

WINCOM[®]



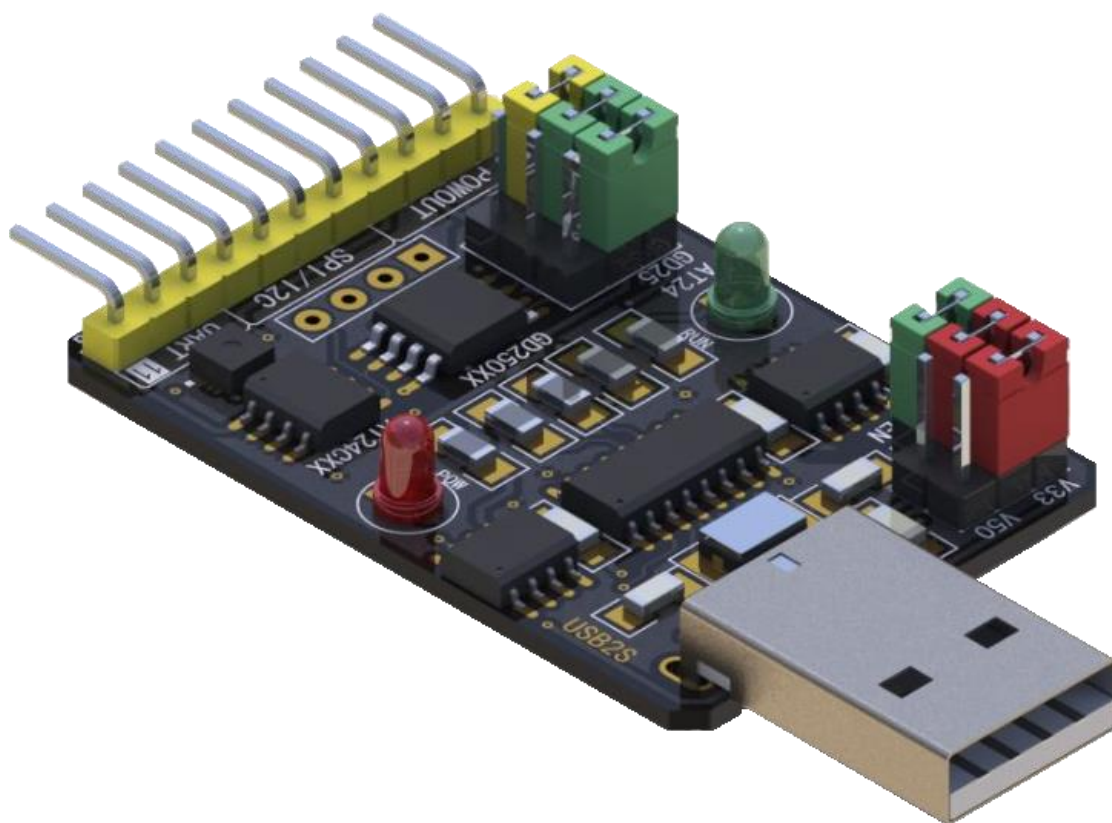
可编程 USB 转串口适配器

UART/I2C/SMBus/1-Wire/SPI/CAN/PWM

USB2S

(USB To Serial ports)

用户手册



硬件版本: V1.10 固件版本: V1.31 手册版本: V1.36

河北稳控科技有限公司

2021 年 5 月

概述

可编程 USB 转 UART/I2C/SMBus/SPI/CAN/1-Wire 适配器 USB2S (USB To Serial ports) 是多种数字接口物理层协议转发器, 自带强大灵活的 S2S 协议固件程序, 支持嵌入 C 语言程序开发, 可实现 Windows/Android/Wince 操作系统 USB 接口与串行接口以及串行接口之间的双向通讯, 还可用作脉冲计数、数字示波器、电压比较器。广泛应用于电子设备开发、芯片测试、工业数字接口转换、数字接口学习验证等领域。

功能特点

■ 电源

- **供电:** VIN3.0~10V, 可 USB 接口供电, 即插即用。
- **工作电压:** VWK3.3V 或者 VIN, 跳线切换。
- **电源输出:** VIN、3.3V 及 VWK 供其它设备使用, 最高 500mA 输出能力。
- **功耗:** 无操作自动休眠, 休眠电流低至 5uA。

■ 通讯速率 (非 IO 模拟, 温补晶振, 时钟精准)

- **UART:** 通讯速率 1200~921600bps (1.2kbps~0.9Mbps)。
- **I2C/SMBus:** 通讯速率 0.33kHz~1.8MHz, 可搜索总线上所有从设备。
- **SPI:** 通讯速率 2.75kHz~5.5MHz, 从模式, 可设置时钟极性和相位。
- **1-Wire:** 通用读写, 可连接多种型号 1-Wire 设备 (芯片)。

■ 扩展功能

- 板载 IIC 接口 AT24CxxEEPROM 芯片、SHT 温湿度传感器芯片、16 位差分 AD 采集。
- 板载 SPI 接口 GD25Qxx 芯片, 可用于 SPI 接口测试或者存储用户数据。
- 板载 1-Wire 接口 DS18B20 温度传感器/DS28EC20/2431/2502EEPROM 芯片
- IIC 屏幕专用接口, 连接 OLED12832 屏幕。

■ 外形尺寸

- 30mm*45mm*3mm (PCB 尺寸, 不包含 USB 连接器)

■ 工作温度: -40~85℃

■ 其它

驱动程序: WinXP~Win10、WinCE、Android

全球唯一识别码

公开 PCB 原理图及测试源码

支持 Keil 编程, 在线仿真, 最多 10 个 GPIO, 电压比较器, PWM

内置专用透传协议, 无需开发即可实现复杂外设数据交互

应用领域

■ USB 与串行口物理层透明通讯

■ 串行接口转发器

■ 电子设备固件 ISP 升级

■ EEPROM、FLASH 读写器、数据存储

■ 串行接口芯片、设备测试、Demo 运行器

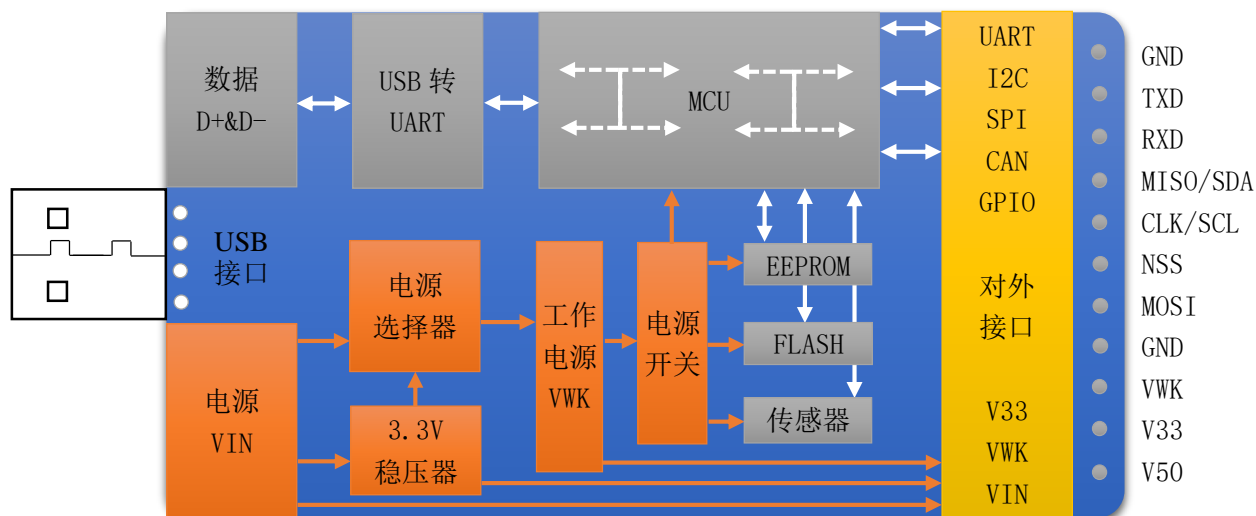
目录

概述	2
功能特点	2
一、结构组成	6
1.1 外形尺寸.....	6
1.2 结构组成.....	6
1.3 接口定义.....	7
二、软硬件资源介绍	8
2.1 电源输入.....	8
2.2 设置工作电压.....	8
2.3 电源输出.....	8
2.4 I2C 总线.....	8
2.5 SPI 总线.....	9
2.6 UART 接口.....	9
2.7 LCD 接口.....	10
2.8 MCU	10
2.9 固件程序.....	10
2.10 扩展资源.....	10
三、通讯协议	11
3.1 自带固件 S2S 功能介绍.....	11
3.1.1 MODBUS 协议保持寄存器汇总.....	11
3.2 S2S 通讯协议.....	11
3.2.1 S2S 协议关键字汇总.....	12
3.2.2 S2S 协议使用举例.....	13
3.2.3 反馈信息说明.....	13
3.3 参数设置.....	14
3.3.1 UART 通讯参数设置.....	14
3.3.2 I2C 通讯参数设置.....	14
3.3.3 SPI 通讯参数设置.....	16
3.3.4 修改系统参数.....	17
3.3.5 保存参数.....	17

3.3.6 参数复位.....	17
四、USB 转 UART 应用	18
4.1 驱动安装.....	18
4.2 USB 转 UART 原理.....	18
五、UART 转 I2C 应用	20
5.1 AT24CxxEEPROM 芯片.....	20
5.1.1 芯片介绍.....	20
5.1.2 向 AT24C02 写入数据.....	20
5.1.3 从 AT24C02 读取数据.....	20
5.1.4 跨页写入数据.....	21
5.2 SHT3x-DIS 温湿度传感器芯片.....	21
5.2.1 芯片介绍.....	21
5.2.2 单次温湿度测量并读取.....	22
5.2.3 连续测量随时读取温湿度.....	22
5.2.4 停止连续测量.....	22
5.2.5 其它常用指令.....	22
5.3 VM5xx 振弦传感器测量模块.....	22
5.3.1 芯片介绍.....	23
5.3.2 修改 VM5xx 寄存器.....	23
5.3.4 保存 VM5xx 寄存器当前值.....	24
5.3.5 读取全部实时寄存器.....	24
六、UART 转 SPI 应用	24
6.1 GD25QxxFLASH 芯片.....	24
6.1.1 芯片介绍.....	24
6.1.2 读取芯片 ID.....	24
6.1.3 芯片写入使能.....	25
6.1.4 擦除首个扇区.....	25
6.1.5 写入数据.....	25
6.1.6 读取数据.....	25
6.2 DS1302 时钟芯片*.....	25
6.2.1 芯片介绍.....	25
4.3.2 使能写操作.....	25

