# FT6XF2XX

# Application note



 

# 目录

# FT6XF2XX 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*8=960bit/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        RA4

#define   RXIO        RA2
```

## 3 应用范例

```
//************************************************************
/*   文件名：Test_62F21X_UART.c
*    功能：       FT62F21X_UART 功能演示
*    IC:          FT62F21X SOP8
*    晶振：       16M/4T
```

```
*     说明：    演示程序中波特率为 9600，RXIO（PA0）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。?
* Memory: Flash 1KX14b, EEPROM 128X8b, SRAM 64X8b
*                                  FT62F21X SOP8
*                                  ----------------
*   DemoPortOut ---------- |1(PA4)                (PA3)8 |----------NC
*   NC-------------------------|2(TKCAP)           (PA0)7 |----------NC
*   NC-------------------------|3(VDD)             (PA1)6 |----------NC
*   NC-------------------------|4(VSS)             (PA2)5 |----------DemoPortIn
*                                  ----------------
*/
//=============================================================
#INCLUDE   <FT62F21X.INC>;
;=============================================================
;RAM DEFINE
TEMP                EQU     0X40
TEMP1               EQU     0X41
TEMP2               EQU     0X42
W_TMP               EQU     0X43
S_TMP               EQU     0X44
UARTDATTEMP  EQU        0X45
buff                EQU     0X46
#DEFINE RXFLAG      buff,0
count               EQU     0x47
READPIN             EQU     0x48

;=============================================================
;CONSTANT DEFINE
;=============================================================
LSB                 EQU     0
MSB                 EQU     7
;=============================================================
;USER DEFINE
;=============================================================
#DEFINE     baud_rate       49          // 49  波特率为 9600
#DEFINE   TXIO          PORTA,4     //  串口的发送脚
#DEFINE   RXIO          PORTA,2     //  串口的接收脚
;=============================================================
;PROGRAM START
;=============================================================
    ORG         0000H
    LJUMP       RESTART
```

```
    ORG         0004H
    LJUMP       interrupt
;=======================================================================
;SYSTEM START
;=======================================================================
RESTART:
    BANKSEL     PORTA
    LCALL       INITIAL
    LCALL       TIMER0_INIT
    LCALL       PA_INT_INITIAL

    BSR         INTCON,T0IE
    BSR         INTCON,GIE

MAIN:
    NOP
    BTSS        RXFLAG
    LJUMP       MAIN
    LCALL       UART_Read_Byte
    LCALL       UART_Write_Byte
    LCALL       PA_INT_INITIAL
    BCR         RXFLAG
    LJUMP       MAIN
;*****************************************************************
;       中断程序
;*****************************************************************
interrupt:
    STR         W_TMP
    SWAPR       STATUS,W
    STR         S_TMP
    BCR         STATUS,RP0

    BANKSEL     INTCON
    BTSC        INTCON,T0IF
    LJUMP       TM0Interrupt
    BTSC        INTCON,PAIF
    LJUMP       GPIOInterrupt
    LJUMP       INT_RET

 TM0Interrupt:
    BCR         INTCON,T0IF
    LDWI        baud_rate
    STR         TMR0
    BCR         INTCON,T0IE
```

```
    LJUMP       INT_RET

GPIOInterrupt:
    BANKSEL         PORTA
    LDR             PORTA,W
    BANKSEL         READPIN
    STR         READPIN
    BANKSEL         INTCON
    BCR             INTCON,PAIF
    BTSC        RXIO
    LJUMP       INT_RET
    BCR             INTCON,PAIE
    BANKSEL     IOCA
    BCR             IOCA,IOCA2
    BANKSEL     PORTA
    bsr         RXFLAG
    LJUMP       INT_RET
INT_RET:
    SWAPR       S_TMP,0
    STR         STATUS
    SWAPR       W_TMP,1
    SWAPR       W_TMP,0
    RETI

;=====================================================================
;SYSTEM INITIAL
;=====================================================================
INITIAL:
    BANKSEL     PORTA
    LDWI        B'00000100'
    STR         PORTA           ;按键赋 1
    BANKSEL     TRISA
    LDWI        B'00000100'   ;PA2  输入
    STR         TRISA           ;SET IO Direction
    LDWI        B'00000100'
    STR         WPUA
    LDWI        B'00000000'
    STR         OPTION_REG    ;SET OPTION
    LDWI        B'01110000'
    STR         OSCCON          ;SET OSCCON
    BANKSEL     INTCON
    LDWI        B'00000000'
    STR         INTCON
CLEAR_RAM:
```

