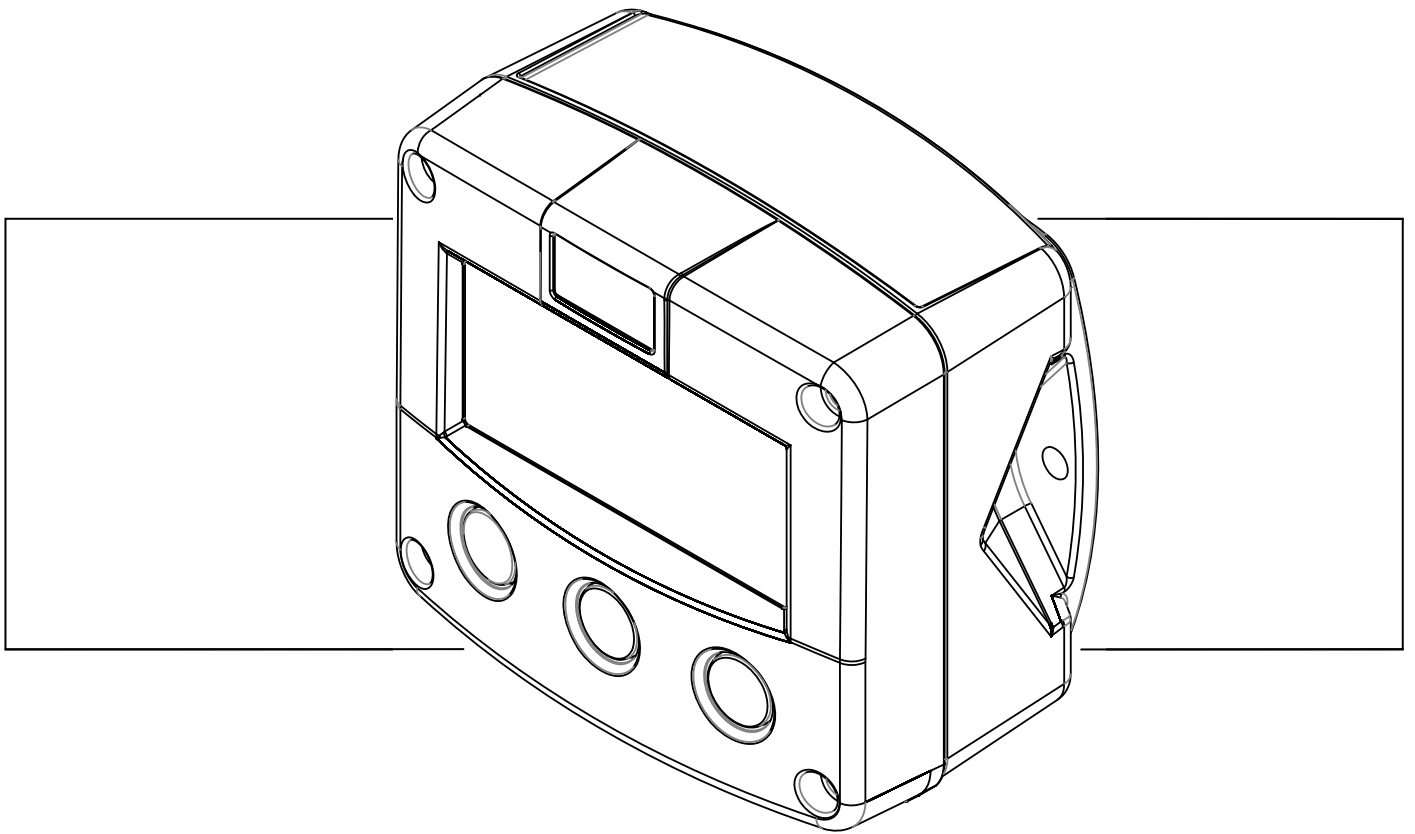


# MODBUS Communication

with Fluidwell F1xx units



## Troubleshooting Guide

## MODBUS communication with Fluidwell F1xx

|  |    |
|--|----|
| <i>Communication settings</i>              | 2  |
| <i>Physical connection</i>                 | 2  |
| RS232                                      | 2  |
| RS485 communication                        | 3  |
| 4-wire RS485/RS422                         | 3  |
| 2-wire RS485                               | 4  |
| <i>Register addressing</i>                 | 5  |
| <i>Variable size</i>                       | 5  |
| <i>Standard Fluidwell Registers table</i>  | 6  |
| <i>Example using MODBUS Poll</i>           | 7  |
| <i>Troubleshooting</i>                     | 9  |
| Using the communication traffic window     | 9  |
| Testing the receive-path                   | 10 |
| Problems with USB and/or signal-converters | 12 |

## Communication settings

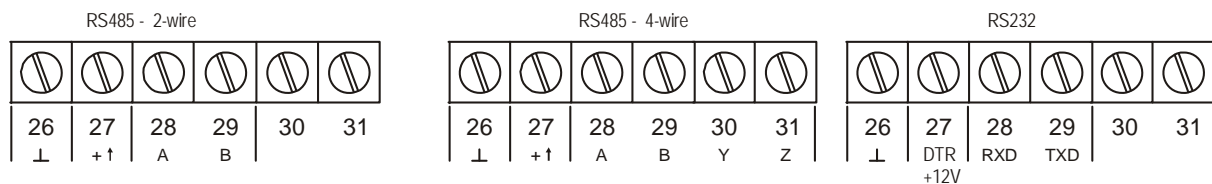
For all F1-models the *communication* settings are: **8 databits, no parity, 1 stopbit.**

The factory default settings are: **9600 BAUD, Modbus address 1**

