

# **FT61F02X**

## **IO Application note**

## 目录

1. I/O 端口 .....	3
1.1. I/O 端口相关寄存器汇总 .....	6
1.2. I/O 配置 .....	7
2. 应用范例 .....	9
联系信息 .....	12

## FT61F02x IO 应用

### 1. I/O 端口

根据封装类型的不同,FT61F02x 系列芯片最多有 14 个 I/O 引脚可用,共分为 2 组:PORTA (8)和 PORTC (6)。[表 1-1](#) 列出了所有 I/O 引脚的功能。

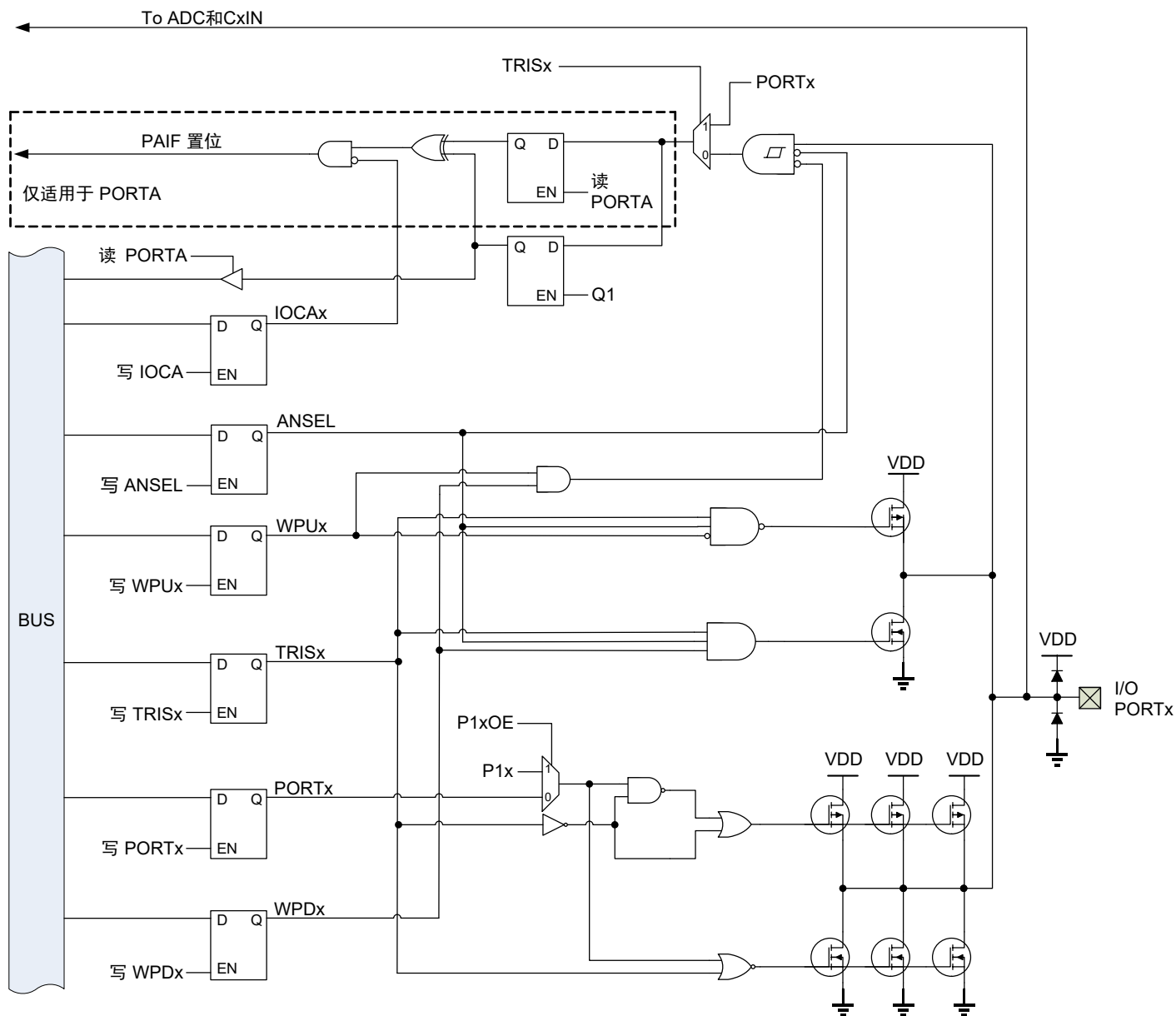


图 1-1 PORT 端口结构框图

I/O 引脚均具有以下功能 ([表 1-3](#), [表 1-4](#)):

- 数字输出
- 弱上拉
- 数字输入
- 弱下拉 (PA4、PC1、PC2、PC3)

此外,部分 I/O 具有以下特殊功能:

1. 烧录调试引脚 (ISP-Data, ISP-CLK), 硬件内部连接, 不需设置。
2. 通过 IDE 界面配置, 且在芯片初始化配置时加载的功能 (表 1-2):
  - 外部时钟/晶振输入 (OSC1, OSC2)
  - 系统外部复位 (/MCLR<sub>B</sub>)
  - 内部时钟输出
3. 通过指令对相应 I/O 引脚进行配置的其他功能, 可分为 3 类:
  - a. 数字输出
    - PWM
    - 增强型 PWM
    - ECCP 比较输出
  - b. 数字输入
    - Timer0 时钟输入
    - Timer1 时钟输入
    - Timer1 门控输入
    - ECCP 捕捉输入
    - 外部边沿中断
    - GPIO 端口变化中断
  - c. 模拟输入
    - LVD / BOR
    - ADC
    - $V_{REF}$
    - 比较器
  - d. 模拟输出
    - $V_{REG}$  稳压器输出

引脚名	ISP 调试	时钟	ADC	稳压器	比较器	中断	ECCP / PWM	数字 I/O 上拉/下拉	源电流 (mA)	灌电流 (mA)
PA0	CLK		AN0		C1IN+	√		√ / --	24	35
PA1	DATA		AN1		C1IN-	√		√ / --	24	35
PA2		T0CKI	AN2		C1OUT	√		√ / --	24	35
PA3			AN3			√		√ / --	24	35
PA4				V <sub>REGP</sub>		√		√ / √	24	35
PA5						√ + /MCLRB		√ / --	24	35
PA6		输出 /OSC-				√		√ / --	24	35
PA7		T1CKI /OSC+				√		√ / --	24	35
PC0			AN4 (V <sub>REF</sub> )		C2IN+		P1F	√ / --	24	35
PC1			AN5		C2IN-	INT	P1E	√ / √	24	35
PC2			AN6				PWM5 / P1D	√ / √	24	35
PC3							PWM4 / P1C	√ / √	24	35
PC4					C2OUT		PWM3 / P1B	√ / --	24	35
PC5				V <sub>REGN</sub>			CCP1 / P1A	√ / --	24	35
注		T1G=PA6							V <sub>DD</sub> =5, V <sub>DS</sub> =0.5	

表 1-1 I/O 端口功能

### 1.1. I/O 端口相关寄存器汇总

名称	功能	默认
RDCTRL	当 TRISx = 0 (输出使能) 时, 读 PORTx 寄存器的返回值 • 输入锁存器 • 输出锁存器	输出锁存器
MCLRE	外部 I/O 复位	关闭
FOSC	• LP: PA7 (+) 和 PA6 (-) 接外部低速晶振 • XT: PA7 (+) 和 PA6 (-) 接外部高速晶振 • EC: PA7 (+) 接外部时钟输入, PA6 为 I/O • INTOSC: PA6 输出“指令时钟”, PA7 为 I/O • INTOSCIO: PA7 和 PA6 为 I/O	INTOSCIO

表 1-2 I/O 相关初始化配置寄存器

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
ANSEL	0x91	ANSEL [7:0]								1111 1111
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111
TRISC	0x87	-	-	PORTC 方向控制						--11 1111
PORTA	0x05	PORTA 输出寄存器								xxxx xxxx
PORTC	0x07	-	-	PORTC 输出寄存器						--xx xxxx
WPUA	0x95	PORTA 弱上拉								1111 1111
WPUC	0x88	-	-	PORTC 弱上拉						--00 0000
WPD	0x89	-	-	-	WPDA4	WPDC1	WPDC2	WPDC3	-	---0 000-
IOCA	0x96	IOCA[7:0]: PORTA 端口变化中断设置								0000 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111

表 1-3 I/O 相关用户寄存器的地址和复位值

名称	状态		寄存器	地址	复位值
TRISA	PORTA	PORT 端口数字输出 (方向控制)	TRISA[7:0]	0x85	RW-1111 1111
TRISC	PORTC	1 = 关闭 0 = 使能 (关闭上拉/下拉)	TRISC[5:0]	0x87	RW-11 1111
ANSEL	1 = 关闭上拉/下拉, 及数字输入 (仅适用于8个ADC通道) 0 = (无动作)		ANSEL[7:0]	0x91	RW-1111 1111
/PAPU	1 = 关闭所有 PORTA 上拉功能 0 = 上拉由 WPUA 控制		OPTION[7]	0x81	RW-1
WPUA	PORTA	弱上拉	WPUA[7:0]	0x95	RW-1111 1111
WPUC	PORTC	1 = 使能 (PORTA 默认值) 0 = 关闭 (PORTC 默认值)	WPUC[5:0]	0x88	RW-00 0000
WPDA4	PORTA	弱下拉	WPD[4]	0x89	RW-0
WPDC	PORTC	1 = 使能      0 = 关闭	WPD[3:1]		RW-000
PORTA	PORTA	数据输出寄存器	PORTA[7:0]	0x05	RW-xxxx xxxx
PORTC	PORTC		PORTC[5:0]	0x07	RW-xx xxxx

表 1-4 I/O 相关用户寄存器

## 1.2. I/O 配置

每个 PORT 端口，均需根据其相应功能配置以下 4 个模块(表 1-5):

