MCFLIB

应用笔记

Rev 1.0　2022/04/29

MCFLIB 用户指南

简介

该文档用于描述基于灵动微电子的 Arm® Cortex®-M0 内核的 MCU 电机控制数学库(MCFLIB)。

目录

1 引言

1.1 概述

该用户指南描述了基于灵动微电子的 Arm® Cortex®-M0 内核 MCU 电机控制数学库(MCFLIB)。该库包含了取 Park、Invpark、Clark、Invclark 等函数。

* 注意:所有例程的测试时间均基于 72MHZ 主频的 MCU 且在 RAM 中执行所得。

1.2 数据类型

MCFLIB 支持的数据类型:(无符号)整型。整数数据类型对于通用计算非常有用。以下列表为库中定义的整数类型:

•无符号 16 位整数—< 0；65535 >最小分辨率 1

•有符号 16 位整数—<-32768；32767 >最小分辨率 1

•无符号 32 位整数—< 0；4294967295 >最小分辨率为 1

•有符号 32 位整数—<-2147483648；2147483647 >最低分辨率为 1

以下列表为库中定义的小数类型:

•定点 16 位小数—<-1；1 - 2^{-15} >最小分辨率为 2^{-15}

•定点 32 位小数—<-1；1 - 2^{-31} >最小分辨率为 2^{-31}

1.3 例程定义

为了实现函数的简单调用，函数名使用前缀和后缀来区分函数版本，请参考以下示例:

```
s16 _Result = MLIB_Mul_Q15(s16Mult1, s16Mult2);
```

其中函数名由三部分组成:

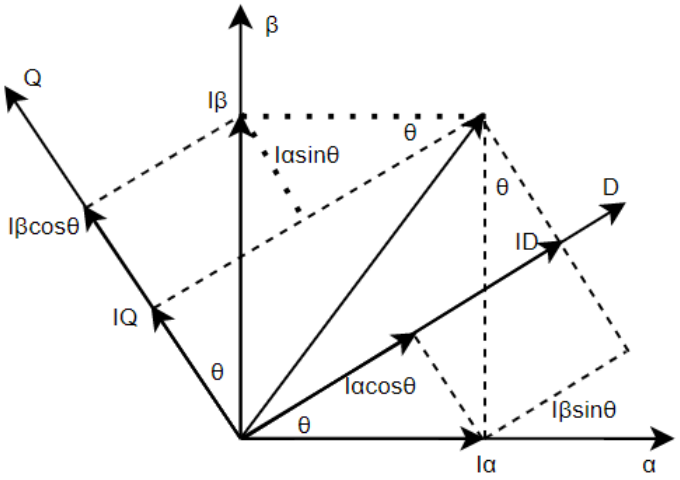
- MLIB—库前缀
- Mul—函数功能
- Q15—定标格式

2 库函数说明

2.1 MCFLIB_Park_Sat_S16

MCFLIB_Park_Sat_S16 函数计算 Park 变换，并且函数含有限幅。根据以下等式将值(磁通、电压、电流)从静止两相(α-β)正交坐标系通过矢量变换到旋转两相(d-q)正交坐标系,Park 坐标变换图和等式如下所示:

图 2.1 Park 坐标变换图



$$d = \alpha \cdot \cos(\theta) + \beta \sin(\theta) \quad (1)$$

$$q = \beta \cdot \cos(\theta) - \alpha \sin(\theta) \quad (2)$$

2.1.1 函数说明

表 2-1 MCFLIB_Park_Sat_S16 函数说明

功能名	输入类型	输出类型	执行时间
MCFLIB_Park_Sat_S16	MCFLIB_2_ALBE_T_S16、sAngle_Trig	CFLIB_2_DQ_T_S16	1500ns

表 2-2 结构体数据类型说明

结构体	变量	输入/输出	数据类型	描述
CFLIB_2_DQ_T_S16	s16D	输出	Q15_t	D 轴输出
	s16Q	输出	Q15_t	Q 轴输出
sAngle_Trig	s16Cos	输入	Q15_t	角度余弦值
	s16Sin	输入	Q15_t	角度正弦值
MCFLIB_2_ALBE_T_S16	s16Alpha	输入	Q15_t	α 输入
	s16Beta	输入	Q15_t	β 输入

