FT64F0AX

IR_Send Application note

目录

FT64F0Ax IR_Send 应用

1. IR 介绍

一个通用的红外遥控系统由发射和接收两大部分组成，如图 1 所示：

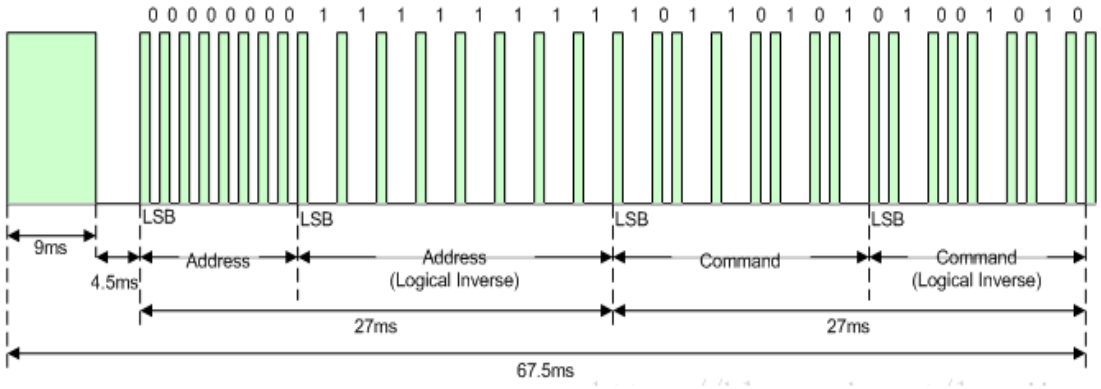

图 1-1

发射部分主要包括键盘矩阵、编码调制、红外发射管； 接收部分包括光、电信号的转换以及放大、解调、解码电路。

举例来说，通常我们家电遥控器信号的发射，就是将相应按键所对应的控制指令和系统码(由 0 和 1 组成的序列)，调制在 32~56kHz 范围内的载波上（目的为：抗干扰及低功率），然后经放大（接三极管）、驱动红外发射管（透明的头）将信号发射出去。

2. IR Send 相关寄存器的设置

本例使用两个定时器，一个是产生38kHz载波频率，另一个定时器是做时基，定时时长是560μS，红外信号的高低电平是560μS的整数倍。

定时器 2 为 16 位，配置成 13μs 中断产生 38K 信号

定时器 4 为 8 位，配置成 560μs 中断一次。

本讲解以 IC FT64F0A5 TSSOP20 为示范，每 2.5 秒钟会发出一次信号，信号的码为 IRData[4] = {0x55,0xAA,0x01,0xFE}

