

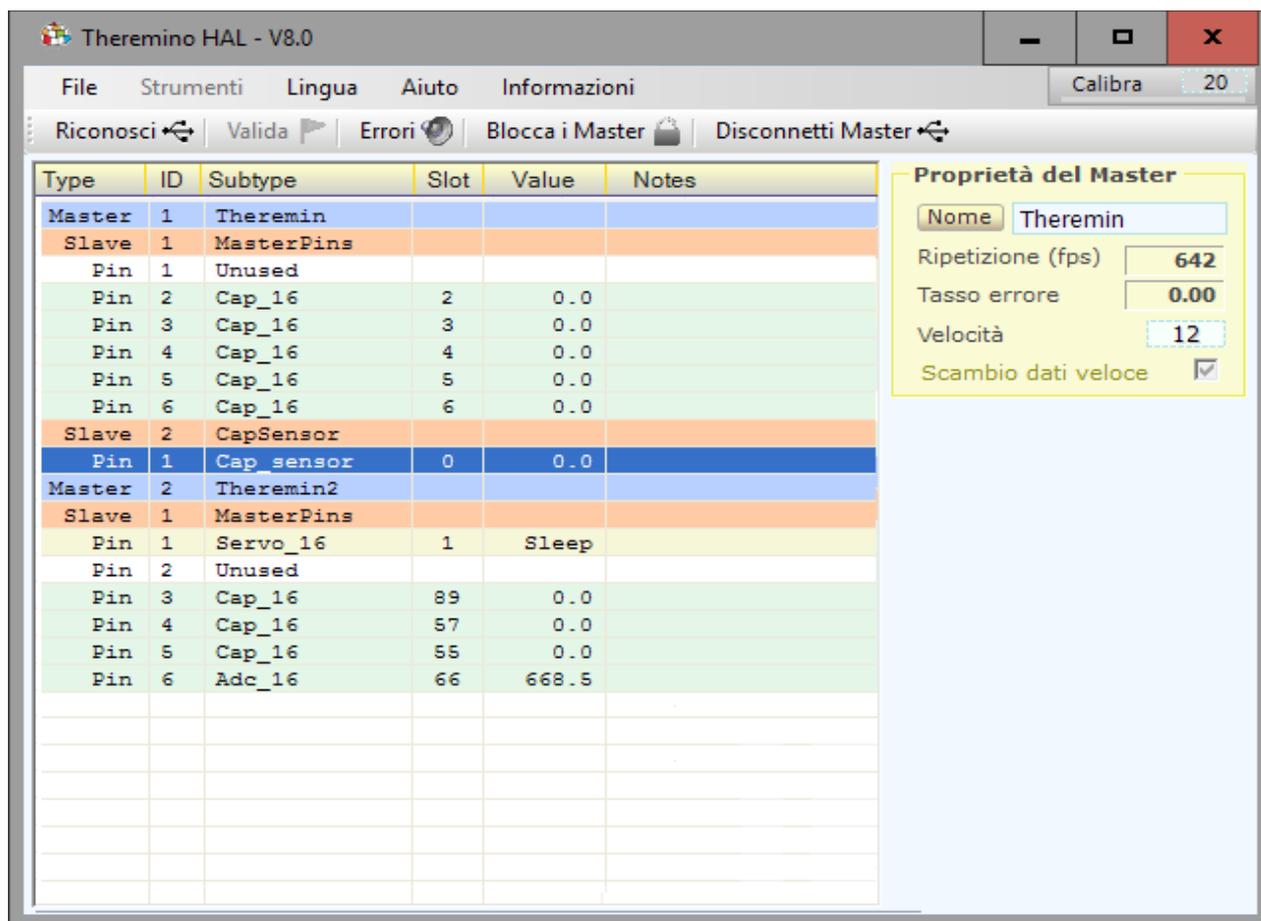
**theremino**  
•the•real•modular•in-out•

系统特雷米诺

# Theremino HAL-V8.x

## 指示

# Theremino 哈尔



## Theremino HAL 连接了两个主站

Theremino HAL (硬件抽象层) 是一个硬件管理器 - 它具有相当简单的界面, 但通过高度优化的算法执行复杂的操作。

Theremino HAL 是与硬件通信的核心, 它知道如何同时与许多主站通信, 它知道 USB 协议和串行通信, 它知道所有最常见的输入输出类型, 并且知道如何识别“从”模块

如果没有 HAL, 与硬件的通信将会很困难 (与 Arduino 一样), 需要大量的时间和工作 (与 Arduino 一样), 最后, 对于每种类型的 InOut, 要移动电机, 甚至只是读取密钥, 您都需要必须编写特定的固件 (与 Arduino 一样)

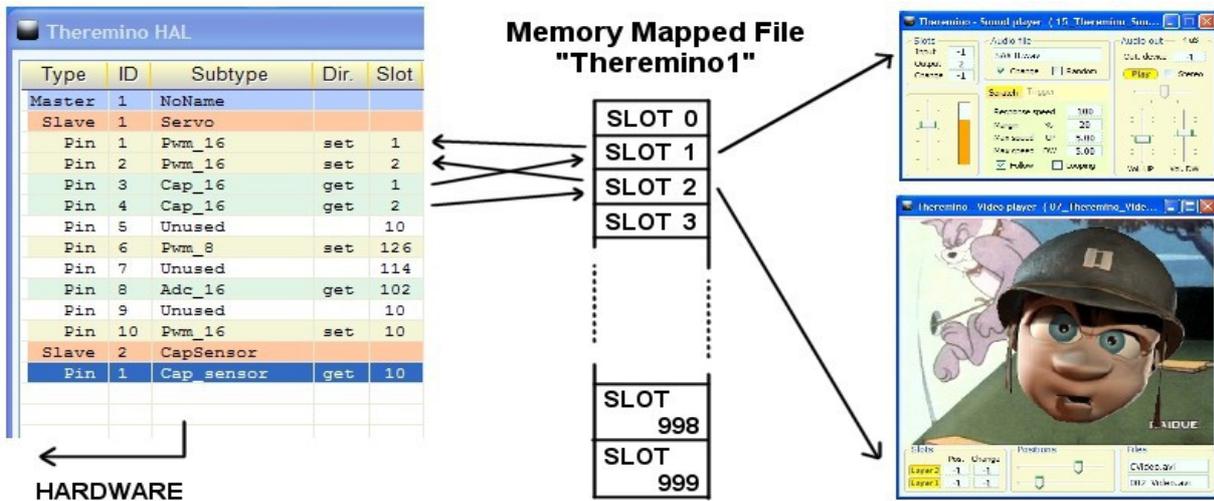
**如果您使用硬件模块** 那么 HAL 是必不可少的, 必须保持打开状态, 可以将其最小化, 但必须保持运行状态。

**如果你不使用硬件** 那么 HAL 就不是必需的, 系统应用程序可以通过 Slots 相互通信, 即使没有 HAL。

**添加或删除从站模块时**, 您会收到配置已更改的通知, 并在列表中显示红线。如果您选择丢失旧配置并适应当前硬件, 则“验证”按钮将使新配置生效。

## “老虎机”

Theremino 系统的“插槽”用 0 到 999 之间的数字进行标识，并且都是名为“Theremino1”的 MemoryMappedFile 的一部分。每个槽都包含一个“浮点数”，可以由 Theremino 系统的任何组件读取或写入。



在此图中，只有 HAL 写入插槽，但实际上系统的所有组件都可以读取和写入任何插槽，即使已被其他插槽使用。

选择要使用的插槽时，必须注意两件事：

- ◆ 检查您是否意外地将同一个插槽用于两个不同的功能。
- ◆ 避免将两个人写在同一个 Slot 上。

写入插槽的输入引脚以浅绿色突出显示。如果两个或多个输入引脚具有相同的插槽，则 HAL 应用程序会用红线和措辞发出警告**插槽冲突**。

中号许多应用程序和许多引脚可以读取同一插槽，但您必须避免配置多个引脚在同一插槽上写入，这样做不会破坏任何东西，但你会得到不确定的结果。

如果你向同一个 Slot 发送多个数据流，那么数据会混合在一起，最后写入的数据流获胜，如果你想有序地合并数据，规则是必要的。

Type	ID	Subtype	Slot	Value	Notes
Master	1	TestSlotCo...			
Slave	1	MasterPins			Firmware V5.0
Pin	1	Adc_16	1	105.3	
Pin	2	Adc_16	2	99.5	
Pin	3	Dig_in	4	0.0	SLOT CONFLICT
Pin	4	Dig_in	4	0.0	SLOT CONFLICT
Pin	5	Dig_in	5	0.0	
Pin	6	Dig_in	6	0.0	
Pin	7	Dig_in pu	7	1000.0	
Pin	8	Unused			

为了在插槽之间建立数学和逻辑规则，并编写复杂的行为算法，使用 Theremino\_Automation 或 Theremino\_Script，或任何编程语言，如 C++、CSharp、VbNet 或 VB6，但也可以使用可视化语言，如最大 MSP、处理、PureData、LabView 和 EyesWeb。此处已为 MaxMSP 准备好插件和示例：

[www.theremino.com/downloads/foundations](http://www.theremino.com/downloads/foundations)

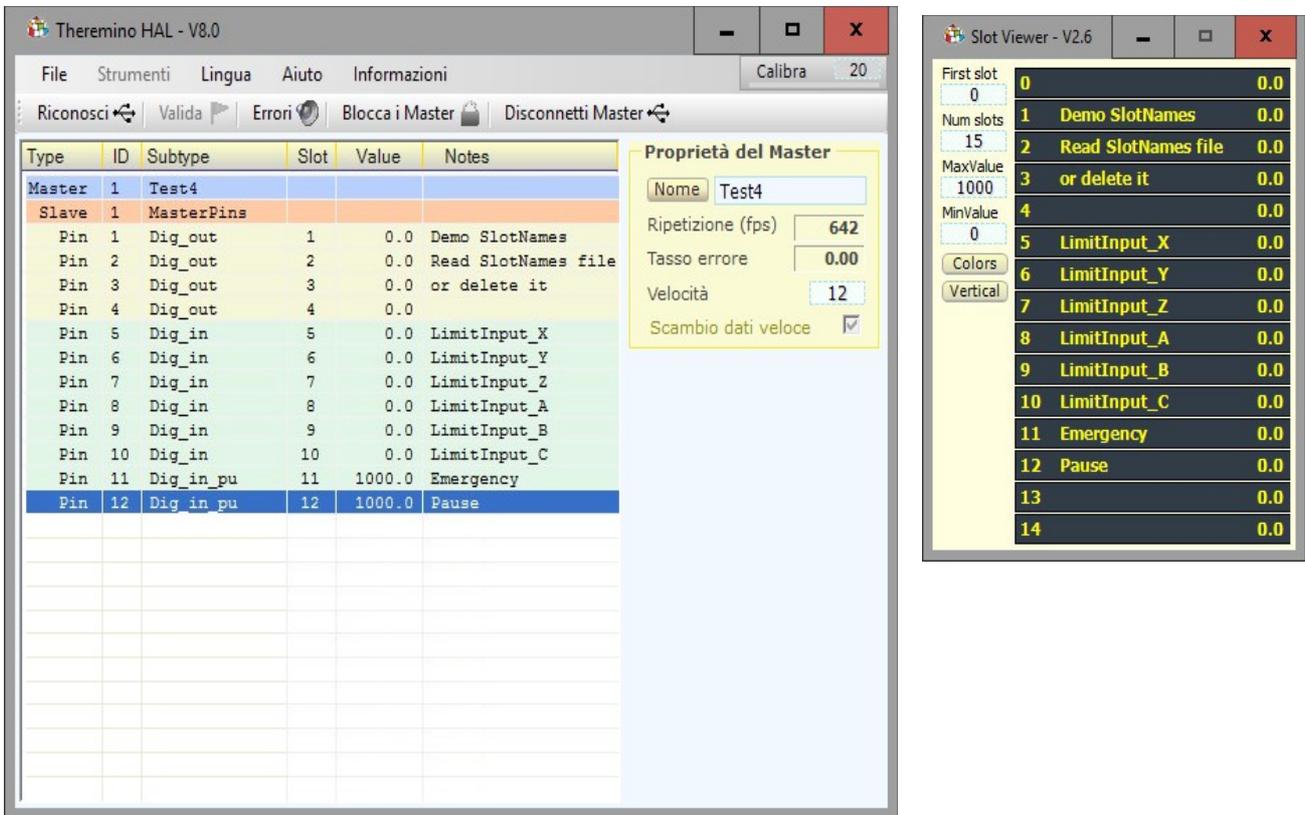
有关这些页面上的通信的更多信息：

[www.theremino.com/technical/communications](http://www.theremino.com/technical/communications)

[www.theremino.com/technical/pin-types](http://www.theremino.com/technical/pin-types)

## 插槽名称

HAL 应用程序（从版本 5.5 开始）和 SlotViewer（从版本 2.6 开始）可以显示插槽的名称（或注释或注释）。



值得注意的是，名称与物理引脚无关，而是与插槽相关。

名称写入文件中，该文件必须名为“SlotNames.txt”，并且必须与“Theremino\_HAL.exe”和“Theremino\_SlotViewer.exe”位于同一文件夹中。如果“SlotNames.txt”文件不存在，注释字段将保留为空。

要编辑插槽名称，请打开“文件”菜单，选择“编辑 SlotNames.txt 文件”，然后使用默认编辑器（通常是记事本或写字板）进行编辑。最后，文件被保存并自动重新加载。

