Protocol FB *Help Rev: 08/04/08*

Changes in V5.00

Known Issue: Previous to V4.54

Changes in V1.40

Changes in V1.36

Use the Communication Protocol operations to enable your controller to exchange data with external devices, such as frequency converters, bar-code readers, and printers via a Vision COM port. Protocol operations are located on the FBs menu.

How the Protocol FB communicates data between Vision controllers and other devices

A device, such as a magnetic card reader, may use its own proprietary protocol. If you know the protocol's structure, you can use the Protocol FB to structure messages accordingly. This enables a Vision controller to exchange data with the device using the device's own protocol.

Note that before you can use Protocol operations, you must configure the COM port to be connected to the external device in accordance with the device's requirements. This is done by placing a COM Port Init FB in your Ladder application.

Basic Example

For more information regarding Protocol operations, refer to the topics below.

FB Operations

Configuration

Scan

Send

Copy Buffer Contents

Reset Buffer

Note
- Reset Buffer **must** be used in the Ladder application to enable new messages to be received.
- If you want to access a PLC via PC, remember that a PC accesses a PLC via a serial port. Vision controller ports are served by three communication buffers. Note that if all three buffers are busy processing communications, new requests are ignored until a buffer is free.

Examples

Sample applications may be found in the VisiLogic Examples folder. This folder contains field-tested VisiLogic (.vlp) sample applications. You can open this folder via the Help Menu.

The folder is typically located at: C:\ProgramFiles\Unitronics\VisiLogic\Examples\Verx.xx, where x.xx indicates the version of VisiLogic.

Related Topics

Variables

Checksum Configuration

Communication Protocol Error Messages

Com_Port Init

PLC-side Modems

Modem Status, Operands, and Error Messages

Modem Troubleshooting

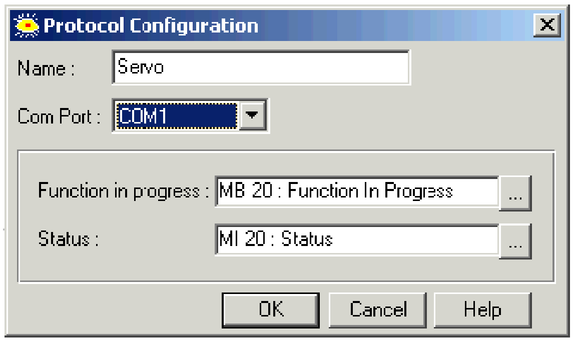
System Operands (SI) (SL) _(SB) _(SDW)

ASCII Table

Protocol: Configuration

You create messages and attach variable data in Protocol Send and Scan operations. Each operation is linked to a Protocol Configuration. When a Send operation is called in your program, the Configuration determines which COM port is used for outgoing messages; when a Scan operation is activated, the PLC monitors that COM port for incoming messages.

Note
- The Configuration should be placed in the Main Routine, before any other FB operations. If the configuration is not active, Protocol operations will not be processed.
- Before you place this FB in your Ladder, you must use a COM Port INIT FB to configure a COM port.



Protocol Configuration

Name : Servo

Com Port : COM1

Function in progress : MB 20 : Function In Progress

Status : MI 20 : Status

Parameter		**Function**
Name		Identifies the configuration. You use the name to link Protocol Send and Scan operations to a configuration.
COM Port		Determines which port is used for communicating the configured protocol.
Function in Progress	MB	Link an MB; this MB will be ON when the Configuration is processing Scan or Send operations.
Status	MI	If errors occur when the Configuration is used for a Send or Scan operation, the value of the linked MI indicates which error has occurred.

Related Topics

Protocol: Send

Protocol: Scan

Protocol: Reset Buffer

Protocol: Copy Buffer Contents

Variables

Checksum Configuration

Protocol Status Messages

Protocol: Scan

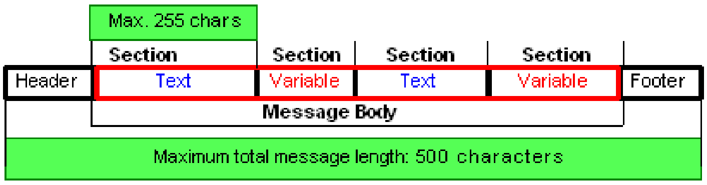
A Protocol Scan operation contains messages. Activating the Scan causes the PLC to check if it has received any of the messages contained within the Scan. The PLC can only receive messages that are contained in a Scan FB.

