FT61F02X

UART Application note

Table of contents

FT61F02x UARTsapplication

1. UARTintroduce

UARTIs a universal serial data bus used for asynchronous communication. The bus bidirectional communication, can realize full-duplex transmission and reception.

UARTprotocol,The meanings of each of them are as follows:

start bit: emit a logic first"0"A signal that indicates the start of a transmitted character.

data bit: Immediately after the start bit, the number of data bits can be4,5,6,7,8etc. to form a character. Usually usedASCII The code is transmitted from the lowest bit and positioned by the clock.

parity bit: After adding this bit to the data bit, it makes "1"The number of bits should be even (even parity) or odd (odd parity) to verify the correctness of data transmission.

stop bit: It is the end mark of a character data. can be1bit,1.5bit,2bit high level. Since data is timed on the transmission line, and each device has its own clock, it is possible for two devices to be slightly out of sync in communication. so stopThe stop bit not only indicates the end of the transmission, but also provides an opportunity for the computer to correct clock synchronization. The more bits available for stop bits, the greater the tolerance for different clock synchronizations, but the slower the data transfer rate.

spare bit: in logic"1"Status, indicating that there is no data transmission on the current line.

baud rate: It is an indicator to measure the data transfer rate. Indicates the number of symbols transmitted per second (symbol). The amount of information (number of bits) represented by a symbol is related to the order of the symbol. For example, the data transfer rate is120characters/sec, transmitted using256order symbols, each symbol represents8bit, then the baud rate is120baud, the bit rate is120*8=960bit/s.

2. UARTsrelated settings

This example simulatesUARTsending part, use the timer to letTXIOGenerate a signal with the same frequency as the set baud rate; for the receiving part, useRXIOThe level shift interrupt identifies the start signal, and then identifies the data content based on the timer scan data.

explain toIC FT61F023 SOP16For demonstration, after receiving the data from the serial port assistant on the computer side, send the same data again.

Data cable for this programTXandRXCorrespondingIOPin:

```
#define    TXIO      PA4
#define    RXIO      PA2
```

