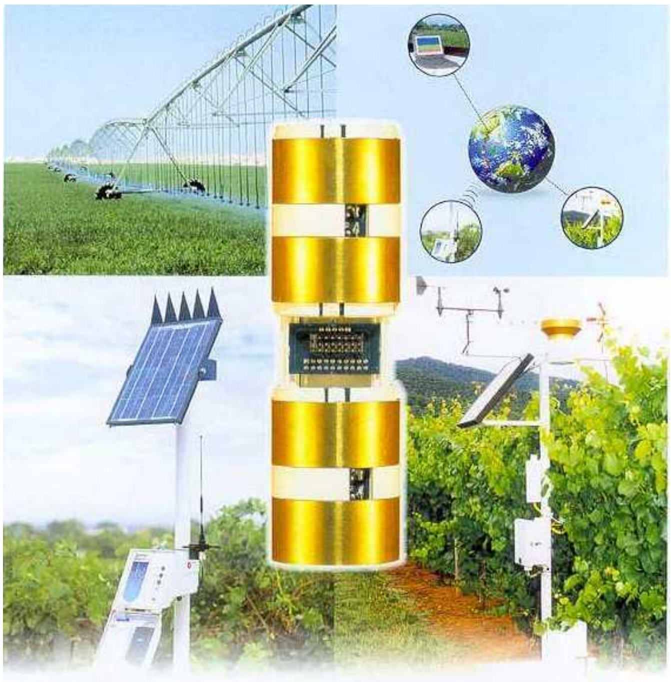
RS232/485 MODBUS



Series II Hardware Manual

Version 3.5

For firmware revision 1.2.3 and above)

Details of the Modbus specification can be obtained from "The Modbus Organization" at www.modbus.org

Sentek Pty Ltd
A.C.N. 007 916 672
77 Magill Road
Stepney, South Australia 5069
Phone: +61 8 8366 1900
Facsimile: +61 8 8362 8400
Internet: www.sentek.com.au

Modbus Rev 3.5 (2012-02-13)
Sentek Spec 2.1.0

EnviroSCAN and EasyAG – Statements of Compliance

FCC note of compliance and statement of liability

Electro-Magnetic Compliance

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorientation or relocation of the receiving antenna.
- Connection of the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consultation with the dealer or an experienced radio/TV technician.

EMC approvals

The EnviroSCAN 232/485 and EasyAG 232/485 Series II probes comply with standard "EN61326:1997 Amendment 1:1998, Amendment 2:2001, Amendment 3:2003 EMC standard for equipment for measurement, control and laboratory use”

The equipment complies with the following specifications:

- EN55011: 1998, Amendment 1: 1999 Radiated and conducted emissions
- EN61000-4-2: 1995 Immunity to Electrostatic Discharge (ESD)
- EN61000-4-3: 2002 Immunity to Radiated Fields (RF)
- EN61000-4-4: 1995 Immunity to Electrical Fast Transients (EFT)/ Bursts
- EN61000-4-5:1995 Immunity to Surges
- EN61000-4-6:1996 Immunity to Conducted RF
- FCC Part 15 Class B

Marking

The above EMC approvals allow the product to be marked CE, C-tick and FCC.

Modifications

Any modifications to any part of the equipment or to any peripherals may void the EMC compliance of the equipment.

Radio Interference

The probe is not to be operated in free air as it may cause interference to radio communication devices

Table of Contents

Table of Figures

The Sentek EnviroSCAN and EasyAG MODBUS Probe Interfaces

This section provides information about the **EnviroSCAN** and **EasyAG** Modbus probe interfaces.

References

- Sentek TriSCAN™ Agronomic User Manual
- Sentek Calibration Manual
- Sentek Probe Configuration Utility User Guide
- Modicon Modbus™ Protocol Reference Guide, PI-MBUS-300 Rev J, June 1996
  MODBUS Over Serial Line FOR LEGACY APPLICATIONS ONLY
  www.modbus-ida.org/specs.php

What are the EnviroSCAN and EasyAG Modbus probe interfaces?

The **EnviroSCAN** and **EasyAG** Modbus probe interfaces are used to allow an RS485 or RS232 - Modbus compatible device to communicate with and retrieve data from multiple Sentek sensors.

The **EnviroSCAN** and **EasyAG** Modbus probe interfaces behave as a slave (DTE – Data Terminal Equipment) device, meaning it only responds to requests from a Modbus master (DCE – Data Communication Equipment) device. When instructed to sample data, the probe interface will retrieve values from each sensor configured on the probe. These values are returned to the Modbus master device upon request.

There are two variants of probe type; EnviroSCAN and EasyAG. The EnviroSCAN probe has user configurable sensor depths and can be installed to great depths. The EasyAG is a smaller diameter probe, compared to the EnviroSCAN, and is more suited to shallow rooted crops and short term applications. Both have an interface card at the top which reads the sensors and provides the Modbus communication (when Sentek Modbus firmware is installed).

The probe and its sensors are configured using the Probe Configuration Utility software.

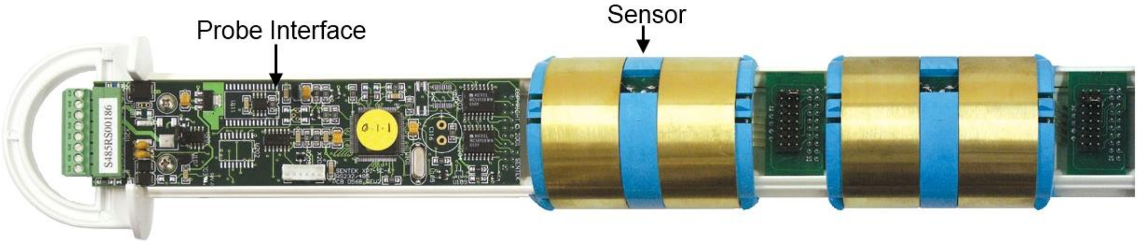


Figure 1 EnviroSCAN probe

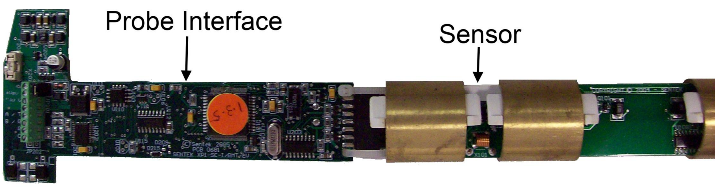


Figure 2 EasyAG probe

What is the Probe Configuration Utility?

The Probe Configuration Utility is used to configure the **EnviroSCAN** and **EasyAG** Modbus probe interfaces with sensor depth location, normalization values (air and water counts) and calibration information for each sensor installed on the probe. This information is stored in non-volatile memory on the probe interface, and is used to produce the calculated value (value that has been processed via the interfaces calibration formula) from each sensor on the probe.

Communication between the Probe Configuration Utility and the **EnviroSCAN** and **EasyAG** Modbus probe interface is done using the **EnviroSCAN** and **EasyAG** Probe Configuration Utility cable from a computer's communication port to the probe interface's TTL port.

Note: Temperature Sensors are installed on some interfaces. These are currently not supported on the output port of Modbus probe interfaces.

Note: Information stored in non-volatile memory will not be lost when power is removed from the probe interface.

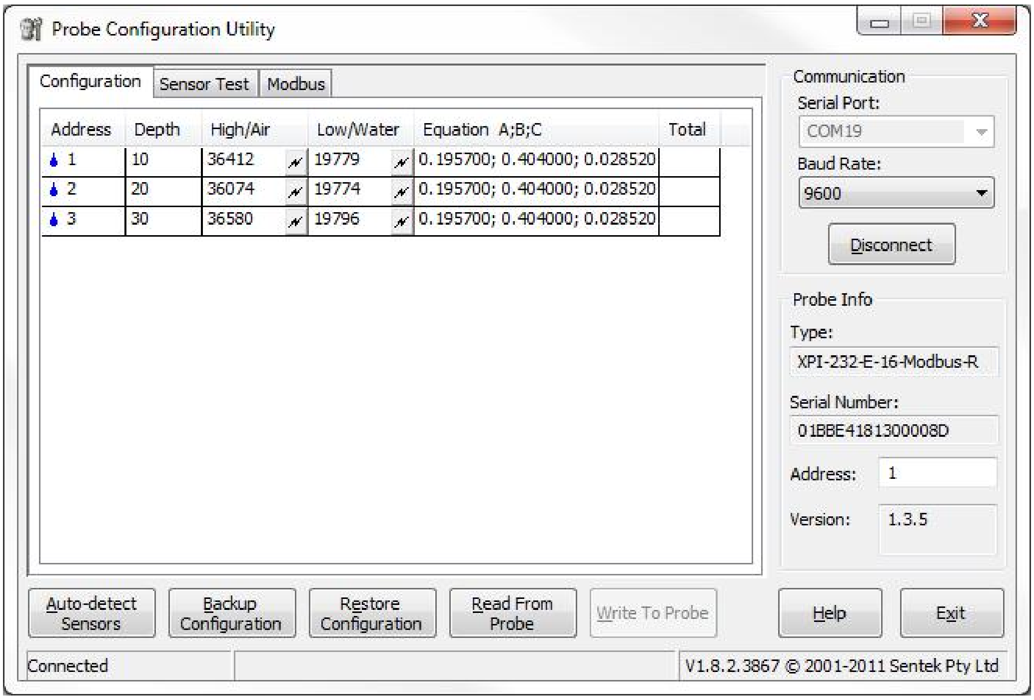


Figure 3 Probe Configuration Utility software

Setting up the probe interface

The Probe Configuration Utility is provided to configure the **EnviroSCAN** and **EasyAG** Modbus probe interfaces with depth location, normalization values (air and water counts) and calibration.

The probe interface must be fully configured before sensors can be read through the Modbus output. The sections below describe each of the areas of configuration on a Sentek Modbus interface.

