Meteorological Instruments Expert

HONGYUV

HY-WDC2E Ultrasonic Anemometer

User Manual

04/30/2024



Table of Contents

Table of Contents 2

1. Introduction 3
   1.1 Overview 3
2. Getting Started 5
   2.1 Wiring 5
   2.2 Communication test 5
   2.3 Mounting 6
   2.4 Select Installation Location 7
3. Communicate with sensor 8
   3.1 MODBUS RTU protocol 9
   3.2 NMEA0183 protocol 12
   3.3 Unsolicited active output in ASCII protocol 15
   3.4 SDI-12 protocol 16
   3.5 Change protocol and serial port configuration 20
   3.6 Explanation of decoding CDAB float type data 22
   3.7 Explanation of CRC calculation 24
   3.8 Explanation of LRC calculation 25
4. Troubleshooting & Support 26
   4.1 No start up string or garbled code received upon power on 26
   4.2 Garbled code received upon power on 26
   4.3 Can't enter configuration mode 26
   4.4 Can't fetch data by MODBUS RTU request command 26
   4.5 Wind speed and direction is invalid 27
5. Calibration 28
6. Warranty 29
   6.1 How do customers get repair/service under warranty? 29
   6.2 What's the charge of repairing? 29

1. Introduction

This User Manual provides necessary information for user to install and communicate with HY-WDC2E ultrasonic anemometer.

1.1 Overview

HY-WDC2E ultrasonic anemometer is very robust with no moving parts, 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-WDC2E can be powered up with 5-30 VDC(4-20mA output require 12~24VDC), please note that some specially customized sensors has narrower power supply range. It can be communicated 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 it when you place order.
It is typically equipped with a 4 pins M16 connector for installation.(4-20mA output has more Pins)

HY-WDC2E ultrasonic anemometer 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.

TECHNICAL SPECIFICATION

Parameters	Range	Accuracy	Resolution
Wind Speed	0 - 40m/s	±5%	0.1m/s
Wind Direction	0 - 360°	±3°	1°
Digital Output	RS485,RS232,SDI-12		
Analog Output	4-20mA(optional with extra cost)		
Baud Rate	1200 ~ 115200 bps		
Communication Protocol	ModBus RTU or NMEA-0183 or SDI-12		
Protection Grade	IP65		
Operating Temperature	-40℃ - +70℃		
Storage Temperature	-50℃ - +80℃		
Working Humidity	0 - 100%		
Power Supply	VDC: 5-30V(3.5V can be customized)		
Power consumption	15mA @5V		
Dimension/Weight	ASA: Φ82×108mm 0.18kg Aluminum alloy: Φ82×125mm 0.38kg		
Material	ASA engineering plastic or Aluminium alloy(optional with extra cost)		

Specifications may be subject to change without prior notice.

