



EG4K BLACKBOX

How to Read and Write MODBUS Parameters for G4K BLACKBOX Device Series

Contents

1	Introduction.....	3
2	Terminology.....	4
2.1	General Terms	4
2.2	MODBUS Message Structure.....	5
2.3	The Address Register	5
2.4	Electrical Parameters Excluding Harmonics	6
2.5	Device Logs	6
2.6	Configuration Parameters	6
2.7	Custom Parameters	6
2.8	Harmonics & Interharmonics Groups & Subgroups	6
2.9	Spectrum Parameters.....	7
3	Configuring the G4K for the MODBUS Communication Protocol	8
3.1	G4K MODBUS Configuration – TCP Connection:	8
3.2	G4K MODBUS Configuration – RTU Connection:	10
4	How to Read Parameters Excluding Harmonics	11
4.1	Reading Specific Parameters	11
4.2	Reading Block Parameters.....	12
5	Reading Harmonics, Interharmonics & Spectrum Parameters	12
5.1	Reading Harmonics & Interharmonics Group & Subgroups.....	12
5.2	Reading Spectrum Parameters.....	12
6	How to Write Parameters.....	13
7	Response Time	14
7.1	Response Time via a MODBUS TCP/IP Connection	14
7.2	Response Time via a MODBUS RTU Connection	14
8	Appendix 2 - Reading & Writing Procedure Simply MODBUS.....	15
8.1	Reading MODBUS - Specific Parameters	15
8.2	Reading MODBUS - Large Block Parameters.....	16
8.3	Writing MODBUS – Single Register	18

1 Introduction

One of the main features of the G4K Device Series is that it supports both MODBUS/RTU and MODBUS over TCP/IP. Furthermore, this gateway allows the client / user to write parameters for data obtained from additional MODBUS Slave Devices into the propriety PQZIP files. The data can then be viewed in correlation with other electrical parameters within the G4K's Investigator Software.

The purpose of this document is to describe how to read and write parameters via MODBUS protocol.

