

# 串口和 CAN 接口拓展芯片 CH9434

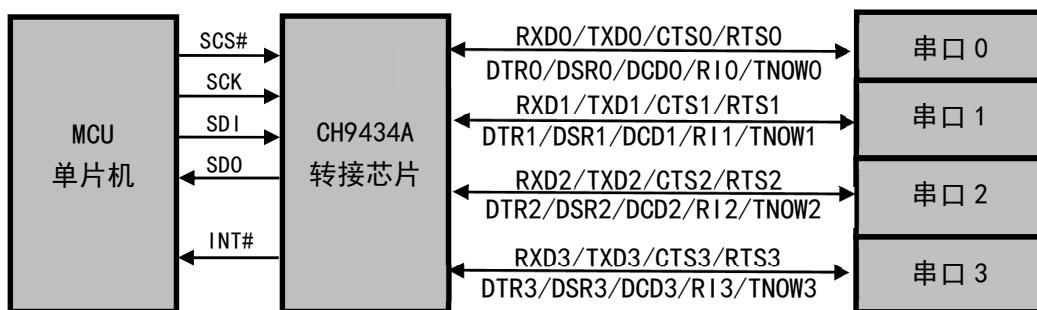
手册

版本: 1.4

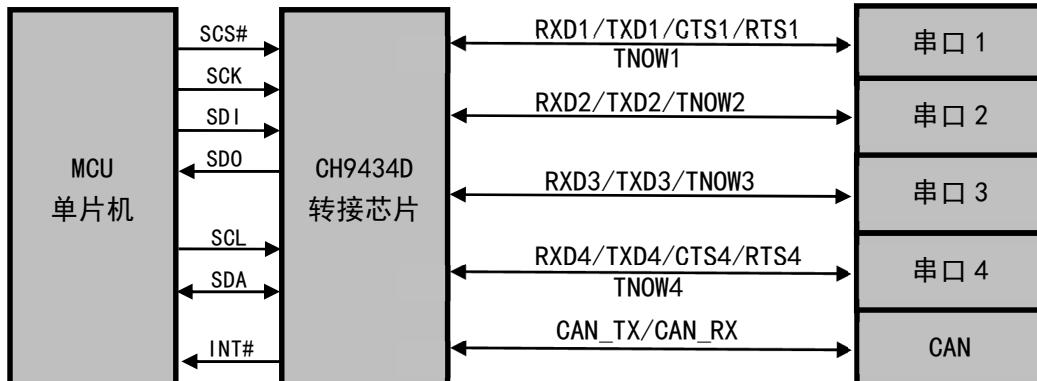
<https://wch.cn>

## 1、概述

CH9434A 是一款 SPI 转四串口转接芯片，提供四组全双工的 9 线异步串口，用于单片机/嵌入式系统扩展异步串口。CH9434A 包含四个兼容 16C550 的异步串口，最高支持 4Mbps 波特率通讯。最多支持 25 路 GPIO，提供半双工收发自动切换引脚 TNOW。



CH9434D 是一款 SPI/IIC 转四串口和一路 CAN 转接芯片，提供四组全双工的 9 线异步串口，用于单片机/嵌入式系统扩展异步串口。CH9434D 包含四个兼容 16C550 的异步串口，最高支持 6Mbps 波特率通讯。最多支持 4 路 GPIO，提供半双工收发自动切换引脚 TNOW，提供一路 CAN 收发控制器。



## 2、特点

- 工作电压: 3.3V
- 支持通讯波特率设置，CH9434A 最大支持 4Mbps，CH9434D 最大支持 6Mbps
- 串口每个方向独立 FIFO 缓存，发送 1536 字节，接收 2048 字节
- 完全独立四个异步串口
- 串口支持数据位设置以及校验位设置
- 提供半双工 RS485 收发使能引脚
- CH9434D 提供一路 CAN 收发控制器，内置收发缓冲区和过滤器
- SPI 最高可达为 20Mbit/s
- 支持低功耗睡眠模式，可通过与主控接口唤醒
- 芯片提供可配置 GPIO 功能

- CH9434A 外接 32M 晶体提供时钟，CH9434D 支持内部时钟或者外接 16M 晶体提供时钟
- CH9434A 封装为 QFN48，CH9434D 封装为 QFN20。无铅封装，兼容 RoHS

### 3、应用领域

- MCU/DSP/嵌入式系统
- 工业自动化 RS-485 通讯
- 串口服务器、多串口卡
- 与蓝牙、4G、Wi-Fi 等串口模块通讯实现无线传输

## 4、引脚排列

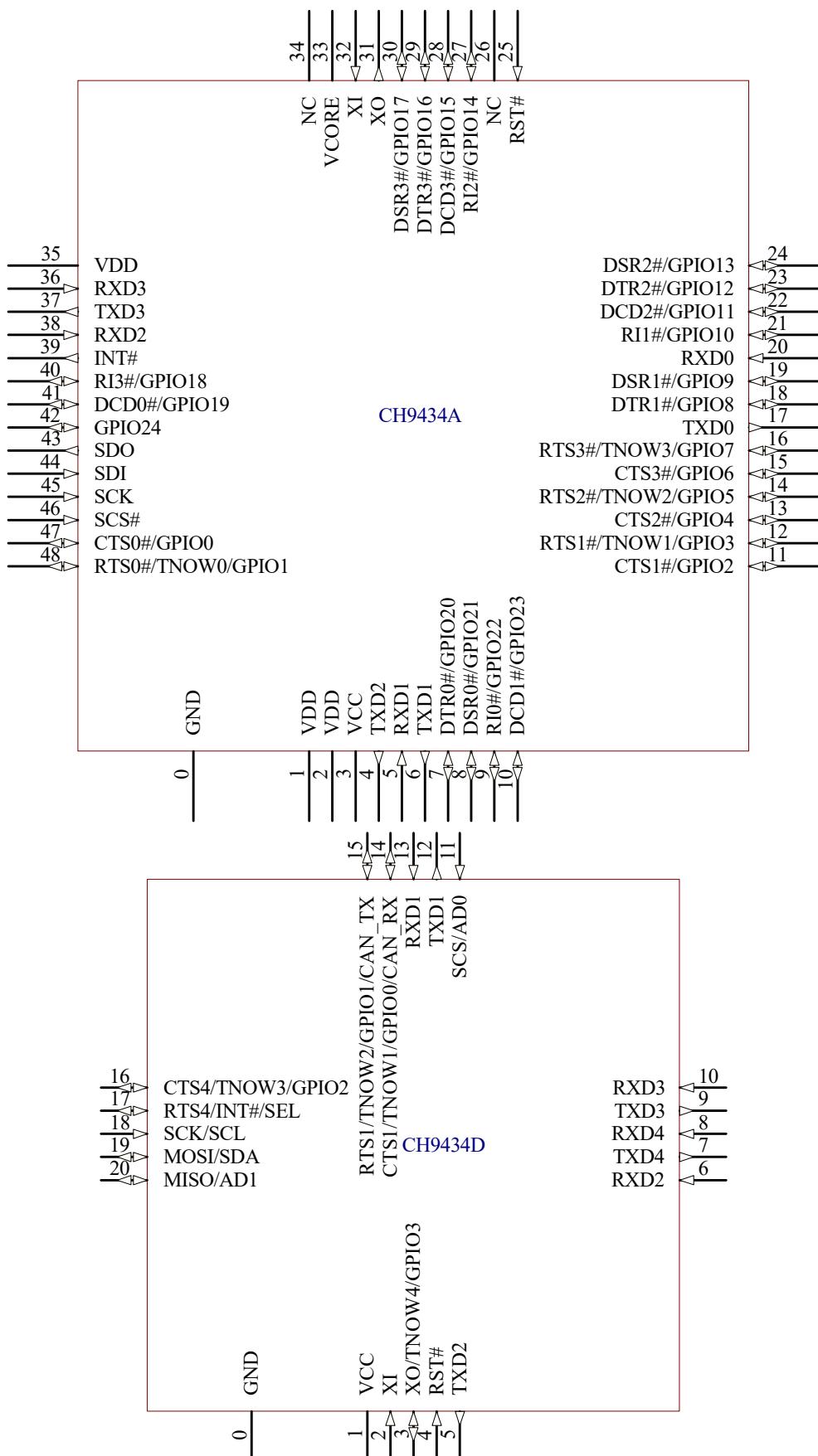


表 4-1 封装说明

| 封装形式  | 塑体尺寸  | 引脚节距   |         | 封装说明       | 订货型号    |
|-------|-------|--------|---------|------------|---------|
| QFN48 | 5*5mm | 0.35mm | 13.8mil | 四边无引线 48 脚 | CH9434A |
| QFN20 | 3*3mm | 0.4mm  | 15.8mil | 四边无引线 20 脚 | CH9434D |

注：CH9434A 基于 CH9434M 升级，引脚相同，封装兼容。

## 5、引脚定义

表 5-1 CH9434A 引脚定义

| 引脚号 | 引脚名称                | 引脚类型 <sup>(1)</sup> | 引脚说明  |
|-----|---------------------|---------------------|---|
| 0   | GND                 | P                   | 电源地。  |
| 1   | V <sub>DD</sub>     | P                   | 芯片内部电源，与其他 V <sub>DD</sub> 短接，需外接退耦电容。<br>电容值建议不小于 0.1uF。                                       |
| 2   | V <sub>DD</sub>     | P                   | 芯片内部电源，与其他 V <sub>DD</sub> 短接。  |
| 3   | V <sub>CC</sub>     | P                   | 芯片电源输入，需要外接不小于 0.1uF 电源退耦电容。  |
| 4   | TXD2                | 0                   | UART2 的串行数据输出。  |
| 5   | RXD1                | I                   | UART1 的串行数据输入。  |
| 6   | TXD1                | 0                   | UART1 的串行数据输出。  |
| 7   | DTR0# /GPIO20       | I/O                 | DTR0#：UART0 的 MODEM 输出信号，数据终端就绪。<br>GPIO20：通用双向数字 I/O 引脚。                                       |
| 8   | DSR0# /GPIO21       | I/O                 | DSR0#：UART0 的 MODEM 输入信号，数据装置就绪。<br>GPIO21：通用双向数字 I/O 引脚。                                       |
| 9   | RIO# /GPIO22        | I/O                 | RIO#：UART0 的 MODEM 输入信号，振铃指示。<br>GPIO22：通用双向数字 I/O 引脚。  |
| 10  | DCD1# /GPIO23       | I/O                 | DCD1#：UART1 的 MODEM 输入信号，载波检测。<br>GPIO23：通用双向数字 I/O 引脚。   |
| 11  | CTS1# /GPIO2        | I/O                 | CTS1#：UART1 的 MODEM 联络输入信号，清除发送，低电平有效。<br>GPIO2：通用双向数字 I/O 引脚。                                  |
| 12  | RTS1# /TNOW1 /GPIO3 | I/O                 | RTS1#：UART1 的 MODEM 联络输出信号，请求发送，低电平有效。<br>TNOW1：UART1 的 RS485 收发切换控制引脚。<br>GPIO3：通用双向数字 I/O 引脚。 |
| 13  | CTS2# /GPIO4        | I/O                 | CTS2#：UART2 的 MODEM 联络输入信号，清除发送，低电平有效。<br>GPIO4：通用双向数字 I/O 引脚。                                  |
| 14  | RTS2# /TNOW2 /GPIO5 | I/O                 | RTS2#：UART2 的 MODEM 联络输出信号，请求发送，低电平有效。<br>TNOW2：UART2 的 RS485 收发切换控制引脚。<br>GPIO5：通用双向数字 I/O 引脚。 |
| 15  | CTS3# /GPIO6        | I/O                 | CTS3#：UART3 的 MODEM 联络输入信号，清除发送，低电平有效。<br>GPIO6：通用双向数字 I/O 引脚。                                  |
| 16  | RTS3# /TNOW3 /GPIO7 | I/O                 | RTS3#：UART3 的 MODEM 联络输出信号，请求发送，低电平有效。<br>TNOW3：UART3 的 RS485 收发切换控制引脚。<br>GPIO7：通用双向数字 I/O 引脚。 |
| 17  | TXD0                | 0                   | UART0 的串行数据输出。  |

|    |                   |     |  |
|----|-------------------|-----|--|
| 18 | DTR1<br>/GPIO8    | I/O | DTR1#: UART1 的 MODEM 输出信号, 数据终端就绪。<br>GPIO8: 通用双向数字 I/O 引脚。  |
| 19 | DSR1<br>/GPIO9    | I/O | DSR1#: UART1 的 MODEM 输入信号, 数据装置就绪。<br>GPIO9: 通用双向数字 I/O 引脚。  |
| 20 | RXD0              | I   | UART0 的串行数据输入。   |
| 21 | RI1#<br>/GPIO10   | I/O | RI1#: UART1 的 MODEM 输入信号, 振铃指示。<br>GPIO10: 通用双向数字 I/O 引脚。    |
| 22 | DCD2#<br>/GPIO11  | I/O | DCD2#: UART2 的 MODEM 输入信号, 载波检测。<br>GPIO11: 通用双向数字 I/O 引脚。   |
| 23 | DTR2#<br>/GPIO12  | I/O | DTR2#: UART2 的 MODEM 输出信号, 数据终端就绪。<br>GPIO12: 通用双向数字 I/O 引脚。 |
| 24 | DSR2#<br>/GPIO13  | I/O | DSR2#: UART2 的 MODEM 输入信号, 数据装置就绪。<br>GPIO13: 通用双向数字 I/O 引脚。 |
| 25 | RST               | I   | 芯片复位引脚, 低电平有效。   |
| 26 | NC                | N   | 无效引脚, 悬空。  |
| 27 | RI2#<br>/GPIO14   | I/O | RI2#: UART2 的 MODEM 输入信号, 振铃指示。<br>GPIO14: 通用双向数字 I/O 引脚。    |
| 28 | DCD3#<br>/GPIO15  | I/O | DCD3#: UART3 的 MODEM 输入信号, 载波检测。<br>GPIO15: 通用双向数字 I/O 引脚。   |
| 29 | DTR3#<br>/GPIO16  | I/O | DTR3#: UART3 的 MODEM 输出信号, 数据终端就绪。<br>GPIO16: 通用双向数字 I/O 引脚。 |
| 30 | DSR3#<br>/GPIO17  | I/O | DSR3#: UART3 的 MODEM 输入信号, 数据装置就绪。<br>GPIO17: 通用双向数字 I/O 引脚。 |
| 31 | X0                | A   | 高频振荡器的反相输出端。   |
| 32 | X1                | A   | 高频振荡器的反相输入端。   |
| 33 | V <sub>CORE</sub> | P   | 芯片内部的电源, 需贴近引脚外接退耦电容。<br>建议不小于 0.1μF。                        |
| 34 | NC                | N   | 无效引脚, 悬空。  |
| 35 | V <sub>DD</sub>   | P   | 芯片内部电源, 与其他 V <sub>DD</sub> 短接, 需外接退耦电容。<br>电容值建议不小于 0.1μF。  |
| 36 | RXD3              | I   | UART3 的串行数据输入。   |
| 37 | TXD3              | O   | UART3 的串行数据输出。   |
| 38 | RXD2              | I   | UART2 的串行数据输入。   |
| 39 | INT#              | O   | 中断输出引脚, 低电平有效。   |
| 40 | RI3#<br>/GPIO18   | I/O | RI3#: UART3 的 MODEM 输入信号, 振铃指示。<br>GPIO18: 通用双向数字 I/O 引脚。    |
| 41 | DCD0#<br>/GPIO19  | I/O | DCD0#: UART0 的 MODEM 输入信号, 载波检测。<br>GPIO19: 通用双向数字 I/O 引脚。   |
| 42 | GPIO24            | I/O | GPIO24: 通用双向数字 I/O 引脚。                                       |
| 43 | SDO               | O   | SPI 串行数据输出。  |
| 44 | SDI               | I   | SPI 串行数据输入。  |
| 45 | SCK               | I   | SPI 串行时钟输入。  |
| 46 | SCS#              | I   | SPI 片选输入, 低电平有效。   |
| 47 | CTS0#             | I/O | CTS0#: UART0 的 MODEM 联络输入信号, 清除发送, 低电平有效。                    |

|    |                           |     |  |
|----|---------------------------|-----|--|
|    | /GPIO0                    |     | GPIO0: 通用双向数字 I/O 引脚。  |
| 48 | RTS0#<br>/TNOW0<br>/GPIO1 | I/O | RTS0#: UART0 的 MODEM 联络输出信号, 请求发送, 低电平有效。<br>TNOW0: UART0 的 RS485 收发切换控制引脚。<br>GPIO1: 通用双向数字 I/O 引脚。 |

