

CPU GL-MC3108-12

SISTEMA DE CONTROL DOMÓTICO

Rev. 1.0
Español

GLOBALCHIP S.L.

Índice

ÍNDICE	2
CAPÍTULO 1: INTRODUCCIÓN.....	3
1.1 PRESENTACIÓN GL-10	3
CAPÍTULO 2: HARDWARE PLC (GL-MC3108T-12).....	6
2.1 INSTALACIÓN (PLC)	9
2.2 PROTECCIÓN DE LAS SALIDAS DIGITALES A TRANSISTOR.....	9
2.3 CPU DE CONTROL DOMÓTICO (PLC).....	10
2.4 IDENTIFICACIÓN DE COMPONENTES	10
2.5 PIN-OUT PUERTOS DE COMUNICACIÓN.....	11
2.6 CABLES DE COMUNICACIÓN	11
2.7 CONEXIONES ELÉCTRICAS, MÓDULO CONTROL DOMÓTICO (CPU).....	12
2.8 DISTRIBUCIÓN GENERAL DE SALIDAS DIGITALES Y ALIMENTACIÓN.....	13
2.9 DISTRIBUCIÓN GENERAL DE ENTADA DIGITALES Y SALIDA 12VDC.....	13
2.10 EJEMPLO DE ASIGNACIÓN ENTRADAS / SALIDAS DIGITALES.....	14
2.11 DISTRIBUCIÓN ENTRADAS DIGITALES EN EL SIMULADOR.....	14
2.12 PIN-OUT PUERTOS DE COMUNICACIÓN.....	15
2.13 PASOS PARA LA PROGRAMACIÓN DE LA CPU.....	15
2.14 CABLES DE COMUNICACIÓN	16
2.15 ESQUEMA CONEXIONES ELEMENTOS DE ENTRADA.....	17
2.16 ESQUEMA CONEXIONES ELEMENTOS DE SALIDA	17
2.17 ESQUEMA CONEXIÓN SALIDA DIGITAL. CONTROL DE UN RELÉ.....	18
2.18 ESQUEMA DE CONEXIÓN SALIDA DIGITAL. CONTROL RELÉ SSR.....	18
2.19 ESQUEMA DE CONEXIÓN SALIDA DIGITAL. CONTROL RELÉ DE BOBINA.....	19
2.20 ESQUEMA DE CONEXIÓN FUENTE DE ALIMENTACIÓN UPS.....	20
2.21 SOFTWARE DE PROGRAMACIÓN	21
2.22 ARQUITECTURA DE LA PROGRAMACIÓN	25
2.23 LAS VARIABLES	26
2.24 LIBRERÍAS ESTÁNDAR DE POWERPRO.....	27
2.25 LIBRERÍAS GL-10	28
2.26 JUEGOS DE INSTRUCCIONES:	30
2.27 TIPOS DE DATOS:	38
2.28 REGISTROS INTERNOS DE LA CPU.....	40

Capítulo 1: Introducción

1.1 Presentación GL-10

Se define como un sistema avanzado para la gestión global de hogares y oficinas. Permite al usuario, configurar múltiples funciones de control y creación de escenas. Una simple pulsación sobre un botón en su pantalla táctil o desde un mando a distancia vía radio, será suficiente para controlar varios puntos de iluminación, mover una cortina y activar su Home Cinema.

GL-10 se destina a su integración en promoción de obra nueva, mediante estructura cableada descentralizada de sus entradas y salidas digitales. Posibilidad de crecer en el futuro, con nuevas funcionalidades y adaptarse a la demanda del cliente mediante comunicación vía radio frecuencia.

Para residencias en rehabilitación, es viable su integración, con escasa obra adicional gracias a su posibilidad de comunicación vía radio.

El sistema GL-10 se divide en 2 partes principales:

a.- PLC (Controlador lógico programable), se distribuyen por la vivienda de forma descentralizada, los PLC's se comunican con el Panel de control mediante un BUS de 3 hilos RS-485. Su programación la realiza el instalador bajo el estándar IEC 61131-3 y se realizará un proyecto específico para cada instalación.

Sus entradas digitales pueden ser activadas por; pulsadores, finales de carrera, barras de seguridad, sensores en general.

Sus salidas digitales actúan sobre relés externos, los cuales pueden controlar; líneas de iluminación, enchufes, motores de persiana, toldos, electroválvulas, sirenas, etc.

El usuario tendrá el control de los dispositivos de su vivienda, actuando sobre los pulsadores asociados a los PLC's independientemente del estado o configuración del Panel de Control. Su memoria no volátil contiene el programa funcional completo de la zona de la vivienda asignada. Esta característica confía una gran estabilidad de funcionamiento al sistema.

b.- PANEL DE CONTROL, Interface gráfico con pantalla táctil, ofrece al usuario un control centralizado de su vivienda, podríamos decir que para las funciones básicas como el control de Iluminación, electrodomésticos, persianas, etc., trabaja en paralelo con los mecanismos instalados en la vivienda.

El Panel de Control ofrece al usuario otras funciones adicionales que los PLC's no pueden realizar, por ejemplo:

- Programación horaria.
- Control de temperatura y transmisión de órdenes por infrarrojos para equipos de Aire acondicionado.
- Central de intrusión y alarmas técnicas con llamada telefónica al usuario.
- Mini-mandos tipo llavero, permiten armar y desarmar las funciones de alarma o controlar escenas vía radio.
- Vigilancia por circuito cerrado de TV y video portero.
- Comunicaciones avanzadas de voz y video TCP/IP.
- Funciones de intercomunicación con la comunidad de vecinos.
- Módem telefónico para el control del sistema de forma remota.
- Mandos multifunción con capacidad de aprender códigos de otros mandos I.R. permitiendo integrar las funciones de control de audio y video, regulación de luces, persianas, toldos, etc.

Es posible instalar varios Paneles de Control en una misma vivienda y conectarlos en RED. También pueden crearse redes comunitarias para Bloques de pisos o urbanizaciones. En las que todos los vecinos estarían intercomunicados si se desea. Posibilidad de acceder a un servidor que facilite información de interés para la comunidad o realizar una llamada a un puesto de control donde un vigilante atienda una llamada de emergencia de un usuario.

GL-10, permite programar múltiples funciones de automatización como el encendido y apagado de dispositivos eléctricos, aire acondicionado, control de la caldera, invernaderos, acuarios, control de luces, recogida de toldos, control de persianas y cortinas, riego automatizado, escenas con funciones multimedia, visualizar la señales de una cámaras CCTV, etc.

Las principales funciones de control pueden iniciarse a partir de:

- Una orden desde un mando a distancia por radiofrecuencia.
- A causa de cambios de estado en detectores de movimiento, magnéticos, detectores de humo, fuga de gas, inundación, etc.
- Una llamada al video portero.
- Programación horaria.
- Una temperatura de consigna.
- Una alarma de intrusión.

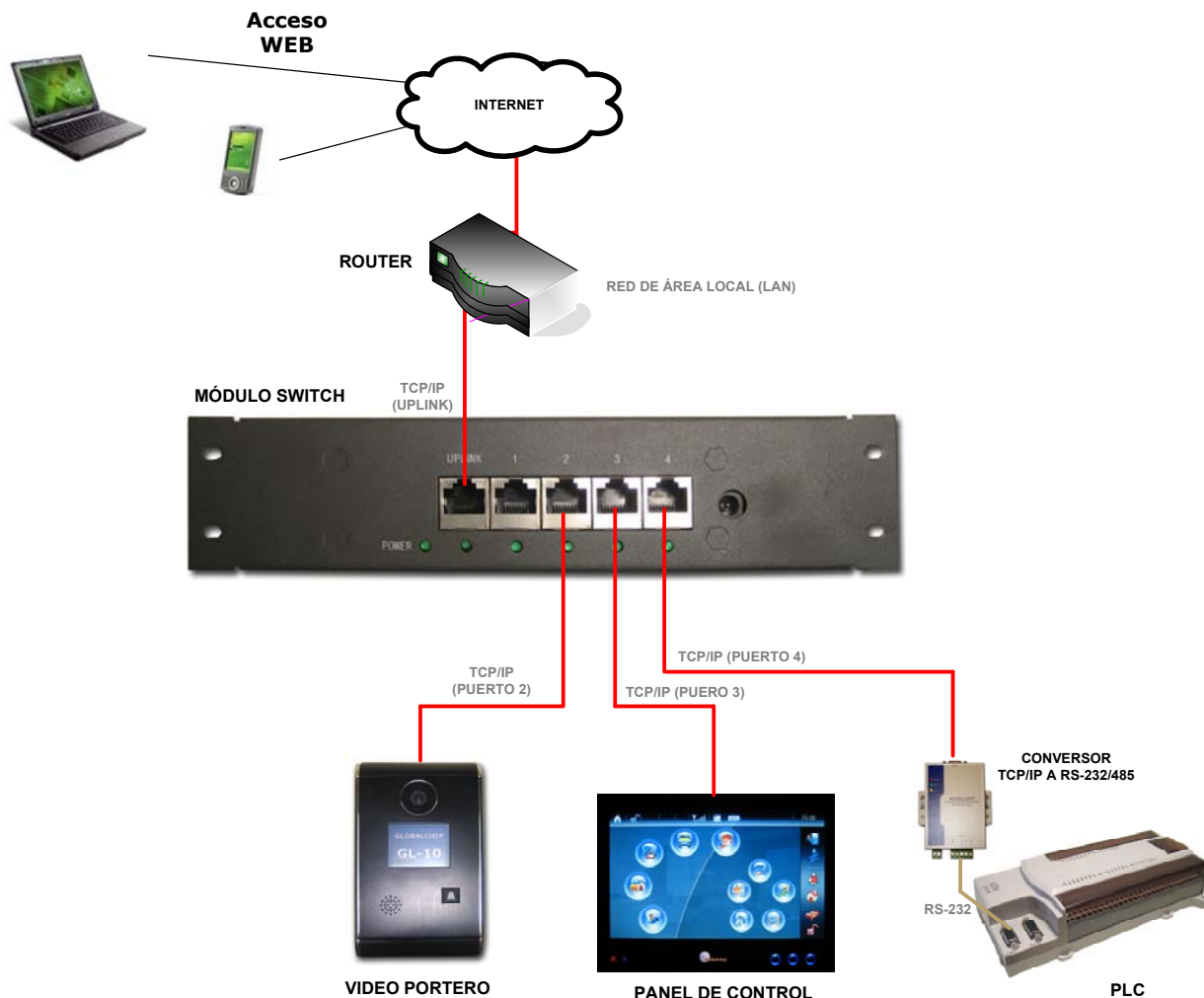
La configuración y personalización de la vivienda se realiza en el propio Panel de Control y no se requiere de software adicional para este fin. Es posible salvar una configuración, recuperarla o transferirla a otro Panel de Control.

Una vez definido el hardware a implementar en la vivienda, finalizado el cableado unifilar, conexionado de los distintos módulos de control mediante el BUS RS-485 y proveer al Panel de Control de un acceso a la RED local, los pasos de configuración serían los siguientes:

- Definición del hardware "Remote Interface Unit" (RIU); **Security Module, PLC Module, RF Module**, etc.
- Definición de estancias y asignación de etiquetas descriptivas; **Salón, Cocina, Aseo**, etc.
- Definición de detectores por estancia y asignación de etiquetas descriptivas, **Pulsador LUZ, Enchufe, TV, Calefacción**, etc.
- Asignación de detectores conectados al modulo de seguridad "Security Module"; **PIR_Entrada, Magnetico_VENTANA_01, Inundación_01**, etc.
- Registro de mandos a distancia, códigos infrarrojos, etc.
- Configuración de alarmas técnicas, intrusión, etc.
- Configuración control de temperatura, programación horaria, etc.
- Programación de escenas.
- Configuración de comunicaciones, direccionamiento IP, etc.
- Puesta en marcha.

■ Ejemplo de interconexión, equipos en RED:

- Las interconexiones de RED se realizarán según las necesidades de cada instalación. Recuerde que la distancia máxima de un cable de RED CATEGORÍA-5 es de 100 metros.
- El SWITCH, condiciona que el tráfico de datos que se genera entre sus puertos, no salga a la RED general, de esta forma, no ocupamos ancho de banda de la LAN con las comunicaciones locales.



NOTA: La conexión del PLC con el conversor de medio a la RED, se destina a servicios de tele-mantenimiento. Las órdenes de control y automatización sobre el sistema, se realizaran desde el Panel de Control sobre el PLC, a través del BUS RS-485.

Capítulo 2: Hardware PLC (GL-MC3108T-12)

Controlador programable (PLC) de 24 entradas digital (DI), 16 canales digitales a transistor (DO) todo ello en un compacto controlador domótico para crear un potente micro PLC. Alimentación 12V y montaje sobre carril DIN.

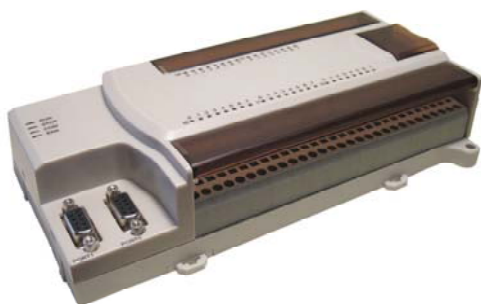
Programable mediante software estándar IEC-61131-3, disponibles múltiples librerías de funciones.

Comunicación con la Panel de control mediante BUS RS-485, proporcionando topología descentralizada del cableado. Fácil configuración mediante asociación de sus entradas y salidas en el Panel de Control. El direccionamiento del protocolo de comunicación permite controlar hasta 7 equipos, la direcciones se repartirán entre PLC's y módulos Repetidores de radiofrecuencia.

