A MICROPROCESSOR-BASED VLF RECEIVER FOR EXPLORATION

Evrim Çabuk and Günal Baki, Istanbul Technical University, Turkey.
Dr. Charles T. Young, Michigan Technological University, Houghton, Michigan

Abstract

A modern low-cost microcomputer and accessories were used to create a VLF receiver for geophysical prospecting. The signals are detected by induction coils, amplified, digitized by a sound card and analyzed by a Raspberry Pi computer, which uses a Linux operating system and the Python programming language. The peripheral devices were commercial "off-the-shelf"' items except for the preamplifiers and the induction coils which were designed and constructed by authors EC and GB. A test profile shows typical VLF anomalies across a weak conductor.

Introduction

The VLF (very low frequency) method of geophysical exploration is used to map shallow subsurface conductors. The signal sources for VLF exploration are a worldwide set of at least 36 transmitters used for government communications, with frequencies from 16300 to 80000 Hz (Loudet, 2015). The VLF signals penetrate typically tens of meters into the subsurface. The survey equipment consists of only the lightweight receiver and processor; no local transmitter is needed. VLF receivers are used primarily to detect narrow conductive zones which may contain minerals, groundwater or contamination plumes. A subsurface conductor is indicated by the tilt of the combined fields or by the presence of a vertical field. Profile data may be filtered to estimate the two-dimensional subsurface current density (Kerous and Hjelt, 1983), thus a profile may be interpreted as a current density cross section, and a gridded survey may be interpreted as a three-dimensional volume of current density or conductivity. In contemporary VLF receivers, the antenna signals are digitized into time series, which are narrow-band filtered and compared; the results are displayed in the field and recorded in digital memory.

Hardware

The VLF signal is received by ferrite-core induction coils. They are 12 inch long, .85 inch diameter plastic-coated ferrite rods manufactured by www.stormwise.com. The rods are wound with a layer and a half of ~1.0 mm diameter solid enameled copper wire creating a coil with an inductance of 35.5 mH. The coil is connected in parallel with a 1 nF capacitor to resonate at approximately 26 KHz. The coil outputs are amplified by preamplifiers made with NE5532 operational amplifier ICs with gains of 100, and are digitized by a Syber 2.0 Stereo Sound Card, which has 16-bit resolution and 96 KHz sample rate. In principle, the coils should be aligned horizontal and vertical for a given measurement. In practice, it is difficult to align the coils accurately. Instead, the system measures the coil inclination with an inclinometer and the data are corrected for the tilt. The computer is a Raspberry Pi 2, designed at the University of Cambridge Computer Laboratory to encourage programming and hands-on hardware use. It was chosen because of its modest price, its capability, and excellent free software including the Python programming language, and also suitable peripheral devices with software. The receiver is powered by a 10000 maH polymer lithium ion battery, which lasts about three hours. Correct time is provided by a DS3231 Mini RTC (real time clock). A five-inch TFT resistive touchscreen monitor provides graphical data display and software-defined pushbutton controls.

The receiver was assembled into two modules: a backpack of wooden slats containing the antennas, preamplifiers, ADC and inclinometer, and a control console, mounted in a plastic box. The antenna module is shown in Figure 1. It contains:
A. and B. induction coil antennas. C. preamplifiers. D. inclinometer.
E. terminal strip connecting the preamplifiers to the ADC and the
inclinometer to the Raspberry Pi. F. ADC. The inside of the control console is shown in Figure 2. It contains: A. Battery, B. Screen, C. Antenna connection, D. Restart/Start button, E. Raspberry Pi GPIO terminal strip, F. USB Cable, G. Raspberry Pi Ethernet port.

