

SPI 扩展 8 串口芯片 CH9438

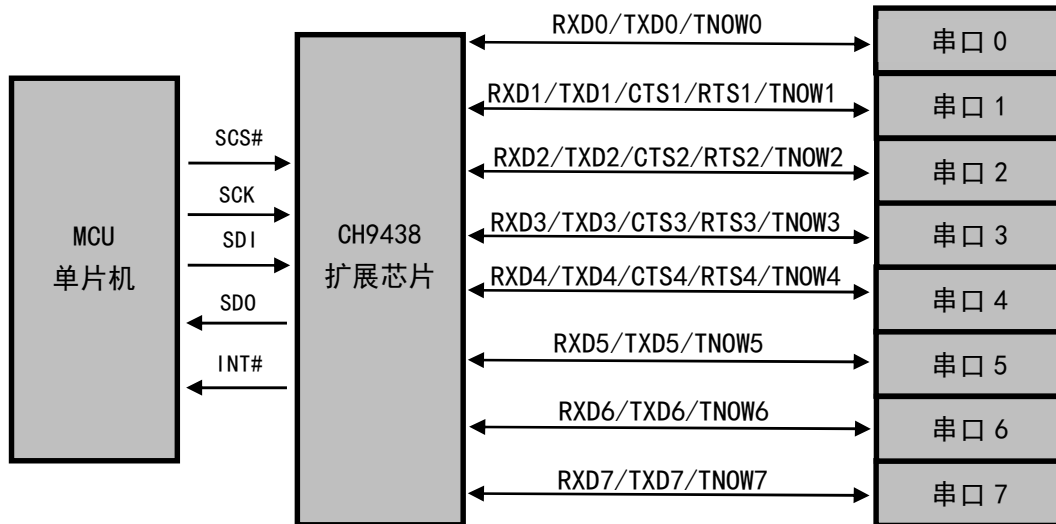
手册

版本: 1.1

<https://wch.cn>

1、概述

CH9438 是一款 SPI 转 8 串口扩展芯片，芯片提供 8 组兼容 16C550 的全双工串口，用于单片机/嵌入式系统扩展异步串口，最高支持 12Mbps 波特率通讯，最多支持 8 路 GPIO，提供半双工收发自动切换引脚 TNOW，支持 RS485。



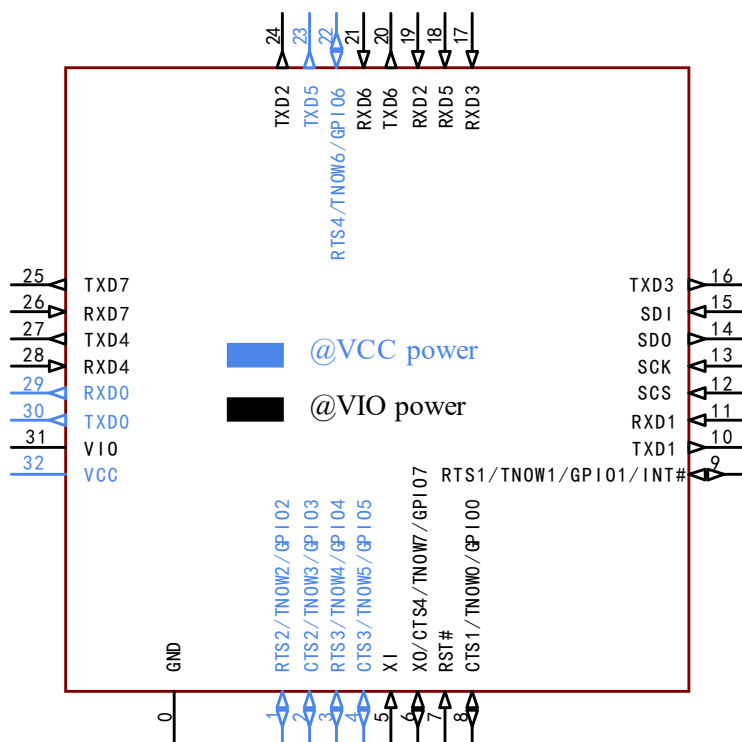
2、特点

- 完全独立的 8 个异步串口，兼容 16C550
- 支持通讯波特率设置，最大支持 12Mbps
- 串口每个方向独立 FIFO 缓存，发送 1536 字节，接收 2048 字节
- 串口支持数据位设置以及校验位设置
- 提供半双工 RS485 收发自动切换引脚，支持 RS485
- I/O 引脚电源电压 V_{IO} 支持 1.8V、2.5V、3.3V
- SPI 最高可达 25Mbit/s
- 支持低功耗睡眠模式，可通过主控接口唤醒
- 提供可配置 GPIO 功能
- 支持内部时钟或者外接 16M 晶体提供时钟
- CH9438F 封装为 QFN32，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、WiFi 等串口模块通讯实现无线传输

