SDI-12

Serial Digital Interface at 1200 Baud

By

Michael Jablonski

Chairman, SDI-12 Support Group

SDI-12
A Serial-Digital Interface Standard
for
Microprocessor-Based Sensors

Version 1.4

January 30, 2021

Prepared By

SDI-12 Support Group
(Technical Committee)
165 East 500 South
River Heights, Utah
435-752-4200
435-752-1691 (FAX)
www.sdi-12.org

The SDI-12 Support Group

I am the chairman of the SDI-12 Support Group, a non-profit corporation with two purposes:

1. Educational: Publish the SDI-12 Specification and help people understand SDI-12. The current version of SDI-12 is 1.4.

2. Scientific: upgrade the Specification as needed to facilitate the collection of environmental data using the SDI-12.

Membership in the SDI-12 Support Group is open to all with an interest in SDI-12.

www.sdi-12.org

History of SDI-12

- SDI-12 was created in 1988 by a group of private firms and the U.S. Geological Survey's Hydrologic Instrumentation Facility (USGS/HIF).
- The key requirement was a low power standard interface between the data recorder and intelligent, micro-processor based sensors at remotes sites.
- The SDI-12 interface was to be very simple to implement in small instruments with limited computing power, such as 8-bit control microprocessors.
- The goal was for all sensors using the SDI-12 standard to work with all data recorders using the standard.
- In 1991 the SDI-12 Support Group was formed as a non-profit corporation to maintain the SDI-12 standard.

Advantages of SDI-12

All property designed SDI-12 sensors will work with all properly designed SDI-12 data recorders.

Sensors can be interchanged without reprogramming the data recorder.

Low power consumption. Power is supplied to sensors through the interface.

Backwards compatibility. A version 1.0 SDI-12 sensor will work with a Version 1.4 SDI-12 Data Recorder.

SDI-12 is Used World-Wide

Australia, Austria, Brazil, Canada, China, Finland, France, Germany, Hungary, Italy, Israel, New Zealand, Norway, Poland, Slovenia, Slovak Republic, South Korea, Spain, Switzerland, United Kingdom, USA

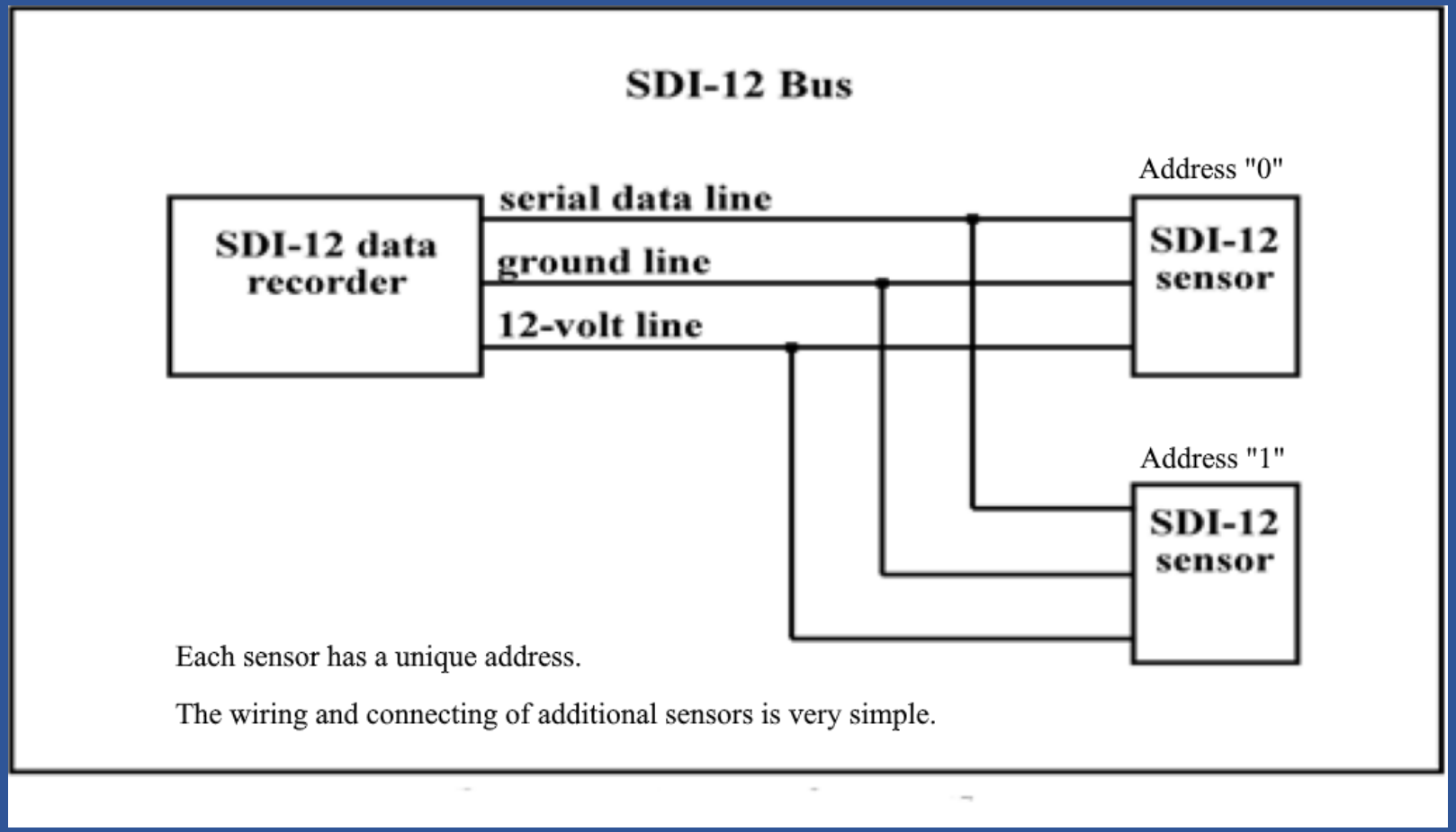
Countries that have SDI-12 product suppliers and users.

SDI-12 is Multi-Drop and Multi-Parameter

- Multi-Drop: more than once sensor can be connected to a data recorder.

- Multi-Parameter: a sensor may take more than one measurement. For example, some SDI-12 water quality sensors measure Temperature, Conductivity, Depth, pH, Dissolved Oxygen (DO), Turbidity, and Oxidation-Reduction Potential (ORP).

The SDI-12 Bus



Basics of SDI-12

- The data recorder sends a command to a specific sensor, as each sensor has a unique address.
- The sensor returns a response to the command.
- All commands and responses are comprised of ASCII letters, digits, and a few other characters.
- You can read the commands and responses because they are meaningful letters and digits.

Printable/Readable ASCII Characters Are Used

Dec	Char	Dec	Char	Dec	Char
32	SPACE	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

There are three exceptions when non-printable/readable ASCII characters are allowed:

1) all responses from an SDI-12 sensor end with a carriage return (13 decimal) and a line feed (10 decimal) character, shown as <CR><LF> in the SDI-12 Specification;

2) sometimes the second and third character of a CRC code may not be printable ASCII characters;

3) the contents of data packets returned by the High-Volume Binary command, because they are binary numbers (more about this later).

ASCII = American Standard Code for Information Interchange

printable/readable
ASCII characters

SDI-12 is a Protocol

- SDI-12, besides having an electrical specification, defines a protocol specifying how commands and response are transmitted and received and the timing of the commands and responses.
- Software running in the data recorders and sensors manages the protocol.
- The byte frame format is 1 start bit, 7 data bits,1 parity bit (even), and 1 stop bit (10 bits total, 8.33 milliseconds per character)

The SDI-12 Protocol

A typical recorder/sensor measurement sequence proceeds as follows:

Step 1. The data recorder wakes all sensors on the SDI-12 bus with a break. A break is continuous spacing, by the data recorder, on the data line for at least 12 milliseconds..