3. 应用范例

```
//*********************************************************
/* 文件名：TEST_64F0Ax_IR_Send.c
* 功能：    FT64F0Ax_IR_Send 功能演示
*  IC：     FT64F0A5  TSSOP20
* 内部：    16M
* 说明：    演示程序中，IR 红外是采用 6122 协议，起始信号是 9ms 低电平，
*           到 4.5ms 高电平，再到低 8 位用户识别码，到高 8 位的用户识别码，
*           8 位数据码，8 位数据码的反码。RXIO（RC3）每次收到遥控器发过来的数据后，
*           数据是合法（两对补码，不对内容判断）的话，LED(RB3)开关状态就改变一次。
*
*            FT64F0A5  TSSOP20
*                ------------------
* NC----------|1(PA5)       (PA4)20|----------NC
* NC----------|2(PA6)       (PA3)19|----------NC
* NC----------|3(PA7)       (PA2)18|----------NC
* NC----------|4(PC0)       (PA1)17|----------NC
* NC----------|5(PC1)       (PA0)16|----------NC
* NC----------|6(PB7)       (PB0)15|----------NC
* GND---------|7(GND)       (PB1)14|----------NC
* NC----------|8(PB6)       (PB2)13|----------NC
* VDD---------|9(VDD)       (PB3)12|--IRSendIO
* NC----------|10(PB5)      (PB4)11|----------NC
*                  ------------------
*/
//*********************************************************
#include    "SYSCFG.h";
#include    "FT64F0AX.h";
//**********************宏定义*****************************
#define     uchar           unsigned char
#define     uint            unsigned int

#define     IRSendIO        PB3             //LED 指示灯的 IO

#define     IRSend_HIGH_1   1               //560μs
#define     IRSend_LOW_1    3               //16800μs

#define     IRSend_HIGH_0   1               //560μs
#define     IRSend_LOW_0    1               //560μs

#define     IRSend_PIN_1    T2UIE=1         //通过 PA4 输出载波
#define     IRSend_PIN_0    T2UIE=0         //通过 PA4 输出载波

#define     Status_NOSend   0               //不发送的状态
```

```
#define     Status_Head     1               //发送引导码的状态
#define     Status_Data     2               //发送数据的状态

uchar   IRSendStatus;                       //发送状态，是发送引导码还是数据
uchar   IRSendData;                         //发送的数据中转变量
uchar   TxBit=0,TxTime=0;;
uchar   Sendbit=0;
uchar   level0,level1;                      //一位数据里发送与关闭的时间值
bit     SendLastBit = 0;
uchar   SaveLastBit=0;
uint    SYSTime5S=0;                        //系统时间，5s 发送一次

uchar IRData[4]={0x00,0xff,0x40,0xbf};      //需要发送的 4 个数据
/*---------------------------------------------
 * 函数名：POWER_INITIAL
 * 功能：   上电系统初始化
 * 输入：   无
 * 输出：   无
 ---------------------------------------------*/
 void POWER_INITIAL(void)
 {
    OSCCON=0B01110001;                      //系统时钟选择为内部振荡器 16MHz,分频比为 1:1
    PCKEN|=0B00001100;                      //使能 TIMER2 和 TIMER4 时钟模块

    INTCON=0;                               //禁止所有中断

    PORTA=0B00000000;
    PORTB=0B00000000;
    PORTC=0B00000000;

    WPUA=0B00000000;                        //弱上拉的开关，0-关，1-开
    WPUB=0B00000000;
    WPUC=0B00000000;

    WPDA=0B00000000;                        //弱下拉的开关，0-关，1-开
    WPDB=0B00000000;
    WPDC=0B00000000;

    TRISA=0B00000000;                       //输入输出设置，0-输出，1-输入
    TRISB=0B00000000;                       //PB3-输出
    TRISC=0B00000000;

    PSRC0=0B11111111;                       //源电流设置最大
    PSRC1=0B11111111;
```

```
    PSRC2=0B00001111;

    PSINK0=0B11111111;                  //灌电流设置最大
    PSINK1=0B11111111;
    PSINK2=0B00000011;

    ANSELA=0B00000000;                  //设置对应的 IO 为数字 IO
}
/*-----------------------------------------------
* 函数名：TIMER4_INITIAL
* 功能：   初始化定时器 4，设置 TIMER4 定时时长 560μs
* 输入：   无
* 输出：   无
-----------------------------------------------*/
void TIMER4_INITIAL(void)
{
    TIM4CR1=0B00000101;                 //允许自动装载，使能计数器
    TIM4IER=0B00000001;                 //允许更新中断
    TIM4SR=0B00000000;
    TIM4EGR=0B00000000;
    TIM4CNTR=0;
    TIM4PSCR=0B00000110;                //预分频器的值
    TIM4ARR=140;                        //自动装载值
}
 /*-----------------------------------------------
* 函数名：TIMER2_INITIAL
* 功能：   初始化定时器 2，用定时器 2 作为 38kHz 红外载波发生器，从 PA4 输出
* 输入：   无
* 输出：   无
-----------------------------------------------*/
void TIMER2_INITIAL(void)
{
    CKOCON=0B00100000;                  //Timer2 倍频时钟占空比调节位 4ns 延迟
    TCKSRC=0B00110000;                  //Timer2 时钟源为 HIRC 的 2 倍频

    TIM2CR1=0B10000101;                 //允许自动装载，使能计数器

    TIM2IER=0B00000000;                 //禁止所有中断