注 1: P = 电源引脚; I = 输入引脚; O = 输出引脚; N = 空脚; 复用功能在前优先。

表 5-2 CH9434D 引脚定义

| 引脚号               | 引脚名称                                 | 引脚类型 <sup>(1)</sup> | 引脚说明   |
|-------------------|--------------------------------------|---------------------|--|
| 0                 | GND                                  | P                   | 电源地。   |
| 1                 | V <sub>cc</sub>                      | P                   | 芯片电源输入, 需要外接不小于 0.1uF 电源退耦电容。  |
| 2                 | XI                                   | A                   | XI: 高频振荡器的反相输入端。   |
| 3                 | XO<br>/TNOW4<br>/GPIO3               | I/O/A               | XO: 高频振荡器的反相输出端。<br>TNOW4: UART4 的 RS485 收发切换控制引脚。<br>GPIO3: 通用双向数字 I/O 引脚。  |
| 4                 | RST#                                 | I                   | 芯片复位引脚, 低电平有效。   |
| 5                 | TXD2 <sup>(2)</sup>                  | O                   | TXD2: UART2 的串行数据输出。   |
| 6                 | RXD2                                 | I                   | RXD2: UART2 的串行数据输入。   |
| 7                 | TXD4 <sup>(2)</sup>                  | O                   | TXD4: UART4 的串行数据输出。   |
| 8                 | RXD4                                 | I                   | RXD4: UART4 的串行数据输入。   |
| 9                 | TXD3 <sup>(2)</sup>                  | O(FT)               | TXD3: UART3 的串行数据输出。   |
| 10                | RXD3                                 | I(FT)               | RXD3: UART3 的串行数据输入。   |
| 11                | SCS<br>/ADO <sup>(4)</sup>           | I(FT)               | SCS: SPI 片选输入, 低电平有效。<br>ADO: IIC 地址设置位 0。   |
| 12                | TXD1 <sup>(2)</sup>                  | O(FT)               | TXD1: UART1 的串行数据输出。   |
| 13                | RXD1                                 | I(FT)               | RXD1: UART1 的串行数据输入。   |
| 14 <sup>(5)</sup> | CTS1<br>/TNOW1<br>/GPIO0<br>/CAN_RX  | I/O(FT)             | CTS1: UART1 的 MODEM 联络输入信号, 清除发送, 低电平有效。<br>TNOW1: UART1 的 RS485 收发切换控制引脚。<br>GPIO0: 通用双向数字 I/O 引脚。<br>CAN_RX: CAN 接口接收。 |
| 15 <sup>(5)</sup> | RTS1<br>/TNOW2<br>/GPIO1<br>/CAN_TX  | I/O(FT)             | RTS1: UART1 的 MODEM 联络输出信号, 请求发送, 低电平有效。<br>TNOW2: UART2 的 RS485 收发切换控制引脚。<br>GPIO1: 通用双向数字 I/O 引脚。<br>CAN_TX: CAN 接口发送。 |
| 16                | CTS4<br>/TNOW3<br>/GPIO2             | I/O(FT)             | CTS4: UART4 的 MODEM 联络输入信号, 清除发送, 低电平有效。<br>TNOW3: UART3 的 RS485 收发切换控制引脚。<br>GPIO2: 通用双向数字 I/O 引脚。                      |
| 17                | RTS4<br>/INT#<br>/SEL <sup>(3)</sup> | I/O(FT)             | RTS4: UART4 的 MODEM 联络输出信号, 请求发送, 低电平有效。<br>INT#: 中断输出引脚, 低电平有效。<br>SEL: 主接口模式选择。  |
| 18                | SCK<br>/SCL                          | I(FT)               | SCK: SPI 串行时钟输入。<br>SCL: IIC 时钟引脚。   |
| 19                | SDI<br>/SDA                          | I(FT)               | SDI: SPI 串行数据输入。<br>SDA: IIC 数据引脚。   |

|    |                            |             |                                       |
|----|----------------------------|-------------|---------------------------------------|
| 20 | SD0<br>/AD1 <sup>(4)</sup> | I/O<br>(FT) | SD0: SPI 串行数据输出。<br>AD1: IIC 地址设置位 1。 |
|----|----------------------------|-------------|---------------------------------------|

注 1: P: 电源引脚; I: 输入引脚; O: 输出引脚; A: 模拟引脚; N: 空脚; FT: 耐受 5V;

注 2: CH9434D 芯片的 TXDx 引脚复位完成后默认设置为推挽输出，其他功能引脚为浮空模式。命令设置相关引脚使能、I/O 功能开启后，引脚会自动设置成对应状态。

注 3: SEL 芯片复位完成后设置成上拉输入模式，用于选择与主控通讯接口，高电平选择 SPI 接口或者低电平选择 IIC 接口与主控通讯。电平判定完成后，自动撤销上拉，设置成浮空模式。

注 4: AD0/AD1 在 SEL 选择 IIC 接口时，自动设置成上拉输入，判断 IIC 地址，电平判定完成后自动撤销上拉，设置成浮空模式。AD0 还是 IIC 接口睡眠唤醒引脚。

注 5: PIN14 引脚在芯片复位期间设置为上拉输入模式，PIN15 引脚在芯片复位期间设置为下拉输入模式，芯片复位完成后设置为浮空输入模式。

## 6、时钟配置

CH9434A 需外接一个晶体配合芯片内部的时钟振荡器提供 32MHz 输入时钟。串口基准时钟可由输入时钟直接产生或开启时钟倍频再分频产生，提供给串口 0~串口 3，倍频系数固定为 15，分频系数通过 R8\_CLK\_CTRL\_CFG 寄存器进行配置。

开启时钟倍频时串口基准时钟 = 32MHz \* 倍频系数/分频系数，使用常规波特率时建议分频系数配置为 13。初始化 CH9434 时可以对芯片时钟方式进行配置，配置后需延时一段时间后进行其他操作。

CH9434D 默认使用芯片内部时钟，通过倍频后产生 96MHz 串口基准时钟，如果应用需要更精确的时钟可以外接一个晶体配合芯片内部的时钟振荡器提供 16MHz 输入时钟，主控再通过寄存器设置打开外部时钟即可，芯片会自动从内部时钟切换到外部时钟，并通过倍频产生 96MHz 串口基准时钟，时钟的切换需要延时一段时间后进行其他操作。

## 7、寄存器

### 7.1 串口寄存器

CH9434 芯片提供 4 个独立的串口模块，分别使用独立的 8 个字节寄存器进行配置。其地址分别为：00H~07H 为串口 0、10H~17H 为串口 1、20H~27H 为串口 2、30H~37H 为串口 3。串口寄存器兼容工业标准 16C550 或者 16C750 并有所增强。表中 DLAB 表示寄存器 LCR 中的位 7，X 表示不关心 DLAB 值，R0 表示寄存器只读，W0 表示寄存器只写，R/W 表示寄存器可读可写。

表 7-1 串口寄存器

| 地址 | DLAB | R/W | 名称  | 位 7      | 位 6      | 位 5      | 位 4      | 位 3      | 位 2      | 位 1      | 位 0     |
|----|------|-----|-----|----------|----------|----------|----------|----------|----------|----------|---------|
| 0  | 0    | R0  | RBR | 位 7      | 位 6      | 位 5      | 位 4      | 位 3      | 位 2      | 位 1      | 位 0     |
| 0  | 0    | W0  | THR | 位 7      | 位 6      | 位 5      | 位 4      | 位 3      | 位 2      | 位 1      | 位 0     |
| 1  | 0    | R/W | IER | RESET    | LOWPOWER | 0        | 0        | IEMODEM  | IELINES  | IETHRE   | IERECV  |
| 2  | X    | RO  | IIR | FIFOENS  | FIFOENS  | 0        | 0        | IID3     | IID2     | IID1     | NOINT   |
| 2  | X    | W0  | FCR | RECVTG1  | RECVTG0  | 0        | 0        | 0        | TFIFORST | RFIFORST | FIFOEN  |
| 3  | X    | R/W | LCR | DLAB     | BREAKEN  | PARMODE1 | PARMODE0 | PAREN    | STOPBIT  | WORDSZ1  | WORDSZ0 |
| 4  | X    | R/W | MCR | 0        | 0        | AFE      | LOOP     | OUT2     | OUT1     | RTS      | DTR     |
| 5  | X    | RO  | LSR | RFIFOERR | TEMPT    | THRE     | BREAKINT | FRAMEERR | PARERR   | OVERR    | DATARDY |
| 6  | X    | RO  | MSR | DCD      | RI       | DSR      | CTS      | △DCD     | △RI      | △DSR     | △CTS    |
| 7  | X    | R/W | SCR | 位 7      | 位 6      | 位 5      | 位 4      | 位 3      | 位 2      | 位 1      | 位 0     |
| 0  | 1    | R/W | DLL | 位 7      | 位 6      | 位 5      | 位 4      | 位 3      | 位 2      | 位 1      | 位 0     |
| 1  | 1    | R/W | DLM | 位 15     | 位 14     | 位 13     | 位 12     | 位 11     | 位 10     | 位 9      | 位 8     |

注：不同封装或型号的芯片如果无对应功能引脚则无此寄存器功能。

下表是串口寄存器在上电复位或者串口软复位之后的默认值。

表 7-2 默认值

| 寄存器名称 | 位 7 | 位 6 | 位 5 | 位 4 | 位 3 | 位 2 | 位 1 | 位 0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| IER   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| IIR   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   |
| FCR   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| LCR   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| MCR   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

|      |                       |    |     |     |    |    |    |    |
|------|-----------------------|----|-----|-----|----|----|----|----|
| LSR  | 0                     | 1  | 1   | 0   | 0  | 0  | 0  | 0  |
| MSR  | DCD                   | RI | DSR | CTS | 0  | 0  | 0  | 0  |
| SCR  | 保持                    | 保持 | 保持  | 保持  | 保持 | 保持 | 保持 | 保持 |
| FIFO | 复位，包括发送 FIFO 和接收 FIFO |    |     |     |    |    |    |    |
| TSR  | 复位，TSR 是串口发送移位寄存器     |    |     |     |    |    |    |    |
| RSR  | 复位，RSR 是串口接收移位寄存器     |    |     |     |    |    |    |    |
| 其它   | 未定义                   |    |     |     |    |    |    |    |

**RBR:** 接收缓冲寄存器，如果 LSR 的 DATARDY 位为 1 则可以从该寄存器读取接收到的数据。如果 FIFOEN 为 1 则从串口移位寄存器 RSR 接收到的数据首先被存放于接收 FIFO 中，然后通过该寄存器读出。

**THR:** 发送保持寄存器，包括发送 FIFO，用于写入准备发送的数据。如果 FIFOEN 为 1 则写入的数据首先被存放于发送 FIFO 中，然后通过发送移位寄存器 TSR 逐个输出。

**IER:** 中断使能寄存器，包括增强功能控制位以及串口中断使能。

RESET：该位置 1 则软复位该串口，该位能够自动清 0，无需软件清 0。

LOWPOWER：该位为 1 则关闭该串口的内部基准时钟，从而使该串口进入低功耗状态。

IEMODEM：该位为 1 则允许调制解调器输入状态变化中断。

IELINES：该位为 1 则允许接收线路状态中断。

IETHRE：该位为 1 则允许发送保持寄存器空中断。

IERECV：该位为 1 则允许接收到数据中断。

**IIR:** 中断识别寄存器，用于分析中断源并处理。

FIFOENS：该位为 FIFO 启用状态，为 1 表示已经启用 FIFO。

表 7-3 IIR 寄存器位

| IIR 寄存器位 |      |      |       | 优先级 | 中断类型       | 中断源                              | 清中断方法        |
|----------|------|------|-------|-----|------------|----------------------------------|--------------|
| IID3     | IID2 | IID1 | NOINT |     |            |                                  |              |
| 0        | 0    | 0    | 1     | 无   | 没有中断产生     | 没有中断                             |              |
| 0        | 1    | 1    | 0     | 1   | 接收线路状态     | OVERR、PARERR、FRAMEERR、BREAKINT   | 读 LSR        |
| 0        | 1    | 0    | 0     | 2   | 接收数据可用     | 接收到的字节数达到 FIFO 的触发点              | 读 RBR        |
| 1        | 1    | 0    | 0     | 2   | 接收数据超时     | 超过 4 个数据的时间未收到下一数据               | 读 RBR        |
| 0        | 0    | 1    | 0     | 3   | THR 寄存器空   | 发送保持寄存器空，IETHRE 从 0 变 1 可以重新使能中断 | 读 IIR 或写 THR |
| 0        | 0    | 0    | 0     | 4   | MODEM 输入变化 | △CTS、△DSR、△RI、△DCD               | 读 MSR        |

**FCR:** 先进先出缓冲区 FIFO 控制寄存器，用于使能和复位 FIFO。

RECVTG1 和 RECVTG0：设置接收 FIFO 的中断和硬件流控制的触发点，00 对应 256 个字节，即接收满 256 个字节产生接收数据可用的中断，并在使能硬件流控制时自动无效 RTS 引脚，01 对应 512 个字节，10 对应 1024 个字节，11 对应 1280 个字节。

TFIFORST：该位置 1 则清空发送 FIFO 中的数据（不含 TSR），该位能够自动清 0，无需软件清 0。

RFIFORST：该位置 1 则清空接收 FIFO 中的数据（不含 RSR），该位能够自动清 0，无需软件清 0。

FIFOEN：该位为 1 则启用 FIFO，该位清 0 则禁用 FIFO，禁用 FIFO 后为 16C450 兼容模式，相当于 FIFO 只有一个字节。注意：CH9434D 的“FIFOEN”设置为 1 开启串口传输，为 0 则为关闭串口，CH9434D 不支持单字节传输模式。

**LCR:** 线路控制寄存器，用于控制串口通讯的格式。

DLAB: 该位为除数锁存器存取使能，为 1 时才能存取 DLL 和 DLM，为 0 时才能存取 RBR/THR/IER。

BREAKEN: 该位为 1 则强制产生 BREAK 线路间隔（CH9434D 不支持）。

PARMODE1 和 PARMODE0: 当 PAREN 为 1 时设置奇偶校验位的格式：00 则奇校验，01 则偶校验，10 则标志位（MARK，置 1），11 则空白位（SPACE，清 0）。CH9434D 仅支持奇偶校验。

PAREN: 该位为 1 则允许发送时产生和接收时校验奇偶校验位，为 0 则无奇偶校验位。

STOPBIT: 该位为 1 则两个停止位，为 0 则一个停止位。

WORDSZ1 和 WORDSZ0: 设置字长度。CH9434A 设置：00 则 5 个数据位，01 则 6 个数据位，10 则 7 个数据位，11 则 8 个数据位。CH9434D 设置：00 为 9 位模式，11 为 8 位模式，其他值无效，其中 9 位模式为校验功能使用。

**MCR:** 调制解调器 MODEM 控制寄存器，用于控制 MODEM 输出。

AFE: 该位为 1 则允许 CTS 和 RTS 硬件自动流控制。如果 AFE 为 1，那么仅在检测到 CTS 引脚输入有效（低电平有效）时串口才继续发送下一个数据，否则暂停串口发送。如果 AFE 为 1 并且 RTS 为 1，那么当接收 FIFO 空时，串口会自动有效 RTS 引脚（低电平有效），直到接收的字节数达到 FIFO 的触发点时，串口才自动无效 RTS 引脚，并能够在接收 FIFO 空时再次有效 RTS 引脚。使用硬件自动控制，可将己方的 CTS 引脚接到对方的 RTS 引脚，并将己方的 RTS 引脚送到对方的 CTS 引脚。（注：CH9434D 仅支持 AFE 和 RTS 位。）

LOOP: 该位为 1 则使能内部回路的测试模式。在内部回路的测试模式下，串口所有对外输出引脚均为无效状态，TXD 内部返回到 RXD（即 TSR 的输出内部返回到 RSR 的输入），RTS 内部返回到 CTS，DTR 内部返回到 DSR，OUT1 内部返回到 RI，OUT2 内部返回到 DCD。

OUT2: 该位为 1 则允许该串口的中断请求输出，否则该串口不产生实际中断请求。

OUT1: 该位为用户可定义 MODEM 控制位，没有连接实际输出引脚。

RTS: 该位为 1 则 RTS 引脚输出有效（低电平有效），否则 RTS 引脚输出无效。

DTR: 该位为 1 则 DTR 引脚输出有效（低电平有效），否则 DTR 引脚输出无效。

**LSR:** 线路状态寄存器，用于查询方式分析串口状态。

RFIFOERR: 该位为 1 表示在接收 FIFO 中存在至少一个 PARERR、FRAMEERR 或 BREAKINT 错误。

TEMTH: 该位为 1 表示发送保持寄存器 THR 和发送移位寄存器 TSR 全空。

THRE: 该位为 1 表示发送保持寄存器 THR 空。

BREAKINT: 该位为 1 表示检测到 BREAK 线路间隔。（CH9434D 不支持）。

FRAMEERR: 该位为 1 表示正在从接收 FIFO 中读取的数据的帧错误，缺少有效的停止位。

PARERR: 该位为 1 表示正在从接收 FIFO 中读取的数据的奇偶校验错。

OVERR: 该位为 1 表示接收 FIFO 缓冲区溢出。

DATARDY: 该位为 1 表示接收 FIFO 中有接收到的数据，读取 FIFO 中所有数据后，该位自动清 0。

**MSR:** 调制解调器 MODEM 状态寄存器，用于查询 MODEM 状态。

DCD: 该位是 DCD 引脚的位反，为 1 表示 DCD 引脚有效（低电平有效）。

RI: 该位是 RI 引脚的位反，为 1 表示 RI 引脚有效（低电平有效）。

DSR: 该位是 DSR 引脚的位反，为 1 表示 DSR 引脚有效（低电平有效）。

CTS: 该位是 CTS 引脚的位反，为 1 表示 CTS 引脚有效（低电平有效）。

△DCD: 该位为 1 表示 DCD 引脚输入状态发生过变化。

△RI: 该位为 1 表示 RI 引脚输入状态发生过变化。

△DSR: 该位为 1 表示 DSR 引脚输入状态发生过变化。

△CTS: 该位为 1 表示 CTS 引脚输入状态发生过变化。

(注：CH9434D 仅支持 CTS 和△CTS 位)

**SCR:** 用户可定义寄存器。

**DLL 和 DLM:** 波特率除数锁存器，DLL 是低字节，DLM 是高字节，两者组成的 16 位除数用于由 16 位计数器构成的串口波特率产生器。

CH9434A 定义该除数=串口基准时钟/8/所需通讯波特率。

CH9434D 定义 DLM 和 DLL 组成 16 位数据，高 12 位为整数分频 (DIV\_M:[15:4])，低 4 位为小数分频 (DIV\_F:[3:0])，计算通讯波特率=串口基准时钟/(16\*(DIV\_M+DIV\_F/16))，根据波特率计算公式得出的分频可能会有偏差，需要尽量选取接近的值。

## 7.2 接口寄存器

表 7-4 接口寄存器

| 地址  | 名称               | 功能         | 位     | 访问  | 位定义                                   | 默认值   |
|-----|------------------|------------|-------|-----|---------------------------------------|-------|
| 41H | R8_TNOW_CTRL_CFG | TNOW 功能设置  | [7:4] | R/W | TNOW 引脚电平反相                           | 0000b |
|     |                  |            | [3:0] | R/W | TNOW 引脚功能控制                           | 0000b |
| 42H | R8_FIFO_CTRL     | FIFO 计数器设置 | [7:5] | R0  | 保留                                    | 000b  |
|     |                  |            | 4     | R/W | 收发缓存区控制<br>0: 接收 FIFO;<br>1: 发送 FIFO。 | 0     |
|     |                  |            | [3:0] | R/W | 串口号                                   | 0000b |
| 43H | R8_FIFO_CNT_L    | FIFO 计数器   | [7:0] | R0  | FIFO 计数器低 8 位                         | XXh   |
| 44H | R8_FIFO_CNT_H    | FIFO 计数器   | [7:0] | R0  | FIFO 计数器高 8 位                         | XXh   |
| 45H | IO_SEL_FUN_CFG   | 引脚功能复用使能   | [7:0] | R/W | 控制命令                                  | 00h   |
|     |                  |            | [7:0] | R/W | 通讯数据 0                                | 00h   |
|     |                  |            | [7:0] | R/W | 通讯数据 1                                | 00h   |
|     |                  |            | [7:0] | R/W | 通讯数据 2                                | 00h   |
| 46H | CAN_REG          | CAN 寄存器访问  | [7:0] | WO  | CAN 寄存器地址                             | 00h   |
|     |                  |            | [7:0] | R/W | CAN 寄存器读写值                            | XXh   |

