软件I2C搭配IP2368参考设计

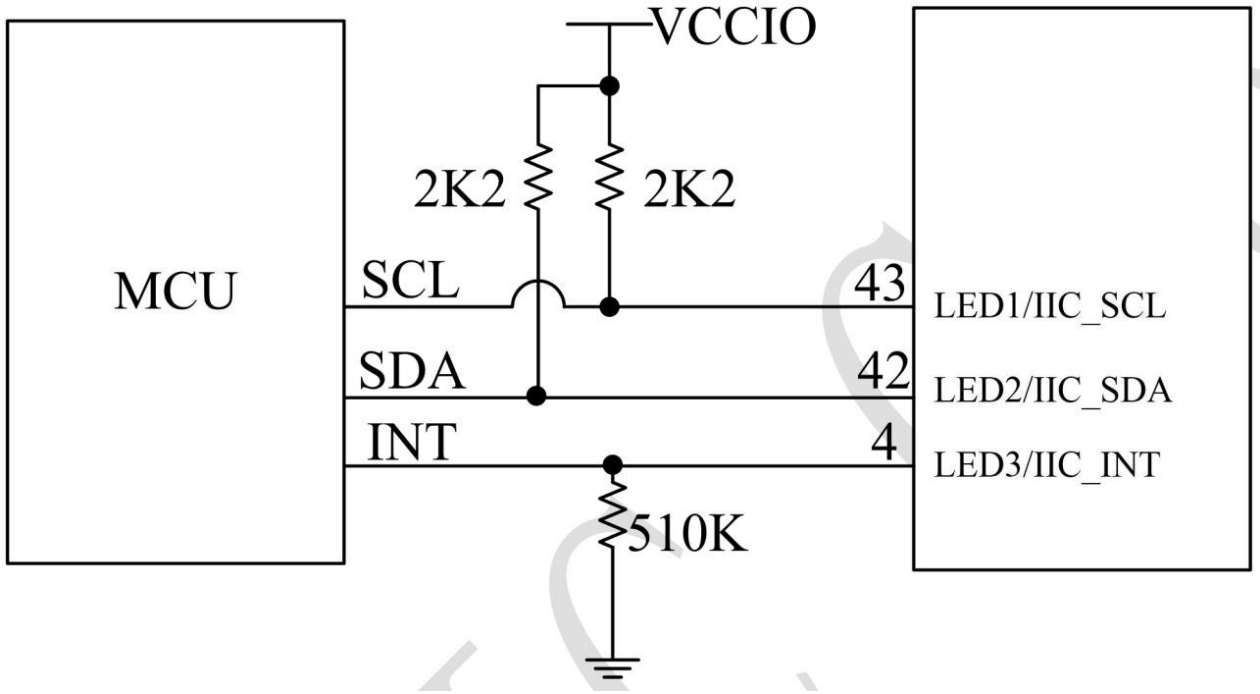
V1.1

-- 彭博

英集芯科技有限公司

IP2368 I2C Interface Hardware Description

PIN may be in an unstable state, affecting the normal judgment of the MCU)



(Note: IP2368 hibernation INT pin for input PIN, so you need to connect the pull-down 510K resistor to ensure that the signal read out from the MCU side for the low level, otherwise the voltage of the INT

The IP2368 is connected to the MCU via three wires:

1、INT -- Used to indicate the working status of IP2368 and MCU manual control

The IP2368 operates with a 510K resistor hooked up to GND;

a. When the MCU needs to get the IP2368 status, the INT pin on the MCU side should be set to **input**, the pull-up and pull-down resistors inside the MCU should be disabled, and the interrupt enable and interrupt wake-up enable of the INT pin should be turned on;

b. When the MCU needs to control the state of IP2368, the INT pin on the MCU side should be set to **OUTPUT**, and output 1 to wake up the IP2368 or force the IP2368 not to enter the hibernation mode;

c. In other cases, the INT pin of the MCU is set to **input.**

2、SCL/SDA -- I2C communication pin, according to the standard I2C protocol for communication.

The 2.2K pull-up resistor should be connected to the VCCIO of IP2368. In the non-communication state, the SCL and SDA of MCU are **input**, and the communication line is kept high by the pull-up resistor.

a. If the IO level of the MCU does not match the VCCIO, it is necessary to add a level conversion line to achieve stable and accurate data communication;

b, MCU needs to output SCL/SDA high level, can not directly**output** high level, need to configure IO for **input**, by the external pull-up resistor to realize the output high level .

