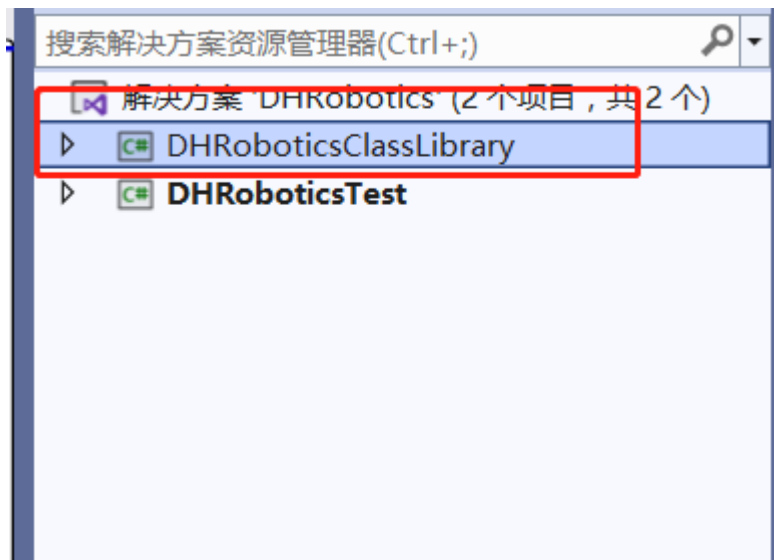


# **DH-Robotics 封装说明**

# 项目引用

添加 DHRoboticsClassLibrary 项目引用



# 使用说明

添加实例：

```
DHRoboticsService robot = new DHRoboticsService ("COM4");  
  
var result = await robot.ReadWriteRegister(Operation.Write,  
  
FunAddress.InitRobotic, 0x01);
```

实例化参数说明：

## 1. 实例化参数说明

```
public DHRoboticsService(string portName, int baudRate = 115200,  
  
Parity paritycheck = Parity.None, int dataBit = 8, StopBits stopBit =  
  
StopBits.One, byte slaveId = 0x01)
```

`portName` 为必填项，为设备端口号。波特率等提供默认值。

## 2. 读写保持寄存器操作说明（ReadWriteRegister）

`Operation` 为枚举类型，有 Read 与 Write 选择

`FunAddress` 为各个功能的寄存器地址，具体请看下面各个功能代码。

`Data` 为数据，short 类型。

## 3. 各个功能说明

### 3.1 初始化夹爪

写（执行初始化）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.InitRobotic,  
(short)InitOption.ClampingOneWayZeroSpinZero);
```

参数说明：InitOption 为初始化类型的枚举，如下：

ClampingOneWayZeroSpinZero：夹持单向回零、旋转回零

ClampingOpenZeroSpinZero：夹持张开回零、旋转回零

ClampingCloseZeroSpinZero：夹持闭合回零、旋转回零

ClampingReMark：夹持重新标定

ClampingOpenZero：夹持张开回零

ClampingCloseZeroSpinZero：夹持闭合回零

SpinZero：旋转逆时针回零

RemarkAndInit：重新标定、初始化(先旋转后夹持)

读（是否在初始化的过程中）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.InitRobotic);
```

返回数据说明：0：未处于初始化过程； 1：处于初始化过程； 2：初始化中

注：此命令读取返回的是当前是否执行初始化，若要查询是否初始化，请使用3.xx中初始化状态反馈命令。

### 3.2 夹爪力值

写（设置夹爪力值）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.Force, data);
```

其中 data 为 short 类型的数据。且后台会判断数据大小（0x14~0x64，表示百分比），不在范围内则会返回错误。

读（读取设置的夹爪力值）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.Force);
```

### 3.3 夹爪位置

写（设置位置）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.Position, data);
```

其中 data 为 short 类型的数据。且后台会判断数据大小（0x00~0x03e8，表示千分比），不在范围内则会返回错误。

读（读取**设置**的位置）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.Position);
```

注：不是实时位置，是设置的位置。

### 3.4 夹爪速度

写（设置速度）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.Speed, data);
```

其中 data 为 short 类型的数据。且后台会判断数据大小（0x01~0x64，表示百分比），不在范围内则会返回错误。

读（读取**设置**的速度）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.Position);
```

### 3.5 绝对角度低位&高位

写（设置绝对旋转角度）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.AbsoluteAngle, data);
```

其中 data 为 short 类型的数据。

注：

角度值为：低位数据+高位数据\*32768，因为data是short即-32768~32767，超过该值则绝对角度加一或减一，但是无特殊情况请勿写入绝对高位（AbsoluteAngleHigh）数据，单位旋转32768°。

读（读取**设置**的位置）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.AbsoluteAngle);
```

