

mcHOME_RESTORE_POS 简介

2023 贝加莱技术团队

发布日期 2023 年 12 月 20 日

版本号 1.0.0

Contents

1. mcHOME_RESTORE_POS 简介3

1.1. MC_Home5

1.1.1. Periodic axes7

1.1.2. Homing modes ("HomingMode")8

1.2. mcHOME_RESTORE 寻参配置案例15

1.3. 功能演示效果与测试案例16

1.4. 使用 mcHOME_RESTORE 命令需注意调用顺序.....17

1.5. 可能存在的问题与注意事项31



1. mcHOME_RESTORE_POS 简介

简要说明

mcHOME_RESTORE_POS 是贝加莱 PLCOpen 回零模式功能块 MC_Home 多种模式中的一种。

从永久保持变量中恢复位置

此模式主要能实现功能为：多圈编码器设备断电后位置记忆及处理

使用限制

使用 MC_ACP10_V5.07.0 及更高版本

PLC 与工控机需要确认支持永久保持变量

测试案例

伺服上电使能后，点击零点标定，确定当前位置为零点，然后让电机运动一段距离，记下该位置值，然后 PLC 和伺服都断电重启，因为绝对值编码器有位置（HW Position）断电记忆功能（编码器位置和程序中实际位置不一定相等），所以程序中会有一个位置值（SW Position），但是这个位置和断电器的位置不相等，需要点击 Home 后，如果位置变为断电前位置，则说明测试成功。

帮助介绍与说明

[mcHOME_RESTORE_POS \(restore position from remanent memory\)](#)

[Example illustrating the procedure for calibrating and restoring the position](#)

目录

[mcHOME_RESTORE 寻参配置案例](#)

[功能演示效果与测试案例](#)

[使用 mcHOME_RESTORE 命令需注意调用顺序](#)

[可能存在的问题与注意事项](#)

常见使用场景

mcHOMING_RESTORE_POS 在处理无穷无尽的位置轴时更有必要，因为在 32 位位置整数每次复叠时，绝对偏移量都会发生变化。

传动速比和多圈 4096 没法整除，所以是 init_home 需要使用
mcHOME_RESTORE_POS (restore position from permanent memory)模式

1.1. MC_Home

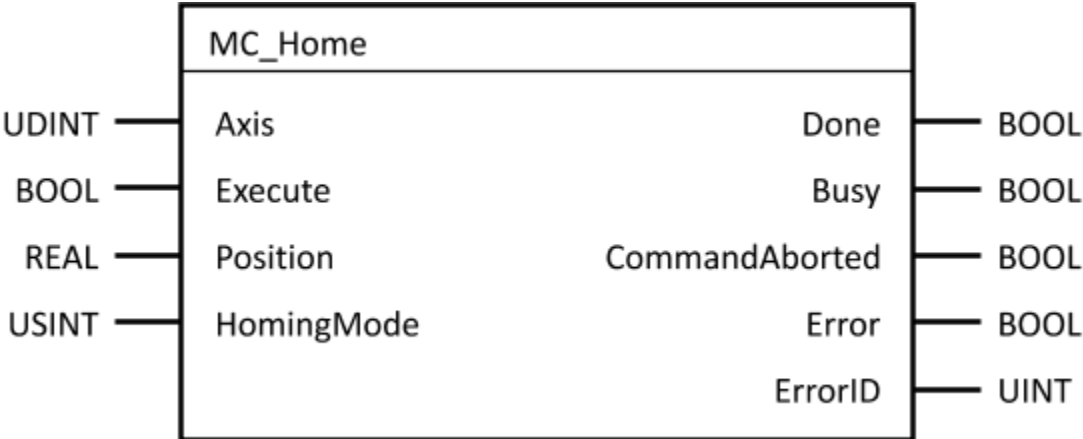
该功能块执行轴的归位。如果所选 "HomingMode（归位模式）"需要一个归位程序，则该功能块也将启动该程序。

信息：
对于虚拟轴，该功能块只能在归位模式为 [mcHOME_DEFAULT](#)、[mcHOME_DIRECT](#)、[mcHOME_AXIS_REF](#) 或 [mcHOME_RESTORE_POS](#) 时使用。在所有其他情况下，它将报告错误 [29217：输入参数无效](#)。

使用要求
允许启用功能块的 PLCopen 轴状态

- [Disabled](#)
- [Standstill](#)

Function block



Parameter

Class	I/O	Parameter	Data type	Description
B	IN	Axis	UDINT	Axis reference
B	IN	Execute	BOOL	Execution of the function block begins on a rising edge of this input.
B	IN	Position	REAL	Absolute position after the reference signal has been detected [PLCopen units] Note:



				In modes mcHOME RESTORE POS and mcHOME AXIS REF , the position is not applied by this input.
E	IN	Homing Mode	USINT	Homing modes: mcHOME DEFAULT (all parameters except "Position" are applied from the init parameter module) mcHOME ABS SWITCH mcHOME SWITCH GATE mcHOME LIMIT SWITCH mcHOME REF PULSE mcHOME DIRECT mcHOME ABSOLUTE (offset for the absolute encoder) mcHOME ABSOLUTE CORR (offset for the absolute encoder with counting range correction) mcHOME DCM (distance-coded reference marks) mcHOME DCM CORR (distance-coded reference marks with counting range correction) mcHOME RESTORE POS (restore position from remanent memory, see MC BR InitEndlessPosition) mcHOME AXIS REF (homing with the parameters from the axis structure) (V2.210 and later) mcHOME BLOCK TORQUE (homing to mechanical limit) (V2.360 and later) mcHOME BLOCK DS (homing to mechanical limit) (V2.360 and later)
B	OUT	Done	BOOL	Execution successful. The function block is finished.
E	OUT	Busy	BOOL	The function block is active and must continue to be called.
E	OUT	CommandAborted	BOOL	Function block aborted by another function block.
B	OUT	Error	BOOL	An error occurred during execution.
E	OUT	ErrorID	UINT	Error numbers

Table: Parameter [MC Home](#)

Topics in this section:

- [Periodic axes](#)
- [Homing modes \("HomingMode"\)](#)
- Error numbers

1.1.1. Periodic axes

For periodic axes, movements can also be started without having first homed the axis.

