

# WK2204 多总线接口 四通道通用异步收发器 无铅封装



## 1. 产品概述

WK2204是首款具备256级FIFO的低功耗并支持 UART/SPI™/IIC总线接口的4通道UART器件。可以通过模式选择使得该芯片工作于以上任何一种主接口模式，将选定的主接口扩展为4个增强功能的UART。

扩展的子通道的UART具备如下功能特点：

- 每个子通道UART的波特率、字长、校验格式可以独立设置，最高可以提供1.5Mbps的通信速率。
- 每个子通道可以独立设置工作在IrDA红外通信、RS-485自动收发控制、9位网络地址自动识别、软件/硬件自动流量控制等高级工作模式下。
- 每个子通道具备收/发独立的256级FIFO，FIFO的中断可按用户需求进行编程触发点且具备超时中断功能。

WK2204采用QFN24绿色环保的无铅封装，可以工作在2.0~3.6V的宽工作电压范围，具备可配置自动休眠/唤醒功能。

[注]：SPI™ 为MOTOLORA公司的注册商标。

## 2. 基本特性

### 2.1 总体特性

- 支持多种主机接口：可以选择 UART，SPI，IIC
- 超大硬件收发缓存，支持256级FIFO
- 低功耗设计，可以配置自动休眠，自动唤醒模式（uS 级唤醒）
- 宽工作电压设计，工作电压为2.0V~3.6V
- 精简的配置寄存器和控制字，操作简单可靠
- 提供工业级产品
- 高速CMOS工艺，子串口速率最高可达1.5Mbps@3.3V、1Mbps@2.5V
- 采用符合绿色环保政策的QFN24无铅封装

### 2.2 扩展子通道UART特性

- 子通道串口独立配置，高速、灵活：  
每个子串口为全双工，每个子串口可以通过软件开启/关闭  
波特率可以独立设置，子串口最高可以达到1.5Mbps  
每个子串口字符格式包括数据长度、停止位数、奇偶校验模式可以独立设置

- 完善的子串口状态查询功能
- 可以实现对单个子串口软件复位
- FIFO功能：
  - 每个子串口具备独立的 256 级发送 FIFO，发送 FIFO 触发点可编程
  - 每个子串口具备独立的 256 级接收 FIFO，接收 FIFO 触发点可编程
  - 软件 FIFO 使能和清空
  - FIFO 状态和计数器输出
- 流量控制：
  - 支持 RTS、CTS 的硬件自动流量控制
  - 支持 XON/XOFF 的软件自动流量控制，XON/XOFF 可编程字符自动发送/识别
- RS-485功能：
  - RTS 控制的自动 RS-485 收发控制
  - RS-485 网络地址自动识别功能
  - RS-485 数据长度可编程配置为 8bit 或 9bit
- 错误检测：
  - 支持奇偶校验错误、数据帧错误、溢出错误及 Line-Break 错误检测
  - 支持起始位错误检测
- 内置符合SIR标准的IrDA红外收发编解码器，传输速度可达115.2K bit/s
- 中断特点：
  - 具备子串口接收 FIFO 超时中断
  - 支持 Line-Break 错误中断

## 2.3 UART主接口特性

- 主接口为标准的三线UART串口 (RX, TX, GND), 无需其它地址信号、控制信号线
- 波特率自适应技术，最高速度可以达到2M bit/s
- 可选择的奇校验，偶校验和无校验模式
- 无需地址线控制的串口扩展方式，通过芯片内置的协议处理器实现多串口扩展
- UART主接口可以通过引脚设置为红外模式
- 支持最长16字节连续收发

## 2.4 SPI主接口特性

- 最高速度10M bit/s
- 仅支持SPI从模式
- SPI模式0
- 支持最长256字节连续收发

## 2.5 IIC主接口特性

- 支持IIC总线接口
- 最高速度1M bit/s
- 仅支持IIC从模式
- 支持最长256字节连续收发

## 3. 应用领域

- 多串口服务器/多串口卡
- 工业/自动化现场RS-485控制
- 通过2G/3G/4G的无线数据传输
- 车载信息平台/车载GPS定位系统
- 远传自动抄表（AMR）系统
- POS/税控POS/金融机具
- DSP/嵌入式系统

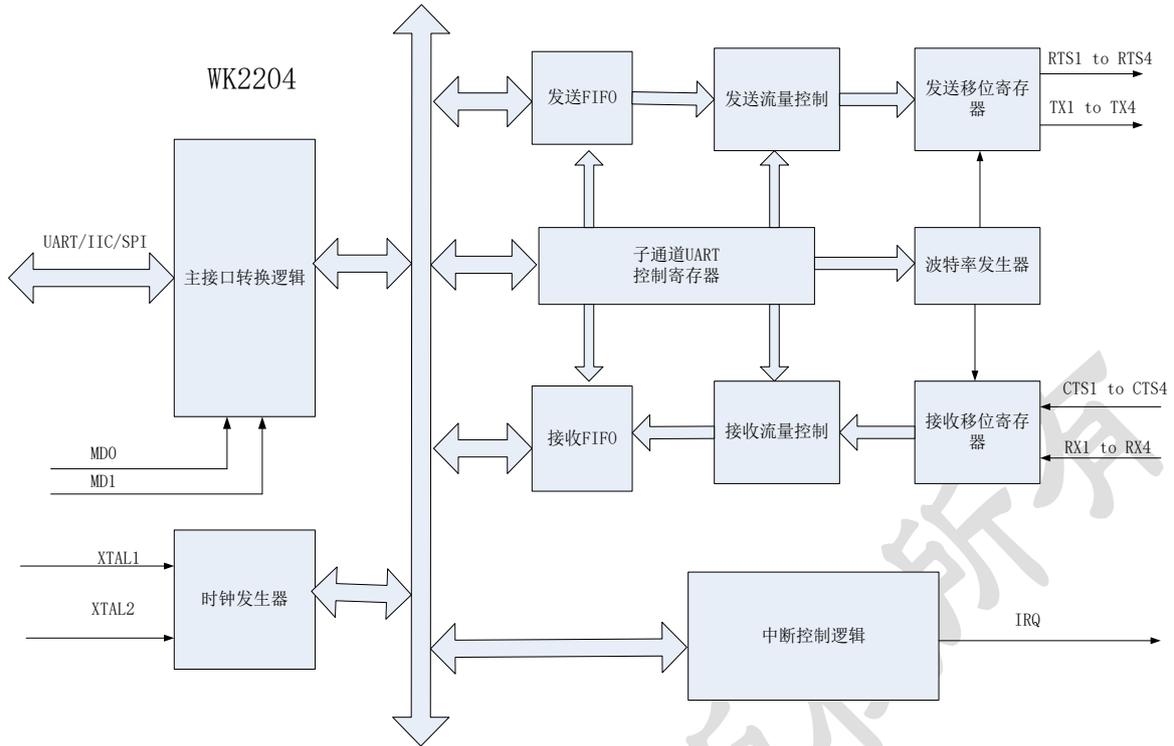
## 4. 订购信息

表4.1 WK2204 订购信息

| 产品型号        | 封装         | 说明                    |
|-------------|------------|-----------------------|
| WK2204-IQNG | QFN24 无铅封装 | 普通工业级； 工作温度 -45℃~+85℃ |

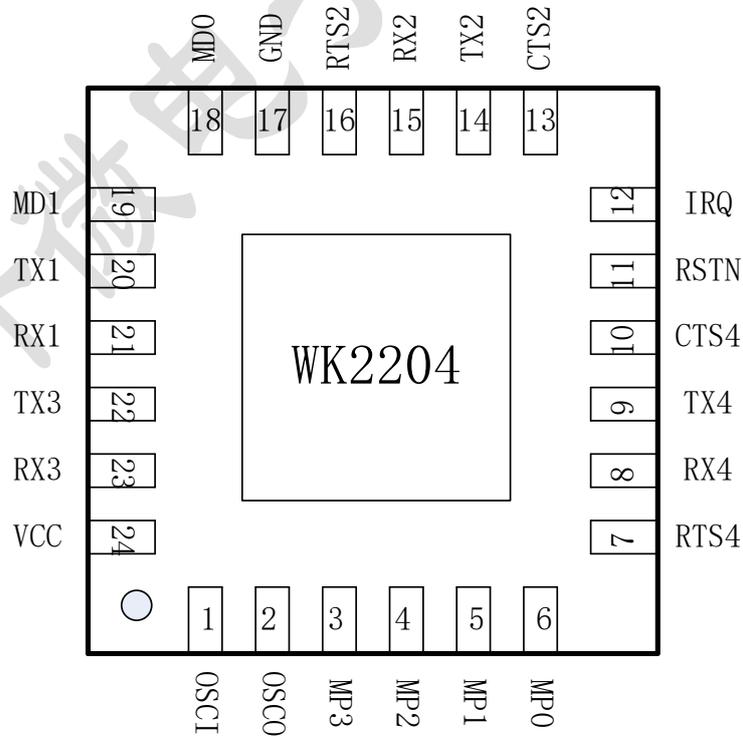
## 5. 原理框图

图5.1 WK2204 原理框图



## 6. 封装引脚

### 6.1 封装图



## 6.2 引脚描述

表6.2 WK2204 引脚描述

| 名称                       | 管脚                  | 类型  | 描述  |
|--------------------------|---------------------|-----|---|
| MP3                      | 3                   | I/O | 当主接口为SPI时，为SSEL (SPI片选) 功能引脚：低电平有效；<br>当主接口为IIC时，为IA1 (IIC器件地址高) 功能引脚；<br>当主接口为UART时，为IR (主口红外通信模式) 功能引脚；<br>IR=0 红外通信模式；<br>IR=1 普通UART通信模式；IR默认为高电平。  |
| MP2                      | 4                   | I/O | 当主接口为SPI时，为SCLK (SPI 时钟输入) 功能引脚；<br>当主接口为IIC时，为SCL (IIC时钟输入) 功能引脚；<br>当主接口为UART时，为MRX (主口UART接收) 功能引脚。  |
| MP1                      | 5                   | I/O | 当主接口为SPI时，为MOSI功能引脚；<br>当主接口为IIC时，为IA0 (IIC器件地址低) 功能引脚；<br>当主接口为UART时，为MTX (主口UART发送) 功能引脚。   |
| MP0                      | 6                   | I/O | 当主接口为SPI时，为MISO功能引脚；<br>当主接口为IIC时，为SDA功能引脚；<br>当主接口为UART时，为NC (空)。  |
| RTS2<br>RTS4             | 16<br>7             | 0   | 子串口2/4的请求发送信号 (Request To Send), 低电平有效。<br>当RTS <sub>x</sub> =0时，表明WK2204的相应子串口接收已准备就绪，请求与其相连的UART发送数据。RTS可以通过子串口流量控制寄存器进行设置。<br>当子串口工作在自动流量控制模式下时，RTS通过自动流量控制逻辑控制数据收发。<br>当子串口工作在RS-485自动收发模式下，该引脚用于控制RS-485数据的自动收发转换。(和RTS <sub>n</sub> 相关的功能只有2/4串口才有) |
| CTS2<br>CTS4             | 13<br>10            | I   | 子串口2/4的清除发送信号 (Clear To Send), 低电平有效。<br>当CTS <sub>x</sub> =0时，表明UART已经准备好接收WK2204相应的子串口发送数据。可以通过读取子串口状态寄存器读取CTS的相应状态。<br>当子串口工作在自动流量控制模式下时，CTS通过自动流量控制逻辑控制数据收发。(和CTS <sub>n</sub> 相关的功能只有2/4串口才有)  |
| RX1<br>RX2<br>RX3<br>RX4 | 21<br>15<br>23<br>8 | I   | 子通道串口串行数据输入。<br>RX 将所连数据UART的串行数据输入WK2204的相应管脚。   |

|      |    |   |  |
|------|----|---|--|
| TX1  | 20 | 0 | 子通道串口串行数据输出。   |
| TX2  | 14 |   | TX 将串行数据输出到与其连接的器件引脚。  |
| TX3  | 22 |   |  |
| TX4  | 9  |   |  |
| RSTN | 11 | I | 硬件复位引脚，低电平复位有效   |
| VCC  | 24 | - | 电源 2.0V~3.6V工作电压   |
| IRQ  | 12 | 0 | 中断输出信号，低电平有效。 <b>建议外接上拉电阻，典型取值5.1K</b>   |
| GND  | 17 | - | 电源地  |
| MD0  | 18 | I | 主接口模式选择信号：<br>MD1 MD0=00 SPI接口；<br>MD1 MD0=10 IIC接口；<br>MD1 MD0=11 UART接口；<br>M1 M0芯片内建上拉电路，悬空时M1 M0=11； |
| MD1  | 19 |   |  |
| OSCI | 1  | I | 晶振输入。 <b>注意：需要和晶振并联1M的电阻。</b>  |
| OSCO | 2  | 0 | 晶振输出。  |

## 7. 寄存器描述

### 7.1 寄存器列表

WK2204的寄存器地址按6位地址编号，地址000000~111111，分为全局寄存器和子串口寄存器。

全局寄存器7个，全局寄存器的地址具体排列见表7.1.1：

表7.1.1 全局寄存器列表

| 寄存器地址[5:0] | 寄存器名称 | 类型  | 寄存器功能描述       |
|------------|-------|-----|---------------|
| 000000     | GENA  | R/W | 全局控制寄存器       |
| 000001     | GRST  | R/W | 全局子串口复位寄存器    |
| 000010     | GMUT  | R/W | 全局主串口控制寄存器    |
| 010000     | GIER  | R/W | 全局中断寄存器       |
| 010001     | GIFR  | R   | 全局中断标志寄存器     |
| 100001     | GPDIR | R/W | 全局 GPIO 方向寄存器 |
| 110001     | GPDAT | R/W | 全局 GPIO 数据寄存器 |