Why do I need to configure the probe interface?

The **EnviroSCAN** and **EasyAG** Modbus probe interface must be configured to ensure valid information is reported to the Modbus master device when data is requested. This information includes the number, depth and type of sensors, normalization values (air and water counts), calibration information and Modbus communication parameters.

Warning:

Incorrect configuration information stored in the EnviroSCAN and EasyAG Modbus probe interface will result in incorrect calculated readings being reported to the Modbus master device.

For more information on setting up the **EnviroSCAN** and **EasyAG** Modbus probe interface, refer to the Probe Configuration Utility online help.

Configuration Page

Auto-detect Sensors will detect all sensors currently installed on the probe. After the sensors are detected the configuration information (including type of sensor) will be displayed in the Probe Configuration list.

After Auto-detecting sensors, the sensor depths need to be entered in the **Depth** Column.

Maximum (air) and minimum (water) sensor values, within a Sentek access tube, need to be obtained and saved in the probe. This process is called normalization and is explained in Sentek Distributor training documents. The normalization process is necessary to adjust for any variances that may occur during the production of the sensor.

The **Equation** column is used to store A, B and C coefficients used in the calibration equation, which determines the final value output by the interface. The coefficients are entered in A, B then C order, separated by semicolons. See the Sentek Calibration Manual for more information.

Setting the Modbus Address

The Modbus address for the probe can be set in the **Address** field on the right hand side of the PConfig screen. The address of the probe should be in the range “1” to “247”.

Changing the Modbus Port Settings

The **Sentek** Probe Configuration Utility (Version 1.2 or later) is used to configure probe address, sensor settings, and the Modbus communication settings. The default settings are RTU mode, 9600 baud, no parity, and 500ms turn-around time. Each field is described in the Probe Configuration Utility Help file.

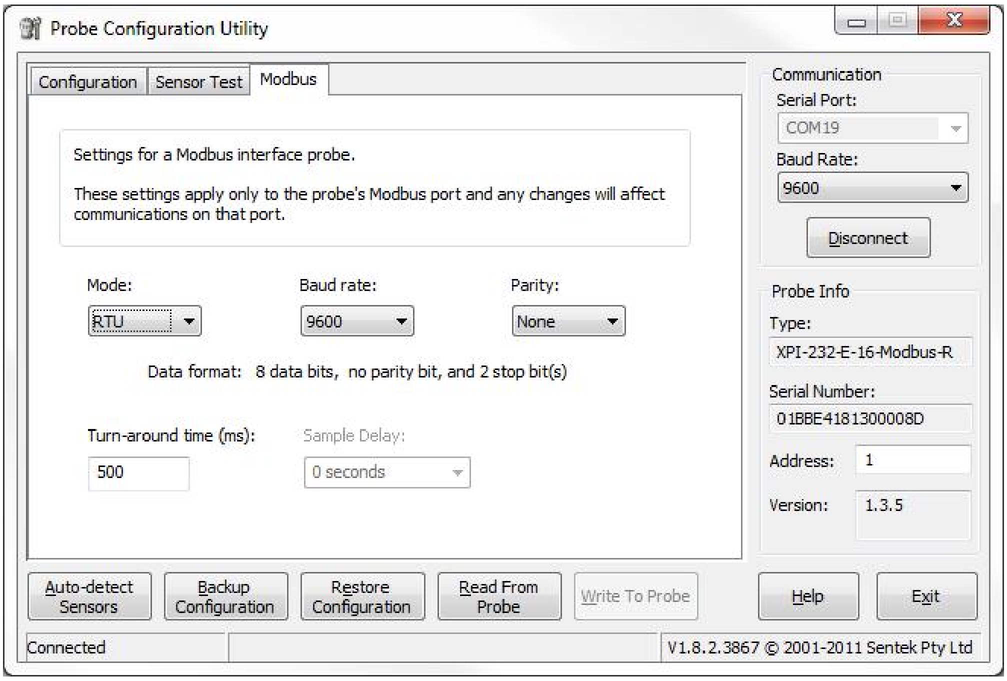


Figure 4 Modbus Configuration Page

Modbus Communication

About the Modbus Communication

The **Sentek** implementation of the Modbus protocol allows initiation of scanning of **Sentek** sensors, either all or selected, and the recovery of calculated values.

The implementation is based on the document “Modicon Modbus Protocol Reference Guide” PI-MBUS-300 Rev J. This document is available at www.modbus.org.

The Modbus Protocol

The Modbus protocol is a master/slave command protocol in which the master sends a query packet to the slave, which performs the requested operation and responds.

Probe interfaces can be configured to support either RTU mode or ASCII mode.

RTU Mode

RTU Mode is a binary mode with 8-bit data bytes being transmitted. Each transmitted byte is in fact 11 bits, consisting of one start bit, 8 data bits, an optional parity bit, and either one (if parity is used) or two (if no parity) stop bits. Parity may be selected as even or odd.

Each query packet from the master consists of a device address (one byte), function code (one byte), optionally one or more 8-bit data bytes, and an error check CRC (two bytes). The packet is delimited by timing, where at least 3.5 characters of silence must precede the start of the packet. No more than 1.5 character times are permitted between characters in the packet, and the packet terminates on at least 3.5 character times of silence.

Start	Address	Function	Data	CRC	End
T1-T2-T3-T4	8 bits	8 bits	*n* x 8 bits	16 bits	T1-T2-T3-T4

Figure 5 RTU message frame

ASCII Mode

ASCII Mode packets consist of readable characters with 7-bit data bytes being transmitted. Each transmitted byte is in fact 10 bits, consisting of one start bit, 7 data bits, an optional parity bit, and either one (if parity is used) or two (if no parity) stop bits. Parity may be selected as even or odd.

The actual packet data in ASCII mode is the same as for RTU except for the error check and each byte being coded as two ASCII hexadecimal digits. A Packet commences with a colon (“:”) character and is terminated with a <CR><LF> pair, with up to 1 second permitted between characters within a packet. In ASCII mode, the error check consists of a two-character “Longitudinal Redundancy Check” formed by adding all characters (starting with the address byte) and forming the twos complement. The lower byte of the sum is encoded as two hexadecimal characters.

ASCII mode transfers data at about 50% of the speed of RTU mode, but has the advantage of being less timing sensitive, and in addition it is easy to capture and decode the packets.

Start	Address	Function	Data	LRC	End
1 char (‘:’)	2 chars	2 chars	*n* chars	2 chars	2 chars (<CR><LF>)

Figure 6 ASCII message frame

Turn-around time

The turn-around time is the minimum time, in milliseconds, before the probe will start sending the response (maximum 1000 ms, minimum 0 ms). The actual minimum is dependent on the time required for the probe to process the request.

Turn-around time applies only to RS485 devices or half-duplex radio devices on RS232.

The probe turn-around time must be set higher than the time it takes the master device to switch from Tx to Rx. Failure to do so may result in the probe transmitting data before the master device is ready to receive data, resulting in loss of data. The master device must be ready to receive data once the turn-around time has elapsed.

Note: This field is disabled if the probe firmware version does not support turn-around time.

Modbus Overview

The Modbus protocol models the interface as a set of 16-bit registers that can be read or written by a Modbus master device.

Note: An attempt to read or write register numbers outside the supported ranges will result in a command being rejected with 'Illegal data address'.

Input Registers

These are read-only registers. The **EnviroSCAN** and **EasyAG** Modbus probe interface supports input registers 30001 through 30020, 30100, 30101 through 30228, and 30257 through30448.

Note: Some of these registers are reserved for future use, and if read will respond with zero.

Holding Registers

These are registers that can be read and written. The **EnviroSCAN** and **EasyAG** Modbus probe interface supports holding registers 40001 through 40007.

Note: When writing to multiple Holding Registers, the higher address registers are written to first, this permits a single write to set the mask and initiate the reading process.

Sleep Mode

The **EnviroSCAN** and **EasyAG** Modbus probe interface will enter into sleep mode after 15 seconds of inactivity on either the TTL Communications port, or the RS232/RS485 port. The interface will wake when any communication is received to either port.

Note: Communication received whilst the interface is in sleep mode, will not be recognized as a valid command. The master device will need to retry any communication that may wake the interface.

Commands

A Modbus query can set the values of one or more Holding Registers, or interrogate the values of one or more Input or Holding Registers. For example, the command (in ASCII mode)

:01 03 00 00 00 02 FA

requests the reading of Holding Registers 40001 through 40002 from probe 1, while

:00 06 00 00 00 02 F8

is a broadcast command (i.e. has a slave address of zero) that will preset Holding Register 40001 to the value 2, which will start a scan of all moisture sensors. As the results become available, they are stored in Input Registers starting at 30257, with two consecutive 16-bit registers holding a single floating point (32-bit) value, high bytes being stored in the lower address.

Broadcast Commands

Commands sent with a slave address of 00 are Broadcast Commands. In a multi-drop environment (RS-485) all connected slaves will receive the command, but make no response.

Modbus Commands

The probe interface supports the following subset of Modbus commands:

Function Code	Name	Notes	Broadcast
03h	Read Holding Registers	Status of measurement process Maximum of 32 in one read	No
04h	Read Input Registers	Selections of sensors, cooked data, configuration of sensors Maximum of 32 in one read	No
06h	Preset Single Register	Control measurement process	Yes
08h	Diagnostic	Subfunctions: 0 - return query data 1 - restart communications 3 - Change ASCII input delimiter	No
0Bh	Fetch Comm Event Counter		No
10h	Preset Multiple Registers	Maximum of 32 in one write	Yes
11h	Report Slave ID		No

Figure 7 Implemented Functions

Note: In reading and presetting multiple registers, a maximum of 32 registers can be read or set with a single command because of limits in maximum command/response length supported.