注：IO\_SEL\_FUN\_CFG 和 CAN\_REG 仅 CH9434D 支持。

**R8\_TNOW\_CTRL\_CFG:** 设置 TNOW 功能开启和极性，极性默认发送数据时，对应 TNOW 引脚拉高，发送结束 TNOW 拉低。

**R8\_FIFO\_CTRL:** FIFO 计数器设置，设置 FIFO 的方向以及串口号后，读取 R8\_FIFO\_CNT\_L 和 R8\_FIFO\_CNT\_H 的值后对应读取的 FIFO 计数器。其中：读取串口发送 FIFO 计数器值为当前剩余 FIFO 数量，主控可以根据这个值填入最大数量的发送数据；读取接收 FIFO 计数器值位当前已接收数据数量，主控可以根据这个值一次将串口数据全部读走。

**IO\_SEL\_FUN\_CFG:** 芯片引脚功能复用设置，通讯时必须按照 4 字节数据完整读写，否则会导致设置失败，功能详见引脚功能使能说明。

**CAN\_REG:** CAN 接口寄存器读写控制，“CAN 寄存器地址”填入需要读写的寄存器地址，读模式下“CAN 寄存器读写值”返回的是寄存器 32 位值，写模式下在“CAN 寄存器读写值”里填入设置寄存器的 32 位值。CAN 相关寄存器全部按照 32 位访问，禁止其他操作方式。CAN 寄存器功能详见 CAN 功能说明。

## 7.3 时钟电源寄存器

表 7-5 时钟电源寄存器

| 地址  | 名称               | 功能                         | 位     | 访问  | 位定义   | 默认值    |
|-----|------------------|----------------------------|-------|-----|---|--------|
| 48H | R8_CLK_CTRL_CFG  | 芯片时钟设置<br>(CH9434D 仅切换时钟源) | [7:6] | R/W | HCLK 系统时钟源模式选择：<br>0x：外部晶振；<br>1x：使用外部晶振并开启倍频。  | 00b    |
|     |                  |                            | 5     | R0  | 保留。   | 0      |
|     |                  |                            | [4:0] | R/W | 分频系数。   | 00000b |
| 4AH | R8_SLEEP_MOD_CFG | 睡眠功能设置                     | [7:3] | R0  | 保留。   | 00000b |
|     |                  |                            | [2:0] | R/W | 芯片低功耗设置：<br>CH9434A<br>0：不进入低功耗状态；<br>1：SLEEP 状态。<br>CH9434D<br>0：不进入低功耗状态；<br>1：IDLE 状态；<br>2：SLEEP 状态；<br>3：SHUT 状态；<br>4-7：保留。 | 000b   |

## 7.4 GPIO 寄存器

表 7-6 CH9434A GPIO 寄存器

| 地址  | 名称                | 功能                | 位     | 访问  | 位定义  | 默认值       |
|-----|-------------------|-------------------|-------|-----|--|-----------|
| 50H | R8_GPIO_FUNCEN_0  | GPIO7-0<br>使能     | [7:0] | R/W | [7:0] 对应 GPIO7-0 使能控制：<br>0: 关闭 GPIO 功能；<br>1: 开启 GPIO 功能。 | 00h       |
| 51H | R8_GPIO_FUNC_EN_1 | GPIO15-8<br>使能    | [7:0] | R/W | [7:0] 对应 GPIO15-8 使能控制，值定义同上。                              | 00h       |
| 52H | R8_GPIO_FUNC_EN_2 | GPIO23-16<br>使能   | [7:0] | R/W | [7:0] 对应 GPIO23-16 使能控制，值定义同上。                             | 00h       |
| 53H | R8_GPIO_FUNC_EN_3 | 保留                | [7:1] | R0  | 保留。  | 00000000b |
|     |                   | GPIO24<br>使能      | 0     | R/W | 对应 GPIO24 使能控制，值定义同上。                                      | 0b        |
| 54H | R8_GPIO_DIR_MOD_0 | GPIO7-0<br>方向设置   | [7:0] | R/W | [7:0] 对应 GPIO7-0 方向设置：<br>0: GPIO 设置成输入；<br>1: GPIO 设置成输出。 | 00h       |
| 55H | R8_GPIO_DIR_MOD_1 | GPIO15-8<br>方向设置  | [7:0] | R/W | [7:0] 对应 GPIO15-8 方向设置，值定义同上。                              | 00h       |
| 56H | R8_GPIO_DIR_MOD_2 | GPIO23-16<br>方向设置 | [7:0] | R/W | [7:0] 对应 GPIO23-16 方向设置，值定义同上。                             | 00h       |
| 57H | R8_GPIO_DIR_MOD_3 | 保留                | [7:1] | R0  | 保留。  | 00000000b |
|     |                   | GPIO24<br>方向设置    | 0     | R/W | 对应 GPIO24 方向设置，值定义同上。                                      | 0b        |
| 58H | R8_GPIO_PU_MOD_0  | GPIO7-0<br>上拉电阻设置 | [7:0] | R/W | [7:0] 对应 GPIO7-0 上拉电阻设置：<br>0: 关闭 GPIO 上拉电阻；               | 00h       |

|     |                   |                                    |       |     |  |           |
|-----|-------------------|------------------------------------|-------|-----|--|-----------|
|     |                   |                                    |       |     | 1:开启 GPIO 上拉电阻。  |           |
| 59H | R8_GPIO_PU_MOD_1  | GPIO15-8<br>上拉电阻设置                 | [7:0] | R/W | [7:0]对应 GPIO15-8 上拉电阻设置，值定义同上。   | 00h       |
| 5AH | R8_GPIO_PU_MOD_2  | GPIO23-16<br>上拉电阻设置                | [7:0] | R/W | [7:0]对应 GPIO23-16 上拉电阻设置，值定义同上。  | 00h       |
| 5BH | R8_GPIO_PU_MOD_3  | 保留                                 | [7:1] | R0  | 保留。  | 00000000b |
|     |                   | GPIO24<br>上拉电阻设置                   | 0     | R/W | 对应 GPIO24 上拉电阻设置，值定义同上。  | 0b        |
| 5CH | R8_GPIO_PD_MOD_0  | GPIO7-0<br>下拉电阻设置                  | [7:0] | R/W | [7:0]对应 GPIO7-0 下拉电阻设置：<br>0:关闭 GPIO 下拉电阻；<br>1:开启 GPIO 下拉电阻。  | 00h       |
| 5DH | R8_GPIO_PD_MOD_1  | GPIO15-8<br>下拉电阻设置                 | [7:0] | R/W | [7:0]对应 GPIO15-8 下拉电阻设置，值定义同上。   | 00h       |
| 5EH | R8_GPIO_PD_MOD_2  | GPIO23-16<br>下拉电阻设置                | [7:0] | R/W | [7:0]对应 GPIO23-16 下拉电阻设置，值定义同上。  | 00h       |
| 5FH | R8_GPIO_PD_MOD_3  | 保留                                 | [7:1] | R0  | 保留。  | 00000000b |
|     |                   | GPIO24<br>下拉电阻设置                   | 0     | R/W | 对应 GPIO24 下拉电阻设置，值定义同上。  | 0b        |
| 60H | R8_GPIO_PIN_VAL_0 | GPIO7-0<br>电平输出、电<br>平读取和上下<br>拉设置 | [7:0] | R/W | [7:0]对应 GPIO7-0 引脚电平写入寄存器。<br>GPIO 为输出时：<br>1: 输出高电平；<br>0: 输出低电平。<br>GPIO 为输入且开启上下拉电阻功能时：<br>1: 设置 IO 上拉电阻；<br>0: 设置 IO 下拉电阻。<br>读取寄存器 GPIO 为输入时：<br>该引脚电平状态。 | 00h       |
| 61H | R8_GPIO_PIN_VAL_1 | GPIO15-8<br>输入输出电平                 | [7:0] | R/W | [7:0]对应 GPIO15-8 引脚电平，值定义同上。   | 00h       |
| 62H | R8_GPIO_PIN_VAL_2 | GPIO23-16<br>输入输出电平                | [7:0] | R/W | [7:0]对应 GPIO23-16 引脚电平，值定义同上。  | 00h       |
| 63H | R8_GPIO_PIN_VAL_3 | 保留                                 | [7:1] | R0  | 保留。  | 00000000b |
|     |                   | GPIO24<br>输入输出电平                   | 0     | R/W | 对应 GPIO24 引脚电平，值定义同上。  | 0b        |

表 7-7 CH9434D GPIO 寄存器

|     |                   |                |       |     |   |     |
|-----|-------------------|----------------|-------|-----|---|-----|
| 50H | R8_GPIO_FUNC_EN_0 | GPIO7-0<br>使能  | [7:0] | R/W | [7:0]对应 GPIO7-0 使能控制：<br>0:关闭 GPIO 功能；<br>1:开启 GPIO 功能。 | 00h |
| 51H | R8_GPIO_FUNC_EN_1 | GPIO15-8<br>使能 | [7:0] | R/W | [7:0]对应 GPIO15-8 使能控制，值定义同上。                            | 00h |
| 54H | R8_GPIO_DIR_MOD_0 | GPIO1-0        | [7:0] | R/W | [7:4]对应 GPIO1 模式设置。                                     | 00h |

|     |                   |                   |       |     |   |     |
|-----|-------------------|-------------------|-------|-----|---|-----|
|     |                   | 模式设置              |       |     | [3:0] 对应 GPIO0 模式设置。<br>每一个 4bit 定义：<br><br>[3:2]：该 IO 功能配置<br>输入模式下：<br>01：浮空输入；<br>10：带有上下拉电阻输入。<br><br>输出模式下：<br>00：推挽输出；<br>01：开漏输出。<br><br>[1:0]：该 IO 的模式设置：<br>00：输入模式；<br>11：输出模式。 |     |
| 55H | R8_GPIO_DIR_MOD_1 | GPIO3-2<br>模式设置   | [7:0] | R/W | [7:4] 对应 GPIO3 模式设置；<br>[3:0] 对应 GPIO2 模式设置；<br>值定义同上。  | 00h |
| 56H | R8_GPIO_DIR_MOD_2 | GPIO5-4<br>模式设置   | [7:0] | R/W | [7:4] 对应 GPIO5 模式设置；<br>[3:0] 对应 GPIO4 模式设置；<br>值定义同上。  | 00h |
| 57H | R8_GPIO_DIR_MOD_3 | GPIO7-6<br>模式设置   | [7:0] | R/W | [7:4] 对应 GPIO7 模式设置；<br>[3:0] 对应 GPIO6 模式设置；<br>值定义同上。  | 00h |
| 58H | R8_GPIO_DIR_MOD_4 | GPIO9-8<br>模式设置   | [7:0] | R/W | [7:4] 对应 GPIO9 模式设置；<br>[3:0] 对应 GPIO8 模式设置；<br>值定义同上。  | 00h |
| 59H | R8_GPIO_DIR_MOD_5 | GPIO11-10<br>模式设置 | [7:0] | R/W | [7:4] 对应 GPIO11 模式设置；<br>[3:0] 对应 GPIO10 模式设置；<br>值定义同上。  | 00h |
| 5AH | R8_GPIO_DIR_MOD_6 | GPIO13-12<br>模式设置 | [7:0] | R/W | [7:4] 对应 GPIO13 模式设置；<br>[3:0] 对应 GPIO12 模式设置；<br>值定义同上。  | 00h |
| 5BH | R8_GPIO_DIR_MOD_7 | GPIO15-14<br>模式设置 | [7:0] | R/W | [7:4] 对应 GPIO15 模式设置；<br>[3:0] 对应 GPIO14 模式设置；<br>值定义同上。  | 00h |
| 5CH | R8_GPIO_SET_0     | GPIO7-0<br>置位     | [7:0] | R/W | [7:0] 对应 GPIO7-0 置位控制<br>对应位写 1 后，IO 输出高电平，<br>写 0 无效，同时影响<br>“GPIO_PIN_VAL” 寄存器。   | 00h |
| 5DH | R8_GPIO_SET_1     | GPIO15-8<br>置位    | [7:0] | R/W | [7:0] 对应 GPIO15-8 置位控制<br>值定义同上。  | 00h |
| 5EH | R8_GPIO_RESET_0   | GPIO7-0<br>复位     | [7:0] | R/W | [7:0] 对应 GPIO7-0 复位控制<br>对应位写 1 后，IO 输出低电平，<br>写 0 无效，同时影响  | 00h |

|     |                   |                            |       |     |   |     |
|-----|-------------------|----------------------------|-------|-----|---|-----|
|     |                   |                            |       |     | “GPIO_PIN_VAL” 寄存器。   |     |
| 5FH | R8_GPIO_RESET_1   | GPIO15-8<br>复位             | [7:0] | R/W | [7:0] 对应 GPIO15-8 复位控制值定义同上。  | 00h |
| 60H | R8_GPIO_PIN_VAL_0 | GPIO7-0<br>电平输出、电平读取和上下拉设置 | [7:0] | R/W | [7:0] 对应 GPIO7-0 引脚电平写入寄存器。<br>GPIO 为输出时：<br>1：输出高电平；<br>0：输出低电平。<br>GPIO 为输入且开启上下拉电阻功能时：<br>1：设置 IO 上拉电阻；<br>0：设置 IO 下拉电阻。<br>读取寄存器 GPIO 为输入时：<br>该引脚电平状态。 | 00h |
| 61H | R8_GPIO_PIN_VAL_1 | GPIO15-8<br>输入输出电平         | [7:0] | R/W | [7:0] 对应 GPIO15-8 引脚电平，值定义同上。   | 00h |

注：不同封装或型号的芯片如果无对应功能引脚则无此寄存器功能。

## 7.5 扩展寄存器

表 7-8 扩展寄存器

| 地址  | 名称                | 功能       | 位     | 访问  | 位定义   | 默认值 |
|-----|-------------------|----------|-------|-----|---|-----|
| 64H | EXT_SPI_CONT_MODE | SPI 传输模式 | [7:1] | R/W | 保留。   | 0   |
|     |                   |          | 0     | R/W | RBR 支持连续模式：<br>0：关闭 RBR 连续读取模式；<br>1：开启 RBR 连续读取模式。 | 0   |
| 65H | CHIP_VER_ADD      | 当前固件版本   | [7:0] | R0  | VER[0]：当前版本号低 8 位。                                  | XXh |
|     |                   |          | [7:0] | R0  | VER[1]：当前版本号高 8 位。                                  | XXh |
|     |                   |          | [7:0] | R0  | VER[2]：版本信息校验。                                      | XXh |
|     |                   |          | [7:0] | R0  | VER[3]：结束字符。  | XXh |

CH9434A 的 EXT\_SPI\_CONT\_MODE 和 CHIP\_VER\_ADD 寄存器是版本 v1.02 开始支持；需注意不支持 CHIP\_VER\_ADD 寄存器的版本读取时解析版本校验数据，版本信息格式 BCD 码。

CH9434D 无需设置 “EXT\_SPI\_CONT\_MODE” 寄存器

## 8、功能说明

### 8.1 中断与查询

CH9434 芯片的 INT#引脚为中断请求输出引脚，CH9434A 默认引脚在芯片上电后为推挽输出模式，低电平有效；CH9434D 芯片的 INT#引脚在芯片上电后为浮空模式，使能后为推挽输出模式，低电平有效。由于四路串口共用一个中断引脚输出，所以当有中断有效信号时，主控 MCU 需要查询所有串口的中断状态以分析是哪个串口有中断请求。

如果串口有多个中断请求输出，当主控 MCU 读取一个有效的中断状态后，CH9434 会将中断引脚暂时拉高，之后再拉低。主控 MCU 可以查询判断，只要中断引脚为低电平就直接进行串口中断状态查询。

### 8.2 串口操作

串口的操作提供了 IIR 和 LSR 寄存器查询串口的收发状态，CH9434 还扩展了一个 FIFO 寄存器用于直接获取当前芯片的串口收发数据量。

当串口中断查询到有接收数据后，可以先读取当前串口的接收 FIFO 长度，然后直接操作 RBR 寄存器读取所有数据。主控 MCU 也可以简化处理直接定时查询 FIFO 当前接收数据数量，然后根据接收 FIFO 的数量操作 RBR 寄存器读取数据。

发送数据可以查询 IIR 和 LSR 寄存器的发送缓存区空状态，然后操作 THR 寄存器进行串口发送数据。主控 MCU 也可以简化处理使用查询 FIFO 大小进行数据传输，查询发送 FIFO 数据长度的值为剩余 FIFO 大小，主控 MCU 可以以此为大小进行将发送数据依次填入 THR 寄存器进行数据发送。

串口流控功能使能是将 AFE 位置 1，CH9434 将自动进行硬件流控。芯片将自动根据 FIFO 大小对流控引脚进行操作。启用自动流控后，CTS 有效时芯片串口将连续发送数据，CTS 引脚无效时，CH9434A 串口最多发送 8 字节数据后停止发送；CH9434D 则立即停止发送。RTS 在触发 FIFO 达到设定的流控字节数目后自动失效。

### 8.3 RS485切换引脚 TNOW

CH9434 串口提供 RS485 切换引脚 TNOW，引脚功能与其他功能复用，TNOW 还支持极性调节，以适应不同的极性使用场景。

### 8.4 GPIO 功能

CH9434 支持功能引脚复用为 GPIO 功能，CH9434A 最多支持 25 路，CH9434D 最多支持 4 路。GPIO 可以设置方向、上拉电阻和下拉电阻配置。GPIO 功能在设置时需要注意该 IO 原来功能的模式和 IO 寄存器的设置顺序，防止 IO 出现“抖动”。

### 8.5 低功耗模式

CH9434 支持低功耗模式设置。CH9434A 仅支持 SLEEP 模式，设置 SLEEP 模式后，芯片暂停运行，电流功耗可降至 4.8mA，可以直接操作 SPI 进行唤醒。

CH9434D 支持 3 档睡眠模式，分别为芯片停止运行的 IDLE 模式、关闭时钟运行的 SLEEP 模式和 SHUT 掉电睡眠模式，唤醒引脚 SPI 接口和 IIC 都使用 CS 引脚，需要注意的是 IIC 接口下 CS 唤醒电平由当前 IO 电平决定，如果是高电平则为下降沿唤醒，反之则为上升沿唤醒主控需注意电平状态防止产生漏电流。

### 8.6 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 和串行数据输出引脚 SDO。SPI 数据位序：高位在前。发送数据格式为第一个字节为操作地址，第二个字节为写入的数据或者读取的数据，操作地址的最高位为操作位，操作位为 1 则为写入数据，为 0 则为读取数据。