Step 2. The recorder transmits a command to a specific, addressed sensor, instructing it to make a measurement.

Step 3. The addressed sensor responds within 15.0 milliseconds returning the maximum time until the measurement data will be ready and the number of data values it will return.

Step 4. If the measurement is immediately available, the recorder transmits a command to the sensor instructing it to return the measurement(s). If the measurement is not ready, the data recorder waits for the sensor to send a request to the recorder, which indicates that the data are ready. The recorder then transmits a command to get the data.

Step 5. The sensor responds, returning one or more measurements.

Name	Command	Response
Break	Continuous spacing for at least 12 milliseconds	None
Acknowledge Active	a!	a<CR><LF>
Send Identification	aI!	allcccccccmmmmmmvvvxxx…xx<CR><LF>
Change Address	aAb!	b<CR><LF> (support for this command is required only if the sensor supports software changeable addresses)
Address Query	?!	a<CR><LF>
Start Measurement*	aM!	atttn<CR><LF>
Start Measurement and Request CRC*	aMC!	atttn<CR><LF>
Send Data	aD0!	a<values><CR><LF> or a<values><CRC><CR><LF>
	.	a<values><CR><LF> or a<values><CRC><CR><LF>
	.	a<values><CR><LF> or a<values><CRC><CR><LF>
	.	a<values><CR><LF> or a<values><CRC><CR><LF>
	aD9!	a<values><CR><LF> or a<values><CRC><CR><LF>
Additional Measurements*	aM1!	atttn<CR><LF>
	.	atttn<CR><LF>
	.	atttn<CR><LF>
	.	atttn<CR><LF>
	aM9!	atttn<CR><LF>
Additional Measurements and Request CRC*	aMC1! … aMC9!	atttn<CR><LF>
Start Verification*	aV!	atttn<CR><LF>
Start Concurrent Measurement	aC!	atttnn<CR><LF>
Start Concurrent Measurement and Request CRC	aCC!	atttnn<CR><LF>
Additional Concurrent Measurements	aC1!	atttnn<CR><LF>
	.	atttnn<CR><LF>
	.	atttnn<CR><LF>
	.	atttnn<CR><LF>
	aC9!	atttnn<CR><LF>
Additional Concurrent Measurements and Request CRC	aCC1! … aCC9!	atttnn<CR><LF>
Continuous Measurements	aR0! … aR9!	a<values><CR><LF> (formatted like the D commands)
Continuous Measurements and Request CRC	aRC0! … aRC9!	a<values><CRC><CR><LF> (formatted like the D commands)

*This command may result in a service request. See section 4.4.6.

Table 5. The SDI-12 basic command/response set

See Tables 12 and 13 for the high-volume commands (pages 24-25), Table 19 for the identify measurement commands (page 29), and Table 20 for the identify measurement parameter commands (page 32).

The Most Basic Commands

1. The Acknowledge Active Command a!
2. The Send Identification Command aI!
3. The Change Address Command aAb!
4. The Measurement Command aM!
5. The Get Data Command(s) aD0! (aD1! … aD9!)

a is the sensor's address, a single ASCII character, usually 0 … 9

Acknowledge Active
The Most Basic SDI-12 command

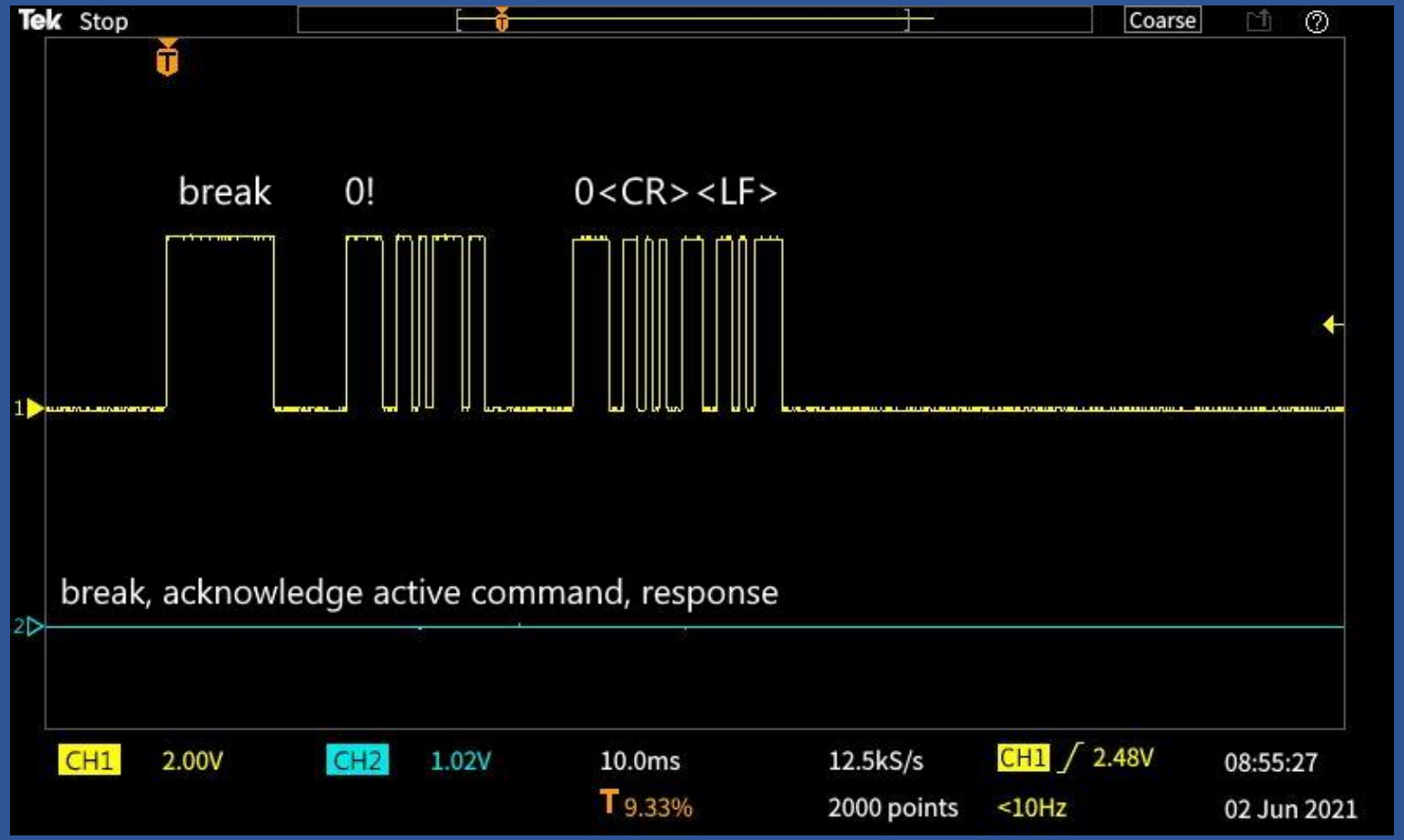
Command		Response
a!		a<CR><LF>
a - the sensor address		a - the sensor address
! - terminates the command		<CR><LF> - terminates the response

Think of this command as heartbeat command. It asks if the sensor is alive and well.

