

以太网协议栈芯片 CH392

版本: v1.0

<http://wch.cn>

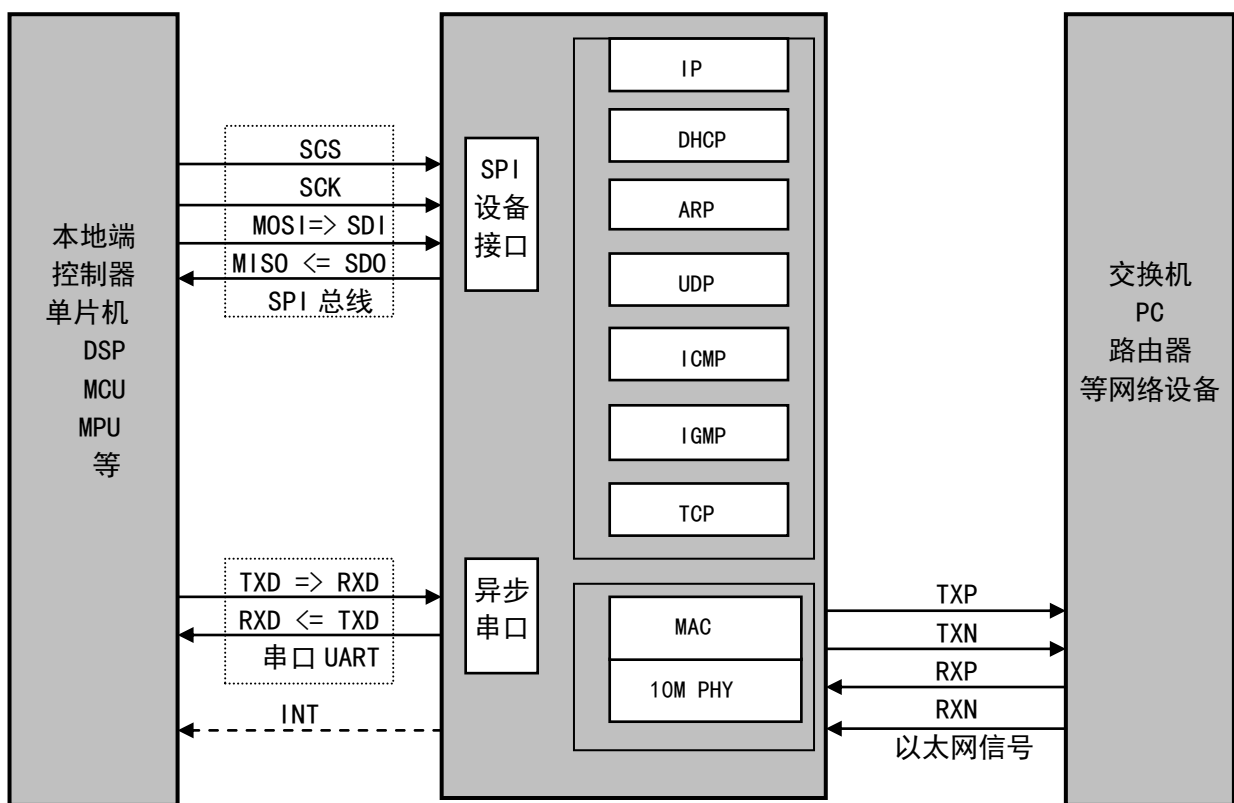
1、概述

CH392 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。

CH392 芯片自带 10M 以太网介质传输层 (MAC) 和物理层 (PHY)，完全兼容 IEEE802.3 协议，内置了 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 CH392 芯片进行网络通讯。

CH392 支持两种通讯接口：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 CH392 芯片进行以太网通讯。