**Communication** mode can be either **ASCII** or **RTU**, depending on the complexity of the unit.

Please verify the various settings menu.

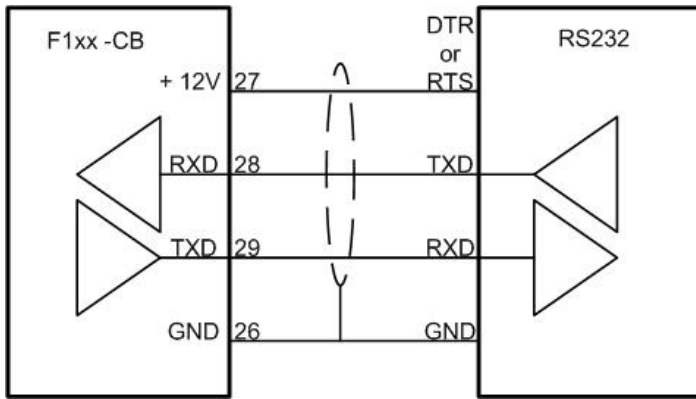
## Physical connection



**Figuur 1** screw terminals

## RS232

When using RS232 communication (option CB), terminal 27 is used for supplying the F1xx's interface. Please connect the DTR (or the RTS) signal of the interface to this terminal and set it active (+12V). If no active signal is available it is possible to connect a separate supply between terminals 26 and 27 with a voltage of 8V to 24V DC.



Figuur 2 RS232 connections

**PC-RS232 connections:**

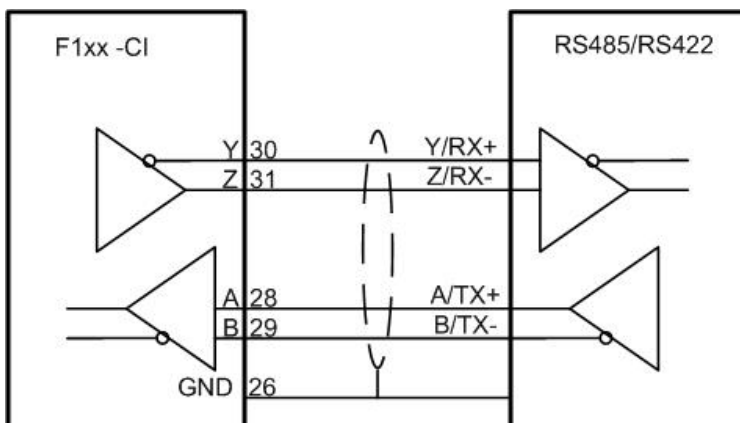
| Signal                    | 9 pin RS232 | 25 pin RS232 |
|---------------------------|-------------|--------------|
| Receive Data (RXD)        | 2           | 3            |
| Transmit Data (TXD)       | 3           | 2            |
| Data Terminal Ready (DTR) | 4           | 20           |
| Ground (GND)              | 5           | 7            |
| Request to Send (RTS)     | 7           | 4            |

