



NextPS User Guide

Last update : 2022/10/19

Author : A. DUMAS

Version : 1.3



SUMMARY

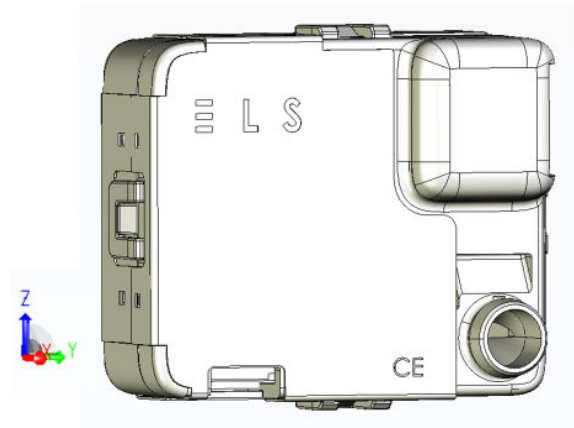
NextPS integration advices	3
NextPS communication's protocols	4
Simplified serial port communication	4
Configuration	4
PIN connector assignment	4
Serial port configuration	5
Send a command	5
NextPS responses	6
Command frame 0x31, 0x32 and 0x33	6
State code	8
Command frame 0x14	8
Command frame 0x15	9
Command frame 0x16	10
Command frame 0x17	10
Command frame 0x22	10
Modbus Protocol	11
Configuration	11
PIN connector assignment	11
Management Modbus functions	14
Particulate matter's data available	15
Advanced Modbus functions	17
Special Modbus function	18
Sample's test software	18



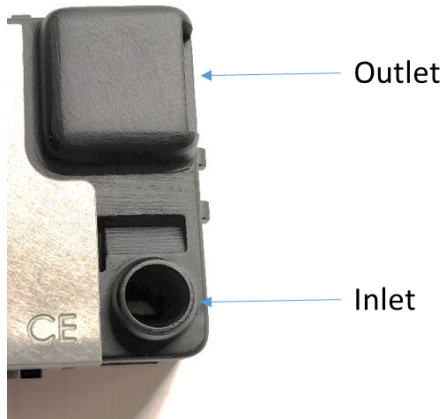
1. NextPS integration advices

To be fully functional, the NextPS must be integrated in order to be in a vertical way with the inlet down and the outlet up.

Plane	Angle tolerance's
XOZ	+/- 45°
YOZ	+/- 30°



You can set the NextPS sensor in another device thanks to the two holes in the back of the sensor. The holes are made to receive self turret screw M2.5 and are 5 mm deep.



The sensor use an active airflow to sample the particles. In order to keep his accuracy, we advice you to put the inlet directly to the air you want to monitor. If the integration is difficult, you can add a duct made with antistatic materials and with an internal diameter of 8 to 9 mm. The maximum length of the duct must be 100 cm.

Moreover, the inlet and the outlet need to be at a similar pressure and not obstructed.



The connector is a 6 PIN one. You can use an ADAM TECH 125CH-B-06 reference to connect with or an equivalent like a MOLEX 51021-0600 reference.



2. NextPS communication's protocols

2.1. Simplified serial port communication

2.1.1. Configuration

PIN connector assignment

1. GND
2. +5V
3. Tx (output)
4. Rx (input)
5. CS
6. GND

The serial communication is available thanks to PIN 3 and 4.



Serial port configuration

The serial port must be configured following these parameters :

- Speed : 115200 bauds,
- 8 bits,
- 1bit parity : even,
- 1 bit stop.

The NextPS reply to a request in more than 350 ms.

The NextPS must be power supplied with + 5 VDC.

The signal amplitude is +3.3V.

Warning 1 : to connect NextPS to a PC, prior, you must use a FTDI cable. One example of FTDI that could be used is : TTL-232R-3V3

Warning 2 : do not connect NextPS sensor directly on the RS232 port of a PC.



2.1.2. Send a command

The NextPS information's reading can be realized through the command frame : Address Command Checksum. To write information, use the command frame : Address Command DATA Checksum. The checksum is calculated in order that the sum of all the frame bytes is equal to a multiple of 256 (0x100).

