

以太网协议栈芯片 CH395

手册

版本: 1D

<http://wch.cn>

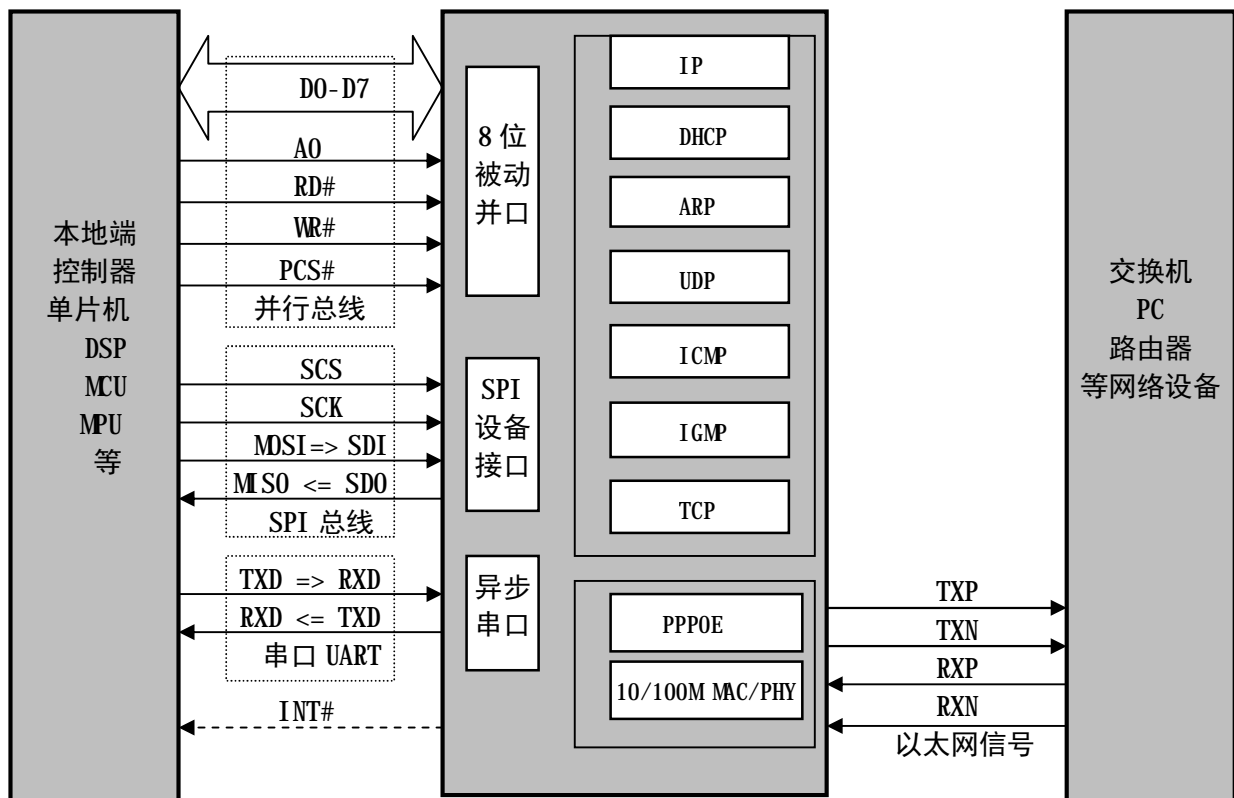
1、概述

CH395 是以太网协议栈管理芯片，用于单片机系统进行以太网通讯。

CH395 芯片自带 10/100M 以太网介质传输层 (MAC) 和物理层 (PHY)，完全兼容 IEEE802.3 10/100M 协议，内置了 PPPoE、IP、DHCP、ARP、ICMP、IGMP、UDP、TCP 等以太网协议栈固件。单片机系统可以方便的通过 CH395 芯片进行网络通讯。

CH395 支持三种通讯接口：8 位并口、SPI 接口或者异步串口，单片机/DSP/MCU/MPU 等控制器可以通过上述任何一种通讯接口控制 CH395 芯片进行以太网通讯。

下图为 CH395 的应用框图。



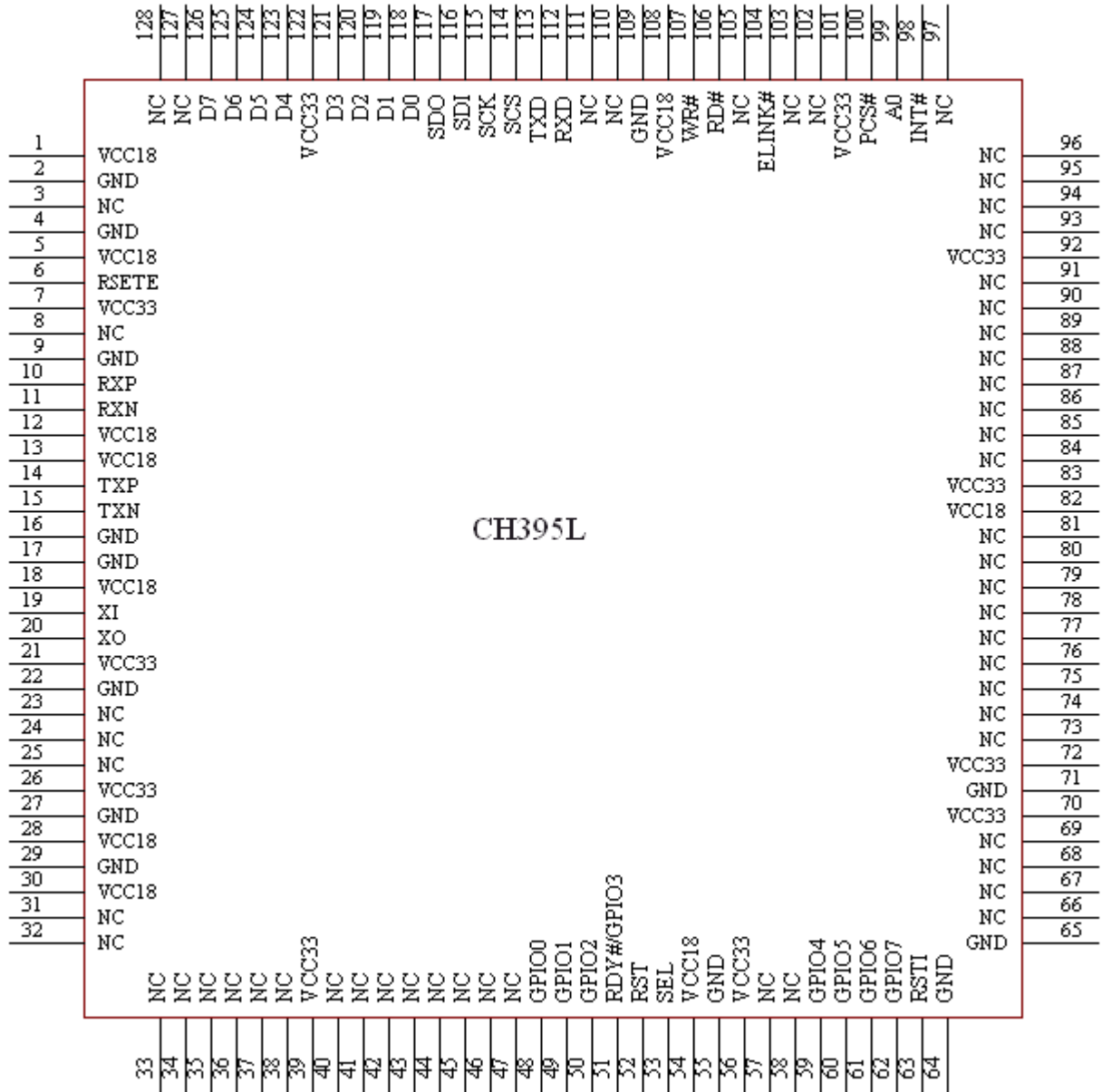
2、特点

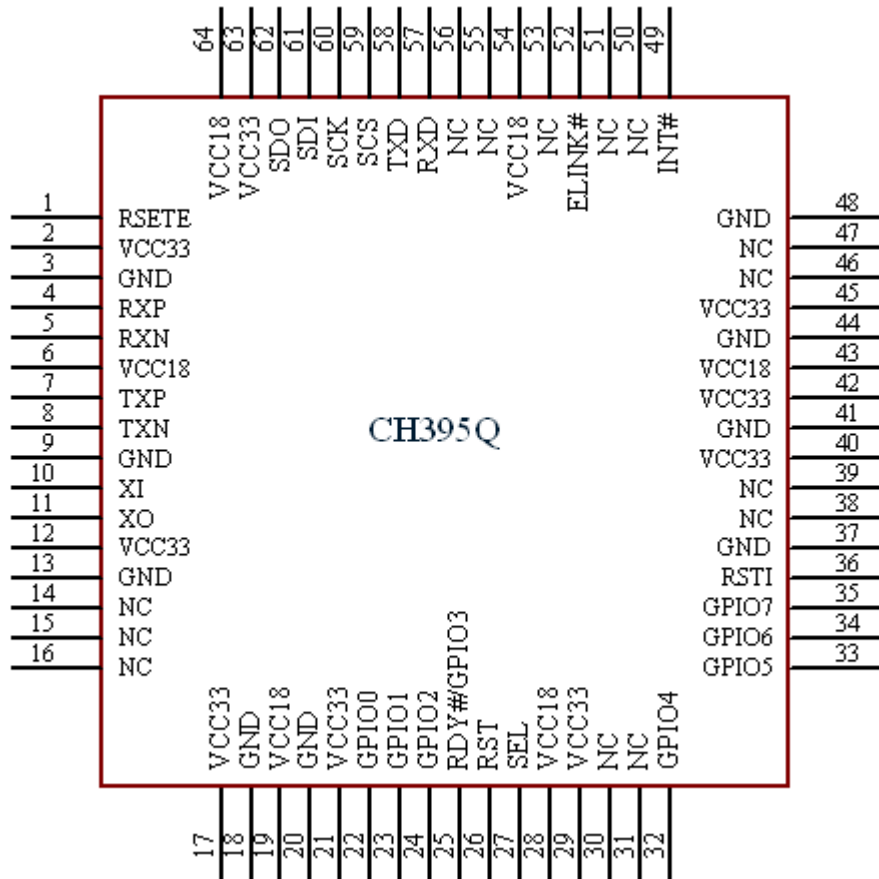
- 内部自带以太网介质传输层 (MAC) 和物理层 (PHY)。
- 支持 10/100M 全双工/半双工自适应，兼容 802.3 协议。
- 支持多种模式的地址过滤。
- 与 802.3x 全双工流控和半双工背压流控完全兼容。
- 支持 MDI/MDIX 线路自动转换。
- 内置 TCP/IP 协议簇，支持 IPv4、DHCP、ARP、ICMP、IGMP、UDP、TCP 协议。
- 内置 PPPoE 协议，支持 PAP 和 CHAP 认证。
- 提供 8 个独立的 Socket 对，可以同时进行数据收发。
- 提供高速 8 位被动并行接口，支持连接到单片机的并行数据总线。
- 提供最高 30MHz 速度的 SPI 设备接口，支持连接到单片机的 SPI 串行总线。

- 提供最高 3Mbps 速度的异步串口，支持连接到单片机的串口，支持通讯波特率动态调整。
- 支持低功耗模式。
- 内置 24K RAM，可用于以太网数据收发，每个 Socket 收发缓冲区可以自由配置。
- 内置 4KB EEPROM。
- 支持 8 路 GPIO。
- 提供 LQFP64M和 LQFP128 无铅封装，兼容 RoHS。

注：仅 CH395Q 支持 PPPoE。

3、封装





芯片型号	芯片封装	
	名称	描述
CH395L	LQFP128	LQFP 封装; 128 脚; 本体 14x14mm
CH395Q	LQFP64M	LQFP 封装; 64 脚; 本体 10x10mm

4、引脚

CH395L 引脚号	CH395Q 引脚号	引脚 名称	类型	引脚说明
7、21、26、39、 56、70、72、83、 92、101、122	2、12、17、 21、29、40、 42、45、63	VCC33	电源	3.3V 正电源输入端, 外接 0.1uF 电源退耦电容
1、5、12、13、 18、28、30、54、 82、108	6、19、28、 43、54、64	VCC18	电源	1.8V 正电源输入端, 外接 0.1uF 电源退耦电容
2、4、9、16、 17、22、27、29、 55、64、65、71、 109	3、9、13、18、 20、37、41、 44、48	GND	电源	公共接地端
3、8、23、24、 25、31、32、33、 34、35、36、37、 38、40、41、42、	14、15、16、 30、31、38、 39、46、47、 50、51、53、			

43、44、45、46、 47、57、58、66、 67、68、69、73、 74、75、76、77、 78、79、80、81、 84、85、86、87、 88、89、90、91、 93、94、95、96、 97、102、103、 105、110、111、 127、128	55、56	NC	-	保留引脚，悬空
6	1	RSETE	输入	外接 12-18K 电阻到地端
10	4	RXP	以太网信号	以太网 RXP 信号
11	5	RXN	以太网信号	以太网 RXN 信号
14	7	TXP	以太网信号	以太网 TXP 信号
15	8	TXN	以太网信号	以太网 TXN 信号
19	10	XI	输入	晶体振荡的输入端，需要外接 30MHz 晶体
20	11	XO	输出	晶体振荡的反相输出端，需要外接 30MHz 晶体
48	22	GPI00	双向三态	GPI00，默认输入
49	23	GPI01	双向三态	GPI01，默认输入
50	24	GPI02	双向三态	GPI02，默认输入
51	25	GPI03/ RDY#	双向三态	GPI03，默认输出，CH395 复位后，输出低电平
52	26	RST	输出	电源上电复位和外部复位输出，高电平有效
53	27	SEL	输入	芯片内部复位期间为接口配置输入，内置上拉电阻
59	32	GPI04	双向三态	GPI04，默认输入
60	33	GPI05	双向三态	GPI05，默认输入
61	34	GPI06	双向三态	GPI06，默认输入
62	35	GPI07	双向三态	GPI07，默认输入
63	36	RSTI	输入	外部复位输入，低电平有效，内置上拉电阻
98	49	INT#	输出	中断请求输出，低电平有效，内置上拉电阻
99	-	A0	输入	并口的地址输入，区分命令口与数据口，内置上拉电阻，当 A0=1 时可以写命令，当 A0=0 时可以读写数据
100	-	PCS#	输入	并口的片选控制输入，低电平有效，内置上拉电阻
106	-	RD#	输入	并口的读选通输入，低电平有效，内置上拉电阻
107	-	WR#	输入	并口的写选通输入，低电平有效，内置上拉电阻
112	57	RXD	输入	异步串口的串行数据输入，内置上拉电阻
113	58	TXD	输入 输出	在芯片内部复位期间为接口配置输入，内置上拉电阻，在芯片复位完成后为异步串口的串行数据输出
114	59	SCS	输入	SPI 接口的片选输入，低电平有效，内置上拉电阻
115	60	SCK	输入	SPI 接口的串行时钟输入，内置上拉电阻
116	61	SDI	输入	SPI 接口的串行数据输入，内置上拉电阻
117	62	SDO	三态输出	SPI 接口的串行数据输出
118~121 123~126	-	DO~D7	双向三态	并口的 8 位双向数据总线，内置上拉电阻
104	52	ELINK	输出	以太网连接通讯指示灯驱动引脚