子串口寄存器25个，其排列为C1C0 REG[3:0]，高两位为子串口通道号，低4位为寄存器地址，按低4位的寄存器地址具体排列见表7.1.2：

表7.1.2 子串口控制寄存器

| 寄存器地址[3:0]   | 寄存器名称 | 类型  | 寄存器功能描述   |        |
|--------------|-------|-----|-----------|--------|
| (C1,C0) 0011 | SPAGE | R/W | 子串口页控制寄存器 |        |
| (C1,C0) 0100 | SCR   | R/W | 子串口控制寄存器  | SPAGE0 |
| (C1,C0) 0101 | LCR   | R/W | 子串口配置寄存器  | SPAGE0 |

|              |        |     |                        |        |
|--------------|--------|-----|------------------------|--------|
| (C1,C0) 0110 | FCR    | R/W | 子串口 FIFO 控制寄存器         | SPAGE0 |
| (C1,C0) 0111 | SIER   | R/W | 子串口中断使能寄存器             | SPAGE0 |
| (C1,C0) 1000 | SIFR   | R/W | 子串口中断标志寄存器             | SPAGE0 |
| (C1,C0) 1001 | TFCNT  | R   | 子串口发送 FIFO 计数寄存器       | SPAGE0 |
| (C1,C0) 1010 | RFCNT  | R   | 子串口接收 FIFO 计数寄存器       | SPAGE0 |
| (C1,C0) 1011 | FSR    | R   | 子串口 FIFO 状态寄存器         | SPAGE0 |
| (C1,C0) 1100 | LSR    | R   | 子串口接收状态寄存器             | SPAGE0 |
| (C1,C0) 1101 | FDAT   | R/W | 子串口 FIFO 数据寄存器         | SPAGE0 |
| (C1,C0) 1110 | FWCR   | R/W | 子串口流量控制寄存器             | SPAGE0 |
| (C1,C0) 1111 | RS485  | R/W | 子串口 RS485 控制寄存器        | SPAGE0 |
| (C1,C0) 0100 | BAUD1  | R/W | 子串口波特率配置寄存器高字节         | SPAGE1 |
| (C1,C0) 0101 | BAUD0  | R/W | 子串口波特率配置寄存器低字节         | SPAGE1 |
| (C1,C0) 0110 | PRES   | R/W | 子串口波特率配置寄存器小数部分        | SPAGE1 |
| (C1,C0) 0111 | RFTL   | R/W | 子串口接收 FIFO 中断触发点配置寄存器  | SPAGE1 |
| (C1,C0) 1000 | TFTL   | R/W | 子串口发送 FIFO 中断触发点配置寄存器  | SPAGE1 |
| (C1,C0) 1001 | FWTH   | R/W | 子串口 FIFO 流控触发点配置寄存器高字节 | SPAGE1 |
| (C1,C0) 1010 | FWTL   | R/W | 子串口 FIFO 流控触发点配置寄存器低字节 | SPAGE1 |
| (C1,C0) 1011 | XON1   | R/W | 子串口 XON1 寄存器           | SPAGE1 |
| (C1,C0) 1100 | XOFF1  | R/W | 子串口 XOFF1 寄存器          | SPAGE1 |
| (C1,C0) 1101 | SADR   | R/W | RS485 地址寄存器            | SPAGE1 |
| (C1,C0) 1110 | SAEN   | R/W | RS485 地址使能寄存器          | SPAGE1 |
| (C1,C0) 1111 | RTSDLY | R/W | RS485 延时控制寄存器          | SPAGE1 |

C1, C0: 子通道号, 00~11分别对应子串口1到子串口4

## 7.2 寄存器描述

### 7.2.1 GENA全局控制寄存器: (000000)

| 位    | 复位值 | 功能描述  | 类型  |
|------|-----|---|-----|
| Bit7 | 1   | M1 M1 引脚电平状态(M1 默认是高电平)                           | R   |
| Bit6 | 1   | M0 M0 引脚电平状态(M0 默认是高电平)                           | R   |
| Bit5 | 1   | RSV(保留位)  | R   |
| Bit4 | 1   | RSV(保留位)  | R   |
| Bit3 | 0   | UT4EN 子串口 4 时钟使能位<br>(关断子串口时钟, 可实现更低功耗)<br>0: 未使能 | W/R |

|      |   |  |     |
|------|---|--|-----|
|      |   | 1: 使能  |     |
| Bit2 | 0 | UT3EN 子串口 3 时钟使能位<br>(关断子串口时钟, 可实现更低功耗)<br>0: 未使能<br>1: 使能 | W/R |
| Bit1 | 0 | UT2EN 子串口 2 时钟使能位<br>(关断子串口时钟, 可实现更低功耗)<br>0: 未使能<br>1: 使能 | W/R |
| Bit0 | 0 | UT1EN 子串口 1 时钟使能位<br>(关断子串口时钟, 可实现更低功耗)<br>0: 未使能<br>1: 使能 | W/R |

## 7.2.2 GRST全局子串口复位寄存器: (000001)

| 位    | 复位值 | 功能描述   | 类型    |
|------|-----|--|-------|
| Bit7 | 0   | UT4SLEEP 子串口 4 休眠状态位<br>(降低功耗, 可自动唤醒)<br>0: 未休眠<br>1: 休眠                 | R     |
| Bit6 | 0   | UT3SLEEP 子串口 3 休眠状态位<br>(降低功耗, 可自动唤醒)<br>0: 未休眠<br>1: 休眠                 | R     |
| Bit5 | 0   | UT2SLEEP 子串口 2 休眠状态位<br>(降低功耗, 可自动唤醒)<br>0: 未休眠<br>1: 休眠                 | R     |
| Bit4 | 0   | UT1SLEEP 子串口 1 休眠状态位<br>(降低功耗, 可自动唤醒)<br>0: 未休眠<br>1: 休眠                 | R     |
| Bit3 | 0   | UT4RST 子串口 4 软复位控制位<br>(该位写 1 复位, 完成后自动置 0)<br>0: 未复位子串口 4<br>1: 复位子串口 4 | W1/R0 |
| Bit2 | 0   | UT3RST 子串口 3 软复位控制位<br>(该位写 1 复位, 完成后自动置 0)<br>0: 未复位子串口 3<br>1: 复位子串口 3 | W1/R0 |
| Bit1 | 0   | UT2RST 子串口 2 软复位控制位<br>(该位写 1 复位, 完成后自动置 0)<br>0: 未复位子串口 2               | W1/R0 |

|      |   |   |       |
|------|---|---|-------|
|      |   | 1: 复位子串口 2  |       |
| Bit0 | 0 | <b>UT1RST</b> 子串口 1 软复位控制位<br>(该位写 1 复位, 完成后自动置 0)<br>0: 未复位子串口 1<br>1: 复位子串口 1 | W1/R0 |

## 7.2.3 GMUT全局主串口控制寄存器: (000010)

| 位         | 复位值 | 功能描述   | 类型    |
|-----------|-----|--|-------|
| Bit7      | 0   | RSV(保留位)   | W1/R0 |
| Bit6 -- 4 | 0   | RSV(保留位)   | R0    |
| Bit3      | 0   | <b>PAEN</b> 主串口校验使能控制位<br>0: 无校验<br>1: 使能校验(根据 PAM1, PAM0 的配置确定校验模式)                               | W/R   |
| Bit2 -- 1 | 0   | <b>PAM1—0</b> 主串口校验模式使能控制位<br>当 PAEN=1 主串口校验使能时:<br>00: 强制 0 校验; 01: 奇校验;<br>10: 偶校验; 11: 强制 1 校验; | W/R   |
| Bit0      | 0   | <b>GSTPL</b> 主串口停止位长度设置位<br>0: 1bit<br>1: 2bits  | W/R   |

## 7.2.4 GIER全局中断寄存器: (010000)

| 位          | 复位值 | 功能描述  | 类型  |
|------------|-----|---|-----|
| Bit7 --- 5 | 000 | RSV(保留位)                                      | R   |
| Bit4       | 0   | RSV(保留位)                                      | W/R |
| Bit3       | 0   | <b>UT4IE</b> 子串口 4 中断使能控制位<br>0: 未使能<br>1: 使能 | W/R |
| Bit2       | 0   | <b>UT3IE</b> 子串口 3 中断使能控制位<br>0: 未使能<br>1: 使能 | W/R |
| Bit1       | 0   | <b>UT2IE</b> 子串口 2 中断使能控制位<br>0: 未使能<br>1: 使能 | W/R |
| Bit0       | 0   | <b>UT1IE</b> 子串口 1 中断使能控制位<br>0: 未使能<br>1: 使能 | W/R |

## 7.2.5 GIFR全局中断标志寄存器：(010001)

| 位          | 复位值 | 功能描述                                   | 类型 |
|------------|-----|--|----|
| Bit7 --- 4 | 000 | RSV(保留位)                               | R  |
| Bit3       | 0   | UT4INT 子串口 4 中断标志位<br>0: 无中断<br>1: 有中断 | R  |
| Bit2       | 0   | UT3INT 子串口 3 中断标志位<br>0: 无中断<br>1: 有中断 | R  |
| Bit1       | 0   | UT2INT 子串口 2 中断标志位<br>0: 无中断<br>1: 有中断 | R  |
| Bit0       | 0   | UT1INT 子串口 1 中断标志位<br>0: 无中断<br>1: 有中断 | R  |

## 7.2.6 GPDIR全局GPIO方向寄存器：(100001)

| 位                   | 复位值 | 功能描述  | 类型  |
|---------------------|-----|---|-----|
| Bit[x]<br>x: 7 -- 0 | 0   | GPDIR [x] GPIOx 输入输出控制位(x: 7 - 0)<br>0: 输入<br>1: 输出 | W/R |

## 7.2.7 GPDAT全局GPIO数据寄存器：(110001)

| 位         | 复位值     | 功能描述   | 类型  |
|-----------|---------|--|-----|
| Bit7 -- 0 | xxxxxxx | 只有当主接口位 UART/SPI/IIC 时,可通过该寄存器访问 GPIO7--- GPIO0 引脚 | W/R |

## 7.2.8 SPAGE子串口页控制寄存器：(0011)

| 位          | 复位值    | 功能描述   | 类型  |
|------------|--------|--|-----|
| Bit7 --- 1 | 000000 | RSV(保留位)   | R   |
| Bit0       | 0      | PAGE 子串口页控制位(子串口寄存器分布在 PAGE0 和 PAGE1 上,不同页之间的切换,通过该寄存器来控制)<br>0: PAGE0<br>1: PAGE1 | W/R |

## 7.2.9 SCR子串口控制寄存器：(PAGE0:0100)

| 位         | 复位值 | 功能描述                       | 类型  |
|-----------|-----|----------------------------|-----|
| Bit7 -- 3 | 000 | RSV(保留位)                   | W/R |
| Bit2      | 0   | SLEEPEN 子串口休眠使能位<br>0: 不使能 | W/R |

|      |   |                                  |     |
|------|---|----------------------------------|-----|
|      |   | 1: 使能                            |     |
| Bit1 | 0 | TXEN 子串口发送使能位<br>0: 不使能<br>1: 使能 | W/R |
| Bit0 | 0 | RXEN 子串口接收使能位<br>0: 不使能<br>1: 使能 | W/R |

## 7.2.10 LCR子串口配置寄存器：(PAGE0:0101)

| 位         | 复位值 | 功能描述  | 类型  |
|-----------|-----|---|-----|
| Bit7 -- 6 | 00  | RSV(保留位)  | W/R |
| Bit5      | 0   | BREAK 子串口 Line-Break 输出控制位<br>0: 正常输出<br>1: Line-Break 输出(TX 强制输出 0)                  | W/R |
| Bit4      | 0   | IREN 子串口红外使能位<br>0: 普通模式<br>1: 红外模式   | W/R |
| Bit3      | 0   | PAEN 子串口校验使能位<br>0: 无校验位 (8 位数据)<br>1: 有校验位 (9 位数据)                                   | W/R |
| Bit2 -- 1 | 0   | PAM1—0 子串口校验模式选择位<br>当 PAEN=1 子串口校验使能时:<br>00: 0 校验 ; 01: 奇校验 ;<br>10: 偶校验 ; 11: 1 校验 | W/R |
| Bit0      | 0   | STPL 子串口停止位长度控制位<br>0: 1bit<br>1: 2bits   | W/R |

## 7.2.11 FCR子串口FIFO控制寄存器：(PAGE0:0110)

| 位          | 复位值 | 功能描述   | 类型  |
|------------|-----|--|-----|
| Bit7 --- 6 | 00  | TFTRIG[1:0] 子串口发送 FIFO 触点设置位<br>当 TFTL[7:0]等于 0 时:<br>00: 8Byte            01:16 Byte<br>10: 24 Byte        11:30 Byte | W/R |
| Bit5 --- 4 | 00  | RFTRIG[1:0] 子串口接收 FIFO 触点设置位<br>当 RFTL[7:0]等于 0 时:<br>00: 8Byte            01:16 Byte<br>10: 24 Byte        11:28 Byte | W/R |
| Bit3       | 0   | TFEN 子串口发送 FIFO 使能位<br>0: 不使能<br>1: 使能   | W/R |
| Bit2       | 0   | RFEN 子串口接收 FIFO 使能位  | W/R |

|      |   |   |       |
|------|---|---|-------|
|      |   | 0: 不使能<br>1: 使能   |       |
| Bit1 | 0 | <b>TFRST</b> 子串口发送 FIFO 复位位<br>(该位写 1 复位, 完成后自动置 0)<br>0: 未使能复位<br>1: 复位 FIFO | W1/R0 |
| Bit0 | 0 | <b>RFRST</b> 子串口接收 FIFO 复位位<br>(该位写 1 复位, 完成后自动置 0)<br>0: 未使能复位<br>1: 复位 FIFO | W1/R0 |

## 7.2.12 SIER子串口中断使能寄存器: (PAGE0:0111)

| 位    | 复位值 | 功能描述  | 类型  |
|------|-----|---|-----|
| Bit7 | 0   | <b>FERR_IEN</b> 接收 FIFO 数据错误中断使能位<br>0: 禁止接收 FIFO 数据错误中断<br>1: 使能接收 FIFO 数据错误中断     | W/R |
| Bit6 | 0   | <b>CTS_IEN</b> CTS 中断使能位<br>0: 禁止 CTS 中断<br>1: 使能 CTS 中断                            | W/R |
| Bit5 | 0   | <b>RTS_IEN</b> RTS 中断使能位<br>0: 禁止 RTS 中断<br>1: 使能 RTS 中断                            | W/R |
| Bit4 | 0   | <b>XOFF_IEN</b> XOFF 中断使能位<br>0: 禁止 XOFF 中断<br>1: 使能 XOFF 中断。当子串口接收到 XOFF 特殊字符时产生中断 | W/R |
| Bit3 | 0   | <b>TFEMPTY_IEN</b> 发送 FIFO 空中断使能位<br>0: 禁止发送 FIFO 空中断<br>1: 使能发送 FIFO 空中断           | W/R |
| Bit2 | 0   | <b>TFTRIG_IEN</b> 发送 FIFO 触点中断使能位<br>0: 禁止发送 FIFO 触点中断<br>1: 使能发送 FIFO 触点中断         | W/R |
| Bit1 | 0   | <b>RXOVT_IEN</b> 接收 FIFO 超时中断使能位<br>0: 禁止接收 FIFO 超时中断<br>1: 使能接收 FIFO 超时中断          | W/R |
| Bit0 | 0   | <b>RFTRIG_IEN</b> 接收 FIFO 触点中断使能位<br>0: 禁止接收 FIFO 触点中断<br>1: 使能接收 FIFO 触点中断         | W/R |