6.2.3 读取时间.....	26
6.2.4 修改时间.....	26
七、UART 转 1-Wire 应用.....	26
7.1DS18B20 温度传感器.....	26
7.2DS2411 序列号芯片.....	26
7.3DH11 温湿度传感器.....	26
7.4DS2438ADC 芯片.....	26
7.5DS2430EEPROM 芯片.....	26
八、专用工具 S2STool 介绍.....	27
8.1S2STool 主界面.....	27
8.2 连接端口.....	27
8.3 参数读取与修改.....	28
8.4 接口类型与芯片选择.....	28
8.5 芯片测试.....	28
九、芯片驱动文件说明.....	29
9.1 芯片驱动结构说明.....	29
9.2 新建芯片驱动.....	30
十、常见问题及注意事项.....	31
文档修改历史.....	31

一、结构组成

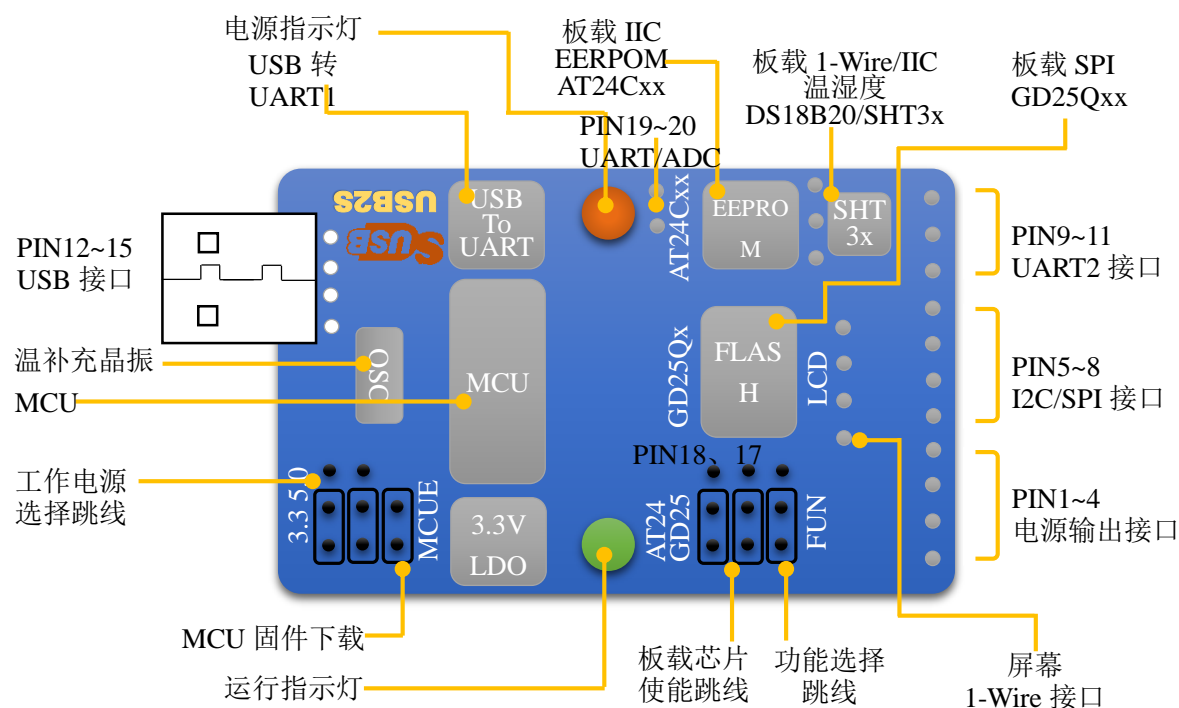


USB2S 功能框图

1.1 外形尺寸

外形尺寸：30*45mm（不含 USB 接口）

1.2 结构组成



1.3 接口定义

序号	名称/标识	功能描述/用法
主接口 (Pin1~Pin11)		
1	V50	电源输出, 3~10V, 使用 USB 供电时为 5.0V
2	V33	3.3V 稳压器输出
3	VCC/VWK	电源选择器输出的主工作电源, 根据跳线设置不同, 电压可为 3.3V 或者 VIN (USB 供电时为 5.0V)
4	GND	电源地
5	MOSI/I03	SPI 串行口主设备输出 (从设备输入)
6	NSS/I04	SPI 串行口从设备片选
7	CLK/SCL	SPI 串行口时钟线/I2C 串行口时钟线(内部已通过 2k 电阻上拉到 VCC)
8	MISO/SDA	SPI 串行口主设备输入 (从设备输出) /I2C 串行口双向数据线 (内部已通过 2k 电阻上拉到 VCC)
9	RXD	UART 接口接收线 (UART2_RXD)
10	TXD	UART 接口发送线 (UART2_TXD)
11	GND	电源地
USB 接口		
12	GND	电源地
13	D+	USB 数据线-
14	D-	USB 数据线+
15	VIN	电源输入, 3~10V, 使用 USB 供电时为 5.0V
板上辅助接口 (16~20)		
16	WP/I01	板载 GD25Qxx 芯片写保护控制管脚, 低电平时写保护
17	HOLD/I02	板载 GD25Qxx 芯片 SPI 通讯使能控制管脚, 低电平时禁止通讯
18		
19	ADC+	差分信号输入正极
20	ADC-	差分信号输入负极

二、软硬件资源介绍

2.1 电源输入

USB2S 使用 USB 接口供电,电压范围为 3~10V,若无特殊说明,以下均使用标准的 USB 接口 5.0V,即 $V_{IN}=5.0V$ 。

2.2 设置工作电压

通过电压选择器跳线可设置 USB2S 的工作电压,如下图所示,跳线帽位于 3.3 一侧时工作电压为 3.3V,跳线帽位于 5.0 侧时工作电压为 V_{IN} (即 USB 供电时的 5.0V)。

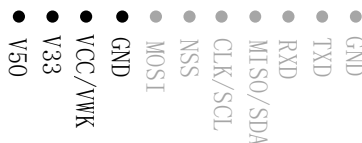
- 设置工作电压时必须两个跳线帽同时调整。
- 本模块片上芯片均支持 3.0~5.5V 工作电压,故此当供电 V_{IN} 为 5.5V 以下时可直接使用 V_{IN} 或者切换为 3.3V,当使用超过 5.5V 的 V_{IN} 为模块供电时,必须将跳线切换至 3.3V 工作电压,否则模块会损毁。



- 除 USB 接口中的 D+和 D-外,其它所有数字接口工作电压均为工作电压 (V_{CC}/V_{WK}),外接设备或芯片时应先用跳线选择工作电压 V_{WK} 为 3.3 或者 5.0,然后再连接设备或芯片。

2.3 电源输出

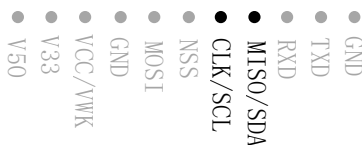
对外接口中的管脚 1~4 为电源输出,可为外接设备供电,4 个管脚依次为: V_{50} 、 V_{33} 、 V_{CC}/V_{WK} 、GND。



- V_{50} : 输出与 V_{IN} 相同的电压,若使用 USB 供电时为 5.0V。
- V_{33} : 使用 V_{IN} 稳压 3.3V 后的输出。
- V_{CC}/V_{WK} : 工作电压输出,电压值由工作电压选择器跳线决定。
- V_{33} 稳压器最大输出电流为 500mA,不得为超过 300mA 的外部设备供电。

2.4 I2C 总线

管脚 7 和 8 分别是 IIC 总线的 SCL 和 SDA 通讯线,板上已有 2k 上拉电阻。
USB2S 的 IIC 接口通讯速率范围为 10kHz~400kHz,支持主模式与从模式。



- 管脚 7 和管脚 8 由 I2S 总线与 SPI 总线共用，同一时间只能使用一种接口，切换方法请详见第 3 章说明。
- IIC 总线上已连接有 EEPROM 芯片和温湿度传感器芯片，可通过跳线设置为强制使能板载 IIC 芯片或者设置为程控使能。



强制为板载 IIC 芯片供电



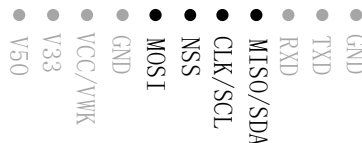
程控板载 IIC 芯片电源

注：第一排的三个针从左向右依次为：VCC、IIC_POW、IO1，当 1 和 2 通过跳线帽短接时，IIC_POW 管脚被强制为 VCC，为板载 IIC 芯片供电，当 2 和 3 通过跳线帽短接时，IIC_POW 被连接到 IO1，受 IO1 控制，通过指令设置 IO1 输出高电平或者低电平来实现是否给板载 IIC 芯片供电的功能。

2.5 SPI 总线

管脚 5、6、7、8 分别是 SPI 总线的 MOSI、NSS、CLK、MISO 线，CLK 和 MISO 已连接有 2k 上拉电阻。

SPI 接口通讯速率最高可达 5MHz，仅支持从机模式，支持 3 线制和 4 线制标准。



- 管脚 7 和管脚 8 由 I2S 总线与 SPI 总线共用，同一时间只能使用一种接口，切换方法请详见第 3 章说明。
- SPI 总线上已连接有 FLASH 芯片，可通过跳线设置为强制使能板载 SPI 芯片或者设置为程控使能。



