7TS7TCP

IGSS Simatic S7 TCP/IP Interface Driver

User's Manual

Seven Technologies
7-Technologies A/S * Bistruphave 3 * DK-3460 Birkerød * Denmark
Phone: +45 45 900 700 * Fax: +45 45 900 701 * CVR no. DK 73 63 41 13
E-mail: sales@7t.dk World Wide Web: http://www.7t.dk

CONTENTS:

1 INTRODUCTION

This document describes how to set up and troubleshoot the IGSS 7TS7TCP Interface Driver.

1.1 SOFTWARE REQUIREMENTS

The driver requires a standard Microsoft TCP/IP protocol stack. This protocol stack is normally installed on all Microsoft systems and can be used as is. DNS and other helper protocols are not used by the driver and thus not required to be installed nor configured.

The driver uses ISO transport on top of TCP/IP as defined by RFC. This means that TCP/IP **port number 102** MUST be available and open for use. If you use any kind of Fire-Wall products then please make sure that TCP/IP port 102 is open for bidirectional use. If the network paths from the IGSS system to the PLC(s) include multiple Fire-Walls then these must be configured correctly as well.

The driver is designed to used with IGSS version 6.0 and higher.

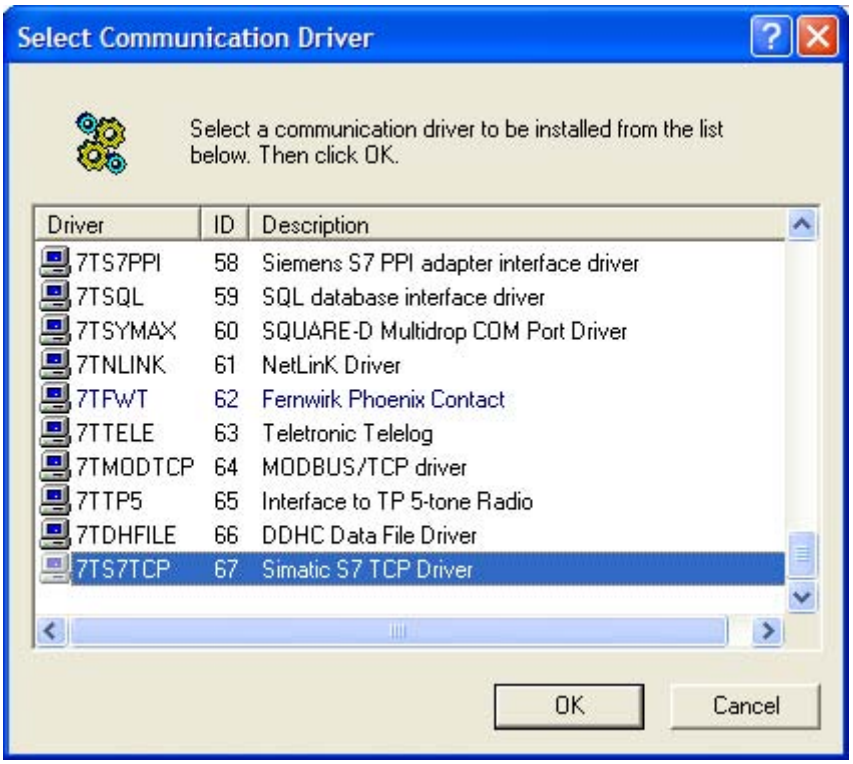
1.2 HARDWARE REQUIREMENTS

The driver requires a standard Ethernet NIC in the PC where the IGSS system is installed. The NIC must have a Microsoft approved NIC driver with an active TCP/IP stack.

2 INSTALLATION

2.1 AUTOMATIC INSTALLATION

The driver is normally installed automatically along with the rest of the IGSS system. To verify if the driver has been installed open the System Configuration (sysconfig.exe) and check if a driver with ID:67 is present in the list of available drivers:



If the driver is present then you can proceed to the next section: “Configuring the Driver”, otherwise install the driver using the manual installation procedure described below.

2.2 MANUAL INSTALLATION

Using the following step-by-step guide will install the driver manually on a PC where the IGSS system has already been installed. You need to stop the IGSS system prior to the installation and you need to be logged in with a user account with “Administrator” rights.

