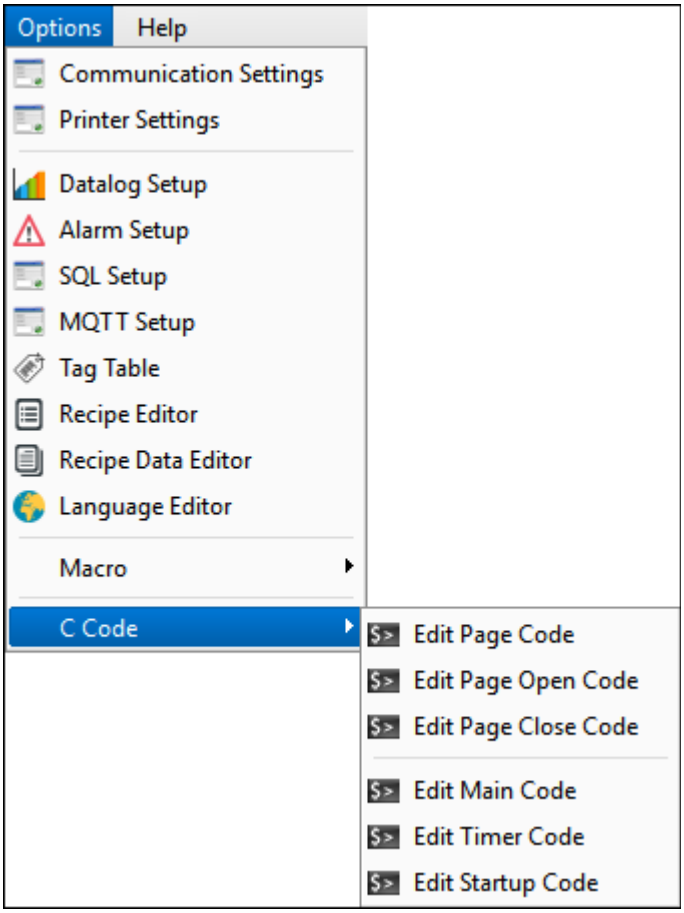
Edit Timer Macro: Project's timer macro codes can be edited.

Edit Startup Macro: Project's startup macro codes can be edited.

B.1.5.11. Editing C Code



Picture 48: Menu Bar>Options>Edit Page C Code



Picture 49: Menu Bar->Options (C Code)

C Code shortcuts in Options menu

Edit Page Code : Current page's C codes can be edited.

Edit Page Open Code: Current page's C codes while opening can be edited.

Edit Page Close Code: Current page's C codes while closing can be edited.

Edit Main Code: Project's main C codes can be edited.

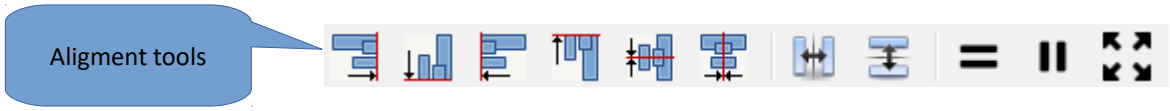
Edit Timer Code: Project's timer macro codes can be edited.

Edit Startup Code: Project's startup macro codes can be edited.

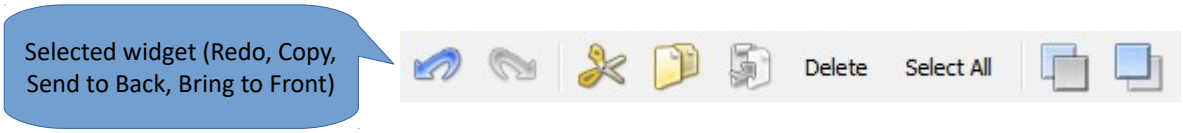
B.2. Tool Bar

Toolbar contains tools for project. The following tools.

Picture 50: Tool Bar->Form

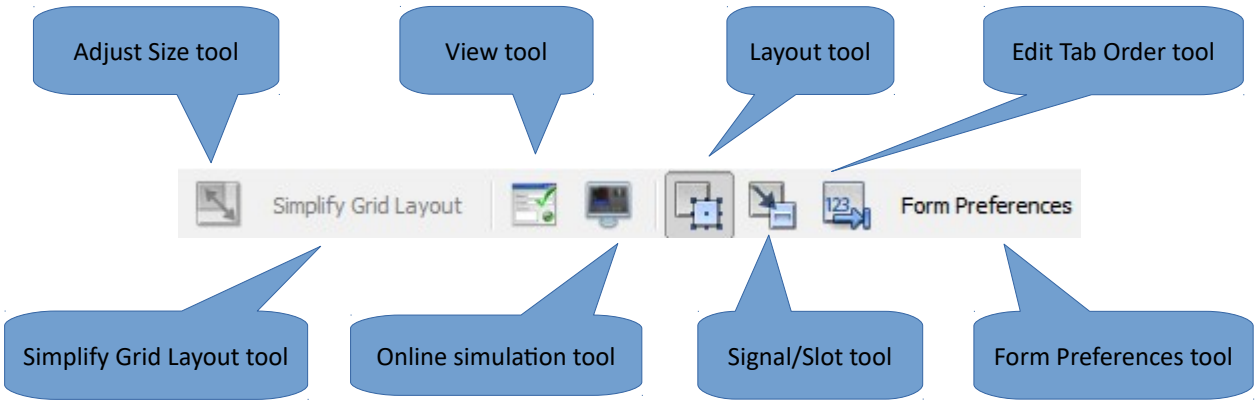


Picture 51: Tool Bar->Alignment



Picture 52: Tool Bar->Selected Widget

Picture 53:Tool Bar-> Group



Picture 54: Tool Bar

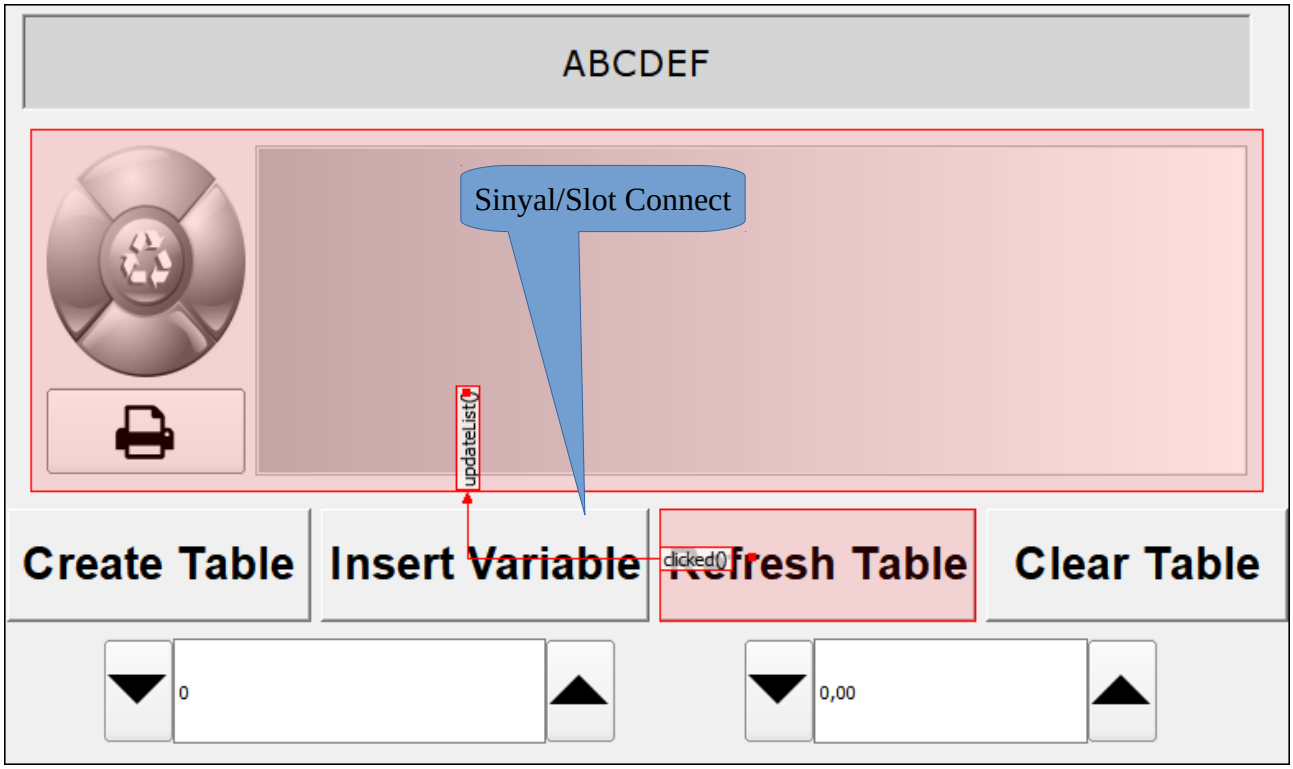
B.2.1. Loyout Tool

Layout tool; is used to ensure the size, position or replacement of elements in the work area. The Layout tool allows us to make any changes to the workspace.

B.2.2. Sinyal/Slot Tool

Signal slot tool; allows the elements to select which event to trigger with the help of ready-made functions.

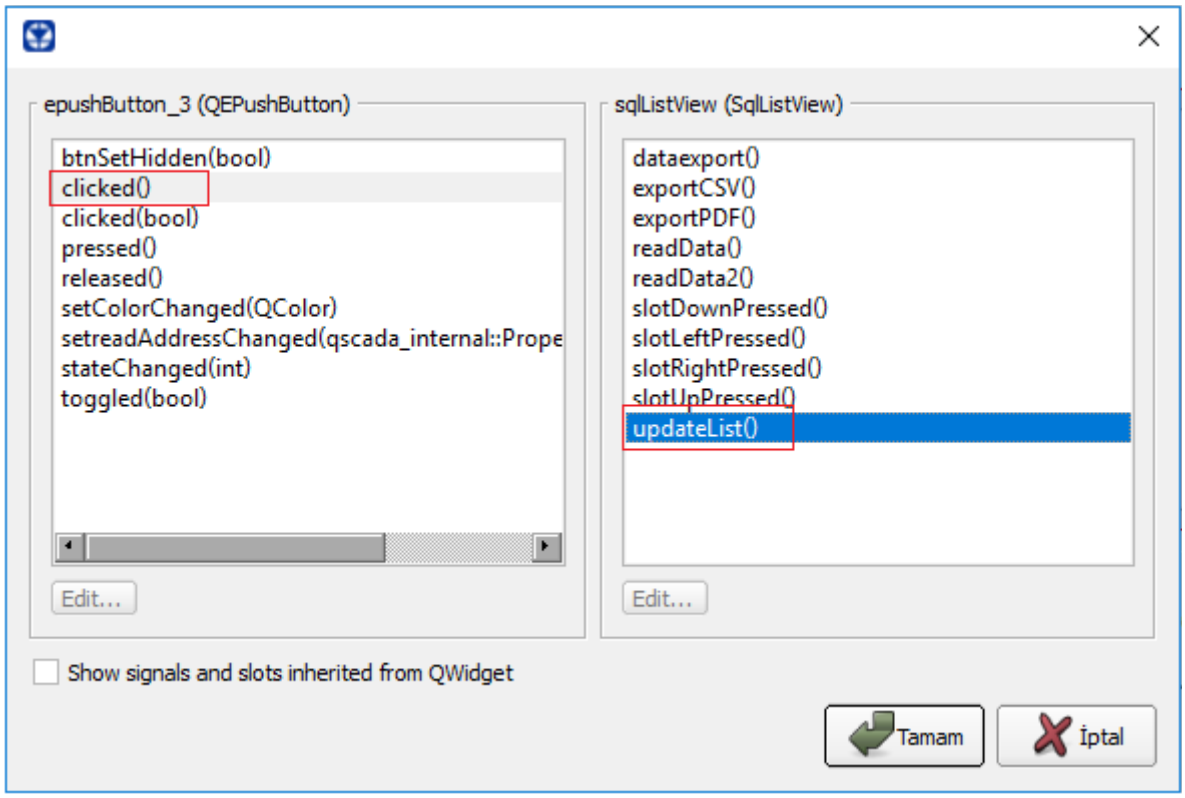
The update of the SQL List element given in the following figure is performed with signal / slot function.



Usage;

- Select tools sinyal/slot 
- The refresh button is held down with the mouse cursor and pulled over the SQL list element, when the cursor is released, the signal / slot function list appears.

- It is understood that when we send a clicked () signal to our button, the process that our SQL list should do is updateList (). After the selection, “OK” is pressed.



NOTE *: The shortest way of processing with macro code is to use Signal / Slot functions.

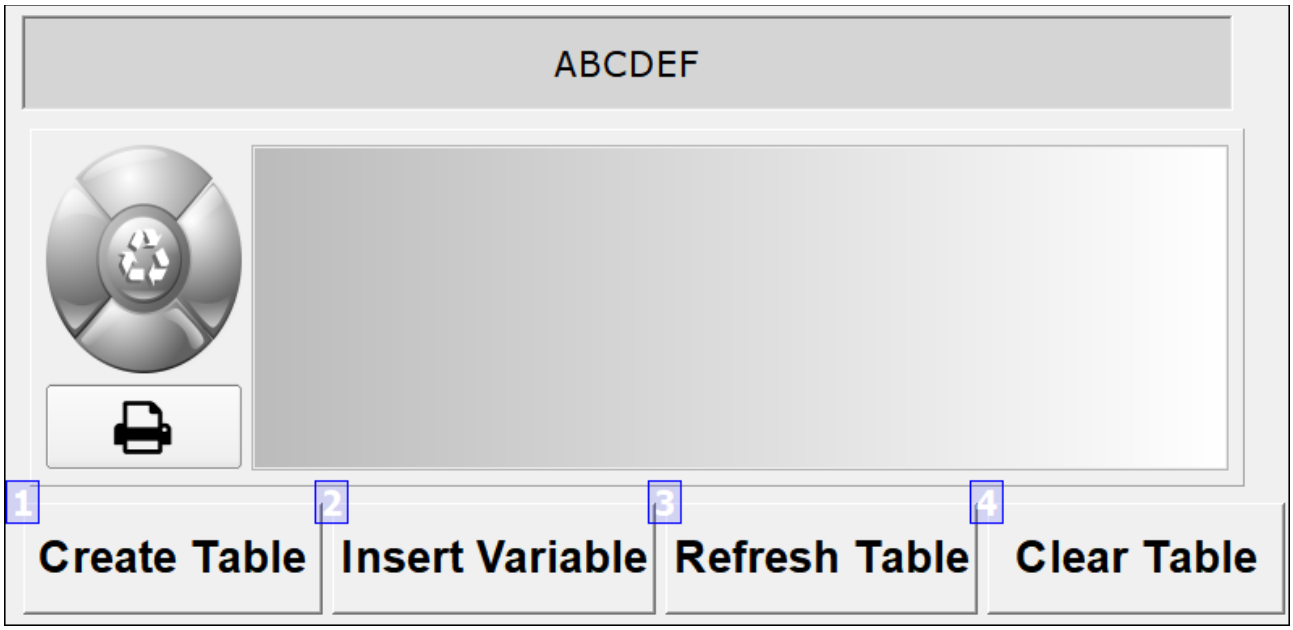
B.2.3. Edit Tab Order

Edit Tab Order; By pressing the tab key on the screen, the elements can be changed to other elements in this order.

Example;

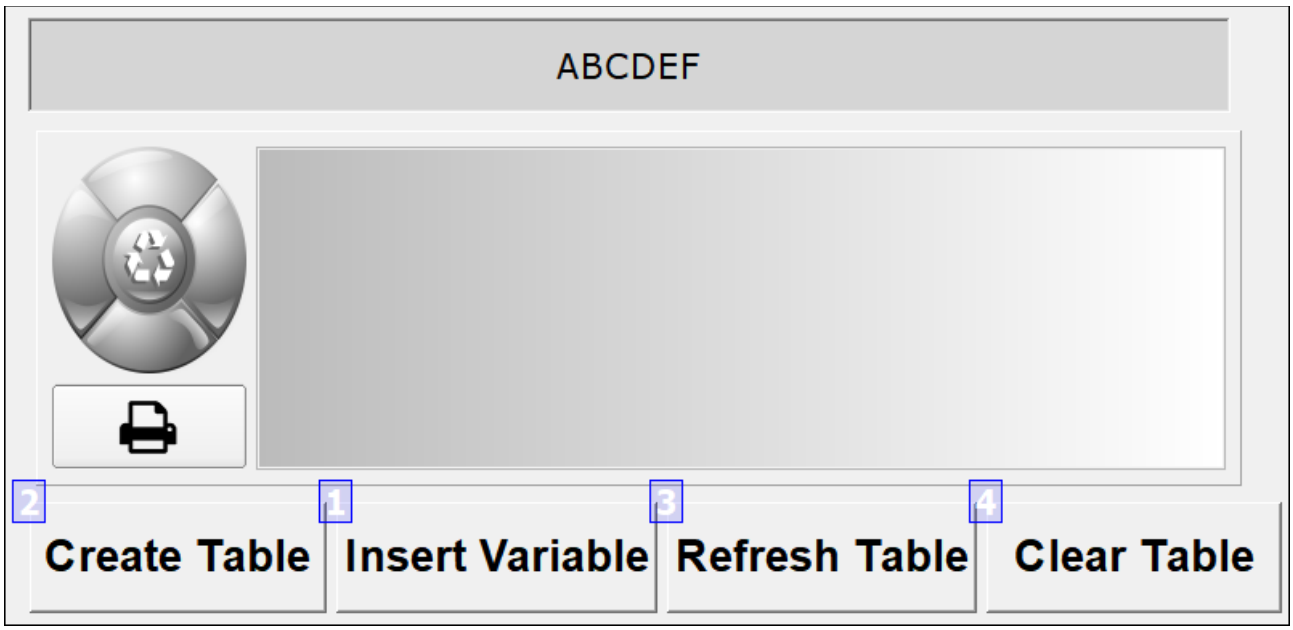
In the picture below, when Tab button is pressed :

Create Table -> Insert Variable -> Refresh →Clear Table



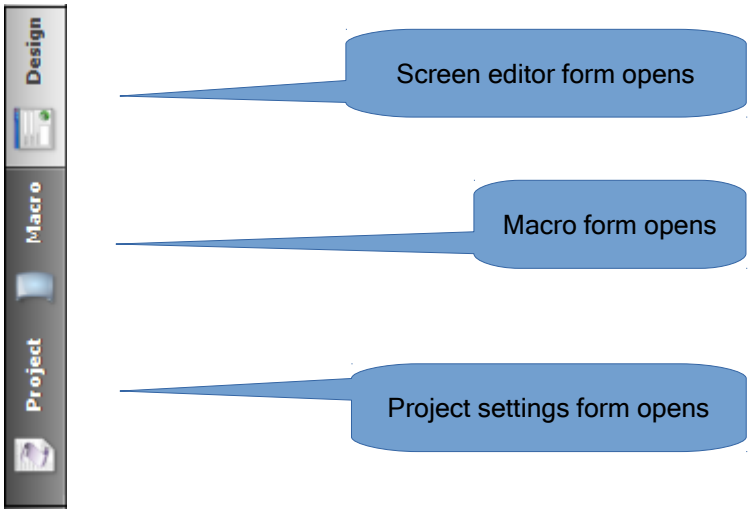
If the tab order specified in the following picture was used:

Insert Variable -> Create Table -> Refresh Table -> Clear Table



B.3. Side Bar

Sidebar is located to the left of the screen editor.



Picture 55: Side Bar

B.4. Element List

The elements that are available on the form page list.

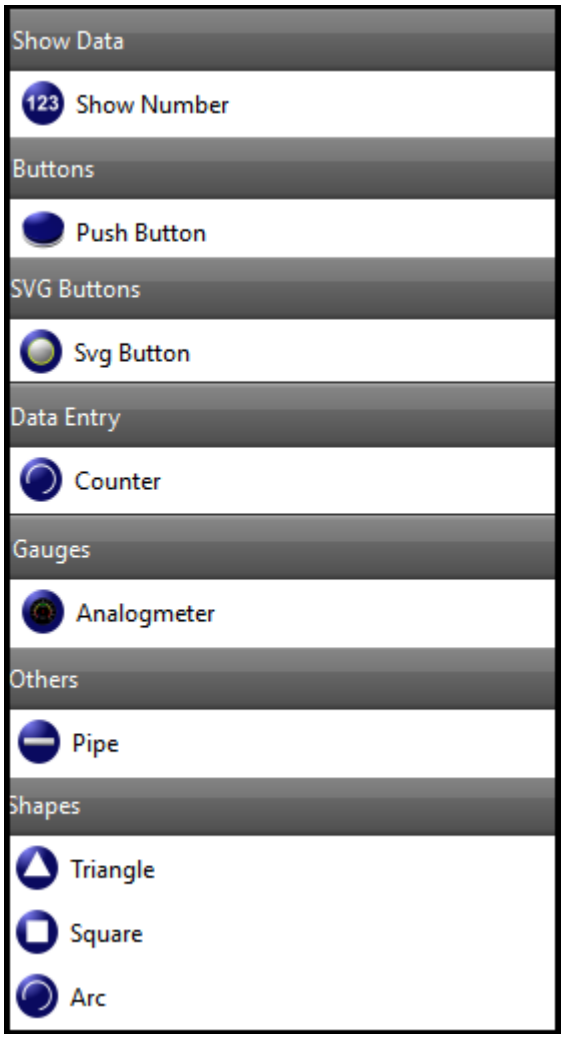
To use can the element tool;

- Select to the element tool.
- Hold down the left button of the mouse to drag the selected object to the form and release.
- Edit the settings using the properties table.

Element tool can search and can find from ***'Filter'*** field.

Element tools consist of 5 parts.

- Show Data
- Buttons
- SVG Buttons
- Data Entry
- Gauges
- Other
- Shapes

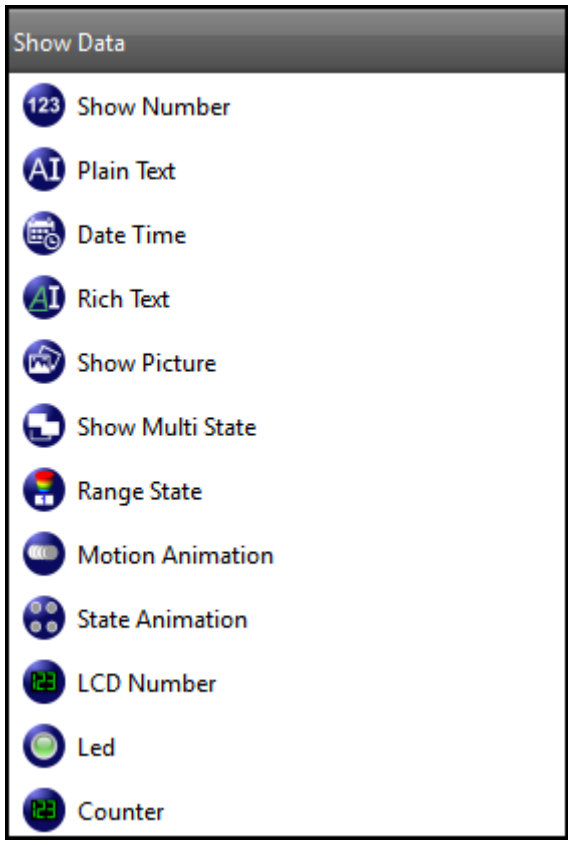


Picture 56: Element List

B.4.1. Show Data

The show data section can be use in the property, when the user want to display the a data, image, number or state.

Buttons divided into functions such as button type, status type, address function and page functions.



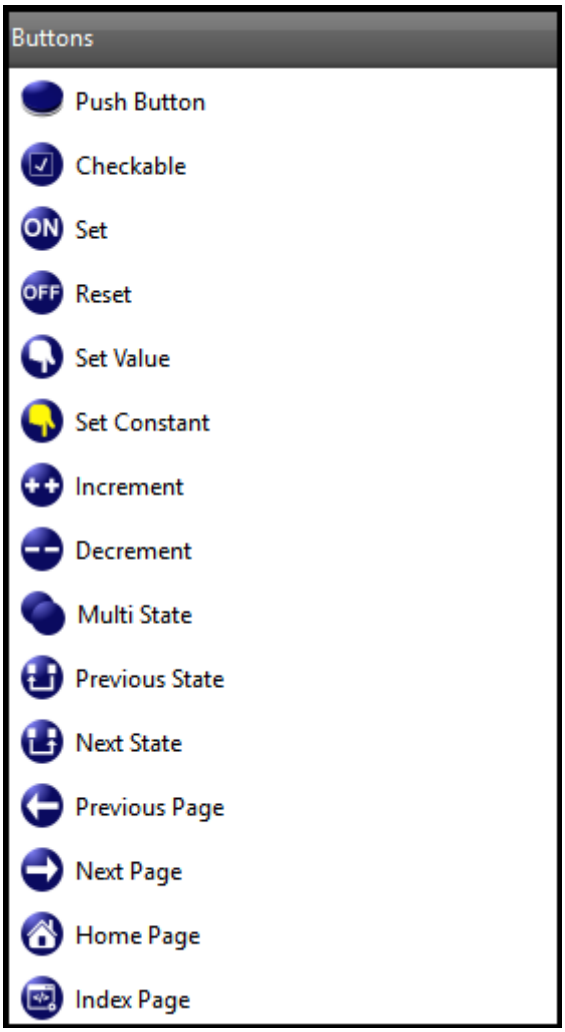
Picture 57: Show Data

Icon	Name	Function
	Show Number	Reads the specified address and display it as a number.
	Plain Text	Displays a text value on the form.
	Date/Time	Displays the date and time on the form.
	Rich Text	Displays a rich text on the form.
	Show Picture	Displays images the selected form in resource.
	Show Multi State	In the editor, displays the different values according to each state.
	Show Range	Displays the different values according to each range.
	Motion Animation	To use the motion animation, create more than one state. Set the desired field from property list for all status.
	State Animation	The state animation is displayed.
	Led	The color change is displayed according to the state of read address value
	Counter	The increase value or decrease value is displayed between the minimum and maximum value.

Table 1: Show Data

B.4.2. Buttons

Buttons divided into functions such as button type, status type, address function and page functions

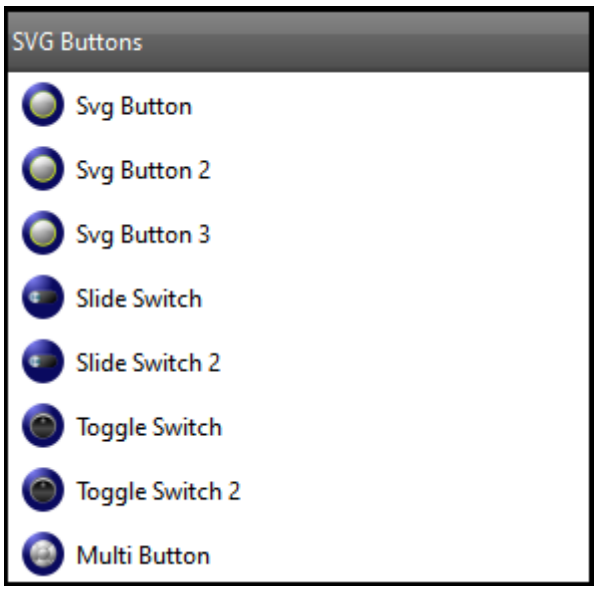


Picture 58: Buttons

Icon	Name	Function
	Push Button	When the push button is pressed, the state of address is ON and when the button is released, it is OFF.
	Checkable	When the push button is pressed, the state of address is ON and when the button is released, it is OFF.
	Set Button	When set button is pressed, the state of the address is ON.
	Reset Button	When set button is pressed, the state of the address is OFF.
	Set Value	When the button is pressed, the entered value will set at the defined address.
	Set Constant	When the button is pressed, the constant value will set at the defined address.
	Increment	When the button is pressed, a constant value will add to the value at the defined address. Then defined address will set added new value To define the constant value, go to **“constant value field”** in the set value section from property list.
	Decrement	When the button is pressed, fixed number is subtracted from the address value.
	Multi State	When the button is pressed, it moves to the next state or previous state. States are edited in the settings section.
	Previous State	When the button is pressed, it moves to the previous state.
	Next State	When the button is pressed, it moves to the next state.
	Previous Page	When the button is pressed, previous page is displayed.
	Next Page	When the button is pressed, next page is displayed.
	Home Page	When the button is pressed, home page is displayed.
	Go to Page	When the button is pressed, the page specified in the page index is displayed.

