

# 以太网协议栈芯片 CH394

手册 1

版本: 1.1

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。

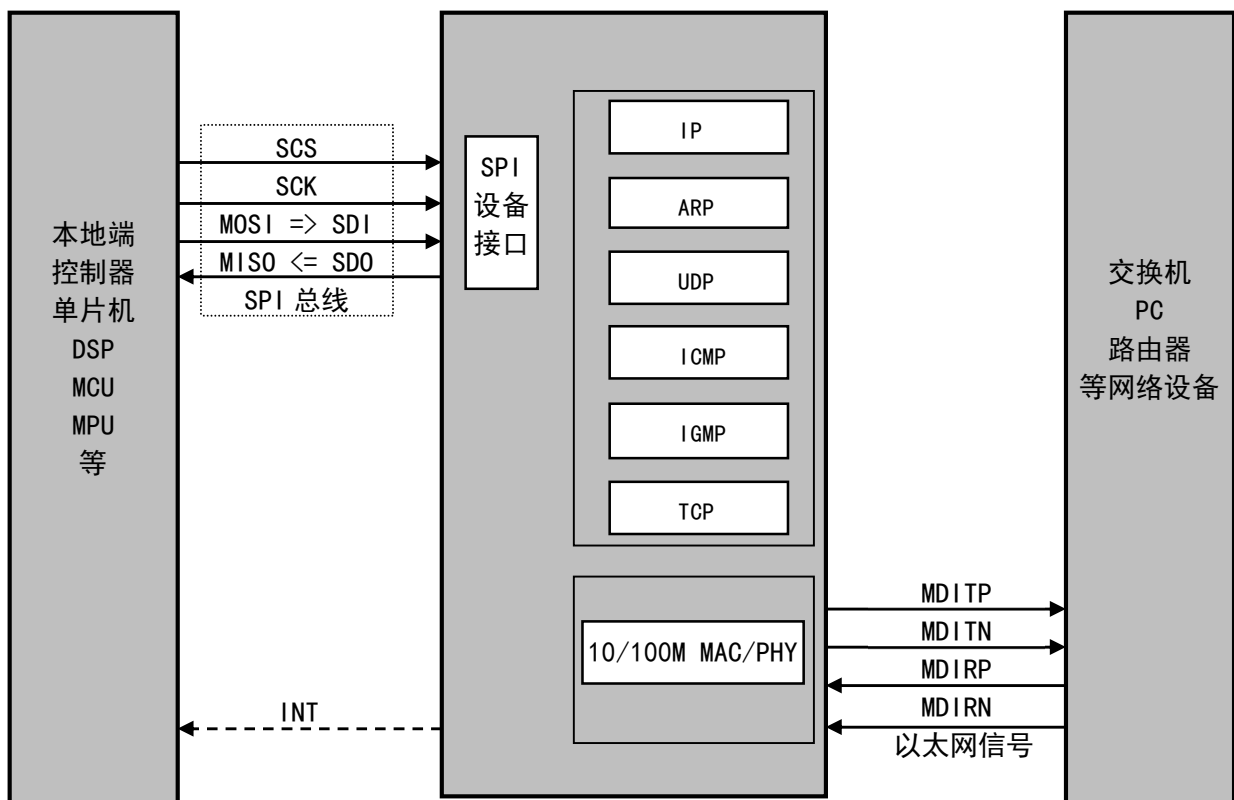
CH394 芯片自带 10/100M 以太网介质传输层 MAC 和物理层收发器 PHY，完全兼容 IEEE802.3 协议，内置了 IP、ARP、ICMP、IGMP、UDP、TCP 等以太网协议栈固件。单片机系统可以方便的通过 CH394 芯片进行网络通讯。CH394 支持网络唤醒模式（WOL）和掉电模式。

CH394Q 提供了 SPI 接口，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 接口通讯接口控制 CH394Q 芯片进行以太网通讯；或者通过 SPI 接口或 8 位并口控制 CH394L 芯片进行以太网通讯。

本手册的内容主要适用于 CH394Q，CH394L 的相关详细信息请参考《CH394DS2》手册。

下图为 CH394Q 的应用框图。

图 1-1 CH394Q 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
- CH394Q 支持 8 个 Socket，CH394L 支持 4 个 Socket，可以同时进行数据收发
- 支持 MACRAW 模式和 IPRAW 模式（IPRAW 模式仅 CH394L 支持）

- 提供最高 40MHz 速度的 SPI 设备接口（SPI 模式 0 或 3），高位在前
- CH394L 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 32K 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 LQFP48 无铅封装

### 3、引脚排列

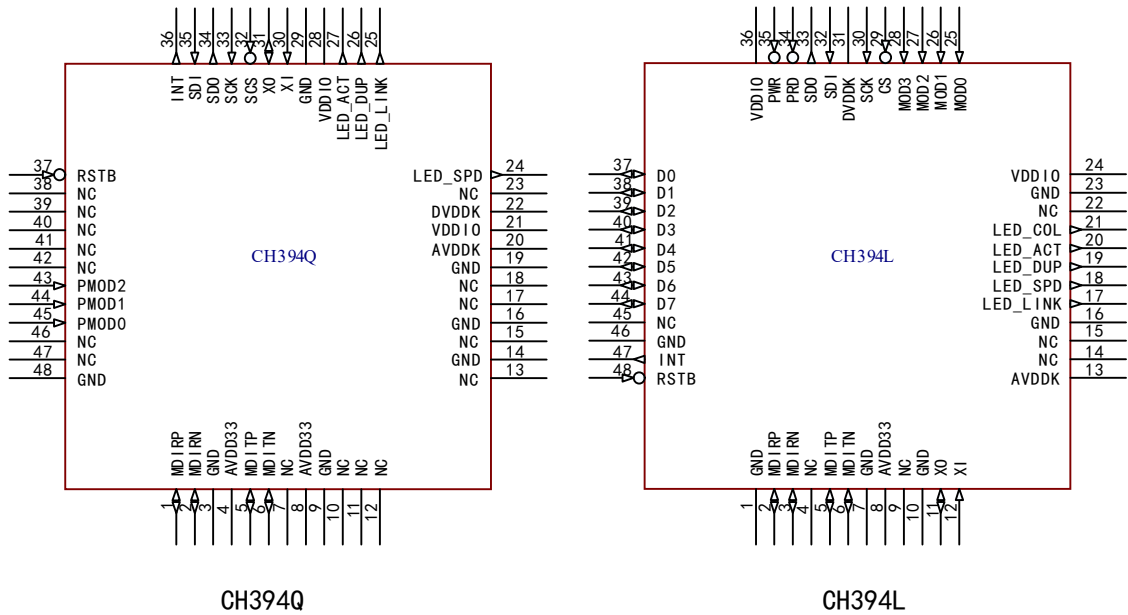


表 3-1 封装说明

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	CH394Q
LQFP48	7*7mm	0.5mm	19.7mil	标准 LQFP48 脚贴片	CH394L

注：CH394Q 引脚定义见本手册，CH394L 引脚定义见手册 2。

### 4、CH394Q 引脚定义

表 4-1 CH394Q 引脚定义

CH394Q 引脚号	引脚名称	类型	引脚说明
1	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。
2	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		
4	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。
8	AVDD33	P	3.3V电源输入，建议外接0.1uF或1uF对地电容。
20	AVDDK	P	外接 1uF 对地电容贴近芯片放置。
21、28	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。
22	DVDDK	P	外接 0.1uF 或 1uF 对地电容贴近芯片放置。
3、9、14、16、19、29、48	GND	P	公共接地端。
7、10、11、12、13、15、17、18、23、38、39、40、41、42、46、47	NC	-	保留引脚，建议悬空。
24	LED_SPD	0	网络速度指示 LED 输出： 低电平表示 100Mbps； 高电平表示 10Mbps。

