

## 目 录

第 1 章 ServoMotor-DK506 实验指导 .....	1
1.1 概述 .....	1
1.2 开环转动实验 .....	1
1.2.1 实验原理 .....	1
1.2.2 实验步骤 .....	1
1.3 速度闭环实验 .....	2
1.3.1 实验原理 .....	2
1.3.2 实验步骤 .....	3
1.4 位置闭环实验 .....	3
1.4.1 实验原理 .....	3
1.4.2 实验步骤 .....	4
1.5 电机控制实验程序解析 .....	4

# 第1章 ServoMotor-DK506 实验指导

## 1.1 概述

ServoMotor-DK506 是广州致远电子股份有限公司研发的一款伺服电机学习板，采用富士通公司的 Cortex-M3 系列单片机 MB9BF506R 和 IR 公司的电机驱动模块 IRAM136-1561，可实现伺服电机的速度控制和位置控制。



图 1.1 ServoMotor-DK506 实物图

## 1.2 开环转动实验

### 1.2.1 实验原理

开环转动框图如图 1.2 所示，角度发生器不断产生 0-360° 的角度给 IPark 变换，进一步产生 SVPWM 驱动伺服电机不断转动。

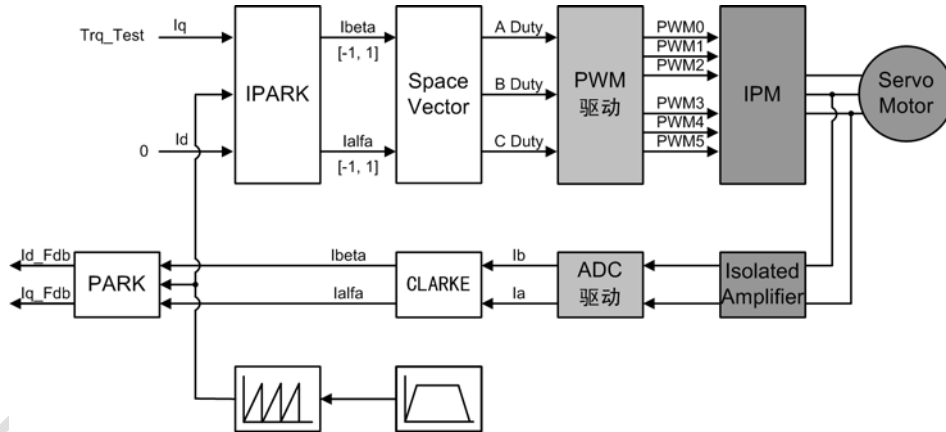


图 1.2 开环转动框图

### 1.2.2 实验步骤

- 1) 断开 JP10 中的 4、5 和 6、7 脚，电机驱动线接口 J8 不接电机，通过 J2 给板上电。
- 2) 用 IAR6.2 打开光盘目录【单片机程序(ServoMotor)】下的 Project 工程，并重新编译。
- 3) 用 AK100 仿真器下载程序到 ServoMotor-DK506 的 MCU 中。
- 4) 板子断电，待板子上的 LED Power 熄灭之后，把电机驱动线接到 J8。
- 5) 重新给板上电，LED1 亮、LED2 灭，电机开始转动。
- 6) 把 JP11 的 1~4 脚分别接入如图 1.3 所示的低通电路。

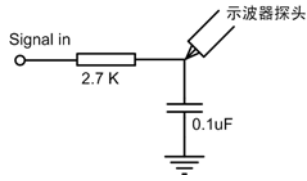


图 1.3 低通电路及示波器测试图

7) 按图 1.3 所示位置接入示波器，分别观察 JP11 的 1~4 脚的输出波形。输出的波形如图 1.4 所示，且  $I_{\beta}$  和  $I_{\alpha}$  相差  $180^\circ$  的相位。

注：JP11 的 1~4 脚分别对应 A\_Duty、B\_Duty、 $I_{\beta}$  和  $I_{\alpha}$ 。

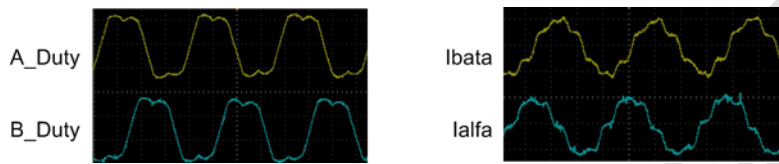


图 1.4 用示波器观察到的 PWM 和电流波形图

### 1.3 速度闭环实验

#### 1.3.1 实验原理

速度闭环框图如图 1.5 所示，相对于开环实验，Park 变换和 IPark 变换的输入角度由编码器通过计算得到。并增加了  $I_d$ 、 $I_q$  和速度 PID 闭环控制。

