

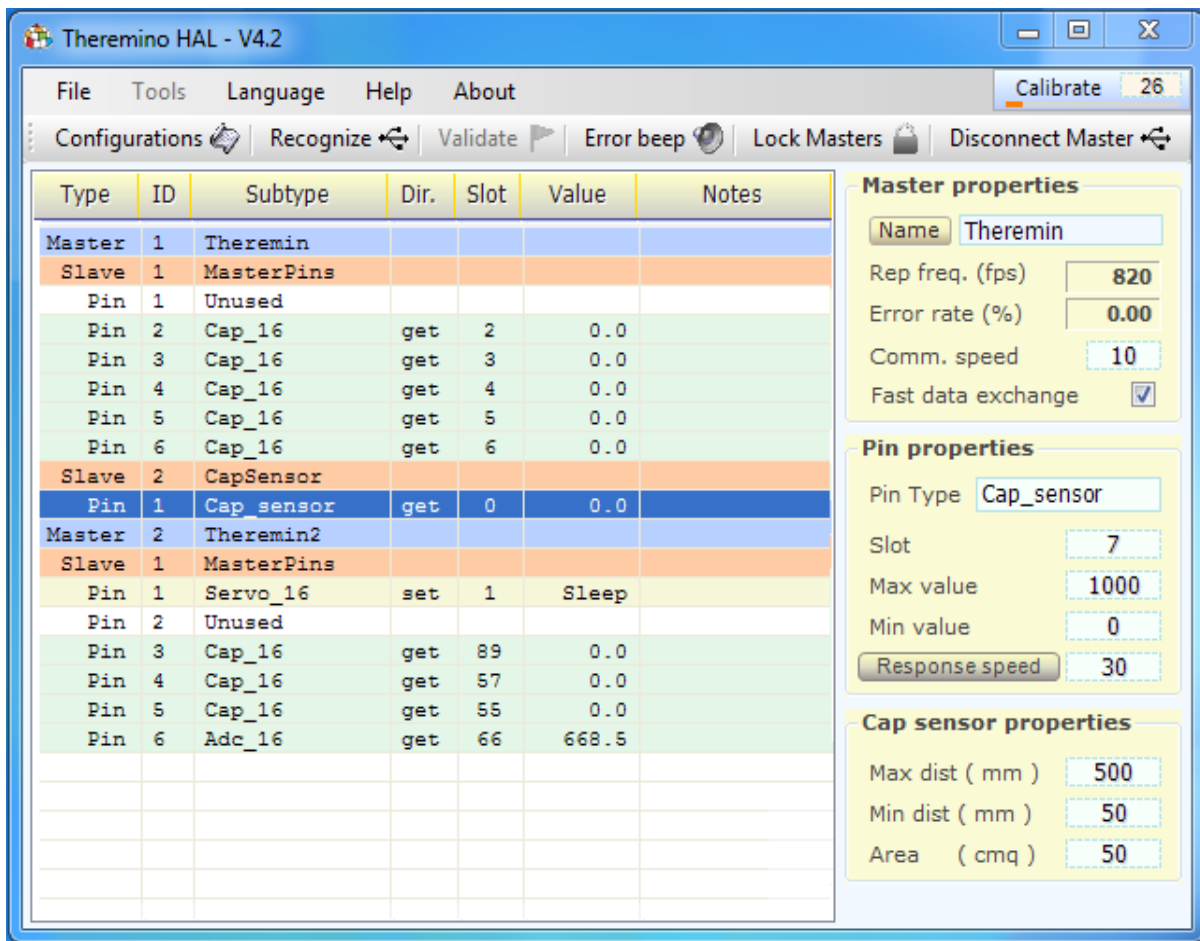
theremino
•the•real•modular•in-out•

System Theremino

Theremino HAL - V4.2

Instructions

Theremino HAL



Theremino HAL with two Masters connected

ThereminoHAL (Hardware Abstraction Layer) is a Hardware Manager. Such a *pretentious name*, for just a *small application*! HAL really deserves it as, in spite of its apparent simplicity, carries out complex operations with highly optimized algorithms.

ThereminoHAL is the heart of the communication with 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 InputOutput 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 type of InOut, like moving a motor or even just reading a key, the appropriate firmware, should be written (same as in Arduino).

