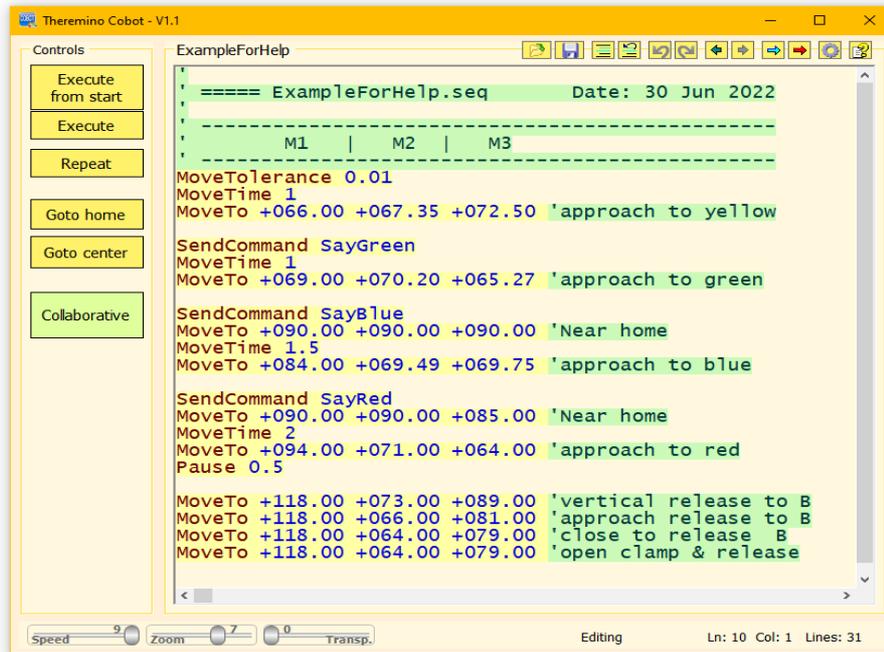


theremino
•the•real•modular•in-out•

系统特雷米诺

Theremino 协作机器人 V1.2

Theremino 协作机器人应用程序



此应用程序包含移动协作机器人所必需的，即机械臂或旨在与人类协作的任何其他机制。

可以控制以任何配置布置的带有一个或多个电机的设备，并且没有必要指定机构的尺寸特性。

我们自愿消除了三维描述及其带来的所有复杂性，我们直接指定每个引擎的位置，以度、毫米或您喜欢的任何其他测量单位。

三维计算总是不准确的，因为您必须设置所有导致错误的参数。但它们是难以测量的参数，并且会根据运输的旋转和重量而变化。

然后，我们将计算委托给模拟计算机，即机械本身，以绝对精确的方式执行计算。考虑到每一个可能的因素，从所涉及的力到重力，到故障，再到结构误差，甚至是齿轮、电机及其反馈回路引起的误差。

这些简化极大地促进了应用程序的使用，但在某些情况下，希望获得更多的线性运动或更好地控制操作。在这种情况下阅读[先进技术](#)在本文档的末尾。

协作机器人是如何制造的

协作机器人旨在在不危及人类的情况下协同工作，因此它们具有与工业机器人不同的结构。

Theremino Cobot 应用程序可以控制任意数量的电机，以它们排列的任何机械配置。可以控制单独的电机，例如用于切割流动产品的跑步机和断头台。或者你可以控制 SCARA 配置的手臂或 DELTA 型机器人，但通常你控制的是拟人化的 Cobot，这是最灵活的，因此可用于中小型行业的许多不同任务。

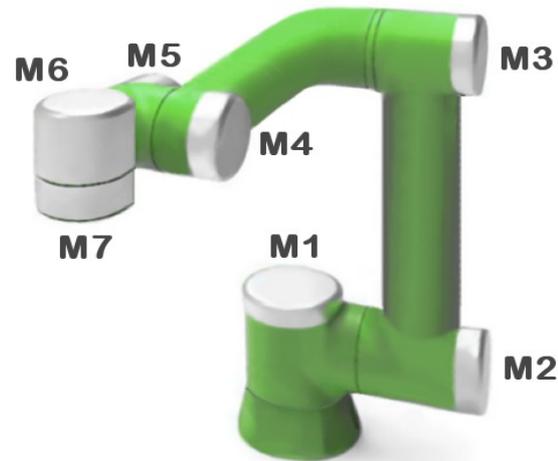
这是一个拟人化协作机器人手臂的示例。

这张图片只是一个方案，形状也可以不同，但主要成分总是一样的，比例也会或多或少这些。

每个银色圆盘代表一个旋转接头，因此也是一个电机。

让我们给每个部分一个有意义的名字：

- **M1 基础版**在水平面上旋转。它也可以 360 度旋转，但通常只能向前工作并且有侧挡。它不需要太多扭矩，但需要精确（反冲小），因为它的误差会乘以整个臂长。
- **M2 肩**举起你的整个手臂。这是必须具有最大扭矩并且还必须非常精确的发动机。
- **M3 弯头**它完成了肩部的工作，让您到达每个位置。该引擎必须平均准确。



关节 **M4,M5,M6** 和 **M7** 我在那里但不是

- **M4 和 M5 手腕**这些关节可以上下左右旋转，它们不需要很精确或有很大的扭矩，但它们确实需要小而轻。
- **M6 旋转**这个关节可以提供连续的旋转，例如控制螺丝刀。在这种情况下，M7 将不存在。
- **M7 夹具**这不是旋转接头，而是用于夹紧、拧紧、焊接、喷漆等的工具.....

