User Manual

Hubit HY-SA6E Weather Station

Overview

The HY-SA6E is an all-in-one weather station designed for comprehensive outdoor environmental monitoring. Using ultrasonic technology, it measures wind speed and direction alongside parameters such as temperature, humidity, pressure, precipitation, solar radiation, and luminance, depending on the configuration.

The weather station supports RS232, RS485, or SDI-12 communication and can operate using protocols such as Modbus RTU, NMEA 0183, and SDI-12. Its rugged, UV-resistant housing and maintenance-free design make it well suited to long-term outdoor monitoring applications.

Key Features

All-in-One Monitoring

Measures multiple environmental parameters from a single device.

Ultrasonic Wind Measurement

Provides accurate wind speed and direction without moving parts.

Multiple Interfaces

Supports RS232, RS485, and SDI-12

Rugged Outdoor Design

UV-resistant ASA housing for long-outdoor use.

Multi-Protocol Support

Compatible with Modbus RTU, NMEA 0183, and SDI-12.

Calibration Accuracy

Each unit is calibrated in a wind tunnel before shipment.

Document Version
v1.0

17/08/2026

Table of Contents

1. Introduction

This User Manual provides necessary information for user to install and communicate with HY-SA6E weather station.

1.1 Overview

HY-SA6E weather station is very robust with no moving parts, and maintenance free. It can simultaneously output wind speed and direction.

Each unit is calibrated in our wind tunnel testing lab prior to shipment.

Typically, HY-SA6E can be powered up with 5-30 VDC (4-20mA output requires 12~24VDC).

Please note that some specially customised sensors have narrower power supply range.

It can be communicated with by one of following communication protocols: SDI-12, MODBUS RTU and NMEA 0183.
Three alternative serial interfaces are selectable: RS232, RS485 and SDI-12.
Communication interface is decided by hardware and can't be changed by software, please specify requirements when you place order.

The device is typically equipped with a 4 pins M16 connector for installation.(4-20mA output is optional with extra cost).
HY-SA6E weather station measures the airflow induced phase changes of an acoustic resonance, four ultrasonic probes take turns to excite an acoustic resonance and inspect acoustic phase change, thereafter wind speed and direction is detected.

The exterior housing is made by ASA engineered plastic, which has high outdoor weather ability. ASA is extremely resistant against UV radiation of sun, it is frost and heat resistant, and can withstand all climatic conditions for over 10 years.

HY-SA6E can realize simultaneous measurement of multi- parameters: wind, temperature, humidity, pressure and precipitation, solar, luminance, Noise, PMS, visibility. Precipitation can be detected by 24G radar or piezoelectric module.

TECHNICAL SPECIFICATION

Signal Output	RS232 or RS485 or SDI-12			
Power Supply	DC：7-24V			
Data Output	1 per second			
Power Consumption	Wind module:10mA@12V Radar rain module consume extra: 110mA@12VDC			
Material of Body	UV-resistant ASA engineering plastic			
Communication Protocol	MODBUS RTU or NMEA-0183 or SDI-12			
Dimension	Ø105 * 219 mm			
Parameters	Principle	Range	Accuracy	Resolution
Wind Speed	Ultrasonic	0~70m/s	±3%	0.01
Wind Direction	Ultrasonic	0~359°	<3°	1°
Air Temperature	MEMS	-40~60℃	±0.5℃	0.1℃
Air Humidity	MEMS	0~100%	±2％	0.1
Air Pressure	MEMS	150~1100hPa	±1 hPa	0.1hPa
Precipitation (Radar module can identify Rain/Hail/Snow)	Radar	Rain:0~500mm/hr	±10%	0.01mm
	Piezoelectric	Rain:0~500mm/hr	±10%	0.01mm
Luminance	Silicon	0~20000 lux	±5%	1 Lux
Solar Radiation	Silicon	0~2000 W/m2	±5%	1 W/m2
PM1.0/PM2.5/PM10	Laser scattering	0~500ug/m3	±10%	1 ug/m3
Visibility	Laser scattering	10~10000 meters	±20%	1 meter
Noise	microphone	30~130 dB	±3dB	0.01 dB