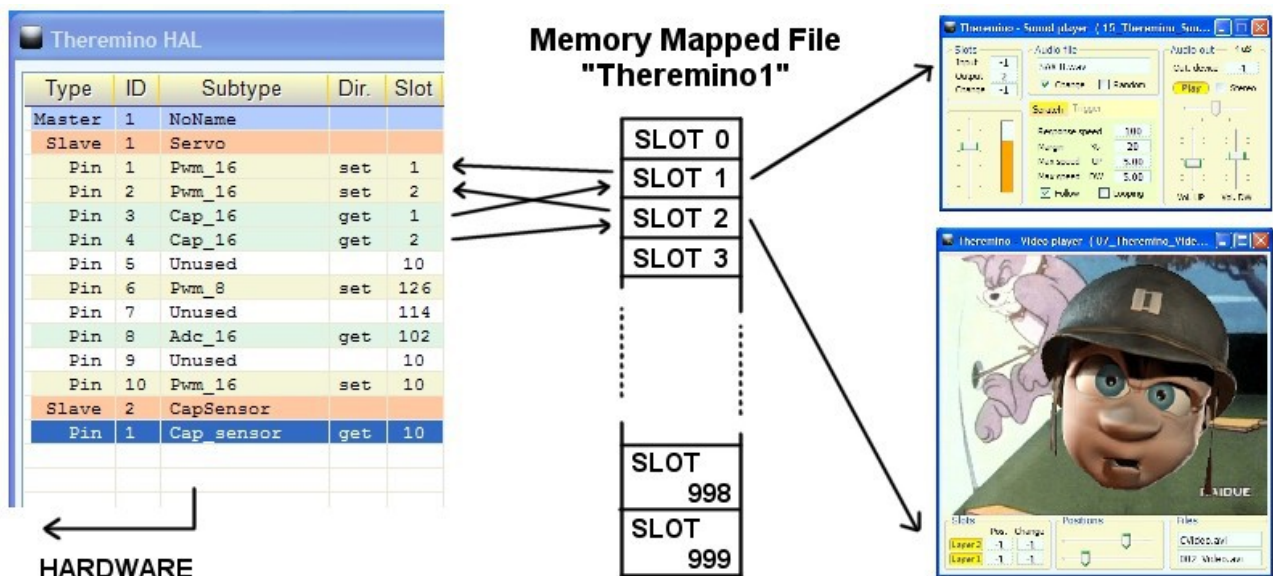
If you use hardware modules, then HAL is essential and must remain active. You can minimize it, but it should remain running.

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.

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.

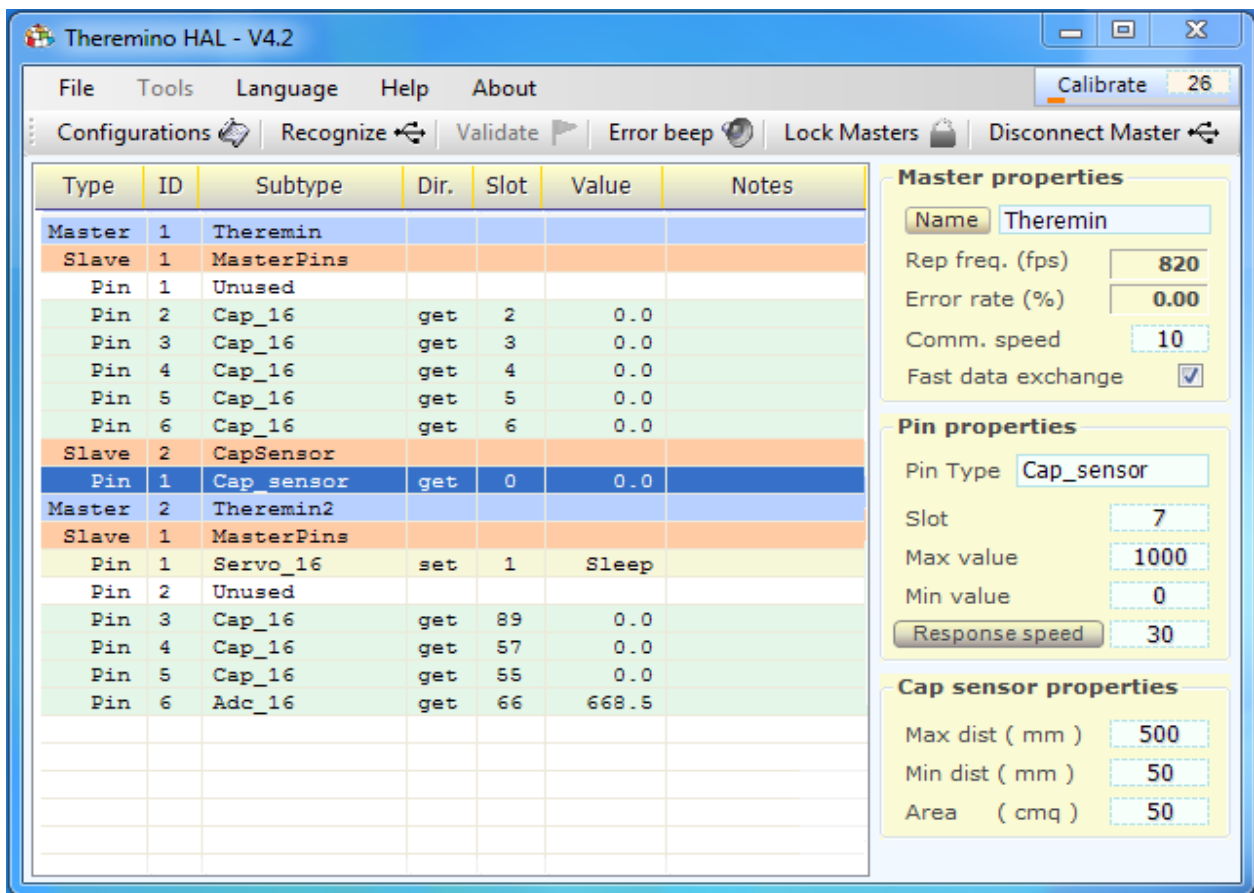
Many applications and pins can read from the same slot. It should then avoid, configuring more than a Pin in writing mode, on the same slot; doing so nothing is broken, but the results are unpredictable.

