

FT62F08X

Application note

FMD Confidential

目录

1. GPIO	4
1.1. 端口和 TRIS 寄存器	5
1.2. 弱上拉	5
1.3. 弱下拉	5
1.4. 开漏输出	5
1.5. ANSELA 寄存器	6
1.6. 源电流选择	6
1.7. 灌电流选择	6
1.8. 管脚输出的优先级	6
1.9. PORTx 功能及优先级	6
1.10. 管脚功能重映射	7
1.11. 外部中断	8
1.12. 与端口相关寄存器汇总	9
1.12.1. PSRC0, 地址 0x11A	9
1.12.2. PSRC1, 地址 0x11B	10
1.12.3. PSINK0, 地址 0x19A	10
1.12.4. PSINK1, 地址 0x19B	10
1.12.5. PSINK2, 地址 0x19C	11
1.12.6. PSINK3, 地址 0x19D	11
1.12.7. IYPE0, 地址 0x11E	11
1.12.8. IYPE1, 地址 0x11F	12
1.12.9. AFPO, 地址 0x19E	12
1.12.10. AFP1, 地址 0x19F	13
1.12.11. AFP2, 地址 0x11D	13
1.12.12. EPS0, 地址 0x118	14
1.12.13. EPS1, 地址 0x119	14
1.12.14. EPIF0, 地址 0x14	15
1.12.15. EPIE0, 地址 0x94	15
1.12.16. ODCON0, 地址 0x21F	15
1.12.17. PORTx, 地址 0x0C, 0D, 0E, 0F	16
1.12.18. TRISx, 地址 0x8C, 8D, 8E, 8F	16
1.12.19. LATx, 地址 0x10C, 10D, 10E, 10F	16

1.12.20.	WPUx, 地址 0x18C, 18D, 18E, 18F.....	17
1.12.21.	WPDx, 地址 0x20C, 20D, 20E, 20F.....	17
1.12.22.	ANSELA, 地址 0x197.....	17
2 应用范例		18

FMD Confidential

FT62F08x IO 应用

1. GPIO

本芯片共包含 30 个 GPIO。这些 IO 除了作为普通输入/输出端口以外还通常具备一些与内核周边电路通讯的功能。

每个端口有 8 个标准寄存器供其操作使用。这些寄存包括：

- TRISx 寄存器（数据方向寄存器）
- PORTx 寄存器（用于读器件引脚上的电平）
- LATx 寄存器（输出锁存器）
- WPUx 寄存器（上拉控制）
- WPDx 寄存器（下拉控制）
- PSRCx 寄存器（源电流选择）
- PSINKx 寄存器（灌电流选择）
- ITYPEx 寄存器（中断类型选择）

