



Etymotic Research

ERO•SCAN API

Version 1.1
January 14, 2014

History of changes:

Version	Date	
1.0	January 3, 2014	Initial Revision
1.1	January 14, 2014	Additional information related to Bluetooth

ERO•SCAN API Specifications

Overview

This documents describes how to interact with an ERO•SCAN instrument using a PC program, as it applies to ERO•SCAN firmware version 101.02.

ERO•SCAN device can be a branded in different ways (as Corti, Welch Allyn etc). This documents applies to all brands.

It is assumed that the reader has working knowledge of test procedures as described in the ERO•SCAN User Manual.

In this context, ERO•SCAN features of particular interest are

- Upload of test results
- Download of patient names

Note about sample code:

It is possible to use any language that allows serial port I/O. Examples in this document are based on Visual Basic .NET 2012. With a future release of this document, we might provide stand-alone projects that can be used with minimal change to illustrate various parts of the API. A this time, however, we can only provide isolated code excerpts.

Samples provided separately:

- TransferResults.vb illustrates the method to retrieve results from ERO•SCAN
- TransferNames.vb is sample code for names download

ERO•SCAN Connection

A PC program can access the instrument as a Serial Port, either via Bluetooth, or via a USB connection. The two methods differ in some details. Most notably, Bluetooth channel must be established (PC program must have open the port) before ERO•SCAN can use it. With Bluetooth, ERO•SCAN must discover and pair with the PC, and use CONNECT menu entry on the instrument to establish connection.

Using Bluetooth

ERO•SCAN supports Bluetooth 2.1 + EDR. On the PC side, you can use a built-in Bluetooth adapter or a USB dongle.

We tested on 32 bit Windows 8.1 with Asus Mini Bluetooth Dongle (USB-BT211). Device description in Device Manager is "Qualcomm Atheros AR3011 Bluetooth 3.0".

We also tested older Windows systems with various dongles. Results varied. IOGEAR Bluetooth 2.1 USB Micro Adapter with Low Power Consumption (GBU421W6) worked well on all platforms we tested (XP SP3 32-bit, Vista SP2 32-bit, Win 7 SP1 64-bit). In general, ensure that discovery is turned on at the PC (often not done by default).

ERO•SCAN API Specifications

Virtual incoming COM port for SPP (Serial Port Protocol) is not always added by default. This can be done manually using Bluetooth settings dialog.

On Windows 8.1, Asus Mini Bluetooth Dongle installed automatically when dongle was inserted. Right-click on Bluetooth icon in notification area of the taskbar and choose "Open Settings". "Bluetooth Settings" dialog appears. Ensure "Allow Bluetooth devices to find this PC" and "Alert me when a new Bluetooth device wants to connect" are checked to enable pairing with ERO•SCAN.

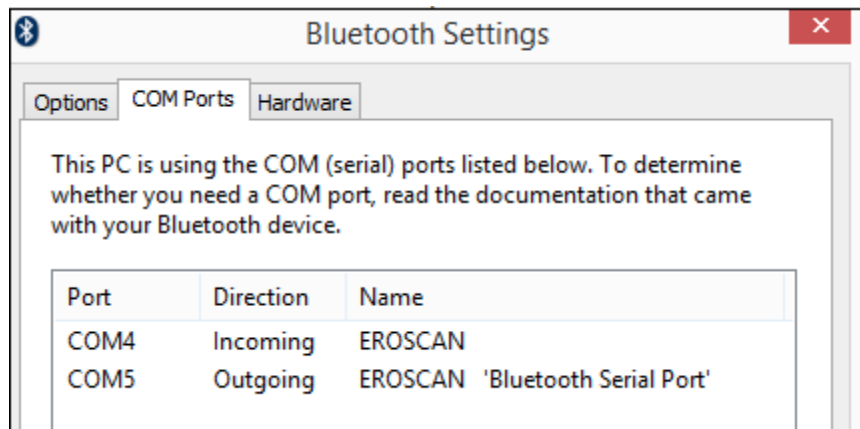
ERO•SCAN Bluetooth name (as shown on the PC) is EROSCAN, or GSI OAE for another brand.

