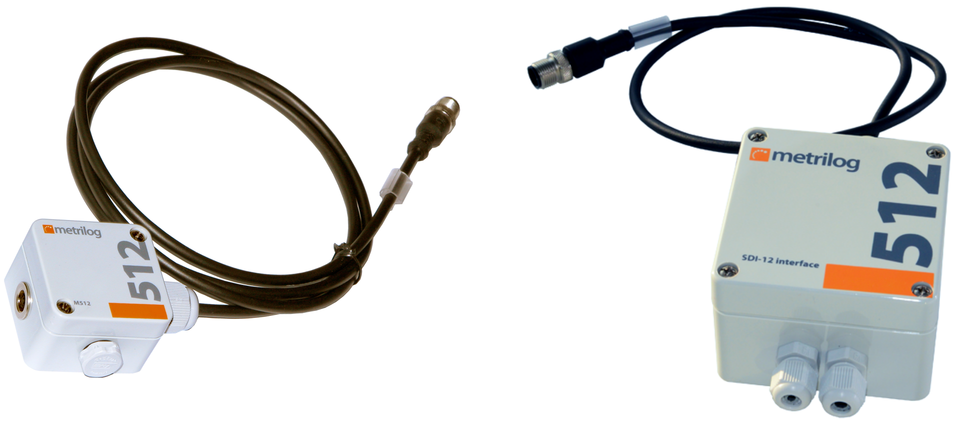
630.020

metrilog



SDI-12 Analog Adapters M512C, M512D

User Guide

Manual rev 1.1, issued 6 Apr 2013

Metrilog Data Services GmbH, Office Park 4, C38, 1300 Wien-Flughafen, Austria
www.metrilog.com

Table Of Contents

Introduction

The 3-Way Analog Adapter model M512C and D is intended to be used on SDI-12 bus systems. It converts several analog, pulse and digital inputs to the SDI-12 bus protocol and specification.

With the M512 you can connect:

- up to 3 analog sensors with output ranges of 0 to 1V, 0 to 1.25, 0 to 2.5V or 0 to 5V (individually programmable) and
- up to 2 pulse counters (32 bit) or
- up to 2 frequency counters (5 kHz max), or
- up to 2 digital inputs (state sense).

The adapter conforms to the SDI-12 bus specification version 1.3. For additional information on the SDI-12 bus, please consult the following document: "SDI-12, A Serial-Digital Interface Standard for Microprocessor-Based Sensors, Version 1.3". The document can be found on the SDI-12 Support Group's web site at *http://www.sdi-12.org*.

While the physical connexion to the bus can be configured either for native SDI-12 signaling or RS-485, the protocol employed is always SDI-12.

This manual addresses the installation and configuration of the M512C and M512D Analog Adapters and all their sub-variants. Previous versions of the M512 Analog Adapter (i.e. M512A or M512B) are not covered in this manual as these devices implement only a subset of the functionality of the the M512C/D Analog Adapter.

A special chapter for advanced operations intended for programmers or support technicians is also included.

Differences Between the M512C and M512D Analog Adapters

The M512C Analog Adapter has only one M8 input connector and has a smaller housing. The M512D Analog Adapter is functionally identical to the M512C Analog Adapter but its housing is larger to accommodate more M8 connectors or cable glands. The M512D Analog Adapter is offered in four sub-variants: M512D-1 with one M8 connector (thus 100% compatible with the M512C), M512D-2 with two M8 connectors, M512D-3 with three M8 connectors and M512D-CC with two cable glands and internal cage clamps (terminal blocks) to connect the sensors' cables. Note that the two, i.e. three M8 connectors are internally wired in parallel.

Except otherwise mentioned, this manual applies to all variants of the M512C and D Analog Adapter. Throughout the manual all references will be made to the M512 Analog Adapter.

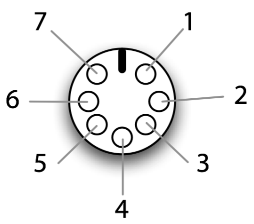
Installation

The M512C Adapter has a 7-pin Binder M8 connector used to attach the sensors, and a cable with an M12 4-pin connector for the SDI-12 bus. To attach more than one sensor to the input connector you can use a special 2- or 3-way M8 T-cable, or a specially designed, custom distribution box.

The M512D Adapter, models M512D-2 and M512D-3 offers a built-in T-cable equivalent, thus making an 2- or 3-way M8 T-cable redundant. These adapters allow the connection of more than one sensor. Note however, that in both cases the inputs are simply connected in parallel, thus when connecting more than one sensor you must pay attention to the sensors' cabling to avoid shorting the outputs.