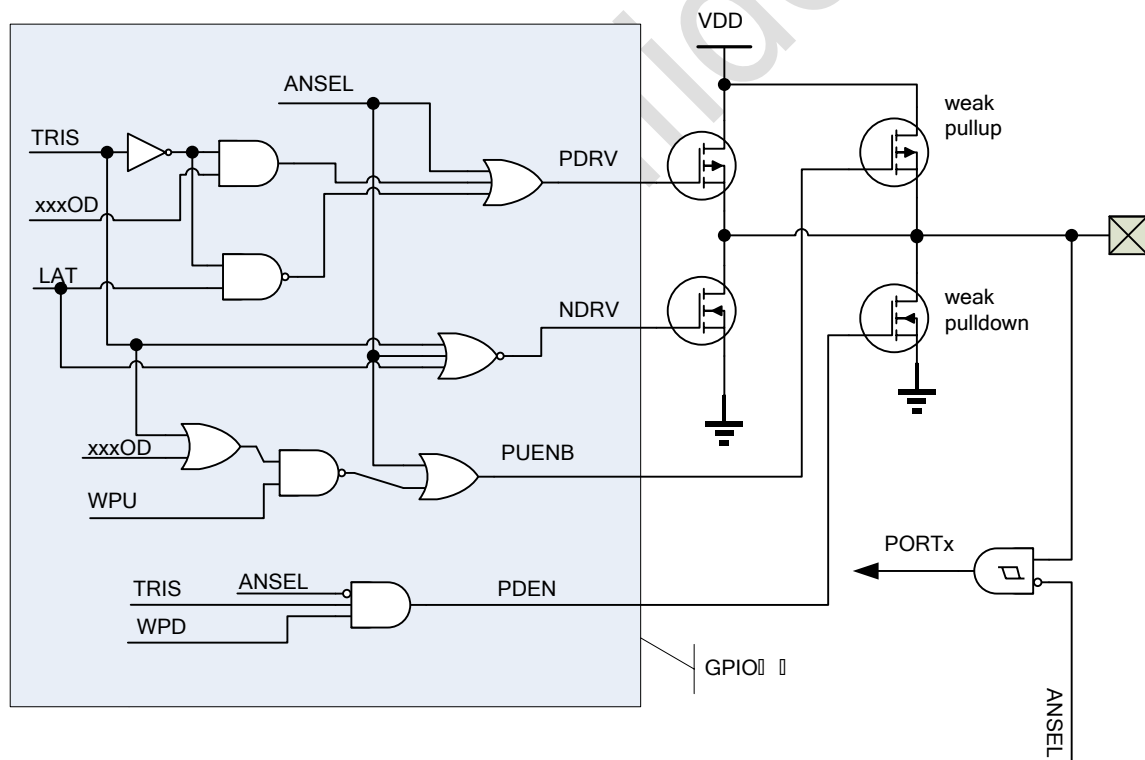


图 17.1 I/O 的结构原理

某些端口可能有以下额外寄存器：

- ANSELx（模拟选择寄存器）

通常情况下，当某个端口上的外设使能时，其相关引脚可能不能用作通用输出引脚，但可读取该引脚。

数据锁存器（LATx 寄存器）用于对 I/O 引脚所驱动的值进行“读-修改-写”操作。对 LATx 寄存器的写操作与对相

应 PORTx 寄存器的写操作具有相同的效果。读 LATx 寄存器将读取保存在 I/O 端口锁存器中的值，而读 PORTx 寄存器将读取实际的 I/O 引脚值。

支持模拟输入的端口具有相关的 ANSELx 寄存器。当 ANSEL 位置 1 时，禁止与该位相关的数字输入缓冲器。禁止输入缓冲器可防止逻辑输入电路产生短路电流。

1.1. 端口和 TRIS 寄存器

所有的管脚 PORTx.y 都是双向端口，其方向控制寄存器就是 TRISx.y。将 TRISx.y 设置为“0”会将该对应 PORTx.y 端口设置为输出端口。在置为输出端口时，输出驱动电路会被打开，输出寄存器里的数据会被放置到输出端口。当 IO 处于输入状态时（TRISA=1），对 PORTx 进行读反映的是输入端口的状态。在 PORTx 上进行写动作时，PORTx 内容会被写入输出寄存器。所有的写操作都是“读-更改-写”这样一个微流程，即数据被读，然后更改，再写入输出寄存器的过程。

当 MCLRE 为 1 时，PORTC[0] 读的值为 0，此时它是作为外部复位管脚。

1.2. 弱上拉

每个端口都有一个可以单独设置的内部弱上拉功能。控制 WPUx 寄存器里的位就可使能或关断这些弱上拉电路。当 GPIO 被设置为输出时，这些弱上拉电路会被自动关断。弱上拉电路在上电复位期间被置为关断状态，因为 WPUx 寄存器复位为无效状态。

当 PORTC[0] 配置为复位脚时，内部上拉是自动打开的，此时 WPUC[0] 不起作用。

1.3. 弱下拉

跟弱上拉功能类似，每个管脚处于数字输入时具有内部弱下拉功能，由寄存器 WPDx 控制。需要注意的是，弱上拉和弱下拉不是互斥的，即它们可以同时 O 打开。

另外，PORTC[0] 作为复位脚时，弱上拉自动打开，但它不反映到 WPUC[0] 上，同时弱下拉自动关闭，WPDC[0] 不起作用。

1.4. 开漏输出

以下 3 种功能管脚支持开漏输出：

SPI_MISO, SPI_MOSI

I2C_SCL, I2C_SDA

UART_TX

开漏输出由寄存器 ODCON0 相关位控制，当相关位为 1 时，该功能所在的管脚即配置为开漏。

注意：

1. 管脚的开漏功能和内部上拉功能可以同时打开；

2. 对于拥有重映射功能的 I2C，开漏设置只应用在对应的被映射的管脚上；

1.5.ANSELA 寄存器

ANSELA 寄存器用于控制 IO 的数模输入，当 ANSELA.x 为 1 时，对应的 IO 口为模拟引脚，IO 的输入上拉、下拉被自动禁止，软件读该 IO 返回的是 0。