## 7.2.13 SIFR子串口中断标志寄存器: (PAGE0:1000)

| 位    | 复位值 | 功能描述                              | 类型  |
|------|-----|-----------------------------------|-----|
| Bit7 | 0   | <b>FERR_INT</b> 接收 FIFO 数据错误中断标志位 | W/R |

|      |   |  |     |
|------|---|--|-----|
|      |   | 0: 无接收 FIFO 数据错误中断<br>1: 有接收 FIFO 数据错误中断                           |     |
| Bit6 | 0 | CTS_INT CTS 中断标志位<br>0: 无 CTS 中断<br>1: 有 CTS 中断                    | W/R |
| Bit5 | 0 | RTS_INT RTS 中断标志位<br>0: 无 RTS 中断<br>1: 有 RTS 中断                    | W/R |
| Bit4 | 0 | XOFF_INT XOFF 中断标志位<br>0: 无 XOFF 中断<br>1: 有 XOFF 中断                | W/R |
| Bit3 | 0 | TFEMPTY_INT 发送 FIFO 空中断标志位<br>0: 无发送 FIFO 空中断<br>1: 有发送 FIFO 空中断   | W/R |
| Bit2 |   | TFTRIG_INT 发送 FIFO 触点中断标志位<br>0: 无发送 FIFO 触点中断<br>1: 有发送 FIFO 触点中断 | W/R |
| Bit1 |   | RXOVT_INT 接收 FIFO 超时中断标志位<br>0: 无接收 FIFO 超时中断<br>1: 有接收 FIFO 超时中断  | W/R |
| Bit0 | 0 | RFTRIG_INT 接收 FIFO 触点中断标志位<br>0: 无接收 FIFO 触点中断<br>1: 有接收 FIFO 触点中断 | W/R |

## 7.2.14 TFCNT子串口发送FIFO计数寄存器: (PAGE0:1001)

| 位         | 复位值      | 功能描述              | 类型 |
|-----------|----------|-------------------|----|
| Bit7 -- 0 | 00000000 | 子串口发送 FIFO 中的数据个数 | R  |

## 7.2.15 RFCNT子串口接收FIFO计数寄存器: (PAGE0:1010)

| 位         | 复位值      | 功能描述              | 类型 |
|-----------|----------|-------------------|----|
| Bit7 -- 0 | 00000000 | 子串口接收 FIFO 中的数据个数 | R  |

## 7.2.16 FSR子串口FIFO状态寄存器: (PAGE0:1011)

| 位    | 复位值 | 功能描述  | 类型  |
|------|-----|---|-----|
| Bit7 | 0   | RFOE 子串口接收 FIFO 中数据溢出出错标志位<br>0: 无 OE 错误<br>1: 有 OE 错误  | R   |
| Bit6 | 0   | RFBI 子串口接收 FIFO 中数据有 Line-Break 错误<br>0: 无 Line-Break 错误<br>1: 有 Line-Break 错误(Rx 信号一直为 0 的状态, 包括校验位和停止位在内) | W/R |
| Bit5 | 0   | RFPE 子串口接收 FIFO 中数据帧错误标志位   | W/R |

|      |   |  |     |
|------|---|--|-----|
|      |   | 0: 无 FE 错误<br>1: 有 FE 错误                                     |     |
| Bit4 | 0 | RFPE 子串口接收 FIFO 中数据校验错误标志位<br>0: 无 PE 错误<br>1: 有 PE 错误       | W/R |
| Bit3 | 0 | RDAT 子串口接收 FIFO 空标志位<br>0: 子串口接收 FIFO 空<br>1: 子串口接收 FIFO 未空  | W/R |
| Bit2 | 0 | TDAT 子串口发送 FIFO 空标志位<br>0: 子串口发送 FIFO 空<br>1: 子串口发送 FIFO 未空  | W/R |
| Bit1 | 0 | TFULL 子串口发送 FIFO 满标志位<br>0: 子串口发送 FIFO 未空<br>1: 子串口发送 FIFO 满 | W/R |
| Bit0 | 0 | TBUSY 子串口发送 TX 忙标志位<br>0: 子串口发送 TX 空<br>1: 子串口发送 TX 忙        | W/R |

## 7.2.17 LSR子串口接收状态寄存器: (PAGE0:1100)

| 位        | 复位值 | 功能描述   | 类型 |
|----------|-----|--|----|
| Bit7 - 4 | 0   | RSV(保留位)   |    |
| Bit3     | 0   | OE 子串口接收 FIFO 中当前被读取的字节溢出出错标志位<br>0: 无 OE 错误<br>1: 有 OE 错误   | R  |
| Bit2     | 0   | BI 子串口接收 FIFO 中当前被读取字节 Line-Break 错误标志位<br>0: 无 Line-Break 错误<br>1: 有 Line-Break 错误(Rx 信号一直为 0 的状态, 包括校验位和停止位在内) | R  |
| Bit1     | 0   | FE 子串口接收 FIFO 中当前被读取字节帧错误标志位<br>0: 无 FE 错误<br>1: 有 FE 错误   | R  |
| Bit0     | 0   | PE 子串口接收 FIFO 中当前被读取字节校验错误标志位<br>0: 无 PE 错误<br>1: 有 PE 错误  | R  |

## 7.2.18 FDAT子串口FIFO数据寄存器: (PAGE0:1101)

| 位         | 复位值      | 功能描述   | 类型  |
|-----------|----------|--|-----|
| Bit7 -- 0 | 00000000 | 写操作时: 写入子串口发送 FIFO 的数据<br>读操作时: 读出子串口接收 FIFO 的数据 | W/R |

## 7.2.19 Fwcr子串口流量控制寄存器: (PAGE0:1110)

| 位         | 复位值 | 功能描述  | 类型  |
|-----------|-----|---|-----|
| Bit7      | 0   | <b>X2RF XON/XOFF</b> 可见设置<br>0: XON/XOFF 字符不可见<br>1: XON/XOFF 字符写入 FIFO, 在主机端可见 XOFF 字符   | W/R |
| Bit6 -- 4 | 000 | <b>FWM2-0</b> 流量控制模式位<br>000/100: 无流控<br>001: 硬件 CTS 流量控制<br>010: 硬件 RTS 流量控制<br>011: 硬件 RTS 和 CTS 流量控制<br>101: 接收 XON/XOFF 流控<br>110: 发送 XON/XOFF 流量控制<br>111: XON/XOFF 流量控制 | W/R |
| Bit3 - 2  | 00  | <b>FWTRIG1-0</b> 流量控制触点设置位<br>具体设置见流量控制章节   | W/R |
| Bit1      | 0   | <b>RTS</b> RTS 引脚控制位 (向该位写入数据, RTS 引脚输出相应电平输出)<br>0: RTS 引脚输出低电平<br>1: RTS 引脚输出高电平  | W/R |
| Bit0      | 1   | <b>CTS</b> CTS 引脚控制位(读该位判断 CTS 引脚电平)<br>0: CTS 引脚低电平<br>1: CTS 引脚高电平  | R   |

## 7.2.20 RS485 子串口485控制寄存器: (PAGE0:1111)

| 位    | 复位值 | 功能描述   | 类型  |
|------|-----|--|-----|
| Bit7 | 0   | <b>RSV</b>   | R   |
| Bit6 | 0   | <b>RS485</b> 485 和 232 模式选择控制位<br>0: 232 模式(8 或 9 位数据)<br>1: 485 模式(9 位数据)   | W/R |
| Bit5 | 0   | <b>ATADD</b> 485 网络地址自动识别使能位<br>0: 禁止网络地址自动识别<br>1: 使能网络地址自动识别   | W/R |
| Bit4 | 0   | <b>DATEN</b> 网络地址可见使能控制位<br>0: 禁止网络地址可见, 网络地址不写入 FIFO<br>1: 使能网络地址可见, 网络地址写入 FIFO<br>当 ATADD=0, DATAEN 需要软件手动控制<br>当 ATADD=1, DATAEN 由硬件自动控制 | W/R |
| Bit3 | 0   | <b>RSV</b>   | R   |
| Bit2 | 0   | <b>RSV</b>   | R   |
| Bit1 | 0   | <b>RTSEN</b> RTS 输出使能位<br>0: 禁止  | W/R |

|      |   |  |     |
|------|---|--|-----|
|      |   | 1: 使能  |     |
| Bit0 | 0 | RTSINV RTS 默认输出控制位<br>0: RTS 引脚默认输出高电平<br>1: RTS 引脚默认输出低电平 | W/R |

## 7.2.21 BAUD1子串口波特率配置寄存器高字节: (PAGE1:0100)

| 位         | 复位值      | 功能描述                      | 类型  |
|-----------|----------|---------------------------|-----|
| Bit7 -- 0 | 00000000 | BAUD[15:8] 子串口波特率配置寄存器高字节 | W/R |

## 7.2.22 BAUD0子串口波特率配置寄存器低字节: (PAGE1:0101)

| 位         | 复位值      | 功能描述                     | 类型  |
|-----------|----------|--------------------------|-----|
| Bit7 -- 0 | 00000000 | BAUD[7:0] 子串口波特率配置寄存器低字节 | W/R |

## 7.2.23 PRES子串口波特率配置寄存器小数部分: (PAGE1:0110)

| 位         | 复位值  | 功能描述      | 类型  |
|-----------|------|-----------|-----|
| Bit7 -- 4 | 0000 | RSV       | R   |
| Bit3 -- 0 | 0000 | PRES[3:0] | W/R |

## 7.2.24 RFTL子串口接收FIFO触发中断寄存器: (PAGE1:0111)

| 位         | 复位值      | 功能描述         | 类型  |
|-----------|----------|--------------|-----|
| Bit7 -- 0 | 00000000 | 接收 FIFO 触点控制 | W/R |

## 7.2.25 TFTL子串口发送FIFO触发中断寄存器: (PAGE1:1000)

| 位         | 复位值      | 功能描述         | 类型  |
|-----------|----------|--------------|-----|
| Bit7 -- 0 | 00000000 | 发送 FIFO 触点控制 | W/R |

## 7.2.26 FWTH子串口FIFO流量触发寄存器高字节: (PAGE1:1001)

| 位         | 复位值      | 功能描述        | 类型  |
|-----------|----------|-------------|-----|
| Bit7 -- 0 | 00000000 | 流量控制触点设置高字节 | W/R |

## 7.2.27 FWTL子串口FIFO流量触发寄存器低字节: (PAGE1:1010)

| 位         | 复位值      | 功能描述        | 类型  |
|-----------|----------|-------------|-----|
| Bit7 -- 0 | 00000000 | 流量控制触点设置低字节 | W/R |

## 7.2.28 XON1子串口XON1寄存器: (PAGE1:1011)

| 位         | 复位值      | 功能描述        | 类型  |
|-----------|----------|-------------|-----|
| Bit7 -- 0 | 00000000 | XON 特殊字符寄存器 | W/R |

## 7.2.29 XOFF1子串口XOFF1寄存器: (PAGE1:1100)

| 位         | 复位值      | 功能描述         | 类型  |
|-----------|----------|--------------|-----|
| Bit7 -- 0 | 00000000 | XOFF 特殊字符寄存器 | W/R |

## 7.2.30 SADR RS485地址寄存器：(PAGE1:1101)

| 位         | 复位值      | 功能描述     | 类型  |
|-----------|----------|----------|-----|
| Bit7 -- 0 | 00000000 | RS485 地址 | W/R |

## 7.2.31 SAEN RS485地址使能寄存器：(PAGE1:1110)

| 位         | 复位值      | 功能描述                               | 类型  |
|-----------|----------|------------------------------------|-----|
| Bit7 -- 0 | 00000001 | RS485 地址使能。(RS485 网络地址为 SADR&SAEN) | W/R |

## 7.2.32 RTSPLY RS485 RTS延时寄存器：(PAGE1:1111)

| 位         | 复位值      | 功能描述                      | 类型  |
|-----------|----------|---------------------------|-----|
| Bit7 -- 0 | 00000000 | RTS 输出延时控制(单位：1/(波特率*16)) | W/R |

## 8. 全局功能描述

### 8.1 复位

WK2204为低电平复位。

各寄存器的复位值见7.2寄存器表中所列。

复位期间及复位后，各子串口处于禁止收发状态。当子串口处于联网模式下时，该特性使得该子串口所在的子节点在上电、复位期间不会对联网的其它节点产生干扰。

每个子串口可独立实现软件复位。

### 8.2 时钟选择

WK2204可以选择使用晶振时钟作为芯片的时钟源。注意：需要和晶振并联1M的启动电阻。见图8.2.1

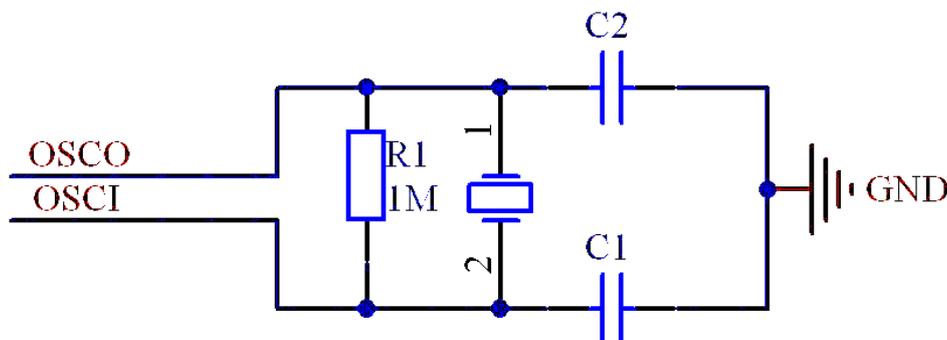


图 8.2.1 WK2204 时钟电路

## 8.3 中断控制

WK2204有两级中断：子串口中断及全局中断。当IRQ引脚指示有中断时，可以通过读取全局中断寄存器GIFR以判断当前中断的类型，然后去读取相应的中断状态寄存器，以确定当前的中断源。

