A Zero-Cost, Real-Time, Windows Signal Laboratory

John Leis, Chris Snook
University of Southern Queensland, Toowoomba, Australia
Correspondence: leis@usq.edu.au

***Abstract**: This paper introduces a Windows-based signal capture, display, and waveform synthesis package called "Win-eLab". The software is able to run on a conventional desktop or laptop with no additional hardware, and can perform real-time Fourier analysis on audio-frequency signals. This paper is intended as an introduction to Win-eLab, aimed at motivating further use of it in both teaching and self-directed learning contexts. The use of the software to familiarize students with the concept of "laboratory" instrumentation is discussed, as well as the usefulness of a simultaneous time-domain/frequency-domain display for understanding signals, particularly in signal processing and communications systems courses. It is anticipated that applications may extend beyond electrical & electronic engineering – for example, as an aid to understanding mechanical vibrations, acoustics, and in other discipline areas.*

Keywords: laboratory work, computer instruments, educational software

Context and Motivation

Laboratory work is well entrenched in most, if not all, engineering curricula. This is ostensibly for two compelling reasons. The first is to give the student a practical grounding in the instruments, hardware or other apparatus which they might encounter in professional life. The second, perhaps more subtle reason, is to support the learning process itself: the theoretical work becomes grounded in practice; the practice, in turn, supports and motivates theory. This is not an isolated view: Edward concurred with Doughty's view that that 'practical experiments lie at the heart of the relationship between meaning and understanding', and that 'in engineering, practical experiments and projects lie at the heart of the relationship between meaning and understanding' 1,2. Somewhat paradoxically, Evans observed that 'many [engineering] students derive the least satisfaction and sense of achievement from the formal laboratory content' 3.

We set out to implement a software instrument that would enable students to grasp the principles of electronic instrumentation, before actual laboratory sessions with real equipment commenced. However, it soon became apparent that a self-contained, PC-based instrument would allow students the opportunity to do much more than simply prepare for laboratory sessions. It would allow students to experiment with "what-if" scenarios in signal analysis and communications systems, without the need for extensive initial grounding in theory. Once the theory is introduced, the displays may be interpreted in the light of the underlying theory. Such an approach has a sound basis in educational psychology. As discussed in Kolb 4, the

Lewinian Experiential Learning Model proceeds in exactly this circular fashion, from concrete experience to reflections, to abstract generalizations, and testing of the underlying theory. Thus, using the PC instrument presented herein, learning is reinforced by practical experimentation – but not in the environment of crowded, rushed laboratory sessions, where is little time for reflective learning.

Simulations and Virtual Instruments

The relationship between academic or theoretical treatment and practical or laboratory work is not always clear, particularly in the mind of the student. This is especially so in the distance-learning context, where theory and practice may be somewhat decoupled. We reasoned that, especially in the distance- or self-directed learning modes, reflective learning would be enhanced if students could access laboratory equipment in their own time and space. One often-used model for such reflective learning is the virtual laboratory (see, for example, 5). This learning paradigm certainly has its place, as do simulated or virtual instruments. In the distance education context, multimedia and Internet-based courseware also has widespread applicability 6. Our contribution is a PC instrument for self-directed learning which is not a simulation, but rather a fully-functional audio spectrum analyser and signal generator.

Win-eLab: A PC-Based Instrument

Motivated by a consideration of the above factors, and realizing that students today typically have powerful PC's with audio input/output capability, we set out to develop a PC-based instrument that could provide signal analysis in real-time. It was not to be a simulation in any way: the software *is* the instrument. The primary goal was to provide an method for students to familiarize themselves with the underlying operational concepts of the oscilloscope, spectrum analyzer and signal generator, prior to undertaking "real" laboratory sessions. However, during the course of the project, it became apparent that the software itself would be a powerful vehicle to enhance learning.

The software, dubbed "Win-eLab", is designed to loosely emulate the "look-and-feel" of conventional instrument control panels and displays, although emphasis is given to simplicity over extensive functionality (Figure 1). Further thoughts on the user-interface and operational design – which we consider to be a crucial aspect if the software is to be successful as a learning tool – are discussed throughout the paper.

