FT64F0AX

IR_Receive Application note

目录

FT64F0Ax 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 所示：

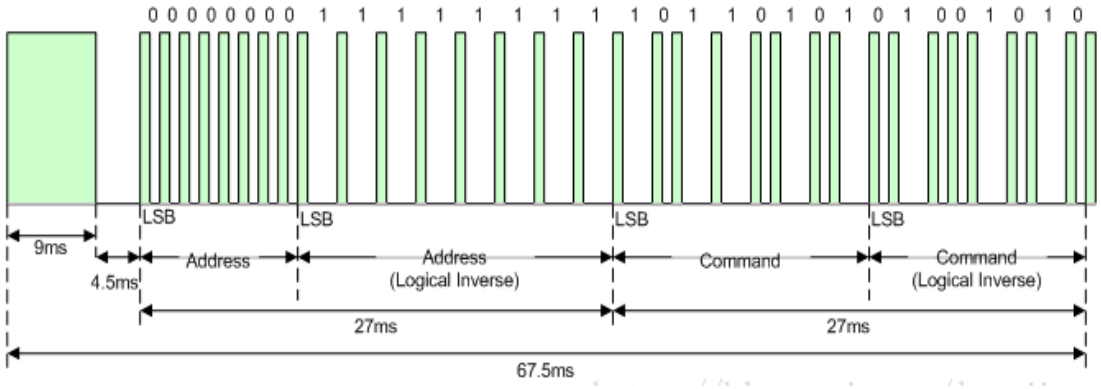

图 1-1

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 FT64F0A5 TSSOP20 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。

