

第 3 章



四旋翼飞行器常用通信协议

3.1 常用通信接口协议

3.1.1 SPI

1. SPI 总线接口概述

串行外设接口(Serial Peripheral Interface, SPI)总线系统是一种同步串行外设接口,是 Motorola 公司首先在其 MC68HCXX 系列处理器上定义的。SPI 接口主要应用于 EEPROM、Flash、实时时钟、AD 转换器,以及数字信号处理器和数字信号解码器之间,允许 MCU 与各种外围设备以串行方式进行通信、数据交换^[9]。SPI 系统可直接与各个厂家生产的多种标准外围器件直接相连,一般使用 4 条线。

① 串行时钟线 SCLK。其主要作用是主设备往从设备传输时钟信号,控制数据交换的时机以及速率。

② 主机输入/从机输出数据线 MISO。也称为 Rx 通道,作为数据的入口,主要用于 SPI 设备接收数据。

③ 主机输出/从机输入数据线 MOSI。也称为 Tx 通道,作为数据的出口,主要用于 SPI 设备发送数据。

④ 从机使能信号 nSS(有的 IC 会标注为 CS,意为片选信号)。用于主设备片选从设备,使被选中的从设备能够被主设备所访问。

SPI 接口芯片带有中断信号线 INT 或 INT*,有的 SPI 接口芯片没有主机输出/从机输入数据线 MOSI。所以,SPI 系统总线只需 3~5 条数据线和控制线即可同具有 SPI

的各种 I/O 器件连接。

SPI 总线可在软件的控制下构成各种简单的或复杂的系统。例如,一个主 MCU 和几个从 MCU;几个从 MCU 相互连接构成多主机系统(分布式系统);一个主 MCU 和一个或几个从 I/O 设备。大多数情况下,使用一个 MCU 作为主机,控制数据向一个或多个从外围器件的传送,从器件只能在主机的 SCLK 信号有时钟跳变时才能传送和采集信号。

图 3.1.1 是对 SPI 设备间通信的一个简单的描述,其中的组件及其含义如下。

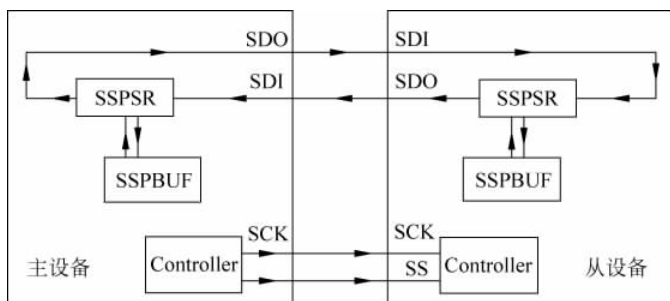


图 3.1.1 SPI 设备通信结构

① SSPBUF(Synchronous Serial Port Buffer),泛指 SPI 设备里面的内部缓冲区,一般在物理上是以 FIFO 的形式保存传输过程中的临时数据。

② SSPSR(Synchronous Serial Port Register),泛指 SPI 设备里面的移位寄存器(Shift Register),它的作用是根据设置好的数据位宽(bit-width)把数据移入或者移出 SSPBUF。

③ Controller,泛指 SPI 设备里面的控制寄存器,可以通过配置它们来设置 SPI 总线的传输模式。

SPI 典型结构组成如图 3.1.2 所示。

2. SPI 总线接口数据传送

SPI 是一个环形总线结构,其时序简单,主要是在 SCLK 的控制下,两个双向移位寄存器进行数据交换,其时序结构如图 3.1.3 所示。上升沿发送数据,下降沿接收数据。其数据的传输格式是高位(MSB)在前,低位(LSB)在后。上升沿到来的时候,SDO 上最高位的电平将被发送到从设备的寄存器中;下降沿到来的时候,SDI 上的电平将被接收到主设备的寄存器中。一个完整的传送周期是 16 位,即两个字节,因为,首先主机要发

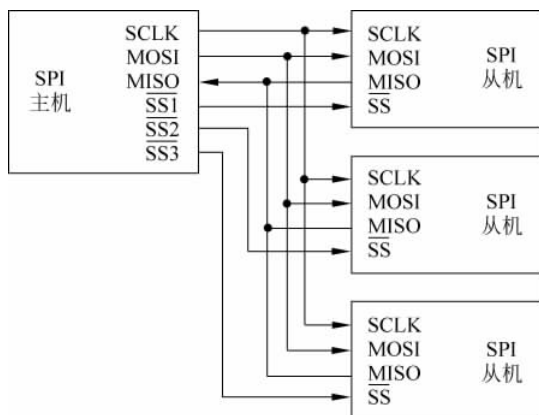


图 3.1.2 SPI 总线结构(1 主机+3 独立从机)

