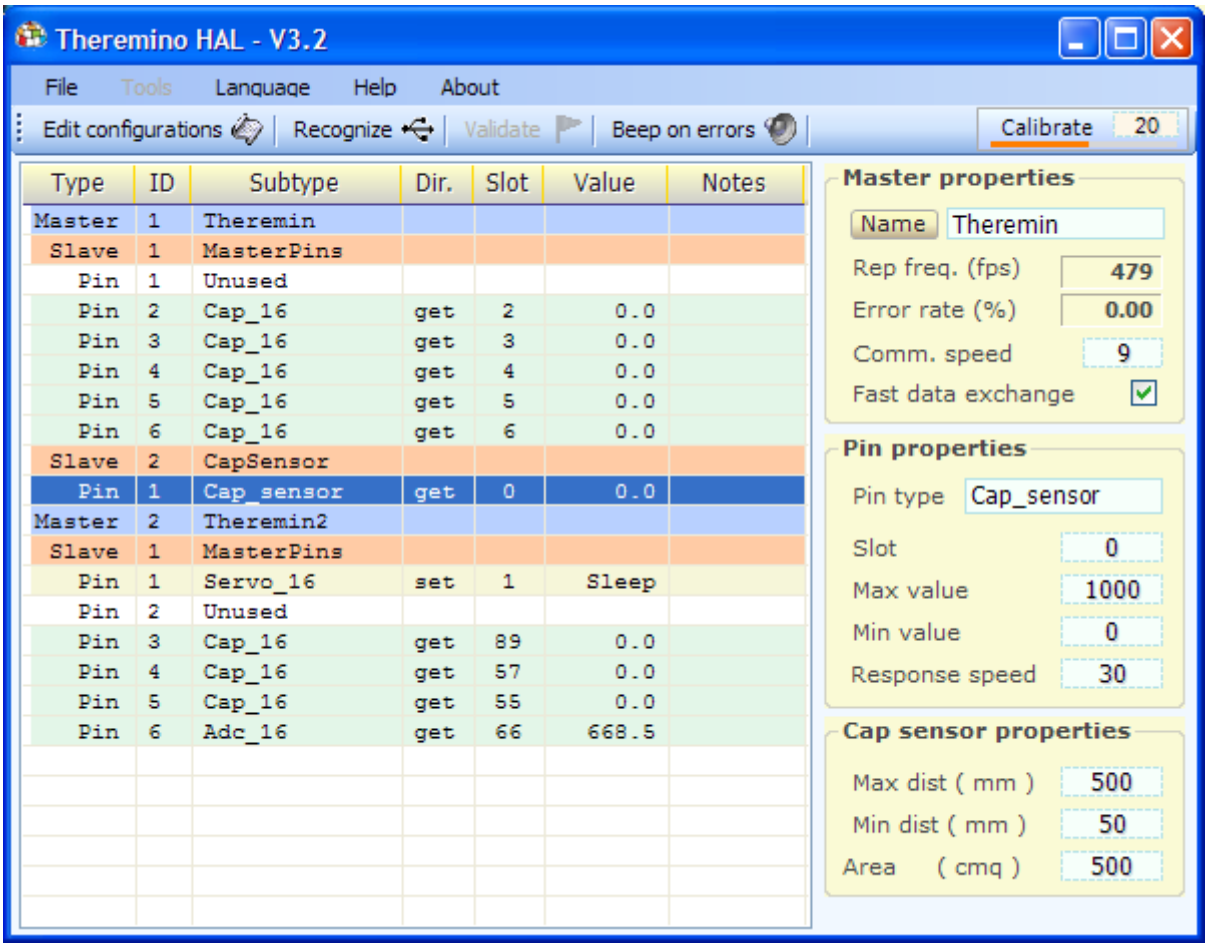
System Theremino

Theremino HAL Instructions

Theremino HAL



Theremino HAL V 3.2 with two master connected

The ThereminoHAL (Hardware Abstraction Layer) is a Hardware Manager - *How many names for a small application!* - But the HAL deserves them because, in spite of its apparent simplicity, carries out complex operations with highly optimized algorithms.

The ThereminoHAL is the heart of communication with the hardware, knows how to communicate with many Master at the same time, know the protocol and USB serial communication, knows all the most common types of InputOutput and knows how to recognize the "Slave" modules.

Without HAL communicate with the hardware would be difficult (as with Arduino), would require much time and labor (as with Arduino) and finally, for each type of InOut, to move a motor or even just to read a key, you should write firmware appropriate (such as Arduino)

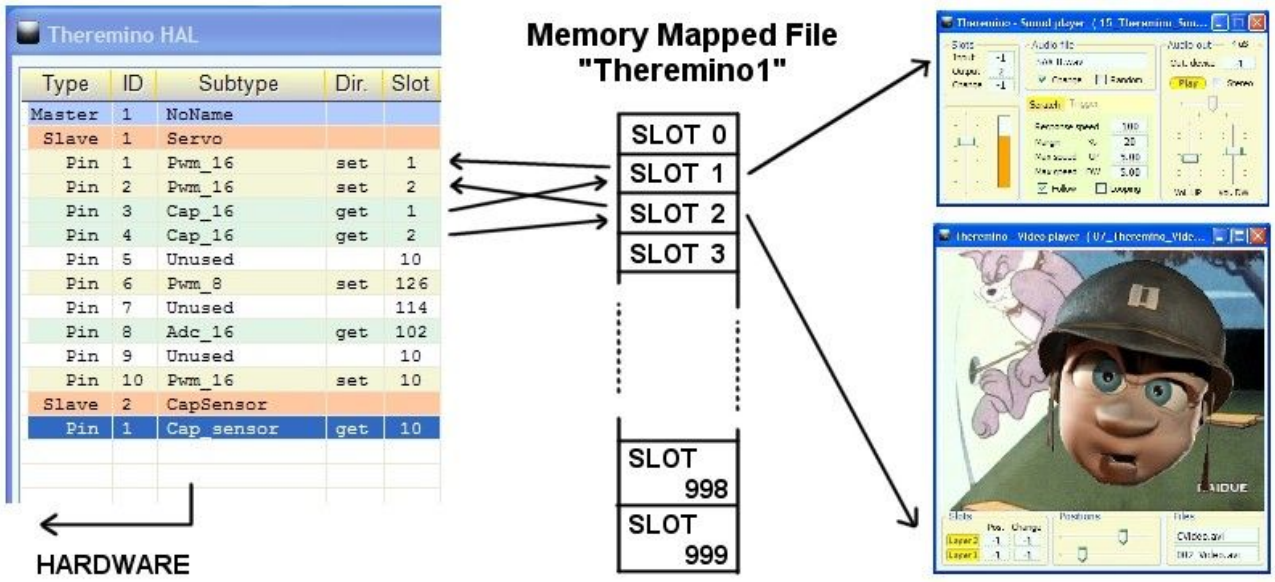
If you use the hardware modules *then the HAL is essential and must remain active, you can minimize it, but it must remain in operation.*

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

The "Slots"

The Theremino System Slots are identified by a number from 0 to 999 and are all part of MemoryMappedFile called "Theremino1"

Each slot contains a "Float" number, which can be read or written by any module of the Theremino System.