**RS485 communication**

When communicating through RS485, it is important to disconnect all termination resistors. F1xx units are designed to work without them for power saving reasons. Even with an external power supply PD (12/24V AC/DC) or PM (230V AC), any termination resistors should be disconnected.

**4-wire RS485/RS422**

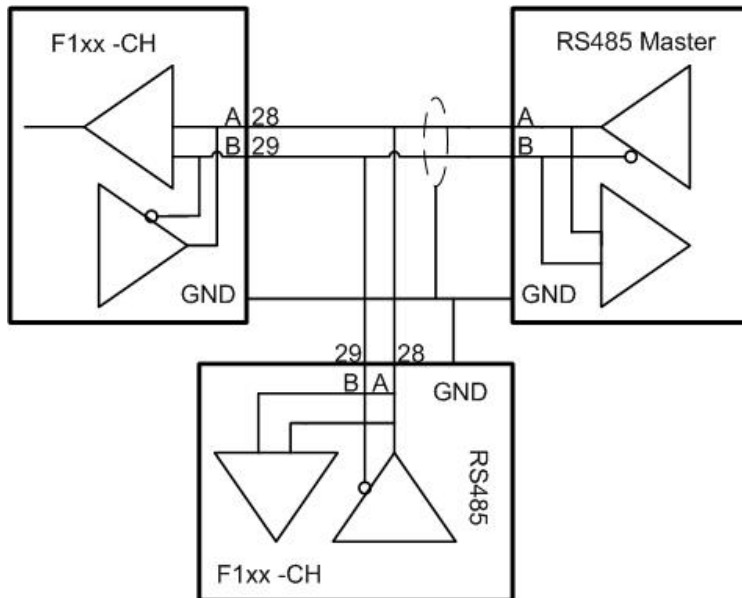
With **4-wire** communication (option CI), you can use either RS485 or RS422 drivers. There is no need to supply power to the interface of the F1xx. The grounds must be connected on both sides. Optional shielding should be connected on one side only.



Figuur 3 RS485 4-wire

## 2-wire RS485

With **2-wire** communication (option **CH**) you need to use RS485 drivers. It is possible to connect up to 32 slaves in a multidrop configuration. There is no need to supply power to the interface of the F1xx's. The grounds must be connected to all drivers. Optional shielding should be connected on one side only.



Figuur 4 RS485 2-wire

## ***Register addressing***

In MODBUS, every register is categorised in a group.

In Fluidwell products **all registers** are located in the '**Holding registers**' group.

By doing this, all registers can be accessed by just two Modbus commands:

**03 Read Holding Registers** and **16 Preset Multiple Registers**.

You can only read or write **one register at a time**.

It is not possible to read consecutive registers by increasing the number of words to read. It is possible to read only a part of the variable by starting the read-action at a different address.

Within a PLC or application with MODBUS interface every Holding registers is assigned an address somewhere between 40001 and 50000.

In the actual physical Modbus databytes these addresses are converted to a value between **0** and **9999** plus an indication of the group.

The registers addresses are in the **Standard Fluidwell Registers table** and in the appendix C of every Fluidwell manual.

When you want to read for example the serialnr. which is located on Modbus address 02 (see Modbus manual), you must set the PLC-address to 40003.

