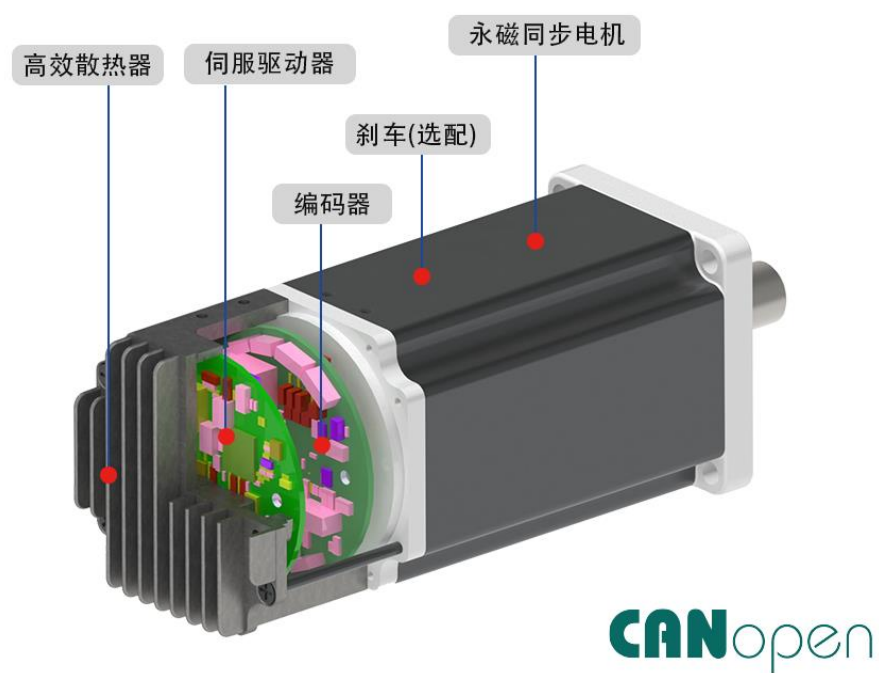




CANopenSDK_MCU 使用手册

版 本 号：A



北京立迈胜控制技术有限公司
Beijing NiMotion Control Technology Co., Ltd.

目 录

1	关于手册	1
1.1	简介	1
1.2	适用范围	1
1.3	版本信息	1
2	概述	2
2.1	软件包介绍	2
2.2	使用环境	4
3	接口函数库说明及其使用	5
3.1	错误码定义	5
3.2	接口库函数结构体	5
3.2.1	MOTOR_INFO	5
3.2.2	NMT_CMD	5
3.2.3	WORK_MODE	5
3.3	接口库函数说明	6
3.3.1	CAN_Mode_Init	6
3.3.2	NiM_isMotorOnline	7
3.3.3	NiM_getMotorInfo	8
3.3.4	NiM_getErrorCode	8
3.3.5	NiM_clearErrorState	8
3.3.6	NiM_readParam	8
3.3.7	NiM_writeParam	9
3.3.8	NiM_saveParams	9
3.3.9	NiM_restoreFactorySettings	9
3.3.10	NiM_changeAddr	9
3.3.11	NiM_changeWorkMode	9
3.3.12	NiM_getCurrentStatus	10
3.3.13	NiM_getCurrentPosition	10
3.3.14	NiM_moveAbsolute	10
3.3.15	NiM_moveRelative	10
3.3.16	NiM_moveVelocity	10
3.3.17	NiM_moveTorque	11
3.3.18	NiM_goHome	11
3.3.19	NiM_powerOn	11
3.3.20	NiM_powerOff	11
3.3.21	NiM_stop	11
3.3.22	NiM_fastStop	11
3.3.23	NiM_NMT	12
4	案例说明	13
4.1	一体化伺服电机在位置模式的应用案例	13
4.1.1	此应用案例要展现以下功能:	13
4.1.2	此应用案例在 nimotiontest.c 文件中具体实现。	13
4.1.3	此应用案例实现的操作流程图如下所示。	13
4.2	一体化伺服电机在速度模式的应用案例	14
4.2.1	此应用案例要展现以下功能:	14
4.2.2	此应用案例在 nimotiontest.c 文件中具体实现。	14
4.2.3	此应用案例实现的操作流程图如下所示。	14

4.3	一体化伺服电机在转矩模式的应用案例.....	15
4.3.1	此应用案例要展现以下功能：	15
4.3.2	此应用案例在 nimotiontest.c 文件中具体实现。	15
4.3.3	此应用案例实现的操作流程图如下所示.....	15

1 关于手册

1.1 简介

本手册用以说明北京立迈胜控制技术有限公司 CANopenSDK_MCU 的操作方法。

1. 关于手册：主要介绍手册、适用范围以及版本信息。
2. 概述：主要用于对工程程序及使用环境进行介绍。
3. 接口函数库说明及其使用：描述了接口函数库的功能及使用方法。
4. 案例说明：提供了一些使用案例，并对其简要说明。