设计安全的协作机器人

协作机器人必须能够在不构成危险的情况下与人类并肩工作。它必须具有圆形表面，并且在所有可能的旋转中，其部件不得彼此靠得太近，否则它们可能会像胡桃夹子一样夹住皮肤或手指。



协作机器人的安全性必须是内在的。

机械师本身不必处理
能够造成伤害的力量和速度。

这是一个简单的原理，容易理解，
没有软件或电子保护。

没有可能出错的规则或协议
没有可以破坏的机制。

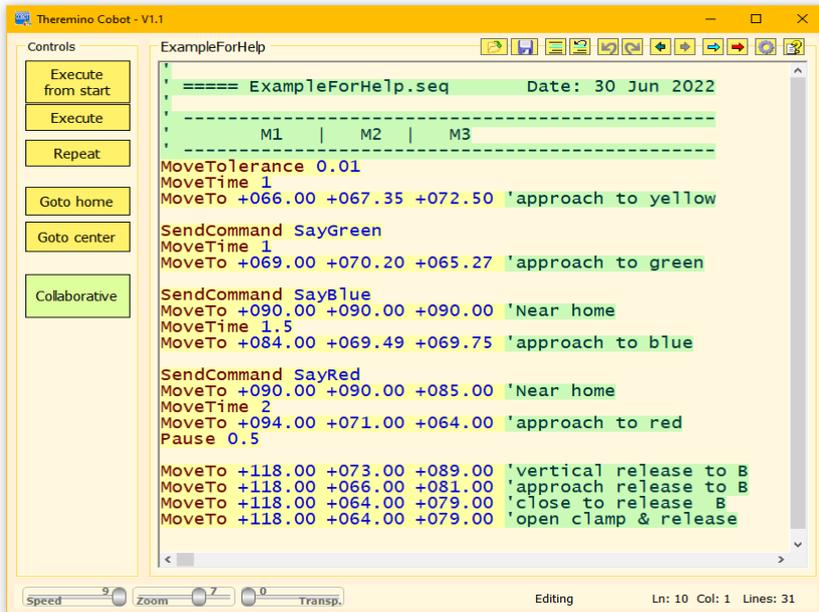
协作机器人的安全性必须是内在的，而不是基于认证、电子系统或软件，因为软件可能出错，电子设备可能损坏，认证可能不完整或被误解。

基于软件、协议和认证的安全机制，除了不够充分之外，还会增加系统的危险性，因为它们对人类产生了错误的信任，从而导致人们盲目信任和冒险。

手表 [这个视频](#) 关于一个打破孩子手指的机器人。

并阅读我们的文档 **Theremino 协作机器人安全**
以英文、意大利文和中文出版。

使用协作机器人应用程序



正如我们将在后面几页中看到的那样，此应用程序可以通过多种方式进行控制，但其基本操作很简单。

您在列表中编写命令行并使用此页面上的命令执行它们。

您通过观察协作机器人的移动来更改列表中的值。

或者您是从另一个应用程序发送命令（通常是 Theremino_Automation）。

以下是主要命令：



紧迫从头开始执行从头开始执行。

和执行执行从光标所在的行开始，即用鼠标或键盘选择的行。

如果启用按钮重复然后在执行完最后一行之后，执行从第一行重新开始。

和回家你让协作机器人移动到模式中的预选位置合作的。

和转到中心使协作机器人移动到模式中预先选择的位置合作的。

和合作的您可以访问我们将在下一页看到的协作模式。

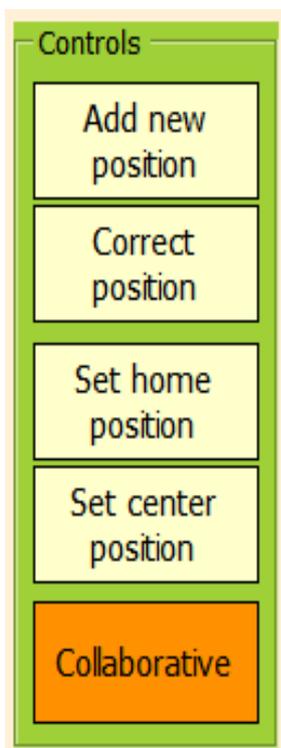
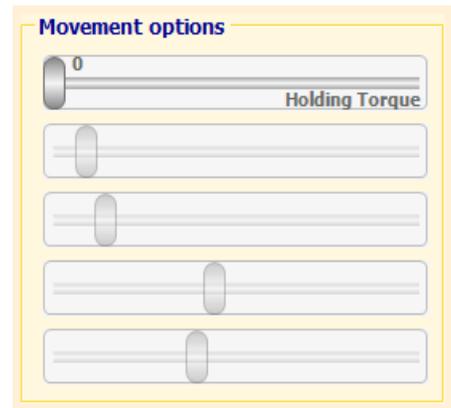
要停止执行，请再次按下执行或单击命令列表中的一行。

“合作”方式

当您进入协作模式时，光标出现在选项窗口中保持力矩用于限制扭矩。

移动保持力矩向左，扭矩受到很大限制，可以手动旋转协作机器人的关节。

此游标和其他四个游标的详细信息将在以下页面中进行说明。



紧迫添加新职位一行被添加到序列中搬去所有引擎的当前位置。

和正确位置线被改变了搬去与电机的当前位置一起选择。

和设置起始位置位置已设置家与电机的当前位置 [注 1](#)