(注：本表中标注为灰色的引脚，只能承受 3.3V 输入电压，未标注的引脚可承受 3.3V 和 5V 输入。)

5、命令

本手册中的数据，后缀 B 为二进制数，后缀 H 为十六进制数，否则为十进制数。

低字节在前 (Little-Endian) 的双字数据 (共 32 位) 指：先是最底字节 (位 7~位 0)，接着是较低字节 (位 15~位 8)，然后是较高字节 (位 23~位 16)，最后是最高字节 (位 31~位 24)。

数据流是指由若干连续字节组成的数据块，数据块的总长度最小为 0，最大为 4096。

下表输入数据和输出数据的括号中的数字为参数的字节数，没有括号则默认为 1 个字节。

本手册中所指的单片机基本适用于 DSP 或者 MCU/MPU/SCM 等。

Socket Pair(插口对)包含源 IP，源端口，目的 IP 和目的端口的四元组，可以唯一确定互联网中的连接双方。本手册简称 Socket。CH395 内部可以同时提供 8 个 Socket，其索引值分别为 0, 1, 2, 3, 4, 5, 6, 7。

本手册约定的 IP 和 MAC 地址高字节和低字节可能与某些文档不同，仅为方便说明：

如 IP 地址为 192.168.1.2，其中 192 为最低字节，2 为最高字节。本文称 (IP) 低字节在前。

如 MAC 地址为 00.01.02.03.04.05，其中 00 为最低字节，05 为最高字节。本文称 (MAC) 低字节在前。

本手册中所有包含 IP 地址输入或者输出的命令，字节顺序均为 IP 低字节在前。

本手册中所有包含 MAC 地址输入或者输出的命令，字节顺序均为 MAC 低字节在前。

代码	命令名称 CMD_	输入数据	输出数据	命令用途
01H	GET_IC_VER		版本号	获取芯片及固件版本
02H	SET_BAUDRATE	波特率系数(3)	(等 1mS) 操作状态	设置串口通讯波特率
03H	ENTER_SLEEP			进入低功耗睡眠挂起状态
05H	RESET_ALL		(等 50mS)	执行硬件复位
19H	GET_GLOB_INT_STATUS_ALL		全局中断状态(2)	获取全局中断状态
06H	CHECK_EXIST	任意数据	按位取反	测试通讯接口和工作状态
20H	SET_PHY	PHY 连接方式		设置 PHY 连接方式
21H	SET_MAC_ADDR	MAC 地址 (6)		设置 MAC 地址
22H	SET_IP_ADDR	IP 地址 (4)		设置 IP 地址
23H	SET_GWIP_ADDR	网关地址 (4)		设置网关 IP 地址
24H	SET_MASK_ADDR	子网掩码 (4)		设置子网掩码
25H	SET_MAC_FILT	过滤模式		设置 MAC 过滤模式
		HASH0(4)		
		HASH1(4)		
26H	GET_PHY_STATUS		PHY 的状态	获取 PHY 的状态
27H	INIT_CH395			初始化 CH395 芯片
28H	GET_UNREACH_IPPORT		不可达信息 (8)	获取不可达 IP, 端口和协议
29H	GET_GLOB_INT_STATUS		全局中断状态	获取全局中断状态
2AH	SET_RETRAN_COUNT	重试次数		设置重试次数, 最大 20 次
2BH	SET_RETRAN_PERIOD	重试周期(2)		设置重试周期, 最大 1000MS
2CH	GET_CMD_STATUS		命令执行状态	获取命令执行状态
2DH	GET_REMT_IPP_SN	Socket 索引	IP 和端口 (6)	获取远端(目的)的 IP 和端口
2EH	CLEAR_RECV_BUF_SN	Socket 索引		清空 Socket 的接收缓冲区
2FH	GET_SOCKET_STATUS_SN	Socket 索引	Socket 状态	获取 Socket 状态
30H	GET_INT_STATUS_SN	Socket 索引	Socket 中断	获取 Socket 的中断状态
31H	SET_IP_ADDR_SN	Socket 索引 目的 IP(4)		设置 Socket 的目的 IP 地址

32H	SET_DES_PORT_SN	Socket 索引		设置 Socket 的目的端口
		目的端口(2)		
33H	SET_SOUR_PORT_SN	Socket 索引		设置 Socket 的源端口
		源端口(2)		
34H	SET_PROTO_TYPE_SN	Socket 索引		设置 Socket 的工作模式
		协议类型		
35H	OPEN_SOCKET_SN	Socket 索引		打开 Socket
36H	TCP_LISTEN_SN	Socket 索引		启动 Socket 监听
37H	TCP_CONNECT_SN	Socket 索引		启动 Socket 连接
38H	TCP_DISNCONNECT_SN	Socket 索引		断开 Socket 的 TCP 连接
39H	WRITE_SEND_BUF_SN	Socket 索引		向 Socket 发送缓冲区写数据
		长度(2)		
		数据流(N)		
3BH	GET_RECV_LEN_SN	Socket 索引	长度(2)	获取 Socket 接收数据长度
3CH	READ_RECV_BUF_SN	Socket 索引	数据流(N)	从 Socket 接收缓冲区接收数据
		长度(2)		
3DH	CLOSE_SOCKET_SN	Socket 索引		关闭 Socket
3EH	SET_IPRAW_PRO_SN	Socket 索引		设置 Socket 的 IP 包的协议字段
		IP 协议字段		
3FH	PING_ENABLE	使能标志		PING 使能
40H	GET_MAC_ADDR		MAC 地址 (6)	获取 MAC 地址
41H	DHCP_ENABLE	使能标志		启动 (停止) DHCP
42H	GET_DHCP_STATUS		DHCP 状态	获取 DHCP 状态
43H	GET_IP_INF		IP 等信息	获取 IP, MASK, DNS 等信息
44H	PPPOE_SET_USER_NAME	PPPOE 用户名		设置 PPPOE 用户名
45H	PPPOE_SET_PASSWORD	PPPOE 密码		设置 PPPOE 密码
46H	PPPOE_ENABLE	使能标志		启动 (停止) PPPOE
47H	GET_PPPOE_STATUS		PPPOE 状态	获取 PPPOE 状态
50H	SET_TCP_MSS	TCP MSS (2)		设置 TCP MSS
51H	SET_TTL	TTL		设置 TTL 值, 最大 128
52H	SET_RECV_BUF	Socket 索引		设置 Socket 接收缓冲区
		起始块地址		
		块个数		
53H	SET_SEND_BUF	Socket 索引		设置 Socket 发送缓冲区
		起始块地址		
		块个数		
55H	SET_FUN_PARA	功能参数(4)		设置功能参数
56H	SET_KEEP_LIVE_IDLE	4 字节时间参数		设置 KEEPLIVE 空闲时间
57H	SET_KEEP_LIVE_INTVL	4 字节时间参数		设置 KEEPLIVE 超时时间
58H	SET_KEEP_LIVE_CNT	重试次数		设置 KEEPLIVE 超时重试次数
59H	SET_KEEP_LIVE_SN	Socket 索引		设置设置 Socket KEEPLIVE
		配置		
E9H	EEPROM_ERASE			擦除 EEPROM
EAH	EEPROM_WRITE	EEPROM地址 (2)		写 EEPROM
		长度		
		数据流(N)		

EBH	EEPROM_READ	EEPROM地址 (2)	数据流(N)	读 EEPROM
		长度		
ECH	READ_GPIO_REG	GPIO寄存器地址	GPIO寄存器值	读 GPIO 寄存器
EDH	WRITE_GPIO_REG	GPIO寄存器地址		写 GPIO 寄存器
		GPIO寄存器值		

表中阴影部分的命令通常需要执行一定的时间并且查询该命令的执行状态，单片机可以通过 GET_CMD_STATUS 来获取状态。(关于状态定义参考 CH395INC.H)

5.1. CMD_GET_IC_VER

该命令用于获取芯片及固件版本。返回的 1 个字节数据是版本号，其位 7 为 0，位 6 为 1，位 5~位 0 为版本号。如果返回值为 41H，去掉位 7 和位 6，版本号为 01H。本文统一称为芯片版本。

5.2. CMD_SET_BAUDRATE

该命令用于设置 CH395 的串口通讯波特率。在 CH395 工作于串口通讯方式时，复位后默认的通讯波特率由 SD0, SDI 和 SCK 三个引脚的电平组合设定 (参考本手册第 6.4 节)，这些引脚都悬空时默认是 9600bps。如果单片机支持较高的通讯速度，可以通过该命令动态调整串口通讯波特率。该命令需要输入三个数据，分别是波特率系数 0、波特率系数 1、波特率系数 2，下表是与波特率的对应关系。

波特率系数 2	波特率系数 1	波特率系数 0	串口通讯波特率 (bps)	误差
00H	12H	00H	4800	0%
00H	25H	80H	9600	0%
00H	4BH	00H	19200	0%
00H	96H	00H	38400	0%
00H	E1H	00H	57600	0%
01H	2CH	00H	76800	0.6%
01H	C2H	00H	115200	0.9%
07H	08H	00H	460800	0%
0EH	10H	00H	921600	7.1%
01H	86H	A0H	100000	0%
0FH	42H	40H	1000000	7.6%
2DH	C6H	00H	3000000	0%

计算公式：

$$\text{BaudRate} = (\text{波特率系数 2} \ll 16) + (\text{波特率系数 1} \ll 8) + \text{波特率系数 0}$$

通常情况下，设置串口通讯波特率在 1ms 时间之内完成，完成后 CH395 以新设定的通讯波特率输出操作状态，所以单片机应该在发出命令后及时调整自身的通讯波特率。

5.3. CMD_ENTER_SLEEP

该命令使 CH395 芯片进入低功耗睡眠挂起状态。当单片机向 CH395 写入新的命令时(没有数据输入的命令，如 CMD_GET_IC_VER)，会退出低功耗状态。对于并口和 SPI 接口通讯方式，SCS 片选有效也将导致 CH395 退出低功耗状态，所以单片机在发出 CMD_ENTER_SLEEP 命令后，应该立即使 SCS 片选无效。

进入睡眠状态 CH395 的 MAC 和 PHY 将会进入掉电模式，断开以太网连接。

通常情况下，CH395 退出低功耗状态需要几毫秒的时间。

5.4. CMD_RESET_ALL

该命令使 CH395 执行硬件复位。通常情况下，硬件复位在 50mS 时间之内完成。

5.5. CMD_CHECK_EXIST

该命令用于测试通讯接口和工作状态，以检查 CH395 是否正常工作。该命令需要输入 1 个字节数据，可以是任意数据，如果 CH395 正常工作，那么 CH395 的输出数据是输入数据的按位取反。例如，输入数据是 57H，则输出数据是 A8H。

5.6. CMD_SET_PHY

该命令用设置 CH395 以太网 PHY 的连接方式。默认为自动协商方式。该命令需要输入 1 个字节数据，该数据为连接方式代码：

连接方式代码为 01H 时断开 PHY 连接；
 连接方式代码为 02H 时 PHY 为 10M 全双工；
 连接方式代码为 04H 时 PHY 为 10M 半双工；
 连接方式代码为 08H 时 PHY 为 100M 全双工；
 连接方式代码为 10H 时 PHY 为 100M 半双工；
 连接方式代码为 20H 时 PHY 自动协商。

CH395 收到此命令后会复位 MAC 和 PHY，按照新设置的连接方式重新进行连接。如果以太网已经连接，则会断开并重新连接。

5.7. CMD_SET_MAC_ADDR

该命令用于设置 CH395 的 MAC 地址，需要输入 6 个字节的 MAC，MAC 地址低字节在前。CH395 芯片将会把 MAC 地址保存在内部的 EEPROM 中，该命令执行需要 100mS。

CH395 芯片出厂时已经烧录了由 IEEE 分配的 MAC 地址，如非必要请勿设置 MAC 地址。

5.8. CMD_SET_IP_ADDR

该命令用于设置 CH395 的 IP 地址，需要输入 4 个字节的 IP 地址，IP 低字节在前。本手册中所有包含 IP 输入或者输出的命令，字节顺序均为 IP 低字节在前，以下将不再说明。

5.9. CMD_SET_GWIP_ADDR

该命令用于设置 CH395 的网关地址，需要输入 4 个字节的 IP 地址。

5.10. CMD_SET_MASK_ADDR

该命令用于设置 CH395 的子网掩码，该命令需要输入 4 个字节的掩码。默认为 255.255.255.0，可以不设置。

5.11. CMD_SET_MAC_FILT

该命令用于设置 MAC 过滤模式。MAC 可以设置多种过滤模式。该命令需要输入 9 个字节数据，第 1 个字节为过滤模式，该数据的各位含义如下：

位	名称	描述
[5:7]	-	保留

4	SEND_ENABLE	发送使能
3	RECV_ENABLE	接收使能
2	RECV_MULTIPKT	接收多播包
1	RECV_ALL	接收所有数据
0	RECV_BROADPKT	接收广播包

以上各位为 1 表示开启, 为 0 表示关闭。CH395 复位后默认 RECV_BROADPKT, RECV_ENABLE, SEND_ENABLE 开启。

下表为各位的含义:

RECV_ENABLE	RECV_ALL	RECV_BROADPKT	RECV_MULTIPKT	描述
1	0	0	0	接收与 MAC 地址相匹配的数据包
1	1	X	X	接收所有数据包
1	0	1	0	接收与 MAC 地址相匹配的数据包 接收广播数据包
1	0	0	1	接收与 MAC 地址相匹配的数据包 接收多播数据包
1	0	1	1	接收与 MAC 地址相匹配的数据包 接收广播数据包 接收多播数据包
0	X	X	X	禁止接收

第 2 至第 5 字节为 HASH0 (哈希表 0), 第 6 至第 9 字节为 HASH1 (哈希表 1), HASH0 与 HASH1 仅在多播开启下有效。

HASH0 和 HASH1 共同构成一个 64 位的 HASH 表, 0-31 位为 HASH0, 32-63 位为 HASH1。

HASH 表计算方法: 将多播地址使用标准的以太网冗余校验 (CRC32) 计算出一个 32 位的 CRC 值, 使用此 CRC 值的高 6 位作为索引值, 将 HASH 表对应的位写 1。例如, 多播地址计算出 CRC 值的高 6 位为 32, 则应该将 HASH1 的第 0 位写 1。

5.12. CMD_GET_PHY_STATUS

该命令用于获取 PHY 的连接状态。收到该命令后, CH395 会查询当前 PHY 连接状态并输出 1 个字节的 PHY 的连接状态代码:

- 连接状态代码为 01H 时表示 PHY 连接断开;
- 连接状态代码为 02H 时表示 PHY 连接为 10M 全双工;
- 连接状态代码为 04H 时表示 PHY 连接为 10M 半双工;
- 连接状态代码为 08H 时表示 PHY 连接为 100M 全双工;
- 连接状态代码为 10H 时表示 PHY 连接为 100M 半双工。

5.13. CMD_INIT_CH395

该命令用于初始化 CH395, 包括: 初始化 CH395 的 MAC, PHY 以及 TCP/IP 协议栈。该命令需要约 350mS 执行完毕。单片机可以发送 GET_CMD_STATUS 来查询是否执行完毕和执行状态。

5.14. CMD_GET_UNREACH_IPPORT

该命令用于获取不可达 IP, 端口以及协议类型。当接收到不可达报文后, CH395 会产生不可达中断。单片机可以用该命令来获取不可达信息, 收到此命令后 CH395 会依次输出 1 字节的不可达代码, 1 个字节协议类型, 2 个字节的端口号 (低字节在前), 4 个字节 IP。单片机可以根据不可达代码来判

断是协议不可达，端口不可达还是 IP 不可达。关于不可达代码请参考 RFC792（CH395INC.H 定义了 4 种常见的不可达代码）。

5.15. CMD_GET_GLOB_INT_STATUS

该命令用于获取全局中断状态，收到此命令后 CH395 会输出 1 个字节的全局中断状态。全局中断状态定义如下：

位	名称	描述
7	GINT_STAT_SOCK3	Socket3 中断
6	GINT_STAT_SOCK2	Socket2 中断
5	GINT_STAT_SOCK1	Socket1 中断
4	GINT_STAT_SOCK0	Socket0 中断
3	GINT_STAT_DHCP 和 GINT_STAT_PPPOE	DHCP 和 PPPOE 中断
2	GINT_STAT_PHY_CHANGE	PHY 状态改变中断
1	GINT_STAT_IP_CONFLI	IP 冲突
0	GINT_STAT_UNREACH	不可达中断

① GINT_STAT_UNREACH: 不可达中断。当 CH395 收到 ICMP 不可达中断报文后，将不可达 IP 数据包的 IP 地址，端口，协议类型保存到不可达信息表中，然后产生此中断，单片机收到此中断，可以发送 GET_UNREACH_IPPORT 命令来获取不可达信息。

② GINT_STAT_IP_CONFLI: IP 冲突中断。当 CH395 检测到自身 IP 地址和同一网段内的其他网络设备 IP 地址相同时会产生此中断。

③ GINT_STAT_PHY_CHANGE: PHY 变化中断。当 CH395 的 PHY 连接有变化时产生此中断，例如 PHY 状态由连接状态变化为断开状态或者由断开状态变化为连接状态。单片机可以发送 GET_PHY_STATUS 命令来获取当前 PHY 连接的状态。

④ GINT_STAT_DHCP 和 GINT_STAT_PPPOE: DHCP 中断和 PPPOE 中断共用此中断源，如果单片机使能 CH395 的 DHCP 功能，CH395 将会产生此中断，单片机可以发送 CMD_GET_DHCP_STATUS 命令获取 DHCP 状态，如果状态为 0，表示成功，否则超时失败。

如果外部单片机使能 CH395 的 PPPOE 功能，CH395 也会产生此中断，单片机可以发送命令 CMD_GET_DHCP_STATUS 获取 PPPOE 的状态，0 表示成功，其他值为错误码。

⑤ GINT_STAT_SOCK0 - GINT_STAT_SOCK3: Socket 中断。当 Socket 有中断事件时 CH395 会产生此中断，单片机需要发送 GET_INT_STATUS_SN 获取 Socket 的中断状态。请参考 GET_INT_STATUS_SN 部分。

此命令执行完毕后，CH395 会将 INT#引脚置为高电平并将全局中断状态清除。

5.16. CMD_GET_GLOB_INT_STATUS_ALL

该命令用于获取全局中断状态，收到此命令后 CH395 会输出 2 个字节的全局中断状态。全局中断状态定义如下：

位	名称	描述
[12: 18]	-	保留
11	GINT_STAT_SOCK7	Socket7 中断
10	GINT_STAT_SOCK6	Socket6 中断
9	GINT_STAT_SOCK5	Socket5 中断
8	GINT_STAT_SOCK4	Socket4 中断
7	GINT_STAT_SOCK3	Socket3 中断

6	GINT_STAT_SOCK2	Socket2 中断
5	GINT_STAT_SOCK1	Socket1 中断
4	GINT_STAT_SOCK0	Socket0 中断
3	GINT_STAT_DHCP 和 GINT_STAT_PPPOE	DHCP 和 PPPOE 中断
2	GINT_STAT_PHY_CHANGE	PHY 状态改变中断
1	GINT_STAT_IP_CONFLI	IP 冲突
0	GINT_STAT_UNREACH	不可达中断

关于位 0-7 请参考 5.15 章节。

GINT_STAT_SOCK4 - GINT_STAT_SOCK7: Socket 中断。当 Socket 有中断事件时 CH395 会产生此中断，单片机需要发送 GET_INT_STATUS_SN 获取 Socket 的中断状态。请参考 GET_INT_STATUS_SN 部分。

CH395 获取可以通过 CMD_GET_GLOB_INT_STATUS 和 CMD_GET_GLOB_INT_STATUS_ALL 两个命令来获取中断状态，前者只能获取到低 8 位的中断状态，后者可以获取全部的中断状态，使用时需要注意，任何版本的芯片都支持 CMD_GET_GLOB_INT_STATUS 命令，如果芯片版本号大于等于 0X44 且使用了 Socket4 - Socket7 则只能用 CMD_GET_GLOB_INT_STATUS_ALL。芯片版本号小于 0X44 不支持 CMD_GET_GLOB_INT_STATUS_ALL 命令。

5.17. CMD_SET_RETRAN_COUNT

该命令用于设置重试次数，需要输入 1 个字节的重试次数，允许最大值为 20，输入数据大于 20，则会按 20 处理。默认重试次数为 12 次，重试仅在 TCP 模式下有效。

5.18. CMD_SET_RETRAN_PERIOD

该命令用于设置重试周期，需要输入 2 个字节（低字节在前）的周期数，单位为毫秒，允许最大值为 1000。重试的总时间为 $N * M$ ，N 为重试次数，M 为重试周期。默认重试周期为 500MS，重试仅在 TCP 模式下有效。

5.19. CMD_GET_CMD_STATUS

该命令用于获取命令执行的状态。CH395 会输出 1 个字节数据，此数据为命令执行的状态。命令执行状态如下：

代码	名称	描述
00H	CH395_ERR_SUCCESS	成功
10H	CH395_ERR_BUSY	忙，表示命令正在执行
11H	CH395_ERR_MEM	内存管理错误
12H	CH395_ERR_BUF	缓冲区错误
13H	CH395_ERR_TIMEOUT	超时
14H	CH395_ERR_RTE	路由错误
15H	CH395_ERR_ABRT	连接中止
16H	CH395_ERR_RST	连接复位
17H	CH395_ERR_CLSD	连接关闭
18H	CH395_ERR_CONN	无连接
19H	CH395_ERR_VAL	值错误
1AH	CH395_ERR_ARG	参数错误

1BH	CH395_ERR_USE	已被使用
1CH	CH395_ERR_IF	MAC 错误
1DH	CH395_ERR_ISCONN	已连接
20H	CH395_ERR_OPEN	已打开

如果单片机收到 CH395_ERR_BUSY，表示 CH395 正在执行命令，单片机则应该延时 2 毫秒以上再次发送 CMD_GET_CMD_STATUS 命令获取状态。

命令码表中阴影部分的命令均需要发送 CMD_GET_CMD_STATUS 来获取执行状态。

5.20. CMD_GET_REMDT_IPP_SN

该命令用于获取远端的 IP 地址和端口号，需要输入 1 个字节的 Socket 索引值，CH395 会输出 4 个字节 IP 地址，2 个字节（低字节在前）端口号。Socket 工作在 TCP Server 模式且连接建立后，单片机可以通过该命令获取远端的 IP 地址及端口号。

5.21. CMD_CLEAR_RECV_BUF_SN

该命令用于清空 Socket 的接收缓冲区，需要输入 1 个字节的 Socket 索引值。收到此命令，CH395 会将此 Socket 的接收长度清零，接收指针指向缓冲区头部。

5.22. CMD_GET_SOCKET_STATUS_SN

该命令用于获取 Socket 的状态，需要输入 1 个字节的 Socket 索引值。CH395 接收到此命令后会输出 2 个字节的状态码。

第 1 个状态码为 Socket 的状态码，Socket 的状态码定义如下：

代码	名称
00H	SOCKET_CLOSED
05H	SOCKET_OPEN

第 2 个状态码为 TCP 的状态码，仅在 TCP 模式且已经被打开时有意义，TCP 状态码定义如下：

代码	名称	描述
00H	TCP_CLOSED	关闭
01H	TCP_LISTEN	监听
02H	TCP_SYN_SENT	SYN 发送
03H	TCP_SYN_RCVD	SYN 接收
04H	TCP_ESTABLISHED	TCP 连接建立
05H	TCP_FIN_WAIT_1	主动关闭方首次发送 FIN
06H	TCP_FIN_WAIT_2	主动关闭方收到 FIN 的 ACK
07H	TCP_CLOSE_WAIT	被动关闭方收到 FIN
08H	TCP_CLOSING	正在关闭
09H	TCP_LAST_ACK	被动关闭方发送 FIN
0AH	TCP_TIME_WAIT	2MS 等待状态

TCP 状态均为 TCP/IP 协议中规定的状态，详细的含义请参考 TCP/IP 协议。

5.23. CMD_GET_INT_STATUS_SN

该命令用于获取 Socket 的中断状态，需要输入 1 个字节的 Socket 索引值，CH395 收到该命令后会输出 1 个字节的 Socket 的中断码，中断码各位定义如下：

位	名称	描述
7	-	保留
6	SINT_STAT_TIM_OUT	超时
5	-	保留
4	SINT_STAT_DISCONNECT	TCP 断开
3	SINT_STAT_CONNECT	TCP 连接
2	SINT_STAT_RECV	接收缓冲区非空
1	SINT_STAT_SEND_OK	发送成功
0	SINT_STAT_SENBUF_FREE	发送缓冲区空闲

① SINT_STAT_SENBUF_FREE，发送缓冲区空闲中断，单片机向 Socket 发送缓冲区写入数据后，CH395 会将数据快速复制到内部的协议栈或者 MAC 缓冲区中，以便封装数据，当数据复制完毕后，会产生此中断，单片机可以继续向发送缓冲区写入后续数据。单片机向 Socket 发送缓冲区写入一次数据后，必须等到产生此中断后才可以写入下一次数据。

② SINT_STAT_SEND_OK，发送成功中断，产生此中断，表示数据包被成功发送，Socket 每发送成功一包数据后会产生此中断。单片机向 Socket 缓冲区写入一次数据后，CH395 可能会封装成若干个数据包进行发送，所以可能会产生若干次发送成功中断。

③ SINT_STAT_CONNECT，TCP 连接中断，仅在 TCP 模式下有效。表明 TCP 连接成功，单片机必须在产生此中断后，才可以进行数据传输。

④ SINT_STAT_DISCONNECT，TCP 连接断开中断，仅在 TCP 模式下有效，表明 TCP 连接断开。

⑤ SINT_STAT_TIM_OUT，TCP 模式下，TCP 连接、断开、发送数据等过程中出现超时，会产生此中断。IPRAW、UDP 模式下，发送数据失败也会产生此中断。

在产生 SINT_STAT_DISCONNECT 和 SINT_STAT_TIM_OUT 中断后，根据 FUN_PARA_FLAG_SOCKET_CLOSE 位为 1 还是为 0，CH395 会做出不同的动作。如果 FUN_PARA_FLAG_SOCKET_CLOSE 为 0 则当产生上述两种中断后，CH395 会主动将 Socket 状态置为关闭状态并将相关缓冲区全部清除。反之则不会对 Socket 状态和相关缓冲区做任何操作，以方便外部单片机在 TCP 断开或者超时后将残留的数据读出，当外部单片机将数据读出后，必须发送关闭命令将 Socket 关闭。

5.24. CMD_SET_IP_ADDR_SN

该命令用于设置 Socket 目的 IP 地址，需要输入 1 个字节的 Socket 索引值和 4 个字节的 IP 地址。在 Socket 工作在 IPRAW、UDP、TCP Client 模式下，必须在发送 CMD_OPEN_SOCKET_SN 命令之前设置目的 IP。

5.25. CMD_SET_DEST_PORT_SN

该命令用于设置 Socket 目的端口，需要输入 1 个字节的 Socket 索引值和 2 个字节的端口（低字节在前）。Socket 工作在 UDP、TCP Client 模式下，此值必须设置。