编写规则很简单，并在示例文件中显示，该示例文件可以在最新版本的 HAL 和 SlotViewer 中找到。

文件的每一行都以插槽号开头，后跟空格和要显示的文本。该行还可以继续添加注释部分，该部分不会显示，前面带有单引号。

如果您想对 HAL 和 SlotViewer 使用相同的注释文件，则需要将文件“SlotNames.txt”、“SlotViewer.exe”和“HAL.exe”保留在同一文件夹中。

## 文本命令槽

从 2024 年初开始，Theremino 系统应用程序和用户创建的应用程序可以发送命令以文本形式并使用两个从 HAL 接收响应文本槽进行沟通。

在接下来的几页中，我们将看到也存在一个老虎机发送数字命令，但是使用文本命令，您可以做更多的事情，几乎是您在 HAL 界面上使用鼠标和键盘可以做的所有事情。

### 何时使用数字槽而不是文本槽

只有一种情况最好使用数字命令槽。

当您想知道已识别且处于活动状态的主设备（即硬件模块）的数量时，使用数字插槽，您只需读取插槽的值，这会在几微秒内发生，而如果我们使用 TextSlots 来做到这一点需要发送命令来询问该值，然后等待响应。这将需要数十毫秒。

出于这个原因，也为了让那些已经在应用程序中使用它们的人更容易，我们还保持了数字槽命令的活动状态。

### 用于 TextSlot 的数字

这文本槽 它们有从 0 到 999 的数字，就像数字槽一样，但它们是两个不同的东西。即使使用相同的号码文本槽和老虎机数字 他们互不干扰。

如果没有设置不同，则文本槽 零对于命令和文本槽 一寻求答案。

### 使用其他插槽代替 TextSlots 0 和 1

通常命令 TextSlots 是零和一，但您可能想在同一台 PC 上使用多个独立的应用程序。在这些情况下，每个应用程序将与其 HAL 一起驻留在单独的文件夹中，并使用“Lock Master”命令访问其主槽位。在这些情况下，可以为每个 HAL 分配不同的命令槽。对于命令您可以使用任何 TextSlot（从 0 到 999），但请注意不要将其用于其他用途。

要将编号分配给命令槽，请手动修改“Theremino\_HAL\_INI.txt”文件的最后几行。例如，要使用 TextSlots 300 和 301，您可以编写：**命令文本槽=300 和响应文本槽=301。**

注意不要删除“=”符号。如果出现错误，则使用插槽 0 和 1，并且 HAL 重写 INI 文件中的正确行。

### 如何发送命令

发送一个命令后，在发送另一个命令之前，最好等待至少 50 或 100 毫秒，以便让应用程序有时间执行写入和读取，即使在最坏的情况下也是如此。

为了最大限度地减少等待时间，您可以检查 CommandTextSlot 中的文本，并在从 HAL 中删除该文本后立即继续。

请参阅自动化示例：**演示程序\SlotText Commands\Commands\_to\_HAL.txt**

## 带有 TextSlots 的命令

在接下来的几页中，我们将列出所有可用的命令，但只有 HAL 应用程序可以识别它们。

ArduHAL、IoTHAL 和 NetHAL 仅使用部分命令，并且许多命令仅在选择使用它们的 Pin 类型时才有效。

以 ON/OFF 结尾的命令是启用。除零或单词 OFF 之外的任何值均被视为 ON。通常使用 1/0 或 ON/OFF。

这两个命令及其附加参数都可以用大写或小写字母的任意组合书写。

命令参数，例如主控名称、引脚类型、滤波器类型和 Adc24 通道类型，仅根据其字母数字字符进行识别。对于参数来说，空格和其他符号并不重要。例如“Dig\_in\_pu”可以写成“diginpu”甚至“DIG-在\_\_PU”并且仍然会被识别并被认为是有效的。

### 顶栏命令

- **认出** 它与识别按钮具有相同的效果。
- **证实** 与“验证”按钮具有相同的效果。
- **错误蜂鸣声开/关** 启用或禁用错误蜂鸣按钮。
- **锁大师开/关** 启用或禁用“锁定主控”按钮。
- **断开** 断开所选主站（模块）的连接。
- **否编辑开/关** 启用或禁用“NoEdits”按钮。
- **已校准** 启用或禁用校准按钮。
- **校准值 否(1..100)** “校准”框的数值。

### 主命令（或模块）

- **选择 Master N（注 1）** 选择主控（如果有多个主控）。
- **加载配置 xxx** 加载所选主站的 xxx 配置。
- **速度无(1..12)** 所选主站的通信速度。
- **快速数据交换开/关** 启用或禁用 FastExchange（仅适用于从属模块）

### (注 1)

如果您编写可能更适合 Arduino 和 ESP32 模块的 SelectModule，则也会识别 SelectMaster 命令。

# 使用 TextSlots 的命令 (第 2 部分)

## 选择 PIN 码

要选择 Pin 图，需要使用文本字符串，但在许多情况下，该字符串仅包含 1 到 12 之间的数字。

但是，当使用 Arduino 和 Esp32 模块时，引脚不按顺序编号，例如 22、35、28、60 等，在某些情况下它们是字母数字，例如 A0、A1、A3、D0、D1、D3 等。

因此，在所有情况下，Pin 的名称都显示为文本字符串，而不是数字。

## 用于选择 PIN 的命令

- 选择引脚 xxx                      文本字符串用于 Pin 名称

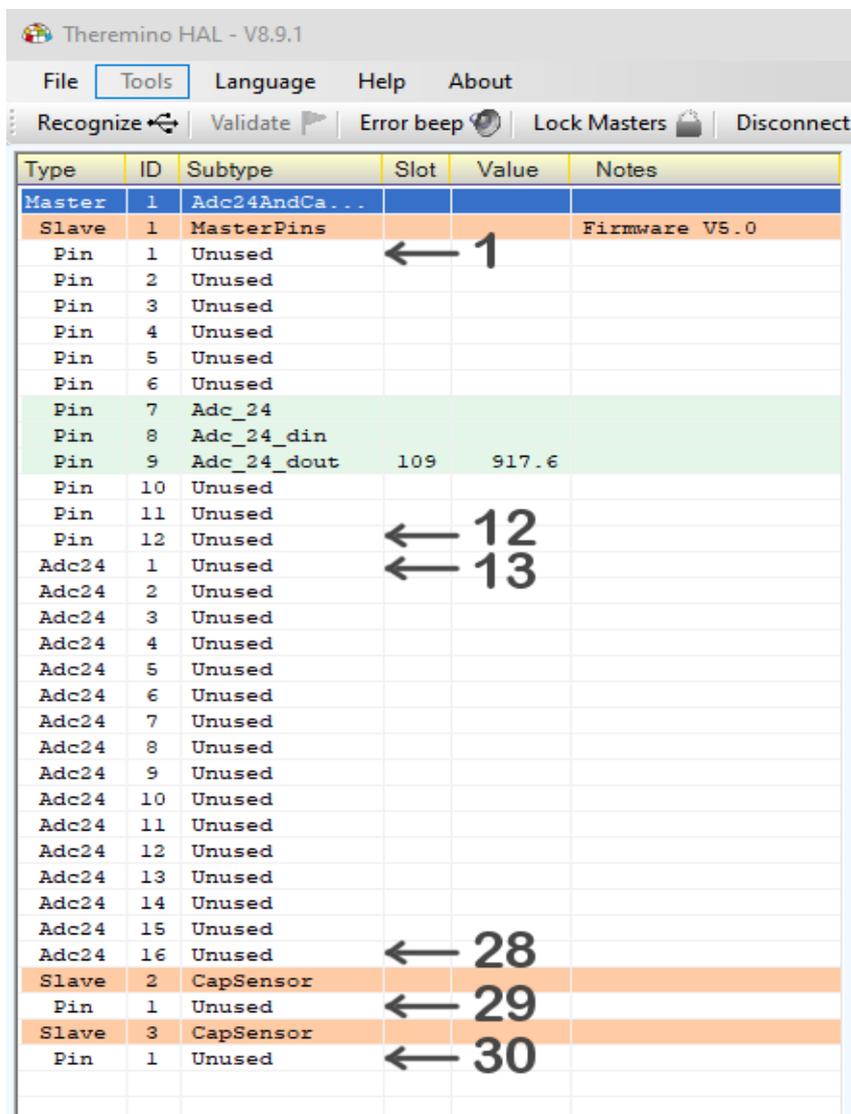
## ADC24 和 CAP-SENSOR 引脚的编号

当使用 Master 与 Theremino HAL 应用程序一起使用并连接 Adc24 和/或一个或多个 CapSensor 时，引脚编号与您想象的不同。

主引脚范围为 1 至 12。

然后继续编号  
表示 13、14 及后续的 Adc24。  
( 在本例中，它们的范围是 13 到 28 )

还有 CapSensors  
继续编号。  
( 在这种情况下，它们用 29 和 30 表示  
但如果没有 Adc24，它们将是 13 和  
14 )



Type	ID	Subtype	Slot	Value	Notes
Master	1	Adc24AndCa...			
Slave	1	MasterPins		1	Firmware V5.0
Pin	1	Unused	←		
Pin	2	Unused			
Pin	3	Unused			
Pin	4	Unused			
Pin	5	Unused			
Pin	6	Unused			
Pin	7	Adc_24			
Pin	8	Adc_24_din			
Pin	9	Adc_24_dout	109	917.6	
Pin	10	Unused			
Pin	11	Unused			
Pin	12	Unused	←	12	
Adc24	1	Unused	←	13	
Adc24	2	Unused			
Adc24	3	Unused			
Adc24	4	Unused			
Adc24	5	Unused			
Adc24	6	Unused			
Adc24	7	Unused			
Adc24	8	Unused			
Adc24	9	Unused			
Adc24	10	Unused			
Adc24	11	Unused			
Adc24	12	Unused			
Adc24	13	Unused			
Adc24	14	Unused			
Adc24	15	Unused			
Adc24	16	Unused	←	28	
Slave	2	CapSensor			
Pin	1	Unused	←	29	
Slave	3	CapSensor			
Pin	1	Unused	←	30	

# 使用 TextSlots 的命令 (第 3 部分)

## 引脚属性 (引脚属性面板)

- 设置引脚类型 xxx 设置当前所选引脚的类型。
- 设置槽位不(0..999) 设置所选引脚的插槽。
- 设置最大值不 为选定的引脚设置 MaxValue。
- 设置最小值不 设置所选引脚的 MinValue。
- 设置引脚类型 xxx 设置所选引脚的类型。
- 设置响应速度按钮开/关 启用或禁用 ResponseSpeed 按钮。
- 设置响应速度 否 (1..100)设置响应速度值。

## PWM 和 SERVO 引脚的属性 (PWM 属性和 SERVO 属性面板)

- 最大时间 否 设置所选 Pin 的最大时间。
- 最短时间 否 设置所选 Pin 图的最短时间。
- 对数响应开/关 启用或禁用对数响应按钮。

## STEPPER 类型引脚的属性 (步进属性面板)

- 最大速度 否 设置最大 sp. (mm/min) 对于所选引脚。
- 最大加速度 设置最大加速度 (mm/sec/s) 对于所选引脚。
- 每毫米步数 N 设置所选 Pin 图的最短时间。
- 链接到先前的开/关 启用或禁用链接到上一个按钮。

## PWM FAST 类型引脚的属性 (Pwm\_Fast 属性面板)

- PwmFast 频率 否 设置 PWM 频率。
- PwmFastDutyCycle 否 (0..1000) 设置占空比。
- 频率来自时隙开/关 从插槽启用或禁用频率控制。
- 插槽占空比开/关 从插槽启用或禁用占空比控制。

## CAP 类型引脚的属性 (触摸属性面板)

- 最小变化号(-1000..+500) 设置最小变化。
- 比例面积(-500..+1000) 设置比例面积。

## COUNTER 和 PERIOD 类型引脚的属性 (频率属性面板)

- 转换频率开/关 启用或禁用频率转换。
- 最大频率 否 设置最大频率。
- 最小频率 否 设置最小频率。

# 使用 TextSlots 的命令 ( 第 4 部分 )

## CapSensor 和 Usound 类型引脚的属性 ( CapSensor 和 Usound 属性面板 )

- **最大距离 否**(50..9999) 设置最大距离 ( 以毫米为单位 ) 。
- **最小距离号**(0..5000) 设置最小距离 ( 以毫米为单位 ) 。
- **地区编号** (0..9999) 以平方毫米为单位设置面积 ( 仅适用于 CapSensors ) 。
- **删除错误开/关** 启用或禁用 Usound 类型的错误检查

## Adc24 类型引脚的属性 ( Adc24 属性面板 )

- **针数 无**(0..16) 设置 PIN 码 ( 仅适用于 Masters , 不适用于 ESP32 ) 。
- **样品编号** (10..19200) 设置采样率。
- **过滤器 XXX** 设置过滤器类型 ( MaxSpeed、Fast、Medium 等.. )

## Adc24\_ch 和 Adc24\_cn\_b 类型引脚的属性 ( Adc24\_channel 属性面板 )

- **类型 XXX** 设置类型 ( 差分、伪差分、单端 ) 。
- **增益 N** 设置增益 ( 0、2、4、8、16、32、64、128 )
- **偏置开/关** 启用或禁用 1.65 伏偏置。

## 价值要求

- **获取 MasterName ( 注 1 )** 请求提供所选主人的姓名 ( 表格 ) 。
- **获取 FPS** 请求当前通信频率。
- **获取引脚类型** 输入所选 Pin 图的请求。

