

PHM320 MCU

用户手册

目录

1. 概述	3
2. 产品特性	3
3. 功能框图	4
4. 功能简介	4
4.1 LDO 功能模块	4
4.2 POR, LVR 功能模块	4
4.3 OSC16M 功能模块	4
4.4 PGA&ADC 功能模块	4
4.5 目标检测功能模块	5
4.6 CPU	5
4.7 内置 ROM	5
4.8 内置 RAM	5
4.9 IR 红外接收模块	5
4.10 TIMER	5
4.11 PWM	5
4.12 GPIO	5
5. 引脚定义及使用说明	6
5.1 引脚定义	6
5.2 Pinshare	7
5.3 引脚使用说明	7
6. 寄存器说明	8
6.1 LDO 功能寄存器	8
6.2 OSC16M 功能寄存器	9
6.3 PGA&ADC 功能寄存器	9
6.4 FDAC 功能寄存器	10
6.5 目标检测功能寄存器	11
6.6 IR 控制寄存器	13
6.7 TIMER 控制寄存器	13
6.8 PWM 控制寄存器	14
6.9 GPIO 功能寄存器	15
7 参考设计	17

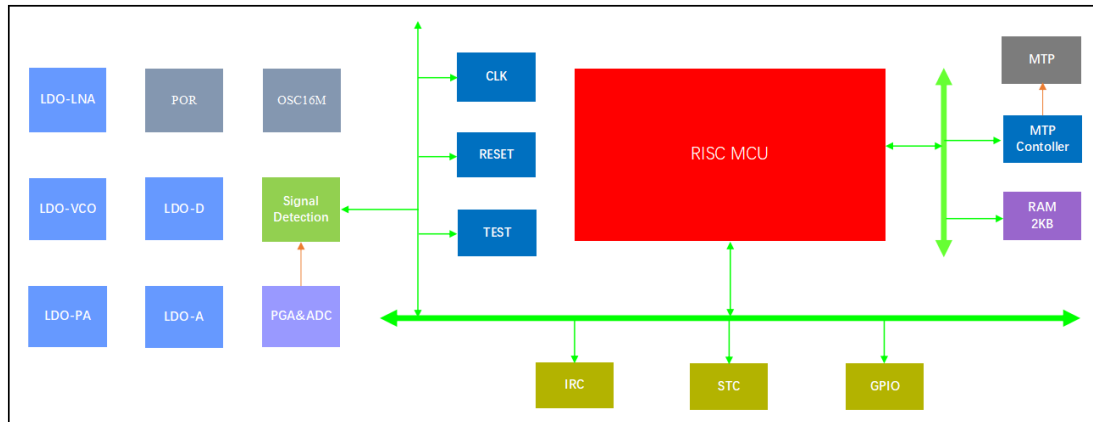
1. 概述

PHM320是一款基于RISC-V内核的32位通用微控制器，主要针对低成本、低功耗的MCU应用领域。

2. 产品特性

- 内置32位RISC-V核，工作频率可到16MHz 或 24MHz
- 内置4K字节的MTP ROM和64字节的EEPROM
- 内置2K字节的XRAM
- 集成了5路32位定时器，其中含有1路看门狗定时器
- 集成了1路12位SAR ADC,最高能达到1Mbps的采样率，搭配芯片内3选1的多路开关，可以外接3个通道的模拟源输入
- 最多支持8个GPIO
- 最多支持5路24位频率可调，脉宽可调的PWM输出
- 集成了红外遥控IR接收器
- 内置雷达中频信号数字处理和检测功能模块
- 内置了高精度的16MHz振荡器
- 内置了POR, LVR, LDOs, FDAC模块
- 内含2总线JTAG调试接口
- 2.97-3.63V的供电电压
- 工作温度范围：-40℃ to 105℃
- 封装：QFN24,4mm x 4mm