送命令过去,然后从机根据主机的命令准备数据,主机在下一个 8 位时钟周期才把数据读回来。

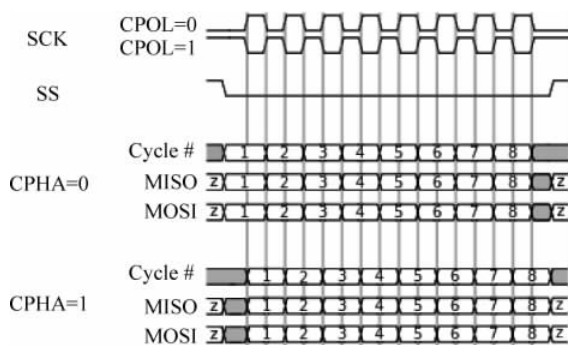


图 3.1.3 SPI 总线时序

SPI 模块为了和外设进行数据交换,根据外设工作要求,其输出串行同步时钟极性和相位可以进行配置,时钟极性(CPOL)对传输协议没有大的影响。如果 CPOL=0,串行同步时钟的空闲状态为低电平;如果 CPOL=1,串行同步时钟的空闲状态为高电平。时钟相位(CPHA)能够配置用于选择两种不同的传输协议之一进行数据传输。如果 CPHA=0,在串行同步时钟的第一个跳变沿(上升或下降)数据被采样;如果 CPHA=1,在串行同步时钟的第二个跳变沿(上升或下降)数据被采样。SPI 主模块和与之通信的外设时钟相位和极性应该一致。

因此根据 CPOL 和 CPHA 的设置, SPI 共可分为 4 种模式, 如表 3.1.1 所示。

表 3.1.1 SPI 4 种模式

SPI 模式	时钟极性(CPOL)	时钟相位(CPHA)
0	0	0
1	0	1
2	1	0
3	1	1

另一种常用的模式表示方式为如“(0,1)”即表明 CPOL=0, CPHA=1。下面给出一段用软件实现 SPI 串行通信的例程。

```
int main(void)
{
SysCtlClockSet(SYSCTL_SYSDIV_2_5 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN | SYSCTL_XTAL_10MHZ);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_SSI1);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    SysCtlGPIOAHBEnable(GPIO_PORTF_BASE);           //挂在 AHB 高性能总线上

    GPIOPinConfigure(GPIO_PF2_SSI1CLK);
    GPIOPinConfigure(GPIO_PF3_SSI1FSS);
    GPIOPinConfigure(GPIO_PF1_SSI1TX);
    GPIOPinTypeSSI(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_3|GPIO_PIN_2);
    //SSI1_BASE/时钟频率/时钟协议, moto 代表 SPI 协议/工作模式: 主机模式/位速率 4M/数据
    帧位数 16
    SSIConfigSetExpClk(SSI1_BASE, SysCtlClockGet(), SSI_FRF_MOTO_MODE_1, SSI_MODE_
    MASTER, 16000000, 16);
    SSIEnable(SSI1_BASE);
    while(1){
        SSIDataPut(SSI1_BASE, 0x2468);
    }
}
```

以上程序能够实现通过 SPI 接口发送消息, 在使用 SPI 发送消息之前, 首先要对 SPI 功能进行配置, 这也是以上程序的重点所在, SPI 初始化程序设计流程框图如图 3.1.4 所示。

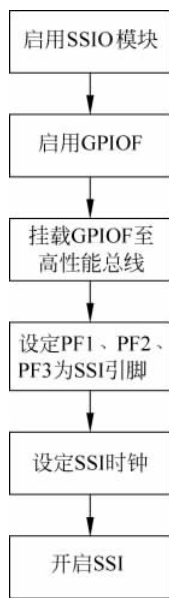
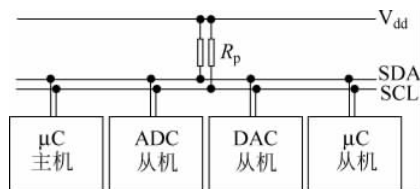


