

## uart

|       |   |       |                 |
|-------|---|-------|-----------------|
| 笔记本：  | arm   |       |                 |
| 创建时间： | 2015/2/26 19:36   | 更新时间： | 2015/7/30 23:54 |
| URL：  | <a href="http://blog.chinaunix.net/uid-24856020-id-2559044.html">http://blog.chinaunix.net/uid-24856020-id-2559044.html</a> |       |                 |

### 3 DESCRIPTION

以下各节介绍UART操作，包括数据传输，数据接收，中断产生，波特率产生，环回模式，红外模式和自动流控制。

#### 3.1数据传输

用于传输的数据帧是可编程的。它由一个起始位，5到8个数据位，一个可选的奇偶校验位，和1至2个停止位，这是由该行控制寄存器（ULCONn）指定。发射机也可以产生终止信号，这使串行输出为逻辑0状态时一帧的传输时间。之后，当前要传输的数据已完全传送后发出终止的信号。终止信号发出后，就可以继续发送数据到TX FIFO器（Tx控股的情况下非FIFO模式寄存器）。

#### 3.2数据接收

类似的数据传输，用于接收所述数据帧也是可编程的。它由一个起始位，5到8个数据位，一个可选的奇偶校验位，并在该行控制寄存器（ULCONn）1至2个停止位。接收器检测溢出错误，奇偶校验错误，帧错误和中断条件，每个设置一个错误标志。

- 溢出错误指示新数据覆写的旧数据的旧数据已经读出之前。
- 奇偶校验错误指示接收器检测到一个意想不到的平价条件。
- 帧错误指示接收到的数据没有一个有效的停止位。
- 休息状态表示RXDN输入的逻辑0状态下保持超过一帧的传输时间。

收到如果在3个字的时间没有接收到的数据（该时间间隔如下字长位的设置）和接收FIFO不为空在FIFO模式下发生超时状况。

#### 3.3自动流控制（AFC）

S5PC100的UART0和UART1支持自动流控制信号NRTS和NCTS。如果通过GPA1CON（GPIO SFR）设置TXD3和RXD3为nRTS2和nCTS2，UART2也可以支持自动流控制。在一般情况下，它可以被连接到外部的UART。一个连接到外部的uart，需要通过在UMCONn禁用自动流控制，使用软件控制NRTS的信号。

在AFC中，nRTS是接收机的接收条件，nCTS信号控制发射器的操作。

如果NCTS信号被激活，FIFO中的数据被UART发送（在AFC，nCTS意味着其它UART的FIFO准备接收数据）。在UART接收数据时，如果它的接收FIFO超过2个字节，NRTS必须被激活。如果其接收FIFO小于1个字节，该NRTS必须被灭活，（在AFC，NRTS意味着自己的接收FIFO已经准备好接收数据）。

#### 3.4例子NON AUTO-FLOW CONTROL（控NRTS和NCTS通过软件）

##### 3.4.1接收操作FIFO

- 1.选择发送模式（中断或DMA模式）。
- 2.检查接收FIFO计数UFSTATn寄存器中的值。如果该值小于16，则必须UMCONn的值设为0为“1”（激活NRTS），并且如果它等于或大于16，必须将值设置为0'（灭活NRTS）。
- 3.重复步骤2。

##### 3.4.2 Tx操作，与FIFO

- 1.选择发送模式（中断或DMA模式）。
- 2.检查UMSTATn值[0]。如果值为“1”（激活NCTS），您必须将数据写入发送FIFO寄存器。
- 3.重复步骤2

##### 3.5 RS-232C接口

要连接到UART调制解调器接口（取代无调制器），NRTS，NCTS，nDSR，nDTR，DCD和NRI信号是必需的。在这种情况下，可以通过软件控制这些信号与通用I / O口，因为AFC不支持RS-232C接口。

##### 3.6中断/ DMA请求产生

该S5PC100的每个UART有七个状态（TX / RX / 错误）信号，即：溢出错误，奇偶校验错误，帧错误，中断，接收缓冲区的数据准备好，发送缓冲空，和发送器空。这些条件由相应的UART状态寄存器（UTRSTATn / UERSTATn）表示。

溢出错误，奇偶校验错误，帧错误和断点条件被称为接收错误状态。如果收到错误状态-中断启用位被设置为1，在控制寄存器 UCONn中，接收错误状态产生接收错误状态中断。如果接收错误状态中断请求被检测到，则可以通过读取UERSTSTn的值确定中断的源。

如果接收机传送的数据的接收器，以接收FIFO寄存器在FIFO模式和接收的数据的数量达到Rx FIFO的触发电平，接收中断产生，如果接收在控制寄存器（UCONn）被设置为1模式（中断请求或轮询模式）。

