VisiLogic™

# SOFTWARE MANUAL

## VISILOGIC: COMMUNICATIONS

V230-21-G23 Rev: 3:00

UNITRONICS®

The information in this document reflects products at the date of printing. Unitronics reserves the right, subject to all applicable laws, at any time, at its sole discretion, and without notice, to discontinue or change the features, designs, materials and other specifications of its products, and to either permanently or temporarily withdraw any of the forgoing from the market.

All information in this document is provided "as is" without warranty of any kind, either expressed or implied, including but not limited to any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Unitronics assumes no responsibility for errors or omissions in the information presented in this document. In no event shall Unitronics be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever arising out of or in connection with the use or performance of this information.

# Table of Contents

# VisiLogic: Communications

# Communications

Vision controllers are capable of a broad variety of communcations.

Information regarding the communications features of specific Vision models is available from your local Unitronics distributor, or export@unitronics.com.

## Function Blocks

Note that VisiLogic offers function blocks to handle communications such as MODBUS, GPRS, SMS and more. Please refer to the VisiLogic: Function Blocks manual for details.

## COM Ports and Data Communications

Below is a general description of Vision communication features.

### Com Ports

| | |
|---|---|
| **Serial** | All Vision controllers comprise RS232 serial communication ports. RS232/RS485 adaptors are available by separate order. Certain models, such as the Vision120, support both RS232 and RS485. For details on communications hardware settings, refer to the User Guides and documentation supplied with relevant models. |
| **CANbus** | Separate CANbus ports are built into specific controller models. |
| **Ethernet** | Ethernet ports are available by separate order. |

| | |
|---|---|
| **Note** • | All ports can be used simultaneously. For example, a single controller may use one serial port to send messages to a modem via RS232, another port to communicate with a frequency converter, while the controller engages in communications via its CANbus port. |
| • | Standard programming cables do not provide connection points for pins 1 and 6. |

Initializing COM ports

- Serial and CANbus communication ports must be initialized in your control program using the COM Init FB, located on the FB's menu.
- The Ethernet port must be initialized using the Ethernet Card Init FB, located under Ethernet on the FB's menu.

Data Communications Options

Data Communications include all of the options shown below:

- CANbus
- Modems, Landline and GSM/GPRS
- GPRS
- Ethernet
- PROFIbus Slave
- DF1 Slave (Allen-Bradley)
- RS232
- RS485 Options

Communication FBs

- SMS messaging
- GPRS
- MODBUS (serial)
- MODBUS IP (Ethernet)
- Communications Protocol FB

PC-Vision communications

- PC-Modem Configuration
- Remote Access: Accessing a PLC via PC
- Accessing a Networked PLC via PC

SD Card Remote Access

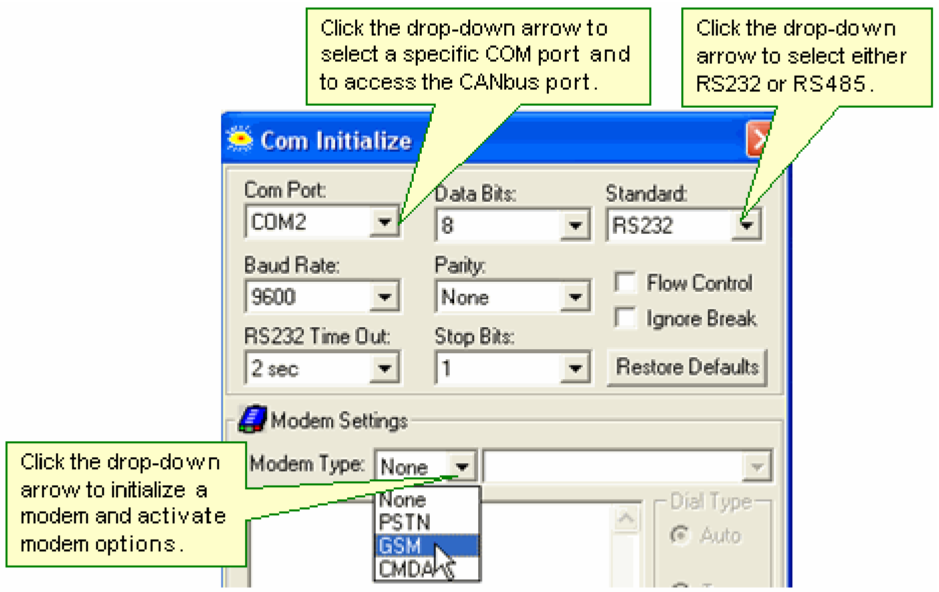
SD Card Explorer

## COM Port: Init

COM Init is located on the Com menu. Use this function block:

- To initialize serial communication port settings and enable the controller to communicate with networked controllers, using protocols such as MODBUS; or to communicate with external devices such as modems.
- To initialize the CANbus port.
- To synchronize port settings, enabling the controller to engage in inter-device communications via protocols such as MODBUS.

| | |
|---|---|
| **Notes** • | COM Init is generally performed once in a program. It is usually a power-up task, however a one-shot transitional contact may also be used. |
| • | All Vision controllers comprise RS232 serial ports. Some Vision controllers do not comprise RS485 ports. Check your Vision model's specifications.<br>To learn how to implement RS485 with different Unitronics' controllers, refer to RS485 Options. |
| • | Note that an Ethernet port is initialized via the Ethernet Card Init FB located on the FBs menu under Ethernet. |
| • | Where appropriate, use the system operands that are connected to the COM ports and that service communications. |



Click the drop-down arrow to select a specific COM port and to access the CANbus port.

Click the drop-down arrow to select either RS232 or RS485.

Click the drop-down arrow to initialize a modem and activate modem options.

Com Port: COM2
Data Bits: 8
Standard: RS232
Baud Rate: 9600
Parity: None
□ Flow Control
□ Ignore Break
RS232 Time Out: 2 sec
Stop Bits: 1
Modem Type: None

Specific uses of the COM Init FB are detailed in the topics listed below.

- Modems
- CANbus Networking

**Examples**

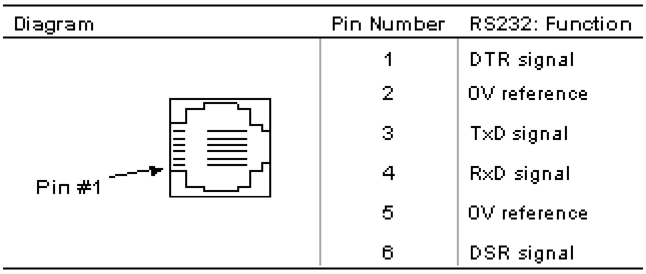
The applications below use the COM Init function. To locate application examples, select Examples from the Help menu.

- SMS messaging.vlp
- GPRS.vlp
- MODBUS Slave.vlp
- MODBUS Master.vlp

## RS232

All Vision controllers comprise RS232 serial communication ports. RS232/RS485 adaptors are available by separate order. Certain models, such as the Vision120, support both RS232 and RS485. For details on communications hardware settings, refer to the User Guides and documentation supplied with relevant models.

*RS232: Pinout*



| Diagram | Pin Number | RS232: Function |
|---|---|---|
| Pin #1 | 1 | DTR signal |
| | 2 | 0V reference |
| | 3 | TxD signal |
| | 4 | RxD signal |
| | 5 | 0V reference |
| | 6 | DSR signal |

## RS485 Options

The information in this topic is common to all Unitronics' controllers networked via RS485. Note that before you carry out any tasks associated with wiring, you must read and fully understand the safety guidelines.

### About RS485

RS485 is a balanced serial interface for the transmission of digital data, which enables you to create a multi-drop network containing up to 32 devices, or nodes

RS485 gives you 2 main advantages over RS232: longer cable lengths and greater immunity to noise. In comparison to RS232, RS485 uses lower voltage and differential signals. RS485 uses a differential voltage loop interface (balanced differential signal); differential data transmission reduces the effects of ground shifts and induced noise signals, even in an electrically noisy environment. The system is based on balanced circuits that rely on twisted-pair wires (A & B). Thus, the data conversion of logical 0 and 1 is made by converting the polarity of the two wires by reference to each other, instead of changing polarity of a single wire by reference to the "SG" (Signal Ground).

The noise immunity results from the fact that, when electromagnetic noise is induced over the differential signals, the same noise is induced on both signals. When the receiver subtracts the differential signals, the result is noise compensation.

The same 2 wires are used for transmitting and receiving; therefore, within RS485 networks, only one device can transmit while all of the other devices 'listen' (receive).

Unitronics' controllers offer different options for networking via RS485, according to the network series.

### Network Topology & Wiring

The network topology is multi-drop bus. Every RS485 network includes 2 types of nodes; node refers to every device that is physically connected to the network.

- End Nodes: The devices attached at both physical ends of the network, containing a network terminator.
- In-line Node: All devices connected to the network that are not end nodes.

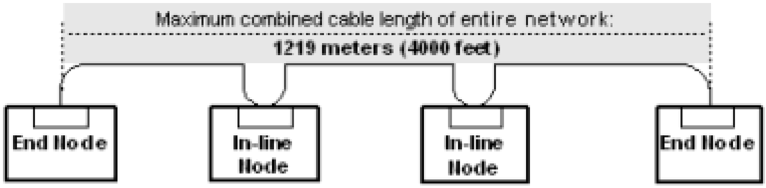
To enable a rapid rate of communication over relatively long distances, the wires function as transmission lines. For this reason, the end nodes of the network must contain network terminators for the purpose of impedance matching. The method for setting network terminators is described individually for each device.

#### RS485 Network Wiring

Use shielded twisted pair (STP) cables to network devices. Recommended cables types are:

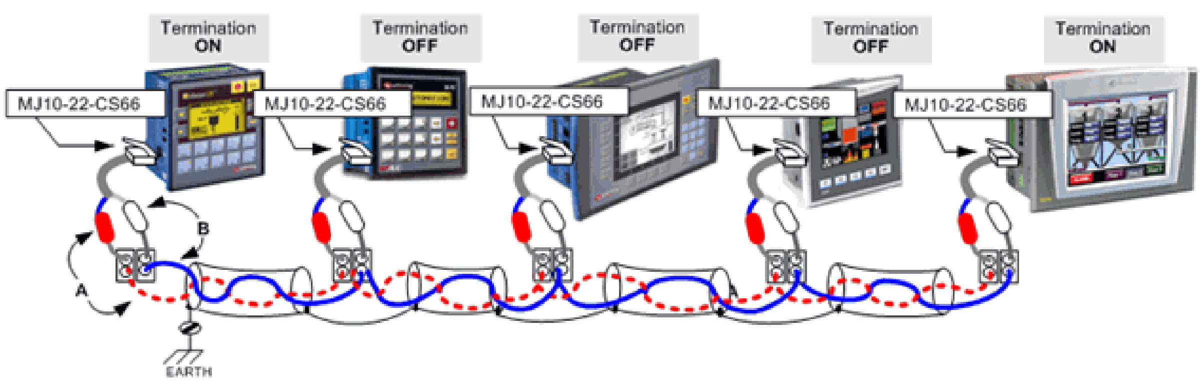
- Twinax cable, type H8106 • Control cable, type due 4001 (0.5mm2 , twisted pair)
- Twinax cable, type H3094 • Control cable, type V45551-F21-B5 (1.5mm2 , twisted pair)

The combined total length of all network cables cannot exceed 1219 meters, as shown below.



#### RS485 wiring considerations

- With the exception of the M90-19-R4, the RS485 signals are NOT isolated. If the controller is used with a non-isolated external device, avoid potential voltage that exceeds ± 10V. To avoid severely damaging the system, all non-isolated device ports should relate to the same 0Vsignal.
- Minimize the stub (drop) length leading from each device to the bus. The stub should not exceed 5 centimeters. Ideally, the main cable must be run in and out of the networked device as shown below. The MJ10-22-CS66 connector is compatible with all Unitronics controllers, and enable this to be easily accomplished.



Note that, in the case of older V2xxx models comprising a third RJ45 COM port, a MJ10-22-CS65 connector is required.

- Do not cross positive (A) and negative (B) signals. Positive terminals must be wired to positive, and negative terminals to negative.
- You must create network termination points by using the two end point devices integrated into your network. The method of creating termination points varies according to the controller series.

## RS485, by controller type

RS485 is implemented differently in Unitronics' controllers, according to model type. These options are summarized below.

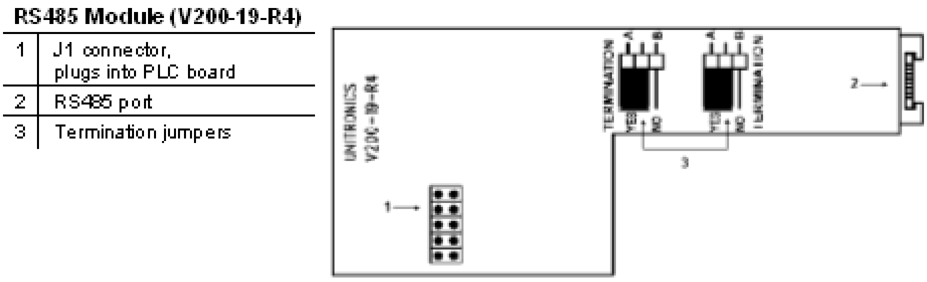
Vision controllers are programmed using VisiLogic software. When Vision controllers are networked via RS485, the COM ports must be initialized to the RS485 standard as explained in COM Port: Init.

In addition, you must assign a unique Unit ID number to each controller, as explained in the Help topic: Assigning a Unit ID number. Note that there is a range of ID numbers reserved for RS485, numbers 64-127.

### Vision 230/260/280

RS485 ports are available by separate order and easily installed.

Installation instructions are provided together with the module when it is ordered separately. The connector type is RJ-45. RS485 termination settings are determined via jumper.

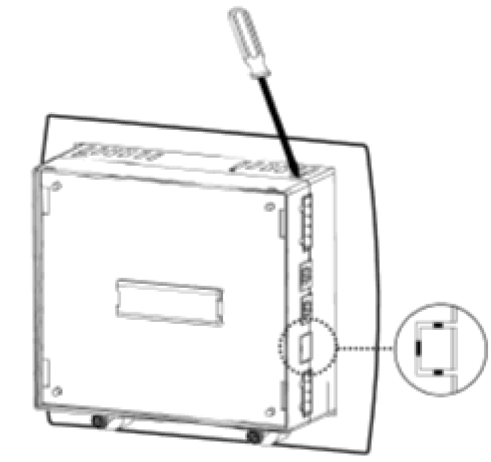

RS485 Module (V200-19-R4)

| | |
|---|---|
| 1 | J1 connector, plugs into PLC board |
| 2 | RS485 port |
| 3 | Termination jumpers |

### Network Termination Settings

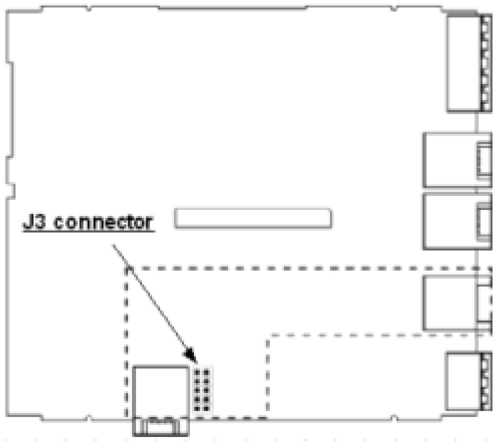
The jumper settings shown above determine whether the controller can function as an end device in a RS485 network. Note that the factory default setting is YES. If the OPLC is not a network end device, set both jumpers to NO.

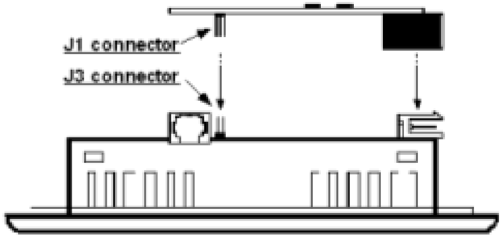
To open the controller in order to access the module and change the jumper settings, follow the relevant instructions listed below.



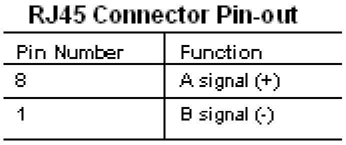
1. Turn power off before opening the controller.
2. If the controller has an installed Snap-in I/O module, remove it. Instructions are given in ‘Removing a Snap-in Module’ in the Vision User Guide.
3. Open the OPLC by inserting a screwdriver into the slots located on the sides of the controller as shown, then carefully prying the cover off.
4. The RS485 port’s location is covered by plastic. Remove the plastic covering by using a razor cutter to cut through the tabs.

4. Locate the J3 connector.



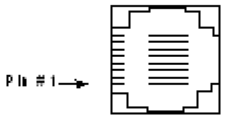


5. Install the module by placing the J1 connector (female) of the module onto the J3 connector (male) in the controller.
6. Make sure that the connection is secure.
7. Close the controller by snapping the plastic cover back in its place. If the module is correctly placed, the cover will snap on easily.
8. If required, reinstall the Snap-in Module.



RJ45 Connector Pin-out

| Pin Number | Function |
|---|---|
| 8 | A signal (+) |
| 1 | B signal (-) |



### Vision 120/ M91

Vision 120 series

Offers 2 serial communication ports.  Each port can be adapted to either the RS232 or RS485 standard, via jumpers located within the controller and VisiLogic software settings. Note that the ports are not isolated. The connector type is RJ-11. RS485 termination settings are determined via jumper.

M91 series

An M91 that contains an RS485/RS232 port has a part number that includes the number '4', for example: M91-19-4UN2.

RS485 communications are via an RJ-11-type serial communication port.  Each port can be adapted to either the RS232 or RS485 standard, via jumpers located within the controller.  The M91 is programmed using U90 Ladder software.
The port mode is determined by SI 64, Set COM Port Mode, as described in the U90 Ladder help topic: COM Port Mode: RS232/RS485 (M91 only).

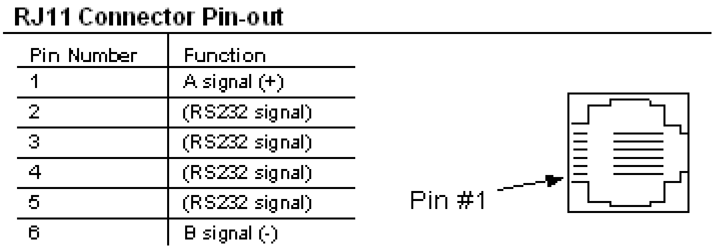
Note that the port is not isolated. RS485 termination settings are determined via jumper.

#### Vision 120/ M91 RS232/RS485 COM ports

The information below applies to both Vision 120 and M91 series controllers. The controllers in these series offer RJ-11-type serial communication ports. Each port can be adapted to either the RS232 or RS485 standard, via jumpers located within the controller. In the case of the Vision 120, appropriate VisiLogic program settings are also required.

### RJ-11 type port

The pinout below is of the RJ-11 type port, when the port is used for RS485.



RJ11 Connector Pin-out

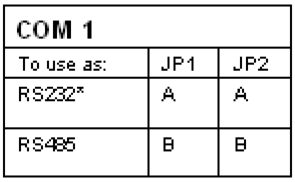
| Pin Number | Function |
|---|---|
| 1 | A signal (+) |
| 2 | (RS232 signal) |
| 3 | (RS232 signal) |
| 4 | (RS232 signal) |
| 5 | (RS232 signal) |
| 6 | B signal (-) |

**Note •** When a port is set to RS485, both RS232 and RS485 can be used simultaneously if flow control signals DTR and DSR are not used.

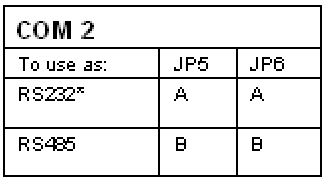
The ports are not isolated.  If the controller is used with a non-isolated external device, avoid potential voltage that exceeds ± 10V.
To avoid damaging the system, all non-isolated device ports should relate to the same ground signal.

#### RS232/RS485 Jumper Settings

Use the jumper settings shown below to change the functionality of the controller's COM port.



| COM 1 | | |
|---|---|---|
| To use as: | JP1 | JP2 |
| RS232* | A | A |
| RS485 | B | B |

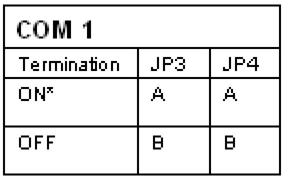


| COM 2 | | |
|---|---|---|
| To use as: | JP5 | JP6 |
| RS232* | A | A |
| RS485 | B | B |

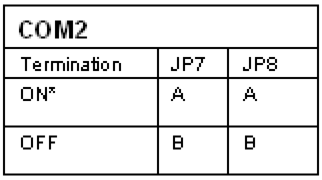
To open the controller and access the jumpers, refer to the instructions below.

### RS485 Network Termination Settings

The jumper settings shown below determine whether the controller can function as an end device in a RS485 network. Note that the factory default setting is ON. If the OPLC is not a network end device, set both jumpers to OFF.



| COM 1 | | |
|---|---|---|
| Termination | JP3 | JP4 |
| ON* | A | A |
| OFF | B | B |



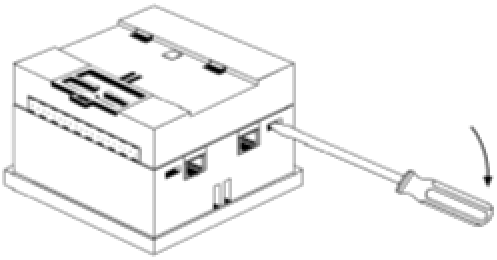
| COM2 | | |
|---|---|---|
| Termination | JP7 | JP8 |
| ON* | A | A |
| OFF | B | B |

* Default factory setting.

### Opening the Controller

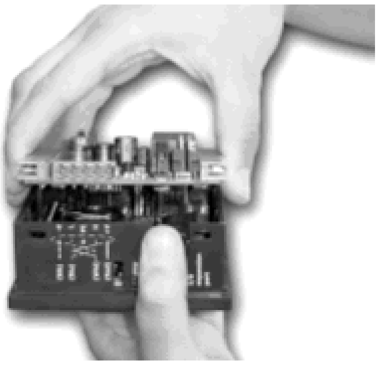
⚠

- Before opening the controller, touch a grounded object to discharge any electrostatic charge.
- Avoid touching the PCB board directly by holding the PCB board by its connectors.



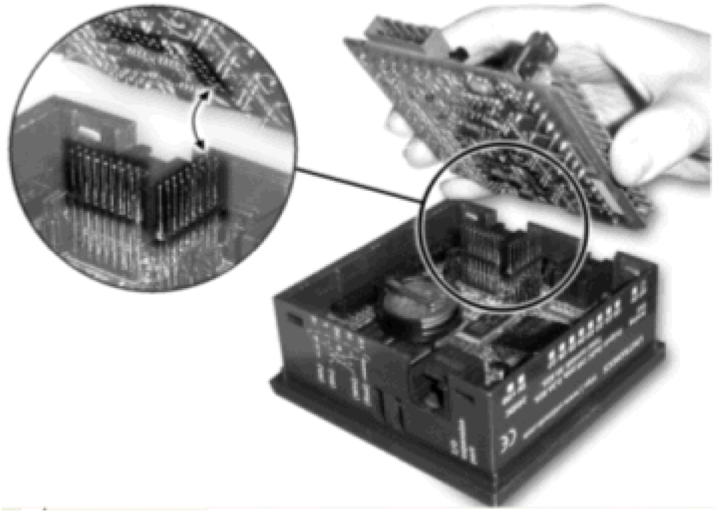
1. Turn power off before opening the controller.
2. Locate the 4 slots on the sides of the controller.
3. Using the blade of a flat-bladed screwdriver, gently pry off the back of the controller as shown.

4. Gently remove the top PCB board:
   - Use one hand to hold the top-most PCB board by its top and bottom connectors as shown.
   - With the other hand, grasp the controller, while keeping hold of the serial ports; this will keep the bottom board from being removed together with the top board.
   - Steadily pull the top board off.



5. Locate the jumpers shown in the figure to the left, then change the jumper settings as required.

5. Gently replace the PCB board as shown. Make certain that the pins fit correctly into their matching receptacle.
   - Do not force the board into place; doing so may damage the controller.
6. Close the controller by snapping the plastic cover back in its place. If the card is placed correctly, the cover will snap on easily.



### M90

These controllers are programmed via U90 Ladder.

RS485 communications are enabled via an external RS232/RS485 converter, such as Unitronics' M90-19-R4, which can be connected to the controller's RS232 port. No U90 Ladder software settings are required.

The M90-19-R4 RS485 port is isolated.

Note that since the M90-19-R4 is an external converter, it is also compatible with other devices, such as PCs.

### Vision / M91 RS485 Port Specifications

The specifications below apply to RS485 ports for all Vision and M91 controllers.

Input Voltage -7 to +12V differential max.

Cable type Shielded twisted pair, in compliance with EIA RS485

Cable length 1200m maximum (4000 feet)

Galvanic Isolation No

Baud rate 110 - 57600 bps

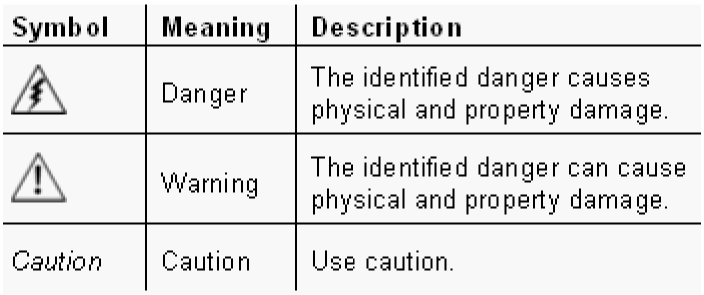
Complete specification for the M90-19-R4, which converts serial data from the RS232 standard to either RS422 or RS485, is located on the Setup CD. It is also available from your local Unitronics distributor. This module is recommended for use with M90 controllers and other devices.

User safety and equipment protection guidelines

This information is intended to aid trained and competent personnel in the installation of this equipment as defined by the European directives for machinery, low voltage, and EMC. Only a technician or engineer trained in the local and national electrical standards should perform tasks associated with the device’s electrical wiring.

Before using a Unitronics' product, it is the responsibility of the user to read and understand this document and any accompanying documentation.

Symbols are used to highlight information relating to the user's personal safety and equipment protection throughout this document. When these symbols appear, the associated information must be read carefully and understood fully.



| Symbol | Meaning | Description |
|---|---|---|
| ⚡ | Danger | The identified danger causes physical and property damage. |
| ⚠ | Warning | The identified danger can cause physical and property damage. |
| *Caution* | Caution | Use caution. |

- Under no circumstances will Unitronics be liable or responsible for any consequential damage that may arise as a result of installation or use of equipment, and is not responsible for problems resulting from improper or irresponsible use of Unitronics devices.
- All examples and diagrams shown are intended to aid understanding. They do not guarantee operation.
- Unitronics accepts no responsibility for actual use of a product based on these examples.
- Only qualified service personnel should open a device or carry out repairs.
- Please dispose of this product in accordance with local and national standards and regulations.

| | |
|---|---|
| ⚡ | • Failure to comply with appropriate safety guidelines can result in severe personal injury or property damage. Always exercise proper caution when working with electrical equipment. |
| ⚠ | • Check the user program before running it.<br>• Do not attempt to use a device with parameters exceeding permissible levels.<br>• Install an external circuit breaker and take appropriate safety measures against short-circuiting in external wiring.<br>• To avoid damaging the system, do not connect or disconnect a device when the power is on.<br>• Do not touch live wires.<br>• Double-check all the wiring before turning on the power supply. |

## About Modems

Unitronics' controllers can be hooked up to PSTN (landline), or GSM/GPRS modems via the RS232 COM port. Unitronics provides kits that comprise modems that have been tested by Unitronics and are supported for use with Vision, Jazz, and M90/91 PLCs.

**Before** you can use modems in your application, **you must use Modem Services to initialize both the PC and PLC-side modems.** This process is referred to as **'Prepare Modem'**.

### Modem services

Modem Services is located on the Connections menu.

To use Modem Services, connect the modem to a PC, using the **cable supplied by the modem manufacturer**. You can then initialize the modem.

Once you have connected initialized modems to your PC and PLC, you can use Modem Services to establish communications with a remote PLC.

### PC-side Modems

You can use a PC modem to access a remote, modem-linked controller and perform any task, just as you would if the PLC were directly connected to your PC. For example, you can

- Dial a remote PLC modem and receive calls from a PLC.
- Download, upload, and edit the controller program via the modem connection.
- Run Online test mode.
- Download an OS to the controller via modem.
- Use OnLine test and Information Mode to troubleshoot problems in remote controllers and applications.
- Read and write data to/from controllers via Remote Access or Unitronics' communication .dll utilities.
- Receive and send SMS messages via SMS options.

### PLC-side modems

Via modem, a Vision controller can communicate data using:

- MODBUS (serial) commands.
- VisiLogic's Communication Protocol FB, which enables Vision controllers to communicate data with most external serial devices, such as bar-code readers and frequency converters, via their proprietary protocols.
- SMS messages. The SMS FB enables text messages, including variable data, to be sent and received via GSM modems.
- e-mail via GPRS (Enhanced Vision only).
- GPRS cellular network, to transmit IP packets of data.

### Modem Tips

**Notes** • The PC-modem cable is not the same type of cable used to connect between the controller and the modem. Ensure that the cable used to connect the PC to the modem provides connection points for all of the modem's pins.

- If calls are routed via a switchboard, note that the switchboard settings may interfere with communications. Consult with your switchboard provider.
- If, within the modem initialization strings, the parameter S7 is too short to permit the PLC's modem to answer, an error will result. For example, if this parameter is set as S7=30, the PC modem will wait for 3 seconds to receive an answer from the PLC's modem. If the PLC modem does not answer before the 3 seconds have elapsed, the S7=30 parameter is exceeded, and the PC modem returns the No Carrier error.
- PC/PLC modem communications: **Both** PC and controller must use the same type of modem: either **landline or GSM**. Internal PC modems **must** be used in conjunction with the driver provided by the modem's manufacturer.

- 
  - **Standard Vision Division**
    Controllers in this division can only support a single modem. You can connect a modem to any COM port. However, note that SB 184 TX Success and SB 185 TX Failed indicate message transmission status **regardless** of the actual COM port connected to the modem.
  - **Enhanced Vision Division**
    Controllers in this division can support a modem on each COM port. Each port is linked to a Succeed and Fail SB: COM1: SB 184 and SB 185, COM2: SB186 and SB 187, COM3: SB 188 and SB 189.

When working with V570 and V290 (color):

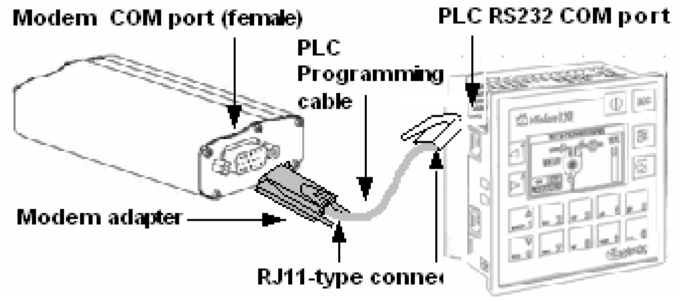
- Set the baud rate in both Modem Services and in COM Init to 57600. In addition, run the Prepare PLC Modem procedure with the baud rate set to 57600.
- Com Init:
  - Select Ignore Break
  - Time Out Reply: set to a minimum of 6 seconds.

- Known compatibility issue: Sony Ericsson Modems.
  Unitronics products are compatible with the following Sony Ericsson Modems:
  - Model GT47 R5xxxx and higher
  - Model GM29 R6xxxx and higher

  Unitronics cannot guarantee compatibility with other models, such as Sony Ericsson Modem model GM29 R4xxxx.

## Modem Connection and Pinouts

The following figure shows you how to connect a controller to a modem.



Note • The programming cable is a **4-wire cable**. The cable is included with the Vision all-in-one kits.

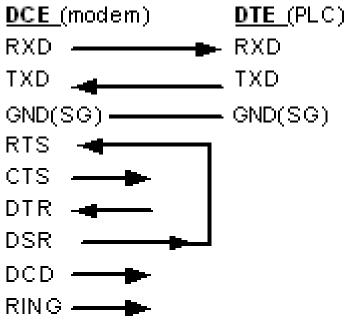
### Connecting a Controller to a Modem

- **Unitronics' modem kits**
  Use with a Unitronics PLC programming cable, as shown in the figure above. Kits contain a modem that is compatible with Unitronics controllers and related hardware. The Products section of the Unitronics web site contains kit descriptions and specifications.
- **Other modems**
  -Use a Unitronics PLC programming cable to connect the PLC to a modem adapter. You can order modem adapters from Unitronics.
  -Construct your own D-type to D-type connector cable to connect the programming cable to the modem's serial port, according to the pin-outs provided below.

### General Information: Modem to Controller Interface (DCE to DTE)

The next figure shows the interface between the Data Communications Equipment (DCE; the modem) and the Data Terminal Equipment (DTE; the controller or PC). The arrows show the direction of data flow. Note that:

- Transmitted data (TXD) is input to the modem, output from the PLC
- Received data (RXD) is input to the PLC, but output from the modem



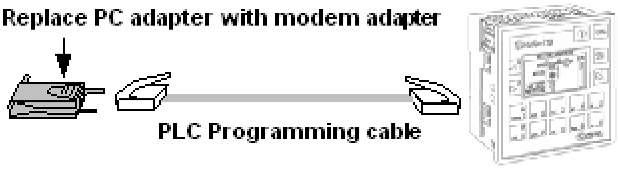
Note • Unitronics controllers do not support all the control lines.

- Modems should be initialized via the Modem Services> Prepare Modem procedure. This procedure:
  - Turns the DSR signal ON
  - Sets the appropriate PLC-modem communication parameters. These are:
    - 19200 bps, 8 bit, no parity, 1 stop bit
- Connecting DSR and RTS signals causes the modem to be always ready to transmit\receive data.

### Using Modem Kits

Unitronics' kits contain all of the elements you need to connect a controller to a modem using the appropriate PLC programming cable, as shown in the first figures at the beginning of this section.

Note that you must remove the PC adapter from the PLC programming cable and replace it with the modem adapter supplied in the kit.



In order to work with Unitronics controllers, you must initialize the modem via the procedure detailed in the Prepare PLC Modem topic.

The following list shows modem adapters supplied with the kits and their pin-outs. For updated information, consult your Unitronics distributor.

Standard Landline and Siemens GSM/GPRS modems

| Modem Adapter MJ10-22-CS76 | | | | PLC Serial Port / Port Module | |
|---|---|---|---|---|---|
| **Adapter signals** | **D-Type 9 pin, male** | **RJ11** | | **RJ11** | **Controller signals** |
| DSR (out) + RTS (in) | 6+7 | 1 | | 6 | DSR (in) |
| GND | 5 | 2 | | 5 | GND |
| RXD (out) | 2 | 3 | | 4 | RXD (in) |
| TXD (in) | 3 | 4 | | 3 | TXD (out) |
| GND | 5 | 5 | | 2 | GND |
| DCD (out) | 1 | 6 | | 1 | DTR (out) |

**Note •** The cable connects RJ11 pins 2,3,4,5 to pins 5,4,3,2 respectively.

Sony Ericsson GM29 and Enfora GSM/GPRS modems

| Modem Adapter MJ10-22-CS72 | | | | PLC Serial Port / Port Module | |
|---|---|---|---|---|---|
| **Adapter signals** | **D-Type 9 pin, male** | **RJ11** | | **RJ11** | **Controller signals** |
| DSR (out) | 6 | 1 | | 6 | DSR (in) |
| GND | 5 | 2 | | 5 | GND |
| RXD (out) | 2 | 3 | | 4 | RXD (in) |
| TXD (in) | 3 | 4 | | 3 | TXD (out) |
| GND | 5 | 5 | | 2 | GND |
| DCD (out) + RTS (in) | 1+7 | 6 | | 1 | DTR (out) |

**Note** The cable connects RJ11 pins 2,3,4,5 to pins 5,4,3,2 respectively.

- |

Wavecom GSM/GPRS modems

| Modem Adapter MJ10-22-CS79 | | | | PLC Serial Port / Port Module | |
|---|---|---|---|---|---|
| **Adapter signals** | **D-Type 15 pin, male** | **RJ11** | | **RJ11** | **Controller signals** |
| DCD (out) | 1 | 1 | | 6 | DSR (in) |
| GND | 9 | 2 | | 5 | GND |
| RXD (out) | 6 | 3 | | 4 | RXD (in) |
| TXD (in) | 2 | 4 | | 3 | TXD (out) |
| GND | 9 | 5 | | 2 | GND |
| DSR (out) + RTS (in) | 7+12 | 6 | | 1 | DTR (out) |

**Note** •
The cable connects RJ11 pins 2,3,4,5 to pins 5,4,3,2 respectively.

- Wavecom modem kits do not contain the PC to modem cable, MJ10-22-CS32, that is required for the Wavecom Prepare Modem procedure. This cable is available by separate order.

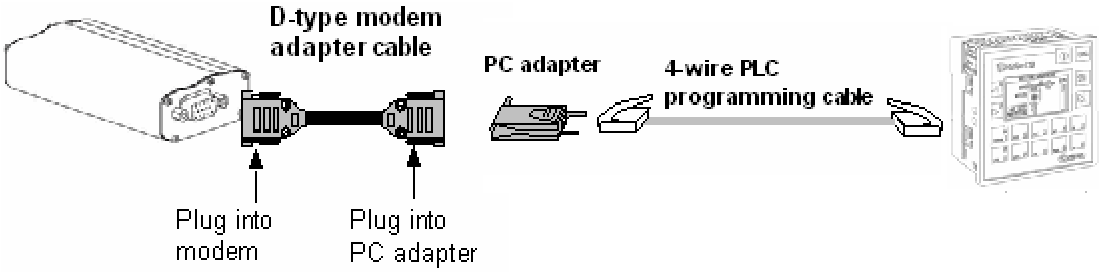
### Using a Unitronics Adapter

If you are using a modem from an independent source, you can order one of the modem adapters described in the preceding section from Unitronics, and use this adapter to connect your modem to the PLC via the PLC programming cable.

### Constructing Adapters

You can construct a D-type modem adapter yourself, using the appropriate pin-outs shown in the preceding section.

You can also construct a cable with 2 male D-type connectors, and then use it to connect a modem's serial port directly to the PC adapter on the PLC programming cable as shown in the next figure.



The tables below give the pin-outs of the programming cables, and show you an example of the pin connection you can use to construct a cable with 2 male D-type connectors to connect a PLC to a standard landline modem.

### Vision 4-wire Programming Cable

| PC -side Adapter MJ10-22-CS25 | | | | PLC-side Programming port | | |
|---|---|---|---|---|---|---|
| **PC signals** | **D-Type 9 pin, female** | **RJ11** | | **RJ11** | **Controller signals** | |
| DTR (out) | 4 | 1 | | 6 | DSR (in) | Unused |
| GND | 5 | 2 | | 5 | GND | |

| | | | | | | |
|---|---|---|---|---|---|---|
| TXD (out) | 3 | 3 | | 4 | RXD (in) | |
| RXD (in) | 2 | 4 | | 3 | TXD (out) | |
| GND | 5 | 5 | | 2 | GND | |
| DSR (in) | 6 | 6 | | 1 | DTR (out) | Unused |

**Note •** The 4-wire programming cable supplied with the controller connects RJ11 pins 2,3,4,5 to pins 5,4,3,2 respectively.

**Example**: D-type modem adapter cable to connect a Vision controller to a standard landline modem

| **Modem-side<br>(D-type 9-pin, male)** | | | **PLC-side<br>(D-type 9-pin, male)** | |
|---|---|---|---|---|
| DSR (out) + RTS (in) | 6+7 | | | |
| GND | 5 | | 5 | GND |
| RXD (out) | 2 | | 3 | RXD (in) |
| TXD (in) | 3 | | 2 | TXD (out) |
| GND | 5 | | 5 | GND |

## PLC-side Modems

Before integrating modems into your applications, check the section Modems-Tips & Cautions.

### How to enable a controller to communicate via landline or GSM/GPRS modem

Preparing the modem as described below initializes it so that it is compatible with Unitronics PLCs.

1. Connect the modem to a PC, using a **cable comprising the full RS232 pinout, either supplied in the modem kit or supplied by the modem manufacturer**.

   Note that using a cable that does not comprise the full RS232 pinout will cause the process to fail.

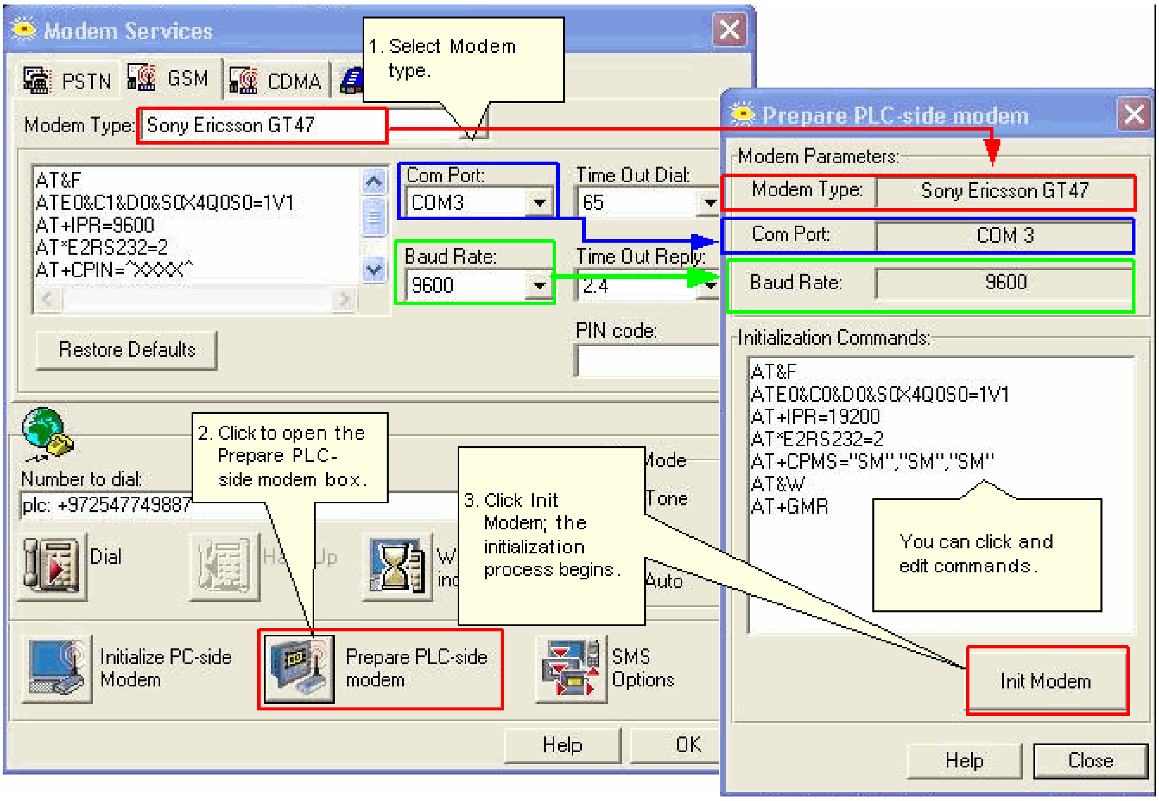
2. Prepare the PLC modem.
   a) Connect the modem to a PC, using the cable supplied by the modem manufacturer.
   b) Open Connection>Modem Services, and select the modem type. If required, you can edit other parameters:
      - Com Port, Baud Rate, Time Out, and Time-Out Reply: use the drop down boxes.

   Enhanced Vision Division:
   Recommended Baud Rate is 57600.

      - PIN code: click to enter the number.

3. Click the Prepare PLC-side Modem button; that dialog box opens.
4. If required, edit initialization commands by clicking in the field and entering text.

5. When all parameters are set, click the Init Modem button; the PC establishes communication with the modem and initializes it.



1. Select Modem type.

2. Click to open the Prepare PLC-side modem box.

3. Click Init Modem; the initialization process begins.

You can click and edit commands.

Modem Services

Modem Type: Sony Ericsson GT47

```
AT&F
ATE0&C1&D0&S0X4Q0S0=1V1
AT+IPR=9600
AT*E2RS232=2
AT+CPIN="XXXX"
```

Com Port: COM3

Time Out Dial: 65

Baud Rate: 9600

Time Out Reply: 2.4

PIN code:

Number to dial: plc: +972547749887

Prepare PLC-side modem

Modem Parameters:

Modem Type: Sony Ericsson GT47

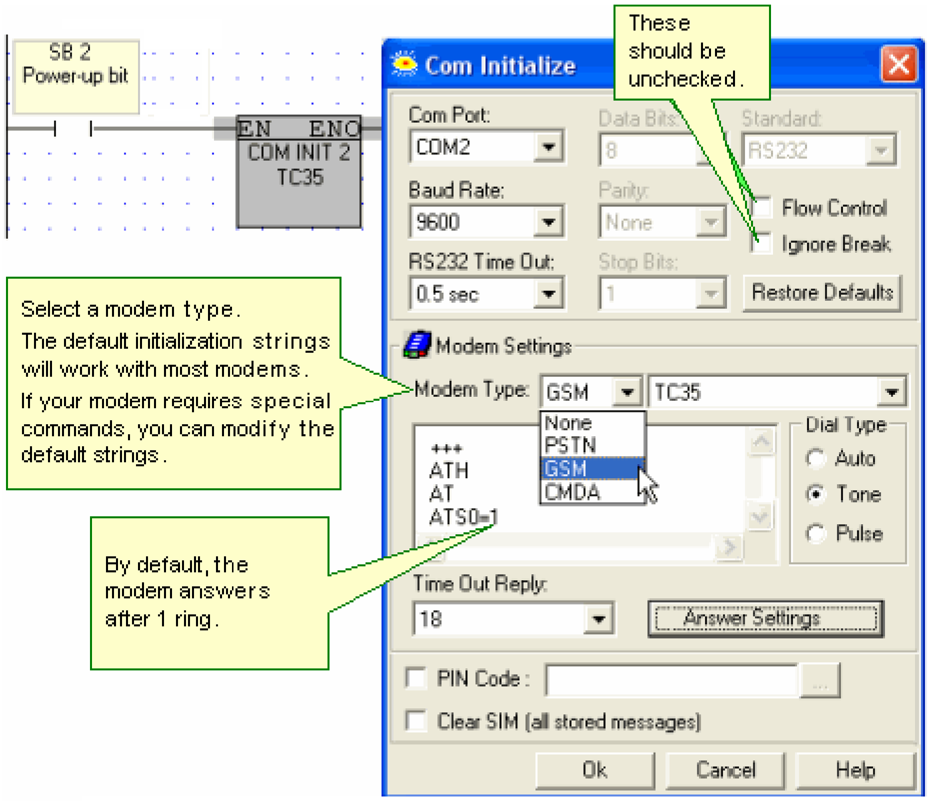
Com Port: COM 3

Baud Rate: 9600

Initialization Commands:

```
AT&F
ATE0&C0&D0&S0X4Q0S0=1V1
AT+IPR=19200
AT*E2RS232=2
AT+CPMS="SM","SM","SM"
AT&W
AT+GMR
```

6. Initialize the PLC port using a COM Init FB, located on the COM menu.



These should be unchecked.

Select a modem type.

The default initialization strings will work with most modems.

If your modem requires special commands, you can modify the default strings.

By default, the modem answers after 1 ring.

SB 2 Power-up bit

COM INIT 2 TC35

Com Initialize

Com Port: COM2

Data Bits: 8

Standard: RS232

Baud Rate: 9600

Parity: None

□ Flow Control

□ Ignore Break

RS232 Time Out: 0.5 sec

Stop Bits: 1

Modem Settings

Modem Type: GSM TC35

```
+++
ATH
AT
ATS0=1
```

Dial Type

○ Auto

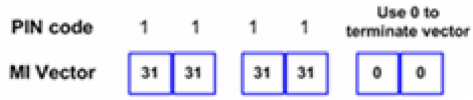
● Tone

○ Pulse

Time Out Reply: 18

□ PIN Code :

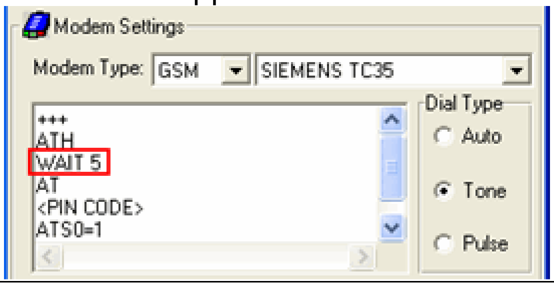
□ Clear SIM (all stored messages)

| **Parameter Name** | **Purpose** |
|---|---|
| Com Port | Select the physically linked Com Port. |
| Baud Rate | Set the baud rate in COM Init to 57600:<br>• When working with Enhanced Vision PLCs.<br>• When working with V120/230/260/280/290 (monochrome) together with modems Enfora orTC65.<br>In all other cases, 9600 is suitable. |
| RS232 Time Out | This is the time that the PLC will wait for the modem to reply. |
| Flow Control | This is generally left OFF. |
| Ignore break | When working with Enhanced Vision PLCs, select Ignore Break |
| Modem Settings | |
| **Modem Type** | Click to select the desired modem. |
| Initialization strings | Click to edit if required. |
| Dial Type | Set to Tone by default. |
| Time Out Reply | When working with Enhanced Vision PLCs, set a **minimum** of <u>**6**</u> seconds. |
| PIN Code | If your provider requires a PIN code, enter a constant number or link to MI. Note that the MI value must be in hexadecimal. For example, the PIN code 1111 requires a vector of 2 MIs containing the value 3131 3131. The vector should be terminated with 'null'.<br><br>PIN code 1 1 1 1 Use 0 to terminate vector<br>MI Vector 31 31 31 31 0 0 |
| Clear SIM | Select this to clear the SIM card when COM Init is activated. |

**Notes** • COM Init is generally performed once in a program. It is usually a power-up task, however a one-shot transitional contact may also be used.

- The initialization commands must match the commands used to initialize the PLC modem.
- You can cause the modem to delay between initialization commands

by typing WAIT, followed by the number of seconds; up to nine seconds is supported.



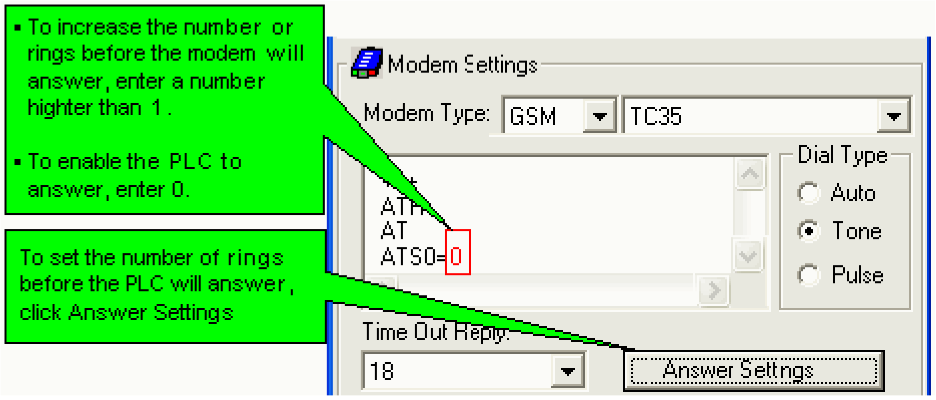
Modem Settings

Modem Type: GSM SIEMENS TC35

+++
ATH
WAIT 5
AT
<PIN CODE>
ATS0=1

Dial Type: ○ Auto ● Tone ○ Pulse

- Communications cannot flow through the port during initialization.
- The PLC cannot answer incoming calls when it is in bootstrap mode.
- The Answer Settings parameter, by default, enables the modem to answer incoming calls. This enables you to download an OS to the PLC via modem.
  However, you can enable the PLC to answer instead. To do this, locate the initialization string ATS0=**1** and enter 0( ATS0=**0**).
  When ATS0=0, an OS cannot be downloaded.



- To increase the number or rings before the modem will answer, enter a number highter than 1.
- To enable the PLC to answer, enter 0.