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

In choosing to use the slots 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.

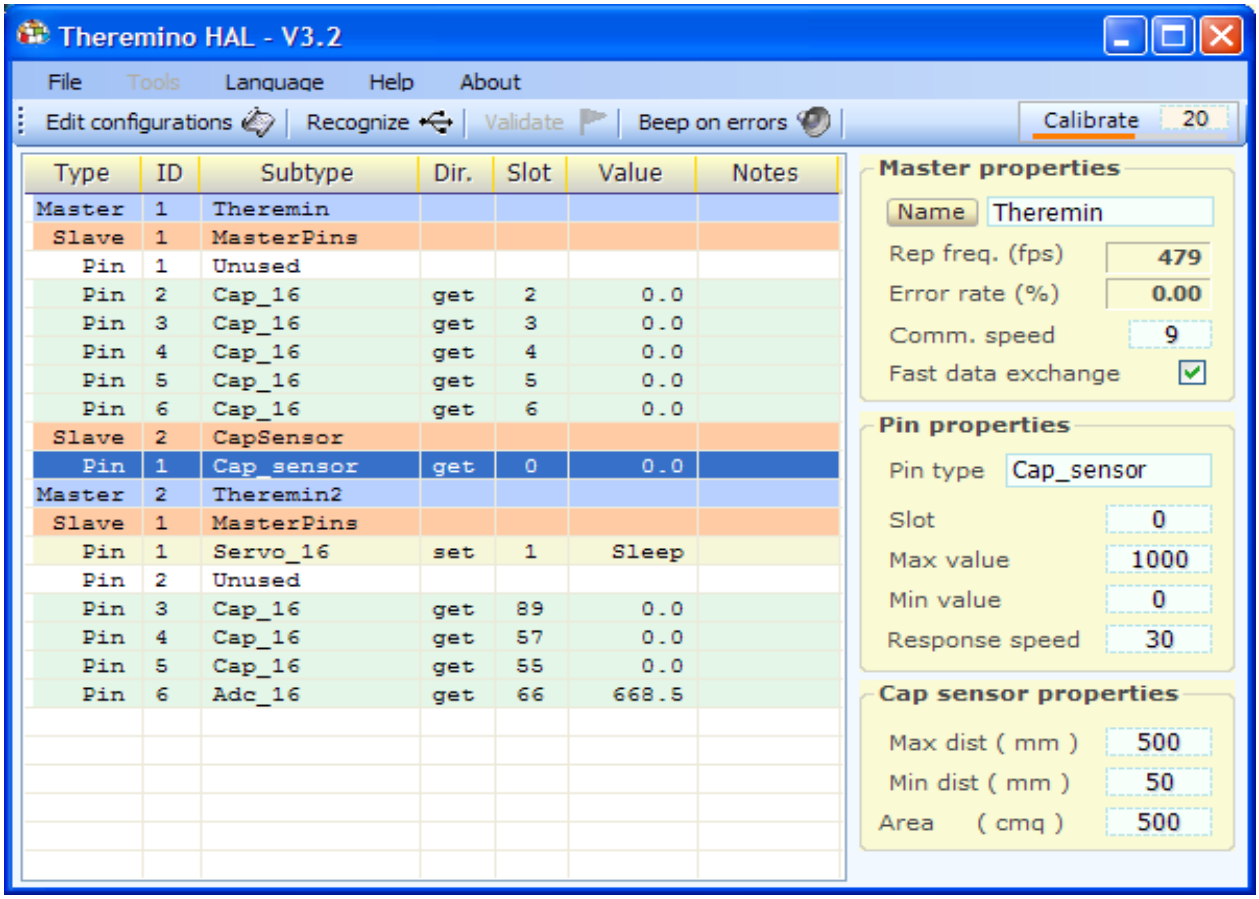
Many applications and many pins can read the same slot but should avoid configuring more than a Pin writing on the same slot, doing so does not break anything, but the results are undefined.

If you send multiple streams of data to the same slot then the data are mixed and wins the last to write, if you want to merge the data in an orderly fashion rules are required.

To establish mathematics and logic rules between the Slots, and also to write complex behavioral algorithms, we use "Theremino_Script" or any programming language such as C + +, CSharp, VBnet or VB6 but you can also use visual languages like MaxMSP, Processing, PureData, LabView and EyesWeb. For MaxMSP there are Plugins and examples ready here: www.theremino.com/downloads/foundations

More information about communications in this page: www.theremino.com/technical/communications

The HAL colors



The color scheme helps to recognize the components and their configuration

The first master (with name Theremin) provides:

A virtual slave called "Master pins"

Six "Pins" of which only the first is not used and the other configured as Cap_16

A slave-type "CapoSensor"

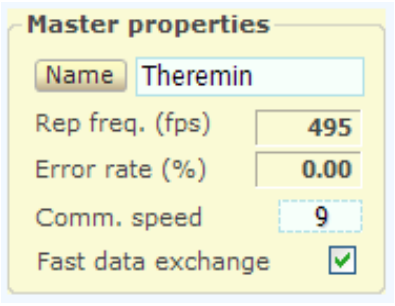
A single pin configured as "Cap sensor" and "Selected"

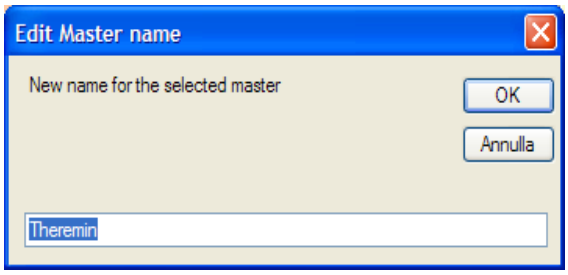
The second master (with name CapTouch_2) provides:

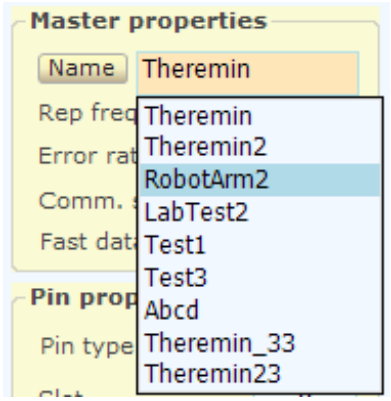
A virtual slave called "Master pins"

Six "Pins" that have been configured as Cap_16

The Master properties - The name







The selected Master can be changed with two methods:

- Pressing the "Name" button and editing the name.
- Clicking on the name and selecting a different configuration.