在非FIFO模式，传输接收器来接收保持寄存器中的数据会导致接收中断请求和轮询模式下的中断。

如果发送FIFO通过移位器传出的数据量达到了 FIFO中断的门限值，Tx中断被产生。在无FIFO模式下，寄存器里的数据被移位器发送后即产生中断。

注意，当fifo中的数据少于触发值时，中断信号始终有效。所以建议先填充缓冲区再设置中断使能。

如果中断目的是发送dma请求，请将中断配置为dma请求模式。

#### 3.7 UART错误状态FIFO

UART具有除了接收FIFO寄存器中的错误状态FIFO。错误状态FIFO指示数据，FIFO寄存器中，收到一个错误。错误中断发出仅当数据，它有一个错误时，已准备好被读出。清除错误状态FIFO，所述URXHn一个错误并UERSTATn必须被读出。

例如，假设在UART Rx FIFO的接收的A，B，C，D，并依次E字符和同时接收“B”出现帧误差，并同时接收“D”发生了奇偶错误。

实际的UART接收错误不会产生任何错误中断。有这个特性，这是接收到了错误，并没有被读取呢。如果字符被读出时，错误中断产生。

##### 3.7.1红外线（IR）模式

该S5PC100 UART模块支持红外（IR）发送和接收。它是通过设置在UART线控制寄存器（ULCONn）的Infrared-mode。图8.1-说明了如何实现IR模式。红外发送模式中，发送3/16脉冲出来，用正常的串行传输速率（如果发送的数据位是0）。的速率在红外线接收模式下，接收器必须检测3/16脉冲周期来识别一个0值（参照图8.1-和8.1-图所示的帧时序图）。

- 1 载波检测(DCD)
- 2 接受数据(RXD)
- 3 发出数据(TXD)
- 4 数据终端准备好(DTR)

- 5 信号地线 (SG)
- 6 数据准备好 (DSR)
- 7 请求发送 (RTS)
- 8 清除发送 (CTS)
- 9 振铃指示 (RI)



| DCE (/DTE) 设备 |              |      | 同类型设备<br>七线制 | DCE (/DTE) 设备 |             |              |
|---------------|--------------|------|--------------|---------------|-------------|--------------|
| 针号<br>(9 针)   | 针号<br>(25 针) | 缩写   |              | 缩写            | 针号<br>(9 针) | 针号<br>(25 针) |
| 1             | 8            | DCD  |              | DCD           | 1           | 8            |
| 2             | 3            | RXD  |              | RXD           | 2           | 3            |
| 3             | 2            | TXD  |              | TXD           | 3           | 2            |
| 4             | 20           | DTR  |              | DTR           | 4           | 20           |
| 5             | 7            | GND  |              | GND           | 5           | 7            |
| 6             | 6            | DSR  |              | DSR           | 6           | 6            |
| 7             | 4            | RTS  |              | RTS           | 7           | 4            |
| 8             | 5            | CTS  |              | CTS           | 8           | 5            |
| 9             | 22           | DELL |              | DELL          | 9           | 22           |

串口通讯—通信协议

所谓通信协议是指通信双方的一种约定。约定包括对数据格式、同步方式、传送速度、传送步骤、检纠错方式以及控制字符定义等问题做出统一规定，通信双方必须共同遵守。因此，也叫做通信控制规程，或称传输控制规程，它属于ISO' S OSI七层参考模型中的数据链路层。

目前，采用的通信协议有两类：异步协议和同步协议。同步协议又有面向字符和面向比特以及面向字节计数三种。其中，面向字节计数的同步协议主要用于DEC公司的网络体系结构中。

一、物理接口标准

1. 串行通信接口的基本任务

- (1) 实现数据格式化：因为来自CPU的是普通的并行数据，所以，接口电路应具有实现不同串行通信方式下的数据格式化的任务。在异步通信方式下，接口自动生成起止式的帧数据格式。在面向字符的同步方式下，接口要在待传送的数据块前加上同步字符。
- (2) 进行串—并转换：串行传送，数据是一位一位串行传送的，而计算机处理数据是并行数据。所以当数据由计算机送至数据发送器时，首先把串行数据转换为并行数才能送入计算机处理。因此串并转换是串行接口电路的重要任务。
- (3) 控制数据传输速率：串行通信接口电路应具有对数据传输速率——波特率进行选择和控制在的能力。
- (4) 进行错误检测：在发送时接口电路对传送的字符数据自动生成奇偶校验位或其他校验码。在接收时，接口电路检查字符的奇偶校验或其他校验码，确定是否发生传送错误。
- (5) 进行TTL与EIA电平转换：CPU和终端均采用TTL电平及正逻辑，它们与EIA采用的电平及负逻辑不兼容，需在接口电路中进行转换。
- (6) 提供EIA-RS-232C接口标准所要求的信号线：远距离通信采用MODEM时，需要9根信号线；近距离零MODEM方式，只需要3根信号线。这些信号线由接口电路提供，以便与MODEM或终端进行联络与控制。

