WINCOM TECHNOLOGY

Vibrating Wire Sensor Reading Module VM501/511/604/608/704S

User Manual

WINCOM Technology Co., Ltd. (Firmware: V3.33)

September 2018 (Document Version: V1.21)

Foreword

Thank you for using the VM5/6/7XX series module produced by WINCOM Technology Co., Ltd. This product is specially designed and developed for single-coil vibrating wire sensor, which can complete the coil excitation, frequency reading and temperature measurement of the sensor, with standard UART (TTL/RS232/RS485) and IIC digital interface, through the digital interface data interaction, you can complete the vibrating wire sensor detection, excitation, reading and other work. This module provides the vibrating wire frequency and temperature sensor read and output functions. The professional circuit and firmware design for the vibrating wire sensor characteristics, as well as the small module size, diverse package, standardized industrial communication protocol and many other features make this product fast. Integrate into your acquisition monitoring device or handheld reading device. Please read this manual carefully before use to understand the features of this module and get the complete usage.

Please be sure to operate or design the peripheral circuit according to the specific instructions in this manual. The company will not be responsible for property damage or personal injury caused by abnormal operation. Please use or design the corresponding products in strict accordance with the technical specifications and parameters in the manual.

Features

- **Small Size:**
  VM501 30.0mmX26mmX4.3mm(LxWxH) SMT package -20
  VM604/608 30.0mmX26mmX4.3mm SMT package -20
  VM511/614 60.0mmX36mmX5.8mm in-line package -22
  VM704S 32mmX32mmX15mm in-line package -20
- **Digital Interface**: Standard UART/RS232/RS485+IIC
- **Communication Rate:**
  UART：9600，N，8，1 9.6~1382.4kbps
  IIC：500kHz
- **Frequency Range:** 30~12000Hz
- **Measurement Rate:** 20Hz (up to 20 times per second in high-speed mode)
- **Strong Compatibility:** It can measure the single-coil vibrating wire sensor of most manufacturers.
  Multi-module combination can complete multi-coil sensor data reading (such as: Anchor rope stress meter, multi-point displacement meter, etc.)
- **High Precision:** frequency measurement accuracy 0.05Hz~0.50Hz (@100Hz~8000Hz)
- **Automatic Sensor Identification:**
  Sensors that can identify coils from 50Ω to 10kΩ
- **Multiple Measurement modes:** Automatic continuous measurement, Single Measurement
- **Multiple Excitation Methods:**
  Programmable high-voltage excitation,
  Low-voltage sweeping frequency,
  intelligent feedback frequency excitation
- **Signal Quality Assessment:**
  sampled data quality assessment algorithm,
  signal amplitude detection,
  signal quality assessment
- **Temperature Detection:** Thermistor/DS18B20/Core temperature
- **Unique Identification Code:** Global Unique Identifier
- **Very Few External Components:** The minimum system requires only external power supply, vibrating wire sensor and digital interface to complete the main work, shorten product development cycle and reduce

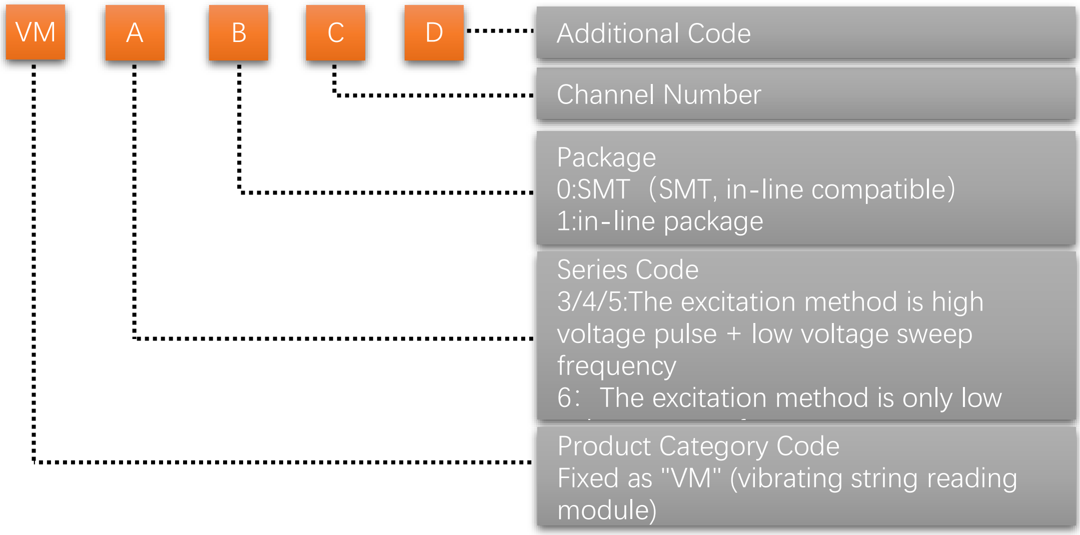
product development cost

- **Industrial Standards:**
  -40℃~+85℃

Application Field

- **Stress and strain:** structural stress and strain, foundation pit support, pipe gallery, underground engineering
- **Instrumentation:** Development of vibrating wire reading instrument
- **Automation, Information Technology:** combined with Internet technology to replace traditional manual detection

Ordering Information



TYPE	SIZE (mm)	Package	Chs	Interface						Precision		Excitation		Prog Amp
				UART			IIC	CAN	AOU	FRE	TEP	H	L	
				TTL	232	485								
VM301	56*37*5.2	S	1+1	●					●I	0.05~0.50	0.5	●	●	
VM311	60*36*4.8	I	1+1	◎	⦿	◎			●I	0.05~0.50	0.5	●	●	
VM401	30*26*4.3	S	1+1	●					●I	0.05~0.50	0.5	●	●	●32
VM411	60*36*2.5	I	1+1	◎	⦿	◎			●I	0.05~0.50	0.5	●	●	●32
VM501	30*26*4.5	S+I	1+1	⦿			◎	●	●V	0.01~0.05	0.3	●	●	●03
VM511	60*36*4.8	I	1+1	◎	⦿	◎	●	●	●V/I	0.01~0.05	0.3	●	●	●03
VM604	30*26*4.5	S+I	4+4	⦿			◎		◗V	0.01~0.05	0.3		●	●03
VM608	30*26*4.5	S+I	8+1	⦿			◎		✱V	0.01~0.05	0.3		●	●03
VM704	30*26*4.5	I	4+4	⦿					◗V	0.01~0.05	0.3	●	●	●03
VM708	30*26*4.5	I	8+1	⦿					✱V	0.01~0.05	0.3	●	●	●03

S: SMT.I: In-Line.Chs: Frequency + Temperature.

⦿◎: Relations as "OR",⦿as default。◗:Four pins, interval output 0 or valid value。✱: Single pin, time output each channel effective value.

●I: Current signal, resolution 1/8000.●V: Voltage signal, resolution 1/4000.


VM301
vibrating wire sensor readout unit
produced by wincom CO., LTD
VM311
produced by wincom CO., LTD
COMWIN
VM401
VM501
VM511
produced by wincom CO., LTD
VM604
VM608

Copyright and Trademark

Contact number: +86-0316-3093523 400-096-5525

Corporate website: www.winkooo.cn

Mailbox: info@geo-ins.com info@geo-explorer.cn

WINCOM Technology Co., Ltd.

About This Manual

This manual is a guide for the use of the VM5/6/7XX modules of our series, which is specially provided to end customers and electronic product design engineers to complete different levels of application.

VM5XX , VM6XX and VM7XX have almost the same usage method. In the following description, VM5XX is taken as an example unless otherwise specified.

The default factory parameters of the vibrating wire module have been set to be suitable for most sensors, generally no modification is required, and can be read directly from "Chapter 5 Quick Test".

Before you use or operate VM5XX, please read it carefully and have a general understanding of the function of the module. Read from the whole to the parts, and if necessary, combine the recommended test tools (VMTool) in this manual. VM35X keeps upgrading and improving, striving for higher integration and easier operation to reduce application difficulty. Please refer to the relevant sections of this manual for specific operations.

VM5XX can be used as a terminal product directly, or further secondary product development based on modules. Different user requirements are different for different purposes. In comparison, For simple use, Users are required to have some common knowledge of computer operation, hardware interface (especially serial port RS232), data instruction frame, etc. when you need to develop new products based on this module, you need to have a basic knowledge of electronic technology and the ability of circuit design and programming. During the use of this module, you may need to consult relevant basic knowledge and concepts through other channels. This manual will not cover all aspects. For the above auxiliary knowledge and operation ability, please solve it yourself.

Chapter 5, "Quick Test" describes how to use the computer to implement the rapid test of this module. In the manual reading process, you can also use the special configuration test tool VMTool to perform parameter modification and verification to deepen the understanding of the product and the parameter understanding.

In this manual, uses specific symbols to represent different types of Numbers, The number prefixed with "0x" is hexadecimal data, and the suffix of "B" is binary data. The number without any prefix or suffix is decimal. "\r\n" means non-visible carriage returns and line feeds.

Registers are usually represented by abbreviations (multiple uppercase letters), and some "bits" in a register are represented by "register abbreviation.

[High bit num: Low bit num]".

A section with a "*" indicates that the function is not stable or has not been fully tested. Please use it after consultation.

Due to product version upgrade or other reasons, the contents of this document will be updated from time to time. Please use the hardware and firmware version corresponding to this manual. If necessary, please ask us for the technical manual that matches your actual use.

Unless otherwise agreed, this document is provided as a guide only. All statements, suggestions and examples in this document do not constitute any warranty, express or implied, and do not assume any responsibility for the damage caused. The final interpretation of the contents in this manual belongs to our company.

Basic Concept

Vibrating Wire Sensor: A resonant sensor that uses a tensioned metal steel wire as a sensitive component. When the length of the wire is determined, the amount of change in the natural vibration frequency can be used to characterize the amount of tension applied to the steel wire. According to this characteristic principle, sensors for measuring different kinds of physical quantities (such as strain sensors, pressure sensors, displacement sensors, etc.) can be produced by a certain physical (mechanical) structure, thereby realizing one between the measured physical quantity and the frequency value. In a correspondence relationship, the amount of change in the measured physical quantity is calculated by measuring the amount of change in the frequency value.

Vibrating Wire Sensor Reading Module: In this manual, the sensor excitation and reading module designed for the characteristics of the vibrating wire sensor. With a series of features of high integration, functional modularity and digital interface, it can perform special targeted functions such as excitation, signal detection, data processing and quality evaluation of vibrating wire sensors, and perform physical frequency analog-to-digital conversion of sensor frequency and temperature. The digital interface implements data interaction. The vibrating wire sensor reading module is the core conversion unit between the vibrating wire sensor and digitization and informationization.

Excitation: It is a necessary process for the acquisition of frequency data of vibrating wire sensors. Only when the sensor receives the appropriate excitation signal can the self-vibration be generated, and only when the vibration wire sensor generates the self-vibration can the frequency signal be output. Further, the reading circuit will detect and read the self-vibration signal of the vibrating wire sensor, so as to obtain the vibration frequency value by calculation. The excitation signal of the vibrating wire sensor (an external signal capable of generating self-vibration of the sensor) is generally divided into two categories, one is a high-voltage short pulse, and the other is a plurality of sets of continuous low-voltage pulse signals of a specific frequency.

High-Voltage Pulse Excitation :(HPM) A process or method that uses a higher voltage (100~200V) to send a short pulse to the vibrating wire sensor coil to generate a self-oscillation of the vibrating wire sensor of any frequency.

Low-Voltage Sweeping Excitation:(LSM) A process or method in which a continuous low-voltage (3~20V) pulse signal is sent to a vibrating wire sensor at a frequency that is comparable (close to) to the sensor's natural frequency, causing the sensor to self-vibrate.

Vibrating Wire Sensor Return Signal: When the sensor generates self-vibration, the steel wire self-vibrates to cut the sensor coil, which generates a weak current in the coil. This sinusoidal electrical signal, which varies with the vibration of steel wires, is called the "return signal of the vibrating wire sensor".

Sampled Value: or “single sample”, especially in this manual, a single sinusoidal signal returned by the sensor. In order to improve the measurement accuracy of the sine wave frequency value, multiple sets of sinusoidal signals need to be collected for comprehensive calculation. Since the sinusoidal signal returned by the sensor gradually disappears from strong to weak, and the signal itself is very weak, the signal strength and duration of the vibrating wire sensor of different manufacturers are different. Therefore, the vibrating wire module uses partial sampling to obtain several samples during data acquisition. The data is calculated comprehensively, and each collected sine wave is called "one sample" or "one sample value".

Standard Deviation: Also known as the mean “square error” in the Chinese environment, is the square root of the difference between the standard value (sample value) of each unit and the arithmetic mean of its average. The standard deviation can reflect the degree of dispersion of a data set (two sets of data with the same average, the standard deviations are not necessarily the same). A large standard deviation represents a large difference between most of the values and their average values; a smaller standard deviation means that these values are closer to the average and the quality is higher.

ADC: (Analog-to-Digital Converter) analog-to-digital converter. A device that converts a continuously changing analog signal into a discrete digital signal.

VREF: (Voltage reference) is a voltage in a circuit that is always constant regardless of load, power supply, temperature drift, time, etc. The reference voltage can be used in power supply system regulators, analog to digital converters and digital to analog converters, as well as many other measurement and control systems.

GPIO: (General Purpose Input Output) A general-purpose input/output (interface) or bus extender. In layman's terms, there are pins that can be output high or low through them or read into the pin's level state.

Frequency Module: According to the vibration differential equation of vibration wire, the relationship between the stress and vibration frequency of steel wire can be deduced $f = \frac{1}{2} \times L \times \sqrt{\frac{\delta}{\rho}}$, That is, the square of frequency f is linearly proportional to the tensile stress of steel wire. Therefore, in actual measurement, the square value of the frequency is often used to directly reflect the stress value. However, since the square of the frequency tends to be large and difficult to read, the square of the frequency is generally used/100 “Frequency module.” A Frequency module is a calculated value derived from a frequency value rather than a measured value.

Signal Amplitude: In this manual, Refers to the signal amplitude of the signal returned by the vibration wire sensor after being filtered and amplified by the module, expressed in percentage.

Sample Quality: In this manual, the module evaluates the quality of the sample after multiple sampling of the sensor signal, expressed as a percentage. Also known as "sampled data quality assessment" or "sample data quality assessment."

Signal Quality: The comprehensive evaluation value of the collected signal. In this manual, the judgment standard is different according to the parameter setting, and it will be clearly explained when the specific function is explained.

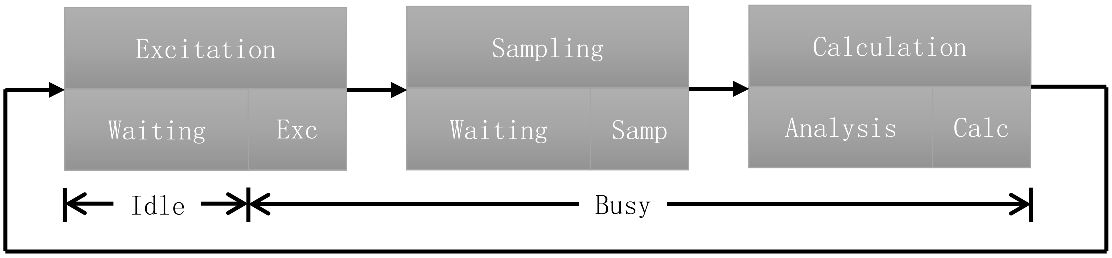
Register: A device that has the ability to temporarily or permanently store data. In this manual, a register that can be powered off is called a FLASH register, and the data it stores is called a “parameter”; a register that has data retention capability only during power-on is called a RAM register. The external interface can directly access the RAM register through the digital interface and indirectly access the FLASH register through the RAM register. The RAM register is divided into two types: read/write and read-only. The readable and writable part is read from the FLASH register at power-on, and the read-only part is responsible for writing real-time information during the running process. The write operation of the RAM readable and writable register through the digital interface triggers the copy operation (permanent storage) of the internal RAM register to the FLASH register of the module, and the read operation of any RAM register can acquire the current parameter and the running state data.

Table of Contents

1, Overview

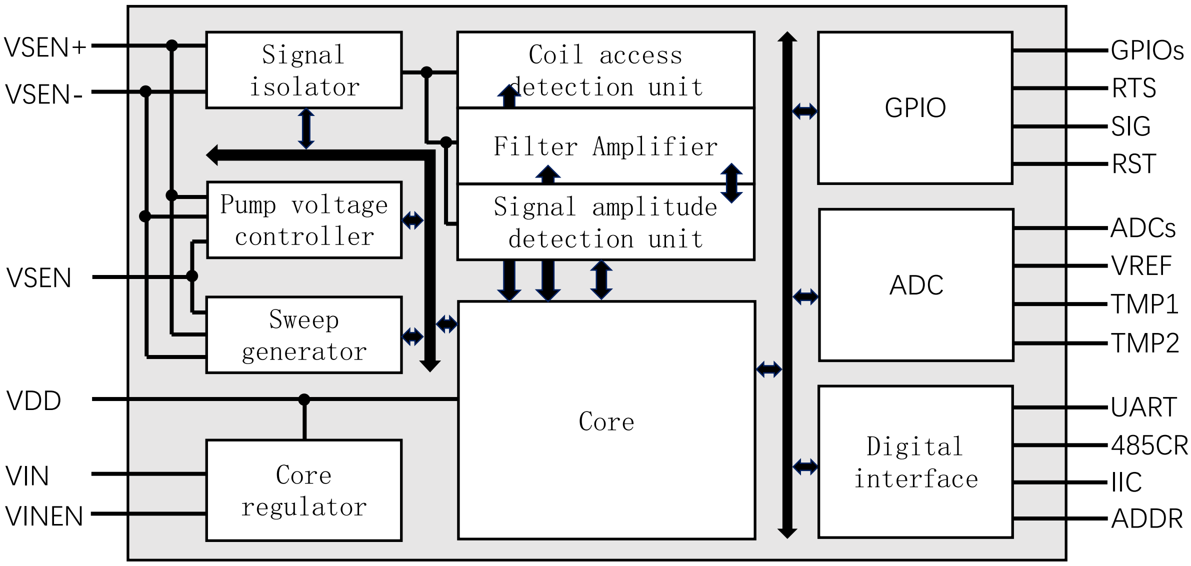
1.1 Product Overview

The VM5/6/7XX series modules are specialized reading modules for single-vibration wire sensor excitation, frequency reading and temperature conversion. They have outstanding features such as high integration, small size, high precision, strong adaptability and few peripheral circuit design. Advanced functions such as multiple excitation methods, sensor access detection, programmable excitation voltage, signal amplitude detection and signal quality assessment, can measure sensor signal quality, frequency, Frequency module, temperature,. VM5XX can be applied to the data reading of most single vibrating wire sensors at home and abroad, and has been widely used in civil engineering, automatic monitoring, geological disasters and other fields.



VM5/6/7XX workflow (main)

1.2 Functional Block Diagram



Functional Block Diagram for VM5/6/7XX

1.3 Absolute Maximum Ratings

parameter	Condition (remarks)	Minimum value①	Typical value	Maximum①	unit
Operating temperature		-40		85	℃
Storage temperature		-65		150	℃
VIN		-0.3		18	V
VSEN②		-0.3		12	V
VDD		-0.3		3.6	V
REF			VDD		
$V_{I/O}$		-0.3		VDD+0.3	V
I_{IN}				100	mA
Junction temperature				125	℃
Note 1: Stresses beyond this listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Note 2: Please ask the sensor manufacturer for the applicable excitation voltage. Excessive excitation voltage sources can cause permanent damage to the sensor or reading module.					

1.4 Recommended Conditions of Use

parameter	condition	Minimum value	Typical value	Maximum	unit
Operating temperature	VM5XXC	0	25	65	℃
	VM5XXI	-40	25	85	℃
V_{IN}		5.0	8.0 (recommended)	16	V
V_{SEN}		5.0	8.0 (recommended)	12	V
V_{DD}		2.5	3.3	3.6	V
I/O		0		VDD+0.3	V

1.5 Characteristics

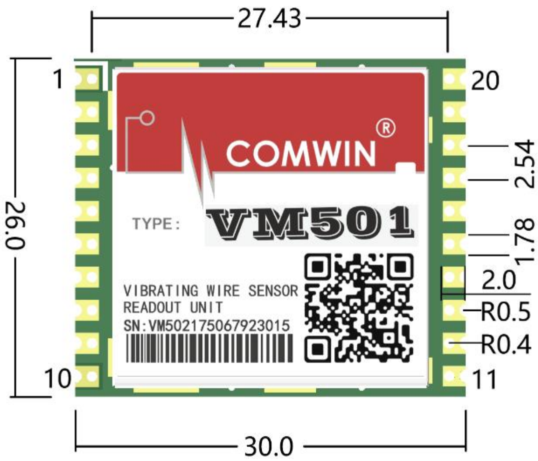
parameter	condition	Minimum	Typical	Maximum	unit
		power supply			
I_{VIN}①	normal operation	---	20	45	mA
I_{VDD}	Idle		30		mA

	Busy		50		mA
I_{VSEN} ①	idle		0		mA
	Busy (Voltage pumping)		10	15	mA
	Busy (sweeping)		20	40	mA
	Sleep		1.3		mA
Sensor excitation and reading					
Time base accuracy	@1000Hz		10^{-4}		
	@5000Hz		10^{-4}		
Frequency resolution	@1000Hz	0.02			Hz
Frequency measurement range		30		12000	Hz
Sweep output accuracy				0.05	%
High voltage excitation voltage		30	120	200	V
Temperature resolution			0.1		℃
Temperature measurement accuracy			0.5	1.5	℃
Random reading error (standard signal)	30~12000Hz	±0.001		±0.01	Hz
Absolute frequency error (standard signal)	300~6000Hz		±0.05	±0.25	Hz
Repeatability				0.01	Hz
Measuring speed			1.0	20.0	Hz
UART					
Communication rate		9600		1382400	bps
High level	TTL interface	1.2	3.3	3.6	V
Low level	TTL interface	0		0.8	V
The above data are tested at room temperature 25 ° C, VIN = 8.0 V, VSEN = 8.0 V, VDD = 3.3 V.					

1.6 Pin Definition

1.6.1VM5XX Pin Definition

VM501 pin definition



VM501 mechanical size

VM501/511 pin definition

symbol	Numbering		Types of	Description
	VM501	VM511		
SEN+	1	1		Connected to both ends of the vibrating wire sensor coil
SEN-	2	2		
TMP	3	3/20		Temperature sensor interface
DAO1	4	4		Analogue output
SIG	5	5		Signal quality indicator pin
REF	6			reference voltage input
IO2/SCL	7	6		SCL
DAO2		7		Analogue output
NC	8			
GND	9	9/10		
VDD	10	8		2.5~3.6V
GND	11	12		
TXD	12	13		UART transmit pin
RXD	13	14		UART receive pin
UDM/485CR	14			UART send instruction, can be used as 485 receive and receive control
UDP	15			
IO1	16	18		
RST/SDA	17	19		Parameter reset detection, active low
RTS	18	15		Busy indication
GND	19	16/22		
VSEN	20	17		sensor excitation power input
VCC		11		5.0~12V
NOTE:VCC and VDD Can't be used at the same time.				

1.6.2VM604/8 Pin Definition

VM604

VM608

VM604/608 pin definition

symbol	Numbering		Types of	Description
	VM604	VM608		
SEN-	1	1		Coils common pin
S1+	2	2	IO	Coil1
S2+	3	3	IO	Coil2
S3+	4	4	IO	Coil3
S4+	5	5	IO	Coil4
REF/TMP	6	6	I/O	604: Reference voltage input 608: Temperature sensor
IO2/SCL	7	7	IO	IIC-SCL
NC	8	8		
GND	9	9		
VDD	10	10		
GND	11	11		
TXD	12	12	O	
RXD	13	13	I	
RTS/IO1	14	14	IO	
RST/SDA	15	15	IO	IIC-SDA
T4/S8+	16	16	I/IO	604: Temperature sensor 4 608: Coil 8
T3/S7+	17	17	I/IO	604: Temperature sensor 3 608: Coil 7
T2/S6+	18	18	I/IO	604: Temperature sensor 2 608: Coil 6
T1/S5+	19	19	I/IO	604: Temperature sensor 1 608: Coil 5
VSEN	20	20		Excites the power input(5~12V)

1.6.3VM614/8 Pin Definition

S- S1+ S2+ S3+ S4+ IO2 RST VDD GND GND VIN

GND T1 T2 T3 T4 VSEN GND NC R/A T/B GND

VM614/714

S- S1+ S2+ S3+ S4+ IO2 RST VDD GND GND VIN

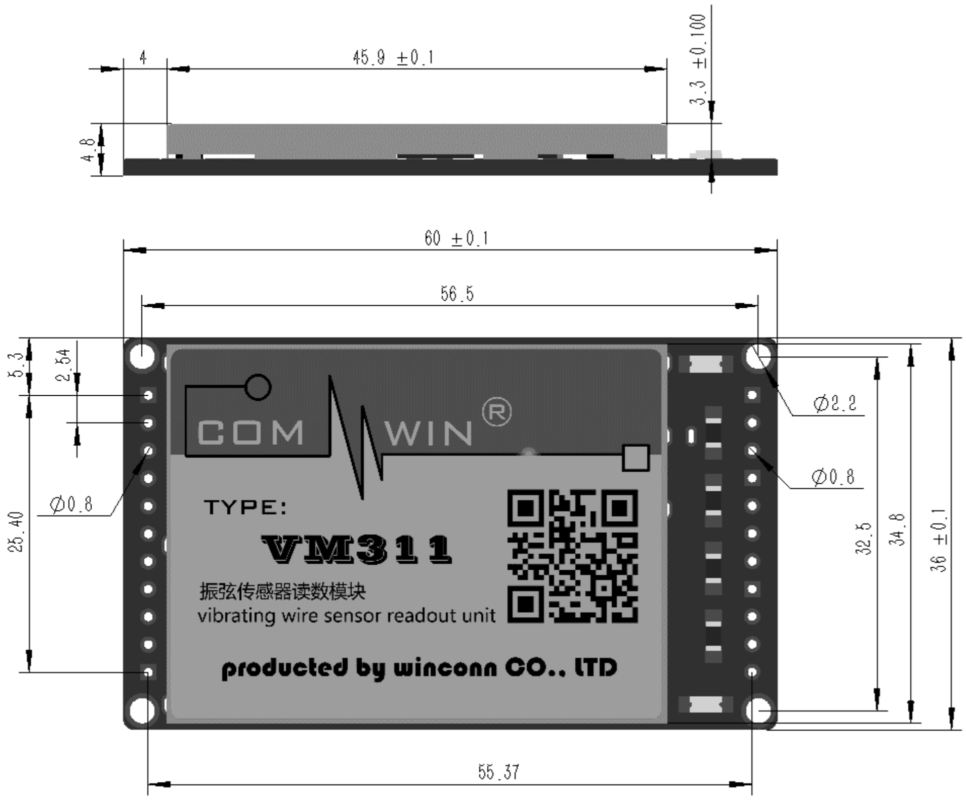
GND S5+ S6+ S7+ S8+ VSEN GND TMP R/A T/B GND

VM618/718

VM614/618 pin definition

symbol	Numbering		Types of	Description
	VM614	VM618		
SEN-	1	1	IO	
S1+	2	2	IO	
S2+	3	3	IO	
S3+	4	4	IO	
S4+	5	5	IO	
IO2/SCL	6	6	IO	
RST/SDA	7	7	IO	
VDD	8	8		
GND	9	9		
GND	10	10		
VIN	11	11		
GND	12	12		
TXD/B	13	13	IO	
RXD/A	14	14	IO	
NC/TMP	15	15	IO	
GND	16	16		
VSEN	17	17		
T4/S8+	18	18	IO	
T3/S7+	19	19	IO	
T2/S6+	20	20	IO	
T1/S5+	21	21	IO	
GND	22	22		

1.7 Package Size



VM511/614/618 mechanical size

2, Hardware Interface

2.1 Power Interface

The VM5XX module has multiple power interfaces: wide voltage power input (VIN), core power supply (VDD), vibrating wire sensor excitation power supply (VSEN). Each power supply shares GND.