ANSELA 寄存器位对数字输出驱动没有影响，换言之，TRIS 位的优先级更高，即当 TRIS 为 0 时，不管相关的 ANSELA.x 是 0 还是 1，对应的 IO 为数字输出 IO。要想配置真正的模拟管脚，TRIS 要置 1，把数字输出驱动关闭。

1.6.源电流选择

每个 I/O 口都支持不同的源电流驱动能力。通过配置相应的选择寄存器 PSRCx，指定的 I/O 端口可支持 4 种级别的源电流驱动能力。仅当对应的引脚被设为输出时，其源电流选择位才有效。否则，这些选择位无效。用户可参考 I/O 电气特性章节为不同应用选择所需的源电流。

1.7.灌电流选择

每个 I/O 都支持 2 种不同的灌电流驱动能力，设置寄存器为 PSINKx，当 I/O 设置为输出管脚时，其灌电流设置位才有效。

1.8.管脚输出的优先级

每个 I/O 管脚均复用了多个功能，当某管脚复用的功能模块都使能输出的情况下，就存在优先级的問題。

因为输入是连到各个功能模块的，故输入不存在优先级问题，例如 PB0 作为 GPIO 输入功能时，同时也作为 TIM2 的捕捉输入。

1.9.PORTx 功能及优先级

管脚名称	功能优先级（由高到低排序）	管脚名称	功能优先级（由高到低排序）
PA0	TIM1_CH1 SPI_MISO PA0	PB0	TIM1_CH3N TIM2_CH1 SPI_SCK PB0
PA1	TIM1_CH2 SPI_MOSI PA1	PB1	CLKO TIM1_CH4 PB1
PA2	ISPCK（处于调试模式） I2C_SCL UART_RX PA2	PB2	I2C_SCL PB2

PA3	TIM1_CH2N PA3	PB3	I2C_SDA PB3
PA4	TIM2_CH2 PA4	PB4	TIM1_CH3 PB4
PA5	TIM2_CH1 UART_CK PA5	PB5	TIM2_CH3 SPI_NSS PB5
PA6	UART_TX PA6	PB6	ISPDAT（处于调试模式） I2C_SDA UART_TX PB6
PA7	PA7	PB7	OSC2（XT 模式） SPI_MOSI PB7

管脚名称	功能优先级（由高到低排序）	管脚名称	功能优先级（由高到低排序）
PC0	MCLR_B（复位脚） TIM1_CH1N PC0	PD0	SPI_NSS PD0
PC1	OSC1（XT 模式） SPI_MISO PC1	PD1	TIM1_CH1 UART_CK PD1
PC2	PC2	PD2	TIM1_CH2 PD2
PC3	PC3	PD3	TIM1_CH3 SPI_SCK PD3
PC4	PC4	PD4	CLKO PD4
PC5	TIM1_CH3N PC5	PD5	TIM1_CH4 PD5
PC6	TIM1_CH2N PC6		
PC7	TIM1_CH1N PC7		