WK2204的中断结构如下图所示：

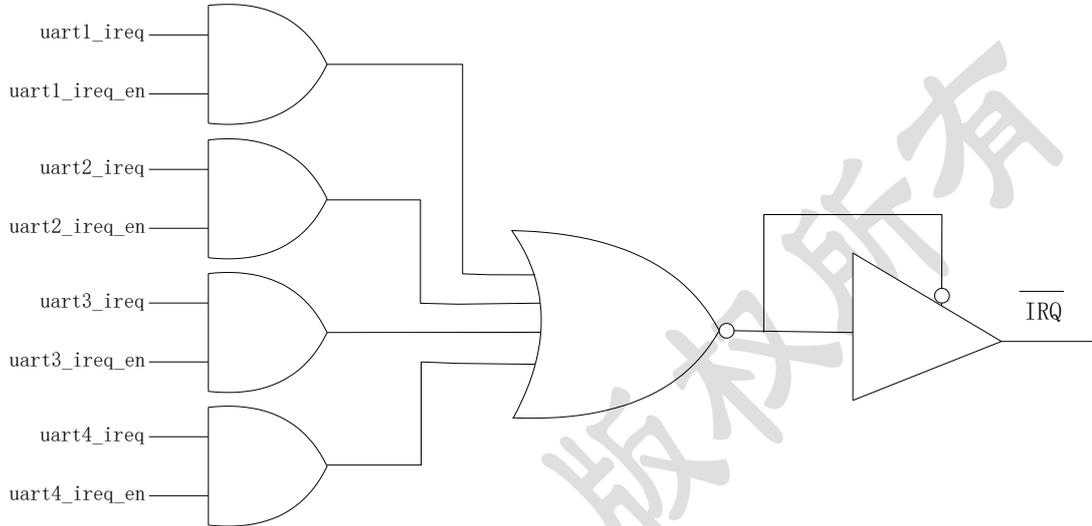


图 8.3.1 WK2204 中断结构图

WK2204的每个子串口都有独立的中断系统，包括：FIFO数据错误中断，CTS中断，RTS中断，XOFF中断，发送FIFO空中断，发送FIFO触发点中断，接收FIFO超时中断，接收FIFO触发点中断。

当任意一个中断使能后，满足中断条件就会产生相应的中断。

### 8.3.1 FIFO数据错误中断

FIFO数据错误中断表明当前接收FIFO中有一个或以上的数据错误，产生错误的条件包括OE（数据溢出错误），FE（数据帧错误），和PE（奇偶校验错），BE（Line-Break错误）。

一旦接收FIFO中有出错数据，当读取FSR寄存器后，该中断消失；也可通过清除错误数据来清除该中断。

### 8.3.2 CTS中断

在自动或手动硬件流量控制模式下，CTS信号从0变为1时或者从1变到0，将产生该中断；当读取CTS中断标志寄存器后将清除该中断。

### 8.3.3 RTS中断

在自动或手动硬件流量控制模式下，当RTS信号从0变为1时，都可以产生该中断。

在自动硬件流量控制模式下，当接收FIFO中的数据个数降低到设定的继续发送触发点时，该中断被清除。

手动硬件流量控制模式下，向RTS寄存器写入0将清除该中断。

### 8.3.4 发送XOFF中断

在软件自动流量控制模式下，当数据接收端准备发送XOFF字符时产生该中断。当其准备发送XON字符时该中断被清除。

### 8.3.5 发送FIFO空中断

当发送FIFO中没有数据，产生该中断。当发送FIFO中的数据个数大于设定的发送FIFO触发点时，该中断被清除。

### 8.3.6 发送FIFO触发点中断

当发送FIFO中的数据个数小于设定的发送FIFO触发点时，产生该中断。当发送FIFO中的数据个数大于等于设定的发送FIFO触发点时，该中断被清除。

### 8.3.7接收FIFO超时中断

当接收FIFO中数据个数小于设定的接收FIFO触发点并且RX引脚4个字节之内没有数据，产生该中断。当接收FIFO中的数据被读走或者RX继续接收数据时，该中断消失。

### 8.3.8 接收FIFO触发点中断

当接收FIFO中的数据个数大于等于设定的发送FIFO触发点时，产生该中断。当接收FIFO中的数据个数小于设定的发送FIFO触发点时，该中断被清除。

## 8.4 红外模式操作

WK2204的主串口和子串口都可以设置成为红外通信模式。当WK2204的UART设置为IrDA模式时，可以与符合SIR红外通信协议标准的设备通信，或者直接应用于光隔离通信中。

在IrDA模式下，一位数据的周期缩短到普通UART一位数据的3/16，小于1/16波特周期的脉冲将被作为干扰而忽略。

### 8.4.1 红外接收操作

在红外数据接收的时序和普通UART数据接收的对应图 如图8.4.1所示：IRX为接收到的红外数据信号，RX为通过红外数据解码后的数据。解码后的数据与IRX上的数据有1

个BIT（ $16 \times \text{CLOCK}$ ）的延迟。接收模式下，与普通UART不同的是，RX在脉冲的中间进行一次采样（区别与普通UART的3次采样），IrDA解码器将IRX上的 $3/16$ 波特周期的脉冲解码为数据0，持续低电平解码为数据1。

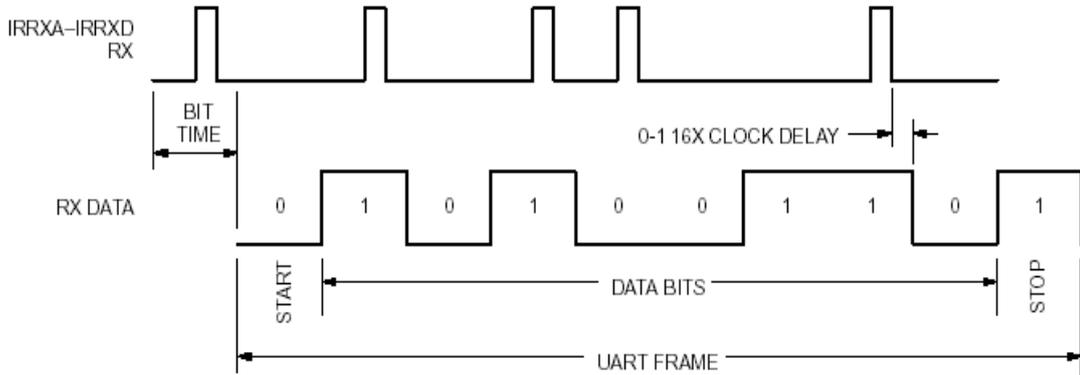


图 8. 4. 1 红外接收时序

#### 8. 4. 2 红外发送操作

红外数据发送和普通UART数据发送的对应图如图8. 4. 2所示，TX为普通UART数据发送时序，IRTXA为红外发送时序。当发送数据0时，红外编码器将产生一个 $3/16$ 位宽的脉冲通过TX发送。当发送数据0时，保持低电平不变。

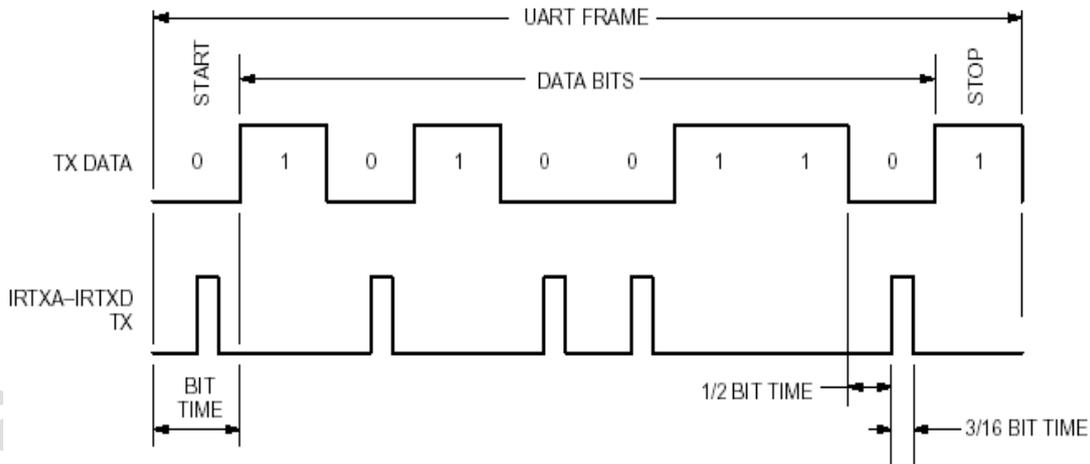


图 8. 4. 2 红外发送时序

### 8. 5 可编程波特率发生器

WK2204的主串口和子串口采用相同的独立可编程波特率发生器。该波特率发生器产生 $16 \times$ 系统时钟的分频系数，分频率可以通过软件设置。

## 8.5.1 常见波特率和晶振对照表

下表给出了在不同系统时钟频率下的串口波特率设置表：

表8.5.1

| BAUD<br>BAUD[15-0] | PRES | 波特率<br>Fosc =<br>1.8432MHz | 波特率<br>Fosc =<br>3.6864MHz | 波特率<br>Fosc =<br>7.3728MHz | 波特率<br>Fosc =<br>11.0592MHz | 波特率<br>Fosc =<br>14.7456MHz |
|--------------------|------|----------------------------|----------------------------|----------------------------|-----------------------------|-----------------------------|
| 0X0002             | 0X00 | 38400                      | 76800                      | 153600                     | 230400                      | 307200                      |
| 0X0005             | 0X00 | 19200                      | 38400                      | 76800                      | 115200                      | 153600                      |
| 0X000b             | 0X00 | 9600                       | 19200                      | 38400                      | 57600                       | 76800                       |
| 0X0017             | 0X00 | 4800                       | 9600                       | 19200                      | 28800                       | 38400                       |
| 0X002f             | 0X00 | 2400                       | 4800                       | 9600                       | 14400                       | 19200                       |
| 0X005f             | 0X00 | 1200                       | 2400                       | 4800                       | 7200                        | 9600                        |
| 0X00bf             | 0X00 | 600                        | 1200                       | 2400                       | 3600                        | 4800                        |
| 0X017f             | 0X00 | 300                        | 600                        | 1200                       | 1800                        | 2400                        |
| 0X0000             | 0X00 | 115200                     | 230400                     | 460800                     | 691200                      | 921600                      |
| 0X0001             | 0X00 | 57600                      | 115200                     | 230400                     | 345600                      | 460800                      |
| 0X0003             | 0X00 | 28800                      | 57600                      | 115200                     | 172800                      | 230400                      |
| 0X0007             | 0X00 | 14400                      | 28800                      | 57600                      | 86400                       | 115200                      |
| 0X000f             | 0X00 | 7200                       | 14400                      | 28800                      | 43200                       | 57600                       |
| 0X001f             | 0X00 | 3600                       | 7200                       | 14400                      | 21600                       | 28800                       |
| 0X003f             | 0X00 | 1800                       | 3600                       | 7200                       | 10800                       | 14400                       |
| 0X007f             | 0X00 | 900                        | 1800                       | 3600                       | 5400                        | 7200                        |

## 8.5.2 任意晶振下波特率计算

$$\text{计算公式: } \frac{f_s}{(\text{baud} * 16)} = \text{Reg}$$

注： $f_s$ 为系统时钟， $\text{baud}$ 为需要设置的波特率，Reg为计算结果（通常需要精确到小数点后两位）

Reg整数部分减一并换算成16进制写入{BAUD1, BAUA0}；如果还有小数部分，则取小数部分第一位写入PRES。如果没有小数部分，只需把整数部分写入{BAUD1, BAUA0}，PRES写入0即可。

例1： $f_s=11.0592\text{MHz}$ ， $\text{baud}=115200$ 。根据公式可得Reg=6。那么填入寄存器的数据为：BAUD1=0X00；BAUD0=0X05；PRES=0X00。

例2： $f_s=12\text{MHz}$ ， $\text{baud}=115200$ 。根据公式可得Reg=6.51（精确到小数点后两位）。那么填入寄存器的数据为BAUD1=0X00；BAUD0=0X05；PRES=0X05。

## 8.5.3 高波特率计算

| BAUD<br>BAUD [15-0] | PRES | 波特率<br>Fosc =<br>8MHz | 波特率<br>Fosc =<br>16MHz | 波特率<br>Fosc =<br>24MHz | 波特率<br>Fosc =<br>32MHz |
|---------------------|------|-----------------------|------------------------|------------------------|------------------------|
| 0X0000              | 0X00 | 500K                  | 1M                     | 1.5M                   | 2M                     |
| 0X0001              | 0X00 | 250K                  | 500K                   | 750K                   | 1M                     |
| 0X0003              | 0X00 | 125K                  | 250K                   | 375K                   | 500K                   |

## 8.6 数据格式设置

### 8.6.1 校验模式

WK2204的UART能提供强制校验，计算校验和无校验的数据格式，通过LCR（子串口配置寄存器）进行设置：

#### 强制校验模式

WK2204支持强1校验，强0校验和用户指定校验模式。在这种模式下，校验设置仅影响数据发送，数据接收将忽略奇偶校验。

在RS-485模式下，推荐使用强制校验模式，在该模式下，可以很方便的区分数据和地址。

#### 计算校验模式

WK2204支持1校验、0校验，奇校验、偶校验模式。在该模式下，接收和发送的数据都进行奇偶校验计算。

### 8.6.2 数据长度

WK2204支持1或2位停止位模式，主串口通过GMUT.GSTPL设置，子串口通过LCR.STPL设置。

## 8.7 休眠和自动唤醒

WK2204支持休眠和自动唤醒模式，并且每个子串口可以单独进行休眠设置。

休眠条件：

- 1、SCR.SLEEPEN=1
- 2、接收FIFO和发送FIFO要为空
- 3、RX上没有数据接收和TX上没有数据发送
- 4、子串口没有任何中断

当同时满足上述条件，并保持以上状态4个Bytes的时间后，子串口自动进入休眠状态，子串口的时钟自动关闭，以降低功耗。此时通过读取GRST来判断子串口是否进入

sleep状态。

当子串口进入休眠状态后，满足下面条件之一，休眠中的子串口可自动唤醒，此时通过读取GRST来判断子串口是否被唤醒。

唤醒条件：

- 1、RX上开始接收数据
- 2、向子串口发送FIFO写数据
- 3、CTS引脚电平变化

## 8.8 FIFO触点设置

WK2204支持每个子串口设置不同的触发点，接收FIFO和发送FIFO可以独立设置不同触发点。设置触点的方法有两种：

- 1、配置固定触点：通过FCR寄存器中的TFTRIG[1:0]和RFTRIG[1:0]位固定编程来配置触发点位置。
- 2、配置任意触点：通过设置TFTL和RFTL两个寄存器来设置任意触发点位置(注意:该方式的优先级高于固定触发点设置，即:当TFTL/RFTL中的值不为0时，中断触点值就是TFTL/RFTL中的值，而FCR中的TFTRIG[1:0]/RFTRIG[1:0]配置出于无效状态)。

具体配置见表8.8.1和8.8.2:

表8.8.1 发送中断触点配置表

| TFTL[7:0] | FCR[7:6]<br>TFTRIG[1:0] |   | 发送中断触点值 |
|-----------|-------------------------|---|---------|
| =0        | 0                       | 0 | 8       |
| =0        | 0                       | 1 | 16      |
| =0        | 1                       | 0 | 24      |
| =0        | 1                       | 1 | 30      |
| !=0       | X                       | X | TFTL    |