Power Input (VIN): Wide voltage VIN pin powers the module (DC5~18V). The recommended voltage is 5.0V~8.0V. VIN can generate the core power VDD. When VIN is used to power the module, VDD pin for output, a regulated 3.3V output is available that can be used to power other circuits. The VDD output capability is 200mA. Try not to use VDD to output excessive current, so as not to affect the normal operation of the module core.

Core Power Supply (VDD): Generated by VIN. When VIN is not used, this pin acts as a power supply and requires an external DC3.3V supply. The supply voltage range is DC2.2~3.6V, and the recommended voltage is 3.3V. The module operates with a peak current of approximately 100 mA. It is recommended to use a voltage source with an output capability of 200 mA or more.

Excitation Power Supply (VSEN): VSEN supplies power to the sensor excitation process. It is recommended to supply VSEN with a voltage source of 50mA or more. The recommended supply voltage is DC8V~12V.

Please pay special attention to the design of the power supply. The vibrating wire sensor return signal is a weak sine wave. To reduce the influence of power supply ripple on the sensor signal, it is recommended that all power supplies use a LDO regulator with a small ripple. When using AC to DC power supply mode, the module ground wire (GND) must be reliably grounded (earth). Some low - end ac - dc adapters will introduce ac interference into dc, seriously affecting the signal processing quality of modules, or even completely out of use.

It is recommended to use a tantalum capacitor (10uF) and a ceramic capacitor (0.1uF) near the power pin to filter interference at high frequencies. At the same time, It is recommended to use a suitable zener diode (500mW) at the power input pin to prevent the chip from being damaged by the surge. For PCB layout, the capacitors and diodes should be as close as possible to the module's power input pins.

Note: It is strictly prohibited to use VIN and VDD to power the module at the same time. When using VDD to supply power to the module, it must be 3.3V.

Note: The module does not have reverse power supply and overvoltage protection measures. When the maximum allowable voltage is exceeded, the module will be permanently damaged.

2.2 Reset Pin

The module comes with a power-on reset function and does not require an external reset. The RST pin is a bidirectional pin and has different functions in different operating phases.

The RST pin is open-drain and has a 1kΩ pull-up resistor connected internally.

This module also has a software reset start method, which is implemented by writing 0x01 to the register FUN. See the subsequent section "3.2 Module Reset" for details.

2.3 Operating Status Indicator

2.3.1 Operation Status Indication

The RTS pin is used as the running status indicator when the module is running normally. The high level indicates that the module is “busy” and the low level indicates that the module is “idle”. For details, see “3.10 Sensor measurement process”.

The module "Busy" means that the module is reading the vibrating wire sensor. In particular, the module has a detection function for whether the sensor is connected. By default, the reading process is initiated only when a valid sensor access is detected. When the sensor connection is not detected, the module will continue to detect. At this time, the RTS pin continuously outputs a 10 Hz pulse square wave. This fast "busy" and "not busy" switching between the two states can be understood as "searching sensor".

Phases	Signal description	Status description
Power on	5 pulses of 100mS period	The module completes initialization and self-test and enters the normal operation phase.
normal operation	Sensor not connected Continuous 10Hz high and low pulse output	No valid sensor access detected Searching for sensors
	Sensor is connected High level: the module is "busy" Low level: module "idle"	Sensor is connected High and low switching frequency is related to multiple parameters Is exciting, sampling the sensor frequency, It is recommended to send an instruction to the module while RTS output to be low.
	Output 10ms high pulse every 3 seconds	The module is in single measurement mode and is waiting for instructions

The RTS pin is a strong push-pull output, which can directly drive the LED indicator to visually represent the working status of the module.

2.3.2 Hardware Handshake Signal

Based on the time domain characteristics of the RTS output signal, this pin can also be used as a hardware handshake signal for the digital interface. When the module's UART interface is RS232, the RTS pin has been converted to an RS232 level signal, which can be directly connected to the CTS of the host computer's RS232 interface.

2.4 Signal Quality Indication

The SIG pin is used to output the return signal quality of the vibrating wire sensor. When the signal quality reaches or exceeds the expected value, it outputs a high level, otherwise it outputs a low level. The expected signal quality is defined by the register EXS_TH. See “Description of the Predetermined Signal Quality Register in 3.12.4” for details.

2.5 Digital Interface 1 (UART/RS232/RS485)

The VM5XX provides a full-duplex serial TTL level UART interface. The default port setting is "9600, N, 8, 1" and is supported by software modification to 9600~ 1382400bps.

The UART's TTL level logic high is VDD, and the logic low is GND. When connecting to a non-3.3V microcontroller, pay attention to the logic level conversion.

TXD is a strong push-pull output pin and RXD is an input pin.

Pin 485CR is the data transceiver indication pin. When the module sends data out, the pin 485CR outputs a high level (strong push-pull), and when it is not transmitting, it outputs a low level. With this logic feature, when the RS485 level conversion chip is connected outside the UART, the transmission indication pin can be used as the transceiver control pin of the half-duplex 485 chip. When the module is an RS485 interface version, the 485CR pin is connected to the 485 chip (VM511/614/618 only) inside the module.

2.6 Digital Interface 2 (IIC)*

The VM5XX supports IIC bus and data transfer protocols and supports communication rates up to 500 kHz.

VM5XX works as a slave device on the IIC bus. The device is connected to the bus through the SCL and SDA lines. Both data lines are open-drain. When using the IIC interface, an external 4.7k pull-up resistor is required, which is performed with a non-3.3V microcontroller. When connecting, pay attention to the conversion of logic levels.

2.6.1 Device Address

VM501 has a default address of 0xA0 and can be modified through the UART instruction.

2.6.2 IIC Protocol Description

VM501 faked itself into an AT24C02 chip. The register reads and writes exactly the same as AT24C02.

2.7 Sensor Coil Interface

The sensor coil interface consists of two pins, SEN+ and SEN-, which are respectively connected to the two ends of the vibrating wire sensor coil. On the one hand, the sensor coil interface transmits the sensor excitation signal to the sensor, so that the internal steel wire of the sensor starts to vibrate; the weak return signal is transmitted to the filter amplifier circuit in the module for further processing.

Normally, the sensor coil does not distinguish between positive and negative electrodes and can be directly connected.

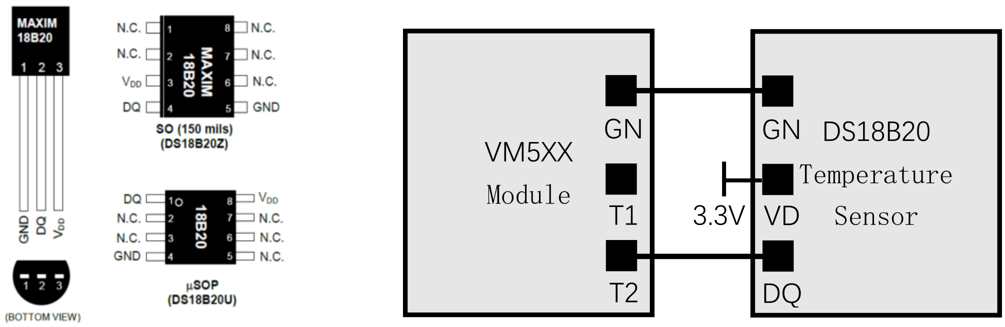
Note: When the sensor excitation mode is set to high voltage excitation, the SEN+ pin will output high voltage periodically (hundreds of milliseconds to several seconds), and attention should be paid to the protection of personnel and peripheral circuits during use.

2.8 Temperature Sensor Interface

The temperature sensor interface consists of two pins, TMP and GND, and a 4.7K pull-up resistor is connected internally. The temperature sensor interface is a multiplexed interface, which can be connected to the digital temperature sensor DS18B20 or the thermistor temperature sensor. At any time, only one type of temperature sensor can be connected (For details, see “3.18 Temperature sensor use").

2.8.1 Connection of digital temperature sensor DS18B20

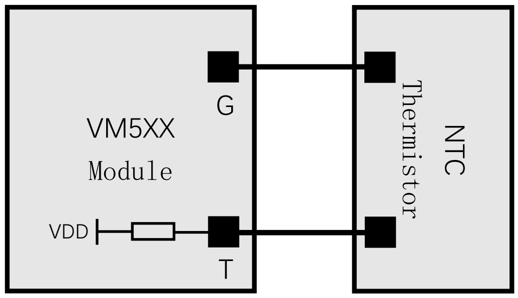
As shown in the figure below, the DS18B20 has three pins, GND, VDD, and DQ. The three pins should be connected to the VM5XX module GND, VDD, and TMP.



DS18B20 digital temperature sensor connection diagram

2.8.2 Thermistor Temperature Sensor Connection

Connected between the two ends of the thermistor and the GND of the VM5XX module, as shown in Figure a below.



Connection thermistorr

2.9 Reference Voltage

Pin VREF is the reference voltage input, and the module has an internal calibration mechanism, so it is not necessary to connect to the special reference voltage reference, but directly to VDD. The pin must be connected and cannot be suspended.

3, Start To Use

The default parameter value of the module can meet the data reading of most of the vibration wire sensor. There is no special case and no need to modify the parameters. Just read the measurement result from the real-time measurement result register. If you need to modify some parameters, be sure to know the working principle of the module and the range of values of the parameters. The wrong parameter values may cause the module to fail to work properly. If necessary, use the parameter reset function to restore the parameters to the factory defaults.

3.1 Startup

3.1.1 Startup Information

The module is powered on and started by itself. After the initialization is completed, the following startup information is output (UART interface).

VM5XX	// Module series name
HW:1.10	//hardware version number
SF:1.01-170626-031	//firmware version number
Addr:001	// Module address
SN=XXXXXXXXXX	// Module machine code (serial number)

3.1.2 Startup Process

(1) Read the stored working parameters and perform parameter verification. If the error is checked, the factory value will be restored automatically;

(2) If the RST pin is low, the recovery register value is the factory value;

(3) Outputting startup information via the UART interface;

(4) Enter the normal working phase according to the working mode defined by the register.

Depending on whether the module is reset and the communication rate is different when the module is powered on, the power-on startup process is 50~100ms. It is recommended to initiate data communication operation to the module after 100ms after the module is powered on.

3.1.3 Version Information and Serial Number

Write the function code 03 to the system function register FUN. The module returns the firmware version information and the unique serial number. For the format of the output information, see “3.1.1 Startup Information”.

3.2 Reset (restart)

The following conditions (or operations) can cause the module to generate a reset action and restart.

(1) Soft reset: during the normal operation of the module, send a soft reset instruction 0x01 to the register FUN;

(2) The VDD voltage is too low or subject to strong electromagnetic interference;

(3) Unknown illegal parameter setting, resulting in abnormal work;

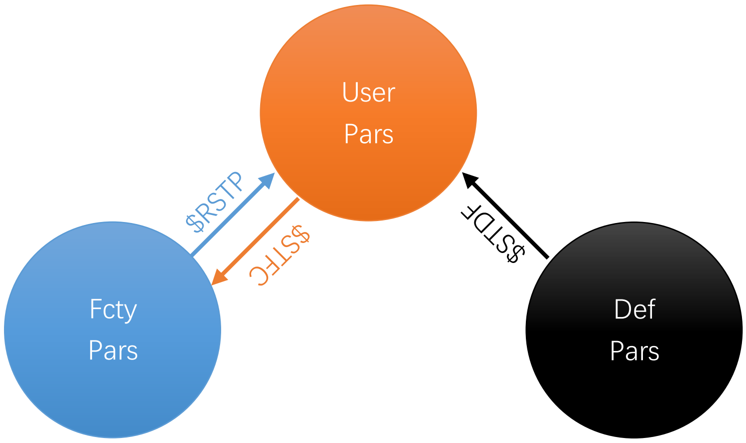
3.3 Restore Factory Parameters

Three types of system parameters are stored inside the device: user parameters, factory parameters and default parameters.

user parameters: Also known as "working parameters", it can be modified and saved, and automatically loaded every time it is powered up. Modification of parameters and operation logic of devices in the process of using devices refer to user parameters, and user parameters are the most frequently used parameter categories.

Factory parameters: Parameters stored in separate partitions. When the "restore factory parameters" instruction is received, the factory parameters are loaded and the user parameters are overwritten. In the process of device startup, if abnormal user parameters are detected, the device will automatically load the factory parameters and cover the user parameters. These parameters has been set by the manufacturer, it is not recommended to modify it (use with caution).

Default parameters: The default parameter is a set of fixed parameters, which can only ensure the communication of the device. The user cannot change the default parameters in any way. When factory parameters are restored, the default system parameters will be automatically loaded if communication parameters are found wrong, so that basic digital communication can be carried out.



3.3.1 Parameters to factory

There are two ways to restore all parameters (registers) to factory values.

(1) Hardware parameter reset: When the module starts, it detects that the RST pin is low, and the reset parameter is the factory default value. The module is in the pause state until the pin level is not restored to the high level.

(2) Software parameter reset: Write 0x02 to register FUN.

Automatically restore factory parameters

In the following cases, the module will automatically return to the factory parameters.

(1) Parameter CRC error: During the power-on process, the parameter area check code error is detected, and the factory parameter value is automatically restored. The UART outputs the prompt message “CRC Err\r\n”.

(2) UART communication rate error: During power-on, the parameter BAUD value is detected as an illegal communication rate value, and is automatically restored to the factory parameter value. The UART uses the default rate to output the prompt message “BAUD Err\r\n”.

3.3.2 Modify factory parameters

Writes to the factory parameter area with the current user parameters. ***This operation is recommended for professional use, ordinary users do not easily modify factory parameters.***

Software method:

$STFC\r\n

Return string after device response: OK\r\n

3.3.3 Restore default parameters

Load the device's pre-set fixed parameters into user parameters. The instructions are as follows:

$STDF\r\n

Return string after device response: OK\r\n

3.4 Communication Protocol

Communication protocol is a series of predetermined data format, transmission steps, communication rate and so on for information interaction between the upper computer and the digital interface of VM5XX module. The host computer must complete the data interaction with VM5XX according to the communication protocol rules described in this chapter.

Register mechanism

VM5XX internal maintenance has several registers, and the module completes the measurement of the vibrating wire sensor under the control of the register parameter values. The value of the register always exists as an integer. The basic operation unit is "word" (2-byte integer, big-end mode). There are two types of power-off saving and power-off loss (corresponding to the two attributes of "read/write" and "read only"). Reading and writing (modification) of registers can be completed through the digital interface of the module. The register write (modify) lifetime is typically 100,000 times, and there is no limit to the number of reads.

Data mode

The register data value is in big endian mode, the high byte of the data is stored in the low address of the memory, and the low byte of the data is stored in the high address of the memory. When the data frame is transmitted, the low byte is transmitted first and then the high byte is transmitted. . Each register corresponds to two bytes, then the value of a single register = low byte value * 256 + high byte value.

Handshake protocol

When reading and writing registers, it is recommended to use the handshake protocol of the VM5XX module (not essential). When the module is detected to be idle, the register operation is initiated (For hardware handshaking, see "2.3.2 Hardware Handshake". For details on the software handshake, see "3.7.2 Software Handshake"). It is strictly forbidden to disconnect the module power supply during the process of modifying the register. In severe cases, all parameters will fail and cannot operate normally.

Conflict resolution

When the module receives the command from the host computer, it will immediately process and return the response message. If the module receives the command from the digital interface when it is “busy”, the VM5XX adopts the principle of measurement priority and waits for the end of the current measurement cycle. Execute the instruction and output the reply message, so the phenomenon that the module does not respond to the upper computer command in time often occurs. Different excitation methods and delayed read parameters will have different "busy" durations (see "3.15 Measurement Duration and Optimization" for details). After the command is sent, if the module fails to respond in time, it should wait at least according to the actual parameter values. A "busy" duration, and then send the next instruction to the module. It does not make sense to send multiple instructions to the module before the module has returned. The module will only respond to the first instruction received.

When the UART instruction is received when the module is idle, it will immediately exit the idle state and trigger a sensor measurement process.

If a busy digital interface is detected, the next measurement will not be carried out until the end of communication.

3.4.1 UART Communication Protocol

The UART interface supports standard industrial MODBUS communication protocols (03, 06, 16 instruction codes) and a custom simple AABB protocol. Both protocols support "One host multiple slave" application structure based on module address and bus connection. In the bus, the VM5XX module is always used as a slave (Passively wait for instructions, do not upload data actively, except "auto upload data" and "software handshake", as detailed in the following corresponding chapters).

It is recommended to use the dedicated VMTool tool for register command generation and testing. For the basic usage of VMTool, please refer to "Chapter 4: Using the Parameter Configuration Tool".

The UART interface adopts the interrupt mechanism to receive data and takes the idle time as the judgment criterion for the completion of a frame of data reception. The receiving idle time is fixed to 10mS. The timing is started when the last byte is received, If more than 10mS does not receive new data, the data transmission of this frame is considered to be over, and the time interval of 10mS or all received during the module "busy" is considered, The data is considered to be the same frame data. Inside the module, an 80-byte buffer is created for UART reception. The first byte of each frame of data is stored at the beginning of the buffer. When the length of one frame exceeds the number of buffer bytes, the newly received data is forcibly stored in the last byte position of the buffer.

The registers can be read and written according to the following communication protocol rules. When using the MODBUS or AABB communication protocol, please confirm that the module's software handshake is off (default). Also pay attention to the impact of frequent data communication on measurement speed.

(1) MODBUS communication protocol

Under the MODBUS protocol, all registers in VM5XX are defined as "hold registers" (see MODBUS communication protocol standard description). The module supports multiple consecutive register reads, single register writes, multiple consecutive register writes. Three kinds of instructions, the corresponding instruction codes are 0x03, 0x04, 0x06, 0x10. The instruction and return data frame format of each instruction code are explained one by one below.

(1) 03 (0x03) ,04(0x04) instruction code: read multiple consecutive register data, the instruction format is as follows

Instruction data frame structure

address code	Function code 0x03	Start address	Number of registers	CRC check
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Return data frame structure

address code	Function code 0x03	Data length	data	CRC check
1 byte	1 byte	1 byte	n bytes	2 bytes

Example: Read the module register value with address 0x01, the register start address is 0, and 10 registers are read continuously.

The host sends the command: 0x01 0x03 0x00 0x00 0x00 0x0A 0xC5 0xCD

The slave return: 0x01 0x03 0x14 <u>0x00 0x01 0x00 0x60 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x01 0xF4 0x00 0x00 0x00 0x64 0x00 0xC8</u> 0x5F 0x8F (underscore is the 10 register values)

The host sends the command: 0x01 0x04 0x00 0x00 0x00 0x0A ***0x70 0x0D***

The slave return: 0x01 0x04 0x14 <u>0x00 0x01 0x00 0x60 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x01 0xF4 0x00 0x00 0x00 0x14 0x14 0xC8</u> ***0xB7 0x62*** (underscore is the 10 register values)

When reading multiple consecutive registers, do not exceed 64 registers in a single read. Do not attempt to read a non-existing register (a register with an address greater than 63).

(2) 06 (0x06) instruction code: modify the value of a single register, the instruction format is as follows

Instruction data frame structure

address code	Function code 0x06	Register address	Register value	CRC check
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Return data frame structure

address code	Function code 0x06	Register address	Register value	CRC check
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example: Modify the value of register 8 in the module with address 0x01 to 100.

The host sends the command: 0x01 0x06 0x00 0x08 0x00 0x64 0x09 0xE3

The slave return: 0x01 0x06 0x00 0x08 0x00 0x64 0x09 0xE3

(3) 16 (0x10) instruction code: modify the value of consecutive multiple registers, the instruction format is as follows

Instruction data frame structure

address code	Function code 0x10	initial address	Number of registers	Number of bytes	Register value	CRC check
1 byte	1 byte	2 bytes	2 bytes	1 byte	n bytes	2 bytes

Return data frame structure

address code	Function code 0x06	initial address	Number of registers	CRC check
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example: Modify the value of the 0~31 register in the module with address 0x01

The host sends a command (hexadecimal command): 01 10 00 00 00 20 40 00 01 00 60 00 00 00 03 00 00 00 01 01 F4 00 00 00 C8 C8 C8 00 01 82 35 00 05 03 E8 00 A0 05 DC 06 40 00 05 00 64 00 00 00 0A 00 0A 00 04 01 77 03 15 00 00 00 00 03 E8 00 01 00 00 00 00 00 00 A3 70 (underlined data is 32 register values)

The slave return: 0x01 0x10 0x00 0x00 0x00 0x20 0xXX 0xXX

The UART receive buffer of VM5XX is 80 bytes. When sending a multi-register write command to the module, the length of the single-frame instruction should not exceed the limit.

Modules with firmware version lower than 3.01 do not support continuous multi-register writing instructions, which need to use a single register to modify instructions.

(2) AABB Communication protocol

AABB communication protocol is a non-standard custom protocol. Compared with MODBUS communication protocol, the structure is simpler and the instruction generation method is easier, which is convenient for quick test. The AABB communication protocol supports single register reads and writes two instructions.

(1) Read a single register

Instruction data frame structure

Command head	address code	Register address	Add check

0xAA 0xBB			
2 bytes	1 byte	1 byte	1 byte

Return data frame structure

Command head 0xAA 0xBB	address code	Register address	Register value	Add check
2 bytes	1 byte	1 byte	2 bytes	1 byte

Command header: fixed to hexadecimal AABB

Address code: The address of the module (1~255, where address 255 is the general address, see the description of the following "Universal Module Address" for details)

Register address: The address of the register to be accessed (0~63). The highest bit of the register address byte is the read/write flag. When it is 0, it indicates the read register. When it is 1, it indicates the write register.

Add check: the sum of all previous data, 0xAA + 0xBB + address code + register address, when the checksum exceeds 255, only the low byte is used. In the following example, the checksum = 0xAA + 0xBB + 0x01 + 0x08 = 0x016E, then only 0x6E is used as the final checksum.

Example: Read the module register value with address 0x01, register address is 8

The host sends the command: 0xAA 0xBB 0x01 0x08 0x6E

The slave returns a response: 0xAA 0xBB 0x01 0x08 0x00 0x60 0xCE

(2) Modify a single register

Instruction data frame structure

Command head 0xAA 0xBB	address code	Register Address|0x80	Register value	Add check
2 bytes	1 byte	1 byte	2 bytes	1 byte

In the write register instruction, the most significant bit of the register address byte should be 1, that is, the address value is “OR” with 0x80.

Return data frame structure

Command head 0xAA 0xBB	address code	Register address	Register value	Add check
2 bytes	1 byte	1 byte	2 bytes	1 byte

Example: Modify the module register value with address 0x01, the register address is 8, and the modified value is 100.

The host sends the command: 0xAA 0xBB 0x01 0x88 0x00 0x64 0x52

Slave return response: 0xAA 0xBB 0x01 0x08 0x00 0x64 0xD2

(3) Universal module address

The AABB communication protocol supports the module's general address.

Regardless of the current address of the module, using 0xFF as the address to send read and write instructions to the module, the module can respond correctly.

Example: Use the general address to read register 8 of any module

The host sends the command: 0xAA 0xBB 0xFF 0x08 0x6C

The slave returns a response: 0xAA 0xBB 0x01 0x08 0x00 0xC8 0x36

Note: When multiple modules are connected to the bus (usually RS485 bus), all modules on the bus will respond to the command when the general address is used, resulting in the instruction not working properly.

Note: It is strictly forbidden to use a general address to modify the module address in a bus connected to multiple VM5XX modules.

(4) Special module address

The module address is stored in the register ADDR.[7:0], and the value ranges from 1 to 255. Among these addresses, 255 is used as the general address in the AABB protocol, and the address 128 (0x80) is used for special purposes. Therefore, The addresses that can be used are: 1~127, 129~254, a total of 253.

Module address register (0x00)

Bit	symbol	value	description	Reset value
bit15:8			Not yet defined	0
bit7:0		1~254	Module address	1

(3) Check code algorithm

Whether the instruction is sent to the module or the response data returned by the receiving module, the data verification should be strictly performed. In rare cases, there will be errors in the response data returned by the module, and the verification of the data frame can completely avoid reading the wrong data.

(1) CRC16-MODBUS algorithm

```
unsigned int crc16(unsigned char *dat, unsigned int len)
{
  unsigned int crc=0xffff;
  unsigned char i;

  while(len!=0)
  {
      crc^=*dat;
      for(i=0;i<8;i++)
      {
          if((crc&0x0001)==0)
              crc=crc>>1;
```

```
            else
            {
                crc=crc>>1;
                crc^=0xa001;
            }
          }
          len-=1;
          dat++;
        }
    return crc;
  }
```

(2) and check algorithm

```
  unsigned char AddCheck(unsigned char *dat,unsigned char count)
  {
    unsigned char i,Add=0;
    for (i=0;i<count;i++)
        Add+=dat[i];
    return Add;
  }
```

3.4.2 Communication Protocol (IIC)*

The IIC communication protocol itself is the physical layer communication protocol based on the device address and register. The VM5XX uses the IIC interface to access the sensor. Please follow the above hardware interface timing and protocol description.

3.4.3 Actively Uploading Measurement Data (UART)

By default, the VM5XX module always performs data interaction with the host as a slave. In this master-slave structure, VM5XX never actively uploads data. The module can actively output measurement data by modifying the automatic upload register (ATSD_SEL). Each bit of the ATSD_SEL register corresponds to a data type, as shown in the following table.

Automatic upload register ATSD_SEL (0x07)

Bit	symbol	value	description	Reset value
bit15		0/1	System running error code, data prefix "$ER"	0
bit14		0/1	Sensor coil resistance value, data prefix "$RE"	0
bit13		0/1	Sample quality, data prefix "$QU"	0
bit12		0/1	Frequency value, data prefix "$FR"	0
bit11		0/1	Frequency modulus value, data prefix	0

			"$FM"	
bit10		0/1	Temperature value, data prefix "$TE"	0
bit9		0/1	Input voltage / ADC01, data prefix "$IV"	0
bit8		0/1	Excitation voltage, data prefix "$SV"	0
bit7		0/1	Current sweep frequency value, data prefix "$SF"	0
bit6:2		0/1	Undefined function	
bit1		0/1	Sampled data, data prefix "$TS" "$TM"	0
bit0		0/1	Real-time signal amplitude data, data prefix "$AV"	0
Note: Active uploading of data destroys the master-slave communication mechanism. In order to avoid conflicts caused by simultaneous uploading of data and response data of the host command, the automatic sending will be suspended for 5 seconds after receiving the host command.				

When a bit is set to 1, the module actively uploads data through the UART interface. In addition to the real-time signal amplitude data, other data is automatically uploaded as a string after each sensor frequency calculation and temperature acquisition. The automatically uploaded data starts with "$" and the string structure is: $aa=bbbb-[cc]-[dd]\r\n

$: fixed symbol

aa: data type identifier, see the description of the above table

=: fixed symbol

bbbb: data value

cc: data unit, no unit data is empty

dd: additional information, when it is the real-time amplitude data of the signal, dd represents the data index value

\r\n: carriage return

Real-time signal amplitude active upload

The automatically uploaded real-time amplitude data transmission interval is 100ms, and the real-time amplitude data of the signal is actively output at a rate of about 10 Hz (if ATSD_SEL.[0] is 1).

Data format (string): $AV=AAA%BBBB\r\n

$AV: fixed identification - real-time signal amplitude

AAA%: The percentage value of the signal amplitude. The digital part is fixed to 3 digits. When it is insufficient, it is 0.