B.4.3. SVG Buttons

Svg buttons have the same function as the buttons and are named differently by the images.

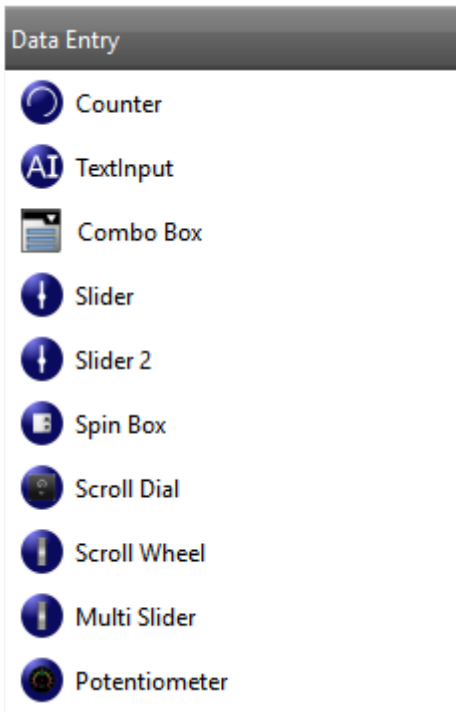


Picture 59: SVG Buttons

Icon	Name	Function
	SVG Button, SVG Button 2, SVG Button 3	It functions the same as the push button.
	Switch1, Switch2, Switch3, Switch4	It functions the same as the checkable.
	Multi Button	Up-down, left-right or center button functions can be used with one element.

B.4.4. Data Entry

The value change in the address is displayed on the screen.

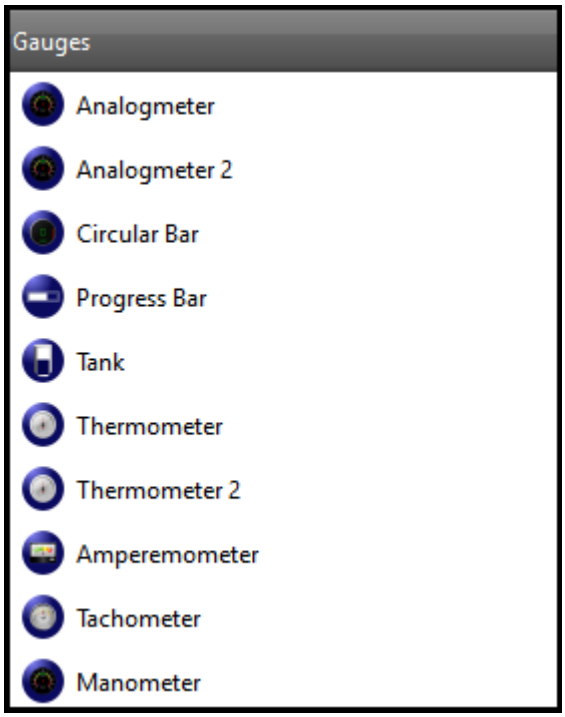


Picture 60: Data Entry

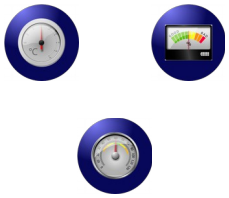
Icon	Name	Function
	Counter	Counter increases and decreases between the minimum and maximum values with buttons.
	Text Input	Text can be entered with this element.
	Combo Box	This is drop down list element.
	Counter, SpinBox	Determine the desired amount of increase and decrease between the minimum and maximum values is displayed.
	Slider, Slider 2, Scroll Dial	The desired amount of increase and decrease between the minimum and maximum values is displayed.
	Scroll Wheel, Multi Slider	It functions the same as the slider.
	Potentiometer	It functions the same as the analog meter.

B.4.5. Gauges

Change value displays is displayed with using data entry elements.

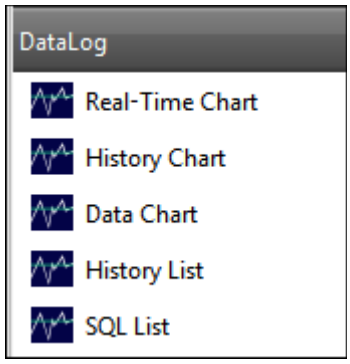


Picture 61: Gauges

Icon	Name	Function
	Analogmeter, Analogmeter 2, Circular Bar	Determine the desired amount of increase or decrease between minimum and maximum values is displayed. In the settings sections, upper limit and lower limit of value, the scala and the needle color are set.
	Progress Bar, Tank	The change of the value at reading address is displayed. Top limit and bottom limit can be colored from settings sections.
	Thermometer, Amperemometer, Tachometre, Manometer	It functions the same as the analogmeter.

B.4.6. DataLog

List and charts help to show logged data.



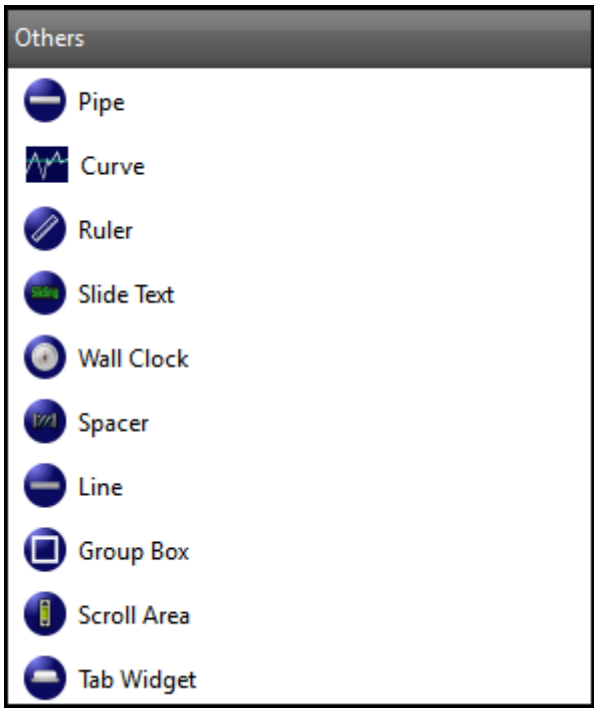
Picture 62: Element List->DataLog

Icon	Name	Function
	Real-Time Chart	It shows datalog chart momentarily.
	History Chart	It shows old datalog’s chart.
	Data Chart	Provides data graphical representation of datalog records.
	History List	It shows old datalog as list.
	SQL List	A list representation of database table records.

Table 2: DataLog

B.4.7. Others

Other elements can be used to display different functions on the screen.



Picture 63: Others

Icon	Name	Function
	Pipe	The motion in the pipe is displayed.
	Graph	The change of the value at the reading address is displayed graphically.
	Ruler	Used to its as units of measure of the value.
	Marquee	The text screen image is displayed by sliding.
	Clock	Displays the current time.
	Space	Leave a space between element tools.
	Line	Draws the line at the desired size on the form screen
	Group Box	It is provides a group box frame with a title.
	Scroll Area	It is provides a scrolling view onto another widget.
	Tab Window	It is provides a stack of tabbed widgets.

B.4.8. Shapes

The shape tools in the element list are used to triangle, square or draw.

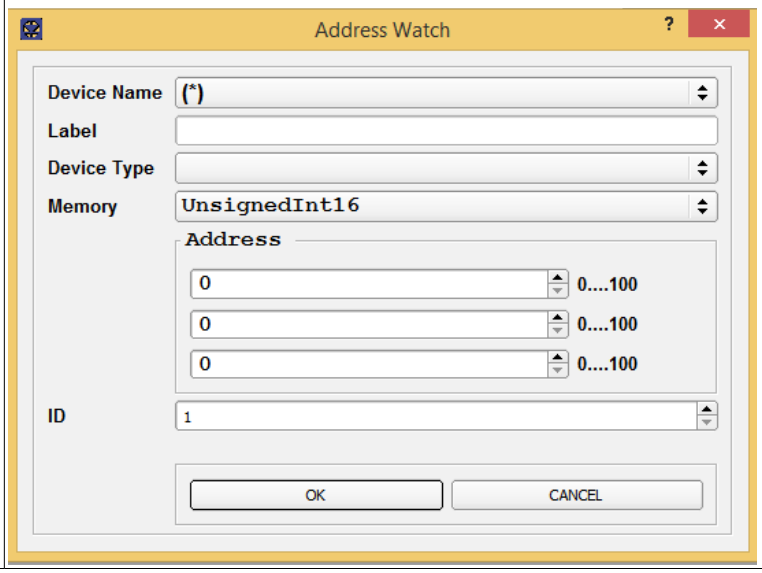
Picture 64: Shapes

Icon	Name	Function
	Triangle, Square, Arc	Triangle, square or arc drawings can be made.

B.5. Properties List

B.5.1. Address

In the project, when the show data, buttons, data entry and gauges tools is used, address field actives.

<table>
<tr><th>Name</th><th colspan="3">Function</th></tr>
<tr><td rowspan="12">Read Address,
Write Address ,
Hide Address</td><td colspan="3">Enter slaveID, to define the read address. DeviceID is the field in which the ID of the device is written.Default value defines between 0-255.</td></tr>
<tr><td colspan="3">Data type options are One of Bit, UnsignedInt16, SignedInt16, UnsignedInt32, SignedInt32, Float, UnsignedInt64, SignedInt64, Double.</td></tr>
<tr><td>Data Type</td><td>Minimum</td><td>Maximum</td></tr>
<tr><td>UnsignedInt16</td><td>-32,768</td><td>32,767</td></tr>
<tr><td>SignedInt16</td><td>0</td><td>65,535</td></tr>
<tr><td>UnsignedInt32</td><td>-2,147,483,648</td><td>2,147,483,647</td></tr>
<tr><td>SignedInt32</td><td>0</td><td>4,294,967,295</td></tr>
<tr><td>Float</td><td>1.8E-38</td><td>3.4E+38</td></tr>
<tr><td>UnsignedInt64</td><td>-9,223,372,036,854,775,807</td><td>9,223,372,036,854,775,807</td></tr>
<tr><td>SignedInt64</td><td>0</td><td>18,446,744,073,709,551,615</td></tr>
<tr><td>Double</td><td>2.2E-308</td><td>1.8E+308</td></tr>
<tr><td>...</td><td colspan="2">Click the icon on the left to open the address watching form. Device Name, Device Type, Label, Memory and ID field are configured from address tracking window.
 
</td></tr>
</table>

B.5.1.1. Address Watch

The device address field settings are configured in the address watching form..

<table>
<tr><th>Address Watch Field</th><th>Function</th></tr>
<tr><td>Device Name</td><td>Internal_io includes internal input-output devices.</td></tr>
<tr><td></td><td>Internal_memory, includes internal memory.</td></tr>
<tr><td>Device Type</td><td>If “internal_io” is selected in the device name field, there are 4 options for device type.
<table>
<tr><td>Device Type</td><td>Phrase</td><td>Memory</td><td>Range</td></tr>
<tr><td>Digital Input</td><td>$IX0.0</td><td>$Ixn.k</td><td>n :0-0 k :0-4</td></tr>
<tr><td>Digital Output</td><td>$QX0.0</td><td>$Qxn.k</td><td>n :0-0 k :0-5</td></tr>
<tr><td>Analog Input</td><td>$IW0</td><td>$IWn</td><td>n :0-1</td></tr>
<tr><td>Analog Output</td><td>$MW0</td><td>$MWn</td><td>n :0-1</td></tr>
</table>
If “internal_memory” is selected in the device name field, there are 5 options for device type.
<table>
<tr><td>Device Type</td><td>Phrase</td><td>Memory</td><td>Range</td></tr>
<tr><td>Volatile Memory</td><td>$0</td><td>$n</td><td>n : 0-65535</td></tr>
<tr><td>Non-Volatile Memory</td><td>$M0</td><td>$Mn</td><td>n : 0-65535</td></tr>
<tr><td>Volatile Memory Bit</td><td>$0.0</td><td>$n.k</td><td>n : 0-65535 k :0-15</td></tr>
<tr><td>Non-Volatile Memory Bit</td><td>$M0.0</td><td>$Mn.k</td><td>n : 0-65535 k :0-15</td></tr>
<tr><td>Internal Settings</td><td>$S</td><td>$Sn</td><td>n :0-65535</td></tr>
</table></td></tr>
<tr><td rowspan="3">Memory</td><td>The memory field includes bit, unsignedInt16, signedInt16, unsignedInt32, signedInt32, float, unsignedInt64, signedInt64, and double.</td></tr>
<tr><td>If “internal_io” is selected in the device name field, default value bit.</td></tr>
<tr><td>If “internal_memory” is selected in the device name field, default value unsignedInt16.</td></tr>
<tr><td>ID</td><td>Identity device</td></tr>
</table>

Table 3: Address Property->Address Watch

B.5.2. Data

When the data show, data entry, gauges and other tools are used, the data section actives.

<table>
<tr><th>Name</th><th>Function</th></tr>
<tr><td>Value</td><td>Read address value.</td></tr>
<tr><td>cFormat</td><td>Writes the code to display the desired format value.</td></tr>
<tr><td>fDigits</td><td>Defines for decimal numbers.</td></tr>
<tr><td rowspan="3">Gain
Offset</td><td>Sets value with mask.</td></tr>
<tr><td>Value=value * gain + offset
y=a.(x)+b
<table><tr><td>Value</td><td>Gain</td><td>Offset</td></tr><tr><td>x</td><td>a</td><td>b</td></tr></table>
Default gain value is ‘1.0’.
For example;
The gain value is ‘4.0’. When the LCD number element actual value is ‘10’,the displayed value is ‘40’.</td></tr>
<tr><td>Default the offset value is ‘0.0’.
For example;
The offset value is ‘1.0’. When the LCD number element actual value is ‘10’,the displayed value is ‘11’.</td></tr>
<tr><td>Rounding</td><td>If this option is selected, value round.</td></tr>
<tr><td>Minimum
Maximum</td><td>Limits are determined of the read address value.</td></tr>
</table>

Table 4: Data

B.5.3. Input

When the data input tools used, input section actives in the property list.

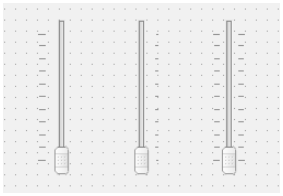
Name	Function
Single Step	Divides the interval between minimum value and maximum value into equal parts. The increase of the value is set.
Button Count	This field actives when the decrement/increment tools used.
Step Button 1-2-3	Defines the button name.
Value	This field actives when the slider 2 tool used. Default value is 50. Displays data value.
Page Step	Default value is 10.
Step Range	This field actives when the slider used. If the value in the step range field increases, the range size decreases and the step count decreases.
Inverted-Control	The Controllers reverses on the keyboard or mouse.
Tracking	If ***"tracking"*** is enabled, the data changes displays on the screen as the scroll button is moved.
Tick Position	This field actives when the slider tool used. Options are notick, tickabove, tickleft, tickbelow, tickright, tickbothsides.  ***Picture 65: Slider*** In Picture-36 above,the positioning field selected vertically and the positions of the steps selected as tickleft, tickright, and tickbothsides, respectively.  ***Picture 66: Slider*** In Picture-37 above, the positioning field selected vertical positions of the steps selected as notick, tickbelow ve tickabove respectively.

Table 5: Input Property

B.5.4. Value

When the gauges elements is used the value actives in the property list.

Name	Function
Minimum Value **Maximum Value**	Defines values of the limits.

Table 6: Value Property

B.5.5. General

The general section is active in all element tools in the property list.

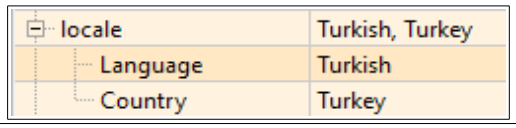
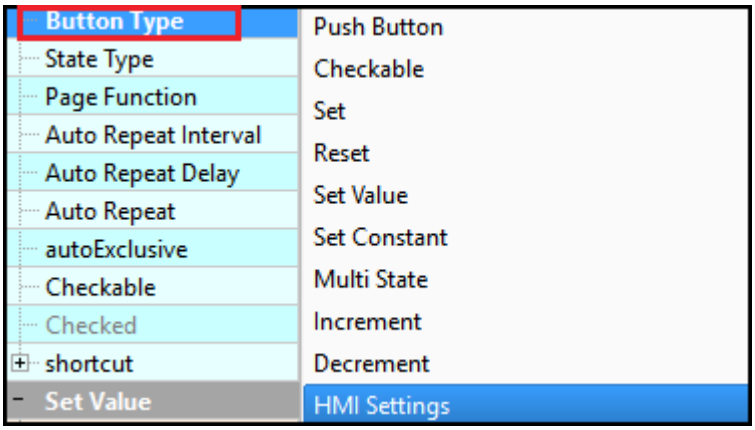
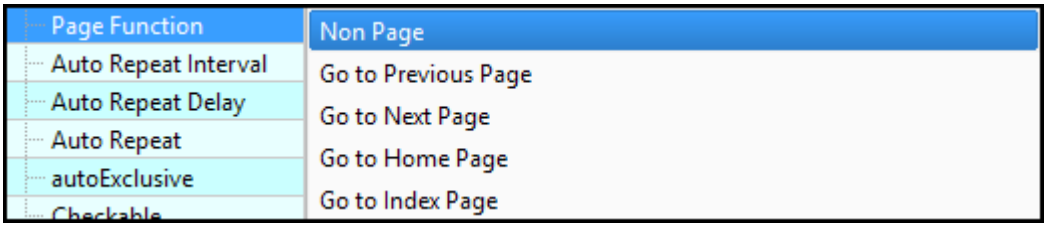
Name	Function
Enabled	If element tool is enabled, element tool can use.
Location	Determines the country where the devices are located 
Current State	Defines the state data of the selected element tool This data can be picture, code written in the style macro window, text, or any value.
nState	Defines The total number of states of the selected element tool
Reverse State	Changes the current state of the value in the defined address changes.
Position	Defines horizontal or vertical of the element tool.

Table 7: Value Property

B.5.6. Button

When the buttons and Svg buttons is used, the button sections actives in the property list.

Name	Function
Button Type	Button types are Push, Checkable, Set, Reset, Value Assignment, Fixed Assignment, Multiple Status, Increase, Decrease and HMI Settings. 
Page Function	The selected button gives the pagination function. Page functions are Non, Go to Previous, Go to Next, Go to Home, Go to Index. 
State Type	Durum types are Next, Previous State. 
Auto Repeat Invertal	Default value is 100 ms. It is used to set the interval time between two movements.
Auto Repeat Delay	Default value is 300 ms. It is used to set delay time for waiting the startup of PLC or external.
Auto Repeat	If ***"auto repeat"*** is enabled, auto repeat repeats the function using in the interval field value as a period.
Checkable	When the button is pressed, button displays checkabled.
Index Page	When the button is clicked, page number is written on which go to page is want to.

B.5.7. Special

When data entry, gauges and other element tools is used, the special section actives in the property list.

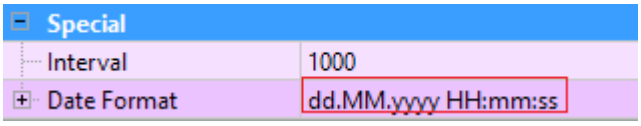
Name	Function	Used Elements
Display Type	Display options are LbNormal, LbMarquee.	Plain Text
	"LbNormal" displays text value.	
	"LbMarquee" displays text value how is marquee.	
Direction	If element tool is enabled, marquee element can uses.	Plain Text
	"RightToLeft", the text skips from right to left.	
	"LeftToRight", the text skips from left to right.	
Speed	Determines the text speed of the marquee element.	Plain Text, Pipe
	The water(fluid) object speed determines in pipe, if element tool is used.	
Pixmap	 Click the icon on the left to open the resources form and selects image or font type.	Show Picture
Scaled Contents	To resize the image.	Show Picture, Multi State, Range State, Motion Animation, State Animation
Range	Determines range count.	Range State
Date Format	Defines date and time at the desired format. The format example can enter as follows. 	Date Time
Interval	Default value is 1000ms. Updates the element tool.	Date Time
Movie Active	When element tool is enabled, the element tool used to show simple animation without sound.	Motion Animation
Percent Speed	Defines the speed value of the picture.	Motion Animation

Table 8: Special Property -1

Name	Function	Used Element
Segment Style	Segment style filled, framed and flat options are as follows.  *Picture 67: Lcd Number*	Lcd Number
Mode	Segment mode options are decimal, bin, hex, oct.	Lcd Number
SmallDecimalPoint	If the field is selected, the segment size decreases in a certain rate.	Lcd Number
Digit Count	Defines the number of digits of the data value.	Lcd Number, Circular Bar
IntValue	It is value at on the screen	Spin Box
Decimals	Defines the number of digits of the decimal part of the data.	Spin Box Counter
Keyboard Tracking	If the ***"keyboard tracking"*** are selected, the data change displays when button is clicked.	Spin Box
	If the ***"keyboard tracking"*** isn't selected, the data change wont be displayed while the button is clicked. Displays the value at the end of the motion.	
Prefix, **Suffix**	Adds the text of the displayed data at front or end.	Spin Box, Thermometer 2, Manometer
correctionMode	If an invalid value is entered in the data field, the data to be assigned to that value is specified as one of the options. The correction options are nearest and previous value.	Spin Box
accelerated	The process varies with acceleration.	Spin Box
correctionMode	If an invalid value is entered in the data field, Defines the mode to correct an Intermediate value. The correction options are nearest and previous value.	Spin Box
specialValueText	It can use as text display.	Spin Box

Table 9: Special Property -2

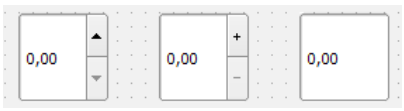
Name	Function	Used Elements
buttonSymbols	Button style options are UpdownArrow, PlusMinus, NoButtons as follow.  ***Picture 68:Increase / Decrease Value-2***	Spin Box
Read Only	If it is enabled, no action(edit) can not be taken on the element tool.	Spin Box, Spin Box 2
Wrapping	If the field is selected return value.	Spin Box, Spin Box 2
Frame	Adds the frame at the element tool.	Spin Box
Enable Numeric Indicator	The field that displays the data change on the screen and writes the data value to the screen.	Analogmeter, Circular Bar
	If the digital meter is not selected, the data change hide. The point is added in place of the indicator. It shows is Picture-40.	
	If the enable ***"numeric indicator"*** is not selected in the circular bar, the data value hides.	
Start Angle **End Angle**	When the start and end angle of the arc is specified, the arc display arranges.	Tachoometer, Analogmeter, Circular Bar, Termometre, Amperemeter,
Step	The value range between minimum and maximum is divided by the value in the step field. Creates steps.	Analogmeter Circular Bar
Steps 2	Divides between two steps equal to the value in the intermediate step field. Creates steps 2.	Analogmeter

Table 10: Special Property -3

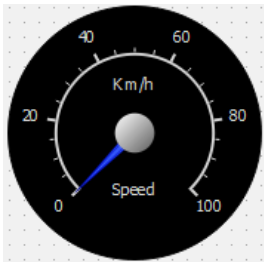
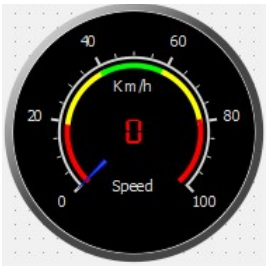
Name	Function	Used Elements
Units	When the gauge element tool is used, this field actives. Determines the unit of the element tool value.	Analogmeter
Enable Crown	When the gauge element tool is used, this field actives. If the gauge is selected, displayed as Picture-40. If the gauge isn't selected, displayed as Picture-41.  *Picture 69: Analogmeter2*  *Picture 70: Analogmeter*	
enableAreas	If the ***"enable areas"*** is enabled, can colors the step ranges. If the area is enabled as displays Picture-40. If the area isn't enabled as displays Picture-41.	Analogmeter
area1-2-3-4-5 begin	Defines the initial values of the step ranges	Analogmeter
area1-2-3-4-5 end	Defines the end values of the step ranges	Analogmeter
area1-2-3-4-5 color	Defines the color of the step ranges	Analogmeter
Circular Bar Enabled	It is the tool in Picture-42 that displays the data exchange.  *Picture 71: Circular Bar*	Circular Bar

Table 11: Special Property -4

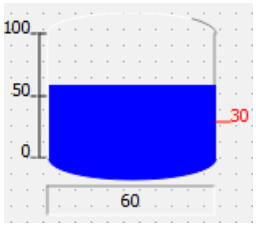
Name	Function	Used Elements
Threshold	Defines the beginning of the threshold value. The image of the down limit arc as in Picture-42 above is red.	Analogmeter, Circular Bar, Tank
Bar Size	Defines size of the circular bar.	Circular Bar
Cover Glass Enabled	When circular bar is used, this field actives. It shines on circular bar.	Circular Bar
Enable Threshold	If the ***"enabled threshold"*** is enabled, it displays on the screen. If the ***"enabled threshold"*** isn't enabled, it hides.	Circular Bar
NumTicks	If the tank element tool is used, this field actives. Divides the value between the minimum and maximum values as shown in Picture-43.  ***Picture 72: Tank***	Tank
showCurrentDate/ Time	If the wall clock is used, this field actives. If this field is selected, the current date/time displays on the screen.	Wall Clock
Date/Time	If the wall clock is used, this filed actives. If showCurrentDate/Time isn't selected, the desired date / time value sets.	Wall Clock
Day Font, Date Font, Time Font, Digit Font	Sets the font of the object.	Thermometer, Manometer, Wall Clock

Table 12: Special Property -5

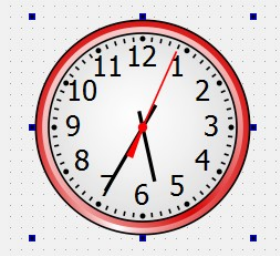
Name	Function	Used Elements
DigitColor, **DateColor,** **DayColor,** **TimeColor**	Sets the color of the object. The wall clock tool shows in Picture-44 below.  *Picture 73: Duvar Saati*	Wall Clock
digitOffset, **dateOffset,** **dayOffset,** **timeOffset**	Sets the distance from the center of the object.	Thermometer, Manometer, Wall Clock

Table 13: Special Property -6

B.5.8. Visual

Visual properties are used in all element tools.