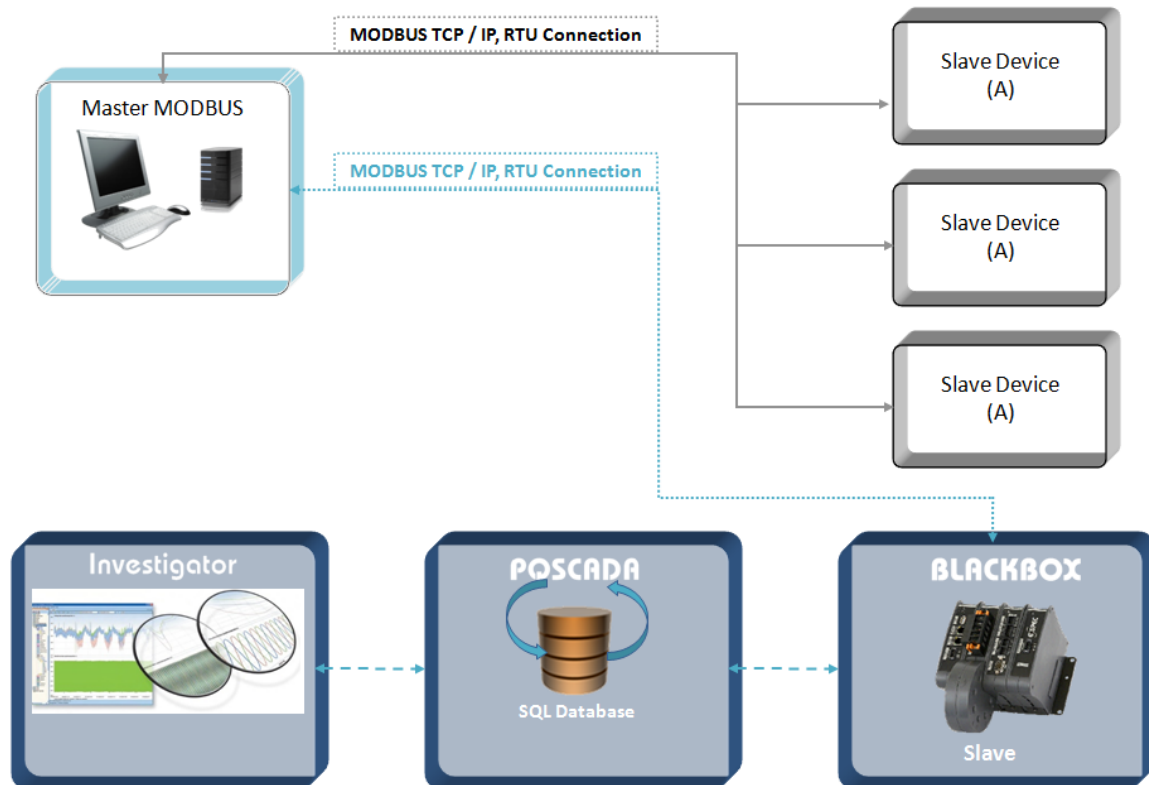


Diagram 1: MODBUS - G4K BLACKBOX Support

2 Terminology

2.1 General Terms

- **MODBUS:** Is a serial communications protocol that utilizes programmable logic controllers
- **Message:** Is the MODBUS's medium (logic controllers) of communication
- **MB:** Is the abbreviation for MODBUS
- **TCP:** *Transmission Control Protocol* is a basic communication protocol which is widely used in private and public (Internet) networks.
- **RTU:** *Remote Terminal Unit* refers to a connection to a microprocessor controlled electronic device which interfaces objects in the physical world via a SCADA system
- **Master Device:** Is a device that utilizes MODBUS and has unidirectional control over one or more other devices
- **Slave Device(s):** Refers to the devices under control of the master device, and acts in many instances as a resource for the master device
- **Parameter's Table:** Refers to the table where users set customized parameters for their individual MODBUS protocol
- **Device Address:** Describes the address of the receiver and is assigned a numeric value. The address remains the same during communication between two devices; therefore the Master can test response. Valid addresses range from 0 to 247, whereas the value 0 is being used as the broadcast address and therefore it is common to use values 1 to 247. When 0 is being used, all slaves will respond to this message.
- **FC:** A *Function Code* defines the message type and the type of action required by the slave. The parameter contains 1 Byte of information, and ranges from 1 to 255. As in the message, the Slave responds in the same function code as per the request. However when an error is detected, the highest bit of the function code is turned on. The Master can differentiate between success and failure responses.
- **Data:** Information that is transferred.
- **Error Check:** Is a numerical value assigned to verify communication.
- **Offset:** Refers to an integer within a data object or an array of numbers, that indicates the distance from the beginning of the object up until a given element or point, within the same object. The concept of a distance is valid only if all elements of the object are the same byte size. For example, given an array of characters 'A', containing 'abcdef', one can say that the element containing the letter 'C' has an offset of '2' from the start of 'A'.

2.2 MODBUS Message Structure

MODBUS utilizes messages to communicate between a MODBUS Master and Slave Devices. Slave devices will only respond to a message sent from the Master if the message is addressed to it. The parameters set within the message sets the message address, and will be the determining factor whether or not the Slave device(s) will be prompted to respond. The MODBUS message consists of four basic elements that are present in each message, and the sequence remains fixed making the content easily identifiable.

The structure of the message is:



2.3 The Address Register

The attached file “Address Register” includes all the MODBUS parameters that will be required for your MODBUS configuration. Each parameter has its own unique **Address Register**.

