Technical specification of KS136/KS136A

Ver. 1.09

Innovative technology originates from Dauxi Technologies

Shenzhen dauxi electrical technology Co.Ltd

Modify Date	Content	Edit	Revision	Note
Nov. 4, 2016	Initial release.	*X.Q.*	1.00	
Jan. 16, 2017	Instruction detail description of command “0x01~0x06”	*X.Q.*	1.01	
Sep. 11,2017	*Command “0x99” added.*	O.Y.Y	1.02	
Jun. 18,2019	*Command “0x99” added.*	O.Y.Y	1.04	beam angle of 49k add
Jul. 13,2019		O.Y.Y	1.05	beam angle of 49k add
Jul. 24,2019		O.Y.Y	1.06	Detection range modification
Nov.26,2019	*Third generation sensor（KS49-19TRWP）with silicone case on page 2.*	S.M.	1.07	Page 2
April 22, 2020	*Added GXX firmware description on page 32，33*	X.T.C	1.08	Page *31，32*
May 14, 2020	*Add return information of command “0x99”*	X.T.C	1.09	Page *29*

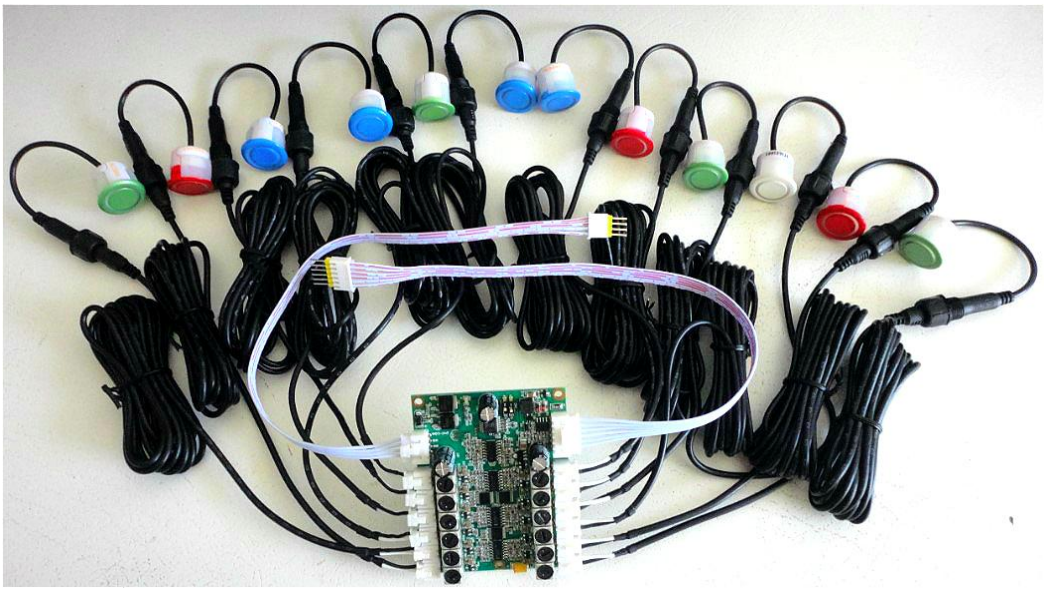
:

:

:

:

:



Summary of KS136's functions:

● **Transceiver integrated** design, 1 main control board is connected to 12 transceiver integrated waterproof probes, each probe works independently;

● The color of the waterproof probe is **black**, white, and silver for common use, **red**, **blue**, **green**, and **yellow** are optional, and other colors can be customized;

● Detection range - single probe: **18cm-450cm**, detection range - double probe: **2cm-450cm**. People can measure up to **110~150cm**;

● The detection frequency can reach 50Hz, that is, it can detect 50 times per second;

● Support ***I²C/TTL*** serial interface, compatible with KS103 protocol; support 485 interface;

● A total of 20 modifiable ***I²C/TTL/485*** addresses, ranging from 0xd0 to 0xfe (except 0xf0, 0xf2, 0xf4, 0xf6, 8-bit addresses);

● If no I²C control command is received within 5s, it will automatically enter uA-level sleep, and can be woken up by the host I²C control command at any time;

● Use industrial grade configuration, working temperature (-30℃~+85℃);

● Wide working voltage range 3.0V~5.5V or 12~24V, 12V power supply is recommended;

● The communication rate of I²C mode is 50~100kbit/s; the default communication rate of TTL/485 serial port is 9600bps; the user can modify it to 115200bps, etc.;

● Using unique **adjustable** filter noise reduction technology, it can still work normally when the power supply voltage is disturbed or the noise is large;

● Autologous acoustic wave recognition technology, which can block 70% of noise, or return 0xeeee error code;

● Environmentally friendly lead-free, ROHS certificate, CE certificate (same as KS136).

Note: The third-generation probe used by KS106A/KS136A on page 3 of this manual (No. KS49-19TRWP, the third-generation probe will be discontinued soon, it is not recommended to select), when using the dual-probe detection mode (KS136A corresponds to the command "0x01, 0x02, 0x03, 0x04, 0x05, 0x06"; KS106A corresponds to the command "0x02, 0x06"), we recommend adding a silicone cover to the third-generation probe to ensure reliability after long-term work such as freezing, dust solidification, etc. . The installation hole should be 24.0~24.5mm in diameter, and the center distance should be 25~30mm. The effect after double probe installation is shown in Figure A below:

Fig.A

Fig.B

Fig.C

If the installation plane is too thin, the third-generation single probe (KS49-19TRWP) is also recommended to put a silicone sleeve (as shown in Figure B) to ensure that ice and dust solidification may cause false triggering after long-term work.

Figure C is a real picture of two black and white silicone sleeves. When the black probe is required, the black silicone sleeve is used, and when the probe is white or non-black, the white silicone sleeve is used. It is advisable to leave 24.0~24.5mm for the installation hole after the silicone sleeve is covered, and the standard is that the structure design is convenient for assembly and will not fall off.

The first-generation, second-generation, and fourth-generation probes do not need this silicone sleeve.

If the third-generation probe is not equipped with this silicone sleeve, when double-probe detection, the probe needs to be fixed on the silica gel with a hardness of 30~40 Shore to avoid false triggering caused by resonance caused by direct rigid connection between the transmitting probe and the receiving probe.

The black main control box and the probe are connected by a waterproof cable. The wire length has the following standard wire lengths:
30cm, 0.85m, 1m, 1.5m, 2.3m (factory default), 4m, 5m, 7m, 10m, 14m, 15m, 15.5m, 16m, 18m, 21m, 24m , 28 meters, these lengths are in stock (more than 2.3 meters, the price has increased)
The line number is: LINE-XH2.54-WM-line length, the bold line number is: LINE24-XH2.54-WM-line length
For example, if the wire length is 30cm, the wire number is: LINE-XH2.54-WM-0.3M
For example, the wire length is 28 meters, the wire number is: LINE-XH2.54-WM-28M
The length of the probe tail line is divided into 10cm and 20cm, and the number is: LINE-MM-TAN-0.14M or LINE-MM-TAN-0.24M.
The probes and parameters used by KS106/KS136/KS106A/KS136A are as follows in Table 1:

model	Model of the probe used	Probe Center Frequency	Recommended mounting hole size	beam angle	MTBF (1 trigger every 80ms)	Probe cost/yuan	Probe Appearance	Probe logo
KS106 KS136	KS40-19TRWP-L10C040	40K	19±0.1mm thickness 1.5~4mm round hole front and back rounding R0.2	70±15°	The defect rate is 1 in 1,000 within 5,000 hours	low		DAUXI.COM (the first generation)
	KS40B-19TRWP-L10C108		19±0.1mm thickness 1.5~4mm round hole front and back rounding R0.2	Vertical arrow direction 55±10° horizontal 100±15°	The non-performing rate within 15,000 hours is 3/10,000	low		DAUXI.COM (the second generation)

KS106A KS136A	KS49-19TRWP-L10C100 (will be discontinued soon, it is not recommended to choose)	49K	19.7±0.1mm thickness 1.5~4mm round hole front and back rounding R0.4. When the third-generation probe is installed with double probes, pay attention to not recommend hard connection between probes, be careful not to squeeze the probe, it is recommended to install the double probe in Shao A35~A40 soft silicone substrate. Or add a silicone sleeve	60±10°	The defect rate within 50,000 hours is less than 1 in 10,000	middle		DAUXI.COM (the third generation will be discontinued soon, it is not recommended to choose)
	KS49B-19TRWP-L10C702		**19±0.1mm thickness 1.5~4mm round hole front and back rounding R0.2**	**Vertical arrow direction 45±10° horizontal 100±15°**	**The defect rate within 50,000 hours is less than 1 in 10,000**	**middle**		**DAUXI.COM (the fourth generation)**

The probe selected for KS106/KS136 is KS40-19TRWP-L10C040 in Table 1 above, and the specifications are as follows in Table 2:

Part Number	KS40-19TRWP
Center frequency kHz	40.0±2.0
Output sound pressure dB@30cm/10V sine wave	≥100
Receive sensitivity dB / v / μbar	≥-73
-6dB Directivity deg.	70±15°
Ringing Time ms	≤1
Capacitance pf @ 120Hz 25℃	2000±20%
Input voltage Vp-p	140
Operating Temperature ℃	-35~+80
Storage Temperature ℃	-40~+85
Materials of the case	Aluminum
Type of the terminal	XHB-2.5AW Connector
Length of the lead wire mm	100/200
Diameter of the lead wire mm	3.05±0.2
Color	RED/YELLOW/BLACK/GREY/GOLDEN
Material of plastic	PC+ABS+ Silica gel 40°

Table 2

The probe selected for KS106/KS136 is KS40-19TRWP-L10C040 in Table 1 above, and the specifications are as follows in Table 3:

Part Number	KS49-19TRWP
Center frequency kHz	49.0±1.5 at 25° ±5℃
Ringing Time ms	1.7ms±0.5 at 25° ±5℃ ≤2.5ms at -40~ +85℃
Receive sensitivity	4V±1 of 1.5m at 25° ±5℃
Beam angle deg.	60±10°
Ringing Time ms	≤1
Capacitance pf @ 120Hz 25℃	2000±20%
Input voltage Vp-p	150
Operating Temperature ℃	-40~ +85℃
Storage Temperature ℃	-40~ +85℃
Materials of the case	Aluminum
Type of the terminal	XHB-2.5AW Connector
Length of the lead wire mm	100/200
Diameter of the lead wire mm	3.05±0.2
Color	RED/YELLOW/BLACK/GREY/GOLDEN
Material of plastic	PC+ABS+ Silica gel 35°

Table 3

The coding rules for the probe model in the second column in Table 1 are as follows in Table 4:

KS49-19TRWP - L10C702	
Among them, KS is the model head, 49 is the frequency of 49k (the other frequency is 40K); 49 with "B" behind it is a double angle, without "B" is a single angle, 19 is the minimum diameter of the probe 19mm, TR is a transceiver integrated, WP is waterproof, L10 means the probe line length is 10cm (another length of 20cm needs to be customized), 702 means the color is bright black (101 means metallic silver; 100 means pearl white; 040 means natural yellow). The default black and white are two colors, which are also the most commonly used colors.	

Table 4

KS136/KS136A electrical performance parameters:

I²C /TTL operating voltage: ***3.0V~5.5V***; or ***12V~24V*** DC power supply (***12V*** recommended).

Use 485 interface working voltage: ***4.5V~5.5V***; or ***12V~24V*** DC power supply (recommended ***12V***).

Please pay attention to ventilation and heat dissipation when using ***24V*** voltage.

Note: If ***3.0V~5.5V*** is connected, it is recommended to disconnect the ***12V~24V*** power supply; if ***12V~24V*** is connected, it is recommended to disconnect the ***3.0V~5.5V*** power supply.

Instantaneous maximum current when working: ***200mA@5.0V, typical***. ***45mA@12.0V, typical***.

Operating current: ***100mA@5.0V, typical. 22mA@12.0V, typical.***

Maximum power consumption during sleep: ***500uA@5.0V, typical*** (do not sleep in serial port mode)

Power consumption: Use nanowatt technology to save power, automatically enter uA-level sleep without receiving I²C control command within 5s, and can be awakened by host I²C control command at any time.

Note: Please ensure that the power supply is directly powered by the battery or output from the linear voltage regulator chip; or it is recommended that the power supply ripple VPP < 120mV.

KS136/KS136A probe quantity selection:

KS136/KS136A have 12 probes by default, CON1, CON2, ..., CON12 on the back of the main control board correspond to probe 1, probe 2, ..., probe 12 respectively.

KS136/KS136A-8 means that the probe is only connected to the CON1~CON8 position, other positions are invalid;

KS136/KS136A-10 means that the probe is only connected to the position of CON1~CON10, other positions are invalid;

KS136/KS136A-12 is the default configuration of 12 probes, all positions are valid.

KS136/KS136A single-angle probe and double-angle probe selection:

Single-angle probes (both horizontal and vertical are 60-70°, see Table 1 for details) and dual-angle probes (horizontal 100-115° and vertical 45-55°, see Table 1 for details). For self-driving cars or large cleaning robots, large AGV robots, crawler vehicles, lifting machinery, etc. with the same size as the car, it is recommended to use dual-angle probes, because the beam angle in the horizontal direction of the dual-angle probes reaches a maximum of 115°, which can effectively reduce Small undetectable area. When the left and right judgment is involved in the robot or AGV, it is recommended to use a single-angle probe to facilitate the robot to judge the left and right. In addition, when using dual-probe combined probing to achieve zero-blind zone probing, it is recommended to select a single-angle probe with no special requirements for installation. In order to prevent defects caused by installation errors in large quantities.

Note: Based on production capacity needs, the default shipments are robot perception priority firmware, not precision priority firmware. If you need precision priority firmware, you need to order it. And need to provide the application environment, probe installation and so on.

Wiring and Mode Switching Instructions

As shown in Figure 1 below, the connection pins on KS136/KS136A are marked with: VCC (3-5.5V), SDA/TX (SDA for short), SCL/RX (SCL for short), GND, 485B, 485A;
CON1 ;CON2;CON3;CON4;CON5;CON6;CON7;CON8;CON9;CON10;CON11;CON12;VIN(12-24 V), GND, GND, VIN(12-24V).

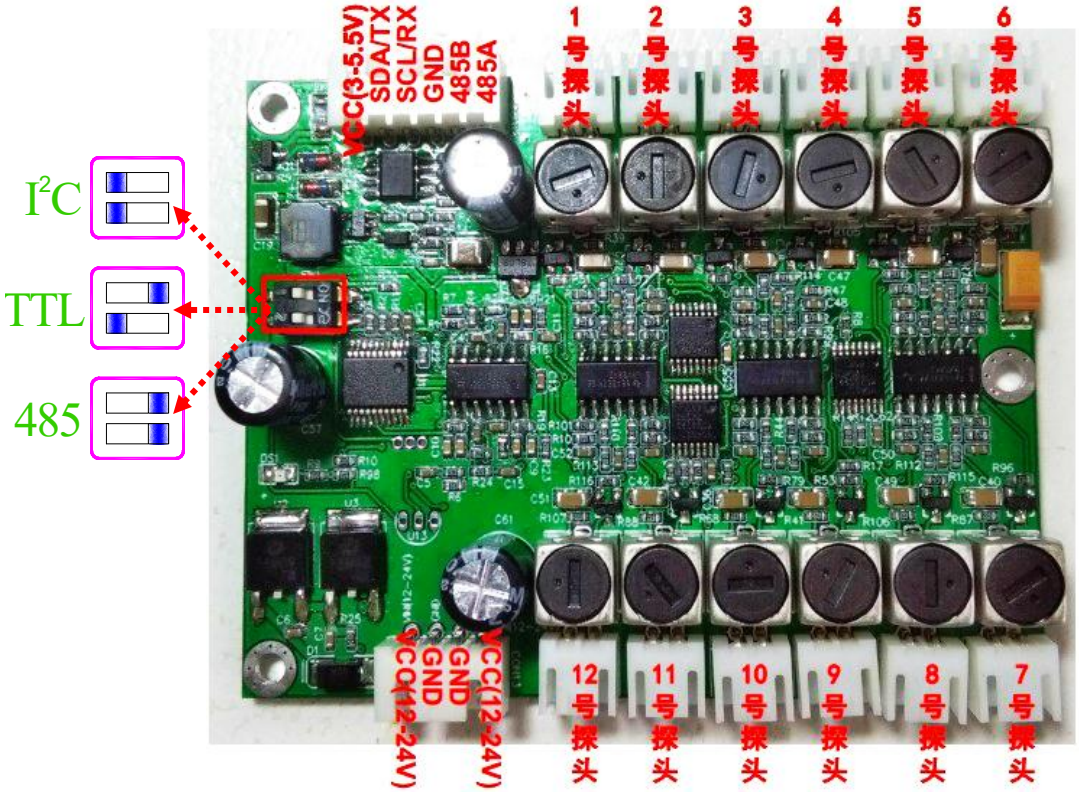


Fig.1

As shown in Figure 1, the whole left of the DIP switch SW1 is I2C mode; the upper right, lower

left and the left of the DIP switch SW1 are TTL serial port mode; the DIP switch SW1 is fully right in the 485 mode.

The connection method of the I²C mode signal line is: SCL/RX is connected to the SCL of the host computer; SDA/TX is connected to the SDA of the host computer.

The connection method of the TTL serial port mode signal line is: SDA/TX is connected to the RXD of the host computer; SCL/RX is connected to the TXD(1) of the host computer.