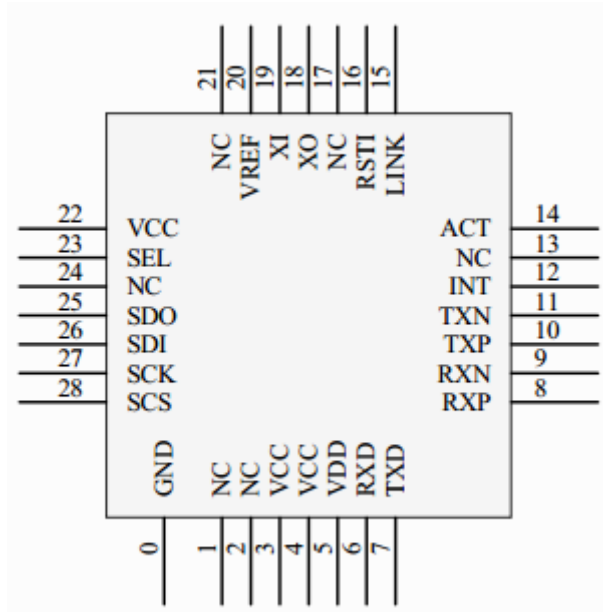
下图为 CH392 的应用框图。



2、特点

- 内部自带以太网介质传输层 (MAC) 和物理层 (PHY)。
- 支持 10M，全双工/半双工自适应，兼容 802.3 协议。
- 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。
- 提供 4 个独立的 Socket 对，可以同时进行数据收发。
- 提供最高 8MHz 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 921600bps 速度的异步串口，支持连接到单片机的串行口，支持通讯波特率动态调整。
- 内置 4KB 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。
- 提供 QFN28 封装。

3、封装



芯片型号	芯片封装	
	名称	描述
CH392F	QFN28	QFN 封装；28 脚；本体 4x4mm

4、引脚

CH392F 引脚号	引脚名称	类型	引脚说明
0	GND	电源	芯片接地端
3, 4, 22	VCC	电源	内部电源，需外接 0.1uF 退耦电容
5	VDD	电源	3.3V 工作电压输入，外接 2.2uF 退耦电容
6	RXD	输入	异步串口数据输入，内置上拉电阻
7	TXD	输出	异步串口数据输出，内置上拉电阻
8	RXP	以太网信号	以太网 RXP 信号
9	RXN	以太网信号	以太网 RXN 信号
10	TXP	以太网信号	以太网 TXP 信号
11	TXN	以太网信号	以太网 TXN 信号
12	INT	输出	中断信号输出，低电平有效
14	ACT	输出	以太网连接通讯指示灯驱动引脚
15	LINK	输出	PHY 连接指示引脚，低有效
16	RSTI	输入	外部复位输入，低电平有效
18	XO	输出	晶体振荡的反相输出端，需要外接 32MHz 晶振
19	XI	输入	晶体振荡的输入端，需要外接 32MHz 晶振
20	VREF	电源	内部模拟电路电源节点，需外接 1uF 退耦电容
23	SEL	输入	通讯接口选择引脚，内置上拉，高电平选择串口，低电平选择 SPI
25	SDO	输出	SPI 数据输出

26	SDI	输入	SPI 数据输入
27	SCK	输入	SPI 时钟输入
28	SCS	输入	SPI 片选输入
1, 2, 13, 17, 21, 24	NC	NC	保留管脚, 悬空

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。CH392 内部可以同时提供 4 个 Socket, 其索引值分别为 0, 1, 2, 3。

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
05H	RESET_ALL		(等 50mS)	执行硬件复位
06H	CHECK_EXIST	任意数据	按位取反	测试通讯接口和工作状态
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
26H	GET_PHY_STATUS		PHY 的状态	获取 PHY 的状态
27H	INIT_CH392			初始化 CH392 芯片
29H	GET_GLOB_INT_STATUS		全局中断状态	获取全局中断状态
2CH	GET_CMD_STATUS		命令执行状态	获取命令执行状态
2DH	GET_REMOT_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_DEST_PORT_SN	Socket 索引 目的端口 (2)		设置 Socket 的目的端口
33H	SET_SOUR_PORT_SN	Socket 索引 源端口 (2)		设置 Socket 的源端口

