

OpenSCADA 0.7.0

Contents table

OpenSCADA 0.7.0	1
Introduction	1
1 Implementation of planned tasks	3
2 Optimization, increasing of stability, sustainability and system's performance	3
3 Improvement and stabilization of the VCA	5
4 Formation and stabilization of API of user's programming	6
5 System-wide expansions	7
6 Significant improvements and fixes of the individual modules	8
7 Formation of the documentation, translation of documentation and interface	9
8 Unification of the demonstration database	9
9 Solutions, based on OpenSCADA	9
Conclusion	10
Links	10

Introduction

The release of an open SCADA (Supervisory control and data acquisition) system of version 0.7.0 is the first stable, industrial release, which is associated with the completion of beta testing that began from version 0.6.4.

The main purpose of this release is providing the community of users and developers of free software with the platform for the solutions of complex automation systems and related solutions, as well as providing commercial services on the basis of the OpenSCADA project.

This release is the first stable release, for which technical support is available from the developers and for which it is planned to release fixes during a long time.

This document is the processing (compilation) of <ChangeLog> document of the OpenSCADA system version 0.6.4, which is designed to cover brief and clear the new features of the OpenSCADA system. To get acquainted with all the changes in the OpenSCADA system it is possible in the file ChangeLog from the distribution kit of the system, or here (RU): <http://wiki.oscada.org/Works/ChangeLog>.

The key features of this version are:

- Implementation of planned tasks.
- Optimization, increasing of stability, sustainability and system performance.
- Improvement and stabilization of the VCA.
- Formation and stabilization of the user programming API.
- System-wide expansions.
- Significant improvements and fixes of individual modules.
- Formation of the documentation and translations of documentation and interface.
- Unification of the demo database.

New and updated modules:

- Archive.FSArch - 1.3.1 -> 1.4.1, fixes and enhancements to improve performance;
- Archive.DBArch - 0.9.0 -> 0.9.2, fixes and some improvements;
- DB.DBF - 2.0.1 -> 2.0.2, fixes;
- DB.MySQL - 1.6.1 -> 1.6.2, some improvements;
- DB.SQLite - 1.6.1 -> 1.6.2, improvements;
- DB.FireBird - 0.9.1 -> 0.9.5, significant improvements;

- DB.PostgreSQL - 0.9.0, new module;
- DAQ.DiamondBoards - 1.2.0 -> 1.2.1, API change adaptation;
- DAQ.System - 1.7.0 -> 1.7.2, code cleanup and API change adaptation;
- DAQ.BlockCalc - 1.2.1 -> 1.4.0, improvements and fixes;
- DAQ.JavaLikeCalc - 1.5.0 -> 1.8.0, significant improvements, bug fixes and optimization;
- DAQ.LogicLev - 1.0.1 -> 1.1.2, improvements and fixes;
- DAQ.SNMP - 0.4.0 -> 0.4.1, API change adaptation;
- DAQ.Siemens - 1.2.1 -> 1.2.3, fixes and API change adaptation;
- DAQ.ModBus - 1.0.1 -> 1.1.1, improvements and fixes;
- DAQ.DCON - 0.3.0 -> 0.3.3, improvements and fixes;
- DAQ.ICP_DAS - 0.5.0 -> 0.7.2, significant improvements and fixes;
- DAQ.DAQGate - 0.8.0 -> 0.9.1, significant improvements and fixes;
- DAQ.SoundCard - 0.6.0 -> 0.6.1, API change adaptation;
- DAQ.OPC-UA - 0.6.0, new module;
- Transport.Sockets - 1.4.0 -> 1.4.5, improvements and fixes;
- Transport.SSL - 0.8.2 -> 0.9.5, improvements and fixes;
- Transport.Serial - 0.5.0 -> 0.7.1, significant improvements and fixes;
- Protocol.HTTP - 1.4.1 -> 1.5.0, significant improvements;
- Protocol.SelfSystem - 0.9.1 -> 0.9.3, fixes and API change adaptation;
- Protocol.UserProtocol - 0.6.0, new module;
- Protocol.ModBus - 0.5.0 -> 0.6.1, improvements and fixes;
- Protocol.OPC-UA - 0.6.0, new module;
- Special.FLibComplex1 - 1.0.4 -> 1.0.6, improvements and fixes;
- Special.FLibMath - 0.5.1 -> 0.5.2, API change adaptation;
- Special.FLibSYS - 0.9.0 -> 0.9.2, improvements and fixes;
- Special.SystemTests - 1.3.7 -> 1.5.0, remaking and the unification of the tests;
- UI.QTStarter - 1.5.3 -> 1.6.0, improvements;
- UI.QTCfg - 1.8.1 -> 1.9.0, significant improvements and fixes;
- UI.WebCfg - 1.5.3 -> 1.5.4, fixes and API change adaptation;
- UI.WebCfgD - 0.6.2 -> 0.6.5, fixes;
- UI.VCAEngine - 0.9.0 -> 1.0.0, significant improvements, bug fixes and optimization;
- UI.Vision - 0.9.0 -> 1.0.0, significant improvements, bug fixes and optimization;
- UI.WebVision - 0.7.0 -> 0.9.0, significant improvements and fixes;
- UI.WebUser - 0.6.0, new module;