2. Getting Started

2.1 Wiring

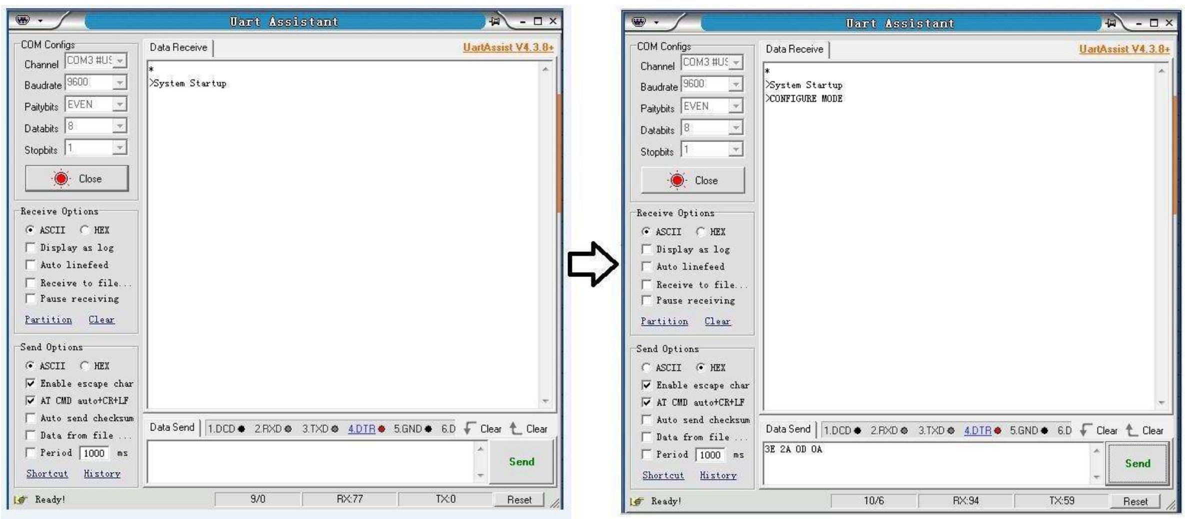
Color of cable		Red	Black	Yellow	Green
Definition	RS485	Power+	Power- & RS232 GND	DA+	DB-
	RS232			RXD	TXD
	SDI-12			-	SDI-12

2.2 Communication test

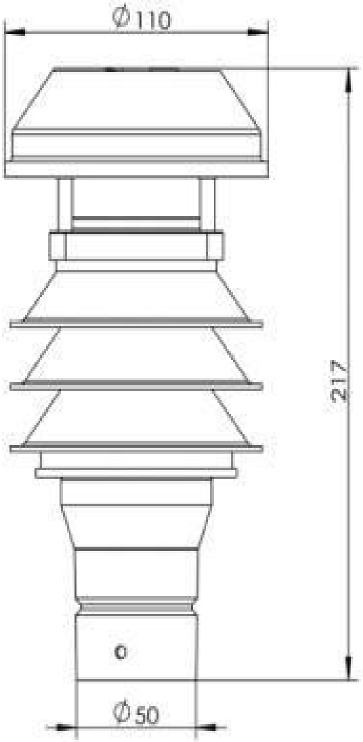
Note: We do not recommend the user installs sensors or connects to their data logger before testing communication through the procedures below. We highly recommend you to use corresponding communication interface to USB converter to test our sensor on the computer before conducting any further operation.

Our sensor will send the ASCII string “>System Startup” once you power it up.

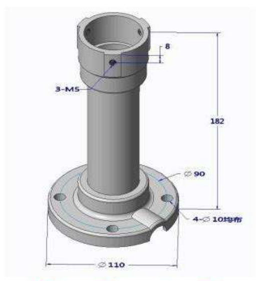
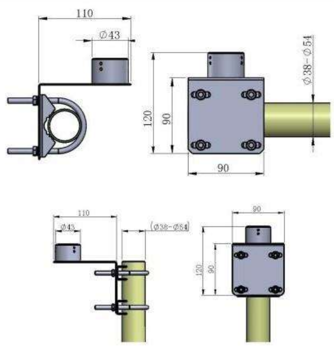
Thereafter, you should receive “CONFIGURE MODE” after sending “>*\r\n” to it. (or send in HEX: 3E2A0D0A).
You should then be able to confirm communication with the sensor has been established.



2.3 Mounting



Note: We also provide following accessories:

 Material: Aluminium	 Material: Cast Aluminium	 Material: SUS304
Tripod Installation Kit (Note: Tripod not included)	**Mounting Mast**	**Mounting Bracket**