3. 功能框图



4. 功能简介

4.1 LDO 功能模块

芯片内共有 LDO-LNA, LDO-VCO, LDO-PA, LDO-D, LDO-A 5 个 LDO, 其中 LDO-LNA, LDO-VCO, LDO-PA 3 个 LDO 是给外部的雷达芯片供电用, 输出典型值 1.2V, 可以通过寄存器微调。LDO-D 是给芯片内部的数字部分电路用, 输出典型值 1.2V, 可以通过寄存器微调。LDO-A 是给芯片内部的模拟部分电路用, 输出典型值 1.2V。

4.2 POR, LVR 功能模块

芯片上电, 掉电复位电路, POR 电路能够很好的完成上电时序要求。LVR 防止掉电和突发电压下降时导致的 CPU 死机或跑飞的问题。

4.3 OSC16M 功能模块

内部高精度时钟振荡电路, 给 CPU, 总线等数字模块提供时钟。典型的输出频率是 16MHz, 可以通过寄存器调整输出频率, 不同的芯片具有一定的差异性。

4.4 PGA&ADC 功能模块

芯片内的 PGA 模块介于中频 IF 输入和 ADC 之间, 起到放大输入信号的作用, 通过寄存器配置可以选择 bypass PGA 或者不同的放大倍数。ADC 模块最高能达到 1Mbps 的采样率, 可以同寄存器修改相关参数, 把 IF 信号转换成数字信号, 进行目标检测。

4.5 目标检测功能模块

功能模块集成了移动目标检测的算法，通过寄存器配置，可以设置不同的参数来完成移动目标的检测，并产生相应的触发信号。

4.6 CPU

PHM320 集成了平头哥的 E902 CPU, 32 位 RISC-V 核，支持精简指令集，低功耗，工作频率典型为 16MHz

4.7 内置 ROM

PHM320 内部集成了 4K 字节的 MTP ROM 和 64 字节的 EEPROM，CPU 通过 MTP Control 对 MTP 进行指令操作和对 EEPROM 进行相关数据的操作。

4.8 内置 RAM

PHM320 内部集成了 2K 字节的 XRAM,跟 CPU 同频，可以进行快速的数据交互。

4.9 IR 红外接收模块

PHM320 内部集成了 IR 红外接收模块,能够解码从遥控接收头过来的信号,减少了 CPU 的占用时间，提升效率。

4.10 TIMER

芯片内置了 5 个 TIMER,其中一个看门狗 TIMER,4 个通用 TIMER,每个 TIMER 都有 32bit 来设置定时时间。每个 TIMER 都具有中断功能。

4.11 PWM

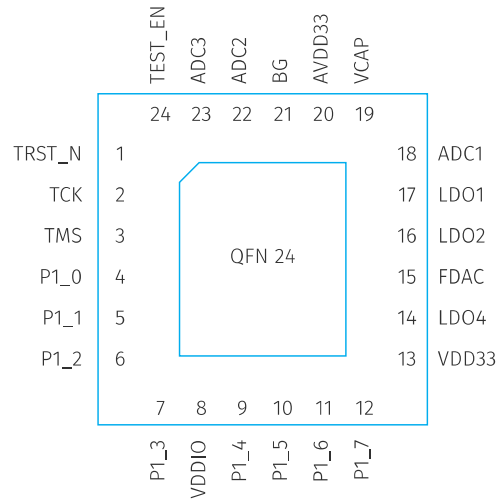
芯片内含有 5 路 PWM 控制器，通过 32bit 寄存器可以控制 PWM 输出不同的频率和占空比，能够实现脉宽调制功能。