1 Implementation of planned tasks

In accordance with the plan of release the following tasks were implemented:

- *Implementation of the primitives "Link" and "Function" of the VCA.* - In a view of the unprincipled to the industrial release and unclaimed at the moment, this task has been moved beyond the first industrial release.
- *The stabilization and release of industrial branch of OpenSCADA.* - The big work on stabilization and optimization of the industrial release was done, as detailed in the next section.
- *Implementation of the database ODBC module, PostgreSQL and transport module of the DB.* - Maxim Lysenko has done the implementation of the "PostgreSQL" DB module. Implementation of the database module "ODBC" was considered inappropriate, because the "ODBC" technology is actually the interface of SQL-queries that do not address specific SQL-dialects of the real database, and therefore can not be right, without regard to the type of database, used in solving OpenSCADA basic tasks. The prospect of using "ODBC" is to implement access to the database that have no "native" API of direct access to the database, but there is the ODBC driver, or to provide a simple interface of SQL-queries in OpenSCADA. The implementation of the transport module of DB was moved beyond the first industrial release.
- *Implementation of the new mechanism/module of OpenSCADA system's test.* - To perform the intra-system tests of OpenSCADA the mechanism of testing module was implemented as the user's API functions, which on the one hand retains the possibility of an autonomous call of static tests and on the other it provides an opportunity to call tests one by one, manually, as well as from user's scripts in any order and with the necessary number of individual tasks.

2 Optimization, increasing of stability, sustainability and system's performance.

During the work on this version, as well as its practical adaptation, it was found and corrected in a total of about 200 errors.

In addition to correcting of big number of errors it had been done the work on optimization of the various OpenSCADA's components. Especially, notable optimization was made in the Visual Control Area (VCA) and its visualizers.

Lets list the most significant errors, correction of which significantly affected the increasing of stability:

- *System-wide:*
 - The incorrect setting of policy for real-time threads is fixed. It has not been set the PTHREAD_EXPLICIT_SCHED thread flag.
 - The control of dual capture of the resource in the single thread and capture the resource by timeout.
 - Mechanism to prevent the multiple startup is corrected for verification of PID of the blocked process due to removal by the system the lock-files in the directory /tmp.
 - The packing of classes structure's data using the # pragma pack (push, 1) is disabled because of the appearance of implicit buzz bug of one of the threads and blocking the rest on the OpenSCADA firmware of LP8x81 PLC.
 - Initialization and updating code of modules is updated to avoid the double initialization of shared libraries.
 - The function of the task creation is corrected to prevent crashing when creating the automatically closed tasks during the rapid closing.
 - The hang of function cron() is fixed at reason zero scheduling set.
- *Transports:*
 - *Serial, DAQ.DCON, DAQ.ModBus:* Transport of the serial interface is optimized, is made the typical one and is corrected for several errors. The transport's clients are adapted to use the new mechanism.