和设置中心位置位置已设置中央与电机的当前位置。

按下按钮合作的您退出协作模式并返回正常操作模式。

协作模式仅用于在进行更改后进行更改
按下“协作”按钮并返回正常操作模式。

注 1- 如果电机超过 360 度使用，则起始位置必须在第一圈（对于 FeeTech 从 0 到 4095）并且在关闭手臂之前，它应该回到起始位置。

编辑命令序列

在正常操作期间，无需进入协作模式，您可以编辑命令的序列与键盘并与鼠。如果序列正在运行，只要您将鼠标光标放在命令列表上或使用向上和向下箭头，它就会自动停止。

右键单击命令会打开一个菜单，我们将在本文档的最后几页中详细介绍。

我们将在接下来的页面中看到的有效命令以黄色背景色突出显示，而错误则以红色突出显示。

添加地点

您可以通过手动键入命令来添加位置搬去或通过复制和粘贴其他现成的线条。

您还可以使用协作模式添加位置，甚至可以在手动将协作机器人移动到所需位置时使用外部按钮。

逐行运行命令

您可以通过使用双击鼠标在该行的第一个单词上。

如果选择了一行（蓝色背景），那么您可以使用键盘上的向上和向下箭头来执行上一行和下一行。

用鼠标改变位置

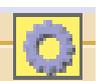
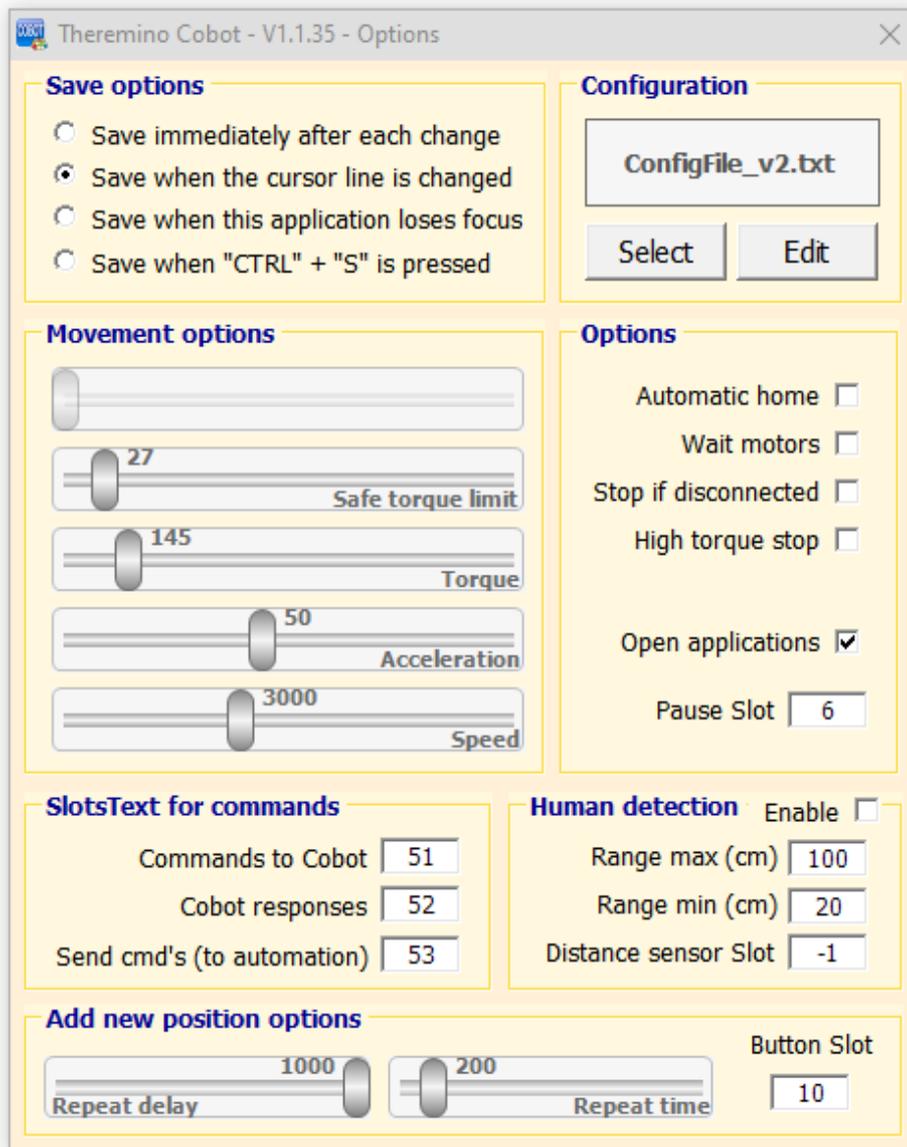
行旋转值可以改变搬去并在进行更改时查看协作机器人的实际运动。

- 与双击选择 MoveTo 行的数值。
- 只要选择了数值，鼠标滚轮就会对其进行修改，您可以通过查看协作机器人立即看到移动，而无需注意鼠标光标的位置。

数值的变化速度可以通过按键修改 SHIFT, CTRL 键和 ALT 使用鼠标滚轮时。

通常没有按键时，鼠标滚轮的每一次点击速度是一个整数，而 SHIFT、CTRL 和 ALT 的速度是 10、0.1 和 0.01。可以在以下页面中解释的配置中修改这些值。

选项面板



按下位于主窗口右上角的齿轮按钮可打开此面板。



在接下来的几页中，我们将了解此面板各个区域的含义和用途。

选项面板- 保存选项

Save options

- Save immediately after each change
- Save when the cursor line is changed
- Save when this application loses focus
- Save when "CTRL" + "S" is pressed

这些选项确定命令列表何时保存到磁盘。

它可以通过多种方式自动保存，也可以在您按下 **CTRL-S** 时自动保存。在所有情况下，列表也会在从文件加载另一个列表或关闭应用程序之前保存。

选项面板-配置

Configuration

ConfigFile_v2.txt

Select

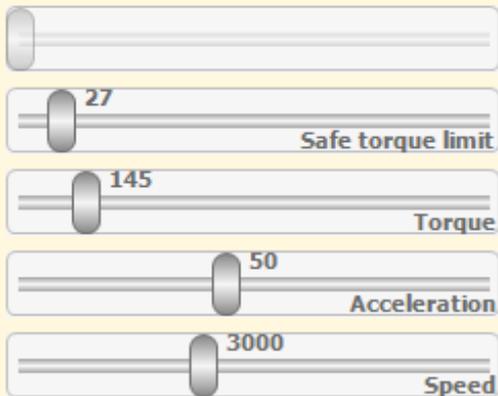
Edit

使用这些按钮，您可以选择要使用的配置文件并在文本编辑器（通常是记事本）中打开它进行编辑。

配置的详细信息在接下来的页面中解释。

选项面板 - 运动选项

Movement options



顶部的第一个滑块在这里被禁用，用于限制“协作”模式下的配对。

如果您右键单击滑块转到默认值。这些游标的最大值和默认值在配置中。

价值观力矩,加速度和速度它们用作电机本身内部的限制。

在某些引擎 (FeeTech) 中，固件中的扭矩限制没有得到很好的控制，当它降低时，加速度不再有效，因此最好将其保持在最大值 [注 1](#)

