



北京福星晓程电子科技股份有限公司

Beijing Fuxing Xiaocheng Electronic Technology Stock Co.,Ltd

PL5010 电力线载波通信专用芯片 数据手册

密级：公开使用

版本：V1.1

日期：2015.6.15

北京福星晓程电子科技股份有限公司

目 录

目 录.....	1
1 芯片概况.....	4
1.1 芯片简介.....	4
1.2 芯片特性.....	4
1.3 技术指标和封装参数.....	5
1.4 引脚框图.....	8
1.5 引脚定义.....	8
2 增强型 8051 的微处理器 MCU.....	11
2.1 PL5010 芯片的存储器及 CPU 时钟介绍.....	11
2.2 8051 MCU 资源配置.....	29
2.3 中断控制.....	30
2.4 IR38kHz 红外调制功能.....	43
2.5 看门狗定时器.....	45
3 电能计量模块.....	46
3.1 电能计量工作原理.....	46
3.2 寄存器.....	49
3.3 计量公式.....	61
3.4 参考电路.....	63
3.5 校表流程.....	64
3.6 应用例程.....	64
4 载波通信.....	67
4.1 功能简述.....	67
4.2 编程指南.....	67
4.3 寄存器.....	68
4.4 应用例程.....	73
5 四通道硬件定时器.....	77

5.1 功能简述.....	77
5.2 编程指南.....	77
5.3 相关寄存器.....	77
5.4 应用例程.....	79
6 ISO7816 接口.....	81
6.1 功能简述.....	81
6.2 编程指南.....	81
6.3 寄存器.....	82
7 FGEN 可编程频率发生器.....	89
7.1 功能简述.....	89
7.2 编程指南.....	89
7.3 寄存器.....	89
8 PWM.....	91
8.1 功能简述.....	91
8.2 编程指南.....	91
8.2 寄存器.....	91
9 差分串行通信单元.....	93
9.1 功能简述.....	93
9.2 编程指南.....	93
9.3 寄存器.....	93
10 低功耗电源管理模块.....	97
10.1 功能简述.....	97
10.2 低功耗设计.....	97
10.3 编程指南.....	98
10.4 寄存器.....	98
11 RTC 控制单元.....	102
11.1 32768 实时钟晶振连接方式.....	102

11.2 功能简述.....	102
11.3 寄存器.....	102
12 SPI 接口.....	107
12.1 功能简述.....	107
12.2 编程指南.....	107
12.3 寄存器.....	107
13 嵌入式 ROM 数据存储器.....	110
13.1 功能简述.....	110
13.2 编程指南.....	110
13.3 寄存器.....	111
13.4 应用例程.....	113
14 程序与数据的下载.....	115
附录 1 特殊功能寄存器（SFR）表.....	116
附录 2 扩展寄存器表 外部设备地址分配表.....	119
附录 3 电能计量单元扩展寄存器表.....	125

1 芯片概况

1.1 芯片简介

PL5010 是一颗拥有多项自主知识产权的多功能，高性能，低功耗单相多功能智能电表专用 SOC 芯片，内部集成了增强型 8051 处理器，高精度多功能计量引擎，窄带载波通信，电源管理，ISO7816 接口，带数字校准 RTC 电路，支持 ISP 和 IAP 编程接口，其中计量输入采用 3 路高性能 16 位，sigma-delta ADC，采样数据速率为 6.25KHz，支持双通道同时计量有功功率，电压和电流有效值，支持断相防窃电，载波通信支持过零点和连续传输方式，载波兼容 PL4000 系列，封装形式采用 LQFP64 封装。

1.2 芯片特性

- 采用 0.13um 超大规模数/模混合 CMOS 制造工艺
- 拥有多项自主知识产权的 SoC (System on Chip) 设计
- 内嵌增强型 8051 兼容微处理器
- 内置高精度数字多功能电能计量电路，计量标准完全符合国标 GB/T 17883 和 GB/T 17215 (等效于 IEC687/1036)。计量精度在 1000:1 的动态范围内误差小于 0.1%
- 电流通道内置可程序设定增益放大器 (PGA)
- 内置双通道电流采样、正/负功指示数字逻辑电路
- 内置有功功率计量电路
- 内置功率因数、电压/电流有效值、相电压频率测量电路
- 内置电压质量监测电路
- 内置小信号校表加速功能
- 采用 DBPSK 窄带调制方式和自适应可变速率直序扩频通信技术，最高通信速率达 500bps
- 通信频带和信道带宽符合欧洲 EN50065 等行业规范
- 内置数字同步锁相器，支持过零点通信模式
- 全数字收发机，两套独立的收发系统分别实现过零点和连续通信方式

- 内置 6 阶高性能带通开关电容滤波
- 内置 256 bytes + 7936 bytes SRAM
- 内置 64K bytes 嵌入式 FLASH 存储器
- 内置三个可灵活配置的全双工多功能 UART 和两个不需要定时器产生波特率的全双工多功能 UART
- 内置四个 8/16 位定时/计数器，除此之外还有四个独立的不带中断的硬件定时器，一个看门狗定时器以及四个外部中断源
- 内置双通道 ISO7816 接口电路，完全符合国标 GB/T 16649
- 内置可编程频率发生器电路
- 内置 PWM 产生电路
- 内置可数字频率校正的实时钟，并具有秒脉冲输出
- 内置可中断的差分串行通信单元
- 内置标准 SPI 接口
- 内置电源管理模块，支持掉电睡眠及唤醒模式
- 内置完善的电源电压监测电路
- 内置 $2.5V \pm 8\%$ 电压源基准
- 内置串行程序存储器编程接口，支持在系统编程（ISP）和在应用编程（IAP）
- 单一 5V 电源供电；所有 I/O 兼容 3.3V 电平输入
- 采用 LQFP64 封装
- 温度适用范围（工业级标准） $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$

1.3 技术指标和封装参数

1.3.1 电气参数

电特性（ $T_a = 25^{\circ}\text{C}$ ， $AVDD = 5V$ ， $DVDD = 5V$ ，晶体为 12MHz， $F_{osc} = 6\text{MHz}$ ）

1.3.2 极限参数

测量项目	符号	测试条件	测量点	最小	典型	最大	单位
工作电流	I_{wmax}	所有功能			10		mA

		使能					
睡眠模式下电流	Isleep	VBATI=3.6V			2		uA
工作电压	Vw	标准			5±10%		V
参考电压	VREF				2.5±5%		V
参考电压温度系数					30		ppm/°C
时钟输入	OSC				12		MHz
Schmitt 输入	L→H	VDD = 5V		2.3		2.9	V
	H→L	VDD = 5V		1.7		2	V
输出	VOH	IOH = 15mA		4.0			V
	VOL	IOL = 15mA				0.4	V
直流电源抑制 (输出频率误差)		±250mV			0.2%		
交流电源抑制 (输出频率误差)		200mV, 100Hz, 纹波			±0.3%		

1.3.3 封装参数

项目	符号	极值	单位
储藏温度	TSTR	-60 — +150	°C
结温	TSR	+150	°C
焊接温度 (焊接 10 秒)	TILT	+260	°C
汽相焊接 (60 秒)	TS	+215	°C
红外焊接 (15 秒)	TIF	+220	°C
工作温度	TOPR	-40 — +85	°C
电源电压	AVDD, DVDD	6	V

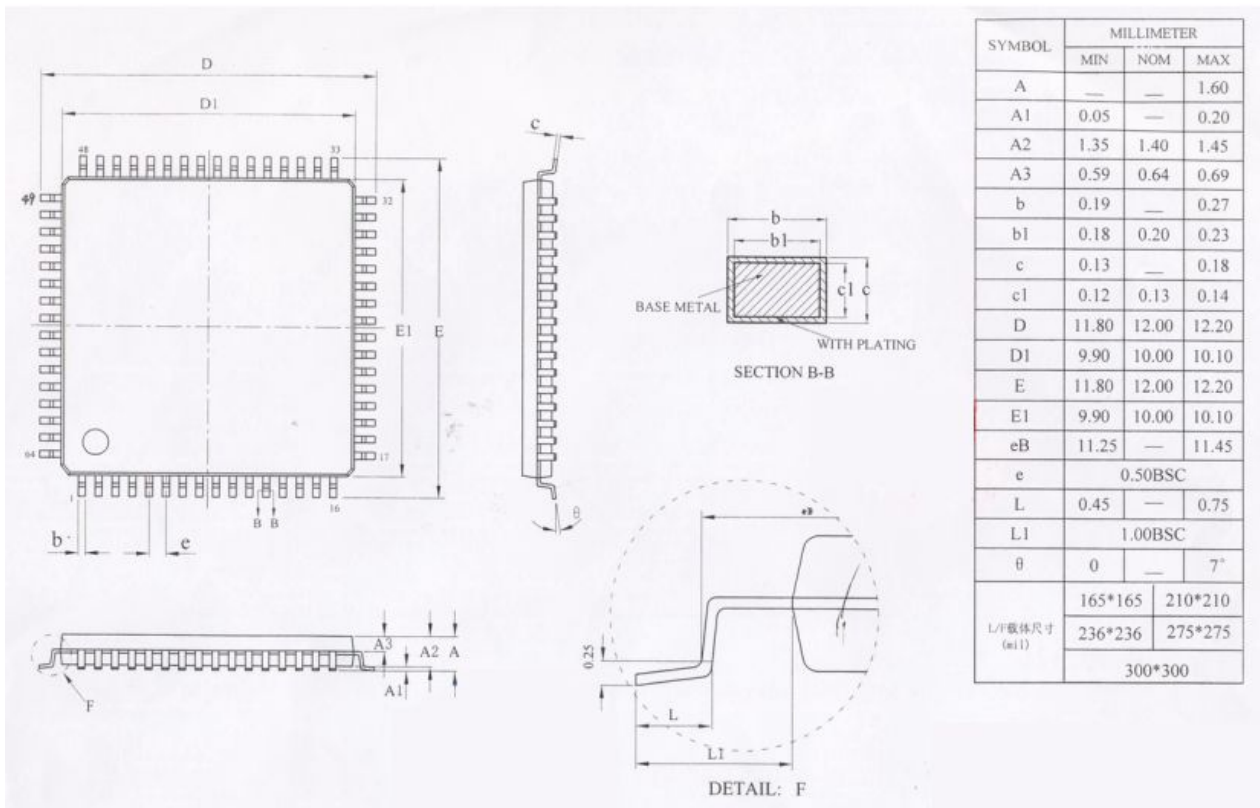


图2 PL5010 芯片封装图

1.4 引脚框图

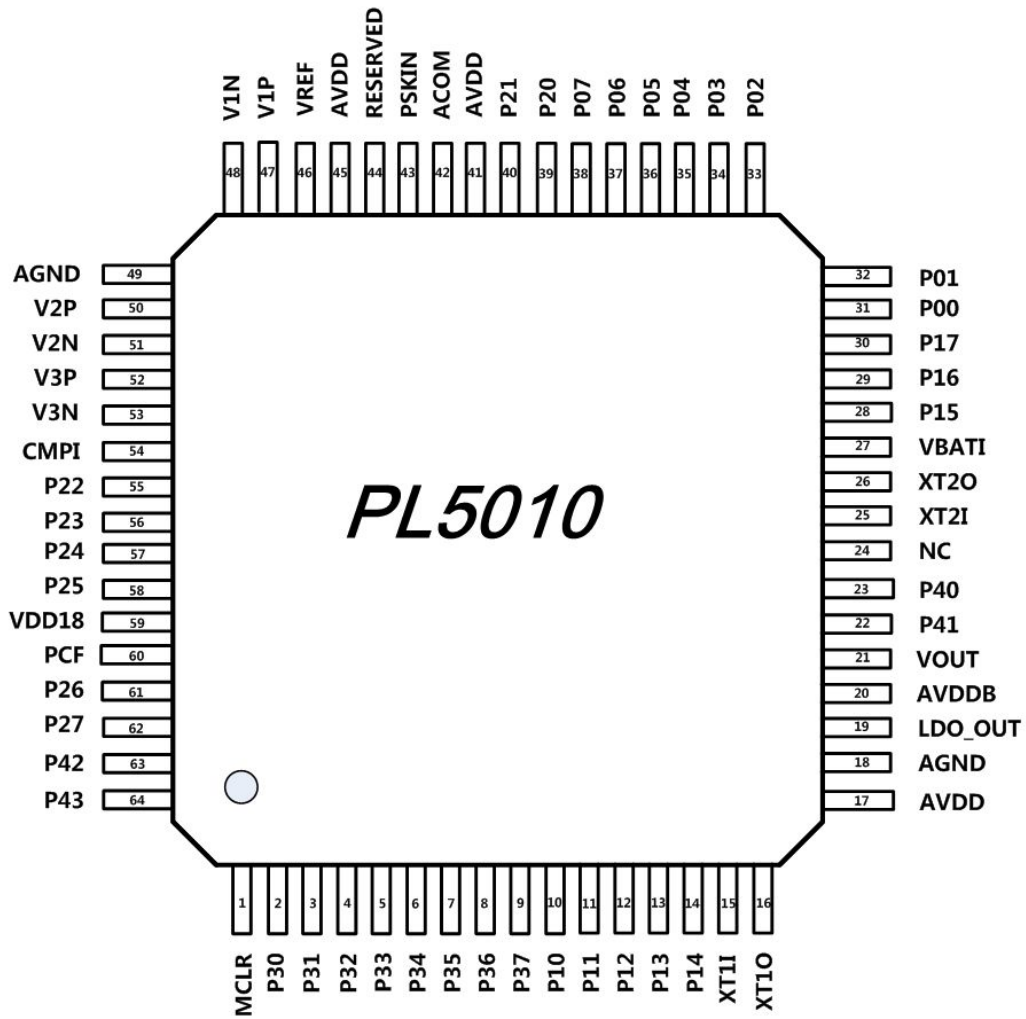


图 3 PL5010 引脚分布图

1.5 引脚定义

引脚名称	序号	类型	说明
MCLR	1	I	芯片复位管脚，上升沿、下降沿均触发 (正常工作时要为高电平)
P30/RXD0/SCL	2	I/O	UART0_RXD
			芯片编程接口 SCL
P31/TXD0/SDA	3	I/O	UART0_TXD
			芯片编程接口 SDA
P32/INT0	4	I/O	外部中断 0

P33/INT1	5	I/O	外部中断 1
P34/T0	6	I/O	定时器 0 计数脉冲输入
P35/T1	7	I/O	定时器 1 计数脉冲输入
P36/RXD3	8	I/O	UART3_RXD
P37/TXD3	9	I/O	UART3_TXD
P10/RXD1	10	I/O	UART1_RXD
P11/TXD1	11	I/O	UART1_TXD
P12/INT2	12	I/O	外部中断 2
P13/INT3	13	I/O	外部中断 3
P14	14	I/O	IO
XT1I	15	I	12MHz 晶体振荡器输入，对地接 22pF 电容
XT1O	16	O	12MHz 晶体振荡器输出，对地接 22pF 电容
AVDD	17	P	模拟电源 5V
AGND	18	G	模拟地
LDO_OUT	19	P	内部 LDO，对地接 1uF 电容
AVDDB	20	P	睡眠模式下的唤醒引脚(P40、P41)专用供电上拉电源，对地接 1uF 电容
VOUT	21	O	睡眠模式对外供电引脚，对地接 1uF 电容
P41	22	I/O	睡眠模式下的唤醒引脚（下降沿触发，需要接 AVDDB 作为上拉供电电源）
P40	23	I/O	睡眠模式下的唤醒引脚（下降沿触发，需要接 AVDDB 作为上拉供电电源）
NC	24		NC
XT2I	25	I	32768Hz 实时钟振荡器输入，对地接 15pF 电容
XT2O	26	O	32768Hz 实时钟振荡器输出，对地接 15pF 电容
VBATI	27	I	低功耗电池供电电源
P15/PWM	28	I/O	PWM 输出
P16/T2	29	I/O	定时器 2 计数脉冲输入
P17/T3	30	I/O	定时器 3 计数脉冲输入
P00/RXD2/IR_I	31	I/O	UART2_RXD/IR38K_IN
P01/TXD2/IR_O	32	I/O	UART2_TXD/IR38K_OUT
P02/IC_DIN	33	I/O	IC 数据
P03/IC_CLK	34	I/O	IC 时钟
P04/ESAM_CLK	35	I/O	ESAM 时钟
P05/ESAM_DIO	36	I/O	ESAM 数据
P06/DIFF_OUTP	37	I/O	差分串行通信同向端
P07/DIFF_OUTN	38	I/O	差分串行通信反向端

P20/PSKO	39	I/O	载波发送端口
P21/CLK50HZ	40	I/O	50Hz 信号输入接口(用于载波过零点发送模式)
AVDD	41	P	数字电源 5V
ACOM	42	O	内部电源, 对地接 104 电容
PSKIN	43	I	载波输入接口
RESERVED	44	I	保留引脚, 悬空
AVDD	45	P	数字电源 5V
VREF	46	O	内置电压基准电源, 对地接 104 电容
V1P	47	I	电流通道 1 差分正向输入端
V1N	48	I	电流通道 1 差分负向输入端
AGND	49	G	模拟地 (接地)
V2P	50	I	电流通道 2 差分正向输入端
V2N	51	I	电流通道 2 差分负向输入端
V3P	52	I	电压通道 差分正向输入端
V3N	53	I	电压通道 差分负向输入端
CMPI	54	I	主电源掉电监测比较器输入端 (掉电检测功能不用时接地)
P22/SPI_CS	55	I/O	SPI_NSS
P23/SPI_SCK	56	I/O	SPI_SCK
P24/SPI_DI	57	I/O	SPI_MISO
P25/SPI_DO	58	I/O	SPI_MOSI
VDD18	59	O	内部 1.8V 电源, 对地接 106 电容
PCF	60	O	有功瞬时功率脉冲输出 (校表用)
P26/CLK1HZ	61	I/O	可配置 1Hz 信号输出
P27/FGEN	62	I/O	可编程频率发生器输出
P42/RXD4	63	I/O	UART4_RXD
P43/TXD4	64	I/O	UART4_TXD

2 增强型 8051 的微处理器 MCU

2.1 PL5010 芯片的存储器及 CPU 时钟介绍

2.1.1 PL5010 存储器配置

PL5010 内置的嵌入式准 16 位微处理器存储器配置方式和 8051 基本相同，在物理结构上分成 4 个储存空间：PL5010A 有 44K Byte 程序存储器，4K Byte 用户自定义固件程序存储区，16K Byte 嵌入式数据存储器。PL5010B 有 60K Byte 程序存储器，4K Byte 用户自定义固件程序存储区，没有嵌入式数据存储器。PL5010 都有 8K Byte SRAM（其中 7936 Byte 片外 SARM 及 256Byte 片内 RAM）。从用户的使用角度来说，PL5010 和 8051 一样，其存储器地址空间分为四类。

PL5010A 存储空间包括程序存储区、数据存储区及固件存储区。PL5010B 只包括程序存储区和固件存储区。

1. 片内统一编址 0000H~AFFFH 的 44K（PL5010B 为 60K）字节的程序存储空间（MOVC 可访问）；
2. 片内统一编址 F000H~FFFFH 的 4K 用户自定义固件程序存储空间；
3. 16K 嵌入式数据存储器（PL5010B 没有），需要通过外部扩展地址寄存器和指定地址的外部 SRAM 区访问；
4. 7936Byte SRAM，地址为 0000H~1EFFH（用 13 位地址，MOVX 可访问）；
5. 256Byte SRAM（用 8 位地址，用指令 MOV 访问）。

PL5010 内嵌的微处理器可访问的存储地址空间分为程序存储空间、数据存储空间及 SRAM 空间。

（一）程序存储地址空间

程序存储器可用于存储程序及相关常量。程序存储器使用 16 位程序计数器（PC）寻址，占用 44K Byte。

（二）数据存储地址空间

数据存储区用于存储程序中一些重要的变量，掉电仍保存在 Data Flash 中。PL5010 的数据存储空间大小 16KByte，128Pages*128Byte，程序可通过 SRAM 地址 0x1E00 开始区域

及相关指令进行读写操作。

(三)

SRAM 存储空间 8K Byte，分为内部 256Byte 空间和外部 7936Byte 空间。

1. 7936 Byte 外部 RAM

7936 Byte 外部 RAM 和 256 Byte 内部 RAM 的空间低地址 00H~FFH 是重叠的，7936 Byte 外部 RAM 和 256 Byte 内部 RAM 分别通过指令 MOVX 和 MOV 来访问；

2. 256 Byte 内部 RAM 和特殊功能寄存器

该 RAM 的最大寻址空间为 256 个字节单元，它们又分为两部分，低 128 字节(00H~7FH)是可直接寻址（或间接寻址）的低段 RAM 区；高 128 字节（80H~FFH）分别映射为可直接寻址的特殊功能寄存器（SFR）区和可间接寻址的高段 RAM 区。

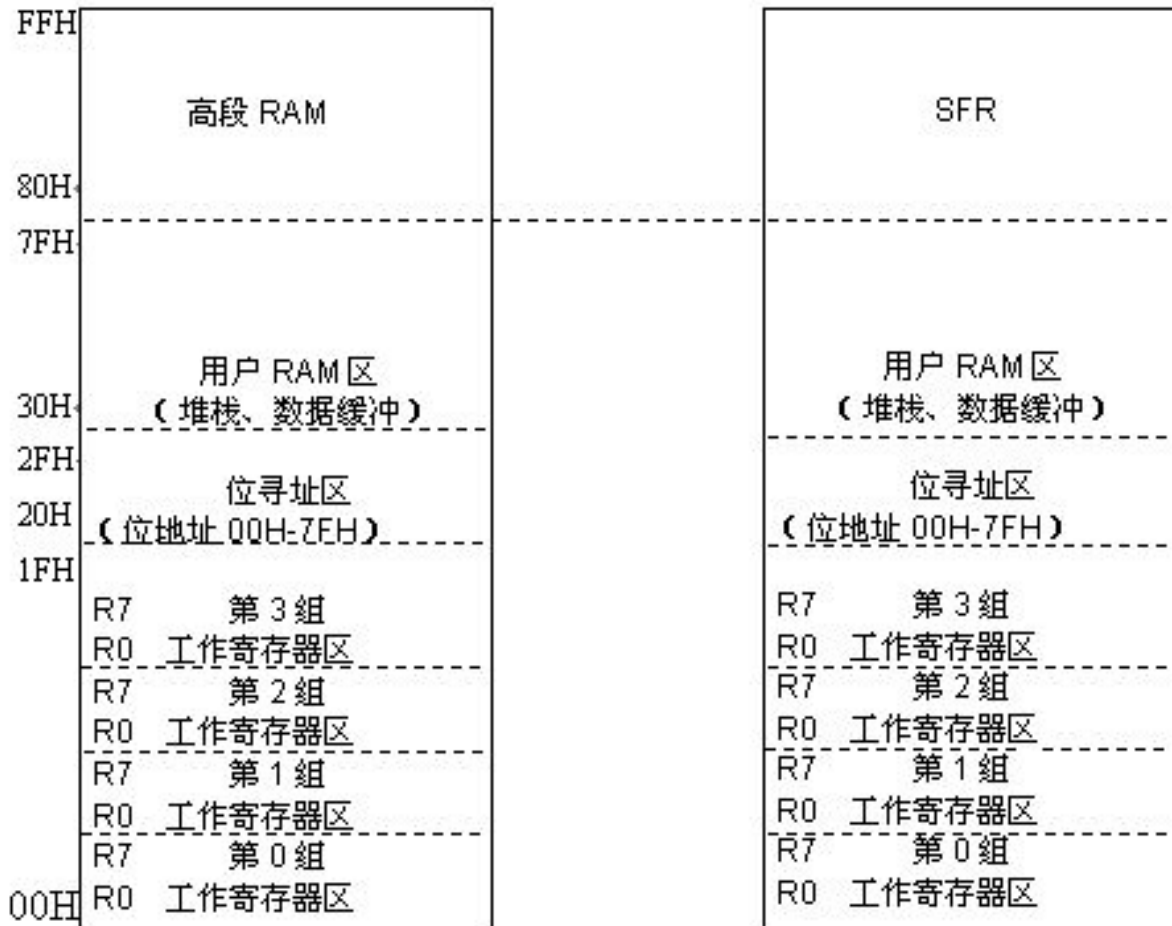


图 4 数据储存器地址空间

① 低 128 字节 RAM

PL5010 内嵌的微处理器可访问的 32 个工作寄存器与 RAM 安排在同一队列空间里，

统一编址并使用同样的寻址方式（直接寻址和间接寻址）。

00H~1FH 地址安排为四组工作寄存器区，每组有 8 个工作寄存器（R0~R7），共占 32 个单元。通过对程序状态字 PSW 中 RS1、RS0 的设置，每组寄存器均可选作 CPU 的当前工作寄存器组。若程序中并不需要四组，那么其余可用作一般 RAM 单元。CPU 复位后，选中第 0 组工作寄存器。