2.1.2 描述

对 MCFLIB_Park_Sat_S16 ()函数有以下声明：

```
static inline void MCFLIB_Park_Sat_S16(const MCFLIB_2_ALBE_T_S16 *psIn,
const sAngle_Trig *psAnglePos, MCFLIB_2_DQ_T_S16 *psOut)
```

2.1.3 使用例程

以下示例为 MCFLIB_Park_Sat_S16 函数的用法：

```
#include "mlib.h”
MCFLIB_2_ALBE_T_S16  sIAIBe;
MCFLIB_2_DQ_T_S16  sIdq;
sAngle_Trig  Angle;
int main(void)
{
   sIAIBe.s16Alpha = Q15(0.1);
   sIAIBe.s16Beta = Q15(0.2);
   Angle.s16Sin = Q15(0.5);
   Angle.s16Cos = Q15(0.866);
}
void ISR(void)
{
   MCFLIB_Park_Sat_S16(&sIAIBe, &Angle, &sIdq);
}
```

2.2 MCFLIB_Park_S16

MCFLIB_Park_S16 函数计算 Park 变换，且函数没有限幅。根据以下等式将值(磁通、电压、电流)从静止两相(α - β)正交坐标系变换到旋转两相(d-q)正交坐标系:等式如下所示:

$$d = \alpha \cdot \cos(\theta) + \beta \sin(\theta) \quad (3)$$

$$q = \beta \cdot \cos(\theta) - \alpha \sin(\theta) \quad (4)$$

2.2.1 函数说明

表 2-3 MCFLIB_Park_S16 函数说明

功能名	输入类型	输出类型	执行时间
MCFLIB_Park_S16	MCFLIB_2_ALBE_T_S16、sAngle_Trig	MCFLIB_2_DQ_T_S16	594ns

表 2-4 结构体数据类型说明

结构体	变量	输入/输出	数据类型	描述

CFLIB_2_DQ_T_S16	s16D	输出	Q15_t	D 轴输出
	s16Q	输出	Q15_t	Q 轴输出
sAngle_Trig	s16Cos	输入	Q15_t	角度余弦值
	s16Sin	输入	Q15_t	角度正弦值
MCFLIB_2_ALBE_T_S16	s16Alpha	输入	Q15_t	α 输入
	s16Beta	输入	Q15_t	β 输入

2.2.2 描述

对 MCFLIB_Park_S16 ()函数有以下声明：

```
static inline void MCFLIB_Park_Sat_S16(const MCFLIB_2_ALBE_T_S16
*psIn,const sAngle_Trig *psAnglePos, MCFLIB_2_DQ_T_S16 *psOut )
```

2.2.3 使用例程

以下示例为 MCFLIB_Park_S16 函数的用法：

```
1.  #include "mlib.h”
2.  MCFLIB_2_ALBE_T_S16  sIAIBe;
3.  MCFLIB_2_DQ_T_S16  sIdq;
4.  sAngle_Trig  Angle;
5.  int main(void)
6.  {
7.     sIAIBe.s16Alpha = Q15(0.1);
8.     sIAIBe.s16Beta = Q15(0.2);
9.     Angle.s16Sin = Q15(0.5);
10.    Angle.s16Cos = Q15(0.866);
11. }
12. void ISR(void)
13. {
14.    MCFLIB_Park_S16(&sIAIBe, &Angle, &sIdq);
15. }
```

2.3 MCFLIB_InvPark_Sat_S16

MCFLIB_InvPark_Sat_S16 函数计算 Park 反变换，并且函数含有限幅。根据以下等式将值(磁通、电压、电流)从旋转两相(d-q)正交坐标系到静止两相(α - β)正交坐标系变换，等式如下所示:

$$\alpha = d\cos(\theta) - q\sin(\theta) \quad (5)$$

$$\beta = d\sin(\theta) + q\mathrm{s}\cos(\theta) \quad (6)$$

2.3.1 函数说明

表 2-5 MCFLIB_InvPark_Sat_S16 函数说明

功能名	输入类型	输出类型	执行时间
MCFLIB_InvPark_Sat_S16	CFLIB_2_DQ_T_S16 、sAngle_Trig	MCFLIB_2_ALBE_T_S16	1496ns

