ADDITIONS TO THE SDI-12 SPECIFICATION

(April 18, 2016, to be integrated into the formal SDI-12 Specification document)

1.0 IDENTIFY MEASUREMENT COMMANDS

The identify measurement commands provide a means to get the response to a command without actually initiating a measurement. These commands are formed by placing the capital letter I into the measurement commands immediately after the address. The response is identical to having issued the command without the capital letter I following the address. This format is attn.<CR><LF>, atttnn<CR><LF>, or atttnnn<CR><LF> depending on the measurement command. Note: The Continuous Measurement commands (aR0! … aR9! and aRC0! … aRC9!) provide their data instantly and therefore do not have an Identify Measurement Command.

Command	Response
aIM!	atttn<CR><LF>
aIMC!	atttn<CR><LF>
aIM1!	atttn<CR><LF>
.	atttn<CR><LF>
.	atttn<CR><LF>
.	atttn<CR><LF>
aIM9!	atttn<CR><LF>
aIMC1!	atttn<CR><LF>
.	atttn<CR><LF>
.	atttn<CR><LF>
.	atttn<CR><LF>
aIMC9!	atttn<CR><LF>
aIV!	atttn<CR><LF>
aIC!	atttnn<CR><LF>
aICC!	atttnn<CR><LF>
aIC1!	atttnn<CR><LF>
.	atttnn<CR><LF>
.	atttnn<CR><LF>
.	atttnn<CR><LF>
aIC9!	atttnn<CR><LF>
aICC1!	atttnn<CR><LF>
.	atttnn<CR><LF>
.	atttnn<CR><LF>
.	atttnn<CR><LF>
aICC9!	atttnn<CR><LF>

Command	Response
aIHA!	atttnnn<CR><LF>
aIHB!	atttnnn<CR><LF>

Table 1. The identify measurement commands

1.1 Examples of the Identify Measurement Commands

a. One measurement will be immediately available after the M command:

8IM!80001<CR><LF>

b. Nine measurements will be available 10 seconds after the M command:

8IM!80109<CR><LF>

c. Ninety-nine measurements will be available 10 seconds after the C5 command:

8IC5!801099<CR><LF>

2.0 THE IDENTIFY MEASUREMENT PARAMETER COMMANDS

The identify measurement parameter commands provide details about the parameters returned by a particular command. The form of the command is an expansion of the Identify Measurement Commands. An underscore character ("_") plus a three digit decimal number is placed immediately before the exclamation point ("!"). The decimal number is the data value of interest.

The response is a comma separated value (CSV) string with several fields that provide information about the data value of interest. Two fields are required. Additional fields can be added by the sensor manufacturer. Fields are expected to contain printable ASCII characters other than the comma character (“,”) or the semicolon character (“;”) since they are used to delimit fields.

2.1 Field One

The first field contains a concise identification of the parameter, which is the data value of interest. The recommendation is to use a Standard Hydrometeorological Exchange Format (SHEF) code.

SHEF codes are published by the National Oceanic and Atmospheric Administration (NOAA), National Weather Service in the “Standard Hydrometeorological Exchange Format (SHEF) Code Manual.” This document is available on the Internet at:

http://www.nws.noaa.gov/oh/hrl/shef/indexshef.htm

The SHEF codes are listed in “Appendix G, Physical Element Definitions,” in the NOAA document.

If an appropriate SHEF code does not exist for the parameter, or if the sensor manufacturer chooses not to use a SHEF code for the parameter, then field one may contains a concise identification of the parameter as determined by the sensor manufacturer. The recommendation, however, is to a SHEF codes when an appropriate SHEF code does exist.

2.2 Field Two

Field two contains the units for the parameter. If the parameter is unit-less, the field must still be present. A single space character is recommended for an empty field to make it easier to read.

2.3 Optional Fields

The sensor manufacturer may provide additional information relevant to the parameter by adding additional fields. This may be a more descriptive name than found in field one. It may contain calibration data or dates. If the parameter represents a probe that has a unique serial number, there may be a field that contains that serial number. The only limit on the number of additional fields is that the maximum length of the response, up through the terminating semicolon, is 75 characters.

The last field is terminated by a semicolon “;” and then followed by either the <CR><LF> sequence or a three character CRC followed by the <CR><LF> sequence if the measurement command was one that returns a CRC.

If the parameter number, nnn, is invalid for the measurement, then the response shall be: a<CR><LF> or a<CRC><CR><LF> if a CRC was expected. Note: To poll the Continuous Measurement commands (aR0! … aR9! and aRC0! … aRC9!) to identify their measurement parameters, increment sequentially through the parameters till the response indicates the parameter number, nnn, is invalid for the measurement.