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.

To establish mathematics and logic rules between the Slots and to write complex behavioral algorithms, as well, we use "Theremino_Script" or 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/downloads/foundations

For more information about communication, please check 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 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 CapTouch_2) provides:

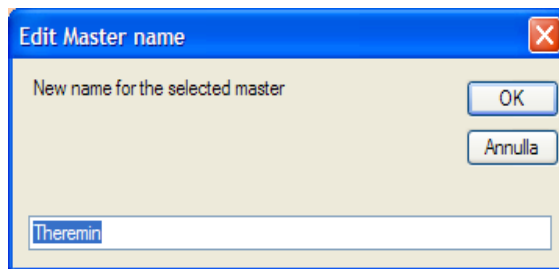
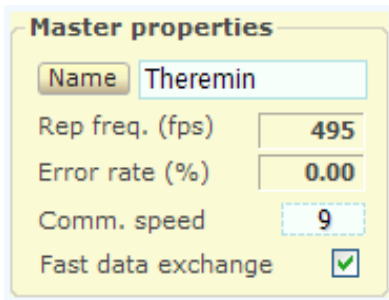
A virtual slave called "Master pins"

A "Pin" configured as "Servo_16"

A "Pin" configured as "Unused"

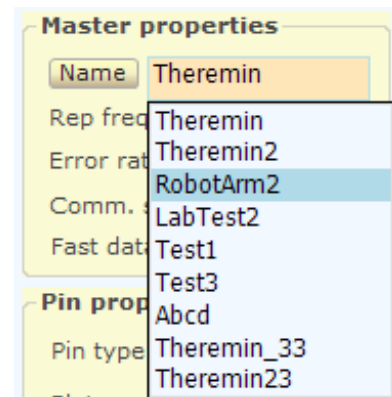
Four "Pins" configured as Cap_16

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 "No name". 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, just the first one is used.

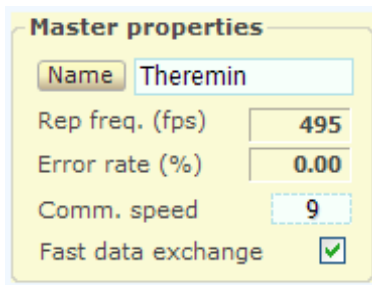
If the USB port is changed, the order of the Masters in the list changes, but the configuration of each Master remains unchanged. It is therefore important to give different names to each Master (unless you want to have spare Masters, with the same name)

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



The screenshot shows a dialog box titled 'Master properties' with a yellow background. It contains the following fields and controls:

Field	Value
Name	Theremin
Rep freq. (fps)	495
Error rate (%)	0.00
Comm. speed	9
Fast data exchange	<input checked="" type="checkbox"/>

- 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.

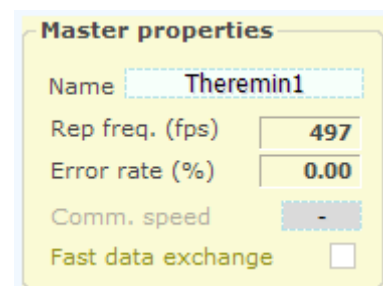
For many applications, such as capacitive keys with velocity, it is good to maintain the fps, as high as possible, at least at 400 or 450.

To increase the "fps":

- 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

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

With no serial communication, repetition frequency is independent from the properties "Comm speed" and "Fast data exchange", which are then disabled.



The screenshot shows a dialog box titled 'Master properties' with a yellow background. It contains the following fields and controls:

Field	Value
Name	Theremin1
Rep freq. (fps)	497
Error rate (%)	0.00
Comm. speed	-
Fast data exchange	<input type="checkbox"/>

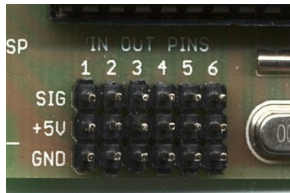
Slaves and Pins

Type	ID	Subtype	Dir.	Slot	Val
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			
Pin	1	Cap_16	get	1	

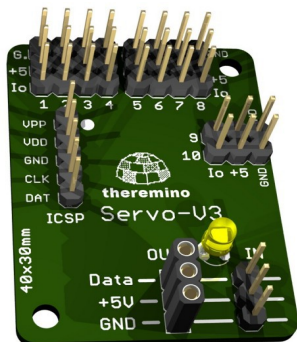
Slaves, here indicated by red arrows, have no regulations, being only **Pin containers**. The slaves usually have between 1 to 12 pins.

