



Universidad de Oviedo

Memoria del Trabajo Fin de Máster realizado por

MARTA FERNÁNDEZ GARCÍA

para la obtención del título de

Máster en Ingeniería de Automatización e Informática Industrial

**APLICACIÓN PARA LA ADQUISICIÓN Y
GESTIÓN DE PARÁMETROS ENERGÉTICOS**

**MANUAL DE
PROGRAMADOR**

Enero 2016



INDICE DE CONTENIDO

MANUAL DE PROGRAMADOR	8
1. Audiencia	8
1.1 Aplicabilidad	8
1.2 Propósito	8
2. Utilización de Software Necesario	9
2.1 PCWorx.....	9
2.2 WebVisit	13
3. Descripción de Programa PLC.....	15
3.1 Estructura del Programa	15
3.2 Modo de Ejecución del Programa PLC.	66
3.3 Listado de Variables	69
4. Descripción de Programa Visualización Web.	129
4.1 Login_MACRO.teq.....	129
4.2 Principal_def.teq	131
4.3 Principal_operario.teq	132
4.4 Sistema: Settings_ILC.teq.....	133
4.5 Configuración_Data_Comu	135
4.6 Dispositivos	140
5. Descarga y Preparación de la Aplicación para su Uso.	151
5.1 Descarga de Programa en el Controlador ILC 191 ME/AN	151
5.2 Descarga de Visualización Web en el Controlador ILC 191 ME/AN	154



5.3 Configuración del Servidor FTP para la Transmisión y Almacenamiento de Archivos..... 157

5.4 Creación y Configuración de Bases de Datos en Servidor XAMPP para el Almacenamiento de Datos. 162

INDICE DE ILUSTRACIONES

Ilustración 1 Entorno PcWorx..... 9

Ilustración 2 Selección ILC 10

Ilustración 3 Configuración de Bus 11

Ilustración 4 Area de Trabajo..... 12

Ilustración 5 Variables PDD..... 13

Ilustración 6 Entorno WebVisit..... 14

Ilustración 7 Sistema..... 15

Ilustración 8 Cambiar_IP 16

Ilustración 9 Set_IP 17

Ilustración 10 Conf_Reset_IP..... 18

Ilustración 11 Get_IP..... 18

Ilustración 12 Cambiar_Hora 19

Ilustración 13 Set_Time 20

Ilustración 14 Configuracion_Data_Comu..... 21

Ilustración 15 AV_Empro 22

Ilustración 16 AV_PulsesS0..... 23

Ilustración 17 AV_AnalogTemp 24



Ilustración 18 AV_PM	25
Ilustración 19 AV_IOLink.....	26
Ilustración 20 AV_MBus.....	27
Ilustración 21 bbdd_wRITE	28
Ilustración 22 BBDD	29
Ilustración 23 Comando MySQL	29
Ilustración 24 BBDD	30
Ilustración 25 Act_Periodicamente	31
Ilustración 26 StrBBDD.....	33
Ilustración 27 Sel_Val.....	34
Ilustración 28 SelectTCPRTU	35
Ilustración 29 Datalogger_FTP	37
Ilustración 30 Datalogger_PLC.....	38
Ilustración 31 Envío Fichero a FTP	38
Ilustración 32 Envío Fichero a PLC	39
Ilustración 33 Email:VariablesWVvisit.....	40
Ilustración 34 Email.....	41
Ilustración 35 Settings TCP	42
Ilustración 36 Settings RTU	42
Ilustración 37 Dispositivos	43
Ilustración 38 PulsesS0_main	44
Ilustración 39 ME_Empro	46



Ilustración 40 Modbus TCP	48
Ilustración 41 Array TCP.....	49
Ilustración 42 Modbus RTU	52
Ilustración 43 Array RTU	53
Ilustración 44 ME_PM.....	55
Ilustración 45 PM_Values	57
Ilustración 46 ME_AnalogTemp.....	58
Ilustración 47 ME_IOLink_Estructura	59
Ilustración 48 ME_IOLink.....	61
Ilustración 49 IOLink_Command.....	62
Ilustración 50 ME_Mbus.....	64
Ilustración 51 Rendimiento PLC Tarea por Defecto	67
Ilustración 52 Rendimiento PLC tarea ciclica 20ms.....	68
Ilustración 53 Rendimiento PLC tarea ciclica 50ms.....	68
Ilustración 54 Variables Cambiar_IP	70
Ilustración 55 Variables Set_IP	71
Ilustración 56 Conf_Reset_IP.....	72
Ilustración 57 Variables Get_IP.....	73
Ilustración 58 Cambiar_Hora	74
Ilustración 59 Variables Set_Time	75
Ilustración 60 Variables AV_Empro	77
Ilustración 61 Variables AV_Pulses0.....	78



Ilustración 62 Variables AV_AnalogTemp 79

Ilustración 63 Variables AV_PM 82

Ilustración 64 Variables AV_IOLink..... 83

Ilustración 65 Variables AV_MBus..... 84

Ilustración 66 Variables BBDD_Write 94

Ilustración 67 Variables Act_Periodicamente 95

Ilustración 68 Variables StrBBDD..... 105

Ilustración 69 SelectTCPRTU 106

Ilustración 70 FTP_Settings..... 108

Ilustración 71 Variables Email..... 110

Ilustración 72 Variables Settings_Modbus 111

Ilustración 73 Variables ME_PulsesSO 112

Ilustración 74 Variables Modbus_TCP..... 114

Ilustración 75 Variables Values_Online_Advan..... 117

Ilustración 76 Variables Modbus_RTU 119

Ilustración 77 Variables Modbus_RTU 122

Ilustración 78 Variables ME_PM..... 124

Ilustración 79 Variables ME_AnalogTemp..... 125

Ilustración 80 Variables ME_IOLink..... 127

Ilustración 81 Variables ME_MBus..... 128

Ilustración 82 Login_Macro.teq..... 130

Ilustración 83 Principal_def.teq..... 131



Ilustración 84 Principal_operario.teq 132

Ilustración 85 Settings_ILC.teq 133

Ilustración 86 Tabla Variables PDD Cambio Hora 133

Ilustración 87 Tabla variables PDD Cambio IP 134

Ilustración 88 AV_def.teq 135

Ilustración 89 Tabla Variables PDD AV 137

Ilustración 90 Communication_Settings_def.teq 138

Ilustración 91 Tabla Variables PDD Communication Settings 139

Ilustración 92 Empro600_Seleccion_def.teq 140

Ilustración 93 Tabla Variables PDD Modbus 141

Ilustración 94 Empro600_def.teq 141

Ilustración 95 Tabla Variables PDD Empro_def 142

Ilustración 96 ValoresAvanzados_def.teq 143

Ilustración 97 Tabla Variables PDD Valores Avanzados 145

Ilustración 98 Meters_Analogicos_Temp_def.teq 145

Ilustración 99 Tabla Variables PDD ME_Analog 145

Ilustración 100 Meters_PM_def.teq 146

Ilustración 101 Tabla Variables PDD Meters_PM 146

Ilustración 102 Meters_PulsesS0_def.teq 147

Ilustración 103 Tabla Variables PDD ME_PulsesS0 148

Ilustración 104 Meters_IOLink_Temp_def.teq 149

Ilustración 105 Tabla Variables Meters_IOLink_Temp_def 149



Ilustración 106 ME_MBus.teq 150

Ilustración 107 Tabla Variables ME_Mbus 150

Ilustración 108 Compilación proyecto 151

Ilustración 109 Descarga Proyecto 152

Ilustración 110 PLC Modo RUN 153

Ilustración 111 Entorno WVisit 154

Ilustración 112 Compilacion WebVisit 154

Ilustración 113 Conexión WebVisit 155

Ilustración 114 Descargar programa WebVisit 156

Ilustración 115 Descarga WebVisti correcta 156

Ilustración 116 FileZilla 157

Ilustración 117 Dirección IP servidor FTP 157

Ilustración 118 Conexión Servidor FTP Correcta 158

Ilustración 119 Configuración Servidor FTP 159

Ilustración 120 Creación Usuario FTP 160

Ilustración 121 Configuración Ruta Almacenamiento FTP 161

Ilustración 122 Apache Modo RUN 162

Ilustración 123 Conexión Apache 163

Ilustración 124 Creación BBDD 163

Ilustración 125 Configuracion BBDD 164

Ilustración 126 Configuración Campos BBDD 164

Ilustración 127 BBDD Creada 165



MANUAL DE PROGRAMADOR

1. Audiencia

El presente documento está dirigido a todo personal técnico encargado del mantenimiento de la aplicación, así como, toda aquella persona que pudiera necesitar la comprensión del código de programación utilizado para el desarrollo de la aplicación.

1.1 Aplicabilidad

El presente manual describe la programación realizada sobre el entorno de programación PCWORX para el desarrollo de una aplicación de adquisición, gestión y almacenamiento de datos energéticos, en un PC con SO Windows XP.

1.2 Propósito

Este manual tiene como propósito la descripción de la estructura de programación llevada a cabo para el desarrollo de la aplicación previamente explicada, así como el manejo y gestión de variables, librerías y demás herramientas de programación utilizadas en el proyecto, con el fin de que, cualquier personal técnico y con conocimientos de programación pueda entender y comprender la estructura de programación llevada a cabo.



2. Utilización de Software Necesario

A continuación, se realiza una breve introducción a los principales entornos de programación empleados en el desarrollo del presente proyecto: PCWorx y WebVisit.

2.1 PCWorx

De acuerdo con la normativa IEC 61131-3, se prevé una separación completa entre el hardware y el software. Esta es la razón por la cual se definen tres áreas de trabajo:

- **Programación IEC:** Dónde se escribe el programa.
- **Configuración del Bus:** Dónde se establece la configuración física del controlador.
- **Datos de proceso:** Dónde se realiza la asociación entre el software y los elementos de entrada/salida físicos.

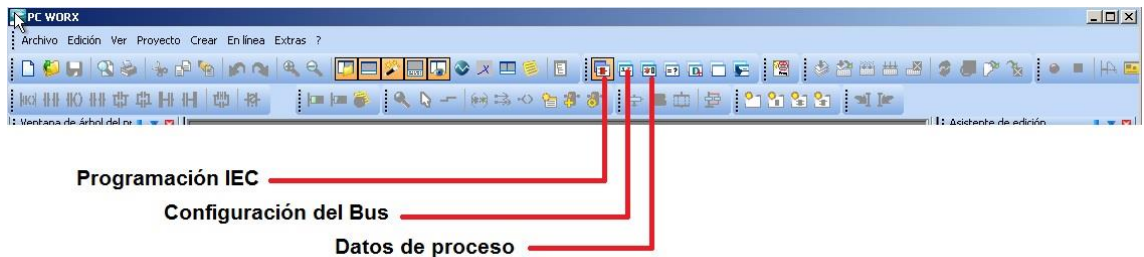


Ilustración 1 Entorno PcWorx



Es en **Programación IEC** dónde se creará un nuevo proyecto. Para ello, es necesaria la selección del controlador que se va a utilizar.

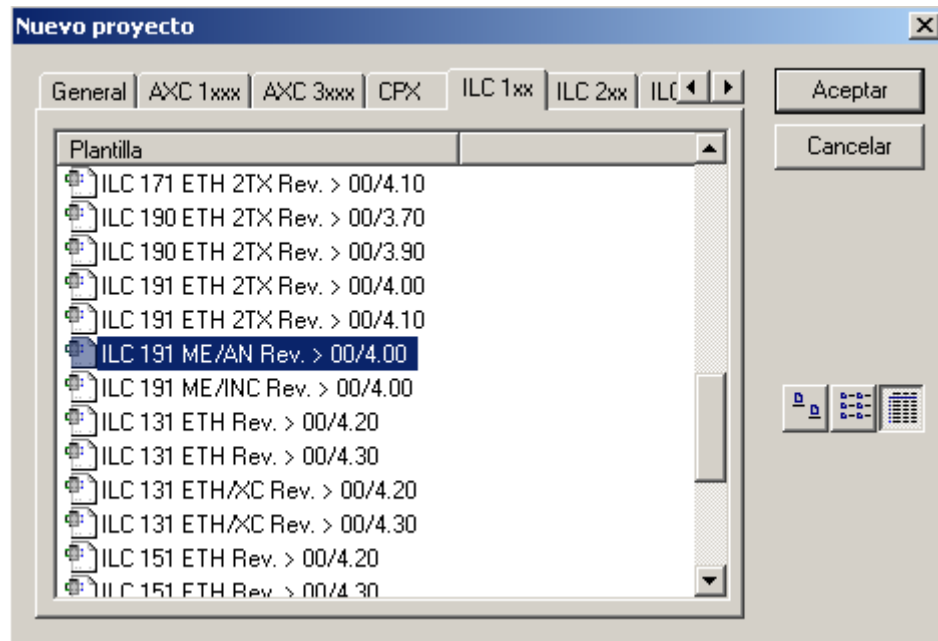


Ilustración 2 Selección ILC



En el área de trabajo **Configuración del Bus** encontrará toda la información relativa al proyecto. Es en este espacio dónde deberá configurar la dirección IP del controlador así como el conjunto de módulos y terminales(módulo IOLink, Mbus, etc) instalados en el PLC, mediante el Catalogo de Dispositivos.

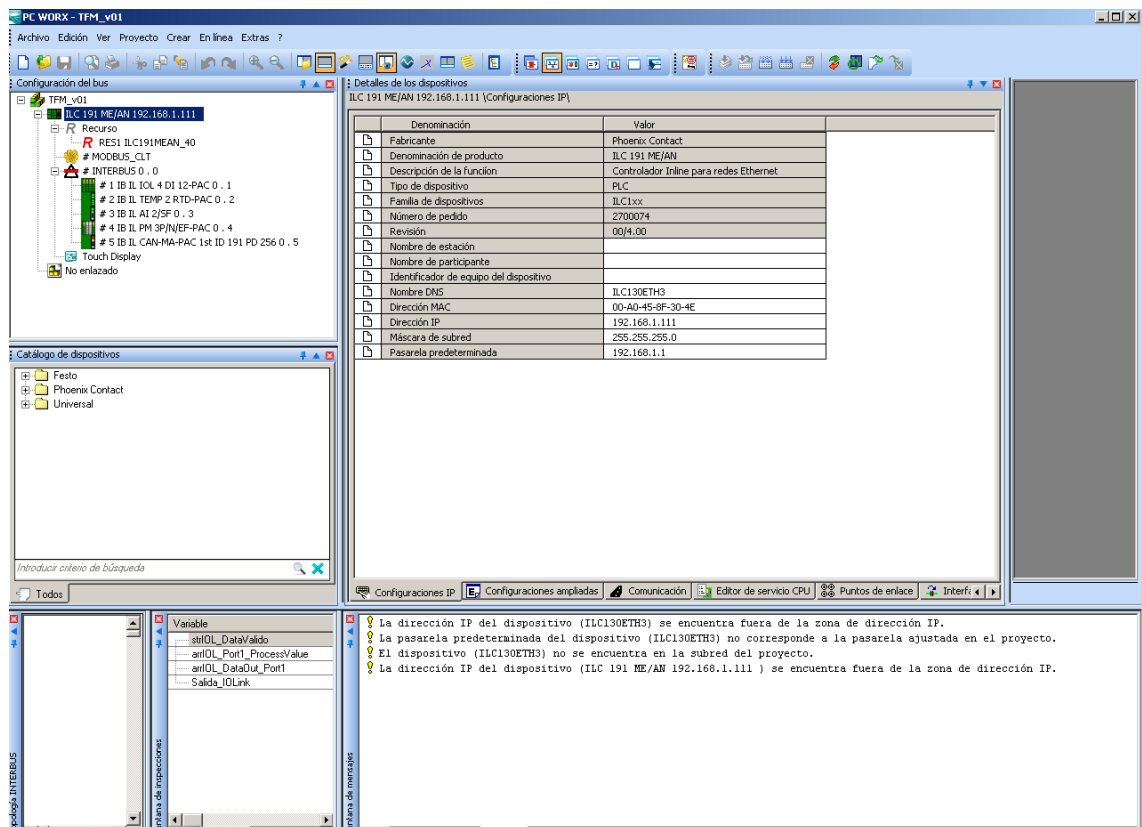


Ilustración 3 Configuración de Bus



Finalmente, es en el área de trabajo **Datos de proceso** dónde se han de asociar las variables definidas en el software a los puntos de entrada/salida disponibles en los módulos de la configuración hardware

The screenshot shows the 'PC WORX - TFM_v01' software interface. The main window is titled 'Asignación de datos de proceso' (Process Data Assignment). It features a tree view on the left for 'Símbolos/Variables' (Symbols/Variables) and a tree view on the right for 'TFM_v01' hardware configuration. Below these is a table mapping process data to hardware points.

Símbolo/Variable	Tipo de datos	Dato de proceso	Dispositivo	Dato de proceso	I/Q	Tipo de datos	Byte Bit
V007	BOOL		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	COM - State Port 1	I	BOOL	0.4
V008	BOOL		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	COM - State Port 2	I	BOOL	0.5
ariA196_IL	IO_Link_AR...	# 1 IB IL IOL 4 DI 12-PAC 0 . 1 1-AI 96	# 1 IB IL IOL 4 DI 12-PAC 0 . 1	COM - State Port 3	I	BOOL	0.6
ariA096_IL	IO_Link_AR...	# 1 IB IL IOL 4 DI 12-PAC 0 . 1 1-AO 96	# 1 IB IL IOL 4 DI 12-PAC 0 . 1	COM - State Port 4	I	BOOL	0.7
V009	BOOL		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	SIO IN 1	I	BOOL	1.0
V010	BOOL		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	SIO IN 2	I	BOOL	1.1
V011	BOOL		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	SIO IN 3	I	BOOL	1.2
ariSysModul	IO_Link_AR...		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	SIO IN 4	I	BOOL	1.3
V012	WORD		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	PD-Valid Port 1	I	BOOL	1.4
wIOL_ProcessValue_Port1	WORD		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	PD-Valid Port 2	I	BOOL	1.5
ariIOL_Port1_ProcessValue	IO_Link_AR...		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	PD-Valid Port 3	I	BOOL	1.6
ari_W_RESULT	IO_Link_AR...		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	PD-Valid Port 4	I	BOOL	1.7
ariMbusInput	EDCL_ARR...	# 5 IB IL CAN-MA-PAC 1st ID 191 PD 256 0 .	# 1 IB IL IOL 4 DI 12-PAC 0 . 1	IOL 1 IN	I	WORD	2.0
ariMbusOutput	EDCL_ARR...	# 5 IB IL CAN-MA-PAC 1st ID 191 PD 256 0 .	# 1 IB IL IOL 4 DI 12-PAC 0 . 1	IOL 2 IN	I	WORD	4.0
Salida_IOLink	REAL		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	IOL 3 IN	I	WORD	6.0
DataLogReal_V0_18_14	REAL		# 1 IB IL IOL 4 DI 12-PAC 0 . 1	IOL 4 IN	I	WORD	8.0

At the bottom of the window, there is a 'Ventana de mensajes' (Message Window) with the following error messages:

- La dirección IP del dispositivo (ILC130ETH3) se encuentra fuera de la zona de dirección IP.
- La contraseña predeterminada del dispositivo (ILC130ETH3) no corresponde a la contraseña ajustada en el proyecto.
- El dispositivo (ILC130ETH3) no se encuentra en la subred del proyecto.
- La dirección IP del dispositivo (ILC 191 ME/AN 192.168.1.111) se encuentra fuera de la zona de dirección IP.

Ilustración 4 Area de Trabajo



2.2 WebVisit

WebVisit es un paquete software para generar páginas web. En el presente proyecto, se utiliza para visualizar las variables globales del Controlador creadas con PcWorx.

Para que una variable pueda ser utilizada en la página web, se debe declarar como **VAR_GLOBAL**. Además se ha de habilitar la opción “PDD”.

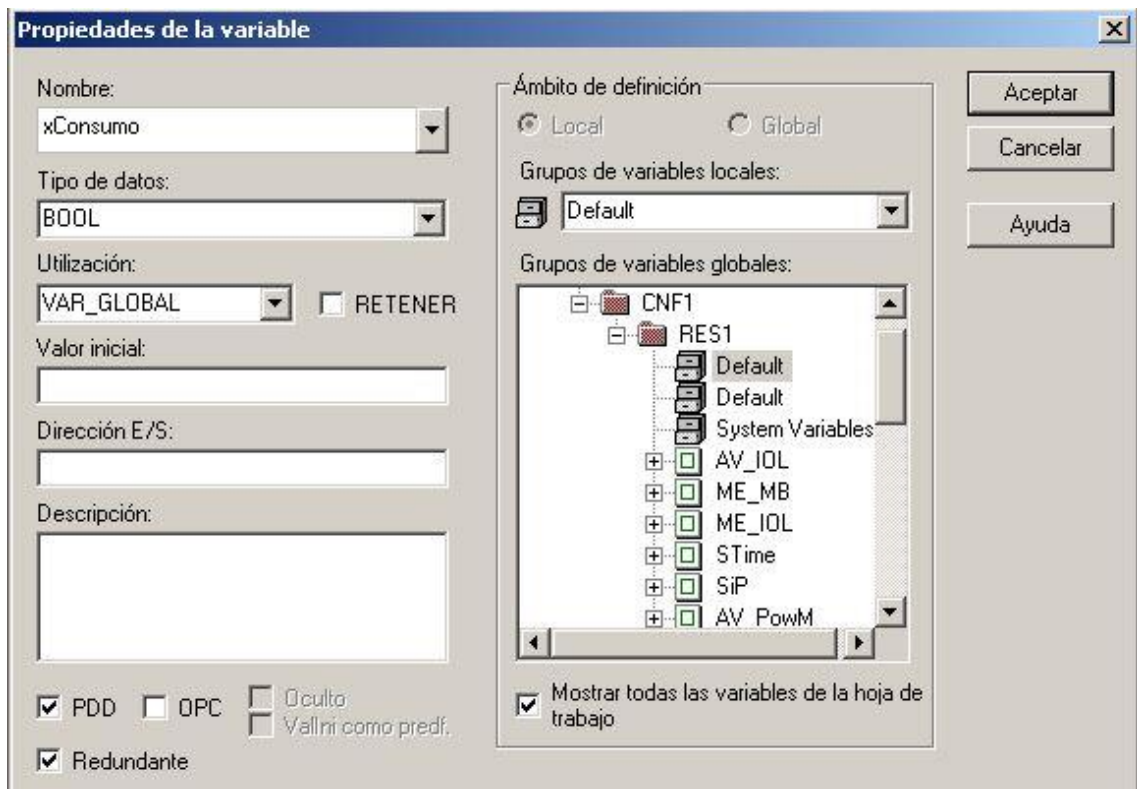


Ilustración 5 Variables PDD

Cuando se compila un proyecto en PCWorx, se genera un fichero “pdd.csv”. Dicho fichero debe incluirse en el proyecto de WebVisit, para así, hacer posible el acceso a las variables del controlador desde la web.



Una vez el proyecto WebVisit ha sido creado y vinculado al proyecto PcWorx, se puede empezar a diseñar la visualización web:

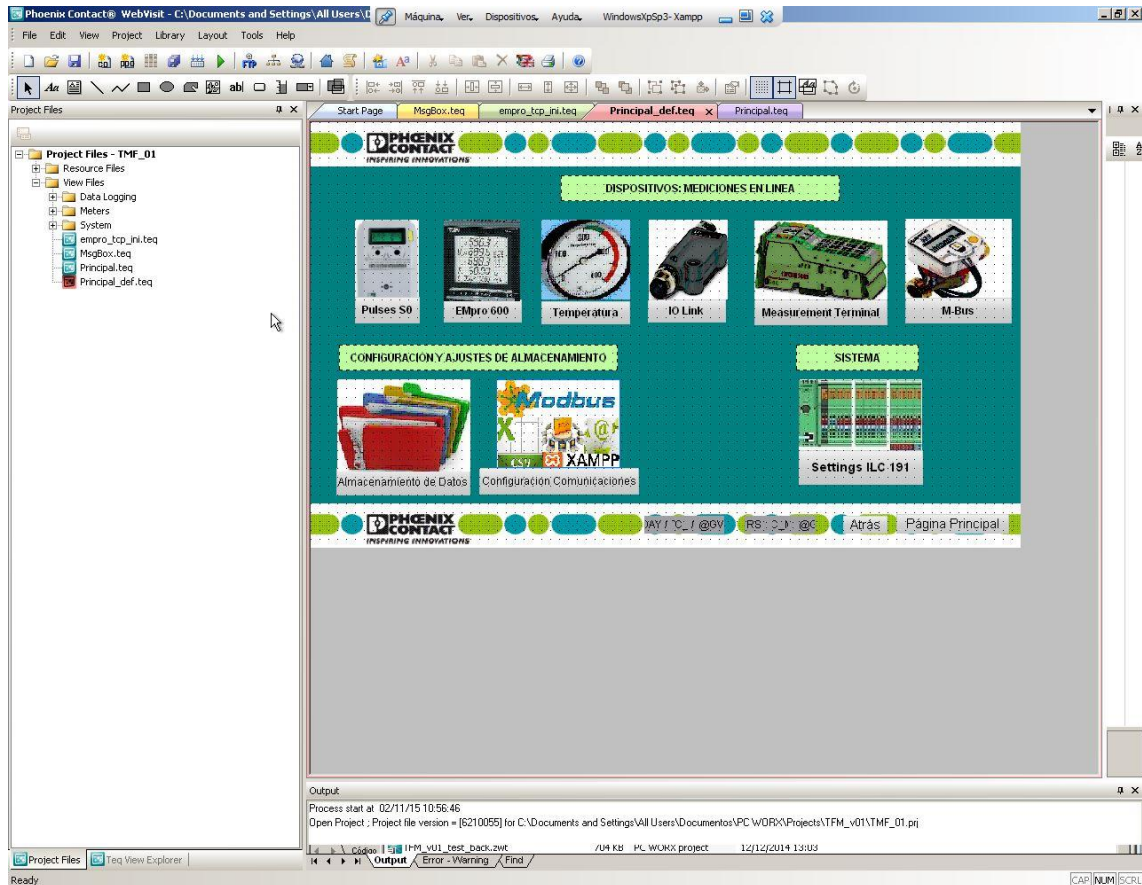


Ilustración 6 Entorno WebVisit



3. Descripción de Programa PLC

3.1 Estructura del Programa

3.1.1 SISTEMA

3.1.1.1 IP

Módulo programado para habilitar el cambio y configuración de direccionamiento IP del PLC. Para ello se requiere acceder a variables internas del PLC para poder modificar los parámetros de dirección IP, máscara de red y puerta de enlace del PLC. Una vez dentro del sistema del PLC, se pueden modificar dichos parámetros. Una vez los cambios deseados sean cargados, es necesario resetear el PLC para que actualice y refresque su dirección IP. A continuación, se exponen los diferentes scripts y funciones programados para realizar este proceso.

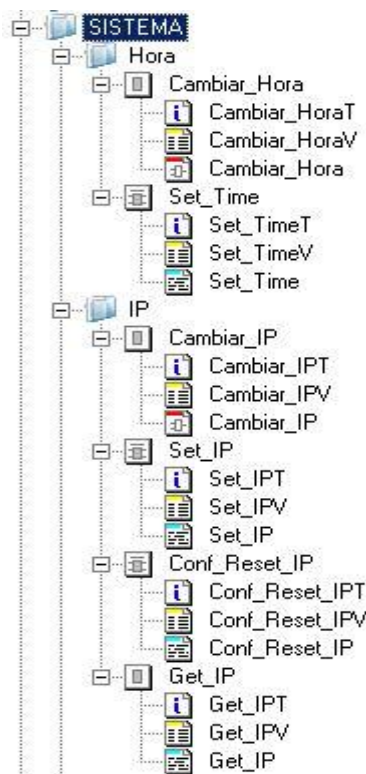


Ilustración 7 Sistema



3.1.1.1.1 Cambiar_IP

Se trata del programa principal, que realiza las acciones requeridas para realizar un cambio del direccionamiento IP del sistema.

En primer lugar, realiza una llamada a la función Set_IP, que almacena los nuevos valores para el direccionamiento. Una vez adquiridos los nuevos datos, el bloque CPU_CONTROL_1 los envía al PLC.

El siguiente paso realizado es el reseteo del PLC. Para ello, se llama a la función Conf_Reset_IP_1, que almacena los datos de configuración requeridos para esta tarea, y que, posteriormente, el bloque CPU_CONTROL_2 envía al propio PLC.

Finalmente, se dispone de un tercer bloque CPU_CONTROL_3, encargado de leer la dirección actual IP del PLC. En el apartado Get_IP, se explica su funcionamiento.

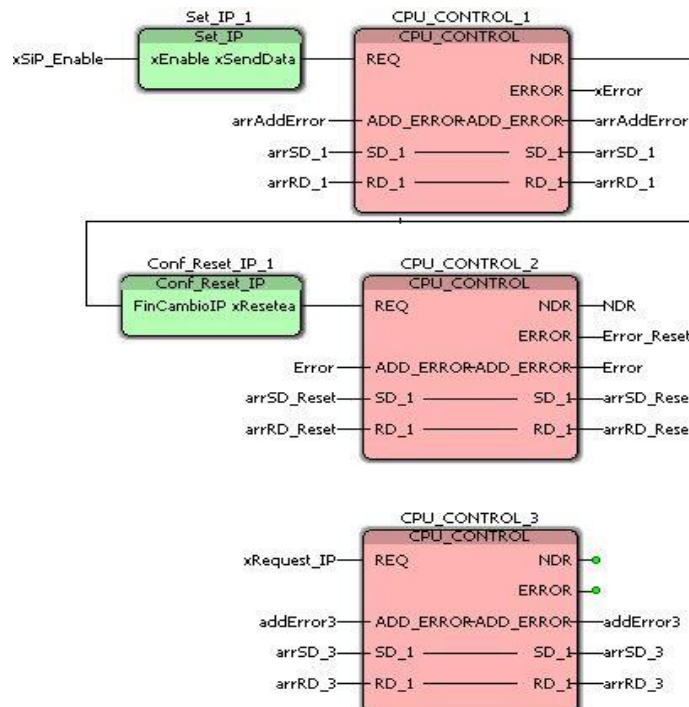


Ilustración 8 Cambiar_IP



3.1.1.1.2 Set_IP

Para la lectura y carga de la nueva dirección IP del PLC existe un array de Words para los datos del sistema, en el que en función de la operación que se quiera realizar en él, es necesario cargar unos valores u otros en las palabras 0 y 1 del array arrSD_1. En este caso, el valor de configuración necesario para el cambio de dirección IP se corresponde con 029B en hexadecimal para la palabra 0 del array, y para la palabra 1 del array el valor 8.

```
arrSD_1[0] := WORD#16#029B;  
arrSD_1[1] := WORD#16#8;
```

El resto de words del array es dónde se cargan los nuevos valores que conforman la dirección IP. De manera que, cada byte de la palabra se corresponde con un parámetro de esa dirección IP. Es decir, cada Word contiene dos parámetros, tal y cómo se muestra en la siguiente ilustración.

```
arrSD_1[2].B1 := IP1;  
arrSD_1[2].B0 := IP2;  
arrSD_1[3].B1 := IP3;  
arrSD_1[3].B0 := IP4;  
arrSD_1[4].B1 := MASK1;  
arrSD_1[4].B0 := MASK2;  
arrSD_1[5].B1 := MASK3;  
arrSD_1[5].B0 := MASK4;  
arrSD_1[6].B1 := GATEWAY1;  
arrSD_1[6].B0 := GATEWAY2;  
arrSD_1[7].B1 := GATEWAY3;  
arrSD_1[7].B0 := GATEWAY4;  
arrSD_1[8].B1 := byte#0;  
arrSD_1[8].B0 := byte#0;  
arrSD_1[9].B1 := byte#0;  
arrSD_1[9].B0 := byte#0;
```

Ilustración 9 Set_IP



3.1.1.1.3 *Conf_Reset_IP*

Función en la que se carga en el array de datos del sistema los valores de configuración necesarios para resetear el sistema, una vez los nuevos parámetros de la IP hayan sido cargados.

```
arrSD_Reset[0] := WORD#16#0ABA;  
arrSD_Reset[1] := WORD#16#0000;
```

Ilustración 10 *Conf_Reset_IP*

3.1.1.1.4 *Get_IP*

Cada vez que el PLC se reinicia, o se enciende se realiza una lectura de su dirección IP. Para ello es necesario cargar en el array de datos del sistema los siguientes datos, tal y cómo muestra la siguiente ilustración.

Este programa se realiza para que una vez realizada la visualización web, el usuario pueda leer la dirección IP que tiene el sistema en cada momento.

```
arrSD_3[0] := WORD#16#029C;  
  
IP1_actual:=arrRD_3[3].B1;  
IP2_actual:=arrRD_3[3].B0;  
IP3_actual:=arrRD_3[4].B1;  
IP4_actual:=arrRD_3[4].B0;  
  
MASK1_actual:=arrRD_3[5].B1;  
MASK2_actual:=arrRD_3[5].B0;  
MASK3_actual:=arrRD_3[6].B1;  
MASK4_actual:=arrRD_3[6].B0;  
  
GWAY1_actual:=arrRD_3[7].B1;  
GWAY2_actual:=arrRD_3[7].B0;  
GWAY3_actual:=arrRD_3[8].B1;  
GWAY4_actual:=arrRD_3[8].B0;
```

Ilustración 11 *Get_IP*



3.1.1.2 Hora

Se programa este módulo para poder habilitar el cambio de la hora del PLC. Para ello, y al igual que para el cambio de IP, es necesario acceder a las variables del sistema con una configuración determinada. Se realiza un programa principal (“Cambiar_Hora”) encargado de cargar los nuevos datos horarios y de enviárselos al PLC. Además se programa una función (“Set_Time”) para leer los nuevos datos introducidos por el usuario. A continuación se explica cada uno de estos scripts.