Whilst the original motivation for the project was to prepare students for laboratory work (especially those from varying cultural backgrounds, who sometimes have inhibitions in working in typical group-based laboratory situations), it became apparent during the development process that the concept of a simultaneous time-frequency display would have significant pedagogical advantages. This is illustrated in Figure 2, where a synthesized waveform is shown on the time-domain display (left), with the corresponding frequency analysis (right).

Capabilities

The following sections describe the salient features of the program, and Figures 1-5 illustrate various waveforms. This section is not intended to be an exhaustive discussion of capabilities and features, merely an introduction – the interested reader is referred to the online tutorial at http://www.usq.edu.au/users/leis/software/

Waveform Display

The sampled waveform is displayed on the left-hand panel (Figure 1). The display parameters such as sweep per graticule division, vertical gain, and DC offset may be set in a manner similar to a conventional instrument. The trigger is simply a zero-crossing detector – this is adequate for the purpose, and more importantly does not overcomplicate the display for new users. Note that the amplitude scaling is necessarily relative, and not calibrated as such. This is one aspect which should be borne in mind if using Win-eLab as an instructional tool.

Spectral Analysis Display

The amplitude spectrum of the sampled waveform is displayed on the right-hand panel. Figures 1-3 show sampled audio, sinusoidal and sawtooth waveforms, respectively. The frequency components are calculated using a Fast Fourier Transform (FFT) and displayed with a fixed update time. The update time was chosen so as to give fast frequency display whilst not overburdening the CPU. The display parameters such as span per graticule division, vertical gain, and start frequency may be set in a manner similar to a conventional instrument. The start frequency may be from the left-hand edge, or the center, of the display panel. Linear vertical scaling is the default, although logarithmic scaling may also be selected.

Waveform Synthesis

The lower control panel interfaces to the signal synthesis module. Standard waveforms able to be generated are sine, random, square, triangle, and sawtooth (ramp). The frequency and amplitude may be set for each. In addition, amplitude modulated and frequency modulated waveforms may be selected. In either case, the modulation parameters are then able to be set. Note that in this figure, the “loopback” mode of operation is selected: the output buffer is simply copied to the input buffer in this mode.

The AM and FM waveforms require additional parameters to be specified. When these are selected, two additional controls ‘fm’ and ‘m’ become visible, these being interpreted as the modulation frequency and modulation index, respectively. In either case, the main frequency control becomes the carrier frequency.

Representative AM and FM waveforms are shown in Figures 4 and 5, respectively.

Similar Products

A number of products are available to perform broadly similar tasks to what Win-eLab is capable of. Our subjective opinion on them, considered as a whole and when applied to the educational context, is briefly summarised as follows.

Cost
Although commercial products are available at relatively modest cost, it is felt that students are more likely to take advantage of the opportunity if the download is freely available.

Installation
Some of the products trialled had lengthy download and installation times, and may require specialist expertise during installation. Whilst this may not be an issue per-se, the issue of technical support becomes fraught when dealing with large student bodies, and/or campus laboratories where various security restrictions are in place. Win-eLab is contained in a single small (40kbyte) executable file, and requires no installation other than uncompressing (unzipping) a single file.

Dual-Mode Capability
Win-eLab was designed from the outset to include both time and frequency displays, with a signal generator added during development. Many of the products surveyed provided only time display, or only frequency, with signal generation usually not provided. Whilst these were probably appropriate for their intended application, our application domain benefits considerably from having all the above capabilities in one place.

Platform
Win-eLab is available for Windows platforms. There is at least one Linux oscilloscope package available, but considering that a great many students have Windows as their default operating system, it was felt that this was a more appropriate target. Furthermore, the plethora of Linux distributions meant that installation and configuration issues may have presented themselves.

Dual display capability
Win-eLab was designed from the outset to include both time and frequency displays simultaneously. This was considered a fundamental feature, for pedagogical reasons. Separate displays would mean the student would have to be comfortable operating in either domain, and switch between he two as needed. More importantly, we wanted to reinforce the "one-signal, two-view" paradigm. A single, side-by-side display serves this goal admirably.

