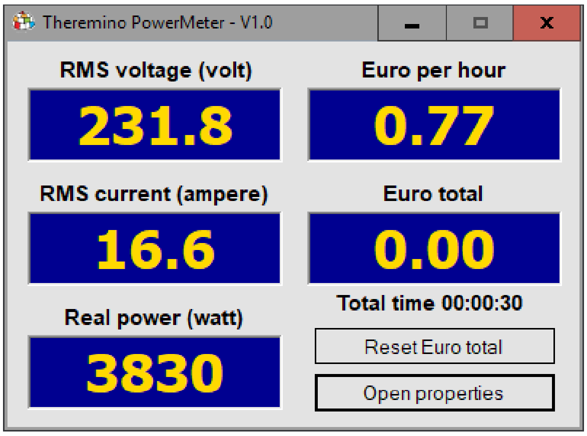
Theremino System

PowerMeter V1.0 Instructions

The Power Meter application



RMS voltage (volt): 231.8
Euro per hour: 0.77
RMS current (ampere): 16.6
Euro total: 0.00
Total time 00:00:30
Real power (watt): 3830

The Power Meter application is downloaded *from this page*, and it can be used in multiple ways:

- The easiest way is to read only the current, and multiply it by a fixed voltage value. In this case one obtains a power "apparent" that does not take account of any variations in the voltage, nor for any phase differences between current and voltage, and even of the waveform of current and voltage.
- Or you can read current and voltage and multiply them together. In this case one obtains an output "apparent", which does not take into account any phase shifts between current and voltage, and even the waveform of voltage and current.
- Or you can read the RMS values (the root of the sum of the squares) of the current and voltage, as well as correcting the phase shift of the measuring circuits. In this case we obtain the effective values of current and voltage, and calculates the real power.

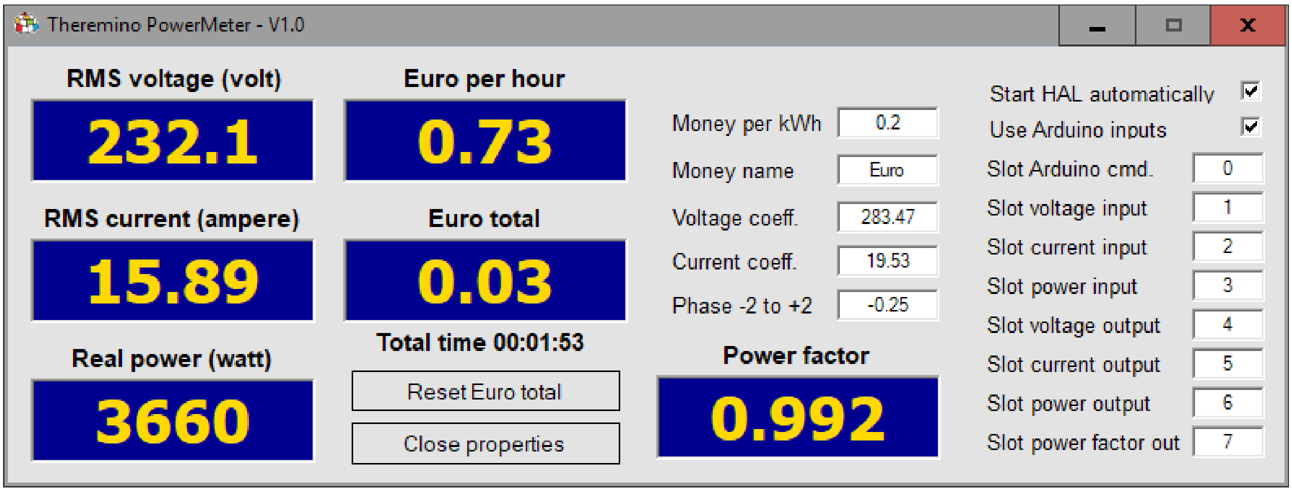
The real power is the one that actually affects on the electricity bill, and it is always lower than that apparent. A negative real power indicates a current flow in the opposite direction (useful for those who have solar panels). The difference between apparent and real power can be:

- Small, for heavy loads that are largely resistive (boiler, iron, oven, washing machine).
- Medium, for loads consisting mainly of an engine (refrigerator, vacuum cleaner, air conditioner).
- High, for small switching power supplies (computers, televisions, fluorescent and LED lamps, chargers, etc ..).

Measure the voltage, current and power “real values” is more complex, we will not use a Master but an Arduino, with the application and the library ArduHAL Theremino ***that are downloaded from here***. Finally, we must also be programmed with the Arduino ***firmware that you download from here***.

To change the operating mode, reading various sensors, adjust the calibration and use a Theremino Master module or an Arduino, you press "Open properties" and change the options, as we shall see in the following pages.

Adjusting properties



RMS voltage (volt) 232.1
RMS current (ampere) 15.89
Real power (watt) 3660
Euro per hour 0.73
Euro total 0.03
Total time 00:01:53
Money per kWh 0.2
Money name Euro
Voltage coeff. 283.47
Current coeff. 19.53
Phase -2 to +2 -0.25
Power factor 0.992
Start HAL automatically ☑
Use Arduino inputs ☑
Slot Arduino cmd. 0
Slot voltage input 1
Slot current input 2
Slot power input 3
Slot voltage output 4
Slot current output 5
Slot power output 6
Slot power factor out 7

The right half of the application contains the properties. With them we can run the application in many different ways, adjust the calibration coefficients, the name of the currency and also set which data read and write, and which Slots to use to read and write data.

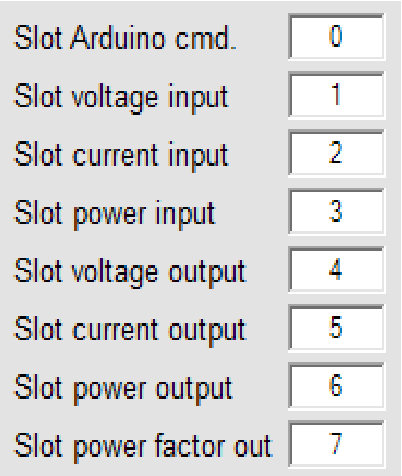
- **Money per kWh** - Cost of electricity that will be used to update the boxes "Euro per hour" and "Euro total".
- **Money name** - Name of the local currency, such as: Dollar, Euro, Yen, etc..
- **Voltage coefficient** - Full scale value for voltage measurement. The precise adjustment of this value must be carried out by comparison with a tester.
- **Current coefficient** - Full scale value for current measurement. The precise adjustment of this value must be performed by comparison with a tester, or with a current clamp.
- **Phase -2 to +2** - The adjustment for the phase shift produced by the the voltage and current measuring circuits. To trim this value you connect a resistive load (for example a 2 kW air heater) and adjust the number from -2.000 to 2.000, to achieve the maximum "Power factor". If possible you should get it to "1.000".
- **Start HAL automatically** - If you enable this checkbox, the HAL application starts automatically while opening the Power Meter, and stops when you close it. The Theremino_HAL and Theremino_ArduHAL applications must be in the same folder of the PowerMeter application.
- **Use Arduino inputs** - If this box is checked then the input data must be prepared in the Slot by an Arduino, otherwise by a Master module.
- **Slots** - These boxes indicates to the application, the Slots where write the commands for Arduino, from which to read the input data, and in which to write the output data. If you set a box with the "-1" value, then the function is disabled. In this way you set the behavior of the application, as we shall see in the next pages.

The input and output Slot

The Power Meter application follows the tradition of Theremino system, so it does not read the sensors directly, but uses the Slots.

The input values are read by the Slot where another application, typically an HAL, writes the data read by the sensors via a Master or an Arduino module.

The output values are written in the Slot, then other applications of the system will be able to read them and use them in various ways.



Slot Arduino cmd.	0
Slot voltage input	1
Slot current input	2
Slot power input	3
Slot voltage output	4
Slot current output	5
Slot power output	6
Slot power factor out	7

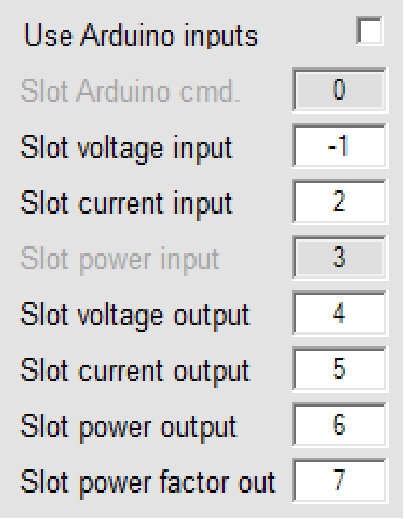
The Slot "Arduino cmd." Is used to send commands and the phase calibration to the Arduino (if you use it).

The Slots "Voltage, Current, and Power input" are used to read the voltage and current values measured by the sensors.

The Slots "Voltage, Current, Power and Power-Factor output" are used to send the calculated values and calibrated to other system applications.

Normally in this application the Slot numbers are from 0 to 7, as in this image. But if necessary, they could be set with any number up to 999. In this way you could also use multiple applications simultaneously on the same PC. By changing the values of the Slots you should also adjust appropriately the Slot values in the HAL application Pins.

Unused and disabled Slots



Use Arduino inputs	☐
Slot Arduino cmd.	0
Slot voltage input	-1
Slot current input	2
Slot power input	3
Slot voltage output	4
Slot current output	5
Slot power output	6
Slot power factor out	7

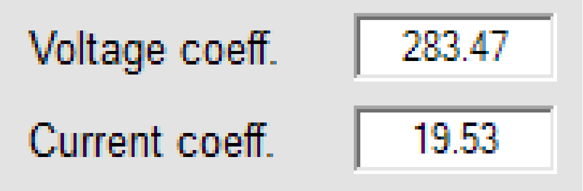
To set a Slot as unused, you set the value to "-1"

In this example we see the "Slot voltage input" with a "-1". In this way you inform the application that the voltage sensor does not exist.

