
SEND_REQ: Sending Requests

This document contains the following topics:

Topic	Page
Description	4
List of UNI-TE Requests	6
Assisted Entry Screen	11
Example of How to Send a UNI-TE Request	13
Example: Changing IP Parameters with SEND_REQ	15
Using the SEND_REQ Function	16

Description

Function Description

The `SEND_REQ` function is used to code and send all UNI-TE and Modbus/Jbus requests, as well as receive the associated responses.

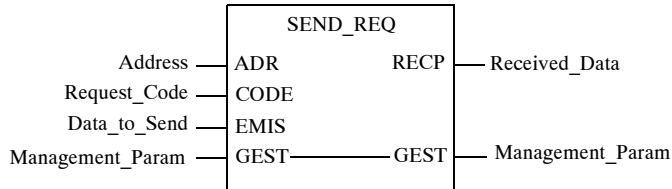
Coding details for UNI-TE requests are provided in the reference manual: Communication Reference TSX DR NET.

Details on coding the Modbus/Jbus requests are provided in the TSX DG MDB manual.

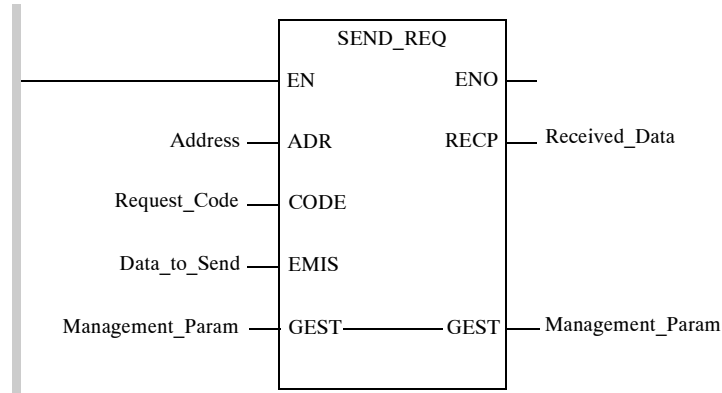
Modbus requests common to all Schneider devices are presented in the **Architecture and communication services** manual.

The additional parameters `EN` and `ENO` can be configured.

FBD Representation



LD Representation



IL Representation

LD Address
`SEND_REQ Request_Code, Data_to_Send, Management_Param, Received_Data`

**ST
Representation**

```
SEND_REQ(Address, Request_Code, Data_to_Send,
Management_Param, Received_Data);
```

**Parameter
Description**

The following table describes the input parameters:

Parameter	Type	Comment
Address	ARRAY [0.. 5] OF INT	Address of the destination entity of the exchange. The type of address depends on the request sent. It is therefore possible, for example, to broadcast the RUN request (ALL) whereas it is impossible to send an identification request to several devices simultaneously.
Request_Code	INT	Request to be sent to the destination device, also called server . Requests may be UNI-TE requests (See <i>List of UNI-TE requests, p. 6</i>) or Modbus requests.
Data_to_Send	ARRAY [n... m] OF INT	Integer table to be sent to the destination device of the request. This table depends on the request sent. It must have a minimum length of 1 element, even if the request used does not require data to be sent (for example an identification request). Note: It is imperative that the length of the data to be sent (in bytes) be assigned to the fourth word of the management table before launching the function, in order for this to be correctly executed.

The following table describes the input/output parameters:

Parameter	Type	Comment
Management_Param	ARRAY [0.. 3] OF INT	Exchange management table

The following table describes the output parameters:

Parameter	Type	Comment
Received_Data	ARRAY [n... m] OF INT	Integer table containing the data returned by the server device which was the destination of the request. Although certain requests do not require a response (for example a Run request), it is nevertheless necessary to reserve a minimum table of 1 integer each time the SEND_REQ function is used. Note: The size of the data received is written automatically by the system in the fourth word of the management table. Note: In certain cases (reading of word tables for example) it is necessary to resequence the objects received by using the ROR1_ARB function (shift of a byte in a table).

List of UNI-TE Requests

At a Glance

The UNI-TE protocol is used to:

- identify and diagnose all the devices that have a UNI-TE server,
- provide a set of services giving read/write access to type data,
- download data from device to device,
- protect a server against concurrent connections during a critical period.

These different services can be obtained using the `SEND_REQ` function, with coding of the UNI-TE request to be sent.

Note: to find out details and the list of requests recognized by each device, refer to the TSX DR NET reference manual.

The following tables provide a non-exhaustive list of the requests recognized by Premium devices.

General Usage Requests

These requests are used to identify and diagnose all the types of devices that have a UNI-TE server.

Request Name	Request Code	Report Code	Comment
IDENTIFICATION	16#0F	16#3F	Provides the following information <ul style="list-style-type: none"> ● product range, ● specific application type, ● product type, ● catalogue reference.
READ_CPU	16#4F	16#7F	Performs system diagnostics on any device.
PROTOCOL_VERSION	16#30	16#60	Used to adapt the version of the protocol between two communicating entities.
MIRROR	16#FA	16#FB	Tests the correct routing of data between two communicating devices.

Dynamic Modification of IP Parameters

Change IP Parameters

Request Name	Request Code	Comment
REQUEST_CODE	16#37	request code function
CHANGE_IP_PARAMETERS	16#13	changes IP parameter subfunction

To ensure that the ETY module is ready for operation, allow the PLC to run (in run mode) for 15 seconds after the last stop mode before issuing the SEND_REQ function. After the user issues the CHANGE_IP_PARAMETERS command and the ETY module accepts the new parameters, the ETY module resets and begins operations with the new parameters.

Note: FDR clients that use the ETY module for a server must be rebooted after the ETY's IP address change is completed. Otherwise, these FDR clients can not update their parameter files on the FDR server (the ETY module).

Note: Constant words contain the original configuration parameters, not those that were updated after the IP address change.

Note: You can see the new configuration (IP parameters, subnet mask, and gateway address) in the Unity Pro ETY debug screen. You can also see the newly assigned IP address in the Ethernet Module Statistics web page, however, you must be aware that the IP Address, Subnetwork mask, and Gateway data on this page reflect the old configuration.

Parameter Usage for IP Address Change

This table is based on a sample IP address of 139.158.10.7, subnet mask 255.255.248.0, and gateway address of 139.158.8.1.

Parameter	Type	Value	Comment
ADDRESS	array [0...5] of INT	ADDR ('rack.slot.channel.SYS')	Example: ADDR (0.x.0.SYS') x = slot in which ETY module is installed
REQUEST_CODE	INT	16#37	

Parameter	Type	Value	Comment
Data_to_Send	array [0...8] of INT	byte 1: subfunction (13h)	high byte
		byte 2: subfunction (96h)	low byte
		byte 3: 0	Value is ignored.
		byte 4: 0	Value is ignored.
		byte 5: IP address 2 (158)	1-239 (high byte)
		byte 6: IP address 1 (139)	0-255 (low byte)
		byte 7: IP address 4 (7)	0-255 (high byte)
		byte 8: IP address 3 (10)	0-255 (low byte)
		byte 9: subnet mask 2 (255)	255
		byte 10: subnet mask 1 (255)	0-255
		byte 11: subnet mask 4 (0)	0-255
		byte 12: subnet mask 3 (248)	0-255
		byte 13: gateway 2 (158)	1-239 (Gateway address must be on the same subnet as the IP address.)
		byte 14: gateway 1 (139)	0-255
		byte 15: gateway 4 (1)	0-255
		byte 16: gateway 3 (8)	0-255
Manage_Param	<i>word number</i>	<i>high byte</i>	<i>low byte</i>
	1	activity report	00
	2	operation report (see note 1)	communication report (see note 1)
	3	time out (ms)	
	4	18 (INT)	
	Note 1: The following table gives details for IP address codes (successful and error)		

Refer to the example for changing IP parameters with SEND_REQ (See *Example: Changing IP Parameters with SEND_REQ*, p. 15).

Change IP Address Codes

Operation Report	Communication Report	Meaning
<i>successful code</i>		
FE (hex)	00 (hex)	send request (SEND_REQ) successfully changed IP address
<i>error codes</i>		
01 (hex)	FF (hex)	invalid request code value (e.g., not 16#37)
00 (hex)	03 (hex)	send request to ETY IP address instead of across backplane
00 (hex)	07 (hex)	bad address mapping to the ETY
16 (hex)	FF (hex)	invalid IP address
17 (hex)	FF (hex)	invalid subnet
18 (hex)	FF (hex)	invalid gateway address
19 (hex)	EF (hex)	invalid network address
1A (hex)	FF (hex)	ETY's IP address was already set to send request (SEND_REQ)
FD (hex)	00 (hex)	send request (SEND_REQ) did not successfully changed IP address

Accessing Objects

These requests provide a set of services that give read/write access to internal bit and word-type data, system bit and word-type data, floating point data, constant data and SFC data.

Request Name	Request Code	Report Code	Comment
READ_OBJECT	16#36	16#66	Used to read one or more consecutive objects of the same type.
WRITE_OBJECT	16#37	16#FE	Used to write one or more consecutive objects of the same type.
READ_INTERNAL_BIT	16#00	16#30	Used to read the value of an internal bit.
WRITE_INTERNAL_BIT	16#10	16#FE	Used to write the value of an internal bit.
READ_INTERNAL_WORD	16#04	16#34	Used to read the value of an internal word.
WRITE_INTERNAL_WORD	16#14	16#FE	Used to write the value of an internal word.

Management of Operating Modes

These requests provide a set of services that can be used to manage a processor's operating modes.

Request Name	Request Code	Report Code	Comment
RUN	16#24	16#FE	Used to launch the execution of a processor's tasks.
STOP	16#25	16#FE	Used to stop the execution of a processor's tasks.
INIT	16#33	16#63	Used to trigger a hot or cold re-start.

Reservation Management

These requests provide a reservation mechanism that is used to protect a server against concurrent connections during a critical period.

Request Name	Request Code	Report Code	Comment
RESERVE	16#1D	16#FE	Enables a client to generally or partially reserve a server's functions.
RELEASE	16#1E	16#FE	Enables a client to free up the reserved server.
I_AM_ALIVE	16#2D	16#FE	Used to maintain the reservation.

Assisted Entry Screen

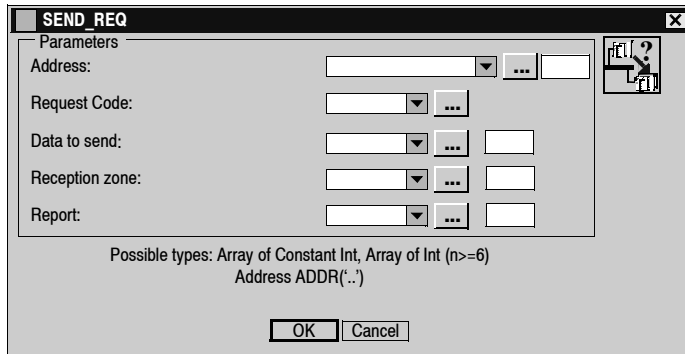
At a Glance

For this communication function, you can call up the assisted entry screen.