The name of the Master is written to the hardware module and is used to recognize it when connected.

A new Master just connected is called "No name" and it is good to take the habit of giving him, immediately, a name different from all the others.

In the names of the master the letter case (uppercase or lowercase) does not count.

If there are two master master with the same name in the database then the first is used.

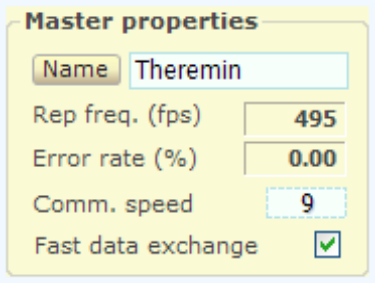
If you change the USB port then the order of the listed Masters changes but the configuration of each Master remains unchanged. It is therefore important to give different names to each Master (unless you want to have spare Master with the same name as the main one)

The HAL program can almost always use the right configuration when you unplug, replace and restore components but changing the master names using a different computer or with another application (HAL situated a separate folder - so with separate parameters) or other difficult and complicated cases, then the alignment between hardware configuration is losed.

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 file to all or only part of configurations ranging from one application to another HAL, on another computer or to another folder.

When the configuration is invalid to change the name of the master does not modify the configuration file, but only the name written in hardware, so it is possible to change the names of the master to match them to 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 Baud)
- Selection of the type of communication "Single" or "Fast" (Note 1)

The number of messages per second "Fps" should normally be from 480 to 500, if the serial communication to the slaves and the physical pin exceeds a certain number of bytes and the transmission speed is low then this number decreases.

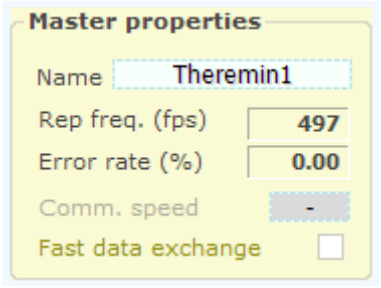
For many applications, such as capacitive keys with velocity, fps is good to maintain as high as possible, at least 400 or 450.

To increase the number of "fps":

- 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 increases the speed)
- Divide up the serial lines and connect the pins on the critical lines less loaded
- Divide up the serial lines and connect the pins on one or more critical master serial no.
- Reduce the number of bytes used by configuring as "Unused" all possible pin
- Decrease the number of bytes used by configuring the 8-bit all pins that do not require great resolution

If the Master has not selected slaves connected on the serial line, then the value "Rep freq. (fps)" refers only communication via USB which is always at the highest frequency possible.

If there is no serial communication repetition rate is independent of the properties "Comm speed" and "Fast data exchange", which are then disabled.



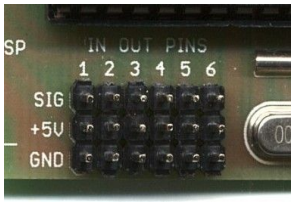
The slaves and the Pin

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			

The slaves, Here indicated by the red arrows, have no regulations, are only **Pin containers.** The slaves have usually from 1 to 12 pin

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

The master module has a built-in slave (slave called virtual) that provides six Pin.



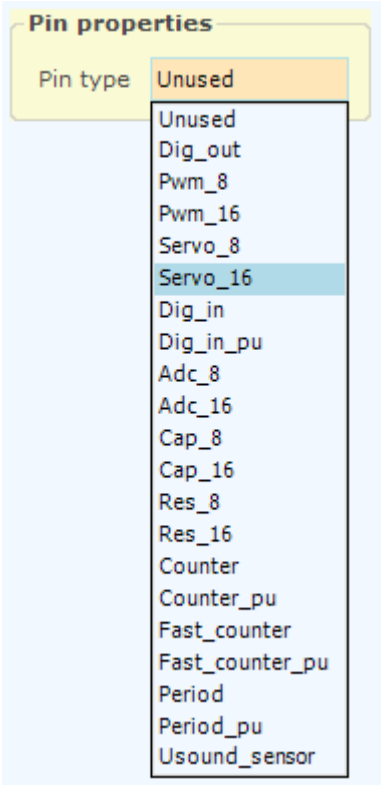
The "Slave" modules of type "Servo" have 10 Pin.

The "Slave" modules of type "Cap sensor" has a single, high quality, "Pin" (can feel the movement of large objects up to several meters away)

The Pin types

Pins can be configured as:

- Not used
- Digital Output
- PWM output
- Output for servos
- Digital input
- ADC input for potentiometers and transducers
- Input for capacitive buttons
- Input for resistive transducers
- Counter input, frequency and period
- Input for special transducers
- Input modules CapSensor



The special pins:

- The 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 configurable as "Fast counter"
- The pin 9 of the slaves "Servo" is the only configurable as a "Period" and as "Usound sensor"
- The single "CapSensor" Pin can be configured only as "Unused" or "Cap sensor"

The best pin to be used as ADC and ZIP:

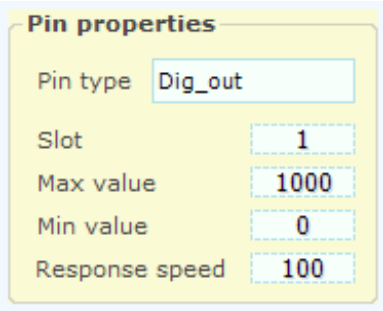
- The best pin to be used as ADC and CAP are pins 3,4,5,6
- Pins 7 and 8 have leakage current and capacity doubles (second choice for ADC and CAP)
- Pins 1 and 2 leakage current and capacity four times greater (third choice for ADC and CAP)

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 exchanges per second.

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

The types with pullup, whose name ends in "_pu" allow you to easily connect switches, buttons, and open-collector devices, without having to add external resistors.

The parameters common to all pins



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

Warning: Many applications and many pin can be read from the same slot,
but you must not configure more than one pin writing to the same slot,
doing so does not break anything, but the results are undefined.

"Max value" normally held to 1000, indicates the value that the pin must have when it is at maximum.

"Min value" usually held to zero and the value that indicates the Pin must have when it's at 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 sensors turn the reading of that act to the contrary.