2. Getting Started

2.1 Wiring

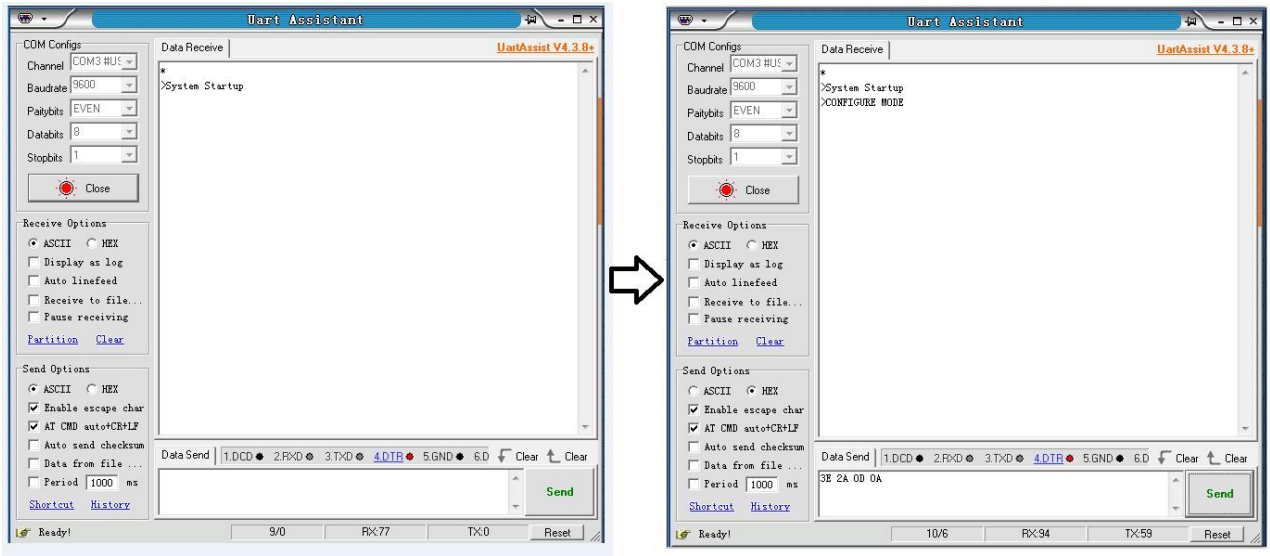
Color of cable		Red	Black	Yellow	Green
Definition	RS485	Power+	Power- & RS232/TTL GND	DA+	DB-
	RS232			RXD	TXD
	SDI-12			-	SDI-12
	TTL			TXD	RXD

2.2 Communication test

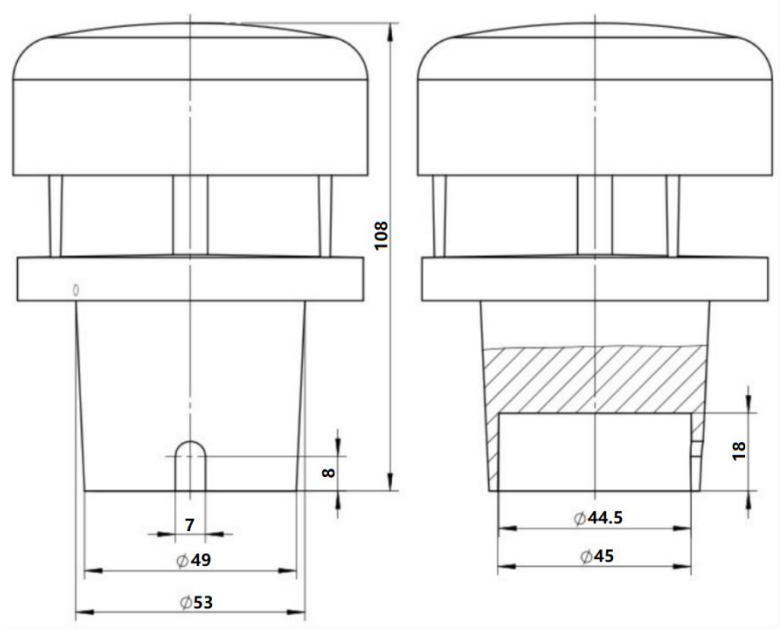
Note: We don't recommend user to install sensor or connect it to your data logger before testing communication by procedure below. We highly recommend you to use corresponding communication interface to USB converter to to test our sensor on computer before conducting any further operation.

Our sensor will send 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) Until now, you can confirm communication with sensor is well established.