The Slots "Arduino cmd." and "Power input" are gray and disabled because we do not use Arduino. In fact the "Use Arduino inputs" box is disabled.

Adjusting the spin boxes

The numeric boxes of this application (and all other applications of Theremino system) have been developed by us *(Note 1)* to be more comfortable and flexible than the Microsoft's original TextBox.



Voltage coeff. 283.47

Current coeff. 19.53

Numerical values are adjustable in many ways

- By clicking and holding down the left mouse button and moving it up and down.
- With the mouse wheel.
- With the arrow-keys up and down arrow on the keyboard.
- With PagUp and PagDown keyboard keys (variations ten times larger).
- With the usual methods that are used to write numbers with the keypad.
- With conventional methods of selection and copy-paste.

- Pressing SHIFT the rate of change it is multiplied by a hundred.
- Pressing CTRL the rate of change is multiplied by ten.
- Pressing ALT the rate of change is divided by ten.

- Move the mouse up and down allows large and fast regulations.
- The mouse wheel allows a comfortable and immediate adjustment.
- The arrow keys allow fine adjustments, without having to look away from what you are adjusting.

***(Note 1)** Like all our software the source files are available (Freeware and Open Source) and can be downloaded **from here**. These controls can be used freely in any project, also without name the source. They are "Open source" and this guarantee that we have not included malware.*

Circuits for measuring the AC voltages and currents

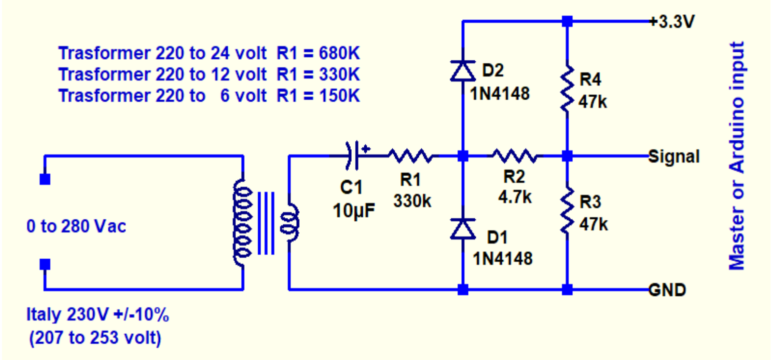
The Power Meter application uses the values of voltage and current that are in the Slots. These values may also come from sensors for direct current, but usually you measure the voltage and the alternating current of the mains.

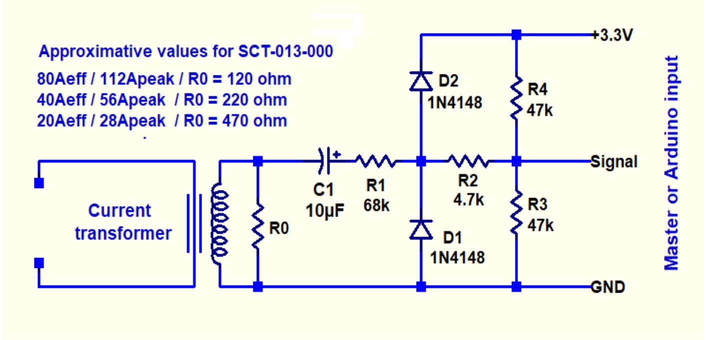
For the measures we will use the type of ADC inputs of a Theremino Master module or an Arduino. In both cases we have to lower the mains voltage and transform the current values ranging from zero to 3.3 volts, because these are the voltages measurable by the Master and Arduino Pins.

Connect your PC to the mains requires special attention.
So we will use transformers or other methods (Hall sensor)
to provide a total galvanic isolation.

If you have no experience get help from an electrician
or better from an electronics expert.
We will not accept liability.

These two schemes are used to measure the mains voltage and current.




Approximative values for SCT-013-000
80Aeff / 112Apeak / R0 = 120 ohm
40Aeff / 56Apeak / R0 = 220 ohm
20Aeff / 28Apeak / R0 = 470 ohm
+3.3V
D2
1N4148
R4
47k
Signal
Current
transformer
R0
C1
10µF
R1
68k
R2
4.7k
R3
47k
D1
1N4148
GND
Master or Arduino input

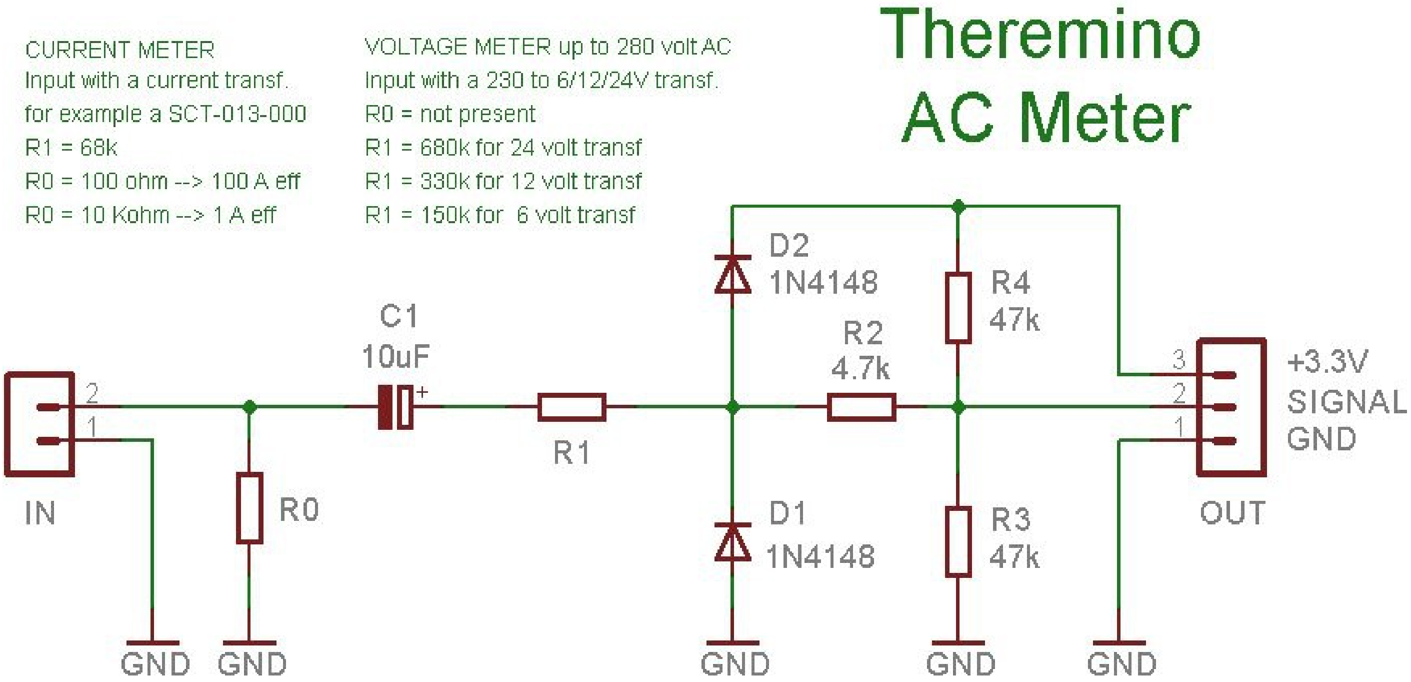
Circuit explanation

- The capacitor C1 removes the DC component from the signal and must be an electrolytic capacitor (polarized) or ceramic (non-polarized). The working voltage of this capacitor must be at least six volts.
- R3 and R4 produce a voltage midway between 3.3 volts and zero volt, so the center of the signal is positioned in the middle of the measuring range of the ADC.
- The D1 and D2 diodes limit the voltage and R2 limits the current to the ADC input, to no more than a few tens of microamperes. The protection given by D1, D2 and R2 is sufficient to protect the measuring circuits also in case of extra-voltages of tens of thousands of volts.
- In the left side circuit for measuring the voltage, R1 adjusts the scale, that is the maximum voltage that can be measured. Raise its value to increase the scale.
- In the right side circuit for measuring the current, the resistor R0 adjusts the full scale from the current to be measured, which also depends also a bit from the current transformer model. To increase the scale, decrease the R0 value.

Wiring of the measuring circuits

The two circuits, to measure the alternating voltage and current, can be easily mounted on a small piece of a thousand-holes, according to the scheme and following assembly plans.

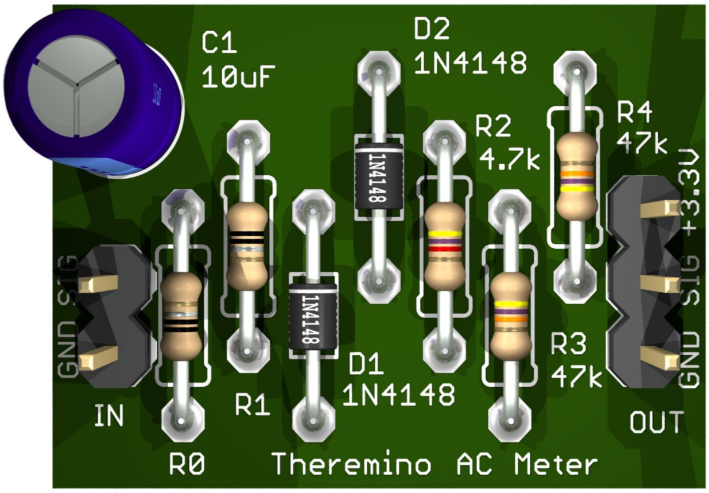
The circuit is identical to measure both voltage and current. The only differences are the presence or not of R0, the values of R0 and R1, and of course, the voltage or current transformer connected to the input. The details are explained in the next pages schemes.