5.26. CMD_SET_SOURCE_PORT_SN

该命令用于设置 Socket 源端口，需要输入 1 个字节的 Socket 索引值和 2 个字节的源端口（低字节在前）。如果两个或者多个 Socket 都采用相同的模式，则源端口号不得相同。例如 Socket 0 为 UDP 模式，源端口为 600，Socket 1 同样为 UDP 模式，不可以再使用源端口 600，否则可能会导致打开失败。

5.27. CMD_SET_PROTO_TYPE_SN

该命令用于设置 Socket 工作模式，需要输入 1 个字节的 Socket 索引值和 1 个字节的工作模式，工作模式定义如下：

代码	名称	描述
03H	PROTO_TYPE_TCP	TCP 模式
02H	PROTO_TYPE_UDP	UDP 模式
01H	PROTO_TYPE_MAC_RAW	MAC 原始报文模式
00H	PROTO_TYPE_IP_RAW	IP 原始报文模式

此命令必须在 CMD_OPEN_SOCKET_SN 之前执行，详细步骤请参考 8.3 应用参考步骤。

5.28. CMD_OPEN_SOCKET_SN

该命令用于打开 Socket，使用 Socket 的必要步骤，需要输入 1 个字节的 Socket 索引值。在发送完此命令后，单片机应该发送 GET_CMD_STATUS 来查询命令执行状态。在 UDP、IPRAW、MACRAW 模式下打开 Socket 并返回成功后，即可进行数据传输。在发送此命令前，必须对目的 IP、协议类型、源端口、目的端口等进行必要设置，详细步骤请参考 8.3 应用参考步骤。

5.29. CMD_TCP_LISTEN_SN

该命令仅在 TCP 模式下有效，使能 Socket 进入监听模式即 TCP Server 模式。需要输入 1 个字节的 Socket 索引，该命令必须在 OPEN_SOCKET_SN 之后执行。在发送完此命令后，单片机应该发送 GET_CMD_STATUS 来查询命令执行的状态。

在 TCP Server 模式下，Socket 会一直检测连接事件，直到连接成功后，会产生 SINT_STAT_CONNECT 中断。每个 Socket 只可以建立一个连接。如果再次接收到符合条件的连接事件，Socket 将向试图连接的远端发送 TCP RESET。

5.30. CMD_TCP_CONNECT_SN

该命令仅在 TCP 模式下有效，使能 Socket 进入连接模式即 TCP Client 模式，需要输入 1 个字节的 Socket 索引值。在发送完此命令后，单片机应该发送 GET_CMD_STATUS 来查询命令执行的状态。

在收到此命令后，Socket 将会发起连接事件，成功连接后，会产生 SINT_STAT_CONNECT 中断。如果连接过程中出现异常或者超过一定时间后未连接成功，会产生 SINT_STAT_TIM_OUT 中断。单片机接收到此中断，如果需要再次连接，需要重新打开 Socket，并进行 TCP_CONNECT_SN。

5.31. CMD_TCP_DISCONNECT_SN

该命令仅在 TCP 模式下有效，断开当前 TCP 连接，需要输入 1 个字节的 Socket 索引值，在发送完此命令后，单片机应该发送 GET_CMD_STATUS 来查询命令执行的状态。当前的 TCP 成功断开后会产生 SINT_STAT_DISCONNECT。

5.32. CMD_WRITE_SEND_BUF_SN

该命令用于向 Socket 的发送缓冲区写入数据，需要输入 1 个字节的 Socket 索引值，2 个字节的长度（低字节在前）和若干字节的数据流。输入数据的长度不得大于发送缓冲区的大小，但是在 MACRAW 模式下，输入的数据的长度最大只能为 1514，多余的数据将被丢弃。当外部单片机写完数据后，CH395 会根据 Socket 的工作模式进行数据包的封装，然后将其发送。在单片机未收到

SINT_STAT_SENBUF_FREE 之前, 不得向该 Socket 发送缓冲区再次写入数据。

5.33. CMD_GET_RECV_LEN_SN

该命令用于获取当前接收缓冲区的有效数据长度, 需要输入 1 个字节的 Socket 索引值, CH395 接收到该命令后输出 2 个字节的长度(低字节在前)。

5.34. CMD_READ_RECV_BUF_SN

该命令用于从 Socket 接收缓冲区读取数据, 需要输入 1 个字节的 Socket 索引值, 2 个字节的长度(低字节在前), CH395 会根据长度值输出若干字节的数据流。实际应用中可以先发送 RECV_LEN_SN 命令来获取当前缓冲区的实际有效长度。读取的数据长度可以小于缓冲区的实际有效数据长度, 未读取的数据仍然保留在接收缓冲区中, 单片机可以通过此命令继续读取。

在 MACRAW 模式下, 处理方式有所不同, 在 MACRAW 模式下, 接收缓冲区为帧缓冲区, 只可以缓存 1 帧以太网数据, CH395 在处理完 READ_RECV_BUF_SN 命令后, 会将 Socket0 的接收缓冲区全部清除, 所以单片机应一次将缓冲区的有效数据全部读出。

5.35. CMD_CLOSE_SOCKET_SN

该命令用于关闭 Socket, 需要输入 1 个字节的 Socket 索引值。在关闭 Socket 后, Socket 的接收发送缓冲区被清空, 但是配置信息仍然保留, 下次使用该 Socket 时只需再次打开 Socket 即可。

在 TCP 模式下, 关闭 Socket 前 CH395 会自动断开 TCP 连接。

5.36. CMD_SET_IPRAW_PRO_SN

该命令仅在 IPRAW 模式下有效, 需要输入 1 个字节的 Socket 索引值, 1 个字节的 IP 包协议字段的协议码。该命令必须在 OPEN_SOCKET_SN 之前执行。

如果多个 Socket 均采用了 IPRAW 模式, 则协议码不得重复使用。例如 Socket 0 和 Socket 1 均采用了 IPRAW 模式, 则两个 Socket 的协议码不得相同, 否则可能会导致打开 Socket 时失败。

IPRAW 数据处理优先级高于 UDP 和 TCP, 所以协议码也必须不得与其他 Socket 相同。例如 Socket 0 采用 IPRAW 模式, 协议码为 17 (UDP 协议)。Socket 1 采用了 UDP 模式。这可能会导致 Socket 1 的数据被 Socket 0 接收到。

5.37. CMD_PING_ENABLE

该命令用于开启或者关闭 PING, 需要输入 1 个字节的标志码, 标志码为 1 表示开启 PING, 为 0 则表示关闭 PING。

5.38. CMD_GET_MAC_ADDR

该命令用于获取 MAC 地址, 接收到该命令后, CH395 会输出 6 个字节的 MAC 地址。

5.39. CMD_DHCP_ENABLE

该命令用于启动或者停止 DHCP, 需要输入 1 个字节的标志码, 如果标志码为 1 则表示启动 DHCP, 如果为 0 表示关闭 DHCP, 在启动 DHCP 之前必须先要对 CH395 初始化。

在启动 DHCP 后, CH395 将会向网络上广播 DHCPDISCOVER 报文, 用于发现 DHCP Server, 找到 DHCP Server 后申请地址以及其他配置参数, 然后产生 GINT_STAT_DHCP 中断。产生此中断后, 单片机可以发送命令 GET_DHCP_STATUS 来获取 DHCP 的状态。如果状态为 0, 表示成功, 则单片机可以发送命令 GET_IP_INF 来获取 IP、MASK 等信息。如果状态为 1, 则表示错误, 一般是超时导致的, 例如没有发

现 DHCP Server。

DHCP 在启动后，一直处于工作状态，除非接收到单片机的关闭 DHCP 的命令。在此过程中，如果 DHCP Server 重新给 CH395 分配了一个配置，且此配置和原来不相同，CH395 仍然会产生中断。

产生超时中断后，假如没有发现 DHCP Server，CH395 仍然会继续发送 DHCPDISCOVER 报文，间隔时间约为 16 秒。

此命令大约需要 20MS 执行时间，单片机可以发送 GET_CMD_STATUS 来查询是否执行完毕和执行状态

5.40. CMD_GET_DHCP_STATUS

该命令用于获取 DHCP 的状态，一般在单片机接收到中断后，发生此命令获取 DHCP 的执行状态，CH395 收到此命令后，将 DHCP 状态码输出，状态码有 2 个分别为 0 和 1，其含义如下：

如果状态为 0 表示成功，则单片机可以发送命令 GET_IP_INF 来获取 IP，MASK 等信息。

如果状态为 1 则表示错误，一般是超时导致的，例如没有发现 DHCP Server。

5.41. CMD_GET_IP_INF

该命令用于获取 IP、GatewayIP、MASK、DNS 等信息，收到此命令后，CH395 会依次输出 20 个字节的数据，分别为：4 字节 IP 地址、4 字节网关 IP、4 字节子网掩码、4 字节 DNS1(主 DNS)、4 字节 DNS2(次 DNS)。

在 DHCP 或者 PPPPOE 成功后，可以发送此命令来获取当前 CH395 的信息。如果在 DHCP 或者 PPPPOE 中某些配置没获取到则该配置为 0，例如在局域网络 DHCP 时并不一定会分配 DNS，发送该命令获取配置信息时 DNS1，DNS2 全为 0。

5.42. CMD_PPPOE_SET_USER_NAME

该命令用于设置 PPPPOE 用户名，用户名必须是 ASCII 码且最大长度不得大于 31 个字节。

5.43. CMD_PPPOE_SET_PASSWORD

该命令用于设置 PPPPOE 用密码，密码必须是 ASCII 码且最大长度不得大于 31 个字节。

5.44. CMD_PPPOE_ENABLE

该命令用于启动或者停止 PPPPOE，需要输入 1 个字节的标志码，如果标志为 1 则表示启动 PPPPOE，如果为 0 表示关闭 PPPPOE，在启动 PPPPOE 之前必须要将 CH395 初始化且设置正确的 PPPPOE 用户名和密码。

CH395 内部 PPPPOE 协议栈支持 PAP 和 CHAP 两种认证，和 PPPPOE Server 协商自动选择其中一种认证方式。

PPPPOE 成功或者失败都会产生 GINT_STAT_PPPOE 中断，单片机接收到此中断后，可以发送命令 GET_PPPOE_STATUS 来获取 PPPPOE 状态，其返回的状态码参考 5.44 章节。如果 PPPPOE 的状态为成功，则单片机可以发送命令 GET_IP_INF 来获取 IP，MASK 等信息。

此命令大约需要 20MS 执行时间，单片机可以发送 GET_CMD_STATUS 来查询是否执行完毕和执行状态。

5.45. CMD_GET_PPPOE_STATUS

此命令用于获取 PPPPOE 的状态，一般单片机在接收到 GINT_STAT_PPPOE 中断后可以发送该命令来获取 PPPPOE 的执行状态，状态码如下：

代码	名称	描述
00H	CH395_PPP_SUCCESS	成功
01H	CH395_PPPERR_PARM	参数错误
02H	CH395_PPPERR_OPEN	打开 PPP 会话失败
03H	CH395_PPPERR_DEVICE	设备错误
04H	CH395_PPPERR_ALLOC	内存分配失败
05H	CH395_PPPERR_USER	用户中断
06H	CH395_PPPERR_CONNECT	连接错误
07H	CH395_PPPERR_AUTHFAIL	挑战鉴别失败
08H	CH395_PPPERR_PROTOCOL	协议握手失败
09H	CH395_PPPERR_TIME_OUT	超时
0AH	CH395_PPPERR_CLOSE	关闭失败

5.46. CMD_SET_TCP_MSS

该命令用于设置 TCP MSS，需要输入 2 字节的 TCP MSS 值，最大值为 1460，最小值为 60。应该在初始化 CH395 之前设置此值，初始化之后不允许设置。

5.47. CMD_SET_TTL

该命令用于设置 Socket 的 TTL，需要输入 1 字节的 Socket 索引和 1 字节的 TTL 值。应该在打开 Socket 之后进行设置，最大值为 128。

5.48. CMD_SET_RECV_BUF

该命令用于设置 Socket 的接收缓冲区，需要输入 3 字节数据，第一个字节为 Socket 索引，第二个字节为缓冲区的起始块，第三个字节为块数。

第 0 块	第 1 块	第 2 块	第 46 块	第 47 块
-------	-------	-------	-------	--------	--------

CH395 内部缓冲区结构如上图，一共由 48 个块组成，每个块的长度为 512 字节。单片机可以自由分配每个 Socket 接收缓冲区的大小。CH395 初始化完成后对缓冲区的分配如下：

Socket	缓冲区	起始块	块数
0	接收缓冲区	0	8
	发送缓冲区	8	4
1	接收缓冲区	12	8
	发送缓冲区	20	4
2	接收缓冲区	24	8
	发送缓冲区	32	4
3	接收缓冲区	36	8
	发送缓冲区	44	4
4	接收缓冲区	NULL(空, 未分配, 下同)	0
	发送缓冲区	NULL	0
5	接收缓冲区	NULL	0
	发送缓冲区	NULL	0
6	接收缓冲区	NULL	0

	发送缓冲区	NULL	0
7	接收缓冲区	NULL	0
	发送缓冲区	NULL	0

由上表可以看出, CH395 初始化完成后, 将所有的缓冲区分配给 Socket0 - 3, 接收缓冲区为 8 块 (4KB), 发送缓冲区为 4 块 (2KB)。如果单片机需要用到的 Socket 个数大于 4, 需要重新对缓冲区进行分配。

5.49. CMD_SET_SEND_BUF

该命令用于设置 Socket 的发送缓冲区, 需要输入 3 字节数据, 第 1 个字节为 Socket 索引, 第 2 个字节为缓冲区的起始块, 第 3 个字节为块数。

关于缓冲区的定义和分配请参考 5.47 章节。

5.50. CMD_EEPROM_ERASE

该命令用于擦除 EEPROM, CH395 芯片自带 4KB 的 EEPROM, 擦除后数据全部为 0xFF。在对 EEPROM 写操作前必须保证目的区域的数据全部为 0xFF。

5.51. CMD_EEPROM_WRITE

该命令用于写 EEPROM, 需要输入 2 个字节的地址、1 个字节的长度和若干字节的数据流, 字节流的长度不得大于 64 字节。

5.52. CMD_EEPROM_READ

该命令用于读 EEPROM, 需要输入 2 个字节的地址和 1 个字节的长度, CH395 根据长度输出若干字节的数据流, 长度值不得大于 64 字节。外部单片机在发送完长度后, 应该等待 1MS 再读取数据。