El módulo CPU GL_MC3108T-12, combina un microprocesador, una fuente de alimentación integrada y 40 canales digitales con posibilidad de ampliar mediante módulos adicionales de entradas y salidas digitales, analógicos, entradas especiales para sondas de temperatura y comunicaciones MODBUS sobre TCP/IP.

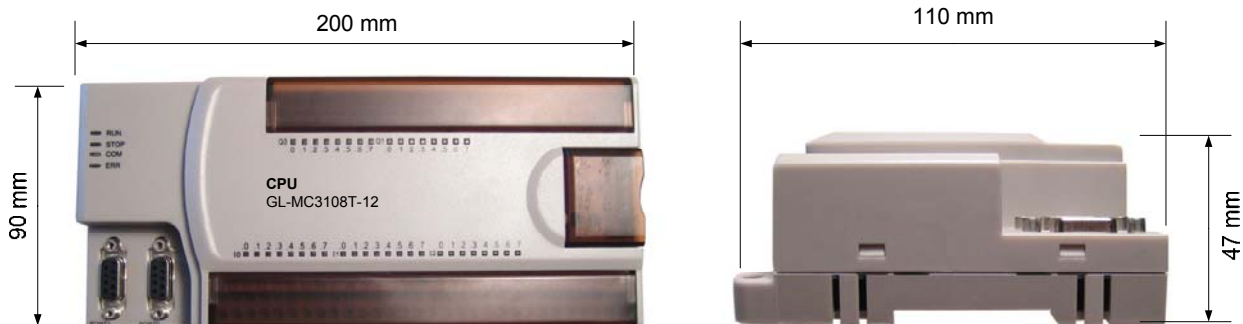
Referencia Producto:

GL-MC3108T-12



Alimentación	12VDC (9 a 18VDC)
Consumo máx.	200 mA.
Entradas digitales	24 (optoacopladas) (9 a 18VDC)
Salidas digitales	16 (Transistor NPN) (5 a 32VDC)
Nº Máximo módulos de Expansión	4 (en preparación)
Potenciómetros ajustables	2 con 8 bits de resolución (0 a 255)
Memoria de programa	120Kb.
Memoria no volátil	6Kb.
Puerto 0 (no aislado)	RS-485 (comunicación con el Panel de Control GL-10).
Puerto 1 (no aislado)	RS-232 (programación)
Temperatura de trabajo	0 a 50°C
Humedad	<=90% sin condensación
IP	20
Dimensiones	200 x 90 x 47 mm
Material carcasa	ABS

- Dimensiones:



■ Características generales:

Supply Voltage: 12VDC

DI: 24× 12VDC

DO: 16× Transistor (5 to 30VDC)

Output Power Supply: 12VDC 200mA

Program Capacity: 120K byte

Input Storage: 512 byte

Output Storage Area: 512 byte

Intermediate Storage Area: 8K byte

Global Area: 24K byte

EEPROM: 6K byte

Timer Extra long no limit (1ms as smallest unit, up to 49 days)

Counter: No limit

Password Protection: Yes

Real-time Clock, Power-off EEPROM Time: 10 days

User Program, Power-off, EEPROM Time: 10 years

Operation Speed (Basic Command): 0.65us

Programming Language In accordance with IEC61131-3 International Standards with 5 programming languages: LD, IL, FBD, SFC, ST

PowerPro Internal Commands: 59items

PowerPro External Commands: 18 items

PowerPro Function Blocks Over: 70 items

Physical Interface RS232 (Non-isolating), RS485 (Non-isolating)

Interrupt Input (rising/ descending optional) 3 points

Pulse Catch Input (rising/descending) 3 points

Single-phase high-speed counter: 3 points, 100KHz

High-speed Counter, Double-phase high-speed counter: 2 points, 100KHz supporting addition and subtraction double counting pulse and 90 degree cross counting pulse

Output mode: pulse serial output PTO, pulse width output PWM

High-speed Output transistor (Output frequency): 20KHz

Max. Extension modules: 7

Storage temperature: -20~700°C

Working temperature: 0~55°C

Environment Humidity: 5%-95%, non-condensing.

Electrostatic Discharge Immunity:

In accordance with IEC 61000-4-2: 1995 standard level 3

Electrical Fast Transient/Burst Immunity:

In accordance with IEC 61000-4-4: 1995 standard level 3

Continuous Wave Conducted Immunity:

In accordance with IEC 61000-4-6: 1995 standard level 2

Isolation Pressure-Resisting: AC1500V 1min (between earth terminal and other terminals)

Isolation Resistance Above: 500M

Working Environment: No corrosive gas, no dust

Extension Module

LM3210 8×DC24V Input

LM3211 8×AC220V Input

LM3212 16×DC24V Input

LM3213 16×AC220V Input

LM3214 32×DC24V Input

LM3215 32×AC220V Input

LM3220 8×DC24V Transistor Output

LM3221 16×DC24V Transistor Output

LM3222 8×Relay Output

LM3223 16×Relay Output

LM3224 32×DC24V Transistor Output Extension I/O Module

LM3225 32×Relay Output

LM3230 DI 4×DC24V, DO 4×DC24V Transistor

LM3231 DI 4×DC24V, DO 4×Relay

LM3232 DI 8×DC24V, DO 8×DC24V Transistor

LM3233 DI 8×DC24V, DO 8×Relay

LM3234 DI 16×DC24V, DO 16×DC24V Transistor

LM3235 DI 16×DC24V, DO 16×Relay

LM3236 DIO 8×DC24V Transistor

LM3310 4 Channel AI (4-20mA/0-20mA /0-10V Optional)

LM3311 4 Channel thermocouple input (J,K,E,N,T,B,R,S type)

LM3312 4 channel RTD input (Cu50, PT100 type)

LM3313 8 Channel AI (4-20mA/0~20mA/0-10V Optional)

LM3320 2 Channel Analog Output (0-20mA /0-10V Optional)

LM3330 4 Channel AI, 1 Channel AO (4-20mA/0-20mA/0-10V Optional)

Extension Function Module

LM3401 Profibus-DP Protocol slave station interface module

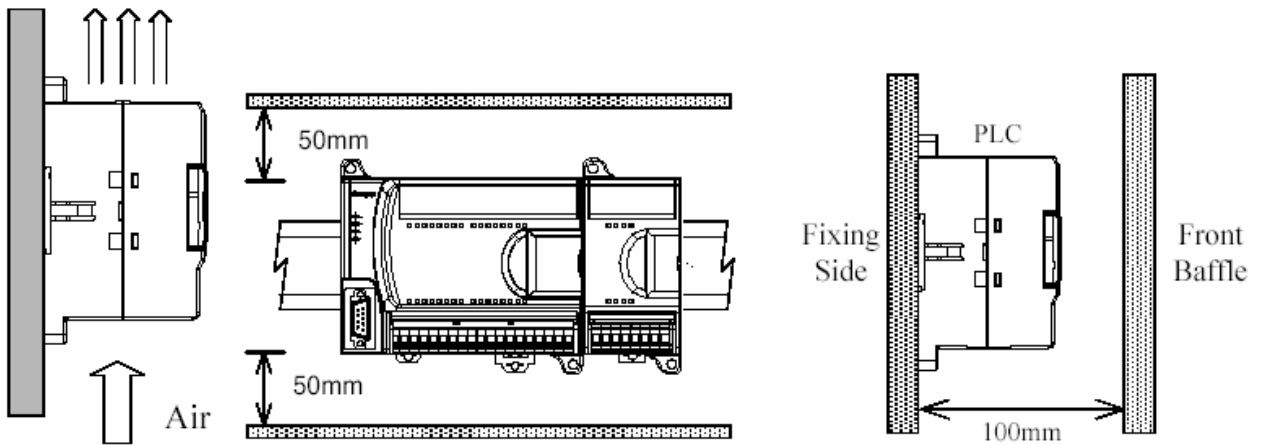
Maximum I/O Configuration

	Digital Input	Digital output	Analog Input	Analog Output
CPU Local	24	16		
Max. Digital I/O	24+224	16+224		
Max. Analog I/O			56	14

2.1 Instalación (PLC)

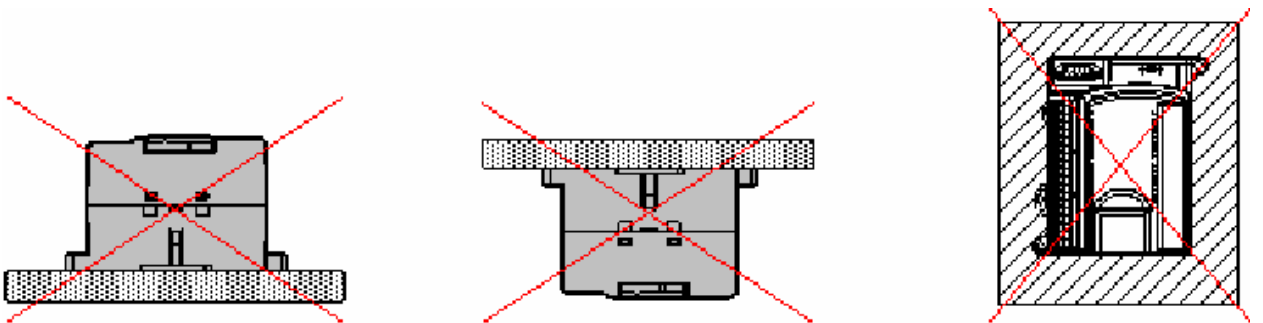
■ Ventilación:

Respete las siguientes indicaciones para procurar una correcta ventilación de los equipos.



■ Instalación:

No instale el PLC en las siguientes posiciones, esto obstruirá la correcta ventilación por convección en los equipos.

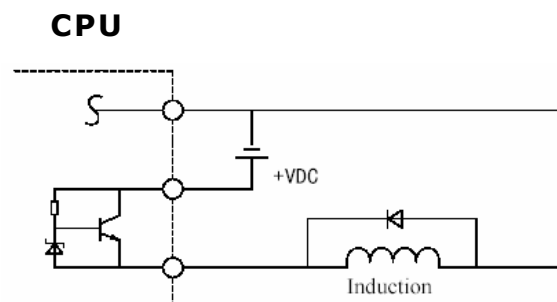


2.2 Protección de las salidas digitales a transistor

Es muy importante proteger los transistores de las salidas digitales, frente a corrientes inversas cuando se controlan cargas inductivas, bobinas, motores, relés, etc.

Coloque un diodo de protección en la bobina a controlar, tal como se indica en la siguiente figura. Un modelo 1N-4007 puede ser adecuado.

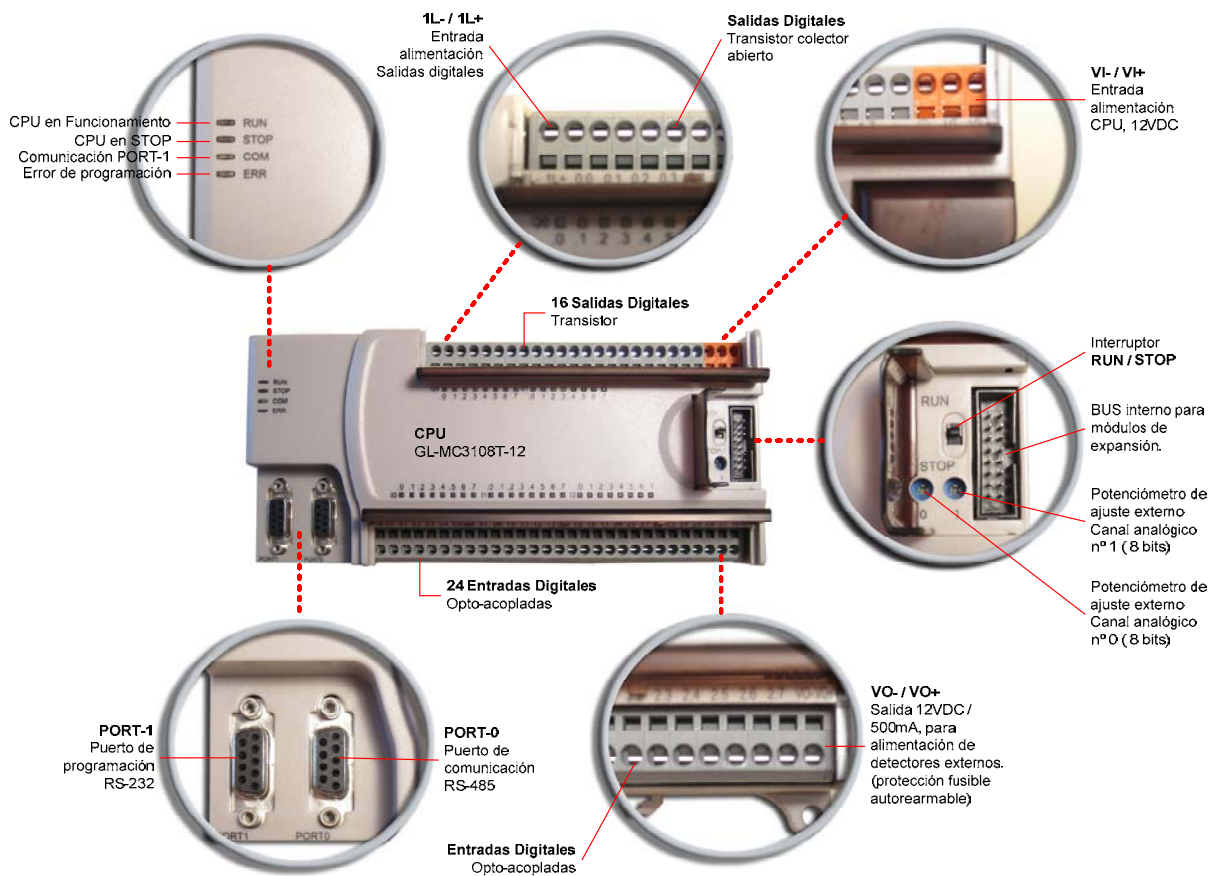
Recuerde que las salidas a transistor no pueden controlar cargas con un consumo superior a 500mA.