Command	Response
aIM_001!	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
aIM_009!	a,field1,field2;<CR><LF>
aIMC_001!	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
aIMC_009!	a,field1,field2;<CRC><CR><LF>
aIM1_001! … aIM1_009!	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
aIM9_001! … aIM9_009!	a,field1,field2;<CR><LF>
aIMC1_001! … aIMC1_009!	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
aIMC9_001! … aIMC9_009!	a,field1,field2;<CRC><CR><LF>
aIV_001! … aIV_009!	a,field1,field2;<CR><LF>
aIC_001! … aIC_099!	a,field1,field2;<CR><LF>
aICC_001! … aICC_009!	a,field1,field2;<CRC><CR><LF>
aIC1_001! ... aIC1_009!	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
aIC9_001! … aIC9_099!	a,field1,field2;<CR><LF>
aICC1_001! … aICC1_099!	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
aICC9_001! … aICC9_099!	a,field1,field2;<CRC><CR><LF>

aIR0_001! … aIR0_099!	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
.	a,field1,field2;<CR><LF>
aIR9!_001! … aIR9_099!	a,field1,field2;<CR><LF>
aIRC0_001! … aIRC0_099!	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
.	a,field1,field2;<CRC><CR><LF>
aIRC9_001! … aIRC9_099!	a,field1,field2;<CRC><CR><LF>
aIHA_001! … aIHA_999!	a,field1,field2;<CRC><CR><LF>
aIHB_001! … aIHB_999!	a,field1,field2;<CRC><CR><LF>

Table 2. The identify measurement parameter commands

3.0 START HIGH VOLUME ASCII MEASUREMENT

The High Volume ASCII measurement commands expand the concurrent measurement command to allow up to 999 parameters to be returned from a sensor.

Command Name	Command	Response
High Volume ASCII	aHA!	atttnnn<CR><LF>

Table 3. High Volume ASCII Measurement

The commands to Get High Volume Data After the aHA! command are: aD0! … aD999!

If after obtaining the data values from aD9! there are still more data values to obtain, continue to send data requests with aD10!...aD99! as needed. If after obtaining the data values from aD99! there are still more data values to obtain, continue to the send data requests with aD100!...aD999! as needed. Leading zeros are not placed after the "D".

The responses to the send data commands follow the same rules as with the Concurrent Measurement send data commands. The maximum number of characters that can be returned in the <values> part of the response is 75 and a three character <CRC> is appended to the data before the <CR><LF>. The CRC must be present.

4.0 START HIGH VOLUME BINARY MEASUREMENT

The High Volume Binary Measurement allows for collection of large volumes of data from a sensor more efficiently than the ASCII transfer methods.

Command Name	Command	Response
High Volume Binary	aHB!	atttnnn<CR><LF>

Table 4. High volume binary measurement

The commands to Get High Volume Binary Data After the aHB! command are: aDB0! … aDB999!

If after obtaining the data values from aDB9! there are still more data values to obtain, continue the send data requests with aDB10!...aDB99! as needed. If after obtaining the data values from aDB99! there are still more data values to obtain, continue the send data requests with aDB100!...aDB999! as needed.

Responses to the aDB0! … aDB999! Commands are an exception to section 4.1 Table 2 because the byte frame format is 8 data bits, no parity bit. The SDI-12 address is transmitted as the ASCII character, but no parity bit. The remaining fields are encoded as binary numbers. The least significant byte is transferred first for multi-byte binary numbers.

SDI-12 Address	Packet Size	Data Type	Binary Data Payload	CRC
ASCII	16 bit unsigned integer, indicates the size, in bytes, of the binary data payload	8 bit unsigned integer, indicates the data type in the binary data payload	must be <= 1,000 bytes	16 bit CRC value, using the same algorithm as the other measurement commands that request a CRC, but encoded in binary (not converted to 3 byte ASCII)

Table 5. Data Packet

If the value of n in aDBn! is invalid due to being higher than necessary to return data values, then the data pack must have a value of zero in the Packet Size field.

All data values in a particular response must be of the same type, but data types can differ between aDBn! Commands.

Data Type	Range	Size
0	Indicates an invalid request	No data returned
1	-128 to 127	Signed 8-bit integer
2	0 to 255	Unsigned 8-bit integer
3	-32,768 to 32,767	Signed 16-bit integer
4	0 to 65,535	Unsigned 16-bit integer
5	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer
6	0 to 4,294,967,295	Unsigned 32-bit integer
7	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64-bit integer
8	0 to 18,446,744,073,709,551,615	Unsigned 64-bit integer
9	$\pm 1.18\times10^{-38}$ to $\pm 3.4\times10^{38}$	IEEE 32-bit floating point single precision/binary32
10	$\pm 2.23\times10^{-308}$ to $\pm 1.80\times10^{308}$	IEEE-64 bit floating point double precision/binary64

Table 6. Data types