To set the number of rings before the PLC will answer, click Answer Settings

Modem Settings

Modem Type: GSM TC35

ATH
AT
ATS0=0

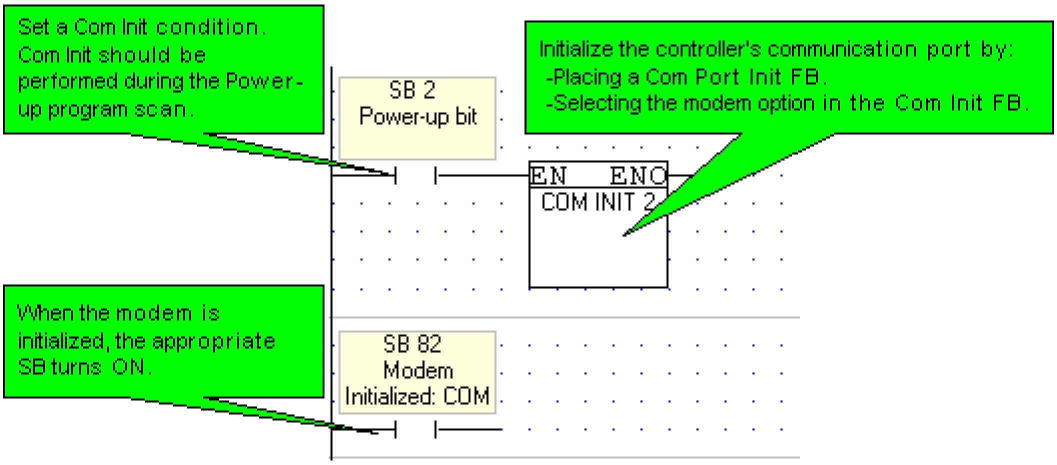
Dial Type: ○ Auto ● Tone ○ Pulse

Time Out Reply: 18

1. Build a Ladder program containing the correct conditions and elements.

**Note** • Communications cannot flow through the port during initialization. To avoid conflicts in your program, use the COM Port initialization SBs 80-85**.**

- COM Init may take approximately a minute to complete.



Set a Com Init condition. Com Init should be performed during the Power-up program scan.

Initialize the controller's communication port by:
-Placing a Com Port Init FB.
-Selecting the modem option in the Com Init FB.

When the modem is initialized, the appropriate SB turns ON.

2. Download the application to the PLC.

3. Connect the modem to the PLC.

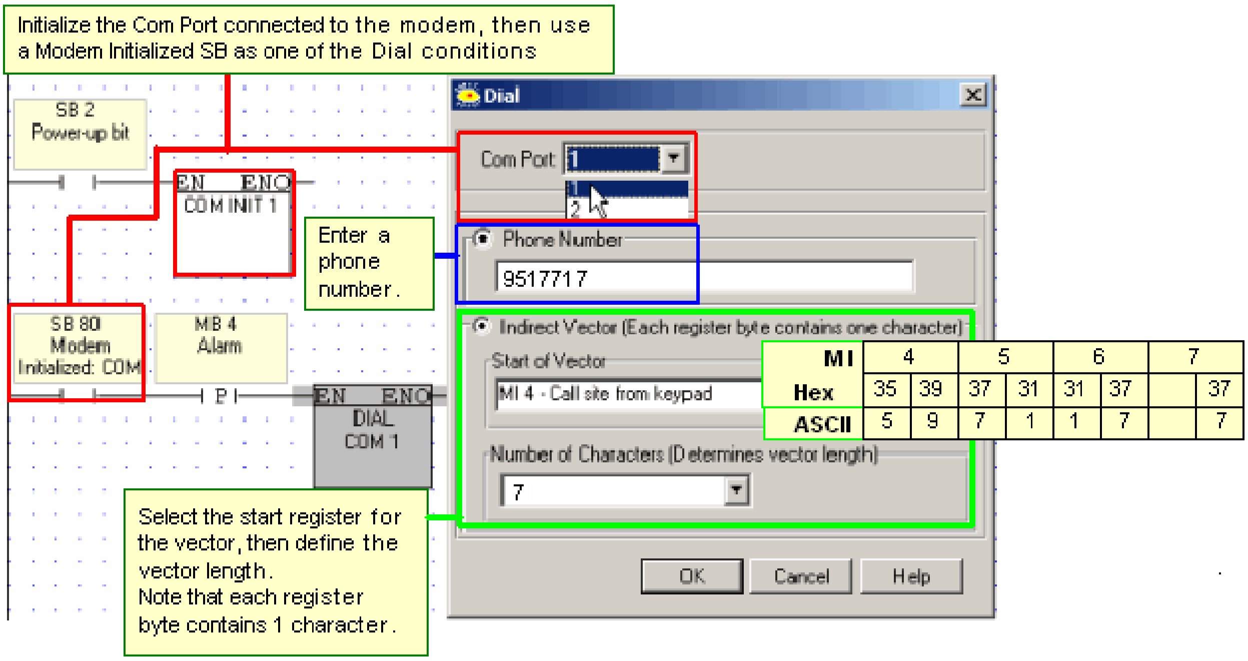
After the modem is enabled and successfully initialized by the PLC (SBs 80, 82, 84 turn ON), the controller can either be accessed via modem or can dial a remote modem to establish a data link.

**How to enable a PLC to dial a remote modem (Ladder)**

Landline modems

**Note •** In the conditions used to activate the Dial function, include the appropriate Modem Initialized System Bit: SB 80, SB 82, or SB 84.

For more information regarding Ladder conditions and other details, refer to the topic Dial and Hang-up.



Initialize the Com Port connected to the modem, then use a Modem Initialized SB as one of the Dial conditions

Enter a phone number.

Select the start register for the vector, then define the vector length.
Note that each register byte contains 1 character.

Dial

Com Port: 1

● Phone Number: 9517717

● Indirect Vector (Each register byte contains one character)

Start of Vector: MI 4 - Call site from keypad

Number of Characters (Determines vector length): 7

| MI | 4 | | 5 | | 6 | | 7 | |
|---|---|---|---|---|---|---|---|---|
| Hex | 35 | 39 | 37 | 31 | 31 | 37 | | 37 |
| ASCII | 5 | 9 | 7 | 1 | 1 | 7 | | 7 |

GSM modems

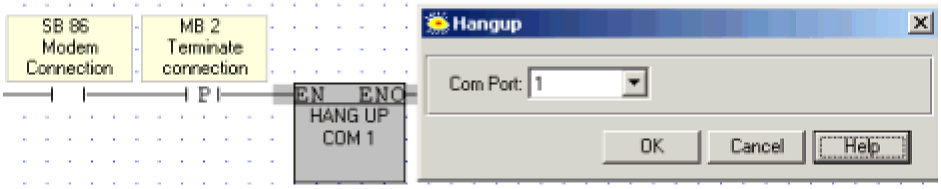
**Note •** SMS operations can conflict with applications that use the modem for other data communication processes. To prevent conflicts, use the Modem Busy (GSM) MB, and use an MB to indicate when the modem is in use by another data communications process.

For more information on SMS messaging, refer to the SMS topics.

## How to terminate the link--Hang-up

This enables you to use Ladder conditions to break the connection.

**Note •** Before activating Hang-up, check whether the connection exists, via the appropriate Modem Connection Status System Bit: SB 86, SB 87, or SB 88



Hangup

Com Port: 1

## PC-Side Modems (Modem Services)

Modem Services enables you to initialize modems for both PC and PLC. Once you have configured a PC-side modem, you can use a PC modem to access a remote, modem-linked controller and perform any task, just as you would if the PLC were directly connected to your PC. For example, you can:

- Download, upload, and edit the controller program via the modem connection.
- Run Online test mode.
- Download an OS to the controller via modem.
- Use OnLine test and Information Mode to troubleshoot problems in remote controllers and applications.
- Read and write data to/from controllers via Remote Access or Unitronics' communication .dll utilities.

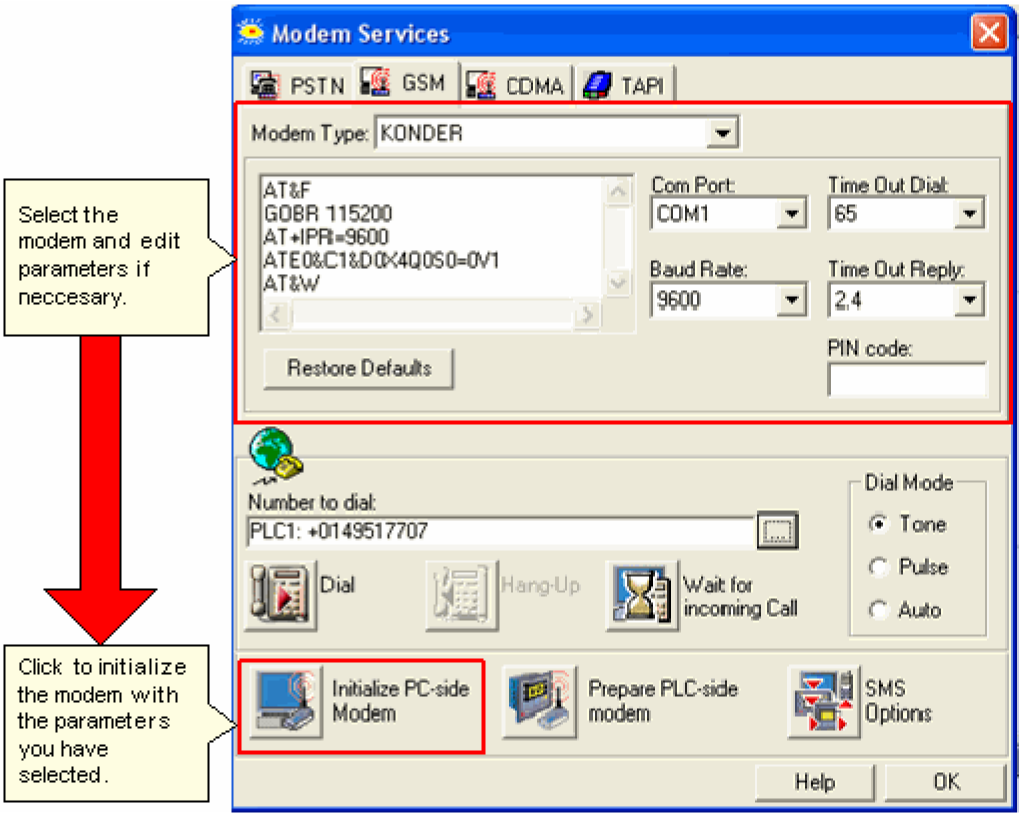
Before integrating modems into your applications, check the section Modems-Tips & Cautions.

### Initializing a PC-side Modem (Prepare Modem)

1. Connect the modem to a PC, using a **cable comprising the full RS232 pinout, either supplied in the modem kit or supplied by the modem manufacturer**.

   Note that using a cable that does not comprise the full RS232 pinout will cause the process to fail.

2. Open Modem Services from the Connection menu.
3. Select and enter the modem parameters:
   a) At the top of Modem Services, select a tab; the Modem Type selection box shows the options.
   b) Select the PC modem type; the initialization strings change accordingly. Selecting TAPI displays the settings of telephony devices that appear in Windows>Control Panel>Phone and Modem Options.
   c) If required, you can edit other parameters:
      - Initialization commands: click in the field and enter text.
      - Com Port, baud rate, Time Out, and Time-Out Reply: use the drop down boxes
      - PIN code: click to enter the number.
4. When all parameters are set, click the Initialize PC Modem button; the PC establishes communication with the modem and initializes it.



Select the modem and edit parameters if neccesary.

Click to initialize the modem with the parameters you have selected.

Modem Services

PSTN | GSM | CDMA | TAPI

Modem Type: KONDER

```
AT&F
G0BR 115200
AT+IPR=9600
ATE0&C1&D0X4Q0S0=0V1
AT&W
```

Com Port: COM1

Time Out Dial: 65

Baud Rate: 9600

Time Out Reply: 2.4

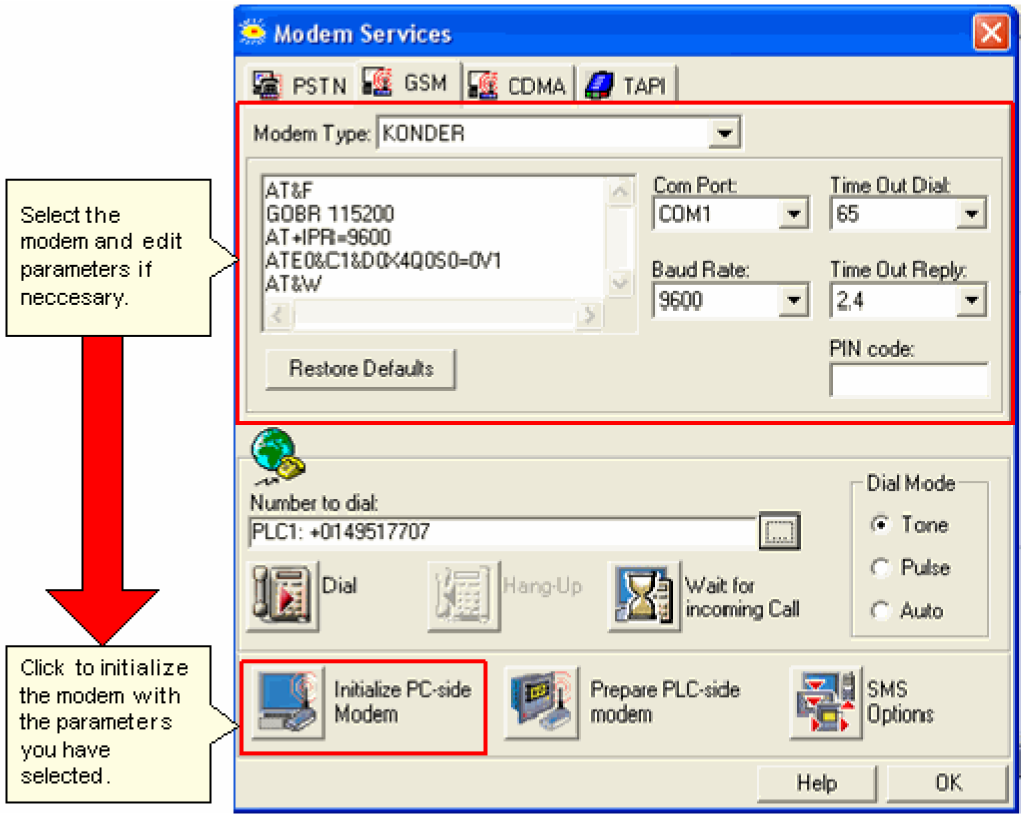
PIN code:

Number to dial: PLC1: +0149517707

Dial Mode: ● Tone ○ Pulse ○ Auto

### Modem Service Options

In addition to modem initialization, Modem Services offer other options:



```
AT&F
G0BR 115200
AT+IPR=9600
ATE0&C1&D0X4Q0S0=0V1
AT&W
```

PLC1: +0149517707

Select the modem and edit parameters if neccesary.

Click to initialize the modem with the parameters you have selected.

| Option | Description |
|---|---|
| **Dial & Hang-Up** | To dial:<br>1. Click the Number to Dial field; the Favorites list opens.<br>2. Enter or highlight the desired number.<br>3. Close Favorites, and click Dial.<br>4. To break the data link, click Hang-Up. |
| **Dial Mode** | Selecting Auto enables the modem to adapt to the signal provided by the telephony service operator. |
| **SMS Options** | This option is available if you have selected a GSM modem.<br>1. Select the destination number.<br>2. Enter the SMS text, then click Send SMS.<br>Note that an SMS can be used to cause the PLC to call the PC. |
| **Wait for Incoming Call** | Places the PC modem in auto-answer mode. |
| **Prepare PLC Modem** | Use this option to initialize a PLC-side modem. Full instructions are in the topic PLC-side Modems. |

### How to use the PC modem to access a PLC

1. Prepare and connect the PLC-side modem as described in the topic PLC-side Modems, in the section 'How to enable a controller to communicate via landline or GSM/GPRS modems.
2. Dial the PLC to establish the communication link:
    a) Favorites (Telephony). You can keep a list of frequently-used numbers in Favorites.
    b) Click a line to enter or select a number and description.
    c) To access outside lines, enter the access number required, a comma, then the phone number.

Click Dial to establish the data link, then enter OnLine mode.

You can now perform any task that can be performed via a direct PC-PLC connection.
Note that when the modem is connected, the Modem Connected icon appears on screen.

## Dial & Hang-up

These functions are located on the Com menu. Via the Ladder application, they enable a PLC connected to a modem to establish or terminate a data link to another remote modem.

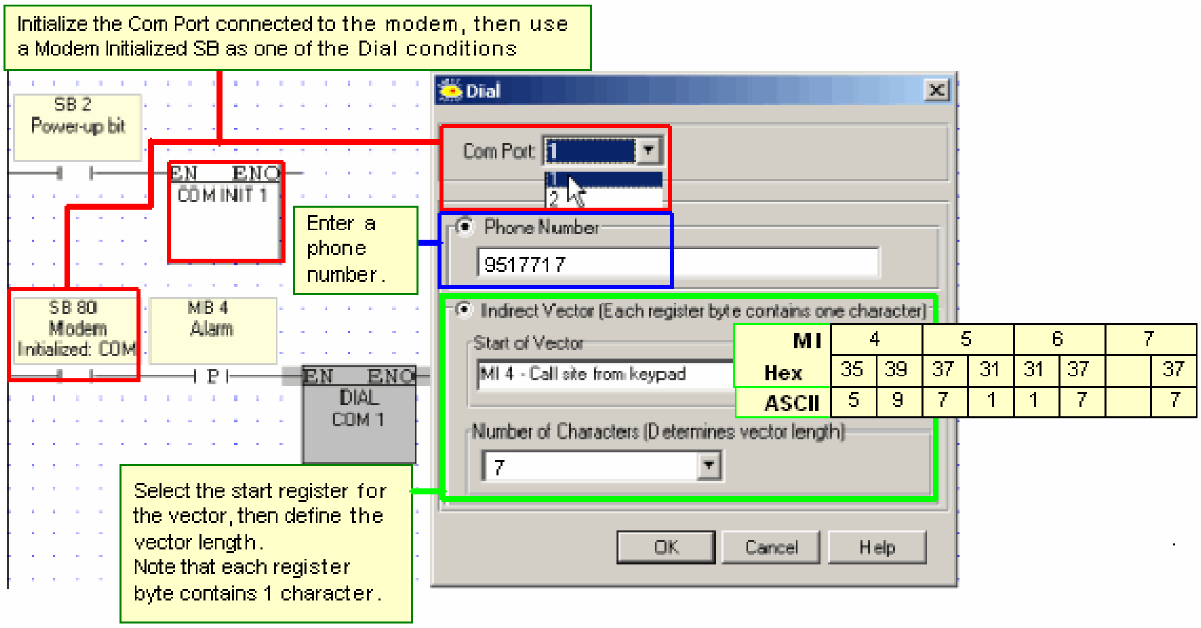
Before you dial, you must enable the Vision controller to communicate via modem.

Dial

This enables you to use Ladder conditions to dial a modem.

**Notes** • In the conditions used to activate Dial, include the appropriate Modem Initialized SB:  80, 82, or 84

• SMS operations can conflict with applications that use the modem for other data communication processes. To prevent conflicts, use the Modem Busy (GSM) MB, and use an MB to indicate when the modem is in use by another data communications process.



Initialize the Com Port connected to the modem, then use a Modem Initialized SB as one of the Dial conditions

Enter a phone number.

Select the start register for the vector, then define the vector length.
Note that each register byte contains 1 character.

Com Port: 1

○ Phone Number
9517717

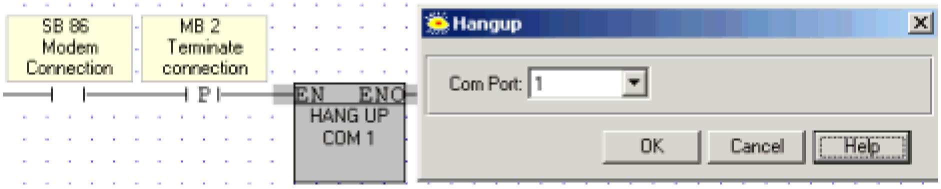
● Indirect Vector (Each register byte contains one character)
Start of Vector: MI 4 - Call site from keypad
Number of Characters (Determines vector length): 7

| MI | 4 | | 5 | | 6 | | 7 | |
|---|---|---|---|---|---|---|---|---|
| Hex | 35 | 39 | 37 | 31 | 31 | 37 | | 37 |
| ASCII | 5 | 9 | 7 | 1 | 1 | 7 | | 7 |

Hang-up

This enables you to use Ladder conditions to break the connection.

**Note** • Before activating Hang-up, check connection status via a Modem Connection SB: 86, 87, or 88.



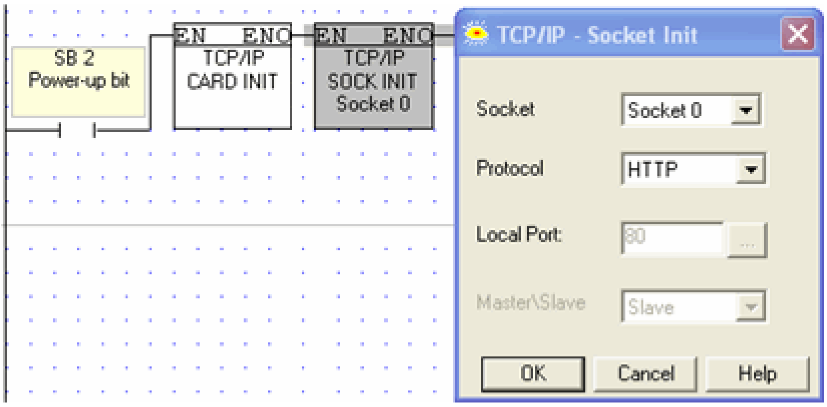
Com Port: 1

## Web Server

Enhanced Vision controllers can host web pages. The controller must comprise a TCP/IP port, and must be connected to an Ethernet network. If the controller contains web pages, a remote user can enter the IP address of the controller into a web browser and view the pages.

### Ladder Application

1. Configure Ethernet in your application by building a net that comprises the following elements:
    a) Com>TCP/IP> Card Init function.
    b) Com>TCP/IP>Socket Init function, set to HTTP.
    Selecting HTTP sets the local port to 80 and the controller to slave.



Socket: Socket 0

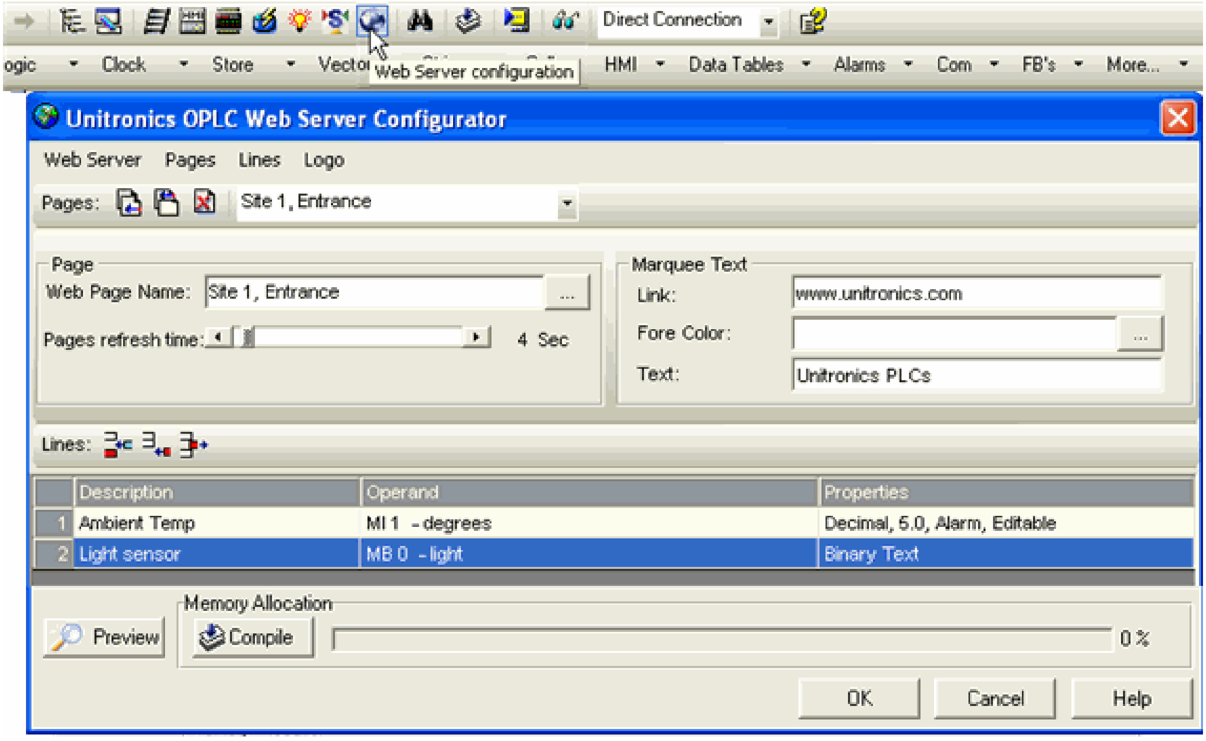
Protocol: HTTP

Local Port: 80

Master\Slave: Slave

### Configuring Web Pages

1. Click the Web Server Configuration icon on the VisiLogic toolbar to open the utility.
2. Click in the Web Page Name field and type in a page name.
3. You can also edit the Marquee text, which scrolls across the web page. The default causes the words 'Unitronics PLCs' to scroll across the page; clicking those words opens the Unitronics web site. You can replace the default text, change the website to another, or delete all of the text to remove the Marquee.



Unitronics OPLC Web Server Configurator

Pages: Site 1, Entrance

Page

Web Page Name: Site 1, Entrance

Pages refresh time: 4 Sec

Marquee Text

Link: www.unitronics.com

Fore Color:

Text: Unitronics PLCs

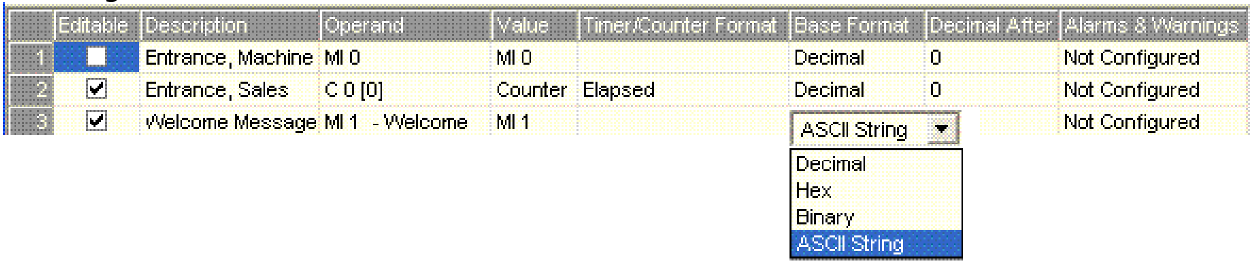
Lines:

| | Description | Operand | Properties |
|---|---|---|---|
| 1 | Ambient Temp | MI 1 - degrees | Decimal, 5.0, Alarm, Editable |
| 2 | Light sensor | MB 0 - light | Binary Text |

Memory Allocation

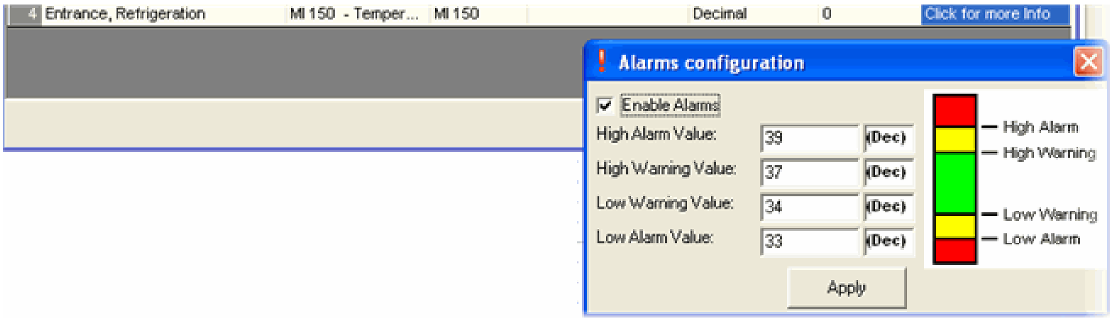
0 %

3. Select Editable if you want to enable users to edit register, counter, timer, and even text messages values online via the web page. Click on the different fields in the line to assign a Description and Operand.
   Note that the Operand type you select determines other options such as: Value, Timer/Counter Format, Base Format, Decimal, and Alarms and Warnings.



| | Editable | Description | Operand | Value | Timer/Counter Format | Base Format | Decimal After | Alarms & Warnings |
|---|---|---|---|---|---|---|---|---|
| 1 | □ | Entrance, Machine | MI 0 | MI 0 | | Decimal | 0 | Not Configured |
| 2 | ☑ | Entrance, Sales | C 0 [0] | Counter | Elapsed | Decimal | 0 | Not Configured |
| 3 | ☑ | Welcome Message | MI 1 - Welcome | MI 1 | | ASCII String | | Not Configured |

**Note •** When data is being edited, the page will not refresh.
You can also configure Alarm colors for register values by clicking in the Alarms & Warnings column.



| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | Entrance, Refrigeration | MI 150 - Temper... | MI 150 | Decimal | 0 | Click for more Info |

Alarms configuration

☑ Enable Alarms

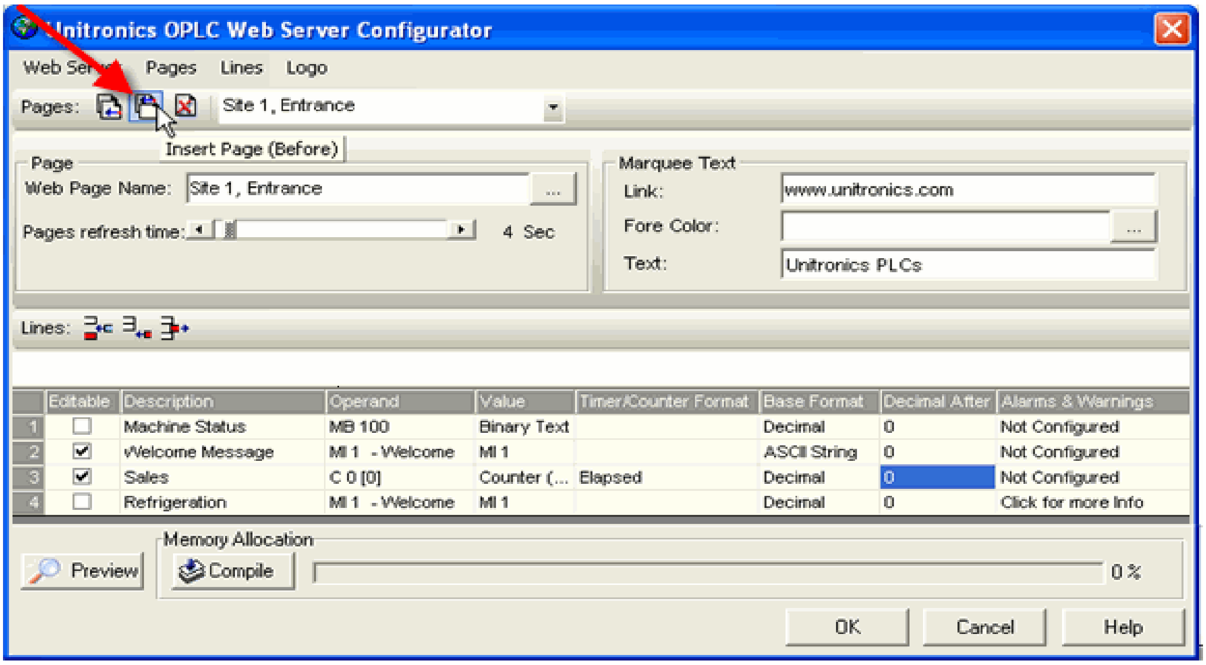
High Alarm Value: 39 (Dec)

High Warning Value: 37 (Dec)

Low Warning Value: 34 (Dec)

Low Alarm Value: 33 (Dec)

4. Use the toolbar to add and delete line and pages. You can use the slider to adjust the refresh time; this controls the frequency at which values are updated when the page is viewed via PC. Note that you cannot display an ASCII string longer than 32 characters.



Pages: Site 1, Entrance

Web Page Name: Site 1, Entrance

Pages refresh time: 4 Sec

Link: www.unitronics.com

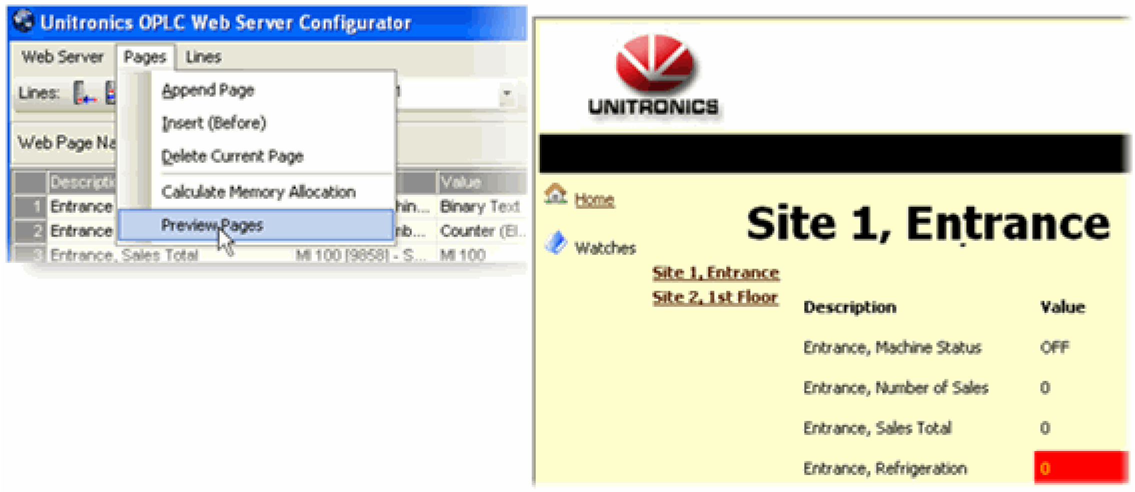
Fore Color:

Text: Unitronics PLCs

| | Editable | Description | Operand | Value | Timer/Counter Format | Base Format | Decimal After | Alarms & Warnings |
|---|---|---|---|---|---|---|---|---|
| 1 | □ | Machine Status | MB 100 | Binary Text | | Decimal | 0 | Not Configured |
| 2 | ☑ | Welcome Message | MI 1 - Welcome | MI 1 | | ASCII String | 0 | Not Configured |
| 3 | ☑ | Sales | C 0 [0] | Counter (... | Elapsed | Decimal | 0 | Not Configured |
| 4 | □ | Refrigeration | MI 1 - Welcome | MI 1 | | Decimal | 0 | Click for more Info |

Memory Allocation 0 %

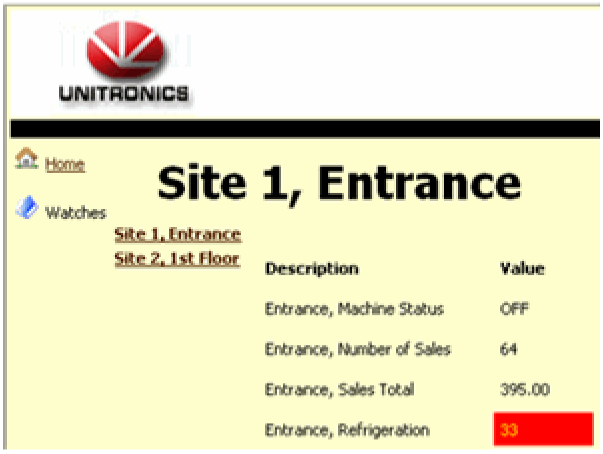
5. Click Page>Preview Pages to check your work.



Site 1, Entrance

| Description | Value |
|---|---|
| Entrance, Machine Status | OFF |
| Entrance, Number of Sales | 0 |
| Entrance, Sales Total | 0 |
| Entrance, Refrigeration | 0 |

When the page in the PLC is viewed via PC, the page will look as it did in preview mode, with the real-time values displayed.

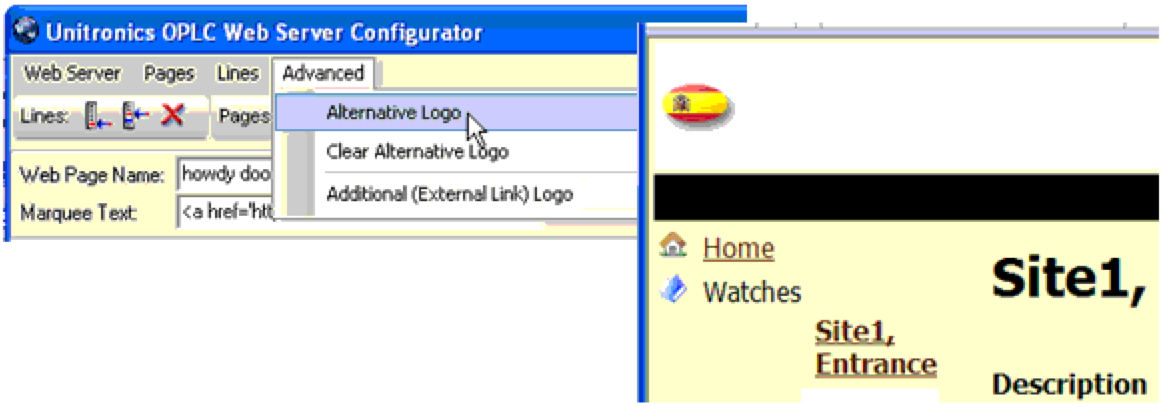


Site 1, Entrance

| Description | Value |
|---|---|
| Entrance, Machine Status | OFF |
| Entrance, Number of Sales | 64 |
| Entrance, Sales Total | 395.00 |
| Entrance, Refrigeration | 33 |

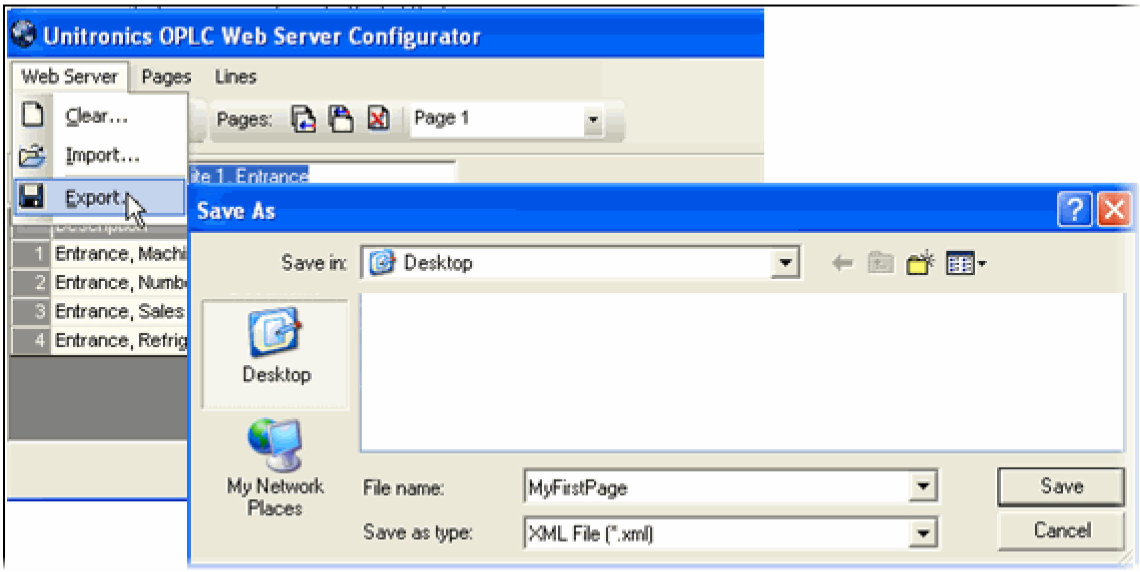
### Logos

You can customize the logo via the Advanced menu. You can either import a logo, or link to a .url.



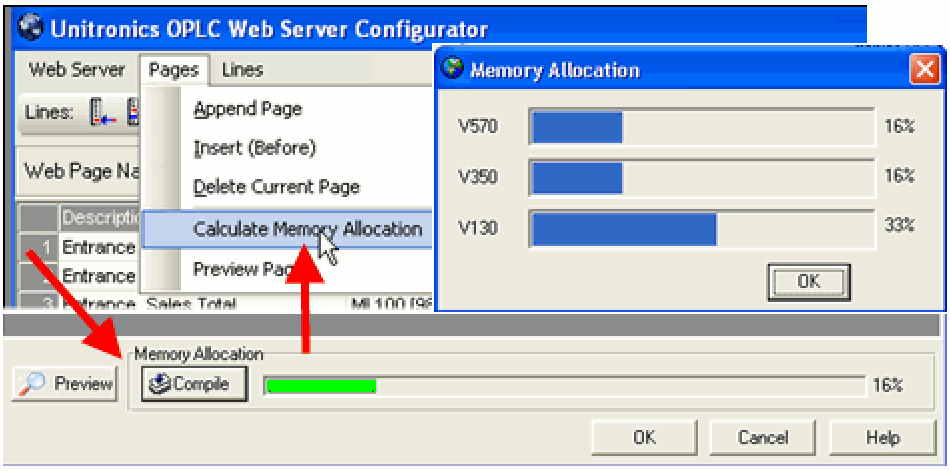
### Import/Export

You can export and import the configuration by selecting option on the Web Server menu.



### Web Page Capacity

The capacity for controllers from the V570 and V350 series is 128k. The capacity of the V130 series is 64K. You can check the requirements of your pages by clicking the Compile button and then checking Pages>Calculate Memory Allocation.



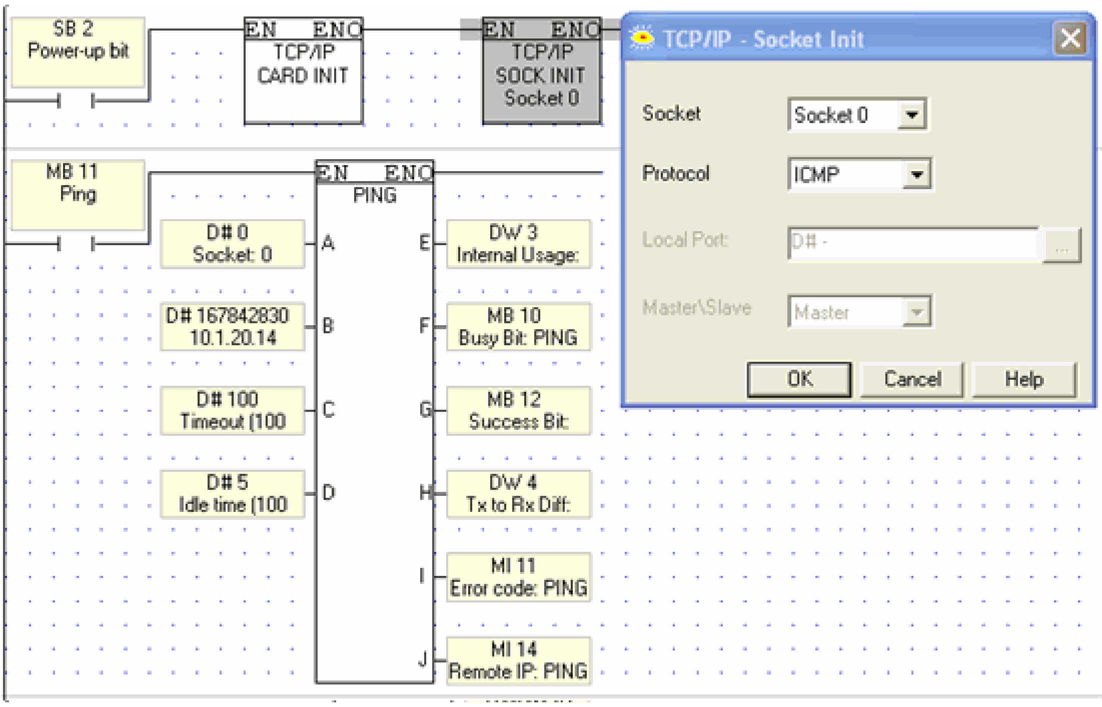
## Flickering

By default, the controller displays the web page using AJAX technologies to automatically refresh variable data on the page. Note that AJAX may not be compatible with certain older browsers. If this applies to your application, you can disable AJAX by clicking it on the Web Server> Use AJAX option.

## Ping

Use the Com>TCP/IP> Ping function to ping a remote device.

**Note •** The socket used to send Ping data must be initialized to ICMP.



Socket: Socket 0
Protocol: ICMP
Local Port: D# -
Master\Slave: Master

| Parameter Name | Type | Purpose |
|---|---|---|
| Socket | Constant | The PLC sends the data through this socket. Select a socket that is initialized to ICMP |

<table>
<tr><td>Remote PLC</td><td>Constant, MI, XI</td><td>The IP of the remote device</td></tr>
<tr><td>Timeout (100ms)</td><td>Constant or register</td><td>User-defined. If no answer is received from the remote device within the Timeout period, the function waits for the Idle time to pass before resending the Ping data.</td></tr>
<tr><td>Idle time (100ms)</td><td>Constant or register</td><td>If no answer is received from the remote device, the function wait for the Timeout + Idle Time before resending the Ping request.<br>If a reply is received, the function wait for the Idle time, before resending.</td></tr>
<tr><td>Internal usage</td><td>DW, XDW</td><td>The function uses this for internal management. Set a Power-up value of 0.</td></tr>
<tr><td>Busy Bit</td><td>MB, XB</td><td>Turns ON when the function begins waiting for a reply<br>Turn OFF when the reply arrives, or when timeout is exceeded.</td></tr>
<tr><td>Success Bit</td><td>MB, XB</td><td>Turns ON when reply is received without error.<br>Reset by user.</td></tr>
<tr><td>Rx to Tx Time</td><td>DW, XDW</td><td>Resolution:<br>• Standard: 2.5 mSec<br>• Enhanced: 1 µSec<br>Valid value ONLY when the Success bit rises.</td></tr>
<tr><td>Error Code</td><td>Register</td><td>Ping supports the standard ICMPv4 Destination Unreachable Message Subtypes listed in the following table
<table>
<tr><th>Error Codes (ICMPv4)</th><th>Purpose</th></tr>
<tr><td>-2</td><td>Timeout exceeded– no reply at all</td></tr>
<tr><td>-1</td><td>No error</td></tr>
<tr><td>0</td><td>Network Unreachable<br>The datagram could not be delivered to the network specified in the network ID portion of the IP address. Usually means a problem with routing but could also be caused by a bad address.</td></tr>
<tr><td>1</td><td>Host Unreachable<br>The datagram was delivered to the network specified in the network ID portion of the IP address but could not be sent to the specific host indicated in the address. Again, this usually implies a routing issue.</td></tr>
<tr><td>2</td><td>Protocol Unreachable<br>The protocol specified in the Protocol field was invalid for the host to which the datagram was delivered.</td></tr>
<tr><td>3</td><td>Port Unreachable<br>The destination port specified in the UDP or TCP header was invalid.</td></tr>
<tr><td>4</td><td>Fragmentation Needed and DF Set<br>Normally, an IPv4 router will automatically fragment a datagram that it receives if it is too large for the maximum transmission unit (MTU) of the next physical network link the datagram needs to traverse.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>However, if the DF (Don't Fragment) flag is set in the IP header, this means the sender of the datagram does not want the datagram ever to be fragmented. This puts the router between the proverbial rock and hard place, and it will be forced to drop the datagram and send an error message with this code.<br>This message type is most often used in a “clever” way, by intentionally sending messages of increasing size to discover the maximum transmission size that a link can handle. This process is called MTU path discovery.</td></tr>
<tr><td></td><td></td><td>5</td><td>Source Route Failed<br>Generated if a source route was specified for the datagram in an option but a router could not forward the datagram to the next step in the route.</td></tr>
<tr><td></td><td></td><td>6</td><td>Destination Network Unknown<br>Not used; Code 0 is used instead.</td></tr>
<tr><td></td><td></td><td>7</td><td>Destination Host Unknown<br>The host specified is not known. This is usually generated by a router local to the destination host and usually means a bad address.</td></tr>
<tr><td></td><td></td><td>8</td><td>Source Host Isolated<br>Obsolete, no longer used.</td></tr>
<tr><td></td><td></td><td>9</td><td>Communication with Destination Network is Administratively Prohibited<br>The source device is not allowed to send to the network where the destination device is located.</td></tr>
<tr><td></td><td></td><td>10</td><td>Communication with Destination Host is Administratively Prohibited<br>The source device is allowed to send to the network where the destination device is located, but not that particular device.</td></tr>
<tr><td></td><td></td><td>11</td><td>Destination Network Unreachable for Type of Service<br>The network specified in the IP address cannot be reached due to inability to provide service specified in the Type Of Service field of the datagram header.</td></tr>
<tr><td></td><td></td><td>12</td><td>Destination Host Unreachable for Type of Service<br>The destination host specified in the IP address cannot be reached due to inability to provide service specified in the datagram's Type Of Service field.</td></tr>
<tr><td></td><td></td><td>13</td><td>Communication Administratively Prohibited<br>The datagram could not be forwarded due to filtering that blocks the message based on its contents.</td></tr>
<tr><td></td><td></td><td>14</td><td>Host Precedence Violation<br>Sent by a first-hop router (the first router to handle a sent datagram) when the Precedence value in the Type Of Service field is not permitted.</td></tr>
<tr><td></td><td></td><td>15</td><td>Precedence Cutoff In Effect<br>Sent by a router when receiving a datagram whose Precedence value (priority) is lower than the minimum allowed for the network at that time.</td></tr>
<tr><td>Remote IP</td><td></td><td colspan="2">The IP from the replying device. Note that it might be different than the “input Remote IP” due to network topology.</td></tr>
</table>

## Send e-mail

This function enables a controller to send an e-mail in response to Ladder conditions. In order to send e-mail, the controller must comprise a TCP/IP port, and must be connected to an Ethernet network with access to a mail server.

Before you begin you need the following information:

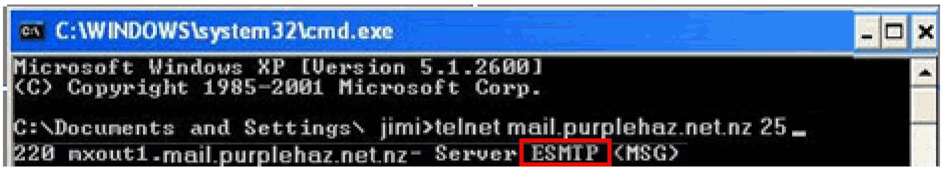
- The type of protocol your mail server uses, SMTP or ESMTP
- Your mail server's IP address

There are sample applications that show how e-mail may be sent via Ethernet, and via GPRS modem.

### Determining the Protocol Type

To determine which protocol your server supports, run Telnet.

1. Click Windows Start > Run, then enter CMD.
2. In the CMD window, enter the command: telnet RemoteMailServer 25, where RemoteMailServer is the name of your mail server, and 25 is the port via which you contact the server. The command returns the protocol type used by the server.

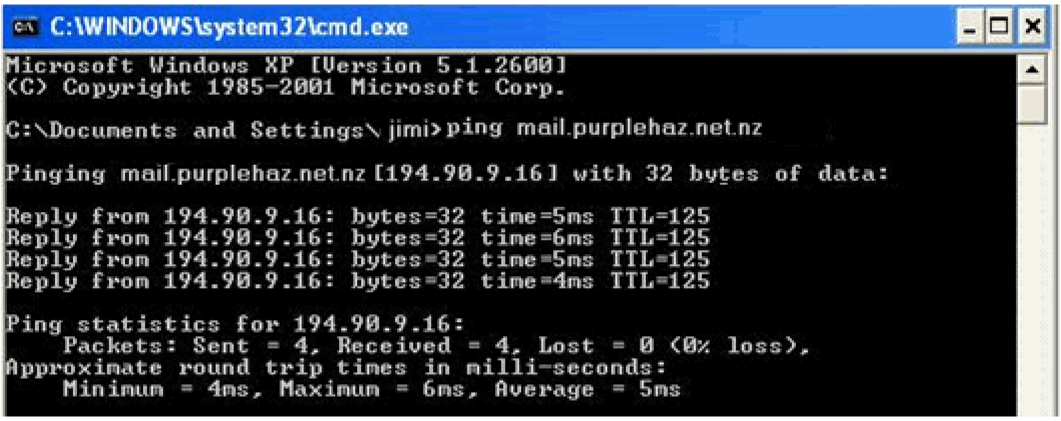


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ jimi>telnet mail.purplehaz.net.nz 25
220 mxout1.mail.purplehaz.net.nz- Server ESMTP <MSG>
```

### Determining the IP

Use the 'ping' command to determine the IP address of your mail server.

1. Click Windows Start > Run, then enter CMD.
2. In the CMD window, enter the command: ping RemoteMailServer, where RemoteMailServer is the name of your mail server; the command returns the IP address of  the server.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ jimi>ping mail.purplehaz.net.nz

Pinging mail.purplehaz.net.nz [194.90.9.16] with 32 bytes of data:

Reply from 194.90.9.16: bytes=32 time=5ms TTL=125
Reply from 194.90.9.16: bytes=32 time=6ms TTL=125
Reply from 194.90.9.16: bytes=32 time=5ms TTL=125
Reply from 194.90.9.16: bytes=32 time=4ms TTL=125

Ping statistics for 194.90.9.16:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 6ms, Average = 5ms
```

### Intranet (LAN) mail

If your controller is sending e-mail to addresses within a local network, and you know the name of the server in your network, communicate with it directly.

