SAN2PC: A Spectrum Analyzer to PC Interface

How to connect your old analyzer to a PC with a simple, low cost, interface and plot or print spectra.

Roland Cordesses, F2DC

Introduction

The Spectrum Analyzer (SA) is an incredibly useful tool to the homebrewer. Unfortunately, it is very expensive when purchased new, so the SAs found in the workshops of many hams are of the secondhand type, or even home built. As a result, they often lack functionalities available on more modern equipment. For instance:

• It is not possible to store the displayed spectrum in a file for later computer processing.

• Getting a hard copy of the displayed spectrum is not an easy job: dedicated cameras are scarce and expensive, and X-Y plotters are cumbersome.

In order to solve these two points, I have recently designed and built a simple interface to connect my spectrum analyzer to a PC. This unit, named SAN2PC, digitizes analog spectrum data, then processes and transmits that data to the computer via the RS232 line. SAN2PC automatically takes care of the spectrum analyzer scan time. During initial calibration, relevant information is stored in the PIC EEPROM. The heart of SAN2PC is a PIC 18F2525 processor, running embedded software written in C language.[1]

Any terminal program can be used to receive and store RS232 data in a file. This file can be processed afterward, or simply read by any spreadsheet program: The spectrum can then be displayed on the computer screen and a copy printed if needed.

SAN2PC has been tested on my HP141T analyzer, but I think it could be used with other models and brands, insofar as they have

[1]Notes appear on page 29.

26 rue du Montant
63540 Romagnat, France
Roland.cordesses@free.fr

analog spectrum voltage and pen lift signal output connectors. Limited tests have been conducted on a Tektronix 2755P.

I have also written dedicated software (in the *Python* language) displaying not only the spectrum, but also the spectrum analyzer's settings (center frequency, scan width, bandwidth, and so on) on the PC screen. This picture can be stored in a file for future use. The program runs under *Windows XP* and has been tested under *Linux* (the *Python* software I developed is not limited to the *Windows* world).

Presentation

Most SAs of the kind we are talking about have the following auxiliary connectors, usually intended for connection to an X-Y plotter:

• Vertical Output (VO) is a voltage corresponding to the spectral power for each frequency step, and is similar to the vertical cathode ray tube (CRT) deflection voltage. VO can be positive or negative, according to the SA model, and generally has an amplitude on the order of a volt.
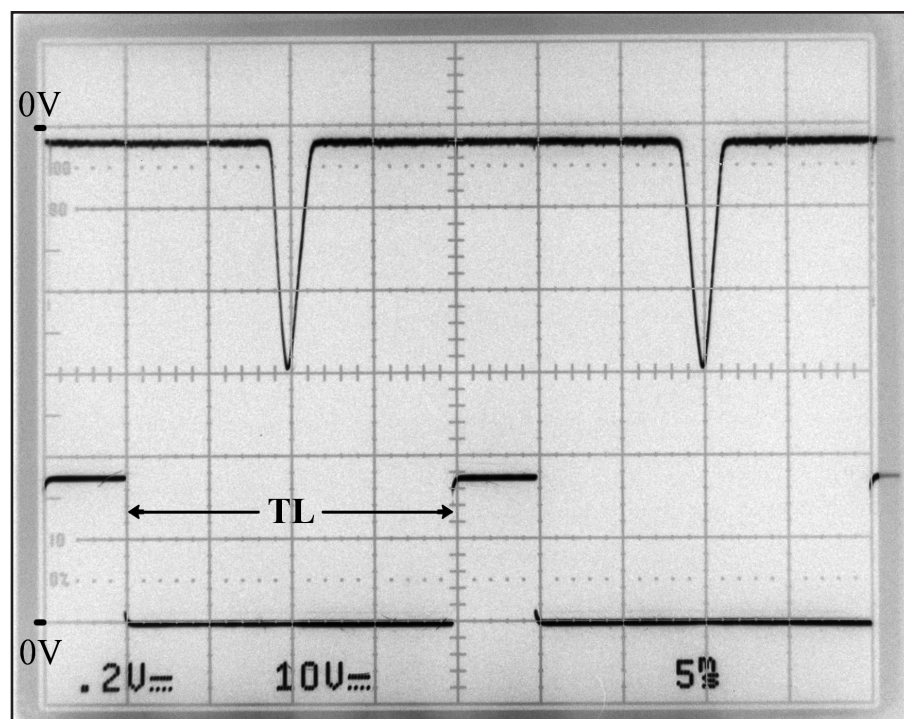

Figure 1 — VO and PL signals, as displayed on an oscilloscope screen.