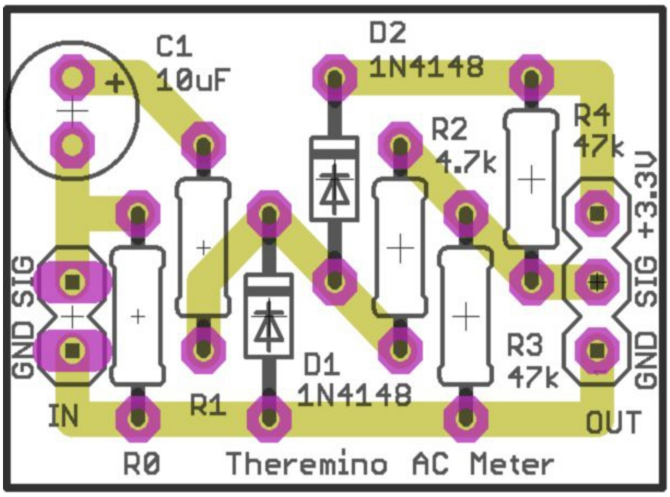


CURRENT METER
Input with a current transf.
for example a SCT-013-000
R1 = 68k
R0 = 100 ohm --> 100 A eff
R0 = 10 Kohm --> 1 A eff

VOLTAGE METER up to 280 volt AC
Input with a 230 to 6/12/24V transf.
R0 = not present
R1 = 680k for 24 volt transf
R1 = 330k for 12 volt transf
R1 = 150k for 6 volt transf

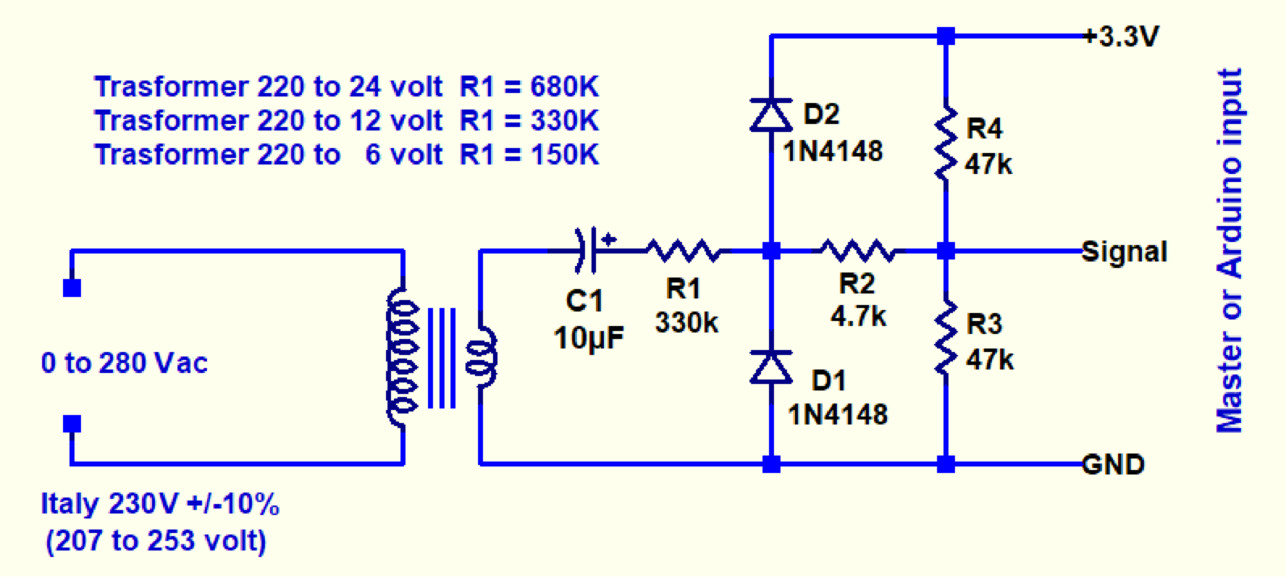
Theremino
AC Meter





For those wishing to print or milling this circuit, the Eagle files and images can be downloaded from the *PowerMeter page*.

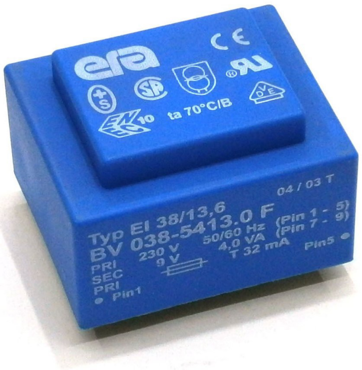
Circuit for measuring the alternating voltage



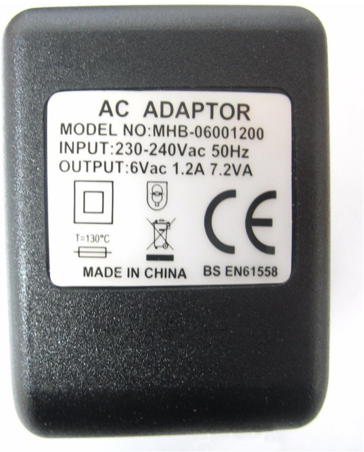
To measure the alternating voltage of the network, we will use this circuit connected to an ADC input of the Master module or Arduino.

The transformer and the resistor R1 must be adequate to the full-scale required. To measure the European network, from 230 volts +/- 10%, will choose a full-scale a bit more of the maximum 253 volts that the specific manufacturer with + 10%.

A 280 volt full-scale leave a few tens of volts of margin to be able to also measure any network anomalies and momentary extra-voltages caused by lightning.



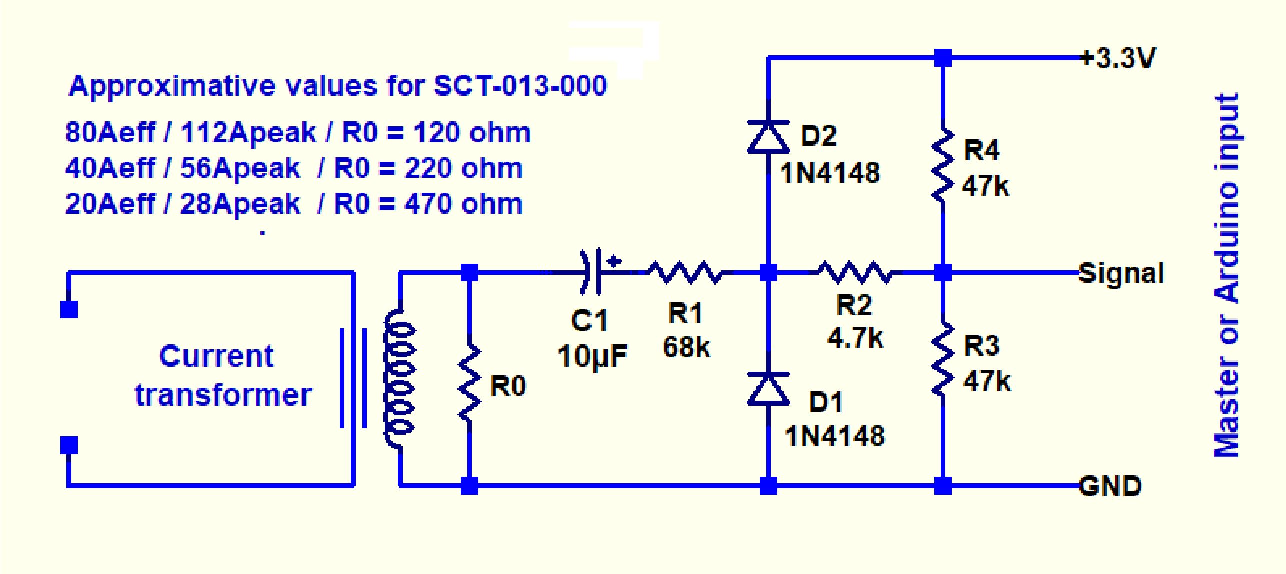
We could use any few watts transformer. A small transformer from one or two watts would be ideal.



Better still if we find one with a sealed plastic container and the cable with the mains plug. In this way we will not fiddle with the the network voltage wires and the safety will be greater.

But beware that inside there must be only one transformer and nothing else. So on the data there must be the symbol of the alternating (a horizontal S), and not two horizontal sections with a dashed. Or there has to be written 6Vac, 9Vac, 12Vac or 24Vac.

Circuit for measuring the alternating current

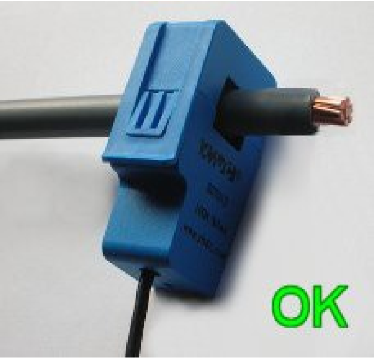

Approximative values for SCT-013-000
80Aeff / 112Apeak / R0 = 120 ohm
40Aeff / 56Apeak / R0 = 220 ohm
20Aeff / 28Apeak / R0 = 470 ohm
Current transformer
R0
C1
10µF
R1
68k
D2
1N4148
R2
4.7k
D1
1N4148
R4
47k
R3
47k
+3.3V
Signal
GND
Master or Arduino input

To measure the total alternating current (near the counter from the electrical network), or the current going to a load (electrical appliances, motors, PC, battery charger, etc ..), we use a current transformer, connected to an ADC input of the Master or Arduino module, by means of this circuit.