## 答案

- **好的** 命令执行成功。
- **错误文本** 它还包括生成错误的命令。
- **必填参数** 使用命令之一请求的值得到。

## (注 1)

即使您编写 GetModuleName , GetMasterName 命令也会被识别 , 这可能更适合 Arduino 和 ESP32 模块。

## 通过数字槽发出的命令

Theremino 系统应用程序或用户创建的其他应用程序可以使用数字槽进行通信，发送命令并从 HAL 接收数据。

例如，应用程序可以修改所有引脚的参数，重写配置文件，然后发送“识别”命令。或者应用程序可以检查实际连接了多少个主设备，发送识别命令，然后读取命令槽上的数量。或者，音乐应用程序可以通过发送“校准”命令来校准 CapSensors 或电容式按钮。

## 使用其他插槽代替插槽 0

通常命令槽为零，但您可能想在同一台 PC 上使用多个独立的应用程序。在这些情况下，每个应用程序将与其 HAL 一起驻留在单独的文件夹中，并使用“Lock Master”命令访问其主槽块。在这些情况下，可以为每个 HAL 分配不同的命令槽。对于命令，您可以使用任何插槽（从 0 到 999），但请注意不要将其分配给任何引脚。

要为命令槽指定非零的数字，请手动修改“Theremino\_HAL\_INI.txt”文件的最后一行。例如，要使用插槽 300，您可以编写：**命令槽=300**。注意不要删除“=”号。如果出现错误，则使用插槽 0，并且 HAL 重写 INI 文件中的正确行。

## 如何发送命令

目前定义了两个命令：

- ◆ 认出 S 我发送“NAN\_Recognize”，或数字“1”
- ◆ 校准 S 我发送“NAN\_Calibrate”，或数字“2”

无法发送特殊 NAN（非数字）数字的应用程序可以使用数字“1”和“2”来代替“NAN\_Recognize”和“NAN\_Calibrate”值。

为了安全起见，命令“1”和“2”之前必须有一个序列。该序列涉及两个数字（333 和 666），它们实际上是浮点数，具有七位精度，即 333.0000 和 666.0000。因此，ADC 或其他设备实际上不可能错误地发送该序列。

## 回复消息

响应和错误消息通过命令槽中的数字进行通信。

- ◆ -1 “识别”命令仍在运行。
- ◆ 0 未找到大师，大师列表为空。
- ◆ 从 1 开始 被认可的大师数量。
- ◆ NAN\_Master 错误 已连接的 Master 之一已停止通信。

## 数字命令槽 - 示例

要发送“识别”命令，请编写：

```
----- VbNet  
Slots.WriteSlot(0, NAN_Recognize)
```

```
----- CSharp  
Slots.WriteSlot(0, NAN_Recognize);
```

```
----- Theremino 脚本  
WriteSlot ( 0 , NAN_Recognize )
```

如上一页所述，某些应用程序（例如 Theremino 自动化），他们可能无法使用特殊号码南。如果不使用 NAN，前面的示例将变为：

```
----- VbNet  
插槽.WriteSlot(0, 333)
```

```
系统.Threading.Thread.Sleep(50)
```

```
插槽.WriteSlot(0, 666)
```

```
系统.Threading.Thread.Sleep(50)
```

```
插槽.WriteSlot(0, 1)
```

```
----- CSharp
```

```
插槽.WriteSlot(0, 333);
```

```
系统.Threading.Thread.Sleep(50);
```

```
插槽.WriteSlot(0, 666);
```

```
系统.Threading.Thread.Sleep(50);
```

```
插槽.WriteSlot(0, 1);
```

```
----- Theremino 自动化
```

```
插槽 0 = 333
```

```
等待秒 0.05
```

```
插槽 0 = 666
```

```
等待秒 0.05
```

```
插槽 0 = 1
```

```
----- Theremino 脚本
```

```
写槽 (0, 333)
```

```
线程.线程.睡眠(50)
```

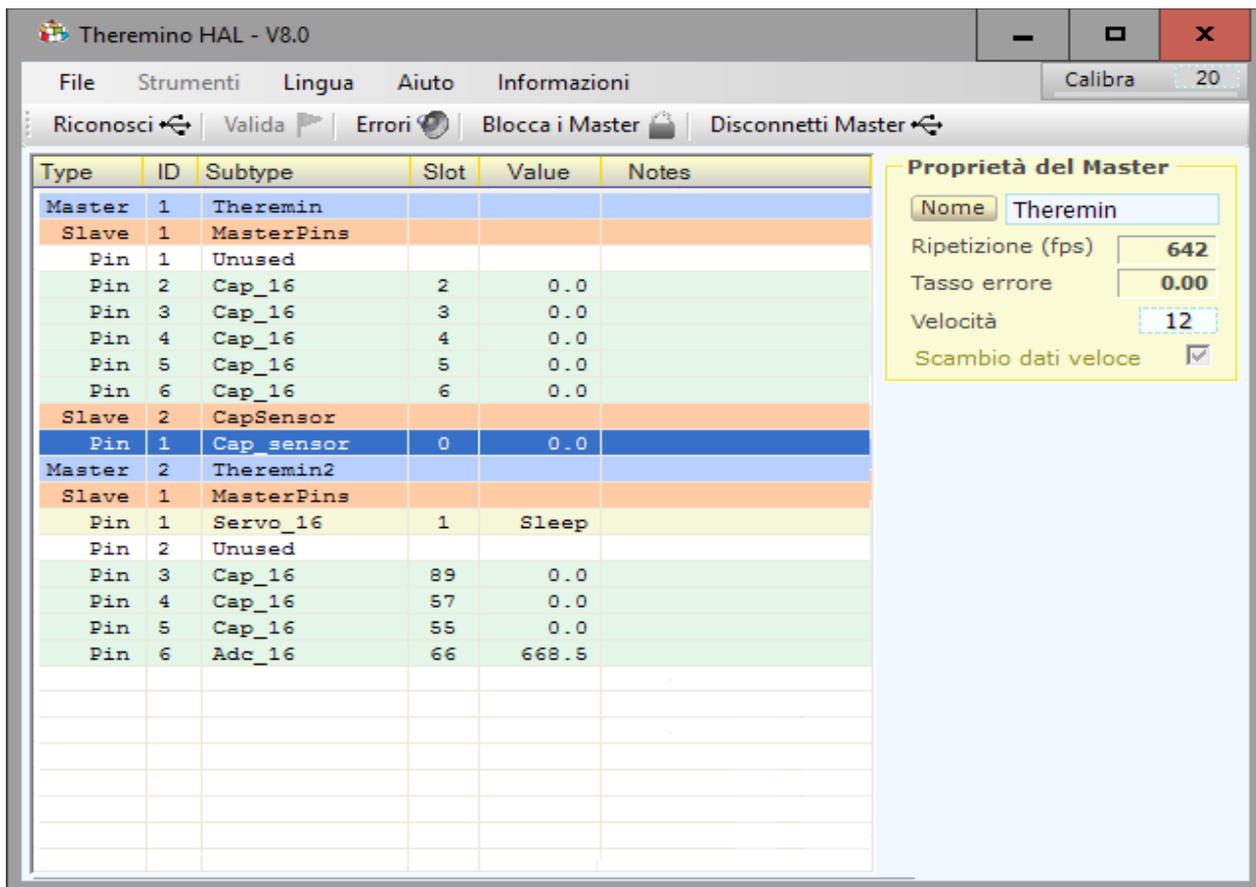
```
写入槽 (0, 666)
```

```
线程.线程.睡眠(50)
```

```
写槽 (0, 1)
```

50 毫秒的暂停用于为 HAL 提供读取 Slot 的时间。

## HAL 的颜色



### 配色方案可帮助您识别组件及其配置

- 浅绿色表示输入

- 浅黄色表示输出

第一位大师 (名为 Theremin) 提供:

称为 'Master Pins' 的虚拟从属设备

六个 "Pin", 其中只有第一个 "未使用", 其他配置为 'Cap\_16'

"CapoSensor" 型从站

只需一个 "Pin" 配置为 "帽传感器" 和 "已选择"

第二位大师 (名为 Theremin2) 提供:

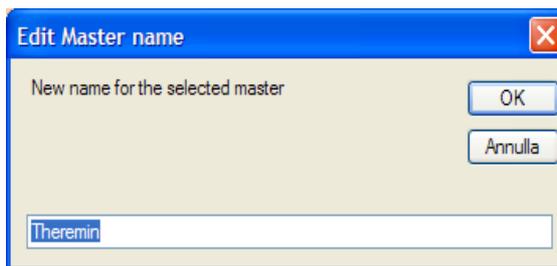
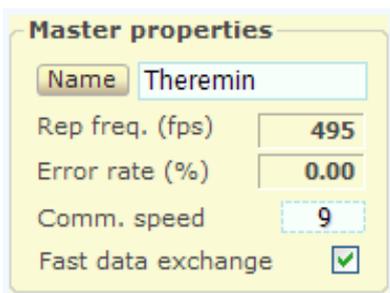
称为 'Master Pins' 的虚拟从属设备

一个 "别针" 配置为 'Servo\_16'

配置为 "未使用" 的 "Pin"

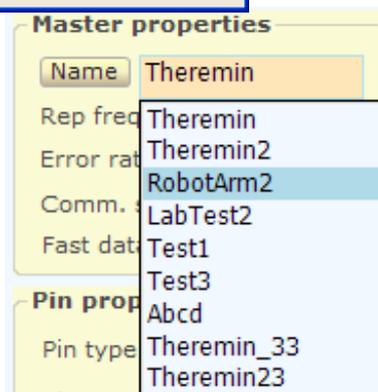
四个 "针" 配置为 'Cap\_16'

# 大师的属性 - 名字



可以通过两种方式更改所选 Master 的名称：

- ◆ 按“名称”按钮，然后更改其名称。
- ◆ 单击名称框并从下拉菜单中选择不同的配置。



大师的名字它被写入硬件模块并用于在重新连接时识别它。

刚刚连接的新Master称为“NoName”。最好养成立即给它起一个不同的名称的习惯，以将其与其他名称区分开来。

在大师的名字中，字母的“大小写”（大写或小写）不计算在内。

如果Master数据库中有两个同名的Master，则第一个Master的配置将用于两者。因此，为每个大师指定不同的名称非常重要（除非您想要有与主大师同名的备用大师）

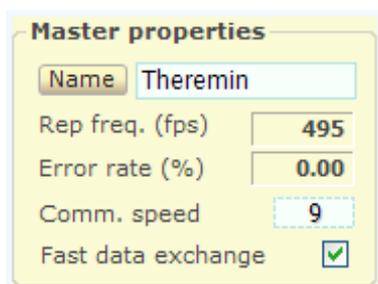
Masters 始终按字母顺序列出，因此，如果您更改 USB 端口，Masters 的顺序不会改变。

HAL 程序在断开、更换和恢复组件时几乎总是设法使用正确的配置，但如果您使用不同的计算机或使用另一个 HAL 应用程序（位于单独的文件夹中 - 因此具有单独的参数）或使用单独的参数更改 Master 的名称，或者在其他困难和复杂的情况下，配置和硬件之间的一致性就会丢失。

如果对齐丢失，您应该手动恢复配置，一次一个 Pin，但专家可以编辑配置文件，并可能完全复制该文件或仅部分配置，从一个 HAL 应用程序到另一台计算机上的另一个应用程序，或者在另一个文件夹中。

当配置无效时，修改Master的名称并不会修改配置文件，而只是修改硬件中写入的名称，因此可以修改Master的名称，直到与配置中的正确名称匹配为止。

# 大师的属性——沟通



- 每秒通信次数
- 串行线路上的错误百分比 (通常为零)
- 串行通信速度 (从 1 Kilo Baud 到 4 Mega Baud)
- 选择“单一”或“快速”通信类型 (注 1)

每秒通信次数“fps”通常应为 480 到 500，如果针对物理从站和引脚的串行通信超过一定数量的字节并且传输速度较低，则该数字会减少。

对于许多应用，例如具有速度的电容式按键，最好保持 fps 尽可能高，至少 400 或 450。

## 增加“fps”数量 (连接从站) :

- 增加“通讯速度” (与串行连接的长度兼容)
- 使用“快速数据交换” (最大字节数减少到64，但速度提高)
- 拆分串行线路并将关键引脚连接到负载较少的线路上
- 拆分串行线并将关键引脚连接到一个或多个没有串行的主设备上。
- 通过将所有可能的引脚配置为“未使用”来减少使用的字节数。
- 通过将所有不需要高分辨率的引脚配置为 8 位来减少使用的字节数

## 调整“fps”数量 (没有连接从站) :

通过“通信速度”值，即使没有连接到串行线路的从站，也可以调整“fps”更新速度。在这种情况下，值“Rep freq.(fps)”指的是仅用于通过 USB 进行通信。

要提高响应速度，最好将交换频率增加到最大，然后将“通讯速度”设置为“12”。但对于许多应用程序来说，每秒 100 次交换就足够了，因此您通常可以将“通信速度”从 8 调整到 10，从而减少 CPU 的负载。

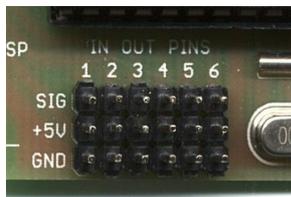
# 奴隶与别针

Type	ID	Subtype	Slot	Value
Master	1	Theremin		
Slave	1	MasterPins		
Pin	1	Unused		
Pin	2	Cap_16	2	
Pin	3	Cap_16	3	
Pin	4	Cap_16	4	
Pin	5	Cap_16	5	
Pin	6	Cap_16	6	
Slave	2	CapSensor		
Pin	1	Cap_sensor	0	
Master	2	Theremin1		
Slave	1	MasterPins		
Pin	1	Cap_16		