25	LED_LINK	0	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。																			
26	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。																			
27	LED_ACT	0	载波感应指示 LED 输出： LED 闪烁表示有载波感应信号。																			
30	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。																			
31	XO	0	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。																			
32	SCS	I, PU	SPI 片选输入，低电平有效。																			
33	SCK	I	SPI 时钟输入，支持模式 0 或 3。																			
34	SDO	0	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。																			
35	SDI	I	SPI 串行数据输入，连接主机 SPI 接口的 MOSI。																			
36	INT	0	中断请求输出，低电平有效。																			
37	RSTB	I, PU	复位输入，低电平有效。																			
43	PMOD2	I, PU	PHY 工作模式选择： PMOD 引脚内置上拉电阻，此时 PHY 处于自协商模式。若无特殊需求，建议保持 PMOD 引脚悬空。																			
44	PMOD1	I, PU																				
45	PMOD0	I, PU																				
				<table border="1"> <thead> <tr> <th>PMOD[2:0]</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>10M 半双工，关闭自协商</td> </tr> <tr> <td>001</td> <td>10M 全双工，关闭自协商</td> </tr> <tr> <td>010</td> <td>100M 半双工，关闭自协商</td> </tr> <tr> <td>011</td> <td>100M 全双工，关闭自协商</td> </tr> <tr> <td>100</td> <td>100M 半双工，启动自协商</td> </tr> <tr> <td>101</td> <td>保留</td> </tr> <tr> <td>110</td> <td>保留</td> </tr> <tr> <td>111</td> <td>启动自协商（建议的默认模式）</td> </tr> </tbody> </table>	PMOD[2:0]	说明	000	10M 半双工，关闭自协商	001	10M 全双工，关闭自协商	010	100M 半双工，关闭自协商	011	100M 全双工，关闭自协商	100	100M 半双工，启动自协商	101	保留	110	保留	111	启动自协商（建议的默认模式）
PMOD[2:0]	说明																					
000	10M 半双工，关闭自协商																					
001	10M 全双工，关闭自协商																					
010	100M 半双工，关闭自协商																					
011	100M 全双工，关闭自协商																					
100	100M 半双工，启动自协商																					
101	保留																					
110	保留																					
111	启动自协商（建议的默认模式）																					

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

## 5、CH394Q 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。CH394Q 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。

### 5.1 工作模式

CH394Q 可以与其他 SPI 设备共用 SPI 接口，外设主机只需要与 CH394Q 通信时拉低 SCS 引脚与 CH394Q 通信，非通信时段接口可供其他设备使用。

### 5.2 数据帧

#### 5.2.1 数据帧格式

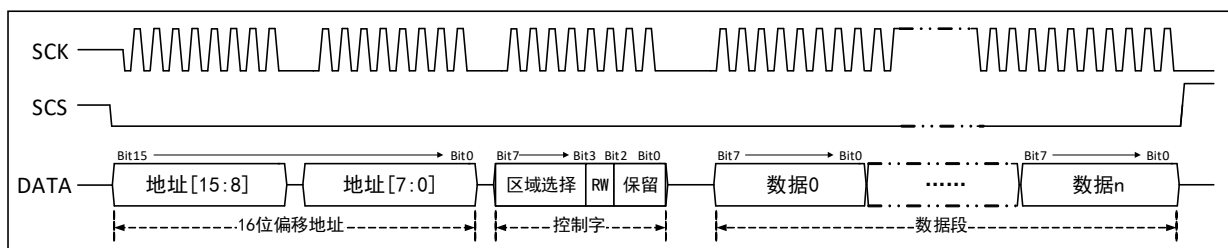
CH394Q 数据帧分包含三个部分：16 位偏移地址，8 位控制字，以及 N 字节数据段。

偏移地址为 CH394Q 寄存器地址或者 RX/TX 缓存的偏移地址。

控制字用于界定地址段中设定的偏移区域的所有权，并确定读/写模式。

SCS 拉低代表开始一包新的数据帧，SCS 拉高代表一包数据帧结束。

图 5-1 CH394Q 数据帧



CH394Q 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。

#### 5.2.2 控制字

CH394Q 控制字规定了：

- 1、地址段中设定的偏移区域的所有权；
- 2、SPI 的读/写模式；

表 5-1 控制字

位	名称	描述																																
[7:3]	BS	区域选择位： 用于界定地址段中设定的偏移区域的所有权。 CH394Q 内含 1 个通用寄存器，8 个 Socket 寄存器，以及对应的 Socket 读写缓存区。																																
		<table border="1"> <thead> <tr> <th colspan="2">BS [7:3]</th> <th rowspan="2">说明</th> </tr> <tr> <th>段号选择 [7:5]</th> <th>功能 [4:3]</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>00</td> <td>通用寄存器</td> </tr> <tr> <td>000</td> <td>01</td> <td>Socket 0 寄存器</td> </tr> <tr> <td>000</td> <td>10</td> <td>Socket 0 发送缓存</td> </tr> <tr> <td>000</td> <td>11</td> <td>Socket 0 接收缓存</td> </tr> <tr> <td>001</td> <td>00</td> <td>无意义</td> </tr> <tr> <td>001</td> <td>01</td> <td>Socket 1 寄存器</td> </tr> <tr> <td>001</td> <td>10</td> <td>Socket 1 发送缓存</td> </tr> <tr> <td>001</td> <td>11</td> <td>Socket 1 接收缓存</td> </tr> <tr> <td>010</td> <td>00</td> <td>无意义</td> </tr> </tbody> </table>	BS [7:3]		说明	段号选择 [7:5]	功能 [4:3]	000	00	通用寄存器	000	01	Socket 0 寄存器	000	10	Socket 0 发送缓存	000	11	Socket 0 接收缓存	001	00	无意义	001	01	Socket 1 寄存器	001	10	Socket 1 发送缓存	001	11	Socket 1 接收缓存	010	00	无意义
BS [7:3]		说明																																
段号选择 [7:5]	功能 [4:3]																																	
000	00	通用寄存器																																
000	01	Socket 0 寄存器																																
000	10	Socket 0 发送缓存																																
000	11	Socket 0 接收缓存																																
001	00	无意义																																
001	01	Socket 1 寄存器																																
001	10	Socket 1 发送缓存																																
001	11	Socket 1 接收缓存																																
010	00	无意义																																

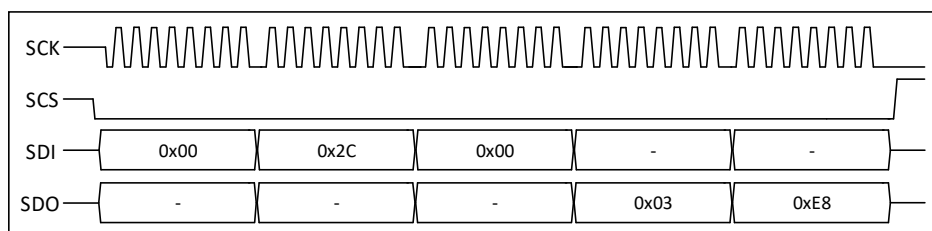
		010	01	Socket 2 寄存器
		010	10	Socket 2 发送缓存
		010	11	Socket 2 接收缓存
		011	00	无意义
		011	01	Socket 3 寄存器
		011	10	Socket 3 发送缓存
		011	11	Socket 3 接收缓存
		100	00	无意义
		100	01	Socket 4 寄存器
		100	10	Socket 4 发送缓存
		100	11	Socket 4 接收缓存
		101	00	无意义
		101	01	Socket 5 寄存器
		101	10	Socket 5 发送缓存
		101	11	Socket 5 接收缓存
		110	00	无意义
		110	01	Socket 6 寄存器
		110	10	Socket 6 发送缓存
		110	11	Socket 6 接收缓存
		111	00	无意义
		111	01	Socket 7 寄存器
		111	10	Socket 7 发送缓存
		111	11	Socket 7 接收缓存
2	RW	读/写模式选择位： 0：SPI 读； 1：SPI 写。		
[1:0]	Reserved	保留，默认为 0。		