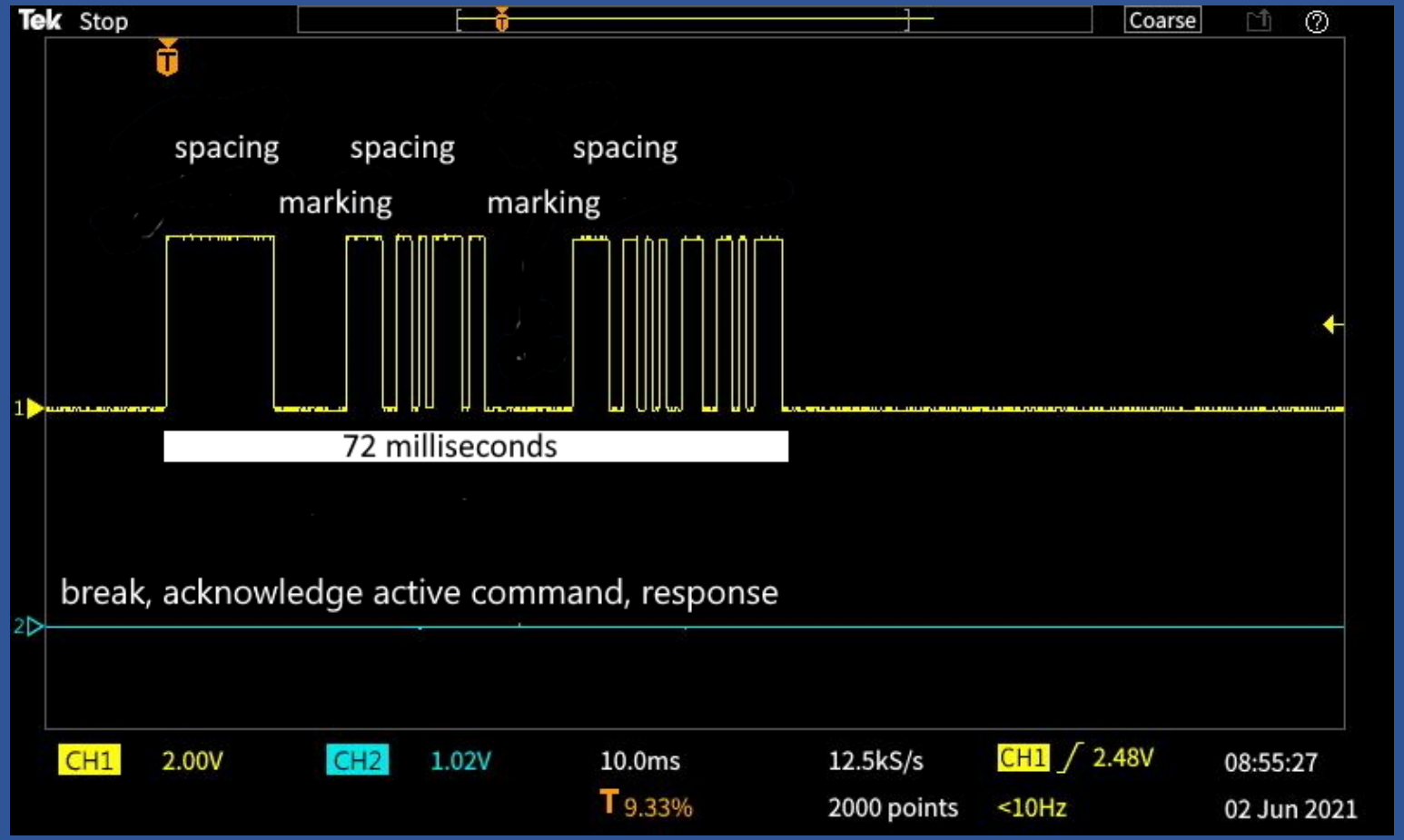
.

Example: 0!0<CR><LF>

The Acknowledge Active Command View of the SDI-12 Data Line



Marking and Spacing on the SDI-12 Data Line



The Send Identification Command

Command		Response
aI!		allccccccccmmmmmmvvvxxx . . . xxx<CR><LF>
a - the sensor address		a - the sensor address
I – the send identification command		ll - the SDI-12 version number, indicating SDI-12 version compatibility; for example, version 1.4 is encoded as 14
! - terminates the command		cccccccc - an 8 character vendor identification, usually a company name or its abbreviation
		mmmmmm - 6 characters specifying the sensor model number
		vvv - 3 characters specifying the sensor version
		xxx . . . xx - an optional field, up to 13 characters, used for a serial number or other specific sensor information that is not relevant for operation of the data recorder
		<CR><LF> - terminates the response

Example: 0I!014NR_Sys__Verify3.0_sensor_simul<CR><LF>

The Change Address Command

Command		Response
aAb!		b<CR><LF>
a - the sensor address		b - the address of the sensor (will equal the new address or the original address if the sensor is unable to change the address)
A - the change address command b - the address to change to ! - terminates the command		<CR><LF> - terminates the response

Example: 0A1!1<CR><LF>

The Measurement Command (aM!)

Command		Response
aM!		atttn<CR><LF>
a - the sensor address		a - the sensor address
M - the start measurement		ttt - the specified time, in seconds, until the sensor will have the measurement(s) ready
! - terminates the command		n - the number of measurement values the sensor will make and return in one or more subsequent D commands; n is a single digit integer with a valid range of 1 to 9

Example: 0M!00001<CR><LF>
In zero seconds one data value will be ready.

Getting The Measurement(s), The Send Data Commands (0D0! … 0D9!)

Command		Response
aD0! (aD1! . . . aD9!)		a<values><CR><LF>
		or
		a<values><CRC><CR><LF>
a - the sensor address		a - the sensor address
D0 - the send data command, D1 . . . D9 additional send data commands		values
! - terminates the command		<CR><LF> - terminates the response
		<values> - pd.d
		p - the polarity sign (+ or -)
		d - numeric digits before the decimal place
		. - the decimal point (optional)
		d - numeric digits after the decimal point the maximum number of digits for a data value is 7, even without a decimal point the minimum number of digits for a data value (excluding the decimal point) is 1 the maximum number of characters in a data value is 9 (the (polarity sign + 7 digits + the decimal point)) <CRC> - 3 character CRC code, appended if data was requested with the aMC!, aMC1! ... aMC9!, aCC!, or aCC1! ... aCC9! commands (see section 4.4.12)

