

FT62F21X

Application note

目录

1	PWM1 特性	3
1.1.	周期	3
1.2.	占空比	3
1.3.	时钟源选择	3
1.4.	睡眠下 PWM 状态	4
1.5.	P1A 的死区时间	4
1.6.	故障刹车	4
1.6.1.	刹车状态	5
1.6.2.	故障清除	5
1.6.3.	自动重启	5
1.7.	关于周期和占空比寄存器的更新	5
1.8.	与 PWM1 相关寄存器汇总	6
1.8.1.	P1ADTL 寄存器, 地址 0x0E	7
1.8.2.	P1BDTL 寄存器, 地址 0x0F	7
1.8.3.	P1CDTL 寄存器, 地址 0x10	8
1.8.4.	TMR2L 寄存器, 地址 0x11	8
1.8.5.	TMR2H 寄存器, 地址 0x13	8
1.8.6.	T2CON0 寄存器, 地址 0x12	8
1.8.7.	P1ADTH 寄存器, 地址 0x14	9
1.8.8.	P1BDTH 寄存器, 地址 0x15	10
1.8.9.	P1CDTH 寄存器, 地址 0x1A	10
1.8.10.	P1DDTL 寄存器, 地址 0x8	10
1.8.11.	P1DDTH 寄存器, 地址 0x9	10
1.8.12.	P1CON 寄存器, 地址 0x16	11
1.8.13.	P1BR0 寄存器, 地址 0x17	11
1.8.14.	P1BR1 寄存器, 地址 0x19	12
1.8.15.	P1OE 寄存器, 地址 0x90	12
1.8.16.	PR2L 寄存器, 地址 0x91	13
1.8.17.	PR2H 寄存器, 地址 0x92	13
1.8.18.	P1POL 寄存器, 地址 0x99	14
1.8.19.	P1AUX 寄存器, 地址 0x1E	14
2	应用范例	15

FT62F21x PWM1 应用

1 PWM1 特性

- 16bit 的分辨率
- 周期和占空比匹配双缓冲设计
- 1 路带死区控制的 PWM 输出：P1A
- 4 路独立占空比的 PWM 输出：P1A, P1B, P1C, P1D
- 每路 PWM 输出极性可独立设置
- 故障刹车以及自动重启

1.1. 周期

PWM 周期由 Timer2 的 PR2 寄存器指定。用公式 9.1 可计算 PWM 周期。

$$PWM \text{ 周期} = (PR2 + 1) * T_{T2CK} * (TMR2 \text{ 预分频值}) \quad \text{公式 9.1}$$

当 TMR2 等于 PR2 时，下一次递增周期将发生以下三个事件：

- TMR2 被清零
- P1A0, P1A2, P1A4, P1B, P1C 置 1（3 路 PWM 都是高有效的情况下）
- 内部的周期寄存器 PR2ACT 和占空比寄存器 P1xDTACT 被更新

1.2. 占空比

通过对以下几个寄存器写入 16 位值可指定 PWM 占空比：

P1xDTL(x= A, B, C)

P1xDTH(x= A, B, C)

其中，P1xDTH 保存的是 3 路 PWM 占空比寄存器的 8 位 MSb，P1xDTL 则是低 8 位。由于内部的双缓冲设计，占空比寄存器在任何时候被写入，它在软件新占空比时 PWM 的不会产生毛刺起重要作用。

公式 9.2 用于计算 PWM 脉宽。

公式 9.3 用于计算 PWM 占空比。

$$\text{脉冲宽度} = P1xDL * T_{T2CK} * (TMR2 \text{ 预分频值}) \quad \text{公式 9.2}$$

$$\text{占空比} = P1xDL / (PR2 + 1) \quad \text{公式 9.3}$$

1.3. 时钟源选择

PWM1 使用的时基定时器为 Timer2，Timer2 的时钟源有以下选择：

- 系统时钟
- 指令时钟（即系统时钟的 2 分频或 4 分频）
- HIRC 的 2 倍频
- 外部时钟 2 倍频（只有当 FOSC 配置为 EC 模式时才有效）
- HIRC
- LIRC