By changing the R0 value you set the full-scale from 1 ampere (R0 from 10 to 20k) to 100 amperes (R0 from 100 to 200 ohms). The R0 precise value depends on the transformer model that is used and in particular by its number of turns.

The R0 approximate values for the CR-3110 model are: 680 ohms for 20 ampere, 330 ohms for 40 ampere, and 270 ohms for 75 ampere of full scale. Instead, for the SCT 013-000 the values are: 470 ohms for 20 ampere, 220 ohms for 40 ampere, and 120 ohms for 80 ampere of full scale.

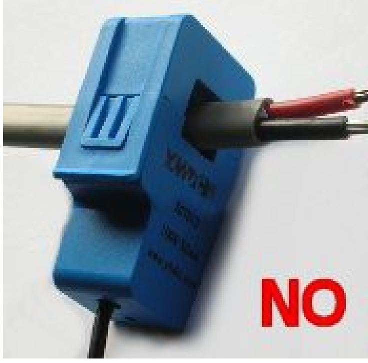
To test the current actual full scale, we use a strong resistive load (such as a 2000 watt hot air space heater) which consumes almost ten amperes. The precise value of this current it is measured with a tester or a current clamp. Then we start with R0 low and increase it, until the signal arrives at full scale (value of 1000 measured by the HAL in the current Slot if using an Arduino, or values changing rapidly from 0 to 1000 if using a Master). Finally, with a linear proportion, we establish the R0 value to obtain the desired full scale.



The current transformers have a large number of turns on the secondary. While the primary is a wire that passes into its hole (one of the two wires that feed the load).

In the transformer hole you must pass a single wire, as in the image on the left. And not the full power cable, consisting of two or three wires, as the image on the right.

On the next page are the data of some of the most common models.



Current transformers

The SCT-013-000 model is readily available on eBay for ten Euro (shipping included) or even five Euro (shipping included) by buying it in HongKong or the Chinese website ***store-ino***.

Model SCT-013-000 It has 1800 turns and maximum current of 100 amps. Warning, it must be really a 013-000 and not a 013-030 for example. All models that do not end with 000 have a fixed scale, and it can not be changed by changing the value of R0.

The model CR Magnetics CR 3110 is easily located, and costs about 15 Euro, has 3000 turns and can measure a maximum current of 75 amperes.

Useful information for those who want to do calculations and simulations

The voltage produced by a current transformer is calculated with this formula:

V = I * R / N

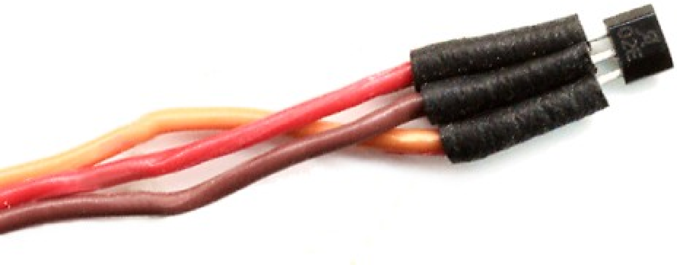
- V is the output voltage
- I is the current
- R is the load resistance
- N is the number of turns

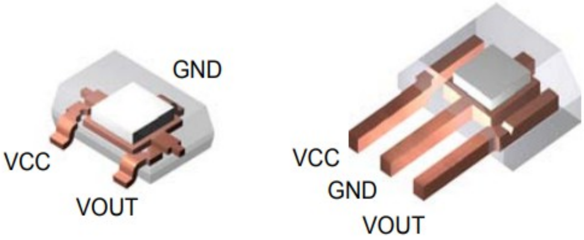
In reality, considering the magnetic resistive losses, the output current and therefore also the voltage, are a bit lower than it would be considering only the turns ratio.

In fact the model SCT-013-000 provides a current of 50 mA with a current on the primary of 100 A. So his report is 2000 times, even though its number of turns is 1800. And the CR-3110 model provides 24 mA, with a current of 75 amps on the primary, so its ratio is 3100 times, even though its number of turns is 3000.

Hall sensor for measuring the alternating current

As an alternative to the previous circuit you can measure the alternating current with a single component, an HALL effect magnetic field sensor. You put the sensor next to an alternating current wire and connect directly the sensor (without the circuits shown in the preceding pages) to an ADC input of the Master module or an Arduino.





- *GND is connected to GND of the Master or dell'Arduino*
- *VCC is connected to the 3.3 volts of the Master or dell'Arduino*
- *VOUT to the Master or Arduino signal input pin, configured as ADC.*

The sensor must be stably coupled to the wire of the phase or of the neutral (preferably the neutral if possible). Usually the neutral wire is blue and that of black or brown phase.

I remember for beginners that wires must be well insulated, DO NOT REMOVE THE INSULATING PLASTIC. The magnetic signal passes easy through the colored plastic coating.

To measure strong currents (full-scales of several tens of amperes), the signal should be enough, even with not particularly sensitive HALL sensors.

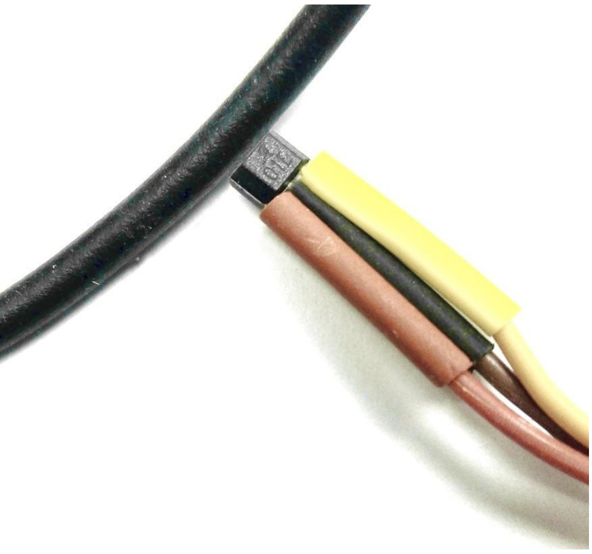
If the signal is too much, and you want to increase the full-scale, just slightly move the sensor from the wire. But no more than ten millimeters otherwise it risks that magnetic noise of the surrounding environment can influence the measurements.

To increase the signal use a more sensitive sensor, for example the DRV5056-A1 that is seen in the page tables ***Hall sensors available on the market***.

To greatly increase the sensitivity, and thus obtain a very low scale, you could make a coil, with many tens of turns of insulated wire and place the sensor in the middle. To increase further you could add an iron core or ferrite.

The instructions on the next page explain how to position the sensor and other ways to increase or decrease the sensitivity.

Place the Hall sensor

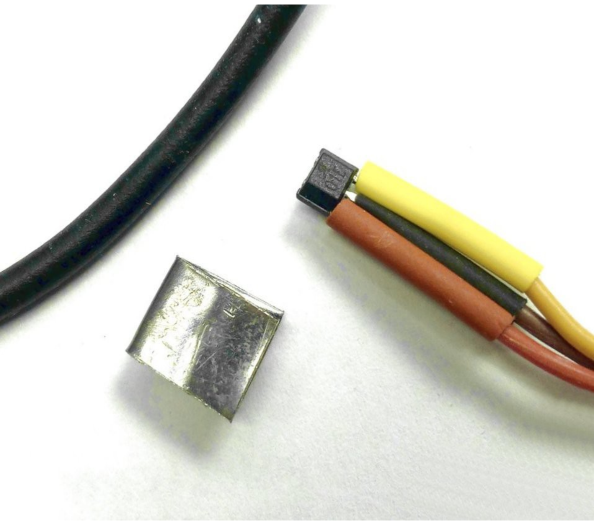


By adjusting the distance between the sensor and wire and you can achieve the desired scale.

In case you want to obtain the maximum sensitivity, the sensor must be positioned horizontally and with the tip against the wire, as seen in this image.

If, however, the sensitivity is too much you can move or rotate the sensor with respect to the wire. But if you have to reduce it is much better to use a less sensitive model.

Both the wire that the sensor must be very firm, just a small movement, or even just bend the wire, could change the calibration. So you could paste them on a hard surface, with cyano-acrylate glue or hot glue.

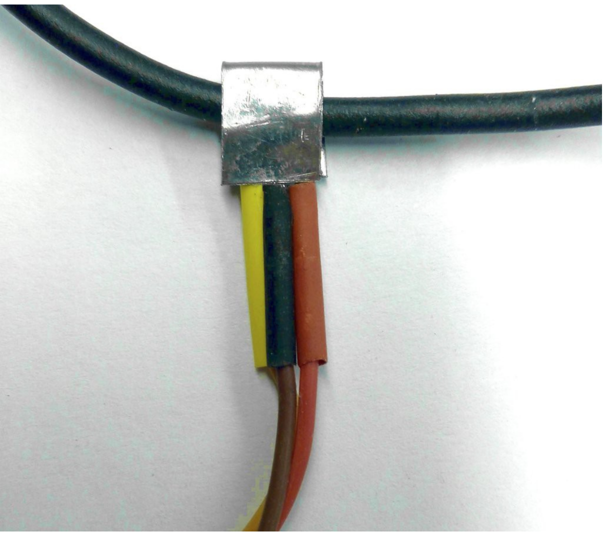


Increase the sensitivity with a bent iron rectangle.

A tin rectangle folded in a "U" can greatly increase the sensitivity (decreases the full scale).

The clip that is seen here on the left is composed of a 25 mm long and 7 mm wide rectangle. Composed by a 0.4 mm thick tinned iron plate. The length will be increased a bit for a very large diameter wire.