1.10. 管脚功能重映射

为提高应用的灵活性，有部分功能输入/输出支持重映射功能，即可以在两个管脚之间选择输入或输出，由寄存器 AFP0、AFP1 和 AFP2 设置。

1.11. 外部中断

所有 I/O 都可被选择为外部中断源，但同一时刻最多只有 8 个 IO 可以作为外部中断管脚，它们具备以下特性：

- 上升沿中断
- 下降沿中断
- 双边沿中断
- 低电平中断

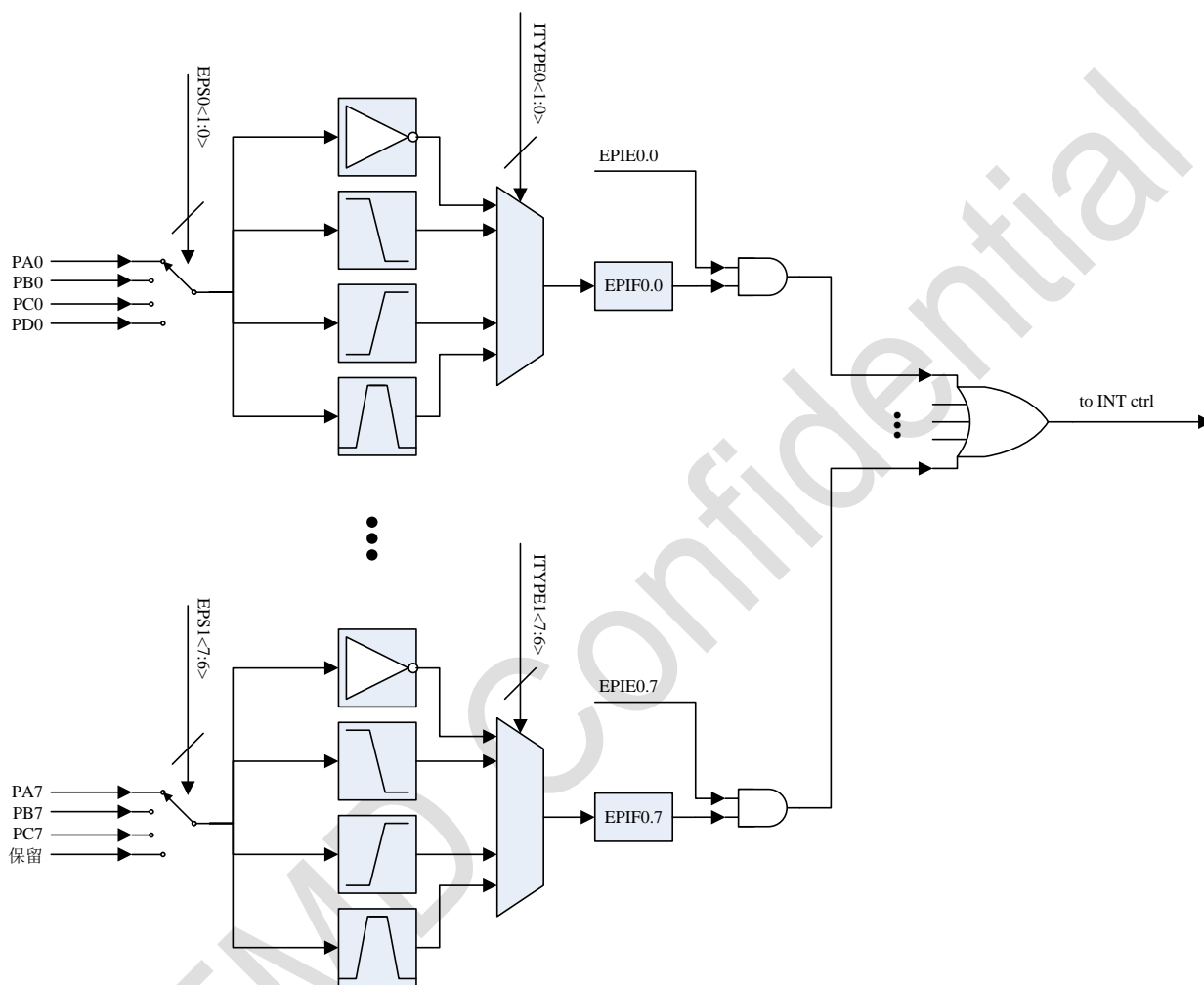


图 17.2 外部中断结构框图

中断类型的选择通过寄存器 ITYPE0、1 设置。

ITYPEx[1:0]/[3:2]值	中断触发类型
00	低电平
01	上升沿
10	下降沿
11	双边沿

外部中断源的选择通过 EPS0，EPS1 设置。

1.12. 与端口相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PSRC0	0x11A	管脚的源电流设置 0								0000 0000
PSRC1	0x11B	管脚的源电流设置 1								0000 0000
PSINK0	0x19A	管脚的灌电流设置 0								0000 0000
PSINK1	0x19B	管脚的灌电流设置 1								0000 0000
PSINK2	0x19C	管脚的灌电流设置 2								0000 0000
PSINK3	0x19D	—	—	管脚的灌电流设置 3						--00 0000
ITYPE0	0x11E	管脚中断类型设置 0								0000 0000
ITYPE1	0x11F	管脚中断类型设置 1								0000 0000
AFP0	0x19E	管脚重映射寄存器 0								0000 0000
AFP1	0x19F	—	管脚重映射寄存器 1							-000 0000
AFP2	0x11D	—	—	—	管脚重映射寄存器 2					---0 0000
EPS0	0x118	外部中断管脚选择 0								0000 0000
EPS1	0x119	外部中断管脚选择 1								0000 0000
EPIF0	0x14	外部管脚中断标志位								0000 0000
EPIE0	0x94	外部管脚中断使能位								0000 0000
ODCON0	0x21F	—	—	—	—	—	SPIOD	I2COD	UROD	---- -000
PORTA	0x0C	端口 A 读管脚寄存器								xxxx xxxx
PORTB	0x0D	端口 B 读管脚寄存器								xxxx xxxx
PORTC	0x0E	端口 C 读管脚寄存器								xxxx xxxx
PORTD	0x0F	—	—	端口 D 读管脚寄存器						--xx xxxx
TRISA	0x8C	端口 A 方向控制								1111 1111
TRISB	0x8D	端口 B 方向控制								1111 1111
TRISC	0x8E	端口 C 方向控制								1111 1111
TRISD	0x8F	—	—	端口 D 方向控制						--11 1111
LATA	0x10C	端口 A 数据锁存器								xxxx xxxx
LATB	0x10D	端口 B 数据锁存器								xxxx xxxx
LATC	0x10E	端口 C 数据锁存器								xxxx xxxx
LATD	0x10F	—	—	端口 D 数据锁存器						--xx xxxx
