



**SolarA<sup>2</sup>**

## **从 3065 到 3066M/3065P 芯片移植开发指南**

文档版本 01

发布日期 2024-12-20

版权所有 © 海思技术有限公司2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



**HISILICON**、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 海思技术有限公司

地址：上海市青浦区虹桥港路2号101室 邮编：201721

网址：<https://www.hisilicon.com/cn/>

客户服务邮箱：[support@hisilicon.com](mailto:support@hisilicon.com)



# 前言

## 概述

本文档详细描述了3066M/3065P系列和3065H/3061H系列SolarA<sup>2</sup>软件包的区别，并提供从3065H/3061H系列芯片到3066M/3065P芯片的移植开发指南。

## 产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
SolarA2	1.1.0




## 读者对象

本文档主要适用于升级的操作人员。

- 技术支持工程师
- 软件开发工程师

## 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 <b>危险</b>	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 <b>警告</b>	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 <b>注意</b>	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。



符号	说明
<div>须知</div>	用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
<div>说明</div>	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

## 修订记录

修订日期	版本	修订说明
2024-08-23	00B01	第1次临时版本发布。
2024-12-20	01	第1次正式版本发布。



# 目录

前言.....	i
1 简介与使用范围.....	1
2 模块移植.....	4
2.1 地址空间映射对比.....	4
2.2 中断特性对比.....	8
2.3 Flash 特性对比.....	9
2.4 ADC 特性对比.....	10
2.5 DMA 特性对比.....	11
2.6 CRC 特性对比.....	12
2.7 WDG 特性对比.....	12
2.8 SPI 特性对比.....	13
2.9 GPT 特性对比.....	14
2.10 UART 特性对比.....	14
2.11 CMM 特性对比.....	16
2.12 CFD 特性对比.....	16
2.13 TIMER 特性对比.....	16
2.14 CAPM 特性对比.....	17
2.15 QDM 特性对比.....	18
2.16 CRG 特性对比.....	18
2.17 I2C 特性对比.....	20
2.18 ACMP 特性对比.....	20
2.19 DAC 特性对比.....	21
2.20 PGA 特性对比.....	21
2.21 APT 特性对比.....	22
2.22 CAN 特性对比.....	23
2.23 GPIO 特性对比.....	24
2.24 PMC 特性对比.....	24
2.25 IOCMG 特性对比.....	24
3 软件包移植.....	26
3.1 ADC 驱动移植.....	26
3.1.1 接口移植.....	26
3.1.2 配置参数变更.....	29



3.1.2.1 SOC_Param.....	29
3.2 DMA 驱动移植.....	30
3.2.1 接口移植.....	31
3.2.2 配置参数变更.....	32
3.3 CRC 驱动移植.....	32
3.3.1 接口移植.....	33
3.3.2 配置参数变更.....	34
3.4 WDG 驱动移植.....	34
3.4.1 接口移植.....	34
3.4.2 配置参数变更.....	36
3.5 SPI 驱动移植.....	36
3.5.1 接口移植.....	37
3.5.2 配置参数变更.....	39
3.6 GPT 驱动移植.....	39
3.6.1 接口移植.....	39
3.6.2 配置参数变更.....	41
3.7 UART 驱动移植.....	41
3.7.1 接口移植.....	41
3.7.2 配置参数变更.....	44
3.8 CMM 驱动移植.....	44
3.8.1 接口移植.....	45
3.8.2 配置参数变更.....	45
3.9 CFD 驱动移植.....	46
3.9.1 接口移植.....	46
3.9.2 配置参数变更.....	46
3.10 TIMER 驱动移植.....	47
3.10.1 接口移植.....	47
3.10.2 配置参数变更.....	48
3.11 CAPM 驱动移植.....	48
3.11.1 接口移植.....	48
3.11.2 配置参数变更.....	49
3.12 QDM 驱动移植.....	49
3.12.1 接口移植.....	50
3.12.2 配置参数变更.....	51
3.13 Flash 驱动移植.....	51
3.13.1 接口移植.....	51
3.13.2 配置参数变更.....	52
3.14 CRG 驱动移植.....	52
3.14.1 接口移植.....	53
3.14.2 配置参数变更.....	54
3.15 GPIO 驱动移植.....	54
3.15.1 接口移植.....	54



3.15.2 参数配置变更.....	56
3.16 PMC 驱动移植.....	56
3.16.1 接口移植.....	56
3.16.2 参数配置变更.....	57
3.17 I2C 驱动移植.....	57
3.17.1 接口移植.....	57
3.17.2 参数配置变更.....	60
3.18 ACMP 驱动移植.....	60
3.18.1 接口移植.....	60
3.18.2 参数配置变更.....	61
3.19 DAC 驱动移植.....	61
3.19.1 接口移植.....	62
3.19.2 参数配置变更.....	62
3.20 PGA 驱动移植.....	62
3.20.1 接口移植.....	62
3.20.2 参数配置变更.....	63
3.21 APT 驱动移植.....	63
3.21.1 接口移植.....	63
3.21.2 参数配置变更.....	67
3.22 CAN 驱动移植.....	67
3.22.1 接口移植.....	67
3.22.2 参数配置变更.....	68
3.23 IOCMG 驱动移植.....	68
3.23.1 接口移植.....	68
3.23.2 参数配置变更.....	70
3.24 SMBUS 驱动移植.....	70
3.24.1 接口移植.....	71
3.24.2 参数配置变更.....	74
<b>4 参考资料.....</b>	<b>75</b>
<b>A 缩略语.....</b>	<b>76</b>



## 表格目录

表 1-1 芯片系列对照表.....	1
表 2-1 地址空间映射对比.....	4
表 2-2 3066M/3065P 系列和 3065H/3061H 系列中断差异.....	8
表 2-3 3066M/3065P 系列和 3065H/3061H 系列 Flash 差异.....	9
表 2-4 3066M/3065P 系列和 3065H/3061H 系列 ADC 差异.....	10
表 2-5 3066M/3065P 系列和 3065H/3061H 系列 DMA 差异.....	11
表 2-6 3066M/3065P 系列和 3065H/3061H 系列 CRC 差异.....	12
表 2-7 3066M/3065P 的 IWDG 和 3065H/3061H 的 WDG 差异.....	12
表 2-8 3066M/3065P 的 WWDG 和 3065H/3061H 的 WDG 差异.....	13
表 2-9 3066M/3065P 系列和 3065H/3061H 系列 SPI 差异.....	13
表 2-10 3066M/3065P 系列和 3065H/3061H 系列 GPT 差异.....	14
表 2-11 3066M/3065P 系列和 3065H/3061H 系列 UART 差异.....	14
表 2-12 3066M/3065P 系列和 3065H/3061H 系列 CMM 差异.....	16
表 2-13 3066M/3065P 系列和 3065H/3061H 系列 CFD 差异.....	16
表 2-14 3066M/3065P 系列和 3065H/3061H 系列 TIMER 差异.....	16
表 2-15 3066M/3065P 系列和 3065H/3061H 系列 CAPM 差异.....	17
表 2-16 3066M/3065P 系列和 3065H/3061H 系列 QDM 差异.....	18
表 2-17 3066M/3065P 系列和 3065H/3061H 系列 CRG 差异.....	18
表 2-18 3066M/3065P 系列和 3065H/3061H 系列 I2C 差异.....	20
表 2-19 3066M/3065P 系列和 3065H/3061H 系列 ACMP 差异.....	20
表 2-20 3066M/3065P 系列和 3065H/3061H 系列 DAC 差异.....	21
表 2-21 3066M/3065P 系列和 3065H/3061H 系列 PGA 差异.....	21
表 2-22 3066M/3065P 系列和 3065H/3061H 系列 APT 差异.....	22
表 2-23 3066M/3065P 系列和 3065H/3061H 系列 CAN 差异.....	23
表 2-24 3066M/3065P 系列和 306x 系列 PMC 差异.....	24
表 2-25 3066M/3065P 系列和 3065H/3061H 系列 IOCMG 差异.....	24
表 3-1 接口移植对照表.....	26
表 3-2 SOC_Param 参数差异点对比.....	30
表 3-3 3066M/3065P 系列和 3065H/3061H 系列 DMA 接口对比.....	31
表 3-4 3066M/3065P 系列和 3065H/3061H 系列 CRC 接口对比.....	33
表 3-5 3066M/3065P 系列 IWDG 和 3065H/3061H 系列 IWDG 接口对比.....	34
表 3-6 3066M/3065P 系列 WWDG 和 3065H/3061H 系列 WDG 接口对比.....	35
表 3-7 3066M/3065P 系列和 3065H/3061H 系列 SPI 接口对比.....	37





表 3-8 3066M/3065P 系列和 3065H/3061H 系列 GPT 接口对比.....	39
表 3-9 3066M/3065P 系列和 3065H/3061H 系列 UART 接口对比.....	41
表 3-10 3066M/3065P 系列和 3065H/3061H 系列 CMM 接口对比.....	45
表 3-11 3066M/3065P 系列和 3065H/3061H 系列 CFD 接口对比.....	46
表 3-12 3066M/3065P 系列和 3065H/3061H 系列 TIMER 接口对比.....	47
表 3-13 3066M/3065P 系列和 3065H/3061H 系列 CAPM 接口对比.....	48
表 3-14 3066M/3065P 系列和 3065H/3061H 系列 QDM 接口对比.....	50
表 3-15 3066M/3065P 系列和 3065H/3061H 系列 Flash 接口对比.....	51
表 3-16 3066M/3065P 系列和 3065H/3061H 系列 CRG 接口对比.....	53
表 3-17 3066M/3065P 系列和 3065H/3061H 系列 GPIO 接口对比.....	54
表 3-18 3066M/3065P 系列和 3065H/3061H 系列 PMC 接口对比.....	56
表 3-19 3066M/3065P 系列和 3065H/3061H 系列 I2C 接口对比.....	57
表 3-20 3066M/3065P 系列和 3065H/3061H 系列 ACMP 接口对比.....	60
表 3-21 3066M/3065P 系列和 3065H/3061H 系列 DAC 接口对比.....	62
表 3-22 3066M/3065P 系列和 3065H/3061H 系列 PGA 接口对比.....	62
表 3-23 3066M/3065P 系列和 3065H/3061H 系列 APT 接口对比.....	63
表 3-24 3066M/3065P 系列和 3065H/3061H 系列 CAN 接口对比.....	67
表 3-25 3066M/3065P 系列和 3065H/3061H 系列 IOCMG 接口对比.....	68
表 3-26 3066M/3065P 系列和 3065H/3061H 系列 SMBus 接口对比.....	71
表 A-1 缩略语.....	76



# 1 简介与使用范围

3066M/3065P系列产品基于高性能RISC-V CPU核，工作频率最高到200MHz，集成了FPU（Floating Point Unit）浮点处理单元，支持浮点乘法、除法和开方等复杂数学运算指令，8个物理内存保护单元（PMP，Physical Memory Protection），支持硬件中断嵌套机制，中断优先级从0 ~ 15。支持64KB的SRAM（Static Random Access Memory）和最高512KB的Flash 存储单元。该MCU（Microcontroller Unit）集成最多9组高级PWM（Pulse Width Modulation）定时器，最多可支持2路独立电机PWM控制和3路PFC（Power Factor Correction）控制应用。内置3个12bit 3MSPS ADC（Analog Digital Converter）、3个10bit DAC（Digital to Analog Conversion）、4个正交解码器、3个比较器和3个PGA（Programmable Gain Amplifier）放大器。最多支持2个SPI（Serial Peripheral Interface）、5个UART（Universal Asynchronous Receiver Transmitter）、1个I2C（Inter-Integrated Circuit）等通讯接口、2个看门狗模块、4个基本Timer、1个系统Timer，支持多达73个通用GPIO（General-purpose input/output）管脚。

3065H/3061H系列产品基于高性能RISC-V CPU核，工作频率最高到200MHz，集成了FPU浮点处理单元，支持浮点乘法，除法和开方等复杂数学运算指令，16个物理内存保护单元，中断优先级从0 ~ 7，支持16KB的SRAM和最高160KB的Flash存储单元。该MCU集成最多9组高级PWM定时器，最多可支持两路独立电机PWM控制和两路PFC控制应用。内置最多3个12bit ADC（2MSPS，最多23通道），3个8bit DAC，1个编码器，3个比较器和3个PGA放大器。最多支持1个SPI，3个UART，1个I2C等通讯接口，2个看门狗模块，3个基本Timer，1个系统Timer，支持多达52个通用GPIO管脚。

表 1-1 芯片系列对照表

Part Number	3066M/3065P系列					3065H/3061H系列		
型号	3066 MNPI RH	3065P NPIM H	3065P NPIR H	3065P NPIRE	3065P NPIRA	3065 HRPIR Z	3065 HRPIC Z	3061 HRPIK Z
封装	LQFP6 4	LQFP8 0	LQFP64			LQFP6 4	LQFP4 8	LQFP3 2
eAI	Yes支持	No不支持						



Part Number		3066M/3065P系列				3065H/3061H系列		
Code Flash		512KB ( 含 eAI )	512KB	256KB	128KB	152KB		
Data Flash		Configurable				8KB		
SRAM		64KB				16KB		
CPU		RISC-V, 200MHz						
		FPU						
PWM	APT ( 每组 APT 含2 路 PWM )	9 ( APT0 ~ 8)				9 ( APT0 ~ 8 )	6 ( APT0/1/2/6/7/8 )	4 ( APT0/1/2/6 )
	GPT	2				2		
	CAP M	3				3		
	QD M	4				1	-	
通用32位 timer		4				3		
通讯	SPI	2				1		-
	UART	5				3		
	I2C	1				1		
	CAN	2				1		-
模拟	ADC	3				ADC0 ~ 2	ADC0 ~ 2	ADC1 ~ 2
	通道数	23ch				(4ch/10ch/9ch)	(3ch/10ch/9ch)	(6ch/4ch)
	DAC	3				3 ( DAC0 ~ 2 )		1 ( DAC1 )



Part Number		3066M/3065P系列			3065H/3061H系列		
	AC MP	3			3 ( ACMP0 ~ 2 )		1 ( AC MP1 )
	PGA	3			3 ( PGA0 ~ 2 )		2 ( PGA 1/2 )
	TSe nsor	1			1		
GPI O	GPI O总数	58	73	58	52	40	26
	5V 容忍 GPI O数	27	42	27	8	3	2
Watchdog		2, WDG+IWDG					
时钟		内部时钟：25MHz ±1%，32kHz ±3%					
		外部时钟：4MHz ~ 30MHz					
DMA通道数		6			4		
供电范围		2.4V ~ 3.63V					
工作温度		T <sub>A</sub> ：-40℃ ~ +105℃（环境温度）					
		T <sub>J</sub> ：-40℃ ~ +125℃（工作结温）					



# 2 模块移植

## 2.1 地址空间映射对比

表 2-1 地址空间映射对比

-	3066M/3065P		3065H/3061H		备注
模块	起始地址	大小	起始地址	大小	-
BOOTROM	0x0380_1000	12 KB	0x0100_0000	4 KB	-
SYSRAM_ITCM	-	-	0x0200_0000	16 KB	-
SYSRAM	0x0200_0000	64 KB	-	-	3066M/3065P系列将sram_itcm和sram_dtcmm合并。
FLASH	0x0300_0000	16 MB	0x0300_0000	16 MB	-
SYSRAM_DTCM	0x0400_0000	-	0x0400_0000	16 KB	-
CRG	0x1000_0000	4 KB	0x1000_0000	4 KB	-
CMM	0x1001_0000	4 KB	0x1001_0000	4 KB	-
CFD	0x1002_0000	4 KB	0x1002_0000	4 KB	-
SYSCTRL	0x1010_0000	64 KB	0x1010_0000	64 KB	-
UART0	0x1400_0000	4 KB	0x1400_0000	4 KB	-



-	3066M/3065P		3065H/3061H		备注
UART1	0x1400_1000	4 KB	0x1400_1000	4 KB	-
UART2	0x1400_2000	4 KB	0x1400_2000	4 KB	-
UART3	0x1400_3000	4 KB	-	-	-
UART4	0x1400_4000	4 KB	-	-	-
I2C0	0x1410_0000	4 KB	0x1410_0000	4 KB	3065H/3061H 模块名称为I2C
SPI0	0x1420_0000	4 KB	0x1420_0000	4 KB	3065H/3061H 模块名称为SPI
SPI1	0x1420_1000	4 KB	-	-	-
TIMER0	0x1430_0000	4 KB	0x1430_0000	4 KB	-
TIMER1	0x1430_1000	4 KB	0x1430_0020		-
TIMER2	0x1430_2000	4 KB	0x1430_1000	4 KB	-
TIMER3	0x1430_3000	4 KB	0x1430_1020		-
MTIMER	0x1438_0000	4 KB	-	-	-
WDG	-	-	0x1440_0000	4 KB	-
WWDG	0x1440_0000	4 KB	-	-	-
IWDG	0x1440_1000	4 KB	0x1440_1000	4 KB	-
GPIO0	0x1450_0000	4 KB	0x1450_0000	4 KB	-
GPIO1	0x1450_1000	4 KB	0x1450_1000	4 KB	-
GPIO2	0x1450_2000	4 KB	0x1450_2000	4 KB	-
GPIO3	0x1450_3000	4 KB	0x1450_3000	4 KB	-