This clip increases the sensitivity of approximately six times and thus allows to measure currents six times lower with the same precision.

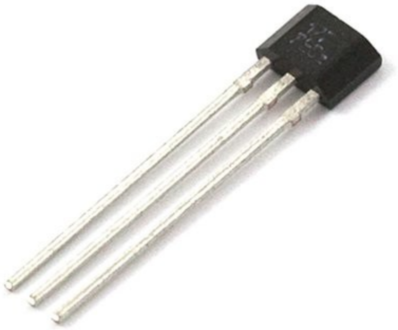


To obtain an increased sensitivity, the material must be magnetic. Instead, to maintain the sensitivity inaltered, one could use non-magnetic materials such as stainless steel, aluminum or plastic.

To check whether a material is magnetic, just try if it is attracted by a magnet. But be careful, take the test on a crop which then throw it away, otherwise the clip will remain magnetized and unbalance the measuring range.

To check that the clip is not magnetized, move away the wire and checked that the measured value by the ADC does not change approaching the clip to the sensor. Otherwise you could use a compass.

Hall sensors available on the market



These sensors must be connected to the 3.3 volt, because the voltage of 5 volts coming from the USB is too unstable and noisy.

So we will feed at 3.3 volts even models designed to be powered at 5 volts.

This table list the 3.3 volt sensors. The 5056-A1 and 5055-A1 models are very sensitive but not found on eBay. The A1319 model is also on eBay and is sensitive enough to measure medium and high currents (full-scale of twenty amps and up). To measure low currents the best is the first, ie theDRV5056-A1.

SENSOR (3.3 volt supply)	Sensitivity (mV / Gauss)	Max range (Gauss)
DRV5056-A1	20	+/- 95
DRV5056-A2	10	+/- 190
DRV5056-A3	5.0	+/- 390
DRV5056-A4	2.5	+/- 780
DRV5055-A1	10	+/- 220
DRV5055-A2	5.0	+/- 440
DRV5055-A3	2.5	+/- 880
DRV5055-A4	1.25	+/- 1760
A1319	5.0	+/- 500
A1318-2T	2.5	+/- 1000
A1318-1T	1.35	+/- 1850

This table list the sensors that, according to the specification, should be fed with 5 volts. But we tried them at 3.3 volts (and even less) and work perfectly. The advantage of these models is that they can be found more easily on eBay.

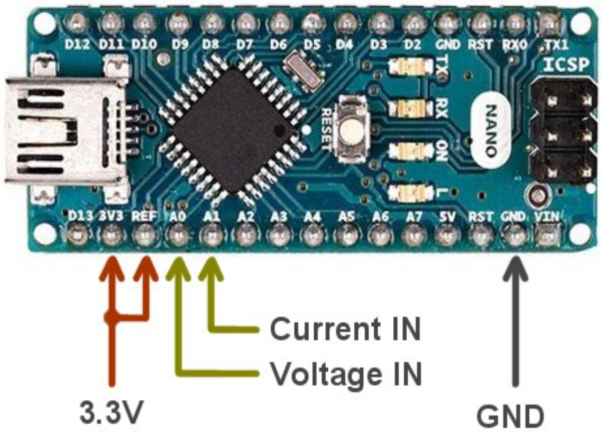
SENSOR (5 volt supply)	Sensitivity (mV / Gauss)	Max range (Gauss)
A1321	5.0	+/- 500
A1322	3.1	+/- 800
A1323	2.5	+/- 1000
A1301	2.5	+/- 1000
A1302	1.3	+/- 1900

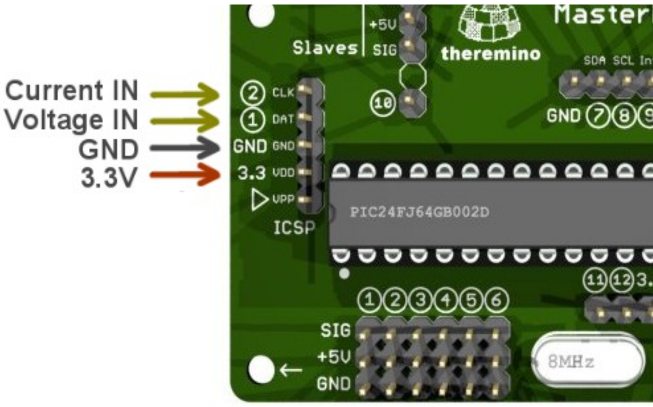
Electrical connections

All the input circuits on the previous pages can be connected to either to a Master or Arduino module.

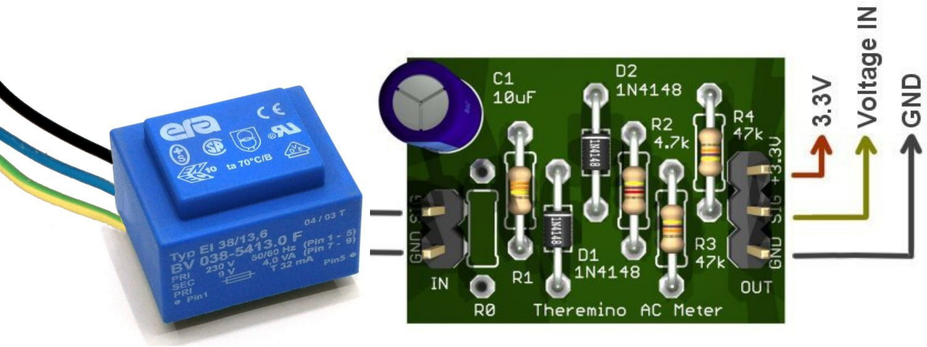
The Pins to use are GND, + 3.3V and Signal (Voltage IN and Current IN).

Here is the location of these Pins on the modules

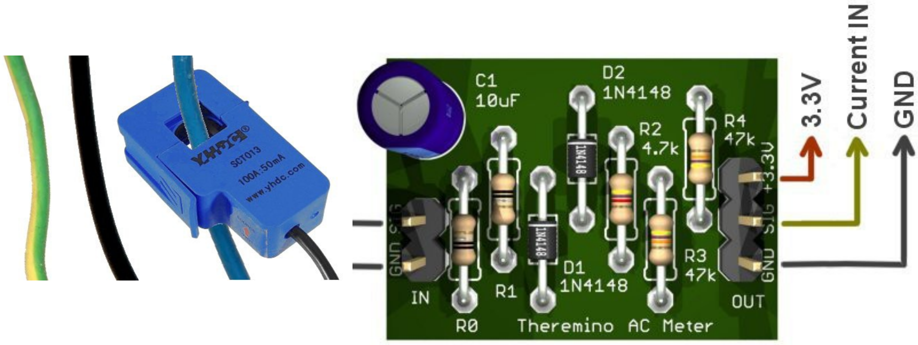


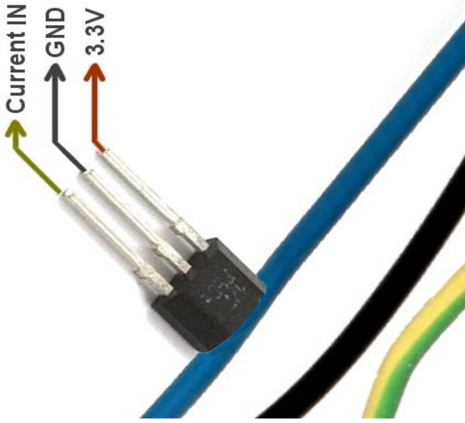


The voltage transformer and the circuit to measure the voltage

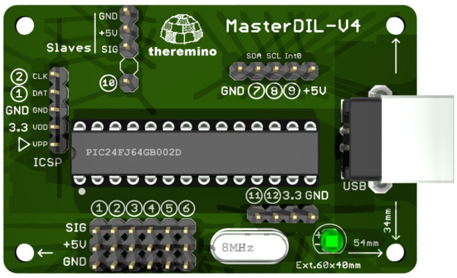


The current transformer and the circuit to measure it, or the magnetic sensor (right image)





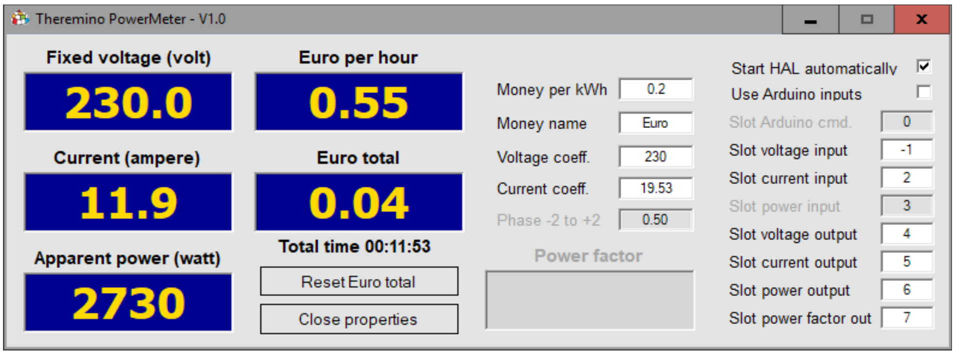
Working with a Theremino Master module



Using a Master module and a minimum of components is possible to calculate the apparent power, the precision of which is sufficient in many cases. An example would be to control a streetlights power supply line, to check that all the lamps are working, and that there are no abnormal consumption due to rain.

In the simplest case you need just a single component (an Hall sensor). With a single Master you could connect up to six sensors, and keep under control six feeder lines simultaneously.