• Pen Lift output (PL) is a logic level (0-20 V or TTL) to lift the X-Y plotter pen up or down. Generally, a PL low level lifts the pen down, thus meaning that spectrum voltage VO is valid.

• Scan Output (SO) is a sawtooth waveform signal similar to the one used for horizontal CRT deflection. Its polarity and amplitude are different from one model to another.

At the very beginning of the project, I thought of using the three outputs, but it quickly turned out that only the first two were necessary: the idea is to digitize VO (around 1000 samples per spectrum, value not critical) as soon as PL is going down. Figure 1 shows the two signals generated by my HP141T, VO being the upper trace and PL the lower.

The PIC first measures the time, TL, during which PL is low, computes the sampling period, TS, waits for the next falling edge of PL, and then starts 1030 analog to digital (A/D) conversions. Thanks to this approach, the sampling period, TS, is automatically modified each time the spectrum analyzer scan time is changed. The drawback is that it takes twice the TL time to complete a full spectrum measurement.

Have a look at the SAN2PC block diagram presented on Figure 2:

• The PIC 18F2525 is pin-compatible with the 16F873, but features, among other improvements, a larger user RAM (nearly 4 kbytes) allowing storage of the full spectrum data without any problem. Its internal A/D converter offers a 10-bit resolution and a conversion time short enough for this application. Moreover, A/D conversions can be triggered by an internal 16 bit counter/timer. The input A/D voltage range is 0 to 5 V full scale.

• The purpose of the level converter boxes is to transform PL and VO to voltage levels PL1 and VO1, accepted by the PIC:

• If PL is a TTL signal, the level converter is omitted, and if it is a 0-18 V signal, a simple resistive bridge will do the job (that is the HP141T case). Just remember that this bridge must be compatible with the load accepted by the SA Pen Lift output: Check your equipment operation manual.

• When it comes to VO, there are several choices for polarity and amplitude.

The HP141T VO swings from 0 to –800 mV for the full 80 dB RF input range of the SA: An operational amplifier in inverting mode provides the 18F2525 A/D converter with the required positive voltage.

On some other SAs, VO can go from 0 to +800 mV or more, depending on models and brands: A non-inverting amplifier must then be used.

The amplifier gain is adjustable to allow fine calibration of the interface.

• A 20 MHz crystal is connected between the oscillator pins of the PIC. A MAX232 driver/receiver provides the level shifting for the RS232 link to the PC.

• SAN2PC is powered by any standard wall power adapter (8 V dc or more) thanks to on-board regulators and dc/dc converter. The required current is less than 100 mA.

Circuit Description

The schematic diagram presented in Figure 3 shows that the circuitry is very simple.

• The operational amplifier is a classic dual one (TL82, TS272 or similar op amp). One can note that the second half is not used: It is just a spare part for a future project!

The Printed Circuit Board (PCB) is designed in such a way that U4A may be wired as an inverting or a noninverting amplifier: delete R12 and connect VO to J1 in the first case, or ground J1 and properly choose R12 and R10, VO being connected to J14 in the second case. If the VO voltage from your spectrum analyzer is different from that of the HP141T, don't forget to modify the resistors around U4A.

• R3 and R11 are the voltage divider resistors going from PL to the RB0/INT pin of the PIC. You may adapt their values in accordance with the PL voltage of your analyzer.

• Analog voltage on pin AN0 (A/D converter input) must be kept inside a –0.3 V to +5.3 V range: Schottky diode D7 deals with the negative side and we supply U4 with ± 6.5 V to limit its positive output voltage around 5 V. In case of a positive only VO, the dc/dc converter, U2-U3 regulators and D7

could be omitted, and a single supply (5 V) op amp of the TS272 variety could be used instead of a TL82.

• Two RS232 speeds are available: 9600 and 115200 Bauds. This speed is selected by a rear panel switch.

• RC2 pin is wired to a front panel Normal / Calibrate switch.