**PLC-address = Register address as stated in the manual + 40001**

## ***Variable size***

The length of the parameters in the manual is in bytes, the default length in Modbus systems is in words.

**Modbus register width = (number of data bytes as stated in the manual)/2 rounded to the next whole number.**

All variables are stored as 'unsigned integers' meaning they are always positive and you can 'extract' the value by a standard hexadecimal to decimal conversion.

## **Standard Fluidwell Registers table**

Below is a table of the registers that are implemented in all Fluidwell models. When making this table a long time ago, one mistake was made in that a register can be more than one word (2 bytes) long, without interfering with the next register.

For instance, register 02 Serial number is 2 words long, so it should cover register 02 and 03. In our implementation it only covers register 02.

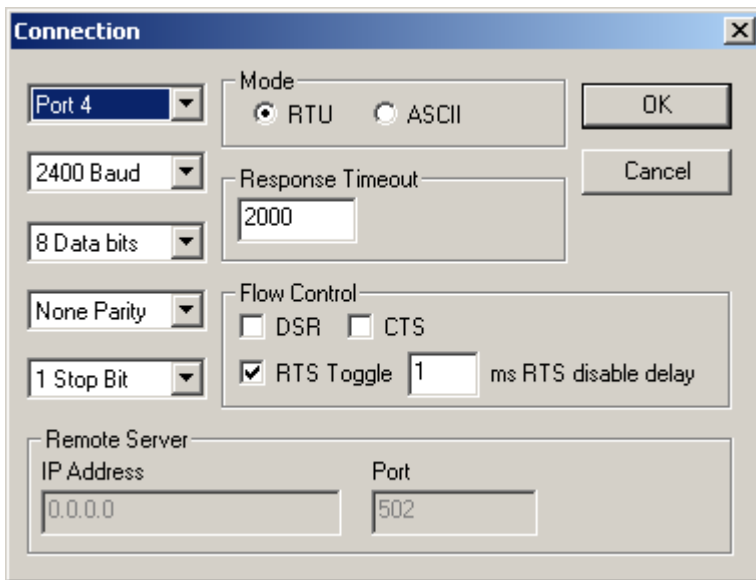
| <b>Nr.</b>  | <b>VARIABLE NAME</b>  | <b>LENG TH</b> | <b>NOT WRITABLE</b>      |
|---|-----------------------|----------------|--------------------------|
| 00  | Kind of product       | 2 bytes        | never writable           |
| 01  | Model number          | 2 bytes        | never writable           |
| 02  | Serial number product | 4 bytes        | never writable           |
| 03  | Hardware options      | 2 bytes        | never writable           |
| 04  | Software options      | 2 bytes        | never writable           |
| 05  | Protocol options      | 2 bytes        | never writable           |
| 06  | Software version      | 2 bytes        | never writable           |
| 07  | Protocol version      | 2 bytes        | never writable           |
| 08  | Display languages     | 2 bytes        | never writable           |
| 09  | Bus address           | 1 byte         | during local programming |
| 10  | Baud rate             | 2 bytes        | during local programming |
| 11  | Communication mode    | 1 byte         | during local programming |
| 12  | Comm. index use       | 2 bytes        | always writable          |
| 13  | Comm. Index           | 1 byte         | always writable          |
| 14  | Comdelay              | 2 bytes        |                          |
| 15  | general status        | 2 bytes        | never writable           |
| 16  | indexed status        | 1 byte         | never writable           |
| 17  | Reserved              |                |                          |
| 18  | Reserved              |                |                          |
| 19  | Reserved              |                |                          |
| 20  | Inputs                |                | never writable           |
| 21  | Outputs               |                | never writable           |
| 22  | Beep command          | 1 byte         | always                   |
| 23  | Time and date         | 6 bytes        | during local programming |
| 24  | Print command         |                | not implemented          |
| 25  | Reboot                | 2 bytes        | always writable          |
| 26..<br>29  | Reserved              |                |                          |
| 30.....Further product dependant variables only!! |                       |                |                          |