INT pin application note

- IP2368 INT 应用说明：IP2368 休眠时检测到 INT 为高就会唤醒，唤醒之后，IP2368 主动拉高 INT，100ms 之后，MCU 可进行 I2C 通信，进行寄存器的读写操作；IP2368 在进入休眠之前，会切换为输入高阻来检测 INT 状态，如果为高电平，则认为 MCU 不允许 IP2368 进入休眠，如果为低电平，则 IP2368 进入休眠；MCU 在检测到 INT 为低后，16ms 内要停止访问 IC；

Note the **following points** in this documentation note:

1, IP2368 normal wake-up (charging, device insertion), by the IP2368 to control the INT, before hibernation to detect whether the INT PIN is pulled high by the external MCU, if it is pulled high, it does not enter the hibernation until the INT PIN of the external MCU withdraws the high level. MCU in this application scenario need to set the INT pin as **input**, and open the INT PIN interrupt and interrupt wake-up, so that the MCU can and quickly respond to the INT of the IP2368. The MCU in this application scenario needs to set the INT pin to input and turn on the INT PIN interrupt and interrupt wakeup, so that the MCU can被唤醒 and quickly respond to the INT状态变化 of the IP2368.

2, IP2368 if the external MCU's INT PIN high level to wake up, IP2368 wake up, INT PIN control will be transferred back to the IP2368, high and low levels indicate the work / hibernation state, the external INT can be controlled by the IP2368 can not enter the hibernation.

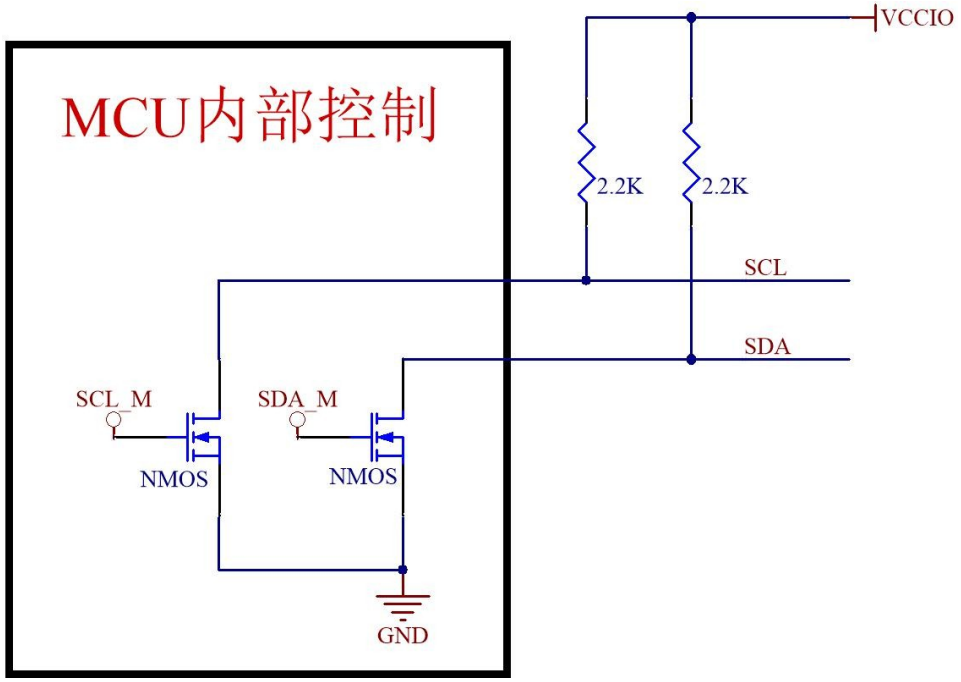
3, the MCU's INT PIN control can wake up the IP2368, you can let the IP2368 does not enter the hibernation,

but can not directly control the IP2368 to enter the hibernation state, IP2368 The hibernation control is still determined by the internal state machine.

4, according to the description of the document, MCU read INT PIN 0, 16ms to stop accessing the IP2368, so it is recommended that the MCU in the read and write data before judging INT

The IP2368 will not start reading or writing until the state of the IP2368 is high, and will not read I2C data from the IP2368 until the state of the IP2368 is low.

SCL/SDA pin application note



According to the I2C protocol, the output stage of the device connected to the I2C bus must be open drain or open collector in order to perform the function of the line with, and most of the MCU GPIOs may not have the open drain function, in which case we can use the following way to **simulate the open drain**:

1. **Disable the** GPIO pull-up and pull-down resistors;

