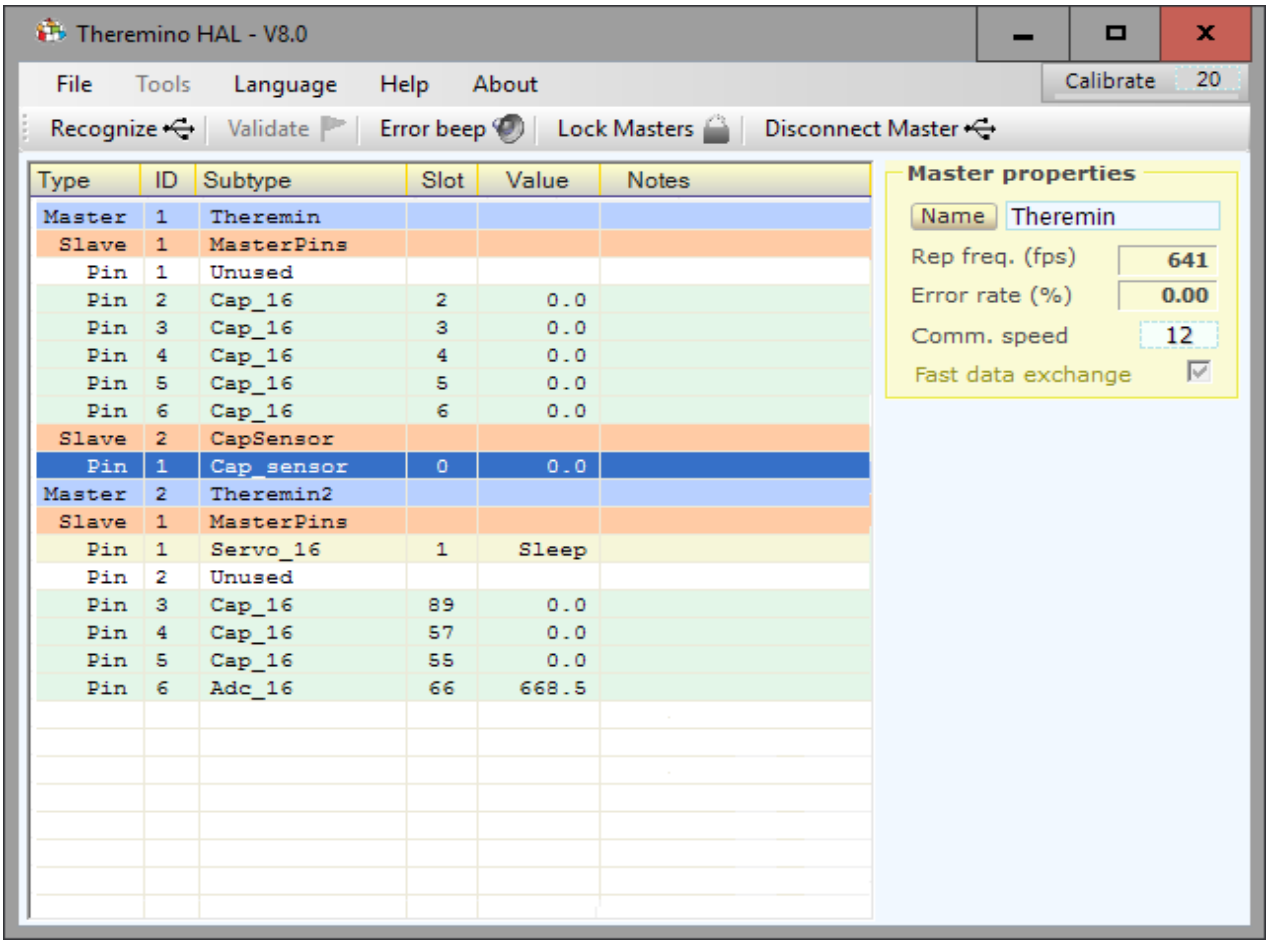
System Theremino

Theremino HAL - V8.x Instructions

Theremino HAL



Theremino HAL with two Masters connected

Theremino HAL (Hardware Abstraction Layer) is a Hardware Manager. It appears with a simple interface, but carries out complex operations, with highly optimized algorithms.

Theremino HAL is the heart of the communication with the hardware, it knows how to communicate with many Masters at the same time, deals with the protocol and USB serial communication, manages all the most common types of Input-Output and knows how to recognize the "Slave" modules.

Without HAL, communicating with hardware would be difficult (same as in Arduino), would require a lot of time and labor (same as in Arduino) and finally for each In-Out, like moving a motor or even just reading a key, the appropriate firmware, should be written (same as in Arduino).

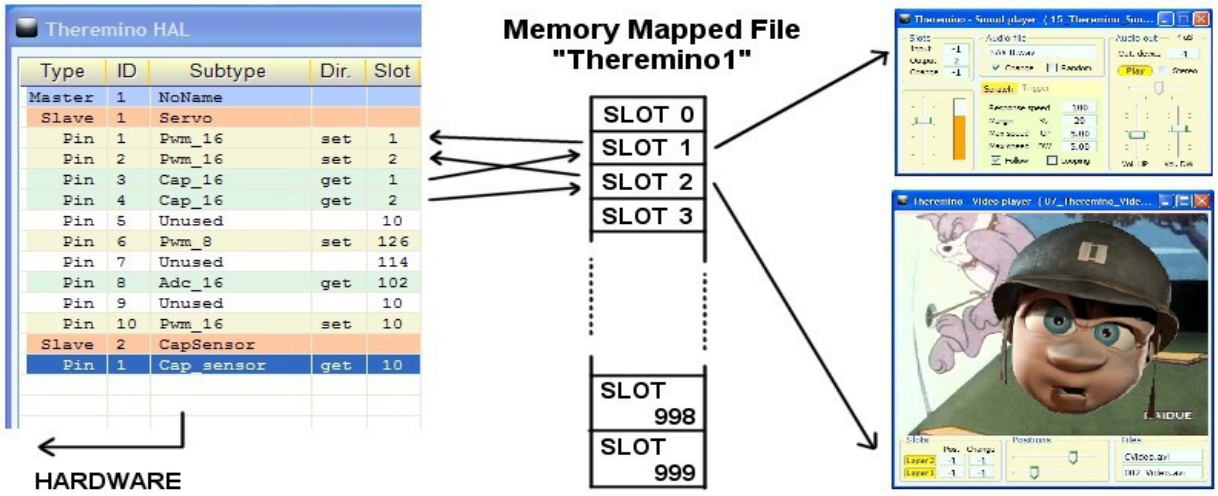
If you use hardware modules, *then HAL is essential and must remain active. You can minimize it, but it should remain running.*

If you do not use hardware *then the HAL is not necessary, the applications of the system can communicate with each other through Slots, even without HAL.*

When you add or subtract Slave modules from the chain, *you are warned that the configuration has changed, with some red lines in the list. If you choose to loose the old configuration and accept the actual hardware, the button "Validate" makes the new configuration valid.*

The "Slots"

Theremino System Slots are identified by a number from 0 to 999 and are all part of a MemoryMappedFile called "Theremino1". Each Slot contains a "Float" number, which can be read or written by any module of the Theremino System.



In this picture, only HAL writes in the Slots, but in reality all the components of the system can both read and write in any of the Slots, even if already used by others.

Choosing the right Slot, you should be aware of two things:

- Ensure you do not use the same Slot by mistake, for two different functions.
- Avoid writing on the same Slot, with two or more components.

Input Pins, that are writing in the Slots, are highlighted in light green. If two of more input pins have the same Slot, then the HAL application warn with red lines and the text **SLOT CONFLICT**.

Many applications and Pins can read from the same Slot. But avoid configuring more than one Pin writing on the same Slot; doing so nothing is broken, but the results are unpredictable.

Sending multiple streams of data to the same Slot, all data are mixed and the last who writes wins. if you want to merge data in order, some rules are required.

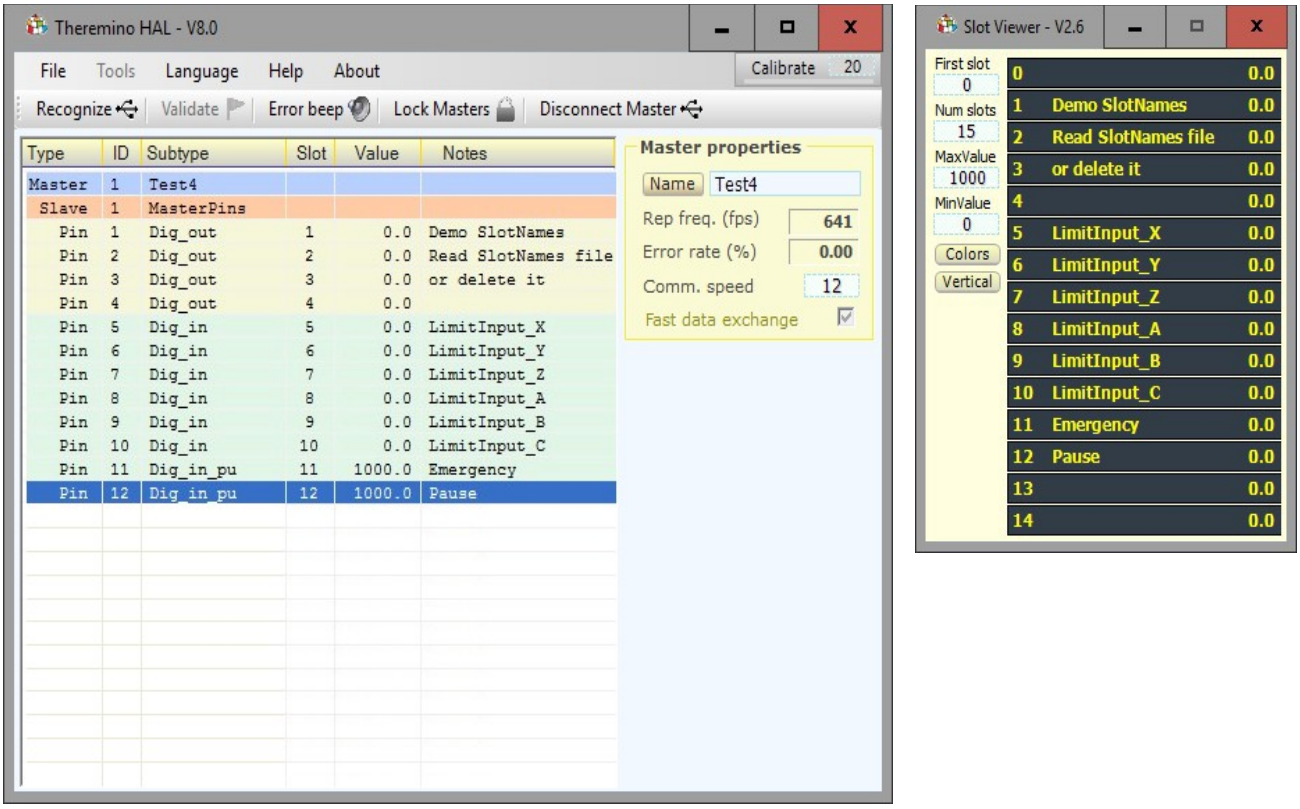
Type	ID	Subtype	Slot	Value	Notes
Master	1	TestSlotCo...			
Slave	1	MasterPins			Firmware V5.0
Pin	1	Adc_16	1	105.3	
Pin	2	Adc_16	2	99.5	
Pin	3	Dig_in	4	0.0	SLOT CONFLICT
Pin	4	Dig_in	4	0.0	SLOT CONFLICT
Pin	5	Dig_in	5	0.0	
Pin	6	Dig_in	6	0.0	
Pin	7	Dig_in_pu	7	1000.0	
Pin	8	Unused			

To establish mathematics and logic rules between the Slots and to write complex behavioral algorithms, as well, we use Theremino Automation or Theremino_Script, or else any other programming language, such as C + +, CSharp, VBnet or VB6. Visual languages like MaxMSP, Processing, PureData, LabView and EyesWeb, can also be used. Plugins and examples for MaxMSP, are ready made here:
www.theremino.com/en/downloads/foundations

For more information about communication, please check this page:
www.theremino.com/en/technical/communications
www.theremino.com/en/technical/pin-types

The Slot names

The application HAL (from version 5.5) and the SlotViewer (from version 2.6), can display the name of the Slot (or annotations or comments).