4.12 GPIO

芯片一共支持 8 路 GPIO,分别是 P1_0 到 P1_7。这些 GPIO 具有 PINshare 功能，通过配置不同寄存器可以设置成不同功能的输入输出。

5. 引脚定义及使用说明

5.1 引脚定义



引脚	名称	类型	描述
1	TRST_N	I	JTAG 调试口复位信号
2	TCK	I	JTAG 调试口时钟信号
3	TMS	I/O	JTAG 调试口数据信号
4	P1_0	I/O	通用数字 IO
5	P1_1	I/O	通用数字 IO
6	P1_2	I/O	通用数字 IO
7	P1_3	I/O	通用数字 IO
8	VDDIO	V	数字 IO 模块 3.3V 电源输入
9	P1_4	I/O	通用数字 IO
10	P1_5	I/O	通用数字 IO
11	P1_6	I/O	通用数字 IO
12	P1_7	I/O	通用数字 IO
13	VDD33	V	内部 LDO 模块 3.3V 电源输入
14	LDO4	O	1.2V LDO 电压输出
15	FDAC	O	0.8-1.2V DAC 电压输出
16	LDO2	O	1.2V LDO 电压输出
17	LDO1	O	1.2V LDO 电压输出
18	ADC1	I	ADC 通道 1 输入
19	VCAP	I	VCAP 电压，外接电容到地
20	AVDD33	V	3.3V 模拟电源输入
21	BG	I	BG 电压，外接电容到地

22	ADC2	I	ADC 通道 2 输入
23	ADC3	I	ADC 通道 3 输入
24	TEST_EN	I	测试模式使能输入
25	GND	V	EPAD 引脚

5.2 Pinshare

PIN No.	PIN Name	Function 1	Function 2	Function 3	Function 4
4	P1_0	GPIO	PWM1		
5	P1_1	GPIO	PWM2		
6	P1_2	GPIO	PWM3		
7	P1_3	GPIO	PWM4		
9	P1_4	GPIO	PWM5		
10	P1_5	GPIO	IR	adc_done	osc_16m
11	P1_6	GPIO	adc_sample		

5.3 引脚使用说明

● TRST_N/TCK/TMS

2 线 JTAG 调试口，可以通过这个接口给 PHM320 下载程序，也可以通过这个口调试软件。3.3V 电平，配套专门的下载器一起工作。如果为了省对外的引脚，可以不连 TRST_N 这个 PIN，悬空。

● P1_0 到 P1_7

通用输入输出 IO 口，内部没有上下拉电路，3.3V 电平。其中 P1_0 到 P1_4 还可以设置成 PWM 输出。P1_5 可以设置成 IR 的输入信号,adc_done 和 osc_16m 的输出信号。P1_6 可以设置成 adc_sample 的输出信号。

● LDO1,LDO2,LDO4

是 3 路 1.2VLDO 输出引脚，每一路的输出电流最大不超过 30mA，可以直接跟 816，816C 的电源管脚连接，给它们供电。支持脉冲供电模式，其中频率受 ADC 采样频率控制，供电的脉冲宽度可以通过寄存器调整。

● FDAC

是电压可以调节的 DAC 输出，输出范围 0.8V-1.2V。直接接 816C 的 16 脚,通过设置不同电压微调 816C 的 RF 频率。

● ADC1 到 ADC3

模数转换输入，可以作为雷达芯片的 IF 信号输入，光敏传感器输入，温度传感器输入

等，输入电压范围根据芯片内部设置的不同参数而不同，最大支持到 3.3V 输入。

● VDDIO,VDD33,AVDD33,VCAP,BG

VDDIO:IO 模块的供电电压，3.3V 输入，靠近管脚加 1uF 电容到地；

VDD33:LDO1,LDO2,LDO4 的输入电压，3.3V 输入，靠近管脚加 1uF 电容到地；

AVDD33:内部 LDO-D,LDO-A 的输入电压，3.3V 输入，靠近管脚加 1uF 电容到地；

VCAP:VCAP 电压，管脚加 1uF 电容到地；

BG: BG 电压，管脚加 1uF 电容到地；

● TEST_EN

测试模式选择输入脚，正常模式直接接地。

6.寄存器说明

6.1 LDO 功能寄存器

0x1f000010	0xC48	R/W	rf4_en	[31]	rf ldo4 enable
			rflldo4_trim	[30:28]	rf ldo4 trim, rf ldo4 can't be controlled by adc_sample
			rf3_en	[27]	rf ldo3 enable
			rflldo3_trim	[26:24]	rf ldo3 trim
			rf2_en	[23]	rf ldo2 enable
			rflldo2_trim	[22:20]	rf ldo2 trim
			rf1_en	[19]	rf ldo1 enable
			rflldo1_trim	[18:16]	rf ldo1 trim
			ldo_dig_trim	[12:11]	1.2V ldo trim
			iref_trim	[10:8]	ireference trim