扭矩限制委托给协作机器人应用程序，当扭矩上升到超过中设置的限制时，它将降低电机扭矩安全扭矩限制

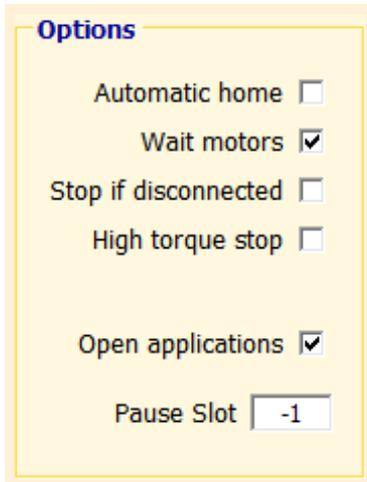
[注 1](#) -如果必须将扭矩保持在最大值，则可以将

并非所有引擎都在固件中实现这些设置，
唯一使用协作机器人应用程序所有功能的

他们是 FeeTech 和 TMOT (Theremino Motors) (2022 年计划中)。

选项面板 - 选项

这些选项控制应用程序的一些一般行为。



- **Automatic home** - 每次应用程序启动和关闭时启用自动移动到起始位置 [注 1](#)
- **Wait motors** - 通常你保持启用和禁用它只是为了尝试没有电机的命令序列。
- **Stop if disconnected** - 如果启用它，如果电机电压降至 2 伏以下，它会停止执行序列 [笔记 2](#)
- **High torque stop** - 如果启用它，它会在扭矩过大时停止执行序列。
- **Open applications** - 如果启用此选项，则所有名为 `Theremino_xx.exe` 的应用程序将在启动和关闭时启动

和关闭。要启动的应用程序必须位于同一文件夹和 `Theremino_Cobot.exe` 文件的子文件夹中。

通常最好创建一个“Apps”文件夹来包含所有要启动的应用程序。

如果您使用 `Theremino_Automation` 应用程序那么最好也将此任务留给它 [注 1](#) 并禁用此选项。

- **Pause Slot** - 用于通过外部按钮控制暂停。通过设置有效的 Slot 编号，如果该 Slot 的值超过 500，则执行将暂停。当该值低于 500 时，将重新开始执行。通过设置 -1，此选项处于非活动状态。

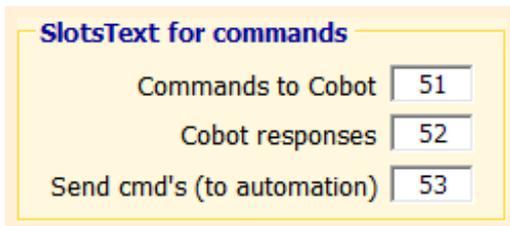
[注_1](#)

如果您使用 `Theremino_Automation` 应用程序，那么您将获得更强大的操作，从自动化驱动一切并禁用选项 **自动回家**和**打开应用程序**。此外，通过自动化，还可以更好地管理断开连接事件和发送命令 `StopExecution` 必要时。在这种情况下，建议也禁用该选项 **如果断开连接则停止**。

[笔记 2](#)

从与每个电机相关的插槽中读取电机电压。插槽值在配置中。

选项面板 - 命令的 SlotsText



SlotsText for commands

Commands to Cobot

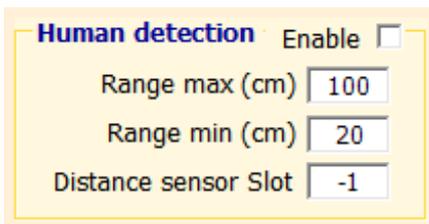
Cobot responses

Send cmd's (to automation)

这里设置了插槽文本 [注 1](#) 从外部应用程序接收命令，给出答案并命令外部应用程序（通常是 Theremino_Automation）。

如果您不使用它们，请禁用它们将它们设置为值 -1 [注_1](#)-文本槽用于在应用程序之间传递文本消息。

选项面板 - 人体检测



Human detection Enable

Range max (cm)

Range min (cm)

Distance sensor Slot

这些选项调整渐进式安全区的开始和结束。低于 Range min 时，速度被限制在最低限度。超出 Range max 不受限制。在这两个极端之间存在速度的渐进限制。

为了使这个机制起作用，距离传感器插槽设置一个有效的插槽值，然后一个距离传感器连接到这个插槽。

为了获得最大的可靠性，我们建议使用多个传感器 [正如我们网站的这一部分所解释的](#)。

通过增加传感器的数量（甚至增加到 17 个而花费很少），并将它们指向不同的方向，你就会确信至少有一个传感器将检测安全区内是否有人。

选项面板 - 添加新位置/更正所选位置



Add new position / Correct selected position

Repeat delay

Repeat time

Button Slot

这些选项调整初始延迟和重复时间（以毫秒为单位）以及用于外部按钮的插槽。然后将设置 HAL 应用程序，以便通过按下此按钮，插槽中的值超过 500。

按下按钮时，将在命令列表中添加一个新行，其中包含所有电机的当前位置，如果长时间按住，将添加许多行，并通过两个光标调整延迟和节奏。

如果选择了一行（以蓝色背景突出显示），则外部按钮会对其进行修改，而不是添加新行。

要禁用此选项，请将值设置为 -1

了解和更改配置

我们将在以下页面中解释的许多操作取决于配置中写入的值。在接下来的几页中，我们将一一解释配置的各个部分。

每个引擎的引擎类型和参数

马达- 以“Motor”开头的行数决定了电机的数量

姓名- 电机的类型决定了运动的一些特性

老虎机 - 这个引擎的插槽从这个开始**基本号码**

分钟- 限制电机运动（必须小于 Max）

最大限度- 限制电机运动（必须大于最小值）

Type	Name	Slots	Min	Max
Motor	STS3215	100	-30000	+30000
Motor	STS3215	200	-30000	+30000
Motor	STS3215	300	-30000	+30000
Motor	STEPPER	400	+0	+360
Motor	SERVO	500	+50	+950
'Motor	SERVO	600	+50	+950
'Motor	Sente150	700	-860000	860000