### 5.3 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）：

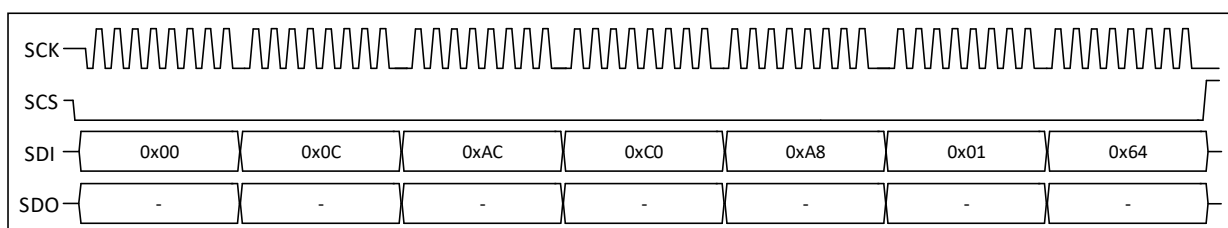
图 5-2 示例 1



#### (2) 写 4 字节数据

配置 Socket5 目的 IP 为 192.168.1.100:

图 5-3 示例 2



## 6、CH394Q 寄存器与缓存区

CH394Q 有 1 个通用寄存器区，8 个 Socket 寄存器区，8 个接收缓存区与 8 个发送缓存区。

每个寄存器区中有多个可配置的寄存器，同一个区中每个寄存器偏移地址不同，通过地址段中不同的偏移地址来确定所要读写的寄存器，通过控制字的 BS 位来界定地址段中设定的偏移地址具体属于那个寄存器区。

CH394Q 共有 16K 发送缓存区，8 个 Socket 每个默认 2K；16K 接收缓存区，8 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，控制字中的 BS 位控制具体所读/写的收/发缓冲区。

### 6.1 寄存器表

#### 6.1.1 通用寄存器表

CH394Q 通用寄存器配置了 CH394Q 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，且将控制字的 BS 部分设为 00000b。

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	0x00
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	IPO	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	SINT	RO	Socket 中断寄存器	0x00
0x0018	SINTE	RW	Socket 中断使能寄存器	0x00
0x0019	RTIME0	RW	重传时间寄存器	0x07
0x001A	RTIME1			0xD0
0x001B	RCNT	RW	重传次数寄存器	0x08
0x001C	-	-	保留	-
...	-	-		-
0x0027	-	-		-
0x0028	UNIP0	RO	不可达 IP 寄存器	0x00
...	...			...
0x002B	UNIP3			0x00
0x002C	UNPORT0	RO	不可达端口寄存器	0x00
0x002D	UNPORT1			0x00
0x002E	PHY_CFG	RW	PHY 配置寄存器	0xB8
0x002F	-	-	保留	-
...	-	-		-

0x0038				
0x0039	CHIPV	RO	芯片版本寄存器	0xXX

### 6.1.2 Socket 寄存器表

GH394Q 提供 8 个 Socket，每个 Socket 对应一个 Socket 寄存器区，通过控制字的 BS 位来选择对应 Socket  $n(0 \leq n \leq 7)$  的寄存器区(详参 5.2.2)，通过偏移地址决定所要读/写的 Socket  $n$  的寄存器。