The table below defines and lists abbreviations, and terminology used within the Register:

Parameter	Description
Code	Parameter
Code Description	Describes the parameter
Size	Size of the parameter in Bit
R, W, RW	Parameter functionality: R: for reading from the Register W: for writing into the Register RW: for both reading & writing into the Register
Type	Type of parameter i.e. – be that be an integer, a float or a date etc.
FC4 (16Bit)	Input Register Address Without 30,001 Offset
MB FC4 With Offset	Input Register Address With 30,001 Offset

The parameters are arranged into various sub-groups, which will simplify the MODBUS reading and configuration procedure for the G4K. The groups read as follows:

- Electrical Parameters Excluding Harmonics,
- Device Logs,
- Configuration Parameters,
- Custom Parameters,
- Harmonics & Inter-Harmonics Groups & Sub-Groups, and
- Spectrum Parameters

2.4 Electrical Parameters Excluding Harmonics

Both “10/12 Cycles 200ms” and “IEC-61000-4-30 Class A” aggregated parameters are readable by MODBUS. The registers that fall under this category with their corresponding MODBUS addresses are listed in attached **Appendix 1 - TABLE 1, Electrical Parameters Sheet**.

2.5 Device Logs

The registers that fall under this category with their corresponding MODBUS address are listed in **Appendix 1 - TABLE 1, Device Log Sheet**.

2.6 Configuration Parameters

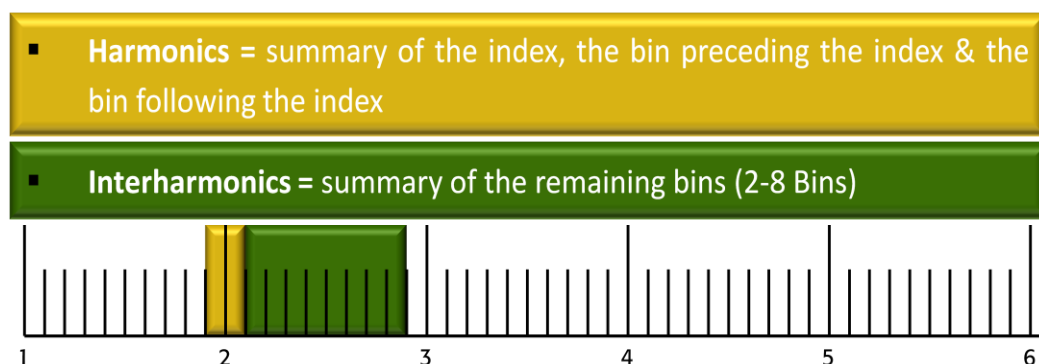
The registers that fall under this category with their corresponding MODBUS address are listed in **Appendix 1 - TABLE 1, Configuration Parameters Sheet**.

2.7 Custom Parameters

The registers that fall under this category with their corresponding MODBUS address are listed in **Appendix 1 - TABLE 1, Custom Parameters Sheet**.

2.8 Harmonics & Interharmonics Groups & Subgroups

In total 50 Harmonics & Interharmonics groups & subgroups are calculated for every aggregated period. Graph 1 below defines the subgroup calculation graphically:



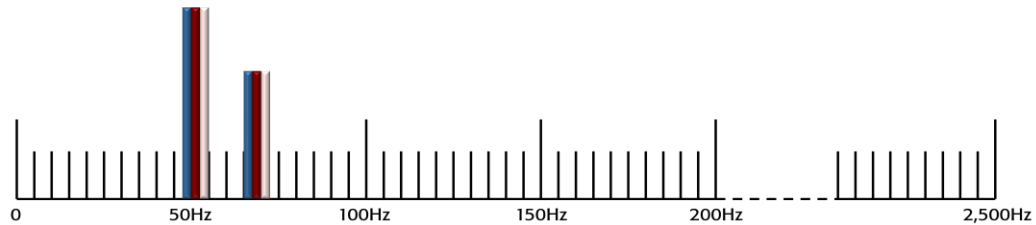
Graph 1: Harmonic & Interharmonic Subgroup Calculation

These registers with their corresponding MODBUS codes are listed in **Appendix 1 - TABLE 1, Harmonics & Interharmonics Sheet**.