1.4. 睡眠下 PWM 状态

当 T2CKRUN=1 而且 Timer2 的时钟源不是选择指令时钟时，MCU 进入睡眠后，PWM 可以保持在运行状态，由 T2CKSRC 所选择的时钟源不会关闭。否则，Timer2 将停止计数，而 PWM 各管脚电平保留在执行完 SLEEP 指令后的状态。

1.5. P1A 的死区时间

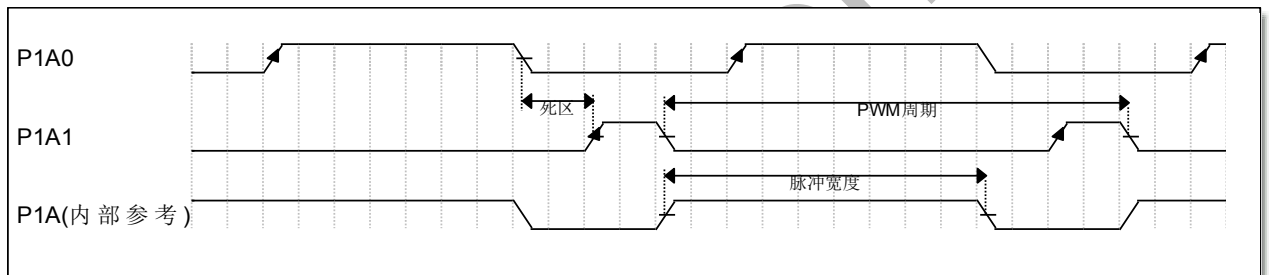


图 9.5.1 死区 PWM 示意图

有 1 路 PWM 带互补输出，P1Ax。其中 P1A0 定义为正输出，[P1A0]为互补输出。P1A 的 PWM 带有死区插入功能，其死区时间由 P1DC[6:0]控制。死区定时器以 Timer2 时钟作为计数时钟源。

注意：{P1A0,[P1A0N]}这对互补输出共享同一死区设置。

1.6. 故障刹车

PWM1 模块支持故障刹车模式，它会在发生外部刹车事件时禁止 PWM 输出，同时 Timer2 及预分频器处于复位状态。刹车模式会将 PWM 输出引脚置于预定状态，该模块用于防止发生故障条件时 PWM 损坏应用。

使用 P1BR0 寄存器的 P1BKS 位可选择故障源，故障事件可以是以下几种：

- BK0 管脚为低电平
- BK0 管脚为高电平
- LVDW 被 LVD 模块置 1
- LVDW=1 或 BK0=0
- LVDW=1 或 BK0=1

刹车状态由 P1BR0 寄存器的 P1BEVT 位指示。如果该位为 0，PWM1 引脚正常工作。如果该

位为 1，PWM1 输出处于关闭状态。

1.6.1. 刹车状态

发生故障时，故障下的 PWM1 管脚电平状态由寄存器 P1xSS 位选择，有以下几种：

- PWM1 置于有效电平
- PWM1 置于无效电平
- PWM1 关闭，处于悬空高阻状态
- TMR2 定时器处于复位状态
- TMR2ON 位不受影响

有效电平由 P1POL 各寄存器位决定。

1.6.2. 故障清除

故障刹车条件是基于电平的信号，而非基于边沿的信号。只要故障条件有效，就故障状态一直保持，软件不能清除；只有当相关故障输入或 LVD 事件消除了，P1BEVT 才可能被清 0。

1.6.3. 自动重启

PWM1 可配置为在故障条件被清除时自动重启 PWM1 信号。通过将 P1CON 寄存器中的 P1AUE 位置 1 使能自动重启。

自动重启使能时，只要故障条件有效，P1BEVT 位就保持置 1。当故障条件被清除时，P1BEVT 位将被硬件清零，TMR2 恢复计数，在下次计数溢出时，PWM1 恢复正常输出。

