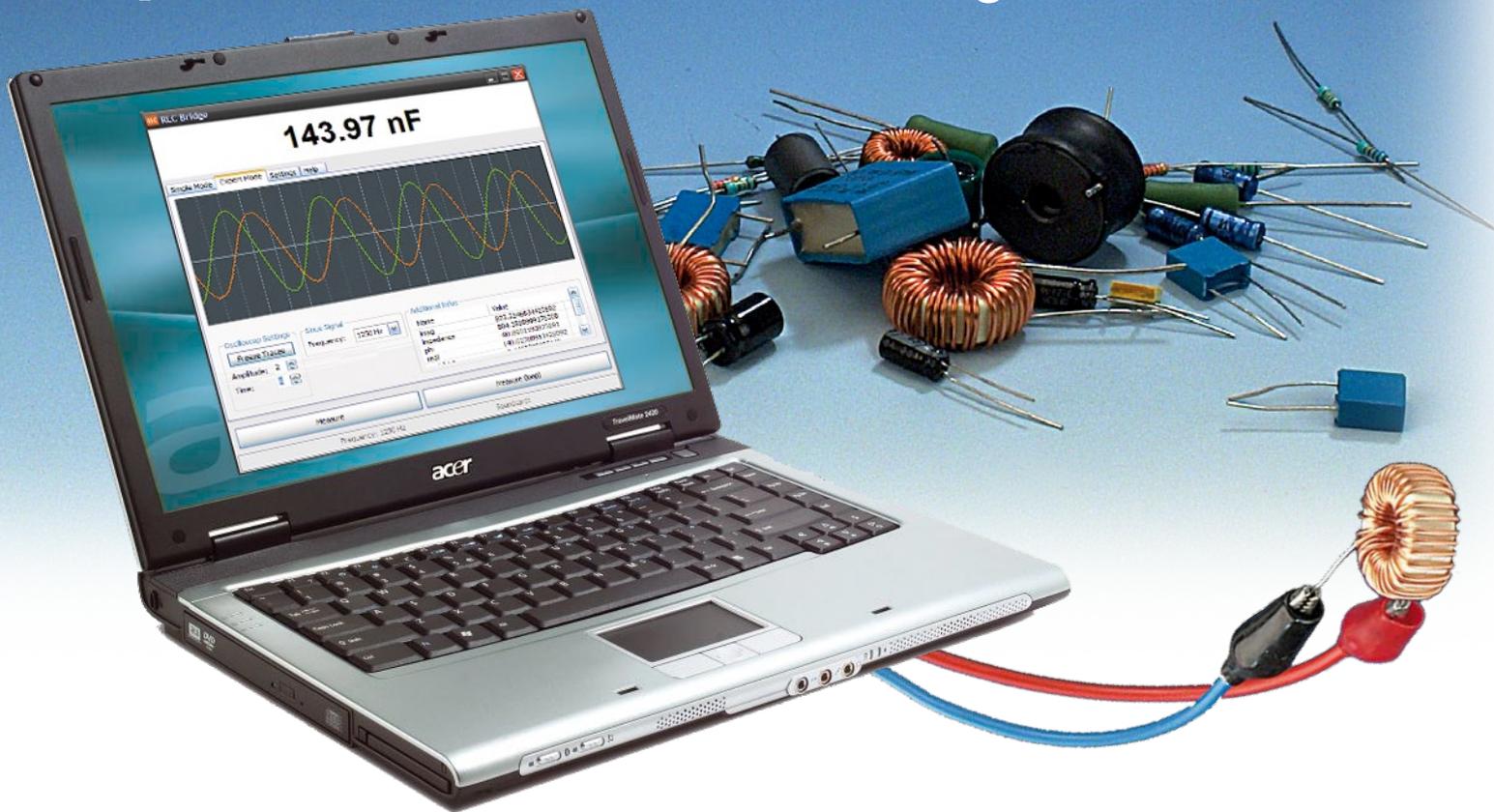


# 2-Pound RLC Meter

## impedance measurement using a sound card



Martin Klaper and Heinz Mathis

**Is it possible to make an RLC meter for less than two pounds? In this article the authors answer this question with a resounding 'yes' in the form of a simple and compact circuit that will enable you to make RLC measurements rapidly, accurately, and, above all, cheaply.**

