

FT64F0AX

IR_Send Application note

目录

1. IR 介绍.....	3
2. IR SEND 相关寄存器的设置	3
3. 应用范例.....	4
联系信息	17

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：

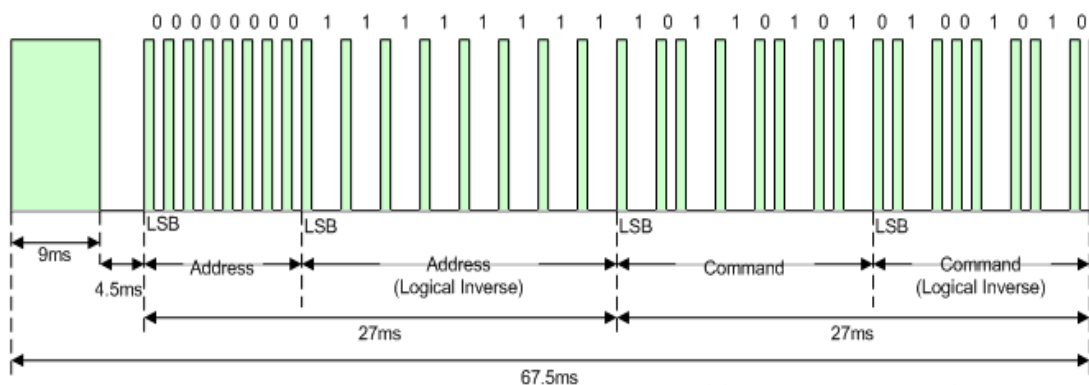


图 1-1

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: ASM_64F0Ax_IR_Send.ASM
* 功能:    FT64F0Ax_IR_Send 功能演示
* IC:      FT64F0A5  TSSOP20
* 内部:    16M/4T
* 说明:    当按键按下后, 发送自己所想要发送的红外码, 按键不松开时发送连发码,
*          此程序所发红外码为 0x55, 0xaa, 0x01, 0xfe
*
*          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
* key-----|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_data
* NC-----|10(PB5)      (PB4)11|-----NC
*
*          -----
*/
=====
#include <FT64F0AX.INC>;
=====
;RAM DEFINE
=====
    TEMP            EQU        0X40
    TEMP1           EQU        0X41
    TEMP2           EQU        0X42
    W_TMP           EQU        0X43
    S_TMP           EQU        0X44
    READPIN         EQU        0X45
    IRDATTEMP       EQU        0X46
    buff            EQU        0X47
    #define         f_key     buff,1
    key_short_count EQU        0X48
    count           EQU        0X49
    countbyte       EQU        0X4A
    f_2ms           EQU        0X4B
=====
;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	0x80	;自动装载周期高 8 位 80H
TIM1ARRL_DEF	EQU	0x83	;自动装载周期低 8 位 83H
EPS0_DEF	EQU	B'00000000'	;外部中断管脚选择为 PA3
EPS1_DEF	EQU	B'00000000'	
ITYPE0_DEF	EQU	B'11000000'	;中断触发类型为双边沿
ITYPE1_DEF	EQU	B'00000000'	
EPIE0_DEF	EQU	B'00001000'	;允许外部中断 3

;=====

```

;USER  DEFINE
;=====
#define      ir_data      PORTB,3
#define      key          PORTA,5
#define      IOCA3        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      INTCON
      BSR          INTCON,GIE      ;使能全局中断
      BANKSEL      TIM1IER
      BSR          TIM1IER,T1UIE   ;允许 TIM1 中断
MAIN:
      CLRWDWDT
      BANKSEL      buff
      BTSS         f_2ms,0
      LJUMP        MAIN
      BANKSEL      buff
      BCR          f_2ms,0
      LCALL        scanky_key
      LJUMP        MAIN
;=====
;INT_PROGRAM
;=====
INT_PROGRAM:
      BANKSEL      TIM1SR1
      BTSC         TIM1SR1,T1UIF
      LJUMP        TIM1Interrupt
      BANKSEL      EPIF0