Note (1): The TTL serial port here is not a 232 serial port. The TTL level can be directly connected to the TXD/RXD of the microcontroller, but it cannot be directly connected to the 232 serial port (direct connection will burn the module). A MAX232 level conversion is required. Convert the TTL level to 232 level.

In 485 serial port mode, the signal line connection method is: 485A to 485A; 485B to 485B (2).

Note (2): The screen printing of KS136/KS136A-V101 adopts the cross screen printing method, that is, 485B is printed at the position of 485A, and 485A is printed at the position of 485B. Pay attention when wiring.

The power connection methods of the above three modes are:

Method 1: The VCC (3-5.5V) of the six-pin socket is connected to the positive pole of the power supply with a voltage range of 3-5.5V, and the adjacent GND is connected to the negative pole;

Method 2: The VCC (12-24V) of the four-pin socket is connected to the positive pole of the power supply with a voltage range of 12-24V, and the adjacent GND is connected to the negative pole.

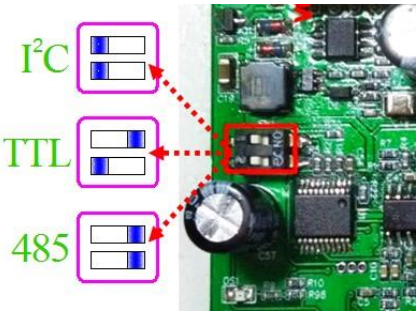
It is recommended to use 3-5.5V power supply (VPP<120mV) for power supply. If only 12-24V power supply is available, 12V power supply is recommended. If using a 24V power supply, please pay attention to ventilation and ensure good heat dissipation. In order to pursue high power quality and high anti-interference, KS136/KS136A does not use switch chip PWM to step down, but uses linear voltage regulator to obtain a better anti-interference experience.

Connect the 3-5.5V power supply, keep the 12-24V power supply floating; connect the 12-24V power supply, keep the 3-5.5V power supply floating.

When using the 485 interface, it is recommended that the power supply voltage should not be lower than 4.5V.

The following describes the I²C mode, TTL serial port mode and 485 serial port mode in detail.

I²C mode



KS136/KS136A connect：

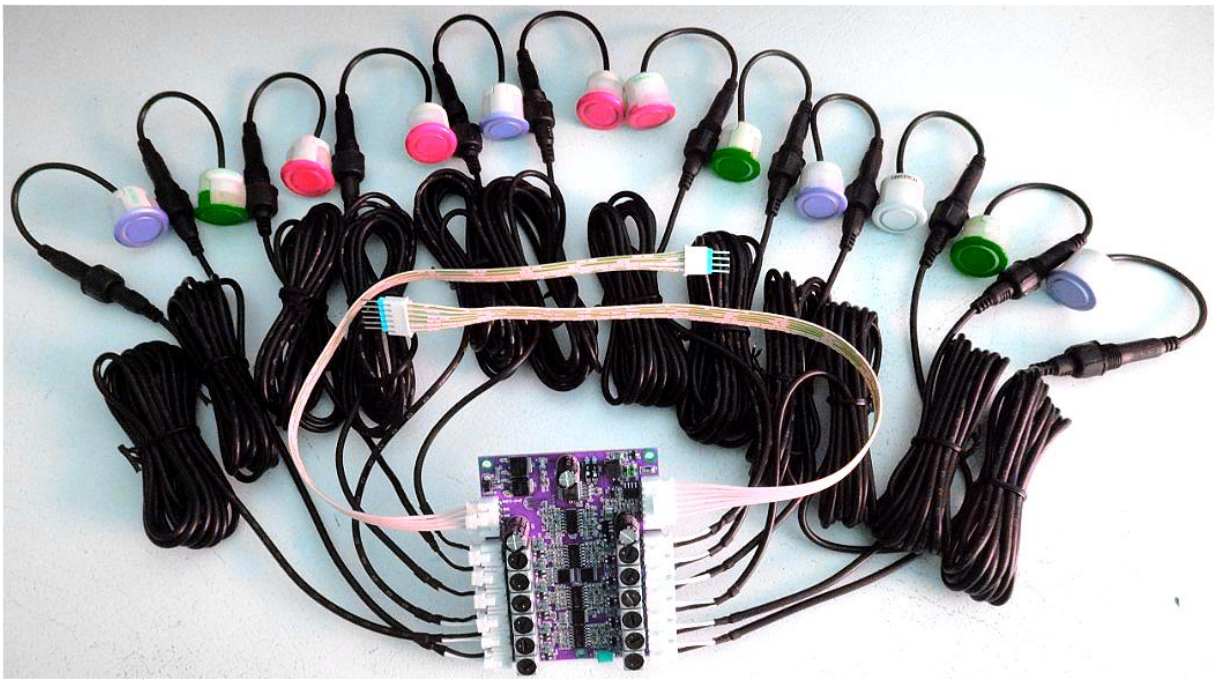


Fig.2

As shown in Figure 2, KS136/KS136A includes a main control board, which is connected to the main control board through a self-locking 2PIN socket (probe 1, probe 2, ..., probe 12 corresponds to CON1, CON2, ..., CON12) 12 2.5m cables (other lengths can be customized), and 12 waterproof probes connected with the 2.5m cables with waterproof threads. The self-locking 6PIN socket on KS136/KS136A is an external data and power interface. The pins are marked as follows: VCC (3-5.5V), SDA/TX (SDA for short), SCL/RX (SCL for short), GND, 485B, 485A; the other extended 12-24V power interface pins are: VIN(12-24V), GND, GND, VIN(12-24V).

VCC in the 6-pin socket is used to connect +5V (3.0~5.5V range) power supply (3), SDA/TX is the data line of I^2C communication, SCL/RX pin is the clock line of I^2C communication, GND is used to connect power ground. Both SCL and SDA lines need to be connected to VCC by a 4.7K (resistance value of 1~10K) resistor from the host. The I^2C communication rate of KS136/KS136A is recommended not to be higher than 100kbit/s.

Note (3): It is recommended to use 4.5~5.5V power supply in 485 communication mode. Anti-reverse polarity protection is added to the voltage input. If the indicator light is not on, please check whether the power cord is reversed.

The two VINs (12-24V) in the 4-pin socket are actually shorted internally, and the two GNDs are also shorted internally. Therefore, only one of the pins can be connected during testing. Full connection is recommended to reduce contact resistance during mass production. VIN (12-24V) voltage is recommended to use 12V. If using a 24V power supply, please pay attention to ventilation and ensure good heat dissipation. In order to pursue high power quality and high anti-interference, KS136/KS136A do not use switch chip PWM step-down, but use linear regulator to obtain better anti-interference effect.

The connection between the probe and the line is shown in Figure 3 below, using a waterproof thread connection. After plugging in, please pay attention to tighten it to meet the needs of waterproofing.

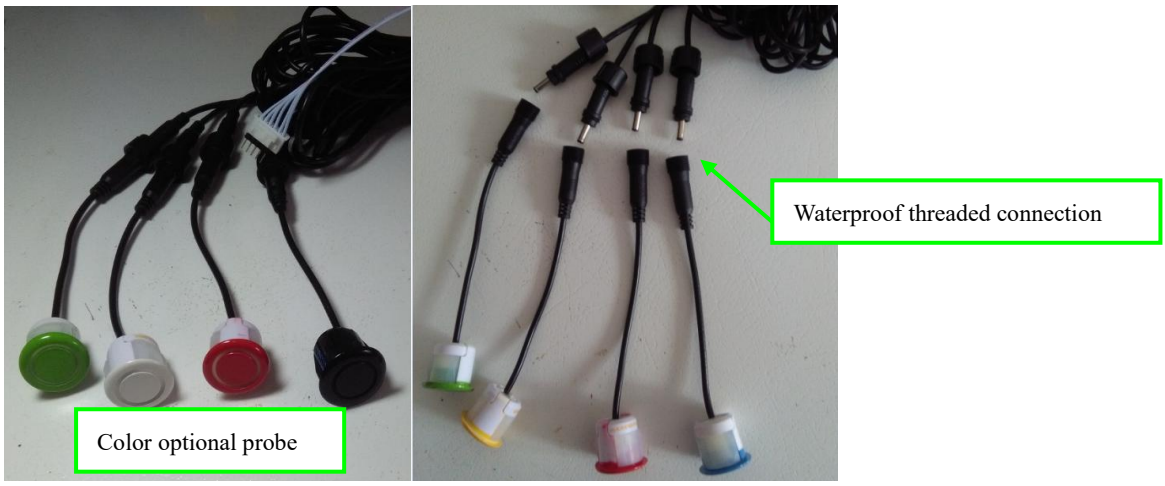


Fig.3

Specific connection of I²C mode is shown in figure 4 below（up to 20）:

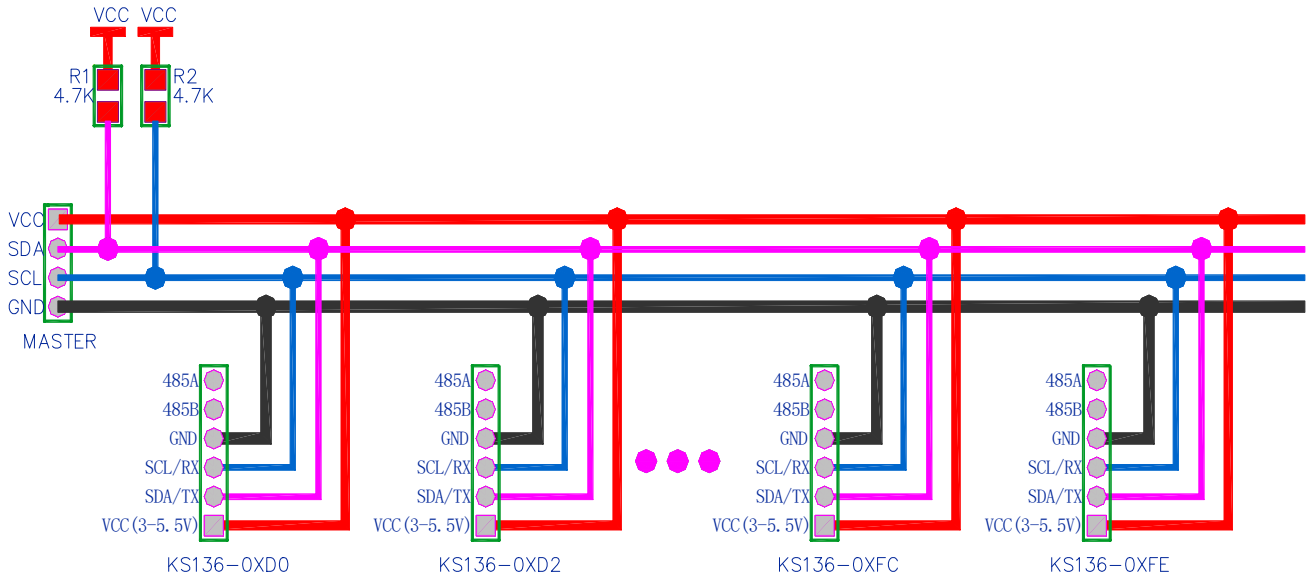


Fig.4

The default address of KS136/KS136A is 0xe8, the user can modify the address to any one of 20 addresses: 0xd0, 0xd2, 0xd4, 0xd6, 0xd8, 0xda, 0xdc, 0xde, 0xe0, 0xe2, 0xe4, 0xe6, 0xe8, 0xea, 0xec, 0xee, 0xf8, 0xfa, 0xfc, 0xfe. (4)

Note (4): Please note that the above addresses do not include 0xf0, 0xf2, 0xf4, 0xf6, these 4 addresses are reserved for the 10-bit address of the I²C slave. The host device that controls this module may only support the 7-bit I²C slave address. In this case, the 8-bit address needs to be shifted to the right by 1 bit to use as the address. For example, the default address of this module is 0xe8, which corresponds to the 7-bit address of 0x74.

Modify the timing of the I²C address:

Address	2	**0x9a**	Delay 1ms	**Address**	2	**0x92**	Delay 1ms	**Address**	2	**0x9e**	Delay 1ms	**Address**	2	**New Address**	Delay 100ms

Modifying the I²C address must be done in strict accordance with the sequence, and the delay time in the sequence is the minimum time. For the 51 single-chip host, it can be implemented by calling the change_i2c_address(addr_old, addr_new) function shown in Annex 3.

After the modification, please power on the KS136/KS136A again, and you can observe that the LED displays the new address. In the process of modifying the I^2C address of the KS136/KS136A, it is strictly forbidden to power off the KS136/KS136A suddenly. Please do not put the function of modifying the address in the while(1) loop, and ensure that it only runs once after the program is powered on.

After the I^2C address is set to be different, 20 KS136/KS136A can be connected simultaneously on the two I^2C buses of the host. When the host controls one of the KS136/KS136A modules, the other modules automatically enter the microwatt-level power consumption sleep mode, so there is no need to worry about insufficient current supply.

KS136/KS136A's Workflow:

When the KS136/KS136A is powered on, the system will first start a self-check, which takes about 1200ms. During this self-checking process, KS136/KS136A will check whether all probes are plugged in normally, and check whether each configuration is normal. If there is any abnormality, the fault position of the probe will be automatically reported. Please refer to Table 1 for details of reported error codes. In I^2C mode, the error code can be obtained by reading the corresponding register value.

After the self-test is completed, the LED shown in Figure 5 will flash in binary mode to display its 8-bit I^2C address. Two quick flashes represent "1", and one slow flash represents "0". For example, display 0xea address, its binary number is 0B11101010, green LED **flashes twice quickly → off → flashes twice quickly → off → flashes twice quickly → off → flashes once slowly → off → flashes twice quickly → Off → flashes once slowly → off → flashes twice quickly → off → flashes once slowly → off.** (5)

Note 3: The green light when the LED flashes may irritate the eyes. Please try not to look directly at the working LED at close range. You can use the peripheral vision of your eyes to observe the flashing.

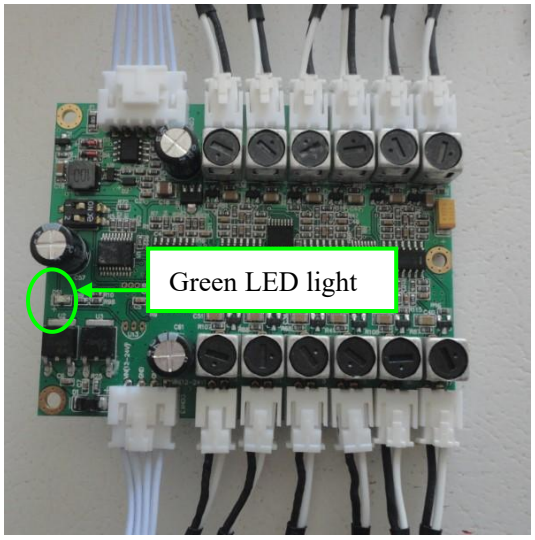
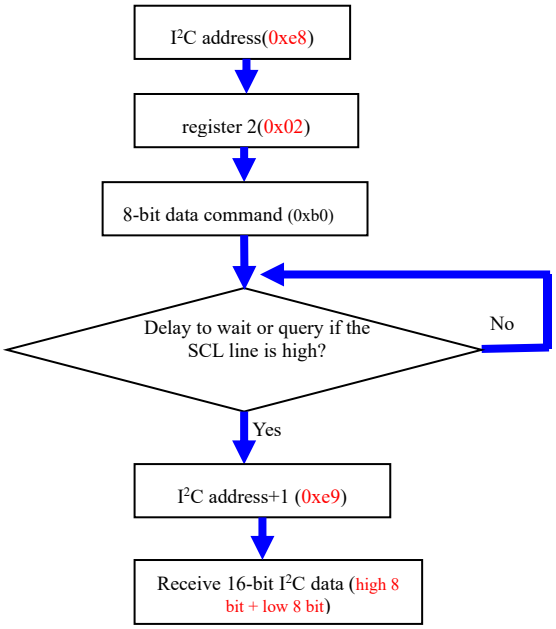


Fig.5

After the KS136/KS136A starts up, if it receives a valid data command from the host, the LED will stop flashing immediately. Enter command detection mode.

KS136/KS136A use the I^2C interface to communicate with the host, and automatically respond to the host's I^2C control commands. The command is 8-bit data, and the command sending process is as follows:



Detection end intelligent recognition

After the KS136/KS136A sends the detection command, it needs to wait for a period of time to obtain the correct 16-bit I²C data. The user only knows the maximum detection time, but does not know the actual detection time each time. KS136/KS136A adopts the detection end intelligent identification technology. During the detection process, SCL will always be kept at a low level. Users can wait by querying whether the SCL line becomes a high level, that is, a while(!SCL) statement. When the SCL line becomes a high level, it indicates that the detection is completed and can start receiving through the I²C bus. to the 16-bit data detected by the KS136/KS136A. Note that after sending the detection command, it is necessary to delay more than 40us to check whether the SCL line becomes high. The 40us is the response delay of KS136/KS136A. Different from KS103, because KS136/KS136A adopts detection protection, it is recommended to delay about 1ms before judging the SCL line for each detection. This will neither interrupt the ongoing detection nor reduce the detection efficiency. It is also possible to start receiving 16-bit I²C data by delaying for a period of time. (6)

Note (6): This bus clamp detection method can obtain greater detection speed and efficiency for customers, instead of waiting at least 65ms for each detection through timer delay or delay function delay. In other words, most of the time, users only need to quickly know whether there are obstacles within 2m. The specific delay time should be greater than the maximum detection time of each instruction listed in Table 1.