电机参数插槽

这些是引擎寄存器映射中要添加到**基本号码**引擎的。

MOTOR SLOT INCREMENTS
SlotDestination1
SlotPosition2
Slot Velocity3
Slot Torque4
SlotVoltage5
SlotTemperature6
SlotMoving7
SlotCurrent8
SlotTorqueLimit20
SlotAccLimit21
SlotSpeedLimit22
SlotOffOnCenter23

MoveTo 指令编号的整数和小数位

3 和 2 的示例: `MoveTo +123.45 -123.45 ----- +000.00`

=====

= DIGITS (3 to 9) and DECIMALS (0 to 9) for edit numbers

=====

digits3

decimals 2

鼠标滚轮调整位置的速度

通过更改这些乘数，您可以达到您喜欢的速度。

=====

= EDIT SPEED MULTIPLIER

= -----

= Normally 1 (with Feetech or Theremino motors)

= Set to 10, 100 or more to increase the mouse wheel effect

=====

EditSpeedMultiplierSHIFT 10

EditSpeedMultiplierCTRL 0.1

EditSpeedMultiplierALT 0.01

EditSpeedMultiplier 1

GotoCenter 和 Home 命令的公差

这是将在 GotoCenter 和 GotoHome 命令之后使用的公差

=====

= END MOVING TOLERANCE

= -----

= Before to update Acc Or Speed and after GotoCenter and Home

=====

EndMovingTolerance10

扭矩、加速度和速度幻灯片的默认值

当您用鼠标右键单击时，光标会转到由这些行建立的值。

=====

= SLIDERS

=====

SafeTorqueMax500'usually 200 to 400

Safe Torque Default150'usually 150

Torque Default500' min=1 max=1000

AccDefault50' min=1 max=255

SpeedDefault3000' min=1 max=65535

命令序列

移动电机的命令

MoveTo n n n n.. 数字 n 是以步、毫米或度为单位的目的地

MoveTolerance n 数字 n 表示以步长、毫米或度为单位的公差

MoveSpeed n 以步进速度、毫米或每秒度数进行插值

MoveTime n 以秒为单位插入固定时间

命令 **MoveTolerance** 在每个 **MoveTo** 结束时导致等待，直到所有电机都在指定的公差范围内到达目的地。

最后一条命令 **MoveTolerance**、**MoveSpeed** 或者 **MoveTime** 遇到它，它用于所有后续行，直到遇到另一个。

命令 **MoveSpeed** 和 **MoveTime** 他们造成运动插补。到达目的地的路线来了每秒发送一百次将总距离分成许多小部分。这样所有引擎一起到达目的地。

对于一些通信缓慢且没有加速的引擎（例如 **Sentel**），插值效果很差。要禁用它，请写 **MoveTime 0**

扭矩、加速度和速度调节

Safe Torque n 仅适用于智能电机（不包括步进电机和伺服电机）

Torque n 仅适用于智能电机（不包括步进电机和伺服电机）

Acceleration n 仅适用于智能电机（不包括步进电机和伺服电机）

Speed n 仅适用于智能电机（不包括步进电机和伺服电机）

检查

Breaks n 暂停“n”秒

Restart 从第一行重新启动程序

Stop 停止程序的执行

与其他应用程序的通信命令

Slot x = n 将值 n 写入槽 x

SendCommand string 向自动化或其他应用程序发送命令

所有用“n”表示的数字也可以是小数
对于小数，您可以使用小数点或逗号。

以秒为单位的时间也可以表示分数，精确到千分之一秒。

命令——解释

搬去

将引擎移动到指定的目的地。

这里有些例子：

`MoveTo +000 +000 +000` 所有三个引擎都在 0 度

`MoveTo +360 ---- ----` 电机 1 到 360 度，电机 2 和 3 保持原位

`MoveTo +89.9 +11.1 +3.00` 引擎 1、2 和 3 有一个整数和两个小数

`MoveTo -89.9 -11.1 -3.00` 与上一行相同的值，但为负数

在配置中（在前面的页面中解释过），您可以指定要显示多少个小数位和多少个整数位。

通过正确设置位数和小数位数，可以使所有数字的列保持良好对齐，即使是正数、负数和有很多小数的情况。

破折号表示引擎的目的地与之前的目的地相同。所以为不得移动的电机设置破折号。

移动公差

公差由指令使用 `MoveTo` 用于等待所有电机动作完成。

程序继续到下一行仅当目标位置和当前位置之间的差异小于指定的公差时。

这里有些例子：

`MoveTolerance 0.5` 等待所有引擎都在目的地半度范围内

`MoveTolerance 15` 快速逼近路径

移动速度

该程序运行每操作说明 `MoveTo` 以预设速度。

每一条指令 `MoveTo` 将发送到引擎许多中间目的地，每十毫秒一个，直到时间用完。

时间用以下公式计算： $\text{小路} \times \text{速度}$ （单位，即步/度或毫米）划分速度（以每秒为单位），使用最远的引擎路径。

如果所有引擎都足够快以遵循这些指示，那么所有引擎将一起到达目的地。

这里有些例子：

`MoveSpeed50` '发送许多中间位置以遵守指示的速度'

`MoveSpeed999` '进行得非常快'

在后一种情况下，电机可能会滞后，因此程序将等到所有电机都到达 `MoveTolerance` 范围内，然后再转到下一行。

移动时间

该程序运行每操作说明 `MoveTo` 在预先设定的时间内。

每一条指令 `MoveTo` 将发送到引擎许多中间目的地，每十毫秒一个，直到时间用完。

如果所有引擎都足够快以遵循这些指示，那么所有引擎将一起到达目的地。

这里有些例子：

`MoveTime 2` '在两秒内插入许多位置'

`MoveTime 0.01` '进行得非常快。'

在后一种情况下，电机可能会滞后，因此程序将等到所有电机都到达 `MoveTolerance` 范围内，然后再转到下一行。

与一些通信缓慢且没有加速的引擎
(例如 `Sentel`) 插值效果不佳，电机不稳定。
要禁用它，请写 `MoveTime 0`

安全扭矩

该指令设置用于调整安全扭矩的 **SafeTorque** 滑块。命令仅对智能电机有效（步进电机和伺服电机无效）。

目前这个命令正在建设中，很大程度上取决于所使用的电机类型。并非在所有情况下它都会按预期工作。

这里有些例子：

Safe Torque 300 '所有安全扭矩为 300 的电机'