强制为板载 SPI 芯片供电



程控板载 SPI 芯片电源

注：第二排的三个针从左向右依次为：VCC、SPI_POW、IO2，当 1 和 2 通过跳线帽短接时，SPI_POW 管脚被强制为 VCC，为板载 SPI 芯片供电，当 2 和 3 通过跳线帽短接时，SPI_POW 被连接到 IO2，受 IO2 控制，通过指令设置 IO2 输出高电平或者低电平来实现是否给板载 SPI 芯片供电的功能。

注：IO1 和 IO2 同时也是板载 SPI 芯片 GD25Qxx 的写保护 WP 和通讯保持 HOLD 控制管脚，当操作 GD25Qxx 时，必须使 IO2 为高电平（低电平时此芯片会忽略所有 SPI 信号），当需要向其写数据时，必须使 IO1 为高电平（低电平时仅会执行读取指令，而写入指令会被忽略）。

2.6 UART 接口

本模块有两组 UART 接口，分别叫做 UART1 和 UART2，UART1 是指 USB 与计算机连接后计算机上出现的 COMx 串口，计算机通过此串口与 USB2S 通讯，UART2 是指 USB2S 上的对外接口中的 UART

接口（TXD 和 RXD 管脚）。

管脚 9 和 10 分别是 UART2 异步串行口的 RXD 和 TXD 通讯线，RXD 为弱上拉输入，TXD 为强推挽输出。

UARTx 通讯速率范围为 1200bps~921600bps，支持数据校验和停止位设置。默认通讯速率为 9600bps。



2.7LCD 接口

板上预留有基于 IIC 协议的 LCD 屏幕接口，可选配屏幕（3.3V/5.0V 兼容）。

2.8 差分 ADC 接口

16 位差分芯片 ADS1110 的信号输入正负引脚。

2.9MCU

USB2S 使用了 STC 单片机作为可编程载体，具有片上 2kRAM、4kEEPROM 和 16kFLASH，超高速 1T（1 周期指令）8051 内核，支持休眠和脉冲信号唤醒和定时器唤醒，5 个 16 位定时器，1 路电压比较器，4 路外部中断，硬件 I2S、SPI、UART 接口，支持在线仿真调试。

3.0 固件程序

自带固件程序可完成不同串行接口之间的数据透明传输和预定义通讯协议的数据解析转发，通过修改内部寄存器参数和使用上位机软件工具，可进行几乎所有的串行接口测试应用。

内嵌 MODBUS 通讯协议。

丰富的应用例程，用户可在例程基础上开发自己的固件程序。

2.10 扩展资源

VM501 测试板、常用串行接口芯片测试板、例程

更多扩展资源请访问

<https://shop142244378.taobao.com/?spm=a230r.7195193.1997079397.1.309b5a12nM0J6I>

（淘宝店铺：稳控自动化）

或者我公司官网

<http://www.winkooo.com>

本产品接受用户功能定制，请联系销售人员或者发送邮件到：

INFO@GEO-INS.COM

三、通讯协议

USB2S 支持基于 STC 单片机的二次开发，若有需要，可参照原理图和单片机型号手册自行开发具有特殊功能的固件程序。本手册仅对设备出厂自带固件功能进行说明。

3.1 自带固件 S2S 功能介绍

默认情况下 USB2S 出厂时自带厂家固件程序 S2S，并提供固件程序文件，若有需要可以随时重新下载到 MCU。

S2S 固件不断升级中，请访问我公司官网 www.winkooo.com 下载最新固件程序。

S2S (Serial port To Serial port) 固件可实现任意一个串行口的数据转发到另一个指定的串行口，并自动转换接口硬件层协议，使用的协议为 S2S 串口转发协议。

S2S 固件支持 MODBUS 通讯协议的 03 和 06 指令码，关于 MODBUS 协议内容请自行参阅标准协议手册，或者向我们索要使用说明，MODBUS03 和 06 指令码所操作的保持寄存器定义见下表。

3.1.1 MODBUS 协议保持寄存器汇总

地址	名称	说明	取值范围	默认值	单位
0x01 (1)	ADDR	设备地址	1~255	10	
0x02 (2)	BAUD_U1	UART1 通讯速率	12~9216	96	百 bps
0x03 (3)	BAUD_U2	UART2 通讯速率			
0x04 (4)	BAUD_IIC	IIC 通讯速率码	0~63	52	
0x05 (5)	BAUD_SPI	SPI 通讯速率码	700~5500	700	
0x09 (9)	CLKDIV	系统时钟分频	1~200	1	
0x0C (12)	UART_ECHO	回显示数据	0/1	0	
0x0D (13)	DEBUG_MSG	调试信息	0/1	1	
0x0E (14)	UART12_TT	UART 透传	0/1	1	
0x0F (15)	TGT_PORT	目标端口	0~8	3	
			1:UART1		
			2:UART2		
			3:IIC		
			4:SPI		
			5:		
0x10 (16)	ISCW_XSB	数据位顺序	0:LSB	1	
			1:MSB		
0x11 (17)	SCK_PO	时钟极性	0/1	0	
0x12 (18)	SCK_PH	时钟相位	0/1	0	

3.2 S2S 通讯协议

S2S 通讯协议规则十分简单，说明如下：

➤ 最小收发单位为字符串帧，帧由若干关键字和数据组成。

- 控制关键字以 '[' 起始，以 ']' 结束。连续的数据字符串之间用空格分隔。
- 关键字可单个使用也可以任意多个组合使用。

注意事项：

UART 接口的接收缓存为 254 字节，指令（关键字）数量最多 64 个，数据写入和读出缓存分别为 128 字节，超出部分会被舍弃。

10 进制数据可以用 1~3 个字符表示（每个字符必须为 '0' ~ '9'），16 进制数据必须用 2 个字符表示（每个字符必须为 '0' ~ '9' 或大写 'A' ~ 'F'）。

3.2. 1S2S 协议关键字汇总

关键字	功能描述
目标端口设置关键字	
[HELP]	输出帮助信息
[UART1]	设置目标端口为 UART1，即向 UART1 转发数据，下同
[UART2]	设置目标端口为 UART2
[IIC]	设置目标端口为 IIC
[SPI]	设置目标端口为 SPI
[WIRE]	设置目标端口为 1Wire
读写控制关键字	
[WTnd]	要向目标端口写的字节数，n 为字节数，d 表示后续数据类型，当为 H 时表示后续为 16 进制，当为 D 时表示后续为 10 进制，当为 S 时表示后续为字符串。
[RDna]	要从目标端口读的字节数，n 为要读取的字节数，a 表示接收完最后一个字节后发送应答或非应答信号，A 表示应答，N 表示非应答。
[WRn]	向目标端口写的同时读取数据，专用于 SPI，表示要写/读的字节数
[START]	向目标端口发送开始信号
[STOP]	向目标端口发送停止信号
[DELAYx]	延时，单位为 ms，x 为要延时的毫秒值
[FIND]	搜索总线上所有设备（芯片）并返回地址值
[SSL]	片选输出低电平
[SSH]	片选输出高电平
[IOxH]	指定某 IO 管脚输出高电平
[IOxL]	指定某 IO 管脚输出低电平
参数配置关键字	
[STU1]	修改 UART1 通讯参数
[STU2]	修改 UART2 通讯参数
[STIIC]	修改 IIC 通讯参数
[STSPI]	修改 SPI 通讯参数
[SYSP]	修改系统参数
[SAVE]	保存参数
[STDEF]	恢复参数为默认值
[RESET]	设备重启
提示信息	
[ERRxxx]	通讯过程中发生错误，xxx 为错误码

3.2. 2S2S 协议使用举例

单个关键字

[IIC] 设置数据的目标端口为 IIC 接口

[WT3H]A0 A1 A2 向目标端口（IIC）写 3 个字节 16 进制数据，即：从 IIC 端口输出 3 个字节数据。

[RD05N] 从 IIC 接口读取 5 字节数据，每读取 1 个字节后输出应答信号，最后 1 个字节读取完成后发送非应答信号。

[HELP]或者[HELPxx] 获取帮助信息，xx 为信息类别码(2 位数字)。

连续多个关键字

[IIC][START][WT6H]A0 00 00 01 02 03[STOP] 设置目标端口为 IIC，从 IIC 发出开始信号，从 IIC 输出 6 字节数据，从 IIC 发出停止信号。若之前已经设置过目标端口为 IIC，则上面的指令帧可省略[IIC]，即可以为：[START][WT6H]A0 00 00 01 02 03[STOP]

[STIIC]40 MSB[SAVE][IIC][START][WT6H]A0 00 00 01 02 03[STOP] 先设置 IIC 接口通讯参数，永久保存，再开始一次通讯。

[IIC][FIND]搜索 IIC 总线上所有从设备，并返回从设备地址值。

3.2.3 反馈信息说明

当开启了调试信息输出或者数据回显功能时（默认），每帧关键字指令发送后会输出对指令的解析和执行信息。这些信息仅在关键字指令帧执行不正常时需要查看分析错误原因，正常使用时无需关心，可使用指令关闭这些功能（3.3.4）。

例如：

向设备发送 [IIC][START][WT6H]A0 00 00 01 02 03[STOP]

设备返回 [IIC][START][WT6H]A0 00 00 01 02 03[STOP] //回显接收到的内容
S2SCMD 0003 0901 0106 0902 //关键字指令码
S2SDAT A0 00 00 01 02 03 //指令码数据

解析码（指令码） 与本帧数据中每个关键字指令一一对应，指令码为 16 位，高字节表示指令类别，低字节表示本指令的附加参数。

指令码数据 执行指令码时所需要的数据，如：要发送的数据，要设置的参数值等。

指令码			
高字节-类别码		低字节-参数项	
类别码	类别名称	参数值	参数说明
0x00	设置目标端口	0x01	UART1
		0x02	UART2
		0x03	IIC
		0x04	SPI
		0x05	CAN
		0x06	1-Wire