Step 1: Verify that the files:

7TS7TCP.DLL
7TS7TCPc.DLL
7TS7TCP.REG

exists in the GSS\ directory. If the files doesn’t exists run the IGSSUpdateClient to get the files from the 7T WEB server – or contact 7T Support (support@7t.dk) to get the files via e-mail.

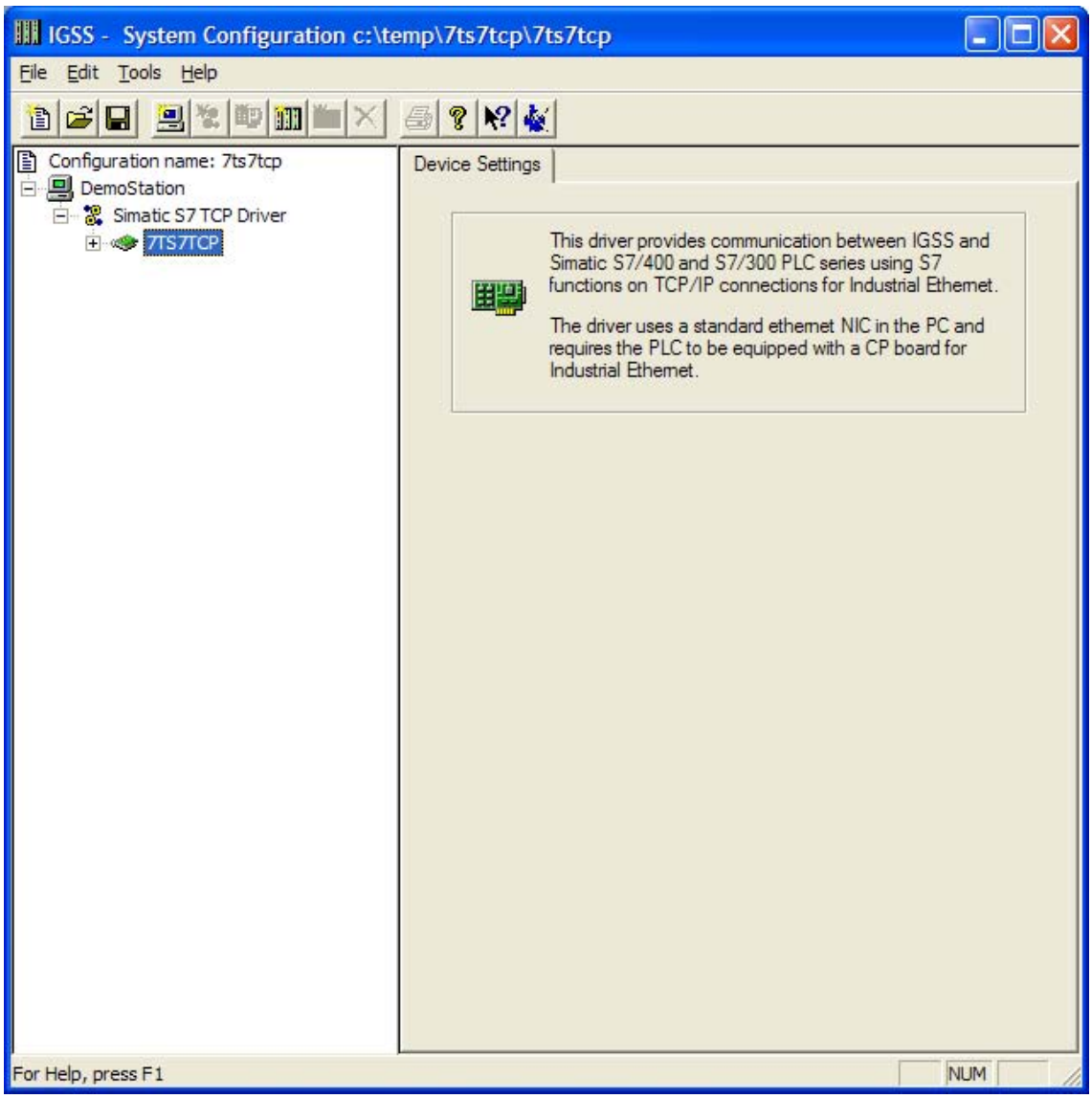
Step 2: Double-click on the 7TS7TCP.REG file to import the registry settings needed for the system to recognize the driver.

