Sensorex

Description interface logiciel/ Software interface description

High performance MEMS
Inclinometer / Accelerometer

690 040 xxx

MEGGiTT
smart engineering for
extreme environments

Presentation

SX40xxx products have an RS485 serial communication bus and a USB connection.

Via these digital buses, the AppSx40000 software mainly allows:

- Acquire and display in real time information from the SX40xxx connected to the RS485 link or the USB link,
- Save acquisitions to a file,
- Configure some product parameters (bandwidth, baud rate, ...)

The RS485 link responds to the MODBUS RTU protocol and the USB 2.0 connection (full-speed) responds to a standard HID "plug & play" interface defined in the Device Class Definition for Human Interface Devices (HID) v1.11). The available low-level MODBUS RTU command sets are defined in this document.

For USB connection, a USB cable ref 90507537 is available for purchase from MEGGITT SENSOREX to easily connect your PC or USB host to the product sx40xxx.

In this document are also described the addresses RAM and EEPROM of the product accessible by the end user.

Software AppSx40000

Installation :

It is strongly recommended to install this application on a PC with the operating system **windows SEVEN at a minimum**

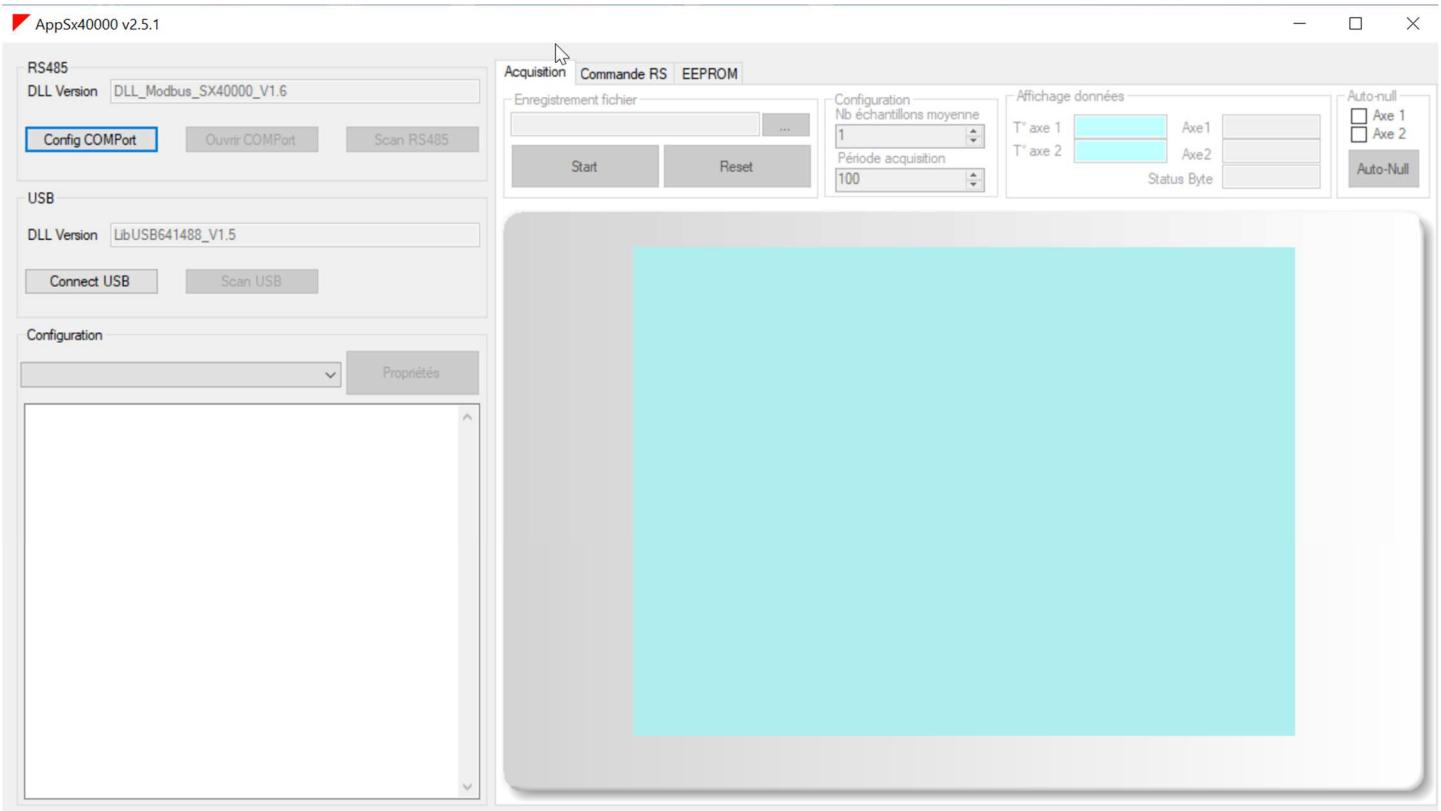
For a 32bit system processor, install setup file **x86**

For a 64bit system processor, install setup file **x64**

Presentation :

The application is split into two parts.

The left side is dedicated to the access configuration of the product SX40000 and the right part is dedicated to the display and the recording of the data read from the product.



Access configuration :

Before accessing the SX40000 product, the user must indicate on which type of bus the product SX40000 is connected.

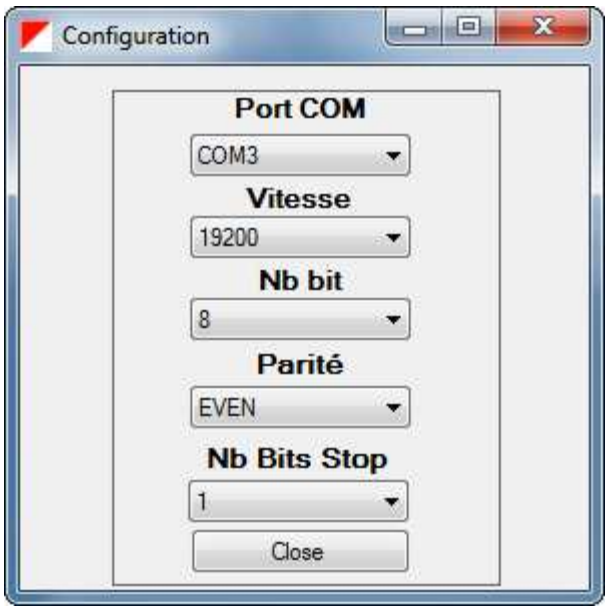
If the SX40000 is connected via an RS485 bus, the user must first configure the communication port used.