图 3.1.4 Tm4C 芯片 SPI 配置流程框图

3.1.2 I²C

1. I²C 总线接口概述

I²C(Inter-Integrated Circuit)是飞利浦(Philips)公司推出的一种高性能芯片间串行传输总线,与 SPI 不同,它仅以两根线实现了完善的全双工同步数据传送,可以极方便地构成多机系统和外围器件扩展系统^[10]。其总线的基本结构如图 3.1.5 所示,I²C 总线采用了器件地址的设置方法,通过软件寻址完全避免了器件的片选寻址的弊端,从而使硬件系统具有更简单、更灵活的扩展方法。

图 3.1.5 I²C 总线的基本结构

I²C 总线进行数据传输时只需两根信号线,一根是双向的数据线 SDA,另一根是时钟线 SCL。所有连接到 I²C 总线上的设备,其串行数据都连接到总线的 SDA 线上,而各设备的时钟线均接到总线的 SCL 线上。

I²C 总线是一个多主机总线,即一个 I²C 总线可以有一个或多个主机,总线运行由主机控制。这里所说的主机是指启动数据的传送(发起始信号)、发出时钟信号、传送结束时发出终止信号的设备。通常,主机由各类单片机或其他微处理器担当。被主机寻访的设备为从机,它可以是各种单片机或其他微处理器,也可以是其他器件,如存储器、LED 或 LCD 驱动器、A/D 或 D/A 转换器、时钟日历器件等。

I²C 总线为双向同步串行总线,因此 I²C 总线接口内部为双向传输电路。总线接口输出为开漏结构,所以总线上必须有上拉电阻。

为了进行通信,每个连接到 I²C 总线上的设备都有一个唯一的地址。主机与从机之间的数据传送可以是由主机发送到总线上的其他设备,这时主机称为发送器。从总线上接收数据的设备为接收器。

在多主机系统中,可能同时有几个主机企图启动总线传送数据。为了避免混乱,保证数据的可靠传送,任意时刻总线只能由某一台主机控制,所以,I²C 总线要通过总线裁决,来决定由哪一台主机控制总线。若有两个或两个以上的主机企图占用总线,一旦一个主机送“1”,而另一个或多个送“0”,送“1”的主机退出竞争。在竞争过程中,时钟信号是各个主机产生异步时钟信号线“与”的结果。

I²C 总线上的时钟总是对应于主机的。传送数据时,每个主机产生自己的时钟,主机产生的时钟仅在慢速的从机拉宽低电平加以改变或在竞争中失败而改变。

当总线空闲时,两根总线均为高电平。连接到总线上的器件其输出级必须是漏极或集电极开路,任意设备输出的低电平都将使总线的信号变低,即各设备的 SDA 及 SCL 都是线与的关系。

2. I²C 总线的数据传送

1) 总线上数据的有效性

在 I²C 总线上,每一位数据位的传送都与时钟脉冲相对应,逻辑“0”和逻辑“1”的信号电平取决于相应的正端电源 V_{DD} 的电压。I²C 总线进行数据传送时,在时钟信号为高电平期间,数据线上必须保持有稳定的逻辑电平状态,高电平为数据 1,低电平为数据 0。只有在时钟线低电平期间,才允许数据线上的电平状态变化,如图 3.1.6 所示。

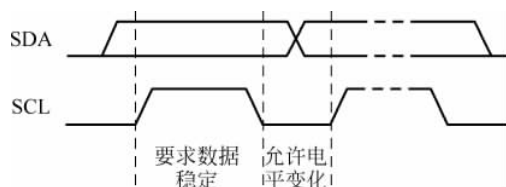


图 3.1.6 数据位的有效性规定

2) 数据传送的起始信号和终止信号

起始信号和终止信号如图 3.1.7 所示。

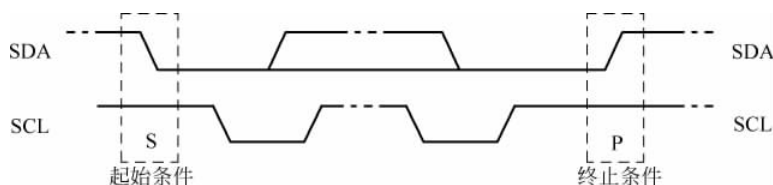


图 3.1.7 起始信号和终止信号

根据 I²C 总线协议的规定：SCL 线为高电平期间，SDA 由高电平向低电平的变化表示起始信号；SCL 线为高电平期间，SDA 线由低电平向高电平的变化表示终止信号。