The driver is now installed.

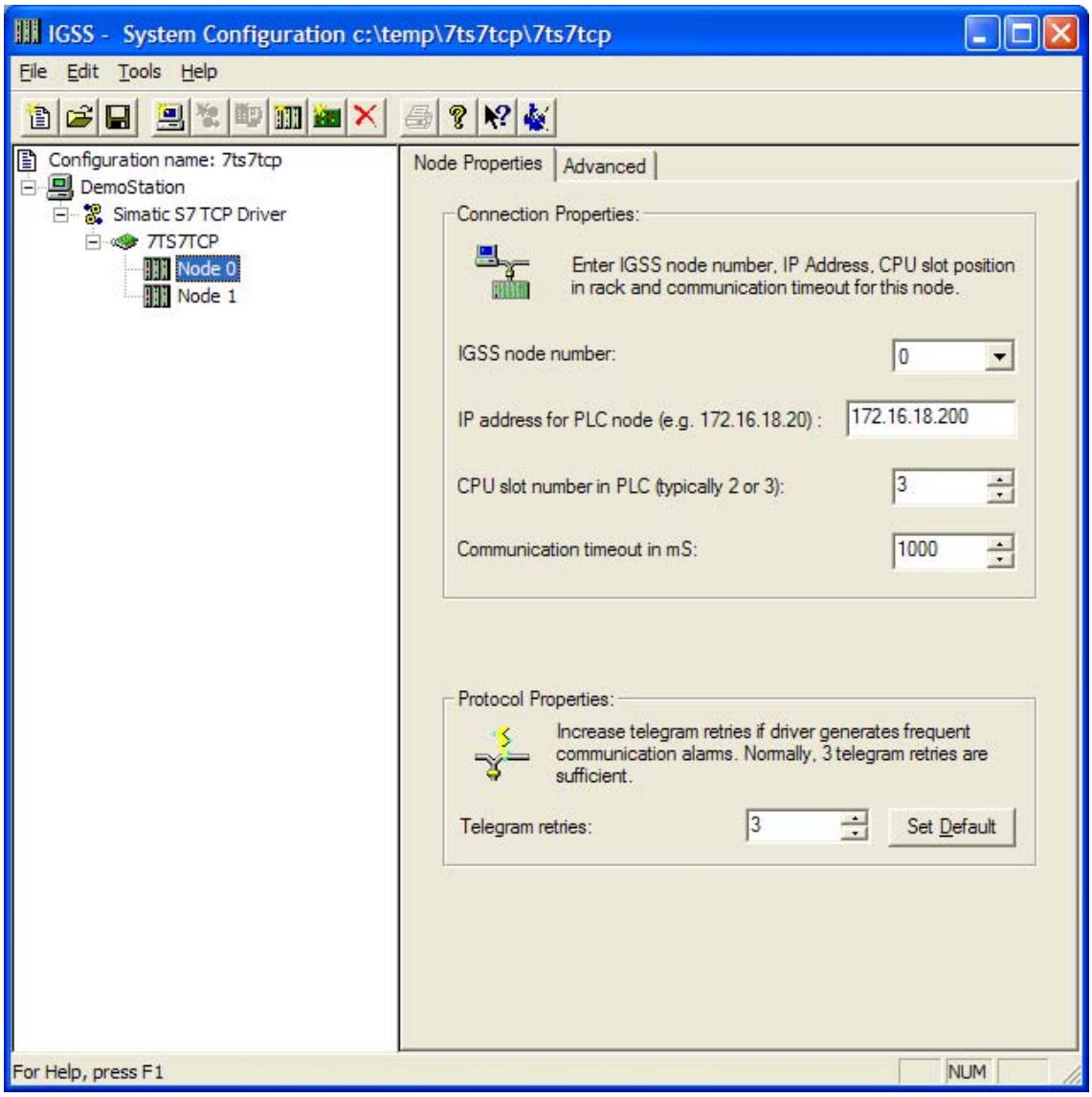
3 CONFIGURING THE DRIVER

This section describes how to configure the driver parameters. All parameters must be configured by using the System Configuration (sysconfig.exe) application. Please note that t**he IGSS system MUST be stopped and restarted** for the configured parameters to take effect.