If you don't want the SCL line to be pulled low during detection, you can send the command 0xc3 command, and then the SCL line will not be pulled low after the KS136/KS136A is powered off and restarted. If you want to restore the I²C clamping and SCL pull-down functions, send the 0xc2 command.

The configuration method is very simple. Send the command sequence to this module: "I²C address + register 2 + 0xc2/0xc3". Please delay at least 2 seconds after sending, so that the system can automatically complete the configuration. and start working according to the new configuration.

Taking the program shown in Annex 3 as an example, the configuration code is as follows:

write_byte(0xe8,2,0xc2);

delayms(2000);

After the detection is over, the intelligent identification function will be automatically saved after the configuration is completed, and will work according to the new configuration immediately. The

KS136/KS136A will operate in the new configuration after power cycle.

probe command

After the detection command is sent, the KS136/KS136A will enter the corresponding detection mode according to the detection command. The host must wait for a while before starting to query the detection result through the I²C bus. Querying the I²C bus too early will obtain a value of 0xff. Note that the format of each frame of detection command is:

I²C address	Register 2	8-bit data

All the **I²C control** commans and readable register are summarized as follows：

Register	command	Return value range (decimal)	Return value range (Hexadecimal)	Remarks
0		1~254	0x01~0xff	Program version identification and manufacturer identification. Refer to Appendix 3 for example function, return value=read_byte(0xe8,0);
1		1~252	0x01~0xfc	Date of manufacture logo. The upper 8 bits of the 16-bit data are the year of manufacture, and the lower 8 bits are the month of manufacture. 11 for the start of manufacture; 2 for the 12-year start of manufacture; ...; since the year 25 the manufacturing identification is F; since the year 26 the manufacturing identification is 0; since the year 27 the manufacturing identification is 1. Month: January is marked as 1; by analogy, October is marked as A; December corresponds to C. Refer to Appendix 3 for example function, return value=read_byte(0xe8,1);
2	0x01	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 1 transmits and probe 12 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
2	0x02	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 2 transmits and probe 11 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 2 to probe 11.
2	0x03	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 3 transmits and probe 10 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 3 to probe 10.
2	0x04	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 4 transmits and probe 9 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 4 to probe 9.

2	0x05	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 5 transmits and probe 8 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 5 to probe 8.
2	0x06	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 6 transmits and probe 7 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 6 to probe 7.
2	**0x10**	**138~5267mm**	**0x88~0x1493mm**	**Probe 1 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value**
2	**0x12**	**800~32639us**	**0x320~0x7f7fus**	**Probe 1 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value**
2	**0x14**	**138~3015mm**	**0x88~0xbc7mm**	**Probe 1 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value**
2	**0x16**	**182~5267mm**	**0xb6~0x1493mm**	**Probe 1 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value**
2	**0x17**	**1055~65278us**	**0x41f~0x7f7fus**	**Probe 1 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value**
2	**0x18**	**138~5267mm**	**0x88~0x1493mm**	**Probe 2 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value**
2	**0x1a**	**800~32639us**	**0x320~0x7f7fus**	**Probe 2 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value**
2	**0x1c**	**138~3015mm**	**0x88~0xbc7mm**	**Probe 2 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value**
2	**0x1e**	**182~5267mm**	**0xb6~0x1493mm**	**Probe 2 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value**
2	**0x1f**	**1055~65278us**	**0x41f~0x7f7fus**	**Probe 2 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value**
2	**0x20**	**138~5267mm**	**0x88~0x1493mm**	**Probe 3 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value**
2	**0x22**	**800~32639us**	**0x320~0x7f7fus**	**Probe 3 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value**

2	0x24	138~3015mm	0x88~0xbc7mm	Probe 3 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x26	182~5267mm	0xb6~0x1493mm	Probe 3 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x27	1055~65278us	0x41f~0x7f7fus	Probe 3 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x28	138~5267mm	0x88~0x1493mm	Probe 4 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x2a	800~32639us	0x320~0x7f7fus	Probe 4 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x2c	138~3015mm	0x88~0xbc7mm	Probe 4 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x2e	182~5267mm	0xb6~0x1493mm	Probe 4 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x2f	1055~65278us	0x41f~0x7f7fus	Probe 4 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x30	138~5267mm	0x88~0x1493mm	Probe 5 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x32	800~32639us	0x320~0x7f7fus	Probe 5 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x34	138~3015mm	0x88~0xbc7mm	Probe 5 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x36	182~5267mm	0xb6~0x1493mm	Probe 5 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x37	1055~65278us	0x41f~0x7f7fus	Probe 5 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x38	138~5267mm	0x88~0x1493mm	Probe 6 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x3a	800~32639us	0x320~0x7f7fus	Probe 6 transceiver works independently, with an effective detection range of 13cm-4.5m. return us

				value
2	0x3c	138~3015mm	0x88~0xbc7mm	Probe 6 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x3e	182~5267mm	0xb6~0x1493mm	Probe 6 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x3f	1055~65278us	0x41f~0x7f7fus	Probe 6 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x40	138~5267mm	0x88~0x1493mm	Probe 7 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x42	800~32639us	0x320~0x7f7fus	Probe 7 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x44	138~3015mm	0x88~0xbc7mm	Probe 7 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x46	182~5267mm	0xb6~0x1493mm	Probe 7 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x47	1055~65278us	0x41f~0x7f7fus	Probe 7 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x48	138~5267mm	0x88~0x1493mm	Probe 8 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x4a	800~32639us	0x320~0x7f7fus	Probe 8 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x4c	138~3015mm	0x88~0xbc7mm	Probe 8 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x4e	182~5267mm	0xb6~0x1493mm	Probe 8 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x4f	1055~65278us	0x41f~0x7f7fus	Probe 8 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x50	138~5267mm	0x88~0x1493mm	Probe 9 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x52	800~32639us	0x320~0x7f7fus	Probe 9 transceiver works independently, with an

				effective detection range of 13cm-4.5m. return us value
2	0x54	138~3015mm	0x88~0xbc7mm	Probe 9 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x56	182~5267mm	0xb6~0x1493mm	Probe 9 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x57	1055~65278us	0x41f~0x7f7fus	Probe 9 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x58	138~5267mm	0x88~0x1493mm	Probe 10 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x5a	800~32639us	0x320~0x7f7fus	Probe 10 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x5c	138~3015mm	0x88~0xbc7mm	Probe 10 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x5e	182~5267mm	0xb6~0x1493mm	Probe 10 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x5f	1055~65278us	0x41f~0x7f7fus	Probe 10 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x60	138~5267mm	0x88~0x1493mm	Probe 11 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x62	800~32639us	0x320~0x7f7fus	Probe 11 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x64	138~3015mm	0x88~0xbc7mm	Probe 11 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x66	182~5267mm	0xb6~0x1493mm	Probe 11 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x67	1055~65278us	0x41f~0x7f7fus	Probe 11 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x68	138~5267mm	0x88~0x1493mm	Probe 12 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value

2	**0x6a**	**800~32639us**	**0x320~0x7f7fus**	**Probe 12 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value**
2	**0x6c**	**138~3015mm**	**0x88~0xbc7mm**	**Probe 12 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value**
2	**0x6e**	**182~5267mm**	**0xb6~0x1493mm**	**Probe 12 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value**
2	**0x6f**	**1055~65278us**	**0x41f~0x7f7fus**	**Probe 12 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value**
2	0x70	None	None	The first level of noise reduction. All command commands will work at the first level of noise reduction. Suitable for battery powered.
2	0x71	None	None	The second level of noise reduction. All command commands will work at the first level of noise reduction. Factory default settings. Suitable for battery powered.
2	0x72	None	None	The third level of reduction. All command commands will work at the first level of noise reduction. Suitable for USB powered.
2	0x73	None	None	The fourth level of noise reduction. All command commands will work at the first level of noise reduction. Suitable for long distance USB power supply.
2	0x74	None	None	The fifth level of noise reduction. All command commands will work at the first level of noise reduction. Suitable for switching power supply.
2	0x75	None	None	The sixth level of noise reduction. All command commands will work at the first level of noise reduction. Suitable for switching power supply.
2	0x76	None	None	Configure the serial communication baud rate to 1200bps
2	0x77	None	None	Configure the serial communication baud rate to 9600bps, the factory default setting
2	0x78	None	None	Configure the serial communication baud rate to 57600bps
2	0x79	None	None	Configure the serial communication baud rate to 115200bps
2	0x7a	None	None	About 65° beam angle in the vertical direction and 120° beam angle in the horizontal direction (the "UP

				arrow" on the back of the probe points vertically upward)
2	0x7b	None	None	About 60° beam angle in the vertical direction and 115° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x7c	None	None	About 55° beam angle in the vertical direction and 110° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x7d	None	None	About 50° beam angle in the vertical direction and 105° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x7e	None	None	About 50° beam angle in the vertical direction and 100° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x95	None	None	0x70-0x7f parameters configure the second timing
2	0x98	None	None	0x70-0x7f parameters configure the third timing
2	0x9c	None	None	0x70-0x7f parameters configure the first timing
2	0x92	None	None	Modify the second timing of the address
2	0x9a	None	None	Modify the first timing of the address
2	0x9e	None	None	Modify the third timing of the address
2	0xc2	None	None	The SCL line of I^2C is forcibly pulled low during detection, the default
2	0xc3	None	None	The SCL line of I^2C is not pulled low when probing
2	0xc4	None	None	Wait for 5s sleep
2	0xc5	None	None	Wait for 1s sleep
2		0~255	0~0xff	When reading data, register 3 is used in conjunction with register 2. Register 2 returns the upper 8 bits of the 16-bit data detection result, and register 3 returns the lower 8 bits of the 16-bit data. The return value of this register can be queried only after sending the ***address + register 2 + probe command***. Please refer to the example function in Annex 3, return value=read_byte(0xe8,2);
3		0~255	0~0xff	When reading data, register 3 is used in conjunction with register 2. Register 2 returns the upper 8 bits of the 16-bit data detection result, and register 3 returns the lower 8 bits of the 16-bit data. The return value of this register can be queried only after sending the ***address + register 2 + probe command***. Please refer to the example function in Annex 3, return

				value=read_byte(0xe8,3);
4			0x76~0x79	This register stores the serial communication baud rate 0x76~0x79 for query. 0x76 corresponds to baud rate 2400bps; 0x77 corresponds to baud rate 9600bps; 0x78 corresponds to baud rate 57600bps; 0x79 corresponds to baud rate 115200bps. Please refer to the example function in Annex 3, return value=read_byte(0xe8,4);
5			0xd0~0xfe	This register stores 20 I^2C or serial port addresses, excluding 0xf0, 0xf2, 0xf4, 0xf6, for query. Please refer to the example function in Annex 3, return value=read_byte(0xe8,5);
6			0x70~0x75	This register stores the noise reduction level 0x70~0x75 for query. Please refer to the example function in Annex 3, return value=read_byte(0xe8,6);
7			0x7a~0x7e	This register stores the beam angle size 0x7a~0x7e for query. Please refer to the example function in Annex 3, return value=read_byte(0xe8,7);
8			0xe0	normal initialization. Please refer to the example function in Annex 3, return value=read_byte(0xe8,8);
			0xe1	Hardware error.
9			0xcf	initialization in progress
			0	Initialize end flag. Please refer to the example function in Annex 3, return value=read_byte(0xe8,9);
10		0~15	0~0x0f	This register stores the working status of the four probes 12, 11, 10, and 9. This register is converted to binary number as 0B0000XXXX, and it stores the working status of four probes 12, 11, 10, and 9 in turn from left to right; "1" means normal, "0" means the probe is not connected or abnormal. KS136/KS136A does not include these probes, so the return is always 0x00. Please refer to the example function in Annex 3, return value=read_byte(0xe8,10);
11		0~255	0~0xff	This register stores the working status of eight probes 8, 7, 6, 5, 4, 3, 2, and 1. This register is converted into a binary number of 0B XXXX XXXX, which stores the working states of 8, 7, 6, 5, 4, 3, 2, 1 four

				probes in sequence from left to right; "1" means normal, "0" means not connected Probe or abnormal. CON1, CON2, CON3, CON4 of KS136/KS136A correspond to bits 5, 6, 7, and 8 of this register in turn. Other bits are always 0. Please refer to the example function in Annex 3, return value=read_byte(0xe8,11);
12		0~12	0~0x0c	This register stores the number of probes that the KS136/KS136A actually works normally after power-on. Please refer to the example function in Annex 3, return value=read_byte(0xe8,12);
13~36				Reserved for upgrade.

Table 5

Distance Detection - Single Probe Mode

The detection range of single-probe mode is 14cm-4.5m. For specific parameters and control instructions, see Table 1 above.

Fig.6

When installing, pay attention to the back of the probe, as shown in Figure 6 above, the arrow on the back of the probe is directed vertically upward. The beam angle in the direction of the probe arrow (that is, the vertical direction) is factory-set to a beam angle of about 60°, and the direction perpendicular to the direction of the probe arrow (that is, the horizontal direction) is a 115° beam angle.

Through the "I²C address + register 2 + distance detection command" sequence, delay or wait for the corresponding time specified in the above table, and then use the read function to read the value of register 2 and register 3, you can obtain 16-bit distance data . The returned mm distance value is the distance value converted from the actual detection time according to the 25°C standard; the returned us value represents the time it takes for the ultrasonic wave to be reflected and retracted from the time it encounters an obstacle.

For example, using the "0x30" probe command, the control probe 5 will be probed independently; using the "0x38" probe command, the control probe 6 will be probed independently; using the "0x40" probe command, the control probe 7 will be probed independently; using the "0x48" probe command, The control probe 8 will be probed independently.

When using single probe mode, when corresponding to 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, 0x40, 0x48, 0x50, 0x58, 0x60, 0x68 and other commands, please note that the number of the probe should be in one-to-one correspondence with the number of the main control board , cannot be mixed.

Distance Detection - Dual Probe Mode

The dual-probe mode can be installed through beam or reflection.

The dual probe mode control command is 0x01-0x06, and the specific command description is as follows in Table 6:

command	Return value range (decimal)	Return value range (Hexadecimal)	explanation
0x01	34~32639us	0x22~0x7f7fus	Probe 1 transmits and receives probe 12 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x02	34~32639us	0x22~0x7f7fus	Probe 2 transmits and receives probe 11 and returns the us value, which is the flight time of the ultrasonic wave from probe 2 to probe 11.
0x03	34~32639us	0x22~0x7f7fus	Probe 3 transmits and receives probe 10 and returns the us value, which is the flight time of the ultrasonic wave from probe 3 to probe 10.
0x04	34~32639us	0x22~0x7f7fus	Probe 4 transmits and receives probe 9 and returns the us value, which is the flight time of the ultrasonic wave from probe 4 to probe 9.
0x05	34~32639us	0x22~0x7f7fus	Probe 5 transmits and receives probe 8 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x06	34~32639us	0x22~0x7f7fus	Probe 6 transmits and receives probe 7 and returns the us value, which is the flight time of the ultrasonic wave from probe 6 to probe 7.

Table 6

Using GS precision-first firmware, the return value is as follows:

command	Return value range (decimal)	Return value range (Hexadecimal)	explanation
0x01	650us~	0x22~0x7f7fus	Probe 1 transmits and receives probe 12 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x02	650us~	0x22~0x7f7fus	Probe 2 transmits and receives probe 11 and returns the us value, which is the flight time of the ultrasonic wave from probe 2 to probe 11.
0x03	650us~	0x22~0x7f7fus	Probe 3 transmits and receives probe 10 and returns the us value, which is the flight time of the ultrasonic wave from probe 3 to probe 10.
0x04	650us~	0x22~0x7f7fus	Probe 4 transmits and receives probe 9 and returns the us value, which is the flight time of the ultrasonic wave from probe 4 to probe 9.
0x05	650us~	0x22~0x7f7fus	Probe 5 transmits and receives probe 8 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x06	650us~	0x22~0x7f7fus	Probe 6 transmits and receives probe 7 and returns the us value, which is the flight time of the ultrasonic wave from probe 6 to probe 7.

Table 7

Note that dual-probe detection returns us time values. Divide by 5.8 if you need to convert to mm distance values. For dual-probe reflection installation, please refer to the installation diagram as follows:

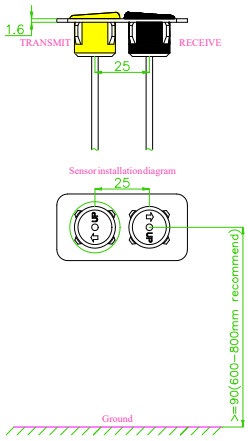


Fig.7

During installation, leave two circular holes with a distance of 25mm and a diameter of 19.2mm on the 1.6mm thick plate (see the installation hole size in Table 1). The probe is pressed and assembled in place according to the figure.

Power noise reduction commands (0x70, 0x71, 0x72, 0x73, 0x74, 0x75**), different object detection configuration commands (**0x7a-0x7e**) and baud rate (**0x77/0x79**) modification commands**

The default power supply of KS136/KS136A is recommended to be battery powered. If a noisy power supply is used, the range value may fluctuate erratically. Users can configure the clutter suppression function of the KS136/KS136A ranging module by sending 0x70, 0x71, 0x72, 0x73, 0x74, 0x75 commands. 0x70 is the level for factory testing, and the 0x71 command will configure the module as the first level of noise reduction, which is suitable for battery-powered occasions and is also the factory default setting. The 0x72 command will configure this module as the second-level noise reduction, which is suitable for occasions with certain high-frequency noise such as USB power supply. The 0x73 command will configure this module as the third-level noise reduction, which is suitable for the occasion of long-distance USB power supply. The 0x74 command will configure this module as the fourth-level noise reduction, which is suitable for the occasion of switching power supply.

