

# Modbus Communication with GWAB

Skreven av	Ruben Jönsson
Datum	2010-05-02
Revision	- (Not revised)
Dokumentnamn	GWABModbus.doc

AB Liros Electronic  
Box 9124  
200 39 Malmö  
Tel: 040-142080  
Fax: 040-947388  
Email: [mail@liros.se](mailto:mail@liros.se)

# Table of contents

---

<b>1. Introduction .....</b>	<b>3</b>
<b>2. The Modbus protocol .....</b>	<b>4</b>
2.1. Limitations in the GWAB Modbus protocol .....	4
2.2. Serial line settings.....	4
2.3. Exception codes .....	5
2.4. Function code 03 – Read Holding Registers .....	5
2.5. Function code 07 – Read Exception Status.....	10
2.6. Function code 17 – Report Slave ID.....	10
<b>Appendix A – Modbus device address in GWAB .....</b>	<b>11</b>
<b>Appendix B – Determining a sensorline address.....</b>	<b>12</b>

# 1. Introduction

---

This document describes how to communicate with the GrainWatch component GWAB01/02 using serial communication and the Modbus RTU protocol. The description here is valid for GWAB firmware V1.13 and later.

The GWAB is a device that acquires measurement data from grain storage facilities using GrainWatch sensors or group of sensors. A group of sensors, usually temperature sensors, are evenly mounted, at intervals of 2-3 meters on wires that is placed inside a protective, steel reinforced conduit cable. This cable is designed to be suspended in the ceiling of a silo and thereby taking measurements on the grain throughout the entire height of the silo. This cable is called a sensorline in the GrainWatch terminology and in the rest of this document.

The sensor types used in a GrainWatch system can be for measurement of humidity, weight, digital on/off signals and the most common which is for temperature measurement. V1.13 of the Modbus protocol only supports temperature sensors.

The GWAB01 has an RS485 serial port and can operate as a slave device according to the Modbus RTU communication protocol on a half duplex RS485 bus. The GWAB02 has an RS232 port and can operate as the single Modbus slave device on this port.

One sensorline can contain up to 32 sensors and one GWAB can be connected to up to 68 sensorlines. The GWAB has four inputs for sensorlines and these can be multiplexed using a device called GWNET01.

For more information about the GrainWatch components, please consult the individual installation manuals.

The Modbus protocol in the GWAB is implemented according to the documents "Modbus application protocol specification V1.1b" and "Modbus over Serial Line Specification and Implementation Guide V1.02". The complete specification for this protocol can be found as public documents on the internet at <http://www.modbus.org>

## 2. The Modbus protocol

---

### 2.1. Limitations in the GWAB Modbus protocol

Due to some limitations in the hardware of the GWAB, there are some differences between the actual implementation of the protocol in the GWAB and the official Modbus protocol specification:

- The Modbus protocol specification says that a slave device can have an address between 1 and 247. The GWAB uses 2 rotary switches with 10 positions each for communication protocol settings. Only two positions on one of the switch are used for the Modbus device address which means that device addresses between 0-19 can be set. Since device address 0 can't be used, the GWAB device address can be set to 1-19.
- According to the Modbus protocol specification, a slave device should be able to handle read requests for up to 125 16 bit registers per request. In the GWAB this is limited to a maximum of 77 registers per request.
- The Modbus Diagnostic function (function code 08) is not implemented in this version of the firmware.
- A Modbus RTU end of message frame is recognized by the GWAB with more than 3.5 character lengths of silence on the communication line as the protocol specifies. The Modbus protocols also specifies that if there is a delay between characters for more than the time for 1.5 characters but less than the time for 3.5 characters then the message should be discarded. This is not implemented in the firmware instead all characters up until the 3.5 character delay time is added to a message frame and handled as one message at the end of the frame. Messages are always sent with less than 1.5 character delays between characters though.

### 2.2. Serial line settings

When the GWAB is set to use the Modbus protocol, its serial ports have the following settings:

19200 bits per second  
1 startbit  
8 databits  
1 parity bit – even parity  
1 stop bit

These settings are fixed in the GWAB firmware and can not be changed by user settings or Modbus commands.