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

Further information about Slots, Pins and Modules

More information about Pins *on this page:* **www.theremino.com/technical/pin-types**

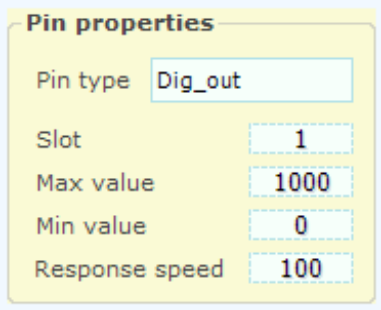
Features *of the individual modules:* **www.theremino.com/hardware/devices**

Datasheets *of the modules:* **www.theremino.com/technical/schematics**

Blog and Tips *about the use modules usage:* **www.theremino.com/blog/master-and-slave**

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

DIG_OUT

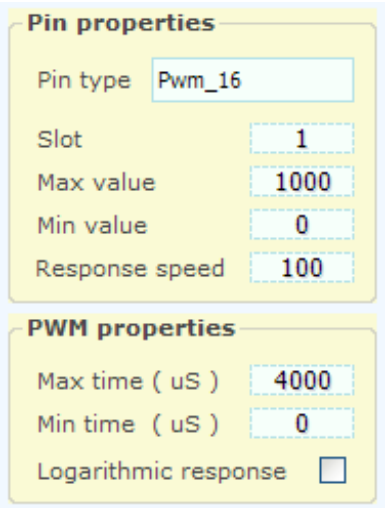


This type of pin provides a digital output.

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 it exceeds the Pin lights otherwise turns off.

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

Pwm_8 and Pwm_16



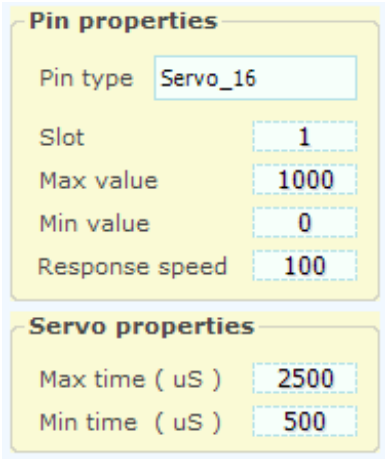
This type of pin provides a PWM (pulse width modulation) output.

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

The pin is pulsing between the voltages 0 V (off) and 3.3 volts (on) and the output current is limited to approximately + / - 10 mA

Servo_8 and Servo_16



This type of pin directly drives the servo commands.

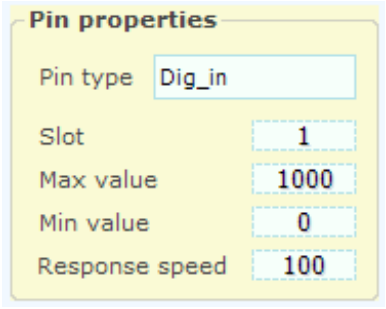
The value coming from a slot, limited between "Min value" and "Max value" and filtered by "Response speed" is converted to pulse of width between "Min time (uS)" and "Max time (uS)"

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

The pin provides voltages of 0 and 3.3 volts, suitable for all normal servo powered from 3 to 6 volts and a current sufficient to drive tens of paralleled servo.

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

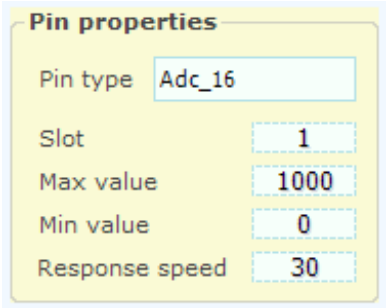
Dig_in and Dig_in_pu



This type of pin provides a digital input.

The voltage value is read with a SchmittTrigger with low threshold = 1 Volt and high threshold = 2 Volt, and transformed into a On/Off information that ultimately become "Max value" and "Min value". The value is then filtered with "Response speed" and finally written into the slot. The filtering can produce intermediate values roughly proportional to the ratio of time between On and Off

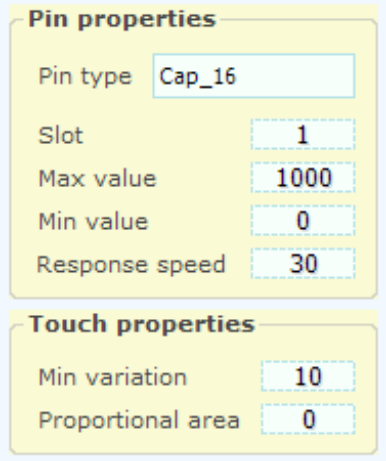
Adc_8 and Adc_16



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 then 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



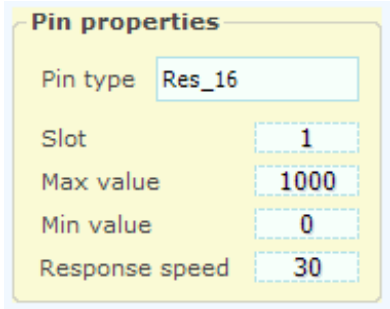
This type of pin allows to read simple keys as with a Makey Makey (*http://vimeo.com/60307041 #*) but with superior performance.

In addition to the ON-OFF indication of a Makey Makey you can also get a gradual control, as with the sliders "slider", or the control of "expression" determined by the speed of key presses.

The keys are not resistive but capacitive and therefore can be adjusted to work with just by touching them, without contact, through an insulator and without additional wire connection of the earth.

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

Res_8 and Res_16

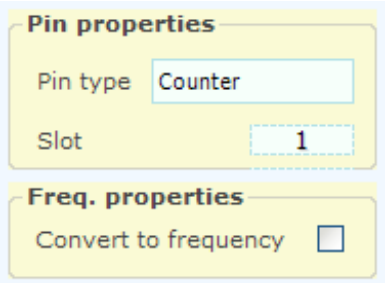


This type of pin allows to read a resistance value between zero and 50 Kohm. Very useful for reading potentiometers using only two wires. Not using power will eliminate the interference caused by the 5 volts of USB without adding a regulator and without connecting to the 3.3 Volt regulated already available on the pins of the special master.

The experiments of finding the acupuncture points and the classics jars meter Scientology gave interesting results.