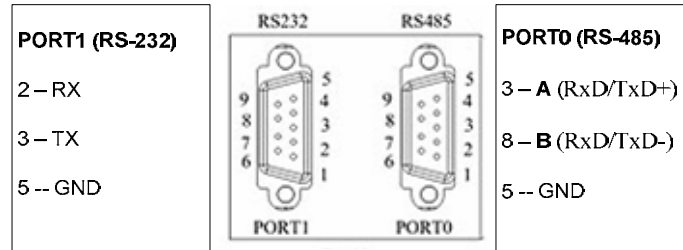
2.3 CPU de control domótico (PLC)



2.4 Identificación de componentes



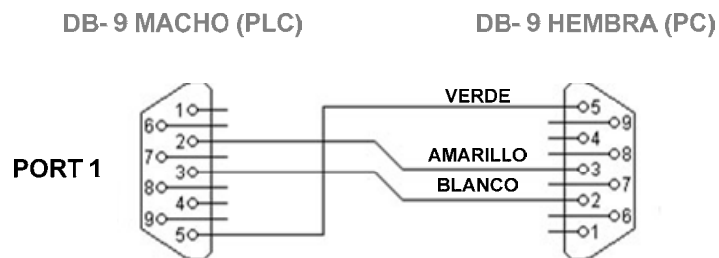
2.5 Pin-out puertos de comunicación



NOTA: Los pines de los conectores no referenciados no tiene uso.

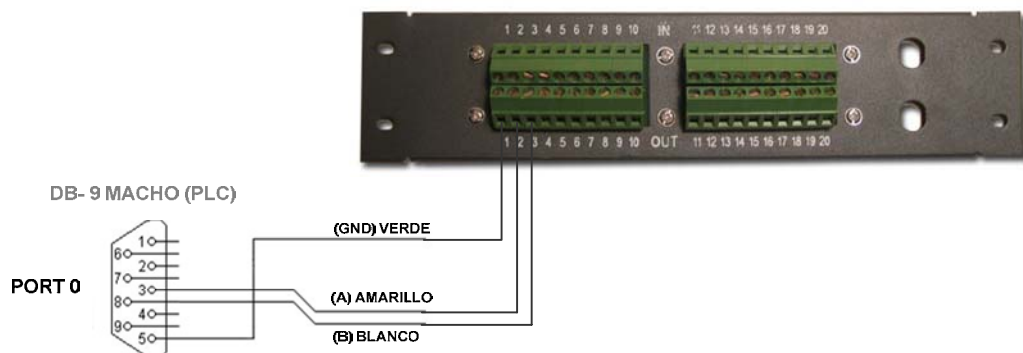
2.6 Cables de comunicación

■ Cable de programación RS-232 (PORT 1):



PORT 0	Conector 2	Función	
2	3 (Amarillo)	Rx ←	Tx
3	2 (blanco)	Tx →	Rx
5	5 (verde)	GND	

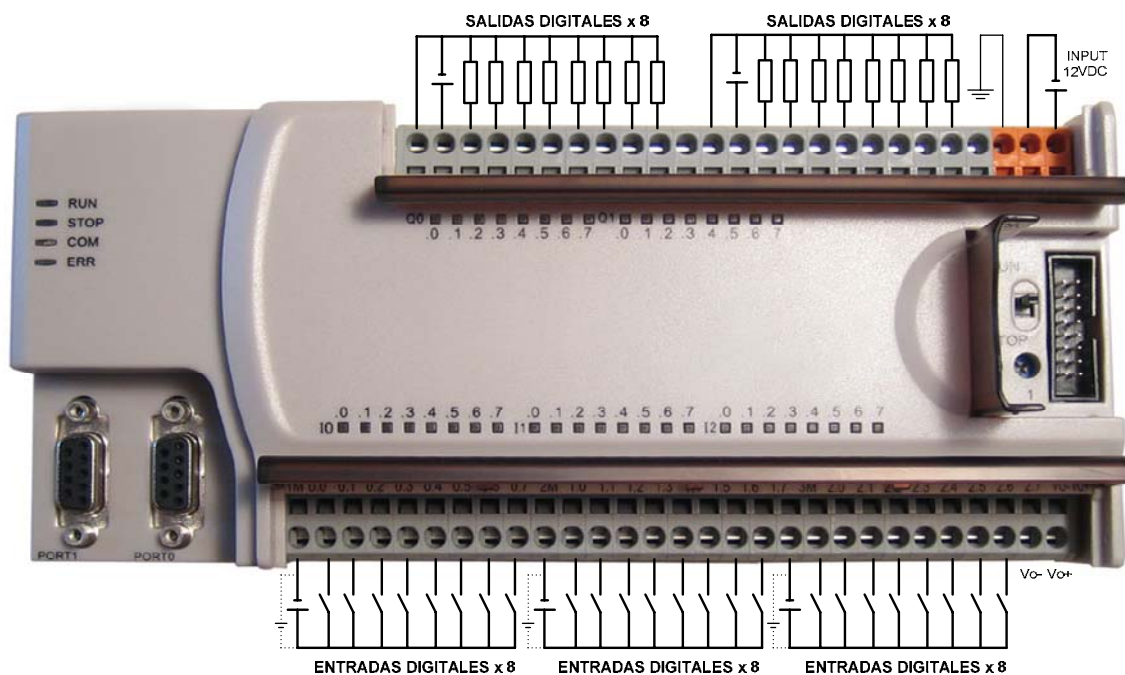
■ Cable de comunicación RS-485 (PORT 0):



PORT 0	Manguera 4hilos	UTP (Par-4)	Armario de distribución (Bloque de regleta nº 1)	Función
5	Verde	Verde	1	GND
3	Amarillo	Marrón	2	A
8	Blanco	Marrón/Blanco	3	B


2.7 Conexiones eléctricas, Módulo control domótico (CPU)

1. El número de borna se incrementa de izquierda a derecha.
2. Las bornas 1L y 2L, son los comunes de alimentación para las salidas digitales. Las salidas son a transistor (MOSFET) y entregan positivo sobre la carga, por lo tanto las cargas tendrán como común la línea de masa.
3. En la parte superior derecha junto a las bornas de salidas digitales, los terminales L y N (de color naranja), son la entrada de alimentación 12VDC de la CPU y la borna \oplus puede ser conectada a tierra como protección.
4. Las bornas marcadas con 1M, 2M, y 3M son los comunes de cada uno de los 3 bytes de entradas digitales, pueden referenciarse independientemente con el positivo o con el negativo de la fuente de alimentación.
5. En la parte inferior derecha, junto a las bornas de entradas digitales, los terminales marcados como VO+ y VO-, son positivo y negativo de la salida de alimentación (12VDC) desde la CPU hacia los pulsadores y sensores del sistema. Está salida está limitada en corriente con protección contra sobrecargas.
6. **Importante:** las salidas digitales Q0.3 y Q1.1, no son adecuadas para controlar relés de estado sólido SSR, utilizar relés de bobina mecánica para el control de cargas. Las bornas marcadas con * no tiene uso



2.8 Distribución general de salidas digitales y alimentación

▪ Byte 0 y 1:

- 1L -** → (GND) Masa entrada alimentación, común Byte 0
- 1L +** → (VCC) Positivo Entrada alimentación 9 a 32VDC
- Q0.0** → Salida general +
- Q0.1** → Salida general +
- Q0.2** → Salida general +
- Q0.3** → Salida general + / salida contador alta velocidad (máximo 10mA)
- Q0.4** → Salida general +
- Q0.5** → Salida general +
- Q0.6** → Salida general +
- Q0.7** → Salida general +
- * --
- 2L -** → (GND) Masa entrada alimentación, común Byte 1
- 2L +** → (VCC) Positivo Entrada alimentación 9 a 32VDC
- Q1.0** → Salida general +
- Q1.1** → Salida general + / salida contador alta velocidad (máximo 10mA)
- Q1.2** → Salida general +
- Q1.3** → Salida general +
- Q1.4** → Salida general +
- Q1.5** → Salida general +
- Q1.6** → Salida general +
- Q1.7** → Salida general +
- * --
-  → Borna de Tierra
- VI -** → (GND) Masa entrada alimentación CPU
- VI +** → (VCC) Positivo entrada alimentación CPU +12VDC

2.9 Distribución general de Entada digitales y salida 12VDC

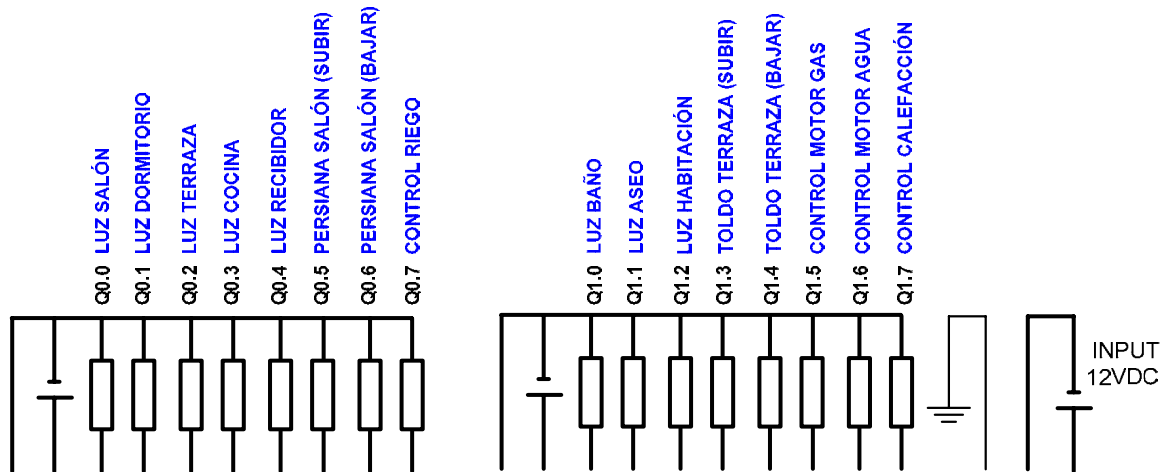
▪ Byte entrada 0, 1 y 2:

- 1M** → (VCC o GND) Común Byte 0
- I0.0** → Entrada general / Entrada contador alta velocidad.
- I0.1** → general input / control the importation of high-speed counter -
- I0.2** → Entrada general / Entrada contador alta velocidad.
- I0.3** → general input / control the importation of high-speed counter -
- I0.4** → Entrada general / Entrada contador alta velocidad.
- I0.5** → Entrada general / Entrada contador alta velocidad.
- I0.6** → Entrada general / Entrada contador alta velocidad. / External interrupt input / pulse capture input
- I0.7** → general input / external interrupt input / Pulse capture input

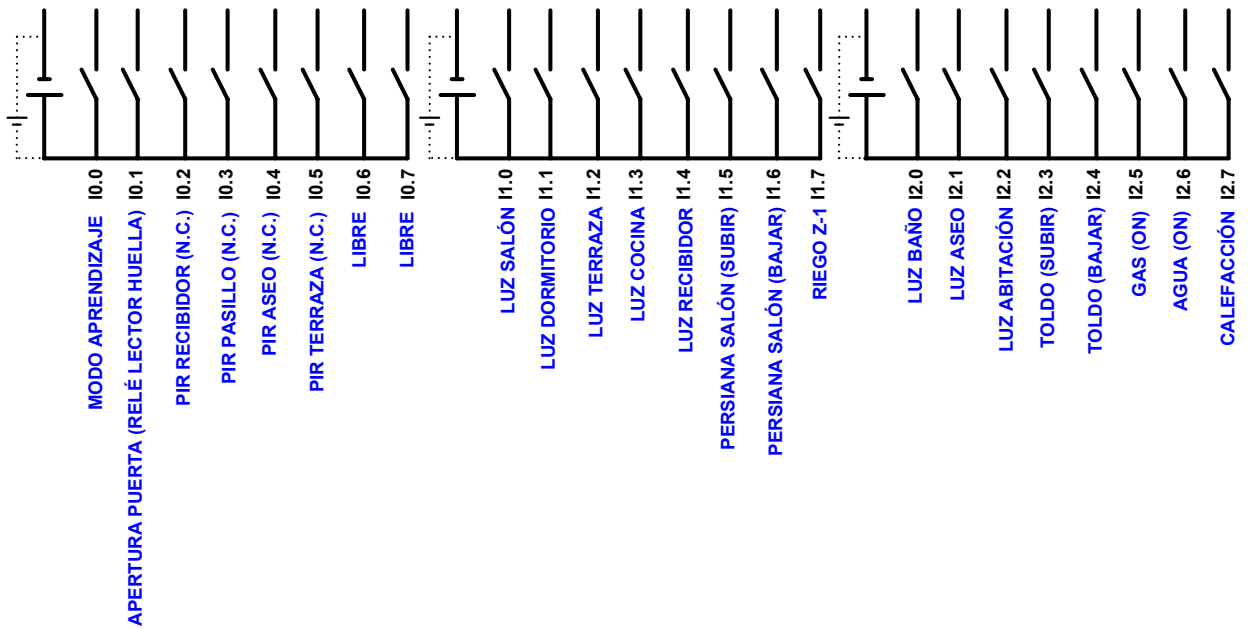
- 2M** → (VCC o GND) Común Byte 1
- I1.0** → Entrada general / External interrupt input / Pulse capture input
- I1.1** → Entrada general / External interrupt input / Pulse capture input
- I1.2** → Entrada general
- I1.3** → Entrada general
- I1.4** → Entrada general
- I1.5** → Entrada general
- I1.6** → Entrada general
- I1.7** → Entrada general

- 3M** → (VCC o GND) común Byte 2
- I2.0** → Entrada general
- I2.1** → Entrada general
- I2.2** → Entrada general
- I2.3** → Entrada general
- I2.4** → Entrada general
- I2.5** → Entrada general
- I2.6** → Entrada general
- I2.7** → Entrada general
- VO -** → (GND) Masa, Salida 12V limitada en corriente a 400mA
- VO +** → Positivo Salida 12V limitada en corriente a 400mA

2.10 Ejemplo de asignación entradas / salidas digitales



2.11 Distribución Entradas Digitales en el Simulador



2.12 Pin-out puertos de comunicación

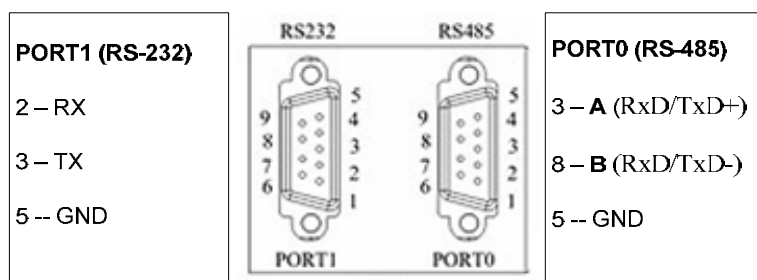


Figura 3

NOTA: Los pines de los conectores no referenciados no tiene uso.

