# 软件详细设计

SWDD -NXP电机驱动方案

# 1 总览

## 文档作用

本文档属于软件详细设计文档，用于软件开发人员开发测试，便于软件团队内部交流学习。

不对外公开。

## 简写

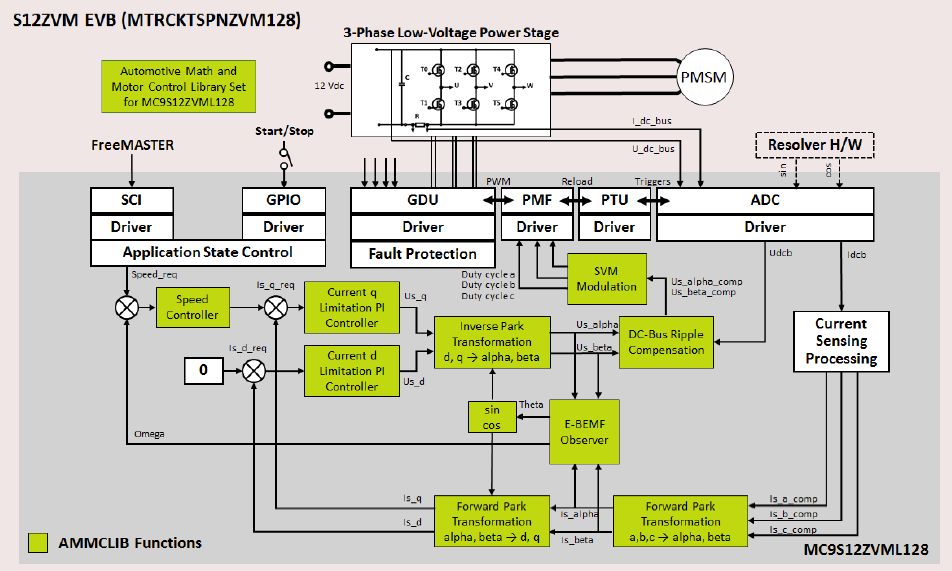
## 标准

标准详见 冷却风扇单元代码编写规范\_v1.0

## 历史记录

版本：v1.0  
发布时间：2023/07/14  
更改内容：初版

# 2 整体系统



# 3 模块设计（软件硬件）

**芯片型号：MC9S12ZVML128MKH**

**参考软件版本： S12ZVMxHCEVB\_SW\_lin0**

## 3.1 硬件方案

### 3.1.1 芯片资源

图示, 示意图

描述已自动生成

### 3.1.2 驱动方案

图示, 示意图

描述已自动生成

NXP方案采用单电阻采样，所以只需要采集母线电流，在软件中进行电流重构，分解成三相电流IU IV IW。

### 3.1.3 电流采样电路

R600 采集母线电流, 经过RC滤波处理

图示, 示意图

描述已自动生成

接入单片机内部运放（AMP），进行放大输出至AMP\_OUT，采集获得电流值。

图标

描述已自动生成

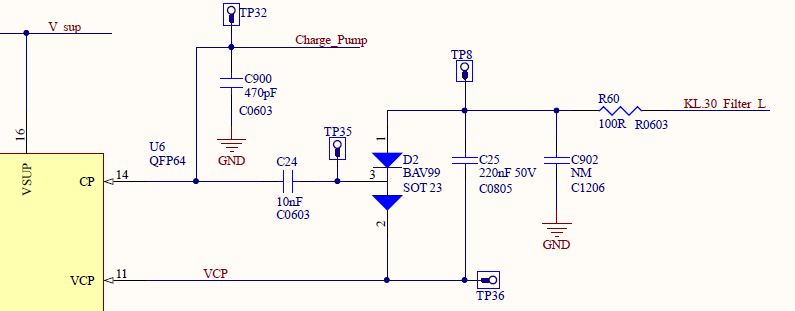
运放输出电压可以参考以下公式（R43不焊）：

[（Vdd-Vsen）/(R40+R41)+ Vsen/R45]\*(R45+R47)=Vout

其中：

* Vdd ：传感器供电电压
* Vsen ：采样电阻端电压
* Vout ：运放输出电压

### 3.1.4 防反接电路



通过上图的电荷泵电路获得高于供电电平KL30\_Filter\_L的输出电压VCP,用以开启防反

NMOS管Q101 。

图示, 示意图

描述已自动生成

* B\_F+网点：外部12V供电输入正极。
* KL.30\_Filter\_L ：经过防反MOS管后的供电正极。
* VCP ：电荷泵输出，接入防反NMOS门极。