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_DISCONNECT_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		
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
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		
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		

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 CH392 INC. 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。

5.2. CMD_SET_BAUDRATE

该命令用于设置 CH392 的串口通讯波特率。在 CH392 工作于串口通讯方式时，复位后默认的通讯波特率默认是 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，计算公式：

BaudRate = (波特率系数 2 << 16) + (波特率系数 1 << 8) + 波特率系数 0。

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)
---------	---------	---------	---------------

00H	12H	C0H	4800
00H	25H	80H	9600
00H	4BH	00H	19200
00H	96H	00H	38400
00H	E1H	00H	57600
01H	2CH	00H	76800
01H	C2H	00H	115200
07H	08H	00H	460800
0EH	10H	00H	921600

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 CH392 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_RESET_ALL

该命令使 CH392 执行硬件复位。通常情况下，硬件复位在 50mS 时间之内完成。

5.4. CMD_CHECK_EXIST

该命令用于测试通讯接口和工作状态，以检查 CH392 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 CH392 正常工作，那么 CH392 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。

5.5. CMD_SET_MAC_ADDR

该命令用于设置 CH392 的 MAC 地址，需要输入 6 个字节的 MAC，MAC 地址低字节在前。CH392 芯片将会把 MAC 地址保存在内部的 EEPROM 中，该命令执行需要 100mS。

CH392 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。

5.6. CMD_SET_IP_ADDR

该命令用于设置 CH392 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。

5.7. CMD_SET_GWIP_ADDR

该命令用于设置 CH392 的网关地址，需要输入 4 个字节的 IP 地址。

5.8. CMD_SET_MASK_ADDR

该命令用于设置 CH392 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。

5.9. CMD_GET_PHY_STATUS

该命令用于获取 PHY 的连接状态。收到该命令后，CH392 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；

5.10. CMD_INIT_CH392

该命令用于初始化 CH392，包括：初始化 CH392 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。

5.11. CMD_GET_GLOB_INT_STATUS

该命令用于获取全局中断状态，收到此命令后 CH392 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	DHCP 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 CH392 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 CH392 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 CH392 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：DHCP 中断如果单片机使能 CH392 的 DHCP 功能，CH392 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。

⑤ GINT_STAT_SOCK0 - GINT_STAT_SOCK3：Socket 中断。当 Socket 有中断事件时 CH392 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，CH392 会将 INT#引脚置为高电平并将全局中断状态清除。

5.12. CMD_GET_CMD_STATUS

该命令用于获取命令执行的状态。CH392 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	CH392_ERR_SUCCESS	成功
10H	CH392_ERR_BUSY	忙，表示命令正在执行
11H	CH392_ERR_MEM	内存管理错误
12H	CH392_ERR_BUF	缓冲区错误
13H	CH392_ERR_TIMEOUT	超时
14H	CH392_ERR_RTE	路由错误
15H	CH392_ERR_ABRT	连接中止

16H	CH392_ERR_RST	连接复位
17H	CH392_ERR_CLSD	连接关闭
18H	CH392_ERR_CONN	无连接
19H	CH392_ERR_VAL	值错误
1AH	CH392_ERR_ARG	参数错误
1BH	CH392_ERR_USE	已被使用
1CH	CH392_ERR_IF	MAC 错误
1DH	CH392_ERR_ISCONN	已连接
20H	CH392_ERR_OPEN	已打开

如果单片机收到 CH392_ERR_BUSY，表示 CH392 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。

5.13. CMD_GET_REMOT_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，CH392 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。

5.14. 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，CH392 会将此 Socket 的接收长度清零，接收指针指向缓冲区头部。

5.15. 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。CH392 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	2MLS 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。