To do this, click on the "Config COMPort" button.

In the window, the user must then set:

- The port number used
- The speed of transmission
- The number of bits (8 for the SX40000)
- Parity (even for the SX40000)
- The number of stop bits (1 for the SX40000)

After clicking on "Close", the "Open COMPort" button is accessible.

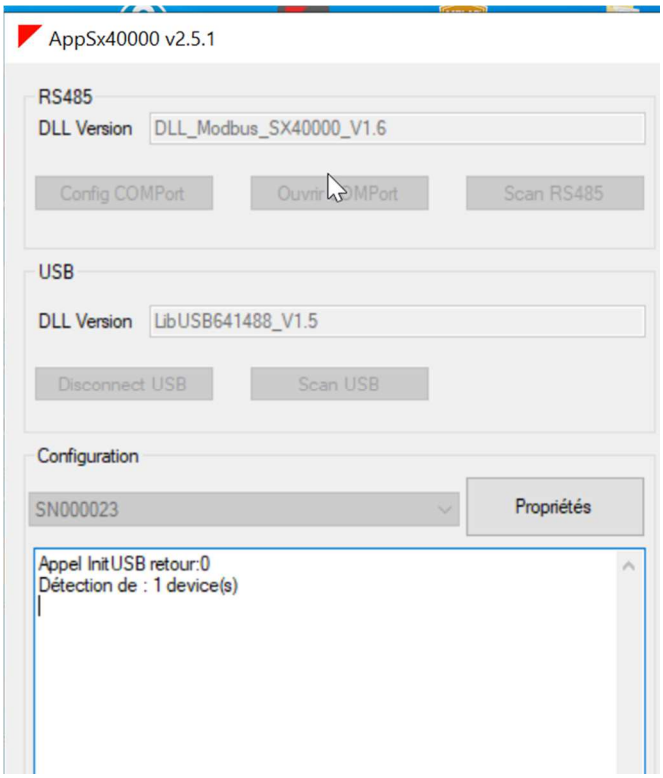


SX40000 products scan

Click on the "Open COMPort" button to access the "Scan RS485" button, or the "Connect USB" button to access the "USB Scan" button depending on the connection mode of your product.

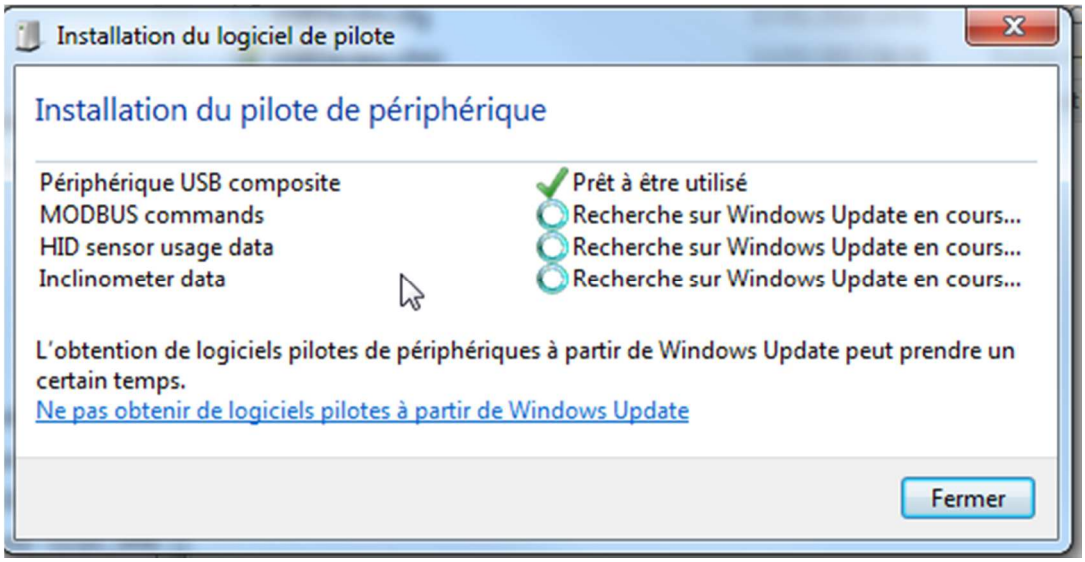
Before you can access the SX40000 product it is necessary to detect at least one SX40000 product connected to the selected port. For this the user must click on the button "Scan RS485" or "Scan USB". The scan operation on the RS485 bus is interruptible by the user.

After detecting an SX40000 product, it can be accessed via the drop-down menu in the "Configuration" section.



IMPORTANT note :

