

## Table of Contents

Table of Contents .....	1
Specifications .....	1
SDI-12 Command Nomenclature .....	1
SDI-12 Commands .....	2
<i>Setup Commands</i> .....	2
<i>Measurement Commands</i> .....	2
<i>Request Measurement</i> .....	2
<i>Request Measurement with CRC</i> .....	3
<i>Concurrent Measurement</i> .....	4
<i>Concurrent Measurement with CRC</i> .....	4
Modbus Register Definitions .....	5

*Original Release: September 23, 2010*

*Edit: March 12, 2018 for re-branding*

## Specifications

Power supply voltage: 9.0 – 16.0VDC

Power supply current – Active: 3.5mA Typ. @ 12.0V

Power supply current – Sleep: 60µA Typ. @ 12.0V

Measurement Latency: Approx. 1.4s

Default SDI-12 Address: 0

## SDI-12 Command Nomenclature

a= Sensor address

{crc} = SDI-12 compatible 3-character CRC

<cr> = ASCII carriage return character

<lf> = ASCII line feed character

highlighted values indicate variable data

### SDI-12 Commands

#### Setup Commands

Name	Command	Response
Sensor Identification	a!	a13 INWUSA TempHi2.4sssssssss<cr> <lf> <i>Note: 2.4 will change to reflect current firmware revision, ssssssssss = device serial #</i>
Acknowledge Active	a!	a<cr> <lf>
Address Query	?!	a<cr> <lf>
Start Verification	aV!	a0000<cr> <lf>
Change Address	aAb!	b<cr> <lf> Change address from a to b

#### Measurement Commands

**Note:** The default units setting for temperature is Celsius. To change this, use the Direct Read Units option under the Configure | Advanced menu in the Aqua4Plus Control Software. When using the M!, MC!, C!, or CC! command, all mV channels will report in either pH, ppm, or Eh, depending on channel type (pH = pH, ISE = ppm, ORP = Eh). When using the M1!, MC1!, C1! or CC1! command, all mV channels will report in mV.

**Note:** For firmware version 2.0 – 2.3, the M1 and M1! commands are swapped. Likewise for the MC1/MC1!, C1/C1!, and CC1/CC1! commands.

#### Request Measurement

Name	Command	Response
Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH</li> <li>• ISE</li> <li>• ORP</li> </ul>	aM!	atttn<cr> <lf> <i>"n" values available after "ttt" seconds</i> <i>Sample: a0024&lt;cr&gt; &lt;lf&gt;</i>
Read results	aD0!	aValue1Value2Value3Value4<cr> <lf> <i>Value1 = temperature</i> <i>Value2 = pH in pH units</i> <i>Value3 = ISE in ppm</i> <i>Value4 = ORP in Eh</i> <i>Sample: a+21.345+7.181+.053+.459.431 &lt;cr&gt; &lt;lf&gt;</i>

# TempHion Technical Bulletin

## SDI-12 & Modbus® Interface Specs



Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH in mV</li> <li>• ISE in mV</li> <li>• ORP in mV</li> </ul>	aM1!	atttn<cr><lf> <i>"n" values available after "ttt" seconds</i> Sample: a0024<cr><lf>
Read results	aD0!	aValue1Value2Value3Value4<cr><lf> <i>Value1 = temperature</i> <i>Value2 = pH in mV</i> <i>Value3 = ISE in mV</i> <i>Value4 = ORP in mV</i> Sample: +21.34-134.458+100.48+.84.404<cr><lf >

### Request Measurement with CRC

Name	Command	Response
Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH</li> <li>• ISE</li> <li>• ORP</li> </ul>	aMC!	atttn<cr><lf> <i>"n" values available after "ttt" seconds</i> Sample: a0024<cr><lf>
Read results	aD0!	aValue1Value2Value3Value4{crc}<cr><lf> <i>Value1 = temperature</i> <i>Value2 = pH in pH units</i> <i>Value3 = ISE in ppm</i> <i>Value4 = ORP in Eh</i> Sample: a+21.345+7.181+.053+.459.431{crc}<cr><lf>
Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH in mV</li> <li>• ISE in mV</li> <li>• ORP in mV</li> </ul>	aMC1!	atttn<cr><lf> <i>"n" values available after "ttt" seconds</i> Sample: a0024<cr><lf>
Read results	aD0!	aValue1Value2Value3Value4{crc}<cr><lf> <i>Value1 = temperature</i> <i>Value2 = pH in mV</i> <i>Value3 = ISE in mV</i> <i>Value4 = ORP in mV</i> Sample: a+21.34-134.458+100.48+.84.404{crc}<cr><lf >