3.Application example

```
//****************************************************
********** /*file name:TEST_61F02x_UART.c
* Function:     FT61F02x-UARTDemo
*IC:            FT61F023 SOP16
* Crystal:      16M/4T
* illustrate:   The baud rate in the demo program is9600,RXIO(PA2) After receiving data from the external serial port each time, TXIO (PA4)Send
*                the received data again. When the start bit is received, the level change interrupt is used to identify it, and the level change
*                interrupt is turned off later.
*
*                    FT61F023 SOP16
*                    ----------------------------
* VDD----------|1(VDD)      (VSS)16|----------GND
* NC------------|2(PA7)      (PA0)15|------------NC
* NC------------|3(PA6)      (PA1)14|------------NC
* NC------------|4(PA5)      (PA2)13|---------RXIO
* NC------------|5(PC3)      (PA3)12|------------NC
* NC------------|6(PC2)      (PC0)11|------------NC
* TXIO---------|7(PA4)       (PC1)10|------------NC
* NC------------|8(PC5)      (PC4)09|------------NC
*                    ----------------------------
*/
//**************************************************** ***********
# include   "SYSCFG.h";
# include   "FT61F02X.h";
//***********************Macro definition **************************
#define       uchar unsigned char

#define       TXIO    PA4                    //serial send pin
#define       RXIO    PA2                    //Receive pin of serial port

#define       Bord    49                     //Baud rate provided by timer
uchar         RXFLAG = 0;
uchar         ReadAPin;
/*---------------------------------------------- -
* Function name:POWER_INITIAL
* Function:    MCUinitialization function
* enter:       none
* output:      none
  ----------------------------------------------
--------- */ void POWER_INITIAL(void)
{
      OSCCON = 0B01110001;                   //IRCF=111=16MHz/4=4MHz,0.25µs//Temporarily
      INTCON = 0;                            disable all interrupts
```

```
    PORTA = 0B00000000;
    TRISA = 0B00000100;                 //PAinput Output0-output1-enter
    PORTC = 0B00000000;
    TRISC = 0B00000000;                 //PCinput Output0-output1-enter

    WPUA = 0B00000100;                  //PAPort pull-up control1-pull up0-close pull //PCPort
    WPUC = 0B00000000;                  pull-up control1-pull up0-close pull //Bit3=0, Timer0
    OPTION = 0B00000000;             MODE, PS=000=Timer0 RATE 1:2
    MSCKCON = 0B00000000;
    //Bit6->0,prohibitPA4,PC5Regulated output
    //Bit5->0,TIMER2the clock isFosc //Bit4->0,
    prohibitLVR

    CMCON0 = 0B00000111;                //turn off the comparator,Cxfor numbersIOmouth
}
/*------------------------------------------------ ----
* Function name:TIMER0_INITIAL
* Function: Initialize and set the timer0
* set upTMR0Timing duration =(1/System clock frequency)*instruction cycle*prescaler value*TMR0
*                     =(1/16000000)*4*2*208=104us ----------------------------------- ----------------*/
void TIMER0_INITIAL (void) {

    OPTION = 0B00000000;
    //Bit5:       T0CS Timer0Clock Source Selection
    //          1-External pin level changeT0CKI 0-internal clock (FOSC/2)
    //Bit4:       T0CKIpin trigger mode1-falling edge PSA0-rising edge Prescaler Allocation
    //Bit3:       Bits0-Timer0                     1-WDT
    //Bit[2:0]:   PS 8prescaler000 - 1:2

    TMR0 = Bord;
    T0IF = 0;                           //emptyT0software interrupt
}
/*------------------------------------------------ -
* Function name:PA2_Level_Change_INITIAL
* Function:    PAport(PA2)Interrupt-on-change initialization None
* enter:
* output:    none
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - */ void PA2_Level_Change_INITIAL(void) {

    TRISA2 =1;                          //set upPA2enter
    CMCON0 = 0B00000111;                //turn off the comparator,Cxfor numbersIOMouth dedicated //
    ReadAPin = PORTA;                   clearPAinterrupt-on-change
    PAIF = 0;                               //clearPA INTinterrupt flag //
    IOCA2 = 1;                              EnablePA2interrupt-on-change //
    PAIE = 1;                               EnablePA INTto interrupt
}
/*------------------------------------------------ -
* Function name:interrupt ISR
* Function:   Interrupt handling, including timers0Interrupts and External Interrupts
* enter:      none
* output:     none
* illustrate: timer generated104μsinterrupt, corresponding9600baud rate1000000÷9600=104
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - * / void interrupt ISR(void)
{
    //timer0at the break
    if(T0IE && T0IF)                        //104μs
    {
        TMR0 = Bord;                   //note: yesTMR0reassignmentTMR0No change in two cycles
        T0IF = 0;
        T0IE = 0;
    }

    //PAinterrupt-on-change
    if(PAIE && PAIF)
    {
        ReadAPin = PORTA;                   //readPORTAdata clearingPAIFthe sign
        PAIF = 0;                           //clearPAIFflag bit
        if(RXIO == 0)
        {
            PAIE = 0;                       //Temporarily bannedPAinterrupt-on-change
            IOCA2 = 0;                      //prohibitPA2interrupt-on-change
            RXFLAG = 1;
        }
    }
}
/*------------------------------------------------ -
* Function name:WaitTF0
* Function:   After the query timer overflows, turn off the timer in the interrupt, and turn on the timer again
* enter:      without
* output:     none
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - */ void WaitTF0( void )
{
     while(T0IE);
     T0IE=1;
```

```
}
/*----------------------------------------------- -
* Function name:WByte
* Function:   UARTsend a byte
* enter:      input
* output:     none
  --------------------------------------------------

- - - - - * / void WByte(uchar input)
{
                                                  //send start bit
     uchar i=8;
     TXIO = 1;
     TMR0 = Bord;
     T0IE = 1;
     WaitTF0();
     TXIO=0;
     WaitTF0();
                                                  //send8bit data bit
     while(i--)
     {
          if(input&0x01)                          //pass low first
          {
               TXIO=1;
          }
          else
          {
               TXIO = 0;
          }
          WaitTF0();
          input=input>>1;
     }
                                                  //Send check digit (none)
                                                  //send end bit
     TXIO=(bit)1;
     T0IE=0;
}
/*----------------------------------------------- -
* Function name:RByte
* Function:   UARTreceive a byte
* enter:      none
* output:     output
  --------------------------------------------------

- - - - - */uchar RByte()
{
     uchar Output=0;
     uchar i=8;
     T0IE=1;                                     //start upTimer0
     TMR0 = Bord;
     WaitTF0();
     T0IE=1;                                     //start upTimer0
     TMR0 = Bord;
     WaitTF0();                                  //waited for start bit
                                                  //send8bit data bit
     while(i--)
     {
          Output >>=1;
          if(RXIO)
          {
               output      |=0x80;         //close low first
          }
          WaitTF0();                          //inter-bit delay
     }
     T0IE=0;                                     //stopTimer0
     return Output;
}
/*------------------------------------------------ -
* Function name:main
* Function:     main function
* enter:     none
* output:    none
  ---------------------------------------------------

- - - - - */ void main(void)
{
     uchar rdata = 0;
     POWER_INITIAL();
     TIMER0_INITIAL();
     PA2_Level_Change_INITIAL(); GIE
     =   1;                                      //open interrupt
     T0IE = 1;                                   //ON Timer/Counter0to interrupt
     while(1)
     {
          if(RXFLAG)                          //The falling edge of the external interrupt triggers the
          {
               rdata = RByte();
               WByte(rdata);

               IOCA2 =1;                   //EnablePA2interrupt-on-change //
               PAIE=1;                     EnablePA INTto interrupt
```

```
                    RXFLAG = 0;
                }
            }
        }
```

contact information

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, Veristong 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. 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.