注：若需要读取实时角度，请使用下方的**旋转角度实时反馈**。

### 3.6 旋转速度

写（设置旋转速度）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.SpinSpeed,  
data);
```

其中 data 为 short 类型的数据。且后台会判断数据大小（0x01~0x64,表示百分比），不在范围内则会返回错误。

读（读取**设置**的旋转速度）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.SpinSpeed);
```

### 3.7 旋转力

写（设置旋转力）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.SpinForce,  
data);
```

其中 data 为 short 类型的数据。且后台会判断数据大小（0x014~0x64，表示百分比），不在范围内则会返回错误。

读（读取**设置**的旋转力）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.SpinForce);
```

### 3.8 相对旋转角度

写（设置相对旋转角度）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.RelativeAngle,  
data);
```

其中 data 为 short 类型的数据。

读（读取**设置**的相对旋转角度）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.RelativeAngle);
```

### 3.9 初始化状态反馈

读（读取初始化状态反馈）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.InitStatus);
```

返回值说明：0：未初始化；1：初始化成功

### 3.10 夹持状态反馈

读（读取夹持状态）：

```
robot.ReadWriteRegister(Operation.Read,  
FunAddress.ClampingStatus);
```

返回值说明：

00：夹爪处于正在运动状态。01：夹爪停止运动，且夹爪未检测到夹到物体。02：夹爪停止运动，且夹爪检测到夹到物体。03：夹爪检测到夹住物体后，发现物体掉落。

### 3.11 夹爪位置反馈

读（读取夹爪实时位置）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.RealPosition);
```

### 3.12 错误/警告/信息/反馈

读（错误信息）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.ErrorCode);
```

返回数据说明：0：无任何问题;04 过热;08 过载;11过速

### 3.13 旋转角度实时反馈

读（读取实时旋转角度）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.RealTimeAngle);
```

### 3.14 旋转初始化状态反馈

读（读取旋转初始化状态）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.SpinInitStatus);
```

返回值说明：0：未初始化；1：初始化成功；2：正在初始化

### 3.15 旋转状态反馈

读（读取旋转状态）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.SpinStatus);
```

返回数据说明：0: 运动中, 1: 到达位置; 2: 堵转

### 3.16 写入保存

写（将修改的数据写入 flash,以便下次上电能继续使用）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.Save);
```

### 3.17 初始化方向

写（设置夹爪初始化方向）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.InitDirection,  
data);
```

其中data为0x00或者0x01,当写入0时, 夹爪会运行到最大的张开位置, 并作为初始起点。当写入1时, 夹爪会运行到最小的闭合位置, 并作为初始起点。

读（读取夹爪初始化方向）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.InitDirection);
```

### 3.18 旋转停止

写（停止旋转）：写入 0x01

```
robot.ReadWriteRegister(Operation.Write, FunAddress.StopSpin, 0x01);
```

### 3.19 自动初始化

写（设置上电自动初始化）：

```
robot.ReadWriteRegister(Operation.Write,  
FunAddress.PowerOnAutomaticInitialization, data);
```

其中data为0x00或者0x01, 0: 上电不初始化； 1: 上电自动初始化。

注：写完之后需要再3.16 Save后，重新上电有效

读（读取是否上电初始化）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.  
PowerOnAutomaticInitialization);
```

### 3.20 旋转堵转停转配置

写（设置夹爪设置旋转堵转停转配置指令）：

```
robot.ReadWriteRegister(Operation.Write, FunAddress.BlockStop,  
data);
```

其中data为0x00或者0x01, 0: 旋转堵转不停转； 1: 旋转堵转停转。

注：写完之后需要再3.16 Save后，重新上电有效

读（读取夹爪设置旋转堵转停转配置指令）：

```
robot.ReadWriteRegister(Operation.Read, FunAddress.BlockStop);
```

### 3.21 复位旋转角度

写（手动复位绝对旋转角度）：

robot.ReadWriteRegister(Operation.**Write**, FunAddress.**Reset**, data);

其中data为0x01或者0xA5,

写入**01**: 将外部轴位置复位为**1圈**内的当前位置, 夹爪不动: 复位前的位置大于0度时: 0~360, 复位前的位置小于0度时: -360~0

写入**A5**: 将外部轴位置复位为**1圈**内的位置, 夹爪运动到0°。

### 3.21 其他命令

可设置设备ID (**ID**)、波特率(**BoundRat**)、停止位 (**StopBit**)、校验位 (**CheckBit**)、IO参数测试 (**IOPparameterTest**)、IO模式开关 (**IOOpenClose**)

向对应的地址写入参数即可, 由于这些命令不常用, 在此不一一列举。

## 注意事项:

1. 本次封装使用了 NModbus 及 System.IO.Ports 库。
2. 设置完寄存器值后, 若想下次上电继续使用, 需要写入保存。