起始信号和终止信号都是由主机发出的，在起始信号产生后，总线就处于被占用的状态；在终止信号产生一定时间后，总线就处于空闲状态。连接到 I²C 总线上的设备若具有 I²C 硬件接口，则很容易检测到起始信号和终止信号。对于不具备 I²C 总线接口的 MCU 来说，为了能准确检测起始信号和终止信号，必须保证在总线的—个时钟周期内对数据线至少采样两次。

从机收到—个完整的数据字节后，有可能需要完成—些其他工作，如处理内部中断服务等，可能使它无法立刻接收—个字节。这时从机可以将 SCL 线拉成低电平，从而使主机处于等待状态，直到从机准备好可以接收—个字节时，再释放 SCL 线使之—为高电平，数据传送继续进行。

3) 数据传送格式

(1) 字节传送与应答

利用 I²C 总线进行数据传送时，对传送的字节数是没有限制的，但是—个字节必须保证是 8 位长度，并且首先发送的数据位为最高位，每传送—个字节数据后都必须跟随

一位应答信号,与应答信号相对应的时钟由主机产生,主机必须在这一时钟位上释放数据线,使其处于高电平状态,以便从机在这一位上送出应答信号,如图 3.1.8 所示。

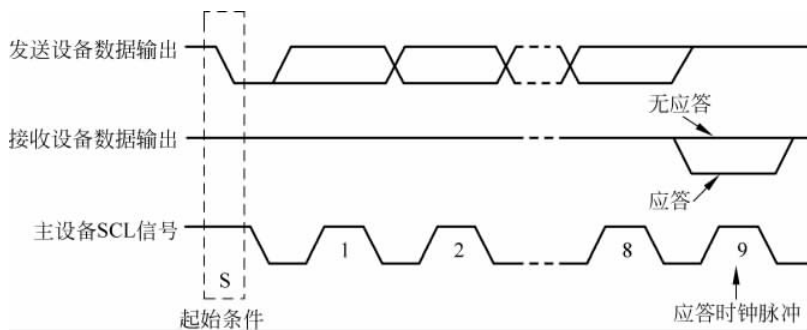


图 3.1.8 I²C 总线应答时序

应答信号在第 9 个时钟位上出现,从机输出低电平为应答信号,表示继续接收,若从机输出高电平则为非应答信号,表示结束接收。由于某种原因,从机不对主机寻址信号应答时(如从机正在进行实时性的处理工作而无法处理接收总线上的数据),它必须释放总线,将数据线置于高电平,然后由主机产生一个终止信号以结束总线的数据传送。

如果从机对主机进行了应答,但在数据传送一段时间后无法继续接收更多的数据时,从机可以通过发送非应答信号通知主机,主机则应发出终止信号来结束数据的继续传送。

当主机接收数据时,它收到最后一个数据字节后,必须向从机发送一个非应答信号,使从机释放 SDA 线,以便主机产生终止信号,从而停止数据传送。

(2) 数据传送格式

I²C 总线上传输的数据信号是广义的,既包括地址信号,又包括真正的数据信号。I²C 总线数据传送时必须遵守规定的的数据传送格式,如图 3.1.9 所示。

按照总线规约,起始信号表明一次数据传送的开始,其后为寻址字节,寻址字节由高 7 位地址和最低 1 位方向位组成,高 7 位地址是被寻址的从机地址,方向位是表示主机与从机之间的数据传送方向,方向位为“0”时表示主机发送数据“写”,方向位为“1”时表示主机接收数据“读”。在寻址字节后是将要传送的数据字节与应答位,在数据传送完后主机必须发送终止信号。但是,如果主机希望继续占用总线进行新的数据传送,则可以产生终止信号,马上再次发出起始信号对另一从机进行寻址。