bit [31], bit [27], bit [23], bit [19] 是 LDO4-LDO1 的开关使能位，1 为使能开；

bit [30: 28], bit [26: 24], bit [22: 20], bit [18: 16] 是 LDO4-LDO1 输出电压微调位；

bit [12: 11] 是数字部分 LDO 的输出电压微调位；

其中 LDO4 是不受 ADC 采样控制的，在脉冲供电的时候输出电压是一直存在的。其他 LDO3-LDO1 是受 ADC 采样控制，输出脉冲电压。

bit [10: 8] 是内部模块所使用的参考电流的调整位，可以微调参考电流的大小；

0x1f000014	0x010101	R/W	rf3_delay	[22:16]	rf ldo3 startup delay (after delay count, RF will work) , PA ctrl
			rf2_delay	[14:8]	rf ldo2 startup delay (after delay count, RF will work), VCO ctrl

			rf_en_sel	[7]	rf ldo enable select, 0->adc sample enable ldo, 1->by reg rfx_en12 setting
			rf1_delay	[6:0]	rf ldo1 startup delay (after delay count, RF will work), LNA ctrl

bit 【22: 16】，bit 【14: 8】，bit 【6: 0】，是 LDO3-LDO1 开启输出延时，意思是在脉冲供电输出时，adc 过来的开启 LDO 的控制信号需要延时一段时间才能真正的开启 LDO，保证 ADC 已经准备完毕。影响脉冲宽度，具体是：供电脉宽=adc_sample_cnt-rf_delay;

bit 【7】是脉冲和正常供电的选择位，1 是正常供电，0 是脉冲供电;

6.2 OSC16M 功能寄存器

0x1f000010	0xC48	R/W	osc16m_trim	[7:0]	rc osc frequency trim
------------	-------	-----	-------------	-------	-----------------------

bit 【7: 0】是调整内部 16MHz 频率，一共有 8 个 bit 来设置，可以调整频率从 7MHz 左右到 26MHz 左右;

6.3 PGA&ADC 功能寄存器

PGA 部分:

0x1f000014	0x010101	R/W	pga_en	[31]	PGA enable
			pga_bypass	[30]	PGA bypass(1-bypass)
			pga_dc_trim	[29:24]	PGA output dc level setting

bit 【31】: PGA 功能开关使能位，1 使能，0 关闭;

bit 【30】: PGA bypass 开关使能位，1 bypass PGA;

bit 【29: 24】: PGA 输出的直流电平的调整位，一共有 8 个 bit 来调整，根据 PGA 输入信号的不同，调节 PGA 的直流电平输出来满足应用需求;

0x1f000018	0x80088200	R/W	pga_gain	[27:24]	PGA gain ctrl(0~7 ->X1/X1.5/X2/X3/X4/X6/X8/X12)
------------	------------	-----	----------	---------	---

bit 【27: 24】: PGA 放大倍数的控制位，一共有 8 个档位可以调整;

ADC 部分:

0x1f000018	0x80088200	R/W	adc_sample_cnt	[23:16]	adc sample time(base on 16M osc clock count, RF work time = adc_sample_cnt - rfX_delay)
------------	------------	-----	----------------	---------	---

			adc_vref_trim	[15:12]	adc reference trim
			adc_vcm_trim	[11:10]	adc vcom trim
			adc_vref_sel	[9:8]	adc reference select, 00->2V, 01->2.5V, 10->3V, 11->VDD
			adc_clk_sel	[7]	adc clock select, 0-> from digital clk, 1-> analog
			adc_clk_div	[6]	adc clock divide from analog clk
			adc_data_clrn	[5]	adc data clear, 0 ->clear
			adc_mode	[4]	adc mode, 0-> trigger mode, 1-> continues sample mode
			adc_en	[3]	ADC enable
			adc_ch_mux	[2:0]	adc input channel select, (000-ADC1(IF), 001~100: ADC2~ADC5, 101-VDD/2, 110-VCM, 111-pulse)

bit 【23: 16】: ADC 采样脉冲宽度的设置位, 不能设置太小, 在脉冲供电时和 RF_DELAY 一起确定脉冲供电的脉冲宽度;

bit 【15: 12】: ADC ref 电压的微调位, 可以调整 ref 电压的精准度;

bit 【11: 10】: ADC vcom 电压的微调位, 可以调整 vcom 电压的精准度;

bit 【9: 8】: ADC ref 电压的档位选择位, 一共有 4 个档位可以选择;

bit 【7】: ADC 模块的时钟选择位;

bit 【6】: ADC 模块的时钟分频选择位;

bit 【5】: ADC 数据清除开关位;

bit 【4】: ADC 工作模式选择位, 0 为触发模式, 1 为连续采样模式;

bit 【3】: ADC 模块的开关使能位, 1 为打开 ADC;

bit 【2: 0】: ADC 输入通道选择位, 001-100 选择不同通道, 其他测试用;

0x1f000014	0x010101	R/W	adc_save_pow	[15]	power down after adc_done
------------	----------	-----	--------------	------	---------------------------

bit 【15】: ADC 采样完成后关闭功能, 为了节省功耗使用, 1 为使能此功能;

6.4 FDAC 功能寄存器

0x1f000018	0x80088200	R/W	freq	[31:28]	VCO freq trim
------------	------------	-----	------	---------	---------------

bit 【31: 28】: 调整 FDAC 的输出电压, 范围为: 0.8V-1.2V, 可以用这个电压来校正 816C 的频率;

6.5 目标检测功能寄存器

● 设置 ADC 的采样频率

0x1f00001C	0x1F4	R/W	top_ctl4	[11:0]	adc smaple clk divider
------------	-------	-----	----------	--------	------------------------

bit 【11: 0】: 设置 ADC 采样频率的第一级分频, 是从 16MHz 频率分下来, 比如现在默认设置的是 0x1F4, 十进制是 500, 第一级分频后的频率是: $16000K/500=32K$;

0x1f000024	0x18402	R/W	bb_time_ctl	[7:0]	sample rate divider, minimum 2
------------	---------	-----	-------------	-------	--------------------------------

bit 【7: 0】: 设置 ADC 采样频率的第二级分频, 是从第一级频率分下来, 比如现在默认设置的是 0x02, 十进制是 2, 第二级分频后的频率是: $32K/2=16K$;

0x1f000020	0x0	R/W	top_ctl5	[17:16]	adc_triger select 1x: bit 16 0x: adc_triger_bb
------------	-----	-----	----------	---------	---

bit 【17: 16】: adc 触发源的选择, 在默认状态下选用的是目标检测模块过来的触发信号;

● 设置门限

0x1f000028	0x26C	R/W		[31:12]	Reserved
			bb_adc_dc_init	[11:0]	adc dc initial value for manual mode

bit 【11: 0】: 设置手动模式下的直流电平的初始值, 一共有 12 个 bit;

0x1f00002C	0x20000	R/W		[31:26]	Reserved
			bb_adc_thresh	[25:0]	threshold for ac detection

bit 【25: 0】: 设置目标检测的门限, 高于这个门限, 触发有效, 一共有 26 个 bit;

0x1f000030	0x10000	R/W		[31:26]	Reserved
			bb_adc_noise	[25:0]	noise level

bit 【25: 0】: 设置噪声更新的门限, 一共有 26 个 bit;