M8 Input Connector

The pin-out of the input connector(s) is given below (as seen from the outside).



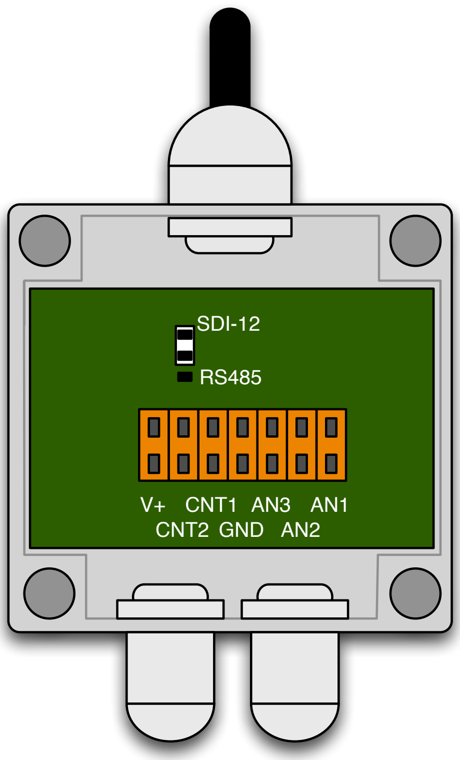
1	Analog Input 1 (AN1)
2	Analog Input 2 (AN2)
3	Analog Input 3 (AN3)
4	GND, minus
5	Counter 1, Digital 1 (CNT1, FQ1, DIG1)
6	Counter 2, Digital 2 (CNT2, FQ2, DIG2)
7	+Vbatt, switched

The power to sensors is supplied on pin 7 (+Vbatt) before the sensors are sampled. The time elapsed between applying power and the actual sensor sampling is set to 2 seconds by default, but it can be changed (see the XGV command). All input sensors must be referenced to the ground pin (pin 4). Keep the analog input cables short and always use only shielded cable! The maximum recommended cable length is 10 meter.

M512D-CC Input Cage Clamps

The M512D-CC variant has two PG7 cable glands that can accept cables from 2,5 to 6,5 mm in diameter. To reach the cage clamps (terminal blocks) you need to open the unit. To attach the wires to the cage clamps, use a suitable tool or small screwdriver. Press with the tool downwards while inserting the wire sideways. The cage clamp connections are given below:

AN1 — Analog Input 1



AN2 — Analog Input 2

AN3 — Analog Input 3

GND — Ground (shielding)

CNT1 — Counter 1, Digital 1 (also FQ1, DIG1)

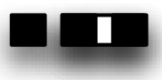
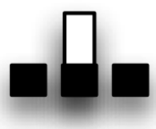
CNT2 — Counter 2, Digital 2 (also FQ2, DIG2)

V+ — +Vbatt, switched

The SDI-12/RS-485 jumper is factory delivered in the position SDI-12. For additional details on the various configuration possibilities in RS-485 mode, see the section “SDI-12/RS-485 Jumper” below.

SDI-12/RS-485 Jumper

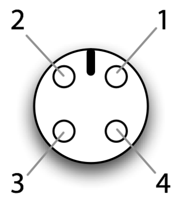
By default the M512 is delivered pre-configured for SDI-12 signaling. You can change it to RS-485 by opening the unit and properly placing the jumper, as described below (the drawing is valid for the M512C variant, but the electrical description is valid for all units).

		SDI-12 (default)
		RS-485, terminating unit; that means this device is situated at one end of the RS-485 bus and the internal terminating resistor (110 ohm) will be enabled.
		RS-485, on the bus but not at one of its ends; in this position, no terminating resistor is enabled.

Note: after opening the unit, make sure that the O-ring is properly seated before fastening the lid.

M12 SDI-12/RS-485 Plug

The pin-out of the SDI-12/RS-485 plug is shown below.

	1	+Vbatt (6.5 to 20 Volt)	brown
	2	-Vbatt, GND	white
	3	RS-485 A (data)	blue
	4	SDI-12, RS-485 B (data)	black

Notes: The colors in the last column are valid for the standard cable delivered with the device. Pin 3 is not used in SDI-12 mode.

Operating modes

The M512 adapter can be operated in standard "one-shot" mode or in periodic mode. Standard mode means that the device will not sample inputs or do frequency measurements as long as it is not instructed by means of a M or C command. In periodic mode, the device will perform measurements at specific intervals and will store the intermediate results (average, minimum and maximum) for subsequent retrieval by the data logger. By default the periodic mode is disabled; to enable it see the XGS command. Note that by enabling the periodic mode, the power consumption will increase, and it is directly related to the number of samples per minute.