    TIM2ARRH=0x01;                      //自动装载高 8 位 01H
    TIM2ARRL=0xA0;                      //自动装载低 8 位 A0H

    INTCON=0B11000000;                  //使能总中断和外设中断
}
```

```
/*-------------------------------------------------
* 函数名：SendCtrl
* 功能：   发送数据函数
* 输入：   无
* 输出：   无
-------------------------------------------------*/
void SendCtrl(void)
{
    if(IRSendStatus==Status_NOSend)          //不发送的状态
    {
        IRSend_PIN_0;
        Sendbit=0;
        TxTime=0;
    }
    else if(IRSendStatus==Status_Head)       //发送引导码
    {
        TxTime++;
        if(TxTime<17)                        //发送 9ms 信号
        {
            IRSend_PIN_1;
        }
        else if(TxTime<24)                   //4.5ms 不发送
        {
            IRSend_PIN_0;
        }
        else
        {
            TxTime=0;
            IRSendStatus=Status_Data;
        }
        IRSendData=IRData[0];
        TxBit=0x01;
    }
    else if(IRSendStatus==Status_Data)       //发送数据
    {
        if(IRSendData&TxBit)                 //1，是 1:3 的时间
        {
            level1=IRSend_HIGH_1;
            level0=IRSend_LOW_1;
        }
        else                                 //0，是 1:1 的时间
        {
            level1=IRSend_HIGH_0;
            level0=IRSend_LOW_0;
```

```
}
TxTime++;
if(TxTime<=level1)                    //发送信号
{
    IRSend_PIN_1;
}
else if(TxTime<=(level0+level1))      //不发送信号
{
    IRSend_PIN_0;
}
else if(Sendbit<4)                    //发送 4 位数据未完成
{
    TxTime=1;
    IRSend_PIN_1;
    SaveLastBit=IRSendData&TxBit;
    TxBit<<=1;
    if(TxBit==0x00)                   //发送完一个字节
    {
        TxBit=0x01;
        Sendbit++;
        IRSendData=IRData[Sendbit];
        if(Sendbit>3)                 //最后一位要注意，因为发送完了还要有一个脉冲
        {
            SendLastBit=1;
        }
    }
}
else                                  //数据完成了，要补脉冲
{
    if(SendLastBit)
    {
        TxTime++;
        if(SaveLastBit)
        {
            if(TxTime<3)
            {
                IRSend_PIN_0;
            }
            else if(TxTime<4)
            {
                IRSend_PIN_1;
            }
            else
            {
```

```
                        IRSend_PIN_0;
                        IRSendStatus=Status_NOSend;
                        IRSend_PIN_0;
                        SendLastBit=0;
                        TxBit=0;
                        TxTime=0;
                    }
                }
                else
                {
                    if(TxTime<5)
                    {
                        IRSend_PIN_0;
                    }
                    else if(TxTime<6)
                    {
                        IRSend_PIN_1;
                    }
                    else
                    {
                        IRSend_PIN_0;
                        IRSendStatus=Status_NOSend;
                        IRSend_PIN_0;
                        SendLastBit=0;
                        TxBit=0;
                        TxTime=0;
                    }
                }
            }
        }
    }
}
/*-----------------------------------------------
 * 函数名：interrupt ISR
 * 功能：   定时器 4 与 2 中断和外部中断
 * 输入：   无
 * 输出：   无
 -----------------------------------------------*/
void interrupt ISR(void)
{
    //定时器 4 的中断处理
    if(T4UIE&&T4UIF)                    //560μs 中断一次，红外每一位都是 560μs 的倍数
    {
        T4UIF=1;                        //写 1 清零标志位
```

```
            SendCtrl();
            SYSTime5S++;
        }

        //定时器 2 的中断处理
        if(T2UIE&&T2UIF)
        {
            T2UIF=1;                        //写 1 清零标志位

            IRSendIO=~IRSendIO;             //翻转电平，产生 38kHz
        }
}

/*---------------------------------------------
 * 函数名：main
 * 功能：   主函数
 * 输入：   无
 * 输出：   无
 ---------------------------------------------*/
void main(void)
{
    POWER_INITIAL();                        //系统初始化
    TIMER4_INITIAL();
    TIMER2_INITIAL();
    GIE=1;                                  //开启总中断

    while(1)
    {
        if(SYSTime5S>5000)                  //定时 2.5s
        {
            SYSTime5S=0;
            IRSendStatus=Status_Head;
        }
    }
}
```

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