Note: The symbols are accepted.

Illustration

The following screen shows an example of the assisted entry screen of the function:



Address

The possible type of object is:

- ADDR (STRING).
- ARRAY [0 . . 5] OF INT.

Note: if you enter a value directly in the field, the assisted address entry button turns gray.

Request Code

The possible objects are of the INT type:

- variables,
- constants,
- immediate value.

Note: if you enter a constant, an entry field appears to enter the constant. If you enter a variable, it can be located or not.

Data to Send These data to be sent are stored in the form of an integer array. This array can be located or not.

Reception Zone The reception zone is an array of integers. This array can be located or not, and its size depends on the request code used.

Report The report is an array of 4 integers.

Note: be careful not to use several identical memory areas for report tables, otherwise the variable read function may not work.

Example of How to Send a UNI-TE Request

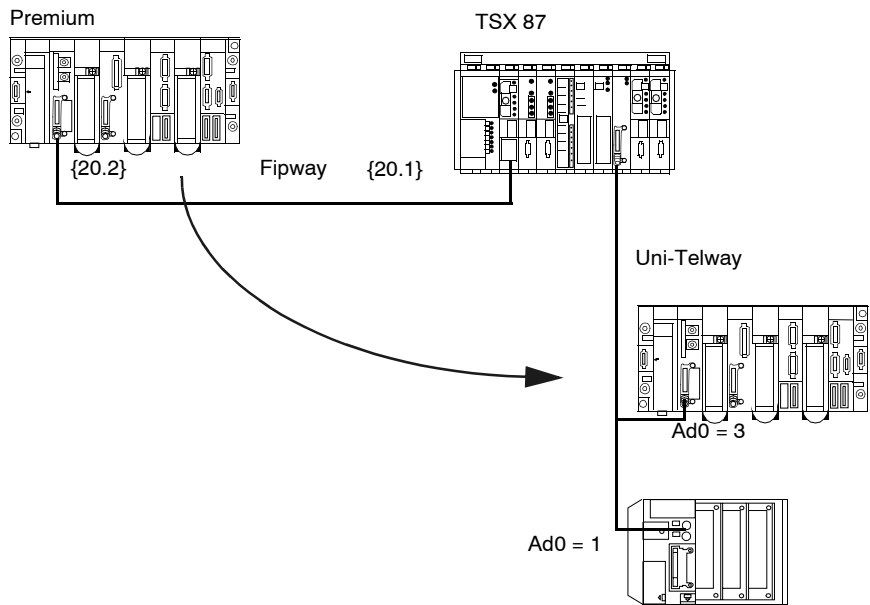
At a Glance

Station 2 of network 20 must send an identification request to the device with address Ad0=3 on the Uni-Telway bus of station 1 of the same network. The identification request has the decimal code 15 (or 16#0F).

The management table is located at %MW10 : 4.

Illustration

The two stations are connected via a Fipway network:



Programming

Programming in ST:

```
IF RE(%I0.3.2) AND NOT %MW10.0 THEN
  SEND_REQ(ADDR(' {20.1}0.5.1.3' ), 15, %MW0:1,
  %MW10:4, %MW100:24);
END_IF;
```

Request parameters:

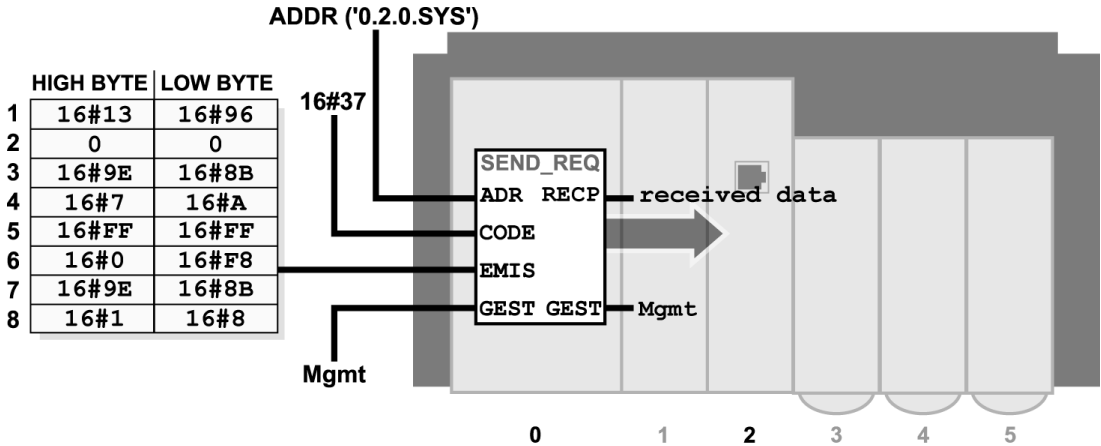
Parameters	Description
ADDR('{20.1}0.5.1.3')	<ul style="list-style-type: none"> ● {20.1}: network 20, station 1 ● 0: rack ● 5: module ● 1: channel 1 ● 3: destination address
15	Request 15 (or 16#0F if the coding is hexadecimal)
%MW0:1	Data sent (for example: no data to send)
%MW10:4	Management table
%MW100:24	Contents of the response (reception of 24 words)

Note: Each time the function is launched, initialize the length parameter (in the example: %MW13 = 0).

Example: Changing IP Parameters with SEND_REQ

Illustration

The graphic shows you how to set IP parameters for the ETY module in slot 2 with the SEND_REQ block:



Note:

- ADR: reflects the ETY's position in slot 2.
- CODE: reflects the value of the REQUEST_CODE.
- EMIS: contains the IP parameters in Data_to_Send:
 - address (139.158.10.7)
 - subnet mask (255.255.248.0)
 - gateway (139.158.8.1)
- GEST: reflects Management_Param (management parameters). You have to assign a time to the third word of Management_Param. The fourth word should have the INT value 18.
- RECP: This parameter requires a minimum INT value of 1, even when no response message is returned, as in the case of an IP change request..

Using the SEND_REQ Function

At a Glance

The `SEND_REQ` function is used to code and send all UNI-TE and Modbus/Jbus requests, as well as receive the associated responses.

In certain cases (reading of word tables for example) it is necessary to resequence the objects received by using the `ROR1_ARB` function (shift of a byte in a table). .

Example

Objects to read:

16#0201

16#0403

16#0605

16#0807

16#0A09

Reception table after execution of a SEND_REQ (read object):

%MW100=16#0107

%MW101=16#0302

%MW102=16#0504

%MW103=16#0706

%MW104=16#0908

%MW105=16#000A

Reception table after a ROR1_ARB(%MW100:6):

%MW100=16#0201

%MW101=16#0403

%MW102=16#0605

%MW103=16#0807

%MW104=16#0A09

%MW105=16#0700

Nachtrag

Dynamischen IP Adressfunktion
der Premium Ethernet Module

TSX ETY 4103 / PORT / 5103

3/2005

SEND_REQ: Senden von Requests

Dieses Dokument enthält die folgenden Themen:

Thema	Seite
Beschreibung	4
Liste der UNI-TE-Requests	6
Fenster zur unterstützten Eingabe	11
Beispiel für das Senden eines UNI-TE-Requests	13
Beispiel: Ändern der IP-Parameter mittels SEND_REQ	15
Mittels der Funktion SEND_REQ	16

Beschreibung

Funktions- beschreibung

Die Funktion `SEND_REQ` wird verwendet, um alle UNI-TE- und Modbus/Jbus-Requests zu codieren und zu senden und um die zugehörigen Antworten zu empfangen.

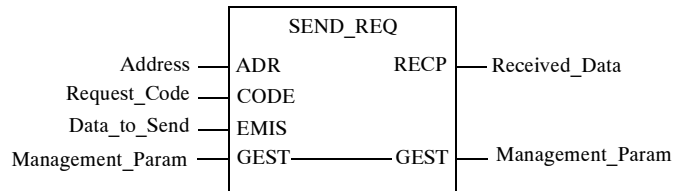
Ausführliche Informationen zur Codierung von UNI-TE-Requests finden Sie im Referenzhandbuch: Kommunikation TSX DR NET.

Ausführliche Informationen zur Codierung der Modbus/Jbus-Requests finden Sie im TSX DG MDB-Handbuch.

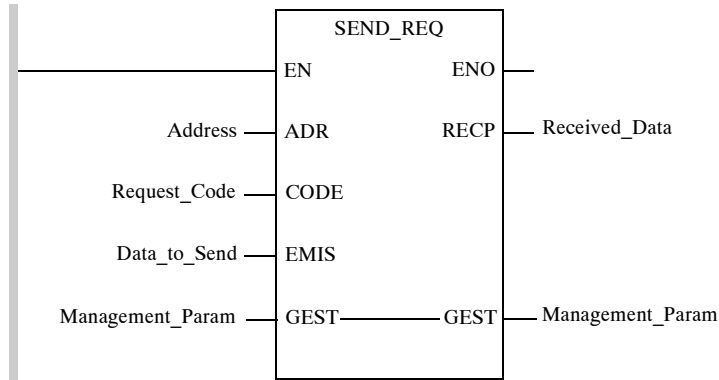
Für alle Schneider-Geräte gültige Modbus-Requests sind im Handbuch **Architektur und Kommunikationsdienste** beschrieben.

Die zusätzliche Parameter `EN` und `ENO` können konfiguriert werden.

Darstellung in FBD



Darstellung in LD



Darstellung in IL

Darstellung:

LD-Adresse

`SEND_REQ Request_Code, Data_to_Send, Management_Param, Received_Data`

Darstellung in ST Darstellung:
`SEND_REQ(Address, Request_Code, Data_to_Send,
Management_Param, Received_Data);`

Parameter- beschreibung

In der folgenden Tabelle sind die Eingangsparameter beschrieben.