3.1.1.2.1 Cambiar_Hora

Programa en lenguaje de bloque funcional, en el que se realiza una llamada a la función Set_IP para leer los nuevos datos horarios. Posteriormente, y a través del bloque CPU_CONTROL_2 se envían dichos datos al PLC.

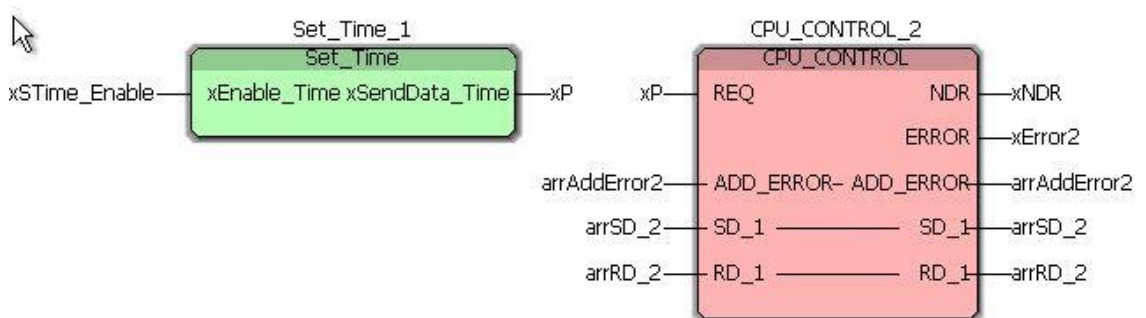


Ilustración 12 Cambiar_Hora



3.1.1.2.2 *Set_Time*

Función programada en texto estructurado, en la que se configura el PLC para un cambio de sus variables de tiempo. Para ello se carga en el array de datos del sistema la siguiente configuración:

```
arrSD_2[0] := WORD#16#0296;  
arrSD_2[1] := WORD#16#6;
```

Una vez preparado el sistema para dicha operación, se cargan los nuevos datos en el mismo array. De manera que, se cargan segundos, minutos, horas, día, mes y año en la palabra 2, y consecuentes, del array “arrSD_2”, respectivamente.

```
arrSD_2[2] :=wSecond;  
arrSD_2[3] :=wMinute;  
arrSD_2[4] :=wHour;  
arrSD_2[5] :=wDay;  
arrSD_2[6] :=wMonth;  
arrSD_2[7] :=wYear;
```

Ilustración 13 *Set_Time*



3.1.2 CONFIGURACIÓN_DATA_COMU

Todo la programación requerida que concierne al almacenamiento de datos y configuración de protocolos de comunicación, así como, configuración de formato de almacenamiento de datos puede encontrarse en este grupo de POUs.

Se distinguen dos partes de programación:

- **ArchiveValues**
- **Communication_Settings**

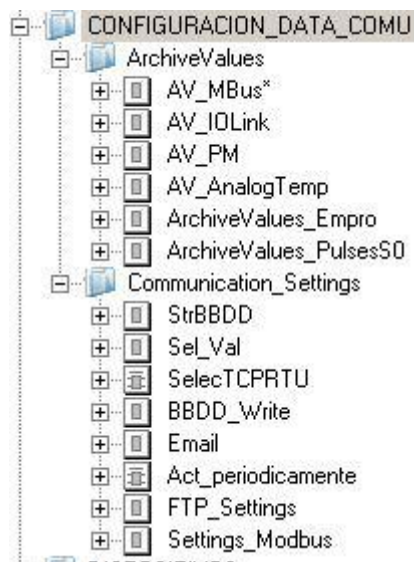


Ilustración 14 Configuracion_Data_Comu



3.1.2.1 ArchiveValues

Grupo de POU's en el que se implementa la programación necesaria para que el usuario de la aplicación pueda seleccionar que parámetros medidos que desea almacenar. Se realiza un script para cada uno de los dispositivos de medida.

3.1.2.1.1 *ArchiveValues_Empro*

Script programado en lenguaje de texto estructurado mediante el cual se habilita la opción de almacenar un parámetro, si este ha sido seleccionado por el usuario, en función también si la comunicación sea por Modbus TCP o RTU.

Se necesita incluir la librería `DataLogger_VO1_18Eval_08` para que, en caso de que un parámetro haya sido seleccionado para almacenamiento, este se pueda enviar y guardar en un archivo `.csv`. Para ello se utilizan los bloques tipo `DataLogReal_VO_18`, para el envío de datos al `.csv`

```
if xTCP then

    if xU12 then
        DataLogReal_VO_18_1(rData:=U12_tcp,
            strColumnName:='U12',
            udtDataLogger:=Data);
        Data:=DataLogReal_VO_18_1.udtDataLogger;
    end_if;

if xRTU then
    if xU12 then
        DataLogReal_VO_18_53(rData:=U12_rtu,
            strColumnName:='U12',
            udtDataLogger:=Data);
        Data:=DataLogReal_VO_18_53.udtDataLogger;
    end_if;
```

Ilustración 15 AV_Empro



3.1.2.1.2 *ArchiveValues_Pulses S0*

Script que permite el almacenamiento y envío de datos a un archivo formato .csv de los datos correspondientes a Pulses S0 que haya seleccionado el usuario mediante el interface web.

Si por ejemplo, el usuario ha seleccionado almacenar la variable Nombre de los contadores, se enviará dicho dato al csv, mediante el uso de los bloques que forman la librería DataLogger_VO1_18Eval_08.

```
if xNombre then
    DataLogStr_VO_18_1(strData:=Entrada,
    strColumnName:='Input',
    udtDataLogger:=Data);
    Data:=DataLogStr_VO_18_1.udtDataLogger;
end_if;

if xConsumo then
    DataLogInt_VO_18_2(iData:=Consumo_sal,
    strColumnName:='Consumo',
    udtDataLogger:=Data);
    Data:=DataLogInt_VO_18_2.udtDataLogger;
end_if;

if xValorPulso then
    DataLogInt_VO_18_3(iData:=Valor_Pulso_sal,
    strColumnName:='Valor del pulso',
    udtDataLogger:=Data);
    Data:=DataLogInt_VO_18_3.udtDataLogger;
end_if;

if xUnidades then
    DataLogStr_VO_18_2(strData:=Unidades_sal,
    strColumnName:='Unidades',
    udtDataLogger:=Data);
    Data:=DataLogStr_VO_18_2.udtDataLogger;
end_if;
```

Ilustración 16 AV_PulsesS0



3.1.2.1.3 AV_AnalogTemp

Script programado en lenguaje de texto estructurado, para el almacenamiento y envío de datos tomados por los sensores de temperatura, en caso que el usuario seleccione dichos datos.

```
if xTemp01 then
    DataLogReal_VO_18_1(rData:=Temperatura01,
        strColumnName:='Sensor_TEMP_Canal1',
        udtDataLogger:=Data);
    Data:=DataLogReal_VO_18_1.udtDataLogger;
end_if;

if xTemp02 then
    DataLogReal_VO_18_2(rData:=Temperatura02,
        strColumnName:='Sensor_TEMP_Canal2',
        udtDataLogger:=Data);
    Data:=DataLogReal_VO_18_2.udtDataLogger;
end_if;
```

Ilustración 17 AV_AnalogTemp



3.1.2.1.4 AV_PM

Script en texto estructurado que habilita el envío y almacenamiento de datos tomados por el módulo de medición de parámetros eléctricos añadido al controlador.

Se enviarán al archivo .csv los datos que hayan sido seleccionados por el usuario. De nuevo, la librería DataLogger_V01_18Eval_08 es necesaria.

```
if xPM_V1 then
  DataLogReal_VO_18_1(rData:=PM_V1(* REAL *),
  strColumnName:='V1_PM'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_1.udtDataLogger;
end_if;

if xPM_V2 then
  DataLogReal_VO_18_2(rData:=PM_V2(* REAL *),
  strColumnName:='V2_PM'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_2.udtDataLogger;
end_if;

if xPM_V3 then
  DataLogReal_VO_18_3(rData:=PM_V3(* REAL *),
  strColumnName:='V3_PM'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_3.udtDataLogger;
end_if;
```

Ilustración 18 AV_PM



3.1.2.1.5 AV_IOLink

Script en texto estructurado que habilita el envío y almacenamiento de datos tomados por el terminal IO-Link conectado al controlador, y mediante el cual los contadores de aire comprimido o todo aquel dispositivo conectado aguas abajo del nombrado terminal envían información y parámetros de lectura al controlador.

Se enviarán al archivo .csv los datos que hayan sido seleccionados por el usuario. De nuevo, la librería DataLogger_VO1_18Eval_08 es necesaria.

```
if xIOLink then
  DataLogReal_VO_18_1(rData:=Salida_IOLink(* REAL *),
    strColumnName:='IOLink_Puerto1' (* STRING *),
    udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_1.udtDataLogger;
end_if;
```

Ilustración 19 AV_IOLink



3.1.2.1.6 AV_MBus

Script en texto estructurado que habilita el envío y almacenamiento de datos tomados por el terminal de Mbus conectado al controlador, y mediante el cual se envían los parámetros referentes a consumo, temperatura, volumen y/o horas de funcionamiento del dispositivo a un fichero csv.

Se enviarán al archivo .csv los datos que hayan sido seleccionados por el usuario. De nuevo, la librería DataLogger_VO1_18Eval_08 es necesaria.

```
if xMbus_horasfuncionamiento then
  DataLogReal_VO_18_1(rData:=lrWorkingTime(* REAL *),
  strColumnName:='Mbus_HorasFuncionamiento'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_1.udtDataLogger;
end_if;

if xConsumoporHora then
  DataLogReal_VO_18_3(rData:=lrConsumoporhora(* REAL *),
  strColumnName:='Consumo_Mbus'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_3.udtDataLogger;
end_if;

if xVolumen then
  DataLogReal_VO_18_5(rData:=lrVolumen(* REAL *),
  strColumnName:='Volumen_Mbus'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_5.udtDataLogger;
end_if;

if xTempIN then
  DataLogReal_VO_18_7(rData:=lrTempIN(* REAL *),
  strColumnName:='Mbus_TempIn'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_7.udtDataLogger;
end_if;

if xTempOUT then
  DataLogReal_VO_18_9(rData:=lrWorkingTime(* REAL *),
  strColumnName:='Mbus_TempOut'(* STRING *),
  udtDataLogger:=Data(* udtDL_V3 *));
  (* udtDL_V3 *)Data:=DataLogReal_VO_18_9.udtDataLogger;
end_if;
```

Ilustración 20 AV_MBus



3.1.2.2 Communication Settings

3.1.2.2.1 BBDD

Conjunto de scripts que incluyen la programación necesaria para la comunicación del controlador con una base de datos, y el envío de datos a la misma.

3.1.2.2.1.1 *BBDD_Write:*

Se trata del programa principal, encargado del establecimiento de conexión del controlador con la base de datos, y el envío de los datos a la misma.

En primer lugar, se realiza una copia de los parámetros introducidos por el usuario, correspondientes a la base de datos.

```
xActivarConexion_BBDD:=BBDD_Habilitar;  
strUser_BBDD:=BBDD_User;  
strPassWord_BBDD:=BBDD_Password;  
strIP_BBDD:=BBDD_IP;  
strName_BBDD:=BBDD_Name;
```

Ilustración 21 *bbdd_WRITE*

En segundo lugar, se establece la conexión con la base de datos mediante la función `DBFL_MySQL_ACCESS_V1_15_1`. El usuario, contraseña, dirección IP, puerto de comunicación de la base de datos, así como, su nombre son los parámetros de entrada de esta función. Una vez la conexión haya sido establecida se activan las salidas `xBBDD_TCP_Ready` y `xBBDD_SQL_Ready`.



```

DBFL_MySQL_ACCESS_V1_15_1(IP_ACTIVATE:=xActivarConexion_BBDD,
DE_ACTIVATE:=xConectar_BBDD,
DE_USER:=strUser_BBDD, (*'user1', *)
DE_PASSWORD:=strPassWord_BBDD, (*'1234', *)
IP_ADDRESS:=strIP_BBDD, (*'192.168.1.110' *) (* STRING *)
IP_PORT:='3306' (* STRING *),
DE_NAME:=strName_BBDD, (*'tfm_v01' (*'tfm_v01' (* STRING *)
TIME_OUT:=time#10s (* TIME *),
(*NO_KEEPAALIVE:=(* BOOL *) (*,
(*RESET:=(* BOOL *) (**)
SQL_IN:=SendBuffer.Data(* DBFL_ARR_BYTE_0_1439 *),
RCV_BUFFER:=ReceiveBuffer(* DBFL_ARR_BYTE_0_1439 *));
(* BOOL *)xBBDD_TCP_Ready:=DBFL_MySQL_ACCESS_V1_15_1.TCP_READY;
(* BOOL *)xBBDD_SQL_Ready:=DBFL_MySQL_ACCESS_V1_15_1.SQL_READY;
(* BOOL *)xBBDD_SQL_DONE:=DBFL_MySQL_ACCESS_V1_15_1.SQL_DONE;
(** DBFL_UDT_MySQL_Server_Info *) (*:=DBFL_MySQL_ACCESS_V1_15_1.SQL_SERVER_INFO;
(** BOOL *)xError_bbdd:=DBFL_MySQL_ACCESS_V1_15_1.ERROR;
(** DINT *)xtcpstatus_bbdd:=DBFL_MySQL_ACCESS_V1_15_1.TCP_STATUS;
(** DINT *) (*:=DBFL_MySQL_ACCESS_V1_15_1.SQL_STATUS;
(* DBFL_UDT_STRING255 *)strBBDD_ErrMsg:=DBFL_MySQL_ACCESS_V1_15_1.SQL_ERRMSG;
(* DINT *)diSizDat:=DBFL_MySQL_ACCESS_V1_15_1.RCV_SIZE;
(* DINT *)diRowCnt:=DBFL_MySQL_ACCESS_V1_15_1.ROW_CNT;
(* DINT *)diColCnt:=DBFL_MySQL_ACCESS_V1_15_1.COL_CNT;
(* DBFL_ARR_BYTE_0_1439 *)SendBuffer.Data:=DBFL_MySQL_ACCESS_V1_15_1.SQL_IN;
(* DBFL_ARR_BYTE_0_1439 *)ReceiveBuffer:=DBFL_MySQL_ACCESS_V1_15_1.RCV_BUFFER;

```

Ilustración 22 BBDD

Es en este momento, cuando puede empezar la comunicación en MySQL con la BBDD, de manera que se dispone de la función DBF_StartComT2_V1_01_1, cuyo parámetro de entrada strBBDD_StartCommand es el parámetro dónde se han de introducir los comandos correspondientes al envío de datos en lenguaje MySQL. De manera que el contenido de dicha variable se corresponde con:

INSERT INTO tfm_v01 VALUES(

```

DBFL_StartComT2_V1_01_1(IN_xActivate:=xBBDD_Almacenar,
IN_strStartString:=strBBDD_StartCommand,
INOUT_udtComBuffer:=SendBuffer);
xOUT_StartCom:=DBFL_StartComT2_V1_01_1.OUT_DataValid;
SendBuffer:=DBFL_StartComT2_V1_01_1.INOUT_udtComBuffer;

```

Ilustración 23 Comando MySQL



El resto del comando se refiere a los datos a enviar. Por ello se utiliza una función del tipo *DBFL_TIPODATOT2_V1_02_1* para cada dato enviado a la base de datos, dónde *TIPODATO* se corresponde al tipo de la variable que se envíe.

```
*FECHA*)
DBFL_DateTimeStrT2_V1_02_1(IN_xActivate:=xOUTStartCom,
    IN_iYear:=RTC_YEAR,
    IN_iMonth:=RTC_MONTH,|
    IN_iDay:=RTC_DAY,
    IN_iHour:=RTC_HOURS,
    IN_iMinute:=RTC_MINUTES,
    IN_iSecond:=RTC_SECONDS,
    IN_bLastChar:=Byte_NotLastChar,
    INOUT_udtComBuffer:=SendBuffer);
xOUTDateTime:=DBFL_DateTimeStrT2_V1_02_1.OUT_DataValid;
SendBuffer:=DBFL_DateTimeStrT2_V1_02_1.INOUT_udtComBuffer;

(*NOMBRE PULSES SO*)
DBFL_StrToComT2_V1_01_2(IN_xActivate:=xOUTDateTime,
    IN_strString:=Entrada_BBDD,
    INOUT_udtComBuffer:=SendBuffer);
xOUTEntrada:=DBFL_StrToComT2_V1_01_2.OUT_DataValid;
SendBuffer:=DBFL_StrToComT2_V1_01_2.INOUT_udtComBuffer;

(*VALOR PULSO*)
DBFL_IntToComT2_V1_01_1(IN_xActivate:=xOUTEntrada,
    IN_iValue:=Valor_Pulso_sal,
    IN_bLastChar:=Byte_NotLastChar,
    INOUT_udtComBuffer:=SendBuffer);
xOUTValPulso:=DBFL_IntToComT2_V1_01_1.OUT_DataValid;
SendBuffer:=DBFL_IntToComT2_V1_01_1.INOUT_udtComBuffer;

(*Consumo*)
DBFL_IntToComT2_V1_01_2(IN_xActivate:=xOUTValPulso,
    IN_iValue:=Consumo_sal,
    IN_bLastChar:=Byte_NotLastChar,
    INOUT_udtComBuffer:=SendBuffer);
xOUTConsumo:=DBFL_IntToComT2_V1_01_2.OUT_DataValid;
SendBuffer:=DBFL_IntToComT2_V1_01_2.INOUT_udtComBuffer;
```

Ilustración 24 BBDD



Se activa la función DBFL_StartComT2_V1_01_1 periódicamente, de esta manera se realiza un envío de parámetros a la base de datos periódico. Para realizar esta activación se hace llamada a la función Act_Periodicamente.

```
if (xBDD_TCP_Ready and xBDD_SQL_Ready) then
  Act_periodicamente_1(xInput:=true,
  time_pul:=t#1s);
  xBDD_Almacenar:=Act_periodicamente_1.xSal;
else Act_periodicamente_1.xInput:=false;;
END_IF;
```

3.1.2.2.1.2 Act_Periodicamente:

Función realizada para que dada una entrada booleana activa, su salida xSal se active periódicamente, según la variable de entrada Time_pul. Esta variable también es una entrada de la función de modo que este tiempo ha de definirse cada vez que se realice llamada a la función.

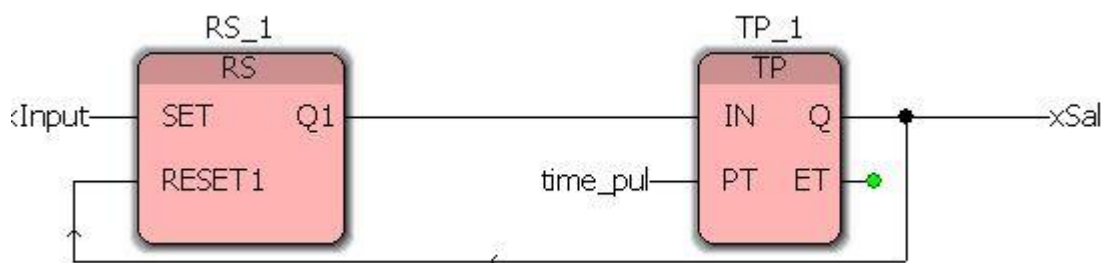


Ilustración 25 Act_Periodicamente



3.1.2.2.1.3 StrBBDD:

Para que el envío de los datos sea correcto, se necesita enviar los datos tipo string con un determinado formato. Por ejemplo, una variable tipo string cuyo contenido sea HOLA, no puede enviarse así directamente, sino que debería ser "HOLA", es decir, entre comillas y con una coma al final. Por ello, se realiza la conversión del carácter * a tipo string, mediante la variable de entrada "BUFFER", dicha entrada ha de ser en formato ASCII, de la función BUF_TO_STRING_1.

Se realiza también una conversión del carácter , del mismo modo que se hizo con la comilla, mediante la función BUF_TO_STRING_2.

Una vez convertidos estos caracteres, se concatenan con los datos tipo string que se desean enviar a la base de datos, para que tengan la topología requerida para su envío.



```
BUF_TO_STRING_1(REQ:=true(* BOOL *),
  BUF_FORMAT:=true(* BOOL *),
  BUF_OFFS:= buf_offs(*DINT *),
  BUF_CNT:= buf_cnt(* DINT *),
  BUFFER:=ascii_comilla(* ANY *),
  DST:=comilla(* ANY *));
(* BOOL *)done_buftostr:=BUF_TO_STRING_1.DONE;
(* BOOL *)xError_buftostr:=BUF_TO_STRING_1.ERROR;
(* INT *)(*:=BUF_TO_STRING_1.STATUS;
(* ANY *)ascii_comilla:=BUF_TO_STRING_1.BUFFER;
(* ANY *)comilla:=BUF_TO_STRING_1.DST;

BUF_TO_STRING_2(REQ:=true(* BOOL *),
  BUF_FORMAT:=true(* BOOL *),
  BUF_OFFS:=buf_offs(* DINT *),
  BUF_CNT:=buf_cnt(* DINT *),
  BUFFER:=ascii_coma(* ANY *),
  DST:=coma(* ANY *));
(* BOOL *)xdone_coma:=BUF_TO_STRING_2.DONE;
(* BOOL *)(*:=BUF_TO_STRING_2.ERROR;
(* INT *)(*:=BUF_TO_STRING_2.STATUS;
(* ANY *)ascii_coma:=BUF_TO_STRING_2.BUFFER;
(* ANY *)coma:=BUF_TO_STRING_2.DST;

unid_prin:=CONCAT(comilla, Unidades_sal);
unid_med:=CONCAT(unid_prin,comilla);
Unidades_BBDD:=CONCAT(unid_med,coma);

input_prin:=CONCAT(comilla,Entrada);
input_med:=CONCAT(input_prin,comilla);
Entrada_BBDD:=CONCAT(input_med,coma);
```

Ilustración 26 StrBBDD

3.1.2.2.1.4 Sel_Val:

Cómo los analizadores de redes pueden estar comunicados por Modbus TCP o RTU, y esto significa tener unos datos u otros, se realiza una selección de los datos válidos en función del protocolo de comunicación utilizado. Para ello, se realiza una llamada a la función SelectTCPRTU para cada uno de los datos medidos por el analizador de redes.



Una vez seleccionados los datos correctos, estos serán los que se envían a la base de datos.

```
-----  
SelectTCPRTU_2 (INPUT_TCP:=U12_tcp,  
INPUT_RTU:=U12_rtu(* REAL *));  
(* REAL *)U12:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_3 (INPUT_TCP:=U23_tcp,  
INPUT_RTU:=U23_rtu(* REAL *));  
(* REAL *)U23:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_4 (INPUT_TCP:=U31_tcp,  
INPUT_RTU:=U31_rtu(* REAL *));  
(* REAL *)U31:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_5 (INPUT_TCP:=V1_tcp,  
INPUT_RTU:=V1_rtu(* REAL *));  
(* REAL *)V1:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_6 (INPUT_TCP:=V2_tcp,  
INPUT_RTU:=V2_rtu(* REAL *));  
(* REAL *)V2:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_7 (INPUT_TCP:=V3_tcp,  
INPUT_RTU:=V3_rtu(* REAL *));  
(* REAL *)V3:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_8 (INPUT_TCP:=I1_tcp,  
INPUT_RTU:=I1_rtu(* REAL *));  
(* REAL *)I1:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_9 (INPUT_TCP:=I2_tcp,  
INPUT_RTU:=I2_rtu(* REAL *));  
(* REAL *)I2:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_10 (INPUT_TCP:=I3_tcp,  
INPUT_RTU:=I3_rtu(* REAL *));  
(* REAL *)I3:=SelectTCPRTU_2.salida;  
  
SelectTCPRTU_11 (INPUT_TCP:=In_tcp,  
INPUT_RTU:=In_rtu(* REAL *));  
-----
```

Ilustración 27 Sel_Val



3.1.2.2.1.5 *SelectTCPRTU*:

Con el objetivo de no consumir recursos innecesarios del controlador, se realiza una función que seleccione que datos de los analizadores de redes, se envían a la base de datos, en función de que estos estén conectados por Modbus RTU o Modbus TCP.

De manera que, las entradas de la función sean las variables tanto de TCP como de RTU, y en función de que protocolo de comunicación se use para ese esclavo, se seleccione el dato correspondiente al mismo, siendo éste el parámetro de salida de la función.

```
IF xTCP THEN
    salida:=INPUT_TCP;
ELSE IF xRTU THEN
    salida:=INPUT_RTU;
END_IF;
END_IF;

VOO1 := VOO1 ;
```

Ilustración 28 *SelectTCPRTU*



3.1.2.2.2 FTP

Conjunto de scripts que incluyen la programación necesaria para el almacenamiento de información en un fichero .csv, almacenado en un servidor FTP y en la memoria flash del propio controlador ILC 191.

3.1.2.2.2.1 FTP_Settings:

Script en lenguaje estructurado que se encarga de establecer la conexión con un servidor FTP y que almacena tanto en la memoria local del PLC cómo en dicho servidor, los ficheros .csv que contienen los datos seleccionados por el usuario en el apartado ArchiveValues.

En primer lugar, y dado que es el usuario quien enviará los parámetros de configuración para este modo de almacenamiento, se realiza una copia de las variables que modifica el usuario desde la interface web.

```
(*Variables Visualizacion_FTP*)
str_FileName:=WVis_FileName;
str_IPserver:=WVis_IPserver;
str_User_ftp:=WVis_User;
str_password_ftp:=WVis_Password;
xHabilitar_CSV:=WVis_CSV;
```

Si el usuario pulsa el botón Almacenar, se activan periódicamente dos datalogger para el almacenamiento , tanto local como en servidor FTP, de los ficheros .csv. El periodo de recogida de datos varía en función de la ubicación del archivo. Para el almacenamiento en servidor FTP se toman medidas cada 3 minutos (“temporización_FTP”). Sin embargo, para el almacenamiento local se establece un tiempo de 5 minutos (“temporización_PLC”). Se realiza esta distinción de tiempos porque la memoria flash del PLC es limitada, pudiendo con un periodo mayor de



recogida de datos, disminuir el tamaño de los mismos, y almacenar más datos.

```
(*Si el usuario pulsa almacenar, se activan dos datalogger para el almacenamiento de  
ficheros .csv en un servidor FTP y almacenamiento local en la memoria del PLC*)  
IF xAlmacenar then  
  xActivar_Datalogger:=true;  
  Act_periodicamente_3(xInput:=xReady_Datalogger_FTP,  
    time_pul:=temporizacion_FTP);  
  xReq_Datalogger_FTP:=Act_periodicamente_3.xSal;  
  Act_periodicamente_4(xInput:=xReady_Datalogger_PLC,  
    time_pul:=temporizacion_PLC);  
  xReq_Datalogger_PLC:=Act_periodicamente_4.xSal;  
end_if;
```

Aunque el envío de datos esté activo, es necesario localizar los datos que se van a enviar. Dichos datos, son los seleccionados en Archive Values. Se realiza una captura de dichos datos (“Data_captu” y “Data_captu_PLC”) antes de proceder a su envío. Esta operación se realiza tanto para los dos ficheros (servidor FTP y almacenamiento local).

```
(*DATALOGGER PARA ALMACENAMIENTO DE DATOS CADA 3 MINUTOS EN FTP*)  
DataLogger_VO_18_2(  
  xActivate:=xActivar_Datalogger,  
  xReq:=xReq_Datalogger_FTP,  
  xAddTimeToRow:=xAddTime,  
  udtDataLogger:=Data,  
  udtDl_If:=Data_captu);  
xReady_Datalogger_FTP:=DataLogger_VO_18_2.xActive;  
xError_Datalogger_FTP:=DataLogger_VO_18_2.xError;  
wDiagCode_Datalogger_FTP:=DataLogger_VO_18_2.wAddDiagCode;  
Data:=DataLogger_VO_18_2.udtDataLogger;  
Data_captu:=DataLogger_VO_18_2.udtDl_If;
```

Ilustración 29 Datalogger_FTP



```
(*DATALOGGER PARA ALMACENAMIENTO DE DATOS CADA 5 MINUTOS EN PLC*)
DataLogger_VO_18_3(
  xActivate:=xActivar_Datalogger,
  xReq:=xReq_Datalogger_PLC,
  xAddTimeToRow:=xAddTime,
  udtDataLogger:=Data,
  udtDl_If:=Data_captu_PLC);
xReady_Datalogger_PLC:=DataLogger_VO_18_3.xActive;
xError_Datalogger_PLC:=DataLogger_VO_18_3.xError;
wDiagCode_Datalogger_PLC:=DataLogger_VO_18_3.wAddDiagCode;
Data:=DataLogger_VO_18_3.udtDataLogger;
Data_captu_PLC:=DataLogger_VO_18_3.udtDl_If;
```

Ilustración 30 Datalogger_PLC

En el caso de almacenamiento de ficheros en el servidor FTP, se precisa del establecimiento de conexión con el mismo. Para ello se utiliza el siguiente bloque, al que se le pasan cómo parámetros de entrada la dirección IP del servidor, el puerto por el que realizará la conexión, los datos, usuario y contraseña necesarios para el acceso al servidor o el periodo de creación de un nuevo archivo.

```
(*Envío de datos a servidor FTP*)
DataLogFTP_VO_18_1(xActivate:=xReady_Datalogger,
  iCreateNewFileAt_n_oClock:=icreatenewfile,
  strFileName:=str_FileName,
  iSaveAfter_n_Records:=10,
  iCreateNewFileAfter_n_Days:=1,
  strFTP_Server_IP:=str_IPserver,
  strIP_PLC:='192.168.1.111',
  strFTP_User:=str_User_ftp,
  strFTP_Password:=str_password_ftp,
  bConf:=byte#16#07,
  iPort:=iPort_FTP,
  xCreateNewFile:=xnewFile, (* BOOL *)
  udtDl_If:=Data_captu,
  strActFile:=strActFile_ftp,
  strLastFile:=strLastFile_ftp);

xReady_FTP:=DataLogFTP_VO_18_1.xActive;
iusedmemory_ftp:=DataLogFTP_VO_18_1.iUsedMemory;
xnewFile_CSV:=DataLogFTP_VO_18_1.xNewFileCreated;
strLastResponse:=DataLogFTP_VO_18_1.strLastResponse;
xError_ftp:=DataLogFTP_VO_18_1.xError;
Data_captu:=DataLogFTP_VO_18_1.udtDl_If;
strActFile_ftp:=DataLogFTP_VO_18_1.strActFile;
strLastFile_ftp:=DataLogFTP_VO_18_1.strLastFile;
```

Ilustración 31 Envío Fichero a FTP



En el caso de almacenamiento de ficheros en el servidor FTP, se precisa del establecimiento de conexión con el mismo. Para ello se utiliza el siguiente bloque, al que se le pasan cómo parámetros de entrada el nombre del fichero, cada cuantos registros se guardarán los datos (“iSaveAfter_n_Records”) y el número de archivos máximos que se permiten almacenar de este modo, tal que una vez llegado a esa cantidad se procede a su borrado.

```
DataLogCSV_VO_18_1(  
    xActivate:=xReady_Datalogger_PLC,  
    iCreateNewFileAt_n_oClock:=0,  
    strFileName:=str_FileName,  
    iSaveAfter_n_Records:=50,  
    iCreateNewFileAfter_n_Days:=10,  
    (* bConf:=(* BYTE *)  
    xCreateNewFile:=xNewFileCreate_PLC, (* BOOL *)  
    iMaxFiles:=5,  
    strActFile:=strActFile_csv,  
    strLastFile:=strLastFile_csv,  
    udtDl_If:=Data_captu_PLC);  
xReady_CSV:=DataLogCSV_VO_18_1.xActive;  
(* INT *) iUsedMem_PLC:=DataLogCSV_VO_18_1.iUsedMemory;  
(* BOOL *) xnewFile_CSV_PLC:=DataLogCSV_VO_18_1.xNewFileCreated;  
xError_CSV:=DataLogCSV_VO_18_1.xError;  
strActFile_csv:=DataLogCSV_VO_18_1.strActFile;  
strLastFile_csv:=DataLogCSV_VO_18_1.strLastFile;  
Data_captu_PLC:=DataLogCSV_VO_18_1.udtDl_If;
```

Ilustración 32 Envío Fichero a PLC



3.1.2.2.3 EMAIL

3.1.2.2.3.1 *Email:*

Script en lenguaje estructurado que se encarga de realizar el envío de los archivos .csv , almacenados de manera local, por correo electrónico.