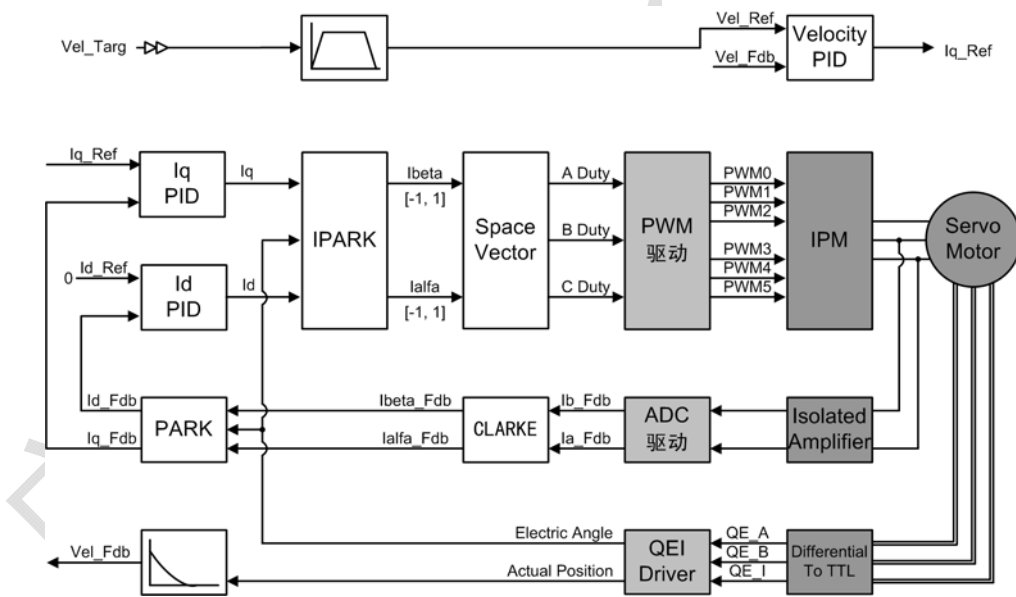


图 1.5 速度闭环框图

使用编码器计算角度有一个前提，就是找到编码器的 0 度参考点。下面简要介绍找到编码器 0 度参考点的方法：如图 1.6 所示，把  $I_q$  电流置 0， $I_d$  电流等于一个恒定值，且输入角度给定 0，一段时间后，电机转子将转到编码器的 0 度参考点位置，此时将编码器清零，即找到编码器的 0 度参考点。为了可靠把电机转子拖动到编码器的 0 度参考点位置，需先将电机转子拖动到编码器的 90 度参考点位置。