-	3066M/3065P		3065H/3061H		备注
GPIO4	0x1450_4000	4 KB	0x1450_4000	4 KB	-
GPIO5	0x1450_5000	4 KB	0x1450_5000	4 KB	-
GPIO6	0x1450_6000	4 KB	0x1450_6000	4 KB	-
GPIO7	0x1450_7000	4 KB	0x1450_7000	4 KB	-
GPIO8	0x1450_8000	4 KB	-	-	-
GPIO9	0x1450_9000	4 KB	-	-	-
CAN0	0x1460_0000	4 KB	0x1460_0000	4 KB	3065H/3061H 模块名称为CAN
CAN1	0x1460_1000	4 KB	-	-	-
GPT0	0x1470_0000	4 KB	0x1470_0000	4 KB	-
GPT1	0x1470_1000	4 KB	0x1470_1000	4 KB	-
EFC	0x1471_0000	96 KB	0x1471_0000	96 KB	-
PMC	0x147E_0000	4 KB	0x147E_0200	3584 B	-
IOCMG	0x147F_0000	-	0x147E_0000	512 B	3065H/3061H 模块名称为IOCMG_AON
IOCMG_CORE	-	-	0x147F_0000	4 KB	-
CRC	0x1480_0000	4 KB	0x1480_0000	4 KB	-
IOCMG_ANA	-	-	0x149F_0000	4 KB	-
APT0	0x14A0_0000	4 KB	0x14A0_0000	4 KB	-
APT1	0x14A0_1000	4 KB	0x14A0_1000	4 KB	-



-	3066M/3065P		3065H/3061H		备注
APT2	0x14A0_2000	4 KB	0x14A0_2000	4 KB	-
APT3	0x14A0_3000	4 KB	0x14A0_3000	4 KB	-
APT4	0x14A0_4000	-	0x14A0_4000	4 KB	-
APT5	0x14A0_5000	-	0x14A0_5000	4 KB	-
APT6	0x14A0_6000	-	0x14A0_6000	4 KB	-
APT7	0x14A0_7000	-	0x14A0_7000	4 KB	-
APT8	0x14A0_8000	-	0x14A0_8000	4 KB	-
CAPM	0x14B0_0000	16 KB	0x14B0_0000	4 KB	-
QDM0	0x14C0_0000	4 KB	0x14C0_0000	4 KB	3065H/3061H 模块名称为QDM
QDM1	0x14C0_1000	4 KB	-	-	-
QDM2	0x14C0_2000	4 KB	-	-	-
QDM3	0x14C0_3000	4 KB	-	-	-
ADC0	0x1800_0000	4 KB	0x1800_0000	4 KB	3065H/3061H 模块名称为ADC
ADC1	0x1800_1000	4 KB	0x1800_1000	4 KB	-
ADC2	0x1800_2000	4 KB	0x1800_2000	4 KB	-
PGA0	0x1820_0000	4 KB	0x1820_0000	4 KB	-
PGA1	0x1820_1000	4 KB	0x1820_1000	4 KB	-
PGA2	0x1820_2000	4 KB	0x1820_2000	4 KB	-





-	3066M/3065P		3065H/3061H		备注
ACMP0	0x1830_0000	4 KB	0x1830_0000	4 KB	3065H/3061H系列ACMP和DAC基址相同，寄存器偏移地址不同。
ACMP1	0x1830_1000	4 KB	0x1830_1000	4 KB	
ACMP2	0x1830_2000	4 KB	0x1830_2000	4 KB	
DAC0	0x1840_0000	4 KB	0x1830_0000	4 KB	
DAC1	0x1840_1000	4 KB	0x1830_1000	4 KB	
DAC2	0x1840_2000	4 KB	0x1830_2000	4 KB	
TSENSOR	0x1850_0000	4 KB	-	-	3065H/3061HTSENSOR配置寄存器位于SYSCTRL，偏移0xA000。
ANA_CTRL_TOP	0x1860_0000	4 KB	-	-	-
DMA	0x1C00_0000	4 KB	0x1C00_0000	4 KB	-

## 2.2 中断特性对比

表 2-2 3066M/3065P 系列和 3065H/3061H 系列中断差异

特性	3066M/3065P	3065H/3061H
中断优先级	0 ~ 15	0 ~ 7
硬件断点数量	8	8
PMP数量	8	16



## 2.3 Flash 特性对比

表 2-3 3066M/3065P 系列和 3065H/3061H 系列 Flash 差异

特性	3066M/3065P	3065H/3061H
存储空间	512 KB。	160 KB。
存储空间管理	main_rgn0_0: 程序与用户参数存储区，大小用户可配。 main_rgn0_1: 程序与用户参数存储区，大小用户可配。 main_rgn0_2: 程序与用户参数存储区，大小用户可配。 main_rgn1: 用户参数存储区，大小用户可配。 info_rgn0: 固定参数存储区。 info_rgn1: 受保护参数存储区。 info_rgn2: 受保护参数存储区。 info_rgn3: 受保护参数存储区。 info_rgn4: 启动引导程序区。	main_rgn0: 程序存储区。 main_rgn1: 用户数据存储区。 info_rgn0: 固定参数存储区。 info_rgn1: 受保护参数存储区。
页大小	1 KB。	8 KB。
读数据宽度	64bit。	128bit。
编程数据宽度	64bit。	128bit。
主存储区JTAG读保护	main_rgn0_0; main_rgn0_1; main_rgn0_2。	main_rgn0。
信息存储区JTAG操作保护	info_rgn1; info_rgn2; info_rgn3; info_rgn4。	info_rgn1。
直接读取操作缓存	256 Cache Lines ( 2KB ) 。	64 Cache Lines ( 1KB ) 。
错误码纠错	64bit, 纠一检二。	128bit, 纠一检二。



## 2.4 ADC 特性对比

表 2-4 3066M/3065P 系列和 3065H/3061H 系列 ADC 差异

特性	3066M/3065P	3065H/3061H
采样精度	12bit。	12bit。
最高采样率	3 MSPS。	2 MSPS。
模拟输入	20模拟信号输入。	16个模拟输入。
转换模式	单次转换、连续转换。	单次转换、连续转换。
同步采样	-	支持。
过采样	支持。 选择单个SOC对同一个输入采样。 过采样结果直接从寄存器读出。	支持。 选择多个SOC对同一个输入采样。 过采样结果需要使用者自行累加计算。
转换SOC	16个SOC。 优先级：轮询优先级组和高优先级组。	16个SOC。 优先级：轮询优先级组和高优先级组。
触发源	29个触发源： 软件触发。 APT0_ADCA, APT0_ADCB; APT1_ADCA, APT1_ADCB; APT2_ADCA, APT2_ADCB; APT3_ADCA, APT3_ADCB; APT4_ADCA, APT4_ADCB; APT5_ADCA, APT5_ADCB; APT6_ADCA, APT6_ADCB; APT7_ADCA, APT7_ADCB; APT8_ADCA, APT8_ADCB; GPT0 ~ GPT3; TIMER0 ~ TIMER3; GPIO1_7、GPIO4_6、 GPIO_6_3、GPIO0_5。	23个触发源： 软件触发。 APT0_ADCA, APT0_ADCB; APT1_ADCA, APT1_ADCB; APT2_ADCA, APT2_ADCB; APT3_ADCA, APT3_ADCB; APT4_ADCA, APT4_ADCB; APT5_ADCA, APT5_ADCB; APT6_ADCA, APT6_ADCB; APT7_ADCA, APT7_ADCB; APT8_ADCA, APT8_ADCB; TIMER0 ~ TIMER3; GPIO(ADTRG)。
中断	4个数据中断、1个异常中断、1个事件中断。	4个数据中断、1个异常中断。
DMA模式	支持。	支持。



特性	3066M/3065P	3065H/3061H
后处理功能	数据偏移、误差计算、阈值检测、过零点检测、采样延迟记录。	采样延迟记录。
参考电压	内置VREF：1.1V。	内置VREF：2.0V/2.5V。

## 2.5 DMA 特性对比

表 2-5 3066M/3065P 系列和 3065H/3061H 系列 DMA 差异

特性	3066M/3065P	3065H/3061H
数据传输位宽	8bit、16bit、32bit数据位宽传输。	8bit、16bit、32bit数据位宽传输。
支持DMA（ Direct Memory Access ）通道数	6个DMA通道。	4个DMA通道。
Master总线接口	支持1个位宽32bit的Master总线接口。	支持2个位宽32bit的Master总线接口。
通道优先级	支持软件优先级可配，支持4级优先级。	不支持软件配置优先级，默认按通道序号优先级排序。
软件配置DMA请求	不支持。	支持。
Burst传输	支持。	支持。
流控方式	DMA流控和外设流控。	DMA流控和外设流控。
链表模式	支持。	支持。
源和目的地址自增模式	支持自动递增或不递增。	支持自动递增或不递增。
Master字节序配置	不支持字节序配置。	支持字节序配置为大端或小端格式。



## 2.6 CRC 特性对比

表 2-6 3066M/3065P 系列和 3065H/3061H 系列 CRC 差异

特性	3066M/3065P	3065H/3061H
CRC (Cyclic Redundancy Check) 算法多项式	支持4种多项式可配 (CRC8-07 / CRC16-1021 / CRC16-8005 / CRC32-04C11DB7)。	支持4种多项式可配 (CRC8-07 / CRC16-1021 / CRC16-8005 / CRC32-04C11DB7)。
输入数据有效位宽	支持8/16/32bit有效位宽配置。	支持8/16/32bit有效位宽配置。
CRC初值配置	支持。	支持。
CRC码值计算	支持。	支持。
CRC输入数据反序	支持按字节反序。	不支持。
CRC输出数据反序	支持按有效位宽反序。	不支持。
CRC输出数据与指定值异或	支持。	不支持。
CRC输入数据大小端配置	支持。	不支持。
总线反压超时中断	不支持。	支持。

## 2.7 WDG 特性对比

3065H/3061H的WDG中包含独立看门狗和看门狗，没有进行模块区分。为了方便用户理解，3066M/3065P把WDG拆分成IWDG和WWDG两个模块。

表 2-7 3066M/3065P 的 IWDG 和 3065H/3061H 的 WDG 差异

特性	3066M/3065P ( IWDG )	3065H/3061H ( WDG )
计数器	支持12bit减法计数器。	支持32bit减法计数器。
计数初值配置	支持。	支持。
寄存器锁定	支持。	支持。
超时中断产生	不支持。	支持。
复位信号产生	支持。	支持。
调试模式	支持。	支持。



特性	3066M/3065P ( IWDG )	3065H/3061H ( WDG )
窗口模式	支持，窗口外喂狗会触发复位。	不支持。
计数器当前值回读	支持。	支持。
硬件看门狗	支持。	不支持。
安全看门狗	支持。	不支持。
低功耗下工作和唤醒	支持。	不支持。

表 2-8 3066M/3065P 的 WWDG 和 3065H/3061H 的 WDG 差异

特性	3066M/3065P ( WWDG )	3065H/3061H ( WDG )
计数器	支持16bit减法计数器。	支持32bit减法计数器。
计数初值配置	支持。	支持。
寄存器锁定	支持。	支持。
超时中断产生	支持。	支持。
复位信号产生	支持。	支持。
调试模式	支持。	支持。
窗口模式	支持，窗口外喂狗会触发复位。	不支持。
计数器当前值回读	支持。	支持。

## 2.8 SPI 特性对比

表 2-9 3066M/3065P 系列和 3065H/3061H 系列 SPI 差异

特性	3066M/3065P	3065H/3061H
时钟频率可配置	支持。	支持。
主从模式配置	支持。	支持。
片选配置	双片选，片选支持软件控制。	双片选，片选不支持软件控制。
发送和接收 FIFO深度	16*16bit。	256*16bit。



特性	3066M/3065P	3065H/3061H
数据帧长度	支持4bit ~ 16bit可编程。	支持4bit ~ 16bit可编程。
单帧或连续帧	支持。	支持。
Motorola SPI 全双工	支持。 支持三线通信。	支持。
MicroWire半双工	支持。	支持。
TI格式全双工	支持。	支持。
DMA操作	支持。	支持。

## 2.9 GPT 特性对比

表 2-10 3066M/3065P 系列和 3065H/3061H 系列 GPT 差异

特性	3066M/3065P	3065H/3061H
时钟分频	支持12位分频（1 ~ 4096分频）。	支持1/2/4/8分频。
输出PWM波模式	支持无限或有限个PWM波。	支持无限或有限个PWM波。
占空比配置	支持。	支持。
设置周期结束触发周期中断、DMA请求、ADC采样	支持。	不支持。
设置输出完成触发完成中断、DMA请求、ADC采样	支持。	不支持。
支持输出脉冲个数配置	支持0 ~ 1023个脉冲数配置。	支持0 ~ 1023个脉冲数配置。
计数器	16bit计数。	26bit计数。
寄存器缓存加载模式	支持。	不支持。

## 2.10 UART 特性对比

表 2-11 3066M/3065P 系列和 3065H/3061H 系列 UART 差异

特性	3066M/3065P	3065H/3061H
发送FIFO（First In First Out）	支持8x8bit。	支持256x8bit。



特性	3066M/3065P	3065H/3061H
接收FIFO	支持8x12bit。	支持256x12bit。
数据位编程	支持5bit/6bit/7bit/8bit。	支持5bit/6bit/7bit/8bit。
停止位编程	支持1bit/2bit。	支持1bit/2bit。
校验方式	支持奇/偶校验，0/1校验或无校验。	支持奇/偶校验，0/1校验或无校验。
支持传输速率（波特率）可编程	支持。	支持。
中断支持	支持字符匹配中断、自动波特率检测错误中断、自动波特率检测完成中断、接收FIFO满中断、接收FIFO非空中断、接收FIFO空中断、发送FIFO非空中断、发送FIFO空中断、溢出错误中断、break错误中断、校验中断、帧错误中断、接收FIFO水线中断、发送FIFO水线中断、接收超时中断、CTS（Clear To Send）调制状态中断。	支持接收 FIFO 中断、发送 FIFO 中断、接收超时中断、错误中断、调制状态中断。
禁止 UART 模块或者 UART 发送/接收功能以降低功耗	支持。	支持。
关断 UART 时钟以节省功耗	支持。	支持。
DMA 操作（与 DMA 模块配合使用）	支持。	支持。
波特率自动检测	支持。	不支持。
字符检测，检测字符可配	支持。	不支持。
12 ~ 16倍过采样可配置	支持（8 ~ 16倍过采样率）。	不支持。
发送数据和接收数据位序可配置	支持。	不支持。





## 2.11 CMM 特性对比

表 2-12 3066M/3065P 系列和 3065H/3061H 系列 CMM 差异

特性	3066M/3065P	3065H/3061H
参考时钟类型	支持4选1。	支持4选1。
参考时钟分频	最大32分频。	最大32分频。
目标时钟类型	支持4选1。	支持5选1。
目标时钟分频	最大8192分频。	最大8192分频。
检测目标时钟的频率	支持。	支持。

## 2.12 CFD 特性对比

表 2-13 3066M/3065P 系列和 3065H/3061H 系列 CFD 差异

特性	3066M/3065P	3065H/3061H
检测目标时钟是否失效	支持	支持
失效后自动保护，产生系统事件输出到APT（Advanced PWM Timer）模块	支持	支持

## 2.13 TIMER 特性对比

表 2-14 3066M/3065P 系列和 3065H/3061H 系列 TIMER 差异

特性	3066M/3065P	3065H/3061H
可编程3档预分频：1、16、256倍分频	支持	支持
32bit/16bit减法定时/计数	支持	支持
计数时钟可配置：总线时钟的1、2、4、8分频	不支持	支持
3种计数模式：自由运行模式、周期模式、单次计数模式	支持	支持



特性	3066M/3065P	3065H/3061H
2种载入计数初值方法	支持	支持
随时读取当前计数值	支持	支持
当计数值减到0时会 产生timer定时中断	支持	支持
DMA请求溢出中断	支持	不支持
产生DMA请求、产 生触发ADC采样信 号	支持	支持

## 2.14 CAPM 特性对比

两个系列芯片，CAPM支持的特性一致。

表 2-15 3066M/3065P 系列和 3065H/3061H 系列 CAPM 差异

特性	3066M/ 3065P	3065H/ 3061H
3通道捕获：CAPM0、CAPM1、CAPM2	支持	支持
输入滤波（过滤毛刺宽度 1 ~ 8192 个时钟周期）、输入 预分频（2 ~ 510偶数分频）	支持	支持
输入电平实时检测	支持	支持
边沿计数	支持	支持
32bit时间计数、计数分频	支持	支持
单个CAPM（Capture Module）最多捕获4个事件，保存 最多4个时间戳，支持捕获溢出检测	支持	支持
循环捕获、单轮次捕获	支持	支持
计数相位同步、APT硬件同步、软件同步	支持	支持
工作状态复位，配置寄存器不复位	支持	支持
DMA触发、DMA传输、DMA溢出检测	支持	支持
仿真器接入	支持	支持