1.2 适用范围

适用于 STM32F103 系列单片机。

1.3 版本信息

手册版本	日期	修改记录
A	2020/12/1	创建

2 概述

CANOpenSDK_MCU 使用手册是为了方便用户开发控制程序而设计的一体化电机二次开发包，该软件内部提供了一套简洁易用的函数接口，极大的简化了用户开发控制程序的难度和工作量。本 SDK 适用于北京立迈胜控制技术有限公司的 CANopen 总线协议的一体化电机。

2.1 软件包介绍

1. 解压压缩包后，打开文件夹，里面有一个工程程序，如图 1 所示。

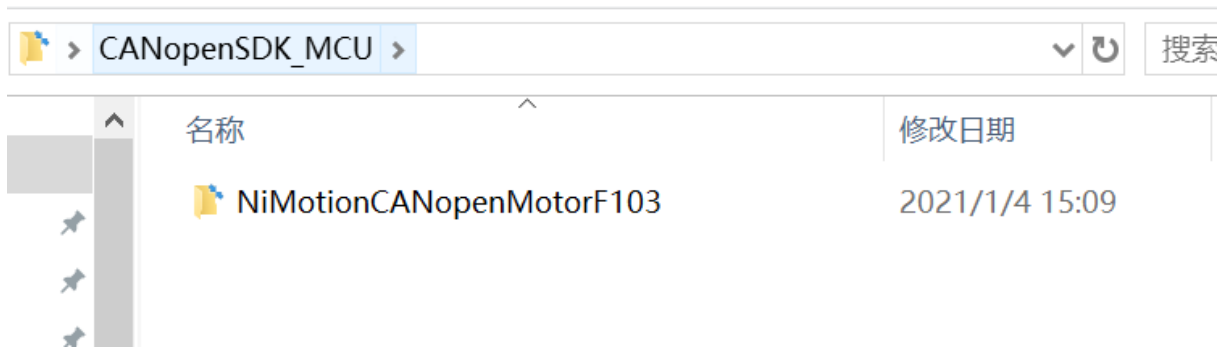


图 1

2. 打开工程程序，里面有五个文件，如图 2 所示。

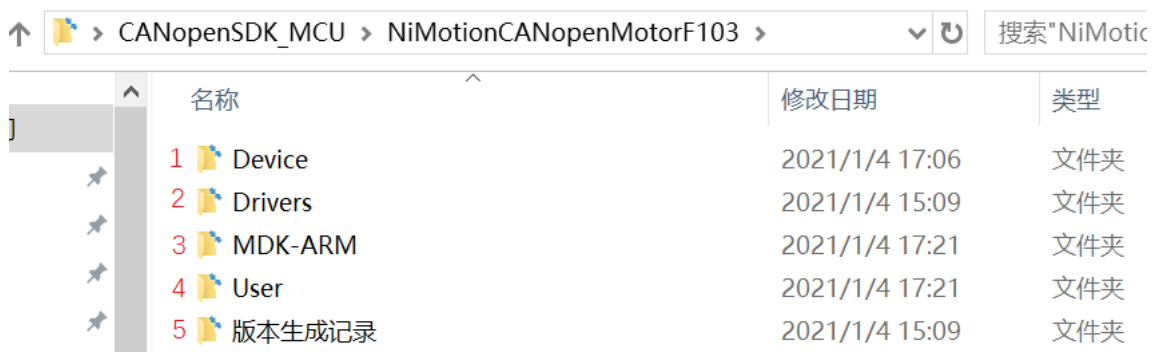


图 2

文件夹 1 存放的是驱动文件。

文件夹 2 存放的是 STM32F10X 的标准库。

文件夹 3 存放的是 Keil 工程文件。

文件夹 4 存放的是 main 文件和案例文件。

文件夹 5 存放的是版本生成记录。

3. 打开文件夹 4，里面有两个 C 文件，如图 3 所示。

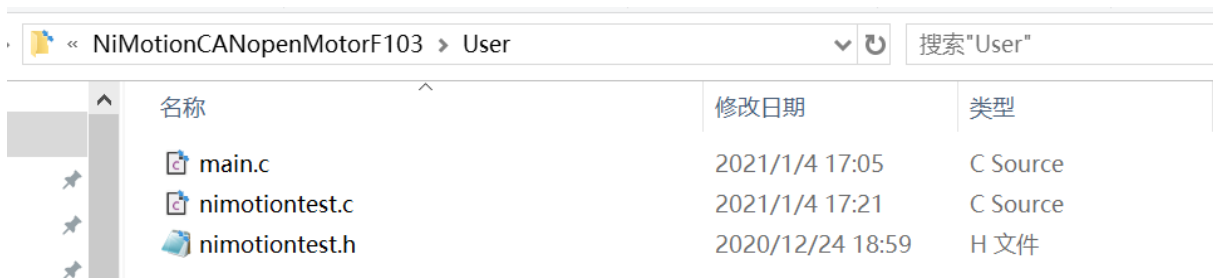


图 3

其中 main.c 文件为用户应用程序，是此工程的主函数。

其中 `nimotiontest.c` 文件为北京立迈胜控制技术有限公司控制 CANOpen 通信一体化电机使用案例。

4. 打开文件夹 1，里面有 5 个 C 文件，如图 4 所示。

NiMotionCANOpenMotorF103 > Device > C Source				搜索"Device"
名称	修改日期	类型		
1 can.c	2020/12/21 9:16	C Source		
2 delay.c	2020/12/24 17:14	C Source		
3 nimotion.c	2021/1/4 14:55	C Source		
4 stm32f10x_it.c	2020/12/24 19:13	C Source		
5 timer.c	2020/12/25 9:55	C Source		

图 4

文件 1 为 CAN 通信初始化配置文件。

文件 2 为系统延时文件，提供基于 1ms 的延时函数。

文件 3 为北京立迈胜控制技术有限公司控制 CANOpen 通信的一体化电机程序模板

文件 4 为 STM32 单片机库文件。

文件 5 为定时器计时文件，提供精确 1ms 和 2ms 计时。

5. 打开工程文件夹，双击打开工程文件，如图 5 所示。

NiMotionCANOpenMotorF103 > MDK-ARM >				搜索"MDK-ARM"