Example: 0D0!+0.0<CR><LF>

Transparent Mode

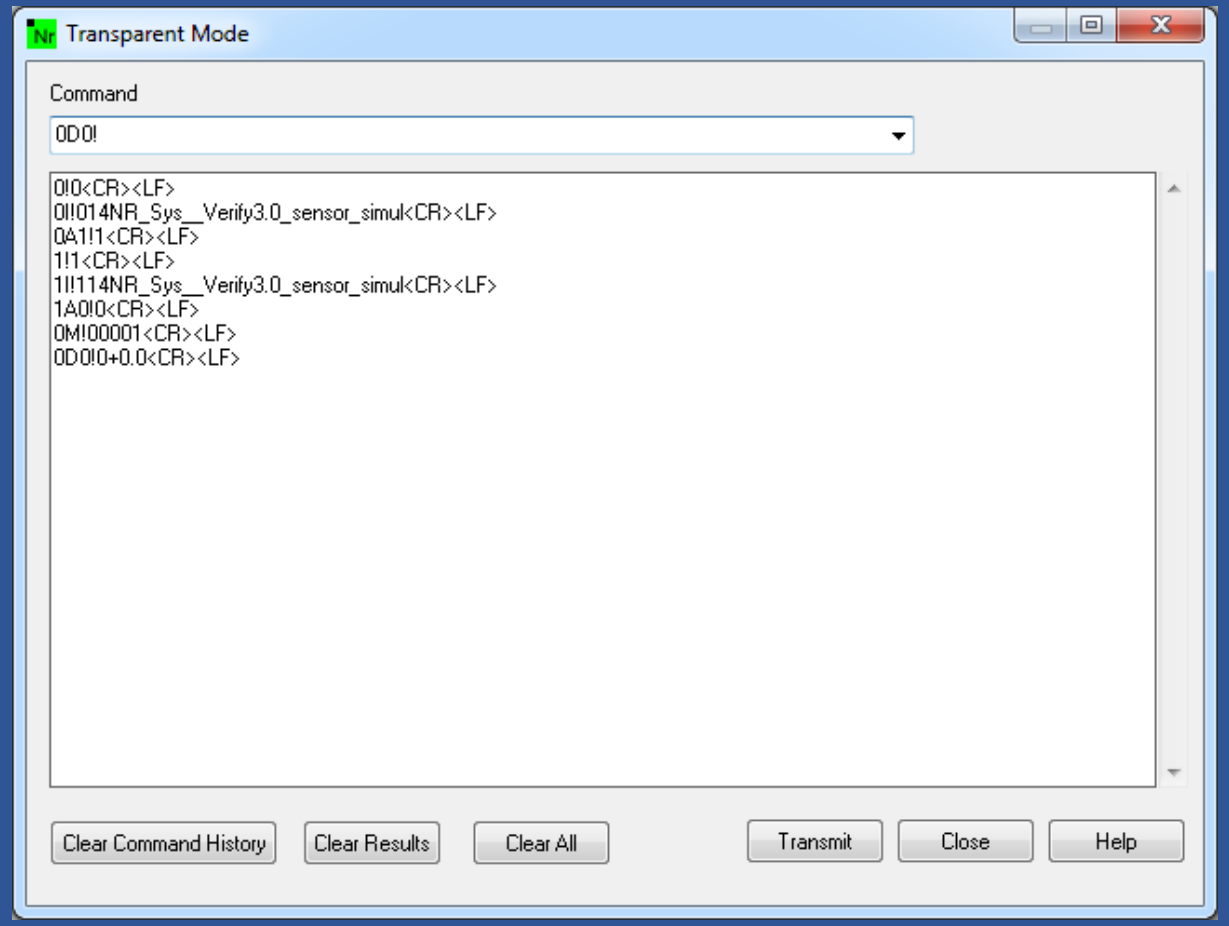
SDI-12 data recorders must have a mode in which commands can be sent to sensors. This is called the transparent mode. The transparent mode has the following characteristics.

The data recorder buffers a command string received from a computer, terminal, or modem, until the command string is terminated.

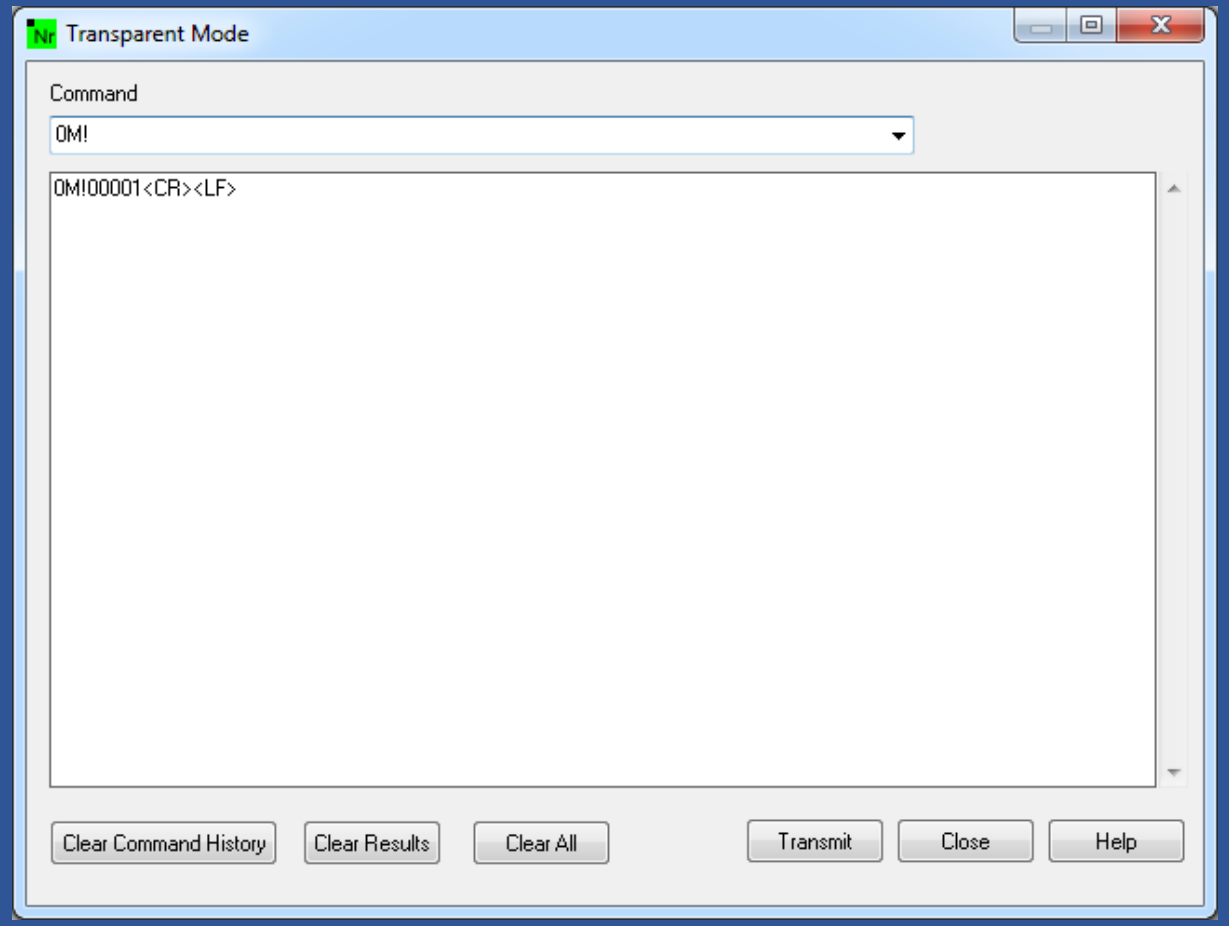
The data recorder wakes the sensor with a break, then it sends the buffered command to the sensor, using the SDI-12 protocol.

The data recorder receives the response from the sensor and transmits the response to the computer, the terminal, or the modem.