ANSELA	0x197	模拟管脚设置								0000 0000

1.12.1. PSRC0, 地址 0x11A

Bit	7	6	5	4	3	2	1	0
Name	PSRCB				PSRCA			
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
-----	------	----------

7:4	PSRCB	PSRCB[3:2], 控制 PORTB[7:4]源电流能力 PSRCB[1:0], 控制 PORTB[3:0]源电流能力
3:0	PSRCA	PSRCA[3:2], 控制 PORTA[7:4]源电流能力 PSRCA[1:0], 控制 PORTA[3:0]源电流能力

1.12.2. PSRC1, 地址 0x11B

Bit	7	6	5	4	3	2	1	0
Name	PSRCD				PSRCC			
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:4	PSRCD	PSRCD[3:2], 控制 PORTD[5:4]源电流能力 PSRCD[1:0], 控制 PORTD[3:0]源电流能力
3:0	PSRCC	PSRCC[3:2], 控制 PORTC[7:4]源电流能力 PSRCC[1:0], 控制 PORTC[3:0]源电流能力

PSRCx[1:0]/[3:2]值	源电流能力
00	L0 (最小)
01	L1
10	L2
11	L3 (最大)

1.12.3. PSINK0, 地址 0x19A

Bit	7	6	5	4	3	2	1	0
Name	PSINK0							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PSINK0	PORTA 的灌电流能力设置 0: L0 (最小) 1: L1 (最大)

1.12.4. PSINK1, 地址 0x19B

Bit	7	6	5	4	3	2	1	0
Name	PSINK1							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PSINK1	PORTB 的灌电流能力设置 0: L0 (最小) 1: L1 (最大)

1.12.5. PSINK2, 地址 0x19C

Bit	7	6	5	4	3	2	1	0
Name	PSINK2							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PSINK2	PORTC 的灌电流能力设置 0: L0 (最小) 1: L1 (最大)

1.12.6. PSINK3, 地址 0x19D

Bit	7	6	5	4	3	2	1	0
Name	PSINK3							
Reset	—	—	0	0	0	0	0	0
Type	RO-0	RO-0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	NA	保留位
5:0	PSINK3	PORTD 的灌电流能力设置 0: L0 (最小) 1: L1 (最大)

1.12.7. ITYPE0, 地址 0x11E

Bit	7	6	5	4	3	2	1	0
Name	ITYPE0							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:4	ITYPE0[7:4]	ITYPE0[7:6], 控制 PORTx.3 中断类型 ITYPE0[5:4], 控制 PORTx.2 中断类型
3:0	ITYPE0[3:0]	ITYPE0[3:2], 控制 PORTx.1 中断类型