```

```
BTSC      EPIF0,IOCA3
LJUMP     GPIOInterrupt
LJUMP     INT_RET
```

TIM1Interrupt:

```
BANKSEL   TIM1SR1
BSR       TIM1SR1,T1UIF
```

```
BANKSEL   f_2ms
BSR       f_2ms,0
LJUMP     INT_RET
```

GPIOInterrupt:

```
BANKSEL   PORTA
LDR       PORTA,W
BANKSEL   READPIN
STR       READPIN
BANKSEL   EPIF0
LDWI      0X00
STR       EPIF0
LDWI      0X00
STR       EPIE0
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 2.056ms
;=====
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

=====

;函数名: PA_INT_INITIAL

;功能: PA 口电平变化中断初始化

=====

PA_INT_INITIAL:

BANKSEL TRISA
BSR TRISA,3

BANKSEL PORTA
LDR PORTA,W
BANKSEL READPIN
STR READPIN

BANKSEL EPIE0
BSR EPIE0,IOCA3
BSR EPIF0,IOCA3

BANKSEL EPS0
LDWI EPS0_DEF
STR EPS0
LDWI EPS1_DEF
STR EPS1

BANKSEL ITYPE0
LDWI ITYPE0_DEF

```

    STR        ITYPE0
    LDWI       ITYPE1_DEF
    STR        ITYPE1

    BANKSEL    EPIE0
    LDWI       EPIE0_DEF
    STR        EPIE0

    RET

;=====
;DELAY_8US 16MHz/4T
;=====
DELAY_8US:
    LDWI       0x08
    STR        TEMP

DELAY_8USLOOP:
    CLRWDW
    DECRSZ     TEMP,F
    LJUMP      DELAY_8USLOOP
    RET

;=====
;DELAY_18US 16MHz/4T
;=====
DELAY_18US:
    LDWI       0x11
    STR        TEMP

DELAY_18USLOOP:
    CLRWDW
    DECRSZ     TEMP,F
    LJUMP      DELAY_18USLOOP
    RET

;=====
;函数名: IR_Start
;功能:   红外发送的引导码 9ms 发送 4.5ms 停止
;=====
IR_Start:
    CLRR       count
IR_Start_working:
    LDWI       0XA2                      ;9ms 高电平
    SUBWR      count,0
    BTSC       STATUS,0
    LJUMP      IR_Start_no_work

```

```

    INCR        count,1
    BSR         ir_data
    LCALL       DELAY_8US
    BCR         ir_data
    LCALL       DELAY_18US
    BSR         ir_data
    LCALL       DELAY_8US
    BCR         ir_data
    LCALL       DELAY_18US
    LJUMP       IR_Start_working
IR_Start_no_work:
    CLRR        count
IR_Start_LOOP:
    LDWI        0X50                ;4.5ms 低电平
    SUBWR       count,0
    BTSC        STATUS,0
    RET
    INCR        count,1
    BCR         ir_data
    LCALL       DELAY_8US
    BCR         ir_data
    LCALL       DELAY_18US
    BCR         ir_data
    LCALL       DELAY_8US
    BCR         ir_data
    LCALL       DELAY_18US
    LJUMP       IR_Start_LOOP
;=====
;函数名: IR_Stop
;功能:   红外发送的连发码 9ms 发送 2.25ms 停止
;=====
IR_Stop:
    CLRR        count
IR_Stop_working:
    LDWI        0XA2                ;9ms 高电平
    SUBWR       count,0
    BTSC        STATUS,0
    LJUMP       IR_Stop_no_work
    INCR        count,1
    BSR         ir_data
    LCALL       DELAY_8US
    BCR         ir_data
    LCALL       DELAY_18US
    BSR         ir_data

```

```

    LCALL    DELAY_8US
    BCR      ir_data
    LCALL    DELAY_18US
    LJUMP    IR_Stop_working
IR_Stop_no_work:
    CLRR     count
IR_Stop_LOOP:
    LDWI     0X28                ;2.25ms 低电平
    SUBWR    count,0
    BTSC     STATUS,0
    RET
    INCR     count,1
    BCR      ir_data
    LCALL    DELAY_8US
    BCR      ir_data
    LCALL    DELAY_18US
    BCR      ir_data
    LCALL    DELAY_8US
    BCR      ir_data
    LCALL    DELAY_18US
    LJUMP    IR_Stop_LOOP

```

=====

;函数名: IR_Send_Byte

;功能: 红外发送一个字节

=====

IR_Send_Byte:

```
    CLRR     countbyte
```

IR_Send_Byte_LOOP:

```

    LDWI     0X08
    SUBWR    countbyte,0
    BTSC     STATUS,0
    RET
    INCR     countbyte,1
    BTSS     IRDATTEMP,0
    LJUMP    $+3
    LCALL    IR_Send_data1
    LJUMP    $+2
    LCALL    IR_Send_data0

```

```

    BCR      STATUS,0
    RRR      IRDATTEMP,1
    LJUMP    IR_Send_Byte_LOOP

```

=====

;函数名: IR_Send_data0

;功能： 红外发送数据 0

=====

IR_Send_data0:

CLRR count

IR_Send_data0_working:

LDWI 0X11 ;0.56ms 高电平

SUBWR count,0

BTSC STATUS,0

LJUMP IR_Send_data0_no_work

INCR count,1

BSR ir_data

LCALL DELAY_8US

BCR ir_data

LCALL DELAY_18US

LJUMP IR_Send_data0_working

IR_Send_data0_no_work:

CLRR count

IR_Send_data0_LOOP:

LDWI 0X11 ;0.56ms 低电平

SUBWR count,0

BTSC STATUS,0

RET

INCR count,1

BCR ir_data

LCALL DELAY_8US

BCR ir_data

LCALL DELAY_18US

LJUMP IR_Send_data0_LOOP

=====

;函数名: IR_Send_data1

;功能： 红外发送数据 1

=====

IR_Send_data1:

CLRR count

IR_Send_data1_working:

LDWI 0X11 ;0.56ms 高电平

SUBWR count,0

BTSC STATUS,0

LJUMP IR_Send_data1_no_work

INCR count,1

BSR ir_data

LCALL DELAY_8US

BCR ir_data

LCALL DELAY_18US

```

        LJUMP          IR_Send_data1_working
IR_Send_data1_no_work:
        CLRR          count
IR_Send_data1_LOOP:
        LDWI          0X37                      ;1.69ms 低电平
        SUBWR          count,0
        BTSC          STATUS,0
        RET
        INCR          count,1
        BCR          ir_data
        LCALL          DELAY_8US
        BCR          ir_data
        LCALL          DELAY_18US
        LJUMP          IR_Send_data1_LOOP
;=====
;函数名: scanky_key
;功能:   当按键按下时候发送所需要发送的红外数字, 长按发送连发码
;=====
scanky_key:
        BTSS          key
        LJUMP          scanky_key_down
        LJUMP          scanky_key_up
scanky_key_down:
        BTSC          f_key
        LJUMP          scanky_key_long
        INCR          key_short_count,1
        LDWI          0X0A
        SUBWR          key_short_count,0
        BTSS          STATUS,0
        RET
        CLRR          key_short_count
        BSR          f_key
        LCALL          IR_Start
        LDWI          0X55
        STR          IRDATTEMP
        LCALL          IR_Send_Byte
        LDWI          0XAA
        STR          IRDATTEMP
        LCALL          IR_Send_Byte
        LDWI          0X01
        STR          IRDATTEMP
        LCALL          IR_Send_Byte
        LDWI          0XFE
        STR          IRDATTEMP

```

```
    LCALL    IR_Send_Byte
    LCALL    IR_Send_data0
    RET
scanky_key_long:
    LCALL    IR_Stop
    RET
scanky_key_up:
    CLRR     key_short_count
    BCR      f_key
    RET
```

```
;=====
```

```
;函数名: sleep_mode
```

```
;功能:   无操作进入睡眠
```

```
;=====
```

```
sleep_mode:
    BTSS     key
    RET
    BTSC     f_key
    RET

    LCALL    PA_INT_INITIAL
    SLEEP
    NOP
    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/>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices (SZ) 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 (SZ) Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices (SZ) Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices (SZ) Corporation. The FMD logo is a registered trademark of Fremont Micro Devices (SZ) Corporation. All other names are the property of their respective owners.