

# 以太网协议栈芯片 CH394

手册 2

版本: 1.0

https://wch.cn

## 1、概述

CH394 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。

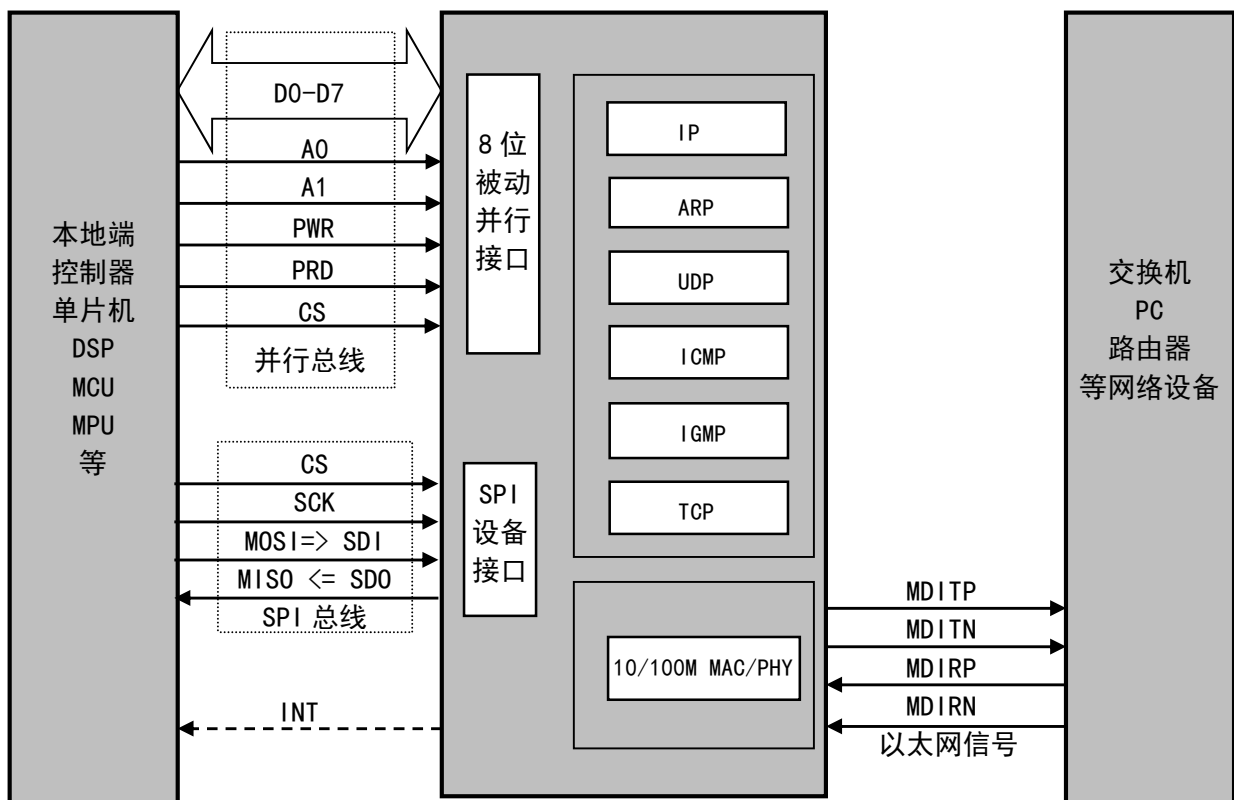
CH394 芯片自带 10/100M 以太网介质传输层 MAC 和物理层收发器 PHY，完全兼容 IEEE802.3 协议，内置了 IP、ARP、ICMP、IGMP、UDP、TCP 等以太网协议栈固件。单片机系统可以方便地通过 CH394 芯片进行网络通讯。CH394 支持网络唤醒模式（WOL）和掉电模式。

CH394L 提供了 SPI 接口和 8 位被动并行接口。单片机/DSP/MCU/MPU 等控制器可以通过 SPI 接口或 8 位并行接口控制 CH394L 芯片进行以太网通讯。

本手册的内容主要适用于 CH394L，CH394Q 的相关详细信息请参考《CH394DS1》手册。

下图为 CH394L 的应用框图。

图 1-1 CH394L 应用框图

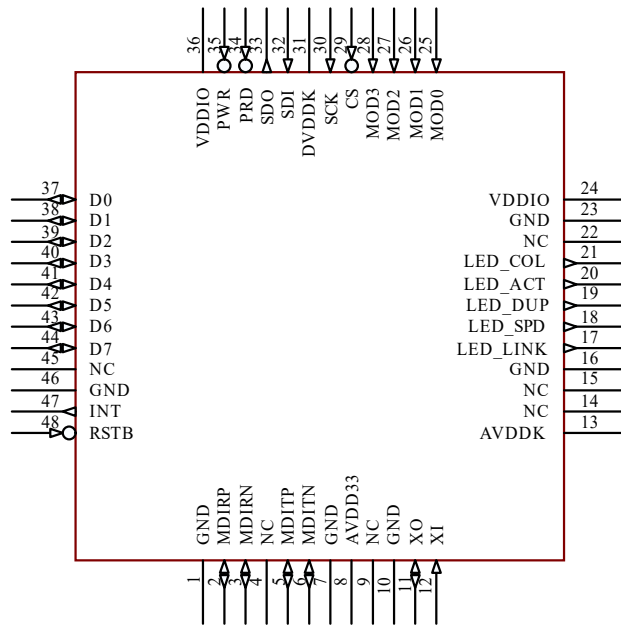


## 2、特点

- 内部自带以太网介质传输层 MAC 和物理层收发器 PHY
- 支持 10/100M，全双工/半双工自适应，兼容 IEEE802.3 协议
- 支持 MDI/MDIX 线路自动转换
- I/O 口支持 3.3V、2.5V、1.8V 供电，兼容多种电压标准的单片机
- 内置 50Ω 阻抗匹配电阻，内置 25MHz 晶体振荡器所需电容，外围电路精简
- 内置 TCP/IP 协议簇，支持 IPv4、ARP、ICMP、IGMP、UDP、TCP 协议
- 支持 4 个 Socket，可以同时进行数据收发
- 支持 MACRAW 模式和 IPRAW 模式
- 提供最高 40MHz 速度的 SPI 设备接口（SPI 模式 0 或 3），高位在前
- 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。

- 支持网络唤醒模式（WOL）和掉电模式
- 提供 LED 状态显示（Link, ACT, 10/100M, 全双工/半双工等）
- 内置 16K RAM，可用于以太网数据收发，每个 Socket 收发缓存区可以自由配置
- 提供 LQFP48 无铅封装

### 3、CH394L 引脚排列



CH394L

表 3-1 封装说明

封装形式	塑体尺寸	引脚节距		封装说明	订货型号
LQFP48	7*7mm	0.5mm	19.7mil	标准 LQFP48 脚贴片	CH394L

### 4、CH394L 引脚定义

表 4-1 CH394L 引脚定义

CH394L 引脚号	引脚名称	类型	引脚说明
2	MDIRP	I/O	10BASE-T/100BASE-TX MDI 模式下的差分输入； 10BASE-T/100BASE-TX MDIX 模式下的差分输出。
3	MDIRN		
5	MDITP	I/O	10BASE-T/100BASE-TX MDI 模式下的差分输出； 10BASE-T/100BASE-TX MDIX 模式下的差分输入。
6	MDITN		
8	AVDD33	P	3.3V主电源输入，建议0.1uF并联10uF或4.7uF对地电容贴近芯片放置，或单个1uF~4.7uF。
13	AVDDK	P	外接1uF对地电容贴近芯片放置。
31	DVDDK	P	外接0.1uF或1uF对地电容贴近芯片放置。
24、36	VDDIO	P	I/O接口的电源输入，建议0.1uF或1uF对地电容贴近芯片放置。
1、7、10、16、23、46	GND	P	公共接地端。
4、9、14、15、22、45	NC	-	保留引脚，建议悬空。
11	XI	I	晶体振荡器输入，需外接25MHz晶体一端，或外部时钟输入，内置晶振匹配电容。
12	XO	O	晶体振荡器反相输出，需外接25MHz晶体另一端，内置晶振匹配电容。
17	LED_LINK	O	网络连接指示LED输出： 低电平表示已连接；

			高电平表示未连接。						
18	LED_SPD	0	网络速度指示 LED 输出： 低电平表示 100Mbps； 高电平表示 10Mbps。						
19	LED_DUP	0	双工指示 LED 输出： 低电平表示全双工； 高电平表示半双工。						
20	LED_ACT	0	载波感应指示 LED 输出： LED 闪烁表示有载波感应信号。						
21	LED_COL	0	冲突检测指示 LED 输出： 低电平表示数据传输过程中有冲突； 高电平表示数据传输过程中无冲突。						
25	MOD0	I, PD	接口模式选择： MOD 引脚内置下拉电阻，默认选择 SPI 接口。 如需选择并口模式，只需将 MOD2 上拉，其他引脚悬空或下拉即可。						
26	MOD1	I, PD							
27	MOD2	I, PD							
28	MOD3	I, PD	<table border="1"> <thead> <tr> <th>MOD[3:0]</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>采用 SPI 接口</td> </tr> <tr> <td>010x</td> <td>采用 8 位被动并行接口</td> </tr> </tbody> </table>	MOD[3:0]	说明	0000	采用 SPI 接口	010x	采用 8 位被动并行接口
MOD[3:0]	说明								
0000	采用 SPI 接口								
010x	采用 8 位被动并行接口								
29	CS	I, PU	片选输入，低电平有效。						
30	SCK	I	SPI 时钟输入，支持模式 0 或 3。						
32	SDI	I	复用引脚 当 CH394L 采用 SPI 接口时，作为： SPI 串行数据输入，连接主机 SPI 接口的 MOSI。 当 CH394L 采用并行接口时，作为： 地址 0 (A0)。						
33	SDO	0	复用引脚 当 CH394L 采用 SPI 接口时，作为： SPI 串行数据输出，连接主机 SPI 接口的 MISO，SCS 为高时，SDO 为高阻态。 当 CH394L 采用并行接口时，作为： 地址 1 (A1)。						
34	PRD	I	处理器并行接口读取控制信号输入，低电平有效。						
35	PWR	I	处理器并行接口写入控制信号输入，低电平有效。						
37~44	D0~D7	I/O	并行接口的 8 位双向数据总线。						
47	INT	0	中断请求输出，低电平有效。						
48	RSTB	I, PU	复位输入，低电平有效。						

注 1: I = 输入; 0 = 输出; I/O = 输入/输出; P = 电源;  
PU = 内置上拉电阻; PD = 内置下拉电阻。

## 5、CH394L 数据格式

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

CH394L 支持 SPI（串行外设接口）和 8 位并行接口，当 CH394L 经历上电或复位流程时会根据 MOD[3:0] 引脚的状态来确定使用 SPI 接口或 8 位并行接口。单片机系统可以通过任意选定接口实现对 CH394L 内部寄存器的读写操作，以及与收发缓存区的数据交互。

CH394L 支持对数据进行连续的读取或写入操作，从起始地址开始，每传输完一个偏移地址的数据后，偏移地址会自动加 1 传输接下来的数据。

### 5.1 SPI 接口数据帧

当 MOD[3:0] 引脚为“0000”时 CH394L 采用 SPI 接口。

CH394L SPI 数据帧结构中包含三个部分：一个 8 位的控制字段，一个 16 位的偏移地址字段，以及一个可变长度的数据字段。

控制字用于明确数据传输的方向——读取或写入操作，可设为：

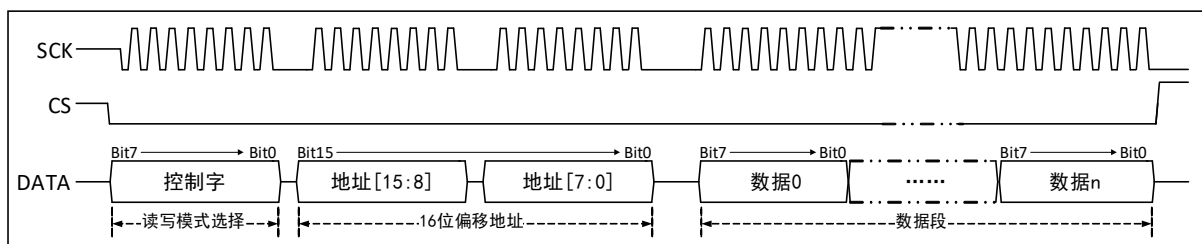
表 5-1 控制字

模式	控制字值
写操作	0xF0
读操作	0x0F

偏移地址为 CH394L 寄存器地址或者 RX/TX 缓存的 16 位偏移地址。

CS 拉低代表开始一包新的数据帧，CS 拉高代表一包数据帧结束。

图 5-1 SPI 接口数据帧

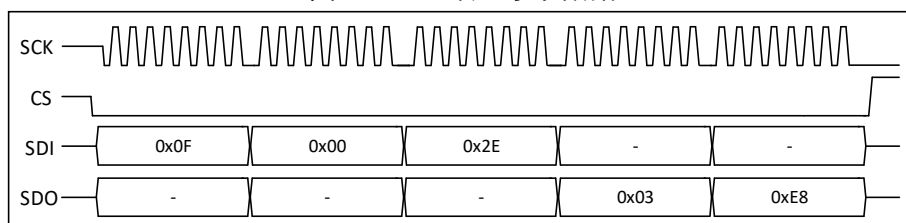


#### 5.1.1 SPI 数据帧示例

##### (1) 读 2 字节数据

从通用寄存器读（0x002E）两字节不可达端口信息（假设不可达端口为 0x03e8）：

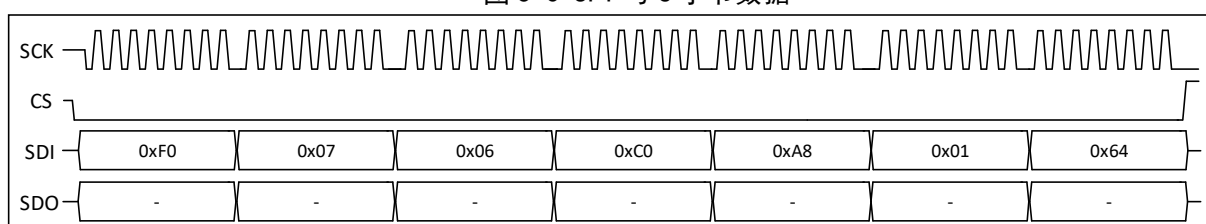
图 5-2 SPI 读 2 字节数据



##### (2) 写 6 字节数据

配置 Socket3 目的 IP (0x070C=0x040C+0x0100\*3) 为 192.168.1.100:

图 5-3 SPI 写 6 字节数据



## 5.2 并行接口数据帧

当 MOD[3:0] 引脚为 “010X” 时 CH394L 采用 8 位被动并行接口。

CH394L 并行接口数据帧结构中包含两个部分：一个 16 位的偏移地址字段，以及一个可变长度的数据字段。

并行接口采用 8 位数据线 D0~D7，2 位地址线 A0、A1（和 SD1、SD0 脚复用），一位写使能 PWR，一位读使能 PRD。

2 位地址线 A0~A1 控制了传输数据的模式：

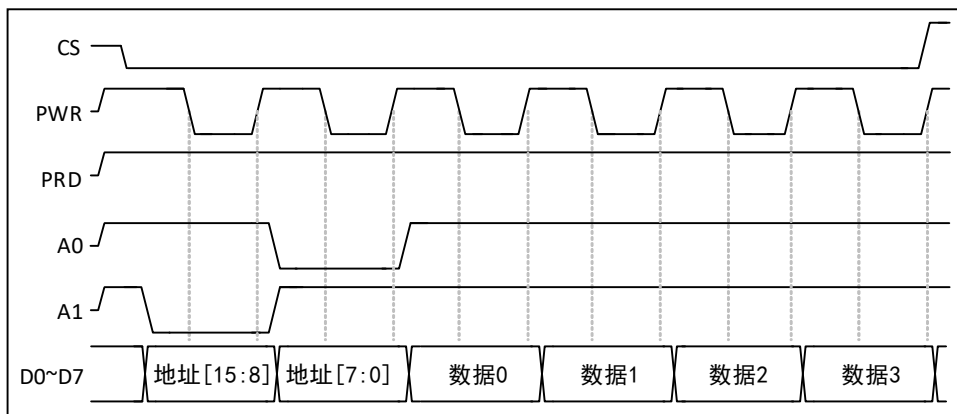
表 5-2 模式控制

A1	A0	模式
0	1	传输高 8 位地址
1	0	传输低 8 位地址
1	1	传输数据

偏移地址为 CH394L 寄存器地址或者 RX/TX 缓存的 16 位偏移地址。

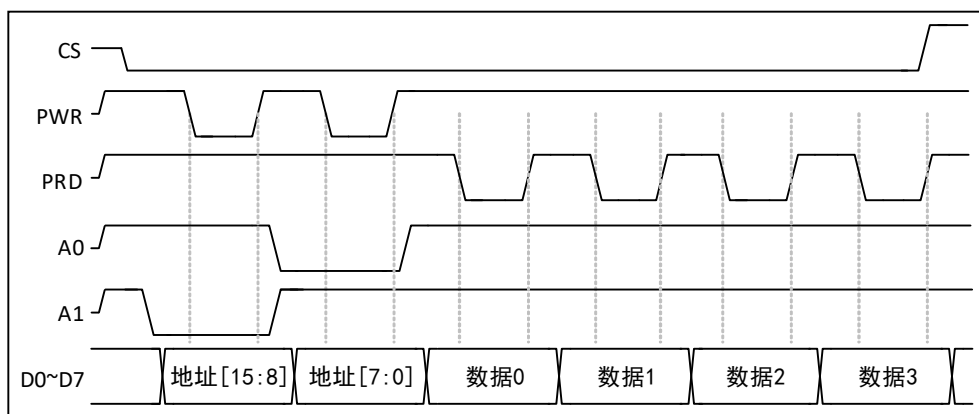
### 5.2.1 并口写入

图 5-4 并口写入



### 5.2.2 并口读取

图 5-5 并口读取

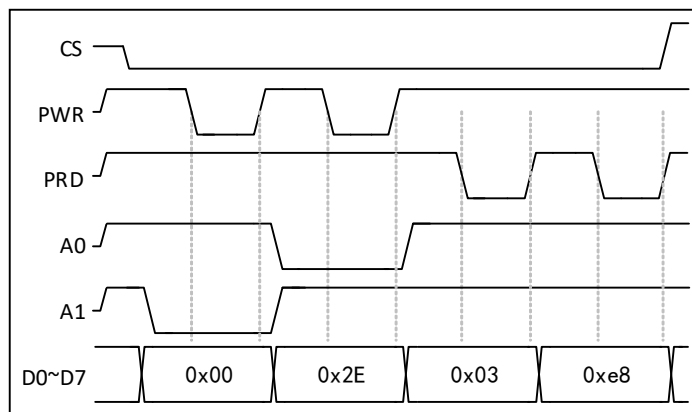


### 5.2.3 并口数据帧示例

#### (1) 读 2 字节数据

从通用寄存器读两字节不可达端口信息（假设不可达端口为 0x03e8）：

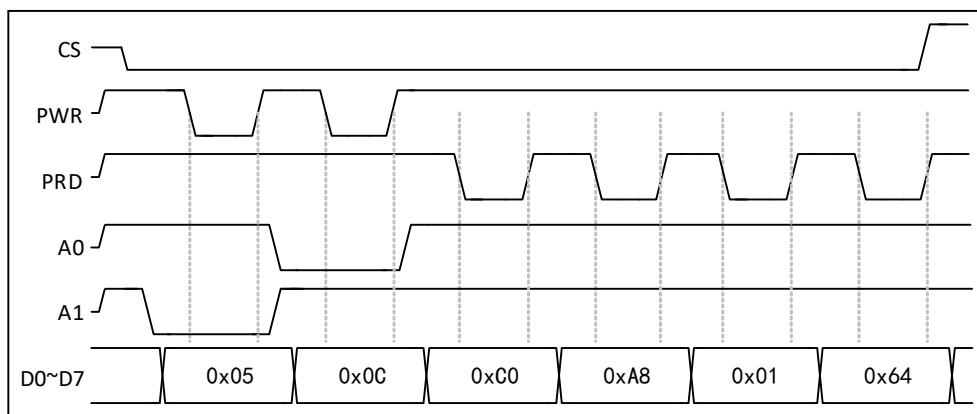
图 5-6 并口读 2 字节数据



(2) 写 4 字节数据

配置 Socket1 目的 IP (0x050C=0x040C+0x0100\*1) 为 192.168.1.101:

图 5-7 并口写 4 字节数据



## 6、CH394L 寄存器与缓存区

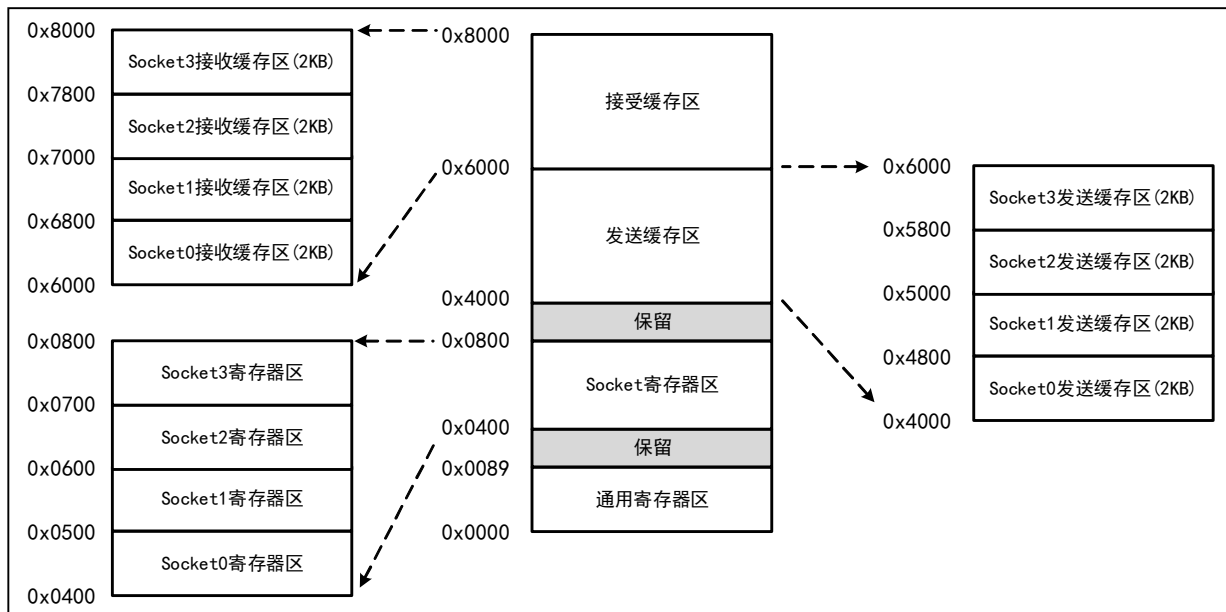
CH394L 有 1 个通用寄存器区，4 个 Socket 寄存器区，4 个接收缓存区与 4 个发送缓存区。

在 CH394L 的寄存器空间中，每个寄存器都被赋予了唯一的地址。单片机系统可以通过配置不同的寄存器从而实现对其各项功能的设置。

CH394L 共有 8K 发送缓存区，4 个 Socket 每个默认 2K；8K 接收缓存区，4 个 Socket 每个默认 2K。

当读/写 Socket 收/发缓存区时，地址段中的偏移地址（由查询 RD/WR 寄存器得来）代表了存放数据的起始地址。

图 6-1 寄存器与缓存区地址分配



### 6.1 寄存器表

#### 6.1.1 通用寄存器表

CH394L 通用寄存器配置了 CH394L 的模式，IP，MAC 等，读/写通用寄存器需要在数据帧的偏移地址部分填入寄存器对应的偏移地址。

表 6-1 通用寄存器表

偏移地址	名称	访问	描述	复位值
0x0000	MODE	RW	模式寄存器	0x03
0x0001	GWIP0	RW	网关地址寄存器	0x00
...	...			...
0x0004	GWIP3			0x00
0x0005	SMIP0	RW	子网掩码寄存器	0x00
...	...			...
0x0008	SMIP3			0x00
0x0009	MAC0	RW	MAC 地址寄存器	0xXX
...	...			...
0x000E	MAC5			0xXX
0x000F	IP0	RW	源 IP 地址寄存器	0x00
...	...			...
0x0012	IP3			0x00
0x0013	IIT0	RW	中断间隔时间寄存器	0x00
0x0014	IIT1			0x00



0x0015	GINT	RW	全局中断寄存器	0x00
0x0016	GINTE	RW	全局中断使能寄存器	0x00
0x0017	RTIME0	RW	重传时间寄存器	0x07
0x0018	RTIME1			0xD0
0x0019	RCNT	RW	重传次数寄存器	0x07
0x001A	SRXBUF_SIZE	RW	Socket 接收缓存区大小寄存器	0x55
0x001B	STXBUF_SIZE	RW	Socket 发送缓存区大小寄存器	0x55
0x001C ... 0x001F	-	-	保留	-
0x0020	EINT	RW	扩展中断寄存器	0x00
0x0021	EINTE	RW	扩展中断使能寄存器	0x00
0x0022 ... 0x0029	-	-	保留	-
0x002A ... 0x002D	UNIP0 ... UNIP3	RO	不可达 IP 寄存器	0x00 ... 0x00
0x002E 0x002F	UNPORT0 UNPORT1	RO	不可达端口寄存器	0x00 0x00
0x0030	ESET	RW	扩展设置寄存器	0x40
0x0031 ... 0x003B	-	-	保留	-
0x003C	PHYSTA	RO	PHY 状态寄存器	0x00
0x003D 0x003E	-	-	保留	-
0x003F	PHYA	RW	PHY 寄存器地址寄存器	0x00
0x0040 0x0041	PHYDINO PHYDIN1	RW	PHY 数据输入寄存器	0x00 0x00
0x0042 0x0043	PHYDOUT0 PHYDOUT1	RO	PHY 数据输出寄存器	0x00 0x00
0x0044	PHYVC	RW	PHY 读写控制寄存器	0x00
0x0045	-	-	保留	-
0x0046	PHYM	RW	PHY 模式寄存器	0x00
0x0047	PHYC	RW	PHY 控制寄存器	0x41
0x0048 ... 0x004B	-	-	保留	-
0x004C	ND	RW	网络诊断寄存器	0x00
0x004D 0x004E	NDRT0 NDRT1	RW	网络诊断超时重传时间寄存器	0x07 0xD0
0x004F	NDRC	RW	网络诊断超时重传次数寄存器	0x00
0x0050 ... 0x0053	NDDIP0 ... NDDIP3	RW	网络诊断目的 IP 寄存器	0x00 ... 0x00
0x0054 ... 0x0059	NDDMAC0 ... NDDMAC5	RO	网络诊断目的 MAC 寄存器	0x00 ... 0x00
0x005A 0x005B	PINGSEQ0 PINGSEQ1	RW	PING 序列号寄存器	0x00 0x00

0x005C	PINGID0	RW	PING ID 号寄存器	0x00
0x005D	PINGID1	RW		0x00
0x005E	NDINTE	RW	网络诊断中断使能寄存器	0x00
0x005F	NDINT	RW	网络诊断中断寄存器	0x00
0x0060	-	-	保留	-
...				
0x0070				
0x0071	NETLOCK	RW	网络锁定寄存器	0x3A
0x0072	PHYLOCK	RW	PHY 锁定寄存器	0x00
0x0073	-	-	保留	-
...				
0x007F				
0x0080	CHIPV	RO	芯片版本寄存器	0xXX
0x0082	TCNT0	RO	定时器计数值寄存器	0x00
0x0083	TCNT1			0x00
0x0088	TCNTCLR	RW	定时器计数值清除寄存器	0x00

### 6.1.2 Socket 寄存器表

CH394L 提供 4 个 Socket，每个 Socket 关联一个专属的寄存器区域，其中每一项寄存器均关联至一个唯一的地址。

下表 6-4 所列表的地址代表了各寄存器的基地址，而要获取特定 Socket n 寄存器的实际地址需要加上 Socket n 对应的偏移量。

实际地址计算公式为：Socket n 实际地址 = 寄存器基地址 + (0x0100\*n)。

例 1：Socket n 控制寄存器 Sn\_CTRL 的基地址为 0x0401，则：

表 6-2 例 1

寄存器	地址
SO_CTRL	0x0401
S1_CTRL	0x0501
S2_CTRL	0x0601
S3_CTRL	0x0701

例 2：Socket n 重传次数寄存器 Sn\_RC 的基地址为 0x0434，则：

表 6-3 例 2

寄存器	地址
SO_RC	0x0434
S1_RC	0x0534
S2_RC	0x0634
S3_RC	0x0734

下表 6-4 涵盖了所有的 Socket 寄存器的基地址信息，可通过该地址使用上述公式计算出 4 个 Socket 寄存器的实际物理地址。

表 6-4 Socket 寄存器表

基地址	名称	访问	描述	复位值
0x0400	Sn_MODE	RW	Socket n 模式寄存器	0x00
0x0401	Sn_CTRL	RW	Socket n 控制寄存器	0x00
0x0402	Sn_INT	RW	Socket n 中断寄存器	0x00
0x0403	Sn_STA	RO	Socket n 状态寄存器	0x00
0x0404	Sn_PORT0	RW	Socket n 源端口寄存器	0x00
0x0405	Sn_PORT1			0x00
0x0406	Sn_DMACH0	RW	Socket n 目的 MAC 寄存器	0x00
...	...			...