## Example using MODBUS Poll

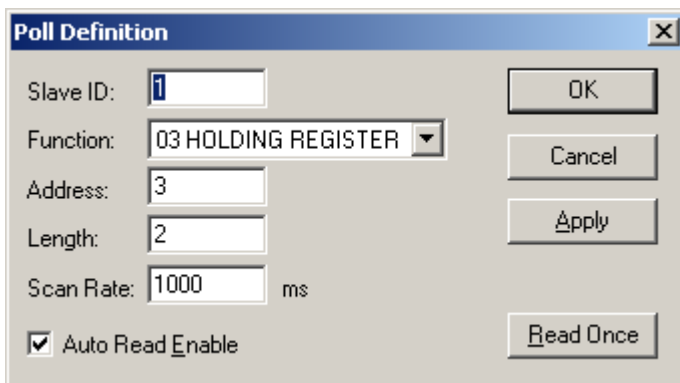
Below are a few screen shots of configuring with MODBUS Poll, a PC-utility that can easily be obtained from the internet.

The connection settings:

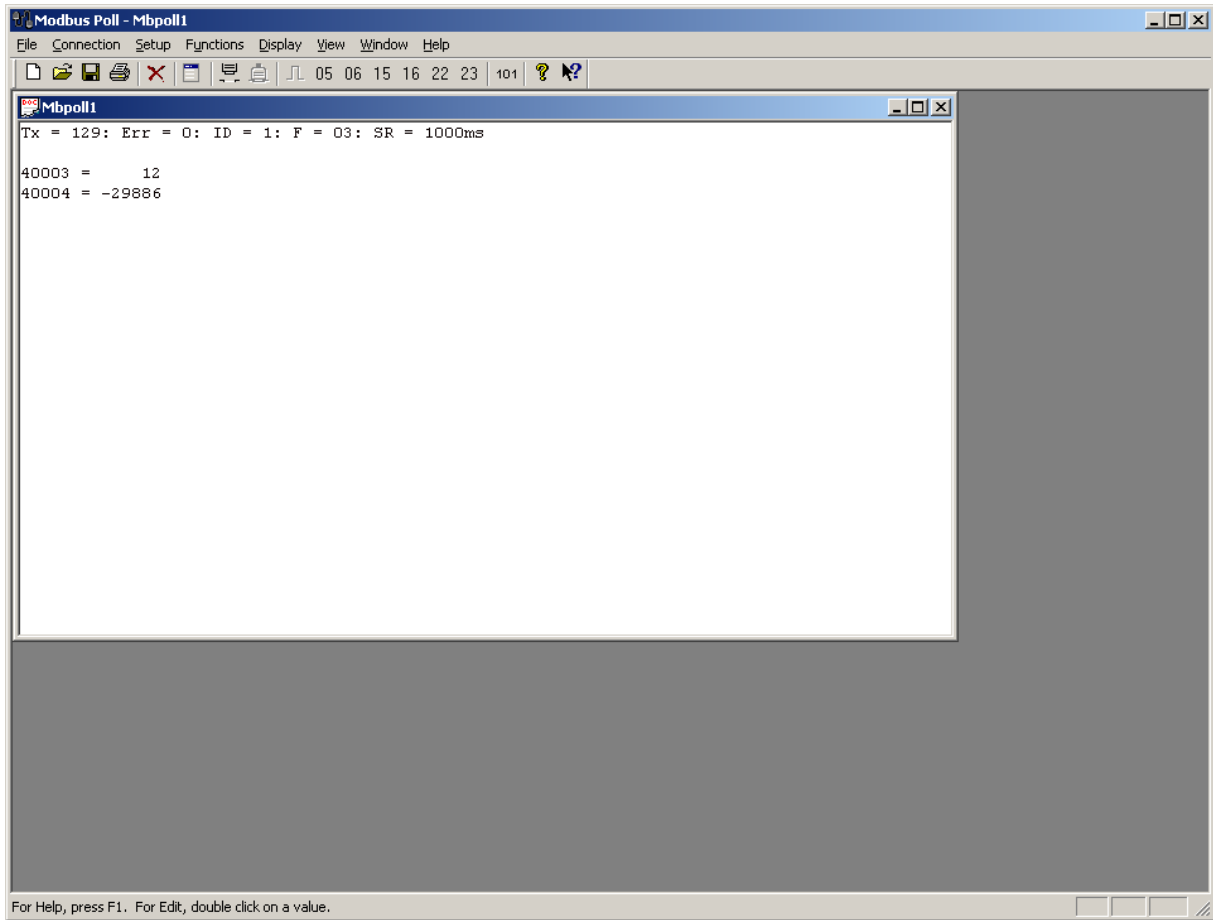
**When using an RS232 to RS485 converter, as in the example, switch on the Flow control, RTS Toggle box.**