Start the System Configuration application and add the driver 7TS7TCP (ID:67) to the requested station.



Once the driver has been added to the relevant station then you are ready to proceed with adding PLC nodes. This is done by right-clicking on the driver and select “New Node” menu point.



Each PLC node requires a few fundamental parameters:

IGSS node number: This is the node number which IGSS uses to reference a unique PLC. This node number is required when binding and IGSS atom (tag) to a register in the PLC. Any number from the drop down list can be used.

IP Address for PLC node: Here you must enter the IP address of the PLC node. The PLC programmer will be able to provide you with this number. The format must be in dotted decimal and the IP address must be reachable from the PC – either through a gateway or directly on the subnet.

CPU slot number in the PLC: Here you must fill in the slot number where the **CPU** is physically located in the PLC rack. This will normally be either slot number 2 or slot number 3 but is may vary depending on the PLC type and/or the power supply used in the rack. The PLC programmer will be able to give you information about which slot number is used for the CPU.

Communication Timeout: Normally you don't need to change this parameter. The default value of 1000mS will be OK on most networks, but if you are going to use the driver to communicate through the internet and/or trough a complex network then you might want to increase this parameter in order to avoid undesired alarms from the communication monitor.

Please note that it is EXTREMELY important that both the IP Address and the CPU slot number parameters are configured correctly. If either of these parameters are not correct then communication cannot be established.

If the PLC and the PC are connected to the network and are up running then the IP address can be verified by using the “ping” command. E.g.

ping 172.16.18.20

Please note that some Fire-Walls don’t allow ping request/responses and in this case ping can’t be used to verify the IP address.

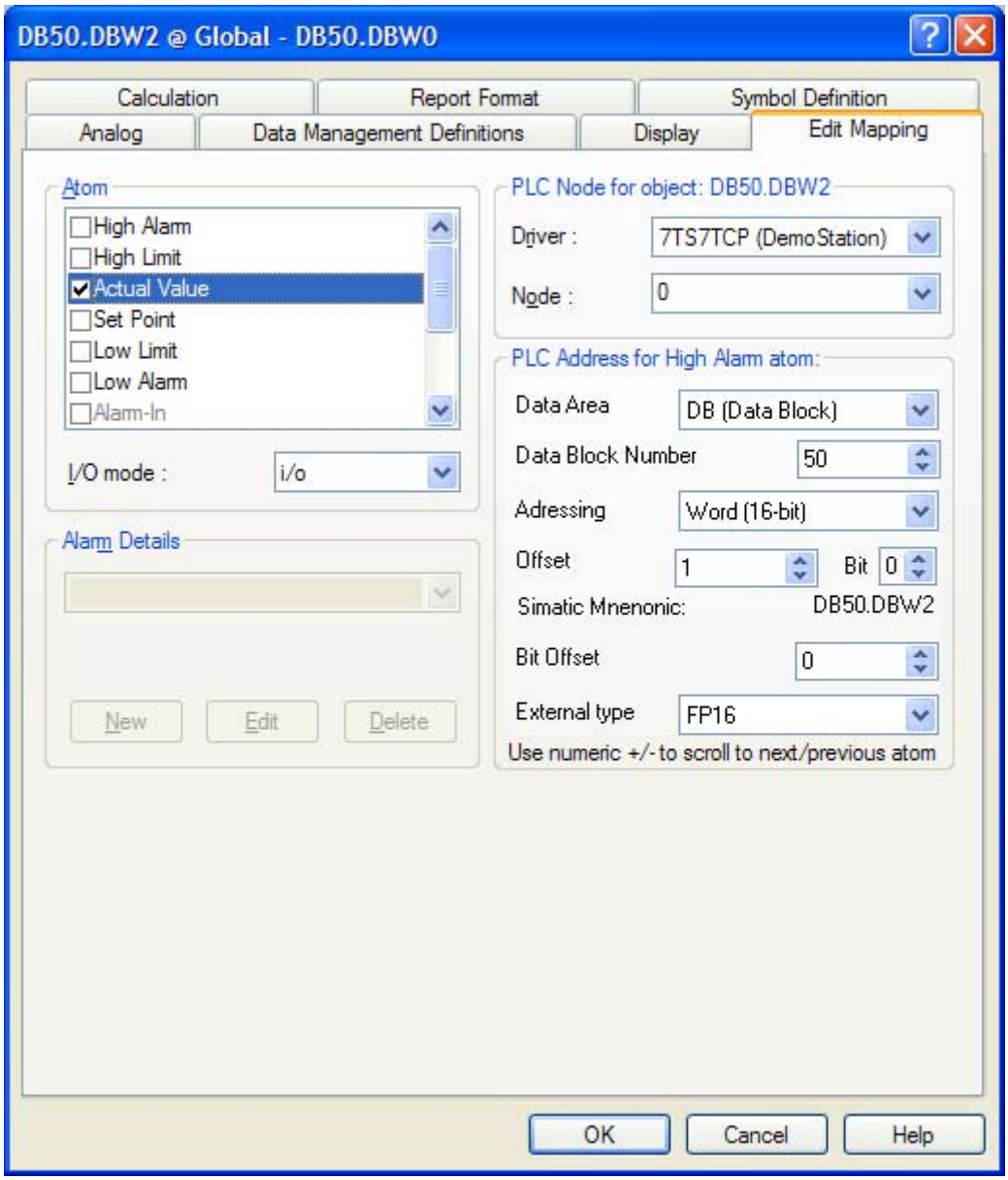
The CPU slot number parameter can’t be verified easily and it is thus important to be absolutely sure that this parameter is set correct otherwise communication can’t be established.