对于周期性轴，也可以在未归位的情况下开始运动。

1.1.2. Homing modes ("HomingMode")

除 [mcHOME_AXIS_REF](#) 和 [mcHOME_RESTORE_POS](#) 外，对于下文所述的所有模式，均使用轴结构中除 "位置 "和归位模式外的所有参数。

Topics in this section:

- mcHOME_ABS_SWITCH (performs homing with absolute reference switch)
- mcHOME_SWITCH_GATE (homing with reference switch gate)
- mcHOME_LIMIT_SWITCH (homing with hardware limit switch)
- mcHOME_DIRECT (direct homing without reference pulse)
- mcHOME_REF_PULSE (direct homing with reference pulse)
- mcHOME_ABSOLUTE (homing by setting the homing offset)
- mcHOME_ABSOLUTE_CORR (homing by setting the homing offset with counting range correction)
- mcHOME_DCM (homing using interval-encoded reference marks)
- mcHOME_DCM_CORR (homing using distance-coded reference marks with counting range correction)
- [mcHOME_RESTORE_POS \(restore position from remanent memory\)](#)
- mcHOME_AXIS_REF (performs homing with the data from the axis structure)
- mcHOME_BLOCK_TORQUE (performs homing to mechanical limit, torque as criteria)

1.1.2.1. mcHOME_RESTORE_POS (restore position from remanent memory)

The address of a remanent variable of type MC_ENDLESS_POSITION must be initialized with function block [MC_BR_InitEndlessPosition](#).

The ACP10_MC library uses this variable to store all information necessary for restoring the axis position after a warm or cold restart.

The application is not permitted to write to this variable! This data is protected by a checksum. An error message will appear when attempting to restore the position if this data is overwritten.

A movement does not take place in this homing mode. The "Position" input is ignored.

An absolute measurement system must be used (at least within one encoder revolution).

When turned off, the encoder is not permitted to change by more than one half of the counting range in order for the position to be successfully restored. With a single-turn encoder, this can be ensured using a motor holding brake. A multi-turn encoder must be used in the event that the axis could coast to a standstill during a power failure.

With a virtual axis, the last position stored before the restart or power failure took place is restored.

This homing mode is intended for use with endless axes (e.g. in packaging machines and rotary tables). Nevertheless, it can also be used with axes with a limited range of movement (e.g. handling systems). This mode can also be used to handle a multi-turn encoder overflow in the middle of a limited axis (alternative to the "mcHOME_ABSOLUTE_CORR" homing mode).

必须使用功能块 [MC_BR_InitEndlessPosition](#) 初始化类型为 MC_ENDLESS_POSITION 的重定向变量地址。

ACP10_MC 库使用该变量存储热重启或冷重启后恢复轴位置所需的所有信息。

应用程序不得写入该变量！该数据受校验和保护。如果该数据被覆盖，尝试恢复位置时将出现错误信息。

在此归位模式下不进行移动。位置 "输入将被忽略。

必须使用绝对测量系统（至少在编码器旋转一圈的范围内）。

关闭时，编码器的变化不允许超过计数范围的一半，这样才能成功恢复位置。对于单圈编码器，可使用电机保持制动器来确保这一点。如果在断电期间轴可能会停转，则必须使用多圈编码器。

使用虚拟轴时，将恢复重启或断电前存储的最后一个位置。

这种归位模式适用于无端轴（例如包装机和转台）。不过，它也可用于运动范围有限的轴（如搬运系统）。该模式还可用于处理有限轴中间的多圈编码器溢出（"mcHOME_ABSOLUTE_COURR"归位模式的替代模式）。

Information:

All other homing variants correspond to a calibration of the axis; the data in remanent memory is reinitialized.

信息:

所有其他归位变体都与轴的校准相对应；重置存储器中的数据将被重新初始化。

Information:

After calibrating the axis and before restoring the axis position, the following parameters are not permitted to be changed. Otherwise, the position is restored incorrectly.

-
- [Axis period](#)
- [Axis factor](#)
- [encoder_if.parameter](#)

信息:

在校准轴之后和恢复轴位置之前，不允许更改以下参数。否则，将错误地恢复位置。

-
- [轴周期](#)
- [轴系数](#)
- [编码器参数](#)

Danger!

An incorrect position will be calculated if the axis is moved too far when turned off! The function block is unable to detect this problem, and an error message will not be output! The maximum distance that the axis is permitted to be moved when the power is switched off can be read using function block [MC_ReadParameter](#) (parameter: [mcMAX_POSITION_CHANGE](#)).

危险!