工作寄存器区后的 16 字节单元（20H~2FH）为位寻址区，可用位寻址方式访问。共计 $16 \times 8 = 128$ 个位，这 128 个位的位地址为 00H~7FH，其位地址分布见表 2。

表 1 工作寄存器地址表

组	RS1	RS0	R0	R1	R2	R3	R4	R5	R6	R7
0	0	0	00H	01H	02H	03H	04H	05H	06H	07H
1	0	1	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
2	1	0	10H	11H	12H	13H	14H	15H	16H	17H
3	1	1	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH

表 2 RAM 位寻址区位地址表

字节地址	MSB 位 地 址							LSB	
2FH	7F	7E	7D	7C	7B	7A	79	78	
2EH	77	76	75	74	73	72	71	70	
2DH	6F	6E	6D	6C	6B	6A	69	68	
2CH	67	66	65	64	63	62	61	60	
2BH	5F	5E	5D	5C	5B	5A	59	58	
2AH	57	56	55	54	53	52	51	50	
29H	4F	4E	4D	4C	4B	4A	49	48	
28H	47	46	45	44	43	42	41	40	
27H	3F	3E	3D	3C	3B	3A	39	38	
26H	37	36	35	34	33	32	31	30	
25H	2F	2E	2D	2C	2B	2A	29	28	
24H	27	26	25	24	23	22	21	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	16	15	14	13	12	11	10	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	

低 128 字节 RAM 单元地址范围也是 00H~7FH，PL5010 内嵌的微处理器采用不同寻址方式来加以区分，即访问 128 个位地址用位寻址方式，访问低 128 字节单元用直接寻址和间接寻址，这样就可以区分开 00H~7FH 是位地址还是字节地址。这些可寻址位，通过执行特定指令可直接对某一位操作，如置 1、清 0 或判 1、判 0 等，可用作软件标志位或用于位（布尔）处理。

② 高 128 字节 RAM

高 128 字节 RAM 单元地址范围是 80H~FFH，PL5010 内嵌的微处理器只能通过间接寻址方式对来访问这部分存储器。

③ 特殊功能寄存器（SFR）

PL5010 内嵌的微处理器可直接寻址访问的高 128 字节地址空间被分配给了特殊功能寄存器，除程序计数器 PC 外，还有 44 个专用寄存器（SFR），也称特殊功能寄存器，它们离散地分布在 80H~0FFH 的地址空间中。访问特殊功能寄存器仅允许使用直接寻址方式。这些特殊功能寄存器见附录 1。

在 44 个特殊功能寄存器 SFR 中，有 15 个特殊功能寄存器具有位寻址能力，它们的字节地址正好能被 8 整除。

下面介绍部分特殊功能寄存器（SFR）。

2.1.2 特殊功能寄存器 SFR

● 累加器 ACC（S: E0H）

BIT	7	6	5	4	3	2	1	0
FIELD	ACC							
RESET	0							

Bit Number	Bit Mnemonic	Function
7~0	ACC	累加器 ACC 是 PL5010 最常用、最繁忙的 8/16 位特殊功能寄存器，许多指令的操作数取自于 ACC，许多运算结果也存放于 ACC 中。在指令系统中采用 A 作为累加器 ACC 的助记符。

● 寄存器 B（S:F0H）

BIT	7	6	5	4	3	2	1	0
FIELD	B							

RESET	0
-------	---

Bit Number	Bit Mnemonic	Function
7~0	B	在乘、除指令中，用到了 8/16 位 B 寄存器。乘法指令的两个操作数分别取自 A 和 B，乘积存于 B、A 两个 8/16 位寄存器中。除法指令中，A 中存放被除数，B 中放除数，商存放于 A 中，B 中存放余数。在双字节的乘、除法，还用到了 ACC_H 和 B_H 来参与运算。

● 程序状态寄存器 PSW (S:D0H)

BIT	7	6	5	4	3	2	1	0
FIELD	CY	AC	F0	RS1	RS0	OV		P
RESET	0	0	0	0	0	0		0

Bit Number	Bit Mnemonic	Function
7	CY	进位标志位 在执行加法（或减法）运算指令时，如果运算结果最高位（位 7）向前有进位（或借位），CY 位由硬件置“1”；如运算结果最高位（位 7）无进位（或借位），则 CY 由硬件清“0”。CY 也是 PL5010 在进行位操作（布尔操作）时的位累加器 C。
6	AC	半进位标志位 也称辅助进位标志。当执行加法（或减法）操作时，其运算结果（和或差）产生由低半字节（位 3）向高半字节有半进位（或借位）时 AC 位将被硬件自动置“1”，否则 AC 被硬件自动清“0”。
5	F0	用户标志位 用户可以根据自己的需要对 F0 位赋予一定含义，由用户置位、复位，作为软件标志。
4	RS1	工作寄存器组选择控制位 这两位的值决定选择哪一组工作寄存器为当前工作寄存器组。由用户用软件改变 RS1 和 RS0 值的组合，以切换当前选用的工作寄存器组。
3	RS0	PL5010 上电复位后，(RS1) = (RS0) = 0，微处理选择第 0 组为当前工作寄存器组。根据需要，可利用传送指令对 PSW 整字节操作或用位操作指令改变 RS1 和 RS0 的状态，以切换当前工作寄存器组。这样的设置对程序中保护现场提供了方便。
2	OV	溢出标志位 当进行补码运算时，如有溢出，即当运算结果超出-128~+127 的范围时，OV 位由硬件自动置 OV = 1；无溢出时，OV = 0。
1		

0	P	奇偶校验标志位 每条指令执行完后，该位始终跟踪指示累加器 A 中 1 的个数。如结果 A 中有奇数个“1”，则置 P=1，否则 P=0。常用于校验串行通信中的数据传送是否出错。
---	---	--

● 堆栈指针 SP (S:81H)

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	0	0	0	0	0	1	1	1

堆栈指针 SP 为 8 位特殊功能寄存器，SP 的内容（即堆栈指针）可指向 PL5010 片内 00H~FFH RAM 的任何单元。系统复位后，SP 初始化为 07H，即指向 07H 的 RAM 单元。下面介绍一下堆栈的概念。

PL5010 中内嵌的微处理器同大多数微机处理器一样，设有堆栈。在片内 RAM 中专门开辟出来一个区域，数据的存取是以“后进先出”的结构方式处理的。这种数据结构方式对于处理中断、调用子程序都非常方便。

堆栈的操作有两种：一种叫数据压入（PUSH）；另一种叫数据弹出（POP）。假若有 8 个 RAM 单元，每个单元都在其右面编有地址。栈顶有堆栈指针 SP 自动管理。每次进行压入或弹出操作以后，堆栈指针便自动调整以保持指示堆栈顶部的位置。这些操作在这里就不叙述了。

在使用堆栈之前，先给 SP 赋值，以规定堆栈的起始位置，称为栈底。当数据压入堆栈后 SP 自动加 1，即 RAM 地址单元加 1 以指出当前栈顶位置。PL5010 的这种堆栈结构属于向上生长型的堆栈（有的微处理器使用向下生长的堆栈）。

PL5010 的堆栈指针 SP 是一个双向计数器。在压栈时 SP 内容自动增值，出栈时自动减值，存取信息必须遵循“后进先出”或“先进后出”的原则。

● 数据指针 DPTR (S:83H,82H)

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	0	0	0	0	0	0	0	0

DPTR 是一个 16 位的特殊功能寄存器，其高位字节寄存器用 DPH 表示（地址为 83H），低位字节寄存器用 DPL 表示（地址为 82H）。DPTR 既可以作为一个 16 位寄存器来处理，

也可以作为二个独立的 8 位寄存器 DPH 和 DPL 使用。

● I/O 端口 P0~P3 (S:80H,90H,A0H,B0H)

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	1	1	1	1	1	1	1	1

P0~P3 为四个 8 位特殊功能寄存器，分别是四个并行 I/O 端口的锁存器。它们都有字节地址，每一个口锁存器还有位地址，所以每一条 I/O 线独立地用作输入或输出时，数据可以锁存；作输入时，数据可以缓冲。其复用功能在引脚说明中给出。

I/O 端口某一位用于输入信号时，对应的锁存器位须先置“1”。

● STATUS 状态寄存器 (S:87H) *

BIT	7	6	5	4	3	2	1	0
FIELD	SMOD	SMOD1	WDT	PU		LP_MOD	VBFO	PFI
RESET	0	0	0	0	—	0	0	0

Bit Number	Bit Mnemonic	Function
7	SMOD	串口 0 波特率加倍控制位 SMOD 为 1，其波特率加倍，SMOD 为 0，其波特率不加倍。
6	SMOD1	串口 1 的波特率加倍控制 用法同 SMOD。
5	WDT	复位标志位 当系统复位后，该标志位自动置 1，可软件清零。
4	PU	上电标志位 当系统上电运行后，该标志位自动置 1，可软件清零。
3		
2	LP_MOD	系统电压标志位 检测 AVDD 引脚 5V 系统电压。系统有电时为 0，无电时为 1。
1	VBFO	电池电压检测标志位 检测 VBATI 引脚电压。电池电压大于等于 2.8V 为 0，低于 2.6V 为 1。
0	PFI	掉电标志位 当 CMPI 引脚的输入电压高于 2.8V 时，该位自动置 1。

● PCON 电源控制及波特率选择寄存器 (S:87H) *

BIT	7	6	5	4	3	2	1	0
FIELD	SMOD	SMOD1					STOP	IDLE
RESET	0	0	—	—	—	—	0	0

Bit Number	Bit Mnemonic	Function
7	SMOD	串口 0 波特率加倍控制位 SMOD 为 1，其波特率加倍，SMOD 为 0，其波特率不加倍。
6	SMOD1	串口 1 的波特率加倍控制 用法同 SMOD。
5~2		
1	STOP	系统停止控制位 当该位置 1 后，系统停止，只有外部中断可唤醒。
0	IDLE	系统休眠控制位 当该位置 1 后，系统休眠，定时和外部中断均可唤醒。

*注：寄存器 STATUS 和 PCON 地址都为 87H,读操作针对 STATUS 写操作针对 PCON，但是 STATUS 的 WDT 和 PU 标志位特殊可读可清 0。

● TCON0 定时/计数器控制寄存器 0 (S:88H)

BIT	7	6	5	4	3	2	1	0
FIELD	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	TF1	T1 计数溢出标志位 当 T1 计数溢出时，该位由硬件置 1，申请中断，进入中断后自动清零，也可软件清零。
6	TR1	T1 计数运行控制位 由软件置 1 或清 0。
5	TF0	T0 计数溢出标志位 当 T0 计数溢出时，该位由硬件置 1，申请中断，进入中断后自动清零，也可软件清零。
4	TR0	T0 计数运行控制位 由软件置 1 或清 0。
3	IE1	外部中断 1 的申请标志 当检测到有外部中断信号时，由硬件自动置 1。
2	IT1	外部中断 1 触发方式控制位 当 IT1 = 1 时外部中断 1 为边缘触发方式，当 IT1 = 0 是外部中断 1 为

		电平触发方式。
1	IE0	外部中断 0 的申请标志 当检测到有外部中断信号时，由硬件自动置 1。
0	IT0	外部中断 0 触发方式控制位 当 IT0 = 1 时外部中断 0 为边缘触发方式，当 IT0 = 0 是外部中断 0 为电平触发方式。

● TCON1 定时/计数器控制寄存器 1 (S:C8H)

BIT	7	6	5	4	3	2	1	0
FIELD	TF3	TR3	TF2	TR2	IE3	IT3	IE2	IT2
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	TF3	T3 计数溢出标志位 当 T2 计数溢出时，该位由硬件置 1，申请中断，进入中断后自动清零，也可软件清零。
6	TR3	T3 计数运行控制位 由软件置 1 或清 0。
5	TF2	T2 计数溢出标志位 当 T2 计数溢出时，该位由硬件置 1，申请中断，进入中断后自动清零，也可软件清零。
4	TR2	T2 计数运行控制位 由软件置 1 或清 0。
3	IE3	外部中断 3 的申请标志 当检测到有外部中断信号时，由硬件自动置 1。
2	IT3	外部中断 3 触发方式控制位 当 IT3 = 1 时外部中断 3 为边缘触发方式，当 IT3 = 0 是外部中断 3 为电平触发方式。
1	IE2	外部中断 2 的申请标志 当检测到有外部中断信号时，由硬件自动置 1。
0	IT2	外部中断 2 触发方式控制位 当 IT2 = 1 时外部中断 2 为边缘触发方式，当 IT2 = 0 是外部中断 2 为电平触发方式。

● TMOD0/ TMOD1 定时/计数器工作方式，状态寄存器 0/1 (S:89H/C9H)

BIT	7	6	5	4	3	2	1	0
------------	----------	----------	----------	----------	----------	----------	----------	----------

FIELD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
RESET	0	0	0	0	0	0	0	0

TMOD0 的高 4 位用于控制 T1 的工作模式,低 4 为用于控制 T0 的工作模式。TMOD1 的低 4 位用于控制 T2 的工作模式,高 4 位用于控制 T3 的工作模式。

Bit Number	Bit Mnemonic	Function
7、3	GATE	门控制位 GATE = 0 时, 只要用软件将 TR0 (TR1 或 TR2 或 TR3) 置 1 就可以启动定时器, 而不管 INT0 (INT1/2) 的电平高低。GATE = 1 时, 只有 INT0 (INT1/2/3) 引脚为高电平且由软件将 TR0 (TR1 或 TR2 或 TR3) 置 1 才能启动定时器。
6、2	C/T	计数器/定时器方式选择位 C/T = 0 时, 设置为定时方式。C/T = 1 时, 设置为计数器方式。
5、1	M1	操作模式控制位 定义见下表
4、0	M0	操作模式控制位 定义见下表

M0 和 M1: 操作模式控制位。定义如下:

M1	M0	工作模式	功能描述
0	0	模式 0	13 位计数器
0	1	模式 1	16 位计数器
1	0	模式 2	8 位自装载计数器
1	1	模式 3	定时器 0: 分成两个 8 位计数器, 定时器 1: 停止计数, 定时器 2: 没有这种模式

● WDT_CON 看门狗控制寄存器 (S:8EH)

BIT	7	6	5	4	3	2	1	0
FIELD	RST1	RST0	WDT2	WDT1	WDT0			
RESET	0	0	0	0	0	—	—	—

Bit Number	Bit Mnemonic	Function
7	RST1	复位时间长度控制位
6	RST0	
5	WDT2	控制看门狗喂狗时间长度

4	WDT1	当 WDT2/WDT1/1DT0 = 000 时，看门狗复位间隔时间为 87 毫秒，当 WDT2/WDT1/1DT0 = 001 时，看门狗复位间隔时间为原来的 2 倍，即 174 毫秒。依此类推，当 WDT2/WDT1/1DT0 = 111 时，看门狗复位间隔时间为 87×8 =696 毫秒。WDT2/WDT1/1DT0 默认为 0。
3	WDT0	
2~0		

RST1 RST0	0 0	0 1	1 0	1 1
复位时间	88ms	176ms	264ms	352ms

● SCON0 串行口控制寄存器 0 (S:98H)

BIT	7	6	5	4	3	2	1	0
FIELD	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
RESET	0	0	0	0	0	0	0	0

SCON0 为串口 0 的控制寄存器，SCON1 为串口 1 的控制寄存器，SCON2 为串口 2 的控制寄存器。三个寄存器的对应位除了作用对象不一样外，功能及操作方式都一样，因此各位定义参考 SCON1。

● SCON1 串行口控制寄存器 1 (S:C0H)

BIT	7	6	5	4	3	2	1	0
FIELD	E1SM0	E1SM1	E1SM2	E1REN	E1TB8	E1RB8	E1TI	E1RI
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	E1SM0	串行口工作方式选择 对应关系见下表
6	E1SM1	
5	E1SM2	多机通讯控制位 主要用于方式 2 和方式 3。若置 E1SM2 = 1，则允许串口 1 多机通讯，若 E1SM2 = 0，则不属于多机通讯情况。
4	E1REN	允许接收控制位 由软件置 1 或清 0，只有当 E1REN = 1，串口 1 才允许接收，E1REN = 0 时，串口 1 禁止接收数据。
3	E1TB8	发送数据的第 9 位 (D8) 装入 E1TB8 中在方式 2 或方式 3 中，根据发送数据的需要由软件置位或清 0，在方式 0 或方式 1 中，该位没用。
2	E1RB8	接收数据的第 9 位，在方式 2 或方式 3 中，接收到的第 9 位数据放在

		E1RB8 中，在方式 1 中，如 E1SM2 = 0，E1RB8 中存放的是接收到的停止位，方式 0 中未用该位。
1	E1TI	发送中断标志位 在串口 1 将一帧数据发送完毕后被置 1，串口 1 发送中断被响应后，E1TI 不会自动清 0，必须软件清 0。
0	E1RI	接收中断标志位 在串口 1 接收到一帧数据后由硬件置 1，在串口 1 接收中断被响应后，E1RI 不会自动清 0，必须软件清 0。

E1SM0/E1SM1：串口 1 工作方式选择。对应关系如下：

E1SM0	E1SM1	工作方式	说明	波特率
0	0	方式 0	同步移位寄存器	Fosc/12
0	1	方式 1	10 位异步收发	由定时器控制
1	0	方式 2	11 位异步收发	Fosc/32 或发 Fosc/64
1	1	方式 3	11 位异步收发	由定时器控制

● SCON2 串口控制寄存器 2 (S:F8H)

BIT	7	6	5	4	3	2	1	0
FIELD	E2SM0	E2SM1	E2SM2	E2REN	E2TB8	E2RB8	E2TI	E2RI
RESET	0	0	0	0	0	0	0	0

SCON0 为串口 0 的控制寄存器，SCON1 为串口 1 的控制寄存器，SCON2 为串口 2 的控制寄存器。三个寄存器的对应位除了作用对象不一样外，功能及操作方式都一样，因此各位定义参考 SCON1。

● IE (扩展) 中断允许控制寄存器 (S:A8H)

BIT	7	6	5	4	3	2	1	0
FIELD	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EA	中断允许总控制位 EA = 0，屏蔽所有中断申请；EA = 1，CPU 开放中断。
6	ES1	串口 1 中断允许位 ES1 = 0，禁止串口 1 中断，ES1 = 1，允许串口 1 中断。
5	ET2	定时/计数器 T2 的溢出中断允许位

		ET2 = 0, 禁止 T2 中断, ET2 = 1, 允许 T2 中断。
4	ES0	串口 0 中断允许位 ES0 = 0, 禁止串口 0 中断, ES0 = 1, 允许串口 0 中断。
3	ET1	定时/计数器 T1 的溢出中断允许位 ET1 = 0, 禁止 T1 中断, ET1 = 1, 允许 T1 中断。
2	EX1	外部中断 1 中断允许位 EX1 = 0, 禁止外部中断 1 中断, EX1 = 1, 允许外部中断 1 中断。
1	ET0	定时/计数器 T0 的溢出中断允许位 ET0 = 0, 禁止 T0 中断, ET0 = 1, 允许 T0 中断。
0	EX0	外部中断 0 中断允许位 EX0 = 0, 禁止外部中断 0 中断, EX0 = 1, 允许外部中断 0 中断。

● EIE (扩展) 中断允许控制寄存器 (S:A9H)

BIT	7	6	5	4	3	2	1	0
FIELD					EX3	ES2	ET3	EX2
RESET	—	—	—	—	0	0	0	0

Bit Number	Bit Mnemonic	Function
7~4		
3	EX3	外部中断 3 中断允许位 EX3 = 0, 禁止外部中断 3 中断, EX3 = 1, 允许外部中断 3 中断。
2	ES2	串口 2 中断允许位 ES2 = 0, 禁止串口 2 中断, ES2 = 1, 允许串口 2 中断。
1	ET3	定时/计数器 T3 的溢出中断允许位 ET3 = 0, 禁止 T3 中断, ET3 = 1, 允许 T3 中断。
0	EX2	外部中断 2 中断允许位 EX2 = 0, 禁止外部中断 2 中断, EX2 = 1, 允许外部中断 2 中断。

● IP 扩展中断优先级控制寄存器 (S:B8H)

BIT	7	6	5	4	3	2	1	0
FIELD		PS1	PT2	PS0	PT1	PX1	PT0	PX0
RESET	—	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7		
6	PS1	串行口 1 中断优先级控制位

5	PT2	定时器/计数器 T2 中断优先级控制位
4	PS0	串行口 0 中断优先级控制位
3	PT1	定时器/计数器 T1 中断优先级控制位
2	PX1	外部中断 1 中断优先级控制位
1	PT0	定时器/计数器 T0 中断优先级控制位
0	PX0	外部中断 0 中断优先级控制位

● EIP 扩展中断优先级控制寄存器 (S:B9H)

BIT	7	6	5	4	3	2	1	0
FIELD					PX3	PS2	PT3	PX2
RESET	—	—	—	—	0	0	0	0

Bit Number	Bit Mnemonic	Function
7~4		
3	PX3	外部中断 3 中断优先级控制位
2	PS2	串行口 2 中断优先级控制位
1	PT3	定时器/计数器 T3 中断优先级控制位
0	PX2	外部中断 2 中断优先级控制位

● EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

PL5010 通过扩展地址寄存器 (D9H) 来选择和访问系统扩展的外部设备：实时钟，ISO7816 通信模块，差分串行通信单元，FGEN 可编程频率发生器，载波通信模块，嵌入式 ROM 数据存储器等。当程序通过扩展地址寄存器选择相应的外设后，就可以在扩展数据寄存器 (D8H) 中对相应外设读出或写入数据。

可访问的寄存器见附录 2 扩展寄存器表。

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

PL5010 通过扩展地址寄存器 (D9H) 来选择和访问系统扩展的外部设备：实时钟，

ISO7816 通信模块，差分串行通信单元，FGEN 可编程频率发生器，载波通信模块,嵌入式 ROM 数据存储器等。当程序通过扩展地址寄存器选择相应的外设后，就可以在扩展数据寄存器（D8H）中对相应外设读出或写入数据。

可访问的寄存器见附录 2 扩展寄存器表。

● PMU_ADR 电能计量扩展地址寄存器（S:E9H）

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

PL5010 通过电能计量单元扩展地址寄存器（E9H）来选择和访问系统扩展的电能计量单元外部设备。当程序通过电能计量单元扩展地址寄存器选择相应的外设后，就可以在电能计量单元扩展数据寄存器（E8H）中对相应外设读出或写入数据。

可访问的寄存器见附录 3 电能计量单元扩展寄存器表。

● PMU_DAT 电能计量扩展数据寄存器（S:E8H）

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

PL5010 通过电能计量单元扩展地址寄存器（E9H）来选择和访问系统扩展的电能计量单元外部设备。当程序通过电能计量单元扩展地址寄存器选择相应的外设后，就可以在电能计量单元扩展数据寄存器（E8H）中对相应外设读出或写入数据。

可访问的寄存器见附录 3 电能计量单元扩展寄存器表。

● TL0/TH0, TL1/TH1, TL2/TH2, TL3/TH3 是定时/计数器 0/1/2/3 的计数寄存器
地址 S:8AH/8CH, 8BH/8DH, CCH/CDH, CEH/CFH

TH0/TL0 是定时/计数器 0 的高低字节, TH1/TL1 是定时/计数器 1 的高低字节, TH2/TL2 是定时/计数器 2 的高低字节, TH3/TL3 是定时/计数器 3 的高低字节。四者的用法一致，这里以定时/计数器 2 加以说明。当定时/计数器配置为定时模式 1 时，TH2/TL2 组成一个 16 位的寄存器（模式 1），以 $F_{osc}/12$ 的速度自动加 1，直到加到 FFFFH 时，加 1 为 0000H。在计数模式中，每当 T2 口来 1 个脉冲，TH2/TL2 组成的寄存器就自动加 1，直到加到 FFFFH 时，加 1 为 0000H。

● WDT_RST 复位控制器 (S:8FH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7~0		看门狗计数器复位寄存器 PL5010 内置看门狗复位逻辑。看门狗计数器是一个 15 位的寄存器，以一定的速度进行递减，当该计数器递减溢出系统产生复位。当向 WDT_RST 寄存器写入“A1H”后，看门狗计数器自动复位为 7FFF。

2.1.3 CPU 时钟

2.1.3.1 片内振荡器及时钟信号的产生

PL5010 片内有一高增益反相放大器，用于构成振荡器。反向放大器输入端为 XT1I，输出端为 XT1O。在 XT1I 和 XT1O 两端跨接石英晶体及两个电容就构成了稳定自激振荡器。电容器 C1 和 C2 通常都取 22pF 左右，对振荡频率有微调作用。振荡频率范围是 1.2~12MHz。