Parameter	Typ	Kommentar
Adresse	ARRAY [0.. 5] OF INT	Adresse des Zielgeräts des Austauschs. Der Adresstyp hängt vom gesendeten Request ab. Es ist daher beispielsweise möglich, den RUN-Request (ALL) rund zu senden, während es nicht möglich ist, einen Identifikationsrequest gleichzeitig an verschiedene Geräte zu senden.
Request_Code	INT	An das Zielgerät, auch Server genannt, zu sendender Request. Requests können UNI-TE-Requests (Siehe <i>Liste der UNI-TE-Requests</i> , S. 6) oder Modbus-Requests sein.
Data_to_Send	ARRAY [n... m] OF INT	An das Zielgerät des Requests zu sendende Ganzzahltablette. Diese Tabelle hängt vom gesendeten Request ab. Sie muss eine minimale Länge von 1 Element haben, selbst wenn der verwendete Request keinen Datenversand erfordert (zum Beispiel ein Identifikationsrequest). Hinweis: Die Länge der zu sendenden Daten (in Bytes) muss unbedingt dem vierten Wort der Verwaltungstabelle zugeordnet werden, bevor die Funktion gestartet wird, damit diese richtig ausgeführt wird.

In der folgenden Tabelle sind die Ein-/Ausgangsparameter beschrieben.

Parameter	Typ	Kommentar
Management_Param	ARRAY [0.. 3] OF INT	Verwaltungstabelle des Austauschs

In der folgenden Tabelle sind die Ausgangsparameter beschrieben.

Parameter	Typ	Kommentar
Received_Data	ARRAY [n... m] OF INT	Ganzzahltablette, die die Daten enthält, die vom Server -Gerät, das der Empfänger des Requests gewesen ist, zurückgesendet wurden. Obwohl bestimmte Requests keine Antwort erfordern (zum Beispiel ein RUN-Request), ist es dennoch erforderlich, bei jeder Verwendung der Funktion <code>SEND_REQ</code> eine Tabelle mit mindestens einer Ganzzahl zu reservieren. Hinweis: Die Größe der empfangenen Daten wird vom System automatisch in das vierte Wort der Verwaltungstabelle geschrieben. Hinweis: In bestimmten Fällen (Lesen von Worttabellen beispielsweise) ist es erforderlich, die empfangenen Objekte mittels der Funktion <code>ROR1_ARB</code> (Byteverschiebung in einer Tabelle) in eine neue Reihenfolge zu bringen.

Liste der UNI-TE-Requests

Auf einen Blick

Das UNI-TE-Protokoll wird verwendet, um:

- alle Geräte, die über einen UNI-TE-Server verfügen, zu identifizieren und zu diagnostizieren,
- eine Reihe von Diensten anzubieten, die einen Schreib-/Lesezugriff auf Typdaten bieten,
- Daten von einem Gerät in ein anderes Gerät zu laden,
- einen Server gegen konkurrierende Verbindungen während einer kritischen Periode zu schützen.

Diese verschiedenen Dienste können mittels der Funktion `SEND_REQ` mit einer Codierung des zu sendenden UNI-TE-Requests erreicht werden.

Hinweis: Ausführliche Informationen sowie die Liste der von jedem Gerät erkannten Requests finden Sie im TSX DR NET-Referenzhandbuch.

Die folgenden Tabellen enthalten nicht erschöpfende Listen der von Premium-Geräten erkannten Requests.

Requests zur allgemeinen Verwendung

Diese Requests werden verwendet, um alle Gerätetypen, die über einen UNI-TE-Server verfügen, zu identifizieren und zu diagnostizieren:

Requestname	Requestcode	Berichtscod	Kommentar
IDENTIFICATION	16#0F	16#3F	Bietet die folgenden Informationen: <ul style="list-style-type: none"> • Produktbereich • spezifischer Anwendungstyp • Produkttyp • Katalog-Artikelnummer
READ_CPU	16#4F	16#7F	Führt eine Systemdiagnose an jedem Gerät durch.
PROTOCOL_VERSION	16#30	16#60	Wird verwendet, um die Version des Protokolls zwischen zwei miteinander kommunizierenden Einheiten anzupassen.
MIRROR	16#FA	16#FB	Testet die richtige Leitweglenkung der Daten zwischen zwei miteinander kommunizierenden Geräten.

Dynamische Änderung von IP-Parametern

CHANGE IP PARAMETERS

Requestname	Requestcode	Kommentar
REQUEST CODE	16#37	Requestcode-Funktion
CHANGE IP PARAMETERS	16#13	Ändert die IP-Parameter-Unterfunktion

Um zu gewährleisten, dass das ETY-Modul betriebsbereit ist, lassen Sie die SPS 15 Sekunden nach dem letzten Stopp-Modus laufen (im Modus RUN), bevor Sie die Funktion SEND_REQ ausführen. Nachdem der Benutzer den Befehl CHANGE IP PARAMETERS ausführt und das ETY-Modul die neuen Parameter annimmt, wird das ETY-Modul zurückgesetzt, und es beginnt den Betrieb mit den neuen Parametern.

Hinweis: FDR-Clients, die das ETY-Modul für einen Server verwenden, müssen nach Abschluss der IP-Adressänderung des ETY-Moduls neu hochgefahren werden. Andernfalls können diese FDR-Clients nicht ihre Parameterdateien auf dem FDR-Server (dem ETY-Modul) aktualisieren.

Hinweis: Konstantwörter enthalten die Original-Konfigurationsparameter und nicht die, die nach der IP-Adressänderung aktualisiert wurden.

Hinweis: Sie können die neue Konfiguration (IP-Parameter, Subnetzmaske und Gatewayadresse) im Unity Pro ETY-Debugfenster sehen. Sie können die neu zugewiesene IP-Adresse auch auf der Web-Seite zu den Ethernet-Modulstatistiken sehen. Sie müssen sich jedoch bewusst sein, dass die IP-Adresse, die Subnetzmaske und die Gateway-Daten auf dieser Seite der alten Konfiguration entsprechen.

Parameterverwendung für die IP-Adressänderung

Diese Tabelle basiert auf der IP-Beispieladresse 139.158.10.7, der Beispiel-Subnetzmaske 255.255.248.0 und einer Gateway-Beispieladresse von 139.158.8.1.

Parameter	Typ	Wert	Kommentar
ADDRESS	Bereich [0 - 5] von INT	ADDR ('Rack.Steckplatz.Kanal.SYS')	Beispiel: ADDR (0.x.0.SYS') x = Steckplatz, in den das ETY- Modul eingesteckt ist
REQUEST_ CODE	INT	16#37	

Parameter	Typ	Wert	Kommentar
Data_to_send	Bereich [0 - 8] von INT	Byte 1: Unterfunktion (13h)	Hochwertiges Byte
		Byte 2: Unterfunktion (96h)	Niederwertiges Byte
		Byte 3: 0	Wert wird ignoriert.
		Byte 4: 0	Wert wird ignoriert.
		Byte 5: IP-Adresse 2 (158)	1-239 (höherwertiges Byte)
		Byte 6: IP-Adresse 1 (139)	0-255 (niederwertiges Byte)
		Byte 7: IP-Adresse 4 (7)	0-255 (höherwertiges Byte)
		Byte 8: IP-Adresse 3 (10)	0-255 (niederwertiges Byte)
		Byte 9: Subnetzmaske 2 (255)	255
		Byte 10: Subnetzmaske 1 (255)	0-255
		Byte 11: Subnetzmaske 4 (0)	0-255
		Byte 12: Subnetzmaske 3 (248)	0-255
		Byte 13: Gateway 2 (158)	1-239 (Die Gatewayadresse muss sich im selben Teilnetz wie die IP-Adresse befinden.)
		Byte 14: Gateway 1 (139)	0-255
		Byte 15: Gateway 4 (1)	0-255
		Byte 16: Gateway 3 (8)	0-255
Manage_Param	Wortnummer	Hochwertiges Byte	Niederwertiges Byte
	1	Aktivitätsbericht	00
	2	Betriebsbericht (siehe Hinweis 1)	Kommunikationsbericht (siehe Hinweis 1)
	3	Timeout (ms)	
	4	18 (INT)	
	Hinweis 1: Die folgende Tabelle enthält Details zu den IP-Adresscodes (erfolgreich oder Fehler)		

Siehe das Beispiel für die Änderung der IP-Parameter mittels SEND_REQ (Siehe *Beispiel: Ändern der IP-Parameter mittels SEND_REQ*, S. 15).

Ändern der IP-Adresscodes

Betriebsbericht	Kommunikationsbericht	Bedeutung
<i>Erfolgscodes</i>		
FE (hex)	00 (hex)	Der Senderequest (SEND_REQ) hat die IP-Adresse erfolgreich geändert.
<i>Fehlercodes</i>		
01 (hex)	FF (hex)	Ungültiger Requestcodewert (z. B. nicht 16#37)
00 (hex)	03 (hex)	Senderequest an die IP-Adresse des ETY-Moduls anstatt über den Baugruppenträger
00 (hex)	07 (hex)	Falsche Adresszuordnung zum ETY-Modul
16 (hex)	FF (hex)	Ungültige IP-Adresse
17 (hex)	FF (hex)	Ungültiges Teilnetz
18 (hex)	FF (hex)	Ungültige Gateway-Adresse
19 (hex)	EF (hex)	Ungültige Netzwerkadresse
1A (hex)	FF (hex)	Die IP-Adresse des ETY-Moduls war bereits auf Senderequest gesetzt (SEND_REQ).
FD (hex)	00 (hex)	Der Senderequest (SEND_REQ) hat die IP-Adresse nicht erfolgreich geändert.

Zugreifen auf Objekte

Diese Requests bieten eine Reihe von Diensten, die einen Schreib-/Lesezugriff auf interne Bit- und Wortdaten, Systembit- und Systemwortdaten, Gleitkommatdaten, Konstantdaten und SFC-Daten bieten.

Requestname	Requestcode	Berichtscod	Kommentar
READ_OBJECT	16#36	16#66	Verwendet, um ein oder mehrere aufeinander folgende Objekte desselben Typs zu lesen.
WRITE_OBJECT	16#37	16#FE	Verwendet, um ein oder mehrere aufeinander folgende Objekte desselben Typs zu schreiben.
READ_INTERNAL_BIT	16#00	16#30	Verwendet, um den Wert eines internen Bits zu lesen.
WRITE_INTERNAL_BIT	16#10	16#FE	Verwendet, um den Wert eines internen Bits zu schreiben.
READ_INTERNAL_WORD	16#04	16#34	Verwendet, um den Wert eines internen Worts zu lesen.
WRITE_INTERNAL_WORD	16#14	16#FE	Verwendet, um den Wert eines internen Worts zu schreiben.

Verwaltung der Betriebsarten

Diese Requests bieten eine Reihe von Diensten, die verwendet werden können, um die Betriebsarten eines Prozessors zu verwalten.

Requestname	Requestcode	Berichtscod	Kommentar
RUN	16#24	16#FE	Verwendet, um die Ausführung von Tasks eines Prozessors zu starten.
STOP	16#25	16#FE	Verwendet, um die Ausführung von Tasks eines Prozessors zu stoppen.
INIT	16#33	16#63	Verwendet, um einen Warm- oder Kaltstart auszulösen.