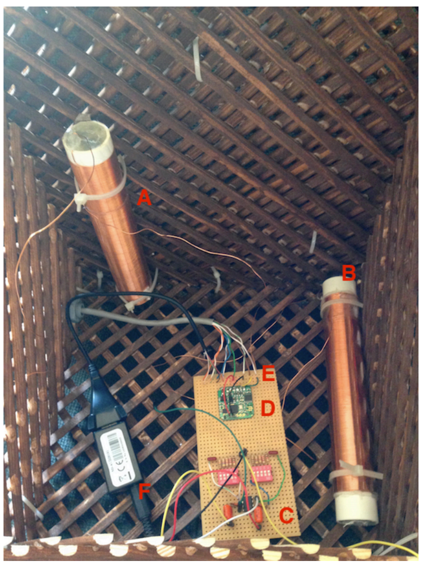

Figure 1. VLF receiver antenna module.

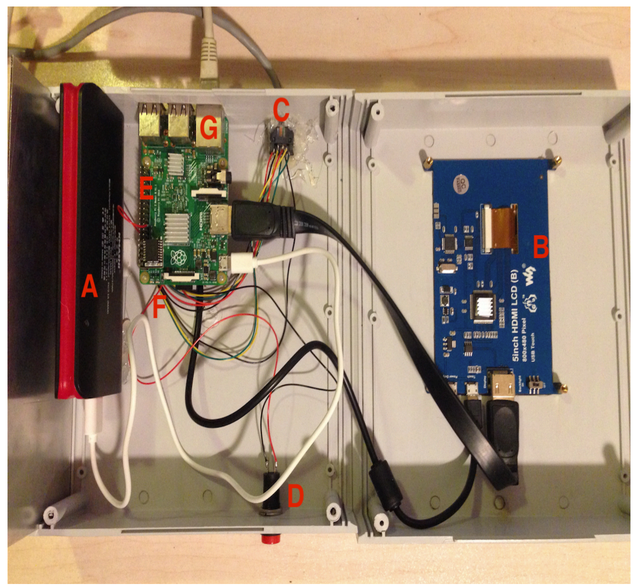

Figure 2. VLF Receiver control console mounted in a plastic box.

Operating System and Utility Software

The Raspberry PI operating system is Wheezy Linux (Beech, P. et al, 2016). The programming language was Python 3; its scientific, mathematical and graphical applications packages make it a rival to Matlab.

These software libraries were used:
Qt & PyQt Qt (Boddie, 2015): to create the graphical user interface (GUI) in Python.
matplotlib (Hunter, 2015): 2D drawing library for Python for plotting.
SQLAlchemy (Bayer, 2015): Object-Relational Mapping, to connect to the SQLite database.
pyserial (Liechti, 2015): serial port communication library, to read data from the inclinometer and UART communication ports.
pyalsaaudio (Immisch and Wilstrup, 2015): Python commands to control Linux ALSAMIXER commands which controls data acquisition from the soundcard.
ALSAMIXER (Griffen et al, 2015): Advanced Linux Sound Architecture (ALSA) Linux command line controls for the sound card.

SQLite database (Wyrick and Hipp, 2015): for storing and manipulating data.
VLFRX Receiver Software Toolkit (Nicholson, 2015): provides about 50 command line Linux routines for acquisition and analysis of multichannel VLF signals. Routine VTCARD reads a digitized data stream from a sound card using Linux ALSAMIXER routines. Routine VTCMP carries out narrow-band filtering and compares the amplitude and phase of two channels.
Data logging and display software was prepared using these libraries with Python to view signal spectra, set signal frequencies, control data sampling intervals, view narrow band signal ratios, inclinometer values, acquire data at individual stations, store and plot profile data.

Graphical User Interface Display Screens

Several graphical user interface (GUI) display screens were created. The main operating screen (not shown) provides software buttons which select sections of the program to view signal spectra, select the VLF station to receive, set the profile number and station interval, list calculated or measured profile data, acquire data and plot the data profile. Figures 3 and 4 show the GUI spectrum display and spectrum setting screens.