表8.8.2 接收中断触点配置表

| RFTL[7:0] | FCR[5:4]<br>RFTRIG[1:0] |   | 接收中断触点值 |
|-----------|-------------------------|---|---------|
| =0        | 0                       | 0 | 8       |
| =0        | 0                       | 1 | 16      |
| =0        | 1                       | 0 | 24      |
| =0        | 1                       | 1 | 28      |
| !=0       | X                       | X | RFTL    |

## 9. SPI 接口模式操作

### 9.1 SPI与主机的连接

如图9.1.1所示 SPI接口包括如下四个信号：

MISO：SPI从器件数据输出。

MOSI：SPI从器件数据输入。

SCLK：SPI串行时钟。

SSEL：SPI片选（从属选择）。

WK2204与主机的连接如图9.1.1所示：

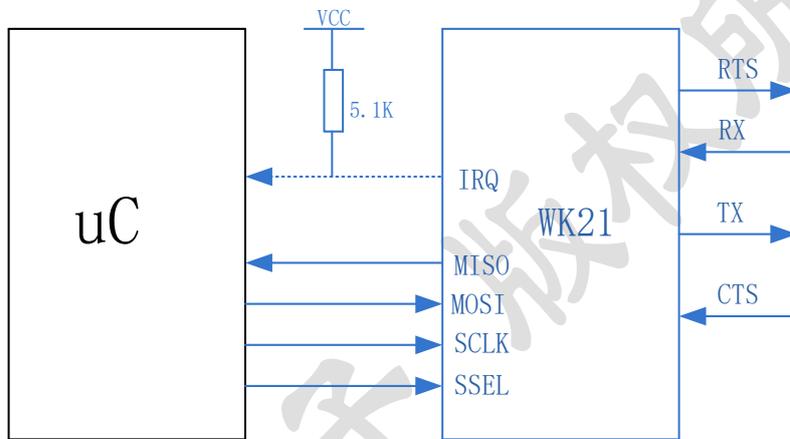


图 9.1.1 SPI 与主机连接图

### 9.2 SPI接口的操作时序

WK2204工作在SPI同步串行通信的从机模式下，支持SPI模式0标准。为实现主机和WK2204的通信，在主机端需要设置CPOL=0 (SPI时钟极性选择位), CPHA=0 (SPI时钟相位选择位)。

WK2204 SPI接口的操作时序如下所示：

写寄存器操作时序如图9.2.1所示：先写入一个命令字节（Command Byte），随后写入相应的数据字节，数据字节的寄存器地址自动增加。

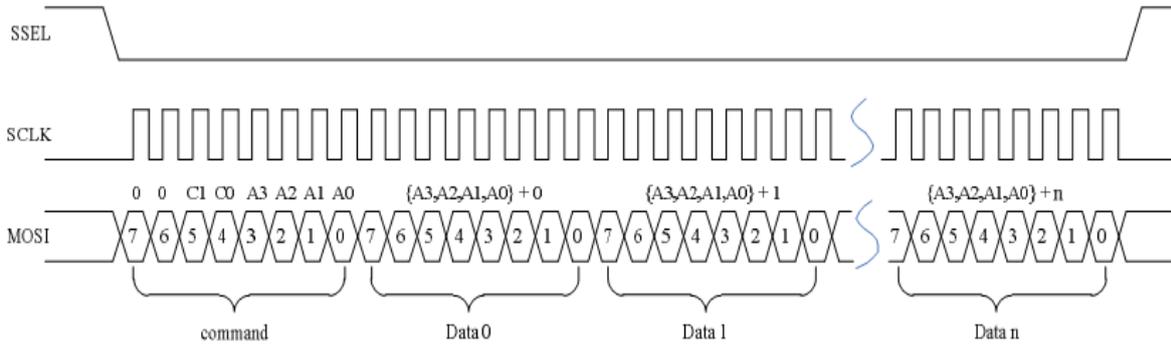


图 9. 2. 1 SPI 写寄存器时序图

读寄存器操作时序如图9. 2. 2所示：先写入一个命令字节（Command Byte），随后芯片MISO线上会返回相应的数据字节。返回数据字节的寄存器地址自动增加。

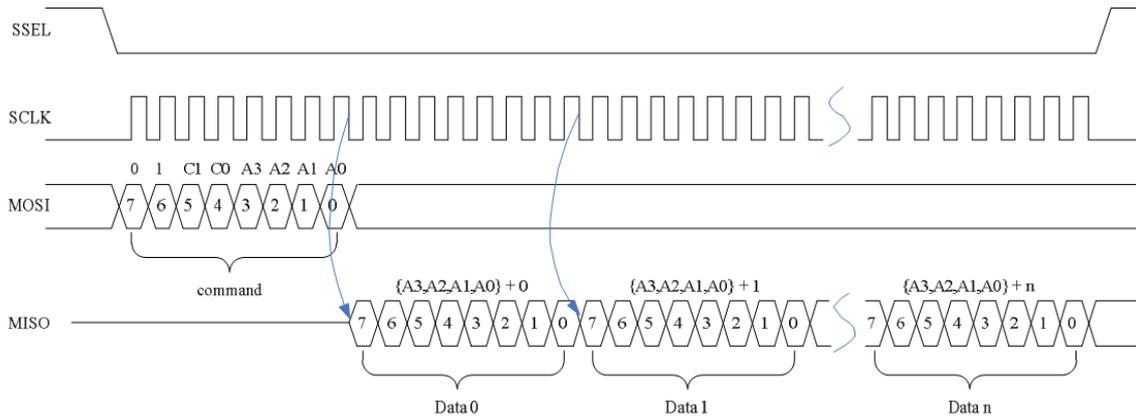


图 9. 2. 2 SPI 读寄存器时序图

写FIFO操作时序如图9. 2. 3所示：先写入一个命令字节（Command Byte），随后再写入相应的数据字节。FIFO地址自动增加。

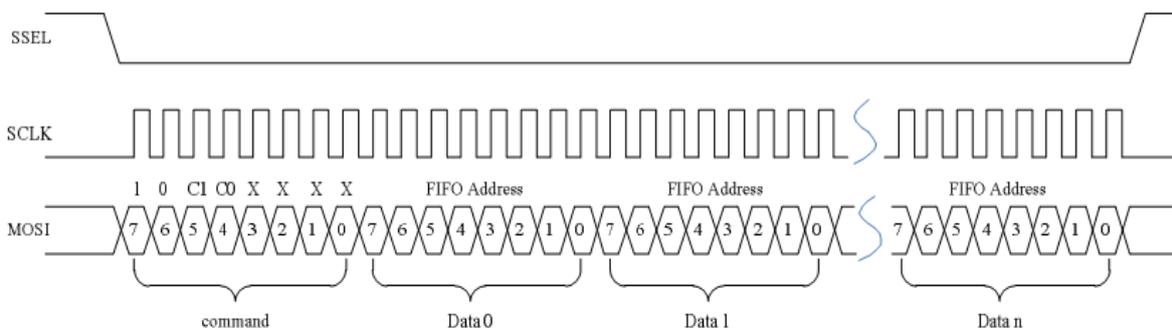


图 9. 2. 3 SPI 写 FIFO 时序图

读FIFO操作时序如图9.2.4所示：先写入一个命令字节（Command Byte），随后芯片MISO线上会返回相应的数据字节。FIFO地址自动增加。

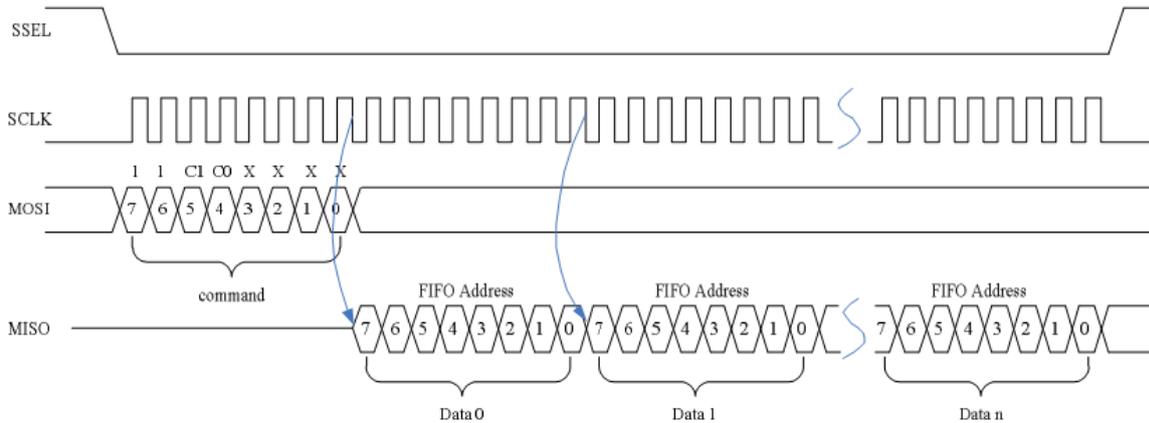


图 9.2.4 SPI 读 FIFO 时序图

### 9.3 SPI总线通信协议描述

#### 9.3.1. SPI写寄存器

| SPI  | 控制字节 CMD |    |    |    |    |    |    |    | 数据字节 DB(写 N 个数据字节, 寄存器地址自动增加) |     |     |     |     |     |     |     |
|------|----------|----|----|----|----|----|----|----|-------------------------------|-----|-----|-----|-----|-----|-----|-----|
| BIT  | 7        | 6  | 5  | 4  | 3  | 2  | 1  | 0  | 7                             | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| MOSI | 0        | 0  | C1 | C0 | A3 | A2 | A1 | A0 | D7t                           | D6t | D5t | D4t | D3t | D2t | D1t | D0t |
| MISO | HZ       | HZ | HZ | HZ | HZ | HZ | HZ | HZ | HZ                            | HZ  | HZ  | HZ  | HZ  | HZ  | HZ  | HZ  |

#### 9.3.2. SPI读寄存器

| SPI  | 控制字节 CMD |    |    |    |    |    |    |    | 数据字节 DB(读 N 个数据字节, 寄存器地址自动增加) |     |     |     |     |     |     |     |
|------|----------|----|----|----|----|----|----|----|-------------------------------|-----|-----|-----|-----|-----|-----|-----|
| BIT  | 7        | 6  | 5  | 4  | 3  | 2  | 1  | 0  | 7                             | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| MOSI | 0        | 1  | C1 | C0 | A3 | A2 | A1 | A0 | X                             | X   | X   | X   | X   | X   | X   | X   |
| MISO | HZ       | HZ | HZ | HZ | HZ | HZ | HZ | HZ | D7t                           | D6t | D5t | D4t | D3t | D2t | D1t | D0t |

#### 9.3.3. SPI写FIFO

| SPI  | 控制字节 CMD |    |    |    |    |    |    |    | 数据字节 DB(写 N 个数据字节到 {C1C0} 的 FIFO, FIFO 地址自动增加) |     |     |     |     |     |     |     |
|------|----------|----|----|----|----|----|----|----|--|-----|-----|-----|-----|-----|-----|-----|
| BIT  | 7        | 6  | 5  | 4  | 3  | 2  | 1  | 0  | 7  | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| MOSI | 1        | 0  | C1 | C0 | X  | X  | X  | X  | D7t  | D6t | D5t | D4t | D3t | D2t | D1t | D0t |
| MISO | HZ       | HZ | HZ | HZ | HZ | HZ | HZ | HZ | HZ   | HZ  | HZ  | HZ  | HZ  | HZ  | HZ  | HZ  |

### 9.3.4. SPI读FIFO

| SPI  | 控制字节 CMD |    |    |    |    |    |    |    | 数据字节 DB(从{C1C0}的FIFO读N个数据字节, FIFO地址自动增加) |     |     |     |     |     |     |     |     |
|------|----------|----|----|----|----|----|----|----|--|-----|-----|-----|-----|-----|-----|-----|-----|
|      | BIT      | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| MOSI | 1        | 1  | C1 | C0 | X  | X  | X  | X  | X  | X   | X   | X   | X   | X   | X   | X   | X   |
| MISO | HZ       | HZ | HZ | HZ | HZ | HZ | HZ | HZ | HZ                                       | D7t | D6t | D5t | D4t | D3t | D2t | D1t | D0t |

说明:

- 1、C1 C0: 子串口通道号 00~11分别对应子串口1到子串口4
- 2、A3-A0: 子串口寄存器地址
- 3、D7t···D0t: 8位数据字节

## 10. UART 接口模式操作

### 10.1 UART接口与主机的连接

当WK2204的主接口为UART时,仅需要RX, TX连接主机。采用标准的UART协议进行通信。主接口UART可以实现波特率自适应。上电复位后,先向WK2204写入0x55, WK2204可以根据写入的数据自动测得此时MCU的波特率并把主接口UART的波特率锁定,以后就以此波特率进行通信;如果主接口需要再次更换波特率,需要对芯片硬件复位,然后再次进行波特率测试和锁定。

WK2204与主机的接口如图10.1.1所示:

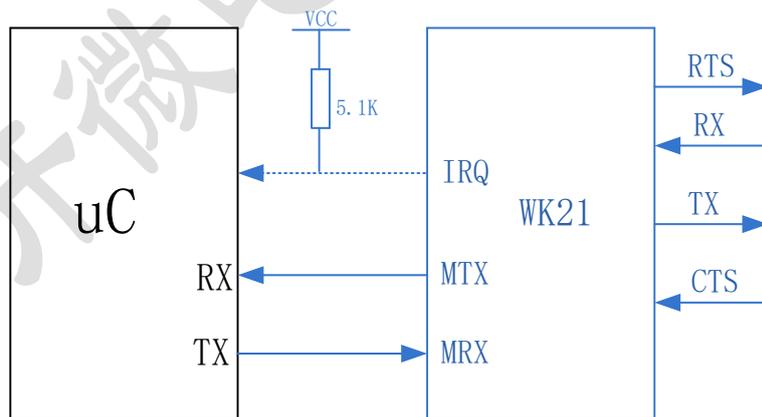


图 10.1.1 UART 接口与主机连接图

### 10.2 主UART接口的操作时序

写操作时,先向WK2204的MRX写入一个命令字节(Command Byte),随后写入相应的数据字节,其操作时序(无校验模式)如图10.2.1所示:

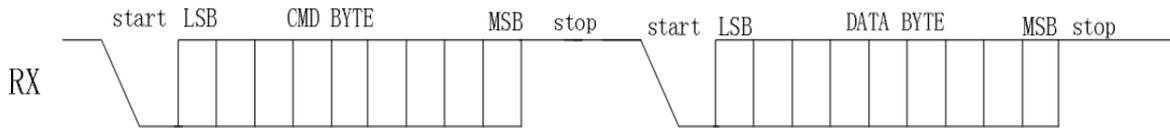


图 10.2.1 UART 主接口写操作时序

读操作时，先向WK2204的RX写入命令字节，相应的数据字节从MTX读取，其操作时序（无校验模式）如图10.2.2所示

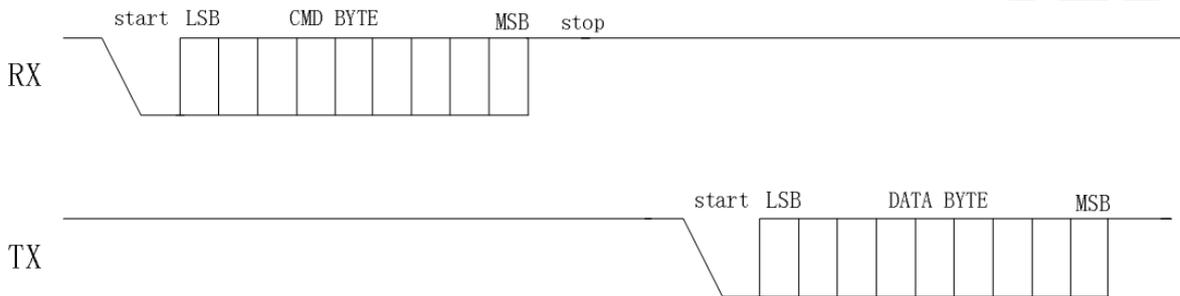


图 10.2.2 UART 主接口读操作时序

### 10.3 主UART通信传输协议描述

#### 10.3.1. 写寄存器

| UART | 控制字节 CMD |   |    |    |    |    |    |    | 1 个数据字节 DB(下行) |    |    |    |    |    |    |    |
|------|----------|---|----|----|----|----|----|----|----------------|----|----|----|----|----|----|----|
| BIT  | 7        | 6 | 5  | 4  | 3  | 2  | 1  | 0  | 7              | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| TX   | 0        | 0 | C1 | C0 | A3 | A2 | A1 | A0 | D7             | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| RX   |          |   |    |    |    |    |    |    |                |    |    |    |    |    |    |    |

#### 10.3.2. 写FIFO: (多字节写入)

| UART | 控制字节 CMD |   |    |    |    |    |    |    | [N3 N2 N1 N0] 个数据字节 DB(下行) |    |    |    |    |    |    |    |
|------|----------|---|----|----|----|----|----|----|----------------------------|----|----|----|----|----|----|----|
| BIT  | 7        | 6 | 5  | 4  | 3  | 2  | 1  | 0  | 7                          | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| TX   | 1        | 0 | C1 | C0 | N3 | N2 | N1 | N0 | D7                         | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| RX   |          |   |    |    |    |    |    |    |                            |    |    |    |    |    |    |    |

#### 10.3.3. 读寄存器

| UART | 控制字节 CMD |   |    |    |    |    |    |    | 1 个数据字节 DB(上行) |    |    |    |    |    |    |    |
|------|----------|---|----|----|----|----|----|----|----------------|----|----|----|----|----|----|----|
| BIT  | 7        | 6 | 5  | 4  | 3  | 2  | 1  | 0  | 7              | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| TX   | 0        | 1 | C1 | C0 | A3 | A2 | A1 | N0 |                |    |    |    |    |    |    |    |
| RX   |          |   |    |    |    |    |    |    | D7             | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

## 10.3.4. 读FIFO: (多字节读取)

| UART | 控制字节 CMD |   |    |    |    |    |    |    | [N3 N2 N1 N0]个数据字节 DB(上行) |    |    |    |    |    |    |    |
|------|----------|---|----|----|----|----|----|----|---------------------------|----|----|----|----|----|----|----|
| BIT  | 7        | 6 | 5  | 4  | 3  | 2  | 1  | 0  | 7                         | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| TX   | 1        | 1 | C1 | C0 | N3 | N2 | N1 | N0 |                           |    |    |    |    |    |    |    |
| RX   |          |   |    |    |    |    |    |    | D7                        | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

说明:

- 1、C1,C0: 子串口通道号, 00~11分别对应子串口1到子串口4。
- 2、A3,A2,A1,A0:子串口寄存器地址;
- 3、N3,N2,N1,N0:写入/读取FIFO的数据字节个数;当其为0000时,表明后接1个数据字节;当其为1111时,表明后接16个数据字节;
- 4、向子串口读/写数据有两种方法:
  - a) 读/写寄存器方式, 对子串口FIFO寄存器FDAT (1111) 进行读/写操作, 一次只能读/写一个字节
  - b) 读/写FIFO方式, 对接收/发送FIFO直接进行读/写操作, 一次最多可以读写16个连续数据

## 10.4 主UART接口红外操作模式

当主串口IR引脚接高电平时, WK2204主UART工作在红外模式下, 主UART与主机的通信遵从红外通信协议, 其操作时序参见8.4红外模式操作。

当主串口IR引脚接低电平时, WK2204工作在普通模式下。

## 11. IIC 接口总线模式操作

两线IIC总线包含一根串行数据线SDA和一根串行时钟线SCL。当总线处于空闲状态时, 两根线都通过上拉电阻拉到正电源电压。每一个设备都有一个独立的地址。如图11:

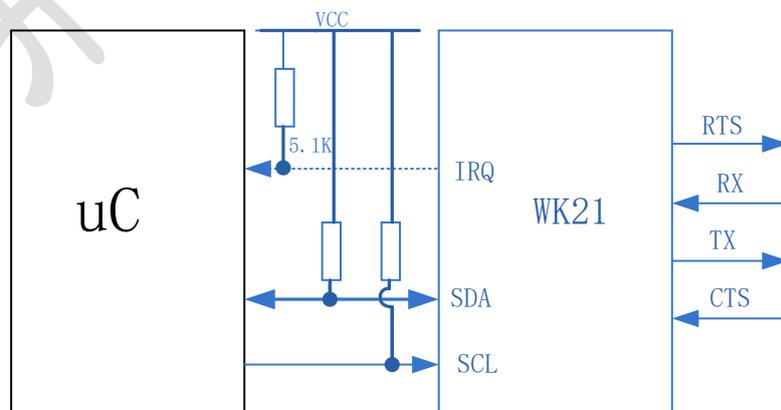


图 11 IIC 主接口示意图

### 11.1 数据传输

每一位数据都是通过一个始终脉冲进行传输的。在SCL为高的周期SDA线上的数据必须保持稳定。在此时改变SDA线上的数据会被认为是控制信号。当SCL为高的时候SDA线数据由高到低的跳变表示一个起始位，一个由低到高的跳变表示一个停止位。总线在起始位以后被认为处于忙的状态；在停止位后被认为处于空闲状态。

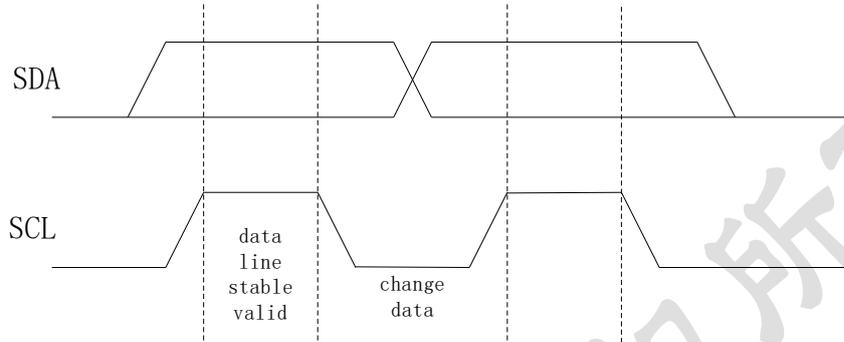


图 11.1.1 数据传输

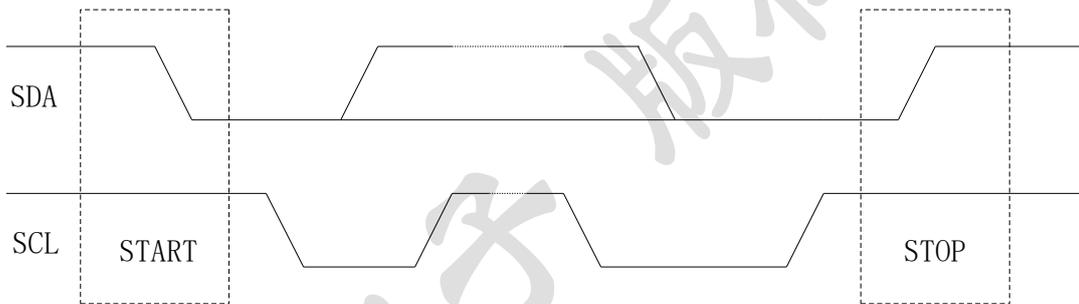


图 11.1.2 起始位和停止位

在起始位和停止位之间的由主机到从机的数据必须是8bit(比特)长，高位在前并且必须有一个应答位。与应答位相匹配的时钟是由主机产生的。当主机释放总线时，应答的设备必须在应答周期内拉低SDA线。

### 11.2 主IIC接口的操作时序

#### 11.2.1. IIC写寄存器

写寄存器操作时序如图11.2.1所示，先写入一个命令字节（Command Byte），随后写入寄存器地址字节，最后写入相应的数据字节，数据字节的寄存器地址会自动增加。

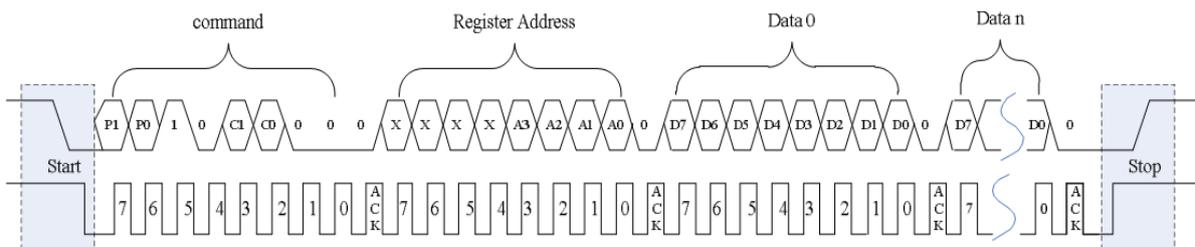


图 11.2.1 IIC 写寄存器操作时序

### 11.2.2. IIC读寄存器

读寄存器操作时序如图11.2.2所示：IIC读寄存器操作分两次完成。第一次先写入一个命令字节（Command Byte），随后写入寄存器地址字节，完成本次写入操作。紧接着开始第二次操作，先写入一个命令字节（Command Byte），随后读入相应的数据字节，寄存器地址自动增加。

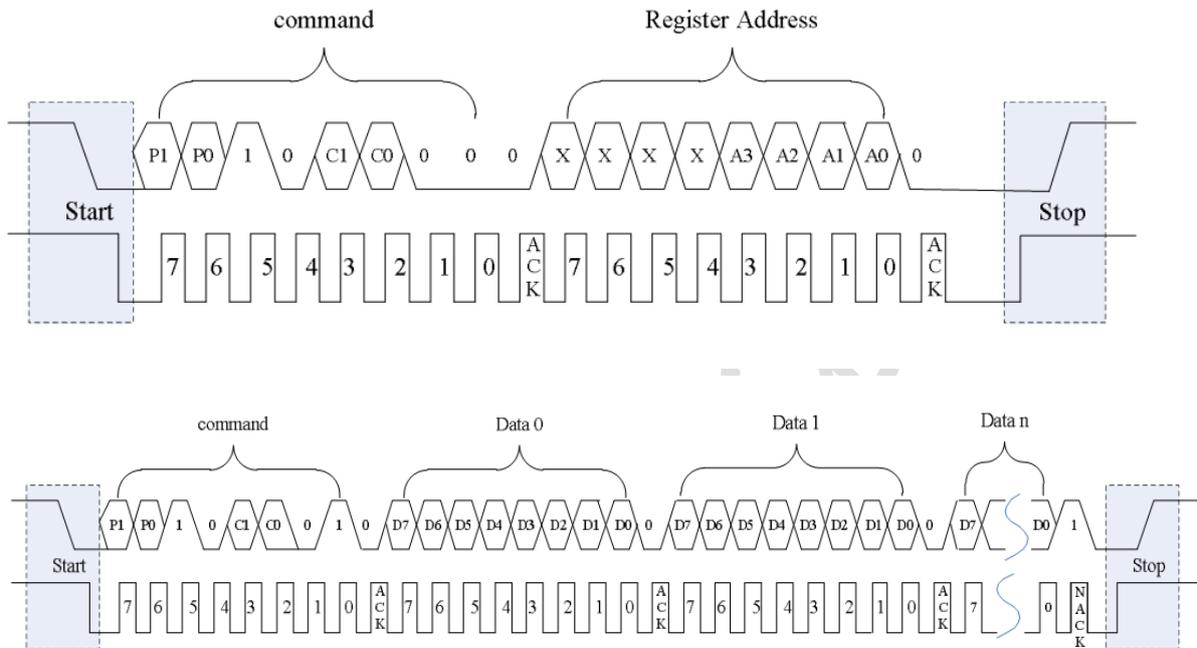


图 11.2.2 IIC 读寄存器操作时序

### 11.2.3. IIC写FIFO

写FIFO操作时序如图11.2.3所示，先写入一个命令字节（Command Byte），随后写入N个字节数据，FIFO地址自动增加。

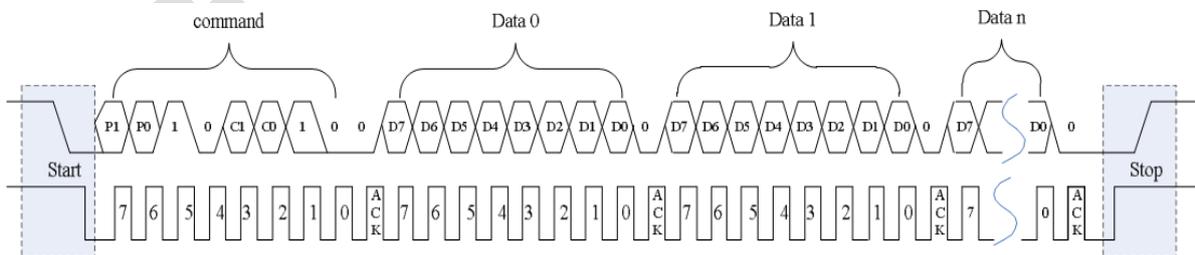


图 11.2.3 IIC 写 FIFO 操作时序

### 11.2.4. IIC读FIFO

读FIFO操作时序如图11.2.4所示，先写入一个命令字节（Command Byte），随后读取N个数据字节，FIFO地址自动增加。

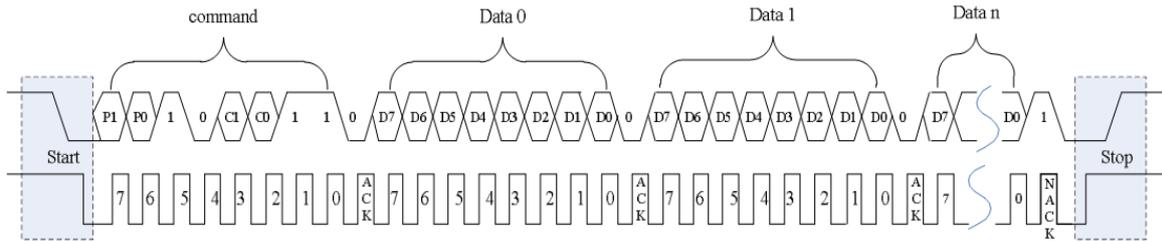


图 11.2.4 IIC 读 FIFO 操作时序

### 11.3 地址

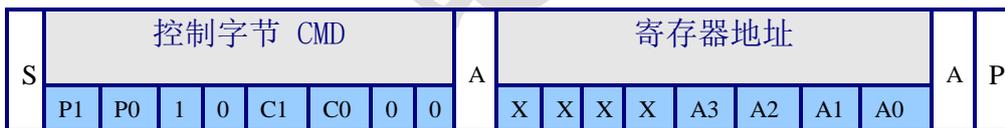
每一个挂在总线上的设备都必须有自己独特的地址。数据在总线上传输以前，主机在总线上发送从机的地址以开始一次传输。所有从机都会比较地址，如果网络里有地址相同当然会应答主机的地址请求。地址在起始位以后传输的第一个字节的高2位传输。每个器件的地址由A1A0引脚控制，那么编程中只需要P1P0的值和IA1、IA0的值对应即可。