5.53. CMD_READ_GPIO_REG

该命令用于读 GPIO 寄存器, 需要输入 1 个字节的寄存器地址, CH395 输出 1 个字节的寄存器值。所有的寄存器都是 8 位, bit0-7 分别对应 GPIO 0-7, 其地址及含义入下表:

地址	名称	描述
80H	GPIO_DIR_REG	方向寄存器, 1: 输出; 0: 输入
81H	GPIO_IN_REG	输入数据寄存器
82H	GPIO_OUT_REG	输出数据寄存器
83H	GPIO_CLR_REG	0: 保持; 1: 清除
84H	GPIO_PU_REG	下拉寄存器, 1: 上拉使能; 0: 上拉禁止
85H	GPIO_PD_REG	上拉寄存器, 1: 下拉使能; 0: 下拉禁止

CH395 有 8 路 GPIO, GPIO3 默认为输出, 在 CH395 初始化成功后, 输出低电平, 其余 GPIO 默认为输入状态。

5.54. CMD_WRITE_GPIO_REG

该命令用于写 GPIO 寄存器, 需要输入 1 个字节的寄存器地址和 1 个字节的寄存器值。

5.55. CMD_SET_FUN_PARA

该命令用于设置功能参数，需要输入 4 字节的参数，参数各位表示的含义如下：

Bit	名称	复位值	描述
0	-	-	保留, 必须为 0
1	FUN_PARA_FLAG_TCP_SERVER	0	TCP 服务器多连接模式使能位, 0x44 及以后版本支持
2	FUN_PARA_FLAG_LOW_PWR	0	低能耗模式使能位, 0x44 及以后版本支持
3	FUN_PARA_FLAG_SOCKET_CLOSE	0	Socket 关闭模式, 0x46 及以后版本支持
4	FUN_PARA_FLAG_DISABLE_SEND_OK	0	禁用 Socket SEND_OK 中断, 0x46 及以后版本支持
5:31			保留

FUN_PARA_FLAG_TCP_SERVER: TCP 服务器多连接模式，此位为 1，TCP 服务器可以连接多个客户端，使用方法请参考 8.3.6，全局配置，对所有的 Socket 有效。

FUN_PARA_FLAG_LOW_PWR: 低能耗模式，此位为 1，CH395 进入低能耗模式。CH395 工作模式分为标准模式和低能耗模式，低能耗模式的工作电流比标准模式约少 50mA，适用于对功耗要求较高的场合。

FUN_PARA_FLAG_SOCKET_CLOSE: 本参数主要用于 TCP，此位为 0 表示在产生 SINT_STAT_DISCONNECT 或者 SINT_STAT_TIM_OUT 中断后，CH395 会主动关闭 Socket。此位为 1 表示在产生上述两个中断后由外部单片机来关闭 Socket。关闭 Socket 就意味着 CH395 会将此 Socket 内部缓冲区和一些相关变量清除，在某些应用中，如果 CH395 内部自行关闭 Socket 可能会导致残留数据被强行清除，这种情况下可以由单片机可以将数据读出来，然后再关闭 Socket。

FUN_PARA_FLAG_DISABLE_SEND_OK: 禁用 SINT_STAT_SEND_OK 中断。此位为 1，CH395 则不在产生 SINT_STAT_SEND_OK 中断。

注意，CMD_SET_FUN_PARA 命令必须在初始化之前设置，且一旦初始化完毕后，不可以再随意设置。另外需要注意参数适用的版本号。

5.56. CMD_SET_KEEP_LIVE_IDLE

该命令用于设置 KeepLive 空闲时间，需要输入 4 字节的时间值，单位为毫秒。该命令仅用于 TCP 连接。KeepLive 空闲时间是指 TCP 连接无数据收发后至发送 KeepLive 数据包的时间。默认值为 20000。

5.57. CMD_SET_KEEP_LIVE_INTVL

该命令用于设置 KeepLive 超时时间，需要输入 4 字节的时间值，单位为毫秒。该命令仅用于 TCP 连接。KeepLive 超时时间是指 KeepLive 数据包发送后等待应答的时间。默认值为 15000。

5.58. CMD_SET_KEEP_LIVE_CNT

该命令用于设置 KeepLive 超时次数，需要输入 1 字节的超时次数。该命令仅用于 TCP 连接。KeepLive 超时次数是指允许 KeepLive 数据包连续无应答的最大次数。默认值为 9。

假设 KeepLive 空闲时间为 IDLE，超时时间为 INTVL，超时次数为 CNT。

如果 Socket 连接启动了 KeepLive 功能，则当 TCP 连接空闲（无数据收发）IDLE 毫秒时，Socket 会启动发送 KeepLive 包，如果远端在 INTVL 毫秒内应答 ACK，则认为该连接正常。否则 Socket 在 INTVL 毫秒后认为超时，再次启动发送 KeepLive 包，如果 CNT 次内都没有收到 ACK 包，则认为当前连接断

开, 会产生 SINT_STAT_TIM_OUT 或 SINT_STAT_DISCONNECT 中断。

如果需要配置 IDLE, INTVL, CNT 三个变量, 需要注意 IDLE 必须大于 INTVL, 且均为 500 的倍数。

5.59. CMD_SET_KEEP_LIVE_SN

该命令用于开启或者关闭 Socket 的 KeepLive 功能, 需要输入 1 字节的 Socket 索引和 1 字节的配置值, 配置值为 0 表示关闭 Socket 的 KeepLive 功能, 1 为开启。默认为关闭。

当 Socket 为 TCP 客户端时, 创建 Socket 后使用此命令打开 KeepLive 功能。

当 Socket 为 TCP 服务器时, 在产生 SINT_STAT_CONNECT 后使用此命令打开 KeepLive 功能。

6、功能说明

6.1. 单片机通讯接口

CH395 与单片机之间支持三种通讯接口: 8 位并行接口、SPI 同步串行接口、异步串口。在芯片上电复位时, CH395 将采样 SEL 和 TXD 引脚的状态, 根据这 2 个引脚状态的组合选择通讯接口, 参考下表 (表中 X 代表不关心此位, 0 代表低电平, 1 代表高电平或者悬空)。

SEL 引脚	TXD 引脚	选择通讯接口
1	1	异步串口
1	0	SPI 接口
0	1	8 位并口
0	0	错误接口

CH395 芯片 INT# 引脚输出的中断请求默认是低电平有效, 可以连接到单片机的中断输入引脚或普通输入引脚, 单片机可以使用中断方式或查询方式获知 CH395 的中断请求。

6.2. 并行接口

并口信号线包括: 8 位双向数据总线 D7~D0、读选通输入引脚 RD#、写选通输入引脚 WR#、片选输入引脚 PCS# 以及地址输入引脚 A0。CH395 芯片的 PCS# 由地址译码电路驱动, 用于当单片机具有多个外围器件时进行设备选择。通过被动并行接口, CH395 芯片可以很方便地挂接到各种 8 位单片机、DSP、MCU 的系统总线上, 并且可以与多个外围器件共存。

对于类似 Intel 并口时序的单片机, CH395 芯片的 RD# 引脚和 WR# 引脚可以分别连接到单片机的读选通输出引脚和写选通输出引脚。对于类似 Motorola 并口时序的单片机, CH395 芯片的 RD# 引脚应该接低电平, 并且 WR# 引脚连接到单片机的读写方向输出引脚 R/-W。

下表为并口 I/O 操作的真值表 (表中 X 代表不关心此位, Z 代表 CH395 三态禁止)。

PCS#	WR#	RD#	A0	D7-D0	对 CH395 芯片的实际操作
1	X	X	X	X/Z	未选中 CH395, 不进行任何操作
0	1	1	X	X/Z	虽然选中但无操作, 不进行任何操作
0	0	1/X	1	输入	向 CH395 的命令端口写入命令码
0	0	1/X	0	输入	向 CH395 的数据端口写入数据
0	1	0	0	输出	从 CH395 的数据端口读出数据
0	1	0	1	输出	从 CH395 的命令端口读取接口状态: 位 7 是中断标志, 低有效, 等效于 INT# 引脚

CH395 芯片占用两个地址位, 当 A0 引脚为高电平时选择命令端口, 可以写入新的命令, 或者读出接口状态; 当 A0 引脚为低电平时选择数据端口, 可以读写数据。

单片机通过 8 位并口对 CH395 芯片进行读写, 所有操作都是由一个命令码、若干个输入数据和若干个输出数据组成, 部分命令不需要输入数据, 部分命令没有输出数据。命令操作步骤如下:

- ① 单片机在 A0=1 时向命令端口写入命令代码;
- ② 如果该命令具有输入数据, 那么在 A0=0 时依次写入输入数据, 每次一个字节;
- ③ 如果该命令具有输出数据, 那么在 A0=0 时依次读取输出数据, 每次一个字节;
- ④ 命令完成, 有些命令在执行完成后会需要查询执行, 单片机可以暂停或者转到①继续执行下一个命令。

6.3. SPI 串行接口

SPI 同步串行接口信号线包括: SPI 片选输入引脚 SCS、串行时钟输入引脚 SCK、串行数据输入引脚 SDI、串行数据输出引脚 SDO。通过 SPI 串行接口, CH395 可以用较少的连线挂接到各种单片机、DSP、MCU 的 SPI 串行总线上, 或者进行较远距离的点对点连接。

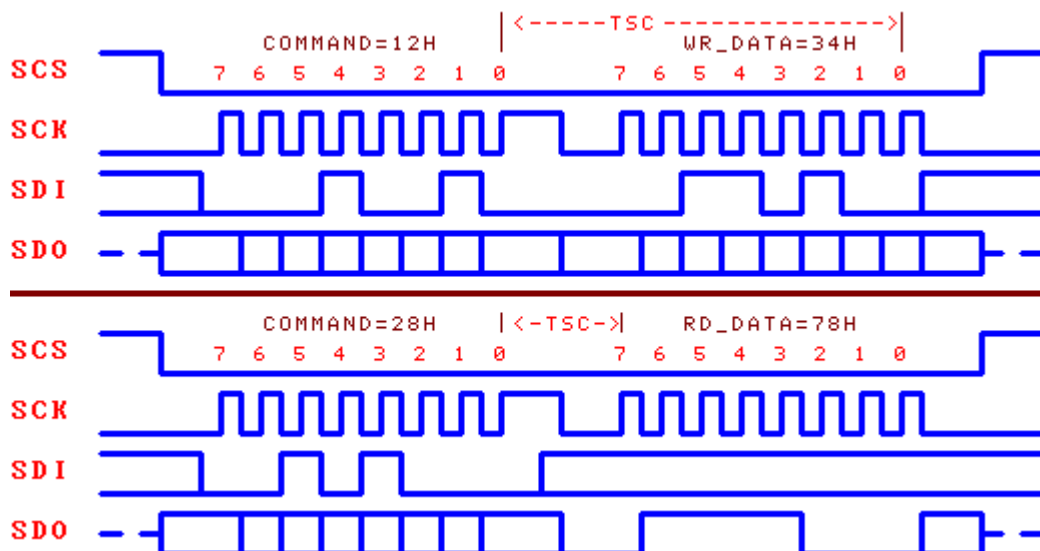
CH395 芯片的 SCS 引脚由单片机的 SPI 片选输出引脚或者普通输出引脚驱动, SCK 引脚由单片机的 SPI 时钟输出引脚 SCK 驱动, SDI 引脚由单片机的 SPI 数据输出引脚 SDO 或 MOSI 驱动, SDO 引脚则连接到单片机的 SPI 数据输入引脚 SDI 或 MISO。对于硬件 SPI 接口, 建议 SPI 设置是 CPOL=CPHA=0 或者 CPOL=CPHA=1, 并且数据位顺序是高位在前 MSB first。CH395 的 SPI 接口也支持单片机用普通 I/O 引脚模拟 SPI 接口进行通讯。

CH395 的 SPI 接口支持 SPI 模式 0 和 SPI 模式 3, CH395 总是从 SPI 时钟 SCK 的上升沿输入数据, 并在允许输出时从 SCK 的下降沿输出数据, 数据位顺序是高位在前, 计满 8 位为一个字节。

SPI 的操作步骤是:

- ① 单片机产生 CH395 芯片的 SPI 片选, 低电平有效;
- ② 单片机按 SPI 输出方式发出一个字节的的数据, CH395 总是将 SPI 片选 SCS 有效后收到的首个字节当做命令码, 后续字节当做数据;
- ③ 单片机延时 TSC 时间 (约 1.5uS) 等待 CH395 的 SPI 接口空闲;
- ④ 如果是写操作, 单片机向 CH395 发出一个字节的待写数据, 等待 SPI 接口空闲后, 单片机继续发出若个字节的待写数据, CH395 依次接收, 直到单片机禁止 SPI 片选;
- ⑤ 如果是读操作, 单片机从 CH395 接收一个字节的的数据, 等待 SPI 接口空闲后, 单片机继续从 CH395 接收若个字节的的数据, 直到单片机禁止 SPI 片选;
- ⑥ 单片机禁止 CH395 芯片的 SPI 片选, 以结束当前 SPI 操作。

下图是 SPI 接口的逻辑时序图, 前一个是发出命令 12H 并写入数据 34H, 后一个是发出命令 28H 并读出数据 78H。



6.4. 异步串口

异步串口信号线包括: 串行数据输入引脚 RXD 和串行数据输出引脚 TXD。通过串行接口, CH395 可以用最少的连线与单片机、DSP、MCU 进行较远距离的点对点连接。

CH395 芯片的 RXD 和 TXD 可以分别连接到单片机的串行数据输出引脚和串行数据输入引脚。

CH395 的串行数据格式是标准的字节传输模式，包括 1 个起始位、8 个数据位、1 个停止位。

CH395 既支持硬件设定默认的串行通讯波特率，也支持单片机随时通过 `CMD_SET_BAUDRATE` 命令选择合适的通讯波特率。每次上电复位后，CH395 默认的串行通讯波特率由 `SD0`、`SDI`、`SCK` 三个引脚的电平组合设定，参考下表（表中 0 代表低电平，1 代表高电平或者悬空）。