名称	修改日期	类型		
DebugConfig	2021/1/4 17:21	文件夹		
Listings	2021/1/4 15:09	文件夹		
Objects	2021/1/4 17:21	文件夹		
NiMotionCANOpenMotorF1.uvguix.asus	2021/1/4 17:21	ASUS 文件		
NiMotionCANOpenMotorF1.uvoptx	2021/1/4 17:21	UVOPTX 文件		
NiMotionCANOpenMotorF1.uvprojx	2021/1/4 17:21	磳ision5 Project		

图 5

打开文件后如图 6 所示。

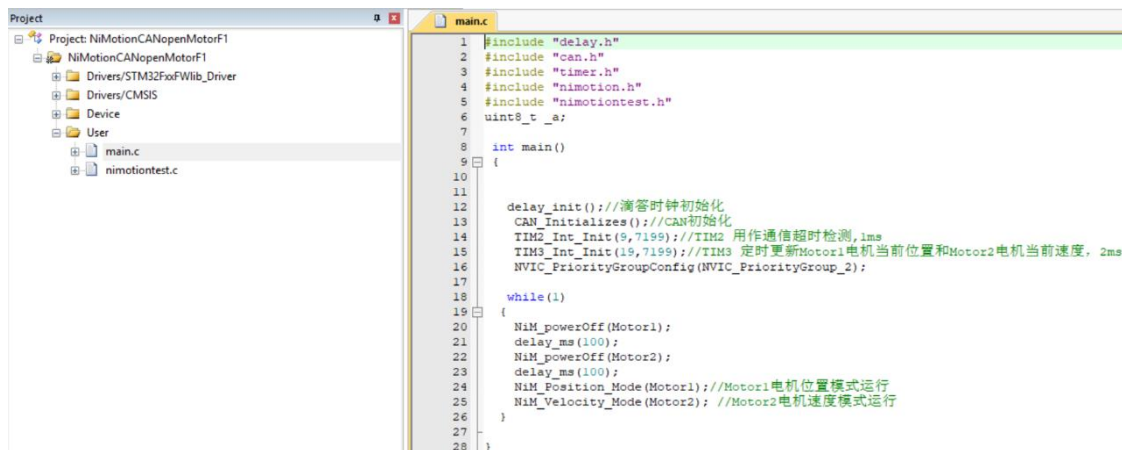


图 6

2.2 使用环境

表 1 使用环境

名称	型号
NiMotionUSBCAN 转换器 (可选)	SCM-USBCANA-A
keil5 软件 (必须安装 STM32F1 单片机 MDK 包)	5.25 版本
STM32F10X 固件库	标准库
STM32 单片机最小系统板	STM32F103CBT6
NiMotion 一体化伺服电机	CANopen 通信一体化伺服电机

3 接口函数库说明及其使用

3.1 错误码定义

请参见《一体化低压伺服电机 CANopen 通信用户手册》第 9.1 章节

3.2 接口库函数结构体

具体参见 nimotion.h 文件

3.2.1 MOTOR_INFO

描述：MOTOR_INFO 结构体包含设备信息。

```
typedef struct _MOTOR_INFO
```

```
{
```

```
    uint8_t nAddr; //电机地址
```

```
    uint32_t szSerialNumber; //电机序列号
```

```
    uint32_t szSoftVersion; //软件版本号
```

```
} MOTOR_INFO,*P_MOTOR_INFO;成员:
```

nAddr

电机地址。

szSerialNumber

电机序列号。

szSoftVersion

软件版本号。

3.2.2 NMT_CMD

描述：NMT_CMD 枚举是 NMT 网络管理枚举，即表示设备的 NMT 服务。

```
typedef enum _NMT_CMD
```

```
{
```

```
    OPERATIONAL = 0x01, //操作状态
```

```
    PRE_OPERATIONAL = 0x80, //预操作状态
```

```
    STOPPED = 0x02, //停止状态
```

```
} NMT_CMD;成员:
```