For retrieving the data accumulated during the periodic mode use the C2 or CC2 commands (there is no equivalent M2 command); note that while the periodic mode is activated, the standard M and C commands are still available. You should however take care of the pulse counters: if the reset mode is set to automatic, the periodic mode may interfere with the standard mode, and the other way around as the counters are cleared after being read by either command, i.e. M/D0 or C2/D0.

Analog Inputs

The analog inputs can be individually configured for the following input ranges (see the XA command):

- 0 to 0,625 volt
- 0 to 1 volt
- 0 to 1,25 volt
- 0 to 2,5 volt

- 0 to 5 volt

By default, all inputs are set to 2,5 volt range.

Before sampling the analog inputs, the power is applied to the switched power output (pin 7); the amount of time elapsed between the power is applied to the sensors and the actual sampling is set using the XGV command (by default it is 2 seconds). This is needed to allow the sampled sensors to settle their outputs.

If the periodic mode is activated, each analog input can be set to one of the following averaging types (also by means of the XA command):

- none — in this case the last sample is returned.
- arithmetic — a standard arithmetic average of all samples will be returned (the sum of all samples divided by the number of samples).
- circular — this is a special type of averaging suitable for wind direction sensors; it takes into account the fact that the values are situated on the perimeter of a 360° circle.
- leaf-wetness table — a custom table devised for certain leaf-wetness sensors; it returns values between 0 and 10.

The output values of the sampled sensors are scaled by means of the minimum and maximum values. These values represent what should the device return in case of 0 volt input (minimum value) as well as for the maximum input range set (i.e. 0,625, 1, 1,25, 2,5, or 5 volt — max. value). For more details regarding the configuration of the analog input channels see the XA command.

Pulse Counters

The two pulse counters can independently count pulses with a frequency of up to 5 kHz (PC1) or 4 kHz (PC2). If both counters are used in parallel, then the maximum input frequency is 2 kHz for both. If the pulses to be counted have a frequency in excess of 30 Hz, then you need to reduce the counters de-bouncing delay (default 20 ms); for frequencies in excess of 150 Hz, you must also change the sleep mode (see also the XGC and XGF commands). The tradeoff is that the power consumption will increase.

The counters are 32-bit wide, that means they will reset after 4.294.967.295 pulses. The counters can be reset either automatically or manually (using the XCR command). By default the counters are configured for automatic reset; you can change this behavior by using the XP command.

Note that the pulse counter inputs are pulled-up to 5 volt; a simple contact to the ground suffices to activate the counter.

Frequency Counters

The two frequency counters can measure frequencies up to 5 kHz (FQ1) or 4 kHz (FQ2). The same features and limitations as for the pulse counters apply. If both frequency counters are used in parallel, then the maximum measurable input frequency is 2 kHz for both. If one counter is used to measure lower frequencies (e.g. FQ2 for up to 100 Hz), then the other can be used to the full specified extent (e.g. FQ1 for up to 5 kHz). If the frequencies to be measured are in excess of 30 Hz, then you need to reduce the counter's de-bouncing delay; for frequencies in excess of 150 Hz, you must also change the sleep mode (see also the XGC and XGF commands). The tradeoff is that the power consumption will increase.

The sampled frequency can be scaled to fit the measured parameter (e.g. a wind speed sensor) by using a multiplying factor (see the XF command). In addition, the precision of the measurement can be improved by increasing the time the gate is open, again at the expense of the power consumption (see the XGP command).

Digital Inputs

The two digital inputs return the status at sampling time (logical 0 or 1, false or true). Left open, the inputs will return 1 (true) as they are pulled-up to the internal 5 volt line. The inputs are EMI protected.

Configuration

The M512 supports the SDI-12 protocol version 1.3. It has various operating modes and can be configured to accept a large variety of sensors. All configuration changes are done by means of SDI-12 extended commands (aX!). The changes are non-volatile, that is, they are stored in EEPROM and are maintained even if the power is removed.

The M512 includes following logical sensors:

- three analog inputs, independently programmable as 0 to 1V, 0 to 1,25V, 0 to 2,5V or 0 to 5V; they are identified as AN1, AN2 and AN3 respectively.
- two digital inputs that return either 0 or 1 when read; they are identified as DIG1 and DIG2.
- two 32-bit pulse counters that can be reset manually (through an X command) or automatically when read; they are identified as CNT1 and CNT2.
- two frequency counters; they are identified as FQ1 and FQ2.

Certain commands may also return the analog sensors supply voltage or the time elapsed between successive measurements (in seconds).