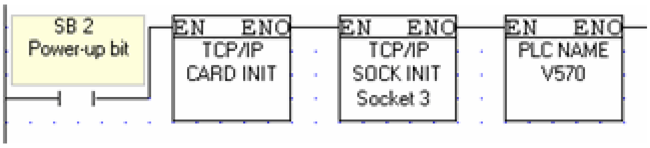
### e-mail via GPRS

Note that you can also send e-mail via GPRS modem. When you prepare the modem via Modem Services, set it to a baud rate of 9600. In addition, the COM Init function should be set to 9600.

### How to Configure and Send Messages

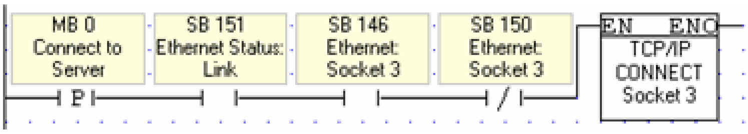
1. Configure Ethernet in your application by building a net that comprises the following elements:

a. Com>TCP/IP> Card Init function.
b. Com>TCP/IP>Socket Init function,
set to **TCP Master**.
c. A Com>Set PLC Name function.



**Note •** Dedicate one socket to sending e-mail.

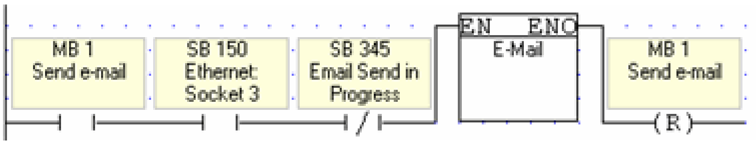
2. Establish the Ethernet Connection using the Com>TCP/IP> TCP/IP Connect Socket function. The function must be configured to your mail server IP, and to Port 25, which is the outgoing messages port.



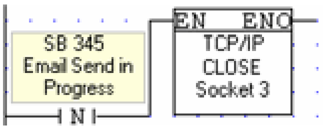
**Note •** It is recommended that you include a time elapse of a few seconds after the Ethernet Card initializes and before activating Socket Connect. A timer may be used for this purpose.

5. Send the e-mail using the Com>TCP/IP>Send e-mail function.

- Use the appropriate SB to ensure that the socket is connected before sending the e-mail.
- Use the status of SB 345 Email Send in Progress to avoid communication conflicts and ensure that the e-mail function is free before sending the e-mail. This bit turns ON when the e-mail function is activated, and turns OFF when the message has been sent to the server.
Note that:
- only one e-mail can be sent at a time, and that an application should use the same socket to send all e-mails.
- you can send a number of e-mails before closing the connection.

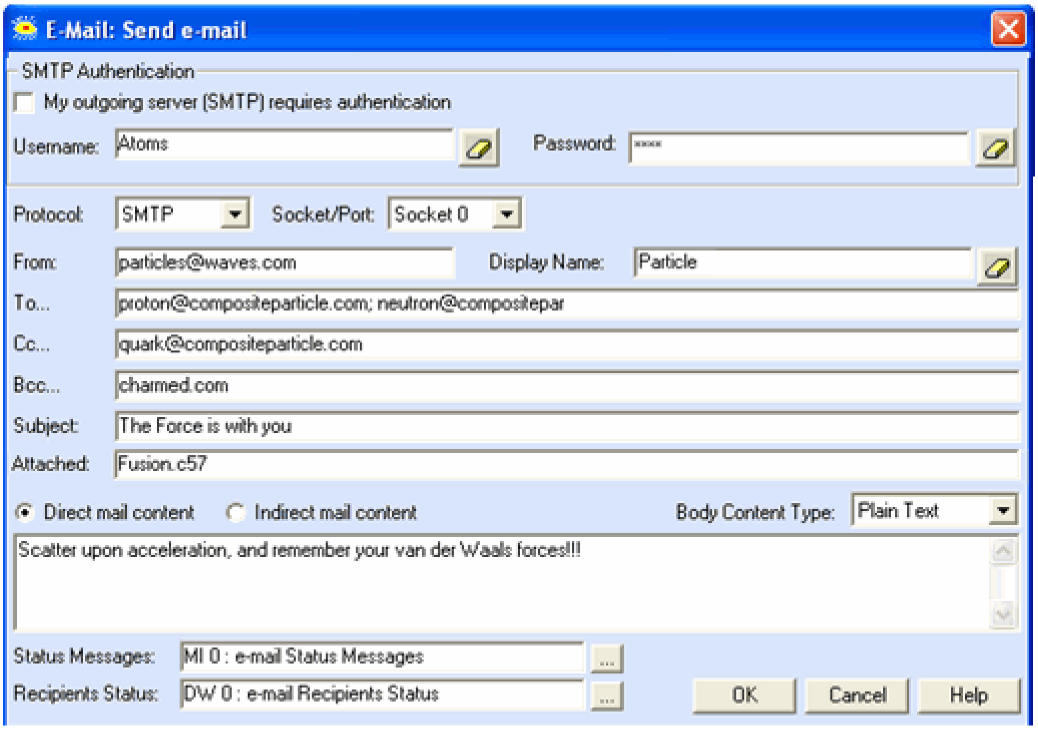


6. Close the connection and free the socket using the Com>TCP/IP>TCP/IP Close Socket function.
   After closing the socket, wait 30 seconds before sending additional e-mails.



### Configuring the Send e-mail Function

Fill in the parameters according to the parameter table given below



E-Mail: Send e-mail

SMTP Authentication

□ My outgoing server (SMTP) requires authentication

Username: Atoms

Password: ****

Protocol: SMTP

Socket/Port: Socket 0

From: particles@waves.com

Display Name: Particle

To... proton@compositeparticle.com; neutron@compositepar

Cc... quark@compositeparticle.com

Bcc... charmed.com

Subject: The Force is with you

Attached: Fusion.c57

● Direct mail content ○ Indirect mail content

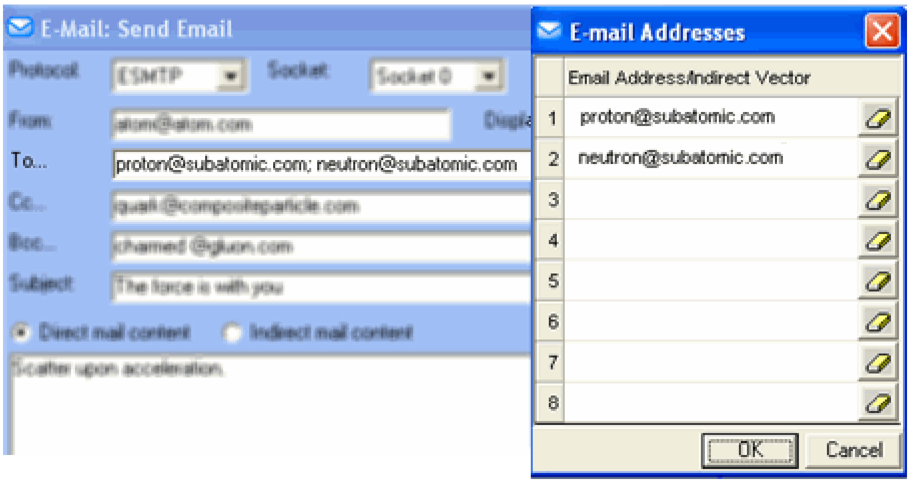
Body Content Type: Plain Text

Scatter upon acceleration, and remember your van der Waals forces!!!

Status Messages: MI 0 : e-mail Status Messages

Recipients Status: DW 0 : e-mail Recipients Status

| Parameter Name | Purpose |
|---|---|
| Protocol | Select the protocol your server uses.<br>Both SMTP and ESMTP are supported. |
| Socket | Select the socket you are using for this function. |
| From | You can either type in an address, or link a vector of registers |
| To/Cc/Bcc | You can select up to 8 recipients per category<br>Click a line to open the address book. |

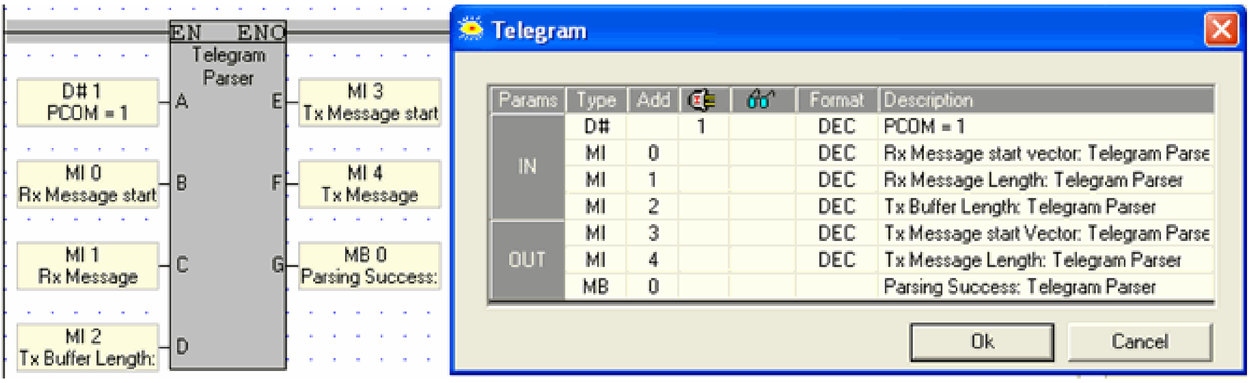
| | |
|---|---|
| | <br>E-Mail: Send Email<br>Protocol: ESMTP<br>Socket: Socket 0<br>From: atom@atom.com<br>To...: proton@subatomic.com; neutron@subatomic.com<br>Cc...: quark@compositeparticle.com<br>Bcc...: charmed@gluon.com<br>Subject: The force is with you<br>● Direct mail content ○ Indirect mail content<br>Scatter upon acceleration.<br>E-mail Addresses<br>Email Address/Indirect Vector<br>1 proton@subatomic.com<br>2 neutron@subatomic.com<br>3<br>4<br>5<br>6<br>7<br>8<br>OK Cancel |
| Subject | Click to enter text or link a vector. The vector will end either at a null character, or when reaching the  maximum of 50 characters |
| Attached | Enhanced Vision controllers with SD cards enable you to attach up to 8 files from the SD card.<br>Click to set the attachment names.<br>Note that:<br>- The file name must adhere to the 8.3 file format, up to 8 characters for the name, 3 for the extension.<br>- File name may be provided by constant text or register. Note that if the name comes from an MI, the function copies a vector 8 bytes long, or until it finds a 'null' character<br>- The size of the attachments is limited in SI 58. maximum attachment size is 10 MB.<br>- Power-up default is 1<br>- Maximum per attachment = 10 (10 MB)<br>Note that the file size must not be changed while the Send is in Progress. |
| Direct mail content | Select to type your e-mail message into the content field. You can enter up to 800 characters. |
| Indirect mail content | Select to link to a vector. The vector will end either at a null character, or when reaching the  maximum of 800 characters |
| The following operands should be assigned Power-up Values; bits should be reset, and registers initialized to 0. | |
| Status Messages | When the In Progress bit turns ON, the current status message turns from 0 to 1. When the<br>Status Message<br>0 – IDLE<br>1 – IN PROGRESS<br>2 - SUCCESS<br>3 – SMTP server does not respond ( HELO command Failed)<br>4 – ESMTP server does not respond ( EHLO command Failed)<br>5 – 'From' address format incorrect<br>6 – 'To/Cc/Bcc' address format incorrect. Check Address Status Operand<br>7 – The server cannot receive data (DATA mode error)<br>8 – Mail was not sent: error occurred while sending e-mail header (name, subject, etc..)<br>9 - Mail was not sent: error occurred while sending e-mail data (text, attachments, etc..)<br>10 - Mail was not sent: error occurred while sending e-mail end of data/closing the e-mail<br>11 – Mail was not sent: error occurred since authentication is not enabled or it's not supported<br>12 – Mail was not sent: error occurred while sending username with authentication mode<br>13 – Mail was not sent: error occurred while sending password with authentication mode<br>14 – Mail was not sent: error occurred while sending username with |

| | |
|---|---|
| | plain login mode<br>15 – Mail was not sent: error occurred while sending password with plain login mode |
| Recipients Status | If the Status Message is 6, use this register to determine which addresses are in incorrect format.<br>• Bits 0-7: 'To' address X is illegal<br>• Bits 8-15: 'Cc' address X is illegal<br>• Bits 16-23: 'Bcc' address X is illegal<br>The bit corresponds with the incorrect address. If, for example, bit 9 is ON, the address on the second Cc line is incorrect. |

**Note •** The function does not support authentication, SSL or TSL encryption.

## Telegram Parser

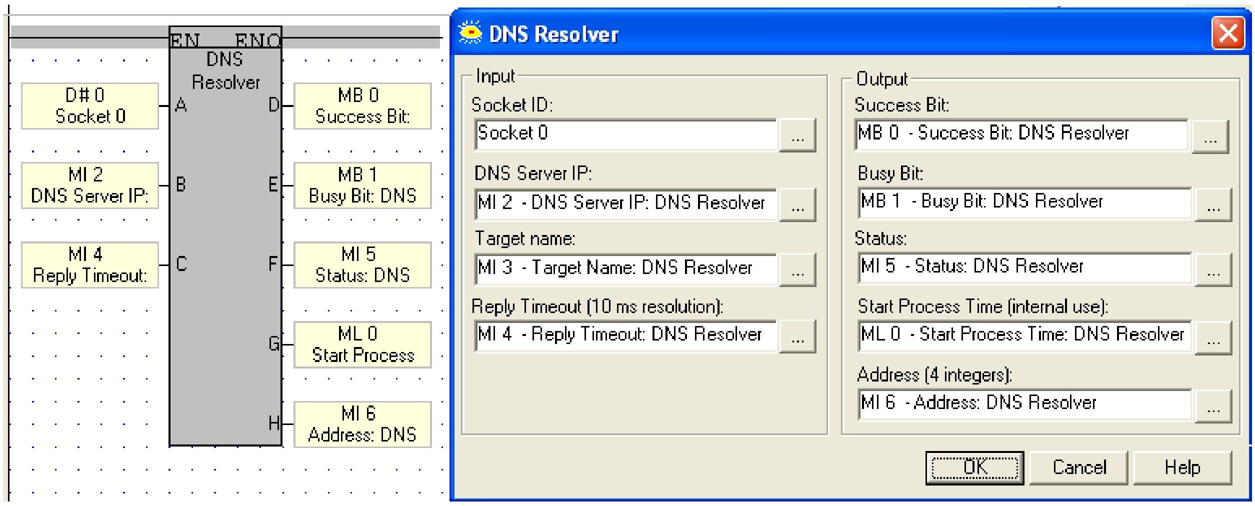
This function parses data sent to a vector of operands. You can use this function in conjunction with FB Protocol to parse messages.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 1 | | DEC | PCOM = 1 |
| | MI | 0 | | | DEC | Rx Message start vector: Telegram Parse |
| | MI | 1 | | | DEC | Rx Message Length: Telegram Parser |
| | MI | 2 | | | DEC | Tx Buffer Length: Telegram Parser |
| OUT | MI | 3 | | | DEC | Tx Message start Vector: Telegram Parse |
| | MI | 4 | | | DEC | Tx Message Length: Telegram Parser |
| | MB | 0 | | | | Parsing Success: Telegram Parser |

## DNS Resolver

Use this function to resolve a server IP address from its domain name. The DNS Resolver uses RFC 1035. Note that the socket must be UDP.



Input

Socket ID: Socket 0

DNS Server IP: MI 2 - DNS Server IP: DNS Resolver

Target name: MI 3 - Target Name: DNS Resolver

Reply Timeout (10 ms resolution): MI 4 - Reply Timeout: DNS Resolver

Output

Success Bit: MB 0 - Success Bit: DNS Resolver

Busy Bit: MB 1 - Busy Bit: DNS Resolver

Status: MI 5 - Status: DNS Resolver

Start Process Time (internal use): ML 0 - Start Process Time: DNS Resolver

Address (4 integers): MI 6 - Address: DNS Resolver

Status messages

0 – OK

1 – In progress

2 – No answer

3 – Name does not exist

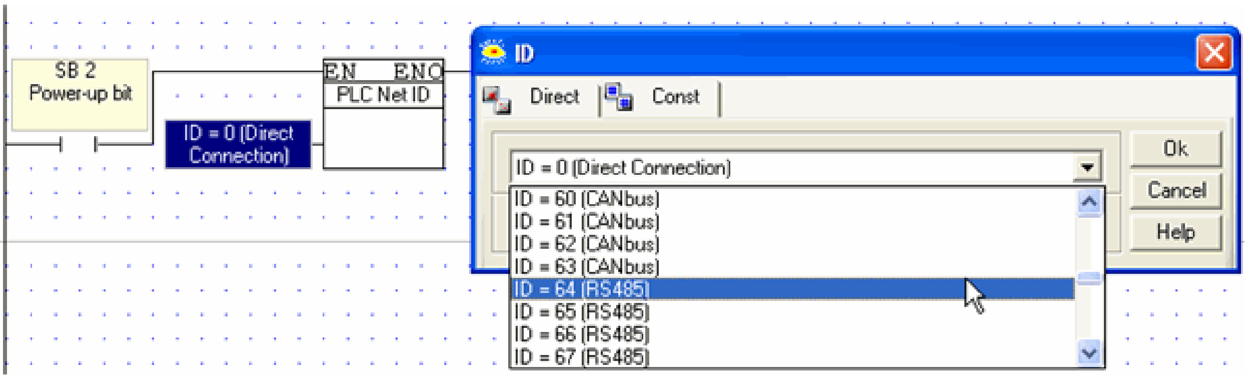
4 – Ethernet inactive, or socket is not UDP

5 – Protocol Error

## Set PLC ID Number

Located on the COM menu, this function enables you to assign a unique ID number to a PLC. This name can, for example, be used to identify the PLC for CANbus or RS485 networking purposes.

This should be assigned as a power-up task.



**Notes** • When the function is activated, the ID is written into the PLC. If the value is supplied via Indirect Vector, note that simply storing a value into the vector will not rewrite the ID. To rewrite the ID, the value must first be stored in the appropriate vector, and then the function must be activated.

• This feature is not supported by the V120-12 series.

Set PLC Name

Located on the COM menu, this function enables you to assign a unique name to a PLC. This name can, for example, be used to identify the PLC for Ethernet networking purposes.

The PLC name should be assigned as a power-up task.

Notes • When the function is activated, the name is written into the PLC. If the name is supplied via Indirect Vector, note that simply storing a value into the vector will not rewrite the PLC name. To rewrite the name, the value must first be stored in the appropriate vector, and then the Set PLC Name FB must be activated.

- This feature is not supported by the V120-12 series.

## Modem Troubleshooting

Note that Vision PLCs have a built-in communications 'sniffer'. This is accessible in Information Mode, under System.

- Touch screen models: select Serial or Ethernet, then click Monitor. Note the button that allows you to toggle between Hex and ASCII.
- Non-touch screen models: Select Communication, then select Serial or Ethernet> COM Buffer. Use F2 to toggle between Hex and ASCII. In

addition, press Enter, and then the Down key to toggle between Tx and Rx.

## Modem Commands

**Note** • The modem must reply with either OK or READY to each command entered. If the modem fails to answer, the command has not been processed.

| | |
|---|---|
| +++ | Escape Sequence. This causes the modem to close connections and go back to command mode |
| AT | This command means Attention; and is used to begin a session |
| AT&F | Restores factory default settings |
| ATZ | Resets the modem. This command may take time to implement, so the response from the modem may be delayed |
| ATE0 | No Echo |
| V1 | Enable Verbose (long) response |
| Q0 | Respond |
| X4 | Detailed answers |
| &D0 | Ignore DTR |
| &S0 | DSR always ON.<br>Since the DSR can be permanently set to ON, connecting it to the RTS causes the terminal always be ready to transmit\receive data |
| &S1 | DSR OFF in command and test modes |
| &C1 | Give the user a signal for the DCD |
| ATS0=1 | Auto-Answer after 1 ring |
| &W | Burn the configuration into the modem's non-volatile memory |

## PC-side modem, error messages

This deals with errors that may result from the PC's modem

| Message | Cause |
|---|---|
| COM Port not open, or modem does not exist | The PC was unable to access the PC port.<br>The port may:<br>-Already be in use.<br>-Be damaged. |
| Modem not connected | The PC receives no reply from the modem following the 'AT' command.<br>Check that:<br>-The modem is connected to the same PC port you have defined in PC-modem Configuration.<br>-The PC-modem cable is in proper order. |
| Modem not initialized | The modem was not successfully initialized, or<br>The modem answered, but did not approve all strings.<br>Check the topic: Using Hyperterminal for Modem Troubleshooting |
| The messages below describe the modem 's status if the PC dial attempt ( ATD+ number) fails. Any one of these errors aborts the Dial process. | |
| Modem Busy | |
| Modem Error | |
| No Dial Tone | |
| No Carrier | **Note** • This can occur if, within the modem initialization strings, the parameter S7 TimeOut, is to short to permit the PLC's modem to answer. For example, if this parameter is set as S7=30, the PC modem will wait for 3 seconds to receive an answer from the PLC's modem. If, however, the PLC program's COM Init FB Answer Settings are set to 'Answer after 6 rings,' the PLC modem will not be able to answer before the 3 seconds have elapsed. In this case, the TimeOut parameter is exceeded, and the PC modem will return the No Carrier error. |
| Dial time-out exceeded | No reply was received from the modem within the defined time. |

The messages below only relate to unsuccessful GSM modem initialization.

| | |
|---|---|
| GSM SIM card blocked | |
| GSM SIM card does not exist | |
| Illegal GSM PIN code | |
| GSM Network not found | |
| CDMA Network not found | |
| SMS message too long | A message containing only English characters may contain up to 160 characters.<br>A message containing non-English characters may contain up to 70 characters. |
| Time-out exceeded | |

### PLC modems

These errors may result from problems in the PLC-side modem

| Message | Possible cause | Recommended action |
|---|---|---|
| Modem Busy | Modem is engaged,<br>or is being initialized | Check that the line is free.<br>Use the **SBs: Modem Initialization Status** listed above to check the COM port status; communications cannot flow through the port during initialization. For more information check the topic COM Port Init. |
| Handshake between modems complete ('CONNECT'),<br>PLC does not reply | Modem adapter cable | Check the PLC-to-modem connection and pin-out, particularly that the DSR is connected to the RTS on the modem side. |

| Problem | SI Value (80, 82, 84) | Possible Cause & Recommended Action |
|---|---|---|
| Modem fails to initialize (SB 81, 83, 85 ON) | 3 | • PLC-to-modem cable:<br>Make sure that the cable is securely connected. Check the modem connection and pin-out of the PLC-to-modem adapter cables. Note that if you use cables comprising this pin-out, you must set the parameter Flow Control to N (none) in the COM Port Init FB.<br>• Incompatible communication settings.<br>Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. You may need to manually change your modem's communication settings. |
|  | 0 | You may have selected the wrong type of modem in the COM Port Init FB. |

A complete list of error messages for SI 80, 92, 84, as well as other System Operands containing modem error is given in the topic COM Port/Modem Status, System Operands, & Error Messages.

### Modem Connection

**Cable Pin-out**

The Unitronics' cable provided with modem kits does not provide a standard connection. This connection is adapted to support the fact that Unitronics controllers do not support the control lines. The cable shorts the DSR and the DTR together, which ensures that the terminal is always ready to receive data. For more information, refer to the topic Modem Connection and Pinouts.

### Data Flow Direction

Generally, when you transmit data, you send it out. Note, however, that transmitted data (TXD) is input to the DCE. A Receive Data signal (RXD) is input to the DTE, but output from the DCE.

Therefore, the RXD and TXD signals are crossed within the majority of modems. This means that a straight through "one to one" cable is generally all that is necessary between a modem and a controller or PC serial port.

## RS-232 signal information

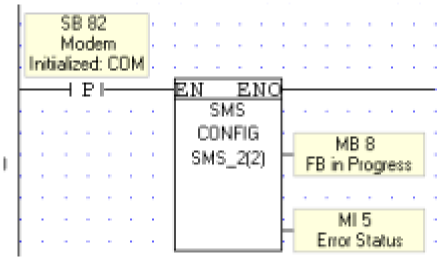
| | |
|---|---|
| RXD<br>Receive Data | Input for DTE devices (Receive), output for DCE devices. This is the data channel from the DCE device to the DTE device. |
| TXD<br>Transmit Data | Output for DTE devices (Send), input for DCE devices. This is the data channel from the DTE device to the DCE device. |
| GND<br>Signal Ground | Signal return for all signal lines. |
| RTS<br>Request To Send | Terminal is ready to receive data. When the DTE is ready to receive data, the DTE serial port RTS signal is ON. |
| CTS<br>Clear To Send | Terminal is ready --not related to data transfer. |
| DTR<br>Data Terminal Read | It is an output for DTE devices and an input for DCE devices. This signal is typically used to show that the port has been activated or "opened". |
| DSR<br>Data Set Ready | Detects if the RS232 is actually connected. |
| DCD<br>Data Carrier Detect | Turns ON when the modems connect. |
| RING | Turns ON when someone is calling the DTE. |

## GSM modems

| Problem | SI Value (81, 83, 85) | Possible Cause & Recommended Action |
|---|---|---|
| Wrong PIN number | 3 | Check the PIN number in the COM Port Init FB; leave it empty if your SIM card has no PIN number. |
| Failed Registration | 4 | GSM modem did not register successfully, for example if no network was found, or if the modem antenna is not functioning. |
| PUK number needed | 5 | The SIM card is locked due to too many attempts to enter an incorrect PIN number. |

| Problem | Possible Cause & Recommended Action |
|---|---|
| Cell phone does not receive message | Check the cell phone's SIM card; it may be full. You can clear the SIM card using the Clear option in the COM Port Init FB. |
| PLC modem does not respond to cell phone call | GSM modems generally recognize whether the incoming call is voice or data. If the modem has been set to answer (via the ATS0=x command, where x is set to a value greater than 0), the modem will not answer an incoming cell phone call, since it will be recognized as a voice message. |

**Note •** The appropriate Modem: Initialized SBs [80 (COM 1), SB 82 (COM 2), SB 84 (COM 3)] must turn ON before activating an SMS Config FB using that COM port; ideally the SBs should be used as an activating condition.

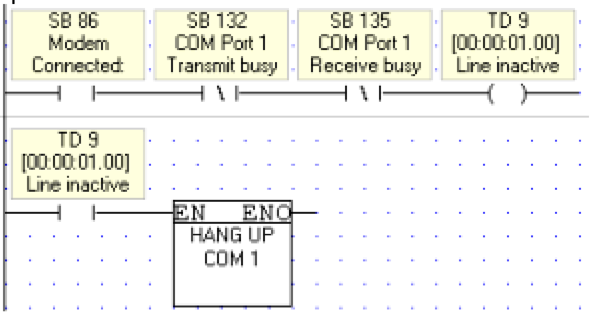


Changing Baud Rate

Note that baud rates for PC and PLC modems can be edited during the Prepare Modem process.

### Modem Troubleshooting Table

Note that in many cases, checking System Operand status can help you in the troubleshooting process.

| Problem | Possible cause | Recommended Action |
|---|---|---|
| PLC stays connected when modem connection fails | A modem which is reset, or to which the power supply is interrupted, may lose the connection and may not send a 'No Carrier string to the PLC.<br>In this case, the SB remains ON, as though the data link still exists.<br>(Note that Modem Connected SBs, 86-88, turn ON when the relevant COM port receives the 'Connect' modem string. Receiving a 'No Carrier' string causes the SB to turn OFF.) | Modem Connection and COM Port Transmit/Receive SBs can be used in conjunction with a timer to check if the line is inactive; the timer state is used to trigger Hang-up as shown below.<br><br> SB 86 Modem Connected<br>SB 132 COM Port 1 Transmit busy<br>SB 135 COM Port 1 Receive busy<br>TD 9 [00:00:01.00] Line inactive<br>TD 9 [00:00:01.00] Line inactive<br>EN ENO<br>HANG UP COM 1 |
| VisiLogic is not able to communicate with the PLC:<br>PLC has initialized a modem. The modem is disconnected, and a PC is plugged into the COM port. | Once the modem is connected to the PLC, and the modem is initialized, the Modem Initialized SB turns ON.<br>As long as this SB is ON, the COM port stays initialized to the baud rate required in order to work with the modem. If VisiLogic attempts to access the PLC via a different baud rate, the attempt will fail.<br>The Synchronization process (using break signals) is disabled while the SB is ON. | Either:<br>Reset the PLC, which will reset the Modem Initialized SB,<br>-or-<br>Via Vision Communication PC Settings, set VisiLogic's baud rate to match the baud rate to which the COM port was initialized to communication with the modem. |

## Ethernet

Information regarding Ethernet is provided in the manual VisiLogic – Communications.

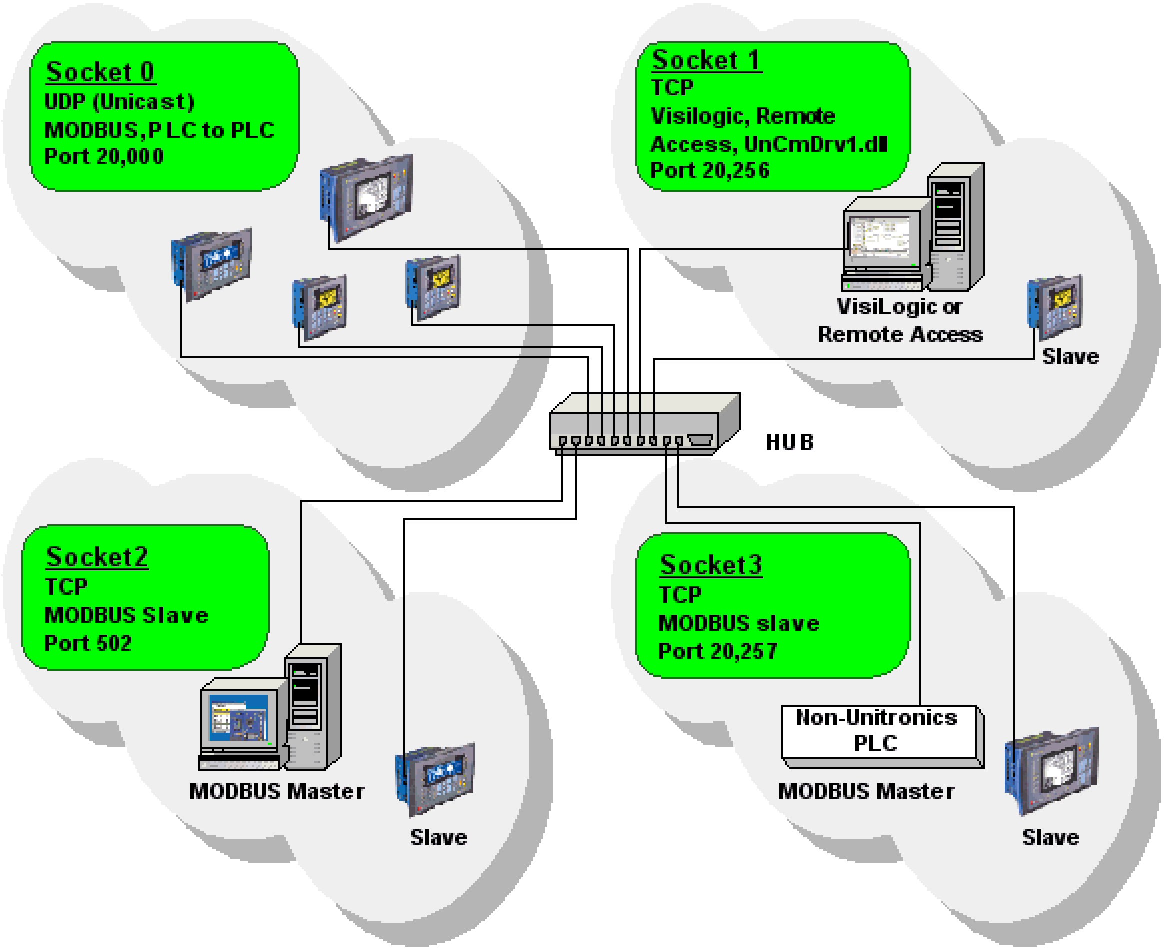
Data communications via Ethernet are supported by:

MODBUS IP FBs and Protocol TCP/IP FBs (see the manual VisiLogic - Function Blocks)

- Remote PLC DataCom (UDP) and UDP Raw (Vision to Vision), described below.

### Default Socket Configuration

The **default** socket configuration enables you to implement these communication options as shown below:



### MODBUS

Use the MODBUS IP FBs to:

- Communicate data within a PLC network.
- Use a PC to access a PLC via MODBUS over TCP.
- Use MODBUS over TCP to enable non-Unitronics PLCs to access Unitronics PLCs, via MODBUS.

### PLC networks, PLC to PLC

Any controller within the network can be both master and slave. In order to be read by the master, a slave's application must contain the MODBUS IP Scan FB.

### UDP: controller-to-controller communication

In order to communicate via Ethernet throughout your controller network, you must include an Ethernet Card Init FB in the ladder application of each networked controller. **When using UDP, do not use the Socket: Connect or Socket: Close elements; these are only required by TCP applications.**

#### Master

The master PLC Ladder application must include the elements shown below.

**Step 1: Initializing the Ethernet card and configuring MODBUS**

The MODBUS Configuration is linked to Socket 0, which is by default set to UDP.

**Note •** A PLC defined as a UDP master can communicate with a number of slave devices.

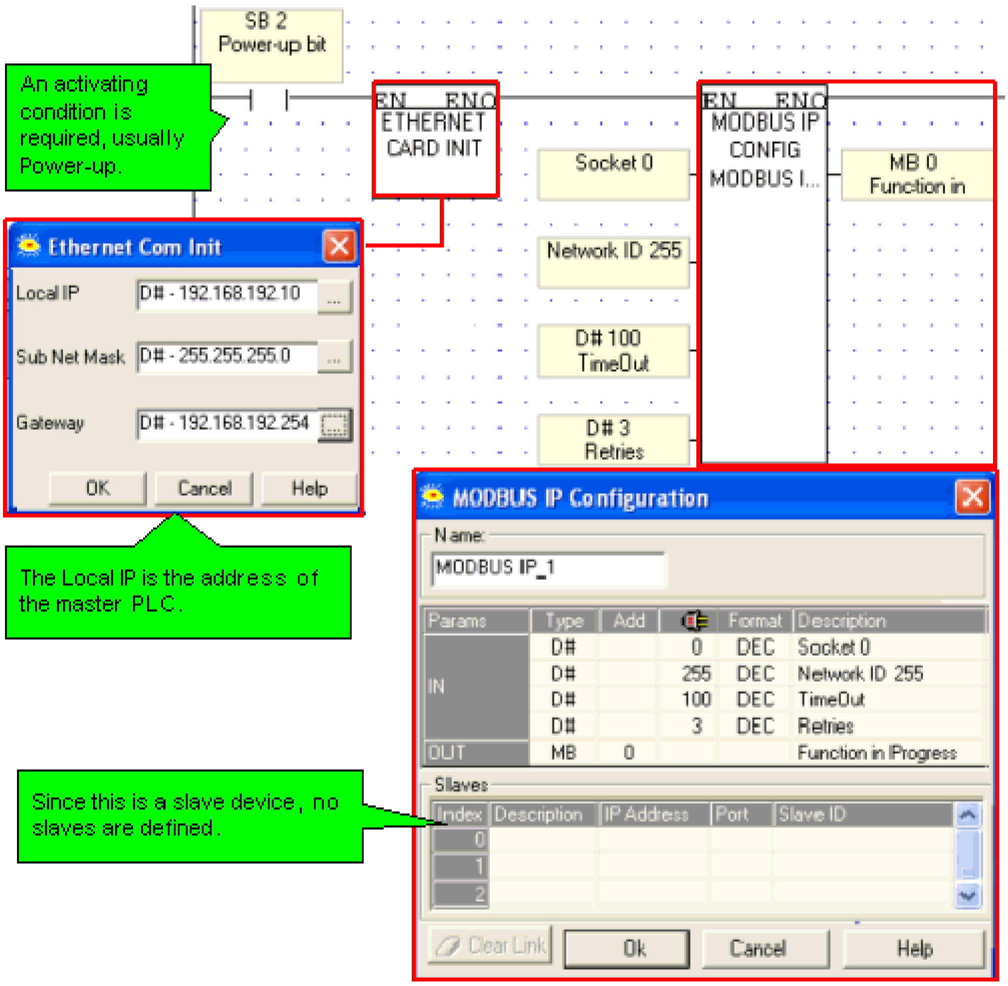
**Step 2: Using MODBUS Commands**

**Note •** Note that the operand addresses in slave PLCs are indirect addresses (pointers). In the figure below,the Slave: Start of Vector parameter is 15. This means that the master will begin reading from MI 15 in the slave PLC. Since the Read: Vector Length parameter is 3, the function takes the values in MI 15, 16 and 17.
The Master: Start of Vector parameter is 17; therefore the values will be written into MI 17, 18, and 19 in the master device.

#### Slave

The slave PLC Ladder application must include the elements shown below.

**Step 1: Initializing the Ethernet card and configuring MODBUS**



An activating condition is required, usually Power-up.

**Ethernet Com Init**

| | |
|---|---|
| Local IP | D# - 192.168.192.10 |
| Sub Net Mask | D# - 255.255.255.0 |
| Gateway | D# - 192.168.192.254 |

The Local IP is the address of the master PLC.

**MODBUS IP Configuration**

Name: MODBUS IP_1

| Params | Type | Add | | Format | Description |
|---|---|---|---|---|---|
| IN | D# | | 0 | DEC | Socket 0 |
| | D# | | 255 | DEC | Network ID 255 |
| | D# | | 100 | DEC | TimeOut |
| | D# | | 3 | DEC | Retries |
| OUT | MB | 0 | | | Function in Progress |

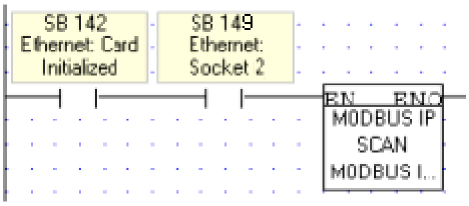
Slaves

| Index | Description | IP Address | Port | Slave ID |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |

Since this is a slave device, no slaves are defined.

### Step 2: Scan

To enable the master PLC to access the slave, include a MODBUS Scan FB in the slave's application.



## TCP: controller-to-controller communication

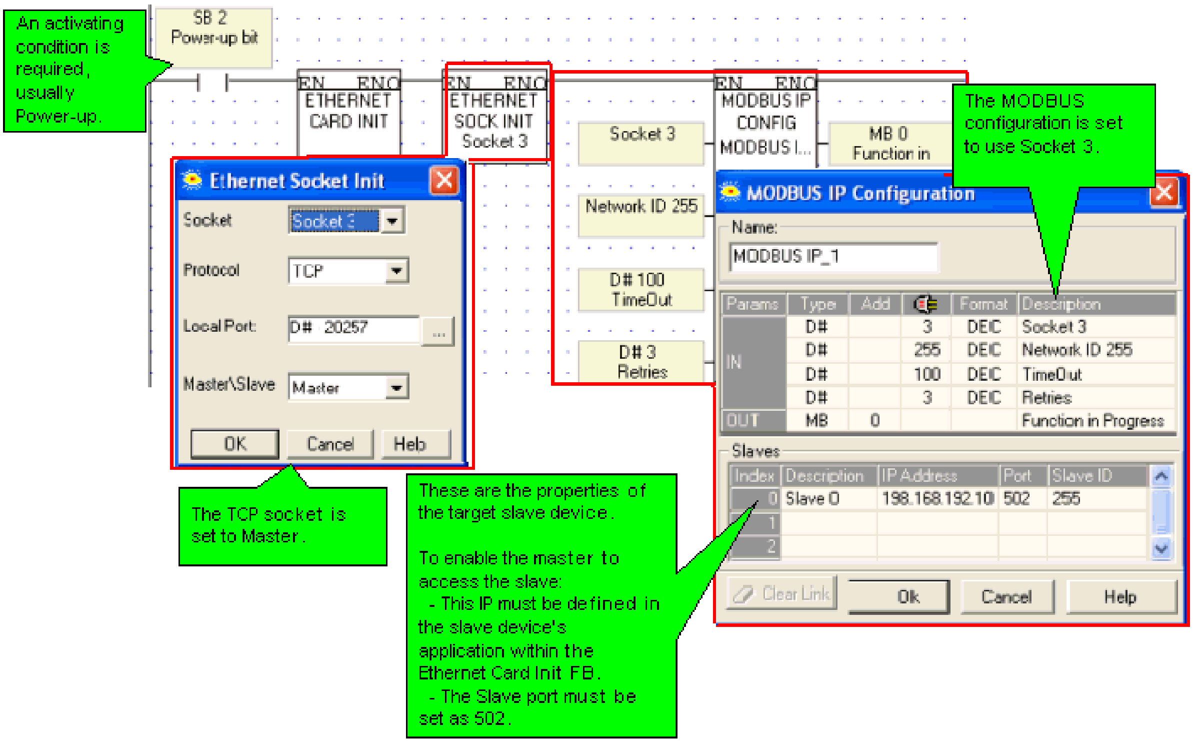
In order to communicate via Ethernet throughout your controller network, you must include an Ethernet Card Init FB in the ladder application of each networked controller. **When using TCP, you must use the Socket: Connect or Socket: Close element.**

Master

The master PLC Ladder application must include the elements shown below.

### Step 1: Initializing the Ethernet card, Socket, and Configuring MODBUS

In the figure below, the socket is configured to use TCP.



An activating condition is required, usually Power-up.

The MODBUS configuration is set to use Socket 3.

The TCP socket is set to Master.

These are the properties of the target slave device.

To enable the master to access the slave:
- This IP must be defined in the slave device's application within the Ethernet Card Init FB.
- The Slave port must be set as 502.

Ethernet Socket Init

Socket: Socket 3
Protocol: TCP
Local Port: D# 20257
Master\Slave: Master

MODBUS IP Configuration

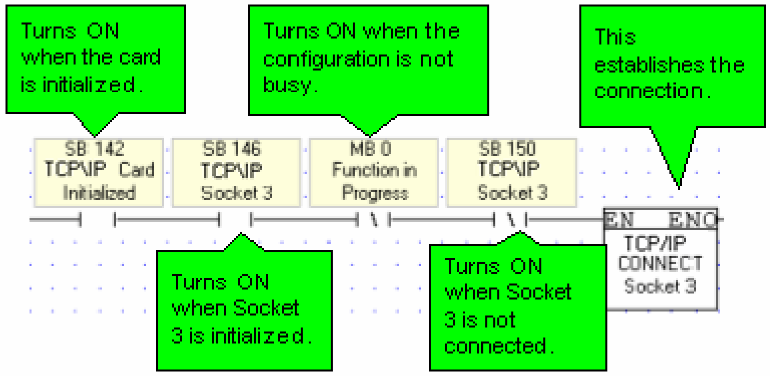
Name: MODBUS IP_1

| Params | Type | Add | | Format | Description |
|---|---|---|---|---|---|
| IN | D# | | 3 | DEC | Socket 3 |
| | D# | | 255 | DEC | Network ID 255 |
| | D# | | 100 | DEC | TimeOut |
| | D# | | 3 | DEC | Retries |
| OUT | MB | 0 | | | Function in Progress |

Slaves

| Index | Description | IP Address | Port | Slave ID |
|---|---|---|---|---|
| 0 | Slave 0 | 198.168.192.10 | 502 | 255 |
| 1 | | | | |
| 2 | | | | |

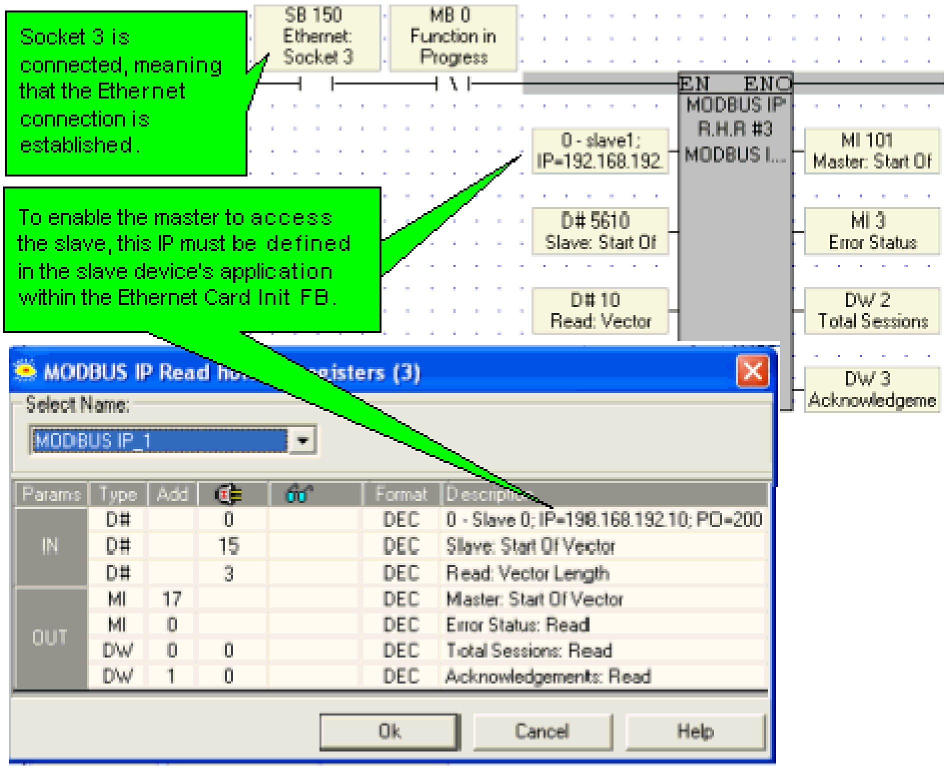
### Step 2: Establishing the Ethernet Connection: Connect Socket


Turns ON when the card is initialized.
Turns ON when the configuration is not busy.
This establishes the connection.
SB 142 TCP\IP Card Initialized
SB 146 TCP\IP Socket 3
MB 0 Function in Progress
SB 150 TCP\IP Socket 3
EN ENO
TCP/IP CONNECT Socket 3
Turns ON when Socket 3 is initialized.
Turns ON when Socket 3 is not connected.

**Note •** It is recommended that there be a time elapse of a few seconds after the Ethernet Card Initialization and before activating Socket Connect. A timer may be used for this purpose.

### Step 3: Using MODBUS Commands

**Note •** Note that the operand addresses in slave PLCs are indirect addresses (pointers). In the figure below, the Slave: Start of Vector parameter is 15. This means that the master will begin reading from MI 15 in the slave PLC. Since the Read: Vector Length parameter is 3, the function takes the values in MI 15, 16 and 17.
The Master: Start of Vector parameter is 17; therefore the values will be written into MI 17, 18, and 19 in the master device.



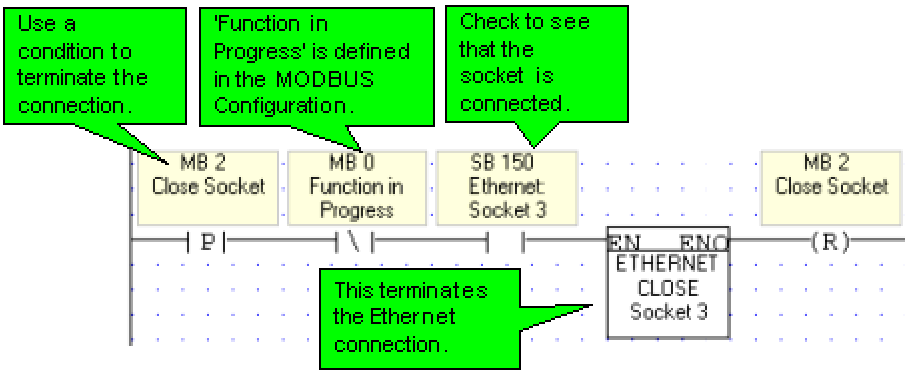
Socket 3 is connected, meaning that the Ethernet connection is established.

To enable the master to access the slave, this IP must be defined in the slave device's application within the Ethernet Card Init FB.

| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 0 | | DEC | 0 - Slave 0; IP=198.168.192.10; PO=200 |
| | D# | | 15 | | DEC | Slave: Start Of Vector |
| | D# | | 3 | | DEC | Read: Vector Length |
| OUT | MI | 17 | | | DEC | Master: Start Of Vector |
| | MI | 0 | | | DEC | Error Status: Read |
| | DW | 0 | 0 | | DEC | Total Sessions: Read |
| | DW | 1 | 0 | | DEC | Acknowledgements: Read |

### Step 4: Terminating the Ethernet connection: Close Socket

When you terminate the connection, use the 'Function in Progress' MB to ensure that you do not terminate the connection while data is being communicated.



Use a condition to terminate the connection.

'Function in Progress' is defined in the MODBUS Configuration.

Check to see that the socket is connected.

This terminates the Ethernet connection.

Slave

The slave PLC Ladder application must include the elements shown below.

### Step 1: Initializing the Ethernet card, Socket, and Configuring MODBUS

In the figure below, the socket is configured to use TCP.

### Step 2: Scan

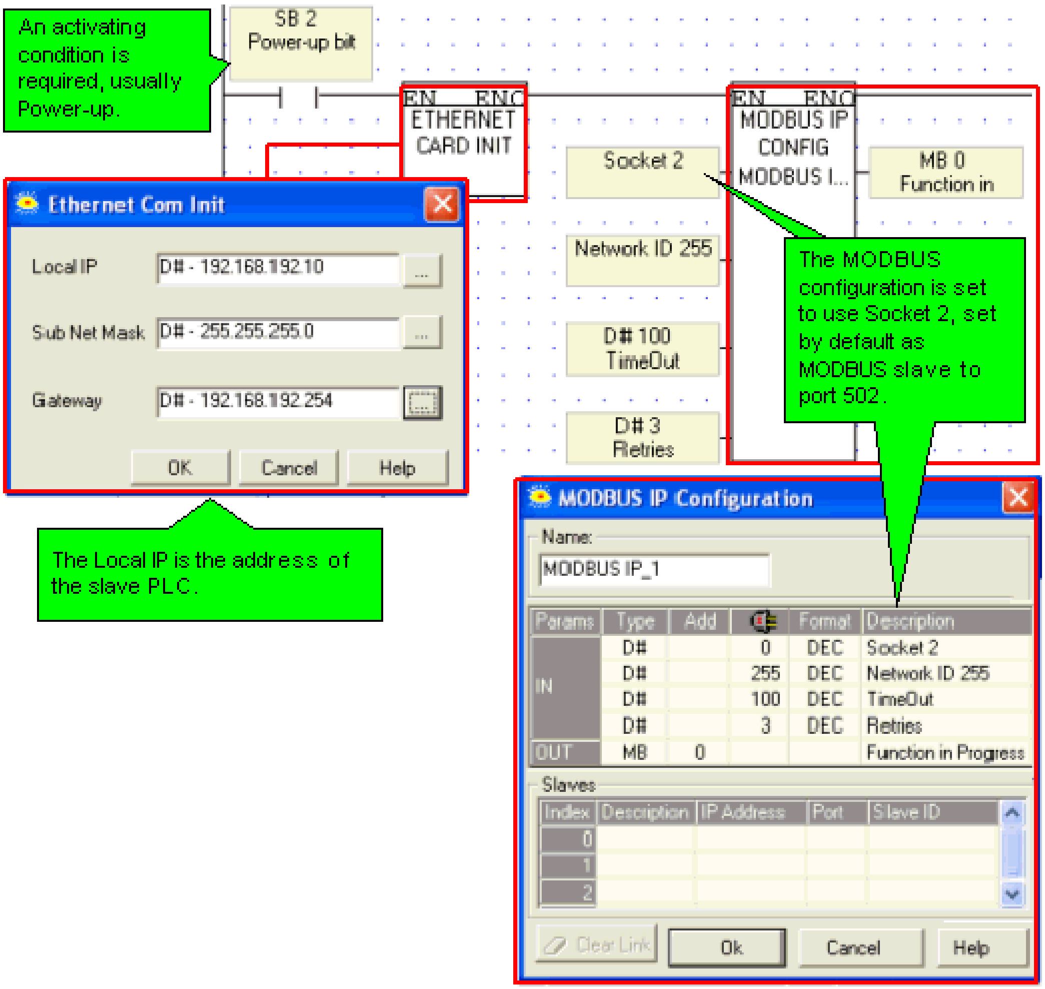
To enable the master PLC to access the slave, include a MODBUS Scan FB in the slave's application.