## 2.15 QDM 特性对比

表 2-16 3066M/3065P 系列和 3065H/3061H 系列 QDM 差异

特性	3066M/3065P	3065H/3061H
4种类型的增量编码器：正交型、脉冲方向型、非标准TYPE1型、非标准TYPE2型	支持。	支持。
A/B/Z相信号滤波和极性选择，A/B相信号互换	支持。	支持。
1X, 2X, 4X倍频解码	支持。	支持。
PPU ( Position Process Unit ) 位置计数	支持独立使能，支持3种计数模式。	支持独立使能，支持3种计数模式。
PPU位置计数复位	4种复位模式。	4种复位模式。
PPU位置计数初始化	3种初始化模式。	3种初始化模式。
PPU位置计数锁存	3种Z相锁存模式，2种锁存模式。	3种Z相锁存模式，2种锁存模式。
PPU位置计数比较	支持位置比较缓存模式，支持比较输出同步信号。	支持位置比较缓存模式，支持比较输出同步信号。
TSU ( Time Stamp Unit ) 时间戳记录，可用于速度计算	支持。	支持。
PTU ( Period Trigger Unit ) 周期触发，可用于位置记录	支持。	支持。
PTU看门狗模式，可用于检测A/B信号输入	支持。	支持。
DMA访问	支持。	支持。
仿真器接入	支持3种接入模式。	支持3种接入模式。

## 2.16 CRG 特性对比

表 2-17 3066M/3065P 系列和 3065H/3061H 系列 CRG 差异

特性	3066M/3065P	3065H/3061H
高速外部晶振或外部时钟源	支持4MHz ~ 30MHz。	支持2MHz ~ 30MHz。



特性	3066M/3065P	3065H/3061H
PLL ( Phase Locked Loop ) 前置分频系数	支持1/2/3/4/5/6/7/8分频。	支持1/2/4分频。
PLL反馈分频系数	支持6 ~ 127范围可配。	支持6 ~ 63范围可配。
PLL后置分频系数	支持两个后置分频器，postdiv1和postdiv2。 postdiv1支持1/2/3/4/5/6/7/8分频。 postdiv2支持1/2/3/4/5/6/7/8分频。	支持1个后置分频器。 支持1/2/4/8/16/32/2/4/8/16/32分频。
PLL VCO输出时钟频率	允许频率范围200MHz ~ 400MHz。	允许频率范围100MHz ~ 200MHz。
PLL输出时钟频率	支持两路时钟输出，clk_pll_pst1和clk_pll_pst2。 clk_pll_pst1允许频率最大200MHz。 clk_pll_pst2允许频率最大100MHz。	支持一路时钟输出，clk_pll。 clk_pll允许频率最大200MHz。
系统主工作时钟	为高速外设提供时钟，包括：DMA、CMM ( Clock Monitor Module )、GPIO0 ~ 9、APT0 ~ 8、GPT0 ~ 1、QDM0 ~ 3、CRC、CAPM0 ~ 2、SPI0 ~ 1、I2C、UART0 ~ 4、TIMER0 ~ 3、EFC、ADC控制器、PGA控制器、ACMP控制器、DAC控制器、ADCVREF控制器。	为高速外设提供时钟，包括：DMA、EFC、CORESIGHT、APT0 ~ 8、CAPM0 ~ 2、QDM、CRC、ADC控制器0 ~ 2、PGA控制器0 ~ 2、ACMP ( Analog Comparator ) 控制器0 ~ 2、DAC控制器0 ~ 2、DAC控制器0 ~ 2。 为部分低速外设提供2分频作为工作时钟，部分低速外设包括：I2C、GPIO0 ~ 7。 为部分低速外设提供4/8/16分频作为工作时钟，部分低速外设包括：SPI、UART0 ~ 2、GPT0 ~ 1、Timer0 ~ 3、WDOG。
ADC工作时钟	最大支持100 MHz。 支持时钟源有：clk_pst2_100m ( 默认值)、hclk。	最大支持40MHz。 支持时钟源有：clk_hosc ( 默认值 )、clk_xtal、clk_vco 的分频时钟。
输出时钟	支持的时钟有：clk_hosc、clk_losc、clk_xtal、clk_bg_chopper、clk_adc0/4、hclk/6、clk_hosc_div。	支持的时钟有：clk_pll/128、clk_hosc、clk_losc、clk_xtal、clk_dac0、clk_dac1、clk_dac2、clk_adc0/2、clk_adc1/2、clk_adc2/2。



## 2.17 I2C 特性对比

表 2-18 3066M/3065P 系列和 3065H/3061H 系列 I2C 差异

特性	3066M/3065P	3065H/3061H
模式	支持主机、从机、主从机模式。	支持主机模式。
FIFO	支持16 x 12bit的TX FIFO； 支持16 x 8bit的RX FIFO。	支持 64 x 8bit的TX FIFO； 支持 64 x 8bit的RX FIFO。
寻址模式	支持标准（7bit）和扩展地址（10bit）。	支持标准（7bit）和扩展地址（10bit）。
DMA操作	支持。	支持。
通信速率	支持标准模式（100k bit/s）和快速模式（400k bit/s）。	支持标准模式（100k bit/s）和快速模式（400k bit/s）。
支持灵活配置ACK/NACK	支持。	不支持。
Slave地址	支持2个。	不支持。
数字滤波	支持SCL时钟线和SDA数据线Spike Suppression（数字滤波）功能。	不支持。
SCL（Serial Clock Line）延长	支持SCL延长。	不支持。
从机地址与掩码	支持2个从机地址与地址掩码。	不支持。
SCL低电平超时检测	支持总线SCL时钟低电平超时检测。	不支持。

## 2.18 ACMP 特性对比

表 2-19 3066M/3065P 系列和 3065H/3061H 系列 ACMP 差异

特性	3066M/3065P	3065H/3061H
比较器	支持3个2输入比较器。	支持1个2输入比较器。
比较源	支持4组比较源可配置选择。	支持6组比较源可配置选择。
最小有效差分输入电压	支持最小有效差分输入电压为20mV。	支持最小有效差分输入电压为20mV。



特性	3066M/3065P	3065H/3061H
共模输入电压范围	支持0V ~ 输入电源大小。	支持0V ~ 输入电源大小。
比较结果屏蔽	支持。	支持。

## 2.19 DAC 特性对比

表 2-20 3066M/3065P 系列和 3065H/3061H 系列 DAC 差异

特性	3066M/3065P	3065H/3061H
分辨率	10bit分辨率	8bit分辨率
并行数据输入	支持	支持
转换速率	支持300kHz转换速率	支持300kHz转换速率

## 2.20 PGA 特性对比

3066M/3065P系列和3065H/3061H系列的PAG模块无差异。

表 2-21 3066M/3065P 系列和 3065H/3061H 系列 PGA 差异

特性	3066M/3065P	3065H/3061H
全差分输入	支持。	支持。
内部电阻模式，增益配置	支持增益可编程为2、4、8或16。	支持增益可编程为2、4、8或16。
外部电阻模式，灵活可调增益	支持。	支持。
外部电阻模式，前馈电容补偿	支持。	不支持。
压摆率	最小值：8V/us。	最小值：15V/us。 典型值：20V/us。



## 2.21 APT 特性对比

表 2-22 3066M/3065P 系列和 3065H/3061H 系列 APT 差异

特性	3066M/3065P	3065H/3061H
分频器	支持12位分频器。	支持12位分频器。
分频计数器的参考值配置	不支持。	支持。
计数器	支持16位计数器。	支持16位计数器。
两路PWM波特征	支持两路独立PWM输出。	支持两路独立PWM输出。
死区插入	支持在任一路PWM波的上升沿和/或下降沿插入死区。	支持在任一路PWM波的上升沿和/或下降沿插入死区。
保护	支持在发生故障时控制输出波形，以进行保护。	支持在发生故障时控制输出波形，以进行保护。
定时中断	支持在设定的时间点，周期性地上报定时中断。	支持在设定的时间点，周期性地地上报定时中断。
事件中断	支持在检测到异常事件时，上报异常事件中断。	支持在检测到异常事件时，上报异常事件中断。
ADC 启动控制信号	支持在设定的时间点，周期性地产生 2 路 ADC 启动控制信号（SOC 信号）。	支持在设定的时间点，周期性地产生2路ADC启动控制信号（SOC 信号）。
DMA 请求	支持在设定的时间点，周期性地产生DMA burst 或 DMA single请求。	支持在设定的时间点，周期性地产生DMA burst或DMA single 请求。
同步信号	支持发送同步信号，作为其它APT或CAMP的同步源。	支持发送同步信号，作为其他APT或CAMP的同步源。
计数器同步	支持接收其他APT或CAMP的同步信号，对内置计时器进行同步。	支持接收其他APT或CAMP的同步信号，对内置计时器进行同步。
同时启动	支持与任意APT同时启动。	支持与任意APT同时启动。
缓存加载	支持缓存关键配置参数，在设定的时间或事件到来时，加载参数。	支持缓存关键配置参数，在设定的时间或事件到来时，加载参数。



特性	3066M/3065P	3065H/3061H
事件管理	支持接收 IO 管脚、ACMP的信号，以及3个系统异常信号（core_debug_mode、clk_fail、sys_mem_fail），用于产生合并事件、同步控制事件、计数器同步事件和输出保护事件。	支持接收IO管脚、ACMP的信号，以及3个系统异常信号（core_debug_mode、clk_fail、sys_mem_fail），用于产生合并事件、同步控制事件、计数器同步事件和输出保护事件。
谷底开关配置	不支持。	支持。
APT子模块	支持计数控制子模块、波形生成子模块、死区生成子模块、输出控制子模块、中断生成子模块、ADC启动子模块、事件管理子模块。	支持计数控制子模块、波形生成子模块、死区生成子模块、输出控制子模块、中断生成子模块、ADC启动子模块、事件管理子模块。
状态查询	支持POE电平状态，PWM输出状态可查。	不支持查询POE电平状态，PWM输出状态。

## 2.22 CAN 特性对比

3066M/3065P系列和3065H/3061H系列的CAN模块无差异。

表 2-23 3066M/3065P 系列和 3065H/3061H 系列 CAN 差异

特性	3066M/3065P	3065H/3061H
标准技术规范 CAN 2.0A 和 CAN 2.0B	支持。	支持。
多设备时的总线仲裁	支持。	支持。
传输速率可编程	支持。	支持。
报文对象	最大支持32个报文对象。	最大支持32个报文对象。
故障节点的隔离	支持主动错误和被动错误的自我判定以及故障节点的隔离。	支持主动错误和被动错误的自我判定以及故障节点的隔离。
错误的自我修复	支持。	支持。
自动重传模式	支持。	支持。
报文接收过滤功能	支持。	支持。





特性	3066M/3065P	3065H/3061H
中断屏蔽	支持。	支持。
连续报文接收	支持。	支持。
测试模式	支持测试模式，支持Loop-back、Silent模式以及两者同时开启。	支持测试模式，支持Loop-back、Silent模式以及两者同时开启。

## 2.23 GPIO 特性对比

3066M/3065P与3065H/3061H系列芯片，GPIO支持特性一致。

## 2.24 PMC 特性对比

表 2-24 3066M/3065P 系列和 306x 系列 PMC 差异

特性	3066M/3065P	3065H/3061H
深睡眠模式	支持	支持
关机模式	不支持	支持
PVD功能	支持	支持
WAKEUP管脚唤醒源选择	支持	支持
WAKEUP管脚唤醒源极性可配	支持	支持

## 2.25 IOCMG 特性对比

3066M/3065P与3065H/3061H系列芯片开漏输出特性不一致。

表 2-25 3066M/3065P 系列和 3065H/3061H 系列 IOCMG 差异

特性	3066M/3065P	3065H/3061H
上/下拉	支持上拉、下拉、不进行上下拉。	支持上拉、下拉、不进行上下拉。
施密特输入开/关	支持打开/关闭。	支持打开/关闭。
输出驱动能力	支持4个档位水平。	支持4个档位水平。
输出信号边沿快/慢	支持快沿输出和慢沿输出。	支持快沿输出和慢沿输出。



特性	3066M/3065P	3065H/3061H
数字/模拟模式切换	支持数字模式和模拟模式。	支持数字模式和模拟模式。
复用功能选择	支持。	支持。
开漏输出	支持。	不支持。



# 3 软件包移植

## 3.1 ADC 驱动移植

### 3.1.1 接口移植

表 3-1 接口移植对照表

功能	3066M/3065P ADC HAL接口	3065H/3061H ADC HAL接口	是否差异
初始化和配置	BASE_StatusType HAL_ADC_Init(ADC_Handle *adcHandle)	BASE_StatusType HAL_ADC_Init(ADC_Handle *adcHandle)	相同
	BASE_StatusType HAL_ADC_Deinit(ADC_Handle *adcHandle)	BASE_StatusType HAL_ADC_Deinit(ADC_Handle *adcHandle)	相同
	BASE_StatusType HAL_ADC_ConfigureSoc(ADC_Handle *adcHandle, ADC_SOCNumber soc, SOC_Param *socParam)	BASE_StatusType HAL_ADC_ConfigureSoc(ADC_Handle *adcHandle, ADC_SOCNumber soc, SOC_Param *socParam)	相同
采样使能和结果读取	BASE_StatusType HAL_ADC_StartDma(ADC_Handle *adcHandle, unsigned int startSoc, unsigned int endSoc, unsigned int *saveData)	BASE_StatusType HAL_ADC_StartDma(ADC_Handle *adcHandle, unsigned int startSoc, unsigned int endSoc, unsigned int *saveData)	相同
	BASE_StatusType HAL_ADC_StartIt(ADC_Handle *adcHandle)	BASE_StatusType HAL_ADC_StartIt(ADC_Handle *adcHandle)	相同



功能	3066M/3065P ADC HAL接口	3065H/3061H ADC HAL接口	是否差异
	unsigned int HAL_ADC_GetConvResult(ADC_Handle *adcHandle, unsigned int soc)	unsigned int HAL_ADC_GetConvResult(ADC_Handle *adcHandle, unsigned int soc)	相同
	BASE_StatusType HAL_ADC_CheckSocFinish(ADC_Handle *adcHandle, unsigned int soc)	BASE_StatusType HAL_ADC_CheckSocFinish(ADC_Handle *adcHandle, unsigned int soc)	相同
	BASE_StatusType HAL_ADC_SoftTrigSample(ADC_Handle *adcHandle, unsigned int soc)	BASE_StatusType HAL_ADC_SoftTrigSample(ADC_Handle *adcHandle, unsigned int soc)	相同
	BASE_StatusType HAL_ADC_SoftTrigMultiSample(ADC_Handle *adcHandle, ADC_SoftMultiTrig syncTrig)	BASE_StatusType HAL_ADC_SoftTrigMultiSample(ADC_Handle *adcHandle, ADC_SoftMultiTrig syncTrig)	相同
同步采样	-	BASE_StatusType HAL_ADC_StartSyncSampleEx(ADC_Handle *adcHandle, SOC_SyncParam *syncParam)	差异
软件校准采样结果	-	unsigned int HAL_ADC_ActiveCalibrateRetEx(ADC_RegStruct * const adcx, unsigned int soc, unsigned int originalRet)	差异
中断与回调	void HAL_ADC_IrqHandlerInt0(void *handle)	void HAL_ADC_IrqHandlerInt1(void *handle)	差异
	void HAL_ADC_IrqHandlerInt1(void *handle)	void HAL_ADC_IrqHandlerInt2(void *handle)	差异
	void HAL_ADC_IrqHandlerInt2(void *handle)	void HAL_ADC_IrqHandlerInt3(void *handle)	差异
	void HAL_ADC_IrqHandlerInt3(void *handle)	void HAL_ADC_IrqHandlerInt4(void *handle)	差异
	void HAL_ADC_IrqHandlerOver(void *handle)	void HAL_ADC_IrqHandlerOver(void *handle)	相同



功能	3066M/3065P ADC HAL接口	3065H/3061H ADC HAL接口	是否差异
事件中断	void HAL_ADC_IrqHandlerAllEvent(void *handle)	-	差异
注册回调函数	void HAL_ADC_RegisterCallBack(ADC_Handle *adcHandle, ADC_CallbackFunType typeId, ADC_CallbackType pCallback)	void HAL_ADC_RegisterCallBack(ADC_Handle *adcHandle, ADC_CallbackFunType typeId, ADC_CallbackType pCallback)	相同
后处理功能	BASE_StatusType HAL_ADC_EnablePPBxEventIntEx(ADC_Handle *adcHandle)	-	差异
	BASE_StatusType HAL_ADC_DisablePPBxEventIntEx(ADC_Handle *adcHandle)	-	差异
	BASE_StatusType HAL_ADC_ConfigurePPBxEx(ADC_Handle *adcHandle, ADC_SOCNumber soc, ADC_PPBNumber ppb, PPB_Function *fun)	-	差异
	BASE_StatusType HAL_ADC_SetPPBxOffsetEx(ADC_Handle *adcHandle, ADC_PPBNumber ppb, int offset)	-	差异
	BASE_StatusType HAL_ADC_SetPPBxThresholdEx(ADC_Handle *adcHandle, ADC_PPBNumber ppb, int up, int dn)	-	差异
	BASE_StatusType HAL_ADC_SetPPBxErrorRefEx(ADC_Handle *adcHandle, ADC_PPBNumber ppb, unsigned int ref)	-	差异
	int HAL_ADC_GetPPBxErrorResultEx(ADC_Handle *adcHandle, ADC_PPBNumber ppb)	-	差异