### 3.1.5 高侧NMOS驱动自举供电

日程表

中度可信度描述已自动生成

与英飞凌方案完全依靠电荷泵电路输出的VCP作为高侧驱动供电不同，恩智浦每相半桥均使用自举电路来给高侧NMOS驱动供电。

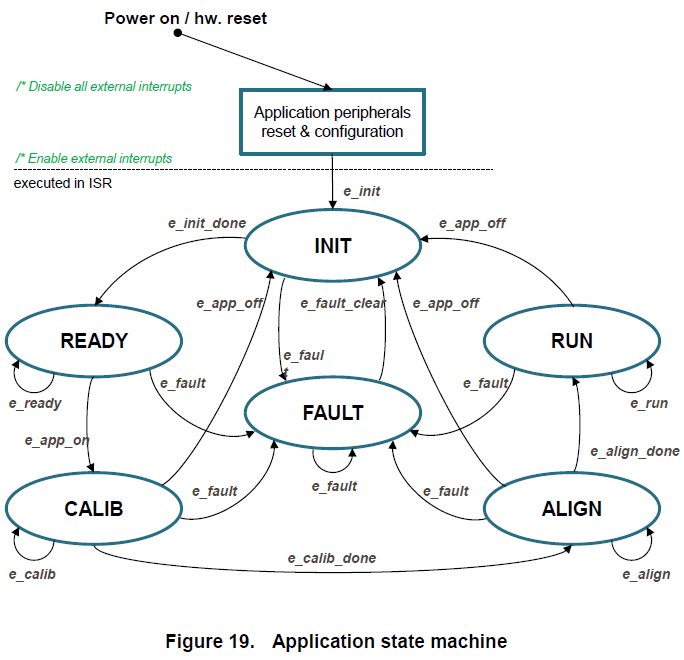
## 3.2软件方案

### 3.2.1 软件主体架构

图示

描述已自动生成

### 3.2.2 运行状态机流转结构



### 3.2.3 电流采样及重构计算

电流offset电压采集通过函数“Meas\_CalibCurrentSense()”获得，在CALIB状态时调用，此后运行时每次电流采集ADC值需减去这个offset值获得实际电流对应值。

运行状态时每个周期会采集两相的瞬时电流值，第三相电流值根据基尔霍夫定律计算得出。相关计算过程请参考函数“Meas\_Get3PhCurrent（）”。

* 需要特别注意的是，为了避免出现某相电流维持时间过短无法实现准确采集的问题，本方案采取了被称为“双开关(double switching)”的PWM等效变换方式,相关描述参考下图：

**图表, 图示, 箱线图

描述已自动生成**

这种调制方式的优点：

* 不影响控制精度的前提下，可以精确地获得极短等效时间内的电流值。

这种调制方式的缺点：

* 如上所述，引入了第三电压矢量，引起谐波失真。增加开关损耗和死区时间，

必然会引起电压利用率下降。

### 3.2.4 Clark 变换

Clark变换函数 由NXP官方提供

GMCLIB\_Clark(&drvFOC.iAlBeFbck,&drvFOC.iAbcFbck)

* drvFOC.iAbcFbck ：三相电流值
* drvFOC.iAlBeFbck ：正交坐标系α和β轴电流值

本函数用前述采集重构算法获得的三相电流值作为输入，输出正交坐标系电流值。

### 3.2.5 Park变换

Park变换函数 由NXP官方提供

GMCLIB\_Park(&drvFOC.iDQFbck,&drvFOC.thTransform,&drvFOC.iAlBeFbck)