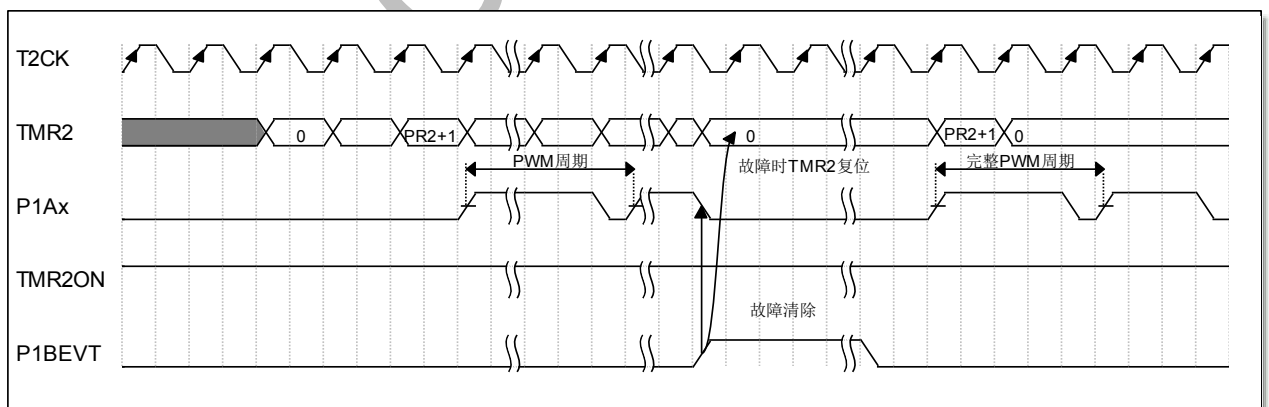


图 9.6.1 PWM 的自动重启时序图

1.7. 关于周期和占空比寄存器的更新

在 Timer2 已经开启的情况下，周期和各占空比寄存器的更新需要 TMR2 和 PR2 的匹配事件，如果用户不想等待，可以通过写 PR2U 位来立即更新。

当 TMR2ON 为 0 时，软件对 PR2, {P1xDTH, P1xDTL} 寄存器的写会马上更新到对应的工作寄存器，此时 PWM 输出保持旧值，不会因为 PR2 或 P1xDT 寄存器变化而变化。

注意：工作寄存器 xxxACT 对软件不可见，软件只能读 PR2 和 P1xDTL, P1xDTH。

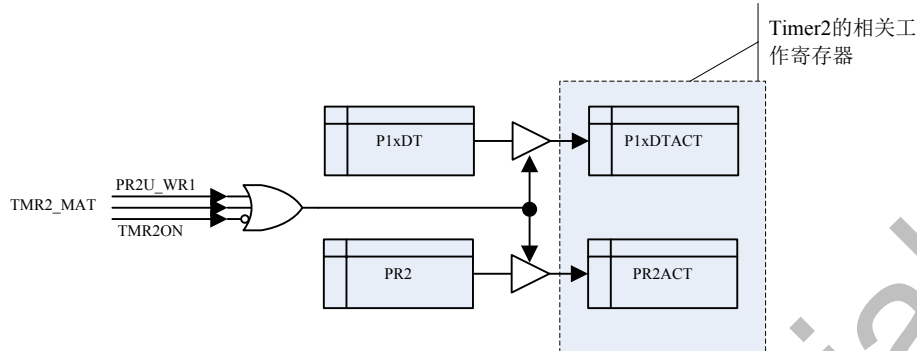


图 9.7.1 T2 工作寄存器的更新

虽然周期和占空比的双缓冲在很大程度保证 PWM 输出不会产生毛刺，但如果软件非常靠近 TMR2 匹配时刻去写这此寄存器，特别是在 T2 时钟频率比系统时钟频率快的情况下，则有可能出现不可预料的情况，导致工作寄存器组的值不是期望值，见下图 9.7。