Reservierungsverwaltung

Diese Requests bieten einen Reservierungsmechanismus, der verwendet wird, um einen Server während einer kritischen Phase gegen konkurrierende Verbindungen zu schützen.

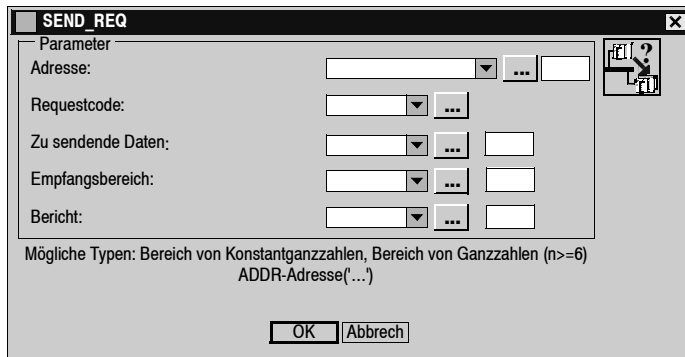
Requestname	Requestcode	Berichtscod	Kommentar
RESERVE	16#1D	16#FE	Ermöglicht es einem Client, die Funktionen eines Servers allgemein oder teilweise zu reservieren.
RELEASE	16#1E	16#FE	Ermöglicht es einem Client, die Reservierung eines Servers aufzuheben.
I_AM_ALIVE	16#2D	16#FE	Wird verwendet, um die Reservierung aufrecht zu erhalten.

Fenster zur unterstützten Eingabe

Auf einen Blick Für diese Kommunikationsfunktion können Sie den das Fenster zur unterstützten Eingabe aufrufen.

Hinweis: Die Symbole werden akzeptiert.

Abbildung Die folgende Abbildung zeigt ein Beispiel für das Fenster zur unterstützten Eingabe der Funktion:



Adresse

Mögliche Objekttypen sind:

- ADDR (ZEICHENFOLGE).
- BEREICH [0 - 5] VON GANZZAHL.

Hinweis: Wenn Sie einen Wert direkt in das Feld eingeben, wird die Schaltfläche zur unterstützten Adresseingabe grau angezeigt.

Requestcode

Mögliche Objekte des Typs GANZZAHL sind:

- Variablen,
- Konstanten,
- Sofortwert.

Hinweis: Wenn Sie eine Konstante eingeben, wird ein Eingabefeld zur Eingabe der Konstanten angezeigt. Wenn Sie eine Variable eingeben, kann diese lokalisiert sein oder nicht.

Zu sendende Daten	Diese zu sendenden Daten werden in Form eines Ganzzahlbereichs gespeichert. Dieser Bereich kann lokalisiert sein oder nicht.
Empfangsbereich	Der Empfangsbereich ist ein Ganzzahlbereich. Dieser Bereich kann lokalisiert sein oder nicht, und seine Größe hängt vom verwendeten Requestcode ab.
Bericht	Der Bericht ist ein Bereich von 4 Ganzzahlen.

Hinweis: Achten Sie darauf, nicht verschiedene identische Speicherbereiche für Berichtstabellen zu verwenden, da andernfalls die Variablen-Lesefunktion möglicherweise nicht funktioniert.

Beispiel für das Senden eines UNI-TE-Requests

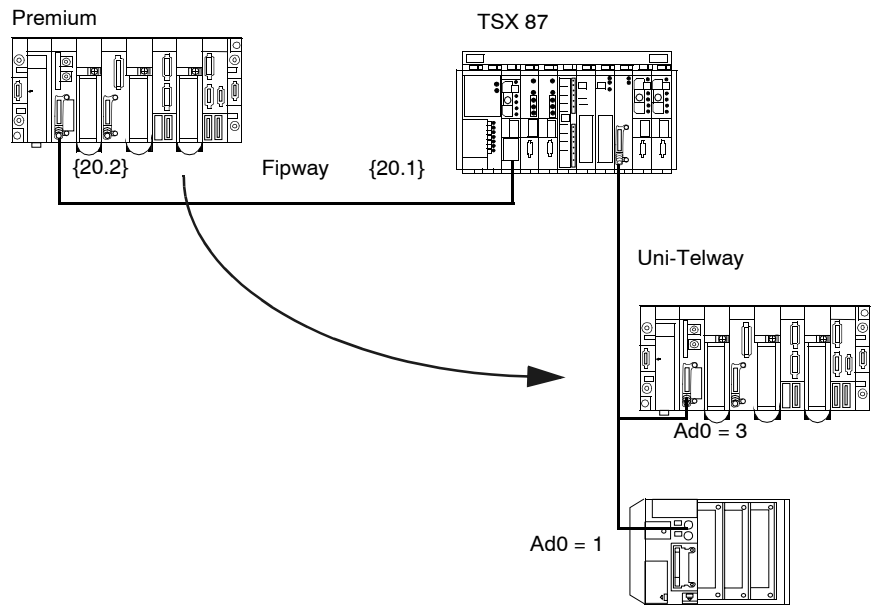
Auf einen Blick

Station 2 von Netzwerk 20 muss einen Identifikationsrequest an das Gerät mit der Adresse Ad0=3 auf dem Uni-Telway-Bus der Station 1 in demselben Netzwerk senden. Der Identifikationsrequest hat den Dezimalcode 15 (oder 16#0F).

Die Verwaltungstabelle befindet sich an $\%MW10:4$.

Abbildung

Die beiden Stationen sind über ein Fipway-Netzwerk miteinander verbunden:



Programmierung Programmierung in ST:

```
IF RE(%IO.3.2) AND NOT %MW10.0 THEN
    SEND_REQ(ADDR(' {20.1}0.5.1.3' ),15,%MW0:1,
    %MW10:4,%MW100:24);
END_IF;
```

Requestparameter:

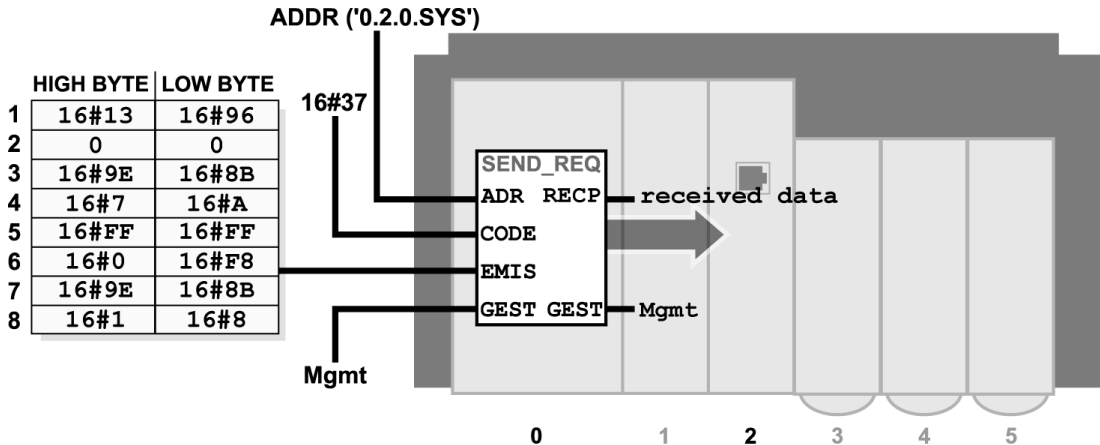
Parameter	Beschreibung
ADDR('{20.1}0.5.1.3')	<ul style="list-style-type: none"> ● {20.1}: Netzwerk 20, Station 1 ● 0: Rack ● 5: Modul ● 1: Kanal 1 ● 3: Zieladresse
15	Request 15 (oder 16#0F bei hexadezimaler Codierung)
%MW0:1	Gesendete Daten (z. B.: keine zu sendenden Daten)
%MW10:4	Verwaltungstabelle
%MW100:24	Inhalte der Antwort (Empfang von 24 Wörtern)

Hinweis: Initialisieren Sie jedes Mal, wenn die Funktion gestartet wird, den Längenparameter (im Beispiel: %MW13 = 0).

Beispiel: Ändern der IP-Parameter mittels SEND_REQ

Abbildung

Die nachfolgende Abbildung veranschaulicht, wie die IP-Parameter für das ETY-Modul in Steckplatz 2 mit dem Baustein SEND_REQ geändert werden.



Hinweis:

- ADDR: gibt die Position des ETY-Moduls in Steckplatz 2 an.
- CODE: gibt den Wert des REQUEST_CODE an.
- EMIS: enthält die IP-Parameter in Data_to_Send:
 - Adresse (139.158.10.7)
 - Subnetzmaske (255.255.248.0)
 - Gateway (139.158.8.1)
- GEST: gibt Management_Param (Managementparameter) an. Sie müssen dem dritten Wort von Management_Param eine Zeit zuweisen. Das vierte Wort muss den INT-Wert 18 haben.
- RECP: Dieser Parameter erfordert einen minimalen INT-Wert von 1, selbst wenn keine Antwortmeldung zurückgesendet wird, wie dies bei einem IP-Änderungsrequest der Fall ist.

Mittels der Funktion SEND_REQ

Auf einen Blick Die Funktion `SEND_REQ` wird verwendet, um alle UNI-TE- und Modbus/Jbus-Requests zu codieren und zu senden und um die zugehörigen Antworten zu empfangen.

In bestimmten Fällen (Lesen von Worttabellen beispielsweise) ist es erforderlich, die empfangenen Objekte mittels der Funktion `ROR1_ARB` (Byteverschiebung in einer Tabelle) in eine neue Reihenfolge zu bringen. .

Beispiel

Zu lesende Objekte:

16#0201
16#0403
16#0605
16#0807
16#0A09

Empfangstabelle nach Ausführung eines `SEND_REQ` (Objekt lesen):

%MW100=16#0107
%MW101=16#0302
%MW102=16#0504
%MW103=16#0706
%MW104=16#0908
%MW105=16#000A

Empfangstabelle nach einem `ROR1_ARB(%MW100:6)`:

%MW100=16#0201
%MW101=16#0403
%MW102=16#0605
%MW103=16#0807
%MW104=16#0A09
%MW105=16#0700

Addendum

La Fonction Adresse IP Dynamique
pour les Modules Ethernet Premium

TSX ETY 4103 / PORT / 5103

3/2005

SEND_REQ: envoi de requêtes

Ce document contient les sujets suivants :

Sujet	Page
Description	4
Liste des requêtes UNI-TE	6
Ecran de saisie assistée	11
Exemple d'envoi d'une requête UNI-TE	13
Exemple : Modification des paramètres IP avec SEND_REQ	15
Utilisation de la fonction SEND_REQ	16

Description

Description de la fonction

La fonction `SEND_REQ` sert à coder et à envoyer toutes les requêtes UNI-TE et Modbus/Jbus et à recevoir les réponses correspondantes.