SD0 引脚	SDI 引脚	SCK 引脚	上电复位后默认的串行通讯波特率
1	1	1	9600 bps
1	1	0	57600 bps
1	0	1	115200 bps
1	0	0	460800 bps
0	1	1	250000 bps
0	1	0	1000000 bps
0	0	1	3000000 bps
0	0	0	921600 bps

为了区分命令码和数据，CH395 要求单片机先通过串口发送两个同步码字节（57H 和 ABH），然后再发送命令码，接着是发送数据或者接收数据。CH395 会检查上述两个同步码字节之间、同步码与命令码之间的间隔时间，如果间隔时间大于串口输入超时时间 `SER_CMD_TIMEOUT`（约 40ms），那么 CH395 将丢弃该同步码及命令包。串口命令操作步骤如下：

- ① 单片机通过串口向 CH395 发出第一同步码 57H；
- ② 单片机向 CH395 发出第二同步码 0ABH；
- ③ 单片机向 CH395 发出命令码；
- ④ 如果该命令具有输入数据，那么依次向 CH395 发出输入数据，每次一个字节；
- ⑤ 如果该命令具有输出数据，那么依次从 CH395 接收输出数据，每次一个字节；
- ⑥ 命令完成，有些命令在执行完成后会产生中断通知并通过串口直接发出中断状态码，单片机可以暂停或者转到①继续执行下一个命令。

6.5. 其它硬件

CH395 芯片内部集成了 10/100M 以太网 MAC 和 PHY、CRC 数据校验、被动并行接口、SPI-Slave 控制器、异步串口、SRAM 高速 MCU、固件程序、晶体振荡器和 PLL 倍频器、电源上电复位电路等。

CH395 芯片的 `ELINK#` 引脚用于以太网状态连接与通讯指示，可以外接串了限流电阻的发光二极管 LED，用于指示连接和通讯状态。

CH395 芯片的 `RXP`、`RXN`、`TXP` 和 `TXN` 为以太网的信号线。CH395 的 PHY 支持 MDI/MDIX 线路自动转换，但是仅在自动协商模式下有效。

CH395 芯片内置了电源上电复位电路，一般情况下，不需要外部提供复位。`RSTI` 引脚用于从外部输入异步复位信号；当 `RSTI` 引脚为低电平时，CH395 芯片被复位；当 `RSTI` 引脚恢复为高电平后，CH395 会继续延时复位 35ms 左右，然后进入正常工作状态。为了在电源上电期间可靠复位并且减少外部干扰，可以在 `RSTI` 引脚与地之间跨接一个容量为 0.1μF 左右的电容。

CH395 芯片的 `RST` 引脚是高电平有效的复位状态输出引脚，可以用于向外部单片机提供上电复位信号。当 CH395 电源上电复位或者被外部强制复位以及复位延时期间，`RST` 引脚输出高电平；CH395 复位完成并且通讯接口初始化完成后，`RST` 引脚恢复到低电平。

CH395 芯片正常工作时需要外部为其提供 30MHz 的时钟信号。CH395 芯片内置了晶体振荡器和振荡电容，一般情况下，时钟信号由 CH395 内置的振荡器通过晶体稳频振荡产生，外围电路只需要在 `XI` 和 `XO` 引脚之间连接一个标称频率为 30MHz 的晶体。

7、参数

7.1. 绝对最大值 (临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏)

名称	参数说明	最小值	最大值	单位
TA	工作时的环境温度 VCC33=3.3V VCC18=1.8V	-40	85	°C
TS	储存时的环境温度	-55	125	°C
VCC33	电源电压 (VCC33 接电源, GND 接地)	-0.4	4.2	V
VCC18	电源电压 (VCC18 接电源, GND 接地)	-0.4	2.3	V
VI0	输入或者输出引脚上的电压	-0.4	VCC33+0.4	V
VI05	支持 5V 耐压的输入或者输出引脚上的电压	-0.4	5.4	V

7.2. 电气参数 (测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V)

名称	参数说明	最小值	典型值	最大值	单位	
VCCxx	电源电压	VCC33	2.7	3.3	3.6	V
		VCC18	1.65	1.8	1.95	
ICC	工作时的总电源电流 VCC33=3.3V		90(50)	220(195)	mA	
ISLP	低功耗状态的电源电流 I/O 引脚悬空/内部下拉 VCC33=3.3V		15		mA	
VIL	低电平输入电压	-0.4		0.7	V	
VIH	高电平输入电压	2.0		VCC33+0.4	V	
VOL	低电平输出电压 (4mA 吸入电流)			0.4	V	
VOH	高电平输出电压 (4mA 输出电流)	VCC33-0.4			V	
IUP	内置上拉电阻的输入端的输入电流	20	40	100	uA	
IDN	内置下拉电阻的输入端的输入电流	-20	-40	-100	uA	
VR	电源上电复位的电压门限	1.4	1.5	2.5	V	

注: 括号内的参数为低能耗模式。

7.3. 时序参数 (测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V, 参考附图)

名称	参数说明	最小值	典型值	最大值	单位
FCLK	XI 引脚的输入时钟的频率	29.995	30.00	30.005	MHz
TPR	内部电源上电的复位时间	15	50	60	nS
TRI	外部复位输入的有效信号宽度	100			nS
TRD	外部复位输入后的复位延时	15	50	60	nS
TWAK	从低功耗状态退出的唤醒时间	3	7	15	nS
TE1	CMD_RESET_ALL 命令的执行时间	10	50	60	nS
TE2	CMD_INIT_CH395 命令的执行时间	300	350	400	nS
TE3	需要获取状态的命令的执行时间		1	5	nS
TE4	CMD_SET_BAUDRATE 命令的执行时间	200	1000	2000	uS
TE0	其余命令的执行时间		0.8	1.5	uS
TSX	命令码与命令码之间的间隔时间	1.5			uS
TSC	命令码与数据之间的间隔时间	0.6			uS

TSD	数据与数据之间的间隔时间	0.3			uS
TINT	收到 GET_STATUS 命令到 INT#引脚撤消中断		1	2	uS

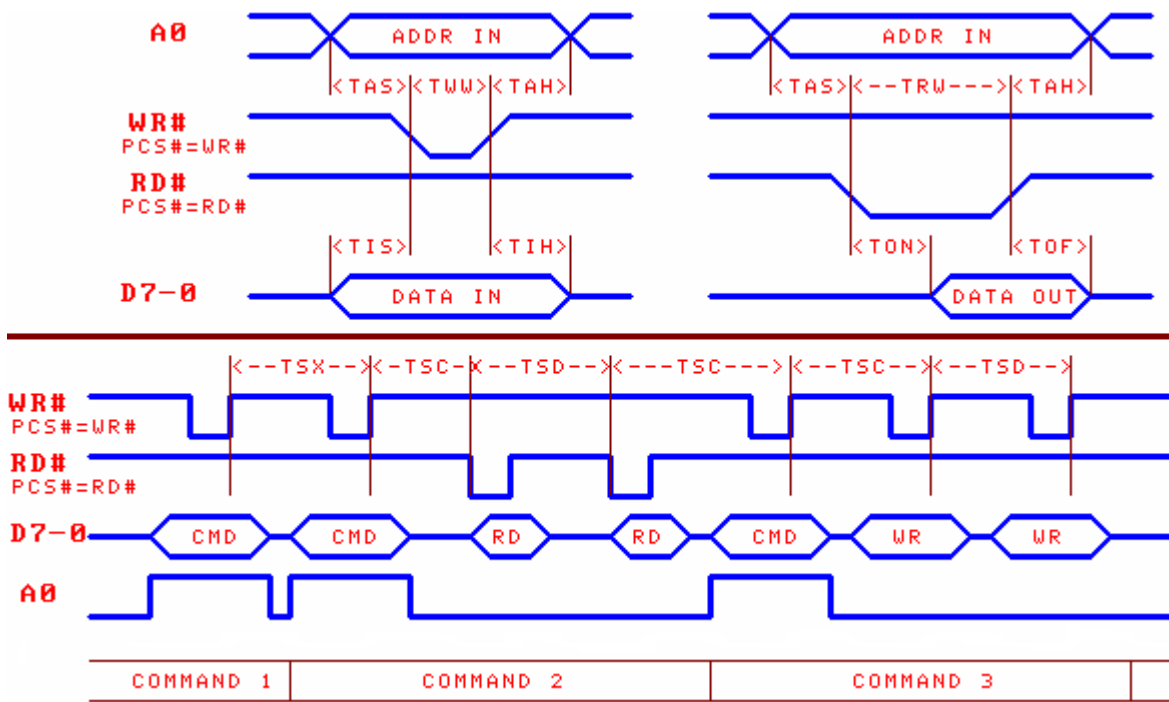
7.4. 并口时序参数 (测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V, 参考附图)

(RD 是指 RD#信号有效并且 PCS#信号有效, RD#=PCS#=0 执行读操作)

(WR 是指 WR#信号有效并且 PCS#信号有效, WR#=PCS#=0 执行写操作)

(低能耗模式下, 表中所有的参数乘以 2)

名称	参数说明	最小值	典型值	最大值	单位
TWW	有效的写选通脉冲 WR 的宽度	30			nS
TRW	有效的读选通脉冲 RD 的宽度	35			nS
TAS	RD 或 WR 前的地址输入建立时间	3			nS
TAH	RD 或 WR 后的地址输入保持时间	3			nS
TIS	写选通 WR 前的数据输入建立时间	0			nS
TIH	写选通 WR 后的数据输入保持时间	3			nS
TON	读选通 RD 有效到数据输出有效	2	7	15	nS
TOF	读选通 RD 无效到数据输出无效	3	12	25	nS

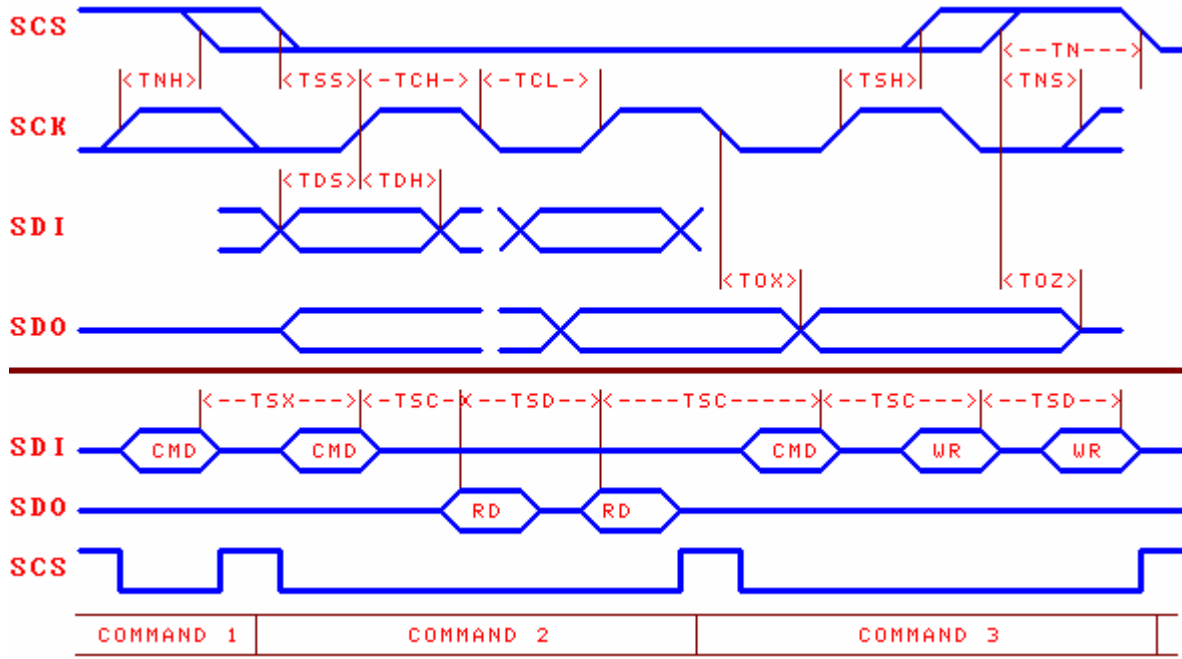


7.5. SPI 时序参数 (测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V, 参考附图)

(低能耗模式下, 表中所有的参数乘以 2)

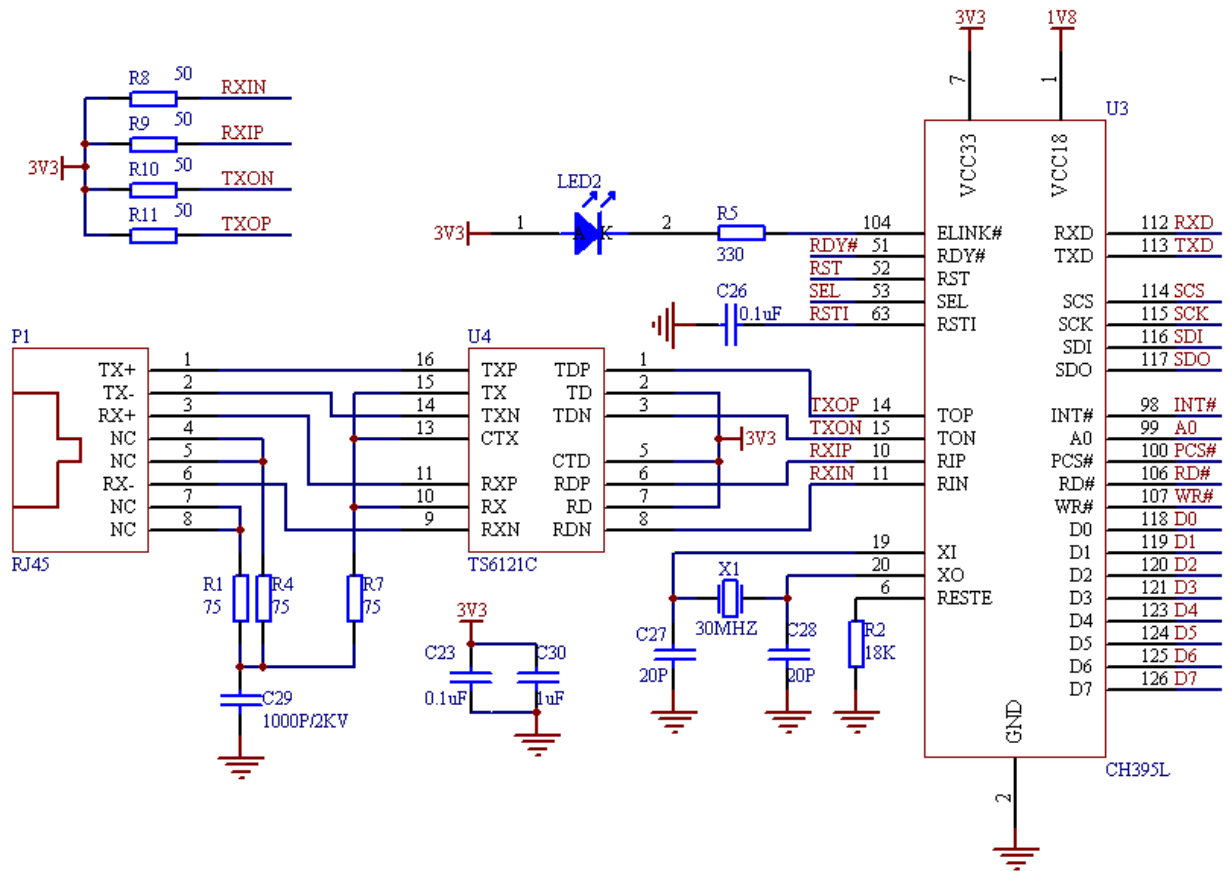
名称	参数说明	最小值	典型值	最大值	单位
TSS	SCK 上升沿之前 SCS 有效的建立时间	25			nS
TSH	SCK 上升沿之后 SCS 有效的保持时间	12			nS
TNS	SCK 上升沿之前 SCS 无效的建立时间	12			nS
TNH	SCK 上升沿之后 SCS 无效的保持时间	12			nS
TN	SCS 无效时间 (SPI 操作间隔时间)	60			nS