● 设置触发延时和保护时间

0x1f000034	0xEA600	R/W		[31:24]	Reserved
			t1_value	[23:0]	t1 value, io output high cnt @32KHz

bit 【23: 0】: 触发输出的延时, 也就是从触发输出开始经过多长时间后关闭触发输出, 一共有 24 个 bit, 用的是第一级 adc 分频后的时钟来计算;

0x1f000038	0x17700	R/W		[31:24]	Reserved
			t2_value	[23:0]	t2 value, io output low cnt @32KHz

bit 【23: 0】: 触发输出的保护延时, 也就是从关闭触发输出后开始经过多长时间后才能再触发输出有效, 一共有 24 个 bit, 用的是第一级 adc 分频后的时钟来计算;

● 设置目标检测其他参数

0x1f000024	0x18402	R/W	bb_time_ctl	[31:19]	Reserved
				[18:17]	dc average dynamic caculate time select 00:4 cycle 01:8 cycle 10:16 cycle 11:32 cycle
				[16:15]	dc average initial caculate time select 00:4 cycle 01:8 cycle 10:16 cycle 11:32 cycle
				[14]	bb proc enable 1:enable
				[13]	dc average select 0:auto 1>manual
				[12]	bb wakeup output enable 1:enable
				[11]	bb wakeup select 1:io triger 0:update_en
				[10:8]	total sum samples setting 0:1x256 1:2x256 2:4x256 3:8x256 4:16x256 5:32x256 6&7:64x256

bit 【18: 17】: 平均直流电平动态更新计算的时间选择, 有 4 个时间周期可以选择;

bit 【16: 15】: 平均直流电平初始值计算的时间选择, 有 4 个时间周期可以选择;

bit 【14】: 目标检测功能开关使能位, 1, 打开目标检测功能;

bit 【13】: 平均直流电平的选择位: 1, 自动更新。2, 手动设置;

bit 【12】: 目标检测到后唤醒输出的使能位: 1, 使能有效;

bit 【11】: 目标检测唤醒方式选择: 1, 选择 IO 触发, 2, 选择 updata_en 触发

bit 【10: 8】: 采样积分周期的选择: 一共有 7 个选项可以选择;

6.6 IR 控制寄存器

0x1f03000c	0x0	RO	ir_data	[31:0]	ir read data
------------	-----	----	---------	--------	--------------

读到的 IR 的数据，一共 4 个 byte，其中高 2 个 BYTE 是地址码，低 2 个 BYTE 是命令码；

0x1f030000	0x0	RO	ir_repeat	[1]	ir get a repeat command
	0x0	RO	ir_int	[0]	ir get a data or repeat signal

读到的是单次按键还是重复键，bit **【0】**：只要接收到遥控按键就置 1，不管是单次还是重复，bit **【1】**：重复键置为 1，通过 bit **【0】** 和 bit **【1】** 组合来判断是什么键；

0x1f030008	0x0	WC	rf_int_clr	[0]	write to this bit can clear int signal
------------	-----	----	------------	-----	--

清除接收到的按键中断标志；
其它寄存器就使用默认的设置；

6.7 TIMER 控制寄存器

0x1f020000	0x0	R/W	timer_rel[3]	[11]	timer 3 int clear, write clear
	0x0	R/W	timer_rel[2]	[10]	timer 2 int clear, write clear
	0x0	R/W	timer_rel[1]	[9]	timer 1 int clear, write clear
	0x0	R/W	timer_rel[0]	[8]	timer 0 int clear, write clear
	0x0	R/W	timer_en[3]	[7]	timer 3 enable
	0x0	R/W	timer_en[2]	[6]	timer 2 enable
	0x0	R/W	timer_en[1]	[5]	timer 1 enable
	0x0	R/W	timer_en[0]	[4]	timer 0 enable
	0x0	R/W	wdg_rst_en	[2]	watch dog reset enable
	0x0	R/W	wdg_en	[1]	watch dog counter enable

bit **【11】**，bit **【10】**，bit **【9】**，bit **【8】**：TIMER 3-TIMER 0 清中断位，置 1 清掉中断状态；
bit **【7】**，bit **【6】**，bit **【5】**，bit **【4】**：TIMER 3-TIMER 0 使能位，置 1 使能 TIMER
bit **【2】**：看门狗定时器复位使能；置 1 使能；
bit **【1】**：看门狗定时器计数使能；置 1 使能；

