

FT60F01X

UART Application note

目录

1. UART 介绍	3
2. UART 相关设置	3
3. 应用范例	4
联系信息	9

FT60F01x UART 应用

1. UART 介绍

UART是一种通用串行数据总线，用于异步通信。该总线双向通信，可以实现全双工传输和接收。

UART 协议，其中各位的意义如下：

起始位：先发出一个逻辑“0”的信号，表示传输字符的开始。

资料位：紧接着起始位之后，资料位的个数可以是 4、5、6、7、8 等，构成一个字符。通常采用 ASCII 码，从最低位开始传送，靠时钟定位。

奇偶校验位：资料位加上这一位后，使得“1”的位数应为偶数(偶校验)或奇数(奇校验)，以此来校验资料传送的正确性。

停止位：它是一个字符数据的结束标志。可以是 1 位、1.5 位、2 位的高电平。由于数据是在传输线上定时的，并且每一个设备有其自己的时钟，很可能在通信中两台设备间出现了小小的不同步。因此停止位不仅仅是表示传输的结束，并且提供计算机校正时钟同步的机会。适用于停止位的位数越多，不同时钟同步的容忍程度越大，但是数据传输率同时也越慢。

空闲位：处于逻辑“1”状态，表示当前线路上没有资料传送。

波特率：是衡量资料传送速率的指标。表示每秒钟传送的符号数（symbol）。一个符号代表的信息量（比特数）与符号的阶数有关。例如资料传送速率为 120 字符/秒，传输使用 256 阶符号，每个符号代表 8bit，则波特率就是 120baud，比特率是 $120 \times 8 = 960 \text{ bit/s}$ 。

2. UART 相关设置

本例子是模拟UART发送部分，使用定时器让TXIO产生与设定好的波特率一样频率的信号；接收部分，使用RXIO电平转换中断识别起始信号，然后再根据定时器扫描数据识别数据内容。

讲解以IC FT60F011A SOP8为示范，在收到电脑端的串口助手发来的数据后，又把同样的数据发出去。

本程序数据线TX与RX所对应的IO引脚：

```
#define TXIO PA4
```

```
#define RXIO PA2
```

3. 应用范例

```
//*****
/* 文件名: TEST_60F01x_UART.c
* 功能:    FT60F01x-UART 功能演示
* IC:      FT60F011A SOP8
* 晶振:    16M/4T
* 说明:    演示程序中波特率为 9600, RXIO (PA2) 每次收到外部串口发过来的数据后,
*          TXIO(PA4)把收到的数据再发送出去。接收起始位时是用电平变化中断识别,
*          后面关闭电平变化中断。
*
*          FT60F011A SOP8
*          -----
* VDD-----|1(VDD)   (GND)8|-----GND
* RXIO-----|2(PA2)   (PA4)7|-----TXIO
* NC-----|3(PA1)    (PA5)6|-----NC
* NC-----|4(PA3)    (PA0)5|-----NC
*
*          -----
*/
//*****
#include "SYSCFG.h";
#include "FT60F01X.h";
//*****宏定义*****
#define uchar    unsigned char

#define TXIO      PA4    //串口的发送脚
#define RXIO      PA2    //串口的接收脚

#define Bord      49     //通过定时器提供波特率
uchar RXFLAG = 0;
uchar ReadAPin;
/*-----*/
* 函数名: POWER_INITIAL
* 功能:   MCU 初始化函数
* 输入:   无
* 输出:   无
*-----*/
void POWER_INITIAL(void)
{
    OSCCON = 0B01110000;    //IRCF=111=16MHz/4T=4MHz, 0.25μs
    INTCON = 0;             //暂禁止所有中断
    PORTA = 0B00000000;
    TRISA = 0B00000100;     //PA 输入输出 0-输出 1-输入
                           //PA4-OUT PA2-IN
    WPUA = 0B00000100;     //PA 端口上拉控制 1-开上拉 0-关上拉
}
```

```

//PA2 开上拉
OPTION = 0B00001000; //Bit3=1, WDT MODE, PS=000=WDT RATE 1:1
MSCKCON = 0B00000000;
//Bit4->0,禁止 LVR(60F01x O 版之前)
//Bit4->0,LVREN 使能时,开启 LVR(60F01x O 版及 O 版后)
//Bit4->1,LVREN 使能时,工作时开启 LVR,睡眠时自动关闭 LVR(60F01x O 版及 O 版之后)
}
/*-----
* 函数名: TIMER0_INITIAL
* 功能: 初始化设置定时器
* 设置 TMR0 定时时长=1/系统时钟频率*4*预分频值*208
          =(1/16000000)*4*2*208=104μs
-----*/
void TIMER0_INITIAL(void)
{
    OPTION = 0B00000000; //时钟源为指令时钟, 预分频比为 1:256
    //Bit5: T0CS Timer0 时钟源选择
    // 1-外部引脚电平变化 T0CKI 0-内部时钟(FOSC/4)
    //Bit4: T0SE T0CKI 引脚触发方式 1-下降沿 0-上升沿
    //Bit3: PSA 预分频器分配位 0-Timer0 1-WDT
    //Bit[2:0]: PS 8 位预分频比 000 - 1:2
    TMR0 = Bord;
    T0IF = 0; //清空 T0 软件中断
}
/*-----
* 函数名: PA2_Level_Change_INITIAL
* 功能: PA 端口(PA2)电平变化中断初始化
* 输入: 无
* 输出: 无
-----*/
void PA2_Level_Change_INITIAL(void)
{
    TRISA2 = 1; //设置 PA2 为输入
    ReadAPin = PORTA; //清 PA 电平变化中断
    PAIF = 0; //清 PA_INT 中断标志位
    IOCA2 = 1; //使能 PA2 电平变化中断
    PAIE = 1; //使能 PA_INT 中断
}
/*-----
* 函数名: interrupt ISR
* 功能: 中断处理, 包括定时器 0 中断和外部中断
* 输入: 无
* 输出: 无
-----*/