The "Input" Pin types - Counter

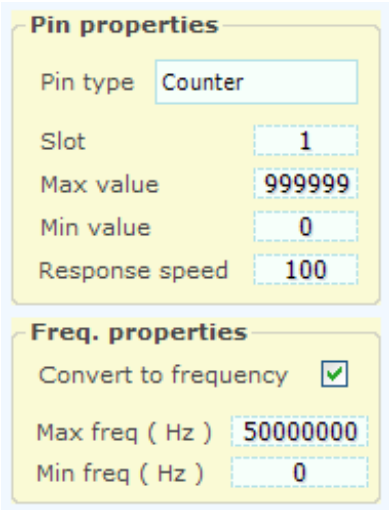
Counter and Counter_pu



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 have to use a FastCounter.

Counter and Counter_pu with the "Freq"



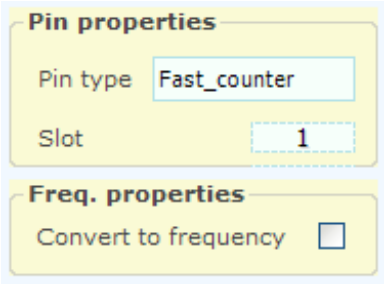
The pins programmed as Counter or Counter_pu can be transformed from counters to frequency-meters.

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

The "Counter" and "Counter_Pu" pin-types use 16 bits for the data transmission.

The "Input" Pin types - Fast_counter

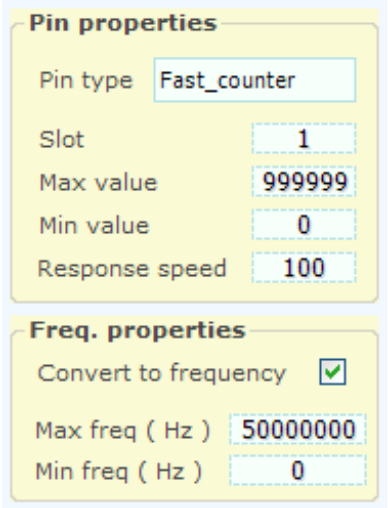
Fast_counter and Fast_counter_pu



Only some 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 the duty-cycle must be 50%)

Fast_counter and Fast_counter_pu with the "Freq"



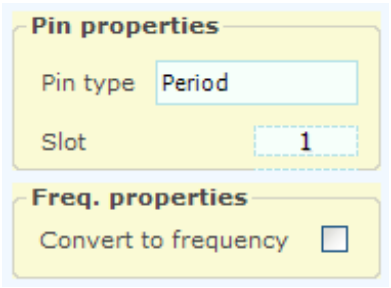
The pins programmed as Fast_counter or Fast_counter_pu can be transformed from counters to frequency-meters.

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

The "Fast_counter" and "Fast_counter_pu" pin-types use 16 bits for the data transmission.

The "Input" Pin types - Period

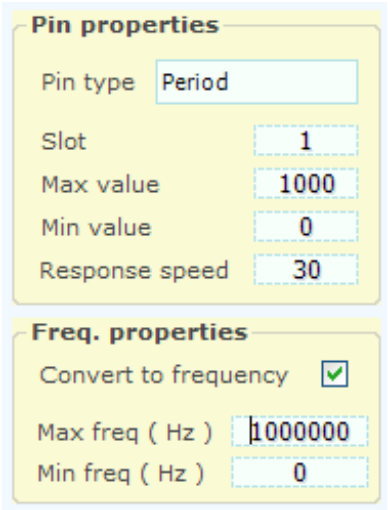
Period and Period_pu



This type of Pin measure the period of a repetitive waveform, from rising edge to rising edge, up to a maximum period of about 260 seconds.

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

Period and Period_pu with the "Freq"



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

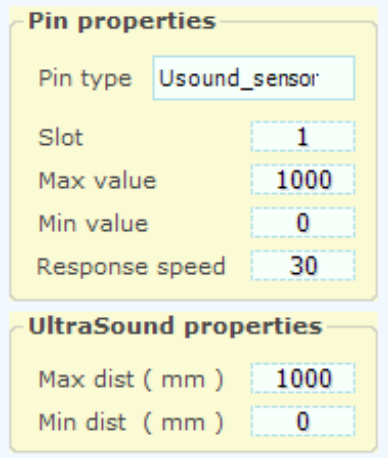
This technique allows to measure very low frequencies (up to about a tenth of a Hertz) with very high resolution.

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

The "Period" and "Period_pu" pin-types use 32-bit for the data transmission.

The "Input" Pin types - Usound and CapSensor

Usound_sensor



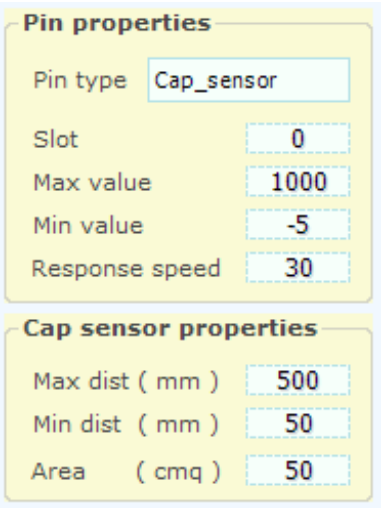
Many ultrasonic distance sensors such as the model SRF05, can be read with this type of Pin.

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

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

The "Usound_sensor" pin-type use 16 bits for the data transmission.

CapSensor_HQ



This Pin type 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 then compared between "Min value" and "Max value" filtered "Response speed" and finally sent to the slot.

The "CapSensor_HQ" pin-type use 24 bits for data transmission.

Warning: the CapSensor values "Min dist" and "Max dist" are only approximate. The exact range is not important, this is not a measuring device. Maybe future improvements of the linearization formula could improve the precision mainly at the little distances.

Resistive or capacitive buttons

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

But 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, need an additional wire as a ground reference and do not work through insulating materials such as plastic.

Also the buttons on the Makey Makey are only six (not expandable), each Makey Makey could provide up to 20 keys and you could also connect more Makey Makey, but all the keys are sent to the keyboard that bears a maximum of six: www.makeymakey.com/faq

Finally, the Makey Makey have an on/off operation, without intermediate adjustments and do not feel the speed of key presses.

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

Moreover system buttons Theremino can also provide a gradual control such as with a cursor slider type, and also the control of "expression", determined by 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 (adjust for maximum output)
  "Proportional area" -30 (adjust to a maximum of about 1000)

The "Min Variation" and "Proportional Area"

Min variation eliminates small variations and prevents electrical noise can trigger keys without touching them.

Raising this parameter, keys become less sensitive so you should keep it as low as possible, consistent with the need to eliminate all the 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 facilitate this adjustment temporarily set "Proportional area" with a negative number large enough, for example, -50

Proportional Area is adjusted to have about a thousand when the finger is in the position of the slider up or when you press the buttons as fast as possible.