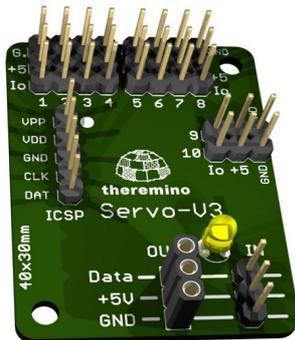
奴隶们，这里用红色箭头表示，没有任何调整，它们只是针容器。从站通常有 1 至 12 个引脚

针脚它们都是相同的，并且可以通过多种不同的方式进行配置。

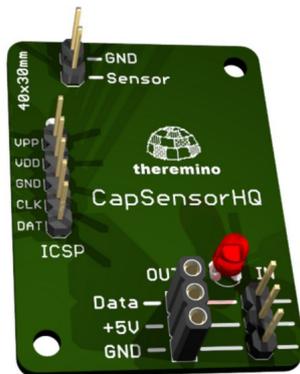
主控模块它有一个内置从站（称为虚拟从站），根据其固件版本提供六到十二个引脚。



“伺服”型“从属”模块他们有 10 个 Pin。



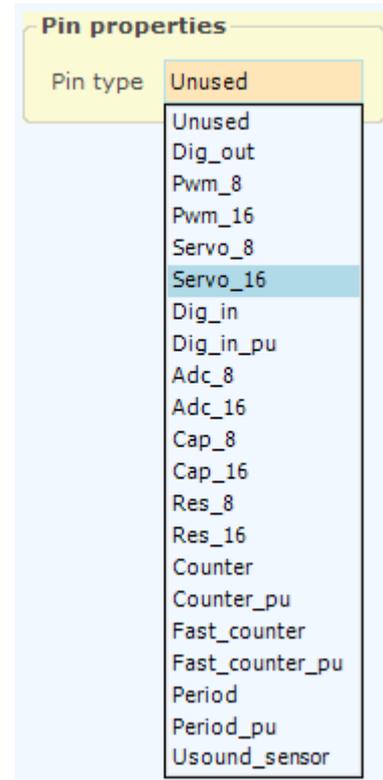
“盖传感器”类型“从”模块它有一个高质量的“Pin”（它可以听到数米外的大型物体的运动）。新的“HS”固件还具有快速响应（< 1ms）。



# 引脚类型

引脚可以配置为：

- ◆ 不曾用过
- ◆ 数字输出
- ◆ 脉宽调制输出 ( 250Hz )
- ◆ PwmFast 输出 ( 250Hz 至 5 MHz )
- ◆ 伺服控制输出
- ◆ 步进电机输出
- ◆ 数字输入
- ◆ 电位器和传感器的 ADC 输入
- ◆ 电容式按键输入
- ◆ 电阻传感器输入
- ◆ 计数、频率和周期输入
- ◆ 特殊传感器的输入
- ◆ CapSensor 模块的输入
- ◆ 两相编码器输入
- ◆ [Adc24 的控制通道](#) ( 引脚 7、8、9 )
- ◆ [Adc24 通道](#) ( Adc24 引脚 1 至 16 )



大师的特殊徽章：

- ◆ 引脚 1 至 6 是最灵活的，您几乎可以将它们配置为任何类型。
- ◆ 引脚 7、8 和 9 除其正常功能外，还用于连接 Adc24 模块。
- ◆ 引脚 11 和 12 不能配置为步进或 PwmFast

特殊从属引脚：

- ◆ “Servo”从站的引脚 9 和 10 不能配置为 ADC、CAP 和 RES。
- ◆ 引脚 8 在“伺服”从站中，它是唯一可以配置为“快速计数器”的从站。
- ◆ 引脚 9 在“伺服”从站中，它是唯一可以配置为“周期”和“声音传感器”的从站。
- ◆ CapSensor 从机的单个引脚只能配置为“未使用”或“Cap 传感器”。

用作 ADC 和 CAP 的最佳引脚：

- ◆ 用作 ADC 和 CAP 的最佳引脚是引脚 3、4、5、6
- ◆ 引脚 7 和 8 具有双倍漏电流和电容 ( ADC 和 CAP 的第二选择 )
- ◆ 引脚 1 和 2 具有四倍的漏电流和容量 ( ADC 和 CAP 的第三选择 )

所有引脚均可配置为“未使用”，这可以让你减少通过的字节数串行线和 USB 上，并最大化每秒交换的数量。

8 位和 16 位之间的选择，可用于多种类型的引脚，允许您拥有最大分辨率 ( 16 位 ) 或较低分辨率 ( 8 位 )，但可以节省更多位，从而获得最大通信速度。

那些做引体向上的家伙其名称以“\_pu”结尾，可让您轻松连接开关、按钮和集电极开路器件，而无需添加外部电阻 ( 典型上拉电流 = 250 uA )。

对于 Pin Stepper 和 PwmFast 需要以下内容：具有固件 3.0 和 HAL 版本 5.0 ( 或更高版本 ) 的主站  
对于编码器类型引脚，需要以下内容：具有固件 4.0 和 HAL 版本 5.3 ( 或更高版本 ) 的主站  
要连接 Adc24 模块，您需要：具有固件 5.0 和 HAL 版本 6.5 ( 或更高版本 ) 的主站

# 了解有关插槽、引脚和模块的更多信息

有关 Pin 图的更多信息：

[www.theremino.com/technical/pin-types](http://www.theremino.com/technical/pin-types)

各个模块及其引脚的特性：

[www.theremino.com/hardware/devices](http://www.theremino.com/hardware/devices)

步进销的信息：

[www.theremino.com/hardware/outputs/motors](http://www.theremino.com/hardware/outputs/motors)

连接Adc24模块的信息：

[www.theremino.com/hardware/adapters#adc24](http://www.theremino.com/hardware/adapters#adc24)

模块的电气特性：

[www.theremino.com/technical/schematics](http://www.theremino.com/technical/schematics)

可连接的输出设备（执行器）：

[www.theremino.com/hardware/outputs](http://www.theremino.com/hardware/outputs)

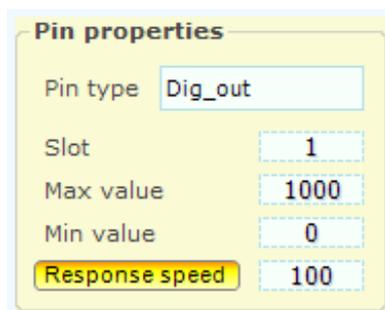
可连接的输入设备（传感器）：

[www.theremino.com/hardware/inputs](http://www.theremino.com/hardware/inputs)

关于使用表单的博客和技巧：

[www.theremino.com/blog/Master-and-Slaves](http://www.theremino.com/blog/Master-and-Slaves)

## 所有 Pin 通用参数



Pin properties	
Pin type	Dig_out
Slot	1
Max value	1000
Min value	0
Response speed	100

“老虎机”指示写入或读取数据的位置。有一千个插槽，编号从 0 到 999，可以从 Theremino 系统的所有引脚和所有应用程序读取或写入它们。

*注意：许多应用程序和许多引脚可以从同一个插槽读取，但必须避免在同一插槽上配置多个写入引脚，这样做不会破坏任何东西，但你会得到不确定的结果。*

“最大值”通常设置为 1000，表示 Pin 处于最大值时必须具有的值。

“最小值”通常它保持为零，并指示引脚在最小值时必须具有的值。

通过用 0 和 1000 以外的值调整 Max 值和 Min 值，可以获得任意缩放和校准比率。如果两个值交换（最小值大于最大值），则刻度会翻转，这对于反转伺服控制的运动或翻转以相反方式起作用的传感器的读数非常有用。

「反应速度」调整 IIR（无限脉冲响应）滤波器，以实现噪声和响应速度之间的最佳折衷。当值为 100 时，过滤器被禁用，您将获得最大响应速度；当值为 1 时，您将获得最大过滤（消除任何闪烁），但响应非常慢（大约一秒）。通常使用值 30，它提供良好的过滤并且响应相当快。

如果按钮「反应速度」按下时，IIR 滤波器会适应变化，以便在变化较大时获得更大的响应性，在变化较小时获得更大的阻尼。结果，获得了良好的图形稳定性，而不会过多地影响稳定时间。

按下“响应速度”时，某些传感器信号可能会出现故障。对于产生围绕高基线值变化较小的信号的传感器来说尤其如此。在这种情况下，信号未达到最终值，或者到达速度非常慢。如果您遇到此情况，请禁用「反应速度」。

# “输出”中的引脚类型 --> 挖掘/Pwm/ 伺服

## ◆ 挖出

Pin properties	
Pin type	Dig_out
Slot	1
Max value	1000
Min value	0
Response speed	100

这种类型的引脚提供数字输出。

从插槽到达的值，限制在“最小值”和“最大值”之间并通过“响应速度”过滤，与“最小值”和“最大值”之间的中间值进行比较，如果超过它，则引脚点亮否则它会关闭。

该引脚只能承受 0 伏（关闭）和 3.3 伏（开启）电压，并且输出电流限制为大约 +/- 10 mA

## ◆ Pwm\_8 和 Pwm\_16

Pin properties	
Pin type	Pwm_16
Slot	1
Max value	1000
Min value	0
Response speed	100

这种类型的引脚提供 PWM（脉冲宽度调制）输出

从插槽到达的值，限制在“最小值”和“最大值”之间并按“响应速度”过滤，转换为宽度在“最小时间 ( )”和“最大时间 (uS)”之间的脉冲

脉冲重复时间为 4000 uS (250 Hz)，足以以可变强度打开 LED。对于需要真正可变电压的用户，会增加一个低通滤波器，通常由电阻和电容组成。

PWM properties	
Max time ( uS )	4000
Min time ( uS )	0
Logarithmic response	<input type="checkbox"/>

该引脚提供 0 伏（关）和 3.3 伏（开）电压之间的脉冲，输出电流限制在大约 +/- 10 mA

## ▲ Servo\_8 和 Servo\_16

Pin properties	
Pin type	Servo_16
Slot	1
Max value	1000
Min value	0
Response speed	100

这种类型的引脚直接驱动伺服控制。

从插槽到达的值，限制在“最小值”和“最大值”之间并按“响应速度”过滤，转换为宽度在“最小时间 (uS)”和“最大时间 (uS)”之间的脉冲

脉冲重复时间对于在最小和最大时间之间旋转约 180 度的正常建模伺服系统来说是足够的。

Servo properties	
Max time ( uS )	2500
Min time ( uS )	500

该引脚提供 0 至 3.3 伏的电压，足以满足所有由 3 至 6 伏供电的普通伺服系统，并且电流足以并行驱动数十个伺服系统。

# “输出”中的引脚类型 --> 踏步机

每个步进电机需要两个物理引脚，一个用于步骤，一个用于方向。微控制器允许您根据需要定位引脚，但我们决定通过固定位置来限制 Stepper 和 Stepper\_Dir 类型引脚的混乱（位置为：1-2、3-4、5-6、7-8、9-10）。

步进销从插槽中读取一个值，该值只是以毫米为单位的目的地。最简单的应用程序可以指定一个远程目的地，并让固件完成这一切。要求更高的应用程序可以自行计算路线并发送频繁的中间目的地。通过这种技术，应用程序可以控制工作速度（进给），并精确地建立路径，即使在多个维度上也是如此。为了获得平稳的移动，每秒 20 个目的地就足够了（对于最苛刻的应用程序最多 50 个）。

要反转轴的运动方向，只需交换“1000 表示毫米”和“0 表示毫米”框中的值“1000”和“0”即可。

## 步进式插针具体参数

Stepper properties	
Max sp. (mm/min)	900
Max acc. (mm/s/s)	100
Steps per mm	200

**最大速度**-这是最大速度，以毫米每分钟为单位。固件不断检查软件发送的目的地。如果软件对电机要求过高，固件会限制其速度，以防止丢失步数。提高该值，直到您看到电机失去步数（发出尖锐的噪音并停止），然后将其降低 20.50%，以返回到安全区域。在负载下或通过手动制动发动机重复测试，以确保有一定的余量。

**最大加速度** -这是最大加速度（和减速度），以毫米/分钟为单位。固件不断检查软件发送的目的地。如果软件对电机要求过高，固件会限制其加速度，以防止其丢失步数。提高该值，直到您看到电机在改变方向时失去步数（发出尖锐噪音并停止），然后将其降低 20.50%，以返回安全区域。在负载下或通过手动制动发动机重复测试，以确保有一定的余量。

**每毫米步数** -在这里，您必须设置电机旋转一圈的步数，乘以控制器中设置的微步，然后除以电机旋转一圈产生的毫米数。如果每转产生一毫米的位移，且电机每转有两百步，并且不使用微步，则设置该值： $200 \text{ (每转步数)} \times 1 \text{ (微步)} / 1 \text{ (每转毫米)} = 200$  如果使用十六个微步，则设置值： $200 \text{ (每转步数)} \times 16 \text{ (微步)} / 1 \text{ (每转毫米)} = 3200$

**链接到之前的内容**——某些版本的 HAL 具有此已弃用的命令，该命令将不会被实现。

**Stepper\_Dir 类型的引脚**他们没有需要调整的参数。它们只是物理输出引脚的占位符，用于设置电机的方向。不必使用这些引脚写入插槽的值，但某些应用程序可能会发现它很有用。写入槽中的值是距目的地的距离，以毫米为单位（最多为千分之一毫米）。此信息可用于诊断目的，或用于必须满足指定容差的算法。有了这些信息，软件就可以在闭环中工作并始终以最大速度运行。通过不断检查每个电机距目的地的距离，该软件可以在需要时准确地减速，而无需对速度、轨迹和加速度进行复杂的计算。