4、引脚排列



封装形式	塑体尺寸	引脚节距		封装说明	订货型号
QFN32	4*4mm	0.4mm	15.7mil	四边无引线 32 脚	CH9438F

5、引脚定义

引脚号	引脚名称 ^{(2) (3)}	引脚类型 ⁽¹⁾	引脚说明
0	GND	P	电源地。
1	RTS2 /TNOW2 /GPIO2	I/O	RTS2: UART2 的 MODEM 联络输出信号, 请求发送, 低电平有效。 TNOW2: UART2 的 RS485 收发切换控制引脚。 GPIO2: 通用双向数字 I/O 引脚。
2	CTS2 /TNOW3 /GPIO3	I/O	CTS2: UART2 的 MODEM 联络输入信号, 清除发送, 低电平有效。 TNOW3: UART3 的 RS485 收发切换控制引脚。 GPIO3: 通用双向数字 I/O 引脚。
3	RTS3 /TNOW4 /GPIO4	I/O	RTS3: UART3 的 MODEM 联络输出信号, 请求发送, 低电平有效。 TNOW4: UART4 的 RS485 收发切换控制引脚。 GPIO4: 通用双向数字 I/O 引脚。
4	CTS3 /TNOW5 /GPIO5	I/O	CTS3: UART3 的 MODEM 联络输入信号, 清除发送, 低电平有效。 TNOW5: UART5 的 RS485 收发切换控制引脚。 GPIO5: 通用双向数字 I/O 引脚。
5	XI	A	XI: 高频振荡器的反相输入端。
6	XO /CTS4	I/O/A, FT	XO: 高频振荡器的反相输出端。 CTS4: UART4 的 MODEM 联络输入信号, 清除发送, 低电平有效。

	/TNOW7 /GPIO7		TNOW7: UART7 的 RS485 收发切换控制引脚。 GPIO7: 通用双向数字 I/O 引脚。
7	RST#	I	芯片复位引脚, 低电平有效。
8	CTS1 /TNOW0 /GPIO0	I/O, FT	CTS1: UART1 的 MODEM 联络输入信号, 清除发送, 低电平有效。 TNOW0: UART0 的 RS485 收发切换控制引脚。 GPIO0: 通用双向数字 I/O 引脚。
9	RTS1 /TNOW1 /GPIO1 /INT#	I/O, FT	RTS1: UART1 的 MODEM 联络输出信号, 请求发送, 低电平有效。 TNOW1: UART1 的 RS485 收发切换控制引脚。 GPIO1: 通用双向数字 I/O 引脚。 INT#: 中断输出引脚, 低电平有效。
10	TXD1	O	TXD1: UART1 的串行数据输出。
11	RXD1	I	RXD1: UART1 的串行数据输入。
12	SCS	I	SCS: SPI 片选输入, 低电平有效。
13	SCK	I	SCK: SPI 串行时钟输入。
14	SD0	O	SD0: SPI 串行数据输出。
15	SDI	I	SDI: SPI 串行数据输入。
16	TXD3	O	TXD3: UART3 的串行数据输出。
17	RXD3	I	RXD3: UART3 的串行数据输入。
18	RXD5	I, FT	RXD5: UART5 的串行数据输入。
19	RXD2	I, FT	RXD2: UART2 的串行数据输入。
20	TXD6	O, FT	TXD6: UART6 的串行数据输出。
21	RXD6	I, FT	RXD6: UART6 的串行数据输入。
22	RTS4 /TNOW6 /GPIO6	I/O, FT	RTS4: UART4 的 MODEM 联络输出信号, 请求发送, 低电平有效。 TNOW6: UART6 的 RS485 收发切换控制引脚。 GPIO6: 通用双向数字 I/O 引脚。
23	TXD5	O, FT	TXD5: UART5 的串行数据输出。
24	TXD2	O, FT	TXD2: UART2 的串行数据输出。
25	TXD7	O, FT	TXD7: UART7 的串行数据输出。
26	RXD7	I	RXD7: UART7 的串行数据输入。
27	TXD4	O	TXD4: UART4 的串行数据输出。
28	RXD4	I	RXD4: UART4 的串行数据输入。
29	RXD0	I	RXD0: UART0 的串行数据输入。
30	TXD0	O	TXD0: UART0 的串行数据输出。
31	V _{IO}	P	I/O 电源输入, 需要外接不小于 0.1μF 电源退耦电容。
32	V _{CC}	P	芯片电源输入, 需要外接不小于 0.1μF 电源退耦电容。

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: 串口发送引脚, 初始化后设置为推挽输出高电平模式。其他功能引脚上电后默认为浮空模式, 命令设置相关 I/O 功能开启后引脚会自动设置成对应状态。

注 3: 标记为蓝色的 I/O 引脚由 V_{CC} 供电, 标记为黑色的 I/O 引脚由 V_{IO} 供电。

6、时钟配置

CH9438 默认使用芯片内部时钟，通过倍频后产生 192MHz 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 192MHz 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