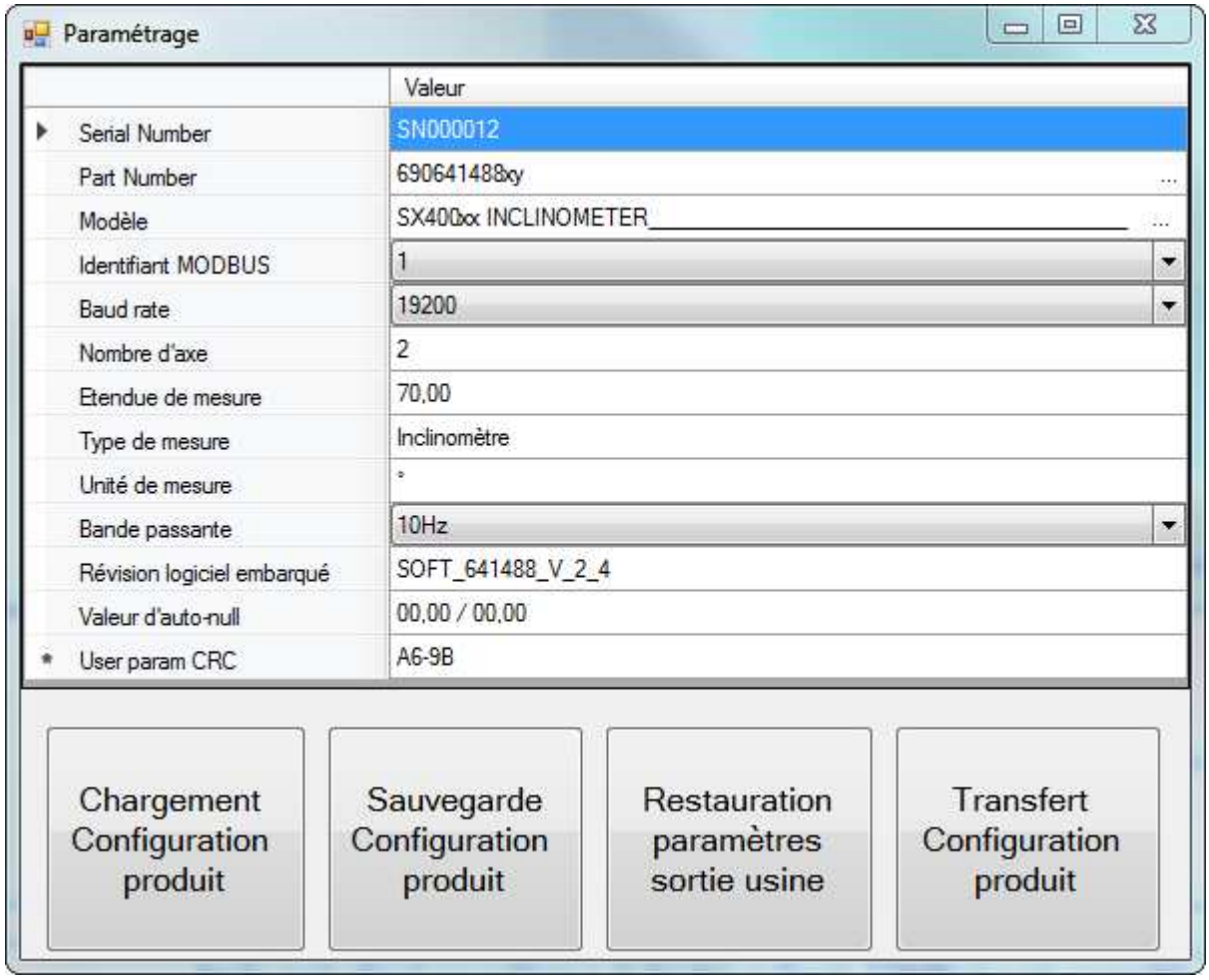
Only at the first connection of the product to a USB port of a new PC, before pressing "USB scan", wait until the end of the detection of the product by the PC (the 4 interfaces must be identified "ready to be used ").



SX40000 settings

The user can configure certain SX40000 parameters via the "Properties" button.

The configuration and product identification settings are displayed in the following window:



The user can configure:

- The MODBUS identifier (from 1 to 247)
- The baud rate on the RS485 port
- The bandwidth of the SX40000 axis (s)

After selecting the desired parameters, click on "Transfer Product Configuration" to apply the user's choices.

A click on the button "Restore factory settings" restores the configuration of the product on delivery.

The "Product Configuration Loading" and "Product Configuration Backup" buttons allow the user to maintain the product configuration in a file.

Data view of SX40000

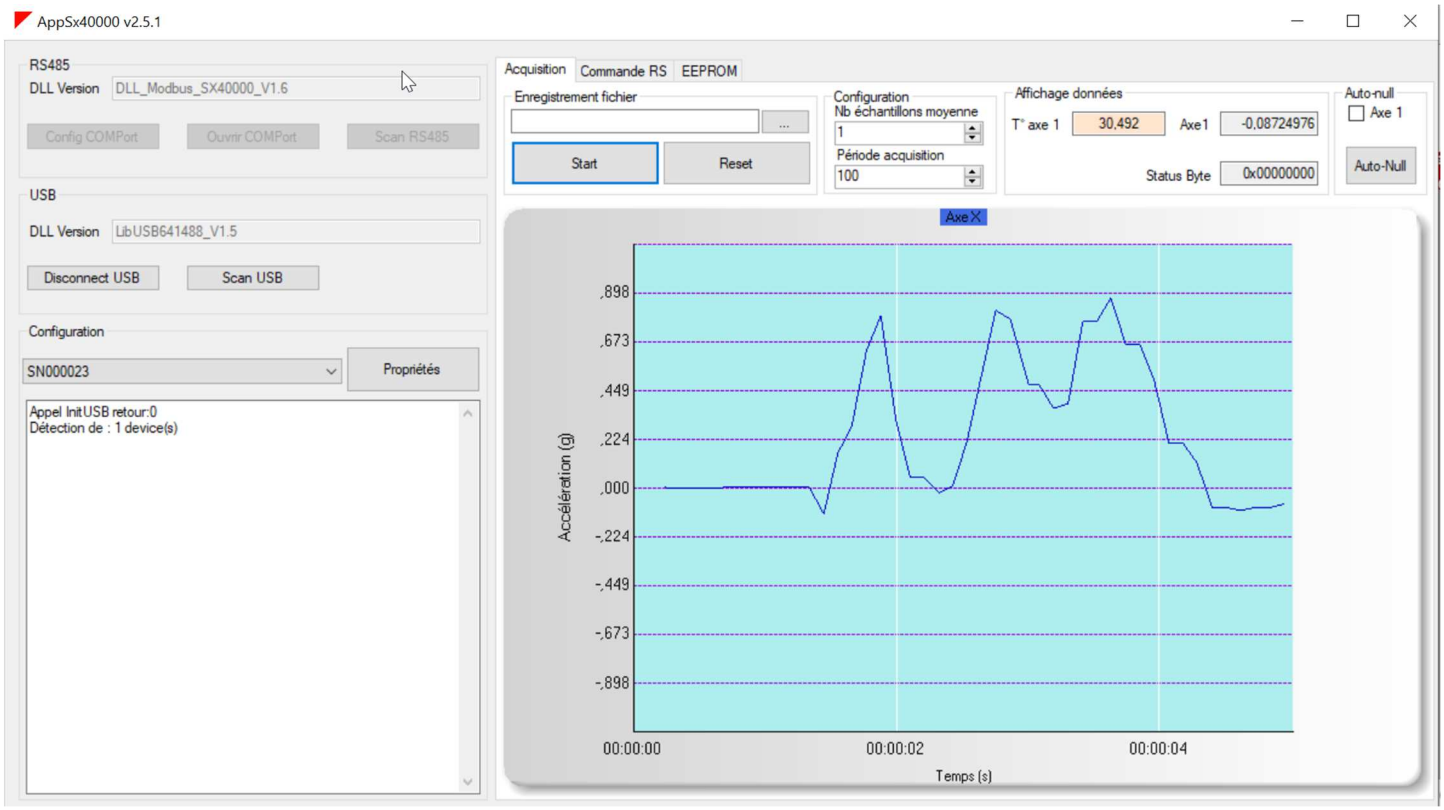
On the right side of the HMI is all the information returned by the SX40000 product.