# “输出”中的引脚类型 --> Stepper 和 Stepper\_Dir

## 步进类型引脚的值

报告从插槽读取的值（“1000 表示毫米”和“0方法mm”）并转换为 0 到 1 之间的值。

如果设置“1000 表示毫米”= 1000 且“0方法mm”= 0，则不执行比例转换，并且来自插槽的值被视为“毫米”。

从这里开始，值始终以毫米为单位。“零”表示零毫米，“一”表示 1000 毫米。该值不限于 0 到 1 之间，而是介于 20 亿正步和 20 亿负步之间。如果您使用“每毫米步数 = 200”，则限制为：+10 Km 和 -10 Km。

然后使用 IIR 滤波器（线性或指数）对该值进行滤波，可通过“响应速度”进行调整。滤波器的输出值称为“Filtered”

发送到硬件的最终值是步数（预先乘以“每毫米步数”值）并代表“目的地”。

特殊值 NAN\_Reset 具有重置轴的特殊含义。当您在步进类型引脚的插槽中写入复位时，电机立即停止。随后，将写入槽中的第一个值将代表“参考零”值。NAN\_Reset 在 Theremino 自动化中作为“重置”提供，或者在新类“ThereminoSlots”中可用，可通过 Theremino 自动化源下载。

## Stepper\_Dir类型Pins的值

每个 Stepper\_Dir 引脚始终与一个步进器引脚关联。

这是一个特定的引脚，既是输出又是输入。为电机提供电气方向信号（因此是输出），但同时它充当输入并向软件提供信息。

硬件读取的原始值是到达指定“目的地”时缺少的步数（正或负）。HAL 应用程序计算毫米（和分数），将原始值除以“每毫米步数”值。最后，这个以毫米为单位的值被写入槽中，并且可以由其他应用程序（通常是 CNC 应用程序）读取。

CNC 应用程序知道剩余距离和目的地（由其自身指定），可以通过简单的减法计算出电机的实际位置。通过始终了解每个电机的位置，控制算法得到简化，操作也更加精确。

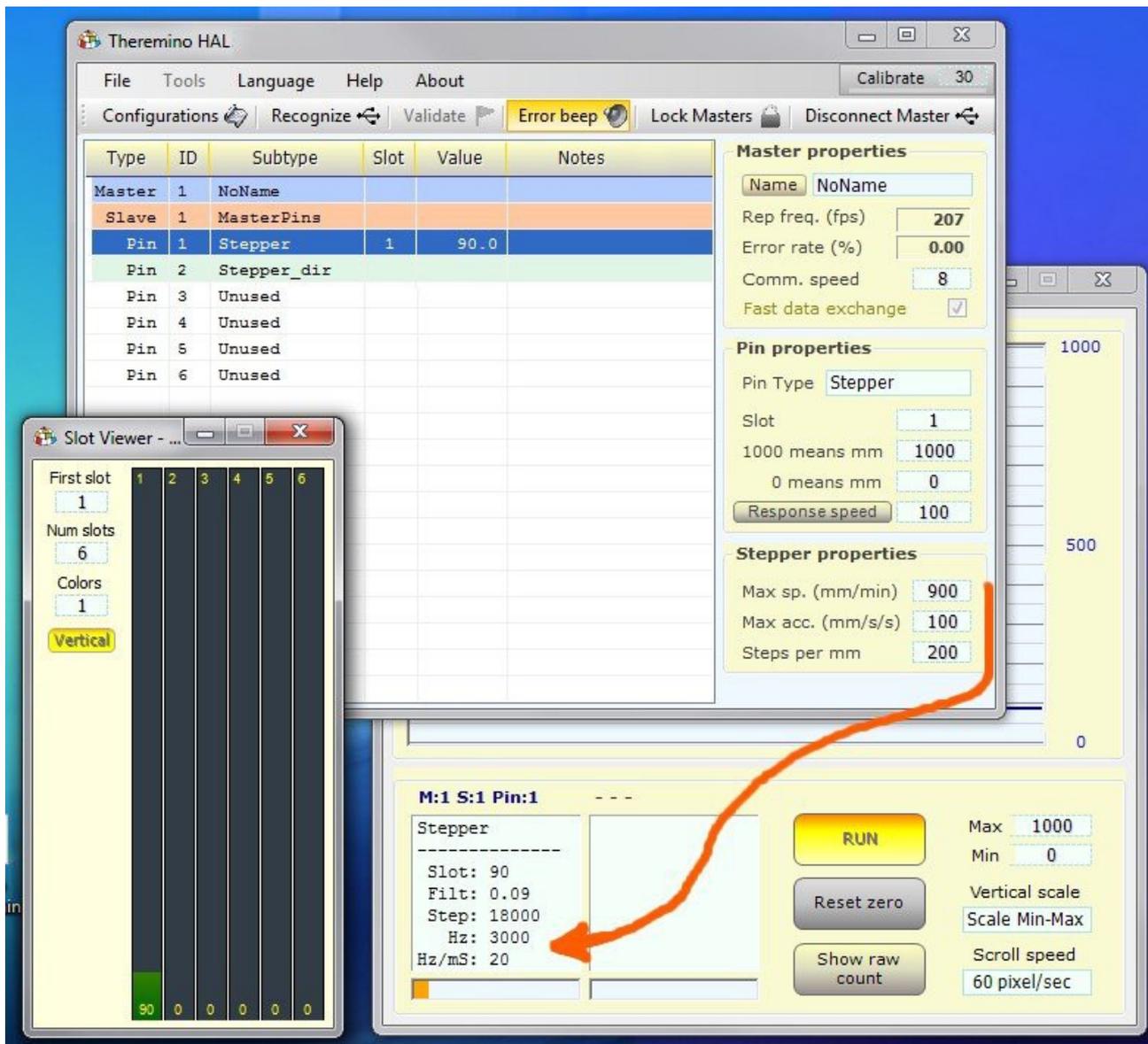
-----

更详细的信息关于步进电机的步进类型引脚和驱动器，在这些页面中：

<http://www.theremino.com/hardware/outputs/motors>

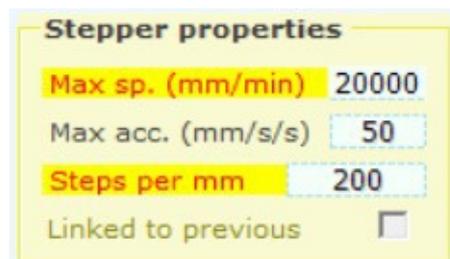
<http://www.theremino.com/technical/pin-types>

## “输出”中的引脚类型 --> 步进器细节



要打开详细信息窗口，请双击引脚线（步进类型）。在第二个窗口的底部，您可以阅读所选 Pin 图的详细信息。

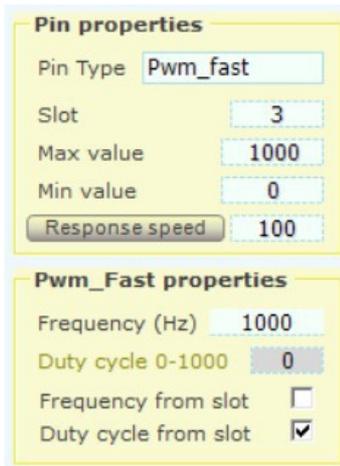
引脚详细信息显示最大速度和每毫米步数（以 Hz 为单位）（每秒步数）的乘积。这些值在测试过程中非常有用，可以确定要使用多少微步。在某些情况下，了解原始目标（以步为单位）而不是以毫米为单位的目標可能会很有用。



最大步进频率为 65000 Hz，如果超过它，您会收到“MaxSpeed”和“Steps per mm”框的警告，它们的颜色是黄色和橙色。

在这种情况下，必须降低“MaxSpeed”。或者，您可以通过减少驱动器上的微步调节来减少“每毫米步数”。

# “输出”中的引脚类型 --> 快速脉宽调制



可以生成的最小频率为 245 Hz，最大频率约为 5.3 MHz。占空比从零（始终为低输出信号）到 100%（始终为高输出信号）。

到启用“Frequency from Slot”，传入值设置频率。从插槽到达的值（通常在 0 到 1000 之间）首先被过滤，然后转换为“最小值”和“最大值”之间的频率值。

通过启用“插槽占空比”，输入值可设置低信号和高信号之间的时间比率。从 Slots 到达的值，通常在 0 到 1000 之间，首先被过滤，然后相乘或相除，修改“最小值”和“最大值”。通常您设置 Min=0 / Max=1000 并调整占空比，提供从 0 到 1000 的值。

调整的粒度取决于设置的频率。

- ◆ 1000 Hz 时，占空比精度为 16 位（误差：0.0015%），频率精度为 14 位（误差：0.006%）
- ◆ 16 KHz 时，占空比精度为 12 位（误差：0.024%），频率精度为 10 位（误差：0.1%）
- ◆ 在 1 MHz 时，占空比精度降至仅 6 位（误差：1.5%），频率精度降至仅 4 位（误差：6%）

由于粒度的原因，最高频率为：5.333 MHz / 4 MHz / 3.2 MHz / 2.666 MHz / 2.286 MHz / 2 MHz / 1.777 MHz / 1.6 MHz / 1.454 MHz / 1.333 MHz / 1.231 MHz / 1.066 MHz / 1 MHz

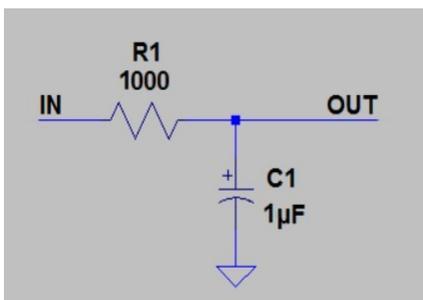
有关 Pin 图的更多信息，请阅读此页面：

<http://www.theremino.com/technical/pin-types#precision>

-----

通过 PwmFast 型引脚和简单的适配器，您可以获得模拟输出电压，可在 0 至 3.3 伏之间精确调节。

将频率设置为 15 KHz 并启用“DutyCycle from Slot”。



有关适配器的更多信息，请阅读此页面：

<http://www.theremino.com/hardware/adapters>

# “输入”中的引脚类型 <--挖掘/ADC/上限/Res

## ◆ Dig\_in 和 Dig\_in\_pu

Pin properties	
Pin type	Dig_in
Slot	1
Max value	1000
Min value	0
Response speed	100

这种类型的引脚提供数字输入。

使用施密特触发器读取电压值，低阈值 = 1 伏，高阈值 = 2 伏，并转换为开关信息，最终成为“最大值”和“最小值”。该值最后经过“响应速度”过滤后写入Slot。过滤产生中间值并且大约与On和Off之间的时间比率成正比

## ◆ Adc\_8 和 Adc\_16

Pin properties	
Pin type	Adc_16
Slot	1
Max value	1000
Min value	0
Response speed	30

这种类型的引脚提供模拟输入。

从 0 伏到 3.3 伏的电压值转换为“最小值”和“最大值”之间的数字。该值最后经过“响应速度”过滤后写入Slot。滤波可减少输入信号中的噪声，但会减慢响应速度。值 30 代表速度和噪声之间的良好折衷。

## ◆ Ch\_8 和 Ch\_16

Pin properties	
Pin type	Cap_16
Slot	1
Max value	1000
Min value	0
Response speed	30

这种类型的 Pin 允许您读取简单的按键，就像使用 Makey Makey (<http://vimeo.com/60307041#>) 但具有卓越的性能。

( 按键不是电阻式的，而是电容式的，因此只需触摸它们即可调整工作，无需接触，通过绝缘体，无需额外的接地线 )

除了经典的MakeyMakey式操作外，还可以获得渐进式控制，如“滑块”光标、由按键速度或原始电容值的读取决定的“表情”控制，例如湿度传感器。

Touch properties	
Min variation	10
Proportional area	0

有关这些按钮的更多信息，请参见第 16、17、18、19 和 20 页

## ◆ Res\_8 和 Res\_16

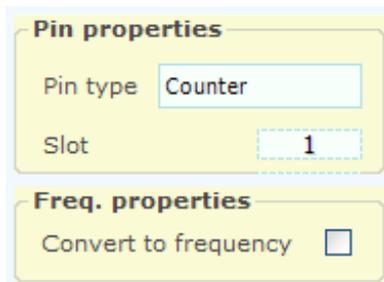
Pin properties	
Pin type	Res_16
Slot	1
Max value	1000
Min value	0
Response speed	30

这种类型的引脚允许您读取 0 到 50 Kohm 之间的电阻值。对于仅使用两根线读取电位计非常有用。通过不使用电源，可以消除由 USB 5 伏电压引起的干扰，无需添加稳压器，也无需连接到主设备特殊引脚上已调节的 3.3 伏电压。

六位研究实验与经典山达基测量罐产生了有趣的结果。

## “输入”中的引脚类型 <-- 柜台

### ◆ 计数器和Counter\_pu

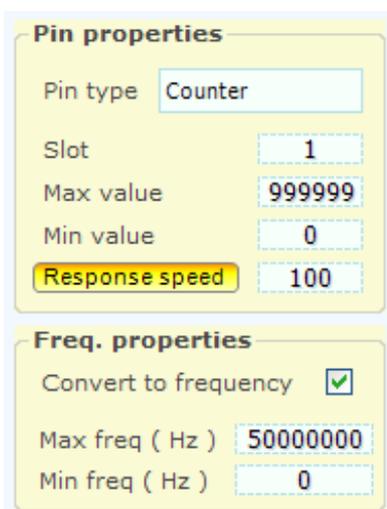


The screenshot shows the 'Pin properties' dialog box for a 'Counter' pin type. The 'Pin type' is set to 'Counter' and the 'Slot' is set to '1'. The 'Freq. properties' section has the 'Convert to frequency' checkbox unchecked.