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

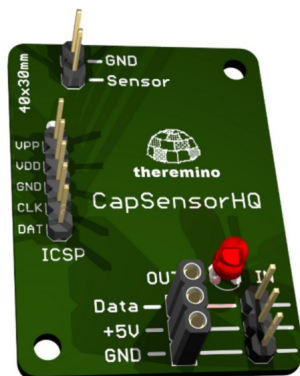
The master module has a built-in slave (called virtual slave) that provides six Pins.



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



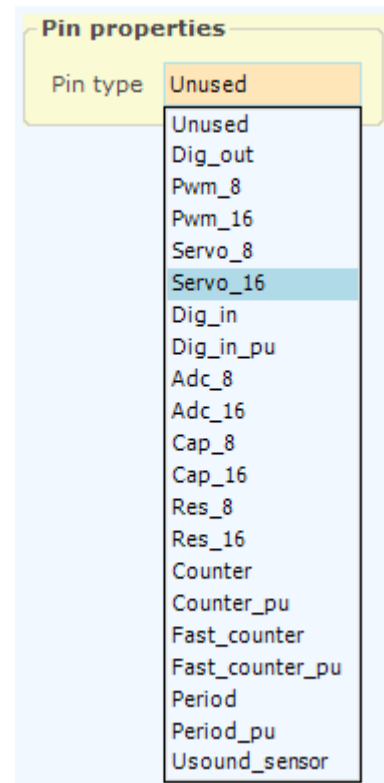
"Cap sensor" type "Slave" modules, have a single, high quality "Pin" (capable of detecting movements of large objects, up to several meters away)



Pin types

Pins can be configured as:

- ◆ Not used
- ◆ Digital Output
- ◆ PWM Output
- ◆ Servos Output
- ◆ Digital input
- ◆ Potentiometers and transducers ADC input
- ◆ Capacitive buttons Input
- ◆ Resistive transducers Input
- ◆ Frequency and Period, Counter input
- ◆ Input for special transducers
- ◆ CapSensor modules Input



Special 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 ZIP:

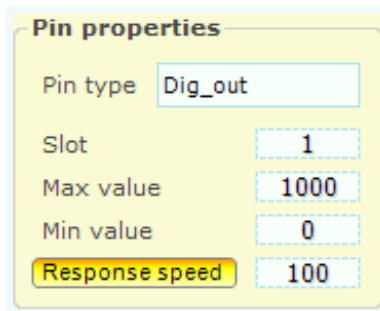
- ◆ The best pins to be used as ADC and CAP are pins 3,4,5,6
- ◆ Pins 7 and 8 have leakage current and double capacity (second choice for ADC and CAP)
- ◆ Pins 1 and 2 have leakage current and a capacity four times greater (third choice for ADC and CAP)

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.

The parameters common to all pins



The image shows a 'Pin properties' dialog box with the following fields and values:

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

"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, avoid to configure more than one pin, writing to the same slot. Doing so doesn't damage anything, but the results are not valid.

"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 on the contrary.

"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.

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 modules usage: www.theremino.com/blog/master-and-slave

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

◆ DIG_OUT

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

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

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

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

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 3.3Volts (on), the output current is limited to approximately +/-10mA

◆ Servo_8 and Servo_16

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

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

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

◆ Dig_in and Dig_in_pu

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

This type of pin provides a digital input.

The voltage value is read with a Schmitt Trigger set to low threshold = 1 Volt and high threshold = 2Volts, and transformed into a On/Off information, afterward becoming "Max value" and "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

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

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

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

Touch properties	
Min variation	10
Proportional area	0

This type of pin allows to read from improvised keys, same as Makey Makey (<http://vimeo.com/60307041#>) but with superior performance. (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

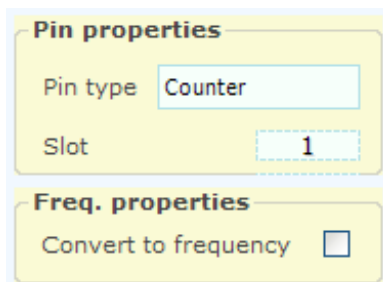
Pin properties	
Pin type	Res_16
Slot	1
Max value	1000
Min value	0
Response speed	30

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

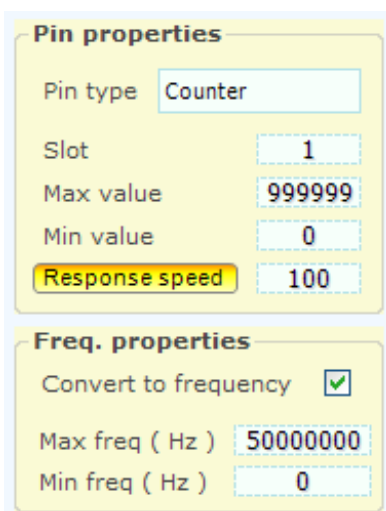


The screenshot shows a dialog box with two sections. The top section, titled "Pin properties", contains a "Pin type" dropdown menu set to "Counter" and a "Slot" input field with the value "1". The bottom section, titled "Freq. properties", contains a "Convert to frequency" checkbox which is currently unchecked.