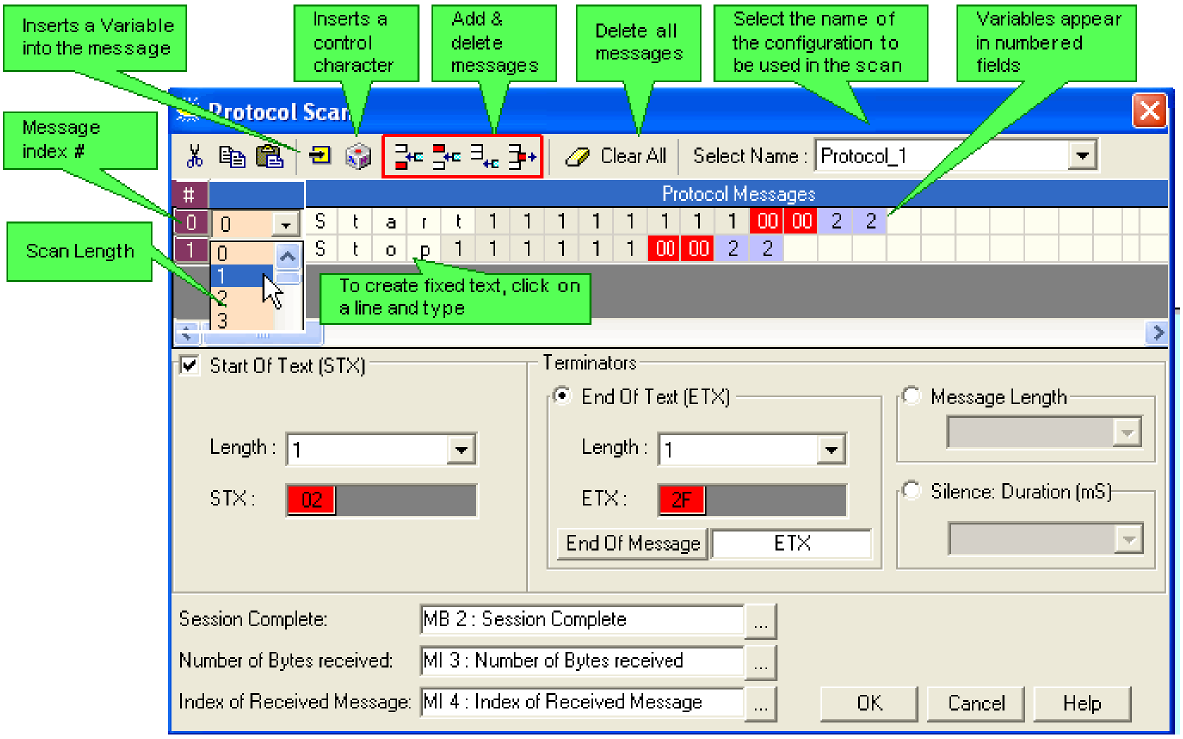
Each Protocol Scan operation is linked to a particular Protocol Configuration.
The Scan only registers messages that are received via the COM port defined in that Configuration.

Scan is located under FBs>Protocol.

Note ♦ In order to receive new messages, you must clear the communication buffer by using the Reset Buffer operation. You can save the contents of the buffer at any time by using the Copy Buffer operation.

♦ Maximum message length is 500 characters; a message may include up to 50 variables. Note that with in the body of a message, no section--whether text or variable--may exceed 255 characters.




Inserts a Variable into the message
Inserts a control character
Add & delete messages
Delete all messages
Select the name of the configuration to be used in the scan
Variables appear in numbered fields
Message index #
Scan Length
To create fixed text, click on a line and type
Protocol Scan
Clear All
Protocol Messages
#
OK
Cancel
Help

Select Name : Protocol_1

#																		
0	0	S	t	a	r	t	1	1	1	1	1	1	1	1	00	00	2	2
1	0	S	t	o	p	1	1	1	1	1	1	00	00	2	2			

☑ Start Of Text (STX)
Length : 1
STX : 02

Terminators
● End Of Text (ETX)
Length : 1
ETX : 2F
End Of Message: ETX
○ Message Length
○ Silence: Duration (mS)

Session Complete: MB 2 : Session Complete
Number of Bytes received: MI 3 : Number of Bytes received
Index of Received Message: MI 4 : Index of Received Message

Message index — Identifies the incoming message.

Scan Length — This enables you to define the length of each incoming message, via the drop-down box that follows Message index #. Scan Length causes the Scan FB to use the length of a message--including header and terminator--to define whether an incoming message is legal.
The length of an incoming message cannot be less than the length specified in the Scan operation.
Note ♦ This parameter overrides Start Of Text and Terminator settings.
If you use Scan Length, arrange the messages in ascending order, for example, message #0 may contain 3 bytes, #1 may contain 4 bytes, #2 may contain 6 bytes, etc.

Start of Text (STX) — Check Start Of Text to use an STX parameter to indicate the beginning of a message.

- Length:
  Enter the number of bytes required by the external device's protocol to mark STX.
- STX Character: Click a 'byte' box and select an ASCII character .