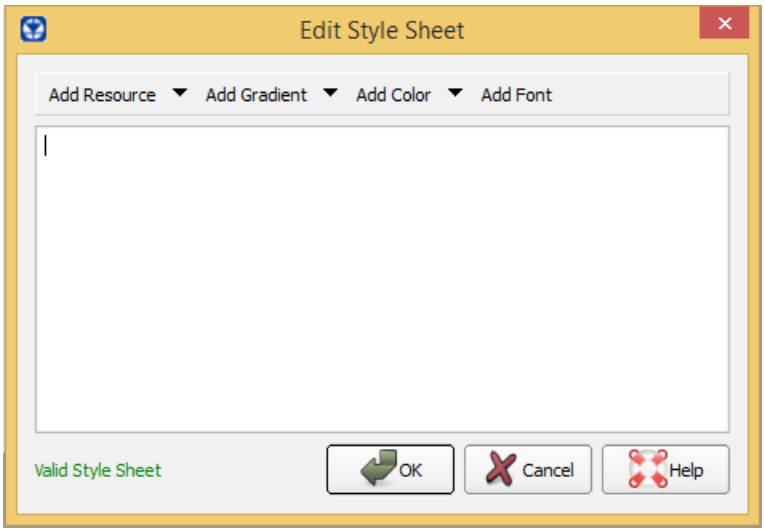
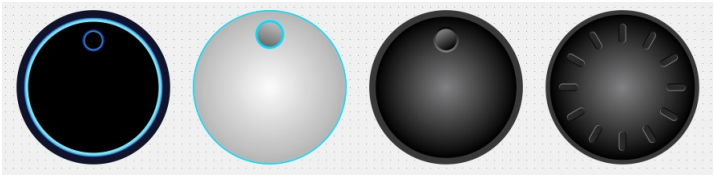
Name	Function
Visible	If the button tool is used, this field actives. If the ***"visible"*** field is selected, it displays or hides of the element tool.
Style Sheet	... When the icon is clicked on the left, style edit form open. For the element tool view, user can add source image, gradient, add font option. Style code can add to the area where cursor is located.  *Picture 74: Style Sheet->Edit Style Sheet*
Frame Style	If the potentiometer tool is used, this field actives.  *Picture 75: Potantiometer Options*

Table 14: Visual Property -1

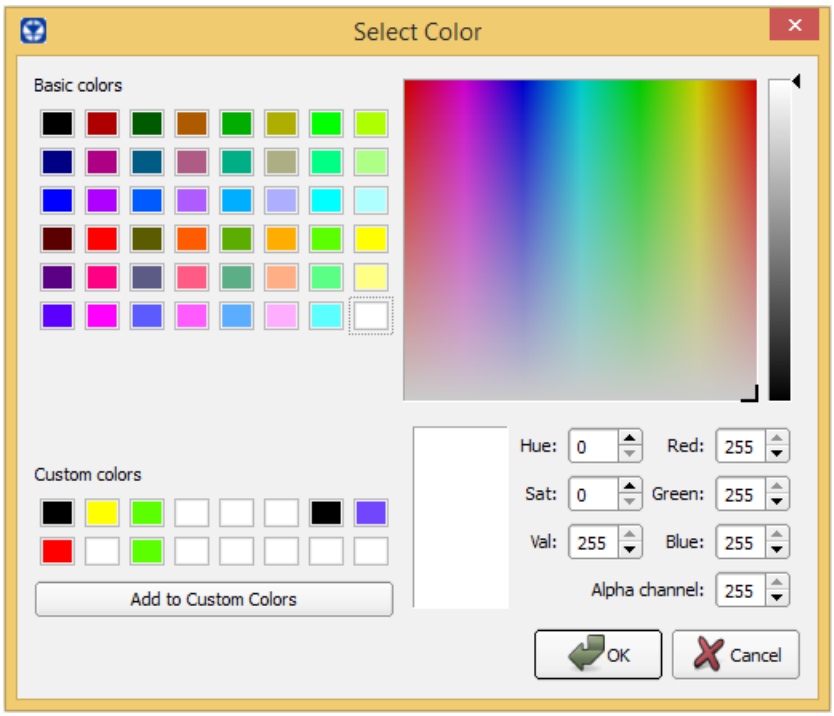
Name	Function
Text	Displays the desired text.
Label	Element tool is name.
LabelPosition	The label position are the left, right, top, bottom or center.
Background Color	If the ***"flat"*** is enabled in the general section, background color sets of the button tool.  ***Picture 76: Background Color->Select Color***
Foreground Color	When the analogmeter is used, this field actives. Foreground color sets of the analogmeter tool.
Font Style	When ***"intermittent"*** is selected, text displays with fixed range.
	When ***"sliding"*** is selected, marquee displays.
Font Type	Selects the font types.
Font Color	Selects the font color.
pixlbPicture	... If the button tool is used, this field actives. When the icon is clicked on the left, style edit form opens.
Picture Alignment	The picture alignment options are horizontally and vertically.

Table 15: Visual Property -2

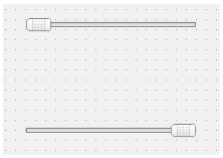
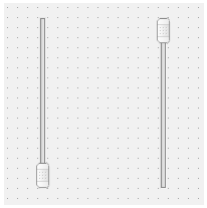
Name	Function	
Flat	To upload the desired image, the **“flat”** field must enabled.	
Icon Size	Defines the width and height values of the icon.	
Icon		When the icon is clicked on the left, style edit form opens.
Word Wrap	If this field is enabled, the text is wrapped where necessary at word-breaks.	
Focus	Focus type options are Nofocus, Tabfocus, Clickfocus, Stringfocus, Wheelfocus.	
Font Format	Font format options are Richtext, Plaintext, Logtext, Ototext.	
Text Direction	If the multi slider tool is used, this field actives. Text direction options are TopToBottom, BottomToTop.	
Orientation	Selects the leftoright or bottomtotop the slider button direction.  **Picture 77: Slider**  **Picture 78: Slider**	
Text Visible	If the multi slider or progress bar tools are used, this field actives. The value displays on the screen as text format.	
Aligment	Alignment options are vertical alignment and horizontal alignment.	
	Text is alignment at the left, right or center on vertically.	
	Text is alignment at the top, bottom or horizontal on horizontally.	
Title	If analogmeter 2 and group box is used, this field actives. Text is displays on the screen.	
Margin	The width of the margin.	
Indent	Text indent in pixels.	
Scroll Direction	Scroll button direction options are TopToBottom, BottomToTop.	
Tank Color		If the tank tool is used, this field actives. When the icon is clicked on the left, fluid color selects at the tank tool.

Table 16: Visual Property -3

Name	Function
Show Navigation	Show / hide navigation feature in list and graphic elements.

B.5.9. Geometry

When the gauges and other element tools are used, this section actives.

Name	Function
Geometry	The coordinates of the selected element are determined according to the position on the page.
Size Policy, Base Size, Size Increment, Minimum Size, Maximum Sİze	Determines the minimum and maximum size of the selected element tool.

Table 17: Geometry Property

B.5.10. Set Value

When the button element is used, this section actives in the property list.

Name	Function
Step Value **Minimum** **Maximum**	Minimum and maximum value is determined of the percent value.
Constant Value	Constant value is set at the element tool.

Table 18: Set Value Property

B.5.11. Macro

When the button element is used, this section actives in the property list

Name	Function
Before **Pressed** **Release**	When the before, pressed or release is clicked, opens ***'edit makro form'***.
Script Select Macro	In SQL, it retrieves the data drawn with the query ile SELECT.

Table 19: Macro Property

B.5.12. Frame

When the data entry, shapes and other tools are used, the frame section actives.

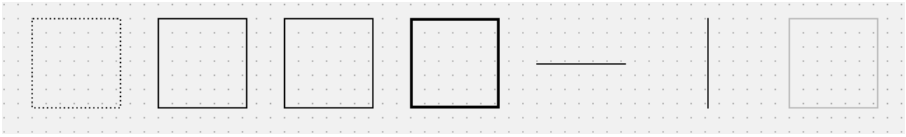
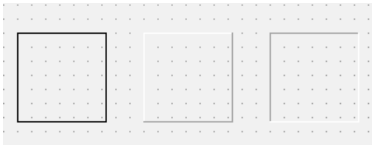
Name	Function
Frame Shape	The options are NoFrame, Box, Panel, WinPanel, Hline, Vline, StyledPanel. The frame shapes show in the following order.  ***Picture 79: Frame Shape***
Frame Shadow	Options are Plain, Raised, Sunken. The frame shadows show in the following order.  ***Picture 80: Frame Shape***
Line Width	Determines the bold of the frame.
Mid Line Width	The field that draws a line horizontally on the element tool and determines the line width.

Table 20: Frame Property

B.5.13. Shape

When the shape tools are used, the shape section actives in the property list.

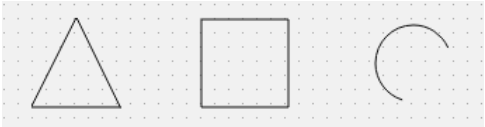
Name	Function
Line Color **Ground Color**	The user determines of the desired color or background color for the shape tool.
Line Width	The user determines of the line width for the shape tool.
Shape	There are options square, arc and triangle of the shape element tool. If arc is drawn, start angle and end angle must determine.  ***Picture 81: Shapes***
Start Angle **End Angle**	If the arc is used, this field actives. An arc can draw determining the start and end angles.

Table 21: Frame Property

B.5.14. Line

Only ruler element tool uses this section.

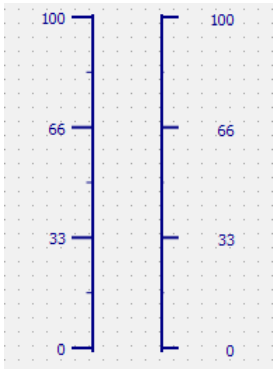
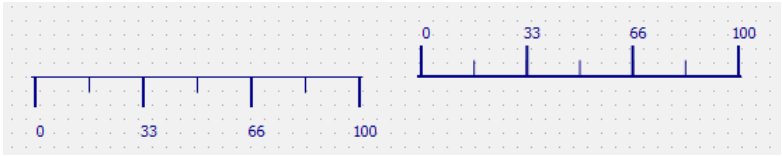
Name	Function
Rotation	The rotation options are horizontal, vertical, rotation_180 and rotation_270.  *Picture 82: Ruler* The above picture, the rotation of the ruler set to standard and rotation_270 according to the order of the picture.  *Picture 83: Ruler* The above picture, the rotation of the ruler set to rotation_90 and rotation_180 according to the order of the picture.

Table 22: Line Property

B.5.15. Pipe

When the pipe tools are used, the pipe section actives in the property list.

Name	Function	
Background Color		When the icon is clicked on the left, the water(fluid) color determine in the pipe tool.
Rotation	The user select the element tool direction. The options are standard(horizontally) and rotation_90(vertically).	
State	This option determines the state of the water(fluid).	
	If the "***disable'"***is selected, the water(fluid) does not move in the pipe.	
	If the "***enable"*** is selected, the water(fluid) moves in the pipe.	
vDirection	The direction of the water(fluid) can selects from left to right or from right to left. The rotation is horizontal.  ***Picture 84: Pipe***	
	The direction of the water(fluid) can selects from top to bottom or from bottom to top.The rotation is vertical.  ***Picture 85: Pipe***	

Table 23: Line Property

B.5.16. Scale

When the tachometer tools are used, the scale section actives in the property list.

Name	Function
Needle Origin x **Needle Origin y**	If the tachometer is used, this field actives. The position define of the needle on the element tool.

Table 24: Scale Property

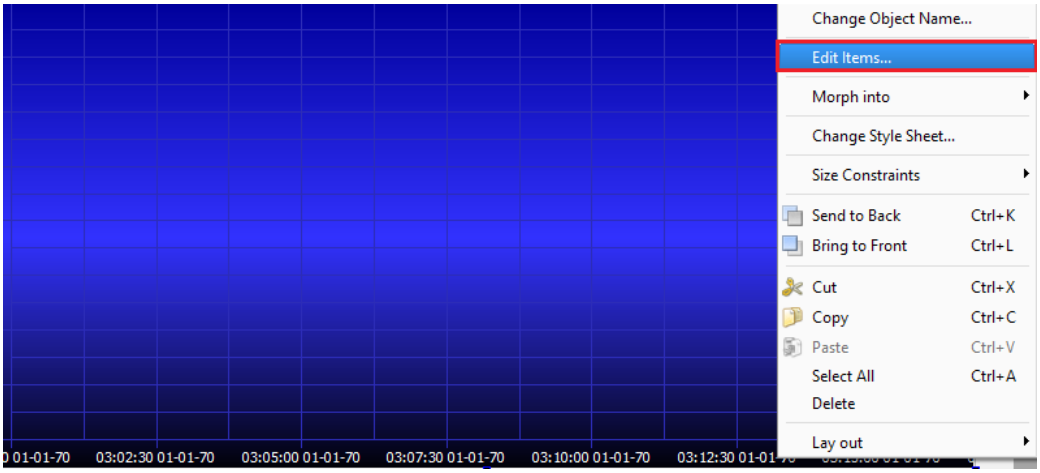
B.5.17. Chart

When the chart element tool is used, the chart section actives in the property list.

Name	Function	
Top Background Color, **Bottom Background Color,** **GridColor**		The user can make custom color selection on the appeared color picker dialog.
Period	Default values is 1000ms. Graph channels sampling interval time.	
Position	Scrolls the active visible area.	
Size	Default value is 10000. If the size value increases, it will read more than the X-axis value.	
Zoom	If the zoom value increases, the graphic will display in detail.	
Xmesh,	This field is half of the number of grids on the horizontal.	
yMesh	This field is half of the number of grids on the vertical.	
xSubMesh	The ***'xSubMesh'*** divides between both grids on the horizontal.	
ySubMesh	The ***'ySubMesh'*** divides between both grids on the vertical.	
showGrid	If ***"showGrid"*** is enabled, the vertical and horizontal grids will display. If ***"showGrid"*** is not enabled, the vertical and horizontal grids will hide.	
showScale	If the display is selected, the data values will display at the horizontal.	
showLegend	If the display is selected, the text title will display of the values. To define the ***'text title'***, right click on the cursor while the cursor is over the chart element tool. More then click the ***'edit items'*** title from open	

	window.
Antialiasing	Enables the ‘antialising’ feature.

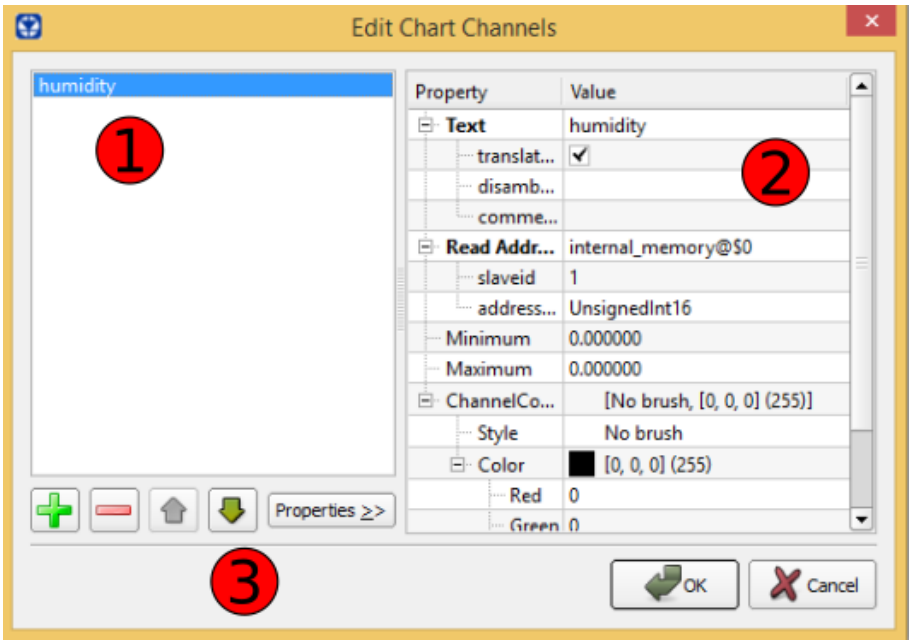
Table 25: Chart Property



Picture 86: Chart

To edit the chart element tool;

- Right click on the cursor, while the cursor is over the chart element tool.
- Click the ***'edit items'*** title from opened window.
- A new form will open as below.



Picture 87: Chart->Edit Chart Channels

- This window edits curves(channels) in the chart.
- On number field 1, curves list and Curve name selects the desired.
- On number field 2, properties edit of the selected chart.
- On number field 3, create a new curve, delete the selected curve, move the curve up or down the list.

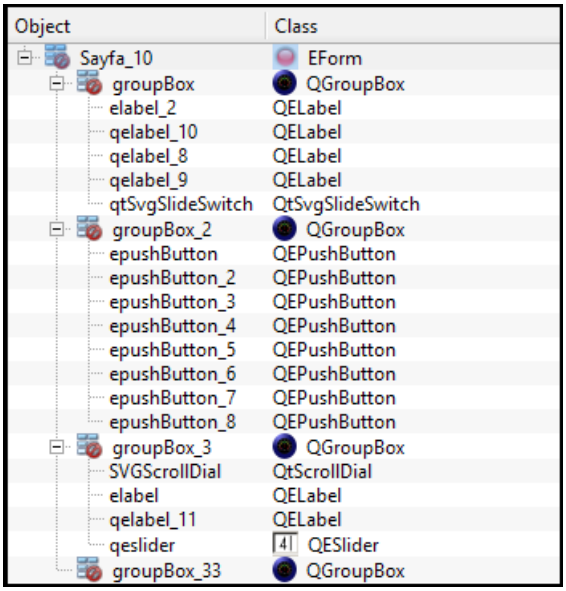
B.5.18. Barcode

These properties are used in the Barcode element.

<table>
<tr><th>Name</th><th>Function</th></tr>
<tr><td>Barcode Type</td><td>It is the type selection property of the Barcode element.
Example : CODE 128, QRCODE, EXCODE etc.</td></tr>
<tr><td>Character Length</td><td>Specifies the maximum length of the character length of the read or entered value of the element. Characters with more than this number cannot be entered or displayed.</td></tr>
<tr><td>Show Text</td><td>Show / Hide the text on the element.</td></tr>
<tr><td>Print Scale</td><td>The size of the element to be reduced can be adjusted.</td></tr>
<tr><td>Height</td><td>Uses to adjust the line height within the barcode element.
<table>
<tr><th>Height</th><th>Output</th></tr>
<tr><td>barcodeHeight 20</td><td>  </td></tr>
<tr><td>barcodeHeight 40</td><td>  </td></tr>
</table></td></tr>
<tr><td>White Space</td><td>Used to adjust the amount of space left of the barcode element from the edges.</td></tr>
<tr><td>Border Type</td><td>Adds a frame to the Barcode element. (Ribbon, Box, etc.)</td></tr>
<tr><td>Border Width</td><td>Used to adjust the size of the frame edges.</td></tr>
</table>

B.6. Element Tree

Lists the used element tools as tree on the form screen.



Picture 88: Element Tree

C) Macro

For more information, you can examine Macro Wizard window.

C.1. Variable Types

Operator	**global**
Comment	Defines a global variable to use in all of macro code.
Example	`global var1;` `//A global variable named var1 was created.` `var1 = 5;` `//Variable 5 is assigned to variable var1`

Operator	**local**
Comment	Defines a variable to use in the function it contains.
Example	`local var1;` `// A local variable named var1 was created.` `var1 = 10;` `//Variable 10 is assigned to variable var1.`

Operator	**$n**
Comment	Volatile variable specifies at assigned in the internal memory
Example	`$10` `//The volatile variable number 10 is` `//specifies to address`

Operator	**$Mn**	
Comment	Non-volatile specifies at addressing in the internal memory.	
Example	`$M10`	`//The non-volatile variable number 10 is //specifies to address`
Operator	**{device name}device id@n**	
Comment	Specifies the variable assignment at the desired address of the connected device.	
Example	`AMF}1@10`	`//This usage specifies the address 10 of the // device named Amf with device ID 1.`

C.2. Arithmetic Operators

Operator	**+**	
Comment	Used to the sum of two values.	
Example	`var1 = 10 + 20;`	`//Adds 10 to 20 and assigns the result to //variable var1.`

Operator	**-**	
Comment	Used to the substract of the two values.	
Example	`var1 = 20 - 10;`	`//Subtracts the value of 10 from 20 and //assign the result to variable var1`

Operator	*****	
Comment	Used to multiplication of the two values.	
Example	`var1 = 10 * 20;`	`//Multiplies the value 10 by 20 and assign` `//the result to variable var1`

Operator	**/**	
Comment	Used to division of the two values.	
Example	`var1 = 20 / 10;`	`//Divides 20 by 10 and assign the result to` `//variable var1`

Operator	**=**	
Comment	Used to assign value at variable or assign value of the other value at variable.	
Example	`var1 = var2`	`//Assign the value of var2 to var1`

Operator	**sqrt(n)**	
Comment	Used to find square root of the value.	
Example	`var1 = sqrt(9);`	`//The square root of the value 9 is assigned` `//to var1.`

C.3. Boolean Operators

Boolean operators are used with the if and while operators and return the comparison results as true or false.

Operator	**<**
Comment	Returns true if the value to the left of the operator is less than right, false otherwise.
Example	`if var1 < 10 //if the value var1 is less than 10`

Operator	**>**
Comment	Returns true if the value to the left of the operator is greater than right, false otherwise.
Example	`if var1 > 10 //if var1 is greater than 10` `//if the value var1 is greater than 10`

Operator	**<=**
Comment	Returns true if the value to left of the operator is less than or equal to right, false otherwise.
Example	`if var1 <= var2 //if the value var1 is less than or equal to //var2`

Operator	**>=**
Comment	Returns true if the value to left of the operator is greater than or equal to the right, false otherwise.
Example	`if var1 >= var2 //if the value var1 is greater than or equal //to var2`

Operator	**==**
Comment	Returns true if the value to left of the operator is equal to right, false otherwise.
Example	`if var1 == var2 //if the value var1 is equal to var2`

Operator	**!=**
Comment	Returns true if the value to left of the operator isn't equal to right, false otherwise.
Example	`if var1 != var2 //if the value var1 isn't equal to var2`

Operator	**||**
Comment	Returns true if the condition on the left of the operator or the condition on the right is true, false otherwise.
Example	`if var1 < 5 || var2 > 5 //if the value var1 is less than 5 or //greater than 5`

Operator	**&&**
Comment	Returns true if the condition on the left of the operator and the condition on the right is true, false otherwise.
Example	`if var1 == 0 && var2 != 2` `//if the value var1 is equal to 0 and` `//if the value var1 isn't equal to 2`

C.4. Logical Operators

The conditional operator “if” compares using the boolean operators and executes the desired code columns.

```
if expression1

  statement1

else

  statement2

endif;
```

If expression1 is true, statement1 will be executed.

If expression2 is false, it will run expression2.

End if should be placed end of.

Example:

```
if var1 == 0                    //if var1 is equal to 0
 var2 = 10;                     //var2 is equal to 10
else                            //if var1 not equal to 0
 var2 = 20;                     //var2 is equal to 20
endif;                          //end
```

The conditional loop operator ***"while"*** compares using the boolean operators and executes the code column in a loop according to the specified condition.

```
while expression
  ...
endw;
```

While loop executes the code into the loop as long as expression1 is true.

endw should be placed end of.

Example:

```
while var1 != 100               //as long as the value of var1 is not 100
  var2 = var2 + 1;              //increase var2 by 1

  var1 = var2;                  //equal var2 to var1
endw;                           //end
```

The loop operator **“for”** executes the code column in a loop as the specified number of times.

```
for variable1 = value1 to value2 do

...

endfor;
```

When the for loop is used with to, the value of variable 1 is initialized equal to value1.

Increase by 1 in each loop.

The for loop is executes in a loop until it reaches value2.

endfor should be placed end of.

```
for variable1 = value1 downto value2 do

  ...

Endfor;
```

When the **for** loop is used with **downto**, the value of variable1 is started equal to value1.

Decrease by 1 in each loop.

The for loop is executes in a loop until it reaches value2.

endfor should be placed en of.

Example:

```
for var1 = 0 to 100 do                  //var1 loop from 0 to 100
var2 = var2 + 1;                        //increase var2 by 1 at the each loop
endfor;                                 //end
for var1 = 50 downto 0 do               //var1 loop from 50 to 0
var2 = var2 - 1;                        //decrease var2 by 1 at the each loop
endfor;                                 //end
```

C.5. Others

Operator	**func - endf**
Comment	Used to definition a function.
Example	`func function1()` //define function1 `...` `endf` //end
Operator	**call**
Comment	Used to call/execute a function.
Example	`call function1();` //execute/call function1

Operator	**sleep**
Comment	Used to wait for a period of time in milliseconds.
Example	`sleep(1000);` `//wait 1000 millisecond`
Operator	**endp**
Comment	Comes at the end of the macro code and specifies that the macro code ends here.

Operator	**getsystick**
Comment	Represents an increasing value in internal memory as milliseconds.
Example	`if(getsystick() - $10 > 5000)` `//Increase a variable by 1 //for 5000 ms` `a = a + 1;` `$10 = getsystick();` `endif;`

Operator	**getsystime**
Comment	Retrieves system timing information.
Example	`$0 = getsystime();` `//system time is assigned //to $0.`

Operator	**getsystouch**
Comment	is used to get elapsed time since last interaction with screen.
Example	`$0 = getsystouch(); //get elapsed time since last //touch`

Operator	**writeonce**
Comment	is used to shift address and write value.
Example	`for i = 0 to 2 do //addresses are shifted` `writeonce($10, i) = getonce($20,i); //as much as i's value` `endfor; //then read and written`

Operator	**getonce**
Comment	is used to shift address and read value.
Example	`for i = 0 to 2 do //addresses are shifted` `writeonce($10, i) = getonce($20,i); //as much as i's value` `endfor; //then read and written`

Operator	**writesync**
Comment	is used to write value synchronously.
Example	`varMod1 = //if writesync can write` `writesync("modbus1@40001", 1); //value it returns 1 if` `//not -1`

Operator	putbuf - writebuf
Comment	putbuf puts values to buffer and writebuf sends.
Example	varBuf = mw_putbuf("modbus1@40001",$1); //if writebuf can varBuf = mw_putbuf("modbus1@40002",$2); //send buffer varMod1 = mw_writebuf(); //successfully //returns 1 if not -1

Operator	putbuf - writebufsync
Comment	putbuf puts values to buffer and writebufsync writes.
Example	varBuf = mw_putbuf("modbus1@40001",$1); //if writebuf can varBuf = mw_putbuf("modbus1@40002",$2); //write buffer varMod1 = mw_writebuf(); //successfully //returns 1 if not -1

Operator	loadrecipe
Comment	is used to call prepared recipes.
Example	loadrecipe(“Dough”,”Bread”); //This operation has //"Dough" named recipe //and "bread" named //recipe data

Operator	resetcomm
Comment	is used to reset communication.
Example	resetcomm(“modbus”,”1”); //reset modbus device

C.6. Type Conversion

This is a feature that helps to convert various data types between each others. It is used with codes in macro page.

Conversions are like:

(int) : integer conversion.

(double) : double conversion.

(float) : float conversion.

In this example, value of $1 address is converted to double value and written to 40001 address of modbus1 device.