TCH	SCK 时钟的高电平时间	15			nS
TCL	SCK 时钟的低电平时间	15			nS
TDS	SCK 上升沿之前 SDI 输入的建立时间	4			nS
TDH	SCK 上升沿之后 SDI 输入的保持时间	2			nS
TOX	SCK 下降沿到 SDO 输出改变	2	6	12	nS
TOZ	SCS 无效到 SDO 输出无效	3	10	18	nS



8、应用

8.1. 硬件电路设计



如果需要将 CH395 配置为 8 位并口通讯方式 PARALLEL, 那么 SEL 引脚应该接 GND, TXD 引脚悬空。该接口与外围单片机通讯使用的引脚有 A0、PCS#、RD#、WR#、D0-D7、INT#(可选)、RSTI(可选)。

如果需要将 CH395 配置为 SPI 串行通讯方式 SPI, 那么 TXD 引脚应该接 GND, SEL 引脚悬空。该接口与外围单片机通讯使用的引脚有 SCS、SCK、SD0、SDI、INT#、RSTI(可选)。

如果需要将 CH395 配置为异步串口通讯方式 UART/SERIAL, 那么 SEL 引脚和 TXD 引脚应该都悬空。该接口与外围单片机通讯使用的引脚有 TXD、RXD、INT#、RSTI(可选)。默认的串口通讯波特率由 SD0、SDI、SCK 三个引脚设定。如果需要动态修改 CH395 串口的通讯波特率, 那么建议由单片机的 I/O 引脚控制 CH395 的 RSTI 引脚, 便于在必要时复位 CH395 以恢复到默认的通讯波特率。

由于 INT#引脚和 TXD 引脚在 CH395 复位期间只能提供微弱的高电平输出电流, 在进行较远距离的连接时, 为了避免 INT#或者 TXD 在 CH395 复位期间受到干扰而导致单片机误操作, 可以在 INT#引脚或者 TXD 引脚上加阻值为 $2\text{K}\Omega \sim 5\text{K}\Omega$ 的上拉电阻, 以维持较稳定的高电平。在 CH395 芯片复位完成后, INT#引脚和 TXD 引脚将能够提供 4mA 的高电平输出电流或者 4mA 的低电平吸入电流。

8 位并口方式下通过查询 CH395 的状态端口 (即命令端口) 获取接口状态, 位 7 是中断标志位, 低有效, 等效于查询 INT#引脚, 位 7 为 0 时说明有中断请求。在 SPI 和 UART 方式下, 必须通过 INT#来获取中断。

P1 为 RJ45 端口, 用于连接交换机、路由器等网络设备。含有两对以太网差分信号, RJ45 未使用的引脚, 应通过 $75\ \Omega$ 电阻对地接 $1000\text{P}/2\text{KV}$ 电容。

U4 为网络变压器, 主要有电气隔离、阻抗匹配等功能, 内侧 (靠近 CH395 一侧) 中心抽头应接 3.3V , 外侧 (靠近 RJ45 一侧) 未使用的引脚应通过 $75\ \Omega$ 电阻对地接 $1000\text{P}/2\text{KV}$ 电容。

R2 为以太网信号调节电阻, 可选 $12\text{K} \sim 18\text{K}$ 的电阻, 12K 最优, 18K 功耗最小。

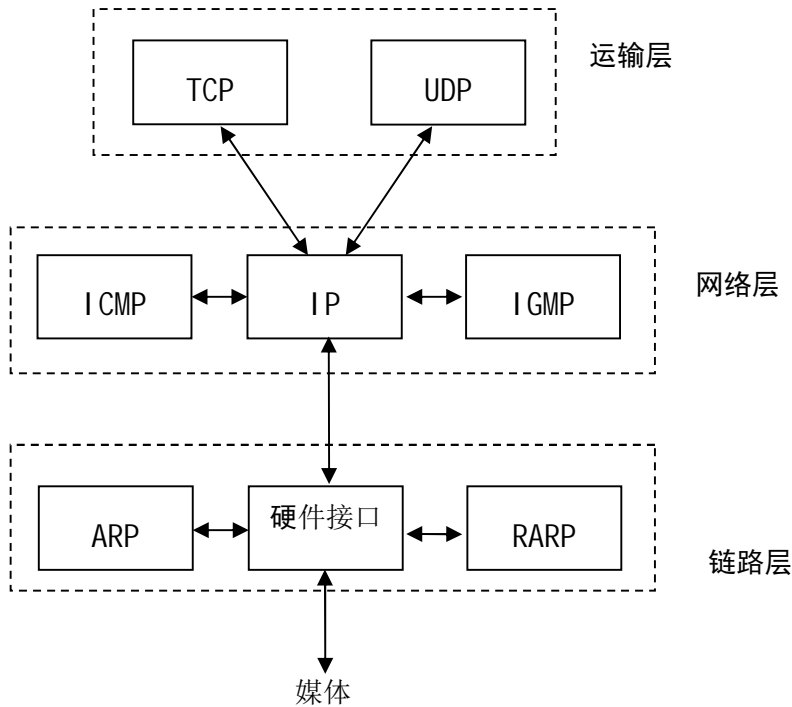
本图中没有给出 GPIO, 具体可以参考 CH395EVT 的原理图。

实际制作印刷电路板时 (PCB), R8-R11, C23, C30 应尽量靠近 U4 的第 5 脚。TXOP(RXIP)与 TXON(RXIN)

为差分信号，布线时应贴近平行走线，尽量在两侧提供地线或者覆铜，减少来自外界的干扰。尽量缩短晶体 XI 和 XO 相关信号的长度，为了减少高频时钟对外界的干扰，可以在相关元器件周边环境绕底线或者覆铜。

8.2. 应用基础

CH395 内部集成了 IPv4、ARP、ICMP、IGMP、UDP、TCP 等协议，其关系图如下：



TCP 和 UDP 是两种比较重要的传输层协议，两者都使用 IP 作为网络层协议。

TCP 是一种面向连接的传输，能够提供可靠的字节流传输服务。

UDP 是一种简单的面向数据报的传输层协议，与 TCP 不同的是 UDP 无法保证数据报文准确达到目的地。

TCP 为网络设备提供了高可靠性的通讯，它所做的工作包括把应用程序交给他的数据分成合适的小块交给下面的网络层，确认接收到的分组，设置超时时钟等，由于运输层提供了高可靠性的端到端的通信，应用层客户忽略所有细节。而 UDP 则为应用层提供一种非常简单的服务，速度较 TCP 快，它只是把数据报从一个网络终端发送到另一个网络终端，但是并不保证该数据报能够达到另一端，任何必需的可靠性都必须由应用层来提供。

IP 是网络层上的协议，同时被 TCP 和 UDP 使用，TCP 和 UDP 的每组数据都通过 IP 层在网络中进行传输。

ICMP 是 IP 协议的附属协议，IP 层用它来与其他主机或者路由器交换错误报文或者其他重要信息，例如 CH395 产生不可达中断，就是通过 ICMP 来进行错误报文交换的。PING 也使用了 ICMP 协议。

IGMP 是 Internet 组管理协议，主要用来把一个 UDP 数据报多播到多个主机。

ARP 为地址解析协议，用来转换 IP 层和网络接口层使用的地址。

关于以太网帧、IP、UDP、TCP 报文等基本格式可以参考 8.3 章节。

8.3. 应用参考步骤

本章节介绍常用的操作步骤，具体可以参考例子程序。

8.3.1. 初始化 CH395，必要操作

- ① 发送命令 `CMD_SET_MAC_ADDR` 设置 CH395 的 MAC 地址；
- ② 发送命令 `CMD_SET_IP_ADDR` 设置 CH395 的 IP 地址；
- ③ 发送命令 `CMD_SET_GWIP_ADDR` 设置 CH395 的网关 IP 地址；
- ④ 发送命令 `CMD_SET_MASK_ADDR` 设置 CH395 的子网掩码；
- ⑤ 发送命令 `CMD_INIT_CH395` 对 CH395 初始化；

⑥ 延时 2MS 以上发送命令 `CMD_GET_CMD_STATUS` 获取 `CMD_INIT_CH395` 执行状态，如果返回 `CH395_ERR_BUSY` 表示 CH395 内部正在执行命令，需要再次执行⑥；如果返回 `CH395_ERR_SUCCESS` 表示命令执行成功。`CMD_INIT_CH395` 一般需要 350ms 才能执行完毕。

步骤①一般不需要，CH395 在出厂时，已经烧录了 IEEE 分配的 MAC 地址。

如果需要启动 DHCP 或者 PPPoE，则步骤②-④不需要。

上述步骤④为可选操作，默认子网掩码为 255.255.255.0。一般不需要设置。

CH395 接收到 `CMD_INIT_CH395` 命令后，初始化内部 TCP/IP 协议栈，初始化 MAC 和 PHY，此时 MAC 和 PHY 会被初始化为自动协商模式。如果需要 PHY 工作在其他模式，如 10M 全双工模式，则需要先在 `CMD_INIT_CH395` 成功执行之后，发送 `CMD_SET_PHY` 命令进行设置。

8.3.2. 初始化 Socket 为 MACRAW 模式

初始化步骤如下：

- ① 发送命令 `CMD_SET_PROTO_TYPE_SN` 设置 Socket 工作在 MACRAW 模式；
- ② 发送命令 `CMD_OPEN_SOCKET_SN` 打开 Socket；

③ 延时 2MS 以上发送命令 `CMD_GET_CMD_STATUS` 获取 `CMD_OPEN_SOCKET_SN` 执行状态，如果返回 `CH395_ERR_BUSY` 表示 CH395 内部正在执行命令，需要再次执行③；如果返回 `CH395_ERR_SUCCESS` 表示命令执行成功。其他值表示打开 Socket 失败。

IEEE802.3 以太网帧格式：

目的 MAC	源 MAC	类型	数据	CRC32
6 Byte	6 Byte	2 Byte	46-1500 Byte	4 Byte

在 MACRAW 模式下，CH395 会透明传输以太网和单片机之间的数据，不会对数据进行 TCP/IP 封装，CH395 接收数据时会以以太网冗余校验 CRC32 进行校验，如果校验错误，数据包不会转发给单片机。CH395 发送数据时会在数据包尾部加入以太网冗余校验 CRC32。单片机每次向 CH395 写入的数据长度不得大于 1514，CH395 会将单片机每次写入的数据封装成一帧数据进行发送。当 CH395 从以太网收到数据后会通知单片机，此时单片机应立即将所有数据从 CH395 内部接收缓冲区读走。

仅 Socket0 可以设置此模式，且其他 Socket 将不可用。

8.3.3. 初始化 Socket 为 IPRAW 模式

初始化步骤如下：

- ① 发送命令 `CMD_SET_PROTO_TYPE_SN` 设置 Socket 工作在 IPRAW 模式；
- ② 发送命令 `CMD_SET_IP_ADDR_SN` 设置目的 IP 地址；
- ③ 发送命令 `CMD_SET_IPRAW_PRO_SN` 设置协议字段；
- ④ 发送命令 `CMD_OPEN_SOCKET_SN` 打开 Socket；

⑤ 延时 2MS 以上发送命令 `CMD_GET_CMD_STATUS` 获取 `CMD_OPEN_SOCKET_SN` 执行状态，如果返回 `CH395_ERR_BUSY` 表示 CH395 内部正在执行命令，需要再次执行⑤；如果返回 `CH395_ERR_SUCCESS` 表示命令执行成功，其他值表示打开 Socket 失败。

IP 报文结构：

目的 MAC	源 MAC	类型	IP 首部	IPRAW 数据	CRC32
6 Byte	6 Byte	2 Byte	20 Byte	最大 1480 Bytes	4 Byte

单片机向 CH395 写入若干字节数据流后，CH395 将此 Socket 的协议字段封装在 IP 首部，并将数据流封装在 IPRAW 数据部分进行发送。IPRAW 一包允许发送的最大长度为 1480 字节，如果单片机写入的数据流长度大于 1480 字节，CH395 会将数据流封装成若干个 IP 包进行发送且每包成功发送后都会产生 SINT_STAT_SEND_OK 中断。单片机每次写入的字节长度不得大于发送缓冲区长度，且收到 SINT_STAT_SENBUF_FREE 中断后才可以进行下一次写数据。如果产生 SINT_STAT_TIM_OUT 中断表示数据发送失败，导致发送数据失败一般有以下两个原因：