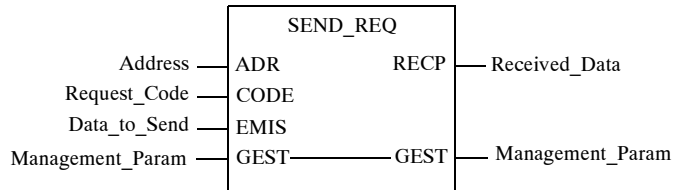
Les détails de codage des requêtes UNI-TE sont fournis dans le manuel de référence : Référence Communication TSX DR NET.

Quant au codage des requêtes Modbus/Jbus, il est expliqué dans le manuel TSX DG MDB.

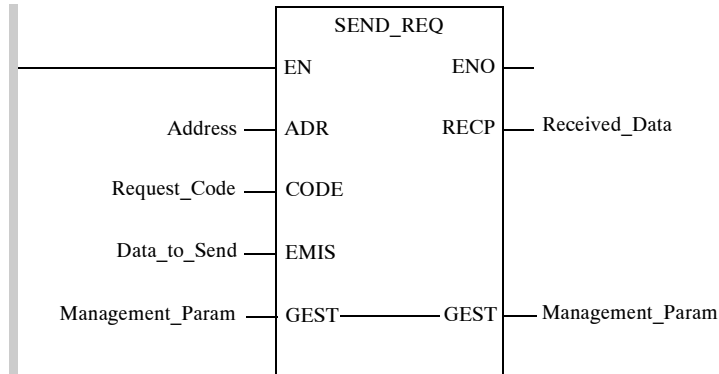
Les requêtes Modbus communes à tous les équipements Schneider sont présentées dans le manuel **Architecture et services de communication**.

Les paramètres `EN` et `ENO` supplémentaires peuvent être configurés.

Représentation FBD



Représentation LD



Représentation IL

Représentation :
 LD Address
 SEND_REQ Request_Code, Data_to_Send, Management_Param,
 Received_Data

**Représentation
ST**

Représentation :
 SEND_REQ(Address, Request_Code, Data_to_Send,
 Management_Param, Received_Data);

**Description des
paramètres**

Le tableau suivant décrit les paramètres d'entrée :

Paramètre	Type	Commentaire
Address	ARRAY [0.. 5] OF INT	Adresse de l'entité destinataire de l'échange. Ce type d'adresse dépend de la requête envoyée. Il est donc possible, par exemple, de diffuser la requête RUN (ALL), alors qu'il est impossible d'envoyer une requête d'identification simultanément à plusieurs équipements.
Request_Code	INT	Requête à envoyer à l'équipement destinataire, également appelé serveur . Il peut s'agir de requêtes UNI-TE (Voir <i>Liste des requêtes UNI-TE, p. 6</i>) ou de requêtes Modbus.
Data_to_Send	ARRAY [n... m] OF INT	Table d'entiers à envoyer à l'équipement destinataire de la requête. Cette table dépend de la requête envoyée. Sa longueur minimum doit être d'au moins 1 élément, même si la requête utilisée ne nécessite pas d'envoi de données (par exemple, requête d'identification). Remarque : Il est impératif d'affecter la longueur des données à envoyer (en octets) au quatrième mot de la table de gestion avant de lancer la fonction, pour que celle-ci soit exécutée correctement.

Le tableau suivant décrit les paramètres d'entrée/de sortie :

Paramètre	Type	Commentaire
Management_Param	ARRAY [0.. 3] OF INT	Table de gestion d'échange

Le tableau suivant décrit les paramètres de sortie :

Paramètre	Type	Commentaire
Received_Data	ARRAY [n... m] OF INT	Table d'entiers contenant les données renvoyées par l'équipement serveur destinataire de la requête. Même si certaines requêtes ne nécessitent pas de réponse (comme dans le cas d'une requête Run), il est toutefois nécessaire de réserver une table comportant au moins 1 entier chaque fois que la fonction SEND_REQ est utilisée. Remarque : La taille des données reçues est automatiquement écrite par le système dans le quatrième mot de la table de gestion. Remarque : Dans certains cas (lors de la lecture de tables de mots, par exemple), il est nécessaire de reséquencez les objets reçus à l'aide de la fonction ROR1_ARB (décalage d'un octet dans une table).

Liste des requêtes UNI-TE

Présentation

Le protocole UNI-TE est utilisé pour :

- identifier et diagnostiquer tous les équipements disposant d'un serveur UNI-TE ;
- fournir un ensemble de services donnant l'accès en lecture/écriture aux données de type ;
- télécharger les données d'un équipement à un autre ;
- protéger un serveur contre les connexions concurrentes en période critique.

Ces différents services sont disponibles grâce à la fonction `SEND_REQ`, moyennant le codage de la requête UNI-TE à envoyer.

Note : Pour obtenir des informations détaillées et la liste des requêtes reconnues par chaque équipement, reportez-vous au manuel de référence TSX DR NET.

Les tableaux suivants fournissent une liste non-exhaustive des requêtes reconnues par les équipements Premium.

Requêtes d'utilisation générale

Ces requêtes sont utilisées pour identifier et diagnostiquer tous les types d'équipement disposant d'un serveur UNI-TE :

Nom de requête	Code de requête	Code de rapport	Commentaire
IDENTIFICATION	16#0F	16#3F	Fournit des informations sur : <ul style="list-style-type: none"> ● la gamme de produits ; ● le type d'application spécifique ; ● le type de produit ; ● la référence catalogue.
READ_CPU	16#4F	16#7F	Réalise un diagnostic système sur n'importe quel équipement.
PROTOCOL_VERSION	16#30	16#60	Sert à adapter la version du protocole entre deux équipements communicants.
MIRROR	16#FA	16#FB	Teste le routage correct des données entre deux équipements communicants.

**Modification
dynamique des
paramètres IP****CHANGE IP PARAMETERS**

Nom de requête	Code de requête	Commentaire
REQUEST_CODE	16#37	Fonction de codage d'une requête.
CHANGE_IP_PARAMETERS	16#13	Modifie la sous-fonction des paramètres IP.

Pour garantir que le module ETY est prêt à fonctionner, laissez l'automate s'exécuter (en mode Run) pendant 15 secondes après le dernier arrêt avant de lancer la fonction SEND_REQ. Après le lancement par l'utilisateur de la commande CHANGE_IP_PARAMETERS et l'acceptation par le module ETY des nouveaux paramètres, le module ETY est réinitialisé et lance les opérations en fonction des nouveaux paramètres.

Note : Les clients FDR qui utilisent le module ETY comme serveur doivent le redémarrer après une modification de l'adresse IP du module ETY. Sinon, ces clients FDR ne peuvent pas mettre à jour leurs fichiers de paramètres sur le serveur FDR (le module ETY).

Note : Les mots constants contiennent les paramètres de configuration d'origine et non ceux ayant été mis à jour après modification de l'adresse IP.

Note : Vous trouverez la nouvelle configuration (paramètres IP, masque de sous-réseau et adresse de la passerelle) sur l'écran de débogage Unity Pro ETY. Vous trouverez également la nouvelle configuration d'adresse IP dans la page Web des statistiques du module Ethernet. Notez toutefois que les données d'adresse IP, de masque de sous-réseau et de passerelle de cette page reflètent l'ancienne configuration.

Utilisation des paramètres en cas de modification de l'adresse IP

Ce tableau est basé sur un exemple avec l'adresse IP 139.158.10.7, le masque de sous-réseau 255.255.248.0 et l'adresse de passerelle 139.158.8.1.

Paramètre	Type	Valeur	Commentaire
ADDRESS	array[0...5] of INT	ADDR (<code>'rack.slot.channel.SYS'</code>)	Exemple : ADDR (0.x.0.SYS) x = emplacement dans lequel le module ETY est installé.
REQUEST_CODE	INT	16#37	
Data_to_Send	array[0...8] of INT	octet 1 : sous-fonction (13h)	octet de poids fort
		octet 2 : sous-fonction (96h)	octet de poids faible

Paramètre	Type	Valeur	Commentaire
		octet 3 : 0	La valeur est ignorée.
		octet 4 : 0	La valeur est ignorée.
		octet 5 : adresse IP 2 (158)	1-239 (octet de poids fort)
		octet 6 : adresse IP 1 (139)	0-255 (octet de poids faible)
		octet 7 : adresse IP 4 (7)	0-255 (octet de poids fort)
		octet 8 : adresse IP 3 (10)	0-255 (octet de poids faible)
		octet 9 : masque de sous-réseau 2 (255)	255
		octet 10 : masque de sous-réseau 1 (255)	0-255
		octet 11 : masque de sous-réseau 4 (0)	0-255
		octet 12 : masque de sous-réseau 3 (248)	0-255
		octet 13 : passerelle 2 (158)	1-239 (l'adresse de la passerelle doit se trouver sur le même sous-réseau que l'adresse IP.)
		octet 14 : passerelle 1 (139)	0-255
		octet 15 : passerelle 4 (1)	0-255
		octet 16 : passerelle 3 (8)	0-255
Manage_Param	numéro du mot	octet de poids fort	octet de poids faible
	1	rapport d'activité	00
	2	rapport d'opération (voir remarque 1)	rapport de communication (voir remarque 1)
	3	temporisation (ms)	
	4	18 (INT)	
	Remarque 1 : Le tableau suivant fournit des informations détaillées sur les codes d'adresse IP (corrects et incorrects).		

Reportez-vous à l'exemple de la section Modification des paramètres IP avec SEND_REQ (Voir *Exemple : Modification des paramètres IP avec SEND_REQ*, p. 15).

Modification des codes d'adresse IP

Rapport d'opération	Rapport de communication	Signification
<i>code correct</i>		
FE (hex)	00 (hex)	La requête envoyée (SEND_REQ) a permis de modifier correctement l'adresse IP.
<i>codes d'erreur</i>		
01 (hex)	FF (hex)	Valeur de code de requête incorrecte (par exemple, différente de 16#37)
00 (hex)	03 (hex)	Requête envoyée à l'adresse IP ETY au lieu d'être envoyée sur l'embase
00 (hex)	07 (hex)	Mappage d'adresse vers le module ETY incorrect
16 (hex)	FF (hex)	Adresse IP incorrecte
17 (hex)	FF (hex)	Sous-réseau incorrect
18 (hex)	FF (hex)	Adresse de passerelle incorrecte
19 (hex)	EF (hex)	Adresse réseau incorrecte
1A (hex)	FF (hex)	Adresse IP du module ETY déjà réglée pour envoyer une requête (SEND_REQ)
FD (hex)	00 (hex)	La requête envoyée (SEND_REQ) n'a pas permis de modifier l'adresse IP.