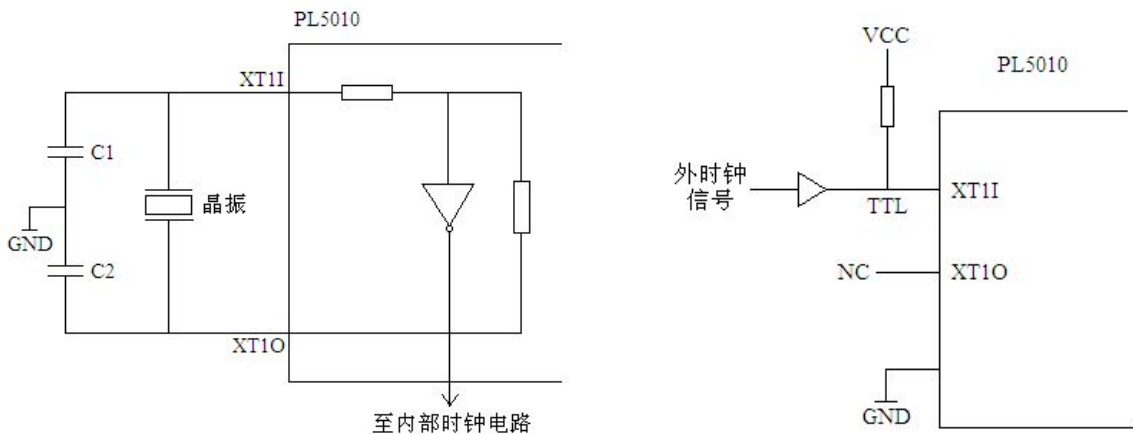


图 9 片内片外时钟信号产生示意图

PL5010 也可使用外部振荡脉冲信号，由 XT1I 端引脚输入，XT1O 悬空。外部振荡脉冲源方式常用于多块 PL5010 同时工作，以便于同步，对外部脉冲信号要求高低电平的持续时间大于 20ns，一般为低于 12MHz 的方波。晶体振荡器的振荡信号从 XT1O 端输出到片

内的时钟发生器上。

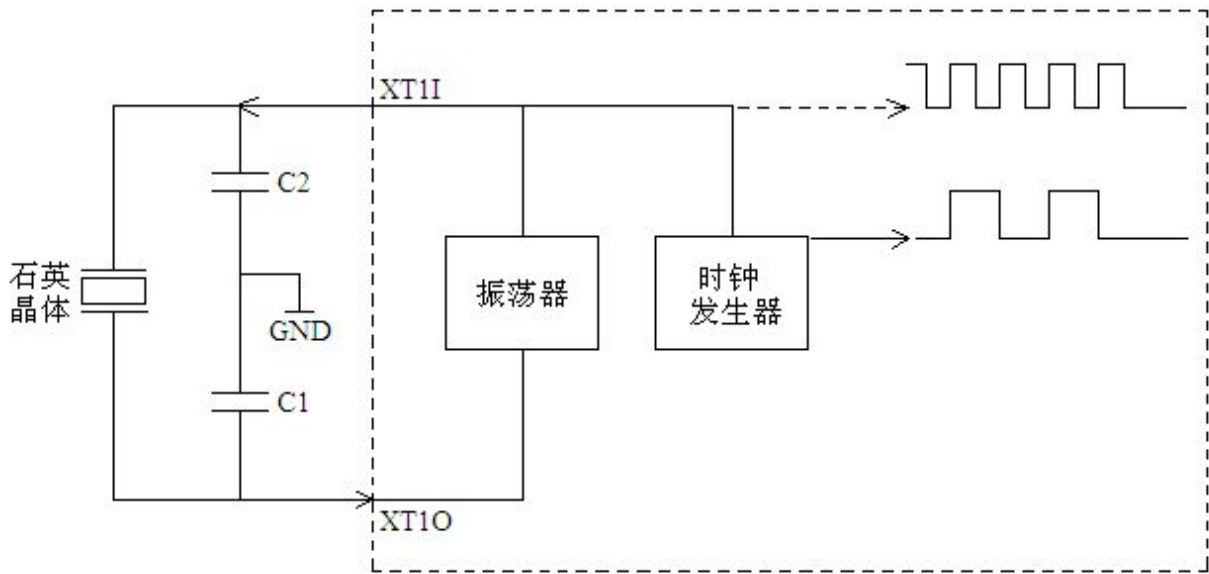


图 10 CPU 时钟信号产生示意图

PL5010 时钟发生器是一个二分频触发器电路，它将振荡器的信号频率除以 2 得到 F_{osc} 即给 PL5010 内置 CPU 提供的时钟信号。时钟信号的周期称为机器状态时间，它是外部振荡周期的 2 倍。在每个时钟周期，CPU 就以此时钟为基本节拍指挥各个部件协调地工作。

2.1.3.2 机器周期和指令周期

PL5010 内置的 CPU 完全兼容 MCS-51 单片机指令集，同时还增加了双字节乘除法指令，计算的一条指令由若干个字节组成。执行一条指令需要多少时间则以机器周期为单位。所谓一个机器周期是指 CPU 访问存储器一次所需要的时间。例如取指令、读存储器、写存储器等等。PL5010 内置 CPU 的一个机器周期为 2 个振荡周期，12MHZ 晶体振荡器，工作主时钟频率为二分频 6MHZ。每条指令都由一个或几个机器周期组成。在 PL5010 CPU 系统中，有单周期指令、双周期指令，三周期指令，四周期指令及九周期指令。九周期指令用于单、双字节乘、除指令。指令的运算速度和它的机器周期数直接相关，机器周期数较少则执行速度快。在编程时要注意选用具有同样功能而机器周期数少的指令。

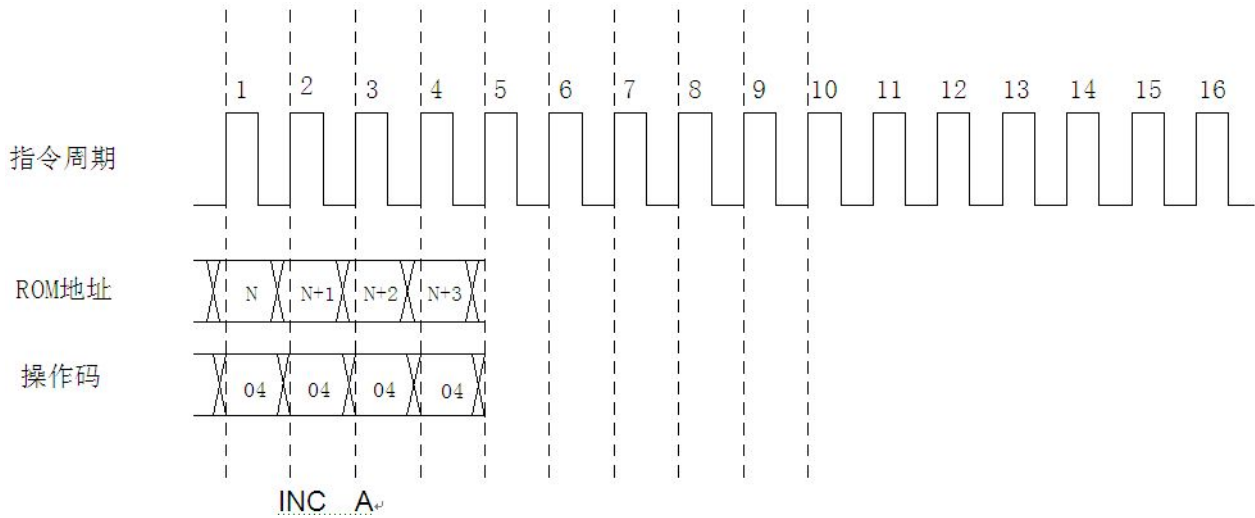
1. CPU 取指令、执行指令周期时序

每一条指令的执行都可以包括取指令和执行两个阶段。在取指令阶段，CPU 从内部 FLASH 中取出指令操作码及操作数，然后再执行这条指令。

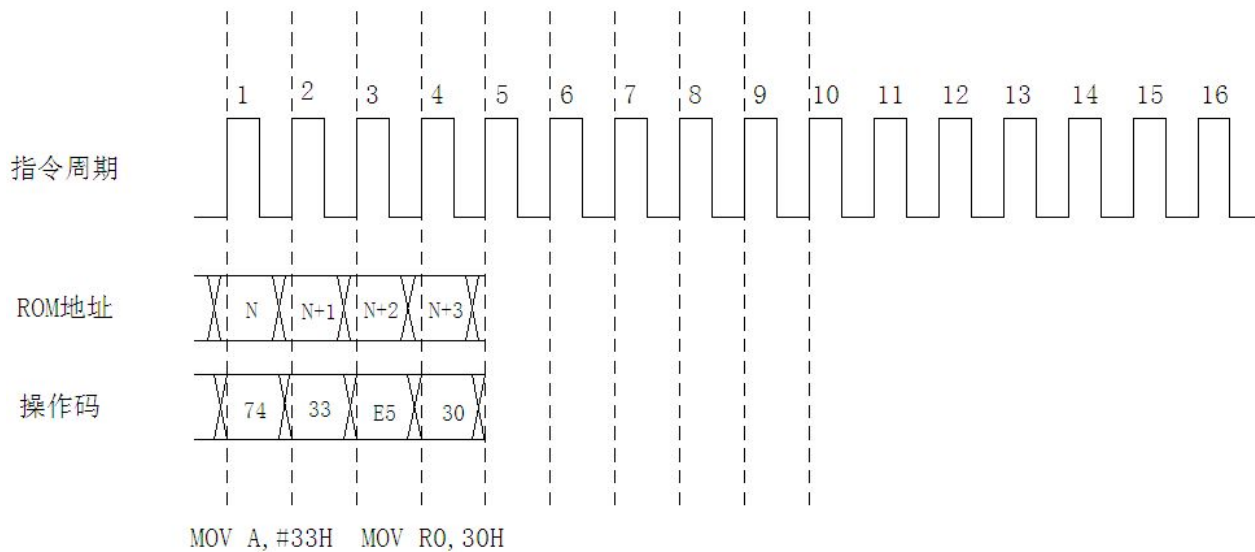
在 51 指令集中，根据各种操作的繁简程度，其指令可由单字节、双字节和三字节组成。

CPU 将地址送入 FLASH 的地址输入端，FLASH 将相应的操作码送入输出端口，在指令周期的上升沿，将操作码被锁存于指令寄存器内。下面列举了几种典型指令的取值和执行时序：

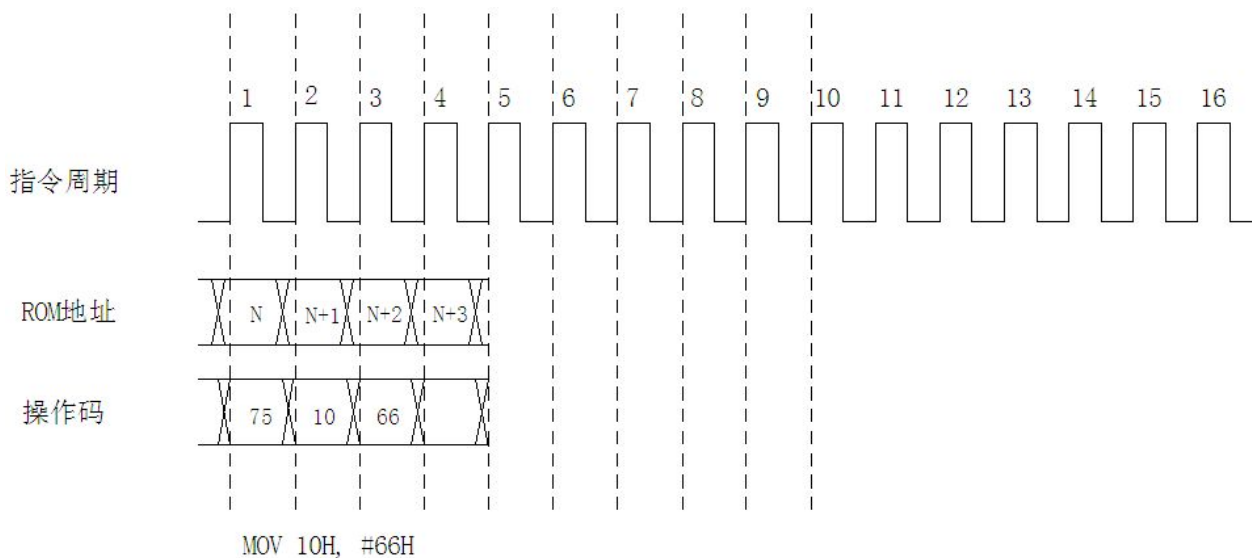
- 单字节单周期指令，如：INC A



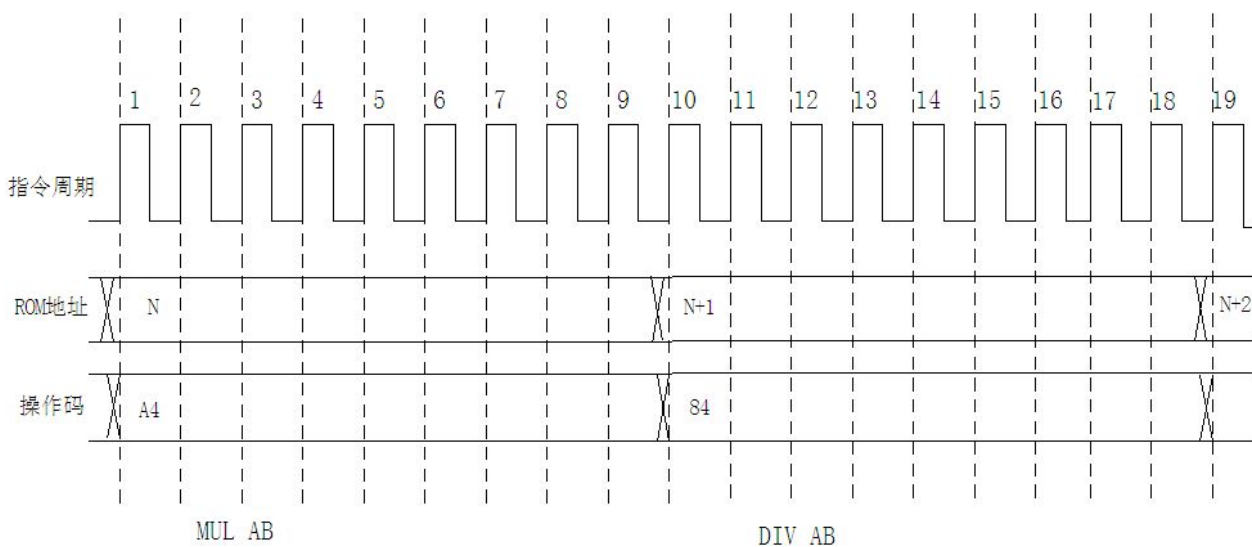
- 双字节双周期指令 如：MOV A, #data; MOV Rn, direct



- 三字节三周期指令，如：MOV direct, #data



- 单、双字节乘法，如：MUL AB；DIV AB



2.2 8051 MCU 资源配置

内嵌的 MCU 采用 8/16 位的增强型 8051 兼容微处理器。嵌入式 MCU 的主时钟由外部晶振 12MHz 二分频后得到，即 8051 的时钟频率为 $F_{osc} = 6\text{MHz}$ ，由于内部的译码和执行机构采用了 RISC 指令流水技术，使得程序平均运行速度较标准 8051 微处理器快达 8 倍之多。

1. 嵌入式 MCU 的地址空间分配：

- 内部 SRAM：256 bytes (00H~FFH)，其中 80H~FFH 为 SFR
- 外部 SRAM：7936 bytes (0000H~1EFFH)

2. 嵌入式 MCU 具有：

- 5 个全双工 UART 串行口 (UART2 支持 IR38K) ,其中 UART0/1/2 与 8051 使用方式一致; UART3/4 不需要定时器产生波特率, 停止位为 2 位
- 看门狗复位逻辑
- 四个 8/16 位定时/计数器
- 四个外部中断
- 一个通用的 SPI 接口;
- 一路频率可配和占空比可调的 PWM 输出;
- 一个异步半双工字符传输方式的差分通信模块;

2.3 中断控制

PL5010 的 8051 微处理器有 15 个中断源, 优先级顺序从高到低, 并受 8051 中断优先级寄存器 IP 控制。其操作控制为标准 8051 方式, 中断名称及其入口地址如下:

中断名称	中断入口地址	中断号
外部中断 INT0:	0003H	0
定时/计数器 T0 溢出中断:	000BH	1
外部中断 INT1\载波过零中断:	0013H	2
定时/计数器 T1 溢出中断:	001BH	3
串口中断 UART0:	0023H	4
定时器 T2 溢出中断:	002BH	5
串口中断 UART1:	0033H	6
外部中断 INT2\载波连续中断:	003BH	7
定时/计数器 T3 溢出中断:	0043H	8
串口中断 UART2:	004BH	9
外部中断 INT3:	0053H	10
串口中断 UART3:	005BH	11
串口中断 UART4:	0063H	12
SPI 中断 SPI:	006BH	13
差分通信单元中断 DIFF:	0073H	14

其中 INT0, INT1 中断 T0, T1 中断和 UART0, UART1 中断与普通 8051 完全一致。

下面对其他几个中断详细说明。外部中断 INT1、INT2 在作为载波中断使用时不再作为外部中断使用。

2.3.1 定时/计数器 T2、T3 溢出中断

定时/计数器 T2 溢出中断与 T0 和 T1 的溢出中断相似。当 TH2/TL2 (CCH/CDH) 计数溢出后，TF2 置 1，同时产生中断申请，T3 与 T2 一致，仅寄存器地址不同。中断入口地址为：002BH (T2)，0043H (T3)，相关的控制器/位如下：

● IE 中断使能寄存器 (S: A8H)

BIT	7	6	5	4	3	2	1	0
FIELD	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EA	中断允许总控制位 EA = 0，屏蔽所有中断申请；EA = 1，CPU 开放中断。
6	ES1	串口 1 中断允许位 ES1 = 0，禁止串口 1 中断，ES1 = 1，允许串口 1 中断。
5	ET2	定时/计数器 T2 的溢出中断允许位 ET2 = 0，禁止 T2 中断，ET2 = 1，允许 T2 中断。
4	ES0	串口 0 中断允许位 ES0 = 0，禁止串口 0 中断，ES0 = 1，允许串口 0 中断。
3	ET1	定时/计数器 T1 的溢出中断允许位 ET1 = 0，禁止 T1 中断，ET1 = 1，允许 T1 中断。
2	EX1	外部中断 1 中断允许位 EX1 = 0，禁止外部中断 1 中断，EX1 = 1，允许外部中断 1 中断。
1	ET0	定时/计数器 T0 的溢出中断允许位 ET0 = 0，禁止 T0 中断，ET0 = 1，允许 T0 中断。
0	EX0	外部中断 0 中断允许位 EX0 = 0，禁止外部中断 0 中断，EX0 = 1，允许外部中断 0 中断。

● EIE 扩展中断使能控制寄存器 (S: A9H)

BIT	7	6	5	4	3	2	1	0
FIELD	EDS	ESPI	ES4	ES3	EX3	ES2	ET3	EX2
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EDS	差分串行通信单元中断允许位 EDS = 0, 禁止差分串行通信中断, EDS = 1, 允许差分串行通信中断。
6	ESPI	SPI 中断允许位 ESPI = 0, 禁止 SPI 中断, ESPI = 1, 允许 SPI 中断。
5	ES4	串口 4 中断允许位 ES4 = 0, 禁止串口 4 中断, ES4 = 1, 允许串口 4 中断。
4	ES3	串口 3 中断允许位 ES3 = 0, 禁止串口 3 中断, ES3 = 1, 允许串口 3 中断。
3	EX3	外部中断 3 中断允许位 EX3 = 0, 禁止外部中断 3 中断, EX3 = 1, 允许外部中断 3 中断。
2	ES2	串口 2 中断允许位 ES2 = 0, 禁止串口 2 中断, ES2 = 1, 允许串口 2 中断。
1	ET3	定时/计数器 T3 的溢出中断允许位 ET3 = 0, 禁止 T3 中断, ET3 = 1, 允许 T3 中断。
0	EX2	外部中断 2 中断允许位 EX2 = 0, 禁止外部中断 2 中断, EX2 = 1, 允许外部中断 2 中断。

● IP 扩展中断优先级控制寄存器 (S:B8H)

BIT	7	6	5	4	3	2	1	0
FIELD		PS1	PT2	PS0	PT1	PX1	PT0	PX0
RESET	—	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7		
6	PS1	串行口 1 中断优先级控制位
5	PT2	定时器/计数器 T2 中断优先级控制位
4	PS0	串行口 0 中断优先级控制位
3	PT1	定时器/计数器 T1 中断优先级控制位
2	PX1	外部中断 1 中断优先级控制位
1	PT0	定时器/计数器 T0 中断优先级控制位
0	PX0	外部中断 0 中断优先级控制位

● EIP 扩展中断优先级控制寄存器 (S:B9H)

BIT	7	6	5	4	3	2	1	0
FIELD	PDS	PSPI	PS4	PS3	PX3	PS2	PT3	PX2
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	PDS	差分串行通信单元中断优先级控制位
6	PSPI	SPI 中断优先级控制位
5	PS4	串行口 4 中断优先级控制位
4	PS3	串行口 3 中断优先级控制位
3	PX3	外部中断 3 中断优先级控制位
2	PS2	串行口 2 中断优先级控制位
1	PT3	定时器/计数器 T3 中断优先级控制位
0	PX2	外部中断 2 中断优先级控制位

● TMOD1 定时器工作方式 (S:C9H)

BIT	7	6	5	4	3	2	1	0
FIELD	GATE_T3	C/T_T3	M1_T3	M0_T3	GATE_T2	C/T_T2	M1_T2	M0_T2
RESET	0	0	0	0	0	0	0	0

TMOD1 的高 4 位用于控制 T3 的工作模式，低 4 位用于控制 T2 的工作模式。

以下 x 用于区别定时器 T2 和 T3。

Bit Number	Bit Mnemonic	Function
7、3	GATE_Tx	门控制位 GATE_Tx = 0 时，只要用软件将 TRx 置 1 就可以启动定时器，而不管 INTx 的电平高低。GATE_Tx = 1 时，只有 INTx 引脚为 1 且由软件将 TRx 置 1 才能启动定时器。
6、2	C/T_Tx	计数器/定时器方式选择位 C/T_Tx = 0 时，设置为定时方式， C/T_Tx = 1 时，设置为计数器方式。
5、1	M1_Tx	操作模式控制位 定义见下表
4、0	M0_Tx	操作模式控制位 定义见下表

M0 和 M1：操作模式控制位。定义如下：

M1_Tx	M0_Tx	工作模式	功能描述
-------	-------	------	------

0	0	模式 0	13 位计数器
0	1	模式 1	16 位计数器
1	0	模式 2	8 位自装载计数器
1	1	模式 3	定时器 0: 分成两个 8 位计数器, 定时器 1: 停止计数, 定时器 2: 没有这种模式

● PCON1 波特率选择寄存器 (S:86H)

BIT	7	6	5	4	3	2	1	0
FIELD	SMOD2	T1_SEL	T2_SEL	T3_SEL				
RESET	0	0	0	0	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	SMOD2	串口 2 波特率加倍控制位 SMOD2 为 1, 其波特率加倍, SMOD2 为 0, 其波特率不加倍。
6	T1_SEL	用于选择计数器/定时器 1 作为定时器使用时的计数速率, T1_SEL = 1 时, 为 Fosc 的频率, T1_SEL = 0 时, 为 Fosc/12 (默认)。
5	T2_SEL	用于选择计数器/定时器 2 作为定时器使用时的计数速率, T2_SEL = 1 时, 为 Fosc 的频率, T2_SEL = 0 时, 为 Fosc/12 (默认)。
4	T3_SEL	用于选择计数器/定时器 3 作为定时器使用时的计数速率, T3_SEL = 1 时, 为 Fosc 的频率, T3_SEL = 0 时, 为 Fosc/12 (默认)。
3~0		

2.3.2 扩展串行口中断 UART1、UART2、UART3、UART4

PL5010 提供了 5 个串口。其扩展串口中断与原有的串行口中断的使用方式一样。其中断入口地址为: 0033H (UART1), 004BH (UART2), 005BH (UART3), 0063H (UART4)。其中 UART0、UART1、UART2 分别由定时器 T1、T2、T3 产生波特率; UART3、UART4 为简化的串口, 不需要定时器产生波特率, 通过相应的寄存器对 6M 的时钟分频产生波特率, 其停止位为 2 位。串口相应的控制寄存器如下:

● IE 中断使能寄存器 (S: A8H)

BIT	7	6	5	4	3	2	1	0
FIELD	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EA	中断允许总控制位 EA=0, 屏蔽所有中断申请; EA=1, CPU 开放中断。
6	ES1	串口 1 中断允许位 1: ENABLE 0:DISABLE
5	ET2	定时/计数器 T2 的溢出中断允许位 1: ENABLE 0:DISABLE
4	ES0	串口 0 中断允许位 1: ENABLE 0:DISABLE
3	ET1	定时/计数器 T1 的溢出中断允许位 1: ENABLE 0:DISABLE
2	EX1	外部中断 1 中断允许位 1: ENABLE 0:DISABLE
1	ET0	定时/计数器 T0 的溢出中断允许位 1: ENABLE 0:DISABLE
0	EX0	外部中断 0 中断允许位 1: ENABLE 0:DISABLE