## PC to PLC: Accessing PLC via SCADA

To enable the SCADA application to access the PLC, the PLC is defined as a slave device. The slave PLC Ladder application must include the elements shown below.

### Step 1: Initializing the Ethernet card and configuring MODBUS

Port 502 is the well-known port for MODBUS applications.



An activating condition is required, usually Power-up.

Local IP: D# - 192.168.192.10

Sub Net Mask: D# - 255.255.255.0

Gateway: D# - 192.168.192.254

The Local IP is the address of the slave PLC.

The MODBUS configuration is set to use Socket 2, set by default as MODBUS slave to port 502.

Name: MODBUS IP_1

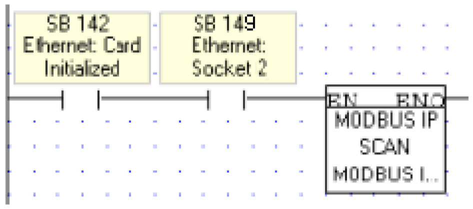
| Params | Type | Add | | Format | Description |
|---|---|---|---|---|---|
| IN | D# | | 0 | DEC | Socket 2 |
| | D# | | 255 | DEC | Network ID 255 |
| | D# | | 100 | DEC | TimeOut |
| | D# | | 3 | DEC | Retries |
| OUT | MB | 0 | | | Function in Progress |

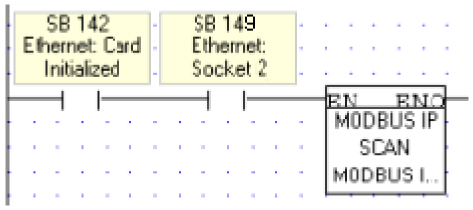
Slaves

| Index | Description | IP Address | Port | Slave ID |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |

### Step 2: Scan

To enable the SCADA application to access the slave, include a MODBUS Scan FB in the slave's application.





## PROFIBUS Slave

A V350 or V130 that is installed with the V100-17-PB1 communication card can function as a PROFIBUS DP slave

Note that the Vision can act **ONLY** as a slave devices, via GSD configuration.

### Implementing Profibus

The PROFIBUS Configuration defines a total of four vectors:

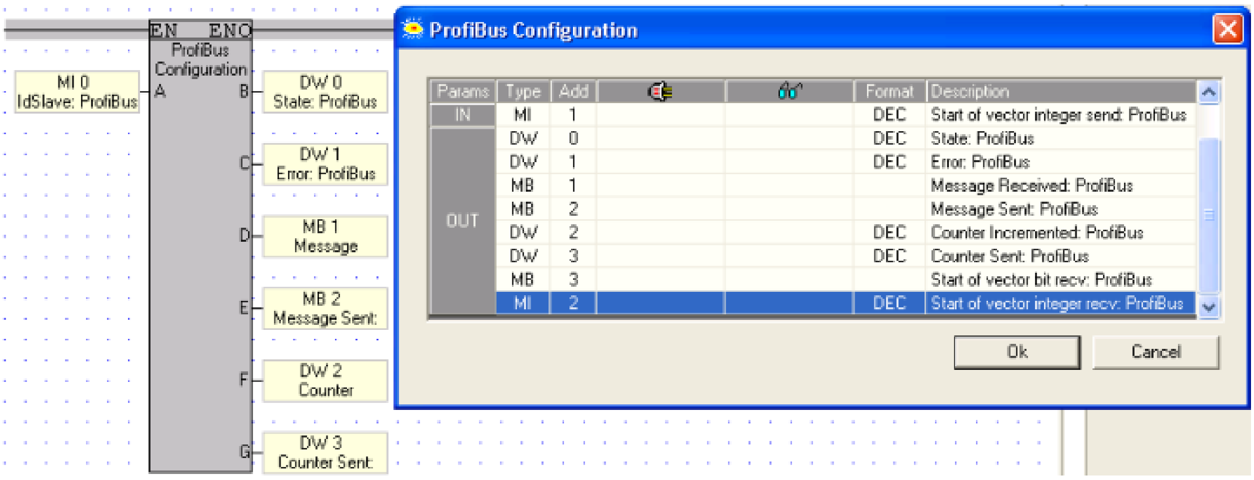
- Two vectors, MB/XB vector and MI/XI, from which data is **sent** at the master's request,
- Two vectors, MB/XB vector and MI/XI, which **receive** data from the master.

Bit vectors are 128 bytes long; integer vectors are 96 integers longs.

**Notes** ♦ An application may contain only one PROFIBUS Configuration.

♦ The PROFIBUS Configuration should be a power-up task.

♦ PROFIBUS is not supported in Interrupt routines.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | MI | 1 | | | DEC | Start of vector integer send: ProfiBus |
| OUT | DW | 0 | | | DEC | State: ProfiBus |
| | DW | 1 | | | DEC | Error: ProfiBus |
| | MB | 1 | | | | Message Received: ProfiBus |
| | MB | 2 | | | | Message Sent: ProfiBus |
| | DW | 2 | | | DEC | Counter Incremented: ProfiBus |
| | DW | 3 | | | DEC | Counter Sent: ProfiBus |
| | MB | 3 | | | | Start of vector bit recv: ProfiBus |
| | MI | 2 | | | DEC | Start of vector integer recv: ProfiBus |

| Parameter | Type | Purpose |
|---|---|---|
| Slave Unit ID | MI | This must be the ID number assigned in the **PROFIBUS masters' configuration.** |
| PROFIBUS: Read, Start Bit Vector | MB, XB | When a master reads the slave's bit data, the vector that is read starts from this bit.<br>Vector length = 128 bytes |
| PROFIBUS:Read, Start Register Vector | MI, XI | When a master reads the slave's integer data, the vector that is read starts from this register.<br>Vector length = 96 registers |
| PROFIBUS: Status messages | DW | (decimal values)<br>0 - Waiting for GSD parameters<br>16 - Waiting for GSD Configuration<br>32 - PROFIBUS Data Exchange in Progress<br>48 - PROFIBUS DP Error |
| PROFIBUS: Error Messages | DW | When PROFIBUS is functioning correctly, this = 1.<br>If there is no master-slave data exchange, and this value is any value other than 1, contact support@unitronics.com |
| PROFIBUS: Bit Read | MB, XB | Turns ON when slave data is read by a master.<br>Reset by user. Do not use a Positive Transition (Rise) contact as a reset condition. |
| PROFIBUS: Register Write | MB, XB | Turns ON when a master writes data to the slave.<br>Reset by user. Do not use a Positive Transition (Rise) contact as a reset condition. |
| PROFIBUS: Counter: # of Reads | DW | Increments each time a master reads the slave. |
| PROFIBUS: Counter: # of Writes | DW | Increments each time a master writes to the slave. |
| PROFIBUS: Write, Start Bit Vector | | When a master writes bit data to the slave, the data is written starting from this bit.<br>Vector length = 128 bytes |
| PROFIBUS:Write, Start Register Vector | | When a master writes integer data to the slave, the data is written starting from this register.<br>Vector length = 96 integers |

# DF1

Use the COM>DF1 Scan function to enable an Enhanced Vision to be accessed by devices using the DF1 AB protocol.

Note that the Vision can act **ONLY** as a slave device.

**Allen-Bradley project**

**Controller Properties**

Set as follows:

- System Protocol: DF1 Master, Error Detection: CRC, Polling Mode: Message Based
- Serial Port Properties:System, 9600, 8, None, 1, No Handshake, 0, 0

Ladder program:

Set up a message call for each data block. The program must toggle messages one by one.

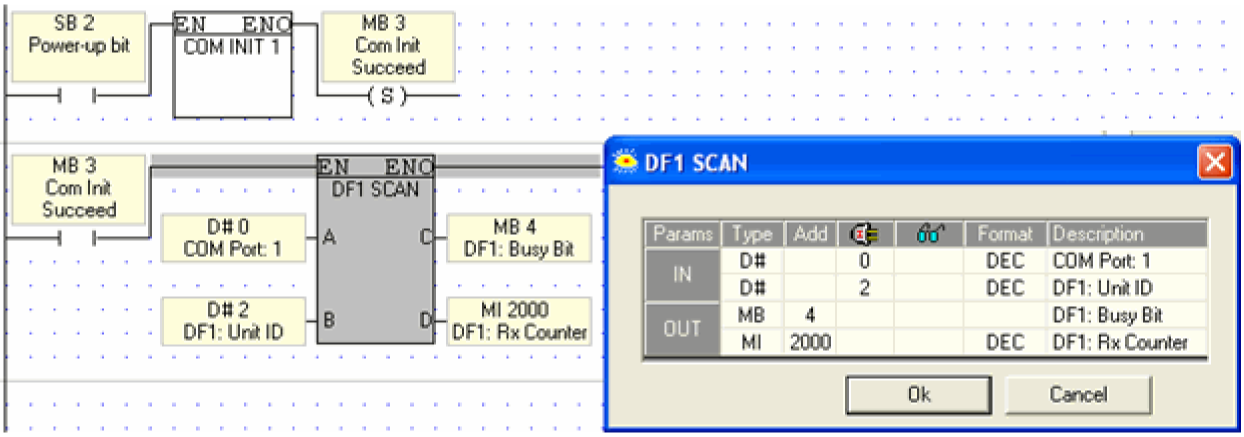
**Message Configuration**

- Select SLC typed Read or Write
- Destination element: note the differences in addressing as shown in the File Number/Type Conversion table.  This shows, for example that A-B N41:# is equivalent to Unitronics' XI memory area. N41:0 is not mapped to XI0; but to XI256. If the destination is set to N7:0, the Unitronics equivalent is MI1792 and up. Floating point data F0 is located to MF0.

Communication:

- Path: Serial port and Receiving Station

The serial port does not require any specific configuration.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 0 | | DEC | COM Port: 1 |
| | D# | | 2 | | DEC | DF1: Unit ID |
| OUT | MB | 4 | | | | DF1: Busy Bit |
| | MI | 2000 | | | DEC | DF1: Rx Counter |

| Parameter | Purpose |
|---|---|
| Com Port | Select any port. |

| | |
|---|---|
| DF1: Unit ID | This must be the ID number assigned in the **Allen-Bradley project.** |
| DF1: Busy Bit | ON while communication is active. |
| DF1 RX Counter | The number of received requests. |

Vision supports 7 DF1 commands:

1. Echo – the PLC returns the exact data received.
2. Unprotected Read – from the DT.
3. Unprotected Write – to DT.
4. Protected Write using 2 Address Fields – Write operands vector*.
5. Protected Read using 2 Address Fields – Read operands vector*.
6. Protected Write using 3 Address Fields – Write operands vector.
7. Protected Read using 3 Address Fields – Read operands vector.

* Does not support Timers and counters

File number / Type conversion table

| **Unitronics terminology** | **File Type** | **File Number** | **Sub Element** | **Example using operand index - #** |
|---|---|---|---|---|
| Output | O | 0-3 | 0 | O0:# |
| Inputs | I | 0-3 | 0 | I1:# |
| MB | B | 0-39 | 0 | B2:# |
| XB | B | 40-79 | 0 | B43:# |
| SB | B | 80-119 | 0 | B90:# |
| MI | N | 0-39 | 0 | N7:# (N7:0=MI1792) |
| XI | N | 40-79 | 0 | N41:# (N41:0=XI256) |
| SI | N | 80-119 | 0 | N90:# |
| MF | F | 0 | 0 | F0:#(F0=MF0) |
| ML | L | 0-39 | 0 | L9:# |
| XL | L | 40-79 | 0 | L59:# |
| SL | L | 80-119 | 0 | L99:# |
| MDW | L | 120-159 | 0 | L127:# |
| XDW | L | 160-199 | 0 | L177:# |
| SDW | L | 200-239 | 0 | L208:# |
| Counter Preset | C | 0 | 1 | C0:#.PRE |
| Counter Current | C | 0 | 2 | C0:#.ACC |
| Counter Bit | C | 0 | 13 | C0:#.DN |
| Timer Preset | T | 0 | 1 | T0:#.PRE |
| Timer Current | T | 0 | 2 | T0:#.ACC |
| Timer Bit | T | 0 | 13 | T0:#.DN |

**Notes •** In the event that operand types share file types (MI/XI/SI), each operand type receives 40 file numbers.

**Commands**

| **Command Name** | **Purpose** |
|---|---|
| **Echo** | Tests Link.<br>Field: Data to be Received. |
| **Unprotected Read** | Reads from PLC Data Tables<br>1 Field:<br>• Address – Word.<br>The address is WORD address. There is no way to read odd addresses, as most of SLC/5 does.<br>• Bytes To Read – Byte<br>Maximum: limited to 244 by the protocol, but is not limited by the PLC. |
| **Unprotected Write** | Writes to PLC Data Tables<br>2 Fields:<br>• Address – Word.<br>The address is WORD address. There is no way to read odd addresses, |

| | |
|---|---|
| | as most of SLC/5 does.<br>• Bytes To Read – Byte<br>Maximum: limited to 244 by the protocol, but is not limited by the PLC. |
| **Protected Read, 2 Address Fields** | Reads operand vector.<br>4 fields:<br>• Bytes to be read (bytes, not operands number)<br>• File Number + File Type (parsed to **operand type**)<br>• Element Number (operand index)<br>This command cannot read timers and counters (for this, use a command with 3 address fields)<br>Refer to the table in section 1 for File number / Type details |
| **Protected Write, 2 Address Fields** | Writes to operand vector.<br>5 fields:<br>• Bytes to be read (bytes, not operands number)<br>• File Number + File Type (parsed to **operand type**)<br>• Element Number (operand index)<br>• Data to be written<br>This command cannot read timers and counters (for this, used command with 3 address fields)<br>Refer to the table in section 1 for File number / Type details |
| **Protected Read, 3 Address Fields** | Reads operands vector.<br>5 fields:<br>• Bytes to be read (bytes, not operands number)<br>• File Number + File Type (parsed to **operand type**)<br>• Element Number (operand index)<br>• Sub Element (used for Preset / current / bit. 0 for other operands type)<br>Refer to the table in section 1 for File number / Type details |
| **Protected Write, 3 Address Fields** | Writes to operands vector.<br>6 fields:<br>• Bytes to be read (bytes, not operands number)<br>• File Number + File Type (parsed to **operand type**)<br>• Element Number (operand index)<br>• Sub Element (used for Preset / current / bit. 0 for other operands type)<br>• Data to be written<br>Refer to the table in section 1 for File number / Type details |
| | |

## ASCII Character Table

| Name | Hex | Dec |
|---|---|---|
| . (period) | 2E | 046 |
| 0 | 30 | 048 |
| 1 | 31 | 049 |
| 2 | 32 | 050 |
| 3 | 33 | 051 |
| 4 | 34 | 052 |
| 5 | 35 | 053 |
| 6 | 36 | 054 |
| 7 | 37 | 055 |
| 8 | 38 | 056 |
| 9 | 39 | 057 |

| Name | Hex | Dec |
|---|---|---|
| A | 41 | 065 |
| B | 42 | 066 |
| C | 43 | 067 |
| D | 44 | 068 |
| E | 45 | 069 |
| F | 46 | 070 |
| G | 47 | 071 |
| H | 48 | 072 |
| I | 49 | 073 |
| J | 4A | 074 |
| K | 4B | 075 |

| Name | Hex | Dec |
|---|---|---|
| L | 4C | 076 |
| M | 4D | 077 |
| N | 4E | 078 |
| O | 4F | 079 |
| P | 50 | 080 |
| Q | 51 | 081 |
| R | 52 | 802 |
| S | 53 | 083 |
| T | 54 | 084 |
| U | 55 | 085 |
| V | 56 | 086 |

| Name | Hex | Dec |
|---|---|---|
| W | 57 | 087 |
| X | 58 | 088 |
| Y | 59 | 089 |
| Z | 5A | 090 |

# PC-PLC Communications

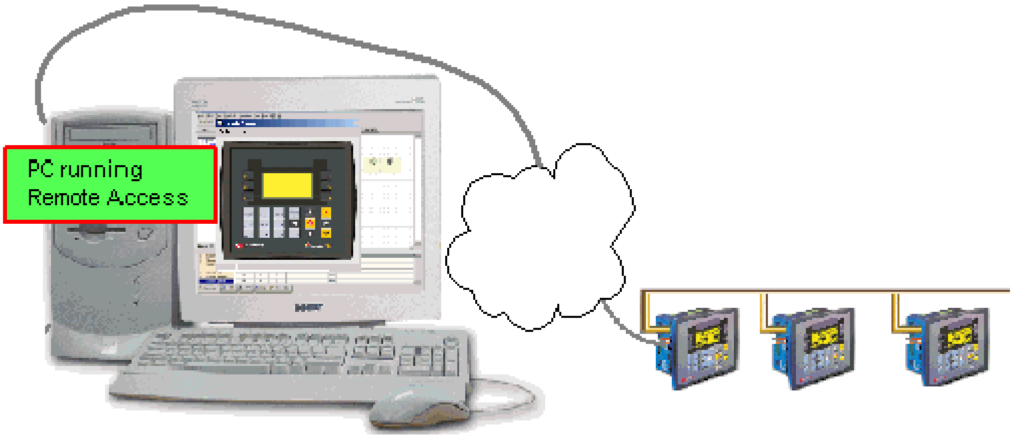
## Remote Access: Accessing a PLC via PC

Use VisiLogic to activate access a remote Vision controller and:

- Download and upload projects
- Remotely operate the controller's HMI through your PC
- Run On-Line Test mode on the remote controller
- Run Information Mode.

You can access:

- Stand-alone controllers that are directly connected to the PC via a cable.
- Controllers within a CANbus or TCP/IP network
- Either stand-alone or networked controllers via GSM or landline modem.



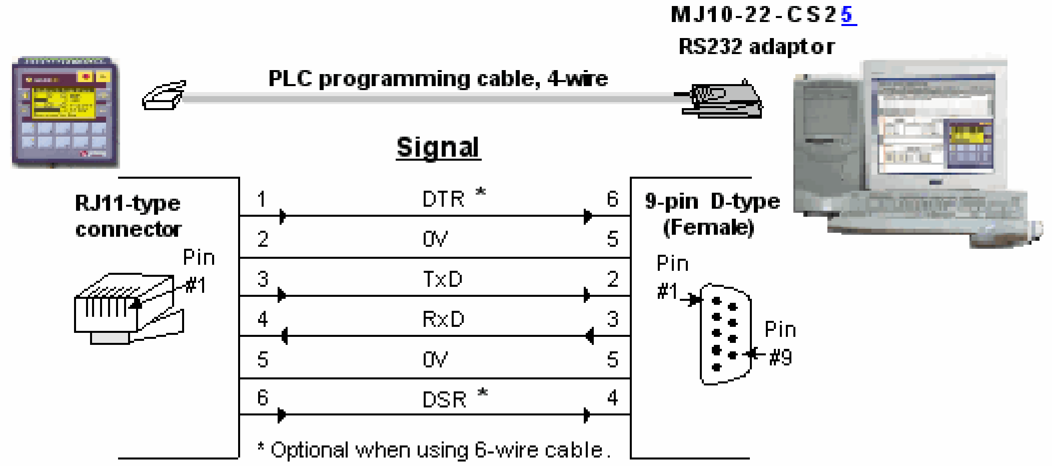
**Note •** In addition to using VisiLogic to access a remote Vision, Unitronics provides a stand-alone utility called Remote Access. This utility can also access Unitronics M90/91 and Jazz controllers. It may be freely downloaded from http://www.unitronics.com.

Accessing a PLC via VisiLogic

Before you can access a controller, you must establish a communication link:

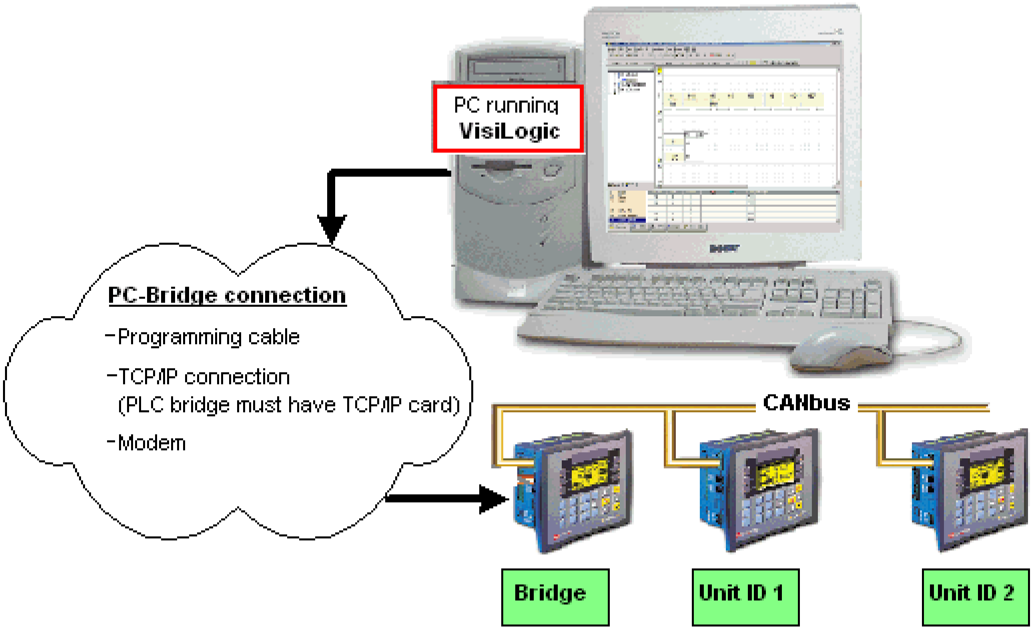
### Direct Connection: PC-Controller

1. Connect your PC to any controller using the programming cable supplied with the controller kit.
   In the case of the V1040, the USB cable may be used; note that COM port 1 function is suspended when this port is physically connected to a PC



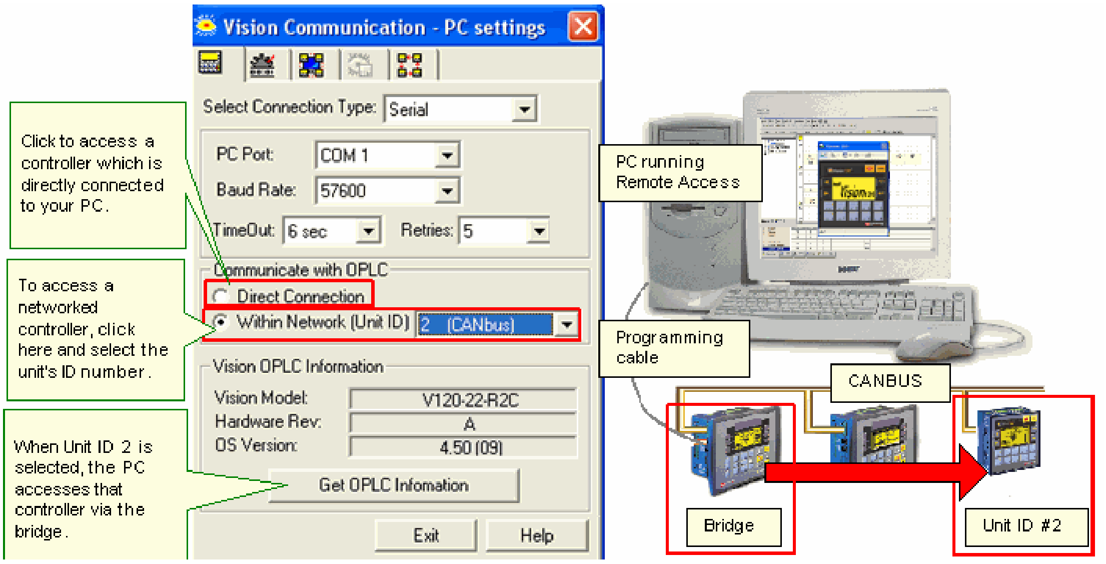
### Accessing a Networked Controller

1. Connect your PC to any controller in the network using the programming cable supplied with the controller kit.



**Note •** Different PCs can access a network at the same time, using different controller units as bridges. However, 2 different PCs cannot simultaneously access the same controller unit.

2. Select a networked controller by opening Communication & OS from the Connection menu, and then entering the Unit ID number.



Vision Communication - PC settings

Select Connection Type: Serial

PC Port: COM 1

Baud Rate: 57600

TimeOut: 6 sec Retries: 5

Communicate with OPLC

○ Direct Connection

● Within Network (Unit ID) 2 (CANbus)

Vision OPLC Information

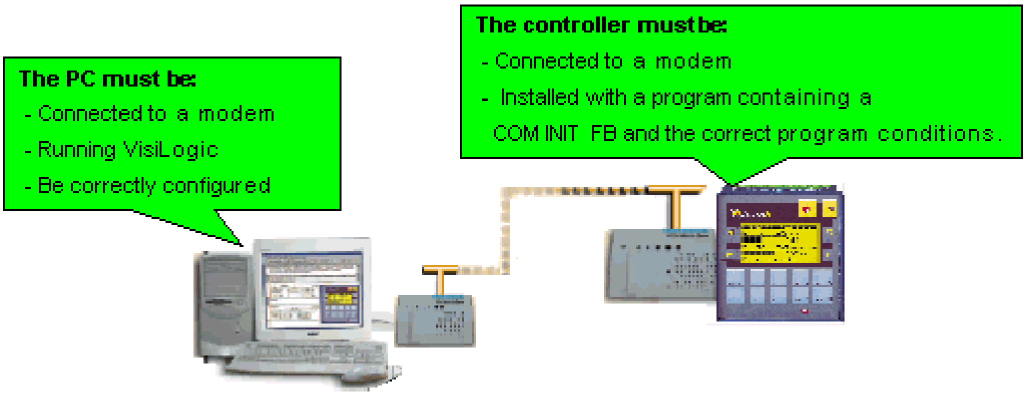
Vision Model: V120-22-R2C

Hardware Rev: A

OS Version: 4.50 (09)

Get OPLC Information
Exit
Help
Click to access a controller which is directly connected to your PC.
To access a networked controller, click here and select the unit's ID number.
When Unit ID 2 is selected, the PC accesses that controller via the bridge.
PC running Remote Access
Programming cable
CANBUS
Bridge
Unit ID #2

## Accessing a Controller via Modem



The PC must be:

- Connected to a modem
- Running VisiLogic
- Be correctly configured

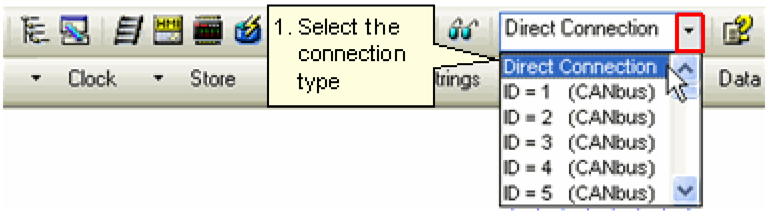
The controller must be:

- Connected to a modem
- Installed with a program containing a COM INIT FB and the correct program conditions.

1. Prepare and connect your PLC-side modem as described in the topic PLC-side Modems, in the section 'How to enable a controller to communicate via landline, GSM/GPRS modem'.
2. Prepare the PC-side modem as described in the topic PC-Side Modems (Modem Services), in the section 'How to Configure a PC-side Modem'.
3. Via Connection>Modem Services, dial the remote PLC's controller to establish the data link.

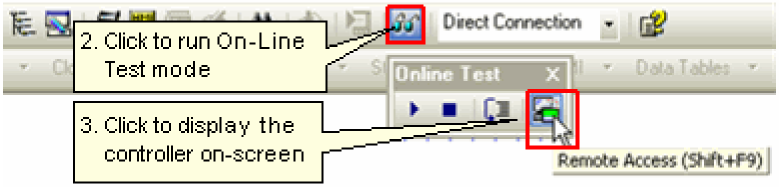
After completing the preceding steps according to your communication type, proceed as follows:

1. Select a connection type using the drop-down selection box on the toolbar.



1. Select the connection type

2. Click the On-line Test mode button or press <F9> to enter On-Line Test mode; the left Ladder rail turns red and real-time values are displayed in the Output window.
3. Click the Remote Access button on the On-line Test toolbar to display the remote controller on your PC screen. You can toggle the controller image on and off using <Shift>+<F9>.

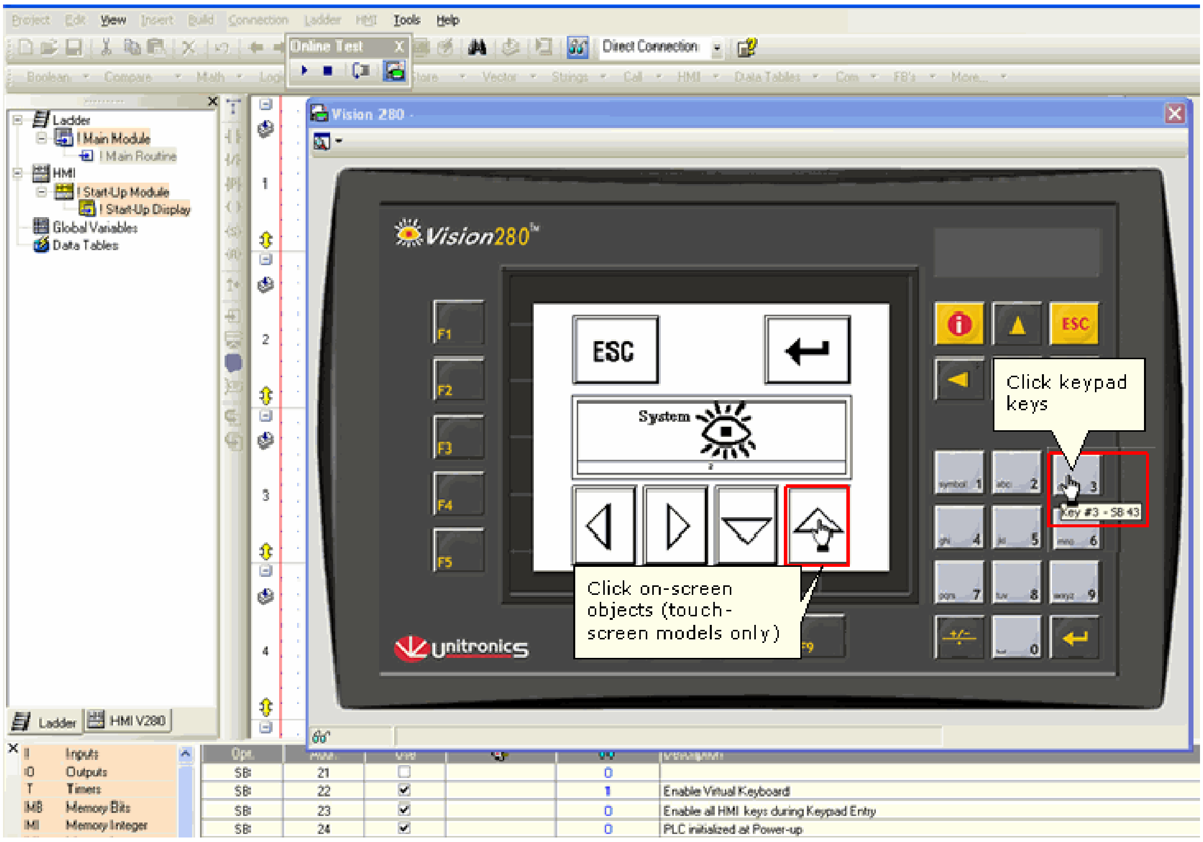


2. Click to run On-Line Test mode

3. Click to display the controller on-screen

4. Remotely operate the controller's HMI by using:
   - Your PC keyboard, pressing arrow, alphanumeric, and function keys <F1> to <F8>). Note that the Vision <ESC> key is the <E> key on the PC keyboard
   - Your mouse to click keypad keys on the Remote Access image on the PC screen. In the case of touch-screen models, you can also click on-screen objects.

To enter Information Mode, press the <i> key on your PC keyboard, or by clicking it on-screen with your cursor.



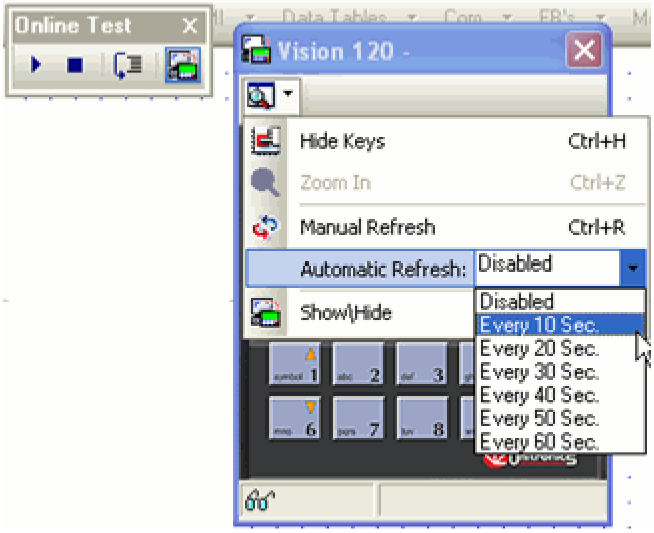
Click keypad keys

Click on-screen objects (touch-screen models only)

| Opr. | Addr. | | | | |
|---|---|---|---|---|---|
| SB | 21 | □ | | 0 | |
| SB | 22 | ☑ | | 1 | Enable Virtual Keyboard |
| SB | 23 | ☑ | | 0 | Enable all HMI keys during Keypad Entry |
| SB | 24 | ☑ | | 0 | PLC initialized at Power-up |

### Remote Access options

Use the options to set display options and refresh rate.



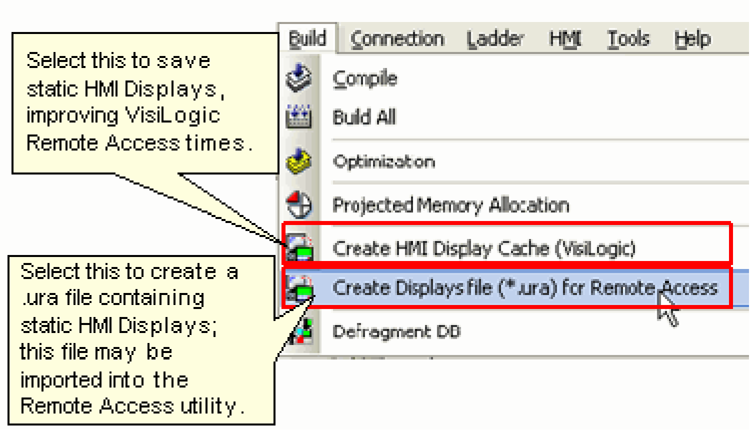
**Note •** The Zoom option can be activated only if you select Hide Keys. Zoom cannot be used with or V280 controllers.

**Improving Remote Access run times:**

Cache files enable Displays to load more quickly. These files enable Remote access to refer to HMI elements stored in the PC, instead of taking them from the PLC. If you have been provided with a static HMI file, select it to improve Remote Access run times.

Monochrome Vision only

- To use a temporary memory cache during a session, select Project> Cache from the Build menu.
- The HMI cache files are in .ura format. To create a .ura file containing static displays, select Export Displays to

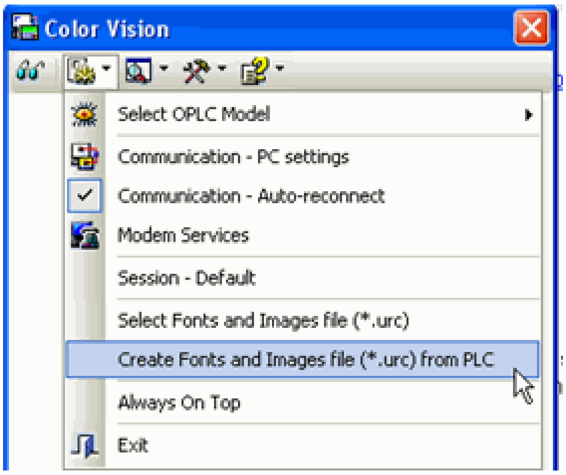

Build
Connection
Ladder
HMI
Tools
Help
Compile
Build All
Optimization
Projected Memory Allocation
Create HMI Display Cache (VisiLogic)
Create Displays file (*.ura) for Remote Access
Defragment DB

Select this to save static HMI Displays, improving VisiLogic Remote Access times.

Select this to create a .ura file containing static HMI Displays; this file may be imported into the Remote Access utility.

Color Vision only

The static HMI files are in .urc format. Such files can be created in Remote Access by selecting the option Create Fonts and Images (*.urc) from PLC shown in the following figure. The .urc file may include either fonts, images, or both. However, note that if the .urc file does not include graphics, Remote Access will not display images.


Color Vision
Select OPLC Model
Communication - PC settings
Communication - Auto-reconnect
Modem Services
Session - Default
Select Fonts and Images file (*.urc)
Create Fonts and Images file (*.urc) from PLC
Always On Top
Exit

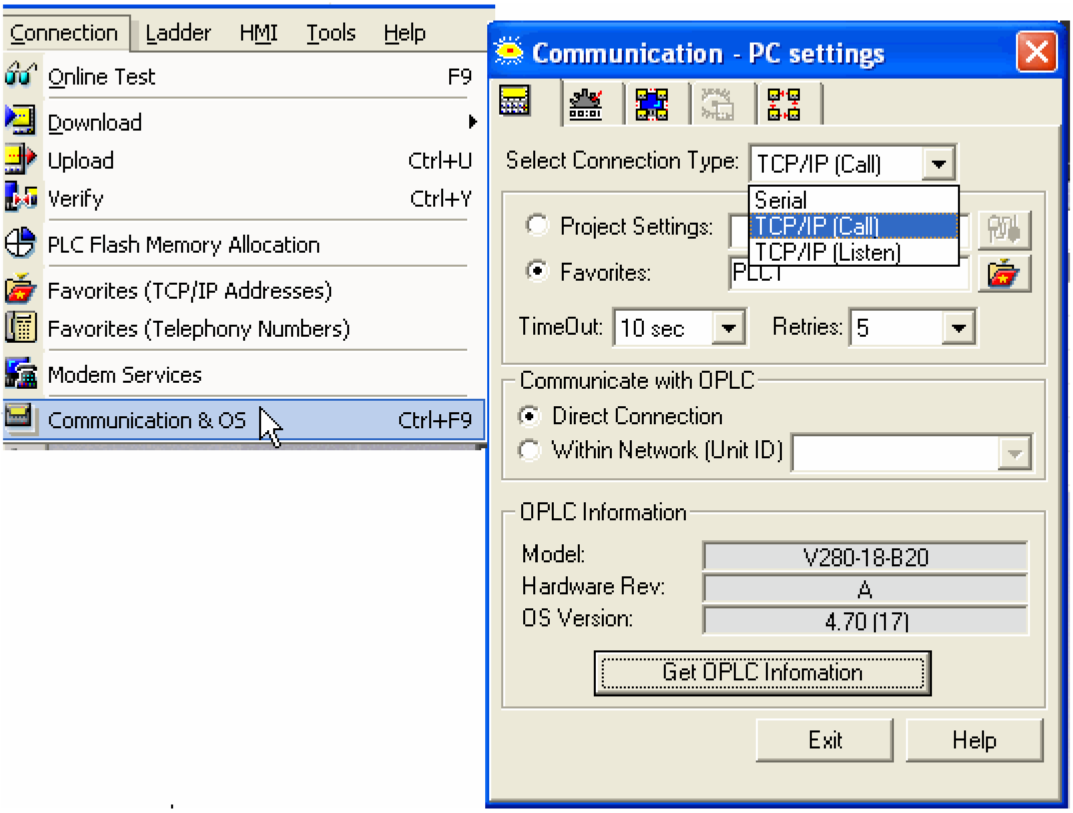
**Note**

- When Remote Access creates a .urc file, the Vision enters 'System Mode'; the PLC continues running while displaying a system image. The HMI application is not visible.
  In addition, note that an interruption in communications may leave the PLC inaccessible. In this case, the PLC may require reset, which requires an operator to be **physically** present near the PLC.

Vision Communication PC Settings

This defines the connection VisiLogic will use when downloading a program or carrying out other communication tasks. To display the current communication settings, select Communication & OS from the Connection menu.

Note that you can cause the Unit ID# to be permanently assigned to the project via Project Properties.



Select Connection Type: TCP/IP (Call)

○ Project Settings:

● Favorites: PLC1

TimeOut: 10 sec Retries: 5

Communicate with OPLC

● Direct Connection

○ Within Network (Unit ID)

OPLC Information

Model: V280-18-B20

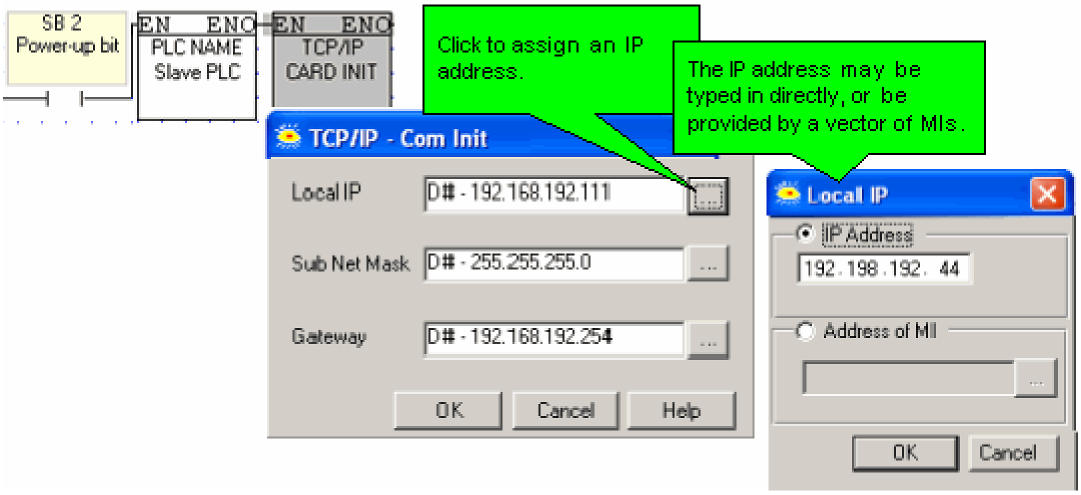
Hardware Rev: A

OS Version: 4.70 (17)

| | |
|---|---|
| **Select Connection Type** | If your Vision contains an Ethernet port, you can select the TCP/IP options. Serial is the default communication mode; note that if you select TCP/IP and close the project, the setting reverts to Serial. |
| **PC COM Parameters** | Port, Retries and Time-Out are the communication settings between VisiLogic and the controller.<br>**Note** • If you are working with a network, the TimeOut should be greater than 1 second.<br>• If you are working with a GPRS modem, set the TimeOut to its maximum of 10 seconds. |
| **Communicate with OPLC** | Use these options to communicate with networked controllers.<br>**Direct Connection**: select this to communicate with any controller that is connected to your PC via the download cable, including a network bridge.<br>**Within Network**:select this to communicate with a controller that is integrated into a network, then select the controller's ID number<br>**Note** • ID numbers 1-63 are reserved for controllers linked via CANbus; ID numbers 64-127 are reserved for controllers networked via RS485. Using this range of ID numbers prevents a polled controller from attempting to act as a CANbus bridge, preventing it from attempting to locate the requested controller. |
| **Vision OPLC Information** | Click **Get OPLC Information** to display information about the controller you have selected in **Communicate with OPLC.** |

### TCP\IP: Card Init

This function is located on the Com>TCP/IP menu.



Click to assign an IP address.

The IP address may be typed in directly, or be provided by a vector of MIs.

Local IP: D# - 192.168.192.111

Sub Net Mask: D# - 255.255.255.0

Gateway: D# - 192.168.192.254

● IP Address: 192 . 198 . 192 . 44

○ Address of MI

If you assign an IP address indirectly, via an MI vector, note that the vector is 4 MIs long. The low byte of each MI provides the number for an octet within the IP address.

If, for example, the IP address is linked to MI 0, and the low bytes of MI 0 to MI 3 contain the values 192, 198, 192, 45, the IP address will be 192.198.192. 45.

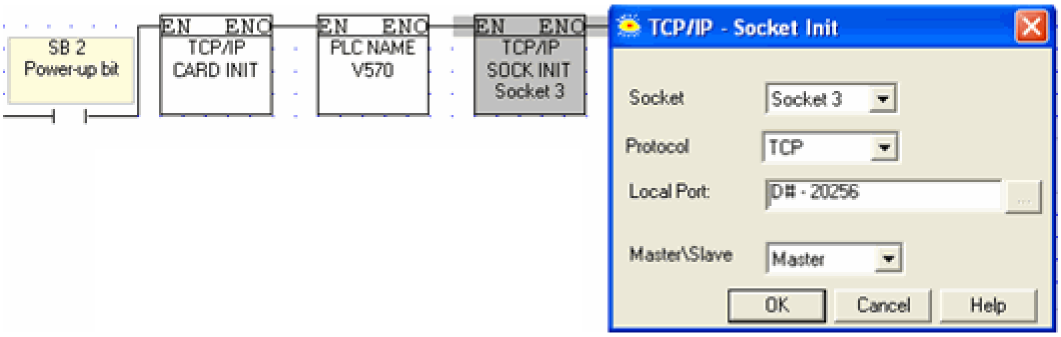
| | |
|---|---|
| **Note •** | In order to implement Ethernet, a controller must be assigned an IP address. This is done via the TCP\IP Init FB, which must be included in the Ladder applications of both master and slave controllers. Information on IP addressing is given in the topic About Ethernet |
| • | When the Ethernet card finishes initialization, SB 142 rises. Use this as a condition before activating any Ethernet element, such as Socket: Connect. |
| • | An activating condition must be placed before the Ethernet Card Init FB. This may be assigned as a power-up task; however a one-shot transitional contact may also be used. |
| • | If you have linked the IP address to a vector of MIs, and this condition is not activated, the IP address will not be assigned to the controller. Make sure, for example, that if you have used a power-up condition, that the controller does go through power-up. |

### TCP\IP: Socket Init

This function is located on the Com>TCP/IP menu.

Vision controllers currently offer 4 sockets. Sockets may be configured to TCP, UDP, UDP RAW, or HTTP.

The default configuration means that, for most applications, you do not need to include a Socket Init FB in the ladder application. However, if, for example, your application requires 4 sockets for TCP, change the default configuration of Socket 0 from UDP to TCP via the Socket Init FBs.



TCP/IP - Socket Init

Socket: Socket 3

Protocol: TCP

Local Port: D# - 20256

Master\Slave: Master

The default socket configuration includes:

| Socket | Protocol | Port Number | Function |
|---|---|---|---|
| 0 | UDP | 20,000 | Enables data to be both transmitted and received within a PLC network, via MODBUS.<br>**Note** • If you are using the default settings for Socket 0, note that data is sent to IP: 255.255.255.255. port: 20,000 plus the last byte of the IP address originally assigned to the device.<br>This is why Port numbers 20,000-20,255 are reserved for Socket 0. |
| 1 | TCP | 20,256 | Enables PC to PLC communication via UnCmDrv1.dll, including VisiLogic, Remote Access, and other Unitronics communication applications. |
| 2 | TCP | 502 | Set to 'listen' as slave (server), enables MODBUS applications such as OPC servers and SCADA systems which use MODBUS TCP over IP. |
| 3 | TCP | 20,257 | Set to 'listen' as slave (server), enables non-Unitronics PLCs to access Unitronics PLCs, via MODBUS. |

**Note** •

UDP Unicast (device to device)
- Socket 0 is set to broadcast by default. To set it to work via UDP (device to device)by turning SB 159 OFF.
- If you set Sockets1-3 to UDP, they will be in Unicast mode.

•

Enabling VisiLogic to communicate with a PLC over UDP
-- Socket 0 can be set to work via UDP Unicast by turning SB 159 OFF, and then running Socket Init to initialize Socket 0 to UDP.

•

Select TCP Master to configure a socket to enable the PLC to send e-mail.

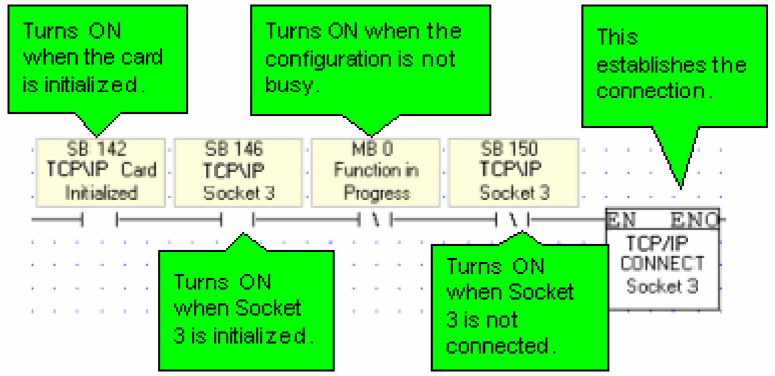
•

Select HTTP to configure a socket to enable the PLC to function as a Web Server.

### TCP\IP: TCP Connect \ TCP Close

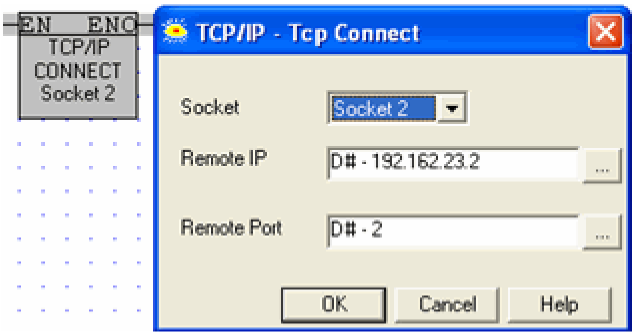
TCP applications require you to use a TCP: Connect FB to establish the Ethernet connection after the Ethernet card is initialized and before activating any of the MODBUS IP commands.

To terminate the session, use the TCP: Close FB. Both elements are located on the **Com>TCP/IP menu**.


Turns ON when the card is initialized.
Turns ON when the configuration is not busy.
This establishes the connection.
SB 142 TCP\IP Card Initialized
SB 146 TCP\IP Socket 3
MB 0 Function in Progress
SB 150 TCP\IP Socket 3
EN ENO
TCP/IP CONNECT Socket 3
Turns ON when Socket 3 is initialized.
Turns ON when Socket 3 is not connected.

The TCP Connect function dedicates a socket to communicate with the device whose IP is specified in the function, causing it to ignore communication attempts from other devices.

Such is not the case with UDP. Since there is no formal handshake, communications can continue to flow through a socket even when there are multiple requests.



Socket: Socket 2

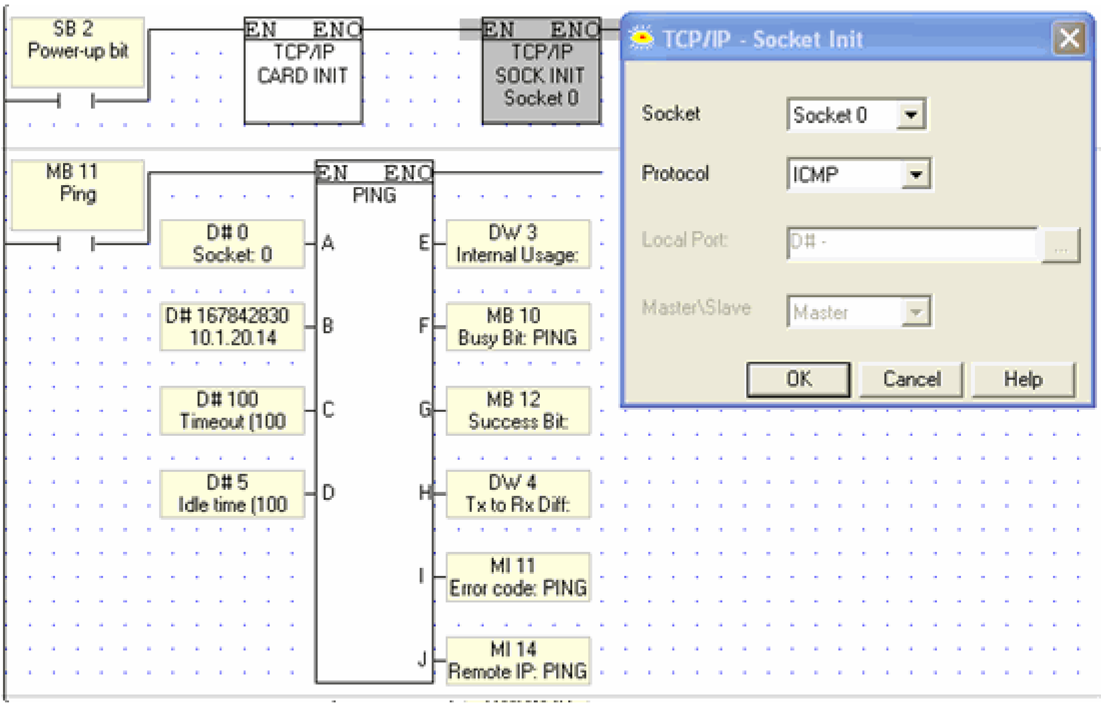
Remote IP: D# - 192.162.23.2

Remote Port: D# - 2

## Ping

Use the Com>TCP/IP> Ping function to ping a remote device.