BBBB: Signal amplitude index number (value), zeroed each time the measurement starts, up to 9999.

Signal amplitude data example

$AV=65%07\r\n$AV=65%08\r\n$AV=60%09\r\n$AV=……$AV=23%230\r\n

Note: When the real-time amplitude active upload function is enabled, try not to send other commands to the module (except for turning off the automatic upload command). It is strictly forbidden to call time-consuming functions (such as reading version information, reading parameters, saving parameters, etc).

Sampling dataset active upload

The active upload of the sampled data coincides with the sensor frequency value calculation process (if ATSD_SEL.[1] is 1), and all the original values sampled are continuously output using one frame of data, separated by "|".

Sampled data

$TM=000001\r\n	Data multiplier, all data below * multiplier = true value
$TS=xxxx.x|xxxx.x|……\r\n	

Other data is actively uploaded

In addition to the above two types of automatic upload data, other automatic upload data is output after the sensor frequency value calculation is completed (if ATSD_SEL.[x] is 1).

Example

$FR=1234.5Hz\r\n	Current sensor frequency value is 1234.5Hz
$FM=15239.9\r\n	Current sensor frequency modulo value is 15239.9, no unit
$TE=28.6'C\r\n	Current temperature value is 28.6 °C

Automatic upload conflict resolution with master-slave protocol

When the automatic upload function is set and the MODBUS or AABB protocol command of the master-slave structure needs to be sent to the module, the module adopts the principle of priority of the master-slave protocol. In the process of automatic transmission, if the master-slave instruction is received, the automatic upload is immediately suspended. 5 seconds. A phenomenon caused by this is that when the modified automatic transmission register (master-slave protocol instruction) is sent to the module, the automatic upload data of the module is not immediately received, but is received after about 5 seconds.

3.5 Register Overview

The working process of the VM5XX module is completely dependent on the register (parameter) value. The register is a 16-bit binary representation of an integer, which is divided into a readable and writable register and a read-only register. The readable and writable registers are further divided into a power-down save and a power-on reset. Types can be accessed through the UART or IIC digital interface to modify various parameters of the module to achieve the purpose of controlling the module and interacting with the module.

The following summary table lists all register and summary function descriptions. More detailed register usage will be specified in the following functions.

Different firmware versions may have slightly different definitions of registers. Before operating the registers, verify that the firmware version corresponds.

Register Summary Table (HW3.12)

address	symbol		name	Def	unit
0x00(0)	ADDR	R/W/S	Module address	1	
0x01(1)	BAUD①	R/W/S	Communication rate	96	100bps
0x02(2)	AUX	R/W/S	Auxiliary function register	5	
0x03(3)	SYS_FUN	R/W/Re	System function	0	
0x04(4)					
0x05(5)	WKMOD	R/W/S	work mode	1	
0x06(6)	MM_INTE	R/W/S	Continuous measurement interval	500	ms
0x07(7)	ATSD_SEL	R/W/Re	Automatic upload	0	
0x08(8)	RD_INTE	R/W/S	Delayed sampling	200	ms
0x09(9)	RD_COUNT	R/W/S	Expected number of samples	200	
0x0A(10)	EX_METH	R/W/S	excitation method	1	
0x0B(11)		R/W/S	internal use		
0x0C(12)		R/W/S	internal use		
0x0D(13)	HP_DUR	R/W/S	Pumping time	500	ms
0x0E(14)	HP_EXP	R/W/S	Expected voltage	130②	V
0x0F(15)	FS_FMIN	R/W/S	Sweeping lower limit	1500	Hz
0x10(16)	FS_FMAX	R/W/S	Sweeping upper limit	1600	Hz
0x11(17)	FS_STEP	R/W/S	Sweep stepping	5	Hz
0x12(18)	FS_SCNT	R/W/S	Sweep single step cycle		0xC814
0x13(19)	FIT_TYPE	R/W/S	Software filtering method	0	
0x14(20)	FIT_COUNT	R/W/S	Number of filtered	30	
0x15(21)	CAL_PAR1	R/W/S	Gross error rejection parameter factor	10	

0x16(22)	CAL_PAR2	R/W/S	Quality sample limit factor	4	
0x17(23)	AMP①	R/W/S	Signal amplification	15	
0x18(24)	FSG_TH	R/W/S	Feedback progressive frequency upper and lower limits	0x1414	Hz
0x19(25)	DAO_TH	R/W/S	analog output		100Hz
0x1A(26)	TEMP_PAR1①	R/W/S	Temperature calculation parameter 1	0	
0x1B(27)	TEMP_PAR2①	R/W/S	Temperature calculation parameter 2	100	0.01
0x1C(28)	TEMP_EX①	R/W/S	Temperature sensor setting	1	
0x1D(29)	EXS_TH	R/W/S	Signal quality limit	80	%
0x1E(30)	SIG_TH	R/W/S	Upper and lower limits of signal amplitude	0x6400	%, %
0x1F(31)	CRC	R	Parameter CRC check		
0x20(32)	SYS_STA	R/W/Re	System status register	0	
0x21(33)	SFV	R	Current sweep frequency	0	Hz
0x22(34)	SMP_QUA	R	Quality sample quality rating	0	%
0x23(35)	S_FRQ	R	Sensor frequency value	0	0.1Hz
0x24(36)	F_REQM	R	Frequency modulus high 16 bits	0	$100Hz^2$
0x25(37)			Frequency modulus low 16 bits		
0x26(38)	V_POW/ADC1	R	Input supply voltage / ADC01	0	0.01V
0x27(39)	S_RES	R	Coil resistance	0	Ω
0x28(40)	V_SEN	R	Real-time excitation voltage	0	0.01V
0x29(41)	TEMP	R	Temperature value	0	0.1℃
0x2A(42)	SMP_STD	R	Sample standard deviation	0	Hz
0x2B(43)	HQ_COUNT	R	"Quality" sample size	0	
0x2C(44)	SIG_VAL	R	Signal amplitude	0	%, %
0x2D(45)					%, %
0x2E(46)	GPIO	R/W/Re	GPIO register	0	
0x2F(47)	ADC02	R	AD value	0	
0x30(48)	ADC03	R	AD value	0	

0x31(49)	ADC04	R	AD value	0	
R: readable; W: can be modified; S: power is not lost; Re: reset to default after power-on Do not write non-zero data to reserved or undefined bits and registers. ①: These registers (parameters) take effect the next time they are started ②: The actual value is 0x8082, that is, the expected voltage function is enabled, and the expected voltage is 120V.					

3.6 Address Operation (UART)

Change the module address with address 0x01 to 0x02

3.6.1 Modifying the Address of a Known Device Address

Module address 0x01

MODBUS instruction: 0x01 0x06 0x00 0x00 0x00 0x02 0x08 0x0B

AABB instruction: 0xAA 0xBB 0x01 0x80 0x00 0x02 0xE8

3.6.2 Reading the Address of an Unknown Device Address

Modules with unknown addresses can only use the general address 0xFF of the AABB protocol.

AABB instruction: 0xAA 0xBB 0xFF 0x00 0x64

3.6.3 Modifying the Address of an Unknown Device Address

AABB instruction: 0xAA 0xBB 0xFF 0x80 0x00 0x02 0xE6

After modifying the module address, the new address takes effect immediately, and the part of the module address code in the received response message is the new address value. Subsequent instructions should operate on the module with the new address.

3.7 Communication Rate and Software Handshake (UART)

3.7.1 Communication Rate

The UART interface of VM5XX supports the communication rate of 9600~1382400bps. The communication rate is changed by setting the register BAUD.[13:0]. The unit of BAUD.[13:0] is “hundreds bits per second” or “hundreds of bps”. The value of the register and the corresponding communication rate are as follows.

UART communication rate register BAUD (0x02)

Bit	symbol	value	description			Reset value
bit15			Whether to enable the software handshake protocol function			0
bit14			Whether to ignore the "measure is busy" flag and immediately respond to the command			0
bit13:0		value	rate	value	rate	
		96	9600bps (def)	1152	115200bps	96
		128	12800bps	1280	128000bps	
		144	14400bps	1536	153600bps	

		192	19200bps	2304	230400bps	
		288	28800bps (NR)	2560	256000bps	
		384	38400bps (NR)	4608	460800bps	
		560	56000bps (NR)	9216	921600bps	
		576	57600bps (NR)	13824	1382400bps (NR)	
		768	76800bps (NR)			
The communication rate takes effect at the next startup, and the illegal communication rate value will cause the parameter to be restored to the factory default value. NR is not recommended						

When conditions permit, try to use a higher communication rate and shorten the data transmission time. Before applying a higher communication rate, please confirm whether the host computer communicating with VM5XX has high-rate communication capability. In general, 9600bps is the most common communication speed supported by all serial devices; Most mainstream serial devices work well in 115200bps; a few serial devices support 921600bps; and 1382400bps has exceeded the communication capabilities of most serial devices (including computers and some high-end USB to serial data lines), try not to use.

3.7.2 Software Handshake

When the module starts a measurement, the XOFF signal (0x13) is actively sent from the UART interface, indicating that the module starts to be busy measuring data. When the measurement is completed, the XON signal (0x11) is actively sent, indicating that the module is in the idle state after the measurement is completed. In a bus application with one master and multiple slaves, it is forbidden to turn on the software handshake function of the module.

After turning on the software handshake function of the module, if it is necessary to send an instruction to the module, it is recommended that the communication flow of the UART be: first wait for the module to return XON signal (0x13), and send an instruction to the module immediately after receiving the XON signal or waiting for timeout.

3.7.3 Modifying the Communication Rate of a Known Device Address

Module address 0x01, the baud rate is modified to 115200bps

MODBUS instruction: 0x01 0x06 0x00 0x01 0x04 0x80 0xDB 0x6A

AABB instruction: 0xAA 0xBB 0x01 0x81 0x04 0x80 0x6B

3.7.4 Reading the Communication Rate of an Unknown Device Address

Modules with unknown addresses can only use the general address 0xFF of the AABB protocol.

AABB instruction: 0xAA 0xBB 0xFF 0x01 0x65

Module return: 0xAA 0xBB 0x01 0x01 0x00 0x60 0xC7

3.7.5 Modifying the Communication Rate of an Unknown Device Address

AABB instruction: 0xAA 0xBB 0xFF 0x81 0x04 0x80 0x69

After modifying the UART communication rate, the new value will take effect at the next startup. The module will continue to use the communication rate before modification before restarting. After the restart, the new communication rate should be used to communicate with the module.

Except for the above module address and communication rate register, all other registers are accessed in the same way. This manual will no longer give examples.

3.8 System Status

3.8.1 Working Status

The VM5XX module has three working states, namely idle state, busy state, and sleep state, and the module automatically completes switching of several states.

Busy state: The module is performing sensor excitation or sampling the frequency data returned by the sensor. In the busy state, the pin RTS outputs a high level. During this period, try to avoid accessing modules during this time through the digital interface.

Idle state: The "non-busy" status of the module can be considered as "idle".

Sleep state: The module core sleeps to achieve lower current consumption, and both the UART and IIC instructions can wake the module from sleep. (This feature is not enabled yet)

3.8.2 Running Status

The register SYS_STA contains the status information during the running of VM5XX. By reading this register, you can get the current working status of the module and whether some error has occurred. The system status register only writes some specific position 1 when conditions are met. After reading and acquiring the status register, write 0 to it to clear the status flag.

Run status register SYS_STA (0x20)

Bit	symbol	value	description	Reset value
Bit15		0/1	No valid coil detected	0
bit14:7		0	Reserved, undefined function	
Bit6		0/1	Sweep timeout	0
bit5		0/1	Frequency overflow	0
bit4		0/1	Measurement completed	0
bit3		0/1	Low signal quality	0
bit2		0/1	Sampling timeout	0
bit1		0/1	UART receive overflow	0
bit0		0/1	Instruction check error	0

Instruction check error: Received an error command.

UART Overflow: The single frame data received by the UART exceeds the buffer size (80 bytes).

Sampling Timeout: The specified number of signal acquisitions were not completed and the sampling process exceeded the predetermined duration.

Low Signal Quality: The signal quality does not meet the expected requirements (register EXS_TH).

Measurement Completion: This bit is 1 to indicate that a measurement has been completed. The frequency result of this measurement can be obtained by reading the frequency register S_FRQ. When it is continuous measurement, it will be set every time the measurement is completed. When it is a single measurement, it will be set only after the specified number of measurements are completed, indicating that the single measurement is completed (see “3.9.2 single measurement” for details.).

Frequency Overflow: The measured sensor frequency exceeds 6553.5Hz, then the real frequency value = measured value +6553.6Hz, see "3.13.4 Real-time frequency value register S_FRQ in frequency calculation and quality assessment".

3.9 Measurement Mode

The module has two measurement modes: continuous measurement and single measurement. The working mode is switched by writing the corresponding value to the measurement mode register WKMOD.[0]. Writing 1 enters the continuous measurement mode, and writing 0 enter Single measurement mode. WKMOD.[15] is used to set whether to disable the digital interface when the module is "busy". When the digital interface is disabled, the module will not receive any data or instructions transmitted via the digital interface. When the digital interface is not disabled, the module maintenance sensor measures priority logic. The received command will respond after the module completes the current measurement. Therefore, in general, it is not necessary to set WKMOD.[15] to 1, and keep the default value(0).

Working mode register WKMOD (0x05)

Bit	symbol	value	description	Reset value
bit15		Digital interface work strategy		0
		0	Never close the digital interface	
		1	Turn off the digital interface when busy	
bit14①		0	Synchronous update to EEPROM when modifying parameters	0
		1	Do not update to EEPROM when modifying parameters	
bit13:4		0	Reserved	0
bit3:1		Frequency module register display content setting		0
		0	Display frequency modulus	
		1	Display high-resolution frequency value, resolution: 0.01Hz	
		2~7	Undefined function	
bit0		Working mode setting		1
		0	Switch module to single measurement mode	
		1	Switch module to continuous measurement mode	
Note 1: When it is necessary to modify the module parameters frequently, it is recommended to set this bit to 1 to reduce the impact of write operations on EEPROM lifetime.				

3.9.1 Continuous Measurement Mode

In continuous measurement mode, the module automatically performs sensor excitation and data read operations at regular intervals. The interval between two adjacent measurements is set by the register MM_INTE in milliseconds.

Continuous measurement interval register MM_INTE (0x06)

Bit	symbol	value	description	Reset value
bit15:0		5~65535	Continuous measurement interval,	500

			unit: mS	

In fact, whether it is continuous measurement mode or single measurement mode, it will wait for MM_INTE milliseconds before sending the excitation signal to the sensor every time. Therefore, the continuous measurement time interval is also called "waiting time before excitation".

3.9.2 Single Measurement Mode

A single measurement means that the module is always in the "idle" state, and when a single measurement command is received, the working mode of the single measurement process is started immediately (hence, the single measurement mode of operation can also be understood as "stop measurement"). In the single measurement mode, a short high pulse (10ms) is output on the RTS pin every 3 seconds to indicate that it is currently in the single mode. Three types of instructions can trigger a single measurement.

(1) Use a dedicated single measurement instruction

A specific single measurement instruction is sent to the module through the UART interface, and the frequency measurement result is actively uploaded after the module measurement is completed, and the instruction format is:

Instruction data frame structure

Command Head 0xAA 0xAA	address code	function code	Add check
2 bytes	1 byte	1 byte	1 byte

Return data frame structure

Command Head 0xAA 0xAA	address code	function code	Frequency value	Add check
2 bytes	1 byte	1 byte	2 bytes	1 byte

Instruction header: fixed to two bytes in hexadecimal AAAA.

Function code: 0x1x,0x3x or 0x7x, the end "x" means several excitation and reading operations. The 0x1x function code indicates that x measurement readings are taken directly, and 0x3x indicates that historical data is cleared before measurement (history data affects data filtering, see "3.14 Data Filtering" for details), 0x7x means to stop the measurement process immediately when the sensor signal quality is detected to be qualified or the specified number of measurements is reached.

When using the AAAA command for single frequency measurement, the frequency value will be actively output after the specified x measurements are completed (the above "return data frame"). If the filtering function is currently set, the output frequency value is the filter value of x times. If the filtering function is not set, the output frequency value is the real-time frequency value of the last measurement. The frequency value is represented by 2 bytes, and the high byte is first, and the unit is 0.1 Hz.

For a single measurement, the number of measurements should be as follows >= 3 times, and it is recommended to use the high-voltage excitation method and turn on the historical data filtering function. If the sweep method is used, the first measurement failure may occur, affecting the final result calculation (if historical data filtering is used).

Example: in single measurement mode

Send a single measurement command to the module: AA AA 01 13 68

The module starts to complete 3 measurements, the frequency measurement result data is output: AA AA 01 13 34 3A D6, the currently measured frequency value is (0x34*256+0x3A)/10=1337.0Hz.

(2) Write a single measurement instruction code to the system function register

Use the digital interface, any communication protocol, to write a 0x1x or 0x3x instruction code to the system function register SYS_FUN to trigger a single measurement. When using this method, the module follows the master-slave communication mechanism and does not actively upload data after the measurement is completed. It can be judged whether the current measurement has been completed by reading the system status register SYS_STA.[4]. When the completion of the detection, the frequency register S_FRQ is read to obtain the frequency result of this measurement.

Example: Using the MODBUS protocol in a single measurement mode

Send a single measurement command to the module: 01 06 00 03 00 13 38 07

The module returns the MODBUS response data: 01 06 00 03 00 13 38 07

The module starts to complete 3 measurements. When it does not output any information after completion, it needs to read SYS_STA.[4] to judge whether the measurement is completed.

(3) Read the frequency register S FRQ directly

In the single measurement mode, the frequency register S_FRQ is read using MODBUS or AABB communication protocol, and the VM module executes 0x73 instruction code and returns register values according to the communication protocol used.

Example of single measurement instruction (module address is 0x01)

Instruction	Response	NOTE
AA AA 01 13 68	AA AA 01 13 ***35 B3*** 50	Perform 3 measurements and return the frequency value
AA AA 01 33 88	AA AA 01 33 ***35 B4*** 71	After clearing the historical frequency value, make 3

		measurements and return the frequency value.
AA AA 01 73 C8	AA AA 01 73 ***35 B4*** B1	Stop measuring and return the frequency value immediately after the qualified return signal is detected (up to 3 measurements).
01 06 00 03 00 13 38 07	01 06 00 03 00 13 38 07①	Make 3 measurements and save the frequency in the frequency register.
01 06 00 03 00 33 39 DF	01 06 00 03 00 33 39 DF①	After clearing the history frequency, make 3 measurements and save it in the frequency register.
01 06 00 03 00 73 38 2F	01 06 00 03 00 73 38 2F①	Stop the measurement immediately after the qualified return signal is detected and save the frequency value in the frequency register (up to 3 measurements).
AA BB 01 83 00 13 FC	AA BB 01 03 00 13 7C①	
AA BB 01 83 00 33 1C	AA BB 01 03 00 33 9C①	
AA BB 01 83 00 73 5C	AA BB 01 03 00 73 DC①	
01 03 00 23 00 01 75 C0	01 03 02 ***35 B0*** AE A0	Stop measuring and return the frequency value immediately after the qualified return signal is detected (up to 3 measurements).
AA BB 01 23 89	AA BB 01 23 ***35 B0*** 6E	
Note 1: this data is the module's standard response to the instruction to modify the system function register (non-frequency value). Italic is the frequency value returned by the module.		

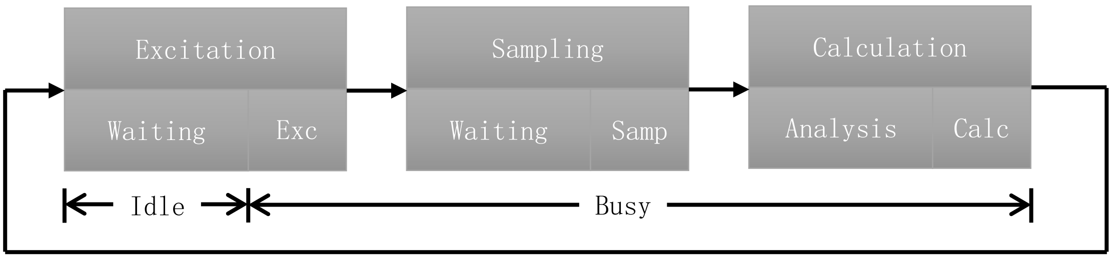
3.10 Vibrating Wire Sensor Measurement Process

As shown in the figure below, the measurement process of VM5XX is divided into three major steps: excitation, sampling, and calculation. Each large step can be split into several sub-processes. In the continuous measurement mode, a new measurement process is restarted immediately after the calculation is completed. In the single measurement mode, the specified number of measurement processes are triggered only after a single measurement command is received, and after the measurement is completed, wait for the next measurement instruction.

Excitation: The high-voltage pulse or low-voltage sweep method is used to send an excitation signal to the sensor to make the sensor steel wire self-oscillate. This module supports ten excitation methods.

Sampling: Collecting sinusoidal signals generated by self-oscillation of sensor steel wires.

Calculation: The collected sensor signals are subjected to quality evaluation and adjustment calculation, and the vibration frequency of the sensor steel wire is calculated.



Vibrating wire sensor measurement process

The general flow of VM5XX module measurement is:

(1) Check if the sensor is connected;

(2) Delay for a period of time

(3) Transmitting a specific excitation signal to the sensor coil to generate self-oscillation of the sensor steel wire;

(4) Delay for a period of time, waiting for the sensor return signal to be stable;

(5) Detecting the signal returned by the sensor coil, and performing quality assessment and result calculation when the signal meets the predetermined requirements;

(6) Reading the temperature sensor;

(7) Update the running status and calculation result to the corresponding register.

(8) If the automatic upload data is set, the specified data is sent actively.

3.11 Sensor Connection Detection

The module can detect the access status of the sensor coil in real time. As mentioned above, when no sensor is detected, the RTS pin outputs a periodic level signal of 10 Hz (high level 50 ms, low level 50 ms).

The criterion for determining whether the sensor is connected is the value of the sensor coil resistance, and the coil resistance value is stored in the register S_RES. When it is detected that the resistance between SEN+ and SEN- pins is between 50Ω~10kΩ, the sensor is considered to be connected; when the resistance value is less than 50Ω, it should be checked whether SEN+ and SEN- are short-circuited; when the resistance value is 10kΩ~30kΩ When checking whether the sensor is in good contact, when the resistance value is 30kΩ or more, it is basically judged that the sensor is not connected.

By default, the module sends an excitation signal to the sensor only after a valid sensor access is detected, and the vibrating wire sensor frequency reading is completed. The register EX_METH.[4] defines whether or not to ignore the sensor access detection and force the excitation signal function. See the EX_METH register description in the subsequent excitation method.

Coil resistance value register (0x27)

Bit	symbol	value	description	Reset value
bit15:0		0~65535	Sensor coil resistance, unit: ohm	0

3.12 Sensor Excitation Method

VM5XX supports three basic excitation models (methods) and three custom (combination) excitation methods based on basic methods. The selection of the excitation method is done by modifying the register EX_METH.[3:0]. EX_METH.[4] is used to set whether to ignore or not. The sensor's access detection forces the excitation signal to be sent.

The three basic excitation methods are: high voltage pulse excitation method, step low voltage sweep method, progressive low voltage sweep method, and three custom excitation methods are: frequency feedback fixed frequency sweep method (high voltage pulse excitation method + step low voltage) Sweeping method), frequency feedback interval frequency sweeping method (high voltage pulse excitation method + progressive low voltage sweep method), segmental progressive low voltage sweep method (4 frequency bands).

Excitation Method Register EX_METH (0x0A)

Bit	symbol	value	description	Reset value
bit15:7		0	Reserved, not write non-zero data	0
bit6:5		First excitation method definition (only valid when		0

		"feedback excitation method")		
		0	First excitation method: high voltage pulse	
		1	First excitation method: segment sweep - custom frequency band	
		2	First excitation method: full frequency sweep method	
bit4		Whether to force the excitation signal		0
		0	Send an excitation signal only after detecting the sensor access	
		1	Forced to send an excitation signal (ignoring whether the sensor is connected)	
bit3:0		Excitation method definition		1
		1	High voltage pulse excitation method (default)	
		2	Stepping low voltage sweep	
		3	Progressive low-voltage sweep	
		4	Frequency feedback fixed frequency sweep method (recommended)	
		5	Frequency feedback interval frequency sweeping	
		8	Segmented progressive sweep method - custom frequency band (FMIN~FMAX)	
		9	Segmented progressive sweeping method - the first frequency band (300~1500Hz)	
		10	Segmented progressive sweep method - the second frequency band (1500~2700Hz)	
		11	Segmented progressive sweep method - the third frequency band (2700 ~ 3900Hz)	
		12	Segmented progressive sweeping method - Band 4 (3900~5100Hz)	
		13	Full-band sweep method (FMIN~FMAX)	
① The default value is only for the sake of consistent with the previous firmware. In the current version recommended: First excitation method=2, Excitation method=4 or 5.				

3.12.1 High Voltage Pulse Excitation Method (HPM)

A method of transmitting a single instantaneous high-voltage pulse signal to a vibrating wire sensor to generate an autonomous vibration of the steel wire (HPM). In the high-voltage pulse excitation method, the process of raising the low voltage to a high voltage (generally between 100V and 200V) with VSEN as the voltage source is called “pump voltage”, The value of high voltage produced by pump voltage and the amount of electricity released to sensor are related to parameters such as pump voltage duration and pump voltage source voltage.

VM5XX can generate high-voltage pulse excitation signal of 30~220V, and externally connect VSEN with independent pins to achieve a certain degree of high-voltage custom function. Under other conditions, higher VSEN voltage can obtain higher voltage excitation signal. Excessive voltage excitation signals may affect the life of the vibrating wire sensor and also cause the “forced vibration” time to become longer after the sensor is self-oscillating. See “Explanation of forced vibration in 3.13”.

Pumping duration register HP_DUR (0x0D)

Bit	symbol	value	description	Reset value
bit15		Whether to intelligently terminate the pumping process early		0
		1	End the pumping process immediately after the expected voltage is reached	
		0	Complete the pumping vol with a fixed length of time	
bit14:12		0	Reserved, undefined function	0
bit11:0		0~4095	Pump vol process duration. Unit: ms	1000

Expected voltage register HP_EXP (0x0E)

Bit	symbol	value	description	Reset value
bit15		Whether to enable the expected voltage limit function		1
		0	Do not enable the expected voltage regulation function	
		1	Enable expected voltage regulation	
bit14:8		0	Reserved, undefined function	0
bit7:0		0~240	The voltage value that the module strives to maintain during pumping. Unit: V	130
Example: If you want to maintain 100V, you should set this parameter to 0x8000+0x0064=0x8064(32868)				

According to the test experience, the high-voltage pulse excitation signal of 80V~200V can make the vibrating wire vibrate well, so as not to affect the life of the sensor. Under the premise of meeting the measurement requirements, the HP_EXP register should be used to keep the high-voltage excitation signal as low as possible. High voltage may cause the sensor coil to burn down.

Whether it is a high voltage pulse excitation or a low voltage sweep excitation, the actual voltage value applied to the sensor during the last sensor excitation can be obtained by reading the register VSEN_RT in units of 0.01V.

Excitation voltage value register VSEN_RT (0x28)

Bit	symbol	value	description	Reset value
bit15		0	Reserved, undefined function	0
bit14:0		0~24000	The actual voltage value that was loaded onto the sensor when the sensor was last excited. Unit: 0.01V (or 10mV)	0

3.12.2 Low Voltage Step Frequency Sweeping Method (LSF)

Low-voltage sweeping means that a periodic pulse excitation signal is sent to the sensor coil using a low voltage. When the excitation signal frequency is close to the self-vibration frequency of the sensor steel wire, the steel wire generates self-vibration. The VSEN voltage is the sweep voltage during low voltage sweep.