表 2-6 结构体数据类型说明

结构体	变量	输入/输出	数据类型	描述
CFLIB_2_DQ_T_S16	s16D	输入	Q15_t	D 轴输入
	s16Q	输入	Q15_t	Q 轴输入
sAngle_Trig	s16Cos	输入	Q15_t	角度余弦值
	s16Sin	输入	Q15_t	角度正弦值
MCFLIB_2_ALBE_T_S16	s16Alpha	输出	Q15_t	α 输出
	s16Beta	输出	Q15_t	β 输出

2.3.2 描述

对 MCFLIB_InvPark_Sat_S16 ()函数有以下声明：

```
static inline void MCFLIB_InvPark_Sat_S16 (const MCFLIB_2_DQ_T_S16
*psIn,const sAngle_Trig  *psAnglePos, MCFLIB_2_ALBE_T_S16 *psOut)
```

2.3.3 使用例程

以下示例为 MCFLIB_InvPark_Sat_S16 函数的用法：

```
#include "mlib.h”
MCFLIB_2_ALBE_T_S16  sIAIBe;
MCFLIB_2_DQ_T_S16  sIdq;
sAngle_Trig  Angle;
int main(void)
{
   sIAIBe.s16D = Q15(0.1);
   sIAIBe.s16Q = Q15(0.2);
   Angle.s16Sin = Q15(0.5);
   Angle.s16Cos = Q15(0.866);
}
void ISR(void)
{
   MCFLIB_InvPark_Sat_S16(&sIdq, &Angle, &sIAIBe);
}
```

2.4 MCFLIB_InPark_S16

MCFLIB_InPark _S16 函数计算 Park 反变换，且函数没有限幅。根据以下等式将值(磁通、电压、电流)从旋转两相(d-q)正交坐标系到静止两相(α - β)正交坐标系变换:等式如下所示:

$$\alpha = d\cos(\theta) - q\sin(\theta) \quad (7)$$

$$\beta = d\sin(\theta) + q\mathrm{s}\cos(\theta) \quad (8)$$

2.4.1 函数说明

表 2-7 MCFLIB_InPark_S16 函数说明

功能名	输入类型	输出类型	执行时间
MCFLIB_InPark_S16	CFLIB_2_DQ_T_S16、 sAngle_Trig	MCFLIB_2_ALBE_T_S16	596ns

表 2-8 结构体数据类型说明

结构体	变量	输入/输出	数据类型	描述
CFLIB_2_DQ_T_S16	s16D	输入	Q15_t	D 轴输入
	s16Q	输入	Q15_t	Q 轴输入
sAngle_Trig	s16Cos	输入	Q15_t	角度余弦值
	s16Sin	输入	Q15_t	角度正弦值
MCFLIB_2_ALBE_T_S16	s16Alpha	输出	Q15_t	α 输出
	s16Beta	输出	Q15_t	β 输出

2.4.2 描述

对 MCFLIB_InPark_S16 ()函数有以下声明：

```
static inline void MCFLIB_InPark_S16 (const MCFLIB_2_DQ_T_S16
*psIn, const sAngle_Trig *psAnglePos, MCFLIB_2_ALBE_T_S16 *psOut)
```

2.4.3 使用例程

以下示例为 MCFLIB_InPark_S16 函数的用法：

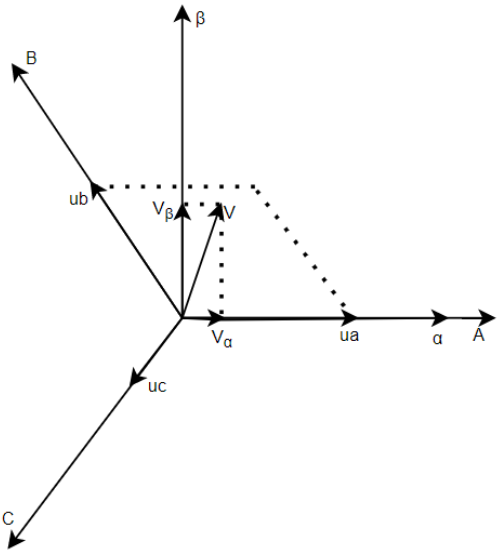
```
#include "mlib.h”
MCFLIB_2_ALBE_T_S16  sIAIBe;
MCFLIB_2_DQ_T_S16  sIdq;
sAngle_Trig  Angle;
int main(void)
{
   sIAIBe.s16D = Q15(0.1);
   sIAIBe.s16Q = Q15(0.2);
```

```
9.      Angle.s16Sin = Q15(0.5);
10.     Angle.s16Cos = Q15(0.866);
11.  }
12.  void ISR(void)
13.  {
14.     MCFLIB_InvPark_Sat_S16(&sIdq, &Angle, &sIAIBe);
15.  }
```