Read Holding Registers

The command

:01 03 00 00 00 02 FA

is parsed as

01 Slave Address
03 Command - Read Holding Registers
00 00 Address; add 40001 to get the absolute address
00 02 Number of registers
FA LRC

The slave will respond with a packet such as

:01 03 04 01 03 02 77 7B

which is to be interpreted as

01 Slave Address
03 Function code
04 Data byte count
01 03 Contents of 40001
02 77 Contents of 40002
7B LRC

A maximum of 32 registers can be requested at once. There is no response in broadcast mode.

Read Input Registers

The command

:01 04 00 02 00 03 F6

is parsed as

01 Slave Address
04 Command - Read Input Registers
00 02 Address; add 30001 to get the absolute address

00 03 Number of registers
F6 LRC

The response is similar to that for a Read Holding Registers command.

A maximum of 32 registers can be requested at once. There is no response in broadcast mode.

Preset Single Register

The command

:01 06 00 00 00 02 F8

is parsed as

01 Slave Address
06 Command - Preset Single Registers
00 00 Address; add 40001 to get the absolute address
00 02 16 bit data value
F7 LRC

The response is an echo of the command. There is no response in broadcast mode, but the register is updated and the requested action performed.

Diagnostic Command

The following sub-commands are supported by the EnviroSCAN and EasyAG Modbus probe interface.

Return Query Data

This command is simply echoed back to the Master

:01 08 00 00 A5 37 1B

is parsed as

01 Slave Address
08 Command - Diagnostic
00 00 Subfunction: Return Query Data
A5 37 Data
F6 LRC

This is simply a request to echo the data; the response should be identical to the command.
There is no response in broadcast mode.
Note that the response data is always two bytes long.

Restart Communications Option

This command causes a power on reset of the communication interface. All event counters are reset by this function.

:01 08 00 01 00 00 F6

is parsed as

01 Slave Address
08 Command - Diagnostic
00 01 Subfunction: Restart Communications
00 00 Data must be either 0000 or FF00
F6 LRC

The response is sent before the reset and is identical to the command. There is no response in broadcast mode. In either case the Modbus subsystem is reinitialized as if the device was power cycled:

- Clears measurement status
- Clears sensor selection mask
- Clears scanned sensors mask

- Clears diagnostic counters
- Resets ASCII input delimiter to <LF> (not reset by power cycling)

Change ASCII Input Delimiter

Normally, in ASCII mode, a command/reply is terminated with <CR><LF>. There may be cases where the <LF> character causes problems in communication, and it can be changed (but not the <CR>) using diagnostics sub-function 3.

:01 08 00 03 58 00 9C

is parsed as

01 Slave Address
08 Command - Diagnostic
00 03 Subfunction: Change Delimiter
58 00 Data: Delimiter is 0x58 (“X”)
9C LRC

The <LF> delimiter (for both received and transmitted data) will now be “<CR>X”; the response is sent after the change and is identical to the command. Any character (00 to FF) can be set as the delimiter; the second (low) data byte must be zero, or the command will be rejected with an Invalid Data exception.

This command should be used with care, because once changed the probe will no longer respond to commands terminated with <CR><LF>. The only way to change it back in ASCII mode is a new command with the correct delimiter; alternatively, the probe can be changed to RTU mode and a new delimiter set.

The delimiter is not reset by removing power from the probe, but is reset by Diagnostic sub-function 1 (Restart Communications Option).

Fetch Comm Event Counter

This returns a count of successfully executed queries since the probe interface was last powered on or reset.

:01 0B F4

is parsed as

01 Slave Address
0B Command - Fetch Event Counter
F4 LRC

The response might be

01 0B FF FF 01 08 ED

which is interpreted as

01 Slave Address
0B Function Code
FF FF Status: Slave is busy measuring
01 08 Message Count
F4 LRC

The status word is FFFF if the slave is in the process of taking a measurement, else 0000.

There is no response in broadcast mode.

Preset Multiple Registers

This command allows up to 32 Holding Registers to be preset. (In fact, as noted above, there are only three valid holding registers)

:01 10 00 00 00 03 06 11 11 22 22 33 33 1A

is parsed as

01 Slave Address
10 Command: Preset Multiple Registers
00 00 Address - Add 40001 to get absolute address

00 03	Register Count
06	Data Byte Count
11 11	New Value of 40001
22 22	new Value of 40002
33 33	New Value of 40003
1A	LRC

The response is the “header” part of the command

01 10 00 00 00 03 EC

which is interpreted as

01	Slave Address
10	Function Code
00 00	Address
00 03	Register Count
EC	LRC

Note: In writing to multiple Holding Registers, the higher address registers are in fact written to first; this permits a single write to both set the mask and initiate a reading.

There is no response in broadcast mode, but the registers are updated and the requested action is performed.

Report Slave ID

This requests information about the slave.

:01 11 EE

is parsed as

01	Slave Address
11	Command - Report Slave Id
EE	LRC

The response might be

:01 11 14 00 FF 53 45 4E 54 45 4B 01 00 00 01 10 D1 7A 0F 00 08 00 34 6F

which is interpreted as

01	Slave Address
11	Function Code
14	Byte Count
00	Slave ID (always 00)
FF	Run Indicator (always FF)
53 45 4E 54 45 4B	“SENTEK”
01 00 00 01	Revision Code (1.0.01)
10 D1 7A 0F 00 08 00 34	Probe Serial Number
6F	LRC

There is no response in broadcast mode.

The revision code is interpreted as

01	Major Revision
00	Minor Revision
00 01	Sub-minor Revision (2 bytes, most significant first)

Exception Responses

Illegal commands result in an exception response. The reply has the top bit of the Function Code set, and contains a one-byte exception code.

:01 81 01 7D

which is interpreted as

01 Slave Address
81 Function Code - Exception
01 Exception Code (Illegal Function)
7D LRC

Exception codes are

1 Illegal Function

2 Illegal Data Address

3 Illegal Data Value

6 Slave Busy

Register Map

This table shows all of the registers available through the Modbus interface.

The register address of specific data is dependant on the maximum number of sensors and number of sensor types. This information is configured in register 30100.

The actual register addresses can be calculated by formula, using the following information:

1. The maximum number of sensors is given by
   MaxNumSens = low byte of input register 30100
2. The number of sensor types is given by
   NumTypes = high byte of input register 30100
   If NumTypes = 0 Set NumTypes = 3
3. The three sensor types (SensType) are
   MoistureType = 1
   SalinityType = 2
   reserved = 3
4. The three mask tables (TblNum) are
   SensorSelection = 0
   SensorConfigured = 1
   SensorScanned = 2
5. Each sensor mask is packed 16 bits per word, so each mask table occupies the following number of words
   WordsPerMask = INT((MaxNumSens + 15)/16)
6. The start address of a mask for a sensor type in a table is then
   MaskStartAddr = 30003 + WordsPerMask × (TblNum × NumTypes + (SensType - 1))
   MaskLength = WordsPerMask
   The end address of the mask is
   MaskEndAddr = MaskStartAddr + WordsPerMask – 1
7. The Sensor depth range for a sensor type is
   SensorDepthAddr = 30101 + MaxNumSens × (SensType - 1)
   SensorDepthLength = MaxNumSens
8. The Cooked range for a sensor type is
   CookedAddr = 30257 + 2 × MaxNumSens × (SensType - 1)
   CookedLength = 2 × MaxNumSens
9. The start address of holding register masks is
   MaskStartAddr = 40002 + WordsPerMask × (SensType - 1)
   MaskLength = WordsPerMask

Input Registers

Absolute Address (Input registers) *See Register map formula for description of actual addresses calculations.*	**Relative Address**	**Description**
30001	0x0000	Last measurement selection This register is updated with the value from holding register 40001 when the measurement is started. It remains this value after the measurement is completed.
30002	0x0001	Measurement Status 0 - No measurement 1 - Measurement in progress 2 - Measurement completed, with no errors 3 - Measurement completed, but errors occurred An error means a sensor failed during reading or no configured sensors were selected for sampling (sensors with a depth of zero are not configured). The cooked data will have IEEE Not-a-number in the failing sensor(s). Note that Not-a-number also occurs if the raw to cooked calculation produces a value outside the valid range.
Sensors Scan Selection Masks These registers specify which sensors will be scanned when 40001 is set to 1 (measure selected sensors). These registers are set using the sensor selection mask from holding registers 40002 to 40008. The bits in this mask are in the same order as the detected sensor mask.		
Start = 30003 + WordsPerMask×(0×NumTypes+0) **Length** = WordsPerMask *(for MaxNumSens=32 it is 30003 – 30004)*	0x0002	**Moisture** Sensor Selection for current scan (Bit 0 is first sensor, Bit 1 is second sensor, Bit 16 is 17th sensor, etc)
Start = 30003 + WordsPerMask×(0×NumTypes+1) **Length** = WordsPerMask *(for MaxNumSens=32 it is 30005 – 30006)*		**Salinity** Sensor Selection for current scan (Bit 0 is first sensor, Bit 1 is second sensor, … Bit 16 is 17th sensor, etc)
Start = 30003 + WordsPerMask×(0*NumTypes+2) **Length** = WordsPerMask *(for MaxNumSens=32 it is 30007 – 30008)*		*Reserved*
Detected (and configured) Sensors Mask The bits in this mask are in sensor address order. Un-configured sensors do not appear in this mask (not detected or depth=0)		
Start = 30003 + WordsPerMask×(1×NumTypes+0) **Length** = WordsPerMask *(for MaxNumSens=32 and NumTypes=3 it is 30009 – 30010)*		Mask of detected **moisture** sensors (Bit 0 is first sensor, Bit 1 is second sensor, … Bit 16 is 17th sensor, etc)
Start = 30003 + WordsPerMask×(1×NumTypes+1) Length = WordsPerMask *(for MaxNumSens=32 and NumTypes=3* it is 30011 – 30012)		Mask of detected **salinity** sensors (Bit 0 is first sensor, Bit 1 is second sensor, … Bit 16 is 17th sensor, etc)
Start = 30003 + WordsPerMask×(1*NumTypes+2) Length = WordsPerMask *(for MaxNumSens=32 and NumTypes=3 it is 30013 – 30014)*		Reserved