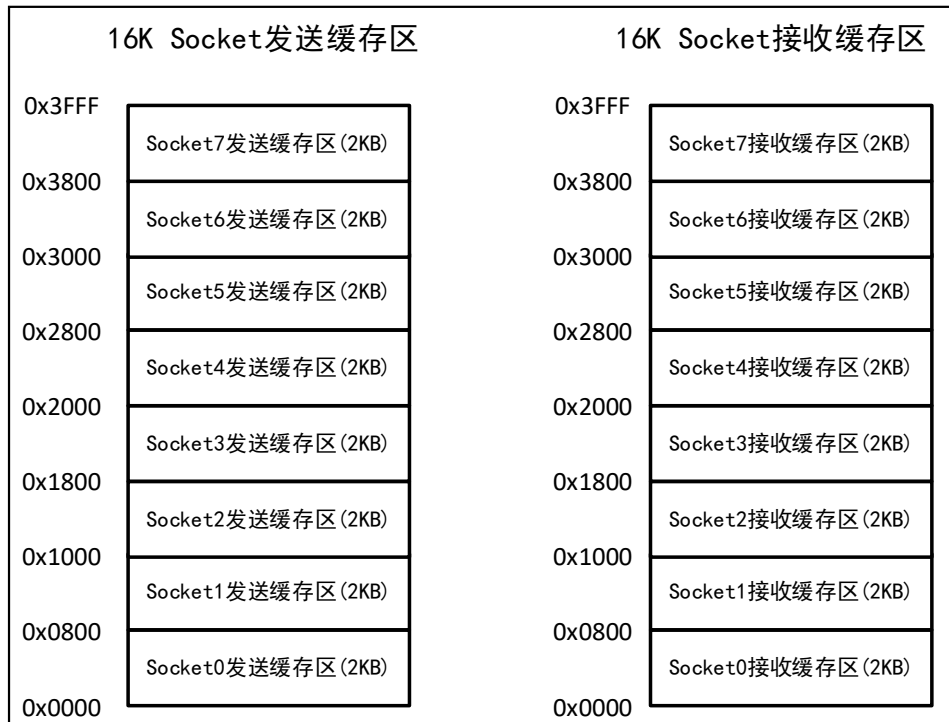
表 6-2 Socket 寄存器表

偏移地址	名称	访问	描述	复位值
0x0000	Sn_MODE	RW	Socket $n$ 模式寄存器	0x00
0x0001	Sn_CTRL	RW	Socket $n$ 控制寄存器	0x00
0x0002	Sn_INT	RW	Socket $n$ 中断寄存器	0x00
0x0003	Sn_STA	RO	Socket $n$ 状态寄存器	0x00
0x0004	Sn_PORT0	RW	Socket $n$ 源端口寄存器	0x00
0x0005	Sn_PORT1			0x00
0x0006	Sn_DMAC0	RW	Socket $n$ 目的 MAC 寄存器	0x00
...	...			...
0x000B	Sn_DMAC5			0x00
0x000C	Sn_DIP0	RW	Socket $n$ 目的 IP 寄存器	0x00
...	...			...
0x000F	Sn_DIP3			0x00
0x0010	Sn_DPORT0	RW	Socket $n$ 目的端口寄存器	0x00
0x0011	Sn_DPORT1			0x00
0x0012	Sn_MTU0	RW	Socket $n$ 最大传输单元寄存器	0x00
0x0013	Sn_MTU1			0x00
0x0014	-	-	保留	-
0x0015	Sn_TOS	RW	Socket $n$ IP 服务类型寄存器	0x00
0x0016	Sn_TTL	RW	Socket $n$ IP 生存时间寄存器	0x80
0x0017	...	-	保留	-
0x001D	-			
0x001E	Sn_RXBUF_SIZE	RW	Socket $n$ 接收缓存区大小寄存器	0x02
0x001F	Sn_TXBUF_SIZE	RW	Socket $n$ 发送缓存区大小寄存器	0x02
0x0020	Sn_TX_FS0	RO	Socket $n$ 空闲发送缓存长度寄存器	0x08
0x0021	Sn_TX_FS1			0x00
0x0022	Sn_TX_RD0	RO	Socket $n$ 发送缓存区读指针寄存器	0xXX
0x0023	Sn_TX_RD1			
0x0024	Sn_TX_WR0	RW	Socket $n$ 发送缓存区写指针寄存器	0xXX
0x0025	Sn_TX_WR1			
0x0026	Sn_RX_RS0	RO	Socket $n$ 接收数据长度寄存器	0x00
0x0027	Sn_RX_RS1			0x00
0x0028	Sn_RX_RD0	RW	Socket $n$ 接收缓存区读指针寄存器	0xXX
0x0029	Sn_RX_RD1			
0x002A	Sn_RX_WR0	RO	Socket $n$ 接收缓存区写指针寄存器	0xXX
0x002B	Sn_RX_WR1			
0x002C	Sn_INTE	RW	Socket $n$ 中断使能寄存器	0xFF
0x002D	Sn_IPF0	RW	Socket $n$ IP 分片寄存器	0x40
0x002E	Sn_IPF1			0x00
0x002F	Sn_KEEPA_LIVE	RW	Socket $n$ KeepAlive 时间寄存器	0x00



## 6.2 接收/发送缓存区

图 6-1 接收发送缓冲区



CH394Q 为每个 Socket  $n$  ( $n = 0 \sim 7$ ) 提供了独立的发送和接收内存缓冲区，共有 16K 接收缓存区与 16K 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（接收缓存区共 16KB，发送缓存区共 16KB）。用户可根据需求，通过配置 Socket 发送缓存大小寄存器（ $S_n\_TXBUF\_SIZE$ ）和 Socket 接收缓存大小寄存器（ $S_n\_RXBUF\_SIZE$ ），重新分配这 16KB 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 16KB，以防止数据传输错误。

## 7、CH394Q 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。CH394Q 内部可以同时提供 8 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

CH394Q 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	0
6	Reserved	保留。	RO	0
5	WOL	网络功能唤醒使能: 0: 关闭网络唤醒; 1: 开启网络唤醒。 CH394Q 集成了魔法包唤醒功能, 使能魔法包唤醒后, CH394Q 收到 UDP 发来的匹配自身 MAC 地址的魔法包 (0xFFFFFFFF+16 个 CH394Q 的 MAC 地址) 后, 会产生 WOL 中断。	RW	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	Reserved	保留。	RO	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	RW	0

		发送 ARP 请求。		
0	Reserved	保留。	R0	0

### 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。

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，低字节地址储存网关地址的高字节。

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 CH394Q 的 MAC 地址，低字节地址储存 MAC 地址的高字节。

CH394Q 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 CH394Q 的 IP 地址，低字节地址储存 IP 地址的高字节。

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，CH394Q 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，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 SINT 为 0，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 CH394Q 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 CH394Q 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。	RW1Z	0

		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。		
5	Reserved	保留	RO	0
4	MP	魔法包唤醒中断： 当启用魔法包功能且通过 UDP 收到 Magic Packet 唤醒包时会产生此中断。	RW1Z	0
[3:0]	Reserved	保留	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，CH394Q 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	Reserved	保留。	RO	0
4	MP	魔法包唤醒中断使能位： 使能该位即允许产生魔法包唤醒中断。	RW	0
[3:0]	Reserved	保留。	RO	0

#### 7.1.9 Socket 中断寄存器 (SINT) [0x0017]

该寄存器用于获取 Socket 中断状态，当 Socket n 出现中断事件时，Sn\_INT 寄存器对应位置 1 且 SINT 的第 n 位置 1，SINTE 对应位为 1 时 INT 脚也将被拉低。当 Sn\_INT 为 0 时，SINT 对应位也将清零。当在 SINT 和 GINT 均为 '0x00' 时，INT 脚拉高。

表 7-4 Socket 中断寄存器

位	名称	描述	访问	默认值
7	S7_INT	Socket7 中断。	RO	0
6	S6_INT	Socket6 中断。	RO	0
5	S5_INT	Socket5 中断。	RO	0
4	S4_INT	Socket4 中断。	RO	0
3	S3_INT	Socket3 中断。	RO	0
2	S2_INT	Socket2 中断。	RO	0
1	S1_INT	Socket1 中断。	RO	0
0	S0_INT	Socket0 中断。	RO	0

#### 7.1.10 Socket 中断使能寄存器 (SINTE) [0x0018]

该寄存器用于控制哪些 Socket 能够触发中断，每个使能位对应 Socket 中断寄存器 (SINT) 一个位。

当 Socket 中断事件产生后，只有 SINTE 中对应位为 1 时，CH394Q 才会拉低 INT 脚，产生中断。

表 7-5 Socket 中断使能寄存器

位	名称	描述	访问	默认值
7	S7_INT	Socket7 中断使能位： 使能该位即允许产生 Socket7 中断。	RW	0
6	S6_INT	Socket6 中断使能位： 使能该位即允许产生 Socket6 中断。	RW	0
5	S5_INT	Socket5 中断使能位：	RW	0

		使能该位即允许产生 Socket5 中断。		
4	S4_INT	Socket4 中断使能位： 使能该位即允许产生 Socket4 中断。	RW	0
3	S3_INT	Socket3 中断使能位： 使能该位即允许产生 Socket3 中断。	RW	0
2	S2_INT	Socket2 中断使能位： 使能该位即允许产生 Socket2 中断。	RW	0
1	S1_INT	Socket1 中断使能位： 使能该位即允许产生 Socket1 中断。	RW	0
0	S0_INT	Socket0 中断使能位： 使能该位即允许产生 Socket0 中断。	RW	0

#### 7.1.11 重传时间寄存器 (RTIME) [0x0019-0x001A]

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。

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，CH394Q 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	0x0019	0x001A
数据	0x13	0x88

#### 7.1.12 重传次数寄存器 (RCN) [0x001B]

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 8。当重传超过设定次数时，对应的 Socket 的超时中断位将被置 1。

#### 7.1.13 不可达 IP 寄存器 (UNIP) [0x0028-0x002B]

该寄存器用于获取不可达 IP 地址。当 CH394Q 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，CH394Q 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	0x0028	0x0029	0x002A	0x002B
数据	0xC0	0xA8	0x01	0xC8

#### 7.1.14 不可达端口寄存器 (UNPORT) [0x002C-0x002D]

该寄存器用于获取不可达端口。当 CH394Q 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，CH394Q 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	0x002C	0x002D
数据	0x07	0xD0

#### 7.1.15 PHY 配置寄存器 (PHY\_CFG) [0x002E]

该寄存器配置 PHY 相关的状态，该寄存器也指示了当前 PHY 的连接状态。

表 7-6 PHY 配置寄存器

位	名称	描述	访问	默认值
7	RST	复位： 0: PHY 复位； 1: 正常工作。 注：PHY 复位完成后该位需手动置 1。	RW	1
6	OPWD	工作模式选择位：	RW	0