Exemple :

$$0x81 + 0x16 + 0x69 = 0x100$$

$$0x81 + 0x22 + 0x03 + 0x5A = 0x100$$

Thus :

$$\text{Checksum} = 0x100 - \text{MOD}(\text{sum of the other bytes}, 256).$$

The address is freezed at 0x81.

Below, the command possibilities :

Cmd_Id	Description	Exemple
0x31	Concentrations reading's averaged over 10 seconds and updated every 1 second	0x81 0x31 0x4E
0x32	Concentrations reading's averaged over 60 seconds and updated every 10 seconds	0x81 0x32 0x4D
0x33	Concentrations reading's averaged over 900 seconds and updated every 60 seconds	0x81 0x33 0x4C
0x14	Temperature and humidity readings	0x81 0x14 0x6B
0x15	Power on or sleep mode	0x81 0x15 0x6A (cf note 1)
0x16	Sensor state's readings	0x81 0x16 0x69
0x17	Firmware version readings	0x81 0x17 0x68
0x22	Modbus address read Modbus address 3 write	0x81 0x22 0x00 0x5D 0x81 0x22 0x03 0x5A (cf note 2)

Note 1 : Each 0x15 command frame send, the NextPS change its functional state alternately. To know its state before to send the command frame, you can send 0x16 command frame : if the 0 bit of the code state is 1, the NextPS is in sleep mode.

Note 2 : The 0x22 command is used to read the Modbus address (the third byte must be 0) or to write the new Modbus address (the third byte is this address, 3 in this example).



2.1.3. NextPS responses

The NextPS replies to a command frame by a frame that always begins by its address (0x81) following by the command frame asked and ending by a checksum.

The datas sent are function of the command frame :

Command frame 0x31, 0x32 and 0x33

address	Cmd id	State (1 byte)	PM >0.3µm pcs/m3 (4 bytes)	PM >0.5µm pcs/m3 (4 bytes)	PM >1µm pcs/m3 (4 bytes)	PM >2.5µm pcs/m3 (4 bytes)	PM >5µm pcs/m3 (4 bytes)	Checksum
0x81	0x31	0x00	0x00A86CE9	0x000BE42D	0x00026B9C	0x00000000	0x00000000	0x2C
0x81	0x32	0x00	0x0370F1EF	0x0007BAEF	0x0002349B	0x00003044	0x00003044	0x91
0x81	0x33	0x00	0x02DCAE93	0x00091866	0x0001DC4D	0x00005071	0x00005071	0xFA
0x81	0x16	0x04						0x65

The particulate matter concentration in pcs/m3 are coded with 4 bytes (32 bits).

For example :

The response 0x81 0x32 0x00 0x03 0x70 0xF1 0xEF 0x00 0x07 0xBA 0xEF 0x00 0x02 0x34 0x9B 0x00 0x00 0x30 0x44 0x00 0x00 0x30 0x44 0x91

Signifies that the results are averaged over 1 minute. No error occurred during the measurement and the state code is 0.



In the example below, the measured concentrations are :

	4 bytes datas	Coef	Results
PM >0.3µm (pcs/m ³)	0x0370F1EF	1	57 733 615 pcs/m ³
PM >0.5µm (pcs/m ³)	0x0007BAEF	1	506 607 pcs/m ³
PM >1µm (pcs/m ³)	0x0002349B	1	144 539 pcs/m ³
PM >2.5µm (pcs/m ³)	0x00003044	1	12 356 pcs/m ³
PM >5µm (pcs/m ³)	0x00003044	1	12 356 pcs/m ³

The concentrations are calculated from the 4 bytes read data and multiplied by a coefficient (Coef).

The state code must always be read, it highlights the functional state of the NextPS and allows to know the validity of the sent values.

The checksum can be checked, the sum of all the frame bytes is equal to a multiple of 256 (0x100), here :

$0x81 + 0x32 + 0x00 + 0x03 + 0x70 + 0xF1 + 0xEF + 0x00 + 0x07 + 0xBA + 0xEF + 0x00 + 0x02 + 0x34 + 0x9B + 0x00 + 0x00 + 0x30 + 0x44 + 0x00 + 0x00 + 0x30 + 0x44 + 0x91 = 0x700$ is a multiple of 0x100, le checksum is OK.

