uart_demo程序

```
/*
 * uart.c
 *
 *  Created on: Feb 13, 2017
 *      Author: user
 *      berif :
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <errno.h>
#include <limits.h>

#define DEV_NAME  "/dev/ttyLP2"

int main (int argc, char *argv[])
{
            int iFd = 0, i = 0;
            int len = 0;
            unsigned char ucBuf[4] = {0xc0, 0x38, 0xbc, 0x7e};
            unsigned char ubbuf[7] = "kong";

            /**
              POSIX 终端属性描述结构 struct termios
              struct termios {
                    tcflag_t c_cflag;    //控制标志(波特率，数据位，奇偶校验，停止位，流控)
                    tcflag_t c_iflag;    //输入标志
                    tcflag_t c_oflag;    //输出标志
                    tcflag_t c_lflag;    //本地标志
                    tcflag_t c_cc[NCCS]; //控制字符
              }
             */
            struct termios opt;

      /* 打开串口设备文件，得到文件操作符 */
            iFd = open(DEV_NAME, O_RDWR | O_NOCTTY);
            if (iFd < 0) {
                  perror(DEV_NAME);
                  return -1;
            }

       /**
```

```
 * @brief：获取串口设备参数
 * @Parma：fd      ：文件操作符
 *         termptr：串口设备的termios结构
 */
      tcgetattr(iFd, &opt);
      if (tcgetattr(iFd, &opt) < 0) {
            return -1;
      }

/* 设置输入输出串口波特率，也就是设置termios结构的参数 */
      cfsetispeed(&opt, B115200);
cfsetospeed(&opt, B115200);

/************   c_lflag（本地标志）   ********************
 ISIG 启用终端产生的信号 NOFLSH 在中断或退出键后禁用刷清
 ICANON 启用规范输入     IEXTEN 启用扩充的输入字符处理
 XCASE 规范大/小写表示   ECHOCTL 回送控制字符为(char)
 ECHO 进行回送           ECHOPRT 硬拷贝的可见擦除方式
 ECHOE 可见擦除字符      ECHOKE Kill 的可见擦除
 ECHOK 回送 kill 符      PENDIN 重新打印未决输入
 ECHONL 回送 NL          TOSTOP 对于后台输出发送 SIGTTOU
*****************************************************/
      opt.c_lflag   &= ~(ECHO | ICANON | IEXTEN |ISIG);

      /****************   c_iflag(输入标志)   ************************
       * INPCK  打开输入奇偶校验 IXOFF  启用/停止输入流控
* IGNPAR 忽略奇偶错字符   IGNBRK 忽略BREAK条件
* PARMRK 标记奇偶错       INLCR  将输入的 NL 转换为 CR
* ISTRIP 剥除字符第 8 位  IGNCR  忽略 CR
* IXON 启用/停止输出流控  ICRNL  将输入的 CR 转换为 NL
*************************************************************/
      opt.c_iflag   &= ~(BRKINT | ICRNL | INPCK | ISTRIP | IXON);

      /*****************   c_oflag(输出标志)       ******************
  BSDLY 退格延迟屏蔽              OLCUC  将输出的小写字符转换为大写字符
  CMSPAR 标志或空奇偶性           ONLCR  将 NL 转换为 CR-NL
  CRDLY CR 延迟屏蔽               ONLRET NL执行 CR 功能
  FFDLY 换页延迟屏蔽              ONOCR 在 0 列不输出 CR
  OCRNL 将输出的CR转换为NL        OPOST 执行输出处理
  OFDEL 填充符为 DEL，否则为NULL  OXTABS 将制表符扩充为空格
  OFILL 对于延迟使用填充符
       *************************************************************/
      opt.c_oflag &= ~(OPOST);

       /**************   c_cflag(控制标志)   ***************
   CSIZE 数据位屏蔽 CS7 7 位数据位
   CS5 5 位数据位   CS8 8 位数据位
   CS6 6 位数据位   PARENB 进行奇偶校验
   PARODD 奇校验，否则为偶校验
*************************************************************/
      opt.c_cflag  &= ~(CSIZE | PARENB);
```

```
        opt.c_cflag |= CS8;

        /**************      c_cc(控制字符)      ***************
        VINTR 中断   VEOL 行结束
    VQUIT 退出   VMIN 需读取的最小字节数
    VERASE 擦除  VTIME 与“ VMIN”配合使用，是指限定的传输或等待的最长时间
    VEOF 行结束
     ************************************************************/
        opt.c_cc[VMIN]  = 255;
        opt.c_cc[VTIME] = 150;

    /**
     * @brief ：设置参数函数
     * @detail：TCSANOW  ： 更改立即发生
     *        TCSADRAIN： 发送了所有输出后更改才发生，若更改输出参数则应用此选项
     *        TCSAFLUSH： 发送了所有输出后更改才发生，在更改发生时未读的所有输入数据被删除（Flush）
     */
        if (tcsetattr(iFd, TCSANOW, &opt) < 0) {
            return -1;
        }

        /**
         * @brief ：清空终端未完成的输入输出请求及数据
         * @detail：TCIFLUSH：清除正收到的数据，且不会读取出来。
     *        TCOFLUSH：清除正写入的数据，且不会发送至终端。
     *        TCIOFLUSH：清除所有正在发生的I/O数据。
         */
        tcflush(iFd, TCIOFLUSH);

    /**
     * @brief ：发送数据函数
     * @detail：向文件操作符写入要发送的数据
     */
    write(iFd, ucBuf, sizeof(ucBuf));

    printf("write success~\n");

    len = read(iFd, ubbuf, sizeof(ubbuf));

    printf("get data: %d \n", len);
    for (i = 0; i < len; i++) {
        printf("%x", ubbuf[i]);
    }
    printf("\n");

        close(iFd);
        return 0;
}
```