串口通信

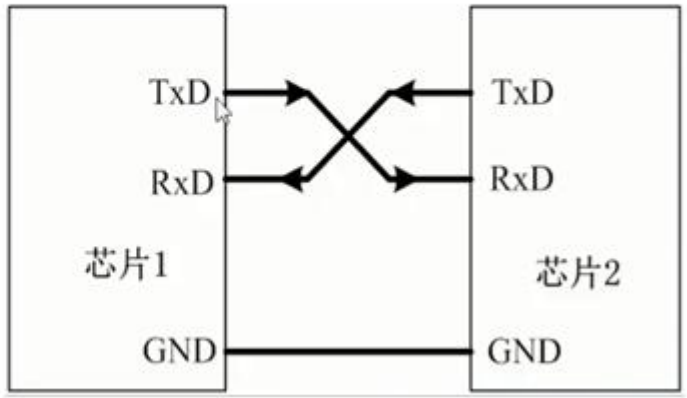
https://blog.csdn.net/eastmoon502136/article/details/8190262

WHAT

- 串口是计算机上一种非常通用设备通信的协议。
- 串口按位（bit）发送和接收字节。

RXD：数据输入引脚，数据接受；

TXD：数据发送引脚，数据发送。



帧格式

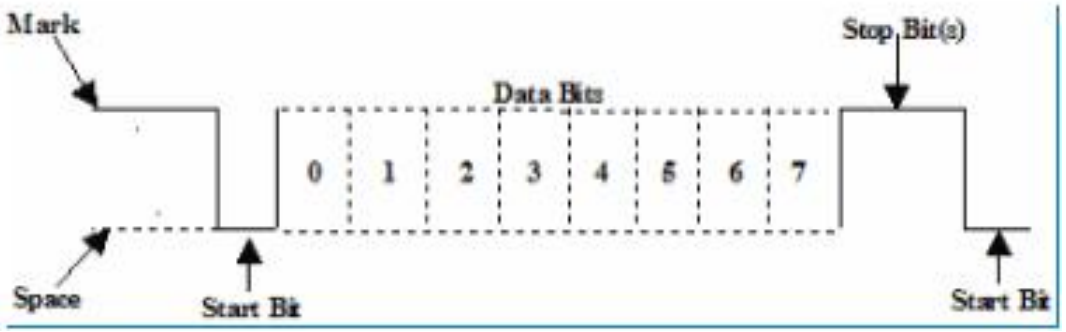
- start bit 、 data bits 、 parity bit 、 stop bits

start bit固定为1bit;

data bits可以为5、6、7、8或者9bits，不过常用的都是8bits;

parity bit是非必须的，一般为0bit;

stop bit可以是1bit和2bits两种，一般都是1bit。



串口操作需要的头文件

```
#include <errno.h>
#include <fcntl.h>
#include <string.h>
#include <termios.h>
#include <unistd.h>
```

打开串口

串口位于/dev/ttyUSB*

```
std::string serial_path = "/dev/ttyUSB" + std::to_string(i);
fd = open(serial_path.c_str(), O_RDWR | O_NOCTTY);
```

波特率、奇偶校验设置

```
int SerialCom::set_interface_attribs(int fd, int speed, int parity)
{
    struct termios tty;
    memset(&tty, 0, sizeof tty);
    if (tcgetattr(fd, &tty) != 0)
    {
        // error_message ("error %d from tcgetattr", errno);
        return -1;
    }

    cfsetospeed(&tty, speed);
    cfsetispeed(&tty, speed);

    tty.c_cflag = (tty.c_cflag & ~CSIZE) | CS8; // 8-bit chars
    // disable IGNBRK for mismatched speed tests; otherwise receive break
    // as \000 chars
    tty.c_iflag &= ~IGNBRK; // disable break processing
```

串口读写

```
int SerialCom::receive()
{
    // unsigned char recieve_data[RXPackSize] = {};
    int fd = this->fd;
    int n = read(fd, recvBuffer, sizeof(recvBuffer));

    // std::cout <<"receive size in receive:" <<n <<std::endl;
    // std::cout <<"receive data" <<recvBuffer <<std::endl;

    if ((n < RXPackSize) || !Verify_CRC8_Check_Sum(recvBuffer, RXPackSize))
        return -1;
    else
        return n;
}

int SerialCom::send(unsigned char *str, int len)
{
    int n = write(this->fd, str, len);
    return n;
}
```