State code

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Laser Error	Memory Error	Fan Error	T/RH Error	Heat Error	Not Ready	Degraded State	Sleep State

The bit 0 is set to 1 when the sensor is set to sleep state : the laser, the fan and the heat are switch off. The only command frame still possible is 0x16 (Read the NextPS State), the NextPS will respond to any other command frame as if it is the 0x16 command frame.

The bit 1 is set to 1 each time a minor error is detected, the sensor part in error is set to 1 in the state code, the NextPS can still send datas but with less accuracy.



The minor errors are the following :

- Heat Error, the relative humidity stay above 60% during more than 10 minutes,
- T/RH Error, the sensor reading are out of specification,
- Fan Error, the fan speed is out of range but the fan is still working.
- Memory Error, the sensor can't access its memory, some internal smart functions will not be available.

The sensor will be set to Default State if the fan or the laser are broken. If the sensor is in the Default State, the degraded state flag is set to 0, the default part is indicated by its flag set to 1 and the bit 0 is set, the sensor is thus in sleep state.

If the sensor replies by a 0x16 command frame, it means that the NextPS has no data to send neither because the sensor has just been switch on nor because the sensor is in the Default State or Sleep State.

Command frame 0x14

address	Command id	State (1 byte)	Temperature (2 bytes)	Humidity (2 bytes)	Checksum
0x81	0x14	0x00	0x0B40	0x13E7	0x26
0x81	0x16	0x76			0xF3

The temperature and relative humidity are sent with 2 bytes, you need to divide by 100 the obtained value in order to have the real value.

Example: the NextPS replies 0x0B40, thus 2880 for a 28.80 °C temperature and 0x13E7, thus 5095 for a 50.95% relative humidity.

If the sensor replies by a 0x16 command frame, it means that the NextPS has no data to send, neither because the NextPS has just been switch on, nor because the NextPS is in the T/RH Error State or Sleep State.



Command frame 0x15

address	Command id	State (1 byte)	Checksum
0x81	0x15	0x01	0x69
0x81	0x16	0x01	0x68

When the command frame 0x15 is sent to set the sensor in sleep mode, the NextPS replies with a state code with the flag SLEEP set to 1 and stop to work.

During the SLEEP mode, if the sensor receives a new 0x15 command frame, then the NextPS will be switch on and will send the first PM datas after 15 seconds.

During the SLEEP mode, the NextPS reply to all the other command frame by sending 0x16 command frame with the state code.

If the NextPS is in the Default State, the 0x15 command frame allowed to try to switch on the NextPS.

In SLEEP mode, the sensor current consumption is about 20mA.

Command frame 0x16

address	Command id	State (1 byte)	Checksum
0x81	0x16	0x33	0x36

The 0x16 command frame send the State code. It's the default response when there is no data to transmit or when the asked command frame can't be done (for example, when the sensor is in SLEEP mode).



Command frame 0x17

address	Command id	State (1 byte)	Firmware Version (2 bytes)	Checksum
0x81	0x17	0x00	0x0034	0x34
0x81	0x16	0x01		0x68

The NextPS will send the firmware version. In the above example, the firmware version is 0x0034. If the NextPS is in the SLEEP mode, it will only reply its state code.

Command frame 0x22

address	Command id	State (1 byte)	Modbus Address (1 byte)	Checksum
0x81	0x22	0x00	0x03	0x5A
0x81	0x16	0x01		0x68

The NextPS will send the Modbus address. In the above example, the Modbus Address is 0x03. If the NextPS is in the SLEEP mode, it will only reply its state code.



2.2. Modbus Protocol

2.2.1. Configuration

PIN connector assignment

1. GND
2. +5V
3. Tx (output)
4. Rx (input)
5. CS
6. GND

The serial communication is available thanks to PIN 3 and 4.

The communication is a Modbus RTU (binary):

- Speed : 115200 bauds,
- 8 bits,
- 1bit de parity, even,
- 1 bit stop.

The NextPS responds to a request in more than 350 ms.