The temperature, angle, or average acceleration information and status information of the SX40000 are displayed and updated cyclically.

In the "Display Data" section, you will find temperature values in ° C as well as average angles. The average of each angle / acceleration is calculated on the last N values displayed. N is equal to the value in the "Average samples" box.

The sliding display window represents at most the last 20 seconds of acquisition.

A second window « Mesure temps réel » allows user to monitor more easily sensor measure displayed in « Affichage des données » part in main window.



It is possible to zoom on the graph to better visualize the curves then to move the X and Y axes.

Graphic or acquisition file unit

It's possible to modify grah or acquisition file unit by double clicking on graph before pushing « start » button to start acquisition.

Following units are available :

- Inclinometer SX40000 : degree, percent, radians
- Accelerometer SX40000 : g or m/s²

Remote reset

It's possible to reset sensor using "Reset" button (instead of power OFF/ON)

It may be useful after a baud rate modification for example or other: So, it allows to not necessary operate a manual sensor power OFF/ON.

Data recording

If the user wishes to save the data from the SX40000 product, it is possible to specify a registration file.

The data acquisition period is taken from the value contained in the "Acquisition Period" box. This value is given in ms.

The acquisition file will contain:

- 1-byte counter
- The value of axis 1 and its unit
- The value of axis 2 and its unit
- The temperature of axis 1 in ° C
- The temperature of axis 2 in ° C
- The status of the SX40000
- The date of receipt of the sample in ms

Inclinometer auto-null configuration

The application AppSx40000 allows a self-calibration of the zero at a given inclination of the inclinometer / accelerometer permanently even after a restart of the product.

(Recommended) auto-calibration procedure for zero (or offset):

- Filter the sound of the sensor as much as possible by setting the product bandwidth to Fc = 0.2Hz (for example),
- Position the product at the desired auto-zero inclinations,
- Start an acquisition of the sensor,
- Wait for the stabilization of the sensor at least 5x1 / Fc is about 30seconds @ Fc = 0.2Hz,
- Perform the auto-zero of the sensor:
    - o Choose the auto-zero axis (s) by checking the "Axis 1" and / or "Axis 2" checkboxes,
    - o Click on the "Auto-Null" button

How to remove zero auto-calibration:

To restore the SX40000's blank auto-null settings before resetting the axes,

- Click on the "Properties" button and then "Restore factory settings".

- Then, in order not to lose your configuration (modbus ID or baud rate), just after having clicked on "restore factory setting", re-configure your MODBUS ID, your baud rate and your lowpass filter frequency desired and finally click on "Transfer Product Configuration" button.

Note: The "RS Command" and "EEPROM" tabs are reserved for internal MSS use

Dynamic and static data access

Various dynamic data (RAM) and static parameters (EEPROM) are accessible to the user via RS485 (MODBUS) or USB (HID class) interfaces.

If the user wishes to communicate with the product, without using the PC application, you will find below the description of low-level commands MODBUS or USB to achieve it.

RS485 interface (MODBUS)

The list of available MODBUS commands are as follows:

Access static parameters:

- Reading static parameters (EEPROM): Read holding register

- Writings of static parameters (EEPROM): Write multiple register

Dynamic Data Access:

- Reading dynamic data (RAM): Read input register

- FIFO buffer reading of the last 15 acceleration / tilt measurements: Read FIFO queue

Utility commands:

- Reset product command: Microcontroller reset

- Auto-null command: Autonull

- Restore Factory Settings Command: Restore Factory settings

Details of these orders are provided in Appendix 1.

USB interface (HID class)

The low-level USB commands correspond to those described in the USB standard "DEV Device Class Definition for Human Interface Devices (HID) Firmware Specification-6/27/01 Version 1.11" and more specifically:

- Get report request (reading)

- Set report request (writing)

The product has 2 main USB interfaces type HID:

- The interface 0 (MODBUS commands)

- Interface 1 (inclinometer data)

The reading / writing of the data of each interface can only be realized by using control requests (SETUP control transfer).

Note: The AppSx40000 software uses only the 0 interface to communicate with the product. This interface emulates a request / response format identical to the MODBUS RTU application protocol on the RS485 bus. It can read or write different dynamic data or static parameters via their respective addresses provided in this document.

Data library

The list of available static parameters is described in Appendix 2.

The list of available dynamic data is described in Annex 3.

Annex 1 : MODBUS RTU commands description

The set of MODBUS RTU commands used by the product are described below.

Note: All relevant data is transmitted MSB in the lead. That is, the 32-bit word following 0x01020304 will be transmitted in the following order:

0x01	0x02	0x03	0x04

Read holding registers command

This command is used to read static data / parameters (EEPROM) included in the product.

Request format:

Command	Start address	Registry quantity number	CRC
0x03	2 bytes	2 bytes	2 bytes

Champ	*Description*
Start address	EEPROM address to read (32bits aligned)
Registry quantity number	Number of register to read multiple of 2 obligatorily. (=N)

Valid response format (valid request):

Command	Byte count	Registers values	CRC
0x03	1 byte =2 x N (with N, qty of registers requested)	N * 2 bytes	2 bytes

Field	*Description*
Byte count	Total data byte count

Registers values	EEPROM registers requested values

Error response format (request error):

Command	Error code	CRC
0x83	1 byte	2 bytes

Champ	*Description*
Error code	0x01, 0x02, 0x03 ou 0x04

The error codes are given by the MODBUS RTU standard and have the following meanings:

Code d'erreur	*Signification*	*Description*
0x01	Illegal function	Unknown command
0x02	Illegal data address	Unknown addresses
0x03	Illegal data value	Illegal data value
0x04	Server device failure	Système failure

Read input registers command

This command is used to read RAM data contained in the product. The available addresses are described in chapter 0.