User can configure baud rate by sending 0x77, 0x79. See Table 1.

The user can select the command according to the actual obstacle situation by sending 0x7a, 0x7b, 0x7c, 0x7d, 0x7e. See Table 1.

The configuration method is very simple, send the command sequence to this module: "I^2C address + register 2 + 0x9c; I^2C address + register 2 + 0x95; I^2C address + register 2 + 0x98; I^2C address + register 2 + 0x70/0x71/0x72/0x73 /0x74/0x75/0x77/0x79/0x7a/0x7b/0x7c/0x7d/0x7e", please delay at least 2 seconds after sending, so that the system can automatically complete the configuration. and start working according to the new configuration.

Take the program shown in Annex 3 as an example, configure this module as secondary noise reduction, and the configuration code is as follows:

config_0x71_0x7d(0xe8,0x72); //If the I^2C address is 0xe8

delayms(2000);

To configure this module to the maximum beam angle, the configuration code is as follows:

config_0x71_0x7d(0xe8,0x7a); //If the I^2C address is 0xe8

delayms(2000);

Configure this module to 115200 baud rate, the configuration code is as follows:

config_0x71_0x7d(0xe8,0x79); //If the I²C address is 0xe8

delayms(2000);

Please put the configuration code in the initialization function of the program, that is, before the while(1) loop, to protect the module. After KS136/KS136A receives a valid configuration command, the LED light will be on for 5s, indicating that the configuration is successful.

The KS136/KS136A will run in the new configuration permanently after power cycle. No need to configure again.

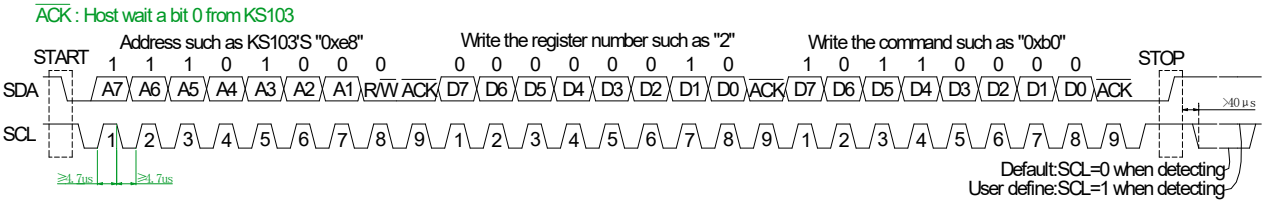
Temperature detection (KS136/KS136A do not currently support)

Temperature detection includes 4 detection instructions, 0xc9, 0xca, 0xcb, and 0xcc. Through the sequence of "I²C address + register 2 + 0xc9/0xca/0xcb/0xcc", delay or wait for the corresponding time specified in the above table, and then use The read function reads the value of register 2 and register 3, and the obtained 16-bit data follows the temperature reading rules of the DS18B20 chip. For details, please refer to the chip data of DS18B20. Take the 0xcc instruction as an example, it will obtain a total of 16 bits of probe data. The first 5 bits in the 16-bit data are sign bits. If the measured temperature is greater than 0, these 5 bits are 0. Just divide the 16-bit data by 16 or multiply by 0.0625 to obtain an ambient temperature value accurate to 0.0625 degrees Celsius. If the temperature is less than 0, these 5 bits are 1, and the actual negative temperature value can be obtained by inverting the measured 16-bit data bit by bit, adding 1 and multiplying by 0.0625. For example, when the returned 16 data is 0xfe6a, the binary value of 0xfe6a is 0B1111 1110 0110 1010, and there are 5 1s in the highest bit, so it is a negative temperature. After bitwise inversion, the binary value is 0B0000 0001 1001 0101, and the corresponding decimal value is 405, add 1 to 406, 406 times 0.0625 equals 25.375, then the ambient temperature is -25.375℃. If the returned 16-bit data is 0x1c6, its binary value is 0B0000 0001 1100 0110, and the upper 5 bits are 0, so directly multiply by 0.0625, that is, 454 times 0.0625, which is equal to 28.375°C.

Timing diagram

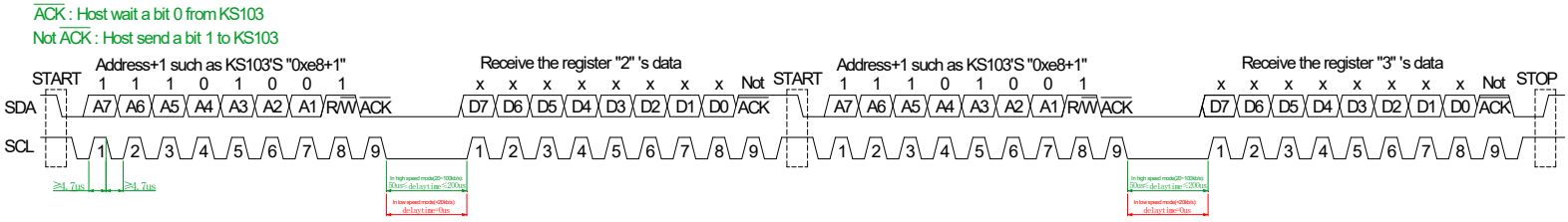
Timing diagram 1: Send a probe command, the command format is (Such as register 2):

I²C address	Register 2	8-bit data command



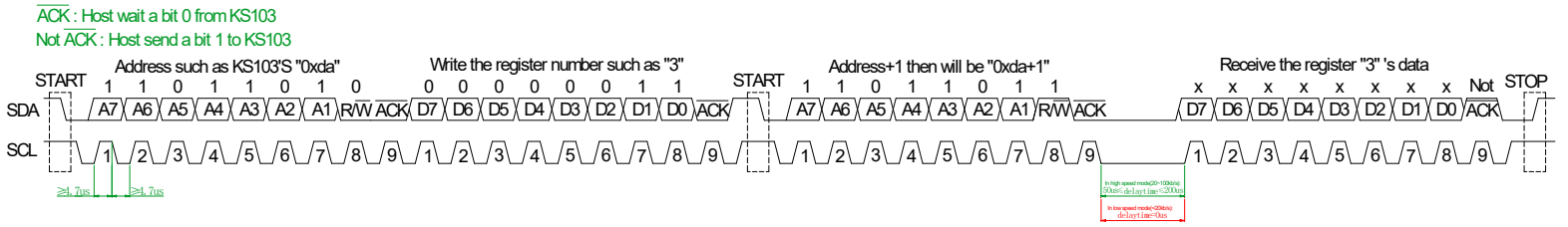
Timing diagram 2: After executing timing diagram 1, wait for SCL to become higher or after a delay of 100ms to receive 16-bit data, first high and then low, the command format is:

I²C address+1	Read register 2	I²C address+1	Read register 3



Timing diagram 3: After executing timing diagram 1, wait for SCL to become higher or receive the data of register x after a delay of 100ms (register 3 in this example), and read any register instruction format (Such as register 3)：(7)

I²C address	Register 3	I²C address+1	Read register 3



Note (7): When using the read arbitrary register command, if register 2 and register 3 are read, the probe command for register 2 must be sent first. Note that all probe instructions are stored in register 2. In the routine, the sequence of sending the probe command and then reading any register command (read register 2 + read register 3) is adopted. After writing " I²C address + 1" to KS136/KS136A, when the I²C communication rate is 20~100kb/s, 8bit data cannot be received immediately, and it is necessary to wait for the effective response of ACK low level, or delay at least 50us (delaytime), the data of the register can be received. If a delay time of at least 50us is added between writing " I²C address + 1" and "reading register 2/3", the I²C communication rate can be increased and the KS136/KS136A can still communicate reliably. When the I²C communication rate is less than 20kb/s, the delay of at least 50us (delaytime) mentioned above can be omitted. In addition, for the distance detection less than 10cm, it is recommended that the interval be greater than 1ms, otherwise there may be a problem that the last ultrasonic wave is received by the next detection. In short, there are two keys to ensure the successful establishment of I²C communication: first, the high and low level delays should not be less than 4.7us; second, the KS136/KS136A receives valid detection data from the host. The green LED flashes quickly but the return value is incorrect When the host needs to add a delay time of not less than 50us, the correct data can be obtained. Please follow the timing diagrams 1~3.

Sleep wait time setting

The default sleep mode is to wait for 5s. If no detection command is received within 5s, it will automatically enter the sleep mode. Another 1s mode is available for users to choose from. Send the data command 0xc5 through the I²C bus to enter the 1s sleep mode; send 0xc4 to restore the 5s sleep mode.

The configuration method is very simple. Send the command sequence to this module: " I²C address + register 2 + 0xc4/0xc5". After sending, please delay at least 2 seconds so that the system can automatically complete the configuration. and start working according to the new configuration.

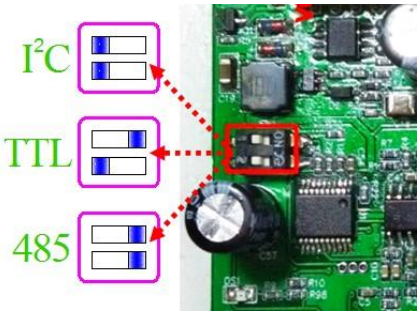
Taking the program shown in Annex 3 as an example, the configuration code is as follows:

```
write_byte(0xe8,2,0xc4);
delayms(2000);
```

After the sleep waiting time is set, the KS136/KS136A will be automatically saved and will work

according to the new configuration immediately. The KS136/KS136A will operate in the new configuration after power cycle.

TTL serial port and 485 serial port mode



The TTL or 485 serial port mode of KS136/KS136A has a baud rate of 9600bps, 1 start bit, 8 data bits, 1 stop bit, no parity bit, and TTL level. The baud rate of 9600bps can be modified to other baud rates such as 115200.

KS136/KS136A connection：

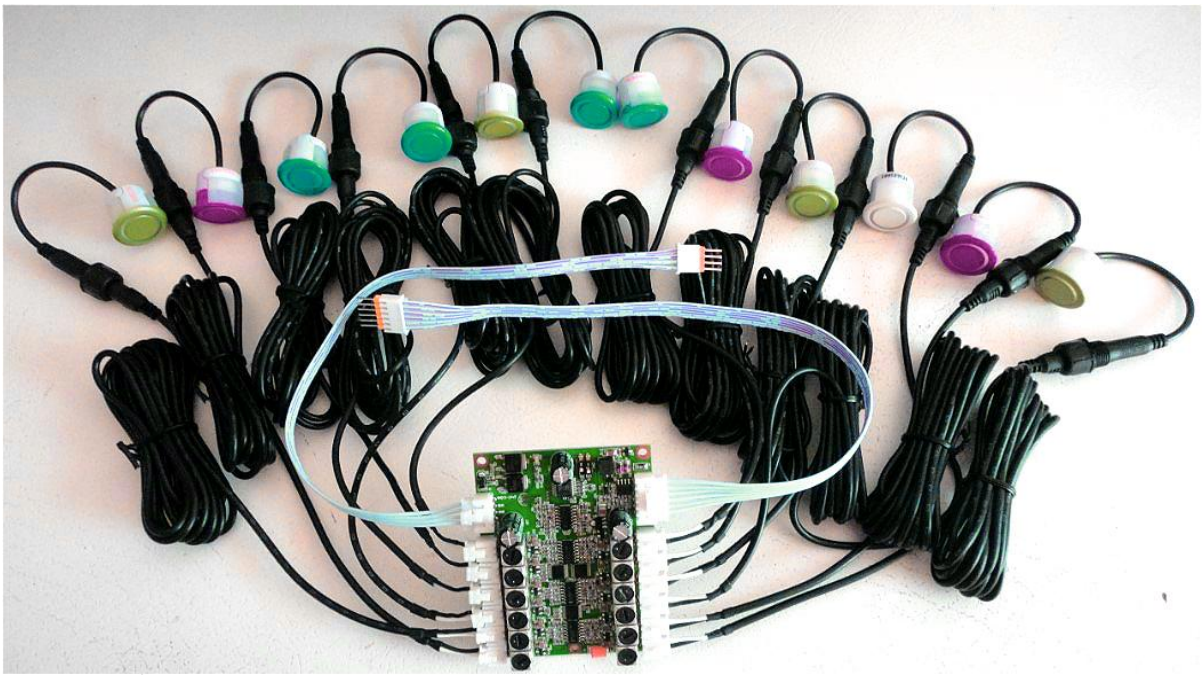


Fig.9

As shown in Figure 9, KS136/KS136A includes a main control board, which is connected to the main control board through self-locking 2PIN sockets (probe 1, probe 2, ..., probe 12 corresponds to CON1, CON2, ..., CON12) 12 2.5m cables (other lengths can be customized), and 12 waterproof probes connected with the 2.5m cables with waterproof threads. The self-locking 6PIN socket on KS136/KS136A is an external data and power interface. The pins are marked as follows: VCC (3-5.5V), SDA/TX (SDA for short), SCL/RX (SCL for short), GND, 485B, 485A; the other extended 12-24V power interface pins are: VIN(12-24V), GND, GND, VIN(12-24V).

The two VINs (12-24V) in the 4-pin socket are actually shorted internally, and the two GNDs are also shorted internally. Therefore, only one of the pins can be connected during testing. Full connection is recommended to reduce contact resistance during mass production. VIN (12-24V) voltage is recommended to use 12V. If using a 24V power supply, please pay attention to ventilation and ensure good heat dissipation. In order to pursue high power quality and high anti-interference, KS136/KS136A do not use switch chip PWM step-down, but use linear regulator to obtain better anti-interference effect.

The connection between the probe and the line is shown in Figure 10 below, using a waterproof thread connection. After plugging in, please pay attention to tighten it to meet the needs of waterproofing.

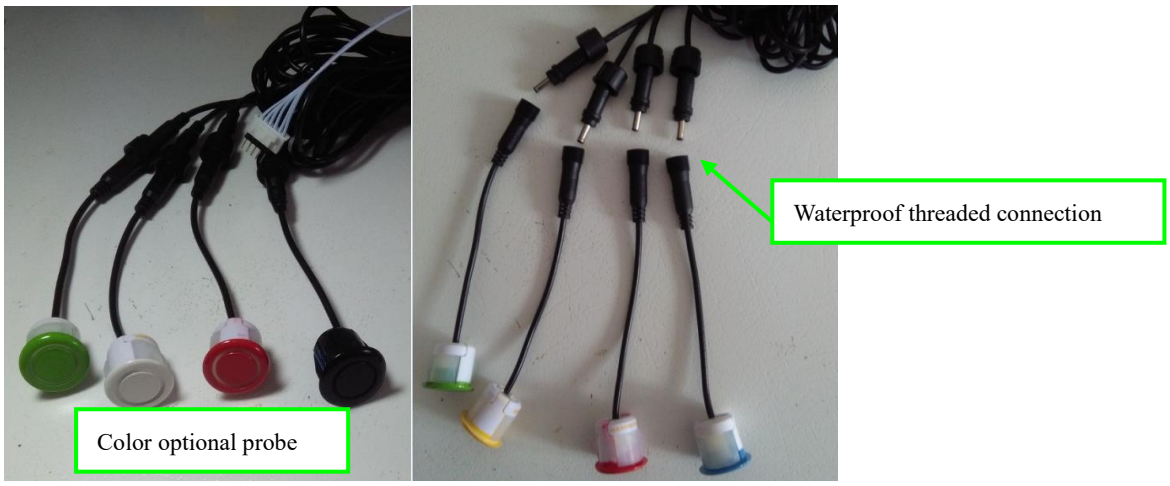


Fig.10

The TTL serial port mode connection is as follows: VCC (3-5.5V) of the six-pin socket is connected to the positive pole of the power supply with a voltage range of 3-5.5V, and the adjacent GND is connected to the negative pole; SDA/TX is connected to the RXD of the host computer; SCL/RX is connected to TXD of the host computer. The TTL serial port here is not a 232 serial port. The TTL level can be directly connected to the TXD/RXD of the microcontroller, but it cannot be directly connected to the 232 serial port (direct connection will burn the module). A MAX232 level conversion is required to convert the TTL level. Only 232 levels can be used.

The specific connection of TTL serial port mode is shown in Figure 11 below (**up to 2**):

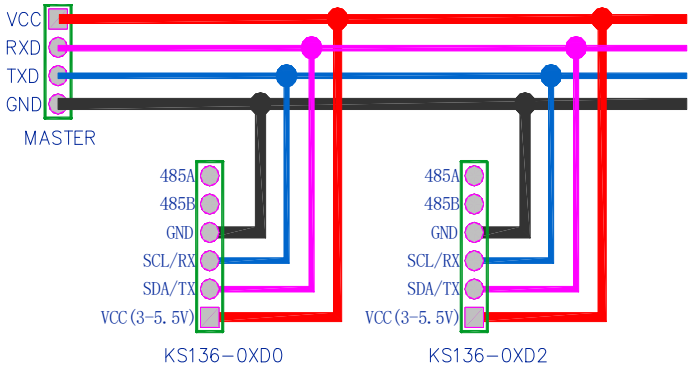


Fig.11

When using +5V power supply to supply power to KS136/KS136A, the signal line connection method in 485 serial port mode is: the VCC (3-5.5V) of the six-pin socket is connected to the positive pole of the power supply with a voltage range of 3-5.5V, and the adjacent GND is connected to the negative pole; 485A to 485A; 485B to 485B. The screen printing of KS136/KS136A-V101 adopts the cross screen printing method, that is, 485B is printed at the position of 485A, and 485A is printed at the position of 485B. Pay attention when wiring.