OPERATIONAL

操作状态。

PRE_OPERATIONAL

预操作状态。

STOPPED

停止状态。

3.2.3 WORK_MODE

描述：WORK_MODE 枚举是运动模式枚举，表示电机的运动模式。

```
typedef enum _WORK_MODE
```

```
{
```

```
    POSITION_MODE = 1,
```



```

    VELOCITY_MODE = 3,
    TORQUE_MODE = 4,
    GOHOME_MODE = 6
} WORK_MODE;

```

成员：

POSITION_MODE

位置模式。

VELOCITY_MODE

速度模式。

TORQUE_MODE

转矩模式。

GOHOME_MODE

原点回归模式。

3.3 接口库函数说明

序号	函数名	所在文件	功能描述
1	CAN_Mode_Init	can.c	此函数用以初始化 CAN 通信
2	NiM_isMotorOnline	nimotion.c	此函数用以判断电机是否在线
3	NiM_getMotorInfo	nimotion.c	此函数用以获取电机基本信息
4	NiM_getErrorCode	nimotion.c	此函数用以获取电机故障码
5	NiM_clearErrorState	nimotion.c	此函数用以清除电机故障
6	NiM_readParam	nimotion.c	此函数用以获取电机参数值
7	NiM_writeParam	nimotion.c	此函数用以设置电机参数值
8	NiM_saveParams	nimotion.c	此函数用以保存电机参数
9	NiM_restoreFactorySettings	nimotion.c	此函数用以恢复电机出厂参数设置
10	NiM_changeAddr	nimotion.c	此函数用以改变电机地址
11	NiM_changeWorkMode	nimotion.c	此函数用以改变电机运行模式
12	NiM_getCurrentStatus	nimotion.c	此函数用以获取当前电机状态字
13	NiM_getCurrentPosition	nimotion.c	此函数用以获取当前电机当前位置
14	NiM_moveAbsolute	nimotion.c	此函数用以电机绝对位置运动
15	NiM_moveRelative	nimotion.c	此函数用以电机相对位置运动
16	NiM_moveVelocity	nimotion.c	此函数用以速度模式下目标速度运行
17	NiM_moveTorque	nimotion.c	此函数用以转矩模式下目标转矩运行
18	NiM_goHome	nimotion.c	此函数用以原点回归
19	NiM_powerOn	nimotion.c	此函数用以电机抱机
20	NiM_powerOff	nimotion.c	此函数用以电机脱机
21	NiM_stop	nimotion.c	此函数用以电机停止当前动作
22	NiM_fastStop	nimotion.c	此函数用以电机急停
23	NiM_NMT	nimotion.c	此函数用以网络管理

3.3.1 CAN_Mode_Init

描述： 此函数用以初始化 CAN 通信。

unsigned char CAN_Mode_Init(unsigned char tsjw, unsigned char tbs2, unsigned char tbs1, uint16 brp, unsigned char mode)

参数：

tsjw

重新同步跳跃时间单元 范围：CAN_SJW_1tq~CAN_SJW_4tq

tbs2

时间段 2 的时间单元 范围：CAN_BS2_1tq~CAN_BS2_8tq

tbs1

时间段 1 的时间单元 范围：CAN_BS1_1tq~CAN_BS1_8tq

brp

波特率分频器

mode

CAN 通信模式：正常模式和环回模式。

CAN 波特率推荐配置如下表所示：

波特率	相关配置
10 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_6TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler = 450;
20 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_7TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler = 200;
50 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_6TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler = 90;
100 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_6TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler = 45;
125 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_6TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler = 36;
250 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_6TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler = 18;
500 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_6TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler = 9;
800 kBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_12TQ; can.Init.BS2 = CAN_BS2_2TQ; can.Init.Prescaler = 3;
1 MBit/sec	can.Init.SJW = CAN_SJW_1TQ; can.Init.BS1 = CAN_BS1_7TQ; can.Init.BS2 = CAN_BS2_1TQ; can.Init.Prescaler= 4;

3.3.2 NiM_isMotorOnline

描述：此函数用以判断电机是否在线。

uint8_t NiM_isMotorOnline(uint8_t nAddr, uint16_t* pOnline);

参数:**nAddr**

电机地址。

pOnline

指针，返回在线状态。

返回值: 0 成功，其它表示错误码。

3.3.3 NiM_getMotorInfo**描述:** 此函数用以获取电机基本信息。`uint8_t NiM_getMotorInfo(uint8_t nAddr, MOTOR_INFO* pInfo);`**参数:****nAddr**