Normally this value should be higher for Pin 1 and 2 (less sensitive) and in case of long wires and large objects.

Zero calibration of the capacitive buttons

If you change the mechanical arrangement of the keys, or their position, or moving the wires that connect them are approaching or metal objects with the program in HAL function, 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 take away capacity of the keys (shorten the wires or ground them away from metal objects) calibration is automatically rebuilt immediately but it is not possible to distinguish an increase in capacity due to a finger or a shift of the wires.

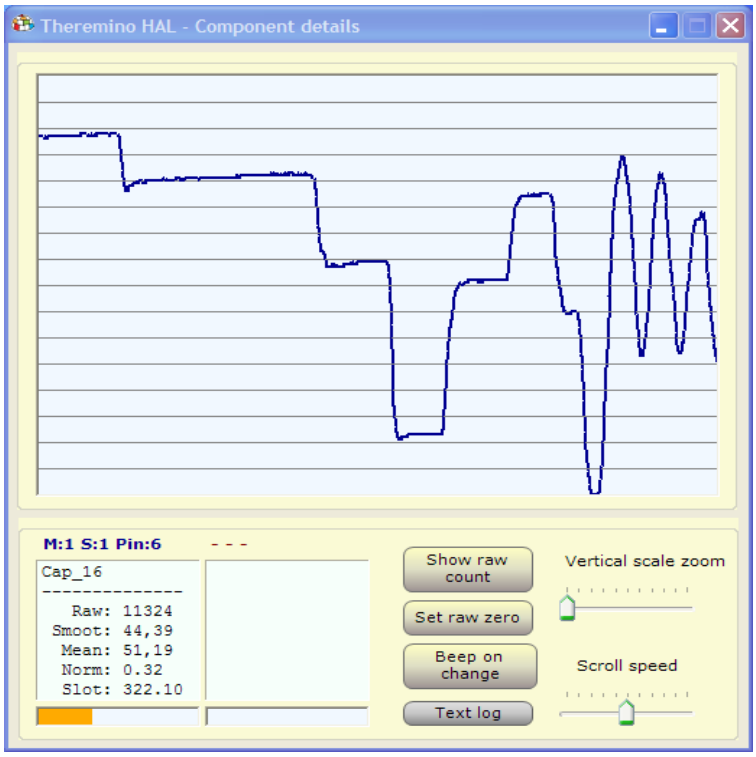
We have tried many methods of automatic recognition methods with slow drift and calibration methods with timed, but none worked well and all the less accurate the normal key operation.

So you should not 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 recede hands from the button and occurs in the details of its pin that the values "Smoot" and "Mean" are equal to each other or very close (not more than one point of difference)

When in doubt, do not usually please make checks but you press Recognize (keep your hands away from the keys while performing the zero calibration)

The capacitive keys of type "Slider"



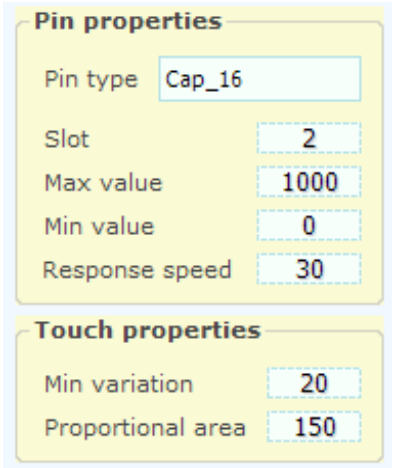
"Proportional area" must be a positive number, this determines the "Proportional" operation.

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 in high = 1000, all in low = 0

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

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



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

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

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

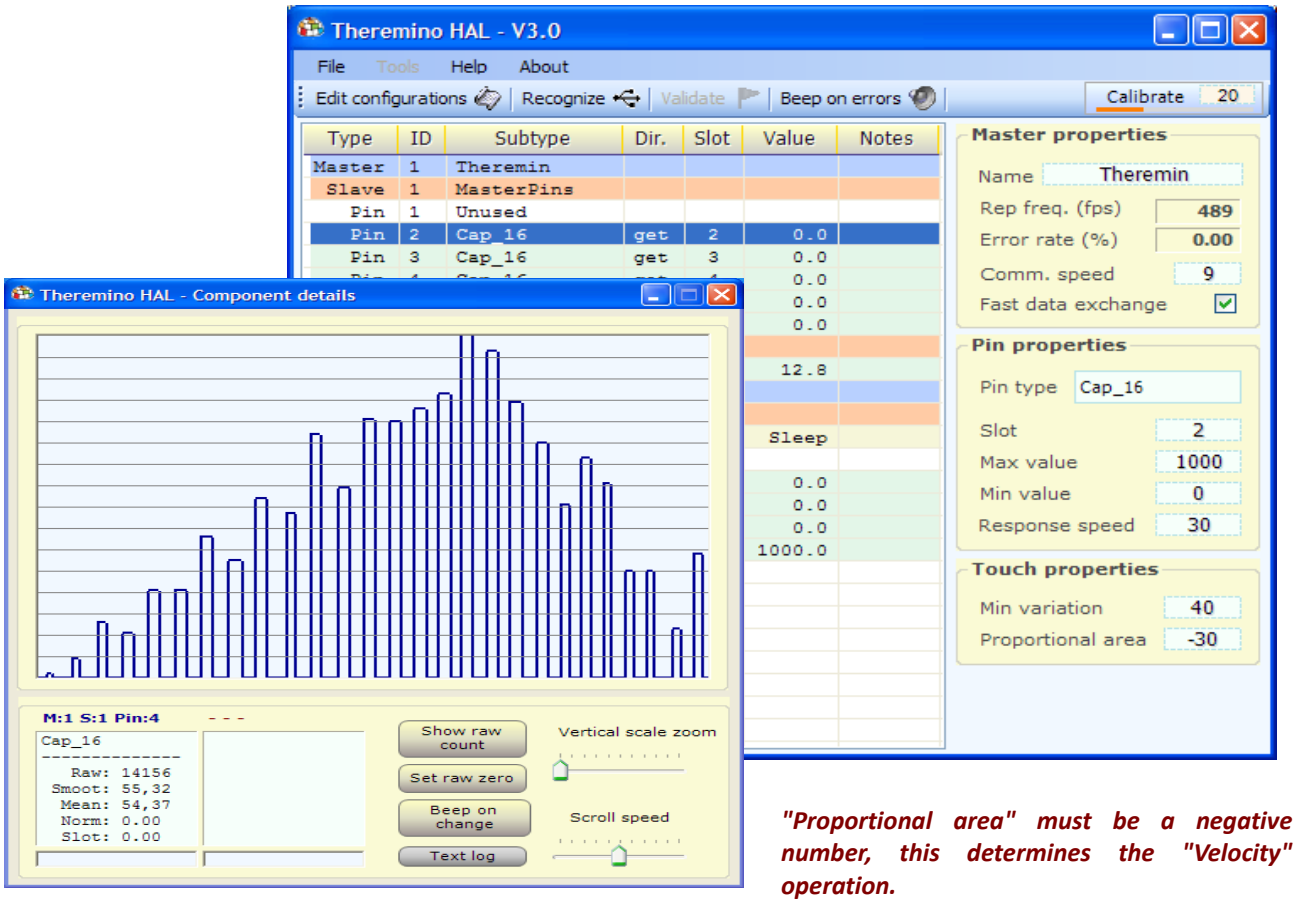
"Min variation" is set normally from 10 to 100 (better to raise it slightly to obtain the maximum sensitivity in the lower part)