The low-voltage step sweep method is to output the low-voltage sweep excitation signal of the specified period with a fixed frequency interval in a specified frequency interval (specify the starting and ending frequencies). The registers related to the low-voltage step sweep method are: start frequency register (FS_FMIN), stop frequency register (FS_FMAX), frequency step register (FS_STEP), and a single-step sweep signal period number register (FS_SCNT). For the measurement process, see "3.15 Measurement Duration and Optimization".

Start frequency register FS_FMIN (0x0F)

Bit	symbol	value	description	Reset value
bit15:13		0	Reserved, undefined function	0
bit12:0		300~8000	The starting frequency value of the sweep excitation. Unit: Hz	1500

Termination frequency register FS_FMAX (0x10)

Bit	symbol	value	description	Reset value
bit15:13		0	Reserved, undefined function	0
bit12:0		300~8000	The frequency of the termination frequency of the sweep excitation. Unit: Hz	1600

Frequency step register FS_STEP (0x11)

Bit	symbol	value	description	Reset value

bit15:8		0	Reserved, undefined function	0
bit7:0		0~255	The frequency difference between adjacent sweep excitation signals. Unit: Hz	5
The recommended value is 5~20, and the priority is 5 or 10Hz.				

Single-step sweep signal cycle number register FS_SCNT (0x12)

Bit	symbol	value	description	Reset value
bit15:8				
bit7:0				
Please set it to 0xC814				

Current sweep frequency register SFV_RT (0x21)

Bit	symbol	value	description	Reset value
bit15:13		0	Reserved, undefined function	0
bit12:0		0~8000	The frequency value currently being used by the sweep signal output process. Unit: Hz	0
This register is read-only and the content is refreshed by the module.				

Note: When low-voltage step sweep method, each step excitation signal output is considered to be a complete excitation process, that is, the sensor return signal detection, sampling, and calculation are performed after each step excitation. Several other excitation methods are performed after the entire excitation process is completed.

3.12.3 Low Voltage Gradual Frequency Sweeping Method (LGF)

The low-voltage gradual frequency excitation signal from low to high is output to the sensor in a short period of time(LGF) (generally not exceeding 1000 mS). The registers related to the low-voltage gradual frequency excitation method are: start frequency register (FS_FMIN), stop frequency register (FS_FMAX), frequency step register (FS_STEP), and a single-step sweep signal period number register (FS_SCNT).

The above register interpretation and value implies the same as the low-voltage step frequency excitation method. It should be noted that in the step sweep, the sensor return signal detection is performed after each step excitation is completed, and the gradual sweep method is performed, the sensor return signal detection is performed after the completion of the entire excitation process. Therefore, special attention should be paid to the time consuming of the entire process (within 1000 ms). The excessively long duration may cause the sensor to self-vibrate and the module cannot correctly acquire the sensor return signal. See “3.15 Measurement Duration and Optimization” for details.

3.12.4 Frequency Feedback Fixed Frequency Sweeping Method (FFF)

When the first excitation is performed, the pre-specified "first excitation method" is used to sample, evaluate, calculate and calculate the sensor return signal. If the signal quality reaches a predetermined value (register EXS_TH.[7:0]), the subsequent excitation is automatically changed For a fixed-frequency low-voltage sweep method(LFF), the frequency of the excitation signal is the most recently calculated sensor frequency value. In the low-voltage sweeping process, when the detected signal quality is lower than the predetermined target, automatically switched to the pre-specified "first excitation method" to excite the sensor. The above steps are repeated.

Scheduled signal quality register EXS_TH (0x1D)

Bit	symbol	value	description	Reset value
bit15:12		0	Reserved, undefined function	0
bit11:8		Which data is used as the criterion for evaluating signal quality		0
		0	Sampling evaluation quality value	
		1	Signal amplitude average (recommended)	
		2	Percentage of quality samples	
		3	Sample standard deviation - all	
		4	Sample standard deviation - quality	
bit7:0		0~100	The signal quality threshold, which is equal to or higher than this value, indicates that the signal quality meets the requirements.	80

Note: This parameter can be combined with other related parameters to realize multi-condition judgment rules. For example, set the remaining sample limit register CAL_PAR2 to 2, and set this register to EXS_TH=0x0050(That is: decimal 80) to achieve a quality sample that must be larger than the expected number of samples 50%, and the sampling quality must be greater than 80%.

3.12.5 Frequency Feedback Gradual Frequency Sweeping Method (FFG)

When the first excitation is performed, the pre-specified "first excitation method" is used to sample, evaluate, calculate and calculate the sensor return signal. If the signal quality reaches a predetermined value (register EXS_TH.[7:0]), the subsequent excitation is automatically changed For the progressive low-voltage sweep method(LGF), in the aggressive low-voltage sweep method, the start frequency and the end frequency are automatically set to the last calculated sensor frequency value (center frequency value) 20Hz above and below (default value, can be modified by modifying the register FSG_TH) Upper and lower limits of the frequency range). In the low-voltage sweeping process, when the detected signal quality is lower than the predetermined target, the sensor is automatically switched to the pre-specified "first excitation method" to excite the sensor. The above steps are repeated.

Feedback interval frequency sweep upper and lower limits FSG_TH (0x18)

Bit	symbol	value	description	Reset value
bit15:8		0~255	Sweep frequency lower limit distance from center frequency difference, in Hz	20
bit7:0		0~255	Sweep frequency upper limit distance from center frequency difference, in Hz	20

3.12.6 Segmental Gradual Frequency Sweep Method (SGF)

The module automatically specifies the frequency range to be divided into four small frequency bands for frequency sweep and sensor signal detection. When the frequency range is specified by the user, the register FS_FMIN and the register FS_FMAX should be modified. When the frequency range is determined by the module, the module divides the possible frequency of the vibrating wire sensor (300Hz~5000Hz) into four large frequency bands, respectively 300. ~1500Hz, 1500~2700Hz, 2700~3900Hz, 3900~5100Hz, determine which frequency band to use by the value of the excitation method register EX_METH.[3:0], and the module USES the low-voltage sweep method to send the excitation signal to the sensor in the specified frequency range.

For a selected predetermined frequency band, the module continues to divide each frequency band into 4 small segments with an interval of 300 Hz during the process of transmitting the excitation signal, respectively transmitting the excitation signal by using the Low-voltage gradual frequency sweeping method(LGF) and reading the sensor return signal, recording each The return signal quality and frequency calculation results are used once, and the best quality frequency value among the four sets of data is taken as the current measurement result. The Low-voltage gradual frequency sweeping method (LGF) is time-consuming, measuring 3 to 5 seconds each time. During this process, the module is always in the “busy” state and will not respond to digital interface commands (still receiving instructions). Before using this function, the delay read register needs to be set to 0mS, and the frequency step FS_STEP and the output period register FS_SCNT are set. The recommended values are RD_INTE.[11:0]=0, FS_STEP=5Hz, FS_SCNT=10, when the sensor frequency is lower than At 500 Hz, to satisfy the principle that the duration of a single frequency gradual excitation signal does not exceed 1000 ms, FS_STEP should be increased or FS_SCNT should be reduced.

3.12.7 Full Frequency Sweeping Method (FFS)

The user specifies the start frequency value of the sweep FS_FMIN and the end frequency value FS_FMAX (Usually for 300 ~ 5000 Hz), and the module automatically scans from the low frequency to the high frequency. When it is detected that the signal quality returned by the sensor reaches the predetermined standard (see “EXS_TH register” for details) immediately exit and read the signal returned by the sensor and calculate the signal as a frequency value updated to the real-time frequency register.

Assuming that the specified sweep range is full frequency band (300Hz~5000Hz), step FS_STEP=5Hz, FS_SCNT=10, it takes about 12 seconds to complete the full-frequency scan (the longest time, automatically abort when the sensor signal is scanned).

Recommended register value

register	symbol	Recommended value	description
FS_FMIN		300	The sweep frequency starts at 300Hz
FS_FMAX		5000	Sweep stop frequency is 5000Hz
FS_STEP		5	Sweep step is 5Hz
FS_SCNT		10	Output 10 signals per step
EXS_TH.[11:8]		1	Taking the signal return signal amplitude as the signal quality criterion
EXS_TH.[7:0]		60	Signal quality threshold is 60%

The latter of the excitation methods (FFF, FFG and SGF, FFS) are only based on the limited combination of the first three basic excitation methods. Application, according to the first three basic methods, there are also many other different combinations of excitation methods, through the relevant parameter control, measurement results feedback free customization, to adapt to different practical needs.

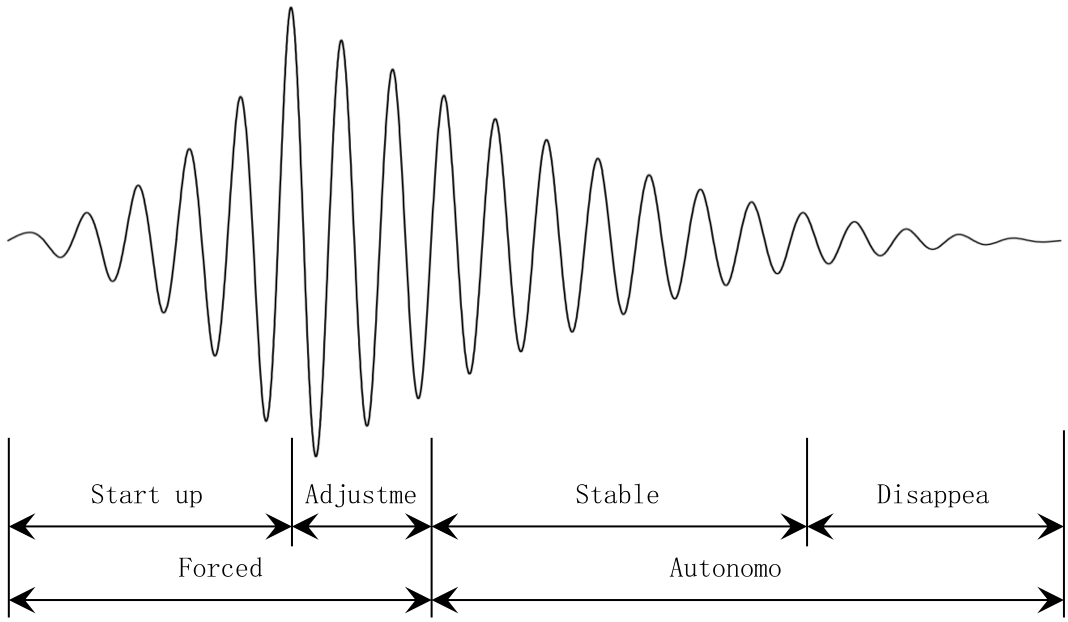
3.13 Signal Detection and Analysis Calculation

3.13.1 Delayed Sampling

As shown in the figure below, after the string of the vibrating wire sensor starts to oscillate, the signal intensity quickly reaches its maximum in a short time, and then gradually returns to rest under the action of the steel wire tension and air resistance. We can divide the whole vibration process into vibration starting, adjustment, stability and disappearance stages. In the above several stages, the vibration in the starting and adjusting phases is also called forced vibration, and the stable and disappearing phases are called independent vibration.

Forced vibration: It means that the output waveform of the sensor is affected by the excitation signal, and the output vibration signal is not very stable and cannot fully represent the vibration of its own natural frequency.

Autonomous vibration: Regular vibration (resonance) is carried out at the vibration frequency of the sensor steel wire.



In order to obtain the true frequency value of the sensor, it is necessary to sample and calculate the frequency of the periodic signal during the autonomous vibration. Therefore, when the excitation of the sensor is completed, it takes a delay to start sampling the sensor return signal. This delay length is controlled by the value of the read delay register RD_INTE.

The read delay register RD_INTE specifies the delay time after excitation and whether the delay length is adjusted by the module based on the signal quality calculated from the previous measurement. The value in RD_INTE.[11:0] determines how long the delay after the excitation signal is sent and then starts the sampling of the sensor return signal. The unit of duration is defined by RD_INTE.[14], which can be “millisecond value” or “signal num value”. RD_INTE.[15] is used to set whether to use the intelligent delay duration adjustment function. When RD_INTE.[15] is 1, the module automatically adjusts the delay of this delay register if the last measured signal quality does not meet the expected requirements. The duration value is 1/2 of the set value.

In comparison, the forced vibration time of the sensor after excitation by the high-voltage excitation method is longer than that of the low-voltage sweep method. Therefore, it is recommended to set the sampling delay register to 0 when using the low-voltage sweep excitation method. That is: if the high-vol pulse method is used to excite the sensor, it takes a delay to start sampling after the excitation is completed. When the low-voltage sweep method excites the sensor, the sampling can be started immediately after the excitation is completed, without waiting.

Read delay register RD_INTE (0x10)

Bit	symbol	value	description	Reset value
bit15		Read delay method setting		0
		0	Always use a fixed delay time	
		1	Dynamically adjust the delay time based on the quality of the last measurement return signal	
bit14		Delay unit		0
		0	The delay unit is: mS	
		1	The delay unit is: return signal period	
bit13:12		0	Reserved, undefined function	0
bit11:0		0~4095	Read wait delay value	200

3.13.2 Signal Amplitude Detection

The signal amplitude refers to the amplitude of the signal after the original signal output by the sensor is self-oscillated, and is expressed by a percentage, that is, the representation method of the strength of the return signal of the sensor is expressed by a percentage. The signal amplitude is defined as 0%~100% in percentage form. 90%~100% means the signal is too strong, 60%~90% is excellent, 40%~60% can get the higher precision sampling value, close to or less than 30% is poor or no signal. The amplitude of the signal is measured at different stages of the measurement process. The amplitude of the first return signal is stored in SIG_VALH.[15:8], and the amplitude of the signal at the start of sampling is stored in SIG_VALH.[7:0]. At the end of sampling, the signal amplitude is stored in SIG_VALL.[15:8], and the average of the three signal amplitudes is stored in SIG_VALL.[7:0].

If the average signal amplitude is greater than 90%, there may be a return signal overshoot.

The magnitude of signal directly affects the reliability of sensor frequency. The signal amplitude is greatly affected by the excitation signal. If the signal amplitude is not satisfactory, the excitation method of the sensor should be adjusted and the excitation voltage should be adjusted to improve the signal amplitude.

Signal amplitude real-time value register SIG_VALH (0x2C)

Bit	symbol	value	description	Reset value
bit15:8		0~100	First signal amplitude after excitation, unit: %	0
bit7:0		0~100	First signal amplitude at the time of sampling, unit: %	0

Signal amplitude real-time value register SIG_VALL (0x2D)

Bit	symbol	value	description	Reset value
bit15:8		0~100	Signal amplitude at the end of sampling, in %	0
bit7:0		0~100	Average signal amplitude in %	0

Note: The signal amplitude function can be used only for hardware HW1.20 or later. This value is invalid for older versions.

3.13.3 Signal Detection and Sampling

The VM5XX internally has a signal detection and validity detection mechanism based on the characteristics of the vibrating wire sensor. When the signal amplitude is within a preset reasonable interval, data sampling is performed, and the signal quality analysis is performed immediately after a sufficient number of sample samples are completed. The obtained frequency, Frequency module value and multiple signal quality representation values are updated in the corresponding read-only registers, and the current measurement result data and signal quality can be obtained by reading these register values.

There are two events that can cause the module to terminate (or complete) data sampling, one is to collect the specified number of samples （RD_COUNT.[8:0]）, the other is that the sampling amount is not reached but the sampling time is over(RD_COUNT.[15:9] default is 1000mS).

Signal amplitude limit register SIG_TH (0x1E)

Bit	symbol	value	description	Reset value
bit15:8		0~100	Signal amplitude upper limit, unit: %	100
bit7:0		0~100	Signal amplitude lower limit, unit: %	0

During the signal sampling process, the amplitude of the current signal is detected when each signal occurs. When the amplitude of the signal is between the upper and lower limits specified by SIG_TH, it is actually sampled as a valid sample data. The default value of SIG_TH is 0x6400, that is, the upper limit is 100% and the lower limit is 0%. All data is considered to be "valid".

Under the premise that the sensor return signal is good, in order to achieve the purpose of collecting as many moderate amplitude signals as possible, this register should be set. The general upper limit is 90 and the lower limit is 50 (0x5A32). This module has another sample error culling rule in the sampling calculation process. Even if the signal amplitude limit is not entered, the abnormal data will be eliminated during the culling calculation. Therefore, the default value will not affect the measurement result. See “3.13.4 Frequency Calculation and Quality Assessment” for details.

3.13.4 Frequency Calculation and Quality Assessment

Using the collected signal sample data, first estimate a frequency value, called "pseudo frequency value"; then In a module to eliminate abnormal data algorithm model, the register CAL_PAR1 value as the main decision parameters, and Each sampling value is calculated with the pseudo-frequency value, and the abnormal data that does not meet the requirements is removed, The remaining data was identified as a "good" sample; the original sample standard deviation and the high quality sample standard deviation are stored in the registers SIG_STD.[15:8] and SIG_STD. In 7:0], the quality sample quantity is updated to the register HQ_COUNT, the good sample quality rating value is stored in the register SMP_QUA, and the final sensor frequency value and the Frequency module value are updated to the register S_FRQ and the register F_REQM, respectively. When the remaining "quality" sample quantity is lower than the CAL_PAR2 limit or the standard deviation is too large, the measurement sample quality evaluation result is forced to 0%.

Expected Sample Count Register RD_COUNT (0x09)

Bit	symbol	value	description	Reset value
bit15:9		1~127	Sampling timeout period, unit 100ms	0/10
bit8:0		0~300	Expected number of samples	200

RD_COUNT is a combined parameter register containing two parameters: expected number of samples and sample timeout. RD_COUNT.[8:0] specifies the expected number of samples. RD_COUNT.[15:9] defines the sampling timeout period. If the sampling process exceeds at this time, the expected number of samples is still not completed, forcing the sampling process to end. The sampling timeout unit is 100ms, and the parameter range is 1~127 (that is, the maximum timeout length can be 127*100ms=12.7 seconds). When the timeout length is set to 0, the default timeout period is 1000ms. The theoretical sampling duration is related to the sensor frequency and the expected number of samples. The lower the sensor frequency and the more samples are expected, the longer the theoretical sampling time should be. The correct timeout should be set to about 1.5 * theoretical sampling duration.

$$\text{Theoretical sampling duration} = \frac{\text{Expected number of samples}}{\text{Sensor frequency value}} \times 1000\text{ms}$$

When the sensor frequency is unknown, a lower frequency (such as 500Hz) should be estimated. If the expected number of samples is 200, the theoretical sampling duration is $\frac{200}{500Hz} \times 1000 = 400ms$, the timeout period = 400 * 1.5 = 600ms.

Error limit register CAL_PAR1 (0x15)

Bit	symbol	value	description	Reset value
bit15:12			A method for removing large error data	0
		0	Median ratio	
		1	Standard deviation (Pauta criterion)	
bit11:0		0~100	data rejection factor When the data rejection method is median ratio will $(\text{Single sample value} - \text{pseudo frequency value}) > \left(\frac{\text{Pseudo frequency value}}{CAL\_PAR1}\right)$ Sampling data rejection When the data rejection method is the Pauta criterion, the sampling data larger than the standard deviation of CAL_PAR1* is eliminated (normally 3 standard deviation)	20

Remaining sample limit register CAL_PAR2 (0x16)

Bit	symbol	value	description	Reset value
bit15:0		0~100	Abandon frequency and quality assessment calculations (mandatory 0%) when the remaining "high quality" samples are too low. $\text{Number of thresholds} = \frac{\text{Expected number of samples}}{CAL\_PAR2}$	4

Signal Synthesis Quality Register SMP_QUA (0x22)

Bit	symbol	value	description	Reset value
bit15:8		0	Reserved, undefined function	0

bit7:0		0~100	The sensor returns the combined quality of the signal. unit:%	0

Signal quality: Also known as “sample data quality assessment” or “sample data quality assessment”, this data is the quality of the “quality” sample, because the final frequency result is calculated from the “quality” sample, so SMP_QUA The value can reflect the reliability and credibility of the frequency results. The sample quality is expressed as a percentage. In general, the frequency value when the sample mass is 50% or more can represent the true frequency of the sensor, while under 50%, the reliability of the frequency value is considered to be poor or untrustworthy, during the use of the module. Try to use the frequency value of the sample quality above 85% as the final result.

A good quality sample is a relative concept. The number of high quality samples is directly affected by the expected error register CAL_PAR1. A looser expected error increases the number of good samples, but it also introduces a sample with a larger error into the frequency calculation process, and vice versa. The quality sample quantity, sampling quality evaluation result, signal amplitude and standard deviation are all indicative quantities of the accuracy of the measurement results, which need comprehensive consideration.

Sensor frequency value register S_FRQ (0x23)

Bit	symbol	value	description	Reset value
bit15:0		1000~65535	The frequency value of the sensor read from the last measurement Unit: 0.1Hz (100.0~6553.5Hz)	0
Register overflow occurs when reading frequencies above 6553.5 Hz, then the actual frequency = measured value + 6553.5 Hz **When the frequency overflow occurs, the bit5 of register SYS_STA is automatically set to 1.**				

Sensor frequency modulus register F_REQM (0x24-0x25)

Bit	symbol	value	description	condition	Reset value
bit31:16	F_REQM_H	0~2^{32}	High 16 bits of Frequency module value[①]	WKMOD.[3:1]=0	0
bit15:0	F_REQM_L		Low 16 bits of Frequency module value[①]		
bit31:16	FRQ_H	0~655350	High 16 bits of Frequency value[②]	WKMOD.[3:1]=1	0
bit15:0	FRQ_L		Low 16 bits of Frequency		

			value②		

Note 1: Frequency module value=frequency value(Hz)*frequency value(Hz)/100
Frequency module value = High 16 bits of Frequency module value*65536 + Low 16 bits of Frequency module value. The frequency modulus value unit is: hundred Hz ^{2}.
Note 2: Frequency value = (FRQ_H*65536 + FRQ_L) / 100Hz
Note 2: Due to the difference in the accuracy of the calculation process, the display result when displayed as a high-precision frequency value will be slightly different from the result displayed in the S_FRQ register.

When reading the frequency value of the real-time measurement result, if the measured value is too different from the nominal frequency of the sensor, you should read the operation status register SYS_STA.[5] to determine whether the frequency register overflow has occurred, if SYS_STA.[5] is 1, the frequency value should be equal to (S_FRQ/10 + 6553.5) Hz. The value of the Frequency module is not affected by the frequency overflow, and the direct reading is the true value.

Good Sample Count Register HQ_COUNT (0x2B)

Bit	symbol	value	description	Reset value
bit15:9		0	Reserved, undefined function	0
bit8:0		0~300	Good Sample Count.	0

Sample standard deviation register SIG_STD (0x2C)

Bit	symbol	value	description	Reset value
bit15:8		0~255	Original sample standard deviation, unit: Hz	0
bit7:0		0~255	Standard deviation of good samples, unit: Hz	0

The numerical reliability of the current frequency can be determined by the following methods:

(1) The average signal amplitude is greater than 60%, the quality sample size is greater than 50% of the expected sample quantity and not less than 50, and the quality sample evaluation value is greater than 80%.

(2) The number of high-quality samples is greater than 50% of the expected number of samples and not less than 50, and the evaluation value of the high-quality samples is greater than 80%.

(3) The quality sample evaluation value is greater than 80%.

3.14 Data Filtering

The filtering method described below only works for the frequency value register S_FRQ.

Data filtering refers to a data processing method (recursive filtering) that performs smooth filtering on multiple adjacent measurement results. The filtering method is specified by setting the filter method register FIT_TYPE.[3:0], and the filtered sample number register FIT_COUNT.[7:0] is used to specify the number of historical data participating in the calculation. Only when the sampling data quality evaluation result is greater than zero value, will bring new value into the filter sample, i.e., if the new measurement sampling data quality evaluation result is 0, otherwise the filtering results will continue to follow the previous value (the value of the register S_FRQ no change), When no filtering method is used, register S_FRQ is the real-time measurement result of each time, independent of the sampling data quality assessment value.

VM5XX supports four kinds of historical data filtering methods, namely: median filtering method, arithmetic average filtering method, median value average filtering method, and weighted average filtering method. The historical data is based on the recursive storage of each measurement result, and internally maintains a predetermined number of historical data sequences, the real-time frequency of the frequency after each measurement is stored in the sequence, and the oldest data is discarded (FIFO first in first out) The historical data is used for filtering calculation, and the calculation result is updated to the register S_FRQ as the final frequency value. The historical data sequence is cleared each time the module is started. When the module is in the single measurement mode, the single measurement instruction code 0x3x will also clear the historical data sequence.

(1) Median filtering method: Sort the specified number of historical data, and take the value in the middle position as the final value.

(2) Arithmetic average filtering method: The average value of the specified number of historical data is used as the final value. When the probability of random errors in data reading is relatively large, it is recommended not to use this filtering method. Randomly occurring erroneous data will participate in the filtering calculation for a period of time, affecting the filtering result during this period.

(3) Median average filtering method: Sort the specified number of historical data, remove the maximum value and the minimum value, and calculate the average value of the remaining data as the final value. It can effectively eliminate occasional erroneous data.

(4) Weighted average filtering method: Backtracking a specified number of historical

data, the closer the time point is to the current time, the larger the data weight (the current value weight is the largest), and the average value is calculated according to different weights as the final value.

Filter method register FIT_TYPE (0x13)

Bit	symbol	value	description	Reset value
bit15:4		0	Reserved, undefined function	0
bit3:0		Historical data filtering method		0
		0	NONE (default)	
		1	Median filtering	
		2	Arithmetic average filtering	
		3	Median average filtering method (recommended)	
		4	Weighted average filtering	

Filter sample number register FIT_COUNT (0x14)

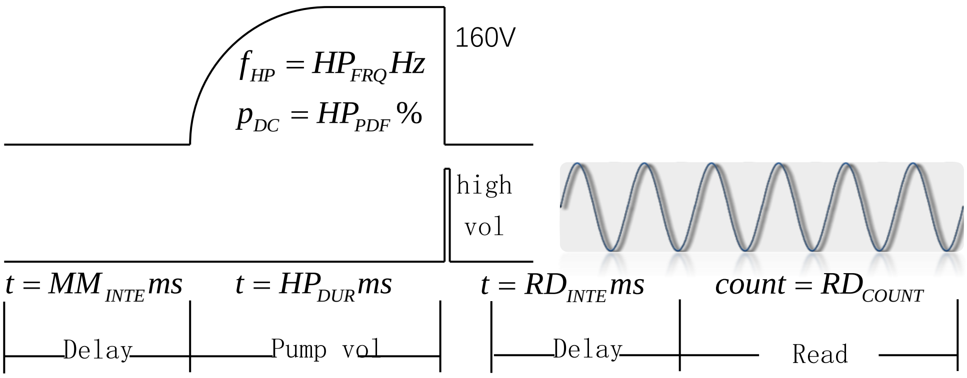
Bit	symbol	value	description	Reset value
bit15:8		0	Reserved, undefined function	0
bit7:0		3~30	How many historical data are used for filtering operations	10

The historical data filtering function is suitable for application scenarios where a fixed sensor frequency is measured for a long time. There must be enough historical data (measured enough times) to gradually show the filtering effect. When the measured sensor is not unique or needs to get the measurement result quickly, the historical data filtering function should be turned off or the module measurement rate should be increased by adjusting the parameters (for example, 5 measurements per second, see "3.16 Fast Measurement" for details). So that multiple measurements can be made in a short time to complete the filtering.