电机地址。

pInfo

结构体指针，返回电机信息。

返回值: 0 成功，其它表示错误码。

3.3.4 NiM_getErrorCode**描述:** 此函数用以获取电机故障码。`uint8_t NiM_getErrorCode(uint8_t nAddr, uint16_t* pErrorCode);`**参数:****nAddr**

电机地址。

pErrorCode

指针，返回电机故障码。

返回值: 0 成功，其它表示错误码。

3.3.5 NiM_clearErrorState**描述:** 此函数用以清除电机故障。`uint8_t NiM_clearErrorState(uint8_t nAddr);`**参数:****nAddr**

电机地址。

返回值: 0 成功，其它表示错误码。

3.3.6 NiM_readParam**描述:** 此函数用以获取电机参数值。`uint8_t NiM_readParam1(uint8_t nAddr, uint16_t index, uint8_t subindex, uint8_t *data, uint8_t len);`**参数:****nAddr**

电机地址。

Index

对象索引。

Subindex

对象子索引。

data

指针，返回参数值。

Len

对象长度。

返回值: 0 成功，其它表示错误码。

3.3.7 NiM_writeParam

描述：此函数用以设置电机参数值。

uint8_t NiM_writeParam(uint8_t nAddr,uint16_t index, uint8_t subindex ,uint8_t *data, uint8_t len);

参数：

nAddr

电机地址。

Index

对象索引。

Subindex

对象子索引。

data

指针，写入参数值。

Len

对象长度。

返回值：0 成功，其它表示错误码。

3.3.8 NiM_saveParams

描述：此函数用以保存电机参数。

uint8_t NiM_saveParams(uint8_t nAddr);

参数：

nAddr

电机地址。

返回值：0 成功，其它表示错误码。

3.3.9 NiM_restoreFactorySettings

描述：此函数用以恢复电机出厂参数设置。

uint8_t NiM_restoreFactorySettings(uint8_t nAddr);

参数：

nAddr

电机地址。

返回值：0 成功，其它表示错误码。

3.3.10 NiM_changeAddr

描述：此函数用以改变电机地址。

uint8_t NiM_changeAddr(uint8_t nCurAddr, uint8_t nNewAddr);

参数：

nCurAddr

当前电机地址。

nNewAddr

新的电机地址

返回值：0 成功，其它表示错误码。

3.3.11 NiM_changeWorkMode

描述：此函数用以改变电机运行模式。

uint8_t NiM_changeWorkMode(uint8_t nAddr, WORK_MODE nMode);

参数：

nAddr

电机地址。

nMode

运行模式。

返回值：0 成功，其它表示错误码。

3.3.12 NiM_getCurrentStatus

描述: 此函数用以获取当前电机状态字。

uint8_t NiM_getCurrentStatus(uint8_t nAddr, uint16_t* pStatusWord);

参数:

nAddr

电机地址。

pStatusWord

指针，返回状态字。

返回值: 0 成功，其它表示错误码。

3.3.13 NiM_getCurrentPosition

描述: 此函数用以获取当前电机当前位置。

uint8_t NiM_getCurrentPosition(uint8_t nAddr, int32_t* pPosition);

参数:

nAddr

电机地址。

pPosition

指针，返回当前位置。

返回值: 0 成功，其它表示错误码。

3.3.14 NiM_moveAbsolute

描述: 此函数用以电机绝对位置运动。

uint8_t NiM_moveAbsolute(uint8_t nAddr, uint8_t nType, int32_t nPosition);

参数:

nAddr

电机地址。

nType

0-立即更新/1-非立即更新。

nPosition

目标位置。

返回值: 0 成功，其它表示错误码。

3.3.15 NiM_moveRelative

描述: 此函数用以电机相对位置运动。

uint8_t NiM_moveRelative(uint8_t nAddr, uint8_t nType, int32_t nDistance);

参数:

nAddr

电机地址。

nType

0-立即更新/1-非立即更新。

nDistance

运动距离。

返回值: 0 成功，其它表示错误码。

3.3.16 NiM_moveVelocity

描述: 此函数用以速度模式下目标速度运行。

uint8_t NiM_moveVelocity(uint8_t nAddr, int32_t nVelocity);

参数:

nAddr

电机地址。

nVelocity

目标速度。

返回值: 0 成功, 其它表示错误码。

3.3.17 NiM_moveTorque

描述: 此函数用以转矩模式下目标转矩运行。

uint8_t NiM_moveTorque(uint8_t nAddr, int16_t nTorque);

参数:

nAddr

电机地址。

nTorque

目标转矩。

返回值: 0 成功, 其它表示错误码。

3.3.18 NiM_goHome

描述: 此函数用以原点回归。

uint8_t NiM_goHome(uint8_t nAddr, uint8_t nType);

参数:

nAddr

电机地址。

nType

原点回归方式。

返回值: 0 成功, 其它表示错误码。

3.3.19 NiM_powerOn

描述: 此函数用以电机抱机。

uint8_t NiM_powerOn(uint8_t nAddr);