0x01	向目标端口写数据	0~127	要写的字节数
0x02	从目标端口读数据	0~256	Bit6:0 要读的字节数 bit7=1 表示读取完成后发送非应答信号（高电平），为 0 表示发送应答信号（低电平）
0x03	写读目标端口写的同时读取	0~127	要写的字节数（同时也表示要读的字节数）
0x07	延时	0~255	延时的毫秒值
0x08	参数设置	0x01~0x06	设置指定的端口参数
		0x10	系统参数修改
		0x11	保存所有参数
0x09	从目标端口发出信号	0x01	开始信号
		0x02	停止信号
		0x03	片选信号-NSS 管脚输出低电平
		0x04	非片选信号-NSS 管脚输出高电平
		0x10	I01 输出低电平
		0x11	I01 输出高电平
		0x12	I02 输出低电平
		0x13	I02 输出高电平
		0x14	I03 输出低电平
		0x15	I03 输出高电平

3.3 参数设置

3.3.1 UART 通讯参数设置

UART1 和 UART2 默认通讯参数为 9600, N, 8, 1，可通过以下关键字指令进行修改。

[STU1]B N D S

B 通讯速率，单位 bps，可以为 1200~921600 之间的任一通讯速率

N 校验位，N 表示无校验；A 表示奇校验；E 表示偶校验

D 数据位，可为 8 或者 9

S 停止位，可为 1 或 2

例如：设置 UART1 的通讯参数为 9600, N, 8, 1，则应发送指令[STU1]9600 N 8 1

注：除通讯速率外，其它三个参数固定为 N81，暂时无法修改。

注：执行此指令后，保持寄存器 BAUD_U1 和 BAUD_U2 的值被同步更新，但未保存。

注：修改任意一个 UART 接口时，另外一个 UART 接口也被同时修改，即：两个 UART 的通讯参数总是相同的。

3.3.2 I2C 通讯参数设置

I2C 默认通讯参数为：52, MSB，可以通过以下关键字指令进行修改。

[STIIC]B xSB

B 通讯速率码，可以为 0~63，对应的通讯速率见下表。

xSB 传输数据时高位在前还是低位在前，MSB 表示高位在前，LSB 表示低位在前

注：执行此指令后，保持寄存器 BAUD_IIC、ISCW_XSB 的值被同步更新，但未保存。

注：系统时钟分频系数会影响 IIC 和 SPI 的通讯速率。

下表是时钟分频为 1 时 IIC 取值与实测速率

速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)
0	1866	8	502	16	288.8	32	158
1	1377	9	460.6	18	262.5	36	141.4
2	1104	10	426	20	239.8	40	128.3
3	923.4	11	394.2	22	220.7	44	117.5
4	788.6	12	368	24	204	48	108.4
5	690.6	13	345.4	26	189.3	52	100.3
6	614.6	14	325	28	177.9	57	92
7	551.6	15	306.1	30	166.8	63	83.5

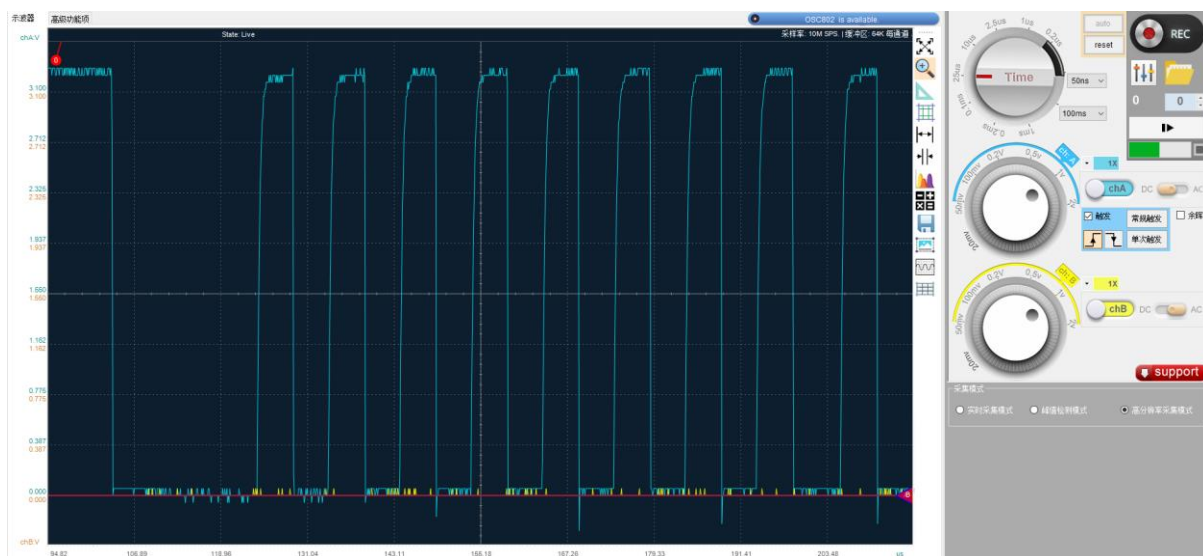
注：IIC 通讯速率受上拉电阻限制，此表为上拉电阻为 2k 时实测，当使用更小阻值的上拉电阻时，实际通讯速率会有所提高，反之会有所下降。下同。

下表是时钟分频为 4 时 IIC 取值与实测速率

速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)
0	687.8	8	137.8	16	76.7	32	40.9
1	459.5	9	125.3	18	69.8	36	36.5
2	344.4	10	115.0	20	62.8	40	32.9
3	306.6	11	106.0	22	57.7	44	30.2
4	230.1	12	98.5	24	53.6	48	27.6
5	197.2	13	91.9	26	49.3	52	25.8
6	172.6	14	86.2	28	46.2	57	23.5
7	153.4	15	81.2	30	43.3	63	21.3

下表是时钟分频为 8 时 IIC 取值与实测速率

速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)	速率码	通讯速率 (kHz)
0	394.1	8	70.8	16	38.9	32	
1	251.2	9		18	34.9	36	18.3
2	184.3	10	58.7	20		40	
3	145.3	11		22	29.1	44	15.1
4	120.2	12	50.3	24		48	
5	102.3	13		26	24.8	52	12.8
6	89.0	14	43.8	28		57	
7	78.9	15		30	21.7	63	10.67



实测 100kHz 时钟信号

3.3.3 SPI 通讯参数设置

SPI 默认通讯参数为：700kHz, MSB, L, B, 可通过以下关键字指令进行修改。

[STSPI]baud xSB po ph

Baud 通讯速率，单位 kHz，可以为 5500、2700、1400、700

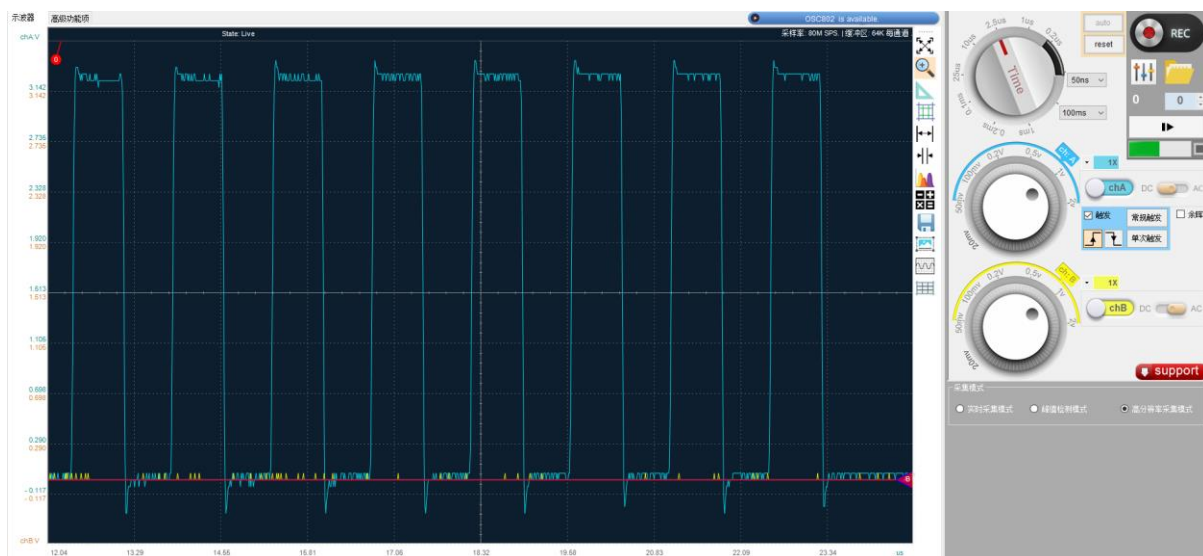
xSB 传输数据时高位在前还是低位在前，MSB 表示高位在前，LSB 表示低位在前

Po 时钟极性，H 表示空闲时 SCK 为高电平，L 表示空闲时 SCK 为低电平

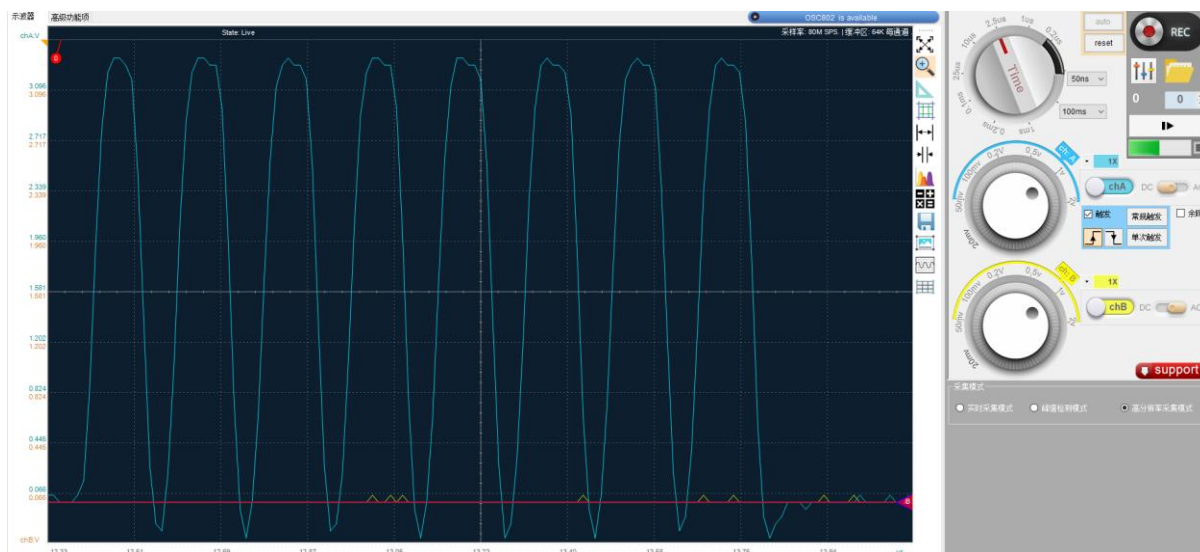
Ph 时钟相位，F 表示在前时钟沿输出数据，B 表示在后时钟沿输出数据

注：执行此指令后，保持寄存器 BAUD_SPI、ISCW_XSB 的值被同步更新，但未保存。

注：系统时钟分频系数会影响 IIC 和 SPI 的通讯速率。



通讯速率 700kHz 实测≈691kHz

通讯速率 5500kHz 实测 \approx 5543kHz

下表是不同的时钟分频时 SPI 速率取值与实测速率

速率取值	通讯速率 (kHz)					
	1 分频	2 分频	4 分频	8 分频	10 分频	20 分频
700	695.0	345.2	172.6	86.3	68.9	
1400	1377.4	689.0	345.4	172.9	138.1	
2700	2762.4	1381.2	694.2	344.8	275.6	
5500	5568.0	2751.8	1372.5	690.8	548.2	