0x040B	Sn_DMAC5			0x00
0x040C	Sn_DIP0			0x00
...	...	RW	Socket n 目的 IP 寄存器	...
0x040F	Sn_DIP3			0x00
0x0410	Sn_DPORT0			0x00
0x0411	Sn_DPORT1	RW	Socket n 目的端口寄存器	0x00
0x0412	Sn_MTU0			0x00
0x0413	Sn_MTU1	RW	Socket n 最大传输单元寄存器	0x00
0x0414	Sn_IPPRO	RW	Socket n IP 协议寄存器	0x00
0x0415	Sn_TOS	RW	Socket n IP 服务类型寄存器	0x00
0x0416	Sn_TTL	RW	Socket n IP 生存时间寄存器	0x80
0x0417				
...	-	-	保留	-
0x041D				
0x041E	Sn_RXBUF_SIZE	RW	Socket n 接收缓存区大小寄存器	0x02
0x041F	Sn_TXBUF_SIZE	RW	Socket n 发送缓存区大小寄存器	0x02
0x0420	Sn_TX_FS0			0x08
0x0421	Sn_TX_FS1	RO	Socket n 空闲发送缓存长度寄存器	0x00
0x0422	Sn_TX_RD0			0x00
0x0423	Sn_TX_RD1	RO	Socket n 发送缓存区读指针寄存器	0x00
0x0424	Sn_TX_WR0			0x00
0x0425	Sn_TX_WR1	RW	Socket n 发送缓存区写指针寄存器	0x00
0x0426	Sn_RX_RS0			0x00
0x0427	Sn_RX_RS1	RO	Socket n 接收数据长度寄存器	0x00
0x0428	Sn_RX_RD0			0x00
0x0429	Sn_RX_RD1	RW	Socket n 接收缓存区读指针寄存器	0x00
0x042A	Sn_RX_WR0			0x00
0x042B	Sn_RX_WR1	RO	Socket n 接收缓存区写指针寄存器	0x00
0x042C	Sn_INTE	RW	Socket n 中断使能寄存器	0xFF
0x042D	Sn_IPF0			0x40
0x042E	Sn_IPF1	RW	Socket n IP 分片寄存器	0x00
0x042F	Sn_ESET	RW	Socket n 扩展配置寄存器	0x00
0x0430	Sn_KEEPA_LIVE	RW	Socket n KeepAlive 时间寄存器	0x00
0x0431				
...	-	-	保留	-
0x0432	Sn_RT0			0x00
0x0433	Sn_RT1	RW	Socket n 重传时间寄存器	0x00
0x0434	Sn_RC	RW	Socket n 重传次数寄存器	0x00

## 6.2 接收/发送缓存区

CH394L 为每个 Socket n (n = 0~3) 提供了独立的发送和接收内存缓存区, 共有 8K 接收缓存区与 8K 发送缓存区。

初始状态下, 每个 Socket 的发送和接收缓存区均分配为 2KB。用户可根据需求, 通过配置 Socket 发送缓存大小寄存器 (Sn\_TXBUF\_SIZE) 和 Socket 接收缓存大小寄存器 (Sn\_RXBUF\_SIZE), 或者 SRXBUF\_SIZE 和 STXBUF\_SIZE 寄存器, 重新分配这 8KB 的内存资源给各个 Socket, 但需确保所有 Socket 的接收缓存与发送缓存大小与之和均不超过 8KB, 以防止数据传输错误。

读/写接收/发送缓存区具体操作详参 11.2 应用参考步骤。

## 7、CH394L 寄存器说明

Socket Pair (插口对) 是包含源 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。CH394L 内部可以同时提供 4 个 Socket, 其索引值依次为 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 为最高字节, 2 为最低字节。

如 MAC 地址为 0x11:0x22:0x33:0x44:0x55:0x66, 其中 0x11 为最高字节, 0x66 为最低字节。

寄存器中位属性缩写描述:

寄存器位属性	属性描述
RF	只读属性, 读出固定值。
RO	只读属性, 由硬件改变。
RZ	只读属性, 读操作后自动位清 0
WO	只写属性 (不可读, 读值不确定)
WA	只写属性, 安全模式下可写入。
WZ	只写属性, 写操作后自动位清 0
RW	可读, 可写。
RWA	可读, 安全模式下可写入。
RW1	可读, 写 1 有效, 写 0 无效。
RW0	可读, 写 0 有效, 写 1 无效。
RW1T	可读, 写 0 无效, 写 1 翻转。
RW1Z	可读, 写 1 清除此位, 写 0 无效。

### 7.1 通用寄存器说明

CH394L 寄存器采用大端模式进行数据存储。在内存中, 低字节被放置于高地址位, 而高字节则位于低地址位。

#### 7.1.1 模式寄存器 (MODE) [0x0000]

该寄存器用于控制工作模式。

表 7-1 模式寄存器

位	名称	描述	访问	默认值
7	RST	置 1 软件复位, 复位完成后自动清 0。	RW, SC	0
[6:5]	Reserved	保留。	RO	0
4	PB	PING 包屏蔽使能: 0: 响应 PING 包请求; 1: 不响应 PING 请求。	RW	0
[3:0]	Reserved	保留。	RO	11b

#### 7.1.2 网关 IP 地址寄存器 (GWIP) [0x0001-0x0004]

该寄存器用于设置网关地址, 低字节地址储存网关地址的高字节。

该寄存器只有在 NETLOCK (网络锁定) 状态为“解锁”时可更改。

例如: 网关 IP 为 192.168.1.1。

地址	0x0001	0x0002	0x0003	0x0004
数据	0xC0	0xA8	0x01	0x01

#### 7.1.3 子网掩码 IP 地址寄存器 (SMIP) [0x0005-0x0008]

该寄存器用于设置子网掩码地址, 低字节地址储存子网掩码地址的高字节。

该寄存器只有在 NETLOCK (网络锁定) 状态为“解锁”时可更改。

例如：子网掩码 IP 为 255.255.255.0。

地址	0x0005	0x0006	0x0007	0x0008
数据	0xFF	0xFF	0xFF	0x00

#### 7.1.4 MAC 地址寄存器 (MAC) [0x0009-0x000E]

该寄存器用于设置 CH394L 的 MAC 地址，低字节地址储存 MAC 地址的高字节。

该寄存器只有在 NETLOCK (网络锁定) 状态为“解锁”时可更改。

CH394L 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。

例如：MAC 地址为 0x38:0x3B:0x26:0x11:0x22:0x33。

地址	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E
数据	0x38	0x3B	0x26	0x11	0x22	0x33

#### 7.1.5 源 IP 地址寄存器 (IP) [0x000F-0x0012]

该寄存器用于设置 CH394L 的 IP 地址，低字节地址储存 IP 地址的高字节。

该寄存器只有在 NETLOCK (网络锁定) 状态为“解锁”时可更改。

例如：源 IP 地址为 192.168.1.100。

地址	0x000F	0x0010	0x0011	0x0012
数据	0xC0	0xA8	0x01	0x64

#### 7.1.6 中断间隔时间寄存器 (IIT) [0x0013-0x0014]

该寄存器用于设置中断生效的等待时间，默认为 0。当所有中断标志处置完毕后，INT 信号线恢复高电平状态，需经过设定的 IIT 间隔时间，CH394L 才会通知下一个中断，且将 INT 信号线拉低。

间隔时间计算公式为： $Time = 1/120000000 * 4 * (IIT + 1)$ 。

例如：寄存器值设为 1000，则间隔时间约为 33.3us。

地址	0x0013	0x0014
数据	0x03	0xE8

#### 7.1.7 全局中断寄存器 (GINT) [0x0015]

该寄存器用于获取全局中断状态。当出现中断事件，GINT 寄存器对应位将被置为 1。对该位写 1 可以清除该中断标志位 (仅适用于 IP\_CONFLI 中断以及 UNREACH 中断，Socketn 中断在对应的 Sn\_INT 寄存器为 0x00 时，自动清零)，若 GINTE 中该中断的使能位亦为 1，则 INT 脚电平变低。此后，一旦该事件对应位被清除或对应的中断使能位被置为 0，且其他中断寄存器无待处理中断，INT 脚则恢复为高电平。

表 7-2 全局中断寄存器

位	名称	描述	访问	默认值
7	IP_CONFLI	IP 冲突中断： 当 CH394L 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。	RW1Z	0
6	UNREACH	不可达中断： 当 CH394L 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，可以查询 UNIP、UNPORT 寄存器来获取不可达信息。	RW1Z	0
[5:4]	Reserved	保留	R0	0
3	Socket3	Socket3 产生中断 当 S3_INT 寄存器为 0x00 时，该位自动清零	R0	0
2	Socket2	Socket2 产生中断 当 S2_INT 寄存器为 0x00 时，该位自动清零	R0	0

1	Socket1	Socket1 产生中断 当 S1_INT 寄存器为 0x00 时, 该位自动清零	RO	0
0	Socket0	Socket0 产生中断 当 S0_INT 寄存器为 0x00 时, 该位自动清零	RO	0

### 7.1.8 全局中断使能寄存器 (GINTE) [0x0016]

该寄存器用于控制哪些中断源能够触发中断, 每个中断事件使能位对应全局中断寄存器 (GINT) 一个位。

当中断事件产生后, 只有 GINTE 中对应位为 1 时, CH394L 才会拉低 INT 脚, 产生中断。

表 7-3 全局中断使能寄存器

位	名称	描述	访问	默认值
7	IP_CONFLI	IP 冲突中断使能位: 使能该位即允许产生 IP 冲突中断。	RW	0
6	UNREACH	不可达中断使能位: 使能该位即允许产生不可达中断。	RW	0
[5:4]	Reserved	保留。	RO	0
3	Socket3	Socket3 中断使能位: 使能该位即允许产生 Socket3 中断。	RW	0
2	Socket2	Socket2 中断使能位: 使能该位即允许产生 Socket2 中断。	RW	0
1	Socket1	Socket1 中断使能位: 使能该位即允许产生 Socket1 中断。	RW	0
0	Socket0	Socket0 中断使能位: 使能该位即允许产生 Socket0 中断。	RW	0

### 7.1.9 重传时间寄存器 (RTIME) [0x0017-0x0018]

该寄存器用于设置 TCP 通讯中的重传时间, 以及 ARP 的重传时间。如果 TCP 传输或者 ARP 请求在 RTIME 时间内没有收到对方的回应, 即会进行重传操作或者触发超时中断。

该寄存器所设的重传时间为所有 Socket 默认的重传时间, 用户也可以通过设置 Sn\_RT 寄存器为任一特定 Socket 独立设定重传时间。

RTIME 需要两个字节的的时间值, 单位为 0.1ms。默认值为 2000 (0x07D0), 即 200ms (2000\*0.1ms), RTIME 寄存器可设为任意值, CH394L 会自动向下取整为 1000 的倍数。

例如: 设重传时间为 5000 (0x1388), 即 500ms。

地址	0x0017	0x0018
数据	0x13	0x88

### 7.1.10 重传次数寄存器 (RCNT) [0x0019]

该寄存器用于设置重传超时次数。重传超时次数指允许重传数据包连续无应答的最大次数, 默认为 7。当重传超过设定次数时, 对应的 Socket 的超时中断位将被置 1。

该寄存器所设的重传次数为所有 Socket 默认的重传次数, 用户也可以通过设置 Sn\_RC 寄存器为任一特定 Socket 独立设定重传次数。

### 7.1.11 Socket 接收缓存区大小寄存器 (SRXBUF\_SIZE) [0x001A]

该寄存器用于设置 Socket n 的接收缓存区大小, 默认值为 0x55 (01010101b)。CH394L 默认每个 Socket 分配 2K 空间。

CH394L 内部接收缓存区一共由 8 个块组成, 每个块的长度为 1024 字节。单片机可以自由分配每个 Socket 接收缓存区的大小。设置接收缓存区时需注意所有 Socket 接收缓存区总和不能超过 8KB。

该寄存器每两个位 (S0、S1) 控制一个 Socket 接收缓存区大小, 详见下表 7-4。

表 7-4 Socket 接收缓存区大小寄存器

位	对应 Socket	描述	访问	默认值
---	-----------	----	----	-----



7	Socket3	S1	RW	0
6		S0	RW	1
5	Socket2	S1	RW	0
4		S0	RW	1
3	Socket1	S1	RW	0
2		S0	RW	1
1	Socket0	S1	RW	0
0		S0	RW	1

S0、S1 所设值对应缓存区大小见下表 7-5。

表 7-5 S0、S1 所设值对应

S0	S1	缓存区大小
0	0	1KB
0	1	2KB
1	0	4KB
1	1	8KB

该寄存器与 Sn\_RXBUF\_SIZE 二者均具备配置 Socket 接收缓存区大小的功能，用户可任选一种方式进行设置。

#### 7.1.12 Socket 发送缓存区大小寄存器 (STXBUF\_SIZE) [0x001B]

该寄存器用于设置 Socket n 的发送缓存区大小，默认值为 0x55 (01010101b)。CH394L 默认每个 Socket 分配 2K 空间。

CH394L 内部发送缓存区一共由 8 个块组成，每个块的长度为 1024 字节。单片机可以自由分配每个 Socket 发送缓存区的大小。设置发送缓存区时需注意所有 Socket 发送缓存区总和不能超过 8KB。

该寄存器每两个位 (S0、S1) 控制一个 Socket 发送缓存区大小，详见下表 7-6。

表 7-6 Socket 发送缓存区大小寄存器

位	对应 Socket	描述	访问	默认值
7	Socket3	S1	RW	0
6		S0	RW	1
5	Socket2	S1	RW	0
4		S0	RW	1
3	Socket1	S1	RW	0
2		S0	RW	1
1	Socket0	S1	RW	0
0		S0	RW	1

S0、S1 所设值对应缓存区大小见下表 7-7。

表 7-7 S0、S1 所设值对应

S0	S1	缓存区大小
0	0	1KB
0	1	2KB
1	0	4KB
1	1	8KB

该寄存器与 Sn\_TXBUF\_SIZE 二者均具备配置 Socket 发送缓存区大小的功能，用户可任选一种方式进行设置。

#### 7.1.13 扩展中断寄存器 (EINT) [0x0020]

该寄存器可用于获取 WOL 中断状态。启用魔法包功能后，当出现 WOL 中断事件，EINT 寄存器 Bit0 将被置为 1。对该位写 1 可以清除该中断标志位，若 EINTE 中该中断的使能位亦为 1，则 INT 脚电平变低。此后，一旦该事件对应位被清除或对应的中断使能位被置为 0，且其他中断寄存器无待处理中断，INT 脚则恢复为高电平。

表 7-8 扩展中断寄存器

位	名称	描述	访问	默认值
[7:1]	Reserved	保留。	R0	0
0	WOL	WOL 唤醒中断： 当启用 WOL 功能且通过 UDP 收到 Magic Packet 唤醒包时会产生此中断。	RW1Z	0

## 7.1.14 扩展中断使能寄存器 (EINTE) [0x0021]

该寄存器可用于控制 WOL 唤醒包是否能够触发中断。当中断事件产生后，只有 EINTE 中 Bit0 为 1 时，CH394L 才会拉低 INT 脚，产生中断。

表 7-9 扩展中断使能寄存器

位	名称	描述	访问	默认值
[7:1]	Reserved	保留。	R0	0
0	WOL	WOL 中断使能位： 使能该位即允许产生 WOL 中断。	RW	0

## 7.1.15 不可达 IP 寄存器 (UNIP) [0x002A-0x002D]

该寄存器用于获取不可达 IP 地址。当 CH394L 给一个不可达的端口发数据时，对方会回复一个 ICMP 包（目标端口不可达）。接收到不可达报文后，CH394L 会产生不可达中断。单片机可以用该寄存器来获取不可达地址。

例如：不可达 IP 为 192.168.1.200。

地址	0x002A	0x002B	0x002C	0x002D
数据	0xC0	0xA8	0x01	0xC8

## 7.1.16 不可达端口寄存器 (UNPORT) [0x002E-0x002F]

该寄存器用于获取不可达端口。当 CH394L 给一个不可达的端口发数据时，对方会回复一个 ICMP 包（目标端口不可达）。接收到不可达报文后，CH394L 会产生不可达中断。单片机可以用该寄存器来获取不可达端口。

例如：不可达端口为 2000。

地址	0x002E	0x002F
数据	0x07	0xD0

## 7.1.17 扩展设置寄存器 (EST) [0x0030]

该寄存器用于配置 CH394L 的扩展设置，默认值为 0x40（INT 脚启用）。

表 7-10 扩展设置寄存器

位	名称	描述	访问	默认值
7	Reserved	保留。	R0	0
6	INTE	INT 脚功能使能： 0：禁用 INT 脚（INT 脚电平始终为高）； 1：启用 INT 脚（中断请求输出，低电平有效）。	RW	1
5	TCPRST	TCP RST 包回复功能： 0：正常回复 RST 包； 1：不回复 RST 包。 在面对来自对端向 CH394L 非活动状态端口的 TCP 数据包传输尝试时，CH394L 将自动产生并回传 RST (Reset) 数据包，以此宣告并终止此次无效连接尝试。 用户可置该位为 1，来主动抑制 RST 响应的生成。	RW	0
4	UDPUNPORT	UDP ICMP 包回复功能：	RW	0