2.5 MCFLIB_Clark_Sat_S16

MCFLIB_Clark_Sat_S16 函数计算 Clark 变换，并且函数含有限幅。根据以下等式将值(磁通、电压、电流)从三相坐标系矢通过量变换到两相(α-β)正交坐标系，Clark 坐标变换图和等式如下所示:

图 2.2 Clark 坐标变换图



$$\alpha = a \tag{9}$$

$$\beta = \frac{1}{\sqrt{3}}b - \frac{1}{\sqrt{3}}c \tag{10}$$

2.5.1 函数说明

表 2-9 MCFLIB_Clark_Sat_S16 函数说明

功能名	输入类型	输出类型	执行时间
MCFLIB_Clark_Sat_S16	MCFLIB_3_ABC_T_S16	MCFLIB_2_ALBE_T_S16	668ns

表 2-10 结构体数据类型说明

结构体	变量	输入/输出	数据类型	描述
MCFLIB_2_ALBE_T_S16	s16Alpha	输出	Q15_t	α 输出
	s16Beta	输出	Q15_t	β 输出
MCFLIB_3_ABC_T_S16	s16A	输入	Q15_t	A 相输入

	s16B	输入	Q15_t	B 相输入
	s16C	输入	Q15_t	C 相输入

2.5.2 描述

对 MCFLIB_Clark_Sat_S16 ()函数有以下声明：

```
static inline void MCFLIB_Clark_Sat_S16 (const MCFLIB_3_ABC_T_S16
*psIn, MCFLIB_2_ALBE_T_S16 *psOut)
```

2.5.3 使用例程

以下示例为 MCFLIB_Clark_Sat_S16 函数的用法：

```
#include "mlib.h”
MCFLIB_3_ABC_T_S16  sIabc;
MCFLIB_2_ALBE_T_S16  sIAIBe;
int main(void)
{
   sIabc.s16A = Q15(0.1);
   sIabc.s16B = Q15(0.2);
   sIabc.s16C = Q15(0.4);
}
void ISR(void)
{
   MCFLIB_Clark_Sat_S16(&sIabc, & sIAIBe);
}
```

2.6 MCFLIB_InClark_Sat_S16

MCFLIB_InClark_Sat_S16 函数计算 Clark 反变换，并且函数含有限幅。根据以下等式将值(磁通、电压、电流)从两相(α - β)正交坐标系变换到三相坐标系，等式如下所示:

$$a = \alpha \tag{11}$$

$$b = -\frac{1}{2}\alpha + \frac{\sqrt{3}}{2}\beta \tag{12}$$

$$c = -(a + b) \tag{13}$$

2.6.1 函数说明

表 2-11 MCFLIB_InCark_Sat_S16 函数说明

功能名	输入类型	输出类型	执行时间
MCFLIB_InClark_Sat_S16	MCFLIB_2_ALBE_T_S16	MCFLIB_3_ABC_T_S16	1188ns

表 2-12 结构体数据类型说明

结构体	变量	输入/输出	数据类型	描述
MCFLIB_2_ALBE_T_S16	s16Alpha	输入	Q15_t	A 轴输入
	s16Beta	输入	Q15_t	B 轴输入
MCFLIB_3_ABC_T_S16	s16A	输出	Q15_t	A 相输出
	s16B	输出	Q15_t	B 相输出
	s16C	输出	Q15_t	C 相输出

2.6.2 描述

对 MCFLIB_InClark_Sat_S16 ()函数有以下声明：

```
static inline void MCFLIB_InClark_Sat_S16 (const MCFLIB_2_ALBE_T_S16
*psIn, MCFLIB_3_ABC_T_S16 *psOut)
```