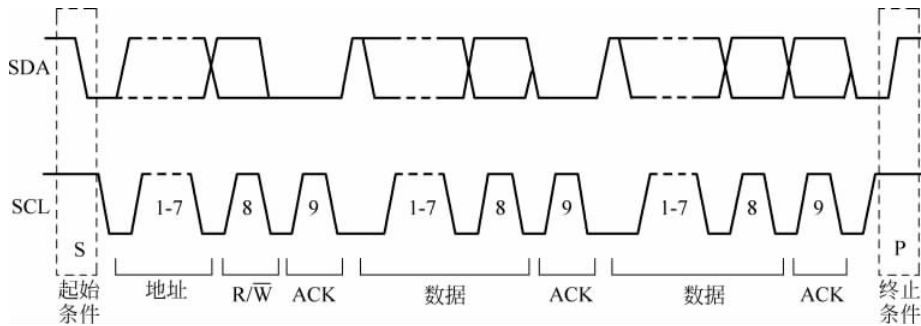


图 3.1.9 I²C 总线一次完整的数据传输格式

因此,在总线的一次数据传送过程中,可以有几种读、写组合方式。

① 主机向从机发送 n 个数据,数据传送方向在整个传送过程中不变,如图 3.1.10 所示。

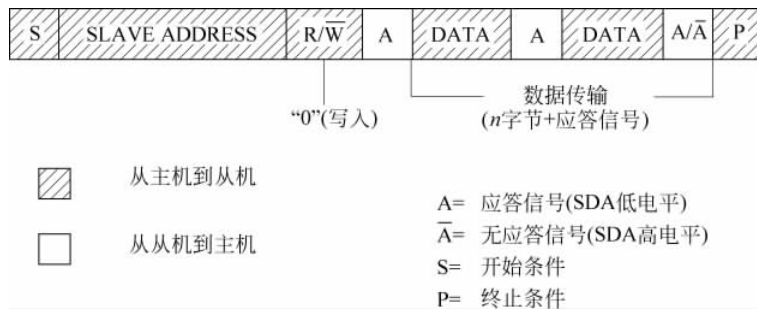


图 3.1.10 I²C 数据传输方式 1

主机在第一个字节后,立即由从机读数据,在整个过程中,除寻址字节外,都是从机发送、主机接收,如图 3.1.11 所示。

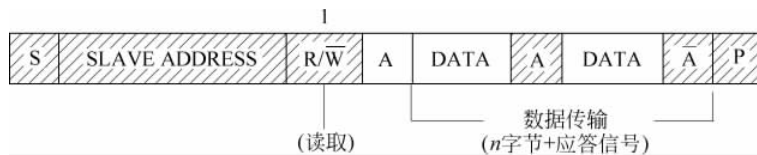
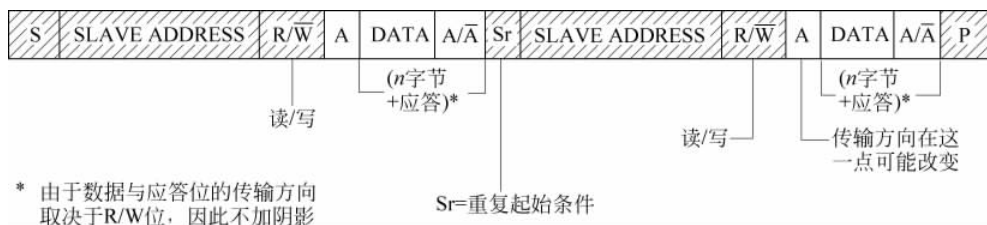


图 3.1.11 I²C 数据传输方式 2

② 在传送过程中,当需要改变传送方向时,起始信号和从机地址都被重复产生一次,但两次读/写方向位正好反相,如图 3.1.12 所示。

图 3.1.12 I²C 数据传输方式 3

由以上格式可见，无论采用哪种方式，起始信号、终止信号和地址均由主机发送，数据字节的传送方向由寻址字节中方向位规定；每个字节的传送都必须有应答信号位相随。

4) I²C 总线的寻址约定

I²C 总线是多主总线，总线上的各个主机都可以争用总线，在竞争中获胜者马上占有总线控制权。I²C 总线协议对有权使用总线的主机寻址做出了明确的规定：采用 7 位的寻址字节，寻址字节是起始信号后的第一个字节。

(1) 寻址字节的位定义

从高到低分别为 7 位数据，最低位 LSB 为数据传送方向位，如图 3.1.13 所示。主机发送地址时，总线上的每个从机都将这 7 位地址码与自己的期间地址进行比较，如果相同则认为自己正被主机寻址，根据读/写位将自己确定为接收器或发送器。

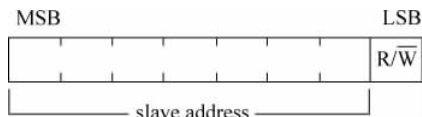


图 3.1.13 寻址字节

从机的地址是由一个固定部分和一个可编程部分组成。固定部分为器件的编号地址，表明了器件的类型，出厂时是固定的，不可更改；可编程部分为器件的呼叫地址，视硬件接线而定，引脚地址数决定了同一种器件可接入到 I²C 总线中的最大数目。如果从机为单片机，则 7 位地址为纯软件地址。