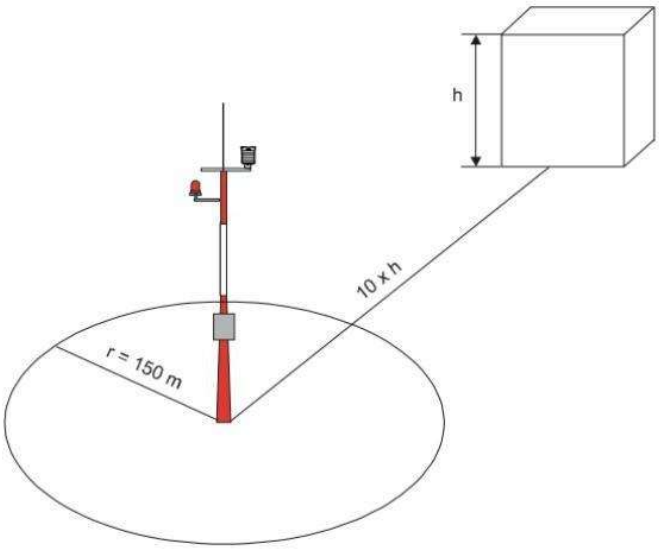
Installation Location

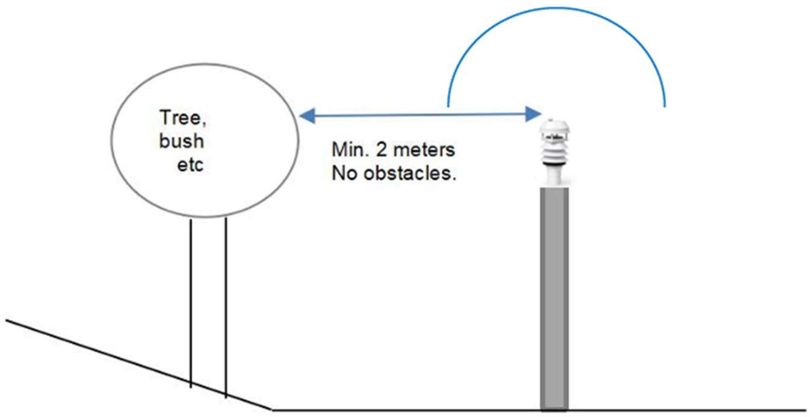
- The mounting location must not be sheltered from the wind or in the lee of objects such as chimneys or satellite systems.
- The anemometer must be mounted in an upright position; otherwise, water can enter the anemometer and destroy it.
- If possible, install the anemometer in the center of flat roofs. Installing the anemometer at the edge of the roof may cause air turbulence that may distort the measuring results.
- Default length of cable with the anemometer is 4 meters. This cable may be cut shorter or extended to a maximum of 1000 meters (for RS485) or 10 meters (for RS232).

Following World Meteorological Organization (WMO) guidelines, a general recommendation is that there is at least 150m open area in all directions from the mast.
Any object of height (h) does not significantly disturb wind measurement at a minimum distance of 10 times the height of the object.

The recommended minimum length (h) for the mast that is installed on top of a building is 1.5 times the height of the building (H). When the diagonal (W) is less than the height (h), the minimum length of the mast is 1.5W.

Please follow the application specific instructions and local regulations when placing the anemometer.





The radar module is under the top cover. You can wave your palm up and down for 20 seconds above top cover to simulate radar wave echoed by rain.

Please note that the radar rain module won't be accurate indoor and minimum height of falling distance should be over 15 meters to get naturally falling raindrops.

If your sensor comes with a radar rain module ensure there are no obstacles within upper semi-sphere within a 2 meter diameter.

3. Communications with sensor

Note: Our sensor has three kinds of digital interface:

RS485, RS232, SDI-12 for option.

Analog interface: 4-20mA, 0-5VDC for option.

Please advise on the above requirements when you place your order. They cannot be changed after production.

Only the advanced SA6E anemometer comes with averaged wind and gust, and there is surcharge. Ensure this is mentioned when you place an order.

Additionally, also note that the following protocols are for reference only, and you cannot fetch reading for parameters that you didn't order.
For regular sensor, please access register from 1~49th.

For advanced sensor, you can access register from 1~96th. Sensor will respond with error code if you use regular sensor and request data from 50~96th registers.

3.1 MODBUS RTU protocol

Unless you request for other configuration, the sensor will adopt the following default configuration:
Baud Rate: 9600 bps;
Parity: NONE;
Data Bits: 8 bits;
Stop Bits: 1 bit;
Data structure of MODBUS command and response as below:

Device Address	Function Code	Data	CRC checksum

MODBUS protocol is based on time-sharing poll mechanism. It is essential to assign a unique device address to each sensor on the same communication bus line.
Function Code: We use 03 to read data from registers CRC: Our checksum use CRC-16/MODBUS

An example to read data from four registers:

Inquiry: 01 03 00 00 00 04 44 09

Response: 01 03 08 5D FF 00 24 00 00 40 00 1F BA

	Device Address	Function Code	Data					CRC Check sum
Inquiry command			Read data start from register *			Read * registers		
	01	03	0000			0004		4409
	Description		Read data start from register 1			Read 4 registers		
Response message			Bytes QTY	Register 1	Register 2	Register 3	Register 4	
	01	03	08	5DFF	0024	0000	4000	1FBA
	Description		8 bytes	Reserved	Wind Dir:36° HEX to Decimal	Wind Speed:2.0m/s IEEE754 Standard CDAB float type		