• Two front panel LEDs keep the user informed of the SAN2PC status: waiting during TL measurement or sampling spectrum during data acquisition.

• The Start button is connected to the PIC MCLR reset pin.

The PCB file can be downloaded from my Web site at **roland.cordesses.free.fr**. Download the file **san2pcbd.pdf**.[2]

Figure 4 shows the inside of the unit and Figure 5 is a view of the interface standing on the top of the HP141T.

How It Works

Normal Mode

When the PL1 falling edge is detected on the INT pin of the 18F2525 (see Figure 1), at the beginning of the frequency scan, the software enters a procedure to measure the duration of the low level until the following PL1 rising edge appears. This time TL, related to the scan time setting of the SA, is stored as a 32 bit integer. The sampling period, TS, is then computed:
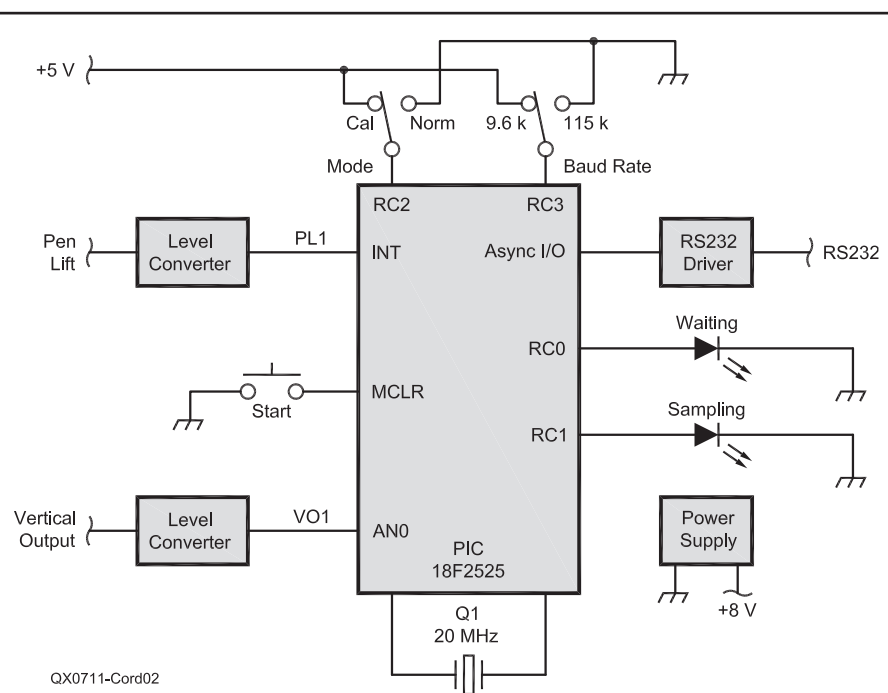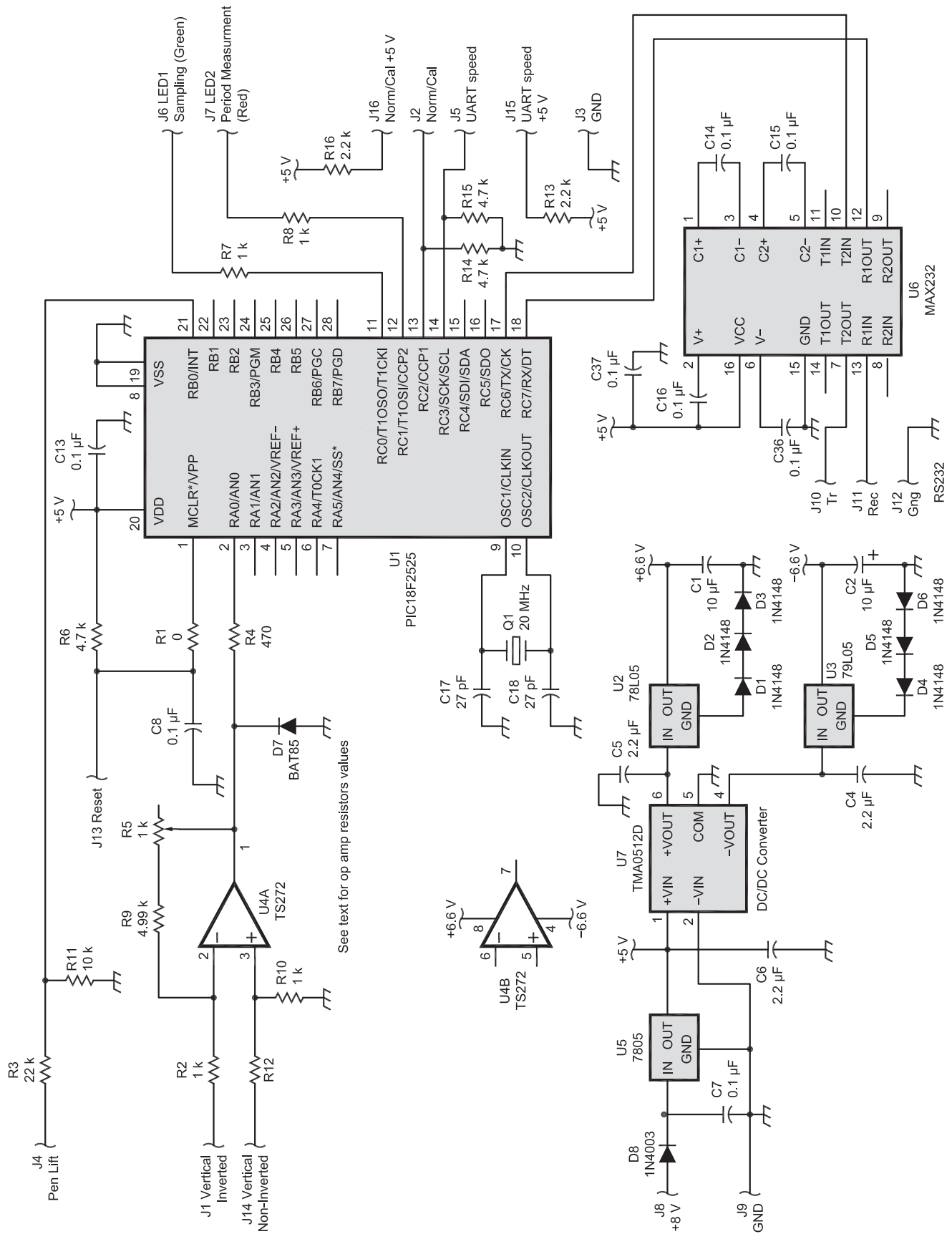
$$TS = TL / \textbf{sizemax} \quad (1)$$

