

**5100
DIGITAL
INDICATOR**

**Modbus
Communications
Manual**

**For use with Software
Versions 1.2x +**

1. INTRODUCTION..... 1

2. CONNECTION OF THE 5100 NETWORK 2

 2.1 RS232 CONNECTION2

 2.2 RS485/RS422 CONNECTION2

3. MODBUS REGISTER DEFINITIONS FOR THE 5100 3

 3.1 HOLDING REGISTERS3

 3.2 INPUT REGISTERS4

 3.3 COMMANDS (IMPLEMENTED AS COILS)5

4. MODBUS ASCII COMMUNICATIONS 6

 4.1 INTRODUCTION6

 4.2 FRAMING6

 4.3 ADDRESS FIELD6

 4.4 FUNCTION FIELD.....6

 4.5 DATA FIELD.....6

 4.6 LRC FIELD6

 4.7 EXCEPTION RESPONSES7

5. COMMAND DETAILS 8

 5.1 FUNCTION 01: READ COIL STATUS8

 5.2 FUNCTION 03: READ HOLDING REGISTERS9

 5.3 FUNCTION 04: READ INPUT REGISTERS10

 5.4 FUNCTION 05: FORCE SINGLE COIL11

 5.5 FUNCTION 06: PRESET SINGLE REGISTER11

 5.6 FUNCTION 15 FORCE MULTIPLE COILS12

 5.7 FUNCTION 16 PRESET MULTIPLE REGISTERS13

1. Introduction

This manual details the Modbus communications option of the Ranger 5100. The Modbus communications option is a separate software license that may be installed Ranger at the time of purchase or installed afterwards via the use of the CONFIG menu. Details of the 5100 licensing options and how to install them may be found in the Reference Manual.

The Modbus protocol was originally developed for use by Modicon programmable controllers but has been adopted by many manufacturers as a common protocol standard.

The Modicon standard specifies two transmissions modes: ASCII or RTU.

The 5100 supports the ASCII transmission mode only.

Information in this manual details the underlying communications protocol but in practice this is generally looked after by the PLC directly. All that is normally required for setup is to specify the addresses of the registers of interest to your application. Much of the data from the 5100 is implemented as 2 consecutive registers to make up a complete 32 bits of data. However unless the 5100 has been setup with a fullscale of more than 30000 units it is possible to use the lower order 16 bits contained in a single register.

2. Connection of the 5100 Network

2.1 RS232 Connection

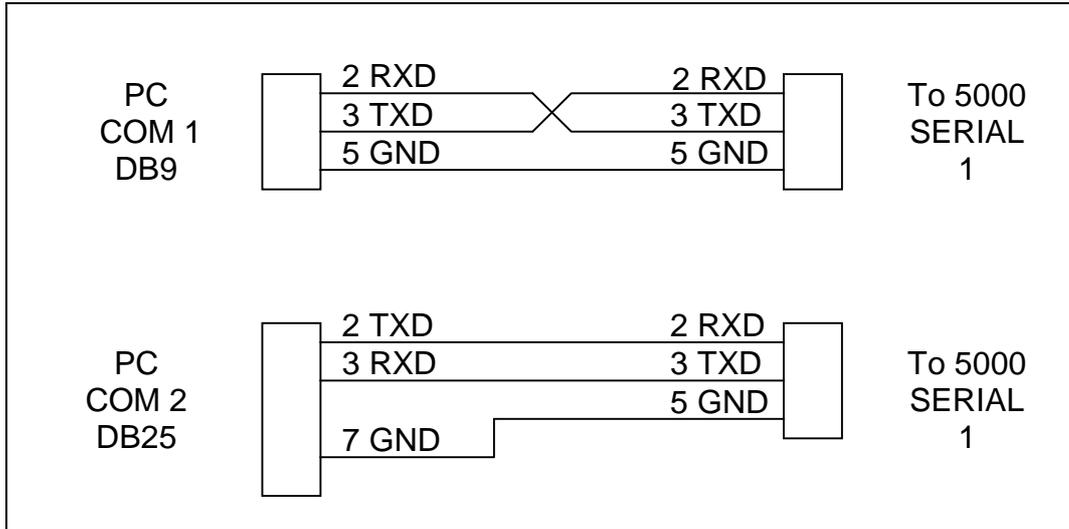


Figure 1: Connection of a single 5100 unit to an IBM PC either using COM1 or COM2 on the PC.