The NextPS is powered supply by + 5VDC.

The signals of the communication have an amplitude of + 3.3V.

The Modbus protocol complies with V1.1b specifications.





The frame for reading registers in the NextPS is in the following format :

A0 C1 R1 R2 N1 N2 V1 V2

Example:

01 03 00 01 00 01 D5 CA (in hexadecimal) reads the firmware version

- The A0 address of the NextPS module is 0x01.
- C1 is the read command: 0x03
- The address of the start register is coded on 2 bytes R1R2 (R1 most significant) 0x00 0x01.
- The number of register is coded on 2 bytes N1N2 (N1 most significant) 0x00 0x01.
- The frame ends with two bytes of checksums V1V2 0xD5 0xCA:

Polynomial : 0xA001 // Polynomial = $2^{15} + 2^{13} + 2^0 = 0xA001$

INIT CRC : 0xFFFF

Example of checksum calculation :

Private Function CRC16(buf() As Byte, lbuf As Integer) As Integer

' returns the MODBUS CRC of the lbuf first bytes of "buf" buffer (buf is a global array of bytes)

*Dim Crc As Integer
Dim mask As Integer
Dim i As Integer
Dim j As Integer*

*Crc = &HFFFF ' init CRC
For i = 0 To lbuf - 1 Step 1 ' for each byte
 Crc = Crc Xor buff(i)

 For j = 0 To 7 Step 1 ' for each bit

 mask = 0
 If Crc / 2 <> Int(Crc / 2) Then mask = &HA001
 Crc = Int(Crc / 2) And &H7FFF: Crc = Crc Xor mask

 Next j
Next i*

CRC16 = Crc

End Function



V1 = CRC16 AND 0xFF

V2 = INT (CRC16 / 256) AND 255

The NextPS must answer:

A1 C1 N1 R1 ... Rn V1 V2

01 03 02 00 42 38 75 (in hexadecimal)

- A1 is the address of the NextPS: 0x01
- C1 is the callback of the read command: 0x03
- N1 is the number of bytes transmitted, coded on one byte
- R1 ... Rn correspond to the data (bytes) transmitted, in the example, 0x00 0x42, firmware 4.2
- V1 and V2 is the checksum calculated as before.

Another example :

We want to read 10 16-bit registers from register 1 :

01 03 00 01 00 0A 94 0D

The NextPS must answer:

01 03 14 00 42 00 83 00 01 33 3C 00 00 D9 F0 00 00 01 43 00 04 00 00 29 5A

It sends 2 * 10 bytes (0x14).

Beware, depending on the configuration of the sensor, the registers described below are not always accessible.



2.2.2. Management Modbus functions

The following registers are only readable.

A register is coded on 16 bits.

List of decimal's registers :

Register	Name	description
1	FIRMWARE_VERSION	Embedded software version

The operating status or error code is displayed in register 19.

19	OPC_STATUS	Error code or status of NextPS
----	------------	--------------------------------

PM_STATUS consists of 9 bits :

Bit 7 R	Bit 6 R	Bit 5 R	Bit 4 R	Bit 3 R	Bit 2 R	Bit 1 R	Bit 0 R/W
Laser error	Memory error	Fan error	T°C/Hum Error	Heat error	Not Ready	Degraded mode	Mode sleep

Bit 0 is set to 1 when the sensor is in standby: the laser, the fan, the heating are then deactivated. Note that it is not possible to switch the sensor to sleep mode with VbModbus.

Bit 1 is set to 1 when a minor error is detected, the status of the erroneous element goes to 1 in the status code, the sensor can still send data.

Minor errors are the followings :

- memory error, the memory is no longer accessible, there is no more correction of the aging of the sensor, the configuration parameters are those originally programmed at the factory during configuration.
- heating error, the humidity level stays above 60% for more than 10 minutes,
- temperature and humidity measurements error, their reading is not in the specifications,
- Fan error, its speed is not in the specified operating range but the fan is not blocked.

There is no degraded mode in the event of a laser fault or in the event of a fan breakdown. In both cases, the sensor switches to fault mode after 3 start attempts, bit 9 goes to 1. In fault mode, the degraded mode flag is at 0, the faulty element is indicated by its flag at 1 and bit 0 indicates that the sensor is at a standstill.