Figure 2 — This block diagram of the SAN2PC interface shows the basic circuit operation.

Figure 3 — The SAN2PC schematic diagram.

where **sizemax** = 1030 is the number of samples per spectrum (as said earlier, this value is not critical).

The program then waits for the next PL1 high to low transition, and enables the interrupts. It then enters a loop during which **sizemax** values of VO1 are sampled every TS by the A/D converter and stored in the PIC RAM as an array called **specarray**. The interrupts are then disabled, **specarray** samples are read from indexes **ileft** to **iright** (see calibration mode), processed, converted to dBm and sent to the PC via the UART.

We use the Compare and Reset function offered by the **Capture/Compare/PWM (CCP)** module of the 18F2525 to start A/D conversions. In **Compare** mode, the 16 bit CCP2 register is constantly compared against the **Timer1** register value. This latter register is incremented on every internal instruction cycle, TI (Fclock/4), where Fclock is the PIC crystal frequency. When a match occurs, and if the A/D converter has been enabled, a conversion is triggered and the result is stored in **specarray**. **Timer1** is then reset and the process goes on until the last sample has been stored in **specarray**.

CCP2 is loaded with a 16 bit integer, **NCCP2,** corresponding to the TS period, with **NCCP2** being computed from TL and TI. As TL ranges from a few tens of miliseconds up to 100 seconds, according to the analyzer scan time setting, **NCCP2** is automatically updated each time the scan time is changed. Moreover, TI is possibly prescaled to prevent CCP2 overflow, when long scan times are selected.