		<p>0: 通过硬件引脚 (PMOD[2: 0]) 配置 PHY 模式; 1: 通过 PHY 配置寄存器[5:3]位配置 PHY 模式。 CH394Q 上电时或通过 RSTB 脚复位时会检测 PMOD[2: 0]引脚的电平状态来设定 PHY 的工作模式。 当用户想要通过 OPMDc 来控制 PHY 模式时, 需要:</p> <ol style="list-style-type: none"> <li>1、将 OPWD 位置 1;</li> <li>2、将 OPMDc 设置为所需的模式值。</li> </ol> <p>当用户再次想采用 PMOD[2: 0]引脚的电平状态来设定 PHY 的工作模式时, 只需要将 OPWD 位由 1 置位 0 即可, 无需再次上电或复位。</p>																				
[5:3]	OPMDc	<p>PHY 模式配置位:</p> <table border="1"> <thead> <tr> <th>[5:3]</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>10M 半双工, 关闭自协商</td> </tr> <tr> <td>001</td> <td>10M 全双工, 关闭自协商</td> </tr> <tr> <td>010</td> <td>100M 半双工, 关闭自协商</td> </tr> <tr> <td>011</td> <td>100M 全双工, 关闭自协商</td> </tr> <tr> <td>100</td> <td>100M 半双工, 启动自协商</td> </tr> <tr> <td>101</td> <td>保留</td> </tr> <tr> <td>110</td> <td>掉电模式</td> </tr> <tr> <td>111</td> <td>启动自协商</td> </tr> </tbody> </table> <p>默认为自协商模式。</p>	[5:3]	说明	000	10M 半双工, 关闭自协商	001	10M 全双工, 关闭自协商	010	100M 半双工, 关闭自协商	011	100M 全双工, 关闭自协商	100	100M 半双工, 启动自协商	101	保留	110	掉电模式	111	启动自协商	RW	111b
[5:3]	说明																					
000	10M 半双工, 关闭自协商																					
001	10M 全双工, 关闭自协商																					
010	100M 半双工, 关闭自协商																					
011	100M 全双工, 关闭自协商																					
100	100M 半双工, 启动自协商																					
101	保留																					
110	掉电模式																					
111	启动自协商																					
2	DUPS	<p>双工模式指示位:</p> <p>0: 半双工; 1: 全双工。</p>	RO	0																		
1	SPDS	<p>速度指示位:</p> <p>0: 10Mb/s; 1: 100Mb/s。</p>	RO	0																		
0	LINKS	<p>连接状态指示</p> <p>0: 物理层未建立链接; 1: 物理层建立链接。</p>	RO	0																		

### 7.1.16 芯片版本寄存器 (CHIPV) [0x0039]

该寄存器用于获取芯片版本。

## 7.2 Socket 寄存器说明

### 7.2.1 Socket n 模式寄存器 (Sn\_MODE) [0x0000]

该寄存器用于配置 Socket n 的工作模式。

表 7-7 Socket n 模式寄存器

位	名称	描述	访问	默认值
7	MUL_MFEN	<p>当 Socket n 工作在 UDP 模式时:</p> <p>0: 关闭组播模式; 1: 开启组播模式。 使用组播模式时, 需要在 Socket 打开前, 通过 Sn_DIP、Sn_DMACH 提前配置好组播 IP、组播 MAC。</p> <p>当 Socket n 工作在 MACRAW 模式时:</p> <p>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,</p>	RW	0

		Socket n 只会接收网络上目的 MAC 为 CH394Q 自身 MAC 地址以及广播地址的包。												
6	BCASTB	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 UDP 和 MACRAW 模式下，Socket n 将不接收网络上的广播包。	RW	0										
5	NA_MV_MM	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。 当 Socket n 工作在 MACRAW 模式时： 0: 不屏蔽组播包； 1: 屏蔽组播包。 该位置 1 后，在 MACRAW 模式下，Socket n 将不接收网络上的组播包。	RW	0										
4	UCASTB_MI P6B	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。 当 Socket n 工作在 MACRAW 模式时： 0: 不屏蔽 IPv6 包； 1: 屏蔽 IPv6 包。 该位置 1 后，在 MACRAW 模式下，Socket n 将不接收网络上的 IPv6 包。	RW	0										
[3:0]	PMD	设置 Socket 工作模式： <table border="1" style="margin-left: auto; margin-right: auto;"> <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>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	0100	MACRAW	RW	0
[3:0]	说明													
0000	关闭													
0001	TCP													
0010	UDP													
0100	MACRAW													

### 7.2.2 Socket n 控制寄存器 (Sn\_CTRL) [0x0001]

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。

表 7-8 Socket n 控制寄存器

代码	名称	描述						
01H	OPEN	打开 Socket n： 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： <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Sn_MODE[3:0]</th> <th>Sn_STA</th> </tr> </thead> <tbody> <tr> <td>0000 (关闭)</td> <td>-</td> </tr> <tr> <td>0001 (TCP)</td> <td>0x13 (INIT)</td> </tr> </tbody> </table>	Sn_MODE[3:0]	Sn_STA	0000 (关闭)	-	0001 (TCP)	0x13 (INIT)
Sn_MODE[3:0]	Sn_STA							
0000 (关闭)	-							
0001 (TCP)	0x13 (INIT)							



			0010 (UDP)	0x22 (UDP)	
			0100 (MACRAW)	0x42 (MACRAW)	
02H	LISTEN	设置 Socket n 为监听态： 该命令只有在 Socket n 为 TCP 模式，且为 INIT 态时生效。 当 Socket n 设为 LISTEN 态时，Socket n 为 TCP Sever，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。 如果连接失败 Sn_STA 寄存器值变为 0x00 (CLOSE)，Sn_INT[3] 置 1。			
04H	CONNECT	使能 Socket n 进入连接模式： 该命令只有在 Socket n 为 TCP 模式，且为 INIT 态时生效。 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。 当 Socket 连接失败时： 1: ARP 失败，无法获得目的 MAC 地址 (Sn_INT[3] 置 1)； 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)； 3: 收到 RST 包，被对方主动断开连接。 出现上述情况时 Sn_STA 寄存器值变为 0x00 (CLOSE)。			
08H	DISCONNECT	Socket n 断开连接： 该命令只有在 Socket n 为 TCP 模式时生效。 主动关闭：主动向对端设备发送 FIN 包； 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。 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)。			
10H	CLOSE	关闭 Socket n： 如果 Socket n 为 UDP 或 MACRAW 模式此命令会直接关闭 Socket。 如果 Socket n 为 TCP 模式，此命令会发送 RST 包关闭 Socket，不同于标准的 TCP 关闭流程，该命令关闭时并不会发 FIN 包。			
20H	SEND	Socket n 发送数据： 发送 Socket n 发送缓存区内的数据，具体操作详参 11.2.3 节数据发送流程。			
21H	SNED_MAC	Socket n 无 ARP 发送数据： 该命令只有在 Socket n 为 UDP 模式时生效。 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_DMAL 所设 MAC 地址发送。			
22H	SEND_KEEP	Socket n 发送 KeepAlive 包： 该命令只有在 Socket n 为 TCP 模式时生效。 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)。			
40H	RECV	Socket n 接收数据： 完成 Socket n 接收数据的流程，详细步骤请参考 11.2.2 章节数据接收流程。			

### 7.2.3 Socket n 中断寄存器 (Sn\_INT) [0x0002]

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 7-9 Socket n 中断寄存器

位	名称	描述	访问	默认值
[7:5]	Reserved	保留。	R0	0
4	INT_SEND_SUC	发送完成中断：	RW1Z	0



		当 Socket n SEND 命令完成后产生此中断。		
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) [0x0003]

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 7-10 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 服务器模式，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)。 Sn_STA 为 0x17 (ESTABLISHED) 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。
42H	MAC_RAW	MACRAW 模式： 表示 Socket n 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 7-11 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)。
1BH	TIME_WAIT	
1DH	LAST_ACK	