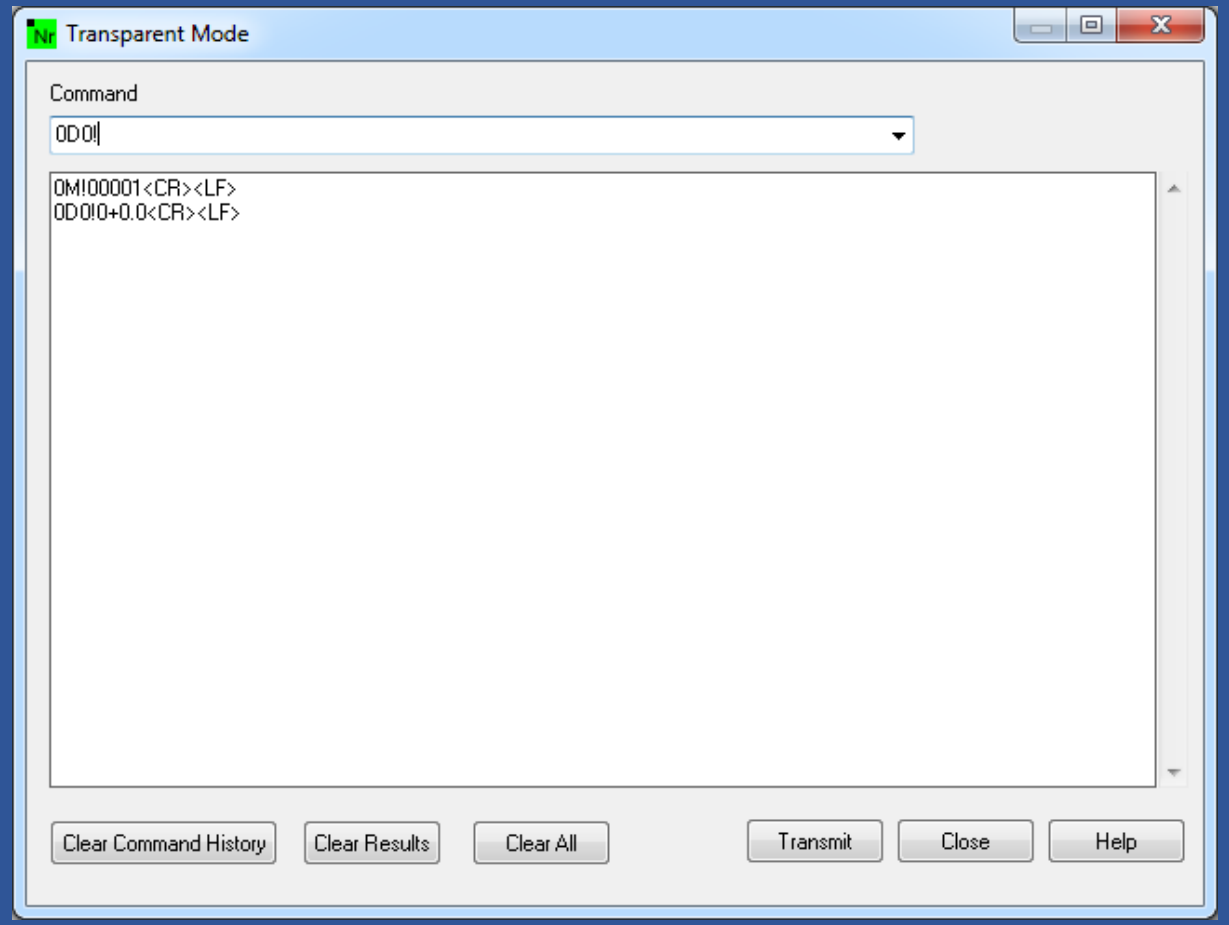
Examples of the Basic Commands



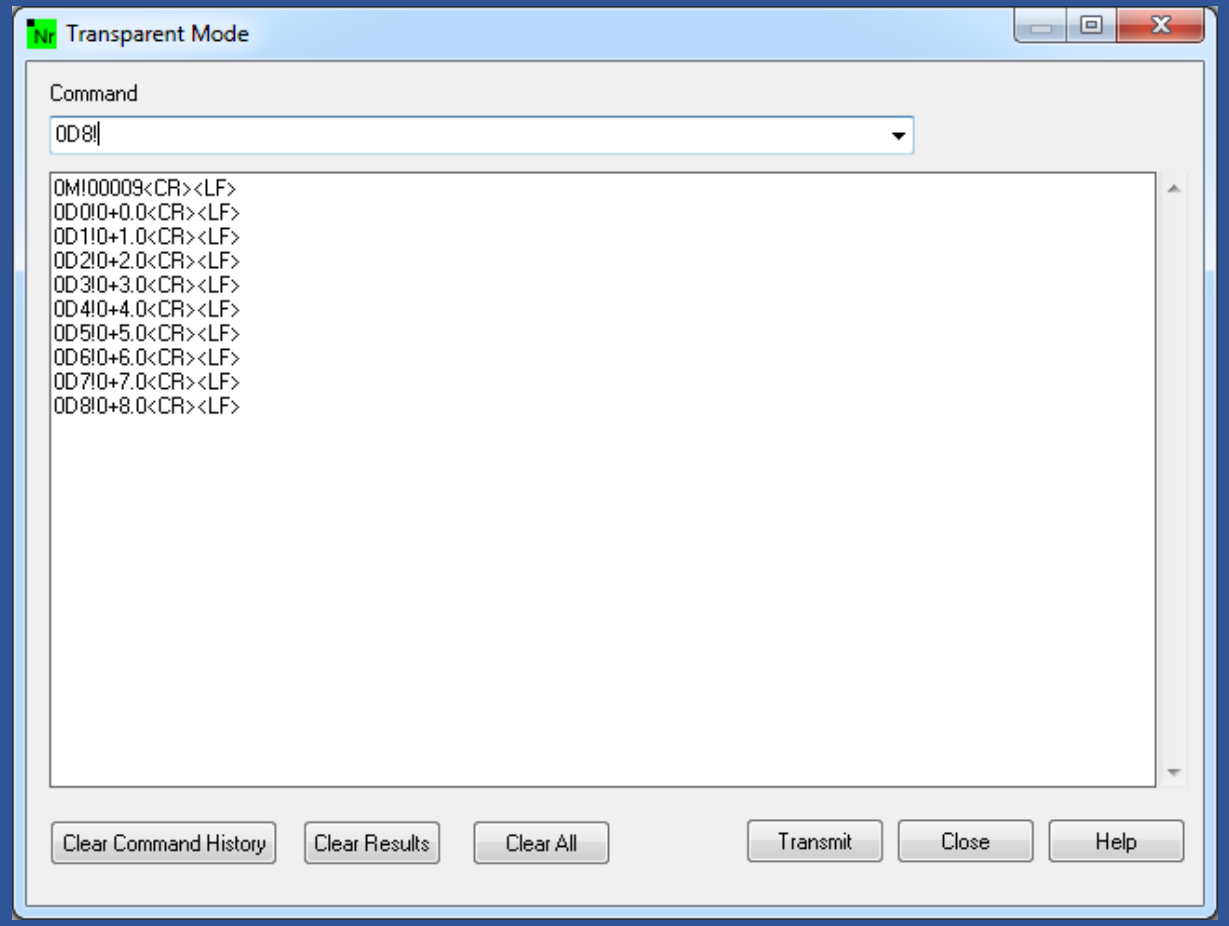
Transparent Mode Example of the Measurement Command (aM!)



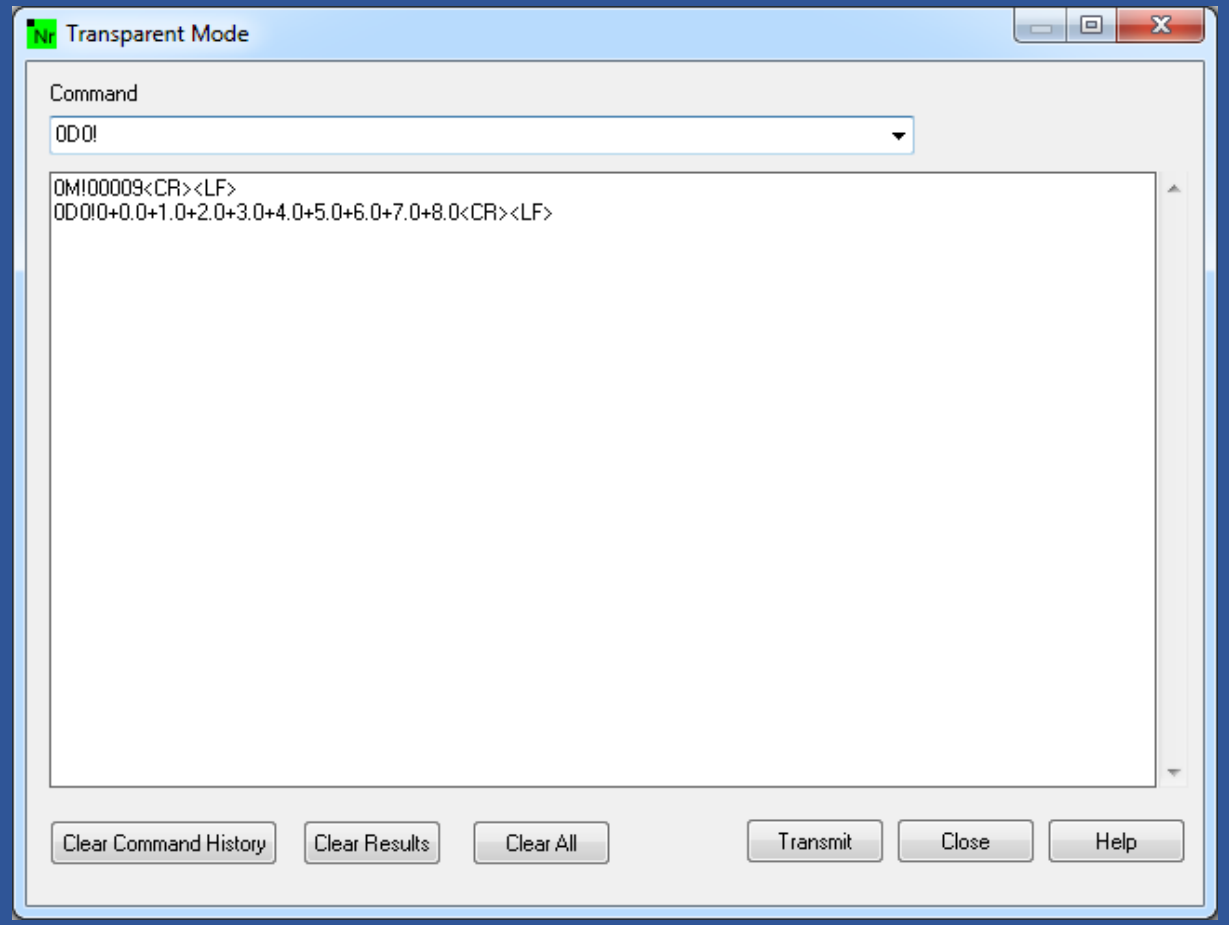
Getting a Measurement With The aD0! Command