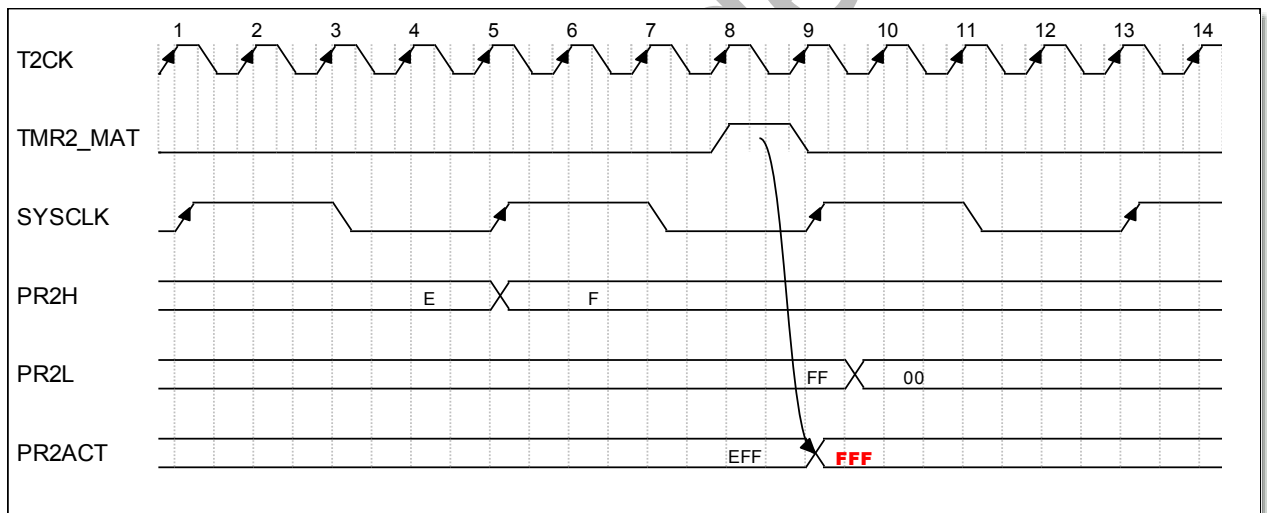


图 9.7 PR2ACT 被更新为意外值 FFF（期望值是 F00）

所以强烈建议更新 PR2 和 xxxDT 只在 TMR2 匹配中断里面做。

1.8. 与 PWM1 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
----	----	------	------	------	------	------	------	------	------	-----

P1ADTL	0xE	P1A 占空比低 8 位						0000 0000
P1BDTL	0xF	P1B 占空比低 8 位						0000 0000
P1CDTL	0x10	P1C 占空比低 8 位						0000 0000
P1DDTL	0x08	P1D 占空比低 8 位						0000 0000
TMR2L	0x11	Timer2 计数器低 8 位						0000 0000
TMR2H	0x13	Timer2 计数器高 8 位						0000 0000
T2CON0	0x12	PR2U	TOUTPS			TMR2ON	T2CKPS	0000 0000
T2CON1	0x9E	—			P1OS	P1BZM	T2CKSRC	---0 0000
P1ADTH	0x14	P1A 占空比高 8 位						0000 0000
P1BDTH	0x15	P1B 占空比高 8 位						0000 0000
P1CDTH	0x1A	P1C 占空比高 8 位						0000 0000
P1DDTH	0x09	P1D 占空比高 8 位						0000 0000
P1CON	0x16	P1AUE	P1DC					0000 0000
P1BR0	0x17	P1BEVT	P1BKS			P1BSS	P1ASS	0000 0000
P1BR1	0x19	P1DSS	P1D2SS			P1CALT	P1DALT	P1CSS
P1OE	0x90	P1COE	P1BOE	P1DOE	—	P1C2SS[1:0]	P1A0NOE	P1A0OE
PR2L	0x91	PR2[7:0]						1111 1111
PR2H	0x92	PR2[11:8]						1111 1111
P1POL	0x99	P1CP	P1BP	P1DP	—	—	—	P1A0NP
P1AUX	0x1E	—	—	—	—	—	—	P1A0P
								000- --00
								P1DF2E
								P1DF2
								---- --00