2.2 RS485/RS422 Connection

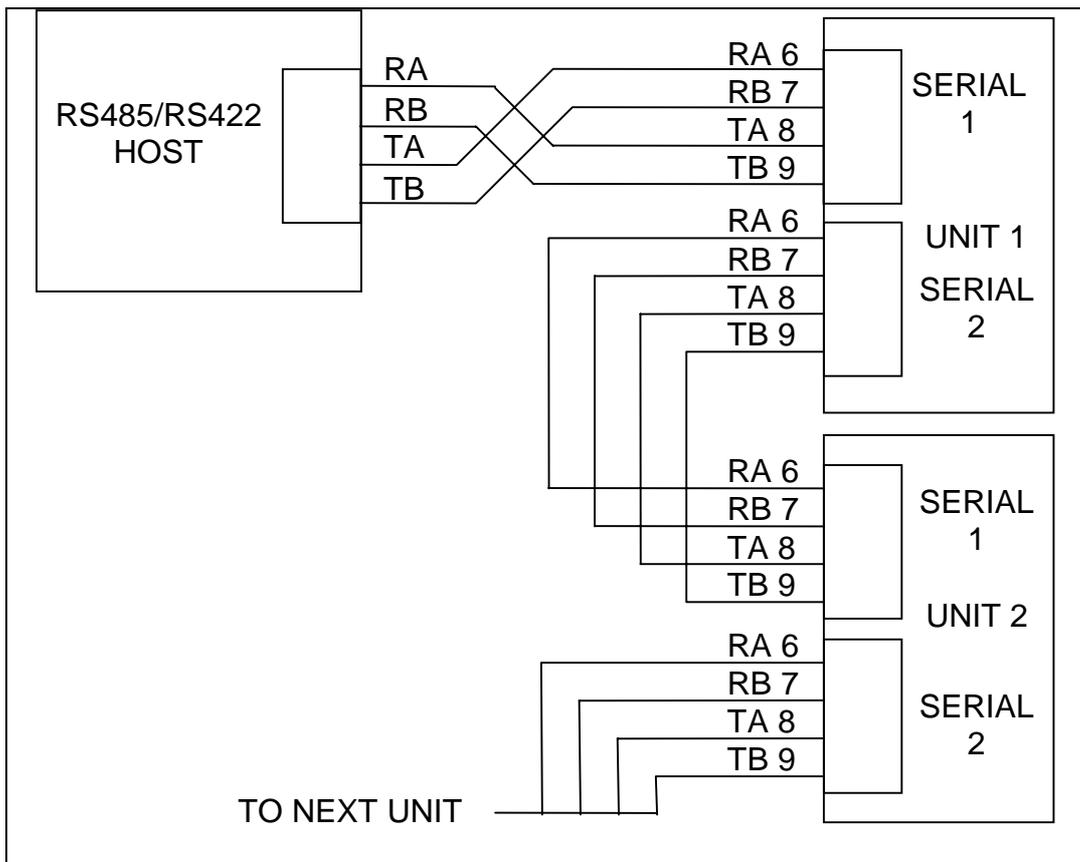


Figure 2: Connection of a RS485/RS422 network.

3. MODBUS Register Definitions for the 5100

The 5100 implements a number of holding registers, input registers and coils. Generally all that is required to configure your MODBUS master to work with the 5100 is to specify the address of the 5100 unit, and the addresses of the registers that you are interested in. Following is a list of the 5100 MODBUS registers.