#### 7.2.5 Socket n 源端口寄存器 (Sn\_PORT) [0x0004-0x0005]

该寄存器用于设置 Socket n 的源端口号。

例如: 源端口号为 2000。

地址	0x0004	0x0005
数据	0x07	0xD0

#### 7.2.6 Socket n 目的 MAC 寄存器 (Sn\_DMAC) [0x0006-0x000B]

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

地址	0x0006	0x0007	0x0008	0x0009	0x000A	0x000B
数据	0x38	0x3B	0x26	0x44	0x55	0x66

#### 7.2.7 Socket n 目的 IP 寄存器 (Sn\_DIP) [0x000C-0x000F]

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、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: 目的 IP 地址为 192.168.1.200。

地址	0x000C	0x000D	0x000E	0x000F
数据	0xC0	0xA8	0x01	0xC8

#### 7.2.8 Socket n 目的端口寄存器 (Sn\_DPORT) [0x0010-0x0011]

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 模式下, 发送前需将目标端口号写入该寄存器。

例如: 目标端口号为 3000。

地址	0x0010	0x0011
数据	0x0B	0xB8

#### 7.2.9 Socket n 最大传输单元寄存器 (Sn\_MTU) [0x0012-0x0013]

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, CH394Q 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。

例如：设 MTU 为 1000。

地址	0x0012	0x0013
数据	0x03	0xE8

#### 7.2.10 Socket n IP 服务类型寄存器 (Sn\_TOS) [0x0015]

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。

#### 7.2.11 Socket n IP 生存时间寄存器 (Sn\_TTL) [0x0016]

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。

#### 7.2.12 Socket n 接收缓存区大小寄存器 (Sn\_RXBUF\_SIZE) [0x001E]

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 16，对应 0 到 16KB。CH394Q 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 7 接收缓存区	
第 0 块	第 1 块	第 2 块	第 3 块	.....	第 14 块	第 15 块
2K		2K			2K	

CH394Q 内部接收缓冲区默认分配如上图，一共由 16 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 接收缓冲区的大小。Sn\_RXBUF\_SIZE 配置完成后，就会按照 Socket 0 到 7 的顺序将接收缓存重新分配。设置接收缓冲区时需注意所有 Socket 接受缓存区总和不能超过 16KB。

#### 7.2.13 Socket n 发送缓存区大小寄存器 (Sn\_TXBUF\_SIZE) [0x001F]

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 16，对应 0 到 16KB。CH394Q 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 7 发送缓存区	
第 0 块	第 1 块	第 2 块	第 3 块	.....	第 14 块	第 15 块
2K		2K			2K	

CH394Q 内部发送缓冲区默认分配如上图，一共由 16 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 发送缓冲区的大小。Sn\_TXBUF\_SIZE 配置完成后，就会按照 Socket 0 到 7 的顺序将发送缓存重新分配。设置发送缓冲区时需注意所有 Socket 发送缓存区总和不能超过 16KB。

#### 7.2.14 Socket n 空闲发送缓存长度寄存器 (Sn\_TX\_FS) [0x0020-0x0021]

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 11.2.3 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	0x0020	0x0021
数据	0x03	0xE8

#### 7.2.15 Socket n 发送缓存区读指针寄存器 (Sn\_TX\_RD) [0x0022-0x0023]

该寄存器用于获取 Socket n 的发送读指针，指示了 CH394Q 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，CH394Q 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冲区，并更新 Sn\_TX\_WR 指针后，通过发出 SEND 指令，CH394Q 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 11.2.3 节发送数据流程。

#### 7.2.16 Socket n 发送缓存区写指针寄存器 (Sn\_TX\_WR) [0x0024-0x0025]

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 11.2.3 节发送数据流程。

例如：Socket n 发送缓存区写指针为 2048。

地址	0x0024	0x0025
数据	0x08	0x00

#### 7.2.17 Socket n 接收数据长度寄存器 (Sn\_RX\_RS) [0x0026-0x0027]

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 11.2.2 节接收数据流程。

例如：Socket n 接收数据长度为 1024。

地址	0x0026	0x0027
数据	0x04	0x00

#### 7.2.18 Socket n 接收缓存区读指针寄存器 (Sn\_RX\_RD) [0x0028-0x0029]

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 11.2.2 节接收数据流程。

例如：Socket n 接收缓存区读指针为 2048。

地址	0x0028	0x0029
数据	0x08	0x00

#### 7.2.19 Socket n 接收缓存区写指针寄存器 (Sn\_RX\_WR) [0x002A-0x002B]

该寄存器用于获取 Socket n 的接收缓存区写指针，指示了 CH394Q 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，CH394Q 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 CH394Q 从以太网接收到符合要求的数据并载入接收缓冲区后，会相应更新 Sn\_RX\_WR 指针。具体操作详参 11.2.2 节接收数据流程。

#### 7.2.20 Socket n 中断使能寄存器 (Sn\_INTE) [0x002C]

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) CH394Q 才会拉低 INT 脚，产生中断。

表 7-12 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	-
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

## 7.2.21 Socket n IP 分片寄存器 (Sn\_IPF) [0x002D-0x002E]

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	0x002D	0x002E
数据	0x40	0x00

## 7.2.22 Socket n KeepAlive 时间寄存器 (Sn\_KEEPA L I V E) [0x002F]

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) 传输完成一包后，进入空闲（无数据流通）状态时，系统将按照 Sn\_KEEPA L I V E 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 L I V E 为 0 时，用户可以通过 SEND\_KEE P 命令手动发送 KeepAlive 包。当 Sn\_KEEPA L I V E 不为 0 时，SEND\_KEE P 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	0x002F
数据	0x04

## 8、CH394Q 功能说明

### 8.1 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，CH394Q 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。

CH394Q 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。CH394Q 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。

CH394Q 的 SPI 接口支持 SPI 模式 0 和 SPI 模式 3，CH394Q 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章节 5。

### 8.2 其它硬件

CH394Q 芯片内部集成了 10M/100M 以太网 PHY、MAC、SPI-Slave 控制器、SRAM、高速 MCU 和 PLL 倍频器、电源上电复位电路等。

CH394Q 芯片的 MDIRP、MDIRN、MDITP、MDITN 为以太网的信号线。CH394Q 的 PHY 支持 MDI/MDIX 线路自动转换。

CH394Q 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，CH394Q 芯片被复位；当 RSTB 引脚恢复为高电平后，CH394Q 将进入初始化阶段约 15ms，在这段时间内主机禁止操作 CH394Q。



## 9、CH394Q 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	°C
$T_J$	结温度范围	-40	100	°C
$T_S$	存储时的环境温度	-55	150	°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 ( $AV_{DD33} = 3.3V$ ,  $V_{DD10} = 3.3V$ ,  $T_A = 25^\circ C$ )

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 ( $AV_{DD33} = 3.3V$ ,  $V_{DD10} = 3.3V$ ,  $T_A = 25^\circ C$ )