Important to note that the names are not related to physical Pins, but to the Slots.

The names are written in a file, that should be called “SlotNames.txt” and that must be in the same folder as “Theremino_HAL.exe” and “Theremino_SlotViewer.exe”. If the file “SlotNames.txt” is not present then the comment field will remain empty.

To modify the Slot names you open the “File” menu, choose “Edit slot names file”, and edit it with the default system editor (normally NotePad). Finally you save the file and it will be reloaded automatically.

The rules are simple and are shown in the sample file, located in the latest versions of HAL and SlotViewer.

Each line of the file begins with the Slot number, followed by a space and the text to be displayed. The line can also continue with a comment, that does not appear, preceded by a single quote.

If you want to use the same file of comments, for both HAL and SlotViewer, you have to keep the files “SlotNames.txt”, “SlotViewer.exe” and “HAL.exe”, all in the same folder.

The "Command Slot"

The Theremino System applications, or other applications created by users, can communicate with the HAL, sending commands or receiving data, using a special Slot to communicate.

For example an applications may change the parameters of all the Pins, rewriting the configuration file, and then sending the command "Recognize". Or an application could verify how many Masters are really connected, sending the “Recognize” command, and then reading their number from the Command Slot. Or a musical application could calibrate the CapSensor modules or the CapKeys, sending the “Calibrate” command.

Using other Slots in place of the Slot zero.

Normally the Command Slot is zero, but may happen that you want to use multiple independent applications on the same PC. In these cases, each application would reside in separate folders together with its HAL, and accesses its Master modules using the "Lock master" command. In these cases, a different communication Slot can be assigned to each HAL. For the commands you can use any Slot (from 0 to 999) but be careful not to assign it to any Pin.

To assign a non-zero number to the Command Slot, manually edit the last line of the file "Theremino_HAL_INI.txt". So to assign, for example, the Slot 300 for the commands, you would write: `CommandSlot = 300`. Be careful not to delete the "=" sign. If something is wrong, the Slot zero is used, and the HAL rewrites the corrected line in the INI file.

How to send commands

Currently two commands are defined:

- Recognize — You send the “NAN_Recognize”, or the number “1”
- Calibrate — You send the “NAN_Calibrate”, or the number “2”

The applications that are NOT able to send the NANs (Not A Number) special numbers, can use the the numbers “1” and “2” instead of the “NAN_Recognize” and “NAN_Calibrate”.

If you use the numbers “1” and “2”, they must be preceded by a sequence. This sequence has two numbers (333 and 666) that correspond to actually floating point numbers, with seven figures of precision, 333.0000 and 666.0000. So it is virtually impossible for an ADC or other devices to send this sequence by mistake.

Response messages

Responses, and error messages, are communicated with numbers in the command slot.

- -1 — The “Recognize” command is still executing.
- 0 — No Master have been found. The list is completely white.
- 1 and up — The number of Masters that has been recognized.
- NAN_MasterError — One of the connected Masters has stopped communicating.

The "Command Slot" - Examples

To send the "Recognize" command you write:

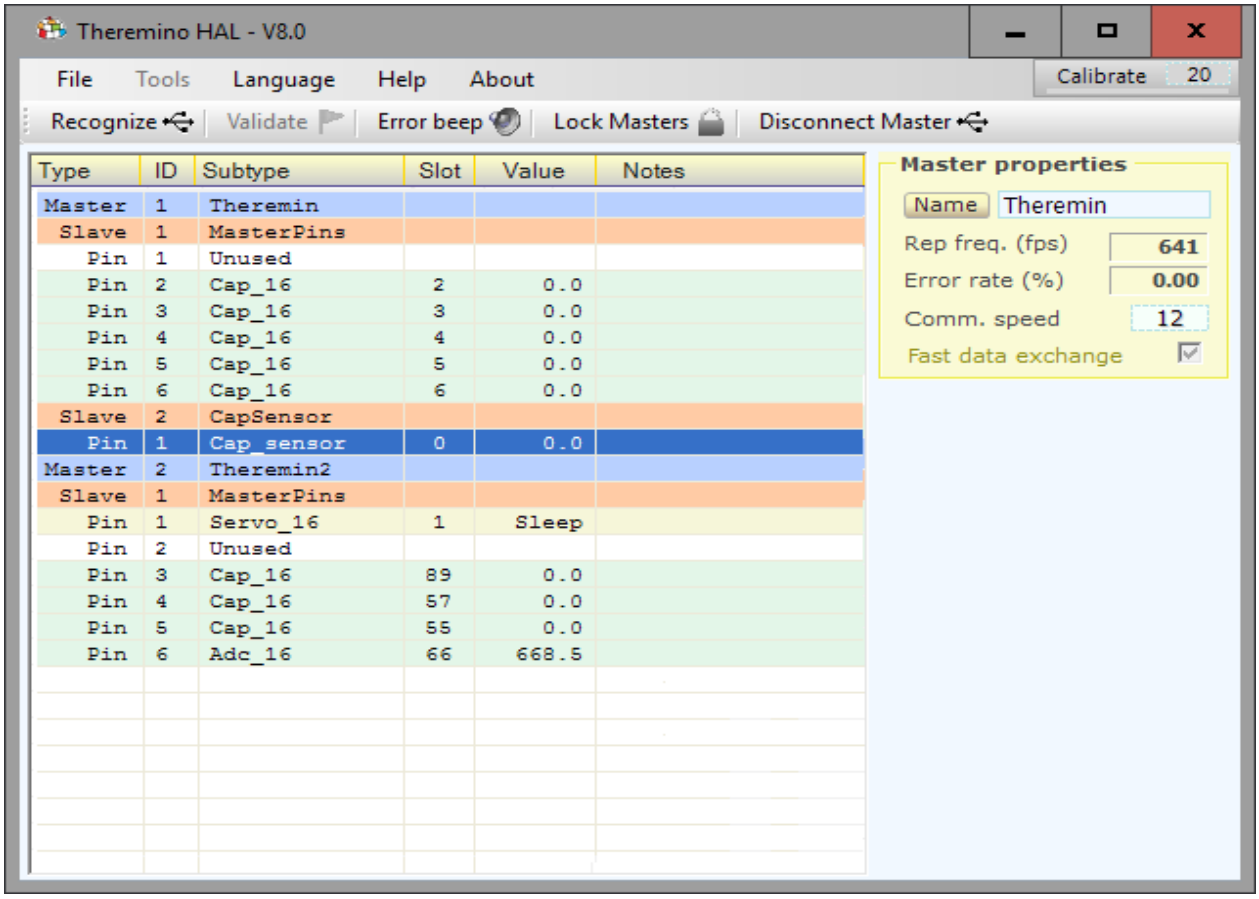
```
---------------------------------------- VbNet
Slots.WriteSlot (0, NAN_Recognize)

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

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

As explained on the previous page, some applications (Theremino Automation for example) are not capable to use the NAN special numbers. Not using NANs the previous examples would become:

```
---------------------------------------- VbNet
Slots.WriteSlot (0, 333)
System.Threading.Thread.Sleep(50)
Slots.WriteSlot (0, 666)
System.Threading.Thread.Sleep(50)
Slots.WriteSlot (0, 1)

---------------------------------------- CSharp
Slots.WriteSlot(0, 333);
System.Threading.Thread.Sleep(50);
Slots.WriteSlot(0, 666);
System.Threading.Thread.Sleep(50);
Slots.WriteSlot(0, 1);

---------------------------------------- Theremino Automation
Slot 0 = 333
Wait Seconds 0.05
Slot 0 = 666
Wait Seconds 0.05
Slot 0 = 1