Wider Benefits

Computer-based instruction has become widespread, closely following the widespread availability of the Internet (and, more recently, broadband access). In some respects, this is often simply a mirror of paper-based education and lecture-style instruction: the learning is very much a one-way process. Software such as that presented here allows the student the opportunity for interaction, and by implication this aids in forming their own mind-view of the conceptual problem. Furthermore, students are given the opportunity to "play" with new software, and learning is not quite the chore it might be with more traditional approaches.

Apart from the motivational aspect – which is difficult to gauge – reflection on the possible benefits of Win-eLab include:

1. Better utilization of limited laboratory time, by preparing students beforehand.
2. Emphasis of the commonly-used but difficult-to-grasp concept of the "frequency domain", which is central to signal and communications engineering**.**
3. Nil cost and wide the availability make it attractive to all students, but especially those in developing countries where resourcing is a problem 7, 8.

Design Aspects

As mentioned, the program is capable of both sampling audio and generating (synthesizing) predetermined waveforms. The sampling rate is 22kHz; this being a compromise between temporal quantization and processor loading. The sample buffers are chosen as a compromise between acceptable screen update rates and processor loading; 4k-sample buffers are used in the present implementation, with 16 bit per sample resolution. Although the student need not be aware of these parameters, it should be borne in mind by the instructor that this is a necessary limitation of all digital sampling systems. Indeed, once students are familiar with the package, the inherent limitations of digital sampling may be introduced in the context of everyday devices such as portable MP3 players and mobile phones.

The program, written in C using standard Windows Application Programming Interface (API) system calls, occupies less than 40kbytes when compiled into a single stand-alone executable. It is compiled using a Windows port of the well-established ‘lcc’ compiler available from Virginia Tech 9, 10.

A deliberate decision was made to have only one user-interface screen. There are no hidden configuration or setup screens. More importantly, both the measured time waveform and corresponding frequency components are displayed simultaneously, in real-time.

The application is able to be used with relatively modest computer facilities. On a 2GHz Pentium, the CPU usage of Win-eLab is typically less than 40-50%, with around 2Mbytes of physical memory used. This allows for plenty of overhead for the real-time sampling and display. On Windows XP, a CPU usage graph is available, although since the kernel-level sampling services for processor utilization are not available on Windows 2000. The reason for including a realtime CPU usage estimator was to facilitate error reporting on older systems with lower clock rates, which may not be able to cope with the computational demands.

Availability, Hardware Requirements, Installation and Use

Although CPU clock speeds have increased significantly in recent times, the computational burden of the program is considerable. In order to function effectively without introducing screen update flicker or perceptible gaps in the audio waveform generation, a number of code optimization and synchronization techniques are employed. The processing requirements include:

1. Sampling the input waveform.
2. Calculating the Fourier transform for spectral analysis.
3. Generating the desired waveform without any perceptible gaps.
4. Seamlessly updating the display whilst avoiding flicker.
5. Maintaining an acceptable user-response time.

A number of double-buffering techniques are employed, including pixel-mapped memory update buffers for both time and frequency displays. Double-buffering is also used to ensure seamless audio output from the signal synthesizer.

The executable code is available from the website http://www.usq.edu.au/users/leis/software/

It is distributed as a stand-alone zip archive of approximately 15kbytes. It does not need to be installed in any particular directory, does not write any local files nor modify any system registry entries. Thus, it is suitable for networked installation or even temporary installation (for example, from a USB drive). The audio hardware is automatically detected.

Combined, these features mean that there should be little likelihood of installation failures. A tutorial explaining the major features using screen snapshots is also available from the download site.

Summary & Conclusions

This paper has introduced a free, Windows-based instrument which is able to generate and display waveforms in real-time. The control system is designed to be a simplified version of that found in real instruments. The salient features of the instrument software were discussed, with a view to motivating potential applications in engineering education and elsewhere.

It is envisaged that the engineering education community will be able to make use of this "software instrument", both for introducing the concepts of signals, systems, time and frequency domain, and for introducing the specifics of test equipment.