2.9 Spectrum Parameters

In total 512 Spectrum values are calculated for every aggregated period. Graph 2 below defines the calculation graphically:

- **Spectrum** = displays the Harmonic values in the Frequency Domain



Graph 2: Spectrum Calculation

These registers with their corresponding MODBUS codes are listed in **Appendix 1 - TABLE 1, Spectrum Sheet**.

3 Configuring the G4K for the MODBUS Communication Protocol

The G4K can be integrated into any MODBUS Master. The procedure below outlines the general configuration via either a TCP or RTU connection, including the address registration in the MODBUS Master application. The TCP or RTU connection is done via G4K's Web Interface, and in addition you will need to define the usual settings in your MODBUS Master application (such as baud rate, slave address, and other miscellaneous configurations).

3.1 G4K MODBUS Configuration – TCP Connection:

1. In order to configure the MODBUS TCP/IP connection, login to the G4K's Web Interface with **administrative rights** (default password: **12345**)



Figure 1: G4K's Web Interface – Login Administrator

2. Go to the **“Configuration”** & select **“Network”** option.

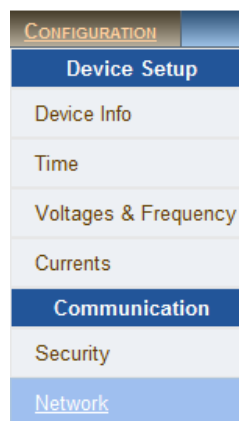


Figure 2: G4K's Web Interface – Network Option Selection

3. Set the MODBUS TCP Settings as per the screen below:
 - a. In the **"Slave Address"**, enter the "Slave Address" you would like to set for the G4K (the default address is 159)
 - b. In the **"MODBUS Port"** enter the "MODBUS port" that you would like to set for the G4K (the default address is 502)
 - c. Select "Apply Changes" Tab

The screenshot displays the G4K's Web Interface with the following sections and settings:

- LAN1:** Auto DHCP (Disable), IP Address (100.100.100.69), Subnet Mask (255.255.255.0)
- LAN2/LCD:** Auto DHCP (Disable), IP Address (192.168.168.168), Subnet Mask (255.255.255.0)
- Port Setup:** HTTP Port (80), FTP Daemon (21), FTP Data (20), SMTP Port (25)
- Outer Access:** Gateway (100.100.100.253), SMTP Server (0.0.0.0)
- Network Interface:**

Interface	Link	Speed	Duplex	Mode
LAN1 [Link]	On	100 Mbits	Full	Auto negotiate
LAN2 [LCD]	Off	10 Mbits	Full	10Mbit FD
- Connections:**

HTTP Active	OPC Active	LCD Active	FTP Active	FTP Max
1	0	0	0	5
- Modbus TCP:**

Slave Address	159	Modbus Port	502
---------------	-----	-------------	-----

Red arrows indicate the following actions:

- a:** Points to the Slave Address field in the Modbus TCP section.
- b:** Points to the Modbus Port field in the Modbus TCP section.
- c:** Points to the Apply Changes button at the top left.

Figure 3: G4K's Web Interface – MODBUS TCP Settings



NOTE: The G4K BLACKBOX supports up to **THREE** concurrent **MODBUS** users (**over three consecutive ports while only the first port is configurable**). Each user can read & write different parameters, without the sessions colliding with one another. For example Elspec's default setting of **502** – the additional port settings will read **503 & 504**.

3.2 G4K MODBUS Configuration – RTU Connection:

1. In order to configure the MODBUS/RTU connection, login to the G4K's Web Interface with **administrative rights** (default password: **12345**):