On ERO•SCAN, first enter the Setup mode. This is accomplished by pressing Down button repeatedly until you see SETUP label with the down arrow. Press Down one more time, which brings you to the Date/Time Change screen. Press and hold Down button until green LED goes out (about 5 seconds). Upon button release you will see NEW BT DEVICE screen. Press Left or Right arrow (DISCOVER). After a delay, you may see BT DEVICE NOT FOUND. If that is the case, ensure your PC Bluetooth is turned on and discoverable.

Normally you will see one or more device names. You can show various discovered Bluetooth devices by pressing Left or right arrows when CHANGE is shown. Locate the desired device and press Down arrow (PAIR).

On the PC, you will be prompted for the PIN, which is 1234. Enter the PIN quickly, or ERO•SCAN will time out. In that case press Down arrow (PAIR) again.

When pairing is complete, look at "Bluetooth Settings" dialog on the PC. The COM Ports tab will show something like this:



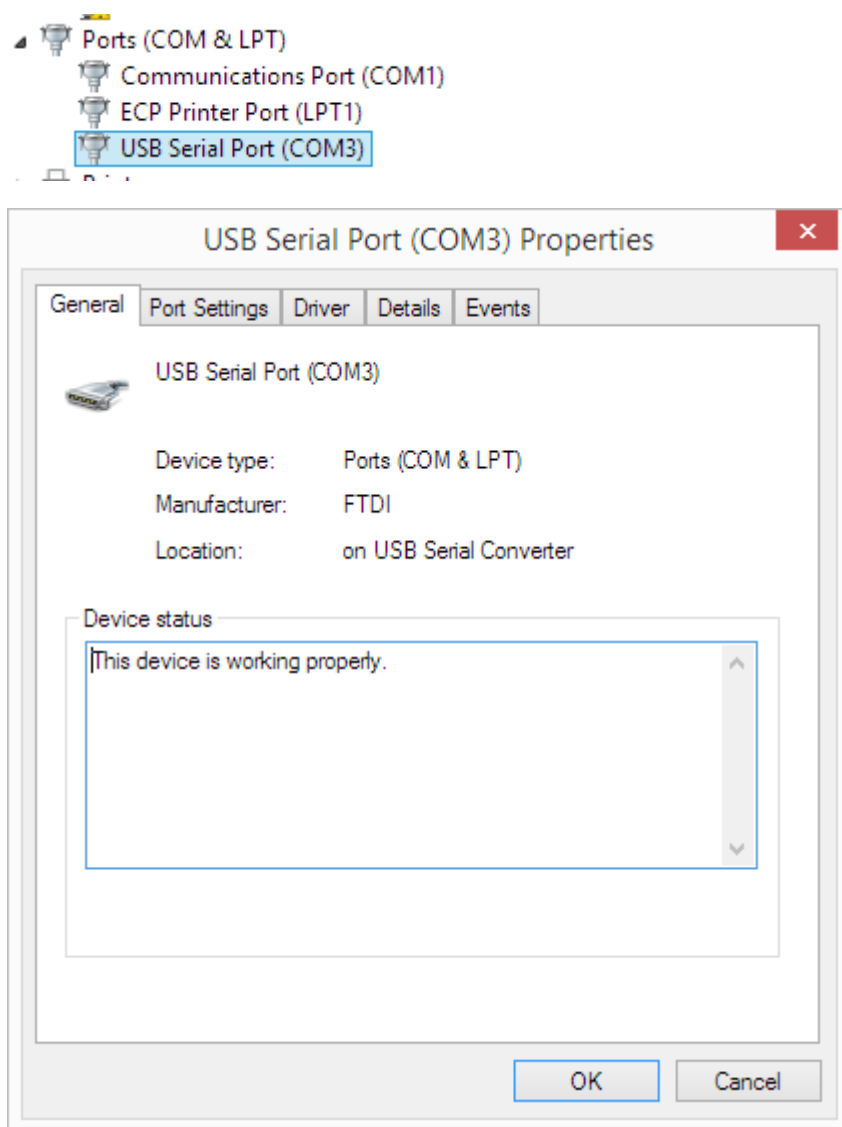
If Incoming port is not listed, create it manually. This port (COM4 in this case) is the one that PC program must open to communicate with ERO•SCAN. General flow of communications is as follows:

- Open the COM port in the PC program.
- On ERO•SCAN, press Up button. In CONNECT TO screen, press Left or Right. You should see WAITING FOR PC. If you see BT ERR, PC program likely did not open the correct port.
- PC program communicates with ERO•SCAN in the same way as with a USB connection.

ERO•SCAN API Specifications

Using USB

USB connection can only be established after the appropriate driver is installed. This happens automatically (via windows Update) when ERO•SCAN is first plugged in. Driver install may take some time. The device will show in the Device Manager in a way similar to this:



ERO•SCAN API Specifications

In this example, the port to use from the PC program would be COM3.

When ERO•SCAN is connected to a PC via the USB cable and turned on, “WAITING ON PC” is displayed on its screen. If this message is not displayed, likely there is a driver problem, and PC communications will not work.

Polling and ERO•SCAN Modes

ERO•SCAN is designed to use the same communications channel with PC or a portable Bluetooth printer. All ERO•SCAN communication (with the printer or with the PC) are based on request-response message pairs. The master device (which is ERO•SCAN after power-up) issues a request and expects a response from the slave device.

When ERO•SCAN detects USB connection, it will send a “poll” message (a two character sequence, “ESC” “v”). Baud rate used by ERO•SCAN at this point is 115200. There may be no response, in which case ERO•SCAN will repeat the same message after a short delay (approximately 1 second). If there is a response, it may be from a printer, in which case ERO•SCAN will send print content for test results.

If it is the PC program that responds, it will respond differently from a printer. PC response is a single byte, 0xF2. With this response, ERO•SCAN will enter “Remote Program Mode”, which means that it will expect the PC to be the master (meaning it will initiate requests to which ERO•SCAN will respond). ERO•SCAN display will show “CONNECTED” while Remote Program Mode is in effect. See TransferResults.vb to see how the PC code analyzes the poll and responds to it.

ERO•SCAN Functions

During Remote Program Mode, PC program will invoke various “functions”, ERO•SCAN firmware routines identified by a number. Each function expects parameters and returns results. These functions are summarized separately.

See DetermineFirmwareVersions.vb as an illustration. This routine first uses function 9. It sends (see sendByteWithEcho) two bytes (0 and 9) to invoke fun9. This function is used to obtain main board firmware version. There are no input parameters, and there is a one word response. A “word” in this context, if not specified otherwise, means 3 packed bytes from ERO•SCAN DSP.

Likewise, fun34 returns display board firmware version.

ERO•SCAN contains flash memory that is organized in sectors. It also contains non-volatile memory (referred to as CMOS in code, although actual hardware implementation may no longer be CMOS). In any case, PC program needs to be aware of internal organization of both types of memory. Moreover, as sector size had changed over time, variables like SectorSize may refer to “virtual”, rather than real, sector size. This is function dependant.

ERO•SCAN API Specifications

12 Frequency Support

Current firmware supports 12 frequency tests, so you can assume that will be the case in all the sample code. Following our sample, SectorLength will be 512, and TestStartSector will be 655360.

CMOS Memory Pointers

Fun32 is used to obtain pointers used for test retrieval. See Summary below.