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

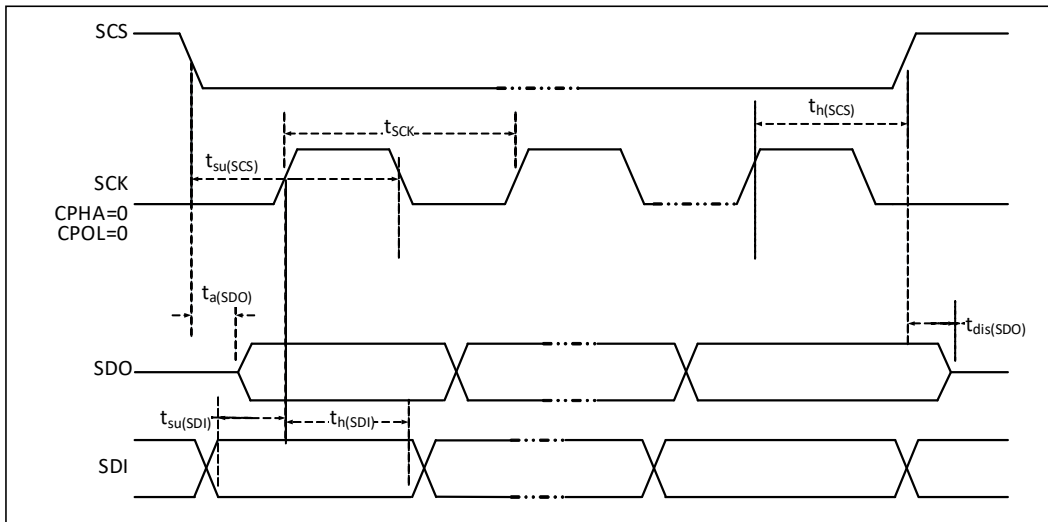


图 9-2 SPI 模式 3 时序图

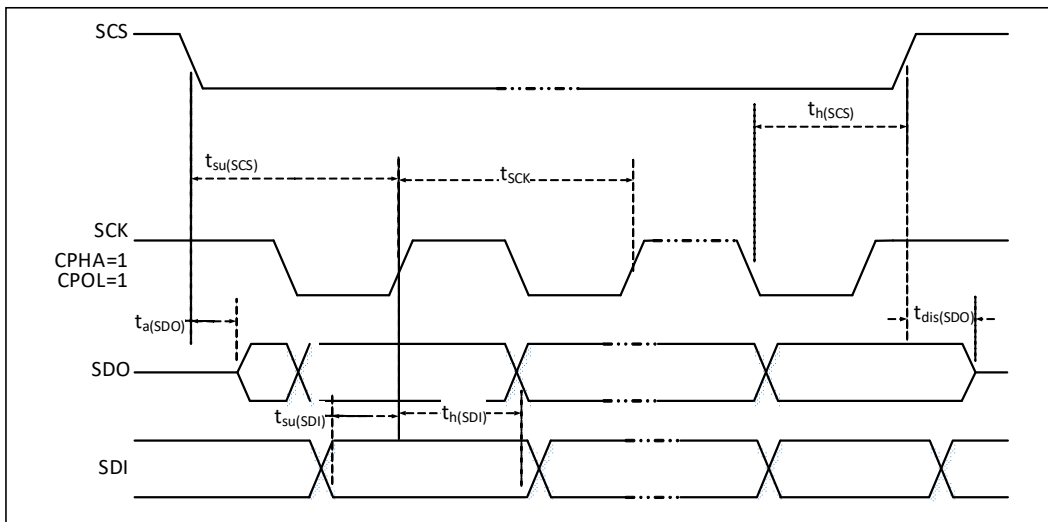


图 9-3 SPI 串行接口图

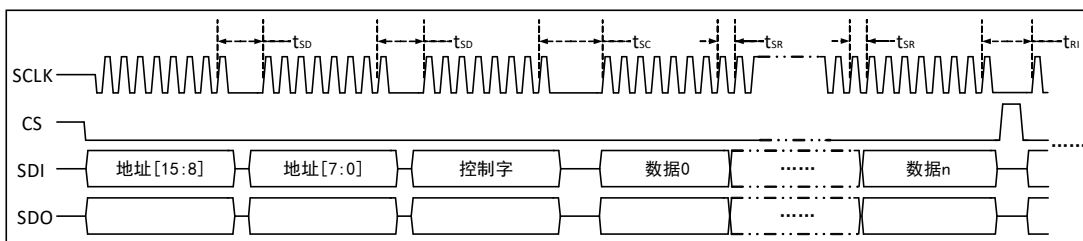




表 9-4 SPI 参数表 ( $V_{DD33} = 3.3V$ ,  $V_{DD10} = 3.3V$ ,  $T_A = 25^\circ C$ )

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
$t_{SC}$	控制字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		
$t_{SD}$	偏移地址和控制字每个字节间隔		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 振荡器及晶振时序

表 9-5 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。

#### 9.4.3 复位时序

表 9-6 复位时序参数表

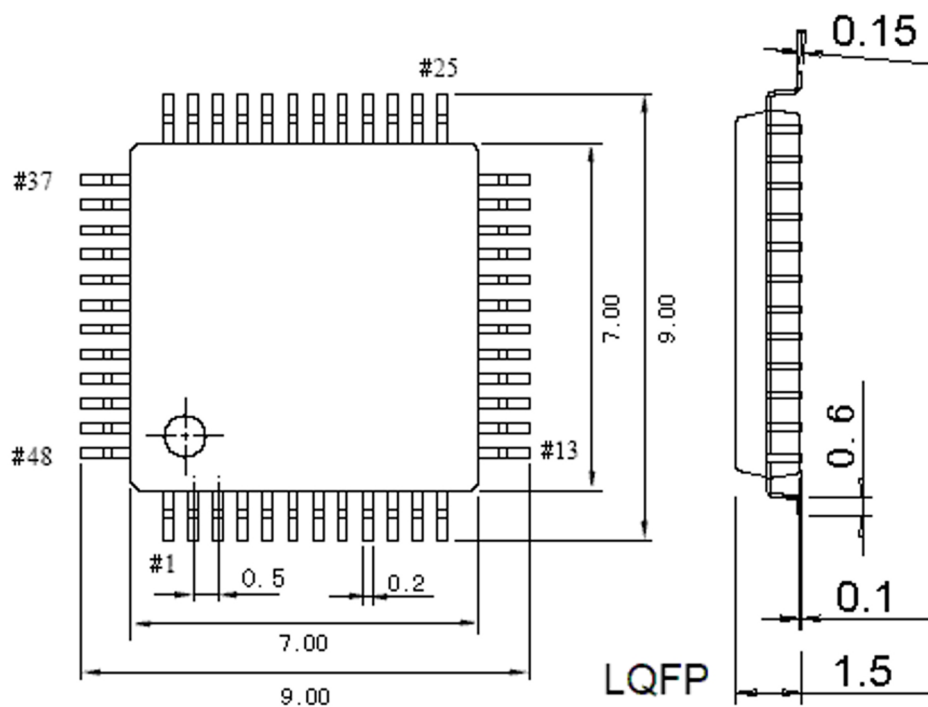
符号	参数	最小值	典型值	最大值	单位
$t_{RSTTEMP0}$	RSTB 低电平宽度	1			us
$t_{RSTTEMP1}$	RSTB 高电平至主机可操作	12	15	19	ms
$t_{RSTTEMP2}$	复位命令生效至主机可操作	12	15	19	ms
$t_{RSTTEMP3}$	电源上电复位至主机可操作	27	30	35	ms

## 10、CH394Q 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、CH394Q 应用

### 11.1 应用基础

CH394Q 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 CH394Q 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 CH394Q，必要操作

表 11-1 初始化 CH394Q 步骤

步骤	动作	操作寄存器	地址	操作
1	设置 CH394Q 的 MAC 地址	MAC	0x0009	写入 MAC 地址
2	设置 CH394Q 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，CH394Q 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 接收数据流程

