

简介

PY32MD540 系列芯片是普冉推出的高集成度、高性价比电机控制 SOC。采用先进的 55nm 工艺，其内部集成了高性能的 32 位 ARM® Cortex®-M0+内核、宽电压工作范围的 MCU。同时合封了先进预驱和低压降 LDO，是一颗专门用于驱动 6 个外部 N 沟道 Mosfet 的电机控制芯片。

PY32MD540 内部的 MCU 是一颗先进的 Arm Cortex M0+核心，包含专为电机设计的 PWM 单元：易于实现三相变频控制，可轻松实现三相同步 SVPWM、移相 SVPWM；内部有转换速度速率高达 2Mbps 的 12bit ADC，可以满足高转速的同步电机控制；同时自带与 PWM/ADC 单元联动的比较器单元，除了可以满足高速无感方波控制的需求，还可以在极短时间可靠的完成电机的过电流、过电压保护等；丰富的通信单元，可以支持从消费类、家电到工业和汽车控制的多种多样的通信需求。

PY32MD540 可以运行在 88Mhz 的高主频，带有大容量的 Flash 和 RAM，这使得芯片可以支持从单电阻 FOC 控制、无感方波控制、双电阻 FOC 控制等广泛的电机拓扑。

以上这些优秀特性的整合，使得 PY32MD540 成为一颗使用范围广泛、高度集成、性价比极高的电机控制 SOC，为各个应用的开发带来显著的性能提升以及 BOM 节约。

关键特性

- ARM Cortex-M0+ CPU @88MHz
- 集成 3 相 6N 驱动，耐压 600V
- 推荐工作电压范围 AC 110~220V
- 集成 5V LDO
- 最大 64KB flash, 8KB RAM
- 12-bit ADC, 最高采样速度 2Mbps
- 2 x 增益可配置的差分
- 2x 快速比较器.
- 支持 UART、LPUART、LIN、SPI、I2C 等通信总线
- 片上自带温度传感器
- LQFP48 封装
- -40 to +125°C 工作结温