Settings to calculate the apparent power, by measuring the current alone



Theremino PowerMeter - V1.0

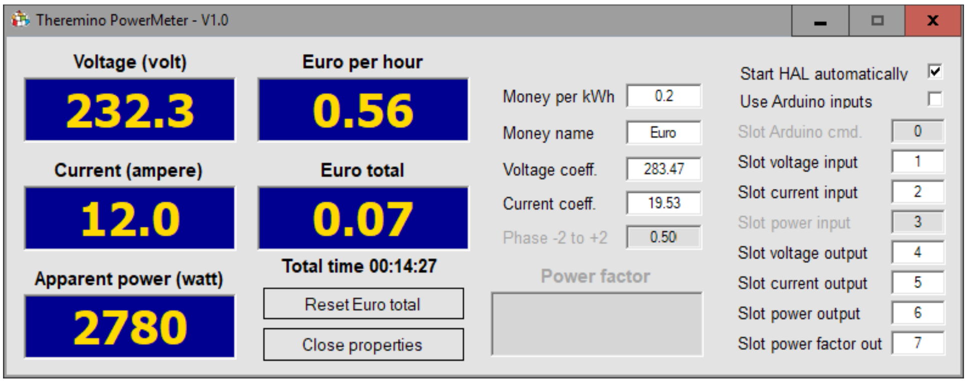
Fixed voltage (volt): 230.0
Euro per hour: 0.55
Current (ampere): 11.9
Euro total: 0.04
Apparent power (watt): 2730
Total time 00:11:53

Money per kWh: 0.2
Money name: Euro
Voltage coeff.: 230
Current coeff.: 19.53
Phase -2 to +2: 0.50

Start HAL automatically ☑
Use Arduino inputs □
Slot Arduino cmd.: 0
Slot voltage input: -1
Slot current input: 2
Slot power input: 3
Slot voltage output: 4
Slot current output: 5
Slot power output: 6
Slot power factor out: 7

- The "Slot voltage input" set to "-1", indicates not to read the voltage.
- The "Voltage coefficient." is set manually to the line voltage (measured with the tester).
- The "Current coefficient." is set with the calibration value, see the page ***Calibration***.

Settings to calculate the apparent power, by measuring voltage and current



Theremino PowerMeter - V1.0

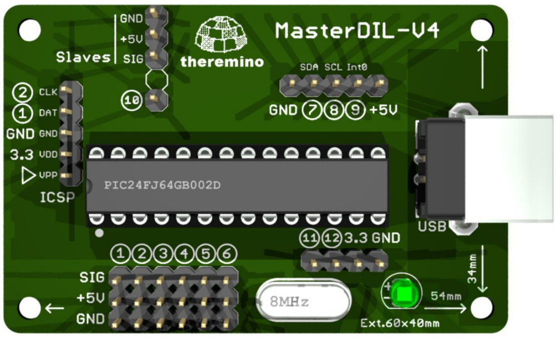
Voltage (volt): 232.3
Euro per hour: 0.56
Current (ampere): 12.0
Euro total: 0.07
Apparent power (watt): 2780
Total time 00:14:27

Money per kWh: 0.2
Money name: Euro
Voltage coeff.: 283.47
Current coeff.: 19.53
Phase -2 to +2: 0.50

Start HAL automatically ☑
Use Arduino inputs □
Slot Arduino cmd.: 0
Slot voltage input: 1
Slot current input: 2
Slot power input: 3
Slot voltage output: 4
Slot current output: 5
Slot power output: 6
Slot power factor out: 7

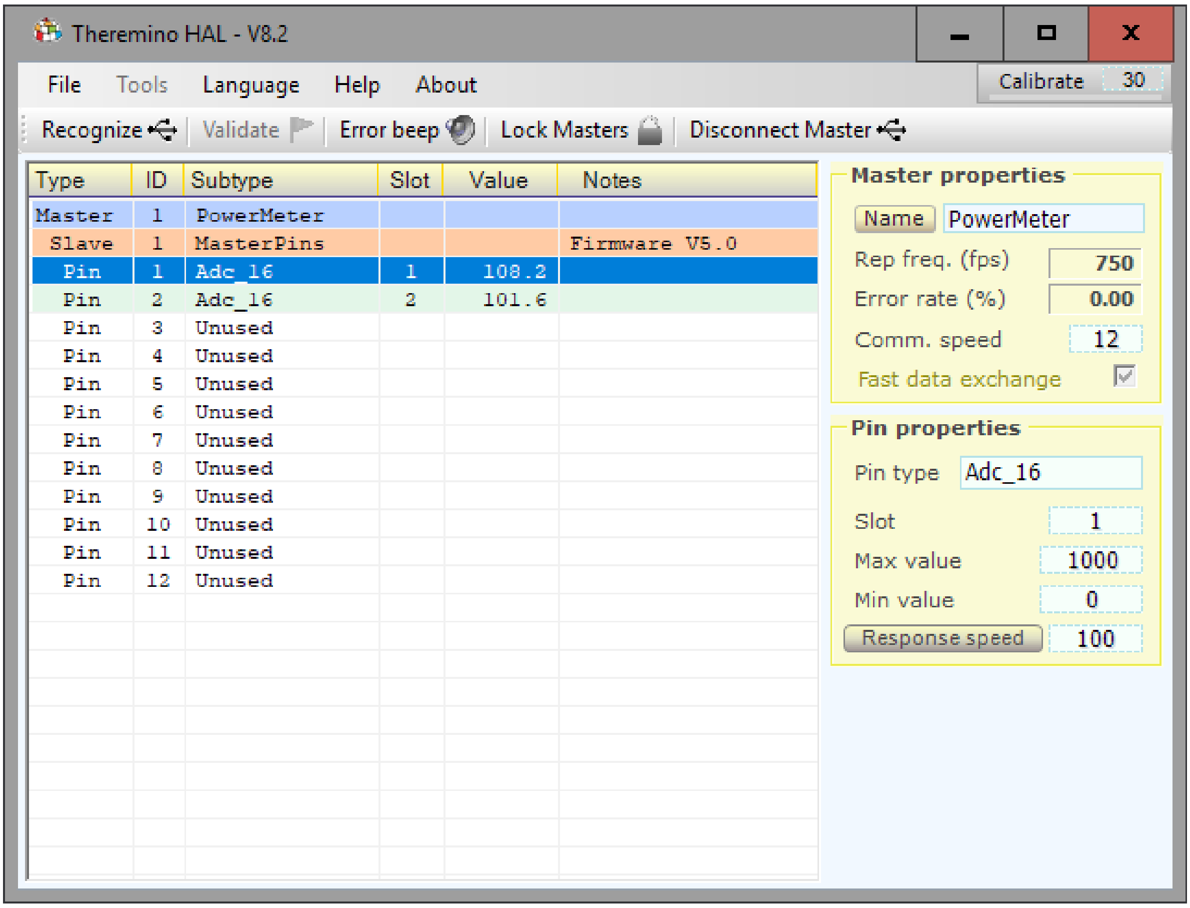
- The "Voltage coefficient." is set with the calibration value, see the page ***Calibration***.
- The "Current coefficient." is set with the calibration value, see the page ***Calibration***.

The Theremino HAL Settings



The communication with the Theremino Master module takes place by means of Theremino_HAL application. You will not need to download it separately, because it is already in the download of the application power meter.

To automatically start the Theremino_HAL application, his Theremino_HAL.exe file must be located in the same Theremino_PowerMeter.exe folder, so do not move files from the download locations.



Type	ID	Subtype	Slot	Value	Notes
Master	1	PowerMeter			
Slave	1	MasterPins			Firmware V5.0
Pin	1	Adc_16	1	108.2	
Pin	2	Adc_16	2	101.6	
Pin	3	Unused			
Pin	4	Unused			
Pin	5	Unused			
Pin	6	Unused			
Pin	7	Unused			
Pin	8	Unused			
Pin	9	Unused			
Pin	10	Unused			
Pin	11	Unused			
Pin	12	Unused			

Master properties

Name: PowerMeter
Rep freq. (fps): 750
Error rate (%): 0.00
Comm. speed: 12
Fast data exchange ☑

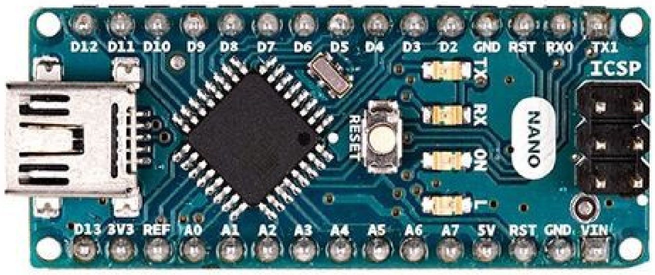
Pin properties

Pin type: Adc_16
Slot: 1
Max value: 1000
Min value: 0
Response speed: 100

- Comm speed must be 12. The Master read the voltage samples and alternating current directly via USB, therefore more exchanges are frequent and greater will be the accuracy.
- The Pin 1 and 2 must be configured as "Adc_16" and with Slot 1 and 2 (or as specified in the PowerMeter application).
- All Pin must have MaxValue = 1000, MinValue = 0, Response Speed = 100 and the “Response speed” button not pressed (yellow as in this image).