**Note •** The socket used to send Ping data must be initialized to ICMP.



Socket: Socket 0

Protocol: ICMP

Local Port: D# -

Master\Slave: Master

| Parameter Name | Type | Purpose |
|---|---|---|
| Socket | Constant | The PLC sends the data through this socket. Select a socket that is initialized to ICMP |
| Remote PLC | Constant, MI, XI | The IP of the remote device |
| Timeout (100ms) | Constant or register | User-defined. If no answer is received from the remote device within the Timeout period, the function waits for the Idle time to pass before resending the Ping data. |
| Idle time (100ms) | Constant or register | If no answer is received from the remote device, the function wait for the Timeout + Idle Time before resending the Ping request.<br>If a reply is received, the function wait for the Idle time, before resending. |
| Internal usage | DW, XDW | The function uses this for internal management. Set a Power-up value of 0. |
| Busy Bit | MB, XB | Turns ON when the function begins waiting for a reply<br>Turn OFF when the reply arrives, or when timeout is exceeded. |
| Success Bit | MB, XB | Turns ON when reply is received without error.<br>Reset by user. |
| Rx to Tx Time | DW, XDW | Resolution:<br>• Standard: 2.5 mSec<br>• Enhanced: 1 µSec<br>Valid value ONLY when the Success bit rises. |

| | | |
|---|---|---|
| Error Code | Register | Ping supports the standard ICMPv4 Destination Unreachable Message Subtypes listed in the following table |

| **Error Codes (ICMPv4)** | **Purpose** |
|---|---|
| -2 | Timeout exceeded– no reply at all |
| -1 | No error |
| 0 | Network Unreachable<br>The datagram could not be delivered to the network specified in the network ID portion of the IP address. Usually means a problem with routing but could also be caused by a bad address. |
| 1 | Host Unreachable<br>The datagram was delivered to the network specified in the network ID portion of the IP address but could not be sent to the specific host indicated in the address. Again, this usually implies a routing issue. |
| 2 | Protocol Unreachable<br>The protocol specified in the Protocol field was invalid for the host to which the datagram was delivered. |
| 3 | Port Unreachable<br>The destination port specified in the UDP or TCP header was invalid. |
| 4 | Fragmentation Needed and DF Set<br>Normally, an IPv4 router will automatically fragment a datagram that it receives if it is too large for the maximum transmission unit (MTU) of the next physical network link the datagram needs to traverse. However, if the DF (Don't Fragment) flag is set in the IP header, this means the sender of the datagram does not want the datagram ever to be fragmented. This puts the router between the proverbial rock and hard place, and it will be forced to drop the datagram and send an error message with this code.<br>This message type is most often used in a “clever” way, by intentionally sending messages of increasing size to discover the maximum transmission size that a link can handle. This process is called MTU path discovery. |
| 5 | Source Route Failed<br>Generated if a source route was specified for the datagram in an option but a router could not forward the datagram to the next step in the route. |
| 6 | Destination Network Unknown<br>Not used; Code 0 is used instead. |
| 7 | Destination Host Unknown<br>The host specified is not known. This is usually generated by a router local to the destination host and usually means a bad address. |
| 8 | Source Host Isolated<br>Obsolete, no longer used. |
| 9 | Communication with Destination Network is Administratively Prohibited<br>The source device is not allowed to send to the network where the destination device is located. |
| 10 | Communication with Destination Host is Administratively Prohibited<br>The source device is allowed to send to the network where the destination device is located, but not that particular device. |
| 11 | Destination Network Unreachable for Type of Service<br>The network specified in the IP address cannot be |

| | | | |
|---|---|---|---|
| | | | reached due to inability to provide service specified in the Type Of Service field of the datagram header. |
| | | 12 | Destination Host Unreachable for Type of Service<br>The destination host specified in the IP address cannot be reached due to inability to provide service specified in the datagram's Type Of Service field. |
| | | 13 | Communication Administratively Prohibited<br>The datagram could not be forwarded due to filtering that blocks the message based on its contents. |
| | | 14 | Host Precedence Violation<br>Sent by a first-hop router (the first router to handle a sent datagram) when the Precedence value in the Type Of Service field is not permitted. |
| | | 15 | Precedence Cutoff In Effect<br>Sent by a router when receiving a datagram whose Precedence value (priority) is lower than the minimum allowed for the network at that time. |
| Remote IP | | The IP from the replying device. Note that it might be different than the “input Remote IP” due to network topology. | |

### Send e-mail

This function enables a controller to send an e-mail in response to Ladder conditions. In order to send e-mail, the controller must comprise a TCP/IP port, and must be connected to an Ethernet network with access to a mail server.

Before you begin you need the following information:

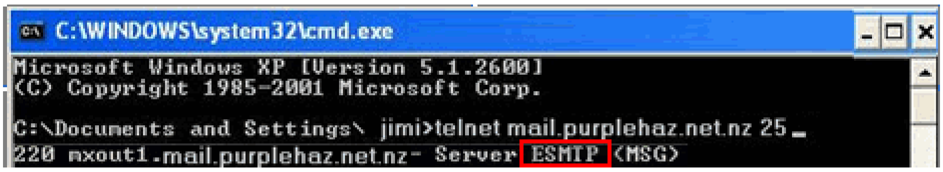
- The type of protocol your mail server uses, SMTP or ESMTP
- Your mail server's IP address

There are sample applications that show how e-mail may be sent via Ethernet, and via GPRS modem.

#### Determining the Protocol Type

To determine which protocol your server supports, run Telnet.

1. Click Windows Start > Run, then enter CMD.
2. In the CMD window, enter the command: telnet RemoteMailServer 25, where RemoteMailServer is the name of your mail server, and 25 is the port via which you contact the server. The command returns the protocol type used by the server.



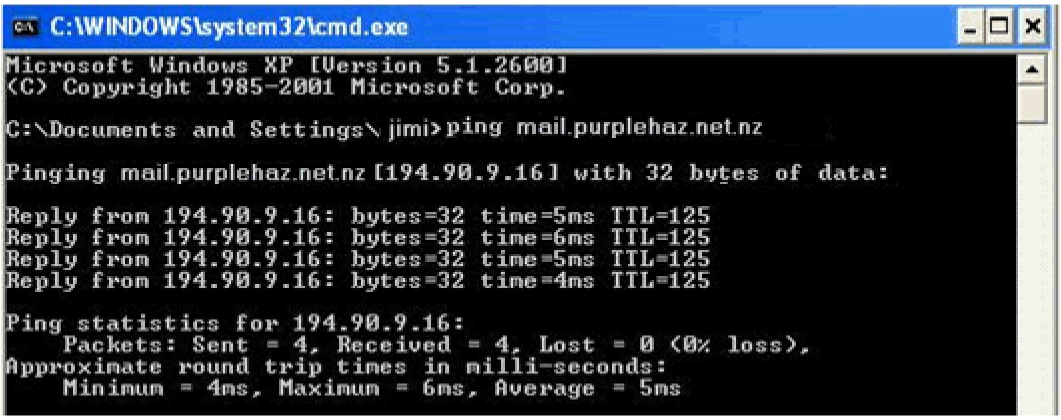
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ jimi>telnet mail.purplehaz.net.nz 25_
220 mxout1.mail.purplehaz.net.nz- Server ESMTP (MSG)
```

#### Determining the IP

Use the 'ping' command to determine the IP address of your mail server.

1. Click Windows Start > Run, then enter CMD.

2. In the CMD window, enter the command: ping RemoteMailServer, where RemoteMailServer is the name of your mail server; the command returns the IP address of  the server.



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ jimi>ping mail.purplehaz.net.nz

Pinging mail.purplehaz.net.nz [194.90.9.16] with 32 bytes of data:

Reply from 194.90.9.16: bytes=32 time=5ms TTL=125
Reply from 194.90.9.16: bytes=32 time=6ms TTL=125
Reply from 194.90.9.16: bytes=32 time=5ms TTL=125
Reply from 194.90.9.16: bytes=32 time=4ms TTL=125

Ping statistics for 194.90.9.16:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 6ms, Average = 5ms
```

### Intranet (LAN) mail

If your controller is sending e-mail to addresses within a local network, and you know the name of the server in your network, communicate with it directly.

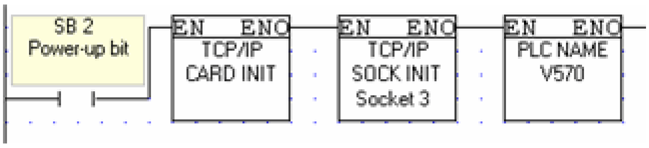
### e-mail via GPRS

Note that you can also send e-mail via GPRS modem. When you prepare the modem via Modem Services, set it to a baud rate of 9600. In addition, the COM Init function should be set to 9600.

### How to Configure and Send Messages

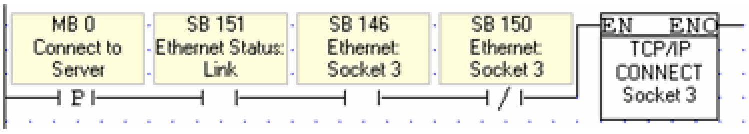
1. Configure Ethernet in your application by building a net that comprises the following elements:

a. Com>TCP/IP> Card Init function.
b. Com>TCP/IP>Socket Init function, set to **TCP Master**.
c. A Com>Set PLC Name function.



**Note •** Dedicate one socket to sending e-mail.

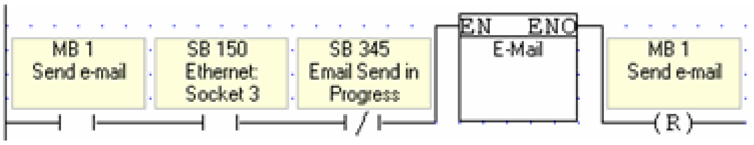
2. Establish the Ethernet Connection using the Com>TCP/IP> TCP/IP Connect Socket function. The function must be configured to your mail server IP, and to Port 25, which is the outgoing messages port.



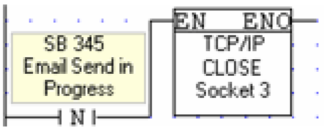
**Note •** It is recommended that you include a time elapse of a few seconds after the Ethernet Card initializes and before activating Socket Connect. A timer may be used for this purpose.

3. Send the e-mail using the Com>TCP/IP>Send e-mail function.

- Use the appropriate SB to ensure that the socket is connected before sending the e-mail.
- Use the status of SB 345 Email Send in Progress to avoid communication conflicts and ensure that the e-mail function is free before sending the e-mail. This bit turns ON when the e-mail function is activated, and turns OFF when the message has been sent to the server.
  Note that:
  - only one e-mail can be sent at a time, and that an application should use the same socket to send all e-mails.
  - you can send a number of e-mails before closing the connection.

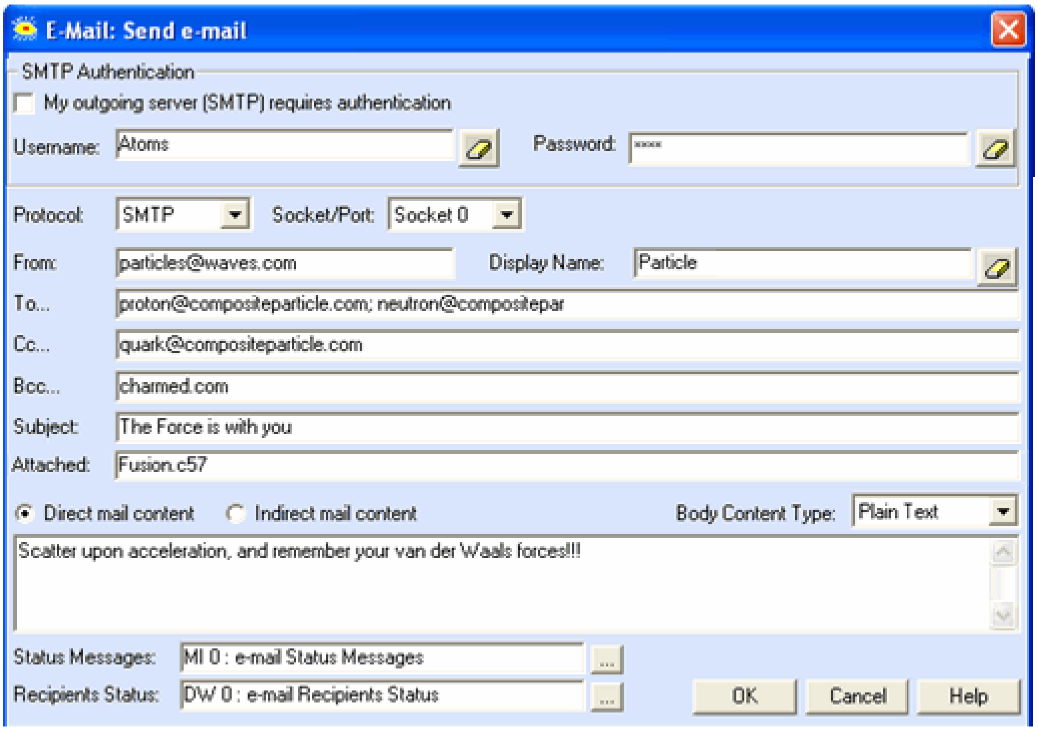


4. Close the connection and free the socket using the Com>TCP/IP>TCP/IP Close Socket function.
   After closing the socket, wait 30 seconds before sending additional e-mails.



**Configuring the Send e-mail Function**

Fill in the parameters according to the parameter table given below



SMTP Authentication

□ My outgoing server (SMTP) requires authentication

Username: Atoms Password: ****

Protocol: SMTP Socket/Port: Socket 0

From: particles@waves.com Display Name: Particle

To... proton@compositeparticle.com; neutron@compositepar

Cc... quark@compositeparticle.com

Bcc... charmed.com

Subject: The Force is with you

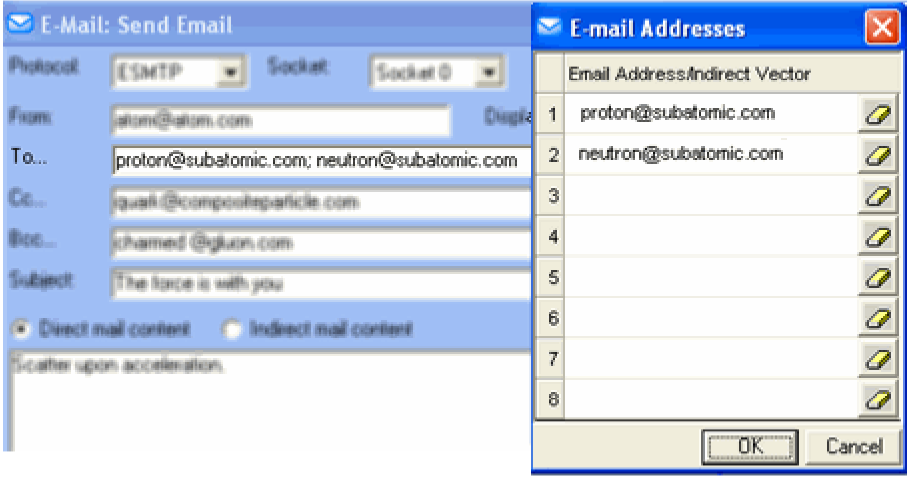
Attached: Fusion.c57

● Direct mail content ○ Indirect mail content Body Content Type: Plain Text

Scatter upon acceleration, and remember your van der Waals forces!!!

Status Messages: MI 0 : e-mail Status Messages

Recipients Status: DW 0 : e-mail Recipients Status

| **Parameter Name** | **Purpose** |
|---|---|
| Protocol | Select the protocol your server uses.<br>Both SMTP and ESMTP are supported. |
| Socket | Select the socket you are using for this function. |
| From | You can either type in an address, or link a vector of registers |
| To/Cc/Bcc | You can select up to 8 recipients per category<br>Click a line to open the address book.<br><br> <br>Protocol: ESMTP Socket: Socket 0<br>From: atom@atom.com<br>To...: proton@subatomic.com; neutron@subatomic.com<br>Cc...: quark@compositeparticle.com<br>Bcc...: charmed@gluon.com<br>Subject: The force is with you<br>● Direct mail content ○ Indirect mail content<br>Scatter upon acceleration<br> <br>Email Address/Indirect Vector<br>1 proton@subatomic.com<br>2 neutron@subatomic.com<br>3<br>4<br>5<br>6<br>7<br>8<br> |
| Subject | Click to enter text or link a vector. The vector will end either at a null character, or when reaching the maximum of 50 characters |
| Attached | Enhanced Vision controllers with SD cards enable you to attach up to 8 files from the SD card.<br>Click to set the attachment names.<br>Note that:<br>- The file name must adhere to the 8.3 file format, up to 8 characters for the name, 3 for the extension.<br>- File name may be provided by constant text or register. Note that if the name comes from an MI, the function copies a vector 8 bytes long, or until it finds a 'null' character<br>- The size of the attachments is limited in SI 58. maximum attachment size is 10 MB.<br>- Power-up default is 1<br>- Maximum per attachment = 10 (10 MB)<br>Note that the file size must not be changed while the Send is in Progress. |
| Direct mail content | Select to type your e-mail message into the content field. You can enter up to 800 characters. |
| Indirect mail content | Select to link to a vector. The vector will end either at a null character, or when reaching the maximum of 800 characters |
| The following operands should be assigned Power-up Values; bits should be reset, and registers initialized to 0. | |
| Status | When the In Progress bit turns ON, the current status message turns from 0 to 1. |

| | |
|---|---|
| Messages | When the<br>Status Message<br>0 – IDLE<br>1 – IN PROGRESS<br>2 - SUCCESS<br>3 – SMTP server does not respond ( HELO command Failed)<br>4 – ESMTP server does not respond ( EHLO command Failed)<br>5 – 'From' address format incorrect<br>6 – 'To/Cc/Bcc' address format incorrect. Check Address Status Operand<br>7 – The server cannot receive data (DATA mode error)<br>8 – Mail was not sent: error occurred while sending e-mail header (name, subject, etc..)<br>9 - Mail was not sent: error occurred while sending e-mail data (text, attachments, etc..)<br>10 - Mail was not sent: error occurred while sending e-mail end of data/closing the e-mail<br>11 – Mail was not sent: error occurred since authentication is not enabled or it's not supported<br>12 – Mail was not sent: error occurred while sending username with authentication mode<br>13 – Mail was not sent: error occurred while sending password with authentication mode<br>14 – Mail was not sent: error occurred while sending username with plain login mode<br>15 – Mail was not sent: error occurred while sending password with plain login mode |
| Recipients Status | If the Status Message is 6, use this register to determine which addresses are in incorrect format.<br>• Bits 0-7: 'To' address X is illegal<br>• Bits 8-15: 'Cc' address X is illegal<br>• Bits 16-23: 'Bcc' address X is illegal<br>The bit corresponds with the incorrect address. If, for example, bit 9 is ON, the address on the second Cc line is incorrect. |

**Note •** The function does not support authentication, SSL or TSL encryption.

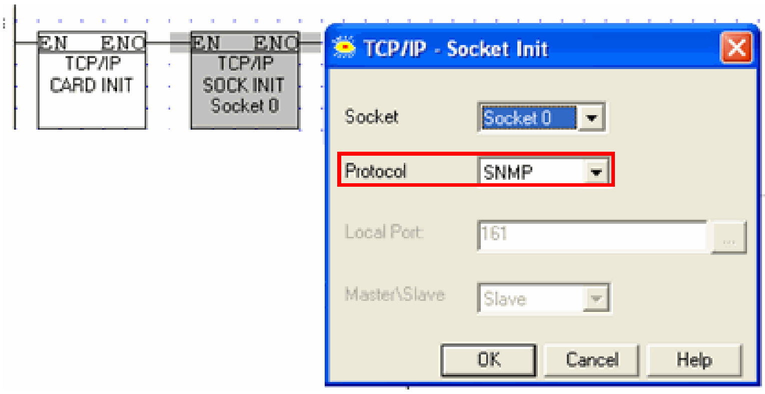
## SNMP

An Enhanced Vision that is Ethernet-enabled can function as an:

- SNMP agent (slave)
- SNMP Trap

### SNMP Agent (slave)

To enable the PLC to act as an SNMP agent, initialize a socket to SNMP as shown below.



Socket: Socket 0

Protocol: SNMP

Local Port: 161

Master\Slave: Slave

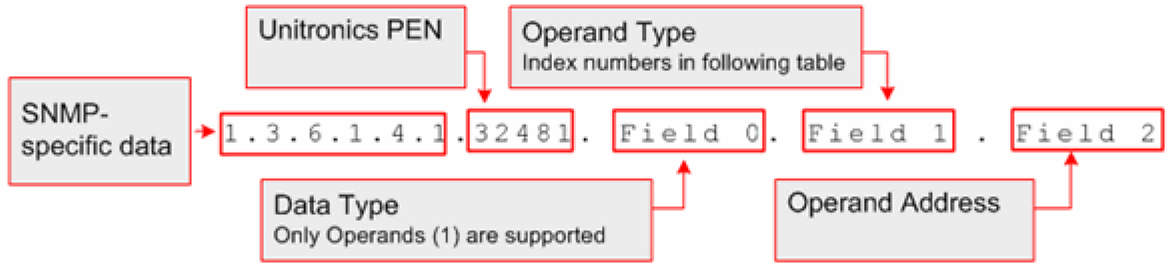
This enables any SNMP manager to access operand values in the Vision via UDP, port 161. Vision supports the following SNMPv1 commands, sent to the PLC from the network:

1. Get
2. Get Next
3. Set

The commands support 32-bit integers.

Note that the unique PEN number assigned to Unitronics is 32481.

The figure below shows the SNMP command format.


Unitronics PEN
Operand Type
Index numbers in following table
SNMP-specific data
1.3.6.1.4.1.32481.Field 0.Field 1 . Field 2
Data Type
Only Operands (1) are supported
Operand Address

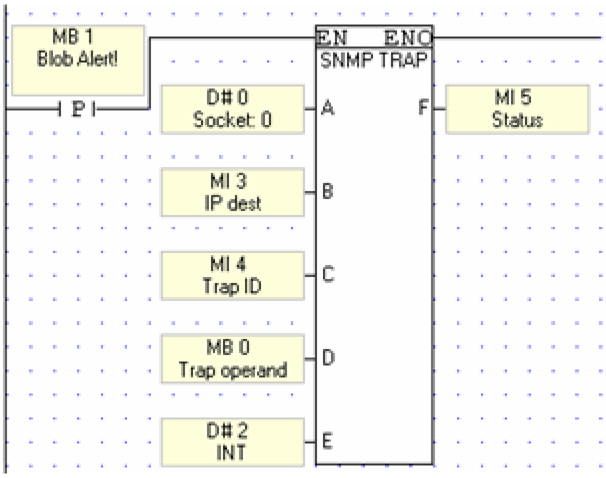
The table below shows the index numbers for operand types in Unitronics' controllers.

| | | Number of Operands | | |
|---|---|---|---|---|
| **Name** | **Operand type index (hex)** | **V570** | **V350** | **130** |
| MB | 0x01 | 8192 | 8192 | 4096 |
| SB | 0x02 | 1024 | 1024 | 512 |
| MI | 0x03 | 4096 | 4096 | 2048 |
| SI | 0x04 | 1024 | 1024 | 512 |
| ML | 0x05 | 512 | 512 | 256 |
| SL | 0x06 | 64 | 64 | 56 |
| MF | 0x07 | 64 | 64 | 24 |
| MDW | 0x10 | 256 | 256 | 64 |
| SDW | 0x24 | 64 | 64 | 64 |
| XB | 0x40 | 1024 | 1024 | 1024 |
| XI | 0x41 | 512 | 512 | 512 |
| XL | 0x42 | 256 | 256 | 256 |
| XDW | 0x43 | 64 | 64 | 64 |

SNMP Trap

In addition, the Vision can send Trap messages in response to Ladder conditions, via the function COM>TCP/IP>SNMP Trap.

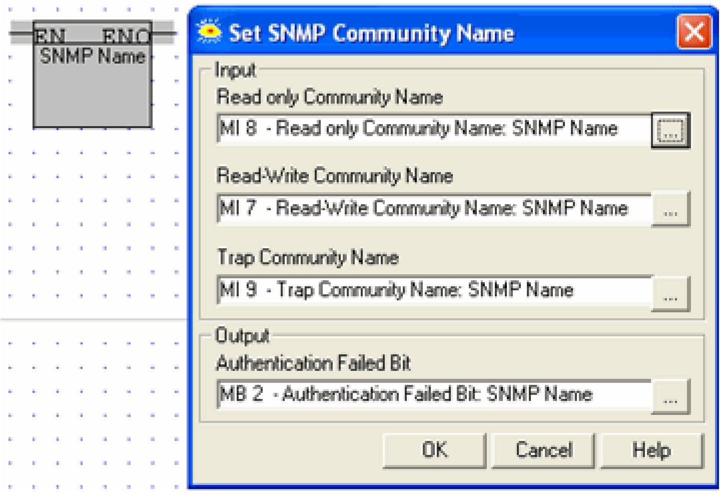
The PLC supports integer values, strings, and null.



| Parameter | Purpose |
|---|---|
| Socket | Select any socket |
| IP: Target device | The operand value will be sent to this IP |
| Trap ID | Use this to assign an SNMP number to the device |
| Trap Operand | This is the operand that is sent to the target device. If the data sent is a string, the Trap operand is the start of a vector; the PLC will send data from the vector until it reaches a null. |
| Trap Format | Select integer, string, or null |
| SNMP Trap Status: | • 0: Valid trap.<br>• 1: Invalid port (the Rx socket is not Port 161)<br>• 2: The destination IP is in an invalid format |

SNMP Community Name

Use this function to assign the controller a name, enabling it to be accessed via SNMP.



Set SNMP Community Name

Input

Read only Community Name: MI 8 - Read only Community Name: SNMP Name

Read-Write Community Name: MI 7 - Read-Write Community Name: SNMP Name

Trap Community Name: MI 9 - Trap Community Name: SNMP Name

Output

Authentication Failed Bit: MB 2 - Authentication Failed Bit: SNMP Name

-

### Ethernet TCP\IP: PC to Vision

In order to use a PC to access a Vision controller via Ethernet:

1. The Vision PLC must contain an Ethernet port. Ethernet ports for V2xx Vision OPLCs are available by special order.
2. Both the PC and PLC must be connected to an Ethernet network, and be assigned valid IP addresses; the PLC must be assigned a unique name via the Set PLC Name.
3. The PLC must be defined in either TCP\IP's Ethernet Project Settings or in TCP\IP Favorites.
4. In VisiLogic's Vision Communication - PC Settings:
   - Ethernet must be selected
   - the target PLC must be selected from either Favorites or TCP\IP Project Settings.

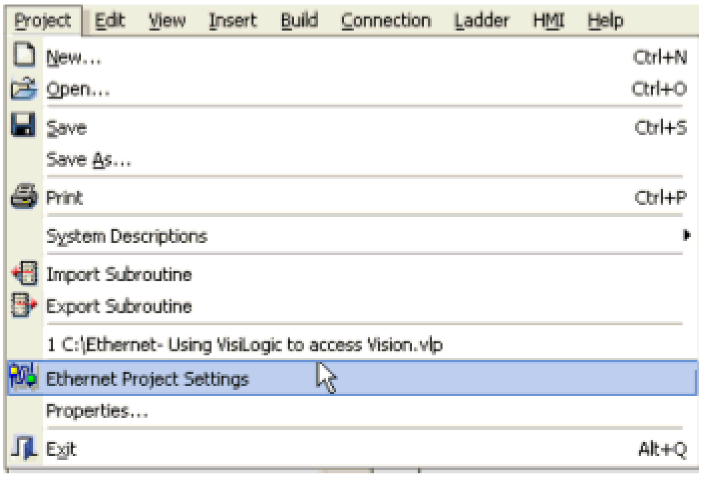
These conditions enable VisiLogic to access a PLC via Ethernet in order to download programs and carry out other tasks.

### TCP/IP Project Settings

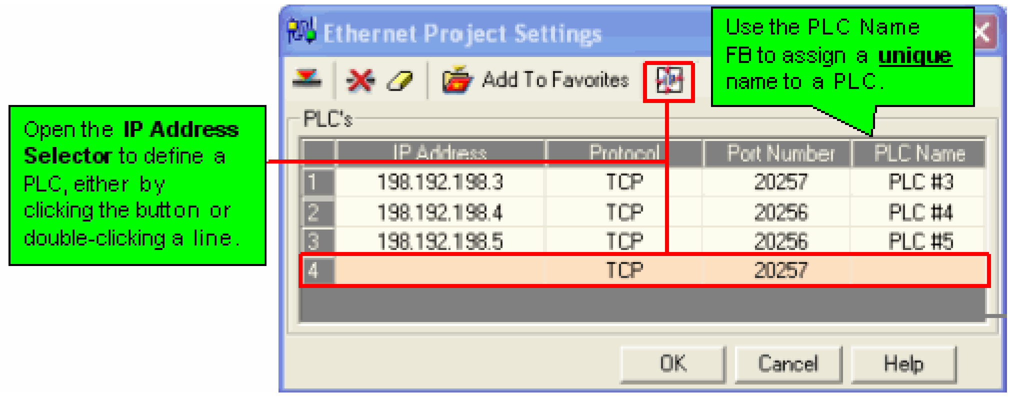
TCP/IP Project Settings enable VisiLogic to access a Vision PLC via an Ethernet connection. Each PLC included in the project will be accessed according to the protocol and port number assigned to it.

TCP/IP Project Settings contain IP addresses and settings that are specific to a particular VisiLogic project. However, you can add any of the IP addresses it contains to the Favorites file, which is a common, global file that can be accessed from any VisiLogic project.

1. Select TCP/IP Settings from the Project menu.



2. Open the IP Address Selector in order to enter the IP of the PLC s in the project.

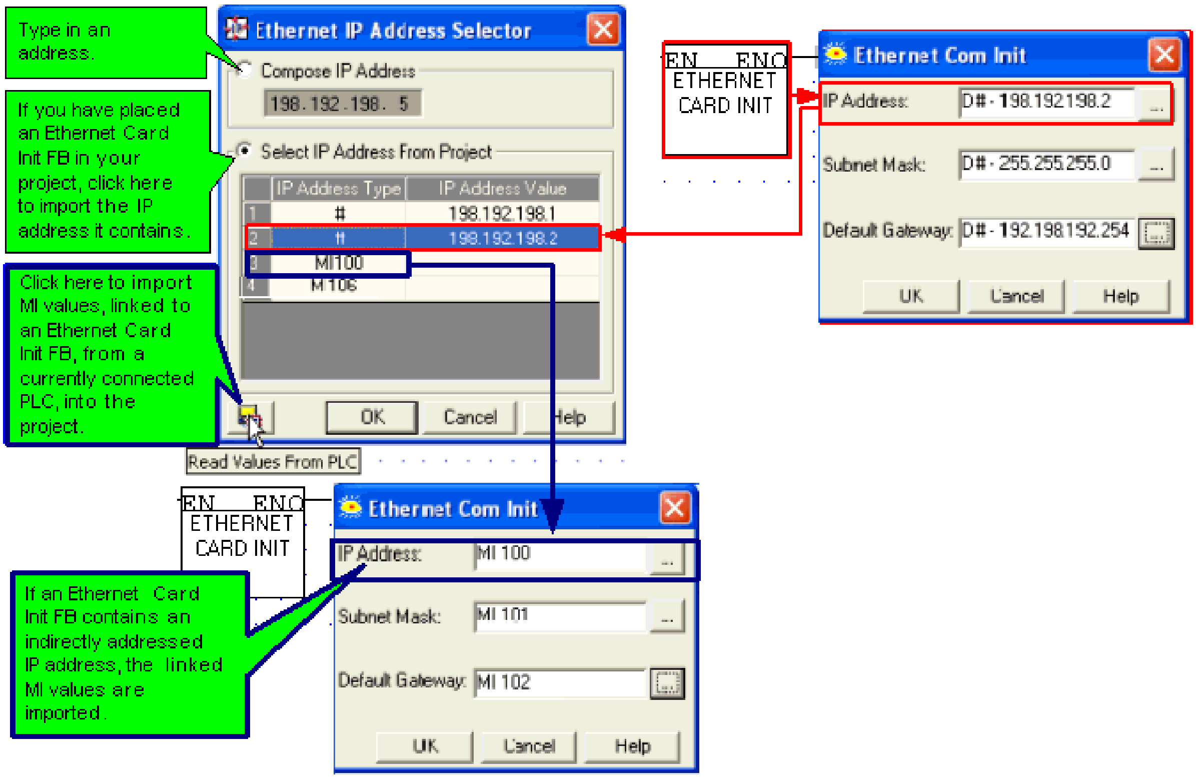


Open the **IP Address Selector** to define a PLC, either by clicking the button or double-clicking a line.

Use the PLC Name FB to assign a **unique** name to a PLC.

| | IP Address | Protocol | Port Number | PLC Name |
|---|---|---|---|---|
| 1 | 198.192.198.3 | TCP | 20257 | PLC #3 |
| 2 | 198.192.198.4 | TCP | 20256 | PLC #4 |
| 3 | 198.192.198.5 | TCP | 20256 | PLC #5 |
| 4 | | TCP | 20257 | |

3. Enter an IP address either by:
   - Typing it,
   - Importing any IP addresses you may have defined in your project within TCP/IP Init FBs
   - Importing the IP of a PLC currently linked to your PC via a valid communication connection.



Type in an address.

If you have placed an Ethernet Card Init FB in your project, click here to import the IP address it contains.

Click here to import MI values, linked to an Ethernet Card Init FB, from a currently connected PLC, into the project.

If an Ethernet Card Init FB contains an indirectly addressed IP address, the linked MI values are imported.

Ethernet IP Address Selector

○ Compose IP Address: 198. 192. 198. 5

● Select IP Address From Project

|  | IP Address Type | IP Address Value |
|---|---|---|
| 1 | # | 198.192.198.1 |
| 2 | # | 198.192.198.2 |
| 3 | MI100 |  |
| 4 | MI 106 |  |

Ethernet Com Init

IP Address: D# - 198.192.198.2

Subnet Mask: D# - 255.255.255.0

Default Gateway: D# - 192.198.192.254

Ethernet Com Init

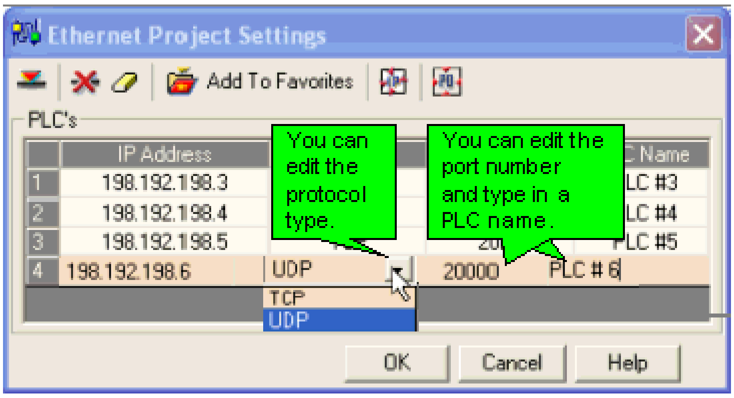
IP Address: MI 100

Subnet Mask: MI 101

Default Gateway: MI 102

**Note •** If you import a Vision's IP, its protocol type (UDP or TCP) and port number are also imported.
If you have typed the IP address, you must also type the port number.

4. Before exiting TCP/IP Project Settings, you must set the PLC name. This is a unique name that is assigned to the controller via the Set PLC Name FB.



Ethernet Project Settings

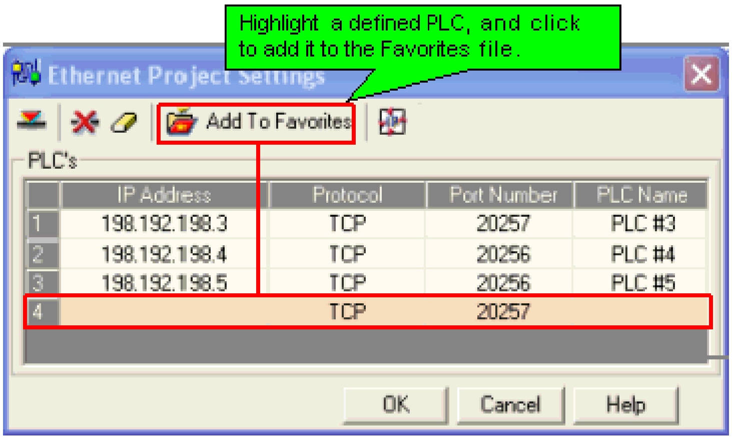
You can edit the protocol type.

You can edit the port number and type in a PLC name.

PLC's

|  | IP Address |  |  | Name |
|---|---|---|---|---|
| 1 | 198.192.198.3 |  |  | PLC #3 |
| 2 | 198.192.198.4 |  |  | PLC #4 |
| 3 | 198.192.198.5 |  | 20 | PLC #5 |
| 4 | 198.192.198.6 | UDP | 20000 | PLC # 6 |

**Note •** You can add any PLC defined in TCP/IP Project Settings to the Ethernet Favorites file.


Highlight a defined PLC, and click to add it to the Favorites file.
Ethernet Project Settings
Add To Favorites
PLC's
OK
Cancel
Help

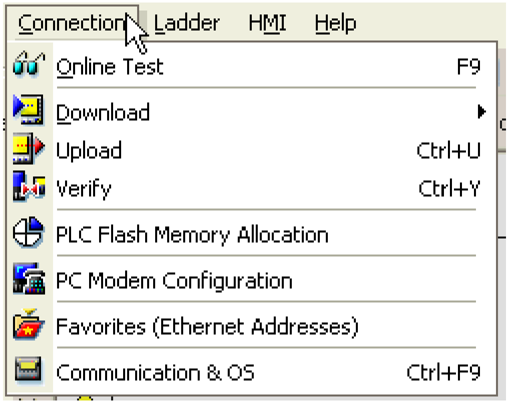
| | IP Address | Protocol | Port Number | PLC Name |
|---|---|---|---|---|
| 1 | 198.192.198.3 | TCP | 20257 | PLC #3 |
| 2 | 198.192.198.4 | TCP | 20256 | PLC #4 |
| 3 | 198.192.198.5 | TCP | 20256 | PLC #5 |
| 4 | | TCP | 20257 | |

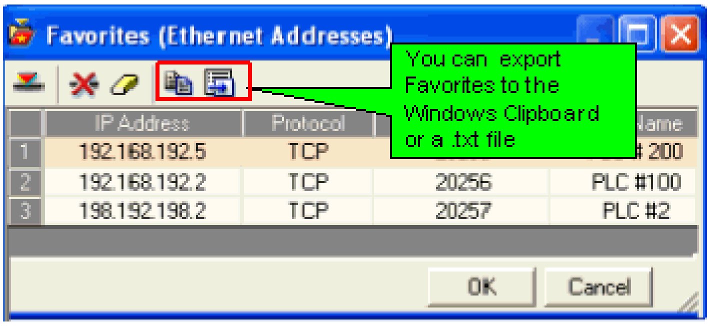
## TCP/IP Favorites

Favorites is a common, global file that can be accessed from any VisiLogic project, as well as by other programs such as Remote Access.

Favorites is a file that is created by VisiLogic after you have added a PLC to the favorites list. It is a global file stored in the Program\Common files\Unitronics folder as Eth_Favorites.evb.

Ethernet Favorites is located on the Connection menu.




Favorites (Ethernet Addresses)
You can export Favorites to the Windows Clipboard or a .txt file
OK
Cancel

| | IP Address | Protocol | | Name |
|---|---|---|---|---|
| 1 | 192.168.192.5 | TCP | | # 200 |
| 2 | 192.168.192.2 | TCP | 20256 | PLC #100 |
| 3 | 198.192.198.2 | TCP | 20257 | PLC #2 |

Once Favorites have been created, you can access them in order to:

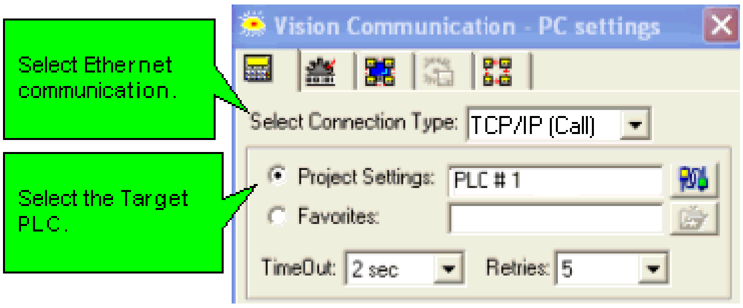
- Import the definitions into TCP/IP Project Settings.
- Select a PLC listed in Favorites to enable VisiLogic to access a PLC via Vision Communication - PC Settings.

## Vision Communication - PC Settings--Selecting the target PLC

Once the conditions above have been met, VisiLogic can access a selected PLC via Ethernet.

1. Select Communication and OS from the Connection menu.

2. Under Connection Type, select TCP/IP (Call), and then select the target PLC from either Project Settings or Favorites.



Vision Communication - PC settings

Select Connection Type: TCP/IP (Call)

○ Project Settings: PLC # 1

○ Favorites:

TimeOut: 2 sec Retries: 5

VisiLogic is now ready to communicate via Ethernet.

### UDP Raw Send/Receive

These functions are located on the COM>TCP/IP>UDP menu. Use them to send a vector of raw data to a remote Vision, broadcast the data to the entire network, or receive a stream of data and write it to a vector of registers.

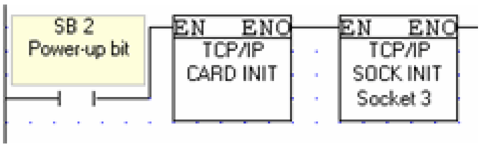
The number of bytes a message can comprise varies according to Vision model:

- Standard: 576
- Enhanced: 1024

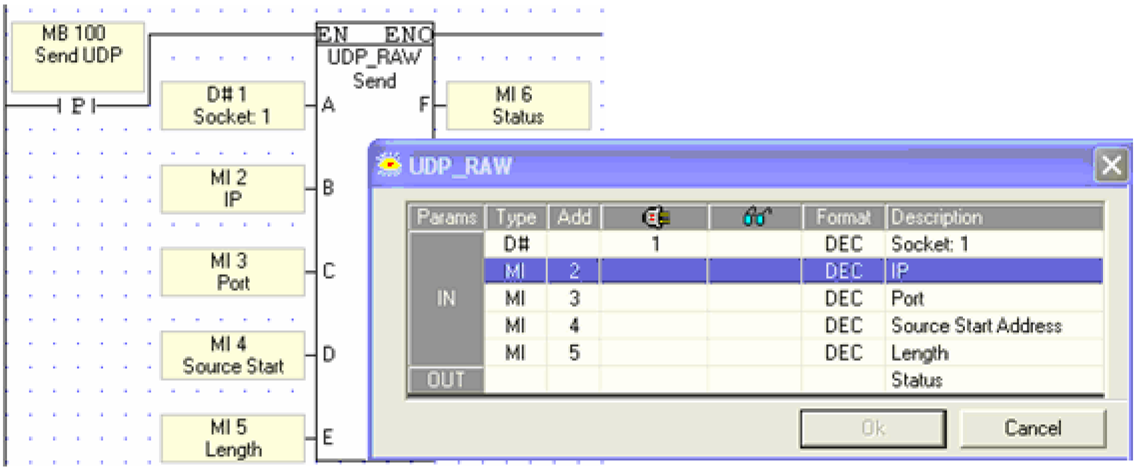
**Send**

1. Configure Ethernet in your application by building a net that comprises the following elements:

Com>TCP/IP> Card Init function. Com>TCP/IP>Socket Init function, set to UDP_RAW. You can use any socket **except** Socket 0.



2. Configure a UDP RAW Send function to determine the data to send to a specific PLC.



| Params | Type | Add |  |  | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# |  | 1 |  | DEC | Socket: 1 |
|  | MI | 2 |  |  | DEC | IP |
|  | MI | 3 |  |  | DEC | Port |
|  | MI | 4 |  |  | DEC | Source Start Address |
|  | MI | 5 |  |  | DEC | Length |
| OUT |  |  |  |  |  | Status |

**Note •** Use a Send condition; do not place it directly on the left Ladder rail.

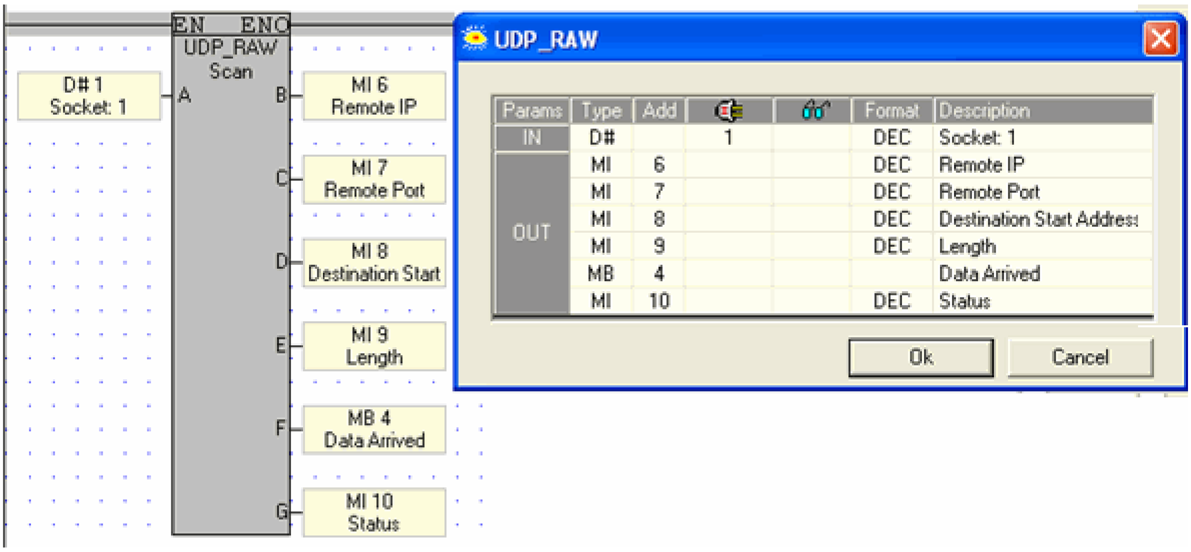
| Parameter Name | Purpose |
|---|---|
| Sending Socket | The PLC sends the data through this socket. Select a socket that is initialized to UPD_RAW |
| IP of remote PLC | The IP of the remote device<br>To broadcast the data to the network, send to IP255.255.255.255 |
| Sending Port | The PLC sends the data through this port |
| Source Start Address | Start address of the vector containing the data |
| Vector Length | Determines the vector length |
| Status | Shows the status of the message that is sent<br>Value Message<br>0 Data Sent Successfully<br>-2 Illegal IP address<br>-3 Data vector exceeds the range for that data type |

## Scan

The Scan function enables a PLC to receive data from a specified remote PLC and write it to a vector of registers.

1. Configure the application in the PLC that receives the data by building a net that comprises a Card Init function and a Socket Init function, set to UDP_RAW.
2. Configure a UDP RAW Scan function to receive the data.

**Note •** In most applications, Scan should be active all the time. To accomplish this, place the Scan function in the main routine, directly on the left Ladder rail.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 1 | | DEC | Socket: 1 |
| OUT | MI | 6 | | | DEC | Remote IP |
| | MI | 7 | | | DEC | Remote Port |
| | MI | 8 | | | DEC | Destination Start Addres |
| | MI | 9 | | | DEC | Length |
| | MB | 4 | | | | Data Arrived |
| | MI | 10 | | | DEC | Status |

| **Parameter Name** | **Purpose** |
|---|---|
| Socket | This is the socket that receives the data. This socket must be initialized to UPD_RAW |
| IP of remote PLC | The IP of the device that sent the data<br>Note that this will user a vector of 4 MI |
| Remote Port | The port of the device that sent the data |
| Destination Start Address | Start address of the vector to which the received data is written |
| Vector Length | The number of bytes received. Is reset by OS when the element activates. |
| Data Arrived | Use this to monitor message status. Is reset by OS when the element activates. |
| Status | Value Message<br>-3 Data vector exceeds the range for that data type |

### TCP Raw Send/Receive

These functions are located on the COM>TCP/IP>TCP menu. Use them to send a vector of raw data to a remote Vision, broadcast the data to the entire network, or receive a stream of data and write it to a vector of registers.

The number of bytes a message can comprise varies according to Vision model:

- Standard: 576
- Enhanced: 1024

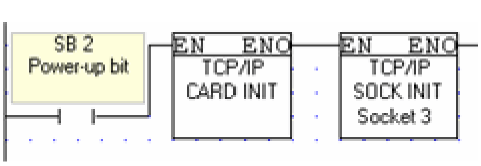
TCP Raw is a master/slave protocol. In addition to other TCP/IP elements, the master application contains a TCP Connect function.

**Send**

1. Configure Ethernet in your application by building a net that comprises the following elements:

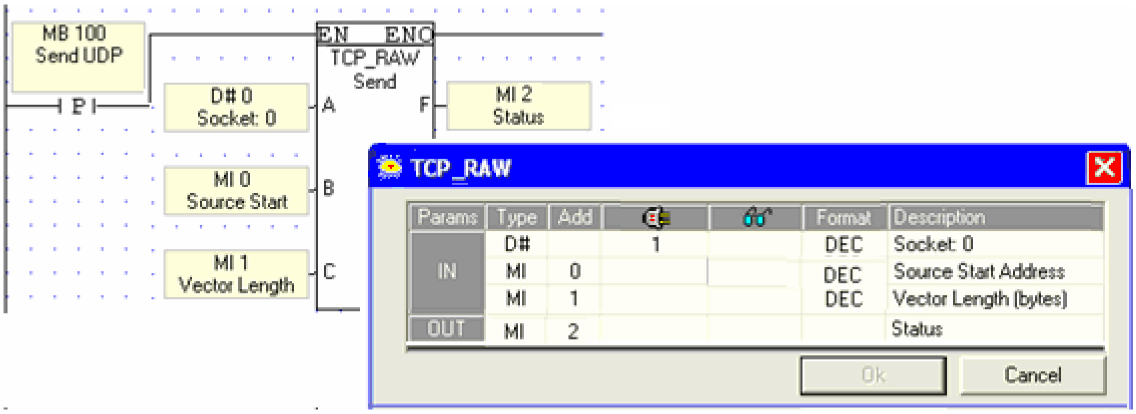
a. Com>TCP/IP> Card Init function.

b. Com>TCP/IP>Socket Init function, set to TCP_RAW.



2. Include a TCP\IP: TCP Connect function.

3. Configure a TCP RAW Send function to determine the data to send to a specific PLC.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 1 | | DEC | Socket: 0 |
| | MI | 0 | | | DEC | Source Start Address |
| | MI | 1 | | | DEC | Vector Length (bytes) |
| OUT | MI | 2 | | | | Status |

**Note** • Use a Send condition; do not place it directly on the left Ladder rail.

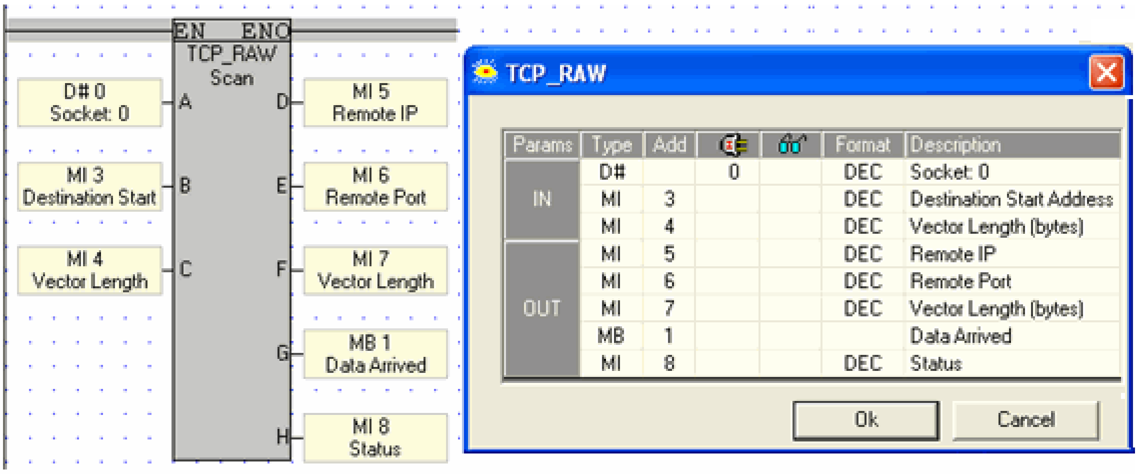
| **Parameter Name** | **Purpose** |
|---|---|
| Sending Socket | The PLC sends the data through this socket. Select a socket that is initialized to UPD_RAW |
| Source Start Address | Start address of the vector containing the data |
| Vector Length | Determines the vector length |
| Status | Shows the status of the message that is sent<br>Value Message<br>1 Data Sent Successfully<br>-1 Illegal Socket (not 0-3)<br>-2 Data length is longer than Ethernet buffer<br>-3 Illegal operand<br>-4 Socket not initialized<br>-5 Protocol not TCP<br>-6 Send failed |

### Scan

The Scan function enables a PLC to receive data from a specified remote PLC and write it to a vector of registers.