* drvFOC.iAlBeFbck： 正交坐标系电流值
* drvFOC.thTransform: 转子位置（角度）
* drvFOC.iDQFbck： Q向与D向电流值

本函数用输入的正交坐标系电流和转子位置角，换算出目前位置对应的Iq和Id电流值。

### 3.2.6 SIN COS 计算

NXP官方以级数近似计算法获得SIN 和COS 计算值：

#define GFLIB\_Sin(...) macro\_dispatcher(GFLIB\_Sin, \_\_VA\_ARGS\_\_)(\_\_VA\_ARGS\_\_)

/\* This function implements polynomial approximation of sine function. \*/

#define GFLIB\_Cos(...) macro\_dispatcher(GFLIB\_Cos, \_\_VA\_ARGS\_\_)(\_\_VA\_ARGS\_\_)

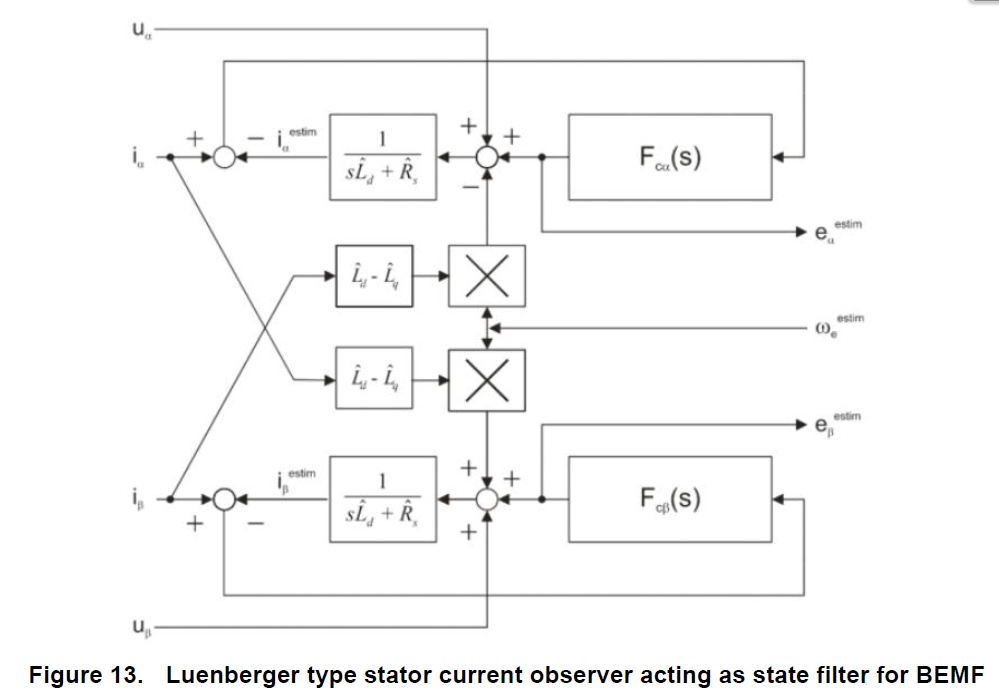
/\* This function implements polynomial approximation of cosine function. \*/