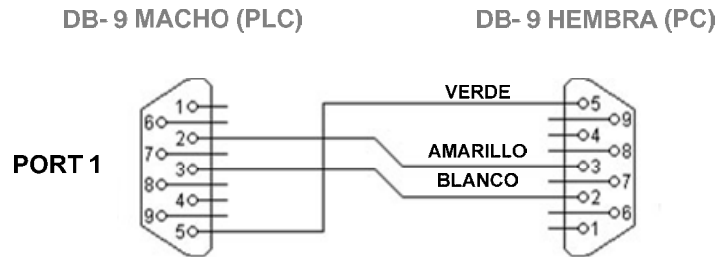
2.13 Pasos para la programación de la CPU

1. La **Figura 3** muestra el PORT1 (RS-232), que se emplea en la programación de la CPU. Puede ser de forma local mediante una conexión con cable RS-232 entre PC y PLC o de forma remota mediante un convertor RS-232 a TCP/IP.
2. Antes de transferir el proyecto en el PLC, asegúrese de que ha sido compilado correctamente.
3. Verifique que el software de programación no se encuentra en modo simulación.
4. Asegúrese de que los parámetro de comunicación son correctos, puerto sobre el que vamos a transferir los datos (conexión física entre el PC y el PLC), velocidad de comunicación 38400 bps, 1 bit de STOP y sin paridad.
5. Ejecutar en PowerPro [Online]->[Login], se iniciará la transferencia del programa al PLC.

Nota: Por cuestiones de seguridad, no es posible leer el programa del PLC al PC. Asegúrese de guardar y etiquetar de forma clara sus proyectos.
6. Una vez finalizada la transferencia del nuevo programa al PLC, dispone de un intervalo de tiempo muy breve, para confirma en un pantalla emergente de PowerPro, la solicitud de confirmación de guardar el programa en la memoria no volátil.
 - a. Si confirma este proceso, el programa queda residente en el PLC.
 - b. Si no lo confirma o se produce un time-out, el nuevo programa se pierde y el PLC continuará con el programa antiguo sin ser alterado.
7. Para finalizar el proceso, ejecutar en PowerPro [Online]->[RUN], para arrancar el nuevo programa instalado en la CPU.

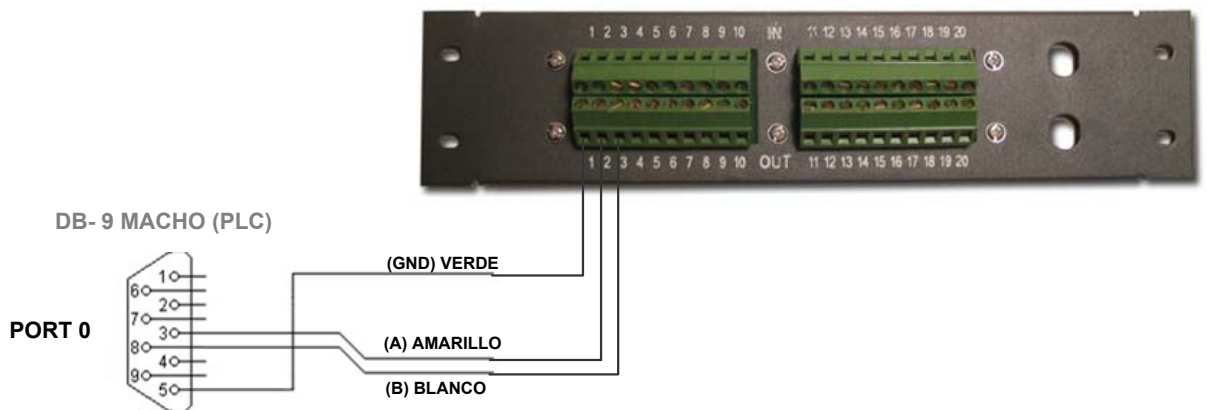
2.14 Cables de comunicación

■ Cable de programación RS-232 (PORT 1):



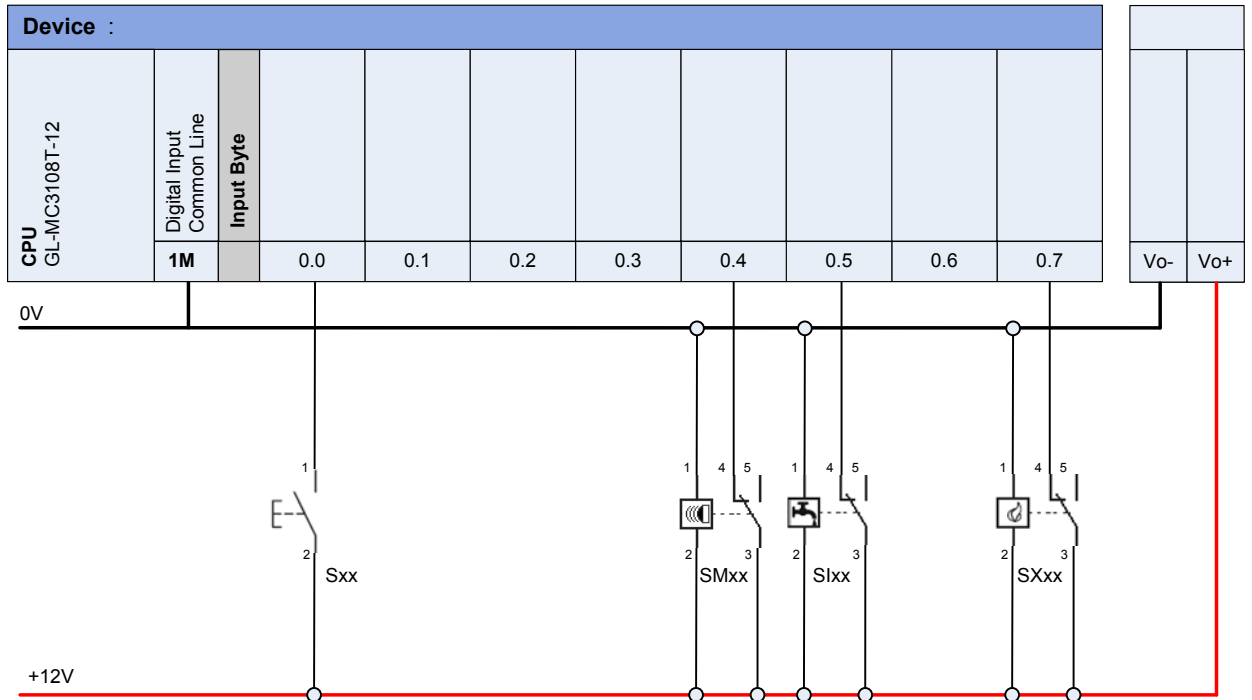
PORT 0	Conector 2	Función
2	3 (Amarillo)	Rx ← Tx
3	2 (blanco)	Tx → Rx
5	5 (verde)	GND

■ Cable de comunicación RS-485 (PORT 0):

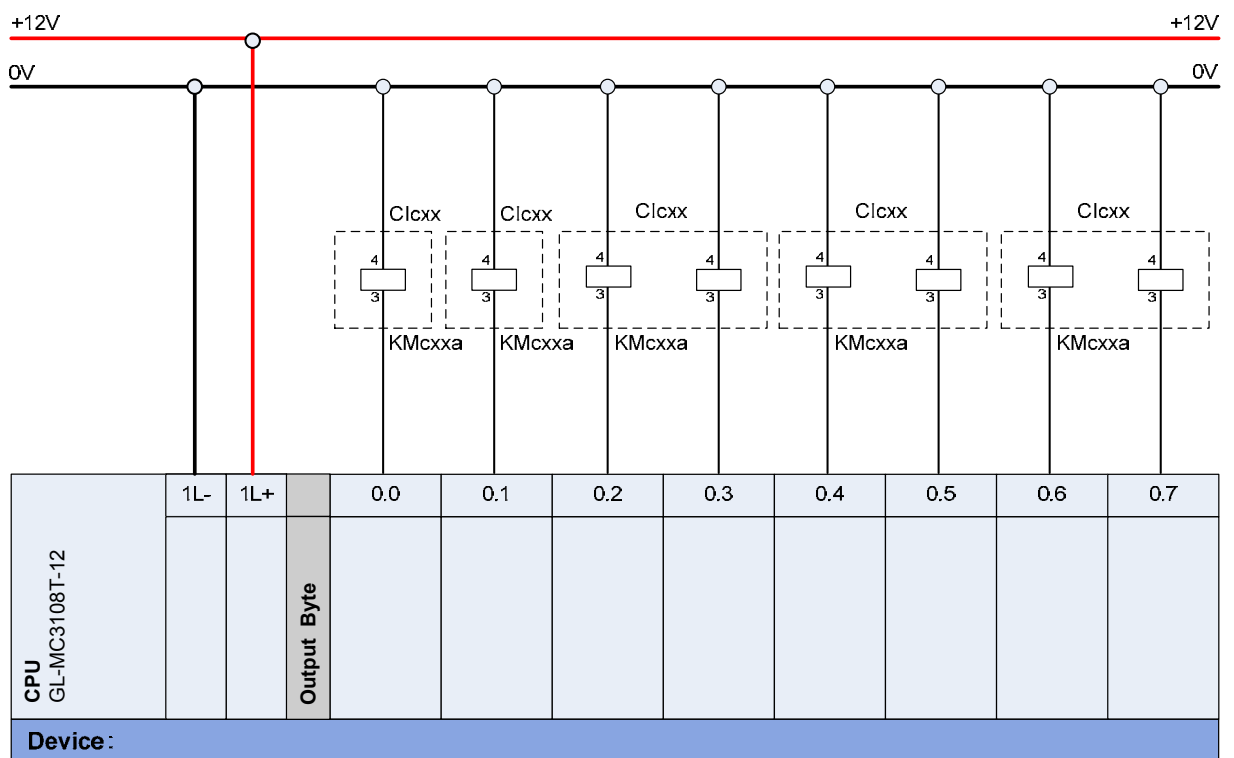


PORT 0	Manguera 4hilos	UTP (Par-4)	Armario de distribución (Bloque de regleta nº 1)	Función
5	Verde	Verde	1	GND
3	Amarillo	Marrón	2	A
8	Blanco	Marrón/Blanco	3	B

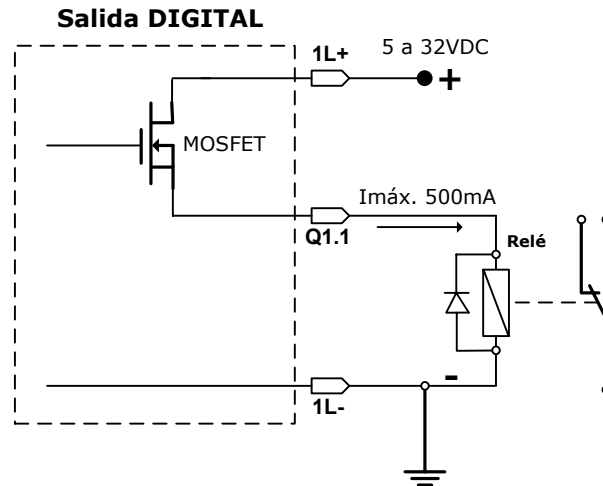
2.15 Esquema conexiones elementos de entrada