如果轴在关闭状态下移动过远，将计算出错误的位置！功能块无法检测该问题，也不会输出错误信息！可以使用功能块 [MC_ReadParameter](#)（参数：[mcMAX_POSITION_CHANGE](#)）读取断电时允许轴移动的最大距离。

Limitations 限制

The modulo position is calculated in the ACP10MAN task class. It is important to make sure that this task class is called twice per load period! It should also be noted that the network cycle is not permitted to be longer than this task class time.

模数位置在 ACP10MAN 任务类中计算。必须确保该任务类在每个负载周期内被调用两次！还应注意的是，网络周期不得长于该任务类的时间。

Example 例子

Load period = 1000 PLCopen units

Max. speed = 50,000 PLCopen units/s

Max. cycle time of task class = Load period / Max. speed / 2

=> Max. cycle time of task class = 10 ms

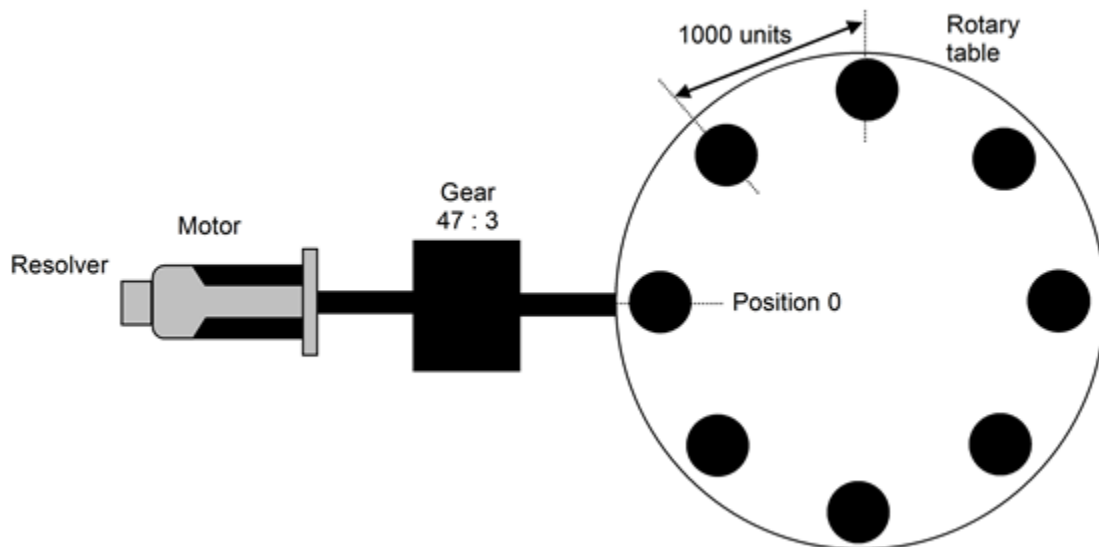
Topics in this section:

- [Example illustrating the procedure for calibrating and restoring the position](#)

1.1.2.1.1. Example illustrating the procedure for calibrating and restoring the position

A rotary table containing 8 product holders is being controlled. The space between the product holders (= load period) is equal to 1000 units. A gear with a gear ratio of 47:3 is located between the motor and rotary table. The motor has a measurement system that captures motor revolutions absolutely. Due to the high gear ratio, the load cannot be moved when turned off.

控制一个包含 8 个产品托架的旋转台。产品支架之间的空间 (= 负载周期) 等于 1000 个单位。电机和旋转台之间有一个齿轮，齿轮比为 47:3。电机有一个测量系统，可绝对捕捉电机的转数。由于齿轮比高，负载在关闭时无法移动。



• Parameter settings 参数设置

INIT parameter module:

```
encoder_if.parameter.scaling.load.units = 24000(1000 * 8 * 3)
```

```
encoder_if.parameter.scaling.load.rev_motor = 47
```

NC mapping table:

In "Additional Data" ... PLCopen_ModPos="1000,1"

The [MC_BR_InitModPos](#) function block can also be used to change the period and factor instead of doing so with the mapping table or after starting the controller.

(This must be done before calling [MC_Home](#), however.)

也可以使用 [MC_BR_InitModPos](#) 功能块来更改周期和系数，而不是使用映射表或在启动控制器后更改。(不过，这必须在调用 [MC_Home](#) 之前完成)。

- **One-time reading of the position (calibration during startup) 一次性读取位置（启动时校准）**

The load position is measured or the rotary table is set to the 0 position and the axis is homed to this position or to 0 using homing mode "[mcHOME DIRECT](#)".
测量负载位置或将旋转台设置到 0 位置，然后使用归位模式"[mcHOME DIRECT](#)"将轴归位到该位置或 0 位置。

- **The position changes 位置变化**

The position changes between 0 and 999 within the period once the axis is in motion ([MC_ReadActualPosition](#)).
一旦轴处于运动状态 ([MC_ReadActualPosition](#))，位置就会在 0 到 999 之间变化。