Request format:

Command	Start address	Registry quantity number	CRC
0x04	2 bytes	2 bytes	2 bytes

Champ	*Description*
Start address	RAM address to read (32bits aligned)
Registry quantity number	Number of register to read multiple of 2 obligatorily. (=N)

Valid response format (valid request):

Command	Byte count	Registers values	CRC
0x04	1 byte =2 x N (with N, qty of registers requested)	N * 2 bytes	2 bytes

Field	*Description*
Byte count	Total data byte count
Registers values	RAM registers requested values

Error response format (request error):

Command	Error code	CRC
0x84	1 byte	2 bytes

Champ	*Description*
Error code	0x01, 0x02, 0x03 ou 0x04

Write multiple registers command

This command is used to write EEPROM data into the product. The available addresses are described in chapter 0.

Request format:

Command	Start address	Qty of Registry	Byte number	data	CRC
0x10	2 bytes	2 bytes	1 byte	Register qty x 2 bytes	2 bytes

Champ	*Description*
Start address	EEPROM address to write (32bits aligned)
Register qty	Number of writing register multiple of 2 obligatorily. This field must be between 2 and 122.

Byte number	= 2 * « *Qty of register*»
Data	Data to write into EEPROM

Valid response format (valid request):

Command	Start address	Qty of registers	CRC
0x10	2 bytes	2 bytes	2 bytes

Field	*Description*
Start address	EEPROM address to write (32bits aligned)
Qty of register	Registers number written

Error response format (request error):

Command	Error code	CRC
0x90	1 octet	2 octets

Field	*Description*
Error code	0x01, 0x02, 0x03 ou 0x04

Read FIFO queue

This command is used to read FIFO data contained in the product. These data constitute the last N measurements of the inclinometer / accelerometer at the frequency of 1KHz. The data type (= desired axis) can be selected via the address passed in parameter. The returned data is all 32-bit floating type.

Request format:

Command	FIFO address	CRC
0x18	2 bytes	2 bytes

Field	*Description*

FIFO address	Address to select the type of data returned: • 0x2B00 (15 x last 32bit-float pitch tilt/acceleration data), • 0x2C00 (15 x last 32bit-float roll tilt/acceleration data),

Valid response format (valid request):

Command	Bytes count	FIFO registers count	FIFO registers values	CRC
0x18	2 bytes	2 bytes	« FIFO count » * 2 bytes	2 bytes

Champ	*Description*
Bytes count	Bytes count
FIFO registers count	16bits registers number (≤ 31) FIFO count register field included
FIFO registers values	32bit float FIFO data result.

Error response format (request error):

Command	Error code	CRC
0x98	1 octet	2 bytes

Field	*Description*
Error code	0x01 ou 0x02

Commande microcontroller reset

This command reboots the product without having to cycle Power Off / Power On.

The product issues a response before restarting. The product is inaccessible during its restart period.

Request format:

Command	CRC
0x41	2 bytes

Valid response format (valid request):

Commande	CRC
0x41	2 bytes

Error response format (request error):

Commande	Code d'erreur	CRC
0xC1	1 octet	2 bytes

Field	*Description*
Error code	0x01

Autonull command

This command adds a permanent offset to a sensor. This command is reversible by sending the Restore Factory Settings command. The offset will be equal to the value of the sensor at the moment the product receives the request.

Request format:

Command	Sensor number	CRC
0x44	1 octet	2 bytes

Champ	*Description*
Sensor number	• 1: axis 1 (pitch) • 2: axis 2 (roll)

Valid response format (valid request):

Command	CRC

0x44	2 bytes

Error response format (request error):

Command	Error code	CRC
0xC4	1 byte	2 bytes

Field	*Description*
Error code	0x01 ou 0x03

Commande Restore Factory Settings

This command resets the EEPROM values of the user part to the factory settings.

Request format :

Command	Key	CRC
0x46	8 bytes	2 bytes

Field	*Description*
Clé	Must be equal to : • 0xFF00FF00FF00FF00

Valid response format (valid request):

Command	CRC
0x46	2 bytes

Error response format (request error):

Commande	Error code	CRC
0xC6	1 byte	2 bytes

Field	*Description*
Error code	0x01 ou 0x03

Calcul du CRC16 MODBUS RTU

The MODBUS RTU standard includes an error check based on the Cyclical Redundancy Checking (CRC) method applied to the message data.
The CRC field is used to check all the data in the message. It is calculated without taking into account the parity bits of each byte of the message.
The CRC field contains a 16-bit word implemented as two 8-bit elements. The CRC 16bits is added at the end of the message. The low byte of the 16-bit CRC is first added to the message followed by the high byte. The high byte of the message is the last byte of the message. The 16-bit CRC is calculated by the sender of the message The receiver recalculates the 16-bit CRC of the received message and compares it with the value contained in the received message. If the two values are not identical, an error is thrown.