Acknowledgements

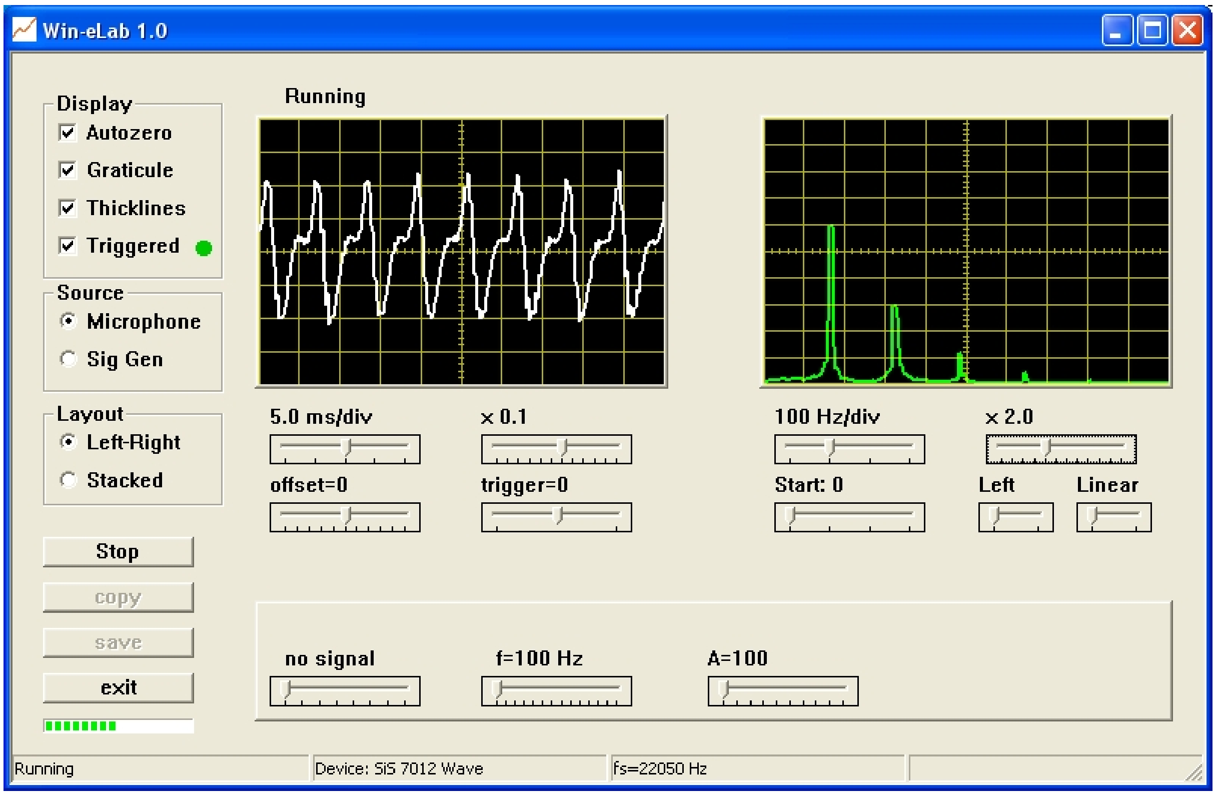
The FFT routines used in the work were adapted from earlier work by E. O. Brigham, Jim Pisano and Mike Stump. The concept for a side-by-side time-frequency display originated from Bruce Varnes.