(2) 寻址字节中的特殊地址

I²C 总线地址统一由 I²C 总线委员会实行分配，其中两组编号地址 0000 和 1111 已被保留，留作特殊用途，如表 3.1.2 所示。

表 3.1.2 I²C 总线特殊地址

地址位	读/写位	用途
0000 000	0	通用广播地址
0000 000	1	起始地址
0000 001	×	CBUS 地址
0000 010	×	待定
0000 011	×	
0000 1××	×	
1111 1××	×	
1111 0××	×	10 位从机地址

① 广播地址。起始信号之后第一个字节为“0000 0000”时称为通用广播地址。广播地址用于寻访接到 I²C 总线上的所有器件,并向它们广播数据。

② 起始字节是提供给没有 I²C 总线接口的单片机查询 I²C 总线时使用的特殊字节。对于不具备 I²C 总线接口的单片机,采用软件模拟时序的方法,也可以接入 I²C 总线系统。

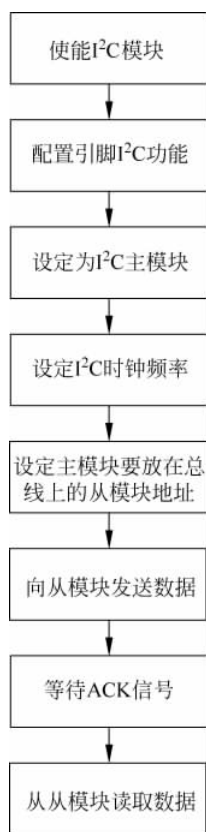
下面给出在本系统采用 Ti 飞行控制板芯片的开发板 TIVA C Launchpad 上实现 I²C 主从模块消息互通的主模块程序设计流程图(如图 3.1.14 所示)。从模块程序设计主模块类似,但不用设置时钟频率等。

3.1.3 USART

1. USART 功能概述

USART(Universal Synchronous/Asynchronous Receiver/Transmitter)是一个全双工通用同步/异步串行收发模块,该接口是一个高度灵活的串行通信设备。与 UART 相比,它增加了同步功能,能够提供主动时钟来实现同步通信。它提供了一种灵活的方法与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。USART 利用分数波特率发生器提供宽范围的波特率选择。它支持同步单向通信和半双工单线通信。它还允许多处理器通信。

接口通过 3 个引脚与其他设备连接在一起,任何 USART 双向通信至少需要两个脚,即接收数据输入(RX)和发送数据输出(TX)。USART 通信具有以下特性。

图 3.1.14 I²C 主模块程序设计流程框图

- ① 总线在发送或接收前应处于空闲状态。
- ② 一个起始位。
- ③ 一个数据字(8 位或 9 位),最低有效位在前。
- ④ 0.5、1.5 这两个的停止位,由此表明数据帧的结束。
- ⑤ 使用分数波特率发生器——12 位整数和 4 位小数的表示方法。
- ⑥ 一个状态寄存器(USART_SR)。
- ⑦ 一个数据寄存器(USART_DR)。
- ⑧ 一个波特率寄存器(USART_BRR),12 位整数和 4 位小数。
- ⑨ 一个智能卡模式下的保护时间寄存器(USART_GTPR)。

2. USART 数据传输

如图 3.1.15 所示,字长可以通过编程 USART_CR1 寄存器中的 M 位,选择成 8 位

或 9 位。在起始位期间, Tx 脚处于低电平, 在停止位期间 Tx 脚处于高电平。空闲符号被视为完全由“1”组成的一个完整的数据帧, 后面跟着包含了数据的下一帧的起始位 (“1”的位数也包括了停止位的位数)。