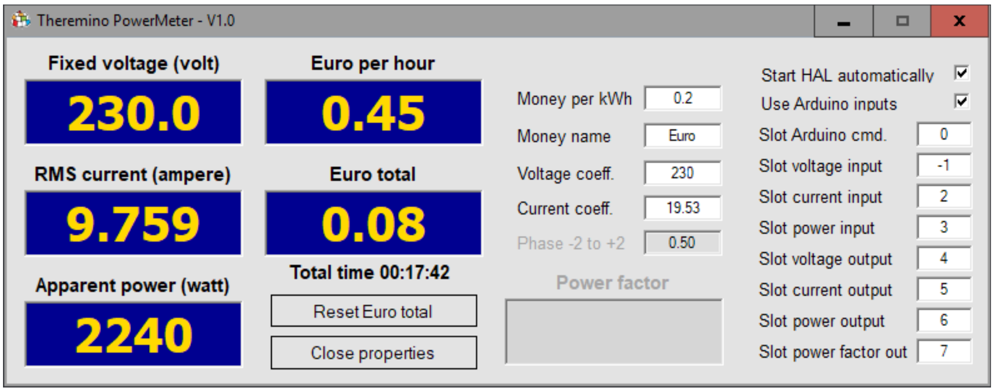
You will find these ready-made settings both in the NoName and PowerMeter configurations. You could select a configuration by clicking on the name.

Operation with an Arduino


D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
ICSP
TX
RX
RESET
ON
NANO
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

Using an Arduino module (better if a Nano) you can measure the effective values of voltage and current, as well as compensate for the phase shifts of the measuring circuits and calculate the real power with good precision.

Settings to calculate the apparent power, by measuring the current alone



Fixed voltage (volt): 230.0

Euro per hour: 0.45

RMS current (ampere): 9.759

Euro total: 0.08

Apparent power (watt): 2240

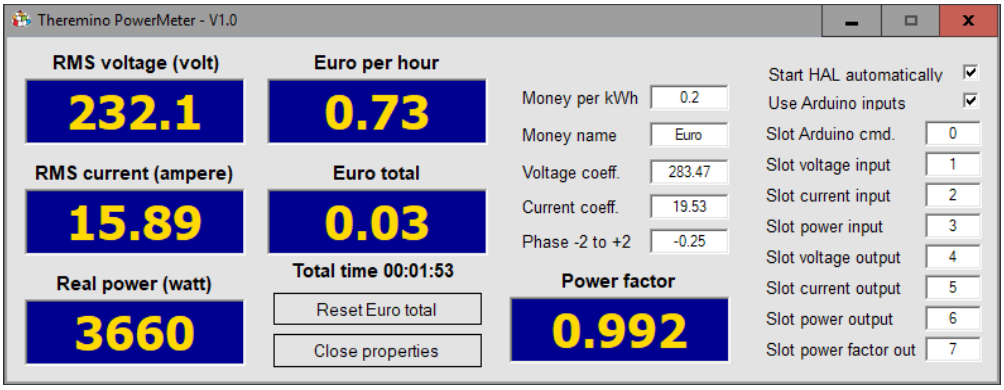
Total time 00:17:42

Money per kWh: 0.2
Money name: Euro
Voltage coeff.: 230
Current coeff.: 19.53
Phase -2 to +2: 0.50
Power factor

Start HAL automatically ☑
Use Arduino inputs ☑
Slot Arduino cmd.: 0
Slot voltage input: -1
Slot current input: 2
Slot power input: 3
Slot voltage output: 4
Slot current output: 5
Slot power output: 6
Slot power factor out: 7

- The "Slot voltage input" set to "-1", indicates not to read the voltage.
- The "Voltage coefficient." is set manually to the line voltage (measured with the tester).
- The "Current coefficient." is set with the calibration value, see the page ***Calibration***.

Settings for calculating the real power, by measuring voltage and current



RMS voltage (volt): 232.1

Euro per hour: 0.73

RMS current (ampere): 15.89

Euro total: 0.03

Real power (watt): 3660

Total time 00:01:53

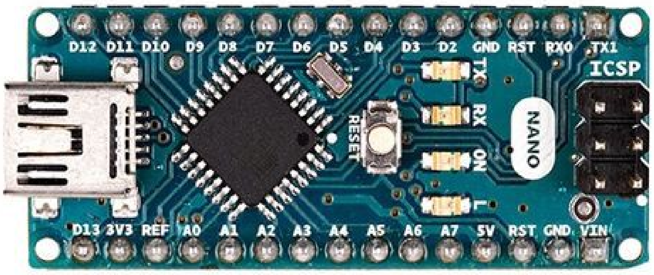
Money per kWh: 0.2
Money name: Euro
Voltage coeff.: 283.47
Current coeff.: 19.53
Phase -2 to +2: -0.25
Power factor: 0.992

Start HAL automatically ☑
Use Arduino inputs ☑
Slot Arduino cmd.: 0
Slot voltage input: 1
Slot current input: 2
Slot power input: 3
Slot voltage output: 4
Slot current output: 5
Slot power output: 6
Slot power factor out: 7

- The "Voltage coefficient." is set with the calibration value, see the page ***Calibration***.
- The "Current coefficient." is set with the calibration value, see the page ***Calibration***.
- The "Phase" box is set with phase calibration value see ***Adjustment of the phase shift***.

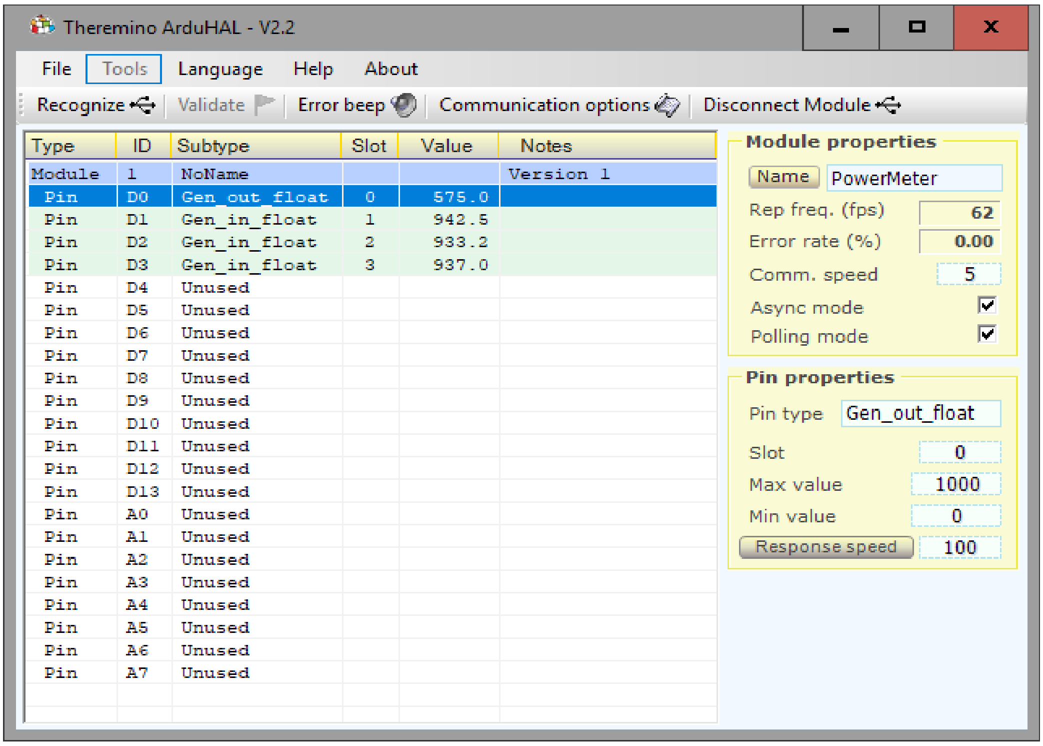
The ArduHAL settings

The communication with the Arduino module takes place by means of the Theremino_ArduHAL application. You will not need to download it separately, because it is already in the download of the application power meter.



To start automatically the ArduHAL application, his Theremino_ArduHAL.exe file must be located in the same Theremino_PowerMeter.exe file folder, so do not move files from how you download from the site.

The Arduino module must be programmed with the library Theremino ***download from here*** and with the PowerMeter sketch ***that you download from here***.



Type	ID	Subtype	Slot	Value	Notes
Module	1	NoName			Version 1
Pin	D0	Gen_out_float	0	575.0	
Pin	D1	Gen_in_float	1	942.5	
Pin	D2	Gen_in_float	2	933.2	
Pin	D3	Gen_in_float	3	937.0	
Pin	D4	Unused			
Pin	D5	Unused			
Pin	D6	Unused			
Pin	D7	Unused			
Pin	D8	Unused			
Pin	D9	Unused			
Pin	D10	Unused			
Pin	D11	Unused			
Pin	D12	Unused			
Pin	D13	Unused			
Pin	A0	Unused			
Pin	A1	Unused			
Pin	A2	Unused			
Pin	A3	Unused			
Pin	A4	Unused			
Pin	A5	Unused			
Pin	A6	Unused			
Pin	A7	Unused			

Module properties

Name: PowerMeter
Rep freq. (fps): 62
Error rate (%): 0.00
Comm. speed: 5
Async mode ☑
Polling mode ☑

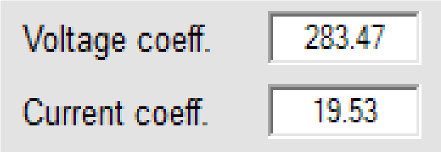
Pin properties

Pin type: Gen_out_float
Slot: 0
Max value: 1000
Min value: 0
Response speed: 100

- Comm speed could be 5 (because Arduino sends only few data per second)
- Async mode and polling mode must be enabled.
- The D0 Pin must be configured as "Gen_out_float" and D1, D2 and D3 Pins must be configured as "Gen_in_float"
- The Slots must be 0, 1, 2 and 3 (or as specified in the PowerMeter application)
- All Pin must have MaxValue = 1000, MinValue = 0, Response Speed = 100 and the “Response speed” button not pressed (yellow as in this image).

You will find these ready-made settings both in the NoName and PowerMeter configurations. You could select a configuration by clicking on the name.