En primer lugar, y dado que es el usuario quien envía la configuración del correo electrónico, se realiza una copia de las variables que modifica el usuario desde la interface web.

```
strAsunto:=WV_Asunto;
Destinatarios.AddressList[1]:=WV_Destinatario;
strMensaje:=WV_Mensaje;
strServer_Address:=WV_ServerAddr;
strIP_Server_Address:=WV_IP_ServerAddress;
strRemitente:=WV_Remitente;
strUser:=WV_User;
strPassword:=WV_Password;

Destinatarios.AddressCnt:=1;
```

Ilustración 33 Email:VariablesWVvisit

En segundo lugar, si el usuario activa la opción de envío de datos por correo electrónico (variable “xOpcion_Mail” activa), cada vez que un nuevo archivo sea creado, se activa el envío. Una vez finalizado el envío se espera que otro archivo sea creado para realizar un nuevo envío.

```
if xOpcion_Mail THEN

    if xnewFile_CSV then
        xActivar_Mail:=TRUE;
    END_IF;
    IF xDone_Mail then
        xActivar_Mail:=False;
    end_if;
```



La activación del proceso de envío se hace mediante la variable xActivar_Mail del bloque SMTP_Client_V1_16_1. Esta función, es la encargada de realizar el envío, y por ello, todas las entradas de dicho bloque se corresponden con los parámetros de configuración del mail. La activación de archivos adjunto, la cantidad de archivos a adjuntar, el puerto utilizado para el envío de correo electrónico, dirección de destinatario, mensaje, nombre de archivo adjunto, dirección IP y DNS del servidor al que se envía el correo, dirección del remitente, y finalmente usuario y contraseña del mismo, son las entradas correspondientes a esta función. Cada vez que el envío se haya realizado satisfactoriamente, se activa la salida xDone_Mail de la función.

```

fileNames[1]:=strLastFile_Itf;
SMTP_Client_V1_16_1(xSendMail:=xActivar_Mail,
  strSubject:=strAsunto,
  xAuthentication:=TRUE,
  xFileAttachment:=TRUE,
  iFileAttachmentCnt:=1,
  xHostname:=TRUE,
  xTextAsBuffer:=FALSE,|
  (*iTextBufferSize:=, (* INT *)
  (*strTimeZone:=, (* SMTP_STR_5*)
  tTimeOut:=t#30s,
  strPort:='25',
  udtMailTo:=Destinatarios(* SMTP_UDT_EmailAddress_V1 *),
  strText:=strMensaje, (* SMTP_STR_200 *)
  bFirstByteOfTextBuffer:=FirstByte(* BYTE *),
  arrFileNames:=FileNames, (* SMTP_ARR_STR_1_10 *)
  strSMTP_ServerAddress:=strServer_Address(* STRING *),
  strDNS_IP_Address:=strIP_Server_Address(* STRING *),
  strMailFrom:=strRemitente(* STRING *),
  strUser:=strUser(* STRING *),
  strPassword:=strPassword(* STRING *),
  strDataRecStr:=strDataRec(* SMTP_STR_200 *));

(* BOOL *)xDone_Mail:=SMTP_Client_V1_16_1.xDONE;
(* BOOL *)xError_mail:=SMTP_Client_V1_16_1.xERROR;
(* WORD *)xWdiagCode:=SMTP_Client_V1_16_1.wDiagCode;
(* WORD *)wAddDIAGCODE:=SMTP_Client_V1_16_1.wAddDiagCode;
(* SMTP_UDT_EmailAddress_V1 *)Destinatarios:=SMTP_Client_V1_16_1.udtMailTo;
(* SMTP_STR_200 *)strMensaje:=SMTP_Client_V1_16_1.strText;
(* BYTE *)FirstByte:=SMTP_Client_V1_16_1.bFirstByteOfTextBuffer;
(* SMTP_ARR_STR_1_10 *)FileNames:=SMTP_Client_V1_16_1.arrFileNames;
(* STRING *)strServer_Address:=SMTP_Client_V1_16_1.strSMTP_ServerAddress;
(* STRING *)strIP_Server_Address:=SMTP_Client_V1_16_1.strDNS_IP_Address;
(* STRING *)strRemitente:=SMTP_Client_V1_16_1.strMailFrom;
(* STRING *)strUser:=SMTP_Client_V1_16_1.strUser;
(* STRING *)strPassword:=SMTP_Client_V1_16_1.strPassword;
(* SMTP_STR_200 *)strDataRec:=SMTP_Client_V1_16_1.strDataRecStr;

```



3.1.2.2.4 MODBUS

3.1.2.2.4.1 Settings_Modbus:

Script en lenguaje estructurado que se encarga de guardar la configuración del protocolo Modbus TCP y Modbus RTU introducida por el usuario.

En primer lugar, se realiza una copia de los parámetros correspondientes a Modbus TCP, en este caso correspondiente a la dirección IP del esclavo correspondiente:

```
(*Modbus TCP*)  
  
IP_TCP:=Cf_IP_TCP;
```

Ilustración 35 Settings TCP

Seguidamente, se realiza una copia de los parámetros de configuración relativos al protocolo Modbus RTU, velocidad de transmisión, y número de esclavo.

```
(*Modbus RTU*)  
  
diBaudrate_RTU:=Cf_diBaudrate;  
Slave_Addr_ValOnline_RTU:=Cf_uiSlave_Addr_RTU;  
Slave_Addr_ValAdvan_RTU:=Cf_uiSlave_Addr_RTU;
```

Ilustración 36 Settings RTU

Finalmente, se cargan y se guardan los parámetros de configuración. Para realizar cualquier cambio en dicha modificación, se establece un tiempo de espera de 5 segundos.

```
IF xGuardar_Config THEN  
  xMsj_Guardado:=true;  
  TOF_3 (IN:=xMsj_Guardado,PT:=time#5s);  
  IF TOF_3.Q THEN  
    xMsj_Guardado:=false;  
    xGuardar_Config:=false;  
  end_if;  
end_if;
```



3.1.3 DISPOSITIVOS

Se trata de un grupo de POU's programados para la lectura de los valores medidos por cada uno de los dispositivos. Se realiza uno o varios scripts para cada dispositivo, en función de los requerimientos de configuración que cada uno de ellos necesita. Para el caso del analizador de redes (Empro) y contador energético (Pulses S0) es necesario la creación de un grupo de POU's para cada uno. Sin embargo, para el módulo de medición de potencia y entradas analógicas con un script, respectivamente, es suficiente. La estructura del grupo de POU's llamado "DISPOSITIVOS" es la siguiente:

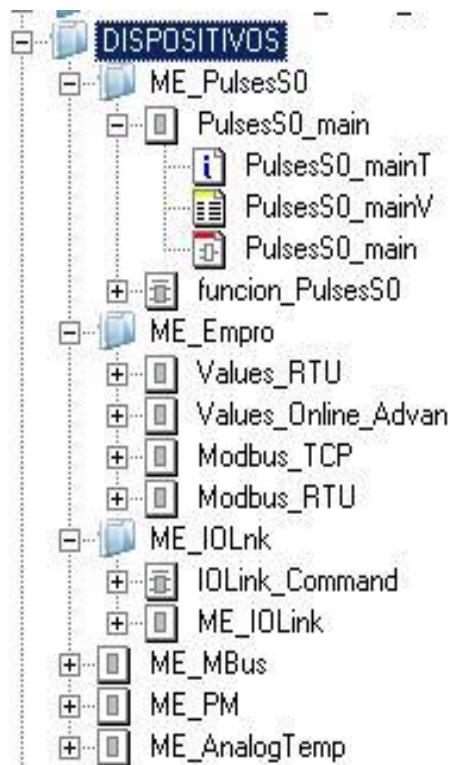


Ilustración 37 Dispositivos



3.1.3.1 ME_PulsesS0

Conjunto de POUS en la que se realiza la programación necesaria para la lectura de datos de los contadores energéticos. Se prepara dicha programación para la conexión de 7 contadores diferentes. Dicha programación se estructura en dos scripts: "PulsesS0_main", programa principal y que realiza una llamada a la función: "función_PulsesS0". Véase explicación de ambos scripts a continuación.

3.1.3.1.1 PulsesS0_main

Script programado en bloque de funciones que realiza llamada a una función cuyas entradas se corresponden con las entradas digitales del PLC a las que están conectados los contadores energéticos, y cuyas salidas, se corresponden con los valores medidos para cada uno de dichos contadores. De manera que:

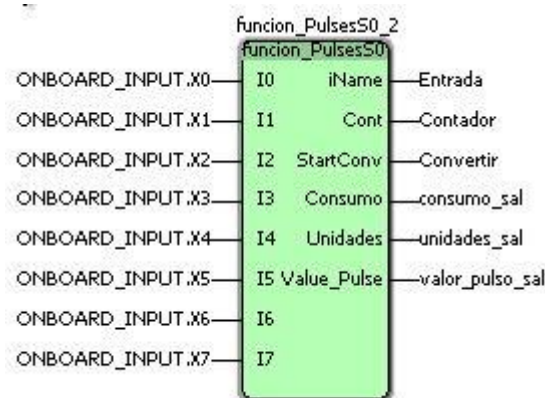


Ilustración 38 PulsesS0_main

Dónde la salida Entrada se corresponde con el valor del contador que ha incrementado su contador, "Contador" es el número de pulsos registrados para dicho contador, "consumo_sal" el valor actual de consumo de dicho contador, "valor_pulso_sal" el valor cuantitativo de cada pulso del contador y finalmente, "unidades" se corresponde con la unidad de cada valor del pulso del contador activo. La salida "convertir" se pone a true cada vez que un contador envía un pulso.



En este caso, sólo se realiza una llamada a la función por que se disponen de 8 contadores, en caso de tener una instalación mayor número de contadores, con realizar una llamada a la función “función_Pulses0” por cada 8 contadores sería suficiente.

3.1.3.1.2 *Función_Pulses0*

Función implementada en Texto Estructurado para la lectura de los valores de los contadores conectados al PLC.

Cada vez que el PLC recibe un pulso de alguno de los contadores, se incrementa un contador correspondiente a ese mismo contador energético, y se almacena qué contador ha enviado el pulso, en función de la entrada digital que se active. Cada vez que una de estas entradas digitales se activa, se activa la salida StartConv. Si todas las entradas están desactivadas StartConv será igual a FALSE. En caso de que StartConv sea TRUE, se activará el almacenamiento de datos.



3.1.3.2 ME_Empro

Grupo de POU programado para la lectura de datos procedentes de analizadores de redes Empro 600 de Phoenix Contact. Se realizan 4 scripts para dicha lectura, y que difieren entre ellos en función del protocolo de comunicación implementado, bien sea Modbus TCP o Modbus RTU. Para ambos casos se requiere incluir en el proyecto la librería “Modbus_V1_07”, utilizada para la configuración del protocolo de comunicación Modbus. A continuación se explica la programación realizada para cada una de estas opciones.

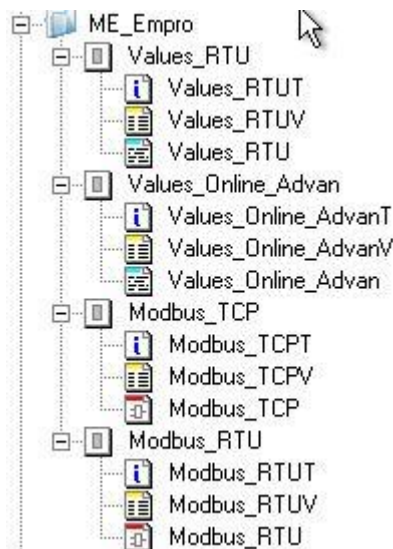


Ilustración 39 ME_Empro



3.1.3.2.1 Modbus_TCP

Script programado en lenguaje de bloque de funciones. En primer lugar, se configura el cliente Modbus para la aplicación, mediante el bloque “MB_TCP_Client_V1_22_2”, que tiene cómo entrada la dirección IP del analizador de redes, mediante la variable de “IP_TCP”. Es necesario configurar el puerto correspondiente al protocolo de comunicación Modbus TCP, mediante la entrada Port del bloque, correspondiente al valor 502. La variable de entrada/salida “Client_Com_Struct” se corresponde con la estructura de datos donde se almacenarán los datos, tanto de configuración, cómo los de lectura del analizador.

Se programan dos bloques llamados “MT_ReadHoldingReg_V1_20_1” y “MT_ReadHoldingReg_V1_20_2” para la lectura de los datos deseados del analizador de redes. La necesidad del uso de dos bloques de este tipo viene dada por que los datos deseados a leer del analizador de redes se encuentran en diferentes direcciones del mapa de memoria del analizador de redes. Los datos a leer son valores energéticos básicos (leídos por el bloque “MT_ReadHoldingReg_V1_20_1”) y valores avanzados, que se corresponden con la media y máximo de los datos básicos (leídos por el bloque “MT_ReadHoldingReg_V1_20_2”).

La primera dirección de memoria en la que se encuentran los valores básicos se corresponde con 50514, variable de entrada “First_Addr_TCP” y la cantidad de direcciones a leer son 52, variable de entrada “Quantity_TCP” (“MT_ReadHoldingReg_V1_20_1”). Sin embargo, los valores avanzados se encuentran en la dirección de memoria 51024, dato introducido por la variable de entrada “First_Addr_TCP_ValAdvan” y la cantidad de direcciones a leer son 46, que se corresponde con la variable “Quantity_TCP_ValAdvan” del bloque “MT_ReadHoldingReg_V1_20_2”.



Los valores básicos leídos se almacenan en un array de words llamado “Values_Online”, parámetro de salida del bloque “MT_ReadHoldingReg_V1_20_1”. Por otro lado, los valores avanzados (medios y máximos de los valores básicos) se almacenan en un array de words llamado “Values_Advan”, variable de salida del bloque “MT_ReadHoldingReg_V1_20_2”.

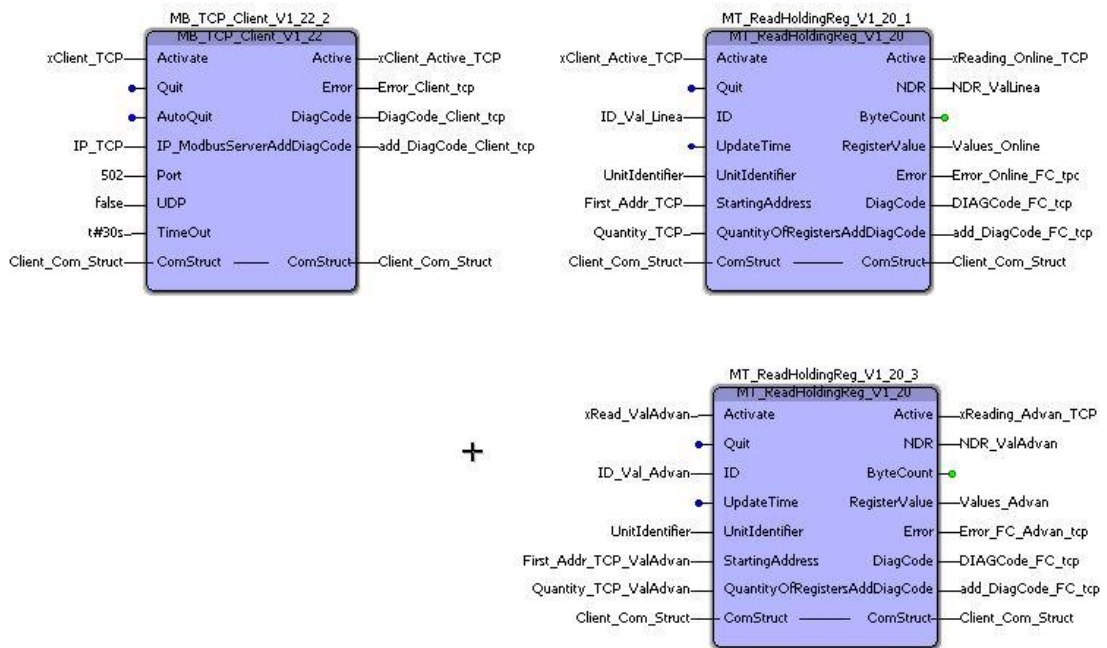


Ilustración 40 Modbus TCP

3.1.3.2.2 Values_Online_Advan

Script programado en lenguaje texto estructurado encargado de la gestión y tratamiento de los datos tomados en el script Modbus_TCP. Se realiza el tratamiento necesario para que las variables “Values_Online” y “Values_Advan” sean legibles, de modo que cada Word del array se corresponda con un dato energético.

Se realiza una conversión de tipo Word a real para cada dato, y se almacena en una variable de tipo real cada dato, que finalmente, se mostrará en la visualización web.



Finalmente, se vuelve a almacenar en un array de reales todos estos datos leídos. Esto se realiza para poder tener en un array todos los datos correspondientes a los tomados mediante el protocolo Modbus TCP. (Se realizará otro para Modbus RTU en el script correspondiente a dicho protocolo).

```
xRead_ValAdvan:=TRUE;
U12_med_tcp:=word_to_real(Values_Advan[2])/100.0;
U23_med_tcp:=word_to_real(Values_Advan[4])/100.0;
U31_med_tcp:=word_to_real(Values_Advan[6])/100.0;

V1_med_tcp:=word_to_real(Values_Advan[8])/100.0;
V2_med_tcp:=word_to_real(Values_Advan[10])/100.0;
V3_med_tcp:=word_to_real(Values_Advan[12])/100.0;

frec_med_tcp:=word_to_real(Values_Advan[14])/100.0;

I1_med_tcp:=word_to_real(Values_Advan[16])/1000.0;
I2_med_tcp:=word_to_real(Values_Advan[18])/1000.0;
I3_med_tcp:=word_to_real(Values_Advan[20])/1000.0;
In_med_tcp:=word_to_real(Values_Advan[22])/1000.0;

Pt_med_tcp:=word_to_real(Values_Advan[24])/100.0;
Qt_med_tcp:=word_to_real(Values_Advan[28])/100.0;
St_online_med_tcp:=word_to_real(Values_Advan[32])/100.0;

/val.Máximos*/
V1_max_tcp:=word_to_real(Values_Advan[40])/100.0;
V2_max_tcp:=word_to_real(Values_Advan[42])/100.0;
V3_max_tcp:=word_to_real(Values_Advan[44])/100.0;

frec_max_tcp:=word_to_real(Values_Advan[46])/100.0;

I1_max_tcp:=word_to_real(Values_Advan[48])/1000.0;
I2_max_tcp:=word_to_real(Values_Advan[50])/1000.0;
I3_max_tcp:=word_to_real(Values_Advan[52])/1000.0;
In_max_tcp:=word_to_real(Values_Advan[54])/1000.0;

Pt_max_tcp:=word_to_real(Values_Advan[56])/100.0;
Qt_max_tcp:=word_to_real(Values_Advan[60])/100.0;
St_online_max_tcp:=word_to_real(Values_Advan[64])/100.0;
>_IF;
```

Ilustración 41 Array TCP



```
U12_tcp:=word_to_real(Values_Online[2])/100.0;
U23_tcp:=word_to_real(Values_Online[4])/100.0;
U31_tcp:=word_to_real(Values_Online[6])/100.0;

V1_tcp:=word_to_real(Values_Online[8])/100.0;
V2_tcp:=word_to_real(Values_Online[10])/100.0;
V3_tcp:=word_to_real(Values_Online[12])/100.0;

frec_tcp:=word_to_real(Values_Online[14])/100.0;

I1_tcp:=word_to_real(Values_Online[16])/1000.0;
I2_tcp:=word_to_real(Values_Online[18])/1000.0;
I3_tcp:=word_to_real(Values_Online[20])/1000.0;
In_tcp:=word_to_real(Values_Online[22])/1000.0;

Pt_tcp:=word_to_real(Values_Online[24])/100.0;
Qt_tcp:=word_to_real(Values_Online[26])/100.0;
St_online_tcp:=word_to_real(Values_Online[28])/100.0;
fdp_tcp:=word_to_real(Values_Online[30])/1000.0;

P1_tcp:=word_to_real(Values_Online[32])/100.0;
P2_tcp:=word_to_real(Values_Online[34])/100.0;
P3_tcp:=word_to_real(Values_Online[36])/100.0;

Q1_tcp:=word_to_real(Values_Online[38])/100.0;
Q2_tcp:=word_to_real(Values_Online[40])/100.0;
Q3_tcp:=word_to_real(Values_Online[42])/100.0;

S1_tcp:=word_to_real(Values_Online[44])/100.0;
S2_tcp:=word_to_real(Values_Online[46])/100.0;
S3_tcp:=word_to_real(Values_Online[48])/100.0;

fdp1_tcp:=word_to_real(Values_Online[50])/1000.0;
fdp2_tcp:=word_to_real(Values_Online[52])/1000.0;
fdp3_tcp:=word_to_real(Values_Online[54])/1000.0;
```



3.1.3.2.3 *Modbus_RTU*

Script programado en lenguaje de bloque de funciones. En primer lugar, se configura el maestro Modbus RTU para la aplicación, mediante el bloque “MB_RTU485_Master_V1_01_1”, que tiene cómo entrada la variable “diBaudrate_RTU” correspondiente a la velocidad de transmisión del esclavo, en este caso 19200 baudios. La topología de los datos usada en el protocolo correspondiente al número de bits de datos, bit de paridad y número de bits de parada se configura mediante el parámetro de entrada “bDataWidth”. Para este dispositivo se utilizan 8 bits de datos, paridad par y dos bits de parada.

Una vez configurado el maestro Modbus y realizada la conexión con los esclavos del sistema, se procede a la solicitud de datos a dichos esclavos. Para ello se programan dos bloques llamados “MB_RTU_FCC03_v1_02_1” y “MB_RTU_FCC03_v1_02_2”, para los valores básicos y valores avanzados, respectivamente. El código de función necesario para la lectura de varios registros se corresponde con el código 3, de ahí la elección y nombre de los bloques así.

Ambos bloques operan del mismo modo, ambos necesitan cómo parámetro de entrada el número de esclavo del dispositivo al que van a realizar una solicitud, “Slave_Addr_ValOnline_RTU” (para el bloque destinado a valores básicos) y “Slave_Addr_ValAdvan_RTU”(para el bloque destinado a valores avanzados. Ambas variables de entrada toman el valor 1.

Al igual que en Modbus TCP, se necesita saber la primera dirección del mapa de memoria del esclavo que van a leer, “Start_Addr_Valonline_RTU” (para valores básicos con valor 50514) y “Start_Addr_ValAdvan_RTU” (para valores avanzados con valor 51024). La cantidad de direcciones de memoria a leer viene dada por las entradas “quantity_valonline_RTU” y “quantity_valadvan_RTU” (toma valor 52 para valores básicos y 46 para valores avanzados).



En ambos bloques, se habilita la solicitud de datos mediante polling, mediante la activación de la entrada booleana (“xPoll”), el periodo de encuesta o polling viene dado por la entrada “tpollInterval”.

Una vez establecida la conexión con el esclavo pertinente, se activa la salida del bloque “xReady”, y cada vez que se recibe un nuevo dato, este se almacena en un array llamado “Received_Data_RTU” (para los valores básicos) y “Received_ValAdvan_RTU” para los valores avanzados (medias y máximos).

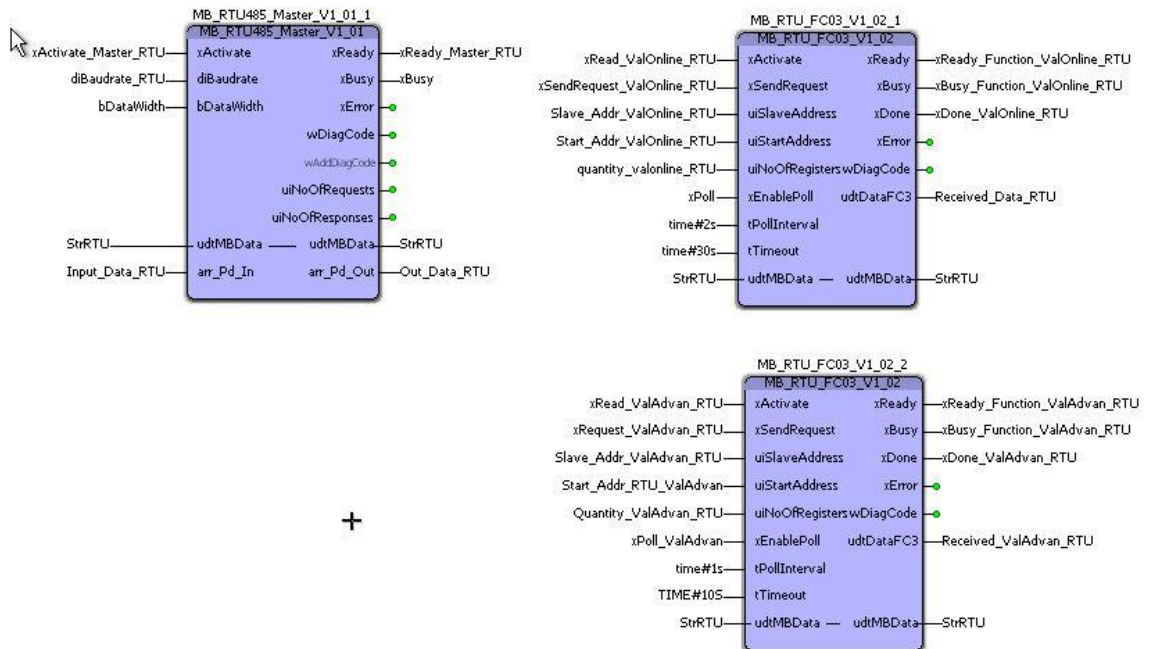


Ilustración 42 Modbus RTU



3.1.3.2.4 *Values_RTU*

Script programado en lenguaje texto estructurado encargado de la gestión y tratamiento de los datos tomados en el script Modbus_RTU. Se realiza el tratamiento necesario para que las variables “Received_Data_RTU” y “Received_ValuesAdvan_RTU” sean legibles, de modo que cada Word del array se corresponda con un dato energético.

Se realiza una conversión de tipo Word a real para cada dato, y se almacena en una variable de tipo real cada dato, que finalmente, se mostrará en la visualización web.

Finalmente, se vuelve a almacenar en un array de reales todos estos datos leídos. Esto se realiza para poder tener en un array todos los datos correspondientes a

los tomados mediante el protocolo Modbus RTU.

```
U12_med_rtu:=word_to_real(Received_Val&Advan_RTU[2])/100.0;  
U23_med_rtu:=word_to_real(Received_Val&Advan_RTU[4])/100.0;  
U31_med_rtu:=word_to_real(Received_Val&Advan_RTU[6])/100.0;  
  
V1_med_rtu:=word_to_real(Received_Val&Advan_RTU[8])/100.0;  
V2_med_rtu:=word_to_real(Received_Val&Advan_RTU[10])/100.0;  
V3_med_rtu:=word_to_real(Received_Val&Advan_RTU[12])/100.0;  
  
frec_med_rtu:=word_to_real(Received_Val&Advan_RTU[14])/100.0;  
  
I1_med_rtu:=word_to_real(Received_Val&Advan_RTU[16])/1000.0;  
I2_med_rtu:=word_to_real(Received_Val&Advan_RTU[18])/1000.0;  
I3_med_rtu:=word_to_real(Received_Val&Advan_RTU[20])/1000.0;  
In_med_rtu:=word_to_real(Received_Val&Advan_RTU[22])/1000.0;  
  
Pt_med_rtu:=word_to_real(Received_Val&Advan_RTU[24])/100.0;  
Qt_med_rtu:=word_to_real(Received_Val&Advan_RTU[28])/100.0;  
St_online_med_rtu:=word_to_real(Received_Val&Advan_RTU[32])/100.0;  
  
(*Val. Máximos*)  
V1_max_rtu:=word_to_real(Received_Val&Advan_RTU[40])/100.0;  
V2_max_rtu:=word_to_real(Received_Val&Advan_RTU[42])/100.0;  
V3_max_rtu:=word_to_real(Received_Val&Advan_RTU[44])/100.0;  
  
frec_max_rtu:=word_to_real(Received_Val&Advan_RTU[46])/100.0;  
  
I1_max_rtu:=word_to_real(Received_Val&Advan_RTU[48])/1000.0;  
I2_max_rtu:=word_to_real(Received_Val&Advan_RTU[50])/1000.0;  
I3_max_rtu:=word_to_real(Received_Val&Advan_RTU[52])/1000.0;  
In_max_rtu:=word_to_real(Received_Val&Advan_RTU[54])/1000.0;  
  
Pt_max_rtu:=word_to_real(Received_Val&Advan_RTU[56])/100.0;  
Qt_max_rtu:=word_to_real(Received_Val&Advan_RTU[60])/100.0;  
St_online_max_rtu:=word_to_real(Received_Val&Advan_RTU[62])/100.0;
```

Ilustración 43 Array RTU



```
U12_rtu:=word_to_real(Received_Data_RTU[2])/100.0;
U23_rtu:=word_to_real(Received_Data_RTU[4])/100.0;
U31_rtu:=word_to_real(Received_Data_RTU[6])/100.0;

V1_rtu:=word_to_real(Received_Data_RTU[8])/100.0;
V2_rtu:=word_to_real(Received_Data_RTU[10])/100.0;
V3_rtu:=word_to_real(Received_Data_RTU[12])/100.0;

frec_rtu:=word_to_real(Received_Data_RTU[14])/100.0;

I1_rtu:=word_to_real(Received_Data_RTU[16])/1000.0;
I2_rtu:=word_to_real(Received_Data_RTU[18])/1000.0;
I3_rtu:=word_to_real(Received_Data_RTU[20])/1000.0;
In_rtu:=word_to_real(Received_Data_RTU[22])/1000.0;

Pt_rtu:=word_to_real(Received_Data_RTU[24])/100.0;
Qt_rtu:=word_to_real(Received_Data_RTU[26])/100.0;
St_online_rtu:=word_to_real(Received_Data_RTU[28])/100.0;
fdp_rtu:=word_to_real(Received_Data_RTU[30])/1000.0;

P1_rtu:=word_to_real(Received_Data_RTU[32])/100.0;
P2_rtu:=word_to_real(Received_Data_RTU[34])/100.0;
P3_rtu:=word_to_real(Received_Data_RTU[36])/100.0;

Q1_rtu:=word_to_real(Received_Data_RTU[38])/100.0;
Q2_rtu:=word_to_real(Received_Data_RTU[40])/100.0;
Q3_rtu:=word_to_real(Received_Data_RTU[42])/100.0;

S1_rtu:=word_to_real(Received_Data_RTU[44])/100.0;
S2_rtu:=word_to_real(Received_Data_RTU[46])/100.0;
S3_rtu:=word_to_real(Received_Data_RTU[48])/100.0;

fdp1_rtu:=word_to_real(Received_Data_RTU[50])/1000.0;
fdp2_rtu:=word_to_real(Received_Data_RTU[52])/1000.0;
fdp3_rtu:=word_to_real(Received_Data_RTU[54])/1000.0;
```

3.1.3.3 ME_PM

Se trata de la programación requerida para la lectura de datos provenientes del módulo de lectura de potencia añadido al PLC. Se precisa incluir la librería “PowerMeasurement_V1_01” en el proyecto. Así como la configuración mediante PCP del módulo. En este caso se asignará el valor 3 al CR del dispositivo para que la comunicación sea posible.

Se utilizan los siguientes bloques funcionales en el siguiente orden para hacer posible la lectura de parámetros energéticos de dicho módulo. Se utilizan también dos estructuras de datos, en función de los datos que almacenen. “uParam” es la estructura que almacena todos aquellos datos referentes con la configuración del módulo. Sin embargo, “uData” es la estructura que almacena todos aquellos datos leídos por el módulo.



El bloque PM_Configuration_V1_00_1 es necesario para la configuración de las comunicaciones con él módulo. Es necesario configurar los parámetros de entrada “iSecondaryLineCurrent” y “bConnectionType” en función de la carga conectada al módulo. Se ha configurado para una carga monofásica. Para otro tipo de configuración véase hoja de características de la librería. Dichas instrucciones se encuentran anexas al presente documento.

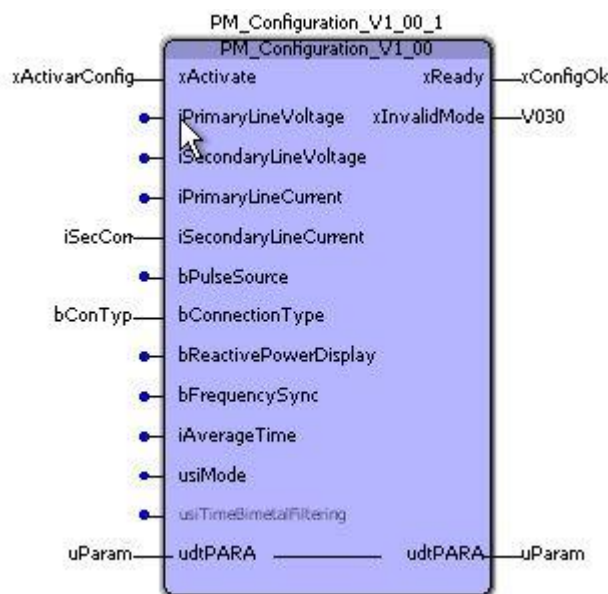
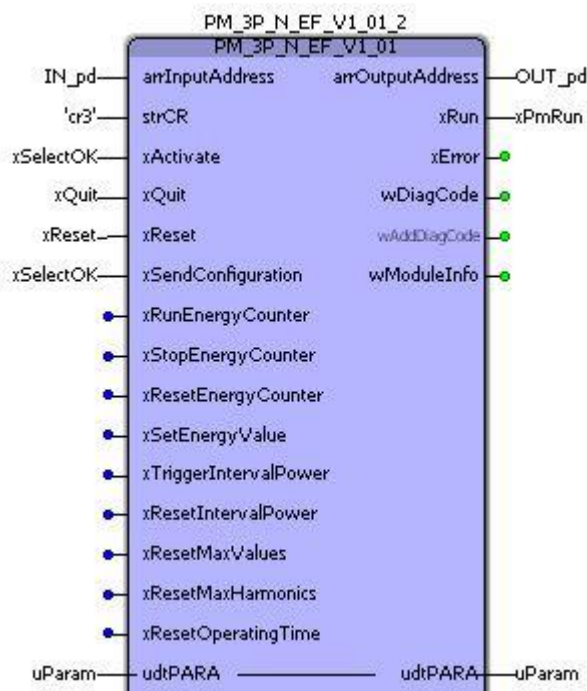


Ilustración 44 ME_PM

Seguidamente, se utiliza el bloque funcional “PM_Select_PD_V1_00_1” utilizado para la selección de datos que se desean leer BasicValue y/o ExtendedValue, es decir, valores energéticos básicos (tensión, corriente, potencia) y valores energéticos extendido (tensión de fase, corriente filtrada, frecuencia, dirección de la energía, etc.)