---------------------------------------- Theremino Script
WriteSlot (0, 333)
Threading.Thread.Sleep(50)
WriteSlot (0, 666)
Threading.Thread.Sleep(50)
WriteSlot (0, 1)
```

The 50 milliseconds waiting instructions are used to give time to the HAL to read the Slot.

The HAL colors



The color scheme, helps to recognize the components and their configuration

- The light green color indicates an Input

- The light yellow color indicates an Output

The first Master (with name Theremin) provides:

A virtual slave called "Master pins"

Six "Pins" of which, only the first one is "Unused", the others are configured as "Cap_16"

A slave of type "Cap Sensor"

A single Pin configured as "Cap sensor" and "Selected"

The second Master (named Theremin2) provides:

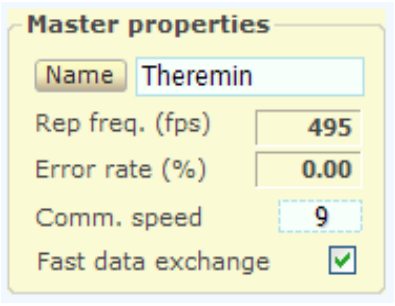
A virtual slave called "Master pins"

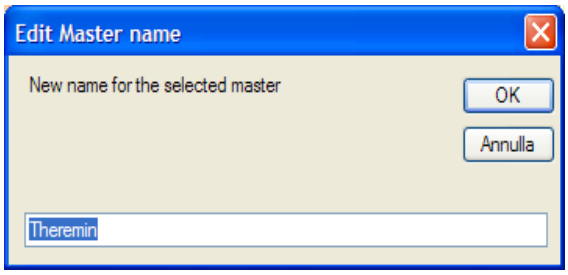
A "Pin" configured as "Servo_16"

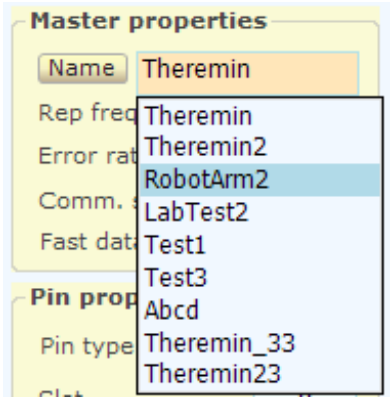
A "Pin" configured as "Unused"

Four "Pins" configured as Cap_16

Master's properties - The name







The selected Master name, can be changed in two ways:

- Pressing the "Name" button and editing it.
- Clicking on the name and select a different configuration, from the pop-up menu.

The Master's name**,** is written on the hardware module and is used to recognize it, when connected.

When a new Master is connected, it is automatically called "NoName". We suggest you to rename the card differently, to distinguish it from all the others.

While dialing the name, the letter case (uppercase or lowercase) does not count.

If in the database two Masters are present with the same name, the first configuration is used for both the Masters. It is therefore important to give different names to each Master (unless you want to have spare Masters, with the same name).

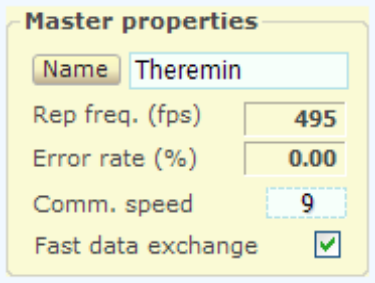
Masters are always listed in alphabetic order, If the USB port is changed, the order of the Masters does not changes.

HAL should always set the right configuration when you unplug, replace or restore components, but if Master names are changed on a different computer or through another HAL application (on a separate folder - with then, separate parameters) or other difficult and complex cases, the alignment between the configuration and the hardware, might get lost.

If you lose the alignment you should restore the configuration manually, one Pin at a time, but experts can edit the configuration file and possibly copy this whole file or only parts of it, from one HAL application, to another, on a different computer or in another folder.

When the configuration is invalid, changing the name of the Master, does not modify the configuration file, but only the name written in the hardware. It is possible to change the names of the Masters, until they match the right ones, in the configuration.

The Master properties - Communication



- Number of reports per second
- Percentage of errors on the serial line (usually zero)
- Serial communication speed (from 1 to 4 Mega Kilo Baud)
- Selection of the type of communication "Single" or "Fast" (Note 1)

The number of messages (frames) per second "Fps", should normally be set from 480 to 500, if the serial communication toward the physical Slaves and the Pins exceeds a certain number of bytes and the transmission speed is low, then this number decreases.

For many applications, such as capacitive keys with velocity, it is good to maintain the fps, as high as possible, at least at 400 or 450.

To increase the "fps" (with connected Slaves):

- Increase the "Comm speed" (consistent with the length of the serial link)
- Use "Fast data exchange" (to decrease the maximum number of bytes to 64, but increasing speed)
- Split the serial lines and connect the critical Pins, to the less loaded lines
- Divide up the serial lines and connect the critical Pins, on one or more Masters not connected to the serial
- Reduce the number of bytes used, by configuring as "Unused" all possible Pins
-Decrease the number of bytes used, by setting at 8-bits all Pins that do not require great resolution

To trim the "fps" (with no Slaves connected):

With the "Comm speed" value, you can adjust the refresh rate "fps", even without Slaves connected to the serial line. In this case the value "Rep freq. (Fps)" refers only to the USB communication.

To increase the response speed would be good to maximize the exchange rate, and set "Comm Speed" to "12". But for most applications 100 fps are more than enough, so normally it can adjust the "Comm Speed" from 8 to 10 and reduce the CPU load.

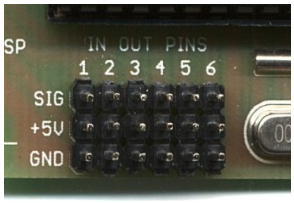
Slaves and Pins

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

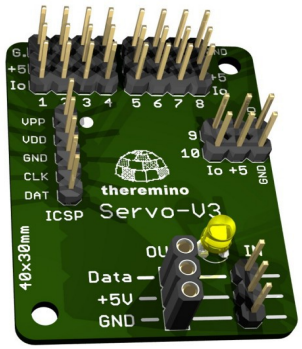
Slaves, here indicated by red arrows, have no regulations, being only Pin **containers.** The Slaves usually have between 1 to 12 Pins.

Pins are all equal to each other and configured in many different ways.

The Master module has a built-in slave (called virtual slave), that provides from six to twelve Pins, depending on the firmware version.

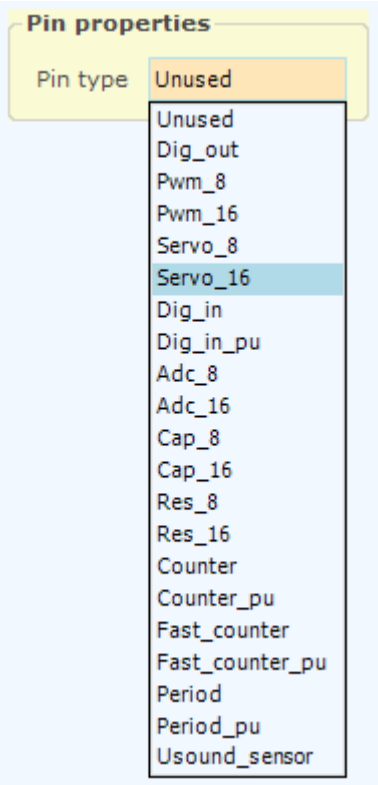


"Slave" modules of "Servo" type, have 10 Pins.



"Cap sensor" type "Slave" modules, have a single, high quality "Pin" (capable of detecting movements of large objects, up to several meters away). The new “HS” firmware has faster response (< 1mS).

Pin types



Pins can be configured as:

- Not used
- Digital Output
- PWM output (250 Hz)
- PwmFast output (from 250Hz to 5 MHz)
- Servos output
- Stepper motors output
- Digital input
- Potentiometers and transducers ADC input
- Capacitive buttons input
- Resistive transducers input
- Frequency, Period and Counters input
- Special transducers input
- CapSensor modules Input
- Two phase Encoders input
- ***Adc24 control channels*** (Pins 7, 8, 9)
- ***Adc24 channels*** (Adc24 Pins from 1 to 16)

The special Master Pins:

- Pins from 1 to 6 are the more flexible. You can configure them with almost all types.
- The Pins 7, 8 and 9 in addition to the normal functions are used to connect the Adc24 module.
- The Pins 11 and 12 can not be configured as Stepper or PwmFast.

The special Slave Pins:

- Pins 9 and 10 of the Slaves "Servo" can not be configured as ADC, CAP and RES
- Pin 8 of the Slaves "Servo" is the only one, configurable as "Fast counter"
- Pin 9 of the Slaves "Servo" is the only Pin configurable as "Period" and as "Usound sensor"
- The single "CapSensor" Pin, can only be configured as "Unused" or "Cap sensor"

The best Pins to be used as ADC and CAP:

- The best Pins to be used as ADC and CAP are Pins 3,4,5,6
- Pins 7 and 8 have leakage current and double capacity (second choice for ADC and CAP)
- Pins 1 and 2 have leakage current and a capacity four times greater (third choice for ADC and CAP)

All Pins can be configured as "Unused", this allows to decrease the number of bytes transmitted on the serial and USB and maximize the number of data exchange per second.

The choice between 8 and 16 bit, available on many types of Pins, allows to obtain the maximum resolution (16 bit) or a lower resolution (8 bit) but a greater bits saving, obtaining the maximum communication speed.

The types with pullup, which name ends in "_pu", allow you to easily connect switches, buttons, and open-collector devices, without having to add external resistors (typical PullUp current = 250 uA).

For Stepper and PwmFast Pins, you need: Masters with firmware 3.0 and HAL version 5.0 (or later)
For Encoders Pins, you need: Masters with firmware 4.0 and HAL version 5.3 (or later)
To connect the Adc24 modules you need: Masters with firmware 5.0 and HAL version 6.5 (or later)

Further information about Slots, Pins and Modules

More information about Pins:

www.theremino.com/en/technical/pin-types

Features and Pins of the individual modules:

www.theremino.com/en/hardware/devices

Information about Stepper Pins:

www.theremino.com/en/hardware/outputs/motors

Information about the Adc24 module:

www.theremino.com/en/hardware/adapters#adc24

Data-sheets of the modules:

www.theremino.com/en/technical/schematics

Output Devices (actuators):

www.theremino.com/en/hardware/outputs

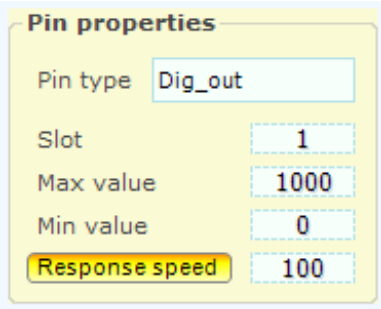
Input Devices (sensors):

www.theremino.com/en/hardware/inputs

Blog and Tips about the in-out modules:

www.theremino.com/en/blog/master-and-slaves

The parameters common to all Pins



"Slot" indicates where to write or read data. Slots are 1000, numbered from 0 to 999 and can be read or written by all Pins and by all the Theremino System applications.

Please note: Many applications and Pins can read from the same Slot,
but avoid to configure more than one Pin, writing to the same Slot.
Doing so doesn't damage anything, but the results are undefined.

"Max value" normally set to 1000, indicates the value that the Pin must have, when at its maximum.