### Concurrent Measurement

Name	Command	Response
Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH</li> <li>• ISE</li> <li>• ORP</li> </ul>	aC!	attnn<cr><lf> <i>"nn" values available after "ttn" seconds</i> Sample: a00204<cr><lf>
Read results	aD0!	aValue1Value2Value3Value4<cr><lf> Value1 = temperature Value2 = pH in pH units Value3 = ISE in ppm Value4 = ORP in Eh Sample: a+21.345+7.181+.053+.459.431<cr><lf>
Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH in mV</li> <li>• ISE in mV</li> <li>• ORP in mV</li> </ul>	aC1!	attnn<cr><lf> <i>"nn" values available after "ttn" seconds</i> Sample: a00204<cr><lf>
Read results	aD0!	aValue1Value2Value3Value4<cr><lf> Value1 = temperature Value2 = pH in mV Value3 = ISE in mV Value4 = ORP in mV Sample: a+21.34-134.458+100.48+.84.404<cr><lf>

### Concurrent Measurement with CRC

Name	Command	Response
Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH</li> <li>• ISE</li> <li>• ORP</li> </ul>	aCC!	attnn<cr><lf> <i>"nn" values available after "ttn" seconds</i> Sample: a00204<cr><lf>

Read results	aD0!	aValue1Value2Value3Value4{crc}<cr><lf> <i>Value1 = temperature</i> <i>Value2 = pH in pH units</i> <i>Value3 = ISE in ppm</i> <i>Value4 = ORP in Eh</i> Sample: a+21.345+7.181+.053+.459.431{crc}<cr><lf>
Request measurement: <ul style="list-style-type: none"> <li>• temperature</li> <li>• pH in mV</li> <li>• ISE in mV</li> <li>• ORP in mV</li> </ul>	aCC1!	atttnn<cr><lf> <i>"nn" values available after "ttt" seconds</i> Sample: a00204<cr><lf>
Read results	aD0!	aValue1Value2Value3Value4{crc}<cr><lf> <i>Value1 = temperature</i> <i>Value2 = pH in mV</i> <i>Value3 = ISE in mV</i> <i>Value4 = ORP in mV</i> Sample: a+21.34-134.458+100.48+.84.404{crc}<cr><lf >

## Modbus Register Definitions

### Modbus® Register Definitions

All Modbus data is accessed as holding registers (4xxxxx), using the Read Holding Registers (command type 3) and Preset Multiple Holding Registers (command type 0x10) Modbus commands.

### Parameter data

Modbus parameter data is returned as 32-bit ieee floating point values. Each value is 32 bits in length, and therefore occupies two Modbus registers. As a result, the parameter registers must be read as pairs.

For firmware 2.5 and higher:

Modbus Register	Data Address	Description
462593	0xF480 (62592)	Temperature
462595	0xF482 (62594)	pH in pH units
462597	0xF484 (62596)	ISE in ppm
462599	0xF486(62598)	ORP in Eh
462601	0xF488 (62600)	pH in mV
462603	0xF48A (62602)	ISE in mV
462605	0xF48C (62604)	ORP in mV

For firmware prior to 2.5:

Modbus Register	Data Address	Description
462593	0xF480 (62592)	Temperature
462595	0xF482 (62594)	pH in mV
462597	0xF484 (62596)	ISE in mV
462599	0xF486(62598)	ORP in mV
462601	0xF488 (62600)	pH in pH units
462603	0xF48A (62602)	ISE in ppm
462605	0xF48C (62604)	ORP in Eh

**Note:** The default units setting for temperature is Celsius. To change this, use the Direct Read Units option under the Configure | Advanced menu in the Aqua4Plus Control Software. When reading the registers 462595, 462597, or 462599, values will report in mV. When reading the registers 462601, 462603, or 462605, values will report in either pH, ppm, or Eh, depending on channel type. (pH = pH, ISE = ppm, ORP = Eh)

### Sensor configuration/control

Modbus registers are provided for the following configuration and control functions.

Modbus Register	Data Address	Description
462977=a	0xF600 (62976)	Set sensor Modbus address = a (Write Only)
464545=b	0xFC20 (64544)	Set Modbus baud rate according to b (Write Only) b=0:38400 b=1:19200 b=2:9600

---

Modbus is a registered trademark of Schneider Electric.