1. Configure the application in the PLC that receives the data by building a net that comprises a Card Init function and a Socket Init function, set to UDP_RAW.
2. Configure a UDP RAW Scan function to receive the data.

**Note •** In most applications, Scan should be active all the time. To accomplish this, place the Scan function in the main routine, directly on the left Ladder rail.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 0 | | DEC | Socket: 0 |
| | MI | 3 | | | DEC | Destination Start Address |
| | MI | 4 | | | DEC | Vector Length (bytes) |
| OUT | MI | 5 | | | DEC | Remote IP |
| | MI | 6 | | | DEC | Remote Port |
| | MI | 7 | | | DEC | Vector Length (bytes) |
| | MB | 1 | | | | Data Arrived |
| | MI | 8 | | | DEC | Status |

| Parameter Name | Purpose |
|---|---|
| Socket | This is the socket that receives the data. This socket must be initialized to TCP_RAW |
| Destination Start Address | Start address of the vector to which the received data is written |
| Vector Length | The number of bytes received. Is reset by OS when the element activates. |
| Data Arrived | Use this to monitor message status |
| Status | Shows the status of the message that is received:<br>Value Message<br>o Data Received Successfully<br>-1 Card not initialized, or link fail<br>-2 Illegal Socket<br>-3 Socket not initialized<br>-4 Protocol not TCP<br>-5 Illegal operand<br>-6 Data length is longer than Ethernet buffer d |

### Ethernet TCP\IP: SBs & SIs

| # | Description | Turns ON when: | Turns OFF when: | Comments |
|---|---|---|---|---|
| SB 141 | Ethernet: Card Exists | Ethernet card is found | No Ethernet card is installed | When the Ethernet: Card Initialization FB runs, the PLC checks whether an Ethernet card is installed. |
| SB 142 | Ethernet: Card Initialized | Ethernet card initialization succeeds | Ethernet card initialization fails | |
| SB 143 | Ethernet: Socket 0 Initialized | Socket 0 initialization succeeds | Socket 0 initialization fails | |
| SB 144 | Ethernet: Socket 1 Initialized | Socket 1 initialization succeeds | Socket 1 initialization fails | |
| SB 145 | Ethernet: Socket 2 Initialized | Socket 2 initialization succeeds | Socket 2 initialization fails | |
| SB 146 | Ethernet: Socket 3 Initialized | Socket 3 initialization succeeds | Socket 3 initialization fails | |
| SB 147 | Ethernet: Socket 0 Connected | Connection established via Socket 0 | Socket 0 is free | SBs 147-150 turn ON when:<br>• Link exists<br>• Ethernet Card initialization complete<br>• Socket initialization complete<br>• Hardware TCP/IP Socket state is Connection Established |
| SB 148 | Ethernet: Socket 1 Connected | Connection established via Socket 1 | Socket 1 is free | |
| SB 149 | Ethernet: Socket 2 Connected | Connection established via Socket 2 | Socket 2 is free | |
| SB 150 | Ethernet Status: Socket 3 Connected | Connection established via Socket 3 | Socket 3 is free | |
| SB 151 | Ethernet Link: Communication established | A link exists (cable plugged in) | No link exists (cable disconnected) | This refers to the physical Ethernet cable |
| SB 152 | Ethernet Link: 10baseT | When a 10baseT link is detected, during data transmit/ receive. | When a 10baseT link is not detected, during data transmit/ receive. | |
| SB 153 | Ethernet Link: 100baseT | When a 100baseT link is detected, during data transmit/ receive. | When a 100baseT link is not detected, during data transmit/ receive. | |
| SB 154 | Ethernet: data collision | More than one device is transmitting data over the Ethernet network | One or no devices are transmitting data over the Ethernet network | |
| SB 155 | Ethernet: Socket 0 Send in Progress | Data is being transmitted via Socket 0 | Data is not being transmitted via Socket 0 | |
| SB 156 | Ethernet: Socket 1 Send in Progress | Data is being transmitted via Socket 1 | Data is not being transmitted via Socket 1 | |
| SB 157 | Ethernet: Socket 2 Send in | Data is being transmitted via | Data is not being transmitted via | |

| | | | | |
|---|---|---|---|---|
| | Progress | Socket 2 | Socket 2 | |
| SB 158 | Ethernet: Socket 3 Send in Progress | Data is being transmitted via Socket 3 | Data is not being transmitted via Socket 3 | |
| SB 159 | Enable Unicast, Socket 0 | Turn ON (ON by default) to enable Socket 0 for Unicast | Turn OFF to disable Socket 0 for Unicast | |
| SB 162 | Ethernet Reconnect parameters saved | | | |
| SB 163 | Connection is Closed (Socket 0) | By OS, when connection is closed | | SB turns ON when Close Connection is performed. This is after Transmit / Receive buffers are empty or 1-second timeout has passed. Socket is initialized. |
| SB 164 | Connection Closed (Socket 1) | By OS, when connection is closed | | |
| SB 165 | Connection Closed (Socket 2) | By OS, when connection is closed | | |
| SB 166 | Connection Closed (Socket 3) | By OS, when connection is closed | | |
| SB 167 | Ethernet Critical error | Turns ON at critical error | | OS reads Ethernet card registers to SI 300 - 427, then re initializes the card. User must reset PLC |
| SB 168 | Enable "Link lost" auto recover | Automatically retry link | Do not automatically retry (default) | SB 168 should be turned ON at power-up.<br>It is OFF by default to preserve backwards compatibility with applications created previous to OS 4.70 B14.<br>If SB 168 is ON, when the Ethernet link fails, the OS saves the Ethernet parameters and resets:<br>• SB 142 Card Initialized<br>• SBs 143-146 Socket initialized (Sockets 0-3)<br>• SBs 147-150 Socket connected (Sockets 0-3)<br>When the Ethernet link is reestablished, the O/S performs CARD INIT and SOCKET INIT for all 4 sockets according to the saved parameters. |
| SB 169 | Automatic reconnect requested, in progress (Socket 0) | At Ladder | | TCP - Used as internal flags by O/S in order to perform auto reconnect (user parameters SI 107 – 110) |
| SB 170 | Automatic reconnect requested, in progress (Socket 1) | | | |
| SB 171 | Automatic reconnect requested, in progress (Socket | | | |

| | | | |
|---|---|---|---|
| | 2) | | |
| SB 172 | Automatic reconnect requested, in progress (Socket 3) | | |
| SB 173 | Automatic reconnect requested (Socket 0) | | |
| SB 174 | Automatic reconnect requested (Socket 1) | | |
| SB 175 | Automatic reconnect requested (Socket 2) | | |
| SB 176 | Automatic reconnect requested (Socket 3) | | |

| # | Description | Value | Comments |
|---|---|---|---|
| SI 101 | TCP/IP retries base time out | Legal values are 1 to 10, units of 100 mSec (1 stands for 100 mSec etc.)<br>Default value is 200 mSec. | Same value is for ALL 4 sockets<br>Requires CARD INIT<br>Illegal value request will be rejected (no change) |
| SI 102 | Retries count | Legal values are from 1 to 50<br>Default value is 6 | Illegal value request will be rejected (no change) |
| SI 103 | TCP/IP Connection Keep Alive (Socket 0) | Units of 100 mSec Note- When value is '0', the function is disabled | Enables the PLC to disconnect if there is no communication from the connected device.<br>When TCP/IP connection is established (SI 145-148 = 6) check data transport (SDW 14 – 21).<br>If no data transport occurred during the defined time – perform 'Socket Init'.<br>Keep Alive counter is SDW 38 |
| SI 104 | TCP/IP Connection Keep Alive (Socket 1) | | |
| SI 105 | TCP/IP Connection Keep Alive (Socket 2) | | |
| SI 106 | TCP/IP Connection Keep Alive (Socket 3) | | |
| SI 107 | TCP/IP Keep Master Connection (Socket 0) | Units of 100 mSec Note- When value is '0', the function is disabled | Enables the PLC to reconnect when there is no communication from the connected device for the defined time.<br>Note that the value per socket should be higher than the regular "keep alive" (SI 103 – 106)<br>Counter of the operation in SDW 45 |
| SI 108 | TCP/IP Keep Master Connection (Socket 1) | | |
| SI 109 | TCP/IP Keep Master Connection (Socket 2) | | |
| SI 110 | TCP/IP Keep Master Connection (Socket 3) | | |
| SI 140 | Ethernet Send has failed, per socket ( bitmap) | Bit is ON when Send is not successful | Bit Map:<br>\| UDP S3 \| UDP S2 \| UDP S1 \| UDP S0 \| TCP S3 \| TCP S2 \| TCP S1 \| TCP S0 \| |
| SI 141 | Ethernet Socket 0: Protocol Type | 0=PC application (default)<br>1=MODBUS | (Read-only)<br>Sockets are set to Protocol Type 0 by default. Activating MODBUS Configuration changes the Protocol Type to 1. |
| SI 142 | Ethernet Socket 1: Protocol Type | | |
| SI 143 | Ethernet Socket 2: Protocol | | |

| | | | |
|---|---|---|---|
| | Type | | |
| SI 144 | Ethernet Socket 3: Protocol Type | | |

| Parameter | Function | | SI Value | Message |
|---|---|---|---|---|
| S1 145 | Socket 0: Status | | 0 | Initialized to UDP, status: Closed |
| SI 146 | Socket 1: Status | | 2 | Initialized to TCP, status: Listen |
| SI 147 | Socket 2: Status | | 14 | Initialized to UDP, status: Ready |
| SI 148 | Socket 3: Status | | 15 | Initialized to UDP, status: Engaged in Transmit/Receive |

| Parameter | Description | Value | Comments |
|---|---|---|---|
| SDW 14 | Socket 0: Number of sent transmissions | Updated after each data transmission via Socket 0 | |
| SDW 15 | Socket 1: Number of sent transmissions | Updated after each data transmission via Socket 1 | |
| SDW 16 | Socket 2: Number of sent transmissions | Updated after each data transmission via Socket 2 | |
| SDW 17 | Socket 3: Number of sent transmissions | Updated after each data transmission via Socket 3 | |
| SDW 18 | Socket 0: Number of received transmissions | Updated after each data packet received via Socket 0 | |
| SDW 19 | Socket 1: Number of received transmissions | Updated after each data packet received via Socket 1 | |
| SDW 20 | Socket 2: Number of received transmissions | Updated after each data packet received via Socket 2 | |
| SDW 21 | Socket 3: Number of received transmissions | Updated after each data packet received via Socket 3 | |
| SDW 37 | MODBUS Slave: Receive bitmap | Increments a 4-bit field each time a slave receives data | High - >low:<br>\| Eth port 3\| Eth port 2\| Eth port 1\| Eth port 0\| \|spare\| \| com 3\| com 2\| com 1\| |
| SDW 38 | TCP/IP Keep Alive counter | Increments a 8-bit field each time the O/S initializes the socket due to 'Keep Alive' (SI 103-106) | \| Eth port 3\| Eth port 2\| Eth port 1\| Eth port 0\| |
| SDW 39 | Ethernet general critical error | 8-bit counters | Bits 24-31: "Ethernet card init" – if the MS Byte (xxx.yyy.zzz.kkk, MS means the xxx part) of the IP/ SUBNET/ GATEWAY is zero – do not init the Ethernet.<br>Bits 16-23: Check once in second if local IP SUB and GATEWAY are ok. Cause set of SB 167.<br>Bits 8-15: TCP (connect) & UDP (send) IP is defined – Read HW remote IP to verify. Case verifies failed: Ignore connect or send.<br>Bits 0-7: While getting message from |

| | | | |
|---|---|---|---|
| | | | socket – if the high part of the remote IP high is zero. Cause set of SB 167. |
| SDW 45 | TCP/IP Keep Master Connection | Increments a 8-bit field each time the O/S attempts to Keep Master Connection' (SI 107-110) | \| Eth port 3\| Eth port 2\| Eth port 1\| Eth port 0\| |

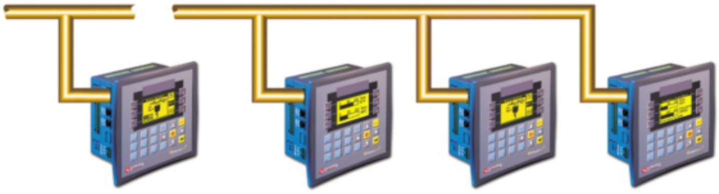
# Networks (CAN, Serial)

## About Networks

Vision controllers offer different networking options:

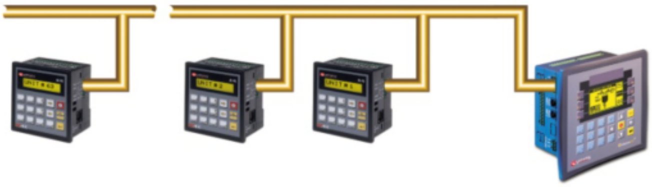
**CANbus**

You can create a decentralized control network of up to 63 controllers using CANbus. This is sometimes called a multi-master network. In such a network, CANbus enables inter-PLC data exchange. Technical specifications and wiring diagrams are given in the User Guide. Vision controllers also support data exchange via the MODBUS protocol



.

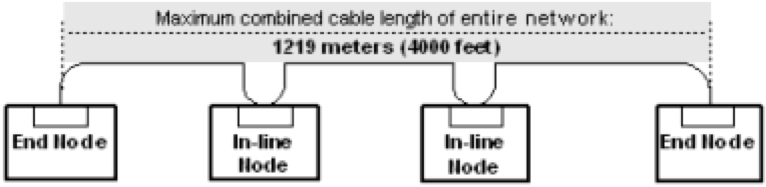
You can network M90 Micro-OPLCs and Vision OPLC controllers.



Via RS232 and RS485, you can implement communication protocols such as MODBUS.

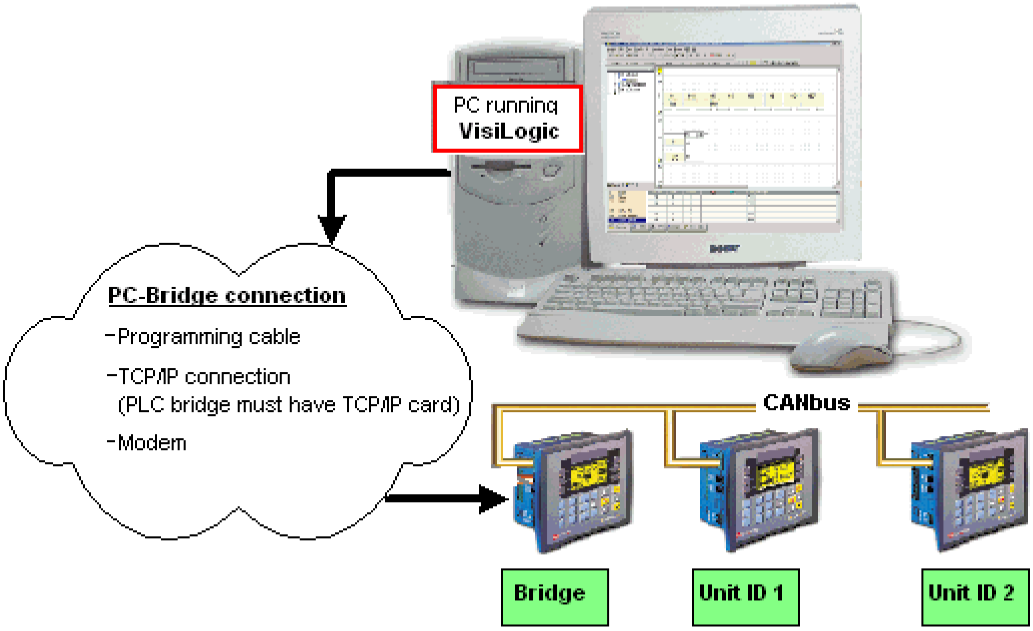
RS485

RS485 is a balanced serial interface for the transmission of digital data, which enables you to create a multi-drop network containing up to 32 devices, or nodes.



**Network Access via Serial COM Ports**

You can access a networked controller unit via its RS232/RS485 port using a PC, whether directly connected or via modem link. Using any networked controller as a bridge, you can view, read, and write data into any unit, as well as upload and download programs.



This feature can also allow you to view your network via a SCADA program.

## CANbus Networking

If your model controller includes a CANbus port, you can implement the following methods of data exchange:

**PLC to PLC**

- UniCAN DataCOM
  This enables a PLC in the CANbus network to exchange data with up to 60 other networked PLCs.
- CANbus ISC (Inter-Slave Connection)
  Data is broadcast over the CANbus network via system operands, where each unit can read information from up to **8** other PLCs in the network, including M90/91 PLCs that support CANbus.

**PLC to Remote Device**

- CANopen
  Supports data communication with remote devices according to the CANopen standard.

- CANbus, Layer 2
  Enables Vision controllers to send and receive data with remote devices that support CANbus protocol.

**Access a PLC via PC**

- You can access a PLC within a network by using one of the PLCs as a 'bridge'.
  Accessing a Networked PLC via PC

**'Simultaneous Modes' Enhanced Vision only**

- You can run the following modes simultaneously **if you select the relevant option in the COM Init function**:
  - CANopen + UniCAN
  - CANopen + CANbus, Layer 2
  - CANopen + J1939

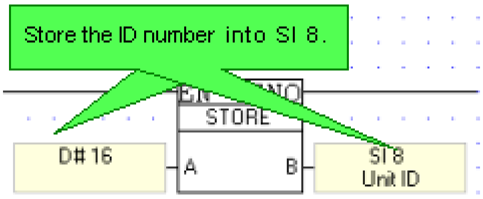
**Note •** Information Mode offers a CANbus 'sniffer' to enable you to monitor communication flow.

### Getting Started with CANbus

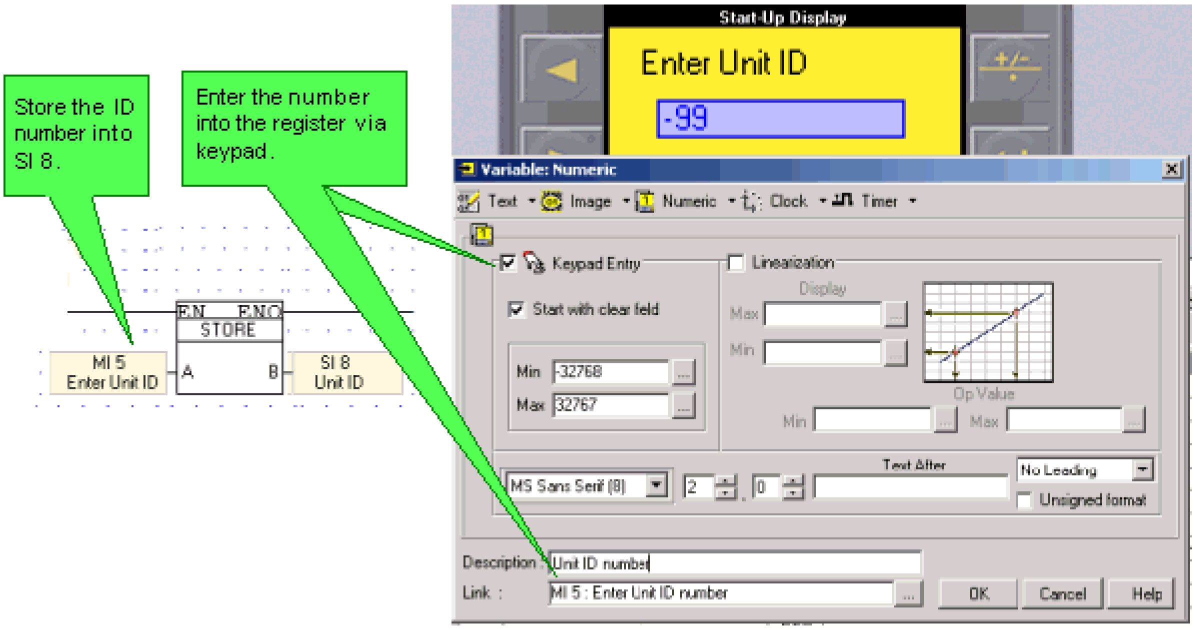
How to assign a unique Unit ID number to each networked controller

CANbus ID numbers range from 1 to 63. The ID number is contained in SI 8. You can assign an ID number by:

- Entering a value into SI 8 via Information Mode.
- Storing a value into SI 8 as shown below, either by storing a constant number or by linking a register value.



Linking a register value allows you to enable an end-user to enter an Unit ID number via the HMI panel. You can create a Display for this purpose as shown below.



☑ Keypad Entry

☑ Start with clear field

Min -32768

Max 32767

□ Linearization

MS Sans Serif (8) 2 0

No Leading

□ Unsigned format

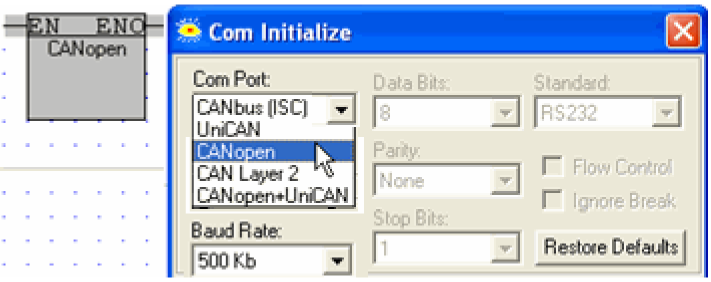
Description: Unit ID number

Link : MI 5 : Enter Unit ID number

**Note**
- The default ID # is **1**.
- UniCAN ID # range is from 1-60.

How to initialize the CANbus port

You must initialize the CANbus ports of all controllers in the network to the desired CANbus mode, by using a COM Port Init FB as shown below.



Com Port: CANbus (ISC)
- UniCAN
- CANopen
- CAN Layer 2
- CANopen+UniCAN

Baud Rate: 500 Kb

Data Bits: 8

Standard: RS232

Parity: None

Stop Bits: 1

□ Flow Control

□ Ignore Break

**Note**
- The baud rates of all controllers in the network must be set to the same rate. Baud rate is limited by cable length, as described in the CANbus Specifications section below.
- The COM Init FB used to initialize the CANbus port should be included in the Main Routine of the Ladder application.

CANbus Specifications

Power Requirements: 24VDC ( ±4%), 40mA max. per unit

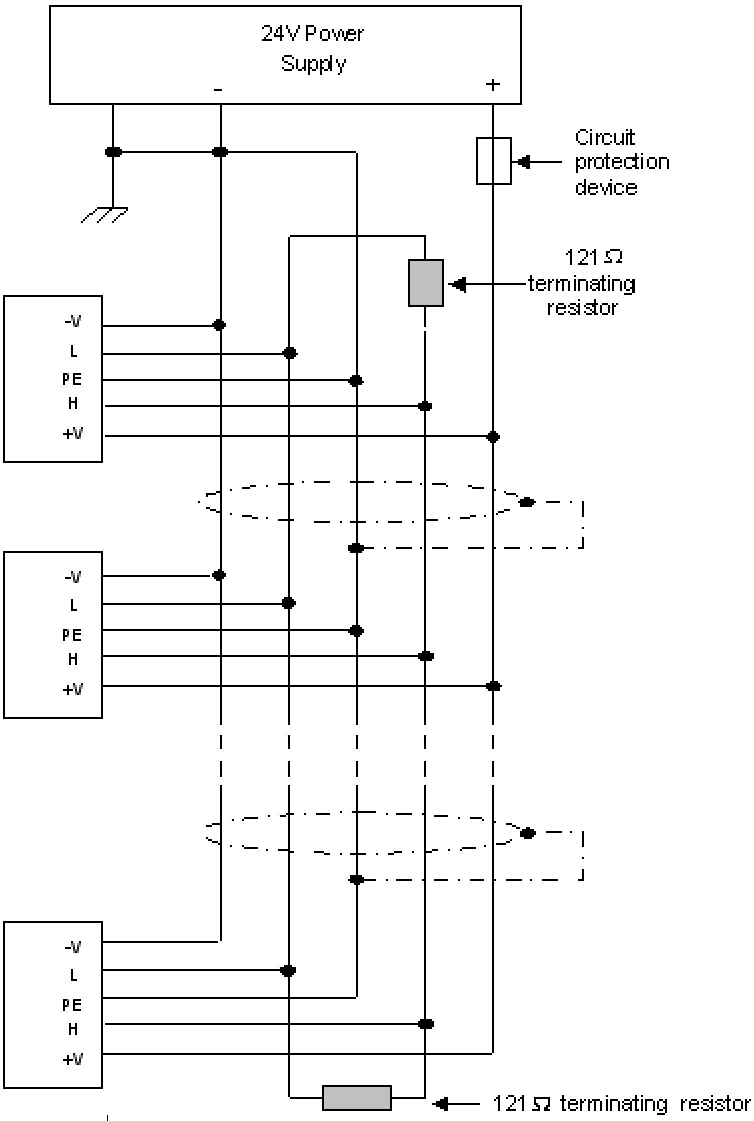
Galvanic Isolation between CANbus and controller: Yes

| **Baud rate** | **Max. Network Cable Length:** |
|---|---|
| 1 Mbit/s | 25 m |
| 500 Kbit/s | 100 m |
| 250 Kbit/s | 250 m |
| 125 Kbit/s | 500 m |
| 100 Kbit/s | 500 m |
| 50 Kbit/s | 1000 m |
| 20 Kbit/s | 1000 m |

**Note •** Cable lengths over 500 meters require an additional power supply.

| **Wiring Considerations** |
|---|
| Use twisted-pair cable. DeviceNet® thick shielded twisted pair cable is recommended. |
| Network terminators: These are supplied with the controller. Place terminators at each end of the CANbus network. Resistance must be set to 1%, 121Ω, 1/4W. |
| Connect the ground signal to the earth at only one point, near the power supply. |
| The network power supply need not be at the end of the network. |
| Maximum number of controllers in a network: 63. |

Wiring Diagram


24V Power Supply
-
+
Circuit protection device
121 Ω terminating resistor
-V
L
PE
H
+V
-V
L
PE
H
+V
-V
L
PE
H
+V
121 Ω terminating resistor

**Note •** Unitronics' CANbus control network is run by a **separate** isolated power supply that is **not** part of the network power supply.

### About CANbus System Operands

The function of some operands depends on whether the CANbus network is defined as CANbus ISC, CANopen or UniCAN. The operand function changes according to the type selected in the COM Init function

### CANbus, SBs 200-237

To learn how to use these operands to communicate data, check the topic CANbus ISC, via Network Operands.

| # | Description | Turns ON when: | Turns OFF when: | Reset by: |
|---|---|---|---|---|
| SB 236 | CANbus Network communication error | | Error is fixed. | |
| SB 237 | CANbus Network disabled | | | |

## When using CANbus ISC

To learn how to use these operands to communicate data, check the topic CANbus ISC, via Network Operands.

| # | Description | Turns ON when: | Turns OFF when: | Reset by: |
|---|---|---|---|---|
| SB 200 | CANbus Network operand | | | |
| SB 201 | CANbus Network operand | | | |
| SB 202 | CANbus Network operand | | | |
| SB 203 | CANbus Network operand | | | |
| SB 204 | CANbus Network operand | | | |
| SB 205 | CANbus Network operand | | | |
| SB 206 | CANbus Network operand | | | |
| SB 207 | CANbus Network operand | | | |
| SB 208 | CANbus Network operand | | | |
| SB 209 | CANbus Network operand | | | |
| SB 210 | CANbus Network operand | | | |
| SB 211 | CANbus Network operand | | | |
| SB 212 | CANbus Network operand | | | |
| SB 213 | CANbus Network operand | | | |
| SB 214 | CANbus Network operand | | | |
| SB 215 | CANbus Network operand | | | |
| SB 237 | CANbus Network disabled | | | |

When using UniCAN

| SB# | Description | Turned ON | Turned Off | Comments |
|---|---|---|---|---|
| 200 | Broadcast bit | When UniCAN broadcast MB is received whose status is ON. | By user | The user must initialize this SB |
| 201 | High Priority Send Buffer Status | When full | When not full | |
| 202 | Low Priority Send Buffer Status | When full | When not full | |

## CANopen, SBs 240-243

| SB# | Description | Turns ON when: | Turns OFF when: | Reset by: |
|---|---|---|---|---|
| SB 240 | CANopen: Configuration downloaded | The CANopen Configuration FB is downloaded to the PLC | No CANopen Configuration is present | PLC |
| SB 241 | CANopen: Configured | CANopen Configuration was successful | CANopen Configuration failed | PLC |
| SB 242 | CANopen: SDO in Progress | SDO is busy transferring data | SDO is not in progress | PLC |
| SB 243 | CANopen: SDO transfer failed | SDO data transfer fails | SDO transfer begins | PLC |

## CANbus, SIs 200-201, 236-237, 240-245

When using CANopen

| SI# | Description | Value | Comments |
|---|---|---|---|
| SI 211 | CANopen: Number of received messages | Shows the number of received messages in the Receive buffer (except for SDOs) | Maximum number of messages=128 |
| SI 212 | CANopen: Number of Send PDO1 | Shows the number of PDO1 messages currently in the PDO1 Send buffer | |
| SI 213 | CANopen: Number of Send PDO2 | Shows the number of PDO2 messages currently in the PDO2 Send buffer | |
| SI 214 | CANopen: Number of Send PDO3 | Shows the number of PDO3 messages currently in the PDO3 Send buffer | |
| SI 215 | CANopen: Number of Send PDO4 | Shows the number of PDO4 messages currently in the PDO4 Send buffer | |
| SI 216 | CANopen: Number of Send RTR PDO1 | Shows the number of RTR PDO1 messages currently in the PDO1 Send buffer | |
| SI 217 | CANopen: Number of Send RTR PDO2 | Shows the number of RTR PDO2 messages currently in the PDO2 Send buffer | |
| SI 218 | CANopen: Number of Send RTR PDO3 | Shows the number of RTR PDO3 messages currently in the PDO3 Send buffer | |
| SI 219 | CANopen: Number of Send RTR PDO4 | Shows the number of RTR PDO4 messages currently in the PDO4 Send buffer | |
| SI 220 | CANopen: Number of Send SDOs | Shows the number of SDO messages currently in the Send buffer | |
| SI 221 | CANopen: Number of Send NMTs | Shows the number of NMT module control messages currently in the NMT Send buffer | |
| SI 222 | CANopen: Number of Send RTR NMTs | Shows the number of RTR NMT messages currently in the Send buffer | |
| SI 223 | CANopen: Send Buffer full (per type) | The bits in this register represent the different Send buffers (except for SDOs) | Maximum number of messages per buffer=8 |

When a bit is ON, the corresponding buffer is full.
High byte:| - | - | - | - | - | - | -| NMT mc |
Low byte: | PDO | -| RCV NMT | RCV PDO4| RCV PDO3 | RCV PDO2 | RCV PDO1 | RCV Emergency |

| | | | |
|---|---|---|---|
| SI 224 | CANopen: Number of received SDO messages | Shows the number of received SDOs currently in the Receive buffer | |
| SI 225 | CANopen: SDO status | The status codes are given below. | |

| Value | Message |
|---|---|
| 0 | No error |
| 1 | PLC in STOP mode |
| 2 | CANopen not configured, SB 241 is not set (after configuration) |
| 3 | Remote ID is 0 |

| | |
|---|---|
| 4 | Maximum SDO Upload length set to 0 |
| 5 | SDO in Progress; Download/Upload started while SB242 is ON |
| 6 | SDO in Progress Error; SB242 turned OFF during data transfer (system problem) |
| 7 | Illegal Operands used in SDO data transfer |
| 8 | Number of operands in data type exceeded |
| 9 | Process buffer not cleared before SDO Send (system problem) |
| 10 | Response Timeout exceeded |
| 11 | Receive Error |
| 12 | Reserved by CIA |
| 13 | Receive Buffer full; more than 127 segments in a block (system problem) |
| 14 | Receive Error Toggle bit ON (error in domain segment) |
| 15 | Receive domain segment Abort; error code given in SDW 34 |
| 16 | Byte number error |
| 17 | Number of bytes is zero |
| 18 | Number of bytes exceeds the maximum upload length |
| 19 | Machine State error (system problem) |
| 20 | Receive Error in block size transferred from the remote device |
| 21 | Send Timeout exceeded |
| 22 | Sequence error in the number of segments in block transfer |
| 23 | CRC error, block transfer |

### When using CANbus ISC

| | | |
|---|---|---|
| SI 200 | CANbus Network operand | |
| SI 201 | CANbus Network operand | |
| SI 236 | CANbus Network communication error code | **Value** **Message**<br>1 No Acknowledgement<br>2 CANbus OFF<br>4 CANbus Warning error<br>10 ISC receiving TimeOut |
| SI 237 | CANbus Network: failed unit ID | |
| SI 240 | SIs 240-243 comprise a bitmap indicating which unit is in error. If, for example, the network includes unit ID numbers 8, 9 and 13, and PLC #9 cannot be accessed, then the ninth bit in SI240 will turn ON. When the error is fixed, the bit falls to OFF | |
| SI 241 | | |
| SI 242 | | |
| SI 243 | | |

When using UniCAN

| SI# | Description | Comments |
|---|---|---|
| 200 | When a UniCAN Broadcast message is received, SI 200 contains the ID number of the sending unit. | The user must initialize these SIs |
| 201 | When a UniCAN Broadcast message is received, SI 201 contains the value of the MI that is broadcast. | |
| 202 | Number of Send messages waiting in High Priority buffer | Automatically updates |
| 203 | Number of Send messages waiting in Low Priority buffer | |
| 204 | Number of Received messages waiting in buffer | |
| 240 | These provide a bitmap of controllers 1-60 in | When the controller receives a message, |

| | | |
|---|---|---|
| 241 | the UniCan network. | the appropriate bit turns ON. These bits are reset by the Answer Received function. |
| 242 | | |
| 243 | | |

## CANbus SDWs

| # | Description | Value | Comments |
|---|---|---|---|
| SDW 7 | UniCAN, CANbus ISC Error | If not 0, contact technical support | |
| | CANopen: Number of failed Send attempts | Number of times that data send failed | |
| SDW 8 | CANopen: Number of failed Sync attempts | Number of times that send SYNC failed | |
| SDW 56 | UniCAN Send message counter | Is initialized when CANbus Port Init runs, then increments at every UniCAN Send. | Note that only messages sent from a UniCAN Send are counted |
| | CANopen: PDO Send Counter | Byte structure:<br>PDO4 \| PDO3 \| PDO2 \| PDO1 | |
| SDW 57 | UniCAN Receive message counter | Is initialized when CANbus Port Init runs, then increments at every UniCAN Receive. | Note that only messages received from a UniCAN Send are counted, not Broadcast messages or Check if Alive responses. |
| | CANopen: NMT/SDO Send Counter | High bits: NMT<br>Low bits: SDO | |
| SDW 29 | CANopen: Bus is OFF Counter | Number of times bus was OFF | |
| SDW 33 | CANopen: SDO Number of Bytes | SDO upload: number of bytes received<br>SDO download: number of bytes sent | |
| SDW 34 | CANopen: Abort Code in SDO Abort | | |

| Value | Message |
|---|---|
| 0503 0000h | Toggle bit not alternated |
| 0504 0000h | SDO protocol timed out |
| 0504 0001h | Client/server command specifier not valid or unknown |
| 0504 0002h | Invalid block size (block mode only) |
| 0504 0003h | Invalid sequence number (block mode only) |
| 0504 0004h | CRC error (block mode only) |
| 0504 0005h | Out of memory |
| 0601 0000h | Unsupported access to an object |
| 0601 0001h | Attempt to read a write only object |
| 0601 0002h | Attempt to write a read only object |
| 0602 0000h | Object does not exist in the object dictionary |
| 0604 0041h | Object cannot be mapped to the PDO |
| 0604 0042h | The number and length of the objects to be mapped would exceed PDO length |
| 0604 0043h | General parameter incompatibility reason |
| 0604 0047h | General internal incompatibility in the device |
| 0606 0000h | Access failed due to a hardware error |

| | |
|---|---|
| 0607 0010h | Data type does not match, length of service parameter does not match |
| 0607 0012h | Data type does not match, length of service parameter too high |
| 0607 0013h | Data type does not match, length of service parameter too low |
| 0609 0011h | Sub-index does not exist |
| 0609 0030h | Invalid value for parameter (upload only) |
| 0609 0031h | Value of parameter written too high (upload only) |
| 0609 0032h | Value of parameter written too low (upload only) |
| 0609 0036h | Maximum value is less than minimum value |
| 060A 0023h | Resource not available: SDO connection |
| 0800 0000h | General error |
| 0800 0020h | Data cannot be transferred or stored to the application |
| 0800 0021h | Data cannot be transferred or stored to the application because of local control |
| 0800 0022h | Data cannot be transferred or stored to the application because of the present device state |
| 0800 0023h | Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error) |
| 0800 0024h | No data available |

| | |
|---|---|
| SDW 36 | CANopen: Bus OFF error |

| Value | Message |
|---|---|
| 0 | No error |
| 1 | Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed |
| 2 | Form Error: Wrong format in fixed format part of a received frame |
| 3 | AckError: The message this CAN controller transmitted was not acknowledged by another node |
| 4 | Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (“1”), but the monitored bus value was dominant |
| 5 | During busoff recovery this is set each time a sequence of 11 recessive bits is monitored. This enables the CPU to monitor the proceeding of the busoff recovery sequence (indicates that the bus is not stuck at dominant or continuously disturbed |
| 6 | CRCError: The received CRC check sum is incorrect |
| 7 | Unused code: may be written by the CPU to check for updates |

## Older Versions Issue: CANopen compatibility break

The CANopen System Operand addressing scheme changed as of **VisiLogic version 7.00, OS 2.00 (Standard Division) OS 5.02 (Enhanced Division**).
You can choose to continue editing older applications using previous VisiLogic versions contained in the VisiLogic Version Swapper. Note that in this case the controller **must** use an older OS.
However, if you choose to update the application and OS, note that you must change the system operands according to the following table.

| **System Integers** | |
|---|---|
| **Old** | **New** |
| 200 | 211 |
| 201 | 212 |
| 202 | 213 |
| 203 | 214 |
| 204 | 215 |
| 205 | 216 |
| 206 | 217 |
| 207 | 218 |
| 208 | 219 |
| 240 | 220 |
| 241 | 221 |
| 242 | 222 |
| 243 | 223 |
| 244 | 224 |
| 245 | 225 |

| **System Double Words** | |
|---|---|
| **24** | 7 |
| 25 | 8 |
| 26 | 56 |
| 28 | 57 |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

### CANbus Compilation Errors

At compilation or download, VisiLogic may display this error: "Illegal CAN Communication Driver specification - see help"

This may be because the project contains:

- CANopen, UniCAN, CAN Layer 2, or CANbus ISC operands; but the CANbus port is not initialized to the correct CANbus **protocol type**
- More than one CANopen Configuration element
- More than one CANbus port initialization

## Assigning a Unit ID number

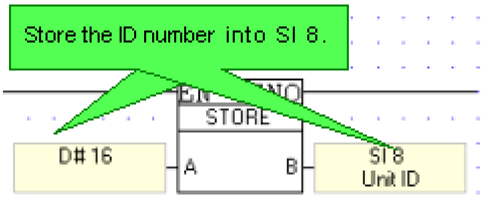
When you create an controller network, you must assign a Unit ID number to each controller. A Unit ID number is unique. The same ID number **must not** be assigned to more than one device within a network.

You use this number for two purposes:

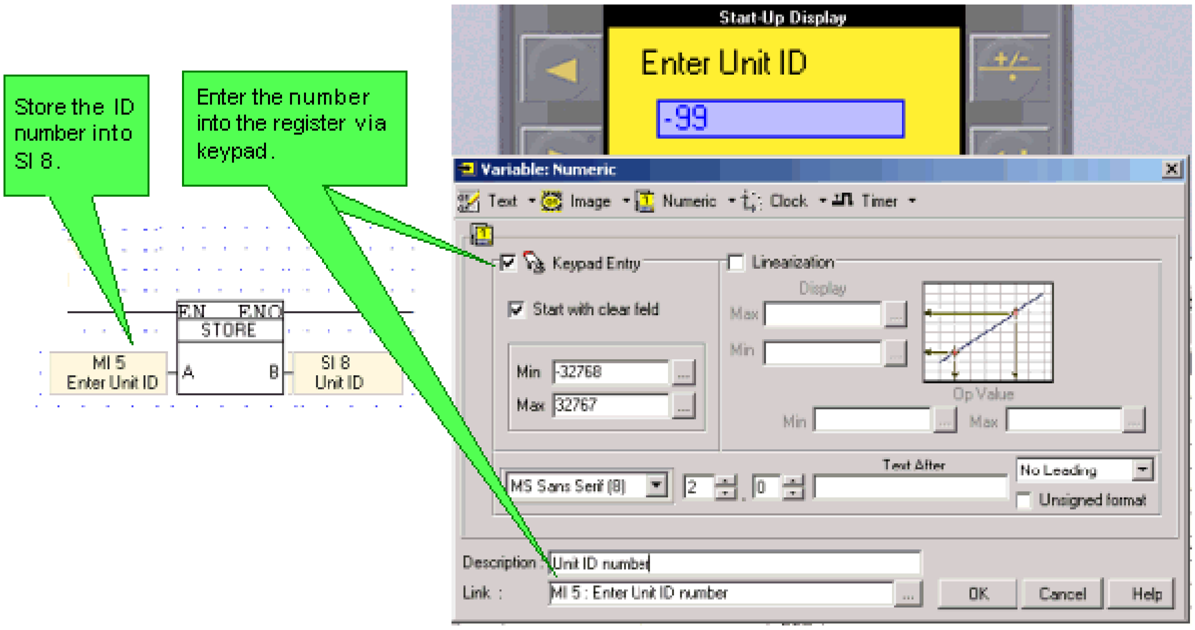
- To enable controllers to exchange data.
- To access a networked controller via your PC.

The ID number is contained in SI 8. You can assign an ID number by:

- Entering a value into SI 8 via Information Mode.
- Storing a value into SI 8 as shown below, either by storing a constant number or by linking a register value.
- Using the Set PLC ID Number function.


Store the ID number into SI 8.

Linking a register value allows you to enable an end-user to enter an Unit ID number via the HMI panel. You can create a Display for this purpose as shown below.


Store the ID number into SI 8.

Enter the number into the register via keypad.

Start-Up Display

Enter Unit ID

-99

Variable: Numeric

☑ Keypad Entry

☑ Start with clear field

Min -32768

Max 32767

□ Linearization

MS Sans Serif (8) 2 0

Text After

No Leading

□ Unsigned format

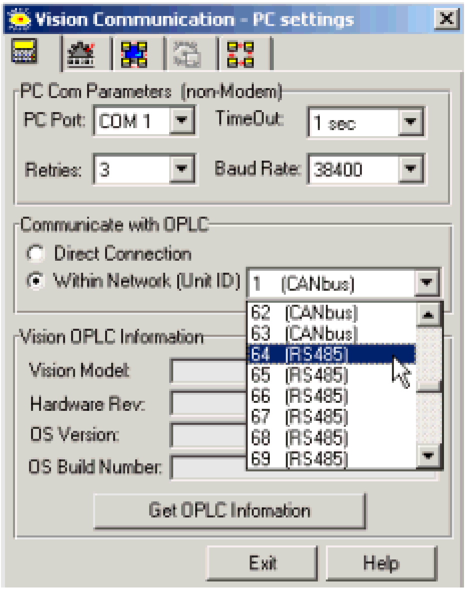
Description : Unit ID number

Link : MI 5 : Enter Unit ID number

**Note** • The default ID # is **1**.

UniCAN ID numbers range is from 1-60.

• ID numbers 1-63 are reserved for controllers linked via CANbus ISC; ID numbers 64-127 are reserved for controllers networked via RS485. Using this range of ID numbers prevents a polled controller from attempting to act as a CANbus bridge, preventing it from attempting to locate the requested controller.



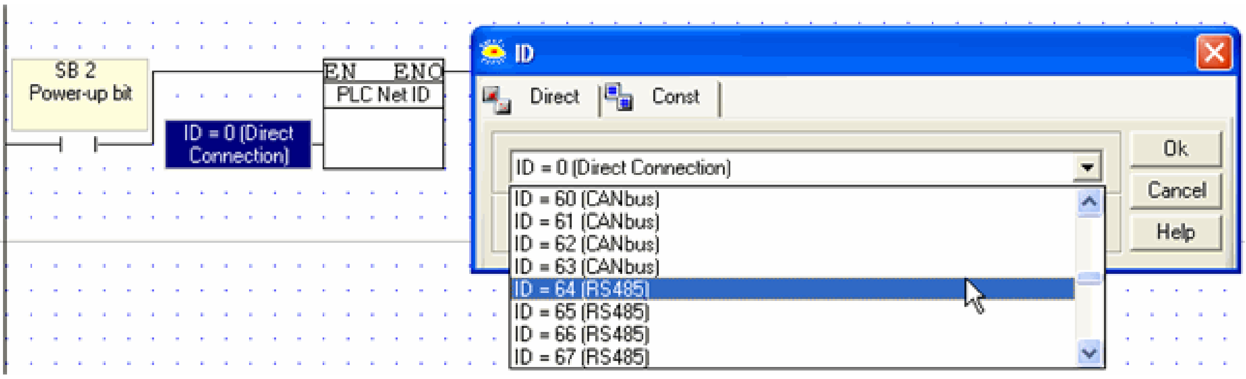
PC Port: COM 1
TimeOut: 1 sec
Retries: 3
Baud Rate: 38400
○ Direct Connection
● Within Network (Unit ID) 1 (CANbus)

62 (CANbus)
63 (CANbus)
64 (RS485)
65 (RS485)
66 (RS485)
67 (RS485)
68 (RS485)
69 (RS485)

## Set PLC ID Number

Located on the COM menu, this function enables you to assign a unique ID number to a PLC. This name can, for example, be used to identify the PLC for CANbus or RS485 networking purposes.

This should be assigned as a power-up task.



ID = 0 (Direct Connection)

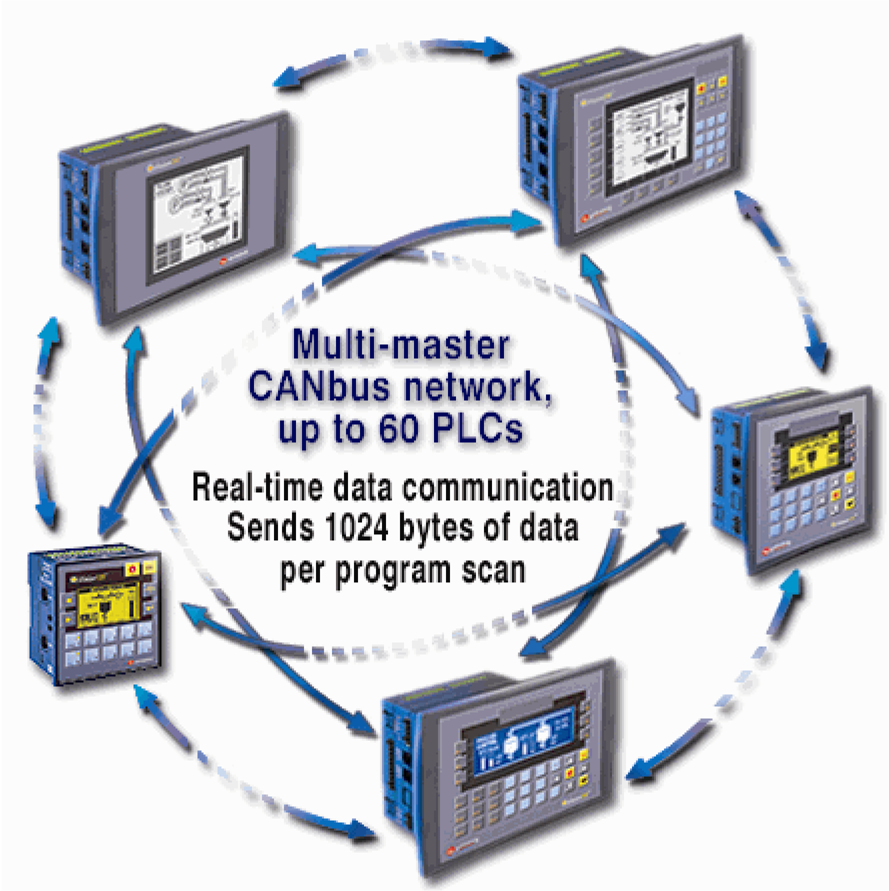
ID = 60 (CANbus)
ID = 61 (CANbus)
ID = 62 (CANbus)
ID = 63 (CANbus)
ID = 64 (RS485)
ID = 65 (RS485)
ID = 66 (RS485)
ID = 67 (RS485)

**Notes** • When the function is activated, the ID is written into the PLC. If the value is supplied via Indirect Vector, note that simply storing a value into the vector will not rewrite the ID. To rewrite the ID, the value must first be stored in the appropriate vector, and then the function must be activated.

• This feature is not supported by the V120-12 series.

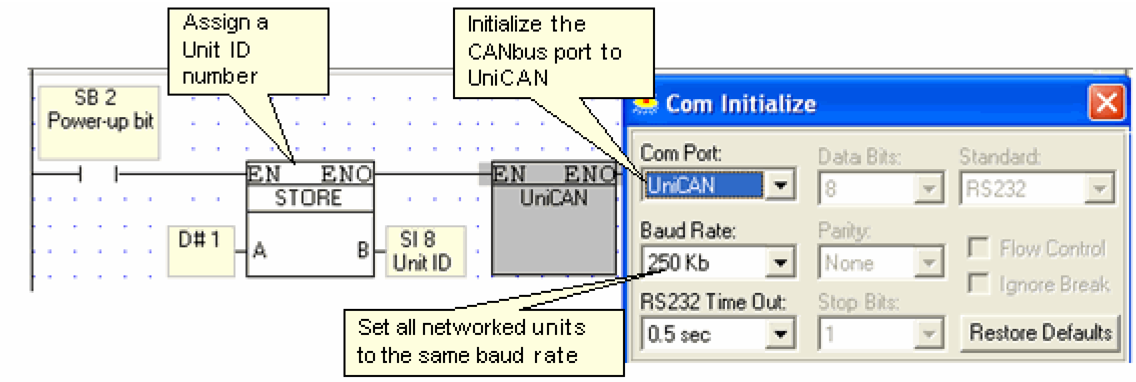
## CANbus UniCAN

UniCAN enables fast data communications. Via UniCAN, a Unitronics' PLC can exchange data with up to 60 other networked PLCs. A UniCAN application can send up to 32 messages, each containing 16 integer values, totaling 512 MI register values during a single program scan.



When PLCs are connected to the CANbus network and their CANbus ports initialized to UniCAN, you can use the UniCAN functions, located on the Communications menu, to transfer data between units or check a unit's status.

Before using UniCAN functions, assign each networked controller a unique Unit ID number and initialize the CANbus port to UniCAN. Note that you can run both UniCAN and CANopen if you select the relevant Com Port option in the COM Init function.



Assign a Unit ID number

Initialize the CANbus port to UniCAN

Set all networked units to the same baud rate

Com Port: UniCAN

Data Bits: 8

Standard: RS232

Baud Rate: 250 Kb

Parity: None

□ Flow Control

□ Ignore Break

RS232 Time Out: 0.5 sec

Stop Bits: 1

Restore Defaults

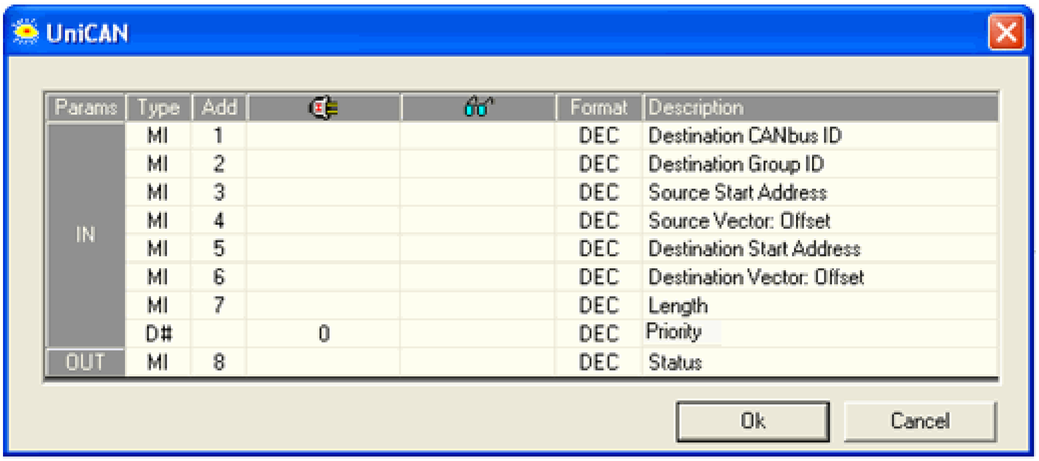
**Note •** Baud Rate is a function of cable length, as described in CANbus Hardware Specifications.