- *Sockets, SSL*: The possibility of incomplete recording by the write() function is fixed. When writing to the socket, this function writes only the part of the size of its buffer (50k). For complete record of large messages the repeat of record is added.
- *Transport.SSL*: It is finally fixed the waiting for data response from the server by waiting the data processing in the BIO_read() functions.
- *Transport.Sockets*: The processing of null result of the waiting the tail of messages is fixed to avoid hangs on the queries in DAQ.System.HDDTemp.
- *Transport.Serial*: CSTOPB flag using for stop bits number set is fixed.

* *Data acquisition*:

- *ModBus*: The loss of resources during the data acquisition is fixed. For this reason it is excluded the possibility of assigning the string directly to the object of the resource string.
- *JavaLikeCalc*: The using of functions for the parameters of the DAQ controllers from the IO functions for first use is fixed.
- *DAQ, BlockCalc, JavaLikeCalc, LogicLev, Siemens*: The receiving of the erroneous data on the redundancy scheme in the nonredundant tasks and with the absence of redundant stations is fixed.
- *Protocol.HTTP, UI.WebCfg, UI.WebCfgD, UI.WebUser, UI.Vision*: The value of the HTTP-parameter "Content-Type" is fixed for the "Charset" element.
- *UI.VCAEngine*: The crash of the VCA when changing the mutable attributes, that are multiply inherited, is fixed.

Optimization and performance improvement:

- *SYS.XML*: Conversion function of an XML tree into a string is optimized over the time.
- *SYS.Archive*: The mechanism for adaptive generation of requests to the archives of various qualities in the case of a general request is added. It prevents the long processing time in the case of requesting the large intervals of time and the availability of lower quality archives.
- *DAQ, DAQ.DAQGate*: Optimization of request between redundant stations over the requests' size and time is done. Only archive attributes of DAQ parameters and attributes, planned to request, are requested.
- *Archive.FSArch*: The creation of information files of the packaged archives without storing these data in the database is added. This feature allows quick connection of large archives to other stations.
- *UI.Vision*: In the "ElFigure" primitive for the drawing it is used QPixmap instead of QImage, it greatly improved the performance.
- *UI.Vision*: To construct the image of the "Diagram" primitive the QImage is used instead of QPixmap, it greatly improved the performance.
- *TFunction, DAQ.JavaLikeCalc*: The enclosed context of the functions is added in order to improve the performance of external calls.

Optimization of the memory usage:

- *SYS*: The system is switched to the traditional "map" instead of "hash_map" for all objects in order to reduce memory consumption.
- *UI.VCAEngine*: The object of the "Attr" attribute is divorced from the "TCntrNode" object in order to reduce memory consumption by up to 25%.

3 Improvement and stabilization of the VCA.

Notable changes were made in the visual control area (VCA), namely, in the VCA engine (UI.VCAEngine) modules: UI.Vision and UI.WebVision visualizers. These changes were aimed at stabilizing, optimization of memory consumption and improvement of user's properties of VCA.

Improvements of the VCA:

- The full support for vertical scaling of the diagrams is added.
- The support for alpha channel in color attributes of all primitives is added.
- *UI.VCAEngine*:
 - It is included and made safe the inheritance of the description of mutable attributes.
 - The new user's API functions for the sessions are added: `wdgAdd()`, `wdgDel()`, `link()` and `linkSet()`. Addition of these functions allowed to implement the concept of a fully dynamic formation of the user interface.
 - Properties of access rights to the widget are implemented by the attributes "owner" and "perm" to increase the flexibility of control.
 - The mechanism for automatically creating and running session at startup is added.
- *UI.Vision*:
 - In the attributes inspector the ability of group setting of same attributes of various widgets is added. Setting is done in the group container of the properties of attributes' inspector that appears when you select multiple widgets simultaneously.
 - In the "ElFigure" primitive for the drawing the QPixmap is used instead of QImage, it greatly improved the performance.
 - To construct the image of the "Diagram" primitive The QImage is used instead of QPixmap, it greatly improved performance.
 - The support of properties of the user interface is added. The control and changing of the toolbar icons' size is added.
 - The ability to copy the values of attributes and links through the context menu of attributes' and links' inspectors is added.
 - The search function in the "TextEdit" element is added.
- *UI.WebVision*:
 - The support of turning the "Text" primitive is added.
 - The generation of the basic mouse events is added. For processing of some events the stack of processing functions is added.
 - For the sender-user the access to its session and control is adapted.