The poll-definition for reading the Serial number, which is 4 bytes long corresponding to PLC address 40003 and 40004.



*\*Address should be: 2*





## Troubleshooting

### Using the communication traffic window

Anytime you can not get the communication to work after checking all settings, it is wise to examine the Communication Traffic window. This window displays all data that is sent and received across the communication line.

For example:

The poll-definition for reading the Serial number, which is 4 bytes long corresponding to PLC address 40003 and 40004.

**Poll Definition**

Slave ID:

Function:

Address:

Length:

Scan Rate:  ms

Auto Read Enable

Buttons: OK, Cancel, Apply, Read Once

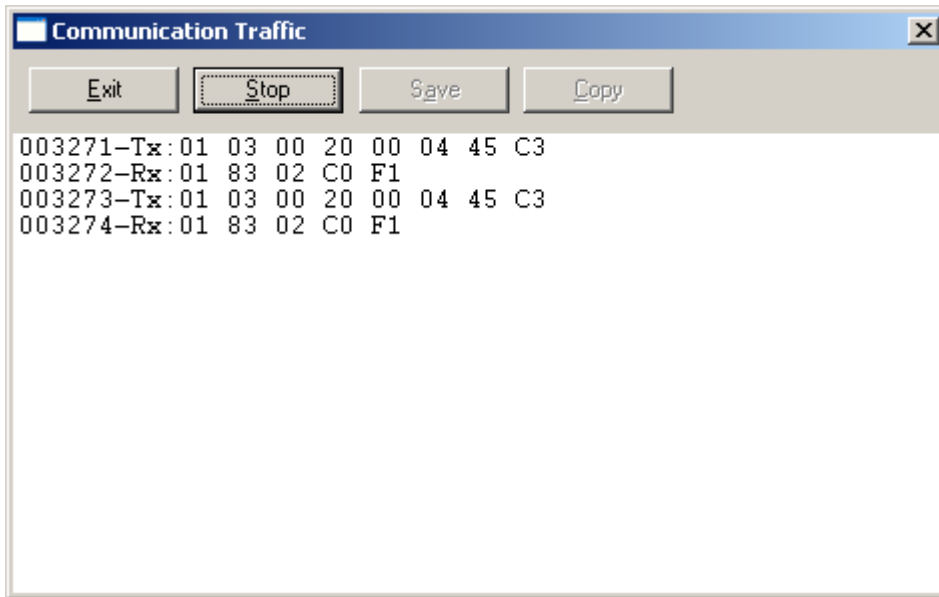
Results in the following databytes (in case all is fine !)