● EIE 扩展中断使能控制寄存器 (S: A9H)

BIT	7	6	5	4	3	2	1	0
FIELD	EDS	ESPI	ES4	ES3	EX3	ES2	ET3	EX2
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EDS	差分串行通信单元中断允许位 1: ENABLE 0:DISABLE
6	ESPI	SPI 中断允许位 1: ENABLE 0:DISABLE
5	ES4	串口 4 中断允许位 1: ENABLE 0:DISABLE
4	ES3	串口 3 中断允许位 1: ENABLE 0:DISABLE
3	EX3	外部中断 3 中断允许位 1: ENABLE 0:DISABLE
2	ES2	串口 2 中断允许位 1: ENABLE 0:DISABLE
1	ET3	定时/计数器 T3 的溢出中断允许位 1: ENABLE 0:DISABLE
0	EX2	外部中断 2 中断允许位 1: ENABLE 0:DISABLE

● IP 扩展中断优先级控制寄存器 (S: B8H)

BIT	7	6	5	4	3	2	1	0
FIELD		PS1	PT2	PS0	PT1	PX1	PT0	PX0
RESET	—	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7		
6	PS1	串行口 1 中断优先级控制位
5	PT2	定时器/计数器 T2 中断优先级控制位

4	PS0	串行口 0 中断优先级控制位
3	PT1	定时器/计数器 T1 中断优先级控制位
2	PX1	外部中断 1 中断优先级控制位
1	PT0	定时器/计数器 T0 中断优先级控制位
0	PX0	外部中断 0 中断优先级控制位

● EIP 扩展中断优先级控制寄存器 (S:B9H)

BIT	7	6	5	4	3	2	1	0
FIELD	PDS	PSPI	PS4	PS3	PX3	PS2	PT3	PX2
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	PDS	差分串行通信单元中断优先级控制位
6	PSPI	SPI 中断优先级控制位
5	PS4	串行口 4 中断优先级控制位
4	PS3	串行口 3 中断优先级控制位
3	PX3	外部中断 3 中断优先级控制位
2	PS2	串行口 2 中断优先级控制位
1	PT3	定时器/计数器 T3 中断优先级控制位
0	PX2	外部中断 2 中断优先级控制位

● SCON1 串行口控制寄存器 1 (S:C0H)

BIT	7	6	5	4	3	2	1	0
FIELD	E1SM0	E1SM1	E1SM2	E1REN	E1TB8	E1RB8	E1TI	E1RI
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	ExSM0	串行口工作方式选择
6	ExSM1	
5	ExSM2	多机通讯控制位 主要用于方式 2 和方式 3。若置 ExSM2=1，则允许串口多机通讯，若 ExSM2=0，则不属于多机通讯情况。
4	ExREN	允许接收控制位 由软件置 1 或清 0，只有当 ExREN=1，串口才允许接收，ExREN=0 时，串口禁止接收数据。
3	ExTB8	发送数据的第 9 位 (D8) 装入 ExTB8 中在方式 2 或方式 3 中，根据

		发送数据的需要由软件置位或清 0，在方式 0 或方式 1 中，该位没用。
2	ExRB8	接收数据的第 9 位，在方式 2 或方式 3 中，接收到的第 9 位数据放在 ExRB8 中，在方式 1 中，如 ESM2=0，ExRB8 中存放的是接收到的停止位，方式 0 中未用该位。
1	ExTI	发送中断标志位 在串口将一帧数据发送完毕后被置 1，串口发送中断被响应后，ExTI 不会自动清 0，必须软件清 0。
0	ExRI	接收中断标志位 在串口接收到一帧数据后由硬件置 1，在串口接收中断被响应后，ExRI 不会自动清 0，必须软件清 0。

ESM0/ESM1：串行口工作方式选择。对应关系如下：

ExSM0	ExSM1	工作方式	说明	波特率
0	0	方式 0	同步移位寄存器	Fosc/12
0	1	方式 1	10 位异步收发	由定时器控制
1	0	方式 2	11 位异步收发	Fosc/32 或发 Fosc/64
1	1	方式 3	11 位异步收发	由定时器控制

● SCON2 串行口控制寄存器 2 (S:F8H)

BIT	7	6	5	4	3	2	1	0
FIELD	E2SM0	E2SM1	E2SM2	E2REN	E2TB8	E2RB8	E2TI	E2RI
RESET	0	0	0	0	0	0	0	0

SCON0 为串口 0 的控制寄存器，SCON1 为串口 1 的控制寄存器，SCON2 为串口 2 的控制寄存器。三个寄存器的对应位除了作用对象不一样外，功能及操作方式都一样，因此各位定义参考 SCON1。

UART3、UART4 相关寄存器为扩展寄存器，需要通过 EAT_ADR 及 EXT_DAT 来读写。

● EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

注：本寄存器为地址指针，中断时需入栈保护。

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

● UART3_CON 串行口控制寄存器 (EXT_ADR = BAH)

BIT	7	6	5	4	3	2	1	0
FIELD	ENA	REC_ENA	MOD	TB8				
RESET	0	0	0	0	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	ENA	使能位
6	REC_ENA	允许接收控制位 由软件置 1 或清 0，只有当 REC_ENA=1，串口才允许接收，REC_ENA=0 时，串口禁止接收数据
5	MOD	串行口工作方式选择 0 为 8 位 1 为 9 位
4	TB8	发送数据的第 9 位 (D8) 装入 TB8 中在方式 1 中，根据发送数据的需要由软件置位或清 0，在方式 0 中，该位没用
3~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● UART3_STATUS 串行口控制寄存器 (EXT_ADR = BBH)

BIT	7	6	5	4	3	2	1	0
FIELD	TI3	RI3		RB8				
RESET	0	0	—	0	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	TI3	发送中断标志位
6		
5	RI3	接收中断标志位
4	RB8	接收数据的第 9 位，在方式 1 中，接收到的第 9 位数据放在 RB8 中，方式 0 中未用该位
3~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● UART3_DVI_H 分频控制寄存器 (EXT_ADR = BCH)

BIT	7	6	5	4	3	2	1	0
FIELD			D13	D12	D11	D11	D9	D8
RESET	—	—	0	0	0	0	0	0

分频寄存器高 6 位。

● UART3_DVI_L 分频控制寄存器 (EXT_ADR = BDH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

分频寄存器低 8 位。

此 14 位寄存器用来对 6M 时钟分频来产生串口 3 的波特率。

● UART4_CON 串行口控制寄存器 (EXT_ADR = BFH)

BIT	7	6	5	4	3	2	1	0
FIELD	ENA	REC_ENA	MOD	TB8				
RESET	0	0	0	0	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	ENA	使能位
6	REC_ENA	允许接收控制位 由软件置 1 或清 0，只有当 REC_ENA=1，串口才允许接收，REC_ENA=0 时，串口禁止接收数据
5	MOD	串行口工作方式选择 0 为 8 位，1 为 9 位
4	TB8	在方式 1 中，发送数据的第 9 位 (D8) 装入 TB8 中，根据发送数据的需要由软件置位或清 0；在方式 0 中，该位没用。
3~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写

● UART4_STATUS 串行口控制寄存器 (EXT_ADR = C0H)

BIT	7	6	5	4	3	2	1	0
FIELD	TI4	RI4		RB8				
RESET	0	0	—	0	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	TI4	发送中断标志位
6		
5	RI4	接收中断标志位
4	RB8	在方式 1 中，接收到的第 9 位数据放在 RB8 中；在方式 0 中，该位没用。
3~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● UART4_DVI_H 串行口控制寄存器 (EXT_ADR = C1H)

BIT	7	6	5	4	3	2	1	0
FIELD			D13	D12	D11	D11	D9	D8
RESET	—	—	0	0	0	0	0	0

● UART4_DVI_L 串行口控制寄存器 (EXT_ADR = C2H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此组 14 位寄存器用来对 6M 时钟分频来产生串口 4 的波特率。

● SBUF1 扩展串口 1 据接收/发送缓冲器 (S:C1H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

● SBUF2 扩展串口 2 据接收/发送缓冲器 (S:F9H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

● SBUF3 扩展串口 2 据接收/发送缓冲器 (EXT_ADR = BEH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● SBUF4 扩展串口 2 接收/发送缓冲器 (EXT_ADR = C3H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

2.3.3 外部中断 INT2、INT3

用作普通的外部中断，INT2、INT3 与其他两个外部中断的用法一样，其中断入口地址为：003BH (INT2)，0053H (INT3)。相应的控制器/位如下：

● EIE 扩展中断使能控制寄存器 (S: A9H)

BIT	7	6	5	4	3	2	1	0
FIELD					EX3			EX2
RESET	—	—	—	—	0	—	—	0

Bit Number	Bit Mnemonic	Function
7~4		
3	EX3	EX3=1: 允许 INT3 中断 EX3=0: 不允许 INT3 中断
2~1		
0	EX2	EX2=1: 允许 INT2 中断 EX2=0: 不允许 INT2 中断

● EIP 扩展中断优先级控制寄存器 (S:B9H)

BIT	7	6	5	4	3	2	1	0
FIELD					PX3			PX2
RESET	—	—	—	—	0	—	—	0

Bit Number	Bit Mnemonic	Function
7~4		
3	PX3	PX3=1: INT3 中断优先级高 PX3=0: INT3 中断优先级低

2~1		
0	PX2	PX2=1: INT2 中断优先级高 PX2=0: INT2 中断优先级低

● TCON1 定时/计数器控制寄存器 1 (S:C8H)

BIT	7	6	5	4	3	2	1	0
FIELD	TF3	TR3	TF2	TR2	IE3	IT3	IE2	IT2
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	TF3	T3 计数溢出标志位 当 T2 计数溢出时，该位由硬件置 1，申请中断，进入中断后自动清零，也可软件清零。
6	TR3	T3 计数运行控制位 由软件置 1 或清 0。
5	TF2	T2 计数溢出标志位 当 T2 计数溢出时，该位由硬件置 1，申请中断，进入中断后自动清零，也可软件清零。
4	TR2	T2 计数运行控制位 由软件置 1 或清 0。
3	IE3	外部中断 3 的申请标志 当检测到有外部中断信号时，由硬件自动置 1。
2	IT3	外部中断 3 触发方式控制位 当 IT3=1 时外部中断 3 为边缘触发方式，当 IT3=0 是外部中断 3 为电平触发方式。
1	IE2	外部中断 2 的申请标志 当检测到有外部中断信号时，由硬件自动置 1。
0	IT2	外部中断 2 触发方式控制位 当 IT2=1 时外部中断 2 为边缘触发方式，当 IT2=0 是外部中断 2 为电平触发方式。

注：外部中断 INT1、INT2 在作为载波中断使用时不再作为外部中断使用。

2.4 IR38kHz 红外调制功能

2.7.1 功能简述

PL5010 内部集成有 38kHz 红外线调制通信电路，UART2 的串行数据输出可调制后从 P0.1 输出。软件可以根据不同主时钟频率配置载波频率的分频比，使用灵活方便。

2.7.2 编程指南

PL5010 内部有一个时钟分频得到的 38kHz 红外调制振荡波（频率可调），可以配合 UART2 的 TX 管脚输出，由 IR38K（EXT_CFG.1）位控制。若 IR38K = 0，则 TX2 管脚输出的是原始的 UART2 信号，若 IR38K = 1，则 TX2 管脚输出的是经过 38kHz 振荡波调制过的 UART2 信号。调制信号与调制前信号的对应示意图如图：

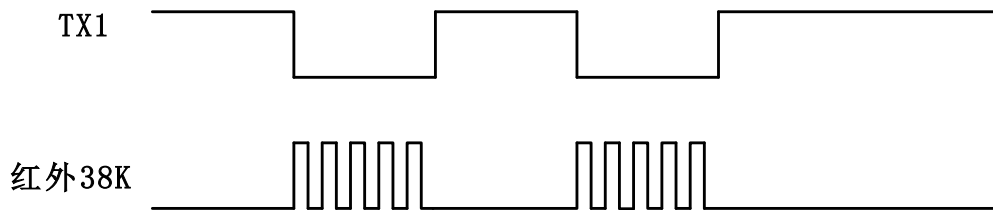


图 11 调制信号与调制前信号的对应示意图

PL5010 的红外通讯载频还可以灵活调节，用户可以通过对特殊寄存器 IR_CNT 设置来调节振荡波的频率。

1. 使能红外通信模式：置 IR38K（EXT_CFG.1）= 1。
2. 配置红外载频：IR_CNT 寄存器写入红外载波分频值。
3. 配置 UART2：如波特率、方式、中断等。

2.7.3 寄存器

● IR_CNT 红外振荡频率控制寄存器（S:DBH）

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

通过修改该寄存器的值可以配置不同的红外载波频率。其计算公式如下：

$f = (F_{osc}/(X+32))/2$ ，当采用 38kHz 载频时，若 $F_{osc} = 6\text{MHz}$ 时，应向 IR_CNT 寄存器写入 $X = 2\text{FH}$ 即可。

● EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

红外调制功能使能位于扩展寄存器内，配置时需通过扩展寄存器 EXT_DAT 和 EXT_ADR 对 EXT_CFG 进行配置。EXT_DAT 存入配置数据，EXT_ADR 存入配置地址。

● EXT_CFG 外部扩展寄存器 (EXT_ADR = FEH)

BIT	7	6	5	4	3	2	1	0
FIELD	TCK_ENA					FGEN_ENA	IR38K_ENA	PWM_ENA
RESET	0	—	—	—	—	0	0	0

Bit Number	Bit Mnemonic	Function
7	TCK_ENA	1Hz 输出使能 1, P26 输出 1Hz; 0, P26 不输出, 当作普通 I/O 口用。
6~3		
2	FGEN_ENA	频率发生器使能位 1, 使能频率发生器功能; 0, 不使能频率发生器功能。
1	IR38K_ENA	红外通讯使能 IR38K=1 开, IR38K=0 关。
0	PWM_ENA	脉宽调制使能位 高有效。

2.5 看门狗定时器

为了防止程序因为意外原因导致死机，专门设计了一套看门狗电路，当程序死机后一定时间内（由寄存器控制，可选为 87ms；174ms；……696ms）可以重新复位 8051。该电路是由一个计时器（WDTimer）来完成。（注意：看门狗功能在芯片上电复位后自动工作且无法关闭）

● WDT_CON 看门狗控制寄存器（S:8EH）

BIT	7	6	5	4	3	2	1	0
FIELD	RST1	RST0	WDT2	WDT1	WDT0			
RESET	0	0	0	0	0	—	—	—

Bit Number	Bit Mnemonic	Function
7	RST1	复位时间长度控制位
6	RST0	
5	WDT2	控制看门狗喂狗时间长度 当 WDT2/WDT1/1DT0 = 000 时，看门狗复位间隔时间为 87 毫秒，当 WDT2/WDT1/1DT0 = 001 时，看门狗复位间隔时间为原来的 2 倍，即 174 毫秒。依此类推，当 WDT2/WDT1/1DT0 = 111 时，看门狗复位间隔时间为 87×8 = 696 毫秒。WDT2/WDT1/1DT0 默认为 0。
4	WDT1	
3	WDT0	
2~0		

RST1 RST0	0 0	0 1	1 0	1 1
复位时间	88ms	176ms	264ms	352ms

● WDT_RST 看门狗复位寄存器（S:8FH）

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

当向该寄存器写入 A1H 时，看门狗定时器（WDTimer）自动复位为 7FFFH，该计数器以 $F_{osc}/(32*(1+WDT[2:0]))$ 的速度递减，当递减到零溢出时，就会产生一个内部复位信号，强制系统重新启动。软件正常运行时，应该定期向该寄存器中写入 A1H，复位看门狗定时器，防止产生不需要的复位。

3 电能计量模块

3.1 电能计量工作原理

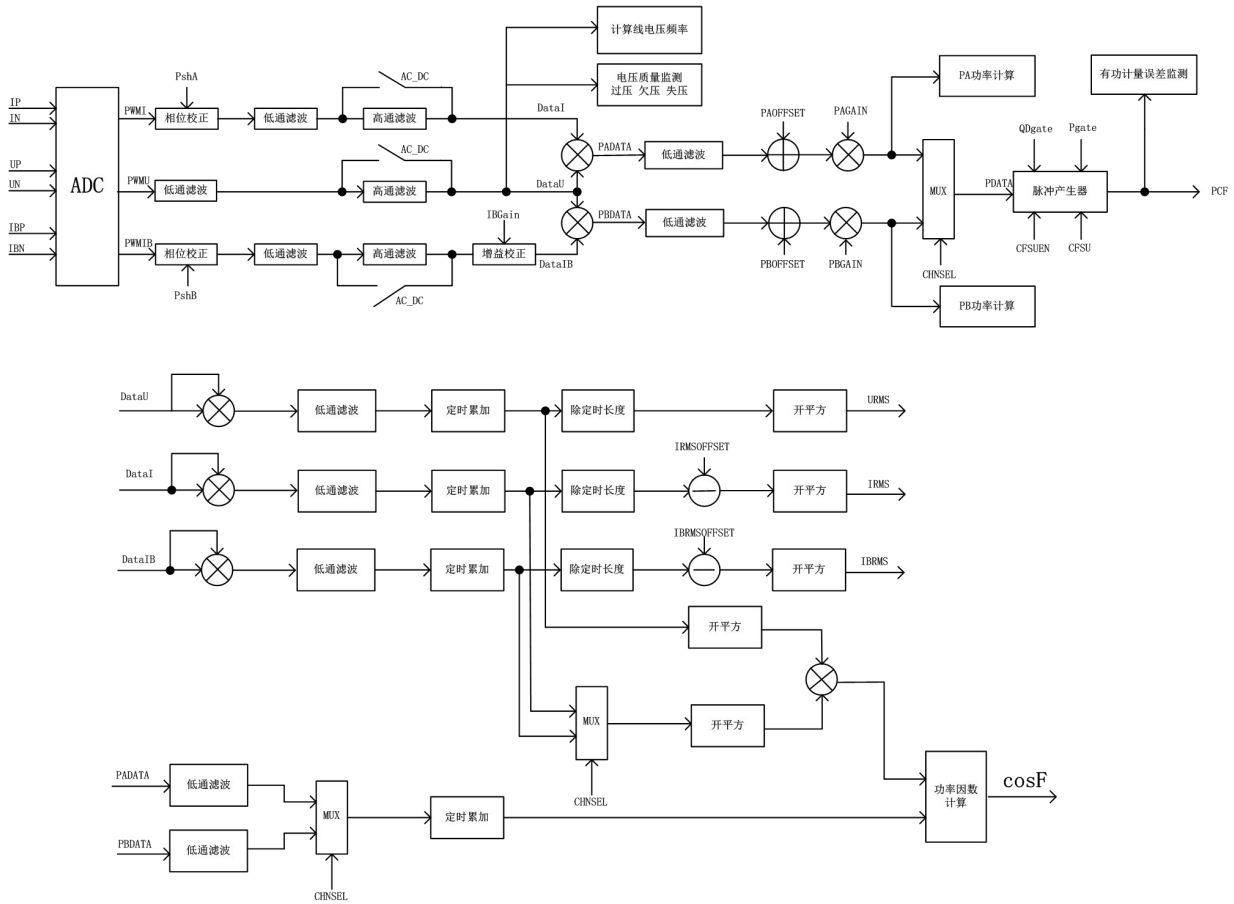


图 12 电能计量模块内部的原理框图

名词解释：TFC：温度/频率转换器

Freq Counter：频率计数器

PGA：程控增益放大器

DPC：数字相位校正

LPF：低通滤波器

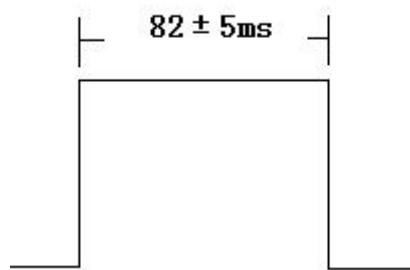
HPF：高通滤波器

DFC：数字/频率转换器

首先，实现对电流、电压的采样；然后，经过放大电路放大，通过 A/D 转换器转换为数字信号；接下来经过数字滤波、移相以及相乘、积分等处理；最终，得到有效的电能计量数据。从而，完成对有功/无功功率、瞬时电流和电压的有效值以及当前的交流频率值的计量、计算。

计量电路可以由软件来控制选取两路电流采样信号中的某一路。在计量模块内部与电压采样结果进行乘法处理，计算得到有功功率、无功功率。上电复位时默认为第一路 ADC 采样作为电流通道的输入。

3.1.1 计量输出设置



有功电能 PCF 如上图所示，符合国家电网计量规范。

校表方式：软件模式，通过 8051 将校表数据写入到校表专用的寄存器中。

有功校表：调整 Pgate_HB、Pgate_LB、Poff_HB、Poff_LB、PGAIN 寄存器的值。

3.1.2 计量部分与 8051 数据交换

计量模块与 8051 之间是并行通讯，通过寄存器以及中断进行控制。PL5010 内部的嵌入式微处理器 8051 可以通过特殊功能寄存器 (SFR) 地址直接访问计量模块内部的寄存器。

电能计量单元的寄存器主要分为控制寄存器、标识寄存器、校表寄存器和数据寄存器四类。

- 控制寄存器：控制计量部分的工作模式

防窃电工作方式：上电初始化为 0，选择电流通路 2。

有功脉冲计数模式选择：上电初始化为 0，绝对相加。

直流和交流电电能计量模式选择：上电初始化为 0，交流模式。

电能计量使能控制：上电初始化为 1，使能有效。

有功功率计量中断/查询模式使能控制：上电初始化为 0，查询模式。

ADC 增益选择 (I1_G1, I1_G0) :

00: 4 倍增益

01: 8 倍增益

10: 16 倍增益

11: 32 倍增益

ADC 增益选择 (I2_G1, I2_G0) :

00: 4 倍增益

01: 8 倍增益

10: 16 倍增益

11: 32 倍增益

● 校表寄存器：8051 将校表数据写入其中对电能计量进行校准

有功计量门限：16 位无符号数，上电初始化为 5fff。

有功线性补偿：16 位有符号数，上电初始化为 0000。

● 标识寄存器：产生相关计量标识

A/D 转换结束标识：1，转换中；0，转换结束。

有功功率计量脉冲标识：1，有效。

有功功率正负标识：1，负功；0，正功。

电流电压采样转换结束标识：1，转换中；0，转换结束。

电压过零点标识：1，过零点；0，非过零点。

电压过零点方向标识：1，正向过零点；0，负向过零点。

● 数据寄存器：传输计量数据到 8051

电压采样转换值：16 位有符号数。

电流采样转换值：16 位有符号数。

有功计量脉冲计数值：8 位无符号数。

电压计量值：16 位无符号数。

电流计量值：16 位无符号数。

电压/电流计量采样周波数：16 位无符号数。

交流频率计量值：16 位无符号数。

3.1.3 电能计量工作工程

1. 系统上电进入电能计量默认工作状态。
2. 通过向写保护寄存器写入控制数据来取消写保护。
3. 通过向控制寄存器写入数据来控制电能计量使能和工作模式。
4. 通过向校表寄存器写入数据来调整电能计量精度。
5. 通过向写保护寄存器写入控制数据来使能写保护。
6. 读取标识寄存器、数据寄存器的值，查询计量结果。

注意：只有通过对 FBH 地址的只读寄存器进行写操作才能启动平均电压、平均电流的测量（平均电压、平均电流是同时进行测量的），数据的转换时间需要 300ms。即完成对 FBH 的写操作后 300ms 后，可以查询 RMS_OV（电压电流采样转换结束）标志若为高时，表明转换结束，可以读取平均电压计量值、平均电流计量值和采样周波数来用以计算平均电压和平均电流的有效值。（RMS_OV 标志在下一次对 FBH 地址进行写操作启动平均电压电流测量时，由硬件自动清零。）电压、电流有效值的计算详见电压、电流计量寄存器说明。

3.1.4 模拟信号输入范围

电流通道模拟信号输入范围：

4 倍模拟增益：4X \pm 375mV

8 倍模拟增益：8X \pm 187.5mV

16 倍模拟增益：16X \pm 93.75mV

32 倍模拟增益：32X \pm 46.875mV

电压通道模拟信号输入范围： \pm 375mV

3.2 寄存器

● PMU_ADR 电能计量扩展地址寄存器（S:E9H）

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0

RESET	0	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---	---

● PMU_DAT 电能计量扩展数据寄存器 (S:E8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

电能计量单元寄存器需从过此两组寄存器扩展进行访问。

● FREQ_LB 电压频率寄存器 (PMU_ADR = 00H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● FREQ_HB 电压频率寄存器 (PMU_ADR = 01H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

频率值是一个无符号数，参数公式如下：

$$F = 312500/\text{Freq}(\text{Hz})$$

电压频率更新周期为 1s。

● IArms 电流有效值寄存器 2 字节 (由高到低) (PMU_ADR = 03H 02H)