For many years the two authors have used a Marconi RLC bridge. To use this device two controls are adjusted until a meter reads 'null', and then the value of the connected resistor, capacitor or inductor can be read from the settings of the controls. It is also possible to use the instrument to measure the loss factor  $\tan \delta$  and the  $Q$  (quality) factor. Not every household is lucky enough to own such an instrument or an expensive RLC meter; PCs and sound cards, however, are ubiquitous. And it turns out that these can be used to make excellent measuring instruments, as one of the authors has already described in

*Elektor* [1]. There we described how to make an ECG (electrocardiogram) recorder using a sound card, and it is a relatively small step from that to the idea of measuring impedance using a sound card.

In outline, this is how it works: using the two input channels of a stereo sound card we can measure two voltages simultaneously. A resistor in series with the device under test ( $R$ ,  $L$  or  $C$ ) is used to convert the current flowing through it into a voltage. If an alternating voltage is applied to the device under test and the resulting current is measured, we can calculate its

(complex) impedance. The alternating test voltage can be provided by an output of the sound card. Could we implement all of this on a PC? After a little contemplation, soldering and programming, the answer turned out to be yes.

Almost any PC can be used for this project, even a dusty old 500 MHz machine. There is even no need to open it up, as we only need access to the external connections of the sound card. Of course, we cannot guarantee that every PC will work, but we have tried a range of machines, desktop and laptop, running Windows XP and Vista,

and all have worked perfectly. Besides the PC very little is required: build the tiny circuit described below, connect it to the PC, and run the software provided.

The hardware consists of just two resistors and a dual operational amplifier: total cost well under two pounds. The circuit can conveniently be built on perforated board or on a breadboard, and then, low-cost impedance meter in hand, you can test inductors, capacitors and resistors to your heart's content. The circuit gives astonishingly good accuracy: its results have been checked against a much more expensive RLC meter on the authors' bench.

### Impedance

Impedance (from the Latin 'impedire', to hinder, hence 'impediment' etc.) is essentially the degree of opposition to current flow. It is a complex quantity, that is, it has real and imaginary parts. The impedance of a pure resistor has zero imaginary part, although the impedance of a practical resistor (and in particular a wirewound resistor) will have a small imaginary part. Ideal inductors and capacitors have a purely imaginary impedance, but again real inductors and capacitors have a small real part to their impedance along with a (hopefully dominant) imaginary part. This deviation from the ideal gives rise to power dissipation or losses in the component. For this project we imagine a real component to comprise an ideal component in series with a pure resistance, the latter representing the losses. At a specified frequency an impedance can be expressed in polar coordinates or in Cartesian coordinates:

$$Z = |Z| \angle \theta = R + jX$$

where  $|Z| = \sqrt{R^2 + X^2}$  and  $\theta = \arctan(X/R)$

Here  $Z$  is the (complex) impedance, measured in ohms,  $|Z|$  is the magnitude of  $Z$ , and  $\theta$  is the argument of  $Z$ .  $Z$  has real part  $R$  and imaginary part  $jX$  (see also the formulae in the accompanying box).

Below we will describe two ways of measuring impedance, which (although both inspired by the previous article in *Elektor* mentioned above) have been developed independently of one another. Method A is an adaptive linear combiner using a least-squares method which makes a measurement at constant frequency, calculat-

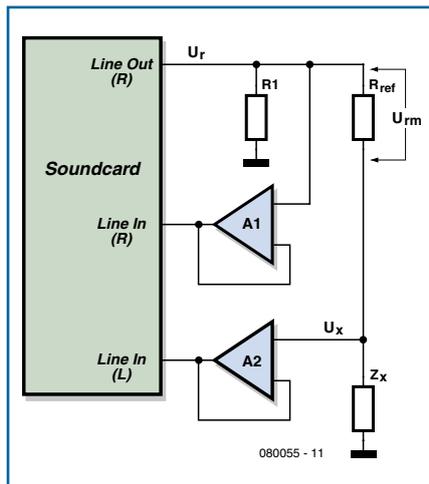


Figure 1. Block diagram for method A.

ing phase lag and loss factor. Method B goes by the delightful name of 'approximation of the characteristic curve using a variable test frequency and the method of least squares'. And now to the gory details.