功能	3066M/3065P ADC HAL接口	3065H/3061H ADC HAL接口	是否差异
	unsigned int HAL_ADC_GetPPBxDelayCntEx( ADC_Handle *adcHandle, ADC_PPBNumber ppb)	-	差异
过采样功能	BASE_StatusType HAL_ADC_CheckOversamplingFinishEx(ADC_Handle *adcHandle)	-	差异
	unsigned int HAL_ADC_GetOversamplingResultEx(ADC_Handle *adcHandle)	-	差异
连续采样功能	BASE_StatusType HAL_ADC_EnableSocContinueModeEx(ADC_Handle *adcHandle, ADC_SOCNumber soc)	-	差异
	BASE_StatusType HAL_ADC_DisableSocContinueModeEx(ADC_Handle *adcHandle, ADC_SOCNumber soc)	-	差异

说明

- 3065H/3061H系列过采样功能，利用多条通道单次采样后，自行累加结果实现过采样功能。3066M/3065P系列过采样功能，利用一条通道多次采样后，自动累加结果至过采样寄存器，对比3065H/3061H需要额外提供读取过采样结果的接口。
- 3065H/3061H系列连续采样功能，使用数据中断进行触发，不需要单独配置使能。3066M/3065P系列连续采样，不需要数据中断进行触发，有单独控制连续采样的寄存器，对比3065H/3061H需要提供额外的使能/去使能接口。

3.1.2 配置参数变更

3.1.2.1 SOC\_Param

3066M/3065P系列SOC\_Param结构体定义。

```
typedef struct {  
    ADC_Input      adcInput;  
    ADC_SOCSampleCycle sampleTotalTime;  
    ADC_TrigSource  trigSource;  
    bool            continueMode;  
    ADC_SOCFinishMode finishMode;  
} SOC_Param;
```

3065H/3061H系列SOC\_Param结构体定义。



```
typedef struct {  
    ADC_Input      adcInput;  
    unsigned int    sampleTotalTime;  
    unsigned int    sampleHoldTime;  
    ADC_SoftTrigSoc softTrigSource;  
    ADC_PeriphTrigSoc periphTrigSource;  
    ADC_IntTrigSoc  intTrigSource;  
    ADC_SOCFinishMode finishMode;  
} SOC_Param;
```

表 3-2 SOC\_Param 参数差异点对比

参数功能	3066M/ 3065P	3065H/ 3061H	差异点
配置采样输入通道	adcInput	adcInput	-
配置采样时间	sampleTotalTime	sampleTotalTime	3066M/3065P：只支持配置采样总时间。
		sampleHoldTime	3065H/3061H：支持配置采样总时间和采样阶段保持器占用的时间。
配置触发源	trigSource	softTrigSource	3066M/3065P：软件触发源和硬件触发源使用一个参数配置。
		periphTrigSource	3065H/3061H：软件触发源配置用softTrigSource，硬件触发源配置用periphTrigSource。
配置连续转换模式	continueMode	intTrigSource	3066M/3065P：参数类型bool，只配置是否使能连续转换功能。 3065H/3061H：类型ADC_IntTrigSoc，配置使用哪个数据中断来触发连续转换。
配置采样转换结束模式	finishMode	finishMode	-

### 3.2 DMA 驱动移植



### 3.2.1 接口移植

表 3-3 3066M/3065P 系列和 3065H/3061H 系列 DMA 接口对比

功能	3066M/3065P DMA HAL接口	3065H/3061H DMA HAL接口	是否差异
初始化和配置	BASE_StatusType HAL_DMA_Init(DMA_Handle *dmaHandle)	BASE_StatusType HAL_DMA_Init(DMA_Handle *dmaHandle)	相同
	BASE_StatusType HAL_DMA_DeInit(DMA_Handle *DmaHandle)	BASE_StatusType HAL_DMA_DeInit(DMA_Handle *DmaHandle)	相同
	BASE_StatusType HAL_DMA_InitChannel(DMA_Handle *dmaHandle, DMA_ChannelParam *channelParam, unsigned int channel)	BASE_StatusType HAL_DMA_InitChannel(DMA_Handle *dmaHandle, DMA_ChannelParam *channelParam, unsigned int channel)	相同
	BASE_StatusType HAL_DMA_InitNewNode(DMA_LinkList *node, const DMA_ChannelParam *param, unsigned int srcAddr, unsigned int destAddr, unsigned int tranSize)	BASE_StatusType HAL_DMA_InitNewNode(DMA_LinkList *node, const DMA_ChannelParam *param, unsigned int srcAddr, unsigned int destAddr, unsigned int tranSize)	相同
	BASE_StatusType HAL_DMA_ListAddNode(DMA_LinkList *head, DMA_LinkList *newNode)	BASE_StatusType HAL_DMA_ListAddNode(DMA_LinkList *head, DMA_LinkList *newNode)	相同
使能搬运与停止	BASE_StatusType HAL_DMA_Start(DMA_Handle *dmaHandle, unsigned int srcAddr, unsigned int destAddr, unsigned int dataLength, unsigned int channel)	BASE_StatusType HAL_DMA_Start(DMA_Handle *dmaHandle, unsigned int srcAddr, unsigned int destAddr, unsigned int dataLength, unsigned int channel)	相同
	BASE_StatusType HAL_DMA_StartIT(DMA_Handle *dmaHandle, unsigned int srcAddr, unsigned int destAddr, unsigned int dataLength, unsigned int channel)	BASE_StatusType HAL_DMA_StartIT(DMA_Handle *dmaHandle, unsigned int srcAddr, unsigned int destAddr, unsigned int dataLength, unsigned int channel)	相同





功能	3066M/3065P DMA HAL接口	3065H/3061H DMA HAL接口	是否差异
	BASE_StatusType HAL_DMA_StopChannel(DMA_Handle *dmaHandle, unsigned int channel)	BASE_StatusType HAL_DMA_StopChannel(DMA_Handle *dmaHandle, unsigned int channel)	相同
	BASE_StatusType HAL_DMA_GetChannelState(DMA_Handle *dmaTHandle, unsigned int channel)	BASE_StatusType HAL_DMA_GetChannelState(DMA_Handle *dmaTHandle, unsigned int channel)	相同
	BASE_StatusType HAL_DMA_StartListTransfer(DMA_Handle *dmaHandle, DMA_LinkList *head, unsigned int channel)	BASE_StatusType HAL_DMA_StartListTransfer(DMA_Handle *dmaHandle, DMA_LinkList *head, unsigned int channel)	相同
	void HAL_DMA_QuickStart(DMA_Handle *dmaHandle, unsigned int channel)	void HAL_DMA_QuickStart(DMA_Handle *dmaHandle, unsigned int channel)	相同
中断处理	void HAL_DMA_IrqHandlerTc(void *handle)	void HAL_DMA_IrqHandlerTc(void *handle)	相同
	void HAL_DMA_IrqHandlerError(void *handle)	void HAL_DMA_IrqHandlerError(void *handle)	相同
注册回调函数	void HAL_DMA_RegisterCallBack(DMA_Handle * dmaHandle, DMA_CallbackFun_Type typeID, Unsigned int channel, DMA_CallbackType pCallback)	void HAL_DMA_RegisterCallBack(DMA_Handle * dmaHandle, DMA_CallbackFun_Type typeID, Unsigned int channel, DMA_CallbackType pCallback)	相同
通道优先级	void HAL_DMA_SetChannelPriorityEx(DMA_Handle *dmaHandle, unsigned int channel, DMA_ChannelPriority priority)	-	差异

### 3.2.2 配置参数变更

DMA模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.3 CRC 驱动移植



### 3.3.1 接口移植

表 3-4 3066M/3065P 系列和 3065H/3061H 系列 CRC 接口对比

功能	3066M/3065P CRC HAL接口	3065H/3061H CRC HAL接口	是否差异
初始化	BASE_StatusType HAL_CRC_Init(CRC_Handle *handle)	BASE_StatusType HAL_CRC_Init(CRC_Handle *handle)	相同
	void HAL_CRC_DeInit(CRC_Handle *handle)	void HAL_CRC_DeInit(CRC_Handle *handle)	相同
	-	void CRC_RspltInit(CRC_Handle *handle)	差异
读写配置	void HAL_CRC_SetCheckInData(CRC_Handle *handle, unsigned int data)	void HAL_CRC_SetCheckInData(CRC_Handle *handle, unsigned int data)	相同
	unsigned int HAL_CRC_LoadCheckInData(CRC_Handle *handle)	unsigned int HAL_CRC_LoadCheckInData(CRC_Handle *handle)	相同
计算校验	unsigned int HAL_CRC_SetInputDataGetCheck(CRC_Handle *handle, unsigned int data)	unsigned int HAL_CRC_SetInputDataGetCheck(CRC_Handle *handle, unsigned int data)	相同
	bool HAL_CRC_CheckInputData(CRC_Handle *handle, const void *pData, unsigned int length, unsigned int crcValue)	bool HAL_CRC_CheckInputData(CRC_Handle *handle, const void *pData, unsigned int length, unsigned int crcValue)	相同
	unsigned int HAL_CRC_Accumulate(CRC_Handle *handle, void *pData, unsigned int length)	unsigned int HAL_CRC_Accumulate(CRC_Handle *handle, void *pData, unsigned int length)	相同
	unsigned int HAL_CRC_Calculate(CRC_Handle *handle, void *pData, unsigned int length)	unsigned int HAL_CRC_Calculate(CRC_Handle *handle, void *pData, unsigned int length)	相同
中断处理	void HAL_CRC_IrqHandler(void *handle)	void HAL_CRC_IRQHandler(void *param)	差异



功能	3066M/3065P CRC HAL接口	3065H/3061H CRC HAL接口	是否差异
注册回调函数	void HAL_CRC_RegisterCallback(CRC_Handle *handle, CRC_CallbackType callBackFunc)	void HAL_CRC_RegisterCallback(CRC_Handle *handle, CRC_CallbackType callBackFunc)	相同
注册服务函数	-	void HAL_CRC_IRQService(CRC_Handle *handle)	差异

### 3.3.2 配置参数变更

CRC模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.4 WDG 驱动移植

### 3.4.1 接口移植

表 3-5 3066M/3065P 系列 IWDG 和 3065H/3061H 系列 IWDG 接口对比

功能	3066M/3065P IWDG HAL接口	3065H/3061H WDG HAL接口	是否差异
初始化	BASE_StatusType HAL_IWDG_Init(IWDG_Handle *handle)	BASE_StatusType HAL_IWDG_Init(IWDG_Handle *handle)	相同
读写配置	void HAL_IWDG_SetTimeValue(IWDG_Handle *handle, unsigned int timeValue, IWDG_TimeType timeType)	void HAL_IWDG_SetTimeValue(IWDG_Handle *handle, unsigned int timeValue, IWDG_TimeType timeType)	相同
	unsigned int HAL_IWDG_GetLoadValue(IWDG_Handle *handle)	unsigned int HAL_IWDG_GetLoadValue(IWDG_Handle *handle)	相同
	unsigned int HAL_IWDG_GetCounterValue(IWDG_Handle *handle)	unsigned int HAL_IWDG_GetCounterValue(IWDG_Handle *handle)	相同



功能	3066M/3065P IWDG HAL接口	3065H/3061H WDG HAL接口	是否差异
启停控制	void HAL_IWDG_Refresh(IWDG_Handle *handle)	void HAL_IWDG_Refresh(IWDG_Handle *handle)	相同
	void HAL_IWDG_Start(IWDG_Handle *handle)	void HAL_IWDG_Start(IWDG_Handle *handle)	相同
	void HAL_IWDG_Stop(IWDG_Handle *handle)	void HAL_IWDG_Stop(IWDG_Handle *handle)	相同
中断处理	void HAL_IWDG_IrqHandler(void *handle)	void HAL_IWDG_IrqHandler(void *handle)	相同
注册回调函数	void HAL_IWDG_RegisterCallback(IWDG_Handle *handle, IWDG_CallbackType callBackFunc)	void HAL_IWDG_RegisterCallback(IWDG_Handle *handle, IWDG_CallbackType callBackFunc)	相同
窗口设置	unsigned int HAL_IWDG_GetWindowValueEx(IWDG_Handle *handle)	-	差异
	void HAL_IWDG_EnableWindowModeEx(IWDG_Handle *handle)	-	差异
	void HAL_IWDG_DisableWindowModeEx(IWDG_Handle *handle)	-	差异

表 3-6 3066M/3065P 系列 WWDG 和 3065H/3061H 系列 WDG 接口对比

功能	3066M/3065P WWDG HAL接口	3065H/3061H WDG HAL接口	是否差异
初始化	BASE_StatusType HAL_WWDG_Init(WWDG_Handle *handle)	BASE_StatusType HAL_WDG_Init(WDG_Handle *handle)	差异
读写配置	void HAL_WWDG_SetTimeValue(WWDG_Handle *handle, unsigned int timeValue, WWDG_TimeType timeType)	void HAL_WDG_SetTimeValue(WDG_Handle *handle, unsigned int timeValue, WDG_TimeType timeType)	差异



功能	3066M/3065P WWDG HAL接口	3065H/3061H WDG HAL接口	是否差异
	unsigned int HAL_WWDG_GetLoadValue(WWDG_Handle *handle)	unsigned int HAL_WDG_GetLoadValue(WDG_Handle *handle)	差异
	unsigned int HAL_WWDG_GetWindowValue(WWDG_Handle *handle)	-	差异
	unsigned int HAL_WWDG_GetCounterValue(WWDG_Handle *handle)	unsigned int HAL_WDG_GetCounterValue(WDG_Handle *handle)	差异
启停控制	void HAL_WWDG_Refresh(WWDG_Handle *handle)	void HAL_WDG_Refresh(WDG_Handle *handle)	差异
	void HAL_WWDG_Start(WWDG_Handle *handle)	void HAL_WDG_Start(WDG_Handle *handle)	差异
	void HAL_WWDG_Stop(WWDG_Handle *handle)	void HAL_WDG_Stop(WDG_Handle *handle)	差异
中断处理	void HAL_WWDG_IrqHandler(void *handle)	void HAL_WDG_IrqHandler(void *handle)	差异
注册回调函数	void HAL_WWDG_RegisterCallback(WWDG_Handle *handle, WWDG_CallbackType callBackFunc)	void HAL_WDG_RegisterCallback(WDG_Handle *handle, WDG_CallbackType callBackFunc)	差异
窗口设置	void HAL_WWDG_EnableWindowModeEx(WWDG_Handle *handle)	-	差异
	void HAL_WWDG_DisableWindowModeEx(WWDG_Handle *handle)	-	差异

### 3.4.2 配置参数变更

WDG模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.5 SPI 驱动移植



### 3.5.1 接口移植

表 3-7 3066M/3065P 系列和 3065H/3061H 系列 SPI 接口对比

功能	3066M/3065P SPI HAL接口	3065H/3061H SPI HAL接口	是否差异
初始化和配置	BASE_StatusType HAL_SPI_Init(SPI_Handle *handle)	BASE_StatusType HAL_SPI_Init(SPI_Handle *handle)	相同
	BASE_StatusType HAL_SPI_Deinit(SPI_Handle *handle)	BASE_StatusType HAL_SPI_Deinit(SPI_Handle *handle)	相同
	BASE_StatusType HAL_SPI_ConfigParameter(SPI_Handle *handle)	BASE_StatusType HAL_SPI_ConfigParameter(SPI_Handle *handle)	相同
	BASE_StatusType HAL_SPI_ChipSelectChannelSet(SPI_Handle* handle, SPI_ChipSelectChannel channel)	BASE_StatusType HAL_SPI_ChipSelectChannelSet(SPI_Handle* handle, SPI_ChipSelectChannel channel)	相同
	BASE_StatusType HAL_SPI_ChipSelectChannelGet(SPI_Handle* handle, SPI_ChipSelectChannel *channel)	BASE_StatusType HAL_SPI_ChipSelectChannelGet(SPI_Handle* handle, SPI_ChipSelectChannel *channel)	相同
读写函数	BASE_StatusType HAL_SPI_ReadBlocking(SPI_Handle *handle, unsigned char *rData, unsigned int dataSize, unsigned int timeout)	BASE_StatusType HAL_SPI_ReadBlocking(SPI_Handle *handle, unsigned char *rData, unsigned int dataSize, unsigned int timeout)	相同
	BASE_StatusType HAL_SPI_WriteBlocking(SPI_Handle *handle, unsigned char *wData, unsigned int dataSize, unsigned int timeout)	BASE_StatusType HAL_SPI_WriteBlocking(SPI_Handle *handle, unsigned char *wData, unsigned int dataSize, unsigned int timeout)	相同
	BASE_StatusType HAL_SPI_WriteReadBlocking(SPI_Handle *handle, unsigned char *rData, unsigned char *wData, unsigned int dataSize, unsigned int timeout)	BASE_StatusType HAL_SPI_WriteReadBlocking(SPI_Handle *handle, unsigned char *rData, unsigned char *wData, unsigned int dataSize, unsigned int timeout)	相同
	BASE_StatusType HAL_SPI_ReadIT(SPI_Handle *handle, unsigned char *rData, unsigned int dataSize)	BASE_StatusType HAL_SPI_ReadIT(SPI_Handle *handle, unsigned char *rData, unsigned int dataSize)	相同



