

FT61F02X

IR_Receive Application note

目录

1. IR 介绍.....	3
2. 应用范例.....	4
联系信息	9

FT61F02x IR_Receive 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 1-1 所示：

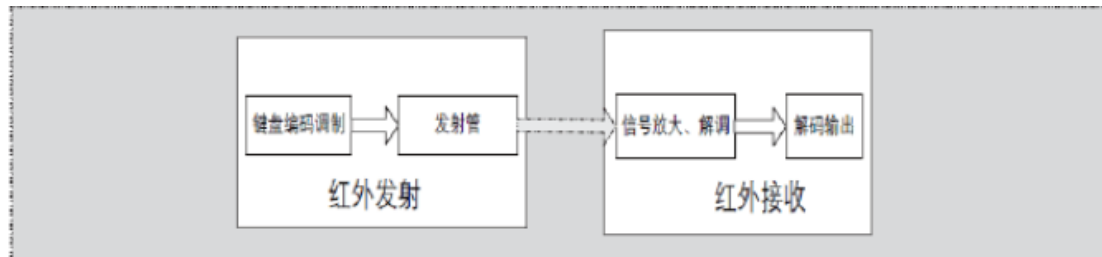


图 1-1

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。

本讲解以 IC FT61F023 SOP16 为示范，采用一体的红外接收头，接收头输出脚连到 MCU 的 IO 口，IO 口通过识别高低电平时间长短来解码，当收到的数据是合法的，指示 LED 的状态（开与关）会翻转一次。接收的 IO 口使用电平变化中断来识别信号，并使用定时器记录电平的时间长短。

本程序 IR 接收与 LED 所对应的 IO 引脚：

```

#define IRRIO    PA4
#define LED      PA2
  
```

2. 应用范例

```
//*****
/* 文件名: TEST_61F02x_IR_Receive.c
* 功能:    FT61F02x-红外接收 功能演示
* IC:      FT61F023 SOP16
* 晶振:    16M/4T
* 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 (PA4) 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(PA2)开关状态就改变一次。
*
*          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|-----LED
* NC-----|5(PC3) (PA3)12|-----NC
* NC-----|6(PC2) (PC0)11|-----NC
* IRRIO-----|7(PA4) (PC1)10|-----NC
* NC-----|8(PC5) (PC4)09|-----NC
*
*/
//*****
#include "SYSCFG.h";
#include "FT61F02X.h";
//*****宏定义*****
#define uchar unsigned char

#define IRRIO PA4      //IR 的接收脚
#define LED PA2        //LED 指示灯的 IO

uchar IRbitNum = 0;    //用于记录接收到第几位数据了
uchar IRbitTime = 0;   //用于计时一位的时间长短
uchar IRDataTimer[4];  //存出来的 4 个数据
uchar bitdata=0x01;    //用于按位或的位数据
uchar ReceiveFinish = 0; //用于记录接收完成
uchar ReadAPin = 0;    //用于读取 IO 口状态, 电平变化中断标志清除
/*-----
* 函数名: POWER_INITIAL
* 功能:   MCU 初始化函数
* 输入:   无
* 输出:   无
*-----*/
```

```

void POWER_INITIAL(void)
{
    OSCCON = 0B01110001; //IRCF=111=16MHz/4=4MHz,0.25μs
    INTCON = 0;          //暂禁止所有中断
    PORTA = 0B00000000;
    TRISA = 0B00010000;  //PA 输入输出 0-输出 1-输入
                        //PA2->输出 PA4->输入

    PORTC = 0B00000000;
    TRISC = 0B00000000;  //PC 输入输出 0-输出 1-输入
    WPUA = 0B00000000;  //PA 端口上拉控制 1-开上拉 0-关上拉
    WPUC = 0B00000000;  //PC 端口上拉控制 1-开上拉 0-关上拉

    OPTION = 0B00001000; //Bit3=1,WDT MODE,PS=000=WDT RATE 1:1
    MSCKCON = 0B00000000;
    //Bit6->0,禁止 PA4, PC5 稳压输出
    //Bit5->0,TIMER2 时钟为 Fosc
    //Bit4->0,禁止 LVR
    CMCON0 = 0B00000111; //关闭比较器, CxIN 为数字 IO 口
}

/*-----
* 函数名称: 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)*指令周期*预分频值*140
*              =(1/16000000)*4*16*140=560μs
*-----*/

void TIMER0_INITIAL (void)
{
    OPTION = 0B00000011;
    //Bit5:   T0CS Timer0 时钟源选择
    //        1-外部引脚电平变化 T0CKI 0-内部时钟(FOSC/2)
    //Bit4:   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]: PS2 8 个预分频比 011 - 1:16

    TMR0 = 118;
    T0IF = 0;          //清空 T0 软件中断
}

/*-----
* 函数名: PA_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 PA_Level_Change_INITIAL(void)
{
    TRISA4 = 1;           //SET PA4 INPUT
    ReadAPin = PORTA;     //清 PA 电平变化中断
    PAIF = 0;             //清 PA INT 中断标志位
    IOCA4 = 1;            //使能 PA4 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)      //560μs
    {
        TMR0 = 118;       // 注意:对 TMR0 重新赋值 TMR0 在两个周期内不变化

        T0IF = 0;
        IRbitTime++;
        if(IRbitTime > 50)
        {
            T0IE = 0;
            IRbitTime = 0;
        }
    }
    //PA 电平变化中断
    if(PAIE && PAIF)
    {
        ReadAPin = PORTA; //读取 PORTA 数据清 PAIF 标志
        PAIF = 0;
        if(IRRIO == 0)
        {
            T0IE = 1;
            if(IRbitTime > 21)
            {
                IRDataTimer[0] = 0;
                IRDataTimer[1] = 0;
                IRDataTimer[2] = 0;
                IRDataTimer[3] = 0;
                IRbitNum = 0;
            }
        }
    }
}
```

```

        bitdata = 0x00;
    }
    else if(IRbitTime > 3)
    {
        IRDataTimer[IRbitNum-1] |= bitdata;
    }
    IRbitTime = 0;
    bitdata<<=1;
    if(bitdata == 0)
    {
        bitdata = 0x01;
        IRbitNum++;
    }
    if(IRbitNum > 4)
    {
        IRbitNum = 0;
        T0IE = 0;
        ReceiveFinish = 1;
    }
}
}
}
/*-----
* 函数名: main
* 功能: 主函数
* 输入: 无
* 输出: 无
-----*/
void main(void)
{
    uchar rdata1,rdata2;
    POWER_INITIAL();
    TIMER0_INITIAL();
    PA_Level_Change_INITIAL();
    GIE = 1;           //开中断
    while(1)
    {
        if(ReceiveFinish)
        {
            ReceiveFinish = 0;
            rdata1 = 0xFF - IRDataTimer[0];
            rdata2 = 0xFF - IRDataTimer[2];
            if((rdata1 == IRDataTimer[1])&&(rdata2 == IRDataTimer[3]))
            {

```

```
        LED = ~LED; //翻转电平
    }
}
}
```


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.