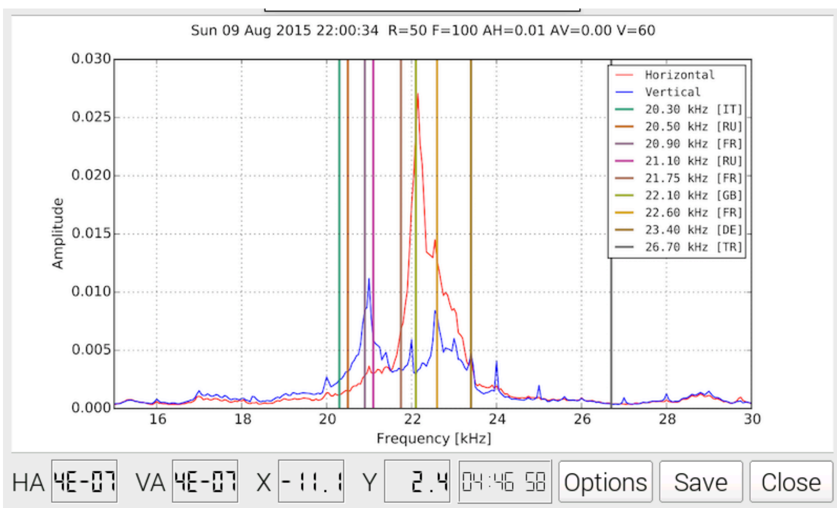

Figure 3. The signal spectrum display, including a table and markers for the frequencies of the signals expected to be received.

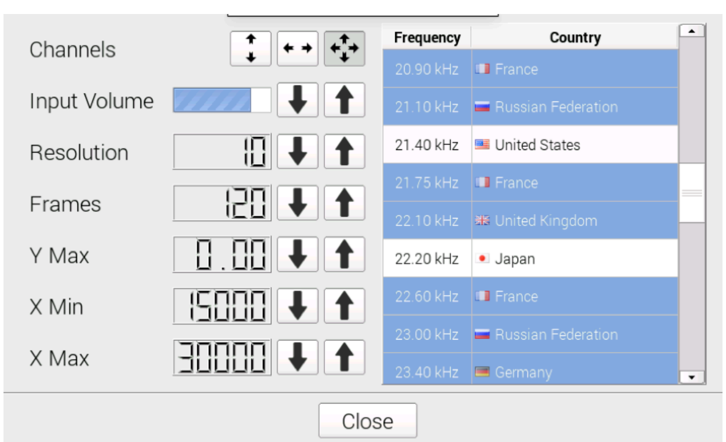

Figure 4. The spectrum settings display. The settings are adjusted by the arrow keys.

Test Profile

The test profile traverses a gravel road, passing over a drainage canal and a bridge. A profile plot of the measured in-phase and out-of-phase data computed in the field is shown in Figure 5. Data are displayed as the percent secondary (vertical) field (in phase and quadrature) divided by the primary (horizontal) field. The results display erratic values of about +- 2 percent along the profile, a clear +5 -10 percent anomaly in the out-of-phase values from about 30 to 50 meters, and about six percent positive in-phase values at the same location. These anomalies coincide with the bridge and are likely created by the steel reinforcing rods in the bridge deck and/or the conductivity of the water and sediment in the drainage canal. The erratic values in the data away from the bridge are quite typical for VLF work. Thus, these anomalies of a few percent over the bridge might be the minimum that could be reliably detected by the system. The profile plot could easily be made smoother and the anomalies made more visible if data were acquired at a closer station spacing and smoothed by a running average. Time constraints prohibited testing at more locations.