```
    LDWI        40H
    STR         FSR
CLEAR_RAM_LOOP:
    CLRR        INDF
    INCR        FSR,F
    LDWI        80H
    XORWR       FSR,W
    BTSS        STATUS,Z
    LJUMP       CLEAR_RAM_LOOP
    RET

;=====================================================================
;Timer0 init
;设置 TMR0 定时时长 104us=(1/16000000)*4*2*208(16M-2T-PSA 1:256- TMR0=255 溢出)
;TMR0 = 0;
;=====================================================================
TIMER0_INIT:
    BANKSEL     OPTION
    LDWI        B'00000000'
    STR         OPTION
    BANKSEL     INTCON
    LDWI        baud_rate
    STR         TMR0
    BCR         INTCON,T0IF
    BANKSEL     T0CON0
    BSR         T0CON0,T0ON
    RET
 /*-----------------------------------------------
 *   函数名: PA_INT_INITIAL
 *   功能：   PA 口电平变化中断初始化
 *   输入：   无
 *   输出：   无
 -----------------------------------------------*/
PA_INT_INITIAL:
    BANKSEL     TRISA
    BSR         TRISA,2         //端口设置为输入

    BANKSEL     PORTA
    LDR         PORTA,W
    BANKSEL     READPIN
    STR         READPIN

    BANKSEL     INTCON
    BCR         INTCON,PAIF     //中断标志清零
```

```
    BANKSEL     IOCA
    BSR         IOCA,IOCA2
    BANKSEL     INTCON
    BSR         INTCON,PAIE         //中断使能
    RET

/*---------------------------------------------
 *  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
 ---------------------------------------------*/
WaitTF0:
    BANKSEL     INTCON
    BSR         INTCON,T0IE     //启动 Timer0
    BTSC        INTCON,T0IE
    LJUMP       $-1
    BSR         INTCON,T0IE
    RET

/*---------------------------------------------
 *  函数名：  UART_Write_Byte
 *  功能：    UART 发送一个字节
 *  输入：    UARTDATTEMP
 *  输出：    无
 ---------------------------------------------*/
UART_Write_Byte:
    CLRR    count       //发送起始位
    BSR     TXIO
    LCALL   WaitTF0
    BCR     TXIO
    LCALL   WaitTF0
UART_Write_Byte_loop:   //发送 8 位数据位
    LDWI    0x08
    SUBWR   count,0
    BTSC    STATUS,0
    LJUMP   UART_Write_Byte_End
    INCR    count,1

    BTSS    UARTDATTEMP,0
    LJUMP   $+3
    BSR     TXIO
    LJUMP   $+2
    BCR     TXIO
```

```
    LCALL   WaitTF0
    BCR         STATUS,0
    RRR         UARTDATTEMP,1

    LJUMP   UART_Write_Byte_loop
UART_Write_Byte_End:    //发送结束位
    BSR     TXIO
    BCR         INTCON,T0IE
    RET

/*-------------------------------------------------
 *  函数名：  UART_Read_Byte
 *  功能：    UART 接收一个字节
 *  输入：    无
 *  输出：    UARTDATTEMP
 -------------------------------------------------*/
UART_Read_Byte:
    CLRR    count     //识别起始位
    CLRR    UARTDATTEMP
    LCALL   WaitTF0
 UART_Read_Byte_loop:
    LDWI    0x08
    SUBWR   count,0
    BTSC    STATUS,0
    LJUMP   UART_Read_Byte_end
    INCR    count,1

    BCR         STATUS,0
    RRR         UARTDATTEMP,1
    BTSC    RXIO
    BSR     UARTDATTEMP,7
    LCALL   WaitTF0
    LJUMP   UART_Read_Byte_loop

 UART_Read_Byte_end:
    BCR         INTCON,T0IE
    RET

;=======================================================================

 END
```

Fremont Micro Devices (SZ) Limited

#5-8, 10/F, Changhong Building, Ke-Ji Nan 12 Road, Nanshan District, Shenzhen, Guangdong 518057
Tel: (86 755) 86117811
Fax: (86 755) 86117810

Fremont Micro Devices (Hong Kong) Limited

#16, 16/F, Blk B, Veristrong Industrial Centre, 34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong
Tel: (852) 27811186
Fax: (852) 27811144

Fremont Micro Devices (USA), Inc.

42982 Osgood Road Fremont, CA 94539
Tel: (1-510) 668-1321
Fax: (1-510) 226-9918

Web Site: http://www.fremontmicro.com/