Terminators — Select one of the terminator parameters to indicate where the message ends.

Parameter **Function**

Session Complete MB

Turns ON when:

- Turns ON when the PLC receives a valid message, including the message's STX and terminator, after the Linked Configuration's Function in Progress MB turns OFF.

Turns OFF when

- Turns OFF whenever the Protocol Scan operation is called by the program.
- Turns OFF when a Reset Buffer operation runs.
- In your program, link the Session Complete bit to a positive transition contact, then use this condition to activate a Copy Buffer FB, as well as number of Bytes Received and Number of Received Message.

Note ♦
-If the PLC receives a message which is not defined in the Scan FB, the message is invalid. The Message Received MB remains OFF.
-If a message in the Scan FB contains a Numeric Variable field', and the PLC receives this message with non-numerical characters in the field (except for leading spaces), the message is invalid. The Message Received MB remains OFF.

-An example of leading spaces is if, for example, the Numeric Variable field is 4 boxes long, and the 'number' 22 is received. In this case, the Scan FB will register the number22 preceded by 2 leading spaces (_ _ 2 2).

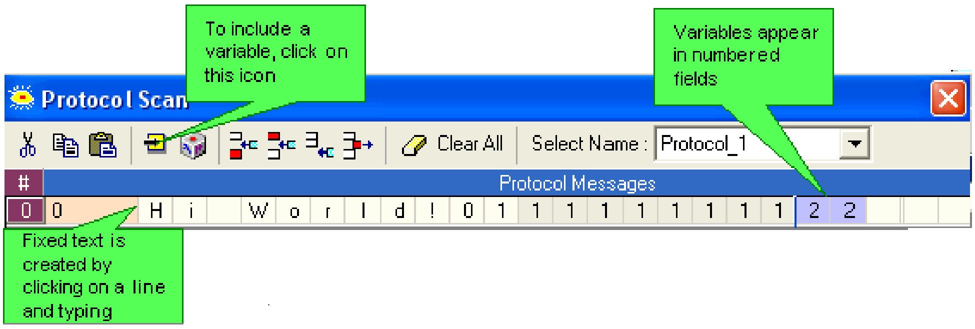
Number of Bytes Received MI — This holds the number of bytes currently in the buffer. It is initialized by the system when a different message is received.

Index of Received Message MI — This holds the index number of the received message. It changes when another message enters the system.
The value in the linked MI is valid only when the Session Complete bit in ON. During this time, the linked MI contains the index number of the last valid incoming message.
Once an invalid message has been received, the first incoming character of any message causes -1 to be written in the linked MI.

Protocol Messages

Messages can include:

- Fixed Text



#																						
0	0	H	i		W	o	r	l	d	!	0	1	1	1	1	1	1	1	1	1	2	2

To include a variable, click on this icon

Variables appear in numbered fields

Fixed text is created by clicking on a line and typing

- Variables
- Control Characters

Example

Scan operations containing more than one message

Related Topics

Protocol: Configuration

Protocol: Scan

Protocol: Reset Buffer

Protocol: Copy Buffer Contents

Variables

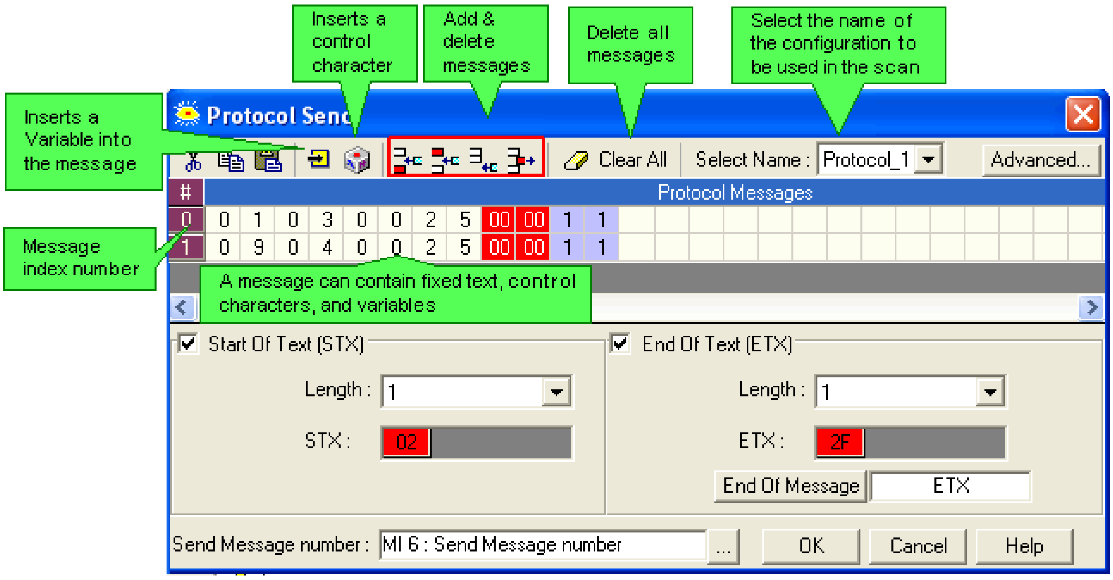
Checksum Configuration

Communication Protocol Error Messages

ASCII Table

Protocol: Send

Each Protocol Send operation is linked to a particular Configuration; outgoing messages are sent via the COM port selected in that Configuration. Send is located on the FBs menu.



Inserts a control character

Add & delete messages

Delete all messages

Select the name of the configuration to be used in the scan

Inserts a Variable into the message

Message index number

A message can contain fixed text, control characters, and variables

#												
0	0	1	0	3	0	0	2	5	00	00	1	1
1	0	9	0	4	0	0	2	5	00	00	1	1

☑ Start Of Text (STX)

Length : 1

STX : 02

☑ End Of Text (ETX)

Length : 1

ETX : 2F

End Of Message ETX

Send Message number : MI 6 : Send Message number

Start of Text (STX)

Check Start Of Text to use an STX parameter to indicate the beginning of a message.

- Length:
  Enter the number of bytes required by the external device's protocol to mark STX.
- STX Character: Click a 'byte' box and select an ASCII character .

End of Text (ETX)

The ETX parameter indicates where the message ends.

- Length:
  Enter the number of bytes required by the external device's protocol to mark ETX.
- ETX: Click a 'byte' box and Select an ASCII character .

End of Message format:
- Note that the protocol may require that a checksum be either part of the message body or part of the End Of Message

	- If your protocol requires a checksum as part of the End of Message,click the End of Message button to define checksum placement and format. - Use the options to configure the checksum according to the requirements of your particular protocol.
Send Message Number	Determines which message in the FB will be sent. In the program, store a message index numbers in the linked register. When the FB is called, message whose index number currently in the linked MI will be sent.

Advanced Button

Protocol Messages

Messages can include:

- Fixed Text
- Control Characters
- Variables

Example

Related Topics

Checksum Configuration

Protocol: Scan

Protocol: Reset Buffer

Protocol: Copy Buffer Contents

Variables

Checksum Configuration

Communication Protocol Error Messages

ASCII Table

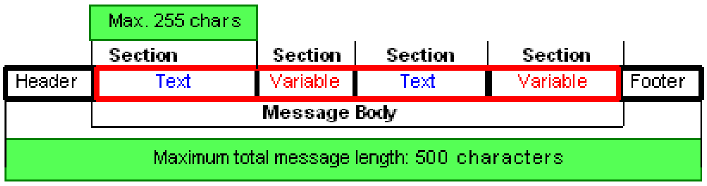
Protocol: Copy Buffer Contents

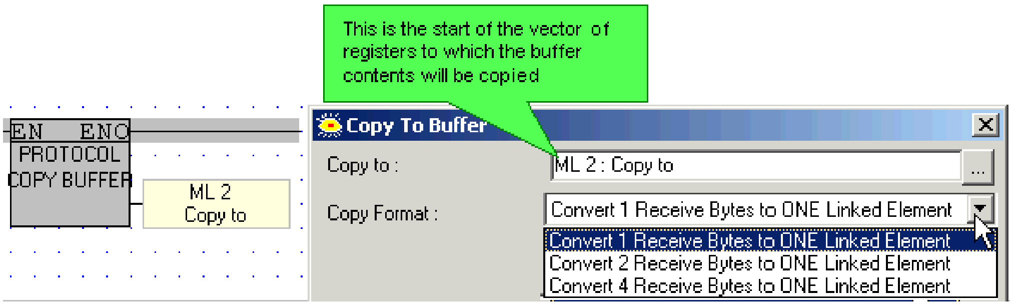
You can copy the current contents of the buffer which receives incoming messages at any point in your application by including the Copy to Buffer operation, which is located on the FBs menu.

This operation copies whatever information is in the buffer at the time the operation is called.

Note ♦ Note that copying the buffer contents is optional.

♦ Maximum message length is 500 characters; a message may include up to 50 variables. Note that with in the body of a message, no section--whether text or variable--may exceed 255 characters.