- **Restoring the position 恢复位置**

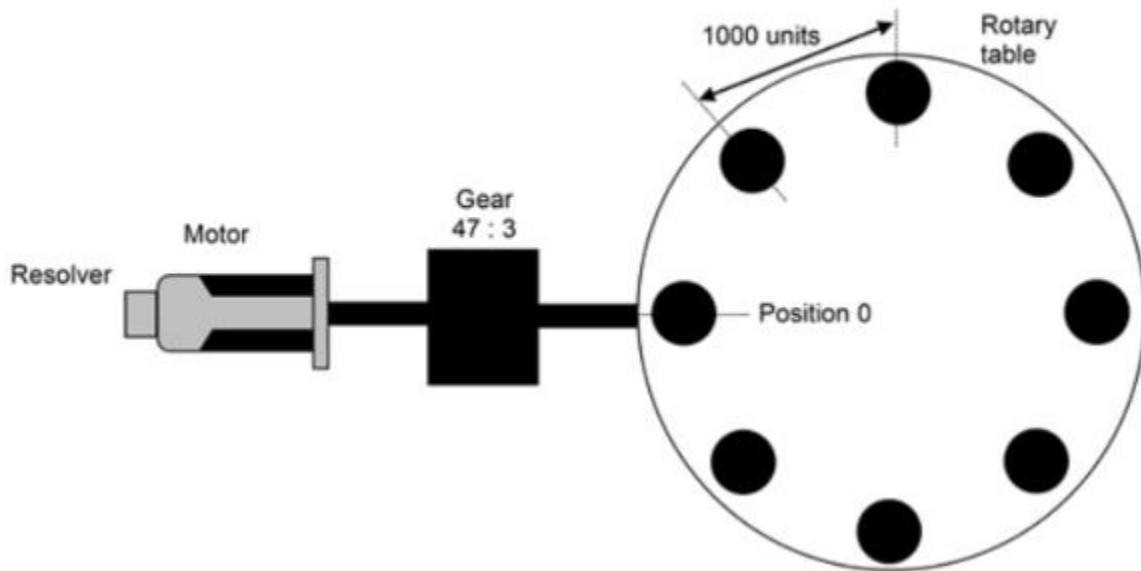
The controller is switched after a warm or cold restart of the target system ([MC_Power](#)). [MC_Home](#) is now called with mode "[mcHOME RESTORE POS](#)". If [MC_Home](#) sets output "Done" to TRUE, the position is restored. [MC_ReadActualPosition](#) then indicates the correct position within the period, and the axis can be moved using any function blocks.

目标系统 ([MC_Power](#)) 在热重启或冷重启后会切换控制器。现在以 "[mcHOME RESTORE POS](#)" 模式调用 [MC_Home](#)。如果 [MC_Home](#) 将输出 "Done" 设置为 "true"，则位置已恢复。然后，[MC_ReadActualPosition](#) ([实际位置读取](#)) 将显示该时间段内的正确位置，并可使用任何功能块移动轴。

1.2. mcHOME_RESTORE 寻参配置案例

问题

机器每个轴都是周期轴 0-360，朝单一方向连续运转，采用多圈绝对值编码器，且减速比不是 2 的倍数，当超过 4096 圈后，机器重启后会出现轴的位置丢失的情况。



解决方式

寻参方式采用 mcHOME_RESTORE_POS，使用功能块 MC_BR_InitEndlessPosition 和 MC_BR_CheckEndlessPosition
功能块中 MC_ENDLESS_POSITION 定义为永久变量，使用
MC_BR_CheckEndlessPosition 判断保存的数据是否有效，有效采用
mcHOME_RESTORE_POS 寻参方式

1.3. 功能演示效果与测试案例

测试案例

伺服上电使能后，点击零点标定，确定当前位置为零点，然后让电机运动一段距离，记下该位置值，然后 PLC 和伺服都断电重启，因为绝对值编码器有位置（HW Postion）断电记忆功能（编码器位置和程序中实际位置不一定相等），所以程序中会有一个位置值（SW Postion），但是这个位置和断电器的位置不相等，需要点击 Home 后，如果位置变为断电前位置，则说明测试成功。

测试结果

使用该模式测试，不管运动距离多少，断电上电后均正常

注意事项

上位用于显示实际的位置的变量

```
gAxis[AxIdx].Status.ActAxisPosition := gAxis[AxIdx].Status.ActPosition - gRcpFix.HomerOffset[AxIdx];
```

使用该模式标定时需注意当前得到的 offset 值大小，如果 offset 的整数部分值太大，如 offset = 20000000.831，则可能在走绝对定位时会出现误差，因为 offset 值是 REAL 型，整数部分过大时，影响小数部分精度，比如当前位置标定是 0 后，绝对定位到 32.563mm 处，则实际需要给定的定位是 MC_MoveAbsolute.Position = 32.563mm + 20000000.831，即 20000033.394mm。因为 REAL 型精度的问题，实际可能绝对定位到 20000033.3929mm 就认为到位，电机就停止运动了，因此当发现 offset 值很大时，可以先使用 mcHOME_DIRECT 模式将当前的位置标定为 0，然后重新进行零点的标定，因此该模式的 Home 使用 DIRECT 模式，Offset 为 0。

```
(* Start homing process
STATE HOME:
    MC_Home_0[AxIdx].Position := 0; //gAxis[AxIdx].Parameter.Position;
    MC_Home_0[AxIdx].HomingMode := mcHOME_DIRECT; //gAxis[AxIdx].Parameter.HomingMode;
    MC_Home_0[AxIdx].Execute := TRUE;
```

且绝对运动时，运动距离需要加上标定的 Offset。

```
STATE MOVE_ABSOLUTE:
    MC_MoveAbsolute_0[AxIdx].Direction := 0;
    MC_MoveAbsolute_0[AxIdx].Position := gAxis[AxIdx].Parameter.Position + gRcpFix.HomerOffset[AxIdx];
    MC_MoveAbsolute_0[AxIdx].Velocity := gAxis[AxIdx].Parameter.Velocity;
```