功能	3066M/3065P SPI HAL接口	3065H/3061H SPI HAL接口	是否差异
	BASE_StatusType HAL_SPI_WriteIT(SPI_Handle *handle, unsigned char *wData, unsigned int dataSize)	BASE_StatusType HAL_SPI_WriteIT(SPI_Handle *handle, unsigned char *wData, unsigned int dataSize)	相同
	BASE_StatusType HAL_SPI_WriteReadIT(SPI_Handle *handle, unsigned char *rData, unsigned char *wData, unsigned int dataSize)	BASE_StatusType HAL_SPI_WriteReadIT(SPI_Handle *handle, unsigned char *rData, unsigned char *wData, unsigned int dataSize)	相同
	BASE_StatusType HAL_SPI_WriteDMA(SPI_Handle *handle, unsigned char *wData, unsigned int dataSize)	BASE_StatusType HAL_SPI_WriteDMA(SPI_Handle *handle, unsigned char *wData, unsigned int dataSize)	相同
	BASE_StatusType HAL_SPI_ReadDMA(SPI_Handle *handle, unsigned char *rData, unsigned int dataSize)	BASE_StatusType HAL_SPI_ReadDMA(SPI_Handle *handle, unsigned char *rData, unsigned int dataSize)	相同
	BASE_StatusType HAL_SPI_WriteReadDMA(SPI_Handle *handle, unsigned char *rData, unsigned char *wData, unsigned int dataSize)	BASE_StatusType HAL_SPI_WriteReadDMA(SPI_Handle *handle, unsigned char *rData, unsigned char *wData, unsigned int dataSize)	相同
	BASE_StatusType HAL_SPI_DMAStop(SPI_Handle *handle)	BASE_StatusType HAL_SPI_DMAStop(SPI_Handle *handle)	相同
中断 与回 调	void HAL_SPI_IrqHandler( void *handle)	void HAL_SPI_IrqHandler( void *handle)	相同
注册 回调 函数	BASE_StatusType HALSPI_RegisterCallback(SPI_Handle *handle, HAL_SPI_CallbackIDcallbackID, SPI_CallbackFuncType pcallback)	BASE_StatusType HALSPI_RegisterCallback(SPI_Handle *handle, HAL_SPI_CallbackIDcallbackID, SPI_CallbackFuncType pcallback)	相同
片选 配置	BASE_StatusType HAL_SPI_SetChipConfigSelectEx( SPI_Handle *handle, HAL_SPI_CHIP_CONFIG mode)	-	差异
	HAL_SPI_CHIP_CONFIG HAL_SPI_GetChipConfigSelectEx( SPI_Handle *handle)	-	差异



### 3.5.2 配置参数变更

SPI模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.6 GPT 驱动移植

### 3.6.1 接口移植

表 3-8 3066M/3065P 系列和 3065H/3061H 系列 GPT 接口对比

功能	3066M/3065P GPT HAL接口	3065H/3061H GPT HAL接口	是否差异
初始化	BASE_StatusType HAL_GPT_Init(GPT_Handle *handle)	BASE_StatusType HAL_GPT_Init(GPT_Handle *handle)	相同
	-	BASE_StatusType HAL_GPT_Rsplnit(GPT_Handle *handle)	差异
配置	void HAL_GPT_Start(GPT_Handle *handle)	void HAL_GPT_Start(GPT_Handle *handle)	相同
	void HAL_GPT_Stop(GPT_Handle *handle)	void HAL_GPT_Stop(GPT_Handle *handle)	相同
	BASE_StatusType HAL_GPT_Config(GPT_Handle *handle)	BASE_StatusType HAL_GPT_Config(GPT_Handle *handle)	相同
	BASE_StatusType HAL_GPT_GetConfig(GPT_Handle *handle)	BASE_StatusType HAL_GPT_GetConfig(GPT_Handle *handle)	相同
	BASE_StatusType HAL_GPT_SetReferCounterAndAction(GPT_Handle *handle, const GPT_REFER_Cfg *refer)	-	差异
	void HAL_GPT_GetReferCounterAndAction(GPT_Handle *handle, GPT_REFER_Cfg *refer)	-	差异
	BASE_StatusType HAL_GPT_SetCountPeriod(GPT_Handle *handle, unsigned int period)	-	差异
	unsigned int HAL_GPT_GetCountPeriod(GPT_Handle *handle)	-	差异





功能	3066M/3065P GPT HAL接口	3065H/3061H GPT HAL接口	是否差异
	BASE_StatusType HAL_GPT_SetDivFactor(GPT_Handle *handle, unsigned int div)	-	差异
	unsigned int HAL_GPT_GetDivFactor(GPT_Handle *handle)	-	差异
	BASE_StatusType HAL_GPT_SetBufferLoad(GPT_Handle *handle, GPT_SetOption bufferLoad)	-	差异
	unsigned int HAL_GPT_GetBufferLoadStatus(GPT_Handle *handle)	-	差异
	BASE_StatusType HAL_GPT_SetOutFinishInt(GPT_Handle *handle, GPT_SetOption outFinishInt)	-	差异
	BASE_StatusType HAL_GPT_SetPeriodInt(GPT_Handle *handle, GPT_SetOption periodInt)	-	差异
	unsigned int HAL_GPT_GetCounterValueEx(GPT_Handle *handle)	-	差异
	unsigned int HAL_GPT_GetCurrentPWM0NumberEx(GPT_Handle *handle)	-	差异
	BASE_StatusType HAL_GPT_SoftInjOutFinIntEx(GPT_Handle *handle, GPT_SetOption softInjOutFin)	-	差异
	BASE_StatusType HAL_GPT_SoftInjPeriodFinIntEx(GPT_Handle *handle, GPT_SetOption softInjPeriod)	-	差异
中断处理	void HAL_GPT_IrqOutFinishHandler(void *handle)	-	差异
	void HAL_GPT_IrqPeriodHandler(void *handle)	-	差异



功能	3066M/3065P GPT HAL接口	3065H/3061H GPT HAL接口	是否差异
注册回调函数	BASE_StatusType HAL_GPT_RegisterCallBack(GPT_Handle *gptHandle,GPT_CallBackFun_Type typeId, GPT_CallBackFunc pCallback)	-	差异
DMA 触发	BASE_StatusType HAL_GPT_TriggerDMAEnableEx(GPT_Handle *handle, GPT_TriggerDMAType triggerDMAType)	-	差异
	BASE_StatusType HAL_GPT_TriggerDMADisableEx(GPT_Handle *handle, GPT_TriggerDMAType triggerDMAType)	-	差异
ADC 触发	BASE_StatusType HAL_GPT_TriggerADCEnableEx(GPT_Handle *handle, GPT_TriggerADCType triggerADCType)	-	差异
	BASE_StatusType HAL_GPT_TriggerADCDisableEx(GPT_Handle *handle, GPT_TriggerADCType triggerADCType)	-	差异

### 3.6.2 配置参数变更

GPT模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.7 UART 驱动移植

### 3.7.1 接口移植

表 3-9 3066M/3065P 系列和 3065H/3061H 系列 UART 接口对比

功能	3066M/3065P UART HAL接口	3065H/3061H UART HAL接口	是否差异
初始化和去初始化配置	BASE_StatusType HAL_UART_Init(UART_Handle *uartHandle)	BASE_StatusType HAL_UART_Init(UART_Handle *uartHandle)	相同



功能	3066M/3065P UART HAL接口	3065H/3061H UART HAL接口	是否差异
	BASE_StatusType HAL_UART_DeInit(UART_Handle *uartHandle)	BASE_StatusType HAL_UART_DeInit(UART_Handle *uartHandle)	相同
获取UART的状态	UART_State_Type HAL_UART_GetState(UART_Handle *uartHandle)	UART_State_Type HAL_UART_GetState(UART_Handle *uartHandle)	相同
阻塞读写	BASE_StatusType HAL_UART_WriteBlocking(UART_Handle *uartHandle, unsigned char *srcData, unsigned int dataLength, unsigned int blockingTime)	BASE_StatusType HAL_UART_WriteBlocking(UART_Handle *uartHandle, unsigned char *srcData, unsigned int dataLength, unsigned int blockingTime)	相同
	BASE_StatusType HAL_UART_ReadBlocking(UART_Handle *uartHandle, unsigned char *saveData, unsigned int dataLength, unsigned int blockingTime)	BASE_StatusType HAL_UART_ReadBlocking(UART_Handle *uartHandle, unsigned char *saveData, unsigned int dataLength, unsigned int blockingTime)	相同
中断读写	BASE_StatusType HAL_UART_WriteIT(UART_Handle *uartHandle, unsigned char *srcData, unsigned int dataLength)	BASE_StatusType HAL_UART_WriteIT(UART_Handle *uartHandle, unsigned char *srcData, unsigned int dataLength)	相同
	BASE_StatusType HAL_UART_ReadIT(UART_Handle *uartHandle, unsigned char *saveData, unsigned int dataLength)	BASE_StatusType HAL_UART_ReadIT(UART_Handle *uartHandle, unsigned char *saveData, unsigned int dataLength)	相同
DMA读写	BASE_StatusType HAL_UART_WriteDMA(UART_Handle *uartHandle, unsigned char *srcData, unsigned int dataLength)	BASE_StatusType HAL_UART_WriteDMA(UART_Handle *uartHandle, unsigned char *srcData, unsigned int dataLength)	相同
	BASE_StatusType HAL_UART_ReadDMA(UART_Handle *uartHandle, unsigned char *saveData, unsigned int dataLength)	BASE_StatusType HAL_UART_ReadDMA(UART_Handle *uartHandle, unsigned char *saveData, unsigned int dataLength)	相同



功能	3066M/3065P UART HAL接口	3065H/3061H UART HAL接口	是否差异
停止读写	BASE_StatusType HAL_UART_StopRead(UART_Handle *uartHandle)	BASE_StatusType HAL_UART_StopRead(UART_Handle *uartHandle)	相同
	BASE_StatusType HAL_UART_StopWrite(UART_Handle *uartHandle)	BASE_StatusType HAL_UART_StopWrite(UART_Handle *uartHandle)	相同
中断处理函数	void HAL_UART_IrqHandler(void *handle)	void HAL_UART_IrqHandler(void *handle)	相同
	BASE_StatusType HAL_UART_RegisterCallBack(UART_Handle *uartHandle, UART_CallbackFun_Type typeId, UART_CallbackType pCallback)	BASE_StatusType HAL_UART_RegisterCallBack(UART_Handle *uartHandle, UART_CallbackFun_Type typeId, UART_CallbackType pCallback)	相同
DMA 循环搬运	BASE_StatusType HAL_UART_ReadDMAAndCyclicallyStored(UART_Handle *uartHandle, unsigned char *saveData, DMA_LinkList *tempNode, unsigned int dataLength)	BASE_StatusType HAL_UART_ReadDMAAndCyclicallyStored(UART_Handle *uartHandle, unsigned char *saveData, DMA_LinkList *tempNode, unsigned int dataLength)	相同
	unsigned int HAL_UART_ReadDMAGetPos(UART_Handle *uartHandle)	unsigned int HAL_UART_ReadDMAGetPos(UART_Handle *uartHandle)	相同
字符检测	BASE_StatusType HAL_UART_OpenCharacterMatchEx(UART_Handle *uartHandle, unsigned char ch)	-	差异
	BASE_StatusType HAL_UART_CloseCharacterMatchEx(UART_Handle *uartHandle)	-	差异
波特率检测	BASE_StatusType HAL_UART_EnableBaudDetectionEx(UART_Handle *uartHandle)	-	差异



功能	3066M/3065P UART HAL接口	3065H/3061H UART HAL接口	是否差异
	BASE_StatusType HAL_UART_DisableBaudDetectionEx(UART_Handle *uartHandle)	-	差异
设置接收超时时间	BASE_StatusType HAL_UART_SetRxWaitTimeEx(UART_Handle *uartHandle, unsigned int cntOfBit)	-	差异
设置过采样倍数	BASE_StatusType HAL_UART_SetOversampleMultipleEx(UART_Handle *uartHandle, UART_OversampleMultiple multiple)	-	差异
设置字节序	BASE_StatusType HAL_UART_SetDataSequenceModeEx(UART_Handle *uartHandle, UART_SequenceMode mode)	-	差异

说明

3066M/3065P设置的接收超时时间为：当接收FIFO非空，且超过配置的时间没有再接收到新的数据时会触发接收超时中断，注意不是帧的间隔时间。

3.7.2 配置参数变更

UART模块3065H/3061H和3066M/3065P配置参数无变化。

3.8 CMM 驱动移植



### 3.8.1 接口移植

表 3-10 3066M/3065P 系列和 3065H/3061H 系列 CMM 接口对比

功能	3066M/3065P CMM HAL接口	3065H/3061H CMM HAL接口	是否差异
初始化、去初始化和配置	BASE_StatusType HAL_CMM_Init(CMM_Handle*handle)	BASE_StatusType HAL_CMM_Init(CMM_Handle*handle)	相同
	BASE_StatusType HAL_CMM_DeInit(CMM_Handle*handle)	BASE_StatusType HAL_CMM_DeInit(CMM_Handle*handle)	相同
	BASE_StatusType HAL_CMM_Config(CMM_Handle*handle, CMM_CFG_TYPE cfgType)	BASE_StatusType HAL_CMM_Config(CMM_Handle*handle, CMM_CFG_TYPE cfgType)	相同
	void HAL_CMM_GetConfig(CMM_Handle*handle)	void HAL_CMM_GetConfig(CMM_Handle*handle)	相同
开启和关闭时钟	void HAL_CMM_Start(CMM_Handle*handle)	void HAL_CMM_Start(CMM_Handle*handle)	相同
	void HAL_CMM_Stop(CMM_Handle*handle)	void HAL_CMM_Stop(CMM_Handle*handle)	相同
事件中断	void HAL_CMM_IrqHandler(void*handle)	void HAL_CMM_IrqHandler(void*handle)	相同
注册回调函数	BASE_StatusType HAL_CMM_RegisterCallback(CMM_Handle*handle, CMM_Interrupt_Type type, CMM_CallBackFuncType callback)	BASE_StatusType HAL_CMM_RegisterCallback(CMM_Handle*handle, CMM_Interrupt_Type type, CMM_CallBackFuncType callback)	相同

说明

3066M/3065P和3065H/3061H的CMM模块没有区别。

### 3.8.2 配置参数变更

CMM模块3065H/3061H和3066M/3065P配置参数无变化。



## 3.9 CFD 驱动移植

### 3.9.1 接口移植

表 3-11 3066M/3065P 系列和 3065H/3061H 系列 CFD 接口对比

功能	3066M/3065P CFD HAL接口	3065H/3061H CFD HAL接口	是否差异
初始化、去初始化和配置	BASE_StatusType HAL_CFD_Init(CFD_Handle *handle)	BASE_StatusType HAL_CFD_Init(CFD_Handle *handle)	相同
	BASE_StatusType HAL_CFD_DeInit(CFD_Handle *handle)	BASE_StatusType HAL_CFD_DeInit(CFD_Handle *handle)	相同
	BASE_StatusType HAL_CFD_Config(CFD_Handle *handle, CFD_CFG_TYPE cfgType)	BASE_StatusType HAL_CFD_Config(CFD_Handle *handle, CFD_CFG_TYPE cfgType)	相同
	void HAL_CFD_GetConfig(CFD_Handle *handle)	void HAL_CFD_GetConfig(CFD_Handle *handle)	相同
开启和关闭时钟	void HAL_CFD_Start(CFD_Handle *handle)	void HAL_CFD_Start(CFD_Handle *handle)	相同
	void HAL_CFD_Stop(CFD_Handle *handle)	void HAL_CFD_Stop(CFD_Handle *handle)	相同
事件中断	void HAL_CFD_IrqHandler(void *handle)	void HAL_CFD_IrqHandler(void *handle)	相同
注册回调函数	BASE_StatusType HAL_CFD_RegisterCallback(CFD_Handle *handle, CFD_Interrupt_Type type, CFD_CallBackFuncType callback)	BASE_StatusType HAL_CFD_RegisterCallback(CFD_Handle *handle, CFD_Interrupt_Type type, CFD_CallBackFuncType callback)	相同

### 3.9.2 配置参数变更

CFD模块3065H/3061H和3066M/3065P配置参数无变化。



## 3.10 TIMER 驱动移植

### 3.10.1 接口移植

表 3-12 3066M/3065P 系列和 3065H/3061H 系列 TIMER 接口对比

功能	3066M/3065P TIMER HAL接口	3065H/3061H TIMER HAL接口	是否差异
初始化、去初始化和配置	BASE_StatusType HAL_TIMER_Init(TIMER_Handle *handle)	BASE_StatusType HAL_TIMER_Init(TIMER_Handle *handle)	相同
	void HAL_TIMER_DeInit(TIMER_Handle *handle)	void HAL_TIMER_DeInit(TIMER_Handle *handle)	相同
	BASE_StatusType HAL_TIMER_Config(TIMER_Handle *handle, TIMER_CFG_TYPE cfgType)	BASE_StatusType HAL_TIMER_Config(TIMER_Handle *handle, TIMER_CFG_TYPE cfgType)	相同
	BASE_StatusType HAL_TIMER_GetConfig(TIMER_Handle *handle)	BASE_StatusType HAL_TIMER_GetConfig(TIMER_Handle *handle)	相同
开启和关闭时钟	void HAL_TIMER_Start(TIMER_Handle *handle)	void HAL_TIMER_Start(TIMER_Handle *handle)	相同
	void HAL_TIMER_Stop(TIMER_Handle *handle)	void HAL_TIMER_Stop(TIMER_Handle *handle)	相同
事件中断	void HAL_TIMER_IrqHandler(void *handle)	void HAL_TIMER_IrqHandler(void *handle)	相同
注册和注销回调函数	BASE_StatusType HAL_TIMER_RegisterCallback(TIMER_Handle *handle, TIMER_InterruptType typeId, TIMER_CallBackFunc callBackFunc)	BASE_StatusType HAL_TIMER_RegisterCallback(TIMER_Handle *handle, TIMER_InterruptType typeId, TIMER_CallBackFunc callBackFunc)	相同
	BASE_StatusType HAL_TIMER_UnRegisterCallback(TIMER_Handle *handle, TIMER_InterruptType typeId)	BASE_StatusType HAL_TIMER_UnRegisterCallback(TIMER_Handle *handle, TIMER_InterruptType typeId)	相同