所有引脚都可以编程为 Counter 或 Counter\_pu，但最大计数速度非常有限，约为几 KHz，具体取决于微控制器上的负载和信号的占空比。

如果您需要更高的速度，您应该使用 FastCounters。

### ◆ 带有“Freq”选项的 Counter 和 Counter\_pu



The screenshot shows the 'Pin properties' dialog box for a 'Counter' pin type. The 'Pin type' is set to 'Counter' and the 'Slot' is set to '1'. The 'Max value' is set to '999999' and the 'Min value' is set to '0'. The 'Response speed' is set to '100'. The 'Freq. properties' section has the 'Convert to frequency' checkbox checked, and the 'Max freq ( Hz )' is set to '50000000' and the 'Min freq ( Hz )' is set to '0'.

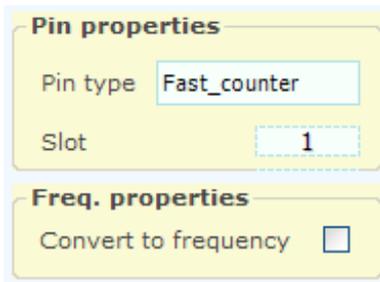
编程为 Counter 或 Counter\_pu 的引脚可以从计数器转换为频率计。

然后将限制在“Min freq”和“Max freq”之间的频率值在“Min vale”和“Max value”之间进行比较，并通过“Response speed”进行过滤，最后发送到Slot。

“Counter”和“Counter\_Pu”类型引脚使用 16 位进行数据传输。

# “输入”中的引脚类型 <-- 快速计数器

## ◆ Fast\_counter 和 Fast\_counter\_pu



Pin properties

Pin type: Fast\_counter

Slot: 1

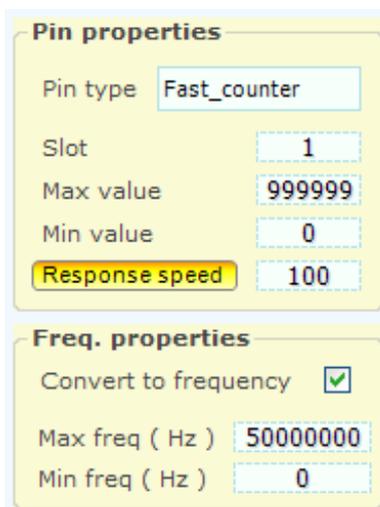
Freq. properties

Convert to frequency:

只有部分引脚可以编程为 Fast\_counter 或 Fast\_counter\_pu

另一方面，它们的最大计数速度为 50 MHz (对于最大计数速度占空比必须为 50%)

## ◆ Fast\_counter 和 Fast\_counter\_pu 带有“Freq”选项



Pin properties

Pin type: Fast\_counter

Slot: 1

Max value: 999999

Min value: 0

Response speed: 100

Freq. properties

Convert to frequency:

Max freq ( Hz ): 50000000

Min freq ( Hz ): 0

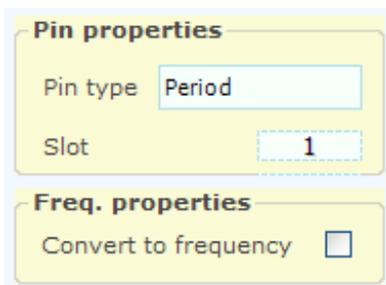
编程为 Fast\_counter 或 Fast\_counter\_pu 的引脚可以从计数器转换为频率计。

然后将限制在“Min freq”和“Max freq”之间的频率值在“Min value”和“Max value”之间进行比较，并通过“Response speed”进行过滤，最后发送到Slot。

引脚类型“Fast\_counter”和“Fast\_counter\_pu”它们使用 16 位进行数据传输。

## “输入”中的引脚类型 <-- 时期

### ◆ 期间和期间\_pu

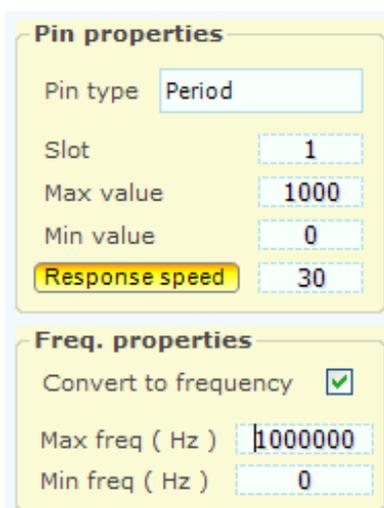


The screenshot shows the 'Pin properties' dialog box. Under 'Pin properties', 'Pin type' is set to 'Period' and 'Slot' is set to '1'. Under 'Freq. properties', the 'Convert to frequency' checkbox is unchecked.

这种类型的引脚测量重复波形的周期，从上升到上升，最长周期约为 260 秒。

在 0C 至 50C 的环境温度范围内，分辨率为半微秒，精度为 +/- 1%

### ◆ 带有“Freq”选项的Period和Period\_pu



The screenshot shows the 'Pin properties' dialog box with 'Freq. properties' expanded. 'Pin type' is 'Period', 'Slot' is '1', 'Max value' is '1000', 'Min value' is '0', and 'Response speed' is '30'. Under 'Freq. properties', 'Convert to frequency' is checked, 'Max freq ( Hz )' is '1000000', and 'Min freq ( Hz )' is '0'.

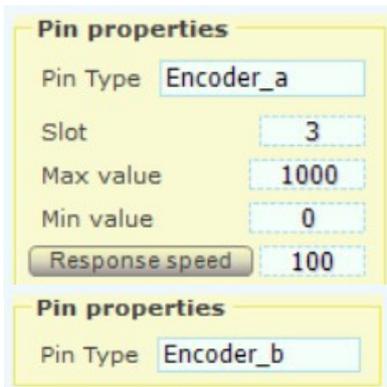
编程为Period 或Period\_pu 的引脚可以从计数器转换为频率计。

该技术允许您以非常高的分辨率测量非常低的频率（高达约十分之一赫兹）。

然后将限制在“Min freq”和“Max freq”之间的频率值在“Min value”和“Max value”之间进行比较，并通过“Response speed”进行过滤，最后发送到Slot。

引脚类型“Period”和“Period\_pu”使用 32 位用于数据传输。

## “输入”中的引脚类型 <-- 编码器



主模块的所有引脚都可以编程为编码器。

对于每个编码器，需要两个输入引脚：Encoder\_A 和 Encoder\_B，或 Encoder\_A\_Pu 和 Encoder\_B\_Pu。

这对输入“正交”读取编码器的两个相位。编码器计数写入与“Encoder\_A”引脚关联的插槽。

“Encoder\_A”类型的每个引脚使用 16 位进行数据传输，而“Encoder\_B”类型的引脚只是占位符，不发送数据。

-----

编码器像电位计一样读取引脚的角度位置，但圈数不受限制。

有类似于小型电位器的编码器（最著名的是下图中的 KY-040）。这些型号是机械式的，每转提供 18、20 或 24 个脉冲，具体取决于制造商。固件从这些脉冲中获取每转 72、80 或 96 个角位置。



最大计数速度限制为大约 10 KHz，并且取决于微控制器上的负载。因此，建议仅使用每转几步的编码器，或者限制旋转速度或降低旋转速度。

编码器生成范围从 0 到 65535（16 位）的计数。当计数超过 65535 时，数字重新从零开始。该系统允许许多彼此异步的应用程序读取渐进数而不会丢失计数。

有关 Pin 图的更多信息，请阅读此页面：

<http://www.thereмино.com/hardware/inputs/sensors#encoders>

# “输入”中的引脚类型 <-- Adc24

Adc24 模块连接到主机的引脚 7、8 和 9。激活模块选择引脚 7 线并将其引脚类型设置为“Adc24”。

Pin	7	Adc_24		
Pin	8	Adc_24_din		
Pin	9	Adc_24_dout	9	133.3

当 Adc24 工作时，引脚 9 ( Adc\_24\_dout ) 上读取的值与采样速度成比例增长 ( 使用 100 sps 和 MaxSpeed 滤波器，该值每秒增长 100 个值 )。

The image shows a series of configuration panels for the Adc24 module. The first panel is titled 'Pin properties' and shows 'Pin Type' set to 'Adc\_24'. The second panel is titled 'Adc24 properties' and shows 'Number of pins' set to 10. The third panel is titled 'Pin properties' and shows 'Pin Type' set to 'Adc\_24'. The fourth panel is titled 'Adc24 properties' and shows 'Number of pins' set to 10, 'Samples/sec.' set to 3200, and 'Filter' set to 'Max Speed'. Below this is a 'Response speed' slider set to 100. The fifth panel is titled 'Adc24\_channel props' and shows 'Type' set to 'Differential', 'Gain' set to 1, and 'Biased to Vmax / 2' with an unchecked checkbox.

## 配置为 Adc24 的引脚 7 的属性

**引脚数量:**活跃从 1 到 16 条模拟输入线。

**每秒采样数:**在所有活动输入之间划分的采样率 ( 例如，对于 600 sps 和三个输入，每个输入每秒采样 200 次 )。

**筛选:**有八种滤波器可供选择，以实现噪声和响应速度之间的最佳折衷。

## “Adc\_24\_ch”引脚的属性

重要注音的是 16 个输入引脚是成对的 ( 1-2, 3-4, 15-16 )，因此它们的类型 ( 差分、伪和单 ) 和增益 ( 从 1 到 128 ) 都是有效的对于该对的两个输入。

相反，可以在 16 个输入中的每一个输入上单独激活“偏置至  $V_{max}/2$ ”。

The image shows configuration panels for the Adc\_24\_ch\_b pin. The first panel is titled 'Pin properties' and shows 'Pin Type' set to 'Adc\_24\_ch\_b'. The second panel is titled 'Adc24\_channel props' and shows 'Type' set to 'Differential', 'Gain' set to 1, and 'Biased to Vmax / 2' with an unchecked checkbox.

## “Adc\_24\_ch”引脚的属性

如果输入对是差分的，则该对的第二个输入变为“Adc\_24\_ch\_b”。

这是一种特殊类型，不会将数据发送到插槽，仅作为传感器参考连接的占位符。

有关 Adc24 模块的信息和用户手册，包含详细解释和示例：

<http://www.theremino.com/hardware/adapters#adc24>

# “输入”中的引脚类型 <--Usound 和 CapSensor

## ◆ 声音传感器

Pin properties	
Pin type	Usound_sensor
Slot	1
Max value	1000
Min value	0
Response speed	30

UltraSound properties	
Max dist ( mm )	1000
Min dist ( mm )	0

许多超声波距离传感器，例如 SRF05 型号，都可以使用此类引脚进行读取。

这种类型的引脚每 33 ms ( 大约 ) 生成一个正“启动”脉冲，并测量 0 到 32000 微秒的返回脉冲时间。

然后，考虑到空气中的声速，HAL 将时间转换为距离。

“Usound\_sensor”类型引脚使用16位进行数据传输。

## ◆ CapSensor\_HQ

Pin properties	
Pin type	Cap_sensor
Slot	0
Max value	1000
Min value	-5
Response speed	30

Cap sensor properties	
Max dist ( mm )	500
Min dist ( mm )	50
Area ( cmq )	50

这种类型的引脚用于测量与导电物体 ( 通常是手 ) 的距离

检测稳定、精确，响应时间非常快 ( 毫秒级 ) 。

然后将“Min dist”和“Max dist”之间的距离值在“Min value”和“Max value”之间进行比较，用“Response speed”进行过滤，最后发送到 Slot。

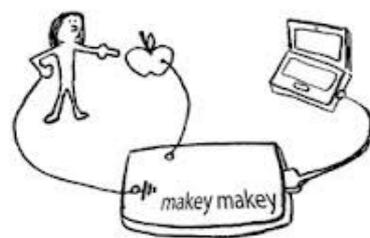
“CapSensor\_HQ”类型的引脚使用 24 位进行数据传输。

**注意：数值CapSensor 的“最小距离”和“最大距离”只是一个近似值。确切的距离并不重要，这些不是测量仪器。未来对线性化公式的改进可能会提高精度，尤其是在短距离内。**

# 电阻式或电容式按键

要了解使用简单的按键可以做什么，请观看这个漂亮的 Makey Makey 视频：<http://vimeo.com/60307041#>

Makey Makey!