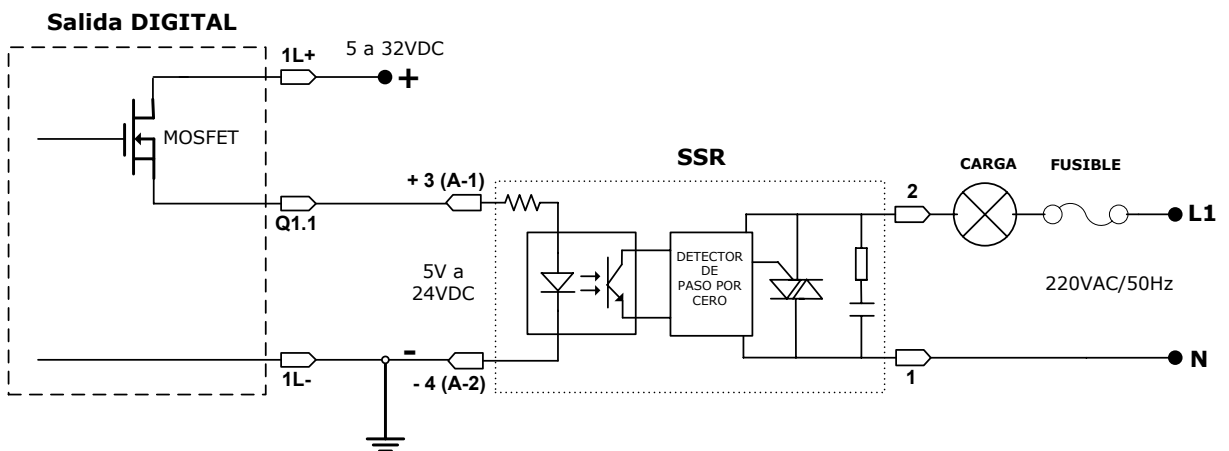
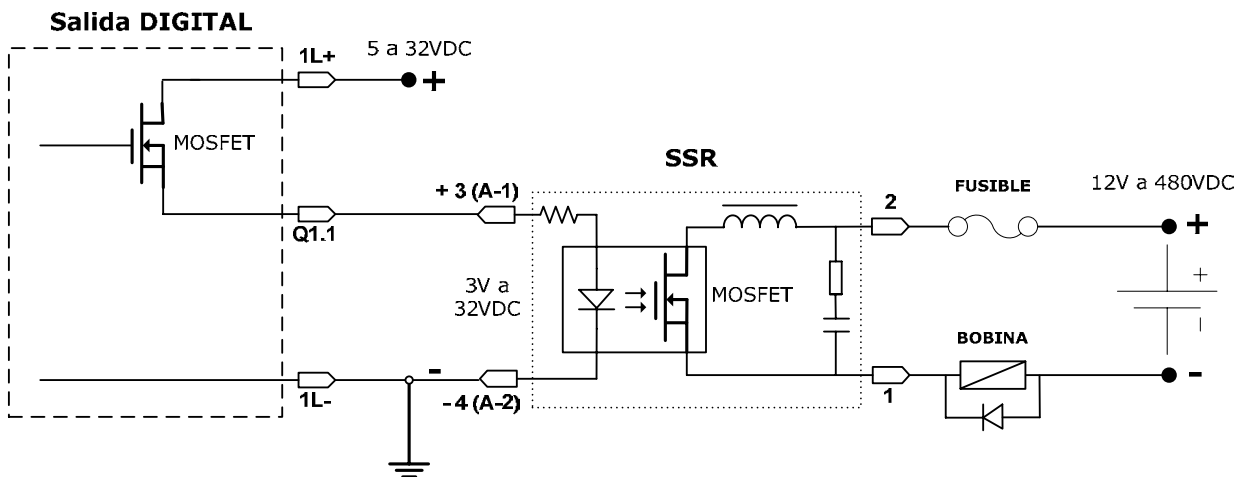
2.16 Esquema conexiones elementos de salida



2.17 Esquema conexión salida digital. Control de un Relé

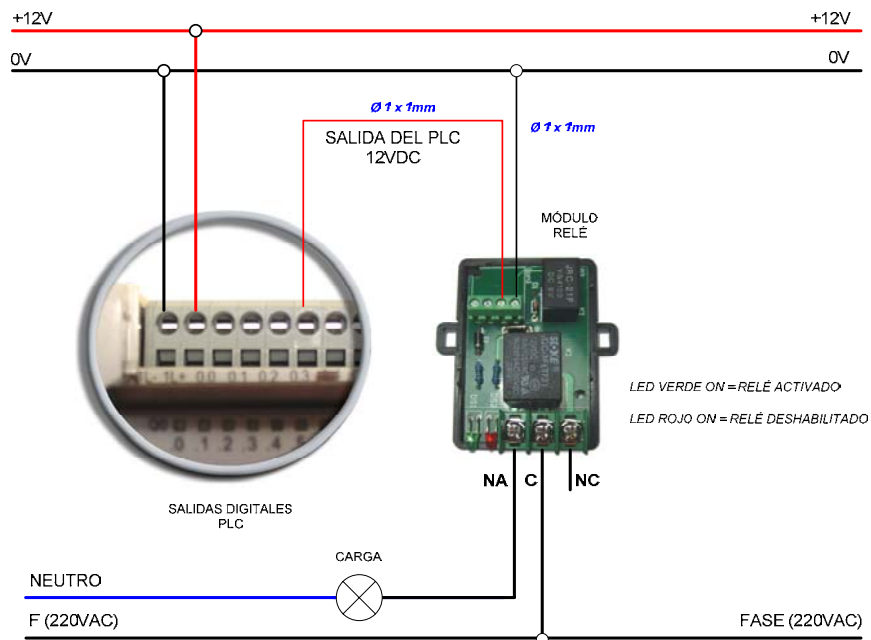


2.18 Esquema de conexión salida digital. Control relé SSR



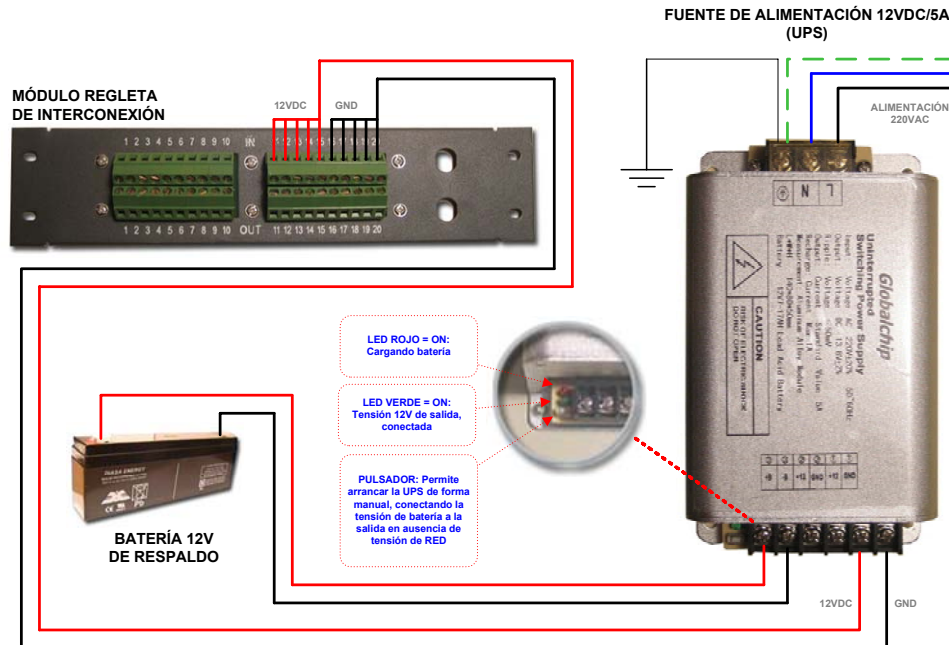
2.19 Esquema de conexión salida digital. Control relé de BOBINA

- Esquema de instalación básica, control ON/OFF de una línea de alumbrado:



2.20 Esquema de conexión y distribución de tensiones en el Módulo regleta de interconexión:

■ Esquema de conexión y distribución de tensiones en el Módulo regleta de interconexión:



■ Recomendaciones:

- Para la entrada de alimentación a la UPS, utilizar cable unifilar de 1,5mm de sección según normativa de baja tensión y realizar la correcta conexión de tierras con la caja del armario de distribución doméstica.
- Para el cableado de baterías y Módulo regleta, se recomienda utilizar cable unifilar aislado de color ROJO y NEGRO de 1,5mm de sección.

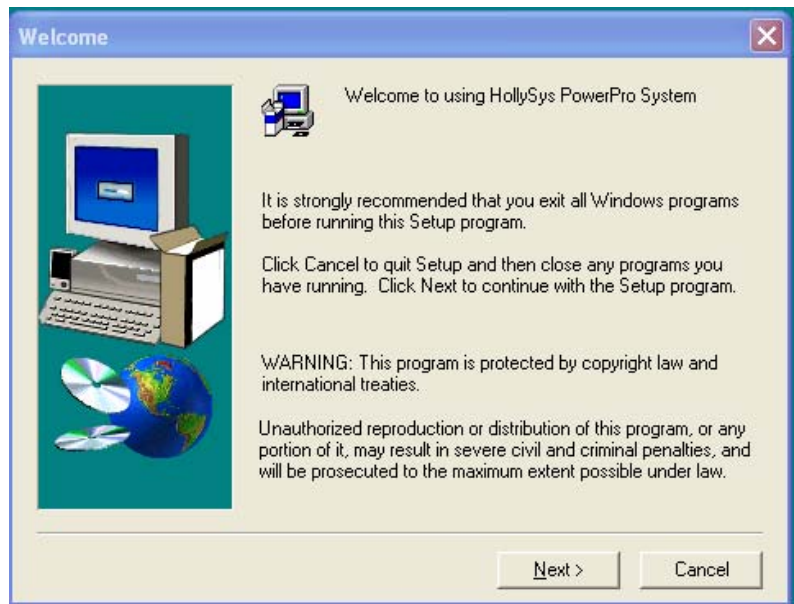
Para la alimentación de dispositivos instalados en exteriores como detectores de velocidad del viento, crepusculares, cámaras, video portero, etc. Se recomienda utilizar una fuente de alimentación independiente a la del sistema para evitar posibles sabotajes.

2.21 Software de programación

La programación del Módulo de Control Domótico está basada en la filosofía PLC, se realiza mediante el software de programación PowerPro de CoDeSys, bajo el estándar IEC 61131-3, siendo este uno de los sistemas más comunes de programación para PLCs y otros equipos industriales.

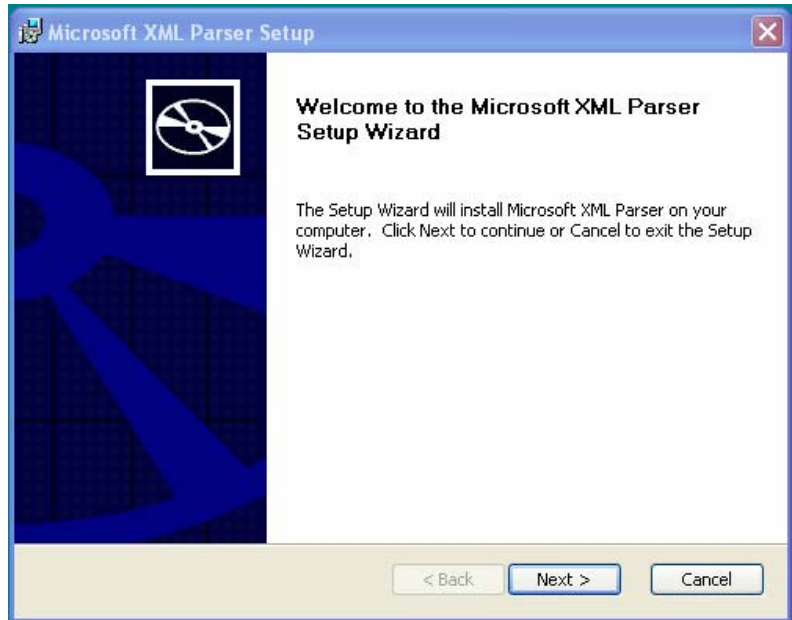
CoDeSys, soporta cinco lenguajes de programación estándar, dispone de visualización integrada y simulador OffLine. Se trata de un software de programación abierto ofrecido por GLOBALCHIP a sus instaladores.

1.- La instalación de PowerPro, se realiza de forma automática ejecutando el programa *Setup.exe* dentro del paquete comprimido **PowerPro2.1.1EN.rar**.



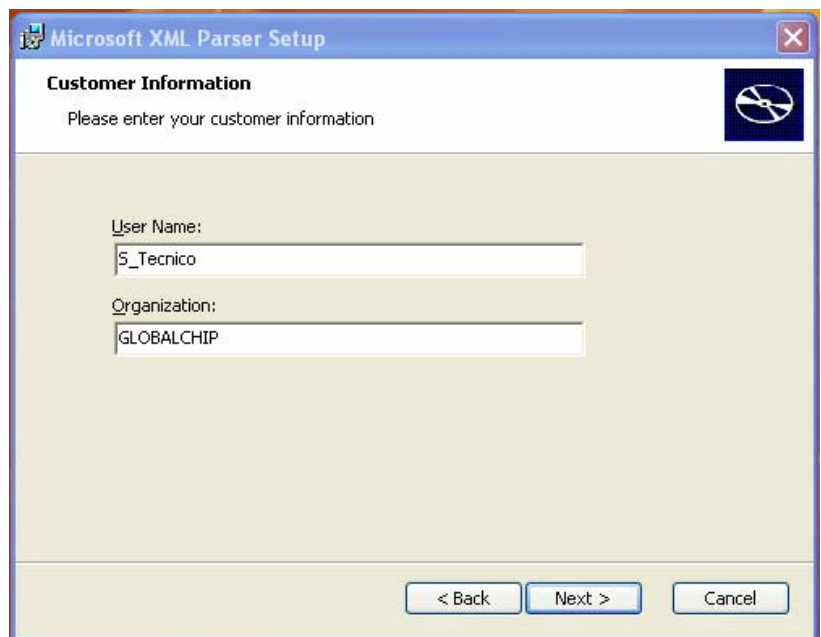
- a.- Haga doble clic sobre *Setup.exe*, para abrir el proceso de instalación, PowerPro interfaz de instalación.
- b.- Haga clic en botón "Next".
- c.- De acuerdo con el acuerdo de derecho de autor, haga clic en botón "Next"
- d.- Seleccione la ruta de instalación. La ruta de instalación por defecto es en la unidad D: \ Hollysys \ PowerPro, no se recomienda modificar. Si es necesario modificarla, haga clic en botón "Examinar" para elegir otro camino. Haga clic en botón "Next" para iniciar la instalación.