4 Formation and stabilization of API of user's programming.

Extensively it has been made the formation of new object API of the user's programming, which provides the integration of user's functions in the objects' tree of OpenSCADA. In addition, a number of changes to the existing library of functions of the user's API was made.

In particular the following changes were done:

- *SYS*:
 - *SYS*, *Special.FLibSYS*: XMLNodeObj object of the user's API moved into the OpenSCADA core (tvariant.h and tvariant.cpp files).
 - User's system API of OpenSCADA based on the TCntrNodeObj object is expanded to control the user's access rights.
 - TAreaObj object is renamed to TArrayObj.
 - Properties of objects, functions' calls and exceptions' processing of user's API are unified.
 - *SYS.TVarObj*: The propList() function is added to provide an ability to scan the attributes.
- *DAQ.JavaLikeCalc*:
 - Functions of the user's API: insert(), replace(), toReal(), toInt(), parse(), parsePath() and path2sep() are added in the object of the string type value.
 - The isEval() function is added to basic types in order to detect the EVAL-values.
- *Special.FLibComplex1*: PID function is expanded with new parameters: Kd, Tzd and followSp. The Tf parameter is removed.
- *Object API of the user's programming*:
 - *SYS*: User's API functions: system(), message(), XMLNode(), cntrReq(), time(), localtime(), strftime(), strptime(), cron(), messDebug(), messInfo(), messNote(), messWarning(), messErr(), messCrit(), messAlert(), messEmerg() и strFromCharCode() are added to the root object of the OpenSCADA core.
 - *DB*: SQLReq() user's API function is added to the database object.
 - *Transport*: messIO() user's API function is added to the object of the outgoing transport.
 - *UI.VCAEngine*:
 - User's API functions: user(), almSndPlay() and almQuittance() are added to the session's object of the project.
 - To the "Widget" object the functions of user's API were added: ownerSess(), ownerPage(), ownerWdg(), attrPresent(), attr(), attrSet(), wdgAdd(), wdgDel(), link() and linkSet().
 - The special attribute "this" is added into all VCA's scripts to access the widget's object and further calling of its functions of user's API.
 - *Archives*: messGet() user's API function is added to the object of subsystem "Archives".
 - *DAQ.Value*: The functions of user's API get() and set() of the attribute are added.

5 System-wide expansions.

To the system-wide API of OpenSCADA system there have been made the significant changes and expansions in order to improve the functionality and overall stabilization:

- *SYS*:
 - The mechanism to prevent re-run is added to the scripts: demo-start and user-start.
 - The global functions to create (register) taskCreate () and to close taskDestroy () of the threads are added. All components of OpenSCADA use new API.
 - The debian/* files for support the building of Debian packages are added.
 - The strParse() function is added for the expansion of the strSepParse() function in order to support of multi-character separators and for the integration of single-character ones.
 - The infrastructure of the debugging counters is added.
 - The ability to assign processors to the threads of OpenSCADA on multi-processor systems and multi-core processors is added.
 - The task manager is added. The id field of the thread's process is added to the task manager.
 - To implement specific functions the wrapper on the tasks of OpenSCADA is added. The "TError" exceptions' processing is added for the tasks inside the wrapper.
 - The core's code of the OpenSCADA is placed in the "OSCADA" scope. Version of the OpenSCADA's core library is increased to 1:0:0.
 - The rules of access to all elements of the control interface are unified.
 - The check for a version 2.0 of the libtool library is added. It is enabled the building of only shared libraries for modules.
 - The ID of the module is placed in the class of the constructor to simplify initialization of translating the text messages of the modules.
 - For all translations the parameters "-C --no-location --no-wrap -k_" of the xgettext utility are used. It is used for cleaning the garbage from the translation files.
 - API versions of the all modular sub-systems are set to 5.
- *SYS.Function, DAQ.BlockCalc, DAQ.LogicLev, UI.VCAEngine*: The mechanism for changes' monitoring in the parameters of "DAQ" subsystem's controller is added.
- *DB*:
 - The automatic opening of the available tables on request to them is added.
 - The transaction control is added to the function sqlReq(), as well as the function transCloseCheck() is added.
- *TFunction, DAQ.JavaLikeCalc*: The enclosed context of functions is added in order to improve the performance of external calls.
- *DAQ*: The error code is added to the state of the controller, which allows us to identify state and display it.
- *Protocol, Transport, DAQ.ModBus, Protocol.UserProtocol*: The function itemListIn() is added to select elements of a protocol. The module's API of the subsystem is changed to 3. The selection of protocol's elements via the function TProtocol::itemListIn() is added.

6 Significant improvements and fixes of the individual modules.

While working on this industrial release the significant changes, improvements and stabilization were made with the individual modules of OpenSCADA:

- *DB.MySQL*:
 - The support for unbounded values of "Real" and "Integer" data types is added.
 - The conversion of EVAL for the real type is added.
- *DB.SQLite*: The transaction mechanism is improved to support user's SQL-queries and closing the old transactions.
- *DB.FireBird*: The support for multilingual text variables is added. The support of transactions is unified.
- *DAQ*:
 - *DAQGate*: The support of ability to request the individual attributes is added. It is used to request the archived and often requested attributes. All other attributes are requested with the period of synchronization by a full request. It is used to reduce the workload and time in exchange of DAQGate with a remote OpenSCADA station.
 - *ICP_DAS*: The status of the DIP-switch for LP-8781 PLC is added. The functions to send the messages via the serial interface are moved into the general serReq() function and all requests use it.
 - *ModBus*: The suffixes "i2, i4, f, b0" are added to the registers (R and RI) for flexible access through a variety of data types.
 - *JavaLikeCalc*:
 - The support of the review loop "for(<i> in <obj>) <code>;" to scan the object's properties is added.
 - Properties of objects, functions' calls and exceptions' processing of user's API are unified.
- *Transport*:
 - *Serial*:
 - The ability to select the type of the thread's control is added. It is supported the hardware (CRTSCTS) and software (IXON IXOFF) thread's control.
 - The support of the modem mode for all input and output transports is added.
 - *SSL*:
 - The support of KeepAlive limits is added. The support of reconnection after the connection loss for prevention the loss of messages is added.
 - The getting of the sender's address is added.
 - *SSL, Sockets*: In the outgoing transport it is added the timings for the individual connection time and time of the subsequent responses control.
 - *Protocol.HTTP*:
 - The function of the outgoing protocol is added.
 - The support of automatic login is added.
- *Archive.FSArch*: The creation of information files of packaged archives is added without storing these data in the database. This feature allows quick connection of large archives to other stations.
- *UI.QTCfg*:
 - The adjustment of rows for tables after adding the new rows is added. The support for copying the contents of the table to the clipboard is added.
 - The search function in the "TextEdit" element is added.
 - The ability to copy the list of selected nodes is added.

7 Formation of the documentation, translation of documentation and interface.

Since this release is positioned as an industrial release with a prolonged period of support, it was finished writing the documentation in Russian and translation into English of all the main documentation. In addition, the translation of the user's interface into English, Russian, Ukrainian and German languages is completed:

- The README translation into Ukrainian is added.
- The translation of the DAQ document and the summarizing document into English is done.
- The document "Quick Start" is added.
- The translation of the document "Quick Start" into English language is added.
- General document "OpenSCADA" in English is updated to include the following parts: "Quick Start" and DAQ.OPC-UA.
- The interface translation of the system core to Russian, Ukrainian and German languages is updated.
- For all translations the parameters "-C - no-location - no-wrap-k_" of the xgettext utility are used. It is used for cleaning the garbage from the translation files.