3.15 Measurement Duration and Optimization

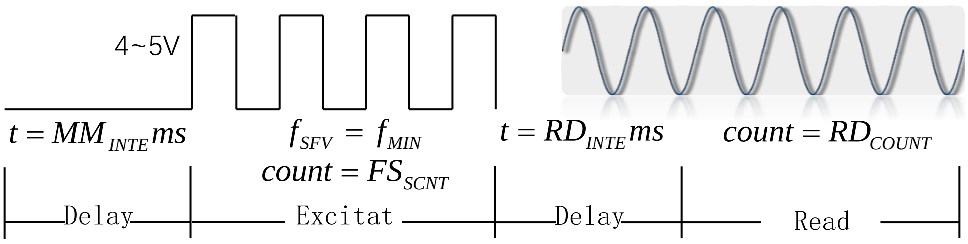
Different excitation methods and delay parameter value setting will lead to different measurement time of sensor. The following three basic excitation methods are used for length analysis. The other combined excitation methods can be derived by these three basic excitation methods.

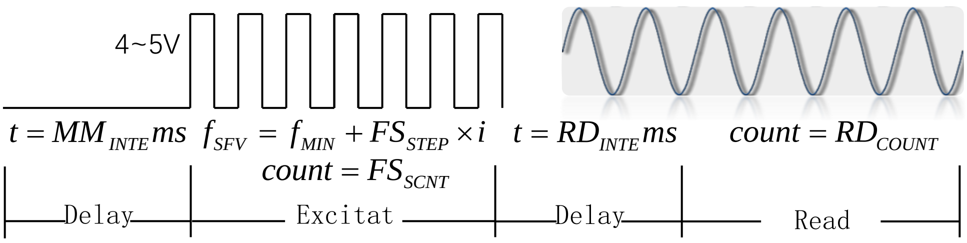
(1) Single high voltage pulse method



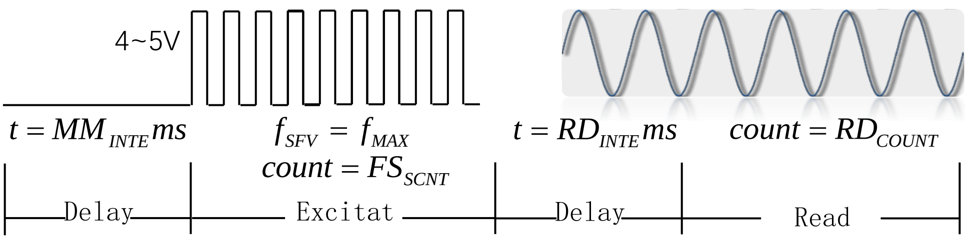
$$T_{SGL} = \left(MM_{INTE} + HP_{DUR} + \frac{1}{f_{sen}} \times RD_{COUNT} \times 1000 \right) ms$$

(2) Stepping low voltage sweep method





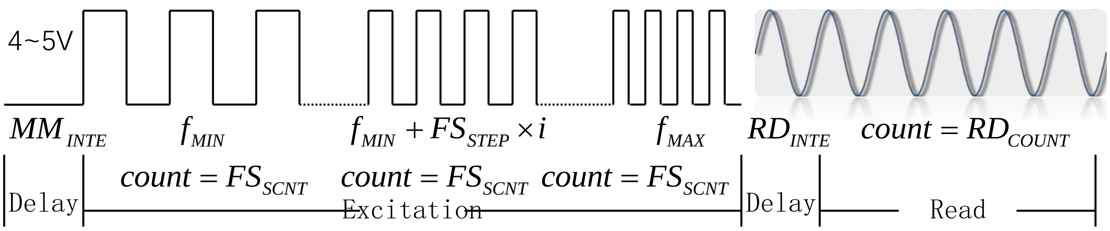
… …



$$T_{SGL} = \left(MM_{INTE} + \frac{1000}{f_{SFV}} \times FS_{SCNT} + \frac{1000}{f_{sen}} \times RD_{COUNT} \right) ms$$

$$T_{TOTAL} = \left(\left(MM_{INTE} + \frac{1000}{f_{sen}} \times RD_{COUNT} + \frac{2000 \times FS_{SCNT}}{f_{MIN} + f_{MAX}} \right) \times \left(\frac{f_{MAX} - f_{MIN}}{FS_{STEP}} + 1 \right) \right) ms$$

(3) Progressive low-voltage sweeping method



$$\begin{cases} T_{FOR} = \left(\sum_{i=1}^{n} \left(\frac{1000}{f_i} \times FS_{SCNT} \right) \right) ms \\ n = \frac{(f_{MAX} - f_{MIN})}{FS_{STEP}} + 1 \\ f_i = f_{MIN} + FS_{STEP} \end{cases} \Longrightarrow T_{FOR} = \left(\frac{2000 \times FS_{SCNT}}{f_{MIN} + f_{MAX}} \times \left(\frac{f_{MAX} - f_{MIN}}{FS_{STEP}} + 1 \right) \right) ms$$

$$T_{SGL} = \left(MM_{INTE} + \frac{1000}{f_{sen}} \times RD_{COUNT} + \frac{2000 \times FS_{SCNT}}{f_{MIN} + f_{MAX}} \times \left(\frac{f_{MAX} - f_{MIN}}{FS_{STEP}} + 1 \right) \right) ms$$

Notes:

After the sensor starts to oscillate, the output signal will continue for a period of time. In order to avoid the next excitation before the sensor is not restored, there will be a forced delay time before each excitation. The delay time can be set by the register MM_INTE.

In the progressive low-voltage sweep method, the total excitation timeT_{FOR}Must not be greater than 1000ms.

After the sensor starts to oscillate, delaying the reading of the signal frequency for a period of time is beneficial to improve the accuracy. For the measurement of the same sensor at different times, the delay time should be the same, and the sensor frequency value obtained by different delay times will vary slightly.

When reading a signal, the number of samples is directly related to the final accuracy, and the number of samples is proportional to the reading time.

Excessive high-voltage excitation has a certain impact on sensor life. It is recommended to use a low-voltage sweep method to excite the sensor when low-voltage sweep is available.

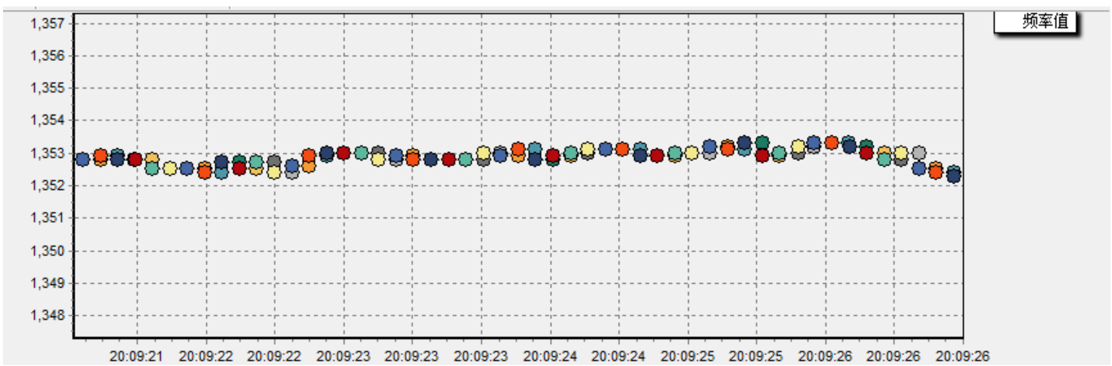
3.16 Fast Measurement (10Hz)

Fast measurement is a specific application of the previous section "Measurement Duration and Optimization". With reasonable time parameters, fast frequency excitation and reading can be achieved up to 10 times per second or higher.

The specific registers are set as follows:

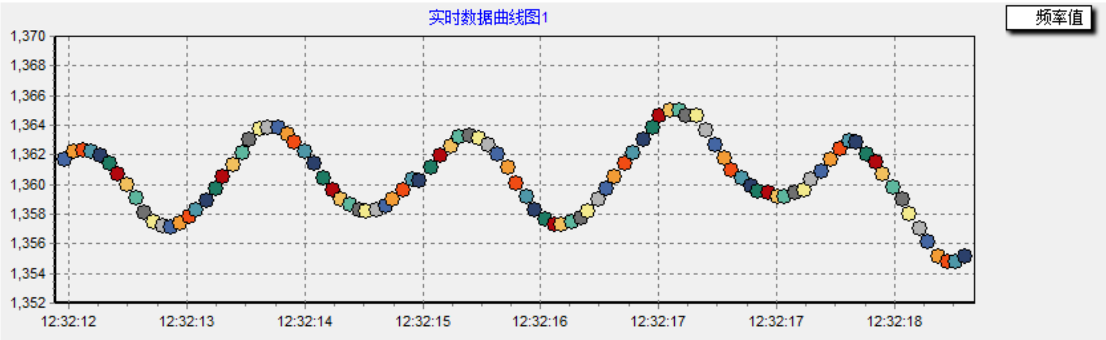
Register name	symbol	value	instruction manual
Excitation method	EX_METH	4	Frequency feedback fixed frequency sweeping
Expected number of samples	RD_COUNT	20	Sampling 20 sets of sensor signals per measurement
Continuous measurement interval	MM_INTE	0	Adjacent measurements do not wait
Sampling delay	RD_INTE	0	Sampling immediately after the excitation, do not wait
Single frequency output period	FS_SCNT	30	Output 30 cycles of low voltage excitation signal
Quality threshold	EXS_TH	70	Use high voltage excitation once when the sampling quality is lower than 70%
Automatic output	ATSD_SEL	0x3000	Non-required, the result is automatically output after the measurement is completed.
Other unlisted registers use factory defaults			

The figure below shows the actual measurement “time-frequency” graph. About 50 times of vibrating wire sensor frequency measurement is completed in about 5 seconds.



Time-frequency curve (10Hz)

Fast measurement is to use a minimum waiting time, as fast as possible excitation signal, as little as possible to reduce the single measurement time, at the cost of loss measurement accuracy. Under the above parameters, the vibrating wire reading accuracy is about 2Hz. (Sensor frequency 1355Hz, static measurement).



Dynamic frequency measurement of the sensor under cyclic external force (15Hz)

When you want to perform sensor excitation and data reading at the fastest rate, you should use the automatic upload function of the measurement results. If you still use the master-slave instruction to send the read command to the module in real time to obtain the data, it will seriously affect the measurement work efficiency of the module, on the other hand, because the module cannot be replied to when it is "busy", in the case of high-frequency measurement, there is an instruction synchronization problem that cannot return every measurement result in real time, and only a small amount of real-time data can be read. The automatic upload function will be reset every time the power is turned on, and you need to send the command to reset it. For details, see "3.4.3 Actively Uploading Measurement Data".

Under certain extreme parameters (such as high-speed measurements in this section), the module may not respond to each received instruction because it is in the "busy" state most of the time. In this case, the method of sending instructions multiple times should be tried and combined with the instruction reply to determine whether the module has answered the instruction.

3.17 Low-voltage Sweeping Frequency Self-calibration

When the sensor is excited by the low-voltage sweep method, the closer the output sweep signal period is to the sensor's natural vibration period, the better the vibration-starting effect of the sensor. When the module outputs the sweep excitation signal, there is inevitably a certain difference between the output signal period and the preset frequency expectation, resulting in inaccurate sweep frequency output affecting the excitation effect on the sensor.

The VM3/4/5XX module has been calibrated for the sweep output frequency when it is shipped from the factory. No correction is required when there is no abnormality. When modifications are needed, the parameters can be emptied, self-corrected, and the frequency error can be viewed by writing a specific instruction code to the system function register.

The self-calibration steps are as follows:

1. Remove the sensor (the sensor coil is separated from the module);
2. Set the forced excitation (EX_METH.[4]=1);
3. Send 0xC0 instruction code to the module to clear the existing calibration parameters;
4. Send the 0xC1 instruction code to the module to complete the parameter correction (this process takes about 3~5 seconds).

Instruction code (hexadecimal)	Functional description
0xC0	Clear the sweep frequency correction parameter.
0xC1	Self-testing and automatically calculating the corrected parameters and storing them①
0xC2	Self-detecting the sweep frequency and outputting the test result
Note 1: The operation of restoring factory parameters does not affect the sweep frequency correction parameters;	

For 0xC2 instructions, the module returns string information.

1000Hz=998.1Hz 5000Hz=4965.4Hz　　　　/ / Self-test frequency error results

3.18 Sensor Return Signal Quality Optimization

Different manufacturers and different types of vibrating wire sensors will have different signal amplitudes under the same excitation conditions. The output signal amplitude (intensity) of the sensor and the appropriate signal amplification factor must be combined to complete the ideal frequency reading. The vibrating wire sensor outputs a weak sine wave signal after self-vibration. There are two ways to optimize the signal quality of the sensor. One is to adjust the excitation voltage to make the sensor vibrate, and the other is to change the signal amplification factor of the module.

3.18.1 Excitation Voltage Optimization

Under the premise of the module's default signal amplification factor and default parameters, if the sensor signal of the ideal amplitude value cannot be obtained, the excitation voltage value should be adjusted first. When the high-voltage pulse excitation method is used, the excitation voltage can be adjusted by setting the expected voltage register HP_EXP without exceeding the pumping capacity of the module. When the module pumping capacity is exceeded, the input voltage of the pin VSEN must be adjusted. When using a low voltage sweep, the excitation voltage can only be adjusted by adjusting the input voltage to the VSEN pin.

The VSEN pin voltage must not exceed 20V. When using high voltage excitation, the high voltage value must not exceed 220V.

Note: High-voltage excitation is not the higher the voltage, the better, under the premise of satisfying the sensor excitation, try to use a lower excitation voltage. The high-voltage excitation may permanently damage the vibrating wire sensor coil (coil burned).

3.19 Temperature Sensor

The VM5XX module supports an external temperature sensor. Select the type of the external temperature sensor by setting the value of the register TEMP_EX. The real-time temperature sensor measurement value is obtained by reading the register TEMP. The registers TEMP_PAR1 and TEMP_PAR2 are temperature calculation parameters. TEMP_EX.[6:0] defines the type of external temperature sensor. TEMP_EX.[7] defines whether the temperature sensor type is intelligently detected at startup. When the sensor type is the thermistor, TEMP_EX.[15:8] is used to define The nominal resistance of the thermistor in KΩ, TEMP_PAR1.[12:0] is the key parameter B value of the thermistor(This value is requested from the thermistor manufacturer). The register TEMP_PAR2 is the temperature conversion coefficient. The original sampled value is converted to a resistance value.

When the external connection is digital temperature sensor DS18B20, the value of register TEMP is the temperature value read. When external for thermistor, VM5XX module real-time acquisition of the thermistor resistance and according to the preset value of B and nominal resistance calculation temperature, temperature value stored in registers TEMP (integer), unit of 0.1 ℃.

$$\mathrm{T1} = \frac{1}{\frac{\log\left(\frac{R_t}{R}\right)}{B} + \frac{1}{T2}}$$

Thermistor resistance-temperature calculation formula

T1: current temperature value, unit: Kelvin

R_T: current resistance value, module measured in real time

R: nominal resistance value, provided by the resistor manufacturer

T2: The temperature value when the resistance is the nominal resistance, in Kelvin (generally constant: 273.15+25)

B: Key parameters of the thermistor, provided by the resistor manufacturer

According to the "Thermistor Resistance - Temperature Calculation Formula", when the external thermistor is used, it is necessary to first obtain the current resistance value of the thermistor to calculate the temperature using the formula, as described above ("temperature sensor interface" section) the thermistor has two methods: direct connection and indirect connection. The resistance values of the two connection methods are calculated differently.

$$\mathrm{R} = \frac{V_{REF} \times R_0 \times AD}{4096 \times V_{DD} - V_{REF} \times AD} \times \frac{TEMP\_PAR2}{100}$$

AD value-resistance calculation formula (direct connection)

$$\mathrm{R} = \frac{V_{REF} \times AD}{4096} \times \frac{TEMP\_PAR2}{100}$$

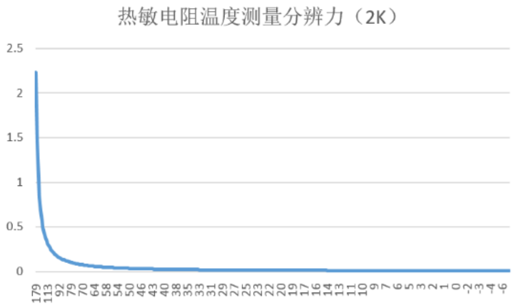
AD value-resistance calculation formula (indirect connection)

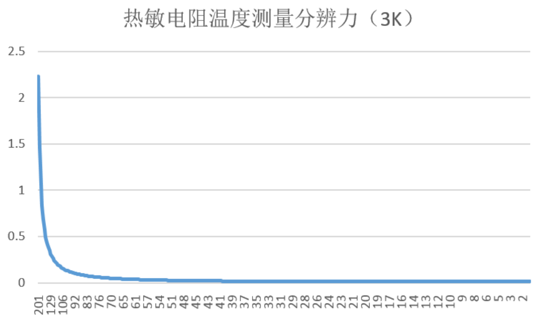
In the above formula: R_0 is the internal pull-up resistor of the module, fixed at 4700Ω (VM4XX module needs external connection); AD is the AD value collected in real time; VREF is the reference voltage, fixed at 2200mV; VDD is the core voltage, fixed at 3300mV.

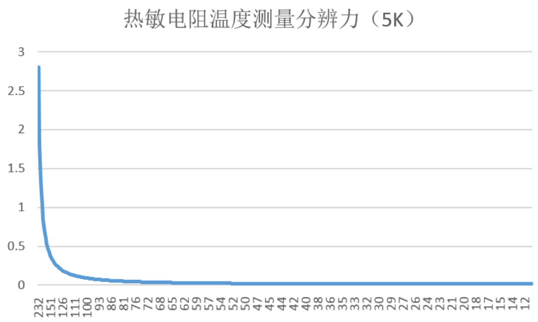
In order to get the correct real-time resistance value, when using the direct connection thermistor measurement method, the module can calculate the resistance value of the thermistor by using the internal voltage divider resistor, which shows strong resistance driving characteristics. The time (TEMP_PAR2/100) should be a resistance correction factor close to ±1.00. When the indirect connection thermistor is used, the partial voltage resistance of the module has lost its function due to the strong output voltage of the differential circuit. At this time, the TMP2 tube is ineffective. The voltage of the foot is completely controlled by the differential measurement circuit, which exhibits strong voltage driving characteristics and the voltage is linear with the resistance. In this case, the register TEMP_PAR2 needs to be modified according to the actual differential circuit to achieve correct voltage-resistance conversion (TEMP_PAR2). The unit of /100) should be mv/Ω.

Directly connect the thermistor range and re

Nominal resistance	Temperature range VM5XX)	
1K		-40~120
2K		-30~200
3K		-23~220
5K		-15~250
Other resistance or temperature range can be customized		







For indirectly connected thermistors, the temperature range is completely determined by the external conversion circuit and is not limited by the temperature ranges listed above.

External temperature sensor type register TEMP_EX (0x1C)

Bit	symbol	value	description	Reset value
bit15:8		1~255	NTC nominal resistance value, unit: KΩ	2
bit7		Whether to intelligently detect the temperature sensor type *		0
		0	no	
		1	Yes	
bit6:0		External temperature sensor type		1
		0	No external temperature sensor (real-time acquisition core temperature)	
		1	DS18B20	
		2	Thermistor (direct connection - resistance characteristic)	
		3	Thermistor (indirect connection - voltage characteristics)	

The intelligent detection function can automatically detect which type of temperature sensor is connected externally, and can detect DS18B20, thermistor or no temperature sensor is connected. When the detection is completed, TEMP_EX.[6:0] is automatically set to the corresponding value. It is not yet possible to detect how thermistors are connected (directly or indirectly), if an indirect connection thermistor is used, set

TEMP_EX.[7] to 0 and TEMP_EX.[6:0] to 3.

Temperature calculation parameter register TEMP_PAR1 (0x1A)

Bit	symbol	value	description	Reset value
bit15:13		0	Reserved, undefined function	0
bit12:0		1000~8000	Thermistor B value, provided by the resistor manufacturer	3950

Temperature calculation parameter register TEMP_PAR2 (0x1B)

Bit	symbol	value	description	Reset value
bit15:0		-32768 ~32767	Resistance conversion coefficient, unit: 0.01 When the thermistor is indirectly connected Voltage value (mV) * TEMP_PAR2 * 0.01 = resistance value When the thermistor is directly connected Sampling resistance value *TEMP_PAR2*0.01=resistance value	100

Temperature value register TEMP (0x29)

Bit	symbol	value	description	Reset value
bit15:0		-32768 ~32767	Temperature register value = temperature value * 10 Unit: 0.1° C, that is, after reading the integer value of the TEMP register, dividing by 10 is the temperature value.	0
Note: When the external temperature sensor type register TEMP_EX is 0, the value in the temperature value register is the module core temperature.				

3.22 Auxiliary function register

Auxiliary function register AUX（0x02）

BITS	symbol	Value	description	default
bit15		UART data bits		0
		0	8bits	
		1	9bits	
bit14:13		UART stop bits		0
		0	1	

		1	1.5	
		2	2	
bit12:11		UART check method		0
		0	NONE	
		1	ODD	
		2	EVEN	
bit10:4		0	Reserved, undefined function	
bit3		1	Enable Half-Power	
bit2		0	Disables vibration avoidance	1
		1	Enable vibration avoidance	
bit1		0	Disable the signal ripple filter function	0
		1	Enable the signal ripple filter function	
bit0		0	Disable analog output	1
		1	Enable analog output	

3.22.1 UART communication parameters

This function can set three parameters of UART communication interface: check bit, data bit and stop bit.

Parity bit: set AUX.[12:11] to 0 to indicate no parity bit, set AUX.[12:11] to 1 to indicate odd parity, and set AUX.[12:11] to 2 to indicate parity.

Data bits: set AUX.[15] to 0 for 8 bits and 1 for 9 bits.

Stop bit: set AUX.[14:13] to 0 means stop bit is 1, 1 means stop bit is 1.5, and 2 means stop bit is 2.

UART Examples of communication parameters

Data bits	Stop bits	Check bits	Register AUX value (binary)	NOTE
8	1	NONE	00000XXX XXXXXXXX	"X" means irrelevant
9	1	NONE	10000XXX XXXXXXXX	
8	1.5	NONE	00100XXX XXXXXXXX	
8	2	NONE	01000XXX XXXXXXXX	
9	1.5	NONE	10100XXX XXXXXXXX	
9	2	NONE	11000XXX XXXXXXXX	
8	1	ODD	00001XXX XXXXXXXX	
9	1	ODD	10001XXX XXXXXXXX	
8	1.5	ODD	00101XXX XXXXXXXX	
8	2	ODD	01001XXX XXXXXXXX	
9	1.5	ODD	10101XXX XXXXXXXX	
9	2	ODD	11001 XXX XXXXXXXX	
8	1	EVEN	00010XXX XXXXXXXX	

9	1	EVEN	10010XXX XXXXXXXX	
8	1.5	EVEN	00110XXX XXXXXXXX	
8	2	EVEN	01010XXX XXXXXXXX	
9	1.5	EVEN	10110XXX XXXXXXXX	
9	2	EVEN	11010XXX XXXXXXXX	

3.22.2Frequency value analog output

The VM module supports the ability to output real-time frequency values from DAO+ and DAO- pins in analog form(Current or Voltage). To use this feature, you need to set the auxiliary function register AUX.[0] to 1, and set the frequency range represented by the analog quantity， DAO_TH.[15:8] is the upper limit of frequency, and DAO_TH.[7:0] is the lower limit of frequency, units are "100 Hz". The default value of DAO_TH register is 0x2100, that is, the maximum and minimum values of the analog amount are respectively 3300Hz and 0Hz(This default value may be different for different versions of firmware, modify these two parameters as needed)。

Analog output frequency rangeDAO_TH（0x19）

BITS	symbol	Value	description	default
bit15:8		0~80	The value of the frequency represented by the maximum value of the analog	33
bit7:0		0~80	The frequency value represented by the minimum analog value	0

Analog output characteristics

TYPE	Analog	Analog range		UNIT	resolution
		Lower limit	Upper limit		
VM3X1/VM4X1	Voltage	0	1000	mV	1/8000
	Current	0	1	mA	
VM5X1	Voltage	0	3300	mV	1/4000
	Current	0	20	mA	
VM5X1 has two voltage output pins, DAO+ output voltage range is 0~3300mV, and DAO- output voltage range is twice DAO+ voltage (i.e., 0~6600mV). When is current output, 0~20mA flows from DAO+ to DAO-.					

The formula of conversion of analog quantity to frequency value is:

$$f = DAO_{THL} + \frac{(DAO_{THH} - DAO_{THL}) \times AN}{AN_{MAX}} Hz$$

In the above formula:

f： Current frequency value, unit Hz

DAO_{THL}: The pre-set minimum analog value (0) corresponds to the frequency value DAO_TH.[7:0]*100Hz

DAO_{THH}: The pre-set maximum analog value (0) corresponds to the frequency value DAO_TH.[7:0]*100Hz

AN: Analog value (voltage or current value) output by the module.

AN_{MAX}: The theoretical maximum of the analog.When is voltage output, take 1000mV for VM3X1/VM4X1 and 3300mV for VM5X1.When output for current, take 1mA for VM3X1/VM4X1 and 20mA for VM5X1.

Example: using the VM511 module, the analog quantity is the voltage output, and the DAO pin voltage is 0.616v, then the frequency value is:

$$f = DAO_{THL} + \frac{(DAO_{THH} - DAO_{THL}) \times AN}{AN_{MAX}} Hz = 1000Hz + \frac{(3000Hz - 1000Hz) \times 616mV}{3300mV} Hz = 1373.33\text{Hz}$$

3.22.3 Signal ripple filtering *

This function is used to filter the ripple interference signal contained in the signal during frequency sampling to avoid the interference signal being collected.Enable this by setting the register AUX.[1] to 1.

This function can filter out the signals with lower amplitude in the return signal of the sensor. It is only suitable for the strong return signal of the sensor and can improve the signal sampling quality to some extent.Enabling this feature when the sensor return signal is weak will result in unsampled frequency data.

3.22.4 Avoid vibration

The excitation signal is sent to the coil only when the inner steel wire of the vibrating wire sensor is at rest, which is conducive to good vibration of the steel wire and can obtain higher quality sampling data. Enable this by setting the register AUX.[2] to 1.When this function is enabled, the reading module will suppress the vibration of the steel wire before excitation and wait for the vibration to stop completely.

3.22.5 Half-Power

Turn off some unnecessary hardware resources when the module is idle to save current consumption.Set AUX.[3] to 1 to enable this feature and 0 to turn it off. (for VM501, VDD current can be reduced from 45mA to 25mA).

4, Parameters Configuration Tool

Normally, Typically, when a module is tested(reading and writing) on the computer side, you can use some common free tools, such as the MODBUS communication protocol-based debugging tool MODSCAN, general-purpose serial debugging assistant, etc. These tools can be downloaded and used through network. It is not listed here one by one.

4.1VMTool Overview

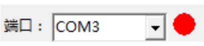
VMTool is a tool specially developed for VM5XX for instruction generation, parameter reading, configuration, module testing, and real-time data reading. This tool has module version identification, parameter import and export and other practical functions, which can realize the rapid use of modules without knowing the communication protocol of modules. In addition, VMTool also provides additional functions such as universal serial port debugging, MODBUS test, real-time data curve drawing, data storage, data network publishing, etc. These functions can be used as small data management software. Please consult our technology for details (0316-3093523 400-096-5525 info@geo-explorer.cn)。

The module firmware for different versions is not identical. Please select a special version program for the module firmware.

The default running interface of VMTool is shown in the figure below. The main interface consists of title bar, instruction area, parameter area, real-time data area, function extension area, status bar and other parts.

Title Bar: Located at the top of the screen, showing the program name and version information and the applicable module firmware version prompts.