		0: 正常回复 ICMP 包; 1: 不回复 ICMP 包。 在面对来自对端向 CH394L 非活动状态端口的 UDP 数据包传输尝试时, CH394L 将自动产生并回一个 ICMP 目的端口不可达的数据包, 以此宣告所尝试访问的目的端口目前处于不可用状态。 用户可置该位为 1, 来主动抑制 ICMP 回复。		
3	WOL	网络功能唤醒使能: 0: 关闭网络唤醒; 1: 开启网络唤醒。 CH394L 集成了魔法包唤醒功能, 使能魔法包唤醒后, CH394L 收到 UDP 发来的匹配自身 MAC 地址的魔法包 (0xFFFFFFFF+16 个 CH394L 的 MAC 地址) 后, 会产生 WOL 中断。	RW	0
2	Reserved	保留。	RO	0
1	FARP	强制 ARP 使能: 0: 关闭强制 ARP 模式; 1: 开启强制 ARP 模式。 开启强制 ARP 后, 每包 UDP 报文发送前会强制发送 ARP 请求。	RW	0
0	Reserved	保留。	RO	0

#### 7.1.18 PHY 状态寄存器 (PHYSTA) [0x003C]

该寄存器用于获取当前 PHY 的连接状态以及 PHYM (PHY 模式寄存器) 所设值。

表 7-11 PHY 状态寄存器

位	名称	描述	访问	默认值
[7:6]	Reserved	保留。	RO	0
5	SAUTO	以太网自协商功能设定值: 0: 启动自协商; 1: 禁用自协商。 该位指示了 PHYM[2] 所设值。	RO	0
4	SSPD	以太网速度设定值: 0: 100Mb/s; 1: 10Mb/s。 该位指示了 PHYM[1] 所设值。	RO	0
3	SDUP	以太网双工模式设定值: 0: 全双工; 1: 半双工。 该位指示了 PHYM[0] 所设值。	RO	0
2	DUP	双工模式指示位: 0: 全双工; 1: 半双工。	RO	0
1	SPD	速度指示位: 0: 100Mb/s; 1: 10Mb/s。	RO	0
0	LINK	连接状态指示: 0: 物理层未建立链接; 1: 物理层建立链接。	RO	0

#### 7.1.19 PHY 寄存器地址寄存器 (PHYA) [0x003F]

该寄存器用于设置 PHY 寄存器地址。

表 7-12 PHY 寄存器地址寄存器

位	名称	描述	访问	默认值
[7:5]	Reserved	保留。	RO	0
[4:0]	ADDR[4:0]	PHY 寄存器地址。	RW	0

## 7.1.20 PHY 数据输入寄存器 (PHYDIN) [0x0040-0x0041]

该寄存器用于向 PHYA 所设寄存器写入数据。

例如：写入数据为 0x3100。

地址	0x0040	0x0041
数据	0x31	0x00

## 7.1.21 PHY 数据输出寄存器 (PHYDOUT) [0x0042-0x0043]

该寄存器用于获取 PHYA 所设寄存器得数据。

例如：寄存器值为 0x2100。

地址	0x0042	0x0043
数据	0x21	0x00

## 7.1.22 PHY 读写控制寄存器 (PHYVC) [0x0044]

该寄存器用于设置 PHYA 所设寄存器读/写类型。

表 7-13 PHY 读写控制寄存器

寄存器值	操作方式
0x01	写寄存器
0x02	读寄存器

## 7.1.23 PHY 模式寄存器 (PHYM) [0x0046]

该寄存器用于设置 PHY 的速度和双工模式。

该寄存器只有在 PHYLOCK (PHY 锁定) 状态为“解锁”时可更改。

表 7-14 PHY 模式寄存器

位	名称	描述	访问	默认值
[7:3]	Reserved	保留。	RO	0
2	AUTO	自动协商 1: 禁止自动协商; 0: 启动自动协商。	RW	0
1	SPD	速度控制位: 0: 100Mb/s; 1: 10Mb/s。	RW	0
0	DUP	双工模式控制位: 0: 全双工; 1: 半双工。	RW	0

## 7.1.24 PHY 控制寄存器 (PHYC) [0x0047]

该寄存器用于控制 PHY 的工作模式。

该寄存器只有在 PHYLOCK (PHY 锁定) 状态为“解锁”时可更改。

表 7-15 PHY 控制寄存器

位	名称	描述	访问	默认值
[7:6]	Reserved	保留。	RO	01b
5	POWERDOWN	PHY 掉电模式控制位: 0: 正常工作模式; 1: 掉电模式。	RW	0
[4:1]	Reserved	保留。	RO	0

0	PHYREST	PHY 复位控制位： 0：正常工作； 1：PHY 复位（复位后自动清零）。	RW	1
---	---------	---	----	---

### 7.1.25 网络诊断寄存器（ND）[0x004C]

该寄存器用于网络诊断，可通过 ARP 或 PING 命令来进行网络诊断。使能命令后，CH394L 会根据 NDDIP 所设地址发送 ARP 或 PING 请求，ARP 请求结果保存在 NDDMAC 寄存器。两条命令不可同时执行，命令执行完成后会报 NDINT 中断，用户可根据中断状态判断请求结果。

表 7-16 PHY 控制寄存器

位	名称	描述	访问	默认值
[7:2]	Reserved	保留。	RO	0
1	ARP	ARP 请求发送命令： 0：准备就绪； 1：发送 ARP 请求包。	RW	0
0	PING	PING 请求发送命令： 0：准备就绪； 1：发送 PING 请求包。	RW	0

### 7.1.26 网络诊断超时重传时间寄存器（NDRT）[0x004D-0x004E]

该寄存器用于设置网络诊断重传时间。如果 PING 请求或者 ARP 请求在 NDRT 时间内没有收到对方的回应，即会进行重传操作或者触发超时中断。

NDRT 需要两个字节的的时间值，单位为 0.1ms。默认值为 2000（0x07D0），即 200ms（2000\*0.1ms），NDRT 寄存器可设为任意值，CH394L 会自动向下取整为 1000 的倍数。

例如：设重传时间为 5000（0x1388），即 500ms。

地址	0x004D	0x004E
数据	0x13	0x88

### 7.1.27 网络诊断超时重传次数寄存器（NDRC）[0x004F]

该寄存器用于设置重传超时次数。重传超时次数指允许重传数据包连续无应答的最大次数，默认为 0。当重传超过设定次数时，NDINT 报超时中断。

### 7.1.28 网络诊断目的 IP 寄存器（NDDIP）[0x0050-0x0053]

该寄存器用于设置 ARP 或 Ping 请求的目的 IP。

例如：目的 IP 为 192.168.1.2。

地址	0x0050	0x0051	0x0052	0x0053
数据	0xC0	0xA8	0x01	0x02

### 7.1.29 网络诊断目的 MAC 寄存器（NDDMAC）[0x0054-0x0059]

该寄存器用于获取 ARP 请求成功后对端设备的 MAC 地址。

例如：MAC 地址为 0x38:0x3B:0x26:0x11:0x22:0x11。

地址	0x0054	0x0055	0x0056	0x0057	0x0058	0x0059
数据	0x38	0x3B	0x26	0x11	0x22	0x11

### 7.1.30 PING 序列号寄存器（PINGSEQ）[0x005A-0x005B]

该寄存器用于设置 PING 请求包序列号（序列号不会自动增加）。

例如：设序列号为 1000（0x03E8）。

地址	0x005A	0x005B
数据	0x03	0xE8

## 7.1.31 PING ID 寄存器 (PINGID) [0x005C-0x005D]

该寄存器用于设置 PING 请求包的 ID。

例如：设 ID 号为 256 (0x0100)。

地址	0x005C	0x005D
数据	0x01	0x00

## 7.1.32 网络诊断中断使能寄存器 (NDINTE) [0x005E]

该寄存器用于控制哪些网络诊断中断源能够触发中断，每个中断事件使能位对应网络诊断中断寄存器 (NDINT) 一个位。

当中断事件产生后，只有 NDINTE 中对应位为 1 时，CH394L 才会拉低 INT 脚，产生中断。

表 7-17 网络诊断中断使能寄存器

位	名称	描述	访问	默认值
[7:3]	Reserved	保留。	RO	0
2	TIMEOUT	超时中断： 使能该位即允许产生超时中断。	RW	0
1	ARP	ARP 中断： 使能该位即允许产生 ARP 成功中断。	RW	0
0	PING	PING 中断： 使能该位即允许产生 PING 成功中断。	RW	0

## 7.1.33 网络诊断中断寄存器 (NDINT) [0x005F]

该寄存器用于获取网络诊断中断状态。当 ARP 或 PING 命令成功或失败后，NDINT 寄存器对应位将被置为 1。对该位写 1 可以清除该中断标志位，若 NDINTE 中该中断的使能位亦为 1，则 INT 脚电平变低。此后，一旦该事件对应位被清除或对应的中断使能位被置为 0，且其他中断寄存器无待处理中断，INT 脚则恢复为高电平。

表 7-18 网络诊断中断寄存器

位	名称	描述	访问	默认值
[7:3]	Reserved	保留。	RO	0
2	TIMEOUT	TIMEOUT 中断： 当 ARP 命令或 PING 命令重传超时后会产生此中断。	RW1Z	0
1	ARP	ARP 中断： 当 ARP 命令成功（收到 ARP 应答）后会产生此中断。	RW1Z	0
0	PING	PING 中断： 当 PING 命令成功（收到 PING 应答）后会产生此中断。	RW1Z	0

## 7.1.34 网络锁定寄存器 (NETLOCK) [0x0071]

该寄存器用于设置网络锁定状态，只有在解锁的状态下才能设置 GWIP、SMIP、MAC、IP 寄存器。该寄存器默认值为 0x3A。

表 7-19 网络锁定寄存器

寄存器值	状态
0x3A	解锁
0xC5	锁定
其他	保持

## 7.1.35 PHY 锁定寄存器 (PHYLOCK) [0x0072]

该寄存器用于设置 PHY 锁定状态，只有在解锁的状态下才能设置 PHYM、PHYC 寄存器。

表 7-20 网络锁定寄存器

寄存器值	状态
0x53	解锁
其他	锁定

## 7.1.36 芯片版本寄存器 (CHIPV) [0x0080]

该寄存器用于获取芯片版本。

## 7.1.37 定时器计数值寄存器 (TCNT) [0x0082-0x0083]

该寄存器用于获取芯片内部定时器值，该定时器在芯片初始化后开始自增，单位为 100us。

## 7.1.38 定时器计数值清除寄存器 (TCNTCLR) [0x0088]

写任意值清除定时器计数值。

## 7.2 Socket 寄存器说明

## 7.2.1 Socket n 模式寄存器 (Sn\_MODE) [0x0400]

该寄存器用于配置 Socket n 的工作模式，须在 Socket n 打开前设置。

表 7-21 Socket n 模式寄存器

位	名称	描述	访问	默认值												
7	MUL	0: 关闭 UDP 组播模式; 1: 开启 UDP 组播模式。 使用组播模式时，需要在 Socket 打开前，通过 Sn_DIP、Sn_DMIC 提前配置好组播 IP、组播 MAC 并置该位为 1。	RW	0												
6	MFEN	0: 关闭 MAC 地址过滤; 1: 开启 MAC 地址过滤。 使用 MACRAW 模式时，打开 MAC 地址过滤后，Socket n 只会接收网络上目的 MAC 为 CH394L 自身 MAC 地址以及广播地址的包。	RW	0												
5	TNA_IGMP	当 Socket n 工作在 TCP 模式时: 0: 关闭无延时 ACK; 1: 使能无延时 ACK。 使能无延时 ACK 后，Socket n 从对端收到数据包后会不做延时，尽快回复 ACK。否则，会等待 RTIME 时间后回复 ACK。 当 Socket n 工作在 UDP 组播模式时: 0: 使用 IGMP 版本 2; 1: 使用 IGMP 版本 1。	RW	0												
4	Reserved	保留。	RO	0												
[3:0]	PMD	设置 Socket 工作模式: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>[3:0]</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>关闭</td> </tr> <tr> <td>0001</td> <td>TCP</td> </tr> <tr> <td>0010</td> <td>UDP</td> </tr> <tr> <td>0011</td> <td>IPRAW</td> </tr> <tr> <td>0100</td> <td>MACRAW</td> </tr> </tbody> </table> <p>注: 只有 Socket 0 可以使用 MACRAW 模式。</p>	[3:0]	说明	0000	关闭	0001	TCP	0010	UDP	0011	IPRAW	0100	MACRAW	RW	0
[3:0]	说明															
0000	关闭															
0001	TCP															
0010	UDP															
0011	IPRAW															
0100	MACRAW															

## 7.2.2 Socket n 控制寄存器 (Sn\_CTRL) [0x0401]

该寄存器用于设置 Socket n 的控制命令，配置完该寄存器后，寄存器会自动清零。设完控制命令后，控制命令通常需要执行一定的时间。用户可以通过 Sn\_INT 或 Sn\_STA 寄存器判断命令执行状态。必须在寄存器清零后才可配置下一条命令。

表 7-22 Socket n 控制寄存器

代码	名称	描述
----	----	----

01H	OPEN	<p>打开 Socket n:</p> <p>按照 Sn_MODE[3:0]所设模式, 初始化和打开 Socket n. 可通过 Sn_STA 查询命令执行状态, 当命令执行成功后, 对应的 Sn_STA 值应为:</p> <table border="1" data-bbox="671 264 1177 499"> <thead> <tr> <th>Sn_MODE[3:0]</th> <th>Sn_STA</th> </tr> </thead> <tbody> <tr> <td>0000 (关闭)</td> <td>0x00 (CLOSE)</td> </tr> <tr> <td>0001 (TCP)</td> <td>0x13 (INIT)</td> </tr> <tr> <td>0010 (UDP)</td> <td>0x22 (UDP)</td> </tr> <tr> <td>0011 (IPRAW)</td> <td>0x32 (IPRAW)</td> </tr> <tr> <td>0100 (MACRAW)</td> <td>0x42 (MACRAW)</td> </tr> </tbody> </table>	Sn_MODE[3:0]	Sn_STA	0000 (关闭)	0x00 (CLOSE)	0001 (TCP)	0x13 (INIT)	0010 (UDP)	0x22 (UDP)	0011 (IPRAW)	0x32 (IPRAW)	0100 (MACRAW)	0x42 (MACRAW)
		Sn_MODE[3:0]	Sn_STA											
		0000 (关闭)	0x00 (CLOSE)											
		0001 (TCP)	0x13 (INIT)											
		0010 (UDP)	0x22 (UDP)											
		0011 (IPRAW)	0x32 (IPRAW)											
0100 (MACRAW)	0x42 (MACRAW)													
02H	LISTEN	<p>设置 Socket n 为监听态:</p> <p>该命令只有在 Socket n 为 TCP 模式, 且为 INIT 态时生效。</p> <p>当 Socket n 设为 LISTEN 态时, Socket n 为 TCP Sever 模式, 等待客户端连接, Sn_STA 寄存器变为 0x14 (LISTEN)。客户端连接成功后, Sn_STA 寄存器值变为 0x17 (ESTABLISHED), 且 Sn_INT[0]置 1。</p> <p>如果连接失败 Sn_STA 寄存器值变为 0x00 (CLOSE), Sn_INT[3]置 1。</p>												
		04H	CONNECT	<p>使能 Socket n 进入连接模式:</p> <p>该命令只有在 Socket n 为 TCP 模式, 且为 INIT 态时生效。</p> <p>使能 Socket 进入连接模式即 TCP Client 模式。执行此命令时, Socket n 根据 Sn_DIP 和 Sn_DPORT 所设值连接到服务器, 当连接成功后 Sn_STA 寄存器值变为 0x17 (ESTABLISHED), 且 Sn_INT[0]置 1。</p> <p>当 Socket 连接失败时:</p> <ol style="list-style-type: none"> <li>1: ARP 失败, 无法获得目的 MAC 地址 (Sn_INT[3]置 1);</li> <li>2: 连接超时, 发送 SYN 包后收不到对方的 SYN/ACK 包, 重传超时后 (Sn_INT[3]置 1);</li> <li>3: 收到 RST 包, 被对方主动断开连接。</li> </ol> <p>出现上述情况时 Sn_STA 寄存器值变为 0x00 (CLOSE)。</p>										
				08H	DISCONNECT	<p>Socket n 断开连接:</p> <p>该命令只有在 Socket n 为 TCP 模式时生效。</p> <p>主动关闭: 主动向对端设备发送 FIN 包;</p> <p>被动关闭: 当收到对方发来的 FIN 包后, 执行此命令回复对方一个 FIN 包。</p> <p>如果 FIN 包并没有收到对方的 ACK 回应, 重传超时后 Sn_INT[3]置 1, Sn_STA 寄存器值变为 0x00 (CLOSE)。</p>								
						10H	CLOSE	<p>关闭 Socket n:</p> <p>Socket n 直接关闭 Socket。</p> <p>如果 Socket n 为 TCP 模式, 此命令会直接关闭 Socket, 不同于标准的 TCP 关闭流程, 该命令关闭时并不会发 FIN 包。</p>						
								20H	SEND	<p>Socket n 发送数据:</p> <p>发送 Socket n 发送缓存区内的数据, 具体操作详参 11.2.4 节数据发送流程。</p>				
21H	SNED_MAC	<p>Socket n 无 ARP 发送数据:</p> <p>该命令只有在 Socket n 为 UDP 模式或 IPRAW 模式时生效。</p> <p>发送流程与 SEND 命令相同, 但发包前不通过 ARP 获取对端 MAC 地址, 直接采用 Sn_DMAC 所设 MAC 地址发送。</p>												
		22H	SEND_KEEP	<p>Socket n 发送 KeepAlive 包:</p> <p>该命令只有在 Socket n 为 TCP 模式时生效。</p> <p>主动向对端设备发送一次心跳包, 如果超时未获得应答, 则 Sn_INT[3]置 1, Sn_STA 寄存器值变为 0x00 (CLOSE)。</p> <p>使用该命令前, Socket n 至少发送一包数据到对端。</p>										
40H	RCV			<p>Socket n 接收数据:</p> <p>完成 Socket n 接收数据的流程, 详细步骤请参考 11.2.3 章节数据接收流程。</p>										