1.8.1. P1ADTL 寄存器，地址 0x0E

Bit	7	6	5	4	3	2	1	0
Name	P1ADTL							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1ADTL	P1A 占空比寄存器低 8 位

1.8.2. P1BDTL 寄存器，地址 0x0F

Bit	7	6	5	4	3	2	1	0
Name	P1BDTL							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1BDTL	P1B 占空比寄存器低 8 位

1.8.3. P1CDTL 寄存器，地址 0x10

Bit	7	6	5	4	3	2	1	0
Name	P1CDTL							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1CDTL	P1C 占空比寄存器低 8 位

1.8.4. TMR2L 寄存器，地址 0x11

Bit	7	6	5	4	3	2	1	0
Name	TMR2L							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR2L	Timer2 计数器低 8 位

1.8.5. TMR2H 寄存器，地址 0x13

Bit	7	6	5	4	3	2	1	0
Name	TMR2H							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR2H	Timer2 计数器高 8 位

1.8.6. T2CON0 寄存器，地址 0x12

Bit	7	6	5	4	3	2	1	0
Name	PR2U	TOUTPS[3:0]				TMR2ON	T2CKPS[1:0]	
Reset	0	0000				0	00	
Type	WO-1	RW				RW	RW	

Bit	Name	Function
7	PR2U	PR2、P1xDTy 寄存器的软件更新控制位，只写 写 1：把 PR2/P1xDTy 缓冲值分别更新到 PR2 寄存器和 P1xDTy_ACT 写 0：无意义
6:3	TOUTPS[3:0]	TOUTPS<3:0> : Timer2 Output Postscaler Select bits 定时器2输出后分频比选择 0000 = 1:1 后分频比 0001 = 1:2 后分频比 0010 = 1:3 后分频比 0011 = 1:4 后分频比 0100 = 1:5 后分频比 0101 = 1:6 后分频比 0110 = 1:7 后分频比 0111 = 1:8 后分频比 1000 = 1:9 后分频比 1001 = 1:10 后分频比 1010 = 1:11 后分频比 1011 = 1:12 后分频比 1100 = 1:13 后分频比 1101 = 1:14 后分频比 1110 = 1:15 后分频比 1111 = 1:16 后分频比
2	TMR2ON	TMR2ON : Timer2 On bit 打开定时器2 1 = Timer2 打开 0 = Timer2 关闭 PWM1 单脉冲模式下，该位自动清 0
1:0	T2CKPS[1:0]	T2CKPS<1:0> : Timer2 Clock Prescale Select bits 定时器2驱动时钟预分频比选择 00 = Prescaler is 1 01 = Prescaler is 4 1x = Prescaler is 16

1.8.7. P1ADTH 寄存器，地址 0x14

Bit	7	6	5	4	3	2	1	0
Name	P1ADTH							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1ADTH	P1A 占空比寄存器高 8 位

1.8.8. P1BDTH 寄存器，地址 0x15

Bit	7	6	5	4	3	2	1	0
Name	P1BDTH							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1BDTH	P1B 占空比寄存器高 8 位

1.8.9. P1CDTH 寄存器，地址 0x1A

Bit	7	6	5	4	3	2	1	0
Name	P1CDTH							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1CDTH	P1C 占空比寄存器高 8 位

1.8.10. P1DDTL 寄存器，地址 0x8

Bit	7	6	5	4	3	2	1	0
Name	P1DDTL							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
3:0	P1DDTH	P1D 占空比寄存器低 8 位