References

1. G. Doughty *et al*, 'Experimentation: how it reinforces self-learning', Innovative Methods in Enging Educ., 43 (1992), pp264-269
2. Norrie S. Edward, 'The role of laboratory work in engineering education: student and staff perceptions', Int. J. of Elect. Enging. Educ., 39 (2002), 1, pp 11-19
3. Alun Evans, Timothy Davies & Stephen Wilks, 'Is your laboratory a turn-off?', Int. J. of Elect. Enging. Educ., 39 (2002), 3, pp 284-291
4. David A. Kolb, `Experiential learning – experience as the source of learning and development', Prentice-Hall, 1984
5. Tim Roberts, 'The virtual machines laboratory', in Proc. Australasian Assoc. for Enging. Educ., 2004, http://www.aaee.com.au/journal/2004/roberts04.pdf
6. Marco Chirico, Giancarlo Parodi & Anna Marina Scapolla, 'A generalisable experience of distance education in electronics', Int. J. of Elect. Enging. Educ., 39 (2002), 1, pp 1-10
7. Oyeshola F. Kofoworola, 'Engineering education in Nigeria: present learning systems and challenges for the future', in Proc. Australasian Assoc. for Enging. Educ., 2003, http://www.aaee.com.au/journal/2003/kofoworola03.pdf
8. J. D. C. Joseph and K. S. Julien, 'Laboratory exercise and programme design', Int. J. of Elect. Enging. Educ., 37 (2000), 4, pp 316-332
9. Chris Fraser & David Hanson. 'The *lcc* compiler', http://www.cs.princeton.edu/software/lcc/
10. Jacob Navia, '*lcc* for Windows', http://www.cs.virginia.edu/~lcc-win32/

FIGURE CAPTIONS FOLLOW.
FIGURES HERE ARE JPEG FOR REFERENCE ONLY.
FIGURES PROVIDED AS SEPARATE TIF FILES.

fig1.jpg



Display
[✓] Autozero
[✓] Graticule
[✓] Thicklines
[✓] Triggered ●

Source
[✓ Microphone] Sig Gen

Layout
[✓ Left-Right] Stacked

5.0 ms/div
× 0.1
offset=0
trigger=0
100 Hz/div
× 2.0
Start: 0
Left
Linear
no signal
f=100 Hz
A=100

Running | Device: SiS 7012 Wave | fs=22050 Hz

Figure 1: The main – and only – screen of *Win-eLab*. Both the time display (left) and spectral analysis display (right) are visible, with the signal generator control panel below. Controls are grouped according to their applicability; for example, the timebase is applicable to the time display (on the left), and appears directly below it.