2.6.3 使用例程

以下示例为 MCFLIB_InClark_Sat_S16 函数的用法：

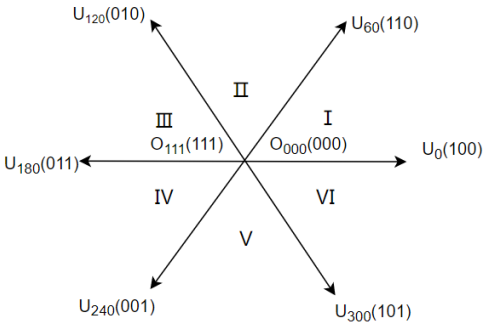
```
1.  #include "mlib.h”
2.  MCFLIB_3_ABC_T_S16  sIabc;
3.  MCFLIB_2_ALBE_T_S16  sIAIBe;
4.  int main(void)
5.  {
6.     sIAIBe.s16Alpha = Q15(0.2);
7.     sIAIBe.s16Beta = Q15(0.4);
8.  }
9.  void ISR(void)
10. {
11.    MCFLIB_InvClark_Sat_S16(&sIAIBe, &sIabc);
12. }
```

2.7 MCF_Svm_7

MCF_Svm_7 函数为 7 段式空间电压矢量调制，函数利用正弦调制技术计算出适当占空比。

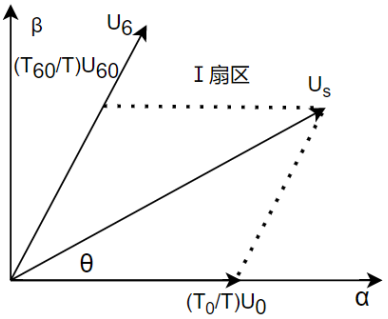
先定义六个非零矢量 U_0、U_{60}、U_{120}、U_{180}、U_{240} 和 U_{300}，以及两个零矢量 O_{111} 和 O_{000}，因此，标准空间矢量调制的原理在于在一定时间内应用适当的开关状态，从而合成参考电压矢量。

图 2-3 六扇区电压空间矢量



且标准空间矢量调制的目的是用基本空间矢量组成的开关模式合适的组合来逼近参考定子电压矢量 U_s，具体说明如下图所示：

图 2-4 电压矢量的投影



具体要实现 SVPWM 信号的实时调制，首先需要知道参考电压矢量 U_s 所在的区间位置，然后利用所在扇区的相邻两电压矢量和适当的零矢量来合成参考电压矢量，具体步骤如下：

定义三个辅助变量：

$$\begin{cases} X = U_\beta \\ Y = \frac{1}{2}(\sqrt{3}U_\alpha + U_\beta) \\ Z = \frac{1}{2}(-\sqrt{3}U_\alpha + U_\beta) \end{cases} \quad (14)$$

再定义两个有效矢量执行时间 t_1、t_2，表示相应扇区中基本空间矢量的占空比。例如，对于第一扇区，t_1 和 t_2 表示基本空间矢量 U_0 和 U_{60} 的占空比；对于第二扇区，t_1 和 t_2 表示基本空间矢量 U_{60} 和 U_{120} 的占空比，依此类推。

表 2-13 列出了 t_1 和 t_2 的表达式。

表 2-13 t_1 和 t_2 表达式

扇区	U_0、U_{60}	U_{60}、U_{120}	U_{120}、U_{180}	U_{180}、U_{240}	U_{240}、U_{300}	U_{300}、U_0
t_1	X	Z	-X	Z	-Z	Y
t_2	-Z	Y	Z	-X	-Y	-X

为了确定辅助变量 X、Y 和 Z，需要计算出扇区的区间，本文采用的方法是使用

改进的 Clark 反变换，将 α 和 β 轴转换为三相变量u_{ref1}、u_{ref2}和u_{ref3}，用于直接计算扇区号，具体如下所示。

$$\begin{cases} u_{ref1} = U_\beta \\ u_{ref2} = \frac{1}{2}(\sqrt{3}U_\alpha - U_\beta) \\ u_{ref3} = \frac{1}{2}(-\sqrt{3}U_\alpha - U_\beta) \end{cases} \quad (15)$$