The Modbus protocol can use two transmission modes, either ASCII or RTU. The GWAB uses Modbus RTU.

## 2.3. Exception codes

When the GWAB receives a Modbus message with some syntactical or logical errors in it, it replies with an exception code instead of the requested data. Note that the GWAB does not reply anything unless the device address for it is matching, the CRC is correct and the message contains no framing error. This is all according to the Modbus specifications. The following exception codes can be replied from the GWAB:

Exception code	Name	Meaning
01	Illegal Function	Response for all function codes except 03 – “Read Holding Registers”, 07 – Read Exception Status and 17 – “Report Slave ID”.
02	Illegal Data Address	Response when a register that is not available in the GWAB Modbus register address range is requested.
03	Illegal Data Value	Response when the implied length of a message is not correct or when the requested number of registers for a multiple number of register operation exceed 77.

## 2.4. Function code 03 – Read Holding Registers

Data from the sensors connected to the GWAB is read with the Modbus function code 03 – Read Holding Registers.

Every sensorline or other type of sensor or group of sensors connected to the GWAB, either directly or via GWNET01's, has an identification number that uniquely identifies the sensors connected to a GWAB. This identification number is called the address of the sensorline. Since each channel in the GWAB can be connected to 17 sensorlines (16 connected through the multiplexers of the GWNET01 and 1 connected directly to the GWAB) and since a GWAB has 4 channels a sensorline can have an address between 0 and 67.

The address of a sensorline is determined by the setting of a selector switch in the GWNET01 that the sensorline is connected to and also by which input in the GWNET01 that is used. For sensorlines that are not connected through the multiplexers in a GWNET01, the address is entirely determined by which input in the GWAB it is connected to. See also Appendix B – Determining sensorline addresses.

Besides the temperatures or other sensor data, some registers with information about the internal state of the GWAB are also available through the Modbus protocol. These are useful when evaluating the readings from the sensors, especially just after power on or reset of the GWAB since it can take quite some time for the GWAB to become ready to deliver sensor data after it has been powered on. The length of this start-up time is entirely depending on how many GWNET01's and sensorlines that are connected to the GWAB. It can take up to one minute and even more after power on or reset before the GWAB is

ready to deliver data from all connected sensors in a large system. During the first part of this time the GWAB is resolving the addresses for all connected sensorlines and before this is completed, the GWAB can't know if a sensorline with the requested address even exists for the current setup. Before the address has been resolved, the GWAB will reply as if the sensorline with this address did not exist. The holding registers for internal status of the GWAB can then be used to get more information about the sensorline.

In the tables below, the registers have two addresses; one is the Modbus device address shown as a decimal value with the offset 40001 added to it. This is usually the value used to describe the location of the register when making a program for a PLC. The other is the address as it appears in the Modbus message on the serial line. This is shown as a hexadecimal value.

Modbus device register	Message register (Hex)	Description
40001	0000	<p>Current status for channel 1, which can be one of the following values:</p> <ol style="list-style-type: none"> <li>1 GWNET's and sensor lines are completely initialized and at least one sensor line has been found and is completely read.</li> <li>2 Initialization for this channel is finished but no sensorlines has been found. If there are sensorlines and/or GWNET's connected to this input, something is wrong with cables, sensorlines, GWNET's or the GWAB.</li> <li>3 The channel is being initialized. All connected network multiplexers in connected GWNET's are being identified.</li> <li>4 The channel is being initialized. All the addresses of the network multiplexers connected to this channel are being resolved according to the selectorswitch in the GWNET.</li> <li>5 The channel is being initialized. All the addresses of the network multiplexers in the GWNET's have been resolved but no sensorline has been sampled yet. If there are no sensorlines connected to any of the inputs in the GWNET's connected to this channel, the status will be 5 continuously.</li> </ol> <p>The value for this register is the same as the blink sequence for the green LED1 STATUS NET1 on the GWAB motherboard.</p> <p>This register can be used to examine the initialization process for channel 1 after power on or reset.</p>
40002	0001	As above for channel 2
40003	0002	As above for channel 3
40004	0003	As above for channel 4