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"



The screenshot shows a dialog box with two sections. The top section, titled "Pin properties", contains a "Pin type" dropdown menu set to "Counter", a "Slot" input field with the value "1", a "Max value" input field with the value "999999", a "Min value" input field with the value "0", and a "Response speed" input field with the value "100". The bottom section, titled "Freq. properties", contains a "Convert to frequency" checkbox which is checked, a "Max freq (Hz)" input field with the value "50000000", and a "Min freq (Hz)" input field with the value "0".

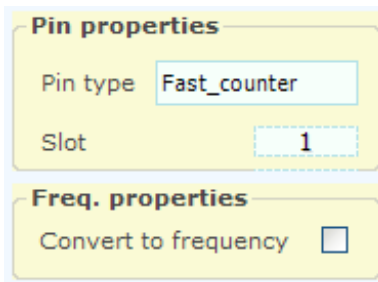
Pins programmed as Counter or Counter_pu, can be transformed from counters to frequency-meters.

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.

"Input" Pin types - Fast_counter

◆ Fast_counter and Fast_counter_pu

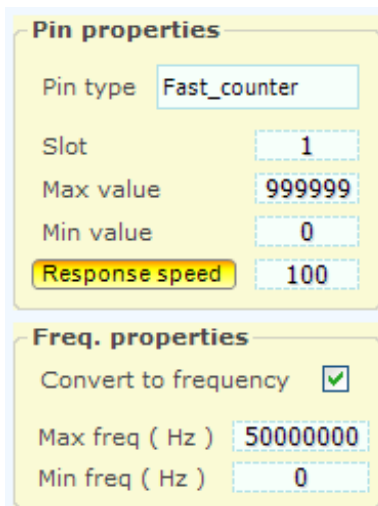


The image shows two configuration panels. The top panel, titled 'Pin properties', has a 'Pin type' dropdown set to 'Fast_counter' and a 'Slot' input field with the value '1'. The bottom panel, titled 'Freq. properties', has a 'Convert to frequency' checkbox which is currently unchecked.

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 image shows two configuration panels. The top panel, titled 'Pin properties', has a 'Pin type' dropdown set to 'Fast_counter', a 'Slot' input field with the value '1', a 'Max value' input field with the value '999999', a 'Min value' input field with the value '0', and a 'Response speed' input field with the value '100'. The bottom panel, titled 'Freq. properties', has a 'Convert to frequency' checkbox which is checked, a 'Max freq (Hz)' input field with the value '50000000', and a 'Min freq (Hz)' input field with the value '0'.

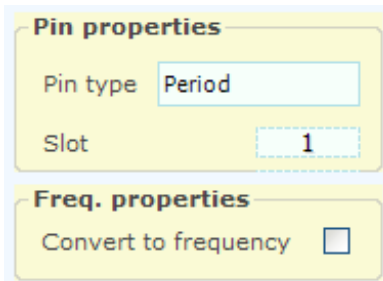
The pins programmed as Fast_counter or Fast_counter_pu, can be transformed from counters to frequency-meters.

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

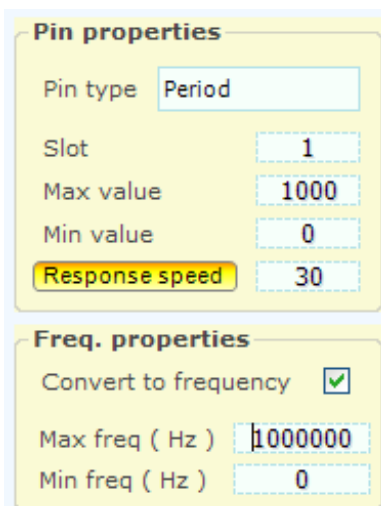


The screenshot shows a dialog box titled "Pin properties" with two sections. The "Pin properties" section has a "Pin type" dropdown set to "Period" and a "Slot" input field set to "1". The "Freq. properties" section has a "Convert to frequency" checkbox which is unchecked.

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



The screenshot shows a dialog box titled "Pin properties" with two sections. The "Pin properties" section has a "Pin type" dropdown set to "Period", a "Slot" input field set to "1", a "Max value" input field set to "1000", a "Min value" input field set to "0", and a "Response speed" input field set to "30". The "Freq. properties" section has a "Convert to frequency" checkbox which is checked, a "Max freq (Hz)" input field set to "1000000", and a "Min freq (Hz)" input field set to "0".