Calcul du CRC

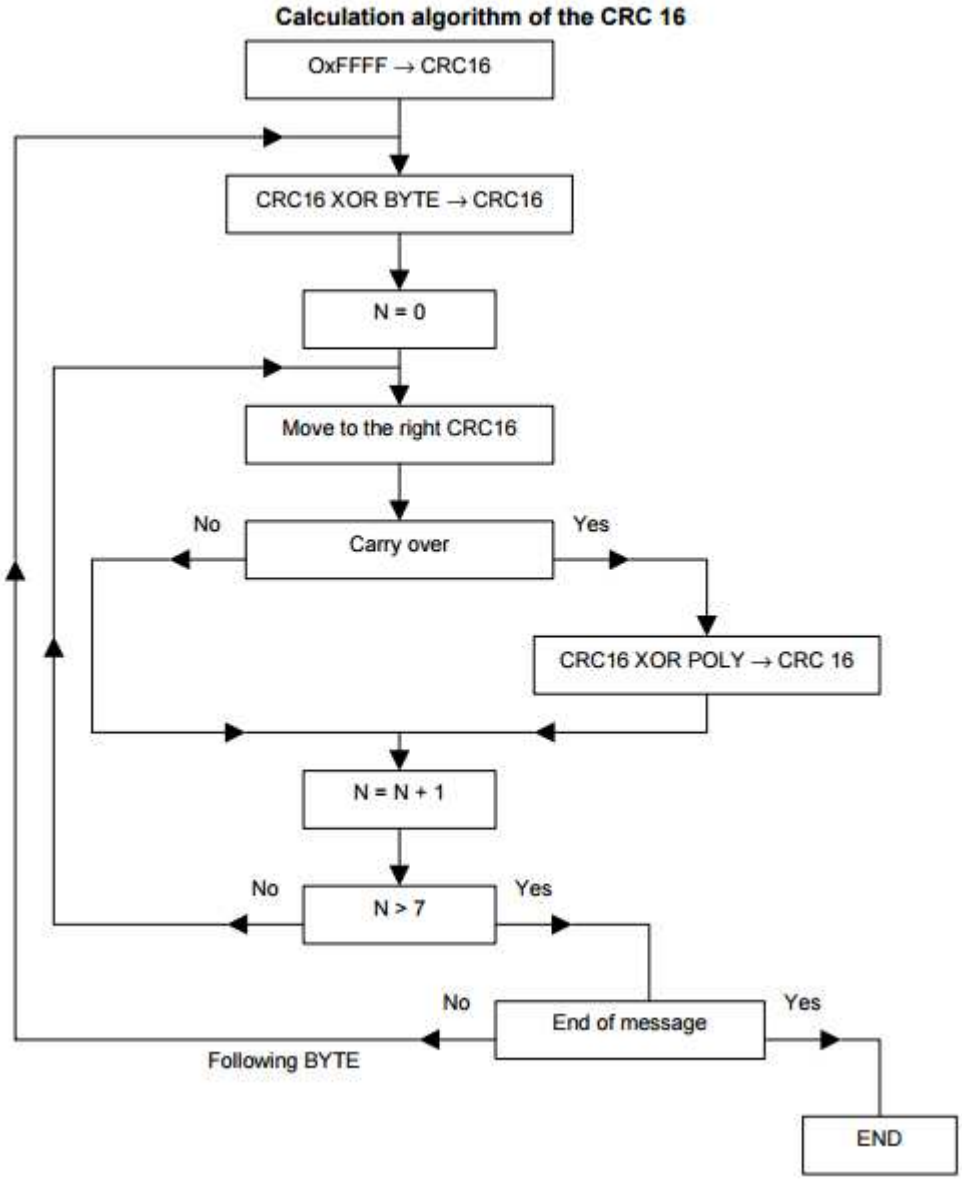
Le calcul du CRC est démarré en chargeant un registre 16 bits à 0xffff.
Ensuite, un processus démarre en appliquant successivement chaque octet du message à la valeur courante du registre. Seulement les 8 bits des données sont utilisées pour générer le CRC. Les bits de start, de stop et de parité ne sont pas appliqués au calcul du CRC.
Pendant la génération du CRC, on exécute un OU exclusif entre chaque octet du message et le contenu du registre de CRC. Ensuite le résultat est décalé vers la droite et le bit de poids fort est mis à 0. Le LSB est alors examiné. Si il est égal à 1, alors on applique un OU exclusif entre le registre du CRC et une valeur prédéfinie. Si le LSB est égal à 0, on ne fait rien.
Ce processus est répété jusqu'à ce que 8 décalages aient été effectués. Après les 8 décalages, on applique un OU exclusif sur le prochain octet avec le contenu du registre du CRC et le processus se continue pour 8 nouveau décalage comme décrit ci-dessus.
Le contenu final du registre, est la valeur du CRC 16 bits.
La procédure de génération du CRC 16 bits est la suivante:

1. Charger un registre 16–bit avec 0xFFFF. Ce registre est le *registre CRC*.
2. Appliquer un OU exclusive sur le premier octet du message avec l'octet de poids faible du *registre CRC*, et mettre le résultat dans le *registre CRC*.
3. Décaler le *registre CRC* à droite d'un bit et mettre un zéro dans le Most Significant Bit. Extraire et examiner le LSB.
4. (Si le LSB est égal à 0): Répéter l'étape 3 (autre décalage).
   (Si le LSB est égal à 1): Appliquer un OU exclusive entre le *registre CRC* et la valeur 0xA001 (1010 0000 0000 0001).
5. Répéter les étapes 3 et 4 jusqu'à ce que 8 décalages aient été effectués.
6. Répéter les étapes 2 à 5 pour le prochain octet du message et ce jusqu'à la fin du message.
7. Le contenu du *registre CRC* est la valeur du CRC 16 bits.
8. L'octet de poids faible du CRC 16bits est placé en premier dans le message et ensuite, l'octet de poids fort est inséré.

The calculation of the CRC is started by loading a 16-bit register at 0xffff.
Then, a process starts by successively applying each byte of the message to the current value of the register.
Only the 8 bits of the data are used to generate the CRC. The start, stop and parity bits are not applied to the calculation of the CRC.
During CRC generation, an exclusive OR is executed between each byte of the message and the contents of the CRC register. Then the result is shifted to the right and the most significant bit is set to 0. The LSB is then examined. If it is equal to 1, then an exclusive OR is applied between the CRC register and a predefined value. If the LSB is equal to 0, nothing is done.

This process is repeated until 8 offsets have been made. After the 8 offsets, an exclusive OR is applied to the next byte with the contents of the CRC register and the process is continued for further shifting as described above.
The final contents of the registry, is the value of the 16-bit CRC.
The procedure for generating 16-bit CRC is as follows:

1. Load a 16-bit register with 0xFFFF. This register is the CRC register.
2. Apply an exclusive OR on the first byte of the message with the low byte of the CRC register, and put the result in the CRC register.
3. Shift the CRC register to the right of a bit and put a zero in the Most Significant Bit. Extract and examine the LSB.
4. (If the LSB is 0): Repeat step 3 (other offset).
(If the LSB equals 1): Apply an exclusive OR between the CRC and 0xA001 (1010 0000 0000 0001).
5. Repeat steps 3 and 4 until 8 offsets have been made.
6. Repeat steps 2 to 5 for the next message byte until the end of the message.
7. The contents of the CRC register is the value of the 16-bit CRC.
8. The low byte of the 16-bit CRC is placed first in the message and then the most significant byte is inserted.



XOR = exclusive or
N = number of information bits
POLY = calculation polynomial of the CRC 16 = 1010 0000 0000 0001
(Generating polynomial = 1 + x2 + x 15 + x 16) In the CRC 16, the 1st byte transmitted is the least significant one.