40005	0004	<p>Current value of the internal state for channel 1. Only the high byte in the register is useful and it can have one of the following values:</p> <p>0x01 Searching for network multiplexers in connected GWNET01's after power on or reset</p> <p>0x02 Mapping the network multiplexers in the connected GWNET01's to GrainWatch sensorline addresses after power on or reset.</p> <p>0x03-0x04 Reading temperature sensors</p> <p>0x05-0x06 Reading sensors in combined humidity and temperature sensorlines.</p>
40006	0005	As above for channel 2.
40007	0006	As above for channel 3.
40008	0007	As above for channel 4.
40009 – 40025	0008 – 0018	<p>An array of 17 registers containing the GrainWatch sensorline addresses of each found network multiplexer or sensorline connected directly to the GWAB input for channel 1. These registers can be used to see which GrainWatch addresses that is located by this channel and after power on or reset, at which stage the GWAB is in the resolving of the addresses. Note that even though an address for a network multiplexer has been found, it is not sure that there actually is a sensorline connected to this multiplexer.</p> <p>These registers can have the following values:</p> <p>-3 The GWAB has just started to scan for network multiplexers and sensorlines but has not found any yet. Only appears very shortly after power on or reset.</p> <p>-2 Has found a sensorline or network multiplexer but does not know its address yet.</p> <p>-1 Finished searching, no address in this position in the array</p> <p>0-3 Found sensorline directly connected to the GWAB input. 0 is for channel 1, 1 for channel 2, 2 for channel 3 and 3 for channel 4.</p> <p>4-67 Found network multiplexer with this address which may or may not have any sensorline connected to it.</p> <p>68-71 Found a combined humidity and temperature sensorline connected to this channel. 68 is for channel 1, 69 for channel 2, 70 for channel 3 and 71 for channel 4.</p> <p>Combined humidity and temperature sensorlines have to be connected directly to the GWAB input, without any GWNET01's or other sensorlines also connected to this input. This means that when a combined humidity and temperature sensorline is connected to the GWAB, its address (68-71) is the first value in the register array and all other registers are set to -1.</p> <p>The addresses in the array are not in any specific order.</p>
40026 – 40042	0019 – 0029	As above for channel 2.
40043 – 40059	002A – 003A	As above for channel 3.
40060 – 40076	003B – 004B	As above for channel 4.

40077	004C	A counter that increments once every half second after power on or reset. When it reaches 0xFFFF it stops there.
40078 – 40256	004D – 00FF	Not currently used. Read as 0.



40257 – 40320	0100 – 013F	<p>64 registers with data for up to 63 sensors in a standard temperature sensorline with GrainWatch address 0 (connected directly to channel 1).</p> <p>The first register is a status register for the entire sensorline and can have one of the following values:</p> <ul style="list-style-type: none"> <li>0 Found a network multiplexer in a GWNET01 or a sensorline connected directly to the GWAB input for this address and its sensors are sampled OK.</li> <li>1 No network multiplexer or sensorline found for this address</li> <li>2 Found a network multiplexer in a GWNET01 or a sensorline connected directly to the GWAB input for this address but its sensors are not yet sampled after power on or reset.</li> <li>3 Found a network multiplexer in a GWNET01 for this address but no sensors are connected to it.</li> <li>4 The network multiplexer in a GWNET01 for this address has reported that its sensorline input is shorted.</li> <li>5 The GWAB has initially found a network multiplexer in a GWNET01 for this address but it is not responding anymore.</li> <li>6 The GWAB has initially found a network multiplexer in a GWNET01 for this address but it can not connect to it anymore because of communication errors.</li> </ul> <p>The 63 following registers contains the sampled temperature data for each sensor on the sensorline, starting with the lowest sensor on the line (at the bottom of the silo). The values can also be an error code for the sensor and the registers can have one of the following values:</p> <ul style="list-style-type: none"> <li>-1792 – 2047 The temperature as it is sampled from the sensor. A 12 bit 2-complement value sign extended to 16 bits. In hexadecimal format this is F900-07FF. The value is the temperature in degrees Celsius multiplied by 16. Divide the register value with 16 to get the actual temperature in degrees C. Even if the register value and the sensor allow a resolution of up to 0.0625 degrees C, the GWAB samples the sensor with a resolution of 0.5 degrees C.</li> <li>-2048 – -1793 The 256 lowest values in the 12 bit range of the sensor value is reserved for an error code if the temperature can not be read from the sensor. In hexadecimal format this is F800-F8FF. Only the following values are used: <ul style="list-style-type: none"> <li>-1793 (F8FF) No sensor has been found for this position in the sensorline.</li> <li>-1804 (F8F4) The GWAB is initialized but no sensor has yet been read for this sensorline. Only present while linestatus is 2.</li> <li>-2048 (F800) When a network multiplexer without any sensors connected to it just has been initialized the error code can be set to this value. After some samplings of this address the value will switch to -1793. For all practical purposes, this value can be treated just like -1793.</li> </ul> </li> </ul> <p>Note that the Modbus protocol has room for 63 sensors for one temperature sensorline but the maximum number of sensors with the current hardware is actually limited to 32 sensors.</p>
------------------	----------------	---