功能	3066M/3065P TIMER HAL接口	3065H/3061H TIMER HAL接口	是否差异
后处理功能	void HAL_TIMER_DMARequestOverflowEx(TIMER_Handle *handle, bool overflow)	-	差异
	BASE_StatusType HAL_TIMER_TriggerAdcEx(TIMER_Handle *handle, bool enable)	-	差异

### 3.10.2 配置参数变更

TIMER模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.11 CAPM 驱动移植

### 3.11.1 接口移植

表 3-13 3066M/3065P 系列和 3065H/3061H 系列 CAPM 接口对比

功能	3066M/3065P CAPM HAL接口	3065H/3061H CAPM HAL接口	是否差异
初始化和去初始化	BASE_StatusType HAL_CAPM_Init(CAPM_Handle *handle)	BASE_StatusType HAL_CAPM_Init(CAPM_Handle *handle)	相同
	BASE_StatusType HAL_CAPM_DeInit(CAPM_Handle *handle)	BASE_StatusType HAL_CAPM_DeInit(CAPM_Handle *handle)	相同
获取 ECR 寄存器值和当前信号边沿类型	unsigned int HAL_CAPM_GetECRValue(CAPM_Handle *handle, CAPM_ECRNum ecrNum)	unsigned int HAL_CAPM_GetECRValue(CAPM_Handle *handle, CAPM_ECRNum ecrNum)	相同
	unsigned char HAL_CAPM_GetCrtEdge(CAPM_Handle *handle)	unsigned char HAL_CAPM_GetCrtEdge(CAPM_Handle *handle)	相同
	unsigned char HAL_CAPM_GetNextLoadECRNum(CAPM_Handle *handle)	unsigned char HAL_CAPM_GetNextLoadECRNum(CAPM_Handle *handle)	相同



功能	3066M/3065P CAPM HAL接口	3065H/3061H CAPM HAL接口	是否差异
	BASE_StatusType HAL_CAPM_GetECRValueDMA(CAPM_Handle *handle, unsigned int *saveData, unsigned int dataLength)	BASE_StatusType HAL_CAPM_GetECRValueDMA(CAPM_Handle *handle, unsigned int *saveData, unsigned int dataLength)	相同
同步相位配置	void HAL_CAPM_SetSyncPhs(CAPM_Handle *handle, unsigned int phase)	void HAL_CAPM_SetSyncPhs(CAPM_Handle *handle, unsigned int phase)	相同
	unsigned int HAL_CAPM_GetSyncPhs(CAPM_Handle *handle)	unsigned int HAL_CAPM_GetSyncPhs(CAPM_Handle *handle)	相同
事件中断	void HAL_CAPM_IrqHandler(void *handle)	void HAL_CAPM_IrqHandler(void *handle)	相同
注册回调函数	void HAL_CAPM_RegisterCallback(CAPM_Handle *capmHandle, CAPM_CallbackFuncType typeId, CAPM_CallbackType pCallback)	void HAL_CAPM_RegisterCallback(CAPM_Handle *capmHandle, CAPM_CallbackFuncType typeId, CAPM_CallbackType pCallback)	相同
			相同

#### 说明

3066M/3065P和3065H/3061H的CAPM模块没有区别。

### 3.11.2 配置参数变更

CAPM模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.12 QDM 驱动移植



### 3.12.1 接口移植

表 3-14 3066M/3065P 系列和 3065H/3061H 系列 QDM 接口对比

功能	3066M/3065P QDM HAL接口	3065H/3061H QDM HAL接口	是否差异
初始化和去初始化	BASE_StatusType HAL_QDM_Init(QDM_Handle *qdmHandle)	BASE_StatusType HAL_QDM_Init(QDM_Handle *qdmHandle)	相同
	BASE_StatusType HAL_QDM_DeInit(QDM_Handle *qdmHandle)	BASE_StatusType HAL_QDM_DeInit(QDM_Handle *qdmHandle)	相同
读取当前转动电机的相位错误状态	void HAL_QDM_GetPhaseErrorStatus(const QDM_Handle *qdmHandle, unsigned int *errStatus)	void HAL_QDM_GetPhaseErrorStatus(const QDM_Handle *qdmHandle, unsigned int *errStatus)	相同
获取当前电机转动位置值和方向	void HAL_QDM_ReadPosCountAndDir(const QDM_Handle *qdmHandle, unsigned int *count, unsigned int *dir)	void HAL_QDM_ReadPosCountAndDir(const QDM_Handle *qdmHandle, unsigned int *count, unsigned int *dir)	相同
获取电机转动速度计算方法	int HAL_QDM_GetSpeedRpmM(QDM_Handle *qdmHandle)	int HAL_QDM_GetSpeedRpmM(QDM_Handle *qdmHandle)	相同
	int HAL_QDM_GetSpeedRpmMT(QDM_Handle *qdmHandle)	int HAL_QDM_GetSpeedRpmMT(QDM_Handle *qdmHandle)	相同
事件中断	void HAL_QDM_IrqHandler(void *handle)	void HAL_QDM_IrqHandler(void *handle)	相同
注册和注销回调函数	void HAL_QDM_RegisterCallback(QDM_Handle *qdmHandle, QDM_CallbackFuncType typeId, QDM_CallbackType pCallback)	void HAL_QDM_RegisterCallback(QDM_Handle *qdmHandle, QDM_CallbackFuncType typeId, QDM_CallbackType pCallback)	相同



说明

3066M/3065P和3065H/3061H的QDM模块没有区别。

3.12.2 配置参数变更

QDM模块3065H/3061H和3066M/3065P配置参数无变化。

3.13 Flash 驱动移植

3.13.1 接口移植

表 3-15 3066M/3065P 系列和 3065H/3061H 系列 Flash 接口对比

功能	3066M/3065P Flash HAL接口	3065H/3061H Flash HAL接口	是否差异
初始化和配置	BASE_StatusType HAL_FLASH_Init(FLASH_Handle *handle)	BASE_StatusType HAL_FLASH_Init(FLASH_Handle *handle)	相同
	BASE_StatusType HAL_FLASH_DeInit(FLASH_Handle *handle)	BASE_StatusType HAL_FLASH_DeInit(FLASH_Handle handle)	相同
注册回调函数	BASE_StatusType HAL_FLASH_RegisterCallback(FLASH_Handle *handle, FLASH_CallbackFunType pcallback)	BASE_StatusType HAL_FLASH_RegisterCallback(FLASH_Handle *handle, FLASH_CallbackFunType pcallback)	相同
编程与擦除	BASE_StatusType HAL_FLASH_WriteBlocking(FLASH_Handle *handle, unsigned int srcAddr, unsigned int destAddr, unsigned int srcLen)	BASE_StatusType HAL_FLASH_WriteBlocking(FLASH_Handle *handle, unsigned int srcAddr, unsigned int destAddr, unsigned int srcLen)	相同
	BASE_StatusType HAL_FLASH_EraseBlocking(FLASH_Handle *handle, FLASH_EraseMode eraseMode, FLASH_SectorAddr startAddr, unsigned int eraseNum)	BASE_StatusType HAL_FLASH_EraseBlocking(FLASH_Handle *handle, FLASH_EraseMode eraseMode, FLASH_SectorAddr startAddr, unsigned int eraseNum)	相同



功能	3066M/3065P Flash HAL接口	3065H/3061H Flash HAL接口	是否差异
	BASE_StatusType HAL_FLASH_WriteIT(FLASH_Handle *handle, unsigned int srcAddr, unsigned int destAddr, unsigned int srcLen)	BASE_StatusType HAL_FLASH_WriteIT(FLASH_Handle *handle, unsigned int srcAddr, unsigned int destAddr, unsigned int srcLen)	相同
	BASE_StatusType HAL_FLASH_EraseIT(FLASH_Handle *handle, FLASH_EraseMode eraseMode, FLASH_SectorAddr startAddr, unsigned int eraseNum)	BASE_StatusType HAL_FLASH_EraseIT(FLASH_Handle *handle, FLASH_EraseMode eraseMode, FLASH_SectorAddr startAddr, unsigned int eraseNum)	相同
读取	BASE_StatusType HAL_FLASH_Read(FLASH_Handle *handle, unsigned int srcAddr, unsigned int readLen, unsigned char *dataBuff, unsigned int buffLen)	BASE_StatusType HAL_FLASH_Read(FLASH_Handle *handle, unsigned int srcAddr, unsigned int readLen, unsigned char *dataBuff, unsigned int buffLen)	相同
中断与回调	void HAL_FLASH_IrqHandler(void *handle)	void HAL_FLASH_IrqHandler(void *handle)	相同
	void HAL_FLASH_IrqHandlerError(void *handle)	void HAL_FLASH_IrqHandlerError(void *handle)	相同

### 3.13.2 配置参数变更

Flash模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.14 CRG 驱动移植



### 3.14.1 接口移植

表 3-16 3066M/3065P 系列和 3065H/3061H 系列 CRG 接口对比

功能	3066M/3065P CRG HAL接口	3065H/3061H CRG HAL接口	是否差异
初始化和配置	BASE_StatusType HAL_CRG_Init(const CRG_Handle *handle)	BASE_StatusType HAL_CRG_Init(const CRG_Handle *handle)	相同
	BASE_StatusType HAL_CRG_DeInit(const CRG_Handle *handle)	BASE_StatusType HAL_CRG_DeInit(const CRG_Handle *handle)	相同
获取 PLL 配置参数	BASE_StatusType HAL_CRG_GetConfig(CRG_Handle *handle)	BASE_StatusType HAL_CRG_GetConfig(CRG_Handle *handle)	相同
设置 CPU 工作时钟源	BASE_StatusType HAL_CRG_SetCoreClockSelect(CRG_Handle *handle)	BASE_StatusType HAL_CRG_SetCoreClockSelect(CRG_Handle *handle)	相同
根据目标频率配置 PLL 参数	BASE_StatusType HAL_CRG_InitWithTargetFrequency(const CRG_Handle *handle, unsigned int targetFreq)	BASE_StatusType HAL_CRG_InitWithTargetFrequency(const CRG_Handle *handle, unsigned int targetFreq)	相同
获取和设置模块时钟门控	BASE_StatusType HAL_CRG_IpEnableSet(const void *baseAddress, unsigned int enable)	BASE_StatusType HAL_CRG_IpEnableSet(const void *baseAddress, unsigned int enable)	相同
	BASE_StatusType HAL_CRG_IpEnableGet(const void *baseAddress, unsigned int *enable)	BASE_StatusType HAL_CRG_IpEnableGet(const void *baseAddress, unsigned int *enable)	相同
获取和设置模块时钟源	BASE_StatusType HAL_CRG_IpClkSelectSet(const void *baseAddress, unsigned int select)	BASE_StatusType HAL_CRG_IpClkSelectSet(const void *baseAddress, unsigned int select)	相同
	BASE_StatusType HAL_CRG_IpClkSelectGet(const void *baseAddress, unsigned int *select)	BASE_StatusType HAL_CRG_IpClkSelectGet(const void *baseAddress, unsigned int *select)	相同
获取和设置模块时钟复位	BASE_StatusType HAL_CRG_IpClkResetSet(const void *baseAddress, unsigned int reset)	BASE_StatusType HAL_CRG_IpClkResetSet(const void *baseAddress, unsigned int reset)	相同



功能	3066M/3065P CRG HAL接口	3065H/3061H CRG HAL接口	是否差异
	BASE_StatusType HAL_CRG_IpClkResetGet(const void *baseAddress, unsigned int *reset)	BASE_StatusType HAL_CRG_IpClkResetGet(const void *baseAddress, unsigned int *reset)	相同
获取和设置模块时钟分频	BASE_StatusType HAL_CRG_IpClkDivSet(const void *baseAddress, unsigned int div)	BASE_StatusType HAL_CRG_IpClkDivSet(const void *baseAddress, unsigned int div)	相同
	BASE_StatusType HAL_CRG_IpClkDivGet(const void *baseAddress, unsigned int *div)	BASE_StatusType HAL_CRG_IpClkDivGet(const void *baseAddress, unsigned int *div)	相同
设置PVD复位	void HAL_CRG_PvdResetEnable(bool enable)	void HAL_CRG_PvdResetEnable(bool enable)	相同
获取PLL输出频率	unsigned int HAL_CRG_GetPllFreq(void)	unsigned int HAL_CRG_GetPllFreq(void)	相同
获取CPU时钟频率	unsigned int HAL_CRG_GetCoreClkFreq(void)	unsigned int HAL_CRG_GetCoreClkFreq(void)	相同
获取模块时钟频率	unsigned int HAL_CRG_GetIpFreq(const void *ipBaseAddr)	unsigned int HAL_CRG_GetIpFreq(const void *ipBaseAddr)	相同

### 3.14.2 配置参数变更

CRG模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.15 GPIO 驱动移植

### 3.15.1 接口移植

表 3-17 3066M/3065P 系列和 3065H/3061H 系列 GPIO 接口对比

功能	3066M/3065P GPIO HAL接口	3065H/3061H GPIO HAL接口	是否差异
初始化和配置	void HAL_GPIO_Init(GPIO_Handle *handle)	void HAL_GPIO_Init(GPIO_Handle *handle)	相同



功能	3066M/3065P GPIO HAL接口	3065H/3061H GPIO HAL接口	是否差异
	void HAL_GPIO_DeInit(GPIO_Handle *handle)	void HAL_GPIO_DeInit(GPIO_Handle *handle)	相同
设置 GPIO管脚的电平、方向、中断类型	void HAL_GPIO_SetDirection(GPIO_Handle *handle, unsigned int pins, GPIO_Direction dir)	void HAL_GPIO_SetDirection(GPIO_Handle *handle, unsigned int pins, GPIO_Direction dir)	相同
	void HAL_GPIO_SetValue(GPIO_Handle *handle, unsigned int pins, GPIO_Value value)	void HAL_GPIO_SetValue(GPIO_Handle *handle, unsigned int pins, GPIO_Value value)	相同
	BASE_StatusType HAL_GPIO_SetIrqType(GPIO_Handle *handle, unsigned int pins, GPIO_InterruptMode mode)	BASE_StatusType HAL_GPIO_SetIrqType(GPIO_Handle *handle, unsigned int pins, GPIO_InterruptMode mode)	相同
获取 GPIO管脚的电平、方向、中断类型	GPIO_InterruptMode HAL_GPIO_GetPinIrqType(GPIO_Handle *handle, GPIO_PIN pin)	GPIO_InterruptMode HAL_GPIO_GetPinIrqType(GPIO_Handle *handle, GPIO_PIN pin)	相同
	GPIO_Value HAL_GPIO_GetPinValue(GPIO_Handle *handle, GPIO_PIN pin)	GPIO_Value HAL_GPIO_GetPinValue(GPIO_Handle *handle, GPIO_PIN pin)	相同
	GPIO_Direction HAL_GPIO_GetPinDirection(GPIO_Handle *handle, GPIO_PIN pin)	GPIO_Direction HAL_GPIO_GetPinDirection(GPIO_Handle *handle, GPIO_PIN pin)	相同
获取同组GPIO所有管脚方向、电平	unsigned int HAL_GPIO_GetAllDirection(GPIO_Handle *handle)	unsigned int HAL_GPIO_GetAllDirection(GPIO_Handle *handle)	相同
	unsigned int HAL_GPIO_GetAllValue(GPIO_Handle *handle)	unsigned int HAL_GPIO_GetAllValue(GPIO_Handle *handle)	相同
翻转同组GPIO所有管脚电平	void HAL_GPIO_TogglePin(GPIO_Handle *handle, unsigned int pins)	void HAL_GPIO_TogglePin(GPIO_Handle *handle, unsigned int pins)	相同





功能	3066M/3065P GPIO HAL接口	3065H/3061H GPIO HAL接口	是否差异
注册回调函数	void HAL_GPIO_RegisterCallBack(GPIO_Handle *handle, GPIO_PIN pin, GPIO_CallbackType pCallback)	void HAL_GPIO_RegisterCallBack(GPIO_Handle *handle, GPIO_PIN pin, GPIO_CallbackType pCallback)	相同
中断与回调	void HAL_GPIO_IrqHandler(void *handle)	void HAL_GPIO_IrqHandler(void *handle)	相同