El bloque **PM_3P_N_EF_V1-01_2** realiza la propia comunicación del módulo y le envía al mismo los parámetros de configuración almacenados. Una vez la comunicación está establecida y la adquisición de datos preparada para realizarse, se activa la salida “**xPMRun**”.



Los bloques “**PM_BasicValues_V1_00_2**” y “**PM_ExtendedValue_V1_00_1**” una vez la comunicación con el módulo ha sido realizada tienen como salida los datos de lectura requeridos, tal que:



Basic Values:

- Tensión de Línea
- Corriente de Línea
- Potencia de Linea consumida y total

Extended Values:

- Tensión de fase
- Frecuencia

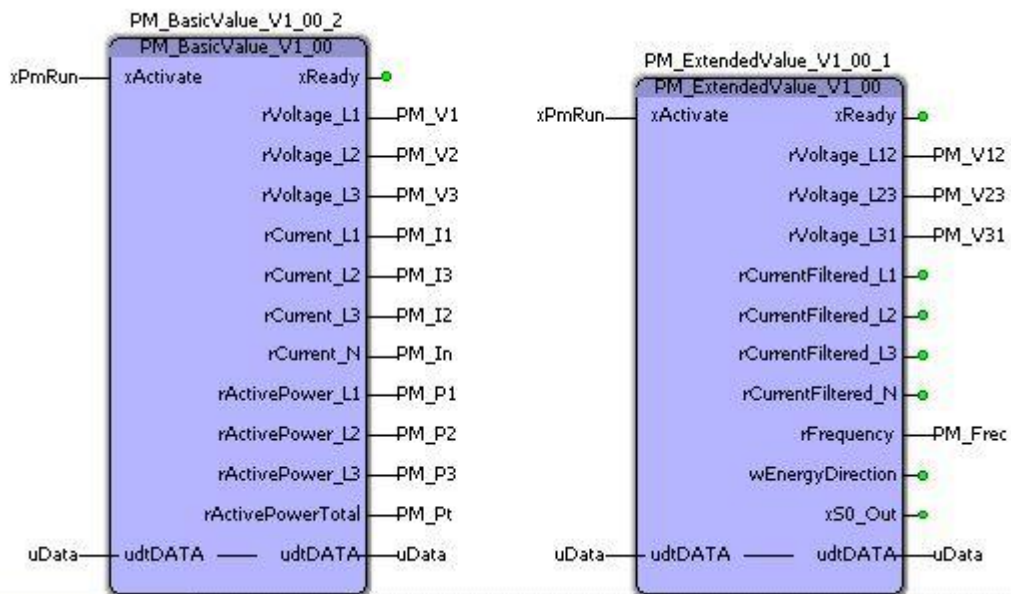


Ilustración 45 PM_Values



3.1.3.4 ME_AnalogTemp

Script desarrollado en lenguaje de texto estructurado para la lectura de entradas analógicas, en este caso, referentes a los sensores de temperatura conectados al canal 1 y 2 de su módulo correspondiente de controlador..

Se realiza un tratamiento de los datos, tal y como son recibidos en crudo, ya que se reciben en una Word, y se convierten a reales. Además se multiplica por 0.1 para ver dichos datos en °C.

Para la lectura de datos es necesario la configuración de los datos de proceso del mismo, de manera que, las variables Input01 y Input02, se correspondan cada una de ellas con un canal de lectura del propio módulo conectado al PLC. *(Ir al apartado Variables de proceso para ver la asignación y configuración apropiada)*

```
Temperatura01:=WORD_TO_REAL(Input01)*0.1;  
Temperatura02:=WORD_TO_REAL(Input02)*0.1;  
|
```

Ilustración 46 ME_AnalogTemp



3.1.3.5 ME_IOLnk

Grupo de POU's programado para la lectura de datos procedentes de dispositivos IOLink de Phoenix Contact. Se realizan 2 scripts para dicha, tal y cómo se muestra en la siguiente Ilustración, dónde ME_IOLink se corresponde con el programa principal y IOLink_Command una función que envía una estructura de datos en función de la operación que se vaya a realizar si lectura o escritura.

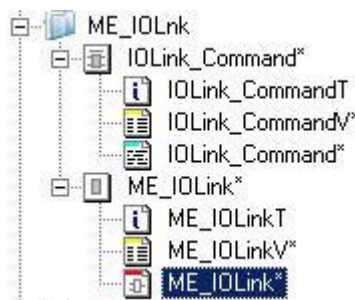


Ilustración 47 ME_IOLink_Estructura

3.1.3.5.1 ME_IOLink

Se necesita incluir en el proyecto la librería Function_Modules_V1_13 para la comunicación con dispositivos IOLink. Se realiza un programa en bloque funcionales para realizar la lectura de los datos.

En primer lugar se necesita configurar el maestro IOLink , mediante el bloque IO_Link_Master_IB_V1_01, que tiene como parámetros de entrada una estructura de datos, dos señales booleanas (“xRead” y x”Write”) que se activan en función de la operación a realizar (lectura o escritura), y de nuevo una estructura de datos que se utiliza para la comunicación de los ajustes de comunicación entre el propio maestro y su puerto de entrada al que se conecten dispositivos (“arrSysModul”). Tanto la entrada arrAI96_IL como la salida arrAO96_IL se corresponden con las variables de proceso del terminal.



La comunicación entre este tipo de terminal y el resto de terminales del ILC se realiza por PCP, de manera que es necesario indicarle su referencia de comunicación. Para ello, se utiliza el bloque IO_Link_Config_V1, que envía en un array de datos al master dicho parámetro de configuración de la comunicación.

Por otro lado, es necesaria la configuración del puerto de lectura o escritura al que se conecta el dispositivo. Para ello se utiliza el bloque IO_Link_Port_V1, que necesita cómo parámetros de entrada, un Word en que almacenará los datos de salida del bloque, un array de datos y de nuevo una señal booleana que indique si desea realizar una operación de escritura o lectura. Una vez la salida booleana xIOL_Ready_Port1 se active, empezará la operación de lectura, y los datos se almacenarán en el array de words arrIOL_Port1_ProcessValue.

Para la lectura de valores del dispositivo, se ha de realizar un tratamiento del array de datos para que los datos finalmente mostrados al usuario sean legibles. Por ello se realiza la conversión de los datos de Word a real. Es necesario tener en cuenta el factor de escala del dispositivo para los datos de salida, sean acordes con los valores reales y no, la medida directa realizada por el terminal, que no deja de ser 0 y 1. El factor de escala del dispositivo es de 27648. Se realiza la conversión pertinente, y finalmente se obtiene cómo salida la variable Salida_IOLink.

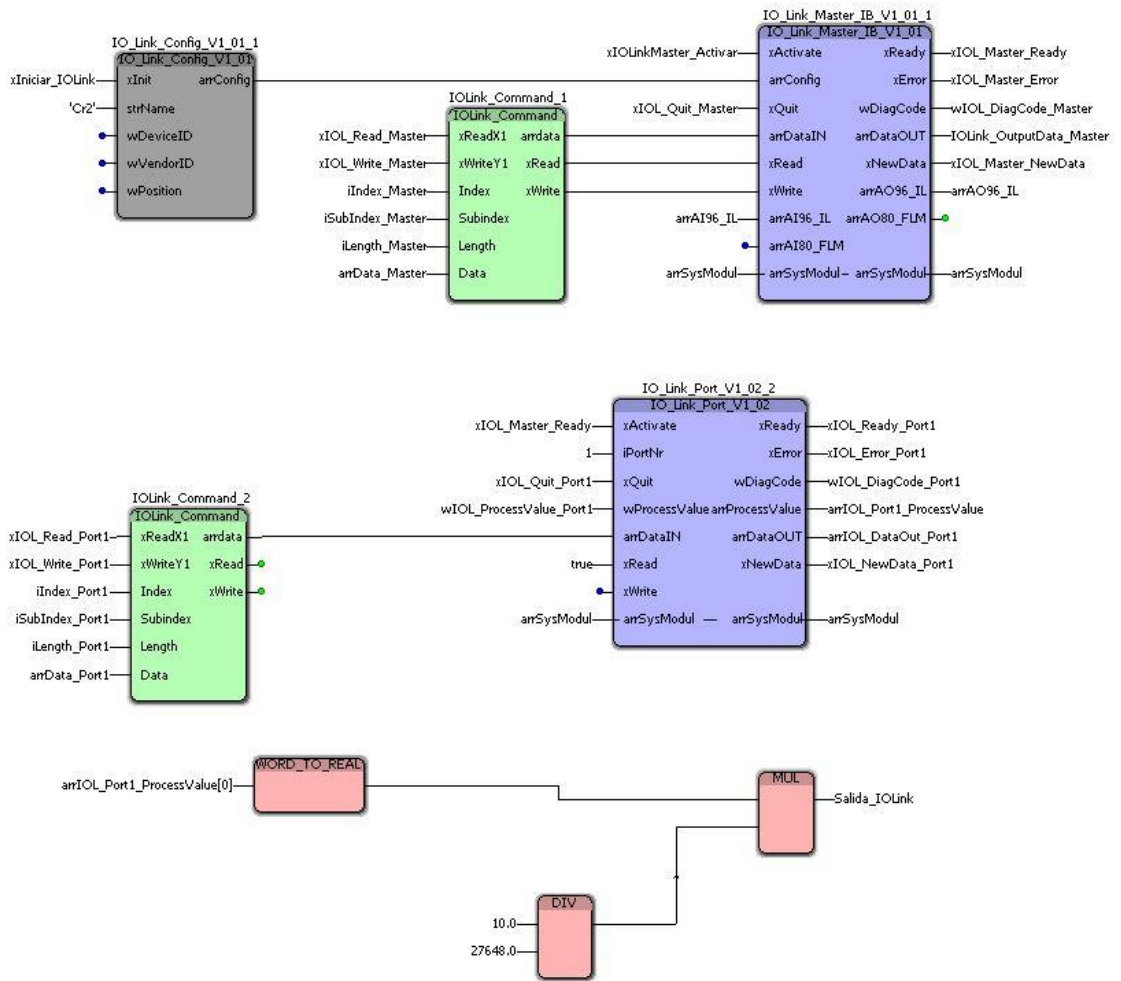


Ilustración 48 ME_IOLink



3.1.3.5.2 IOLink_Command

Función programada en lenguaje estructurado para la creación de una estructura de datos en función de la operación a realizar por el maestro IOLink (lectura o escritura). En caso de que la operación a realizar sea de lectura se activará su entrada xReadX1, y se realiza un array de datos tal y cómo exige la hoja de características del terminal. Para el caso de lectura, se activará la entrada de la función xWriteY1.

```
(***** Read *****)
if xReadX1 then
  arrdata.wIndex      := INT_TO_WORD (Index);
  arrdata.wSubindex  := INT_TO_WORD (Subindex);
  arrdata.xread       := true;
  arrdata.xWrite      := false;
end_if;

R_TRIG_1 (CLK := xReadX1);
xRead := R_TRIG_1.Q;

(***** Write *****)
if xWriteY1 then
  arrdata.wIndex      := INT_TO_WORD (Index);
  arrdata.wSubindex  := INT_TO_WORD (Subindex);
  arrdata.xWrite      := true;
  arrdata.xread       := false;
  arrdata.iLength     := Length;
  arrdata.arrData     := Data;
end_if;

R_TRIG_2 (CLK := xWriteY1);
xWrite := R_TRIG_2.Q;
```

Ilustración 49 IOLink_Command



3.1.3.6 ME_Mbus

Se necesita incluir en el proyecto la librería EDCL_V02 para la comunicación con dispositivos Mbus. Se realiza un programa en bloque funcionales para realizar la lectura de los datos.

En primer lugar, se necesita realizar la configuración del servidor de Mbus y la asignación de los datos de proceso. Para ello, es necesario el uso del bloque EDCL_MBusSvr_V2_00_3, que tiene como parámetros las variables de proceso y una estructura de datos en la que se almacena la configuración del mismo.

A continuación, se realiza un scan en búsqueda de los dispositivos Mbus seleccionados al controlador. Se detectan automáticamente cuántos dispositivos están conectados, además de su configuración de la velocidad de transmisión. Una vez se encuentran dispositivos, se procede a la lectura de datos del dispositivo mediante el bloque EDCL_Generic_MBus_V1_02, cuya activación se realiza periódicamente, mediante la llamada a la función Act_periodicamente, con un periodo de 1 minuto. Cada vez que se realiza una petición de lectura, los datos leídos se almacenan en una estructura de datos llamada MBus_Values.

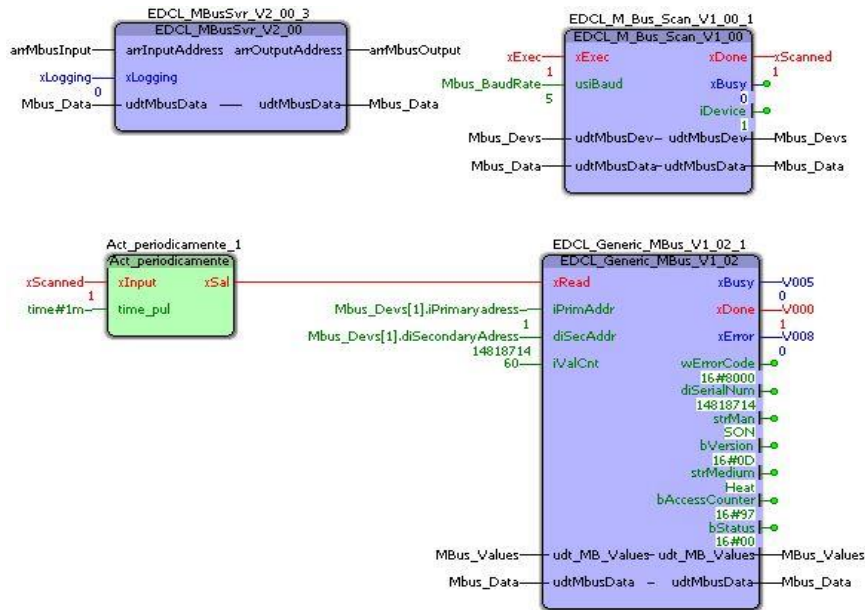
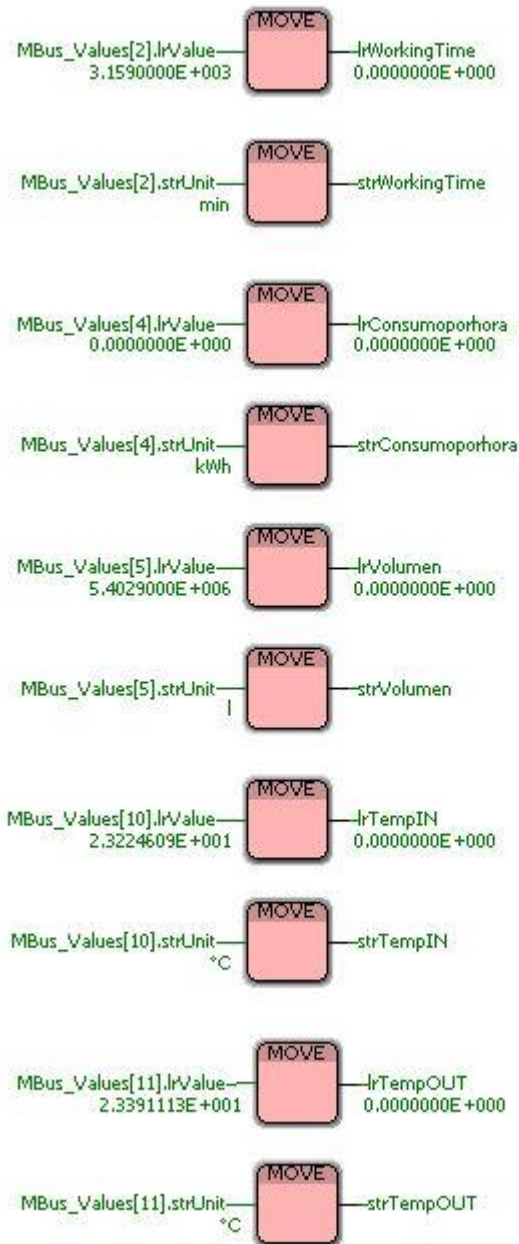


Ilustración 50 ME_Mbus

La extracción de los datos deseados de dicha estructura de datos se realiza a posteriori, dónde se seleccionan los campos necesitados por la aplicación para su posterior almacenamiento y envío a la interface web.





3.2 Modo de Ejecución del Programa PLC.

Una vez programados los scripts correspondientes a la aplicación, es necesario llevar a cabo una planificación del modo de ejecución de los mismos, es decir, la secuencia de ejecución y su periodo. Para ello, hay que definir recursos y tareas. Cada tarea realiza llamada a las instancias de programa que le hayan sido asignadas. Es posible que un mismo recurso realice varias tareas, incluso que estas se repitan. Existen tres tipos de tareas: por defecto, por eventos, del sistema y cíclicas.

En este caso, y siguiendo las recomendaciones del fabricante Phoenix Contact, se define un único recurso "RES1" que realiza una única tarea "td". Para definir el tipo de tarea se realiza un análisis de la carga de la CPU del sistema con distintos tipos de tareas: por defecto, cíclica (50ms) y cíclica (20ms).

A la hora de decidir la configuración de la tarea, hay que tener en cuenta varios aspectos:

- Si se define una tarea por defecto, el PLC realiza la llamada a las instancias de programa cuando cree conveniente. En un principio, puede parecer la mejor opción, pero en el momento que una tarea requiere de más recursos del sistema de lo normal, se corre el peligro de error porque salte el WatchdogTimer del controlador. Se puede observar que la carga de la CPU, en este caso, se corresponde con 70%.

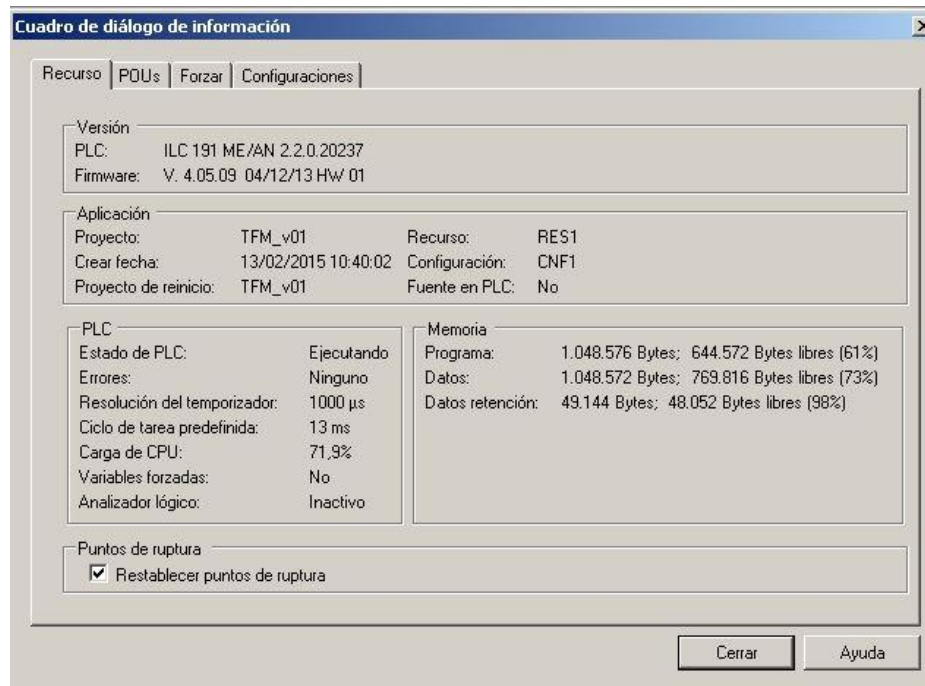


Ilustración 51 Rendimiento PLC Tarea por Defecto

- Si se define una tarea cíclica, es necesario definir el tiempo de ciclo de la misma, y ajustar dicho tiempo a la carga de la CPU del PLC, es decir, a mayor tiempo de ciclo, más lento será el sistema, pero mayor será el rendimiento del PLC, y por tanto más tareas puede ejecutar. Se realizan dos pruebas, cambiando el tiempo de ciclo de ejecución de la tarea: 50ms y 20ms. Cómo se puede observar, la carga a la que se ve sometida la CPU del PLC disminuye aproximadamente la mitad, del 40% al 20%. cuando el periodo de ejecución es de 50 ms.

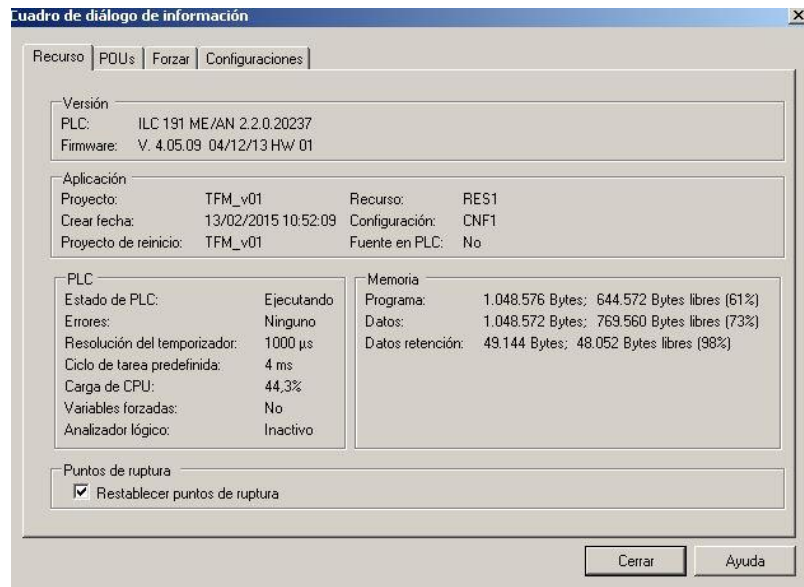


Ilustración 52 Rendimiento PLC tarea ciclica 20ms

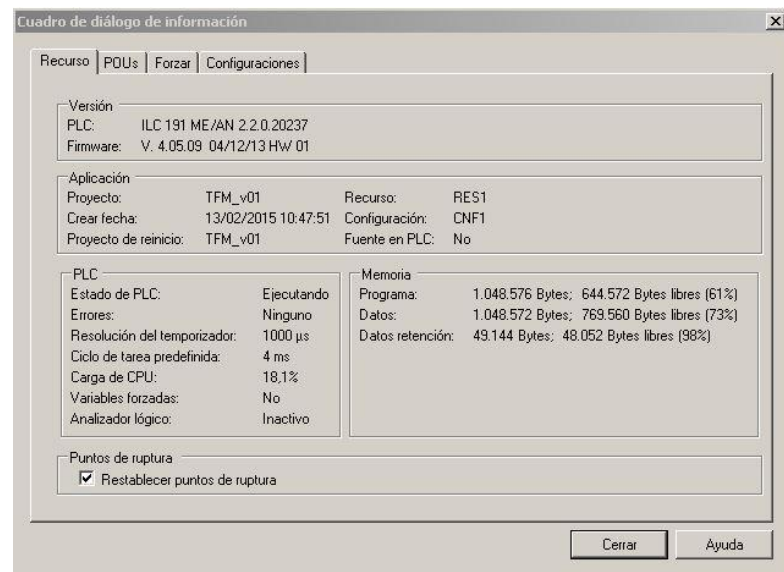


Ilustración 53 Rendimiento PLC tarea ciclica 50ms

Cómo se puede observar, la CPU está menos sobrecargada,(20%), cuando se configura una tarea cíclica con tiempo de ciclo 50ms. Finalmente, se opta por esta configuración ya que, aunque el sistema sea más lento, cumple perfectamente los requerimientos de la aplicación.



3.3 Listado de Variables

A continuación se pueden encontrar una tabla por cada script realizado en el proyecto en el que se encuentra el nombre, tipo, descripción, valor inicial y su asociación (o no) en WebVisit, de cada una de las variables utilizadas en el proyecto.

La columna PDD se corresponde con la asociación de la variable al programa de WebVisit. Nótese que un 0 en esta columna significa que no existe correspondencia, por el contrario, un 1 significa el uso de esta variable en dicho programa.



3.3.1 Sistema

3.3.1.1 IP

3.3.1.1.1 Cambiar_IP

Sistema: Cambiar_Hora				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
CPU_CONTROL_1	CPU_CONTROL	Envio datos Configuración de la CPU para Cambio IP		0
arrSD_1	arrSend_Array	Datos configuracion enviados		0
arrRD_1	arrReceive_Array	Datos configuracion recibidos		0
arrAddError	arr_Add_Error	Código Error		0
xError	BOOL	Error		0
NDR	BOOL	Nuevo Datos Recibidos		0
CPU_CONTROL_2	CPU_CONTROL	Configuración de la CPU para Reseteo		0
Error	arr_Add_Error	Codigo Error Reset		0
Error_Reset	BOOL	Error Reset		0
arrSD_Reset	arrSend_Array	Datos configuracion reseteo enviados		0
arrRD_Reset	arrReceive_Array	Datos configuracion reseteo recibidos		0
Conf_Reset_IP_1	Conf_Reset_IP	Configuracion de la CPU para reseteo		0
Set_IP_1	Set_IP	Recoge nuevos datos IP		0
CPU_CONTROL_3	CPU_CONTROL	Envio datos Configuración de la CPU para reseteo		0
xRequest_IP	BOOL	Solicitud IP actual		0
arrSD_3	arrSend_Array	Datos enviados para configuracion solicitud ip actual		0
arrRD_3	arrReceive_Array	Datos recibidos para configuracion solicitud ip actual		0
addError3	arr_Add_Error	Código Error solicitud ip		0
xiP_Enable	BOOL	Habilita cambio IP		0

Ilustración 54 Variables Cambiar_IP



3.3.1.1.2 Set_IP

Sistema: Set_IP				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
arrSD_1	arrSend_Array	Datos configuracion enviados		0
xEnable	BOOL	Habilita lectura IP		1
R_TRIG_1	R_TRIG	Flanco ascendente		0
xSendData	BOOL	Envia configuración		0
IP1	BYTE	Parametro 1 Dirección IP		1
IP2	BYTE	Parametro 2 Dirección IP		1
IP3	BYTE	Parametro 3 Dirección IP		1
IP4	BYTE	Parametro 4 Dirección IP		1
MASK1	BYTE	Parámetro 1 Mascara de Red		1
MASK2	BYTE	Parametro 2 Mascara de red		1
MASK3	BYTE	Parámetro 3 Mascara de Red		1
MASK4	BYTE	Parámetro 4 Mascara de Red		1
GATEWAY1	BYTE	Parámetro 1 Puerta de Enlace		1
GATEWAY2	BYTE	Parámetro 2 Puerta de Enlace		1
GATEWAY3	BYTE	Parámetro 3 Puerta de Enlace		1
GATEWAY4	BYTE	Parámetro 4 Puerta de Enlace		1

Ilustración 55 Variables Set_IP



3.3.1.1.3 *Conf_Reset_IP*

Sistema: Config_Reset_IP				
Nombre	Tipo	Descripción	Valor Inicial	PDD
FinCambioIP	BOOL	Cambio IP realizado		0
xResetea	BOOL	Habilita Reseteo		0
arrSD_Reset	arrSend_Array	Datos configuracion Reseteo		0

Ilustración 56 *Conf_Reset_IP*



3.3.1.1.4 *Get_IP*

Sistema: Get_IP				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
arrRD_3	arrReceive_Array	Datos configuracion recibidos		0
arrSD_3	arrSend_Array	Datos configuracion enviados		0
GWAY1_actual	BYTE	Parámetro 1 Puerta de Enlace		1
GWAY2_actual	BYTE	Parámetro 2 Puerta de Enlace		1
GWAY3_actual	BYTE	Parámetro 3 Puerta de Enlace		1
GWAY4_actual	BYTE	Parámetro 4 Puerta de Enlace		1
IP1_actual	BYTE	Parametro 1 Dirección IP		1
IP2_actual	BYTE	Parametro 2 Dirección IP		1
IP3_actual	BYTE	Parametro 3 Dirección IP		1
IP4_actual	BYTE	Parametro 4 Dirección IP		1
MASK1_actual	BYTE	Parámetro 1 Mascara de Red		1
MASK2_actual	BYTE	Parametro 2 Mascara de red		1
MASK3_actual	BYTE	Parámetro 3 Mascara de Red		1
MASK4_actual	BYTE	Parámetro 4 Mascara de Red		1
xRequest_IP	BOOL	Solicita actualizacion		0
xStart	BOOL	Empieza actualizar direccion ip		0

Ilustración 57 Variables Get_IP



3.3.1.2 Hora

3.3.1.2.1 *Cambiar_Hora*

Sistema: Cambiar_Hora				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
Set_Time_1	Set_Time	Configuracion Cambio Hora		0
xNDR	BOOL	Nueva Hora Recibida		0
CPU_CONTROL_2	CPU_CONTROL	Envia configuracion a CPU		0
arrRD_2	arrReceive_Array	Datos de configuracion recibidos		0
arrSD_2	arrSend_Array	Datos de configuracion enviados		0
arrAddError2	arr_Add_Error	Código Error		0
xError2	BOOL	Error en cambio hora		0
xP	BOOL	Solicitud cambio hora		0
xSTime_Enable	BOOL	Habilita cambio hora		0

Ilustración 58 Cambiar_Hora



3.3.1.2.2 Set_Time

Sistema: Set_Time				
Nombre	Tipo	Descripción	Valor Inicial	PDD
arrSD_2	arrSend_Array	Datos configuración hora		0
wSecond	WORD	Parámetro segundos (Hora)		0
wMinute	WORD	Parámetro minutos (Hora)		0
wHour	WORD	Parámetro hora (Hora)		0
wDay	WORD	Parámetro día (Fecha)		0
wMonth	WORD	Parámetro Mes (Fecha)		0
wYear	WORD	Parámetro Año (Fecha)		0
xSendData_Time	BOOL	Envía parámetros fecha y hora		0
xEnable_Time	BOOL	Habilita envío datos		0
R_TRIG_2	R_TRIG	Flanco ascendente		0

Ilustración 59 Variables Set_Time



3.3.2 Configuración_Data_Comu

3.3.2.1 Archive Values

3.3.2.1.1 ArchiveValues_Empro

Archive Values: AV_Empro				
Nombre	Tipo	Descripción	Valor Inicial	PDD
xP3	BOOL	Opcion P3 seleccionado Wvisit		1
xPt	BOOL	Opcion Pt seleccionado Wvisit		1
xPt_max	BOOL	Opcion Pt maxima seleccionado Wvisit		1
xPt_med	BOOL	Opcion Pt medio seleccionado Wvisit		1
xQ1	BOOL	Opcion Q1 seleccionado Wvisit		1
xQ2	BOOL	Opcion Q2 seleccionado Wvisit		1
xQ3	BOOL	Opcion Q3 seleccionado Wvisit		1
xQt	BOOL	Opcion Qt seleccionado Wvisit		1
xQt_max	BOOL	Opcion Qt maximo seleccionado Wvisit		1
xQt_med	BOOL	Opcion P3 seleccionado Wvisit		1
xRTU	BOOL	Opcion Modbus RTU seleccionado Wvisit		0
xS1	BOOL	Opcion S1 seleccionado Wvisit		1
xS2	BOOL	Opcion S2 seleccionado Wvisit		1
xS3	BOOL	Opcion S3 seleccionado Wvisit		1
xSt_online	BOOL	Opcion St seleccionado Wvisit		1
xSt_online_max	BOOL	Opcion St maxima seleccionado Wvisit		1
xSt_online_med	BOOL	Opcion St media seleccionado Wvisit		1
xTCP	BOOL	Opcion Modbus TCP seleccionado Wvisit		0



xU12	BOOL	Opcion V12 seleccionado Wvisit		1
xU12_med	BOOL	Opcion V12 media seleccionado Wvisit		1
xU23	BOOL	Opcion V23 seleccionado Wvisit		1
xU23_med	BOOL	Opcion V23media seleccionado Wvisit		1
xU31	BOOL	Opcion V31 seleccionado Wvisit		1
xU31_med	BOOL	Opcion V31 media seleccionado Wvisit		1
xV1	BOOL	Opcion V1 seleccionado Wvisit		1
xV1_max	BOOL	Opcion V1 maxima seleccionado Wvisit		1
xV1_med	BOOL	Opcion V1 media seleccionado Wvisit		1
xV2	BOOL	Opcion V2 seleccionado Wvisit		1
xV2_max	BOOL	Opcion V2 maxima seleccionado Wvisit		1
xV2_med	BOOL	Opcion V2 media seleccionado Wvisit		1
xV3	BOOL	Opcion V3 seleccionado Wvisit		1
xV3_max	BOOL	Opcion V3 maxima seleccionado Wvisit		1
xV3_med	BOOL	Opcion V3media seleccionado Wvisit		1

Ilustración 60 Variables AV_Empo



3.3.2.1.2 *ArchiveValues_PulsesS0*

Archive Values: AV_PulsesS0				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
xUnidades	BOOL	Opcion Unidades seleccionado Wvisit		0
xValorPulso	BOOL	Opcion Valor Pulso seleccionado Wvisit		0
xConsumo	BOOL	Opcion Consumo seleccionado Wvisit		0
xNombre	BOOL	Opcion Nombre seleccionado Wvisit		0
DataLogInt_V0_18_2	DataLogInt_V0_18	Envia dato a estructura de datos		0
DataLogInt_V0_18_3	DataLogInt_V0_18	Envia dato a estructura de datos		0
DataLogStr_V0_18_1	DataLogStr_V0_18	Envia dato a estructura de datos		0
DataLogStr_V0_18_2	DataLogStr_V0_18	Envia dato a estructura de datos		0
Consumo_sal	INT	Dato consumo		0
Valor_Pulso_sal	INT	Dato valor del pulso		0
Entrada	STRING	Dato nombre de entrada		0
Unidades_sal	STRING	Dato unidades		0
Data	udtDL_V3	Estructura de datos		0