"Min value" usually set to zero, indicates the value that the Pin must have, when at its minimum.

By adjusting Max and Min, with values other than 0 and 1000, you can achieve any scale ratio and calibration. If you exchange the two values (min value larger than max), then the scale is reversed, this is useful to reverse the movement of the actuators or to turn the readings of sensors, that act reversed.

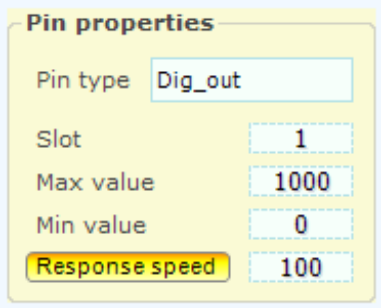
"Response speed" adjusts the filter IIR (Infinite Impulse Response) for the best compromise between noise and response speed. With a value 100, the filter is disabled and the maximum speed of response is obtained. The value 1 produces the maximum filtering, (elimination of any jitter) but a very slow response (approximately one second). Normally we use the value of 30, which provides a good filtering and a fast enough speed.

If the **"Response speed"** button is pressed, the IIR filter adapts to variations in order to obtain a higher reactivity, when there are wide variations and a greater damping, when the changes are minor. As a result you get a good stability of the digits, without too much sacrificing the settling time.

Some sensor signals may malfunction with **"Response speed"** pressed. This is specially true for sensors producing a signal with little variations around a high base value. In this case the signal never arrives to the final value or is very slow to arrive. If you experience this, disable **"Response speed".**

The "Output" Pin types --> Dig / PWM / Servo

DIG_OUT

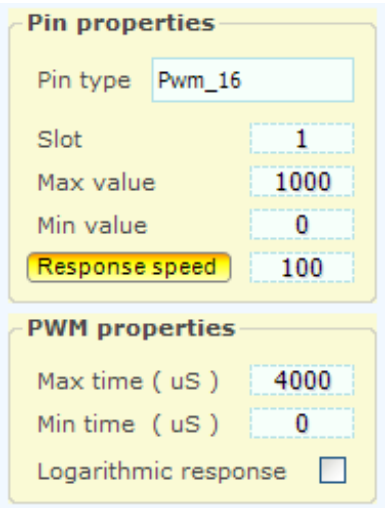


This type of Pin provides a digital output.

The value coming from a Slot, limited between "Min value" and "Max value" and filtered by "Response speed", is compared with the value between "Min value" and "Max value". If exceeded, the Pin turns on, otherwise off.

The Pin can only assume the voltages 0 V (off) and 3.3 V (on), the output current is limited to approximately +/-10mA

Pwm_8 and Pwm_16



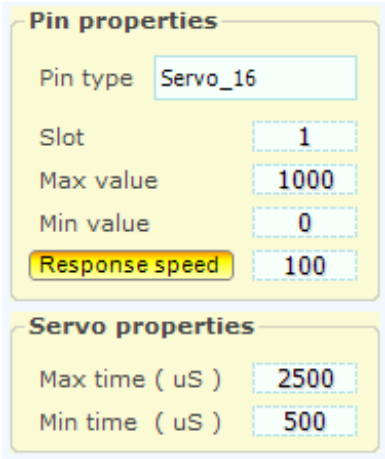
This type of Pins provide a PWM (pulse width modulation) output.

The value coming from a Slot, limited between "Min value" and "Max value" and filtered by "Response speed", is converted to pulses of width between "Min time (uS)" and "Max time (uS)"

The frequency of the pulses is 4000uS (250Hz) fast enough to turn on a LED with variable intensity. For users who require a real variable voltage a low pass filter is added, usually composed of a resistor and a capacitor.

The Pin delivers pulses between 0V (off) and 3.3V (on), the output current is limited to approximately +/-10mA

Servo_8 and Servo_16



This type of Pin, directly drives servo commands.

The value coming from a Slot, limited between "Min value" and "Max value" and filtered by "Response speed" is converted to pulses of width between "Min time (uS)" and "Max time (uS)"

The pulses repetition time is adjusted to normal aero-model servo, spinning around 180 degrees, between the min and max time.

The Pin provides voltages of 0 and 3.3 volts, suitable for all normal servos, powered from 3 to 6 volts and a current sufficient to drive tens of servos in parallel.

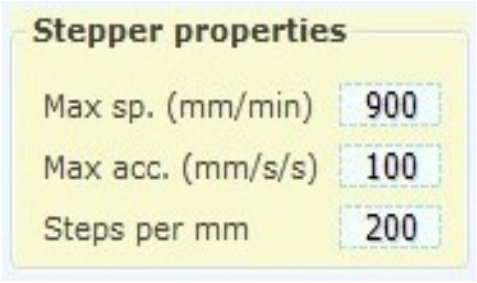
The "Output" Pin types --> Stepper

Each stepper motor needs two physical Pins, one for the STEPS and one for the DIRECTION. The micro-controller would place the Pins to pleasure, but we decided to limit confusion, by specifying preset positions, for Stepper and Stepper_Dir Pins (positions are: 1-2, 3-4, 5-6, 7-8, 9-10).

The Pin type Stepper reading from a value, which is simply the destination in mm. Simple applications, can specify a destination far away, and let it do all the firmware. Most demanding applications, can calculate their own path and send frequent intermediate destinations. With this technique, an application can check the working speed (feed), and determine precisely the path, even in multiple dimensions. To get smooth motion just 20 destinations per second (up to 50 for the most demanding applications).

To reverse the direction of movement of an axis, they swap the values “1000” and “0”, of boxes “1000 means mm” and “0 means mm”.

Specific parameters of the Stepper Pins



Max Speed - This is the fastest speed, in millimeters per minute. The firmware continuously checks the destinations sent by software. If the software is asking too much for the engine, the firmware restricts his speed, to avoid losing steps. Raise this value until you see that the motor will lose steps (It makes a high pitched noise and stops) and then decrease it by a 20..50%, to return to a safe area. Repeat the tests under load, or by braking the motor manually, so make sure you have some room.

Max Acc - This is the maximum acceleration (and deceleration), in millimeters per minute. The firmware continuously checks the destinations sent by software. If the software is asking too much for the engine, the firmware restricts its acceleration, to avoid losing steps. Raise this value until you see that the motor will lose steps during changes of direction (It makes a high pitched noise and stops) and then decrease it by a 20..50%, to return to a safe area. Repeat the tests under load, or by braking the motor manually, so make sure you have some room.

Steps for mm - Here you have to set the step, the engine is in a spin, multiplied by the microstep, set in the controller, and divided into millimeters, produced by a rotation of motor. If each spin, produces a millimeter of movement, and the engine is a 200 steps per revolution, and don't use the microstep, then you set the value: 200 (steps per revolution) x 1 (microstep) / 1 (mm per revolution) = 200. If using sixteen microstep then you set the value: 200 (steps per revolution) x 16 (microstep) / 1 (mm per revolution) = 3200.

Linked to previous - Some HAL versions have this obsolete command. It will not be implemented.

The Stepper_Dir Pin type have no parameters to adjust. I'm just a placeholder for the physical output Pin, establishing the direction of the motor. It is not necessary to use the value, that these Pins are writing into the Slot, but some applications may find it useful. The value that is written into the Slot, is the distance to the destination, in millimeters (and up to fractions of a thousandth of a millimeter). This information can be used for diagnostic purposes, or for algorithms that must meet a specified tolerance. With this information the software can work with closed loop and always at maximum speed. Continually checking distance of each engine by the target, the software can slow down exactly when you need it, without doing complex calculations of speed, trajectories and accelerations.

The "Output" Pin types --> Stepper and Stepper_Dir

The Stepper values

The value read from the Slot, is related (with “1000 means mm” and “0 means mm”) and transformed into a value between zero and one. If you set “1000 means mm” = 1000 and “0 means mm” = 0, then do not run conversions of scale and the value that comes out of the Slot is considered “mm”.

From here on, the value is always in millimeters. “Zero” indicates zero millimeters and “one” indicates 1000 mm. This value is not limited to between zero and one, but between two billion positive step, and two billion negative step. If you are using “Steps for mm = 200” the limits are: +10 Km and -10 Km.

The value is then filtered with an IIR filter (linear or growth), with adjustable “Response Speed”. The output value of the filter is called “Filtered”

The final value that is sent to the hardware is a STEP number (pre-multiplied by the value “Steps for mm”) and represents the “destination”.

The special value NAN_Reset, has the special meaning of resetting the axis. When you write a Reset, a Pin Stepper Slot, the motor stops immediately. Subsequently, the first value that will be written into the Slot, will be the value “zero reference”. The NAN_Reset is available in Theremino Automation as “Reset”, or in the new class “ThereminoSlots”, available with the source code of Theremino Automation.

The Stepper_Dir values

Each Stepper_Dir Pin is always associated with a Stepper Pin.

It is a particular Pin, either exit or entry. It provides the motor the direction of the electrical signal (so an output) but at the same time acts as an input and provides information to the software.

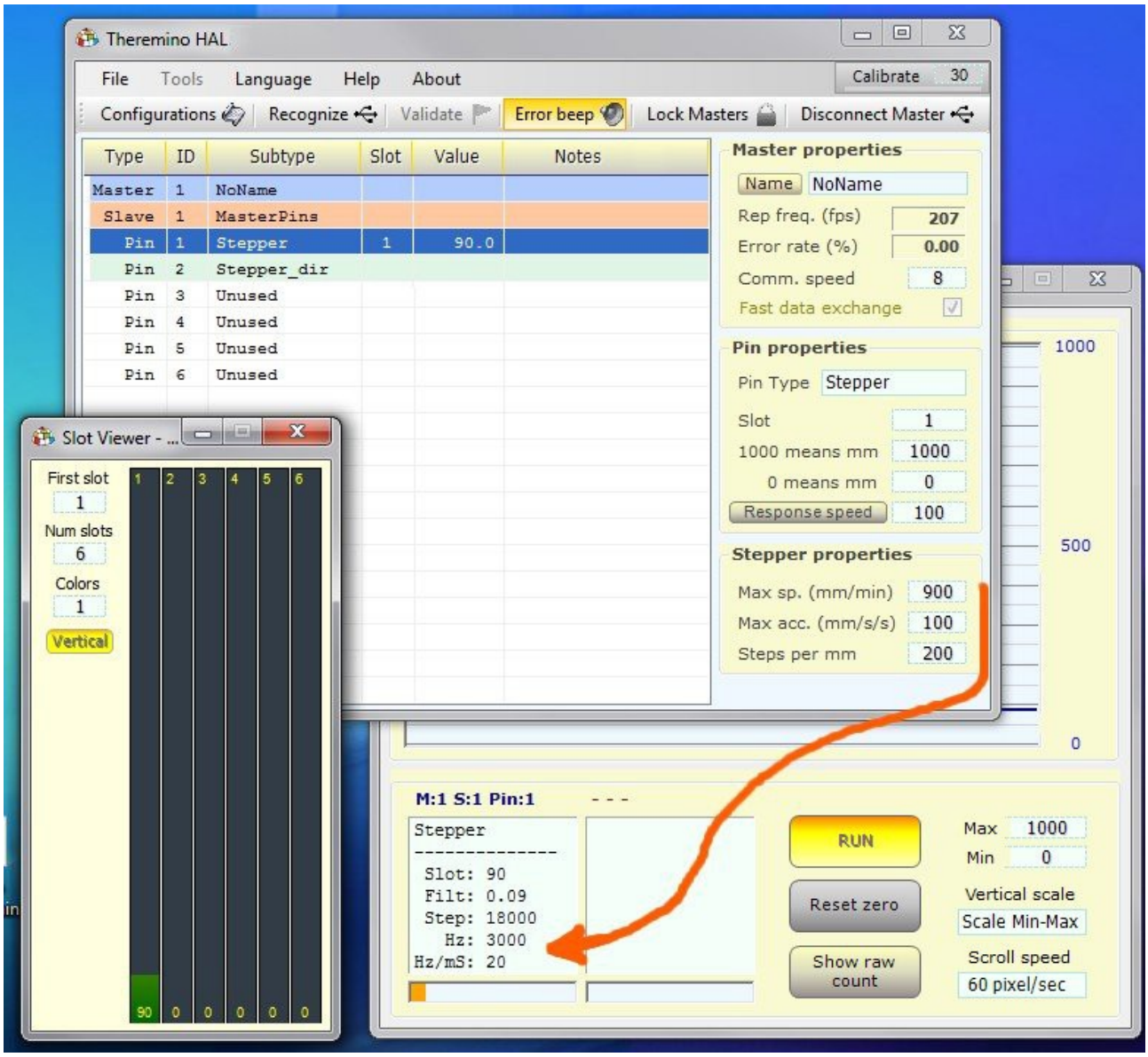
The raw value that is read by the hardware, is the number of steps (positive or negative), missing to reach the “destination” specified. The HAL application calculates mm (and fractions), by dividing the raw value, for the value “Steps for mm”. Finally this value in millimeters, is written into the Slot, and can be read by other applications, normally a CNC application.