		ITYPE0[1:0], 控制 PORTx.0 中断类型
--	--	------------------------------

1.12.8. ITYPE1, 地址 0x11F

Bit	7	6	5	4	3	2	1	0
Name	ITYPE1							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:4	ITYPE1[7:4]	ITYPE1[7:6], 控制 PORTx.7 中断类型 (仅 PORTA/B/C) ITYPE1[5:4], 控制 PORTx.6 中断类型 (仅 PORTA/B/C)
3:0	ITYPE1[3:0]	ITYPE1[3:2], 控制 PORTx.5 中断类型 ITYPE1[1:0], 控制 PORTx.4 中断类型

1.12.9. AFP0, 地址 0x19E

Bit	7	6	5	4	3	2	1	0
Name	AFP0							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function			
7:0	AFP0	位	值	受控功能	复用管脚
		bit0	0	I2C_SDA	PB3
			1		PB6
		bit1	0	ADC_ETR	PA
			1		PB2
		bit2	0	TIM1_CH3N	PB0
			1		PC5
		bit3	0	TIM1_CH2N	PA3
			1		PC6
		bit4	0	TIM1_CH1N	PC0
			1		PC7
		bit5	0	SPI_NSS	PB5
			1		PD0
		bit6	0	TIM1_CH1	PA0
			1		PD1
		bit7	0	UART_CK	PA5
			1		PD1

1.12.10. AFP1, 地址 0x19F

Bit	7	6	5	4	3	2	1	0
Name	AFP1							
Reset	—	0	0	0	0	0	0	0
Type	RO-0	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function			
7	NA	保留位			
6:0	AFP1	位	值	受控功能	复用管脚
		bit0	0	TIM1_CH2	PA1
			1		PD2
		bit1	0	TIM1_CH3	PB4
			1		PD3
		bit2	0	TIM2_CH1	PA5
			1		PB0
		bit3	0	TIM1_BKIN	PB3
			1		PD4
		bit4	0	I2C_SCL	PB2
			1		PA2
		bit5	0	TIM1_CH4	PB1
			1		PD5
		bit6	0	CLKO	PB1
			1		PD4

1.12.11. AFP2, 地址 0x11D

Bit	7	6	5	4	3	2	1	0
Name	AFP2							
Reset	—	—	—	0	0	0	0	0
Type	RO-0	RO-0	RO-0	RW	RW	RW	RW	RW

Bit	Name	Function			
7:5	NA	保留位, 读 0			
4:0	AFP2	位	值	受控功能	复用管脚
		bit0	0	UART_TX	PA6
			1		PB6
		bit1	0	UART_RX	PA7
			1		PA2
		bit2	0	SPI_MISO	PA1
			1		PC1

		bit3	0	SPL_MOSI	PA0	
			1		PB7	
		bit4	0	SPL_SCK	PB0	
			1		PD3	

1.12.12. EPS0, 地址 0x118

Bit	7	6	5	4	3	2	1	0
Name	EPS0							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function			
7:0	EPS0	外部中断 EINT3~0 的管脚选择			
		EPS0[1:0]值	EINT0 管脚	EPS0[3:2]值	EINT1 管脚
		00	PA0	00	PA1
		01	PB0	01	PB1
		10	PC0	10	PC1
		11	PD0	11	PD1
		EPS0[5:4]值	EINT2 管脚	EPS0[7:6]值	EINT3 管脚
		00	PA2	00	PA3
		01	PB2	01	PB3
		10	PC2	10	PC3
		11	PD2	11	PD3

1.12.13. EPS1, 地址 0x119

Bit	7	6	5	4	3	2	1	0
Name	EPS1							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function			
7:0	EPS1	外部中断 EINT7~4 的管脚选择			
		EPS1[1:0]值	EINT4 管脚	EPS1[3:2]值	EINT5 管脚
		00	PA4	00	PA5
		01	PB4	01	PB5
		10	PC4	10	PC5
		11	PD4	11	PD5
		EPS1[5:4]值	EINT6 管脚	EPS1[7:6]值	EINT7 管脚
		00	PA6	00	PA7

		01	PB6	01	PB7
		10	PC6	10	PC7
		11	保留（接 PC6）	11	保留（接 PC7）

1.12.14. EPIF0，地址 0x14

Bit	7	6	5	4	3	2	1	0
Name	EPIF0							
Reset	0	0	0	0	0	0	0	0
Type	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	Name	Function
Bit[7:0]	EPIF0	<p>外部中断 x 标志位</p> <p>0: 外部管脚 x 没触发中断，或已由软件清 0</p> <p>1: 外部管脚 x 触发了中断</p> <p>写操作：</p> <p>写 1 清 0，写 1 无效。 建议只使用 MOVWF, MOVWI 指令，而不是 BSF 位操作</p>