CH9434A 在写入寄存器数据时，地址和数据两个字节间需要 1μs 延时，发送完数据后需要延时 3μs 才可以进行下一次操作。在读取数据时，地址和数据需要延时 3μs，即发送完地址后延时 3μs 进行数据读取。

CH9434D 在写入寄存器数据时，地址和数据两个字节间需要 2μs 延时，发送完数据 CS 释放后需要延时 3μs 才可以进行下一次操作。在读取寄存器数据时，地址和数据需要延时 3μs，即发送完地址后延时 3μs 进行数据读取，读取数据完成 CS 释放后需要延时 3μs 才可以进行下一次操作。

### 8.7 扩展寄存器

CH9434A：命令在版本 v1.02 版本后支持。

开启 RBR 连续读取模式功能后，SPI 接口频率需设置成 2MHz，在数据接收中断触发后，主控 MCU

读取当前 FIFO 接收长度 len，读取 RBR 时，按照 len+1 长度进行连续读取，读取完成后删除第一个字节即可。此模式推荐在系统应用无法控制 uS 延时下使用，可以控制延时的系统或主控 MCU 可以直接通过延时去连续读取。具体的功能可参考相关例子程序。

RBR 接口读取流程 (len=n): RBR+DATA1+DATA2+…+DATAn+DATAn+1，地址和数据之间的延时和数据之间的延时需要去掉，流程结束后保存 DATA2 到 DATAn+1 数据。

CH9434D：默认支持连续读写串口数据寄存器，无需对寄存器进行设置。传输方式：传输寄存器地址后，延时 3uS，后面数据部分可以连续读写，直到缓存结束。

## 8.8 版本号

版本号读取需要读取 4 字节版本信息，其中 VER[0] 和 VER[1] 为版本信息，解析格式为 BCD 码，VER[1] 字段高 4 位为保留信息(固定为 0)，VER[2] 为版本信息校验字段，值为 (VER[0]+VER[1])，VER[3] 为结束字段，值固定为 0x5A，CH9434D 为 0x6B。主控 MCU 在读取的时候需要进行校验字段解析判断版本号是否合法，如果校验字段解析成功则说明芯片有版本信息。

## 8.9 引脚功能使能说明（仅 CH9434D 支持）

CH9434D 芯片上电完成后，TXDx 引脚默认为上拉输出模式，其余功能引脚默认为浮空模式，需要通过主控接口设置后才会打开引脚，如串口引脚、晶振引脚、INT 引脚等。通过 IO\_SEL\_FUN\_CFG 寄存器对引脚功能进行设置。引脚复用寄存器读写按照 4 字节处理，第一字节为命令码，后续三个字节为数据；寄存器对复用引脚、默认引脚进行设置，寄存器进行读写时，需要按照：执行操作—等待应答，两步进行，其中数据内容第三字节为状态标志，主控写入时需置为 0xA5，芯片操作执行完成后会将第三字节置为 0x5A，主控执行操作后建议延时读取，直到第三字节为 0x5A，表示执行结束，如果没有读取到可以重试读取。

主控在 CH9434D 初始化阶段，需要先按照应用对 IO 进行打开操作，否则会导致该 IO 功能无法使用。

寄存器设置命令和数据：

| 控制命令 | 功能           | 数据内容，“[]”内为单字节定义                                    |
|------|--------------|---|
| 0x01 | 设置复用功能使能     | 主控写入：[复用引脚地址][使能][0xA5]<br>读取返回：[复用引脚地址][无定义][0x5A] |
| 0x81 | 读取对应复用引脚使能状态 | 主控写入：[复用引脚地址][无定义][0xA5]<br>读取返回：[复用引脚地址][使能][0x5A] |
| 0x03 | 设置默认引脚使能     | 主控写入：[默认引脚地址][使能][0xA5]<br>读取返回：[默认引脚地址][无定义][0x5A] |
| 0x83 | 读取默认引脚使能状态   | 主控写入：[默认引脚地址][无定义][0xA5]<br>读取返回：[默认引脚地址][使能][0x5A] |

默认引脚地址：使用时打开即可。

| 默认引脚功能      | 默认引脚地址 | 默认引脚 PIN 序号 |
|-------------|--------|-------------|
| UART1_TX_RX | 1      | PIN12、PIN13 |
| UART2_TX_RX | 2      | PIN5、PIN6   |
| UART3_TX_RX | 3      | PIN9、PIN10  |
| UART4_TX_RX | 4      | PIN7、PIN8   |
| HSE_XI_XO   | 5      | PIN2、PIN3   |
| CAN_TX_RX   | 6      | PIN14、PIN15 |
| 保留          | 7      |             |
| CTS1        | 8      | PIN14       |

|      |   |       |
|------|---|-------|
| CTS4 | 9 | PIN16 |
|------|---|-------|

复用引脚地址与对应默认 PIN 脚序号：

| 复用引脚功能 | 复用引脚地址 | 默认 PIN 脚序号 |
|--------|--------|------------|
| INT    | 1      | PIN17      |
| TNOW1  | 2      | PIN14      |
| TNOW2  | 3      | PIN15      |
| TNOW3  | 4      | PIN16      |
| TNOW4  | 5      | PIN3       |
| 保留     | 6-7    |            |
| RTS1   | 8      | PIN15      |
| 保留     | 9-10   |            |
| RTS4   | 11     | PIN17      |
| GPIO0  | 12     | PIN14      |
| GPIO1  | 13     | PIN15      |
| GPIO2  | 14     | PIN16      |
| GPIO3  | 15     | PIN3       |

## 8.10 CAN 功能说明

CAN 功能 CH9434D 支持，CH9434A 不支持。CAN 寄存器必须使用 32 位字的方式来操作。数据内容格式按照小端处理。

CAN 功能主要特性：

- 兼容 CAN 规范 2.0A 和 2.0B
- 可编程的传输速率，最高可达 1Mbit/s
- 支持时间触发通信功能，避免低优先级消息阻塞
- 支持三个发送邮箱，发送报文优先级可由报文标识符或发送请求的次序决定，并可记录发送报文 SOF 时刻的时间戳
- 支持 18 级邮箱深度的 2 个接收 FIFO，14 个报文过滤器组可供配置，每个过滤器组可配置成 32 或 16 位模式，屏蔽位或标识符列表模式，能够尽量减少软件对报文筛选的干预，FIFO 溢出处理方式灵活，并可记录接收报文 SOF 时刻的时间戳

CAN 发送流程：

发送处理流程如下：如果三个发送邮箱中有空置的邮箱，应用层软件仅对空置邮箱的寄存器具有写入权限，对寄存器 CAN\_TXMIRx、CAN\_TXMDTRx、CAN\_RXMDLRx 和 CAN\_RXMDH Rx 进行操作，可以设置报文标识符、报文长度、时间戳和报文数据等。在数据准备好之后，对寄存器 CAN\_TXMIRx 的 TXRQ 位置 1 请求发送，邮箱进入挂号状态，并进行优先级排队；一旦成为最高优先级邮箱，则变为预定发送状态，等待 CAN 总线空闲；当 CAN 总线空闲时，预定发送邮箱的报文立刻进入发送状态；报文发送完毕后，邮箱重新成为空置邮箱，并且寄存器 CAN\_TSTATR 的 RQCP 和 TXOK 位置 1，来指示发送成功；若发送时仲裁失败，寄存器 CAN\_TSTATR 的 ALST 位置 1，若发送错误，则 TERR 位置 1。

CAN 接收流程：

所接收到的报文，根据寄存器 CAN\_FAFIFOR 的设置，分别被存储到两个具有 18 级邮箱深度的 FIFO 中，主控如需获取报文，只能通过接收 FIFO 邮箱来读取有效接收报文。

初始时，接收 FIFO 为空，接收 FIFO 寄存器 CAN\_RFIFOx 的 FMR[7:0]值为 0，接到一个有效接收报文后，变为挂号 1 状态，控制器自动把接收 FIFO 寄存器 CAN\_RFIFOx 的 FMR[7:0]设置为 1；若此时读取邮箱数据寄存器 CAN\_RXMDLRx 和 CAN\_RXMDH Rx，通过对接收 FIFO 寄存器 CAN\_RFIFOx 的 RFOM 位置 1 来释放邮箱，接收 FIFO 状态又变为空；如果在挂号 1 状态时不释放邮箱，下一个有效接收报文

被接到后，接收 FIFO 状态切换为挂号 2 状态，此时接收 FIFO 寄存器 CAN\_RFIFOx 的 FMR[7:0]自动置 2；若读取邮箱数据寄存器并释放邮箱，则状态回到挂号 1；如果在挂号 2 状态不释放邮箱，则接收 FIFO 进入挂号 3 状态；同样在挂号 3 状态下读取报文并释放邮箱，则返回挂号 2 状态；以此类推，若在挂号 18 状态不释放邮箱，则在接收到下一个有效报文时，必然导致报文丢失情况出现。

#### CAN 过滤流程：

模块中有着多达 14 个过滤器组，通过设置过滤器组，每个 CAN 节点都可以接收到符合过滤规则的报文，不符合过滤规则的报文被硬件丢弃。

每个过滤器组由 2 个 32 位寄存器 CAN\_FxR0 和 CAN\_FxR1 组成。过滤器组的位宽都可以通过设置寄存器 CAN\_FSCFGR 的各个位独立配置成 1 个 32 位过滤器或两个 16 位过滤器。每个过滤器组可通过设置寄存器 CAN\_FMCFGR 的各个位配置为屏蔽位或标识符列表模式，各个过滤器组可以通过设置寄存器 CAN\_FWR 的各个位选择启用或禁用。设置寄存器 CAN\_FAFFIFOR 的各个位可以把选择通过过滤器的报文存放到哪个接收 FIFO。

如下表所示，屏蔽位模式下，两个寄存器分别为标识符寄存器和屏蔽寄存器，两者需要配合使用，标识符寄存器每一位指示相应的位期望值为显性或隐性，屏蔽寄存器每一位指示相应位是否需要对应标识符寄存器位期望值一致。

32 位屏蔽位模式下寄存器定义

|            |                 |                       |                |                     |
|------------|-----------------|-----------------------|----------------|---------------------|
| 标识符<br>寄存器 | CAN_FxR1[31:24] | CAN_FxR1[23:16]       | CAN_FxR1[15:8] | CAN_FxR1[7:0]       |
| 屏蔽位<br>寄存器 | CAN_FxR2[31:24] | CAN_FxR2[23:16]       | CAN_FxR2[15:8] | CAN_FxR2[7:0]       |
| 映射         | STID[10:3]      | STID[2:0] EXID[17:13] | EXID[12:5]     | EXID[4:0] IDE RTR 0 |

标识符列表模式下，两个寄存器都被用作标识符寄存器，接收报文标识符必须与其中一个寄存器保持一致才能通过筛选。

32 位标识符列表模式下寄存器定义

|            |                 |                       |                |                     |
|------------|-----------------|-----------------------|----------------|---------------------|
| 标识符<br>寄存器 | CAN_FxR1[31:24] | CAN_FxR1[23:16]       | CAN_FxR1[15:8] | CAN_FxR1[7:0]       |
| 屏蔽位<br>寄存器 | CAN_FxR2[31:24] | CAN_FxR2[23:16]       | CAN_FxR2[15:8] | CAN_FxR2[7:0]       |
| 映射         | STID[10:3]      | STID[2:0] EXID[17:13] | EXID[12:5]     | EXID[4:0] IDE RTR 0 |

在 16 位模式下，寄存器组被拆分成四个寄存器，屏蔽位模式每组过滤器的屏蔽位模式可以有 2 个过滤器，每个过滤器里各包含一个 16 位标识符寄存器和 16 位屏蔽寄存器；标识符列表模式下四个寄存器都用作标识符寄存器。

16 位屏蔽位模式下寄存器定义

|            |                 |                               |
|------------|-----------------|-------------------------------|
| 标识符寄存器 n   | CAN_FxR1[15:8]  | CAN_FxR1[7:0]                 |
| 屏蔽位寄存器 n   | CAN_FxR1[31:24] | CAN_FxR1[23:16]               |
| 标识符寄存器 n+1 | CAN_FxR2[15:8]  | CAN_FxR2[7:0]                 |
| 屏蔽位寄存器 n+1 | CAN_FxR2[31:24] | CAN_FxR2[23:16]               |
| 映射         | STID[10:3]      | STID[2:0] RTR IDE EXID[17:15] |

16 位标识符列表模式下寄存器定义

|          |                 |                 |
|----------|-----------------|-----------------|
| 标识符寄存器 n | CAN_FxR1[15:8]  | CAN_FxR1[7:0]   |
| 屏蔽位寄存器 n | CAN_FxR1[31:24] | CAN_FxR1[23:16] |

|            |                 |                 |     |     |             |
|------------|-----------------|-----------------|-----|-----|-------------|
| 标识符寄存器 n+1 | CAN_FxR2[15:8]  | CAN_FxR2[7:0]   |     |     |             |
| 屏蔽位寄存器 n+1 | CAN_FxR2[31:24] | CAN_FxR2[23:16] |     |     |             |
| 映射         | STID[10:3]      | STID[2:0]       | RTR | IDE | EXID[17:15] |

报文进入 FIFO 邮箱中，会被主控读取，通常主控可以根据报文标识符来区分报文数据。CAN 控制器对接收 FIFO 中通过不同过滤器筛选的报文，提供了过滤器编号，编号被存放在寄存器 CAN\_RXMDTRx 的 FMI[7:0] 中，编号时不考虑过滤器组是否启用。

当出现某个报文能通过多个过滤器的过滤，则接收邮箱中存放的过滤器编号根据过滤器优先级规则来决定存放哪个过滤器的编号，过滤器优先级规则如下：

- 所有 32 位的过滤器优先级均高于 16 位的过滤器
- 对于同样宽度的过滤器，标识符列表的过滤器优先级高于屏蔽位模式的过滤器
- 宽度和模式都一致的过滤器，编号小的过滤器优先级更高

在接收报文时，先把标识符与 32 位标识符列表模式过滤器进行匹配筛选，没有匹配再与 32 位屏蔽位模式过滤器进行匹配筛选，没有匹配则继续与 16 位标识符列表模式过滤器进行匹配筛选，没有匹配最后与 16 位屏蔽位模式过滤器进行匹配筛选，最后如果都没有匹配则丢弃报文，出现匹配则报文存入接收 FIFO 的邮箱，标识符编号存入寄存器 CAN\_RXMDTRx 的 FMI 中。

CAN 位时序：

按照 CAN 总线的标准，将每一位时间分为四段：分别为同步段、传播时间段、相位缓冲段 1 和相位缓冲段 2。这些段由最小时间单元 Tq 组成。CAN 控制器通过采样来监测 CAN 总线变化，通过帧起始位的边沿进行同步。

CAN 控制器把上述四段重新划分为三段，分别为：

- 同步段 (SS)：也就是 CAN 标准里的同步段，固定为 1 个最小时间单元，正常情况下所期望的位跳变发生在本时间段内。
- 时间段 1 (BS1)：包含 CAN 标准里的传播时间段和相位缓冲段 1，可以被设置为包含 1 到 16 最小时间单元，可以被自动延长，用于补偿 CAN 总线上不同节点频率精度误差带来的相位正向漂移。该时间段结束为采样点位置。
- 时间段 2 (BS2)：也就是 CAN 标准里的相位缓冲段 2，可以被设置为 1 到 8 个最小时间单元，可以被自动缩短，以补偿 CAN 总线上不同节点频率精度误差带来的相位负向漂移。

重新同步跳转宽度 (SJW)，是每位中可以延长和缩小的最小时间单元数量上限，范围可设置为 1 到 4 个最小时间单元。

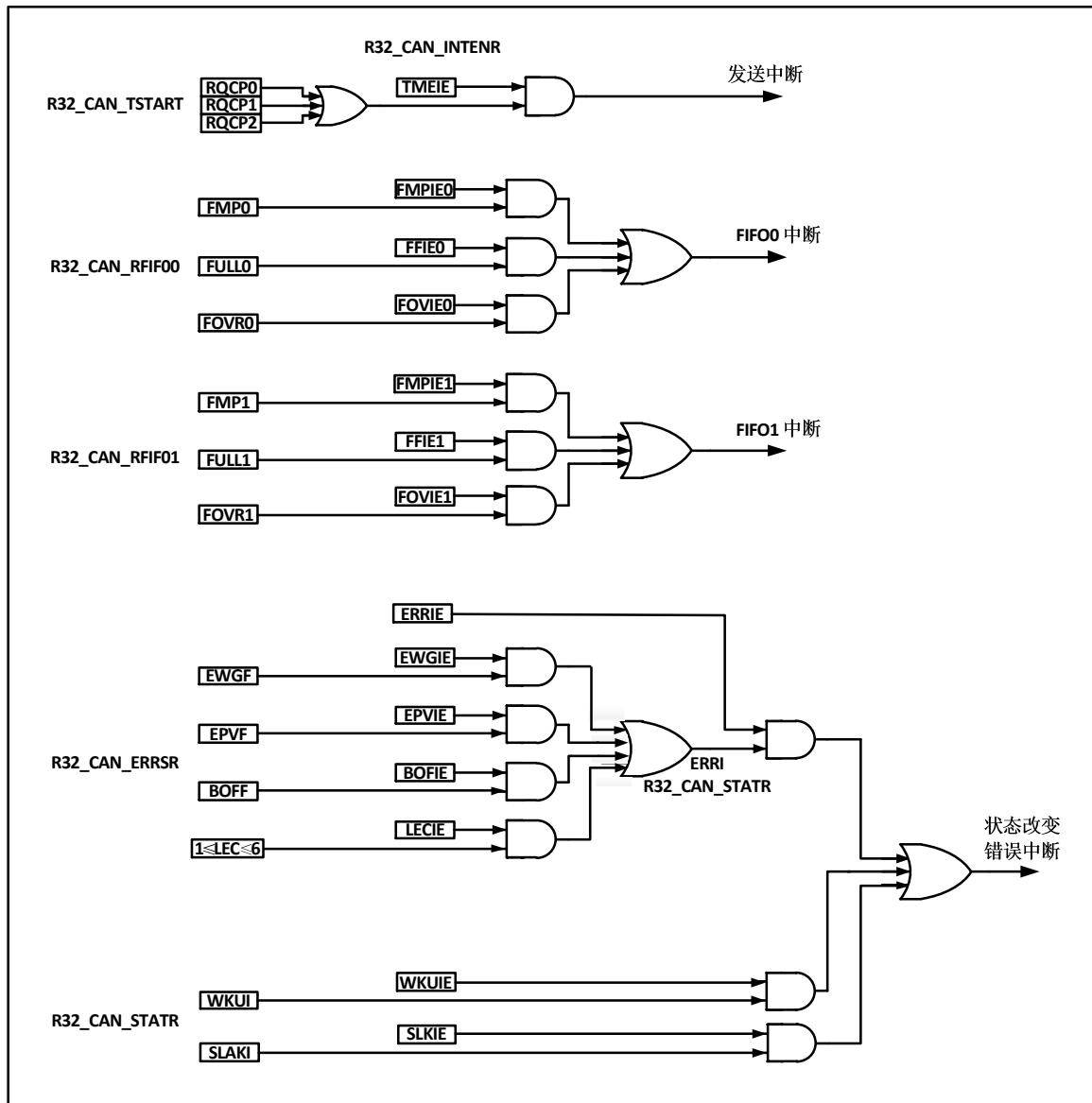
CAN 总线波特率由外设时钟  $f_{CAN}$ 、系统时钟  $f_{clk}$  (96MHz)、分频系数 BRP、时间段 BS1 和 BS2 配置决定，计算公式为：