Before transmitting **specarray** data to the PC, the program sends a header describing the analyzer main settings. Because these settings can't be automatically transmitted by the SA to SAN2PC, the program initiates a dialog with the user before spectrum acquisition. This dialog, controlled by the PIC embedded software, begins just after the push of the start button. The user is asked to enter the Reference Level (dBm), Center Frequency (with units), Scan Width/division (with units), Filter Bandwidth and Video Filter Bandwidth (with units). This information, packed as an ASCII string (the header), will be transmitted as the file data first line, just before spectrum data are sent by SAN2PC to the PC.

The file format, header included, is shown below:

980M 5M 300k 10k *
955.0000 –93.3
955.0506 –93.7

_____ ____

_____ ____
1005.0000 –93.2
*

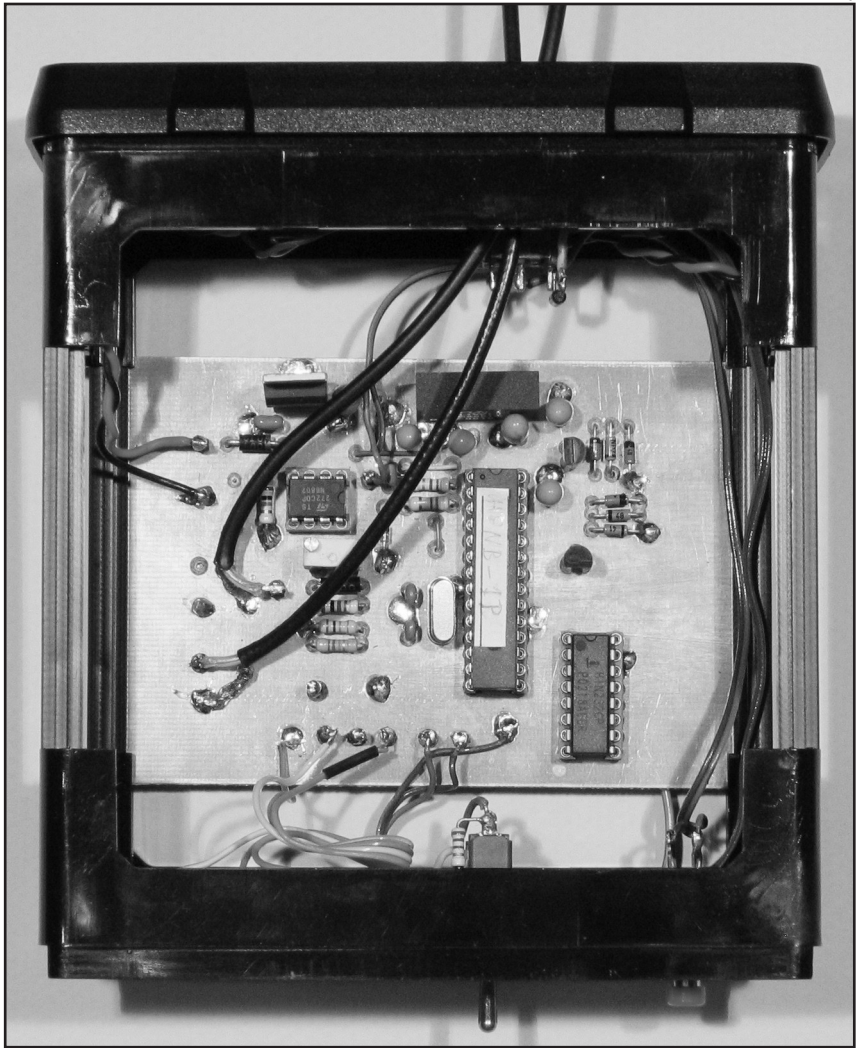The asterisk following the video bandwidth (10k) performs as the "start of data"

Figure 4 — This photo shows a look inside the SAN2PC interface.

Figure 5 — The SAN2PC is a small package that easily sits on top of the HP141T spectrum analyzer.

delimiter. It is followed by the first frequency, then the first RF power level and so on, until the second asterisk (end of data) is found.

The header information will be used by the *Python* plotting program, to display the SA settings in the upper part of the spectrum picture, during the plotting phase.

Calibrate Mode

Upon first power-up, SAN2PC will read the "default" calibration information from the PIC EEPROM: You will probably want to calibrate it more accurately, as these values may not be well suited to your SA. There are two calibration procedures, the first one dealing with amplitude and the second with frequency. The PIC embedded software takes care of these two tasks when Calibrate is selected by the Mode switch.