1.4. 使用 mcHOME_RESTORE 命令需注意调用顺序

关于使用 mcHOME_RESTORE_POS 功能在 monitor.s 停留在超出 (+DINT)范围的保存相位的注意事项

设置说明

轴设置一览:

encoder_if			Encoder Interface
parameter			Parameters
count_dir	ncSTANDARD		Count direction
scaling			Scaling
load			Load
units	1000000	Units	Units at the load
rev_motor	1		Motor revolutions
limit			Limit value
parameter			Parameters
v_pos	1.0E+07	Units/s	Speed in positive direction
v_neg	1.0E+07	Units/s	Speed in negative direction
a1_pos	3000000.0	Units/s?	Acceleration in positive direction
a2_pos	3000000.0	Units/s?	Deceleration in positive direction
a1_neg	3000000.0	Units/s?	Acceleration in negative direction
a2_neg	3000000.0	Units/s?	Deceleration in negative direction
t_jolt	0.0	s	Jolt time
t_in_pos	0.0	s	Settling time before message 'In Position'
pos_sw_end	8388607	Units	Positive SW end
neg_sw_end	-8388608	Units	Negative SW end
ds_waming	500.0	Units	Lag error limit for display of a waming
ds_stop	10000.0	Units	Lag error limit for stop of a movement
a_stop	1.0E+30	Units/s?	Acceleration limit for stop of a movement
dv_stop	0.0	1/s	Speed error limit for stop of a movement
dv_stop_mode	ncOFF		Mode for speed error monitoring
us_stop	0.0	Units	Lag error for stop of a movement
basis			Basis movements
parameter			Parameters
v_pos	9000000.0	Units/s	Speed in positive direction
v_neg	9000000.0	Units/s	Speed in negative direction
a1_pos	2000000.0	Units/s?	Acceleration in positive direction
a2_pos	2000000.0	Units/s?	Deceleration in positive direction
a1_neg	2000000.0	Units/s?	Acceleration in negative direction
a2_neg	2000000.0	Units/s?	Deceleration in negative direction
message			Messages (errors, wamings)

参数表设置:



Name	ID	Value	Unit	Description
Parameters				
SGEN_SW_END_IGNORE	128	3		Limit values: Ignore SW end positions

周期设置:

STATE_INIT_MODPOS:

```
MC_BR_InitModPos_0.Period := 866600;
MC_BR_InitModPos_0.Factor := 1;
MC_BR_InitModPos_0.Execute:= TRUE;
```

第一个测试（mcHOME_RESTORE_POS 标准使用方法）:

标准使用方法是:

先调用 MC_BR_InitModPos 功能块，再进入 mcHOME，
HomingMode=mcHOME_RESTORE_POS 模式

1. 先启用周期轴相关参数，之后进行 mcHOME，HomingMode=mcHOME_DIRECT
(1) 模式，在之后进入 moveVelocity 恒定速度运行状态，片刻后将 monitor.s 停留在超出 (+DINT)范围后的某位置:

Parameter	value	Unit	Description
network.init	ncTRUE		Network initialized
network.service.data_text			Data in text format
network.service.request.par_id	178; STATUS_...		Parameter ID
network.service.response.par_id	178		Parameter ID
monitor.s	-2083043131	Units	Position
monitor.v	0	Units/s	Velocity
message.count.error	0		Count of not acknowledged errors
message.count.waming	0		Count of not acknowledged warnings
message.record.par_id	0		Parameter ID
message.record.number	0		Error number
monitor.status.error	ncFALSE		Error occurred
monitor.status.waming	ncFALSE		Waming occurred
message.record.info	0		Additional Info
limit.parameter.v_pos	1e+007	Units/s	Speed in positive direction

MC_Home_0	MC_Home	local		
Execute	BOOL			FALSE
Position	REAL			0.0
HomingMode	USINT			1
Done	BOOL			FALSE
MC_BR_InitEndlessPosition_0	MC_BR_InitEndl	local		
Execute	BOOL			FALSE
Done	BOOL			TRUE
MC_Power_0	MC_Power	local		
Enable	BOOL			TRUE
MC_Stop_0	MC_Stop	local		
Execute	BOOL			FALSE
BasicControl	basic_typ	local		
MC_MoveVelocity_0	MC_MoveVeloci	local		
Execute	BOOL			FALSE
Velocity	REAL			9000000.0
gAxEndlessPos	MC_ENDLESS_	local		
EndlessPositionData	MC_ENDLESS_			
EndlessPositionData[0]	MC_ENDLESS_			
MTPhase	DINT			65136
MTDiffInteger	DINT			436800
MTDiffFract	DINT			0
RefOffset	DINT			140951
Checksum	UDINT			2528865944
MC_ReadActualPosition_0	MC_ReadActual	local		
Position	REAL			360965.0
STATE_INIT_MODPOS_CMD	BOOL	local		FALSE

断电重启后未做任何处理的数据:

可以看到数据是乱的

MC_Home_0	MC_Home	local		
Execute	BOOL			FALSE
Position	REAL			0.0
HomingMode	USINT			0
Done	BOOL			FALSE
MC_BR_InitEndlessPosition_0	MC_BR_InitEndl	local		
Execute	BOOL			FALSE
Done	BOOL			TRUE
MC_Power_0	MC_Power	local		
Enable	BOOL			FALSE
MC_Stop_0	MC_Stop	local		
Execute	BOOL			FALSE
BasicControl	basic_typ	local		
MC_MoveVelocity_0	MC_MoveVeloci	local		
Execute	BOOL			FALSE
Velocity	REAL			0.0
gAxEndlessPos	MC_ENDLESS_	local		
EndlessPositionData	MC_ENDLESS_			
EndlessPositionData[0]	MC_ENDLESS_			
MTPhase	DINT			65136
MTDiffInteger	DINT			436800
MTDiffFract	DINT			0
RefOffset	DINT			140951
Checksum	UDINT			2528865944
MC_ReadActualPosition_0	MC_ReadActual	local		
Position	REAL			64556.0
STATE_INIT_MODPOS_CMD	BOOL	local		FALSE