2.- Una vez finalizada la instalación de la aplicación, se iniciará automáticamente la segunda parte, que consiste en la instalación de las librerías necesarias para utilizar Microsoft XML 3.0 en PowerPro.



- a.- En el asistente de instalación, haga clic en botón "Aceptar".
- b.- De acuerdo con el acuerdo de derecho de autor, haga clic en botón "Next".

3.- Rellene los campos de, nombre de usuario y nombre de su Organización, haga clic en botón "Next" y confirme la instalación sobre el botón "Install".

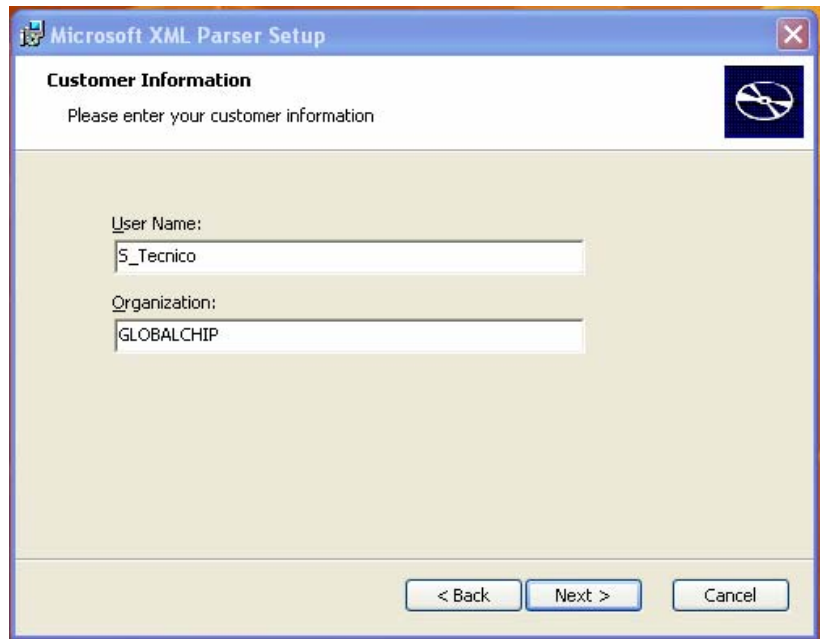


- a.- En el asistente de instalación, confirme la instalación sobre el botón "Install".
- b.- Una vez completada la instalación de Microsoft XML 3.0, haga clic en botón "Finish".

4.- A continuación, se mostrara la interfaz de instalación MSXML 3,0 SP4, haga clic en botón "Aceptar".

a.- En el asistente de instalación, confirme la instalación sobre el botón "Next".

b.- De acuerdo con el acuerdo de derecho de autor, haga clic en botón "Next"



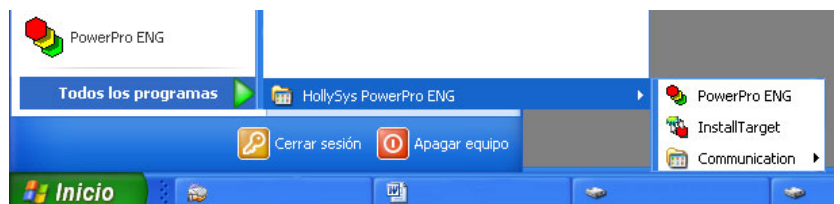
c.- Rellene los campos de, nombre de usuario y nombre de su Organización, haga clic en botón "Next" y confirme la instalación sobre el botón "Install".

d.- Una vez completada la instalación de Microsoft XML 3.0 SP4, haga clic en botón "Finish".

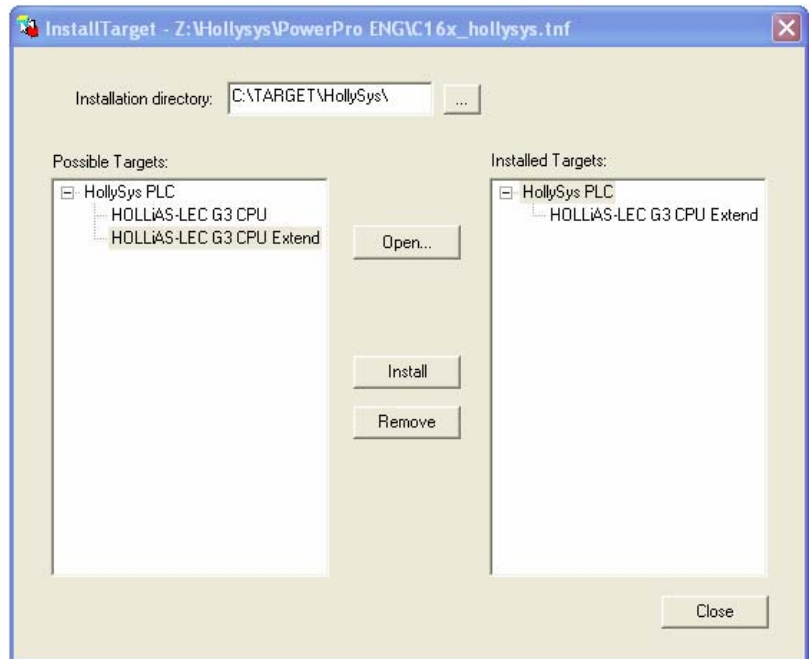
5.- Una vez finalizada de forma correcta la instalación completa de PowerPro, haga clic en botón "Finish".



6.- Finalizada la instalación, se nos crearan 3 iconos en el menú de inicio

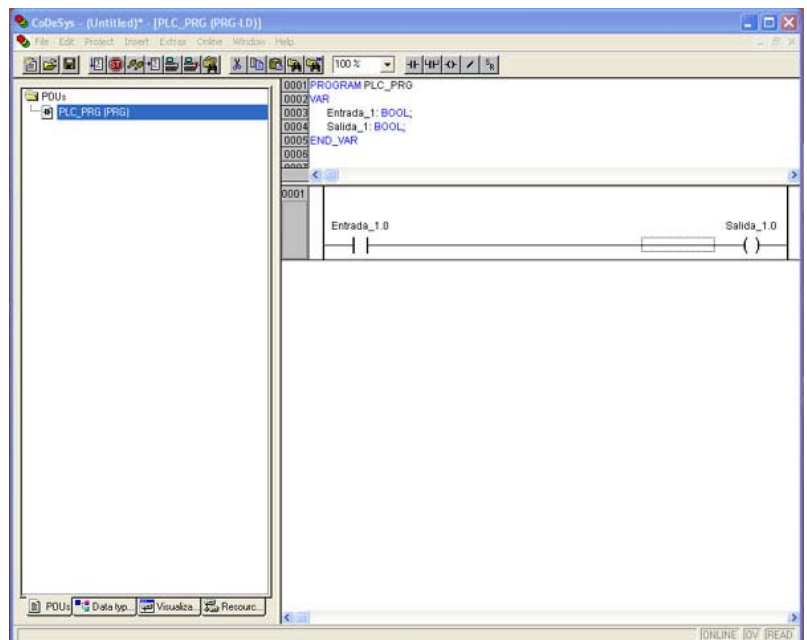


7.- Importante, como PowerPro es un software común para un gran número de equipos industriales, antes de ejecutar la aplicación de PowerPro, tenemos que configurar con que tipo de hardware vamos a trabajar, el "TARGET".



- a.- En el Menú de Inicio, hacer "Clic" sobre el icono **InstallTarget**.
- b.- En la nueva ventana, pulsar el botón de **OPEN**.
- c.- Seleccionar con doble clic el fichero de información "TARGET", "C16x_hollysys.tnf".
- d.- Elegir en la ventana Posible TARGET → HOLLiAS-LEC G3 CPU Extend.
- e.- Pulsar el botón **INSTALL** y **CLOSE**.

9.- Globalchip, proporciona un proyecto base comprimido, que incluye configuración de hardware, variables de trabajo predefinidas, FB's de control domótico (control de luces, motorizaciones, Control de Transceivers, etc). Está implemento el protocolo de comunicación GL-10, una configuración de visualización estados de una vivienda tipo y muchas más funciones, listas para transferir al PLC e iniciar la puesta en marcha.



- a.- Descomprima el proyecto, la ruta de instalación por defecto es en la unidad D: \ Hollysys \ PowerPro \ GL10, no se recomienda modificar esta ruta, se perderán los enlaces con los archivos, librerías e imágenes del proyecto.

2.22 Arquitectura de la programación

En la arquitectura software se identifican los siguientes elementos:

Task: Identifica la unidad mínima de planificación dentro de un recurso. Las tareas se asocian con un recurso particular y se considera que se ejecutarán bajo el control de ese recurso. A cada tarea se le asigna una prioridad y un periodo de ejecución. Como se aprecia en la figura, el estándar IEC 61131 permite asignar los programas y bloques funcionales diseñados a diferentes tareas para ajustar los periodos de su ejecución.

El elemento principal de reutilización de código es el *POU* (Program Organisation Unit) que se diseña (programa) una vez y puede ser utilizado tantas veces como sea necesario y además en diferentes aplicaciones. Existen tres tipos de POU:

Function: Es el nivel más básico es el equivalente a las funciones en los lenguajes de programación de alto nivel como C, PASCAL, etc. Sólo tiene un parámetro de salida que es del tipo de la función. Como ejemplo de Functions pueden ser: ADD, ABS, SQRT ...

Function Block: Equivale a los métodos en los lenguajes de programación en los que se puede tener uno o más parámetros de salida. El estándar considera este tipo de POU como el más adecuado para garantizar la reutilización.

Program: conceptualmente es similar al Function Block. Añade una mayor capacidad de estructuración de la aplicación.

Los POUs pueden ser implementados en cualquiera de los cinco lenguajes que propone el estándar en su parte 3: diagramas de contacto, lista de instrucciones (IL), diagramas de bloques (FBD), texto estructurado y programación secuencial (SFC).

▪ Lenguajes de programación:

Literales:

Lista de instrucciones (IL).

Texto estructurado (ST).

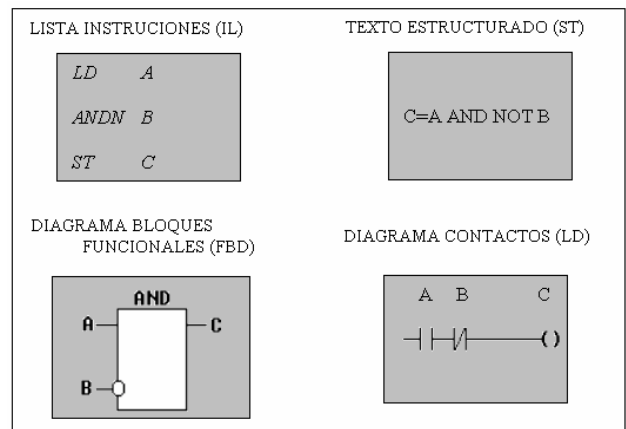
Gráficos:

Diagrama de contactos (LD).

Diagrama de bloques funcionales (FBD).

Secuenciales:

Programa secuencial (SFC).



2.23 Las Variables

Las variables permiten identificar los objetos de datos cuyos contenidos pueden cambiar, por ejemplo, los datos asociados a entradas, salidas o a la memoria del PLC. Una variable se puede declarar como uno de los tipos de datos elementales definidos o como uno de los tipos de datos derivados. De este modo se crea un alto nivel de independencia con el hardware, favoreciendo la reutilización del software.

La extensión de las variables está normalmente limitada a la unidad de organización en la cual han sido declaradas como locales. Esto significa que sus nombres pueden ser reutilizados en otras partes sin conflictos, eliminando una frecuente fuente de errores. Si las variables deben tener una extensión global, han de ser declaradas como globales utilizando la palabra reservada VAR_GLOBAL.

Pueden ser asignados parámetros y valores iniciales que se restablecen al inicio, para obtener la configuración inicial correcta.

Definición de tipos de variables:

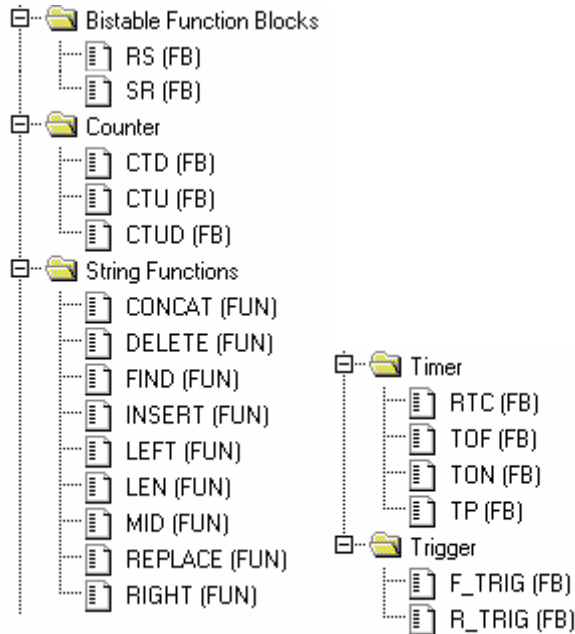
After Online command	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
Reset	-	X	-	X
Reset cold	-	-	-	-
Reset origin	-	-	-	-
Download	-	-	X	X
Online Change	X	X	X	X

x = el valor se mantendrá - = el valor se reiniciará

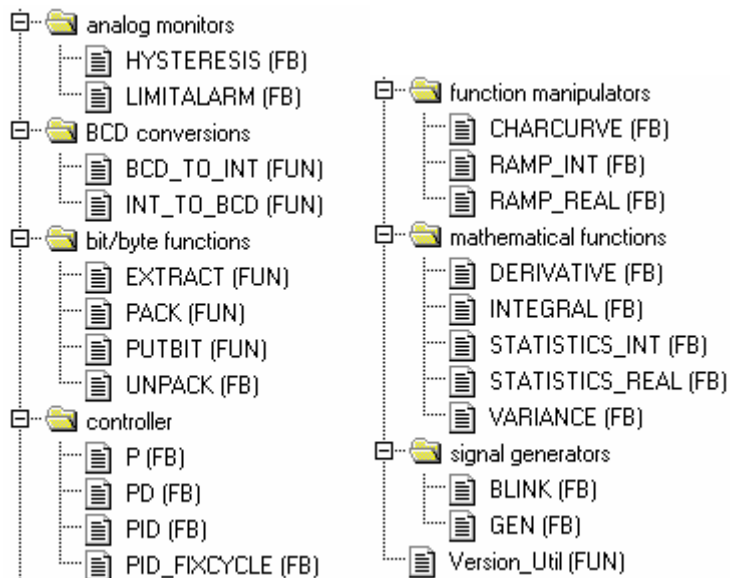
2.24 Librerías estándar de PowerPro

Están disponibles, múltiples librerías que se pueden incluir en el proyecto facilitando la programación, toda esta información está ampliada en el documento **GL10_G3_PLC_MANUAL.pdf**.