本应用笔记将帮助用户了解 PY32MD540 的基本结构和关键特性，并快速着手应用。

表 1. 适用产品

类型	产品系列
微型控制器系列	PY32MD540

PUYA CONFIDENTIAL

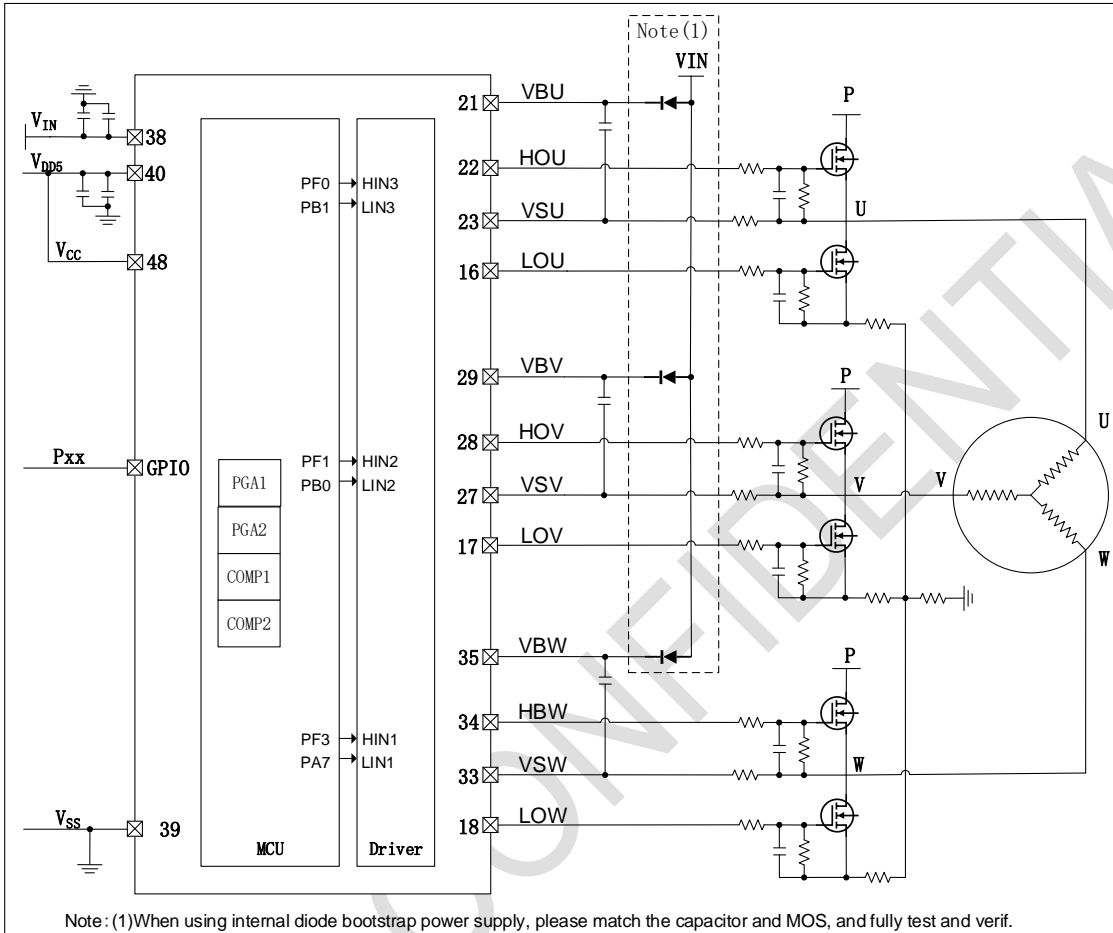
目录

1	芯片构架和 MCU 部分应用指南	4
1.1	芯片构架	4
2	芯片配置要点	4
2.1	时钟配置	4
2.1	引脚配置	7
3	版本历史	9

1 芯片构架和 MCU 部分应用指南

1.1 芯片构架

PY32MD540 采用 PY32F032 和 6N 预驱芯片合封的形式，内部连接示意图如下：



其中关于 MCU 部分的参考手册为：《PY32F032 Reference Manual》

数据手册为《PY32F032 Datasheet》

应用指南请参考《PY32F032 Series Application Notes》：

2 芯片配置要点

2.1 时钟配置

- PY32MD550 内部 MCU 最快工作在 88M，需要特殊配置

代码示例

```
/******
```

Description: 系统时钟配置函数

```

Input:
Output:
Return:
*****/

void systemClock_Init(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /* 配置时钟源 HSE/HSI/LSE/LSI */

    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI |
RCC_OSCILLATORTYPE_LSI;

    RCC_OscInitStruct.HSIState = RCC_HSI_ON; /* 开启 HSI */

    RCC_OscInitStruct.HSIDiv = RCC_HSI_DIV1; /* 不分频 */

    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_8MHz; /* 配置 HSI 输出时
钟为 8MHz */

    RCC_OscInitStruct.HSEState = RCC_HSE_OFF; /* 关闭 HSE */

    //RCC_OscInitStruct.HSEFreq = RCC_HSE_24_32MHz; /* HSE 工作频率范
围 24M~32M */

    RCC_OscInitStruct.LSIState = RCC_LSI_OFF; /* 关闭 LSI */

    RCC_OscInitStruct.LSEState = RCC_LSE_OFF; /* 关闭 LSE */

    //RCC_OscInitStruct.LSEDriver = RCC_ECSCR_LSE_DRIVER_1; /* LSE 默认驱动能
力 */

```

```

RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;           /* 开启 PLL */

RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;   /* PLL 的时钟源 */

RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL11;         /* PLL 11 倍频输出 */
*/
RCC_OscInitStruct.PLL.PLLDiv = RCC_PLL_DIV1;

/* 初始化 RCC 振荡器 */

if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    APP_ErrorHandler();
}

/*初始化 CPU,AHB,APB 总线时钟*/

RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYSCLK |
RCC_CLOCKTYPE_PCLK; /* RCC 系统时钟类型 */

RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
/* 配置 PLL 为系统时钟 */

RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
/* APH 时钟不分频 */

RCC_ClkInitStruct.APBCLKDivider = RCC_HCLK_DIV1;
/* APB 时钟不分频 */

/* 初始化 RCC 系统时钟 */

//if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_3) != HAL_OK)
{
    APP_ErrorHandler();
}
}