5.16. 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，CH392 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，CH392 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，CH392 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，CH392 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，CH392 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。

5.17. 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 UDP、TCP Client 模式下，必须在发送 CMD_OPEN_SOCKET_SN 命令前之前设置目的 IP。

5.18. 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。

5.19. 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。

5.20. 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 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

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。

5.21. 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。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。

5.22. 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。

5.23. 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。

5.24. 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。

5.25. 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，当外部单片

机写完数据后，CH392 会根据 Socket 的工作模式进行数据包的封装，然后将其发送。

5.26. 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，CH392 接收到该命令后输出 2 个字节的长度（低字节在前）。

5.27. 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），CH392 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。

5.28. 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 CH392 会自动断开 TCP 连接。

5.29. CMD_GET_MAC_ADDR

该命令用于获取 MAC 地址，接收到该命令后，CH392 会输出 6 个字节的 MAC 地址。

5.30. 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 CH392 初始化。

在启动 DHCP 后，CH392 将会向网络上广播 DHCP DISCOVER 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 CH392 分配了一个配置，且此配置和原来不相同，CH392 仍然会产生中断。

产生超时中断后，假如没有发现 DHCP Server，CH392 仍然会继续发送 DHCP DISCOVER 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

5.31. 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，CH392 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。

5.32. CMD_GET_IP_INF

该命令用于获取 IP、GatewayIP、MASK、DNS 等信息，收到此命令后，CH392 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 后，可以发送此命令来获取当前 CH392 的信息。如果在 DHCP 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。

5.33. 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。

5.34. 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 7 块	第 8 块
-------	-------	-------	-------	-------	-------

CH392 内部缓冲区结构如上图，一共由 8 个块组成，每个块的长度为 536 字节。单片机可以自由分配每个 Socket 接收缓冲区的大小。CH392 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	2
	发送缓冲区	2	4
1	接收缓冲区	4	6
	发送缓冲区	6	8
2	接收缓冲区	NULL	0
	发送缓冲区	NULL	0
3	接收缓冲区	NULL	0
	发送缓冲区	NULL	0

由上表可以看出，CH392 初始化完成后，将所有的缓冲区分配给 Socket0 - 1，接收缓冲区为 2 块（1072B），发送缓冲区为 2 块（1072B）。如果单片机需要用到的 Socket 个数大于 2，需要重新对缓冲区进行分配。

5.35. 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。

5.36. 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。

5.37. 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。

5.38. 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。

5.39. 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

CH392 与单片机之间支持两种通讯接口：SPI 接口、异步串口。在芯片上电复位时，CH392 将采样引脚的状态，根据这个引脚电平选择通讯接口，参考下表（表中 0 代表低电平，1 代表高电平或者悬空）。

SEL 引脚	选择通讯接口
1	异步串口
0	SPI 接口

CH392 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 CH392 的中断请求。

6.2. 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 接口，CH392 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。

CH392 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。CH392 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。

CH392 的 SPI 接口支持 SPI 模式 0 和 SPI 模式 3，CH392 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 CH392 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，CH392 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；