Parameter	Value	Unit	Description
network.init	ncTRUE		Network initialized
network.service.data_text			Data in text format
network.service.request.par_id	798; AUT_MS...		Parameter ID
network.service.response.par_id	798		Parameter ID
monitor.s	64555	Units	Position
monitor.v	1192.09	Units/s	Velocity
message.count.error	0		Count of not acknowledged errors
message.count.waming	0		Count of not acknowledged waming
message.record.par_id	0		Parameter ID
message.record.number	0		Error number
monitor.status.error	ncFALSE		Error occurred
monitor.status.waming	ncFALSE		Waming occurred
message.record.info	0		Additional Info
limit.parameter.v_pos	1e+007	Units/s	Speed in positive direction

下面是 mcHOME 中的 mcHOME_RESTORE_POS 进行相位恢复的标准用法之后得到的数据：

MC_Home_0	MC_Home	local	
Execute	BOOL		FALSE
Position	REAL		0.0
HomingMode	USINT		11
Done	BOOL		FALSE
MC_BR_InitEndlessPosition_0	MC_BR_InitEndl	local	
Execute	BOOL		FALSE
Done	BOOL		TRUE
MC_Power_0	MC_Power	local	
Enable	BOOL		TRUE
MC_Stop_0	MC_Stop	local	
Execute	BOOL		FALSE
BasicControl	basic_typ	local	
MC_MoveVelocity_0	MC_MoveVeloci	local	
Execute	BOOL		FALSE
Velocity	REAL		0.0
gAxEndlessPos	MC_ENDLESS_	local	
EndlessPositionData	MC_ENDLESS_		
EndlessPositionData[0]	MC_ENDLESS_		
MTPPhase	DINT		64539
MTDiffInteger	DINT		436800
MTDiffFract	DINT		0
RefOffset	DINT		140951
Checksum	UDINT		1571256627
MC_ReadActualPosition_0	MC_ReadActual	local	
Position	REAL		360406.0
STATE_INIT_MODPOS_CMD	BOOL	local	FALSE

Parameter	Value	Unit	Description
network.init	ncTRUE		Network initialized
network.service.data_text			Data in text format
network.service.request.par_id	798; AUT_MS...		Parameter ID
network.service.response.par_id	798		Parameter ID
monitor.s	360406	Units	Position
monitor.v	0	Units/s	Velocity
message.count.error	0		Count of not acknowledged errors
message.count.waming	0		Count of not acknowledged warnings
message.record.par_id	0		Parameter ID
message.record.number	0		Error number
monitor.status.error	ncFALSE		Error occurred
monitor.status.waming	ncFALSE		Warning occurred
message.record.info	0		Additional Info
limit.parameter.v_pos	1e+007	Units/s	Speed in positive direction

可以看到按照此方法将停电前的位置 360965 成功的保存成当前位置 360406，说明相位已经恢复好（因为一圈 100 万，有些许误差，正常范围内）。

第二个测试（mcHOME_RESTORE_POS 错误使用方法）

错误使用方法：先进行 mcHOME，HomingMode=mcHOME_RESTORE_POS 模式，再调用 MC_BR_InitModPos 功能块。

先把周期轴相关参数设定好，之后进行 mcHOME，HomingMode=mcHOME_DIRECT

(1) 模式，在之后进入 moveVelocity 恒定速度运行状态，片刻后将 monitor.s 停留在超出 (+DINT)范围后的某位置：

MC_Home_0	MC_Home	local		
Execute	BOOL			FALSE
Position	REAL			0.0
HomingMode	USINT			1
Done	BOOL			FALSE
MC_BR_InitEndlessPosition_0	MC_BR_InitEndl	local		
Execute	BOOL			FALSE
Done	BOOL			TRUE
MC_Power_0	MC_Power	local		
Enable	BOOL			TRUE
MC_Stop_0	MC_Stop	local		
Execute	BOOL			FALSE
BasicControl	basic_typ	local		
Command	basic_command			
Parameter	basic_parameter			
Status	basic_status_typ			
ErrorID	UINT			0
ErrorText	STRING[79][0..:			
ActPosition	REAL			392569.0
ActVelocity	REAL			0.0
MC_MoveVelocity_0	MC_MoveVeloci	local		
Execute	BOOL			FALSE
Velocity	REAL			9000000.0
gAxEndlessPos	MC_ENDLESS_	local		
EndlessPositionData	MC_ENDLESS_			
EndlessPositionData[0]	MC_ENDLESS_			
MTPhase	DINT			488707
MTDiffInteger	DINT			835000
MTDiffFract	DINT			0
RefOffset	DINT			64557
Checksum	UDINT			2267164679
MC_ReadActualPosition_0	MC_ReadActual	local		
Position	REAL			392569.0
STATE_INIT_MODPOS_CMD	BOOL	local		FALSE

network.init	ncTRUE		Network initialized
network.service.data_text			Data in text format
network.service.request.par_id	178; STATUS_...		Parameter ID
network.service.response.par_id	178		Parameter ID
monitor.s	-2105543127	Units	Position
monitor.v	0	Units/s	Velocity
message.count.error	0		Count of not acknowledged errors
message.count.waming	0		Count of not acknowledged wamings
message.record.par_id	0		Parameter ID
message.record.number	0		Error number
monitor.status.error	ncFALSE		Error occurred
monitor.status.waming	ncFALSE		Waming occurred
message.record.info	0		Additional Info
limit.parameter.v_pos	1e+007	Units/s	Speed in positive direction