BIT	15	14	1	0
RESET	0	0	0	0

在启动电压、电流量 300ms 后，通过查询 RMS_OV 标志为高后，用户便可以在该寄存器中读出一组 16BIT 的数据，利用公式可算出真实电流值。

● Vrms 电压有效值寄存器 2 字节 (由高到低) (PMU_ADR = 05H 04H)

BIT	15	14	1	0
RESET	0	0	0	0

在启动电压、电流量 300ms 后，通过查询 RMS_OV 标志为高后，用户便可以在该寄存器中读出一组 16BIT 的数据，利用公式可算出真实电压值。

● IBrms 零线电流有效值寄存器 2 字节 (由高到低) (PMU_ADR = 07H 06H)

BIT	15	14	1	0
RESET	0	0	0	0

在启动电压、电流计量 300ms 后，通过查询 RMS_OV 标志为高后，用户便可以在该寄存器中读出一组 16BIT 的数据，利用公式可算出真实电流值。

● IAAD_HB 实时电流寄存器高字节 (PMU_ADR = 09H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

是实时电流采样数据滤波后的值，该寄存器可以作为数字校表用，通过该寄存器的数字可以估算出系统的电流测量范围。

● IAAD_LB 实时电流寄存器低字节 (PMU_ADR = 08H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● UAD_HB (PMU_ADR = 0BH) 实时电压寄存器高字节 (S:0BH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● UAD_LB 实时电压寄存器低字节 (PMU_ADR = 0AH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

是实时电压采样数据滤波后的值，该寄存器可以作为数字校表用，通过该寄存器的数字可以估算出系统的电压测量范围。

● IBAD_HB 实时电流寄存器高字节 (PMU_ADR = 0DH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

是实时电流采样数据滤波后的值，该寄存器可以作为数字校表用，通过该寄存器的数字可以估算出系统的电流测量范围。

● IBAD_LB 实时电流寄存器低字节 (PMU_ADR = 0CH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● COSPHI_LB 功率因数寄存器低字节 (PMU_ADR = 0EH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● COSPHI_HB 功率因数寄存器高字节 (PMU_ADR = 0FH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

功率因数寄存器是一个有符号数，最高位为符号位，用于表示有功功率和视在功率的比值，计算公式如下：

$$\cos F = \text{Cosphi} / 2^{15}$$

对 Status (FBH) 寄存器进行写操作（写入任何数据均可），将启动功率因数计算，等待 300ms，查询 Status.7 (RSM_OV) 标志若为高，可以读取数据值。

● IPA1 有功功率寄存器 (PMU_ADR = 10H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IPA2 有功功率寄存器 (PMU_ADR = 11H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IPA3 有功功率寄存器 (PMU_ADR = 12H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IPA4 有功功率寄存器 (PMU_ADR = 13H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

有功功率寄存器是一个有符号数，最高位为符号位，功率更新频率为 1s。

● IPB1 有功功率寄存器 (PMU_ADR = 14H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IPB2 有功功率寄存器 (PMU_ADR = 15H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IPB3 有功功率寄存器 (PMU_ADR = 16H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IPB4 有功功率寄存器 (PMU_ADR = 17H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

有功功率寄存器是一个有符号数，最高位为符号位，功率更新频率为 1s。

● PCNT 有功计量脉冲计数寄存器 (PMU_ADR = 18H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

系统在计量有功时，除了向外部提供用于计量的计量脉冲 PCF 外，在系统内部也在对该计量脉冲进行累加，该寄存器为只读，计满 FF 后清零。

● Poff_A_HB 有功线性补偿寄存器高字节 (PMU_ADR = 1AH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● Poff_A_LB 有功线性补偿寄存器低字节 (PMU_ADR = 19H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

双字节的有功线性补偿值，有符号数，在-32768~+32767 范围内可调，其作用是通过每一次电压电流乘积在积分前的瞬时值进行微小的正负修正，更精确地产生校表脉冲 PCF，完成了校表的微调过程（其主要用途是抵消掉由板级噪声干扰或采样非线性引起的误差）。

● Poff_B_HB 有功线性补偿寄存器高字节（PMU_ADR = 1CH）

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● Poff_B_LB 有功线性补偿寄存器低字节（PMU_ADR = 1BH）

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

双字节的有功线性补偿值，有符号数，在-32768~+32767 范围内可调，其作用是通过每一次电压电流乘积在积分前的瞬时值进行微小的正负修正，更精确地产生校表脉冲 PCF，完成了校表的微调过程（其主要用途是抵消掉由板级噪声干扰或采样非线性引起的误差）。

● PAGAIN_HB 功率增益 A 寄存器（PMU_ADR = 1EH）

BIT	7	6	5	4	3	2	1	0
RESET	1	0	0	0	0	0	0	0

● PAGAIN_LB 功率增益 A 寄存器（PMU_ADR = 1DH）

BIT	7	6	5	4	3	2	1	0
RESET	1	0	0	0	0	0	0	0

● PBGAIN_HB 功率增益 B 寄存器（PMU_ADR = 20H）

BIT	7	6	5	4	3	2	1	0
RESET	1	0	0	0	0	0	0	0

● PBGAIN_LB 功率增益 B 寄存器（PMU_ADR = 1FH）

BIT	7	6	5	4	3	2	1	0
RESET	1	0	0	0	0	0	0	0

16bit A 相 B 相功率增益，0~2 倍增益。

● Pgate_HB 有功门限寄存器高字节 (PMU_ADR = 22H)

BIT	7	6	5	4	3	2	1	0
RESET	0	1	0	1	1	1	1	1

● Pgate_LB 有功门限寄存器低字节 (PMU_ADR = 21H)

BIT	7	6	5	4	3	2	1	0
RESET	1	1	1	1	1	1	1	1

双字节的有功电能计量门限，无符号数，在 0~65535 范围内可调，其作用是为了调节电能计量精度。原理如下：当电能累加量超过该门限，即产生校表脉冲 PCF。显然，该门限越小，校表脉冲 PCF 的出现频率越高，反之，校表脉冲 PCF 的出现频率越低。通过不断调整该门限，直到 PCF 脉冲频率符合实际需要为止，即完成了校表的粗调过程。

● QDGATE 潜动门限寄存器 (PMU_ADR = 23H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

潜动门限寄存器是一个无符号数，其规定了脉冲输出的最大间隔，其计算公式如下：

$$\text{MaxTCF} = \text{QDGATE} * 216/5000(\text{s})$$

当脉冲输出间隔大于此值时，脉冲输出被屏蔽。QDgate = 0 时，禁止防潜动功能。

● IBGAIN_HB 电流增益寄存器 (PMU_ADR = 25H)

BIT	7	6	5	4	3	2	1	0
RESET	1	0	0	0	0	0	0	0

● IBGAIN_LB 电流增益寄存器 (PMU_ADR = 24H)

BIT	7	6	5	4	3	2	1	0
RESET	1	0	0	0	0	0	0	0

B 相电流增益，只针对电流有效值，不参与计量运算，范围 0~2 倍。

● IADPC 相位校正寄存器 (PMU_ADR = 26H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

DPC[7-0]: 相位调整量控制位, 绝对值越大移动越多。

DPC[7-0] = 0x7f: 前移 3.84° (600K), 2.88° (800K)。

DPC[7-0] = 0xff: 后移 3.84° (600K), 2.88° (800K)。

● IBDPC 相位校正寄存器 (PMU_ADR = 27H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

DPC[7-0]: 相位调整量控制位, 绝对值越大移动越多。

DPC[7-0] = 0x7f: 前移 3.84° (600K), 2.88° (800K)。

DPC[7-0] = 0xff: 后移 3.84° (600K), 2.88° (800K)。

● IA_DPC_H 相位校正寄存器 (PMU_ADR = 28H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IB_DPC_H 相位校正寄存器 (PMU_ADR = 29H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● CON_GAIN 截位寄存器 (PMU_ADR = 2AH)

BIT	7	6	5	4	3	2	1	0
RESET			IA_G		IB_G		U_G	

Bit Number	Bit Mnemonic	Function
7~6		
5	IA_G	A 电流通道截位

4		
3	IB_G	B 电流通道截位
2		
1	U_G	电压通道截位
0		

● OV_GATE 过压门限寄存器 (PMU_ADR = 2BH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

用于设定过压门限寄存器。

● SAG_GATE 相位校正寄存器 (PMU_ADR = 2CH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

用于设定欠压门限寄存器值。

● SAG_P 欠压周期数寄存器 (PMU_ADR = 2DH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

用于设定电压连续欠压半波周期数，达到 SAG_P 数后，将给出欠压标志。

● VPEAK 电压峰值寄存器 (PMU_ADR = 2EH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

VPEAK 寄存器记录一个周波内最大的电压 ADC 输出波形绝对值的最大值。

● VQM 电压质量监测寄存器 (PMU_ADR = 2FH)

BIT	7	6	5	4	3	2	1	0
FIELD	VPEAK_ENA					OV_FLAG	SAG_FLAG	V_FAIL
RESET	0	—	—	—	—	0	0	0

Bit Number	Bit Mnemonic	Function
7	VPEAK_ENA	电压峰值使能位 高有效，启动电压峰值计算后，VPEAK_ENA 保持高电平，硬件计算完峰值后，VPEAK_ENA 被置为低电平。R/W
6~3		
2	OV_FLAG	过压标志 高有效，当电压峰值在半个周波内大于 OVgate 值时，OV_FLAG 标志将被置高，硬件自动清零。R
1	SAG_FLAG	欠压标识 高有效，当电压峰值连续 SAG_P 个半周波内低于 SAGgate 值，SAG_FLAG 标志将被置高，必须由软件清零。R
0	V_FAIL	失压标志 25ms 内没有检测到过零点信号，V_FAIL 标志将被置高，出现过零点标志后，V_FAIL 将被置低。R

● IARMS_OFF_HB 电流有效值校正寄存器 (PMU_ADR = 37H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IARMS_OFF_LB 电流有效值校正寄存器 (PMU_ADR = 36H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IBRMS_OFF_HB 电流有效值校正寄存器 (PMU_ADR = 39H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

● IBRMS_OFF_LB 电流有效值校正寄存器 (PMU_ADR = 38H)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

16bit 校正寄存器，补偿系统误差对电流有效值的影响。

● CFCONFIGCFCONFIG 寄存器 (PMU_ADR = FAH)

BIT	7	6	5	4	3	2	1	0
FIELD	CFSUEN	CFSU[1:0]						
RESET	0	0		—	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	CFSUEN	加速使能
6、5	CFSU[1:0]	加速配置 00: 2 倍 01: 4 倍 其他: 8 倍
4~0		

● Write Protect Register 写保护寄存器 (PMU_ADR = FFH)

BIT	7	6	5	4	3	2	1	0
RESET	0	0	0	0	0	0	0	0

向此寄存器写入数据 FFH 时取消写保护（即允许对所有可写寄存器进行写操作）；否则使能写保护（即所有寄存器均为只读、不可写）。

● Config 电能计量控制寄存器 (PMU_ADR = FCH)

BIT	7	6	5	4	3	2	1	0
FIELD	ENA	AC_DC	PIE	PMOD	ENA2	CHNSEL	FS_SEL	
RESET	0	0	0	0	0	0	0	—

Bit Number	Bit Mnemonic	Function
7	ENA	控制 I1_channel 和 U_channel, 高有效。
6	AC_DC	直流/交流电能计量切换控制, 0 交流, 1 直流。
5	PIE	有功计量中断/查询切换控制, 0 查询, 1 中断。
4	PMOD	有功脉冲计数模式选择, 1 考虑正负, 0 不考虑正负。
3	ENA2	控制 I2_channel, 高有效。
2	CHNSEL	电流通道计量选择, 0 选择 I2, 1 选择 I1。
1	FS_SEL	ADC 采样频率选择, 0 选择 800K, 1 选择 600K。
0		

● Iagc_cfgADC 模拟增益选择控制寄存器 (PMU_ADR = FDH)

BIT	7	6	5	4	3	2	1	0
FIELD					I2_G1	I2_G0	I1_G1	I1_G0
RESET	—	—	—	—	0	0	1	1

Bit Number	Bit Mnemonic	Function
7~4		
3~2	I2_G1、I2_G0	ADC 增益选择 (Bit3, Bit2) : 00: 4 倍增益 01: 8 倍增益 10: 16 倍增益 11: 32 倍增益
1~0	I1_G1、I1_G0	ADC 增益选择 (Bit1, Bit0) : 00: 4 倍增益 01: 8 倍增益 10: 16 倍增益 11: 32 倍增益

● Status 标识寄存器 (PMU_ADR = FBH)

BIT	7	6	5	4	3	2	1	0
FIELD	RMS_OV	Z_FLAG	Z_DIR	LDE	PDIR	PCF	Multi_ovf	
RESET	0	0	0	0	0	0	0	—

Bit Number	Bit Mnemonic	Function
7	RMS_OV	平均电压电流采样转换结束标志 0, 数据转换中; 1, 转换结束。
6	Z_FLAG	电压过零点标志 1, 过零点; 0: 非过零点。
5	Z_DIR	电压过零点方向指示 1, 正向过零点; 0, 负向过零点。
4	LDE	A/D 转换结束标识 1, 数据转换过程中; 0 数据转换结束并生效。
3	PDIR	有功功率正负标识 1, 负功; 0, 正功。
2	PCF	有功功率计量脉冲标识

		1, 有效。
1	Multi_ovf	乘法器溢出标志 高有效, 表示当前相乘结果超出范围, 计算结果无效。
0		

3.3 计量公式

1: Poff 计算方法:

关电流, 上电压 (220V), 连续十次 (间隔不小于 0.3s) 读对应 IPA 或 IPB, 算出平均值, 再除以三的值, 写入 Poff.

2:Pgate 门限值校正:

若在对应通道 100%Ib、PF=1.0 上读出误差为 ERROR:

则, $P_GATE_N = P_GATE_{N-1} * (1 + ERROR)$, N 为校表次数;

即 P_GATE_0 为寄存器默认值 0x5FFF;

其中:

P_GATE_N 指本次要写入寄存器的校正值, P_GATE_{N-1} 指寄存器上次的校正值;

注意: 门限值需要多次校准, 直到 err 值达到要求;

3:DPC

A/B 通道相位校正寄存器的计算方法:

若标准表在 A/B 通道, 100%Ib, PF=0.5L 上读出误差为 err, 则相位补偿公式:

$$\theta = \text{Arc sin} \frac{-err}{\sqrt{3}}$$

对 50HZ, PHSA/B 有 0.02°/LSB 的关系, 则有

如果 $\theta \geq 0$, $DPC = INT(\theta / 0.02^\circ)$

如果 $\theta < 0$, $DPC = INT(2^8 + \theta / 0.02^\circ)$

4: 频率:

$$F = 312540/\text{Freq}$$

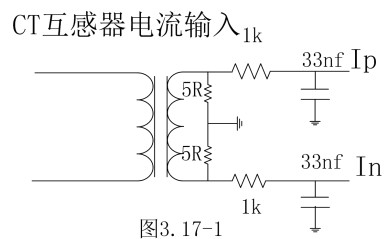
5: 电流，功率增益

$$\text{GAIN} = \text{gain} / 2^{15}$$

GAIN 为实际增益值 (0-2)，gain 为 16 位增益寄存器数据。

6: 电流:

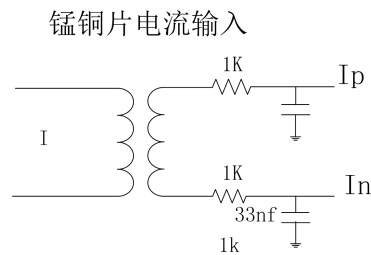
PL5010 根据外围电路和内部寄存器可以算出输入的电流电压值。电流通道分 CT 互感器和锰铜片输入，电压通道一般是使用电阻分压和 PT 互感器输入；



如图 3.17-1 所示的 CT 电流输入，电流真实值计算公式：

$$I(A) = I_{rms} * a / (L * \text{ADC_PGA} * \text{IGAIN} * R)$$

其中 I_{rms} 为有效值； $L=2609$ ； a 为电流转化率，如 (10A-5mA) 的 CT 转化率为 2000；ADC_PGA 为模拟通道增益 (4, 8, 16, 32)，参照外设寄存器模拟通道增益寄存器；IGAIN 为电流通道增益 (0 到 2)，参考电能计量寄存器的电流通道增益寄存器； R 为外围电路的电阻 (5R)。



如图 3.17-2 所示的锰铜片电流输入，电流真实值计算公式：

$$I(A) = I_{rms} / (L * \text{ADC_PGA} * \text{IGAIN} * R)$$

其中 I_{rms} 为有效值； $L=8800$ ；ADC_PGA 为模拟通道增益 (4, 8, 16, 32)，参照外设寄存

器模拟通道增益寄存器；IGAIN 为电流通道增益（0 到 2），参考电能计量寄存器的电流通
道增益寄存器；R 为锰铜片的阻值，单位（ Ω ）。

6: 电压:

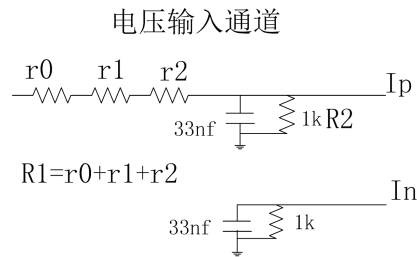


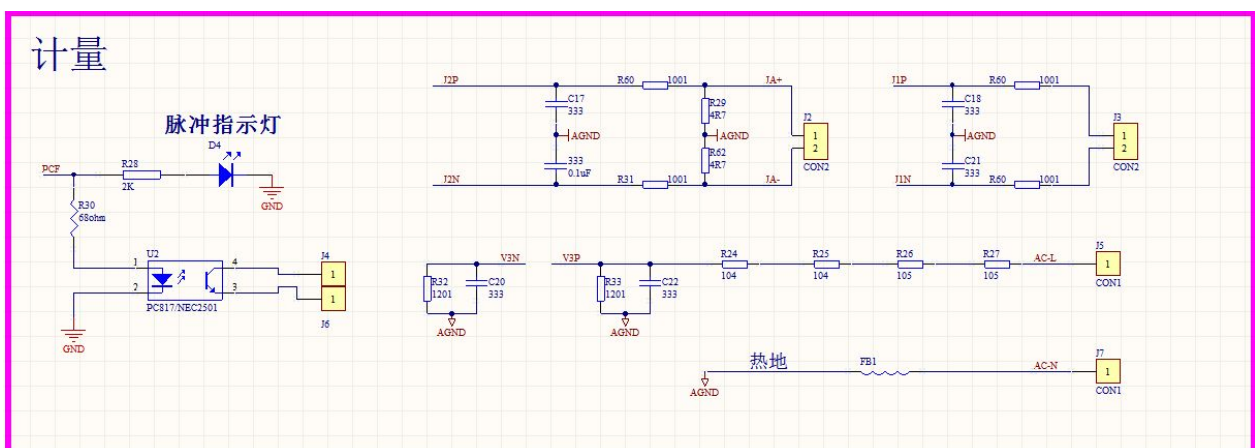
图3. 17-3

如图 3. 17-3 所示的分压电阻输入，电压真实值计算公式：

$$U(v) = V_{rms} * R1 / (L * R2)$$

其中 V_{rms} 为有效值； $L=113662$ ；参考电能计量寄存器的电流通道增益寄存器； $R2$ 为 1k 分压电阻， $R1$ 为 $R2$ 之前所有电阻之和。

3.4 参考电路



3.5 校表流程

一：参数设置

1: 配置 Config 寄存器，使能相应的电压，电流通道

2: 配置 lagc_cfg 寄存器，选择电流增益

二：POFF 校准

在电流输入为零的条件下（220V），读取有功功率寄存器的值，读不少于 10 次（间隔不小于 0.3s）取补码平均值，除以 3 写入 Poff 寄存器。

三：门限校准

100%I_b、PF=1 上读出误差为 err 利用公式算出门限值，写入 Pgate 寄存器；门限值需要多次校准，直到 err 值达到标准。

四：相位补偿校准

100%I_b、PF=0.5L 上读出误差为 err，利用公式算出相位补偿值，写入 DPC 寄存器

3.6 应用例程

假设设计一块 220v（U_n）、10A（I_b）额定输入、表常数为 3200（EC）的样表。A 通道电流采样使用 250 微欧的锰铜，通道 A 模拟通道增益为 32 倍；电压采用电阻分压输入。

1: 参数设置

打开写保护 FFH = 0xff;

配置 Config 寄存器，使能相应的电压，电流通道 Config: (FCH)=84H

配置 lagc_cfg 寄存器，选择电流增益 lagc_cfg(FDH)=03H

2: OFFSET 校准

在 A 电流输入为零的条件下，读 IPA1（10H），IPA2（11H），IPA3（12H），IPA4（13H）10 组数据：

00 00 10 00 00 00 10 00

00 00 03 00 00 00 0E 00

00 00 0F 00 00 00 15 00

00 00 11 00 00 00 11 00

00 00 0D 00 00 00 18 00

10 次取补码平均值，除以 3 为 00 00 09 00

Poff_A(1AH 19H) = 0900H

3: 门限校准

表台读取 Err= -0.7024 利用门限公式: $P_GATE_N = P_GATE_{N-1} * (1 + ERROR)$

P_GATE= 0x5fff* (1-0.7024) =0x1c91

写寄存器: Pgate_HB(22H)=1cH Pgate_LB(21H)=91H

读 err=-0.00050 符合要求，否则再次利用公式计算

4: 相位补偿校准

100%lb、PF=0.5L (30 度) 上读出误差为 err，利用公式算出相位补偿值，写入 DPC 寄存器

表台输入 0.5L，读取表台 err=0.00492，

利用 DPC $\theta = Arc \sin \frac{-err}{\sqrt{3}}$ 公式: $\theta=0.162$

DPC = INT($\theta/0.02^0$) 公式: IB DPC = 8

3.7 操作电能计量寄存器例程

```
sfr PMU_ADR = 0xe9;
sfr PMU_DAT = 0xe8;

void Write_PMU(unsigned char adr,unsigned char dat)
{
    PMU_ADR = adr;
    PMU_DAT = dat;
}

unsigned char Read_PMU(unsigned char adr)
{
    unsigned char Temp_Dat;
    PMU_ADR = adr;
    Temp_Dat = PMU_DAT;
    return Temp_Dat;
}
```

4 载波通信

4.1 功能简述

PL5010 内置的载波通信单元，采用 DBPSK 调制方式和直序扩频技术，载波中心频率为 120kHz，伪随机码速率为 7.5kHz，数据速率为 500bps。提供连续和过零点两种通信方式兼容与 PL4000 通信。

PL5010 提供了过零点通信方式，数据速率为 100bps，载波发送起始时刻可由软件灵活控制，能够有效避开信道上噪声最恶劣的时刻，提高通信质量，同时也有利于相位的甄别。

PL5010 采用按字节发送接收方式并能够自动发送和校验 16 位帧头。相比以往的载波按位的发送接收模式，减小了代码量，降低了编程的难度。

PL5010 采用了匹配滤波器算法实现解扩，相比于 PL3106/PL3107 的滑动相关解扩算法，发送端仅需要发送少量的同步信息位，即可实现收发端扩频码的同步，有助于信道利用率的提升。

PL5010 内置六阶高性能开关电容带通滤波器，无需外置带通滤波器，既降低了成本，又节省了 PCB 板的空间。

PL5010 载波通信单元由两套独立工作的收发电路实现连续工作和过零点通信方式相关资源如下表：

工作方式	中断源
连续方式	INT2
过零点方式	INT1

注：外部中断 INT1、INT2 在作为载波中断使用时不再作为外部中断使用。

4.2 编程指南

载波发送接收流程：

1. 设置预同步字节：向寄存器 SEND_LEADER_VALUE (Addr:A2H) 写入预同步唤醒位数，如：EXT_ADR = 0xA2 ; EXT_DAT = 0x14;

2. 设置同步帧头，如设置同步帧头为 0x09AF，则：EXT_ADR = 0xA4 ; EXT_DAT = 0x09 ; EXT_ADR = 0xA5 ; EXT_DAT = 0xAF ;

3. 设置同步帧头允错位数，一般情况下默认 0x00 即可，不宜太大，若要设置为 0x01，则：EXT_ADR = 0xA6 ; EXT_DAT = 0x01 ;

4. 设置载波接收门限，推荐寄存器分别设置为：0x50, 0x28，即：EXT_ADR = 0xAB; EXT_DAT = 0x50; EXT_ADR = 0xAC; EXT_DAT = 0x28;

5. 若为过零方式：使能载波通讯 CCS_ENA，使能载波过零中断 CCS_IE_ZC，置 ZC_ENA 为 1 设置为过零方式，此时 INT1 作为过零点方式下载波的同步通讯中断，不再作为外部中断 1 用，然后使能 INT1 中断（EX1 = 1），且为边沿触发方式（IT1 = 1）；

若为连续方式：使能载波通讯 CCS_ENA，使能连续中断 CCS_IE，将 ZC_ENA 清 0 配置为连续方式，此时 INT2 作为连续方式下载波的同步通讯中断，不再作为外部中断 2 用，然后使能 INT2 中断(EX2=1)，且为边沿触发方式(IT2=1)；