CH9438 芯片提供 8 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、08H-0FH 为串口 1、10H-17H 为串口 2、18H-1FH 为串口 3、20H-27H 为串口 4、28H-2FH 为串口 5、30H-37H 为串口 6、38H-3FH 为串口 7。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。

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	R0	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	0	PARMODE1	PARMODE0	PAREN	STOPBIT	WORDSZ1	WORDSZ0
4	X	R/W	MCR	0	0	AFE	0	0	0	RTS	0
5	X	R0	LSR	RFIFOERR	TEMT	THRE	0	FRAMEERR	PARERR	OVERR	DATARDY
6	X	R0	MSR	0	0	0	CTS	0	0	0	△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

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。

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	0	0	0	0	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 为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。

IMODEM: 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。

IENTHRE: 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。

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	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	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, 该位为 0 则禁用 FIFO, 串口关闭。

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。

PARMODE1 和 PARMODE0: 当 PAREN 为 1 时设置奇偶校验位的格式, 00 为奇校验, 01 为偶校验。

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, 00 为 9 位模式, 11 为 8 位模式, 其他值无效, 其中 9 位模式为校验位功能使用。

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。

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。

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 错误。

TEMT: 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。

FRAMEERR: 该位为 1 表示正在从接收 FIFO 中读取的数据存在帧错误, 通常由于缺少有效的停止位。

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。

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

△CTS: 该位为 1 表示 CTS 引脚输入状态发生过变化。

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。

CH9438 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

地址	名称	功能	位	访问	位定义	默认值
41H	R8_TNOW_CTRL1_CFG	TNOW0-3 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: 0: 接收 FIFO; 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
47H	R8_TNOW_CTRL2_CFG	TNOW4-7	[7:4]	R/W	TNOW 引脚电平反相。	0000b

		功能设置	[3:0]	R/W	TNOW 引脚功能控制。	0000b
--	--	------	-------	-----	--------------	-------

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 计数器值为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。

7.3 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	时钟设置, 仅切换时钟源	[7:6]	R/W	系统时钟源模式选择: x0: 内部时钟; x1: 使用外部晶振。	00b
			[5:0]	R0	保留。	0
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: 0: 不进入低功耗状态; 1: IDLE 状态; 2: SLEEP 状态; 3: SHUT 状态; 4-7: 保留。	000b

注: R8_CLK_CTRL_CFG: 芯片时钟设置, 默认使用芯片内部 192MHz 时钟作为串口基准时钟。

7.4 GPIO 寄存器

地址	名称	功能	位	访问	位定义	默认值
50H	R8_GPIO_FUNC_EN_0	GPIO7-0 使能	[7:0]	R/W	[7:0]对应 GPIO7-0 使能控制: 0: 关闭 GPIO 功能; 1: 开启 GPIO 功能。	00h
54H	R8_GPIO_DIR_MOD_0	GPIO1-0 模式设置	[7:0]	R/W	[7:4]对应 GPIO1 模式设置; [3:0]对应 GPIO0 模式设置。 每一个 4bit 定义: [3:2]: 该 IO 功能配置: 输入模式下: 01: 浮空输入; 10: 带有上下拉电阻输入。 输出模式下: 00: 推挽输出; 01: 开漏输出。 [1:0]: 该 IO 的模式设置: 00: 输入模式; 11: 输出模式。	00h
55H	R8_GPIO_DIR_MOD_1	GPIO3-2	[7:0]	R/W	[7:4]对应 GPIO3 模式设置;	00h

		模式设置			[3:0]对应 GPIO2 模式设置； 值定义同上。	
56H	R8_GPIO_DIR_MOD_2	GPIO5-4 模式设置	[7:0]	R/W	[7:4]对应 GPIO5 模式设置； [3:0]对应 GPIO4 模式设置； 值定义同上。	00h
57H	R8_GPIO_DIR_MOD_3	GPIO7-6 模式设置	[7:0]	R/W	[7:4]对应 GPIO7 模式设置； [3:0]对应 GPIO6 模式设置； 值定义同上。	00h
5CH	R8_GPIO_SET_0	GPIO7-0 置位	[7:0]	R/W	[7:0]对应 GPIO7-0 置位控制； 对应位写 1 后，IO 输出高电平， 写 0 无效，同时影响 “GPIO_PIN_VAL”寄存器。	00h
5EH	R8_GPIO_RESET_0	GPIO7-0 复位	[7:0]	R/W	[7:0]对应 GPIO7-0 复位控制； 对应位写 1 后，IO 输出低电平， 写 0 无效，同时影响 “GPIO_PIN_VAL”寄存器。	00h
60H	R8_GPIO_PIN_VAL_0	GPIO7-0 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： GPIO 为输出时： 1：输出高电平； 0：输出低电平。 GPIO 为输入且开启上下拉电阻 功能时： 1：设置 IO 上拉电阻； 0：设置 IO 下拉电阻。 读取寄存器： GPIO 为输入时： 该引脚电平状态。	00h