Using +5V power supply 485 serial port mode, the specific connection is shown in Figure 12 (**up to 20**):

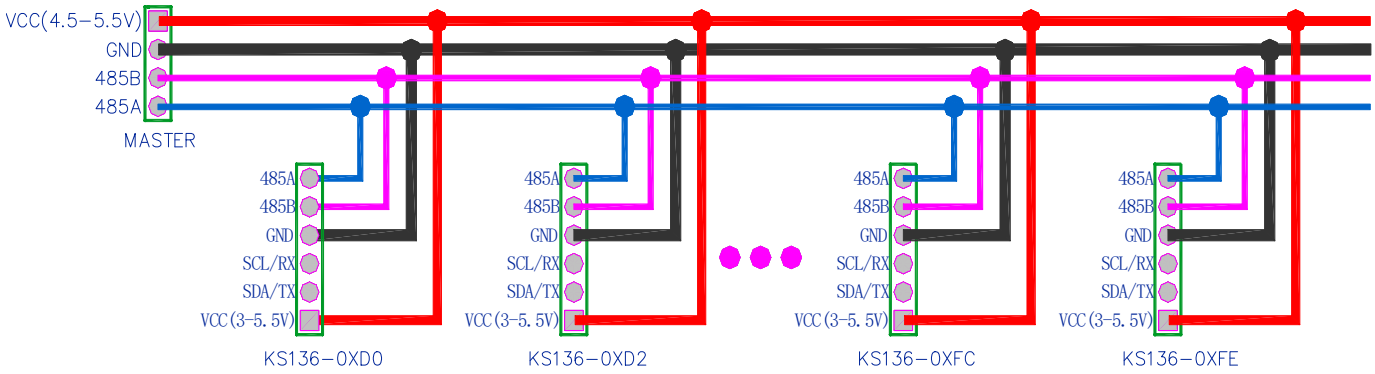


Fig.12

When using a 12V power supply, the VCC (12-24V) of the four-pin socket is connected to the positive pole of the 12-24V power supply, and the adjacent GND is connected to the negative pole.

3-5.5V power supply is recommended, if only 12-24V power supply is available, 12V power supply is recommended. If using a 24V power supply, please pay attention to ventilation and ensure good heat dissipation. In order to pursue high power quality and high anti-interference, KS136/KS136A does not use switch chip PWM to step down, but uses linear voltage regulator to obtain a better anti-interference experience.

Connect the 3-5.5V power supply, keep the 12-24V power supply floating; connect the 12-24V power supply, keep the 3-5.5V power supply floating.

When using the 485 interface, it is recommended that the power supply voltage should not be lower than 4.5V. It is recommended to use 4.5~5.5V power supply in 485 communication mode. Anti-reverse polarity protection is added to the voltage input. If the indicator light is not on, please check whether the power cord is reversed.

When using 12V power supply, the specific connection of the 485 serial port mode is shown in Figure 13 (**up to 20**):

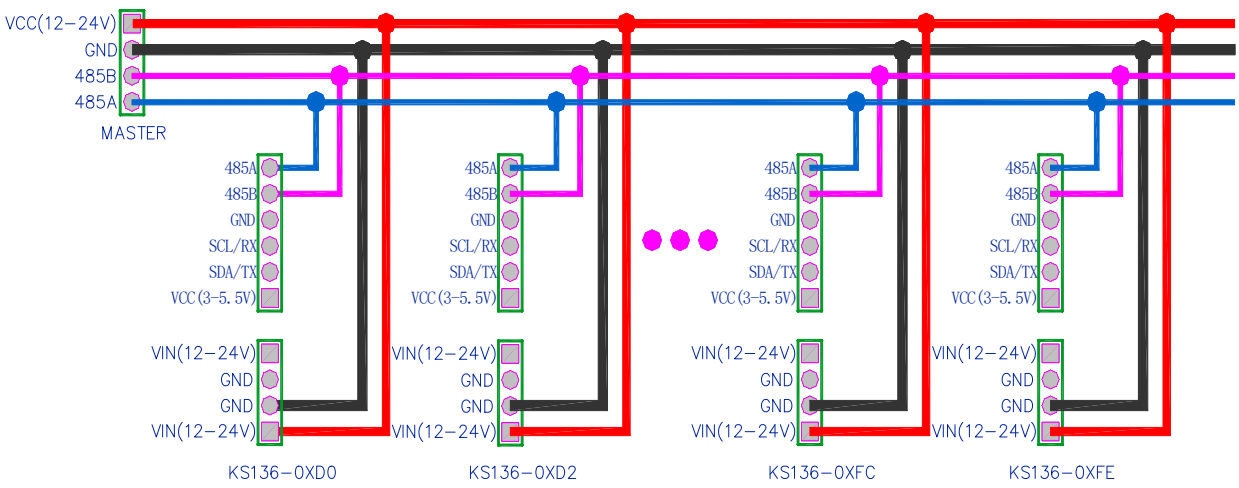


Fig.13

The default serial port address of KS136/KS136A is 0xe8, the user can modify the address to any one of 20 addresses: 0xd0, 0xd2, 0xd4, 0xd6, 0xd8, 0xda, 0xdc, 0xde, 0xe0, 0xe2, 0xe4, 0xe6, 0xe8, 0xea , 0xec, 0xee, 0xf8, 0xfa, 0xfc, 0xfe. (8)

Note (8): Please note that the above addresses do not include 0xf0, 0xf2, 0xf4, 0xf6, which are exactly the same as the I^2C version addresses. In addition, the TTL serial port protocol stipulates one-to-one, so it is recommended to use the TTL serial port mode, it is best to have only one KS136/KS136A on the TTL serial port bus, and **no more than two at most**. When using 485 serial port mode, **up to 20 KS136/KS136As can be connected to the 485 serial port bus**.

The 485 serial port and the TTL serial port are different in wiring, and the control code is completely the same. The following description of "serial port" is equivalent to "485 serial port or TTL serial port".

Modify the timing of the serial port address:

Address	2	**0x9a**	Delay 1ms	**Address**	2	**0x92**	Delay 1ms	**Address**	2	**0x9e**	Delay 1ms	**Address**	2	**New Address**	Delay 100ms

Modifying the serial port address must be done in strict accordance with the sequence, and the delay time in the sequence is the minimum time.

After the modification is completed, the LED light will keep on, power on the KS136/KS136A again, and you can observe that the LED displays the new address. In the process of modifying the serial port address of KS136/KS136A, it is strictly forbidden to power off the KS136/KS136A suddenly. Please do not put the function of modifying the address in the while(1) loop, and ensure that it only runs once after the program is powered on.

After the serial port addresses are set to different, 20 KS136/KS136A (485 mode) can be connected to the two serial lines of the host at the same time; or 2 KS136/KS136A (TTL mode) can be connected at the same time. When the host controls one of the KS136/KS136A modules, other

modules will not be affected.

KS136/KS136A's workflow:

When the KS136/KS136A is powered on, the system will first start a self-check, which takes about 1200ms. During this self-checking process, KS136/KS136A will check whether all probes are plugged in normally, and check whether each configuration is normal. If there is any abnormality, the fault position of the probe will be automatically reported. After initialization, KS136/KS136A will automatically send the following hexadecimal codes to the upper computer through the serial port:
7C 9C 79 E8 72 7B E0 00 0F FF 0C 0A 64 61 75 78 69 2E 63 6F 6D 0A 67 75 69 64 2E 74 61 6F 62 61 6F 2E 63 6F 6D 0A

其中依次地，

0x**7C**：Program version, stored in register 0；

0x**9C**：Manufacturing Date ID, is stored in register 1；

0x**79**：Serial communication baud rate, stored in register 4；

0x**E8**：8-bit

or serial port address, stored in register 5；

0x**72**：is the noise reduction level, stored in register 6；

0x**7B**：Size of beam angle, stored in register 7；

0x**E0**：Error code, stored in register 8；

0x00：Initialization end flag, stored in register 9；

0x**0F**：No. 12~9 probe self-checking flag bit; normally this bit is "1", abnormal or not connected to "0".

0x**FF**：8~No.1 probe self-checking flag bit; normally this bit is "1", abnormal or missing is "0".

0x**0C**：This register stores the number of probes that the KS136/KS136A actually works normally after power-on.

Please refer to Table 2 below for the description of each hexadecimal value. What is returned later is 0x0A, which is the newline identification. Please convert the returned value to character format for observation, which returns information such as the website of the manufacturing company.

Register	command	Return value range (decimal)	Return value range (Hexadecimal)	Remarks
0		1~254	0x01~0xff	Program version identification and manufacturer identification.
1		1~252	0x01~0xfc	Date of manufacture logo. The upper 8 bits of the 16-bit data are the year of manufacture, and the lower 8 bits are the month of manufacture. 11 for the start of manufacture; 2 for the 12-year start of manufacture; ...; since the year 25 the manufacturing identification is F; since the year 26 the manufacturing identification is 0; since the year 27 the manufacturing identification is 1; Month: January is marked as 1; by analogy, October is marked as A; December corresponds to C.

4			0x76~0x79	This register stores the serial communication baud rate 0x76~0x79 for query. 0x76 corresponds to baud rate 2400bps; 0x77 corresponds to baud rate 9600bps; 0x78 corresponds to baud rate 57600bps; 0x79 corresponds to baud rate 115200bps.
5			0xd0~0xfe	This register stores 20 I^2C or serial port addresses, excluding 0xf0, 0xf2, 0xf4, 0xf6, for query.
6			0x70~0x75	This register stores the noise reduction level 0x70~0x75 for query.
7			0x7a~0x7e	This register stores the beam angle size 0x7a~0x7e for query.
8			0xe0	normal initialization.
			0xe1	Hardware error.
9			0xcf	initialization in progress
			0	Initialize end flag.
10		0~15	0~0x0f	This register stores the working status of the four probes 12, 11, 10, and 9. This register is converted to binary number as 0B0000XXXX, and it stores the working status of four probes 12, 11, 10, and 9 in turn from left to right; "1" means normal, "0" means the probe is not connected or abnormal. KS136/KS136A does not include these probes, so the return is always 0x00.
11		0~255	0~0xff	This register stores the working status of eight probes 8, 7, 6, 5, 4, 3, 2, and 1. This register is converted into a binary number of 0B XXXX XXXX, which stores the working states of 8, 7, 6, 5, 4, 3, 2, 1 four probes in sequence from left to right; "1" means normal, "0" means not connected Probe or abnormal. CON1, CON2, CON3, CON4 of KS136/KS136A correspond to bits 5, 6, 7, and 8 of this register in turn. Other bits are always 0.
12		0~12	0~0x0c	This register stores the number of probes that the KS136/KS136A actually works normally after power-on.

Table 8

After the self-test initialization is completed, the LED shown in Figure 14 will flash in binary mode to display its 8-bit serial port address. Two quick flashes represent "1", and one slow flash represents "0". For example, display 0xea address, its binary number is 0B11101010, green LED **flashes twice quickly → off → flashes twice quickly → off → flashes twice quickly → off → flashes once slowly → off → flashes twice quickly → Off → flashes once slowly → off**

→ flashes twice quickly → off → flashes once slowly → off .(9)

Note 9: The green light when the LED flashes may irritate the eyes. Please try not to look directly at the working LED at close range. You can use the peripheral vision of your eyes to observe the flashing.

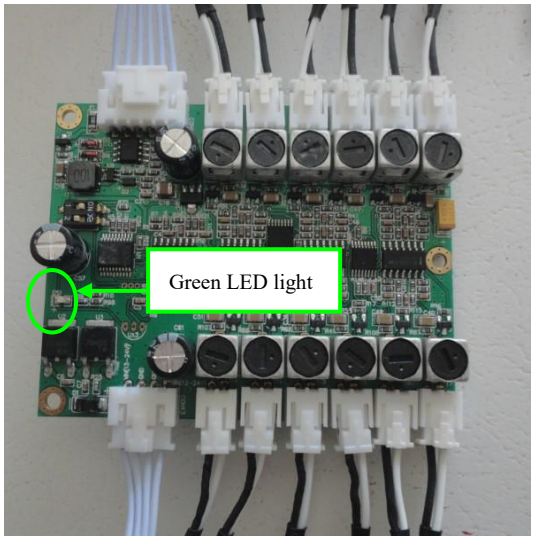


Fig.14

After the KS136/KS136A starts up, if it receives a valid data command from the host, the LED will stop flashing immediately. Enter command detection mode.

When the KS136/KS136A communicates with the host using the serial interface, it automatically responds to the control commands of the host. The command is 8-bit data, and the process of command sending and receiving detection results is as follows:

Serial port address (0xe8) —> delay 20～100us —> register (0x02) —> delay 20～100us —> detection command (0x30) —> receive the high 8 bits of detection data of KS136/KS136A through serial port —> receive KS136 /KS136A probe data lower 8 bits

When KS136/KS136A works in serial port mode, only register 0x02 can be written, and other values will not respond. When the microcontroller receives the detection result of KS136/KS136A, it can enable serial port interrupt to receive the 16-bit detection result. The detection result will send the high 8 bits first, and then send the low 8 bits. After receiving the returned 16-bit detection result, the detection command can be sent for the next round of detection, otherwise the serial port will return an incorrect value.

Detection end intelligent recognition

Since the KS136/KS136A will automatically return 16-bit detection results through the serial port after the detection command is issued, the serial port mode does not have this function.

probe command

After the detection command is sent, the KS136/KS136A will enter the corresponding detection mode according to the detection command. The host opens the serial port interrupt at this time, and the detection command cannot be sent again without receiving the returned detection result. Note that the format of each frame of detection command is:

TTL serial port address	Register 2	8-bit data

All the serial port control commands are summarized as follows：

Register	command	Return value range (decimal)	Return value range (Hexadecimal)	Remarks
2	0x01	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 1 transmits and probe 12 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
2	0x02	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 2 transmits and probe 11 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 2 to probe 11.
2	0x03	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 3 transmits and probe 10 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 3 to probe 10.
2	0x04	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 4 transmits and probe 9 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 4 to probe 9.
2	0x05	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 5 transmits and probe 8 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 5 to probe 8.
2	0x06	34~32639us	0x22~0x7f7fus	Internal test instructions. Probe 6 transmits and probe 7 receives, and returns the μs value, which is the flight time of the ultrasonic wave from probe 6 to probe 7.
2	**0x10**	**138~5267mm**	**0x88~0x1493mm**	**Probe 1 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value**
2	**0x12**	**800~32639us**	**0x320~0x7f7fus**	**Probe 1 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value**
2	**0x14**	**138~3015mm**	**0x88~0xbc7mm**	**Probe 1 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value**
2	**0x16**	**182~5267mm**	**0xb6~0x1493mm**	**Probe 1 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value**
2	**0x17**	**1055~65278us**	**0x41f~0x7f7fus**	**Probe 1 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value**
2	**0x18**	**138~5267mm**	**0x88~0x1493mm**	**Probe 2 transceiver works independently, with an**

				effective detection range of 13cm-4.5m. return mm value
2	0x1a	800~32639us	0x320~0x7f7fus	Probe 2 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x1c	138~3015mm	0x88~0xbc7mm	Probe 2 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x1e	182~5267mm	0xb6~0x1493mm	Probe 2 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x1f	1055~65278us	0x41f~0x7f7fus	Probe 2 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x20	138~5267mm	0x88~0x1493mm	Probe 3 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x22	800~32639us	0x320~0x7f7fus	Probe 3 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x24	138~3015mm	0x88~0xbc7mm	Probe 3 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x26	182~5267mm	0xb6~0x1493mm	Probe 3 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x27	1055~65278us	0x41f~0x7f7fus	Probe 3 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x28	138~5267mm	0x88~0x1493mm	Probe 4 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x2a	800~32639us	0x320~0x7f7fus	Probe 4 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x2c	138~3015mm	0x88~0xbc7mm	Probe 4 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x2e	182~5267mm	0xb6~0x1493mm	Probe 4 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x2f	1055~65278us	0x41f~0x7f7fus	Probe 4 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value