**Communication Traffic**

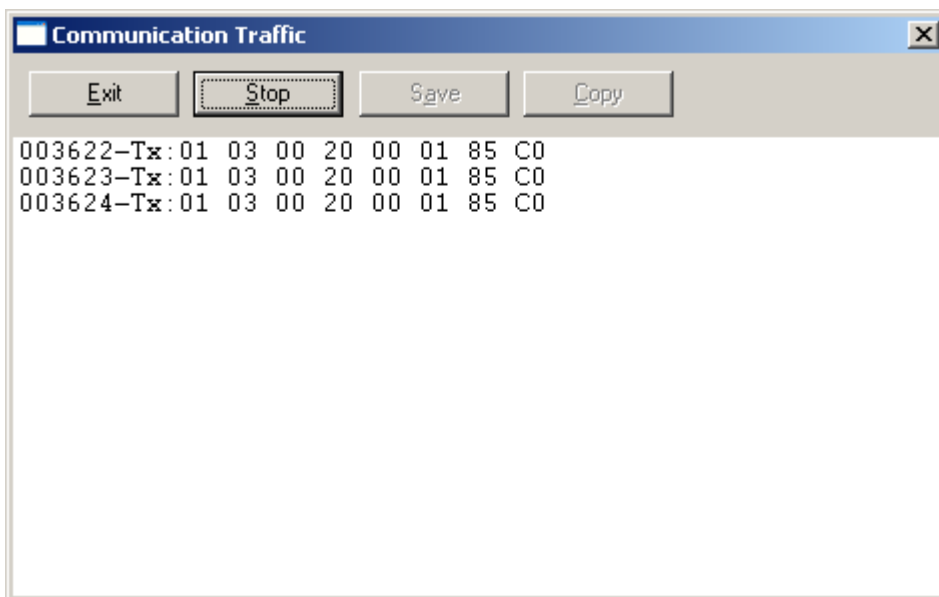
Buttons: Exit, Stop, Save, Copy

```
000065-Tx: 01 03 00 02 00 02 65 CB
000066-Rx: 01 03 04 00 0C 8B 42 DC F1
000067-Tx: 01 03 00 02 00 02 65 CB
000068-Rx: 01 03 04 00 0C 8B 42 DC F1
```

Whenever something is wrong with the incoming message (in the following example the requested length of the parameter is incorrect), the receiver answers with an error-message. In that case the datastream will look something like this:



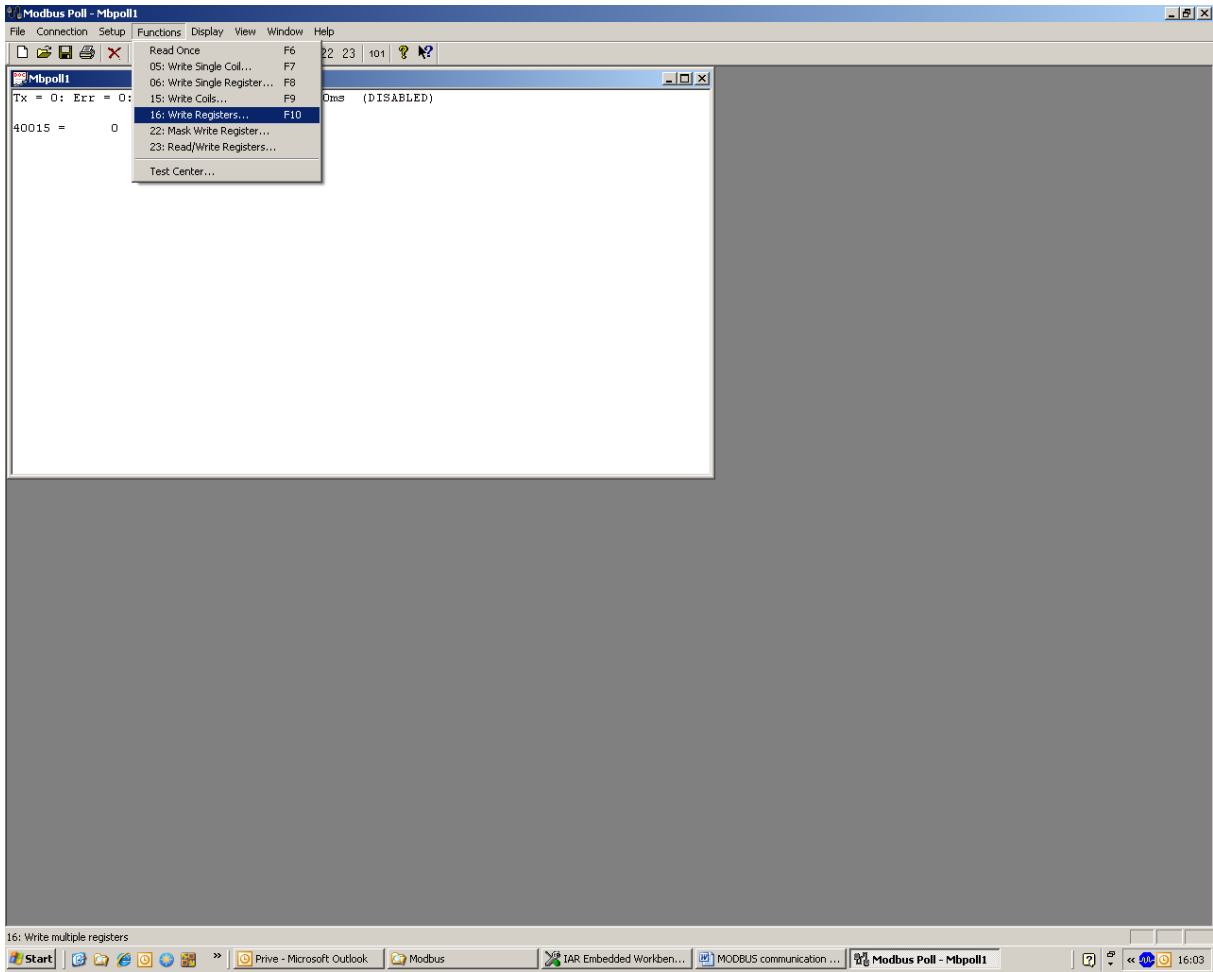
Whenever the receiver somehow does not receive any message, no answers will be sent. This will result in the following datastream