Getting 9 Measurements with 9 D Commands



Getting all 9 Measurements with a Single D Command



An SDI-12 Oddity

The first M command is aM!

The first D command is aD0!

The M commands range from aM!, aM1!, aM2! … aM9!

The D commands range from aD0!, aD1!, aD2! … aD9!

There is no aM0! command.

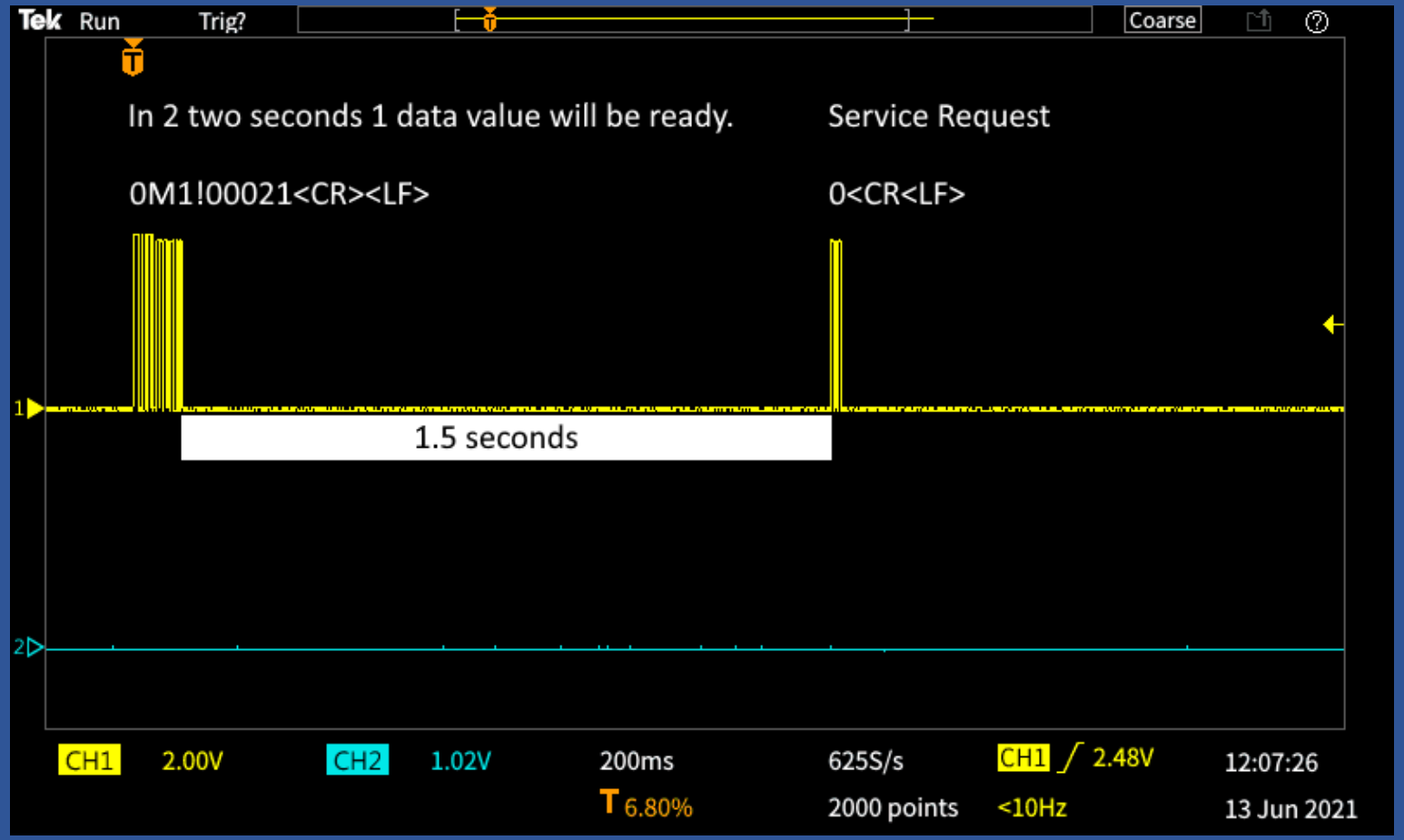
There is no aD! command.