力矩

该指令设置向所有电机发送最大扭矩的扭矩滑块。命令仅对智能电机有效（步进电机和伺服电机无效）。

这里有些例子：

Torque 300 '所有发动机最大扭矩 300'

加速度

该指令设置向所有电机发送加速度的加速度滑块。命令仅对智能电机有效（步进电机和伺服电机无效）。

这里有些例子：

Acceleration 20 '所有加速 20 的引擎'

速度

该指令设置向所有电机发送速度的速度推子。命令仅对智能电机有效（步进电机和伺服电机无效）。

这里有些例子：

Speed 1000 '所有电机都以 1000 速度'

休息时间

该指令在以秒和分数表示的时间内停止执行。

这里有些例子：

Pause 12 '等待 12 秒'

Pause 0.5 '等半秒'

Pause 1.53 '等待 1 秒和 53 美分'

重新开始

该指令重新启动程序

这是一个例子：

Restart '从程序的第一行重新开始'

停止

该语句停止程序

这是一个例子：

Stop '停止程序执行'

老虎机

该指令将一个立即数写入一个 Slot

这里有些例子：

Slots 3 = 1000 '在槽 3 中写入数字 1000'

Slots 9 = 0.33 '在槽 9 中写入数字 0.33'

Slots 999 = 10 '在 Slot 999 中写入数字 10'

发送命令

该指令在选项面板中指示的 SlotText 中发送一个文本字符串：发送命令（到自动化）

发送命令后，SendCommand 指令等待命令被接收（即等待 SlotText 再次为空）。

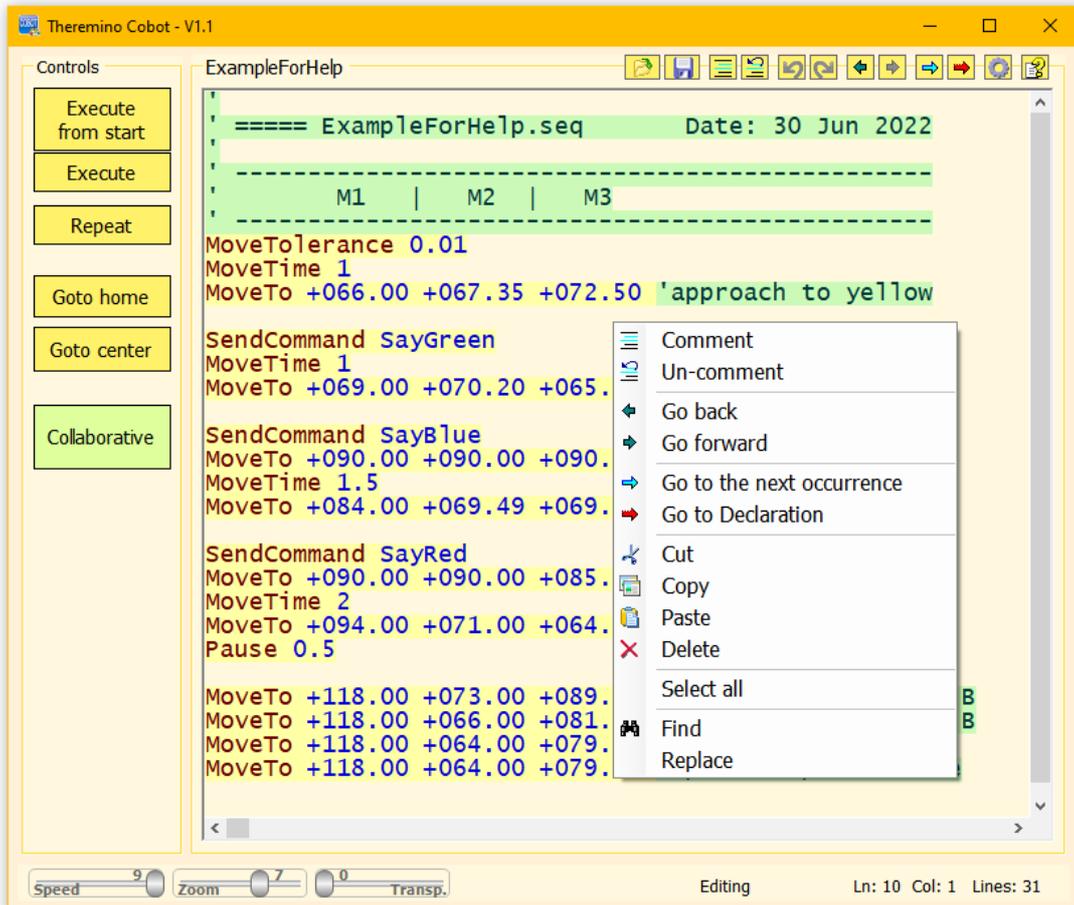
这里有些例子：

SendCommand Beep '发送“哔”命令'

SendCommand Multiple beeps '发送“多声蜂鸣”命令'

程序菜单

通过单击程序区域，使用鼠标右键（或通过触摸触摸屏两秒钟而不松开手指），将打开如下所示的菜单。



“注释”和”取消注释”它们用于对程序的整个区域进行评论（添加初始引述）。或者删除评论。

“回去”和”直走”他们将光标和可见页面移动到之前访问过的程序部分。

“转到下一个事件”搜索所选单词的其他出现。

“去申报”仅在声明行中搜索选定的词。

命令”切”，”复制”，”粘贴”，”删除”和”全选”、复制、粘贴、删除和选择部分程序。相反，您也可以使用 CTRL-X、CTRL-C、CTRL-V、DEL 和 CTRL-A 键。