### 7.2.3 Socket n 中断寄存器 (Sn\_INT) [0x0402]

该寄存器用于获取 Socket n 中断状态，当出现 Socket n 中断事件，Sn\_INT 寄存器对应位将被置为 1，对该位写 1 可以清除该中断标志位。

表 7-23 Socket n 中断寄存器

位	名称	描述	访问	默认值
[7:5]	Reserved	保留	RO	0
4	INT_SEND_SUC	发送完成中断： 当 Socket n SEND 命令完成后产生此中断。	RW1Z	0
3	INT_TIMEOUT	超时中断： 当 Socket n ARP 超时，或者 TCP 重传超时后产生此中断。	RW1Z	0
2	INT_RECV	接收中断： 当 Socket n 收到数据包后产生此中断。	RW1Z	0
1	INT_DISCONNECT	断连中断： 当 Socket n 收到对方的 FIN 或 FIN/ACK 或 RST 包后产生此中断。	RW1Z	0
0	INT_CONNECT	连接中断： 当 Socket n 与对端连接成功后产生此中断。	RW1Z	0

### 7.2.4 Socket n 状态寄存器 (Sn\_STA) [0x0403]

该寄存器用于获取 Socket n 当前的状态，以及运行中的临时态。

表 7-24 Socket n 状态值

代码	名称	描述
00H	CLOSE	关闭： Socket n 处在关闭状态。
13H	INIT	TCP 就绪： 当 Socket n 为 TCP 模式，且 open 成功后，Socket n 处在 INIT 态，只有 Socket n 处在 INIT 态时，用户才可以使用 LISTEN 或者 CONNECT 命令进行下一步连接。
14H	LISTEN	TCP 监听： 该状态代表 Socket n 已作为 TCP 服务器处在监听状态，等待 TCP 客户端连接。 连接成功后 Sn_STA 将变为 0x17 (ESTABLISHED)。
17H	ESTABLISHED	Socket n 建立连接： 当 Socket n 处在 LISTEN 状态下，作为 TCP 服务器，被对端 TCP 客户端连接成功；或使用 CONNECT 命令作为 TCP 客户端连接成功，Sn_STA 将变为 0x17 (ESTABLISHED)。 此状态代表连接成功，此时可以正常进行数据传输。
1CH	CLOSE_WAIT	关闭等待： 当 Socket n 处在 TCP 模式，收到对方发来的断开请求包 (FIN 包)，Sn_STA 则会变为 0x10 (CLOSE_WAIT)。 此时 TCP 连接处在一个半断开状态，若要完全断开连接则需要使用 DISCONNT 命令，如果想要直接关闭，可以使用 CLOSE 命令。
22H	UDP	UDP 模式： 表示 Socket n 处在 UDP 模式下。
32H	IP_RAW	IPRAW 模式： 表示 Socket n 处在 IPRAW 模式下。
42H	MAC_RAW	MACRAW 模式： 表示 Socket 0 处在 MACRAW 模式下。 注：只有 Socket 0 可以使用 MACRAW 模式。

TCP 连接过程中的临时状态：

表 7-25 Socket n TCP 临时状态值

代码	名称	描述
15H	SYN_SENT	SYN 请求已发送： 当使用 CONNECT 命令后，Socket n 会主动向对端发送一包 SYN 请求，此时 Sn_STA 变为 SYN_SENT，当对端回复 SYN/ACK 包后，成功建立连接，Sn_STA 变为 0x17 (ESTABLISHED)。 如果在对端没有回复 SYN/ACK 包，SYN 请求重传超时后，则 Sn_INT[3] 置 1，Sn_STA 寄存器值变为 0x00 (CLOSE)。
16H	SYN_RECV	SYN 请求已接收： 当 Socket n 为 TCP Sever 模式处在监听状态，收到对端发来的连接请求包后，Sn_STA 变为 SYN_RECV。此时 Socket n 会向对端回复 SYN/ACK 包，成功后 Sn_STA 变为 0x17 (ESTABLISHED)。 如果发送 SYN/ACK 包后，对方没有及时回复 ACK，导致重传超时，则 Sn_INT[3] 置 1，Sn_STA 寄存器值变为 0x00 (CLOSE)。
18H	FIN_WAIT	这些状态都代表 Socket n 正在关闭。 应用上一般不需要关心这些状态，芯片会自动处理并更新。 当 Socket n 关闭成功，或超时 (Sn_INT[3] 置 1) 后 Sn_STA 寄存器值变为 0x00 (CLOSE)。
1AH	CLOSING	
1BH	TIME_WAIT	
1DH	LAST_ACK	

#### 7.2.5 Socket n 源端口寄存器 (Sn\_PORT) [0x0404-0x0405]

该寄存器用于设置 Socket n 的源端口号。在 UDP 模式或 TCP 模式下需要在 Socket 打开前设置该寄存器。

例如：Socket 0 源端口号为 2000。

地址	0x0404	0x0405
数据	0x07	0xD0

#### 7.2.6 Socket n 目的 MAC 寄存器 (Sn\_DMAC) [0x0406-0x040B]

该寄存器用于获取或设置 Socket n 的目的 MAC 地址。该寄存器有两种使用方式：

1：获取目标 MAC 地址，ARP 成功后，该寄存器存放了 Socket n ARP 过程中得到的 MAC 地址。

2：设置目标 MAC 地址，UDP 模式下或 IPRAW 模式下使用 SNED\_MAC 命令将会按照所设 MAC 地址直接发送。UDP 组播模式时，该寄存器用来配置组播 MAC。

例如：Socket 0 MAC 地址为 0x38:0x3B:0x26:0x44:0x55:0x66。

地址	0x0406	0x0407	0x0408	0x0409	0x040A	0x040B
数据	0x38	0x3B	0x26	0x44	0x55	0x66

#### 7.2.7 Socket n 目的 IP 寄存器 (Sn\_DIP) [0x040C-0x040F]

该寄存器用于获取或设置 Socket n 的目的 IP 地址。该寄存器有两种使用方式：

1：获取目标 IP 地址，TCP Sever 模式下，连接成功后，该寄存器存放了 TCP 客户端的 IP 地址。

2：设置目标 IP 地址，在 UDP 模式下、IPRAW 模式和 TCP 客户端模式下用来配置目的 IP 地址。

UDP 组播模式时，该寄存器用来配置组播 IP。

例如：Socket 0 目的 IP 地址为 192.168.1.200。

地址	0x0040C	0x040D	0x040E	0x040F
数据	0xC0	0xA8	0x01	0xC8

#### 7.2.8 Socket n 目的端口寄存器 (Sn\_DPORT) [0x0410-0x0411]

该寄存器用于获取或设置 Socket n 的目的 IP 端口。该寄存器有两种使用方式：

1：获取目标端口，TCP Sever 模式下，连接成功后，该寄存器存放了 TCP 客户端的端口号。

2：设置目标端口，在 TCP Client 模式下此值必须设置，在 UDP 或 IPRAW 模式下，发送前需将目



标端口号写入该寄存器。

例如：Socket 0 目标端口号为 3000。

地址	0x0410	0x0411
数据	0x0B	0xB8

#### 7.2.9 Socket n 最大传输单元寄存器 (Sn\_MTU) [0x0412-0x0413]

该寄存器用于设置 Socket n 的最大传输单元 (MTU)，在 TCP 模式和 UDP 模式下生效，当用户数据传输长度超过预设 MTU 尺寸时，CH394L 内置协议栈将自动执行数据分割，确保每包数据不超过 MTU 设定值。不同模式下最大 MTU 见下表。

表 7-26 最大 MTU

模式	最大 MTU
TCP	1460
UDP	1472
IPRAW	1480
MA CRAW	1514

例如：设 Socket 0 MTU 为 1000。

地址	0x0412	0x0413
数据	0x03	0xE8

#### 7.2.10 Socket n IP 协议寄存器 (Sn\_IPPRO) [0x0414]

该寄存器用于设置 Socket n 的 IP 协议号。该寄存器仅在 Socket 为 IPRAW 模式时可用，不可设为 0x06 (TCP)，0x11 (UDP) 和 IGMP (0x01)。

#### 7.2.11 Socket n IP 服务类型寄存器 (Sn\_TOS) [0x0415]

该寄存器用于设置 IP 层的 IP 首部中的 TOS 服务字段，应在 Socket n OPEN 之前设置，默认值为 0，默认无需设置。

#### 7.2.12 Socket n IP 生存时间寄存器 (Sn\_TTL) [0x0416]

该寄存器用于设置 IP 层的 IP 首部中的 TTL 服务字段，应在 Socket n OPEN 之前设置。默认值为 0x80，默认无需设置，最大值为 255。

#### 7.2.13 Socket n 接收缓存区大小寄存器 (Sn\_RXBUF\_SIZE) [0x041E]

该寄存器用于设置 Socket n 的接收缓存区大小，默认值为 2 (2KB)。寄存器值可设为 0 到 8，对应 0 到 8KB。CH394L 默认每个 Socket 分配 2K 空间。

Socket 0 接收缓存区		Socket 1 接收缓存区		Socket 2 接收缓存区		Socket 3 接收缓存区	
第 0 块	第 1 块	第 2 块	第 3 块	第 4 块	第 5 块	第 6 块	第 7 块
2K		2K		2K		2K	

CH394L 内部接收缓存区默认分配如上图，一共由 8 个块组成，每个块的长度为 1024 字节。单片机可以自由分配每个 Socket 接收缓存区的大小。Sn\_RXBUF\_SIZE 配置完成后，就会按照 Socket 0 到 3 的顺序将接收缓存重新分配。设置接收缓存区时需注意所有 Socket 接受缓存区总和不能超过 8KB。

#### 7.2.14 Socket n 发送缓存区大小寄存器 (Sn\_TXBUF\_SIZE) [0x041F]

该寄存器用于设置 Socket n 的发送缓存区大小，默认值为 2 (2KB)。寄存器值可设为 0 到 8，对应 0 到 8KB。CH394L 默认每个 Socket 分配 2K 空间。

Socket 0 发送缓存区		Socket 1 发送缓存区		Socket 2 发送缓存区		Socket 3 发送缓存区	
第 0 块	第 1 块	第 2 块	第 3 块	第 4 块	第 5 块	第 6 块	第 7 块
2K		2K		2K		2K	

CH394L 内部发送缓存区默认分配如上图，一共由 8 个块组成，每个块的长度为 1024 字节。单片机可以自由分配每个 Socket 发送缓存区的大小。Sn\_TXBUF\_SIZE 配置完成后，就会按照 Socket 0 到 3 的顺序将发送缓存重新分配。设置发送缓存区时需注意所有 Socket 发送缓存区总和不能超过 8KB。

#### 7.2.15 Socket n 空闲发送缓存长度寄存器 (Sn\_TX\_FS) [0x0420-0x0421]

该寄存器用于获取 Socket n 的发送缓存区内空闲空间的大小。在没有发包的情况下，空闲长度为 Socket n 发送缓存区的大小。

用户在发送数据前须先查询该寄存器，获取空闲空间的大小（具体操作详参 11.2.4 节发送数据流程），所发的包的长度不得大于空闲空间的大小。

例如：空闲缓存区长度为 1000。

地址	0x0420	0x0421
数据	0x03	0xE8

#### 7.2.16 Socket n 发送缓存区读指针寄存器 (Sn\_TX\_RD) [0x0422-0x0423]

该寄存器用于获取 Socket n 的发送读指针，指示了 CH394L 内部协议栈发送读取指针的当前位置。用户一般无需关心该寄存器的值，CH394L 自动管理更新。

Socket n 初始化成功后，该寄存器指向 Socket n 发送缓存区初始地址。用户在完成数据写入发送缓存区，并更新 Sn\_TX\_WR 指针后，通过发出 SEND 指令，CH394L 将自动处理 Sn\_TX\_RD 至 Sn\_TX\_WR 区间的数据传输，具体操作详参 11.2.4 节发送数据流程。

#### 7.2.17 Socket n 发送缓存区写指针寄存器 (Sn\_TX\_WR) [0x0424-0x0425]

该寄存器用于获取或设置 Socket n 的发送写指针，需要在每次发送数据流程前获取当前 Socket n 的发送写指针，在写入待发送数据后更新该寄存器值。具体操作详参 11.2.4 节发送数据流程。

例如：Socket n 发送缓存区写指针为 2048。

地址	0x0424	0x0425
数据	0x08	0x00

#### 7.2.18 Socket n 接收数据长度寄存器 (Sn\_RX\_RS) [0x0426-0x0427]

该寄存器用于获取 Socket n 接收到的数据的长度。具体操作详参 11.2.3 节接收数据流程。

例如：Socket n 接收数据长度为 1024。

地址	0x0426	0x0427
数据	0x04	0x00

#### 7.2.19 Socket n 接收缓存区读指针寄存器 (Sn\_RX\_RD) [0x0428-0x0429]

该寄存器用于获取或设置 Socket n 接收缓存区读指针，指示了用户接收缓存区读取指针的当前位置。

需要在每次接收数据流程前获取当前 Socket n 的读指针，在读取完数据后更新该寄存器值。具体操作详参 11.2.3 节接收数据流程。

例如：Socket n 接收缓存区读指针为 2048。

地址	0x0428	0x0429
数据	0x08	0x00

#### 7.2.20 Socket n 接收缓存区写指针寄存器 (Sn\_RX\_WR) [0x042A-0x042B]

该寄存器用于获取 Socket n 的接收缓存区写指针，指示了 CH394L 内部协议栈接收数据写指针的当前位置。用户一般无需关心该寄存器的值，CH394L 自动管理更新。