Absolute Address (Input registers) *See Register map formula for description of actual addresses calculations.*	**Relative Address**	**Description**
Sampled/Scanned Sensors Mask At the start of reading all sensors type masks are cleared, regardless of requested type. A bit is set after the cooked value is put in the corresponding input register. The bits in this mask are in the same order as the detected sensor mask. This mask should only be used to indicate the progress in scanning each sensor. It does not reflect failed sensors.		
Start = 30003 + WordsPerMask×(2×NumTypes+0) Length = WordsPerMask *(for MaxNumSens=32 and NumTypes=3* it is 30015 – 30016)		Mask of sampled **moisture** sensors. This updates in real-time during a scan (Bit 0 is first sensor, Bit 1 is second sensor, … Bit 16 is 17th sensor, etc)
Start = 30003 + WordsPerMask×(2×NumTypes+1) Length = WordsPerMask *(for MaxNumSens=32 and NumTypes=3* it is 30017 – 30018)		Mask of sampled **salinity** sensors. This updates in real-time during a scan (Bit 0 is first sensor, Bit 1 is second sensor, … Bit 16 is 17th sensor, etc)
Start = 30003 + WordsPerMask×(2×NumTypes+2) Length = WordsPerMask *(for MaxNumSens=32 and NumTypes=3* it is 30019 – 30020)		Reserved
30100 (Fixed address)	0x0063	Maximum number of sensors per each type supported (low byte) and the number of types allocated in the implementation (high byte). Each type has the same number of sensors. **Note:** type allocation is only controls the register addressing method, and does not necessarily imply hardware support. Low byte **MaxNumSens**. At present this value is 32. High byte **NumTypes** At present, this value can be: 0 - (same configuration as 3) 1 - soil moisture sensors only, 2 - soil moisture and salinity sensors 3 - soil moisture sensors, salinity sensors and *reserved* sensor type In future these values may change. These values should be used to determine the beginning of the sensor type specific address space.
Sensor Depth Information These registers give the depth of each sensor (dimensionless values). Note: in version 1.0.x of the Modbus firmware these registers contained sensor address information.		
Start = 30101 + MaxNumSens×0 Length = MaxNumSens *(for MaxNumSens=32 and NumTypes=3* it is 30101 – 30132)	0x0064	**Moisture** sensor depth information: Depth of first, second, third, etc sensors. The value is multiplied by 10 to give accuracy to one fixed decimal point (e.g. “102” means “10.2”)
Start = 30101+ MaxNumSens×1 Length = MaxNumSens *(for MaxNumSens=32 and NumTypes=3* it is 30133 – 30164)		**Salinity** sensor depth information: Depth of first, second, third, etc sensors. The value is multiplied by 10 to give accuracy to one fixed decimal point

Absolute Address (Input registers) *See Register map formula for description of actual addresses calculations.*	**Relative Address**	**Description**
Start = 30101+MaxNumSens×2 Length = MaxNumSens *(for MaxNumSens=32 and NumTypes=3* it is 30165 – 30196)		Reserved
Sampled Cooked Data (IEEE floating point 32-bit number) The registers are updated with cooked data as the reading progresses. The corresponding bit in the sampled sensor mask is then updated. The master device can read cooked data for a sensor while the status register 30002=1, but only after the sensor's sampled/scanned bit mask is 1. After a read of selected sensors, sensors not selected (scanned) get a cooked value of 0.0. Valid cooked moisture values will always be in the range +0.0 to +101.0. Data that results in values in the range -0.1 to 0.0 will be returned as +0.0. Any data values outside of this range (caused by faulty sensors, incorrect probe installation or inappropriate calibration equation coefficients) will be returned as IEEE Not-a-Number (7FC00000). A failed sensor will also return a value of Not-a-Number.		
Start = 30257+(2×MaxNumSens)×0 Length = 2×MaxNumSens *(for MaxNumSens=32 and NumTypes=3* it is 30257 – 30320)	0x0100	In pairs, cooked **moisture** sensor readings of the first, second, third, etc sensors as 32-bit IEEE floating point numbers (low word in lower register address). Updated dynamically as reading progresses.
Start = 30257+(2×MaxNumSens)×1 Length = 2×MaxNumSens *(for MaxNumSens=32 and NumTypes=3* it is 30321 - 30384)		In pairs, cooked **salinity** sensor readings of the first, second, third, etc sensors as 32-bit IEEE floating point numbers (low word in lower register address). Updated dynamically as reading progresses.
Start = 30257+(2×MaxNumSens)×2 Length = 2×MaxNumSens (for MaxNumSens=32 and NumTypes=3 it is 30385 - 30448)		Reserved

Holding Registers

Absolute Address (output Registers) *See Register map formula for description of actual addresses calculations.*	**Relative Address**	**Description**
40001	0x0000	Command Register; write to this to start a sensor reading 0 = No action 1 = Read sensors specified in mask (40002 through 40007) 2 = Read all (detected) moisture sensors 3 = Read all (detected) salinity sensors Writing other values, or writing while the value is non-zero will cause an exception response 'Illegal data value'. This register should not be updated while a reading is in progress (30002=1)

Absolute Address (output Registers) *See Register map formula for description of actual addresses calculations.*	**Relative Address**	**Description**
Selection mask for the sensors to be sampled Only sensors with their corresponding bit set will be scanned, other sensors will get a cooked value of 0.0. The bits in this mask are in the same order as the detected sensor mask.		
Start = 40002 + WordsPerMask×0 = 40002 **Length** = WordsPerMask (for MaxNumSens=32 it is 40002 – 40003)	0x0001	**Moisture** sensor selection mask register (use in conjunction with 40001 = 1) (Bit 0 is first sensor Bit 1 is second sensor, … Bit 16 is 17th sensor, etc)
Start = 40002 + WordsPerMask×1 **Length** = WordsPerMask (for MaxNumSens=32 it is 40004 – 40005)		**Salinity** sensor selection mask register (use in conjunction with 40001 = 1) (Bit 0 is first sensor Bit 1 is second sensor, … Bit 16 is 17th sensor, etc)
Start = 40002 + WordsPerMask×2 **Length** = WordsPerMask (for MaxNumSens=32 it is 40006 – 40007)		Reserved

Note: In writing to multiple Holding Registers, the higher address registers are in fact written to first; this permits a preset multiple registers command to both set the selection mask and initiate a reading (40001=1).

Register Retention

If the probe is powered down, contents of all Modbus registers are cleared. At power on, the probe will initialize detected sensors mask (registers 30009 through 30014), maximum number of sensors/types (register 30100) and sensor depths (registers 30101 through 30196) from its configuration data when power is restored.

Data Types

As per Modbus Protocol specification, all registers (16 bit) are represented as Big Endian format.

There is no specified representation for 32 bit words in either integer or floating point.

This implementation uses a mixed big/little endian format that appears to be the most commonly implemented method on Intel platforms.

The format of the mixed big/little endian is as follows:

Absolute Address 1 (Relative 0)		Absolute address 2 (Relative 1)	
32 Bit Data			
Low word		High word	
Bits 15 to 8	Bits 7 to 0	Bits 31 to 24	Bits 23 to 16
16 Bit Data			
Bits 15 to 8	Bits 7 to 0	Bits 15 to 8	Bits 7 to 0

Figure 8 Format of mixed big/little endian

The low word is in the lower register address.

Big endian implementations will need to swap Reg1 and Reg2 and concatenate them to obtain a value that is equivalent to the native form.

Little endian implementations should not need to modify anything as most Modbus drivers implemented on Little endian systems internally swap the bytes within the Modbus register before passing up to the application.

Data Reading

Note: Probe interface must be configured correctly to obtain valid readings from the Modbus port. See section Setting up the probe interface for more information.

Data reading using the **EnviroSCAN** and **EasyAG** Modbus probe interface can be achieved using the following steps.

Note: The following example assumes that the Register map (see address 30100) reflects number of sensors of 32 (**MaxNumSens**) and number of sensor types of 3 (**NumTypes**). A different value may result in different input and holding register addresses.

1. Ensure there is no measurement in progress (register 30002 <> 1).
2. To start a reading of all configured moisture sensors, set register 40001 to the value 2 (or 3 for salinity sensors), or
3. To start a reading of selected sensors, set register 40002 through 40007 with selection masks, and then write the value 1 to register 40001. In a selection mask, bit 0 of the register selects the first sensor of that type; bit 1 selects the second sensor, etc. The input registers are updated:
   30001 is the last command executed (i.e. it mirrors the last value written to 40001),
   30002 is the current status,
   30003 through 30008 is the selection mask used (taken from 40002 through 40007).
4. Reading progress can be monitored in register 30002; a value of 1 means the measurement is in progress; 2 means measurement completed, while 3 means sensor failure occurred during measurement. Failed sensors have a cooked value of Not-a-Number.
5. Register 30009 through 30014 is the selection mask of detected sensors.
6. Register 30015 through 30020 is a selection mask of registers processed during a reading; it is updated as the reading progresses.
7. Registers 30101 through 30196 contain the depths of each sensor. The value is an integer scaled up by 10 to give accuracy to one decimal point. There are no units assigned to this value, it is dimensionless.
8. Registers 30257 through 30448 contain cooked data values; Consecutive pairs of registers are an IEEE 32-bit floating point number (the lower register address contains the low word). These are all set to zero on a measurement request, and updated as the request progresses. Invalid conversions (e.g. configured sensor failed) are stored as IEEE "Not a Number."

Salinity Sensor considerations:

When the interface samples salinity sensors it also samples the corresponding moisture sensors. Consequently, after sampling salinity sensors, cooked data is available for both moisture (registers 30257 through 30320) and salinity (registers 30321 through 30384). Hence a separate command to read moisture data is not required.