```

2.1 引脚配置

- 根据 MCU TIM1 和预驱芯片的对应关系，配置 PWM

代码示例

```

#define GPIO_TIM1CH1H_PORT      GPIOF
#define GPIO_TIM1CH1H_PIN      GPIO_PIN_3
#define GPIO_TIM1CH1H_AF      GPIO_AF11_TIM1

#define GPIO_TIM1CH1L_PORT      GPIOA
#define GPIO_TIM1CH1L_PIN      GPIO_PIN_7
#define GPIO_TIM1CH1L_AF      GPIO_AF2_TIM1

#define GPIO_TIM1CH2H_PORT      GPIOF
#define GPIO_TIM1CH2H_PIN      GPIO_PIN_1
#define GPIO_TIM1CH2H_AF      GPIO_AF11_TIM1

#define GPIO_TIM1CH2L_PORT      GPIOB
#define GPIO_TIM1CH2L_PIN      GPIO_PIN_0
#define GPIO_TIM1CH2L_AF      GPIO_AF2_TIM1

#define GPIO_TIM1CH3H_PORT      GPIOF
#define GPIO_TIM1CH3H_PIN      GPIO_PIN_0
#define GPIO_TIM1CH3H_AF      GPIO_AF11_TIM1

/**
 * @brief PWM GPIO 初始化函数
 * @details 配置 TIM1 的三相 PWM 输出引脚为复用推挽模式，关联到 TIM1 的外设功能
 * @note PWM 引脚需要配置为复用模式，才能输出 TIM1 的 PWM 信号，用于电机驱动的上下桥臂
 */
void GpioPwmInit(void)
{
    GPIO_InitTypeDef GPIO_InitStructure; // GPIO 初始化结构体

    // 配置 GPIO 通用参数：复用推挽、下拉电阻、高速模式
    GPIO_InitStructure.Mode = GPIO_MODE_AF_PP; // 模式：复用功能推挽输出
    (Alternate Function Push-Pull)
    GPIO_InitStructure.Pull = GPIO_PULLDOWN; // 上下拉：下拉电阻（防止引脚悬空）
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_HIGH; // 速度：高速

    // 配置 TIM1_CHANNEL_1 高侧引脚（上桥臂）
    GPIO_InitStructure.Pin = GPIO_TIM1CH1H_PIN;

```

```
GPIO_InitStruct.Alternate =GPIO_TIM1CH1H_AF;// GPIO_AF2_TIM1; // 复用
功能: GPIO_AF2_TIM1 (TIM1 的复用功能映射)
HAL_GPIO_Init(GPIO_TIM1CH1H_PORT, &GPIO_InitStruct);

// 配置 TIM1_CHANNEL_1 低侧引脚 (下桥臂, 互补输出)
GPIO_InitStruct.Pin = GPIO_TIM1CH1L_PIN;
GPIO_InitStruct.Alternate =GPIO_TIM1CH1L_AF;// GPIO_AF2_TIM1;
HAL_GPIO_Init(GPIO_TIM1CH1L_PORT, &GPIO_InitStruct);

// 配置 TIM1_CHANNEL_2 高侧引脚
GPIO_InitStruct.Pin = GPIO_TIM1CH2H_PIN;
GPIO_InitStruct.Alternate =GPIO_TIM1CH2H_AF;// GPIO_AF2_TIM1;
HAL_GPIO_Init(GPIO_TIM1CH2H_PORT, &GPIO_InitStruct);

// 配置 TIM1_CHANNEL_2 低侧引脚
GPIO_InitStruct.Pin = GPIO_TIM1CH2L_PIN;
GPIO_InitStruct.Alternate =GPIO_TIM1CH2L_AF;// GPIO_AF2_TIM1;
HAL_GPIO_Init(GPIO_TIM1CH2L_PORT, &GPIO_InitStruct);

// 配置 TIM1_CHANNEL_3 高侧引脚
GPIO_InitStruct.Pin = GPIO_TIM1CH3H_PIN;
GPIO_InitStruct.Alternate =GPIO_TIM1CH3H_AF;// GPIO_AF2_TIM1;
HAL_GPIO_Init(GPIO_TIM1CH3H_PORT, &GPIO_InitStruct);

// 配置 TIM1_CHANNEL_3 低侧引脚
GPIO_InitStruct.Pin = GPIO_TIM1CH3L_PIN;
GPIO_InitStruct.Alternate =GPIO_TIM1CH3L_AF;// GPIO_AF2_TIM1;
HAL_GPIO_Init(GPIO_TIM1CH3L_PORT, &GPIO_InitStruct);
}
```

3 版本历史

版本	日期	更新记录
V1.0	2026.05.08	初版



Puya Semiconductor Co., Ltd.

声 明

普冉半导体(上海)股份有限公司 (以下简称:“Puya”)保留更改、纠正、增强、修改 Puya 产品和/或本文档的权利,恕不另行通知。用户可在下单前获取产品的最新相关信息。

Puya 产品是依据订单时的销售条款和条件进行销售的。

用户对 Puya 产品的选择和使用承担全责,同时若用于其自己或指定第三方产品上的,Puya 不提供服务支持且不对此类产品承担任何责任。

Puya 在此不授予任何知识产权的明示或暗示方式许可。

Puya 产品的转售,若其条款与此处规定不一致,Puya 对此类产品的任何保修承诺无效。

任何带有 Puya 或 Puya 标识的图形或字样是普冉的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代并替换先前版本中的信息。

普冉半导体(上海)股份有限公司 - 保留所有权利