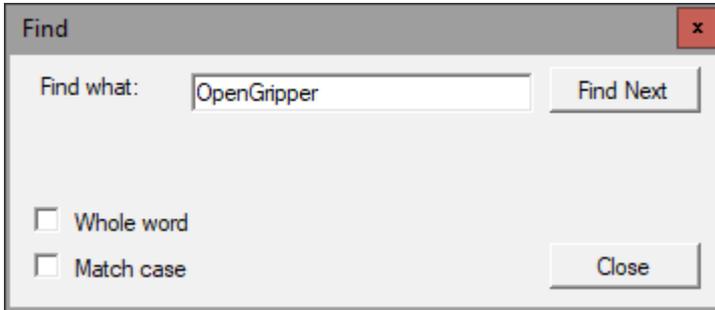
“寻找”（或 CTRL-F）和”代替”，打开用于查找和替换单词和短语的窗口。

查找和替换

应用程序菜单底部的最后一项打开两个类似的窗口。



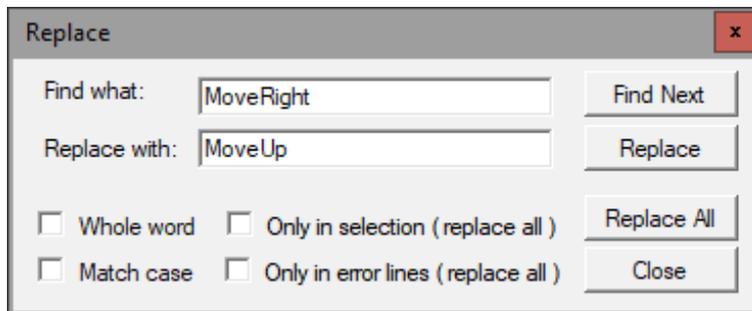
“查找”窗口



使用此窗口，您可以在程序文本中搜索单词或短语。

- ◆ 如果启用“整个单词”，则单词必须是完整的。
- ◆ 如果启用“匹配大小写”，则单词的大小写也必须匹配。
- ◆ 使用“查找下一个”（或使用 **F3**），您可以转到下一个要查找的单词。如果到达程序末尾，则搜索从头重新开始。

“替换”窗口

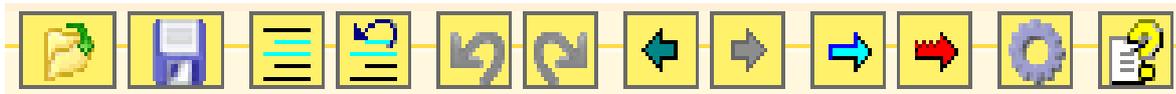


此窗口具有与前一个窗口相同的选项，但它还允许您将单词（或短语）替换为另一个。

如果按下“替换”，则只进行一次替换。取而代之的是“全部替换”，所有出现的地方都被替换。

替换可以发生在整个程序中，也可以只发生在选定的区域，或者只发生在包含错误（或“警告”）的行中。

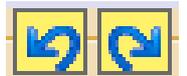
顶部栏控件



前两个按钮加载和保存命令序列。



这两个按钮注释和取消注释所选定行。



两个蓝色箭头它们用于回滚程序更改和重建已删除的更改。



两个深色箭头将光标和可见页面移动到之前访问过的程序部分。



蓝色箭头查找所有出现的函数、变量甚至简单的单词。



红色箭头只搜索 **dichiarations (Button, Key, Label and Variable)** (继承自 Automation, 目前几乎不可用)。

搜索功能很方便，只需选择一个词，甚至只需将光标放在上面，然后反复按箭头即可。



齿轮打开选项窗口。

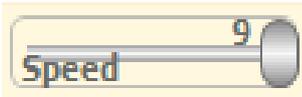


问号以所选语言打开说明文件（帮助）。要使此命令起作用，您需要将首选语言的帮助文件复制到“文档”文件夹中。

如果未找到帮助文件，则会出现一条消息，建议您打开 **Docs** 文件夹并将文件复制到其中。

或者，您可以选择位于“文档”文件夹或任何其他文件夹中的首选语言的帮助文件。
要更改所选文件，请用鼠标右键单击按钮。

底部档控件



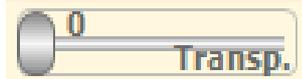
此滑块调整播放速度。

速度范围从“1”（每秒一条指令），据，直到...为止“8”（每秒一万条指令），它在“9”（系统允许的最大速度）。

编写程序时最好使用中等速度。速度一般“5”（每秒 20 条指令），速度足够慢，您可以直观地跟踪程序的执行情况。



ZOOM 滑块确定程序窗口和调试窗口中文本的大小。



此滑块可调整主窗口的透明度，并允许您在其下方也能看到。

Executing line 15

底栏中间显示当前状态，可以是“正在执行第 nn 行”、“就绪”、“正在编辑”和其他指示特殊情况的命令。这些相同的消息也在文本槽中发送 **CobotResponses** 并且可以由 Theremino 自动化或其他应用程序读取。查看有关页面 [与自动化通信](#)。

Ln: 17 Col: 1 Lines: 18

底部栏的右侧部分显示有关程序的信息：

- 总行数
- 光标所在行（从第 1 行开始）
- 光标所在的列（从第 1 列开始）

先进技术

正如我们已经看到的，Cobot 应用程序故意简单化，这极大地促进了它在大多数情况下的使用，但也有一些情况无法简单地管理，需要特殊技术。

在接下来的几页中，我们将解释一些有助于以下操作的技术：

- 做直线运动
- 以正交坐标 X、Y 和 Z 移动
- 将命令发送到 Theremino 自动化应用程序
- 从 Theremino 自动化应用程序或 theremino 系统的其他应用程序接收命令。

直线运动

使用 **MoveSpeed** 或 **MoveTime** 将运动分成许多小段，所有电机每秒驱动一百次（如果使用的通信线路允许）并且所有电机都将一起到达目的地。