1.12.15. EPIE0，地址 0x94

Bit	7	6	5	4	3	2	1	0
Name	EPIE0							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
Bit[7:0]	EPIE0	<p>外部中断 x 允许位</p> <p>0: 禁止外部中断 x</p> <p>1: 允许外部中断 x，当相关标志位 EPIF0.x 为 1 且 GIE 为 1 时，CPU 将执行中断程序</p>

1.12.16. ODCON0，地址 0x21F

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	SPIOD	I2COD	UROD
Reset	—	—	—	—	—	0	0	0
Type	RO-0	RO-0	RO-0	RO-0	RO-0	RW	RW	RW

Bit	Name	Function
7:3	N/A	保留位，读 0

2	SPIOD	SPL_MISO, SPI_MOSI 管脚的开漏输出设置，高有效
1	I2COD	I2C_SCL, I2C_SDA 管脚的开漏输出设置，高有效
0	UROD	UART_TX 管脚的开漏输出设置，高有效

1.12.17. PORTx, 地址 0x0C, 0D, 0E, 0F

Bit	7	6	5	4	3	2	1	0
Name	PORTA/B/C/D							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PORTx	PORTx 管脚寄存器 读返回的是管脚上的电平，写是写到相应的 LATx 寄存器

1.12.18. TRISx, 地址 0x8C, 8D, 8E, 8F

Bit	7	6	5	4	3	2	1	0
Name	TRISA/B/C/D							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TRISx	PORTx 方向控制寄存器 1 = 输入 0 = 输出

1.12.19. LATx, 地址 0x10C, 10D, 10E, 10F

Bit	7	6	5	4	3	2	1	0
Name	LATA/B/C/D							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	LATx	PORTx 数据寄存器

1.12.20. WPUx, 地址 0x18C, 18D, 18E, 18F

Bit	7	6	5	4	3	2	1	0
Name	WPUA/B/C/D							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	WPUx	PORTx 弱上拉控制寄存器 1 = 使能弱上拉 0 = 关闭弱上拉

1.12.21. WPDx, 地址 0x20C, 20D, 20E, 20F

Bit	7	6	5	4	3	2	1	0
Name	WPDA/B/C/D							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	WPDx	PORTx 弱下拉控制寄存器 1 = 使能弱下拉 0 = 关闭弱下拉

1.12.22. ANSELA, 地址 0x197

Bit	7	6	5	4	3	2	1	0
Name	模拟管脚设置寄存器							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ANSELA	模拟选择寄存器 A, 控制 Anx 相关的管脚 1 = Anx 为模拟管脚 0 = Anx 为数字管脚

2 应用范例

```

/*****

```

```

/* 文件名: TEST_62F08x_IO.c
 * 功能: FT62F08x-IO 功能演示
 * IC: FT62F088 LQFP32
 * 内部: 16M
 * empno: 500
 * 说明: 当 DemoPortIn 悬空或者高电平时,
 * DemoPortOut 输出 50Hz 占空比 50%的波形
 * 当 DemoPortIn 接地时, DemoPortOut 输出高电平
 *
 *
 * 参考原理图 TEST_62F08x_sch.pdf
 */

```

```

/*****

```

```

#include "SYSCFG.h"

```

```

/*****

```

```

/*****宏定义*****

```

```

#define unchar unsigned char
#define uint unsigned int
#define ulong unsigned long

```

```

#define DemoPortOut RB3

```

```

#define DemoPortIn RC3

```

```

//volatile unchar mydata; //全局查看变量定义

```

```

/*-----

```

```

 * 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) //PIC_HI-TECH 使用