根据目前知识，可能的计算公式：

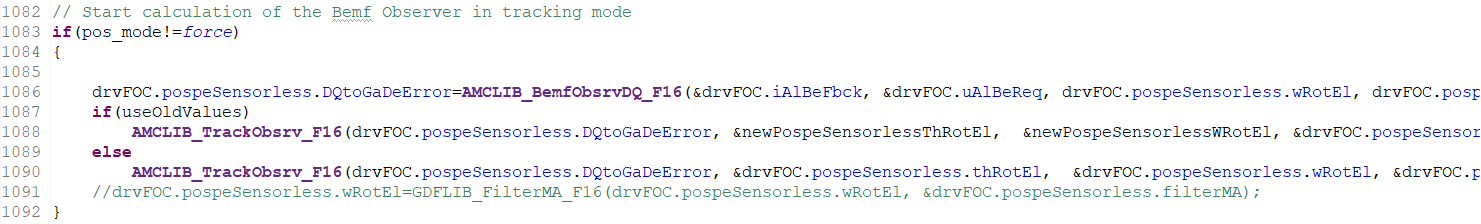
图表, 箱线图

描述已自动生成

### 3.2.7 反电动势观测器和拖动观测器



相关描述参考资料AN5327的3.4节。代码体现如下：

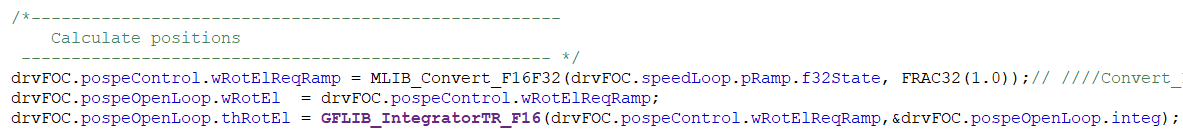


### 3.2.8速度和位置计算

图示

描述已自动生成

代码体现：



### 3.2.9 扇区计算

扇区计算函数， 由NXP官方提供：

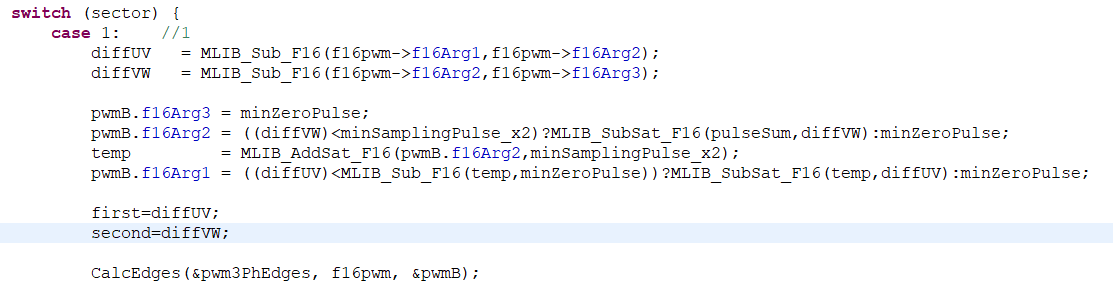
drvFOC.svmSector = GMCLIB\_SvmStd(&(drvFOC.pwm16),&(drvFOC.uAlBeReq))

* drvFOC.pwm16： 占空比配置值，输入计算后实时刷新。
* drvFOC.uAlBeReq: 正交轴电流值α和β输入。
* drvFOC.svmSector： 扇区数值。

### 3.2.10 T1 T2计算

运行占空时间值计算在函数 SetDutycycle(SWLIBS\_3Syst\_F16 \*f16pwm, tU16 sector)