3.3.4 修改系统参数

使用[SYSP]关键字可对其它系统参数进行修改。指令格式如下：

[SYSP]Addr ECHO DEBUG_MSG UART12_TT CLKDIV

Addr 设备地址

ECHO 是否回显接收到的数据

DEBUG_MSG 是否输出调试信息

UART12_TT UART1 和 UART2 之间是否透明传输

CLKDIV 系统时钟分频系数

例如：[SYSP]12 1 0 1 1 表示将设备地址修改为 12，回显接收到的数据，不输出调试信息，两个 UART 口透明传输，系统时钟不分频。

3.3.5 保存参数

前述各种指令在参数修改完成后仅可当时生效，修改后的参数断电不保存。使用[SAVE]关键字可将当前参数保存至 EEPROM，使参数永久保存。

3.3.6 参数复位

参数复位有两种方法，通过 UART 接口向模块发送[STDEF]指令，或者给模块上电前，将 FUN 跳线帽连接至 GND，如下图。参数复位后，请将跳线帽切换至第 2 和 3 脚，防止重复复位。



注：第三排针从左向右依次为 GND、FUN、空，使用跳线帽将 1 和 2 连接，强制将 FUN 拉低，在上电时若检测到 FUN 的低电平会复位系统参数，使用跳线帽将 2 和 3 连接，使 FUN 管脚悬空（内部弱上拉为高电平）。

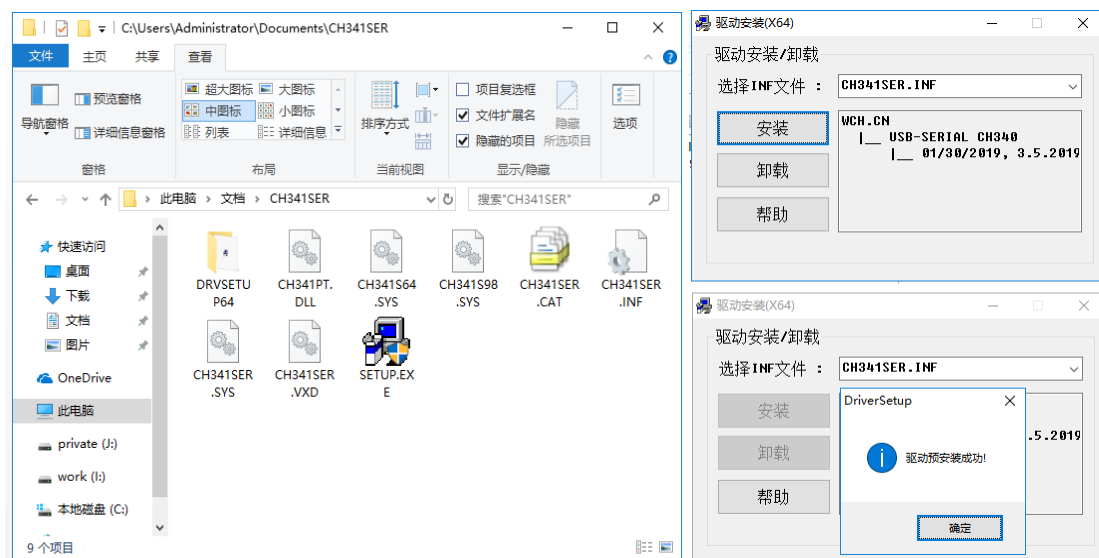
四、USB 转 UART 应用

4.1 驱动安装

USB2S 内置了 USB 转 UART 芯片，可使用 CH340/CH341 驱动程序。驱动安装步骤如下：

双击运行“CH341SER\SETUP.exe”，打开驱动安装窗口。

点击【安装】按钮，稍候会提示“驱动安装成功”。如下图所示。

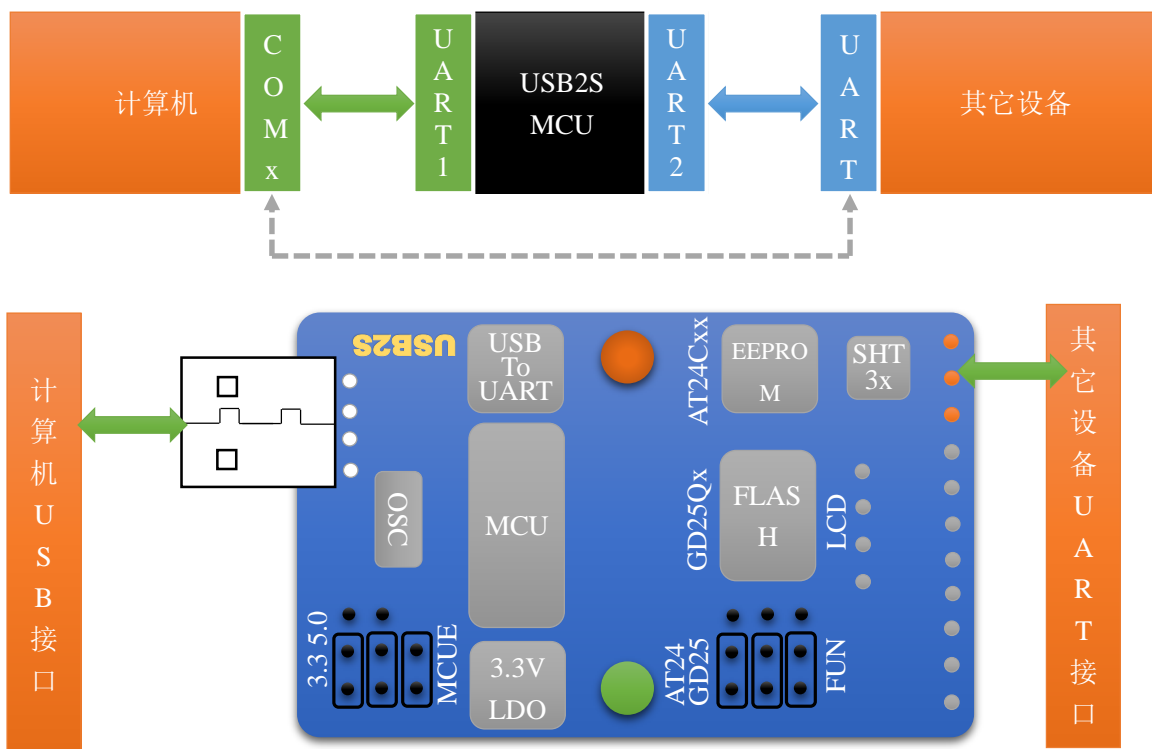


驱动安装成功后，若将 USB2S 插入计算机 USB 接口，在“设备管理器”中会发现新的 COM 接口，如下图所示。



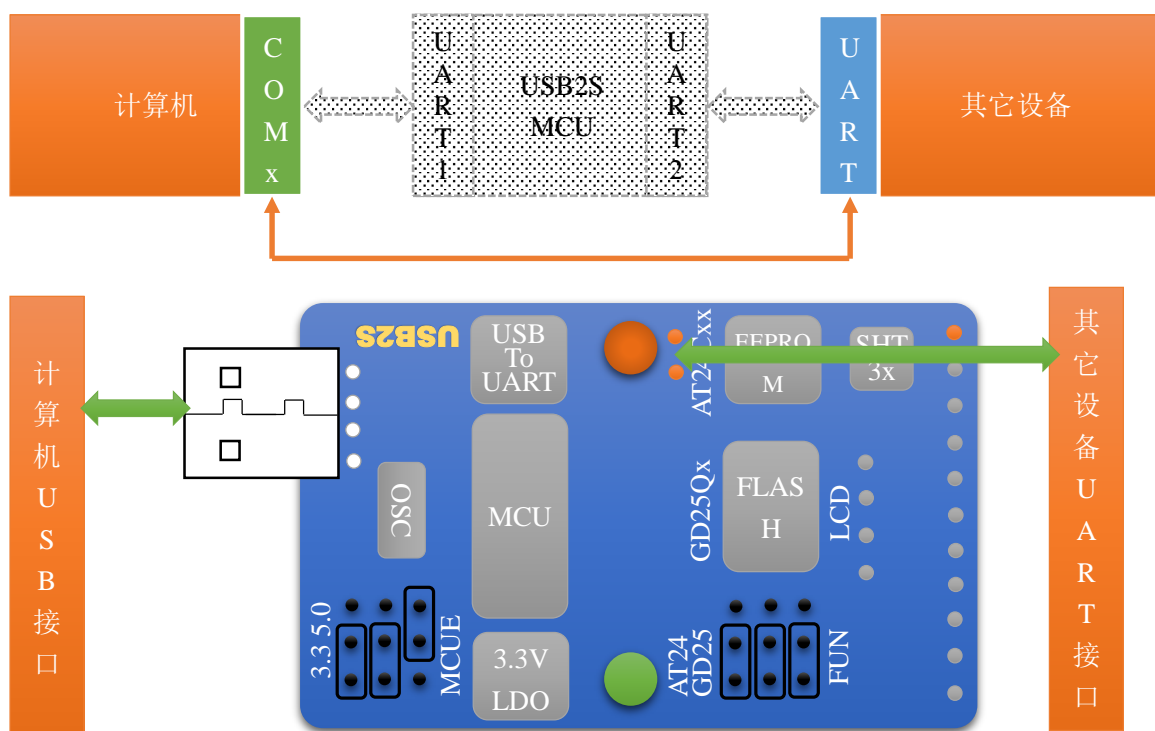
4.2 USB 转 UART 原理

驱动程序安装后，计算机通过 COMx 与 MCU 进行通讯，当 USB2S 的 UART 透明传输功能为开启状态时（默认），MCU 可将 UART1 与 UART2 的双向数据进行透明转发，即：实现了计算机的 COMx 端口与 USB2S 的对外 UART2 端口的双向数据传输。工作原理示意图如下：