Socket n 初始化成功后，该寄存器指向 Socket n 接收缓存区初始地址。当 CH394L 从以太网接收到符合要求的数据并载入接收缓存区后，会相应更新 Sn\_RX\_WR 指针。具体操作详参 11.2.3 节接收数据流程。

## 7.2.21 Socket n 中断使能寄存器 (Sn\_INTE) [0x042C]

该寄存器用于控制 Socket n 哪些中断源能够触发中断，默认为 0xFF。每个中断事件使能位对应 Socket n 中断寄存器 (Sn\_INT) 一个位。

当中断事件产生后，只有 Sn\_INTE 中对应位为 1 时，且 SINTE 第 n 位为 1 时 (Socket n 中断使能) CH394L 才会拉低 INT 脚，产生中断。

表 7-27 Socket n 中断使能寄存器

位	名称	描述	访问	默认值
[7:5]	Reserved	保留	RO	111b
4	INT_SEND_SUC	发送完成中断： 使能该位即允许产生发送完成中断。	RW	1
3	INT_TIMEOUT	超时中断： 使能该位即允许产生超时中断。	RW	1
2	INT_RECV	接收中断： 使能该位即允许产生接收中断。	RW	1
1	INT_DISCONNECT	断连中断： 使能该位即允许产生断连中断。	RW	1
0	INT_CONNECT	连接中断： 使能该位即允许产生连接中断。	RW	1

## 7.2.22 Socket n IP 分片寄存器 (Sn\_IPF) [0x042D-0x042E]

该寄存器用于设置 Socket n IP 分片字段，默认为 0x4000。

例如：Socket n IP 分片字段为 0x4000 (不分片)。

地址	0x042D	0x042E
数据	0x40	0x00

## 7.2.23 Socket n 扩展配置寄存器 (Sn\_ESET) [0x042F]

该寄存器用于配置 Socket n 的扩展模式，须在 Socket n 打开前设置。

表 7-28 Socket n 模式寄存器

位	名称	描述	访问	默认值
7	Reserved	保留	RO	-
6	MBCASTB	广播包屏蔽位： 0：不屏蔽广播包； 1：屏蔽广播包。 该位置 1 后，在 MACRAW 模式下，Socket 0 将不接收网络上的广播包，仅 Socket 0 可用。	RW	0
5	MMB	组播包屏蔽位： 0：不屏蔽组播包； 1：屏蔽组播包。 该位置 1 后，在 MACRAW 模式下，Socket 0 将不接收网络上的组播包，仅 Socket 0 可用。	RW	0
4	MIP6B	IPv6 包屏蔽位： 0：不屏蔽 IPv6 包； 1：屏蔽 IPv6 包。 该位置 1 后，在 MACRAW 模式下，Socket 0 将不接收网络上的 IPv6 包，仅 Socket 0 可用。	RW	0
[3:2]	Reserved	保留	RO	0
1	UBCASTB_PSH	当 Socket n 工作在 UDP 模式时： 0：不屏蔽广播包； 1：屏蔽广播包。 该位置 1 后，在 UDP 模式下，Socket n 将不	RW	0

		接收网络上的广播包。 当 Socket n 工作在 TCP 模式时： 0: 不强制 PSH 位； 1: 强制 PSH 位。 该位置 1 后，在 TCP 模式下，Socket n 发送的每一个数据包都将设置 PSH 标志。		
0	UUB	当 Socket n 工作在 UDP 组播模式时： 0: 不屏蔽单播包； 1: 屏蔽单播包。 该位置 1 后，在 UDP 组播模式下，Socket n 将不接收网络上的单播包。	RW	0

#### 7.2.24 Socket n KeepAlive 时间寄存器 (Sn\_KEEPA\_LIVE) [0x0430]

该寄存器用于设置 Socket n KeepAlive 时间，仅在 TCP 模式下生效。默认值为 0，表示 KeepAlive 功能未激活，欲启用自动 KeepAlive，需设定该寄存器值大于零。寄存器单位时间为 5 秒，即寄存器值为 1 时，KeepAlive 间隔时间为 5 秒，为 2 时为 10 秒……，以此类推。

当 Socket n 处于 TCP 模式且与对端成功建立连接 (Sn\_STA 状态为 0x17) 且 Socket n 发送完一包后进入空闲 (无数据流通) 状态时，系统将按照 Sn\_KEEPA\_LIVE 指定的时间间隔自动发送 KeepAlive 包。

若连续发送 Keepalive 包未获对方 ACK 确认，在超出重传时限后，系统将触发超时中断。

当 Sn\_KEEPA\_LIVE 为 0 时，用户可以通过 SEND\_KEEP 命令手动发送 KeepAlive 包。当 Sn\_KEEPA\_LIVE 不为 0 时，SEND\_KEEP 命令不生效。

例如：Socket n KeepAlive 时间寄存为 4 (空闲时每隔 20 秒自动发送一包 KeepAlive 包)。

地址	0x0430
数据	0x04

#### 7.2.25 Socket n 重传时间寄存器 (Sn\_RT) [0x0432-0x0433]

该寄存器用于设置 Socket n TCP 通讯中的重传时间，以及 ARP 的重传时间。默认情况下，该寄存器的值为 0。当该寄存器为 0 时，Socket n 的重传时间由 RTIME 寄存器来确定；而当该寄存器不为 0 时，则 Socket n 的重传时间将根据该寄存器的设置值来决定。

Sn\_RT 需要两个字节的时间值，单位为 0.1ms。Sn\_RT 寄存器可设为任意值，CH394L 会自动向下取整为 1000 的倍数。

例如：设重传时间为 5000 (0x1388)，即 500ms。

地址	0x0432	0x0433
数据	0x13	0x88

#### 7.2.26 Socket n 重传次数寄存器 (Sn\_RC) [0x0434]

该寄存器用于设置重传超时次数。默认情况下，该寄存器的值为 0。当该寄存器为 0 时，Socket n 的重传次数由通用寄存器 RCNT 寄存器来决定；而当该寄存器不为 0 时，则 Socket n 的重传次数将根据该寄存器的设置值来决定。

## 8、CH394L 功能说明

### 8.1 单片机通讯接口

CH394L 与单片机之间支持两种通讯接口：SPI 接口、8 位并行接口。在芯片上电复位时，CH394L 将采样 MOD[3:0] 引脚的状态，根据这个四个引脚电平选择通讯接口，参考下表（表中 0 代表低电平或悬空，1 代表高电平，x 代表不关心该引脚状态）。

表 8-1 接口选择

MOD[3:0]	说明
0000	采用 SPI 接口
010x	采用 8 位被动并行接口

CH394L 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 CH394L 的中断请求。

### 8.2 SPI 串行接口

SPI 同步串行接口信号线包括：SPI 片选输入引脚 CS、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，CH394L 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。

CH394L 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 MOSI 驱动，SDO 引脚则连接到单片机的 SPI 数据输入引脚 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。CH394L 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。

CH394L 的 SPI 接口支持 SPI 模式 0 和 SPI 模式 3，CH394L 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 数据帧格式及示例详参 5.1 章节。

### 8.3 8 位并行接口

并口信号线包括：8 位双向数据总线 D7~D0、读选通输入引脚 PRD、写选通输入引脚 PWR、片选输入引脚 CS 以及 2 位地址线 A0、A1（和 SDI、SDO 脚复用）。通过被动并行接口，CH394L 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。

下表为并口 I/O 操作的真值表（表中 X 代表不关心此位）。

表 8-2 并口操作真值表

CS	PWR	PRD	A0	A1	D7-D0	对 CH394L 芯片的实际操作
1	X	X	X	X	X	未选中 CH394L，不进行任何操作
0	1	1	X	X	X	虽然选中但无操作，不进行任何操作
0	0	1	1	0	输入	写入高 8 字节地址
0	0	1	0	1	输入	写入低 8 字节地址
0	0	1	1	1	输入	写数据
0	1	0	1	1	输出	读数据

并口数据帧格式及示例详参 5.2 章节。

### 8.2 其它硬件

CH394L 芯片内部集成了 10M/100M 以太网 PHY、MAC、SPI-Slave 控制器、被动并行接口、SRAM、高速 MCU 和 PLL 倍频器、电源上电复位电路等。

CH394L 芯片的 MDIRP、MDIRN、MDITP、MDITN 为以太网的信号线。CH394L 的 PHY 支持 MDI/MDIX 线路自动转换。

CH394L 芯片内置了电源上电复位电路，也可通过 RSTB 引脚拉低控制复位。RSTB 引脚用于从外部输入异步复位信号；当 RSTB 引脚为低电平时，CH394L 芯片被复位；当 RSTB 引脚恢复为高电平后 CH394L 将进入初始化阶段约 15ms，在这段时间内主机禁止操作 CH394L。



## 9、CH394L 参数

### 9.1 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）

表 9-1 绝对最大值

名称	参数说明	最小值	最大值	单位
$T_A$	工作时的环境温度	-40	85	$^{\circ}\text{C}$
$T_J$	结温度范围	-40	100	$^{\circ}\text{C}$
$T_S$	存储时的环境温度	-55	150	$^{\circ}\text{C}$
$AV_{DD33}$	工作电源电压	-0.4	4.0	V
$V_{DD10}$	I/O 电源电压	-0.4	4.0	V
$AV_{DDK}$	内核模拟电路的电源退耦端	-0.4	1.5	V
$DV_{DDK}$	内核数字电路的电源退耦端	-0.4	1.5	V
$V_{ETH}$	ETH 物理信号引脚上的电压	-0.4	$AV_{DD33}+0.4$	V
$V_{IN}$	引脚上的输入电压	-0.4	$V_{DD10}+0.4$	V
$V_{ESD(HBM)}$	普通 I/O 引脚的 ESD 静电放电电压 (HBM)	6K		V
$I_{IO}$	I/O 引脚上的吸收电流		20	mA
	I/O 引脚上的输出电流		20	

### 9.2 电气参数

表 9-2 电气参数 ( $AV_{DD33} = 3.3V$ ,  $V_{DD10} = 3.3V$ ,  $T_A = 25^{\circ}\text{C}$ )

名称	参数说明	最小值	典型值	最大值	单位
$AV_{DD33}$	工作电源电压	3.2	3.3	3.4	V
$V_{DD10}$	I/O 电源电压	1.7	3.3	3.6	V
$V_{IL}$	I/O 引脚, 输入低电平电压	$V_{DD10} = 3.3V$	0	0.8	V
		$V_{DD10} = 1.8V$	0	0.6	
$V_{IH}$	I/O 引脚, 输入高电平电压	$V_{DD10} = 3.3V$	2.0	$V_{DD10}$	V
		$V_{DD10} = 1.8V$	1.2	$V_{DD10}$	
$V_{OL}$	低电平输出电压		0.4	0.6	V
$V_{OH}$	高电平输出电压	$V_{DD10}-0.6$	$V_{DD10}-0.4$		V
$V_{hys}$	I/O 施密特触发器电压迟滞		150		mV
$C_{IO}$	I/O 引脚电容		5		pF
$R_{PU}$	上拉等效电阻	30	40	55	k $\Omega$
$R_{PD}$	下拉等效电阻	30	40	55	k $\Omega$
$t_{f(10)out}$	输出高至低电平的下降时间			6.5	ns
$t_{r(10)out}$	输出低至高电平的上升时间			11	ns

注：以上均为设计参数保证。

### 9.3 功耗

表 9-3 功耗 ( $AV_{DD33} = 3.3V$ ,  $V_{DD10} = 3.3V$ ,  $T_A = 25^{\circ}\text{C}$ )

符号	参数	条件 (所有电流, 含网络变压器)	典型值	单位
$I_{DD0}$	传输状态下的供应电流	100BASE-TX 通路链接成功并且在收发通道上有数据包	77.8	mA
		10BASE-TX 通路链接成功并且在收发通道上有数据包	60.9	
$I_{DD1}$	空闲状态下的供应电流	100BASE-TX 通路链接成功并且在收发通道上无任何数据包	77.6	mA

		10BASE-TX通路链接成功并且在收发通道上无任何数据包	49.9	
$I_{DD2}$	断开状态下的供应电流	100BASE-TX和10BASE-TX通路均未链接成功且PHY处于自动协商状态。	55.9	mA
$I_{DD3}$	PHY掉电状态下的供应电流		15.4	

## 9.4 交流电气特性和时序

### 9.4.1 SPI 时序

图 9-1 SPI 模式 0 时序图

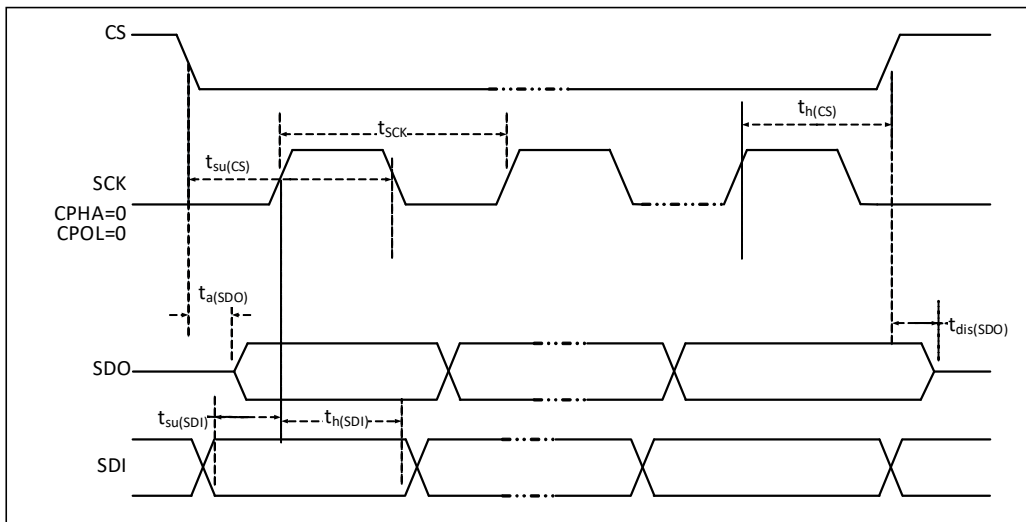


图 9-2 SPI 模式 3 时序图

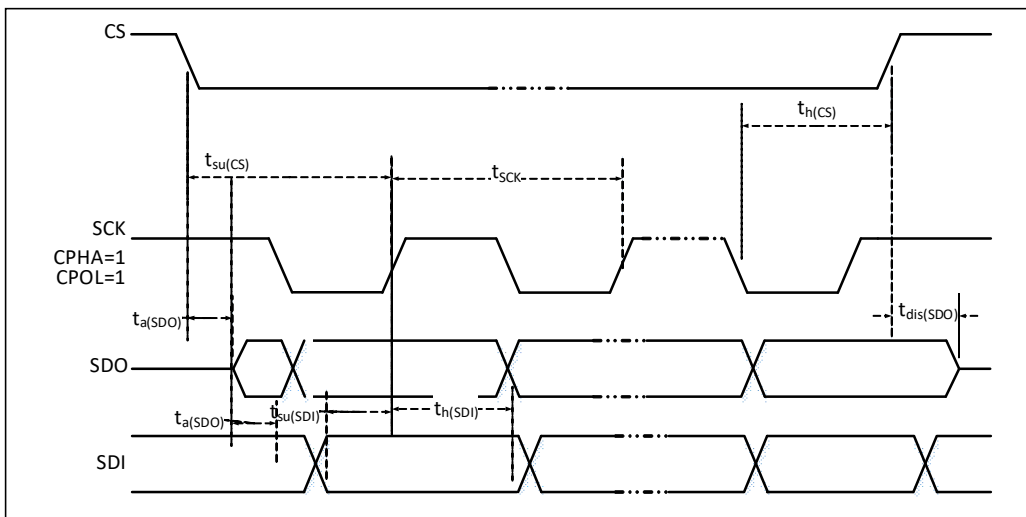


图 9-3 SPI 数据帧时序间隔

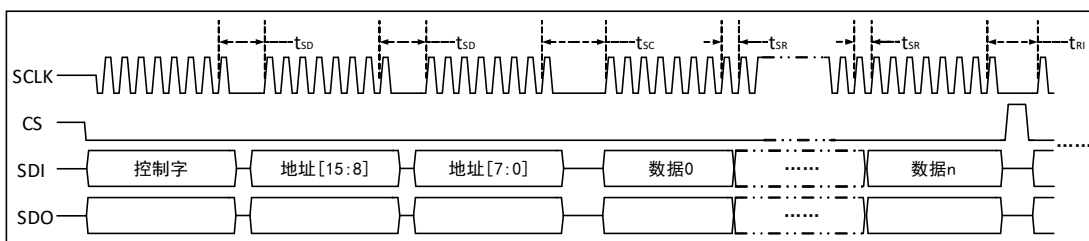




表 9-4 SPI 参数表 ( $V_{DD33} = 3.3V$ ,  $V_{DD10} = 3.3V$ ,  $T_A = 25^\circ C$ )