MoveSpeed 和 **MoveTime** 改善了路径控制，但如果目的地很远，您仍然会看到明显的曲线运动。如果有必要尽量减少此问题，可以使用以下技术：

- 降低 **MoveSpeed**（或增加 **MoveTime**）以检查速度过快是否会产生不需要的转弯。对于某些必须比其他电机进行更大运动的电机，速度可能高于电机中设置的限制（在步进电机的 **HAL** 中或可编程电机的 **Cobot** 应用程序的光标中）。在这些情况下，一些引擎落后，路径遵循的曲线甚至比不可避免的曲线更大，这是由力学的几何形状引起的。
- 如果降低速度还不够，那么建议再次提高速度 **e** 然后将动作分解成多行以创建间隔更近的路点。因此，必须在需要精确移动和位置的地方设置短阶段，反之亦然，在路线精度无关紧要的区域中，阶段彼此远离。
- 将路线的航路点放在合适的地方。一个点必须位于您想要获得最大精度的区域的起点，而其他点必须在必须精确的整个区域中彼此靠近。

3D 扩展

在某些情况下，所提出的方法可能还不够，例如制作一条长而完美的笔直焊缝。

因此，我们也在准备进行三维计算的可能性，并因此指定手臂尖端在三个空间坐标 **X**、**Y** 和 **Z** 中的位置，但请记住：

- 设置功能将变得更加复杂，手臂操作也不会像现在这样简单。
- 如果请求很多，此选项可能会在 **2023** 年可用。

从协作机器人到自动化的命令

在自动化我至标签 `Event_CommandsFromCobot` 用于接收来自 `Theremino_COBOT` 应用程序的命令, 但它还可以从任何其他能够写入“文本槽”的应用程序接收命令。

问当一个外部应用程序写入一个带有命令的字符串时 `Slot_CommandsFromCobot` 通过调用此事件通知事件标签。

Automation 程序被中止, 无论它在做什么, 并且执行 `Label` 之后的指令, 直到 `Return`。命令文本使用“`CommandText`”函数读取, 然后使用“`Select`”结构解码, 如下例所示:

```
Variable Numeric Slot_CommandsFromCobot = 53
Stop

Label Event_CommandsFromCobot
  Select CommandText
    "Beep" houses
      ExecBeep
    '
      "Multiple Beep" Houses
      ClearCommand
      ExecBeepMultiple
    '
  CaseElse
    Print "Unrecognized command: " + CommandText
  EndSelect
  ClearCommand
Return

Label ExecBeep
  Beeps 440 300
  Wait Seconds 0.5
Return

Label ExecBeepMultiple
  For v1 = 1 To 3
    Beeps 880 400
    Wait Seconds 0.5
  Next
Return

Label ClearCommand
  SlotText(Slot_CommandsFromCobot) = ""
Return
```

请注意, 在“多声”房子添加了一条指令清除命令在执行命令之前。该指令告诉 COBOT 应用程序立即继续, 无需等待命令完成执行。

下一页是发送这些命令的序列示例。

从 COBOT 到 Automation 的命令

本章解释了如何将命令从 COBOT 应用程序发送到自动化应用程序。以下示例是移动三个电机并在一次移动和下一次移动之间发送“Beep”和“Multiple Beep”命令的序列。

```
Move Tolerance 1
MoveTime 0.5
MoveTo +00137.000 +00300.000 +00022.000
SendCommandBeep
MoveTo +00153.000 +00303.000 +00025.000
SendCommandMultiple beeps
```

包含此序列的文件由 Theremino_COBOT 应用程序运行，并且必须位于其“序列”文件夹中。

发送到自动化应用程序的命令可以是任何文本字符串，甚至可以用空格分隔，并且在不考虑大写或小写的情况下进行解码。

COBOT 应用程序在发送之前将任何多个 TAB 或空格字符转换为单个空格，并且去除前导和尾随的 TAB 和空格。

用户自己可以编写自己的命令，记住：

- 命令必须按照 COBOT 应用程序的顺序编写。
- 在自动化的 CommandsFromCobot 事件中解码它们的说明。

在 Cobot 应用程序中，您必须设置要用于命令的文本槽。打开选项面板（齿轮在右上角）并找到位于底部的“文本命令槽”部分。

在 Cobot 应用程序中设置的文本槽必须与在自动化变量中声明的相同，名称如下：

Slot_CommandsToCobot, Slot_CobotResponses 和 Slot_CommandsFromCobot。

您还可以发送按下按钮的命令，发送他们的名字，
这些方法在自动化文档中有更好的解释。

另请参阅文件夹中的示例
“演示程序\SlotText 命令”

在下一页，我们将看到命令也可以从自动化应用程序发送到协作机器人应用程序。

从自动化到 COBOT 的命令

要向协作机器人应用程序发送命令，需要设置变量 **Slot_CommandsToCobot** 和 **Slot_CobotResponses** 然后在自动化（或其他应用程序）中使用它们以及写入和读取文本槽的命令。

从自动化到 COBOT 的命令 (中的文本字符串 **Slot_CommandsToCobot**)

- **ExitFromEdit** (退出“编辑”状态)
- **StartExecution** (自动退出“编辑”状态)
- **StopExecution** (自动退出“编辑”状态)
- **GotoHome** (自动退出“编辑”状态)
- **Goto Center** (自动退出“编辑”状态)
- **EnableRepeat**
- **DisableRepeat**
- **SelectLine nnn**
- **ExecuteSelectedLine**
- **ArrowUP**
- **ArrowDOWN**
- **EnableCollaborative**
- **DisableCollaborative**
- **AddNewPosition**
- **CorrectSelectedPosition**
- **DeleteSelectedLine**
- **SetHomePosition**
- **SetCenterPosition**
- **SetHoldingTorque nnn**
- **SetSafeTorqueLimit nnn**
- **SetTorque nos**
- **SetAcceleration nnn**
- **SetSpeed nnn**
- **LoadSequence SeqName.seq** (名称“xxxxx.seq”不能包含空格)

Cobot 应用程序在接收和解释文本后从 Slot 中删除文本。发送命令的应用程序必须等待命令被接收才能发送其他应用程序。

来自协作机器人的答案 (中的文本字符串 **Slot_ResponsesFromCobot**)

- **Editing**
- **Ready**
- **Execution running at line nnn**
- **Motors disconnected**
- **Human detected - Speed reduced**
- **Human too near - Execution stopped**
- **Too much torque - Execution stopped**
- **Motors disconnected - Execution stopped**
- **Execution paused**