Maximum Message Length

The maximum message length of a Modbus message is 255 bytes. Within the current implementation the maximum message lengths are:

- RTU mode
  =32 sensors × 2-bytes + address 1-byte + function 1-byte + CRC 2-bytes
  = 68 bytes
- ASCII mode
  = 32 sensors × 4-bytes + address 2-byte + function 2-byte + LRC 2-bytes + colon-prefix 1-byte + <CR><LF> 2-bytes
  = 137 bytes

Communication Parameters

The Modbus output port communication parameters are set using the Probe Configuration Utility, on the Modbus page. The available parameters are:

Baud rate	1200, 2400, 4800, 9600 (*default*), 19200 bits per second
Data bits	7, 8
Parity	None (*default*), Even, Odd
Stop bits	1, 2 (*default*)

Note: The **EnviroSCAN** and **EasyAG** RS232 probe interfaces use hardware-handshaking. The probe interface will raise RTS before sending a response message, and waits for CTS before transmitting the response. The interface acts as a half-duplex DTE device.

Note: If the probe interface is directly connected to a computer serial port, the following information should be noted.

1. Using an RS232 connection, both CTS and RTS lines must both be open circuit (both attempt to raise RTS)
2. Using an RS485 connection, an RS232 to RS485 converter should be configured as a DCE, 2-wire with no RTS.

Technical Specifications

EnviroSCAN RS232 Series II Technical Specifications

PCB Revision: REV 2.3

Identification Label: XPI-SC-232

RS232 Interface connector type: Brand: Phoenix Contact

MC 1,5/8-ST-3,5 (Plug)

MC 1,5/8-G-3,5 (Socket)

RS232 Interface pin configuration:

1. +V*in*
2. Reserved
3. Reserved
4. Ground
5. TX – RS232 Data
6. RX– RS232 Data
7. Request to send (RTS)
8. Clear to Send (CTS)

Note: Where RTS and CTS lines are not connected to the Modbus master device, it is recommended that these two pins be linked together.

Voltage Supply (RS232 +Vin**):** 4 - 15 Volts (12 Volts DC @ >200mA recommended)

Fuse specification: F201 Littelfuse 0154-500 (500mA fast blow)

RS232 Interface baud rate: 1200, 2400, 9600 (*default*), 19200 and 38400 bits per second (user configurable)

TTL Interface connector type: Brand: JST

B 6B-PH-K (Socket)

PHR- 6 (Plug), SPH-002T-P0.5S (Crimp connectors)

TTL Interface pin configuration:

1. +V*cc*
2. Transmit data (T*x*)
3. Receive data (R*x*)
4. Reserved
5. Reserved
6. Ground

Voltage Supply (TTL +Vcc): 5 Volts, supplied by the EnviroSCAN probe interface

TTL Interface baud rate: 1200, 2400, 9600 (default), 19200 and 38400 bits per second (user configurable)

Total current consumption: 400µA @ standby

105mA @ sampling (Moisture) @12V DC

130mA @ Sampling (TriSCAN) @12V DC

Time to sample 1 sensor: 45 milliseconds maximum (Moisture only)

90 milliseconds maximum (TriSCAN)

Maximum sensors supported: 16 Moisture Sensors

16 TriSCAN Sensors

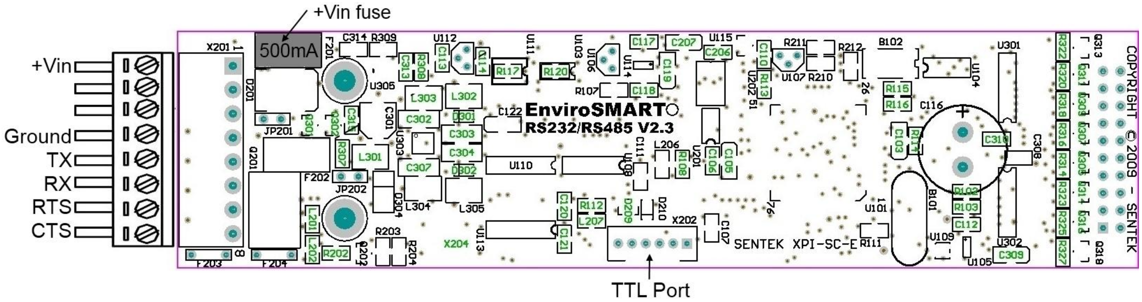


Figure 9 **EnviroSCAN** RS232 Modbus probe interface board layout

EnviroSCAN RS485 Series II Technical Specifications

PCB Revision: REV 2.3

Identification Label: XPI-SC-E-RS485

RS485 Interface connector type: Brand: Phoenix Contact
MC 1,5/8-ST-3,5 (Socket)
MC 1,5/8-G-3,5 (Plug)

RS485 Interface pin configuration:

1. +V*in*
2. Reserved
3. Reserved
4. Ground
5. A – RS485 Data
6. B – RS485 Data
7. Screen
8. *Reserved*

Voltage Supply (RS485 +Vin**):** 4 – 15 Volts (12 Volts DC @ >200mA recommended)

RS485 Interface baud rate: 1200, 2400, 9600 (default), 19200 and 38400 bits per second (user configurable)

TTL Interface connector type: Brand: JST
B 6B-PH-K (Socket)
PHR- 6 (Plug), SPH-002T-P0.5S (Crimp connectors)

TTL Interface pin configuration:

1. +V*cc*
2. Transmit data (T*x*)
3. Receive data (R*x*)
4. *Reserved*
5. *Reserved*
6. Ground

Voltage Supply (TTL +Vcc): 5 Volts, supplied by the EnviroSCAN probe interface

TTL Interface baud rate: 1200, 2400, 9600 (default), 19200 and 38400 bits per second (user configurable)

Total current consumption: 400µA @ standby @ 12V DC
105mA @ sampling (Moisture) @ 12V DC
130mA @ Sampling (TriSCAN) @ 12V DC

Time to sample 1 sensor: 45 milliseconds maximum (Moisture only)
90 milliseconds maximum (TriSCAN)

Maximum sensors supported: 16 Moisture Sensors

16 TriSCAN Sensors

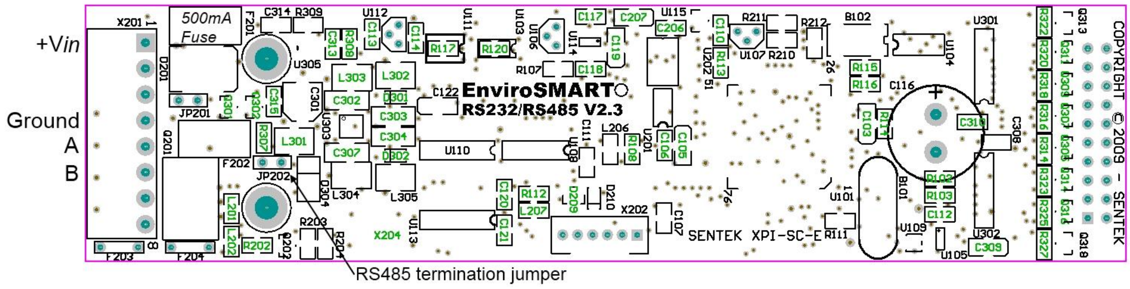


Figure 10 **EnviroSCAN** RS485 Modbus probe interface board layout

EasyAG RS232 Series II Technical Specifications

PCB Revision: PCB 0681 V2.4

Identification Label: XPI-SC-I-RS232

RS232 Interface connector type: Brand: Phoenix Contact

MPT0.5/8-2.54

RS232 Interface pin configuration:

1. +V*in*
2. Reserved
3. Reserved
4. Ground
5. TX – RS232 Data
6. RX– RS232 Data
7. Request to send (RTS)
8. Clear to Send (CTS)

Note: Where RTS and CTS lines are not connected to the Modbus master device, it is recommended that these two pins be linked together.

Voltage Supply (RS232 +Vin**):** 4 – 15 Volts (12 Volts DC @ >200mA recommended)

Fuse specifications: F201 Littlefuse 0154-500 (500mA fast blow)

RS232 Interface baud rate: 1200, 2400, 9600 (default), 19200 and 38400 bits per second (user configurable)

TTL Interface connector type: Brand: JST

S6B-PH-SM3-TB

PHR- 6 (Plug), SPH-002T-P0.5S (Crimp connectors)

TTL Interface pin configuration:

1. +V*cc*
2. Transmit data (T*x*)
3. Receive data (R*x*)
4. Reserved
5. Reserved
6. Ground

Voltage Supply (TTL +Vcc): 5 Volts, supplied by the EasyAG probe interface

TTL Interface baud rate: 1200, 2400, 9600 (default), 19200 and 38400 bits per second (user configurable)

Total current consumption: 400µA @ standby

	102mA @ sampling (Moisture)
	126mA @ Sampling (TriSCAN)
Time to sample 1 sensor:	45 milliseconds maximum (Moisture only)
	90 milliseconds maximum (TriSCAN)
Maximum sensors supported:	5 Moisture Sensors
	5 TriSCAN Sensors