3.1 Holding Registers

Holding Registers in the 5100 are as follows:

| Register | Contents |
|-----------------|---|
| 40001 | Current Recipe |
| 40002 | Recipe Proportion |
| 40003 | Inflight 1 High 16 bits |
| 40004 | Inflight 1 Low 16 bits |
| 40005 | Inflight 2 High 16 bits |
| 40006 | Inflight 2 Low 16 bits |
| ... | ... |
| 40051 | Inflight 25 High 16 bits |
| 40052 | Inflight 25 Low 16 bits |
| 40053 | Current Recipe Preset Tare High 16 bits |
| 40054 | Current Recipe Preset Tare Low 16 bits |
| 40055 | Current Recipe Target 1 High 16 bits |
| 40056 | Current Recipe Target 1 Low 16 bits |
| 40057 | Current Recipe Target 2 High 16 bits |
| 40058 | Current Recipe Target 2 Low 16 bits |
| ... | ... |
| 40103 | Current Recipe Target 25 High 16 bits |
| 40104 | Current Recipe Target 25 Low 16 bits |
| 40105 | Tolerance/Hysteresis 1 High 16 bits |
| 40106 | Tolerance/Hysteresis 1 Low 16 bits |
| 40107 | Tolerance/Hysteresis 2 High 16 bits |
| 40108 | Tolerance/Hysteresis 2 Low 16 bits |
| ... | ... |
| 40153 | Tolerance/Hysteresis 25 High 16 bits |
| 40154 | Tolerance/Hysteresis 25 Low 16 bits |
| 40155 | Recipe ID name (leftmost character) |
| ... | ... |
| 40160 | Recipe ID name (rightmost character) |
| 40161 | Print Header line 1 leftmost character |
| ... | ... |
| 40180 | Print Header line 1 rightmost character |
| 40181 | Print Header line 2 leftmost character |
| ... | ... |
| 40200 | Print Header line 2 rightmost character |

3.2 Input Registers

Input Registers in the 5100 are as follows:

| Register | Contents |
|----------|---|
| 30001 | Setpoint Outputs 9 to 24 (Bit 15 is Output 24, Bit 0 is Output 9) |
| 30002 | Reading Status |
| 30003 | Unused (Always = 0) |
| 30004 | Reading Error |
| 30005 | Displayed Reading High 16 bits |
| 30006 | Displayed Reading Low 16 bits |
| 30007 | Gross Reading High 16 bits |
| 30008 | Gross Reading Low 16 bits |
| 30009 | Net Reading High 16 bits |
| 30010 | Net Reading Low 16 bits |
| 30011 | Peak Displayed Reading High 16 bits |
| 30012 | Peak Displayed Reading Low 16 bits |
| 30013 | Unused (Always = 0) |
| 30014 | Number of Batches or items in the total for current recipe |
| 30015 | Current Recipe Total High 16 bits |
| 30016 | Current Recipe Total Low 16 bits |
| 30017 | Current Recipe Total Time High 16 bits |
| 30018 | Current Recipe Total Time Low 16 bits (time in 0.1 second increments) |
| 30019 | Current Recipe Absolute Error High 16 bits |
| 30020 | Current Recipe Absolute Error Low 16 bits |

The contents of the reading status register are as follows:

| Bit | Function |
|-----|--------------------------|
| 15 | Output 8 |
| 14 | Output 7 |
| 13 | Output 6 |
| 12 | Output 5 |
| 11 | Output 4 |
| 10 | Output 3 |
| 9 | Output 2 |
| 8 | Output 1 |
| 7 | Reading within Zero Band |
| 6 | Range or Interval 2 |
| 5 | Gross reading |
| 4 | Motion |
| 3 | Centre of Zero |
| 2 | Underload |
| 1 | Overload |
| 0 | Instrument Error |

3.3 Commands (implemented as coils)

MODBUS Coils are used in the 5100 to implement specific operator commands. A single coil is set to trigger the function. These coils are self clearing once the function has been executed.

It is possible to configure the operation of the 5 remote keys (See the Reference Manual for details of the available functions). Each of the remote keys has a normal function triggered by a single press and a second function which is triggered by a long 2 second key press. Coils 5 ..12 simulate the operation of these physical buttons. In this way it is possible to operate these extended functions of the 5100 without the need for the Setpoint accessory card or the physical buttons.