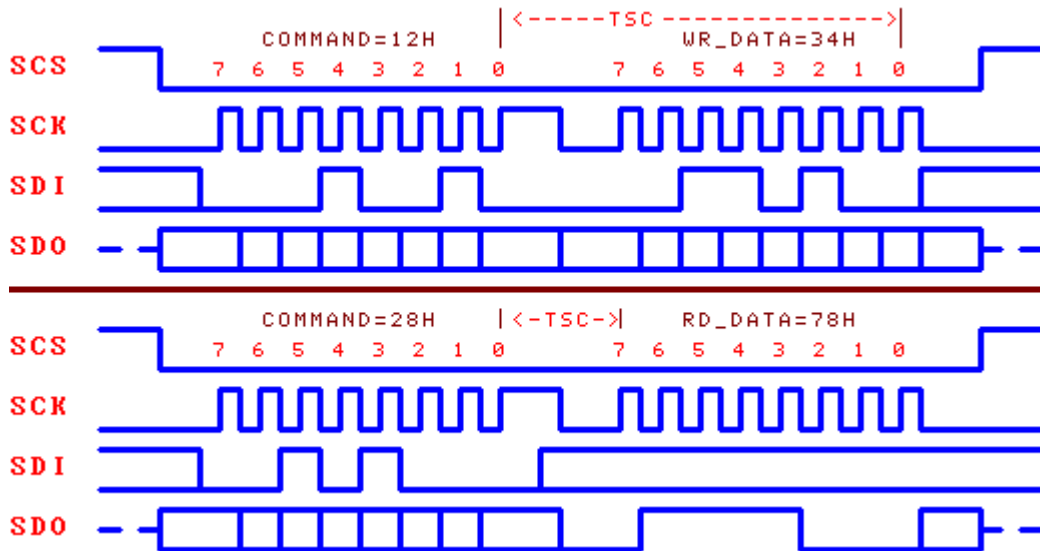
③ 单片机延时 TSC 时间（约 1mS）等待 CH392 的 SPI 接口空闲；

④ 如果是写操作，单片机向 CH392 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，CH392 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 CH392 接收一个字节的数据，等待 SPI 接口空闲后，单片机继续从 CH392 接收若个字节的数据，直到单片机禁止 SPI 片选；

⑥ 单片机禁止 CH392 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。



6.3. 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，CH392 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。

CH392 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。

CH392 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。

CH392 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，CH392 默认的串行通讯波特率为 9600bps。

为了区分命令码和数据，CH392 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。CH392 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 CH392 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 CH392 发出第一同步码 57H；

② 单片机向 CH392 发出第二同步码 ABH；

③ 单片机向 CH392 发出命令码；

④ 如果该命令具有输入数据，那么依次向 CH392 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 CH392 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。

6.4. 其它硬件

CH392 芯片内部集成了 10M 以太 PHY 和 MAC、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。

CH392 芯片的 RXP, RXN, TXP 和 TXN 为以太网的信号线。CH392 的 PHY 支持 MDI/MDIX 线路自动

转换。

CH392 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，CH392 芯片被复位；当 RSTI 引脚恢复为高电平后，CH392 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 0.1 μ F 左右的电容。

CH392 芯片正常工作时需要外部为其提供 32MHz 的时钟信号。CH392 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 CH392 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 X0 引脚之间连接一个标称频率为 32MHz 的晶体。

7、参数

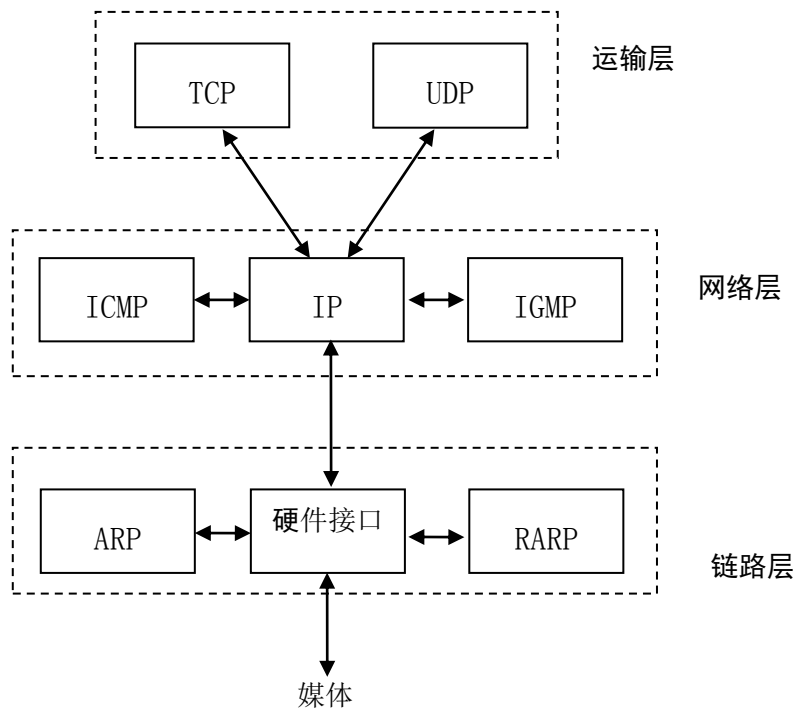
7.1. 绝对最大值（临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏）