▪ Estandar.lib, funciones básicas generales.



▪ Util.lib y Util_no_real.lib, funciones generales.
























2.25 Librerías GL-10

- **Librerías estándar GL-10:**

<p>Hollysys_PLC_ANALOG.lib (Control sobre entradas y salidas analógicas)</p> <ul style="list-style-type: none"> Analog_IN (FB) Analog_OUT (FB) 	<p>Hollysys_PLC_DP_SLAVE.lib (Control protocolo MODBUS - DP)</p> <ul style="list-style-type: none"> DP_Slave (FB)
<p>Hollysys_PLC_COMM.lib (Control sobre el COM RS-232, posibilidad de trabajar con protocolo abierto)</p> <ul style="list-style-type: none"> COMM_RECEIVE (FB) COMM_SEND (FB) Reset_COMM_PRMT (FB) Set_COMM_PRMT (FB) 	<p>Hollysys_PLC_COMM2.lib (Control sobre el COM RS-485, posibilidad de trabajar con protocolo abierto)</p> <ul style="list-style-type: none"> COMM2_RECEIVE (FB) COMM2_SEND (FB) Reset_COMM2_PRMT (FB) Set_COMM2_PRMT (FB)
<p>Hollysys_PLC_HDRTC.lib (Control sobre el Reloj en tiempo real del hardware)</p> <ul style="list-style-type: none"> Get_HD_RTC (FB) Set_HD_RTC (FB) Set_HD_RTC_X (FB) 	<p>Hollysys_PLC_HDRTCALM.lib (Control sobre el reloj de alarmas)</p> <ul style="list-style-type: none"> Get_HDRTC_ALM (FB) Set_HDRTC_ALM (FB)
<p>Hollysys_PLC_PTContr.lib (Control sobre los 2 potenciómetros de ajuste, disponibles en el frontal de la CPU)</p> <ul style="list-style-type: none"> PTContr_0 (FB) PTContr_1 (FB) 	<p>Hollysys_PLC_IO.lib (Control especial sobre entradas y salidas)</p> <ul style="list-style-type: none"> OutPut_Bit (FB) Set_INT_OutPut (FB)
<p>Hollysys_PLC_Cnvt.lib (Modos de conversión)</p> <ul style="list-style-type: none"> E_H (FB) H_E (FB) 	<p>Hollysys_PLC_Math.lib (Creación de números aleatorios)</p> <ul style="list-style-type: none"> Rand (FB)
<p>Hollysys_PLC_EtherNet.lib (Control sobre el módulo de ampliación TCP/IP (MODBUS))</p> <ul style="list-style-type: none"> EtherNet_TCP (FB) 	

▪ **Librerías extendidas GL-10:**

<p>Hollysys_PLC_Ex.lib (Funciones de control sobre el Hardware)</p>	<p>Hollysys_PLC_Ex_CT.lib (Contadores controlados por interrupciones externas)</p>
<ul style="list-style-type: none">  Get_Local_Address (FB)  HD_WDT_Reset (FB)  PDT (FB)  Set_Local_Address (FB) 	<ul style="list-style-type: none">  HD_CTU_T7 (FB)  HD_CTUD_T2 (FB)  HD_CTUD_T3 (FB)  HD_CTUD_T4 (FB)
<p>Hollysys_PLC_Ex_DCT.lib (Contadores ascendentes / descendentes controlados por interrupciones externas)</p>	<p>Hollysys_PLC_Ex_TIMER.lib (Temporizadores controlados por interrupciones)</p>
<ul style="list-style-type: none">  HD_DCTUD_T2 (FB)  HD_DCTUD_T3 (FB)  HD_DCTUD_T4 (FB) 	<ul style="list-style-type: none">  HD_Clear_T7 (FB)  HD_STOP_T7 (FB)  HD_TIMER_T7 (FB)
<p>Hollysys_PLC_Ex_DCT32.lib (Contadores ascendentes / descendentes de 32 bits controlados por interrupciones externas)</p>	<p>Hollysys_PLC_Ex_ExINT.lib (Control por Interrupción externa)</p>
<ul style="list-style-type: none">  HD_DCTUD32_T3 (FB) 	<ul style="list-style-type: none">  Fast_ExINT (FB)  Fast_ExINT_E (FB)
<p>Hollysys_PLC_Ex_PT.lib (Modulación por ancho de pulso)</p>	<p>Hollysys_PLC_Ex_PTRun.lib (Modulación por ancho de pulso)</p>
<ul style="list-style-type: none">  PTO_PWM0 (FB)  PTO_PWM1 (FB) 	<ul style="list-style-type: none">  PTO_PWM0_Run (FB)  PTO_PWM1_Run (FB)

2.26 Juegos de instrucciones:

Juego de instrucciones disponibles en el software de programación PowerPro, toda la información concerniente a la programación está disponible en la propia ayuda del programa o bien remitirse al manual (ingles) **GL10_G3_PLC_MANUAL.pdf**.

Basic Instructions			
Instruction Type	Instruction Description	Instruction Name	Library
Arithmetic Operators	Addition	ADD (FUN)	none
	Multiplication	MUL (FUN)	
	Subtraction	SUB (FUN)	
	Division	DIV (FUN)	
	Modulo Division	MOD (FUN)	
Assignment Operator	Assignment	MOVE (FUN)	none
Logical Operators	Bitwise AND	AND (FUN)	none
	Bitwise OR	OR (FUN)	
	Bitwise XOR	XOR (FUN)	
	Bitwise NOT	NOT (FUN)	
Bit-Shift Operators	Left-shift	SHL (FUN)	none
	Right-shift	SHR (FUN)	
	Rotation to the left	ROL (FUN)	
	Rotation to the right	ROR (FUN)	
Selection Operators	Binary selection	SEL (FUN)	none
	Returns the greater of 2 values	MAX (FUN)	
	Returns the lesser of 2 values	MIN (FUN)	
	Limits the value range	LIMIT (FUN)	
	Select a value out of a group of values	MUX (FUN)	
Comparison Operators	Greater than	GT (FUN)	none
	Less than	LT (FUN)	
	Greater or equal	GE (FUN)	
	Lesser or equal	LE (FUN)	
	Equal	EQ (FUN)	
	Not equal	NE (FUN)	
Type Conversion Instructions	BOOL_TO conversions	BOOL_TO_<TYPE>(FUN)	none
	BYTE_TO conversions	BYTE_TO_<TYPE>(FUN)	
	DATE_TO conversions	DATE_TO_<TYPE>(FUN)	
	DINT_TO conversions	DINT_TO_<TYPE>(FUN)	
	DT_TO conversions	DT_TO_<TYPE>(FUN)	
	DWORD_TO conversions	DWORD_TO_<TYPE>(FUN)	
	INT_TO conversions	INT_TO_<TYPE>(FUN)	

Instruction Type	Instruction Description	Instruction Name	Library
Type Conversion Functions	WORD_TO conversions	WORD_TO_<TYPE>(FUN)	none
	REAL_TO conversions	REAL_TO_<TYPE>(FUN)	
	SINT_TO conversions	SINT_TO_<TYPE>(FUN)	
	STRING_TO conversions	STRING_TO_<TYPE>(FUN)	
	TIME_TO conversions	TIME_TO_<TYPE>(FUN)	
	TOD_TO conversions	TOD_TO_<TYPE>(FUN)	
	UDINT_TO conversions	UDINT_TO_<TYPE>(FUN)	
	UINT_TO conversions	UINT_TO_<TYPE>(FUN)	
	USINT_TO conversions	USINT_TO_<TYPE>(FUN)	
	Conversion from REAL to INT	TRUNC(FUN)	
Elementary Mathematical Functions	Returns the absolute value	ABS(FUN)	none
	Returns the square root	SQRT(FUN)	
	Returns the natural logarithm of a number	LN(FUN)	
	Returns the logarithm of a number in base 10	LOG(FUN)	
	Returns the exponential function	EXP(FUN)	
	Returns the sine	SIN(FUN)	
	Returns the cosine	COS(FUN)	
	Returns the tangent	TAN(FUN)	
	Returns the arc sine	ASIN(FUN)	
	Returns the arc cosine	ACOS(FUN)	
	Returns the arc tangent	ATAN(FUN)	
	Exponentiation	EXPT(FUN)	
Address Operators	Address of the operand	ADR(FUN)	none
	Content operator	^(FUN)	
	Returns the bit offset	BITADR(FUN)	
	Internal index for a POU	INDEXOF (FUN)	
	Number of bytes	SIZEOF (FUN)	
Calling Operators	Calling a function block or a program	CAL(FUN)	none
Initialization Operator	Initialize retain variables	INI(FUN)	none
Bit-string Operators	Concatenation of two strings	CONCAT(FUN)	Standard.lib
	Delete string	DELETE(FUN)	
	Find string	FIND(FUN)	
	Insert string	INSERT(FUN)	

Instruction Type	Instruction Description	Instruction Name	Library
	Left initial string of given size of string	LEFT(FUN)	
	String length	LEN(FUN)	
	Returns a partial string from within a string	MID(FUN)	
	Replace a partial string from a larger string with a third string	REPLACE(FUN)	
	Right initial string of given size of string	RIGHT(FUN)	
Library version check	Get library version	Version_Util(FUN)	Util.lib
Software version check	Get Software version message	SyslibGetVersion2300(FUN)	SysLibC16x.lib
Callback function	Register a definite callback	SysCallbackRegister(FUN)	SysLibCallback.lib
	Un-register a definite callback	SysCallbackUnregistrer(FUN)	
Check instruction	Check for out-of-range errors	CheckBounds(FUN)	Check.lib
	Check whether the divisor is 0 in BYTE division	CheckDivByte(FUN)	
	Check whether the divisor is 0 in WORD division	CheckDivWord(FUN)	
	Check whether the divisor is 0 in DWORD division	CheckDivDword(FUN)	
	Check whether the divisor is 0 in REAL division	CheckDivReal(FUN)	
	Check whether subrange variables of INT are over range	CheckRangeSigned(FUN)	
	Check whether subrange variables of Unsigned are over range	CheckRangeUnsigned(FUN)	
BCD Conversion	Convert a byte in BCD format into an INT value	BCD_TO_INT(FUN)	Util.lib
	Convert an INT value into a byte in BCD format	INT_TO_BCD(FUN)	
bit/byte functions	Extract the Nth bit of input	EXTRACT(FUN)	Util.lib
	Pack 8 bits into 1 byte	PACK(FUN)	
	Set Nth bit value	PUTBIT(FUN)	
	Unpack 1 byte into 8 bits	UNPACK(FB)	
Mathematical Functions	Local derivation	DERIVATIVE(FB)	Util.lib
	Integral	INTEGRAL(FB)	
	Min, Max, Average values in INT format	STATISTICS_INT(FB)	
	Min, Max, Average values in REAL format	STATISTICS_REAL(FB)	
	Variance	VARIANCE(FB)	
Controller	Proportion controller	P(FB)	Util.lib

Instruction Type	Instruction Description	Instruction Name	Library
	PD controller	PD(FB)	
	PID controller	PID(FB)	
	PID controller with fixed cycle time	PID_FIXCYCLE(FB)	
Signal generators	Simulate a turn signal	BLINK(FB)	Util.lib
	Generate some periodic functions	GEN(FB)	
SFC action control	SFC action control	SFCActionControl(FB)	iecsfc.lib
Function Manipulators	Map an input signal on a characteristic curve	CHARCURVE(FB)	Util.lib
	Limit the slope of a value to a certain value	RAMP_INT(FB)	
	Limit the slope of a value to a certain value	RAMP_REAL(FB)	
Analog Monitors	Hysteresis	HYSTERESIS(FB)	Util.lib
	Monitor whether the input is within a set range	LIMITALARM(FB)	
Bistable Function Block	Bistable function, set dominant	SR(FB)	Standard.lib
	Bistable function, reset dominant	RS(FB)	
Trigger	Rising edge detection	R_TRIG(FB)	Standard.lib
	Falling edge detection	F_TRIG(FB)	
Counter	Counter up	CTU(FB)	Standard.lib
	Counter down	CTD(FB)	
	Counter up down	CTUD(FB)	
Timer	Timer pulse	TP(FB)	Standard.lib
	Timer on delay	TON(FB)	
	Timer off delay	TOF(FB)	
	Real time clock	RTC(FB)	