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 - Usound and CapSensor

◆ Usound_sensor

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

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

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

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

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

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 #](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

Touch properties	
Min variation	20
Proportional area	0

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

Touch properties	
Min variation	20
Proportional area	150

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

Touch properties	
Min variation	40
Proportional area	-30

- **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

Touch properties	
Min variation	40
Proportional area	-30

- **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 Recognize (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 MInVariation with a negative value, the meaning of Min Variaton 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 MinVariation value itself.
- The range of usable capacities is from a few pF to a few nanofarad.

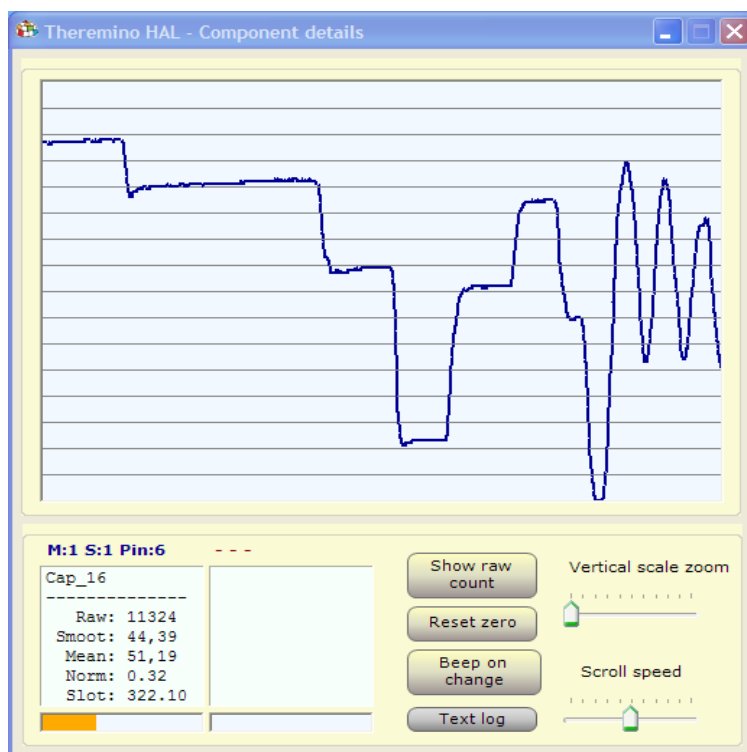
Capacitive keys of the type "Slider"



"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)

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

Touch properties	
Min variation	20
Proportional area	150

"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)

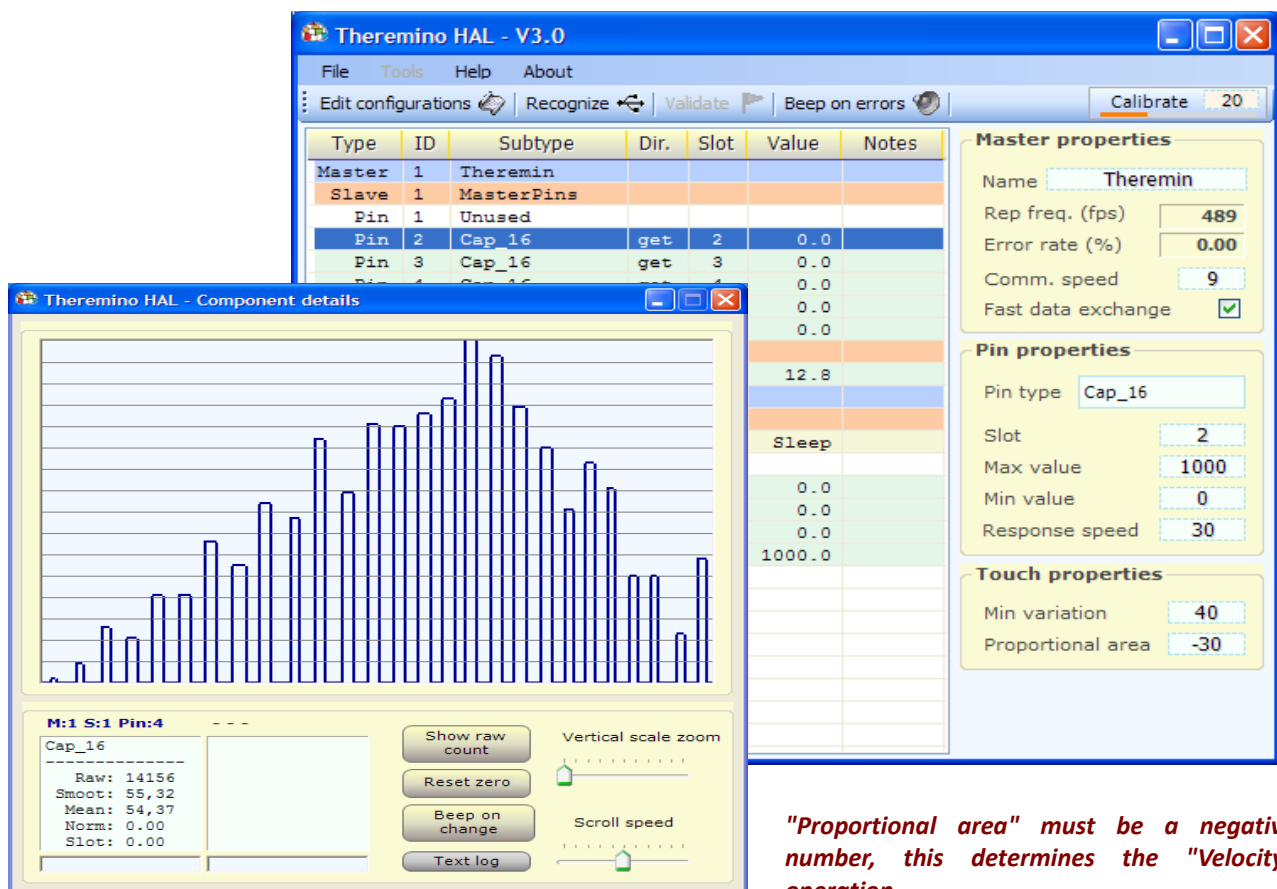
"Min variation" is set normally from 10 to 100 (better to raise it slightly to obtain the maximum sensitivity in the lower part)