Ilustración 61 Variables AV_PulsesS0



3.3.2.1.3 AV_AnalogTemp

Archive Values: AV_AnalogTemp				
Nombre	Tipo	Descripción	Valor Inicial	PDD
xTemp01	BOOL	Opcion Temp1 seleccionado Wvisit		1
DataLogReal_V0_18_1	DataLogReal_V0_18	Envia dato a estructura de datos		0
Temperatura01	REAL	Dato Temperatura Canal 1		0
Data	udtDL_V3	Estructura de datos		0
DataLogReal_V0_18_2	DataLogReal_V0_18	Envia dato a estructura de datos		0
xTemp02	BOOL	Opcion Temp2 seleccionado Wvisit		1
Temperatura02	REAL	Dato Temperatura Canal 2		0

Ilustración 62 Variables AV_AnalogTemp



3.3.2.1.4 AV_PM

Archive Values: AV_PM				
Nombre	Tipo	Descripción	Valor Inicial	PDD
DataLogReal_V0_18_1	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_2	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_3	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_4	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_5	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_6	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_7	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_8	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_9	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_10	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_11	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_12	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_13	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_14	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_15	DataLogReal_V0_18	Envia dato a estructura de datos		0
xPM_V1	BOOL	Opcion V1 seleccionado Wvisit		1
xPM_V2	BOOL	Opcion V2 seleccionado Wvisit		1
xPM_V3	BOOL	Opcion V3 seleccionado Wvisit		1
xPM_V12	BOOL	Opcion V12 seleccionado Wvisit		1



xPM_V23	BOOL	Opcion V23 seleccionado Wvisit	1
xPM_V31	BOOL	Opcion V31 seleccionado Wvisit	1
xPM_I1	BOOL	Opcion I1 seleccionado Wvisit	1
xPM_I2	BOOL	Opcion I2 seleccionado Wvisit	1
xPM_I3	BOOL	Opcion I3 seleccionado Wvisit	1
xPM_In	BOOL	Opcion In seleccionado Wvisit	1
xPM_P1	BOOL	Opcion P1 seleccionado Wvisit	1
xPM_P2	BOOL	Opcion P2 seleccionado Wvisit	1
xPM_P3	BOOL	Opcion P3 seleccionado Wvisit	1
xPM_Frec	BOOL	Opcion Frecuencia seleccionado Wvisit	1
Data	udtDL_V3	Estructura de datos	0
PM_V1	REAL	Dato Tension Linea1	0
PM_V2	REAL	Dato Tension Linea2	0
PM_V3	REAL	Dato Tension Linea3	0
PM_V12	REAL	Dato Tension Fase 12	0
PM_V23	REAL	Dato Tension Fase 23	0
PM_V31	REAL	Dato Tension Fase 31	0
PM_I1	REAL	Dato Intensidad Linea1	0
PM_I2	REAL	Dato Intensidad Linea2	0
PM_I3	REAL	Dato Intensidad Linea3	0
PM_In	REAL	Dato Intensidad nominal	0
PM_P1	REAL	Dato Potencia Linea1	0
PM_P2	REAL	Dato Potencia Linea2	0



PM_P3	REAL	Dato Potencia Linea3	0
PM_Pt	REAL	Dato Potencia Total	0
PM_Frec	REAL	Dato Frecuencia	0
xPM_Pt	BOOL	Opcion Pt seleccionado Wvisit	1

Ilustración 63 Variables AV_PM



3.3.2.1.5 AV_IOLink

Archive Values: AV_IOLink				
Nombre	Tipo	Descripción	Valor Inicial	PDD
Salida_IOLink	REAL	Datos IOLink		0
xIOLink	BOOL	Opcion IOLink seleccionado Wvisit		1
DataLogReal_V0_18_1	DataLogReal_V0_18	Envia dato a estructura de datos		0
Data	udtDL_V3	Estructura de datos		0

Ilustración 64 Variables AV_IOLink



3.3.2.1.6 AV_Mbus

Archive Values: AV_Mbus				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
xMbus_Horasfuncionamiento	BOOL	Opcion HorasFuncionamiento seleccionado Wvisit		1
xConsumoporhora	BOOL	Opcion Consumo seleccionado Wvisit		1
xVolumen	BOOL	Opcion Volumen seleccionado Wvisit		1
xTempIN	BOOL	Opcion Temperatura Entrada seleccionado Wvisit		1
xTempOUT	BOOL	Opcion Temperatura Salida seleccionado Wvisit		1
DataLogReal_V0_18_1	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_2	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_3	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_4	DataLogReal_V0_18	Envia dato a estructura de datos		0
DataLogReal_V0_18_5	DataLogReal_V0_18	Envia dato a estructura de datos		0
IrWorkingTime	REAL	Dato valor del pulso		0
IrConsumoporHora	REAL	Dato nombre de entrada		0
IrVolumen	REAL	Dato unidades		0
IrTempIN	REAL	Estructura de datos		0
IrTempOUT	REAL	Dato unidades		0

Ilustración 65 Variables AV_MBus



3.3.2.2 Communication Settings

3.3.2.2.1 BBDD_Write

Communication Settings: BBDD_Write				
Nombre	Tipo	Descripción	Valor Inicial	PDD
DBFL_RealToComT2_V1_01_43	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_44	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_45	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_46	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_47	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_48	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_49	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_5	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_50	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_51	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_52	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_53	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_54	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_55	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_56	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_57	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_58	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_59	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0



DBFL_RealToComT2_V1_01_6	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_60	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_61	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_62	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_63	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_64	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_65	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_66	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_67	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_68	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_69	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_7	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_70	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_71	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_72	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_8	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_RealToComT2_V1_01_9	DBFL_RealToComT2_V1_01	Envío datos a BBDD		0
DBFL_StartComT1_V1_00_1	DBFL_StartComT1_V1_00	Empieza comunicación con BBDD		0
DBFL_StrToComT1_V1_00_1	DBFL_StrToComT1_V1_00	Envío datos a BBDD		0
diColCnt	DINT	Numero columnas BBDD		0
diRowCnt	DINT	Numero filas BBDD		0
diSizDat	DINT	Tamaño Datos		0
Entrada	STRING	Dato		0
Entrada_BBDD	STRING	Dato		0



fdp1	REAL	Dato	0
fdp2	REAL	Dato	0
fdp3	REAL	Dato	0
fdpt	REAL	Dato	0
Frec	REAL	Dato	0
Frec_max	REAL	Dato	0
Frec_med	REAL	Dato	0
I1	REAL	Dato	0
I1_max	REAL	Dato	0
I1_med	REAL	Dato	0
I2	REAL	Dato	0
I2_max	REAL	Dato	0
I2_med	REAL	Dato	0
I3	REAL	Dato	0
I3_max	REAL	Dato	0
I3_med	REAL	Dato	0
In	REAL	Dato	0
In_max	REAL	Dato	0
In_med	REAL	Dato	0
P1	REAL	Dato	0
P2	REAL	Dato	0
P3	REAL	Dato	0
PM_Frec	REAL	Dato	0
PM_I1	REAL	Dato	0



PM_I2	REAL	Dato		0
PM_I3	REAL	Dato		0
PM_In	REAL	Dato		0
PM_P1	REAL	Dato		0
PM_P2	REAL	Dato		0
PM_P3	REAL	Dato		0
PM_Pt	REAL	Dato		0
PM_V1	REAL	Dato		0
PM_V12	REAL	Dato		0
PM_V2	REAL	Dato		0
PM_V23	REAL	Dato		0
PM_V3	REAL	Dato		0
PM_V31	REAL	Dato		0
Pt	REAL	Dato		0
Pt_max	REAL	Dato		0
Pt_med	REAL	Dato		0
Q1	REAL	Dato		0
Q2	REAL	Dato		0
Q3	REAL	Dato		0
Qt	REAL	Dato		0
Qt_max	REAL	Dato		0
Qt_med	REAL	Dato		0
ReceiveBuffer	DBFL_ARR_BYTE_0_1439	Buffer de datos		0
RTC_DAY	INT	Tiempo de sistema (día)		0



RTC_HOURS	INT	Tiempo de sistema (horas)	0
RTC_MINUTES	INT	Tiempo de sistema (minutos)	0
RTC_MONTH	INT	Tiempo de sistema (mes)	0
RTC_SECONDS	INT	Tiempo de sistema (segundos)	0
RTC_YEAR	INT	Tiempo de sistema (año)	0
S1	REAL	Dato	0
S2	REAL	Dato	0
S3	REAL	Dato	0
S_t	REAL	Dato	0
Salida_IOLink	REAL	Dato	0
SelecTCPRTU_1	SelecTCPRTU	RTU o TCP	0
SelecTCPRTU_2	SelecTCPRTU	RTU o TCP	0
SendBuffer	DBFL_UDT_SQL_COMMAND	Buffer de datos	0
St_max	REAL	Dato	0
St_med	REAL	Dato	0
str_IPserver_BBDD	STRING	IP Servidor BBDD	0
str_Name_BBDD	STRING	Nombre BBDD	0
str_password_BBDD	STRING	Contraseña BBDD	0
str_User_BBDD	STRING	Usuario BBDD	0
strBBDD_ErrMsg	STRING	Mensaje Error BBDD	0
strBBDD_StartCommand	STRING	Comando MySQL para BBDD	0
Temperatura01	REAL	Dato	0
U12	REAL	Dato	0
U12_med	REAL	Dato	0



U23	REAL	Dato		0
U23_med	REAL	Dato		0
U31	REAL	Dato		0
U31_med	REAL	Dato		0
Unidades_BBDD	STRING	Dato		0
Unidades_sal	STRING	Dato		0
V1	REAL	Dato		0
V1_max	REAL	Dato		0
V1_med	REAL	Dato		0
V2	REAL	Dato		0
V2_max	REAL	Dato		0
V2_med	REAL	Dato		0
V3	REAL	Dato		0
V3_max	REAL	Dato		0
V3_med	REAL	Dato		0
Valor_Pulso_sal	INT	Dato		0
WVis_Activar_BBDD	BOOL	Activar comunicación BBDD desde Wvisit		0
WVis_BBDD_BBDD	BOOL	Nombre BBDD enviado desde Wvisit		0
WVis_IPserver_BBDD	STRING	IP Servidor BBDD enviado desde Wvisit		0
WVis_Name_BBDD	STRING	Nombre enviado desde Wvisit		0
WVis_Password_BBDD	STRING	Contraseña enviado desde Wvisit		0
WVis_User_BBDD	STRING	Usuario enviado desde Wvisit		0
xActivarConexion_BBDD	BOOL	Activar comunicación BBDD		0
xActivarTemp	BOOL	Temporizacion		0



xBBDD_Almacenar	BOOL	Activar Almacenar	0
xBBDD_SQL_DONE	BOOL	Comunicación hecha	0
xBBDD_SQL_Ready	BOOL	BBDD Lista	0
xBBDD_TCP_Ready	BOOL	Conexión Lista	0
xConectar_BBDD	BOOL	Conectar con BBDD	0
xError_bbdd	BOOL	Error al conectar con BBDD	0
xOUTConsumo	BOOL	Dato enviado	0
xOUTDateTime	BOOL	Dato enviado	0
xOUTEntrada	BOOL	Dato enviado	0
xOUTfdp1	BOOL	Dato enviado	0
xOUTfdp2	BOOL	Dato enviado	0
xOUTfdp3	BOOL	Dato enviado	0
xOUTfdpt	BOOL	Dato enviado	0
xOUTFrec	BOOL	Dato enviado	0
xOUTFrec_max	BOOL	Dato enviado	0
xOUTFrec_med	BOOL	Dato enviado	0
xOUTI1	BOOL	Dato enviado	0
xOUTI1_max	BOOL	Dato enviado	0
xOUTI1_med	BOOL	Dato enviado	0
xOUTI2	BOOL	Dato enviado	0
xOUTI2_max	BOOL	Dato enviado	0
xOUTI2_med	BOOL	Dato enviado	0
xOUTI3	BOOL	Dato enviado	0
xOUTI3_max	BOOL	Dato enviado	0



xOUTI3_med	BOOL	Dato enviado	0
xOUTIn	BOOL	Dato enviado	0
xOUTIn_max	BOOL	Dato enviado	0
xOUTIn_med	BOOL	Dato enviado	0
xOUTIOLink	BOOL	Dato enviado	0
xOUTP1	BOOL	Dato enviado	0
xOUTP2	BOOL	Dato enviado	0
xOUTP3	BOOL	Dato enviado	0
xOUTPM_I1	BOOL	Dato enviado	0
xOUTPM_I2	BOOL	Dato enviado	0
xOUTPM_I3	BOOL	Dato enviado	0
xOUTPM_In	BOOL	Dato enviado	0
xOUTPM_P1	BOOL	Dato enviado	0
xOUTPM_P2	BOOL	Dato enviado	0
xOUTPM_P3	BOOL	Dato enviado	0
xOUTPM_Pt	BOOL	Dato enviado	0
xOUTPM_V1	BOOL	Dato enviado	0
xOUTPM_V12	BOOL	Dato enviado	0
xOUTPM_V2	BOOL	Dato enviado	0
xOUTPM_V23	BOOL	Dato enviado	0
xOUTPM_V3	BOOL	Dato enviado	0
xOUTPM_V31	BOOL	Dato enviado	0
xOUTPt	BOOL	Dato enviado	0
xOUTpt100	BOOL	Dato enviado	0



xOUTPt_max	BOOL	Dato enviado	0
xOUTPt_med	BOOL	Dato enviado	0
xOUTQ1	BOOL	Dato enviado	0
xOUTQ2	BOOL	Dato enviado	0
xOUTQ3	BOOL	Dato enviado	0
xOUTQt	BOOL	Dato enviado	0
xOUTQt_max	BOOL	Dato enviado	0
xOUTQt_med	BOOL	Dato enviado	0
xOUTS1	BOOL	Dato enviado	0
xOUTS2	BOOL	Dato enviado	0
xOUTS3	BOOL	Dato enviado	0
xOUTSt	BOOL	Dato enviado	0
xOUTSt_max	BOOL	Dato enviado	0
xOUTSt_med	BOOL	Dato enviado	0
xOUTStartCom	BOOL	Dato enviado	0
xOUTTemp	BOOL	Dato enviado	0
xOUTU12	BOOL	Dato enviado	0
xOUTU23	BOOL	Dato enviado	0
xOUTU23_med	BOOL	Dato enviado	0
xOUTU31	BOOL	Dato enviado	0
xOUTU31_med	BOOL	Dato enviado	0
xOUTUnidades	BOOL	Dato enviado	0
xOUTV1	BOOL	Dato enviado	0
xOUTV1_max	BOOL	Dato enviado	0



xOUTV1_med	BOOL	Dato enviado	0
xOUTV2	BOOL	Dato enviado	0
xOUTV2_max	BOOL	Dato enviado	0
xOUTV2_med	BOOL	Dato enviado	0
xOUTV3	BOOL	Dato enviado	0
xOUTV3_max	BOOL	Dato enviado	0
xOUTV3_med	BOOL	Dato enviado	0
xOUTValPulso	BOOL	Dato enviado	0
xtcpstatus_bbdd	DINT	Estado conexión BBDD	0

Ilustración 66 Variables BBDD_Write



3.3.2.2.2 *Act_Periodicamente*

Communication Settings: Act_periodicamente				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
xSal	BOOL	Salida funcion		0
xInput	BOOL	Entrada funcion		0
RS_1	RS	Biestable RS		0
TP_1	TP	Pulso		0
time_pul	TIME	Valor Temporizacion		0

Ilustración 67 Variables Act_Periodicamente



3.3.2.2.3 StrBBDD

Communication Settings: BBDD_Write				
Nombre	Tipo	Descripción	Valor Inicial	PDD
DBFL_RealToComT2_V1_01_43	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_44	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_45	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_46	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_47	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_48	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_49	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_5	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_50	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_51	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_52	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_53	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_54	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_55	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_56	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_57	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_58	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_59	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_6	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_60	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0



DBFL_RealToComT2_V1_01_61	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_62	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_63	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_64	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_65	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_66	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_67	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_68	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_69	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_7	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_70	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_71	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_72	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_8	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_RealToComT2_V1_01_9	DBFL_RealToComT2_V1_01	Envio datos a BBDD		0
DBFL_StartComT1_V1_00_1	DBFL_StartComT1_V1_00	Empieza comunicación con BBDD		0
DBFL_StrToComT1_V1_00_1	DBFL_StrToComT1_V1_00	Envio datos a BBDD		0
diColCnt	DINT	Numero columnas BBDD		0
diRowCnt	DINT	Numero filas BBDD		0
diSizDat	DINT	Tamaño Datos		0
Entrada	STRING	Dato		0
Entrada_BBDD	STRING	Dato		0
fdp1	REAL	Dato		0
fdp2	REAL	Dato		0



fdp3	REAL	Dato	0
fdpt	REAL	Dato	0
Frec	REAL	Dato	0
Frec_max	REAL	Dato	0
Frec_med	REAL	Dato	0
I1	REAL	Dato	0
I1_max	REAL	Dato	0
I1_med	REAL	Dato	0
I2	REAL	Dato	0
I2_max	REAL	Dato	0
I2_med	REAL	Dato	0
I3	REAL	Dato	0
I3_max	REAL	Dato	0
I3_med	REAL	Dato	0
In	REAL	Dato	0
In_max	REAL	Dato	0
In_med	REAL	Dato	0
P1	REAL	Dato	0
P2	REAL	Dato	0
P3	REAL	Dato	0
PM_Frec	REAL	Dato	0
PM_I1	REAL	Dato	0
PM_I2	REAL	Dato	0
PM_I3	REAL	Dato	0



PM_In	REAL	Dato	0
PM_P1	REAL	Dato	0
PM_P2	REAL	Dato	0
PM_P3	REAL	Dato	0
PM_Pt	REAL	Dato	0
PM_V1	REAL	Dato	0
PM_V12	REAL	Dato	0
PM_V2	REAL	Dato	0
PM_V23	REAL	Dato	0
PM_V3	REAL	Dato	0
PM_V31	REAL	Dato	0
Pt	REAL	Dato	0
Pt_max	REAL	Dato	0
Pt_med	REAL	Dato	0
Q1	REAL	Dato	0
Q2	REAL	Dato	0
Q3	REAL	Dato	0
Qt	REAL	Dato	0
Qt_max	REAL	Dato	0
Qt_med	REAL	Dato	0
ReceiveBuffer	DBFL_ARR_BYTE_0_1439	Buffer de datos	0
RTC_DAY	INT	Tiempo de sistema (día)	0
RTC_HOURS	INT	Tiempo de sistema (horas)	0
RTC_MINUTES	INT	Tiempo de sistema (minutos)	0



RTC_MONTH	INT	Tiempo de sistema (mes)	0
RTC_SECONDS	INT	Tiempo de sistema (segundos)	0
RTC_YEAR	INT	Tiempo de sistema (año)	0
S1	REAL	Dato	0
S2	REAL	Dato	0
S3	REAL	Dato	0
S_t	REAL	Dato	0
Salida_IOLink	REAL	Dato	0
SelecTCPRTU_1	SelecTCPRTU	RTU o TCP	0
SelecTCPRTU_2	SelecTCPRTU	RTU o TCP	0
SendBuffer	DBFL_UDT_SQL_COMMAND	Buffer de datos	0
St_max	REAL	Dato	0
St_med	REAL	Dato	0
str_IPserver_BBDD	STRING	IP Servidor BBDD	0
str_Name_BBDD	STRING	Nombre BBDD	0
str_password_BBDD	STRING	Contraseña BBDD	0
str_User_BBDD	STRING	Usuario BBDD	0
strBBDD_ErrMsg	STRING	Mensaje Error BBDD	0
strBBDD_StartCommand	STRING	Comando MySQL para BBDD	0
Temperatura01	REAL	Dato	0
U12	REAL	Dato	0
U12_med	REAL	Dato	0
U23	REAL	Dato	0
U23_med	REAL	Dato	0



U31	REAL	Dato		0
U31_med	REAL	Dato		0
Unidades_BBDD	STRING	Dato		0
Unidades_sal	STRING	Dato		0
V1	REAL	Dato		0
V1_max	REAL	Dato		0
V1_med	REAL	Dato		0
V2	REAL	Dato		0
V2_max	REAL	Dato		0
V2_med	REAL	Dato		0
V3	REAL	Dato		0
V3_max	REAL	Dato		0
V3_med	REAL	Dato		0
Valor_Pulso_sal	INT	Dato		0
WVis_Activar_BBDD	BOOL	Activar comunicación BBDD desde Wvisit		0
WVis_BBDD_BBDD	BOOL	Nombre BBDD enviado desde Wvisit		0
WVis_IPserver_BBDD	STRING	IP Servidor BBDD enviado desde Wvisit		0
WVis_Name_BBDD	STRING	Nombre enviado desde Wvisit		0
WVis_Password_BBDD	STRING	Contraseña enviado desde Wvisit		0
WVis_User_BBDD	STRING	Usuario enviado desde Wvisit		0
xActivarConexion_BBDD	BOOL	Activar comunicación BBDD		0
xActivarTemp	BOOL	Temporizacion		0
xBBDD_Almacenar	BOOL	Activar Almacenar		0
xBBDD_SQL_DONE	BOOL	Comunicación hecha		0



xBBDD_SQL_Ready	BOOL	BBDD Lista		0
xBBDD_TCP_Ready	BOOL	Conexión Lista		0
xConectar_BBDD	BOOL	Conectar con BBDD		0
xError_bbdd	BOOL	Error al conectar con BBDD		0
xOUTConsumo	BOOL	Dato enviado		0
xOUTDateTime	BOOL	Dato enviado		0
xOUTEntrada	BOOL	Dato enviado		0
xOUTfdp1	BOOL	Dato enviado		0
xOUTfdp2	BOOL	Dato enviado		0
xOUTfdp3	BOOL	Dato enviado		0
xOUTfdpt	BOOL	Dato enviado		0
xOUTFrec	BOOL	Dato enviado		0
xOUTFrec_max	BOOL	Dato enviado		0
xOUTFrec_med	BOOL	Dato enviado		0
xOUTI1	BOOL	Dato enviado		0
xOUTI1_max	BOOL	Dato enviado		0
xOUTI1_med	BOOL	Dato enviado		0
xOUTI2	BOOL	Dato enviado		0
xOUTI2_max	BOOL	Dato enviado		0
xOUTI2_med	BOOL	Dato enviado		0
xOUTI3	BOOL	Dato enviado		0
xOUTI3_max	BOOL	Dato enviado		0
xOUTI3_med	BOOL	Dato enviado		0
xOUTIn	BOOL	Dato enviado		0



xOUTIn_max	BOOL	Dato enviado	0
xOUTIn_med	BOOL	Dato enviado	0
xOUTIOLink	BOOL	Dato enviado	0
xOUTP1	BOOL	Dato enviado	0
xOUTP2	BOOL	Dato enviado	0
xOUTP3	BOOL	Dato enviado	0
xOUTPM_I1	BOOL	Dato enviado	0
xOUTPM_I2	BOOL	Dato enviado	0
xOUTPM_I3	BOOL	Dato enviado	0
xOUTPM_In	BOOL	Dato enviado	0
xOUTPM_P1	BOOL	Dato enviado	0
xOUTPM_P2	BOOL	Dato enviado	0
xOUTPM_P3	BOOL	Dato enviado	0
xOUTPM_Pt	BOOL	Dato enviado	0
xOUTPM_V1	BOOL	Dato enviado	0
xOUTPM_V12	BOOL	Dato enviado	0
xOUTPM_V2	BOOL	Dato enviado	0
xOUTPM_V23	BOOL	Dato enviado	0
xOUTPM_V3	BOOL	Dato enviado	0
xOUTPM_V31	BOOL	Dato enviado	0
xOUTPt	BOOL	Dato enviado	0
xOUTPt100	BOOL	Dato enviado	0
xOUTPt_max	BOOL	Dato enviado	0
xOUTPt_med	BOOL	Dato enviado	0



xOUTQ1	BOOL	Dato enviado	0
xOUTQ2	BOOL	Dato enviado	0
xOUTQ3	BOOL	Dato enviado	0
xOUTQt	BOOL	Dato enviado	0
xOUTQt_max	BOOL	Dato enviado	0
xOUTQt_med	BOOL	Dato enviado	0
xOUTS1	BOOL	Dato enviado	0
xOUTS2	BOOL	Dato enviado	0
xOUTS3	BOOL	Dato enviado	0
xOUTSt	BOOL	Dato enviado	0
xOUTSt_max	BOOL	Dato enviado	0
xOUTSt_med	BOOL	Dato enviado	0
xOUTStartCom	BOOL	Dato enviado	0
xOUTTemp	BOOL	Dato enviado	0
xOUTU12	BOOL	Dato enviado	0
xOUTU23	BOOL	Dato enviado	0
xOUTU23_med	BOOL	Dato enviado	0
xOUTU31	BOOL	Dato enviado	0
xOUTU31_med	BOOL	Dato enviado	0
xOUTUnidades	BOOL	Dato enviado	0
xOUTV1	BOOL	Dato enviado	0
xOUTV1_max	BOOL	Dato enviado	0
xOUTV1_med	BOOL	Dato enviado	0
xOUTV2	BOOL	Dato enviado	0



xOUTV2_max	BOOL	Dato enviado	0
xOUTV2_med	BOOL	Dato enviado	0
xOUTV3	BOOL	Dato enviado	0
xOUTV3_max	BOOL	Dato enviado	0
xOUTV3_med	BOOL	Dato enviado	0
xOUTValPulso	BOOL	Dato enviado	0
xtcpstatus_bbdd	DINT	Estado conexión BBDD	0

Ilustración 68 Variables StrBBDD



3.3.2.2.4 *SelectTCPRTU*

Communication Settings: SelectTCPRTU				
Nombre	Tipo	Descripción	Valor Inicial	PDD
INPUT_TCP	REAL	Variable modbus tcp		0
INPUT_RTU	REAL	Variable modbus rtu		0
salida	REAL	salida bloque		0
xRTU	BOOL	rtu activo		0
xTCP	BOOL	rtu tcp		0

Ilustración 69 SelectTCPRTU



3.3.2.2.5 FTP:FTP_Settings

Communication Settings: FTP_Settings				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
Act_periodicamente_3	Act_periodicamente	Bloque temporizacion		0
Act_periodicamente_4	Act_periodicamente	Bloque temporizacion		0
Data	udtDL_V3	Estructura de datos		0
Data_captu	udtDataLoggerInterface_V1	Copia de datos a enviar FTP		0
Data_captu_PLC	udtDataLoggerInterface_V1	Copia de datos a enviar PLC		0
DataLogCSV_V0_18_1	DataLogCSV_V0_18	Datalogger Almacenamiento local		0
DataLogFTP_V0_18_1	DataLogFTP_V0_18	Datalogger Almacenamiento FTP		0
icreatenewfile	INT	Crear nuevo archivo		0
iPort_FTP	INT	Puerto servidor FTP	21	0
iUsedMem_PLC	INT	%Memoria utilizado plc		0
iusedmemory_ftp	INT	%Memoria utilizado ftp		0
ONBOARD_INPUT	WORD	Entradas locales		0
str_FileName	STRING	Nombre fichero		0
str_IPserver	STRING	IP servidor FTP		0
str_password_ftp	STRING	Contraseña FTP		0
str_User_ftp	STRING	Usuario FTP		0
strActFile_csv	STRING	Archivo actual PLC		0
strActFile_ftp	STRING	Archivo actual FTP		0
strLastFile_csv	STRING	Ultimo archivo PLC		0
strLastFile_ftp	STRING	Ultimo archivo FTP		0
strLastResponse	STRING			0
temporizacion_FTP	TIME	Periodo toma datos en FTP	time#3m	0



temporizacion_PLC	TIME	Periodo toma datos en PLC	time#5m	0
wDiagCode_Datalogger_FTP	WORD	Codigo error datalogger FTP		0
wDiagCode_Datalogger_PLC	WORD	Código error datalogger PLC		0
WVis_CSV	BOOL	Activacion CSV LOCAL		1
WVis_FileName	STRING	Nombre fichero enviado desde Wvisit	'hello'	1
WVis_IPserver	STRING	Servidor IP	'192.168.1.110'	1
WVis_Password	STRING	Contraseña FTP	'anonymous'	1
WVis_User	STRING	Usuario FTP	'anonymous'	1
xActivar_Datalogger	BOOL	Activacion datalogger		0
xAddTime	BOOL	Añadir Fecha y Hora al fichero		0
xAlmacenar	BOOL	Almacenar		0
xError_CSV	BOOL	Error datalogger		0
xError_Datalogger_FTP	BOOL	Error datalogger		0
xError_Datalogger_PLC	BOOL	Error datalogger		0
xError_ftp	BOOL	Error datalogger		0
xNewFileCreate_PLC	BOOL	Crear nuvo archivo en PLC		0
xReady_CSV	BOOL	Almacenamiento local preparado		0
xReady_Datalogger_FTP	BOOL	Datalogger FTP Listo		0
xReady_Datalogger_PLC	BOOL	Datalogger PLC listo		0
xReady_FTP	BOOL	Servidor FTP listo para almacenamiento		0
xReq_Datalogger_FTP	BOOL	Solicitud Datalogger FTP		0
xReq_Datalogger_PLC	BOOL	Solicitud Datalogger PLC		0

Ilustración 70 FTP_Settings



3.3.2.2.6 Email

Communication Settings: Email				
Nombre	Tipo	Descripción	Valor Inicial	PDD
SMTP_Client_V1_16_1	SMTP_Client_V1_16			0
strAsunto	STRING			0
WV_Asunto	STRING	Asunto enviado desde WVisit	'Correo ILC'	1
xActivar_Mail	BOOL	Activar Envio Mail		0
WV_Destinataro	STRING	Destinatario enviado desde WVisit	'pcworx_prueba@gmx.es'	1
WV_Mensaje	SMTP_STR_200	Mensaje enviado desde WVisit	'Correo Enviado desde ILC'	1
WV_ServerAddr	STRING	Servidor enviado desde Wvisit	'mail.gmx.com'	1
WV_Remitente	STRING	Remitente enviado desde WVisit	'pcworx_prueba@gmx.es'	1
xDone_Mail	BOOL	Mail enviado		0
WV_IP_ServerAddress	STRING	Direcion IP servidor enviado desde WVisit	'8.8.8.8'	1
FileName	SMTP_ARR_STR_1_10	Nombre archivo		0
FirstByte	BYTE			0
xError_mail	BOOL	Error envio mail		0
xWDiagCode	WORD	Codigo Error		0
WaDDDIAGCODE	WORD	Código Error		0
FileNames	SMTP_ARR_STR_1_10	Array con nombre ficheros		0
strLastFile_ftp	STRING	Ultimo archivo creado FTP		0
xnewFile_CSV	BOOL	Crear nuevo CSV		0
WV_User	STRING	Usuario enviado desde WVisit	'pcworx_prueba@gmx.es'	1
WV_Password	STRING	Contraseña enviado desde WVisit	'12345678'	1



xOpcion_Mail	BOOL	Activar o no Mail	1
strLastFile_csv	STRING	Ultimo archivo creado	0
xnewFile_CSV_PLC	BOOL	Nuevo archivo en PLC	0

Ilustración 71 Variables Email



3.3.2.2.7 Modbus:Settings_Modbus

Communication Settings: Settings Modbus				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
IP_TCP	STRING	Direcion IP TCP		0
Cf_IP_TCP	STRING	Direcion IP TCP enviado desde Wvisit	'192.168.1.112'	1
Cf_diBaudrate	DINT	Baudios RTU enviado desde Wvisit	19200	1
diBaudrate_RTU	DINT	Baudios RTU		0
Cf_uiSlave_Addr_RTU	UINT	Numero de esclavo RTU enviado desde Wvisit		0
Slave_Addr_ValOnline_RTU	UINT	Esclavo RTU		0
Slave_Addr_ValAdvan_RTU	UINT	Esclavo RTU		0
xGuardar_Config	BOOL	Cargar configuracion		0

Ilustración 72 Variables Settings_Modbus



3.3.3 Dispositivos

3.3.3.1 ME_PulsesS0

ME_PulsesS0				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
Convertir	BOOL	Lectura Realizada		0
Entrada	STRING	Dispositivo PulsesS0		0
Contador	INT	Contador pulsos		0
ONBOARD_INPUT	WORD	Local inputs		0
consumo_sal	INT	Consumo total		0
unidades_sal	STRING	Unidad de pulso		1
valor_pulso_sal	INT	Valor por pulso		1
funcion_PulsesS0_1	funcion_PulsesS0	funcion PulsesS0		0