fig2.jpg

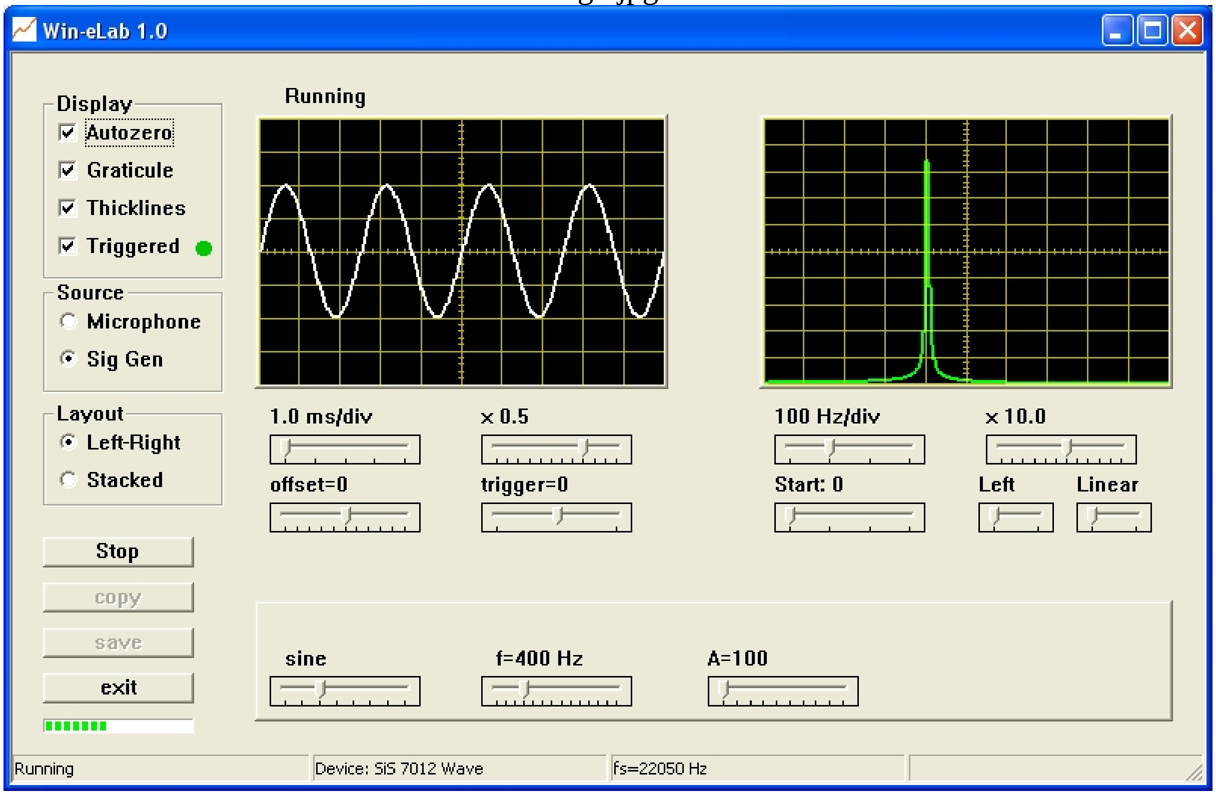
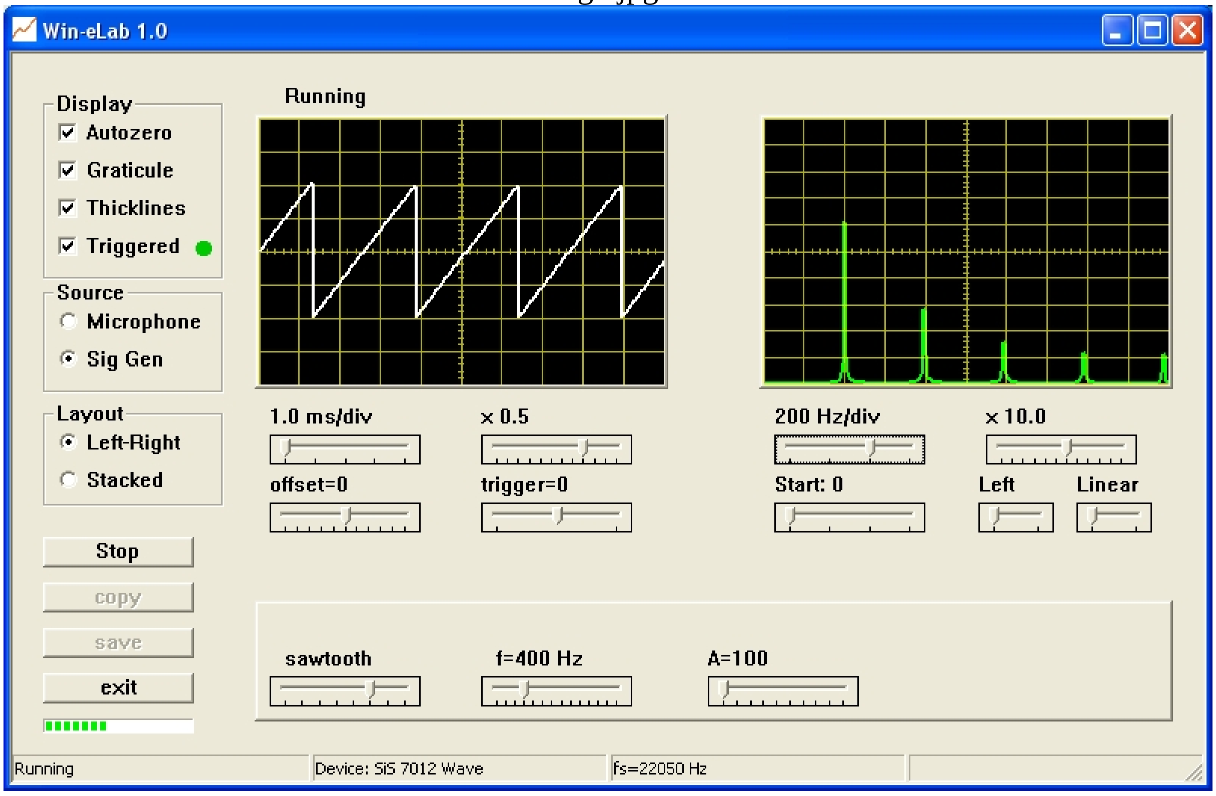