中进行，代码体现如下：



其中

* first： T1时间值
* second：T2时间值
* sector： 所在控制扇区

### 3.2.11 母线电流有效值计算

目前软件中，电流母线计算值在函数Meas\_Get3PhCurrent（）中进行，代码体现如下：

文本, 应用程序

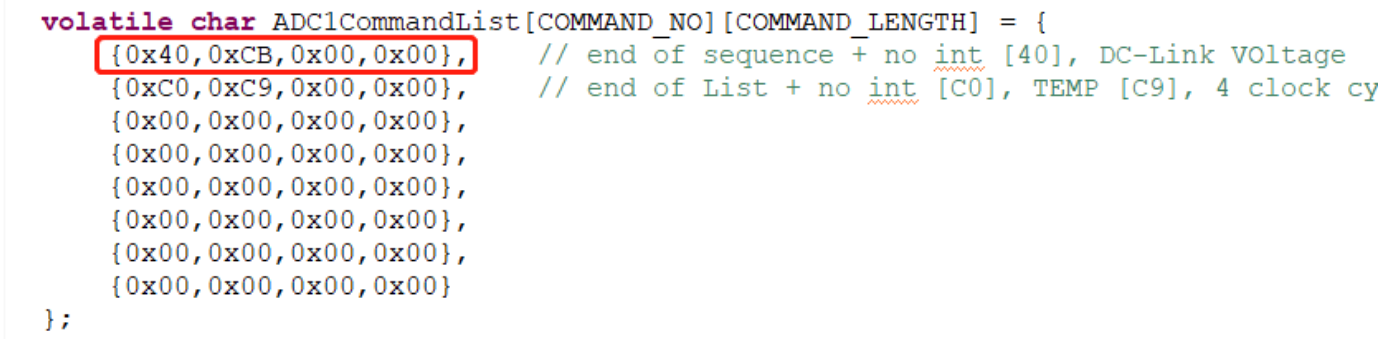
描述已自动生成

其中：

* Ifirst： T1时间采集电流值
* Isecond：T2时间采集电流值
* Unitdev: 电流标定值，各PCBA不同。
* rmsadd：有效电流累加值（待平均）
* Iadd： 有效电流值样本数
* rms： 有效电流值（平均后）

### 3.2.12 母线电压采集值配置计算

目前软件中，AD数据采集均是通过寄存器配置自动采集工作流，完成相关AD值的依次采集，通过DMA将采集值直接放入数据池，母线电压采集是一个比较典型的例子：



图示四个数值分别对应寄存器ADCCMD\_0、ADCCMD\_1、ADCCMD\_2、ADCCMD\_3的配置值，意义描述如下：

* ADCCMD\_0-0x40:采集串结束后重新开始，无中断标志；
* ADCCMD\_1-0xCB: VRH\_1和VRL\_1作为参考电压，选择内部通道3；
* ADCCMD\_2-0x00:采样时间为4个ADC时钟。
* ADCCMD\_3-0x00:功能保留。

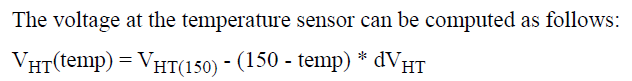
采集值与实际电压换算公式

Vdc=(((Vad>>4)/4096)\*Vref)\* Adiv

* Vdc：母线电压
* Vad：AD采集值
* 常数4：采集值左对齐位移位数
* 常数4096：AD采集满量程值
* Vref：AD采集参考电压，本例典型值为5V
* Adiv: 实际电压与采样电压分压比，本例典型值为5倍。

### 3.2.13 片内温度采集

片内温度采集参考公式：



* Vht(temp):目前温度（temp）下的采集电压值；
* Vht(150):150℃时的采样电压值，典型值2.4V;
* dVht：电压对应温度变化比率，本例为5.25 mV/℃；

### 3.2.14 NTC温度采集

根据对温度数值和AD采集值对应关系的分析比对，认为其关系对应一个以顶点正交轴

半边翻折的抛物线。如下图：

图表, 折线图

描述已自动生成

相关代码实现：

图形用户界面, 文本, 应用程序

描述已自动生成

* NTC\_table[3]:计算三项参数-中间点AD值、中间点温度值、全量程温度系数
* Delt:初始值为测量点AD值与中间点AD值的差值，计算后为测量点温度与中间点温度