MODBUS RTU Register Map:

ADDRESS OF REGISTER	LENGTH	DATA TYPE	DEFINITION	RANGE
0000H	16-BIT	16-BIT INT	Reserved	Reserved
0001H	16-BIT	16-BIT INT	Wind Direction	0 - 359 deg.
0002H	16-BIT	32-BIT FLOAT	Wind Speed	0 - +40 m/s all 32-Bit float date in this protocol comply to IEEE754 Standard 32-Bit Float Little_endian byte swap
0003H	16-BIT			
0004H	16-BIT	32-BIT FLOAT	Temperature	-40 to +60 deg. C
0005H	16-BIT			
0006H	16-BIT	32-BIT FLOAT	Humidity	0 - 100%
0007H	16-BIT			
0008H	16-BIT	32-BIT FLOAT	Pressure	150 - 1100 hPa
0009H	16-BIT			
000AH	16-BIT	16-BIT INT		
000BH	16-BIT	16-BIT INT	Precipitation Type	Refer to Appendix III
000CH	16-BIT	32-BIT FLOAT	Precipitation Intensity	mm/h
000DH	16-BIT			
000EH	16-BIT	32-BIT FLOAT	Accumulated Precipitation	mm
000FH	16-BIT			
0010H	16-BIT	Reserved	Reserved	
0011H	16-BIT	16-BIT INT		
0012H	16-BIT	32-BIT FLOAT		
0013H	16-BIT			
0014H	16-BIT	16-BIT INT		
0015H	16-BIT	32-BIT FLOAT		
0016H	16-BIT			
0017H	16-BIT	32-BIT FLOAT		
0018H	16-BIT			
0019H	16-BIT	32-BIT FLOAT	PM2.5 Concentration	0-500 ug/m^3
001AH	16-BIT			
001BH	16-BIT	32-BIT FLOAT	Visibility	m
001CH	16-BIT			
001DH	16-BIT	32-BIT FLOAT	Illuminance	Lux
001EH	16-BIT			
001FH	16-BIT	32-BIT FLOAT	Sunshine Duration	Unit: seconds Duration of solar radiation > 120 W/m2
0020H	16-BIT			
0021H	16-BIT	32-BIT FLOAT	Solar Radiation Power	W/m^2
0022H	16-BIT			

0023H	16-BIT	32-BIT FLOAT		
0024H	16-BIT			
0025H	16-BIT	32-BIT FLOAT	Altitude	m
0026H	16-BIT			
0027H	16-BIT	32-BIT FLOAT		
0028H	16-BIT			
0029H	16-BIT	Reserved		
002AH	16-BIT	Reserved		
002BH	16-BIT	Reserved		
002CH	16-BIT	Reserved		
002DH	16-BIT	32-BIT FLOAT	PM 1.0 Concentration	0-500 ug/m^3
002EH	16-BIT			
002FH	16-BIT	32-BIT FLOAT	PM 10 Concentration	0-500 ug/m^3
0030H	16-BIT			
0031H	16-BIT	Reserved		
0032H	16-BIT	Reserved		
0033H	16-BIT	Reserved		
0034H	16-BIT	32-BIT FLOAT	10 minutes Averaged Relative Wind Speed	
0035H	16-BIT			
0036H	16-BIT	32-BIT FLOAT	10 minutes Averaged Relative Wind Speed	
0037H	16-BIT			
0038H	16-BIT	32-BIT FLOAT	10 minutes Averaged Relative Wind Speed	
0039H	16-BIT			
003AH	16-BIT	16-BIT INT	10 minutes Averaged Relative Wind Direction	
003BH	16-BIT	16-BIT INT	10 minutes Averaged Relative Wind Direction	
003CH	16-BIT	16-BIT INT	10 minutes Averaged Relative Wind Direction	
003DH	16-BIT	32-BIT FLOAT		
003EH	16-BIT			
003FH	16-BIT	32-BIT FLOAT		
0040H	16-BIT			
0041H	16-BIT	32-BIT FLOAT		
0042H	16-BIT			