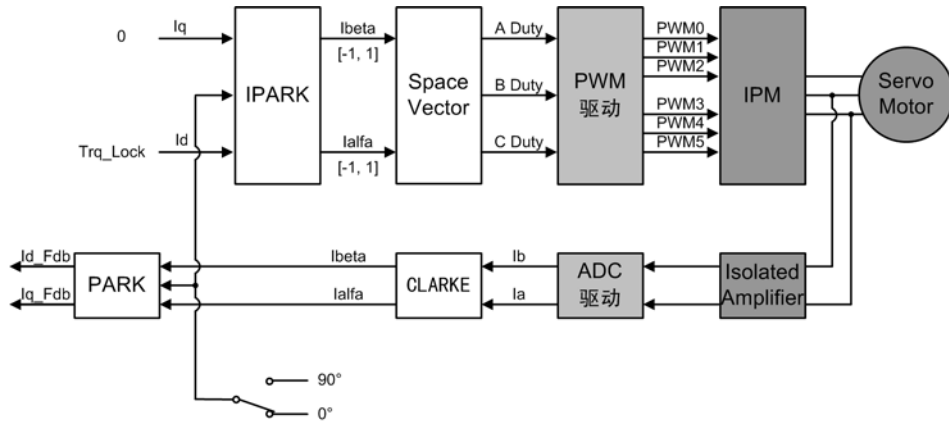


图 1.6 锁定“零点”框图

### 1.3.2 实验步骤

1) 断开 JP10 中的 4、5 脚，短接 JP10 中的 6、7 脚，电机驱动线接口 J8 不接电机，通过 J2 给板上电。

2) 用 IAR6.2 打开光盘目录【单片机程序\ServoMotor】下的 Project 工程，并重新编译。

3) 用 AK100 仿真器下载程序到 ServoMotor-DK506 的 MCU 中。

4) 板子断电，待板子上的 LED Power 熄灭之后，把电机驱动线接到 J8。

5) 重新给板子上电，LED1 灭、LED2 亮，电机开始转动。

6) 按下 KEY1 键，电机转度增加，按下 KEY2 键，电机转速减小。

## 1.4 位置闭环实验

### 1.4.1 实验原理

位置闭环框图如图 1.7 所示，相对于速度闭环实验，增加了位置 P 闭环控制。

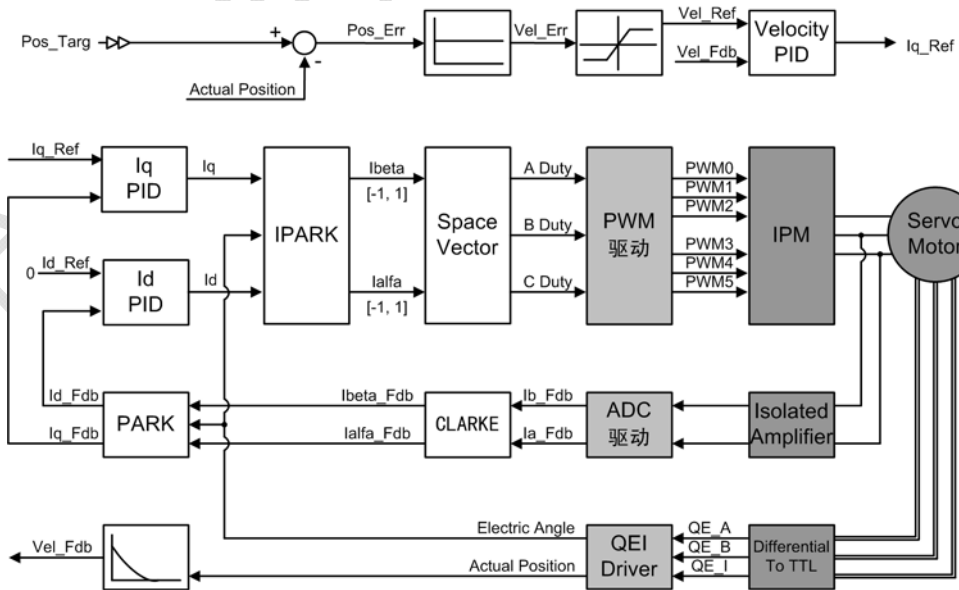


图 1.7 位置闭环框图

## 1.4.2 实验步骤

- 1) 短接 JP10 中的 4、5 脚，断开 JP10 中的 6、7 脚，电机驱动线接口 J8 不接电机，通过 J2 给板子上电。
- 2) 用 IAR6.2 打开光盘目录【单片机程序\ServoMotor】下的 Project 工程，并重新编译。
- 3) 用 AK100 仿真器下载程序到 ServoMotor-DK506 的 MCU 中。
- 4) 板子断电，待板子上的 LED Power 熄灭之后，把电机驱动线接到 J8。
- 5) 重新给板子上电，LED1、LED2 亮，电机开始转动。
- 6) 快速按下 KEY1 键，电机正转 90 度；快速按下 KEY2 键，电机反转 90 度。
- 7) 长按 KEY1 键，电机正转，直到按键松开；长按 KEY2 键，电机反转，直到按键松开。

## 1.5 电机控制实验程序解析

ServoMotor-DK506 的实验都是基于 ZMC (ZLG Motion Control) 库和 ZMD (ZLG MCU Driver) 库的，关于 ZMC 的说明详见《ZLG Motor Control 手册》，ZMD 的说明详见《ZLG MCU Driver 使用手册》。

实验程序主要调用了 ZMC 中 App 层的三类函数：电机初始化函数 `zmcPmsmQeiInit()`、电机运行控制函数 `zmcPmsmQeiRun()` 和 PWM 中断回调函数 `zmcPmsmQeiCallbackOpen()` (或 `zmcPmsmQeiCallbackVel()`、`zmcPmsmQeiCallbackPos()`)，如程序清单 1.1 所示。

程序清单 1.1 ZMC 中 App 层三类函数

```
int main (void)
{
    .....
    zmcPmsmQeiInit();                /* 电机初始化 */
    .....
    for (;;) {
        zmcPmsmQeiRun();            /* 电机运行控制，接收外部命令 */
    }
}

void zmcPmsmQeiCallbackOpen (void) /* 开环控制 PWM 中断回调函数 */
{
    .....
}

void zmcPmsmQeiCallbackVel (void) /* 速度控制 PWM 中断回调函数 */
{
    .....
}

void zmcPmsmQeiCallbackPos (void) /* 位置控制 PWM 中断回调函数 */
{
    .....
}
```

在系统上电时，程序根据 JP10 中 4、5 和 6、7 脚的跳线设置来选择使用不同的 PWM 中断回调函数，如程序清单 1.2 所示。

## 程序清单 1.2 选择不同的 PWM 中断回调函数

```
void zmcPmsmQeiInit (void)
{
    .....

    if (JP10 6、7 脚短接) {
        zmcFocPwmInit(10000,          /* PWM 频率 10KHz          */
                     3000,           /* 死区时间 3000ns        */
                     (void *)0,      /* PWM 错误中断不使用     */
                     &zmcPmsmQeiCallbackVel); /* 使用速度控制回调函数 */
    } else if (JP10 4、5 脚短接) {
        zmcFocPwmInit(10000,          /* PWM 频率 10KHz          */
                     3000,           /* 死区时间 3000ns        */
                     (void *)0,      /* PWM 错误中断不使用     */
                     &zmcPmsmQeiCallbackPos); /* 使用位置控制回调函数 */
    } else {
        zmcFocPwmInit(10000,          /* PWM 频率 10KHz          */
                     3000,           /* 死区时间 3000ns        */
                     (void *)0,      /* PWM 错误中断不使用     */
                     &zmcPmsmQeiCallbackOpen); /* 使用开环控制回调函数 */
    }
    .....
}
```