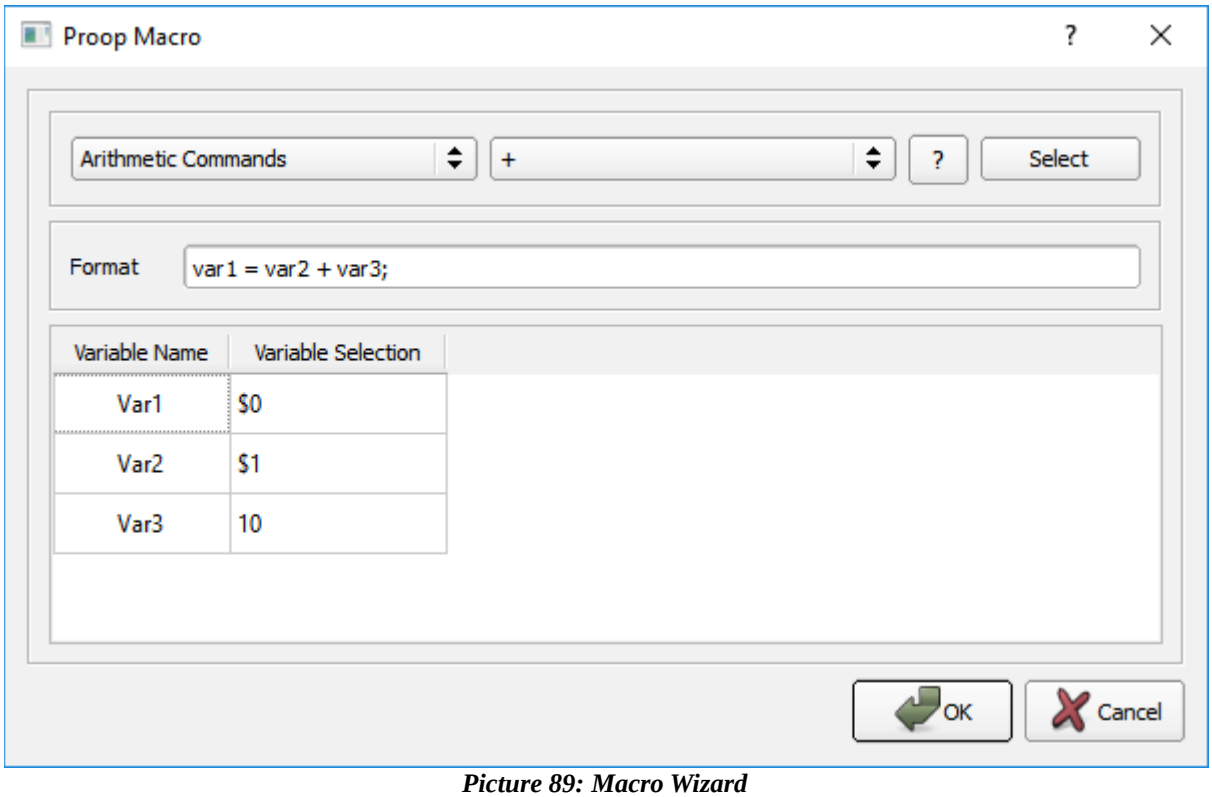
```
func main()

modbus1@40001 = (double) $1;

endf
endp
```

C.7. Macro Wizard

Macro Wizard is a help window which lists all macro commands and it has explanations and examples too. Thanks to this window, macro commands can be prepared in accordance with it's format and added to macro code window.

Macro Wizard can be reached with  this button on macro window.

First drop down list includes group names of commands and the second includes commands. Command can be selected with "Select" button and it's explanation and example can be reached with "?" button. After selection a command, if it has variables, they are listed and for variable selection, cells being under "Variable Selection" column should be clicked and edited. When all variables are edited, "OK" button is clicked. Prepared command is added to cursor's position on macro code window.



Picture 89: Macro Wizard

D) PROOP Connections

D.1. Models

Models	PROOP Types	COM2 RS-485	COM3 RS-232	COM4 RS-232	ETH	Digital Input/Output	Analog Input/Output	USB Host	USB Server
7'' Model Types	7L	✓	✓	✓				✓	✓
	7L.E	✓	✓	✓	✓			✓	✓
	7C	✓	✓	✓		✓		✓	✓
	7C.E	✓	✓	✓	✓	✓		✓	✓
10'' Model Types	10L	✓	✓	✓				✓	✓
	10L.E	✓	✓	✓	✓			✓	✓
	10C	✓	✓	✓		✓		✓	✓
	10C.E	✓	✓	✓	✓	✓		✓	✓
	10P	✓	✓	✓		✓	✓	✓	✓
	10P.E	✓	✓	✓	✓	✓	✓	✓	✓

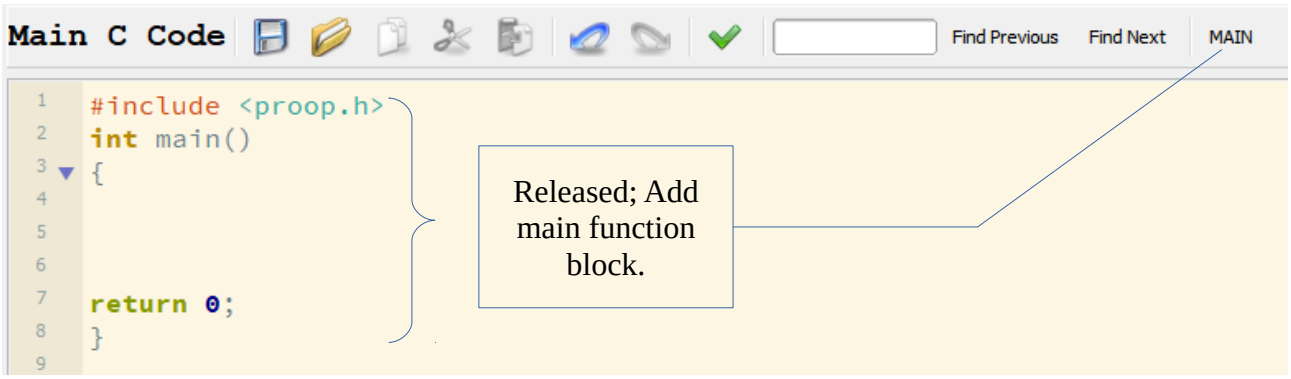
Table 26: PROOP Model List

E) C Code

See the subtitles for C code functions and writing.

E.1. Main Template

The main function template is shown in the picture below. When the button indicated by the red field is pressed, the main function block is added to the code writing area.



E.2. C Code Functions

Function	getData()	
Comment	It waits for the data to be read by communication and is read successfully 0 returns -1 on error.	
Usage	`int getData(void *data,const char * strAddress,unsigned int offset,int bytesize);`	
Example	`char led;` `getData(&led,"$0.0",0,1)`	`// &led -> data` `// $0.0 -> read address` `// 0 -> offset` `// 1 -> bytesize`

Function	setData()	
Comment	It waits for data to be written by communication, and returns 0 if successfully written.	
Usage	`int setData(void *data,const char * strAddress,unsigned int offset,int bytesize);`	
Example	`char led;` `setData(&led,"$0.0",0,1);`	`// &led -> data` `// $0.0 -> read address` `// 0 -> offset` `// 1 -> bytesize`

Function	getDataAsync()	
Comment	It does not wait for the data to be read by communication, it uses the previously read value. If read successfully, 0 returns -1 on error;	
Usage	`int getDataAsync (void *data,const char * strAddress,unsigned int offset,int bytesize);`	
Example	`uint64_t uval;` `getDataAsync(&uval,"$1",0,sizeof(uval));`	`// &uval -> data` `// $1 -> read adsress` `// 0 -> offset` `// sizeof(uval) -> bytesize`

Function	setDataAsync()	
Comment	It does not wait for the data to be written by communication, it assigns the data to be written to the queue. Even if there is a communication error, it always returns 0 unless there is a system error.	
Usage	`int setDataAsync (void *data,const char * strAddress,unsigned int offset,int bytesize);`	
Example	`uint64_t uval;` `setDataAsync(&uval,"$1",0,sizeof(uval));`	`// &uval -> data` `// $1 -> read adsress` `// 0 -> offset` `// sizeof(uval) -> bytesize`

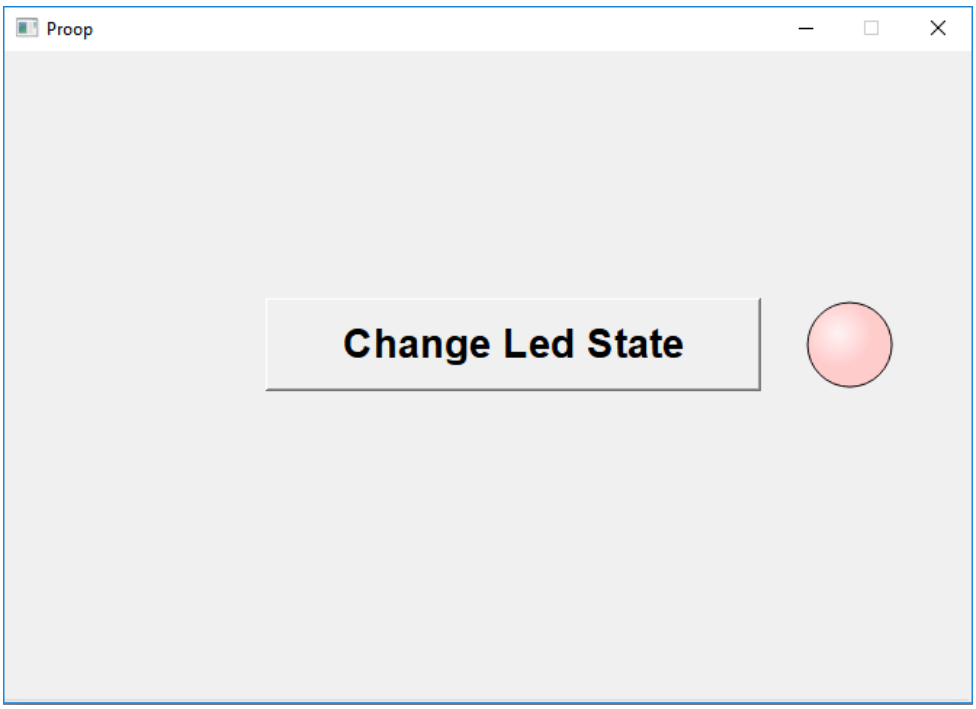
E.3. C Code Function Parameters

Parameter	Function
***data**	Pointer address of write or read data.
****strAddress***	**“devicename@address”** The input format of the internal addresses and the number of bytes to be used as bytesize are as follows; “$0”: 8 “$0.0”: 1 “$M0”: 8 “$M0.0”: 1 “$S0”: $S Since addresses starting with $ S2 are of type int, 4 must be entered as

	bytesize. Bytesize values to be used for Modbus protocol; Holding and Input register addresses 2, 1 for other input and output bits.
***offset**	Address offset value.
***bytesize**	The number of bytes of data to be written or read.

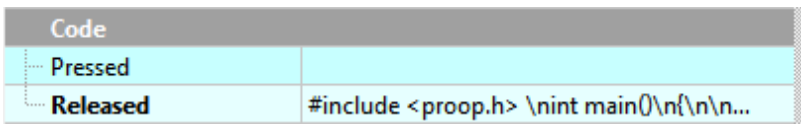
E.4. C Code Examples

Example -1: Changing the status of the LED element



Picture 90: Changing Led State

- **Step 1:** One button and led element is added to the working area.
- **Step 2:** On the Properties menu, under C code, click on Release and type C code in the form that opens.

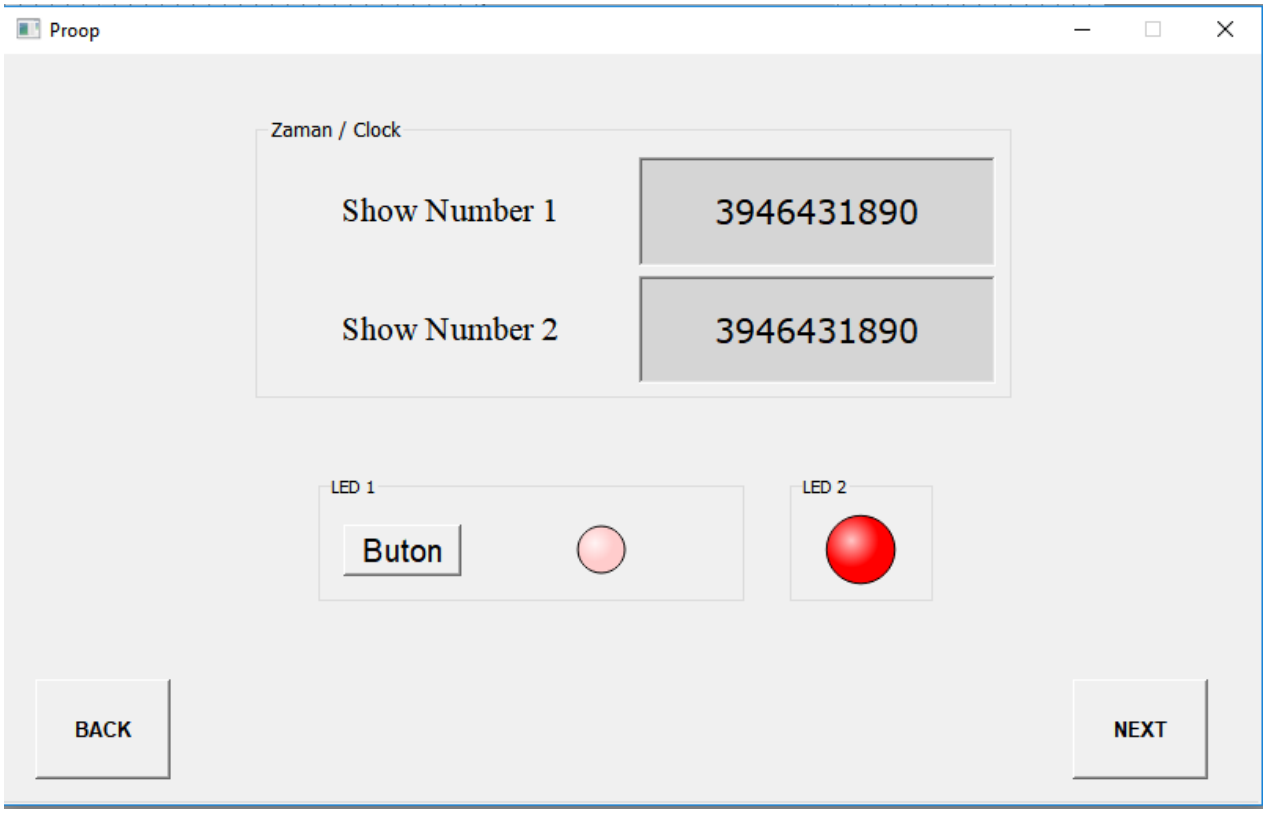


- **Step 3:** “Released”, the macro code is shown in the following table. If the LED status is 0, the LED turns off, if it is 1, the LED turns on.

Using Button	Execute Macro Code	Result
Change Led State Released button	#include <proop.h> int main() { char led; if (!getData(&led, "$0.0", 0, 1)) { led = !led; setData(&led, "$0.0", 0,1); } return 0; }	If status 1;  if status 0; 

Example -2: Showing Time Value

It is an application that shows the system clock and led statuses on the screen with C functions from two different addresses in the main C code and periodic C code editing forms. The screenshot of the application is as follows.



Picture 91: Example-2 Screenshot

Element addresses must be entered as given in the table below.

Element Name	Address Type	Write Address	Read Address
Show Number 1	UnsignedInt64	-	modbus@30001
Show Number 2	UnsignedInt64	-	modbus@30002
Button	Bit	modbus@40001.1	-
LED 1	Bit	-	modbus@40001.2
LED 2	Bit	-	modbus@40001.3

Main C Code

```
#include <proop.h>
#include <time.h>
#include <stdint.h>
#ifdef _WIN32
#include <windows.h>
#endif

void musleep(unsigned long usec)
{
#ifndef _WIN32
	struct timespec res;
	res.tv_sec = usec / 1000000;
	res.tv_nsec = (usec * 1000) % 1000000000;
	clock_nanosleep(CLOCK_REALTIME, 0, &res, NULL);
#else
	Sleep(usec / 1000);
#endif

}
uint64_t getClock()
{
	uint64_t val;
#ifndef _WIN32
	struct timespec sNow;
	clock_gettime(CLOCK_REALTIME, &sNow);
	val = sNow.tv_sec * 1000 + sNow.tv_nsec / 1000000;
#else
	val = GetTickCount();
#endif
	return val;
}

uint64_t uval, uval2;

int main()
{
```

```
      char led = 0;

      //if(getClock() - uval > 500)
      {
       uval = getClock();

       if (!getData(&led, "modbus@40001.1", 0, 1)) {
              setData(&led, "modbus@40001.2", 0, 1);
       }

      }

      uval2 = getClock();
      setDataAsync(&uval2, "modbus@30002", 0, sizeof(uval2));

      return 0;
}
```

Periodic C Code

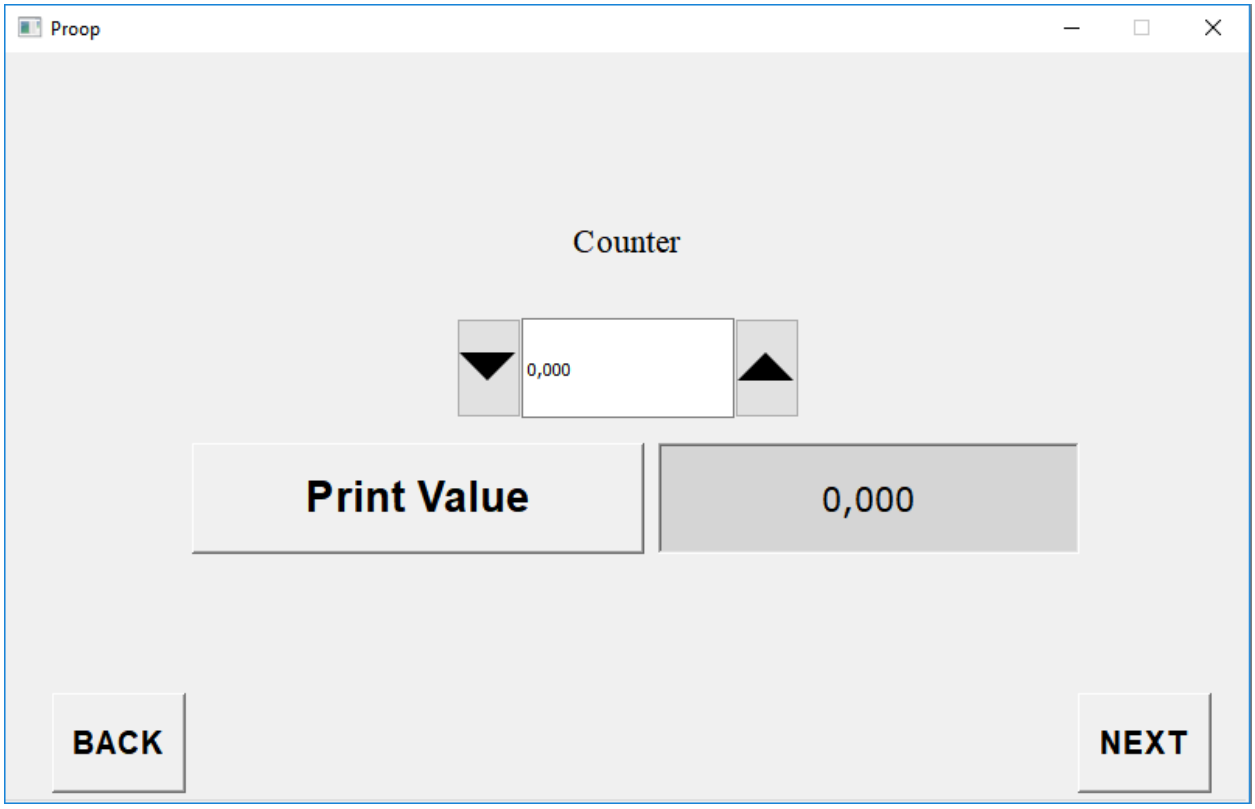
```
#include <proop.h>
#include <time.h>
#include <stdint.h>
#ifdef _WIN32
#include <windows.h>
#endif
void musleep(unsigned long usec)
{
#ifndef _WIN32
    struct timespec res;
    res.tv_sec = usec / 1000000;
    res.tv_nsec = (usec * 1000) % 1000000000;
    clock_nanosleep(CLOCK_REALTIME, 0, &res, NULL);
#else
     Sleep(usec / 1000);
#endif
}
```

```
uint64_t getClock()
{

	uint64_t val;
#ifndef _WIN32
	struct timespec sNow;
	clock_gettime(CLOCK_REALTIME, &sNow);
	val = sNow.tv_sec * 1000 + sNow.tv_nsec / 1000000;
#else
	val = GetTickCount();
#endif
	return val;
}

uint64_t uval, uval2;

int main()
{

	char led = 0;

	if (getClock() - uval > 500) {
	 uval = getClock();
	 //Led 2 status changes according to time.
	 if (!getData(&led, "modbus@40001.3", 0, 1)) {
		led = !led;
		setData(&led, "modbus@40001.3", 0, 1);
	 }

	}

	uval2 = getClock();
	setDataAsync(&uval2, "modbus@30001", 0, sizeof(uval2));

	return 0;
}
```

Example -3 :

An example of displaying the counter value on the screen using the label element using the C function.

Program screenshot is as follows.



Picture 92: Example-3 Screenshot

Element addresses must be entered as given in the table below.

Element Name	Address Type	Write Address	Read Address
Counter	Double	internal_memory@$3	-
Button (Print Value)	-	-	-
Label	Double	-	internal_memory@$4

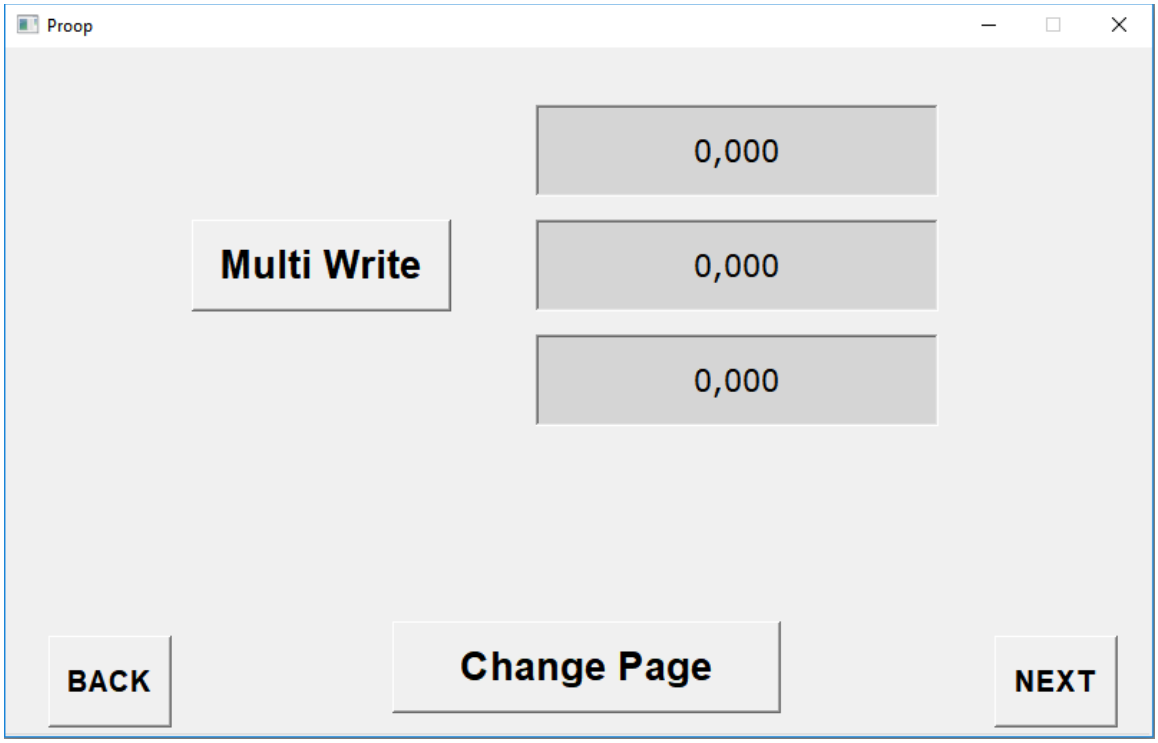
Press the "Print Value" button, then release. The following c code is executed.

Released C Code

```
#include <proop.h>
int main()
{
      double dval;
      if (!getData(&dval, "$3", 0, sizeof(dval))) {
        setData(&dval, "$4", 0, sizeof(dval));
      }

      return 0;
}
```

Example -4:

This example describes the C code programming of multiple writes and Page changes.
The screenshot of the program is as follows.



Picture 93: Example- 4 Screenshot

Element addresses are entered as given in the following table.

Element Name	Address Type	Write Address	Read Address
Multi Write	Double	internal_memory@$3	-
Change Page	-	-	-
Label 1	Double	-	internal_memory@$4
Label 2	Double	-	internal_memory@$5
Label 3	Double	-	internal_memory@$6

With multiple writing operations, three different addresses are written at the same time. The value at $4 is taken, and the value at $4 is printed on addresses $5, $ 6, respectively.