符号	参数	条件	最小值	最大值	单位
$f_{SCK}$	SPI 时钟频率 <sup>(1)</sup>			40	MHz
$t_{SU(SCS)}$	SCS 建立时间		17		ns
$t_{h(SCS)}$	SCS 保持时间		17		ns
$t_{SU(SD1)}$	数据输入建立时间		4		ns
$t_{h(SD1)}$	数据输入保持时间		2		ns
$t_a(SD0)$	数据输出访问时间		0	8	ns
$t_{dis(SD0)}$	数据输出禁止时间		0	10	ns
$t_{SC}$	偏移地址与第一个数据字节间所需间隔	写寄存器	0		ns
		读寄存器	100		
		写收发缓存区	0		
		读收发缓存区	180		
$t_{SD}$	控制字和偏移地址每个字节间隔		100		ns
$t_{SR}$	数据字节间所需间隔	写寄存器	0		ns
		读寄存器	100		
		读写收发缓存区	0		
$t_{RI}$	数据帧操作所需间隔 <sup>(1)</sup>	读写寄存器	$0.3 * N^{(2)} - 0.3$		us
		读写收发缓存区	0		

注:

1. 在对寄存器进行读写操作时, 主机应在完成一次操作后, 等待一个  $t_{RI}$  处理时间, 在处理时间内主机禁止操作其他寄存器。

2.  $N$  为数据段中字节的个数。

3. 如 SPI 速率不高于 5MHz, 由于数据传输自带的延时已经足够, 因此无需额外考虑  $t_{SC}$ 、 $t_{SD}$ 、 $t_{SR}$  等延时参数, 用户无需手动添加延时。

#### 9.4.2 并口时序

图 9-4 并口模式时序图

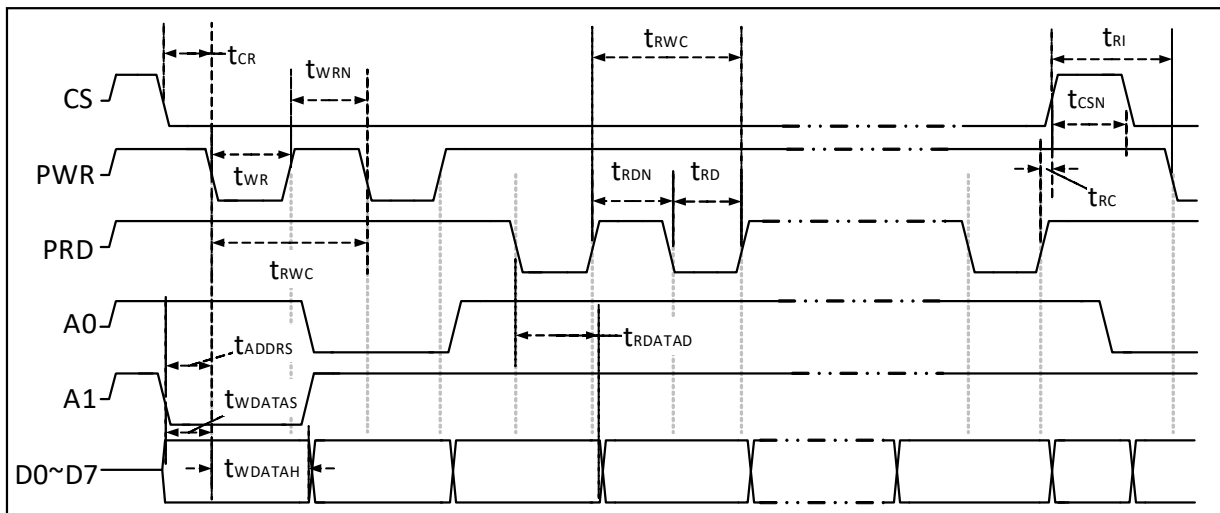
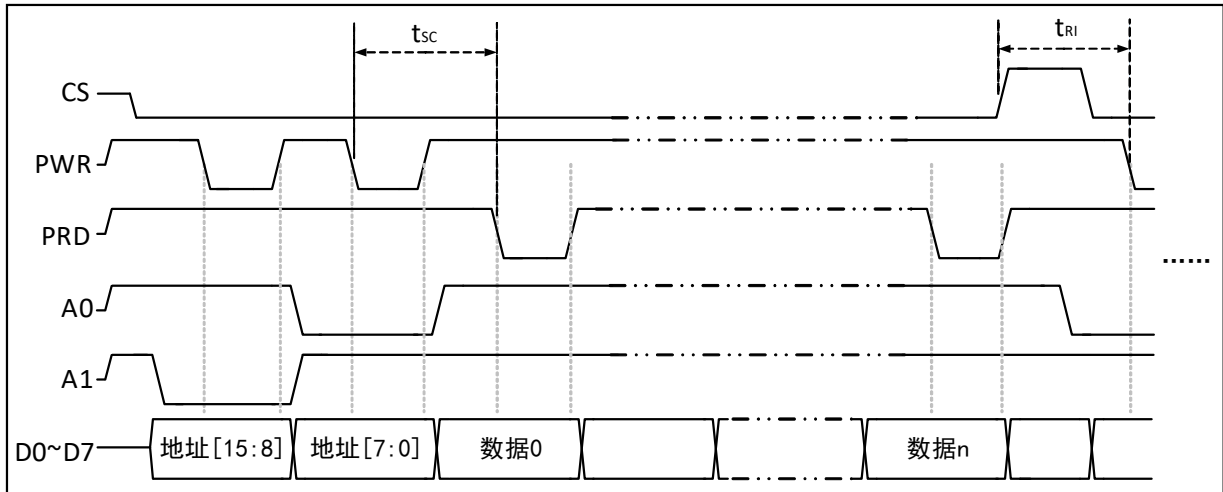


图 9-5 并口数据帧时序图

表 9-5 并口参数表 ( $V_{DD33} = 3.3V$ ,  $V_{DD10} = 3.3V$ ,  $T_A = 25^\circ C$ )

符号	参数	条件	最小值	最大值	单位
$t_{CR}$	CS 建立时间		1		ns
$t_{RC}$	CS 保持时间		0		ns
$t_{ADDRS}$	地址建立时间		1		ns
$t_{WDATAS}$	数据输入建立时间		1		ns
$t_{WDATAH}$	数据输入保持时间		10		ns
$T_{RDATAO}$	数据输出延时时间		15		ns
$t_{RDN}$	PRD 高电平保持时间		25		ns
$t_{RD}$	PRD 低电平保持时间		30		ns
$t_{WRN}$	PWR 高电平保持时间		25		ns
$t_{WR}$	PWR 低电平保持时间		30		ns
$t_{RWC}$	读写周期时间		70		ns
$t_{CSN}$	CS 高电平保持时间		35		ns
$t_{SC}$	偏移地址与第一个数据字节间所需间隔	写寄存器	0		ns
		读寄存器	100		
		写收发缓存区	0		
		读收发缓存区	180		
$t_{RI}$	数据帧操作所需间隔 <sup>(1)</sup>	读写寄存器	$0.6 * N^{(2)} + 0.3$		us
		读写收发缓存区	0		

注:

1. 在对寄存器进行读写操作时, 主机应在完成一次操作后, 等待一个  $t_{RI}$  处理时间, 在处理时间内主机禁止操作其他寄存器。

2.  $N$  为数据段中字节的个数。

#### 9.4.3 振荡器及晶振时序

表 9-6 振荡器及晶振时序参数表

符号	参数	条件	最小值	典型值	最大值	单位
TCKF	晶体频率	建议 30ppm 内	24.999	25	25.001	MHz
TPWH	时钟脉冲宽度高		15	20	25	ns
TPWL	时钟脉冲宽度低		15	20	25	ns

注:  $X1$  和  $X0$  引脚已分别内置负载电容 12pF 的外部晶体所需的两个振荡电容, 外部只需要晶体; 如果另选负载电容 20pF 的外部晶体, 那么  $X1$  和  $X0$  需要分别对地额外加 15pF 的振荡电容。

## 9.4.4 复位时序

表 9-7 复位时序参数表

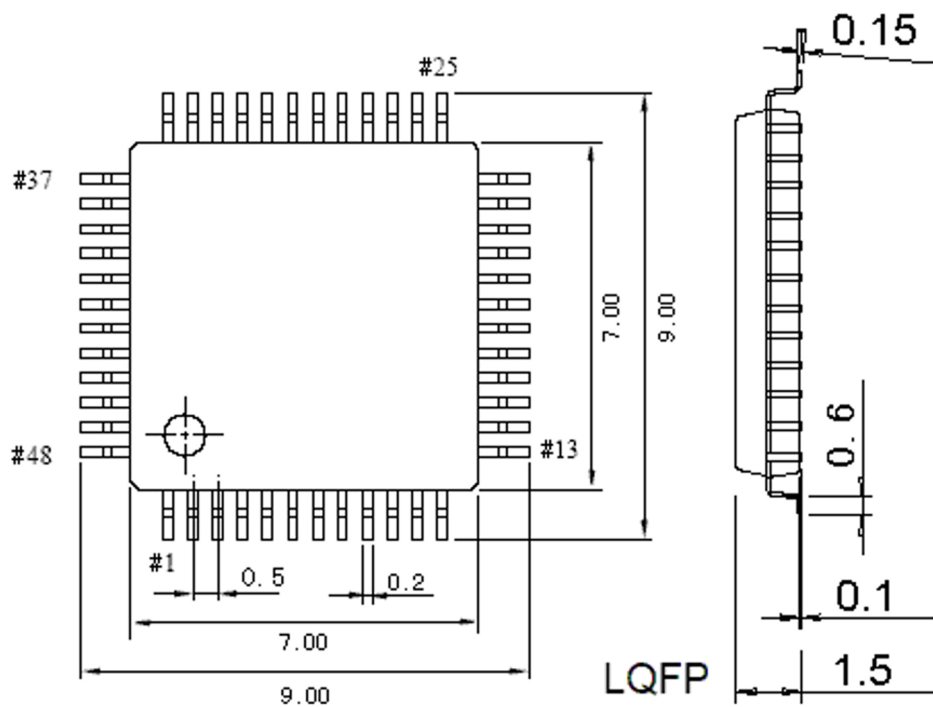
符号	参数	最小值	典型值	最大值	单位
$t_{RSTTEMP0}$	RSTB 低电平宽度	1			us
$t_{RSTTEMP1}$	RSTB 高电平至主机可操作	12	15	19	ms
$t_{RSTTEMP2}$	复位命令生效至主机可操作	12	15	19	ms
$t_{RSTTEMP3}$	电源上电复位至主机可操作	27	30	35	ms

## 10、CH394L 封装信息

说明：尺寸标注的单位是 mm（毫米）。

引脚中心间距是标称值，没有误差，除此之外的尺寸误差不大于 $\pm 0.2\text{mm}$ 。

### 10.1 LQFP48



## 11、CH394L 应用

### 11.1 应用基础

CH394L 内部集成了 IPv4、ARP、ICMP、IGMP、UDP、TCP 等协议。

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 CH394L 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。

### 11.2 应用参考步骤

本章节介绍收发数据的操作流程，具体可以参考例程。

#### 11.2.1 初始化 CH394L，必要操作

表 11-1 初始化 CH394L 步骤

步骤	动作	操作寄存器	地址	操作
1	设置 CH394L 的 MAC 地址	MAC	0x0009	写入 MAC 地址
2	设置 CH394L 的 IP 地址	IP	0x000F	写入 IP 地址
3	设置网关地址	GWIP	0x0001	写入网关地址
4	设置子网掩码地址	SMIP	0x0005	写入子网掩码地址

步骤 1 一般不需要，CH394L 在出厂时，已经烧录了 IEEE 分配的 MAC 地址。

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。

#### 11.2.2 配置 Socket 收发缓存区

CH394L 拥有 4 个 Socket，每个 Socket 默认分配 2K 接收缓存区和 2K 发送缓存区，共 8K 接收缓存区和 8K 发送缓存区。

用户默认无需配置缓冲区大小，但如果根据实际需求需要调整缓冲区大小，可以通过以下两种方式进行更改：

① 通过 SRXBUF\_SIZE 寄存器一次更改 4 个 Socket 的接收缓存区大小，或通过 STXBUF\_SIZE 一次更改 4 个 Socket 的发送缓存区大小。

② 通过 Sn\_RXBUF\_SIZE 更改 Socket n 的接收缓存区大小，或通过 Sn\_TXBUF\_SIZE 更改 Socket n 的发送缓存区大小。

用户可以选择上述任意一种方式进行配置。所有 Socket 的发送缓冲区总和及接收缓冲区总和均不得超过 8K。配置缓冲区后，CH394L 将根据寄存器中设定的值依次为各个 Socket 分配缓冲区空间。若总和超过 8K，则编号靠后的 Socket 可能无法获得其预设大小的缓冲区。

## 11.2.3 接收数据流程

在进行数据接收操作时，须：

表 11-2 接收数据步骤

步骤	动作	操作寄存器	地址	操作
1	获取接收到的数据的长度	Sn_RX_RS	0x0426+n*0x100	读出接收长度
2	获取数据在缓存区内的起始地址	Sn_RX_RD	0x0428+n*0x100	读出原地址 A <sub>0</sub>
3	原地址 A <sub>0</sub> 加接收缓存区基地址 (0x6000) 计算出数据的实际地址 A <sub>0</sub> +0x6000	-	-	-
4	从 A <sub>0</sub> +0x6000 取特定长度的数据 L <sub>0</sub>	-	A <sub>0</sub> +0x6000	读数据
5	将原地址 A <sub>0</sub> + 读的长度 L <sub>0</sub> 写回 Sn_RX_RD 寄存器	Sn_RX_RD	0x0428+n*0x100	写入 A <sub>0</sub> +L <sub>0</sub>
6	向 Sn_CTRL 寄存器写入 RECV 命令	Sn_CTRL	0x0401+n*0x100	写入 RECV 命令
7	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功

例：Socket2 收到了一包长度为 10 的数据

表 11-3 接收数据示例

步骤	动作	操作寄存器	地址	操作
1	读 Sn_RX_RS 寄存器获取到接收数据的长度	Sn_RX_RS	0x0626	读出 10，接收长度为 10
2	获取数据在缓存区内的起始地址	Sn_RX_RD	0x0628	读出 0x1012，读指针地址
3	计算实际地址： 0x1012+0x6000=0x7012	-	-	-
4	从 0x7012 读 10 字节数据	-	0x7012	读 10 位数据
5	将原地址加上读的长度写回 Sn_RX_RD 寄存器： 0x1012+0x0A=0x101C	Sn_RX_RD	0x0628	写入 0x101C
6	向 Sn_CTRL 寄存器写入 RECV 命令	Sn_CTRL	0x0601	写入 0x40
7	等待命令完成	Sn_CTRL	0x0601	读出为 0x00 代表执行成功