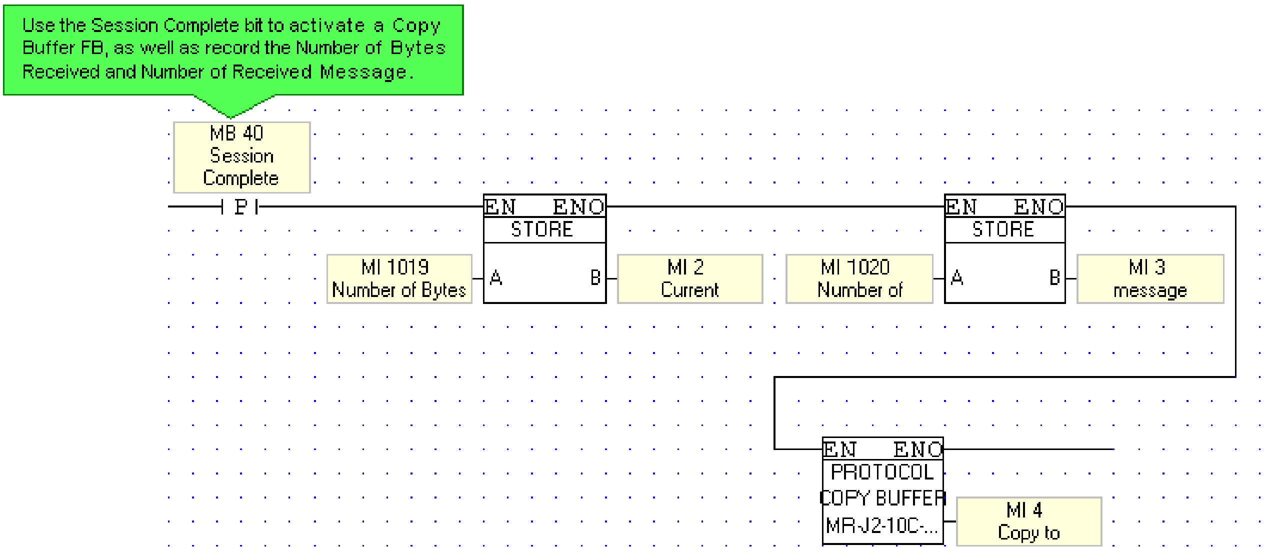

This is the start of the vector of registers to which the buffer contents will be copied

EN ENO
PROTOCOL
COPY BUFFER
ML 2
Copy to
Copy To Buffer
Convert 1 Receive Bytes to ONE Linked Element
Convert 2 Receive Bytes to ONE Linked Element
Convert 4 Receive Bytes to ONE Linked Element

Copy to : ML 2 : Copy to

Copy Format : Convert 1 Receive Bytes to ONE Linked Element

Copy Format



Use the Session Complete bit to activate a Copy Buffer FB, as well as record the Number of Bytes Received and Number of Received Message.

Related Topics

Checksum Configuration

Protocol: Send

Protocol: Scan

Protocol: Reset Buffer

Variables

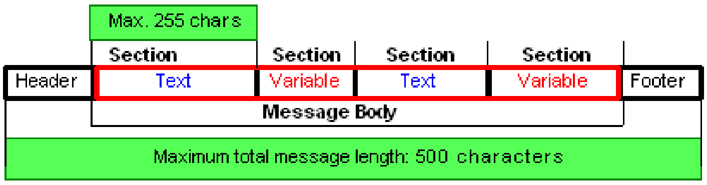
Checksum Configuration

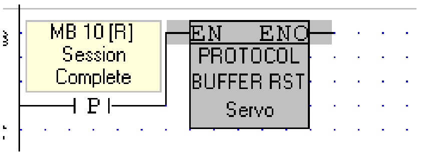
Communication Protocol Error Messages

Protocol: Reset Buffer

In order to receive new messages, you must clear the communication buffer by using the Reset Buffer operation, which is located on the FBs menu.

Note
- Note that the Session Complete bit turns OFF automatically when the Reset Buffer operation runs.
- Maximum message length is 500 characters; a message may include up to 50 variables. Note that with in the body of a message, no section--whether text or variable--may exceed 255 characters.





Related Topics

Protocol: Configuration

Protocol: Send

Protocol: Scan

Protocol: Copy Buffer Contents

Variables

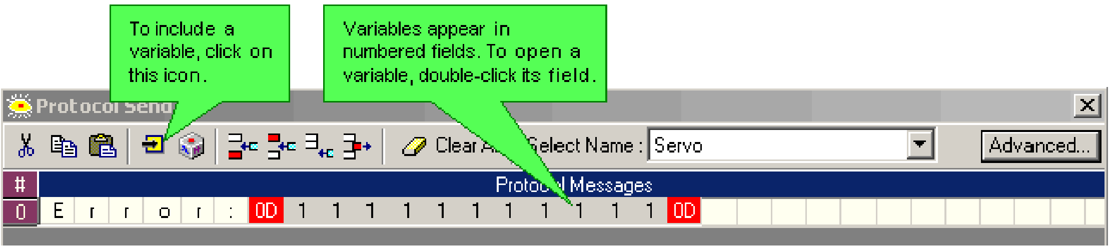
Checksum Configuration

Communication Protocol Error Messages

Variables

Variables enable you to send and receive a variety of dynamic data from your process.