The M512 accepts the following data retrieval commands:

Command	Response	Remarks
M! + Dn! C! + Dn!	AN1, AN2, AN3, CNT1, DIG1, CNT2, DIG2, Vsen	8 values returned
MC! + Dn! CC! + Dn!	AN1, AN2, AN3, CNT1, DIG1, CNT2, DIG2, Vsen, CRC	8 values plus CRC returned
M1! + Dn! C1! + Dn!	FQ1, FQ2	2 values returned
MC1! + Dn! CC1! + Dn!	FQ1, FQ2, CRC	2 values plus CRC returned
C2! + Dn!	AN1avg, AN1min, AN1max, AN2avg, AN2min, AN2max, AN3avg, AN3min, AN3max, FQ1avg, FQ1min, FQ1max, FQ2avg, FQ2min, FQ2max, CNT1, CNT2, Interval	18 values returned; this mode is by default disabled, see also XGS command
CC2! + Dn!	AN1avg, AN1min, AN1max, AN2avg, AN2min, AN2max, AN3avg, AN3min, AN3max, FQ1avg, FQ1min, FQ1max, FQ2avg, FQ2min, FQ2max, CNT1, CNT2, Interval, CRC	18 values plus CRC returned; see also above remark

Examples

In the examples below, default values are used for the SDI-12 address ("A") and most configuration parameters (for default parameter values see also "Extended Commands" below).

```
AM!
A0048
AD0!
A+29.48+14.11+0+0+1+0+1+11.31
```

```
ACC!
A00408
AD0!
A+29.69+13.94+0+0+1+0+1+11.31CCk
```

```
ACC2!
A00718
AD0!
A+29.96+29.69+30.21+13.72+13.43+13.96+0+0+0+0+0+0+0+0+0+0+0+600DAg
```

Other SDI-12 Commands

The device recognizes following commands:

a — Acknowledge active command

Request:

a! (where "a" is the device's address)

Response:

a (where "a" is the device's address)

Description:

This is a standard SDI-12 command. By means of this command a data logger asks a sensor if it is active on the bus.

Example:

```
A!
A
```

I — Identification command

Request:

I!

Response:

ID data.

Description:

This is a standard SDI-12 command that returns ID data of the device (protocol version, manufacturer, type and serial number).

Example:

```
AI!
A13MetrilogM512rC1.3.970#0000000
```

? — Address query command

Request:

?!

Response:

a (where "a" is the device's address)

Description:

This is a standard SDI-12 command, by which a sensor can be queried for its address. Obviously, only one sensor can be attached to the bus, otherwise a bus conflict will occur.

Example:

```
?!
A
```

A — Change address command

Request:

Aa!

Response:

The new address (a).

Description:

This is a standard SDI-12 command that allows changing the address of a device.

Example:

```
AAB!
B
```

V — Start verification command

Request:

V!

Response:

Number of parameters that will be returned by the next D! command. The D! command can be issued immediately and will return the uptime of the device (in minutes) as well as the number of resets received to date. There are four types of resets that are counted, and they are displayed in this order: power-on, watchdog, brown-out, others.

Description:

This is a standard SDI-12 command used to retrieve monitoring information about the device.

Example:

```
AV!
A0005
AD0!
A+20524+1+2+1+0
```

Extended Commands

The configuration of the M512 adapter can be changed through a series of extended commands. All configuration changes are non-volatile, that is, they are stored in EEPROM and are retained even if the power is removed.

For multi-parameter commands not all parameters are mandatory; however if you need to change e.g. only the last parameter in a list, then you have to send all of them.

XA<n> — Set parameters for analog channel <n>

Request:

XA<ANn>,<low>,<hi>,<avg_type>,<range>!

Response:

Address.

Description:

<ANn> is the channel number (1 to 3).
<low> is the value that should be returned when the sensor delivers its lowest value; this is a floating point value.
<hi> is the value that should be returned when the sensor delivers its highest value; this is a floating point value.
<avg_type> is the averaging type; this can be one of the following:
0 — none (last measured value will be returned);
1 — arithmetic average
2 — circular average (suitable e.g. for wind direction sensors)
3 — built-in leaf wetness table (leaf-wetness sensors)
<range> is the voltage input range observed by the adapter and can be:
0 — 0 ... 2,5 V (default)
1 — 0 ... 5 V
2 — 0 ... 1,25 V
3 — 0 ... 1 V
4 — 0 ... 0,625 V

Remarks:

To read the current parameter values, issue the XA<n> command without any parameter.
The default values for <low> and <hi> are:
AN1: -40, +60 (temperature sensor)
AN2: 0, 100 (RH sensor)
AN3: 0, 10 (leaf-wetness sensor)
The <low> and <hi> values can be also negative.