The CNC application, knowing the remaining distance and destination (specified by herself), can calculate, with a simple subtraction, the actual location of the engine. Knowing the location of each engine, in every moment, control algorithms are simplified and their operation is more accurate.

- - - - - - -

More detailed info about Stepper Pins and about stepper motor drivers, here:
http://www.theremino.com/en/hardware/outputs/motors
http://www.theremino.com/en/technical/pin-types

The "Output" Pin types --> Stepper details



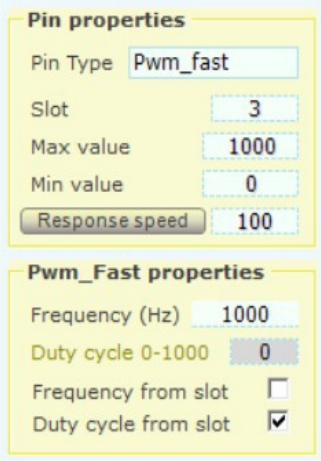
To show Pin details double click on the line of the stepper Pin. In the second window, at the bottom, you read the details of the selected Pin.

Pin details show the product of the “Max Speed” and “Steps per mm” in Hz (steps per second). These values are useful during testing and to determine how many micro-step to use. In some cases, It can be useful to know the rough target (in steps), in place of destination in mm.

The max step frequency is 65000 Hz, overcoming this frequency you are warned by boxes "MaxSpeed" and "Steps to mm" that turn yellow and orange.

In this case you must reduce "MaxSpeed". Or you can reduce the "Steps per mm", decreasing the microsteps adjustment on the driver.

The "Output" Pin types --> PwmFast



The minimum frequency is 245 Hz and maximum about 5.3 MHz. The Duty Cycle goes from zero (output signal always low), until 100% (output signal always high).

Enabling “Frequency from Slot” the incoming value sets the frequency. The incoming values from the Slots, usually between 0 and 1000, are filtered and transformed into a frequency value, between “Min value” and “Max value”.

Enabling “Duty cycle from Slot”, the incoming value sets the duty cycle between low and high signal. The incoming values from the Slots, usually between 0 and 1000, are filtered and then multiplied or divided, by changing “Min value” and “Max value”. Normally you set Min = 0 / Max = 1000 and the Duty Cycle adjusting, providing values to 0 to 1000.

The granularity of the regulations depends on the frequency set.

- At 1000 Hz the Duty Cycle precision is 16 bits (errors: 0.0015%) and the frequency is 14 bit (errors: 0.006%)
- At 16 KHz the Duty Cycle precision is 12 bits (errors: 0.024%) and the frequency is 10 bit (errors: 0.1%)
- At 1 MHz the Duty Cycle precision drops to just 6 bits (errors: 1.5%) and the frequency only 4 bit (errors: 6%)

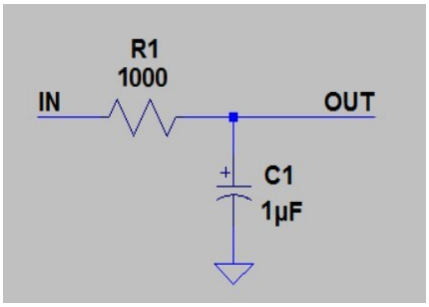
Because of granularity the higher frequencies are: 5.333 MHz / 4 MHz / 3.2 MHz / 2.666 MHz / 2.286 MHz / 2 MHz / 1.777 MHz / 1.6 MHz / 1.454 MHz / 1.333 MHz / 1.231 MHz / 1.066 MHz / 1 MHz

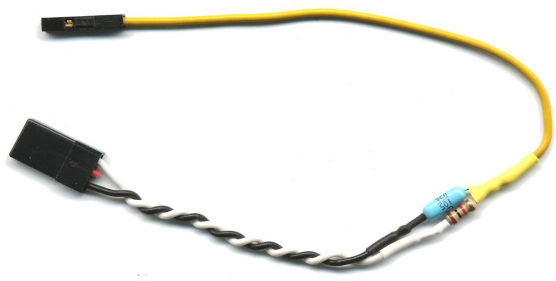
For further info about Pins, visit this page:
http://www.theremino.com/en/technical/pin-types#precision

- - - - - - -

With the PwmFast Pins and a simple adapter, you obtain an analogic voltage, trimmable with precision from 0 to 3.3 Volt

Set the frequency to 15 KHz and enable “DutyCycle from Slot”.

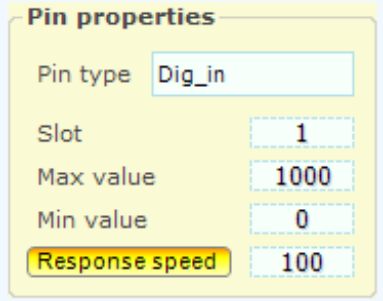




For further info about Adapters, visit this page:
http://www.theremino.com/en/hardware/adapters

The "Input" Pin types <-- Dig / ADC / Cap / Res

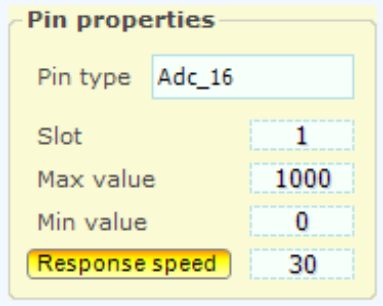
Dig_in and Dig_in_pu



This type of Pin provides a digital input.

The voltage value is read with a Schmitt Trigger with low threshold = 1 Volt and high threshold = 2Volts, and transformed into a On/Off information and finally to "Max value" or "Min value". The value is then filtered with "Response speed" and finally written into the Slot. The filtering produces intermediate values, roughly proportional to the ratio of time, between On and Off.

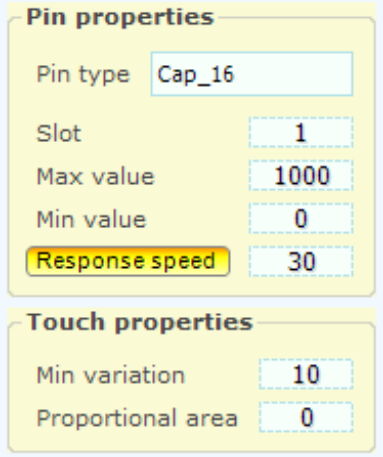
Adc_8 and Adc_16



This type of Pin provides an analog input.

The voltage value from 0 Volt to 3.3 Volt is transformed into a number between "Min value" and "Max value". The value is then filtered with "Response speed" and written into the Slot. The filtering reduces the noise present in the input signal, but slows down the response. The value 30 represents a good compromise between speed and noise.

Cap_8 and Cap_16

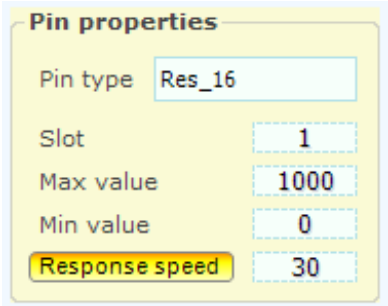


This type of Pin allows to read from improvised keys, same as Makey Makey (*http://vimeo.com/60307041#*) but with superior performances. (keys are not resistive but capacitive, they can be adjusted therefore, to work touch on, without contact, through an insulator and without an additional earth connection).

In addition to the ON-OFF setting of a Makey Makey, you can get a gradual control as well, same as with sliders, allowing to control "expression", with the speed of keys pressing, or reading raw capacitive values, such as humidity sensors.

More information on these keys , at pages 16, 17, 18, 19 and 20

Res_8 and Res_16

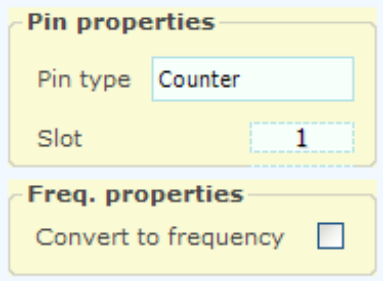


This type of Pin allows to read a resistance value, between zero and 50 Kohm. Very useful for reading potentiometers, using only two wires. Not using power, the interference caused by USB's 5 volts, is eliminated without adding a regulator and with no need to connect to the already regulated 3.3Volts, available on the special Pins of the Master.

Experiments in finding the acupuncture points and the classics jars of the Scientology meter, gave interesting results.

The "Input" Pin types <-- Counter

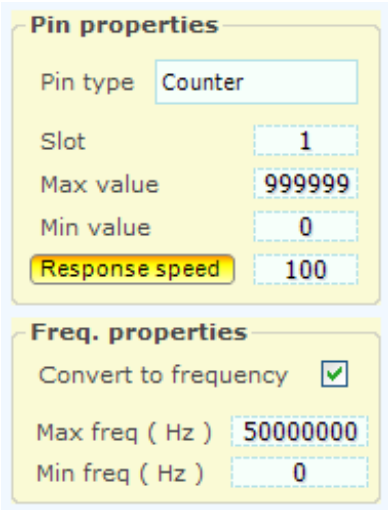
Counter and Counter_pu



All Pins can be programmed as Counter or Counter_pu but the maximum counting speed is quite limited, around a few KHz, depending on the load on the microcontroller and the duty-cycle of the signal.

If you need a higher speed you should use a FastCounter.

Counter and Counter_pu with the "Freq"



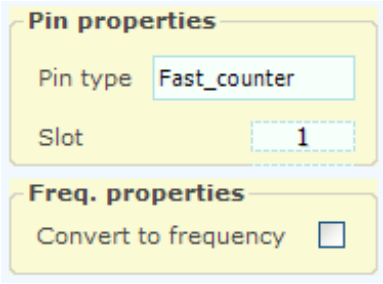
Pins programmed as Counter or Counter_pu, can be transformed from counters to frequency-meters.

The frequency value, limited between "Min Freq" and "Max Freq", is then compared between "Min value" and "Max value" filtered with "Response speed" and finally sent to the Slot.

The "Counter" and "Counter_Pu" Pins, use 16 bits for data transmission.

The "Input" Pin types <-- Fast_counter

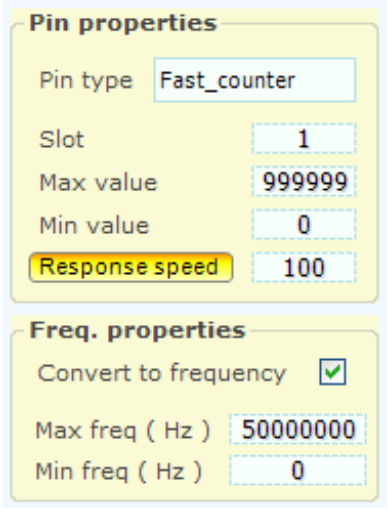
Fast_counter and Fast_counter_pu



Only a few Pins, can be programmed as Fast_counter or Fast_counter_pu

On the other hand, their maximum counting speed is 50 MHz (for maximum counting speed, duty-cycle must be 50%)

Fast_counter and Fast_counter_pu, with "Freq" option



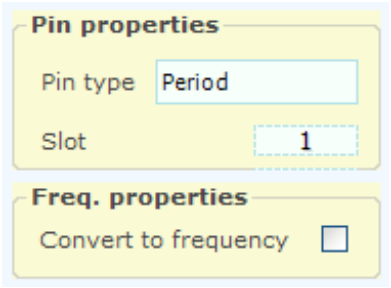
The Pins programmed as Fast_counter or Fast_counter_pu, can be transformed from counters to frequency-meters.

The value of frequency, limited between "Min Freq" and "Max Freq", is compared between "Min value" and "Max value", filtered with "Response speed" and finally sent to the Slot.

"Fast_counter" and "Fast_counter_pu" Pins, use 16 bits for data transmission.

The "Input" Pin types <-- Period

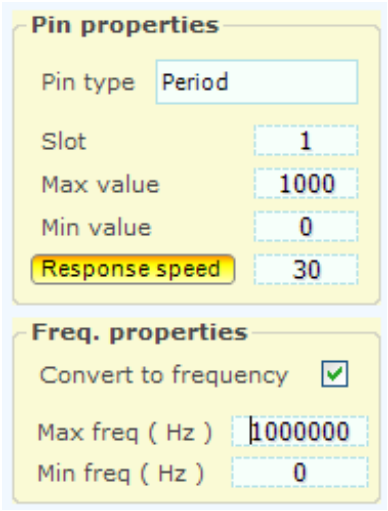
Period and Period_pu



This type of Pin, measures the period of a repetitive waveform, from peak to peak, up to a maximum period of about 260 seconds.

Resolution is half a microsecond and accuracy is +/-1%, over a range of temperature from 0C to 50C.

Period and Period_pu with "Freq" option



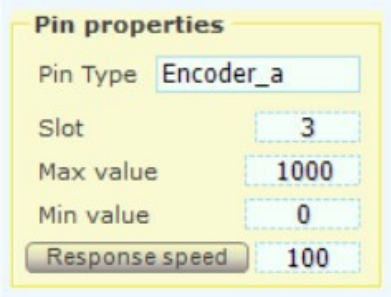
Pins programmed as Period or Period_pu can be transformed from counters to frequency-meters.

This technique, allows to measure very low frequencies (up to about a tenth of a Hertz) with very high resolution.

The value of frequency, limited between "Min Freq" and "Max Freq", is compared between "Min true" and "Max value", filtered with "Response speed" and finally sent to the Slot.

"Period" and "Period_pu" Pins, use 32-bit for data transmission.

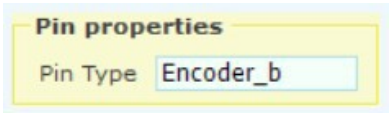
The "Input" Pin types <-- Encoder



All Master modules Pins can be programmed as Encoder.

For each Encoder two input Pins are needed: Encoder_A, and Encoder_B, or Encoder_A_Pu and Encoder_B_Pu.

This pair of inputs reads the two phases of the “quadrature type” Encoders. The count of the Encoder is written into the Slot associated with the Pin “Encoder_A”.



Each “Encoder_A” Pin uses 16 bits for data transmission, while “Encoder_B” Pins are only a placeholder, not sending data.

- - - - - -

The encoders read the angular position of a PIN, as potentiometers, but the number of turns is unlimited.

There are encoders similar to small potentiometers (the best known are the KY-040 of the following images). These models are mechanical and provide 18, 20 or 24 pulses per revolution, depending on the model. The firmware obtains 72, 80 or 96 angular positions per revolution, from these pulses.

The maximum count rate is limited around 10 KHz, and is dependent on the micro-controller load. So you should either use Encoders with a few steps per revolution, or limit the rotation speed, or demultiply them.

The encoder generates a count from 0 to 65535 (16 bit). When the count exceeds 65535 the number restarts from zero. This system allows many applications, to read asynchronously the progressive number, without losing counts.