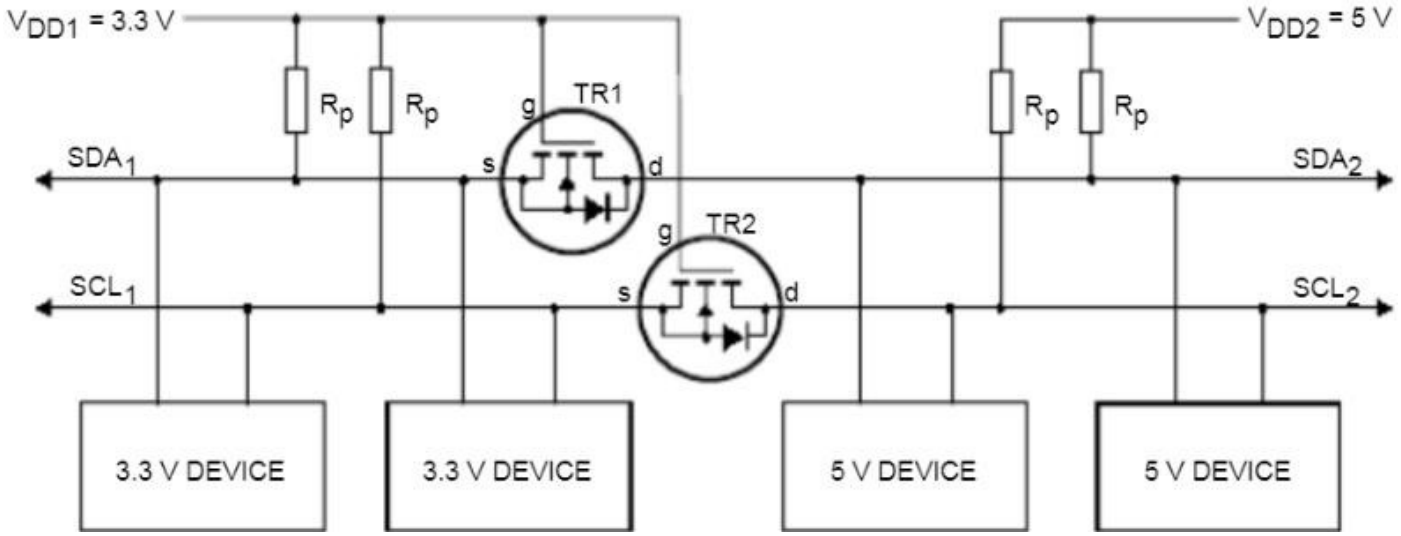
2、When the SCL or SDA of MCU needs to output high level (or idle state), set the GPIO of SCL/SDA to **input** mode, and rely on the external 2.2K pull-up resistor to maintain SCL/SDA at high level;

3, when the MCU's SCL or SDA need to output a low level, the SCL / SDA GPIO set output 0, and then set the GPIO **output** mode (different MCU's this sequence may not be the same, need to be measured), the GPIO output 0, the level on the bus to pull down (the power consumption on the bus current = VCCIO/2.2K);

SCL/SDA pin application note

4. Realize open-drain bus output by the above operation

SCL/SDA Level Conversion



In actual practice, the IP2368 uses a 3.3V I2C bus voltage, but it is possible for the MCU to use 5V or other voltages, so that when the IP2368 is connected to the MCU, it is necessary to add a level conversion circuit to match the voltage difference. Level conversion principle:

1, IP2368 is located at the left end of the line, using 3.3V system, MCU is located at the right end of the line, using 5V system ;

2、When SDA1 and SCL1 output high level, SDA1 and SCL1 are both pulled up to 3.3V by Rp, the GS voltage of TR1 and TR2 is 0, and they do not conduct, SDA2 and SCL2

Pull-up from Rp to 5V for level shifting;

3, when SDA2, SCL2 output high level, SDA2, SCL2 are pulled up to 5V by Rp, TR1, TR2 GS voltage is 0, does not conduct, SDA1 and SCL1 by Rp to 3.3V, to realize the level conversion;

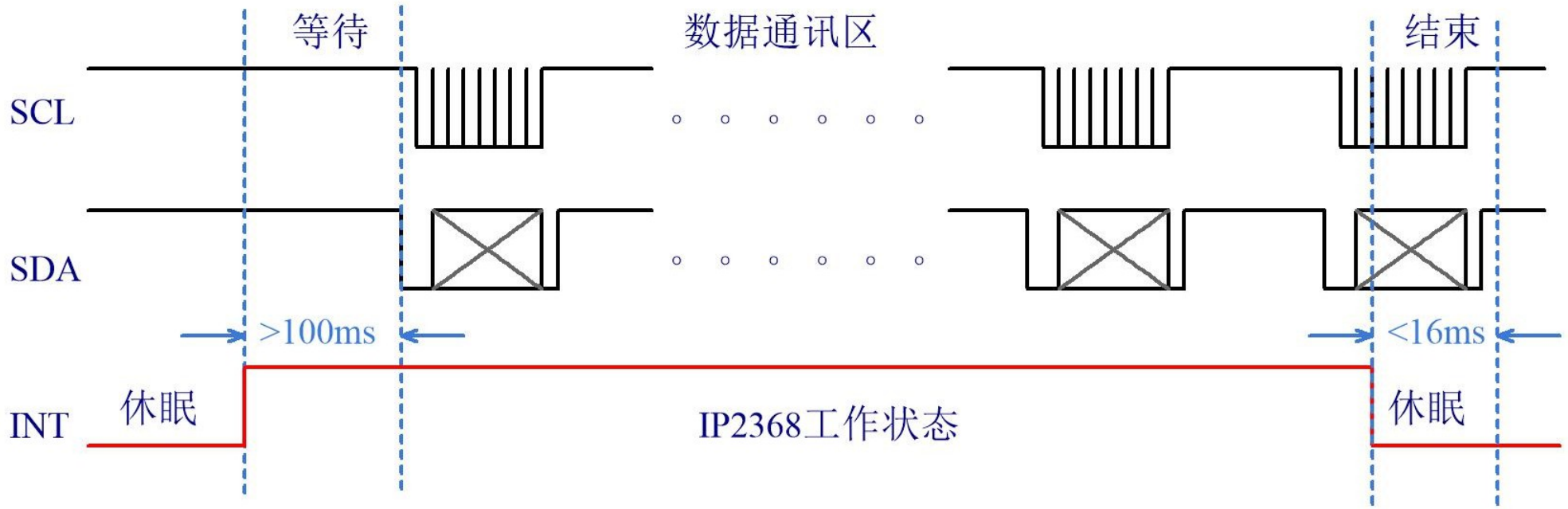
4, when SDA1, SCL1 by IP2368 pulled to a low level, TR1, TR2 GS voltage of 3.3V, DS pole conduction, SDA2,

