Appendix K: MODBUS Protocol Support (if installed)

The Spider has optional support for MODBUS over TCP/IP. If this option is purchased the Spider will act as a slave and accept connections from a management system on TCP port 502 via the LAN interface or dial-up.

The Spider only allows one connection at a time; if a second connection attempt is made the original connection will be closed. The Spider can handle approximately 10 MODBUS requests per second. The TCP layer will allow a small amount of queries to be queued.

For LAN access, the Spider **must be configured to run in “Always On Mode”** as otherwise the Spider will periodically power down to save power.

For dial-up access the management system should dial in via the Spider’s data number, make the required MODBUS requests over the link and then hang up the connection. “Always On” is not needed for this method.

In the following discussion register addresses will be numbered starting from 0 as they are in the MODBUS protocol; however it is common for SCADA software to use point addresses specified with a 5- or 6-digit number where the first digit gives the type of register and the remainder is the register address plus one. E.g. the point address “300013” corresponds to input register 12. For this reason the examples will also give the point address in parenthesis.

When requesting a range of registers, the first register in the range must be valid or exception 02: Illegal Data Address will be returned. In general it is ok for the last register to be outside the valid address range – the invalid registers will simply return dummy data (value returned depends on function).

Reading / Changing Control Outputs (Function 01: Read Coils / Function 05: Write Single Coil)

For both of these functions, the address 0-7 corresponds to Control 1-8. A ‘1’ or ‘ON’ status means that the control output is closed.

Setting a control’s status has the same effect and priority as changing the control via the Spider’s webpage. Therefore the Spider’s built-in alarms (if enabled) will override a change made via MODBUS.

Example:

Control 3 can be read/written in register 2 (000003)

Obtaining Input Readings

Function 02: Read Discrete Inputs

This function returns an on/off logic state depending on the voltage level on the corresponding wired input, and is independent of the configuration of the input.

Valid addresses are 0-7. The returned value is inverted – a 1 means the input is grounded (i.e. the connected switch is closed). Invalid addresses will return ‘0’.

Example:
Input 3 is available in register 2 (100003)

Function 04: Read Input Registers

This function returns the reading that is displayed on the Spider’s screen or web pages for the corresponding input (wired or wireless). If the input is configured as “Analogue” then the value is returned as an IEEE754 32-bit floating-point number. If the input is a “Counter” or “Utility” type, the number is r a 32-bit integer which is the displayed value multiplied by 1000.

As MODBUS registers are only 16-bit wide but the data is 32-bits, two consecutive registers are used, with the least significant 16 bits in the lower numbered register. **It is very important that all reads start on an even numbered register and end on an odd numbered register (e.g. registers 0 to 3).** If this condition is not fulfilled the Spider will return exception 02: Illegal Data Address, as otherwise invalid data could be read.

*NOTE: Invalid or disabled inputs will return ‘1’ for every bit position – this corresponds to “**Negative Not a Number**” (–qNaN) in the IEEE754 standard and should be detected as an error.*

The address of the first register in the pair for a particular input is determined from the following table:

Input channel	First register of pair	Value
Native inputs 1-8	X * 2 - 2 where X is the input number (1-8)	Depends on input configuration – same as Spider’s display
Charger	16	0.0 if off, 1.0 if on
Battery voltage	18	Reading in volts
Unit temperature	20	Reading in degrees Celsius
Wireless input	8182 + X * 8 + Y * 2 Where X is the transmitter number (1-100), Y is the input number on the transmitter (1-4)	Depends on input configuration – same as Spider’s display

Examples (the point address given is the first of the pair):

Native input 3 is available in registers 4-5 (300005)
The second wireless input on transmitter 5 is available in registers 8224-8225 (308225)

Obtaining Input Status (Function 03: Read Holding Registers)

In addition to the current input reading, each input has some associated status information available as a single 16-bit holding register per input. Alarm status is also available via this method.

This information is particularly useful for wireless inputs as it notifies the user of low transmitter battery, input tampering (for WaveFlow) and whether the last wireless communication failed. A ‘1’ in the specified position indicates that a fault is detected.

NOTE: isolated communication errors do not necessarily indicate a fault with the transmitter. It is recommended that a delay of at least two log periods is added to alarms checking this kind of fault to eliminate false alarms.

To simplify alarm conditions, an extra status register (address 8191) is available which combines all the enabled wireless inputs – this register contains a ‘1’ if ANY enabled wireless input shows the fault. It is recommended that users periodically check this register for a value greater than 511 as it indicates a fault with an input.

The address of the register corresponding to a particular input and the encoding of the data is determined from the following table:

Input channel	**First register of pair**	**Value**
Native inputs 1-8	X – 1 where X is the input number (1-8)	Low byte = Type Code
Alarm 1-8	X + 127 where X is the alarm number (1-8)	Bits 3-0 = SMS status code Bits 11-8 = Email status code Bit 15 = Alarm Trigger Active
Wireless input (specific)	4091 + X * 4 + Y Where X is the module number (1-50), Y is the input number on the module (1-4)	Low byte = Type Code Bit 8 = Communication error Bit 9 = Low transmitter battery Bit 10 = Tamper detected
Any wireless input	8191 (408192)	Bit 8 = Communication error Bit 9 = Low transmitter battery Bit 10 = Tamper detected

Examples:

Alarm 3’s status is available in register 130 (400131)
The second wireless input on module 5 is available in register 4113 (404114)

Many of the inputs give a “Type Code” as part of their response this is decoded as follows:

Type Code	**Input Type**
0	Disabled
1	Digital
2	Counter
3	Analogue
7	Utility Meter
8	Wireless Counter
9	Wireless Utility Meter
10	Wireless Analogue

Supported MODBUS Function Codes

Use of any function not listed will generate exception 01: Illegal function.

Function code (decimal)	Use in Spider
01: Read Coils	Get on/off status of control outputs
02: Read Discrete Inputs	Get on/off status of inputs (regardless of configuration)
03: Read Holding Registers	Get status information about configured inputs and alarms
04: Read Input Registers	Get current reading of configured inputs – all values returned as either IEEE754 32-bit floating point or 32-bit integer as a pair of consecutive registers – both address and number of registers in request must be even
05: Write Single Coil	Change on/off state of control output – identical outcome to webpage access
43/14: MEI / Read Device Identification (Basic)	Gives product name and software version.