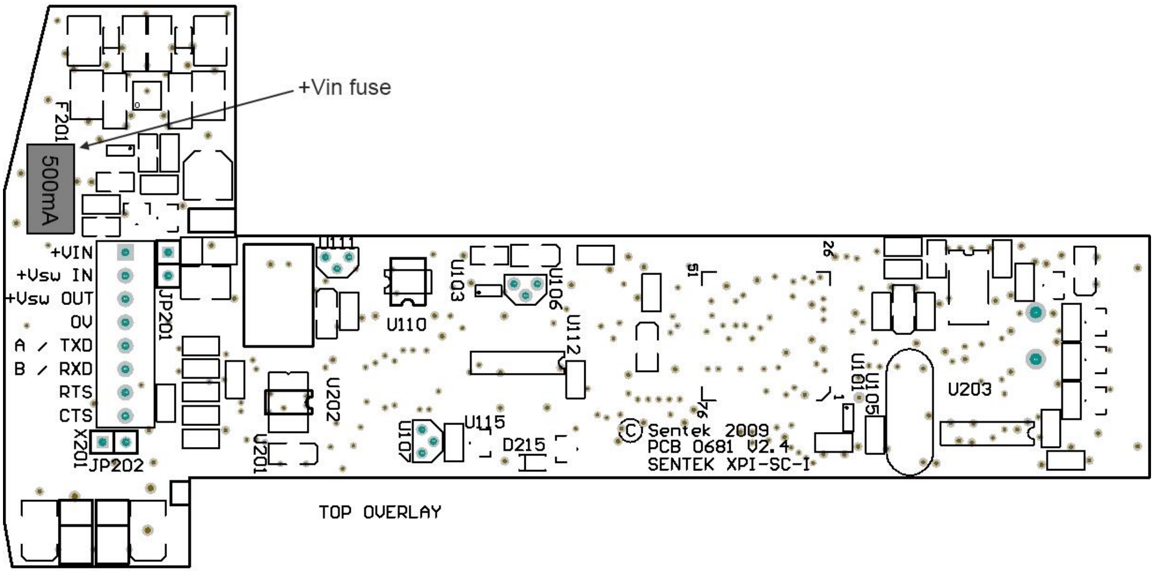


Figure 11 **EasyAG** RS232 Modbus probe interface board layout

EasyAG RS485 Series II Technical Specifications

PCB Revision: PCB 0681 REV 2.4

Identification Label: XPI-SC-I-RS485

RS485 Interface connector type: Brand: Phoenix Contact
MPT0.5/8-2.54

RS485 Interface pin configuration:

1. +V*in*
2. Reserved
3. Reserved
4. Ground
5. A – RS485 Data
6. B – RS485 Data
7. Screen
8. *Reserved*

Voltage Supply (RS485 +Vin**):** 4 – 15 Volts (12 Volts DC @ >200mA recommended)

RS485 Interface baud rate: 1200, 2400, 9600 (default), 19200 and 38400 bits per second (user configurable)

TTL Interface connector type: Brand: JST
B 6B-PH-K (Socket)
PHR- 6 (Plug), SPH-002T-P0.5S (Crimp connectors)

TTL Interface pin configuration:

1. +V*cc*
2. Transmit data (T*x*)

3. Receive data (R*x*)
4. Reserved
5. Reserved
6. Ground

Voltage Supply (TTL +Vcc**):** 5 Volts, supplied by the **EasyAG** probe interface

TTL Interface baud rate: 1200, 2400, 9600 (*default*), 19200 and 38400 bits per second (user configurable)

Total current consumption: 400µA @ standby
102mA @ sampling (Moisture)
130mA @ Sampling (TriSCAN)

Time to sample 1 sensor: 50 milliseconds maximum (Moisture only)
50 milliseconds maximum (TriSCAN)