2、串行通信接口电路的组成

为了完成上述串行接口的任务，串行通信接口电路一般由可编程的串行接口芯片、波特率发生器、EIA与TTL电平转换器以及地址译码电路组成。其中，串行接口芯片，随着大规模集成电路技术的发展，通用的同步(USRT)和异步(UART)接口芯片种类越来越多，如下表所示。它们的基本功能是类似的，都能实现上面提出的串行通信接口基本任务的大部分工作，且都是可编程的。才用这些芯片作为串行通信接口电路的核心芯片，会使电路结构比较简单。

| 芯片       | 同步(USRT) |      | 异步(UART) (起止式) | 传输速率b/s |       |
|----------|----------|------|----------------|---------|-------|
|          | 面向字符     | HDLC |                | 同步      | 异步    |
| INS8250  |          |      | √              |         | 56K   |
| MC6850   |          |      | √              |         | 1M    |
| MC6852   | √        |      |                | 1. 5M   |       |
| MC6854   |          | √    |                | 1. 5M   |       |
| Int8251A | √        |      | √              | 64K     | 19.2K |
| Int8273  |          | √    |                | 64K     |       |
| Z-80 SIO |          | √    | √              | 800K    |       |

3. 有关串行通信的物理标准

为使计算机、电话以及其他通信设备互相沟通，现在，已经对串行通信建立了几个一致的概念和标准，这些概念和标准属于三个方面：传输率，电特性，信号名称和接口标准。

- 1、**传输率**：所谓传输率就是指每秒传输多少位，传输率也常叫波特率。国际上规定了一个标准波特率系列，标准波特率也是最常用的波特率，标准波特率系列为110、300、600、1200、4800、9600和19200。大多数CRT终端都能够按110到9600范围中的任何一种波特率工作。打印机由于机械速度比较慢而使传输波特率受到限制，所以，一般的串行打印机工作在110波特率，点针式打印机由于其内部有较大的行缓冲器，所以可以按高达2400波特的速度接收打印信息。大多数接口的接收波特率和发送波特率可以分别设置，而且，可以通过编程来指定。
- 2、**RS-232-C标准**：RS-232-C标准对两个方面作了规定，即信号电平标准和控制信号线的定义。RS-232-C采用负逻辑规定逻辑电平，信号电平与通常的TTL电平也不兼容，RS-232-C将-5V~-15V规定为“1”，+5V~+15V规定为“0”。图1是TTL标准和RS-232-C标准之间的电平转换。

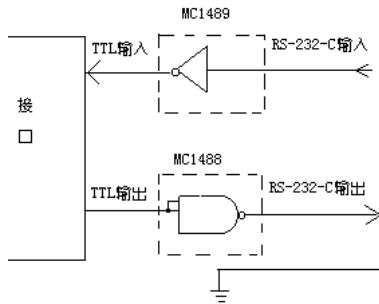


图1

## 二、软件协议

### 2. 串行通信协议

串行通信协议分同步协议和异步协议。

#### (1) 异步通信协议的实例——起止式异步协议

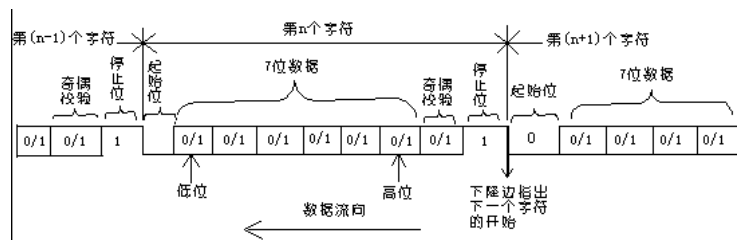


图3

#### 特点与格式：

起止式异步协议的特点是一个字符一个字符传输，并且传送一个字符总是以起始位开始，以停止位结束，字符之间没有固定的时间间隔要求。其格式如图3所示。每一个字符的前面都有一位起始位（低电平，逻辑值0），字符本身有5~7位数据位组成，接着字符后面是一位校验位（也可以没有校验位），最后是一位，或意味半，或二位停止位，停止位后面是不定长度的空闲位。停止位和空闲位都规定为高电平（逻辑值1），这样就保证起始位开始处一定有一个下跳沿。