### Method A

One of the two output channels of the sound card drives a voltage divider, consisting of a reference resistor  $R_{ref}$  and the device under test  $Z_x$ , with a sine wave (Figure 1). The second output of the sound card is unused. It could be used in an enhanced version of the meter to allow for switching ranges automatically (in the current version the range has to be changed manually). The two input channels measure the two voltages on the voltage divider, which allows the ratio between these two quantities to be measured at any instant. In principle it

would be possible to assume that the voltage applied to  $R_{ref}$  is simply proportional to the voltage that has been output by the program to the sound card. In practice, however, sound cards have a certain latency which means that there is a delay between the program sending a signal to the sound card output and its actually appearing there (see our test of sound cards elsewhere in this issue). Using two inputs on the sound card gets around this problem in an elegant way.

$R1$  references the output of the sound card to ground. The two operational amplifiers, each with a gain of one, act as buffers with a high input impedance and a low output impedance. The sinusoidal voltage  $U_r$  (from 'line out'), which is applied to the test circuit, is measured on the right input channel. The voltage across the device under test,  $Z_x$ , is measured using the left input channel. The operational amplifier used is a type LM358, although any similar device would do equally well. The supply voltage for the opamp is relatively low at  $\pm 3V$ , in order to protect the sound card input from damage in the event of something going wrong.

The accuracy of the measurements made depends on the accuracy of reference resistor  $R_{ref}$ . It is therefore very important to know the value of this resistor as accurately as possible.  $R_{ref}$  can be changed to change the measurement range: for best results the value of  $R_{ref}$  should be comparable to the impedance of the device under test. For added convenience it would be possible to provide for automatic range switching using the spare output channel of the sound card.

### The concept

The ratio between the amplitudes of the voltages dropped across the reference resistor and the device under test, and the phase angle between these voltages, are the keys to computing the impedance of the device. The series combination of the reference resistor and the device under test is driven by the signal  $U_r$ , which is one output of the sound card; a voltage is dropped across the device under test whose amplitude and voltage depend on its impedance. Our first approach works as follows: we measure using a constant-frequency stimulus voltage (1250 Hz for example) and take the measured signal at  $U_r$  into an adaptive linear combiner implemented in software. The combiner employs two vari-

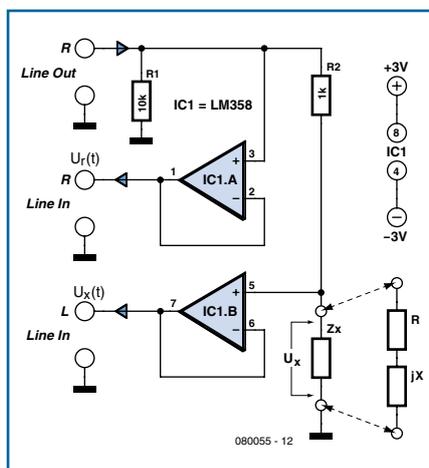


Figure 2. The hardware for method A consists of just a dual opamp and two resistors.