### Send Registers

The Send function enables you to:

- Read a vector of MIs in the source PLC
- Write the values to a vector in a target PLC on the CANbus network

You can send a vector of values up to 16 integers long in a single Send function.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | MI | 1 | | | DEC | Destination CANbus ID |
| | MI | 2 | | | DEC | Destination Group ID |
| | MI | 3 | | | DEC | Source Start Address |
| | MI | 4 | | | DEC | Source Vector: Offset |
| | MI | 5 | | | DEC | Destination Start Address |
| | MI | 6 | | | DEC | Destination Vector: Offset |
| | MI | 7 | | | DEC | Length |
| | D# | | 0 | | DEC | Priority |
| OUT | MI | 8 | | | DEC | Status |

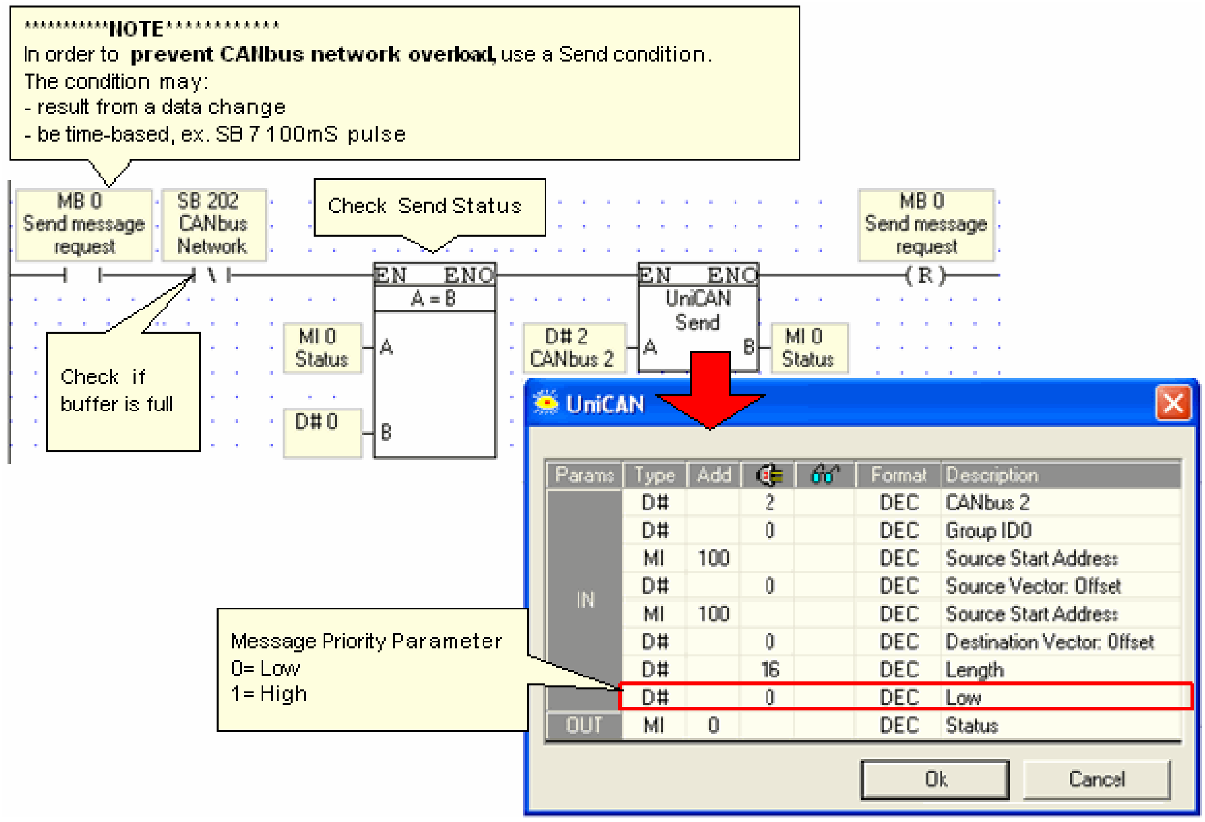
| Parameter | Type | Function |
|---|---|---|
| Destination CANbus ID | Constant or MI | The CANbus ID number of the target PLC. |
| Destination Group ID | Not currently supported. Enter ID0. | |
| Source Start Address | MI | The start address of the source vector in the source PLC. |
| Source Vector: Offset | Constant or MI | The offset from the start address in the source PLC. |
| Destination Start Address | MI | The start address of the destination vector in the target PLC. |

<table>
<tr><td>Destination Vector: Offset</td><td>Constant or MI</td><td>The offset from the destination address in the target PLC.</td></tr>
<tr><td>Length</td><td>Constant or MI</td><td>This determines the length of the MI vector that is read from the source PLC and then written to the target PLC.</td></tr>
<tr><td>Priority</td><td>Constant</td><td>Select High or Low priority.<br>UniCAN uses 2 Send Message buffers, one for High Priority and one for Low Priority. High Priority messages are always sent first. Each buffer is based on a FIFO stack that may contain up to 16 messages containing up to 16 MIs each, a total of 512 integer values for both buffers.</td></tr>
<tr><td>Status</td><td>MI</td><td>The Status MI indicates status and error messages as listed below.<br>The Status MI updates when the Send function is activated. If an error occurs, the status indication updates. If there is no error, and the message is sent to the buffer, the status is 1. Once the message is sent, the MI updates to 0.<br><br>Note that each Send operation has its own Status MI.<br><br>The Status MI should be initialized at Power-up.
<table>
<tr><th>#</th><th>Status Message</th></tr>
<tr><td><b>0</b></td><td>Message successfully sent. Use this to check if the previous message was sent.</td></tr>
<tr><td><b>1</b></td><td>Message is ready to be sent, but the network is currently busy</td></tr>
<tr><td><b>2</b></td><td>Destination unit ID number is greater than 60</td></tr>
<tr><td><b>3</b></td><td>Illegal Group ID number</td></tr>
<tr><td><b>4</b></td><td>Data length exceeds 32 bytes</td></tr>
<tr><td><b>5</b></td><td>Master Controller: Source Start Address is illegal</td></tr>
<tr><td><b>6</b></td><td>Slave Controller: Destination Start Address is illegal</td></tr>
<tr><td><b>7</b></td><td>Priority is not 0 or 1</td></tr>
<tr><td><b>8</b></td><td>Send high priority FIFO is full</td></tr>
<tr><td><b>9</b></td><td>Send low priority FIFO is full</td></tr>
</table>
</td></tr>
</table>

**Note** • In order to prevent CANbus network overload, use a Send condition. Do not place the Send function directly on the left Ladder rail.

When a buffer is full, the related SB turns ON. Check the buffer status by using an inverted contact of the relevant SB as a Send condition: SB 201: High Priority Buffer, SB 202 Low Priority Buffer.

• Check the Status MI before Send.



***********NOTE************
In order to **prevent CANbus network overload**, use a Send condition.
The condition may:
- result from a data change
- be time-based, ex. SB 7 100mS pulse

Check Send Status

Check if buffer is full

Message Priority Parameter
0= Low
1= High

| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 2 | | DEC | CANbus 2 |
| | D# | | 0 | | DEC | Group ID0 |
| | MI | 100 | | | DEC | Source Start Address |
| | D# | | 0 | | DEC | Source Vector: Offset |
| | MI | 100 | | | DEC | Source Start Address |
| | D# | | 0 | | DEC | Destination Vector: Offset |
| | D# | | 16 | | DEC | Length |
| | D# | | 0 | | DEC | Low |
| OUT | MI | 0 | | | DEC | Status |

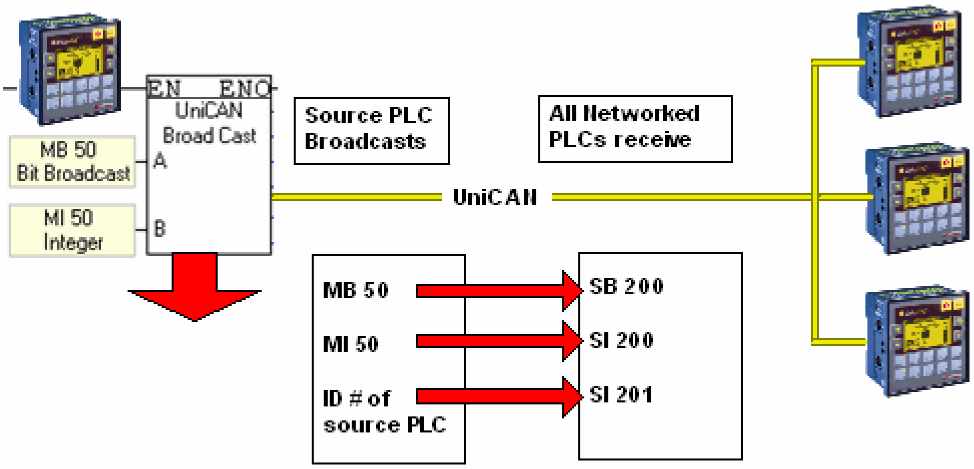
### Broadcast

Use Broadcast to send an MB and an MI value to all controllers in the UniCAN network.

Each networked controller receives the values into system operands: the MB into SB 200, the MI into SI 200, and ID # of the source controller into SI 201.

| | |
|---|---|
| **Note** • | SB 200, SI 200 and SI 201 must be initialized by the user. |
| • | A new Broadcast overwrites any previous values. The values remain until they are overwritten or initialized. |
| • | Use an inverted contact of SB 203, UniCAN Broadcast in Progress, as a Broadcast condition. |

If the source controller in the figure below is assigned ID# 3, contains 123 in MI 50, and has MB 50 ON: **all** controllers in the network will have 3 written into SI 201, 123 written into SI 200, and SB 200 will be turned ON.

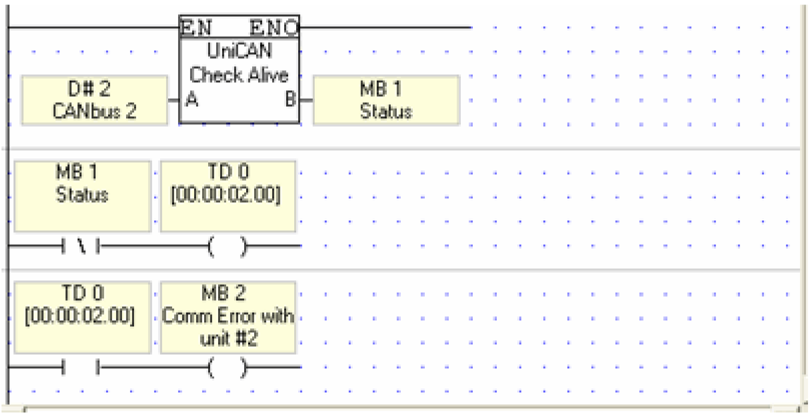


### Check Alive Signal

Each 0.5 seconds, a unit sends an 'Alive' signal to the CANbus network. The Check Alive Signal function receives the signal.

When the signal for the networked unit whose ID # is given is received, the Status MB turns ON for a single scan. This short time period means that you will not be able to see the signal in Online mode.

In the nets below, if an 'Alive' signal from PLC #2 is not registered for 2 seconds, the program turns on an error bit.

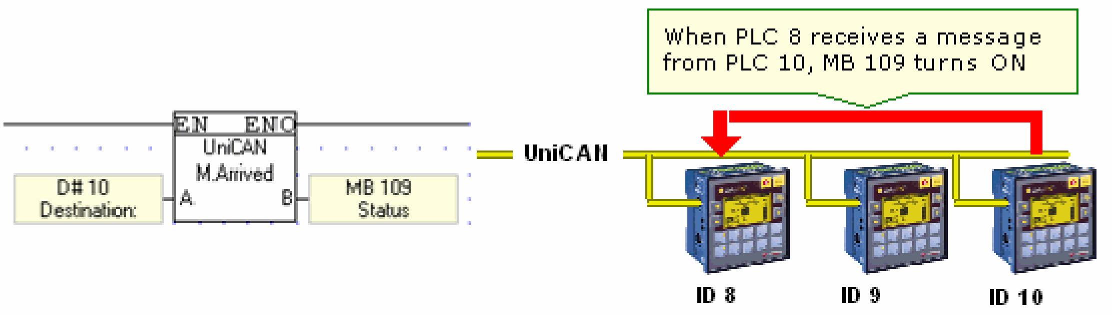


### Message Arrived

Message Arrived should be placed on the left-hand Ladder rail.

When a message arrives from the networked unit whose ID # is given in the function, the Status MB turns ON.

When the message is received, the corresponding bit in SIs-240-243, which provide a bitmap of all units in the UniCAN network, turns OFF.



### UniCAN System Operands

| SB# | Description | Turned ON | Turned Off | Comments |
|---|---|---|---|---|
| 200 | Broadcast bit | When UniCAN broadcast MB is received whose status is ON. | When UniCAN broadcast MB is received whose status is OFF. | The user must initialize this SB |
| 201 | High Priority Send Buffer Status | When full | When not full | Use the negative transition of this SB as a Send UniCAN condition for High Priority messages |
| 202 | Low Priority Send Buffer Status | When full | When not full | Use the negative transition of this SB as a Send UniCAN condition for Low Priority messages |

| SI# | Description | Comments |
|---|---|---|
| 200 | When a UniCAN Broadcast message is received, SI 200 contains the ID number of the sending unit. | The user must initialize these SIs |
| 201 | When a UniCAN Broadcast message is received, SI 201 contains the value of the MI that is broadcast. | |
| 202 | Number of Send messages waiting in High Priority buffer | Automatically updates |
| 203 | Number of Send messages waiting in Low Priority buffer | |
| 204 | Number of Received messages waiting in buffer | |
| 240 | These provide a bitmap of controllers 1-60 in the UniCAN network. | When the controller receives a message, the appropriate bit turns ON.<br>These bits are reset by the Answer Received function. |
| 241 | | |
| 242 | | |
| 243 | | |

| SDW# | Description | Comments |
|---|---|---|
| 24 | If not 0, contact technical support | |
| 26 | Send message counter. Note that only messages sent from a UniCAN Send are counted | SDW 26 and 28 are initialized at COM Init. |
| 28 | Receive message counter. Note that only messages | |

received from a UniCAN Send are counted, not Broadcast messages or Check if Alive responses.

## CANopen

Vision controllers support data communication with remote devices such as frequency converters according to the CANopen standard, compliant with CiA DS 301.

A Vision acts as a CANopen master, supporting the following functions:

- Send PDO/ Send PDO RTR
- Send NMT/ Send NMT RTR
- Send SYNC
- SDO Download STR /Upload STR

**'Simultaneous Modes'**

Enhanced Vision only: You can run the following modes simultaneously if you select the relevant option in the COM Init function:

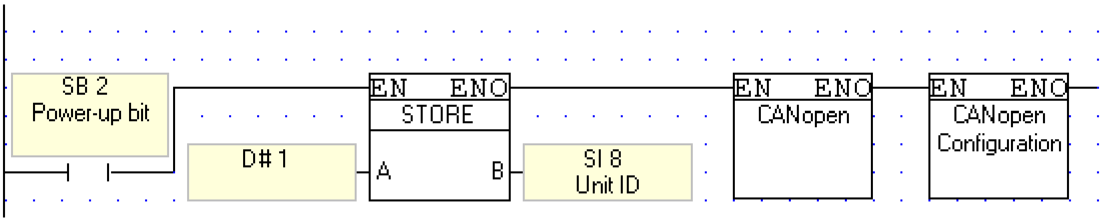
- CANopen + UniCAN
- CANopen + CANbus, Layer 2
  Note that CANopen COB ID numbers are 11-bit. If your CANbus Layer 2 application uses 11-bit identifiers, make sure that the ID/identifiers are unique. If a message arrives through the bus marked with the same number, CANbus Layer 2 will scan the message but CANopen will not.

### Using CANopen

**Note •** Prevent CANbus network overload by using a Send condition to activate functions. Do not place them directly on the left Ladder rail.

At power-up:

1. Assign each networked controller a unique Unit ID number.
2. Initialize the CANbus port to CANopen via a COM Init function.
3. Activate the CANopen configuration.

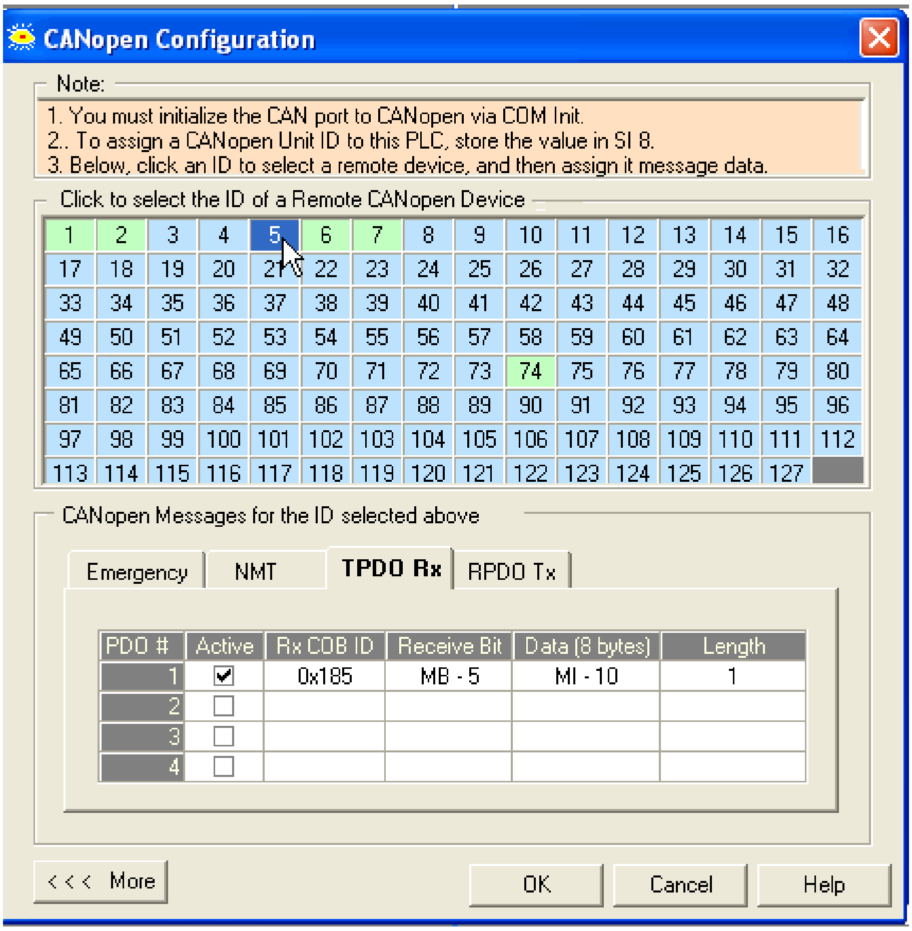


Note that you can use UniEDS to open CANopen EDS files. UniEDS is located on the Tools menu.

### Configuration

A Configuration contains the messages for the remote CANopen node. The Configuration determines:

- Which data are sent to the node
- Which data may be read from the node, and where this data will be stored in the Vision.



Note:

1. You must initialize the CAN port to CANopen via COM Init.
2.. To assign a CANopen Unit ID to this PLC, store the value in SI 8.
3. Below, click an ID to select a remote device, and then assign it message data.

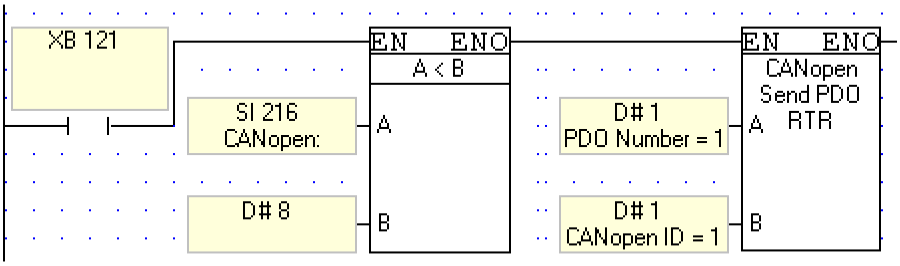
Click to select the ID of a Remote CANopen Device

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |
| 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 |
| 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 |
| 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | |

CANopen Messages for the ID selected above

| PDO # | Active | Rx COB ID | Receive Bit | Data (8 bytes) | Length |
|---|---|---|---|---|---|
| 1 | ☑ | 0x185 | MB - 5 | MI - 10 | 1 |
| 2 | □ | | | | |
| 3 | □ | | | | |
| 4 | □ | | | | |

Send PDO RTR is a 'read' request, where the PLC requests data from the remote node. When the Configuration above runs with a Send PDO RTR as shown below, PDO message 1 is activated; 1 byte will be read from Node 5 into MI 10 in the PLC, and the Receive Bit for that message will turn ON.



### Defining CANopen messages

Click on an ID number and then select the desired tab. When you activate a message, the Select Operand and Address box opens, enabling you to select message parameters.

**Note •** Receive bits must be reset by the user.

| Tab Name | Parameter |
|---|---|
| **Emergency** | This receives the 8-byte CANopen Emergency message. Click Enable to define Emergency parameters: Error Code, Error Register, and Manufacturer Specific bytes. |

| | |
|---|---|
| **NMT** | This receives the NMT node guarding data. |
| **TPDO Rx** | This is the PDO data that is **read from** the node when a Send PDO RTR function runs. You can define 4 TPDO Rx messages for each node. Click Activate to assign registers to a message and determine length (number of bytes). |
| **RPDO Tx** | This is the PDO data that is **written to** the node when a Send PDO function runs. You can define 4 RPDO Tx messages for each node. Click Activate to assign registers to a message and determine length (number of bytes). |

### Sending CANopen messages: Buffer Management

enable enableNote that most CANopen Send messages are queued in an internal buffer.

Each buffer can queue up to a certain number of messages, as shown in the following table.

If the buffer is full, and the ladder application sends a new message, a message will be lost. Each buffer is linked to:

- An SI that shows the number of messages currently in the buffer
- An SB that turns ON when the buffer is full, OFF when the buffer holds less than the maximum.

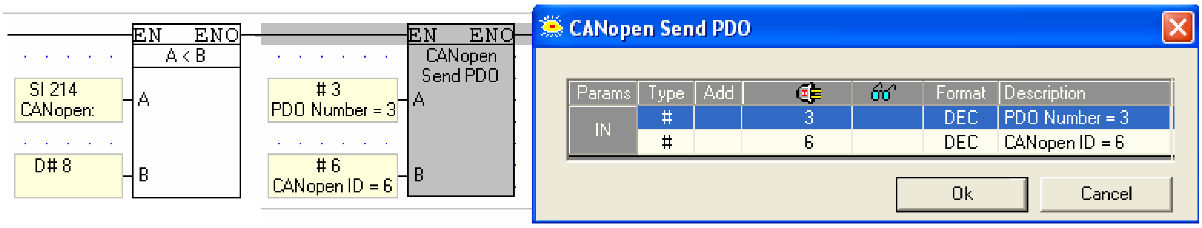
To avoid losing messages, use Send conditions that you link to an inverted contact of the appropriate SBs in your Ladder application to check whether there is room in the buffer.

| **Function Buffer** | **SI** | **SB** | **Maximum # of Messages** |
|---|---|---|---|
| Send PDO1 | 212 | 284 | 8 |
| Send PDO2 | 213 | 285 | 8 |
| Send PDO3 | 214 | 286 | 8 |
| Send PDO4 | 215 | 287 | 8 |
| RTR PDO1 | 216 | 288 | 12 |
| RTR PDO2 | 217 | 289 | 12 |
| RTR PDO3 | 218 | 290 | 12 |
| RTR PDO4 | 219 | 291 | 12 |
| Send NMT MC | 221 | 292 | 8 |
| Send RTR NMT | 222 | 293 | 12 |

### Send PDO / Send PDO RTR

The Send PDO function writes data **to** the node from the PLC master.

The Send PDO RTR function request data from the node, and reads data **from** the node to the PLC master.



CANopen Send PDO

| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | # | | 3 | | DEC | PDO Number = 3 |
| | # | | 6 | | DEC | CANopen ID = 6 |

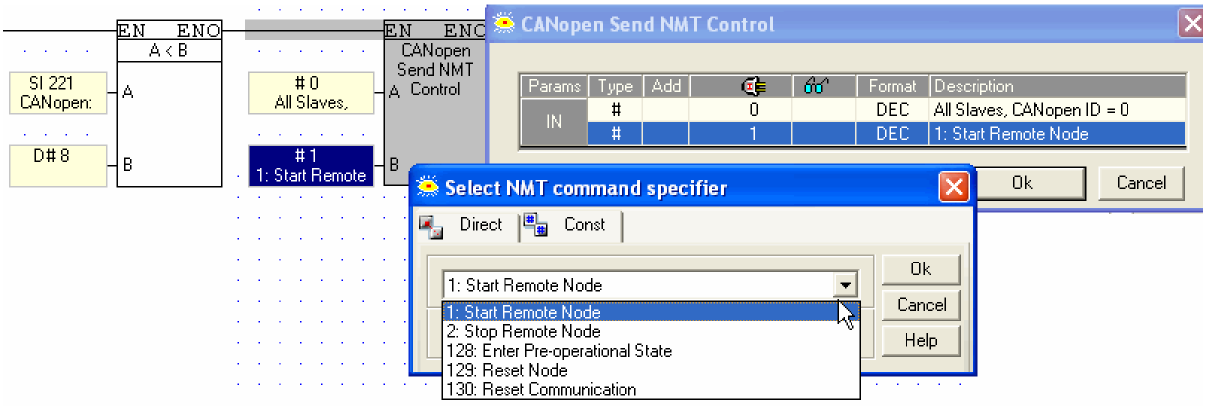
| Parameter | Type | Function |
|---|---|---|
| Select PDO Number | Constant | Select the number of the PDO assigned in the CANopen Configuration |
| Select CANopen: Remote ID | Constant | The CANbus ID number of the target device. |

### Send NMTControl / Send NMT Node Guard (Heartbeat)

This function enables you to send an NMT to ID 0 in order to broadcast to or read from to all nodes. This is sometimes referred to as a 'heartbeat check'.

- The Send NMT function issues NMT Module Control messages.
- The Send NMT Node Guard function provides CANopen Node Guarding, checking node status.

**Note •** When a node sends an NMT Boot-up message to the Vision master, the status is written to the NMT Status MB in the CANopen Configuration.



CANopen Send NMT Control

| Params | Type | Add |  |  | Format | Description |
|---|---|---|---|---|---|---|
| IN | # |  | 0 |  | DEC | All Slaves, CANopen ID = 0 |
|  | # |  | 1 |  | DEC | 1: Start Remote Node |

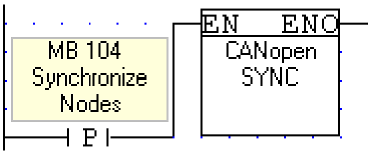
Select NMT command specifier

1: Start Remote Node

1: Start Remote Node
2: Stop Remote Node
128: Enter Pre-operational State
129: Reset Node
130: Reset Communication

### Send SYNC

Use this function to synchronize tasks across the network.

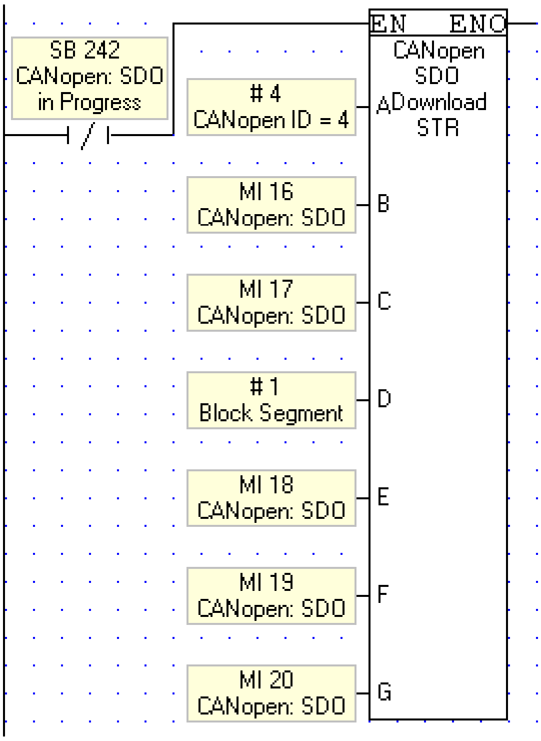


### SDO DownloadSTR / Upload STR

Use this function to:

- Download a segment block of data (bytes) **to** a node
- Upload a segment block of data (bytes) **from** a node.

**Note •** You must use SB 242 SDO in Progress as an activating condition.



| Parameter | Type | Function |
|---|---|---|
| Select CANopen: Remote ID | Constant | The CANbus ID number of the target PLC. |
| SDO Object index | Constant or MI | The index of the remote device. |
| SDO Object subindex | Constant or MI | The subindex of the remote device. |
| Select CANopen: SDO Mode | Constant | Select either:<br>• Domain Segment Transfer<br>• Block Segment Transfer |
| SDO Start of array | MI | The start address for the data array. |
| SDO Length of array | Constant or MI | The length of the data array<br>! This is the number of bytes in the **data** array. |
| Timeout (100 mS) | Constant or MI | This determines the length of the SDO timeout. |

### CANopen System Operands

Note • The CANopen System Operand addressing scheme changed as of VisiLogic version 7.00, OS 2.00 (Standard Division) OS 5.02 (Enhanced Division). If you are editing CANopen applications previous to this version, please refer to the topic Updating Project Versions.

To avoid losing messages, use Send conditions that you link to a an inverted contact of the appropriate SBs in your Ladder application to check whether there is room in the buffer.

| SB# | Description | Turns ON when: | Turns OFF when: | Reset by: |
|---|---|---|---|---|
| 284 | Send PDO1 | Message cannot be sent | Message can be sent | PLC |
| 285 | Send PDO2 | | | |
| 286 | Send PDO3 | | | |
| 287 | Send PDO4 | | | |
| 288 | RTR PDO1 | | | |
| 289 | RTR PDO2 | | | |
| 290 | RTR PDO3 | | | |
| 291 | RTR PDO4 | | | |
| 292 | Send NMT MC | | | |
| 293 | Send RTR NMT | | | |

| SB# | Description | Turns ON when: | Turns OFF when: | Reset by: |
|---|---|---|---|---|
| SB 240 | CANopen: Configuration downloaded | The CANopen Configuration FB is downloaded to the PLC | No CANopen Configuration is present | PLC |
| SB 241 | CANopen: Configured | CANopen Configuration was successful | CANopen Configuration failed | PLC |
| SB 242 | CANopen: SDO in Progress | SDO is busy transferring data | SDO is not in progress | PLC |
| SB 243 | CANopen: SDO transfer failed | SDO data transfer fails | SDO transfer begins | PLC |
| SB 284 | CANopen Buffer full: Send PDO1 | When SI 212 ≥ 8 | When SI 212 < 8 | PLC |
| SB 285 | CANopen Buffer full: Send PDO2 | When SI 213 ≥ 8 | When SI 213 < 8 | PLC |
| SB 286 | CANopen Buffer full: Send PDO3 | When SI 214 ≥ 8 | When SI 214 < 8 | PLC |
| SB 287 | CANopen Buffer full: Send PDO4 | When SI 215 ≥ 8 | When SI 215 < 8 | PLC |
| SB 288 | CANopen Buffer full: RTR PDO1 | When SI 216 ≥ 12 | When SI 216 < 12 | PLC |
| SB 289 | CANopen Buffer full: RTR PDO2 | When SI 217 ≥ 12 | When SI 217 < 12 | PLC |
| SB 290 | CANopen Buffer full: RTR PDO3 | When SI 218 ≥ 12 | When SI 218 < 12 | PLC |
| SB 291 | CANopen Buffer full: RTR PDO4 | When SI 219 ≥ 12 | When SI 219 < 12 | PLC |
| SB 292 | CANopen Buffer full: Send NMT MC | When SI 221 ≥ 8 | When SI 221 < 8 | PLC |
| SB 293 | CANopen Buffer full: Send RTR NM | When SI 222 ≥ 12 | When SI 222 < 12 | PLC |

| SI# | Description | Value | Comments |
|---|---|---|---|
| SI 211 | CANopen: Number of received messages | Shows the number of received messages in the Receive buffer (except for SDOs) | Maximum number of messages=128 |
| SI 212 | CANopen: # of Send PDO1 messages waiting to be sent | Shows the number of PDO1 messages currently in the PDO1 Send buffer | Maximum number of messages that can be queued in buffer=8 |
| SI 213 | CANopen: # of Send PDO2 messages waiting to be sent | Shows the number of PDO2 messages currently in the PDO2 Send buffer | |

| | | | |
|---|---|---|---|
| SI 214 | CANopen: #of Send PDO3 messages waiting to be sent | Shows the number of PDO3 messages currently in the PDO3 Send buffer | |
| SI 215 | CANopen: #of Send PDO4messages waiting to be sent | Shows the number of PDO4 messages currently in the PDO4 Send buffer | |
| SI 216 | CANopen: # of Send RTR PDO1messages waiting to be sent | Shows the number of RTR PDO1 messages currently in the PDO1 Send buffer | |
| SI 217 | CANopen: # of Send RTR PDO2 messages waiting to be sent | Shows the number of RTR PDO2 messages currently in the PDO2 Send buffer | |
| SI 218 | CANopen: # of Send RTR PDO3messages waiting to be sent | Shows the number of RTR PDO3 messages currently in the PDO3 Send buffer | |
| SI 219 | CANopen: #of Send RTR PDO4messages waiting to be sent | Shows the number of RTR PDO4 messages currently in the PDO4 Send buffer | |
| SI 221 | CANopen: # of Send NMTs messages waiting to be sent | Shows the number of NMT module control messages currently in the NMT Send buffer | |
| SI 222 | CANopen: # of Send RTR NMTs messages waiting to be sent | Shows the number of RTR NMT messages currently in the Send buffer | |
| SI 223 | CANopen: Send Buffer full (per type) | The bits in this register represent the different Send buffers (except for SDOs) | Maximum number of messages per buffer=8 |
| When a bit is ON, the corresponding buffer is full.<br>High byte:\| - \| - \| - \| - \| - \| - \| -\| NMT mc \|<br>Low byte: \| PDO \| -\| RCV NMT \| RCV PDO4\| RCV PDO3 \| RCV PDO2 \| RCV PDO1 \| RCV Emergency \| | | | |

| | | | |
|---|---|---|---|
| SI 224 | CANopen: Number of received SDO messages | Shows the number of received SDOs currently in the Receive buffer | |
| SI 225 | CANopen: SDO status | The status codes are given below. | |

| Value | Message |
|---|---|
| 0 | No error |
| 1 | PLC in STOP mode |
| 2 | CANopen not configured, SB 241 is not set (after configuration) |
| 3 | Remote ID is 0 |
| 4 | Maximum SDO Upload length set to 0 |
| 5 | SDO in Progress; Download/Upload started while SB242 is ON |
| 6 | SDO in Progress Error; SB242 turned OFF during data transfer (system problem) |
| 7 | Illegal Operands used in SDO data transfer |
| 8 | Number of operands in data type exceeded |
| 9 | Process buffer not cleared before SDO Send (system problem) |
| 10 | Response Timeout exceeded |
| 11 | Receive Error |
| 12 | Reserved by CIA |
| 13 | Receive Buffer full; more than 127 segments in a block (system problem) |
| 14 | Receive Error Toggle bit ON (error in domain segment) |
| 15 | Receive domain segment Abort; error code given in SDW 34 |
| 16 | Byte number error |
| 17 | Number of bytes is zero |
| 18 | Number of bytes exceeds the maximum upload length |
| 19 | Machine State error (system problem) |
| 20 | Receive Error in block size transferred from the remote device |
| 21 | Send Timeout exceeded |

| | | |
|---|---|---|
| | 22 | Sequence error in the number of segments in block transfer |
| | 23 | CRC error, block transfer |

The type of CANbus (UNICAN or CANopen) that is selected in the COM Init function determines the function of the following SDWs.

| SDW# | Description | Value | Comments |
|---|---|---|---|
| SDW 7 | CANopen: Number of Send Time messages lost | Number of times CANopen Send Time command is lost | Incremented when CANopen Send Time command is lost |
| SDW 8 | CANopen: Number of failed Sync attempts | Number of times that send SYNC failed | |
| SDW 24 Standard Vision | CANopen: Number of Send Time messages lost | Number of times CANopen Send Time command is lost | Incremented when CANopen Send Time command is lost |
| SDW 24 Enhanced Vision | UniCAN: Bit mask | Bit mask:<br>0x00000100 – High Priority Messages FIFO full<br>0x00000200 – Low Priority Messages FIFO full<br>0x00010000 – Rx FIFO full<br>0x80000000 – Bus off | |
| SDW 29 | CANopen: Bus is OFF Counter | Number of times bus was OFF | |
| SDW 33 | CANopen: SDO Number of Bytes | SDO upload: number of bytes received<br>SDO download: number of bytes sent | |
| SDW 34 | CANopen: Abort Code in SDO Abort | | |

| | Value | Message |
|---|---|---|
| | 0503 0000h | Toggle bit not alternated |
| | 0504 0000h | SDO protocol timed out |
| | 0504 0001h | Client/server command specifier not valid or unknown |
| | 0504 0002h | Invalid block size (block mode only) |
| | 0504 0003h | Invalid sequence number (block mode only) |
| | 0504 0004h | CRC error (block mode only) |
| | 0504 0005h | Out of memory |
| | 0601 0000h | Unsupported access to an object |
| | 0601 0001h | Attempt to read a write only object |
| | 0601 0002h | Attempt to write a read only object |
| | 0602 0000h | Object does not exist in the object dictionary |
| | 0604 0041h | Object cannot be mapped to the PDO |
| | 0604 0042h | The number and length of the objects to be mapped would exceed PDO length |
| | 0604 0043h | General parameter incompatibility reason |
| | 0604 0047h | General internal incompatibility in the device |
| | 0606 0000h | Access failed due to a hardware error |
| | 0607 0010h | Data type does not match, length of service parameter does not match |
| | 0607 0012h | Data type does not match, length of service parameter too high |
| | 0607 0013h | Data type does not match, length of service parameter too low |
| | 0609 0011h | Sub-index does not exist |
| | 0609 0030h | Invalid value for parameter (upload only) |
| | 0609 0031h | Value of parameter written too high (upload only) |
| | 0609 0032h | Value of parameter written too low (upload only) |
| | 0609 0036h | Maximum value is less than minimum value |
| | 060A 0023h | Resource not available: SDO connection |

| | | |
|---|---|---|
| | 0800 0000h | General error |
| | 0800 0020h | Data cannot be transferred or stored to the application |
| | 0800 0021h | Data cannot be transferred or stored to the application because of local control |
| | 0800 0022h | Data cannot be transferred or stored to the application because of the present device state |
| | 0800 0023h | Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error) |
| | 0800 0024h | No data available |

| | | | |
|---|---|---|---|
| SDW 36 | CANopen: Bus OFF error | | |

| | Value | Message |
|---|---|---|
| | 0 | No error |
| | 1 | Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed |
| | 2 | Form Error: Wrong format in fixed format part of a received frame |
| | 3 | AckError: The message this CAN controller transmitted was not acknowledged by another node |
| | 4 | Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (1), but the monitored bus value was dominant |
| | 5 | During busoff recovery this is set each time a sequence of 11 recessive bits is monitored. This enables the CPU to monitor the proceeding of the busoff recovery sequence (indicates that the bus is not stuck at dominant or continously disturbed |
| | 6 | CRCError: The received CRC check sum is incorrect |
| | 7 | Unused code: may be written by the CPU to check for updates |

| | | | |
|---|---|---|---|
| SDW 56 | CANopen: PDO Send Counter | | |
| SDW 57 | CANopen: NMT/SDO Send Counter | | |

## CANbus, Layer 2

Vision controllers support data communication with remote devices such as frequency converters according to the CANbus V2.0 standard.

A Vision can both send and receive standard messages with 11-bit identifiers, as well as extended messages with 29-bit identifiers.

**Note**

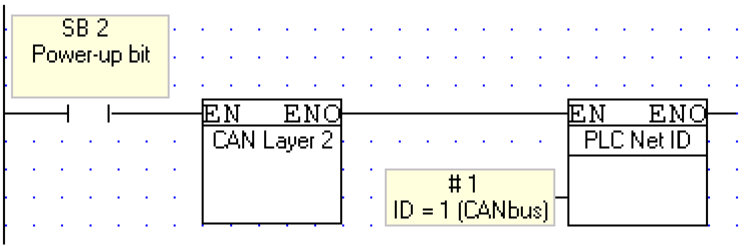
- Enhanced Vision only: note that you can run both CANbus Layer 2 and CANopen if you select the relevant Com Port option in the COM Init function.
- Note that you can initialize the port to Scan or to Scan_Ex.
  -Scan: PLC can receive messages from a specific device via a specific port. You can configure up to 4 ports, and receive up to 4 messages per scan
  -Scan_Ex: PLC can receive messages from any device, and not via a specified port. Only one message may be received per scan.
- Prevent CANbus network overload by using a Send condition to activate functions. Do not place them directly on the left Ladder rail.

### Using CANbus Layer 2

At power-up:

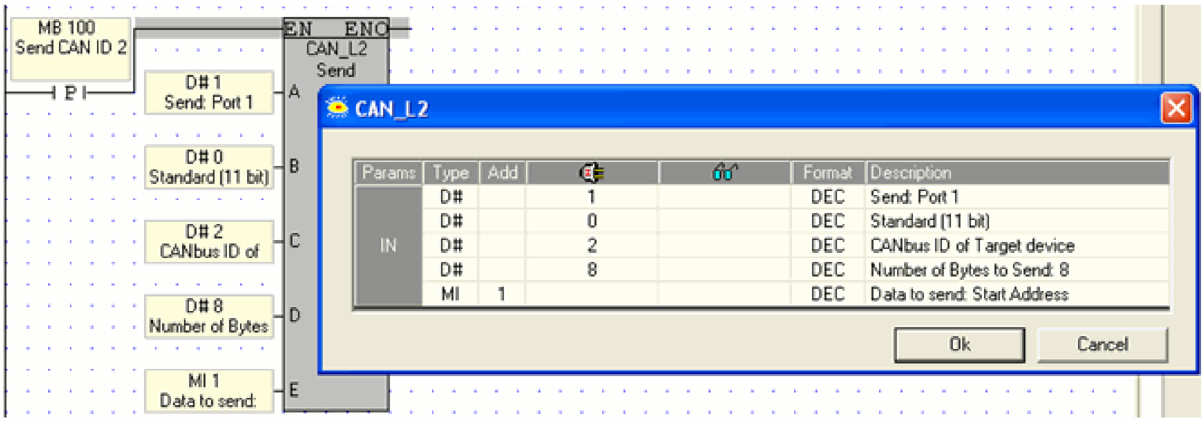
1. Assign each networked controller a unique Unit ID number.

2. Initialize the CANbus port to CANbus Layer2 via a COM Init function.
3. Activate the configuration.



### Send

The Send function determines the data that is sent to a specified remote device.

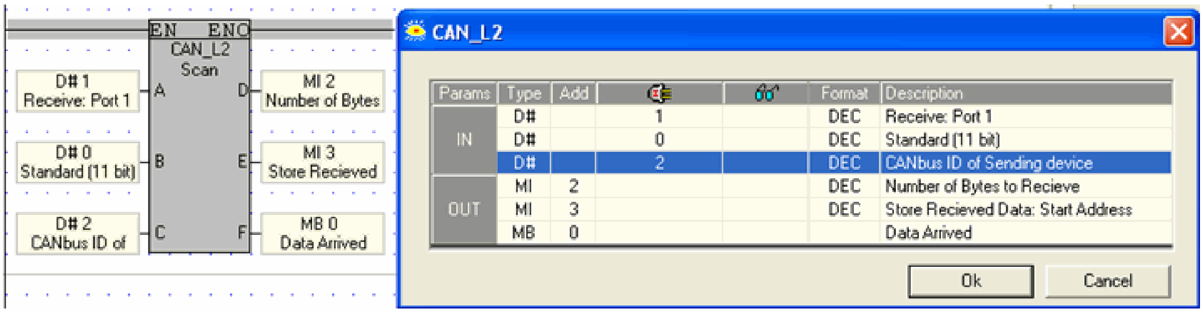


| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 1 | | DEC | Send: Port 1 |
| | D# | | 0 | | DEC | Standard (11 bit) |
| | D# | | 2 | | DEC | CANbus ID of Target device |
| | D# | | 8 | | DEC | Number of Bytes to Send: 8 |
| | MI | 1 | | | DEC | Data to send: Start Address |

| Parameter Name | Purpose |
|---|---|
| Send Port | Select one of the four available ports. |
| Identifier | Select either the 11-bit (default), or the extended 29-bit identifier. |
| CANbus ID of Target Device | Enter the number of the target device either by:<br>-Directly entering it<br>- Select the address of the MI containing the ID number. |
| Number of Bytes to Send | A CANbus message can contain up to 8 data bytes. |
| Data to Send: Start Address | Enter the start address of the vector containing the data. |

### Scan

The Scan function enables a PLC to receive data from a specified remote device and write it to a vector of registers.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | D# | | 1 | | DEC | Receive: Port 1 |
| | D# | | 0 | | DEC | Standard (11 bit) |
| | D# | | 2 | | DEC | CANbus ID of Sending device |
| OUT | MI | 2 | | | DEC | Number of Bytes to Recieve |
| | MI | 3 | | | DEC | Store Recieved Data: Start Address |
| | MB | 0 | | | | Data Arrived |

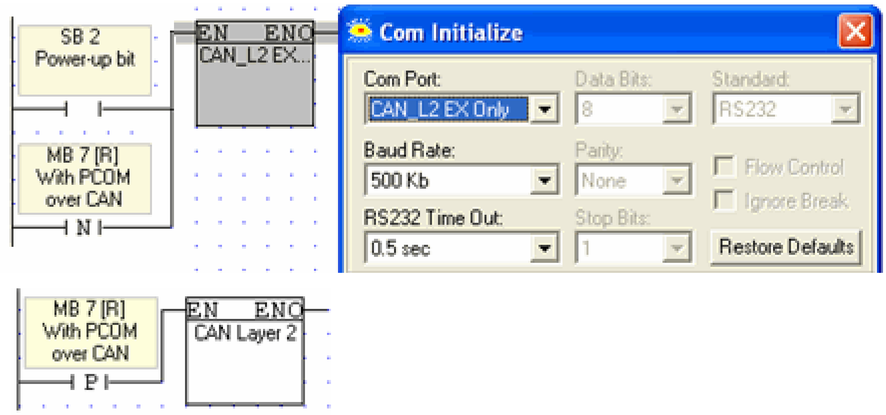
| **Parameter Name** | **Purpose** |
|---|---|
| Receive Port | Select one of the four available ports. |
| Identifier | Select either the 11-bit (default), or the extended 29-bit identifier. |
| CANbus ID of Sending Device | Enter the number of the target device either by:<br>-Directly entering it<br>- Select the address of the MI containing the ID number. |
| Number of Bytes to Receive | A CANbus message can contain up to 8 data bytes. |
| Store Received Data: Start Address | Enter the start address of the vector to contain the data. |
| Data Arrived | Turns ON when the message is received.<br>**Note** • This MB must be reset by the user. |

### Scan_EX

This function enables the controller to receive any messages that arrive via CANbus.

In order to run Scan_Ex, you must initialize the port to Scan_Ex.

Note that Scan_EX is active, you cannot access the PLC via network bridge. If your application requires communication via bridge, you must re-initialize the port using another COM Init function as shown below.



Com Initialize

Com Port: CAN_L2 EX Only

Data Bits: 8

Standard: RS232

Baud Rate: 500 Kb

Parity: None

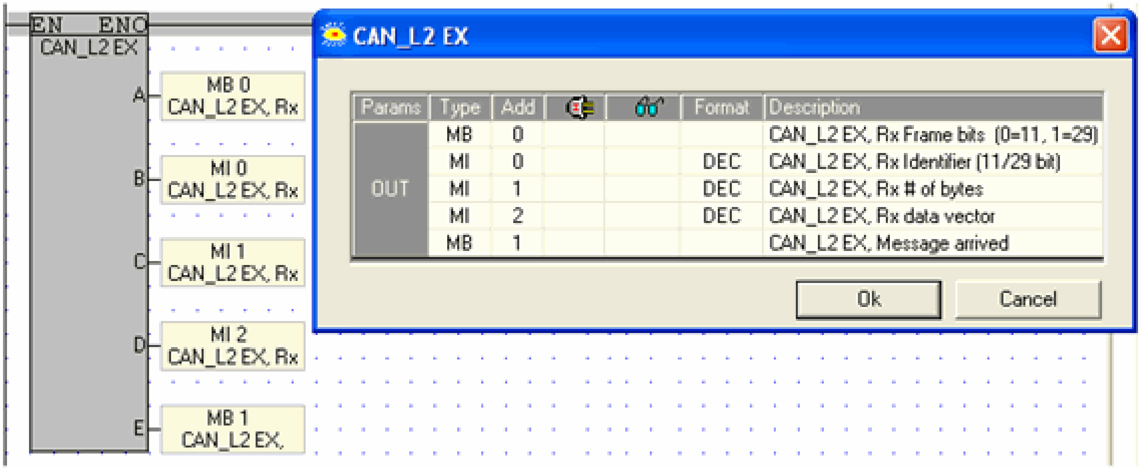
□ Flow Control

□ Ignore Break

RS232 Time Out: 0.5 sec

Stop Bits: 1

.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| OUT | MB | 0 | | | | CAN_L2 EX, Rx Frame bits (0=11, 1=29) |
| | MI | 0 | | | DEC | CAN_L2 EX, Rx Identifier (11/29 bit) |
| | MI | 1 | | | DEC | CAN_L2 EX, Rx # of bytes |
| | MI | 2 | | | DEC | CAN_L2 EX, Rx data vector |
| | MB | 1 | | | | CAN_L2 EX, Message arrived |

| Parameter Name | Purpose |
|---|---|
| RX Frame bits | After the message is received:<br>-if the status of the bit operand =0 the received message contains an 11-bit identifier,<br>-if the status of the bit operand =1 the received message contains a 29-bit identifier, |
| RX Identifier | This operand contains the received message identifier. Note that if the identifier is 29-bit, you must use a DW. |
| RX Number of bytes | This operand contains the number of bytes to be received in the message. A CANbus message can contain up to 8 data bytes. |
| Rx Data Vector | Enter the start address of the vector to contain the data. |
| Message Arrived | Turns ON when the message is received.<br>**Note** • Must be reset by the user. |

**Note •** CANopen COB ID numbers are 11-bit. If your CANbus Layer2 application uses 11-bit identifiers, make sure that the ID/identifiers are unique. If a message arrive through the bus marked with the same number, CANbus Layer 2 will scan the message but CANopen will not.

## CANbus ISC, via Network Operands

CANbus ISC enables inter-PLC data exchange. Via the CANbus port, you can:

- Network up to **64** controllers, where each unit can read information from up to **8** other controllers in the network, including M90/91 PLCs that support CANbus.
- Connect one PC to 63 controllers.

When PLCs are connected to the CANbus network and their CANbus ports initialized to CANbus ISC, each networked PLC broadcasts specific data to the network, stamped with the controller's unique ID number. This data is contained in the following system operands:16 System Bits (SB 200 to SB 215) 16 Inputs (I 0 to I 15), and 2 System Integers (SI 200 and SI 201).

When the data in these operands is constant (unchanging), it is broadcast every 400mS. When the data is dynamic (changing) it is broadcast at a rate not exceeding 20mS.  Even when the data changes are rapid, data is not broadcast at a rate exceeding 20mS; this avoids network overload.