根据u_{ref1}、u_{ref2}和u_{ref3}判断具体的扇区号，扇区号判断方法如下图所示：

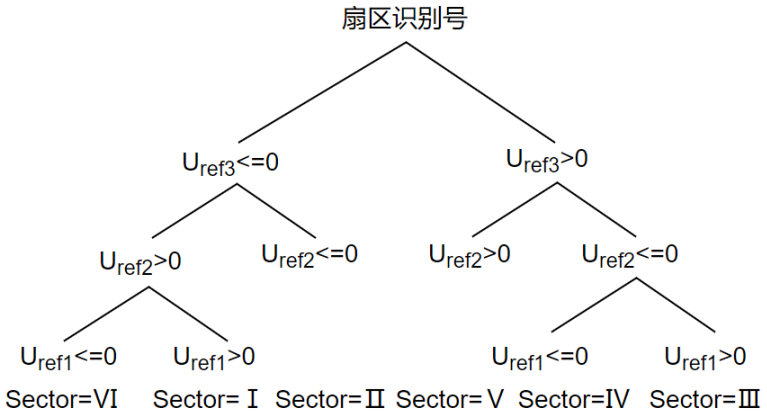


图 2-4 扇区号判断

相应占空比的变量t_1、t_2和t_3根据以下等式计算:

$$\begin{cases} t_1 = (T - t\_1 - t\_2)/2 \\ t_2 = t_1 + t\_1 \\ t_3 = t_2 + t\_2 \end{cases} \quad (16)$$

其中T是切换周期，t_1和t_2是为相应扇区有效矢量执行时间，下一步是将计算得到的占空比t_1、t_2和t_3，从而来控制开关管的开通关断时间，具体见表 2-14。

表 2-14 pwm 计算

扇区	U_0、U_{60}	U_{60}、U_{120}	U_{120}、U_{180}	U_{180}、U_{240}	U_{240}、U_{300}	U_{300}、U_0
pwm_a	t3	t2	t1	t1	t2	t3
pwm_b	t2	t3	t3	t2	t1	t1
pwm_c	t1	t1	t2	t3	t3	t2

表中，pwm_a、pwm_b、pwm_c 为各相 pwm 信号，然后分别将三个值的数据写入对应的比较寄存器就可实现 SVPWM 的算法。

2.7.1 函数说明

表 2-15 MCF_Svm_7 函数说明

功能名	输入类型	输出类型	执行时间
MCF_Svm_S16	SVM_TypeDef	uint16_t	1850ns

表 2-16 结构体数据类型说明

结构体	变量	输入/输出	数据类型	描述
SVM_TypeDef	u8Section	输出	uint8_t	扇区输出（未赋值）
	s16U_Alpha	输入	int16_t	α 轴输入电压
	s16U_Beta	输入	int16_t	β 轴输入电压
	t_1	输出	int16_t	有效矢量执行时间
	t_2	输出	int16_t	有效矢量执行时间
	u16sector	输出	uint16_t	扇区输出（未赋值）
	s16DutyA	输出	int16_t	A 相占空比
	s16DutyB	输出	int16_t	B 相占空比
	s16DutyC	输出	int16_t	C 相占空比

2.7.2 描述

对 MCF_Svm_7 函数有以下声明：

```
static inline uint16_t MCF_Svm_7 (SVM_TypeDef *sSVM);
```

2.7.3 使用例程

以下示例为 MCF_Svm_7 函数的用法：

```
1.  #include "mlib.h”
2.  uint16_t  u16Sector;
3.  SVM_TypeDef  SVPWM;
4.  int main(void)
5.  {
6.     SVPWM.s16U_Alpha = Q15(0.1);
7.     SVPWM.s16U_Beta = Q15(0.05);
8.     SVPWM.s16DutyA = Q15(0.0);
9.     SVPWM.s16DutyB = Q15(0.0);
10.    SVPWM.s16DutyC = Q15(0.0);
11. }
12. void ISR(void)
13. {
14.    u16Sector = MCF_Svm_7 (&SVM);
15. }
```

3 修改记录

表 3-1 修改记录

日期	版本	内容
2022/05/13	1.0	AN6003 初始版本发布