The 5100 commands available as coils are as follows:

| Coil | Function |
|-------------|-------------------------|
| 00001 | Zero |
| 00002 | Tare |
| 00003 | Gross/Net |
| 00004 | Print |
| 00005 | Function Key |
| 00006 | Function Key long press |

| Coil | Function |
|-------------|------------------------------------|
| 00010 | Remote Key 1 function single press |
| 00011 | Remote Key 2 function single press |
| 00012 | Remote Key 3 function single press |
| 00013 | Remote Key 4 function single press |
| 00014 | Remote Key 5 function single press |
| 00020 | Remote Key 1 function long press |
| 00021 | Remote Key 2 function long press |
| 00022 | Remote Key 3 function long press |
| 00023 | Remote Key 4 function long press |
| 00024 | Remote Key 5 function long press |

4. MODBUS ASCII Communications

4.1 Introduction

With Modbus ASCII transmission each 8 bit byte of the message is transmitted as 2 ASCII Hex characters ('0' .. '9', 'A'..'F') for each 4 bits of the byte.

4.2 Framing

The ASCII message starts with a colon(:) character (ASCII 3A hex) and ends with a carriage return line feed (CRLF) pair (ASCII 0D, 0A hex). Delays of up to one second can elapse between characters in a message. A typical message Frame is shown below:

| START | ADDRESS | FUNCTION | DATA | LRC CHECK | END |
|-------|-------------|-------------|----------------|-------------|------|
| ':' | 2 HEX CHARS | 2 HEX CHARS | 2* N HEX CHARS | 2 HEX CHARS | CRLF |

4.3 Address Field

The Modbus protocol supports slave addresses in the range of 0 .. 247 but the 5100 supports slave addresses in the range of 0..31. The address 0 is reserved for broadcast addressing, and the slave addresses range from 1..31.

4.4 Function Field

Valid functions codes range from 1..255. The 5100 supports the following functions:

- 01: Read Coil Status
- 03: Read Holding Registers
- 04: Read Input Registers
- 05: Force Single Coil
- 06: Preset Single Register
- 15: Force Multiple Coils
- 16: Preset Multiple Registers

4.5 Data Field

The data field is constructed of pairs of hexadecimal digits. The size of the data field depends on the message function

4.6 LRC Field

The Longitudinal Redundancy Check (LRC) is calculated on the message contents excluding the leading colon and trailing CRLF characters. The LRC is calculated by adding together each of the

bytes in the message, discarding any carries, and then two's complementing the result. Each pair of hexadecimal digits represents a single byte of data. The 8 bit LRC is sent as two hexadecimal characters.

4.7 Exception Responses

Except for broadcast messages to which there is no response from a slave unit, when the master sends a message there is an expected response from the slave unit. However a number of different events can occur which makes it impossible for the slave unit to respond in the way expected.

If the masters message is corrupted and can't be deciphered or if the check sum is incorrect the slave unit ignores the message and the master will timeout and try again.

If the slave receives a message that it can decipher but it cannot respond to because it does not support that type of message or the registers specified do not exist etc. then there is a standard exception response to warn the master and prevent it from endlessly re-trying the same errant message.

The format of an exception response is as follows:

| START | ADDRESS | FUNCTION | Exception Code | LRC CHECK | END |
|-------|-------------|-------------|----------------|-------------|------|
| “:” | 2 HEX CHARS | 2 HEX CHARS | 2 HEX CHARS | 2 HEX CHARS | CRLF |

The Address is the slave's address. The function field is the function code of the master's message with the most significant bit set to indicate the exception status. For example function 03 hex becomes exception 83 hex.

The exception codes supported by the 5100 are as follows:

- 01 ILLEGAL FUNCTION This function is not supported by the 5100.
- 02 ILLEGAL DATA ADDRESS The register address range is outside the range of registers supported by the 5100.
- 03 ILLEGAL DATA VALUE The data sent is out of range for the particular registers that were addressed.