Expansion Instructions			
Instruction Type	Instruction Description	Instruction Name	Library
Analog modules	Analog input	Analog_IN(FB)	Hollysys_PLC_Analog.lib
	Analog output	Analog_OUT(FB)	
RS232 port communication setting	RS232 free port communication parameter setting	Set_COMM_PRMT(FB)	Hollysys_PLC_Comm.lib
	Send RS232 Free Port Communication Data	COMM_SEND(FB)	
	Receive RS232 Free Port Communication Data	COMM_RECEIVE(FB)	
	Reset RS 232 setting	Reset_COMM_PRMT(FB)	
RS485 port communication setting	RS485 free port communication parameter setting	Set_COMM2_PRMT(FB)	Hollysys_PLC_Comm2.lib
	Send RS485 Free Port Communication Data	COMM2_SEND(FB)	
	Receive RS485 Free Port Communication Data	COMM2_RECEIVE(FB)	

	Reset RS 485 setting	Reset_COMM2_PRMT	
Profibus-DP module call	Profibus-DP slave module call	DP_Slave(FB)	Hollysys_PLC_DPSlave.lib
EtherNet module call	EtherNet module call	EtherNet_TCP(FB)	Hollysys_PLC_EtherNet.lib
Positive and negative action optional PID controller	PID controller	PID2(FB)	Hollysys_PLC_Util.lib
Modbus check instruction	Generate Modbus CRC code	Generate_CRC(FB)	Hollysys_PLC_Modbus_CRC.lib
Hardware real time clock setting	Set RTC (in DT data type)	Set_HD_RTC(FB)	Hollysys_PLC_HdRtc.lib
	Set RTC (in TP data type)	Set_HD_RTC_X(FB)	
	Get RTC date/time/day	Get_HD_RTC(FB)	
HDRTC alarm instruction	Get HDRTC alarm time/day	Get_HDRTC_ALM(FB)	Hollysys_PLC_HdRtcALM.lib
	Set HDRTC alarm time/day	Set_HDRTC_ALM(FB)	
HDRTC alarm of LM3104/5	Get HDRTC alarm time/day	Get_HDRTC_ALM(FB)	Hollysys_PLC_HdRtcALM_N.lib
	Set HDRTC alarm time/day	Set_HDRTC_ALM(FB)	
Multi-segment Pulse Transmission	Channel 1.1 Multi-segment Pulse Transmission	PTOCtrl_0(FB)	Hollysys_PLC_PTOCtrl.lib
	Channel 0.3 Multi-segment Pulse Transmission	PTOCtrl_1(FB)	
Immediate Output	Immediate Output	OutPut_Bit(FB)	Hollysys_PLC_IO.lib
	Set Interruption Immediate Output	Set_INT_OutPut(FB)	
Engineering Unit Conversion	Convert EU to Hexadecimal	E_H(FB)	Hollysys_PLC_Cnvt.lib
	Convert Hexadecimal to EU	H_E(FB)	
Generate A Random Number	Generate A Random Number	Rand(FB)	Hollysys_PLC_Math.lib
Modbus local address	Set Modbus local communication address	SET_LOCAL_ADDRESS (FB)	Hollysys_PLC_Ex.lib
	Get Modbus local communication address	GET_LOCAL_ADDRESS (FB)	
Analog Potentiometer	Get Analog Potentiometer value	POT(FB)	Hollysys_PLC_Ex.lib
Instruction Type	Instruction Description	Instruction Name	Library
System Watch-Dog Reset	System Watch-Dog Reset	HD_WDT_Reset(FB)	Hollysys_PLC_Ex.lib
Single-phase Counter	T2 High Speed Counter	HD_CTUD_T2(FB)	Hollysys_PLC_Ex_CT.lib
	T3 High Speed Counter	HD_CTUD_T3(FB)	
	T4 normal counter	HD_CTUD_T4(FB)	

	T7 High Speed Counter	HD_CTU_T7(FB)	
Diphase Counter	T2 High Speed Counter	HD_DCTUD_T2(FB)	Hollysys_PLC_Ex_DCT.lib
	T3 High Speed Counter	HD_DCTUD_T3(FB)	
	T4 normal counter	HD_DCTUD_T4(FB)	
32-bit High-Speed Counter	32-bit High-Speed Counter	HD_DCTUD32_T3(FB)	Hollysys_PLC_Ex_DCT32.lib
Interruption Timer	Interruption Timer	HD_TIMER_T7(FB)	Hollysys_PLC_Ex_TIMER.lib
	Clear and Download Timer	HD_CLEAR_T7(FB)	
	Stop Timer	HD_STOP_T7(FB)	
External Interruption	Fast External Interruption (fit for 28K memory space)	Fast_ExINT(FB)	Hollysys_PLC_Ex_ExINT.lib
	Fast External Interruption (fit for 120K memory space)	Fast_ExINT_E(FB)	
PTO/PWM Pulse Transmission Out	PTO/PWM Pulse Transmission Out	PTO_PWM0(FB)	Hollysys_PLC_Ex_PT.lib
		PTO_PWM1(FB)	
Pulse Accelerated and Decelerated Transmission	PTO/PWM Pulse Transmission (Accelerated and Decelerated)	PTO_PWM0_Run(FB)	Hollysys_PLC__EX_PTRun.lib
		PTO_PWM1_Run(FB)	

▪ Ejemplos:

ADD

Addition of variables of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

Two TIME variables can also be added together resulting in another time (e.g., $t\#45s + t\#50s = t\#1m35s$)

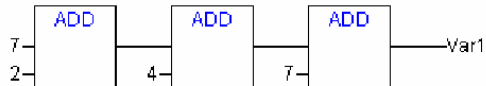
Example in IL:

```
LD 7
ADD 2,4,7
ST Var1
```

Example in ST:

```
var1 := 7+2+4+7;
```

Example in FBD:



MUL

Multiplication of variables of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

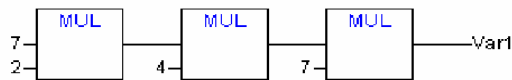
Example in IL:

```
LD 7
MUL 2,4,7
ST Var1
```

Example in ST:

```
var1 := 7*2*4*7;
```

Example in FBD:



SUB

Subtraction of one variable from another of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

A TIME variable may also be subtracted from another TIME variable resulting in third TIME type variable. Note that negative TIME values are undefined.

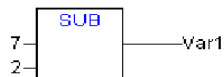
Example in IL:

```
LD 7
SUB 2
ST Var1
```

Example in ST:

```
var1 := 7-2;
```

Example in FBD:



DIV

Division of one variable by another of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

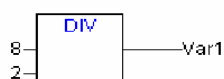
Example in IL:

```
LD 8
DIV 2
ST Var1 (* Result is 4 *)
```

Example in ST:

```
var1 := 8/2;
```

Example in FBD:



MOD

Modulo Division of one variable by another of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT. The result of this function will be the remainder of the division. This result will be a whole number.

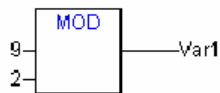
Example in IL:

```
LD 9
MOD 2
ST Var1 (* Result is 1 *)
```

Example in ST:

```
var1 := 9 MOD 2;
```

Example in FBD:

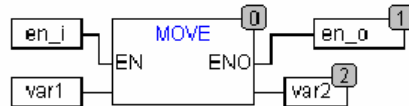


MOVE

Assignment of a variable to another variable of an appropriate type. As MOVE is available as a box in the graphic editors LD, CFC, there the (unlocking) EN/ENO functionality can also be applied on a variable assignment. In the FBD editor this is not possible however.

Example in CFC in conjunction with the EN/ENO function:

Only if en_i is TRUE, var1 will be assigned to var2.



AND

Bitwise AND of bit operands. The operands should be of the type BOOL, BYTE, WORD or DWORD.

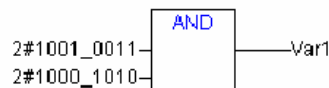
Example in IL:

```
Var1 BYTE
LD 2#1001_0011
AND 2#1000_1010
ST Var1 (* Result is 2#1000_0010 *)
```

Example in ST:

```
var1 := 2#1001_0011 AND 2#1000_1010
```

Example in FBD:



OR

Bitwise OR of bit operands. The operands should be of the type BOOL, BYTE, WORD or DWORD.

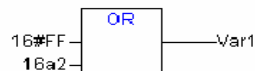
Example in IL:

```
var1 :BYTE;
LD 2#1001_0011
OR 2#1000_1010
ST var1 (* Result is 2#1001_1011 *)
```

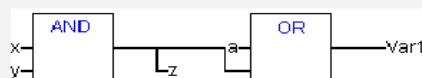
Example in ST:

```
Var1 := 2#1001_0011 OR 2#1000_1010
```

Example in FBD:



Note: If you have a program step in the SFC like the following



and if you use 68xxx- or C-code generators, please note the following: The allocation of the value of the second input variable at the OR operator module to variable z will not be executed ! This is due to the optimized processing in the SFC in case of value FALSE at the input variable.

2.27 Tipos de datos:

- **BOOL:**

Variables may be given the values TRUE and FALSE. 8 bits of memory space will be reserved.

- **INTEGER:**

BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, and UDINT are all integer data types Each of the different number types covers a different range of values. The following range limitations apply to the integer data types:

Type	Lower limit	Upper limit	Memory	space
BYTE	0	255	8	Bit
WORD	0	65535	16	Bit
DWORD	0	4294967295	32	Bit
SINT:	-128	127	8	Bit
USINT:	0	255	8	Bit
INT:	-32768	32767	16	Bit
UINT:	0	65535	16	Bit
DINT:	-2147483648	2147483647	32	Bit
UDINT:	0	4294967295	32	Bit

- **REAL / LREAL:**

REAL and **LREAL** are so-called floating-point types. They are required to represent rational numbers.

32 bits of memory space is reserved for REAL and 64 bits for LREAL.

Valid values for REAL: 1.175494351e-38 to 3.402823466e+38

Valid values for LREAL: 2.2250738585072014e-308 to 1.7976931348623158e+308

see also Chapter 10.11, REAL-/LREAL constants

- **STRING:**

A **STRING** type variable can contain any string of characters. The size entry in the declaration

Determines how much memory space should be reserved for the variable. It refers to the number of

Characters in the string and can be placed in parentheses or square brackets. If no size specification

is given, the default size of 80 characters will be used.

The string length basically is not limited in CoDeSys, but string functions only can process strings of 1

- 255 characters!

Example of a String Declaration with 35 characters:

```
str:STRING(35):='This is a String';
```

see also Chapter 10.11, Operands in CoDeSys, STRING constants

- **Time Data Types:**

The data types **TIME, TIME_OF_DAY** (abb. **TOD**), **DATE** and **DATE_AND_TIME** (abb. **DT**) are

handled internally like DWORD.

Time is given in milliseconds in TIME and TOD, time in TOD begins at 12:00 A.M.

Time is given in seconds in DATE and DT beginning with January 1, 1970 at 12:00 A.M.

See in the following the time data formats used to assign values for time constants:

- **Time constants:**

Always made up of an initial "t" or "T" (or "time" or "TIME" spelled out) and a number sign "#".

This is followed by the actual time declaration which can include days (identified by "d"), hours

(identified by "h"), minutes (identified by "m"), seconds (identified by "s") and milliseconds (identified

by "ms"). Please note that the time entries must be given in this order according to length (d before h

before m before s before m before ms) but you are not required to include all time increments.

Maximum value: 49d17h2m47s295ms (4194967295 ms)

- **Examples of correct TIME constants in a ST assignment:**

TIME1 := T#14ms;

TIME1 := T#100S12ms; (*The highest component may be allowed to exceed its limit*)

TIME1 := t#12h34m15s;

the following would be incorrect:

TIME1 := t#5m68s; (*limit exceeded in a lower component*)

TIME1 := 15ms; (*T# is missing*)

TIME1 := t#4ms13d; (*Incorrect order of entries*)

- **Date Constants:**

Beginning with a "d", "D", "DATE" or "date" followed by "#". You can then enter any date with format

Year-Month-Day. Possible values: 1970-00-00 to 2106-02-06.

- **Examples:**

DATE#1996-05-06

d#1972-03-29

- **Time_OF_DAY Constants:** for storing times of the day:

begin with "tod#", "TOD#", "TIME_OF_DAY#" or "time_of_day#" followed by a time with the format:

Hour:Minute:Second. Seconds can be entered as real numbers or you can enter fractions of a

second. Possible values: 00:00:00 bis 23:59:59.999.

- **Examples:**

TIME_OF_DAY#15:36:30.123

tod#00:00:00

- **DATE_AND_TIME Constants,** combination of date and the time of day:

begin with "dt#", "DT#", "DATE_AND_TIME#" or "date_and_time#". Place a hyphen after the date

followed by the time. Possible values: 1970-00-00-00:00:00 to 2106-02-06-06:28:15.

Examples:

DATE_AND_TIME#1996-05-06-15:36:30

dt#1972-03-29-00:00:00

2.28 Registros Internos de la CPU

Existe una serie de registros internos reservados para el sistema, necesarios para el correcto funcionamiento, ejecución de los programas cargados por el usuario y hardware instalado. Estos registros pueden ser consultados por el programa pero nunca deben sobrescribirse, esta circunstancia puede llevar a un mal funcionamiento del programa en ejecución.

Tamaño memoria de almacenamiento 8K (%MB0 a %MB8191):

%MB0 a %MB10 → Utilizado por las librerías de comunicación, puertos RS-232 y RS-485.

%MB20 a %MB30 → mapa del hardware instalado y estados internos.

%MB31 a %MB99 → Área de diagnósticos.

%MB100 a %MB699 → Reservado comunicación GL-10.

%MB700 a %MB8191 → Área libre para el usuario.