2. 应用范例

```
//=============================================================
/* 文件名：ASM_64F0Ax_IR_Receive.ASM
* 功能：    FT64F0Ax_IR_Receive 功能演示
*  IC：     FT64F0A5      TSSOP20
* 内部：    16M/4T
* 说明：    接收红外发送过来的两对互补字节，并与自己所需要的比对，确认正确后
*           执行所需操作
*
*           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|---------ir_rx
* NC----------|10(PB5)       (PB4)11|----------NC
*                ------------------
*/
//=============================================================
#INCLUDE   <FT64F0AX.INC>;
;=============================================================
;RAM DEFINE
;=============================================================
    TEMP            EQU     0X40
    W_TMP           EQU     0X43
    S_TMP           EQU     0X44

    ir_counttemp    EQU     0X47
    buff            EQU     0X48
    #define         f_finish    buff,0
    #define         f_receive   buff,1
    ir_count        EQU     0X49
    countbyte       EQU     0X4A
    IRDATE0         EQU     0X4B
    IRDATE1         EQU     0X4C
    IRDATE2         EQU     0X4D
    IRDATE3         EQU     0X4E
;=============================================================
;CONSTANT DEFINE
```

```
;=============================================================
    INTCON_DEF      EQU     B'01000000'     ;使能外设中断
    OSCCON_DEF      EQU     B'01110001'     ;16MHz,1:1

    WPUA_DEF        EQU     B'00000000'     ;弱上拉的开关，0-关，1-开
    WPUB_DEF        EQU     B'00000000'
    WPUC_DEF        EQU     B'00000000'

    WPDA_DEF        EQU     B'00000000'     ;弱下拉的开关，0-关，1-开
    WPDB_DEF        EQU     B'00000000'
    WPDC_DEF        EQU     B'00000000'

    TRISA_DEF       EQU     B'00000000'     ;输入输出设置，0-输出，1-输入
    TRISB_DEF       EQU     B'00000000'     ;PB3-OUT
    TRISC_DEF       EQU     B'00000000'

    PSRC0_DEF       EQU     B'11111111'     ;源电流设置最大
    PSRC1_DEF       EQU     B'11111111'
    PSRC2_DEF       EQU     B'00001111'

    PSINK0_DEF      EQU     B'11111111'     ;灌电流设置最大
    PSINK1_DEF      EQU     B'11111111'
    PSINK2_DEF      EQU     B'00000011'

    ANSELA_DEF      EQU     B'00000000'     ;设置对应的 IO 为数字 IO

    PCKEN_DEF       EQU     B'00000010'     ;使能 Timer1 时钟模块
    CKOCON_DEF      EQU     B'00100000'
    ;Timer1 倍频时钟占空比调节位 4ns 延迟
    TCKSRC_DEF      EQU     B'00000001'     ;Timer1 时钟源为 HIRC

    TIM1CR1_DEF     EQU     B'10000101'
    ;允许自动装载，使能计数器，边沿对齐向上计数

    TIM1IER_DEF     EQU     B'00000001'     ;允许更新中断

    TIM1ARRH_DEF    EQU     0x10            ;自动装载周期高 8 位 10H
    TIM1ARRL_DEF    EQU     0x0C            ;自动装载周期低 8 位 0CH
;=============================================================
;USER   DEFINE
;=============================================================
#DEFINE         ir_rx           PORTB,3
;=============================================================
;PROGRAM  START
```

```
;=============================================================
    ORG         0x0000
    LJUMP       RESTART
    ORG         0x0004
    STR         W_TMP
    SWAPR       STATUS,W
    STR         S_TMP
    LJUMP       INT_PROGRAM
;=============================================================
;SYSTEM    START
;=============================================================
RESTART:
    BANKSEL     PORTA
    LCALL       INITIAL
    LCALL       TIMER1_INITIAL

    BANKSEL     NTCON
    BSR         INTCON,GIE          ;使能全局中断
    BANKSEL     TIM1IER
    BSR         TIM1IER,T1UIE       ;允许 TIM1 中断
MAIN:
    CLRWDT
    LCALL       scanky_ir
    LCALL       ir_data_compare
    LJUMP       MAIN
;=============================================================
; INT_PROGRAM
;=============================================================
INT_PROGRAM:
    BANKSEL     TIM1SR1
    BTSC        TIM1SR1,T1UIF
    LJUMP       TIM1Interrupt
    LJUMP       INT_RET

TIM1Interrupt:
    BANKSEL     TIM1SR1
    BSR         TIM1SR1,T1UIF

    BANKSEL     ir_count
    INCR        ir_count,1
    LJUMP       INT_RET

INT_RET:
    SWAPR       S_TMP,0
```

```
    STR             STATUS
    SWAPR           W_TMP,1
    SWAPR           W_TMP,0
    RETI
;=========================================================
;SYSTEM    INITIAL
;=========================================================
INITIAL:
    BANKSEL         OSCCON
    LDWI            OSCCON_DEF
    STR             OSCCON

    BANKSEL         INTCON
    LDWI            INTCON_DEF
    STR             INTCON

    BANKSEL         PORTA
    LDWI            0X00
    STR             PORTA
    STR             PORTB
    STR             PORTC

    BANKSEL         TRISA
    LDWI            TRISA_DEF
    STR             TRISA
    LDWI            TRISB_DEF
    STR             TRISB
    LDWI            TRISC_DEF
    STR             TRISC

    BANKSEL         WPUA
    LDWI            WPUA_DEF
    STR             WPUA
    LDWI            WPUB_DEF
    STR             WPUB
    LDWI            WPUC_DEF
    STR             WPUC

    BANKSEL         WPDA
    LDWI            WPDA_DEF
    STR             WPDA
    LDWI            WPDB_DEF
    STR             WPDB
    LDWI            WPDC_DEF
```

```
    STR             WPDC

    BANKSEL         PSRC0
    LDWI            PSRC0_DEF
    STR             PSRC0
    LDWI            PSRC1_DEF
    STR             PSRC1
    LDWI            PSRC2_DEF