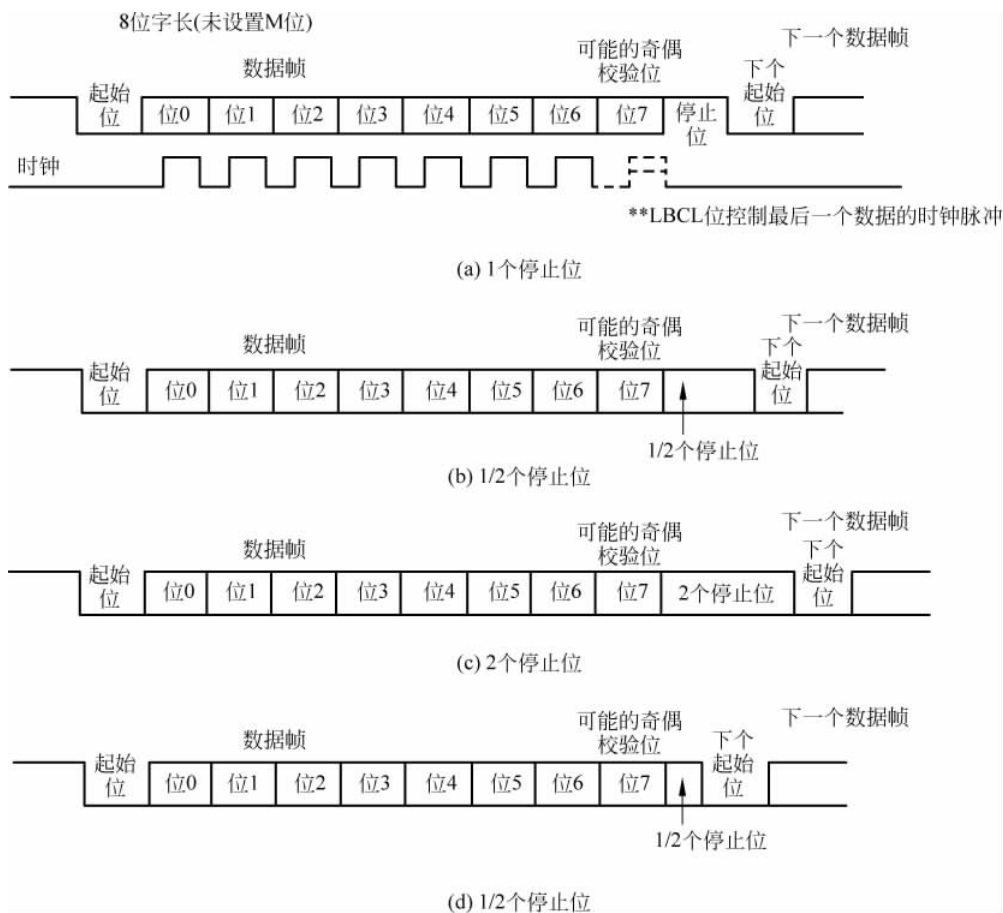


图 3.1.15 USART 的数据帧格式

断开符号被视为在一个帧周期内全部收到“0”(包括停止位期间,也是“0”)。在断开帧结束时,发送器再插入一或两个停止位 (“1”)来应答起始位。发送和接收由一共用的波特率发生器驱动,当发送器和接收器的使能位置位时,分别为其产生时钟。

在 TIVA C Launchpad 上进行 UART 串口通信的程序设计流程框图如图 3.1.16 所示。

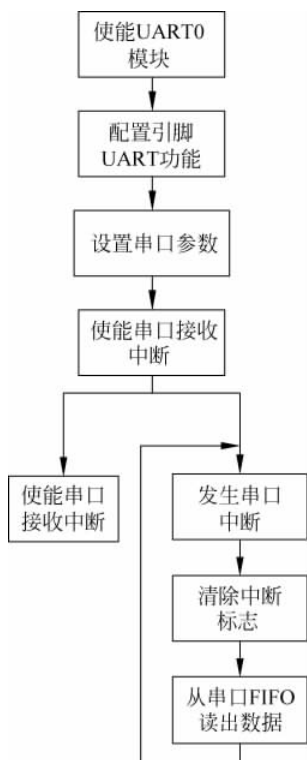


图 3.1.16 串口通信程序设计流程框图

3.2 常用 RC(Radio Controller)通信协议

3.2.1 PPM

PPM(Pulse Position Modulation, 脉宽位置调制)调制信号控制脉冲序列中各脉冲的相对位置(即相位),使各脉冲的相对位置随调制信号变化。此时序列中脉冲幅度和宽度均不变。PPM 是遥控模型中比较通用的一种信号格式,是接收机收到遥控器发出的 4 通道信息后,输出给飞行控制模块的编码信号,飞行控制部分对其进行解码即可得到各个通道的控制量。

PPM 原理是通过检测给定频率的 PWM 信号的占空比来获取指令信号的。如图 3.2.1 所示,信号频率为 50Hz,一个周期为 20ms,每个信号脉宽 0~2ms; 1 帧 PPM

信号长度为 20ms,实际遥控最多可以容纳 9 个通道。一个通道的 2ms 时间内,脉宽仅在 1~2ms 内变化,即脉宽 1ms 表示停转,脉宽 2ms 表示满量程运作,其间各点按比例换算,如脉宽 1.5ms 表示 50%的量程。