SCL/SDA Level Conversion

SCL2 by SDA1, SCL1 pulled to a low level , to realize the level conversion;

5. When SDA2 and SCL2 are pulled to low level by MCU, the GS voltage of TR1 and TR2 is (3.3V-0.7V), the DS pole conducts, and SDA1 and SCL1 are pulled to low level by SDA2 and SCL2 to realize the level conversion.

Relationship between the INT signal and the I2C data bus



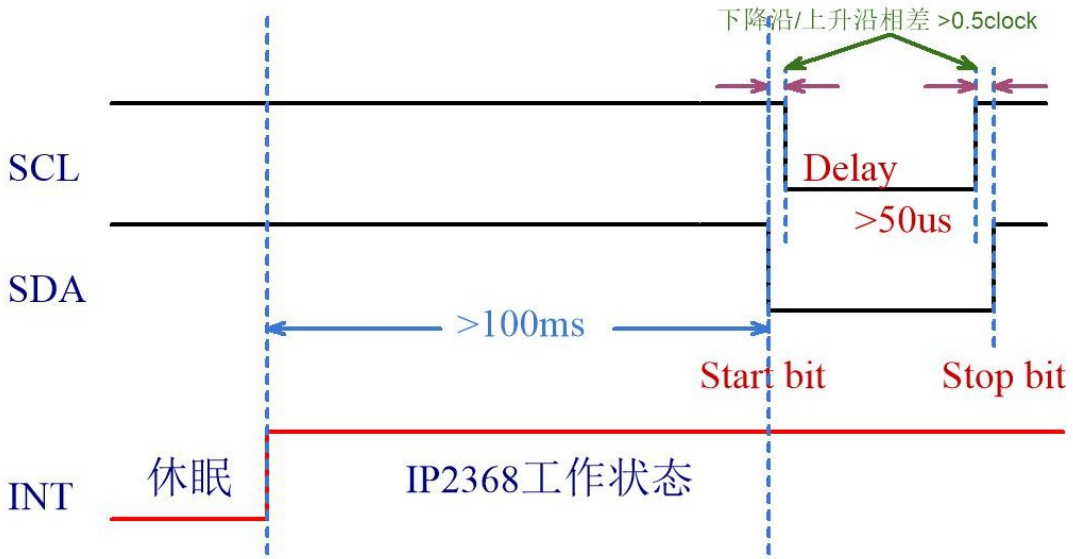
1. When the IP2368 sleeps, the INT pin is continuously low, and I2C communication is not possible at this time;
2. When the rising edge of INT occurs, no matter the MCU is forced to pull high to wake up or the IP2368 wakes up by itself, it has to wait for more than 100ms before I2C communication can be carried out;
3. When the falling edge of INT occurs, there are two cases:
   a, IP2368 is self-wake-up, MCU detects the falling edge of INT, it is best to stop the I2C data communication instantly, if it is already in communication, you need to detect the slave's answer signal in real time to determine whether the frame data is valid or not, in addition,

Relationship between the INT signal and the I2C data bus

after the falling edge of INT, the control of 16ms to stop the I2C communication completely;

b. IP2368 is woken up by MCU pulling up INT. When MCU sets INT low, IP2368 does not directly enter hibernation at this time and waits for hibernation conditions to be established, MCU can determine the current status of IP2368 by reading registers and judging the slave's answer signal.