Voltage and current calibration



Voltage coeff. 283.47

Current coeff. 19.53

With these adjustments, you calibrate the voltage and current measuring inputs.

Grossly these two values correspond to the two full-scale, ie the measurable maximum values of voltage and current. But it is good to make some clarifications:

1) The full-scale values are not determined by these boxes but from the input circuits, but from their components and their electrical properties.

2) To obtain accurate measurements, these two numbers will must be finely adjusted, so as to correspond exactly to the characteristics of the input circuits.

3) The voltages and the input currents can overcome, even much, the scale expected and the values written in these boxes. In these cases, the measures will stop to full scale values, but nothing will break. The input circuits that we propose can withstand tensions thousands of times greater, and currents significantly higher than the expected full scale.

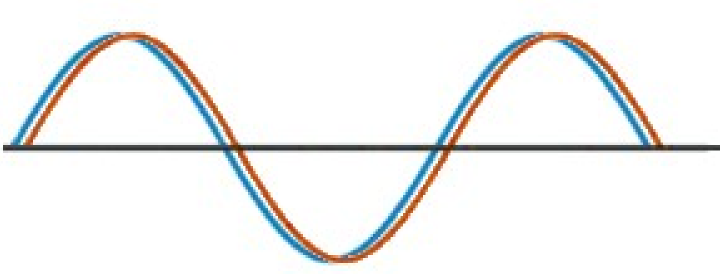
Fine adjustment of the "Voltage coefficient"

- Connect a voltmeter (or a tester) in parallel with the input of the voltage measuring circuit.
- The voltmeter must be adjusted in AC for alternating current measurements and its full scale must be adapted to the voltage to be measured.
- Adjust the “Voltage coeff.” value until the box "Voltage" or "RMS voltage" will show the same voltage as indicated by the voltmeter. If the voltage change over time, try to adjust them quite similar. A tolerance of one percent is already good.

Fine adjustment of the "Current coefficient"

- Connect an ammeter (or a tester) in series to the input of the current measuring circuit. Caution: in series, not in parallel, otherwise you create a dangerous short circuit. If you do not know how to do, ask for help from an electrician.
- The ammeter must be adjusted in AC, for alternating current measurements, and its full scale must be adapted to the current to be measured.
- Connect a heavy load in series with the measuring circuit and the ammeter. Attention: all in series. The load should be resistive and at least half of the full-scale. For example, for a by three kilowatts domestic installation, and a full-scale of about 20 amps, you could use a 2000 watt hot air heater. Better yet, if you use two of them, setting them for a total of about 3000 watts (13 amps at 230 volts).
- Adjust the “Current coeff.” value until the "Current" or "RMS current" box will show the same voltage as indicated by the ammeter. If the current changes over time, try to adjust them quite similar. A tolerance of one percent is already good.

Adjustment of the phase shift

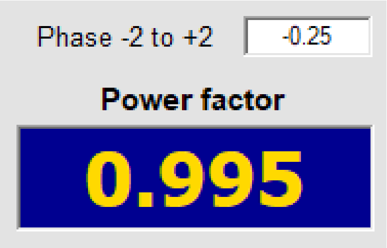


Small phase shift between voltage and current caused by the measuring circuits.

The inductances and capacitances of the measuring circuits could create phase shifts between voltage and current. Phase shifts are small, as can be seen in this image below, but could produce significant errors in the calculation of the "real power" and the "power factor" that is the ratio between real power and apparent power.

Using a unique algorithm (in the firmware we wrote for Arduino), we can correct phase errors up to two positive and negative samples, corresponding to +/- 15 electrical degrees. Our algorithm can change the Phase two times more, respect to similar software (eg ***Open Energy Monitor***). In addition, our algorithm does not cause variations in signal amplitude and does not require a subsequent trimming of the other calibrations.

Making the phase adjustment



Phase -2 to +2 -0.25

Power factor

0.995

This option is available only by using an Arduino module and setting the application to measure the real power.

- Connect a strong resistive load. For example, if the current full-scale is twenty amps, we have to find a load of about ten amperes or more. So an hot air heater would be fine, even better if they use two, and get three kilowatts, for a total of about thirteen amperes.
- Adjust the "Phase" box with values from -2.00 to +2.00 in order to maximize the "power factor". Ideally you should get to 1.000 but also 0.999 will be fine.

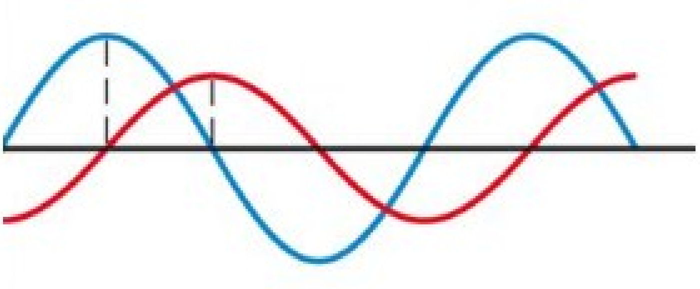
To adjust the phase you could use the keyboard or the mouse, as explained in ***this page***.

Accuracy obtainable

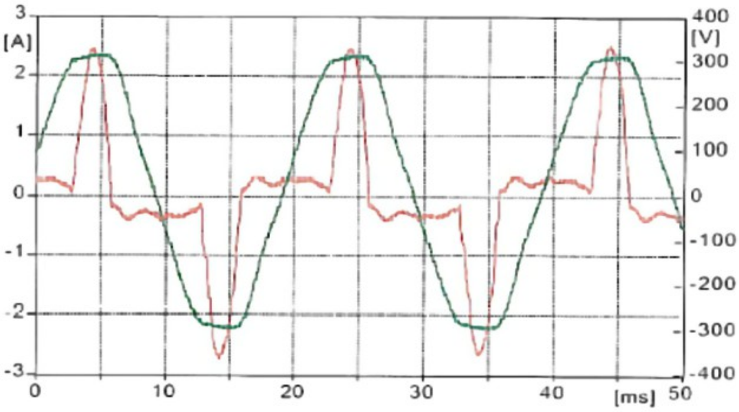
It is impossible to produce anything with absolute precision. Increase the precision of the input circuits would be expensive, and in any case would not lead to a good precision.

The components of the measuring circuits have tolerances. You could buy expensive versions of the resistors but it might not be worth it, because the transformers are all different and therefore a calibration would always be required.

If the calibration is performed by comparison with a good tester, you can be up to one percent (with resistive loads). This may seem low, but instead has an accuracy of respect in this field. The electric meters, which adopt the latest technology, have a precision around two percent, for inductive loads, and of one percent for resistive loads.



Phase shift between voltage (blue line) and current (red line), caused by an inductive load.



Supply voltage (green) and load current (in red). In this case the non-sinusoidal current is due to the switching power supplies of a computer and a monitor.

The flattened voltage on the peaks is due to the electricity grid limits, loaded with numerous feeders diodes and capacitors, which consume power only in the vicinity of the peaks.

For resistive loads the only limit is the accuracy of the tester that we use for comparison, while for non-resistive loads and for non-sinusoidal waveforms, the accuracy is limited around the values between 2% and 5%, due to the Arduino speed, that can calculate the RMS value of only fifty samples per cycle of the alternating voltage.

So, to measure multiple loads, we will use more Arduino modules, one for each load to be measured.

Alternatives to the Power Meter application

We wrote the Power Meter application and the Arduino firmware, to show the algorithms and how to make calculations.

In some cases someone could develop a different application. For example, those who want to keep tabs on solar panels or for those who want to make a LOG of many users simultaneously.

We have therefore tried to simplify the calculations and to maintain ease. The only complex parts of the Power Meter application is the user interface, the numbers presentation and the damping (smoothing), to prevent the numbers to dance too much. If you delete these parts remain a few tens of rows, easily usable in other applications and in other languages.

If you use a Master module you will have to frequently call (approximately 500 times per second), the function "ReadThereminoMasterInputs", which is located at the end of "Module_Inputs" module. This simple function (fifteen rows), continuously reads the instantaneous values of voltage and current, identifies the minimum and maximum, and every second, prepares the values RawInputVoltage and RawInputCurrent, which will be used in the next three lines.

```
Voltage       = RawInputVoltage * VoltageCoeff / 1000
Current       = RawInputCurrent * CurrentCoeff / 1000
ApparentPower = Voltage * Current
```

If you do not use Arduino, voltage and current are not RMS (mean square), and the power is not the real power. So in these three lines we calculate only the apparent power, and not calculate the power factor.

If you use Arduino, that prepares voltage and RMS current and the real power in the Slots, the calculations to do, are only the eight following lines:

```
RawInputVoltage = Slots.ReadSlot_NoNan(SlotVoltageInput)
RawInputCurrent = Slots.ReadSlot_NoNan(SlotCurrentInput)
RawInputPower   = (Slots.ReadSlot_NoNan(SlotPowerInput) - 500) / 500
Voltage         = RawInputVoltage * VoltageCoeff / 1000
Current         = RawInputCurrent * CurrentCoeff / 1000
ApparentPower   = Voltage * Current
RealPower       = RawInputPower * VoltageCoeff * CurrentCoeff
PowerFactor     = RealPower / ApparentPower
```

The Arduino firmware

The firmware for the Arduino is well-studied and will not require changes. However, an Arduino can read only a voltage and a current sensor (limits due to the processor speed). So to measure more than one load, we will use more Arduino modules, and our ArduHAL application will communicate with all modules Arduino simultaneously.

Instead a Master module could be read up to six ADC inputs (therefore probably the mains voltage and five current consumption). In this case, you need to add some lines to the "ReadThereminoMasterInputs" function and, as already explained, you will not get the RMS values and the real power, but only values of voltage, current and apparent power.