    STR             PSRC2

    BANKSEL         PSINK0
    LDWI            PSINK0_DEF
    STR             PSINK0
    LDWI            PSINK1_DEF
    STR             PSINK1
    LDWI            PSINK2_DEF
    STR             PSINK2

    BANKSEL         ANSELA
    LDWI            ANSELA_DEF
    STR             ANSELA
;***************Clear   SRAM*********************
    BANKSEL         PORTA
    LDWI            0X00
    STR             FSR0H
 CLEAR_RAM_BANK0:
    LDWI            20H
    STR             FSR0L
CLEAR_RAM_BANK0_LOOP:
    CLRR            INDF0
    INCR            FSR0L,F
    LDWI            80H
    XORWR           FSR0L,W
    BTSS            STATUS,Z
    LJUMP           CLEAR_RAM_BANK0_LOOP
CLEAR_RAM_BANK1:
    LDWI            0A0H
    STR             FSR0L
CLEAR_RAM_BANK1_LOOP:
    CLRR            INDF0
    INCR            FSR0L,F
    LDWI            00H
    XORWR           FSR0L,W
    BTSS            STATUS,Z
```

```
    LJUMP       CLEAR_RAM_BANK1_LOOP
    INCR        FSR0H,F
CLEAR_RAM_LOOP:
    LDWI        10
    SUBWR       FSR0H,W
    BTSS        STATUS,0
    LJUMP       CLEAR_RAM_BANK0
    RET
;=============================================================
;函数名：TIMER1_INITIAL
;功能：   TIMER1 初始化，设置 TIMER1 定时时长 256μs
;=============================================================
TIMER1_INITIAL:
    BANKSEL     PCKEN
    LDWI        PCKEN_DEF
    STR         PCKEN
    BANKSEL     CKOCON
    LDWI        CKOCON_DEF
    STR         CKOCON

    BANKSEL     TCKSRC
    LDWI        TCKSRC_DEF
    STR         TCKSRC

    BANKSEL     TIM1CR1
    LDWI        TIM1CR1_DEF
    STR         TIM1CR1
    LDWI        TIM1IER_DEF
    STR         TIM1IER

    BANKSEL     TIM1ARRH
    LDWI        TIM1ARRH_DEF
    STR         TIM1ARRH
    LDWI        TIM1ARRL_DEF
    STR         TIM1ARRL

    RET
;=============================================================
;函数:  scanky_ir
;功能：  采集红外接收器收到的数据
;=============================================================
scanky_ir:
    BANKSEL     PORTB
    BTSC        ir_rx
```

```
    RET
    CLRR        ir_count
    BSR         f_finish

scanky_ir_one:
    CLRWDT
    LDWI        0X27
    SUBWR       ir_count,0
    BTSC        STATUS,0
    LJUMP       scanky_ir_wrong
    BTSS        ir_rx
    LJUMP       scanky_ir_one

    LDR         ir_count,0
    STR         ir_counttemp        ;判断启动时候的 9ms 的低电平
    CLRR        ir_count
    LDWI        0X20
    SUBWR       ir_counttemp,0
    BTSS        STATUS,0
    LJUMP       scanky_ir_wrong
    LDWI        0X27
    SUBWR       ir_counttemp,0
    BTSC        STATUS,0
    LJUMP       scanky_ir_wrong

scanky_ir_two:
    CLRWDT
    LDWI        0X13
    SUBWR       ir_count,0
    BTSC        STATUS,0
    LJUMP       scanky_ir_wrong
    BTSC        ir_rx
    LJUMP       scanky_ir_two

    LDR         ir_count,0
    STR         ir_counttemp        ;判断启动时候的 4.5ms 的高电平
    CLRR        ir_count
    LDWI        0X0F
    SUBWR       ir_counttemp,0
    BTSS        STATUS,0
    LJUMP       scanky_ir_wrong
    LDWI        0X13
    SUBWR       ir_counttemp,0
    BTSC        STATUS,0
```

```
    LJUMP       scanky_ir_wrong

    CLRR        IRDATE0
    CLRR        IRDATE1
    CLRR        IRDATE2
    CLRR        IRDATE3
    BCR         f_receive
    LDWI        0X21
    STR         countbyte

scanky_ir_thr:
    CLRWDT
    LDWI        0X05
    SUBWR       ir_count,0
    BTSC        STATUS,0
    LJUMP       scanky_ir_wrong
    BTSS        ir_rx
    LJUMP       scanky_ir_thr

    LDR         ir_count,0
    STR         ir_counttemp
    CLRR        ir_count
    LDWI        0X05
    SUBWR       ir_counttemp,0
    BTSC        STATUS,0
    LJUMP       scanky_ir_wrong

scanky_ir_fou:
    CLRWDT
    LDWI        0X08
    SUBWR       ir_count,0
    BTSC        STATUS,0
    LJUMP       scanky_ir_wrong
    BTSC        ir_rx
    LJUMP       scanky_ir_fou

    DECRSZ      countbyte,1
    LJUMP       $+2
    LJUMP       scanky_ir_receive_end

    BCR         STATUS,0
    RRR         IRDATE3,1
    RRR         IRDATE2,1
    RRR         IRDATE1,1
```

```
    RRR         IRDATE0,1
    LDR         ir_count,0
    STR         ir_counttemp
    CLRR        ir_count
    LDWI        0X05
    SUBWR       ir_counttemp,0
    BTSS        STATUS,0
    LJUMP       scanky_ir_thr
    BSR         IRDATE3,7
    LJUMP       scanky_ir_thr

scanky_ir_wrong:
    BCR         f_receive
    BCR         f_finish
    RET

scanky_ir_receive_end:
    BSR         f_receive
    BCR         f_finish
    RET
;============================================================
;函数名：sir_data_compare
;功能：    比较采集到的数据
;============================================================
ir_data_compare:
    BTSS        f_receive
    RET
    BCR         f_receive

    LDWI        0X00
    XORWR       IRDATE0,0
    BTSS        STATUS,2
    RET

    LDWI        0XFF
    XORWR       IRDATE1,0
    BTSS        STATUS,2
    RET

    LDWI        0X0A
    XORWR       IRDATE2,0
    BTSS        STATUS,2
    RET
```

```
LDWI        0XF5
XORWR       IRDATE3,0
BTSS        STATUS,2
RET
NOP
LDWI        0XFF
STR         TEMP
RET

END
```

联系信息

Fremont Micro Devices (SZ) 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) Corporation

#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/