参数:

nAddr

电机地址。

返回值: 0 成功, 其它表示错误码。

3.3.20 NiM_powerOff

描述: 此函数用以电机脱机。

uint8_t NiM_powerOff(uint8_t nAddr);

参数:

nAddr

电机地址。

返回值: 0 成功, 其它表示错误码。

3.3.21 NiM_stop

描述: 此函数用以电机停止当前动作。

uint8_t NiM_stop(uint8_t nAddr);

参数:

nAddr

电机地址。

返回值: 0 成功, 其它表示错误码。

3.3.22 NiM_fastStop

描述: 此函数用以电机急停。

uint8_t NiM_fastStop(uint8_t nAddr);

参数:

nAddr

电机地址。

返回值：0 成功，其它表示错误码。

3.3.23 NiM_NMT

描述：此函数用以 NMT 网络管理。

`uint8_t NiM_NMT(uint8_t nAddr,NMT_CMD nCmd);`

参数：

`nAddr`

电机地址。

`nCmd`

命令字。

返回值：0 成功，其它表示错误码。

4 案例说明

4.1 一体化伺服电机在位置模式的应用案例

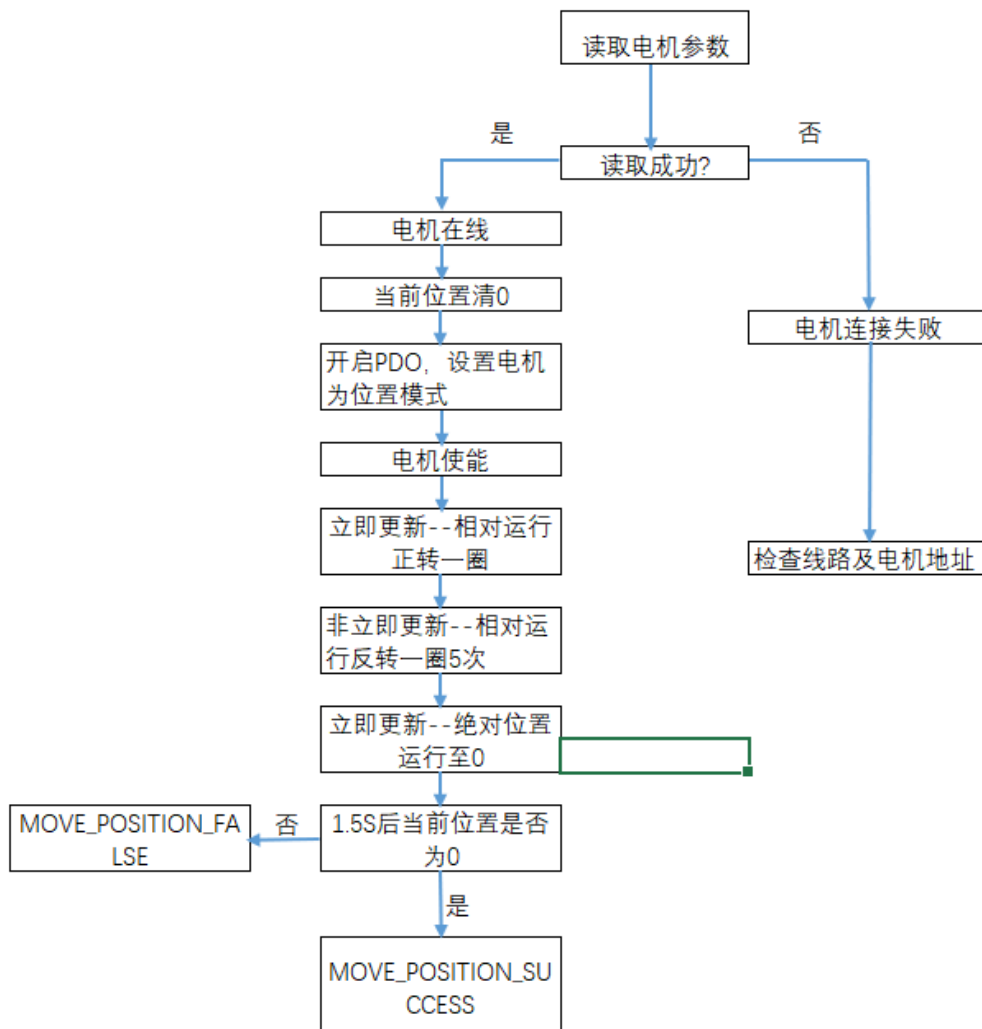
4.1.1 此应用案例要展现以下功能：

- 检测伺服电机是否在线，防止出现伺服电机未连接或连接失败的情况。
- 位置模式下相对位置正反转一圈，为了展现伺服电机的高响应和定位的准确性。
- 位置模式下伺服以立即更新的相对模式/绝对模式运行时，在未到达绝对目标位置之前可以执行最新动作指令。
- 位置模式下伺服以非立即更新的相对模式运行，在未到达绝对目标位置之前可以进行位置指令缓存，最大支持 5 个。