I2C Data Communication Explained - Pre-Operation



The MCU follows the following rules when using I2C to access the IP2368:

1, I2C bus definition, SCL and SDA for the high level of continuous high level for the idle state, so in the unused I2C, need to MCU SCL and SDA are set to input PIN to ensure that the idle state

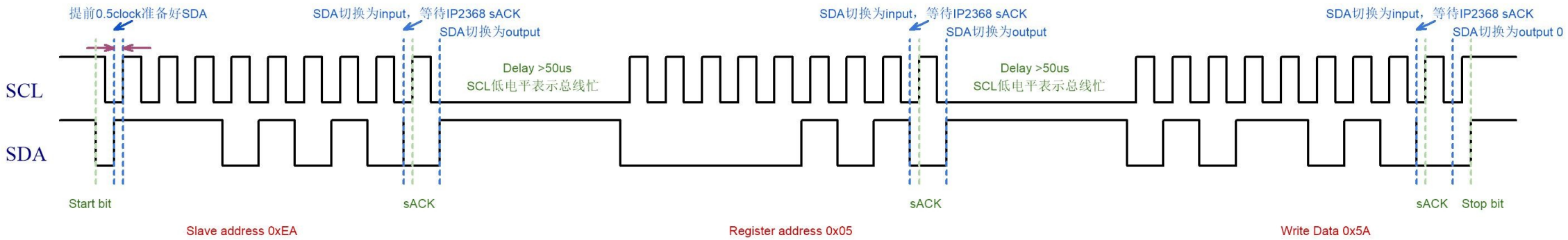
2、After IP2368 wakes up from hibernation, an interval of more than 100ms is required before I2C control is performed.

3, IP2368 first sleep wake up or I2C data without SACK answer or long time no I2C communication, MCU can perform a pre-operation before data access on the I2C bus (Start bit -> Stop bit), reset the data timing on the I2C bus to ensure that the next data frame starts normally.

I2C Data Communication Explained - Pre-Operation

4, the MCU needs to monitor the INT pin status in real time, when the IP2368 output INT rising edge, indicating that the IP2368 is waking up, in the IP2368 sleep before the I2C data written, wake up is reset to the default value, such as data writing requirements, you need to re-write once!

I2C Data Communication Explained - Write Data



IP2368 only supports single-byte data writing, it does not support continuous data writing. The timing waveform of single-byte writing is shown in the above figure, please note the following matters: 1. The level changes of SCL and SDA should be separated from each other by more than 0.5 clocks to ensure that the data can be recognized correctly;

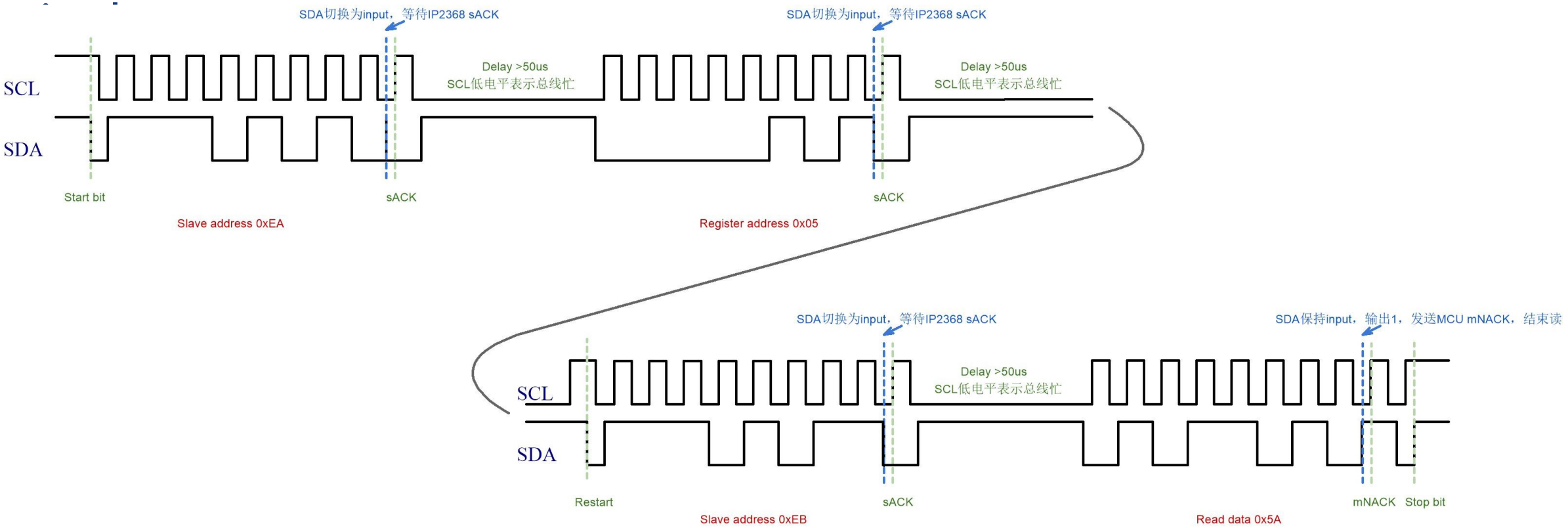
2. sACK is the answer signal of IP2368, driven by IP2368, low level is valid answer, before sACK, MCU needs to set SDA PIN as **input** PIN to ensure that the detection of SDA is answered by IP2368 pull down;

3、If IP2368 does not answer, it means that I2C is abnormal or not ready, you can set to wait for 50us, if the time is up and no sACK is seen, then the data frame is lost and the data is re-transmitted or bus reset pre-operation is performed;

I2C Data Communication Explained - Write Data

4, a complete data frame, each byte of data to be more than 50us interval between the data frame is completed, there needs to be a Stop bit to end the data frame.