0x1f020008	0xFFFF	R/W	Timer0_reg	[31:0]	timer 0 initial value, decrease count
------------	--------	-----	------------	--------	---------------------------------------

0x1f02000C	0xFFFF	R/W	Timer1_reg	[31:0]	timer 1 initial value, decrease count
0x1f020010	0xFFFF	R/W	Timer2_reg	[31:0]	timer 2 initial value, decrease count
0x1f020014	0xFFFF	R/W	Timer3_reg	[31:0]	timer 3 initial value, decrease count

TIMER 0-TIMER 3 初始状态计数值的设置，向下计数，每个通道有 32 个 bit 可以设置，以 16MHz 作为计数时钟；

0x1f020004	0x0	R/W	wdg_cnt_reg	[31:0]	watch dog counter initial value
------------	-----	-----	-------------	--------	---------------------------------

看门狗定时器的初始状态计数值的设置，向下计数，32 个 bit 可以设置，以 16MHz 作为计数时钟；

0x1f020018	0x0	RO	Timer0_cnt	[31:0]	timer 0 counter value
0x1f02001C	0x0	RO	Timer1_cnt	[31:0]	timer 1 counter value
0x1f020020	0x0	RO	Timer2_cnt	[31:0]	timer 2 counter value
0x1f020024	0x0	RO	Timer3_cnt	[31:0]	timer 3 counter value

TIMER 0-TIMER 3 读到的当前计数值

6.8 PWM 控制寄存器

0x1f020000	0x0	R/W	pwm_en[4]	[16]	PWM 4 counter enable
	0x0	R/W	pwm_en[3]	[15]	PWM 3 counter enable
	0x0	R/W	pwm_en[2]	[14]	PWM 2 counter enable
	0x0	R/W	pwm_en[1]	[13]	PWM 1 counter enable
	0x0	R/W	pwm_en[0]	[12]	PWM 0 counter enable

bit 【16】，bit 【15】，bit 【14】，bit 【13】，bit 【12】：PWM 4-PWM 0 计数器使能位，置 1 使能 PWM 功能；

0x1f020040				[31:24]	Reserved
	0x0	R/W	pwm_freq_ctrl1	[23:0]	PWM 0 frequency setting
0x1f020044				[31:24]	Reserved
	0x0	R/W	pwm_freq_ctrl2	[23:0]	PWM 1 frequency setting
0x1f020048				[31:24]	Reserved
	0x0	R/W	pwm_freq_ctrl3	[23:0]	PWM 2 frequency setting
0x1f02004C				[31:24]	Reserved
	0x0	R/W	pwm_freq_ctrl4	[23:0]	PWM 3 frequency setting
0x1f020050				[31:24]	Reserved
	0x0	R/W	pwm_freq_ctrl5	[23:0]	PWM 4 frequency setting

PWM 0-PWM 4 的输出频率的设定，一共有 24 个 bit 来设置，以 16MHz 作为计数时钟；

0x1f020054				[31:24]	Reserved
	0x0	R/W	pwm_freq_duty1	[23:0]	PWM 0 duty setting
0x1f020058				[31:24]	Reserved
	0x0	R/W	pwm_freq_duty2	[23:0]	PWM 1 duty setting
0x1f02005C				[31:24]	Reserved
	0x0	R/W	pwm_freq_duty3	[23:0]	PWM 2 duty setting
0x1f020060				[31:24]	Reserved
	0x0	R/W	pwm_freq_duty4	[23:0]	PWM 3 duty setting
0x1f020064				[31:24]	Reserved
	0x0	R/W	pwm_freq_duty5	[23:0]	PWM 4 duty setting