2.2.3. Particulate matter's data available

20-21	GetTenSecondesAverageResult.Nbsup_0_3	<u>average over 10s</u> of the number of particles / m ³ of size > 0.3 μm
22-23	GetTenSecondesAverageResult.Nbsup_0_5	<u>average over 10s</u> of the number of particles / m ³ of size > 0.5 μm
24-25	GetTenSecondesAverageResult.Nbsup_1_0	<u>average over 10s</u> of the number of particles / m ³ of size > 1.0 μm
26-27	GetTenSecondesAverageResult.Nbsup_2_5	<u>average over 10s</u> of the number of particles / m ³ of size > 2.5 μm
28-29	GetTenSecondesAverageResult.Nbsup_5_0	<u>average over 10s</u> of the number of particles / m ³ of size > 5.0 μm
30-31	GetOneMinuteAverageResult.Nbsup_0_3	<u>average over 60s</u> of the number of particles / m ³ of size > 0.3 μm
32-33	GetOneMinuteAverageResult.Nbsup_0_5	<u>average over 60s</u> of the number of particles / m ³ of size > 0.5 μm
34-35	GetOneMinuteAverageResult.Nbsup_1_0	<u>average over 60s</u> of the number of particles / m ³ of size > 1.0 μm
36-37	GetOneMinuteAverageResult.Nbsup_2_5	<u>average over 60s</u> of the number of particles / m ³ of size > 2.5 μm
38-39	GetOneMinuteAverageResult.Nbsup_5_0	<u>average over 60s</u> of the number of particles / m ³ of size > 5.0 μm
40-41	GetFifteenMinuteAverageResult.Nbsup_0_3	<u>average over 15 min</u> of the number of particles / m ³ of size > 0.3 μm
42-43	GetFifteenMinuteAverageResult.Nbsup_0_5	<u>average over 15 min</u> of the number of particles / m ³ of size > 0.5 μm
44-45	GetFifteenMinuteAverageResult.Nbsup_1_0	<u>average over 15 min</u> of the number of particles / m ³ of size > 1.0 μm
46-47	GetFifteenMinuteAverageResult.Nbsup_2_5	<u>average over 15 min</u> of the number of particles / m ³ of size > 2.5 μm
48-49	GetFifteenMinuteAverageResult.Nbsup_5_0	<u>average over 15 min</u> of the number of particles / m ³ of size > 5.0 μm



The average are coded on 32 bit, thus on 2 registers. The actual value of the average is obtained by dividing by 1000 the value in the two registers (see example below).

Example:

Interrogation of all average concentration:

The data address is 0x20 and coded in 2 byte for particle matter of size >0.3µm

Send command:

01 03 00 20 00 24 44 1B

Sensor's reply:

01 03 48 75 AE 00 21 FE 37 00 08 39 3F 00 00 39 3F 00 00 D1 2B 00 41 AB 41 00 1F B3 9B 00 09 18 8B 00 01 5B 98 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 64 00 00 09 82 00 00 09 82 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B5 FE

Calculation of the 10s average of the particle concentration / m3 of size > 0.3 µm:

This is the first 32 bit coded data:

- 0x75AE(Less significant Byte), 0x0021(Most significant Byte),
- Result: 0x002175AE,
- 2192814 in decimal
- Thus, the final result of PM1 is 2192814 pcs/m³ or 2192,814 pcs/L for the average of 10s

The screenshot shows the 'Simply Modbus Master 8.1.2' interface. The 'Request' field contains the hex string '01 03 00 20 00 24 44 1B'. The 'Response' field contains the hex string '01 03 48 75 AE 00 21 FE 37 00 08 39 3F 00 00 39 3F 00 00 D1 2B 00 41 AB 41 00 1F B3 9B 00 09 18 8B 00 01 5B 98 00 00 00 00 00 00 00 00 00 00 00 00 00 06 64 00 00 09 82 00 00 09 82 00 00 00 00 00 00 00 00 00 00 00 00 B5 FE'. The 'Results' table shows the following data:

register#	bytes	results
40037	75AE	30126
40038	0021	33
40039	FE37	-457
40040	0008	8
40041	393F	14655
40042	0000	0
40043	393F	14655
40044	0000	0
40045	D12B	-11989
40046	0041	65
40047	AB41	-21695
40048	001F	31
40049	B39B	-19557
40050	0009	9
40051	188B	6283



2.2.4. Advanced Modbus functions

The following registers are writable and accessible with a write or read/write multiple holding registers function command (0x10 or 0x17)

A register is coded on 16 bits.