GPIO 设置方式：

- (1) 打开对应 GPIO 使能位；
- (2) 结合 GPIO_DIR_MOD 和 GPIO_PIN_VAL 对 GPIO 进行模式配置；
- (3) 输出 IO 使用 GPIO_SET 和 GPIO_RESET 设置输出高低电平，输入 IO 使用 GPIO_PIN_VAL 读取电平状态；

芯片不包含的 GPIO，则相关寄存器定义无效，禁止设置。

7.5 扩展寄存器

地址	名称	功能	位	访问	位定义	默认值
65H	CHIP_VER_ADD	当前固件版本	[7:0]	RO	VER[0]：当前版本号低 8 位。	XXh
			[7:0]	RO	VER[1]：当前版本号高 8 位。	XXh
			[7:0]	RO	VER[2]：版本信息校验。	XXh
			[7:0]	RO	VER[3]：结束字符。	7Ch

CHIP_VER_ADD：需注意，读取时解析版本校验数据，版本信息格式 BCD 码。版本号仅用于标识，高版本芯片的功能兼容低版本，主控 MCU 不可针对某一版本号比较。

8、功能说明

8.1 中断与查询

CH9438 芯片的 INT# 引脚为中断请求输出引脚, 主控可在芯片上电完成后设置打开 INT# 引脚功能, INT# 引脚打开后, 自动设置推挽输出, 默认高电平, 低电平有效。由于 8 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, CH9438 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 串口特性

CH9438 串口的串行数据包括 1 个低电平起始位、8 个数据位、1 个或 2 个高电平停止位, 支持奇校验或偶校验。CH9438 支持常用通讯波特率: 3600、4800、9600、14400、19200、28800、33600、38400、56000、57600、76800、115200、128000、153600、230400、460800、921600、250K、500K、1M、1.5M、2M、3M、4M、6M、8M、12M。

使用上述常用波特率时, CH9438 串口接收信号的允许波特率误差小于 2%, 串口发送信号的波特率误差小于 1%。

8.3 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, CH9438 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, CH9438 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, 串口最多发送上一次主控填入的串口数据后停止发送。RTS 在触发 FIFO 达到设定的流控字节数目后自动失效, 芯片内部检测串口接收数目为轮询检测, 所以 RTS 触发会有一定的延后。

8.4 RS485 切换引脚 T_{NOW}

CH9438 串口提供 RS485 切换引脚 T_{NOW}, 引脚功能与其他功能复用, 主控可在芯片上电完成后使能对应的 T_{NOW} 引脚, 再设置 T_{NOW} 设置寄存器开启 T_{NOW} 功能。T_{NOW} 还支持极性调节, 以适应不同的极性使用场景。

8.5 GPIO 功能

CH9438 支持功能引脚复用为 GPIO 功能, CH9438F 最多支持 8 路, 每个 IO 都可以独立设置方向、上拉电阻和下拉电阻配置。

8.6 低功耗模式

CH9438 支持低功耗模式设置。CH9438 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, 唤醒引脚为 SPI 接口的 CS 引脚, 下降沿唤醒。芯片接收到新写入寄存器的睡眠设置后, 会在约几十 μ S 后 (当前无串口收发等情况) 芯片根据睡眠模式对时钟、电源、串口等进行设置, 设置完成后进入睡眠。IDLE: 时钟不停止, 芯片暂停运行, 唤醒后几个 μ S 即可进行通讯。SLEEP: 时钟停止, 芯片停止运行, 芯片内部大部分模块都关闭电源, 唤醒后

需要约几个 ms 等待电源和时钟启动，之后可以进行通讯。SHUT：芯片内部大部分模块电源关闭，仅保留一个唤醒模块工作，芯片唤醒后会重新复位，此时主控继续使用需要重新初始化。

8.7 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。

当写入寄存器数据时，地址和数据两个字节间需要 2uS 延时，发送完数据 CS 释放后需要延时 3uS 才可以进行下一次操作。

当读取寄存器数据时，地址和数据需要延时 3uS，即发送完地址后延时 3uS 进行数据读取，读取数据完成 CS 释放后需要延时 3uS 才可以进行下一次操作。

8.8 串口缓存读写

CH9438 默认支持连续读写串口数据寄存器。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。若出现缓存溢出情况，如串口接收，读取数据需要分包读取，以不超过缓存大小 2048 字节进行分包读取。

8.9 版本号

版本号寄存器读取需要读取 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 0x7C。版本号仅用于标识，高版本芯片的功能兼容低版本，主控 MCU 不要锁定单一版本。

8.10 引脚功能使能说明

CH9438 芯片上电完成后，串口发送引脚会配置成推挽输出高电平，其他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，主控必须写入 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 CH9438 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] 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] 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] 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] 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
UART0_TX_RX	1	PIN30、PIN29
UART1_TX_RX	2	PIN10、PIN11
UART2_TX_RX	3	PIN24、PIN19
UART3_TX_RX	4	PIN16、PIN17
UART4_TX_RX	5	PIN27、PIN28
UART5_TX_RX	6	PIN23、PIN18
UART6_TX_RX	7	PIN20、PIN21
UART7_TX_RX	8	PIN25、PIN26
HSE_XI_XO	9	PIN5、PIN6