```

```

{

```

```

}

```

```

/*-----

```

```

 * 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;    //WDT 32KHZ IRCF=111=16MHZ
                             //Bit0=1,系统时钟为内部振荡器
                             //Bit0=0,时钟源由 FOSC<2: 0>决定即编译选项时选择

    INTCON = 0;              //暂禁止所有中断


    PORTA = 0B00000000;
    TRISA = 0B11111111;     //PA 输入输出 0-输出 1-输入
    PORTB = 0B00000000;
    TRISB = 0B11110111;     //PB 输入输出 0-输出 1-输入
    PORTC = 0B00000000;
    TRISC = 0B11111111;     //PC 输入输出 0-输出 1-输入
    PORTD = 0B00000000;
    TRISD = 0B11111111;     //PD 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
    WPUB = 0B00000000;      //PB 端口上拉控制 1-开上拉 0-关上拉
    WPUC = 0B00001000;      //PC 端口上拉控制 1-开上拉 0-关上拉
    WPUD = 0B00000000;      //PD 端口上拉控制 1-开上拉 0-关上拉


    WPDA = 0B00000000;      //PA 端口上拉控制 1-开下拉 0-关下拉
    WPDB = 0B00000000;      //PB 端口上拉控制 1-开下拉 0-关下拉
    WPDC = 0B00000000;      //PC 端口上拉控制 1-开下拉 0-关下拉
    WPDD = 0B00000000;      //PD 端口上拉控制 1-开下拉 0-关下拉


    PSRC0 = 0B11111111;     //PORTA,PORTB 源电流设置最大
    //BIT7~BIT6:PORTB[7:4]源电流能力控制,BIT5~BIT4:PORTB[3:0]源电流能力控制
    //BIT3~BIT2:PORTA[7:4]源电流能力控制,BIT1~BIT0:PORTA[3:0]源电流能力控制


    PSRC1 = 0B11111111;     //PORTC,PORTD 源电流设置最大
    //BIT7~BIT6:PORTD[7:4]源电流能力控制,BIT5~BIT4:PORTD[3:0]源电流能力控制
    //BIT3~BIT2:PORTC[7:4]源电流能力控制,BIT1~BIT0:PORTC[3:0]源电流能力控制


    PSINK0 = 0B11111111;     //PORTA 灌电流设置最大 0:最小, 1:最大
    PSINK1 = 0B11111111;     //PORTB 灌电流设置最大 0:最小, 1:最大
    PSINK2 = 0B11111111;     //PORTC 灌电流设置最大 0:最小, 1:最大
    PSINK3 = 0B11111111;     //PORTD 灌电流设置最大 0:最小, 1:最大


    ANSELA = 0B00000000;     //全为数字管脚

}

/*-----
* 函数名称: DelayUs

```

- * 功能：短延时函数 --16M-2T--大概快 1%左右.
- * 输入参数：Time 延时时间长度 延时时长 Time Us
- * 返回参数：无

-----*/

void DelayUs(unsigned char Time)

```
{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}
```

/*-----*/

- * 函数名称：DelayMs
- * 功能：短延时函数
- * 输入参数：Time 延时时间长度 延时时长 Time ms
- * 返回参数：无

-----*/

void DelayMs(unsigned char Time)

```
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197); //快 1%
        }
    }
}
```

/*-----*/

- * 函数名称：DelayS
- * 功能：短延时函数
- * 输入参数：Time 延时时间长度 延时时长 Time S
- * 返回参数：无

-----*/

void DelayS(unsigned char Time)

```
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<10;b++)
        {
            DelayMs(100);
        }
    }
}
```

```
}
/*-----
 * 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
    while(1)
    {
        DemoPortOut = 1;
        DelayMs(10);      //10ms
        if(DemoPortIn == 1) //判断输入是否为高电平
        {
            DemoPortOut = 0;
        }
        DelayMs(10);
    }
}
```

Fremont Micro Devices (SZ) Limited

#5-8, 10/F, Changhong Building, Ke-Ji Nan 12 Road, Nanshan District, Shenzhen, Guangdong 518057

Tel: (86 755) 86117811

Fax: (86 755) 86117810

Fremont Micro Devices (Hong Kong) Limited

#16, 16/F, Blk B, Veristrong Industrial Centre, 34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong

Tel: (852) 27811186

Fax: (852) 27811144

Fremont Micro Devices (USA), Inc.

42982 Osgood Road Fremont, CA 94539

Tel: (1-510) 668-1321

Fax: (1-510) 226-9918

Web Site: <http://www.fremontmicro.com/>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices, Incorporated (BVI) assumes no responsibility for the consequences of use of such information or for any infringement of patents or 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, Incorporated (BVI). Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices, Incorporated (BVI) products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices, Incorporated (BVI). The FMD logo is a registered trademark of Fremont Micro Devices, Incorporated (BVI). All other names are the property of their respective own.