0043H	16-BIT	32-BIT FLOAT		
0044H	16-BIT			
0045H	16-BIT	32-BIT FLOAT		
0046H	16-BIT			
0047H	16-BIT	32-BIT FLOAT		
0048H	16-BIT			
0049H	16-BIT	32-BIT FLOAT	Gust Maximum 3s Averaged windspeed over past 10 minutes	
004AH	16-BIT			
004BH	16-BIT	32-BIT FLOAT	2 minutes Averaged Relative windspeed	
004CH	16-BIT			
004DH	16-BIT	32-BIT FLOAT	2 minutes Averaged Relative windspeed	
004EH	16-BIT			
004FH	16-BIT	32-BIT FLOAT	2 minutes Averaged Relative windspeed	
0050H	16-BIT			
0051H	16-BIT	16-BIT INT	2 minutes Averaged Relative windspeed	
0052H	16-BIT	16-BIT INT	2 minutes Averaged Relative windspeed	
0053H	16-BIT	16-BIT INT	2 minutes Averaged Relative windspeed	
0054H	16-BIT	32-BIT FLOAT		
0055H	16-BIT			
0056H	16-BIT	32-BIT FLOAT		
0057H	16-BIT			
0058H	16-BIT	32-BIT FLOAT		
0059H	16-BIT			
005AH	16-BIT	32-BIT FLOAT		
005BH	16-BIT			
005CH	16-BIT	32-BIT FLOAT		
005DH	16-BIT			
005EH	16-BIT	32-BIT FLOAT		
005FH	16-BIT			

3.2 NMEA0183 protocol

Unless you request another configuration, otherwise sensor will adopts following default configuration:

Baud Rate: 9600 bps;

Parity: NONE;

Data Bits: 8 bits;

Stop Bits: 1bit;

Wind speed and direction data frame

$WIMWV,180,R,0.01,M,A*06<CR><LF>

E.g.	Format	Name	Description
$WIMWV	String	Fixed frame head	NMEA Wind Protocol Header
180	Number	Wind direction	Wind direction reading in °
R	Character	Type of wind	R: Relative wind speed and direction
0.01	Number	Wind speed	Wind speed reading
M	Character	Unit of Wind speed	K: Km/h M: m/s N: knots
A	Character	Status of data	A: data is valid V: data is invalid
***06**	HEX character	Hex checksum	Checksum, the 2 digit Hex Checksum sum figure is calculated from the Exclusive OR of the bytes between (don't include) the $ and * characters.
<CR><LF>	ASCII character	Line Feed Character	

Self-defined Data Frame

$HYASD,0.23,1.78,0.13,15,332,4,0.73,3.26,1.93,3.0,176.0,88.0,*71<CR><LF>

E.g.	Format	Description/Format
$HYASD	Character	Fixed frame head
0.23	Numerical value	0-10 min Avg. Relative Wind Speed
1.78	Numerical value	0-10 min Max. Relative Wind Speed
0.13	Numerical value	0-10 min Min. Relative Wind Speed
15	Numerical value	0-10 min Min. Relative Wind Direction
332	Numerical value	0-10 min Min. Relative Wind Direction
4	Numerical value	0-10 min Min. Relative Wind Direction
0.73	Numerical value	0-10 min Avg. True Wind Speed
3.26	Numerical value	0-10 min Max. True Wind Speed
1.93	Numerical value	0-10 min Min. True Wind Speed
3.0	Numerical value	0-10 min Avg. True Wind Direction
176.0	Numerical value	0-10 min Max. True Wind Direction
88.0	Numerical value	0-10 min Min. True Wind Direction
***71**	HEX character	Checksum
<CR> <LF>	ASCII character	Line Feed Character (0x0D 0x0A)

Immediately followed by:

Self-defined data frame

$HYMSD,3.78,1.23,2.78,3.13,45,132,24,1.73,4.26,2.93,5.0,276.0,188.0,*46

<CR>< LF>

3.3 Unsolicited active output in ASCII protocol

Unless you request for other configuration, otherwise sensor will adopts following default configuration:

Baud Rate:9600 bps;

Parity:NONE;

Data Bits: 8 bits;

Stop Bits: 1bit; Data frame(ASCII)

E.g. :01,56,0.3,...,<CR><LF>**

Data Segment	Character	Description	Number of Characters	Remark
Frame header	:	Frame header	1	0x3A
Address	01	Device Address	2	0x01~0xFF
Separator	,		1	
Data 1	56	Wind direction	Unfixed length	0~359 °
Separator	,		1	
Data 2	0.03	Wind speed	Unfixed length	0~40 m/s
Separator	,		1	
Data 3	... other data			
Separator	,		1	
Checksum	**	LRC Checksum	2	0x00~0xFF
Frame Ending	<CR>	Frame Ending 1	1	0x0D
	<LF>	Frame Ending 2	1	0x0A

3.3 Unsolicited active output in ASCII protocol

Unless you request for other configuration, otherwise sensor will adopts following default configuration:

Baud Rate:9600 bps;

Parity:NONE;

Data Bits: 8 bits;

Stop Bits: 1bit; Data frame(ASCII)

E.g. :01,56,0.3,...,<CR><LF>**

Data Segment	Character	Description	Number of Characters	Remark
Frame header	:	Frame header	1	0x3A
Address	01	Device Address	2	0x01~0xFF
Separator	,		1	
Data 1	56	Wind direction	Unfixed length	0~359 °
Separator	,		1	
Data 2	0.03	Wind speed	Unfixed length	0~40 m/s
Separator	,		1	
Data 3	... other data			
Separator	,		1	
Checksum	**	LRC Checksum	2	0x00~0xFF
Frame Ending	<CR>	Frame Ending 1	1	0x0D
	<LF>	Frame Ending 2	1	0x0A

3.4 SDI-12 protocol

Unless you request for other configuration, otherwise sensor will adopts default configuration:
Baud Rate:1200 bps;
Parity: EVEN;
Data Bits: 7 bits;
Stop Bits: 1bit;

SDI-12 Commands without CRC
?! Returns Unit Address (default is 0).
a Current unit address letter (factory default is 0, range is 0 to 9, a to z, A to Z).
b New address letter, range is 0 to 9, a to z, A to Z.
aAb! Change unit address from a to b see above.
aM! Address, Relative Wind Direction, Speed, True Wind Direction, Speed, State.
aD0! Request a line of the above data

Command	Description	Response	Example
?!	Inquiry unit address	a<CR><LF>	0<CR><LF>
aAb!	Change the unit address from a to b	b<CR><LF>	1<CR><LF>
aI!	Unit Identification	a...<CR><LF>	014H… 1000002.3000<CR><LF>
aM!	Request to prepare data of Relative Wind Direction, Speed, True Wind Direction, Speed	atttn<CR><LF> a:address ttt:data preparation time in seconds. n:qty of data	00015<CR><LF> Address 0 Data will be ready within 1 second 5 data will be provided
aD0!	Fetch above data	<address> <relative wind direction> <relative wind speed> <true wind direction> <true wind speed> <status> <CR><LF>	0+056+00.03+000+00.00+1100 <CR><LF> 0:address 056:relative wind direction 56° 00.03:relative wind speed 0.03m/s

			000: true wind direction 56° 00.00:true wind speed 0.00m/s 1100:state of each data, 1 stand for valid, 0 stand for invalid
aM6!	Start measurement 6	atttn<CR><LF> ttt is averaged period set by user.(1~600s)	00008<CR><LF>
aD0!	Retrieve a line of the above data	<address> <10 minutes averaged relative wind speed > <10 minutes maximum relative wind speed > <10 minutes minimum relative wind speed > <CR><LF>	0+00.23+01.78+00.13 <CR><LF>
aD1!	Retrieve a line of the Additional measurement data	<address><10 minutes averaged relative wind direction > <10 minutes maximum relative wind direction > <10 minutes minimum relative wind direction > <Gust><CR><LF>	0+015+332+004+03.78+1111111<CR><LF>

SDI-12 Commands with CRC

aMC!   Address, Relative Wind Direction, Speed, True Wind Direction, Speed, State and CRC.
aD0!   Request a line of the above data

Command	Description	Response	Example
aMC!	Request to prepare data of Relative Wind Direction, Speed, True Wind Direction, Speed	atttn<CR><LF> a:address ttt:data preparation time in seconds. n:qty of data	00015<CR><LF> Address 0 Data will be ready within 1 second 5 data will be provided
aD0!	Fetch data of above command	<address> <relative wind direction> <relative wind speed> <true wind direction> <true wind speed> <status><CRC> <CR><LF>	0+056+00.03+000+00.00+1100CJP <CR><LF> 0:address 056:relative wind direction 56° 00.03:relative wind speed 0.03m/s 000: true wind direction 56° 00.00:true wind speed 0.00m/s 1100:state of each data, 1 stand for valid, 0 stand for invalid CJP:CRC Checksum
aMC6!	Start measurement 6	atttn<CR><LF> ttt is averaged period set by user.(1~600s)	00108<CR><LF>
aD0!	Retrieve a line of the above data	<address> <10 min averaged relative wind speed > <10 min max relative wind speed > <10 min Min relative wind speed ><CRC> <CR><LF>	0+00.23+01.78+00.13 GMw<CR><LF>

aD1!	Retrieve a line of the Additional measurement data	<address> <10 min averaged relative wind direction > <10 min max relative wind direction > <10 min Min relative wind direction > <Gust><CRC> <CR><LF>	0+015+332+004+03.7 8+1111111HVs <CR><LF>