Multi Write Released C Code

```
#include <proop.h>
int main()
{
      double dval[3];
      if (!getData(&dval, "$4", 0, sizeof(dval))) {

        dval[0] = dval[0] + 1;
        dval[1] = dval[1] + 1;
        dval[2] = dval[2] + 1;
        setData(&dval, "$4", 0, sizeof(dval));
      }

      return 0;
}
```

Change Page Released C Code

```
#include <proop.h>
int main()
{
      int ival = 1;
      setData(&ival, "$S2", 0, sizeof(ival));
      return 0;
}
```

F) PROOP Access

F.1. Models

ProopBlack Eco	Display Type	Display Colors	Display Resolution	Luminance (Cd/m2)	Device Dimensions	Panel Cut-Out	Weight
Proop.black-4.eco	4.3'' TFT	260K colors	480x272	530	153x105x50mm	143x97mm	350gr
Proop.black-5.eco	5'' TFT	260K colors	800x480	330	153x105x50mm	143x97mm	350gr
Proop.black-7.eco	7'' TFT	260K colors	800x480	300	206x152x50mm	196x142mm	700gr

Proop Lite	Display Type	Display Colors	Display Resolution	Luminance (Cd/m2)	Ethernet (10/100 Mbps)	Wifi (IEEE 802.11b/g/n)
Proop-7L.wi	7'' TFT	260K color	800x480	300		1x
Proop-7L.E	7'' TFT	260K color	800x480	300	1x	
Proop-7L.Ewi	7'' TFT	260K color	800x480	300	1x	1x
Proop-10L	10.1'' TFT	16M color	1024x600	270		
Proop-10L.wi	10.1'' TFT	16M color	1024x600	270		1x
Proop-10L.E	10.1'' TFT	16M color	1024x600	270	1x	
Proop-10L.Ewi	10.1'' TFT	16M color	1024x600	270	1x	1x

ProopBlack Lite	Display Type	Display Colors	Display Resolution	Luminance (Cd/m2)	Ethernet (10/100 Mbps)	Wifi (IEEE 802.11b/g/n)
Proop.black-7L.wi	7'' TFT	260K color	800x480	300		1x
Proop.black-7L.E	7'' TFT	260K color	800x480	300	1x	
Proop.black-7L.Ewi	7'' TFT	260K color	800x480	300	1x	1x
Proop.black-10L	10.1'' TFT	16M color	1024x600	270		
Proop.black-10L.wi	10.1'' TFT	16M color	1024x600	270		1x
Proop.black-10L.E	10.1'' TFT	16M color	1024x600	270	1x	
Proop.black-10L.Ewi	10.1'' TFT	16M color	1024x600	270	1x	1x

Proop Process	Ethernet (10/100 Mbps)	Wifi (IEEE 802.11b/g/n)	Digital Inputs	Digital Outputs (Transistor)	Analogue Inputs	Analogue Outputs
Proop-10P.0.D5.D4.AC.AC			5x	4x	2x 4...20mA	2x 4...20mA
Proop-10P.wi.D5.D4.AC.AC		1x	5x	4x	2x 4...20mA	2x 4...20mA
Proop-10P.E.D5.D4.AC.AC	1x		5x	4x	2x 4...20mA	2x 4...20mA
Proop-10P.Ewi.D5.D4.AC.AC	1x	1x	5x	4x	2x 4...20mA	2x 4...20mA
Proop-10P.E2.D5.D4.AC.AC	2x		5x	4x	2x 4...20mA	2x 4...20mA
Proop-10P.E2wi.D5.D4.AC.AC	2x	1x	5x	4x	2x 4...20mA	2x 4...20mA

ProopBlack Process	Ethernet (10/100 Mbps)	Wifi (IEEE 802.11b/g/n)	Digital Inputs	Digital Outputs (Transistor)	Analogue Inputs	Analogue Outputs
Proop.black -10P.0.D5.D4.AC.AC			5x	4x	2x 4...20mA	2x 4...20mA
Proop.black -10P.wi.D5.D4.AC.AC		1x	5x	4x	2x 4...20mA	2x 4...20mA
Proop.black -10P.E.D5.D4.AC.AC	1x		5x	4x	2x 4...20mA	2x 4...20mA
Proop.black -10P.Ewi.D5.D4.AC.AC	1x	1x	5x	4x	2x 4...20mA	2x 4...20mA
Proop.black -10P.E2.D5.D4.AC.AC	2x		5x	4x	2x 4...20mA	2x 4...20mA
Proop.black -10P.E2wi.D5.D4.AC.AC	2x	1x	5x	4x	2x 4...20mA	2x 4...20mA

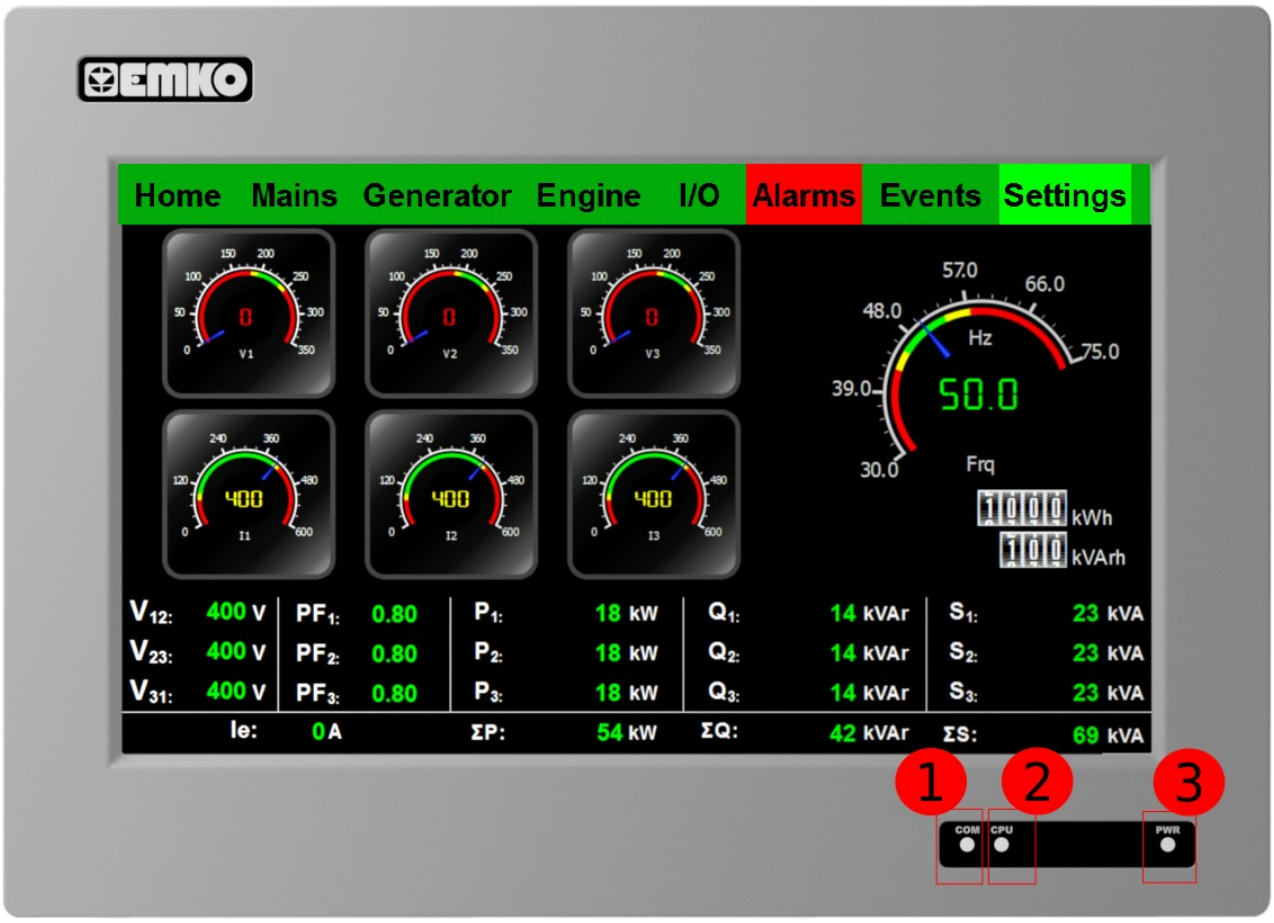
Proop Control	Display Type	Display Colors	Display Resolution	Luminance (Cd/m2)	Ethernet (10/100 Mbps)	Wifi (IEEE 802.11b/g/n)	Digital Inputs	Digital Outputs (Transistor)
Proop-7C	7'' TFT	260K color	800x480	300		1x	4x	4x
Proop-7C.wi	7'' TFT	260K color	800x480	300			4x	4x
Proop-7C.E	7'' TFT	260K color	800x480	300	1x		4x	4x
Proop-7C.E2	7'' TFT	260K color	800x480	300	2x		4x	4x
Proop-7C.Ewi	7'' TFT	260K color	800x480	300	1x	1x	4x	4x
Proop-7C.E2wi	7'' TFT	260K color	800x480	300	2x	1x	4x	4x
Proop-10C	10.1'' TFT	16M color	1024x600	270			5x	4x
Proop-10C.wi	10.1'' TFT	16M color	1024x600	270		1x	5x	4x
Proop-10C.E	10.1'' TFT	16M color	1024x600	270	1x		5x	4x
Proop-10C.Ewi	10.1'' TFT	16M color	1024x600	270	1x	1x	5x	4x

ProopBlack Control	Display Type	Display Colors	Display Resolution	Luminance (Cd/m2)	Ethernet (10/100 Mbps)	Wifi (IEEE 802.11b/g/n)	Digital Inputs	Digital Outputs (Transistor)
Proop.black-7C	7'' TFT	260K color	800x480	300		1x	4x	4x
Proop.black-7C.wi	7'' TFT	260K color	800x480	300			4x	4x
Proop.black-7C.E	7'' TFT	260K color	800x480	300	1x		4x	4x
Proop.black-7C.E2	7'' TFT	260K color	800x480	300	2x		4x	4x
Proop.black-7C.Ewi	7'' TFT	260K color	800x480	300	1x	1x	4x	4x
Proop.black-7C.E2wi	7'' TFT	260K color	800x480	300	2x	1x	4x	4x
Proop.black-10C	10.1'' TFT	16M color	1024x600	270			5x	4x
Proop.black-10C.wi	10.1'' TFT	16M color	1024x600	270		1x	5x	4x
Proop.black-10C.E	10.1'' TFT	16M color	1024x600	270	1x		5x	4x
	10.1'' TFT	16M color	1024x600	270	1x	1x	5x	4x

Proop-I/O	Digital Inputs	Digital Outputs	Analogue Inputs	Analogue Outputs
Proop-I/O.P.2.2.1.3.1.1	8x Digital	8x 1A Transistor (+V)	5x Pt-100 (-200...650°C)	2x 0/4...20mAdc
Proop-I/O.P.2.2.1.3.2.1	8x Digital	8x 1A Transistor (+V)	5x 0/4...20mAdc	2x 0/4...20mAdc
Proop-I/O.P.2.2.1.3.3.1	8x Digital	8x 1A Transistor (+V)	5x 0...10Vdc	2x 0/4...20mAdc
Proop-I/O.P.2.2.1.3.4.1	8x Digital	8x 1A Transistor (+V)	5x 0...50mV	2x 0/4...20mAdc
Proop-I/O.P.2.2.1.3.1.2	8x Digital	8x 1A Transistor (+V)	5x Pt-100 (-200...650°C)	2x 0...10Vdc
Proop-I/O.P.2.2.1.3.2.2	8x Digital	8x 1A Transistor (+V)	5x 0/4...20mAdc	2x 0...10Vdc
Proop-I/O.P.2.2.1.3.3.2	8x Digital	8x 1A Transistor (+V)	5x 0...10Vdc	2x 0...10Vdc
Proop-I/O.P.2.2.1.3.4.2	8x Digital	8x 1A Transistor (+V)	5x 0...50mV	2x 0...10Vdc

F.2. View Panel

The front face of the PRO Operator Panel is as in Picture-61 below and the leds on. Leds are numbered and explained in Table-27.



Picture 94: PROOP-Front View

1	**COM**	Communication led with PLC
2	**CPU**	Displays the current state of the CPU.
3	**PWR**	An energy led.

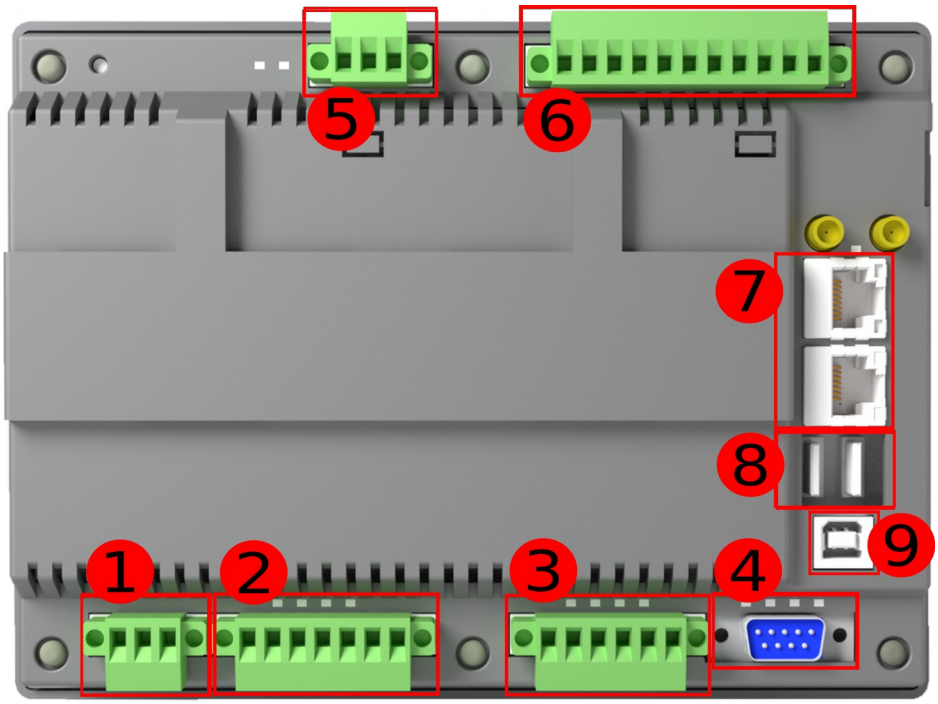
Table 27: PROOP-Front View

The back side of PROOP is different in the models.

The back view of the panels has been examined in two different ways as PROOP 7'' Models and PROOP 10'' Models.

The back of the PROOP 7 '' models is as shown in Picture-61 below.

Inputs are numbered and inputs numbered in Table-28 are explained.



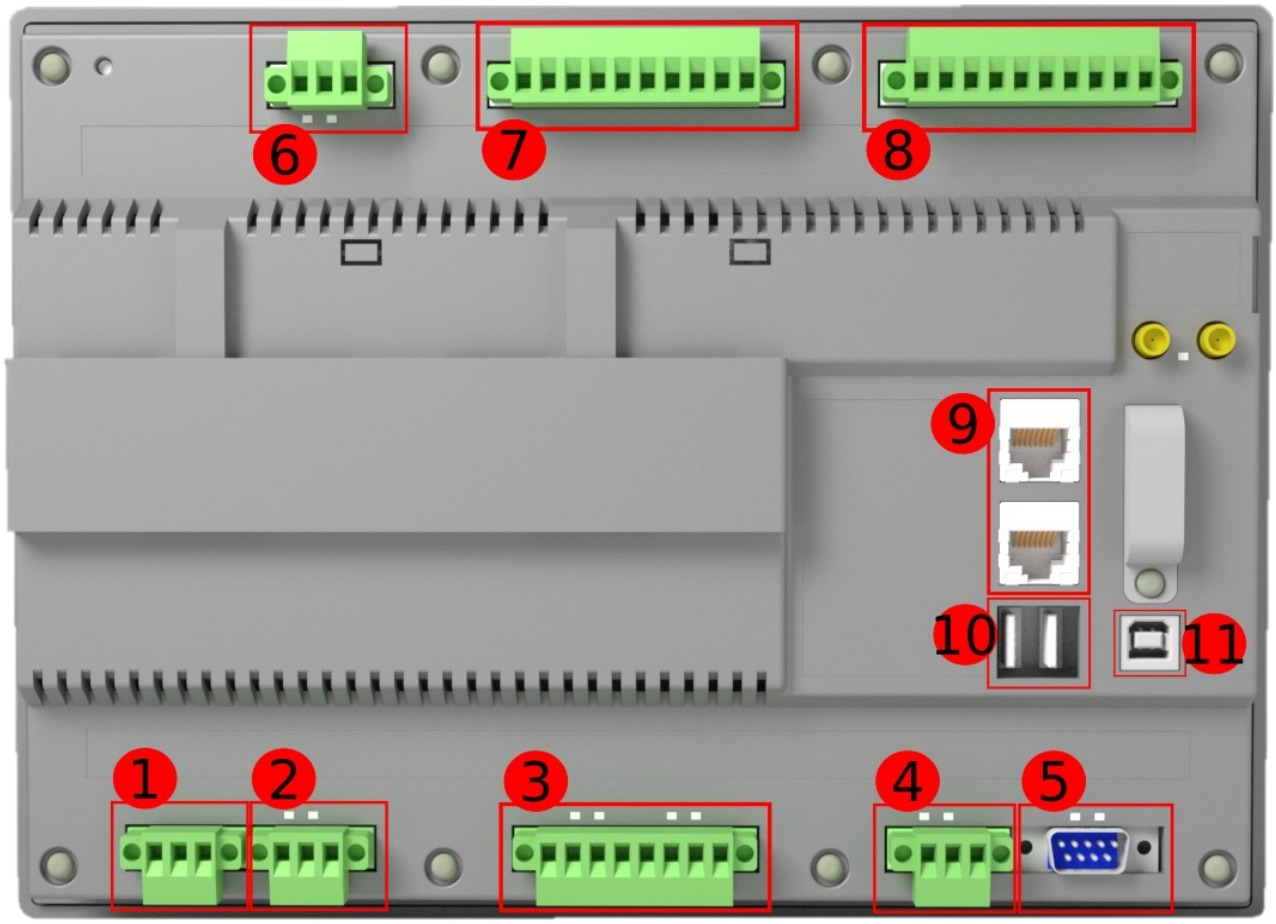
Picture 95: PROOP 7'' Back

1	Energy	**6**	Digital Inout/Output
2	COM1	**7**	Ethernet
3	COM2-COM3	**8**	USB Device
4	COM4	**9**	USB Host
5	Not use		

Table 28: PROOP 7'' Inputs

The back of the PROOP 10'' models is as shown in Picture-62 below.

Inputs are numbered and inputs numbered in Table-29 are explained.



Picture 96: PROOP 10'' Back

1	Energy	**7**	Analog Input
2	Out of use	**8**	Analog Output
3	COM1-COM2	**9**	Ethernet
4	COM3	**10**	USB Device
5	COM4	**11**	USB Host
6	Out of use		

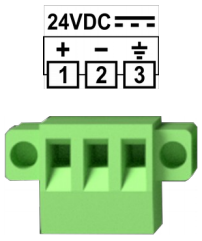
Table 29: PROOP 10'' Inputs

F.2.1. Pin Connections

PROOP 7'' Model and PROOP 10'' Model, the terminals used on the back are different and the pin connections are different.

The different terminals pin connections with PROOP 7"and PROOP 10" terminals are described under separate headings.

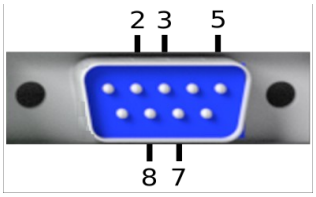
F.2.1.1. Supply



Inputs
+
-
⏚

Table 30: Supply Connections

F.2.1.2. COM4

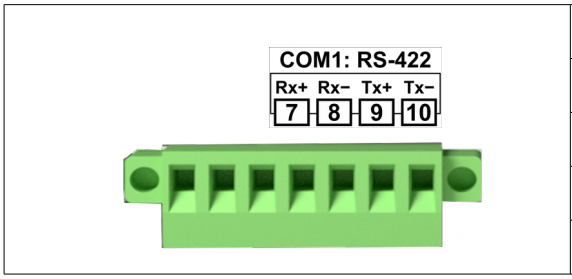


Inputs
2--Rx
3--Tx
5--GND
7--RTS
8--CTS

Table 31: COM4 Pin Connections

F.2.2. Pin Connections in PROOP 7" Models

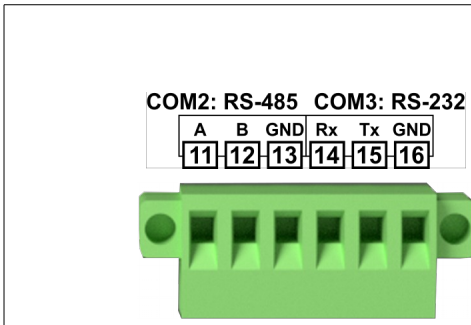
F.2.2.1. COM1



Inputs
Rx+
Rx-
Tx+
Tx-

Table 32: COM1 Pin Connections

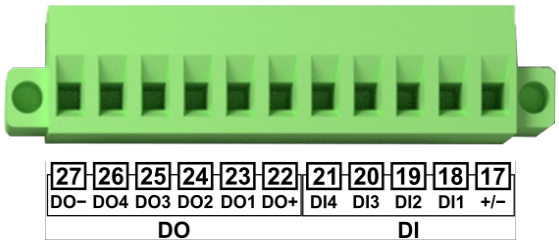
F.2.2.2. COM2-COM3



Inputs
A
B
GND
Rx
Tx
GND

Table 33: COM2- COM3 Pin Connections

F.2.2.3. Digital Inputs/Outputs



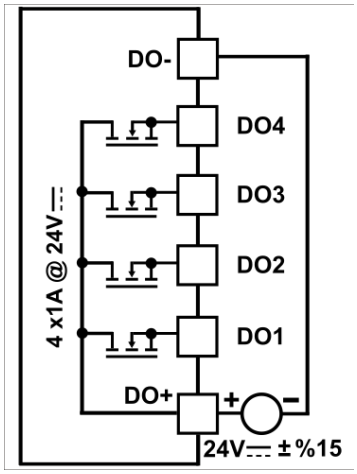
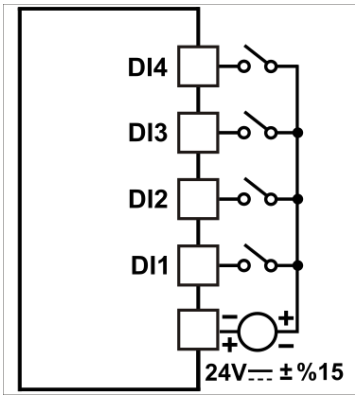
Input	Comment	Connection Scheme
DO-	Digital Output Supply -	
DO4	Digital Outputs	
DO3		
DO2		
DO1		
DO+	Digital Output Supply+	
DI4	Digital Inputs	
DI3		
DI2		
DI1		
+/-	NPN / PNP Selection of Digital Inputs	

Table 34: Digital Input/Output Pin Connection

F.2.3. Pin Connections PROOP 10" Models

F.2.3.1. COM1-COM2

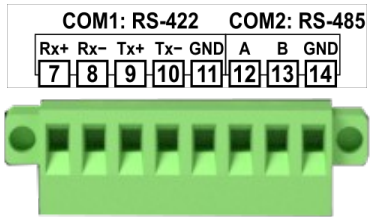
	Inputs
	Rx+
	Rx-
	Tx+
	Tx-
	GND
	A
	B
	GND

Table 35: COM1- COM2 Pin Connections

F.2.3.2. COM3

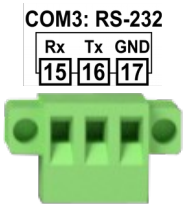
	Inputs
	Rx
	Tx
	GND

Table 36: COM3 Pin Connections

F.2.3.3. Analog/Digital Inputs

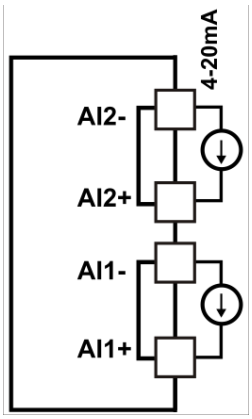
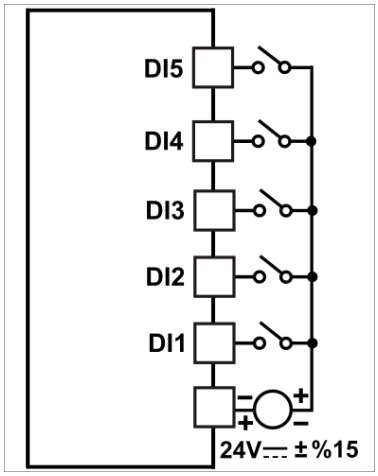
Inputs	Comment	Connection Scheme
AI2-	Analog Input2 -	
AI2+	Analog Input2 +	
AI1-	Analog Input1 -	
AI1+	Analog Input1 +	
DI5	Digital Inputs	
DI4		
DI3		
DI2		
DI1		
+/-	NPN / PNP Selection of Digital Inputs	

Table 37: Analog Inputs Pin Connections

F.2.3.4. Analog/Digital Outputs

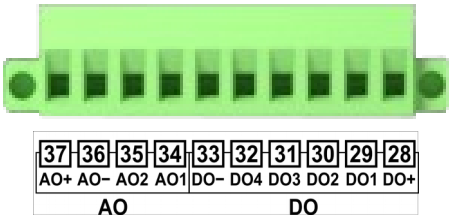
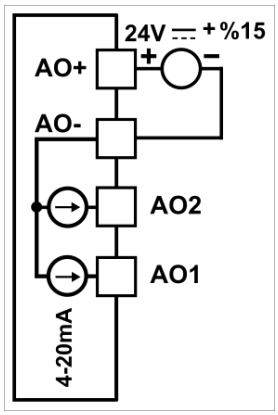
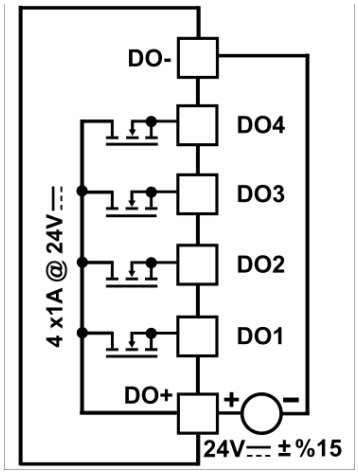
<table>
<tr><td rowspan="10"> 
</td><th>Inputs</th><th>Comment</th><th>Connection Scheme</th></tr>
<tr><td>AO+</td><td>Analog Output Supply +</td><td rowspan="3"> 
</td></tr>
<tr><td>AO-</td><td>Analog Output Supply -</td></tr>
<tr><td>AO2</td><td rowspan="2">Analog Outputs</td></tr>
<tr><td>AO1</td><td rowspan="6"> 
</td></tr>
<tr><td>DO-</td><td>Digital Output Supply -</td></tr>
<tr><td>DO4</td><td rowspan="4">Digital Outputs</td></tr>
<tr><td>DO3</td></tr>
<tr><td>DO2</td></tr>
<tr><td>DO1</td></tr>
<tr><td>DO+</td><td>Digital Output Supply +</td></tr>
</table>

Table 38: Analog Outputs Pin Connections

F.2.4. Internal I/O Address Definitions

Device Type	Format	Range
Digital Input	%IXn.k	n: 0-0 k: 0-4
Digital Output	%QXn.k	n: 0-0 k: 0-3
Analog Input	%IWn	n: 0-1
Analog Output	%MWn	n: 0-1

Table 39: Internal Input / Output Address Definitions

F.2.5. Internal Memory Address Definitions

Device Type	Format	Range
Volatile Memory	$n	n: 0-65535
Non-Volatile Memory	$Mn	n: 0-65535
Volatile Memory Bit	$n.k	n: 0-65535 k: 0-15
Non-Volatile Memory Bit	$Mn.k	n: 0-65535 k: 0-15
Internal Settings	$Sn	n: 0-65535

Table 40: Internal Memory Addresses

F.3. Supported Communication Protocols

Protocols supported by PROOP are addressed.

Supported protocols are listed in the table below.

	Brand	Protocols
1	MODBUS	Modbus-RTU
2	MODBUS	Modbus-ASCII
1	MODBUS	Modbus TCP/IP
2	MODBUS	Modbus-ASCII(Slave)
1	MODBUS	Modbus-RTU(Slave)
2	MODBUS	Modbus TCP/IP(Slave)
3	SIEMENS	S7-200(PPI)
4	SIEMENS	S7-300(ISOTCP)
5	SIEMENS	S7-400(ISOTCP)
6	SIEMENS	S7-1200(ISOTCP)

Table 41: Supported Brands

F.3.1. MODBUS Master Address Definitions

Address formats and address ranges are listed for devices using the Modbus communication protocol in the table below.