上面的方法是通过 MCU 固件程序实现 UART1 与 UART2 之间数据透明传输。

下面的连接方法可实现真正的 USB 转 UART 功能，方法是通过跳线禁用 MCU，外部 UART 设备直接连接到 UART1 接口。



对于绝大部分应用，第一种连接方法即可。

五、UART 转 I2C 应用

5.1 AT24CxxEEPROM 芯片

5.1.1 芯片介绍

AT24Cxx 是可擦写 EEPROM 存储芯片，xx 表示容量，单位为 Kbits。USB2S 板上已有 1 片 AT24Cxx（默认为 AT24C02），设备地址为 0xA0。

通过 AT24Cxx 的数据手册可知，读写时序如下：

写时序：

- (1) 向 IIC 总线发送开始信号
- (2) 向 IIC 总线发送芯片写地址 (0xA0)
- (3) 向 IIC 总线发送要操作的 EEPROM 地址
- (4) 向 IIC 总线连续发送多个要写入的字节数据
- (5) 向 IIC 总线发送停止信号，结束本次通讯

读时序：

- (1) 向 IIC 总线发送开始信号
- (2) 向 IIC 总线发送芯片写地址 (0xA0)
- (3) 向 IIC 总线发送要操作的 EEPROM 地址
- (4) 向 IIC 总线发送开始信号
- (5) 向 IIC 总线发送芯片读地址 (0xA1)
- (6) 从 IIC 总线连续读取芯片输出的数据
- (7) 向 IIC 总线发送停止信号，结束本次通讯

5.1.2 向 AT24C02 写入数据

向 AT24C02 的地址 0~3 写入 4 个字节 (30 31 32 33)

向 UART1 发送字符串：[IIC][START][WT6H]A0 00 30 31 32 33[STOP]

[IIC] 数据帧的目标是 IIC 接口

[START] 从目标接口输出开始信号

[WT6H] 从目标接口输出 6 个字节，后续数据类型为 16 进制。

A0 00 2 字节数据，AT24C02 芯片地址+要操作的 EEPROM 地址

00 01 02 03 向 AT24C02 写入的 4 个字节数据

[STOP] 从目标接口输出停止信号，结束本次通讯

5.1.3 从 AT24C02 读取数据

从 AT24C02 读取 4 个字节数据，数据起始地址为 0。

向 UART1 发送字符串：[IIC][START][WT2H]A0 00[START][WT1H]A1[RD4N][STOP]

从 UART1 返回 4 个字节 (16 进制)：30 31 32 33

[IIC] 数据帧的目标是 IIC 接口

[START] 从目标接口输出开始信号

[WT2H] 从目标接口输出 2 个字节，后续数据类型为 16 进制。

A0 00 2 字节数据，AT24C02 芯片地址+要操作的 EEPROM 地址
 [START] 从目标接口输出开始信号
 [WT1H] 从目标接口输出 1 个字节，后续数据类型为 16 进制。
 A1 AT24C02 芯片的读地址
 [RD4N] 从目标接口连续读取 4 个字节数据
 [STOP] 从目标接口输出停止信号，结束本次通讯

5.1.4 跨页写入数据

AT24C02 每页为 8 字节，EEPROM 地址 0~7 为第 1 页、8~15 为第 2 页，以此类推。当向芯片连续写入数据时，在写完每页最后 1 个字节后必须向其发送停止信号，以使芯片对当前页进行存储，继续写入后续数据时需要重新开始写入时序。

芯片存储数据需要一定时间，向其发起新的通讯前应有适当延时。也可以直接向其继续写入，若返回了[ERRxxx]表示芯片正忙，重新发送直到无错误返回。

例如：从 EEPROM 地址 0 开始写入 10 个字节数据，则应先写入 0~7，延时，再写入 8~9。

[IIC][START][WT10H]A0 00 00 01 02 03 04 05 06 07[STOP][DELAY50][IIC][START][WT4H]A0 08 08 09[STOP]

例如：直接向芯片写入字符串数据

[IIC][START][WT2H]A0 00[WT8S]87654321[STOP][DELAY50][IIC][START][WT4H]A0 08 38 39[STOP]

5.2 SHT3x-DIS 温湿度传感器芯片

5.2.1 芯片介绍

SHT3x-DIS 是 IIC 接口的温度、湿度传感器芯片，可工作于单次测量或连续自动测量模式。USB2S 已有 1 片 SHT31-DIS 芯片，芯片地址为 0x88。

SHT3x-DIS 的输出温度和湿度均为 3 字节，前两字节是温湿度值，第 3 字节是校验字节。转换公式如下：（数值=第 1 个字节*256+第 2 个字节）。

$$\text{温度} = \left(\frac{\text{数值} \times 175}{65535} - 45 \right) ^\circ\text{C} \quad \text{湿度} = \left(\frac{\text{数值} \times 100}{65535} \right) \%$$

写指令时序：

- (1) 向 IIC 总线发送开始信号
- (2) 向 IIC 总线发送芯片写地址（0x88）
- (3) 向 IIC 总线发送 2 字节命令字
- (4) 发送停止信号，结束本次通讯。

读取温湿度时序：

- (1) 向 IIC 总线发送开始信号
- (2) 向 IIC 总线发送芯片写地址（0x88）

- (3) 向 IIC 总线发送 2 字节命令字，延时等待芯片完成测量（工作于单次测量模式时）
- (4) 向 IIC 总线发送开始信号
- (5) 向 IIC 总线发送芯片读地址（0x89）
- (6) 从 IIC 总线读取 2 字节温度和 1 字节 CRC8 校验
- (7) 从 IIC 总线读取 2 字节湿度和 1 字节 CRC8 校验
- (8) 发送停止信号，结束本次通讯。

5.2.2 单次温湿度测量并读取

[IIC][START][WT3H]88 24 00[DELAY100][START][WT1H]89[RD6A][STOP]，或者
[IIC][START][WT3D]136 36 00[DELAY100][START][WT1H]89[RD6A][STOP]
[IIC] 数据帧的目标是 IIC 接口
[START] 从目标接口输出开始信号
[WT3H] 从目标接口输出 3 个字节，后续数据类型为 16 进制。
88 24 00 芯片写地址+2 字节指令码 0x2400 表示高速测量并且禁用芯片控制时钟线的功能
[DELAY100] 延时等待温湿度测量结束
[START] 从目标接口输出开始信号
[WT1H] 从目标接口输出 1 个字节，后续数据类型为 16 进制。
89 芯片的读地址
[RD6N] 从目标接口连续读取 2 个寄存器（每个寄存器 2 字节，共 4 字节）
[STOP] 从目标接口输出停止信号，结束本次通讯
芯片返回 64 B3 BB 3E E3 CC (25779 16099，即：23.8℃和 24.5%)

5.2.3 连续测量随时读取温湿度

切换为连续测量工作模式：[IIC][START][WT3H]88 20 2F[STOP]
随时获取实时温湿度：[IIC][START][WT3H]88 E0 00[START][WT1H]89[RD6N][STOP]

5.2.4 停止连续测量

向芯片发送指令码 0x3093
[IIC][START][WT3H]88 30 93[STOP]

5.2.5 其它常用指令

使用指令复位芯片
向芯片发送指令码 0x30A2
[IIC][START][WT3H]88 30 A2[STOP]

加热器的开启与关闭

发送指令码 0x306D 开启加热器，发送指令码 0x3066 关闭加热器。
[IIC][START][WT3H]88 30 6D[STOP]
[IIC][START][WT3H]88 30 66[STOP]

读取状态寄存器

[IIC][START][WT3H]88 F3 2D[START][WT1H]89[RD3N][STOP]

5.3 VM5xx 振弦传感器测量模块

5.3.1 芯片介绍

VM501/511 是稳控科技生产的振弦传感器测量模块，具有 IIC 接口和 UART 接口。

IIC 地址可通过 UART 任意设置，假设其地址为 0xB0。

VM5xx 的 IIC 通讯协议如下：

写寄存器：

- (1) 向 IIC 总线发送开始信号
- (2) 向 IIC 总线发送模块写地址 (0xB0)
- (3) 向 IIC 总线发送要操作的寄存器地址
- (4) 向 IIC 总线连续发送多个要写入的寄存器值，每个寄存器需要 2 字节
- (5) 向 IIC 总线发送停止信号，结束本次通讯

读寄存器：

- (1) 向 IIC 总线发送开始信号
- (2) 向 IIC 总线发送模块写地址 (0xB0)
- (3) 向 IIC 总线发送要操作的寄存器地址
- (4) 向 IIC 总线发送开始信号
- (5) 向 IIC 总线发送模块读地址 (0xB1)
- (6) 从 IIC 总线连续读取芯片输出的数据（每 2 个字节代表 1 个寄存器值）
- (7) 向 IIC 总线发送停止信号，结束本次通讯

5.3.2 修改 VM5xx 寄存器

向 VM5xx 的寄存器 0~1 写入新值 (0x0001 0x0060)

向 UART1 发送字符串：[IIC][START][WT6H]B0 00 00 01 00 60[STOP]