2.3 Mounting



Noted: We also provide following kits for your option



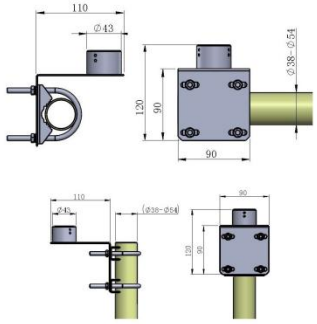
Material: Aluminium

Tripod installation kit 40 USD/UNIT
Tripod is not included



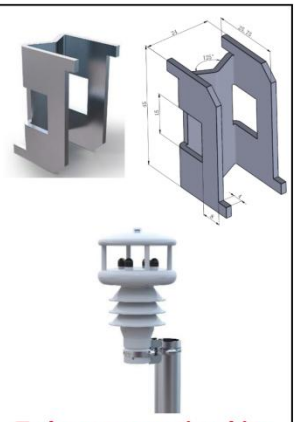
Mounting mast
25 USD/UNIT
Material: Cast Aluminium

Mounting mast 25 USD/UNIT



Material: SUS304

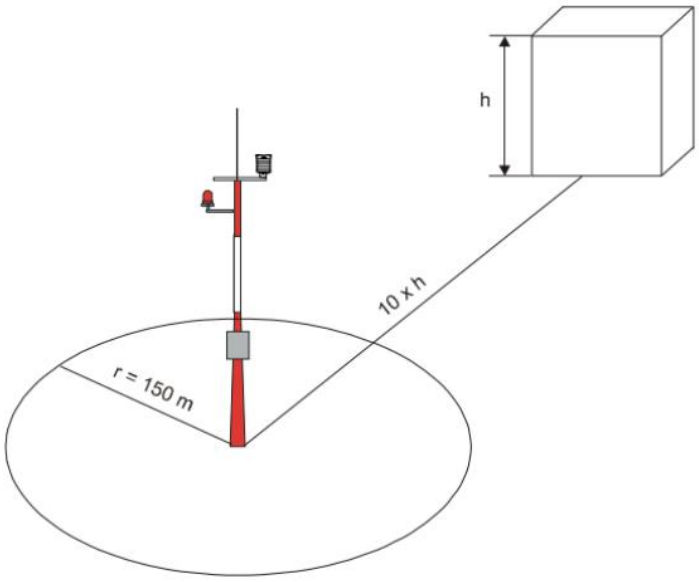
Mounting bracket 25 USD/UNIT



Z shape mounting kits
15 USD/UNIT

2.4 Select 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 lenght of cable comes with anemometer is 4 meters. This cable may be cut to shorter or extended to a maximum of 1000 meter(for RS485) or 10 meters(for RS232)
Following World Meteorological Organization (WMO) guidelines, a general recommendation is that there is at least 150 m 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 1.5 W. However, follow the application specific instructions and local regulations when placing anemometer.



3. Communicate with sensor

Noted: Our sensor has
Three kinds of digital interface: RS485, RS232, SDI-12 for option.
Analog interface: 4-20mA, 0-5VDC for option.
Those are decided when you place order, and they can't be changed after production.
Please note that only advanced WDC2E anemometer come with averaged wind and gust, and there is surcharge. Please mention it when you place order.
Please also note that following protocols are for reference only, you can't fetch reading of parameters that you didn't ordered.
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, otherwise sensor will adopts following default configuration:
Baud Rate:9600 bps; Parity:NONE;
Data Bits: 8 bits; Stop Bits: 1bit;
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's essential to assign an 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 Quantity	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 °
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			
...				
0034H	16 bit	32 bit float	10 minutes averaged wind speed	
0035H	16 bit			
0036H	16 bit	32 bit float	10 minutes Maximum wind speed	
0037H	16 bit			
0038H	16 bit	32 bit float	10 minutes Minimum wind speed	
0039H	16 bit			
003AH	16 bit	16 bit int	10 minutes averaged wind direction	
003BH	16 bit	16 bit int	10 minutes Maximum wind direction	
003CH	16 bit	16 bit int	10 minutes Minimum wind direction	