$$f_{CAN} = \frac{f_{clk}}{(BRP+1) \times (1+BS1+BS2)}$$

上述参数都可以在 CAN 总线时序寄存器 CAN\_BTIMR 里配置。

CAN 中断：CAN 中断支持接收中断、发送中断、错误中断和状态中断这四种类型中断，芯片支持对不同的中断源设置开启或者关闭。

## 芯片中断逻辑示意图



注: SPI 接口操作时会打断 CAN 控制器将数据写入缓存中, 所以 SPI 操作尽量不要占用很多时间。

表 8-1 CAN 寄存器列表

| 访问地址 | 名称           | 功能描述                  |
|------|--------------|-----------------------|
| 00h  | CAN_CTLR     | CAN 主控制寄存器            |
| 01h  | CAN_STATR    | CAN 主状态寄存器            |
| 02h  | CAN_TSTATR   | CAN 发送状态寄存器           |
| 03h  | CAN_RFIFO0   | CAN 接收 FIFO0 控制和状态寄存器 |
| 04h  | CAN_RFIFO1   | CAN 接收 FIFO1 控制和状态寄存器 |
| 05h  | CAN_INTENR   | CAN 中断使能寄存器           |
| 06h  | CAN_ERRSR    | CAN 错误状态寄存器           |
| 07h  | CAN_BTIMR    | CAN 位时序寄存器            |
| 08h  | CAN_TTCCTLR  | CAN 时间触发控制寄存器         |
| 09h  | CAN_TTCNT    | CAN 时间触发计数值寄存器        |
| 0Ah  | CAN_TERR_CNT | CAN 离线恢复错误计数器         |

|     |             |                            |
|-----|-------------|----------------------------|
| 0Bh | CAN_TXMIR0  | CAN 发送邮箱 0 标识符寄存器          |
| 0Ch | CAN_TXMDTR0 | CAN 发送邮箱 0 数据长度和时间戳寄存器     |
| 0Dh | CAN_TXMDLR0 | CAN 发送邮箱 0 低字节数据寄存器        |
| 0Eh | CAN_TXMDHR0 | CAN 发送邮箱 0 高字节数据寄存器        |
| 0Fh | CAN_TXMIR1  | CAN 发送邮箱 1 标识符寄存器          |
| 10h | CAN_TXMDTR1 | CAN 发送邮箱 1 数据长度和时间戳寄存器     |
| 11h | CAN_TXMDLR1 | CAN 发送邮箱 1 低字节数据寄存器        |
| 12h | CAN_TXMDHR1 | CAN 发送邮箱 1 高字节数据寄存器        |
| 13h | CAN_TXMIR2  | CAN 发送邮箱 2 标识符寄存器          |
| 14h | CAN_TXMDTR2 | CAN 发送邮箱 2 数据长度和时间戳寄存器     |
| 15h | CAN_TXMDLR2 | CAN 发送邮箱 2 低字节数据寄存器        |
| 16h | CAN_TXMDHR2 | CAN 发送邮箱 2 高字节数据寄存器        |
| 17h | CAN_RXMIR0  | CAN 接收 FIFO0 邮箱标识符寄存器      |
| 18h | CAN_RXMDTR0 | CAN 接收 FIFO0 邮箱数据长度和时间戳寄存器 |
| 19h | CAN_RXMDLR0 | CAN 接收 FIFO0 邮箱低字节数据寄存器    |
| 1Ah | CAN_RXMDHR0 | CAN 接收 FIFO0 邮箱高字节数据寄存器    |
| 0Bh | CAN_RXMIR1  | CAN 接收 FIFO1 邮箱标识符寄存器      |
| 1Ch | CAN_RXMDTR1 | CAN 接收 FIFO1 邮箱数据长度和时间戳寄存器 |
| 1Dh | CAN_RXMDLR1 | CAN 接收 FIFO1 邮箱低字节数据寄存器    |
| 1Eh | CAN_RXMDHR1 | CAN 接收 FIFO1 邮箱高字节数据寄存器    |
| 1Fh | CAN_FCTLR   | CAN 过滤器主控制寄存器              |
| 20h | CAN_FMCFGR  | CAN 过滤器模式寄存器               |
| 21h | CAN_FSCFGR  | CAN 过滤器位宽寄存器               |
| 22h | CAN_FAFF1OR | CAN 过滤器 FIFO 关联寄存器         |
| 23h | CAN_FWR     | CAN 过滤器激活寄存器               |
| 24h | CAN_FOR1    | CAN 过滤器组 0 寄存器 1           |
| 25h | CAN_FOR2    | CAN 过滤器组 0 寄存器 2           |
| 26h | CAN_F1R1    | CAN 过滤器组 1 寄存器 1           |
| 27h | CAN_F1R2    | CAN 过滤器组 1 寄存器 2           |
| 28h | CAN_F2R1    | CAN 过滤器组 2 寄存器 1           |
| 29h | CAN_F2R2    | CAN 过滤器组 2 寄存器 2           |
| 2Ah | CAN_F3R1    | CAN 过滤器组 3 寄存器 1           |
| 2Bh | CAN_F3R2    | CAN 过滤器组 3 寄存器 2           |
| 2Ch | CAN_F4R1    | CAN 过滤器组 4 寄存器 1           |
| 2Dh | CAN_F4R2    | CAN 过滤器组 4 寄存器 2           |
| 2Eh | CAN_F5R1    | CAN 过滤器组 5 寄存器 1           |
| 2Fh | CAN_F5R2    | CAN 过滤器组 5 寄存器 2           |
| 30h | CAN_F6R1    | CAN 过滤器组 6 寄存器 1           |
| 31h | CAN_F6R2    | CAN 过滤器组 6 寄存器 2           |
| 32h | CAN_F7R1    | CAN 过滤器组 7 寄存器 1           |
| 33h | CAN_F7R2    | CAN 过滤器组 7 寄存器 2           |
| 34h | CAN_F8R1    | CAN 过滤器组 8 寄存器 1           |
| 35h | CAN_F8R2    | CAN 过滤器组 8 寄存器 2           |
| 36h | CAN_F9R1    | CAN 过滤器组 9 寄存器 1           |

|     |              |                   |
|-----|--------------|-------------------|
| 37h | CAN_F9R2     | CAN 过滤器组 9 寄存器 2  |
| 38h | CAN_F10R1    | CAN 过滤器组 10 寄存器 1 |
| 39h | CAN_F10R2    | CAN 过滤器组 10 寄存器 2 |
| 3Ah | CAN_F11R1    | CAN 过滤器组 11 寄存器 1 |
| 3Bh | CAN_F11R2    | CAN 过滤器组 11 寄存器 2 |
| 3Ch | CAN_F12R1    | CAN 过滤器组 12 寄存器 1 |
| 3Dh | CAN_F12R2    | CAN 过滤器组 12 寄存器 2 |
| 3Eh | CAN_F13R1    | CAN 过滤器组 13 寄存器 1 |
| 3Fh | CAN_F13R2    | CAN 过滤器组 13 寄存器 2 |
| 40h | CAN_RX0READ  | CAN 接收邮箱 0 寄存器连续读 |
| 41h | CAN_RX1READ  | CAN 接收邮箱 1 寄存器连续读 |
| 42h | CAN_TX0WRITE | CAN 发送邮箱 0 寄存器连续写 |
| 43h | CAN_TX1WRITE | CAN 发送邮箱 1 寄存器连续写 |
| 44h | CAN_TX2WRITE | CAN 发送邮箱 2 寄存器连续写 |

### 8. 10. 1 CAN 主控制寄存器 (CAN\_CTLR)

| 位       | 名称       | 访问  | 描述   | 复位值 |
|---------|----------|-----|--|-----|
| [31:17] | Reserved | RO  | 保留。  | 0   |
| 16      | DBF      | RW  | 保留, 禁止修改。  | 1   |
| 15      | RST      | RW1 | CAN 控制器软件复位请求, 该位写 0 无效<br>1: 对 CAN 控制器进行复位, 复位后控制器进入睡眠模式, 然后硬件自动清 0;<br>0: CAN 控制器正常状态。                                     | 0   |
| [14:8]  | Reserved | RO  | 保留   | 0   |
| 7       | TTCM     | RW  | 是否允许时间触发模式<br>1: 使能时间触发模式;<br>0: 禁止时间触发模式。<br>时间触发模式主要是配合 TTCAN 协议使用。  | 0   |
| 6       | ABOM     | RW  | 离线自动退出控制<br>1: 硬件检测到 128 次连续 11 个隐性位, 自动退出离线状态;<br>0: 需要软件操作寄存器 CAN_CTLR 的 INRQ 位置 1 然后清 0, 当检测到 128 次连续 11 个隐性位后, 退出离线状态。   | 0   |
| 5       | AWUM     | RW  | CAN 控制器自动唤醒使能<br>1: 当检测到报文时, 硬件自动唤醒, 寄存器 CAN_STATR 的 SLEEP 和 SLAK 位自动清 0;<br>0: 需要软件操作寄存器 CAN_CTLR 的 SLEEP 位清 0, 唤醒 CAN 控制器。 | 0   |
| 4       | NART     | RW  | 报文自动重传功能禁止<br>1: 无论发送成功与否, 报文只能被发送一次;<br>0: CAN 控制器一直重传至发送成功为止。  | 0   |
| 3       | RFLM     | RW  | 接收 FIFO 报文锁定模式使能<br>1: 当接收 FIFO 溢出时, 已接收邮箱报文未读   | 0   |

|   |       |    |  |   |
|---|-------|----|--|---|
|   |       |    | 出，邮箱未释放时，新接收到的报文被丢弃；<br>0：当接收 FIFO 溢出时，已接收邮箱报文未读出，邮箱未释放时，新接收到的报文会覆盖原有报文。   |   |
| 2 | TXFP  | RW | 发送邮箱优先级方式选择<br>1：优先级由发送请求的先后顺序决定；<br>0：优先级由报文标识符来决定。   | 0 |
| 1 | SLEEP | RW | 睡眠模式请求位<br>1：置 1 请求 CAN 控制器进入睡眠模式，当前活动完成后，控制器进入睡眠模式，若 AWUM 位置 1，则在接收到报文时，控制器把 SLEEP 位清 0；<br>0：软件清 0 后，控制器退出睡眠模式。                                    | 1 |
| 0 | INRQ  | RW | 初始化模式请求位<br>1：置 1 请求 CAN 控制器进入初始化模式，当前活动完成后，控制器进入初始化模式，硬件对寄存器 CAN_STATR 的 INAK 位置 1；<br>0：置 0 请求 CAN 控制器退出初始化模式，进入正常模式，硬件对寄存器 CAN_STATR 的 INAK 位清 0。 | 0 |

### 8.10.2 CAN 主状态寄存器 (CAN\_STATR)

| 位       | 名称       | 访问  | 描述  | 复位值 |
|---------|----------|-----|---|-----|
| [31:12] | Reserved | RO  | 保留。   | 0   |
| 11      | RX       | RO  | CAN 控制器接收引脚 RX 当前实际电平。  | 1   |
| 10      | SAMP     | RO  | CAN 控制器接收引脚 RX 上一个接收位的电平  | 1   |
| 9       | RXM      | RO  | 接收模式查询位<br>1：当前 CAN 控制器为接收模式；<br>0：当前 CAN 控制器非接收模式。   | 0   |
| 8       | TXM      | RO  | 发送模式查询位<br>1：当前 CAN 控制器为发送模式；<br>0：当前 CAN 控制器非发送模式。   | 0   |
| [7:5]   | Reserved | RO  | 保留  | 0   |
| 4       | SLAKI    | RW1 | 睡眠中断使能时，即寄存器 CAN_INTENR 的 SLKIE 位置 1 时，中断产生标志位，写 1 清 0，写 0 无效。<br>1：进入睡眠模式时，中断产生，硬件置 1；<br>0：退出睡眠模式时，硬件清 0 也可软件清 0。 | 0   |
| 3       | WKUI     | RW1 | 唤醒中断标志位。当寄存器 CAN_INTENR 的 WKUI 位置 1 时，若 CAN 控制器处于睡眠模式时，检测到 SOF 位，则硬件置 1。软件置 1 清 0，置 0 无效。                           | 0   |
| 2       | ERRI     | RW1 | 出错中断状态标志位。当寄存器 CAN_INTENR 的 ERRIE 位置 1 时，产生错误及状态变化中断。该位软件置 1 清 0，置 0 无效。  | 0   |

|   |      |    |   |   |
|---|------|----|---|---|
| 1 | SLAK | RO | 睡眠模式指示位。<br>1: CAN 控制器正处于睡眠模式；<br>0: CAN 控制器不在睡眠模式。     | 1 |
| 0 | INAK | RO | 初始化模式指示位。<br>1: CAN 控制器正在初始化模式；<br>0: CAN 控制器工作在非初始化模式。 | 0 |

## 8.10.3 CAN 发送状态寄存器 (CAN\_TSTATR)

| 位       | 名称        | 访问  | 描述   | 复位值 |
|---------|-----------|-----|--|-----|
| 31      | LOW2      | RO  | 表示发送邮箱 2 的最低优先级标志位<br>1: 表示发送邮箱 2 的优先级最低；<br>0: 表示发送邮箱 2 的优先级非最低。    | 0   |
| 30      | LOW1      | RO  | 发送邮箱 1 的最低优先级标志位<br>1: 表示发送邮箱 1 的优先级最低；<br>0: 表示发送邮箱 1 的优先级非最低。      | 0   |
| 29      | LOW0      | RO  | 发送邮箱 0 的最低优先级标志位<br>1: 表示发送邮箱 0 的优先级最低；<br>0: 表示发送邮箱 0 的优先级非最低。      | 0   |
| 28      | TME2      | RO  | 表示发送邮箱 2 的空标志位<br>1: 表示发送邮箱 2 无等待发送报文；<br>0: 表示发送邮箱 2 有等待发送报文。       | 1   |
| 27      | TME1      | RO  | 表示发送邮箱 1 的空标志位<br>1: 表示发送邮箱 1 无等待发送报文；<br>0: 表示发送邮箱 1 有等待发送报文。       | 1   |
| 26      | TME0      | RO  | 表示发送邮箱 0 的空标志位<br>1: 表示发送邮箱 0 无等待发送报文；<br>0: 表示发送邮箱 0 有等待发送报文。       | 1   |
| [25:24] | CODE[1:0] | RO  | 邮箱编号<br>当有 1 个以上邮箱为空时，表示下一个为空的邮箱号；当邮箱全空时，表示优先级最低的邮箱号。                | 0   |
| 23      | ABRQ2     | RW1 | 发送邮箱 2 的发送中止请求。软件置 1，可以中止邮箱 2 的发送请求，发送报文被清除时硬件清 0，若邮箱 2 清空，软件置 1 无效。 | 0   |
| [22:20] | Reserved  | RO  | 保留   | 0   |
| 19      | TERR2     | RW1 | 发送邮箱 2 发送失败标志位，当发送邮箱 2 发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。              | 0   |
| 18      | ALST2     | RW1 | 发送邮箱 2 仲裁失败标志位，当发送邮箱 2 仲裁优先级低导致发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。      | 0   |
| 17      | TXOK2     | RW1 | 发送邮箱 2 发送成功标志位<br>1: 上次发送成功；<br>0: 上次发送失败。<br>软件置 1 清 0，软件写 0 无效。    | 0   |

|         |          |     |   |   |
|---------|----------|-----|---|---|
| 16      | RQCP2    | RW1 | 发送邮箱 2 请求完成标志位，当发送邮箱 2 的发送或中止请求完成时，该位自动置 1。软件置 1 清 0，软件写 0 无效。      | 0 |
| 15      | ABRQ1    | RW1 | 发送邮箱 1 的发送中止请求。软件置 1，可以中止邮箱 1 的发送请求，发送报文被清除时硬件清 0。软件写 0 无效。         | 0 |
| [14:12] | Reserved | RO  | 保留  | 0 |
| 11      | TERR1    | RW1 | 发送邮箱 1 发送失败标志位，当发送邮箱 1 发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。             | 0 |
| 10      | ALST1    | RW1 | 发送邮箱 1 仲裁失败标志位，当发送邮箱 1 仲裁优先级低导致发送失败，该位自动置 1。                        | 0 |
| 9       | TXOK1    | RW1 | 发送邮箱 1 发送成功标志位<br>1：上次发送成功；<br>0：上次发送失败。<br>软件置 1 清 0，软件写 0 无效。     | 0 |
| 8       | RQCP1    | RW1 | 发送邮箱 1 请求完成标志位，当发送邮箱 1 的发送或中止请求完成时，该位自动置 1。<br>软件置 1 清 0，软件写 0 无效。  | 0 |
| 7       | ABRQ0    | RW1 | 发送邮箱 0 的发送中止请求。软件置 1，可以中止邮箱 0 的发送请求，发送报文被清除时硬件清 0。软件写 0 无效。         | 0 |
| [6:4]   | Reserved | RO  | 保留  | 0 |
| 3       | TERR0    | RW1 | 发送邮箱 0 发送失败标志位，当发送邮箱 0 发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。             | 0 |
| 2       | ALST0    | RW1 | 发送邮箱 0 仲裁失败标志位，当发送邮箱 0 仲裁优先级低导致发送失败，该位自动置 1。<br>软件置 1 清 0，软件写 0 无效。 | 0 |
| 1       | TXOK0    | RW1 | 发送邮箱 0 发送成功标志位<br>1：上次发送成功；<br>0：上次发送失败。<br>软件置 1 清 0，软件写 0 无效。     | 0 |
| 0       | RQCPO    | RW1 | 发送邮箱 0 请求完成标志位，当发送邮箱 0 的发送或中止请求完成时，该位自动置 1。<br>软件置 1 清 0，软件写 0 无效。  | 0 |