Command area: Located on the left side of the main interface, it contains function buttons such as serial port setting, parameter reading, parameter writing, parameter import and export, etc. The actual operation of the module is completed in this area.



Port connection status indicator, red indicates that the port is connected.

Data transmission indicator, blue indicates that data is being sent to VM5XX.

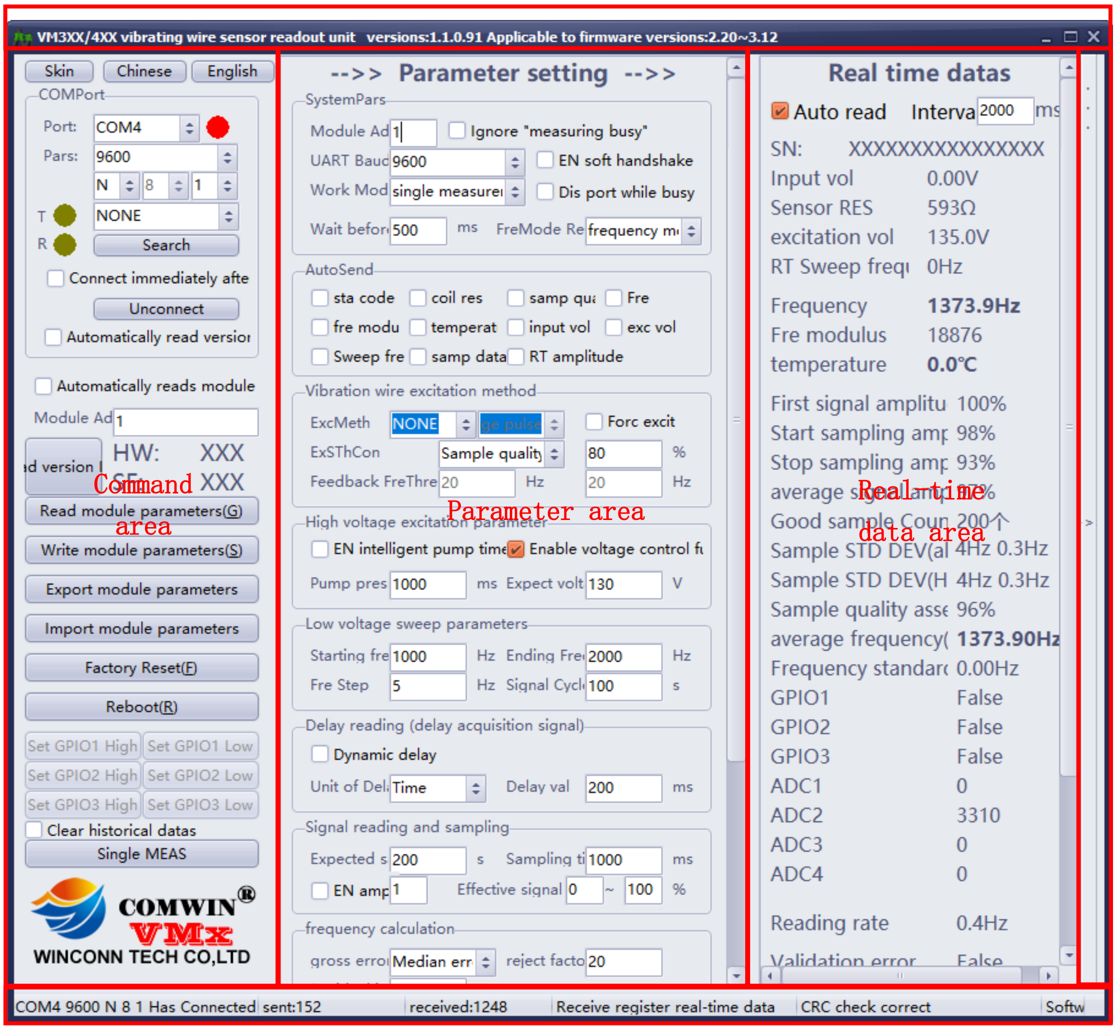
Data reception indicator, red indicates that the data of VM5XX has been received.

Parameter area: Display various parameter information read from the module. You can modify and select parameters in this area, and then use the command area button to complete the operation of the module.

Real-time data area: Displays real-time data (sensor frequency, signal quality, temperature, etc.) returned by the module in read-only form.

Function expansion area: It is located at the far right of the interface. It is not displayed by default. You can switch the display status by double-clicking the extension bar on the right side of the interface.

Status bar: Real-time display of VMTool's various operating states, such as serial port connection, sending and receiving, command sending and module interaction prompts.



VMTool main interface (default streamlined)

4.2 Preparation

(1) Connect the RS232 (or RS485) interface of the VM5XX module to the COM port of the computer;

(2) Connect the vibrating wire sensor and the temperature sensor to the corresponding interface of VM5XX (not required);

(3) Connect the module power supply (DC5~12V);

4.3VMTool Basic Functions

Before performing the following operations or any operation of clicking the button to send an instruction, please try to keep the [Auto Read] check box unchecked to avoid conflicts of instructions sent to the module at the same time.

4.3.1 Connection and Disconnection of Modules

The operation is completed in the [COM Port] combo box in the command area.

[Port] drop-down box: Lists all the COM port names that already exist in the computer. If the port name connected to the module is not listed in the drop-down box, you can enter it freely by manually entering the port name.

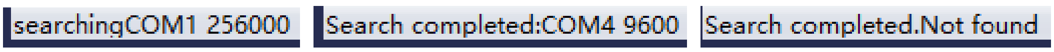
[Rate] drop-down box: Contains the commonly used communication rate, and select the item with the same communication rate as the module (default is 9600bps).

[Search] button: Use all COM ports and communication speeds that may be connected to the module for command detection, and automatically search for the port to which the VM5XX module is currently connected and automatically set to the correct communication rate.

[Connection Module] button: Use the port name of the current [Port] drop-down box and the rate value of the [Rate] drop-down box to perform the "Connection Module" operation. See below for details.

(First) Search module

Click the [Search] button, the program starts to try to send the test command word to the module using all possible ports and communication speeds. The status bar displays the prompt "Searching the COMxx communication rate value". When the correct reply word of the module is found, the search is stopped. The status bar displays "Search Completion COMxx Communication Rate". If the correct module response is not received during the entire search process, the status bar displays "Search Completed Not Found".



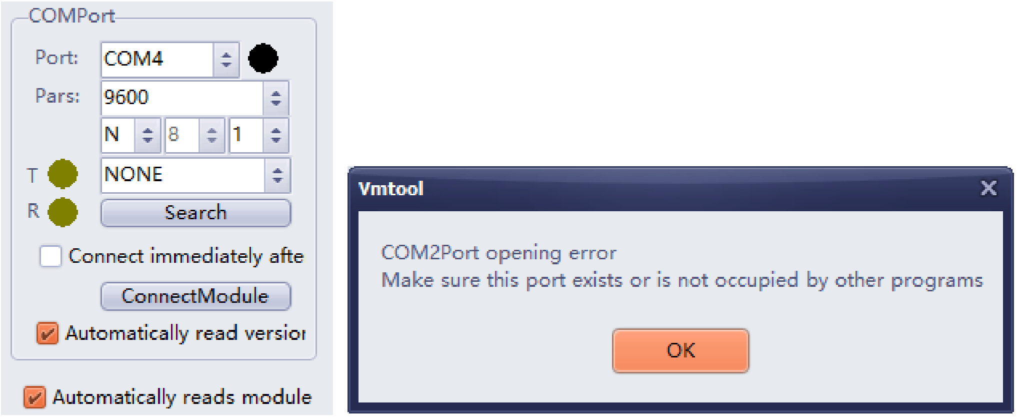
If the check box on the right side of the [Search] button is selected, the following connection module operations will be automatically performed after the module is searched.

(Second) connection module

If the search function in the previous step is not used or the search fails, you need to select the COM port name on the computer connected to the module in the [Port] drop-down box, and select the communication rate in the [Speed] download pull box (the VM5XX module defaults to 9600bps).), click the [Connect Module] button to complete the connection with the module (assuming that the digital interface of the module and the computer has been physically connected and the module is in normal working condition).

When the module is connected, VMTool completes the three tasks of COM port connection, module version reading, and module parameter reading.

(1) COM port connection: According to the selected port name and communication speed, open the COM port of the computer and establish communication channel with VM5XX. If everything is ok, there will be a message like "COM3 9600 N 8 1 Connected" in the status bar. If an error occurs during the connection process, a prompt box will pop up indicating that an error has occurred.



Port connection and error message box

(2) Module version read: If the [Automatically read version information when connected] check box is checked, the firmware version read command is automatically sent to the module, and waits for the version information to be returned. For details, see "4.3.2 Firmware". Version read" section.

(3) Module parameter reading: If the [Automatically read parameters when receiving version information] check box is checked, the parameter reading instruction will be automatically sent to the module after receiving the version information returned in the previous step, and wait the module returns the parameter data, as described in the section "4.3.3 Module Parameter Reading".

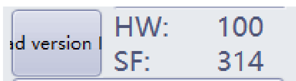
(Third) disconnect module

When connected, the text of the [Connect Module] button is displayed as “Disconnect Module”. Click this button to disconnect VMTool from the module.

When it is in the disconnected state, it cannot communicate with the module. The following contents are all completed in the connected state, and the description will not be repeated.

Note: The port name and communication speed in the [COM Port] combo box will be automatically saved when the program exits, and will be dynamically loaded at the next startup.

4.3.2 Firmware Version Reading



Click the [Read Version] button in the command area to read the firmware version information of the currently connected module. The read version information is displayed on the right side of the button. VMTool will adjust the function and interface according to the version read. Therefore, when using VMTool, the module firmware version should be read first.

4.3.3 Module Parameter Reading

Click the [Read Module Parameter] button in the command area to send a parameter read command (Register 0~31) to the module, and the read parameters are automatically updated to the parameter display area. After the command is sent and the module returns the information, the status bar will have the corresponding prompt message "send command..." and "receive register parameter data".

4.3.4 Modification of Module Parameters

In the parameter area, the current values of all parameter registers of the module are listed. The parameter values are modified on the interface through interface selection and input operations. After the parameter modification is completed, the command area [write module parameters] button must be clicked to set the current parameters. All parameters displayed in the zone are written to the module at one time. For the meaning and function of each parameter in the parameter area, please refer to the detailed description of Chapter 3. Modules with firmware version lower than 3.01 do not support continuous multi-register write instructions. You need to use a single register to modify the instructions. For details, see "Single Register Modification" in "4.4.2 MODBUS Tool Module".

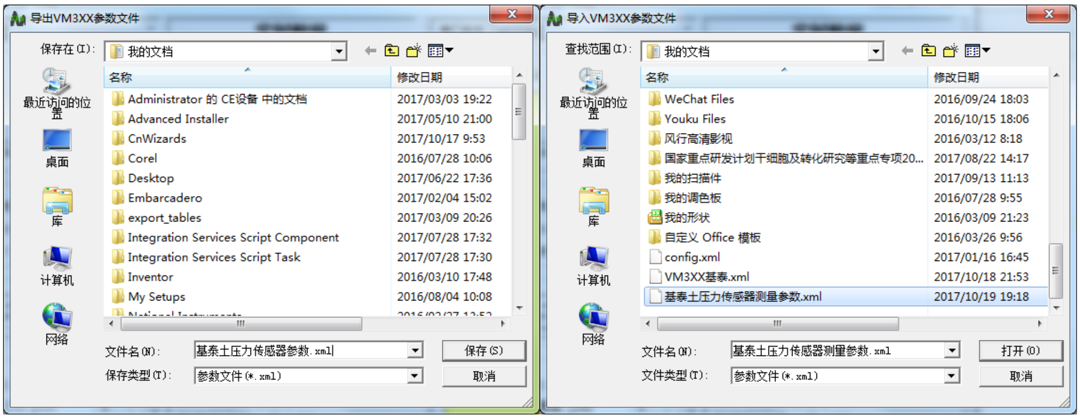
Most of the parameters take effect immediately after the modification is completed, and the modified running state can be directly observed. Only the UART communication rate and signal amplification parameters can take effect at the next startup. If you need to restart the module, you can restart the module by reconnecting the power supply. You can also use the soft command control module to automatically restart by clicking the [Module Reset Restart] button in the command area.

In some cases, the module may not work properly due to incorrect parameter settings. At this time, the module parameters can be reset by clicking the [Restore Factory Settings] button. In some extreme cases, the module cannot receive the command normally. In this case, you need to use the hardware parameter reset method to restore the factory settings. For details, see “3.3 Restoring Factory Parameters”.

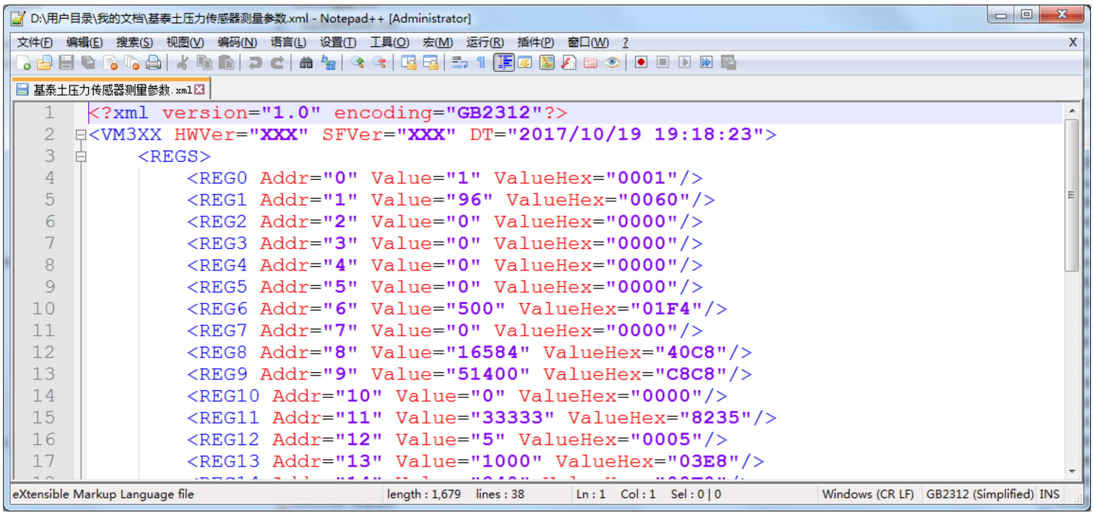
4.3.5 Parameter Import and Export

There are many VM5XX parameter registers, it is not convenient to memorize the specific function definition and value range of each register. In addition, because the vibrating wire sensor manufacturers are heavy, the sensor characteristics and quality are different, different types of sensors may require specific parameter combinations. Effective readings, so VMTool provides parameter import and export function, you can save the configured and verified current parameters as a file, so that you can write to the VM5XX module again when needed.

The parameter import and export are respectively implemented by two buttons: [Export Module Parameter] and [Import Module Parameter] in the command area. The export operation saves the parameters displayed on the current interface as a file, and the import operation displays the specified file to the current interface. To export the module parameters, first click the [Read Module Parameters] button to perform a parameter read, and display the actual parameters of the module to the interface. To import the external parameter file into the module, you need to click [Write] after opening the parameter file. The module parameter] button writes the interface display parameter value to the VM5XX module, and the import and export operations are relatively simple and will not be described in detail. The exported file name is best known for its future use.



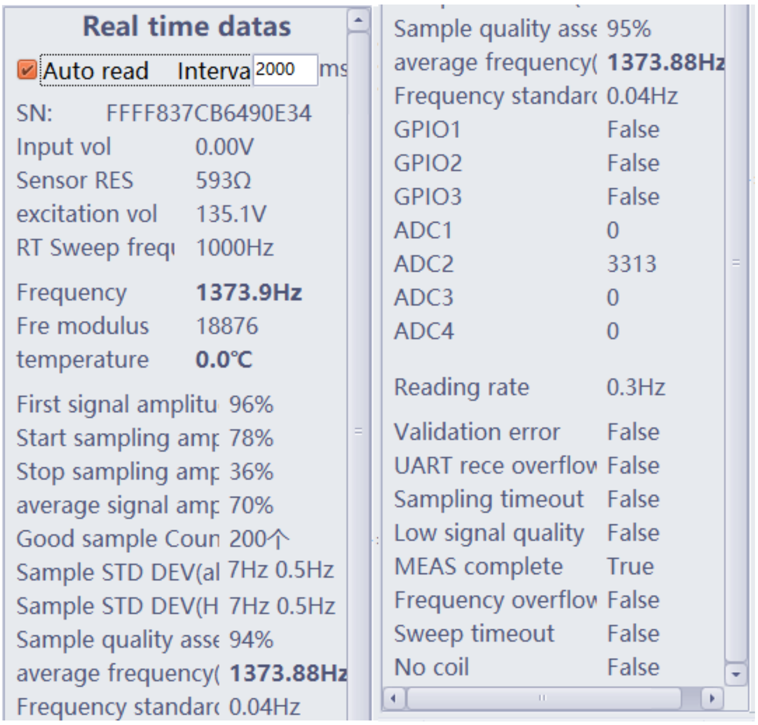
Export and import of module parameters



Module parameter file content

4.3.6 Real-time Data Reading

When VMTool and module are connected (4.3.1 module connection and disconnection), check the [Automatically read] check box in the real-time data area, VMTool will automatically send real-time data read command to the module, modify [time] Interval] The value in the text box changes the time interval between two adjacent read commands, in milliseconds. After the read command is automatically sent, VMTool waits for the module to return real-time data, and does not initiate the next command transmission until the module returns the correct real-time data.



Real-time data reading and display

The description of the relevant registers in Chapter 3 of the physical quantities in the real-time data area.

Measurement Frequency: The data reception rate estimated by VMTool based on the time difference between the last two VM5XX real-time data received. Since the VM5XX module adopts the measurement priority work strategy, if the module receives an instruction when it is "busy", it will wait for the current measurement to complete before responding and executing the instruction, returning the response information of the instruction. Therefore, when using the "introduction" in this section When the function is automatically read, the reading interval is set to a value smaller than the actual measurement rate of the module. At this time, the measurement result of the module is read every time. In this case, the calculated data receiving rate is actually the module. The reading rate. For example, if the reading rate of the module is about 1 Hz, the automatic reading interval should be set to a value lower than 1000 milliseconds, which is recommended to be 500 ms. The measured frequency value obtained is the module reading rate. When the time interval is set to a value greater than 1000ms, the obtained measured frequency value can only indicate the update rate of the interface, and is not directly related to the reading rate of the module.

GPIOx: Displays the current state of a general-purpose IO pin. True means high level and False means low level.

Operation status: including several operation status flags such as instruction check, sampling timeout, True indicates that the status is true (or "Yes"), and Falsh indicates that the status is false (or "No").

4.3.7 Software Handshake Protocol

VMTool supports the software handshake protocol with the module. Under the premise that the module enables the software handshake (see “Description of Software Handshake in Section 3.7”), VMTool can use the software protocol to implement the module only when the module is idle. Send an instruction. Select the [Software Handshake] option by clicking the handshake protocol in the command area of the main interface to open the software handshake function of VMTool. Before the software handshake is enabled, the module usually does not respond immediately after sending the command to the module (the data reception indicator does not flash immediately after the data transmission indicator flashes). After the software handshake is enabled, the data transmission indicator blinks after the data reception indication is observed. The device will also flash immediately, i.e. the command sent to the module gets an immediate response.

If VMTool has enabled the software handshake function and the VM5XX module does not enable this function, VMTool will be in the command ready to send status for a long time because it has not received a valid software handshake protocol signal, and will send the module to the module after waiting for the timeout. This process can be observed in the VMTool status bar, as shown in the following figure.

4.3.8 Generating Register Values

VM5XX has a lot of bit-by-bit registers. It can be easily set using the VMTool tool. When you need to know the actual value of the register, you can get it in the following two ways. (Keep the [Auto Read] check box unchecked)

(1) When the vibrating wire module is connected to the VMTool tool

Parameter setting on the interface;

Click the [Write Module Parameters] button to write the parameters displayed on the current interface to the module.

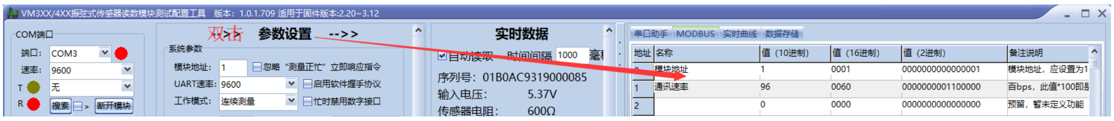
Click the [Read Module Parameters] button, the module's register value will be automatically updated to the table in the MODBUS display area.

(2) When the vibrating wire module is not connected to the VMTool tool

Parameter setting on the interface;

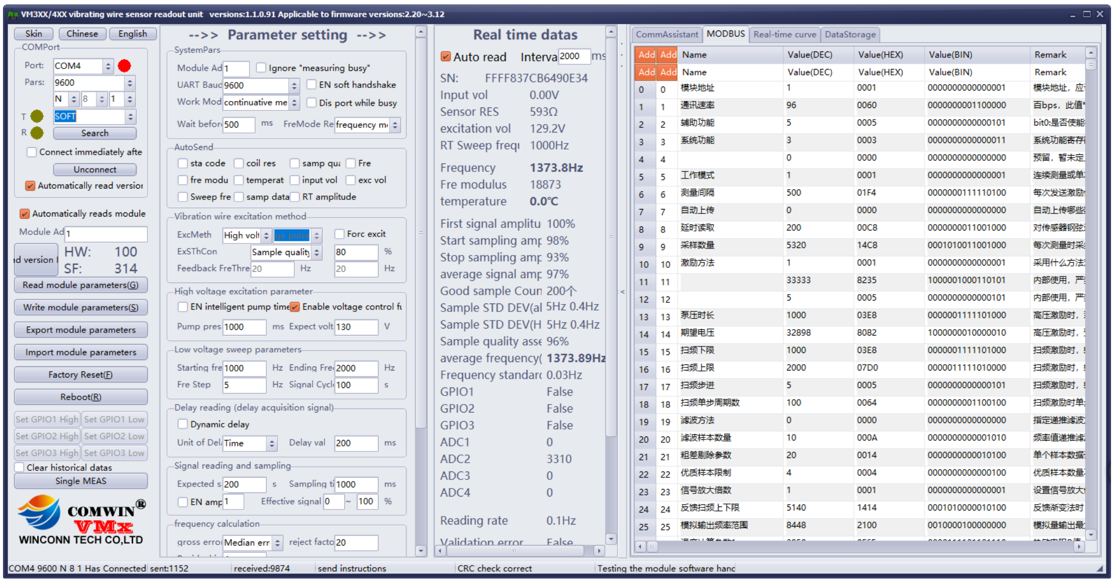
Double-click the title "Parameter Setting" text in the [Parameter Settings] area;

The parameters displayed on the current interface are automatically updated to the table in the MODBUS display area;



4.4VMTool Extension

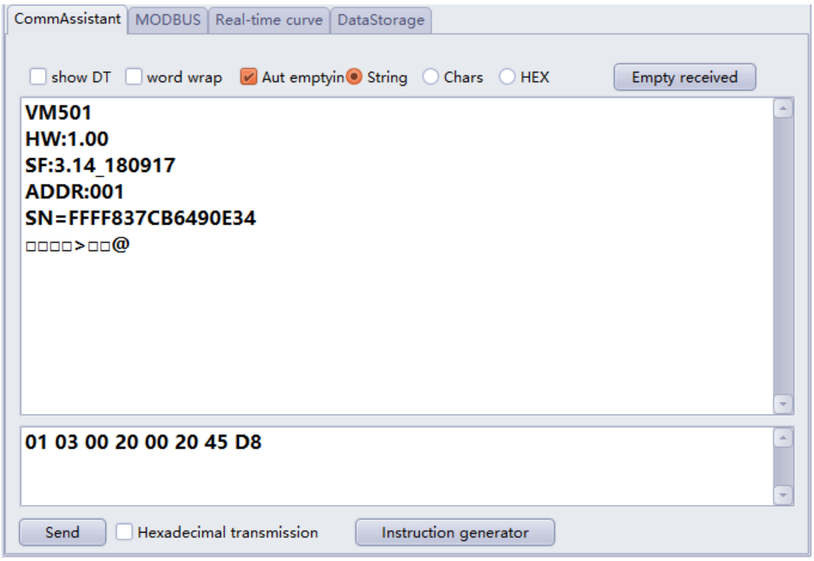
Double-click the extension toolbar on the right side of the main interface to display and hide the extended ribbon. Extended functions include serial port debugging, MODBUS, real-time curve and data storage. The extended function area displays the following effects.



VMTool main interface (extension)

4.4.1 Universal Serial Port Debugging Module

The serial port debugging module directly uses the currently connected COM port to display the received data content in real time, and provides the command manual sending function, as shown in the following figure.



Serial debugging panel

The serial debugging panel consists of the upper receiving area and the lower sending area. Both sending and receiving support both string and hexadecimal data formats.

Show time check box: Displays real-time computer date and time information before displaying the received data.

Auto Clear check box: When the content of the receiving area is over limit, the entire receiving area is automatically cleared.

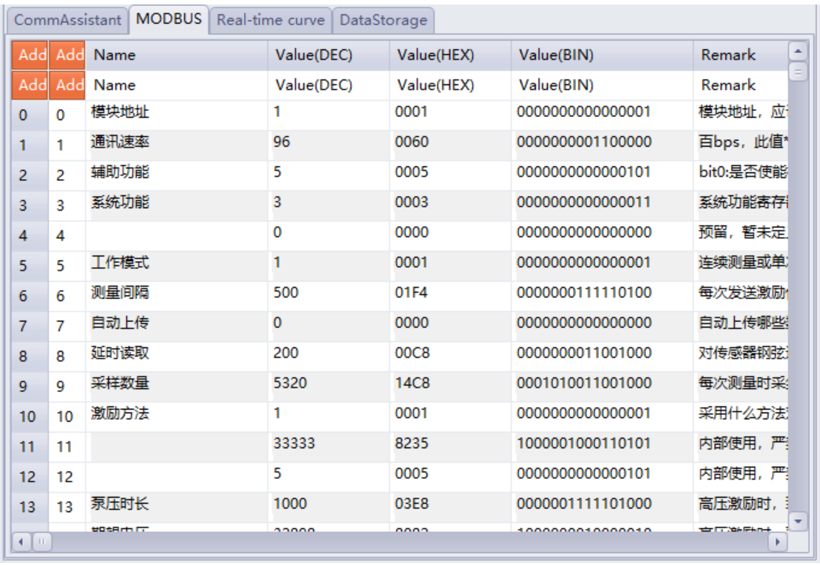
Clear Receive Area button: Clears the entire receive area directly.

The sending area has two functions. One is to manually input the command content, click the [Send] button to realize the manual sending of the command. In addition, all the commands automatically sent during the VMTool working process will be displayed in the sending area to facilitate the observation of the specific Instruction content. For example, after clicking an instruction button, the sending area displays the content of the instruction sent to the module after the button is clicked.

4.4.2 MODBUS Tool Module

(1) Register view

This function module provides standard MODBUS protocol register display and single register modification function. Switch to this module by clicking the [MODBUS] tab of the extended function area, as shown in the figure below.