Device Type	Format	Range	Type
Discreate Output Coils	1000n	n: 1-65535	Read-Write
Discreate Input Coils	2000n	n: 0-65535	Read
Input Registers	3000n	n: 0-65535	Read
Holding Registers	4000n	n: 0-65535	Read-Write
Holding Bit	4000n.k	n: 0-65535 k: 0-15	Read-Write
Input Bit	3000n.k	n: 0-65535 k: 0-15	Read
Holding Registers (Write Multi)	WMn	n: 0-65535	Read-Write

Table 42: MODBUS-RTU Address Definitions

F.3.2. MODBUS Slave Address Definitions

Standard

Internal Memory Name	Modbus Address Range		Modbus Functions
	Start	End	
Volatile Memory	40001	42000	3,6,16
Non-Volatile Memory	42001	44000	3,6,16
Analog Outputs	44001	44002	3,6,16
Internal Settings	45001	45500	3,6,16

Extended

Internal Memory Name	Modbus Address Range		Modbus Functions
	Start	End	
Volatile Memory	410001	420000	3,6,16
Non-Volatile Memory	420001	430000	3,6,16
Analog Outputs	435001	435500	3,6,16
Internal Settings	450001	455000	3,6,16

Internal Memory Name	Modbus Address Range		Modbus Functions
	Start	End	
Analog Inputs	30001	30002	4

Internal Memory Name	Modbus Address Range		Modbus Functions
	Start	End	
Digital Outputs	00001	00004	1,5,15

Internal Memory Name	Modbus Address Range		Modbus Functions
	Start	End	
Digital Inputs	10001	10004/10005*	2

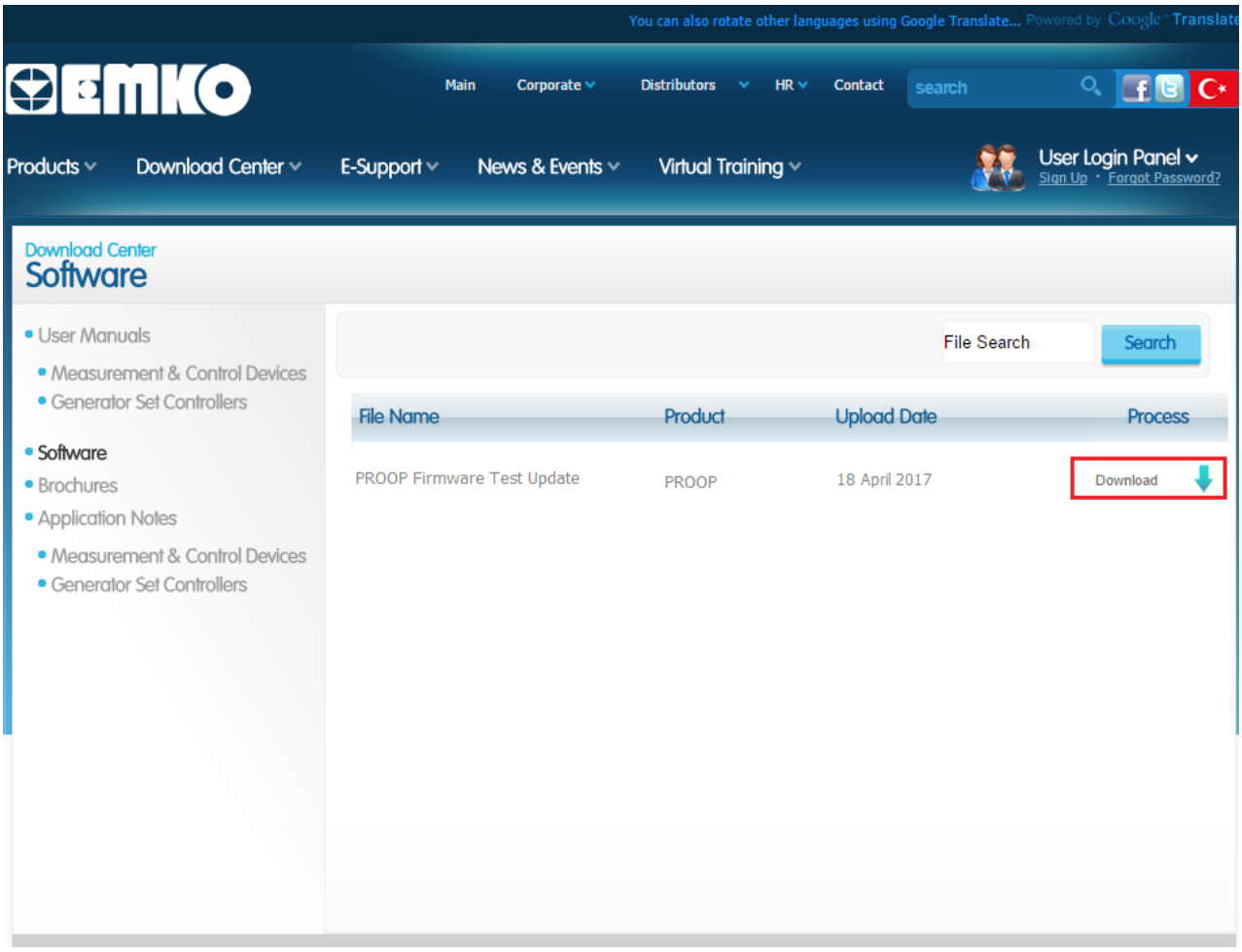
NOT*: 5th digital input only available on Proop.10P and Proop.10P.E

G) PROOP Upgrade

PROOP device is upgraded to the current version with Usb connection.

You can upgrade firmware with following the steps below.

- http://www.emkoelektronik.com.tr/ Download the update file from the Download Center → Software section of the website.
- Or http://www.proopforum.com/ download the update file from Proop Forum Site Technical Docs → Proop HMI Firmware Update.



Picture 97:Access Site For Software Source

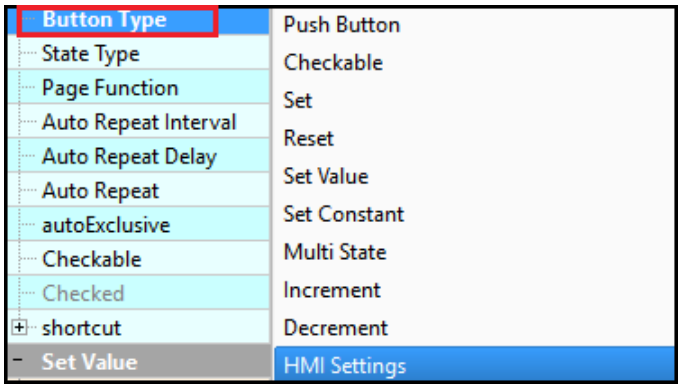
- Copy the compressed file ***'update.tar.gz'*** in the downloaded zip file to the **main directory** of the usb memory.
- Plug the usb memory into the USB port on the back of the device.
- If you cut off the power of the device and you give it again, the installation process will start.
- You can follow the installation process on the device screen as in Picture-65 below.

Picture 98: Software Update

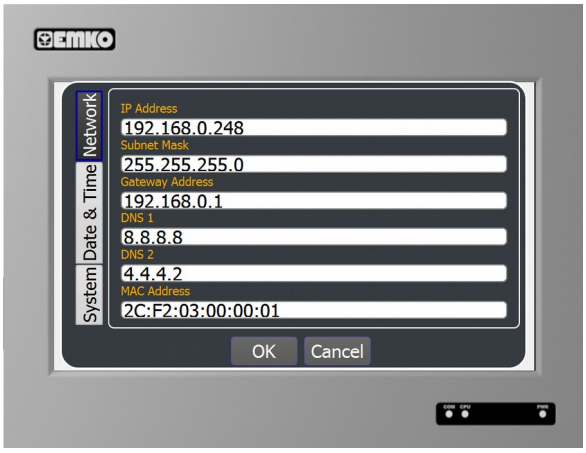
H) HMI Settings

You can follow the steps below to view and edit the Ethernet settings on the PLC screen.

- Drag any of the button elements to the project page you are using in the Proop Builder program.
- To configure the button, select the button type in the button section of the properties list as ***'HMI settings'*** as below.



- Click on 'online simulation' from online simulation tools. 
- Network settings is shown as default tab.
- The screens of the HMI settings are displayed as follows

Tabs	HMI Settings Screens	
Network Settings		The device information are IP address, subnet mask, DNS 1 and DNS 2. The user can edit this information. The MAC address can not be edited and is shown as read only.
Date & Time		Displays the current time and date. The user can edit time and date.
System		**Buzzer:** Adjust the buzzer sound. The left button is OFF, The right button is ON. **Brightness:** Adjust the brightness. The button increases the brightness from left to right and the maximum value is 7. The button decreases from right to left and minimum value is 0.

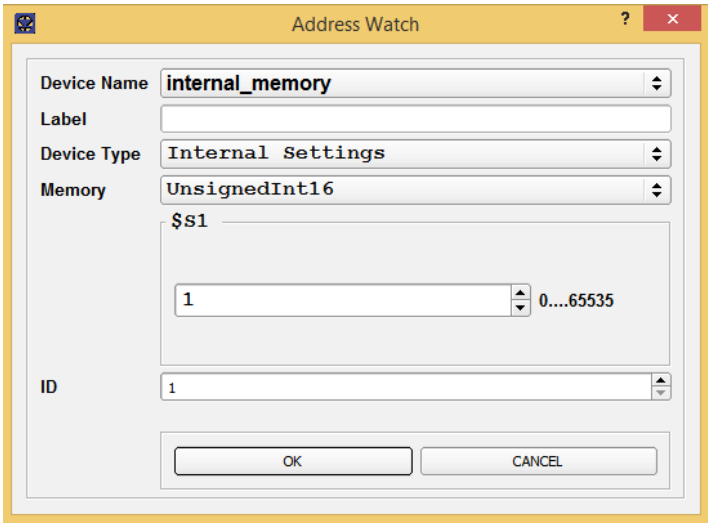
Picture 99: HMI Settings

I) Defining System Settings by Addressing

I.1. Buzzer

To view and edit the PLC buzzer status, you can follow the steps below.

- Drag and drop the Switch 2 element onto the page of the project you are using in the Proop Builder program.
- To edit the Switch 2 element, click on the icon displayed on the right in the properties list-> address-> write address. [...] A new window will open as below.

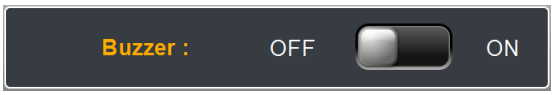


Picture 100:Address Watch(Buzzer)

- Select device name ***'internal_memory'*** as above.
- Select the device type ***'Internal Settings'*** and the memory is displayed as **'$S0'**.
- Write the deviceID in the ID field and click the ***"Ok"*** button.
- The write address field is displayed as follows.

Property	Value
Address	
Read Address	
Write Address	internal_memory@$S0
slaveid	1
addresstype	UnsignedInt16

- Click on 'online simulation' from the tools.
- The buzzer setting is displayed as below.



Picture 101:Buzzer

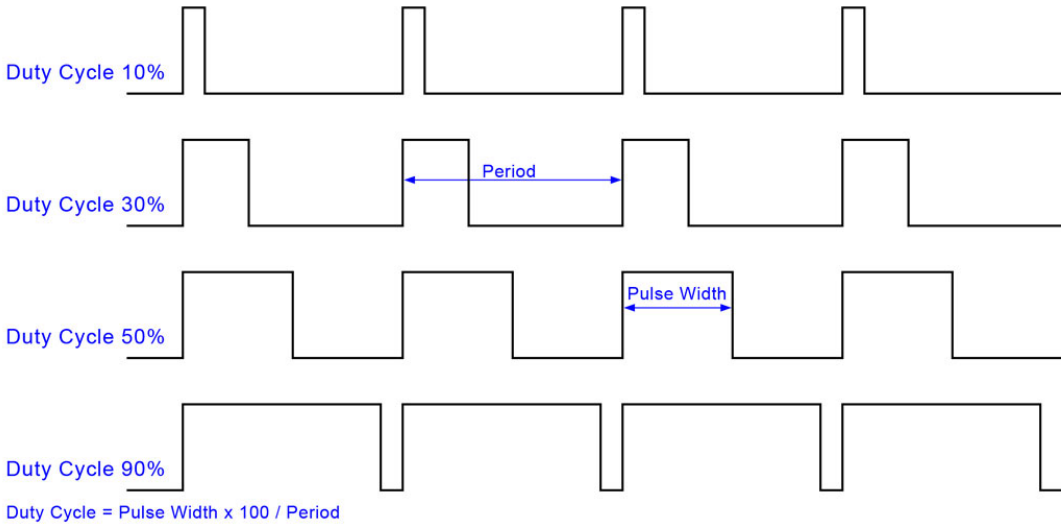
I.1.1. Buzzer / PWM Functions

PWM is the technique to change the output voltage by adjusting the period and the duty cycle of a square wave.

The PWM functions are described in the following tables.

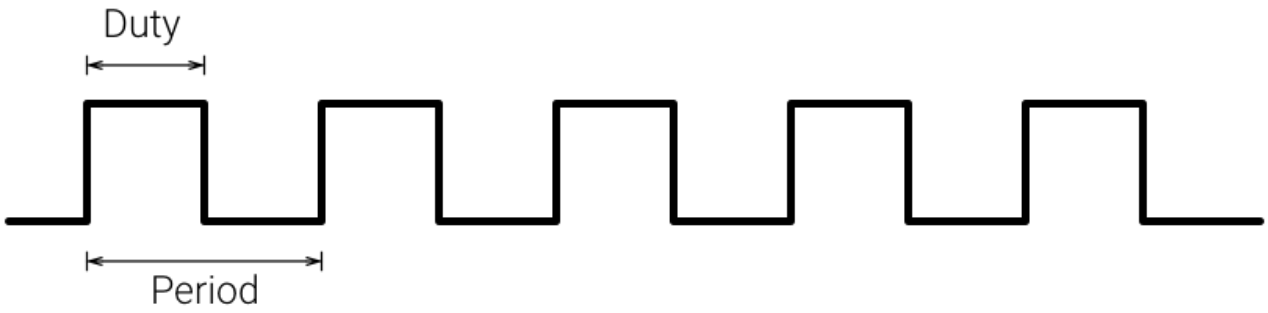
Function	**setpwmperiod()**
Comment	It is a function that holds the total period (T).
Usage	`setpwmperiod (“Output name DO1-DO4”, ms duty cycle);` `// Buzzer periyod(T) 3000 msn`
Example	setpwmperiod("DO1", 1000); //Set the period time of the PWM signal to 1000msec.

Function	**setpwmdutycycle()**
Comment	Used to define the duty cycle of the PWM signal. In 1 period slice, how many msec buzzer is “ON tutul is kept.

	
Usage	`setpwmdutycycle (“Output name DO1-DO4”, ms);`
Example	`setpwmdutycycle("DO1", 500); //The duty cycle of the PWM signal is 500msec.`

Period (T) : It is the time during which a full swing occurs or for a full cycle. Unit is in seconds and inversely proportional to frequency. In this case, the frequency is the number of periods in 1 second T=1/f we can express with the formula. The frequency of a signal with a period of 1 second is 1Hz.

Duty Cycle : The duration of the square wave in position 1.



Picture 102: Square Wave Signal

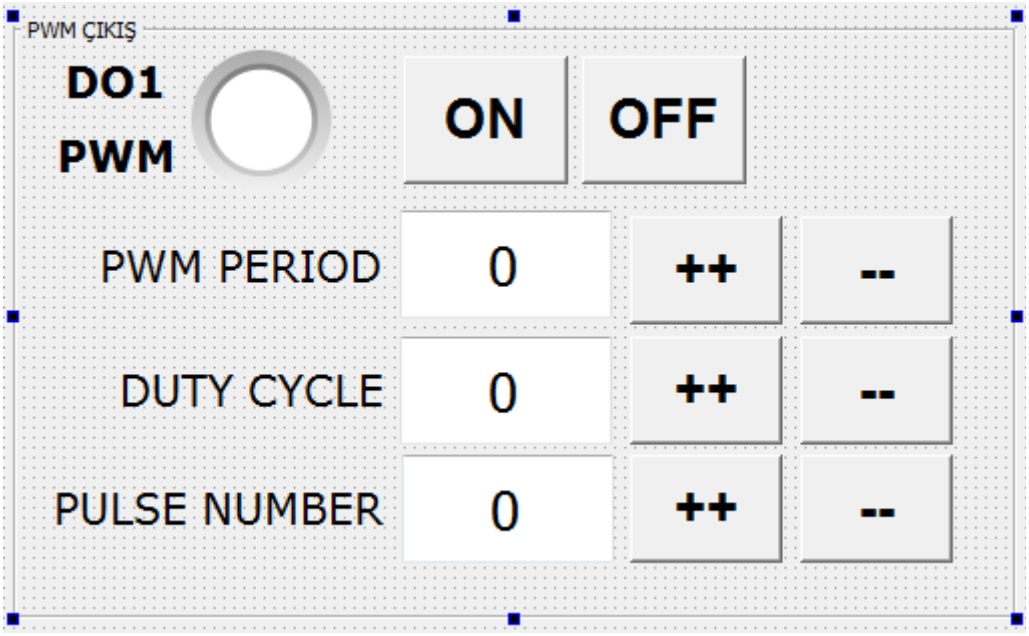
Function	**setpwmpulsenumber()**
Comment	It is used to define the number of square waves to be produced. If this value is defined as zero, a continuous PWM signal is generated. If a number is entered, only the number of PWM signals defined here is generated.
Usage	`setpwmpulsenumber("Output name DO1-DO4",pulse period);`
Example	`setpwmpulsenumber("D01",5); // After generating 5 pulses, the output is OFF.`

Fonksiyon	**setpwmenable()**
Comment	Enables or disables PWM operation. **Note:** The parameters are set before Pwm operation is activated. `setpwmperiod("$S0", 3000);` `setpwmdutycycle("$S0", 300);` `setpwmpulsenumber("$S0", 0);` `setpwmenable("$S0", 1);` The parameters were set and the pwm output was activated.
Usage	`setpwmenable("$S0",1); // 1 ->enable buzzer output` `setpwmenable("$S0",0); // 0->disable buzzer output`

Points to be considered;

- Duty Cycle , Can not be entered larger than pwm period.
- The minimum time between period and duty cycle is 0.5 ms, the minimum duty cycle time is 0.5 milliseconds.
- Period 1 ms. By entering the duty cycle 0.5 ms, a continuous pulse of 2khz can be generated. The maximum frequency is 2khz.

Example -1:



Picture 103: Example -1 Screenshot

The element addresses used are as follows.

Element Name	Address Type	Write Address	Read Address
PWM Period	Double	internal_memory@$M0	internal_memory@$M0
Duty Cycle	Double	internal_memory@$M1	internal_memory@$M1
Pulse Number	Double	internal_memory@$M2	internal_memory@$M2

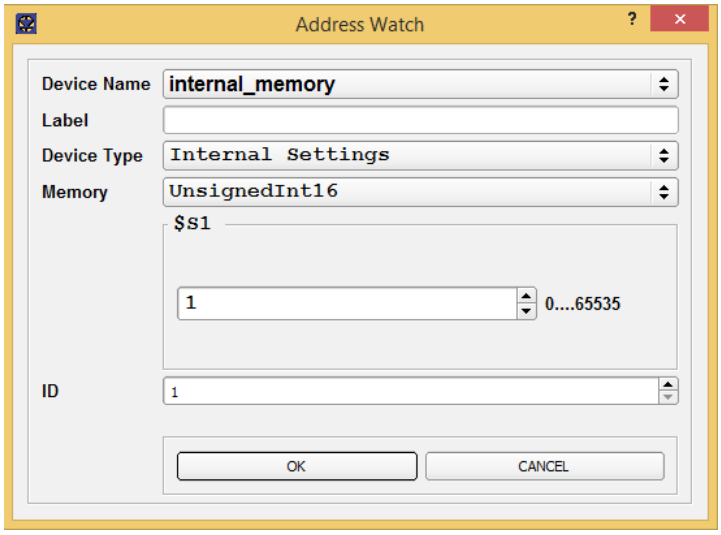
PWM period $M0, Duty Cycle $M1 and pulse number $M2 are defined. The pwm identification codes are written to the key pressing macros of the ON and OFF buttons.

When ON Button Pressed Macro Code	When OFF Button Pressed Macro Code
func main() setpwmperiod("DO1", $M0); setpwmdutycycle("DO1", $M1); setpwmpulsenumber("DO1", $M2); setpwmenable("DO1", 1); endf endp	global g_var1; func main() setpwmenable("DO1", 0); endf endp

I.2. Brightness

You can follow the steps below to view and edit the screen brightness of the PLC screen.

- Drag and drop the scroll bar element to the page of the project you are using in the Proop Builder program.
- To edit the scroll bar, click the icon displayed on the right in the properties list-> address-> write address. ... A new window will open as below.



Picture 104:Address Watch(Brightness)

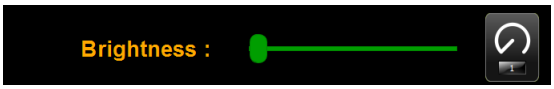
- Select device name ***'internal_memory'*** as above.
- Select the device type ***'Internal Settings'***, and the memory is displayed as **'$S1'**.
- Write the deviceID in the ID field and click the ***"Ok"*** button.
- The write address field is displayed as follows.

Address	
Read Address	
Hide Address	
Write Address	internal_memory@$S1
slaveid	1
addresstype	UnsignedInt16

- To specify the minimum and maximum limits of the brightness value, edit the properties list-> data section. It can be edited as follows.

Set Value	
Value	1.000000
Constant Value	1.000000
Max	7.000000
Min	1.000000

- Click on 'online simulation' from the tools. 
- The brightness setting is displayed as below.



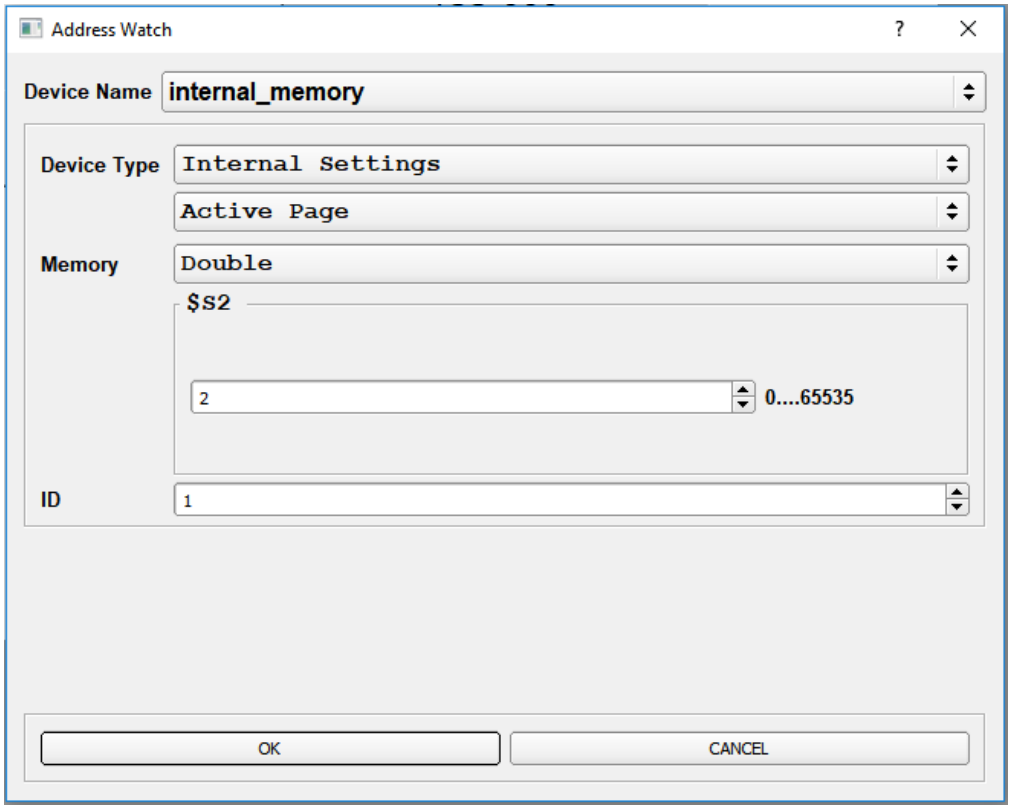
Picture 105:Brightness

I.3. Active Page

You can follow the steps below to view and edit the Active Page address status.

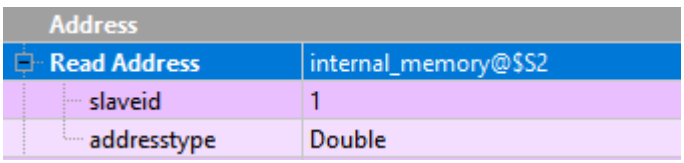
- Drag and drop the element you want to use to the page of the project you are using in Proop Builder.
- Edit element, Features list->Address->read at

  click the icon displayed on the right. [...]. A new window will open as below.



Picture 106: Address Watch(Active Page)

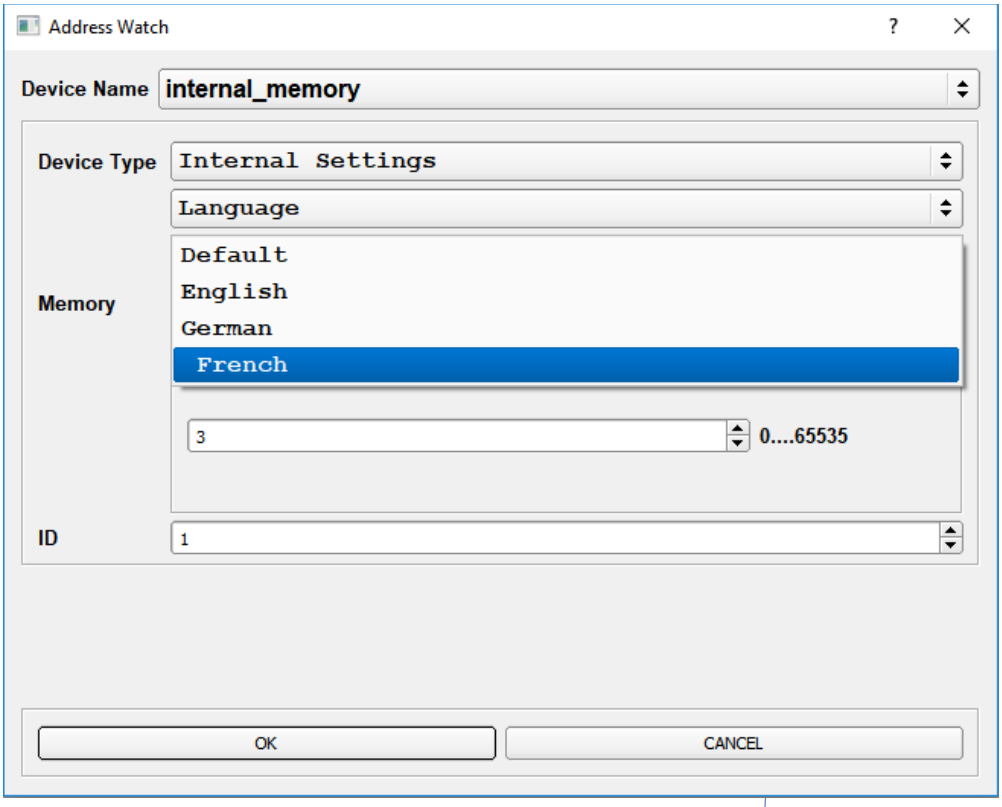
- Select device name ***'internal_memory'*** .
- Select device type '**Internal Settings**' and select '**Active Page**' from the drop-down menu, the memory is displayed as '$S2'.
- Type the device ID in the ID field and click the '**OK**' button. The Write address field appears as follows.



I.4. Language

You can follow the steps below to view and edit the language address status.

- Drag and drop the element you want to use to the page of the project you are using in Proop Builder.
- Edit Element, Features list->Address->read at click the icon displayed on the right.  . A new window will open as below.



Picture 107: Address Watch(Language)

This section lists previously created languages. Which language is used is selected.

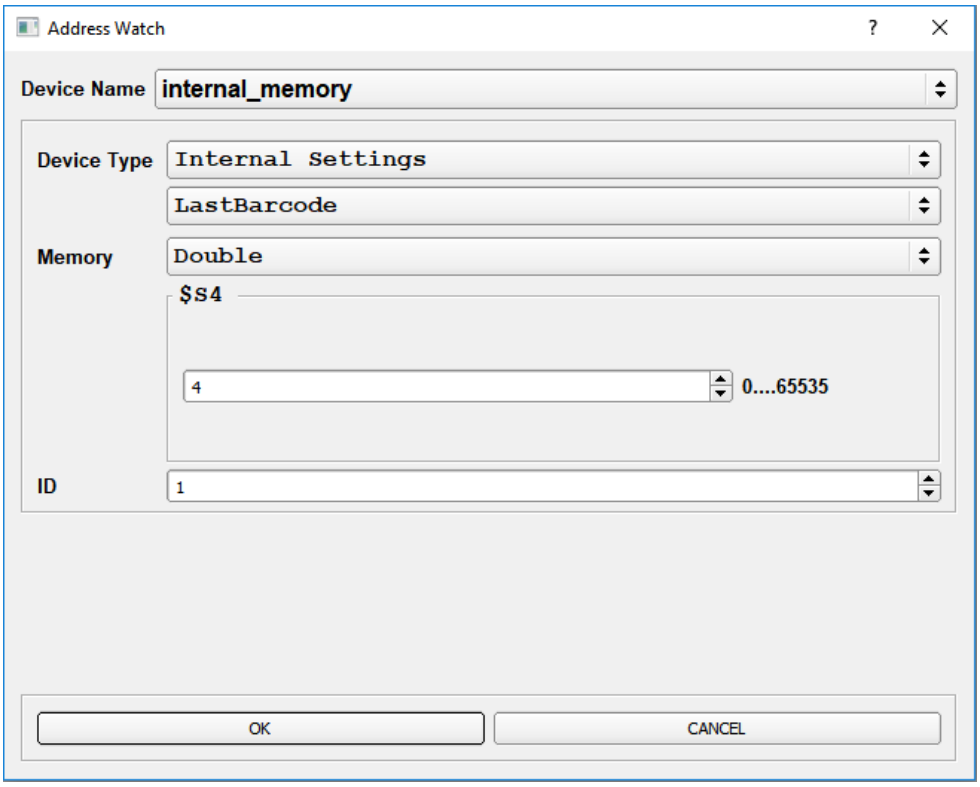
- Select device name ***'internal_memory'***.
- Select device type '**Internal Settings**' and select '**Language**' from the drop-down menu, the memory is displayed as '$S3'.
- Type the device ID in the ID field and click the '**OK**' button. The Write address field appears as follows.