6. 若要改变载波过零点发送时刻，可以修改 ZC_OFFSET 过零点时刻偏移量寄存器的值；

7. 若发送数据：配置寄存器 COMMCONTROL_REGISTER，其中 CCS_ENA 位置 1, CCS_IE 位置 1, R/T 位清 0，置发送态，然后在中断中将待发数据写入 SCC_TXBUF 寄存器；

8. 当 INT1（连续方式为 INT2）有中断产生时，进入中断子程序：通过判断 R/T 标志位区分发送还是接收态。若是处于发送态，可将待发数据位送入 SCC_TXBUF 发送，若是处于接收态，读取 ZC_SYNC（连续方式为 SYNC）标志位状态，若为 1 则表示帧头同步成功；若为 0 则表示帧头同步错误；帧头同步成功后，可以从 SCC_RXBUF1（连续方式为 SSC_RXBUFF0）读取过零（连续）方式接收到的数据。

载波发送完毕后，自动转为接收态；

4.3 寄存器

● EXT_ADR 扩展地址寄存器（S:D9H）

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。注：本寄存器为地址指针，中断时需入栈保护。

● EXT_DAT 扩展数据寄存器（S:D8H）

BIT	7	6	5	4	3	2	1	0
------------	----------	----------	----------	----------	----------	----------	----------	----------

FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

● COMMSTATUS REGISTER0 外部控制寄存器 (EXT_ADR = A0H)

BIT	7	6	5	4	3	2	1	0
FIELD			FHF				CORR	SYNC
RESET	—	—	0	—	—	—	0	0

Bit Number	Bit Mnemonic	Function
7~2		RESERVE 保留位
1	CORR	连续模式下相关标志 CORR = 1, 相关峰大于捕获门限值; CORR = 0, 相关峰小于捕获门限值。
0	SYNC	连续模式下同步标志 SYNC = 1, 帧头同步成功; SYNC = 0, 帧头同步错误

● COMMSTATUS REGISTER1 外部控制寄存器 (EXT_ADR = A1H)

BIT	7	6	5	4	3	2	1	0
FIELD			FHF_zc			VFAIL	ZC_CORR	ZC_SYNC
RESET	—	—	0	—	—	0	0	0

Bit Number	Bit Mnemonic	Function
7~3		
2	VFAL	失压检测标志位 VFAIL = 1, 无过零点信号(不能进行过零点发送/接收); VFAIL = 0, 有过零点信号。电源频率大于 40HZ, VFAL 标志位清零。
1	ZC_CORR	过零点模式下相关标志 ZC_CORR = 1, 相关峰大于捕获门限值; ZC_CORR = 0, 相关峰小于捕获门限值。
0	ZC_SYNC	过零点模式下同步标志 ZC_SYNC = 1, 帧头同步成功; ZC_SYNC = 0, 帧头同步错误。

● SEND_LEADER_VALUE 外部控制寄存器 (EXT_ADR = A2H)

BIT	7	6	5	4	3	2	1	0
------------	---	---	---	---	---	---	---	---

FIELD		D6	D5	D4	D3	D2	D1	D0
RESET	—	0	0	0	0	0	0	0

预同步用的唤醒位可置 0~127 位。

● COMMCONTROL_REGISTER 外部控制寄存器 (EXT_ADR = A3H)

BIT	7	6	5	4	3	2	1	0
FIELD	CCS_EN A	CCS_IE	CCS_IE_ ZC	FH_CLR (WO)	SEND_ MOD	RSSI_E NA	ZC_ENA	R/T
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	CCS_ENA	载波通讯功能使能 CCS_ENA = 1 开启; CCS_ENA = 0 关闭。
6	CCS_IE	载波通讯连续方式中断使能 CCS_IE = 1 开启; CCS_IE = 0 关闭。
5	CCS_IE_ZC	载波通通过零方式中断使能 CCS_IE_ZC = 1 开启; CCS_IE_ZC = 0 关闭。
4	FH_CLR(WO)	误收标志位
3	SEND_MOD	载波发送方式配置位 0, 120K 方波发送模式; 1, DPA 发送模式
2		
1	ZC_ENA	过零点发送方式使能 ZC_ENA = 1, 过零点发送; ZC_ENA = 0, 连续发送。
0	R/T	载波收发状态控制位 R/T = 0, 发送; R/T = 1, 接收。

● CommFrameH Register 载波帧头高字节寄存器 (EXT_ADR = A5H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器用于填写载波发送接收的帧头的高字节, 当载波发送时硬件自动将帧头首先发送。当载波接收时, 硬件会自动校验此寄存器帧头并置标志位。

● CommFrameL Register 载波帧头低字节寄存器 (EXT_ADR = A5H)

BIT	7	6	5	4	3	2	1	0
------------	---	---	---	---	---	---	---	---

FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器用于填写载波发送接收的帧头的低字节，当载波发送时硬件自动将帧头首先发送。当载波接收时，硬件会自动校验此寄存器帧头并置标志位。

● COMM FRAMEH DETECT 外部控制寄存器 (EXT_ADR = A6H)

BIT	7	6	5	4	3	2	1	0
FIELD							D1	D0
RESET	—	—	—	—	—	—	0	0

Bit Number	Bit Mnemonic	Function
7~2		
1~0	COMMFRAMEH DETECT[1: 0]	00: 接收的 16 位帧头全对时帧头接收标志位会置位 01: 接收的 16 位帧头中有 15 位正确时标志位即置位 10: 接收的 16 位帧头中有 14 位正确时标志位即置位 11: 接收的 16 位帧头中有 13 位正确时标志位即置位

● SSC_TXBUF 外部控制寄存器 (EXT_ADR = A7H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

载波发送寄存器，发送载波时写入要发的送字节，过零与连续方式共用此 TXBUFF;

● SSC_RXBUF0 外部控制寄存器 (EXT_ADR = A8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

载波连续模式接收寄存器，当连续方式下载波接收到一个字节后数据放入此寄存器

● SSC_RXBUF1 外部控制寄存器 (EXT_ADR = A9H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0

RESET	0	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---	---

载波过零模式接收寄存器，当过零方式下载波接收到一个字节后数据放入此寄存器

● ZC_OFFSET 过零时刻偏置寄存器 (EXT_ADR = AAH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

在采用过零发送时，发送起始时间可以比 ZCP 引脚输入的方波信号的上升/下降沿提前一定时间，ZC_OFFSET 每增加 1，则时间提前约 26.7μs。

● THRESHOLD_GAIN 载波捕获门限寄存器 (EXT_ADR = ABH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	1	1	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

调节门限增益。

● PRE_THRESHOLD 载波捕获门限寄存器 (EXT_ADR = ACH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	1	0	1	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

载波通信单元为接收状态时，用于设定本地与接收到的伪随机码序列相位同步的捕获门限值，门限值设定越大，相关性检测越严格。

● MAX_CORR 连续模式最大相关峰寄存器 (EXT_ADR = AEH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

指示每个伪码周期内最大的相关峰值，MAX_CORR 值越大，信噪比越好。

● ZC_MAX_CORR 过零点模式最大相关峰寄存器 (EXT_ADR = AFH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

指示每个伪码周期内最大的相关峰值，ZC_MAX_CORR 值越大，信噪比越好。

4.4 应用例程

载波初始化配置：

```
void PLC_Init(void)
{
    DogWatch();
    EXT_ADR = 0xff; //禁能写保护
    EXT_DAT = 0xff;
    //设置预同步唤醒位：40
    EXT_ADR = 0xA2; //设置预同步唤醒位：0-127
    EXT_DAT = 0x28;
    //设置同步帧头：0xF590
    EXT_ADR = 0xA4; //设置同步帧头
    EXT_DAT = 0xF5; //帧头高字节
    EXT_ADR = 0xA5;
    EXT_DAT = 0x90; //帧头低字节
    //同步帧头检测位数：15位
    EXT_ADR = 0xA6;
    EXT_DAT = 0x01;
    //载波捕获门限1：0x50
    EXT_ADR = 0xAB;
    EXT_DAT = 0x50;
    //载波捕获门限2：0x28
    EXT_ADR = 0xAC;
    EXT_DAT = 0x28;
    if (PLCTxType == ZERO_CROSS) //如果为过零点发送
    {
        EXT_ADR = 0xA3; //载波使能·过零中断使能·过零发送方式使能
        EXT_DAT = 0xA3;
        EXT_ADR = 0xAA; //过零时刻偏置
        EXT_DAT = 0x00;
        EXT_ADR = 0xff; //使能写保护
        EXT_DAT = 0x00;
        IT1 = 1; //开启过零中断
        IE |= 0x04;
    }
    else //如果为连续方式发送
    {
        EXT_ADR = 0xA3; //使能连续载波
        EXT_DAT = 0xC1; //120K载波发送方式
        EXT_ADR = 0xff; //使能写保护
        EXT_DAT = 0x00;
        IT2 = 1; //中断触发方式：边沿触发
        EIE |= 0x01; //连续方式载波中断使能
    }
    DogWatch();
}
```

载波发送函数:

```
unsigned char PLC5000_Send(unsigned char *buf, unsigned char len)
{
    DogWatch();
    EXT_ADR = 0xff; //禁能写保护
    EXT_DAT = 0xff;
    if (PLCTxType==NOZERO_CROSS) //如果为连续方式发送
    {
        EXT_ADR = 0xA3; //置为连续方式发送态
        EXT_DAT = 0xC0;
        EXT_ADR = 0xff; //使能写保护
        EXT_DAT = 0x00;
        PLCTxBuf2 = buf; //连续发送指针指向要发送数据地址
        PLCTxLen2 = len; //连续发送数据长度赋值
        PLCTxBit2 = 0; //连续发送计数变量清0
        PLCTxPos2 = 0; //连续发送变量清0
    }
    else //如果为过零点方式发送
    {
        EXT_ADR = 0xA3; //置为过零点方式发送态
        EXT_DAT = 0xA2;
        EXT_ADR = 0xff; //使能写保护
        EXT_DAT = 0x00;
        PLCTxBuf1 = buf; //过零点发送指针指向要发送数据地址
        PLCTxLen1 = len; //过零点发送数据长度赋值
        PLCTxBit1 = 0; //过零点发送计数变量清0
        PLCTxPos1 = 0; //过零点发送变量清0
    }
    return 1;
}
```

连续方式中断处理：

```
void PLCInterrupt2_5000(void) interrupt 7
{
    static unsigned char tempadr2;
    static unsigned char k;
    tempadr2 = EXT_ADR;
    k=0;
    EXT_ADR = 0xff;
    EXT_DAT = 0xff;
    EXT_ADR=0xA3;
    k=EXT_DAT&0x01;
    if(k) //如果为接收态
    {
        EXT_ADR=0xA0;
        k=EXT_DAT&0x01;
        if(k==0x01) //如果帧头同步成功，说明载波成功接收到数据
        {
            EXT_ADR = 0xA8;
            PLCRxBuf2[PLCRxPos2++] = EXT_DAT; //将接收数据寄存器的数读出
        }
        if(PLCRxPos2>=PLC_RX_BUF_LEN)
        {
            PLCRxPos2=0;
        }
    }
    else //如果为发送态
    {
        if(PLCTxPos2>=PLCTxLen2) //判断是否发送完毕
        {
            PLCTxPos2=0;
        }
        else
        {
            EXT_ADR=0xA7;
            EXT_DAT=PLCTxBuf2[PLCTxPos2]; //将数据写入发送寄存器中
            PLCTxPos2++;
        }
    }
    EXT_ADR=tempadr2;
}
```

过零点方式中断处理:

```
void PLCInterrupt1_5000(void) interrupt 2
{
    static unsigned char tempadr1;
    static unsigned char RT_tmp;
    static unsigned char Recv_Judge;
    tempadr1 = EXT_ADR;
    RT_tmp=0;
    Recv_Judge=0;
    EXT_ADR = 0xff;
    EXT_DAT = 0xff;
    EXT_ADR=0xA3;
    RT_tmp =EXT_DAT&0x01;
    if(RT_tmp)
    {
        EXT_ADR=0xA1;
        Recv_Judge = EXT_DAT&0x01;
        if(Recv_Judge==0x01)
        {
            EXT_ADR = 0xA9;
            PLCRxBuf1[PLCRxPos1++] = EXT_DAT;
        }
        if(PLCRxPos1>=PLC_RX_BUF_LEN)
        {
            PLCRxPos1=0;
        }
    }
    else
    {
        if(PLCTxPos1>=PLCTxLen1)
        {
            PLCTxPos1=0;
        }
        else
        {
            EXT_ADR=0xA7;
            EXT_DAT=PLCTxBuf1[PLCTxPos1];
            PLCTxPos1++;
        }
    }
    EXT_ADR = tempadr1;
}
```

5 四通道硬件定时器

5.1 功能简述

PL5010 提供了 4 个通道独立的硬件定时器，可由软件配置 Tx_SEL 位控制单位定时长度为 10ms 或者 100ms。当软件向 TIMERx_VAL 寄存器写入不为零的值时，硬件定时器立刻启动定时，且 Tx_ZERO 位被硬件置为高电平，在每次累加到定时单位长度时（TX_SEL 位决定 10ms 或者 100ms）TIMERx_VAL 寄存器减 1，当 TIMERx_VAL 值为零后，Tx_ZERO 位立刻被硬件置为低电平。MCU 可以通过查询 Tx_ZERO 位的状态，确定定时是否结束。

5.2 编程指南

1. 设置单位定时长度，对 Tx_SEL 位写入“1”单位定时长度为 100ms，写入“0”单位定时长度为 10ms。
2. 设置定时长度，对 TIMERx_VAL 写入需要设定的时间值。
3. MCU 查询 Tx_ZERO 位是否为 0，若为 0 定时完成，若为 1 则等待。

5.3 相关寄存器

- EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

- EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

● TIMER_CON 定时器控制寄存器 (EXT_ADR = A9)

BIT	7	6	5	4	3	2	1	0
FIELD	T4_ZERO	T4_SEL	T3_ZERO	T3_SEL	T2_ZERO	T2_SEL	T1_ZERO	T1_SEL
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	T4_ZERO	定时器通道 4 定时归零标志 T4_ZERO = 1, 定时期间; T4_ZERO = 0, 定时完成。
6	T4_SEL	定时器通道 4 单位定时长度选择位 T4_SEL = 1, 单位定时长度为 100ms; T4_SEL = 0, 单位定时长度为 10ms。
5	T3_ZERO	定时器通道 3 定时归零标志 T3_ZERO = 1, 定时期间; T3_ZERO = 0, 定时完成。
4	T3_SEL	定时器通道 3 单位定时长度选择位 T3_SEL = 1, 单位定时长度为 100ms; T3_SEL = 0, 单位定时长度为 10ms。
3	T2_ZERO	定时器通道 2 定时归零标志 T2_ZERO = 1, 定时期间; T2_ZERO = 0, 定时完成。
2	T2_SEL	定时器通道 2 单位定时长度选择位 T2_SEL = 1, 定时长度为 100ms; T2_SEL = 0, 单位定时长度为 10ms。
1	T1_ZERO	定时器通道 1 定时归零标志 T1_ZERO = 1, 定时期间; T1_ZERO = 0, 定时完成。
0	T1_SEL	定时器通道 1 单位定时长度选择位 T1_SEL = 1, 单位定时长度为 100ms; T1_SEL = 0, 单位定时长度为 10ms。

此寄存器位于扩展地址空间内, 需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● TIMER1_VAL 定时器通道 1 定时值寄存器 (EXT_ADR = B1H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内, 需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

当 TIMER1_VAL 不为零时, 硬件开始启动定时, 直到递减至为 0, 单位定时长度由 T1_SEL 位决定。

● TIMER2_VAL 定时器通道 2 定时值寄存器 (EXT_ADR = B2H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

当 TIMER2_VAL 不为零时，硬件开始启动定时，直到递减至为 0，单位定时长度由 T2_SEL 位决定。

● TIMER3_VAL 定时器通道 3 定时值寄存器 (EXT_ADR = B3H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

当 TIMER3_VAL 不为零时，硬件开始启动定时，直到递减至为 0，单位定时长度由 T3_SEL 位决定。

● TIMER4_VAL 定时器通道 4 定时值寄存器 (EXT_ADR = B4H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

当 TIMER4_VAL 不为零时，硬件开始启动定时，直到递减至为 0，单位定时长度由 T4_SEL 位决定。

5.4 应用例程

利用定时器通道 1，IO P2^7 输出方波：


```

while(1) //四通道硬件定时器
{
    DogWatch();
    EXT_ADR = 0xff;
    EXT_DAT = 0xff; //解除写保护
    EXT_ADR=0xB0;
    EXT_DAT=0x02; //选择单位定时长度10ms,标志位置1
    EXT_ADR=0xB1;
    EXT_DAT=20; //设定计时长度20,即总定时时间=20*10ms=200ms
    EXT_ADR=0xB0;
    do
    {
        cmd=EXT_DAT;
        DogWatch(); //喂狗操作
    }while( (cmd&0x02)==0x02); //判断定时是否完成
    P27=~P27;
    EXT_ADR = 0xff;
    EXT_DAT = 0x00; //使能写保护
}

```

6 ISO7816 接口

6.1 功能简述

PL5010 的 ISO7816 通信模块支持 T=0 协议，为异步半双工字符传输协议，它包括一个起始位，8 个数据位和 1 个奇偶校验位。由一个 ISO7816 逻辑电路通过软件配置进行双通道切换，可以分别同 ESAM/PSAM 模块或 IC 卡进行数据通讯。

6.2 编程指南

ISO7816 通信模块可分别同 ESAM/PSAM 模块或 IC 卡进行通信，工作步骤如下：

1. IC 卡和 ESAM/PSAM 模块的通道选择

通过 ISO7816_CON 寄存器中 IOSEL 的值可以自由选择任意通道。

2. 发送数据

若置 ISO7816_CON 寄存器中的 IO_WEN（发送使能）为 1，则 ISO7816_STATE 寄存器中的 WBYTE_RDY 置被为 0，硬件将 DI_7816 寄存器中的值串行输出：

- 发送时先发低位，并在前面加 1bit 低电平起始位，在字节后加 1bit 奇偶校验位，每个 ETU 发送 1bit，在第 11 个 ETU 将 IO 线拉高，变为接收态，检测线上数据。
- 若检测到线上出现低电平，则在第 11 个 ETU 结束时，令 ISO7816_STATE 寄存器中的 W_PLR_ERR（重发标志）有效，并且 WBYTE_RDY 始终为 0，W_PLR_ERR 必须用软件清除，并且在至少间隔 2 个 ETU 后重新发送该字节。
- 如果在第 11 个 ETU 没有检测到低电平，则在第 12 个 ETU 结束时将 WBYTE_RDY 置为 1，表示一个字符发送完成。
- 如果要继续接收下一个字符则必须重新给出发送使能 IO_WEN。

3. 接收数据

若 ISO7816_CON 寄存器中的 IO_REN（接收使能）为 1，则 ISO7816_STATE 寄存器中的 RBYET_RDY 置为 0，等待接收数据：

- 从检测到起始位开始，总在半个 ETU 时锁存数据，即在第 1.5 个 ETU 锁存第一位数据，在第 9.5 个 ETU 锁存奇偶校验位。

- 如果奇偶校验正确，则在第 11.5 个 ETU 将 RBYET_RDY 置为 1。
- 如果奇偶校验错误，则在第 10.5 个 ETU 发送一个 1 个 ETU 长度的低电平，并且 RBYET_RDY 始终为 0，等待接收重发数据。
- 如果连续三次接收的数据都错误，则在接收第三次接收的字符的 10.5 个 ETU 将 ISO7816_CON 寄存器中 ERR_INTP 信号置为 1，并在 11.5 个 ETU 将 RBYET_RDY 置为 1。
- 如果要继续接收下一个字符，必须在检测到 RBYET_RDY 后重新给出接收使能。

6.3 寄存器

- EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

- EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

- DO_78167816 数据发送缓冲寄存器 (EXT_ADR = 8DH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

由 CPU 将待发的数据送入到该寄存器中。

● DI_78167816 数据接收缓冲寄存器 (EXT_ADR = 8EH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

接收完的数据送入该缓冲寄存器，由 CPU 将数据取走。

● ISO7816_CON (EXT_ADR = 8FH) 7816 控制寄存器 (S:8FH)

BIT	7	6	5	4	3	2	1	0
FIELD	IOSEL		ERR_RST	C_INTP			IO_REN	IO_WEN
RESET	0	—	0	0	—	—	0	0

Bit Number	Bit Mnemonic	Function
7	IOSEL	IC 卡/ESAM 选择位 IOSEL = 1 时选择 IC 卡，IOSEL = 0 时选择 ESAM。
6		
5	ERR_RST	清除错误标志位 用来清除状态寄存器中的欠时错误标 TIME_INADQ，超时错误标志位 TIME_OV，连续三次接收错误标志位 ERR_INTP，以及重发标志位 W_PLR_ERR。ERR_RST 低有效，在清除完错误标志后，由硬件自动置高。
4	C_INTP	软复位 令 ISO7816 逻辑产生复位，C_INTP 高有效，在复位完成后由硬件自动置低。
3~2		
1	IO_REN	接收数据使能位 IO_REN 高有效，由硬件自动置低。
0	IO_WEN	发送数据使能位 IO_WEN 高有效，由硬件自动置低。

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

C_INTP 软复位时间为 3.4us，在软复位期间对 ERR_RSR，IO_REN，IO_WEN，操作无效。

● I_MAX_ETU[23:16]IC 卡字符收发最大 ETU 间隔寄存器 (EXT_ADR = 90H)

BIT	7	6	5	4	3	2	1	0
FIELD	D23	D22	D21	D20	D19	D18	D17	D16
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● I_MAX_ETU[15:8]IC 卡字符收发最大 ETU 间隔寄存器 (EXT_ADR = 91H)

BIT	7	6	5	4	3	2	1	0
FIELD	D15	D14	D13	D12	D11	D10	D9	D8
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● I_MAX_ETU[7:0]IC 卡字符收发最大 ETU 间隔寄存器 (EXT_ADR = 92H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

IC 卡接收和发送两个字符之间的时间间隔不能大于该寄存器设定的 ETU 个数，否则 TIME_OV 标志将有效。

● I_ETU[15:8]IC 卡基本时间单元 (ETU) 寄存器 (EXT_ADR = 93H)

BIT	7	6	5	4	3	2	1	0
FIELD	D15	D14	D13	D12	D11	D10	D9	D8
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● I_ETU[7:0]IC 卡基本时间单元 (ETU) 寄存器 (EXT_ADR = 94H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

用来设定通信中 1 位持续传输时间的周期值，即一个 ETU 时间。

● I_MIN_ETU[8]IC 卡字符收发最小 ETU 间隔寄存器 (EXT_ADR = 95H)

BIT	7	6	5	4	3	2	1	0
FIELD								D8
RESET	—	—	—	—	—	—	—	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● I_MIN_ETU[7:0]IC 卡字符收发最小 ETU 间隔寄存器 (EXT_ADR = 96H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

接收和发送两个字符之间的时间间隔不能小于该寄存器设定的 ETU 数，否则 TIME_INADQ 标志将有效。

● E_MAX_ETU[23:16]ESAM 字符收发最大 ETU 间隔寄存器 (EXT_ADR = 97H)

BIT	7	6	5	4	3	2	1	0
FIELD	D23	D22	D21	D20	D19	D18	D17	D16
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● E_MAX_ETU[15:8]ESAM 字符收发最大 ETU 间隔寄存器 (EXT_ADR = 98H)

BIT	7	6	5	4	3	2	1	0
FIELD	D15	D14	D13	D12	D11	D10	D9	D8
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● E_MAX_ETU[7:0]ESAM 字符收发最大 ETU 间隔寄存器 (EXT_ADR = 99H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

ESAM 接收和发送两个字符之间的时间间隔不能大于该寄存器设定的 ETU 个数，否则 TIME_OV 标志将有效。

● E_ETU[15:8]ESAM 基本时间单元 (ETU) 寄存器 (EXT_ADR = 9AH)

BIT	7	6	5	4	3	2	1	0
FIELD	D15	D14	D13	D12	D11	D10	D9	D8
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● E_ETU[7:0]ESAM 基本时间单元 (ETU) 寄存器 (EXT_ADR = 9BH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

用来设定 ESAM 中 1 位持续传输时间的周期值，即一个 ETU 时间。

● E_MIN_ETU[8]ESAM 字符收发最小 ETU 间隔寄存器 (EXT_ADR = 9CH)

BIT	7	6	5	4	3	2	1	0
FIELD								D8
RESET	—	—	—	—	—	—	—	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● E_MIN_ETU[7:0]ESAM 字符收发最小 ETU 间隔寄存器 (EXT_ADR = 9DH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

ESAM 接收和发送两个字符之间的时间间隔不能小于该寄存器设定的 ETU 数，否则 TIME_INADC 标志将有效。

● ISO7816_STATUSISO7816 状态寄存器 (EXT_ADR = 9EH)