Service Request

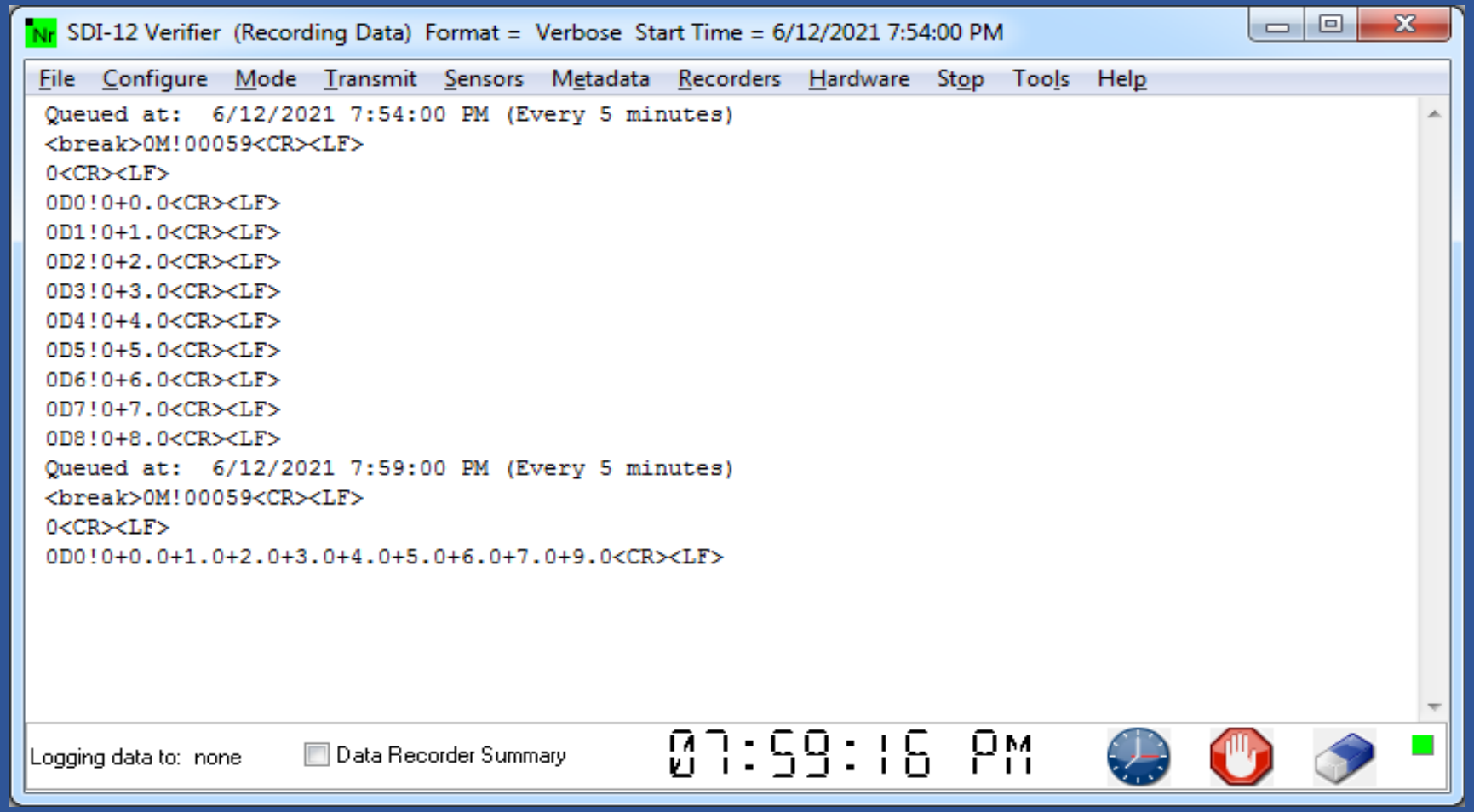
A service request is a response from a sensor. It is not a command. It is sent, after an M command, to tell the data recorder that the sensor has finished its measurement(s) and the data are ready. A service request is issued by the sensor after an M, MC, or V command, when it has finished its measurement.

The format of a service request is : a<CR><LF>

View Of A Service Request



Service Request (a<CR><LF>)



Additional M Commands

If you understand aM! command, then you already understand aM1! … aM9! Commands.

aM1!
aM2!
aM3!
aM4!
aM5!
aM6!
aM7!
aM8!
aM9!

Additional M commands provide a means to request different types of measurements from a sensor or to instruct a sensor to do a calibration or a control function. For example, a sensor could measure pressure and temperature: aM! tells it to measure pressure and aM1! tells it to measure the temperature.

Additional M commands have the same format as the aM! command. Data collection always begins with the D0 command.

Error Detection

To enhance the error detection capability in SDI-12 data collection systems, a variation of the Start Measurement Commands (M!, M1! ... M9!), request that data be returned with a 16 bit Cyclic Redundancy (CRC) appended to it.

The data logger checks the CRC code to see if the data arrived without error. If there is an error, the data recorder has a responsibility to perform retries, as needed, to try and get the data without error.

The details about how CRCs are computed and how they work is beyond the scope of this class.

To add error detection the letter C is appended to the M commands, which causes the CRC to be appended to all responses to all D Commands.

Here is an example:

0MC!00001<CR><LF>
0D0!0+3.14OqZ<CR><LF>

Concurrent Commands

A concurrent measurement is one which occurs while other SDI-12 sensors on the bus are also taking measurements.

Service requests to not apply to concurrent measurement commands.

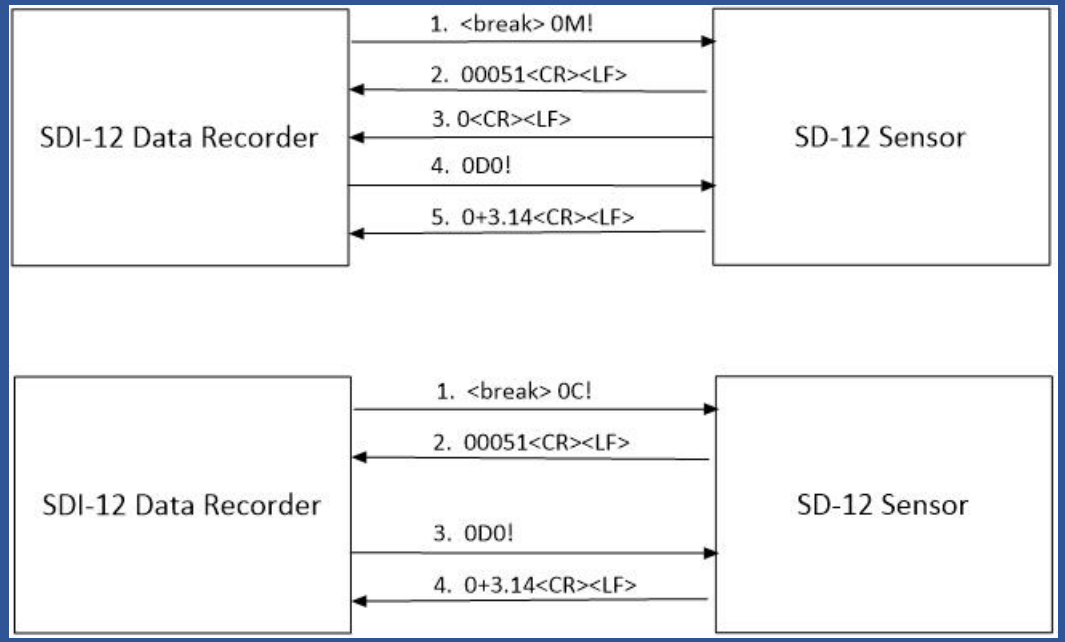
By using concurrent commands, a data recorder can tell two or more sensors to take measurements at the same time.

The concurrent measurement commands work, for the most part, just like the M commands. The difference is that a data recorder can issue breaks and send commands to other sensors on the SDI-12 bus while one more sensors are taking measurements.