从图中可以看出，这种格式是靠起始位和停止位来实现字符的界定或同步的，故称为起止式协议。传送时，数据的低位在前，高位在后，图4表示了传送一个字符E的ASCII码的波形1010001。当把它的最低有效位写到右边时，就是E的ASCII码1000101=45H。

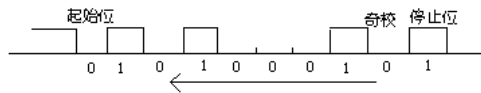


图4

**起 / 止位的作用：**起始位实际上是作为联络信号附加进来的，当它变为低电平时，告诉收方传送开始。它的到来，表示下面接着是数据位来了，要准备接收。而停止位标志一个字符的结束，它的出现，表示一个字符传送完毕。这样就为通信双方提供了何时开始收发，何时结束的标志。传送开始前，收发双方把所采用的起止式格式（包括字符的数据位长度，停止位数，有无校验位以及是奇校验还是偶校验等）和数据传输速率作统一规定。传送开始后，接收设备不断地检测传输线，看是否有起始位到来。当收到一系列的“1”（停止位或空闲位）之后，检测到一个下跳沿，说明起始位出现，起始位经确认后，就开始接收所规定的的数据位和奇偶校验位以及停止位。经过处理将停止位去掉，把数据位拼装成一个并行字节，并且经校验后，无奇偶错才算正确的接收一个字符。一个字符接收完毕，接收设备有继续测试传输线，监视“0”电平的到来和下一个字符的开始，直到全部数据传送完毕。

由上述工作过程可看到，异步通信是按字符传输的，每传输一个字符，就用起始位来通知收方，以此来重新核对收发双方同步。若接收设备和发送设备两者的时钟频率略有偏差，这也不会因偏差的累积而导致错位，加之字符之间的空闲位也为这种偏差提供一种缓冲，所以异步串行通信的可靠性高。但由于要在每个字符的前后加上起始位和停止位这样一些附加位，使得传输效率变低了，只有约80%。因此，起止协议一般用在数据速率较慢的场合（小于19.2kbit/s）。在高速传送时，一般要采用同步协议。

#### (2) 面向字符的同步协议

**特点与格式：**这种协议的典型代表是IBM公司的二进制同步通信协议(BSC)。它的特点是一次传送由若干个字符组成的数据块，而不是只传送一个字符，并规定了10个字符作为这个数据块的开头与结束标志以及整个传输过程的控制信息，它们也叫做通信控制字。由于被传送的数据块是由字符组成，故被称作面向字符的协议。

**特定字符（控制字符）的定义：**由上面的格式可以看出，数据块的前后都加了几个特定字符。SYN是同步字符(synchronous Character)，每一帧开始处都有SYN，加一个SYN的称单同步，加两个SYN的称双同步设置同步字符是起联络作用，传送数据时，接收端不断检测，一旦出现同步字符，就知道是一帧开始了。接着的SOH是序始字符(Start Of Header)，它表示标题的开始。标题中包括院地址、目的地址和路由指示等信息。STX是文始字符(Start Of Text)，它标志着传送的正文(数据块)开始。数据块就是被传送的正文内容，由多个字符组成。数据块后面是组终字符ETB(End Of Transmission Block)或文终字符ETX(End Of Text)，其中ETB用在正文很长、需要分成若干个分数据块、分别在不同帧中发送的场合，这时在每个分数据块后面用文终字符ETX。一帧的最后是校验码，它对从SOH开始到ETX(或ETB)字段进行校验，校验方式可以是纵横奇偶校验或CRC。另外，在面向字符协议中还采用了一些其他通信控制字，它们的名称如下表所示：

| 名 称     | ASCII   |
|---------|---------|
| 序始(SOH) | 0000001 |
| 文始(STX) | 0000010 |
| 组终(ETB) | 0010111 |
| 文终(ETX) | 0000011 |
| 同步(SYN) | 0010110 |
| 送毕(EOT) | 0000100 |
| 询问(ENQ) | 0000101 |
| 确认(ACK) | 0000110 |
| 否认(NAK) | 0010101 |
| 转义(DLE) | 0010000 |

**数据透明的实现：**面向字符的同步协议，不象异步起止协议那样，需要在每个字符前后附加起始和停止位，因此，传输效率提高了。同时，由于采用了一些传输控制字，故增强了通信控制能力和校验功能。但也存在一些问题，例如，如何区别数据字符代码和特定字符代码的问题，因为在数据块中完全有可能出现与特定字符代码相同的数据字符，这就会发生误解。比如正文有个与文终字符ETX的代码相同的数据字符，接收端就不会把它当作普通数据处理，而误认为是正文结束，因而产生差