BIT	7	6	5	4	3	2	1	0
------------	----------	----------	----------	----------	----------	----------	----------	----------

FIELD	TIME_I NADQ	TIME_O V	ERR_IN TP	RBYTE_ RDY			W_PLR_ ERR	WBYTE _RDY
RESET	0	0	0	0	—	—	0	0

Bit Number	Bit Mnemonic	Function
7	TIME_INADQ	发送/接收字符时间间隔小于最小要求错误标志 高有效，当下一次接/发字符间间隔满足要求时该位可由硬件自动清除，也可以通过软件由 ERR_RST 来清除。
6	TIME_OV	发送/接收字符时间间隔大于最大要求错误标志 高有效，只能通过软件由 ERR_RST 来清除。
5	ERR_INTP	连续三次接收数据校验错误标志 高有效，只能通过软件由 ERR_RST 来清除。
4	RBYTE_RDY	接收一个字符完成标志 高有效。
3~2		
1	W_PLR_ERR	重发标志 高有效。在发送数据时，当检测到接收方发送的接收错误信号时，该标志位有效，只能通过软件由 ERR_RST 来清除。
0	WBYTE_RDY	发送一个字符完成标志 高有效。

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● EXT_CTRL 外部控制寄存器 (EXT_ADR = FDH)

BIT	7	6	5	4	3	2	1	0
FIELD	TCKSEL				IC_DIV1	IC_DIV0	ESAM_DIV1	ESAM_DIV0
RESET	0	—	—	—	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	TCKSEL	秒脉冲输出选择位 TCKSEL = 0 时为 1Hz 输出，TCKSEL = 1 时为 1KHz 输出。
6~4		
3	IC_DIV1	IC 卡时钟频率控制 IC_DIV1 = 0, IC_DIV0 = 0, IC 卡时钟被禁止； IC_DIV1 = 0, IC_DIV0 = 1, IC 卡时钟输出为外部晶振频率的 2 分频；
2	IC_DIV0	IC_DIV1 = 1, IC_DIV0 = 0, IC 卡时钟输出为外部晶振频率的 3 分频； IC_DIV1 = 1, IC_DIV0 = 1, IC 卡时钟输出为外部晶振频率的 4 分频。

1	ESAM_DIV1	ESAM 时钟频率控制 ESAM_DIV1 = 0, ESAM_DIV0 = 0, ESAM 时钟被禁止; ESAM_DIV1 = 0, ESAM_DIV0 = 1, ESAM 时钟输出为外部晶振频率的 2 分频;
0	ESAM_DIV0	ESAM_DIV1 = 1, ESAM_DIV0 = 0, ESAM 时钟输出为外部晶振频率的 3 分频; ESAM_DIV1 = 1, ESAM_DIV0 = 1, ESAM 时钟输出为外部晶振频率的 4 分频。

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

7 FGEN 可编程频率发生器

7.1 功能简述

PL5010 的可编程频率发生器复用 IO 管脚 P27 作为频率发生器的输出端，可由软件配置输出频率，其频率变化范围在 1.05KHz~10KHz 之间。

7.2 编程指南

PL5010 的可编程频率发生器的频率输出计算公式为：

$$F_{gen} = F_{osc} / (20Val[7:0] + 600);$$

若 $F_{osc} = 6\text{MHz}$ 时，则 $F_{gen} = 300 / (Val + 30)$;

使用可编程频率发生器的工作步骤如下：

1. 设置可编程频率发生器使能：EXT_ADR 写入 FEH，EXT_DAT.2 置 1。
2. 设置可编程频率发生器频率输出范围：往 FGEN 寄存器写入设定 Val 值。

7.3 寄存器

● EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

● FGEN 可编程频率发生器频率控制寄存器 (EXT_ADR = 9FH)

BIT	7	6	5	4	3	2	1	0
------------	----------	----------	----------	----------	----------	----------	----------	----------

FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

向该寄存器写入值可以调整频率发生器输出频率。

● EXT_CFG 扩展配置寄存器 (EXT_ADR = FEH)

BIT	7	6	5	4	3	2	1	0
FIELD	TCK_ENA					FGEN_ENA	IR38K_ENA	PWM_ENA
RESET	0	—	—	—	—	0	0	0

Bit Number	Bit Mnemonic	Function
7	TCK_ENA	1Hz 输出使能 1, P26 输出 1Hz; 0, P26 不输出, 当作普通 I/O 口用。
6~3		
2	FGEN_ENA	频率发生器使能位 1, 使能频率发生器功能; 0, 不使能频率发生器功能。
1	IR38K_ENA	红外通讯使能 IR38K=1 开, IR38K=0 关。
0	PWM_ENA	脉宽调制使能位 高有效。

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

8 PWM

8.1 功能简述

PL5010 提供了可配频率和占空比的脉宽调制 IO。P14/PWM 引脚可输出可调的 PWM 波形。

8.2 编程指南

打开 PWM 脉宽调整使能。配置寄存器 `FREQ` 两位控制输出脉宽频率。配置 `DUTY_CYCLE` 调节输出脉宽的占空比。

8.2 寄存器

- `EXT_ADR` 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

- `EXT_DAT` 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 `EXT_ADR` 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

- `EXT_CFG` 扩展配置寄存器 (`EXT_ADR = FEH`)

BIT	7	6	5	4	3	2	1	0
FIELD	TCK_ENA					FGEN_ENA	IR38K_ENA	PWM_ENA
RESET	0	—	—	—	—	0	0	0

Bit Number	Bit Mnemonic	Function
------------	--------------	----------

7	TCK_ENA	1Hz 输出使能 1, P26 输出 1Hz; 0, P26 不输出, 当作普通 I/O 口用。
6~3		
2	FGEN_ENA	频率发生器使能位 1, 使能频率发生器功能; 0, 不使能频率发生器功能。
1	IR38K_ENA	红外通讯使能 IR38K=1 开, IR38K=0 关。
0	PWM_ENA	脉宽调制使能位 1: ENABLE; 0: DISABLE

● PWM PWM 配置寄存器 (EXT_ADR = 8AH)

BIT	7	6	5	4	3	2	1	0
FIELD	FREQ[1:0]		DUTY_CYCLE[5:0]					
RESET	0		0					

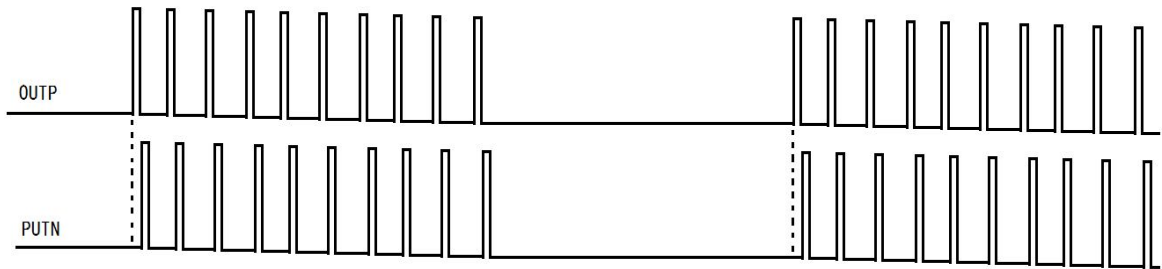
Bit Number	Bit Mnemonic	Function
7~6	FREQ[1:0]	频率配置 00: 1KHZ 01: 2KHZ 10: 4KHZ 11: 7.5KHZ
5~0	DUTY_CYCLE[5:0]	占空比配置 $DUTY_CYCLE = 10\% + value * 1.25\%$ value 为写入的 6bit 值。 value = 0x00 时, 占空比为 10%, 最小占空比。 最大占空比为 88.75%, 即 value = 0x3F。

此寄存器位于扩展地址空间内, 需通过 EXT_DAT 和 EXT_ADR 进行访问。

9 差分串行通信单元

9.1 功能简述

PL5010 的差分串行通信单元, 为异步半双工字符传输方式, 可配置为过零点工作方式, 具有中断。波特率固定为 9600bps。它包括一个起始位 (为 0), 8 个数据位和 1 个可选的奇偶校验位, 2 个停止位 (为 1)。其控制方式与 UART 相似, 通信引脚为 #37、#38 引脚。差分通信可通过线圈隔离通信。



波形示意图

说明: 10 个脉冲构成一个为 0 的比特位, 之间的长的低电平表示一个为 1 的比特位, 如图所示代表数据 010。数据是低位在前高位在后。

9.2 编程指南

打开通信单元控制总使能 ENA、中断总使能 ENA 和差分串行通信单元使能 EDS, 向 DIFF_SBUF 送入数据, 发送完成后 TI 标志位被置 1, 清除 TI 标志位然后发送下一字节数据。

同时打开 ENA 和 REC_ENA 接收使能位, 读取 RI 接收标志位, 若为 1, 数据接收完成, 即可读取 DIFF_RBUF 寄存器, 然后清除 RI 位接收下一字节数据。

发送接收数据可使用查询方式或中断方式来完成。

9.3 寄存器

- EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

● IE 中断使能寄存器 (S:A8H)

BIT	7	6	5	4	3	2	1	0
FIELD	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EA	中断允许总控制位 EA = 0, 屏蔽所有中断申请; EA = 1, CPU 开放中断。

● EIE 扩展中断使能控制寄存器 (S:A9H)

BIT	7	6	5	4	3	2	1	0
FIELD	EDS	ESPI	ES4	ES3	EX3	ES2	ET3	EX2
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EDS	差分串行通信单元中断允许位

● DIFF_CON 差分通信控制寄存器 (EXT_ADR = B5H)

BIT	7	6	5	4	3	2	1	0
FIELD	ENA	REC_ENA	ZC_ENA	MOD	TB8			
RESET	0	0	0	0	0	—	—	—

Bit Number	Bit Mnemonic	Function
7	ENA	通信单元总使能 置 1 打开使能。
6	REC_ENA	接收使能 置 1 打开使能。
5	ZC_ENA	过零点工作模式 置 1 为过零点工作方式。
4	MOD	校验位控制 置 1 为加校验位的 12 位数据长度方式工作，置 0 为 11 位数据长度方式。
3	TB8	发送的第 9 位数据位 可作为奇偶校验位，由软件控制，在 MOD 位为 1 时可用。
2~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● DIFF_STATUS 差分通信状态寄存器 (EXT_ADR = B6H)

BIT	7	6	5	4	3	2	1	0
FIELD	TI	RI	SC_FLAG	RB8				
RESET	0	0	0	0	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	TI	发送数据标志位 DIFF_SBUF 发送完一个字节数据后置 1，由软件清 0。
6	RI	接收数据标志位 DIFF_RBUF 接收完一个字节数据后置 1，由软件清 0。
5	SC_FLAG	数据冲突标志位 在过零模式下当数据发送接收冲撞时该位置 1，由软件清 0。
4	RB8	接收的第 9 位数据位 接收到的第 9 位数据，在 MOD 位为 1 时可用。
3~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● DIFF_SBUF (EXT_ADR = B7H) 发送数据缓冲寄存器 (S:B7H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● DIFF_RBUF 接收数据缓冲寄存器 (EXT_ADR = B8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● PHASECNT 过零点相位监测寄存器 (EXT_ADR = B9H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

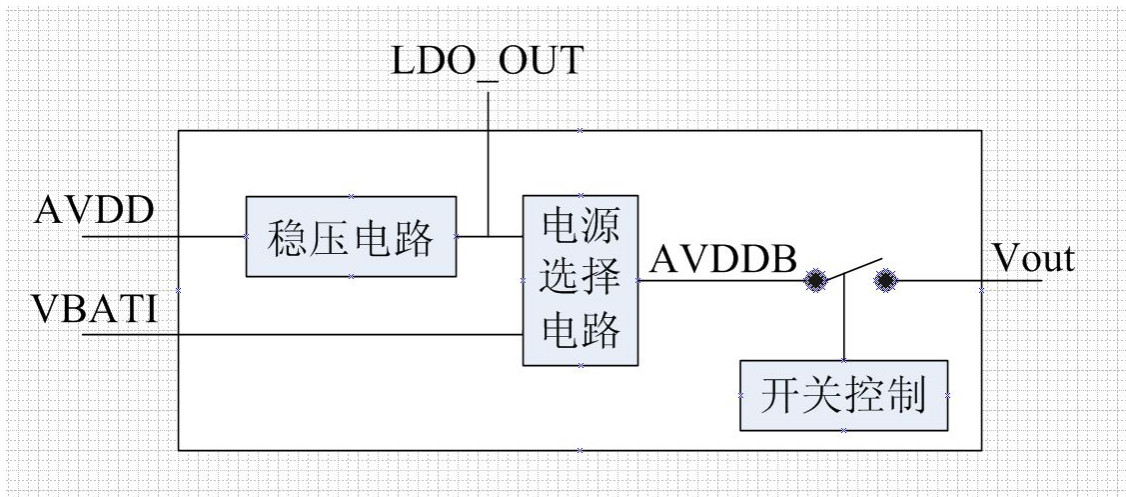
10 低功耗电源管理模块

10.1 功能简述

PL5010 提供了系统电源掉电管理的低功耗工作方式。在监测到系统电源掉电后可以启动睡眠模式通过外部电池供电来维持 RTC 时钟和一些必要掉电唤醒电路工作，同时 CPU 及其他功能模块停止工作，以此达到降低功耗的目的。可以通过定时或外部触发方式唤醒整个芯片工作。

10.2 低功耗设计

低功耗电源系统如下图所示。



低功耗设计时，VBATI 引脚接电池输入，芯片内置稳压电路及电压比较电路，当外部电源掉电，芯片进入掉电模式，自动切换到 VBATI 供电，通过引脚 AVDDB 引脚输出电池电压。

通过发送睡眠指令从掉电模式进入睡眠模式后，开关控制电路断开，VOUT 将无电压输出，但 AVDDB 提供电池电压输出。

所以，在设计初期，要考虑到需要睡眠模式下供电的外围电路接 AVDDB 引脚；睡眠模式下需要断电，但唤醒后需要供电的外围电路接 VOUT 引脚。

VOUT 引脚在系统供电时为 4V，在电池供电时为电池电压。建议与 IO 相连接的外部电路均需要由 VOUT 电源供电，以防止在电池供电时 IO 通过外部电路漏电，造成对电池的不必要消耗（芯片在复位后所有 IO 默认状态都为高，此时很有可能对外漏电）。

10.3 编程指南

首先检测 CMPI 引脚的 PFI 标志位，当标志位为 1 时说明检测到系统电源掉电。

此时应在完全掉电前向片内 ROM 中写入需要保存的数据等进入睡眠模式前需要执行的一些必要操作。

执行完成后向 POWER_DOWN 寄存器中写入 55H 关断电源，进入到睡眠模式。这时芯片(此时只有电池通过 VBATI 引脚供电维持 RTC 时钟和一些必要的掉电唤醒电路工作)。注意这个操作要在 AVDD 电源引脚电压低于 3V 之前操作完成，否则芯片的工作电源会切换到电池供电，这个过程会发生芯片复位。

当需要在睡眠模式唤醒芯片工作时可通过给唤醒引脚 P40 或 P41 一个下降沿信号唤醒或定时中断唤醒芯片工作(此时只有 32768 晶振起振，并作为 CPU 主时钟)。如需继续进入睡眠模式向 POWER_DOWN 寄存器中写入 55H 即可。

当 AVDD 引脚由无电转为上电状态时，芯片会自动从睡眠状态唤醒，芯片重新复位工作。并由电池供电切换到正常供电。

对电池电压检测而言，首先需要打开电池电压检测使能位 VBAT_CON 寄存器的 Bit7 即 ENA=1 打开使能；可选检测频率为 32KHZ 或 1HZ 以方便降低功耗；可选电池电压报警的检测次数来去抖动。读取 VBFO 位是否置位来判断电池是否欠压。

*注意：

- 1、在只有电池供电或睡眠模式下芯片只有 32768 晶振起振并作为 CPU 主时钟。
- 2、在有电池的前提下，如果没有及时进入睡眠模式，AVDD 的上电或掉电都会引起芯片的复位。
- 3、如果需要在只有电池供电时，进行一些外部电路的操作。这些外部电路的供电需要 VOUT 引脚提供。因为 VOUT 在睡眠模式会自动关断而且是芯片内部给出。不会造成对其它电源的漏电。
- 4、P40、P41 两个唤醒引脚需要通过 AVDDDB 来提供上拉。

10.4 寄存器

- EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

● STATUS 状态寄存器 (S:87H)

BIT	7	6	5	4	3	2	1	0
FIELD	SMOD	SMOD1	WDT	PU		LP_MOD	VBFO	PFI
RESET	0	0	0	0	—	0	0	0

Bit Number	Bit Mnemonic	Function
7	SMOD	串口 0 波特率加倍控制位 SMOD 为 1，其波特率加倍，SMOD 为 0，其波特率不加倍。
6	SMOD1	串口 1 的波特率加倍控制 SMOD 为 1，其波特率加倍，SMOD 为 0，其波特率不加倍。
5	WDT	复位标志位 当系统复位后，该标志位自动置 1，可软件清零。
4	PU	上电标志位 当系统上电运行后，该标志位自动置 1，可软件清零。
3		
2	LP_MOD	系统电压标志位 检测 AVDD 引脚 5V 系统电压。系统有电时为 1，无电时为 0。
1	VBFO	电池电压检测标志位 检测 VBATI 引脚电压。电池电压大于等于 2.8 为 0，低于 2.6V 为 1。
0	PFI	掉电标志位 当 CMPI 引脚的输入电压低于 2.8V 时，该位自动置 1。

● LP_CON 外部扩展寄存器 (EXT_ADR = 88H)

BIT	7	6	5	4	3	2	1	0
FIELD	SEC_CN T3	SEC_CN T2	SEC_CN T1	SEC_CN T0	MOD	TIMER_ FLAG	INT1_FL AG	INT2_FL AG
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	SEC_CNT3	定时唤醒的定时时间长度（每比特为 1 秒）
6	SEC_CNT2	
5	SEC_CNT1	
4	SEC_CNT0	
3	MOD	进入低功耗的状态下，为 1 时允许定时唤醒和触发唤醒，为 0 时只允许触发唤醒。
2	TIMER_FLAG	定时唤醒标志位 进入低功耗的状态下，当定时唤醒后置 1。
1	INT1_FLAG	WAKEUP_INT1 引脚的触发中断标志位 进入低功耗的状态下，当检测到 WAKEUP_INT1 引脚下降沿后置 1（软件清 0）。
0	INT2_FLAG	WEEKUP_INT2 引脚的触发中断标志位 进入低功耗的状态下，当检测到 WEEKUP_INT2 引脚下降沿后置 1（软件清 0）

● POWER_DOWN 电源掉电寄存器（EXT_ADR = 89H）

BIT	7	6	5	4	3	2	1	0
FIELD	55H							
RESET	0							

Bit Number	Bit Mnemonic	Function
7~0	55H	写入 55H 使计入到睡眠模式（低功耗）。

● VBAT_CON 外部扩展寄存器（EXT_ADR = 8BH）

BIT	7	6	5	4	3	2	1	0
FIELD	ENA	SEL						
RESET	0	0	—	—	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	ENA	电池电压检测使能 1: ENABLE 0: DISABLE
6	SEL	电池电压检测频率选择位 0: 电池电压检测频率为 1HZ 1: 电池电压检测频率为 32KHZ (32KHZ 检测模式功耗较大在电池供电情况下慎用)
5~0		

● VBAT_CNT 外部扩展寄存器 (EXT_ADR = 8CH)

BIT	7	6	5	4	3	2	1	0
FIELD						CNT2	CNT1	CNT0
RESET	—	—	—	—	—	0	0	0

CNT[3: 0]为电池欠压报警的连续检测次数，只有当连续检测到电池同一电压状态时才会改变 VBFO 位状态。

11 RTC 控制单元

11.1 32768 实时钟晶振连接方式

32768 晶振的输出与芯片引脚 XT2O 间需要串联一个 $100\text{K}\Omega$ 的电阻，晶振管脚对地电容为 15pF 。

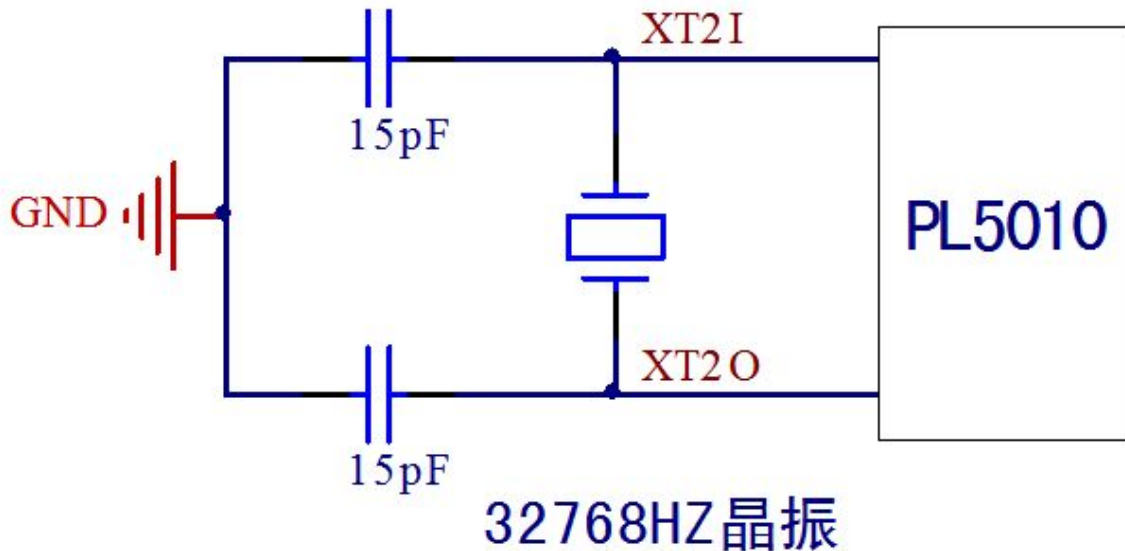


图 13 32768 实时钟晶振连接方式

11.2 功能简述

RTC 单元包括时间校正寄存器、秒寄存器、分钟寄存器、小时寄存器、星期寄存器、天寄存器、月寄存器、年寄存器。在系统掉电的情况下，只要有电池供电，RTC 时钟仍可正常工作。（可配合 P26/CLK1HZ 引脚进行时钟秒校准）

11.3 寄存器

- EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器，写入数据或是读出数据。

● Time_Adjust_Register 时间校准寄存器 (EXT_ADR = 80H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

时钟校准根据 Time_Adjust_Register 寄存器 D0~D7 符号化二进制数设定。以 $\pm 3.05 \times 10^{-6}$ 为单位校准。

若不使用本功能，请将全部位设为 0。

时钟校准功能 10s 动作一次。

调整示例：

调快时间

通过测 P26/CLK1HZ 引脚输出的 1Hz 误差来计算晶振频率为 32767.7Hz，则

$$(32767.7 - 32768) / 32768 \approx -9.16 \times 10^{-6};$$

$$-9.16 \times 10^{-6} / 3.05 \times 10^{-6} \approx -3;$$

-3 转换为十六进制为 FDh，输入 Time_Adjust_Register 寄存器。然后再测试 1Hz 输出，重复上述过程，直到校准为止。

调慢时间

通过测 P26/CLK1HZ 引脚输出的 1Hz 误差来计算晶振频率为 32768.3Hz，则

$$(32768.3 - 32768) / 32768 \approx 9.16 \times 10^{-6};$$

$$9.16 \times 10^{-6} / 3.05 \times 10^{-6} \approx 3;$$

3 转换为十六进制为 03h，输入 Time_Adjust_Register 寄存器。然后再测试 1Hz 输出，重复上述过程，直到校准为止。

● Second_Register 秒寄存器 (EXT_ADR = 81H)

BIT	7	6	5	4	3	2	1	0
FIELD		D6	D5	D4	D3	D2	D1	D0
RESET	—	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● Minute_Register 分钟寄存器 (EXT_ADR = 82H)

BIT	7	6	5	4	3	2	1	0
FIELD		D6	D5	D4	D3	D2	D1	D0
RESET	—	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● Hour_Register 小时寄存器 (EXT_ADR = 83H)

BIT	7	6	5	4	3	2	1	0
FIELD			D5	D4	D3	D2	D1	D0
RESET	—	—	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● Week_Register 星期寄存器 (EXT_ADR = 84H)

BIT	7	6	5	4	3	2	1	0
FIELD						D2	D1	D0
RESET	—	—	—	—	—	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● Day_Register 日期寄存器 (EXT_ADR = 85H)

BIT	7	6	5	4	3	2	1	0
FIELD			D5	D4	D3	D2	D1	D0
RESET	—	—	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● Month_Register 月寄存器 (EXT_ADR = 86H)

BIT	7	6	5	4	3	2	1	0
FIELD				D4	D3	D2	D1	D0
RESET	—	—	—	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● Year_Register 年寄存器 (EXT_ADR = 87H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问。