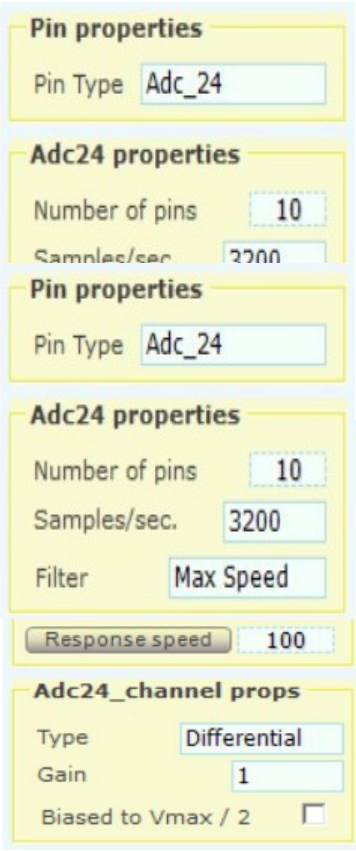
For further info about Encoders, visit this page:
http://www.theremino.com/en/hardware/inputs/sensors#encoders

The "Input" Pin types <-- Adc24

The ADC module 24 is connected to the Pins 7, 8 and 9 of the Master. To activate the module, select the Pin 7 line and set its Pin Type as "Adc24".

Pin	7	Adc_24			
Pin	8	Adc_24_din			
Pin	9	Adc_24_dout	9	133.3	

When Adc24 works, the value that is read on Pin 9 (Adc_24_dout) increases in proportion to the sampling rate (with 100 sps and MaxSpeed filter, the value grows by 100 readings per second).



Pin 7 properties when configured as Adc24

Number of pins: Enables 1 to 16 analog input lines.

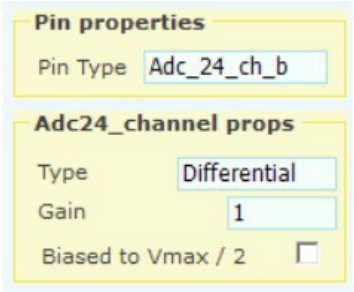
Samples per sec: Sampling rate that is divided among all the active inputs (for example with 600 sps and three inputs, each input is sampled 200 times per second).

Filter: Eight filters are available to choose from for the best compromise between noise and speed of response.

Properties of the “Adc_24_ch” Pin

Important to note that the 16 input Pins are paired (1-2, 3-4 ... 15-16), and then their type (Differential, Pseudo and Single) and the gain (from 1 to 128), are the same for both the inputs of the pair.

The "Biased to Vmax / 2" can, instead, be activated separately, on each of the sixteen inputs.



Properties of the “Adc_24_ch_b” Pin

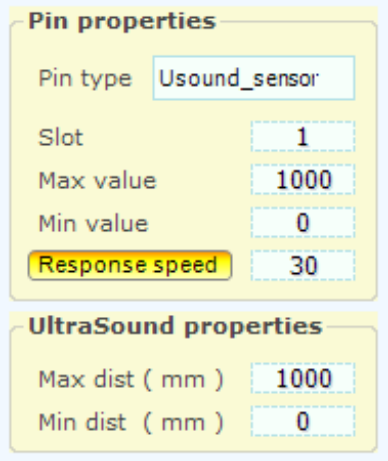
If the pair of inputs is Differential, the second input of the pair becomes "Adc_24_ch_b".

This is a special type that does not send data to the slot, and is only a placeholder, for the sensor reference connection.

More about the Adc24 module and operation guidelines, with detailed explanations and examples:
http://www.theremino.com/en/hardware/adapters#adc24

The "Input" Pin types <-- Usound and CapSensor

Usound_sensor



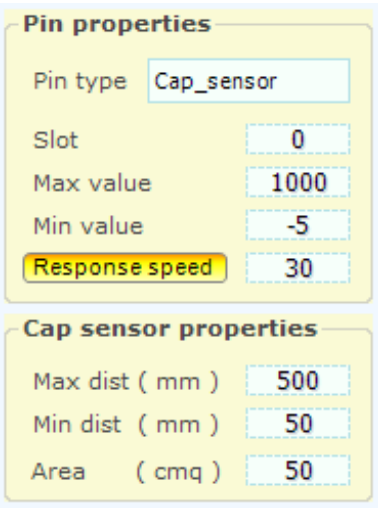
Many ultrasonic distance sensors, such as model SRF05, can be read with this type of Pin.

This type of Pin generates a positive pulse of "Start" every 33mS (approximately), and measures the time of return of the pulse , from 0 to 32000 microseconds.

The time is then converted by "HAL" in distance, taking into account the speed of sound in the air.

The "Usound_sensor" Pin-type uses 16 bits for data transmission.

CapSensor_HQ



This kind of Pin is used to measure the distance of a conductive object (typically a hand).

The detection is stable and precise, with a fast response time, in the order of milliseconds.

The value of distance between "Min dist" and "Max dist", is compared between "Min value" and "Max value", filtered with "Response speed" and finally sent to the Slot.

The "CapSensor_HQ" Pin, use 24 bits for data transmission.

Please note: CapSensor values "Min dist" and "Max dist" are only approximate. The exact range is not important, as this is not a measuring device. Future improvements of the linearization formula, could possibly improve the precision, mainly at small distances.

Resistive or capacitive buttons

To see what you can do with simple buttons, watch this great video of the Makey Makey: ***http://vimeo.com/60307041 #***

The buttons on the Makey Makey are resistive and not capacitive, it will only work if the resistance is less than about 4 Mega Ohm, it needs an additional wire as ground reference and doesn't work through insulating materials such as plastic. Furthermore, the buttons on the Makey Makey are only six (not expandable), each Makey Makey can provide up to 20 keys, you can connect more Makey Makey in chain, but at the end, all keys are sent to the keyboard that manages just a maximum of six: ***www.makeymakey.com/faq*** Finally, the Makey Makey keys have only on/off operation, without intermediate adjustments and do not feel keys pressing speed (Velocity).

Theremino System capacitive keys, can do much more. They can be expanded at will, by adding Master modules (6 keys each) or Servos (8 keys each) in an unlimited number, as shown here: ***www.youtube.com/watch?v=NbC5kIRS_6s*** and here: ***www.youtube.com/watch?v=2RzwUfXhFZY***

Moreover, Theremino System keys, can provide a gradual control as well, like if they were sliders, and control the "expression", determined by the keystrokes speed.

The three types of capacitive keys

- **On/off keys**
  "Min variation" from 10 to 50
  "Proportional area" should be ZERO

- **Proportional keys**
  "Min variation" from 10 to 100
  "Proportional area" from 100 to 200 (for a maximum of about 1000)

- **Keys with velocity**
  "Min variation" from 25 to 50 (adjusted for maximum output)
  "Proportional area" -30 (adjust to a maximum of about 1000)

Generic capacitive measuring

- **Capacitive sensors (humidity sensors, variable caps. etc.)**
  "Min variation" from -1 to -1000 (minimum value setting)
  "Proportional area" from 1 to 1000 (maximum value setting)

Caution: with this type of Pin is not obtained a measure of the electrical capacity, but only the value of a sensor or a mechanical position. Many factors contribute to get a non-linear measurement, first of all the capacity of the connection cable. The cable must be very short, and after calibration, you should not move it. In all cases it will have to make adjustments of scale and appropriate linearizations in the software.

"Min Variation" and "Proportional Area" parameters

Min variation eliminates small variations and prevents electrical noise from triggering keys, without having touched them. Raising this parameter, keys become less sensitive. It should keep as low as possible, just enough to eliminate all noises.

For keys with velocity, the best setting for this parameter is obtained by pressing the button quickly and repeatedly and adjusting "Min variation" with the mouse wheel, in order to get the maximum output signal. To make this adjustment easier, temporarily set "Proportional area" with a negative number large enough, for example -50.

Proportional Area is set at about 1000, when the finger is in the maximum position of the slider, or when you press buttons, as fast as possible.

This value should normally be higher for Pin 1 and 2 (less sensitive), or in case of long wires and large objects.

Zero calibration of the capacitive buttons

If you change the mechanical arrangement of keys or their position, if you move the wires that connect them or if you approach metal objects, while the HAL program is working, it is possible to lose the zero calibration of the keys. If zero is not well calibrated, capacitive keys can become less sensitive or even not work at all.

If you remove capacity from the keys (shortening the wires or take them away from metal objects) calibration is automatically made immediately, but it is impossible to distinguish an increase in capacity due to a finger or a shift in the wires.

We have tried many methods of automatic recognition, with slow drift or timed calibration, but none worked well and all compromise the accuracy of the normal keys operation.

You should therefore, try not to move the wires of the keys, the keys themselves and conductive objects within a radius of about ten centimeters, during operation.

To check if a key is calibrated, release your hands from the button and verify in the details of its Pin, that the values "Smoot" and "Mean" are equal to each other, or are very close (not more than one point of difference)

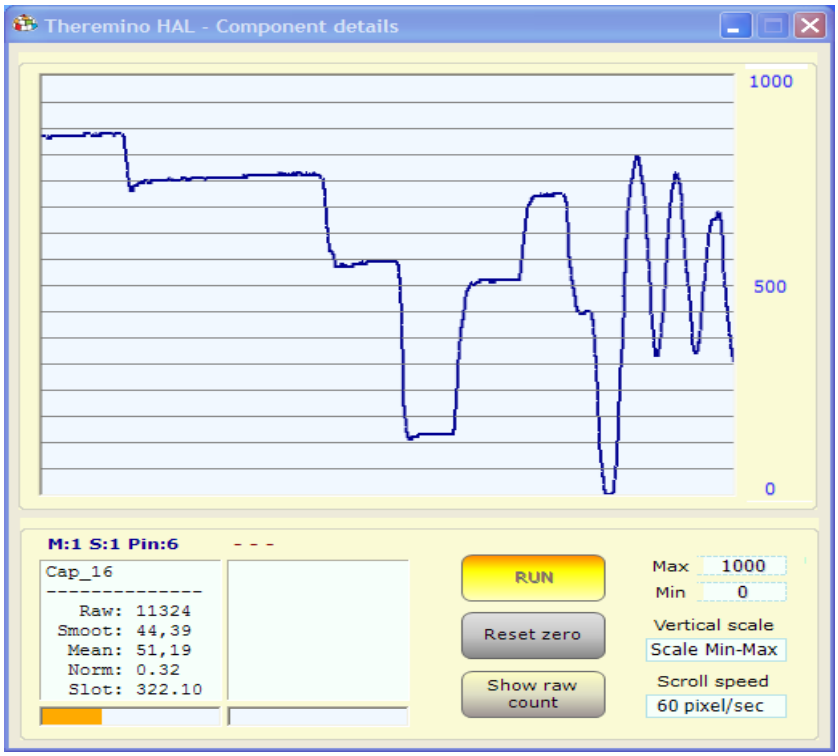
When in doubt, press Calibrate (keep your hands away from the keys while performing the zero calibration)

Reading of capacitive sensors

Setting MinVariation with a negative value completely changes the operating mode. For example, humidity sensors (capacitive models and without control circuit) could be connected. It could also improvise sensors to read the rotation of a Pin or linear displacements. Sensors of this type may be simple, but also very reliable.

Setting Min Variation with a negative value, the meaning of Min Variation and Proportional Area changes:
- Min Variation sets the minimum and Proportional area the maximum, of measurable capacity.
- The calibration button is disabled. The calibration is fixed and is the Min Variation value itself.
- The range of usable capacities is from a few pF to a few nanoFarad.

Capacitive keys of the type "Slider"



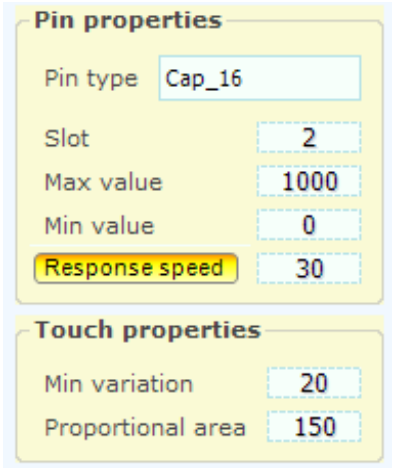
"Proportional area" must be a positive number, this determines the "Proportional" operation.

With a capacitive button of this shape, a continuous adjustment similar to a cursor "slider", can be obtained.

The control is carried out with a finger, all top=1000, all bottom=0

These keys are suitable for volume control and can act as a "panic button" (when you unplug your finger from the button, the volume is zeroed)

These are the standard settings for buttons, of the type "Slider" (note 1)