Add	Add	Name	Value(DEC)	Value(HEX)	Value(BIN)	Remark
Add	Add	Name	Value(DEC)	Value(HEX)	Value(BIN)	Remark
0	0	模块地址	1	0001	0000000000000001	模块地址，应
1	1	通讯速率	96	0060	0000000001100000	百bps，此值
2	2	辅助功能	5	0005	0000000000000101	bit0:是否使能
3	3	系统功能	3	0003	0000000000000011	系统功能寄存
4	4		0	0000	0000000000000000	预留，暂未定
5	5	工作模式	1	0001	0000000000000001	连续测量或单
6	6	测量间隔	500	01F4	0000000111110100	每次发送激励
7	7	自动上传	0	0000	0000000000000000	自动上传哪些
8	8	延时读取	200	00C8	0000000011001000	对传感器钢弦
9	9	采样数量	5320	14C8	0001010011001000	每次测量时采
10	10	激励方法	1	0001	0000000000000001	采用什么方法
11	11		33333	8235	1000001000110101	内部使用，严
12	12		5	0005	0000000000000101	内部使用，严
13	13	泵压时长	1000	03E8	0000001111101000	高压激励时，

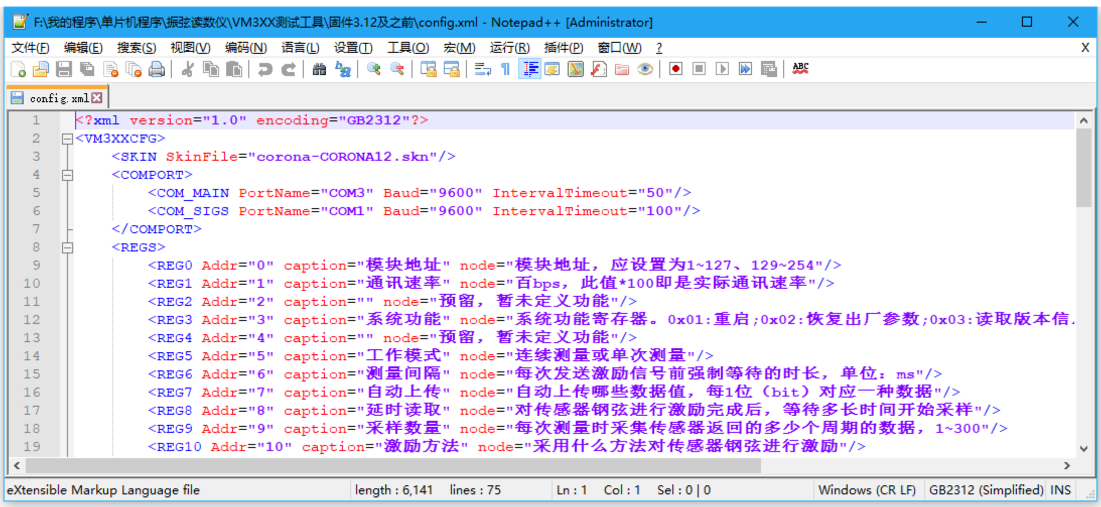
MODBUS panel

This module displays all the registers of VM5XX in tabular form, including register address, name, different hexadecimal values, and register comment description information.

(2) Register name customization

The register name and comment description information can be modified by modifying the config.xml located in the same path of the tool program. The file contents are as shown below.

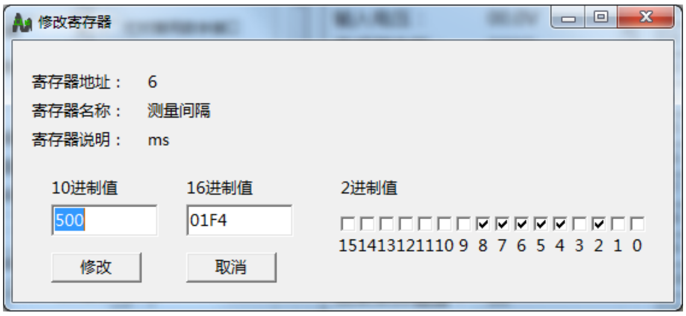
In the figure, the multi-line content located in the [REGS] node is the register definition information, each row corresponds to a register, Add is the register address, caption is the register name, and node is the remark description information of the register. VMTool dynamically reads the contents of this file and displays it in the interface table every time it is started. It can be modified, added or deleted as needed. The registers not in this file are displayed in blank in the interface table.



Register description configuration file

(3) Modification of single register value

In the MODBUS function module, the register value modification window can be called up by double-clicking on a cell, as shown in the figure below.



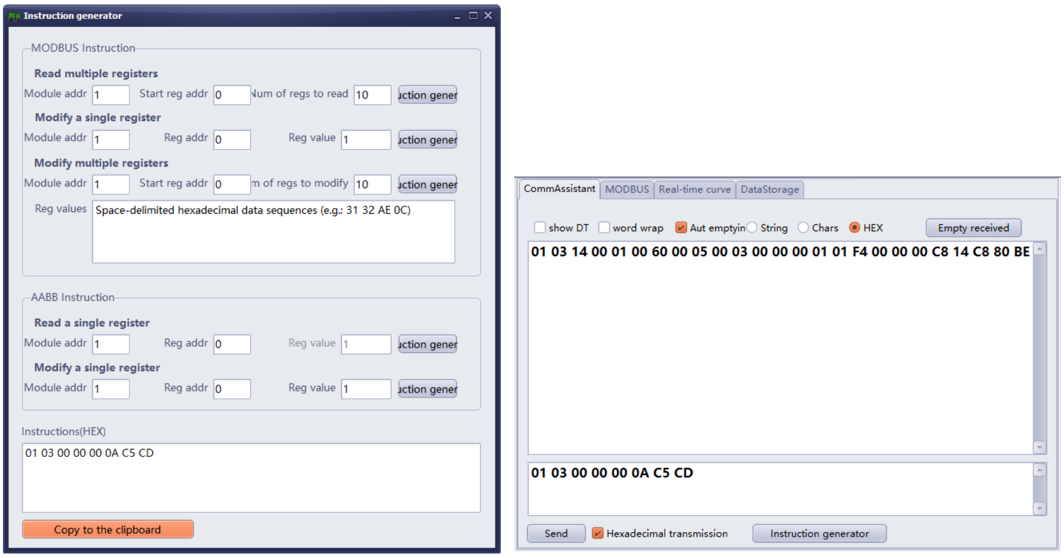
Single register modification window

According to the parameter modification needs, you can choose one of the hexadecimal changes. When you modify one type of hexadecimal data, the other two hexadecimal data will also be updated synchronously. Click the [Modify] button to send a register modification instruction to the module to complete the single register value. Modification operation.

4.4.3 Instruction Generator

(1) Instruction generation

The command generator can generate read and control commands that conform to the MODBUS and AABB communication protocols as needed. The command generator window can be opened by clicking the [Command Generator] button in the serial port debugging tool, as shown below.



Instruction generator and instruction test

In the command generator window, input the register address and register value that need to be modified or read. Click the [Generate Command] button to generate a hexadecimal command in the text box at the bottom of the interface, and click the [Copy to Clipboard] button. The currently displayed instruction content can be copied to the clipboard.

(2) Command test

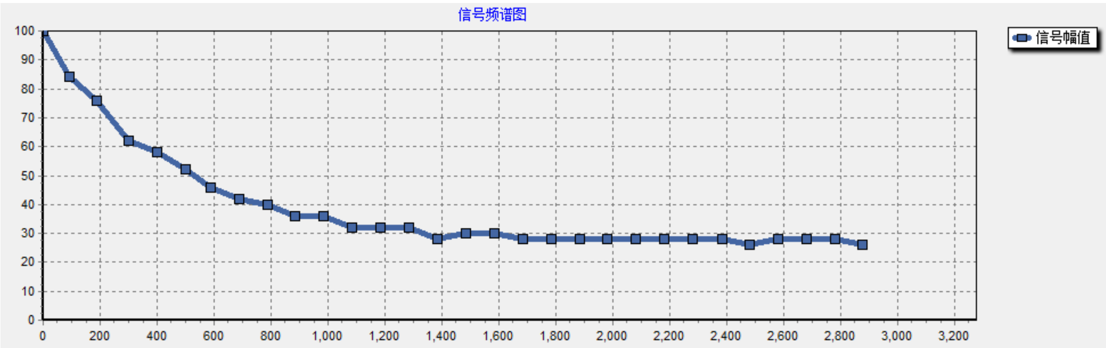
You can paste the generated command into the sending area of the serial debugging tool, click the hexadecimal sending, and click the [Send] button to send the command to the module to verify the correctness of the command.

4.4.4 Real-time Curve

The real-time curve provides a graphical rendering function for several types of real-time data collected, including real-time amplitude spectrum of sensor signals and curve drawing of selectable data types.

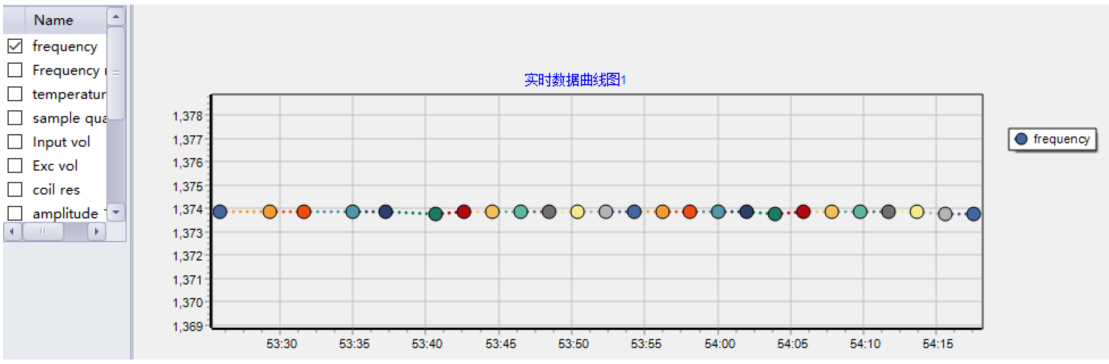
(1) Signal spectrum plotting

To upload the signal amplitude data and sample data uploaded by VM5XX in real time to the art board, you need to open the “real-time amplitude” in the automatic upload.

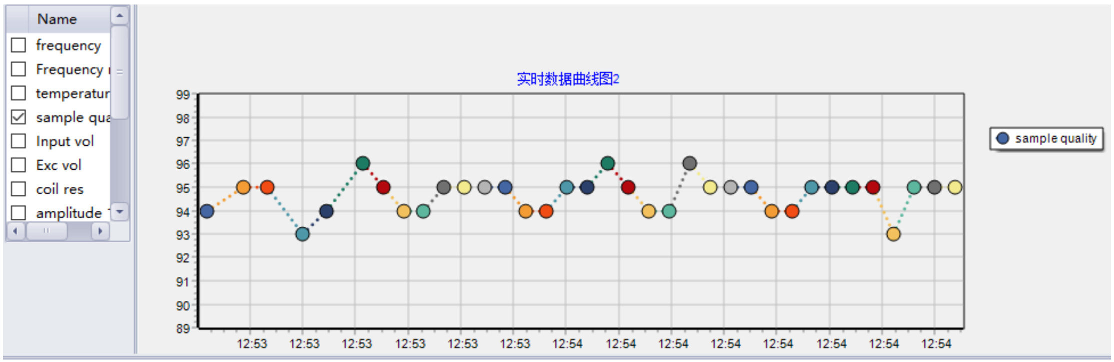


Real-time amplitude graph

(2) Real-time curve drawing



Real-time data graph - frequency value



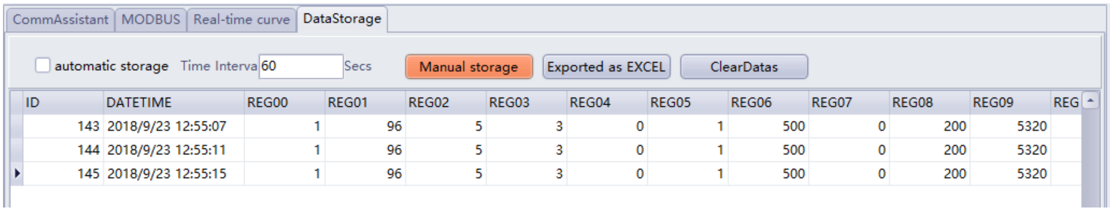
Real-time data graph - sampling quality

4.4.5 Data Storage

The data storage function module supports automatic or manual storage of real-time data register values to the database and supports export to Excel file functions.

(1) Manual storage

Each time you click the [Manual Storage] button in the data storage panel, the current register real-time value is added to the database, as shown in the figure below.

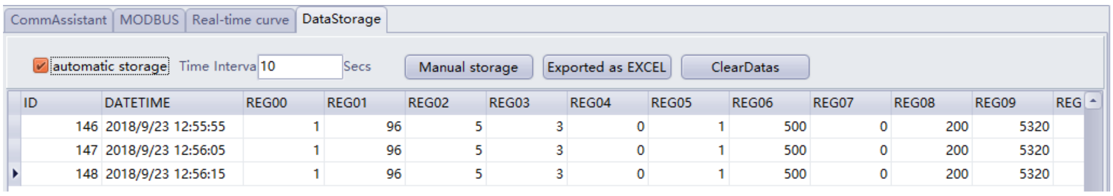


ID	DATETIME	REG00	REG01	REG02	REG03	REG04	REG05	REG06	REG07	REG08	REG09	REG
143	2018/9/23 12:55:07	1	96	5	3	0	1	500	0	200	5320	
144	2018/9/23 12:55:11	1	96	5	3	0	1	500	0	200	5320	
145	2018/9/23 12:55:15	1	96	5	3	0	1	500	0	200	5320	

Manual storage

(2) Automatic storage

Enter the time interval value (in seconds) of the automatically stored data in the time interval text box, check the [Automatic Storage] check box, and then automatically store the register data once every specified interval, as shown below.

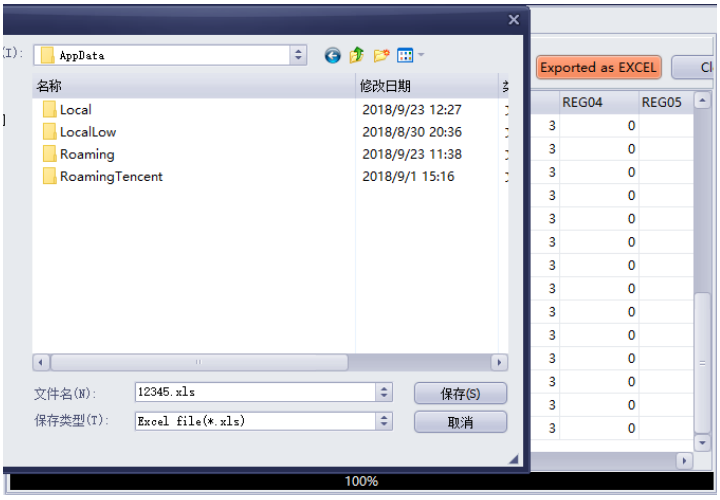


ID	DATETIME	REG00	REG01	REG02	REG03	REG04	REG05	REG06	REG07	REG08	REG09	REG
146	2018/9/23 12:55:55	1	96	5	3	0	1	500	0	200	5320	
147	2018/9/23 12:56:05	1	96	5	3	0	1	500	0	200	5320	
148	2018/9/23 12:56:15	1	96	5	3	0	1	500	0	200	5320	

Automatic storage

(3) Export data

Export all data displayed in the current interface data table to an Excel file for further data processing. Click the [Export to Excel] button to pop up the file save window, select the save path, enter the file name to be exported, click the [Save] button, and the progress prompt of the export will be displayed at the bottom of the data storage panel, as shown below.



(4) Clear data

Click the [Clear Data] button to delete all the stored data. This operation cannot be resumed. Before the operation, please confirm that the data has been exported.

4.5 Precautions when using VMTool tools for different firmware versions

VMTool is developed synchronously with firmware SF301. The SF301 firmware program has more function additions and performance improvements under the pre-compatibility of the previous version. In particular, the refinement of registers is greatly changed. Therefore, pay attention to the module when using the VMTool tool. The firmware version number and the supported functions are different when the version number is different.

SF251 and previous firmware

1. MODBUS continuous multiple register write instructions are not supported. The modification of the register can only be modified by VMTool's MODBUS function module.

2. The version information is not supported during the running process. The module will actively output the version information when the device is powered on. You can re-power the module when the VMTool serial port is connected. VMTool will automatically read the version number and will not Supported feature blocking.

3. GPIO and general purpose ADC operation are not supported.

4. Automatic upload of signal amplitude is not supported.

5. Does not support dynamic voltage regulation in high voltage excitation.

6. Other unsupported functions will disable the interface part function (grayed out) after

reading the module firmware version.

5, Quick Test

This chapter demonstrates reading the vibrating wire sensor data on a computer using the VMTool tool. Suppose your computer already has at least one free COM interface.

5.1 Checking the COM Interface Name

Right click on "My Computer" on the operating system desktop, select [Properties], the computer properties dialog box will pop up, click the [Device Manager] button on the left to pop up the Device Manager window, as shown below.

Find [Ports (COM and LPT)] in the Device Manager window, click on the ▷ icon to expand (see above), view all COM interfaces on the computer, and decide which interface to use to connect to the VM311 module.

5.2 Connecting the VM311 Module

(1) Connection data interface

According to the actual digital interface type, connect several pins of the UART interface of VM311 to the computer digital interface, and ensure that the computer COM interface is consistent with the UART interface level type of VM311, as shown in the following table.

VM311 and computer digital interface connection - RS232

computer COM interface pin	VM311 Interface pin	Description
2	TX/B	The computer receive pin is connected to the transmit pin of the VM module.
3	RX/A	The computer transmit pin is connected to the receive pin of the VM module.
5	GND	Unified reference to "logically"

When VM311 is an RS485 interface, only two data lines need to be connected to the

RS485 interface of the computer. The 485 interface of the computer is generally converted by the RS232 or UAB converter, and the interface has a clear pin identification (A, B).

VM311 and computer digital interface connection - RS485

computer 485 interface pin	VM311 Interface pin	Description
B	TX/B	A is connected to A, B is connected to B, and there is no need to cross
A	RX/A	

When the digital interface of the module is TTL level, if it is connected to a computer, the computer needs to use a USB to TTL level converter. The wiring method is as follows. The TTL converter has an explicit pin name designation (TX, RX, and GND).

VM311 and computer digital interface connection - TTL

computer USB to TTL interface	VM311 Interface pin	Description
RX	TX/B	The computer receive pin is connected to the transmit pin of the VM module.
TX	RX/A	The computer transmit pin is connected to the receive pin of the VM module.
GND	GND	Unified reference to "logically"

(2) Connecting the sensor

Connect the two coil leads of the vibrating wire sensor to the SEN+ and SEN- pins of the VM311 module. Usually, you can connect without any positive or negative electrode.

(3) Connect the module power supply

Connect to the VIN and GND of the VM311 using a 5V DC power supply. The positive supply is connected to the VIN pin and the negative terminal is connected to the GND pin of the module. If everything is normal, you can observe that the VM311 running indicator starts to flash (1~2 seconds).

VM5XX peripheral connection diagram (dotted line is optional)

Please confirm the positive and negative poles and voltage of the power supply before connecting to the module. The module has no reverse power protection measures. If the power supply is reversed or the voltage is not in the applicable range of the module, the module may be damaged and cannot be used.

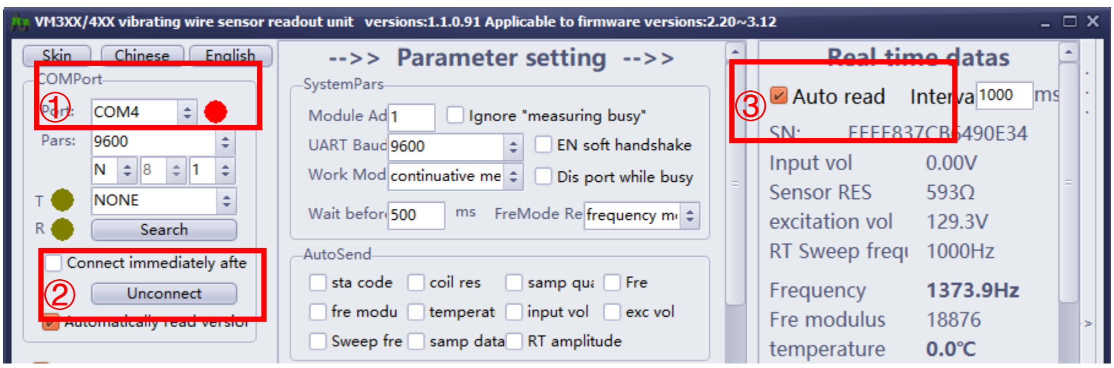
5.3 Sensor Data Reading

Open the VMTool tool and do the following.

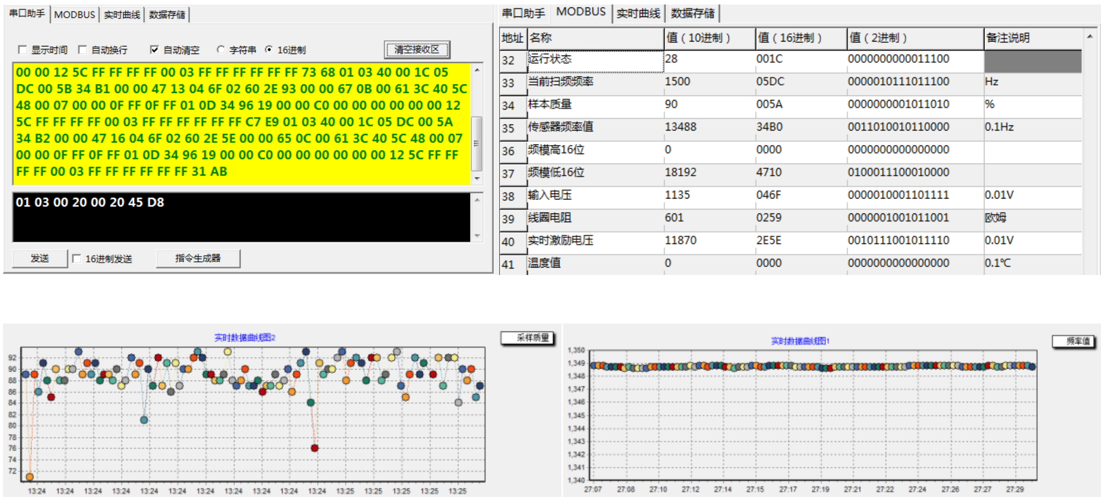
(1) In the upper left corner of the interface, select the computer COM interface name (if there is no serial port name in the optional drop-down box, you can directly input it manually);

(2) Click the [Connect Module] button. If the connection is successful, the button text will change to “Disconnect Module” and automatically read the version information and parameter values of the currently connected module.

(3) Checked the [Automatically read] check box at the top right of the interface. After 2~3 seconds, the measured sensor data can be displayed on the interface.



On the right side of the real-time data (the default is hidden, see "4.4VMTool extended function"), you can view the original received data (hexadecimal) in the serial port assistant, and view the real-time register value in the form of a table in the MODBUS tool. View real-time data curves.



Real-time data viewing in different ways

6, Common Problems

6.1 The Computer Cannot Communicate with the Module

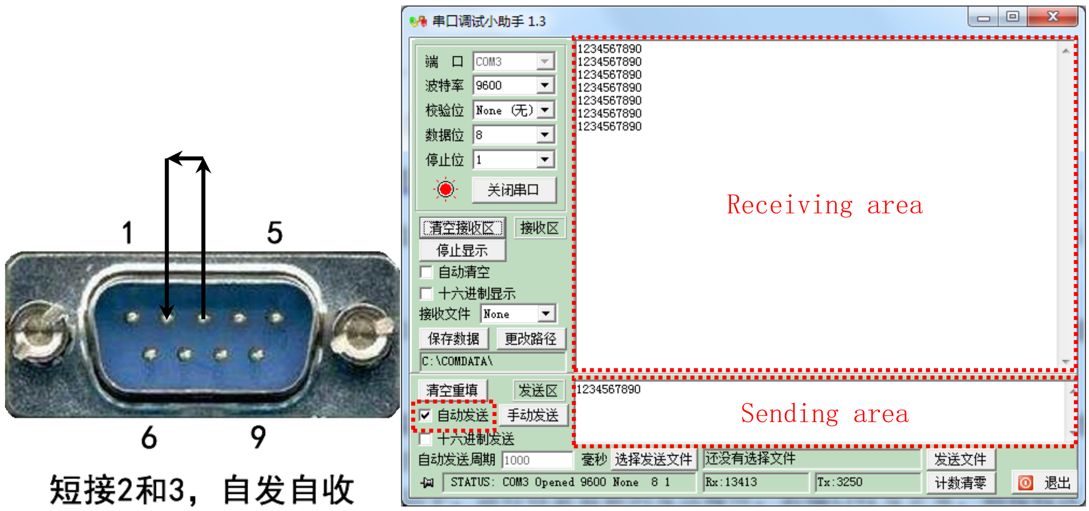
The problem should be checked one by one by the following steps

(1) Observe whether the state lamp of the vibrating wire module flickers normally. If it does not, it is basically determined to be the problem of the module. At this time, you should try to restore the factory parameters of the module.   In high-speed measurement, because the module is "busy" and does not respond to the serial port command, it will also cause the communication to be abnormal. At this time, you can try to send the command multiple times or restore the factory parameters.

(2) Check if the module digital interface type is consistent with the computer COM interface type (RS232 or RS485 or TTL).

(3) Check whether the three wires of the module digital interface and the computer COM interface are correctly connected (the two wires are the RS485 interface).   See "5.2 Connecting VM Modules" for details.

(4) Check if the COM port of the computer can send and receive data normally. Disconnect the COM interface of the computer from the physical connection of the module, short the two pins of the COM interface of the computer COM interface (the DB9 interface of RS232 should be pins 2 and 3), open any serial debugging tool, and perform arbitrary data. Send operation, if the port is sent and received normally, the receiving area will receive the content sent by the sending area, as shown below.



(5) Try different communication rates.   Use the serial port debugging assistant, set the receiving area to “string (not hexadecimal display)”, set the COM port to different communication speed, keep the physical connection between the module digital interface and the computer COM port, and power off the module. Electrical operation, if the serial debugging assistant can not receive the correct startup information, continue to change

the communication rate of the COM port.

(6) Restore the module factory parameters. Change the communication speed of the COM port of the host computer test tool software to 9600, disconnect the power of the module, press the KEY1 button on the module (or short the module TMP1 and GND), turn on the power of the module, and release the KEY1 button after about 500mS. Observe whether the test tool software has received the startup information of the module.

6.2 Sensor Frequency Value is Unstable

The following problems are gradually checked out under the premise of the factory default parameters. If the module parameters are modified, the factory settings should be restored first.

(1) Observe the sampling quality assessment register data. If it is less than 90%, the sensor signal quality can be basically determined to be poor. If the quality is high, the measured data is real sensor data.

(2) Switch to the high voltage excitation method (default value), observe the excitation voltage value, the excitation voltage should be 100V or more. If the excitation voltage is lower than this value, check whether the VSEN pin voltage is normal (3~20V).

(3) Check the resistance of the sensor coil measured by the module. This value should be hundreds of ohms or several thousand ohms (usually 500~600Ω). If the resistance is small, check if the sensor is short-circuited. If the resistance is large, check if the sensor is open (not really connected to the module).

(4) Try to set a higher excitation voltage (see "Power Interface", "High Voltage Excitation Method", etc.).