4.1.2 此应用案例在nimotiontest.c文件中具体实现。

4.1.3 此应用案例实现的流程图如下所示。

位置模式流程图



4.2 一体化伺服电机在速度模式的应用案例

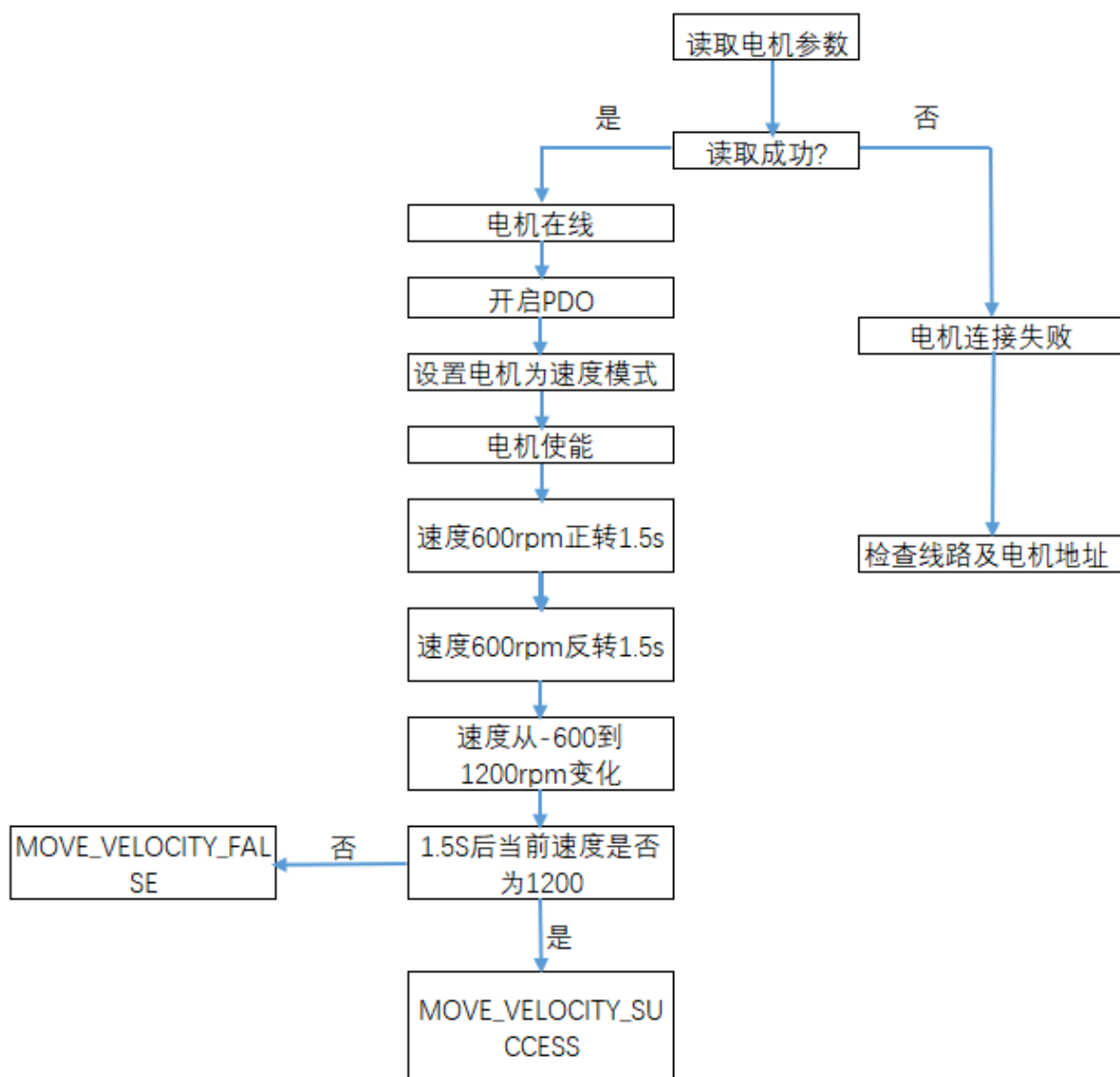
4.2.1 此应用案例要展现以下功能：

- 检测伺服电机是否在线，防止出现伺服电机未连接或连接失败的情况。
- 速度模式下正反转，为了体现在速度模式下伺服电机可随时改变方向。
- 速度模式下速度变化，正转速度 600，反转速度 600，为了体现伺服电机在速度模式下速度可随时改变。
- 速度模式下速度从反转速度 600 到正转速度 1200，为了体现伺服电机的高速响应。