错。因此，协议应具有将特定字符作为普通数据处理的能力，这种能力叫做“数据透明”。为此，协议中设置了转移字符DLE(Data Link Escape)。当把一个特定字符看成数据时，在它前面要加一个DLE，这样接收器收到一个DLE就可预知下一个字符是数据字符，而不会把它当作控制字符来处理了。DLE本身也是特定字符，当它出现在数据块中时，也要在它前面加上另一个DLE。这种方法叫字符填充。字符填充实现起来相当麻烦，且依赖于字符的编码。正是由于以上的缺点，故又产生了新的面向比特的同步协议。

（3）面向比特的同步协议

**特点与格式：**面向比特的协议中最具有代表性的是IBM的同步数据链路控制规程SDLC（Synchronous Data Link Control),国际标准化组织ISO(International Standard Organization)的高级数据链路控制规程HDLC（High Level Data link Control),美国国家标准协会(Americal National Standard Institute)的先进数据通信规程ADCCP(Advanced Data Communication Control Procedure)。这些协议的特点是所传输的一帧数据可以是任意位，而且它是靠约定的位组合模式，而不是靠特定字符来标志帧的开始和结束，故称“面向比特”的协议。这中协议的一般帧格式如图5所示：

| 8位       | 8位  | 8位  | ≥0位 | 16位 | 8位       |
|----------|-----|-----|-----|-----|----------|
| 01111110 | A   | C   | 1   | FC  | 01111110 |
| 开始标志     | 地址场 | 控制场 | 信息场 | 校验场 | 结束标志     |

图5

**帧信息的分段：**由图5可见，SDLC/HDLC的一帧信息包括以下几个场(Filed)，所有场都是从有效位开始传送。

- （1）SDLC/HDLC标志字符：SDLC/HDLC协议规定，所有信息传输必须以一个标志字符开始，且以同一个字符结束。这个标志字符是 01111110，称标志场(F)。从开始标志到结束标志之间构成一个完整的信息单位，称为一帧(Frame)。所有的信息是以帧的形传输的，而标志字符提供了每一帧的边界。接收端可以通过搜索“01111110”来探知帧的开头和结束，以此建立帧同步。
- （2）地址场和控制场：在标志场之后，可以有一个地址场A(Address)和一个控制场C(Control)。地址场用来规定与之通信的次站的地址。控制场可规定若干个命令。SDLC规定A场和C场的宽度为8位或16位。接收方必须检查每个地址字节的第一位，如果为“0”，则后面跟着另一个地址字节；若为“1”，则该字节就是最后一个地址字节。同理，如果控制场第一个字节的第一位为为“0”，则还有第二个控制场字节，否则就只有一个字节。
- （3）信息场：跟在控制场之后的是信息场I(Information)。I场包含有要传送的数据，并不是每一帧都必须有信息场。即数据场可以为0，当它为0时，则这一帧主要是控制命令。
- （4）帧校验信息：紧跟在信息场之后的是两字节的争校验，帧校验场称为FC(Frame Check)场或称为帧校验序列FCS(Frame check Squence)。SDLC/HDLC均采用16位循环冗余校验码CRC（Cyclic Redundancy Code）。除了标志场和自动插入的“0”以外，所有的信息都参加CRC计算。

**实际应用时的两个技术问题：**

- （1）“0”位插入/删除：如上所述，SDLC/HDLC协议规定以01111110为标志字节，但在信息场中也完全有可能有同一种模式的字符，为了把它与标志区分开来，所以采取了“0”位插入和删除技术。具体作法是发送端在发送所有信息（除标志字节外）时，只要遇到连续5个“1”，就自动插入一个“0”，当接收端在接收数据时（除标志字节）如果连续收到5个“1”，就自动将其后的一个“0”删除是，以恢复信息的原有形式。这种“0”位的插入和删除过程是由硬件自动完成的。
- （2）SDLC/HDLC异常结束：若在发送过程中出现错误，则SDLC/HDLC协议常用异常结束(Abort)字符，或称为失效序列使本帧作废。在HDLC规程中，7个连续的“1”被作为失效字符，而在SDLC中失效字符是8个连续的“1”。当然在试销序列中不使用“0”位插入/删除技术。SDLC/HDLC协议规定，在一帧之内不允许出现数据间隔。在两帧之间，发送器可以连续输出标志字符序列，也可以输出连续的高电平，它被称为空闲（Idle）信号。