Maximum sensors supported: 5 Moisture Sensors
5 TriSCAN Sensors

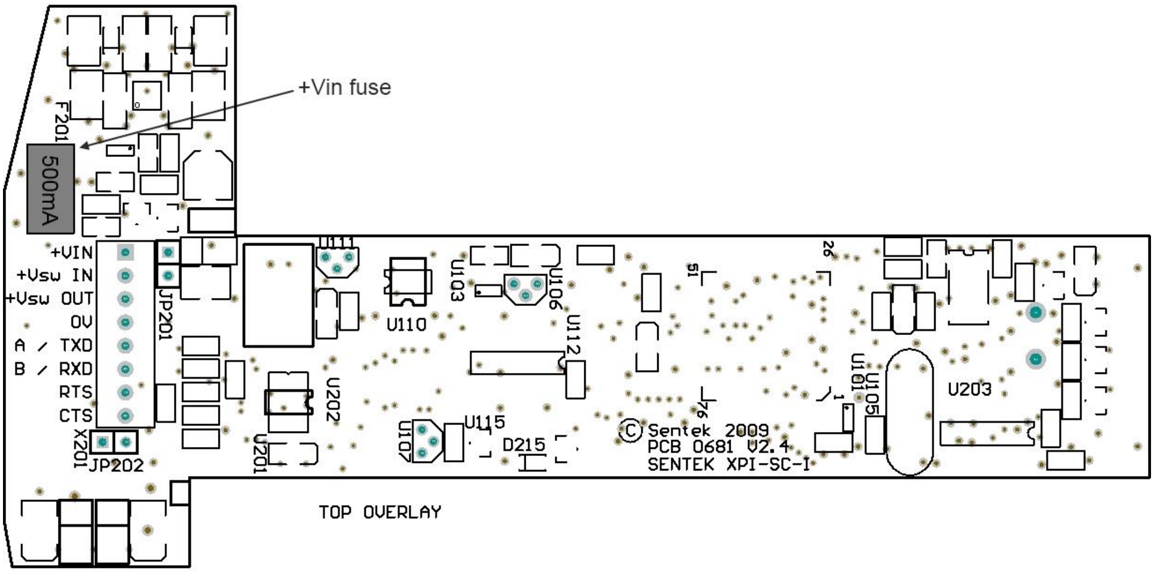


Figure 12 **EasyAG** RS485 Modbus probe interface board layout

EnviroSCAN and EasyAG RS232 and RS485 Circuit Information

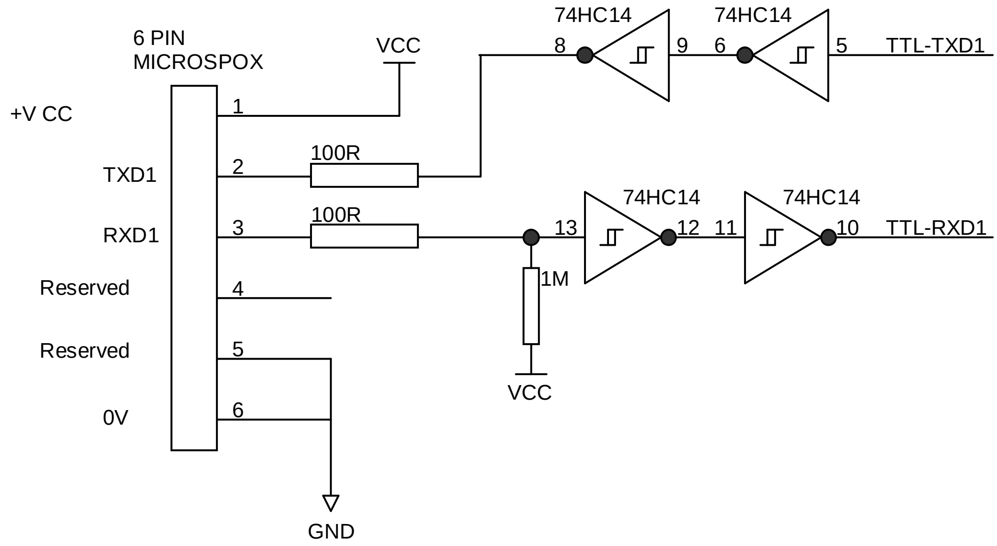


Figure 13: TTL interface circuit diagram

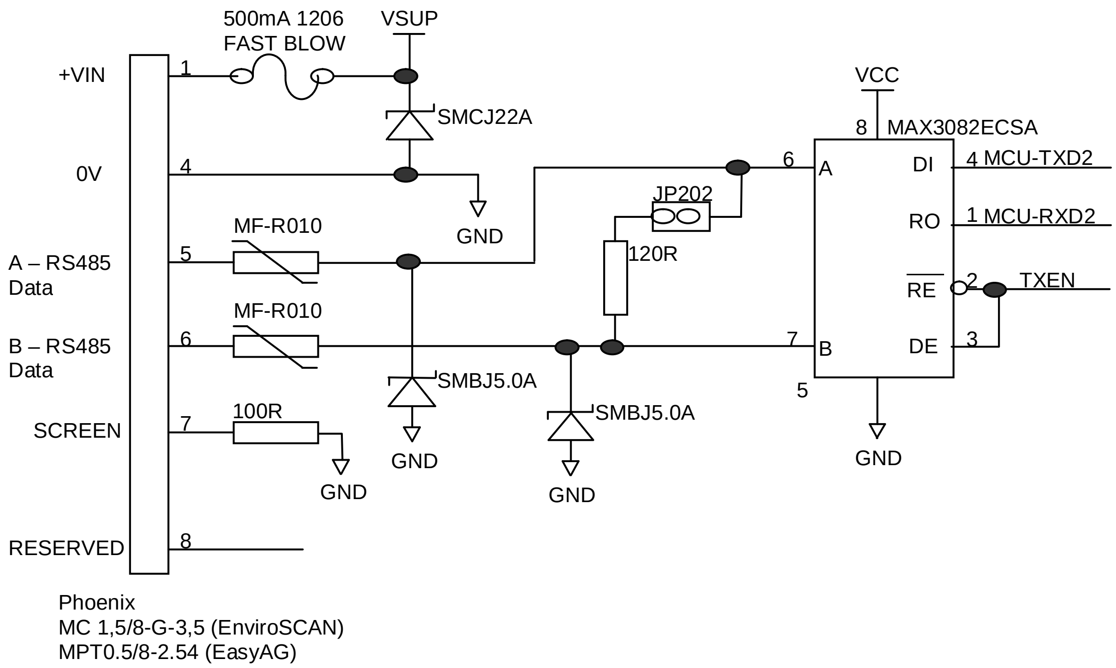


Figure 14: RS485 interface circuit diagram

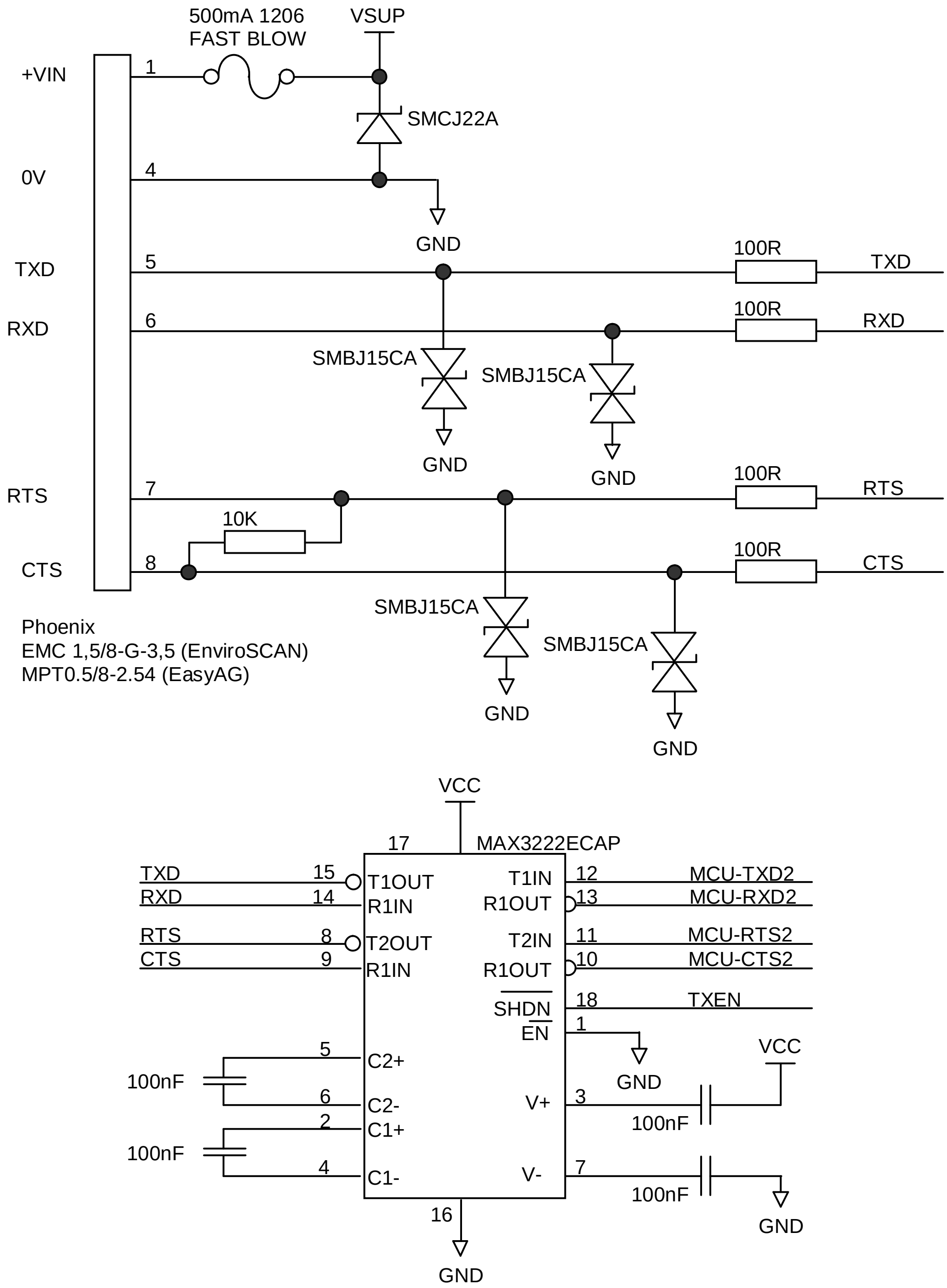


Figure 15: RS232 interface circuit diagram

Appendix A – Sentek Modbus Master Device Guidelines

Objective

The objective of this appendix is to give guidelines to developers/vendors of the Modbus master device for **Sentek** Modbus slave devices, to allow them to develop an implementation of the master device code which would remain compatible with future versions of **Sentek** Modbus slave devices including new types of such devices and **Sentek** Windows Software.

Scope

This appendix does not cover full code or algorithms, which could be deemed as a provision of routines or subroutines. It presents only a high-level description to give an idea and understanding of how the Modbus registers exposed by the Sentek devices should be used properly for maintaining compatibility with future **Sentek** slave devices.

AutoSDB is mentioned as to how the master should determine which data from the slave should be put into the DB (please refer to "AutoSDB Programmers' Manual" for full information on tables and fields available through AutoSDB).

Background

Sentek devices by their nature are highly configurable based on the customer needs. This high level of variation in the configurations, as well as continuous development of new types and sizes/capacities of the devices requires that the master understands the configuration of the device that is attached to it. On top of that, assuming that the logged and retrieved data is then to be transferred into the "**EnviroSCAN** for Windows®" software, the master needs to know where in the database (data columns) the retrieved data belongs.

Falling short of this consistency could result in the Modbus master becoming incompatible with other than tested variations of the product not to mention future products, which could have extended functionality.

Guidelines

Variations/changes that the Modbus master device should be able to handle without any need to change its software are:

- Varying number of sensors on the probe
- Varying sensor addresses for the same sensor depth level
- Varying depth settings for the same sensor address
- Increased maximum number of sensors supported by probe – hence Modbus register addresses
- New sensor types introduced

The master does not have to understand the future types, however the introduction and support of those types in the slave devices need not break the compatibility with the master software.

The Master therefore should follow rules in how to query the configuration, how to trigger the measurement, how to retrieve the data and then on the computer side (under Windows) how the data is to be properly put into the **EnviroSCAN** database.

Rule #1 ("Little Endian" vs. "Big Endian")

Every multi-word (multi-register) data is stored in a "little endian" style, that is, lower word (2 bytes) is stored at the lower address and, higher words are stored at higher addresses. This also applies to IEEE float numbers.

Note that the register (word) itself is communicated between the **Sentek** Modbus slave devices and the Modbus master devices in a "big endian" style (as per standard Modbus specification), that is, higher byte comes first, followed by lower byte.

Rule #2 (Master initialisation)

At the time of master initialization (connection to Sentek slave), retrieve the value of the register 30100 to obtain the maximum number of sensors for each type and the number of sensor types allocated in the Modbus address space.

The current specification states the maximum number of sensors per type is 32 and the number of sensor types allocated is 3 (note that versions of earlier **Sentek** Modbus devices may return 0, which should be treated as 3).

These values may increase or decrease in the future devices.

These two values determine the beginning of the Modbus register addresses.

Rule #3 (Bitmask register sizes)

Bit masks should be considered as a string of bits, not necessarily in 16 bit packets.

Calculate the number of words needed to represent a bit mask for the maximum number of sensors (see "Sentek's Modbus Implementation Specification Ver.2.0).

For example, for a maximum number of sensors of 32 for each type, the bit mask, representing with its bits every sensor, would have to be of 4 bytes (4x8=32). Since every register in Modbus protocol is 16 bits wide, we'd need 2 registers for each sensor type to represent the bit masks (see formula in the specifications document). Please remember "rule #1".

Rule #4 (Address calculations)

Calculate the beginning of the Modbus address space for each sensor type for various types of data instead of hard coding.

The formulas are described in the main section of this document, including constants to be used.

There are various data tables available through the Modbus address space. Each of these tables cover all allocated types of sensors, hence the size of the table is dependent on the number of "sensor types" allocated and the maximum number of sensors allocated per type.

$$A_{TypeStart} = (T-1) \times A_{MoistureStart}$$

where

$A_{TypeStart}$	is the first register address of a specific “info” table for the specific sensor type
T	is the sensor type code [moisture=1, future sensor types may be added]
$A_{MoistureStart}$	is the first register address of a specific “info” table for the moisture type (see following)

Note: The “maximum number of sensors” per type or the “number of types” allocated in the Modbus address space only defines register location. It does not imply hardware support.

Rule #5 (Depth information)

Retrieve the “Depth Information” from the “Depth” table for every sensor detected.

Note: This information has been made available only in firmware versions 1.1.0 and above.

Note: Sentek recommends upgrading the firmware in the Sentek’s Modbus slave devices to the latest version as applicable.

The depth information should be used as a ”key” for matching the retrieved data with the database tables (i.e. **Sentek EnviroSCAN** software) to ensure that the data integrity is maintained automatically even if sensor addressing is changed.

For example, the 3rd party software retrieving the moisture data from the probe would thus know that for instance the first sensor has a depth set up to 10cm, hence when putting the data into the **EnviroSCAN** database, the software is able to find the right table and column (using AutoSDB functions), which hold the data for the depth level of 10cm, to have the data written into the right place.

Note: The 3rd party software linking to **Sentek EnviroSCAN** software (using AutoSDB interface) can query the appropriate database table to determine what column holds the data for which depth level. The depth level information is one of the fields exposed through the AutoSDB interface.

Appendix B – Soil Moisture Management

What soil volume does the probe interface measure at a single sensor?

At a single depth level, a sensor on the probe records volumetric water content from a soil volume outside the access tube, which has a sphere of influence of:

- 10cm vertical height
- 5-10cm radial distance from the outer wall of the access tube

What are the water units?

If a calibrated sensor reads one(1) millimeter, there is one(1) millimeter of volumetric water content in a soil volume 10cm deep.