"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.

Pin properties

Pin type

Slot

Max value

Min value

Response speed

Touch properties

Min variation

Proportional area

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)

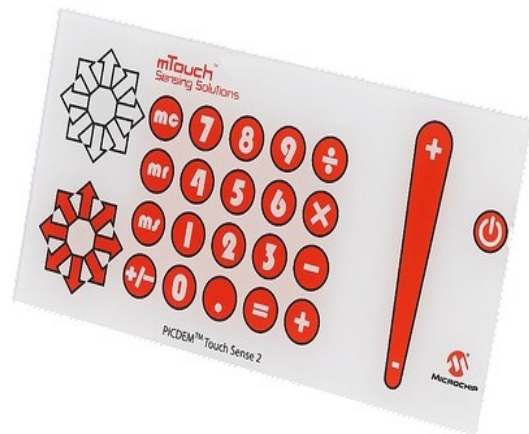
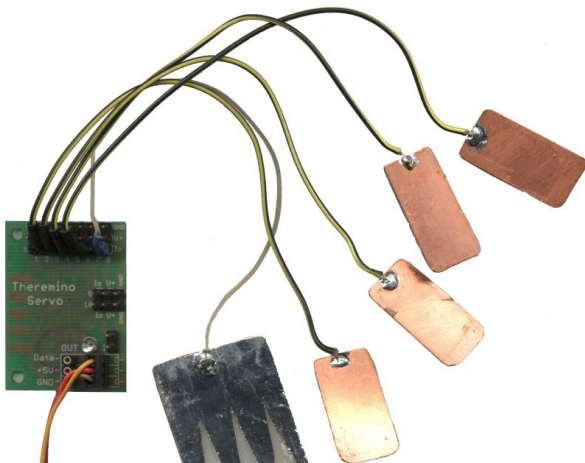
"Response speed" normally set to 30 (best not to edit)

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

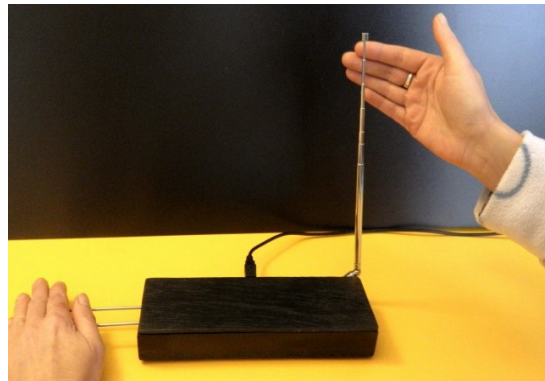
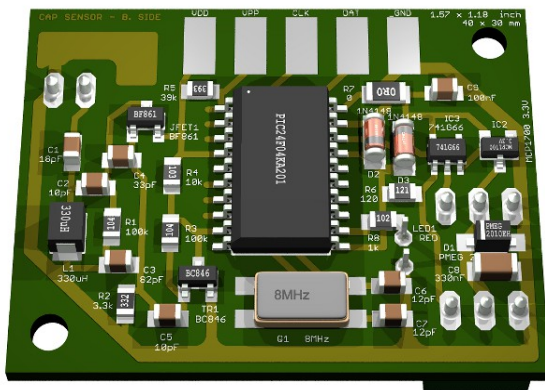
"Proportional area" is set normally to -40 (about -20 for buttons 1 and 2, which are less sensitive or keys with long wires)