Examples:

```
AXA1,0,500,1,1!
A
```

```
AXA1!
A0,500,1,1
```

XP<n> — Set parameters for pulse counter n

Request:

XP<CNTn>,<increment>,<reset_mode>!

Response:

Address.

Description:

<CNTn> is the counter number (1 or 2)
<increment> is the value that will be added to the counter when a pulse is registered; default is 0,2.
<reset_mode> can be set to manual or automatic; in automatic mode, the counter is reset after a M + D or C + D commands sequence, while in manual mode the counter should be cleared when needed by means of the XCR<n> command. Accepted values are:
0 — manual mode
1 — automatic mode (default)

Remarks:

To read the current parameter values, issue the XP<n> command without any parameter. The <increment> value can be also negative.

Examples:

```
AXP1,0.01,1!
A
```

```
AXP1!
A0.01,1
```

XCR<n> — Reset counter n

Request:

XCR<CNTn>!

Response:

Address.

Description:

<CNTn> is the counter number to be reset (1 or 2).

Example:

```
AXCR1!
A
```

XF<n> — Set parameters for frequency counter n

Request:

XF<FQn>,<multiplier>!

Response:

Address.

Description:

<multiplier> is the value the frequency registered on channel <n> should be multiplied with (default 1).

Remarks:

To read the current parameter value, issue the XF<n> command without any parameter. See also the XGF command.

Examples:

```
AXF1,12.7!
A
```

```
AXF1!
A12.7
```

XGV — Set delay between applying power to the sensors and sampling

Request:
XGV<pwr_to_sample>!

Response:
Address.

Description:
<pwr_to_sample> is the delay between the adapter applies power to the sensors and sampling them. This parameter is given in seconds (default is 2).

Remarks:
To read the current parameter value, issue the XGV command without any parameter. All analog channels (AN1 to AN3) are affected. Maximum accepted value is 999 (seconds).

Examples:

```
AXGV5!
A
```

```
AXGV!
A5
```

XGP — Set the sampling period for frequency counters

Request:
XGP<period>!

Response:
Address.

Description:
<period> is the time span where the frequency counter's gate is open to accumulate pulses (the time base). This parameter is given in seconds (default is 1). The frequency counter will compute the frequency as
Freq (Hz) = pulses/period

Remarks:
To read the current parameter value, issue the XGP command without any parameter. Both frequency counters (FQ1 and FQ2) are affected by this command. Acceptable values are between 1 and 100.

Examples:

```
AXGP4!
A
```

```
AXGP!
A4
```

XGS — Set sampling interval

Request:
XGS<sample_interval>!

Response:
Address.

Description:
<sample_interval> is the time elapsed between samples. This parameter is given in seconds; the default is 0, i.e. advanced sampling mode is disabled.

Remarks:

If the value is greater than 0, then the adapter enters in an advanced sampling mode: every <sample_interval> time it will sample the input sensors, will measure the frequency of the input signals (if any), and sum up the pulse counters. For the analog inputs and the frequency counters the minimum, maximum and the average for all measurements are kept in an internal buffer. All these values will be returned when requested by means of the C2! + D! or CC2! + D! commands (see also the data retrieval table), after which the internal buffer is cleared and a new set of values starts being collected.
Note that this mode is power intensive: the power consumption is higher when the <sample_interval> is smaller.
To read the current parameter value, issue the XGS command without a parameter. The maximum accepted value is 65'535 (seconds).

Examples:

```
AXGS!
A0
```

```
AXGS60!
A
```

```
AXGS!
A60
```

XGC — Set counters de-bouncing delay

Request:

XGC<cnt_debounce_time>!

Response:

Address.

Description:

<cnt_debounce_time> is the length of time the input signal is filtered of bounces. This parameter is given in milliseconds; the default is 20.

Remarks:

To read the current parameter value, issue the XGC command without any parameter. Both pulse counters (CNT1 and CNT2), as well as both frequency counters (FQ1 and FQ2) are affected. The maximum accepted value is 255 (milliseconds).

Examples:

```
AXGC4!
A
```

```
AXGC!
A4
```

XGF — Set sleep mode

Request:

XGF<fast_sleep_mode>!

Response:

Address.

Description:

<fast_sleep_mode> can be enabled or disabled and defines the time required by the adapter to switch from sleep mode to active mode. By default the fast sleep mode is disabled. The parameter can have the following values:
0 — fast sleep mode disabled
1 — fast sleep mode enabled

Remarks:

To read the current parameter value, issue the XGF command without any parameter. Note that the power consumption when fast sleep mode is enabled is higher (see also the Technical Specifications at the end of this manual). If you use the frequency counters to measure frequencies in excess of 150 Hz, then the fast sleep mode must be enabled.