When a PLC node has been added you can add another by clicking the “New Node…” menu point again. The driver supports up to 16 PLC nodes.

4 CONFIGURING THE OBJECTS

Once the driver and the PLC nodes have been defined, IGSS Objects and Atoms can be linked to process variable in the PLC. Various different types of PLC memory can be accessed for read/write operations using the driver.

By using the “Edit Mapping” tab in the object properties dialog you can specify the binding between the object’s atoms and the PLC process variables. Start by selecting an atom and select the 7TS7TCP driver in the “Driver” drop down list:



Now select the desired PLC node number and continue by setting be desired Data Area (normally DB – Data Block). Then specify the block number and the addressing method and offset. Note that the corresponding Simatic Mnemonic is displayed and updated as you select the appropriate parameters. This is a help to make sure you always bind to the correct process variable.

Continue this process for each atom on the object and save the parameters by clicking the OK button when finished.

4.1 SUPPORTED MEMORY TYPES

The driver supports a number of different memory types in the PLC and a number of different addressing methods.

Data Blocks can be addressed in the range from DB0 to DB32000. Process variables stored in Data Blocks can be specified using:

Word (16-bit):	The driver will read/write a 16 bit value from/to the PLC. E.g. DB47.DBW4.
Byte (8-bit):	The driver will read/write an 8 bit value from/to the PLC. E.g. DB51.DBB211.
Bit (1-bit):	The driver will read/write a single bit from/to the PLC. E.g. DB311.DBX15.3.

These addressing methods allow you to bind to any type of process value in the PLC Data Block memory area. Please note that if you want to bind to a 32 bit (DWORD) value then use word addressing and specify the external type as one of the supported 32 bit types, then the system will automatically do the correct addressing. If you e.g. want to bind to a floating point process value at DB31.DBD70 you should specify: DB31.DBW70, Word (16-bit) and select the external type FLOAT.

Other less used supported types are:

Q (Output):	Read/write from/to Output addresses directly.
I (Input):	Read from Input addresses directly.
C (Counters):	Read from Counters directly.

IT IS HIGHLY RECOMMENTED to use Data Blocks as the primary memory type for exchanging variable information between IGSS and the PLC. Read- and especially write operations directly to Output or Counters cannot be verified by the PLC problem and might thus lead to hazards.

5 PERFORMANCE AND THROUGHPUT

The driver is designed for maximum throughput on any given TCP/IP network. On a standard PC with a 100 Mbit/s connection to the PLC you should expect a throughput of 10+ request/response cycles pr. second. Each PLC node is handled concurrent and independently. This means that if you add more PLC's to the system then the throughput pr. PLC should only be affected marginally provided that the overall network throughput is sufficient.

IMPORTANT NOTICE: The IGSS communication engine optimizes communication throughput by seeking to group data whenever possible. This means that if the communication engine is required to read e.g. DB100.DBW20 and DB100.DBW40 then it will