Variable Types: Send



To include a variable, click on this icon.

Variables appear in numbered fields. To open a variable, double-click its field.

The PLC sends messages via the Configuration linked to the Send operation. These are the types of variables that can be attached to an outgoing message.

- Binary text
- Numeric
- Stream
- Stream Variable Format
- List of texts
- Checksum
- Date
- Time

Variable Types: Scan

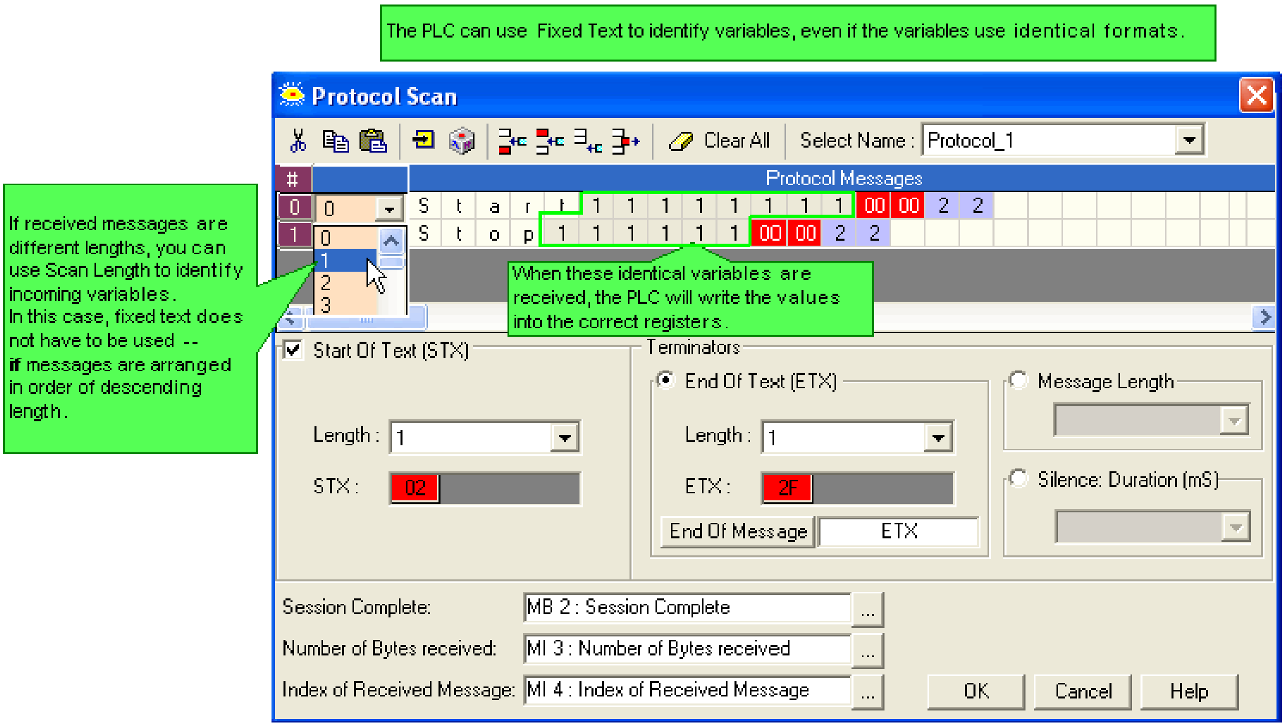
The PLC scans for messages that enter the system via the Configuration linked to the Scan operation. These are the types of variables that can be attached to an incoming message.

To correctly identify incoming variables, use either Scan Length, or include fixed text.

Notes

- Scan Length can be defined for each message via the drop-down box that follows Message index #. Scan length causes the Scan FB to use the length of a message to define whether an incoming message is legal.
  This parameter overrides Start Of Text and Terminator settings.
  If you use Scan Length, arrange the messages in ascending order, for example, message #0 may contain 3 bytes, #1 may contain 4 bytes, #2 may contain 6 bytes, etc.
- If a Scan operation contains more than one message with identical variables that use the same format, the PLC will overwrite the variable values as they are received.

To enable the PLC to differentiate between the messages and their attached variables, include fixed text and/or control characters within the message.



The PLC can use Fixed Text to identify variables, even if the variables use identical formats.

If received messages are different lengths, you can use Scan Length to identify incoming variables. In this case, fixed text does not have to be used -- if messages are arranged in order of descending length.

When these identical variables are received, the PLC will write the values into the correct registers.