Annexe 2 : RAM address description

Name	Description	MODBUS adresses (RAM addresse)	Type	Measurement range	Unit
FinalDigitalFormatOutput Axis1	Final output of acceleration or tilt X axis	0x1004	32bit float (IEEE-754)	[SortieMin, SortieMax]***	° / rad / g
TemperatureAxis1	X-axis sensor temperature	0x1088	16bit signed integer	[-351 ; +736] (⇔ [-40°C; +85°C])	LSB
FinalDigitalFormatOutput Axis2	Final output of acceleration or inclination Y axis	0x1104	32bit float (IEEE-754)	[SortieMin, SortieMax]***	° / rad /g
TemperatureAxis2	Y-axis sensor temperature	0x1188	16bit signed integer	[-351 ; +736] (⇔ [-40°C; +85°C])	LSB
SystemError	Error status of inclinometer (see description table 1)	0x1200	24bits unsigened word	/	LSB
UserParamCrc	Checksum data "User parameters bank" (cf DCI [AD6])	0x0910	16bit unsigned integer	/	LSB
ProductParamCrc	Checksum data "default product parameters bank" (cf DCI [AD6])	0x0920	16bit unsigned integer	/	LSB
CalibrationParamCrc	Checksum data "calibration parameters bank" (cf DCI [AD6])	0x0930	16bit unsigned integer	/	LSB
EepromRevision	Major version of compatible configuration files	0x0800	ASCII (8bits) x 17	/	LSB
SoftwareRevision	Embedded software version	0x0820	ASCII (8bits) x 17	/	LSB

32bit System error data description (*SystemError*):
By default,

Bit number	*Title*	*Description*
0	WdtFault	overflow defect watchdog (= 1)
1	BitOut	general sensor fault (= 1, at least one xxxxFault bit is thrown)
2	SysFault	system fault (oscillator, faulty UART, flash eeprom write error, SPI bus communication fault)
3	Sbit	Startup Built In Test in progress (= 1)
4	OverTemp	Exceeding operating temperature (= 1)
5	CalibMode	factory mode (= 1)
6	EepromUserFault	CRC of EEPROM area non-compliant user or at least one data out of ranges (= 1)
7	EepromProductFault	CRC of the EEPROM area produced non-compliant or at least one data out of ranges (= 1)
8	EepromCalibFault	CRC of the EEPROM area produced non-compliant or at least one data out of ranges (= 1)
9	TriAxisSbitFault	default start-up internal sensors (= 1)
10	Axis1SensorSbitFault	default start-up main sensor axis pitch (= 1)
11	Axis1AnalogSbitFault	default start-up output analog axis pitch (= 1)
12	Axis1OverRange	main sensor axis pitch out of ranges (= 1)
13	Axis1FilterFault	Digital filter of main sensor axis pitch unstable (= 1)
14	Axis1Autonull	Autonull of main sensor axis pitching enabled (= 1)
15	Axis1Uncalibrated	main sensor axis pitch not calibrated (= 1)
16	Axis2SensorSbitFault	default start-up main sensor roll axis (= 1)
17	Axis2AnalogSbitFault	default start-up analog output roll axis (= 1)
18	Axis1OverRange	main sensor roll axis out of ranges (= 1)
19	Axis1FilterFault	Digital filter of main sensor unstable roll axis (= 1)
20	Axis2Autonull	Autonull of main sensor roll axis enabled (= 1)
21	Axis2Uncalibrated	main sensor roll axis un-calibrated (= 1)

Tableau 2: Définition des bits de statut

Annexe 3 : EEPROM addresses description

Name	Definition	Range	Structure	Size	Default value	Units	(0xXXXX) EEPROM address (0x1D07XXXX)
EEPROM data (User parameter bank)							
UserParamCrc	Current bank checksum for data bank integrity checking purpose	/	16bit-word	1	/	LSB	0xB000
AutonullConfAxis1User	bias offset for inclinometer axis 1 autonull function **with unit depend on *TiltOutputUnitAxis1* data**	/	32bit-float	1	0.0	g / ° / rad	0xB004
FilterCoeffEnAxis1User	Axis 1 accelero digital low-pass filters coefficients for input samples defined by respectively Filter 1 (order O1= *FilterOrderAxis1User*): { *FilterCoeffEnAxis1User* [0], *FilterCoeffEnAxis1User* [1], *FilterCoeffEnAxis1User* [2],…, *FilterCoeffEnAxis1User* [N=O1]} With N=0 : coefficient for En N=1 : coefficient for En-1 N=2 : coefficient for En-2 … N=O1 : coefficient for En-O1 (*FilterCoeffEnAxis1User* [x] values are detailed in "Digital low-pass filter design" paragraph in [AD3])	/	64bit-float tab	N=7	/	/	0xB00C + (n x 8)h
FilterCoeffSnAxis1User	Axis 1 accelero digital low-pass	/	64bit-float	N=7	/	LSB	0xB044 +

Name	Definition	Range	Structure	Size	Default value	Units	(0xXXXX) EEPROM address (0x1D07XXXX)
	filters coefficients for ouput samples defined by respectively Filter 1 (order O1= *FilterOrderAxis1User*): { *FilterCoeffSnAxis1User* [0], *FilterCoeffSnAxis1User* [1], *FilterCoeffSnAxis1User* [2],…, *FilterCoeffSnAxis1User* [N=O1]} With N=0 : coefficient for Sn N=1 : coefficient for Sn-1 N=2 : coefficient for Sn-2 … N=O1 : coefficient for Sn-O1 (*FilterCoeffSnAxis1User* [x] values are detailed in "Digital low-pass filter design" paragraph in [AD3])		tab				(n x 8)h
FilterResonanceAxis1User	Resonance frequency gain (M) multiplied by max 23bit signed value (= M x 2^{22}) of inclinometer axis 1	/	64bit-float	1	/	LSB	0xB07C
FilterBandwidthAxis1User	Digital filter bandwidth @-3dB of inclinometer axis 1	≤ 10	32bit-float	1	/	LSB	0xB084
FilterOrderAxis1User	Digital filter order of inclinometer axis 1	≤ 6 and ≥ 0 (=0 means no filtering)	32bit-word	1	/	LSB	0xB088