在进行数据接收操作时，必须：

表 11-2 接收数据示例

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	0x0026	读出接收长度
2	获取数据在缓存区内的起始地址	Sn_RX_RD	0x0028	读出原地址 A <sub>0</sub>
3	从 A <sub>0</sub> 取特定长度的数据 L <sub>0</sub>	-	A <sub>0</sub>	读数据
4	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	0x0028	写入 A <sub>0</sub> +L <sub>0</sub>
5	向 Sn_CTRL 寄存器写入 RECV 命令	Sn_CTRL	0x0001	写入 RECV 命令
6	等待命令完成	Sn_CTRL	0x0001	读出为 0x00 代表执行成功

#### 11.2.3 发送数据流程

在进行数据发送操作时，必须：

表 11-3 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	0x0020	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	0x0024	读出原地址 A <sub>0</sub>
3	向 A <sub>0</sub> 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>	写数据
4	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	0x0024	写入 A <sub>0</sub> +L <sub>1</sub>
5	向 Sn_CTRL 寄存器写入 SEND 命令	Sn_CTRL	0x0001	写入 SEND 命令
6	等待命令完成	Sn_CTRL	0x0001	读出为 0x00 代表执行成功

#### 11.2.4 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 CH394Q 写入若干字节数据流后，CH394Q 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，CH394Q 会自动将数据分包成合适的大小发送。

当 CH394Q 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。由于 UDP 模式下 CH394Q 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。

CH394Q 支持 UDP 单播、UDP 广播和 UDP 组播。

CH394Q 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

CH394Q 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 11-4 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	0x0000	写入 0x02
2	设置 Socket n 源端口号	Sn_PORT	0x0004	写入源端口号
3	打开 Socket n	Sn_CTRL	0x0001	写入 0x01 命令
4	等待命令完成	Sn_CTRL	0x0001	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 MUL\_MFEN 位置为 1 以启用组播功能。在 Socket 建立连接期间，CH394Q 将依据 Sn\_MODE 寄存器中的 NA\_MV\_MM 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。

#### 11.2.5 初始化 Socket 为 TCP Sever 模式

初始化步骤如下：

表 11-5 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	0x0000	写入 0x01
2	设置 Socket n 源端口号	Sn_PORT	0x0004	写入源端口号

3	打开 Socket n	Sn_CTRL	0x0001	写入 0x01 命令
4	等待命令完成	Sn_CTRL	0x0001	读出为 0x00 代表执行成功
5	设 Socket n 为监听模式	Sn_CTRL	0x0001	写入 0x02 命令
6	等待命令完成	Sn_CTRL	0x0001	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	0x0003	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, CH394Q 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 CH394Q 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。

#### 11.2.6 初始化 Socket 为 TCP Client 模式

初始化步骤如下:

表 11-6 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	0x0000	写入 0x01
2	设置 Socket n 源端口号	Sn_PORT	0x0004	写入源端口号
3	打开 Socket n	Sn_CTRL	0x0001	写入 0x01 命令
4	等待命令完成	Sn_CTRL	0x0001	读出为 0x00 代表执行成功
5	设置目的 IP	Sn_DIP	0x000C	写入目的 IP
6	设置目的端口	Sn_DPORT	0x0010	写入目的端口
7	启动 Socket 连接	Sn_CTRL	0x0001	写入 0x04 命令
8	等待命令完成	Sn_CTRL	0x0001	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	0x0003	值变为 0x17

在发送完连接命令后, CH394Q 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, CH394Q 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 CH394Q 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 MACRAW 模式

初始化步骤如下:

表 11-7 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	0x0000	写入 0x04

2	打开 Socket0	Sn_CTRL	0x0001	写入 0x01 命令
3	等待命令完成	Sn_CTRL	0x0001	读出为 0x00 代表执行成功

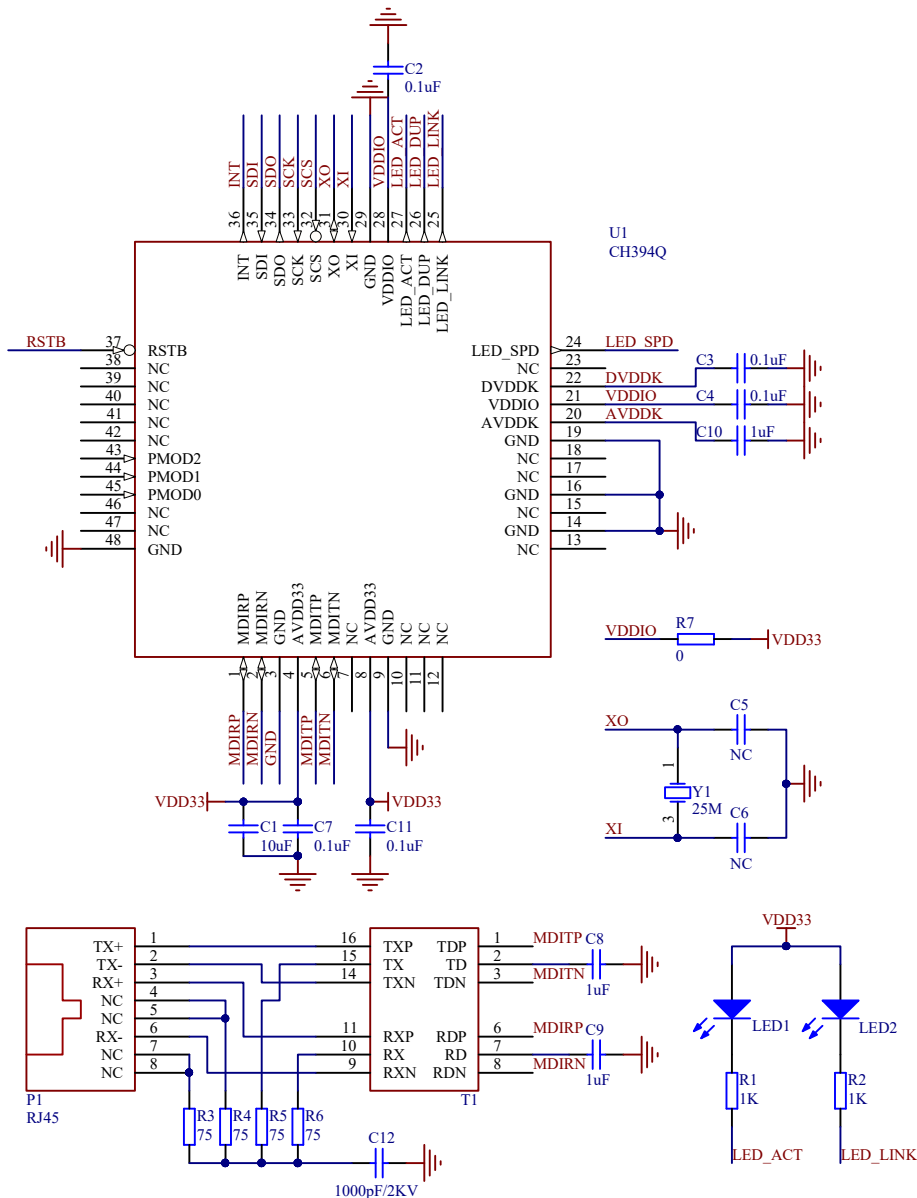
只有 Socket0 支持 MACRAW 模式，在 MACRAW 模式下，CH394Q 会透明传输以太网和单片机之间的数据，不会对数据进行 TCP/IP 封装。MACRAW 模式下，Socket 0 不会接收其他 Socket 接受的包。

CH394Q 会在接收到数据的头部添加 2 个字节的信息表。

数据包长度	数据
2 Byte	N Byte

### 11.3 CH394Q 参考电路

图 11-1 硬件参考电路



CH394Q 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。

CH394Q 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。