88	ModBusAddress	NextPS with Modbus protocol
----	---------------	-----------------------------

Currently, the NextPS address can be set in the range 1 to 15.

89	FanTarget	Fan rotation setpoint expressed in Hertz
----	-----------	--

The setpoint can vary from 30Hz to 400Hz but we strongly recommend to not touch it.

101	HeaterPWM	Cyclic ratio for heating (expressed in %)
-----	-----------	---

The duty cycle varies from 0.00 to 1.00 (where 0.00 is 0% and 1.00 is 100%), it is stored in the register multiplied by 10000.

It will strongly increase the current consumption when turned on.

104	Sleep mode	sleep mode data
-----	------------	-----------------

When this command whose data is equal to 1 is sent, the sensor is put in sleep mode. The sensor continues to reply to requests but its measurements are frozen due to its state.

When this command whose data is equal to 0 is sent, the sensor awakes and relaunches its measurements. The measured concentrations will be then available.

If the data is different from 0 or 1, the sensor will send back an error message.

Note 1: a way to see if the sensor is in sleep mode is to ask for its status.

Note 2: this function is available from the 0x1121 firmware version.

112	Baud rate	Identifier code
-----	-----------	-----------------

The baud rate of the simplified and modbus communication can be changed by writing the identifier code corresponding to the desired baud rate value (see table below).



Identifier code	Baud rate
1	115200
3	38400
4	19200
5	9600
7	2400
8	1200

Note: 57600 baud rate through identifier code 2 and 4800 baud rate through identifier 6 could also be used but it may be necessary to send several requests to get information from NextPS sensor at one of these two baud rates.

Special care should be taken when baud rate is changed, the user should mark or remember the new baud rate associated with the NextPS because if it is forgotten by the user, there is no way to find it except by testing all the baud rate values up to match the baud rate previously programmed.

If the identifier code in the command does not belong from 1 to 8, the sensor will send back an error message.

Note: this function is available from the 0x1121 firmware version.

113	Parity mode	Identifier code
-----	-------------	-----------------

The parity mode of the simplified and modbus communication can be changed by writing the identifier code corresponding to the desired parity mode (see table below).



Identifier code	Parity mode
1	EVEN
2	ODD
3	NONE

Special care should be taken when parity mode is changed, the user should mark or remember the new parity mode

associated with the NextPS because if it is forgotten by the user, there is no way to find it except by testing the parity modes up to match the parity mode previously programmed.

If the identifier code in the command does not belong from 1 to 3, the sensor will send back an error message.

Note: this function is available from the 0x1121 firmware version.

2.2.5. Special Modbus function

The following register is only readable and accessible with a read or read/write multiple holding registers function command (0x03 or 0x17)

The registers 4 (LSB) and 5 (MSB) return the cumulative operating time of the sensor.

4 - 5	TimeOn	in seconds
-------	--------	------------

The register 102 returns the frequency of rotation of the fan if it is available.

102	FanSpeed	Frequency expressed in Hertz
-----	----------	------------------------------

The registers 106 and 107 allow you to know the humidity and the temperature, the values are stored in a register (of 16 bits), it is the real value multiplied by 100.

106	Humidity	Humidity in % multiplied by 100
-----	----------	---------------------------------



107	Temperature	Temperature in °C multiplied by 100
-----	-------------	-------------------------------------

Beware, these data are no more usable if you have activated the heating function.

3. Sample's test software

If you want to test the sensor before to work on the integration, you could find two softwares on our website in order to communicate with the NextPS using a computer running with Windows 10.

Tera Sensor website : <http://www.tera-sensor.com/>

Download the softwares from : [Resources/Software](#)