40321 – 44608	0140 – 11FF	<p>Same as above for sensorlines with addresses 1-67.</p> <p>The Modbus register address for the status value for one temperature sensorline is the sensorline address * 64 + 256.</p> <p>The Modbus register address for a temperature sensor in a sensorline is the sensorline address * 64 + 257 + sensor position, where the lowest sensor has sensor position 0.</p>
------------------	----------------	---

## 2.5. Function code 07 – Read Exception Status

Function code 07 is implemented in the GWAB but it always returns an exception code 0 for now.

## 2.6. Function code 17 – Report Slave ID

Report Slave ID replies with the following response:

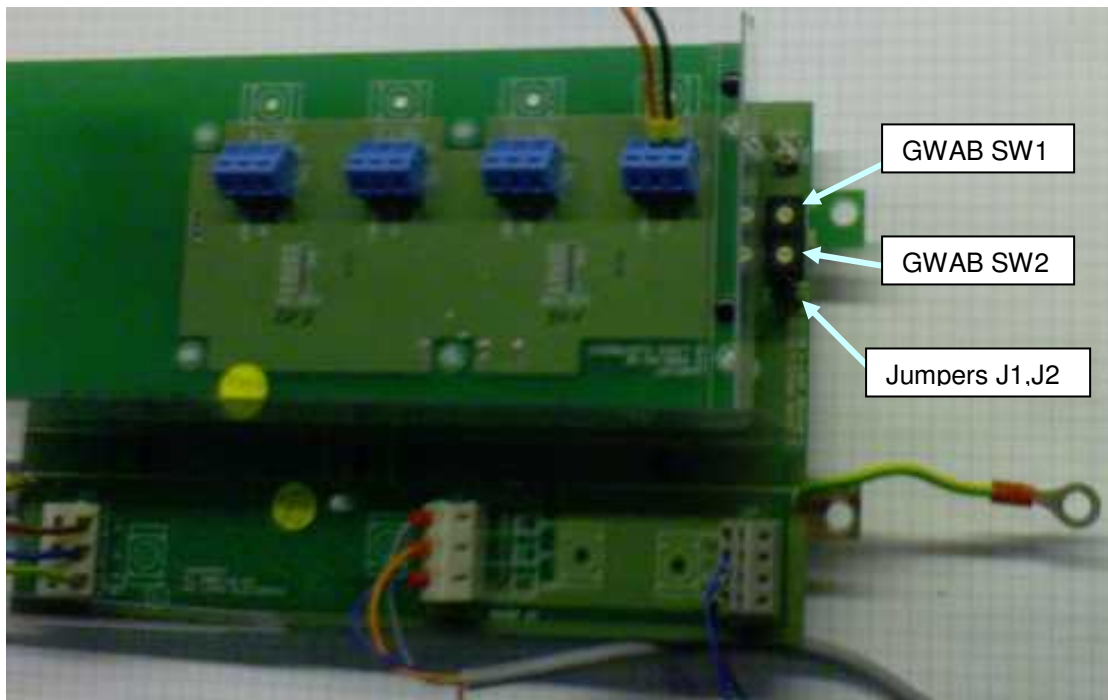
Byte offset	Field name	Value
0	Device address	Device address set with selector switches
1	Function	0x11
2	Bytecount	4
3	Slave ID	Same as device address
4	Run indicator status	0xFF
5	Major Version	Major part of version number 0x01 for v1.13
6	Minor Version	Minor part of version number 0x0D for v1.13
7	CRC Lo	Low byte of 16 bit CRC
8	CRC Hi	High byte of 16 bit CRC

## Appendix A – Modbus device address in GWAB

---

This appendix describes how set the Modbus device address for a GWAB.