Figure 2: A simple sine wave generated by the frequency synthesizer (lower panel), and shown in the time and frequency domains. The 400Hz waveform is audible through the PC's speakers. It is quite easy to change to other wave types or to change the amplitude & frequency.

fig3.jpg



Display
- [✓] Autozero
- [✓] Graticule
- [✓] Thicklines
- [✓] Triggered ●

Source
- ○ Microphone
- [✓ Sig Gen]

Layout
- [✓ Left-Right]
- ○ Stacked

1.0 ms/div
× 0.5
offset=0
trigger=0

200 Hz/div
× 10.0
Start: 0
Left
Linear

sawtooth
f=400 Hz
A=100

Running | Device: SiS 7012 Wave | fs=22050 Hz

Figure 3: A sawtooth waveform generated by the internal synthesizer, and its time-domain display (left). The harmonics present are clearly shown in the spectrum display on the right. The waveform is audible through the PC's speakers.

fig4.jpg

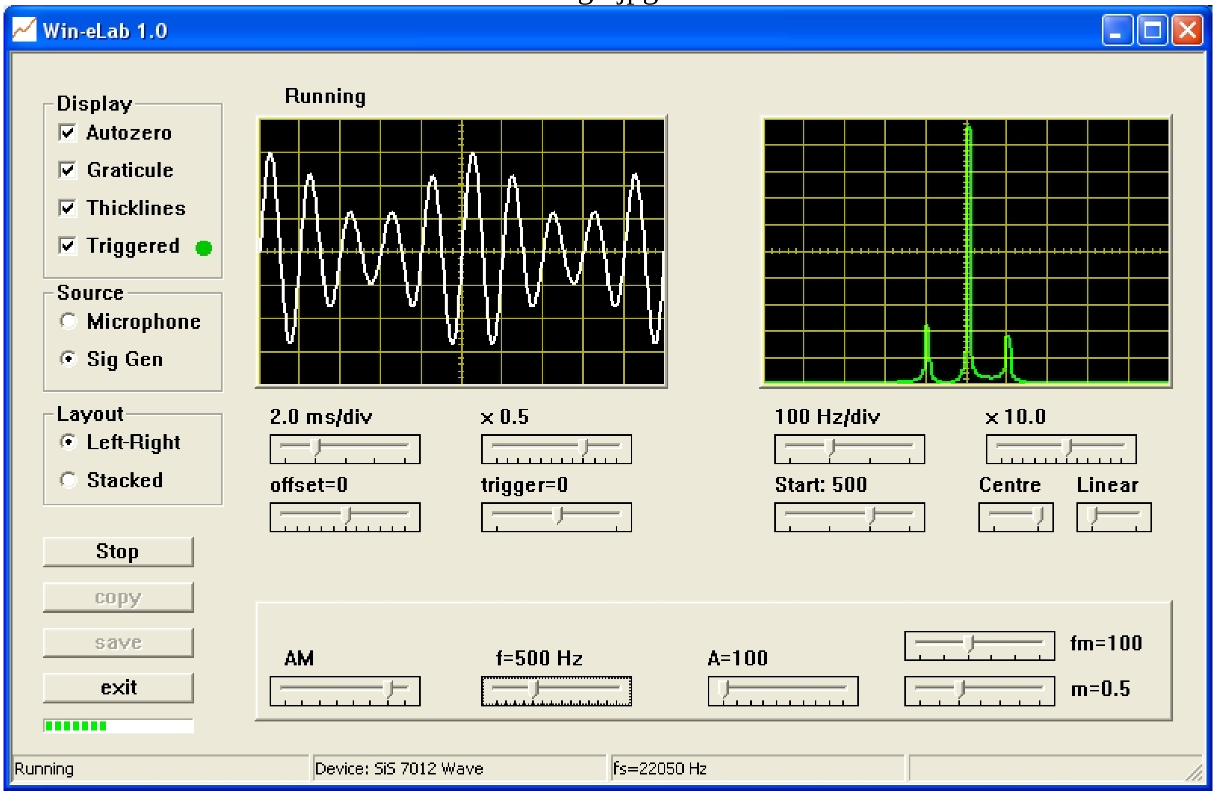


Figure 4: An amplitude modulated (AM) waveform, as generated by the internal synthesizer. The modulation parameters are easily controlled, and the classic "envelope" is visible in the time display. The frequency display shows the carrier and sidebands present. All displays are updated in real-time. A significant advantage of this is that the parameters may be adjusted dynamically and with instantaneous display changes.