2	0x30	138~5267mm	0x88~0x1493mm	Probe 5 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x32	800~32639us	0x320~0x7f7fus	Probe 5 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x34	138~3015mm	0x88~0xbc7mm	Probe 5 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x36	182~5267mm	0xb6~0x1493mm	Probe 5 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x37	1055~65278us	0x41f~0x7f7fus	Probe 5 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x38	138~5267mm	0x88~0x1493mm	Probe 6 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x3a	800~32639us	0x320~0x7f7fus	Probe 6 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x3c	138~3015mm	0x88~0xbc7mm	Probe 6 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x3e	182~5267mm	0xb6~0x1493mm	Probe 6 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x3f	1055~65278us	0x41f~0x7f7fus	Probe 6 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x40	138~5267mm	0x88~0x1493mm	Probe 7 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x42	800~32639us	0x320~0x7f7fus	Probe 7 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x44	138~3015mm	0x88~0xbc7mm	Probe 7 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x46	182~5267mm	0xb6~0x1493mm	Probe 7 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x47	1055~65278us	0x41f~0x7f7fus	Probe 7 transceiver works independently, with an effective detection range of 20cm-5.5m. return us

				value
2	0x48	138~5267mm	0x88~0x1493mm	Probe 8 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x4a	800~32639us	0x320~0x7f7fus	Probe 8 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x4c	138~3015mm	0x88~0xbc7mm	Probe 8 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x4e	182~5267mm	0xb6~0x1493mm	Probe 8 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x4f	1055~65278us	0x41f~0x7f7fus	Probe 8 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x50	138~5267mm	0x88~0x1493mm	Probe 9 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x52	800~32639us	0x320~0x7f7fus	Probe 9 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x54	138~3015mm	0x88~0xbc7mm	Probe 9 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x56	182~5267mm	0xb6~0x1493mm	Probe 9 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x57	1055~65278us	0x41f~0x7f7fus	Probe 9 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value
2	0x58	138~5267mm	0x88~0x1493mm	Probe 10 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value
2	0x5a	800~32639us	0x320~0x7f7fus	Probe 10 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value
2	0x5c	138~3015mm	0x88~0xbc7mm	Probe 10 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value
2	0x5e	182~5267mm	0xb6~0x1493mm	Probe 10 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value
2	0x5f	1055~65278us	0x41f~0x7f7fus	Probe 10 transceiver works independently, with

				an effective detection range of 20cm-5.5m. return us value
2	**0x60**	**138~5267mm**	**0x88~0x1493mm**	**Probe 11 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value**
2	**0x62**	**800~32639us**	**0x320~0x7f7fus**	**Probe 11 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value**
2	**0x64**	**138~3015mm**	**0x88~0xbc7mm**	**Probe 11 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value**
2	**0x66**	**182~5267mm**	**0xb6~0x1493mm**	**Probe 11 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value**
2	**0x67**	**1055~65278us**	**0x41f~0x7f7fus**	**Probe 11 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value**
2	**0x68**	**138~5267mm**	**0x88~0x1493mm**	**Probe 12 transceiver works independently, with an effective detection range of 13cm-4.5m. return mm value**
2	**0x6a**	**800~32639us**	**0x320~0x7f7fus**	**Probe 12 transceiver works independently, with an effective detection range of 13cm-4.5m. return us value**
2	**0x6c**	**138~3015mm**	**0x88~0xbc7mm**	**Probe 12 transceiver works independently, 3 -meter range command ,with an effective detection range of 13cm-3m. return mm value**
2	**0x6e**	**182~5267mm**	**0xb6~0x1493mm**	**Probe 12 transceiver works independently, with an effective detection range of 20cm-5.5m. return mm value**
2	**0x6f**	**1055~65278us**	**0x41f~0x7f7fus**	**Probe 12 transceiver works independently, with an effective detection range of 20cm-5.5m. return us value**
2	0x70	None	None	The first level of noise reduction. All command commands will work at the first level of noise reduction. Suitable for battery powered.
2	0x71	None	None	The second level of noise reduction. All command commands will work at the first level of noise reduction. Factory default settings. Suitable for battery powered.
2	0x72	None	None	The third level of reduction. All command commands will work at the first level of noise reduction. Suitable for USB powered.
2	0x73	None	None	The fourth level of noise reduction. All command commands will work at the first level

				of noise reduction. Suitable for long distance USB power supply.
2	0x74	None	None	The fifth level of noise reduction. All command commands will work at the first level of noise reduction. Suitable for switching power supply.
2	0x75	None	None	The sixth level of noise reduction. All command commands will work at the first level of noise reduction. Suitable for switching power supply.
2	0x76	None	None	Configure the serial communication baud rate to 1200bps
2	0x77	None	None	Configure the serial communication baud rate to 9600bps, the factory default setting
2	0x78	None	None	Configure the serial communication baud rate to 57600bps
2	0x79	None	None	Configure the serial communication baud rate to 115200bps
2	0x7a	None	None	About 65° beam angle in the vertical direction and 120° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x7b	None	None	About 60° beam angle in the vertical direction and 115° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x7c	None	None	About 55° beam angle in the vertical direction and 110° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x7d	None	None	About 50° beam angle in the vertical direction and 105° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x7e	None	None	About 50° beam angle in the vertical direction and 100° beam angle in the horizontal direction (the "UP arrow" on the back of the probe points vertically upward)
2	0x95	None	None	0x70-0x7f parameters configure the second timing
2	0x98	None	None	0x70-0x7f parameters configure the third timing
2	0x9c	None	None	0x70-0x7f parameters configure the first timing
2	0x92	None	None	Modify the second timing of the address
2	0x9a	None	None	Modify the first timing of the address
2	0x9e	None	None	Modify the third timing of the address

2	0x99	N/A	79 E8 72 7B E0 CE **0F** **FF** 0C 08 01 08 01 08 01 98 00 00	If it is not proposed by the customer, the four characters of 79 E8 72 7B can be locked and unchanged, and the E0 CE value is the internal repair detail value and may automatically change. The initialization command is used to query whether each probe is normal. This command can accurately return which probe is faulty. Start from the 7th byte of the returned data. 0x**0F**： No. 12~9 probe self-checking flag bit; normally this bit is "1", abnormal or not connected to "0". 0x**FF**： No. 8~1 probe self-checking flag bit; normally this bit is "1", abnormal or not connected to "0".0C: This register stores the number of probes that the KS136/KS136A actually works normally after power-on.
2	0xc2	None	None	The SCL line of I²C is forcibly pulled low during detection, the default
2	0xc3	None	None	The SCL line of I²C is not pulled low when probing
2	0xc4	None	None	Wait for 5s sleep
2	0xc5	None	None	Wait for 1s sleep

Table 9

Distance Detection - Single Probe Mode

The detection range of single-probe mode is 13cm-4.5m. For specific parameters and control instructions, see Table 9 above.

When installing, pay attention to the back of the probe, as shown in Figure 6 above, the arrow on the back of the probe is directed vertically upward. The beam angle in the direction of the probe arrow (that is, the vertical direction) is factory-set to a beam angle of about 60°, and the direction perpendicular to the direction of the probe arrow (that is, the horizontal direction) is a 115° beam angle.

Through the sequence of "serial port address + register 2 + distance detection command", the host opens the serial port interrupt at this time. After the detection is completed, the value of read register 2 and register 3 can be obtained through the serial port interrupt, and 16-bit distance data can be obtained. The returned mm distance value is the distance value converted from the actual detection time according to the 25°C standard; the returned us value represents the time it takes for the ultrasonic wave to be reflected and retracted from the time it encounters an obstacle.

For example, using the "0x30" probe command, the control probe 5 will be probed independently; using the "0x38" probe command, the control probe 6 will be probed independently; using the "0x40" probe command, the control probe 7 will be probed independently; using the "0x48" probe command, The control probe 8 will be probed independently.

When using single probe mode, when corresponding to 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, 0x40, 0x48, 0x50, 0x58, 0x60, 0x68 and other commands, please note that the number of the probe should be in one-to-one correspondence with the number of the main control board , cannot be mixed.

Distance Detection - Dual Probe Mode

The dual probe mode control command is 0x01-0x06. The specific return value is as follows in

Table 10:

command	Return value range (decimal)	Return value range (Hexadecimal)	explanation
0x01	34~32639us	0x22~0x7f7fus	Probe 1 transmits and receives probe 12 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x02	34~32639us	0x22~0x7f7fus	Probe 2 transmits and receives probe 11 and returns the us value, which is the flight time of the ultrasonic wave from probe 2 to probe 11.
0x03	34~32639us	0x22~0x7f7fus	Probe 3 transmits and receives probe 10 and returns the us value, which is the flight time of the ultrasonic wave from probe 3 to probe 10.
0x04	34~32639us	0x22~0x7f7fus	Probe 4 transmits and receives probe 9 and returns the us value, which is the flight time of the ultrasonic wave from probe 4 to probe 9.
0x05	34~32639us	0x22~0x7f7fus	Probe 5 transmits and receives probe 8 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x06	34~32639us	0x22~0x7f7fus	Probe 6 transmits and receives probe 7 and returns the us value, which is the flight time of the ultrasonic wave from probe 6 to probe 7.

Table 10

Using GS precision-first firmware, the return value is as follows:

command	Return value range (decimal)	Return value range (Hexadecimal)	explanation
0x01	650us~	0x22~0x7f7fus	Probe 1 transmits and receives probe 12 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x02	650us~	0x22~0x7f7fus	Probe 2 transmits and receives probe 11 and returns the us value, which is the flight time of the ultrasonic wave from probe 2 to probe 11.
0x03	650us~	0x22~0x7f7fus	Probe 3 transmits and receives probe 10 and returns the us value, which is the flight time of the ultrasonic wave from probe 3 to probe 10.
0x04	650us~	0x22~0x7f7fus	Probe 4 transmits and receives probe 9 and returns the us value, which is the flight time of the ultrasonic wave from probe 4 to probe 9.
0x05	650us~	0x22~0x7f7fus	Probe 5 transmits and receives probe 8 and returns the us value, which is the flight time of the ultrasonic wave from probe 1 to probe 12.
0x06	650us~	0x22~0x7f7fus	Probe 6 transmits and receives probe 7 and returns the us value, which is the flight time of the ultrasonic wave from probe 6 to probe 7.

Table 11

With GXX navigation priority firmware, the baud rate will become 115200bps; the noise reduction level will be configured to 0x72, and the beam angle level will be configured to 0x7b. Use one of the 0x01~0x06 commands to return the value as shown in Table 12:

Meter ruler to measure distance value A (unit: mm)	Original return value B (unit: us)	Conversion formula B=A/5.8 (unit: mm)	Error (unit: mm)	Conversion formula B=A/5.8 (unit: mm)	Error (unit: mm)

10	Blind area				
20	Blind area				
30	564	97.24137931	67.2413793	127.2413793	97.24138
40	564	97.24137931	57.2413793	127.2413793	87.24138
50	437	75.34482759	25.3448276	105.3448276	55.34483
60	564	97.24137931	37.2413793	127.2413793	67.24138
70	564	97.24137931	27.2413793	127.2413793	57.24138
80	564	97.24137931	17.2413793	127.2413793	47.24138
90	564	97.24137931	7.24137931	127.2413793	37.24138
100	564	97.24137931	-2.7586207	127.2413793	27.24138
110	571	98.44827586	-11.551724	128.4482759	18.44828
120	571	98.44827586	-21.551724	128.4482759	8.448276
130	579	99.82758621	-30.172414	129.8275862	-0.17241
140	579	99.82758621	-40.172414	129.8275862	-10.1724
150	579	99.82758621	-50.172414	129.8275862	-20.1724
160	571	98.44827586	-61.551724	128.4482759	-31.5517
170	631	108.7931034	-61.206897	138.7931034	-31.2069
180	708	122.0689655	-57.931034	152.0689655	-27.931
190	746	128.6206897	-61.37931	158.6206897	-31.3793
200	806	138.9655172	-61.034483	168.9655172	-31.0345
210	858	147.9310345	-62.068966	177.9310345	-32.069
220	918	158.2758621	-61.724138	188.2758621	-31.7241
230	971	167.4137931	-62.586207	197.4137931	-32.5862
240	1031	177.7586207	-62.241379	207.7586207	-32.2414
250	1091	188.1034483	-61.896552	218.1034483	-31.8966
260	1151	198.4482759	-61.551724	228.4482759	-31.5517
270	1211	208.7931034	-61.206897	238.7931034	-31.2069
280	1286	221.7241379	-58.275862	251.7241379	-28.2759
290	1331	229.4827586	-60.517241	259.4827586	-30.5172
300	1398	241.0344828	-58.965517	271.0344828	-28.9655
310	1458	251.3793103	-58.62069	281.3793103	-28.6207
320	1518	261.7241379	-58.275862	291.7241379	-28.2759
330	1593	274.6551724	-55.344828	304.6551724	-25.3448
340	1631	281.2068966	-58.793103	311.2068966	-28.7931
350	1706	294.137931	-55.862069	324.137931	-25.8621
360	1751	301.8965517	-58.103448	331.8965517	-28.1034
370	1803	310.862069	-59.137931	340.862069	-29.1379
380	1843	317.7586207	-62.241379	347.7586207	-32.2414
390	1918	330.6896552	-59.310345	360.6896552	-29.3103
400	1978	341.0344828	-58.965517	371.0344828	-28.9655
410	2031	350.1724138	-59.827586	380.1724138	-29.8276
420	2103	362.5862069	-57.413793	392.5862069	-27.4138
430	2141	369.137931	-60.862069	399.137931	-30.8621

440	2178	375.5172414	-64.482759	405.5172414	-34.4828
450	2238	385.862069	-64.137931	415.862069	-34.1379
460	2291	395	-65	425	-35
470	2351	405.3448276	-64.655172	435.3448276	-34.6552
480	2411	415.6896552	-64.310345	445.6896552	-34.3103
490	2471	426.0344828	-63.965517	456.0344828	-33.9655
500	2523	435	-65	465	-35
550	2786	480.3448276	-69.655172	510.3448276	-39.6552
600	3071	529.4827586	-70.517241	559.4827586	-40.5172
650	3356	578.6206897	-71.37931	608.6206897	-41.3793
700	3686	635.5172414	-64.482759	665.5172414	-34.4828
750	3956	682.0689655	-67.931034	712.0689655	-37.931
800	4263	735	-65	765	-35
850	4533	781.5517241	-68.448276	811.5517241	-38.4483
900	4818	830.6896552	-69.310345	860.6896552	-39.3103
950	5126	883.7931034	-66.206897	913.7931034	-36.2069
1000	5411	932.9310345	-67.068966	962.9310345	-37.069
1050	5703	983.2758621	-66.724138	1013.275862	-36.7241
1100	6011	1036.37931	-63.62069	1066.37931	-33.6207
1150	6318	1089.310345	-60.689655	1119.310345	-30.6897
1200	6566	1132.068966	-67.931034	1162.068966	-37.931
1210	6572	1133.103448	-76.896552	1163.103448	-46.8966

Table 12

Note that the dual-probe detection returns the us time value that is not processed. In order to ensure that the robot is not triggered by mistake, the detection range of the entire GXX firmware is reduced to 3cm-120cm, which can also ensure a faster navigation update rate. Please note that this actual return value is us, and to convert to mm distance value, it can only be approximately divided by 5.8 or divided by 5.8+30mm (higher precision uses a more complex algorithm). At this time, the obtained distance is not an exact distance value, but an approximate estimate value with errors, as shown in Table 12 above. Since we use a segmented algorithm, we use different recognition algorithms for obstacles at different distances to avoid false triggering and ensure effective triggering. Therefore, in terms of error, it presents ultrasonic waves with an accuracy of 1mm (KS109, KS105, KS107) compared with other models of our company. The level of accuracy is completely different. This is a navigation-first error. After several months of repeated debugging with customers, a firmware that is universal. If you want to visually display the distance value, from Table 12, the difference between the result divided by 5.8+30mm and the actual distance value in the entire 3cm-120cm range is smaller.

For dual-probe reflection installation, please refer to Figure 7 for the dual-probe installation diagram.

When installing, leave two circular holes with a distance of 25mm and a diameter of 18.9mm on the 1.6mm thick plate. The probe is pressed and assembled in place according to the figure.

Take the 0x01 command and the power supply voltage of 12v as an example. As shown in Figure 7 above, the No. 1 probe is on the left, the No. 12 probe is on the right, and the arrows on the back of the probes are oriented as shown in Figure 7, that is, the arrow of No. 1 transmitting probe is horizontally left, and the arrow of No. 12 receiving probe is horizontally right. In this way, blind spot

and beam angle similar to KS103 can be achieved. The specific test beam angle diagram is shown in Figure 8:

In Figure 8, the upper right corners of the two charts are marked with the size of the reflector. When detecting people, the maximum can be detected is 1.5 meters.

Power noise reduction commands (0x70, 0x71, 0x72, 0x73, 0x74, 0x75), **different object detection configuration commands (**0x7a-0x7e**) and baud rate modification commands**

The default power supply of KS136/KS136A is recommended to be battery powered. If a noisy power supply is used, the range value may fluctuate erratically. Users can configure the clutter suppression function of the KS136/KS136A ranging module by sending 0x70, 0x71, 0x72, 0x73, 0x74, 0x75 commands. 0x70 is the level for factory testing, and the 0x71 command will configure the module as the first level of noise reduction, which is suitable for battery-powered occasions and is also the factory default setting. The 0x72 command will configure this module as the second-level noise reduction, which is suitable for occasions with certain high-frequency noise such as USB power supply. The 0x73 command will configure this module as the third-level noise reduction, which is suitable for the occasion of long-distance USB power supply. The 0x74 command will configure this module as the fourth-level noise reduction, which is suitable for the occasion of switching power supply.

User can configure baud rate by sending 0x77, 0x79. See Table 1.

The user can select the command according to the actual obstacle situation by sending 0x7a, 0x7b, 0x7c, 0x7d, 0x7e. See Table 1.

The configuration method is very simple, send the command sequence to this module: "TTL serial port address + register 2 + 0x9c; TTL serial port address + register 2 + 0x95; TTL serial port address + register 2 + 0x98; TTL serial port address + register 2 + 0x71/0x72 /0x73/0x74/0x7a/0x7b/0x7c/0x7d", please delay at least 2 seconds after sending, so that the system can automatically complete the configuration. and start working according to the new configuration.

Please put the configuration code in the initialization function of the program, that is, before the while(1) loop, to protect the module. After KS136/KS136A receives a valid configuration command, the LED light will be on, indicating that the configuration is successful.

The KS136/KS136A will run in the new configuration permanently after power cycle. No need to configure again.