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

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

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

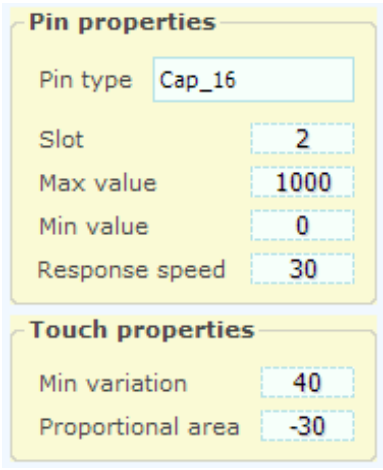
The capacitive keys with "Velocity"



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

The keyboards allowing you to play the notes loud or soft, depending on how you press the keys, are very popular for musical applications. The 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 must adjust the keys, one by one, so as to obtain a maximum value of just over 1000



These are the settings for the keys with "Velocity"

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

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

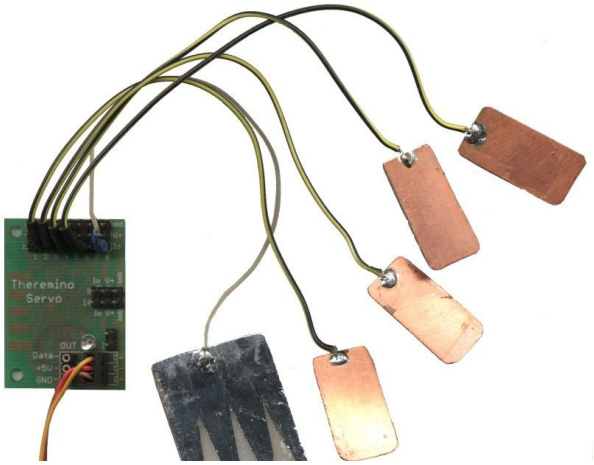
"Response speed" normally held to 30 (best not to edit)

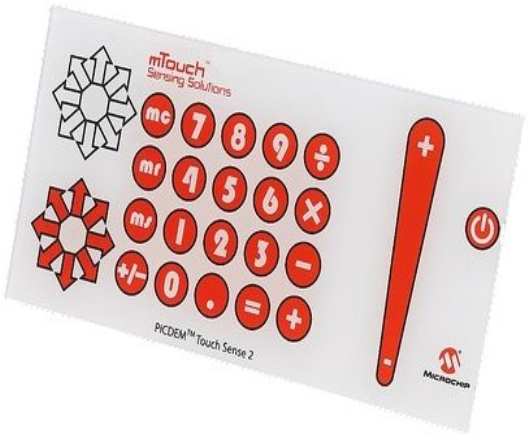
"Min variation" is normally set to 50 (and about 25 for the keys 1 and 2, which are less sensitive or for keys with long wires)

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

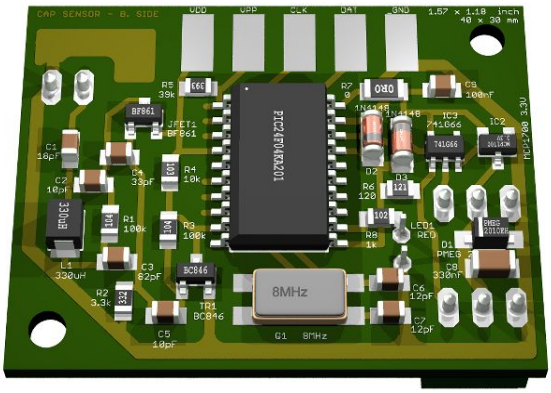
***(Note 1)** To reverse the output signal can be exchanged with Min Max (Min = 0 and Max = 1000)*

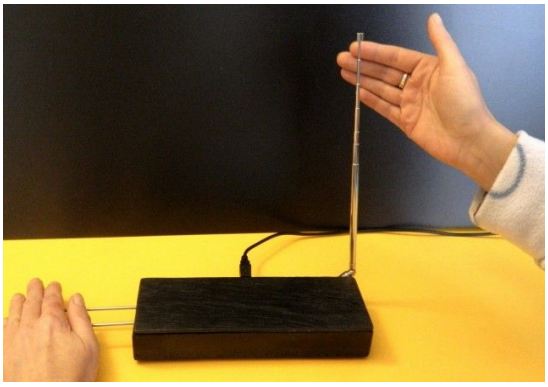
Differences between CapacitiveKeys and CapSensors





The capacitive keys do not replace the 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 can be adjusted for an answer almost perfectly linear. The capacitive keys on the other hand are much cheaper and are better suited to compose keyboards with many keys.