1.8.11. P1DDTH 寄存器，地址 0x9

Bit	7	6	5	4	3	2	1	0
Name	P1DDTH							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

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

7:0	P1DDTH	P1D 占空比寄存器高 8 位
-----	--------	-----------------

1.8.12. P1CON 寄存器，地址 0x16

Bit	7	6	5	4	3	2	1	0
Name	P1AUE	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P1AUE	PWM1 重启使能位 1 = 故障刹车时，P1BEVT 位在退出关闭事件时自动清零，PWM1 自动重启 0 = 故障刹车时，必须用软件将 P1BEVT 清零以重启 PWM1
6:0	P1DC[6:0]	PWM1 死区时间设置 P1DCn = 预定 PWM 信号应转变为有效与 PWM 信号实际转为有效之间的 T2CK 周期数

1.8.13. P1BR0 寄存器，地址 0x17

Bit	7	6	5	4	3	2	1	0
Name	P1BEVT	P1BKS			P1BSS		P1ASS	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P1BEVT	PWM1 故障事件状态位 1 = 发生了故障事件 0 = 未发生故障事件，PWM1 输出正常工作
6:4	P1BKS[2:0]	PWM1 故障源选择位 000 = 禁止故障刹车功能 001 = BK0 为低电平 010 = BK0 为高电平 011 = LVDW=1 100 = BK0 为低电平或 LVDW=1 101 = BK0 为高电平或 LVDW=1 110 = 保留（禁止故障刹车） 111 = 保留（禁止故障刹车）
3:2	P1BSS	故障下，P1B 管脚的状态 00 = 高阻 01 = 无效电平

		1x = 有效电平
1:0	P1ASS	故障下, P1A 管脚的状态 00 = 高阻 01 = 无效电平 1x = 有效电平

1.8.14. P1BR1 寄存器, 地址 0x19

Bit	7	6	5	4	3	2	1	0
Name	P1DSS		P1D2SS		P1CALT	P1DALT	P1CSS	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	P1DSS	故障下, P1D 管脚的状态 00 = 高阻 01 = 无效电平 1x = 有效电平
5:4	P1D2SS	故障下, [P1D]管脚的状态, 只有当 P1DALT 为 1 时才有效 00 = 高阻 01 = 输出 0 1x = 输出 1
3	P1CALT	[P1C]功能映射选择 1 = 输出 P1C 0 = GPIO
2	P1DALT	[P1D]功能映射选择 1 = 输出 P1D, 或者 P1B 和 P1C 的逻辑波形 0 = GPIO
1:0	P1CSS	故障下, P1C 管脚的状态 00 = 高阻 01 = 无效电平 1x = 有效电平

1.8.15. P1OE 寄存器, 地址 0x90

Bit	7	6	5	4	3	2	1	0
Name	P1COE	P1BOE	P1DOE	—	P1C2SS		P1A0NOE	P1A0OE
Reset	0	0	0	—	0	0	0	0
Type	RW	RW	RW	RO.0	RW	RW	RW	RW

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

7	P1COE	P1C 输出使能，高有效 1 = 允许 P1C 输出到管脚 0 = 禁止 P1C 输出到管脚
6	P1BOE	P1B 输出使能，高有效 1 = 允许 P1B 输出到管脚 0 = 禁止 P1B 输出到管脚
5	P1DOE	P1D 输出使能，高有效 1 = 允许 P1D 输出到管脚 0 = 禁止 P1D 输出到管脚
4	—	—
3:2	P1C2SS	故障下，[P1C]管脚的状态，只有当 P1CALT 为 1 时才有效 00 = 高阻 01 = 输出 0 1x = 输出 1
1	P1A0NOE	P1A0NOE 输出使能，高有效 1 = 允许 P1A0N 输出到管脚 0 = 禁止 P1A0N 输出到管脚
0	P1A0OE	P1A0OE 输出使能，高有效 1 = 允许 P1A0 输出到管脚 0 = 禁止 P1A0 输出到管脚