Address	
Read Address	internal_memory@$S3
slaveid	1
addresstype	Double

I.5. LastBarcode

You can follow the steps below to view and edit the LastBarcode address status.

- Drag and drop the element you want to use to the page of the project you are using in Proop Builder.
- Edit Element, Features list->Address->read at click the icon displayed on the right. ... . A new window will open as below.



Picture 108: Address Watch(Last Barcode)

- Select device name ***'internal_memory'*** .
- Select device type '**Internal Settings**' and select '**LastBarcode**' from the drop-down menu, the memory is displayed as '$S4'.
- Type the device ID in the ID field and click the '**OK**' button. The Write address field appears as follows.

Address	
Read Address	internal_memory@$S4
slaveid	1
addresstype	Double

Always hold 2 characters at $ S4. Example ; If the value at $ S4 is sent to an address in the internal memory, the table below shows how it is stored in memory from that address.

Example; “123456789ABCDEFGHIJK” A **20-character** text is stored in **10 memory areas**. Because each address is kept as **2 characters**

Address	Enter TextInput Value	Barcode Text
$M0 = $S4	“123456789ABCDEFGHIJK”	“12”
$M1		“34”
$M2		“56”
$M3		“78”
. . .		. . .
$M9		“JK”

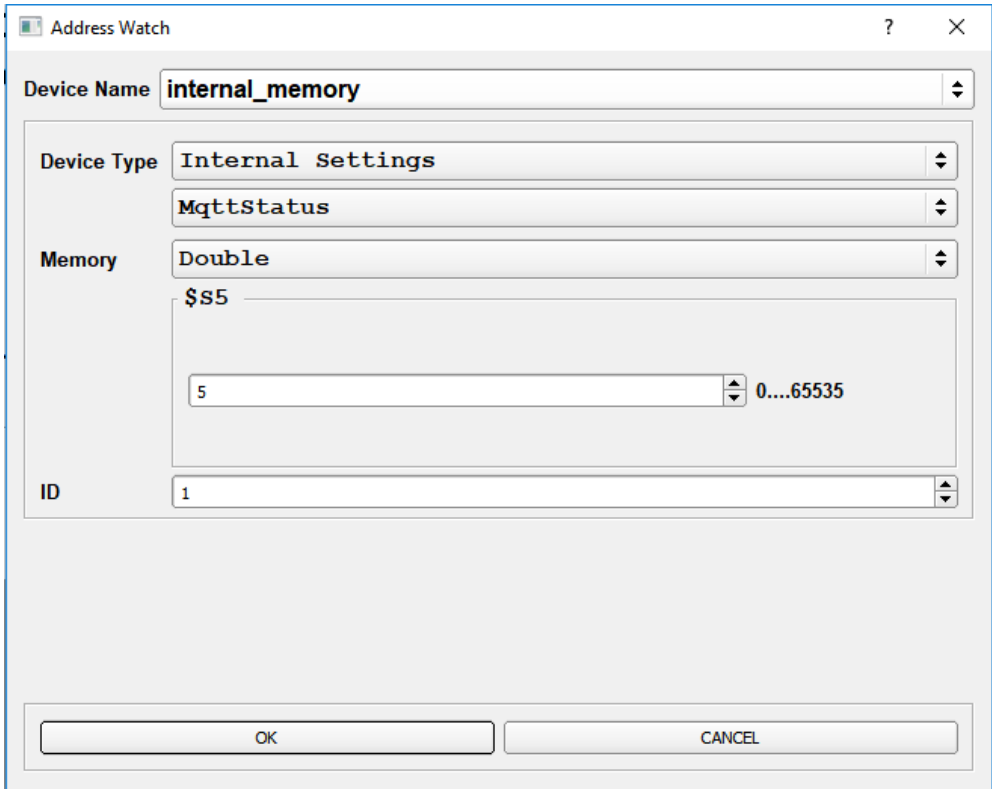
I.6. MqttStatus

The MqttStatus address is the address that indicates the connection status to the MQTT server. The connection states to the Mqtt database server are as follows;

- -1: unsuccessful
- 0: connecting
- 1: connected

You can follow the steps below to view and edit the MqttStatus address status.

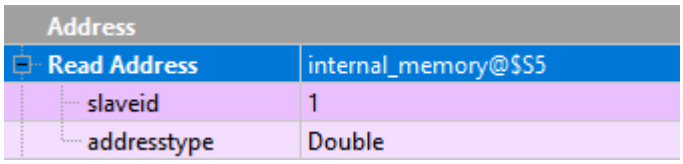
- Drag and drop the element you want to use to the page of the project you are using in Proop Builder.
- Edit Element, Features list->Address->read at click the icon displayed on the right. [...]. A new window will open as below.



Picture 109: Address Watch(MqttStatus)

- Select device name ***'internal_memory'*** .
- Select device type '**Internal Settings**' and select '**MqttStatus**' from the drop-down menu, the memory is displayed as '$S5'.

- Type the device ID in the ID field and click the '**OK**' button. The Write address field appears as follows.



J) Create An Application

To create an application at this part, the steps to be done will be explained.

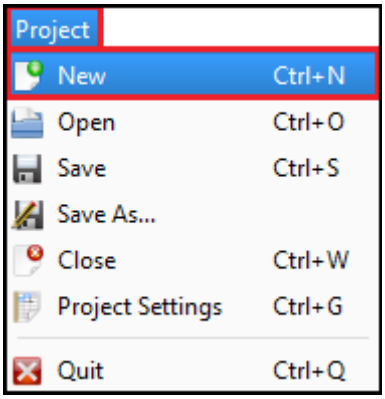
Action to be performed;

- Add a new project and a device.
- Edit connection points information of the device.
- Add a new page of the project and add desired the element tools.
- Define the read or write address of the inserted element.
- Edit the properties section and visual of the element tool.

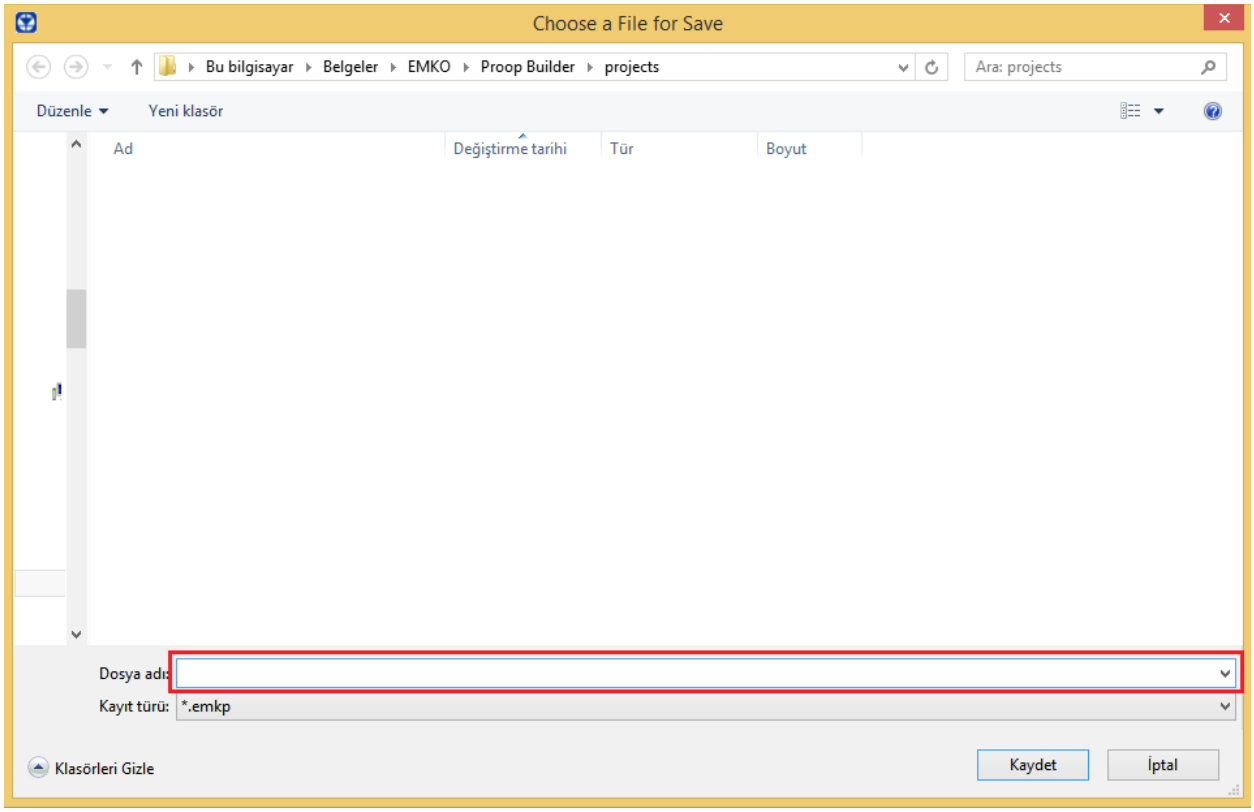
J.1. Create A New Project

To create a new project;

- Click the project from menu tool and click the ***'New'*** from is the opened sub menu



- Write a new project name the ***'Folder Name'*** field and save.

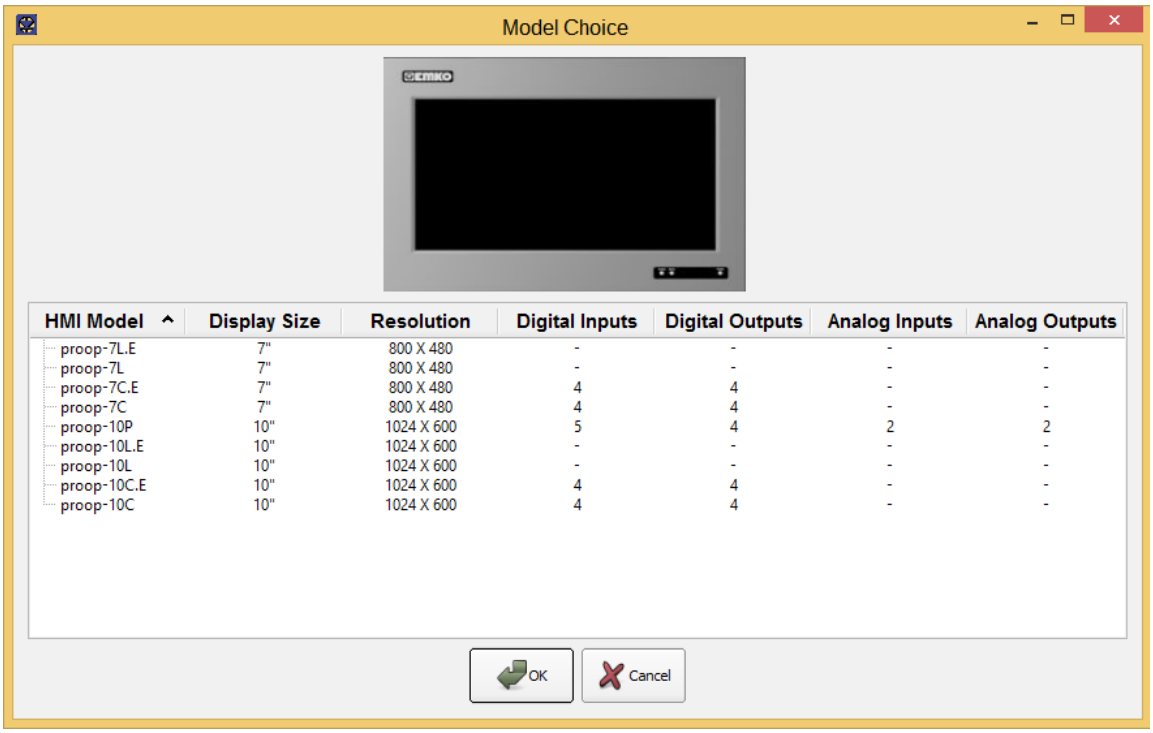


Picture 110: Project-1

The form window opens after saving the project.

- Select a model as below and click the ***'Save'*** button.

Models are explained in detail under the heading of ***'model list'***.



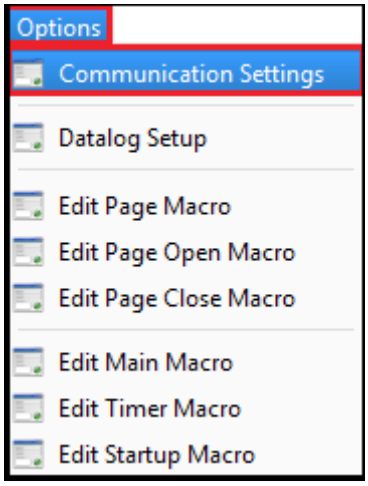
Picture 111: HMI Models

- Screen editor is closed and program is restarted and the added project is opened automatically.

J.2. Add A New Device

To add a new device at the project;

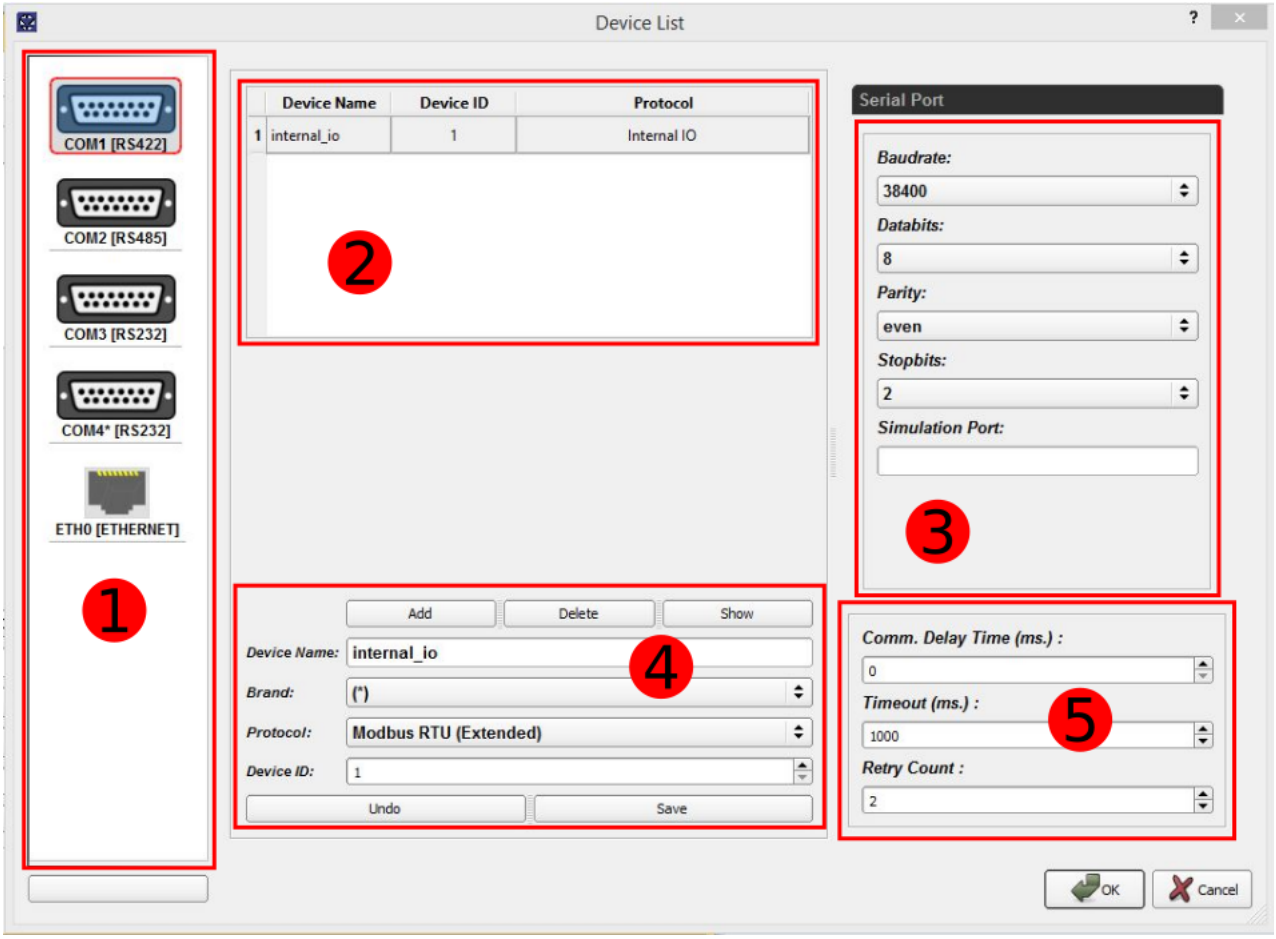
- Click the options from menu tool and click communication settings from sub menu.



Picture 112: Options->Communication Settings

- The communication settings that are opened will contain the setting information for about the devices to be added

- The communication settings are as shown in Picture-73.



Picture 113: Device Lists

- Select the **'connection point of the device'** to be added from the field number 1.

The list of connection point options includes COM1 (RS232), COM2 (RS485), COM3 (RS232), COM4 (RS232) and ETH0 (ETHERNET).

You can access detailed information of the connection points from 'Pin Connections'.

- Enter the device name, brand, protocol and deviceID information from field number 4 and click '***Add***' button.
- The added device is listed in field 2 and edit the serial port settings from field number 3.

The simulation port field in the serial settings specifies the PC comport to be used during online simulation.

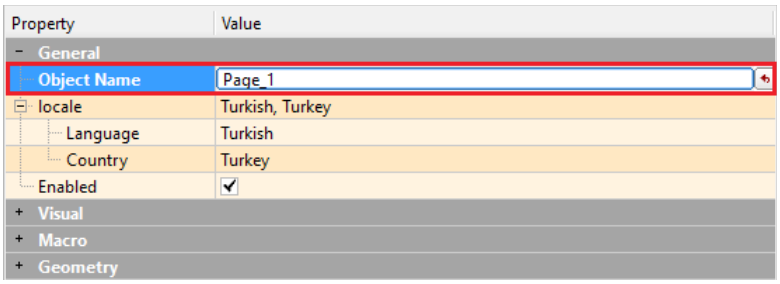
- Finally, edit the options for the connection in field number 5 and click the '***Save***' button to update the device information.

J.3. Add A New Page

To create a new page (form);

- Click on the '***Create New Form***' icon from the toolbar at the top of the editor screen.

- The screen editor section displays the form screen named Page_1 by default.
  The page can be updated from the '***Object Name***' field in the general section of the list of name properties
- Pages can increase at he desired count.



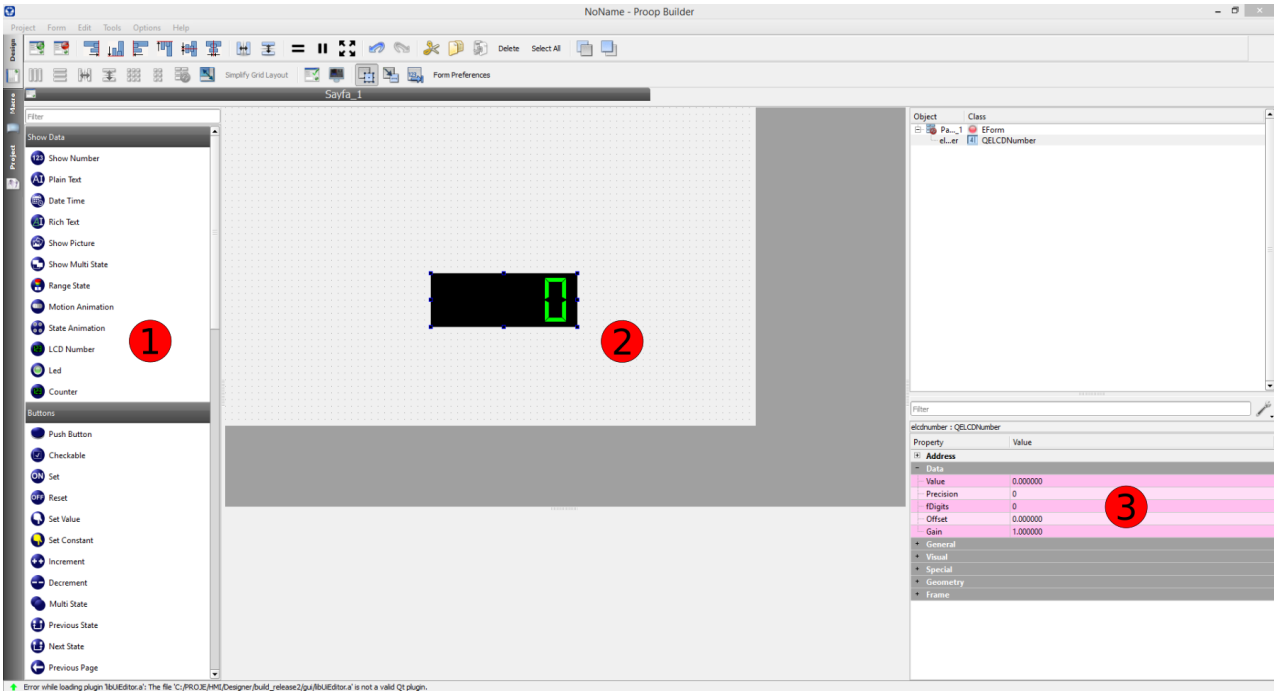
J.4. Add An Element Tool And Edit Property List

Adding elements to the page and editing the list of properties will be explained with examples.

Example-1(Lcd Number)

In example-1, data will be displayed from the address field defined by the LCD number element. After incrementing or decrementing the data value with the help of the buttons/Increment value-Decrement value, writing at the address will be done.

J.4.1. Define Read / Write Address Of Element

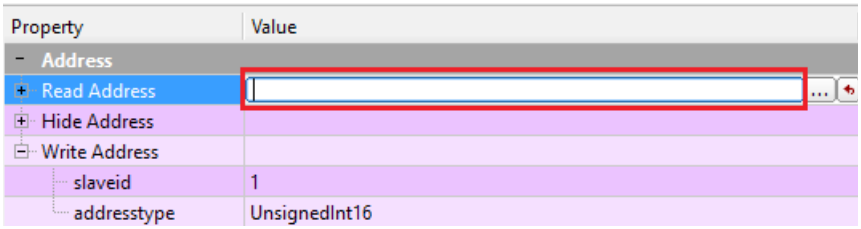


Picture 114: Screen Editor

The LCD number element is specified as the element to be used for data display.

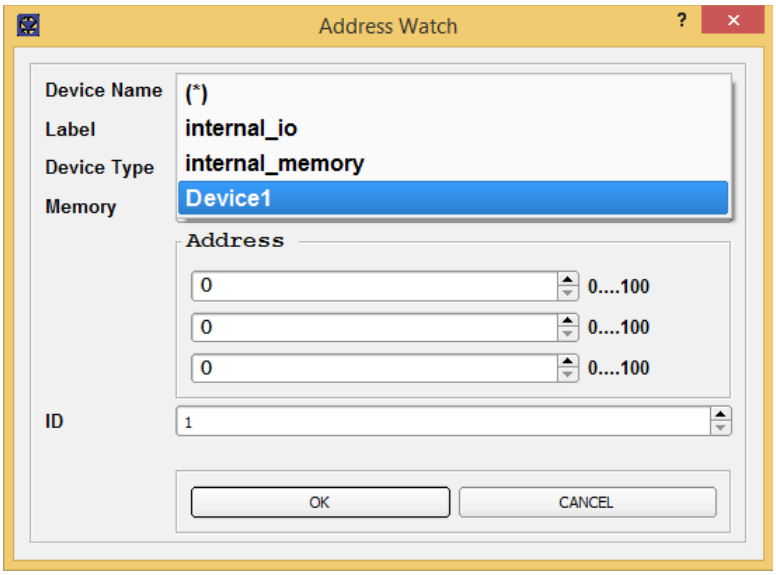
The decrease button and the increase button will be used to set the value at the write address field after the value changed operation.

- In section 1, click on the element you want to use and drag and drop the section number 2.
- Click on the '***Read Address***' field in the address field from the list of properties number 3.



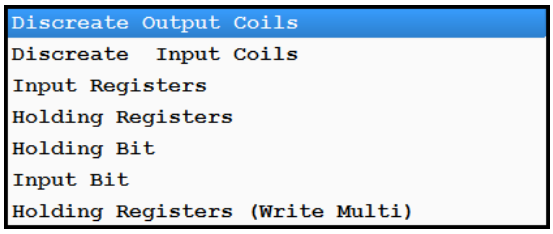
-  When the icon on the left is shown, click on the icon and address watching form will open as the following Picture-71.

- Select the device named '***Device1***' that is added in the device name field

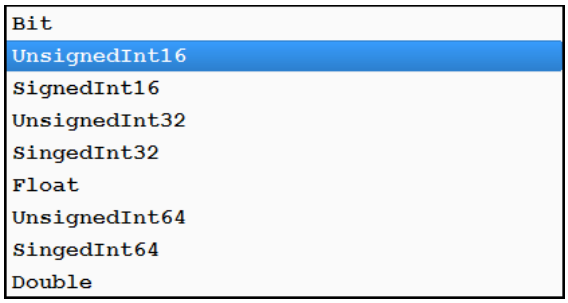


Picture 115: Address Watch

- The device type lists the functions of the access addresses in the memory area and the required device type is determined.
- You can access device details under the heading Device types.



- Listed the memory and value type is specified the desired type.