Examples:

```
AXGF!
A0
```

```
AXGF1!
A
```

```
AXGF!
A1
```

Practical Configuration Examples using Extended Commands

In this section several practical applications of the M512 in conjunction with various sensors are shown.

Pressure sensor

A three-wire pressure sensor delivers 0 to 5 volt output corresponding to a pressure of 0 to 20 bar. We assume the sensor is connected to channel 1. Following command will be required:

```
AXA1,0,20,1,1!
```

This would set channel 1 for a scale of 0 (min) to 20 (max) bar, the average type to arithmetic and the input range of the M512 to 0 to 5 Volt.

RH/Temp/Barometric sensors with average/min/max output

Suppose we have three meteorological sensors delivering the air Relative Humidity, Temperature and Barometric Pressure. In addition, we want to sample all those parameters as often as possible and deliver the average, the minimum and the maximum for all three, measured over a certain time span.

We assume channel 1 for Air Temperature, channel 2 for RH and channel 3 for Barometric Pressure. We assume also that the Air Temperature and RH sensors have a 0 to 1 Volt output, while the Barometric Pressure sensor has a 0 to 5 Volt output. Following configuration steps must be done:

- Set the minimum, maximum, and range for each channel; the temperature range is -40 to +60°C, RH is 0 to 100% and the pressure 800 to 1060 hPa:

```
AXA1,-40,60,1,3!          // channel 1 Temperature
AXA2,0,100,1,3!      // channel 2 RH
AXA3,800,1060,1,1!   // channel 3 Barometer
```

- Enable periodic measurement with 10 seconds intervals (that means the sensors will be sampled every 10 seconds):

```
AXGS10!
```

To retrieve the results, use the AC2! or ACC2! command (see also the commands table at begin of the chapter). Note that the configuration above will draw moderately more power, due to the 10 seconds sensor sampling interval.

Pulse output Wind speed sensor

Wind speed sensors are mostly implemented to deliver a pulse stream output. A reed-relay contact is switched on and off depending on the rotational speed of the wind cups. The manufacturer sensor data is usually specified as a frequency corresponding to a certain wind speed. It may be also expressed as pulses per second.

In this example we assume that the maximum wind speed of the sensor is 60 m/s which would result in an output of 114 pulses/second (114Hz). We assume frequency counter 1 is used:

- To increase the precision, set the sampling period to 10 seconds:

```
AXGP10!
```

- Program the frequency counter 1 in such a way to deliver 60 m/s at 114 Hz (the multiplier is 60/114 = 0,526):

```
AXF1,0.526!
```

- Reduce the de-bouncing delay to 5 milliseconds:

```
AXGC5!
```

Note that for frequencies above 150 Hz, you would have had to change the sleep mode using the XGF command. In our example, as the maximum frequency delivered by the sensor is 114 Hz, this is not required.

To retrieve the results, use the AM1! or AMC1! commands; you may also use the AC2! or ACC2! commands if the periodic mode is enabled, thus you would get both the average and the gust wind speed (see also the previous example).

Advanced Functions

This chapter provides you information not required for normal use. The information in this chapter is intended for well trained technicians to pinpoint malfunctions or other device errors, as well as for developers that wish to implement special functions in their systems (e.g. remote firmware upgrade of the M512 adapter).

System Description

The M512 microcontroller includes two software modules: a boot-loader and the application proper. At reset the control is passed to the boot-loader which tries to determine if the application's area in the flash memory contains valid firmware. If this is true, then the boot-loader will launch the application, otherwise it will wait for specific boot-loader commands. As the device is intended to operate on a SDI-12 bus, all boot-loader commands are SDI-12 like.

The application itself is based on an open source preemptive multi-tasking operating system called "µOS++"; you can find more on this RTOS in Internet at *http://sourceforge.net/projects/micro-os-plus/*. The RTOS has also a limited set of commands that can help pinpoint possible problems.

The following chapters describe the commands specific to the operating system and the boot-loader.

The Operating System

This chapter does not attempt to describe the operating system itself, rather several OS commands adapted to the SDI-12 specification. As expected, they are implemented as SDI-12 X (extended) commands. All OS commands use the prefix X$.

While in the application mode, following OS commands are available:

X$H — List all OS commands

Request:

X$H!

Response:

List of accepted commands.

Example:

```
AX$H!
ATN,TS,TP,Tn,U,V,R,B
```

X$TN — List task names

Request:
X$TN!

Response:
List of tasks.

Remarks:
See also the X$Tn! command.