#### 8.10.4 CAN 接收 FIFO 0 状态寄存器 (CAN\_RFIFO0)

| 位       | 名称       | 访问  | 描述   | 复位值 |
|---------|----------|-----|--|-----|
| [31:19] | Reserved | RO  | 保留。  | 0   |
| 18      | RFOM0    | RW1 | 软件对该位置 1，则释放接收 FIFO_0 的当前邮箱报文，释放完后自动清 0，软件写 0 无效。    | 0   |
| 17      | F0VR0    | RW1 | 接收 FIFO_0 溢出标志位。当 FIFO_0 中有 18 个报文时，又接到新报文，硬件置 1。该位需 | 0   |

|        |           |     |  |   |
|--------|-----------|-----|--|---|
|        |           |     | 要软件置 1 清 0，软件写 0 无效。   |   |
| 16     | FULL0     | RW1 | 接收 FIFO_0 满标志位。当 FIFO_0 中有 18 个报文时，硬件置 1。该位需要软件置 1 清 0，软件写 0 无效。 | 0 |
| [15:8] | Reserved  | RO  | 保留。  | 0 |
| [7:0]  | FMP0[7:0] | RO  | 接收 FIFO_0 报文数目。  | 0 |

#### 8.10.5 CAN 接收 FIFO 1 状态寄存器 (CAN\_RFIFO1)

| 位       | 名称        | 访问  | 描述   | 复位值 |
|---------|-----------|-----|--|-----|
| [31:19] | Reserved  | RO  | 保留。  | 0   |
| 18      | RF0M1     | RW1 | 软件对该位置 1，则释放接收 FIFO_1 的当前邮箱报文，释放完后自动清 0，软件写 0 无效。                        | 0   |
| 17      | FOVR1     | RW1 | 接收 FIFO_1 溢出标志位。当 FIFO_1 中有 18 个报文时，又接到新报文，硬件置 1。该位需要软件置 1 清 0，软件写 0 无效。 | 0   |
| 16      | FULL1     | RW1 | 接收 FIFO_1 满标志位。当 FIFO_1 中有 18 个报文时，硬件置 1。该位需要软件置 1 清 0，软件写 0 无效。         | 0   |
| [15:8]  | Reserved  | RO  | 保留。  | 0   |
| [7:0]   | FMP1[7:0] | RO  | 接收 FIFO_1 报文数目。  | 0   |

#### 8.10.6 CAN 中断使能寄存器 (CAN\_INTENR)

| 位       | 名称       | 访问 | 描述   | 复位值 |
|---------|----------|----|--|-----|
| [31:18] | Reserved | RO | 保留。  | 0   |
| 17      | SLKIE    | RW | 睡眠中断使能位。<br>1：进入睡眠状态时，产生中断；<br>0：进入睡眠状态时，不产生中断。  | 0   |
| 16      | WKUIE    | RW | 唤醒中断使能位。<br>1：当 CAN 控制器被唤醒时，产生中断；<br>0：当 CAN 控制器被唤醒时，不产生中断。  | 0   |
| 15      | ERRIE    | RW | 错误中断使能位，CAN 错误中断总使能位。<br>1：当 CAN 控制器产生错误时，产生中断；<br>0：当 CAN 控制器产生错误时，不产生中断。                               | 0   |
| [14:12] | Reserved | RF | 保留。  | 0   |
| 11      | LECIE    | RW | 上次错误号中断使能位。<br>1：检测到错误时，硬件更新 LEC[2:0]，更新 ERRI 位为 1，触发错误中断；<br>0：检测到错误时，硬件更新 LEC[2:0]，不更新 ERRI 位，不触发错误中断。 | 0   |
| 10      | BOFIE    | RW | 离线中断使能位。<br>1：进入离线状态时，更新 ERRI 位为 1，触发错误中断；<br>0：进入离线状态时，不更新 ERRI 位，不触                                    | 0   |

|   |          |    |   |   |
|---|----------|----|---|---|
|   |          |    | 发错误中断。  |   |
| 9 | EPVIE    | RW | 错误被动中断使能位。<br>1: 进入错误被动状态时, 更新 ERRI 位为 1, 触发错误中断;<br>0: 进入错误被动状态时, 不更新 ERRI 位, 不触发错误中断。                           | 0 |
| 8 | EWGIE    | RW | 错误警告中断使能位。<br>1: 出错次数达到警告阈值时, 更新 ERRI 位为 1, 触发错误中断;<br>0: 出错次数达到警告阈值时, 不更新 ERRI 位, 不触发错误中断。                       | 0 |
| 7 | Reserved | RF | 保留。   | 0 |
| 6 | FOVIE1   | RW | 接收 FIFO_1 溢出中断使能位。<br>1: 当 FIFO_1 溢出, 触发 FIFO_1 中断;<br>0: 当 FIFO_1 溢出, 不触发 FIFO_1 中断。                             | 0 |
| 5 | FFIE1    | RW | 接收 FIFO_1 满中断使能位。<br>1: 当 FIFO_1 为满, 触发 FIFO_1 中断;<br>0: 当 FIFO_1 为满, 不触发 FIFO_1 中断。                              | 0 |
| 4 | FMPIE1   | RW | 接收 FIFO_1 消息挂号中断使能位。<br>1: 当 FIFO_1 更新 FMP 位, 且不为 0, 触发 FIFO_1 中断;<br>0: 当 FIFO_1 更新 FMP 位, 且不为 0, 不触发 FIFO_1 中断。 | 0 |
| 3 | FOVIE0   | RW | 接收 FIFO_0 溢出中断使能位。<br>1: 当 FIFO_0 溢出, 触发 FIFO_0 中断;<br>0: 当 FIFO_0 溢出, 不触发 FIFO_0 中断。                             | 0 |
| 2 | FFIE0    | RW | 接收 FIFO_0 满中断使能位。<br>1: 当 FIFO_0 为满, 触发 FIFO_0 中断;<br>0: 当 FIFO_0 为满, 不触发 FIFO_0 中断。                              | 0 |
| 1 | FMPIE0   | RW | 接收 FIFO_0 消息挂号中断使能位。<br>1: 当 FIFO_0 更新 FMP 位, 且不为 0, 触发 FIFO_0 中断;<br>0: 当 FIFO_0 更新 FMP 位, 且不为 0, 不触发 FIFO_0 中断。 | 0 |
| 0 | TMEIE    | RW | 发送邮箱空中断。<br>1: 当发送邮箱为空时, 产生中断;<br>0: 当发送邮箱为空时, 不产生中断。   | 0 |

### 8.10.7 CAN 错误状态寄存器 (CAN\_ERRSR)

| 位       | 名称       | 访问 | 描述  | 复位值 |
|---------|----------|----|---|-----|
| [31:24] | REC[7:0] | RO | 接收错误计数器。<br>当 CAN 接收出错时, 根据出错条件, 该计数器加 1 或 8; 接收成功后, 该计数器减 1 或设为 120(错误计数值大于 127)。计数器值超过 127 时, CAN 进入错误被动状态。 | 0   |

|         |          |    |  |   |
|---------|----------|----|--|---|
| [23:16] | TEC[7:0] | R0 | 发送错误计数器。<br>当 CAN 发送出错时，根据出错条件，该计数器加 1 或 8；发送成功后，该计数器减 1 或设为 120(错误计数值大于 127)。计数器值超过 127 时，CAN 进入错误被动状态。   | 0 |
| [15:7]  | Reserved | R0 | 保留。  | 0 |
| [6:4]   | LEC{2:0} | RW | 上次错误代号。<br>检测到 CAN 总线上发送错误时，控制器根据出错情况设置，当正确收发报文时，置 000b。<br>000：无错误；<br>001：位填充错误；<br>010：FORM 格式错误；<br>011：ACK 确认错误；<br>100：隐性位错误；<br>101：显性位错误；<br>110：CRC 错误；<br>111：软件设置。<br>通常应用软件读取到错误时，把代号设置为 111b，可以检测到代号更新。 | 0 |
| 3       | Reserved | R0 | 保留。  | 0 |
| 2       | B0FF     | R0 | 离线状态标志位。<br>当 CAN 控制器进入离线状态时，硬件自动置 1；退出离线状态时，硬件自动清 0。  | 0 |
| 1       | EPVF     | R0 | 错误被动标志位。<br>当收发错误计数器达到错误被动阈值时，即大于 127 时，硬件置 1。   | 0 |
| 0       | EWGF     | R0 | 错误警告标志位。<br>当收发错误计数器达到警告阈值时，即大于等于 96 时，硬件置 1。  | 0 |

#### 8.10.8 CAN 位时序寄存器 (CAN\_BTIMR)

| 位       | 名称       | 访问 | 描述  | 复位值   |
|---------|----------|----|---|-------|
| 31      | SILM     | RW | 静默模式设置位。<br>1：进入静默模式；<br>0：退出静默模式。  | 0     |
| 30      | LBKM     | RW | 环回模式设置位。<br>1：进入环回模式；<br>0：退出环回模式。  | 0     |
| [29:26] | Reserved | R0 | 保留，禁止修改。  | 0     |
| [25:24] | SJW[1:0] | RW | 定义了重新同步跳转宽度设置值。<br>实现重新同步时，位中可以延长和缩小的最长时间单元数量上限，实际值为 (SJW[1:0]+1)，范围可设置为 1 到 4 个最长时间单元。 | 0001b |
| 23      | Reserved | R0 | 保留，禁止修改。  |       |

|         |          |    |  |       |
|---------|----------|----|--|-------|
| [22:20] | TS2[2:0] | RW | 时间段 2 设置值。<br>定义了时间段 2 占用了多少个最小时间单元，实际值为 (TS2[2:0]+1)。 | 010b  |
| [19:16] | TS1[3:0] | RW | 时间段 1 设置值。<br>定义了时间段 1 占用了多少个最小时间单元，实际值为 (TS1[3:0]+1)。 | 0011b |
| [15:10] | Reserved | RO | 保留。禁止修改。   | 0     |
| [9:0]   | BRP[9:0] | RW | 最小时间单元长度设置值：<br>$T_q = (BRP[9:0]+1) \times t_{clock}$  | 0     |

## 8.10.9 CAN 时间触发控制寄存器 (CAN\_TTCTLR)

| 位       | 名称           | 访问 | 描述                                  | 复位值   |
|---------|--------------|----|-------------------------------------|-------|
| [31:18] | Reserved     | RO | 保留。                                 | 0     |
| 17      | MODE         | RW | 时间触发模式选择位：<br>1: 增强模式；<br>0: 默认模式。  | 0     |
| 16      | TIMRST       | WZ | 内部计数器复位控制位：<br>写 1 复位内部计数器，硬件自动清 0。 | 0     |
| [15:0]  | TIMCMV[15:0] | RW | 内部计数器计数终值。                          | ffffh |

## 8.10.10 CAN 时间触发计数值寄存器 (CAN\_TTCNT)

| 位       | 名称           | 访问 | 描述       | 复位值 |
|---------|--------------|----|----------|-----|
| [31:16] | Reserved     | RO | 保留。      | 0   |
| [15:0]  | TIMCNT[15:0] | RW | 时间触发计数值。 | 0   |

## 8.10.11 CAN 离线恢复错误计数器 (CAN\_TERR\_CNT)

| 位      | 名称         | 访问 | 描述                          | 复位值 |
|--------|------------|----|-----------------------------|-----|
| [31:9] | Reserved   | RO | 保留。                         | 0   |
| [8:0]  | TX_ERR_CNT | RW | 当前离线恢复错误计数值，修改该计数值可从离线立即恢复。 | 0   |

## 8.10.12 CAN 发送邮箱标识符寄存器 (CAN\_TXMIRy) (y = 0/1/2)

| 位       | 名称                         | 访问 | 描述   | 复位值 |
|---------|----------------------------|----|--|-----|
| [31:21] | STID[10:0]<br>/EXID[28:18] | RW | 标准标识符或扩展标识符的高 11 位。                          | x   |
| [20:3]  | EXID[17:0]                 | RW | 扩展标识符的低 18 位。                                | x   |
| 2       | IDE                        | RW | 标识符选择标志位：<br>1: 选用扩展标识符；<br>0: 选用标准标识符。      | x   |
| 1       | RTR                        | RW | 远程帧(也称遥控帧)选择标志位：<br>1: 当前为远程帧；<br>0: 当前为数据帧。 | x   |
| 0       | TXRQ                       | RW | 数据发送请求标志位：<br>软件置 1 时，请求发送邮箱里的数据，发送          | 0   |

|  |  |  |                |  |
|--|--|--|----------------|--|
|  |  |  | 完毕邮箱为空时，硬件清 0。 |  |
|--|--|--|----------------|--|

## 8. 10. 13 CAN 发送邮箱数据长度和时间戳寄存器 (CAN\_TXMDTRy) (y = 0/1/2)

| 位       | 名称         | 访问 | 描述  | 复位值 |
|---------|------------|----|---|-----|
| [31:16] | TIME[15:0] | RW | 用于发送报文 SOF 时刻的 16 位定时器值。  | x   |
| [15:9]  | Reserved   | RO | 保留。   | 0   |
| 8       | TGT        | RW | 报文时间戳发送选择标志位。该位在 TTCM 置 1，并报文长度为 8 时有效：<br>1：发送时间戳，值为 TIME[15:0] 的即时值，替换了 8 字节报文的最后两个字节；<br>0：不发送时间戳。 | x   |
| [7:4]   | Reserved   | RO | 保留。   | 0   |
| [3:0]   | DLC[3:0]   | RW | 数据帧的数据长度或远程帧请求数据长度，数据长度可设置范围为 0 到 8。  | 0   |

## 8. 10. 14 CAN 发送邮箱低字节数据寄存器 (CAN\_TXMDLRy) (y = 0/1/2)

| 位       | 名称         | 访问 | 描述            | 复位值 |
|---------|------------|----|---------------|-----|
| [31:24] | DATA3[7:0] | RW | 发送数据字节 3 的内容。 | x   |
| [23:16] | DATA2[7:0] | RW | 发送数据字节 2 的内容。 | x   |
| [15:8]  | DATA1[7:0] | RW | 发送数据字节 1 的内容。 | x   |
| [7:0]   | DATA0[7:0] | RW | 发送数据字节 0 的内容。 | x   |

## 8. 10. 15 CAN 发送邮箱高字节数据寄存器 (CAN\_TXMDHRy) (y = 0/1/2)

| 位       | 名称         | 访问 | 描述            | 复位值 |
|---------|------------|----|---------------|-----|
| [31:24] | DATA7[7:0] | RW | 发送数据字节 7 的内容。 | x   |
| [23:16] | DATA6[7:0] | RW | 发送数据字节 6 的内容。 | x   |
| [15:8]  | DATA5[7:0] | RW | 发送数据字节 5 的内容。 | x   |
| [7:0]   | DATA4[7:0] | RW | 发送数据字节 4 的内容。 | x   |

## 8. 10. 16 CAN 接收邮箱标识符寄存器 (CAN\_RXMIRy) (y = 0/1)

| 位       | 名称                          | 访问 | 描述   | 复位值 |
|---------|-----------------------------|----|--|-----|
| [31:21] | STID[10:0]<br>/EXIDH[28:18] | RO | 标准标识符或扩展标识符的高 11 位。                        | x   |
| [20:3]  | EXIDL[17:0]                 | RO | 扩展标识符的低 18 位。                              | x   |
| 2       | IDE                         | RO | 标识符选择标志位：<br>1：选用扩展标识符；<br>0：选用标准标识符。      | x   |
| 1       | RTR                         | RO | 远程帧(也称遥控帧)选择标志位：<br>1：当前为远程帧；<br>0：当前为数据帧。 | x   |
| 0       | Reserved                    | RO | 保留。  | 0   |

## 8. 10. 17 CAN 接收邮箱数据长度和时间戳寄存器 (CAN\_RXMDTRy) (y = 0/1)

| 位       | 名称         | 访问 | 描述                                     | 复位值 |
|---------|------------|----|--|-----|
| [31:16] | TIME[15:0] | R0 | 用于接收报文 SOF 时刻的 16 位定时器值。               | 0   |
| [15:8]  | FMI[7:0]   | R0 | 报文所匹配的过滤器编号。                           | x   |
| [7:4]   | Reserved   | R0 | 保留。                                    | 0   |
| [3:0]   | DLC[3:0]   | R0 | 接收报文数据长度：<br>0-8：表示数据帧长度 0 到 8，远程帧为 0。 | x   |

## 8. 10. 18 CAN 接收邮箱低字节数据寄存器 (CAN\_RXMDLRy) (y = 0/1)

| 位       | 名称         | 访问 | 描述           | 复位值 |
|---------|------------|----|--------------|-----|
| [31:24] | DATA3[7:0] | R0 | 接收报文的数据字节 3。 | x   |
| [23:16] | DATA2[7:0] | R0 | 接收报文的数据字节 2。 | x   |
| [15:8]  | DATA1[7:0] | R0 | 接收报文的数据字节 1。 | x   |
| [7:0]   | DATA0[7:0] | R0 | 接收报文的数据字节 0。 | x   |

## 8. 10. 19 CAN 接收邮箱高字节数据寄存器 (CAN\_RXMDHRy) (y = 0/1)

| 位       | 名称         | 访问 | 描述           | 复位值 |
|---------|------------|----|--------------|-----|
| [31:24] | DATA7[7:0] | R0 | 接收报文的数据字节 7。 | x   |
| [23:16] | DATA6[7:0] | R0 | 接收报文的数据字节 6。 | x   |
| [15:8]  | DATA5[7:0] | R0 | 接收报文的数据字节 5。 | x   |
| [7:0]   | DATA4[7:0] | R0 | 接收报文的数据字节 4。 | x   |

## 8. 10. 20 CAN 过滤器主控制寄存器 (CAN\_FCTLR)

| 位       | 名称       | 访问 | 描述  | 复位值   |
|---------|----------|----|---|-------|
| [31:16] | Reserved | R0 | 保留。   | 0000h |
| [15:8]  | Reserved | R0 | 保留。   | 0     |
| [7:1]   | Reserved | R0 | 保留。   | 0     |
| 0       | FINIT    | RW | 过滤器初始化模式使能标志位：<br>1：过滤器组为初始化模式；<br>0：过滤器组为正常模式。 | 1     |

## 8. 10. 21 CAN 过滤器模式寄存器 (CAN\_FMCFGR)

| 位       | 名称       | 访问 | 描述   | 复位值 |
|---------|----------|----|--|-----|
| [31:14] | Reserved | R0 | 保留。  | 0   |
| [13:0]  | FBMx     | RW | 过滤器组 x 的工作模式控制位, FINIT 为 1 才能写入：<br>0：过滤器组 x 的寄存器为屏蔽位模式；<br>1：过滤器组 x 的寄存器为标识符列表模式。 | 0   |