Ilustración 73 Variables ME_PulsesS0



3.3.3.2 ME_Empro

3.3.3.2.1 Modbus_TCP

Empro: ME_Modbus_TCP				
Nombre	Tipo	Descripción	Valor Inicial	PDD
add_DiagCode_Client_tcp	WORD	Codigo Error Cliente TCP		1
add_DiagCode_FC_tcp	WORD	Codigo Error Funcion Modbus TCP		0
Client_Com_Struct	COM_UDT_COMMUNICATION_V2	Cliente TCP		0
DiagCode_Client_tcp	WORD	Codigo Error Cliente TCP		1
DIAGCode_FC_tcp	WORD	Codigo Error Funcion Modbus TCP		1
Error_Client_tcp	BOOL	Error cliente		1
Error_FC_Advan_tcp	BOOL	Error funcion Valores Avanzados		0
Error_Online_FC_tpc	BOOL	Error funcion valores básicos		0
First_Addr_TCP	WORD	Primera direccion a leer val.basicos		0
First_Addr_TCP_ValAdvan	WORD	Primera direccion a leer val.advan		0
ID_Val_Advan	INT	Identificador de lectura	2	0
ID_Val_Linea	INT	Identificador de lectura	1	0
IP_TCP	STRING	Direccion IP		0
MB_TCP_Client_V1_22_2	MB_TCP_Client_V1_22	Bloque Cliente TCP		0
MT_ReadHoldingReg_V1_20_1	MT_ReadHoldingReg_V1_20	Funcion lectura Modbus		0
MT_ReadHoldingReg_V1_20_3	MT_ReadHoldingReg_V1_20	Funcion lectura Modbus		0
NDR_ValAdvan	BOOL	Nuevo Valor Avanzado Recibido		0
NDR_ValLinea	BOOL	Nuevo Valor Básico Recibido		0



Quantity_TCP	UINT	Cantidad de direcciones a leer		0
Quantity_TCP_ValAdvan	UINT	Cantidad de direcciones a leer		0
UnitIdentifier	UINT	Identificador de lectura	1	0
Values_Advan	MB_TCP_ARR_W_1_125	Array con valores avanzados		0
Values_Online	MB_TCP_ARR_W_1_125	Array con valores básicos		0
xClient_Active_TCP	BOOL	Cliente TCP Activo		0
xClient_TCP	BOOL	Activar Cliente TCP		0
xRead_ValAdvan	BOOL	Leer Valores avanzados		0
xReading_Advan_TCP	BOOL	Leyendo valores avanzados		0
xReading_Online_TCP	BOOL	Leyendo valores básicos		0

Ilustración 74 Variables Modbus_TCP



3.3.3.2.2 *Values_Online_Advan*

ME_Values_AdvanOnline				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
U12_tcp	REAL	Dato		0
U23_tcp	REAL	Dato		0
U31_tcp	REAL	Dato		0
V1_tcp	REAL	Dato		0
V2_tcp	REAL	Dato		0
V3_tcp	REAL	Dato		0
frec_tcp	REAL	Dato		0
I1_tcp	REAL	Dato		0
I2_tcp	REAL	Dato		0
I3_tcp	REAL	Dato		0
In_tcp	REAL	Dato		0
Pt_tcp	REAL	Dato		0
Qt_tcp	REAL	Dato		0
St_online_tcp	REAL	Dato		0
fdp_tcp	REAL	Dato		0
P1_tcp	REAL	Dato		0
P2_tcp	REAL	Dato		0
P3_tcp	REAL	Dato		0
Q1_tcp	REAL	Dato		0



Q2_tcp	REAL	Dato	0
Q3_tcp	REAL	Dato	0
S1_tcp	REAL	Dato	0
S2_tcp	REAL	Dato	0
S3_tcp	REAL	Dato	0
fdp1_tcp	REAL	Dato	0
fdp2_tcp	REAL	Dato	0
V3_med_tcp	REAL	Dato	0
V3_max_tcp	REAL	Dato	0
V2_med_tcp	REAL	Dato	0
V2_max_tcp	REAL	Dato	0
V1_med_tcp	REAL	Dato	0
V1_max_tcp	REAL	Dato	0
U31_med_tcp	REAL	Dato	0
U23_med_tcp	REAL	Dato	0
U12_med_tcp	REAL	Dato	0
St_online_med_tcp	REAL	Dato	0
St_online_max_tcp	REAL	Dato	0
Qt_med_tcp	REAL	Dato	0
Qt_max_tcp	REAL	Dato	0
Pt_med_tcp	REAL	Dato	0
Values_Advan	MB_TCP_ARR_W_1_125	Valores Avanzados	0
Pt_max_tcp	REAL	Dato	0
In_med_tcp	REAL	Dato	0



In_max_tcp	REAL	Dato		0
I3_med_tcp	REAL	Dato		0
I3_max_tcp	REAL	Dato		0
Values_Online	MB_TCP_ARR_W_1_125	Valores Basicos		0
I2_med_tcp	REAL	Dato		0
I2_max_tcp	REAL	Dato		0
I1_med_tcp	REAL	Dato		0
I1_max_tcp	REAL	Dato		0
frec_med_tcp	REAL	Dato		0
frec_max_tcp	REAL	Dato		0
fdp3_tcp	REAL	Dato		0
xRead_ValAdvan	BOOL	Activar Lectura Valores Avanzados		0
NDR_ValAdvan	BOOL	Nuevos Datos valores avanzados recibidos		0
RTC_MINUTES	INT	System time (minutes)		0
arrValores_TCP	arrValores_TCP	Valores Basicos		0

Ilustración 75 Variables Values_Online_Advan



3.3.3.2.3 Modbus_RTU

ME_Empromodbus_RTU				
Nombre	Tipo	Descripción	Valor Inicial	PDD
xSendRequest_ValOnline_RTU	BOOL	Peticion valores basicos		0
xRequest_ValAdvan_RTU	BOOL	Peticion valores avanzados		0
xReady_Master_RTU	BOOL	Maestro RTU listo		0
xReady_Function_ValOnline_RTU	BOOL	Funcion lectura valores basicos		0
xReady_Function_ValAdvan_RTU	BOOL	Funcion lectura valores avanzados		0
xRead_ValOnline_RTU	BOOL	Leer valores basicos		0
xRead_ValAdvan_RTU	BOOL	Leer valores avanzados		0
xPoll_ValAdvan	BOOL	Habilitar polling		0
xPoll	BOOL	Habilitar polling		0
xDone_ValOnline_RTU	BOOL	Valores basicos leidos		0
xDone_ValAdvan_RTU	BOOL	Valores avanzados leidos		0
xBusy_Function_ValOnline_RTU	BOOL	Funcion lectura ocupada		0
xBusy_Function_ValAdvan_RTU	BOOL	Funcion lectura ocupada		0
xBusy	BOOL	Maestro ocupado		0
xActivate_Master_RTU	BOOL	Activar maestro rtu		0
StrRTU	MB_RTU_MTR_Data	Estructura datos		0
Start_Addr_ValOnline_RTU	UINT	Primera direccion lectura val.básicos	50514	0
Start_Addr_RTU_ValAdvan	UINT	Primera direccion lectura val.advan	51024	0
Slave_Addr_ValOnline_RTU	UINT	numero esclavo		0
Slave_Addr_ValAdvan_RTU	UINT	numero esclavo		0



Received_ValAdvan_RTU	MB_RTU_arr_w_1_125	Valores Avanzados		0
Received_Data_RTU	MB_RTU_arr_w_1_125	Valores Basicos		0
quantity_valonline_RTU	UINT	Cantidad direcciones a leer		0
Quantity_ValAdvan_RTU	UINT	Cantidad direcciones a leer	46	0
Out_Data_RTU	MB_COM2_ARR_B_1_12	Estructura datos		0
MB_RTU_FC03_V1_02_2	MB_RTU_FC03_V1_02	Funcion lectura 3		0
MB_RTU_FC03_V1_02_1	MB_RTU_FC03_V1_02	Funcion lectura 3		0
MB_RTU485_Master_V1_01_1	MB_RTU485_Master_V1_01	Maestro RTU		0
Input_Data_RTU	MB_COM2_ARR_B_1_12	Estructura datos		0
diBaudrate_RTU	DINT	Velocidad transmision rtu		0
bDataWidth	BYTE	Topologia de datos	16#0008	0

Ilustración 76 Variables Modbus_RTU

3.3.3.2.4 *Values_RTU*

ME_Values_RTU				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
Received_ValAdvan_RTU	MB_RTU_arr_w_1_125	Valores avanzados recibidos		0
xRead_ValAdvan_RTU	BOOL	Leer valores avanzados		0
R_TRIG_Advan_RTU	R_TRIG	Leer valores avanzados		0
xDone_ValAdvan_RTU	BOOL	Valores Avanzados leídos		0
U12_rtu	REAL	Dato		0
U23_rtu	REAL	Dato		0
U31_rtu	REAL	Dato		0
V1_rtu	REAL	Dato		0
V2_rtu	REAL	Dato		0
Received_Data_RTU	MB_RTU_arr_w_1_125	Datos rtu recibidos		0
V3_rtu	REAL	Dato		0
frec_rtu	REAL	Dato		0
I1_rtu	REAL	Dato		0
I2_rtu	REAL	Dato		0
I3_rtu	REAL	Dato		0
In_rtu	REAL	Dato		0
Pt_rtu	REAL	Dato		0
Qt_rtu	REAL	Dato		0
St_online_rtu	REAL	Dato		0
fdp_rtu	REAL	Dato		0



Trabajo Fin de Máster:



Aplicación para la adquisición y gestión de parámetros energéticos

P1_rtu	REAL	Dato	0
P2_rtu	REAL	Dato	0
P3_rtu	REAL	Dato	0
Q1_rtu	REAL	Dato	0
Q2_rtu	REAL	Dato	0
Q3_rtu	REAL	Dato	0
S1_rtu	REAL	Dato	0
S2_rtu	REAL	Dato	0
S3_rtu	REAL	Dato	0
fdp1_rtu	REAL	Dato	0
fdp2_rtu	REAL	Dato	0
V3_med_rtu	REAL	Dato	0
V3_max_rtu	REAL	Dato	0
V2_med_rtu	REAL	Dato	0
V2_max_rtu	REAL	Dato	0
V1_med_rtu	REAL	Dato	0
V1_max_rtu	REAL	Dato	0
U31_med_rtu	REAL	Dato	0
U23_med_rtu	REAL	Dato	0
U12_med_rtu	REAL	Dato	0
St_online_med_rtu	REAL	Dato	0
St_online_max_rtu	REAL	Dato	0
Qt_med_rtu	REAL	Dato	0
Qt_max_rtu	REAL	Dato	0



Pt_med_rtu	REAL	Dato		0
Pt_max_rtu	REAL	Dato		0
In_med_rtu	REAL	Dato		0
In_max_rtu	REAL	Dato		0
I3_med_rtu	REAL	Dato		0
I3_max_rtu	REAL	Dato		0
I2_med_rtu	REAL	Dato		0
I2_max_rtu	REAL	Dato		0
I1_med_rtu	REAL	Dato		0
I1_max_rtu	REAL	Dato		0
frec_med_rtu	REAL	Dato		0
frec_max_rtu	REAL	Dato		0
fdp3_rtu	REAL	Dato		0
RTC_MINUTES	INT	System time (minutes)		0
xRead_ValOnline_RTU	BOOL	Leer valores básicos RTU		0
xSendRequest_ValOnline_RTU	BOOL	Solicitud datos basicos		0
xAtras_RTU	BOOL	Wvisit		0
xActivate_Master_RTU	BOOL	Activar Maestro RTU		0
arrValores_RTU	arrValores_RTU	Array valores RTU		0

Ilustración 77 Variables Modbus_RTU



3.3.3.3 ME_PM

ME_PM				
Nombre	Tipo	Descripción	Valor Inicial	PDD
PM_Select_PD_V1_00_1	PM_Select_PD_V1_00	Bloque comunicación modulo		0
PM_BasicValue_V1_00_2	PM_BasicValue_V1_00	Bloque comunicación modulo		0
PM_3P_N_EF_V1_01_2	PM_3P_N_EF_V1_01	Bloque comunicación modulo		0
IN_pd	PM_ARR_W_0_11	Entrada		0
OUT_pd	PM_ARR_W_0_11	Salida		0
uParam	PM_UDT_PARA	Estructura de datos		0
uData	PM_UDT_DATA	Estructura de datos		0
PM_Configuration_V1_00_1	PM_Configuration_V1_00	Configuración		0
xActivarConfig	BOOL	Activa configuración	TRUE	0
xConfigOk	BOOL	Configuración correcta		0
PM_ExtendedValue_V1_00_1	PM_ExtendedValue_V1_00	Habilita valores avanzados		0
xSelectOK	BOOL	Configurado		0
xReset	BOOL	Resetea lectura		0
xQuit	BOOL	Cierre lectura		0
xPmRun	BOOL	Puesta en marcha lectura		0
iSecCorr	INT	Corriente del secundario	5000	0
bConTyp	BYTE	Tipo Línea	byte#16#16	0
PM_V1	REAL	Dato		0
PM_V2	REAL	Dato		0
PM_V3	REAL	Dato		0



PM_I1	REAL	Dato	0
PM_I2	REAL	Dato	0
PM_I3	REAL	Dato	0
PM_P1	REAL	Dato	0
PM_P2	REAL	Dato	0
PM_P3	REAL	Dato	0
PM_Pt	REAL	Dato	0
PM_In	REAL	Dato	0
PM_V12	REAL	Dato	0
PM_V23	REAL	Dato	0
PM_V31	REAL	Dato	0
PM_Frec	REAL	Dato	0

Ilustración 78 Variables ME_PM



3.3.3.4 ME_AnalogTemp

ME_AnalogTemp				
Nombre	Tipo	Descripción	Valor Inicial	PDD
Input01	WORD	Variable de proceso canal1		0
Temperatura01	REAL	Dato leído canal1		0
Input02	WORD	Variable de proceso canal2		0
Temperatura02	REAL	Dato leído canal2		0

Ilustración 79 Variables ME_AnalogTemp



3.3.3.5 ME_IOLink

ME_IOLink				
Nombre	Tipo	Descripcion	Valor Inicial	PDD
IO_Link_Master_IB_V1_01_1	IO_Link_Master_IB_V1_01	Maestro IOLINK		0
IO_Link_Config_V1_01_1	IO_Link_Config_V1_01	Configuracion IOLink		0
IOLink_Command_1	IOLink_Command	Datos Iolink		0
IO_Link_Port_V1_02_2	IO_Link_Port_V1_02	Puerto Lectura IOLink		0
IOLink_Command_2	IOLink_Command	Datos Iolink		0
xIOL_Master_Ready	BOOL	Maestro IOLINK Listo		0
xIOL_Master_Error	BOOL	Error maestro Iolink		0
xIOL_Master_NewData	BOOL	Nuevo dato recibido		0
xIOL_Read_Master	BOOL	Leer del maestro Iolink	true	0
xIOL_Write_Master	BOOL	Escribir en maestro Iolink		0
xIOL_Quit_Master	BOOL	Cerrar comunicación maestro		0
xIOL_Quit_Port1	BOOL	Cerrar comunicación puerto		0
wIOL_ProcessValue_Port1	WORD	Datos de proceso		0
arrIOL_Port1_ProcessValue	IO_Link_ARR_W_0_16	Datos de proceso		0
arrIOL_DataOut_Port1	IO_Link_UDT_ModulData	Datos de salida		0
xIOL_NewData_Port1	BOOL	Nuevo dato recibido		0
xIOL_Ready_Port1	BOOL	Puerto 1 listo		0
xIOL_Error_Port1	BOOL	Error puerto 1		0
wIOL_DiagCode_Port1	WORD	Codigo Error Puerto1		0
wIOL_DiagCode_Master	WORD	Codigo Error Maestro		0



IOLink_OutputData_Master	IO_Link_UDT_ModulData	Estructura datos		0
xIOL_Read_Port1	BOOL	Lectura de Puerto 1	1	0
xIOL_Write_Port1	BOOL	Escritura en Puerto 1		0
iIndex_Master	INT	Indice Maestro		0
iIndex_Port1	INT	Indice Puerto		0
iSubIndex_Port1	INT	Subindice puerto		0
iSubIndex_Master	INT	Subindice maestro		0
iLength_Master	INT	Longitud datos		0
iLength_Port1	INT	Longitud datos puerto		0
arrData_Master	IO_Link_ARR_W_0_32	Datos maestro		0
arrData_Port1	IO_Link_ARR_W_0_32	Datos puerto		0
Salida_IOLink	REAL	Datos leídos desde puerto 1		0
xIniciar_IOLink	BOOL	Inicia comunicación y lectura IOLINK		0

Ilustración 80 Variables ME_IOLink



3.3.3.6 ME_Mbus

ME_Mbus				
Nombre	Tipo	Descripción	Valor Inicial	PDD
EDCL_M_Bus_Scan_V1_00_1	EDCL_M_Bus_Scan_V1_00			0
EDCL_Generic_MBus_V1_02_1	EDCL_Generic_MBus_V1_02			0
arrMbusInput	EDCL_ARR_BYTE_0_31			0
arrMbusOutput	EDCL_ARR_BYTE_0_31			0
xLogging	BOOL		0	0
Mbus_Data	EDCL_UDT_MBUS_Data			0
Mbus_Devs	EDCL_ARR_MBUS_PARA			0
Mbus_Values	UDT_ARR_MB_Values			0
xScanned	BOOL			0
xExec	BOOL			0
Mbus_BaudRate	USINT		5	0
EDCL_MBusSvr_V2_00_3	EDCL_MBusSvr_V2_00			0

Ilustración 81 Variables ME_MBus



4. Descripción de Programa Visualización Web.

A continuación, usted puede encontrar una descripción de estructura y correspondencia entre variables entre el programa PLC y el programa realizado para la visualización WEB.

4.1 Login_MACRO.teq

Es la pantalla de bienvenida a la aplicación, y el paso previo al acceso a la página principal. En ella, se debe iniciar sesión introduciendo un nombre de usuario y su contraseña correspondiente. Se realiza la configuración de dos tipos de usuarios, con diferentes privilegios y restricciones, usuario “admin” y usuario “operario”.

Para iniciar sesión cómo administrador se debe introducir el nombre de usuario “admin”. Su contraseña se corresponde con “private”. Este usuario tiene acceso a toda la aplicación, sin restricción alguna. Puede acceder a todos los menús presentados y explicados en el manual de usuario, teniendo así todos los privilegios de la aplicación.

En caso de que el usuario sea un operario de la aplicación, debe iniciar sesión como “operario” cuya contraseña es “1234”. Este usuario tiene acceso a la aplicación con restricciones. No puede entrar en la pantalla correspondiente a Sistema “Settings_ILC”, dónde se pueden realizar cambios de las variables del controlador, y que, sin una autorización o formación previa, puede interferir negativamente en la configuración de la aplicación, pudiendo esta llegar a dejar de funcionar correctamente. Por ello, el usuario “operario” no tiene acceso a este menú de la aplicación, asegurándose así, el correcto funcionamiento de la misma.



Universidad de Oviedo

Trabajo Fin de Máster:

Aplicación para la gestión y adquisición de parámetros energético

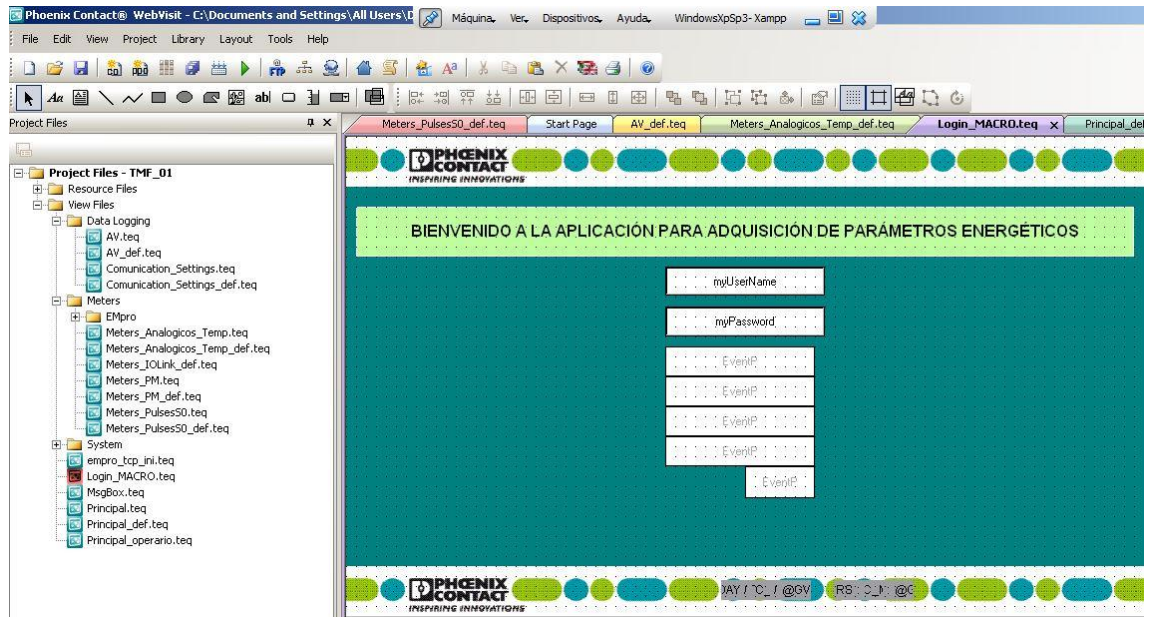


Ilustración 82 Login_Macro.teq



4.2 Principal_def.teq

Una vez iniciado sesión como “admin” el accede a esta pantalla, desde la cual tiene acceso al resto de la aplicación. Está compuesta por una serie de botones que permite el salto a las pantallas correspondientes a cada uno de ellos. Se diferencian tres partes:

- DISPOSITIVOS: Mediciones en Línea
- CONFIGURACIÓN Y AJUSTES DE ALMACENAMIENTO
- SISTEMA

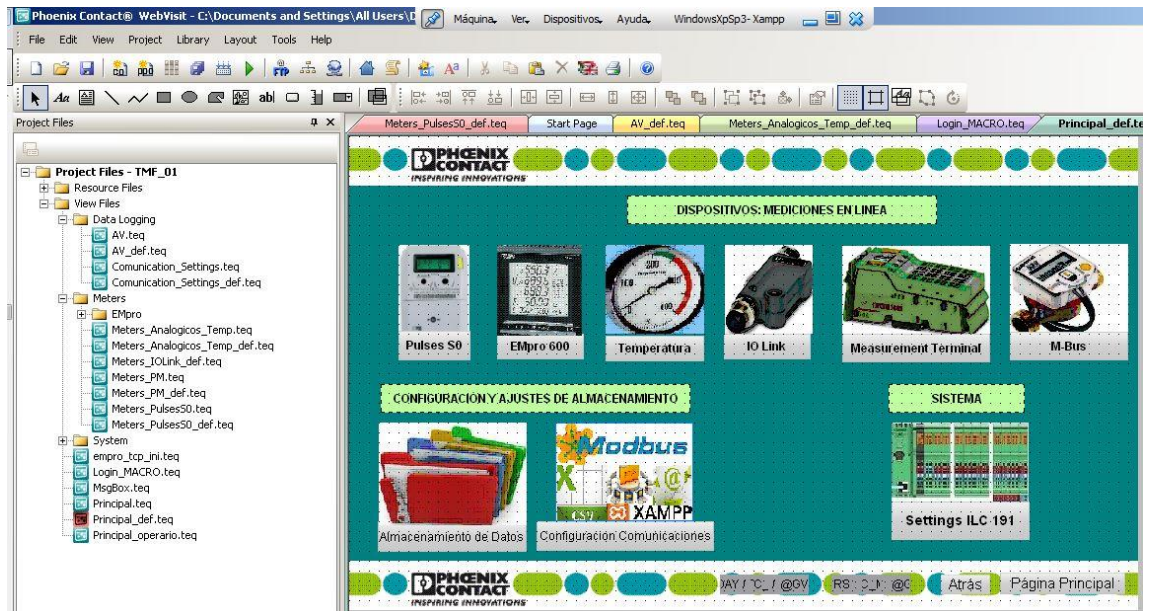


Ilustración 83 Principal_def.teq



4.3 Principal_operario.teq

Una vez iniciado sesión como “operario” el accede a esta pantalla, desde la cual tiene acceso al resto de la aplicación. Está compuesta por una serie de botones que permite el salto a las pantallas correspondientes a cada uno de ellos. Se diferencian dos partes:

- DISPOSITIVOS: Mediciones en Línea
- CONFIGURACIÓN Y AJUSTES DE ALMACENAMIENTO

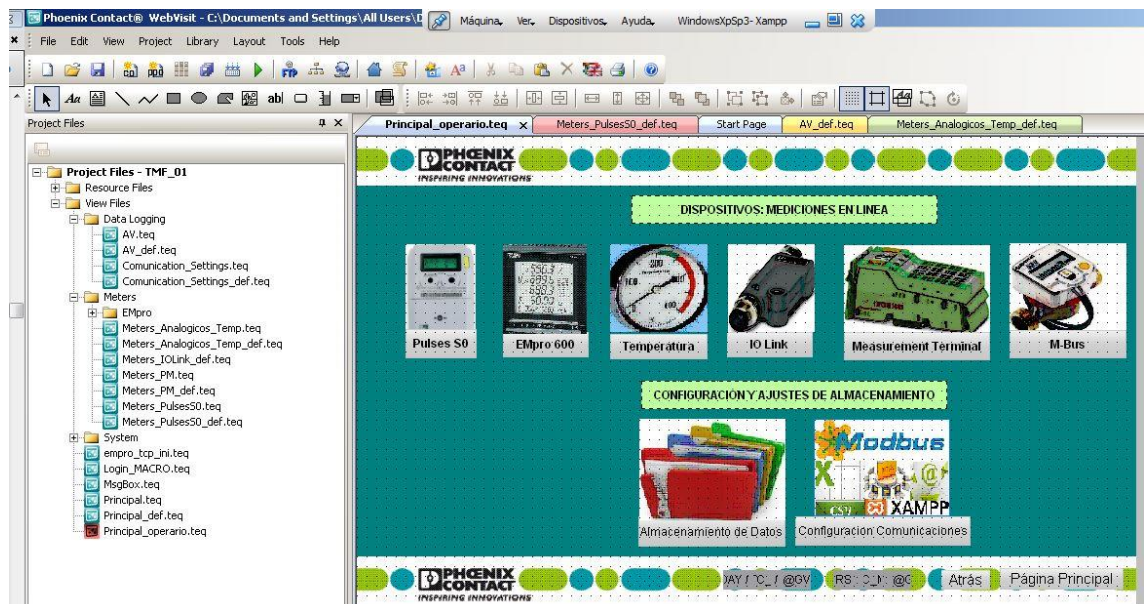


Ilustración 84 Principal_operario.teq



4.4 Sistema: Settings_ILC.teq

Pantalla sólo accesible por el usuario “admin”. Se desarrolla esta pantalla, con el fin de que el usuario pueda realizar el cambio de IP, Fecha y Hora del controlador. Para ello, se permite el acceso a las variables del programa Pcworx expuestas en la tabla.

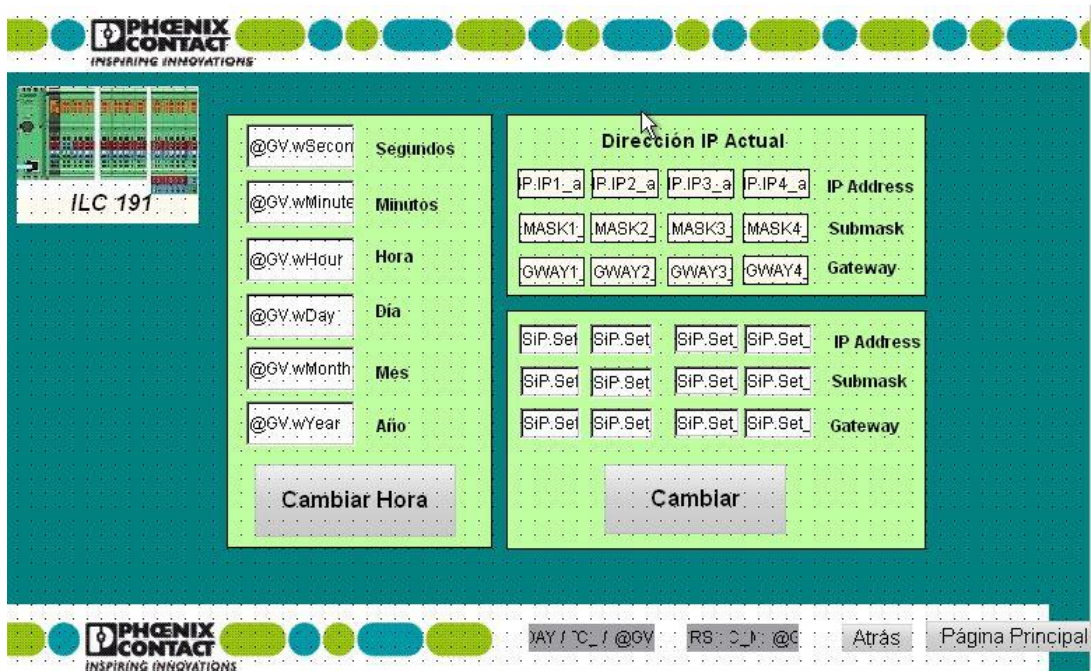


Ilustración 85 Settings_ILC.teq

Cambio de Hora:

Script PcWorx	Variable	Descripción Uso WebVisit
Cambiar Hora	wSecond	Parámetro segundos de la hora local
Cambiar Hora	wMinute	Parámetro minutos de la hora local
Cambiar Hora	wHour	Parámetro hora de la hora local
Cambiar Hora	wDay	Parámetro día de la fecha local
Cambiar Hora	wMonth	Parámetro meses de la fecha local
Cambiar Hora	wYear	Parámetro año de la fecha local
Cambiar Hora	XSTime_Enable	Activar cambio de hora

Ilustración 86 Tabla Variables PDD Cambio Hora



Cambio de IP:

Script PcWorx	Variable	Descripcion Uso WebVisit
Get_IP	IP1_actual, IP2_actual, IP3_actual, IP4_actual	Parámetros Dirección IP actual del PLC.
Get_IP	MASK1_actual, MASK1_actual, MASK1_actual, MASK1 actual.	Parámetros Correspondientes con la dirección de la máscara de red actual del PLC.
Get_IP	GWAY1_actual, GWAY2_actual, GWAY3_actual,	Parámetros Correspondientes con la dirección de la puerta de enlace actual del PLC
Set_IP	IP1, IP2,IP3,IP4	Parámetro Nueva Dirección IP
Set_IP	MASK1,MASK2, MASK3, MASK4	Parámetros Nueva Máscara de Red
Set_IP	GWAY1, GWAY2, GWAY3, GWAY4	Parámetros Nueva Puerta de Enlace
Cambiar_IP	xSIP_Enable	Parámetro hora de la hora local

Ilustración 87 Tabla variables PDD Cambio IP



4.5 Configuración_Data_Comu

Se realizan dos pantallas para la configuración de comunicaciones y almacenamiento.