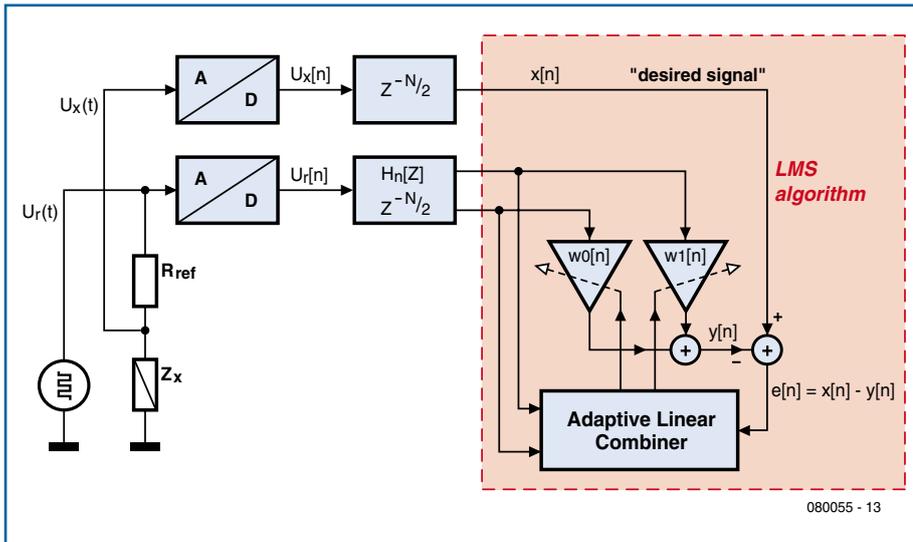


Figure 3. Functional diagram for method A.

able weights,  $w_0$  and  $w_1$ , which can be thought of as equivalent to the potentiometers in an analogue RLC bridge. The program adjusts the weights until the voltage at the output of the combiner matches that across the device under test as accurately as possible in amplitude and in phase. By 'as accurately as possible' we mean that we try to minimise the mean squared difference between the two signals. The complex impedance of the device under test can then be determined from the weight values  $w_0$  and  $w_1$ . The system thus works in a very similar fashion to the authors' trusty Marconi bridge, with the adjustment of the (virtual) potentiometers under software control.

**The algorithm**

We use the so-called LMS (least mean squares) algorithm, which is an iterative approach to finding the optimal

values of the weights  $w_0$  and  $w_1$ . At each time step it adjusts the weights and then recomputes the real and imaginary parts of the resulting signal and thence the resulting error. The adjustment is a small step in the direction which reduces the error as quickly as possible (the 'method of gradient descent'), seeking the minimum of the error function like a skier who always chooses the steepest downhill route with the aim of reaching the lowest point of a valley.

When the error  $e(n)$  falls below a pre-set threshold the execution of the algorithm terminates, and the weight values are taken as correct. Now we can use them to compute the impedance we are trying to measure.

From the block diagram in Figure 1 we have:

$$U_x = I_{R_{ref}} \times Z_x = U_{rm} / R_{ref} \times Z_x$$

where  $U_{rm} = U_r - U_x$ .

Now let  $U_{rm} = A \times \sin(\omega t)$  and  $U_x = A \times \sin(\omega t + \phi)$ , and we can write  $U_x = w_0 A \times \sin(\omega t) + w_1 A \times \cos(\omega t)$ .

Figure 3 shows the adaptive linear combiner.  $U_r$  (from 'line out') is modified using the weights  $w_0$  and  $w_1$  so that it is as close as possible to the desired signal  $U_x$ ;  $e(n)$  is the computed error.

The impedance of the device under test is then given by

$$Z = R_{ref} w_0 + j R_{ref} w_1.$$

**Software**

The program that generates the sine-wave signals, reads back the signals on the line-in input, and calculates the impedance of the device under test is written in Java.

The user interface shows the measured values as well as a display of the voltages across the reference resistor  $R_{ref}$  and the device under test  $Z_x$ . The LMS algorithm can be replaced by a different algorithm (see under the 'Settings' tab). The algorithm must be derived from the class Alg and implement the determine() method (see Figure 4). This allows a variety of different approaches to be compared: for example, a second approach, which uses a variable frequency, is described below. It would also be possible to implement the classic three-voltmeter method. The software architecture is illustrated in Figure 5.