The device address is selected with two 10 position rotary switches in the GWAB.



SW2 determines which protocol to use. Position 4 and 5 is used for the Modbus protocol. SW1 sets the device address to one of 0 – 9 when SW2 is in position 4 and one of 10 – 19 when SW 2 is in position 5. SW1 set to 0 and SW2 set to 4 is not valid since it results in the device address 0, which is used for broadcast messages with Modbus.

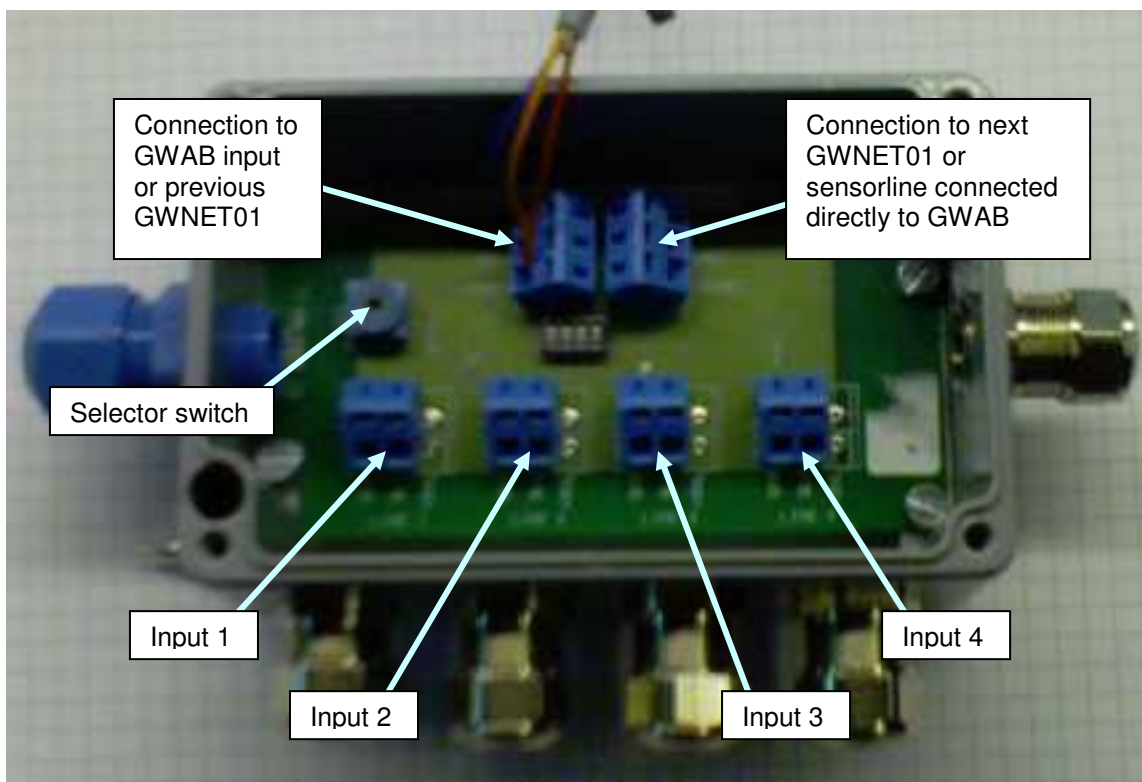
Jumpers J1 and J2 are used to connect the RS485 line termination resistor. This resistor should only be connected for the last Modbus device on the RS485 bus. The shorting jumper should be set to position 1 – 2 in J1 and J2 for the last device on the bus and in position 2 – 3 for all other devices.

## Appendix B – Determining a sensorline address

This appendix describes in detail how to determine the address for a GranWatch temperature sensorline connected to a GWAB, with or without GWNET01's.