Accès aux objets

Ces requêtes fournissent un ensemble de services qui permettent l'accès en lecture/écriture aux données de bit interne et de mot-type, aux données de virgule flottante, aux données constantes et aux données SFC.

Nom de requête	Code de requête	Code de rapport	Commentaire
READ_OBJECT	16#36	16#66	Sert à lire un ou plusieurs objets consécutifs et du même type.
WRITE_OBJECT	16#37	16#FE	Sert à écrire un ou plusieurs objets consécutifs et du même type.
READ_INTERNAL_BIT	16#00	16#30	Sert à lire la valeur d'un bit interne.
WRITE_INTERNAL_BIT	16#10	16#FE	Sert à écrire la valeur d'un bit interne.
READ_INTERNAL_WORD	16#04	16#34	Sert à lire la valeur d'un mot interne.
WRITE_INTERNAL_WORD	16#14	16#FE	Sert à écrire la valeur d'un mot interne.

Gestion des modes de fonctionnement

Ces requêtes fournissent un ensemble de services servant à gérer les modes de fonctionnement d'un processeur.

Nom de requête	Code de requête	Code de rapport	Commentaire
RUN	16#24	16#FE	Sert à lancer l'exécution des tâches d'un processeur.
STOP	16#25	16#FE	Sert à arrêter l'exécution des tâches d'un processeur.
INIT	16#33	16#63	Sert à déclencher un redémarrage à chaud ou à froid.

Gestion des réservations

Ces requêtes fournissent un mécanisme de réservation servant à protéger un serveur contre des connexions concurrentes en période critique.

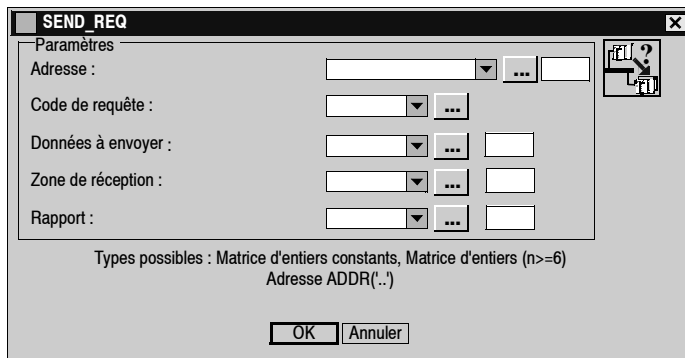
Nom de requête	Code de requête	Code de rapport	Commentaire
RESERVE	16#1D	16#FE	Permet à un client de réserver tout ou partie des fonctions d'un serveur.
RELEASE	16#1E	16#FE	Permet à un client de libérer le serveur réservé.
I_AM_ALIVE	16#2D	16#FE	Sert à maintenir la réservation.

Ecran de saisie assistée

Présentation Pour cette fonction de communication, vous pouvez avoir recours à l'écran de saisie assistée.

Note : Les symboles sont acceptés.

Illustration La capture d'écran suivante est un exemple d'écran de saisie assistée de la fonction :



Adresse Les types d'objet possibles sont les suivants :

- ADDR (STRING)
- ARRAY [0..5] OF INT

Note : Si vous saisissez une valeur directement dans le champ, le bouton de saisie d'adresse assistée est grisé.

Code de requête Les objets possibles sont de type INT :

- Variables
- Constantes
- Valeurs immédiates

Note : Si vous saisissez une constante, un champ de saisie correspondant apparaît. Si vous saisissez une variable, elle peut être localisée ou non.

Données à envoyer Ces données à envoyer sont stockées sous la forme d'une matrice d'entiers. Cette matrice peut être localisée ou non.

Zone de réception La zone de réception est une matrice d'entiers. Celle-ci peut être localisée ou non et sa taille dépend du code de requête utilisé.

Rapport Le rapport est une matrice composée de 4 entiers.

Note : Veillez à ne pas utiliser plusieurs zones de mémoire identiques pour les tables de rapports, sinon les fonctions de lecture des variables risquent de ne pas fonctionner.

Exemple d'envoi d'une requête UNI-TE

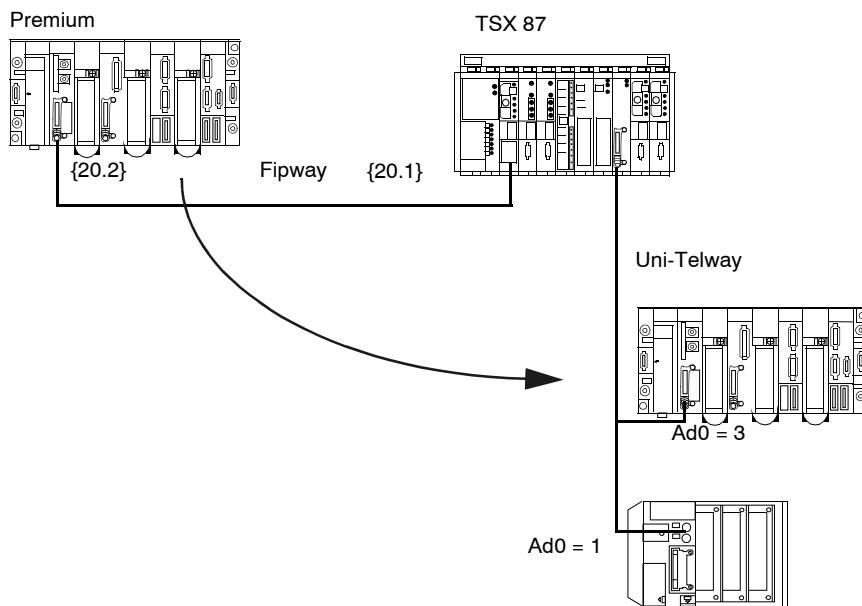
Présentation

La station 2 du réseau 20 doit envoyer une requête d'identification à l'équipement portant l'adresse Ad0=3 sur le bus Uni-Telway de la station 1 du même réseau. La requête d'identification porte le code décimal 15 (ou 16#0F).

La table de gestion se trouve à l'adresse %MW10 : 4.

Illustration

Les deux stations sont connectées via un réseau Fipway :



Programmation Programmation sous ST :

```
IF RE(%IO.3.2) AND NOT %MW10.0 THEN
    SEND_REQ(ADDR(' {20.1}0.5.1.3' ),15,%MW0:1,
    %MW10:4,%MW100:24);
END_IF;
```

Paramètres de la requête :

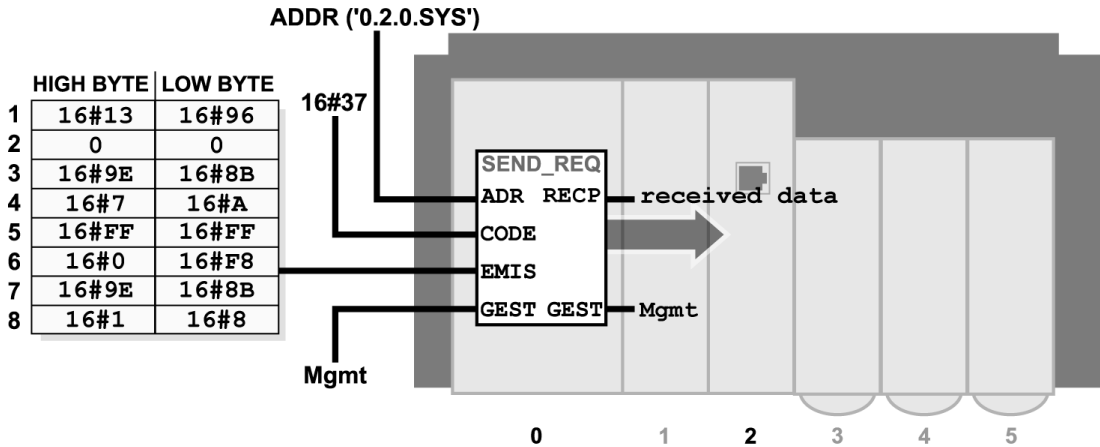
Paramètres	Description
ADDR('{20.1}0.5.1.3')	<ul style="list-style-type: none"> ● {20.1} : réseau 20, station 1 ● 0 : rack ● 5 : module ● 1 : voie 1 ● 3 : adresse de destination
15	Requête 15 (ou 16#0F si le codage est hexadécimal)
%MW0:1	Données envoyées (par exemple : aucune donnée à envoyer)
%MW10:4	Table de gestion
%MW100:24	Contenu de la réponse (réception de 24 mots)

Note : A chaque lancement de la fonction, le paramètre de longueur est initialisé (dans l'exemple : %MW13 = 0).

Exemple : Modification des paramètres IP avec SEND_REQ

Illustration

Le graphique ci-dessous vous indique comment régler les paramètres IP du module ETY dans l'emplacement 2 avec le bloc SEND_REQ :



Remarque :

- ADR : indique la position du module ETY dans l'emplacement 2.
- CODE : indique la valeur de REQUEST_CODE.
- EMIS : contient les paramètres IP dans Data_to_Send :
 - Adresse (139.158.10.7)
 - Masque de sous-réseau (255.255.248.0)
 - Passerelle (139.158.8.1)
- GEST : indique Management_Param (paramètres de gestion).
Vous devez attribuer une durée au troisième mot de Management_Param. Le quatrième mot doit avoir la valeur INT 18.
- RECP : ce paramètre requiert une valeur INT minimum de 1, même lorsque aucun message de réponse n'est renvoyé, comme dans le cas d'une demande de modification IP.

Utilisation de la fonction SEND_REQ

Présentation La fonction SEND_REQ sert à coder et à envoyer toutes les requêtes UNI-TE et Modbus/Jbus et à recevoir les réponses correspondantes.

Dans certains cas (lors de la lecture de tables de mots, par exemple), il est nécessaire de reséquencer les objets reçus à l'aide de la fonction ROR1_ARB (décalage d'un octet dans une table). .