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	PIN9
TNOW0 - TNOW7	2 - 9	PIN8、PIN9、PIN1 PIN2、PIN3、PIN4 PIN22、PIN6
CTS1	11	PIN8
CTS2	12	PIN2
CTS3	13	PIN4
CTS4	14	PIN6
RTS1	19	PIN9
RTS2	20	PIN1
RTS3	21	PIN3
RTS4	22	PIN22
GPI00 - GPI07	26 - 33	PIN8、PIN9、PIN1 PIN2、PIN3、PIN4 PIN22、PIN6

复用引脚地址仅对有对应引脚的地址进行设置，剩余没有对应引脚地址为保留地址。

9、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）

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_S	存储时的环境温度	-40	125	°C
V_{CC}	芯片电源电压（ V_{CC} 接电源，GND 接地）	-0.3	4.0	V
V_{IO}	I/O 引脚电源电压	-0.3	4.0	V
V_{IN}	FT（耐受 5V）引脚上的输入电压	-0.3	5.5	V
	V_{CC} 供电的 I/O 引脚上的电压	-0.3	$V_{CC}+0.3$	V
	V_{IO} 供电的 I/O 引脚上的电压	-0.3	$V_{IO}+0.3$	V

9.2 电气参数（测试条件： $T_A = 25^{\circ}\text{C}$ ， $V_{CC} = 3.3\text{V}$ ， $V_{IO} = 3.3\text{V}$ ）

名称	参数说明	条件	最小值	典型值	最大值	单位
V_{CC}	芯片电源电压		1.8	3.3	3.6	V
V_{IO}	I/O 引脚电源电压		1.65		V_{CC}	V
I_{CC}	工作时的总电源电流			21		mA
V_{IL}	标准 I/O，低电平输入电压	$V_{IO} = 3.3\text{V}$	0		0.8	V
		$V_{IO} = 1.8\text{V}$	0		0.4	V
	FT I/O，低电平输入电压	$V_{IO} = 3.3\text{V}$	0		0.8	V
		$V_{IO} = 1.8\text{V}$	0		0.4	V
V_{IH}	标准 I/O，高电平输入电压	$V_{IO} = 3.3\text{V}$	1.9		V_{IO}	V
		$V_{IO} = 1.8\text{V}$	1.3		V_{IO}	V
	FT I/O，高电平输入电压	$V_{IO} = 3.3\text{V}$	1.9		5	V
		$V_{IO} = 1.8\text{V}$	1.3		5	V
V_{OL}	8mA 电流时	$V_{IO} = 3.3\text{V}$			0.4	V
	低电平输出电压	$V_{IO} = 1.8\text{V}$			0.4	V
V_{OH}	8mA 电流时	$V_{IO} = 3.3\text{V}$	2.9			V
	高电平输出电压	$V_{IO} = 1.8\text{V}$	1.4			V
R_{PU}	上拉等效电阻		30	40	55	K Ω
R_{PD}	下拉等效电阻		30	40	55	K Ω

注：以上均为设计参数保证。

V_{CC} 供电的 I/O 引脚以 V_{CC} 代替表中的 V_{IO} 。

正常工作时， V_{IO} 电压不能高于 V_{CC} 电压。

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

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		11.7		mA
I_{CC2}	低功耗模式2：SLEEP睡眠		50	200	μA
I_{CC3}	低功耗模式3：SHUT 睡眠		2		μA

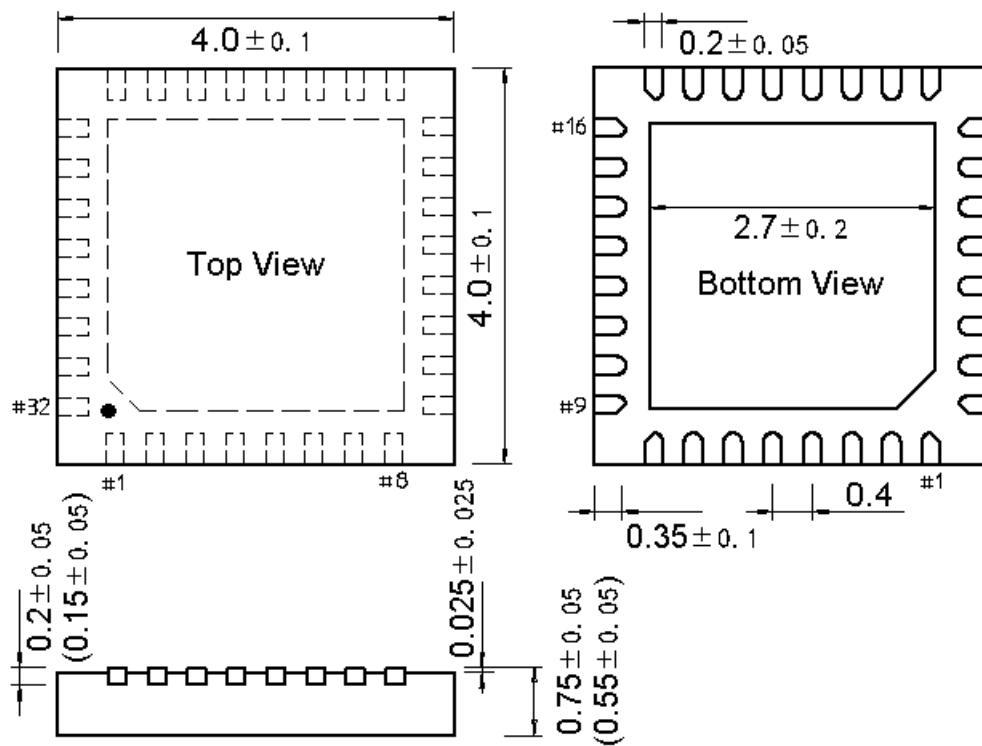
注：低功耗模式需注意 IO 的电平状态和设置，避免浮空产生漏电流。以上参数均为典型值。