## 11.2.4 发送数据流程

在进行数据发送操作时，须：

表 11-4 发送数据步骤

步骤	动作	操作寄存器	地址	操作
1	确保发送缓存区有足够的空间	Sn_TX_FS	0x0420+n*0x100	读出空闲发送缓存区长度 L <sub>0</sub>
2	获取数据应被写入的缓存区原地址	Sn_TX_WR	0x0424+n*0x100	读出原地址 A <sub>0</sub>
3	原地址 A <sub>0</sub> 加发送缓存区基地址 (0x4000) 计算出数据的实际地址 A <sub>0</sub> +0x4000	-	-	-
4	向 A <sub>0</sub> +0x4000 写特定长度的数据 L <sub>1</sub> (L <sub>1</sub> 需要小于 L <sub>0</sub> )	-	A <sub>0</sub> +0x4000	写数据
5	将原地址 A <sub>0</sub> + 写入的长度 L <sub>1</sub> 写回 Sn_TX_WR 寄存器	Sn_TX_WR	0x0424+n*0x100	写入 A <sub>0</sub> +L <sub>1</sub>
6	向 Sn_CTRL 寄存器写入 SEND 命令	Sn_CTRL	0x0401+n*0x100	写入 SEND 命令
7	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功

例：向 Socket1 写一包长度为 5 的数据

表 11-5 发送数据示例

步骤	动作	操作寄存器	地址	操作
1	确保发送缓存区有足够的空间	Sn_TX_FS	0x0520	读出 2048, 空闲空间为 2048
2	获取数据应被写入的缓存区原地址	Sn_TX_WR	0x0524	读出 0x0820, 写指针地址
3	计算实际地址: 0x0820+0x4000=0x4820	-	-	-
4	向 0x4820 写 5 字节数据	-	0x4820	写 5 位数据
5	将原地址加上写入的长度写回 Sn_TX_WR 寄存器: 0x0820+0x05=0x0825	Sn_TX_WR	0x0524	写入 0x0825
6	向 Sn_CTRL 寄存器写入 SEND 命令	Sn_CTRL	0x0501	写入 0x20
7	等待命令完成	Sn_CTRL	0x0501	读出为 0x00 代表执行成功

### 11.2.5 初始化 Socket 为 UDP 模式

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 CH394L 写入若干字节数据流后, CH394L 将数据流封装在 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, CH394L 会自动将数据分包成合适的大小发送。

当 CH394L 接收到 UDP 报文后, 将 UDP 数据复制到 Socket 接收缓冲区中并产生 INT\_RECV 中断, 单片机收到此中断后, 可以读取缓冲区的数据 (参考 11.2.3 章节接收数据流程)。由于 UDP 模式下 CH394L 无法提供流控, 建议接收到的数据及时快速的读完, 以免数据丢失。

CH394L 支持 UDP 单播、UDP 广播和 UDP 组播。

CH394L 会在接收到数据的头部添加 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。

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

CH394L 直接将数据发送给指定的目的 IP 和端口, 单片机在发送前设置目的 IP 和目的端口即可。

初始化 Socket 为 UDP 模式步骤如下表:

表 11-6 初始化 Socket 为 UDP 模式步骤

步骤	动作	操作寄存器	地址	操作
1	设置 Socket n 为 UDP 模式	Sn_MODE	0x0400+n*0x100	写入 0x02
2	设置 Socket n 源端口号	Sn_PORT	0x0404+n*0x100	写入源端口号
3	打开 Socket n	Sn_CTRL	0x0401+n*0x100	写入 0x01 命令
4	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功

当 Socket 设为 UDP 单播或者广播时, 在数据发送前, 设置 Sn\_DIP 和 Sn\_DPORT 为对应的单播或广播地址即可。

当 Socket 设为 UDP 组播时, 应在 Socket 初始化之前, 将 Sn\_DIP 与 Sn\_DMAC 分别设定为指定的组播 IP 地址及 MAC 地址, 并且应将 Sn\_MODE 寄存器中的 MU 位置为 1 以启用组播功能。在 Socket 建立连接期间, CH394L 将依据 Sn\_MODE 寄存器中的 TNA\_IGMP 位选择 IGMP 协议版本, 并据此发送组成员加入 (JOIN) 消息。

用户可通过 CLOSE 命令直接关闭 Socket。



## 11.2.6 初始化 Socket 为 TCP Sever 模式

初始化步骤如下：

表 11-7 初始化 Socket 为 TCP Sever 模式步骤

步骤	动作	操作寄存器	地址	操作
1	设置 Socket n 为 TCP 模式	Sn_MODE	0x0400+n*0x100	写入 0x01
2	设置 Socket n 源端口号	Sn_PORT	0x0404+n*0x100	写入源端口号
3	打开 Socket n	Sn_CTRL	0x0401+n*0x100	写入 0x01 命令
4	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功
5	设 Socket n 为监听模式	Sn_CTRL	0x0401+n*0x100	写入 0x02 命令
6	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功
7	等待 TCP 客户端连接	-	-	-
8	TCP 客户端连接成功，Socket n 变为 ESTABLISHED 态	Sn_STA	0x0403+n*0x100	值变为 0x17

在 TCP 服务器模式下，Socket 将维持在 LISTEN 状态，持续侦听来自客户端的连接请求而不触发超时中断机制，直至接收到客户端发出的 SYN 数据包。如果 TCP 握手过程完成并建立连接，GH394L 将触发 INT\_CONNECT 中断；如果连接失败（即未能接收到对方确认的 ACK 数据包），则会触发 INT\_TIMEOUT 中断。

在成功建立连接之后，用户可以通过查询 Sn\_DIP 和 Sn\_DPORT 寄存器来获取与之通信的客户端的 IP 地址及端口号信息。

当 CH394L 接收到对端发送的 FIN 数据包时，表明对方请求终止连接，此时系统将产生 INT\_DISCONNECT 中断。用户可以使用 DISCONNECT 指令来发送 FIN 包，从而完成连接的正常关闭流程。

用户可通过 DISCONNECT 命令或 CLOSE 命令主动关闭 Socket。

## 11.2.7 初始化 Socket 为 TCP Client 模式

初始化步骤如下：

表 11-8 初始化 Socket 为 TCP Client 模式步骤

步骤	动作	操作寄存器	地址	操作
1	设置 Socket n 为 TCP 模式	Sn_MODE	0x0400+n*0x100	写入 0x01
2	设置 Socket n 源端口号	Sn_PORT	0x0404+n*0x100	写入源端口号
3	打开 Socket n	Sn_CTRL	0x0401+n*0x100	写入 0x01 命令
4	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功
5	设置目的 IP	Sn_DIP	0x040C+n*0x100	写入目的 IP
6	设置目的端口	Sn_DPORT	0x0410+n*0x100	写入目的端口
7	启动 Socket 连接	Sn_CTRL	0x0401+n*0x100	写入 0x04 命令
8	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功
9	等待连接成功	-	-	-
10	TCP 连接成功，Socket n 变为 ESTABLISHED 态	Sn_STA	0x0403+n*0x100	值变为 0x17

在发送完连接命令后，CH394L 将主动向服务器发送 SYN 数据包以发起 TCP 三次握手过程。如果 TCP 握手过程完成并建立连接，CH394L 将触发 INT\_CONNECT 中断；如果连接失败（即未能接收到对方确认的 SYN/ACK 数据包），则会触发 INT\_TIMEOUT 中断。

当 CH394L 接收到对端发送的 FIN 数据包时，表明对方请求终止连接，此时系统将产生 INT\_DISCONNECT 中断。用户可以使用 DISCONNECT 指令来发送 FIN 包，从而完成连接的正常关闭流程。

用户可通过 DISCONNECT 命令或 CLOSE 命令主动关闭 Socket。



### 11.2.7 初始化 Socket 为 IPRAW 模式

初始化步骤如下：

表 11-9 初始化 Socket 为 IPRAW 模式步骤

步骤	动作	操作寄存器	地址	操作
1	设置 Socket n 为 IPRAW 模式	Sn_MODE	0x0400+n*0x100	写入 0x03
2	设置 Socket n 的 IP 协议类型	Sn_IPPRO	0x0414+n*0x100	设置协议类型
3	打开 Socket n	Sn_CTRL	0x0401+n*0x100	写入 0x01 命令
4	等待命令完成	Sn_CTRL	0x0401+n*0x100	读出为 0x00 代表执行成功

IPRAW 协议类型不涵盖 IPv6、TCP 以及 UDP 协议。

当 Socketn 的协议类型配置为 ICMP（标识符为 0x01）时，Socket n 将会捕获由 CH394L 接收的所有 ICMP 数据包，在此情况下，CH394L 将不会自动回应来自对端的 ICMP 数据包（例如，PING 请求）。所有接收到的 ICMP 数据报将会被存储于 Socket n 的接收缓冲区中。

CH394L 会在接收到数据的头部添加 6 个字节的信表，单片机可以根据信表来获得数据包的来源信息。

IP 地址	数据包长度	数据
4 Byte	2 Byte	N Byte

### 11.2.8 初始化 Socket 为 MACRAW 模式

初始化步骤如下：

表 11-10 初始化 Socket 为 MACRAW 模式步骤

步骤	动作	操作寄存器	地址	操作
1	设置 Socket0 为 MACRAW 模式	Sn_MODE	0x0400	写入 0x04
2	打开 Socket0	Sn_CTRL	0x0401	写入 0x01 命令
3	等待命令完成	Sn_CTRL	0x0401	读出为 0x00 代表执行成功

只有 Socket0 支持 MACRAW 模式，在 MACRAW 模式下，CH394L 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。MACRAW 模式下，Socket 0 不会接收其他 Socket 接受的包。

CH394L 会在接收到数据的头部添加 2 个字节的信表。

数据包长度	数据
2 Byte	N Byte

### 11.2.9 网络诊断功能

CH394L 集成了网络诊断功能，允许用户通过发送 PING 命令与 ARP 命令来验证对端设备的存在性和网络连通性。用户可以通过网络诊断寄存器（ND）发送 PING 请求包或者 ARP 请求包：

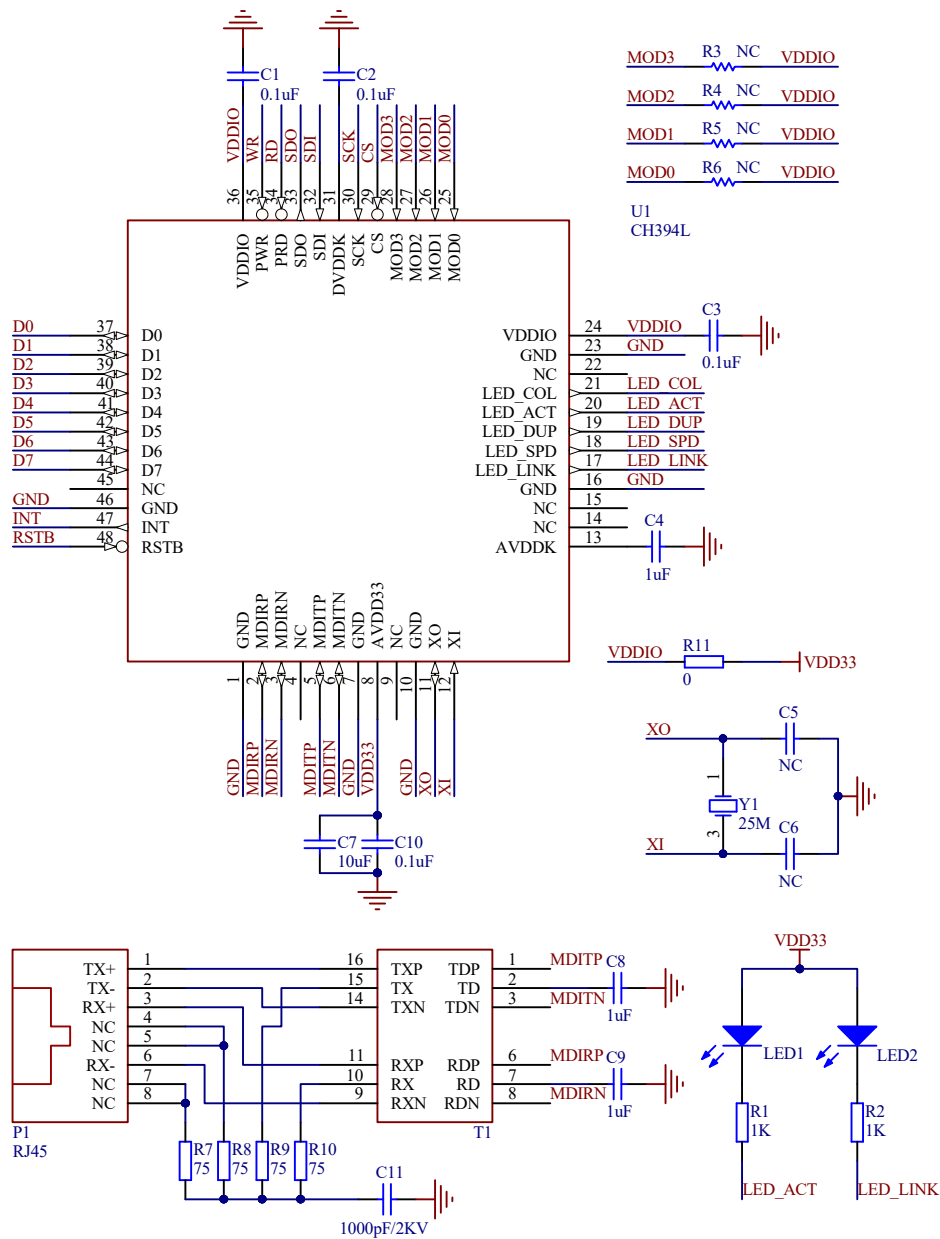
1、当使用 PING 功能进行诊断时，用户需要通过 PING ID 寄存器（PINGID）、PING 序列号寄存器（PINGSEQ）以及网络诊断目的 IP 寄存器（NDDIP）来配置目标 IP 地址以及 PING 数据包的 ID 和序列号。一旦 PING 请求成功，CH394L 将通过网络诊断中断寄存器（NDINT）触发 PING 相关的中断通知；如果 PING 请求未能得到响应，则会通过同一中断寄存器报告 TIMEOUT 中断。

2、在使用 ARP 功能时，用户仅需在网络诊断目的 IP 寄存器（NDDIP）中设定目标 IP 地址。ARP 请求成功后，CH394L 将同样经由网络诊断中断寄存器（NDINT）触发 ARP 中断，此时用户可从网络诊断目的 MAC 寄存器（NDDMAC）中读取到对端设备的 MAC 地址；若 ARP 请求失败，则同样通过 NDINT 寄存器产生 TIMEOUT 中断。

用户可以通过网络诊断超时重传次数寄存器（NDRC）和网络诊断超时重传时间寄存器（NDRT）单独设置网络诊断重传参数。

## 11.3 CH394L 参考电路

图 11-1 硬件参考电路



CH394L 已内置了晶体 Y1 的部分振荡电容，C5 和 C6 可以根据晶体参数调节。对于负载电容为 12pF 的 Y1，无需 C5 和 C6；对于负载电容为 20pF 的 Y1，C5 和 C6 建议各 15pF。

CH394L 已内置以太网 50Ω 阻抗匹配电阻，外部不要再接 49.9Ω 或 50Ω 电阻，等效于电压驱动。

T1 是 ethernet 网络变压器，其中心抽头分别通过电容 C8 和 C9 接地，不要接任何电源。