PWM 0-PWM 4 的输出占空比的设定，一共有 24 个 bit 来设置，以 16MHz 作为计数时钟；

6.9 GPIO 功能寄存器

0x1f00001C	0x1F4	R/W	top_ctl4	[23]	P1_6 control 1:adc_sample 0:GPIO
				[22:21]	P1_5 control 11:osc_16m 10:adc done 01:IR 00:GPIO
				[20]	P1_4 control 1:PWM4_IO output 0:GPIO
				[19]	P1_3 control 1:PWM3_IO output 0:GPIO
				[18]	P1_2 control 1:PWM2_IO output 0:GPIO
				[17]	P1_1 control 1:PWM1_IO output 0:GPIO
				[16]	P1_0 control 1:PWM0_IO output 0:GPIO

bit 【23】: P1_6 管脚的功能选择: 1, adc 采样信号的输出; 0, GPIO 口功能;
bit 【22: 21】: P1_5 管脚的功能选择: 11, osc_16m 信号的输出; 10, adc done 信号的输出;
01, IR 红外遥控信号输入; 00, GPIO 口功能;
bit 【20】: P1_4 管脚的功能选择: 1, PWM4_IO 信号的输出; 0, GPIO 口功能;
bit 【19】: P1_3 管脚的功能选择: 1, PWM3_IO 信号的输出; 0, GPIO 口功能;
bit 【18】: P1_2 管脚的功能选择: 1, PWM2_IO 信号的输出; 0, GPIO 口功能;
bit 【17】: P1_1 管脚的功能选择: 1, PWM1_IO 信号的输出; 0, GPIO 口功能;
bit 【16】: P1_0 管脚的功能选择: 1, PWM0_IO 信号的输出; 0, GPIO 口功能;

0x1f000020	0x0	R/W	top_ctl5	[13:12]	PWM4_IO output select 10:io_value 11:io_value inv 0x:PWM4
				[11:10]	PWM3_IO output select 10:io_value 11:io_value inv 0x:PWM3
				[9:8]	PWM2_IO output select 10:io_value 11:io_value inv 0x:PWM2
				[7:6]	PWM1_IO output select 10:io_value 11:io_value inv 0x:PWM1
				[5:4]	PWM0_IO output select 10:io_value 11:io_value inv 0x:PWM0

bit 【13: 12】: PWM4_IO 输出功能选择: 10, 目标检测触发信号信号高电平输出; 11, 目标检测触发信号信号低电平输出; 0x, PWM4 功能输出;
bit 【11: 10】: PWM4_IO 输出功能选择: 10, 目标检测触发信号信号高电平输出; 11, 目标检测触发信号信号低电平输出; 0x, PWM4 功能输出;
bit 【9: 8】: PWM4_IO 输出功能选择: 10, 目标检测触发信号信号高电平输出; 11, 目标检测触发信号信号低电平输出; 0x, PWM4 功能输出;
bit 【7: 6】: PWM4_IO 输出功能选择: 10, 目标检测触发信号信号高电平输出; 11, 目标检测触发信号信号低电平输出; 0x, PWM4 功能输出;
bit 【5: 4】: PWM4_IO 输出功能选择: 10, 目标检测触发信号信号高电平输出; 11, 目标检测触发信号信号低电平输出; 0x, PWM4 功能输出;

0x1f0000C0				[31:8]	Reserved
	0x0	R/W	gpio_out	[7:0]	write:p1_out read:p1_in

bit 【7: 0】: GPIO P1_0-P1_7 输入输出时的数据, 1 为输出或输入高电平; 0 为输出或输入低电平;

0x1f0000C4				[31:24]	Reserved
	0xFF	R/W	gpio_oen	[7:0]	p1_oen

bit【7: 0】: GPIO P1_0-P1_7 输出使能设置, 1 为输入使能, 作为输入模式; 0 为输出使能, 作为输出模式;

7 参考设计