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 $\pm 0.2\text{mm}$ 。

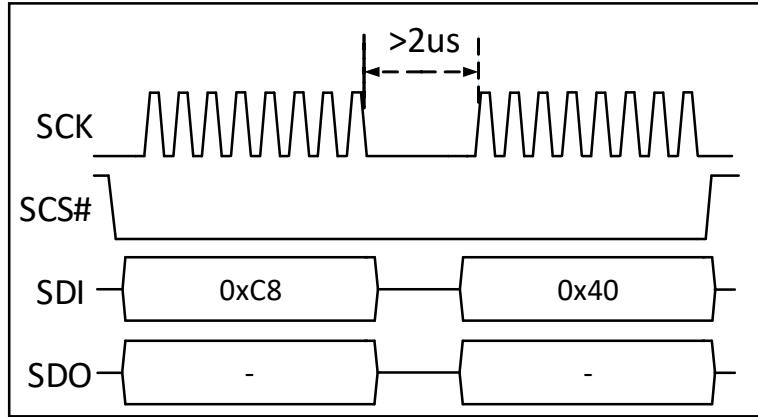
10.1 QFN32



11、应用

11.1 设置时钟

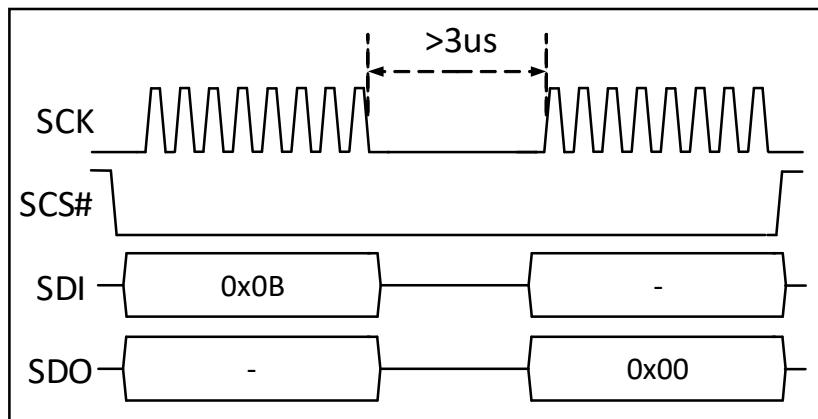
例 1：设置 CH9438 的时钟源为使用外部晶振。



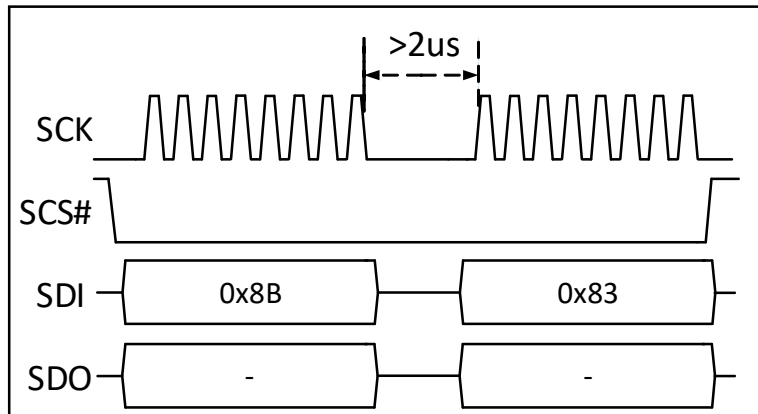
11.2 初始化串口

例 1：CH9438 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。