Name	Definition	Range	Structure	Size	Default value	Units	(0xXXXX) EEPROM address (0x1D07XXXX)
AnalogSbitChoiceAxis1User	Factory default of analog output monitoring SBIT functionality choice of inclinometer axis 1 **(=0 means disable)**	[0; 1]	32bit-word	1	0	LSB	0xB08C
SensorSbitAxis1User	SCA103T SBIT enable/disable choice. (in order to be possible to obtain less than 1 second start-up time when SCA103T SBIT is disable)	[0; 1]	32bit-word	1	0	LSB	0xB090
AutonullConfAxis2User	bias offset for inclinometer axis 2 autonull function **with unit depend on *TiltOutputUnitAxis2* data**	/	32bit-float	1	0.0	g / ° / rad	0xB094
FilterCoeffEnAxis2User	Axis 2 accelero digital low-pass filters coefficients for input samples defined by respectively Filter (order O1= *FilterOrderAxis2User*): { *FilterCoeffEnAxis2User* [0], *FilterCoeffEnAxis2User* [1], *FilterCoeffEnAxis2User* [2],…, *FilterCoeffEnAxis2User* [N=O1]} With N=0 : coefficient for En N=1 : coefficient for En-1 N=2 : coefficient for En-2 … N=O1 : coefficient for En-O1 (*FilterCoeffEnAxis2User* [x] values are detailed in "Digital	/	64bit-float tab	N=7	/	/	0xB09C + (n x 8)h

Name	Definition	Range	Structure	Size	Default value	Units	(0xXXXX) EEPROM address (0x1D07XXXX)
	low-pass filter design" paragraph in [AD3])						
FilterCoeffSnAxis2User	Axis 2 accelero digital low-pass filters coefficients for ouput samples defined by respectively Filter (order O1= *FilterOrderAxis2User*): { *FilterCoeffSnAxis2User* [0], *FilterCoeffSnAxis2User* [1], *FilterCoeffSnAxis2User* [2],…, *FilterCoeffSnAxis2User* [N=O1]} With N=0 : coefficient for Sn N=1 : coefficient for Sn-1 N=2 : coefficient for Sn-2 … N=O1 : coefficient for Sn-O1 (*FilterCoeffSnAxis2User* [x] values are detailed in "Digital low-pass filter design" paragraph in [AD3])	/	64bit-float tab	N=7	/	LSB	0xB0D4 + (n x 8)h
FilterResonanceAxis2User	Resonance frequency gain (M) multiplied by max 23bit signed value (= M x 2^{22}) of inclinometer axis 2	/	64bit-float	1	/	LSB	0xB10C
FilterBandwidthAxis2User	Digital filter bandwidth @-3dB of inclinometer axis 2	≤ 10	32bit-float	1	/	LSB	0xB114
FilterOrderAxis2User	Digital filter order of inclinometer axis 2	≤ 6 and	32bit-word	1	/	LSB	0xB118

Name	Definition	Range	Structure	Size	Default value	Units	(0xXXXX) EEPROM address (0x1D07XXXX)
		≥ 0 (=0 means no filtering)					
AnalogSbitChoiceAxis2User	Factory default of analog output monitoring SBIT functionality choice of inclinometer axis 2 **(=0 means disable)**	[0; 1]	32bit-word	1	0	LSB	0xB11C
SensorSbitAxis2User	SCA103T SBIT enable/disable choice. (in order to be possible to obtain less than 1 second start-up time when SCA103T SBIT is disable)	[0; 1]	32bit-word	1	0	LSB	0xB120
Rs485BaudRateUser	UART baud rate selection	[19200; 115000]	32bit-word	1	19200	baud	0xB140
Rs485IdentifierUser	UART MODBUS identifier	[1;246]	Byte	1	1	LSB	0xB144
EEPROM data (Default product configuration parameter bank)							
ConfFileVersion	Configuration file version (major and minor version)	/	Character (byte) tab	24	/	LSB	0xC1A0
SxPartNumber	Part number (690641490vv) with version(=vv).	/	Character (byte) tab	12	690040140xy	LSB	0xC1C0
ProductDefinition	Brief description including range and unit	/	Character (byte) tab	60	690040140 INCLINOMETER 1 axe +/-70°	LSB	0xC1E0
WiredCiPartNumber	Part number of wired integrated circuit (490432xxx) without version	/	Character (byte) tab	12	490432xxx	LSB	0xC220
EEPROM data (Specific calibration parameter bank)							
SerialNumber	Serial number '(SNxxxxxx)	/	UNICODE characters	8	SNxxxxxx	LSB	0xD004

Name	Definition	Range	Structure	Size	Default value	Units	(0xXXXX) EEPROM address (0x1D07XXXX)
			(16bit-word) tab				
ManufacturingDate	Date of manufacturing "ddmmyyyy" (dd=day, mm=month, yyyy=year	/	Character (byte) tab	12	xx/xx/xxxx	LSB	0xD014
CalibrationDate	Last calibration date "ddmmyyyy" (dd=day, mm=month, yyyy=year	/	Character (byte) tab	12	xx/xx/xxxx	LSB	0xD024
WiredCiSerialNumber	Wired PCB serial number '(SNxxxxxxx)	/	Character (byte) tab	12	SNxxxxxxx	LSB	0xD034

Filtrage des capteurs

Each accelerometer and gyrometric sensor can be filtered independently of the others. For this the user has to calculate filter coefficients. The digital filtering performed by the product is of the following form:

$$S_n=A_nE_n+ A_{n-1}E_{n-1}+\ldots+A_0E_0 - (B_{n-1}S_{n-1}+B_{n-2}S_{n-2}+\ldots+B_0S_0)$$

The product supports filters whose cutoff frequency is less than 10Hz. When calculating the filter coefficients, it will be necessary to take into account the sampling frequency of the sensors at 1kHz.

After calculating the filter coefficients, the user must update the following values in EEPROM:

- FilterCoeffEnXXXXXUser []: filter coefficients of the In part. The index 0 of the table is the coefficient of the term In, the index 1 of the term En-1, etc.
- FilterCoeffSnXXXXXUser []: filter coefficients of the Sn part. The index 0 of the table is the coefficient of the term Sn, the index 1 of the term Sn-1, etc.
- FilterBandwidthXXXXXUser: Cutoff frequency of the filter in Hz. Must be less than or equal to 200Hz.
- FilterOrderXXXXXUser: Filter order. Must be less than or equal to 6.

Note: XXXXX represents the sensor whose cutoff frequency is to be changed. It is therefore equal to GyroX, or GyroY, or GyroZ, or AcceleroX or AcceleroY or AcceleroZ