3.5 Change protocol and serial port configuration

Following commands can be used to change communication address, baud rate, parity, data bit, stop bit

Commands		Content	Response
Instruction 1	ASCII	>*\r\n	>CONFIGURE MODE\r\n
	HEX	3E 2A 0D 0A	0A 3E 43 4F 4E 46 49 47 55 52 45 20 4D 4F 44 45 0D 0A
Remark	Enter Setting Mode		
Instruction 2	ASCII	>CUS 9600 8-N-1\r\n	>CMD IS SET
Remark	Configure serial port configuration as Baud Rate 9600 bps; Data bits:8 bits; Parity: NONE; Stop bits:1 bit.		
Instruction 3	ASCII	>ID 2\r\n	>CMD IS SET
Remark	Configure address of device as 2. Inquiry address command is HEX: 3E 49 44 0D 0A		
Instruction 4	ASCII	>RESET\r\n	System start ok!\r\n
Remark	Reboot device		
Instruction 5	ASCII	>!\r\n	
Remark	Exit setting mode to normal mode.		
Instruction 6	ASCII	>UartPro:3\r\n	
Remark	Change protocol as MODBUS. Four options below are available. 0: active ASCII output 3:MODBUS RTU 4:NMEA0183 5:SDI-12		
Instruction 7	ASCII	>SaveConfig\r\n	
Remark	Save changes		
Notice: 1. Characters "\r\n" is CRLF Carriage-Return Line-Feed, corresponding to HEX (0x0D,0x0A)			

Procedures:

No.	Function	Instructions
1	Set Communication Address	1→3→5→4
2	Set Serial Port Configuration	1→2→5→4
3	Change communication protocol	6→7→4

3.6 Explanation of decoding CDAB float type data

All 32-bit float data in this protocol complies to IEEE754 Standard 32-Bit Float Little_endian byte swap.

For example:
Send command to fetch data in register 02H and 03H: Send: 01 03 00 02 00 02 65 CB
Received: 01 03 04 51 EC 40 18 1A F0
Now we get raw data of wind speed: 51 EC 40 18
1. Swap higher and lower byte, then we get 40 18 51 EC
2. Convert it into binary:
HEX:401851EC
=>Binary:01000000000110000101000111101100 (32 bits)

1st bit is sign bit, 0 stand for positive and 1 stand for negative.
Exponent = (convert 2nd~9th bits to decimal - 127)=128-1=1
Mantissa = (convert 10th~32nd bits to decimal) / (2^23) = 1593836/8388608 = 0.190000057220458984375
Final result = sign*(1+Mantissa)*2^Exponent = 2.38000011444091796875

Now we successfully decode CDAB float data “401851EC” as 2.38, which means we have wind speed of 2.38 m/s.

You can also use C language’s sub-function to transform 4 bytes (HEX) as float data
(C language)

```
union
{
float TestData_Float; unsigned char
TestArray[4];
}TData;
```

Analysis example:

D3	D2	D1	D0
Higher byte of register 2	Lower byte of register 2	Higher byte of register 1	Lower byte of register 1
40	AC	19	DF
Higher byte	Middle byte 1	Middle byte 2	Lower byte

After transformed to float data, value: 5.378 Subfunction:
float Tempfloat;

```
TData.TestArray [3]= 0x40; //input higher byte
TData.TestArray [2]= 0xac; // TData.TestArray [1]= 0x19;
// TData.TestArray [0]= 0xdf; //input lower byte
Tempfloat = TData.TestData_Float; //return result 5.378
```

3.7 Explanation of CRC calculation

Our checksum use CRC-16/MODBUS

The CRC we are using is 16 bits, lower byte comes first.

The cyclic redundancy check (CRC) field is two bytes which contain 16 bits binary value. The value of the CRC appended to the message is calculated by the transmitting device. When receiving the message, the receiving device recalculates the CRC value and compares the calculated result with the actual received CRC value. If the two values are not equal, it is an error.

During the generation of CRC, each 8-bits characters are XOR with the value in the register. The result then shifts 1 bit in the LSB direction, while the MSB position is charged to zero. Then extract and check LSB: if LSB is 1, the value in the register is XOR with a fixed preset value; if LSB is 0, no XOR operation is performed.

This process will be repeated until 8 shifts have been performed. After the last (8th) shift and related operations, the next 8-bit byte is XOR with the current value of the register, and then repeat 8 times as described above. The final value in the register obtained after all sub sections of the message are calculated is CRC.

Procedure of calculating a CRC:

1. Load a 16 bit register with hexadecimal FFFF (all 1). Call it CRC register.

2. XOR the first byte of 8 bits in message with the low er byte of the 16 bit CRC register, and place the result in the CRC register.