**The sound card**

The sound card samples the incoming signals 44100 times per second, and we allow a choice of test frequency from 1000 Hz, 1250 Hz and 2200 Hz. The sound card must have a stereo

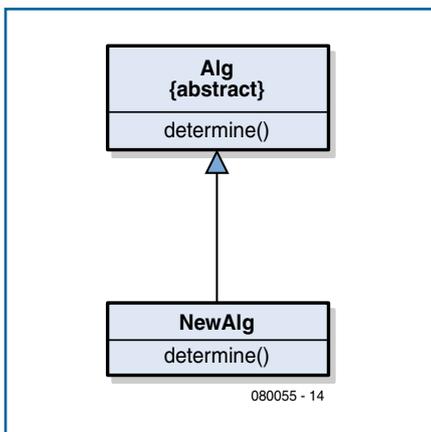


Figure 4. Deriving a new algorithm.

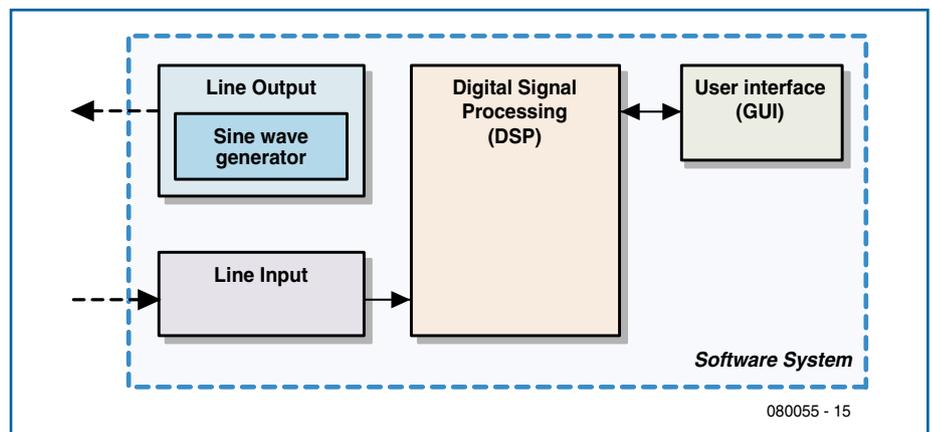


Figure 5. Block diagram of the software.

## About the authors

Martin Klaper studied Electronic Engineering at the Swiss Federal Institute of Technology in Zurich, graduating in 1977. He then worked in development for 20 years at Crypto AG. From 2000 to 2005 he lectured in computing and telecommunications at the Solothurn University of Applied Sciences, and currently he lectures at the School of Engineering and Architecture in Horw, near Lucerne. He is also a keen radio amateur (call sign HB9ARK) and is particularly interested in the ideas behind software-defined radio.



Heinz Mathis studied Electronic Engineering at the Swiss Federal Institute of Technology in Zurich, graduating in 1993. After several years working as a development engineer in industry at various companies in Switzerland and in Britain, in 1997 he returned to the Swiss Federal Institute of Technology to become a research assistant. He received his doctorate in the field of signal processing in 2001 and went on to work for u-blox AG developing GPS receivers. Since 2002 he has lectured on mobile communications at Rapperswil University of Applied Sciences. His main interests are in the fields of high-frequency engineering and digital signal processing for mobile radio and GPS applications.



input (watch out: some laptops only have a single input channel).

There are two important points to note: if  $U_i$  has a very high amplitude, it may distort. Although the actual amplitude of  $U_i$  does not enter into the final calculation, it must nevertheless be a pure sinewave. It is also possible to overdrive the input amplifiers on the sound card, which will also lead to distortion.

The program is able to display the input waveforms to allow the user to check for distortion. The relevant levels can be adjusted using the PC's audio mixer settings. For a more precise measurement of the degree of distortion, connect an oscilloscope (or, even better, an audio spectrum analyser). It is also essential to ensure that the balance between left and right channels is set exactly in the centre.

The software oscilloscope display gives a good demonstration of the time relationship between the voltage across a capacitor or inductor and the current flowing through it.

### Installation and operation

The RLC meter software can be downloaded from the *Elektor* website [3]. It is also necessary to have the Java runtime environment [2] installed. Then it is a simple matter of unpacking the software and running it. The Java program consists of three parts, (rlc.jar, swt.jar and swt-win32-3236.

dll) which must be kept in the same directory. The program is run by double-clicking on rlc.jar.