Q. What does 1mm volumetric water content / 10cm soil depth mean?

A. You require one(1) liter of water to cover one(1) square meter (m²) to a soil depth of one(1) millimeter.

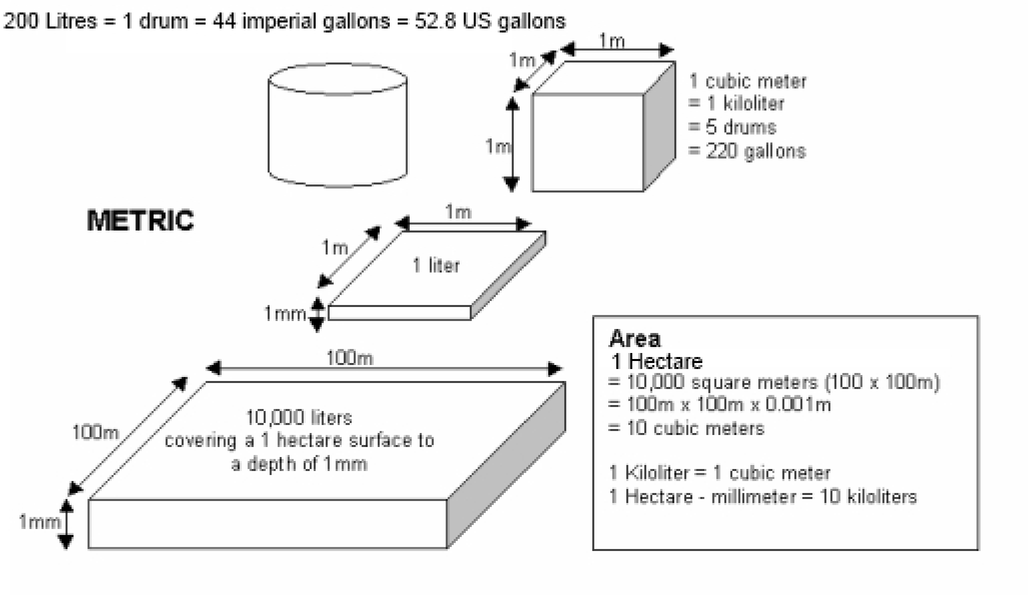


Figure 13 Measurements using metric units

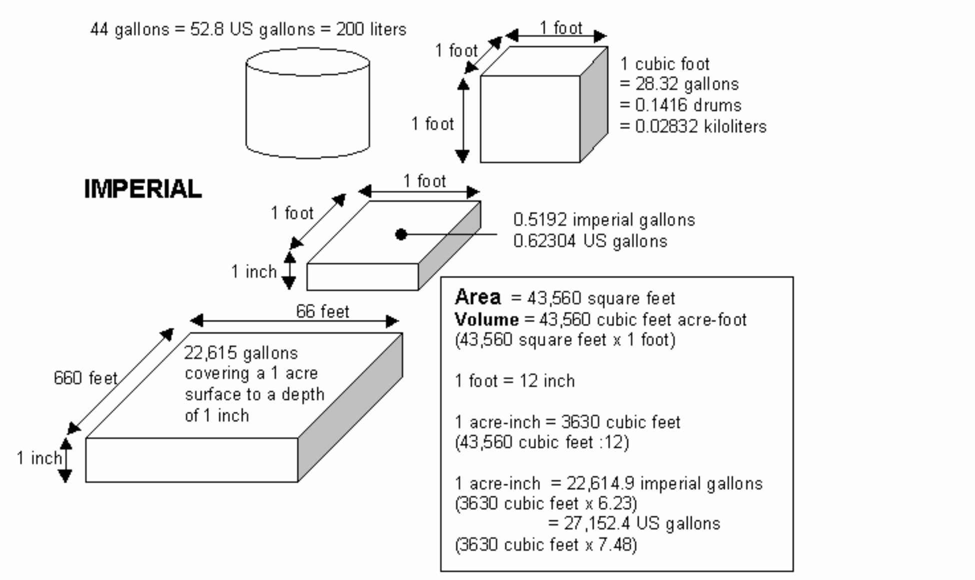


Figure 14 Measurements using imperial units

What part of the soil profile do multiple sensors on the probe measure?

Probes in almost all cases have more than one sensor to monitor the depth of irrigation and the depth of the root zone. The first sensor is located at a soil depth of 10cm (if the datum plate of the top cap sits on ground level) measuring effectively the soil profile slice of 5-15cm depth. The next sensor is located at 20 cm measuring effectively 15 –25cm soil depths. With further sensors at 10cm intervals on the probe rod, the measurement depth would be respectively (25-35cm, 35-45cm and so on).

If you raise the datum plate of the top-cap 5cm above the ground surface, placing the center of the first sensor effectively at 5cm soil depth, the sphere of influence of the sensor will measure a soil slice from 0-10cm. For the other sensors at 10cm depth intervals on the probe rod, the measurement depth would be respectively (20-30cm, 30-40cm and so on).

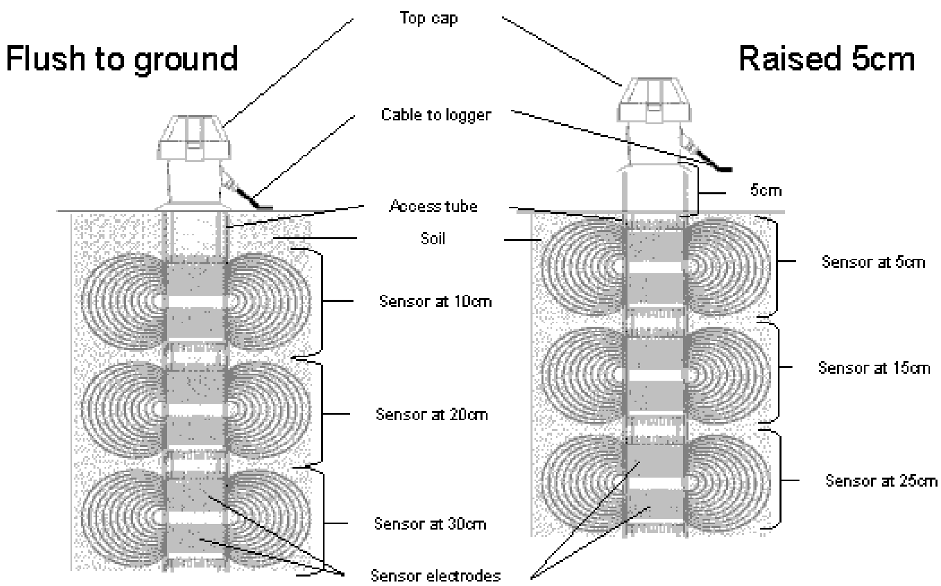


Figure 15 Measurements of multiple sensors on the EnviroSCAN probe

Appendix C – Salinity Management

Please refer to the "TriSCAN™ Agronomic User Manual" for detailed information on measuring and interpreting salinity. The following information has been extracted from that manual.

Sensor output and measurement units

The TriSCAN sensor provides two outputs.

The first output is a signal of dimensionless frequency (raw count), that is converted via a normalization equation and then a default or user-defined calibration equation into volumetric soil water content. The measurement unit is thus volumetric water content (Vol %) or millimetres of water per 100 mm of soil depth

The second output is also a dimensionless frequency (raw count) that, in conjunction with the first output signal, is proportional to changes in soil water content and salinity. A proprietary data model processes the changes of both output signals simultaneously to reflect the changes that are due to soil salinity changes only. The output of the data model is a nominal Volumetric Ion Content (VIC). Measurement units of VIC can be quantitatively related (calibrated) to the soil EC through site specific physical soil sampling and analysis. Similar to the soil water data, the sensor output of VIC can be presented as dynamic trend changes over a chosen time scale.

Measurement Range

The effective measurement range of TriSCAN is between 0 and 17 dS/m.

Resolution and Accuracy

The resolution and accuracy of the sensor can be considered in terms of the two different outputs: Volumetric Water Content: and Salinity:

To date, statements on the resolution and accuracy of the salinity output can only be made with regards to the particular type of sand that has been used to develop the TriSCAN data processing models. These specifications do not necessary reflect the resolution and accuracy of the sensor on any other soil type.

The accuracy and precision of the sensor in the tested sand were as follows:

- In saturated soil conditions at low EC 55µS/cm:
  Repeatable change of VIC's related to a resolution of 1 µS/cm (Accuracy ±1.8%)
- In saturated soil conditions at medium EC up to 5600 µS/cm:
  Repeatable change of VIC's related to a resolution of 25 µS/cm (Accuracy ± 0.4%)

Temperature Effects

The temperature effects on TriSCAN data output are currently unknown. It is however, known that the electrical conductivity of solutions and of soils increases by approximately 2.25% per degree Celsius. To simplify the interpretation of salinity data, it is customary in the industry either to take the measurements at a standard reference temperature or to determine the temperature at which the measurement is made, and then, by means of a correction table or model, to convert the measurement to the standard-reference temperature.

The TriSCAN model currently does not include such a temperature correction model.

Why TriSCAN does not measure individual plant nutrients (ions)?

Soil salts are chemical compounds that are dissolved in soil solutions. Salts are comprised of anions such as carbonates, chlorides, sulfates and nitrates, and cations such as potassium, magnesium, calcium and sodium. In ambient conditions, these compounds are present in proportions that create a balanced ionic or charged solution.

Under natural soil conditions TriSCAN™ is unable to discern individual salt compounds.

As equivalent concentrations of some salts show a varying degree of conductivity, the distinction between different single salt solutions can only be made by measuring the conductivity of an equivalent salt concentration of a single salt source in solution comparatively. TriSCAN™ would detect this difference in conductivity, but it could not indicate the identity of the single salt compounds.

In what soil types does TriSCAN work?

The effective measurement range of TriSCAN is between 0 and 17 dSm-1 in sand, loamy sand and sandy loam textures (Australian Soil and Land Survey Field Handbook). Use of TriSCAN at salinity levels and soil textures outside this range is currently unsupported by Sentek.

Data logging requirements for TriSCAN

Consideration should be made to the power and timing requirements when using TriSCAN sensors. In particular:

- Taking a salinity reading requires 100 milliseconds per sensor
- Salinity sensors use slightly higher current draw than moisture sensors.

See the Technical Specifications section for more detail.