[IIC] 数据帧的目标是 IIC 接口

[START] 从目标接口输出开始信号

[WT6H] 从目标接口输出 6 个字节，后续数据类型为 16 进制。

A0 00 2 字节数据，VM5xx 模块地址+要操作的寄存器地址

00 01 00 60 向 VM5xx 模块写入的 2 个寄存器值（每个寄存器 2 字节，共 4 个字节）

[STOP] 从目标接口输出停止信号，结束本次通讯

5.3.3 读取 VM5xx 寄存器

从 VM5xx 读取 2 个寄存器数据，寄存器起始地址为 0。

向 UART1 发送字符串：[IIC][START][WT2H]B0 00[START][WT1H]B1[RD4N][STOP]

从 UART1 返回 4 个字节（16 进制）：00 01 00 60

[IIC] 数据帧的目标是 IIC 接口

[START] 从目标接口输出开始信号

[WT2H] 从目标接口输出 2 个字节，后续数据类型为 16 进制。

A0 00 2 字节数据，VM5xx 模块写地址+要操作的寄存器地址

[START] 从目标接口输出开始信号

[WT1H] 从目标接口输出 1 个字节，后续数据类型为 16 进制。

A1 VM5xx 模块的读地址

[RD4N] 从目标接口连续读取 2 个寄存器（每个寄存器 2 字节，共 4 字节）

[STOP] 从目标接口输出停止信号，结束本次通讯

5.3.4 保存 VM5xx 寄存器当前值

当通过 IIC 接口修改 VM5xx 单个寄存器后，被修改的寄存器立即保存（断电不丢失），但连续寄存器的写入仅当时修改生效，模块重启后会自动恢复。为了能够使寄存器永久保存，可以单独向功能寄存器 03 写入指令码 0x000C 来强制保存所有寄存器。

[IIC][START][WT4H]B0 03 00 0C[STOP]

5.3.5 读取全部实时寄存器

VM5xx 测量值和运行状态全部位于寄存器 32~64，这些寄存器由模块自动更新，可以被部分读取或者全部读取。

[IIC][START][WT2H]B0 20[START][WT1H]B1[RD60N][STOP]

从模块的寄存器 0x20(32)开始连续读取 60 个字节（30 个寄存器）。

六、UART 转 SPI 应用

6.1GD25QxxFLASH 芯片

注意：1-Wire 接口与 SPI 接口不可同时连接被测芯片。

6.1.1 芯片介绍

GD25Qxx 是四线 SPI 接口的 FLASH 芯片，最大容量可达 16Mbytes。板上集成有 GD25Q64 芯片，每页 256 字节，每扇区 16 页（4k 字节），每块区 256 页（64k），写入前必须先擦除，本芯片支持扇区擦除、块区擦除和整片擦除。

读时序：

- （1）向芯片发送 1 字节指令码 0x03
- （2）向芯片发送要读取的起始地址 3 字节 0xaaaaaa
- （3）连续读取 n 字节数据

写时序：

- （1）向芯片发送 1 字节指令码 0x02
- （2）向芯片发送要写入的起始地址 3 字节 0xaaaaaa
- （3）连续写入 n 字节数据（不超过 256）

擦除时序：

擦除扇区：

- （1）向芯片发送 1 字节指令码 0x20
- （2）向芯片发送要擦除扇区的起始地址 3 字节 0xaaaaaa（4096 的整倍数）

擦除区块：

- （1）向芯片发送 1 字节指令码 0xD8
- （2）向芯片发送要擦除区块的起始地址 3 字节 0xaaaaaa（65536 的整倍数）

整片擦除：

- （1）向芯片发送 1 字节指令码 0xC7

6.1.2 读取芯片 ID

[SPI][SSL][WT4H]90 FF FF FF[RD2N][SSH]

返回：C8 16

C8：芯片生产厂商标识码，GigaDevice（兆易创新）

16：器件容量标识码，16 表示 64Mbit（8MBytes）

6.1.3 芯片写入使能

擦除或者写入之前需要保证芯片为写使能状态，如果已设置过写使能，则此步骤或省略。

[SPI][SSL][WT1H]06[SSH]

6.1.4 擦除首个扇区

首个扇区的地址为 0~4095，此扇区首地址为 0。

[SPI][SSL][WT4H]20 00 00 00[SSH]

注：GD25Qxx 芯片有写保护功能，在写操作前最好先发送 06 指令码，使能擦写功能。

[SPI][SSL][WT1H]06[SSH][SPI][SSL][WT4H]20 00 00 00[SSH]

6.1.5 写入数据

从首地址开始写入 10 个字节：0x30~0x39

[SPI][SSL][WT14H]02 00 00 00 30 31 32 33 34 35 36 37 38 39[SSH]

写入之前禁用写保护

[SPI][SSL][WT1H]06[SSH][SPI][SSL][WT14H]02 00 00 00 30 31 32 33 34 35 36 37 38 39[SSH]

6.1.6 读取数据

从首地址开始读取 20 个字节

[SPI][SSL][WT4H]03 00 00 00[RD20N][SSH]

6.2DS1302 时钟芯片*

测试外接 SPI 芯片时，要防止板载 SPI 芯片 GD25Qxx 的影响，有以下两种方法使 GD25Qxx 失效。

方法 1：将 GD25Q 跳线帽拔掉（断开 GD25Qxx 芯片的电源）。

方法 2：设置 GD25Qxx 芯片的 HOLD 管脚为低电平（指令为 IO2L），使 GD25Qxx 芯片的 SPI 通讯管脚失效。

推荐方法 2，即在测试外接 SPI 芯片时，在指令中添加一个[IO2L]指令码，此码仅是为了禁用板载 GD25Qxx 芯片，并不是针对外接芯片的指令。

6.2.1 芯片介绍

DS1302 是实时时钟芯片，SPI 接口，可以对年、月、日、周、时、分、秒进行计时，且具有闰年补偿等多种功能。DS1302 内部有一个 31×8 的用于临时性存放数据的 RAM 寄存器。

DS1302 使用 1 个字节的控制字+读写时序来完成数据交互，控制字节的最高有效位(位 7)必须是逻辑 1，如果它为 0，则不能把数据写入 DS1302 中，位 6 如果为 0，则表示存取日历时钟数据，为 1 表示存取 RAM 数据；位 5 至位 1 指示操作单元的地址；最低有效位(位 0)如为 0 表示要进行写操作，为 1 表示进行读操作，控制字节总是从最低位开始输出。

DS1302 不是标准的四线 SPI，它的发送和接收统一使用 IO 管脚，并且它没有 SPI 协议中的使能控制管脚。另外它有读写使能的 CE 控制管脚也是 SPI 协议中没有的。

4.3.2 使能写操作

[SPI][SSH][WT2H]8E 00[SSL]

上例中，借用了 SPI 接口中的 SS 管脚来实现对 DS1302CE 管脚的控制时序。下同。

6.2.3 读取时间

读取年 [SPI][SSH][WT1H]8D[RD1][SSL]

读取周 [SPI][SSH][WT1H]8B[RD1][SSL]

读取月 [SPI][SSH][WT1H]89[RD1][SSL]

读取日 [SPI][SSH][WT1H]87[RD1][SSL]

读取时 [SPI][SSH][WT1H]85[RD1N][SSL]

读取分 [SPI][SSH][WT1H]83[RD1N][SSL]

读取秒 [SPI][SSH][WT1H]81[RD1N][SSL]

也可以一次连续读取：[SPI][SSH][WT1H]81[RD7N][SSL]

6.2.4 修改时间

修改年 [SPI][SSH][WT2H]8C 18[SSL]

修改周 [SPI][SSH][WT2H]8A 02[SSL]

修改月 [SPI][SSH][WT2H]88 05[SSL]

修改日 [SPI][SSH][WT2H]86 15[SSL]

修改时 [SPI][SSH][WT2H]84 08[SSL]

修改分 [SPI][SSH][WT2H]82 30[SSL]

修改秒 [SPI][SSH][WT2H]80 00[SSL]

以上实现了修改日期时间为：2018-05-15 08:30:00（星期二），下面的指令可一次性写入

[SPI][SSH][WT8H]80 00 30 08 15 05 02 18[SSL]

七、UART 转 1-Wire 应用

注意：1-Wire 接口与 SPI 接口不可同时连接被测芯片。

7.1DS18B20 温度传感器

请使用 S2STool 进行测试，本手册不再一一描述。下同。

7.2DS2411 序列号芯片

7.3DH11 温湿度传感器

7.4DS2438ADC 芯片

7.5DS2430EEPROM 芯片

八、专用工具 S2STool 介绍

S2STool 是为 S2S 固件开发的测试工具，运行于 Windows 平台，内置串口调试助手和 S2S 参数配置工具，支持动态解析自定义的芯片驱动，用户可自行编写、修改、增加任何芯片的驱动程序，方便对芯片进行测试和评估。

8.1 S2STool 主界面

双击 S2STool.exe 启动程序，主界面如下图所示。



S2STool 主界面及组成

- **标题栏：**显示工具名称和版本信息，S2STool 工具与 S2S 固件一一对应，请根据 S2S 固件版本来选择适用的 S2STool 工具版本。
- **状态栏：**显示工具的运行状态。
- **计算机端口区：**选择 USB2S 与计算机连接的 COM 端口，设置 COM 端口通讯参数。
- **S2S 参数：**S2S 固件的参数读取和修改操作区。
- **接口与芯片选择区：**选择外接芯片（或模块、设备）的接口类型和芯片型号名称。
- **芯片操作区：**对选择的具体芯片的所有预编程测试功能按钮区。
- **串口助手：**通用串口助手，可收发字符串或者 16 进制数据。

8.2 连接端口

连接 COM 端口的目的是建立计算机与 USB2S 的通讯链路，以完成参数读写、芯片测试、串口收发等后续工作。

在计算机端口区选择与 USB2S 连接的 COM 端口名称（见“4.1 驱动安装”），选择与 USB2S 一致的通讯速率（默认为 9600bps），点击【连接端口】按钮，若连接成功，则连接状态图标由黑色变为红色，状态栏提示信息为“COMx 已连接”。