5. Command Details

5.1 Function 01: Read Coil Status

This function reads the status of the discrete coils in the 5100. All such coils in the 5100 are implemented as one-time commands and are internally cleared. This function therefore always returns the OFF status for these coils. The function is included to make the 5100 compatible with systems that insist on reading the status of all coils automatically. Broadcast is not supported.

See section 0 page 4 for a list of all of the coils.

Here is an example of a command to read all 24 coils from the 5100.

| Field Name | Example (Hex) |
|---------------------|----------------------|
| Slave Address | 02 |
| Function | 01 |
| Data: | |
| Starting Address Hi | 00 |
| Starting Address Lo | 00 |
| Number of points Hi | 00 |
| Number of points Lo | 18 |

Response

The status of each coil is returned as one bit per coil in the data field.

Here is an example of a response to the above query.

| Field Name | Example (Hex) |
|---------------------|----------------------|
| Slave Address | 02 |
| Function | 01 |
| Data: | |
| Byte Count | 02 |
| Data (Coils 8..1) | 00 |
| Data (Coils 16..9) | 00 |
| Data (Coils 24..17) | 00 |

5.2 Function 03: Read Holding Registers

This function reads the contents of holding registers (Addressed as 4xxxx) in the slave. Broadcast is not supported.

Query

The query specifies the starting register and the number of registers to read. Registers are addressed starting at zero - so register 40001 is addressed as zero.

Here is an example of a request to read registers 40003 to 40004 from slave address 2.

| Field Name | Example (Hex) |
|---------------------|---------------|
| Slave Address | 02 |
| Function | 03 |
| Data: | |
| Starting Address Hi | 00 |
| Starting Address Lo | 02 |
| Number of points Hi | 00 |
| Number of points Lo | 02 |

Response

The contents of the response include the slave address and function so the master can verify that the response is from the correct unit. Also included is a byte count followed by the block of data requested.

Here is an example of a response to the above query.

| Field Name | Example (Hex) |
|-------------------|---------------|
| Slave Address | 02 |
| Function | 03 |
| Data: | |
| Byte Count | 04 |
| Reg 40001 Data Hi | 00 |
| Reg 40001 Data Lo | 00 |
| Reg 40002 Data Hi | 03 |
| Reg 40002 Data Lo | E8 |

This example reads the full 32 bit contents of setpoint target 1 represented as the two 16 bit registers 40001 and 40002. The target 1 value in this example is 1000 kg or (000003E8 hex).

5.3 Function 04: Read Input Registers

This function reads the contents of the input registers (Addressed as 3xxxx) in the slave. Broadcast is not supported.

Query

The query specifies the starting register and the number of registers to read. Registers are addressed starting at zero - so register 30001 is addressed as zero.

Here is an example of a request to read registers 30005 to 30006 from slave address 2.

| Field Name | Example (Hex) |
|---------------------|----------------------|
| Slave Address | 02 |
| Function | 04 |
| Data: | |
| Starting Address Hi | 00 |
| Starting Address Lo | 04 |
| Number of points Hi | 00 |
| Number of points Lo | 02 |

Response

The contents of the response include the slave address and function so the master can verify that the response is from the correct unit. Also included is a byte count followed by the block of data requested.

Here is an example of a response to the above query.

| Field Name | Example (Hex) |
|-------------------|----------------------|
| Slave Address | 02 |
| Function | 04 |
| Data: | |
| Byte Count | 04 |
| Reg 40001 Data Hi | 00 |
| Reg 40001 Data Lo | 00 |
| Reg 40002 Data Hi | 29 |
| Reg 40002 Data Lo | 42 |

This example reads the full 32 bit contents of the current displayed reading represented as the two 16 bit registers 30005 and 30006. The weight reading value in this example is 10562 kg or (00002942 hex).

5.4 Function 05: Force Single Coil

This function forces a single coil in a slave unit. When broadcast (address = 0) the same coil is set in all units. Coils are addressed as 0xxxx.

Query

The query specifies the coil address and whether the coil is to be forced on or off. Coils are addressed starting at zero - so coil 00001 is addressed as zero. A data value of FF00 hex forces a coil on and a value of 0000 forces the coil off.