### 3.15.2 参数配置变更

GPIO模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.16 PMC 驱动移植

### 3.16.1 接口移植

表 3-18 3066M/3065P 系列和 3065H/3061H 系列 PMC 接口对比

功能	3066M/3065P PMC HAL接口	3065H/3061H PMC HAL接口	是否差异
初始化和配置	void HAL_PMC_Init(PMC_Handle *handle)	void HAL_PMC_Init(PMC_Handle *handle)	相同
	void HAL_PMC_DeInit(PMC_Handle *handle)	void HAL_PMC_DeInit(PMC_Handle *handle)	相同
进入低功耗模式	void HAL_PMC_EnterSleepMode(void)	void HAL_PMC_EnterSleepMode(void)	相同
	void HAL_PMC_EnterDeepSleepMode(PMC_Handle *handle)	void HAL_PMC_EnterDeepSleepMode(PMC_Handle *handle)	相同
	-	void HAL_PMC_EnterShutdownMode(PMC_Handle *handle)	差异



功能	3066M/3065P PMC HAL接口	3065H/3061H PMC HAL接口	是否差异
获取唤醒自何种低功耗类型	PMC_LowpowerType HAL_PMC_GetWakeupType(PMC_Handle *handle)	PMC_LowpowerType HAL_PMC_GetWakeupType(PMC_Handle *handle)	相同
注册回调函数	void HAL_PMC_RegisterCallback(PMC_Handle *handle, PMC_CallbackID callbackID, PMC_CallbackType pCallback)	void HAL_PMC_RegisterCallback(PMC_Handle *handle, PMC_CallbackID callbackID, PMC_CallbackType pCallback)	相同
中断与回调	void HAL_PMC_IrqHandler(void *handle)	void HAL_PMC_IrqHandler(void *handle)	相同

### 3.16.2 参数配置变更

PMC模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.17 I2C 驱动移植

### 3.17.1 接口移植

表 3-19 3066M/3065P 系列和 3065H/3061H 系列 I2C 接口对比

功能	3066M/3065P I2C HAL接口	3065H/3061H I2C HAL接口	是否差异
初始化和配置	BASE_StatusType HAL_I2C_Init(I2C_Handle *handle)	BASE_StatusType HAL_I2C_Init(I2C_Handle *handle)	相同
	BASE_StatusType HAL_I2C_Deinit(I2C_Handle *handle)	BASE_StatusType HAL_I2C_Deinit(I2C_Handle *handle)	相同
注册回调函数	BASE_StatusType HAL_I2C_RegisterCallback(I2C_Handle *handle, I2C_CallbackId callbackID, I2C_CallbackFunType pcallback)	BASE_StatusType HAL_I2C_RegisterCallback(I2C_Handle *handle, I2C_CallbackId callbackID, I2C_CallbackFunType pcallback)	相同



功能	3066M/3065P I2C HAL接口	3065H/3061H I2C HAL接口	是否差异
数据接收和发送	BASE_StatusType HAL_I2C_MasterReadBlocking(I2C_Handle *handle, unsigned short devAddr, unsigned char *rData, unsigned int dataSize, unsigned int timeout)	BASE_StatusType HAL_I2C_MasterReadBlocking(I2C_Handle *handle, unsigned short devAddr, unsigned char *rData, unsigned int dataSize, unsigned int timeout)	相同
	BASE_StatusType HAL_I2C_MasterWriteBlocking(I2C_Handle *handle, unsigned short devAddr, unsigned char *wData, unsigned int dataSize, unsigned int timeout)	BASE_StatusType HAL_I2C_MasterWriteBlocking(I2C_Handle *handle, unsigned short devAddr, unsigned char *wData, unsigned int dataSize, unsigned int timeout)	相同
	BASE_StatusType HAL_I2C_SlaveReadBlocking(I2C_Handle *handle, unsigned char *rData, unsigned int dataSize, unsigned int timeout)	-	差异
	BASE_StatusType HAL_I2C_SlaveWriteBlocking(I2C_Handle *handle, unsigned char *wData, unsigned int dataSize, unsigned int timeout)	-	差异
	BASE_StatusType HAL_I2C_MasterReadIT(I2C_Handle *handle, unsigned short devAddr, unsigned char *rData, unsigned int dataSize)	BASE_StatusType HAL_I2C_MasterReadIT(I2C_Handle *handle, unsigned short devAddr, unsigned char *rData, unsigned int dataSize)	相同
	BASE_StatusType HAL_I2C_MasterWriteIT(I2C_Handle *handle, unsigned short devAddr, unsigned char *wData, unsigned int dataSize)	BASE_StatusType HAL_I2C_MasterWriteIT(I2C_Handle *handle, unsigned short devAddr, unsigned char *wData, unsigned int dataSize)	相同
	BASE_StatusType HAL_I2C_SlaveReadIT(I2C_Handle *handle, unsigned char *rData, unsigned int dataSize)	-	差异
	BASE_StatusType HAL_I2C_SlaveWriteIT(I2C_Handle *handle, unsigned char *wData, unsigned int dataSize)	-	差异



功能	3066M/3065P I2C HAL接口	3065H/3061H I2C HAL接口	是否差异
	BASE_StatusType HAL_I2C_MasterReadDMA(I2C_Handle *handle, unsigned short devAddr, unsigned char *rData, unsigned int dataSize)	BASE_StatusType HAL_I2C_MasterReadDMA(I2C_Handle *handle, unsigned short devAddr, unsigned char *rData, unsigned int dataSize)	相同
	BASE_StatusType HAL_I2C_MasterWriteDMA(I2C_Handle *handle, unsigned short devAddr, unsigned char *wData, unsigned int dataSize)	BASE_StatusType HAL_I2C_MasterWriteDMA(I2C_Handle *handle, unsigned short devAddr, unsigned char *wData, unsigned int dataSize)	相同
	BASE_StatusType HAL_I2C_SlaveReadDMA(I2C_Handle *handle, unsigned char *rData, unsigned int dataSize)	-	差异
	BASE_StatusType HAL_I2C_SlaveWriteDMA(I2C_Handle *handle, unsigned char *wData, unsigned int dataSize)	-	差异
中断与回调	void HAL_I2C_IrqHandler(void *handle)	void HAL_I2C_IrqHandler(void *handle)	相同
传输字节序	BASE_StatusType HAL_I2C_SetDataTransferSequenceEx(I2C_Handle *handle, I2C_DataTransferSequenceType sequence)	-	差异
SCL时钟延长	BASE_StatusType HAL_I2C_SetSclStretchModeEx(I2C_Handle *handle, I2C_ClockStretchType clkStretch)	-	差异
设置SCL低电平超时阈值	BASE_StatusType HAL_I2C_SetSclLowTimeoutEx(I2C_Handle *handle, unsigned int sclLowTimeout)	-	差异
设置总线空闲阈值	BASE_StatusType HAL_I2C_SetBusFreeTimeEx(I2C_Handle *handle, unsigned int busFreeTime)	-	差异



功能	3066M/3065P I2C HAL接口	3065H/3061H I2C HAL接口	是否差异
接收特殊地址和字节	BASE_StatusType HAL_I2C_Set10BitSlaveEnableEx(I2C_Handle *handle)	-	差异
	BASE_StatusType HAL_I2C_SetDeviceIdAddressEnableEx(I2C_Handle *handle)	-	差异
	BASE_StatusType HAL_I2C_SetStartByteEnableEx(I2C_Handle *handle)	-	差异
地址匹配掩码	BASE_StatusType HAL_I2C_SetOwnAddressMaskEx(I2C_Handle *handle, unsigned int addrMask)	-	差异
	BASE_StatusType HAL_I2C_SetOwnXmbAddressMaskEx(I2C_Handle *handle, unsigned int addrMask)	-	差异

### 3.17.2 参数配置变更

I2C模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.18 ACMP 驱动移植

### 3.18.1 接口移植

表 3-20 3066M/3065P 系列和 3065H/3061H 系列 ACMP 接口对比

功能	3066M/3065P ACMP HAL接口	3065H/3061H ACMP HAL接口	是否差异
初始化和去初始化配置	BASE_StatusType HAL_ACMP_Init(ACMP_Handle *acmpHandle)	BASE_StatusType HAL_ACMP_Init(ACMP_Handle *acmpHandle)	相同
	BASE_StatusType HAL_ACMP_DeInit(ACMP_Handle *acmpHandle)	BASE_StatusType HAL_ACMP_DeInit(ACMP_Handle *acmpHandle)	相同
迟滞电压配置	void HAL_ACMP_SetHystVol(ACMP_Handle *acmpHandle, ACMP_HystVol voltage)	void HAL_ACMP_SetHystVol(ACMP_Handle *acmpHandle, ACMP_HystVol voltage)	相同



功能	3066M/3065P ACMP HAL接口	3065H/3061H ACMP HAL接口	是否差异
屏蔽使能配置	void HAL_ACMP_BlkingValid(ACMP_Handle *acmpHandle)	void HAL_ACMP_BlkingValid(ACMP_Handle *acmpHandle)	相同
	void HAL_ACMP_BlkingInvalid(ACMP_Handle *acmpHandle)	void HAL_ACMP_BlkingInvalid(ACMP_Handle *acmpHandle)	相同
比较输出结果选择	BASE_StatusType HAL_ACMP_ResultSelect(ACMP_Handle *acmpHandle, ACMP_ResultSelect resultSelect)	BASE_StatusType HAL_ACMP_ResultSelect(ACMP_Handle *acmpHandle, ACMP_ResultSelect resultSelect)	相同
中断服务函数	void HAL_ACMP_IrqHandler(void *handle)	void HAL_ACMP_IrqHandler(void *handle)	相同
中断注册函数	BASE_StatusType HAL_ACMP_RegisterCallBack(ACMP_Handle *uartHandle, ACMP_CallBackFun_Type typeId, ACMP_CallBackType pCallback)	BASE_StatusType HAL_ACMP_RegisterCallBack(ACMP_Handle *uartHandle, ACMP_CallBackFun_Type typeId, ACMP_CallBackType pCallback)	相同
TRIM寄存器配置	BASE_StatusType HAL_ACMP_SetTrimValueEx(ACMP_Handle *acmpHandle, unsigned char trimValue)	不涉及	差异

### 3.18.2 参数配置变更

ACMP模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.19 DAC 驱动移植



### 3.19.1 接口移植

表 3-21 3066M/3065P 系列和 3065H/3061H 系列 DAC 接口对比

功能	3066M/3065P DAC HAL接口	3065H/3061H DAC HAL接口	是否差异
初始化和去初始化配置	BASE_StatusType HAL_DAC_Init(DAC_Handle *dacHandle)	BASE_StatusType HAL_DAC_Init(DAC_Handle *dacHandle)	相同
	BASE_StatusType HAL_DAC_DeInit(DAC_Handle *dacHandle)	BASE_StatusType HAL_DAC_DeInit(DAC_Handle *dacHandle)	相同
转换值配置	void HAL_DAC_SetValue(DAC_Handle *dacHandle, unsigned int value)	void HAL_DAC_SetValue(DAC_Handle *dacHandle, unsigned int value)	相同
正弦模式配置	-	BASE_StatusType HAL_DAC_SetSineModeEx(DAC_Handle *dacHandle, unsigned short intervalValue)	差异
	-	void HAL_DAC_DisableSineModeEx(DAC_Handle *dacHandle)	差异

### 3.19.2 参数配置变更

DAC模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.20 PGA 驱动移植

### 3.20.1 接口移植

3066M/3065P系列和3065H/3061H系列的PGA模块无差异。

表 3-22 3066M/3065P 系列和 3065H/3061H 系列 PGA 接口对比

功能	3066M/3065P PGA HAL接口	3065H/3061H PGA HAL接口	是否差异
初始化和去初始化配置	BASE_StatusType HAL_PGA_Init(PGA_Handle *pgaHandle)	BASE_StatusType HAL_PGA_Init(PGA_Handle *pgaHandle)	相同



功能	3066M/3065P PGA HAL接口	3065H/3061H PGA HAL接口	是否差异
	BASE_StatusType HAL_PGA_DeInit(PGA_Handle *pgaHandle)	BASE_StatusType HAL_PGA_DeInit(PGA_Handle *pgaHandle)	相同
配置 运放 倍数	void HAL_PGA_SetGain(PGA_Handle *pgaHandle, PGA_GainValue gain)	void HAL_PGA_SetGain(PGA_Handle *pgaHandle, PGA_GainValue gain)	相同
PGA 使能 配置	void HAL_PGA_Start(PGA_Handle *pgaHandle)	void HAL_PGA_Start(PGA_Handle *pgaHandle)	相同
	void HAL_PGA_Stop(PGA_Handle *pgaHandle)	void HAL_PGA_Stop(PGA_Handle *pgaHandle)	相同

### 3.20.2 参数配置变更

PGA模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.21 APT 驱动移植

### 3.21.1 接口移植

表 3-23 3066M/3065P 系列和 3065H/3061H 系列 APT 接口对比

功能	3066M/3065P APT HAL接口	3065H/3061H APT HAL接口	是否差异
初始化 和去初 始化配 置	BASE_StatusType HAL_APT_PWMInit(APT_Handle *aptHandle)	BASE_StatusType HAL_APT_PWMInit(APT_Handle *aptHandle)	相同
	BASE_StatusType HAL_APT_PWMDelInit(APT_Handle *aptHandle)	BASE_StatusType HAL_APT_PWMDelInit(APT_Handle *aptHandle)	相同
APT使 能配置	void HAL_APT_StartModule(unsigned int aptRunMask)	void HAL_APT_StartModule(unsigned int aptRunMask)	相同
	void HAL_APT_StopModule(unsigned int aptRunMask)	void HAL_APT_StopModule(unsigned int aptRunMask)	相同





功能	3066M/3065P APT HAL接口	3065H/3061H APT HAL接口	是否差异
APT保护配置	BASE_StatusType HAL_APT_ProtectInit(APT_Handle *aptHandle, APT_OutCtrlProtect *protect)	BASE_StatusType HAL_APT_ProtectInit(APT_Handle *aptHandle, APT_OutCtrlProtect *protect)	相同
	BASE_StatusType HAL_APT_ProtectDeInit(APT_Handle *aptHandle, APT_OutCtrlProtect *protect)	BASE_StatusType HAL_APT_ProtectDeInit(APT_Handle *aptHandle, APT_OutCtrlProtect *protect)	相同
	BASE_StatusType HAL_APT_ProtectInitEx(APT_Handle *aptHandle, APT_OutCtrlProtectEx *protect)	BASE_StatusType HAL_APT_ProtectInitEx(APT_Handle *aptHandle, APT_OutCtrlProtectEx *protect)	相同
	BASE_StatusType HAL_APT_ProtectDeInitEx(APT_Handle *aptHandle, APT_OutCtrlProtectEx *protect)	BASE_StatusType HAL_APT_ProtectDeInitEx(APT_Handle *aptHandle, APT_OutCtrlProtectEx *protect)	相同
强制输出低电平	void HAL_APT_ForcePWMOutputLow(APT_Handle *aptHandle)	void HAL_APT_ForcePWMOutputLow(APT_Handle *aptHandle)	相同
同步输出配置	BASE_StatusType HAL_APT_MasterSyncInit(APT_Handle *aptHandle, unsigned short syncOutSrc)	BASE_StatusType HAL_APT_MasterSyncInit(APT_Handle *aptHandle, unsigned short syncOutSrc)	相同
	BASE_StatusType HAL_APT_SlaveSyncInit(APT_Handle *aptHandle, APT_SlaveSyncIn *slaveSyncIn)	BASE_StatusType HAL_APT_SlaveSyncInit(APT_Handle *aptHandle, APT_SlaveSyncIn *slaveSyncIn)	相同
占空比配置	BASE_StatusType HAL_APT_SetPWMDuty(APT_Handle *aptHandle, unsigned short cntCmpLeftEdge, unsigned short cntCmpRightEdge)	BASE_StatusType HAL_APT_SetPWMDuty(APT_Handle *aptHandle, unsigned short cntCmpLeftEdge, unsigned short cntCmpRightEdge)	相同
	BASE_StatusType HAL_APT_SetPWMDutyByNumber(APT_Handle *aptHandle, unsigned int duty)	BASE_StatusType HAL_APT_SetPWMDutyByNumber(APT_Handle *aptHandle, unsigned int duty)	相同



功能	3066M/3065P APT HAL接口	3065H/3061H APT HAL接口	是否差异
ADC采样触发配置	BASE_StatusType HAL_APT_SetADCTriggerTime( APT_Handle *aptHandle, unsigned short cntCmpSOCA, unsigned short cntCmpSOCB)	BASE_StatusType HAL_APT_SetADCTriggerTime( APT_Handle *aptHandle, unsigned short cntCmpSOCA, unsigned short cntCmpSOCB)	相同
中断服务函数	void HAL_APT_EventIrqHandler(void *handle)	void HAL_APT_EventIrqHandler(v oid *handle)	相同
	void HAL_APT_TimerIrqHandler(void *handle)	void HAL_APT_TimerIrqHandler(v oid *handle)	相同
中断函数注册	void HAL_APT_RegisterCallBack(APT _Handle *aptHandle, APT_InterruputType typeId, APT_CallbackType pCallback)	void HAL_APT_RegisterCallBack(A PT_Handle *aptHandle, APT_InterruputType typeId, APT_CallbackType pCallback)	相同
事件管理	BASE_StatusType HAL_APT_EMInit(APT_Handle *aptHandle, APT_EventManage *eventManage)	BASE_StatusType HAL_APT_EMInit(APT_Handl e *aptHandle, APT_EventManage *eventManage)	相同
	unsigned short HAL_APT_EMGetCapValue(APT _Handle *aptHandle)	unsigned short HAL_APT_EMGetCapValue(A PT_Handle *aptHandle)	差异
	void HAL_APT_EMSetWdOffsetAnd Width(APT_Handle *aptHandle, unsigned short offset, unsigned short width)	void HAL_APT_EMSetWdOffsetAn dWidth(APT_Handle *aptHandle, unsigned short offset, unsigned short width)	差异
	void HAL_APT_EMSetValleySwitchSo ftDelay(APT_Handle *aptHandle, unsigned short calibrate)	void HAL_APT_EMSetValleySwitch SoftDelay(APT_Handle *aptHandle, unsigned short calibrate)	差异
更改 PWM 输出动作	BASE_StatusType HAL_APT_ChangeOutputType(A PT_Handle *aptHandle, APT_PWMChannel channel, APT_PWMChannelOutType aptAction)	BASE_StatusType HAL_APT_ChangeOutputTyp e(APT_Handle *aptHandle, APT_PWMChannel channel, APT_PWMChannelOutType aptAction)	相同