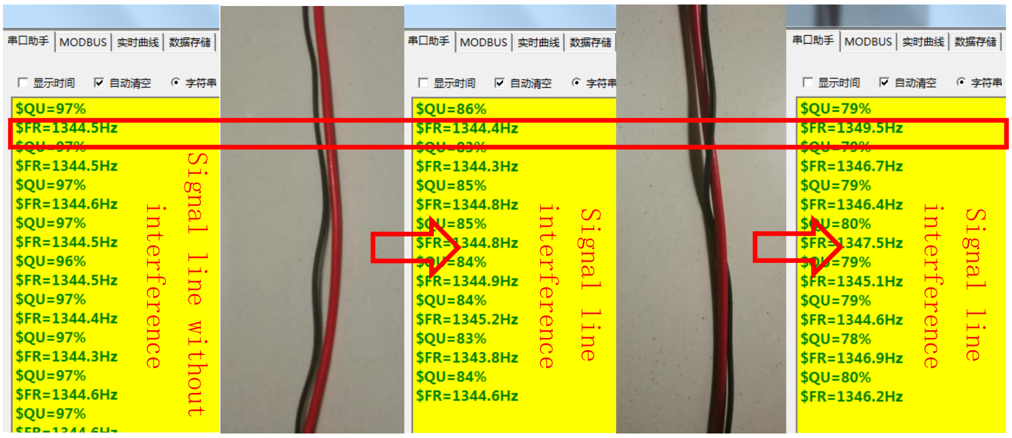
(5) Try adjusting the magnification resistor and use a higher signal amplification factor.

(6) When using in AC/DC hybrid environment, the module must be reliably grounded.

(7) Using the battery to power the module or to replace different types of power adapter, the power adapter will string the ac into the vibration wire signal, and when serious, it will not work normally at all.

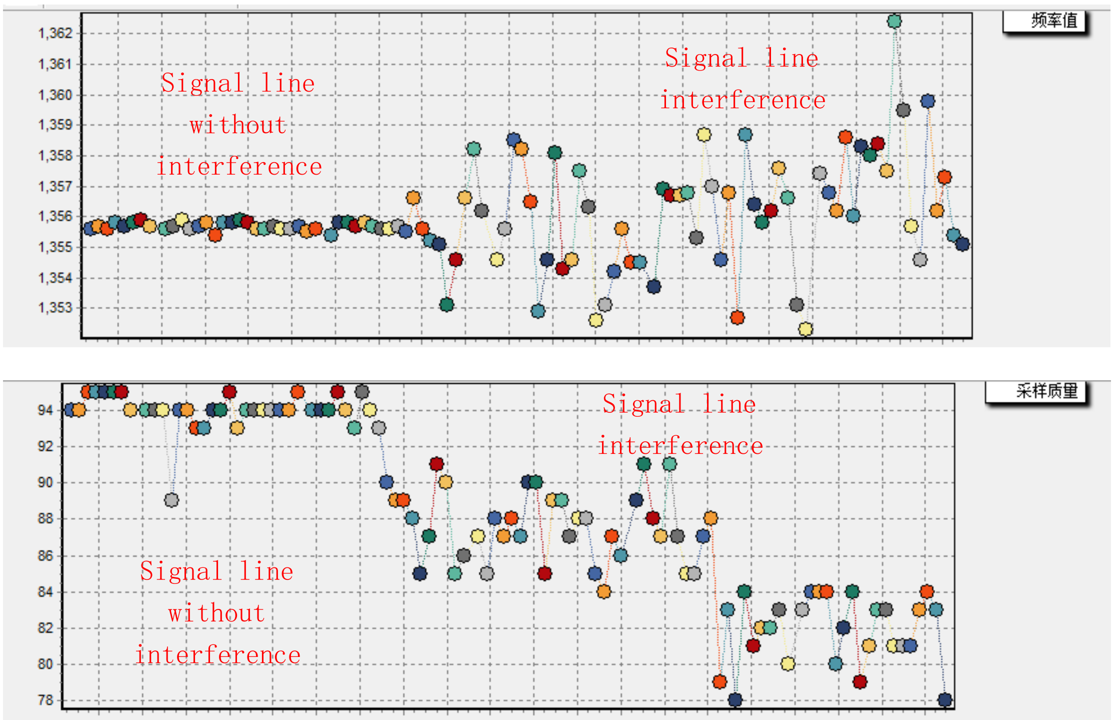
(8) Use a shorter signal transmission line (it is recommended to use a signal transmission line of no more than 50 meters when troubleshooting).

(9) It is strictly forbidden to contact the sensor signal line with other live lines (including other weak current or signal lines). The figure below compares the measurement accuracy of the sensor with separate traces of the vibrating wire sensor line and crossover with other weak electrical signals.



Influence of different wiring methods on signal quality

In the above figure, the left side is the measurement result that is not interfered by other signals, the signal quality is 97%, the vibrating wire frequency is basically pulsating at 0.2 Hz after the decimal point, and the middle is the measurement result of the parallel line of the vibrating wire sensor signal line and the 5V DC wire. The signal quality is reduced to about 80%, the data jitter is up to 1.5 Hz, and the right side is the measurement result of winding the 5V wire and the vibrating wire signal line, and the data jitter is up to 5 Hz.



Signal quality comparison when there is interference

In addition, the frequency change of the vibrating wire sensor is also highly susceptible to vibration. If there are constructions and large vehicles running around, it will also cause fluctuations in the measured value. This is a problem that the sensor of the vibrating wire principle cannot avoid. It can be used for multiple acquisitions. The software filtering adjustment method removes this random interference.

6.3VMTool Communication Error

The status bar prompts "MODBUS data length error: xx, xx".

This problem is usually caused by the low configuration of the computer or the use of unsound hardware to receive data. The processing methods are: Open the config.xml file under the same path as VMTool using the text view software, and modify the IntervalTimeout value as shown in the figure below to be larger, such as "20".

```xml
<?xml version="1.0" encoding="GB2312"?>
<VM3XXCFG>
    <COMPORT>
        <COM_MAIN PortName="COM3" Baud="9600" IntervalTimeout="10"/>
        <COM_SBQ PortName="COM1" Baud="9600" IntervalTimeout="100"/>
    </COMPORT>
    <REGS>
```

After modifying the configuration file, click save and restart the VMTool. Under the premise of ensuring normal communication with the module, IntervalTimeout value should be as small as possible, which is conducive to improving data transmission efficiency.. Excessive IntervalTimeout value will result in abnormal data reception during fast measurement.

6.3 Other Issues

The main difference between civil and industrial modules

The industrial or civil grade is mainly distinguished by the operating temperature range of the module. The civil grade can basically guarantee the measurement accuracy in the range of 0~65 °C (it can still work below 0 °C, but the accuracy will be affected more obviously), and the industrial grade has more High measurement accuracy and wide operating temperature range (-40~85°C).

Precautions when measuring remotely

When the distance between the module and the sensor is far, it is recommended to use a cable with excellent shielding performance for connection. The cable conductor is not less than 0.3 square. The quality of the signal line (especially the shielding layer) and the field wiring directly affect the data reading of the vibrating wire sensor. When both conditions are ideal, the distance between the sensor signal line and the sampling module can reach several kilometers. The wire resistance also affects the strength of the signal. The 0.3 square cable has a resistance of about 70 Ω per kilometer. When measuring, the resistance is 70*2=140 Ω, and the resistance of the vibrating wire sensor coil is generally Approximately 500 Ω will produce a large partial voltage effect, reducing the amplitude of the effective excitation signal to the coil. When measuring the sensor thermistor, the above-mentioned wire resistance problem also occurs, resulting in a large measured resistance value and a low temperature value (the thermistor is a negative temperature coefficient resistor).

Why is the frequency of the sensor getting smaller and smaller?

When continuously exciting, the sensor frequency will decrease slightly, which is a normal phenomenon (It has to do with material mechanics), basically at 1~2Hz.

Why the resolution 0.1Hz but the accuracy is can reach 0.2Hz?

The output frequency value resolution is 0.1 Hz in order to use a single 16-bit integer to represent a value within 6000 Hz, simplifying data reading. If an output format that retains two decimal places is used, the 16-bit integer will overrun. In fact, floating point is adopted in the module, and the frequency resolution is much higher than 0.1Hz. Firmware version V3.12 has added the sensor frequency value high resolution register, can display 0.01Hz.

Why is the frequency value accuracy read is not 0.2?

The reading accuracy of the measuring module can only be measured by standard signals (such as high-precision signal generators). When the sensor is actually connected, it is affected by various factors such as the accuracy of the sensor itself, the interference of the field, and the attenuation of the signal transmission. The accuracy of the signal received by the module itself is degraded. In general, low signal amplitude indicates insufficient excitation, and poor sampling quality indicates more interference. In addition, the lower the frequency, the higher the measurement accuracy, and vice versa.

Can I measure low frequency sensors (eg 300Hz)?

Yes, this module supports frequency acquisition from 30Hz to 12000Hz. It should be noted that the low frequency sensor takes longer to sample, and a reasonable sampling timeout period should be set according to the actual frequency and the expected number of samples.

Can the measurement frequency (reading speed) be higher?

The module is currently only capable of reading speeds of up to 20 Hz, which is sufficient for most applications. High-frequency data acquisition is generally used for vibration measurement. Due to the measurement principle of the vibrating wire sensor, the external vibration itself is a kind of interference to the sensor. Therefore, the vibrating wire sensor is only suitable for measuring static force or displacement and other physical quantities. To measure vibration, in engineering applications, the acceleration sensor should be used for vibration measurement, not the vibrating wire sensor. Readings of dozens or hundreds of times a second have no application significance.

7, Appendix

7.1 Function Code

function code	Function Description
0x0001	Reset restart
0x0002	Restore factory parameters
0x0003	Read firmware version information
0x001x	Single measurement
0x003x	Single measurement, clear historical data before measurement
0xC0	Sweep frequency self-calibration parameter clear
0xC1	Sweep frequency self-calibration
0xC2	Sweep frequency error self-test

7.2 Error Code

No

7.3 ErrMsg

Prompt content	Description
CRC Err	A parameter verification error has been detected and has been restored to the factory value.
BAUD Err	UART communication rate error detected, has been restored to factory defaults
KEY1	The button reset button has been detected and has been restored to the factory default value.
SNReadErr	Module serial number read error
Unregistered	Module not registered, cannot be used
FlashWriteErr	Internal EEPROM memory write error
FlashReadErr	Internal EEPROM memory read error
AMPErr	Dynamic magnification setting error

7.4 Register Parameter Summary Table (by Bit)

Register address		name		Functional description (bitwise)																
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
0	00	ADDR	Module address (UART)	Reserved								Module address 1~127, 129~254								
			Defaults	0								1								0x0001
1	01	BAUD①	Communication rate (UART)			UART communication rate, unit: hundred bps														Bit15: Whether the software handshake
			Defaults	0	0	96, bit14: Whether to ignore the "measurement is busy" flag and immediately respond to the command														0x0060
2	02	AUX	Auxiliary function						Reserved								1	0	1	3.22
				0	0	0	0	0	0								0	0	0	
3	03	SYS_FUN	System function register	Reserved								System function code								
			Default value (power-on reset)	0								0								0x0000
4	04		Reserved																	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
5	05	WKMOD	Operating mode			Reserved, see “3.9 Measurement Mode” for details.														Bit14:1: Reserved
			Defaults	0	0	0										0			1	0x0001
6	06	MM_INTE	Continuous	Continuous measurement interval, unit: mS																

			measurement interval																	
			Defaults	500																0x1F4
7	07	ATSD_SEL	Automatic upload																	3.4.3 Actively uploading
			Default value (power-on reset)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000
8	08	RD_INTE	Delayed sampling			Res erved		Delay value. Bit15: dynamic delay bit14: delay unit, default mS												3.13.1 Delayed sampling
			Defaults	0	0	0		200												0x00C8
9	09	RD_COUNT	Expected number of samples	Sampling timeout value in 100mS							Expected number of samples, unit:									
			Defaults	10							200									0x14C8
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
10	0A	EX_METH	Excitation method	Reserved. Bit6: 5 first excitation method selection												Excitation method				Bit4: Forced excitation
			Defaults	0									0		0	1				0x0001
11	0B	HP_FRQ		Internal use, do not modify the value of this register																
			Defaults	33333																0x8235
12	0C	HP_PDF		Internal use, do not modify the value of this register																
			Defaults	5																0x0005
13	0D	HP_DUR	Pumping time		Reserved			Pumping time mS												Bit15: Intelligent pumping time
			Defaults	0	0			1000												0x03E8
14	0E	HP_EXP	Expected voltage		Reserved						Expected high voltage, unit V									Bit15: Enable voltage regulation

			Defaults	1	0							130								0x8082
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
15	0F	FS_FMIN	Sweeping lower limit	Reserved			Sweep lower limit, unit Hz													
			Defaults	0			1500													0x05DC
16	10	FS_FMAX	Sweeping upper limit	Reserved			Sweep upper limit, unit Hz													
			Defaults	0			1600													0x0640
17	11	FS_STEP	Sweep step	Reserved								Sweep stepping, unit Hz								
			Defaults	0								5								0x0005
18	12	FS_SCNT	Sweep single step cycle	Reserved								Single frequency output period, unit:								
			Defaults	0								100								0x0064
19	13	FIT_TYPE	Filtering method	Reserved												Filtering method				
			Defaults	0												0				0x0000
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
20	14	FIT_COUNT	Number of filtered samples	Reserved								Filter sample size 3~30								
			Defaults	0								10								0x000A
21	15	CAL_PAR1	rejection parameter	Gross error elimination method				Gross error rejection factor												
			Defaults	0				20												0x0014
22	16	CAL_PAR2	Quality sample limit	Quality sample limit factor																

			Defaults	4																0x0004
23	17	AMP①	Signal amplification		Reserved										Magnification level 0~31					Bit15: Enable dynamic zoom
			Defaults	0	0										15					0x000F
24	18	FSG_TH	Feedback sweep upper and lower limits	Feedback sweep frequency upper limit difference								Feedback sweep frequency lower limit difference								
			Defaults	20								20								0x1414
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
25	19	DAO_TH	Analog output	Upper frequency value								Lower frequency value								3.22
			Defaults	33								0								0x2100
26	1A	TEMP_PAR1①	Temperature calculation parameter 1	Reserved			Thermistor B value													
			Defaults	0			3950													0x0F6E
27	1B	TEMP_PAR2①	Temperature calculation parameter 2	Thermistor resistance value correction factor, unit 0.01																
			Defaults	100																0x0064
28	1C	TEMP_EX①	Temperature sensor setting	Thermistor nominal value in KΩ									Temperature sensor type							3.17 temperature sensor reading
			Defaults	2								0	1							0x0201
29	1D	EXS_TH	Signal quality limit	Reserved				Limit value category				Limit value								3.12.4 Frequency feedback sweep method
			Defaults	0				0				80								0x0050

DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
30	1E	SIG_TH	Upper and lower limits of signal amplitude	Signal amplitude upper limit								Signal amplitude lower limit								
			Defaults	100								0								0x6400
32	20	SYS_STA	Operating status		Reserved															3.8.2 Operating status
			Read only	0	0								0	0	0	0	0	0	0	0x0000
33	21	SFV	Real-time sweep frequency	Reserved			Real-time sweep frequency, unit: Hz													
			Read only	0			0													0x0000
34	22	SMP_QUA	Sampling quality	Reserved								Quality sampling evaluation quality value								
			Read only	0								0								0x0000
DEC	**HEX**	**symbol**	**name**	**15**	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**	**instruction manual**
35	23	S_FRQ	Frequency value	Vibrating wire sensor real-time frequency value, unit 0.1Hz																
			Read only	0																0x0000
36	24	F_REQMH	Frequency module value Read only	WKMOD.[3:1]=0: Real-time Frequency module value of vibrating wire sensor=F_REQMH*65536+F_REQML WKMOD.[3:1]=1: Real-time high-resolution frequency value=F_REQMH*65536+F_REQML																
																				0x0000
37	25	F_REQML																		
																				0x0000
38	26	V_POW/ADC1	General AD conversion 1	ADC1/VIN real-time acquisition value																
			Read only	0																0x0000

39	27	S_RES	Coil resistance	Vibrating wire sensor coil resistance value, unit: ohm																
			Read only	0																0x0000
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
40	28	V_SEN	Excitation voltage		Sensor real-time excitation voltage value, unit: 0.01V															
			Read only	0	0															0x0000
41	29	TEMP	temperature	Real-time temperature value, unit: 0.1 °C																
			Read only	0																0x0000
42	2A	SMP_STD	Sample standard deviation	All sampling standard deviation, unit: Hz								High quality sampling standard deviation, unit: Hz								
			Read only	0								0								0x0000
43	2B	HQ_COUNT	Sample quality	Reserved							Quality sample size value									
			Read only	0							0									0x0000
44	2C	SIG_VAL1	Signal amplitude 1	Excitation complete signal amplitude								Start sampling signal amplitude								
			Read only	0								0								0x0000
DEC	HEX	symbol	name	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	instruction manual
45	2D	SIG_VAL2	Signal amplitude 2	Sampling end signal amplitude								Average signal amplitude								
			Read only	0								0								0x0000
46	2E	GPIO	General IO settings	Input/output direction setting								Pin level status read/modify								
			Read and write	0								0								0x0000
47	2F	ADC02	General AD conversion 2	Reserved				12-bit AD value												
			Read only	0				0												0x0000

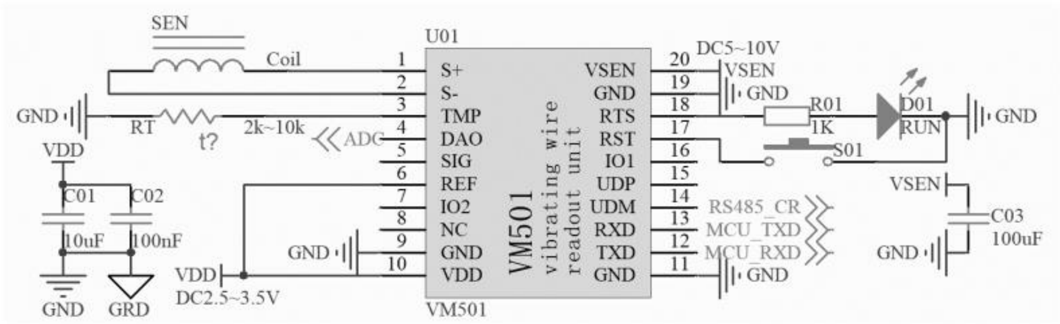
48	30	ADC03	General AD conversion 3	Reserved	12-bit AD value	
			Read only	0	0	0x0000
49	31	ADC04	General AD conversion 4	Reserved	12-bit AD value	
			Read only	0	0	0x0000
①: These parameters take effect the next time the module starts						

7.5 Different Firmware Version Register Difference Table (V2.51 and before)

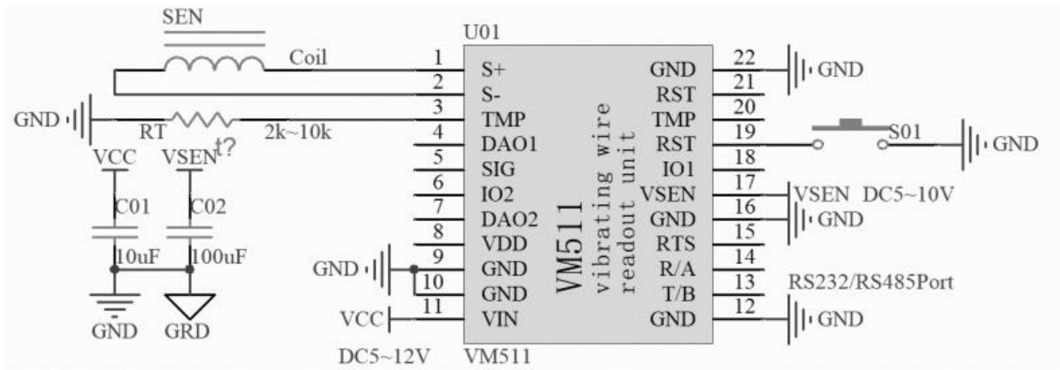
Register symbol	address	Changes
ADDR	0x00(0)	No change
BAUD	0x01(1)	No software handshake setting bit
SYS_FUN	0x03(3)	No change
WKMOD	0x05(5)	No "Turn off digital interface when busy" bit
MM_INTE	0x06(6)	No change
ATSD_SEL	0x07(7)	No "signal amplitude data item"
RD_INTE	0x08(8)	No "Dynamic Delay" and "Delay Unit" setting bits
RD_COUNT	0x09(9)	No sampling timeout setting
EX_METH	0x0A(10)	No feedback excitation and segmentation excitation method
HP_FRQ	0x0B(11)	No change
HP_PDF	0x0C(12)	No change
HP_DUR	0x0D(13)	No "smart stop pumping" bit
HP_EXP	0x0E(14)	This register has not taken effect
FS_FMIN	0x0F(15)	No change
FS_FMAX	0x10(16)	No change
FS_STEP	0x11(17)	No change
FS_SCNT	0x12(18)	No change
FIT_TYPE	0x13(19)	No change
FIT_COUNT	0x14(20)	No change
CAL_PAR1	0x15(21)	No change
CAL_PAR2	0x16(22)	No change
AMP	0x17(23)	No such register
TEMP_PAR1	0x1A(26)	Temperature plus constant
TEMP_PAR2	0x1B(27)	Temperature multiplication constant
TEMP_EX	0x1C(28)	Thermistor definition is different
EXS_TH	0x1D(29)	Extended interface control registers (screens and relays)
SIG_TH	0x1E(30)	No such register

7.6 Application Circuit

VM501



VM511



8, Revised History

8.1 Firmware Revision History

Firmware version V3.20

Modify the firmware version number v3.20_181202.

Factory parameter modification function is disclosed, and three string instructions $RSTP, $STFC and $STDF are added accordingly.

　　$RSTP: set the parameter to the factory value

　　$STFC: writes the current parameter to the factory parameter area

　　$STDF: returns the parameters to a set of fixed defaults set by the manufacturer

Added support for MODBUS command code 04.

Half-power consumption function is added, corresponding to AUX of auxiliary function register.[3].

Other optimization

　　UART instruction response speed optimization.

Firmware version V3.14

Modify the firmware version number v3.14_180917.
Auxiliary function register adds AUX.[15]UART data bit, AUX.[14:13]UART stop bit and AUX.[12:11]UART check bit parameter bit definition.
The analog output algorithm is modified from the original VDD scale output to the absolute value physical quantity output based on voltage reference.
Add single measurement instruction code 0x7X.
Add direct read frequency register to activate single measurement function.

Firmware version V3.14

Modify the firmware version number v3.14_180901.
Auxiliary function register adds AUX.[1] ripple filter and AUX.[2] vibration avoidance parameter bit definition.

Firmware version V3.14

Modify the firmware version number v3.14_180801.
The frequency value analog output function is increased, and the analog frequency upper and lower register DAO_TH and auxiliary function register AUX are correspondingly increased.

Firmware version V3.13

Modify the firmware version number v3.13_180701.
Fixed a BUG that automatically switches to continuous measurement mode after receiving a single measurement instruction when the sensor is not detected
Correction of overflow flag dislocations when frequency exceeds 6553.6hz.
Correct the inconsistency between the real time frequency register and the high precision frequency register.
Optimize low voltage sweep output frequency algorithm.
The conflict between the first excitation method and the output period of the fixed sweep frequency is corrected.

Firmware version V3.12

Modify the firmware version number to V3.12_180621.
New parameter UART.[14] to ignore the "Measure Busy" flag and respond to the command immediately.
The AABB read command response section adds protection (reduces the probability of a response code error).
Add new function the low-voltage sweep frequency self-calibration, the parameters are fixed, and do not change with the factory parameters.
Add new register "Feedback gradual frequency sweeping method (FFG) upper and lower limits" register FSG_TH (0x18).
Improve the stability and correctness of the excitation voltage display value.

Add real-time data stability control algorithm.
Add frequency mode register display content setting parameter WKMOD.[3:1].
Add a new excitation method "full-band sweeping method"
The excitation method register EX_METH.[6:5] is defined as the "first excitation method" setting bit.
Correct the bug that automatically switches to continuous measurement mode after receiving a single measurement command when no sensor is detected.
Correct the error that the overflow flag is not set when the frequency exceeds 6553.6Hz.
Correct the error that the real-time frequency register is inconsistent with the high-resolution frequency register value.
Optimize the low-voltage swept output frequency algorithm.
Corrected the conflict between the first excitation method and the number of fixed sweep frequency output cycles in the feedback fixed sweep method.

Firmware version V3.11

Modify the firmware version number to V3.11_180318.
The signal amplitude limit function is in effect (previously this register did not work).
Fixed a BUG that could not be used when the UART communication rate was 19200bps.

Firmware version V3.10

On February 1, 2018, the firmware version number was changed to V3.10_180201.
On February 1, 2018, the custom segmentation domain sweep method was added (method 8).
On February 1, 2018, The excitation method register EX_METH.[5] is defined as the "first excitation method" setting bit.
On March 14, 2018, the firmware version number was changed to V3.10_180314.
On March 14, 2018, Fixed BUG where sending other instructions to the module while correcting real-time amplitude output might cause the module to crash.
On March 14, 2018, the bug of the feedback excitation method during the mandatory excitation was revised.
On March 14, 2018, the BUG of the custom segment domain swept excitation method with incorrect output frequency was corrected.

Firmware version V3.02

On January 10, 2018, the firmware version number was changed to V3.02_180110.
On January 10, 2018, the BUG that the power-on initialization cannot be correctly identified when the sensor coil resistance is large (close to or greater than 1k) is repaired.
On January 10, 2018, the factory default parameters were modified. Enable the Expect voltage control function (HP_EXP.[15]=1) and the Expect voltage is 130V (HP_EXP.[7:0]=130).
On January 10, 2018, the run status register SYS_STA.[6] is defined as the excitation duration timeout flag.
On January 10, 2018, the operating status register SYS_STA.[15] is defined as the coil

not connected status flag.
On January 10, 2018, when no excitation was applied (no coils were detected and forced excitation was not set), some status registers were set to unambiguous values, preventing false positives when reading these registers.

8.2 Document Revision History

Manual version	Firmware version	Date Time	Modify content
V1.00	V3.01	June 2015	First created
V1.01	V3.01	July 2017	Correct temperature sensor register description
			Add pump voltage time dynamic adjustment register description
V1.02	V3.01	August 2017	Add application circuit
V1.03	V3.01	October 2017	Add VM4XX series
V1.04	V3.02	January 2018	
Modify the default value of the register HP_EXP in the "Register Overview" section, the "Register Parameter Summary Table", and the "High Voltage Pulse Excitation Method" section. Modify the definition of SYS_STA.[6].[15] in the "Register Parameter Summary Table" and "Operation Status" subsections.			
V1.10	V3.10	March 2018	
The sensor excitation method section adds a description of the "segmented progressive low-voltage sweep method - self-defined frequency band" method description and function description. Add EX_METH.[6:5] to the "First Excitation Method" setting bit description. A description of the "first excitation method" has been added to the feedback excitation method section. Add "VM4XX internal op amp connection block diagram" Fix VM301 size chart			
V1.11	V3.12	June 2018	
Add UART.[14] parameter description Add the sweep frequency self-calibration chapter Add Feedback gradual frequency sweeping method (FFG) upper and lower limit register description Add the working mode register WKMD.[3:1] parameter description Improve the description of the Frequency module register Modify the input voltage, excitation voltage recommended range Correct the wiring error description of the external temperature sensor DS18B20			

Add a new excitation method "full-band sweeping method" **Add the frequency of the sweep output frequency self-correction operation method**			
V1.12	V3.14	August 2018	
Add VM501/511 related content. **Add the frequency value analog output section 3.22.1.** **Add the signal ripple filter section 3.22.2.** **Add auxiliary register AUX1 specification.** **Modify the expression of RTS pin output signal, change the original high and low level to logic plus or minus.**			
V1.12	V3.14	September 2018	
Modify section 3.22 and add data bit, stop bit and check bit parameter description of UART communication interface. **Modify analog output text to indicate absolute value (voltage or current).** **Add single measurement instruction code 0x7X description.** **Add "direct read frequency register" to activate single measurement function.**			
V1.20	V3.20	December 2018	
Modify the communication protocol description, add MODBUS04 function code support **Add factory parameters and default parameter setting instructions** **Add VM501 temperature range table**			