● EXT_CTRL 外部控制寄存器 (EXT_ADR = FDH)

BIT	7	6	5	4	3	2	1	0
FIELD	TCKSEL							
RESET	0	—	—	—	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	TCKSEL	秒脉冲输出选择位 TCKSEL 置 0 时为 1Hz 输出，TCKSEL = 1 时为 1KHz 输出。
6~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

● EXT_CFG 外部控制寄存器 (EXT_ADR = FEH)

BIT	7	6	5	4	3	2	1	0
FIELD	TCK_EN A							
RESET	0	—	—	—	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	TCK_ENA	秒脉冲输出使能位

		TCK_ENA 置 0 时为关闭 P26 引脚的 1Hz 输出，置 1 时使能 P26 引脚的 1Hz 输出。
6~0		

此寄存器位于扩展地址空间内，需通过 EXT_DAT 和 EXT_ADR 进行访问读写。

12 SPI 接口

12.1 功能简述

PL5010 提供了一个通用的 SPI 接口，它可以使 MCU 与各种外围设备以串行方式进行通信以交换信息，SPI 通讯时 PL5010 只能作为主机使用。

12.2 编程指南

SPI 外设使能 SPE 置 1，配置时钟分频，发送时先清 SPIF 标志位，然后向 SPI_DATA 中写入要发送数据的第一个字节，此字节数据发送完成后便进入 SPI 中断，清 SPIF 标志位，然后再向 SPI_DATA 中送入下一个字节，本字节发完后将再一次进入中断，如此往复，直至数据全部发送完毕。SPI 在发送一个字节后，同时会收到一个字节。在接收数据时，主机仍需要发送 0x00 无效数据来继续为从机提供 CLK 时钟，每次在进入发送中断时，向 SPI_DATA 中写入要发送的数据后，即可读取 SPI_DATA 中的数据。

12.3 寄存器

- IE 中断使能寄存器 (S: A8H)

BIT	7	6	5	4	3	2	1	0
FIELD	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	EA	中断允许总控制位 EA = 0，屏蔽所有中断申请；EA = 1，CPU 开放中断。

- EIE 扩展中断使能控制寄存器 (S: A9H)

BIT	7	6	5	4	3	2	1	0
FIELD	EDS	ESPI	ES4	ES3	EX3	ES2	ET3	EX2

RESET	0	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---	---

Bit Number	Bit Mnemonic	Function
6	ESPI	SPI 中断允许位

● SPI_CON 外部控制寄存器 (EXT_ADR =C4H)

BIT	7	6	5	4	3	2	1	0
FIELD	SPE	CSMOD	CPOL	CPHA	SPR3	SPR2	SPR1	SPR0
RESET	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Function
7	SPE	SPI 外设使能 1: ENABLE 0: DISABLE
6	CSMOD	片选模式 0: 用户模式, 用户通过 IO 控制片选信号 1: 通用模式, 按字节控制
5	CPOL	时钟极性 0: 时钟空闲时为低电平 1: 时钟空闲时为高电平
4	CPHA	时钟相位 0: 时钟第一个边沿采集数据 1: 时钟第二个边沿采集数据
3~0	SPR[3: 0]	时钟分频 为 0~15 对 6M 时钟分频

● SPI_STATUS 外部控制寄存器 (EXT_ADR =C5H)

BIT	7	6	5	4	3	2	1	0
FIELD	SPIF	WCOL						
RESET	0	0	—	—	—	—	—	—

Bit Number	Bit Mnemonic	Function
7	SPIF	传输完一个字节标志位置 1, 软件清 0
6	WCOL	写冲突标志位, 软件清 0

● SPI_DATA 外部控制寄存器 (EXT_ADR =C6H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

SPI 数据收发寄存器

13 嵌入式 ROM 数据存储器

13.1 功能简述

PL5010A 提供了 16K 字节的片内 ROM 数据存储区，PL5010B 未提供数据存储区，仅包含 60K 字节的程序存储区，CPU 可以通过特定的指令序列启动芯片内部的硬件逻辑，将指定的 RAM 中的数据写入到 ROM 数据存储区指定的位置，或是将 ROM 数据存储区中的数据读出到 RAM 中指定的位置。在编程或读取数据期间，CPU 将被自动挂起；完成操作后，硬件自动激活 CPU，程序将继续运行。

13.2 编程指南

对 ROM 数据存储区的操作，需要在一定的时间内（64 个指令周期）连续的进行一系列特定的指令序列才可以启动硬件逻辑，若是在指定的时间内没有完成指定序列的操作，硬件逻辑将认为超时，已经执行的序列指令被视为无效，需要重新开始执行该序列指令。需要注意的是，同片内 ROM 数据存储区交互的 RAM 为外部 RAM，地址范围为 1E00H—1EFFH，启动指令序列如下：

1. 解除写保护，EXT_ADR 写入 FFH，EXT_DAT 写入 FFH。
2. 设置 ROM 数据存储区的起始字节地址，EXT_ADR 写入 E0H，EXT_DAT 写入要设定的数据存储区的起始低位地址。EXT_ADR 写入 E1H，EXT_DAT 写入要设定的数据存储区的高位地址。
3. 设置 RAM 的起始地址，EXT_ADR 写入 E2H，EXT_DAT 写入要设定的 RAM 起始地址。
4. 设置 RAM 操作字节的个数，EXT_ADR 写入 E3H，EXT_DAT 写入要设定的 RAM 操作个数。
5. 启动读或是编程使能，EXT_ADR 写入 E4H，EXT_DAT 写入 01H 将启动对 ROM 数据存储区的编程使能，若写入 10H 则将启动读使能。
6. 写保护使能，EXT_ADR 写入 FFH，EXT_DAT 写入 00H。

注意事项:

1. 看门狗在对数据存储区进行操作时也会工作, 当要进行大容量的数据交换时注意喂狗时间, 需要定时喂狗。
2. 数据存储区以页为单位, 每页为 128 个字节。
3. 对数据存储区编程只能以页为单位, 并且不能够跨页操作。
4. 对数据存储区进行编程操作时, 如果仅修改一页中的部分数据字节, 需要先将该页的数据读出到 RAM 中, 修改后, 再将该页数据写回原地址。

表 3 16K Bytes 数据存储区页地址分配表

页数	起始地址	结束地址
0	0000h	007fh
1	0080h	00ffh
2	0100h	017fh
3	0180h	01ffh
...
30	0f00h	0f7fh
31	0f80h	0f00h
...
126	3F00H	3F7FH
127	3F80H	3FFFH

13.3 寄存器

● EXT_ADR 扩展地址寄存器 (S:D9H)

BIT	7	6	5	4	3	2	1	0
FIELD	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
RESET	0	0	0	0	0	0	0	0

用于寄存器组寄存器地址的选取。

● EXT_DAT 扩展数据寄存器 (S:D8H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

根据 EXT_ADR 的地址所对应的寄存器组的寄存器, 写入数据或是读出数据。

● EBYTE_ADR_L 数据存储区起始地址低位寄存器 (EXT_ADR = E0H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

16K 数据存储区页地址低 8 位；

● EBYTE_ADR_H 数据存储区起始地址高位寄存器 (EXT_ADR = E1H)

BIT	7	6	5	4	3	2	1	0
FIELD		D5	D4	RESERVE	D3	D2	D1	D0
RESET	—	0	0	-	0	0	0	0

16K 数据存储区页地址高 6 位；

● RAM_ADR_L RAM 起始地址寄存器 (EXT_ADR = E2H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

读、写 ROM 数据存储区时，该寄存器用于写入与 ROM 区域交互的外部 RAM 地址的低字节。

● RAM_N RAM 操作个数寄存器 (EXT_ADR = E3H)

BIT	7	6	5	4	3	2	1	0
FIELD		D6	D5	D4	D3	D2	D1	D0
RESET	—	0	0	0	0	0	0	0

读、写 ROM 数据存储区时，该寄存器用于写入与 ROM 区域交互的外部 RAM 字节数，最大字节数为 128。

● PGM/READ 编程/读使能寄存器 (EXT_ADR = E4H)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

当设定的数据为 01H 时，将启动编程模式，设定的数据为 10H 时，将启动读模式。当设定的数据为 11H 时，将读取芯片序列号，序列号长度 13 个字节。

● WP_REGISTER 写保护使能寄存器 (EXT_ADR = FFH)

BIT	7	6	5	4	3	2	1	0
FIELD	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0	0	0	0	0	0	0	0

向 WP_REGISTER 写入 FFH 时，打开写保护，写入其他数据时关闭写保护。当要对数据存储区进行读写操作时必须打开写保护。

13.4 应用例程

数据存储区写数据操作：

```

/*****
** *srt : 写入数据的首字节地址
** Addr_H: 数据存储区地址的高字节
** Addr_L: 数据存储区地址的低字节
** wLen : 写入数据长度
*****/
void WriteData_Flash(const unsigned char *srt, unsigned char Addr_H, unsigned char Addr_L, unsigned int wLen)
{
    unsigned int wCnt=0;
    DogWatch();
    EA=0;
    pRom=0x1E00; //pRom定义: unsigned char *pRom=(unsigned char *)0;
    for (wCnt=0;wCnt<wLen;wCnt++)
    {
        pRom[wCnt]=srt[wCnt];
    }
    EXT_ADR = 0xff;
    EXT_DAT = 0xff;
    EXT_ADR = 0xE0;
    EXT_DAT = Addr_L;
    EXT_ADR = 0xE1;
    EXT_DAT=(Addr_H&0x0F) | ((Addr_H<<1) &0x60);
    EXT_ADR = 0xE2;
    EXT_DAT = 0x00;
    EXT_ADR = 0xE3;
    EXT_DAT = wLen;
    EXT_ADR = 0xE4;
    EXT_DAT = 0x01;
    EXT_ADR = 0xff;
    EXT_DAT = 0x00;
    EA=1;
}

```

数据存储区读数据操作:

```
/*
*****
** *dst: 读出数据的首字节地址
** Addr_H: 数据存储区地址的高字节
** Addr_L: 数据存储区地址的低字节
** rLen: 读取数据长度
*****
void ReadData_Flash(unsigned char *dst, unsigned char Addr_H, unsigned char Addr_L, unsigned int rLen)
{
    unsigned int rCnt=0;
    DogWatch();
    EA=0;
    pRom=0x1E00; //pRom定义: unsigned char *xdata *pRom=(unsigned char *)0;
    EXT_ADR = 0xff;
    EXT_DAT = 0xff;
    EXT_ADR = 0xE0;
    EXT_DAT = Addr_L;
    EXT_ADR = 0xE1;
    EXT_DAT = (Addr_H&0x0F) | ((Addr_H<<1)&0x60);
    EXT_ADR = 0xE2;
    EXT_DAT = 0x00;
    EXT_ADR = 0xE3;
    EXT_DAT = rLen;
    EXT_ADR = 0xE4;
    EXT_DAT = 0x10;
    EXT_ADR = 0xff;
    EXT_DAT = 0x00;
    for(rCnt=0; rCnt<rLen; rCnt++)
    {
        dst[rCnt]=pRom[rCnt];
    }
    EA=1;
    DogWatch();
}
```

14 程序与数据的下载

PL5010 的嵌入式 MCU 是 8051 指令兼容的微处理器内核,所以用户在开发时只需使用 8051 的软件仿真/编译器即可(比如伟福、万利、TKS Studio 等)。将编译后的目标代码 HEX 文件通过晓程公司的下载软件 PL4000PRO.exe, 配合串行编程器下载到内的 FLASH 程序存储器中。

本公司提供的在系统编程下载器以及相关的 PC 机软件 PL4000PRO.exe。

注意: 芯片中一定要有固件程序, 当没有固件程序时, 用户主程序无法正常运行。

附录 1 特殊功能寄存器 (SFR) 表

REGISTER	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ADDRESS	ACCESS
P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	80H	R/W
SP									81H	R/W
DPL									82H	R/W
DPH									83H	R/W
PCON1	SMOD2	T1_SEL	T2_SEL	T3_SEL					86H	R/W
STATUS	SMOD	SMOD1	WDT	PU		LP_MOD	VBFO	PFI	87H	R⊕
PCON	SMOD	SMOD1	WDT	PU			STOP	IDLE	87H	W
TCON0	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88H	R/W
TMOD0	GATE	C/T	M1	M0	GATE	C/T	M1	M0	89H	R/W
TL0									8AH	R/W
TL1									8BH	R/W
TH0									8CH	R/W
TH1									8DH	R/W
CKCON	RST1	RST0	WDT2	WDT1	WDT0	CK2	CK1	CK0	8EH	R/W
WDT_RST	看门狗复位寄存器 (写入 A1H 喂狗)								8FH	R/W
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	90H	R/W
SCON0	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	98H	R/W
SBUF0									99H	R/W

P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	A0H	R/W	
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	A8H	R/W	
EIE	EDS	ESPI	ES4	ES3	EX3	ES2	ET3	EX2	A9H	R/W	
P3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	B0H	R/W	
P4	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	B4H	R/W	
IP		PS1	PT2	PS0	PT1	PX1	PT0	PX0	B8H	R/W	
EIP	PDS	PSPI	PS4	PS3	PX3	PS2	PT3	PX2	B9H	R/W	
SCON1	E1SM0	E1SM1	E1SM2	E1REN	E1TB8	E1RB8	EITI	E1RI	C0H	R/W	
SBUF1									C1H	R/W	
TCON1	TF3	TR3	TF2	TR2	IE3	IT3	IE2	IT2	C8H	R/W	
TMOD1	GATE	C/T	M1	M0	GATE	C/T	M1	M0	C9H	R/W	
TL2									CCH	R/W	
TH2									CDH	R/W	
TL3									CEH	R/W	
TH3									CFH	R/W	
PSW	CY	AC	F0	RS1	RS0	OV		P	D0H	R/W	
EXT_DAT②	D7	D6	D5	D4	D3	D2	D1	D0	D8H	R/W	
EXT_ADR③	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	D9H	R/W	
IR_CNT		38K 红外发送分频器								DBH	R/W
ACC									E0H	R/W	
ACC_H									E1H	R/W	
PMU_DAT④	D7	D6	D5	D4	D3	D2	D1	D0	E8H	R/W	

PMU_ADR ⑤	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	E9H	R/W
B									F0H	R/W
B_H									F1H	R/W
SCON2	E2SM0	E2SM1	E2SM2	E2REN	E2TB8	E2RB8	E2TI	E2RI	F8H	R/W
SBUF2									F9H	R/W

*注 1：STATUS 的 WDT 和 PU 标志位特殊可读可清 0

*注 2、3：EXT_ADR、EXT_DAT 寄存器为访问扩展寄存器使用的组合寄存器。EXT_ADR 用于送入扩展寄存器地址，EXT_DAT 用于送入扩展寄存器数据。

*注 4、5：PMU_ADR、PMU_DAT 寄存器为访问电能计量单元扩展寄存器使用的组合寄存器。PMU_ADR 用于送入扩展寄存器地址，PMU_DAT 用于送入扩展寄存器数据。

附录 2 扩展寄存器表 外部设备地址分配表

REGISTER	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ADDRESS	ACCESS	
Time_Adjust_Register	D7	D6	D5	D4	D3	D2	D1	D0	80H	R/W	
Second_Register		D6	D5	D4	D3	D2	D1	D0	81H	R/W	
Minute_Register		D6	D5	D4	D3	D2	D1	D0	82H	R/W	
Hour_Register			D5	D4	D3	D2	D1	D0	83H	R/W	
Week_Register						D2	D1	D0	84H	R/W	
Day_Register			D5	D4	D3	D2	D1	D0	85H	R/W	
Month_Register				D4	D3	D2	D1	D0	86H	R/W	
Year_Register	D7	D6	D5	D4	D3	D2	D1	D0	87H	R/W	
LP_CON	SEC_CNT3	SEC_CNT2	SEC_CNT1	SEC_CNT0	MOD	TIMER_FLAG	INT1_FLAG	INT2_FLAG	88H	R/W	
POWER_DOWNS	55H								89H	R/W	
PWM	FREQ[1:0]		Duty_Cycle[5:0]							8AH	R/W
VBAT_CON	ENA	SEL	ADJ5	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0	8BH	R	
VBAT_CNT						CNT2	CNT1	CNT0	8CH	R	

DI_7816									8DH	W
DO_7816									8EH	R
IO7816_CON	IOSEL		ERR_RST	C_INTP			IO_REN	IO_WEN	8FH	R/W
I_MAX_ETU [23:16]									90H	W
I_MAX_ETU [15:8]									91H	W
I_MAX_ETU [7:0]									92H	W
I_ETU[15:8]									93H	W
I_ETU[7:0]									94H	W
I_MIN_ETU [8]									95H	W
I_MIN_ETU [7:0]									96H	W
E_MAX_ETU [23:16]									97H	W
E_MAX_ETU [15:8]									98H	W
E_MAX_ETU [7:0]									99H	W
E_ETU[15:8]									9AH	W

E_ETU[7:0]									9BH	W
E_MIN_ETU [8]									9CH	W
E_MIN_ETU [7:0]									9DH	W
ISO7816_STA TUS	TIME_INA DQ	TIME_OV	ERR_INTP	RBYTE_R DY			W_PLR_E RR	WBYTE_R DY	9EH	R
FGEN	D7	D6	D5	D4	D3	D2	D1	D0	9FH	R/W
CommStatus Register0			FHF				CORR	SYNC	A0H	R
CommStatus Register1			FHF_zc			VFAIL	ZC_CORR	ZC_SYNC	A1H	R
Send_Leader_ Value		D6	D5	D4	D3	D2	D1	D0	A2H	R/W
CommControl Register	CCS_ENA	CCS_IE	CCS_IE_Z C	FH_CLR(W O)	SEND_MO D	RSSI_ENA	ZC_ENA	R/~T	A3H	R/W
CommFrameH Register	D15	D14	D13	D12	D11	D10	D9	D8	A4H	R/W
CommFrameL Register	D7	D6	D5	D4	D3	D2	D1	D0	A5H	R/W
CommFrame Detect							D1	D0	A6H	R/W
SSC_TXBUF	D7	D6	D5	D4	D3	D2	D1	D0	A7H	R/W

SSC_RXBUF0	D7	D6	D5	D4	D3	D2	D1	D0	A8H	R
SSC_RXBUF1	D7	D6	D5	D4	D3	D2	D1	D0	A9H	R
ZC_OFFSET	D7	D6	D5	D4	D3	D2	D1	D0	AAH	R/W
THRESHOLD_GAIN	D7	D6	D5	D4	D3	D2	D1	D0	ABH	R/W
PRE_THRESH_OLD	D7	D6	D5	D4	D3	D2	D1	D0	ACH	R/W
RSSI	D7	D6	D5	D4	D3	D2	D1	D0	ADH	R
MAX_CORR	D7	D6	D5	D4	D3	D2	D1	D0	AEH	R
ZC_MAX_CO RR	D7	D6	D5	D4	D3	D2	D1	D0	AFH	R
TIMER_CON	T4_ZERO	T4_SEL	T3_ZERO	T3_SEL	T2_ZERO	T2_SEL	T1_ZERO	T1_SEL	B0H	R/W
TIMER1_VAL									B1H	W
TIMER1_VAL									B2H	W
TIMER3_VAL									B3H	W
TIMER4_VAL									B4H	W
DIFF_CON	ENA	REC_ENA	ZC_ENA	MOD	TB8				B5H	R/W
DIFF_STATU S	TI	RI	SC_FLAG	RB8					B6H	R/W
DIFF_SBUF	D7	D6	D5	D4	D3	D2	D1	D0	B7H	W
DIFF_RBUF	D7	D6	D5	D4	D3	D2	D1	D0	B8H	R
PHASECNT	D7	D6	D5	D4	D3	D2	D1	D0	B9H	R

UART3_CON	ENA	REC_ENA	MOD	TB8					BAH	R/W
UART3_STAT US	TI	RI		RB8					BBH	R
UART3_DIV_ H			D13	D12	D11	D10	D9	D8	BCH	R/W
UART3_DIV_ L	D7	D6	D5	D4	D3	D2	D1	D0	BDH	R/W
UART3_SBUF									BEH	R/W
UART4_CON	ENA	REC_ENA	MOD	TB8					BFH	R/W
UART4_STAT US	TI	RI		RB8					C0H	R
UART4_DIV_ H			D13	D12	D11	D10	D9	D8	C1H	R/W
UART4_DIV_ L	D7	D6	D5	D4	D3	D2	D1	D0	C2H	R/W
UART4_SBUF	D7	D6	D5	D4	D3	D2	D1	D0	C3H	R/W
SPI_CON	SPE	CSMOD	CPOL	CPHA	SPR3	SPR2	SPR1	SPR0	C4H	R/W
SPI_STATUS	SPIF	WCOL							C5H	R
SPI_DATA	D7	D6	D5	D4	D3	D2	D1	D0	C6H	R/W
EPAGE_ADR						D2	D1	D0	E0H	W
EBYTE_ADR		D6	D5	D4	D3	D2	D1	D0	E1H	W
RAM_ADR_L	D7	D6	D5	D4	D3	D2	D1	D0	E2H	W

RAM_N		D6	D5	D4	D3	D2	D1	D0	E3H	W
PGM/READ	D7	D6	D5	D4	D3	D2	D1	D0	E4H	W
EXT_CTRL	TCKSEL				IC_DIV1	IC_DIV0	ESAM_DI V1	ESAM_DI V0	FDH	R/W
EXT_CFG	TCK_ENA					FGEN_EN A	IR38K_EN A	PWM_EN A	FEH	R/W
Write Protect Register	数据为 FFH 时，打开写保护，否则将关闭写保护或写保护不能被打开								FFH	W

注：访问本表寄存器需通过 EXT_ADR、EXT_DAT 两个寄存器访问。EXT_ADR 用于送入地址，EXT_DAT 用于送入数据。如：
EXT_ADR=0xFF；EXT_DAT=0xFF；为打开写保护功能。

注：对本表的寄存器进行写操作时，均要打开写保护！

附录 3 电能计量单元扩展寄存器表

REGISTER	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ADDRESS	ACCESS
Freq_LB									00H	R
Freq_HB									01H	R
IArms_LB									02H	R
IArms_HB									03H	R
Vrms_LB									04H	R
Vrms_HB									05H	R
IBrms_LB									06H	R
IBrms_HB									07H	R
IAAD_LB									08H	R
IAAD_HB									09H	R
UAD_LB									0AH	R
UAD_HB									0BH	R
IBAD_LB									0CH	R
IBAD_HB									0DH	R
Cosphi_LB									0EH	R
Cosphi_HB									0FH	R
IPA1									10H	R
IPA2									11H	R

IPA3									12H	R
IPA4									13H	R
IPB1									14H	R
IPB2									15H	R
IPB3									16H	R
IPB4									17H	R
PCNT									18H	R
Poff_A_LB									19H	R/W
Poff_A_HB									1AH	R/W
Poff_B_LB									1BH	R/W
Poff_B_HB									1CH	R/W
PAGAIN_LB									1DH	R/W
PAGAIN_HB									1EH	R/W
PBGAIN_LB									1FH	R/W
PBGAIN_HB									20H	R/W
Pgate_LB									21H	R/W
Pgate_HB									22H	R/W
QDGATE									23H	R/W
IBGAIN_LB									24H	R/W
IBGAIN_HB									25H	R/W
IADPC									26H	R/W
IBDPC									27H	R/W

IA_DPC_H									28H	R/W
IB_DPC_H									29H	R/W
CON_GAIN			IA_G		IB_G		U_G		2AH	R/W
OV_GATE	D7	D6	D5	D4	D3	D2	D1	D0	2BH	R/W
SAG_GATE	D7	D6	D5	D4	D3	D2	D1	D0	2CH	R/W
SAG_P	D7	D6	D5	D4	D3	D2	D1	D0	2DH	R/W
VPEAK	D7	D6	D5	D4	D3	D2	D1	D0	2EH	R
VQM	VPEAK_ENA					OV_FALG	SAG_FLAG	V_FAIL	2FH	R/W
IArms_off_LB									30H	R/W
IArms_off_HB									31H	R/W
IBrms_off_LB									32H	R/W
IBrms_off_HB									33H	R/W
CFCONFIG	CFSUEN	CFSU[1:0]							FAH	R/W
Status	RMS_OV	Z_FLAG	Z_DIR	LDE	PDIR	PCF	Mulit_ovf		FBH	R
Config	ENA	AC_DC	PIE	PMOD	ENA2	CHNSEL	FS_SEL		FCH	R/W
Iagc_cfg					I2_G1	I2_G0	I1_G1	I1_G0	FDH	R/W
Write Protect Register	数据为 FFH 时打开写保护，否则将关闭写保护或写保护不能被打开								FFH	W

注：访问本表寄存器需通过 PMU_ADR、PMU_DAT 两个寄存器访问。PMU_ADR 用于送入地址，PMU_DAT 用于送入数据。如：PMU_ADR=0xFF；PMU_DAT=0xFF；为打开写保护功能。

注：对本表的寄存器进行写操作时，均要打开写保护！