I2C Data Communication Explained - Read Data



IP2368 supports single-byte data reading, the timing waveform of single-byte reading is shown in the figure above, please note the following matters: 1. The level changes of SCL and SDA should be separated from each other by more than 0.5 clock to ensure that the data can be recognized correctly;

2, sACK for IP2368 answer, driven by the IP2368, low level for a valid answer, sACK before the MCU needs to set the SDA PIN to **input** PIN, to ensure that the detection of SDA by the IP2368 pull down

I2C Data Communication Explained - Read Data

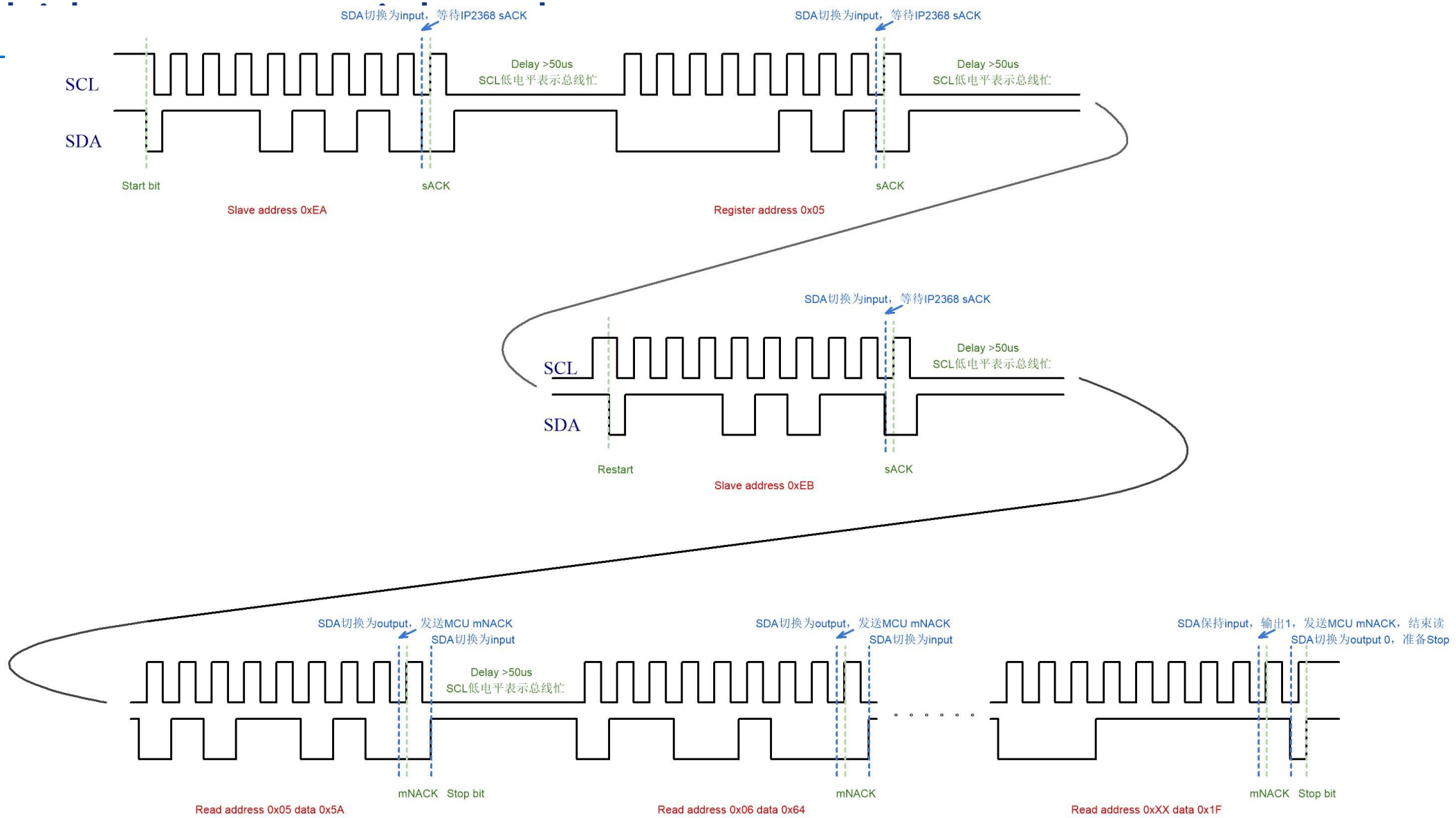
Single Byte

answer. If IP2368 does not answer, it means that I2C is abnormal or not ready, you can set to wait for 50us, if the time is up and no sACK is seen, then the data frame will be lost and the data will be resent or bus reset pre-operation will be performed;

3、After the data byte is read, mNACK is answered by MCU, a high level indicates that the data has been read, and then Stop bit can be used.