• Amplitude: U4A gain must be trimmed until the number of A/D points corresponding to the RF reference level, (the upper graticule line of the SA screen) is 800. This step may be performed with the SA calibrator used as an RF source, the scan width control on zero (no scan) and the input attenuator adjusted to put the trace just on the upper graticule. With 800 points related to the 80 dB SA input range, the amplitude resolution is 0.1 dB.

• Frequency: We said earlier that **sizemax** samples are written in **specarray** during *Normal* mode operation. Figure 6 (not to scale) shows that the PL falling edge appears earlier than the time the analyzer CRT spot crosses over the left graticule: it is thus necessary to know this delay, **t1,** in order to get a good frequency accuracy. Index **ileft** is a measure of this delay, and **iright** is representative of the delay between the time the spot crosses over the last right graticule and the rising edge of PL. During transmission to the PC, the only **specarray** values included in **ileft** to **iright** are sent.

The frequency calibration is very simple: you connect an RF source to the SA input (the calibrator for instance) and are asked to put the displayed "peak" (using the SA frequency control) on the first left graticule then to press Enter. Then you put the "peak" on the last right analyzer graticule and press Enter again. The software calculates the RAM indexes **ileft** and **iright** of the two maxima and stores them in EEPROM: **ileft** and **iright** will be the values used by the **specarray** reading procedure during Normal mode operation, when sending data to the PC.

Software

The PIC Embedded Software

The PIC program, named **san2pc.c,** is written in C language for easier development and debugging. For some years I have developed projects using the Custom Computer Services (CCS) C-Compiler.[3] This compiler

is easy to use and a lot of useful tools as well as sample programs are available as part of the package.[4] I would also like to thank the CCS French representative, Hi Tech Tools, for their kindness.[5]

Programming the PIC was done with a home made programmer inspired by the ProPic2 and using the WinPic800 software.[6,7]

In addition, the CCS bootloader was of great help during program development, speeding up debugging.

The HEX file (**san2pc.hex**) can be downloaded from the Spectrum Analyzer Interface directory of my Web site or from the *QEX* Web site as part of the **11x07_Cordesses.zip** file. See Note 2.
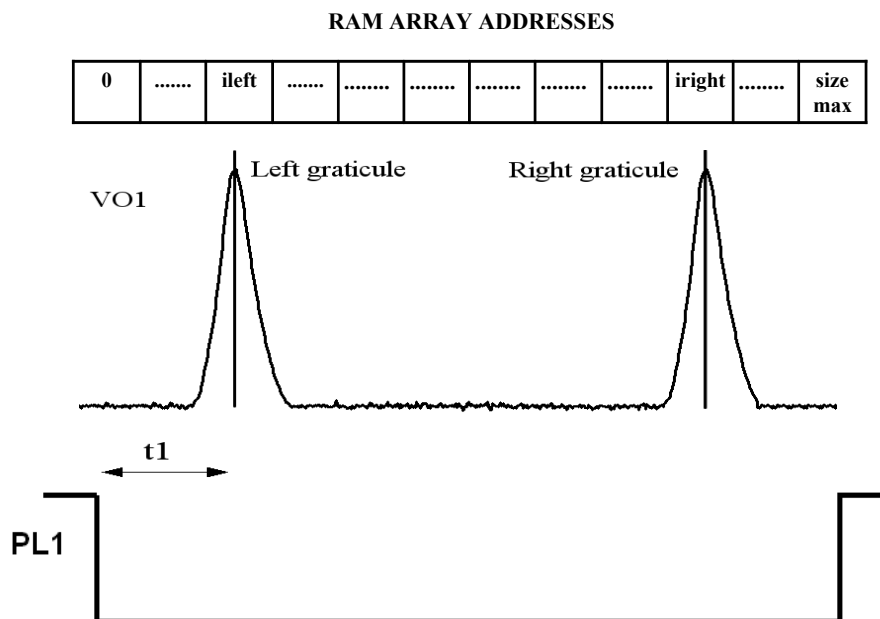