- ① 如果目的 IP 地址和 CH395 在同一个子网，则可能目的 IP 地址的网络设备不在线。
- ② 如果目的 IP 地址和 CH395 不在同一个子网，则可能 CH395 的网关不在线。

当 CH395 收到 IP 数据包后，首先检测协议字段和 Socket 设置的协议字段是否相同，如果相同则将 IPRAW 数据包复制到接收缓冲区中并产生 SINT_STAT_RECV 中断，单片机收到此中断后，可以发送命令 CMD_GET_RECV_LEN_SN 来获取接收缓冲区数据的长度，然后发送命令 CMD_READ_RECV_SN 来读取缓冲区的数据。单片机可以一次将所有数据读出也可以分多次读取，由于 IPRAW 模式下 CH395 无法进行流控，建议单片机查询到接收数据中断口后应立即将所有数据读出，以免被后续的数据覆盖。

关于协议字段设置的注意事项

CH395 处理 IPRAW 的优先级高于 UDP 和 TCP，如果 IP 协议字段设置为 17(UDP) 或者 6(TCP)，则可能存在和其他 Socket 冲突的可能性，在使用时应当注意避免，下面列举两种情况进行说明：

- ① Socket0 设置为 IPRAW 模式，IP 协议字段为 17，Socket1 为 UDP 模式。在 UDP 模式下，IP 包的协议字段也是 17，这样就会导致 Socket1 通讯的数据会被 Socket0 拦截，无法接收到数据。
- ② Socket0 设置为 IPRAW 模式，IP 协议字段为 6，Socket1 为 TCP 模式。在 TCP 模式下，IP 包的协议字段也是 6，这样就会导致 Socket1 通讯的数据会被 Socket0 拦截，无法接收到数据。

8.3.4. 初始化 Socket 为 UDP 模式

初始化步骤如下：

- ① 发送命令 CMD_SET_PROTO_TYPE_SN 设置 Socket 工作在 UDP 模式；
- ② 发送命令 CMD_SET_IP_ADDR_SN 设置目的 IP 地址；
- ③ 发送命令 CMD_SET_DEST_PORT_SN 设置目的端口；
- ④ 发送命令 CMD_SET_SOUR_PORT_SN 设置源端口；
- ⑤ 发送命令 CMD_OPEN_SOCKET_SN 打开 Socket；
- ⑥ 延时 2MS 以上发送命令 CMD_GET_CMD_STATUS 获取 CMD_OPEN_SOCKET_SN 执行状态，如果返回 CH395_ERR_BUSY 表示 CH395 内部正在执行命令，需要再次执行⑥；如果返回 CH395_ERR_SUCCESS 表示命令执行成功，其他值表示打开 Socket 失败。

UDP 报文结构：

目的 MAC	源 MAC	类型	IP 首部	UDP 首部	UDP 数据	CRC32
6 Byte	6 Byte	2 Byte	20 Byte	8 Byte	最大 1472 Bytes	4 Byte

UDP 是一个简单的，不可靠的，面向数据报文的运输层协议，传输速度较快，不能保证数据能达到目的地，必须由应用层来保证传输的可靠稳定。

单片机向 CH395 写入若干字节数据流后，CH395 数据流封装在 UDP 数据部分进行发送。UDP 一包可以发送的最大长度为 1472 字节，如果单片机写入的数据流长度大于 1472 字节，CH395 会将数据流封装成若干个 UDP 包进行发送且每包成功发送后都会产生 SINT_STAT_SEND_OK 中断。单片机每次写入的字节长度不得大于发送缓冲区长度，且收到 SINT_STAT_SENBUF_FREE 中断后才可以进行下一次写数据。如果数据发送失败会产生 SINT_STAT_TIM_OUT 中断，导致发送数据失败一般是下面两个原因：

- ① 如果目的 IP 地址和 CH395 在同一个子网，则可能目的 IP 地址的网络设备不在线。
- ② 如果目的 IP 地址和 CH395 不在同一个子网，则可能 CH395 的网关不在线。

当 CH395 接收到 UDP 报文后，将 UDP 数据复制到 Socket 接收缓冲区中并产生 SINT_STAT_RECV 中断，单片机收到此中断后，可以发送命令 CMD_GET_RECV_LEN_SN 来获取接收缓冲区数据的长度，然后发送命令 CMD_READ_RECV_SN 来读取缓冲区的数据。由于 UDP 模式下 CH395 无法提供流控，建议接收到的数据及时快速的读完，以免被后续的数据覆盖。

CH395 支持两种 UDP 模式：UDP 客户端和 UDP 服务器。UDP 客户端只能和指定的 IP 和端口进行通讯，UDP 服务器可以和任何远端的 IP 和端口进行通讯。

两者在使用上有些区别：

① 初始化步骤②，如果目的 IP 地址为 0xFFFFFFFF，则此 Socket 会进入 UDP 服务器模式，否则为 UDP 客户端模式。

② 单片机从 CH395 读取数据，客户端模式下 CH395 直接将接收到的数据流发送给单片机，单片机可以一次将所有数据读出也可以只读出部分数据。服务器模式下 CH395 会在数据的头部添加 8 个字节的信息表，单片机可以根据信息表来获得数据包的来源信息，单片机必须一次性将数据全部读出。

数据包长度	端口	IP 地址	数据
2 Byte	2 Byte	4 Byte	N Byte

③ 单片机发送数据，客户端模式下 CH395 直接将数据发送给初始化时指定的目的 IP 和端口。服务器模式下 CH395 可以将数据发送到任何 IP 和端口，单片机在发送前设置目的 IP 和目的端口即可。

8.3.5. 初始化 Socket 为 TCP 客户端模式

初始化步骤如下：

① 发送命令 `CMD_SET_PROTO_TYPE_SN` 设置 Socket 工作在 TCP 模式；

② 发送命令 `CMD_SET_IP_ADDR_SN` 设置目的 IP 地址；

③ 发送命令 `CMD_SET_DEST_PORT_SN` 设置目的端口；

④ 发送命令 `CMD_SET_SOUR_PORT_SN` 设置源端口；

⑤ 发送命令 `CMD_OPEN_SOCKET_SN` 打开 Socket；

⑥ 延时 2MS 以上发送命令 `CMD_GET_CMD_STATUS` 获取 `CMD_OPEN_SOCKET_SN` 执行状态，如果返回 `CH395_ERR_BUSY` 表示 CH395 内部正在执行命令，需要再次执行⑥；如果返回 `CH395_ERR_SUCCESS` 表示命令执行成功，其他值表示打开 Socket 失败；

⑦ 发送命令 `CMD_TCP_CONNECT_SN` 进行 TCP 连接；

⑧ 延时 2MS 以上发送命令 `CMD_GET_CMD_STATUS` 获取 `CMD_TCP_CONNECT_SN` 执行状态，如果返回 `CH395_ERR_BUSY` 表示 CH395 内部正在执行命令，需要再次执行⑧；如果返回 `CH395_ERR_SUCCESS` 表示命令执行成功。其他值表示命令执行失败。返回 `CH395_ERR_SUCCESS` 仅表示命令执行成功，不代表 TCP 连接成功。如果 TCP 连接成功，CH395 将会产生 `SINT_STAT_CONNECT` 中断。如果连接失败 CH395 将产生 `SINT_STAT_TIM_OUT` 中断，如果需要再次连接，再次从⑤开始执行。

TCP 报文结构：

目的 MAC	源 MAC	类型	IP 首部	TCP 首部	TCP 数据	CRC32
6 Byte	6 Byte	2 Byte	20 Byte	20 Byte	最大 1460 Byte	4 Byte

TCP 提供面向连接的，可靠的字节流服务。

CH395 产生 `SINT_STAT_CONNECT` 表明 TCP 连接建立，可以进行数据收发。在连接未建立前不得进行数据发送操作。

单片机向 CH395 写入若干字节数据流后，CH395 数据流封装在 TCP 数据部分进行发送。TCP 一包可以发送的最大长度为 TCP MSS 个字节，如果单片机写入的数据流长度大于 TCP MSS 字节，CH395 会将数据流封装成若干个 TCP 包进行发送且每包成功发送后都会产生 `SINT_STAT_SEND_OK` 中断。单片机每次写入的字节长度不得大于发送缓冲区长度，且收到 `SINT_STAT_SENBUF_FREE` 中断后才可以进行下一次写数据操作。在 TCP 模式下，如果数据发送失败会产生 `SINT_STAT_TIM_OUT` 中断，CH395 会自动将此 Socket 关闭(在 `FUN_PARA_FLAG_SOCKET_CLOSE` 为 0 情况下，参考 5.55)。

当 CH395 接收到 TCP 报文后，将 TCP 数据复制到 Socket 接收缓冲区中并产生 `SINT_STAT_RECV` 中断，单片机收到此中断后，可以发送命令 `CMD_GET_RECV_LEN_SN` 来获取接收缓冲区数据的长度，然后发送命令 `CMD_READ_RECV_SN` 来读取缓冲区的数据。单片机可以一次将所有数据读出也可以只读出部分数据，接收缓冲区剩余空间即为 TCP 的窗口，单片机每次读数据后，CH395 会对接收缓冲区的剩余

空间进行检查并向 TCP 服务器通告当前窗口的大小。

8.3.6. 初始化 Socket 为 TCP 服务器模式

初始化步骤如下:

① 发送命令 `CMD_SET_PROTO_TYPE_SN` 设置 Socket 工作在 TCP 模式;

② 发送命令 `CMD_SET_SOUR_PORT_SN` 设置源端口 Sport;

③ 发送命令 `CMD_OPEN_SOCKET_SN` 打开 Socket;

④ 延时 2MS 以上发送命令 `CMD_GET_CMD_STATUS` 获取 `CMD_OPEN_SOCKET_SN` 执行状态, 如果返回 `CH395_ERR_BUSY` 表示 CH395 内部正在执行命令, 需要再次执行④; 如果返回 `CH395_ERR_SUCCESS` 表示命令执行成功, 其他值表示打开 Socket 失败。

在 TCP 服务器模式下, 如果客户端进行连接, 在 Socket 一直处于监听状态, 不会产生超时中断, 如果 TCP 连接成功, CH395 会产生 `SINT_STAT_CONNECT` 中断, 此时单片机可以发送命令 `CMD_GET_REMT_IP_SN` 来获取客户端的 IP 地址和端口号。

默认的设置下, 服务器多连接功能被关闭, TCP 服务端模式下, 每个 Socket 只可以建立一个 TCP 连接。

在多连接模式使能的情况下, TCP 服务器可以连接多个 TCP 连接, 单片机需要设置 Socket 的源端口和服务器的源端口一致, 如果 TCP 服务器监听到连接, CH395 会查找当前所有的 Socket 的源端口是否和当前服务器一致, 协议类型为 TCP, 且处于被关闭状态, 如果找到则立即将此 Socket 打开, 将连接分配给此 Socket, 并通知单片机有连接事件, 如果没找到, 则将连接复位。此模式下服务器的 Socket 仅用来监听, 单片机需要分配其它 Socket 用于该服务器的连接。例如 Socket0 设置为服务器模式, Socket1 和 Socket2 用于此服务器的连接, 步骤如下:

Socket0 执行①-④;

⑤ 向 Socket1 发送命令 `CMD_SET_SOUR_PORT_SN` 设置源端口 Sport;

⑥ 向 Socket1 发送命令 `CMD_SET_PROTO_TYPE_SN` 设置 Socket 工作在 TCP 模式;

⑦ 向 Socket2 发送命令 `CMD_SET_SOUR_PORT_SN` 设置源端口 Sport;

⑧ 向 Socket2 发送命令 `CMD_SET_PROTO_TYPE_SN` 设置 Socket 工作在 TCP 模式;

关于数据结构, 发送数据和接收数据流程可以参考 TCP 客户端模式。

8.3.7. DHCP

在开始下面的步骤前, 首先要对 CH395 进行初始化。

DHCP 步骤如下:

① 发送命令 `CMD_DHCP_ENABLE` 参数为 1 启动 DHCP;

② 等待 CH395 产生 `GINT_STAT_DHCP` 中断;

③ 等到 `GINT_STAT_DHCP` 中断后, 发送命令 `CMD_GET_DHCP_STATUS` 获取 DHCP 状态, 如果状态码为 0 则表示成功, 单片机可以发送命令 `CMD_GET_IP_INF` 来获取 IP、MASK 等信息。如果状态码为 1 表示失败, 可能是 CH395 和 DHCP Server 之间的连接出现问题, 例如 DHCP Server 不在线, 虽然 CH395 通过中断向单片机通告 DHCP 错误, 但是内部仍然会一直重试来查找 DHCP Server, 单片机可以发送 `CMD_DHCP_ENABLE` 命令, 参数为 0 来停止 DHCP。

8.3.8. PPPoE

仅 CH395Q 支持 PPPoE 功能。在开始下面的步骤前, 首先要对 CH395 进行初始化。

PPPoE 步骤如下:

① 发送命令 `CMD_PPPOE_SET_USER_NAME` 设置 PPPoE 用户名;

② 发送命令 `CMD_PPPOE_SET_PASSWORD` 设置 PPPoE 密码;

③ 发送命令 `CMD_PPPOE_ENABLE`, 参数为 1 启动 PPPoE;

④ 等待 `GINT_STAT_PPPOE` 中断;

⑤ 等到 `GINT_STAT_PPPOE` 中断后, 发送命令 `CMD_GET_PPPOE_STATUS` 获取 PPPoE 状态, 如果为 0 表示成功, 单片机可以发送命令 `CMD_GET_IP_INF` 来获取 IP、MASK 等信息。如果为其他值, 则表示错误。

8.3.9. 关于 TCP MSS 和缓冲区

CH395 支持修改 TCP MSS, 默认的 TCP MSS 大小为 800。一般而言 TCP MSS 越大通讯的速度和效率就越高。在对 TCP MSS 和接收缓冲区修改时应该遵循一些原则。

- ① (建议) 接收缓冲区的长度不小于 2 倍的 TCP MSS;
- ② (必须) 接收缓冲区的长度不得小于 TCP MSS;
- ③ (建议) 接收缓冲区的长度不大于 6 倍的 TCP MSS;
- ④ (必须) 发送缓冲区大小不超过 8KB。

在对 TCP MSS 修改时尽量能大则大, 如果过小可能会导致以太网上的小包数据增多, 影响通讯效率。