8 Unification of the demonstration database.

During the preparation of the industrial release, considerable attention was paid to improving the demo database as a vivid demonstration of the OpenSCADA system:

- Significant upgrade to support new features and include new improvements from the automation control system of ball mills.
- The support of user's calling of the demonstration database in the full mode on the record is added.
- The scaling of existing elements in the case of absence of rows and/or columns for the overview frame and the result graphics is added.
- It is updated to include new models of machines of technological processes and for adapt all the other models.
- It is updated for:
 - Addition of the signaling objects of the compressors: KM201, KM301, KM202 and KM302.
 - Support of the manual input of variables from the control panel.
 - Translation's update.
 - Fixing of the several bugs.
- Script of the main page of the project is rewritten to use new functions of the user's API.
- The new mechanism to access user's interface elements is used. The real read access for a simple user ("user") is provided.

9 Solutions, based on OpenSCADA

Actually the whole range of automation solutions is built on the basis of OpenSCADA project, which clearly demonstrates the capabilities and is the signal of readiness of the system for the wide industrial application.

In general, on the basis of OpenSCADA the authors of the project solved the following tasks: the full dynamic simulation (modeling) of technological processes (TP), process control at the level of the programmable logic controller (PLC) and formation of HMI of the TP.

Modelling of the TP is presented by the solutions: "Library of models of technological devices", "The dynamic model of real-time of the Anastasievskaja GLKS" and "The dynamic model of the steam boiler number 9 DMKD" . Besides the solution "The dynamic model of real-time of the Anastasievskaja GLKS"

was the basis for the demonstration project database and it is distributed with all distributions of OpenSCADA.

The scope of process control at the level of the programmable logic controller (PLC) is presented by the solution "OpenSCADA into programmable logic controller (PLC)" and "Runtime of the PLC LP-8x81 of the ICP DAS company based on the OpenSCADA".

The scope of the formation of the TP HMI is presented by the solutions: "Library of the graphic elements of the user's interfaces" and " ACS TP of the ball mills "ШБМ 287/410" of the boiler "БК3 160–100 ПТ" ".

One should also note the presence of a number of solutions that are not included in the list above, and made by the developers of the system for the third parties or directly by the users.

Conclusion

OpenSCADA system is ready for practical use in harsh industrial conditions and to perform a wide range of automatic process control tasks.

With the purpose to provide the quality service based on the OpenSCADA solutions, as well as for development in the direction of support the specialized equipment, protocols and interfaces, developers are actively working on the formation the policy of providing the services on the commercial basis. The list of commercial services will include: technical support, consultations and targeted development of extensions, integration and implementation of solutions, and development of customized solutions and products, based on OpenSCADA.

The 0.7.0 version provides the OpenSCADA solutions only for the Linux x86 and x86_64 platforms. To implement the support of other hardware and software platforms, planned by the project, the work will continue. So, for version 0.8.0 it is planned to expand the support of hardware platforms, at least of the ARM platform, as the representative of RISC architecture.

In the appearance of the first industrial version of OpenSCADA actively participated the following people:

- [Roman Savochenko](#): The main works on the projecting, development, documentation's writing and testing.
- [Maxim Lysenko](#): Development of the "ElFigure" VCA primitive, creation of the DB module "PostgreSQL" and translation of the main part of documentation into English.
- [Yashina Ksenia](#): Development of the "ElFigure" VCA primitive.
- *Попкова Ирина*: Translation of the system interface of OpenSCADA and its modules into German language.
- *Almaz Kharimov*: Development of the DCON Protocol module and active testing of the system.
- [Popkov Aleksey](#): Active testing and participation.
- Many other OpenSCADA users, via extensive testing.

Links

Testing protocol of production release: <http://wiki.oscada.org/Works/Tests/release070>