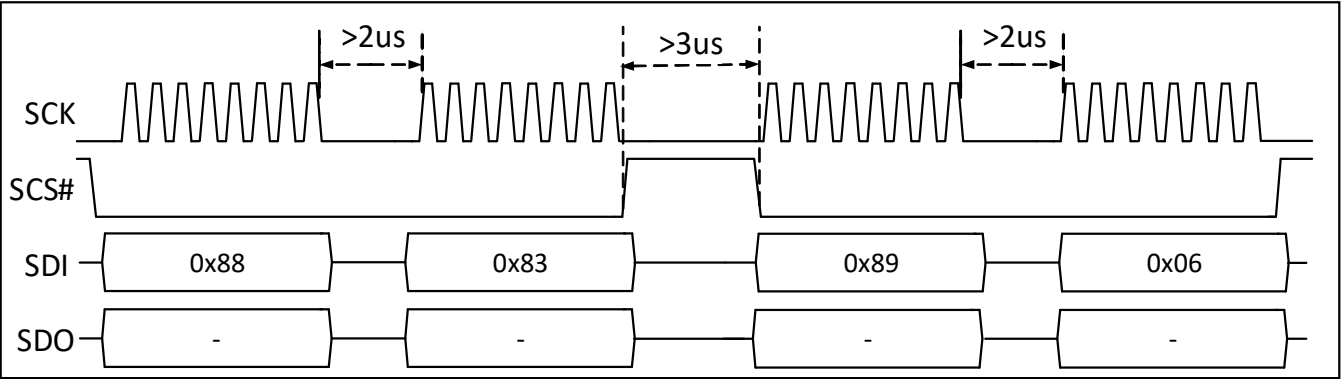


2)、写串口 1 的 LCR 寄存器值，设 DLAB 位，设 8Bit 数据位，1Bit 停止位，无校验。

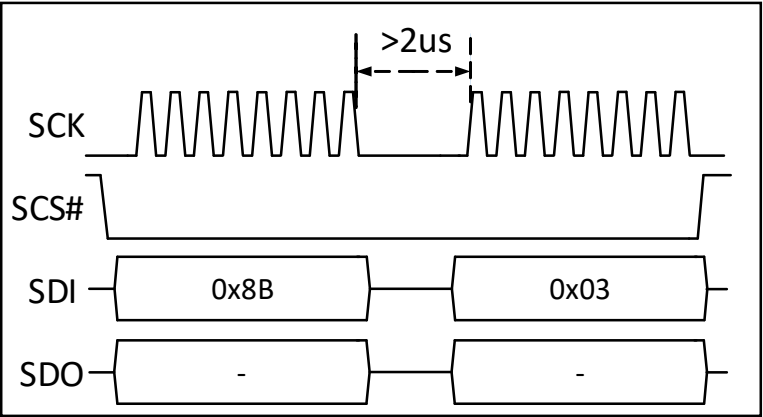


3)、设置串口 1 的波特率除数锁存器 DLL 和 DLM，串口 1 为 115200 波特率时，DLL 为 0x83，DLM 为

0x06。



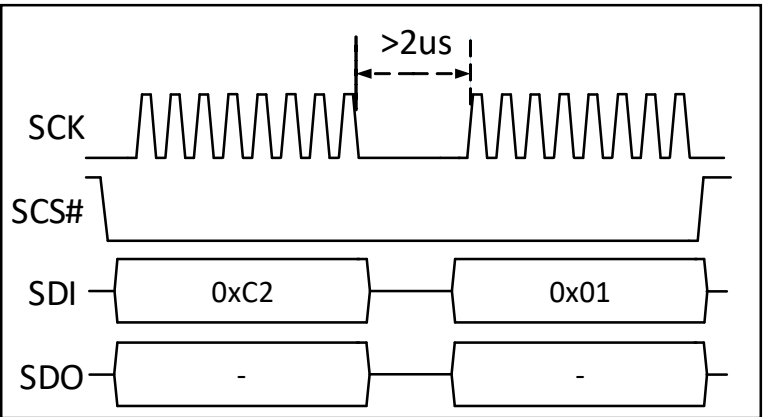
4)、关闭 DLAB 位



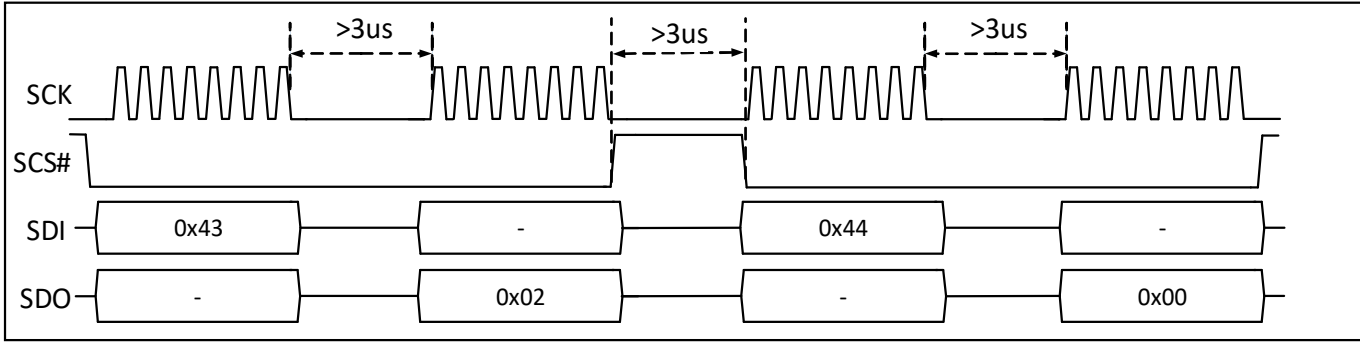