I2C Data Communication Explained - Read Data

Mu



I2C Data Communication Explained - Read Data

Multi-byte sequential read

IP2368 supports continuous reading of multi-byte data. The timing waveform of multi-byte reading is shown on the previous page, please note the following matters: 1. The level changes of SCL and SDA should be separated from each other by more than 0.5 clocks to ensure that the data can be recognized correctly;

2, sACK for IP2368 answer, driven by the IP2368, low level for a valid answer, sACK before the MCU needs to set the SDA PIN to **input** PIN, to ensure that the detection of SDA by the IP2368 pull down answer. If IP2368 does not answer, it means that I2C is abnormal or not ready, you can set to wait for 50us, if the time is up and no sACK is seen, then the data frame will be lost and the data will be resent or bus reset pre-operation will be performed;

3. After reading each byte of data, mNACK is answered by the MCU. If it is necessary to continue to read the next byte of data (the address will be automatically +1), mNACK outputs a low level, indicating that it is necessary to continue to read the data, and then it continues to send the SCL of the next byte of data to read the data;

4, if all the data has been read, then mNACK is set to high level to indicate that the data has been read,

and then Stop bit can be;

5, each byte reading interval is also required to be greater than 50us or more.

I2C Data Communication Test Tool

¥258.00

USB转I2C适配器 模块 USB-IIC/GPIO/

JSTXSZ旗舰店

月成交 0笔 评价 0

¥88.00

FT232H/FT2232HL模块USB转FIFO/S

豪兴杰旗舰店

月成交 27笔 评价 16