断电重启未做任何处理数据:

[-]  MC_Home_0	MC_Home	local		
[-]  Execute	BOOL			FALSE
[-]  Position	REAL			0.0
[-]  HomingMode	USINT			0
[-]  Done	BOOL			FALSE
[-]  MC_BR_InitEndlessPosition_0	MC_BR_InitEndl	local		
[-]  Execute	BOOL			FALSE
[-]  Done	BOOL			TRUE
[-]  MC_Power_0	MC_Power	local		
[-]  Enable	BOOL			TRUE
[-]  MC_Stop_0	MC_Stop	local		
[-]  Execute	BOOL			FALSE
[-]  BasicControl	basic_typ	local		
[-]  Command	basic_command			
[-]  Parameter	basic_parameter			
[-]  Status	basic_status_typ			
[-]  ErrorID	UINT			0
[-]  ErrorText	STRING[79][0..3			
[-]  ActPosition	REAL			488816.0
[-]  ActVelocity	REAL			0.0
[-]  MC_MoveVelocity_0	MC_MoveVeloci	local		
[-]  Execute	BOOL			FALSE
[-]  Velocity	REAL			0.0
[-]  gAxEndlessPos	MC_ENDLESS_	local		
[-]  EndlessPositionData	MC_ENDLESS_			
[-]  EndlessPositionData[0]	MC_ENDLESS_			
[-]  MTPhase	DINT			488834
[-]  MTDiffInteger	DINT			835000
[-]  MTDiffFract	DINT			0
[-]  RefOffset	DINT			64557
[-]  Checksum	UDINT			1974946914
[-]  MC_ReadActualPosition_0	MC_ReadActual	local		
[-]  Position	REAL			488816.0
 STATE_INIT_MODPOS_CMD	BOOL	local		FALSE

Parameter	value	Unit	Description
◆ network.init	ncTRUE		Network initialized
◆ network.service.data_text			Data in text format
◆ network.service.request.par_id	798; AUT_MS...		Parameter ID
◆ network.service.response.par_id	798		Parameter ID
◆ monitor.s	488816	Units	Position
◆ monitor.v	0	Units/s	Velocity
◆ message.count.error	0		Count of not acknowledged errors
◆ message.count.waming	0		Count of not acknowledged wamings
◆ message.record.par_id	0		Parameter ID
◆ message.record.number	0		Error number
◆ monitor.status.error	ncFALSE		Error occurred
◆ monitor.status.waming	ncFALSE		Waming occurred
◆ message.record.info	0		Additional Info
◆ limit.parameter.v_pos	1e+007	Units/s	Speed in positive direction

下面进行 mcHOME，HomingMode=mcHOME_RESTORE_POS 模式再调用 MC_BR_InitModPos 功能块;数据如下：

MC_Home_0	MC_Home	local	
Execute	BOOL		FALSE
Position	REAL		0.0
HomingMode	USINT		11
Done	BOOL		FALSE
MC_BR_InitEndlessPosition_0	MC_BR_InitEndl	local	
Execute	BOOL		FALSE
Done	BOOL		TRUE
MC_Power_0	MC_Power	local	
Enable	BOOL		TRUE
MC_Stop_0	MC_Stop	local	
Execute	BOOL		FALSE
BasicControl	basic_typ	local	
Command	basic_command		
Parameter	basic_parameter		
Status	basic_status_typ		
ErrorID	UINT		0
ErrorText	STRING[79][0..1		
ActPosition	REAL		553373.0
ActVelocity	REAL		0.0
MC_MoveVelocity_0	MC_MoveVeloci	local	
Execute	BOOL		FALSE
Velocity	REAL		0.0
gAxEndlessPos	MC_ENDLESS_	local	
EndlessPositionData	MC_ENDLESS_		
EndlessPositionData[0]	MC_ENDLESS_		
MTPhase	DINT		488871
MTDiffInteger	DINT		835000
MTDiffFract	DINT		0
RefOffset	DINT		64557
Checksum	UDINT		416410121
MC_ReadActualPosition_0	MC_ReadActual	local	
Position	REAL		553373.0
STATE_INIT_MODPOS_CMD	BOOL	local	FALSE

Parameter	Value	Unit	Description
◆ network.init	ncTRUE		Network initialized
◆ network.service.data_text			Data in text format
◆ network.service.request.par_id	798; AUT_MS...		Parameter ID
◆ network.service.response.par_id	798		Parameter ID
◆ monitor.s	553373	Units	Position
◆ monitor.v	0	Units/s	Velocity
◆ message.count.error	0		Count of not acknowledged errors
◆ message.count.waming	0		Count of not acknowledged wamings
◆ message.record.par_id	0		Parameter ID
◆ message.record.number	0		Error number
◆ monitor.status.error	ncFALSE		Error occurred
◆ monitor.status.waming	ncFALSE		Waming occurred
◆ message.record.info	0		Additional Info
◆ limit.parameter.v_pos	1e+007	Units/s	Speed in positive direction