fig5.jpg

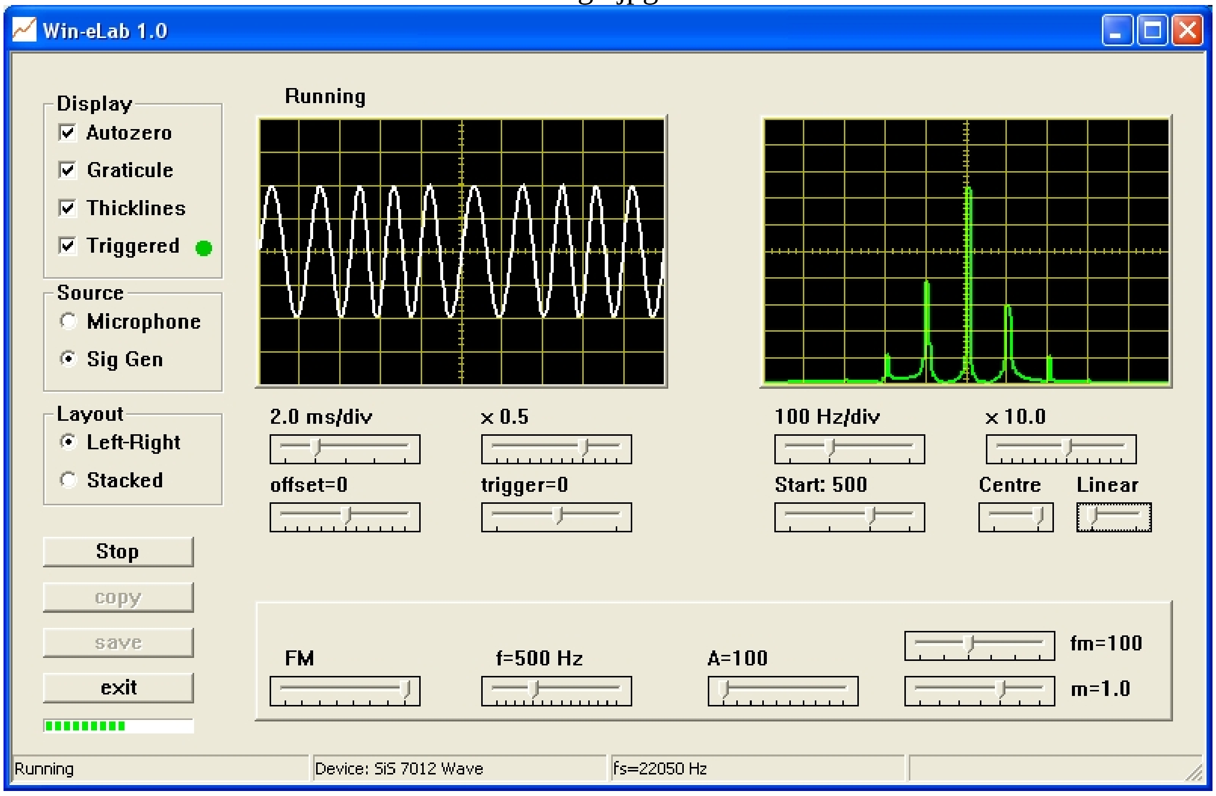


Figure 5: A frequency modulated (FM) waveform, as generated by the internal synthesizer. Again, the modulation parameters are easily controlled, and the classic sideband pattern is visible in the frequency display. Since the parameters may be adjusted dynamically, with instantaneous display changes, the effect of the carrier, modulation frequency, and modulation index is easily demonstrated.