Timing diagram

Send a probe command, the command format is (Only register 2):

TTL serial port address	Delay 20～100us	Register 2	Delay 20～100us	8-bit data command

It is recommended to use serial port interrupt to receive data, so that the microcontroller can take time to do other things. When the microcontroller adopts an analog serial port, please judge the level change of the SDA/TX pin according to the serial port protocol to receive data. The data are as follows:

8 bits higher of the detection result	8 bit lower of the detection result

After receiving the data, the next round of detection instructions (for example: 0xe8+0x02+0xbc) can be sent.

Sleep wait time settings

Serial port mode does not go to sleep.

Detection range (40KHz ultrasonic)

Note that dual-probe detection returns us time values. Divide by 5.8 if you need to convert to mm distance values. For dual-probe reflection installation, please refer to the installation diagram as follows:

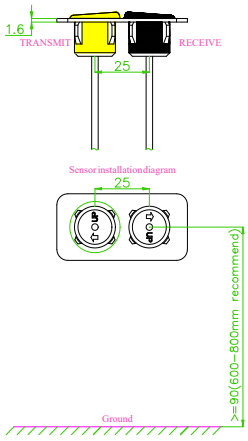


Fig.15

When installing, leave two circular holes with a distance of 25mm and a diameter of 18.9mm on the 1.6mm thick plate. The probe is pressed and assembled in place according to the figure.

Take the 0x01 command and the power supply voltage of 12v as an example. As shown in Figure 7 above, the No. 1 probe is on the left, the No. 12 probe is on the right, and the arrows on the back of the probes are oriented as shown in Figure 7, that is, the arrow of the No. 1 transmitting probe is horizontally left, and the arrow of the No. 12 receiving probe is horizontally right. In this way, blind spot and beam angle similar to KS103 can be achieved. The specific test beam angle diagram of the 40k probe is shown in Figure 8 below:

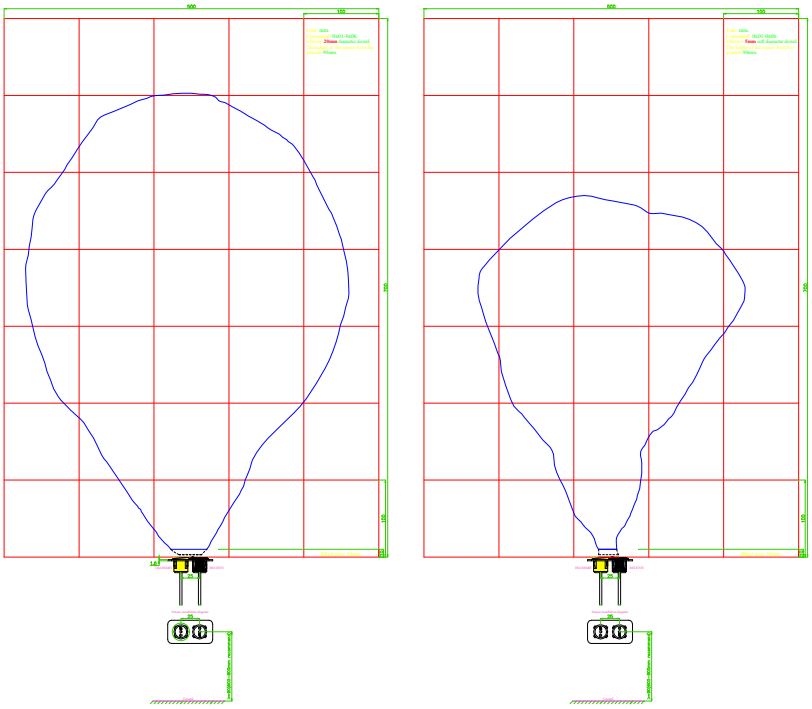


Figure 16: 40k probe beam angle

Figure 16 is a beam angle diagram of a 40k probe. The size of the reflector is marked in the upper right corner of the two charts. When detecting people, the maximum can be detected 1.2~1.5 meters.

Figure 17 is the beam angle diagram of the 49k probe, as shown.

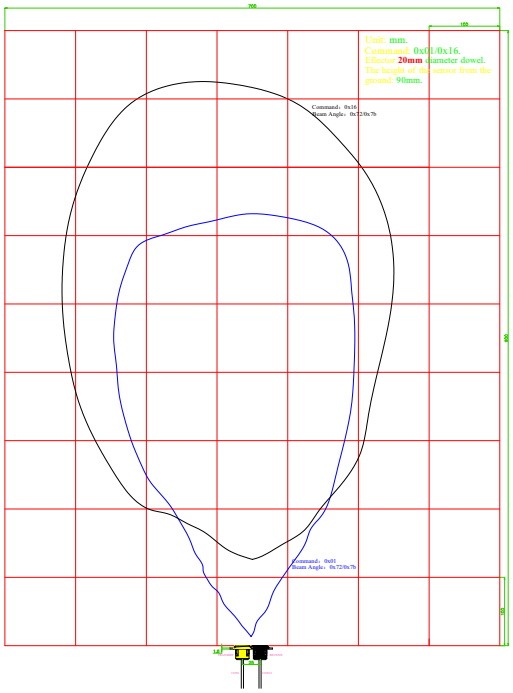


Fig.17：49 probe beam angle

No. condition	reflector size	material	Probe height	Voltage	Noise reduction

			above groud		level
Fig.A	Diameter 5mm	PVC pipe	400mm	5V	0x72
Fig.B	Diameter 20mm	304 stainless steel	400mm	5V	0x72

Remarks: The KS136/KS136A transmitter probe is on the left, the receiver probe is on the right, and the two probes are parallel to the ground; there is no obvious difference between the test results of 5V and 12V.

KS136/KS136A main control board installation dimensions (unit: mm):

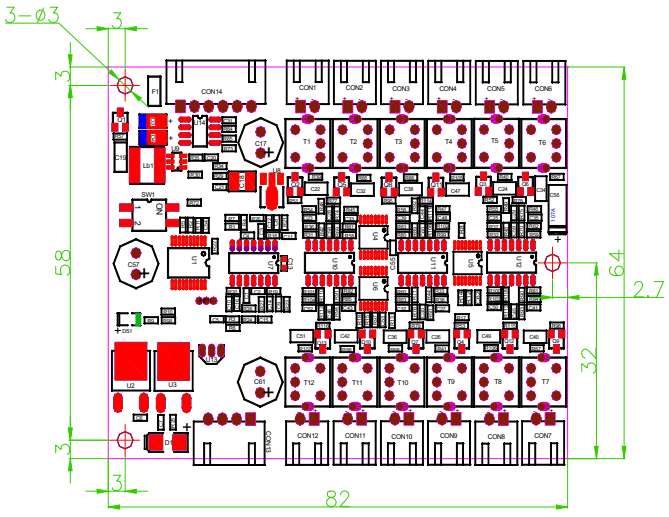


Fig.18

KS136/KS136A main control box installation dimensions (unit: mm):

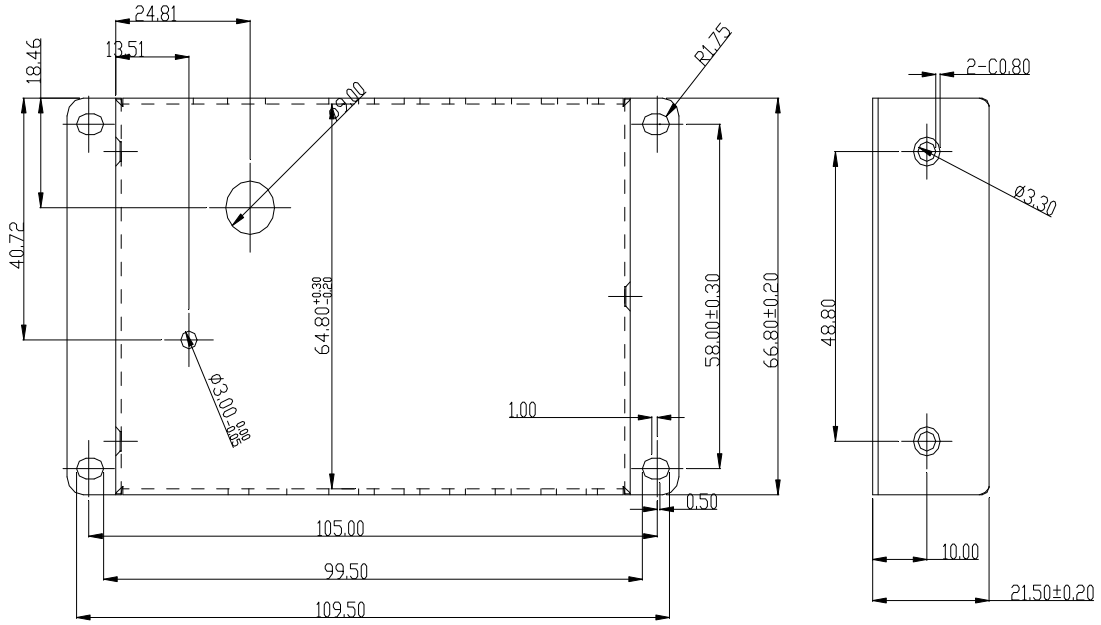


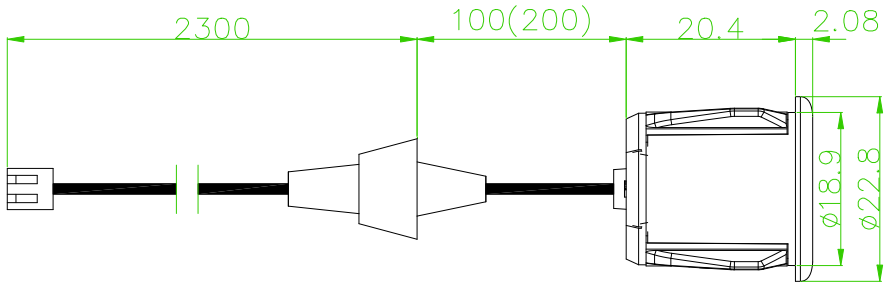
Fig.19

KS136/KS136A cable lengths are available in the following standard lengths:

The wire length has the following standard wire lengths:

30cm, 1m, 1.5m, 2.3m (factory default), 4m, 5m, 7m, 10m, 14m, 15m, 15.5m, 16m, 18m, 21m, 24m, 28m , these lengths are in stock (more than 2.3 meters, the price has increased)

Probe and wire assembly dimensions for KS136A:



All probes in Table 1 have a 1:1 3D map and can be requested from the sales guide.

Packaging (complete 1 set of KS136A-12 probe as shown below)

The delivery list is as follows: KS136A main control module: 1PCS/box; 12 probes with a length of 2.3 meters (or other cable lengths listed in this specification); 1 set of KS136A special data cables for the probe 12PCS.

Due to the needs of product improvement, this information may be modified. If the customer cannot get the modification notice in time, please obtain the latest product information on the company's website www.dauxi.com.

appendix:

1) The PIC16F877A host uses hardware **I²C communication** to connect with the KS136/KS136A to control the C code

2) The PIC16F877A host uses the analog **I²C communication** to connect with the KS136/KS136A to control the C code

3) 51 MCU host computer simulates **I²C communication** and KS136/KS136A's connection to control C code

4) STM32 CORTEX-3 ARM host simulates **I²C communication** and KS136/KS136A 's connection to control C code

1) The PIC16F877A host uses hardware **I²C communication** to connect with the KS136/KS136A to control the C code

/*Circuit connection method: The IO ports SCL and SDA of PIC16F877A are connected to SCL and SDA of KS136/KS136A. The SCL and SDA lines of PIC16F877A need to pull up a 4.7K resistor to the positive pole of the power supply VCC. */

```
#include <pic.h>                          //4MHz oscillator
__CONFIG(0x3d76);                         // open watchdog
#define DELAY() delay(10)
#difine SCL RC3                  // This pin must be pulled up with a 4.7K resistor to VCC
#difine SDA RC4                  // This pin must be pulled up with a 4.7K resistor to VCC
void setup(void);
unsigned int detect_KS101B(unsigned char ADDRESS, unsigned char command);
void delay(unsigned int ms);
void change_address(unsigned addr_old,unsigned char addr_new);
void send_command(unsigned char cmd);
void display(unsigned int distance,unsigned int delay);    // Please write the display function according to the actual wiring of the host
unsigned int distance;
void main(void)
{
    setup();
    //change_address(0xe8,0xe0);   // Change the default address 0xe8 to 0xe0
while(1)
    {
        CLRWDT();
        distance = detect_KS101B(0xe8,0x30);  //Address:0xe8; command:0x30.
                                              //Get detect result from KS136/KS136A, 16 bit data.
        display(distance,100);                //display function,you should apply it to the master
        delayms(200);
    }
}
void display(unsigned int distance,unsigned int delay);    // Please write the display function according to the actual wiring of the host
{
    CLRWDT();
}
void change_address(unsigned addr_old,unsigned char addr_new)
{
    SEN = 1;                                          // send start bit to KS136/KS136A
    while(SEN);                                       // wait for it to clear
```

```
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.

        SSPBUF = addr_old;                      // KS136/KS136A's I2C address
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.

        SSPBUF = 2;                             // write the register number
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.

        SSPBUF = 0x9a;                  //command=0x9a, change I2C address, first sequence
        while(!SSPIF);
        SSPIF = 0;

        PEN = 1;                                // send stop bit
        while(PEN);
DELAY();                                // let KS136/KS136A to break to do something

        SEN = 1;                                // send start bit
        while(SEN);                             // and wait for it to clear
        while(!SSPIF);
        SSPIF = 0;

        SSPBUF = addr_old;                      // KS136/KS136A's I2C address
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.

        SSPBUF = 2;                             // address of register to write to
        while(!SSPIF);                          //
        SSPIF = 0;

        SSPBUF = 0x92;                  //command=0x92, change I2C address, second sequence
        while(!SSPIF);                          //
        SSPIF = 0;

        PEN = 1;                                // send stop bit
        while(PEN);                             //
DELAY();                                // let KS136/KS136A to break to do something
        SEN = 1;                                // send start bit
        while(SEN);                             // and wait for it to clear
        while(!SSPIF);
        SSPIF = 0;

        SSPBUF = addr_old;                      // KS136/KS136A's I2C address
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.

        SSPBUF = 2;                             // address of register to write to
        while(!SSPIF);                          //
        SSPIF = 0;

        SSPBUF = 0x9e;                  //command=0x9e,, change I2C address,third sequence
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.

        PEN = 1;                                // send stop bit
        while(PEN);                             //
DELAY();                                // let KS136/KS136A to break to do something
        SEN = 1;                                // send start bit
        while(SEN);                             // and wait for it to clear
        while(!SSPIF);
        SSPIF = 0;
```

```
        SSPBUF = addr_old;                          // KS136/KS136A's I2C address
        while(!SSPIF);                              // wait for interrupt
        SSPIF = 0;                                  // then clear it.

        SSPBUF = 2;                                 // address of register to write to
        while(!SSPIF);                              //
        SSPIF = 0;

        SSPBUF = addr_new;          //new address, it will be 0xd0~0xfe(without 0xf0,0xf2,0xf4,0xf6)
        while(!SSPIF);                              //
        SSPIF = 0;

        PEN = 1;                                    // send stop bit
        while(PEN);                                 //
DELAY();                                    // let KS136/KS136A to break to do something
}

unsigned int detect_KS101B(unsigned char ADDRESS, unsigned char command)
{
//ADDRESS will be KS136/KS136A's address such as 0x30, command will be the detect command such as 0x30
unsigned int range=0;
        SEN = 1;                                    // send start bit
        while(SEN);                                 // and wait for it to clear
        while(!SSPIF);
        SSPIF = 0;
        SSPBUF = ADDRESS;                           // KS136/KS136A's I2C address
        while(!SSPIF);                              // wait for interrupt
        SSPIF = 0;                                  // then clear it.
        SSPBUF = 2;                                 // address of register to write to
        while(!SSPIF);                              //
        SSPIF = 0;
        SSPBUF = command;
        while(!SSPIF);                              //
        SSPIF = 0;
        PEN = 1;                                    // send stop bit
        while(PEN);                                 //

        TMR1H = 0;                                  // delay while the KS136/KS136A is ranging
        TMR1L = 0;
        T1CON = 0x31;                               //configuration of TIME1
        TMR1IF = 0;                                 //clean TIME1 interrupt flag
        while((!SCL) || (!TMR1IF))display(distance,100);    // To get continuous display, add the display function
here
        TMR1ON = 0;                                 // stop timer
        // finally get the range result from KS136/KS136A
        SEN = 1;                                    // send start bit
        while(SEN);                                 // and wait for it to clear
        ACKDT = 0;                                  // acknowledge bit
        SSPIF = 0;

        SSPBUF = ADDRESS;                           // KS136/KS136A I2C address
        while(!SSPIF);                              // wait for interrupt
        SSPIF = 0;                                  // then clear it.

        SSPBUF = 2;                                 // address of register to read from - high byte of result
        while(!SSPIF);                              //
        SSPIF = 0;                                  //

        RSEN = 1;                                   // send repeated start bit
        while(RSEN);                                // and wait for it to clear
        SSPIF = 0;                                  //
        SSPBUF = ADDRESS+1;             // KS136/KS136A I2C address - the read bit is set this time
```

```
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.
        RCEN = 1;                               // start receiving
        while(!BF);                         // wait for high byte of range
        range = SSPBUF<<8;                      // and get it
        ACKEN = 1;                                  // start acknowledge sequence
        while(ACKEN);                           // wait for ack. sequence to end
        RCEN = 1;                               // start receiving
        while(!BF);                         // wait for low byte of range
        range += SSPBUF;                        // and get it
        ACKDT = 1;                                  // not acknowledge for last byte
        ACKEN = 1;                                  // start acknowledge sequence
        while(ACKEN);                           // wait for ack. sequence to end
        PEN = 1;                                    // send stop bit
        while(PEN);                                 //
        return range;
}

void send_command(unsigned char command)     // Send an 8-bit data command to the KS136/KS136A
{
        SEN = 1;                                    // send start bit
        while(SEN);                                 // and wait for it to clear
        while(!SSPIF);
        SSPIF = 0;
        SSPBUF = ADDRESS;                       // KS136/KS136A I2C address
        while(!SSPIF);                          // wait for interrupt
        SSPIF = 0;                              // then clear it.
        SSPBUF = 2;                                 // address of register to write to
        while(!SSPIF);                          //
        SSPIF = 0;
        SSPBUF = command;
        while(!SSPIF);                          //
        SSPIF = 0;
        PEN = 1;                                    // send stop bit
        while(PEN);                                 //
}

void setup(void)                            //PIC16F877A Hardware I2C initialization configuration
{
        SSPSTAT = 0x80;
        SSPCON = 0x38;
        SSPCON2 = 0x00;
        SSPADD = 50;
        OPTION=0B10001111;//PSA = 1; Switch to 1:128 frequency division to WDT, that is, the watchdog must be
cleared once within 32.64ms
        TRISC=0B00011000;
        PORTC=0x01;
        RBIE=0;
}

void delay(unsigned int ms)
{
 unsigned char i;
 unsigned int j;
 for(i=0;i<70;i++)
        for(j=0;j<ms;j++)CLRWDT();
}
```