### 11.4 传输协议

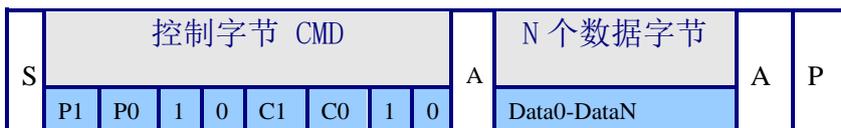
#### 11.4.1. 写寄存器



#### 11.4.2. 读寄存器



#### 11.4.3. 写FIFO



#### 11.4.4. 读FIFO

|   |          |    |   |   |    |    |   |   |   |                  |  |  |  |  |  |  |  |   |         |  |  |  |  |  |  |  |   |   |   |
|---|----------|----|---|---|----|----|---|---|---|------------------|--|--|--|--|--|--|--|---|---------|--|--|--|--|--|--|--|---|---|---|
| S | 控制字节 CMD |    |   |   |    |    |   |   | A | N-1 个数据字节        |  |  |  |  |  |  |  | A | 第N个数据字节 |  |  |  |  |  |  |  | N | A | P |
|   | P1       | P0 | 1 | 0 | C1 | C0 | 1 | 1 |   | Data0-Data (N-1) |  |  |  |  |  |  |  |   | DataN   |  |  |  |  |  |  |  |   |   |   |

说明:

- 1、注意：当MCU不需要继续接收WK2204的数据时，那么在接收完当前字节后，不需要发出应答，只需给出应答时钟即可。
- 2、C1, C0: 子串口通道号，00~11分别对应子串口1到子串口4
- 3、A3, A2, A1, A0: 子串口寄存器地址
- 4、P1, P0: 为器件地址，这个由芯片引脚IA1、IA0控制
- 5、S: 起始位      P: 停止位      A: 应答位      NA: 无应答信号

## 12. 子串口操作描述

### 12.1 子串口使能/禁止

WK2204允许独立使能或禁止每个子串口通道。

在使用中可以禁止不使用的子串口通道。

子串口通道只有处在使能状态才能接收和发送数据。

### 12.2 收发FIFO控制

WK2204提供了独立的256级FIFO接收和发送FIFO。（子串口FIFO控制寄存器）进行设置。

#### 12.2.1 发送FIFO 触发点操作

WK2204为每个通道提供独立的可编程发送FIFO触发点设置，以产生相应的发送FIFO触发点中断。

当发送FIFO触发点中断使能时，发送FIFO中的数据数目小于设定的触发点时产生相应中断。

#### 12.2.2 接收FIFO触发点操作

WK2204为每个通道提供独立的可编程接收FIFO触发点设置，以产生相应的接收FIFO触发点中断。

当接收FIFO触发点中断使能时，接收FIFO中的数据数目大于设定的触发点时产生相应中断。

#### 12.2.3 发送FIFO的使能/禁止

复位后，发送FIFO处于禁止状态。如果希望将数据写入发送FIFO，需要首先使能发送FIFO。

发送FIFO中的数据是否发送，取决于相应的子通道UART是否使能。一旦相应子通道UART处于使能状态，则发送FIFO中的数据将会立即发送，否则，发送FIFO中的数据将不会被发送直到相应的子通道被使能。

#### 12.2.4 接收FIFO的使能/禁止

复位后，接收FIFO处于禁止状态。如果希望接收子串口数据，需要首先使能相应的子串口通道及其接收FIFO。只有相应的UART和接收FIFO使能后，接收到的数据才能写入接收FIFO存储。

如果子串口通道使能而接收FIFO禁止，子串口能接收数据，但数据不会写入接收FIFO而被忽略。

#### 12.2.5 发送FIFO清空

当FCR中发送FIFO清空位（TFRST）被置1时，该子通道发送FIFO中的数据将被清空，发送FIFO计数器和指针都将清零。

TFRST位被置1后，将会在一个时钟后被硬件自动清0。

#### 12.2.6 接收FIFO清空

当FCR中接收FIFO清空位（RFRST）被置1时，该子通道接收FIFO中的数据将被清空，接收FIFO计数器和指针都将清零。

RFRST位被置1后，将会在一个时钟后被硬件自动清0。

#### 12.2.7 发送FIFO 计数器

WK2204用一个8位寄存器来反应当前发送FIFO中的数据数目：当一个字节的数据写入发送FIFO后，发送FIFO计数器自动加1；当一个发送FIFO中的数据被发送后，发送FIFO计数器自动减1。

注意：当发送FIFO计数器为255（11111111）时，如果再写入一个数据则计数器变为0（00000000）。当发送FIFO计数器为1（00000001）时，发送一个数据之后则计数器也变为0（00000000）。因此，当发送FIFO计数器为0时，表明发送FIFO满或者空，在这种情况下，需要结合子串口状态寄存器（FSR）中的相关状态位进行判断。

#### 12.2.8 接收FIFO计数器

WK2204用一个8位寄存器来反应当前接收FIFO中的数据数目：当一个字节的数据写入接收FIFO后，接收FIFO计数器自动加1；当一个接收FIFO中的数据被读取后，接收FIFO计数器自动减1。

注意：当接收FIFO计数器为255（11111111）时，如果再接收一个数据则计数器变为0（00000000）。当接收FIFO计数器为1（00000001）时，读取一个数据之后则计数器也变为0（00000000）。因此，当接收FIFO计数器为0时，表明接收FIFO满或者空，在这种情况下，需要结合子串口状态寄存器（FSR）中的相关状态位进行判断。

## 12.3 流量控制

WK2204提供硬件流量控制，软件流量控制和手动流量控制三种模式可选择。硬件流量控制通过CTS和RTS引脚实现流量控制，可以减少软件开销并提高系统效率。软件流量控制通过XON和XOFF可编程特殊字符实现流量控制操作。相关操作通过FWCR（子串口流量控制寄存器）设置。

在RS485模式下，该功能被禁止。

### 12.3.1 触发点控制

当WK2204设置为自动软件/硬件流量控制时：

当FWTH==0时，FWCR中的FWTRIG1—0用于设置暂停发送触发点，当接收FIFO中的数据个数达到暂停发送触发点时，WK2204将发出暂停发送信号，以通知发送端暂停发送数据。

当FWTH!=0时，FWCR中的FWTRIG1—0用于设置继续发送触发点，在暂停发送状态下，主机口可以通过读取数据操作读取接收FIFO中的数据，当接收FIFO中的数据个数等于设置的继续发送触发点时，WK2204将通知发送端继续发送数据。

当FWTH! =0时，停止发送触点由FWTH的值决定！

当FWTH! =0时，继续发送触点由FWTL的值决定！具体值见表12.3.1

设置时，需要保证暂停发送触发点大于继续发送触发点的数值。WK2204不对该条件做自动判断。

表12.3.1

| FWTH<br>[7:0] | FWTRIG[1:0] |   | RTS<br>Go<br>High or<br>send<br>XOFF | FWTL<br>[7:0] | RFTRIG[1:0] |   | RX<br>Go<br>Low or<br>send<br>XON |
|---------------|-------------|---|--------------------------------------|---------------|-------------|---|-----------------------------------|
| =0            | 0           | 0 | 8                                    | =0            | 0           | 0 | 0                                 |
| =0            | 0           | 1 | 16                                   | =0            | 0           | 1 | 7                                 |
| =0            | 1           | 0 | 24                                   | =0            | 1           | 0 | 15                                |
| =0            | 1           | 1 | 30                                   | =0            | 1           | 1 | 23                                |
| !=0           | X           | X | FWTH                                 | !=0           | X           | X | FWTL                              |

### 12.3.2 自动软件流量控制操作

当WK2204工作在自动软件流量控制模式时，子串口通道通过RX发送和TX接收XOFF和XON字符实现软件流量控制，无需其它控制线。XON和XOFF字符可以通过全局寄存器中的XON和XOFF寄存器设置。

在软件流量控制模式下，传输的数据字节中不能出现XON和XOFF字符，否则将会被作为XON和XOFF控制字符，因此在软件流量控制下，需要对数据中的XON和XOFF字符进行相应的转义处理。

#### 12.3.2.1 XON/XOFF发送操作

在自动软件流量控制模式下，一旦数据接收端接收FIFO中数据的个数达到设定的触发点时，为防止接收FIFO溢出，WK2204将自动通过TX发送一个XOFF字符，数据发送端收到该XOFF字节后，发送完当前字节后即暂停数据发送。

发送端暂停数据发送后，接收端的主机接口读取接收FIFO中的数据以释放接收FIFO空间，当接收FIFO中数据的个数减少到继续发送触发点时，接收端向发送端发送一个XON字符，发送端接收到该字符后，将恢复数据发送。

相关的操作时序如下图所示，XON/XOFF字符的数值由上层软件设置：

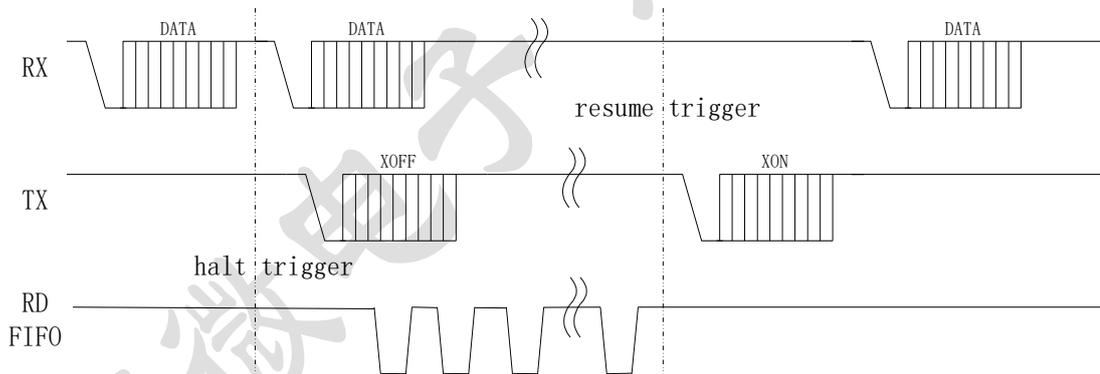


图 12.3.2 XON/XOFF 发送操作

#### 12.3.2.2 XON/XOFF接收操作

在软件流量控制模式下，WK2204接收到数据后，首先会与XOFF中的数据进行比较，当接收到XOFF字符时，在发送完当前字节后即暂停数据发送。

数据暂停发送状态下，接收到XON字符后，将恢复数据发送。

#### 12.3.2.3 XON/XOFF可见设置

在软件流量控制模式下，特殊字符XON/XOFF可以通过设置FWCR（子串口流量控制寄存器）XVEN位，使之在主机端为可见或不可见。

当设置为可见时，XON和XOFF字符作为数据写入接收FIFO。

当设置为不可见时，XON和XOFF字符将作为控制字符不被写入接收FIFO。

### 12.3.3 自动硬件流量控制

当WK2204的子串口工作在自动硬件流量控制模式时，包含自动RTS控制和自动CTS控制。分别通过硬件自动设置RTS信号和判断CTS信号来实现硬件流量控制。

典型的硬件流量控制的通过器件A的RTS连接器件B的CTS，器件A的CTS连接器件B的RTS，将器件A和B都设置为硬件自动流量控制模式即可实现硬件的自动流量控制。其连接示意图如下所示：

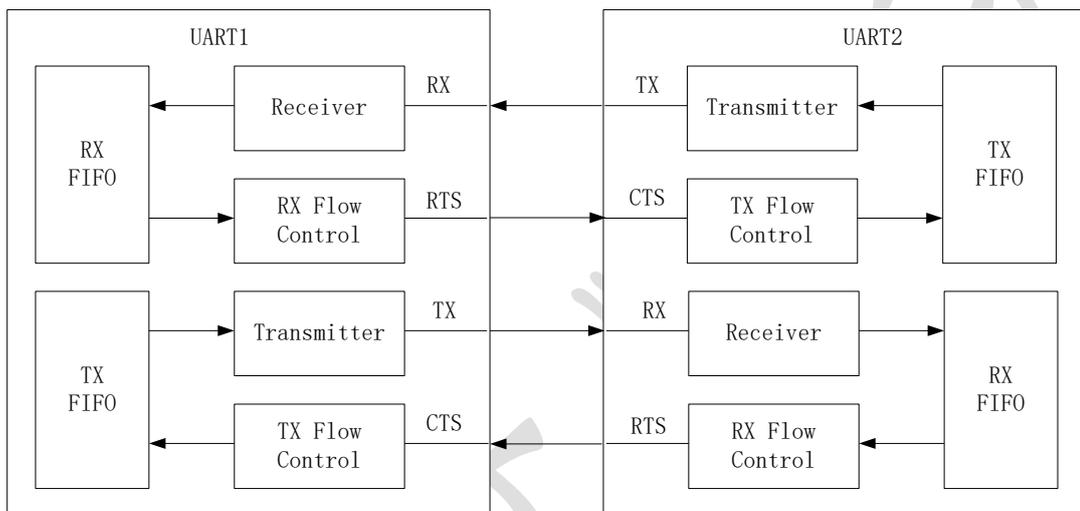


图 12.3.3.1 硬件流量控制示意图

在硬件自动流量控制模式下，一旦数据接收端接收FIFO中数据的个数达到设定的触发点时，为防止接收FIFO溢出，接收端将自动拉高RTS，数据发送端的相应的CTS变高，数据接收端检测到CTS变高后，将发送完当前字节后即暂停数据发送。