¥440.00

高速USB转SPI I2C PWM ADC GPIO U

JSTXSZ旗舰店

月成交 0笔 评价 0

¥619.00

工业级USB转SPI I2C PWM ADC GPIO

JSTXSZ旗舰店

月成交 0笔 评价 0

¥688.00

工业级USB转SPI I2C PWM GPIO UAR

JSTXSZ旗舰店

月成交 0笔 评价 0

¥105.90

FT232H/FT2232HL模块USB转FIFO/S

yourcee旗舰店

月成交 3笔 评价 28

¥2120.00

TP250310 Aardvark I2C/SPI 主机适配

佰斯浩数码专营店

月成交 0笔 评价 0

¥75.90

USB转I2C IIC SPI串口调试工具信号转

瑟雷西旗舰店

月成交 40笔 评价 40

¥50.00

原装CY7C65213-28PVXIT CY7C6521

轩业旗舰店

月成交 0笔 评价 0

¥820.00

原装 吉阳光电GY7512 USB-I2C Adapt

菲事尔旗舰店

月成交 0笔 评价 0

¥32.00

CJMCU-200 USB转I2C模块 FT200XD

豪兴杰旗舰店

月成交 0笔 评价 0

¥880.00

高速USB转SPI I2C PWM ADC GPIO U

hpobop旗舰店

月成交 0笔 评价 0

¥1425.00

(NI USB-8451 USB-SPI/I2C)完全兼容

爱乐通旗舰店

月成交 0笔 评价 0

¥924.00

工业级高速USB转SPI I2C PWM GPIO

zyv旗舰店

月成交 0笔 评价 0

¥834.00

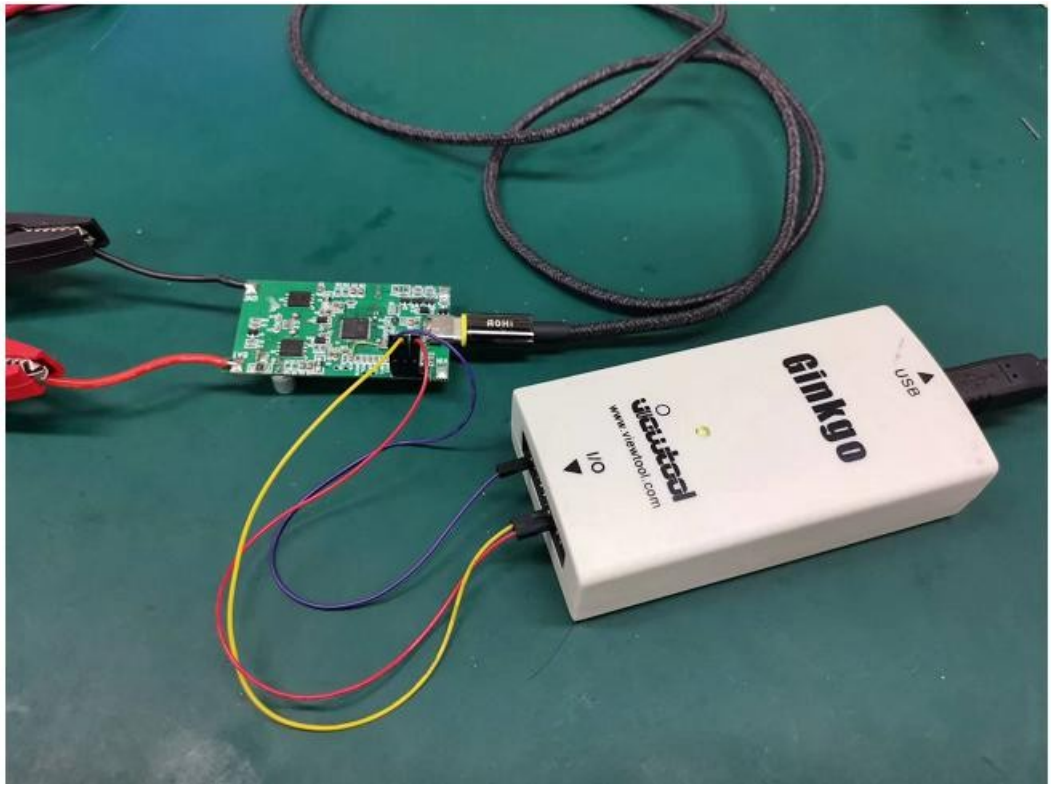
高速USB转SPI I2C PWM ADC GPIO U

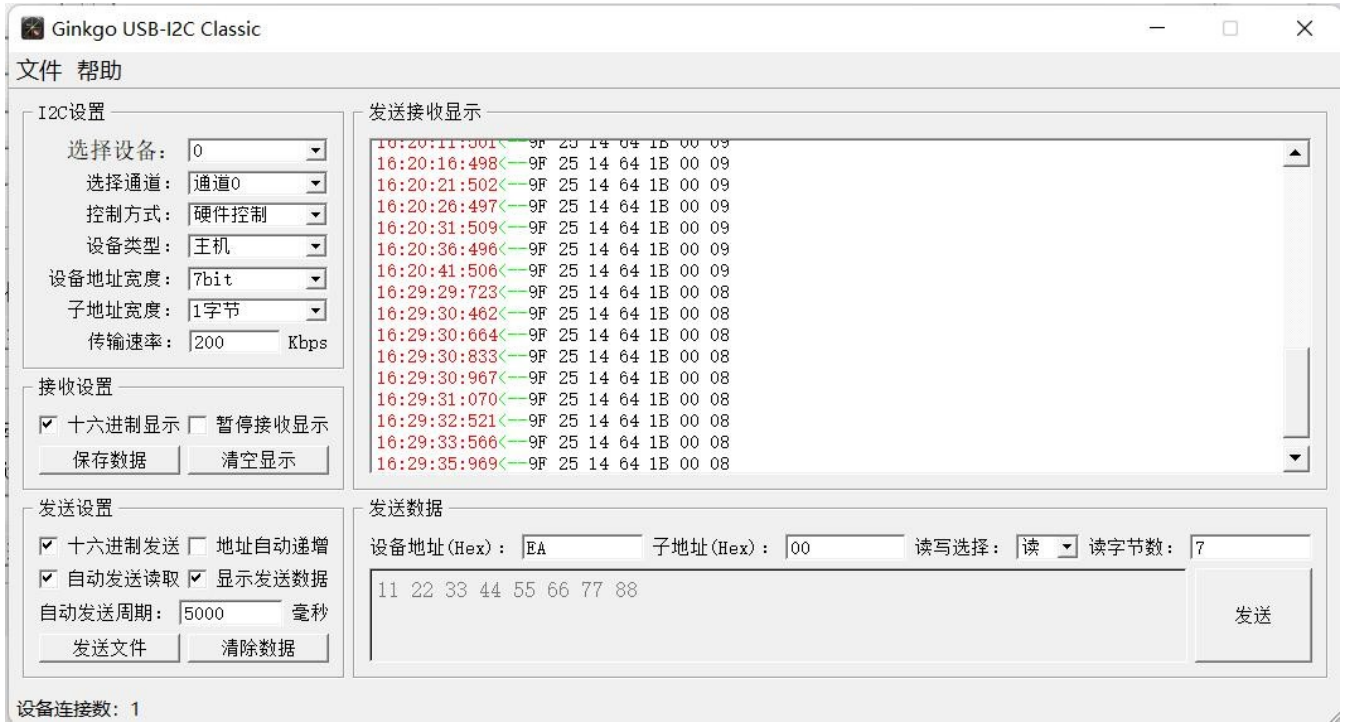
zyv旗舰店

月成交 0笔 评价 0

Development engineers in the use of IP2368 before the I2C, it is best to purchase a USB to I2C tool, the first test to confirm that the I2C is working properly, and can verify the effect of writing various types of registers, in the MCU software I2C development, but also can be used to compare the waveforms to find the problem point.

I2C Data Communication Test Tool





The I2C tool is used as shown above, the tool is connected to the IP2368 demo board through 3 wires: SCL, SDA, GND, after the IP2368 is working (charging or discharging), you can read and write the data, and the APP settings on the PC side are as shown in the right figure, pay attention to the transmission rate is not too high.

INJOINIC

Thanks for viewing!