...				
0049H	16 bit	32 bit float	Gust Maximum 3s averaged wind speed over past 10 minutes	
004AH	16 bit			
004BH	16 bit	32 bit float	2 minutes averaged wind speed	
004CH	16 bit			
004DH	16 bit	32 bit float	2 minutes Maximum wind speed	
004EH	16 bit			
004FH	16 bit	32 bit float	2 minutes Minimum wind speed	
0050H	16 bit			
0051H	16 bit	16 bit int	2 minutes averaged wind direction	
0052H	16 bit	16 bit int	2 minutes Maximum wind direction	
0053H	16 bit	16 bit int	2 minutes Minimum wind direction	

3.2 NMEA0183 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;
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 Avg. Relative Wind Direction
332	Numerical value	0-10 min Max. 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>

E.g.	Format	Description/Format
$HYMSD	Character	Fixed frame head
3.78	Numerical value	Gust (3s)
0.23	Numerical value	0-2 min Avg. Relative Wind Speed
1.78	Numerical value	0-2 min Max. Relative Wind Speed
0.13	Numerical value	0-2 min Min. Relative Wind Speed
15	Numerical value	0-2 min Avg. Relative Wind Direction
332	Numerical value	0-2 min Max. Relative Wind Direction
4	Numerical value	0-2 min Min. Relative Wind Direction
0.73	Numerical value	0-2 min Avg. True Wind Speed
3.26	Numerical value	0-2 min Max. True Wind Speed
1.93	Numerical value	0-2 min Min. True Wind Speed
3.0	Numerical value	0-2 min Avg. True Wind Direction
176.0	Numerical value	0-2 min Max. True Wind Direction
88.0	Numerical value	0-2 min Min. True Wind Direction
***71**	HEX character	Checksum
<CR><LF>	ASCII character	Line Feed Character (0x0D 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>	014HONGYUV 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	0+00.23+01.78+00.13GMw<CR><LF>

		wind speed ><CRC> <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+03.78+1111111HVs <CR><LF>

3.5 Change protocol and serial port configuration

You can download our software for Windows OS to configure serial port by link:
http://www.hongyuv.com/download/Serial_Configuration_Tool.zip
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 date in this protocol comply 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:01000000001100001010001111011 00 (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
=2.380000114440917968 75

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))) ; /*reture to Binary complement*/
}
```

4. Troubleshooting & Support

Note: Each sensor is well tested in our factory 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 our 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 it’s 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 our 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.(Such test charge $300/time)



四川省气象计量检定所检测证书
Test Certificate -Sichuan Institution of Metrology Verificaton

证书编号：CQJ 字第 F2018036 号
Certificate No.

检 测 数 据 / 结 果
Data / Results of Test

检测结果：

实际值 Reference wind speed （m/s）	显示值 HY-WDC2DVSE Wind Speed （m/s）
1.95	1.97
4.87	5.01
9.83	10.10
19.88	19.88
24.96	24.73
30.04	29.53
35.16	34.64

（以下为空白）

2.环境条件:
Environmental condition of the test
地点：四川省气象计量检定所大楼 1 楼风速实验室
Place
环境温度： 8.8℃; 相对湿度： 48 % RH; 大气压力： 961.6hPa
Ambient Temperature Relative Humidity Atmosphere

检测证书续页专用
Continue page of Test Certificate

第 3 页 共 3 页
Page of

6. Warranty

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

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

6.1 How do customers get repair/service under warranty?

99.99% communication issue is caused by incorrect wiring or serial port configuration.
You may contact stefan@hongyuv.com to get further technical guidance by email or remote support by UltraViewer https://www.ultraviewer.net/
You may contact your vendor or HongYuv to solve malfunctions caused by hardware.
Following frequently asked questions will help us understand situation well.

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 24 hours after receiving your email.

6.2 What's the charge of repairing?

Our products are reliable and rarely failed. For quality issue caused by our fault within one/six month since shipment, HongYuv will cover for double-way/single-way shipping charge
For other situations, it's negotiable.
For malfunction caused by client's negligent operation:
HongYuv will only charge for repair fee based on our raw cost, and client shall pay shipping charge forth and back.