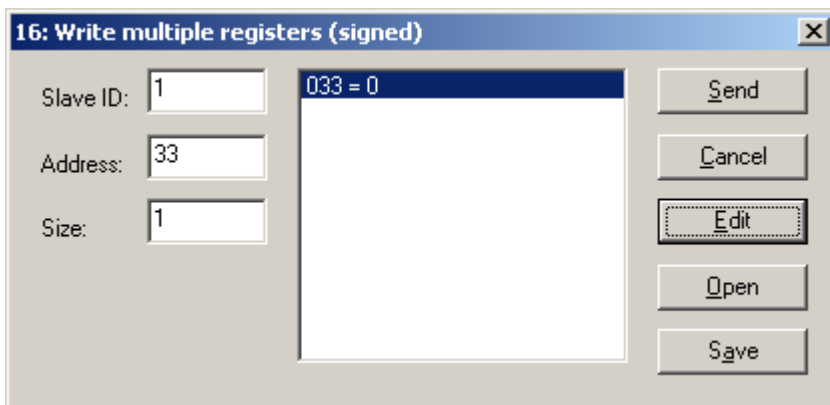
## Testing the receive-path

In order to test if the F1xx is at least receiving a message correctly, you could try to change the unit-indication (eg. 'm3' ) on the LCD (if the unit has one !).

For an F110, in the 'Functions' menu, select 16: Write Registers:



Select address 33 Total Unit



Then press 'Edit' and you can change the value of register 33 by pressing 'Send'  
If the change-message has been received correctly, the unit-indication on the LCD will be changed accordingly.



## Problems with USB and/or signal-converters

When using signal-converters to interface between eg. RS232 and RS485 or RS422, sometimes problems occur due to delay between switching from send to receive of these converters. Especially the combination of a converter and a USB-com port (as used for laptop-applications) can give problems because of the total delay that is introduced in the system.

There is one option you could try if you think this could be causing the problems. There is a possibility to set a communication delay in the F1xx. This delay is then inserted between receiving a message and responding to it.

The delay-register is no. 14. The value that is entered is in mS. The default value is 0, we suggest you start by trying a value 100.