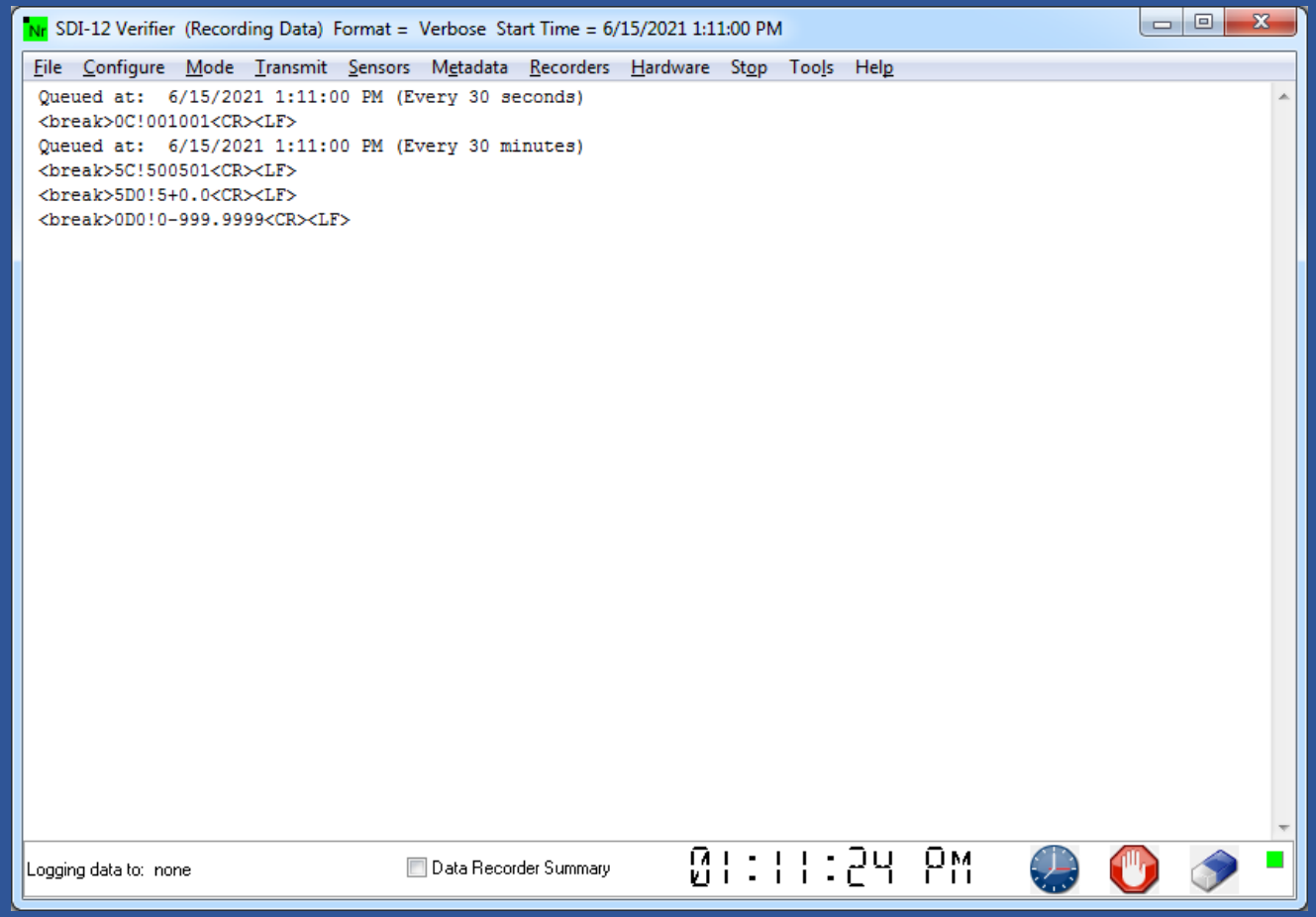
The concurrent commands are: aC! … aC9!.

The concurrent commands with a CRC are: aCC! … aCC9!

No Service Request With Concurrent Commands



Example of Concurrent Commands



Breaks

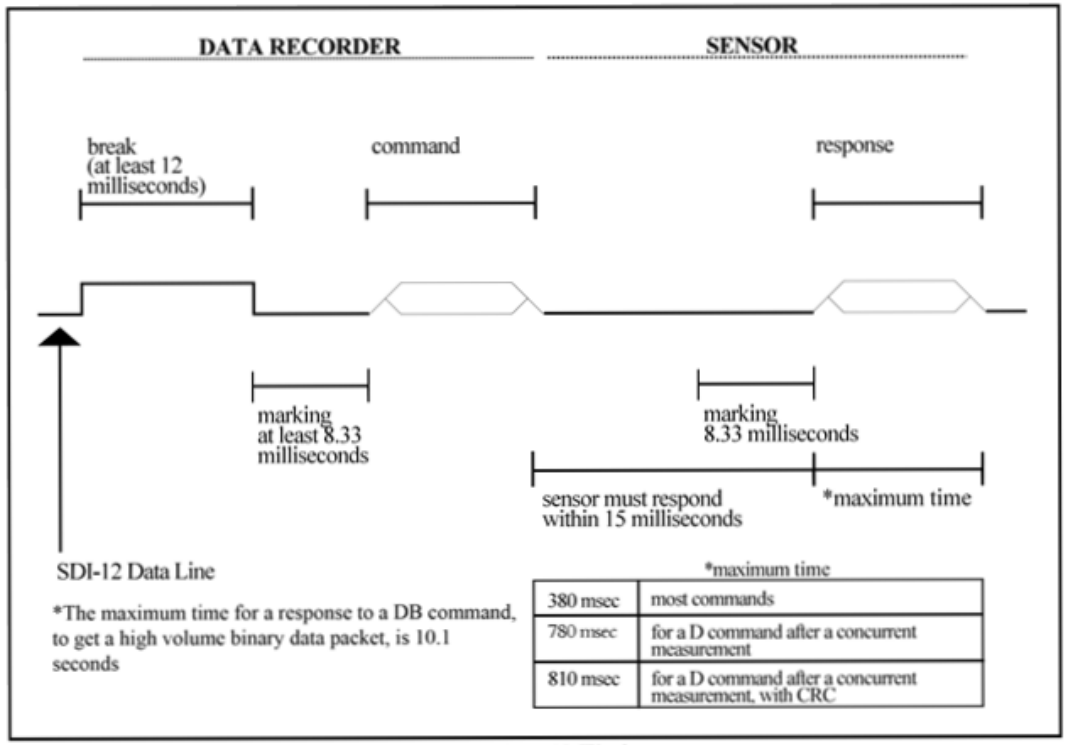
The purpose of a break to wake all sensors on the SDI-12 bus.

If, however, a break is issued while waiting for a service request, then the measurement process for that sensor is terminated. This provides a way to stop a measurement in progress.

For the concurrent commands, however, if a break is transmitted while a sensor is taking a concurrent measurement, the concurrent measurement is NOT stopped.

A concurrent measurement is stopped by sending a command addressed to the sensor that is making a concurrent measurement.

SDI-12 Timing



*maximum time	
380 msec	most commands
780 msec	for a D command after a concurrent measurement
810 msec	for a D command after a concurrent measurement, with CRC

Figure 3. SDI-12 Timing

Metadata Commands
Two Different Commands

1. Identify Measurement Commands
2. Identify Measurement Parameter Commands

Metadata Commands
Identify Measurement Commands

The identify measurement commands provide a means to get the response to a command without actually initiating a measurement.

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

0IM!00001<CR><LF>

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

0IM!00109<CR><LF>

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

0IC5!001099<CR><LF>

Metadata Commands
Identify Measurement Parameter Commands

The identify measurement parameter commands provide details about the parameters returned by a particular command.

Three fields, separated by commas and terminated with a semi-colon:

1. Standard Hydrometeorological Exchange Format (SHEF) code or other description
2. Units
3. Optional Fields - More Detail

a. Generic example:

0IM_001!0,field1,field2,field3;<CR><LF>

b. Metadata for an M command, showing that the M command takes a precipitation measurement, using optional field 3 to describe the data value:

0IM_001!0,PR,mm,precipitation rate per day;<CR><LF>

Identify Measurement Commands Details Metadata Fields 1, 2, and 3

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.

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 contain a concise identification of the parameter as determined by the sensor manufacturer. The recommendation, however, is to use a SHEF code when an appropriate SHEF code does exist.

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.

Field Three, 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. For example, 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, through the terminating semicolon, is 75 characters.

High Volume Commands

There are commands to get up to 999 data values from an SDI-12 sensors as ASCII characters, similar to all other commands.

There are commands to get up to 999 data values as binary numbers, not ASCII characters, which is faster and more efficient.

The details about the high volume commands are beyond the scope of this class.

Versions of SDI-12

1988 – 1994 1.0 & 1.1 (technically identical) the basic command set

1996 - Version 1.2	added concurrent commands
2000 - Version 1.3	add CRC error detection commands
2016 - Version 1.4	added high volume commands
	added metadata commands

A version 1.4 data recorder will work with a version 1.0 sensor.

Help

Send your questions to technical@sdi-12.org.
You will get an answer, usually within two business days.

You can get a copy of the SDI-12 Specification here:
sdi-12.org/specification