Example:

```
AX$TN!
AIDLE,sdi12,dacq,periodic,pulse
```

X$TS — List task stacks

Request:
X$TS!

Response:
A list with pairs of bytes representing the allocated and used stack space.

Description:
The comma separated pairs of numbers (uuu/aaa) represent the amount in bytes used (uuu) from the total allocated (aaa). The task order is the same as the one returned by the X$TN command.

Example:

```
AX$TS!
A83/135,178/255,231/255,129/165,59/125
```

X$TP — List tasks' priorities

Request:
X$TP!

Response:
A list of numbers representing tasks priorities.

Description:
Each element of the comma separated list of numbers represents the priority of a task. The higher the number, the higher is the priority. The IDLE task has the lowest priority (0). The tasks order is the same as the one returned by the X$TN command.

Example:

```
AX$TP!
A0,129,128,128,130
```

X$T<n> — Look-up task number to its name

Request:
X$T<n>!

Response:
The task name.

Examples:

```
AX$T0!
AIDLE
```

```
AX$T1!
Asdi12
```

```
AX$T2!
Adacq
```

X$U — Uptime

Request:

X$U!

Response:

The time elapsed since the last device restart; it is expressed in days, hours and minutes.

Example:

```
AX$U!
A2d,12h,48m
```

X$V — Operating system version and build number

Request:

X$V!

Response:

The µOS++ version number (major, minor) and the build number.

Example:

```
AX$V!
A2,0,411
```

X$R — Cause of last reset

Request:

X$R!

Response:

The MCU Status Register value and the reset cause. Following causes are possible:

— Software reset
— Power ON reset
— External reset (improbable)
— Brown-out reset
— Watchdog reset

Example:

```
AX$R1!
A0,soft
```

X$B<n> — Boot device

Request:

X$B<n>!

Response:

The address.

Description:

This command allows relaunching the application. It can be used to return to the boot-loader. If <n> is 0, then the application will be launched, while if <n> is 1, then the boot-loader will be entered. This allows a simple software reset:

while in application mode, the X$B0! command effectively restarts the application.

Remarks:

To find out in what mode the device is, issue the command without a parameter. If the result is 0, then the device is in application mode; if the result is 1, then the device is in boot-loader mode.

Examples:

```
AX$B!
A0
```

```
AX$B1!
A
```

```
AX$B!
A1
```

The Boot-loader

The boot-loader allows the remote update of the M512 adapter. The solution implemented operates over the SDI-12 bus, even when other SDI-12 sensors are attached to the bus. The commands to re-flash the device remotely were designed to be fully compatible with the SDI-12 specification. Thus, a data-logger that implements the boot-load mechanism allows for remote firmware changes. As an example, the Metrilog model T707-SDI RTU supports this functionality.

A description of all boot-loader commands is provided; a step-by-step method to re-flash the device is also included.

Commands can be sent with a CRC appended or not. It is highly recommended to use CRC for all block commands to avoid flashing the microcontroller with wrong data. The standard SDI-12 CRC is used. By convention, regular commands have the format X$... while the commands ending with a CRC are sent as XC$...

The response has the 3 CRC characters appended just before the CR/LF, as for all other SDI-12 commands. The request should end with *CCC*! (where CCC are the three CRC characters) — a bad CRC invalidates the command. This mode of operation is intended for machine-to-machine communication.

X$S — Read microcontroller signature

Request:

aX$S!

Response:

Processor signature bytes as hex values; for further information please read the Atmel® ATMega microcontroller series user manual.

Description:

Before starting the download process, an application needs to know what microcontroller is used. Based on this information it knows the block length and other programming parameters.

Example:

```
AX$S!
A1E,95,08
```

X$WF — Write a data block to Flash memory

Request:

X$WF<addr>,<len>,<bytes>!

Response:

The result (0 - OK, 1 - ERROR) and the estimated amount of time until ready, in milliseconds (in hex).

Description:

<addr> is the address where the first byte in the block will be flashed; <len> is the block's length; and <bytes> is a block of bytes of length <len>. The block of bytes must be encoded according to the command type: if CRC is specified (XC$WF), then base64 encoding is expected, otherwise hexadecimal. All other parameters are sent as ASCII hex values.
The block of bytes is checked (invalid address, block boundary, CRC if any); the result will be returned together with the estimated amount of time needed for flashing the block of data. If the result is not zero (ERROR), the estimated amount of time is irrelevant.

Remarks:

The block's length must be appropriate for the type of microcontroller detected through the "Read microcontroller signature" command.

X$WF — Flush buffer

Request:

X$WF!