**Figure 6** shows the graphical user interface. Clicking on the relevant tabs

switches between a simple meter mode and an expert mode. In simple mode the program just displays the measurement result in large characters so that it can easily be read from

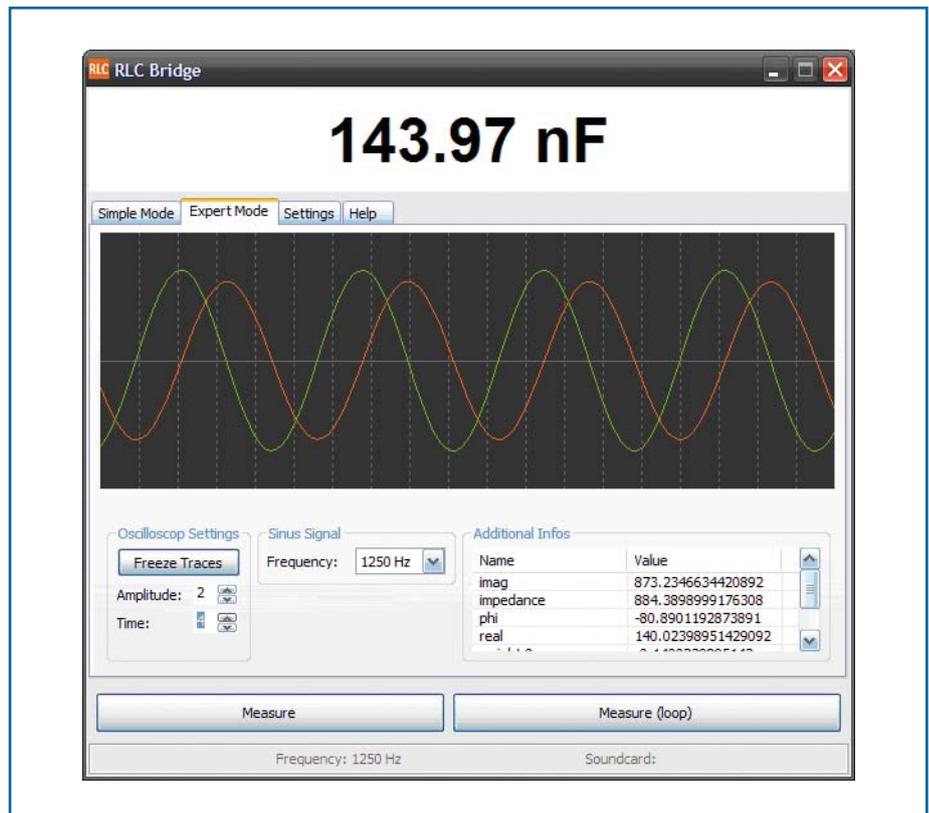


Figure 6. How the software displays its results on the screen.

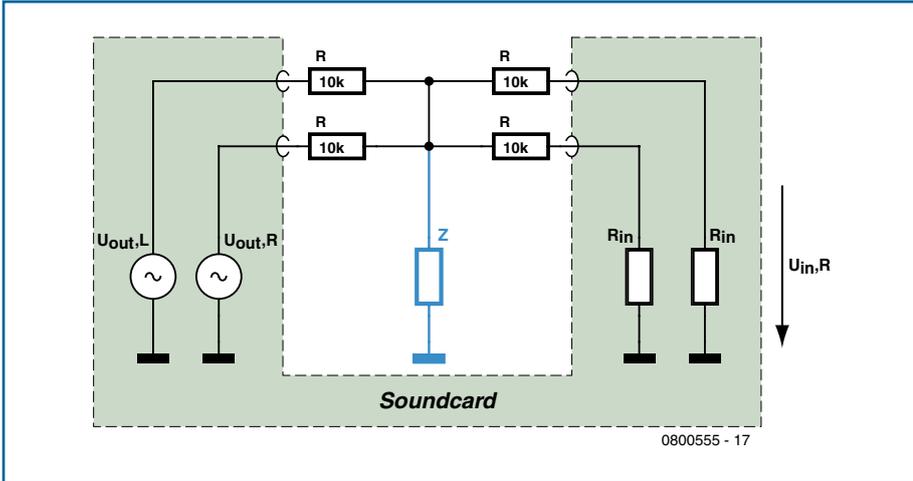


Figure 7. The hardware for method B consists of just four resistors.