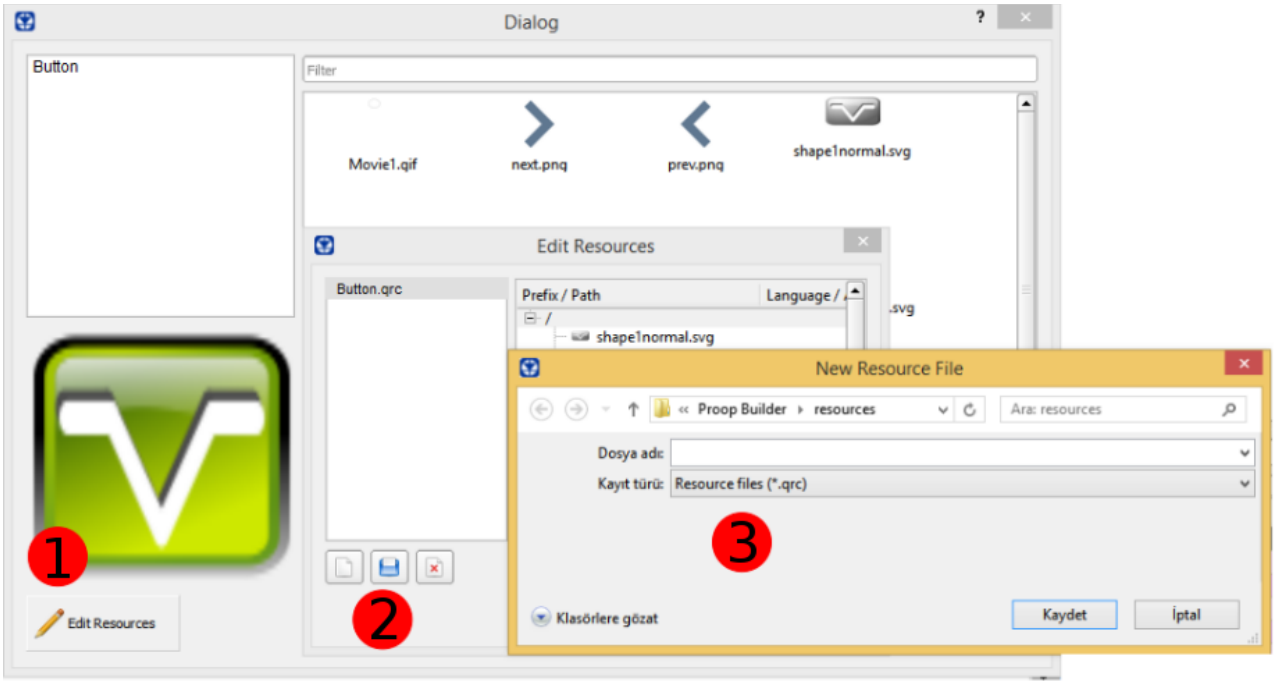


- Enter ID, deviceID of the device and click ***'okey'*** button as the read address value.

The above operations is applied at the increase button and the decrease button. So that, this is done by writing the changed data value with the help of buttons.

J.4.2. Add An Image Of Element

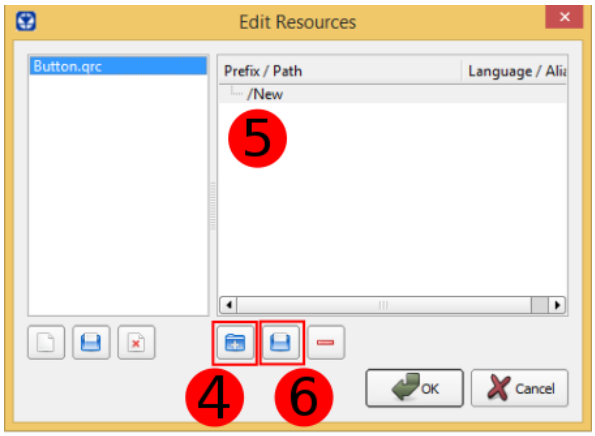
To add an image on the buttons;



Picture 116: Resources

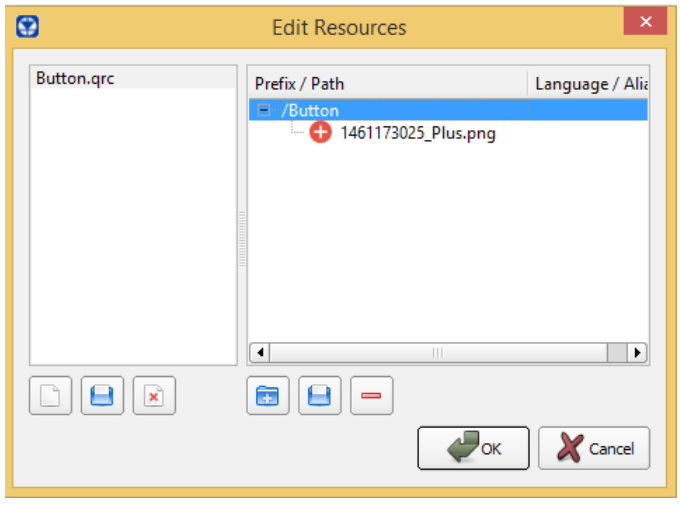
- First, a new library will be created to add images to the buttons. Click on the tools resource editor in the menu bar for this.
- Click on ***'Edit resources'*** from field 1.
- Click ***'Create New Resources File'*** in field 2 from the new window that opens.
- Define ***'File name'*** in field 3 again from the new window that opens.

- **'Create New Resource File'** windows is closed and you are returned to the **'Edit Resources'** window.



Picture 117: Edit Resources

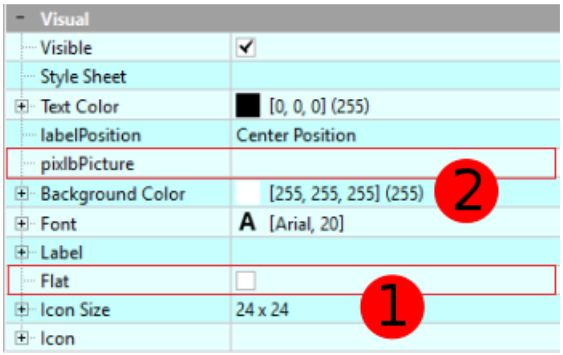
- Click ***'Add Prefix'*** from the field number 4 to insert an image.
- Define the new path name in field number 5.
- Click the ***'Add Files'*** from the field number 6.
- Select the image on opened the **'add files'** window and click the **'Open'** button.
- After closing the window, go back to the **'edit resource'** window and click on the ***'OK'*** button and the image file is created.



Picture 118: Edit Resources

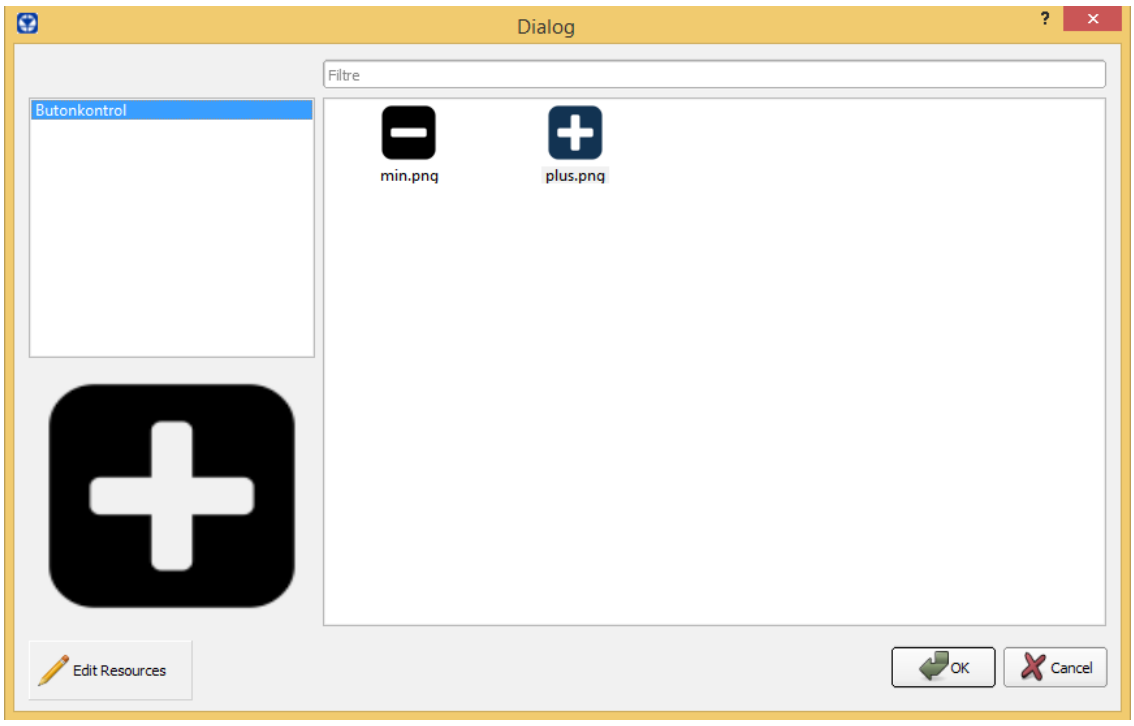
- To load image into the buttons, select ***'Flat'*** field number 1 .

- Once click on the ***'Picture'*** field number 2, click on the icon on the right and click on it.



- The new window that opens is the image files window. Select what you want to upload as in Picture-75.

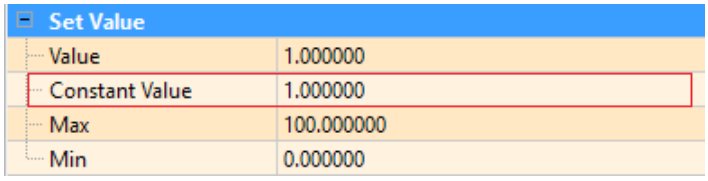
For the example-1 (Lcd Number) made, another value decrement button is added and the image uploading process is completed.



Picture 119: Edit Resources

To define the amount of the increase or decrease buttons;

- The **‘constant value’** field is shown below. Set the desired amount from this area.



According to the value in the constant value field, the lcd number element data value changes as the button is clicked. This value is written to the write address.

J.4.3. Define States Of Element

The operation of determining the state of the elements will be explained with Example-2 .(Multiple Status Indicator Alarm-Running).

Example-2(Multi State)

The multi-state display element will be used to display different properties for each state

Actions to be performed;

- Create a new project. It is explained in detail under the heading **'Create A New Project'**.
- Add a device. It is explained in detail under the heading **'Add A New Device'**.
- Add a new page. It is explained in detail under the heading **'Add A New Page '**.
- Add the desired element tool (multi state). It is explained in detail under the heading **'Add An Element Tool And Edit Property List'**.
- The read address field is defined in the multi state element. It is explained in detail under the heading **'Define Read / Write Address Of Element'**.
- Determine view or function for each state of the address read.

Selection of active status visual property selection;

Picture 120: Multi State(State 0)

Picture 121: Multi State(State 1)

- You can disable the **'visible'** field selection and use the element hiding feature according to the current state.
- You can use these properties according to the active status by clicking on the ***'Text'*** field and writing text, alignment, font, font color, background color.
- You can use this properties according to the active state by selecting the desired picture from the resources opened by clicking on the ***'pixlbPicture'*** field.

Visual	
Visible	☑
Style Sheet	
Text Color	[0, 0, 0] (255)
labelPosition	Center Position
pixlbPicture	
Background Color	[255, 255, 255] (255)
Font	A [Arial, 20]
Label	
Flat	☐
Icon Size	24 x 24
Icon	

To determine the state;

- Set the number of states from the ***'nStates'*** field. In Example-2, the number of states is entered as 2.
- Define the value of the active status you want to edit in the ***'Current State'*** field.
- For each state, enter the state property as the number of states.

The ***'Image'***, ***'Background Color'***, ***'Font'*** and ***'Text'*** fields are edited for the current state 0 and are displayed on the screen as an alarm.

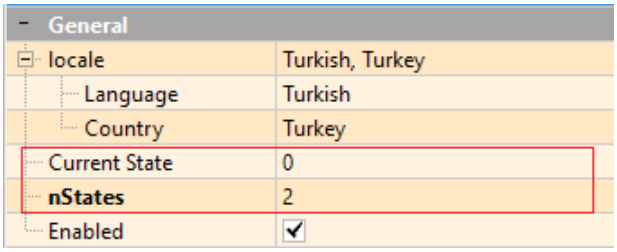
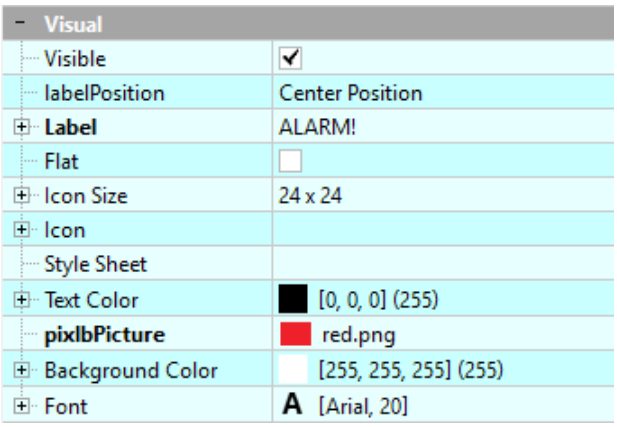
State	Current State:0
Properties List **General Section**	
Properties List **Visual Section**	
Multi State View	

Table 43: Multi State(Current State:0-Alarm)

The ***'Image'***, ***'Background Color'***, ***'Font'*** and ***'Text'*** fields are edited for the current state 1 and are displayed on the screen as running.

State	Current State:1
Properties List **General Section**	Current State: 1 nStates: 2
Properties List **Visual Section**	Visual Visible: ☑ Style Sheet: Text Color: [0, 0, 0] (255) labelPosition: Center Position **pixlbPicture**: green.png Background Color: [255, 255, 255] (255) Font: A [Arial, 20] **Label**: RUNNING Flat: ☐ Icon Size: 24 x 24 Icon:
Multi State View	RUNNING

Table 44: Multi State(Current State:1-Running)

Example-3 (Multi State-Display if Alarm)

A state will be hidden and other state will be displayed with the multi state element tool.

Picture 122: Multi State

(Current State: 0-Hide)

Picture 123: Multi State

(Current State 1-Visible)

Disable the ***'visible'*** field of the visual section to state 0.

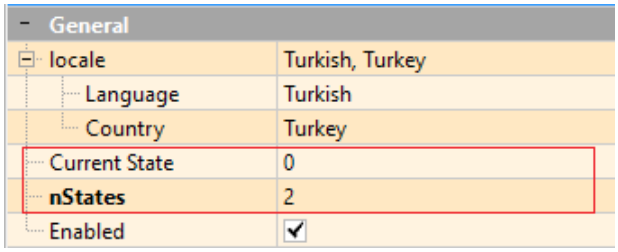
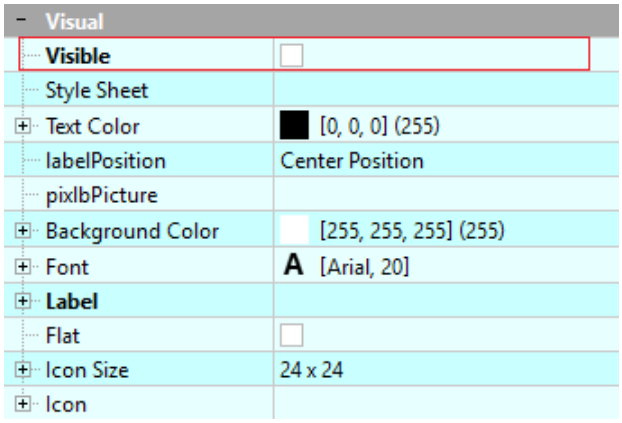
State	State: 0
Properties List **General Section**	
Properties List **Visual Section**	
Multi State View	**Hidden element tool**

Table 45: Multi State(State: 0-No alarm)

Edit the ***'pxlbPicture'***,***'Background Color'***, ***'Font'*** and ***'Label'*** fields**.** Element tool is display as an alarm.

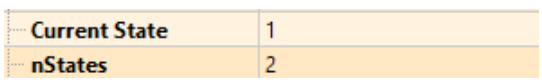
State	State: 0
Properties List **General Section**	
Properties List **Visual Section**	
Multi State View	

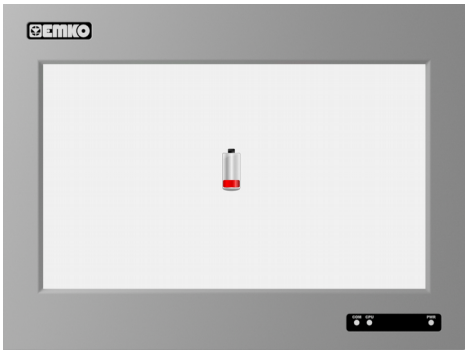
Table 46: Multi State(State: 0-**There is an alarm*****)***

Example-4(Range State)

Actions to be performed;

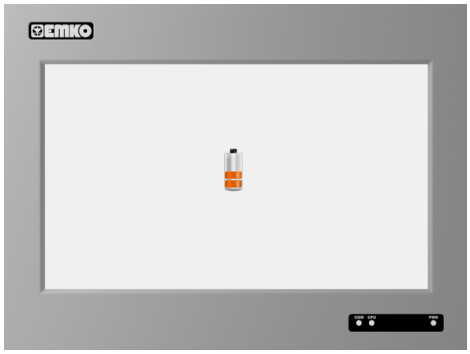
- Add a new page the current project. It is explained in detail under the heading **'Add A New Page '**.
- A range state element tool is add the page. It is explained in detail under the heading **'Define Read / Write Address Of Element'**.
- Determine the range values for for each state of the read address.

The current state of the battery will be displayed by using a range state element



Picture 124: Range State

(Current State: 0-Low)

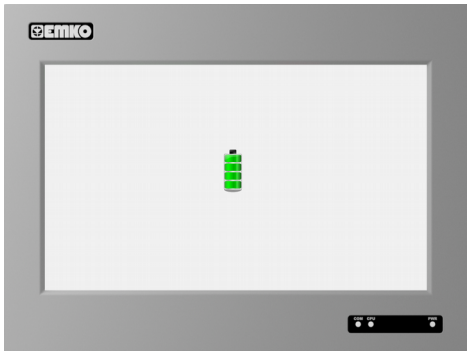


Picture 125: Range State

(Current State: 1-Half)

Picture 126: Range State

(Current State: 2-*Quarter-full*)



Picture 127: Range State

(Current State: 3-Full)

To determine the state;

- Set the number of states from the ***'nStates'*** field. This example, the number of states is entered as 4.
- Set the status value you want to edit in the ***'status'*** field.
- Define a value in the ***'Range'*** field for each state and set the range limit. The status property is displayed until that limit.
- The visual property is edited by the number of states.

After the value is set in the number of states field;

1. While the value in the ***'status'*** field is ***'0(zero)'***, click on the ***'pxlbPicture'*** field in the visual section and select the desired image from the resources.
While the ***'status'*** field value is ***'0(zero)'***, define a value in the ***'range'*** field from the special section and so that the image is displayed on the screen until the defined limit.
2. Increase the status value by moving the value in the ***'status'*** field with the mouse or click on the ***'status'*** field and type ***'1'*** in the status field as the value. Click the ***'pxlbPicture'*** field in the visual section and select the desired image from the resources.

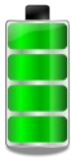
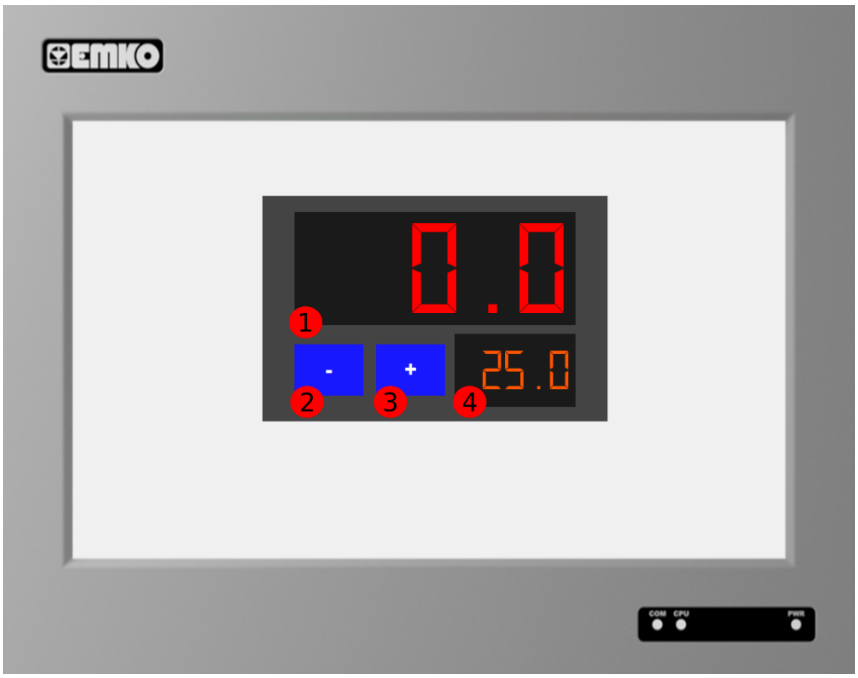
	Low	**Half**	**Quarter Full**	**Full**
nStates	4	4	4	4
State	0	1	2	3
Range	25	50	75	100
pxlbPicture				

Table 47: Range State(States)

Example-5 (Macro)



Picture 128:Macro Application

Element Tool Properties

	Used Element Tool	Read Address	Property
1	LCD Number	internal_memory@$0 (Internal volatile memory address 0)	-
2	LCD Number	internal_memory@M0 (Internal volatile memory address 0)	-
3	Button/ Decrement	internal_memory@M0 (Internal volatile memory address 0)	Step Value:0.1
4	Button/ Increment	internal_memory@M0 (Internal volatile memory address 0)	Step Value:0.1

Table 48: Read Address of the Used Element Tools

Code Main Macro

```
1       func main()                     // Function main macro
2       $0 = %IW0 / 10.0;               // converting decimal display
                                        // of the read value (%IW0)
3       if $0 < $M0
4        %QX0.0 = 1;                    // Digital output 1 enable
5       else
6        %QX0.0 = 0;                    // Digital output 1 disable
7       endif;
8        endf                           // function end
9       endp                            // end code
```

Application temperature control is created in example-5.

The read temperature is displayed at the by element tool first.

In the second element tool,the set value is displayed by the help of the increase and decrease buttons.

The temperature value is read as non decimal of macro code at the row 2 %IW0 from the analog input address. For example, the read value is 245 for 24.5°C value .

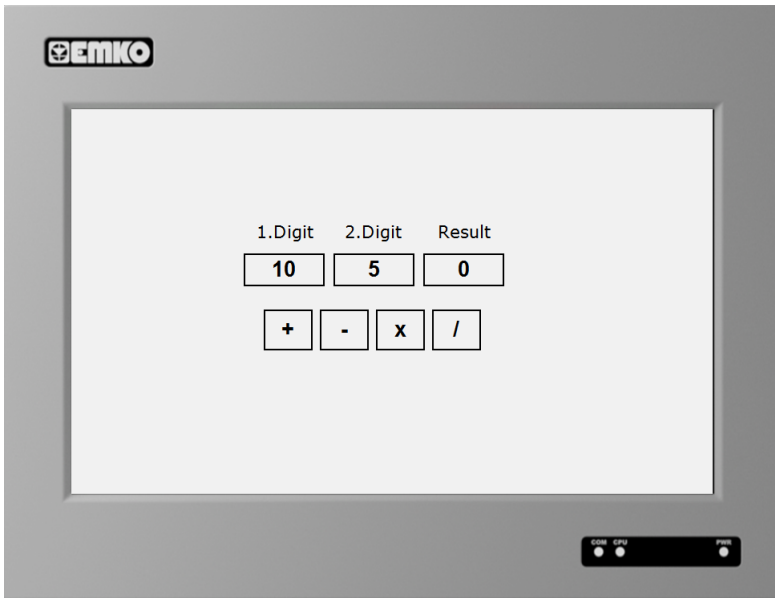
To display this value as a decimal value, the divided value of the set the value from address% IW0 divided by 10 is assigned to address $0 in the row second.

So, the 245 value is displayed as 24.5

The set value at $M0 is compared with the active value at address is $0 in the row third, and if the active value is less than the set value, % QX0.0 digital output is activate the row fourth. If The active value isn't less than disable the digital output.

The main macro period is set to 100 milliseconds from the settings of the project.

Example-6(Macro Application-2)



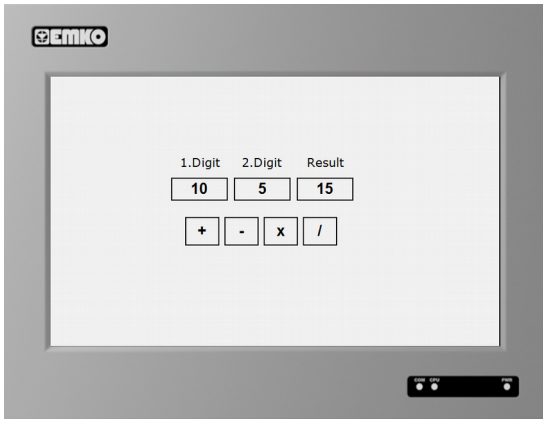
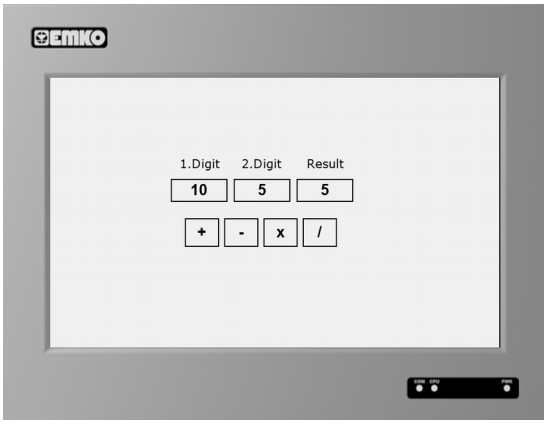
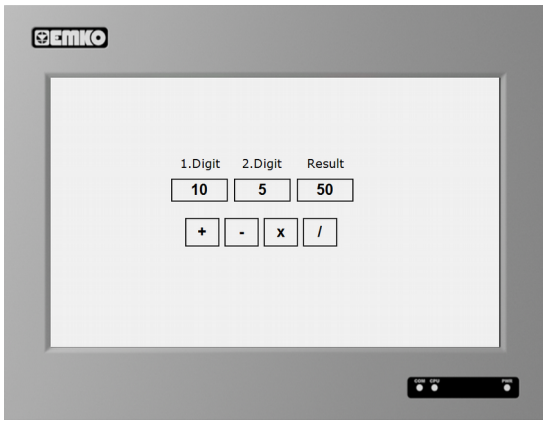
Picture 129: Macro Application

Four operational scenarios were created in example-6.

Value is entered with the value input element tools used for 1.digit and 2.digit.

Then the push buttons used for +, -, x, / (addition, subtraction, multiplication, division).

The first address is $0, the second address is $1, and the result address is $3.

Used Button	Macro Code Executed	Result
For button '+' if button is clicked	1 func main() 2 $3 = $0 + $1; 3 endf endp 4	
For button '-' if button is clicked	1 func main() 2 $3 = $0 - $1; 3 endf endp 4	
For button '*' if button is clicked	1 func main() 2 $3 = $0 * $1; 3 endf endp 4	
For button '/' if button is clicked	1 func main() 2 $3 = $0 / $1; 3 endf 4 endp	

Example-7

Picture 130: Macro Application3

This application LEDs are blinking according to the bits assigned to the example. The start button is set to address $1 and the bits of the LEDs are set and reset at 500 milliseconds. The $1 address is reset and stopped with the Stop button.

Used Button	Macro Code Executed		Result
For button start if button is clicked	1	func main()	
	2	$1 = 1;	
	3	endf	
	4	endp	
For button stop if button is clicked	1	func main()	
	2	$1 = 0;	
	3	endf	
	4	endp	

Periodic macro code:

```
1       func main()
2       if $1 == 1                          //If the start button is pressed
3        if $2 == 0                         // To turn on the lights in order of
4              $0.0 = 0;                    //Address of the led 1th the bit $0.0
5              $0.1 = 1;                    //Address of the led 2th the bit $0.1
6              $2 = 1;                      //Go led 2th
7       else
8         endf                              // function end
9        endp                               // end code
10       $0.0 = 1;
11       $0.1 = 0;
12       $2 = 0;
13      endif;
    sleep(500);                             //Light every other 500 ms
    endif;
    endf
    endp
```