

# Frequently asked questions and the answers (FAQ)

**Q:** Does the OpenSCADA system work under QNX, FreeBSD, MS Windows, ...?

**Re:** OpenSCADA system is developing under multiplatform principles, being based on recognized world standard POSIX and multiplatform libraries. However in view of limited of resources the project is conducted only in OS Linux. Adapting the system to other platforms are planned after the version 0.7.0 release. The further support of a particular platform will depend on community interest and presence of an individual maintainer for a hardware-software platform.

**Q:** How is OpenSCADA built from the source code?

**Re:** Building of any program project from the source code, which complexity is above the average, is rather hard work. Appropriate preparation and preliminary experience of the building are necessary for the solution of this task. If there is an opportunity to get OpenSCADA system already pre-built for your environment, you ought to use it. If there are no packages under your environment or you consciously wish to build a system from source code take advantage of the guide for building from source code: <http://wiki.oscada.org/HomePageEn/Doc/BuildFromSource>.

**Q:** The OpenSCADA system is unclear and hard to understand. Why is this, and are the quickstart manuals available?

**Re:** The OpenSCADA system is depth described, marginally structured and formalized. But the unclear and the complexity impression is predicted. From project's targets, the system is marginally modular, and so there are a lot of configuration and using variants. From one side this makes understanding more difficult but from the other side it increases chances of comprehension of Your special task. It is always necessary to remember that all new things are difficult and hard to understand, but it doesn't mean that it is really so. You must overcome this psychological barrier before using any new things. The quickstart manual is available here: <http://wiki.oscada.org/HomePageEn/Doc/QuickStart>.

**Q:** How is recurrence of archiving values defined?

**Re:** Archiving is independent of controllers and parameters task, and it is carried out by a modular subsystem "Archives". What and how to archive is determined individually for each attribute of parameter. Two modes of archiving are provided: passive and active. Passive archiving is determined by cyclical data acquisition from the controller, by the cycle of the module of data acquisition. Active archiving is provided by a task of the archiving subsystem independently of a cycle and type of data acquisition by "DAQ" module.

**Q:** How is multilingual support provided? Can I execute localization in my native language?

**Re:** Multilingual support is provided on the basis of the standard of internationalization I18n. And files of internationalization of modules are separated from a file of internationalization of core system. It provides high-grade support of independent distribution and development of modules to OpenSCADA system. Translation of the system as a whole and modules can be separately made irrespective of presence of the source code of system. For translation it is enough to receive \*.po or \*.pot files of the necessary component and to make translation of messages in the files in usual text editors from English to the necessary language.

**Q:** How is the speech signaling system carried out?

**Re:** The speech signaling system, as well as other methods of the signaling system, is an element of a subsystem "User interfaces" and organized into the modules of visual control area (VCA), and exactly into VCA engine UI.VCAEngine and visualisators UI.Vision and UI.WebVision.

**Q:** Where and how are additional logic processing of attributes of parameters carried out, including logic linkage in one object?

**Re:** Any mathematical processing of attributes of parameters can be made in computing controllers (modules of subsystem "Data acquisition"), for example in the block computing controller (DAQ.BlockCalc) and the controller on the basis of Java-like language (DAQ.JavaLikeCalc). Besides processing can be carried out at a logic level of parameters (DAQ.LogicLev), in modules of subsystem "Data acquisition", in specially specified for these purposes controllers (working on parameters' patterns) or containing the built-in realization of the mechanism of patterns of parameters. I.e. the user can form parameters with the necessary structure and algorithm of post-processing, forming logically connected objects. A processing part can be executed into VCA, directly at visualization (UI.VCAEngine).

**Q:** How is the sharing of access organized?

**Re:** A security scheme similar to UNIX OC is used. So, necessary components have their owner, belong to group and contain a triad of access "rwxrwxrwx". Besides the given mechanism of security is introduced in the interface of management of system OpenSCADA which, in its turn, penetrates all system.

**Q:** Is it possible to operate the system OpenSCADA by means of a usual WEB-browser?

**Re:** Yes, it is possible. For a configuration of OpenSCADA system from a WEB-browser there were created the modules UI.WebCfg and UI.WebCfgD which are called by module of "Protocol" Protocol.HTTP.

**Q:** Whether it is possible, and how the reservation of parameters is realized?

**Re:** Realized of following schemes of reservation:

- Reservation of sensor: provides an opportunity in one parameter to describe several the same-type sensors which will have own attributes of values. The resulting value of sensors will be located in generalizing attribute of value.
- Reservation of channels: provides an opportunity of consolidation of parameters from different stations/controllers in one multiple parameter. During accessing the active parameter (controller), or preferable, in the case of activity more than one parameter (controller), is selected. The given scheme also distributes of loading on the communication interfaces of various stations/controllers.

**Q:** Where are scales of parameters and various settings of signaling systems checked?

**Re:** Checking of scales and settings can be realized by means of the module of the controller with the signaling through corresponding attributes of parameters. And also at a logic level of parameters for "crude" sources of data. The given analysis can also be made directly in the visual control area (VCA).

**Q:** Does the system support addition/removal/updating of modules while they are working (without a stop)?

**Re:** This feature is incorporated in the system and provided by "Management of modules" subsystem. Actually, updating of modules can automatically occur after detection of the new version of the module.

**Q:** Can the system work with streaming data?

**Re:** Streaming and batch data acquisition works together with archive and its buffer. I.e. the source, having received a package/block of data, directly places it into the archive of attribute of parameter, or takes a package from the buffer of archive at streaming outlet.

**Q:** Can I create my own module for any subsystem?

**Re:** Yes, certainly. For the help in this task the document, where the architecture of a kernel is practically completely described, its functions and API of the modules of various subsystems, is created. This document is here: <http://wiki.oscada.org.ua/Doc/API> (RU). In addition, for each subsystem created a module template, you can quickly create a new module.

**Q:** Is it possible to realize intermodular connections with OpenSCADA?

**Re:** Intermodular connections can be of the following types:

- The standard interface of access - is made by means of the virtual interface of modular subsystems.
- The expanded interface of access - provides export of functions of the external interface by means of the exporter module, and the subsequent connection of the importer module to these functions by means of functions of object <TModule>.
- The user functions - any component of system can register its own user functions which, in a consequence, can be used in an environment of the user programming of OpenSCADA.