voltage divider part from the above circuit using a series resistor R. If we construct a symmetrical T-network it has the advantage that the circuit is symmetric with respect to the inputs and outputs of the sound card. This method also has the advantage that it can be used with single-channel sound cards, although using two channels helps to reduce the noise in the measurement a little. We thus end up with the simple resistor network shown in **Figure 7**.

$R_{in}$  is the input impedance of the sound card, which can be determined from the datasheets of the devices used. The details of the derivation of the formulae for calculating the unknown impedance are available in a supplementary document that can be downloaded from the *Elektor* website.

The task for the software in the meter is now to measure the amplitude of the signal levels at the inputs to the sound card. The measurement is carried out at a range of different audio frequencies. If the unknown impedance is purely resistive the amplitude will exhibit no frequency dependence, whereas for an inductive or capacitive load the amplitude will increase or decrease (respectively) with increasing frequency.

For each of the three quantities R, L and C the software calculates a nominal value which minimises the squared error over the set of test frequencies used. The corresponding residuals (i.e., the normalised distances between the measured amplitudes and the theoretical amplitudes for the calculated component value) are also computed. The value which leads to the smallest residual is taken as the correct one and is displayed as the result, along with its corresponding unit.

(080055-1)

## References and Internet links

[1] Martin Klaper, 'ECG using a Sound Card', *Elektor* October 2006.

<http://www.elektor.com/magazines/2006/october/ecg-using-a-sound-card.58566.lynx>

[2] Java compiler and development environment (JRE and JDK): the Java Runtime Environment (JRE), version 5.0, is required to run the program, and the J2SE development kit (JDK), version 5.0, is required to modify and compile the program.

<http://java.sun.com/javase/downloads/index.jsp>

[3] Software download: [www.elektor.com](http://www.elektor.com); month of publication

Phase shift  $\varphi$ :

$$\varphi = \arctan\left(\frac{\omega_1}{\omega_0}\right)$$

Component characterisation:

$\varphi = 0$ : pure resistance  
 $\varphi < 0$ : capacitor  
 $\varphi > 0$ : inductor

Impedance Z:

$$Z = \sqrt{(\omega_0 \cdot R_{ref})^2 + (\omega_1 \cdot R_{ref})^2}$$

Real (R) and imaginary ( $X_L$  or  $X_C$ ) parts:

Resistor:  $R = Z$

Inductor:

$$L = \frac{X_L}{2\pi f}$$

Capacitor:

$$C = \frac{1}{2\pi f \cdot Z}$$

a distance. The series equivalent circuit is also shown graphically. In the case of a resistor the ohmic value is also displayed as the corresponding colour code.

It is possible to make single measurements or continuous measurements: the latter is particularly useful when tuning components such as coils.

In expert mode the display shows the input signals oscilloscope-fashion on the display, along with some interesting intermediate values computed by the LMS algorithm. Three different measurement frequencies are available. The 'Settings' menu allows the selection of a different sound card if more than one is present, and it is also possible to switch to a different measurement algorithm (if one has been implemented) for comparison purposes.

As mentioned above it is important to know the value of the reference resistor  $R_{ref}$  accurately, as the accuracy of the whole measurement depends on it. Ideally we should aim for an accuracy of 1% or better, and if possible it is worth measuring the value of the resistor before use with a precision bridge. It is also important to use a film resistor for  $R_{ref}$  rather than a wirewound type, as the latter will have significant inductance. Accurate measurements of reactance from  $0.01 R_{ref}$  to  $100 R_{ref}$  are possible. It is a good idea to arrange the circuit so that different resistors can easily be substituted for  $R_{ref}$  to extend the range of the instrument.

### Method B

An even simpler approach, suitable for measuring an unknown impedance Z using a sound card, is to use only the