1.8.16. PR2L 寄存器，地址 0x91

Bit	7	6	5	4	3	2	1	0
Name	PR2L							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR2L	PR2 周期寄存器低 8 位

1.8.17. PR2H 寄存器，地址 0x92

Bit	7	6	5	4	3	2	1	0
Name	PR2H							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR2H	PR2 周期寄存器高 8 位

1.8.18. P1POL 寄存器，地址 0x99

Bit	7	6	5	4	3	2	1	0
Name	P1CP	P1BP	P1A5P	P1A4P	P1A3P	P1A2P	P1A1P	P1A0P
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P1CP	P1C 输出极性设置 1 = P1C 低电平有效 0 = P1C 高电平有效
6	P1BP	P1B 输出极性设置 1 = P1B 低电平有效 0 = P1B 高电平有效
5:0	P1AxP	P1Ax 输出极性设置, x= 0~5 1 = P1Ax 低电平有效 0 = P1Ax 高电平有效

1.8.19. P1AUX 寄存器，地址 0x1E

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	P1DF2E	P1DF2
Reset	—	—	—	—	—	—	0	0
Type	RO-0	RO-0	RO-0	RO-0	RO-0	RO-0	RW	RW

Bit	Name	Function
7:2	N/A	保留位，读 0
1	P1DF2E	当 P1DALT 为 1 时，[P1D]管脚第 2 功能使能 1 = 输出 P1B 和 P1C 的同或，或者异或 0 = 输出 P1D
0	P1DF2	[P1D]管脚第 2 功能选择 1 = 输出 P1B 和 P1C 的同或 0 = 输出 P1B 和 P1C 的异或