## 8. 10. 22 CAN 过滤器位宽寄存器 (CAN\_FSCFGR)

| 位       | 名称       | 访问 | 描述                             | 复位值 |
|---------|----------|----|--------------------------------|-----|
| [31:14] | Reserved | R0 | 保留。                            | 0   |
| [13:0]  | FSCx     | RW | 过滤器组 x 的位宽控制位, FINIT 为 1 才能写入： | 0   |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  | 1: 过滤器组 x 的寄存器为单个 32 位;<br>0: 过滤器组 x 的寄存器为 2 个 16 位。 |  |
|--|--|--|--|--|

### 8.10.23 CAN 过滤器 FIFO 关联寄存器 (CAN\_FAFIFOR)

| 位       | 名称       | 访问 | 描述   | 复位值 |
|---------|----------|----|--|-----|
| [31:14] | Reserved | R0 | 保留。  | 0   |
| [13:0]  | FFAx     | RW | 过滤器组 x 的关联 FIFO 控制位, FINIT 为 1 才能写入:<br>1: 过滤器组 x 被关联到 FIFO_1;<br>0: 过滤器组 x 被关联到 FIFO_0。 | 0   |

### 8.10.24 CAN 过滤器激活寄存器 (CAN\_FWR)

| 位       | 名称       | 访问 | 描述   | 复位值 |
|---------|----------|----|--|-----|
| [31:14] | Reserved | R0 | 保留。  | 0   |
| [13:0]  | FACTx    | RW | 过滤器组 x 的激活控制位:<br>1: 过滤器组 x 激活;<br>0: 过滤器组 x 禁用。 | 0   |

### 8.10.25 CAN 过滤器组的过滤寄存器 (CAN\_FiRx) (i = 0-13, x = 1/2)

| 位      | 名称 | 访问 | 描述   | 复位值 |
|--------|----|----|--|-----|
| [31:0] | FB | RW | 过滤器组中寄存器的标志位, FINIT 为 1 才能写入。<br>标识符模式<br>1: 对应位期望电平为隐性位;<br>0: 对应位期望电平为显性位。<br><br>屏蔽位模式<br>1: 必须和对应的标识符寄存器位一致;<br>0: 不需要和对应的标识符寄存器位一致。 | 0   |

### 8.10.26 CAN 接收邮箱寄存器连续读 (CAN\_RXxREAD) (x = 0/1)

| 读取寄存器       | 内容说明              | 访问 | 描述                  | 复位值 |
|-------------|-------------------|----|---------------------|-----|
| CAN_RXMDHRx | 接收邮箱 x 的数据 0 寄存器值 | R0 | CAN 接收邮箱高字节数据寄存器    | 0   |
| CAN_RXMDLRx |                   | R0 | CAN 接收邮箱低字节数据寄存器    | 0   |
| CAN_RXMDTRx |                   | R0 | CAN 接收邮箱数据长度和时间戳寄存器 | 0   |
| CAN_RXMIRx  |                   | R0 | CAN 接收邮箱标识符寄存器      | 0   |
| CAN_RXMDHRx | 接收邮箱 x 的数据 1 寄存器值 | R0 | CAN 接收邮箱高字节数据寄存器    | 0   |
| CAN_RXMDLRx |                   | R0 | CAN 接收邮箱低字节数据寄存器    | 0   |
| CAN_RXMDTRx |                   | R0 | CAN 接收邮箱数据长度和时间戳寄存器 | 0   |
| CAN_RXMIRx  |                   | R0 | CAN 接收邮箱标识符寄存器      | 0   |
| CAN_RXMDHRx | 接收邮箱 x 的数据 2 寄存器值 | R0 | CAN 接收邮箱高字节数据寄存器    | 0   |
| CAN_RXMDLRx |                   | R0 | CAN 接收邮箱低字节数据寄存器    | 0   |
| CAN_RXMDTRx |                   | R0 | CAN 接收邮箱数据长度和时间戳寄存器 | 0   |
| CAN_RXMIRx  |                   | R0 | CAN 接收邮箱标识符寄存器      | 0   |

|             |                   |    |                     |   |
|-------------|-------------------|----|---------------------|---|
| CAN_RXMDHRx | 接收邮箱 x 的数据 3 寄存器值 | R0 | CAN 接收邮箱高字节数据寄存器    | 0 |
| CAN_RXMDLRx |                   | R0 | CAN 接收邮箱低字节数据寄存器    | 0 |
| CAN_RXMDTRx |                   | R0 | CAN 接收邮箱数据长度和时间戳寄存器 | 0 |
| CAN_RXMIRx  |                   | R0 | CAN 接收邮箱标识符寄存器      | 0 |

接收邮箱连续读固定返回当前 4 个接收 FIFO 报文，如果实际不满 4 个报文，则无数据部分返回为 0，主控使用此命令尽量保证有 4 个以上报文，若不满 4 个接收报文进行读取可以通过空值判断是否有效。使用该寄存器对 FIFO 数据进行读取后，该队列会自动释放，无需操作 CAN\_RFIFO0 或者 CAN\_RFIFO1 清除接收标志。

#### 8.10.27 CAN 发送邮箱寄存器连续写 (CAN\_TXxWRITE) (x = 0/1/2)

| 读取寄存器       | 说明          | 访问 | 描述                  | 复位值 |
|-------------|-------------|----|---------------------|-----|
| CAN_TXMDHRx | 发送邮箱 x 的寄存器 | WO | CAN 发送邮箱高字节数据寄存器    | 0   |
| CAN_TXMDLRx |             | WO | CAN 发送邮箱低字节数据寄存器    | 0   |
| CAN_TXMDTRx |             | WO | CAN 发送邮箱数据长度和时间戳寄存器 | 0   |
| CAN_TXMIRx  |             | WO | CAN 发送邮箱标识符寄存器      | 0   |

发送邮箱连续写可以写入对应邮箱编号 x 的发送寄存器，最后对标识符发送请求位写入后可以启动发送，主控操作连续写前需要确认该邮箱是否为空闲状态。

#### 8.11 IIC 接口

CH9434D 支持 IIC 接口，两线 IIC 总线包含数据线 SDA 和串行时钟线 SCK，引脚为开漏模式需要外接上拉电阻，速率支持 1MHz，7 位地址，地址低两位通过引脚配置 AD1 和 ADO 进行配置，地址高位值为 0x28，即地址值为 0x28|AD1|AD0，上电初始化阶段，芯片根据当前 SEL 引脚选择通讯接口，如果选择 IIC 接口，芯片会初始化 AD1 和 ADO 引脚并记录当前电平状态，地址设置完成后 AD0 和 AD1 引脚会设置成浮空状态。

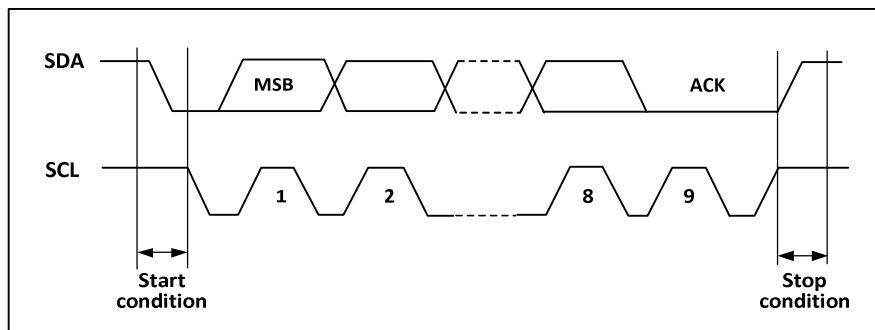
IIC 传输的地址和数据都是高位在前，低位在后，起始事件后第一个字节为地址字节，头 7 位为设备目标地址，第 8 位（下表 Bit0）为传输报文方向，0 表示主设备写入数据到设备，1 表示主设备从从设备读取数据。

地址字节内容图示：

| bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|------|------|------|------|------|------|------|------|
| 0    | 1    | 0    | 1    | 0    | AD1  | AD0  | 0/1  |

IIC 是半双工总线，一个完整的传输中包含起始信号、数据、应答和结束信号。第一个字节为 IIC 地址字节，后续字节为传输协议，芯片根据地址进行匹配，总线上的设备地址必须唯一。

IIC 传输时序图示



IIC 写寄存器：IIC 起始事件后，先写入写方向 IIC 地址字节，随后写入寄存器地址字节，最后写入相应的数据字节，数据写入后发送 IIC 结束事件。

IIC 写寄存器流程图示：

|   |          |   |   |   |   |    |    |   |   |       |   |   |   |   |   |   |   |   |         |  |   |   |
|---|----------|---|---|---|---|----|----|---|---|-------|---|---|---|---|---|---|---|---|---------|--|---|---|
| S | IIC 地址字节 |   |   |   |   |    |    |   | A | 寄存器地址 |   |   |   |   |   |   |   | A | N 字节数据  |  | A | P |
|   | 0        | 1 | 0 | 1 | 0 | A1 | A0 | 0 |   | 1     | x | x | x | x | x | x | x |   | D0-Dn-1 |  |   |   |

写寄存器中 N 字节数据可以是单字节，多个字节、扩展功能地址如 CAN 地址和数据。

IIC 读寄存器：IIC 起始事件后，先写入写方向 IIC 地址字节，随后写入寄存器地址字节，后面可选写入多字节操作地址。再次发送 IIC 起始事件，写入读方向 IIC 地址字节，读取数据，数据读取完成后发送 IIC 结束事件。

IIC 读寄存器流程图示：

|   |          |   |   |   |   |    |    |   |   |           |   |   |   |   |   |   |   |    |         |    |   |
|---|----------|---|---|---|---|----|----|---|---|-----------|---|---|---|---|---|---|---|----|---------|----|---|
| S | IIC 地址字节 |   |   |   |   |    |    |   | A | 寄存器地址     |   |   |   |   |   |   |   | A  | 可选地址    |    | A |
|   | 0        | 1 | 0 | 1 | 0 | A1 | A0 | 0 |   | 0         | x | x | x | x | x | x | x |    | A0-An-1 |    |   |
| S | IIC 地址字节 |   |   |   |   |    |    |   | A | 读取 N 字节数据 |   |   |   |   |   |   |   | NA | P       | NA | P |
|   | 0        | 1 | 0 | 1 | 0 | A1 | A0 | 1 |   | D0-Dn-1   |   |   |   |   |   |   |   |    |         |    |   |

读寄存器中可选地址为扩展功能地址如 CAN 地址，读取的数据可以是单字节或者多字节。

S: IIC 起始信号；

A: IIC 应答信号，字节传输后都需要应答信号，图示省略了多字节中间的应答；

NA: IIC 此字节无应答信号；

P: IIC 结束信号；

A1/A0: 为 IIC 的设置地址即 AD0 和 AD1；

IIC 传输的寄存器地址和数据内容与 SPI 接口一致，数据内容可以参考 SPI 接口。

IIC 传输异常恢复：IIC 传输中防止接口出现异常情况，芯片会监控 IIC 传输流程，当 IIC 传输发生错误或者传输中断超过 40mS，芯片会重置本次传输，并等待下一次起始信号和 IIC 地址。

## 9、参数

9.1 CH9434A 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 |
|-----------------------|------------------------------------|------|----------------------|----|
| T <sub>A</sub>        | 工作时的环境温度                           | -40  | 85                   | °C |
| T <sub>S</sub>        | 存储时的环境温度                           | -40  | 105                  | °C |
| V <sub>CC</sub>       | 电源电压 (V <sub>CC</sub> 接电源, GND 接地) | -0.4 | 3.9                  | V  |
| V <sub>IO</sub>       | 输入或者输出引脚上的电压                       | -0.4 | V <sub>CC</sub> +0.4 | V  |
| V <sub>ESD(HBM)</sub> | 普通 I/O 引脚的 ESD 静电放电电压 (HBM)        | 4K   |                      | V  |

9.2 CH9434D 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 |
|-----------------------|------------------------------------|------|----------------------|----|
| T <sub>A</sub>        | 工作时的环境温度                           | -40  | 85                   | °C |
| T <sub>S</sub>        | 存储时的环境温度                           | -40  | 125                  | °C |
| V <sub>CC</sub>       | 电源电压 (V <sub>CC</sub> 接电源, GND 接地) | -0.3 | 4.0                  | V  |
| V <sub>IN</sub>       | FT (耐受 5V) 引脚上的输入电压                | -0.3 | 5.5                  | V  |
|                       | 其他引脚上的电压                           | -0.3 | V <sub>CC</sub> +0.3 | V  |
| V <sub>ESD(HBM)</sub> | 普通 I/O 引脚的 ESD 静电放电电压 (HBM)        | 4K   |                      | V  |

9.3 CH9434A 电气参数 (测试条件: T<sub>A</sub> = 25°C, V<sub>CC</sub> = 3.3V)

| 名称              | CH9434A 参数说明  | 最小值                  | 典型值                  | 最大值             | 单位 |
|-----------------|---------------|----------------------|----------------------|-----------------|----|
| V <sub>CC</sub> | 系统电源电压        | 2.4                  | 3.3                  | 3.6             | V  |
| I <sub>CC</sub> | 工作时的总电源电流     |                      | 8.9                  |                 | mA |
| V <sub>IL</sub> | 低电平输入电压       | 0                    |                      | 0.9             | V  |
| V <sub>IH</sub> | 高电平输入电压       | 2.0                  |                      | V <sub>CC</sub> | V  |
| V <sub>OL</sub> | 低电平输出电压       | 0                    | 0.3                  | 0.4             | V  |
| V <sub>OH</sub> | 高电平输出电压       | V <sub>CC</sub> -0.4 | V <sub>CC</sub> -0.3 | V <sub>CC</sub> | V  |
| I <sub>DN</sub> | 带下拉电阻输入端的输入电流 | -90                  | -60                  | 25              | uA |
| I <sub>UP</sub> | 带上拉电阻输入端的输入电流 | 25                   | 60                   | 90              | uA |

9.4 CH9434D 电气参数 (测试条件: T<sub>A</sub> = 25°C, V<sub>CC</sub> = 3.3V)

| 名称              | CH9434D 参数说明    | 最小值                  | 典型值 | 最大值  | 单位 |
|-----------------|-----------------|----------------------|-----|------|----|
| V <sub>CC</sub> | 系统电源电压          | 1.8                  | 3.3 | 3.6  | V  |
| I <sub>CC</sub> | 工作时的总电源电流       |                      | 8.1 |      | mA |
| V <sub>IL</sub> | 标准 I/O, 低电平输入电压 | -0.3                 |     | 1.02 | V  |
|                 | FT I/O, 低电平输入电压 | -0.3                 |     | 1.03 | V  |
| V <sub>IH</sub> | 标准 I/O, 高电平输入电压 | 2.0                  |     | 3.6  | V  |
|                 | FT I/O, 高电平输入电压 | 1.83                 |     | 5.5  | V  |
| V <sub>OL</sub> | 低电平输出电压         |                      |     | 0.4  | V  |
| V <sub>OH</sub> | 高电平输出电压         | V <sub>CC</sub> -0.4 |     |      | V  |
| R <sub>PU</sub> | 上拉等效电阻          | 30                   | 40  | 50   | kΩ |
| R <sub>PD</sub> | 下拉等效电阻          | 30                   | 40  | 50   | kΩ |

9.5 CH9434A 低功耗参数 (测试条件:  $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 3.3\text{V}$ )

| 名称        | 参数说明            | 最小值 | 典型值 | 最大值 | 单位 |
|-----------|-----------------|-----|-----|-----|----|
| $I_{CC1}$ | 低功耗模式1: SLEEP睡眠 |     | 4.8 |     | mA |

注: 低功耗模式需注意 I/O 的电平状态和设置, 减少漏电流。

9.6 CH9434D 低功耗参数 (测试条件:  $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 3.3\text{V}$ )

| 名称        | 参数说明            | 最小值 | 典型值  | 最大值 | 单位 |
|-----------|-----------------|-----|------|-----|----|
| $I_{CC1}$ | 低功耗模式1: IDLE睡眠  |     | 4.55 |     | mA |
| $I_{CC2}$ | 低功耗模式2: SLEEP睡眠 |     | 4.5  |     | uA |
| $I_{CC3}$ | 低功耗模式3: SHUT 睡眠 |     | 0.4  |     | uA |