Response:

The result (0 - OK, 1 - ERROR) and the estimated amount of time until ready, in milliseconds (in hex).

Description:

Flush the buffer. Writing to a block after another block automatically triggers the flush process. Manual flush is needed only for the last block.

X$RF — Read a data block from Flash

Request:

X$RF<addr>,<len>!

Response:

The result (0 - OK, 1 - ERROR) and the block of bytes (in hex).

Description:

<addr> is the first address in the microcontroller's flash area; <len> is the number of bytes to be returned.

Remarks:

If CRC is specified (XC$RF), data will be returned as base64 encoded, otherwise in hexadecimal.

Example:

```
AX$RF0,1F!
A0,0C944B000C9493000C9493000C9493000C9493000C9493000C9489120
C941A
```

X$EF — Erase Flash block

Request:

X$EF<addr>,<len>!

Response:

The result (0 - OK, 1 - ERROR) and the estimated amount of time until ready, in milliseconds (in hex). ERROR result may indicate an invalid address.

Description:

Erase the block of <len> bytes in flash memory starting at <addr>.

Example:

```
AX$EF0,2FF!
A0,18
```

X$CF — Compute CRC for a block of bytes

Request:

X$CF<addr>,<len>[,<crc>]!
X$CF!

Response:

The result (0 - OK, 1 - ERROR) and the estimated amount of time until ready, in milliseconds (in hex); or the result and the calculated CRC value (see below).

Description:

<addr> is the block begin address; <len> is the block's length; <crc> is the CRC value (if known, optional).
The command is done in two steps:
— First, issue the command in its first variant;
— After waiting the required amount of time, issue the command in its second variant. If the CRC was known and sent as third parameter with the first command, then the result 0 - OK or 1 - ERROR indicates if the CRC was OK or not. If the CRC was not sent with the first command, then the computed value of the CRC will be returned in addition to the result. In this case, a nonzero result may indicate an invalid address.

Examples:

```
AX$CF0,2FF,D5C1!
A0,18

AX$CF!
A0

AX$CF0,2FF!
A0,18

AX$CF!
A0,D5C1
```

Sequence to flash a device

To re-flash a M512 device, perform the following steps/commands (we assume that the M512 has the default SDI address "A"):

- reboot to boot-loader

```
AX$B1!
A
```

- verify if in boot-loader mode

```
AX$B!
A1
```

- get version and signatures

```
AX$V!
A00,01,0339

AX$S!
A1E,96,0A
```

- erase flash

```
AX$EF0,1234!
A0
```

- program flash

```
AX$WF0,8,16,0123456789ABCDEF!
A0,05

AX$WF8,8,16,0123456789ABCDEF!
A0,05

and so on...

AX$WF
A0,05
```

- at the end, verify CRC

```
AX$CF0,777,65535!
A0,0800
```

- wait 2048 milliseconds and check if CRC is OK

```
AX$CF!

A0
```

- CRC OK, reboot into application mode

```
AX$B0!
A
```

Note: the programming example above is shown without a CRC therefore all data transfers are in hexadecimal. A practical implementation should use CRC and all data transfers will be done as base64 encoded.

Following restrictions for data download apply:

- length should be less than microcontroller's flash block;
- data blocks should not cross microcontroller block boundary;
- the master should not access the bus before the returned duration (the boot-loader will not respond anyway);

The operation is stateless, the block is read, bytes are overwritten, then the block is written back to the device.

Technical Specifications

Parameter	Value
Inputs	3 analog, 2 digital, 2 pulse counters
Analog inputs range	0-1V, 0-1.25 V, 0-2.5V, 0-5V (programmable)
Analog inputs accuracy	better than ±0.3% on any input range and for the whole specified temperature range
Digital inputs levels	5 Volt CMOS (5 Volt pull-up)
Pulse counters levels	5 Volt CMOS (5 Volt pull-up)
Pulse counters max. input frequency	with only one input operated at a time: PC1: 5 kHz, PC2: 4 kHz with both inputs operated simultaneously: PC1, PC2: 2 kHz
Pulse counters width	32 bits (4,294,967,295 pulses before reset)
Output	SDI-12 or RS-485
Power supply	6.5 to 20 Volt
Power consumption (@12 V)	Idle: max. 150uA (fast sleep mode disabled) Idle: max. 800uA (fast sleep mode enabled) Operating: max. 25 mA
Operating temperature	-30º - 70ºC
Environmental protection class	IP65
Dimensions (connexion cable and/or cable glands not included)	M512C: 52x50x35 mm M512D: 82x80x55 mm
Weight (connexion cable not included)	M512C: 65 g M512D: 90 g

Note: all specifications subject to change.