```

```

void interrupt ISR(void)
{
    //定时器 0 的中断处理
    if(T0IE && T0IF)                //104μs
    {
        TMR0 = Bord;                //注意:对 TMR0 重新赋值 TMR0 在两个周期内不变化
        T0IF = 0;
        T0IE = 0;
    }

    //PA 电平变化中断
    if(PAIE && PAIF)
    {
        ReadAPin = PORTA;            //读取 PORTA 数据清 PAIF 标志
        PAIF = 0;                    //清 PAIF 标志位
        if(RXIO == 0)
        {
            PAIE = 0;                //暂先禁止 PA 电平变化中断
            IOCA2 = 0;                //禁止 PA2 电平变化中断
            RXFLAG = 1;
        }
    }
}

/*-----*/
* 函数名: WaitTF0
* 功能:  查询定时器溢出后, 在中断里关闭定时器后, 再次打开定时器
* 输入:  无
* 输出:  无
/*-----*/

void WaitTF0( void )
{
    while(T0IE);
    T0IE=1;
}

/*-----*/
* 函数名: WByte
* 功能:  UART 发送一个字节
* 输入:  input
* 输出:  无
/*-----*/

void WByte(uchar input)
{
    //发送起始位

    uchar i=8;

```

```

TXIO = 1;
TMR0 = Bord;
T0IE = 1;
WaitTF0();
TXIO=0;
WaitTF0();

//发送 8 位数据位

while(i--)
{
    if(input&0x01) //先传低位
    {
        TXIO=1;
    }
    else
    {
        TXIO = 0;
    }
    WaitTF0();
    input=input>>1;
}

//发送校验位(无)
//发送结束位

TXIO=(bit)1;
T0IE=0;
}
/*-----
* 函数名: RByte
* 功能:   UART 接收一个字节
* 输入:   无
* 输出:   Output
*-----*/

uchar RByte()
{
    uchar Output=0;
    uchar i=8;
    T0IE=1; //启动 Timer0
    TMR0 = Bord;
    WaitTF0();
    T0IE=1; //启动 Timer0
    TMR0 = Bord;
    WaitTF0(); //等过起始位
    //发送 8 位数据位

    while(i--)
    {

```

```
        Output >>=1;
        if(RXIO)
        {
            Output|=0x80;           //先收低位
        }
        WaitTF0();                 //位间延时
    }
    T0IE=0;                        //停止 Timer0
    return Output;
}
/*-----
* 函数名:  main
* 功能:   主函数
* 输入:   无
* 输出:   无
-----*/
void main(void)
{
    uchar rdata = 0;
    POWER_INITIAL();
    TIMER0_INITIAL();
    PA2_Level_Change_INITIAL();
    GIE = 1;                       //开中断
    T0IE = 1;                      //开定时器/计数器 0 中断
    while(1)
    {

        if(RXFLAG)                //外部中断下降沿触发了
        {
            rdata = RByte();
            WByte(rdata);

            IOCA2 =1;              //使能 PA2 电平变化中断
            PAIE =1;               //使能 PA INT 中断
            RXFLAG = 0;
        }
    }
}
```


联系信息

Fremont Micro Devices Corporation

#5-8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

#16, 16/F, Block B, Veristrong Industrial Centre,
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.