Name Retrieval

If patient identification was downloaded to ERO•SCAN, it said to be in Names mode. These “names” can be various pieces of information, typically last name, first name – in Unicode, and an ID (binary) used to link to the patient record in the PC database. However, it is also possible to use something else, such as medical ID, for privacy or other reasons. In any case, ERO•SCAN sets aside 120 bytes to record patient identification, and links it with each test as appropriate. The content is not used directly by ERO•SCAN mainboard firmware, but is passed on to back to PC, or to display board so ERO•SCAN can show which patient is being tested.

In the sample provided, you can see that fun15 (see Summary below) is used to retrieve sectors containing name information. NameListAddress has the value of $7 * 256 * 256$. In the code, information is parsed a certain way, but you may adjust it for your needs.

Obtaining Test Results

See RetrieveAndParseTestSector, fun45 and ParseSector12Freq for details. ConvertToWords swaps bytes so they can be properly interpreted by the PC.

ParseSector12Freq is the core routine where SectorBuffer values (raw DSP content) is interpreted in various ways. For example, convert1() converts DSP-specific number notation to Integers, ASCII values are converted to Unicode, scaling is accounted for and so on.

Interpreted test results can then be used as appropriate in your application. ERO•SCAN returns to normal mode by function 21 (“CONNECTED” changes to “WAITING FOR PC”).

Downloading Names

Code flow is initially similar to what is described in Obtaining Test Results. Downloading names erases all tests present in the instrument. In the sample code, the user is warned if there are any. Test pointers obtained by fun32 are used as previously explained.

A special case of name download is clearing all names, without supplying new ones (“No Name” mode). This is accomplished by function 39.

fun38 is used for name download. This function assumes that the 2-dimensional byte array representing sectors was already set up with the appropriate content. When function 38 is invoked, the PC sends a byte containing the

ERO•SCAN API Specifications

number of name sectors. ERO•SCAN responds with a handshake, a single byte of zeroes. PC sends 258 bytes, and then reads them back from ERO•SCAN. The content must match to what was sent. This is repeated for each sector.

ERO•SCAN returns to normal mode by function 21 ("CONNECTED" changes to "WAITING FOR PC").

To ensure your name content will be displayed on ERO•SCAN display, assign the first 30 bytes (15 Unicode characters) to the first identifier (last name, for example). The second identifier should likewise be assigned to the next 30 bytes. The remainder of the sector can be any information useful to your application. With Unicode, you must swap byte order to make 2 words from 6 bytes.

One possible implementation is illustrated in CreateSectorImage. NAME_TOTAL_SECTORS is 25, NAME_BYTES_USED_IN_SECTOR is 252, NAME_TOTAL_RECORDS_PER_SECTOR is 2. This implementation is consistent with ExtractIdentificationData function (from TransferResults sample).

Function Summary

Function Number	Description and Notes	Input	Output
32	<p>Obtain selected information from CMOS memory.</p> <p>When PrintSaveMode equals 1 ("Save All"), ERO•SCAN saves up to 250 tests. Otherwise ("Save Last", it only saves one result for each ear (0, 1 or 2 tests in total are saved).</p> <p>Test that were performed reside between PointerRead and PointerWrite.</p> <p>PointerLeft and Pointerright apply to Save Last mode. These values also can have value 4095 (meaning "no results").</p>	n/a	<p>6 words:</p> <ul style="list-style-type: none">• PointerWrite• PointerRead• PointerLeft• PointerRight• PrintSaveMode• NamesPresent

ERO•SCAN API Specifications

	NamesPresent indicates whether ERO•SCAN has patient names loaded, or it works in No Names mode. See sample code.		
15	Send flash memory to the PC	StartSector	258 bytes
45	Send test sectors to the PC. Similar to fun15, but used specifically for test records. Hides the fact that a test record is not contiguous.	StartSector	510 bytes
21	Terminate "Remote Program Mode"	n/a	n/a
39	Clear names	n/a	n/a
38	Download names	Number of name sectors - N (1 byte) 252 bytes for each of N sectors	Handshake (0x0) Echo each sector