功能	3066M/3065P APT HAL接口	3065H/3061H APT HAL接口	是否差异
更改参考点	BASE_StatusType APT_ConfigRefA(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	BASE_StatusType APT_ConfigRefA(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	相同
	BASE_StatusType APT_ConfigRefB(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	BASE_StatusType APT_ConfigRefB(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	相同
	BASE_StatusType APT_ConfigRefC(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	BASE_StatusType APT_ConfigRefC(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	相同
	BASE_StatusType APT_ConfigRefD(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	BASE_StatusType APT_ConfigRefD(APT_Handle *aptHandle, APT_RefDotParameters *refDotParameters)	相同
	BASE_StatusType HAL_APT_ConfigRefDot(APT_H andle *aptHandle, APT_RefDotSelect refDotSelect, APT_RefDotParameters *refDotParameters)	BASE_StatusType HAL_APT_ConfigRefDot(APT_ Handle *aptHandle, APT_RefDotSelect refDotSelect, APT_RefDotParameters *refDotParameters)	相同
	BASE_StatusType HAL_APT_SetTimerPeriod(APT_ Handle *aptHandle, unsigned short newPeriod, APT_BufferLoadMode prdLoadMode, unsigned int prdLoadEvt)	BASE_StatusType HAL_APT_SetTimerPeriod(AP T_Handle *aptHandle, unsigned short newPeriod, APT_BufferLoadMode prdLoadMode, unsigned int prdLoadEvt)	相同
扩展接口	APT_PoeStatus HAL_APT_GetPoeStatus(APT_H andle *aptHandle, APT_POEx poex)	-	差异
	APT_PwmStatus HAL_APT_GetPwmStatus(APT_ Handle *aptHandle, APT_PWMChannel pwmChannel)	-	差异



说明

3066M/3065P不支持谷底开关等配置，其内部函数接口：  
HAL\_APT\_EMGetCapValue(APT\_Handle \*aptHandle)、  
HAL\_APT\_EMSetWdOffsetAndWidth(APT\_Handle \*aptHandle, unsigned short offset,  
unsigned short width)、HAL\_APT\_EMSetValleySwiThSoftDelay(APT\_Handle \*aptHandle,  
unsigned short calibrate)是空实现。

3.21.2 参数配置变更

APT模块3065H/3061H和3066M/3065P配置参数无变化。

3.22 CAN 驱动移植

3.22.1 接口移植

3066M/3065P系列和3065H/3061H系列的CAN模块驱动无差异。

表 3-24 3066M/3065P 系列和 3065H/3061H 系列 CAN 接口对比

功能	3066M/3065P CAN HAL接口	3065H/3061H CAN HAL接口	是否差异
初始化和去初始化配置	BASE_StatusType HAL_CAN_Init(CAN_Handle *canHandle)	BASE_StatusType HAL_CAN_Init(CAN_Handle *canHandle)	相同
	BASE_StatusType HAL_CAN_DeInit(CAN_Handle *canHandle)	BASE_StatusType HAL_CAN_DeInit(CAN_Handle *canHandle)	相同
CAN读写接口	BASE_StatusType HAL_CAN_ReadIT(CAN_Handle *canHandle, CANFrame *data, CAN_FilterConfigure *filterConfigure)	BASE_StatusType HAL_CAN_ReadIT(CAN_Handle *canHandle, CANFrame *data, CAN_FilterConfigure *filterConfigure)	相同
	BASE_StatusType HAL_CAN_Write(CAN_Handle *canHandle, CANFrame *data)	BASE_StatusType HAL_CAN_Write(CAN_Handle *canHandle, CANFrame *data)	相同
CAN获取状态	CAN_ErrorStatus HAL_CAN_GetErrorStatus(CAN_Handle *canHandle)	CAN_ErrorStatus HAL_CAN_GetErrorStatus(CAN_Handle *canHandle)	相同
	unsigned int HAL_CAN_GetErrorStatusCode(CAN_Handle *canHandle)	unsigned int HAL_CAN_GetErrorStatusCode(CAN_Handle *canHandle)	相同
	CAN_BusOffStatus HAL_CAN_GetBusOffStatus(CAN_Handle *canHandle)	CAN_BusOffStatus HAL_CAN_GetBusOffStatus(CAN_Handle *canHandle)	相同



功能	3066M/3065P CAN HAL接口	3065H/3061H CAN HAL接口	是否差异
	CAN_MessageReceiveStatus HAL_CAN_MessageReceiveStat us(CAN_Handle *canHandle)	CAN_MessageReceiveStatus HAL_CAN_MessageReceiveSt atus(CAN_Handle *canHandle)	相同
	CAN_MessageSendStatus HAL_CAN_MessageSendStatus( CAN_Handle *canHandle)	CAN_MessageSendStatus HAL_CAN_MessageSendStatu s(CAN_Handle *canHandle)	相同
中断处 理函数	void HAL_CAN_IrqHandler(void *handle)	void HAL_CAN_IrqHandler(void *handle)	相同
	BASE_StatusType HAL_CAN_RegisterCallBack(CA N_Handle *canHandle, CAN_CallBackFunType typeId, CAN_CallbackType pCallback)	BASE_StatusType HAL_CAN_RegisterCallBack(C AN_Handle *canHandle, CAN_CallBackFunType typeID, CAN_CallbackType pCallback)	相同

### 3.22.2 参数配置变更

CAN模块3065H/3061H和3066M/3065P配置参数无变化。

## 3.23 IOCMG 驱动移植

### 3.23.1 接口移植

表 3-25 3066M/3065P 系列和 3065H/3061H 系列 IOCMG 接口对比

功能	3066M/3065P TIMER HAL接口	3065H/3061H TIMER HAL接口	是否差异
IO配 置功 能整 体初 始化	IOCMG_Status HAL_IOCMG_Init(IOCMG_Hand le* handle)	IOCMG_Status HAL_IOCMG_Init(IOCMG_H andle* handle)	相同
IO端 口单 个功 能配 置	IOCMG_Status HAL_IOCMG_SetPinAltFuncMod e(unsigned int pinTypedef)	IOCMG_Status HAL_IOCMG_SetPinAltFunc Mode(unsigned int pinTypedef)	相同



功能	3066M/3065P TIMER HAL接口	3065H/3061H TIMER HAL接口	是否差异
	IOCMG_Status HAL_IOCMG_SetPinPullMode(unsigned int pinTypeDef, IOCMG_PullMode pullMode)	IOCMG_Status HAL_IOCMG_SetPinPullMode(unsigned int pinTypeDef, IOCMG_PullMode pullMode)	相同
	IOCMG_Status HAL_IOCMG_SetPinSchmidtMode(unsigned int pinTypeDef, IOCMG_SchmidtMode schmidtMode)	IOCMG_Status HAL_IOCMG_SetPinSchmidtMode(unsigned int pinTypeDef, IOCMG_SchmidtMode schmidtMode)	相同
	IOCMG_Status HAL_IOCMG_SetPinLevelShiftRate(unsigned int pinTypeDef, IOCMG_LevelShiftRate levelShiftRate)	IOCMG_Status HAL_IOCMG_SetPinLevelShiftRate(unsigned int pinTypeDef, IOCMG_LevelShiftRate levelShiftRate)	相同
	IOCMG_Status HAL_IOCMG_SetPinDriveRate(unsigned int pinTypeDef, IOCMG_DriveRate driveRate)	IOCMG_Status HAL_IOCMG_SetPinDriveRate(unsigned int pinTypeDef, IOCMG_DriveRate driveRate)	相同
	IOCMG_Status HAL_IOCMG_SetPinOpenDrainModeEx(unsigned int pinTypeDef, IOCMG_OpenDrainMode openDrainMode)	-	差异
获取IO端口单个功能配置	IOCMG_FuncMode HAL_IOCMG_GetPinAltFuncMode(unsigned int pinTypeDef)	IOCMG_FuncMode HAL_IOCMG_GetPinAltFuncMode(unsigned int pinTypeDef)	相同
	IOCMG_PullMode HAL_IOCMG_GetPinPullMode(unsigned int pinTypeDef)	IOCMG_PullMode HAL_IOCMG_GetPinPullMode(unsigned int pinTypeDef)	相同
	IOCMG_SchmidtMode HAL_IOCMG_GetPinSchmidtMode(unsigned int pinTypeDef)	IOCMG_SchmidtMode HAL_IOCMG_GetPinSchmidtMode(unsigned int pinTypeDef)	相同
	IOCMG_LevelShiftRate HAL_IOCMG_GetPinLevelShiftRate(unsigned int pinTypeDef)	IOCMG_LevelShiftRate HAL_IOCMG_GetPinLevelShiftRate(unsigned int pinTypeDef)	相同



功能	3066M/3065P TIMER HAL接口	3065H/3061H TIMER HAL接口	是否差异
	IOCMG_DriveRate HAL_IOCMG_GetPinDriveRate( unsigned int pinTypedef)	IOCMG_DriveRate HAL_IOCMG_GetPinDriveRate( unsigned int pinTypedef)	相同
	IOCMG_OpenDrainMode HAL_IOCMG_GetPinOpenDrain ModeEx(unsigned int pinTypedef)	-	差异
特殊 IO端 口功 能配 置和 获取	IOCMG_Status HAL_IOCMG_SetOscClkOutput Mode(bool mode)	IOCMG_Status HAL_IOCMG_SetOscClkOutputMode(bool mode)	相同
	IOCMG_Status HAL_IOCMG_SetOscClkFuncMode(bool mode)	IOCMG_Status HAL_IOCMG_SetOscClkFuncMode(bool mode)	相同
	IOCMG_Status HAL_IOCMG_SetOscClkDriveRate( IOCMG_OscClkDriveRate oscClkDriveRate)	IOCMG_Status HAL_IOCMG_SetOscClkDriveRate( IOCMG_OscClkDriveRate oscClkDriveRate)	相同
	bool HAL_IOCMG_GetOscClkOutput Mode(void)	bool HAL_IOCMG_GetOscClkOutputMode(void)	相同
	bool HAL_IOCMG_GetOscClkFuncMode(void)	bool HAL_IOCMG_GetOscClkFuncMode(void)	相同
	IOCMG_OscClkDriveRate HAL_IOCMG_GetOscClkDriveRate(void)	IOCMG_OscClkDriveRate HAL_IOCMG_GetOscClkDriveRate(void)	相同

说明

3066M/3065P和3065H/3061H对外HAL层接口一样。两者主要区别在于IO端口类型、数目不一样，可以在iomap.h文件中查看IO端口具体复用情况。  
另外，3066M/3065P新增了开漏输出功能。

3.23.2 参数配置变更

IOCMG模块相较于3065H/3061H，3066M/3065P配置参数新增开漏输出配置。

3.24 SMBUS 驱动移植



### 3.24.1 接口移植

表 3-26 3066M/3065P 系列和 3065H/3061H 系列 SMBus 接口对比

功能	3066M/3065P SMBus HAL接口	3065H/3061H SMBus HAL接口	是否差异
初始化和配置	BASE_StatusType HAL_SMBUS_Init(SMBUS_Handle *handle)	-	差异
	BASE_StatusType HAL_SMBUS_Deinit(SMBUS_Handle *handle)	-	差异
注册回调函数	BASE_StatusType HAL_SMBUS_RegisterCallback(SMBUS_Handle *handle, SMBUS_CallbackId callbackID, SMBUS_CallbackFunType pcallback)	-	差异
数据接收和发送	BASE_StatusType HAL_SMBUS_MasterReadBlocking(SMBUS_Handle *handle, unsigned short devAddr, SMBUS_DataBuffer buffer, unsigned int timeout, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_MasterWriteBlocking(SMBUS_Handle *handle, unsigned short devAddr, SMBUS_DataBuffer buffer, unsigned int timeout, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_SlaveReadBlocking(SMBUS_Handle *handle, SMBUS_DataBuffer buffer, unsigned int timeout, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_SlaveWriteBlocking(SMBUS_Handle *handle, SMBUS_DataBuffer buffer, unsigned int timeout, unsigned int frameOpt)	-	差异





功能	3066M/3065P SMBus HAL接口	3065H/3061H SMBus HAL接口	是否差异
	BASE_StatusType HAL_SMBUS_MasterReadIT(SMBUS_Handle *handle, unsigned short devAddr, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_MasterWriteIT(SMBUS_Handle *handle, unsigned short devAddr, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_SlaveReadIT(SMBUS_Handle *handle, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_SlaveWriteIT(SMBUS_Handle *handle, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_MasterReadDMA(SMBUS_Handle *handle, unsigned short devAddr, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_MasterWriteDMA(SMBUS_Handle *handle, unsigned short devAddr, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_SlaveReadDMA(SMBUS_Handle *handle, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异
	BASE_StatusType HAL_SMBUS_SlaveWriteDMA(SMBUS_Handle *handle, SMBUS_DataBuffer buffer, unsigned int frameOpt)	-	差异



功能	3066M/3065P SMBus HAL接口	3065H/3061H SMBus HAL接口	是否差异
中断与回调	void HAL_SMBUS_IrqHandler(void *handle)	-	差异
传输字节序	BASE_StatusType HAL_SMBUS_SetDataTransferSequenceEx(SMBUS_Handle *handle, SMBUS_DataTransferSequenceType sequence)	-	差异
SCL时钟延长	BASE_StatusType HAL_SMBUS_SetSclStretchModeEx(SMBUS_Handle *handle, SMBUS_ClockStretchType clkStretch)	-	差异
设置SCL低电平超时阈值	BASE_StatusType HAL_SMBUS_SetSclLowTimeoutEx(SMBUS_Handle *handle, unsigned int sclLowTimeout)	-	差异
设置总线空闲阈值	BASE_StatusType HAL_SMBUS_SetBusFreeTimeEx(SMBUS_Handle *handle, unsigned int busFreeTime)	-	差异
接收特殊地址和字节	BASE_StatusType HAL_SMBUS_Set10BitSlaveEnableEx(SMBUS_Handle *handle)	-	差异
	BASE_StatusType HAL_SMBUS_SetDeviceIdAddressEnableEx(SMBUS_Handle *handle)	-	差异
	BASE_StatusType HAL_SMBUS_SetStartByteEnableEx(SMBUS_Handle *handle)	-	差异
地址匹配掩码	BASE_StatusType HAL_SMBUS_SetOwnAddressMaskEx(SMBUS_Handle *handle, unsigned int addrMask)	-	差异
	BASE_StatusType HAL_SMBUS_SetOwnXmbAddressMaskEx(SMBUS_Handle *handle, unsigned int addrMask)	-	差异



### 3.24.2 参数配置变更

SMBus模块相较于3065H/3061H为3066M/3065P新增驱动模块，所有配置参数为新增参数。



# 4 参考资料

- 《3066M/3065P系列 数据手册》
- 《3066M/3065P系列 技术参考指南》
- 《SolarA<sup>2</sup> 驱动程序说明》



# A 缩略语

表 A-1 缩略语

缩略语	英文	中文
ACMP	Analog Comparator	模拟比较器
ADC	Analog Digital Converter	模数转换器
APT	Advanced PWM Timer	高级PWM定时器
CAMP	Capture Module	捕捉器
CMM	Clock Monitor Module	时钟频率监测
CRC	Cyclic Redundancy Check	循环冗余校验
CTS	Clear To Send	允许发送
DAC	Digital to Analog Conversion	数模转换器
DMA	Direct Memory Access	直接内存访问
FIFO	First In First Out	先进先出
FPU	Floating Point Unit	浮点处理单元
GPIO	General-purpose input/output	通用输入输出
GPT	General PWM Timer	通用PWM定时器
I2C	Inter-Integrated Circuit	集成电路接口
MCU	Microcontroller Unit	微控制器单元
PFC	Power Factor Correction	功率因数校正
PGA	Programmable Gain Amplifier	可编程增益放大器
PLL	Phase Locked Loop	锁相环
PPU	Position Process Unit	位置处理单元
PTU	Period Trigger Unit	周期触发单元



缩略语	英文	中文
PWM	Pulse Width Modulation	脉冲宽度调制
SCL	Serial Clock Line	串行时钟线
SPI	Serial Peripheral Interface	串行外设接口
SRAM	Static Random Access Memory	静态随机存取存储器
TSU	Time Stamp Unit	时间戳单元
UART	Universal Asynchronous Receiver Transmitter	通用异步收发传输器
WDG	Watch Dog	看门狗
SMBUS	System Management Bus	系统管理总线。