Figure 4: G4K's Web Interface – Login Administrator

2. Go to the **"Configuration"** & select **"Serial Ports"** option.

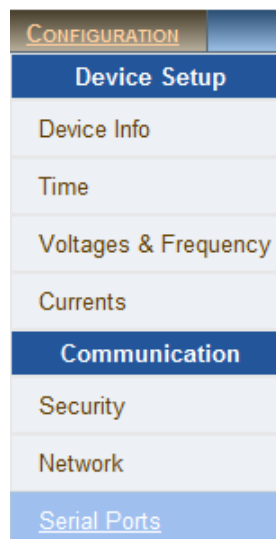


Figure 5: G4K's Web Interface – Network Option Selection

3. The screen below will be displayed.
 - a. In the **"Bitrate Scroll"**, select the applicable **"Bitrate"** as per your system settings (the default rate is 19200),

- b. In the “Serial Mode”, select “MODBUS RTU (SLAVE)” option
- c. In the “MODBUS Slave Address” scroll to your applicable value as per your system settings (the default address is 159)
- d. Select “Apply Changes” Tab:-

The screenshot displays the G4K Web Interface for MODBUS RTU Settings. At the top, there are five buttons: 'Apply Changes' (highlighted with a red arrow 'd'), 'Refresh Data', 'Connect', 'Disconnect', and 'Reset modem'. Below these is the 'RS-485 / RS-422' section. It contains four columns: 'Bitrate' (dropdown menu with 19200 selected, highlighted with a red arrow 'a'), 'Data Bits' (8 bit), 'Parity' (None), and 'Stop Bits' (1). Below the 'Bitrate' dropdown is the 'Serial Mode' dropdown menu, which is open and shows several options: 'TTY', 'ELCOM RS485-OPC GATEWAY', 'MODBUS RTU (SLAVE)' (highlighted with a red arrow 'b'), 'GPS', and 'PPP'. To the right of the 'Serial Mode' dropdown is the 'ModBus Slave Address' field, which contains the value 159 (highlighted with a red arrow 'c'). Below the 'ModBus Slave Address' field is a checkbox for '[Message Log]' which is checked. At the bottom, there is a 'Signal Quality' section showing 'N/A'.

Figure 6: G4K's Web Interface – MODBUS RTU Settings

4 How to Read Parameters Excluding Harmonics

4.1 Reading Specific Parameters

1. Use **Function Code as 4** in your MODBUS Master application
2. Select the applicable parameter you would like to read by cross referencing the parameter from **Appendix 1 – TABLE 1 (All Sheets apart from Harmonics, Interharmonics & Spectrum Parameters)**

SEE ALSO: [Appendix 2 - Reading MODBUS – Specific Parameters “Simply MODBUS”](#)

4.2 Reading Block Parameters

Some of Elspec's "large block" parameters contain unused Registers. In many of these instances, MODBUS may read some of these Registers as an "Error". Elspec accommodates this reading, and such values will appear as "NaN = 0xFFFFFFFF". To make it easier you just need to enter the first parameter's register followed by No. of Registers.

SEE ALSO: [Appendix 2 - Reading MODBUS - Large Block Parameters "Simply MODBUS"](#)



NOTE: To set the right number of registers locate the register size of each parameter from **Appendix 1 – TABLE 1**.

5 Reading Harmonics, Interharmonics & Spectrum Parameters

5.1 Reading Harmonics & Interharmonics Group & Subgroups

Considering the vast amount of calculated values involved for this category, Elspec didn't allocate a dedicated register for each individual Harmonic but instead replaced it with a "Vector Register" (its size is 100 Registers that includes 50 values).



Note: By utilizing this method you may be limited as when the G4K will be disconnected, these pointers in the register will be deleted.

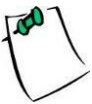
In order to read the mentioned values:

1. First and foremost, **write** the desired parameter code into the register address (**2441 without offset**) or (**32442 with offset**). The parameter code can be found on **Appendix 1 – TABLE 1, Harmonics Sheet**.
2. Using FC4 **read** the vector value from the register address (**2442 without offset**) or (**32443 with offset**).

SEE ALSO: [Appendix 2 - Reading & Writing Procedure "Simply MODBUS"](#)

5.2 Reading Spectrum Parameters

For every aggregated period, 512 Spectrum values are calculated. In order to read each value you will need to define each specific parameter including the Spectrum Index (e.g. 1, 2, 3, ..., 512). The reading parameter will be at a size of 2 Registers.