4.5.1 AV_def.teq

Se programa esta pantalla para que el usuario pueda seleccionar que datos medidos quiere almacenar. Para ello se relaciona cada uno de los campos con las variables del programa PLC tal que:

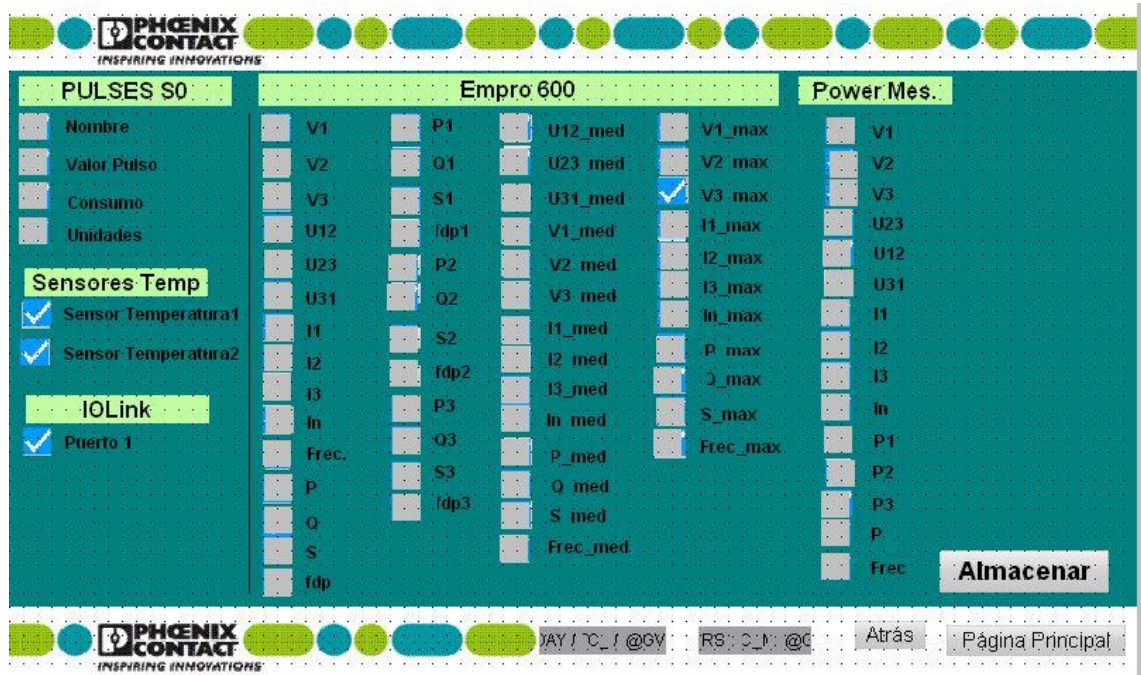


Ilustración 88 AV_def.teq



Script PcWorx	Variable	Descripcion Uso
ArchiveValues_PulsesS0	xNombre, xValorPulso, xConsumo, xUnidades	Habilita almacenamiento de los parámetros correspondientes al PulsesS0
AV_AnalogTemp	xTemp01, xTemp02	Habilita almacenamiento de las temperaturas leídas por canal 1 y/o 2.
AV_IOLink	xIOLink	Habilita almacenamiento de los parámetros leídos a través de IOLink
ArchiveValues_Empro	xU12, xU23, xU31, xV1, xV2, xV3, xfrec, xI1, xI2, xI3, xIn, , xPt, xQt, xSt, , xP1, xQ1, xS1, xP2, xQ2, xS2, xP3, xQ3, xS3, xfdp, xfdp1, xfdp2, xfdp3, xU12_med, xU23_med, xU31_med, xV1_med, xV2_med, xV3_med, xfrec_med, xI1_med, xI2_med, xI3_med, xIn_med, , xPt_med, xQt_med, xSt_med, xV1_max, xV2_max, xV3_max, xfrec_max, xI1_max, xI2_max, xI3_max, xIn_med, , xPt_max, xQt_max, xSt_med , ,	Habilita almacenamiento de los parámetros correspondientes al analizador de Redes Empro600



AV_PM	xPM_U12, Xpm_U23, Xpm_U31, xpm_V1, xpm_V2, xpm_V3, xpm_frec, xpm_I1, xpm_I2, xpm_I3, xpm_In, xpm_P1, xpm_P2, xpm_P3,	Habilita almacenamiento de los parámetros correspondientes al módulo de medición de potencia PM
AV_MBus	xMbus_horasfuncionamiento, xConsumoporHora, xVolumen, xTempIN, xTempOUT	Habilita almacenamiento de parámetros leídos por Mbus
FTP_Settings	xAlmacenar	Habilita el envío de datos al datalogger

Ilustración 89 Tabla Variables PDD AV

4.5.2 *Communication_Settings_def.teq*

Se programa esta pantalla para que el usuario pueda configurar el medio de almacenamiento de los datos seleccionados en AV_def.teq. De manera que, las variables a las que usuario tendría acceso mediante esta pantalla serían:

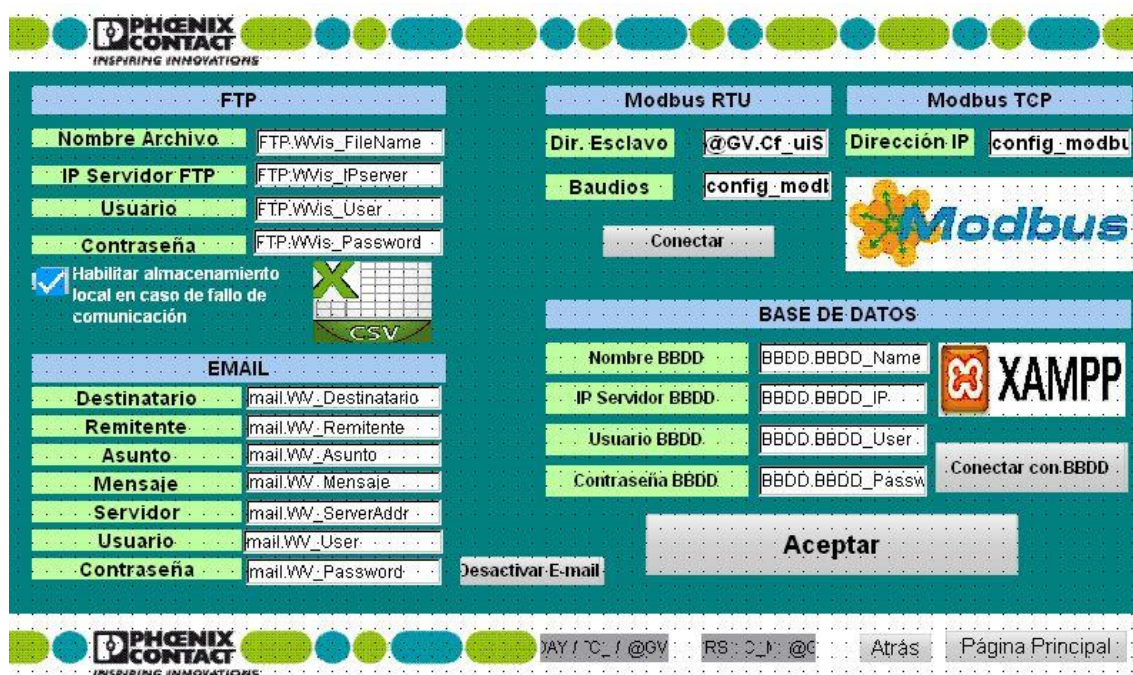


Ilustración 90 *Communication_Settings_def.teq*

Script PcWorx	Variable	Descripcion Uso WebVisit
FTP_Settings	WVis_FileName	Nombre del fichero
FTP_Settings	WVis_IPserver	Dirección IP del servidor FTP
FTP_Settings	WVis_User	Nombre del usuario con acceso al servidor FTP
FTP_Settings	WVis_Password	Contraseña correspondiente al usuario del servidor FTP



Email	WV_Destinario	Dirección de email del destinatario
Email	WV_Remitente	Dirección de email del remitente
Email	WV_Asunto	Asunto del email a enviar
Email	WV_Mensaje	Mensaje del email a enviar
Email	WV_ServerAddr	Dirección del servidor de correo electrónico
Email	WV_User	Nombre de usuario del remitente para acceso al servidor de correo
Email	WV_Password	Contraseña de usuario del remitente para acceso al servidor de correo
Settings_Modbus	Cf_uiSlave_Addr_RTU	Número de esclavo de dispositivo Modbus RTU
Settings_Modbus	Cf_diBaudrate	Velocidad de transmisión de RTU
Settings_Modbus	Cf_IP_TCP	Dirección IP del dispositivo Modbus TCP
Modbus_TCP, Modbus_RTU	xActivate_Client	Botón CONECTAR. Habilita la conexión por RTU o TCP
BBDD_Write	BBDD_Name	Nombre de la base de datos
BBDD_Write	BBDD_IP	Dirección IP donde se aloja la BBDD
BBDD_Write	BBDD_User	Nombre de usuario para acceso a BBDD
BBDD_Write	BBDD_Password	Contraseña de usuario para acceso a BBDD
BBDD_Write	BBDD_Habilitar	Botón "Conectar con BBDD". Habilita la comunicación con la base de datos
		Botón Aceptar. Salta a menú principal

Ilustración 91 Tabla Variables PDD Communication Settings



4.6 Dispositivos

4.6.1 Empro

Conjunto de pantallas programas para la visualización online de los valores medidos por los analizadores de redes.

4.6.1.1 Empro600_Seleccion_def.teq

Pantalla programada para que el usuario seleccione el protocolo de comunicación mediante el cuál se encuentran conectados los analizadores de redes. Para ello se realiza la siguiente relación entre recursos de la pantalla y variables de programa PLC.

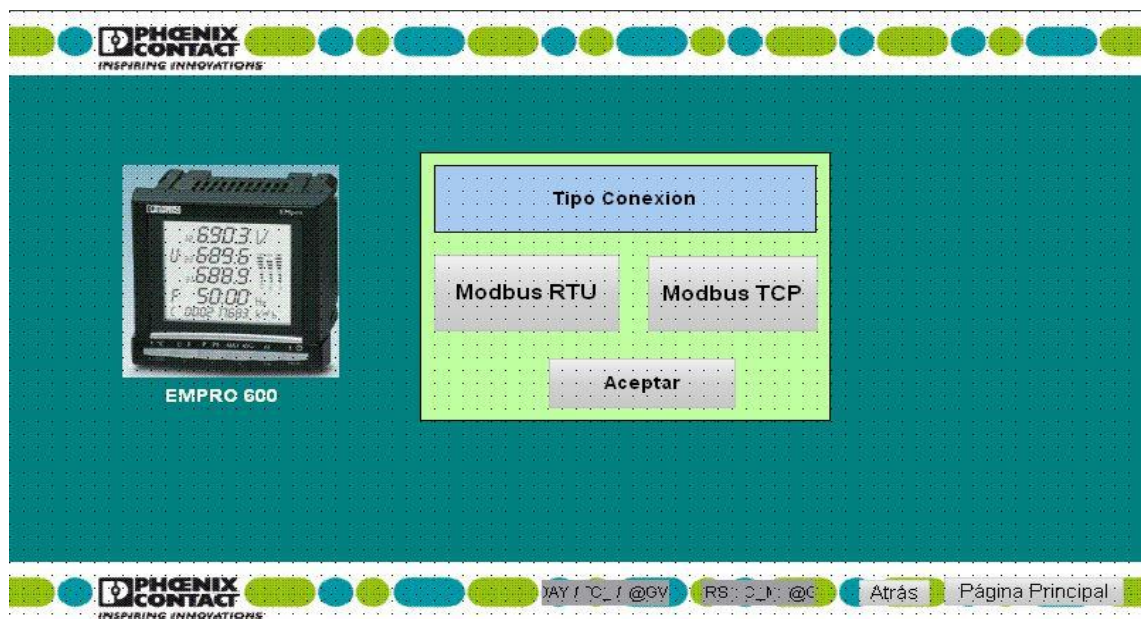


Ilustración 92 Empro600_Seleccion_def.teq



Script PcWorx	Variable	Descripcion Uso WebVisit
Modbus_RTU	xActivate_Master_RTU	Habilita Maestro RTU
Modbus_TCP	xClient_TCP	Habilita Cliente TCP
		Botón ACEPTAR, Salta a Empro600_def.teq

Ilustración 93 Tabla Variables PDD Modbus

4.6.1.2 Empro600_def.teq

Pantalla programada para la visualización online de los valores básicos medidos por los analizadores de redes. Se realiza la visualización de las siguientes variables del programa PLC



Ilustración 94 Empro600_def.teq



Script PcWorx	Variable	Descripcion Uso
Values_Online_Advan	xU12_tcp, xU23_tcp, xU31_tcp, xV1_tcp, xV2_tcp, xV3_tcp, xfrec_tcp, xI1_tcp, xI2_tcp, xI3_tcp, xIn_tcp, , xPt_tcp, xQt_tcp, xSt_tcp, , xP1_tcp, xQ1_tcp, xS1_tcp, xP2_tcp, xQ2_tcp, xS2_tcp, xP3_tcp, xQ3_tcp, xS3_tcp, xfdp_tcp, xfdp1_tcp, xfdp2_tcp, xfdp3_tcp	Visualiza los parámetros básicos leídos del analizador de redes Empro 600 mediante Modbus TCP
Values_RTU	xU12_rtu, xU23_rtu, xU31_rtu, xV1_rtu, xV2_rtu, xV3_rtu, xfrec_rtu, xI1_rtu, xI2_rtu, xI3_rtu, xIn_rtu, xPt_rtu, xQ_rtu, xSt_rtu, xP1_rtu, xQ1_rtu, xS1_rtu, xP2_rtu, xQ2_rtu, xS2_rtu, xP3_rtu, xQ3_rtu, xS3_rtu, xfdp_rtu, xfdp1_rtu, xfdp2_rtu, xfdp3_rtu	Visualiza los parámetros básicos leídos del analizador de redes Empro 600 mediante Modbus RTU
		Botón Valores Avanzados, Salta a ValoresAvanzados_def.teq

Ilustración 95 Tabla Variables PDD Empro_def



4.6.1.3 ValoresAvanzados_def.teq

Pantalla programada para la visualización online de los valores avanzados medidos por los analizadores de redes. Se realiza la visualización de las siguientes variables del programa PLC

The screenshot shows the EMPRO 600 interface with the following data:

Conectado con TCP.IP_TCP **Conectado con esclavo nº dr_**

Valores Medios

Tension	Potencia	Intensidad	Frecuencia
U12 @GV.L V	P @GV.P KW	I1 @GV.I1 A	@GV.fr Hz
U23 @GV.L V	Q @GV.Q KVAR	I2 @GV.I2 A	
U31 @GV.L V	S @GV.S KVA	I3 @GV.I3 A	
V1 @GV.V V		In @GV.Ir A	
V2 @GV.V V			
V3 @GV.V V			

Valores Máximos

Tension	Potencia	Intensidad	Frecuencia
V1 @GV.V V	P @GV.P KW	I1 @GV.I1 A	@GV.fr Hz
V2 @GV.V V	Q @GV.Q KVAR	I2 @GV.I2 A	
V3 @GV.V V	S @GV.S KVA	I3 @GV.I3 A	
		In @GV.Ir A	

At the bottom of the interface, there are navigation buttons: "Atrás" and "Página Principal".

Ilustración 96 ValoresAvanzados_def.teq



Script PcWorx	Variable	Descripcion Uso WebVisit
Values_Online_Advan	xU12_med_tcp, xU23_med_tcp, xU31_med_tcp, xV1_med_tcp, xV2_med_tcp, xV3_med_tcp, xfrec_med_tcp, xI1_med_tcp, xI2_med_tcp, xI3_med_tcp, xIn_med_tcp, , xPt_med_tcp, xQt_med_tcp, xSt_med_tcp, xV1_max_tcp, xV2_max_tcp, xV3_max_tcp, xfrec_max_tcp, xI1_max_tcp, xI2_max_tcp, xI3_max_tcp, xIn_max_tcp, , xPt_tcp, xQt_max_tcp, xSt_max_tcp,	Visualiza los parámetros avanzados (máximos y medios) leídos del analizador de redes Empro 600 mediante Modbus TCP
Values_RTU	xU12_med_rtu, xU23_med_rtu, xU31_med_rtu, xV1_med_rtu, xV2_med_rtu, xV3_med_rtu, xfrec_med_rtu, xI1_med_rtu, xI2_med_rtu, xI3_med_rtu, xIn_med_rtu, xPt_med_rtu, xQt_med_rtu, xSt_med_rtu, xV1_max_rtu, xV2_max_rtu, xV3_max_rtu, xfrec_max_rtu, xI1_max_rtu, xI2_max_rtu, xI3_max_rtu, xIn_max_rtu, xPt_max_rtu, xQt_max_rtu, xSt_max_rtu,	Visualiza los parámetros básicos leídos del analizador de redes Empro 600 mediante Modbus RTU



Ilustración 97 Tabla Variables PDD Valores Avanzados

4.6.2 *Meters_Analogicos_Temp_def.teq*

Pantalla programada para la visualización online de los parámetros medidos por los sensores de temperatura. La variable del programa PLC que se visualiza en esta pantalla se corresponde con:

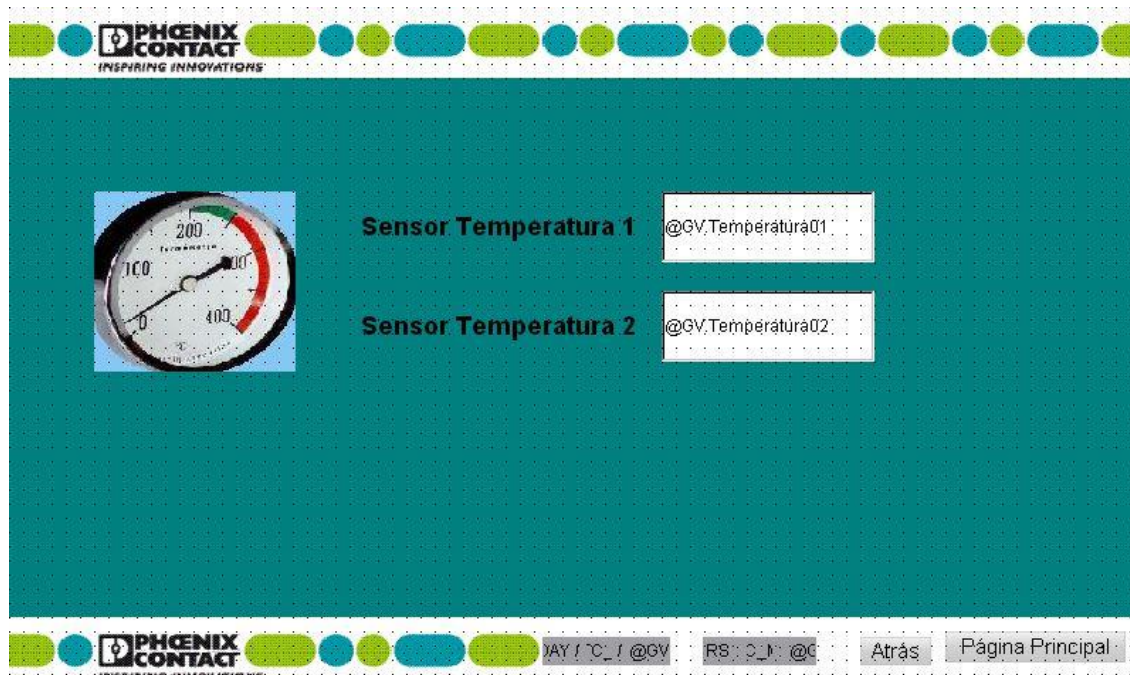


Ilustración 98 Meters_Analogicos_Temp_def.teq

Script PcWorx	Variable	Descripcion Uso WebVisit
ME_AnalogTemp	Temperatura01	Valor Temperatura leída del canal1
ME_AnalogTemp	Temperatura02	Valor Tempearatura leída del canal 2

Ilustración 99 Tabla Variables PDD ME_Analog

4.6.3 Meters_PM_def.teq

Pantalla programada para la visualización online de los parámetros medidos por el módulo de medida de energía eléctrica del controlador. Las variables del programa PLC relacionadas con esta pantalla son:

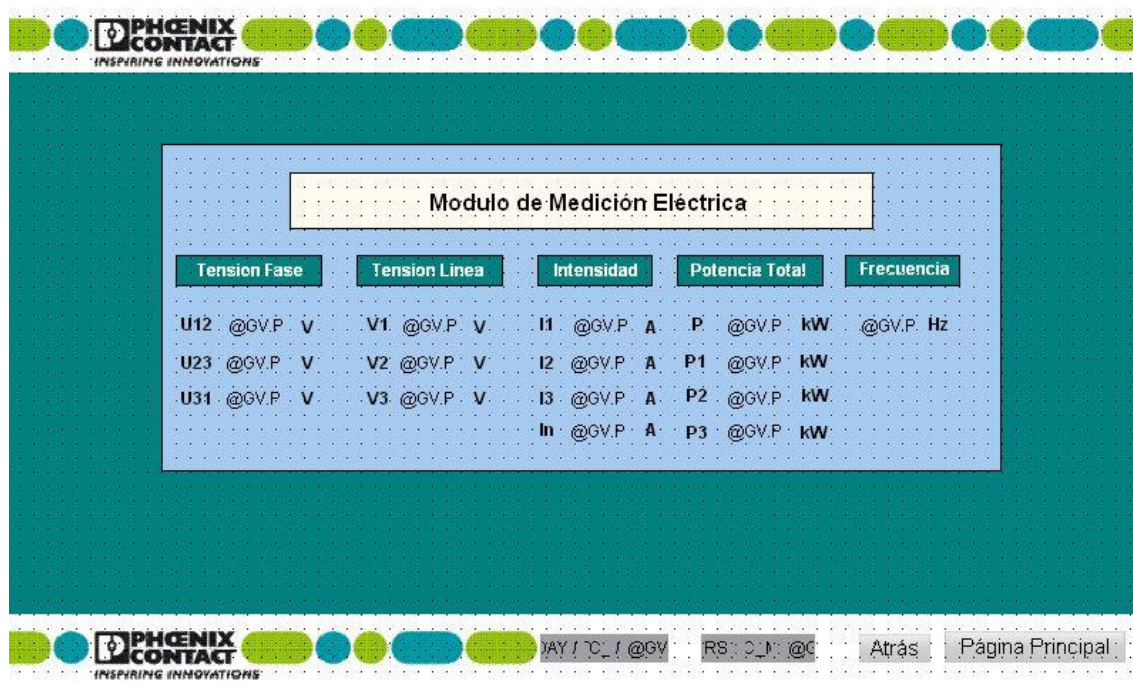


Ilustración 100 Meters_PM_def.teq

Script PcWorx	Variable	Descripcion Uso WebVisit
ME_PM	PM_U12, pm_U23, pm_U31, pm_V1, pm_V2, xpm_V3, pm_frec, pm_I1, xpm_I2, pm_I3, pm_In, xpm_P1, pm_P2, pm_P3,	Valores medidos desde el terminal de medición eléctrica

Ilustración 101 Tabla Variables PDD Meters_PM

4.6.4 Meters_PulsesS0_def.teq

Pantalla programada para la visualización online de los parámetros medidos por los dispositivos PulsesS0. Las variables del programa PLC que se intervienen en esta pantalla se corresponden con:

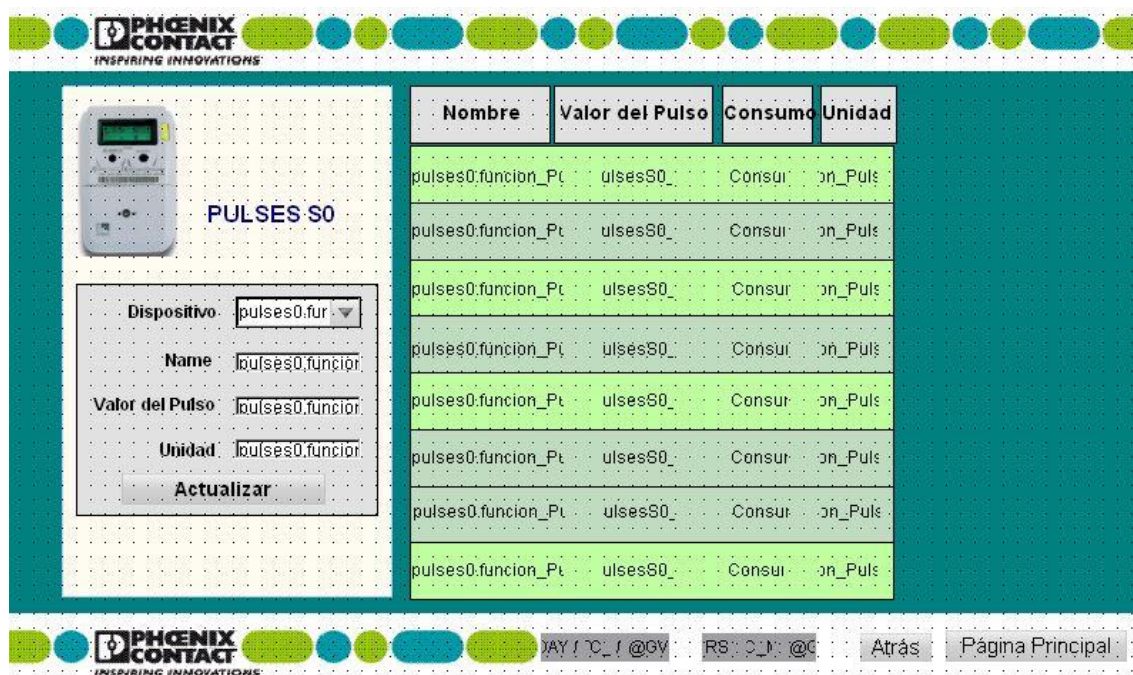


Ilustración 102 Meters_PulsesS0_def.teq

Script PcWorx	Variable	Descripcion Uso WebVisit
Función_pulsesS0	Ndevice	Numero de dispositivo PulsesS0
Función_pulsesS0	NameDevice	Nombre dispositivo PulsesS0
Función_pulsesS0	ValuePulsesDevice	Valor del pulso del dispositivo PulsesS0
Función_pulsesS0	UnitDevice	Unidad del pulso del dispositivo Pulses S0
Función_pulsesS0	Actualizar	Carga valores introducidos por usuario a las variables



Función_pulsesS0	Name0, Name1, Name2, Name3, Name4, Name5, Name6, Name7	Nombre actual de cada dispositivo PulsesS0
Función_pulsesS0	Value_of_pulse0, Value_of_pulse1, Value_of_pulse2, Value_of_pulse3, Value_of_pulse4, Value_of_pulse5, Value_of_pulse6, Value_of_pulse7,	Valor actual del pulso de cada dispositivo PulsesS0
Función_pulsesS0	Consumo_0, Consumo_1, Consumo_2, Consumo_3, Consumo_4, Consumo_5, Consumo_6, Consumo_7,	Valor actual del consumo total de cada dispositivo PulsesS0
Función_pulsesS0	Unit0, unit1, Unit2, unit3, Unit4, unit5, Unit6, unit7,	Unidad actual del consume de cada dispositivo PulsesS0

Ilustración 103 Tabla Variables PDD ME_PulsesS0



4.6.5 Meters_IOLink_Temp_def.teq

Pantalla programada para la visualización online de los parámetros medidos por los dispositivos IOLink. Las variables del programa PLC que se intervienen en esta pantalla se corresponden con:



Ilustración 104 Meters_IOLink_Temp_def.teq

Script PcWorx	Variable	Descripcion Uso WebVisit
ME_IOLink	Salida_IOLink	Valor Leído del puerto 1 IOLink

Ilustración 105 Tabla Variables Meters_IOLink_Temp_def



4.6.6 ME_Mbus.teq

Pantalla programada para la visualización online de los parámetros medidos por los dispositivos Mbus. Las variables del programa PLC que se intervienen en esta pantalla se corresponden con:

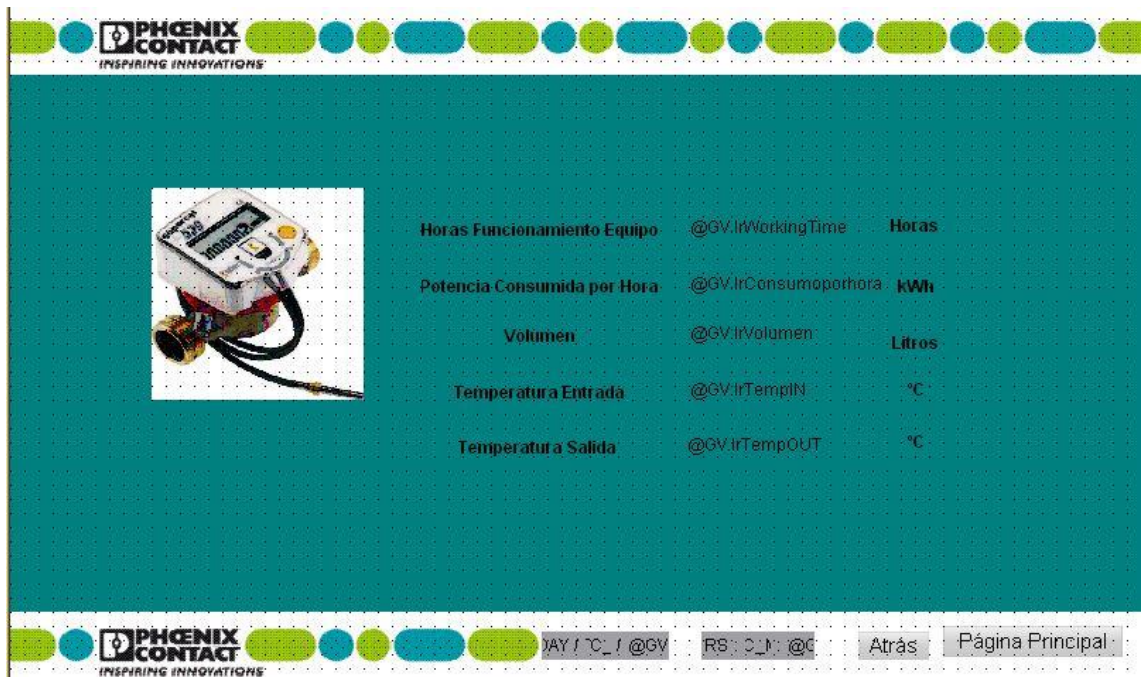


Ilustración 106 ME_MBUS.teq

Script PcWorx	Variable	Descripcion Uso WebVisit
ME_Mbus	IrWorkingTime, IrConsumoporhora, IrVolumen, IrTempIN, IrTempOUT	Valor Leído del puerto 1 IOLink

Ilustración 107 Tabla Variables ME_Mbus

5. Descarga y Preparación de la Aplicación para su Uso.

En caso de haber realizado alguna modificación tanto en el programa del PLC como en la visualización web, es necesario proceder a su descarga en el controlador. Por ello, en los apartados 5.1 y 5.2 se explican los pasos que han de seguirse en caso de darse alguno de los casos anteriores. Además, realizar una configuración previa del servidor FTP y de las bases de datos es necesario para un correcto funcionamiento de la aplicación.

5.1 Descarga de Programa en el Controlador ILC 191 ME/AN

Si se han realizado cambios en la programación debe volver a instalar el programa de PCWorx en el controlador. Para ello, debe seguir los siguientes pasos:

- 1) En el menú **Crear>>Recompilar Proyecto**. El proyecto se compilará. Una vez terminado este proceso, si todo está correcto, en la ventana de mensajes del software aparecerá el siguiente mensaje: 0 errores

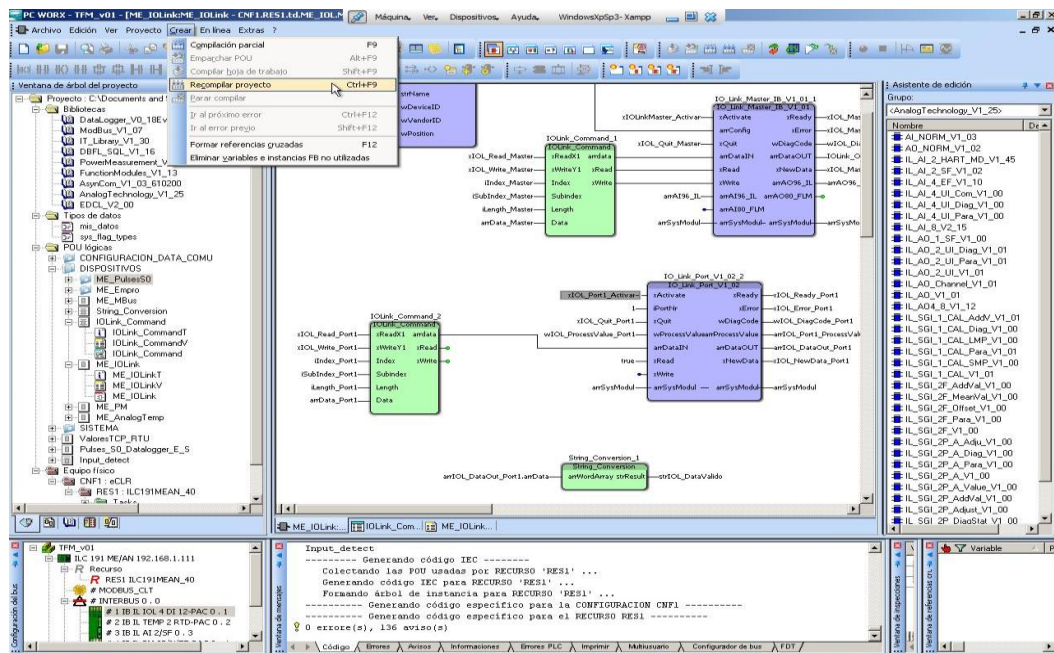


Ilustración 108 Compilación proyecto

- 2) Entrando el menú **En Línea>>Control de proyecto...** aparece una ventana con el nombre del recurso del programa, dónde usted deberá pulsar la opción **Descargar**. El proceso de descarga comenzará.

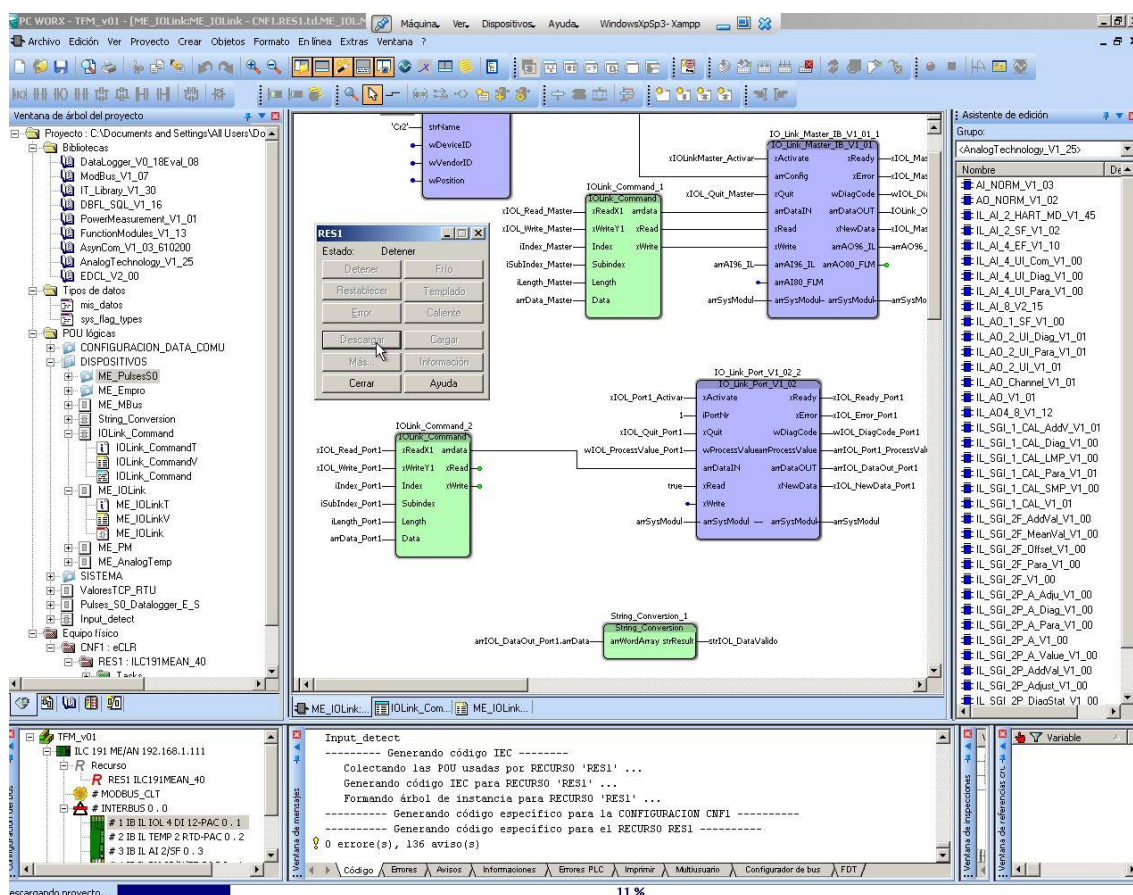


Ilustración 109 Descarga Proyecto



- 3) Una vez se haya realizado la descarga del proyecto correctamente, se debe poner el modo RUN el controlador. Para ello, pulse Caliente, en la ventana de Control de Proyecto.