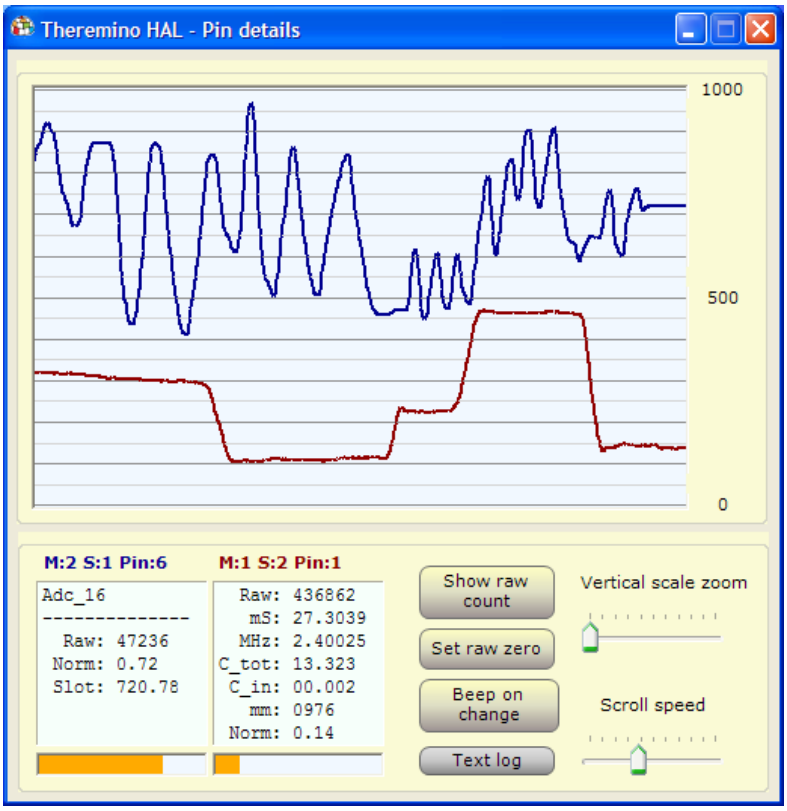
Mechanical construction of the capacitive keys

Make sure that the capacitive keys must be isolated, otherwise it may suffice a small spark of static electricity to produce communication errors, it does not break anything but communication is interrupted and you have to press the button "Recognize"

So normally turns them with copper under and vetronite (thin) above or, even better, is held above and the copper is added to a thin sheet of plastic insulating printed in color with the shape of the keys, as in the image at the beginning of this page.

The wires that go from the keys to the Pin must be as short as possible and be at least 5 or 10 millimeters between them. The key operation and the capture of the disturbances improve by decreasing the capacitance. Experiments were conducted in situations "impossible", with long wires and all kinds of touch, like plants potted flowers and various fruits and individual adjustment has always run all keys.

The display of the details of Pin



Clicking on a pin or on two consecutive pins opens this useful indicator to display the signal of a pin or two pins at the same time.

The normal range is from 0 to 1000 but can be changed with the control "zoom" that magnifies the central area.

In some cases it may be useful to see the raw values.

For "Raw" values is to use the "Show raw count", "Set Raw zero" and possibly the "Zoom"

The "Scroll speed" slider adjusts the speed of the graph.

The two text boxes show the internal details of the pins, the title indicates which pin is, in this image the text "M:2 S:1 Pin:6" means "Master 2, Slave 1, Pin 6"

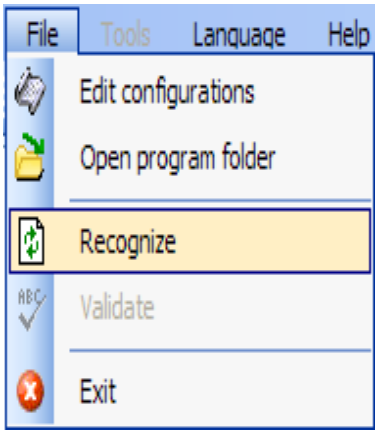
Details of Pin may help in the control and regulation of Input Output devices (sensors and actuators)

Some types of Pin are more complex and have more intermediate values. In general there exists a "Raw" value 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 is the "Simplified" value available on Slots and easily usable by all the high-level software.

- **Raw** Value "crude" which can be a count, 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 coil 330 uH (used only in CapSensorHQ)
- **C_in** The input power capacity addition after calibration (only CapSensorHQ)
- **mm** Approximate distance in millimeters (only CapSensorHQ and ultrasonic sensors)
- **Smoot** Value that was passed in an FIR filter smoothing (used only in Cap8 and CAP16)
- **Mean** Average value (used in type Cap8 and CAP16 as 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 may be worth only "0" or "1" (only used by DigOut)

Although not indicated capacities are always in pF (pF)

Menu commands



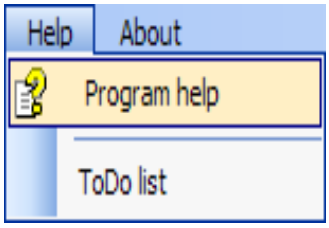
You can edit the settings manually, read the "Frequently Asked Questions" on the last page of this document.

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

"Recognize" is used to recognize the Master and Slaves connected.

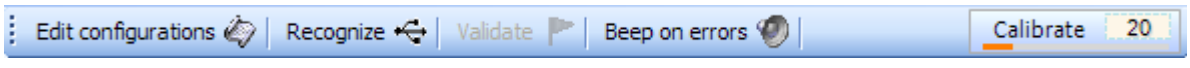
Validate is used to "make good" a configuration if you add or subtract Slaves.

These are used to communicate with the Martians.



These commands are to access the documentation.

The toolbar keys

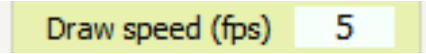


For the meaning of "Edit configuration", "Recognize" and "Validate" read at the beginning of this page.

"Beep on errors" serves to highlight each communication error with a beep sound.

"Calibrate" is only meant for CapSensor slaves. Remove your hands from the sensitive part before pressing it. If there are no movements greater than the number displayed in the small box, then every 30 seconds is an automatic calibration. It is possible to disable the automatic calibration with the value "0"

Adjusting the numerical boxes



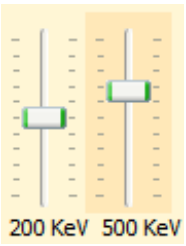
The HAL numerical boxes (and all other Theremino system applications) have been developed by us (note 1) to be more comfortable and flexible than the original Microsoft TextBox.

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 up-arrow and down-arrow keys
- With conventional methods that are used to write numbers with the keyboard
- With the usual selection and copy-paste methods

Moving the mouse up and down allows wide and fast adjustments
The mouse wheel allows a comfortable and immediate setting
The arrow keys allow 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 name a source. The source for "Open" also serves as a guarantee that we have not included malware.*

Adjust the sliders



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

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

The sliders can be adjusted in the following ways
- Clicking the cursor with the right mouse button to "clear 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 adjustment at a glance.
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, but it can be more intuitive to use for the first and second horizontal cursors for vertical sliders.

Questions and Answers

Can I change the text of the panels of the program in different languages?

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

For languages German, French and Spanish 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 modules and configurations Master and Slaves is kept aligned by ThereminoHAL, which uses the names of the master to determine the right configuration to be taken. Normally, the 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 that is on a different computer, or in a different folder, then the alignment between hardware and configuration is losed. In these cases you can click on the drop down list of the master name and restore the alignment by choosing the right configuration for each master.

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

How to reduce the CPU work?

- Reduce "Scroll speed"
- Close or minimize the window "Component details"
- Minimize the main window