名称	参数说明		最小值	最大值	单位
TA	工作时的环境温度	VCC=3.3V	-40	85	°C
TS	储存时的环境温度		-55	125	°C
VCC	电源电压（典型值为 3.3V）		2.1	3.6	V
VIO	输入或者输出引脚上的电压		-0.4	VCC+0.4	V

8、应用

8.1. 应用基础

CH392 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 CH392 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。

8.2. 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。

8.2.1. 初始化 CH392，必要操作

- ① 发送命令 `CMD_SET_MAC_ADDR` 设置 CH392 的 MAC 地址；
- ② 发送命令 `CMD_SET_IP_ADDR` 设置 CH392 的 IP 地址；
- ③ 发送命令 `CMD_SET_GWIP_ADDR` 设置 CH392 的网关 IP 地址；
- ④ 发送命令 `CMD_SET_MASK_ADDR` 设置 CH392 的子网掩码；
- ⑤ 发送命令 `CMD_INIT_CH392` 对 CH392 初始化；
- ⑥ 延时 2ms 以上发送命令 `CMD_GET_CMD_STATUS` 获取 `CMD_INIT_CH392` 执行状态，如果返回

CH392_ERR_BUSY 表示 CH392 内部正在执行命令，需要再次执行⑥；如果返回 CH392_ERR_SUCCESS 表示命令执行成功。CMD_INIT_CH392 一般需要 350mS 才能执行完毕。

步骤①一般不需要，CH392 在出厂时，已经烧录了 IEEE 分配的 MAC 地址。

如果需要启动 DHCP，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。

CH392 接收到 CMD_INIT_CH392 命令后，初始化内部 TCP/IP 协议栈，初始化 MAC 和 PHY。

8.2.2. 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 CH392_ERR_BUSY 表示 CH392 内部正在执行命令，需要再次执行⑥；如果返回 CH392_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 CH392 写入若干字节数据流后，CH392 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，CH392 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 CH392 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 CH392 不在同一个子网，则可能 CH392 的网关不在线。

当 CH392 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 CH392 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。

CH392 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 CH392 读取数据，客户端模式下 CH392 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 CH392 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 CH392 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 CH392 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。

8.3.3. 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 `CH392_ERR_BUSY` 表示 CH392 内部正在执行命令，需要再次执行⑥；如果返回 `CH392_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 `CH392_ERR_BUSY` 表示 CH392 内部正在执行命令，需要再次执行⑧；如果返回 `CH392_ERR_SUCCESS` 表示命令执行成功。其他值表示命令执行失败。返回 `CH392_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，CH392 将会产生 `SINT_STAT_CONNECT` 中断。如果连接失败 CH392 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。

CH392 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 CH392 写入若干字节数据流后，CH392 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，CH392 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，CH392 会自动将此 Socket 关闭(在 `FUN_PARA_FLAG_SOCKET_CLOSE` 为 0 情况下，参考 5.55)。

当 CH392 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，CH392 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。

8.3.4. 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 `CH392_ERR_BUSY` 表示 CH392 内部正在执行命令，需要再次执行④；如果返回 `CH392_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，CH392 会产生 `SINT_STAT_CONNECT` 中断，此时单片机可以发送命令 `CMD_GET_REMOT_IPP_SN` 来获取客户端的 IP 地址和端口号。

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，CH392 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。