发送端暂停数据发送后，接收端的主机接口读取接收FIFO中的数据以释放接收FIFO空间，当接收FIFO中数据的个数减少到继续发送触发点时，接收端的CTS自动变为低电平，发送端相应的RTS变为低电平，发送端检测到RTS为低后，将恢复数据发送。

下图显示了硬件流量控制下的时序操作：

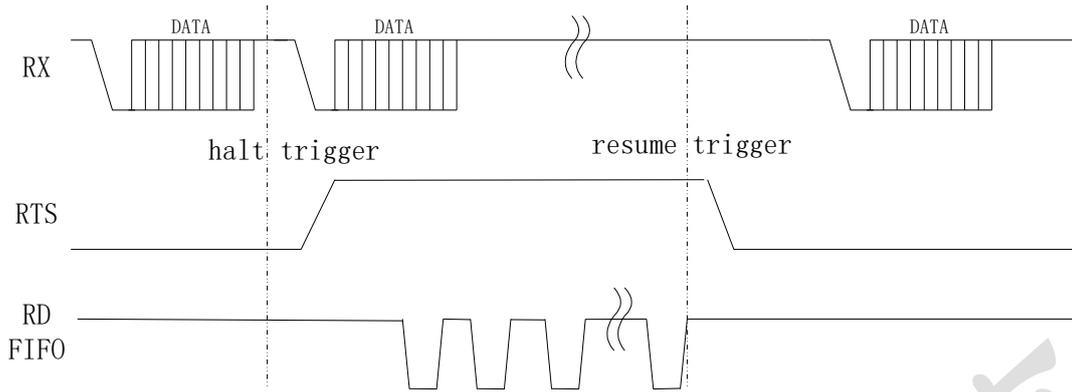


图 12.3.3.2 硬件流量操作时序图

### 12.3.4 手动硬件流量控制

当WK2204的子串口工作在手动模式下，可以通过手动写RTS寄存器拉高或拉低RTS引脚信号。

在该模式下，其它的操作与硬件自动流量控制一样，只是RTS由相应的寄存器控制。手动设置RTS为1可以暂停数据发送端发送数据，设置RTS为0则数据发送端继续发送数据。

## 12.4 RS485操作

WK2204的子串口支持RS485自动收发控制模式和网络地址识别模式。

### 12.4.1 RS485自动收发模式

在RS485自动收发模式下，流量控制将被禁止。RTS信号用于控制RS485收发器的自动收发控制。

RTS的默认电平可以设置，在RS485.RSTEN使能的情况下，通过RS485.RTSINV来调整RTS引脚输出默认电平。默认电平通常是指485芯片处于接收状态时的电平。默认电平确定后，在我们发送数据的时候，RTS引脚就会也相反的电平来控制485芯片进入发送状态，在发送数据完成后，RTS引脚会根据RTSDLY来确定恢复到默认电平的时间。

在RS485自动收发模式下，WK2204和485的收发器的连接如图：

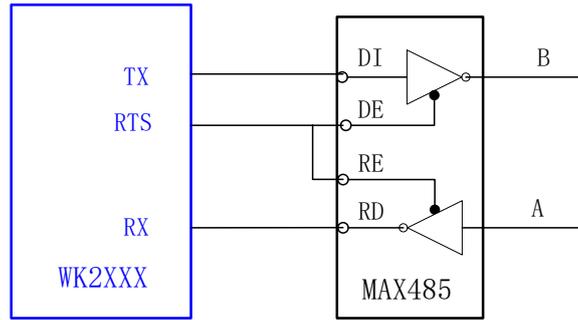


图 12.4.1 RS485 连接图

### 12.4.2 网络地址识别模式

网络地址识别模式必须工作在RS485模式下，每个UART有一个唯一的网络地址，WK2204提供了两个8位寄存器来设置网络地址，SADR为网络地址，SAEN为网络地址使能控制位，即实际的网络地址为SADR&SAEN。

网络地址识别模式可以分为自动网络地址识别和手动网络地址识别。

RS485. ATADD=1时，自动网络地址识别功能使能，WK2204对接收到的数据进行自动识别。

如果检测到地址字节，会报奇偶校验错误中断，然后WK2204会把接收到的地址和自身地址进行比对，如果和设置的子串口地址一致，RS485. DATEN会自动置1，该地址和接下来的数据字节都会进入接收FIFO，知道下次的地址字节检测。如果地址和子串口网络地址不一致，RS485. DATEN会自动置0，该地址和数据都不会进入接收FIFO。

当RS485. ATADD=0时，手动网络地址检测功能开启。

如果检测到地址字节，会报奇偶校验错误中断，然后该地址字节会进入FIFO，这时需要上位机读取该字节并判断与上位机软件程序地址是否一致。如果一致，软件需立刻将RS485. DATEN置1，这样接下来的数据才会进入FIFO，否者数据不会进入FIFO。

## 13. 参数指标

### 13.1 WK2204的静态参数

除非特别说明，满足： $VCC = (2.5V \pm 0.2V)$  或  $(3.3 \pm 0.3V)$  或  $(5V)$ ； $-40^{\circ}C$  到  $+85^{\circ}C$ ；

| 符号              | 说明    | 条件                        | VCC=2.5V |     | VCC=3.0V |     | VCC=5.0V |     | 单位 |
|-----------------|-------|---------------------------|----------|-----|----------|-----|----------|-----|----|
|                 |       |                           | 最小       | 最大  | 最小       | 最大  | 最小       | 最大  |    |
| 电源              |       |                           |          |     |          |     |          |     |    |
| VCC             | 电源电压  |                           | 2.3      | 2.7 | 3.0      | 3.6 | 4.5      | 5.0 | V  |
| ICC             | 工作电流  | 3.6864MHz 晶振              | 0.8      | 2   | 1        | 2   | 2        | 3   | mA |
| ICCSL           | 休眠电流  | 无负载                       | 150      | -   | 200      | -   | 460      | -   | uA |
| 输入逻辑信号          |       |                           |          |     |          |     |          |     |    |
| V <sub>IH</sub> | 输入高电平 |                           | 1.8      | 5.0 | 2.0      | 5.0 | 3.6      | 5.0 | V  |
| V <sub>IL</sub> | 输入低电平 |                           | -        | 0.6 | -        | 0.9 | -        | 1.1 | V  |
| I <sub>IL</sub> | 输入漏电流 | V <sub>I</sub> =5.0 or 0V | -        | ±10 | -        | ±10 | -        | ±10 | uA |
| C <sub>I</sub>  | 输入电容  |                           | -        | 5   | -        | 5   | -        | 5   | pF |
| 输出逻辑信号          |       |                           |          |     |          |     |          |     |    |
| V <sub>OH</sub> | 输出高电平 | I <sub>OH</sub> =3mA      | 1.9      | -   | 2.4      | -   | 4.5      | -   | V  |
| V <sub>OL</sub> | 输出低电平 | I <sub>OL</sub> =-3mA     | -        | 0.4 | -        | 0.4 | 0        | 0.4 | V  |
| I <sub>OL</sub> | 输出漏电流 |                           | -        | ±10 | -        | ±10 | -        | ±10 | uA |
| C <sub>O</sub>  | 输出电容  |                           | -        | 5   | -        | 5   | -        | 5   | pF |

### 13.2 WK2204的动态参数

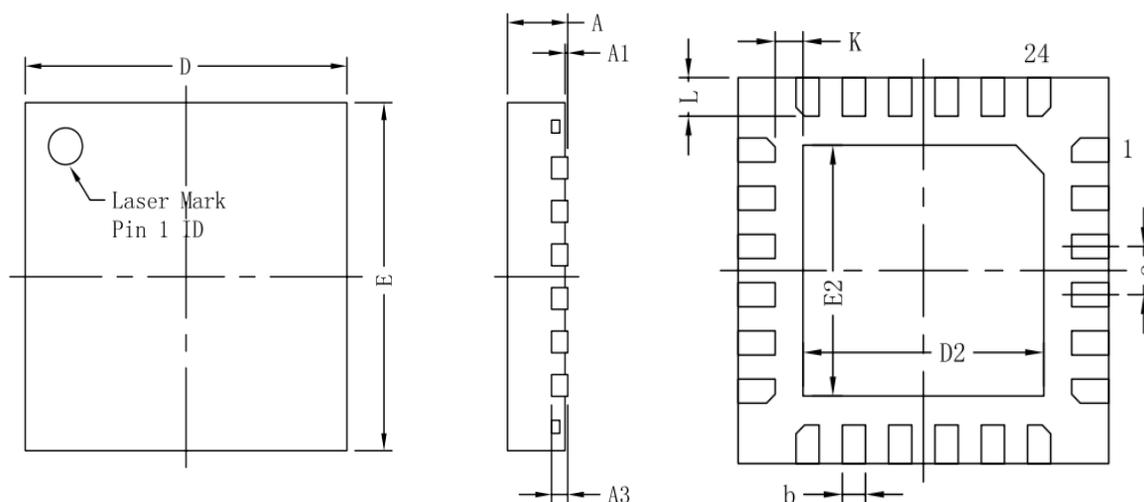
| 符号               | 说明   | 条件 | VCC=2.5V |    | VCC=3.0V |    | VCC=5.0V |    | 单位  |
|------------------|------|----|----------|----|----------|----|----------|----|-----|
|                  |      |    | 最小       | 最大 | 最小       | 最大 | 最小       | 最大 |     |
| F <sub>OSI</sub> | 晶振频率 |    | -        | 16 | -        | 24 | -        | 32 | MHz |

### 13.3 WK2204的极限参数

| 符号               | 说明   | 条件 | 最小   | 最大   | 单位 |
|------------------|------|----|------|------|----|
| VCC              | 电源电压 |    | -0.5 | 5    | V  |
| V <sub>I</sub>   | 输入电压 |    | -0.5 | +5.5 | V  |
| V <sub>O</sub>   | 输出电压 |    | -0.5 | +5.5 | V  |
| P <sub>TOL</sub> | 总功耗  |    | -    | 300  | mW |
| T <sub>O</sub>   | 工作温度 |    | -40  | +85  | °C |
| T <sub>STG</sub> | 存储温度 |    | -65  | +150 | °C |

## 14. 封装信息

WK2204采用QFN24无铅绿色封装



| 尺寸<br>标注 | 最小 (mm)  | 最大 (mm) | 尺寸<br>标注 | 最小 (mm) | 最大 (mm) |
|----------|----------|---------|----------|---------|---------|
| A        | 0.7      | 0.8     | E2       | 2.5     | 2.7     |
| A1       | --       | 0.05    | e        | 0.5TYP  |         |
| A3       | 0.203REF |         | K        | 0.2     | --      |
| b        | 0.2      | 0.3     | L        | 0.3     | 0.5     |
| D        | 3.9      | 4.1     |          |         |         |
| E        | 3.9      | 4.1     |          |         |         |
| D2       | 2.5      | 2.7     |          |         |         |

## 15. 焊接工艺

WK2204采用使用绿色环保材料，引脚采用纯锡电镀。推荐使用峰值温度小于260℃，符合无铅标准的回流焊工艺进行焊接。

所有SMD器件焊接工艺都对湿度敏感(湿度等级及条件见外包装盒)，建议在焊接前进行干燥处理。

采用手工焊接时，应首先焊接两个对角线的引脚进行固定后再焊接其它引脚。焊接温度为300℃，烙铁与引脚的接触时间控制在10秒以内。

## 16. 特别申明

本产品并非为生命保障系统、航空航天系统设计，将本产品应用于该领域而引发的一切后果，为开微电子将不承担任何责任。 为开微电子保留对产品进行性能、功能、

参数修改的权利。对于正式量产的产品，为开微电子做出的修改将以公告方式通告用户。

## 17. 版本历史

---

V1.0以前版本均为未正式公开的内部版本。

| 版本   | 发表日期    | 修改内容 |
|------|---------|------|
| V1.0 | 2014.11 | 创建文件 |
| V1.1 | 2017.09 | 封装信息 |
|      |         |      |
|      |         |      |

## 18. 联系信息

---

请访问为开微电子的网站获取我们的最新联系方式：[www.wkmic.com](http://www.wkmic.com)

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [Interface - Specialised category](#):*

*Click to view products by [WEIKAI manufacturer](#):*

Other Similar products are found below :

[MC33975ATEKR2](#) [SP508EEF-L](#) [TDA8035HN/C2/S1J](#) [TDA8026ET/C3E](#) [TC358870XBG\(NOK\)](#) [MAX13175EETU+](#)  
[MAX17841EVMINIQU#](#) [250375-4Z50-2](#) [NCN5120MNTWG](#) [C100N50Z4A](#) [SN65HVD63RGTT](#) [TDA8035HN/C1/S1J](#) [AD8192ACPZ-RL7](#)  
[AD5700-1BCPZ-R5](#) [AD5700-1BCPZ-RL7](#) [ADV3003ACPZ](#) [ADV7511KSTZ](#) [LTC4332IUFD#PBF](#) [LTC4332HUFD#PBF](#)  
[LTC4332CUFD#PBF](#) [MAX3814CHJ+T](#) [CH351Q](#) [MC33664ATL1EGR2](#) [XL1192D](#) [XD7135](#) [XR34350IL](#) [KTU1109EFAA-TR](#) [CH368L](#)  
[CH7307C-DEF](#) [DS28CM00R-A00+T](#) [DS8500-JND+T&R](#) [MAX16054AZT+T](#) [MAX16545BGPF+](#) [MAX16550GPN+](#) [MAX16550AGPN+](#)  
[MAX16550AGPN+T](#) [MAX96705AGTJ/V+](#) [MAX3816ACUE+](#) [MAX4886ETO+](#) [MAX14502AETL+](#) [SP331ET-L](#) [SP332CT-L](#) [SP338EER1-](#)  
[L](#) [SP3508EF-L](#) [SP335EER1-L](#) [XR3160ECUTR-F](#) [XR3160EIUTR-F](#) [ATECC108A-MAHDA-T](#) [MAX3814CHJ+](#) [LPC47M182-NW](#)