然而，Makey Makey 键不是电容式的，而是电阻式的，只有在电阻小于大约 4 兆欧，它们需要额外的电线作为接地参考，并且不能通过塑料等绝缘材料工作。另外，Makey Makey 上的按键只有 6 个（不可扩展），每个 Makey Makey 可以提供 20 个按键，甚至可以连接更多的 Makey Makey，但最终键盘上的按键最多支持 6 个：

[www.makeymakey.com/faq](http://www.makeymakey.com/faq) 最后，Makey Makeys 只进行开/关操作，没有中间调节，并且不感知按键的速度（Velocity）。

Theremino 系统的电容键可以做更多的事情，可以根据需要通过添加无限数量的主模块（每个 6 个键）或伺服模块（每个 8 个键）来扩展，如下所示：[www.youtube.com/watch?v=NbC5kIRS\\_6s](http://www.youtube.com/watch?v=NbC5kIRS_6s) 还有这里：[www.youtube.com/watch?v=2RzwUfXhFZY](http://www.youtube.com/watch?v=2RzwUfXhFZY)

此外，Theremino 系统的按键还可以提供渐进控制，如滑块型光标，甚至“表情”控制，由按键速度。

## 电容式按键的三种类型

Touch properties	
Min variation	20
Proportional area	0

- 开/关键  
“最小变化”从 10 到 50  
“比例面积”必须为零

Touch properties	
Min variation	20
Proportional area	150

- 比例键  
“最小变化”从 10 到 100  
“比例面积”从 100 到 200（最大约为 1000）

Touch properties	
Min variation	40
Proportional area	-30

- 具有力度的按键  
“最小变化”从 25 到 50（调整最大输出）  
“比例面积”-30（调整最大约 1000）

## 容量的通用衡量标准

Touch properties	
Min variation	-400
Proportional area	300

- 电容式传感器（湿度传感器、可变条件传感器等）  
“最小变化”从 -1 到 -1000（最小值的校准）  
“比例面积”从 1 到 1000（最大值校准）

**注意：**使用这种类型的引脚，您无法获得电容的测量值，而只能获得与传感器或位置成比例的值。许多因素都会导致测量非线性，首先是连接电缆的容量。电缆必须非常短，校准后不得再次移动。在所有情况下，都需要在软件中进行适当的缩放调整和线性化。

# 最小变化和比例面积参数

**最小变化**消除了微小的变化，并防止电噪声导致按键即使在不接触按键的情况下也会发出咔嚓声。通过提高此参数，按键会变得不那么敏感，因此最好将其保持尽可能低，以满足消除所有干扰的需要。

对于有力度的按键，该参数的最佳调整是通过快速重复按键并用鼠标滚轮调整“最小变化”以获得最大的输出信号。为了便于调整，暂时将“比例面积”设置为足够大的负数，例如-50。

**比例面积**当手指位于滑块上的最大位置或以最大可能的速度按下按键时，它被调整为大约一千。

通常，对于引脚 1 和 2（不太敏感）以及长电线和大物体，该值必须更高。

## 电容式按键的零位校准

如果在 HAL 程序运行时更改按键的机械排列或其位置，或者移动连接它们的电线或将金属物体靠近它们，则可能会丢失按键的零位校准。如果调零没有很好地校准，电容式按键可能会变得不那么敏感，甚至根本无法工作。

如果您从按键上移除电容（缩短电线或将其移离金属物体），校准会立即自动重新进行，但无法区分由于手指或电线移动而导致的电容增加。

我们尝试了很多自动识别方法、慢漂移方法、定时校准方法，但效果都不好，而且都降低了正常按键操作的准确性。

因此，建议在操作过程中，不要在半径十厘米左右的范围内移动按键线、按键本身以及导电物体。

要检查按键是否已校准，请将手从按键上移开，然后检查其 Pin 的详细信息，确保“Smoot”和“Mean”值彼此相等或非常接近（不超过 1 个点）不同之处）

如果有疑问，请按校准（执行零校准时，手远离按键）

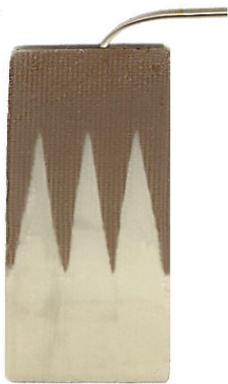
## 电容式传感器的读取

通过将 Min Variation 设置为负值，操作模式将完全改变，并且可以连接湿度传感器（电容变化型且无控制电路）。还可以临时设置传感器来读取销的旋转或线性位移。这种类型的传感器可能很简单，但也非常可靠。

通过将 Min Variation 设置为负值，Min Variation 和 Proportional Area 的含义会发生变化：

- 最小变化设置可测量容量范围的最小值，比例面积设置可测量容量范围的最大值伊尔。
- 校准按钮被禁用。校准是固定的，并且与最小变化的值相同。
- 可用电容值范围从几皮法到几纳法。

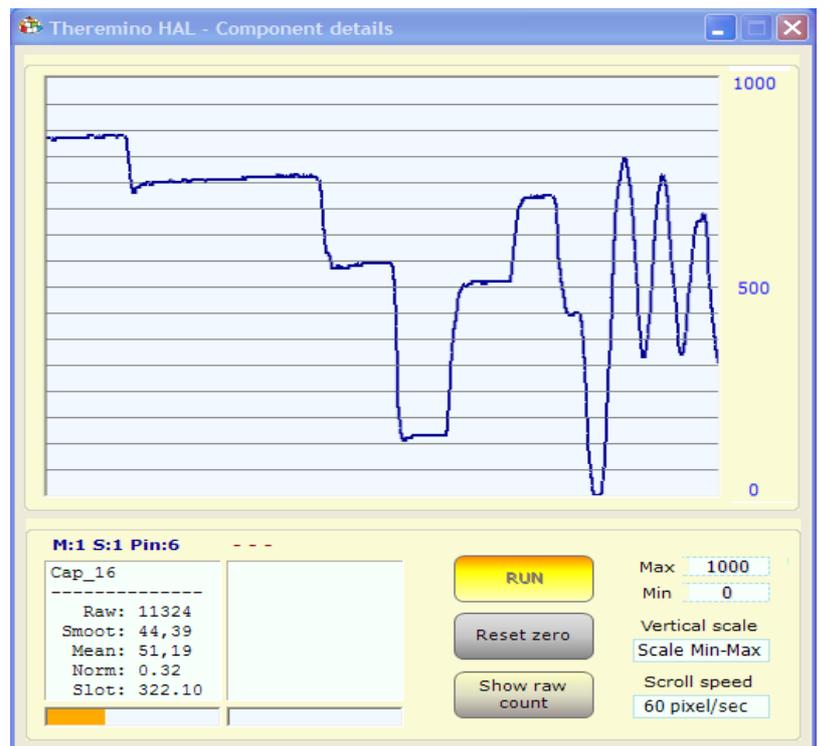
# “滑块”式电容键



**“比例面积”必须是正数，这决定了“比例”类型运算**

具有这种形状的电容式按钮您可以获得类似于“滑块”光标的连续调整

用一根手指即可控制，一路向上=1000，一路向下=0



这些键适用于音量控制，也可用作“紧急按钮”（当您手指从按键上移开时，音量会变为零）

## 这些是“滑块”类型按键的正常设置（注1）

Pin properties	
Pin type	Cap_16
Slot	2
Max value	1000
Min value	0
Response speed	30

Touch properties	
Min variation	20
Proportional area	150

“最大值”通常保持在 1000（笔记2）

“最小值”通常保持为零（笔记2）

“响应速度”一般设置为30（滤光）

“最小变化”通常设置为 10 到 100（最好稍微提高它以在低部分获得最大灵敏度）

“比例面积”通常设置为 200（对于不太敏感的按键或长电线，约为 100）

**(注1)**对于“滑块”类型的按键，最好使用“Cap\_16”

**(笔记2)** 要反转输出信号，您可以将 Max 与 Min 交换（最大值 = 0，最小值 = 1000）

# 具有“速度”功能的电容式按键

The screenshot shows the Theremino HAL V3.0 interface. At the top, there's a menu bar (File, Tools, Help, About) and a toolbar with buttons for 'Edit configurations', 'Recognize', 'Validate', and 'Beep on errors'. A 'Calibrate' button is set to 20. Below this is a table with columns: Type, ID, Subtype, Dir., Slot, Value, Notes. The table lists a Master (ID 1, Subtype Theremin) and a Slave (ID 1, Subtype MasterPins). Below the Slave are five pins (ID 1-5, Subtype Cap\_16) with Dir. 'get' and Slot values 2, 3, 4, 5 respectively. All Value fields are 0.0. To the right of the table are three property panels: 'Master properties' (Name: Theremin, Rep freq: 489, Error rate: 0.00, Comm. speed: 9, Fast data exchange: checked), 'Pin properties' (Pin type: Cap\_16, Slot: 2, Max value: 1000, Min value: 0, Response speed: 30), and 'Touch properties' (Min variation: 40, Proportional area: -30). Below the table is a 'Component details' window for 'M:1 S:1 Pin:4'. It shows a bar chart with a peak of 1000. Below the chart are statistics for 'Cap\_16': Raw: 14156, Smoot: 55,32, Mean: 54,37, Norm: 0.00, Slot: 0.00. There are 'RUN', 'Reset zero', and 'Show raw count' buttons. On the right of this window are 'Max' (1000), 'Min' (0), 'Vertical scale' (Scale Min-Max), and 'Scroll speed' (60 pixel/sec) settings.

**“比例面积”必须为负数，这决定了“速度”**

键盘允许您轻柔或大声地弹奏音符，具体取决于您按下琴键的方式，并且在音

乐应用程序中非常受欢迎。可以调节电容式按键来测量按键速度并将其转换为 0 到 1000 (大约) 之间的值

为了“Velocity”的良好功能，通信速度必须很高 (从 200 到 500 fps)，并且必须逐一调整按键，以获得刚刚超过 1000 的最大值

This block shows a close-up of the 'Pin properties' and 'Touch properties' panels. The 'Pin properties' panel includes: Pin type (Cap\_16), Slot (2), Max value (1000), Min value (0), and Response speed (30). The 'Touch properties' panel includes: Min variation (40) and Proportional area (-30).

这些是带有“Velocity”键的设置

“最大值”通常保持在 1000 (注 1)

“最小值”通常保持为“0” (注 1)

“响应速度”一般保持在 30 (最好不要改变)

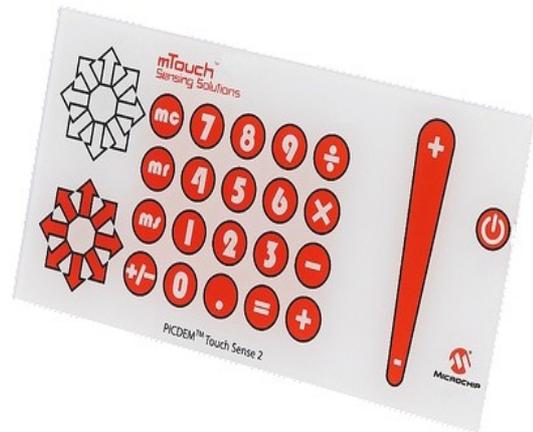
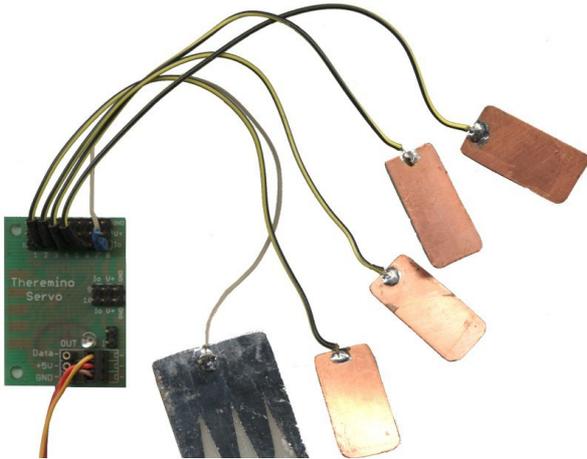
“最小变化”通常设置为 50 (对于灵敏度较低的按键 1 和 2 或具有长电线的按键，大约设置为 25)

“比例面积”通常设置为 -40 (对于灵敏度较低的按键 1 和 2 或电线较长的

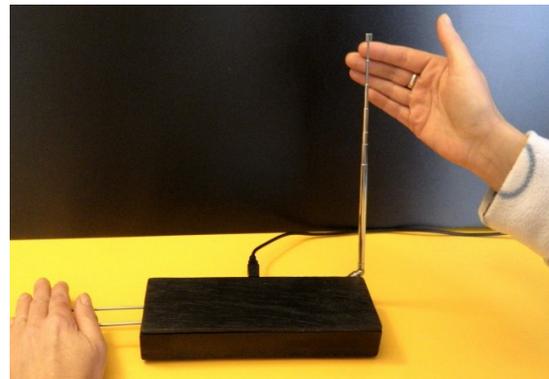
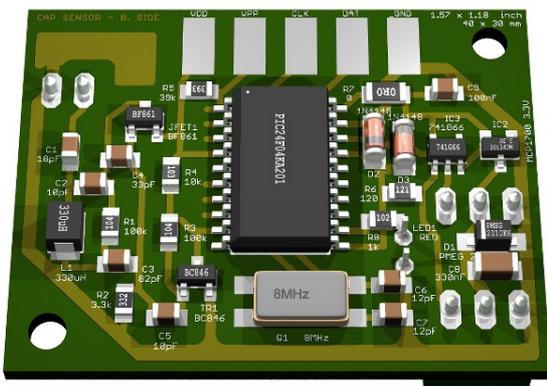
的按键，约为 -20)

(注1)要反转输出信号，您可以将 Max 与 Min 交换 (最大值 = 0，最小值 = 1000)

# 电容式按键和 CapSensor 模块之间的差异



电容式按键不会取代 CapSensor 模块，前者仅在短距离（从几毫米到几厘米）内工作，而 CapSensor 的工作距离可达数米，并且可以调整以获得几乎完美的线性响应。而电容式按键的成本要低得多，更适合组成多按键的键盘。