(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

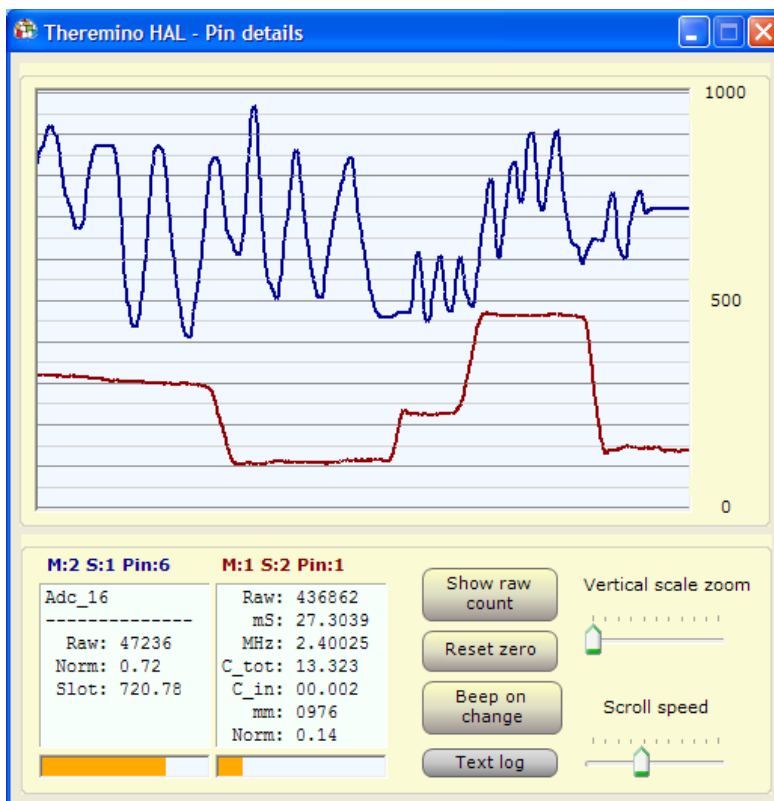


Make sure that capacitive keys are well isolated, otherwise it may be enough a small spark of static electricity, to produce communication errors. Nothing is broken, but communication is interrupted and you have to press the button "Recognize".

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



Clicking on one or two consecutive pins, this useful indicator is opened, displaying the signal of one or two pins at the same time.

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

When using the "Zoom", the button "Set Raw zero" centers the scale on the visualized data.

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

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 analyzed, in this image the text "M:2 S:1 Pin:6" means "Master 2, Slave 1, Pin 6"

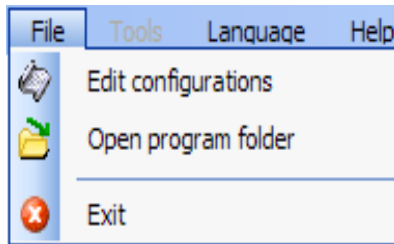
Details of the Pin, may help in the control and regulations of Input Output devices (sensors and actuators).

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 in CapSensorHQ)
- ◆ **C_in** The input power capacity added after calibration (only CapSensorHQ)
- ◆ **mm** Approximate distance in millimeters (only CapSensorHQ 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" (only used by DigOut)

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

Menu commands



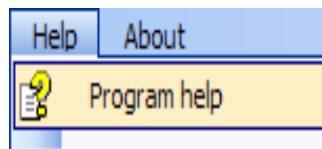
Edit configurations can be useful in some cases. Read the "Frequently Asked Questions" on the last page of this document, for more information.

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



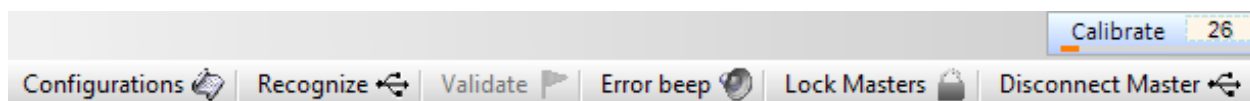
The language files are located in the "Docs" folder, near to the application ThereminoHAL.exe.

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 commands opens the documentation file.

The toolbar keys



Configurations

Allows you to manually edit the configuration. It may be convenient in some cases.

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 appropriate, 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.

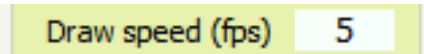
This does not mean that isolated applications can not communicate with each other. The modular communication is always possible and is done through the slot, which are common to all applications.

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

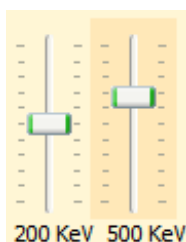
Moving the mouse up and down allows wide and fast adjustments

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 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 more intuitive to use the first for horizontal cursors and the second 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