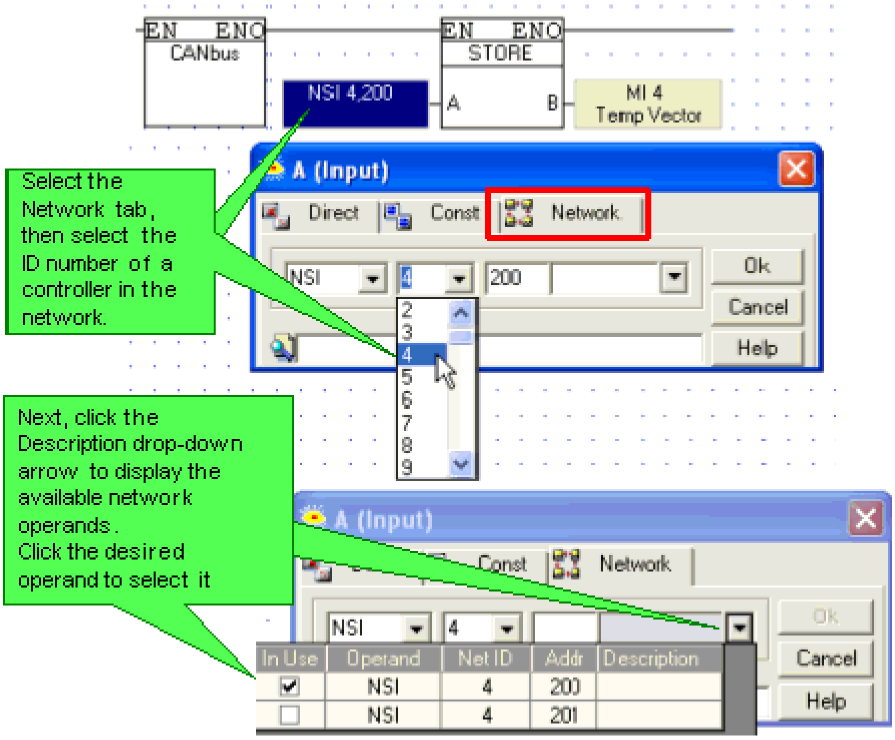
### Implementing inter-PLC ISC data exchange

First, assign each networked controller a unique Unit ID number and initialize the CANbus port to CANbus ISC.

Reading Data from a Networked Controller via Network Operands

When a controller is integrated into a CANbus network, the values in these operands are continually broadcast to the network: SB 200 to SB 215, I 0 to I 15, SI 200 and SI 201.

In order to enable a networked controller to read the values from another networked controller, place the desired function in the net. In the Select Operand Address box, click on the Network tab, then select the ID of the target controller and the desired operand.



Select the Network tab, then select the ID number of a controller in the network.

Next, click the Description drop-down arrow to display the available network operands. Click the desired operand to select it

| In Use | Operand | Net ID | Addr | Description |
|---|---|---|---|---|
| ☑ | NSI | 4 | 200 | |
| □ | NSI | 4 | 201 | |

## SB 237: Enable/Disable Broadcast

By default, SB 237 is set to OFF, enabling the controller to broadcast data. When this SB is ON, the data is not broadcast. This can be used prevent network overload.

Errors are indicated by SB 236; the error is indicated in SI 237.

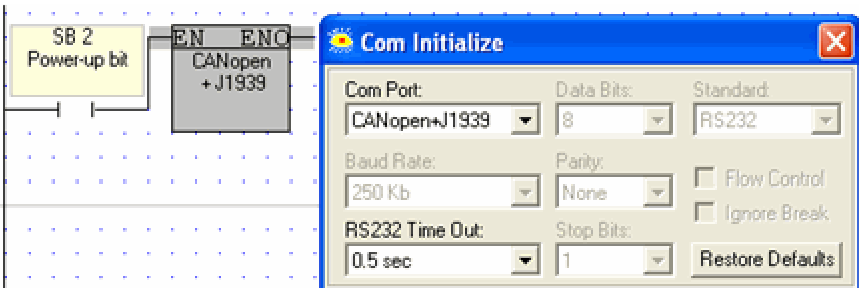
# SAE J1939

SAE J1939 is a CANbus protocol that is supported by Enhanced Division Vision OPLC™ models. Within the J1939 network, the Vision functions as an ECU (Electronic Control Unit). Note that Vision supports transport protocols such as BAM and RTS/CTS.

Depending on the Vision model, the CANbus port may be an integral part of the PLC, or may be purchased and installed separately.

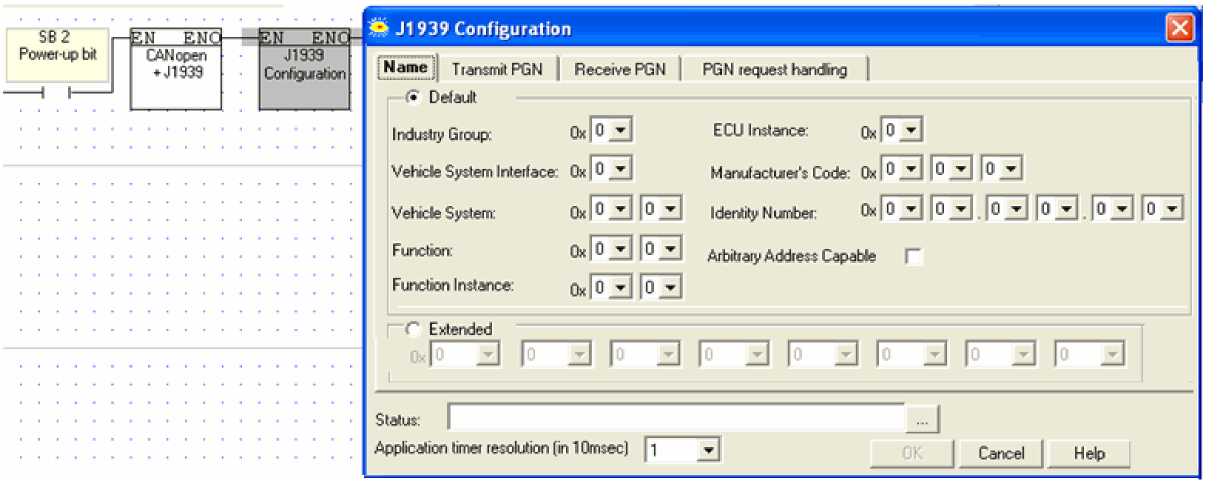
### Configuring J1939

1. Configure the CANbus port using a COM Init function set to CANopen + J1939. Note that the ECU address used by the J1939 protocol is the Vision Unit ID number, which is contained in SI 8 when COM Init runs.



Com Port: CANopen+J1939
Data Bits: 8
Standard: RS232
Baud Rate: 250 Kb
Parity: None
□ Flow Control
□ Ignore Break
RS232 Time Out: 0.5 sec
Stop Bits: 1

2. After the Com Init function, place the J1939 Configuration function.



● Default
Industry Group: 0x 0
Vehicle System Interface: 0x 0
Vehicle System: 0x 0 0
Function: 0x 0 0
Function Instance: 0x 0 0
ECU Instance: 0x 0
Manufacturer's Code: 0x 0 0 0
Identity Number: 0x 0 0 . 0 0 . 0 0
Arbitrary Address Capable □
○ Extended
0x 0 0 0 0 0 0 0 0
Status:
Application timer resolution (in 10msec) 1

The J1939 Configuration has three tabs comprising the parameters listed below. Note that the Status parameter, J1939 Configuration Startup process status, provides status messages for all tabs in the Configuration.

Status messages J1939 Configuration Startup process (DW)

J1939 Protocol Errors

| Byte 1 (MSB) | |
|---|---|
| 0 | No error |
| FF | Error exists |

**Byte 2: Error Location**

| | |
|---|---|
| 0 | Error unit |
| 1 | Data link layer unit |
| 2 | Transport unit |
| 3 | Networking unit |
| 4 | Application unit |
| 5 | Cyclic unit |
| 6 | Memory unit |

**Byte 3: The Error Classes (Priorities)**

| | |
|---|---|
| 0 | Warning, Application OK |
| 1 | Error occurred |
| 2 | Fatal Error |

**Byte 4: The error for each unit**

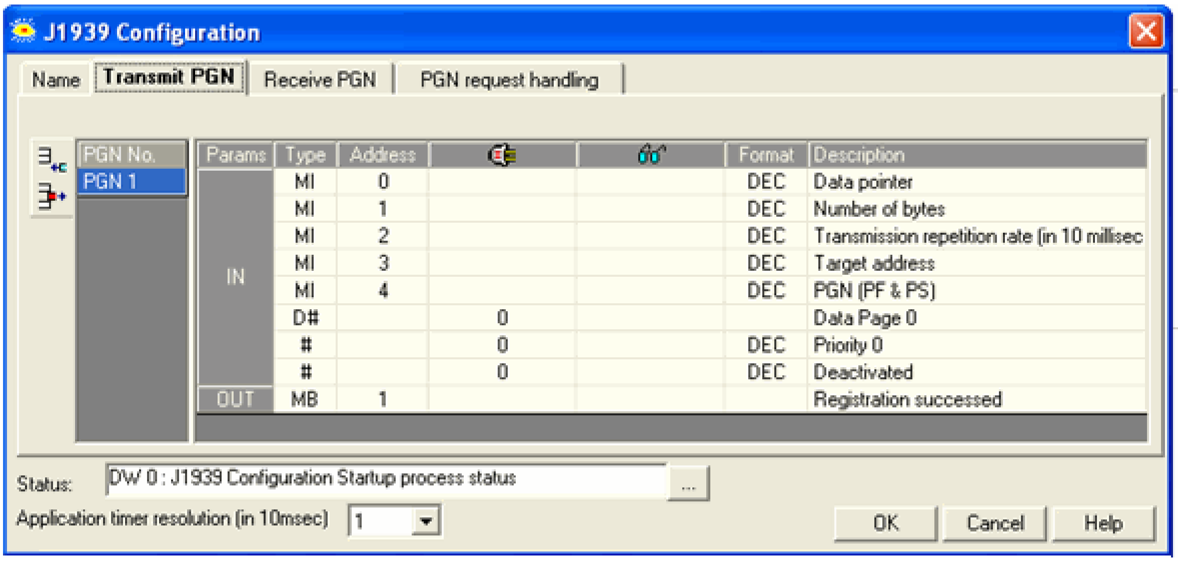
| | | | |
|---|---|---|---|
| Data link layer unit | Synchronous Errors | 1 | overrun in the Tx Queue |
| | | 2 | starting CAN failed |
| | | 3 | reset CAN failed |
| | | 4 | init CAN failed |
| | Asynchronous Errors | 11 | bus error interrupt occurs on CAN |
| | | 12 | error warning interrupt occurs on CAN |
| | | 13 | CAN controller is in status error active |
| | | 14 | data overrun interrupt occurs on CAN |
| | | 15 | overrun in the Rx Queue |
| Transport unit | Synchronous Errors | 1 | No free Entry in the Tx Queue of the TPL available |
| | | 2 | Too many PGNs tried to register in the config filter |
| | Asynchronous Errors | 11 | Unexpected BAM frame received |
| | | 12 | Unexpected RTS frame received |
| | | 13 | Unexpected CTS frame received |
| | | 14 | Unexpected EOM frame received |
| | | 15 | Unexpected CA frame received |
| | | 16 | Unexpected DT frame received |
| | | 17 | No free Rx Msg Buffer in the TPL for a global message available |
| | | 18 | Timer overrun for Tx Timeout T0 |
| | | 19 | Timer overrun for Tx Timeout T1 |
| | | 20 | Timer overrun for Tx Timeout T2 |
| | | 21 | Timer overrun for Rx Timeout T0 |
| | | 22 | Timer overrun for Rx Timeout T1 |
| | | 23 | Timer overrun for Rx Timeout T2 |
| | | 24 | The length of the received message is too long for a segmented message |
| | | 25 | Error by Sending: Can't send a segmented message |
| | | 26 | Sending a CA message failed |
| | | 27 | Sending a NACK message failed |
| Networking unit | Asynchronous Errors | 11 | The device can't claim a free address |
| | | 12 | The address list is full |
| Application unit | Synchronous Errors | 1 | APL_Main() was not called in the required sequence |
| | | 2 | The data field is too long (could not sent message) |
| | | 3 | No memory for the data field allocated |
| | | 4 | An error occurred during registration of a request PGN |
| | Asynchronous Errors | 11 | A request message couldn't be answered from the request handler because there was no place in the queue |
| Cyclic unit | Synchronous Errors | 1 | The maximal number of receive messages is exceeded |
| | | 2 | The maximal number of transmit messages is exceeded |
| | Asynchronous Errors | 11 | A registered message wasn't received in the given time |
| Memory unit | Asynchronous Errors | 11 | Allocate memory failed |
| | | 12 | Free memory failed |

## Configuration Parameters

### Tab: Name

Note that the PC downloads 64 bits of data from this tab to the PLC.
'Name' is a 64 bit (8 bytes) long label which gives an ECU a unique identity on the J1939 network.

| | |
|---|---|
| Inputs | Arbitrary address capable – (1 bit) |
| | Industry group - (3 bit) |
| | Vehicle system interface – (4 bit) |
| | Vehicle system – (7 bit) |
| | Function - (8 bit) |
| | Function instance - (5 bit) |
| | ECU instance - (3 bit) |
| | Manufacture code - (11 bit) |
| | Identity number - (21 bit) |
| Extended | Selecting this enables you to create a 64-bit label according to your own requirements. |



| PGN No. |
|---|
| PGN 1 |

| Params | Type | Address | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | MI | 0 | | | DEC | Data pointer |
| | MI | 1 | | | DEC | Number of bytes |
| | MI | 2 | | | DEC | Transmission repetition rate (in 10 millisec |
| | MI | 3 | | | DEC | Target address |
| | MI | 4 | | | DEC | PGN (PF & PS) |
| | D# | | 0 | | | Data Page 0 |
| | # | | 0 | | DEC | Priority 0 |
| | # | | 0 | | DEC | Deactivated |
| OUT | MB | 1 | | | | Registration successed |

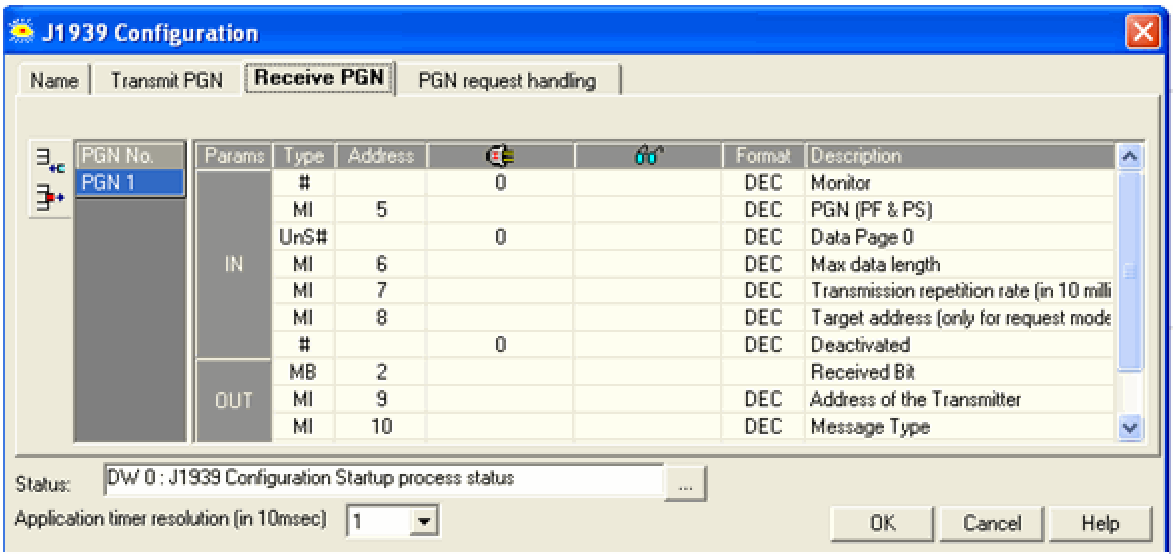
Status: DW 0 : J1939 Configuration Startup process status

Application timer resolution (in 10msec) 1

### Tab: Transmit PGN

You can define up to 32 Parameter Group Numbers (PGNs) (total data length 64 bytes) that will send a message according to the Transmission Repetition rate you set.

| | |
|---|---|
| **Inputs** | Data pointer<br>This is the start of the vector that contains the data that will be transmitted from the PLC. |
| | Number of bytes to send (16 bit) |
| | Transmission repetition rate: units of 10 milliseconds. This is the rate at which the data is sent. |
| | Target address. The message is sent to this address in the receiving device.<br>For a global address, the value 0xFF must be entered. (8 bit) |
| | PGN (PF & PS) operand (16 bit) |
| | Data Page (1 bit). |
| | Data Page Extension for the PGN (1 bit). |
| | Priority – operand (3 bit) |
| | Specifies if the Transmit PGN will be activated or deactivated after registration:<br>0 - Deactivated<br>1 – Activated<br>(1 bit)<br>Note that after power-up, Transmit PGN may be controlled with the Ladder function Activate Cyclic transmission of PGN. |
| **Output** | Registration succeeded |



J1939 Configuration

Name | Transmit PGN | **Receive PGN** | PGN request handling

| PGN No. |
|---|
| PGN 1 |

| Params | Type | Address | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | # | | 0 | | DEC | Monitor |
| | MI | 5 | | | DEC | PGN (PF & PS) |
| | UnS# | | 0 | | DEC | Data Page 0 |
| | MI | 6 | | | DEC | Max data length |
| | MI | 7 | | | DEC | Transmission repetition rate (in 10 milli |
| | MI | 8 | | | DEC | Target address (only for request mode |
| | # | | 0 | | DEC | Deactivated |
| OUT | MB | 2 | | | | Received Bit |
| | MI | 9 | | | DEC | Address of the Transmitter |
| | MI | 10 | | | DEC | Message Type |

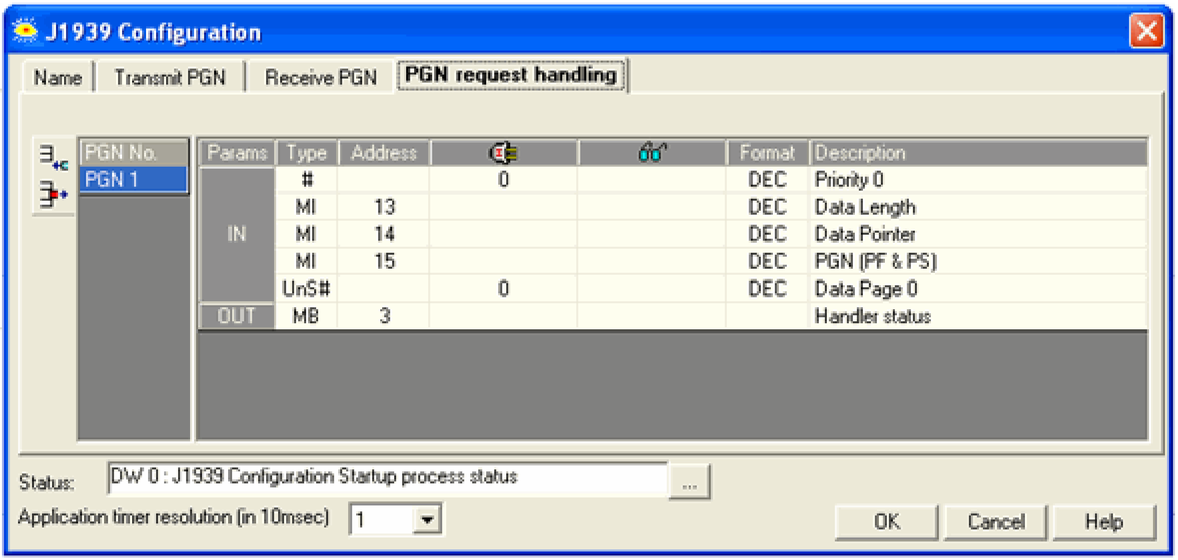
Status: DW 0 : J1939 Configuration Startup process status

Application timer resolution (in 10msec) 1

## Tab: Configure Receive PGN

You can define up to 32 Parameter Group Numbers (PGNs) (total data length 64 bytes) that may be received by the PLC.

| | |
|---|---|
| **Inputs** | Scan Method (0=Monitor 1=Request) |
| | PGN (PF & PS) – operand (16 bit) |
| | Data Page (1 bit). |
| | Max data length MI / XI – maximum number of bytes to receive (16 bit). |
| | Transmission repetition rate: units of 10 milliseconds. This is the rate at which the data is sent. |
| | Target address. The message is sent to this address in the receiving device.<br>For a global address, the value 0xFF must be entered. (8 bit) |
| | Specifies if the Receive PGN will be activated or deactivated after registration:<br>0 - Deactivated<br>1 – Activated<br>Note that after power-up, Receive PGN may be controlled with the Ladder function Transmission for Request. |
| **Outputs** | Received bit |
| | Address of the Transmitter (the device sending the data) (8 bit) |
| | Message type (global or specific) (8 bit) |
| | The number of data bytes received (16 bit) |
| | Pointer to the data-array of the message |



J1939 Configuration

Name | Transmit PGN | Receive PGN | **PGN request handling**

| PGN No. |
|---|
| PGN 1 |

| Params | Type | Address | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | # | | 0 | | DEC | Priority 0 |
| | MI | 13 | | | DEC | Data Length |
| | MI | 14 | | | DEC | Data Pointer |
| | MI | 15 | | | DEC | PGN (PF & PS) |
| | UnS# | | 0 | | DEC | Data Page 0 |
| OUT | MB | 3 | | | | Handler status |

Status: DW 0 : J1939 Configuration Startup process status

Application timer resolution (in 10msec) 1

### Tab: PGN request handling

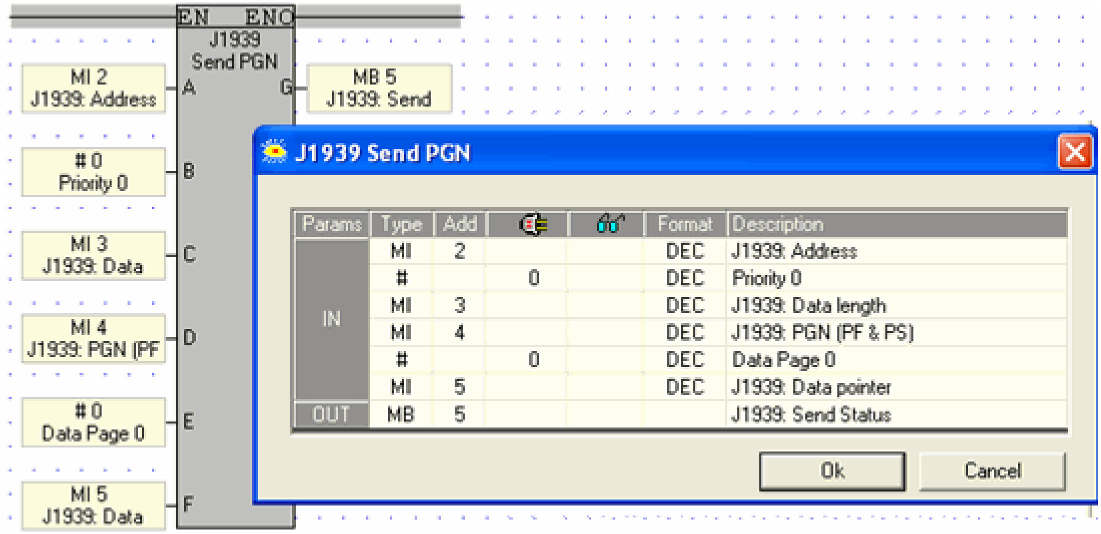
This determines if the PLC will answer PGN requests, and if so, what data the PLC will send.

| | |
|---|---|
| **Inputs** | Priority – operand (3bit) |
| | Data length – number of bytes to send (16 bit) |
| | Data pointer – Pointer to the buffer with the data of this PGN (byte *). |
| | PGN (PF & PS) – operand (16 bit) |
| | Data Page (1 bit). |
| | Extension Data Page Extension for the PGN (1 bit). |
| **Outputs** | Received bit |

### Ladder Functions

You can use the ladder functions to send and receive messages in the Configuration. If, for example, you have deactivated Transmit in the Configuration, you can send a PGN using these functions.

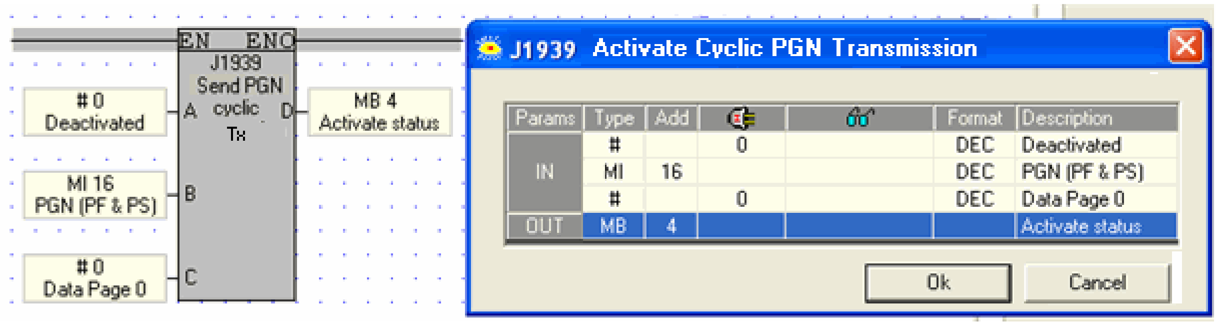
#### Send PGN



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | MI | 2 | | | DEC | J1939: Address |
| | # | | 0 | | DEC | Priority 0 |
| | MI | 3 | | | DEC | J1939: Data length |
| | MI | 4 | | | DEC | J1939: PGN (PF & PS) |
| | # | | 0 | | DEC | Data Page 0 |
| | MI | 5 | | | DEC | J1939: Data pointer |
| OUT | MB | 5 | | | | J1939: Send Status |

#### Activate Cyclic PGN Transmission

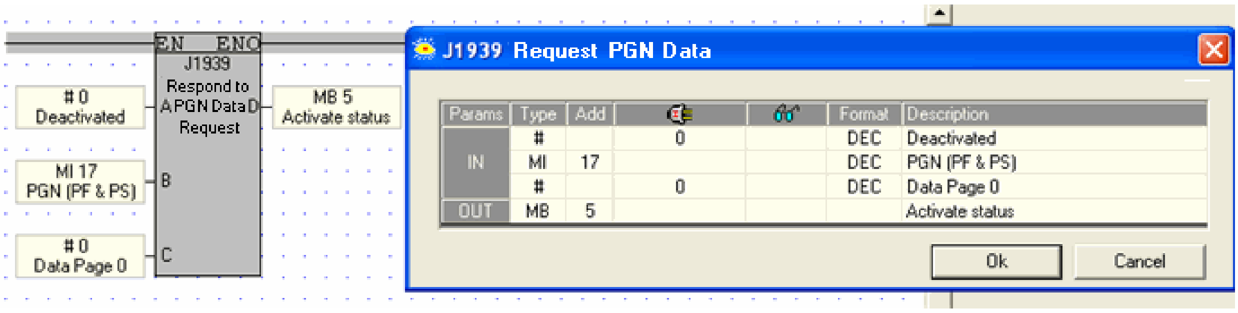
If the Activate/Deactivate bit is OFF in the Configuration>Transmit PGN tab, use this function to activate it in the next scan.



| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | # | | 0 | | DEC | Deactivated |
| | MI | 16 | | | DEC | PGN (PF & PS) |
| | # | | 0 | | DEC | Data Page 0 |
| OUT | MB | 4 | | | | Activate status |

#### Request PGN

Use this to request a specific PGN message from another ECU.



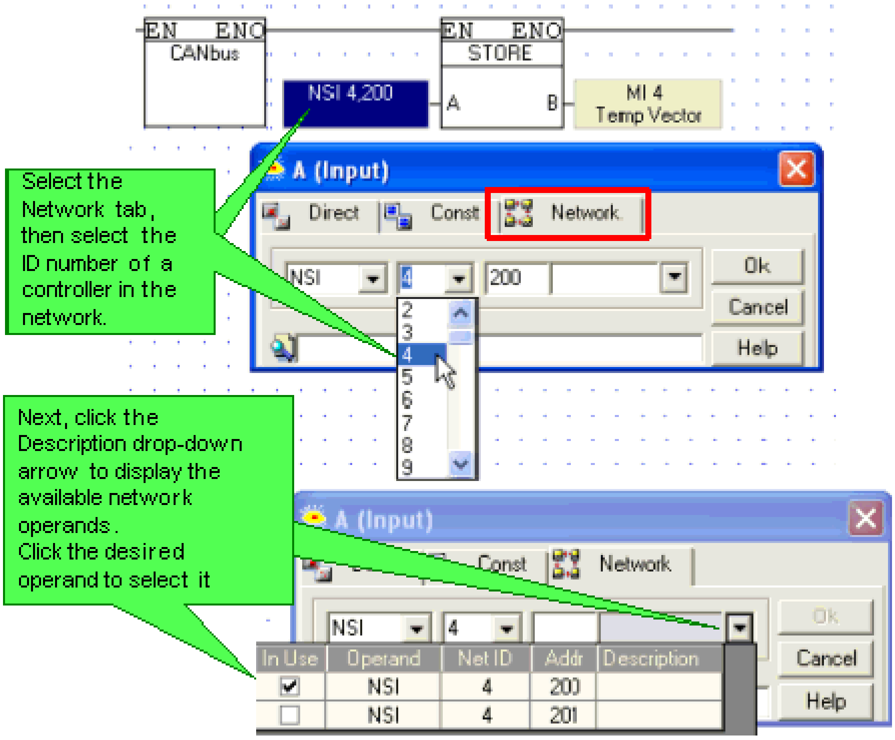
J1939 Request PGN Data

| Params | Type | Add | | | Format | Description |
|---|---|---|---|---|---|---|
| IN | # | | 0 | | DEC | Deactivated |
| | MI | 17 | | | DEC | PGN (PF & PS) |
| | # | | 0 | | DEC | Data Page 0 |
| OUT | MB | 5 | | | | Activate status |

## Network Operands-Communicating Data Via CANbus ISC

When a controller is integrated into a CANbus ISC network, the data contained in certain system operands is continuously broadcast to the network, together with the controller's unique ID number. The data is contained in 16 System Bits (SB 200 to SB 215 (16 Inputs (I 0 to I 15), and 2 System Integers (SI 200 and SI 201).

In order to enable a networked controller to read the values from another networked controller, place the desired function in the net. In the Select Operand Address box, click on the Network tab, then select the ID of the target controller and the desired operand.



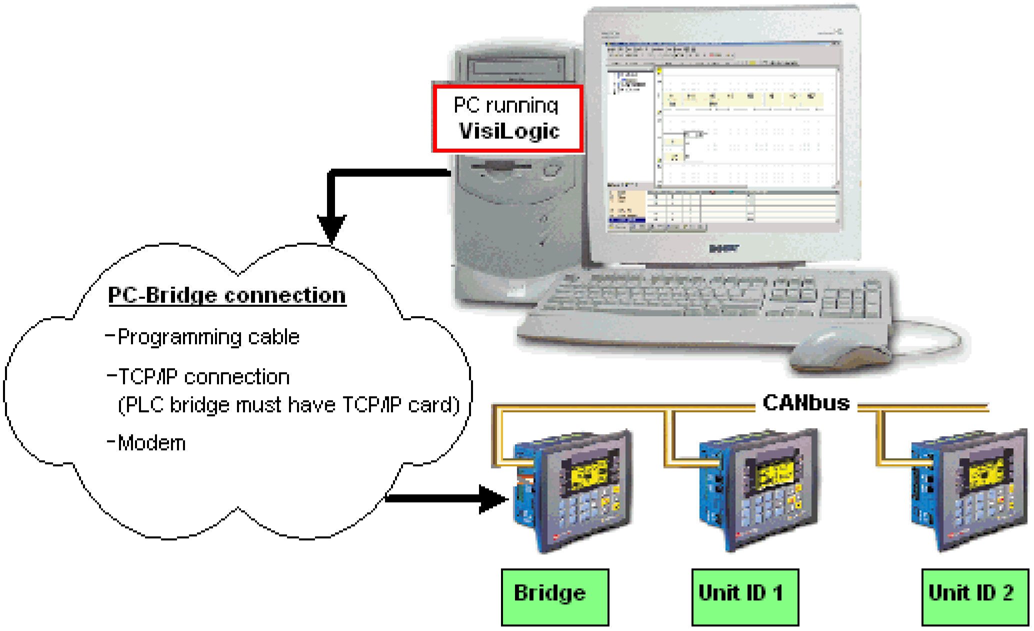
Select the Network tab, then select the ID number of a controller in the network.

Next, click the Description drop-down arrow to display the available network operands.
Click the desired operand to select it

| In Use | Operand | Net ID | Addr | Description |
|---|---|---|---|---|
| ☑ | NSI | 4 | 200 | |
| □ | NSI | 4 | 201 | |

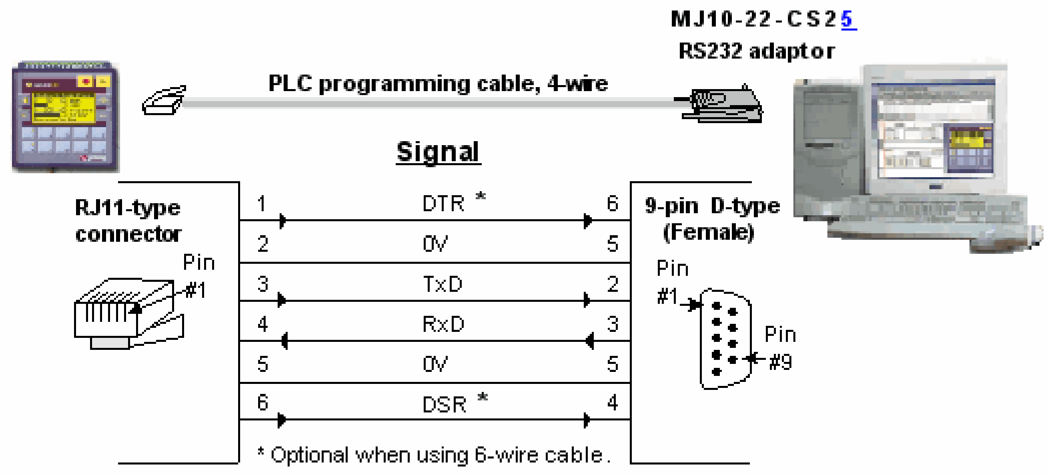
## Accessing a Networked PLC via PC

Use a PC to access controllers within a network either via a direct connection, TCP/IP connection, GSM or landline modem.



### Accessing a Networked Controller

1. Connect your PC to any controller in the network using the programming cable supplied with the controller kit,
   or via TCP/IP.



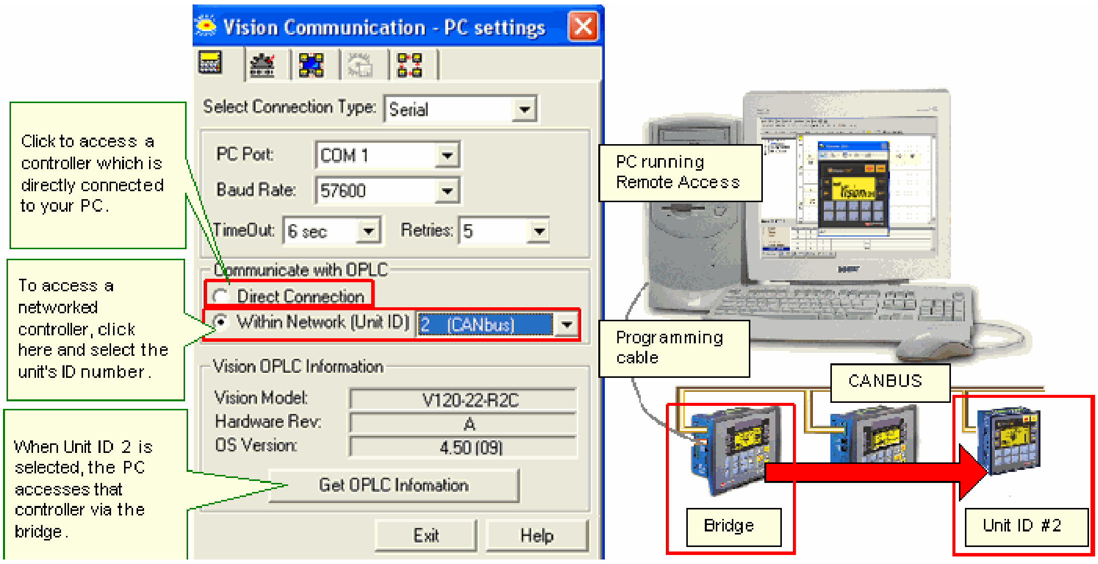
| RJ11 | Signal | 9-pin |
|---|---|---|
| 1 | DTR * | 6 |
| 2 | 0V | 5 |
| 3 | TxD | 2 |
| 4 | RxD | 3 |
| 5 | 0V | 5 |
| 6 | DSR * | 4 |

* Optional when using 6-wire cable.

**Note •** Different PCs can access a network at the same time, using different controller units as bridges. However, 2 different PCs cannot simultaneously access the same controller unit.

• The PC accesses the bridge via a serial port. Vision controllers comprise three communication buffers. Note that if all three buffers are busy processing communications, new requests are ignored until a buffer is free.

2. Select a networked controller by opening Communication & OS from the Connection menu, and then entering the Unit ID number.



Vision Communication - PC settings

Select Connection Type: Serial

PC Port: COM 1

Baud Rate: 57600

TimeOut: 6 sec Retries: 5

Communicate with OPLC

○ Direct Connection

● Within Network (Unit ID) 2 (CANbus)

Vision OPLC Information

Vision Model: V120-22-R2C

Hardware Rev: A

OS Version: 4.50 (09)

Get OPLC Infomation

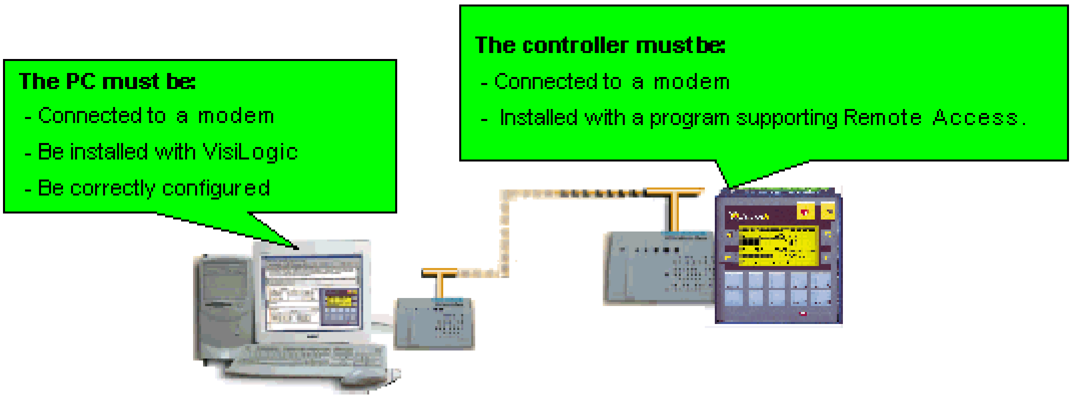
Exit Help

Click to access a controller which is directly connected to your PC.

To access a networked controller, click here and select the unit's ID number.

When Unit ID 2 is selected, the PC accesses that controller via the bridge.

## Accessing a Networked Controller via Modem



**The PC must be:**
- Connected to a modem
- Be installed with VisiLogic
- Be correctly configured

**The controller must be:**
- Connected to a modem
- Installed with a program supporting Remote Access.

| | |
|---|---|
| **Note** • | The PC-modem cable is not the same type of cable used to connect between the controller and the modem. Ensure that the cable used to connect the PC to the modem provides connection points for all of the modem's pins. |
| • | In order to ensure successful operations, it is recommended that you use an external PC modem. Internal modems may not support communications. |
| • | Both PC and controller must use the same type of modem: either landline or GSM. Internal PC modems must be used in conjunction with the driver provided by the modem's manufacturer. |
| • | If calls are routed via a switchboard, note that the switchboard settings may interfere with communications. Consult with your switchboard provider. |

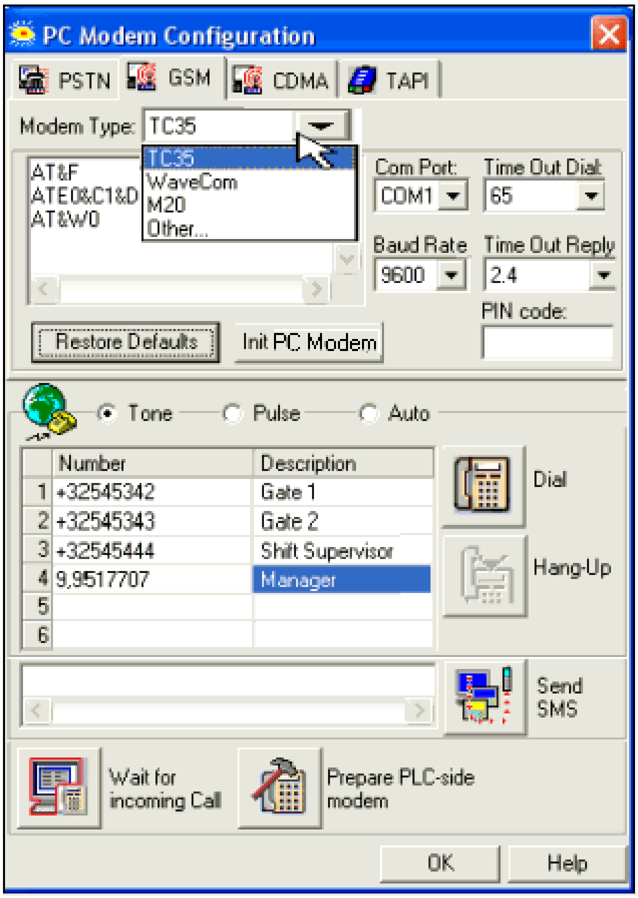
### Modems: Setting Up

PLC-Modem Connection

The Unitronics' cable provided with modem kits does not provide a standard connection. This connection is adapted to support the fact that Unitronics controllers do not support the control lines.  The cable shorts the DSR and the DTR together, which ensures that the terminal is always ready to receive data. For more information, refer to the topic Modem Connection and Pinouts.

PC Modem Configuration

Open PC Modem Configuration from the Connection>Modem Services menu.



Modem Type: TC35

TC35
WaveCom
M20
Other...

AT&F
ATE0&C1&D
AT&W0

Com Port: COM1

Time Out Dial: 65

Baud Rate: 9600

Time Out Reply: 2.4

PIN code:

● Tone ○ Pulse ○ Auto

| | Number | Description |
|---|---|---|
| 1 | +32545342 | Gate 1 |
| 2 | +32545343 | Gate 2 |
| 3 | +32545444 | Shift Supervisor |
| 4 | 9,9517707 | Manager |
| 5 | | |
| 6 | | |

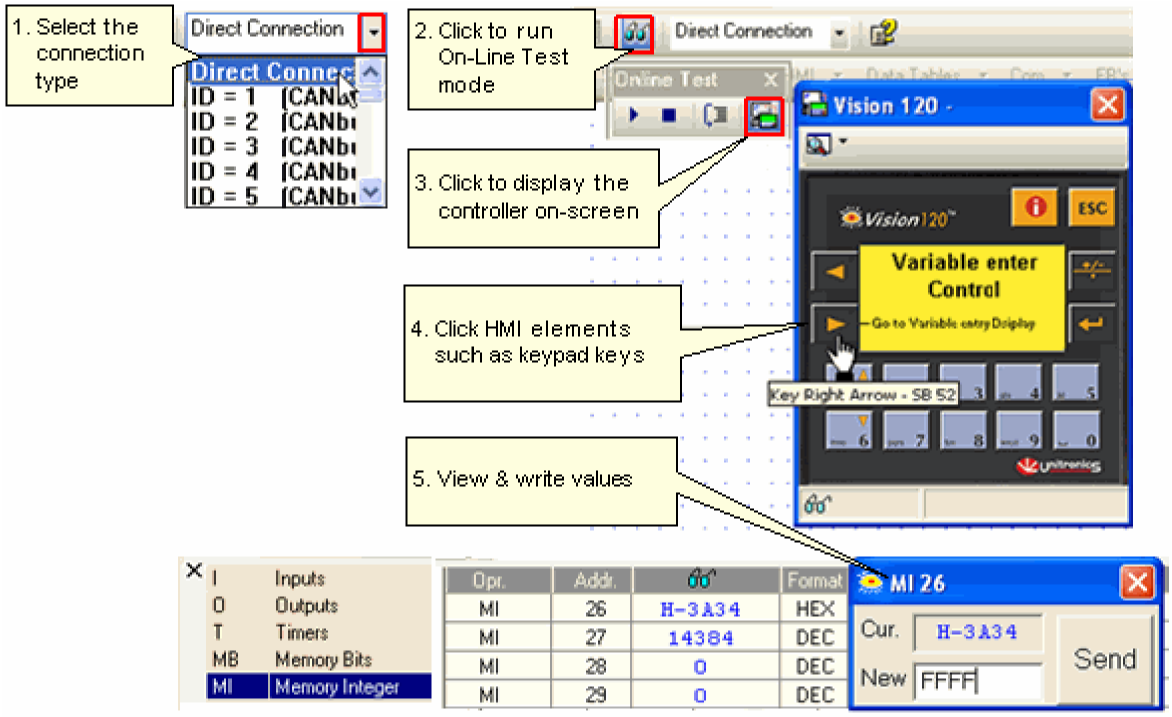
**Note** • If, within the modem initialization strings, the parameter S7 TimeOut, is to short to permit the PLC's modem to answer, an error will result.

For example, if this parameter is set as S7=30, the PC modem will wait for 3 seconds to receive an answer from the PLC's modem. If, however, the PLC program's COM Init FB Answer Settings are set to 'Answer after 6 rings,' the PLC modem will not be able to answer before the 3 seconds have elapsed. In this case, the TimeOut parameter is exceeded, and the PC modem will return the No Carrier error.

## Online Test Mode & Remote Access

You can run Online Test mode and use Remote Access session to display the networked controller on your PC screen.

To do this, click the Online Test button on the toolbar, then click the Remote Access button.



1. Select the connection type

2. Click to run On-Line Test mode

3. Click to display the controller on-screen

4. Click HMI elements such as keypad keys

5. View & write values

| Opr. | Addr. | | Format |
|---|---|---|---|
| MI | 26 | H-3A34 | HEX |
| MI | 27 | 14384 | DEC |
| MI | 28 | 0 | DEC |
| MI | 29 | 0 | DEC |

Cur. H-3A34

New FFFF

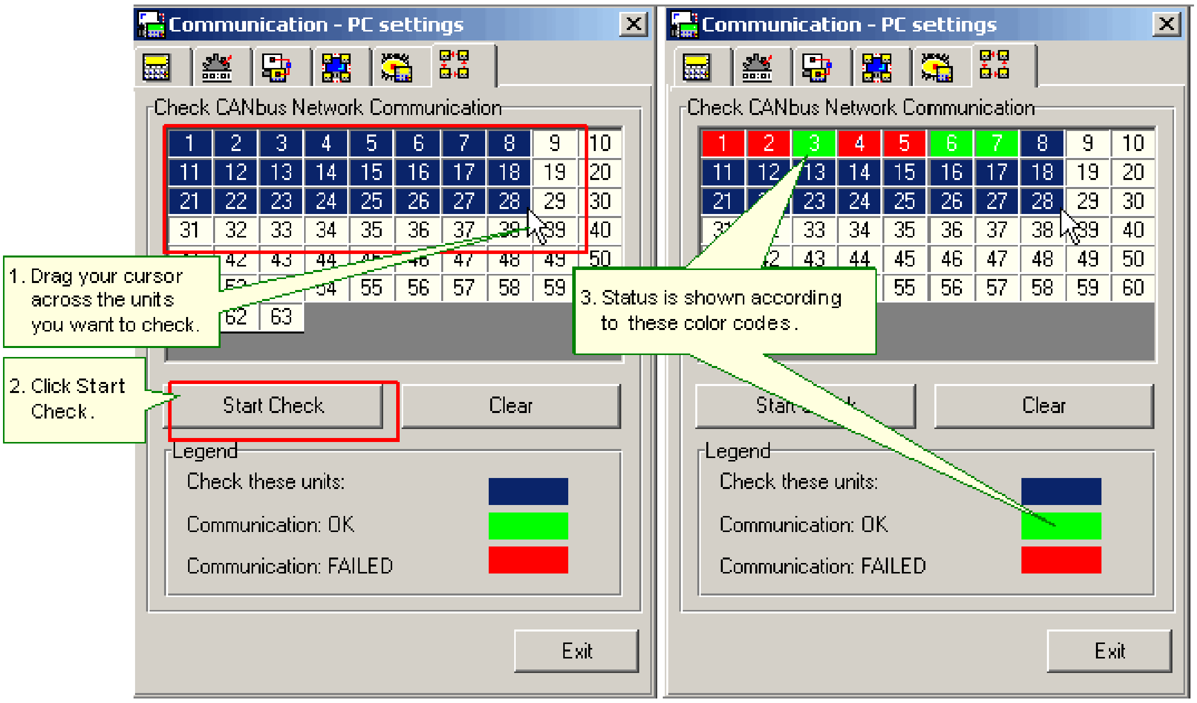
The controller model that is shown on the display is the one selected in Hardware Configuration.

During a Remote Access session, you can:

- Use your cursor to operate the controller keypad and activate touch-screen objects (relevant models).
- Use your PC keyboard to operate the controller keypad (numeric keys, function keys <F1> to <F8>). Note that the Vision <ESC> key is <E> on the PC keyboard, and that <F9> is reserved for activating Online mode.
- Enter Information Mode by pressing the <i> key with your cursor.

## Check CANbus Network Status

The network status is checked via the bridge.  Access Check Network Status by opening the Connection menu, selecting Communication & OS, then clicking the network tab.



Communication - PC settings
Check CANbus Network Communication
Start Check
Clear
Exit

1. Drag your cursor across the units you want to check.

2. Click Start Check.

3. Status is shown according to these color codes.

Legend

| | |
|---|---|
| Check these units: | ■ |
| Communication: OK | 🟩 |
| Communication: FAILED | 🟥 |

## CANbus Network Problems

| | |
|---|---|
| Compiler error 43 | This error may result because:<br>● The project includes both CANbus modes. Note that a controller cannot run UniCan and CANbus ISC simultaneously.<br>● The Ladder application does not include a COM Init FB that initializes the CANbus port. |

SB 236 indicates that there is an error in the CANbus network.

SI 236 CANbus Network: Failed Unit ID. Note that the first 3 bits turn ON only when the controller is unable to broadcast via the CANbus port. The value of SI 236 indicates the following messages:

| Value | Message | May result from: |
|---|---|---|
| 0 | No Acknowledgement | CANbus power failure, crossed wires, incorrectly set termination points. |
| 1 | CANbus OFF | |
| 2 | CANbus Warning error | Poor transmission quality due to faulty wiring, or if the cable length exceeds recommendations. |
| 4 | One or more networked units cannot be read. If this bit is ON, check SI 238, SI 240-243. | |

SI 237 CANbus Network Communication Error Code is a bitmap which indicates the LAST unit that cannot be read. Note that each controller can receive messages from up to 8 others. Example: Assume that there are 6 controllers in the network (3,6,8,13,17,34). Controller 3 is waiting for data from controllers

8 and 13 and 17. If the controller does not receive data from controller 13 (assume a 1 second timeout) then SI 237 will contain a value of 13. Bit 4 in SI 236 will also turn ON. Once controller 3 has received the data, Bit 4 turns OFF.

SIs 240, 241, 242, and 243 serve as a bitmap indicating which unit is in error. If, for example, the network includes unit ID numbers 8, 9 and 13, and PLC #9 cannot be accessed, then the ninth bit in SI 240 will turn ON. When the error is fixed, the bit falls to OFF.

### CANbus troubleshooting

| **Problem** | **Possible cause** | **Recommended Action** |
|---|---|---|
| Failed communication | Baud rate settings | All controllers in the network must be set to the same CANbus baud rate. Baud rate may be set:<br>• By initializing a port via the COM Init FB.<br>• Temporarily via Info mode;however, note that the baud rate will be reset at the next power-up |
| | Termination resistors | Termination resistor settings are provided in the chapter **Communications** in the controller's user guide. |
| | CANbus power supply | Check that the CANbus power supply is properly connected, and that the voltage is in the permissible range as described in the chapter Communications in the controller's user guide. |
| | Incorrect ID number | You may not have assigned the correct unit ID number in your operand addresses (between 1-63). Check the OPLC settings tab Communicate with OPLC. |
| PC cannot communicate with bridge | Incorrect setting | When you communicate with the controller unit that you are using as a bridge to the network, either enter the ID# of the bridge or select Direct Connection in the OPLC settings tab Communicate with OPLC. |
| PC cannot communicate with network | Communication settings | Make sure the current RS232 parameters in your project are the same as the parameters that are actually in the bridge. Check these topics:Check Network Status,Vision Communication PC Settings |
| | Incorrect baud rate | The bridge's RS232 port's baud rate cannot be set below 9600. |

# Index