PPM 信号波形如图 3.2.1 所示。

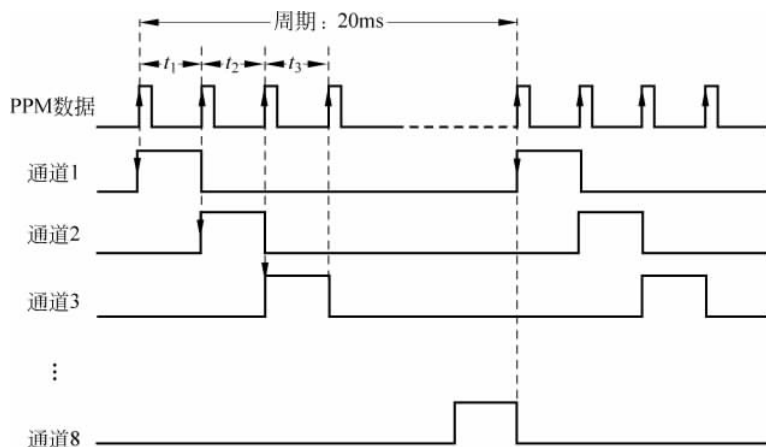


图 3.2.1 PPM 信号图解

3.2.2 PWM

PWM(Pulse Width Modulation,脉冲宽度调制)是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术,一般变换后脉冲的周期固定,但脉冲的占空比会依模拟信号的大小而改变,如图 3.2.2 所示。PWM 被广泛应用在从测量、通信到功率控制与变换的许多领域中。

在模拟电路中,模拟信号的值可以连续进行变化,在时间和值的幅度上几乎没有限制,基本上可以取任何实数值,输入与输出也呈线性变化。所以在模拟电路中,电压和电流可直接用来进行控制对象,如家用电器设备中的音量开关控制、采用卤素灯泡灯具的亮度控制等。但模拟电路有诸多问题:例如,控制信号容易随时间漂移,难以调节;功耗大;易受噪声和环境干扰等。与模拟电路不同,数字电路是在预先确定的范围内取值,在任何时刻其输出只可能为 ON 和 OFF 两种状态,所以电压或电流会以通/断方式的重复脉冲序列加载到模拟负载。

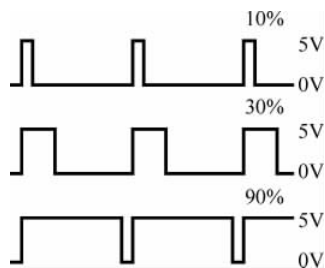


图 3.2.2 不同占空比的 PWM 波

PWM 技术是一种对模拟信号电平的数字编码方法,通过使用高分辨率计数器(调制频率)调制方波的占空比,从而实现对一个模拟信号的电平进行编码。其最大的优点是从事理器到被控对象之间的所有信号都是数字形式的,不需要再进行数模转换过程;而且对噪声的抗干扰能力也大大增强(噪声只有在强到足以将逻辑值改变时,才有可能对数字信号产生实质的影响),这也是 PWM 在通信等信号传输行业得到大量应用的主要原因。

模拟信号能否使用 PWM 进行编码调制,仅依赖带宽,这即意味着只要有足够的带宽,任何模拟信号值均可以采纳 PWM 技术进行调制编码,一般而言,负载需要的调制频率要高于 10Hz,在实际应用中,频率在 1~200kHz 之间。

在信号接收端,需将信号解调还原为模拟信号,目前在很多微型处理器内部都包含有 PWM 控制器模块。

PWM 控制技术以其控制简单、灵活和动态响应好的优点而成为电力电子技术最广泛应用的控制方式,也是人们研究的热点。由于当今科学技术的发展已经没有了学科之间的界限,结合现代控制理论思想或实现无谐振波开关技术将会成为 PWM 控制技术发展的主要方向之一。其根据相应载荷的变化调制晶体管基极或 MOS 管栅极的偏置,来实现晶体管或 MOS 管导通时间的改变,从而实现开关稳压电源输出的改变。这种方式能使电源的输出电压在工作条件变化时保持恒定,是利用微处理器的数字信号对模拟电路进行控制的一种非常有效的技术。

对于具有 PWM 模块的芯片,可以方便地通过硬件方式输出 PWM 波。下面给出配置 PWM 功能的程序设计流程框图(如图 3.2.3 所示)。



图 3.2.3 PWM 功能配置流程框图