Note: The Index number represents the Frequency calculated as:

$$Index = Round \left(\frac{f_{sys}}{10} \right)$$

For Example: In 50Hz index 1 represents 5Hz.

In order to read a Spectrum value, follow the following procedure:

1. **Write** the desired parameter code into register address **(2439 without offset)** or **(32440 with offset)**. The parameter code can be found on **Appendix 1 – TABLE 1, Spectrum Sheet**.
2. Enter the desired Spectrum index (e.g. 1, 2, 3, ..., 512) into the register address **(2440 without offset)** or **(32441 with offset)**
3. Using FC4 **read** the specific spectrum value from register address **(2542 without offset)** or **(32543 with offset)**

SEE ALSO: [Appendix 2 - Reading & Writing Procedure “Simply MODBUS”](#)

6 How to Write Parameters

As mentioned previously Elspec’s MODBUS application supports an integer of 32-bit element types. When following the “writing” procedure, cross reference to the parameters you would like to insert as per Appendix 1 – TABLE 1, in the groups mentioned above.



Note: The writing procedure can only be done in Function Codes 6 & 16.

- For individual parameters use Function Code 6 (FC 6)
 - For block parameters use Function Code 16 (FC16)
-

1. Initiate a MODBUS session to your G4K Unit either via TCP/IP or RTU connection,
2. Insert the Authentication Code in your MODBUS Registration address **998**.
Authentication will be verified automatically. (The default authentication password is 12345).



NOTE: On conclusion of the authentication, the “session timeout” settings will be automatically set to 5 minutes.

3. Write the desired value into applicable register address. The writable registers can be found in **Appendix 1 – TABLE 1, on the Configuration parameters or Custom Parameters Sheets**.

SEE ALSO: [Appendix 2 – Writing MODBUS – Single Register “Simply MODBUS”](#)

7 Response Time

Unlike conventional systems, the time that it takes for the G4K unit to respond is based on the recording measured at the “Last Recorded Time” and not the “Next Recorded Time”. For example if the latest G4K recording time took place at 1.00pm, the G4K will not wait for the next recording time at 3 minutes, before responding to the MODBUS Master command.

7.1 Response Time via a MODBUS TCP/IP Connection

- Depending on the NETWORK and the DATA block size, the G4K will respond to 99% of the requests within 20ms (Enabled in Firmware 4.1 and newer, with a Telnet command).
- 30ms – 100%

7.2 Response Time via a MODBUS RTU Connection

- No more than 100ms

8 Appendix 2 - Reading & Writing Procedure Simply MODBUS

In the Register Table (Appendix 1) Elspec has provided the corresponding MODBUS Function Codes (FC4) and MODBUS Function Codes with offset (FC4 with offset) addresses for each parameter. You will need to cross reference each parameter when entering the applicable MODBUS Register Address into your MODBUS Master application. As mentioned previously, each parameter is classified by a function (**R**: for reading, or a **W**: for writing, or a **RW**: for both reading & writing. In order to demonstrate the procedures Elspec used a demo version “Simply MODBUS” in Procedures 1, 2 & 3 below. You may use any other suitable application as the procedure should be similar:

8.1 Reading MODBUS - Specific Parameters

For this procedure we are using the Voltage Channels ($V_N, V1, V2, V3$) RMS Class A Parameter from the attached Appendix 1 – TABLE 1 Electrical Parameters Sheet select the parameters below:

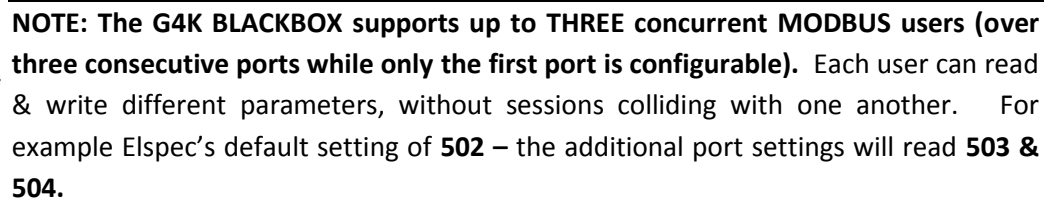
- PRM_CODE_LONG_VN_RMS - Byte Size **4** - MODBUS Codes 3503 - MODBUS Code with Offset **33504**
- PRM_CODE_LONG_V1_RMS - Byte Size **4** - MODBUS Codes 3505 - MODBUS Code with Offset **33506**
- PRM_CODE_LONG_V2_RMS - Byte Size **4** - MODBUS Codes 3507 - MODBUS Code with Offset **33508**
- PRM_CODE_LONG_V3_RMS - Byte Size **4** - MODBUS Codes 3509 - MODBUS Code with Offset **33510**

Set the **Port Settings** as per your Port settings in the Website Configuration (The default is 502)

1. Insert the G4K's IP Address – in this example it is **100.100.100.78**
2. Set the **Slave ID** as per your Slave ID settings in the Website Configuration (The default is 159)
3. Should the data be composed of 16 Bit Registers - Insert the **Function Code as 4**
4. Enter the **MODBUS Register as 33504**
5. Enter the **Minus Offset Value as 30001**
6. Enter the **No. of Registers as 8**



NOTE: MODBUS FUNCTION CODE (FC 4) WORKS ON A BIT FLOAT SIZE REGISTER OF 16. Elspec's parameters each consists of a Bit Size of 32 Bit i.e. (composing of 2 x 16 Bit Float Registers). So the No. of Registers in this instance will be 8 instead of 4.



8.3 Writing MODBUS – Single Register

Writing in MODBUS can be done by using either Function Code 6 (FC6) or Function Code 16 (FC16).

- FC6 is used when writing Single Registers & their Bit Size is less than 16 Bits,
 - FC16 is used when writing Multiple Registers with a bit size \geq 16Bits.
 - In this example we are using PRM_CODE_SAVED_HARMONICS_PRM - Byte Size 2 - MODBUS Codes **2441** - MODBUS Code with Offset **32442** – Harmonic Pointer **592**
1. Set the **Slave ID** as per your Slave ID settings in the Website Configuration (The default is 159)
 2. Enter the **MODBUS Register** as **32442**
 3. Insert the **Function Code** as **6**
 4. Enter the **Harmonic Pointer** as **592** (your data value)

The screenshot shows the 'Simply Write Modbus TCP 1.3.4' application window. It features several input fields and checkboxes for configuring a MODBUS write operation. The 'mode' is set to 'TCP'. The 'Slave ID' is 159, 'First Register' is 32442, and '# Values to Write' is 1. The 'function code' is 6, and 'register size' is 16 bit registers. The 'Data Type' is 16bit INT. The 'Values to Write' field shows 592.0000, and the 'bytes' field shows 0250. The 'Command' field displays the hex sequence 00 02 00 00 00 06 9F 06 09 89 02 50. The 'Response' field shows the hex sequence 00 01 00 00 00 06 9F 06 09 89 02 50. The 'expected response bytes' is 12. The interface includes buttons for 'SEND', 'SAVE CFG', 'RESTORE CFG', 'SAVE LOG', and 'clear log'. A log window at the bottom shows the command and response hex sequences.

Values to Write	register#	bytes	Data Type
592.0000	32442	0250	16bit INT

Command: 00 02 00 00 00 06 9F 06 09 89 02 50

Response: 00 01 00 00 00 06 9F 06 09 89 02 50

expected response bytes: 12

Log:
2010/12/15 09:58:45 >>> 00 01 00 00 00 06 9F 06 09 89 02 50
2010/12/15 09:58:45 <<< 00 01 00 00 00 06 9F 06 09 89 02 50

Figure 9: Writing Procedure Single Register - "Simply MODBUS"