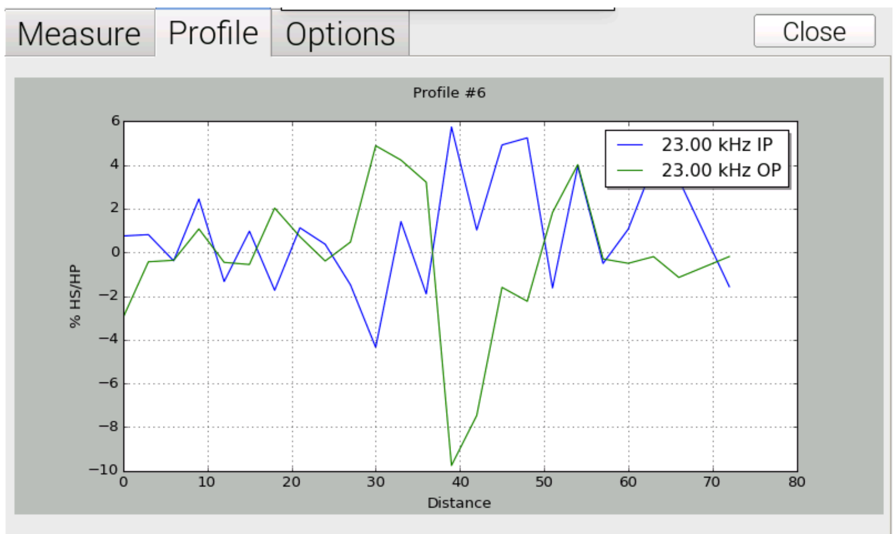

Figure 5. Plot of data obtained along the test profile.

Conclusions

The project created a VLF geophysical receiver using a modern low cost computer, off-the-shelf and custom peripherals and free software. This equipment was designed, assembled and tested as a senior project in the Geophysical Engineering Department, Istanbul Technical University, Turkey. The test profile detected a weak typical VLF anomaly. We suggest that the next generation VLF receiver use an audio card with 24-bit ADC and 192 KHz sample rate, such as the Cirrus Logic Audio Card or the Syba SD-AUD20101 7.1-Ch USB Sound Card.

The greater ADC resolution would reduce or eliminate the need for a hand-wired preamplifier and the higher sampling rate would allow reception of higher frequency VLF transmissions. This card can be controlled with ALSA software, thus it should work with the VLFRX routines.

Acknowledgements

The authors appreciate assistance from: Prof.Dr. Gülçin Özürlan Ağaçgözgü and Yrd.Doç.Dr. Tuğrul Genç, (faculty) Avni Hilmi Pazarbaşı (student) at Istanbul Technical University, Metin Zobu, manufacturer of geophysical equipment, and Paul Nicholson VLFRX webmaster. Istanbul Technical University Faculty of Mines Dept. of Geophysical Engineering provided funds to construct the receiver. Authors Evrim Çabuk and Günal Baki are senior students in the Geophysical Engineering Department, Istanbul Technical University, Turkey. Author Charles Young mentored the project over the Internet, and prepared this SAGEEP abstract from Çabuk and Baki's senior project report.

The software used for this project is available at https://github.com/ecabuk/vlf. Please realize that there is absolutely no guarantee of performance or support for the software.

References

Çabuk, Evrim and Günal Baki, 2015, Very Low Frequency Electromagnetic Method (in Turkish), Senior Thesis, Geophysical Engineering Department, Istanbul Technical University, Turkey, 44 pp.

Beech, P. et al, 2016, Main Raspberry Pi Page, https://www.raspberrypi.org/

Bayer, M., 2015. , Sqlalchemy Documentation, http://www.sqlalchemy.org/

Boddie, D., 2015. PYQT Documentation, https://wiki.python.org/moin/PyQt

Griffen, A., et al, 2015, Main ALSA Page, https://wiki.archlinux.org/index.php/Advanced_Linux_Sound_Architecture .

Hunter, J. , 2015. , Matplotlib Documentation, http://matplotlib.org/

Immisch, L. and Wilstrup, C., 2015. Pyalsaaudio, https://pypi.python.org/pypi/pyalsaaudio

Karous, M. and S.E. Hjelt, 1983, Linear filtering of VLF dip angle measurements, Geophysical Prospecting, 31, pp. 782–794.

Liechti, C., 2015. Pyserial, https://pypi.python.org/pypi/pyserial

Loudet, Lionel, 2016, SID Monitoring Station, http://sidstation.loudet.org/stations-list-en.xhtml

Nicholson, P., 2016, VLF Receiver Software Toolkit. http://abelian.org/vlfrx-tools/

Wyrick, G. and Hipp, D., 2015, Sqlite, https://www.sqlite.org/index.html