"Max value" normally set at 1000 (Note 2)

"Min value" normally held at zero (Note 2)

"Response speed" is normally set to 30 (light filtering)

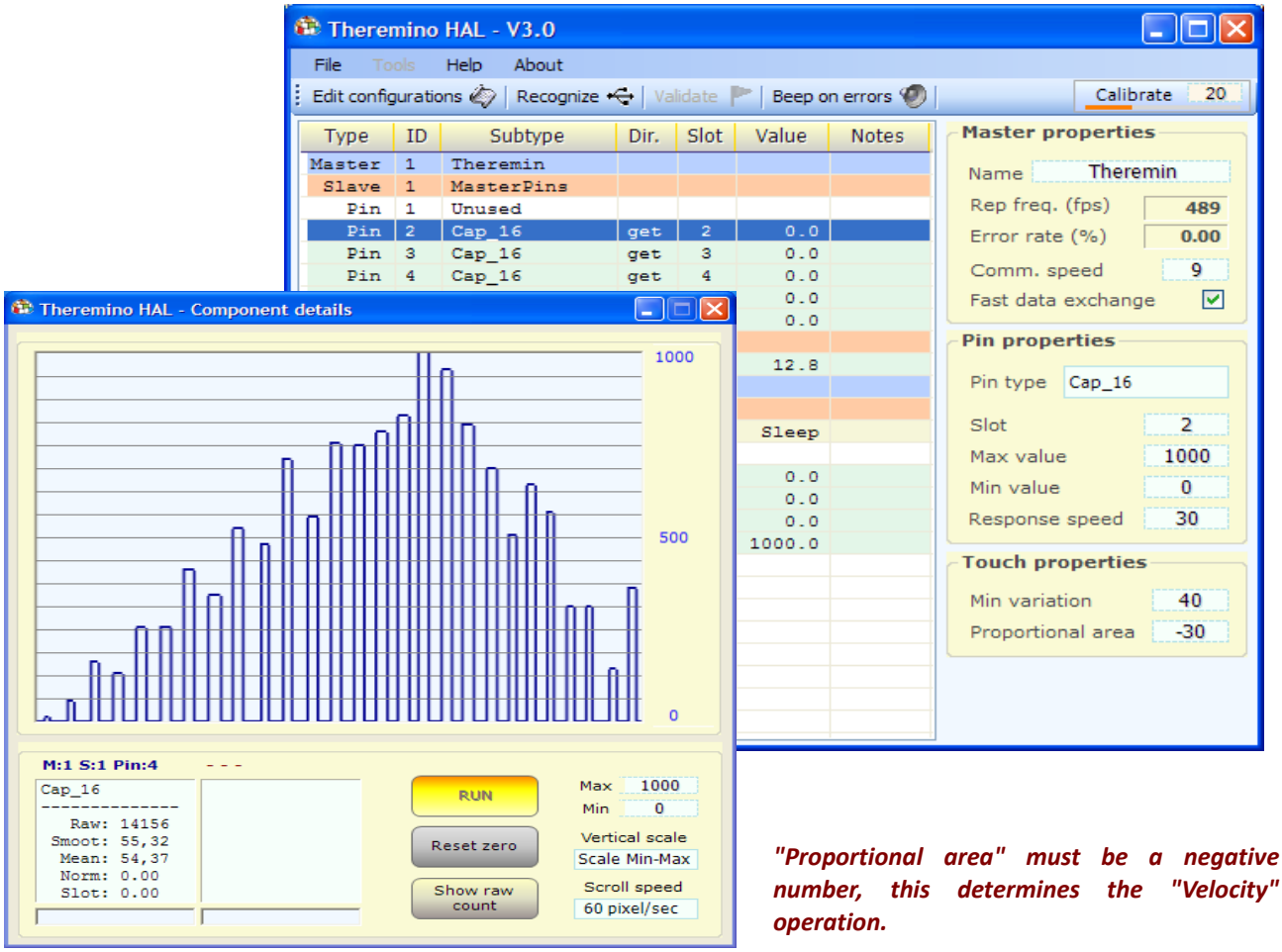
"Min variation" is set normally from 10 to 100 (better to raise it slightly to obtain the maximum sensitivity in the lower part)

"Proportional area" is normally set to 200 (about 100 for less sensitive keys or with long wires)

(Note 1) *For keys of the "Slider" kind, it is always better to use "Cap_16"*

(Note 2) *To reverse the output signal, it can be exchanged Min with Max (Max = 0 and Min = 1000)*

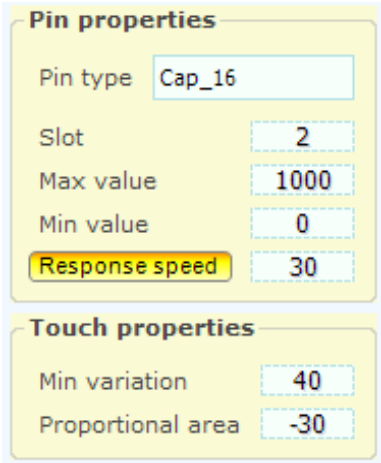
Capacitive keys with "Velocity"



"Proportional area" must be a negative number, this determines the "Velocity" operation.

Keyboards that allow you to play notes loud or soft, depending on how you press the keys, are very popular for musical applications. Capacitive buttons can be set to measure the speed of a key and turn it into a value from 0 to 1000 (approx).

For a good operation of the "Velocity" the communication speed needs to be high (200 to 500 fps), and the keys, must be adjusted one by one, to obtain this way, a maximum value slightly over 1000.



These are the settings for the keys with "Velocity"

"Max value" normally held at 1000 (Note 1)

"Min value" normally set at "0" (Note 1)

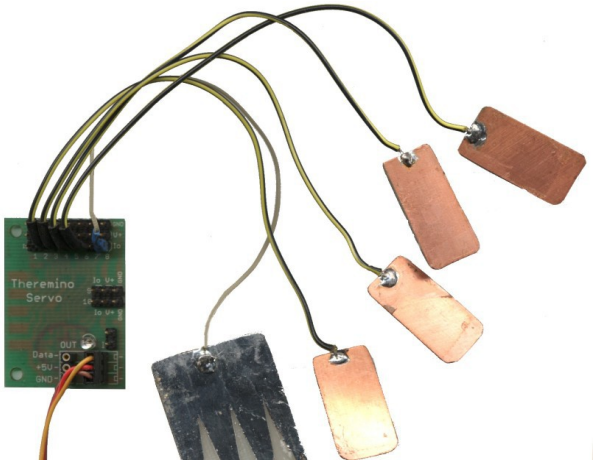
"Response speed" normally set to 30 (best not to edit)

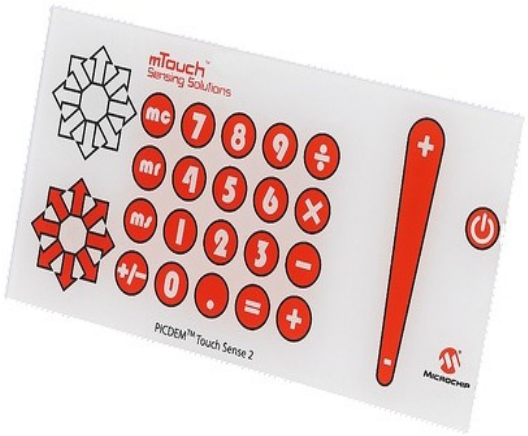
"Min variation" is normally set to 50 (about 25 for keys 1 and 2, which are less sensitive or for keys with long wires)

"Proportional area" is set normally to -40 (about -20 for buttons 1 and 2, which are less sensitive or keys with long wires)

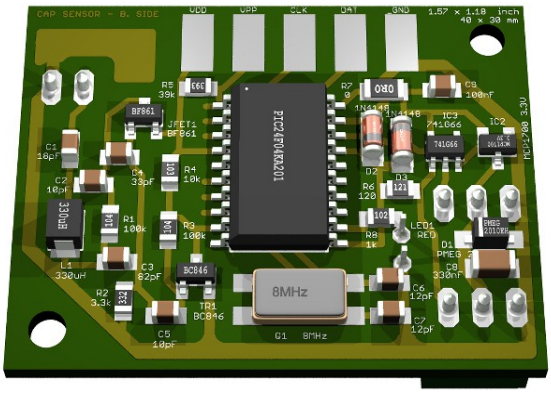
(Note 1) *To reverse the output, signal can be exchanged from Min to Max (Max=0 and Min=1000)*

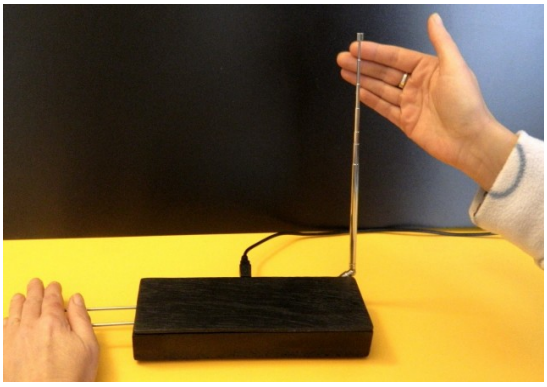
Differences between CapacitiveKeys and CapSensors





Capacitive keys cannot replace CapSensor modules, the first work only at short distances (from a few millimeters to a few centimeters), while CapSensors work up to distances of several meters, and can be adjusted for an almost perfectly linear response. The capacitive keys on the other hand are much cheaper and are better suited to arrange keyboards with many keys.





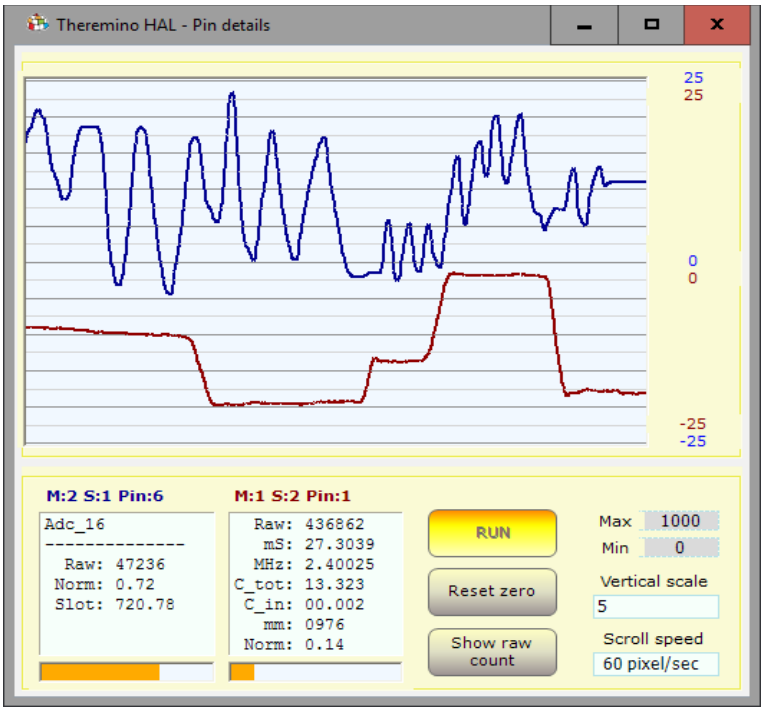
Mechanical construction of the capacitive keys

Make sure that capacitive keys are well isolated, otherwise it may be enough a small spark of static electricity, to produce communication errors. Nothing is broken, but communication is interrupted and you have to press the button "Recognize".

Face them with copper below and vetronite (thin) above, or even better, copper held above, but a thin sheet of insulating plastic is added, possibly printed in color with the shape of the keys, as in the image at the beginning of this page.

The wires going from the keys to the Pins, must be as short as possible and there must be at least 5 or 10 millimeters, between them. The key operation and the insulation from disturbances, improves by decreasing the capacitance. Experiments were conducted in "impossible" situations , with long wires and any kind of keys, from potted flowers to various fruits and with individual adjustment, always was running good.

The “Pin details” display



With a double-click on a active Pin line, this instrument is opened. To display two signals click on the first Pin then on the second Pin, with a single click.

The vertical scale can be set to “Scale Min-Max”, which corresponds to the Min and Max textbox values.

Or it can be set in 24 levels from 0.01 to 50000 points per vertical division (ten dark lines). When the vertical scale is set to those vaues, to center the traces you press "Reset zero."

In some cases it may be useful to check the raw values. For "Raw" values, use the "Show raw count".

The "Scroll speed" adjusts the graph scroll speed, from 0.1 pixel/second to 60 pixel/second.

The two text boxes, show the internal details of the Pins. The title indicates which Pin is analyzed, in this image the text "M:1 S:1 Pin:2" means "Master 1, Slave 1, Pin 2"

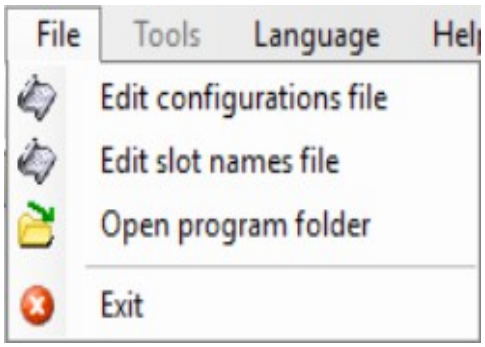
Details of the Pin, may help in the control and regulations of Input Output devices (sensors and actuators).

Some types of Pin are more complex and have more intermediate values. In general, a "Raw" value exists, with very variable values, depending on the type of Pin, a "Normalized" value which always goes from 0 to 1 and a "Slot" value which normally ranges from 0 to 1000 and that can be considered as the "Simplified" value available on Slots, easily usable by all the high-level software.

- **Raw** "Raw" value which can be a counting, a time, a voltage or other.
- **mS** Time in milliseconds
- **uSec** Time in microseconds
- **MHz** Repetition rate
- **C_tot** The total electrical capacity in parallel to the 330uH coil (used only for CapSensor)
- **C_in** The input power capacity added after calibration (used only for CapSensor)
- **mm** Approximate distance in millimeters (used only for CapSensor and ultrasonic sensors)
- **Smoot** Value passed in an FIR filter for smoothing (used only in Cap8 and CAP16)
- **Mean** Average value (used in type Cap8 and CAP16 for zero calibration)
- **Norm** Normalized value between zero and one
- **Slot** Value written to, or read from the Slot associated with the Pin (normally from 1 to 1000)
- **Out** Digitized value that can be only "0" or "1" (used only by DigOut)

Although not indicated, capacities are always in picoFarad (pF)

Menu commands



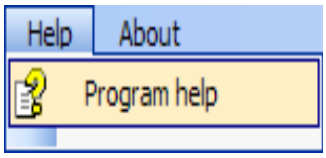
Edit configurations can be useful in some cases. Read the "Frequently Asked Questions" on the last page of this document, for more information.

Edit slot names file comments (or slot names) are explained ***in this page***.

Open program folder can be useful to modify the documentation files and languages.

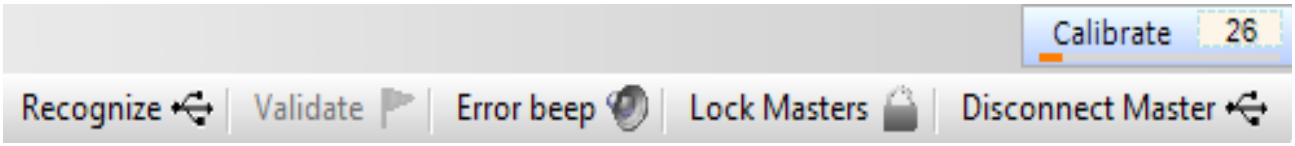
The language files are located in the "Docs" folder, near to the application ThereminoHAL.exe.

To make a new language files just copy the file Language_ENG.txt, change "ENG" with "FRA", "ESP", "DEU" or "JPN" with Notepad and edit the text.



This command opens the documentation file.

The toolbar keys



Recognize
Is useful to recognize Master and Slaves connected to the USB

Validate
When you add or subtract Slave modules from the chain, you are warned that the configuration has changed, with some red lines in the list. If you choose to loose the old configuration and accept the actual hardware, this button makes valid the new configuration.

Error beep
If pressed communication errors are highlighted with a sound.

Lock Masters
If pressed the HAL will connect only to the Masters having their names in the current list. In this way you can keep (in separate folders) different HAL applications, each linked to its specific hardware.

Disconnect Master
Removes the selected Master from the list. In this way we can eliminate the Master side without having to physically disconnect them from the USB. After being eliminated is advisable to press "Block Master", so the next boot will be reloaded the desired Masters only.

Calibrate
Sets the CapSensor and CapKeys calibration value. Remove your hands from the sensitive key before pressing it. If there are no movements greater than the set value, every 30 seconds an automatic calibration is performed. It is possible to disable the automatic calibration, with a "0" value.

Isolated applications

Some Theremino system applications automatically launch its own HAL. This happens if there is a Theremino_HAL.exe in the folder ThereminoHAL located near to your application EXE file. You could also place Theremino_HAL.exe next to the exe file of the application, but it is better that the HAL has its own folder, with the “Docs” sub-folder containing documentation and language files.

These HAL use their own private configuration and if they have the "Master Lock" button, you can only connect to its Master, identifying them by name among those connected to the USB ports. An application composite in this way, will continue to operate even when copied to a different computer, and even if other Theremino System applications are connected with their Master, on other USB ports.

The applications that benefit most from these possibilities, are applications with a specific task, such as: Theremino Geiger, Theremino OilMeter, Theremino Weather, Theremino Theremin, Theremino Arm, Theremino Geo and Theremino EmotionMeter.

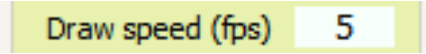
This does not mean that isolated applications can not communicate with each other. The modular communication is always possible and is done through the Slot, which are common to all applications.

To avoid using the same Slot for different tasks we have defined a broad pattern:

```
Experimental 100 slots     000 - 099
   - - -
Theremino_Theremin         100 - 199
Theremino_SlotsToMidi      200 - 299
Theremino_MusicKeys        300 - 329
   - - -
   469 free slots          330 - 799
   - - -
Theremino_OilMeter         800 - 809
Theremino_EEG              810 - 819
Theremino_Meteo            820 - 839
Theremino_Arm              840 - 849
   10 free slots           850 - 859
   10 free slots           860 - 869
   10 free slots           870 - 879
Theremino_EmotionMeter     880 - 889
Theremino_Geiger           900 - 909
Theremino_Bridge           900 - 909
Theremino_GEO              910 - 919
Theremino_GeoPreampTester  920 - 929
Theremino_Radar            930 - 939
   10 free slots           940 - 949
   10 free slots           950 - 959
   10 free slots           960 - 969
   10 free slots           970 - 979
   10 free slots           980 - 989
   10 free slots           990 - 999
```

This scheme is only indicative. You can use the Slots as you like, providing that, the same Slots are not used for different tasks in the same PC. If you make a mistake does not break anything, but the data overlap with undefined results.

Adjusting the numerical boxes



HAL numerical boxes (and all other Theremino system applications) have been developed by us (note 1), to be more comfortable and flexible, than the original Microsoft TextBox.

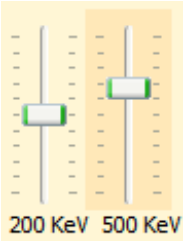
The numerical values are adjustable in many ways
- By clicking and holding down the left mouse button and moving the mouse up or down
- With the mouse wheel
- By pressing the arrow-up and arrow-down keys
- With conventional methods used to write numbers with the keyboard
- With the usual selection and copy-paste methods

- By pressing SHIFT the variation speed is multiplied by one hundred
- By pressing CTRL the variation speed is multiplied by ten
- By pressing ALT the rate of variation is divided by ten

Moving the mouse up and down allows wide and fast adjustments
The mouse wheel allows a comfortable and immediate setting
The arrow keys allows fine adjustments without having to look away from what you are adjusting

*(1) Like all our software, their source files are available (Freeware and Open Source licensed under a Creative Commons) and can be downloaded from here: **www.theremino.com/downloads/uncategorized** (See "Custom controls") These controls can be used freely in any project, without naming the source. The "Open" source, serves as a guarantee that we have not included malware.*

Adjusting the sliders



These are the original Microsoft cursors, they are pretty comfortable, so we just added the orange color and the possibility to reset them.

<<< Non-zero sliders are marked with an orange color, to reset them just click with the right mouse button (not all sliders have a zero, in this case they do not change color and cannot be reset with the mouse)

Sliders can be adjusted in the following ways
- Clicking the cursor with the right mouse button, to reset them
- Clicking the cursor with the left mouse button and moving the mouse up or down
- With the mouse wheel
- Using the left-arrow and right-arrow on your keyboard
- By pressing the up-arrow and down-arrow keys

The method of moving the mouse up and down, allows wide and fast adjustments.
The mouse wheel allows comfortable and immediate adjustments
The arrow keys allow fine adjustments without taking your eyes from what you are adjusting.
The arrow keys left/right or up/down have the same effect, it might be be more intuitive to use the first for horizontal cursors and the seconds for vertical sliders.

Questions and Answers

Can I change the text of the panels of the program, to different languages?

Of course, just edit the file: "..\Docs\Language_Eng.txt" and "..\ Docs\Language_Ita.txt"

For German, French and Spanish languages, just copy the file English three times with the following names: "..\Docs\Language_Deu.txt", "..\Docs\Language_Fra.txt", "..\Docs\Language_Esp.txt"

Can I edit the configuration file?

Normally, the association between configurations and Master and Slaves modules, is kept aligned by ThereminoHAL, which uses the names of the Master to determine the right configuration to be set. Usually HAL can use the right configuration, even if you disconnect and replace Master and Slave modules.

In some cases, if you change the Master names with a HAL installed on a different computer, or in a different folder, the alignment between hardware and configuration, can be lost. In these case, you can click on the pop-up menu of the Master name and restore the alignment by choosing the right configuration for each Master.

To make more complex changes, open the file "Theremino_HAL_ConfigDatabase.txt", with a text editor such as "Notepad" and manually edit the configurations, quite simple task.

How to reduce the CPU work?

- Close or minimize the "Component details" window.
- Minimize the main window.
- Reduce the “Comm speed”, as explained in the first pages of this document.