Here is an example of a request to Tare the reading of slave address 2.

| Field Name | Example (Hex) |
|-----------------|---------------|
| Slave Address | 02 |
| Function | 05 |
| Data: | |
| Coil Address Hi | 00 |
| Coil Address Lo | 02 |
| Force Data Hi | FF |
| Force Data Lo | 00 |

Response

The normal response of a force coil request is to echo the full contents of the message.

5.5 Function 06: Preset Single Register

This function sets the contents of holding registers (Addressed as 4xxxx) in the slave. When broadcast (address = 0) the same register is set in all units.

Query

The query specifies the holding register and the data to be written to it. Registers are addressed starting at zero - so register 40001 is addressed as zero.

See Section 3.1 page 3 for details of the available holding registers.

Here is an example of a command to set register 40004 to 1500 kg on slave address 2.

| Field Name | Example (Hex) |
|---------------------|---------------|
| Slave Address | 02 |
| Function | 03 |
| Data: | |
| Register Address Hi | 00 |
| Register Address Lo | 03 |
| Register Data Hi | 05 |
| Register Data Lo | DC |

Response

The normal response of the preset register command is to echo the full contents of the message.

5.6 Function 15 Force Multiple Coils

This function performs the same task as function 5 but allows multiple coils to be set and reset in the same message. In the 5100 it is only possible to set one coil at a time so this message is not strictly necessary as it is easier to use Function 05. However not all systems implement the forcing of a single coil using Function 05, some use function 15. When broadcast the same coils are forced in every slave unit.

Query

See Section 0 page 4 for details of the available coils.

Here is an example of a command to set coil 1 and trigger a zero command.

| Field Name | Example (Hex) |
|--------------------|---------------|
| Slave Address | 02 |
| Function | 0F |
| Data: | |
| Coil Address Hi | 00 |
| Coil Address Lo | 00 |
| Number of coils Hi | 00 |
| Number of Coils Lo | 0C |
| Number of bytes | 02 |
| Data (Coil 8..1) | 01 |
| Data (Coil 12..9) | 00 |

Response

The normal response of the force multiple coils command is to send the slave address, function code, starting address and quantity of coils forced.

Here is an example of a reply to the above command.

| Field Name | Example (Hex) |
|--------------------|---------------|
| Slave Address | 02 |
| Function | 0F |
| Data: | |
| Coil Address Hi | 00 |
| Coil Address Lo | 00 |
| Number of coils Hi | 00 |
| Number of Coils Lo | 0C |

5.7 Function 16 Preset Multiple Registers

This function sets the contents of holding registers (Addressed as 4xxxx) in the slave. When broadcast (address = 0) the same register is set in all units. This function is the same as Function 06 except that multiple registers may be set at once.

Query

The query specifies the holding registers and the data to be written to them. Registers are addressed starting at zero - so register 40001 is addressed as zero.

See Section 3.1 page 3 for details of the available holding registers.

Here is an example of a command to set inflight 1 (register 40003 & 40004) to 100 kg and inflight 2 (register 40005 & 40006) to 0 kg on slave address 2.

| Field Name | Example (Hex) |
|------------------------|---------------|
| Slave Address | 02 |
| Function | 10 |
| Data: | |
| Starting Address Hi | 00 |
| Starting Address Lo | 02 |
| Number of Registers Hi | 00 |
| Number of Registers Lo | 04 |
| Number of bytes | 08 |
| Register 1 Hi | 00 |
| Register 1 Lo | 00 |
| Register 2 Hi | 00 |
| Register 2 Lo | 64 |
| Register 3 Hi | 00 |
| Register 3 Lo | 00 |
| Register 4 Hi | 00 |
| Register 4 Lo | 00 |

Response

The normal response of the Function 16 is to send the slave address, function code, starting address and quantity of coils forced.

Here is an example of a reply to the above command.

| Field Name | Example (Hex) |
|------------------------|---------------|
| Slave Address | 02 |
| Function | 10 |
| Data: | |
| Starting Address Hi | 00 |
| Starting Address Lo | 02 |
| Number of registers Hi | 00 |
| Number of registers Lo | 04 |