Ilustración 110 PLC Modo RUN



5.2 Descarga de Visualización Web en el Controlador ILC 191 ME/AN

Si usted ha realizado algún cambio en el programa WebVisit para la visualización web de la aplicación, es necesario que descargue estos cambios al controlador. Para ello, siga las siguientes instrucciones:

- 1) En la barra de herramientas, pulse el **botón de descarga de proyecto** correspondiente con un icono FTP

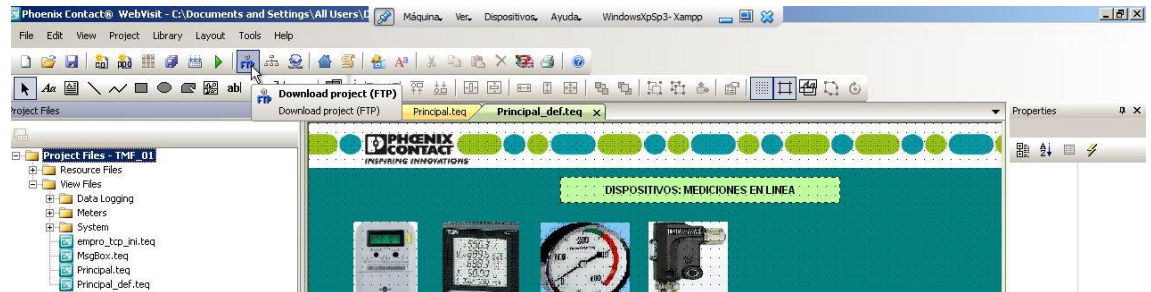


Ilustración 111 Entorno WVisit

- 2) Aparece una ventana preguntando si se desea compilar el proyecto antes de realizar la descarga. Pulsar **Sí**.



Ilustración 112 Compilacion WebVisit

- 3) Aparece ventana, en la que se debe introducir la **dirección IP del controlador**, y pulsar **Connect**.

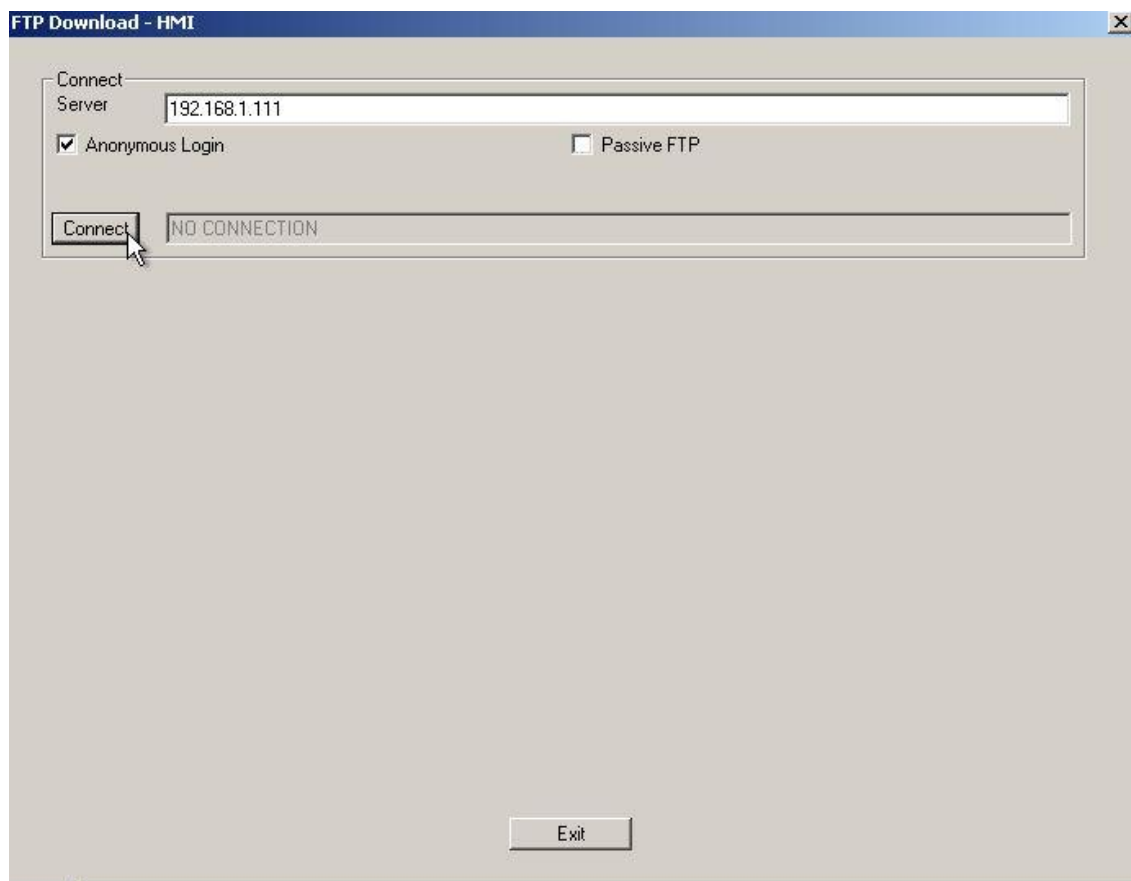


Ilustración 113 Conexión WebVisit

- 4) Una vez se realice la conexión del PC con con el controlador, pulse **Download Project**. La descarga comenzará y tardará unos minutos.

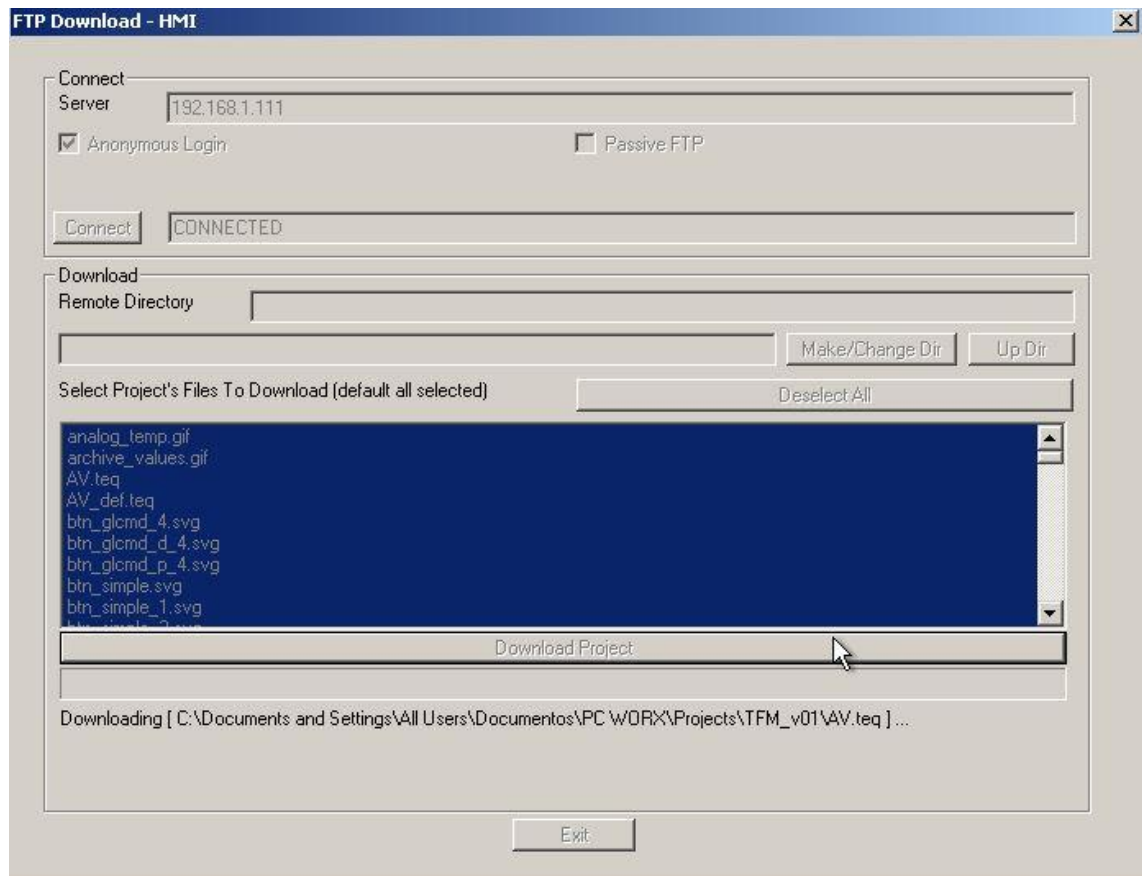


Ilustración 114 Descargar programa WebVisit

- 5) La descarga se ha realizado correctamente, cuando aparezca por pantalla el siguiente mensaje.

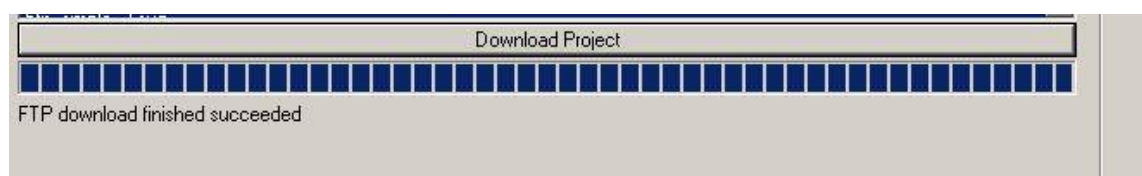


Ilustración 115 Descarga WebVisti correcta

5.3 Configuración del Servidor FTP para la Transmisión y Almacenamiento de Archivos.

El servidor utilizado por la aplicación es Filezilla, por ello, es necesaria la instalación del servidor FTP Filezilla en el PC. Se puede realizar la descarga en: <https://filezilla-project.org/>.

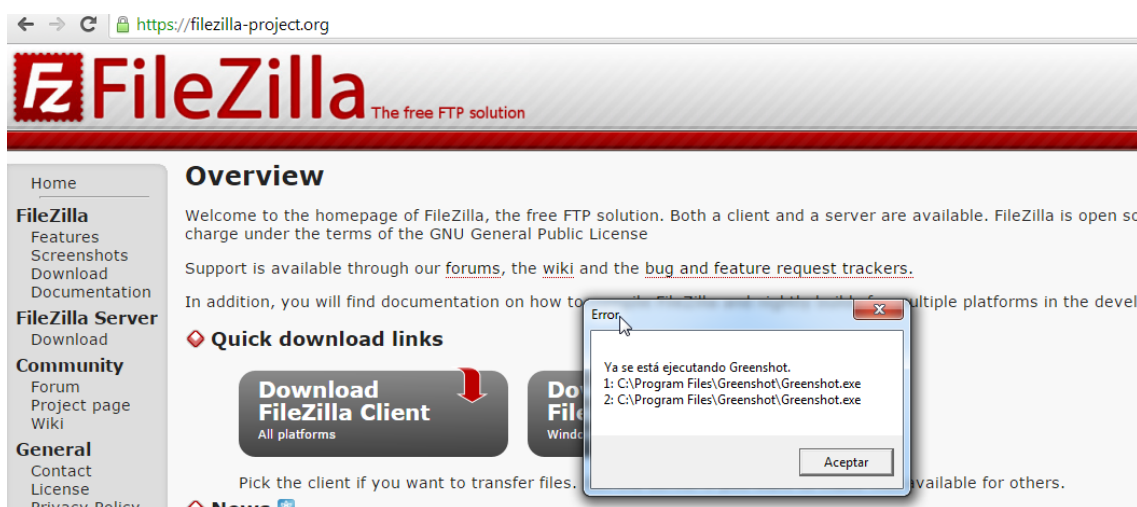


Ilustración 116 FileZilla

Una vez realizada la descarga e instalación, se procede a su configuración. Aparecerá la siguiente pantalla, dónde será necesario introducir la dirección ip del servidor, cómo se desea que el servidor esté alojado en el propio PC, la dirección del mismo será: 127.0.0.1



Ilustración 117 Dirección IP servidor FTP



Se establecerá conexión con el servidor FTP tal y cómo viene configurado por defecto.

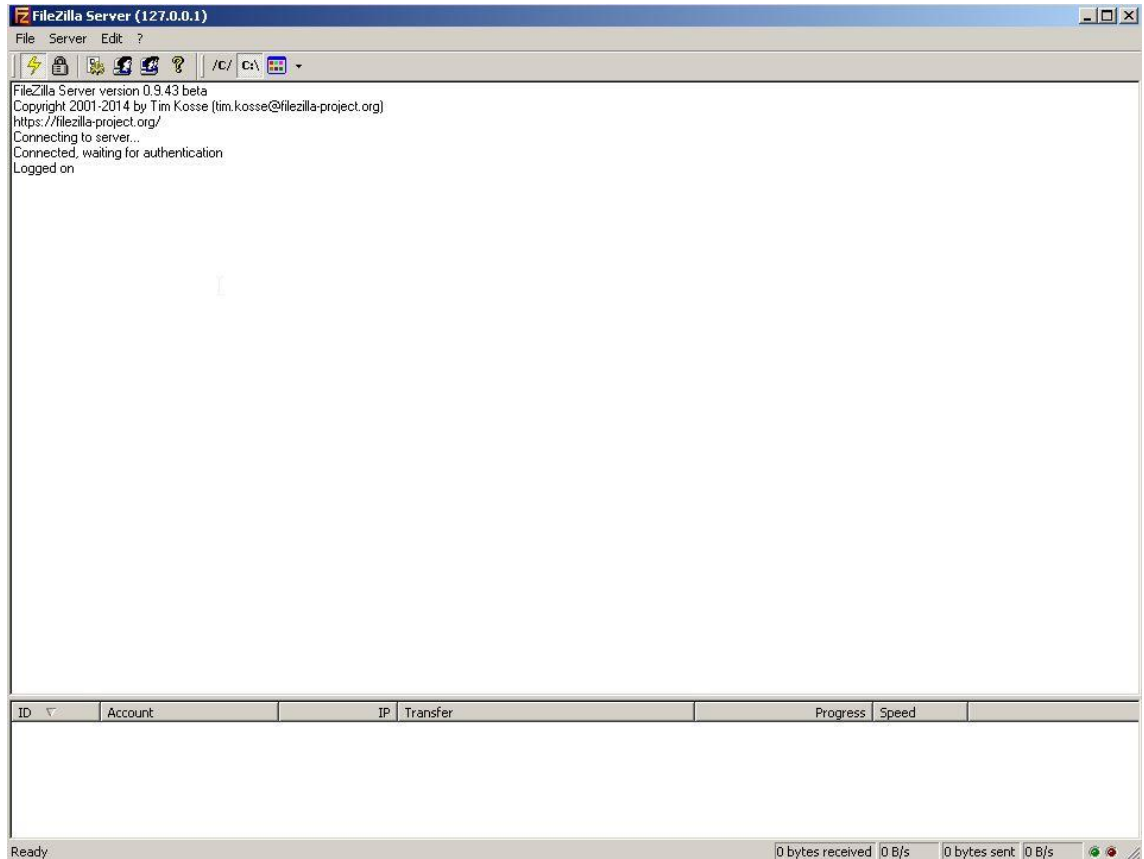


Ilustración 118 Conexión Servidor FTP Correcta

Se requiere un ajuste de configuración del puerto de comunicación que usará el servidor cómo de los parámetros de la dirección IP externa del mismo. Para ello:

- Entrar en el menú **Edit>>Settings>>Passive Mode Settings**
- Habilitar opción: **Use custom port range**. Escribir en la primera casilla el Puerto que se desea utilizar. En este caso, se configura el puerto 21.
- En el menú IPv4specific, habilitar opción: **Use the following IP**. Escribir la dirección IP deseada para el servidor FTP. En este caso, será la misma que el PC.

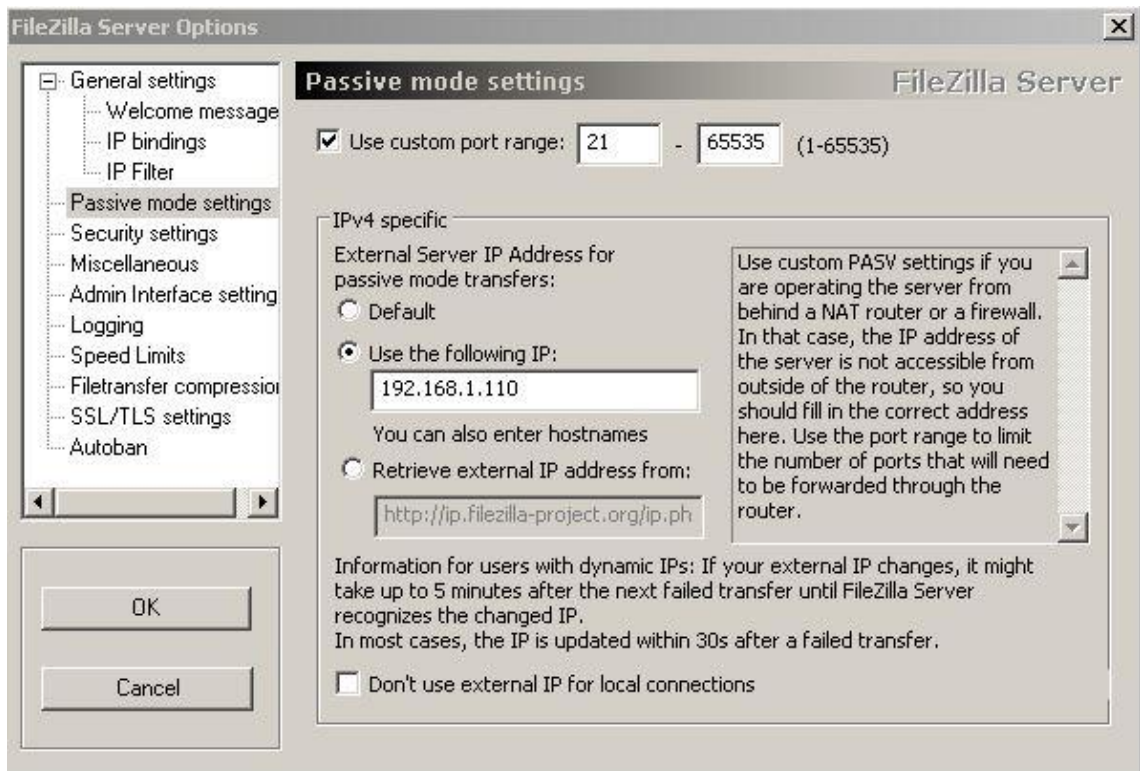


Ilustración 119 Configuración Servidor FTP

Por último, se requiere la configuración de usuarios con acceso al servidor y la ubicación de los archivos alojados en el servidor. Para ello, se deben seguir los siguientes pasos:



CONFIGURACIÓN Y CREACIÓN DE USUARIOS:

- Entrar en el menú **Edit>>Users>>General**. Añadir y configurar un nuevo usuario. En este caso se utilizará el nombre de usuario **“anonymous”** sin contraseña.

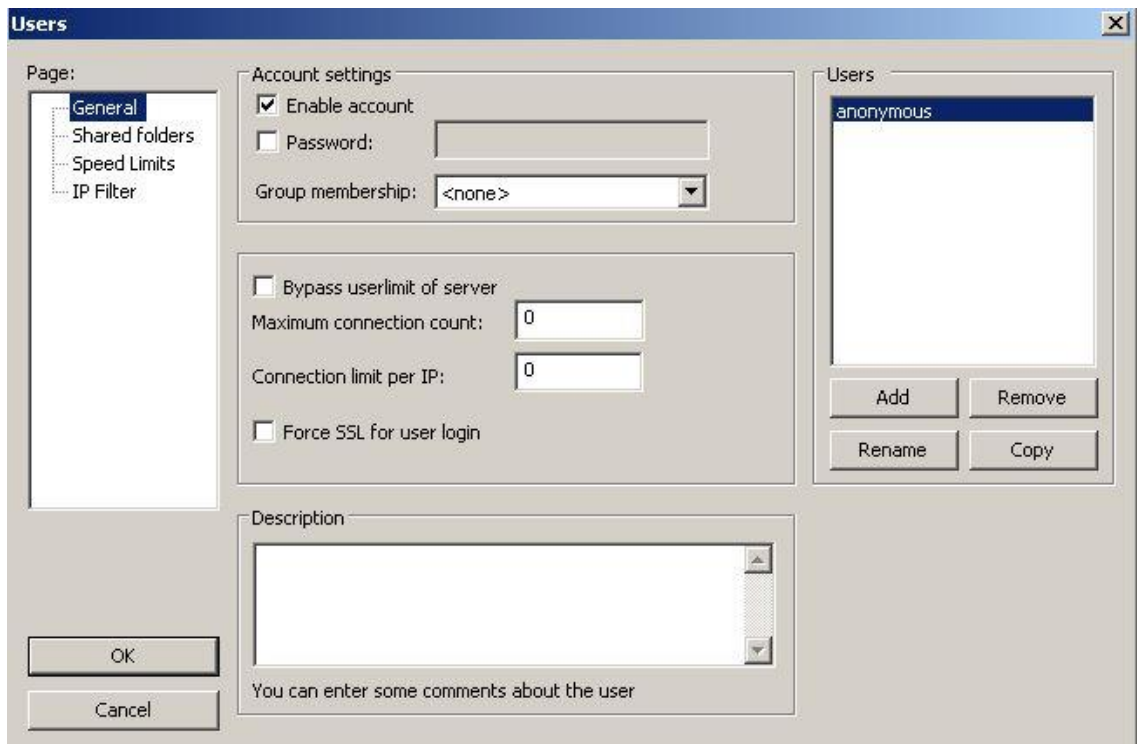


Ilustración 120 Creación Usuario FTP



CONFIGURACIÓN RUTA DE ALMACENAMIENTO DEL SERVIDOR:

- Entrar en el menú **Edit>>Users>>Shared Folders**. Añadir y configurar una nueva ruta dónde se almacenarán los archivos del servidor. Ajustar que usuarios tienen acceso a que carpeta también es una opción en este menú.

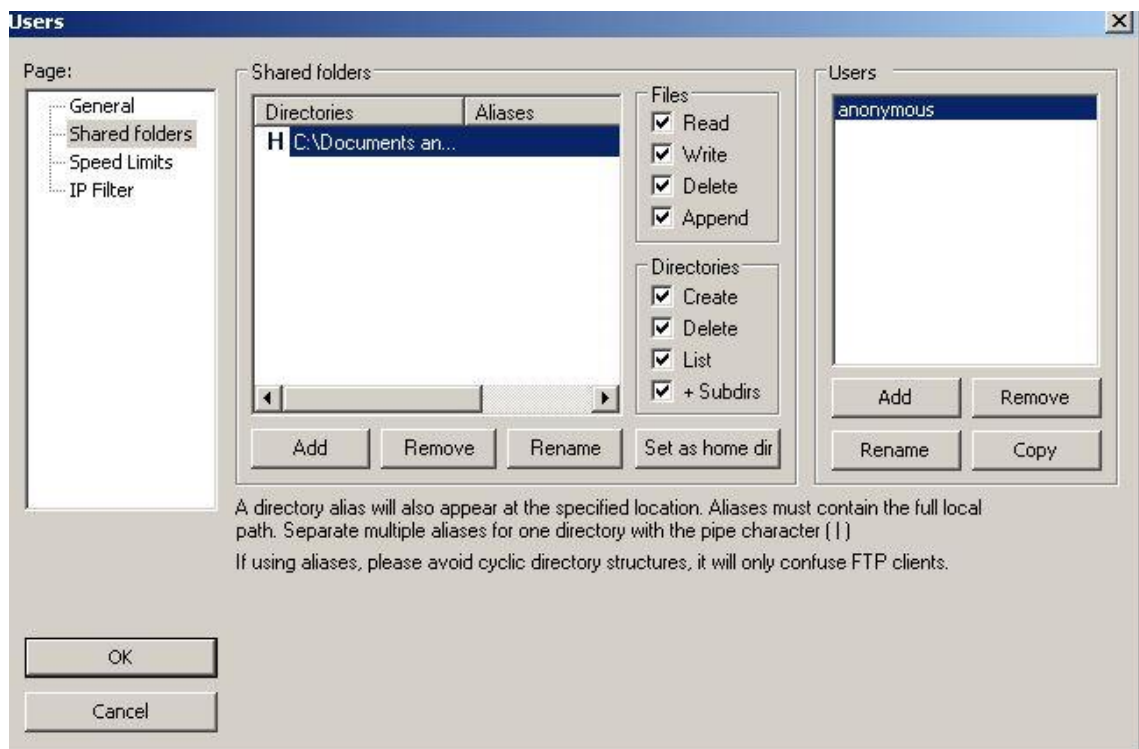


Ilustración 121 Configuración Ruta Almacenamiento FTP



5.4 Creación y Configuración de Bases de Datos en Servidor XAMPP para el Almacenamiento de Datos.

Se requiere de un servidor web a través del cuál la interacción del usuario con la base de datos sea sencilla. Para ello, se debe descargar e instalar XAMPP, un conjunto de aplicaciones software que contiene un servidor web tipo apache y un servidor MySQL, entre otros. Se puede realizar la descarga en: <https://www.apachefriends.org/es/index.html>

Una vez instalado XAMPP, se debe poner en Run el servidor web (Apache) y servidor MySQL.

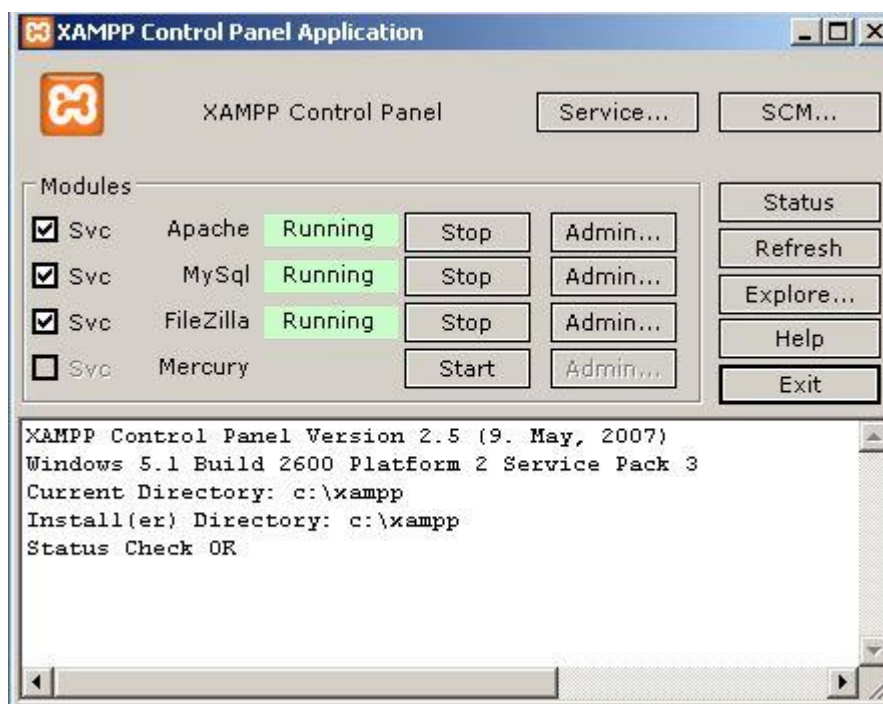


Ilustración 122 Apache Modo RUN

Una vez estos servidores estén en modo Running, se procede a la creación de una base de datos en MySQL, mediante el servidor web. Se han de seguir las siguientes instrucciones:



- 1) **Abrir explorador de internet** de su PC e introducir la url: **127.0.0.1:8080**
 . En la instalación se configura el puerto de comunicación 8080 para el servidor web. El puerto usado por defecto (80) está ocupado por el servidor web de WebVisit.

XAMPP for Windows

English / Deutsch / Français / Nederlands / Polski / Italiano / Norwegian / Español / Português (Brasil) / 中国

XAMPP 1.7.4
[PHP: 5.3.9]

Bienvenido a XAMPP para Windows!

Felicidades: XAMPP se instaló con éxito en su ordenador!

Ahora se puede empezar a trabajar. :) Primero por favor pulse encima de »Estado« en la parte izquierda. De esta manera tendrá una visión de que es lo que funciona ya. Algunas funciones estarán desactivadas. Es intencionado. Son funciones, que no funcionan en todas partes o eventualmente podrían ocasionar problemas.

Atención: XAMPP fue modificado a partir de la versión 1.4.x a una administración de paquete único. Existen los siguientes paquetes/Addons:

- XAMPP paquete básico
- XAMPP Perl addon
- XAMPP Tomcat addon
- XAMPP Cocoon addon
- XAMPP Python addon (developer version)

Y en un futuro:

- XAMPP Utility addon (Accesorio pero aún inactivo)
- XAMPP Server addon (otros servidores aún inactivos)
- XAMPP Other addon (otras cosas útiles aún inactivas)

Por favor "instalar" los paquetes adicionales, que aún necesite, simplemente a continuación. Después de subirlos con éxito, por favor siempre accionar "setup_xampp.bat", para inicializar nuevamente XAMPP. A bueno, las versiones Instalador de los Addons individuales funcionan solo si el paquete básico XAMPP también fue montado a partir de una versión instalador.

Para el soporte OpenSSL utilice por favor el certificado de chequeo con la URL <https://127.0.0.1> ó <https://localhost>

Os deseamos mucha diversión, Kay Vogelgesang + Kai 'Oswald' Seidler

Ilustración 123 Conexión Apache

- 2) Entrar en el menú **Herramientas>>phpMyAdmin>>MySQL local host**.
 En el apartado Crear nueva base de datos, introducir nombre de la nueva BBDD. Pulsar **Crear**

phpMyAdmin localhost

Bases de datos SQL Estado actual Variables Juegos de caracteres Motores Privilegios Replicación Procesos Exportar

Importar Synchronize

Acciones

MySQL localhost

Crear nueva base de datos (B) [BBDD_DOCUMENTACION] Cotejamiento [C] [Crear] [Cotejamiento de las conexiones MySQL: utf8_general_ci] [v] [v]

Interfaz

Idioma - Language [Español - Spanish] [v]

Tema / Estilo: Original [v]

Cambiar el color: [Reiniciar]

Tamaño del font: 82% [v]

MySQL

Servidor: localhost via TCP/IP

- Versión del servidor: 5.5.8
- Versión del protocolo: 10
- Usuario: root@127.0.0.1
- Juegos de caracteres de MySQL: UTF-8 Unicode (utf8)

Servidor web

- Apache/2.2.17 (Win32) mod_ssl/2.2.17 OpenSSL/0.9.8o PHP/5.3.4 mod_perl/2.0.4 Perl/5.10.1
- Versión del cliente: mysqld 5.0.7-dev - 091210 - \$Revision: 304625 \$
- extensión PHP: mysql

phpMyAdmin

- Acerca de esta versión: 3.3.9
- Documentación
- Wiki
- Página oficial de phpMyAdmin
- [ChangeLog] [Git] [Lists]

Ilustración 124 Creación BBDD



- 3) Una nueva BBDD ha sido creada. Es necesario crear una tabla dentro de esta base de datos. En el menú **Crear nueva tabla en la base de datos** cubrir el apartado **Nombre** con el nombre deseado para la nueva tabla y el apartado **Numero de campos** con el número de campos que tendrá la nueva tabla. Pulsar **Continuar**.



Ilustración 125 Configuración BBDD

- 4) Configurar el **nombre de cada campo** de la nueva tabla, el **tipo de dato** correspondiente a cada campo, son valores imprescindibles para la correcta creación de la tabla. Una vez cubiertos estos campos, pulse **Grabar**. *En caso de que los datos sean tipo char, es necesario ajustar la longitud de los mismos.*



Ilustración 126 Configuración Campos BBDD



5) La tabla con los campos configurados ha sido creada.

La Tabla 'BBDD_DOCUMENTACION' 'DOCUMENTACION' se creó.

```

CREATE TABLE `BBDD_DOCUMENTACION`.`DOCUMENTACION` (
  `FECHA` datetime NOT NULL,
  `DISPOSITIVO` varchar(50) NOT NULL,
  `VALOR 1` double NOT NULL,
  `VALOR 2` int(11) NOT NULL,
  PRIMARY KEY (`FECHA`)
) ENGINE = InnoDB;

```

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> FECHA	datetime			No	None		[Edit] [Delete] [Add] [Refresh] [Drop] [Truncate] [Export] [Import] [Print] [View SQL]
<input type="checkbox"/> DISPOSITIVO	varchar(50)	latin1_swedish_ci		No	None		[Edit] [Delete] [Add] [Refresh] [Drop] [Truncate] [Export] [Import] [Print] [View SQL]
<input type="checkbox"/> VALOR 1	double			No	None		[Edit] [Delete] [Add] [Refresh] [Drop] [Truncate] [Export] [Import] [Print] [View SQL]
<input type="checkbox"/> VALOR 2	int(11)			No	None		[Edit] [Delete] [Add] [Refresh] [Drop] [Truncate] [Export] [Import] [Print] [View SQL]

Vista de impresión: [Vista de relaciones] [Planteamiento de la estructura de tabla]

Añadir 1 campo(s) [Al final de la tabla] [Al comienzo de la tabla] [Después de FECHA] [Continuar]

Ilustración 127 BBDD Creada