Exemple

Objets à lire :

16#0201
16#0403
16#0605
16#0807
16#0A09

Table de réception après exécution d'une fonction SEND_REQ (lecture d'objet) :

%MW100=16#0107
%MW101=16#0302
%MW102=16#0504
%MW103=16#0706
%MW104=16#0908
%MW105=16#000A

Table de réception après ROR1_ARB(%MW100:6) :

%MW100=16#0201
%MW101=16#0403
%MW102=16#0605
%MW103=16#0807
%MW104=16#0A09
%MW105=16#0700

Adición

La función de cambio de la
dirección IP en los Módulos
Ethernet de Premium

TSX ETY 4103 / PORT / 5103

3/2005

SEND_REQ: envío de peticiones

Este documento contiene los siguiente apartados:

Apartado	Página
Descripción	4
Lista de peticiones UNI-TE	6
pantalla de entrada asistida	11
Ejemplo de cómo enviar una petición UNI-TE	13
Ejemplo: Cambio de parámetro IP mediante SEND_REQ	15
Uso de la función SEND_REQ	16

Descripción

Descripción de la función

La función `SEND_REQ` se utiliza para codificar y enviar todas las peticiones UNI-TE y Modbus/Jbus, así como recibir las respuestas asociadas.

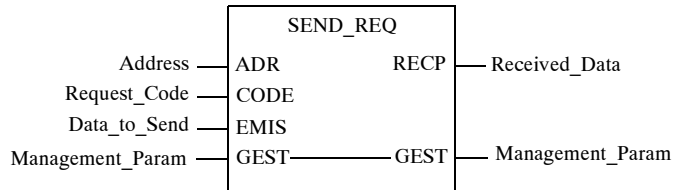
Los detalles de codificación de las peticiones UNI-TE se proporcionan en el manual de referencia de comunicación de TSX DR NET.

Los detalles de codificación de las peticiones Modbus/Jbus se proporcionan en el manual de TSX DG MDB.

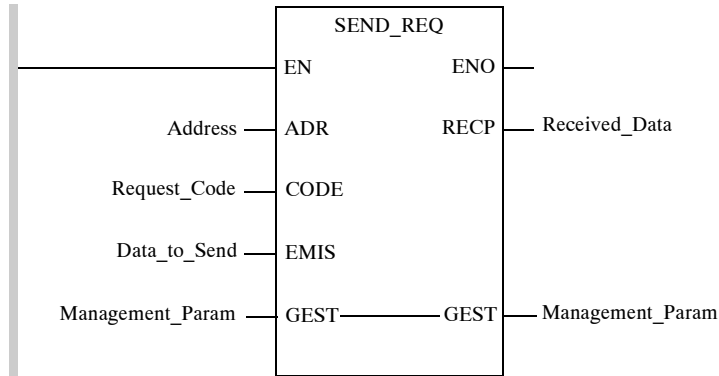
Las peticiones Modbus comunes a todos los dispositivos Schneider se presentan en el manual **Arquitecturas y servicios de comunicación**.

Los parámetros adicionales `EN` y `ENO` pueden configurarse.

Representación FBD



Representación LD



Representación IL

Representación:
 Dirección LD
`SEND_REQ Request_Code, Data_to_Send, Management_Param, Received_Data`

**Representación
ST**

Representación:
SEND_REQ(Address, Request_Code, Data_to_Send,
Management_Param, Received_Data);

**Descripción de
los parámetros**

En la siguiente tabla se describen los parámetros de entrada:

Parámetro	Tipo	Comentario
Dirección	MATRIZ [0.. 5] DE INT	Dirección de la entidad de destino del intercambio. El tipo de la dirección depende de la petición enviada. Por tanto, es posible difundir la petición de ejecución (ALL), mientras que no es posible enviar una petición de identificación a varios dispositivos a la vez.
Request_Code	INT	Petición que se desea enviar al dispositivo de destino, también denominado servidor . Las peticiones pueden ser UNI-TE (Véase <i>Lista de peticiones UNI-TE, p. 6</i>) o Modbus.
Data_to_Send	MATRIZ [n... m] DE INT	Tabla de enteros que se desea enviar al dispositivo de destino de la petición. Esta tabla depende de la petición enviada. Debe tener una longitud mínima de 1 elemento, incluso si la petición utilizada no requiere envío de datos (por ejemplo, una petición de identificación). Nota: Es imprescindible que la longitud de los datos que se desean enviar (en bytes) se asignen a la cuarta palabra de la tabla de gestión antes de iniciar la función para su correcta ejecución.

En la siguiente tabla se describen los parámetros de entrada y salida:

Parámetro	Tipo	Comentario
Management_Param	MATRIZ [0.. 3] DE INT	Tabla de gestión de intercambio

En la siguiente tabla se describen los parámetros de salida:

Parámetro	Tipo	Comentario
Received_Data	MATRIZ [n... m] DE INT	Tabla de enteros que contiene los datos devueltos por el dispositivo servidor , destino de la petición. Aunque algunas peticiones no requieren respuesta, (por ejemplo, una petición de ejecución), sigue siendo necesario reservar una tabla mínima de 1 entero cada vez que se utiliza la función SEND_REQ. Nota: El sistema escribe automáticamente el tamaño de los datos recibidos en la cuarta palabra de la tabla de gestión. Nota: En algunos casos (por ejemplo, la lectura de tablas de palabras) es necesario volver a definir la secuencia de los objetos recibidos mediante la función ROR1_ARB (desplazamiento de un byte en una tabla).

Lista de peticiones UNI-TE

Presentación

Se utiliza el protocolo UNI-TE para:

- Identificar y diagnosticar todos los dispositivos que tienen un servidor UNI-TE.
- Ofrecer un conjunto de servicios que concedan acceso de lectura y escritura para introducir datos.
- Descargar datos de dispositivo a dispositivo.
- Proteger un servidor frente a sus conexiones durante un periodo crítico.

Estos servicios diferentes pueden obtenerse utilizando la función `SEND_REQ`, con la codificación de la petición UNI-TE que se desea enviar.

Nota: Para obtener información detallada y la lista de peticiones reconocidas por cada dispositivo, consulte el manual de referencia de TSX DR NET.

Las tablas siguientes ofrecen una lista no exhaustiva de peticiones reconocidas por los dispositivos Premium.

Peticiones de uso general

Estas peticiones se utilizan para identificar y diagnosticar todos los dispositivos que tienen un servidor UNI-TE:

Nombre de petición	Código de petición	Código de informe	Comentario
IDENTIFICACIÓN	16#0F	16#3F	Ofrece la información siguiente: <ul style="list-style-type: none"> ● Rango de producto ● Tipo de aplicación específica ● Tipo de producto ● Referencia de catálogo
READ_CPU	16#4F	16#7F	Lleva a cabo el diagnóstico de sistema en cualquier dispositivo.
PROTOCOL_VERSION	16#30	16#60	Se utiliza para adaptar la versión del protocolo entre dos entidades comunicantes.
MIRROR	16#FA	16#FB	Comprueba encadenamiento de los datos entre dos dispositivos comunicantes.

Modificación dinámica de parámetros IP

CAMBIO DE PARÁMETROS IP

Nombre de petición	Código de petición	Comentario
REQUEST CODE	16#37	Solicita el código de petición
CHANGE IP PARAMETERS	16#13	Cambia la subfunción de parámetro IP

Para asegurarse que el módulo ETY está listo para funcionar, permita que el UC se ejecute (en modo de ejecución) durante 15 segundos después del último modo de detención antes de enviar la función SEND_REQ. Después de que el usuario envíe el comando CHANGE IP PARAMETERS y de que el módulo ETY acepte los nuevos parámetros, el módulo ETY restablece y comienza las operaciones con los nuevos parámetros.

Nota: Los clientes FDR que utilizan el módulo ETY para un servidor deben reiniciarse después de que se haya terminado de llevar a cabo el cambio de la dirección IP del ETY. De lo contrario, estos clientes FDR no pueden actualizar sus archivos de parámetros en el servidor FDR (el módulo ETY).

Nota: Las palabras constantes contienen los parámetros de configuración originales, no aquellos que se actualizaron después del cambio de dirección IP.

Nota: Puede ver la nueva configuración (parámetros IP, máscara de subred y dirección de camino) en la pantalla de depuración de ETY de Unity Pro. También se puede ver la nueva dirección IP asignada en la página Web de estadística del módulo Ethernet, no obstante, debe tener en cuenta de que la dirección IP, la máscara de subred y los datos de camino de esta página reflejan la configuración antigua.

Uso de parámetros para el cambio de dirección IP

Esta tabla se basa en una dirección IP de muestra de 139.158.10.7, una máscara de subred 255.255.248.0 y una dirección de camino de 139.158.8.1.

Parámetro	Tipo	Valor	Comentario
ADDRESS	matriz [0...5] de INT	ADDR ('rack.slot.channel.SYS')	Ejemplo: ADDR (0.x.0.SYS) x = alojamiento en el que se instala el módulo ETY
REQUEST_CODE	INT	16#37	

Parámetro	Tipo	Valor	Comentario
Data_to_Send	matriz [0...8] de INT	byte 1: subfunción (13 h)	byte alto
		byte 2: subfunción (96 h)	byte bajo
		byte 3: 0	Se hace caso omiso del valor.
		byte 4: 0	Se hace caso omiso del valor.
		byte 5: dirección IP 2 (158)	1-239 (byte alto)
		byte 6: dirección IP 1 (139)	0-255 (byte bajo)
		byte 7: dirección IP 4 (7)	0-255 (byte alto)
		byte 8: dirección IP 3 (10)	0-255 (byte bajo)
		byte 9: máscara de subred 2 (255)	255
		byte 10: máscara de subred 1 (255)	0-255
		byte 11: máscara de subred 4 (0)	0-255
		byte 12: máscara de subred 3 (248)	0-255
		byte 13: camino 2 (158)	1-239 (la dirección del camino debe encontrarse en la misma subred que la dirección IP.)
		byte 14: camino 1 (139)	0-255
		byte 15: camino 4 (1)	0-255
		byte 16: camino 3 (8)	0-255
Manage_Param	Número de palabra	byte alto	byte bajo
	1	informe de actividad	00
	2	informe de operación (consulte la nota 1)	informe de comunicación (consulte la nota 1)
	3	temporización (ms)	
	4	18 (INT)	
Nota 1: La tabla siguiente ofrece información detallada para los códigos de dirección IP (correctos e incorrectos).			

Consulte el ejemplo para cambiar los parámetros IP por SEND_REQ (Véase *Ejemplo: Cambio de parámetro IP mediante SEND_REQ*, p. 15).

Cambio de códigos de dirección IP

Informe de operación	Informe de comunicaciones	Significado
<i>código correcto</i>		
FE (hex)	00 (hex)	La petición de envío (SEND_REQ) cambió con éxito la dirección IP.
<i>Códigos de error</i>		
01 (hex)	FF (hex)	Valor de código de petición no válida (por ejemplo, no 16#37).
00 (hex)	03 (hex)	Enviar petición a dirección IP de ETY en lugar de por la placa de conexiones.
00 (hex)	07 (hex)	Direccionamiento incorrecto a ETY.
16 (hex)	FF (hex)	Dirección IP no válida.
17 (hex)	FF (hex)	Subred no válida.
18 (hex)	FF (hex)	Dirección de camino no válida.
19 (hex)	EF (hex)	Dirección de red no válida.
1A (hex)	FF (hex)	La dirección IP ya se ha definido para enviar una petición (SEND_REQ).
FD (hex)	00 (hex)	La petición de envío (SEND_REQ) no cambió con éxito la dirección IP.