差值。

* MotorDrive\_uiTemperatureNTC\_Digital：实际温度值（保留一位小数）。

### 3.2.15 转速单位转换

程序中，数值转速以16位有符号数，即-32768到32768的数值范围，来表达-MAX到+MAX的全转速范围。换算表达式为：

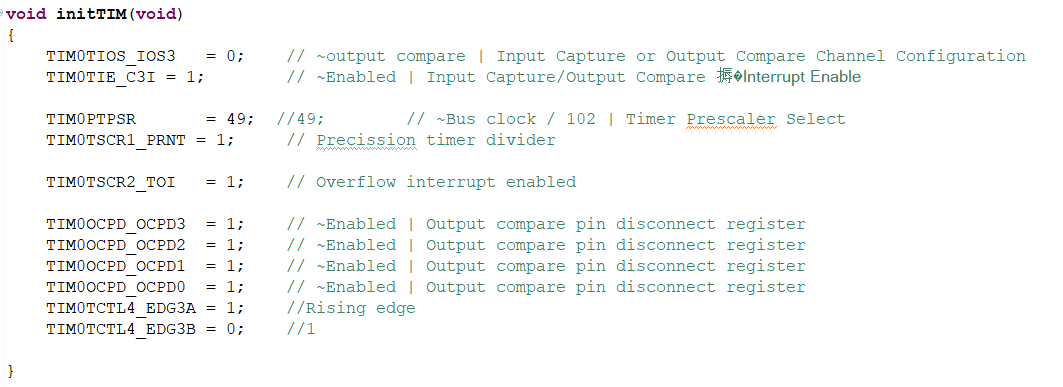
Dwth= FRAC16 (Awth/Amax)

* Dwth: 数值转速，无量纲，用以软件计算输入。
* Awth：实际转速，单位 Rpm，用以表达实际转速。
* FRAC16：浮点数16位整型化计算公式，处理算法如下：



### 3.2.16 输入PWM信号控制

定时器初始化配置代码：



周期及占空比捕捉代码（中断函数TIM0chan3\_ISR）：

图形用户界面, 文本, 应用程序

描述已自动生成

其中

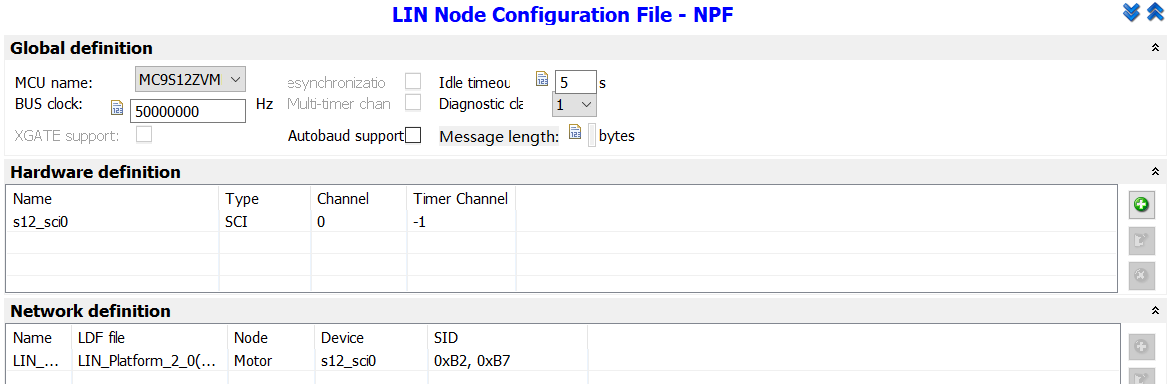
* PWMControlUpdate(TIM0TC3, &pwmControlData)：用以捕捉信号周期及占空比。
* PWMControlGetOutputValue(&pwmControlData)： 用以将有效占空比值换算为转速
* pwmControlData.outputValue: 转速等级数值

### 3.2.17 Lin通信相关配置

LIN通信借助NXP官方提供的NCF TOOL来进行配置，需要按标准输入LDF文件（数据结构和定义）和NPF文件（通信节点配置）来设置生成LIN配置文件。

表格

描述已自动生成



* 要点1：BUS clock 需要根据主频配置准确核算。
* 要点2: 移植LIN协议栈有可能出现文件结构冲突，需核对修改。

### 3.2.18 prm文件配置

可以按需要更改FLASH、EEPROM、RAM等存储资源的分配：

应用程序

低可信度描述已自动生成

* prm文件配置是进行Bootloader功能开发的基础。

### 3.2.19 Bootloader功能框架

目前bootloader功能拟按照AN4723文档中的结构进行软件开发：