Protocol Scan

Select Name : Protocol_1

#		Protocol Messages
0	0	S t a r t 1 1 1 1 1 1 1 1 00 00 2 2
1	0	S t o p 1 1 1 1 1 1 00 00 2 2

☑ Start Of Text (STX)

Length : 1

STX : 02

Terminators

● End Of Text (ETX)

Length : 1

ETX : 2F

End Of Message: ETX

○ Message Length

○ Silence: Duration (mS)

Session Complete: MB 2 : Session Complete

Number of Bytes received: MI 3 : Number of Bytes received

Index of Received Message: MI 4 : Index of Received Message

- Numeric Receive
- Receive Stream
- Checksum

Click to view an ASCII Table

Related Topics

Protocol: Configuration

Protocol: Send

Protocol: Scan

Protocol: Reset Buffer

Protocol: Copy Buffer Contents

Checksum Configuration

Communication Protocol Error Messages

ASCII Table

Checksum Configuration

Use the options to configure the checksum according to the requirements of your particular protocol.

Format	Select the data representation method: Decimal ASCII, Hex as ASCII, or Binary.
Calculate Offset	The two Offset parameters determine the delimiters of the data to be calculated for checksum.
Calculation Type	This is the type of calculation that will be performed on the data defined above. **SUM** **XOR** CRC: acronym for Cyclic Redundancy Check, a procedure used in checking for errors in data transmission. A complex polynomial is used to generate a number that is based on the transmitted data. The sending device performs the calculation before transmission, and then sends the result to the receiving device. The receiving device repeats the same calculation after it receives the data. If both devices arrive at the same result, the transmission is assumed to be error-free. This is called a redundancy check because each transmission includes not only data but extra (redundant) error-checking values **CRC-16** **CRC-CCITT** **2's complement**
Modulus method	256 (100H): 8 bits 65536 (10000H): 16 bits
Checksum Length	This determines the length of the field which will hold the calculated checksum result. Example
Advanced: Set	From Value and To Value determine a range for the checksum result ;if the final value falls within these limits, the

Limits	function can automatically add a set value to the calculated checksum.

Related Topics

Protocol: Configuration

Protocol: Send

Protocol: Scan

Protocol: Reset Buffer

Protocol: Copy Buffer Contents

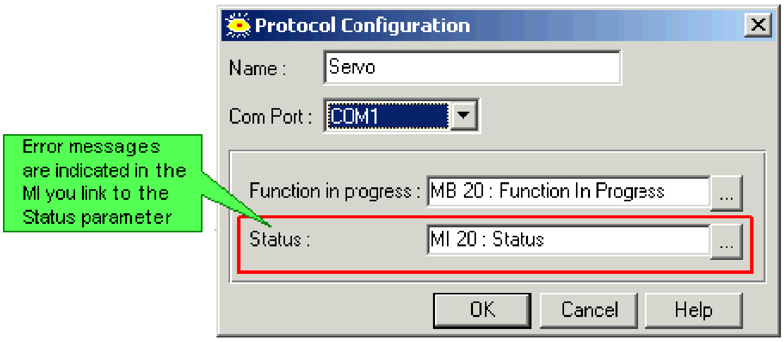
Variables

Communication Protocol Error Messages

Protocol Status Operands and Messages

The Status Operands show the state of Protocol communications.

All of the Status operands linked to Protocol FBs should be assigned Power-up Values; bits should be reset, and registers initialized to 0. To learn about System Operands connected to modems and COM Ports, check the related topics at the bottom of this topic.



Error messages are indicated in the MI you link to the Status parameter

Protocol Configuration

Name : Servo

Com Port : COM1

Function in progress : MB 20 : Function In Progress

Status : MI 20 : Status

Protocol Operation Status Operands
When you place a Protocol Configuration your application, the linked operands indicate the status of Protocol operations.

Function in Progress Shows status of Protocol Configuration	**MB**	**Turns ON when:** ● Configuration begins ● When Send or Scan begins **Note** ♦ Messages cannot be sent while this MB is ON. Use an indirect contact for this MB as an activating condition for Send operations.	**Turns OFF when** ● The Protocol: Configuration is finished. ● When Scan is complete, when the PLC receives the message's STX and terminator. ● The Status Message indication changes
Status Messages	**MI**	● Automatically initialized to 0 when a Protocol operation is activated. ● Updated at the end of each attempt to communicate via the Protocol Configuration. ● Indicates status of Protocol communications, according to the table below. Note that the current value always shows the most **recent** status.	