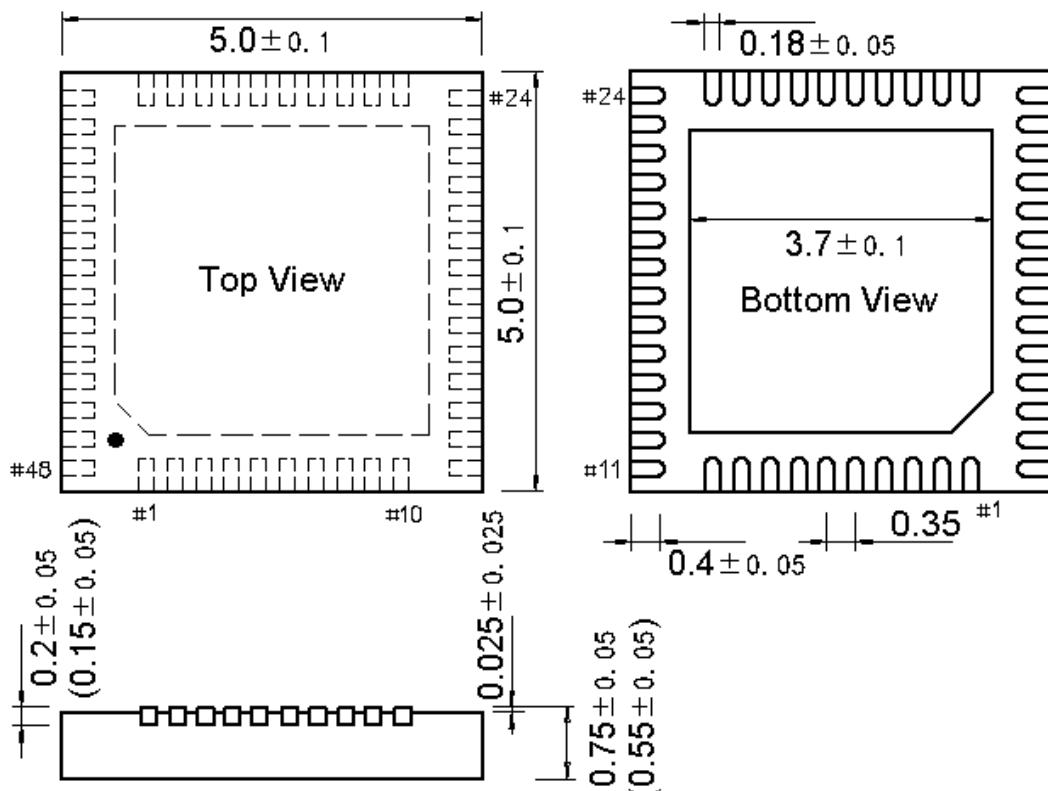
注: 低功耗模式需注意 I/O 的电平状态和设置, 减少漏电流。

## 10、封装信息

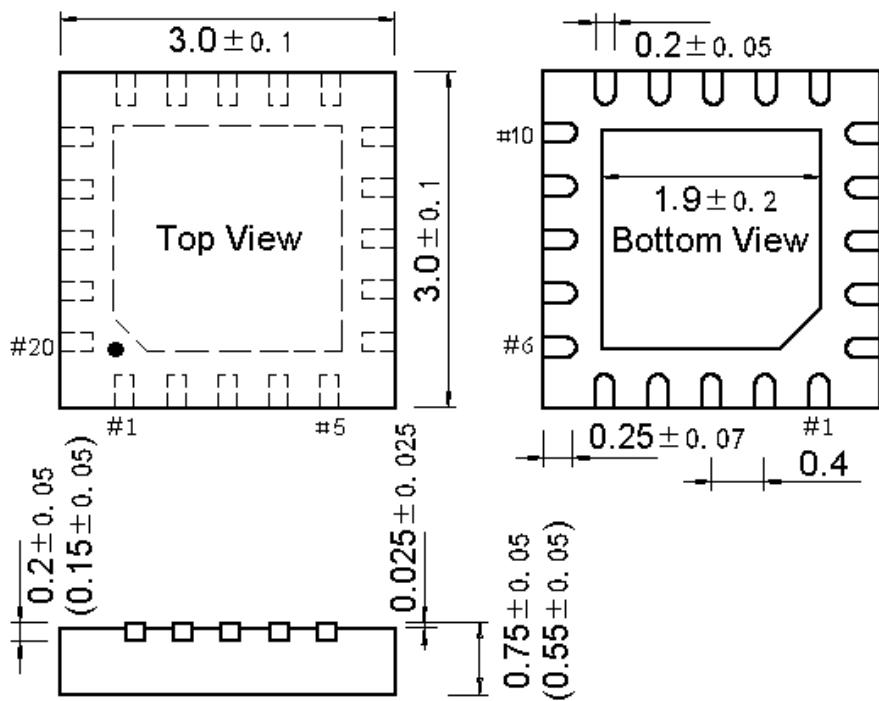
说明：尺寸标注的单位是 mm（毫米）。

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 QFN48



### 10.2 QFN20

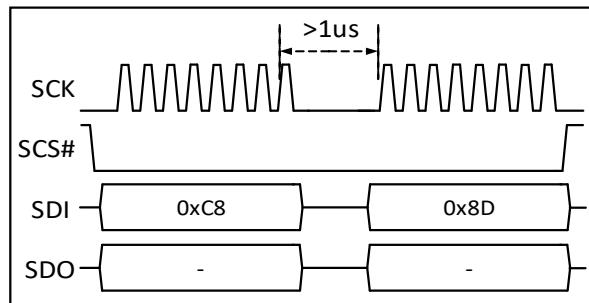




## 11、应用

### 11.1 设置时钟

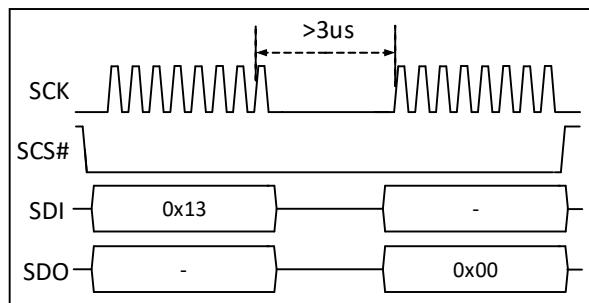
例 1：CH9434A 设置时钟为使用外部晶振，开启时钟倍频，分频系数为默认 13。



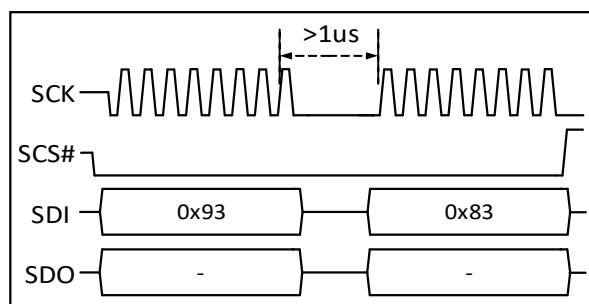
### 11.2 初始化串口

例 1：CH9434A 初始化串口 1 为 115200 波特率，8Bit 数据位，1Bit 停止位，无校验。

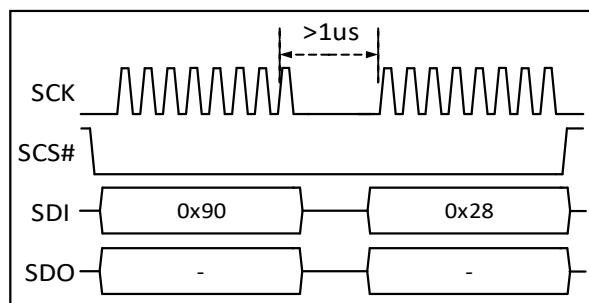
1)、读出串口 1 的 LCR 寄存器值，假设读出 0x00。

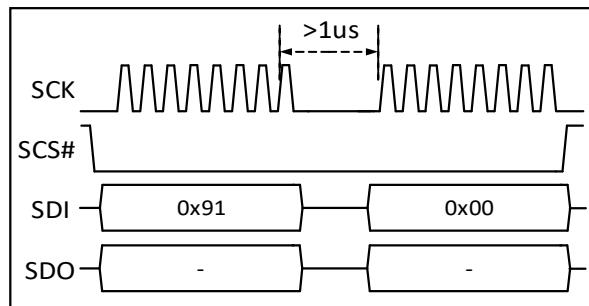


2)、写串口 1 的 LCR 寄存器值，设 DLAB 位，设 8Bit 数据位，1Bit 停止位，无校验。

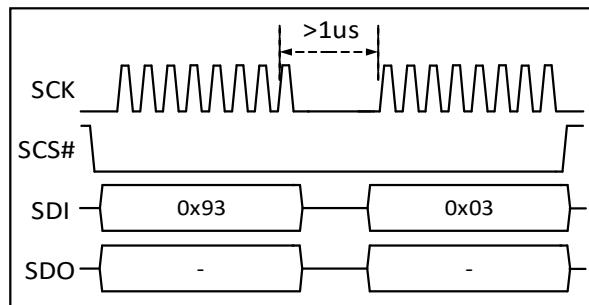


3)、设置串口 1 的波特率除数锁存器 DLL 和 DLM： $32000000 * 15 / 13 / 8 / 115200 \approx 40$ ，DLL 为 0x28，DLM 为 0x00。



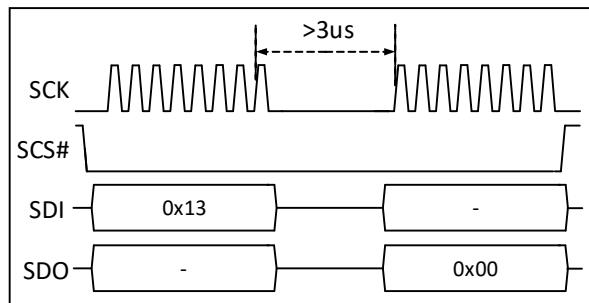


4)、关闭 DLAB。

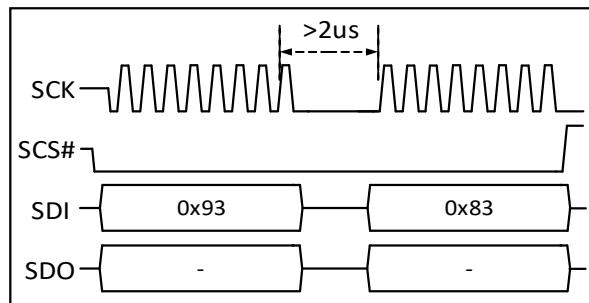


例 2：CH9434D 初始化串口 1 为 115200 波特率，8Bit 数据位，1Bit 停止位，无校验。

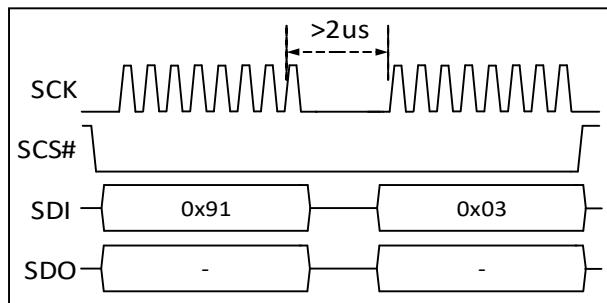
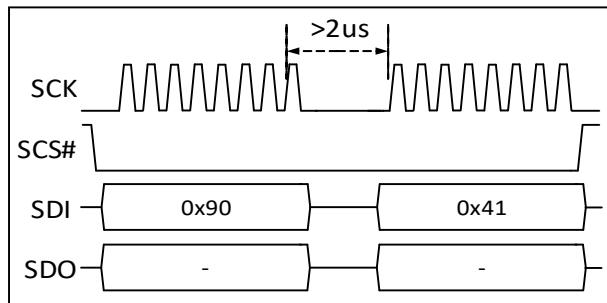
1)、读出串口 1 的 LCR 寄存器值，假设读出 0x00。



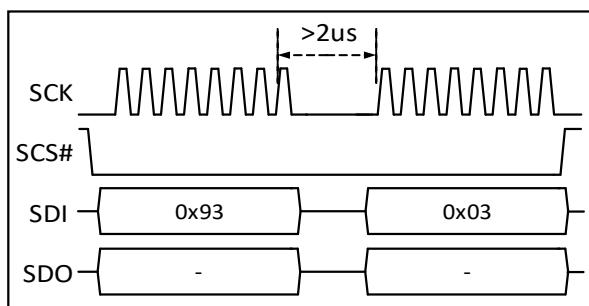
2)、写串口 1 的 LCR 寄存器值，设 DLAB 位，设 8Bit 数据位，1Bit 停止位，无校验。



3)、设置串口 1 的波特率除数锁存器 DLL 和 DLM:  $96000000 * 25 / 4 / 115200 \approx 5208.33$ , DLL 为 0X41, DLM 为 0X03。



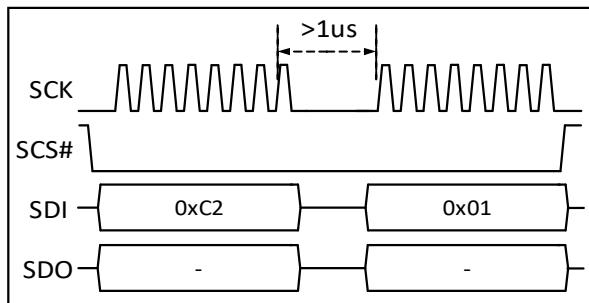
4)、关闭 DLAB。



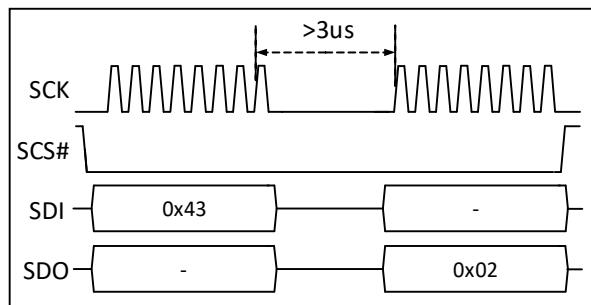
### 11.3 读取串口数据

例 1：CH9434A 串口 1 收到 2 字节数据 0x11、0x22。

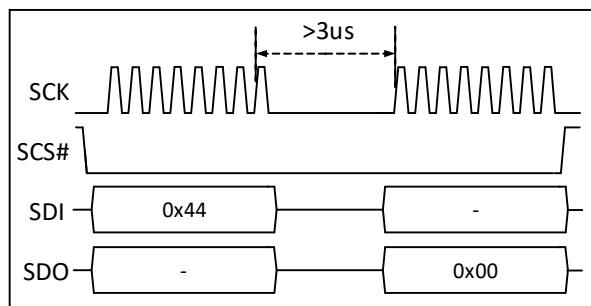
1)、设置读取接收 FIFO 计数器。



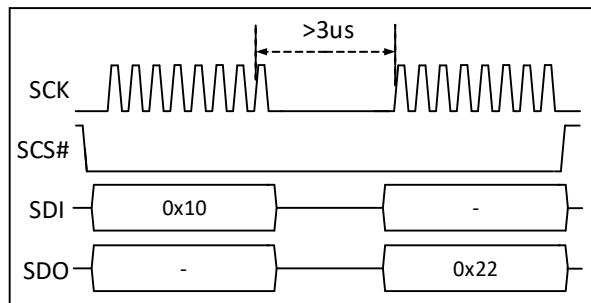
2)、读 FIFO 计数器低字节。



3)、读 FIFO 计数器高字节。

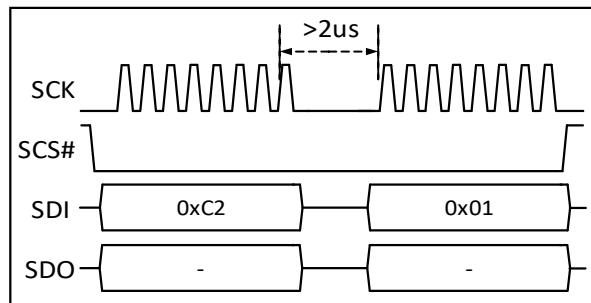


4)、读数据。

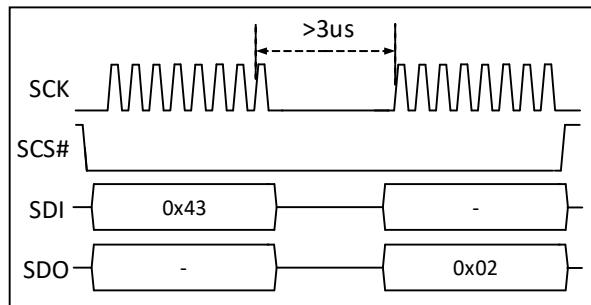


例 2：CH9434A 串口 1 收到 2 字节数据 0x11、0x22。

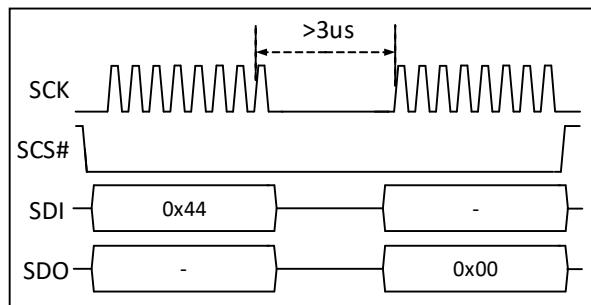
1)、设置读取接收 FIFO 计数器。



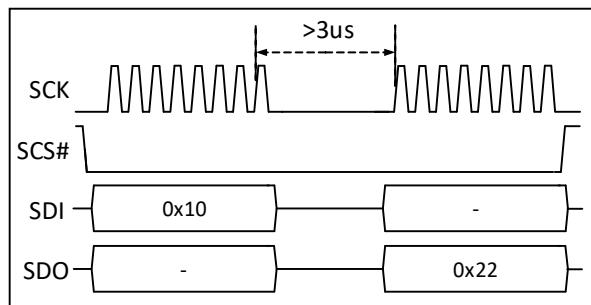
2)、读 FIFO 计数器低字节。



3)、读 FIFO 计数器高字节。

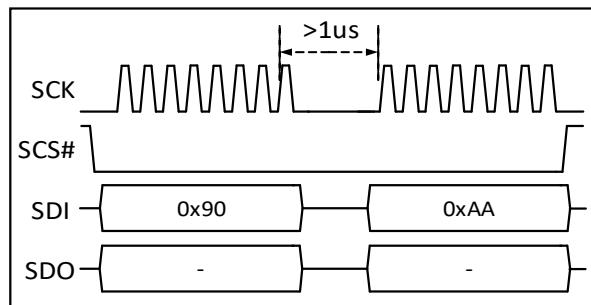


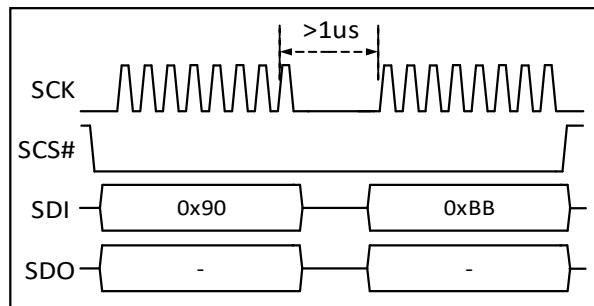
4)、读数据。



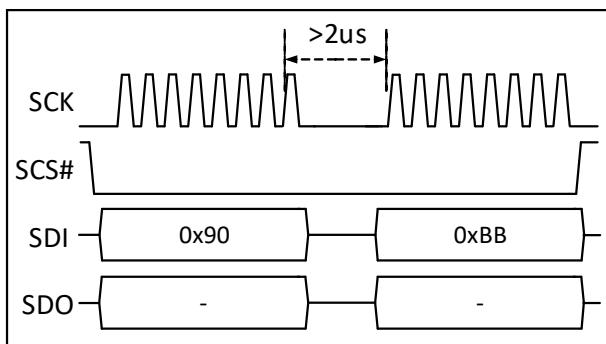
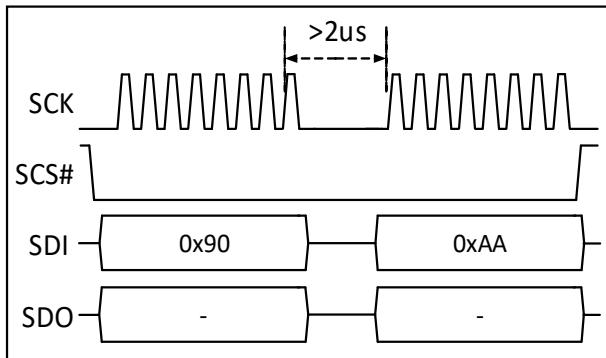
## 11.4 写串口数据

例 1：CH9434A 串口 1 发送 2 字节数据 0xAA、0xBB。





例 2：CH9434D 串口 1 发送 2 字节数据 0xAA、0xBB。



## 11.5 CH9434D 连续写串口数据

例：串口 1 发送 n 字节数据