- 弱上拉
- 弱下拉 (PA4、PC1、PC2、PC3)
- 数字输入
- 数字输出

功能	数字输入	上拉/下拉	数字输出	设置
ISP-DATA	On	Off	On	(硬件内置, 忽略指令)
ISP-CLK	On	Off	Off	(硬件内置, 忽略指令)
/MCLRB	On	上拉	Off	(初始化配置, 忽略指令)
时钟输出	(忽略)	Off	On	(初始化配置, 忽略指令)
OSC+ (EC)	On	(可选)	Off	(初始化配置, 忽略指令)
OSC+ / OSC- (LP, XT)	Off	Off	Off	(初始化配置, 忽略指令)
ADC	Off	Off	Off	TRISx = 1; ANSELx = 1
V <sub>REF</sub>	Off	Off	Off	TRISx = 1
比较器输入	Off	Off	On	TRISx = 1; ANSELx = 1
比较器输出	On	Off	On	TRISx = 0
Timer0 时钟	On	(可选)	Off	TRISx = 1
Timer1 时钟	On	(可选)	Off	TRISx = 1
Timer1 门控	On	(可选)	Off	TRISx = 1
Timer3/4/5 时钟	On	(可选)	Off	TRISx = 1
端口变化中断	On	(可选)	Off	TRISx = 1
PC1-INT	On	(可选)	Off	TRISx = 1
数字输入	On	(可选)	Off	TRISx = 1
PWM	On	Off	On	TRISx = 0
ECCP	On	(可选)	Off	TRISx = 1
稳压器	(忽略)	Off	Off	VREG_OE = 1
数字输出	On	Off	On	TRISx = 0

**表 1-5** I/O 配置标志和用户寄存器

注:

1. TRISx = 0: “数字输出” 使能, “上拉/下拉” 自动关闭 (忽略 WPD, WPUx), TRIS 位的优先级高于 ANSELx。
2. TRISx = 1: “数字输出” 关闭。
3. ANSELx = 1: “上拉”、“电平变化中断”、“数字输入” 自动关闭 (忽略 WPD, WPUx)。
4. 可关闭 “数字输入” 的唯一指令为 “ANSELx = 1”。
5. “/PAPU = 1” 关闭所有 PAX 端口的 “弱上拉” 功能。PCx 没有此类控制位。
6. /MCLR 使能: PA5 的弱上拉功能自动使能 (忽略 WPUA[5]); 读 PORTA[5] 的值为 “0”。

7. 对 PORTx 数据输出寄存器进行写操作，I/O 端口将输出相应的逻辑电平。每组多达 8 个 I/O 的数据寄存器共用相同的地址，写操作实际执行‘读-修改-写’的过程，即先读取该组 PORTx 端口锁存器值 (输出或输入)，然后修改，再写回 PORTx 数据寄存器。
8. 数字输出和数字输入功能可以共存，有些应用需要同时使能数字输出和数字输入。
9. 当 TRISx = 0 时，通过 IDE 界面可选择读取 PORTx 输出或输入锁存器的值。
10. 完全复位或系统复位时，PORTx 寄存器不会复位，但 TRISx 将被重置为“1”，从而关闭输出。
11. 当 IO 同时打开弱上拉和弱下拉时，弱下拉将被禁止，弱上拉起作用。

PC1-INT 和 PORTA 端口变化中断的设置，请参阅 [章节 错误!未找到引用源。](#) “中断”。



## 2. 应用范例

```
//*****
/* 文件名: TEST_61F02x_IO.C
* 功能:    FT61F02x-IO 功能演示
* IC:      FT61F023 SOP16
* 晶振:    16M/2T
* 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
*
*          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|-----NC
* DemoPortIn--|5(PC3)  (PA3)12|--DemoPortOut
* NC-----|6(PC2)  (PC0)11|-----NC
* NC-----|7(PA4)  (PC1)10|-----NC
* NC-----|8(PC5)  (PC4)09|-----NC
*
*/
//*****
#include "SYSCFG.h"
//*****宏定义*****
#define DemoPortOut PA3
#define DemoPortIn PC3
/*-----
* 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;    //IRCF=111=16MHz/2T=8MHz,0.125µs
    INTCON = 0;             //暂禁止所有中断
    PORTA = 0B00000000;
    TRISA = 0B00000000;     //PA 输入输出 0-输出 1-输入
    PORTC = 0B00000000;
    TRISC = 0B00001000;     //PC 输入输出 0-输出 1-输入
                           //PC3-输出
    WPUA = 0B00000000;     //PA 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
}
```

```

    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: DelayUs
* 功能:   短延时函数 --16M-2T--大概快 1%左右.
* 输入:   Time 延时时间长度 延时时长 Time μs
* 输出:   无
-----*/
void DelayUs(unsigned char Time)
{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}
/*-----
* 函数名: DelayMs
* 功能:   短延时函数 快 1%
* 输入:   Time 延时时间长度 延时时长 Time ms
* 输出:   无
-----*/
void DelayMs(unsigned char Time)
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197);
        }
    }
}
/*-----
* 函数名: main
* 功能:   主函数
* 输入:   无
* 输出:   无
-----

```

```
-----*/  
void main()  
{  
    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);  
    }  
}
```

## 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.