8.3 参数读取与修改

8.3.1 参数读取

端口连接成功后，点击 S2S 参数区的【读取模块参数】按钮，稍候（约 1~2 秒）即可完成 S2S 模块的参数读取，状态栏提示“收到参数寄存器数据，CRC 校验正确”，参数更新显示于参数区的各个控件。

8.3.2 参数修改

在 S2S 参数区内修改参数，点击【写入模块参数】按钮，即可将界面显示的参数更新到模块，状态栏显示“参数修改成功”。

8.4 接口类型与芯片选择

在接口与芯片选择区，点选接口类型为 UART、I2C、SPI、1-Wire 之一，程序自动检索已存在的所选接口的驱动文件，并将备选芯片名称加载到【芯片名称下拉框】，在下拉框内选择需要的芯片即可。芯片选择后，芯片名称下方自动显示此芯片的功能描述，此时点击右侧的【芯片介绍】按钮可打开芯片介绍窗口，点击【数据手册】按钮可打开此芯片的数据手册文件。

芯片选择后，在芯片操作区会自动加载此芯片的功能操作按钮和数据项目表格，如图示。

芯片名称下拉框中出现的每一个备选芯片均对应一个驱动文件，用户也可根据驱动文件编写规则自行编写新的芯片文件，也可修改已有芯片的驱动文件内容，详见“第九章：芯片驱动文件说明”。



8.5 芯片测试

在芯片操作区，列出和多个关于所选芯片相关的操作按钮，点击对应按钮即可完成芯片的读写操作。

点击某个按钮时，主界面右侧的串口助手发送区显示此按钮对应的发送的指令内容，接收区显示了芯片对指令的响应过程和返回数据，在芯片的驱动文件内，对芯片的返回数据做了部分运算定义，可自动根据预定公式完成计算，并将运算结果显示于操作按钮右侧的表格内。



通过上图可看出，点击按钮后自动从发送区发送的内容实际上就是根据 S2S 通讯协议和当前所选芯片数据手册所生成的 S2S 关键字指令码（详见前述“第三章：通讯协议”和“4.2SHT3x 温湿度传感器芯片”），用户可根据需要，自行编辑修改发送区的内容，然后点击【发送】按钮实现自定义指令发送。

九、芯片驱动文件说明

芯片选择下拉框内列出的每一个备选芯片的驱动文件和数据手册位于 S2STool 工具文件夹中的 chips 文件夹内，每一个芯片需要 1 个驱动文件和 1 个数据手册文件，用户可自行打开编辑，或者创建新的芯片驱动文件。

芯片驱动文件为 xml 格式，需要手工修改或者创建，芯片数据手册一般来自于网络，下载后复制到 chips 文件夹内即可。

Xml 文件为文本文件，建议使用“Notepad++”编辑器。

9.1 芯片驱动结构说明

下图所示为芯片驱动的模板文件“template.xml”。

```

<?xml version="1.0" encoding="GB2312" ?>
<CHIP>
  <AUTHOR company="作者单位名称" authorName="作者名字"/>
  <DESC desc="芯片的简要功能描述（一句话）"/>
  <DATASHEET fileName="数据手册文件名，如：AT24C02.pdf"/>
  <INTRODUCTION title="XXXXX芯片介绍">
    <SECTION01 body="XXXXX芯片是xxx公司生产的xxxxxx芯片，工作电压x.x~x.xV，xxx封装，xxx总线。"/>
    <SECTION01 body="特点及功能说明，若干行，可增加新行或者删除已有行"/>
    <SECTION01 body="这是一行文字，请修改为具体内容或者删除本行"/>
    <SECTION01 body="....."/>
    <SECTION01 body="....."/>
    <SECTION01 body="1.1XXXXXXX"/>
    <SECTION01 body="....."/>
    <SECTION01 body="....."/>
    <SECTION01 body="1.2XXXXXXX"/>
    <SECTION01 body="XXXX"/>
    <SECTION01 body="(1)....."/>
    <SECTION01 body="(2)....."/>
    <SECTION01 body="(3)....."/>
    <SECTION01 body="XXXX"/>
    <SECTION01 body="(1)....."/>
    <SECTION01 body="(2)....."/>
    <SECTION01 body="(3)....."/>
    <SECTION01 body="1.3XXXXXXX"/>
    <SECTION01 body="....."/>
    <SECTION01 body="....."/>
    <SECTION01 body=""/>
  </INTRODUCTION>
  <COMMANDS>
    <COMMAND01 caption="指令名称，对应主界面上的一个指令按钮，下同" hint="鼠标停留在此按钮上的提示信息" cmdStr="指令内容">
      <RETURN0101 caption="表格显示文字项，若此指令无返回数据可删除本行" startBit="-1" endBit="-1" formula="计算公式，dat[0]表示返回数据中的第1个字节，以此类推" unitStr="℃"/>
    </COMMAND01>
    <COMMAND01 caption="指令名称" hint="XXXXXXX" cmdStr="XXXXXXX">
    </COMMAND01>
    <COMMAND01 caption="指令名称" hint="向芯片发送指令码0x0000获取连续转换完成的温度值" cmdStr="[IIC][START][W73H]88 00 [START][W73H]89 [RD6N][STOP]">
      <RETURN0101 caption="温度" startBit="-1" endBit="-1" formula="(dat[0]*256+dat[1])*175/65535-45" unitStr="℃"/>
      <RETURN0102 caption="湿度" startBit="-1" endBit="-1" formula="(dat[3]*256+dat[4])*100/65535" unitStr="%">
    </COMMAND01>
  </COMMANDS>
</CHIP>

```

芯片驱动文件的结构较为简单，分为芯片说明、描述部分和测试指令部分，芯片描述部分包括作者信息、芯片说明、数据手册名称等，按照实际情况修改即可，下面重点说明测试指令部分。

在驱动文件中的 COMMANDS 节点内，每一对尖括号节点段（< />，下文称“指令节点”）称为一个命令，每个命令对应主界面上芯片操作区的一个按钮，按钮由程序根据驱动文件动态生成。

指令中的属性

- **caption 属性：**显示于按钮上的文字内容
- **hint 属性：**此指令的提示信息（当鼠标移动到此按钮上时会显示此提示）
- **cmdStr 属性：**此指令的具体指令内容（符合 S2S 协议的字符串指令），支持用“|”分隔的多条字符串指令，例如：指令字符串 1|DELAY100|指令字符串 2|DELAY100|.....

指令的子节点-返回数据计算项

对于有返回数据的指令，若希望对返回数据进行一些运算并将运算结果显示于主界面的数据表格中，则应为此指令添加若干子节点，即“返回数据计算项”，返回数据计算项的数量（行数）没有任何限制，可以没有，也可以任意多个。

数据计算项的属性

- **caption 属性：**显示于数据表格中的第一列，即：名称列。
- **formula 属性：**此数据项的计算公式，计算公式的运算结果会更新到数据表格的第二列，即：数值列。
- **startBit 和 endBit 属性：**若是“-1”则表示计算公式结果直接显示于界面表格中，否则将指定的某些位（bit）截取后显示于表格中。要取出的二进制数据可以指定开始位值和结束位值，分别用 startBit 和 endBit 来定义。

关于计算公式

计算公式为一段字符串，支持常用的四则运算符号和初等数学函数、逻辑函数等。在公式中，可使用“dat[0]”表示执行当前指令后芯片返回的首个字节，“dat[1]”表示返回的第 2 个字节，以此类推。

9.2 新建芯片驱动

（1）创建芯片驱动文件

打开“template.xml”文件，根据芯片接口类型，将文件另存到 chips 文件夹内的 I2S、SPI 或

者 Wire 文件夹内，文件名称必须使用芯片名称，方便在主界面芯片备选框中识别此芯片。

(2) 添加数据手册

将下载的数据手册文件复制到与上述芯片驱动文件相同的文件夹内。

(3) 修改芯片驱动内容

按照 9.1 中的驱动文件结构和各部分含义、功能和注意事项，对照芯片数据手册完成驱动文件内容的修改，最后保存驱动文件。

(4) 运行程序并验证驱动文件

将芯片连接到 USB2S 的对外管脚上，运行 S2STool 工具，选择刚刚创建的芯片名称，验证各个命令及返回的数据计算项是否正确。

十、常见问题及注意事项

(1) 外接引线长度

当使用导线连接外部设备或芯片时，导线不可过长，一般控制在 20CM 以内，IIC、SPI、UART 等数字接口数据线驱动能力有限，过长的导线会导致通讯波形迟缓。当导线确实无法缩短时，可通过降低通讯速率的方法来解决、缓解通讯异常问题。

(2) IIC 上拉电阻

IIC 总线的 SDA 和 SCL 在板上已经连接了 2k 的上拉电阻，当外接 IIC 设备或者芯片时，无需连接上拉电阻，当重复连接的上拉电阻低于 1k 时会导致无法正常通讯或者本产品永久性损坏。

(3) UART 通讯速率

修改系统时钟分频系数会导致 UART 通讯速率上限无法正常工作，UART 通讯速率上限与分频参数呈反比关系，最高支持的通讯速率=921600bps/分频值（8 分频时，UART 最高的通讯速率=921600/8=115200bps），在修改系统分频系数时，注意先降低 UART 的通讯速率，以免发生无法通讯的问题。

(4) 电压范围

USB2S 支持 3.0~5.5V 的宽电压(VCC)，每个通讯管脚最高承受电压下限为-0.3V，上限为(VCC+0.3)V，超过正常电压范围会导致模块永久性损坏。外接测试设备或者芯片时应特别注意。

(5) 功耗与突发电流

USB2S 自身功率约为 50mA，使用 USB 接口供电时应注意连接的外部芯片或设备总体功耗不得超过 500mA，插拨瞬间的电流突变会导致电压下降然后自动恢复，在此过程中 USB2S 可能发生复位重启并且与计算机之间的连接产生短时间断开属于正常现象。S2STool 工具具有检测硬件并自动重新连接的功能。当使用其它第三方调试工具时，一般在插拨操作后重新连接一次即可。

文档修改历史



V1.30 (20191028)

增加了[FIND]、[HELP]指令说明

河北稳控科技有限公司

2019 年 10 月