上述结果说明，这个相位未恢复成功。

接着再进行一次 mcHOME，HomingMode=mcHOME_RESTORE_POS 命令。可以看到相位恢复成功了。

MC_Home_0	MC_Home	local	
Execute	BOOL		TRUE
Position	REAL		0.0
HomingMode	USINT		11
Done	BOOL		TRUE
MC_BR_InitEndlessPosition_0	MC_BR_InitEndl	local	
Execute	BOOL		FALSE
Done	BOOL		TRUE
MC_Power_0	MC_Power	local	
Enable	BOOL		TRUE
MC_Stop_0	MC_Stop	local	
Execute	BOOL		FALSE
BasicControl	basic_typ	local	
Command	basic_command		
Parameter	basic_parameter		
Status	basic_status_typ		
ErrorID	UINT		0
ErrorText	STRING[79][0..3		
ActPosition	REAL		392659.0
ActVelocity	REAL		0.0
MC_MoveVelocity_0	MC_MoveVeloci	local	
Execute	BOOL		FALSE
Velocity	REAL		0.0
gAxEndlessPos	MC_ENDLESS_	local	
EndlessPositionData	MC_ENDLESS_		
EndlessPositionData[0]	MC_ENDLESS_		
MTPhase	DINT		488871
MTDiffInteger	DINT		835000
MTDiffFract	DINT		0
RefOffset	DINT		64557
Checksum	UDINT		416410121
MC_ReadActualPosition_0	MC_ReadActual	local	
Position	REAL		392659.0
STATE_INIT_MODPOS_CMD	BOOL	local	FALSE



Parameter	Value	Unit	Description
◆ network.init	ncTRUE		Network initialized
◆ network.service.data_text			Data in text format
◆ network.service.request.par_id	798; AUT_MS...		Parameter ID
◆ network.service.response.par_id	798		Parameter ID
◆ monitor.s	392659	Units	Position
◆ monitor.v	0	Units/s	Velocity
◆ message.count.error	0		Count of not acknowledged errors
◆ message.count.waming	0		Count of not acknowledged warnings
◆ message.record.par_id	0		Parameter ID
◆ message.record.number	0		Error number
◆ monitor.status.error	ncFALSE		Error occurred
◆ monitor.status.waming	ncFALSE		Waming occurred
◆ message.record.info	0		Additional Info
◆ limit.parameter.v_pos	1e+007	Units/s	Speed in positive direction

上述结果标表明先进行 mcHOME，HomingMode=mcHOME_RESTORE_POS 命令再调用 MC_BR_InitModPos 功能块是无法恢复相位的。请注意。

1.5. 可能存在的问题与注意事项

mcHOME_RESTORE 依赖 PLC 的永久保持变量，若因为各种原因，发现永久保持变量存储数据有丢失风险（例如可能不会及时更换电池，或当前硬件组合下，永久保持变量偶发会丢失），建议换用 mcHOMING_ABSOLUTE_CORRECTION。

现象

对于在 ACP10 或 mapp Motion 中使用 "原点复位 mcHOME_RESTORE" 功能的客户，如果经常丢失永久数据，则必须重新执行归位操作。

InitEndlessPosition 的问题在于，当非易失性内存丢失时，位置就会丢失，必须重新执行归位程序，这在某些过程中很难做到。

为了避免这种情况，可以将 MC_ENDLESS_POSITION 保存在一个文件中。启动后，如果永久变量为空，可以在调用归位恢复模式之前将文件中的数据写入 MC_ENDLESS_POSITION 变量。

根据我们的开发，可以将轴的整个永久结构（包括校验和）备份到文件或数据对象中。重要的是要复制整个内存。使用 mapp Motion 时，它也包含校验和。不过，循环保存到数据对象或文件中并不可靠。

建议

基本上可以将 mcHOMING_RESTORE_POSITION 的恢复位置 PV 数据保存在速度较慢的任务类中。

重要的是，至少每半个编码器计数范围保存一次数据。例如：如果使用的编码器多圈转数为 4096 圈，则至少每 2048 圈编码器转数保存一次数据。但也可以预留一些时间（例如在上述例子中每 1024 或 512 转更新一次）。

不过，将数据保存在闪存中可能会导致闪存存在短时间内损坏。由于更新率取决于轴的速度，客户/应用工程师必须检查最大写入周期的速度。但是，我们运动控制方面无法确定每一条写入命令是否都会导致闪存的写入，或者数据是否在 AR 端进行了 "缓冲"。

最佳做法是在项目安装前不久备份数据，因为循环保存并不是重点，但在重启后必须只提供 "最新" 数据，以便恢复轴的位置。

mappMotion 库方面没有关于存储介质的建议。

数据加固 (Data hardening)

一般来说，该功能并非来自运动方面。如果启用了数据硬化功能，启动和关闭时的初始状态也会保持不变。只有在发生错误时，数据加固功能才会发挥作用。

如果还原位置 PV 和 mcHOMING_RESTORE_POSITION 归位出现问题，建议在传输后检查运动端使用的还原位置变量数据，并与传输前的数据进行比较，检查是否发生变化，以及是否需要从文件备份中还原（正确的）数据。

替代方案

单圈 mcHOME_ABSOLUTE

多圈 mcHOME_ABSOLUTE_CORR

mcHOMING_ABSOLUTE_CORRECTION，并将归位偏移保存到配方文件中。这与 mcHOMING_RESTORE_POS 不同，但可以与我们的轴配合使用，而且是一种更稳健的解决方案。