Acceso a objetos

Estas peticiones ofrecen un conjunto de servicios que proporciona acceso de lectura y escritura a los datos de bit interno y de tipo de palabra, a los datos de bit del sistema y de tipo de palabra, a los datos de punto flotante, a los datos de constante y a los datos de SFC.

Nombre de petición	Código de petición	Código de informe	Comentario
READ_OBJECT	16#36	16#66	Se utiliza para leer uno o más objetos consecutivos del mismo tipo.
WRITE_OBJECT	16#37	16#FE	Se utiliza para escribir uno o más objetos consecutivos del mismo tipo.
READ_INTERNAL_BIT	16#00	16#30	Se utiliza para leer el valor de un bit interno.
WRITE_INTERNAL_BIT	16#10	16#FE	Se utiliza para escribir el valor de un bit interno.
READ_INTERNAL_WORD	16#04	16#34	Se utiliza para leer el valor de una palabra interna.
WRITE_INTERNAL_WORD	16#14	16#FE	Se utiliza para escribir el valor de una palabra interna.

Gestión de modos de funcionamiento

Estas peticiones ofrecen un conjunto de servicios que puede utilizarse para gestionar los modos de funcionamiento de un procesador.

Nombre de petición	Código de petición	Código de informe	Comentario
RUN	16#24	16#FE	Se utiliza para iniciar la ejecución de las tareas de un procesador.
STOP	16#25	16#FE	Se utiliza para detener la ejecución de las tareas de un procesador.
INIT	16#33	16#63	Se utiliza para disparar un inicio en frío o en caliente.

Gestión de reservas

Estas peticiones ofrecen un mecanismo de reserva que se utiliza para proteger el servidor frente a conexiones concurrentes durante un periodo crítico.

Nombre de petición	Código de petición	Código de informe	Comentario
RESERVE	16#1D	16#FE	Habilita un cliente para que reserve general o parcialmente las funciones de un servidor.
RELEASE	16#1E	16#FE	Habilita un cliente para que libere el servidor reservado.
I_AM_ALIVE	16#2D	16#FE	Se utiliza para mantener la reserva.

Pantalla de entrada asistida

Presentación Para esta función de comunicación, puede solicitar la pantalla de entrada asistida.

Nota: Los símbolos se aceptan.

Ilustración La pantalla siguiente muestra un ejemplo de la pantalla de entrada asistida de la función:

SEND_REQ

Parámetros

Dirección: [dropdown] [ellipsis] [input]

Código de petición: [dropdown] [ellipsis] [input]

Datos que se desean enviar: [dropdown] [ellipsis] [input]

Zona de recepción: [dropdown] [ellipsis] [input]

Informe: [dropdown] [ellipsis] [input]

Tipos posibles: matriz de enteros constantes, matriz de enteros (n>=6)
Dirección ADDR(..)

Correcto Cancelar

Dirección El tipo posible de objeto es:

- ADDR (STRING).
- MATRIZ [0..5] DE INT.

Nota: Si introduce un valor directamente en un campo, el botón de entrada de dirección asistida se vuelve gris.

Código de petición Los objetos posibles son del tipo INT:

- Variables
- Constantes
- Valor inmediato

Nota: Si introduce una constante, aparecerá un campo de entrada para ello. Si introduce una variable, puede localizarse o no.

Datos que se desean enviar

Estos datos que se desean enviar se almacenan en forma de matriz de enteros. La matriz puede localizarse o no.

Zona de recepción

La zona de recepción es una matriz de enteros. La matriz puede localizarse o no, y su tamaño depende del código de petición utilizado.

Informe

El informe es una matriz de 4 enteros.

Nota: Tenga cuidado de no utilizar varias áreas de memoria idénticas para las tablas de informes, de lo contrario la función de lectura de variables no podrá funcionar.

Ejemplo de cómo enviar una petición UNI-TE

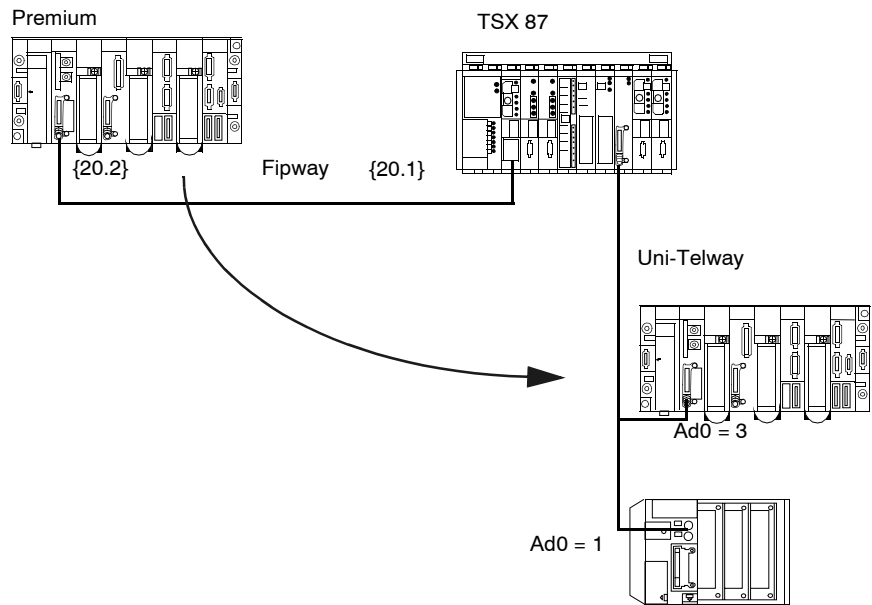
Presentación

La estación 2 de la red 20 debe enviar una petición de identificación al dispositivo con dirección Ad0=3 del bus Uni-Telway de la estación 1 de la misma red. La petición de identificación tiene el código decimal 15 (o 16#0F).

La tabla de gestión se encuentra en %MW10 : 4.

Ilustración

Las dos estaciones se conectan mediante una red Fipway:



Programación Programación en ST:

```
IF RE(%IO.3.2) AND NOT %MW10.0 THEN
    SEND_REQ(ADDR(' {20.1}0.5.1.3' ),15,%MW0:1,
    %MW10:4,%MW100:24);
END_IF;
```

Parámetros de petición:

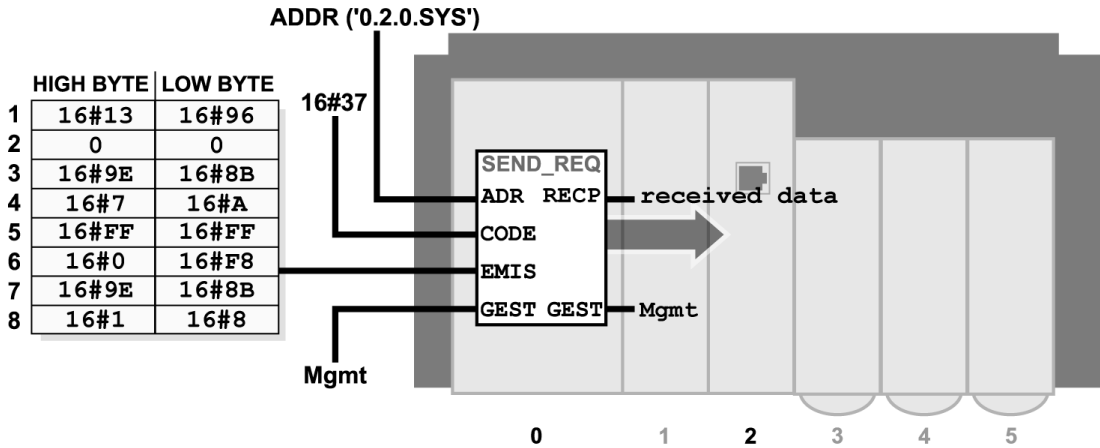
Parámetros	Descripción
ADDR('{20.1}0.5.1.3')	<ul style="list-style-type: none"> ● {20.1}: red 20, estación 1 ● 0: bastidor ● 5: módulo ● 1: canal 1 ● 3: dirección de destino
15	Petición 15 (o 16#0F si la codificación es hexadecimal).
%MW0:1	Datos enviados (por ejemplo: no hay datos que enviar).
%MW10:4	Tabla de gestión
%MW100:24	Contenido de la respuesta (recepción de 24 palabras)

Nota: Cada vez que la función se inicia, inicie el parámetro de longitud (en el ejemplo: %MW13 = 0).

Ejemplo: Cambio de parámetro IP mediante SEND_REQ

Ilustración

El gráfico muestra cómo definir los parámetros IP del módulo ETY del alojamiento 2 con el bloque SEND_REQ:



Nota:

- ADR: refleja la posición del ETY en el alojamiento 2.
- CODE: refleja el valor de REQUEST_CODE.
- EMIS: contiene los parámetros IP de Data_to_Send:
 - dirección (139.158.10.7)
 - máscara de subred (255.255.248.0)
 - camino (139.158.8.1)
- GEST: refleja Management_Param (parámetros de gestión). Tiene que asignar un tiempo a la tercera palabra de Management_Param. La cuarta palabra debería tener el valor INT 18.
- RECP: este parámetro requiere un valor mínimo INT de 1, incluso cuando no se devuelva ningún mensaje de respuesta, como en el caso de una petición de cambio de IP.

Uso de la función SEND_REQ

Presentación

La función SEND_REQ se utiliza para codificar y enviar todas las peticiones UNI-TE y Modbus/Jbus, así como recibir las respuestas asociadas.

En algunos casos (por ejemplo, la lectura de tablas de palabras) es necesario volver a definir la secuencia de los objetos recibidos mediante la función ROR1_ARB (desplazamiento de un byte en un tabla). .

Ejemplo

Objetos que se van a leer:

16#0201
16#0403
16#0605
16#0807
16#0A09

Tabla de recepción tras la ejecución de SEND_REQ (objeto de lectura):

%MW100=16#0107
%MW101=16#0302
%MW102=16#0504
%MW103=16#0706
%MW104=16#0908
%MW105=16#000A

Tabla de recepción tras ROR1_ARB(%MW100:6):

%MW100=16#0201
%MW101=16#0403
%MW102=16#0605
%MW103=16#0807
%MW104=16#0A09
%MW105=16#0700