2 应用范例

```
//*****
/* 文件名:  TEST_FT62F21x_PWM1.c
* 功能:      FT62F21x-PWM1 功能演示
* IC:        FT62F211 SOP8
* 晶振:      16M/4T
* 说明:      P1A0 输出 10kHz 占空比 50%的波形(实测 9.8kHz)
*
*
*              FT62F211 SOP8
*
*      -----
*  NC-----|1(PA4)      (PA3)8|-----P1A0
*  NC-----|2(TKCAP)    (PA0)7|-----NC
*  VDD-----|3(VDD)     (PA1)6|-----NC
*  GND-----|4(VSS)     (PA2)5|-----NC
*
*      -----
*/

//*****
#include "SYSCFG.h";
#include "FT62F21X.h";
//*****

#define OSC_16M      0X70
#define OSC_8M       0X60
#define OSC_4M       0X50
#define OSC_2M       0X40
#define OSC_1M       0X30
#define OSC_500K     0X20
#define OSC_250K     0X10
#define OSC_32K      0X00

#define WDT_256K     0X80
#define WDT_32K      0X00
//*****
//*****宏定义*****
#define unchar      unsigned char
#define uint        unsigned int
#define ulong       unsigned long
/*-----
* 函数名:  中断处理程序
* 功能:    定时器 2 中断
* 输入:    无
* 输出:    无
*-----*/

void interrupt ISR(void)          //PIC_HI-TECH 使用
{
```

```

//定时器 2 的中断处理*****
if(TMR2IE && TMR2IF)           //100us 中断一次
{
    TMR2IF = 0;

    //DemoPortOut = ~DemoPortOut; //翻转电平
}
}
/*-----
* 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 = WDT_32K|OSC_16M|0X00;
    //OSCCON = 0B01110000;           //WDT 32KHZ IRCF=111=16MHZ/4=4MHZ,0.25US/T

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

    PORTA = 0B00000000;
    TRISA = 0B00000000;           //PA 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

    OPTION = 0B00001000;          //Bit3=1 WDT MODE,PS=000=1:1 WDT RATE
    //Bit7(PAPU)=0 由 WPUA 决定是否上拉

    MSCON = 0B00000000;
}
/*-----
* 函数名称: DelayUs
* 功能:     短延时函数 --16M-4T--大概快 1%左右.
* 输入参数: Time 延时时间长度 延时时长 Time*2 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(98);           //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: PWM1_INITIAL  
* 功能: PWM1 初始化函数  
* 输入: 无  
* 输出: 无  
-----*/
```

```
void PWM1_INITIAL(void)  
{  
    T2CON=0B00000001; //T2 预分频 1:4, 后分频 1:1  
    //BIT7: 0: 无意义; 1: 把 PR2/P1xDTy 缓冲值分别更新到 PR2 寄存器和 P1xDTy_ACT
```

```

//BIT6~BIT3: 定时器 2 输出后分频比选择 0000:1:1;0001:1:2;.....1:16
//BIT2:0:关闭定时器 2; 1: 打开定时器 2
//BIT1~0:定时器 2 预分频选择 00:1;01:4;1x:16

T2CON1=0B00000000; //T2 时钟来自系统时钟, PWM 连续模式
//BIT4: PWM 模式选择 0:连续模式; 1: 单脉冲模式
//BIT3: 0:PWM 模式; 1: 蜂鸣器模式
//Timer2 时钟源选择: 000: 指令时钟; 001: 系统时钟; 010: HIRC 的 2 倍频; 100: HIRC;
101: LIRC

TMR2H=0; //定时器 2 计数寄存器
TMR2L=100;

PR2H=0; //周期= (PR2+1) *Tt2ck*TMR2 预分频
PR2L=100;

P1ADTH=0; //脉宽=P1xDt*Tt2ck*TMR2 预分频
P1ADTL=50;

P1OE=0B00000001; //P1A0 输出使能
//BIT7: 0:禁止 P1C 输出到管脚;1:允许 P1C 输出到管脚
//BIT6: 0:禁止 P1B 输出到管脚;1:允许 P1B 输出到管脚
//BIT5: 0:禁止 P1D 输出到管脚;1:允许 P1D 输出到管脚
//BIT3~BIT2: 故障下, [P1C]管脚的状态, 只有当 P1CALT 为 1 时才有效 00=高阻 01=
输出 0 1x=输出 1
//BIT1: 0:禁止 P1A0N 输出到管脚;1:允许 P1A0N 输出到管脚
//BIT0: 0:禁止 P1A0 输出到管脚;1:允许 P1A0 输出到管脚

P1POL=0B00000000; //P1A0 高电平有效
//BIT7: 0:P1C 高电平有效;1:P1C 低电平有效
//BIT6: 0:P1B 高电平有效;1:P1B 低电平有效
//BIT5: 0:P1D 高电平有效;1:P1D 低电平有效
//BIT1: 0:P1A0N 高电平有效;1:P1A0N 低电平有效
//BIT0: 0:P1A0 高电平有效;1:P1A0 低电平有效

P1CON=0B00000000;
//BIT7:PWM1 重启使能位
//1 = 故障刹车时, P1BEVT 位在退出关闭事件时自动清零, PWM1 自动重启
//0 = 故障刹车时, 必须用软件将 P1BEVT 清零以重启 PWM1
//BIT6~0:PWM1 死区时间设置
//P1DCn = 预定 MPWM 信号应转变为有效与 PWM 信号实际转为有效之间的 T2CK 周期
数

TMR2IF=0; //清 TIMER2 中断标志

```

```
TMR2IE=1;           //使能 TIMER2 的中断
TMR2ON=1;           //打开定时器 2
PEIE=1;             //使能外设中断
GIE=1;              //使能全局中断

}

/*-----
 * 函数名: main
 * 功能:   主函数
 * 输入:   无
 * 输出:   无
-----*/
void main()
{
    POWER_INITIAL(); //系统初始化
    PWM1_INITIAL();  //初始化 PWM1
    while(1)
    {
        NOP();
    }
}
```

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 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, 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.