2) PIC16F877A The host uses analog **I²C communication** to connect with KS136/KS136A to control

the C code

```
#include <pic.h>                    //4MHz oscillator
__CONFIG(XT&WDTEN); // open watchdog
#define    SDA    RD6          // This pin must be pulled up with a 4.7K resistor to VCC
#define    SCL    RD5          // This pin must be pulled up with a 4.7K resistor to VCC
#define    SDAPORT    TRISD6 //
#define    SCLPORT    TRISD5     // Pins RD6, RD5 can be replaced with any other I/O pins
bit eepromdi;
bit eepromdo;

void delay(void)
{
        unsigned char k;
        for(k=0;k<180;k++)
                asm("CLRWDT");
}

void delayms(unsigned char ms)//ms Delay function
{
        unsigned int i,j;
        for (i=0;i<ms;i++)
                for(j=0;j<110;j++)
                asm("CLRWDT");
}

void i2cstart(void)    // start the i2c bus
{
        SCLPORT=0;
        SDAPORT=0;
        SCL=1;
        asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");
        SDA=1;
        delay();
        SDA=0;
        delay();
        SCL=0;
        delay();
}

void i2cstop(void)    // stop the i2c bus
{
        SDA=0;
        SCLPORT=0;
        SDAPORT=0;
        SDA=0;
        asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");
        SCL=1;
        delay();
        SDA=1;
        delay();
}

void bitin(void)      //read a bit from i2c bus
{
        eepromdi=1;
        SCLPORT=0;
        SDAPORT=1;
        SCL=1;
        asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");
        eepromdi=SDA;
        asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");      asm("NOP");
        SCL=0;
```

```
    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");
}

void bitout(void)    //write a bit to i2c bus
{
    SCLPORT=0;
    SDAPORT=0;
    SDA=eepromdo;
    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");
    SCL=1;
    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");
    SCL=0;
    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");    asm("NOP");
}

void i2cwrite(unsigned char sedata)    //write a byte to i2c bus
{
    unsigned char k;
    for(k=0;k<8;k++)
    {
    if(sedata&0x80)
    {
        eepromdo=1;
    }
    else
    {
        eepromdo=0;
    }
    sedata=sedata<<1;
    bitout();
    }
    bitin();
}

unsigned char i2cread(void)          //read a byte from i2c bus
{
    unsigned char redata;
    unsigned char m;
    for(m=0;m<8;m++)
    {
    redata=redata<<1;
    bitin();
    if(eepromdi==1)
    {
        redata|=0x01;
    }
    else
    {
        redata&=0xfe;
    }
    asm("NOP");
    }
    eepromdo=1;
    bitout();
    return redata;
}

unsigned char KS101B_read(unsigned char address,unsigned char buffer)
//////////////read register: address + register ,there will be 0xe8 + 0x02/0x03
{
    unsigned char eebuf3;
//  unsigned int range;
    i2cstart();
```

```
        i2cwrite(address);
        i2cwrite(buffer);
        i2cstart();
        i2cwrite(address+1);
        i2cstart();
        eebuf3=i2cread();
        i2cstop();
        return eebuf3;
}

void KS101B_write(unsigned char address,unsigned char buffer,unsigned char command)
  ////////////////////write a command: address + register + command,there will be 0xe8 + 0x02 + 0x30
{
        i2cstart();
        i2cwrite(address);
        i2cwrite(buffer);
        i2cwrite(command);
        i2cstop();
}

void change_i2c_address(addr_old,addr_new) // addr_old is the address now, addr_new will be the new address
{                                           //that you want change to
   delayms(200);                            //Protect the eeprom,you can delete this
   KS101B_write(addr_old,2,0x9a);
   delayms(1);
   KS101B_write(addr_old,2,0x92);
   delayms(1);
   KS101B_write(addr_old,2,0x9e);
   delayms(1);
   KS101B_write(addr_old,2, addr_new);
   delayms(100);                            //Protect the eeprom,you can delete this
}

unsigned int detect_KS101B(unsigned char address, unsigned char command)
{
unsigned int range1;
        KS101B_write(address,2,command);
        delayms(1);                         // Safety delay, if the display is not clear, you can increase the delay
        delayms(80);                        // If it is the detection temperature, the delay needs to be extended here,
use while(!SCL) to delete it here
//SCLPORT=1;while(!SCL);
        // delayms(80) can also be replaced with SCLPORT=1;while(!SCL); Directly querying the SCL line will have the
shortest waiting time and the fastest detection speed
        range1 = KS101B_read(address,2);
        range1 =(range1<<8) + KS101B_read(address,3);
        delayms(5);
        return range1;
}

void main(void)
{
   unsigned int range;
        //change_i2c_address(0xe8,0xfe);      //// Change the default address 0xe8 to 0xfe
        delayms(200);
        while(1)
        {
                asm("CLRWDT");
                range = detect_KS101B(0xe8,0x30);    //you just need the only one sentence to get the range.
                delayms(200);
        }
}
```

3) 51 MCU host simulates **I²C communication** and KS136/KS136A's connection to control C code

```
#include <reg51.h>
#include <intrins.h>
sbit SDA=P3^6;                          // This pin must be pulled up with a 4.7K resistor to VCC
sbit SCL=P3^7;                          // This pin must be pulled up with a 4.7K resistor to VCC
unsigned int range;

void display(unsigned int range)
{
      //input your display function, please.
}

void delay(void)          //short delay    When using a faster microcontroller, the I²C communication may not be normal.
Add 4 to 8 more _nop_(); to this function.
 {
      _nop_(); _nop_(); _nop_(); _nop_();
      _nop_(); _nop_(); _nop_(); _nop_();
      _nop_(); _nop_(); _nop_(); _nop_();
      _nop_(); _nop_(); _nop_(); _nop_();
 }

 void start(void)                  //I2C start
 {
       SDA = 1;
      delay();
      SCL = 1;
      delay();
      SDA = 0;
      delay();
 }

 void stop(void)                     //I2C stop
 {
      SDA = 0;
      delay();
      SCL = 1;
      delay();
      SDA = 1;
      delay();
 }

 void ack(void)                      //ack
 {
      unsigned char i;
      SCL = 1;
      delay();
      while(SDA == 1 && i < 200)
      {
            i++;
      }
      SCL = 0;
      delay();
 }

 void no_ack()                       //not ack
{
      SDA = 1;
```

```
        delay();
        SCL = 1;
        delay();
        SCL = 0;
        delay();
}

void i2c_write_byte(unsigned char dat)          //write a byte
 {
        unsigned char i;
        SCL = 0;
        for(i = 0; i < 8; i++)
        {
              if(dat & 0x80)
              {
                    SDA = 1;
              }
              else
              {
                    SDA = 0;
              }
              dat = dat << 1;
              delay();
              SCL = 1;
              delay();
              SCL = 0;
              delay();
        }
        SDA = 1;
        delay();
 }

 unsigned char i2c_read_byte(void)       //read a byte
 {
        unsigned char i,dat;
        SCL = 0;
        delay();
        SDA = 1;
        delay();
        for(i = 0; i < 8; i++)
        {
              SCL = 1;
              delay();
              dat = dat << 1;
              if(SDA == 1)
              {
                    dat++;
              }
              SCL = 0;
              delay();
        }
        return dat;
 }

 void init_i2c(void)                    //i2c init
 {
        SDA = 1;
        SCL = 1;
 }

 void write_byte(unsigned char address,unsigned char reg,unsigned char command) //address+register+command
 {
        init_i2c();
```

```
    start();
    i2c_write_byte(address);
    ack();
    i2c_write_byte(reg);
    ack();
    i2c_write_byte(command);
    ack();
    stop();
}

unsigned char read_byte(unsigned char address,unsigned char reg)    //address(with bit 0 set) + register
 {
    unsigned char dat;
    init_i2c();
    start();
    i2c_write_byte(address);
    ack();
    i2c_write_byte(reg);
    ack();
    start();
    i2c_write_byte(address+1);
    ack();
    delay();delay();delay();delay();delay();        // The delay here can be deleted for STC89C series microcontrollers.
For fast single//chip microcontrollers, a delay of at least 50us needs to be added to reliably read data.
    dat = i2c_read_byte();
    no_ack();
    stop();
    return dat;
 }

void delayms(unsigned int ms)       //delay ms
{
     unsigned char i;
     unsigned int j;
     for(i=0;i<110;i++)
          for(j=0;j<ms;j++);
}

void change_i2c_address(unsigned char addr_old, unsigned char addr_new)
// addr_old is the address now, addr_new will be the new address
{                                                   //that you want change to
   delayms(2000);                                    // Protect the eeprom ,you can delete this sentence
   write_byte(addr_old,2,0x9a);
   delayms(1);
   write_byte(addr_old,2,0x92);
   delayms(1);
   write_byte(addr_old,2,0x9e);
   delayms(1);
   write_byte(addr_old,2, addr_new);
   delayms(500);                                   //Protect the eeprom, you can delete this sentence
}

void config_0x71_0x7d(unsigned char addr_old, unsigned char flag)
//flag will be 0x71,0x72,0x73,0x74,0x7a,0x7b,0x7c,0x7d
{                                                   //that you want change to
   delayms(2000);                                    // Protect the eeprom ,you can delete this sentence
   write_byte(addr_old,2,0x9c);
   delayms(1);
   write_byte(addr_old,2,0x95);
   delayms(1);
   write_byte(addr_old,2,0x98);
   delayms(1);
   write_byte(addr_old,2, flag);
```

```
    delayms(500);                           //Protect the eeprom, you can delete this sentence
}

unsigned int detect(unsigned char address,unsigned char command)   //0xe8(address) + 0x30(command)
{
        unsigned int distance,count;
        write_byte(address,2,command);                  //use command "0x30" to detect the distance
        delayms(1);                                     // Safety delay, if the display is not clear, you can increase the
delay
        //delayms(80);                                  // If it is the detection temperature, the delay here should be
extended according to the time listed in Table 1
        count=800;
        while(--count || !SCL)                           // Waiting for the detection to end, reducing the count value
will reduce the detection waiting time
        {
                ;                                       // empty statement
                display(range);                         // Display statement, which can be kept or deleted as needed
        }
//      while(!SCL)display(range);                      //you can delete “display(range)”
// By querying the SCL line to intelligently identify whether the detection is over, use this statement to delete the
previous statement (count=800; while...) to save detection time
        distance=read_byte(address,2);
        distance <<= 8;
        distance += read_byte(address,3);
        return distance;                        //return 16 bit distance in millimeter
}

void main(void)
{
        //change_i2c_address(0xe8,0xfe);    //change default address 0xe8 to 0xfe
        while(1)
        {
                range = detect(0xe8,0x30);
                //0xe8 is the address; 0x30 is the command.you just need the only one sentence to get the range.
                //display(range);
                delayms(200);
        }
}
```

4) STM32 CORTEX-3 ARM Host simulates I²C **communication** and KS136/KS136A's connection to control C code

5) // Microcontroller model: STM32F103RBT // Not all system configuration functions are shown in this program

```
#include <stm32f10x_lib.h>
#include "sys.h"
#include "usart.h"
#include "delay.h"

u8 KS136/KS136A _ReadOneByte(u8 address, u8 reg)
{
        u8 temp=0;

       IIC_Start();
       IIC_Send_Byte(address);      // send low address
        IIC_Wait_Ack();
        IIC_Send_Byte(reg);      / send low address
        IIC_Wait_Ack();
```

```
	IIC_Start();
	IIC_Send_Byte(address + 1);                    //Enter receive mode
	IIC_Wait_Ack();

	delay_us(50);          // Add this code to communicate successfully!!!
	temp=IIC_Read_Byte(0);          //read register 3
	IIC_Stop();//generate a condition for stop
	return temp;
}


void KS136/KS136A _WriteOneByte(u8 address,u8 reg,u8 command)
{

	IIC_Start();
	IIC_Send_Byte(address);             // send write command
	IIC_Wait_Ack();
	IIC_Send_Byte(reg);// send high address
	IIC_Wait_Ack();
	IIC_Send_Byte(command);     // send low address
	IIC_Wait_Ack();
	IIC_Stop();//generate a condition for stop
}



void IIC_Init(void)
{
	RCC->APB2ENR|=1<<4;// Enable peripheral IO PORTC clock first
	GPIOC->CRH&=0XFFF00FFF;//PC11/12 Push-pull output
	GPIOC->CRH|=0X00033000;
	GPIOC->ODR|=3<<11;          //PC11,12 high output
}
// Generate IIC start signal
void IIC_Start(void)
{
	SDA_OUT();          //sda line output
	IIC_SDA=1;
	IIC_SCL=1;
	delay_us(10);
	IIC_SDA=0;//START:when CLK is high,DATA change form high to low
	delay_us(10);
	IIC_SCL=0;// Clamp the I2C bus, ready to send or receive data
}
// Generate IIC stop signal
void IIC_Stop(void)
{
	SDA_OUT();//sda line output
	IIC_SCL=0;
	IIC_SDA=0;//STOP:when CLK is high DATA change form low to high
	delay_us(10);
	IIC_SCL=1;
	IIC_SDA=1;// Send I2C bus end signal
	delay_us(10);
}
// Wait for the answer signal to arrive
// Return value: 1, failed to receive response
// 0, the received response is successful
u8 IIC_Wait_Ack(void)
{
	u8 ucErrTime=0;
	SDA_IN();           //set SDA as input
	IIC_SDA=1;delay_us(6);
```

```
        IIC_SCL=1;delay_us(6);
        while(READ_SDA)
        {
                ucErrTime++;
                if(ucErrTime>250)
                {
                        IIC_Stop();
                        return 1;
                }
        }
        IIC_SCL=0;// clock out 0
        return 0;
}
// generate ACK response
void IIC_Ack(void)
{
        IIC_SCL=0;
        SDA_OUT();
        IIC_SDA=0;
        delay_us(10);
        IIC_SCL=1;
        delay_us(10);
        IIC_SCL=0;
}
// not generate ACK response
void IIC_NAck(void)
{
        IIC_SCL=0;
        SDA_OUT();
        IIC_SDA=1;
        delay_us(10);
        IIC_SCL=1;
        delay_us(10);
        IIC_SCL=0;
}
// IIC sends a byte
//Return whether the slave has a response
//1, there is a response
//0, no response
void IIC_Send_Byte(u8 txd)
{
      u8 t;
       SDA_OUT();
      IIC_SCL=0;// Pull the clock low to start data transfer
      for(t=0;t<8;t++)
      {
            IIC_SDA=(txd&0x80)>>7;
            txd<<=1;
              delay_us(10);
              IIC_SCL=1;
              delay_us(10);
              IIC_SCL=0;
              delay_us(10);
      }
}
// Read 1 byte, when ack=1, send ACK, when ack=0, send nACK
u8 IIC_Read_Byte(unsigned char ack)
{
       unsigned char i,receive=0;
       SDA_IN();//set SDA as input
      for(i=0;i<8;i++ )
       {
            IIC_SCL=0;
```

```
            delay_us(10);
              IIC_SCL=1;
            receive<<=1;
            if(READ_SDA)receive++;
              delay_us(5);
      }
      if (!ack)
            IIC_NAck();//send nACK
      else
            IIC_Ack(); //send ACK
      return receive;
}

int main(void)
{
       u16 range;
       Stm32_Clock_Init(9);// System Clock Settings
       delay_init(72);              // Delayed initialization
       uart_init(72,9600); // Serial port 1 initialization
       while(1)
       {
                    KS136/KS136A _WriteOneByte(0XE8,0X02,0x30);
                    delay_ms(80);
                    range = KS136/KS136A _ReadOneByte(0xe8, 0x02);
                    range <<= 8;
                    range += KS136/KS136A _ReadOneByte(0xe8, 0x03);
       }
}
```