11.3 读取串口数据

例 1：CH9438 串口 1 收到两字节数据 0x11、0x22。

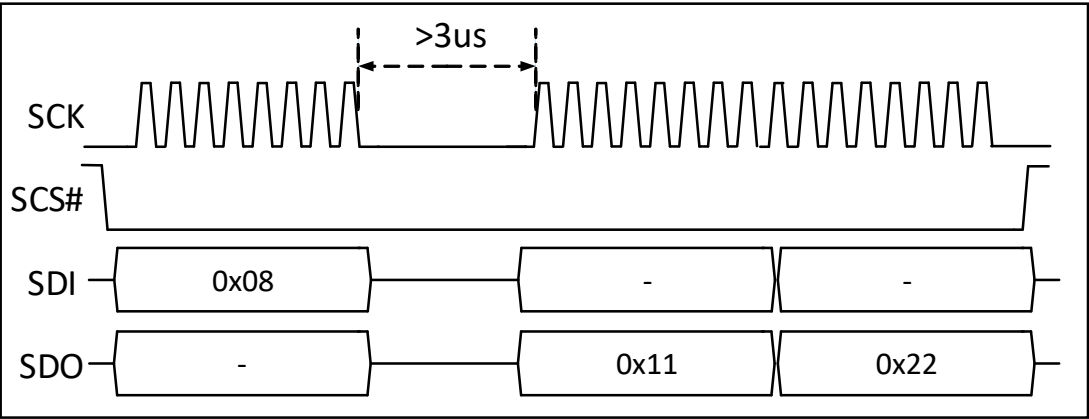
1)、设置 FIFO 计数器控制器



2)、读取接收 FIFO 计数器



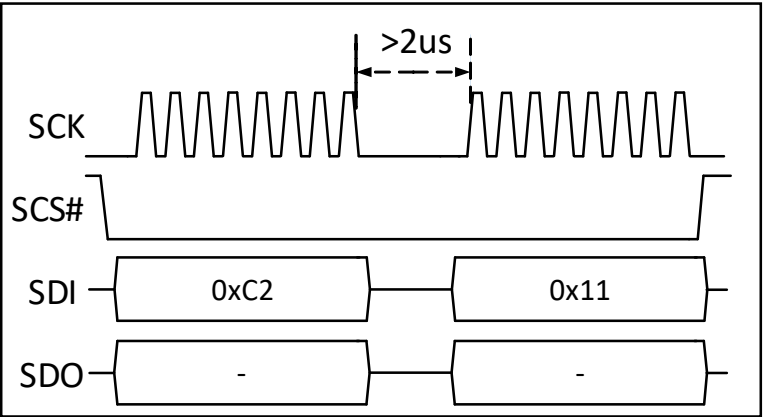
3)、读数据



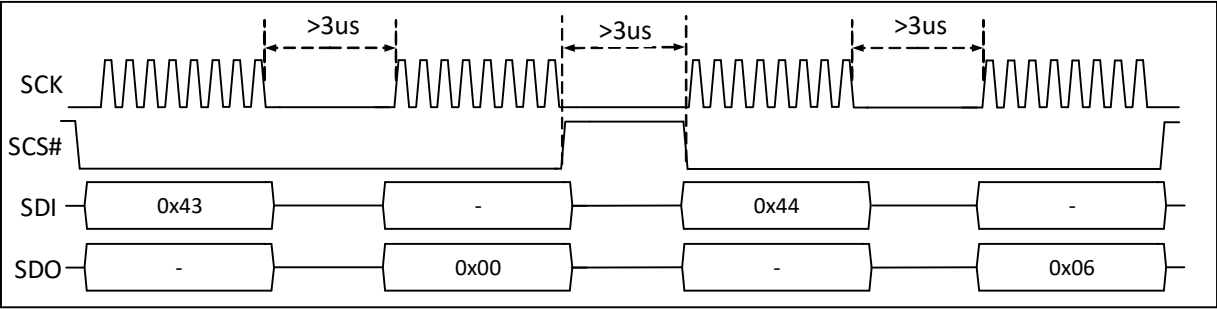
11.4 写串口数据

例 1：CH9438 串口 1 发送 2 字节数据 0xAA、0xBB

1)、设置 FIFO 计数器控制器



2)、读取接收 FIFO 计数器



3)、写数据