A GrainWatch sensorline can be connected to a GWAB input channel in two ways – either directly to the input channel terminal or through a network multiplexer which allows more than one line to be connected to one GWAB input channel.

The GWNET01 is a device which contains 4 network multiplexers with 4 inputs for temperature sensorlines. It also has one selector switch with 16 positions. Up to 4 GWNET01's can be connected in series to one GWAB input channel. If one or more GWNET01's are used on one input channel in the GWAB, one sensorline can still be connected directly to the GWAB input channel but then it is physically connected to the last GWNET01. This means that the last GWNET01 can be connected to 5 sensorlines instead of 4.



The GrainWatch address for a sensorline directly connected to the GWAB is determined by the following formula:

Address=GWAB input -1

Where the GWAB input is numbered from 1 to 4.

The GrainWatch address for a sensorline connected through network multiplexers in the GWNET01 is determined by the following formula:

$$\text{Address} = \text{Selector switch} * 4 + \text{GWNET01 input} - 1$$

Where the GWNET01 input is numbered from 1 to 4. When the selector switch is set to 0, the value 16 instead of 0 should be used in the formula. Note that the selector switch has its value printed in hexadecimal notation, which means that A=10, B=11, C=12, D=13, E=14 and F=15.

When the address is known, the corresponding Modbus register number for the first register for that sensorline equals to  $40257 + \text{address} * 64$ . The message register address equals to  $0x0100 + 0x40 * \text{address}$ .

The following tables shows all possible address combinations for sensorlines connected to one GWAB and the corresponding register number for the status register for that sensorline in the Modbus protocol.

#### Sensorlines connected directly to GWAB inputs

GWAB input	Sensorline address	Modbus device register	Message register (hex)
1	0	40257	0100
2	1	40321	0140
3	2	40385	0180
4	3	40449	01C0

#### Sensorlines connected via network switches in GWNET01's.

GWNET selector switch	GWNET input	Sensorline address	Modbus device register	Message register (hex)
0	1	64	44353	1100
0	2	65	44417	1140
0	3	66	44481	1180
0	4	67	44545	11C0
1	1	4	40513	0200
1	2	5	40577	0240
1	3	6	40641	0280
1	4	7	40705	02C0
2	1	8	40769	0300
2	2	9	40833	0340
2	3	10	40897	0380
2	4	11	40961	03C0
3	1	12	41025	0400

3	2	13	41089	0440
3	3	14	41153	0480
3	4	15	41217	04C0
4	1	16	41281	0500
4	2	17	41345	0540
4	3	18	41409	0580
4	4	19	41473	05C0
5	1	20	41537	0600
5	2	21	41601	0640
5	3	22	41665	0680
5	4	23	41729	06C0
6	1	24	41793	0700
6	2	25	41857	0740
6	3	26	41921	0780
6	4	27	41985	07C0
7	1	28	42049	0800
7	2	29	42113	0840
7	3	30	42177	0880
7	4	31	42241	08C0
8	1	32	42305	0900
8	2	33	42369	0940
8	3	34	42433	0980
8	4	35	42497	09C0
9	1	36	42561	0A00
9	2	37	42625	0A40
9	3	38	42689	0A80
9	4	39	42753	0AC0
A	1	40	42817	0B00
A	2	41	42881	0B40
A	3	42	42945	0B80
A	4	43	43009	0BC0
B	1	44	43073	0B00
B	2	45	43137	0C40
B	3	46	43201	0C80
B	4	47	43265	0CC0
C	1	48	43329	0D00
C	2	49	43393	0D40
C	3	50	43457	0D80

C	4	51	43521	0DC0
D	1	52	43585	0E00
D	2	53	43649	0E40
D	3	54	43713	0E80
D	4	55	43777	0EC0
E	1	56	43841	0F00
E	2	57	43905	0F40
E	3	58	43969	0F80
E	4	59	44033	0FC0
F	1	60	44097	1000
F	2	61	44161	1040
F	3	62	44225	1080
F	4	63	44289	10C0

Rev	Datum	Av	Revision	Godk.
	100502	RJ	Created	RJ