Value	Error Message	Indication
0	No Error	Operations successful
1	Send: General Error	Indicates an unknown system error.
2	Send: COM Port busy	The COM port that is selected in the active Protocol Configuration cannot currently be accessed.
3	Send: Invalid Message Index	The message that is referenced in the Send operation does not exist.
4	Send: Invalid Operand Type	The message references an operand type that does not exist in the system.
5	Send: Invalid Operand Address	The message references an operand address that does not exist in the system.
6	Send: Data Overflow	The message data exceeds 512 bytes.
7	Send: Format Error	The message contains variable fields that are not long enough for the attached variable data.
257	Receive: General Error	Indicates an unknown system error.
258	Receive: Invalid message	The received message does not exist in the system.

259	Receive: Checksum Error	The message may have been corrupted during transmission.

Related Topics

Protocol: Configuration

Protocol: Send

Protocol: Scan

Protocol: Reset Buffer

Protocol: Copy Buffer Contents

Variables

Checksum Configuration

Example: Mitsubishi Frequency Converter

Examples

Basic Example

Advanced Mitsubishi Frequency Converter

Related Topics

Protocol: Configuration

Protocol: Send

Protocol: Scan

Protocol: Reset Buffer

Protocol: Copy Buffer Contents

Variables

Checksum Configuration

ASCII Table

COM Port: Init

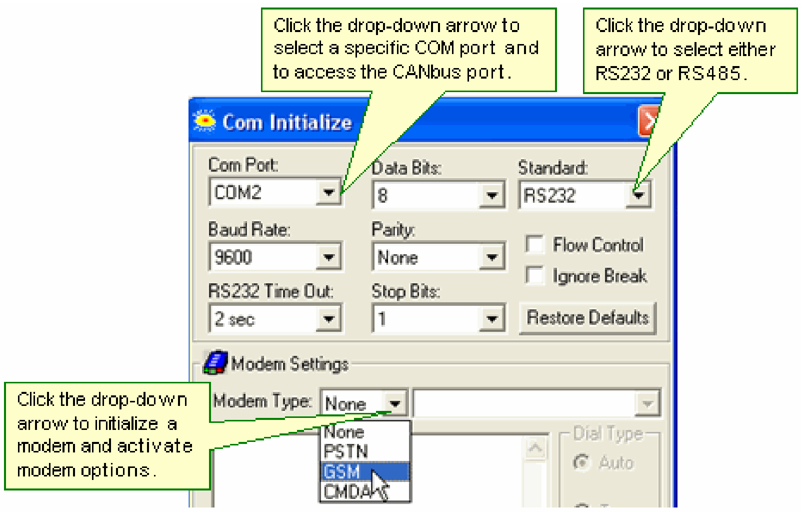
COM Init is located on the Com menu. Use this function block:

- 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.

  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.

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

Related Topics

COM Port and Data Communications

Dial and Hang-up

Dialing from the controller's keypad

Modems

Last Call (CLIP) – Identifying Callers

COM Port/Modem Status, System Operands, & Error Messages

CANbus Networking

RS485 Options

MODBUS

MODBUS IP

Ethernet: Card Init

SMS Messaging

Communications Protocol FB

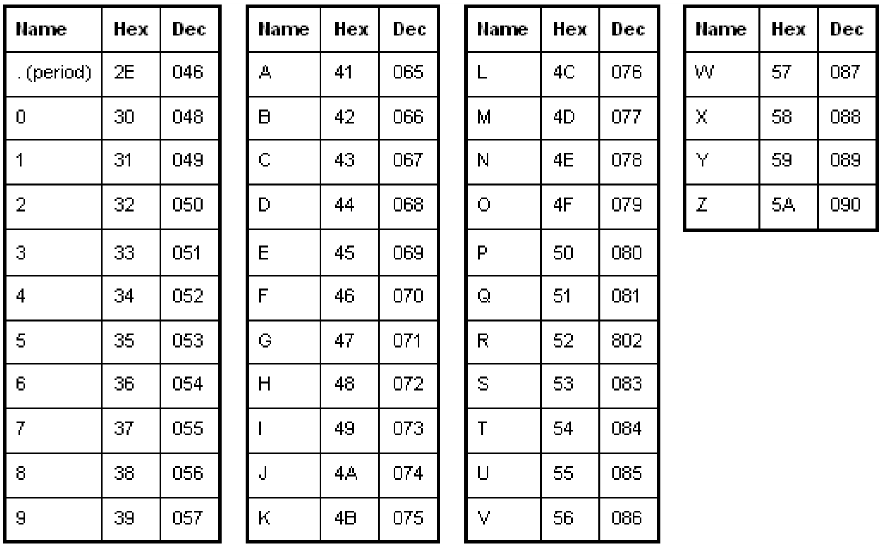
Placing a Function in a Net

System Operands (SI) (SL) (SB) (SDW)

Operands

Find

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
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
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
W	57	087
X	58	088
Y	59	089
Z	5A	090