Figure 6 — This diagram illustrates the concept of the SAN2PC frequency calibration technique.
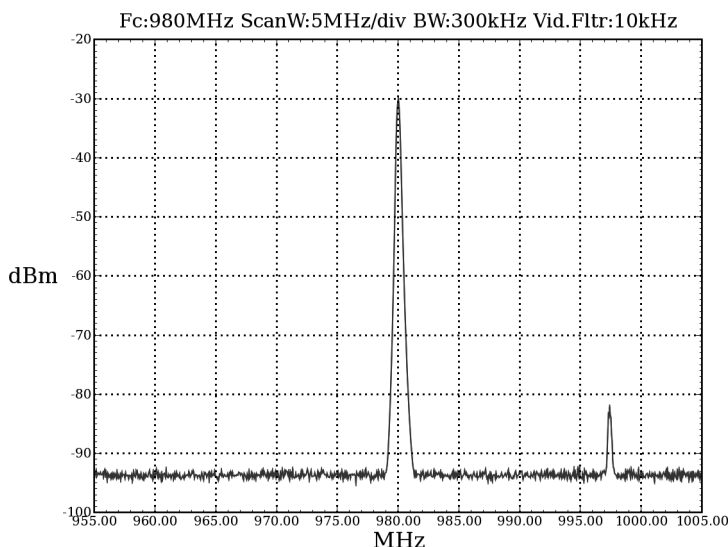

Figure 7 — This spectrum sample display is the result of the author's *Python* plotting program, drawsan.py.

The PC Plotting Software

I have also developed a dedicated program running on my PC under *Windows XP* (and successfully tested under *Gentoo Linux*) in order to plot, on the computer screen, not only the spectrum, but also the analyzer settings (the same way they are displayed on a modern SA screen). This program, named **drawsan.py**, relies on the header of the stored file.[8] I chose to write it in *Python* language, mainly because of the availability of a first class graphics library, Matplotlib.[9, 10] This library emulates Matlab™ commands (but does not require Matlab) and creates very nice plots. The **drawsan** user has the opportunity to save the picture as a png, bmp or vector file. A sample picture of a spectrum plotted with **drawsan.py** is shown in Figure 7.

Please, be lenient with **drawsan.py**: It is my first *Python* program and must be seen as a beginner's exercise!

Operating SAN2PC

It is very easy to use this interface: connect PL and VO lines to the corresponding SA connectors, link SAN2PC to your computer through the RS232 line and run a terminal program. Then push the Start button and answer the questions described earlier.

Please note that SAN2PC amplitude and frequency resolutions are better than the accuracy of most spectrum analyzers of this kind. Keep that in mind when interpreting data files or pictures!

Conclusion

This simple interface may help to solve the hard-copy problem associated with many secondhand spectrum analyzers. It also provides the user with a file for subsequent computer processing. It could probably also be used with other equipment, such as network analyzers, that don't have any printing function.

Acknowledgments

I would like to thank my two sons, Lionel and Joël, who strongly suggested that I use *Python* in order to get nice looking spectral displays. Although the learning curve was sometimes a rough one, I don't regret this contact with a new programming language.

Roland Cordesses, F2DC, was first licensed in 1962, and he has been a member of ARRL since 1964. He graduated as an Electronics Engineer and worked as a Research Engineer in an Observatory devoted to atmospheric and earth sciences.

For nearly 30 years, he worked in the design and development of radar systems for remote sensing of the atmosphere and for monitoring of volcanic eruptions. During his career, he has presented or published many papers related to electronics and geophysics.

Recently retired, he continues to be an avid homebrewer: over the years he has designed and built many receivers, transceivers and measuring-equipment projects.

In addition to Amateur Radio, he enjoys building and flying model aircraft (homemade receivers and transmitters, of course) and hiking in the mountains.

Notes
[1]See **www.microchip.com**.
[2]The circuit board pattern is also available for download from the *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **11x07_Cordesses.zip**.
[3]PIC MCU C Compiler Reference Manual, CCS, Inc, October 2005.
[4]See **www.ccsinfo.com**.
[5]See **www.hitechtools.com**.
[6]See **www.propic2.com**.
[7]See **www.winpic800.com**.
[8]The program files are available for download from the author's Web site at **roland.cordesses.free.fr** as well as from the *QEX* Web site. (See Note 2.)
[9]See the official *Python* Web site: **www.python.org**.