## 电容式按键的机械结构

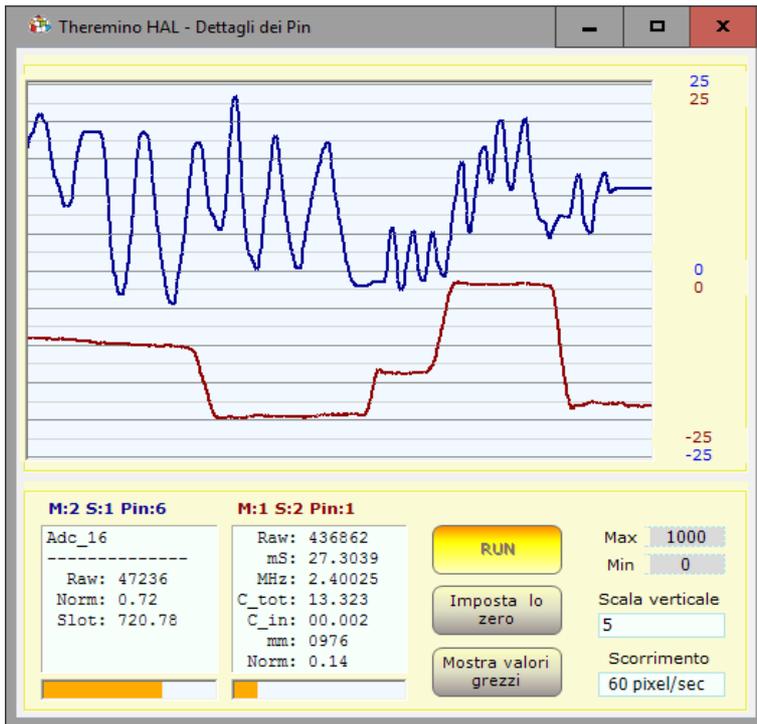


**注意电容按键一定要绝缘，否则一个小小的静电火花就足以产生通讯错误，不会损坏任何东西，但通讯会阻塞，必须按下“识别”按钮**

因此，通常情况下，您将铜放在下面，（薄）玻璃陶瓷放在上面，或者更好的是，将铜放在上面，并添加一层薄薄的绝缘塑料，其颜色印有按键的形状，如图所示在本页的开头。

从按键到引脚的电线必须尽可能短，并且间隔至少 5 或 10 毫米。通过减小电容，可以改善按键的操作和噪音隔离。测试也在“不可能”的情况下进行，使用很长的电线和各种按钮，例如盆栽花卉和各种水果，并且单独的调整总是使所有按钮都工作。

# Pin 详细信息查看器



双击活动引脚的线可打开此工具。对于两个信号，首先单击一个引脚，然后单击第二个引脚。

垂直刻度可以设置为“Scale Min-Max”，对应方框的值最小值和最大值。

或者您可以将其设置为从 0.01 到 50000 点的 24 个级别，每个垂直分区（十条暗线）。如果使用它们这些设置，要使曲目居中，请按“重置零”。

在某些情况下，查看原始值（英文为“Raw”）可能很有用。要查看“Raw”值，请启用“Show raw count”按钮。

“滚动”控件可将图表的滚动速度从每秒 0.1 像素调整到每秒 60 像素。

两个文本框显示了引脚的内部详细信息，标题指示了它是哪个引脚，在此图中，文本“M:1 S:1 Pin:2”表示“Master 1，Slave 1，Pin 2”。

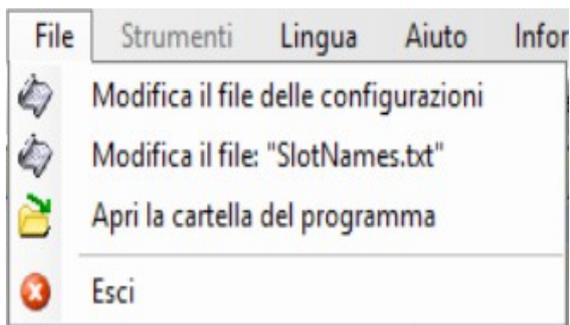
引脚详细信息有助于控制和调节输入输出设备（传感器和执行器）。

某些 Pin 类型更复杂并且具有多个中间值。一般来说，有一个“原始”值，其值根据引脚的类型而变化，一个“标准化”值，其范围始终从 0 到 1，以及一个“槽”值，其范围通常从 0 到 1000，它是“简化”可在插槽上使用，并且可由所有高级软件轻松使用。

- ◆ 生的 “原始”值可以是计数、时间、电压或其他。
- ◆ 多发性硬化症 时间（以毫秒为单位）
- ◆ 微秒 时间以微秒为单位
- ◆ 兆赫兹 重复率
- ◆ 总碳 与 330 uH 线圈并联的总电容（仅用于 CapSensors）
- ◆ C\_in 校准后在输入上添加的电容（仅在 CapSensors 中）
- ◆ 毫米 近似距离（以毫米为单位）（仅用于 CapSensors 和超声波传感器）
- ◆ 斯穆特 已传递到平滑 FIR 滤波器的值（仅在 Ch8 和 Ch16 中使用）
- ◆ 意思是 平均值（在 Cap8 和 Cap16 类型中用作零校准）
- ◆ 规范 值标准化在 0 和 1 之间
- ◆ 老虎机 从与引脚关联的插槽写入或读取的值（通常为 1 到 1000）
- ◆ 出去 只能为“0”或“1”的数字化值（仅由 DigOut 使用）

**即使未注明，容量也始终以皮法 (pF) 为单位**

# 菜单命令



**编辑配置**在某些情况下这可能很方便。有关更多信息，另请阅读本文档最后一页的“问题与解答”。

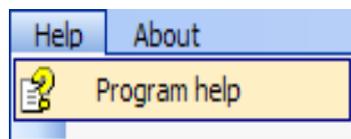
**编辑文件**：“SlotNames”注释（或插槽名称）在中进行了解释[这一页](#)。

打开**工作簿**对于编辑文档和语言文件很有用。



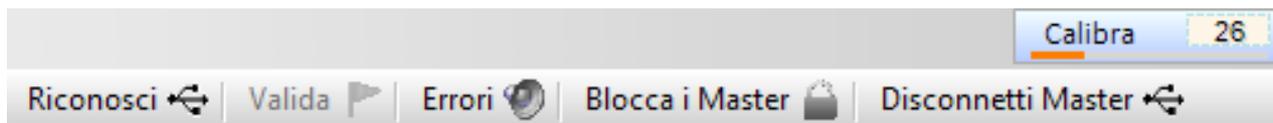
语言文件位于 ThereminoHAL.exe 应用程序旁边的“Docs”文件夹中。

要制作新的语言文件，只需复制 Language\_ENG.txt 文件，将“ENG”更改为“FRA”、“ESP”、“DEU”或“JPN”，然后使用记事本编辑文本。



此命令打开文档文件。

# 工具栏命令



## 认出

用于识别连接到 USB 的 Master 和 Slave

## 有效的

当您从链中添加或删除从站模块时，系统会警告您配置已更改，并在列表中显示红线。如果您选择丢失旧配置并适应当前硬件，则此按钮将使新配置生效。

## 错误

如果按下，通信错误会通过声音突出显示。

## 块大师

如果按下，HAL 仅识别当前列表中存在名称的 Master。通过这种方式，可以（在单独的文件夹中）维护不同的 HAL 应用程序，每个应用程序都与其特定的硬件相结合。

## 断开主控

从列表中删除选定的主控。通过这种方式，您可以删除不需要的母版，而无需物理上断开它们与 USB 的连接。删除它们后，建议按“锁定 Masters”，以便下次启动时仅重新加载所需的 Masters。

## 校准

设置 CapSensors 和 CapKey 的“皮重”。在按压之前将手从敏感部位移开。如果没有超过设定数量的移动，则每 30 秒进行一次自动校准。您可以使用值“0”禁用自动校准

# 隔离应用程序

Theremino 系统的一些应用他们自动启动自己的 HAL。如果 ThereminoHAL 文件夹中存在 Theremino\_HAL.exe 文件（位于应用程序的 EXE 文件旁边），则会发生这种情况。您还可以仅将 Theremino\_HAL.exe 文件放在应用程序的 exe 文件旁边，但 HAL 最好有自己的文件夹，其中 Docs 子文件夹包含文档和语言文件。

这些 HAL 使用自己的私有配置，如果它们具有“块大师按下后”，它们仅连接到自己的主设备，并通过名称在连接到 USB 端口的主设备中识别它们。以这种方式组成的应用程序将继续工作，即使复制到另一台计算机上，即使其他 Theremino 系统应用程序正在其他 USB 端口上与其主控进行通信。

从这些可能性中受益最多的应用程序是具有特定任务的应用程序，例如：Theremino Geiger、Theremino OilMeter、Theremino Meteo、Theremino Theremin、Theremino Arm、Theremino Geo 和 Theremino EmotionMeter。

这并不是说孤立的应用程序不能相互通信。通讯模块化始终是可能的，并通过插槽实现，这对于所有应用程序都是通用的。

为了避免对不同的任务使用相同的槽，我们定义了一个通用方案。

```
实验 100 个槽位 000 - 099
- - -
Theremino_Theremin100 - 199
Theremino_SlotsToMidi200 - 299
Theremino_MusicKeys300 - 329
- - -
469 个免费插槽 330 - 799
- - -
Theremino_油表 800 - 809
Theremino_脑电图 810 - 819
Theremino_Meteo820 - 839
Theremino_Arm840 - 849
10 个免费插槽 850 - 859
10 个免费插槽 860 - 869
10 个免费插槽 870 - 879
Theremino_EmotionMeter880 - 889
Theremino_盖革 900 - 909
Theremino_Bridge900 - 909
Theremino_GEO910 - 919
Theremino_GeoPreampTester920 - 929
Theremino_雷达 930 - 939
10 个免费插槽 940 - 949
10 个免费插槽 950 - 959
10 个免费插槽 960 - 969
10 个免费插槽 970 - 979
10 个免费插槽 980 - 989
10 个免费插槽 990 - 999
```

该图仅供参考。可以根据需要使用插槽，只要同一个插槽不同时在同一台 PC 上用于两个不同的任务即可。如果您犯了错误，不会造成任何影响，但数据会与不确定的结果重叠。

# 调整数字框

Draw speed (fps) 5

HAL 的数字框（以及 Theremino 系统的所有其他应用程序）由我们开发（注 1），比原始的 Microsoft 文本框更加舒适和灵活。

## 数值可以通过多种方式调整

- 通过单击并按住鼠标左键并上下移动鼠标
- 使用鼠标滚轮
- 使用键盘上的向上箭头和向下箭头键
- 使用键盘书写数字的常规方法
- 使用正常选择和复制粘贴方法
  
- 按 SHIFT 键，变化率会乘以一百
- 按 CTRL 键，变化率乘以十
- 按 ALT 变化率除以十

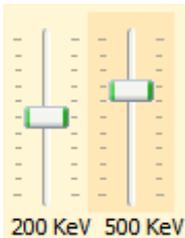
上下移动鼠标可以进行大幅且快速的调整

鼠标滚轮可以方便快捷地进行调整

使用箭头键可以进行精细调整，而无需将视线从正在调整的内容上移开

(注1) 与我们所有的软件一样，它们的源文件是可用的（知识共享许可下的免费软件和开源），可以从这里下载：[www.theremino.com/downloads/uncategorized](http://www.theremino.com/downloads/uncategorized)（“自定义控件”部分）即使不命名源，也可以在任何项目中根据需要使用这些控件。“开放”来源还可以保证我们没有包含恶意软件。

# 滑块调节



这些是原始的 Microsoft 滑块，它们非常舒适，因此我们只是添加了橙色和重置它们的功能。

<<< 非零光标以橙色突出显示，要重置它们，只需单击鼠标右键（并非所有光标都有零，在这种情况下，它们没有着色，无法使用鼠标重置）

## 滑块可通过以下方式调节

- 用鼠标右键单击光标来“重置”它们
- 通过用鼠标左键单击光标并上下移动鼠标
- 使用鼠标滚轮
- 使用键盘上的左箭头和右箭头键
- 使用键盘上的向上箭头和向下箭头键

上下移动鼠标的方法可以进行大幅度且快速的调整。

鼠标滚轮可以方便快捷地进行调整。

使用箭头键可以进行精细调整，而无需将视线从正在调整的内容上移开。

左/右或上/下箭头键具有相同的效果，但使用前者用于水平光标，后者用于垂直光标可能更直观。

## 问题与解答

### 我可以更改各种语言的程序面板文本吗？

当然，只需编辑文件：“..\Docs\Language\_Eng.txt”和“..\Docs\Language\_Ita.txt”

对于德语、法语和西班牙语，只需将英语文件复制三遍，名称如下：

“..\Docs\Language\_Deu.txt”、“..\Docs\Language\_Fra.txt”、“..\Docs\Language\_Esp.txt”

### 我可以编辑配置文件吗？

通常，主从配置和模块之间的关联由 ThereminoHAL 保持一致，它使用主名称来建立要采用的正确配置。通常，即使主模块和从模块断开连接并更换，HAL 也能够使用正确的配置。

在某些情况下，如果您使用位于不同计算机或不同文件夹中的 HAL 更改 Master 的名称，则配置和硬件之间的一致性就会丢失。在这些情况下，您可以单击 Master 名称的下拉菜单，并通过为每个 Master 选择正确的配置来恢复对齐。

要进行更复杂的更改，您可以使用“记事本”等文本编辑器打开“Theremino\_HAL\_ConfigDatabase.txt”文件，然后手动编辑非常简单的配置。

### 如何减少CPU工作量？

- 关闭或最小化“组件详细信息”窗口。
- 最小化主窗口。
- 降低“通信速度”，如本文档第一页所述。