4.2.2 此应用案例在nimotiontest.c文件中具体实现。

4.2.3 此应用案例实现的操作流程图如下所示。

速度模式流程图



4.3 一体化伺服电机在转矩模式的应用案例

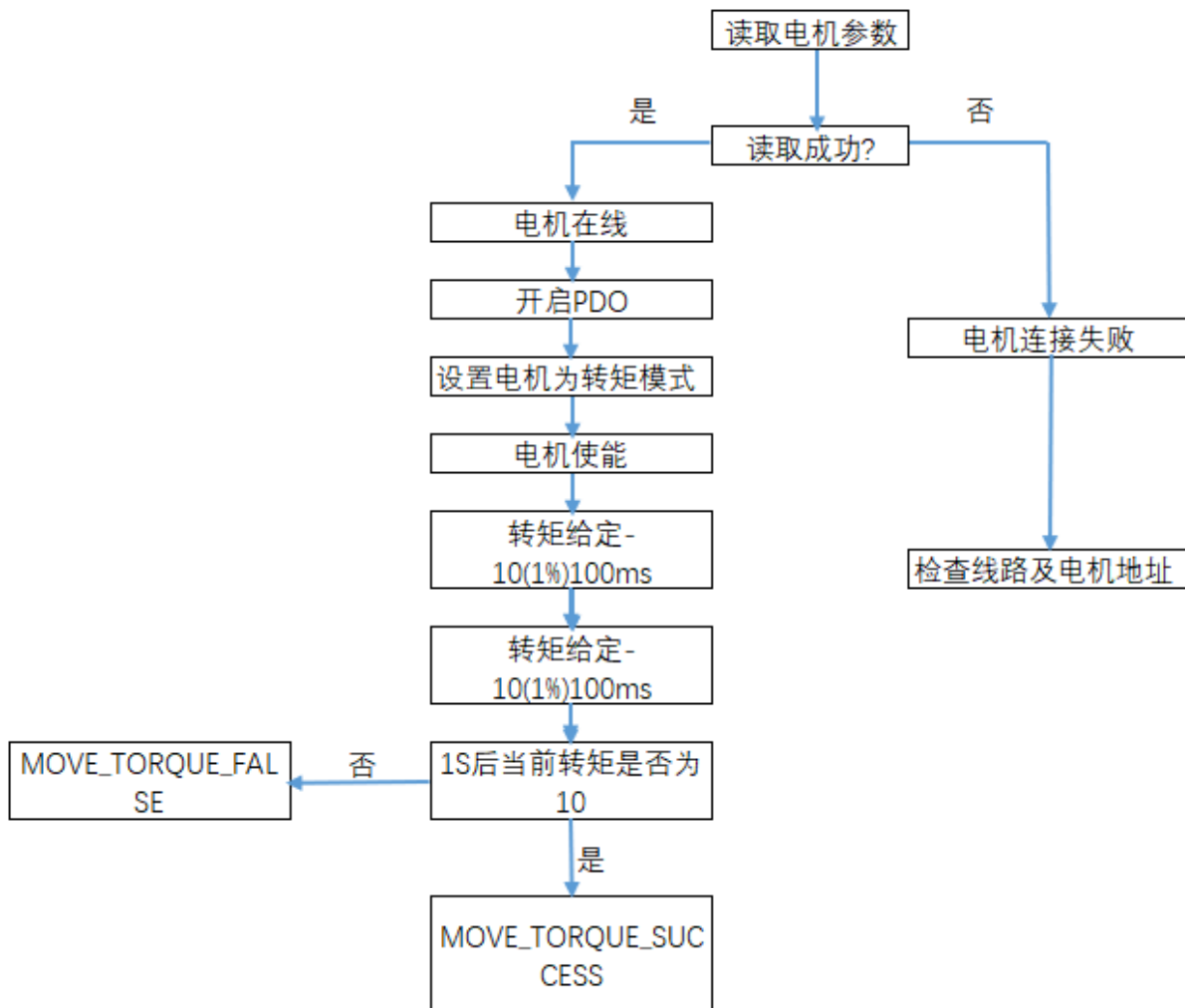
4.3.1 此应用案例要展现以下功能：

- 检测伺服电机是否在线，防止出现伺服电机未连接或连接失败的情况。
- 转矩模式下转矩变化，为了展现伺服电机在转矩模式下转矩可随时改变。
- 转矩模式下的转矩给定，为了展现伺服电机转矩控制的准确性和稳定性。

4.3.2 此应用案例在nimotiontest.c文件中具体实现。

4.3.3 此应用案例实现的操作流程图如下所示

转矩模式流程图



- 本说明书的全部内容或部分内容禁止擅自转载、拷贝。
- 产品性能、规格及外观可能因为改进，会在不经预先通知的情况下发生变化，敬请谅解。
- 我们力求使说明书的内容尽可能正确，如果您发现有什么问题或错误、遗漏之处，请与北京立迈胜控制技术有限公司联系。

北京立迈胜控制技术有限公司
Beijing NiMotion Control Technology Co., Ltd.
北京市大兴区金星路 12 号院 3 号楼
邮编：102628
电话：(010)60213882 传真：(010)60213882
邮箱：NiMotion@NiMotion.com