3. Move the CRC register to the right by 1 bit (in the direction of LSB), fill the MSB with zero, extract and detect LSB

4. If LSB is 0: repeat step 3 (do another shift)
(if LSB is 1): conduct XOR operation with CRC register.

5. Repeat steps 3 and 4 until 8 shifts have been completed. When this is done, the full 8-bit byte operation will be completed.

6. Repeat steps 2 to 5 for the next byte in the message, and continue the operation until all messages are processed.

7. The final content in CRC register is CRC value

8. When placing CRC value in message, as described below, higher and lower byte must be exchanged.

3.8 Explanation of LRC calculation

Using a C language function code to generate LRC values The function code uses 2 independent variables：

```
unsigned char *auchMsg; // To generate the LRC value, point the pointer to the buffer containing the binary data
unsigned short usDataLen; //Number of bytes in the buffer
//This function returns LRC as a type “unsigned char”
// RC check code generation
static unsigned char LRCCheck(auchMsg, usDataLen)
unsigned char *auchMsg; /* calculating by information byte LRC*/ unsigned short usDataLen; /*calculating by information byteLRC*/
{
unsigned char uchLRC = 0 ; /*Initializing LRC characters */ while (usDataLen--)
/*through the data buffer*/
uchLRC += *auchMsg++; /*Add buffer byte Buffer byte no carry*/ return ((unsigned char)(–((char)uchLRC))) ; /*return to Binary complement*/
}
```

4. Troubleshooting & Support

Note: Each sensor is thoroughly tested to guarantee their accuracy and well communication. You can trust it will work well as long as power supply is stable and wiring is correct.
We would recommend you to use computer with R232/RS485 or SDI-12 to USB converter to test the sensor directly, so that we can exclude other potential failure factors.

4.1 No start up string or garbled code received upon power on

All of our sensors send “>System Startup\r\n” at the moment it’s powered up unless its specially requested not to. You can also check power consumption of sensor, sensor itself will consume at least 10mA@12VDC.
If problem still exist, please change communication interface to USB converter and try it again.

4.2 Garbled code received upon power on

Two possibilities that may lead to this phenomenon:

1. Incorrect data cable wiring.
2. Choosing wrong baud rate or parity.

4.3 Can’t enter configuration mode

All of our sensor respond “>CONFIGURE MODE\r\n” after receiving command “>*\r\n”.
Following possibilities may lead to this failure:
1. Parity is incorrect, please try to change parity and try again.
2. You didn’t input “\r\n” character correctly, “\r\n” is CRLF Carriage-Return Line-Feed, corresponding to HEX (0x0D,0x0A). Some UART software use Shift+Enter to enter those characters.
3. Wrongly use RS232 as RS485 or vice versa.

4.4 Can’t fetch data by MODBUS RTU request command

Following possibilities may lead to such failure:
1. Current protocol is not MODBUS RTU.
2. Access registers outside of scope, please reduce quantity of registers in inquiry command.
3. Wrong device address
4. Wrong CRC checksum

4.5 Wind speed and direction is invalid

Please check ultrasonic probes and sampling path. Wind speed and direction will be unavailable if any of them is blocked.

5. Calibration

Each unit is calibrated in a wind-tunnel testing lab prior to shipping.
The anemometer calibration is based on fundamental physical principles and does not change with use. Re-calibration should therefore not be necessary.
However, you may test it by wind tunnel to inspect its accuracy from time to time.
A test report from 3rd party Institution of Meteorological department below for your reference (third party tests will be charged. Pricing to be confirmed).

6. Warranty

Warranty Period: 24 months from the date of delivery. Following situations are not covered by warranty:

- Damage resulting from negligence
- Damage resulting from unauthorized modification of the product
- Damage caused by natural disaster

6.1 How do customers get repair / service under warranty?

99.99% of communication issues are caused by incorrect wiring or serial port configuration.

Please contact HubIT for further technical guidance and any hardware malfunctions.

Following frequently asked questions will help us understand your issue better:

1. When did malfunction happen?
2. Did you change anything before malfunction?
3. What is power consumption of sensor? How many ampere does it consume?
4. Is there startup string when it's powered up?
5. Has it ever supplied with over-voltage?

We will reply you within 48 hours after receiving your email.

6.2 What's the charge of repairing?

Our products are reliable. For quality issues caused through vendor-related issues within one to six months from receipt of shipment, HubIT will cover shipping charges.

Any other situations, will be discussed on a case by case basis.
For malfunction caused by client's negligent operation, HubIT will charge for the repair, and client shall additionally pay shipping charges.