

# OpenSCADA — targeted stabilization

Savochenko Roman Alekseevich  
OpenSCADA Team  
<http://oscada.org>

## Annotation

The project OpenSCADA developing passed to stage of deep stabilization with elements of expanding for existing functions targeted to provide stable and reliable environment for industrial automation and adjacent tasks. The report highlights issues of the current policy of developing solutions for long time support OpenSCADA in cooperation with the system platform environment, and also common achievements of the project for the last time.

## **Introduction**

In the current year the project OpenSCADA silently met the ten years of self foundation by moving to a phase of deep stabilization. The decision in great grade related to activation of the works in adapting to solution, and as result its wide testing with detecting and fixing of the specific problems.

Often by the stabilization more users and developers of programs means the development process stop in facts and then its full stability — the death. That opinion mostly related to the Free Software nature — “Development Just For Fun”. Due the process of stabilization, or support, during long time binds the project to its developer and for its inconsequent development that spread the developer's time with a result here it's few time for same development and fun. That business sort born in more cases to a state of uninterrupted race for new uncertain functions with death in fact for the old version on its state of achieving of the maximum stability and starting work for a new unstable version with sometime crossing the old one at all. This phenomenon is represented too often by more reports about problems and that points to needs for the solution but the developers reaction lack fully.

On the stage of free software building to a distributive the distributors must in fact to do superficial stabilization for last unstable versions and sometime support of previous stable versions, that we can compare to finalization a ware by the file. As a result the solution-distributive releasing not enough stable and user auditory that software left a class of advanced users which are capable to configure, workaround, fix or put up for the problems. On workplaces of users that is acceptable yet but on computers of responsible purpose, for example on consoles of operators and dispatchers of the industrial control, that is unacceptable.

Too bright the problem of lack stability and long support is seen on the desktop environments (DE), were more solution developers don't too often take presented DE for end users, take often for a basis the presented graphical ToolKit (GTK: Maemo, Qt: MeGoo) with the graphical stack and sometime create absolutely new environments with self support (Android). And it is in time when source codes of the solutions allow and for them already done great work!

A good rule and an effective practice is a natural way of the software stabilization in link of the progressive development. That is participation of the target auditory to the stabilization by wide-natural using, feedback and the developer's operative reaction. But the motivation into Free Software for last one is lack and here needs connect the Free Software with an external funding, mostly through offering and/or commercial service providing. In the way of a commercial service and gathering of the offerings and funding of targeted development the project OpenSCADA has gone.

## **OpenSCADA stabilization and development**

The developers team takes considerable attention to stabilization of OpenSCADA, as through operative reaction to notifications and into the target service. In general I want to note with a pride there isn't in that time opened notifications and that operativity of the problems fixing into the source code.

In the stable release of long support 0.8.0 LTS about 150 bugs were fixed and 8 updates were released in the two months cycle. Besides this fixings more changes of simplification and improvement were bore to the stable branch, which don't perform to violation of the back compatibility of the modules' API and ABI.

More attention into the stabilization and the fixing was taken through: analysis of the productivity, usage of the operative memory; revision and simplification of the spare source code; overall effectivity into the work. To fix the measures the special chapter of extended testing, analysis and optimization was created into the knowledge base of the project.

Lets note that most important fixes and changes into the LTS branch:

- *Documentation*: The document "Manual for the creation of modules for OpenSCADA" finished. The document "Quick start" was significant updated and appended and a video-addon was created for the document. Operator-user manual was created on basis of the "AGLKS" model.
- *DB*: Significant improvements, fixes and optimizations for time for documents generation, and a new prescriptions engine implementation.
- *System*:
  - No real task creation from a high-priority thread on some Linux kernels fix, for example it is 2.6.29-rt and it fixed by changing to `TSYS::sysSleep(1e-3)` the waiting by `sched_yield()`.
  - No re-entrant functions replaced: password's hash calculation `crypt()` replaced to `crypt_r()`. Function "getopt" fully replaced to self implemented one `TSYS::getCmdOpt()`. Hidden problematic function `readdir()` replaced to the re-entrant variant `readdir_r()`.
  - System attributes `MessLev`, `LogTarget`, `Lang`, `Lang2CodeBase`, `WorkDB`, `Workdir`, `IcoDir`, `ModDir` allowed for read and write only into the configuration file.
  - To tasks creation function added control for free finished ones but with the object of an undetached task.
  - Adapting to a lost support of the single-char text symbols encoding, checked on KOI8-R.
  - Control to possibility of edition of configuration fields of different objects added, depends from the state "Enabled" and "Started".
  - User objects: An endless recursion fix, on time a string value setting to the Object type. Link to an ancestor of `XMLNodeObj` changed to a direct pointer, for self-blocking and memory leak prevent. `TEValObj` object appended for like to scalars EVAL processing.
  - User API function `SYS.mktime()` added for gets time from epoch 1.1.1970 with its converting from the separated time items.
  - User API function `XMLNodeObj::childIns()` fixed for argument 1 detection as type `XMLNodeObj`.
  - `DAQ.Templates,Protocol.ModBus`: Previous line using as a template at a new one's creation is added. String labels enumeration by `strLabEnum()` added.
  - `SYS.{WorkDir,IcoDir,ModDir},Archive.FSArch,DB.SQLite,Protocol.HTTP,Transport.Serial`: Function `ctrListFS()` added and used for navigation by folders into the configuration fields.
- Subsystem "DB":
  - `DB.SQLite`: Opened transaction closing added in time of the module stopping, to prevent for a data lost after some modules crash at its disable.
- Subsystem "Transports":
  - `Transport.{Serial,Sockets}`: The error number `EAGAIN` processing added into function `write()` for output transports to prevent errors on big data blocks transmission.
  - `Transport.Serial`: Configuration field "Priority" added for a task of the input transport.
- Subsystem "Data acquisition": Memory leak fix into destructor objects `Tval::~~TVal()`. Limit of source text's of procedures expanded to 1000000 symbols. For modules by templates the execution context from DB overloading prevented on the started objects of controller.
  - `DAQ.System`: significant updated and improved for new objects of sources creation at it appearing.
  - `DAQ.SoundCard`: The acquisition rewrote to realtime allow and to control for the sample-rate and the FIFO frames lost. Gathering task, from `PortAudio`, priority setting added. Some notifications about errors added. Some productivity increased.
  - `DAQ.DCON`: Significant changes from Almaz Karimov: analog outputs reading added by a

single method; working of method 1AO (#AA) (blocks NL-1AO, I-7021, ADAM-4021) fixed; method 4AO (#AACN) added for work with blocks ADAM-4024; discrete inputs reading method @AA replaced to \$AA6 for compatibility with ADAM blocks; discrete outputs writing methods 4DO (#AA000F), 8DO (#AA00FF), 12DO (#AA00FFF) added for working with ADAM blocks.

- DAQ.JavaLikeCalc: The assignment operations separation by symbol ',' into the operator “for” of the language is added. The grammar wrong rule "expr OR CONST{" removed. The conditional logical command '||' and '&&' fixed for constants.
- DAQ.ModBus: The wrong setting EVAL fixed for an attribute by the link identifier, at writing error by the link. Support of complex type "String", registers block, was added to DAQ by suffix "\_s" to register "R[I]". Support of complex types "Integer" (32 bit, signed), "Real" and "String" (registers block) was added to the input protocol object.
- DAQ.{BlockCalc,LogicLev,ModBus},Archive.TArchVal,UI.VCAEngine: The navigation by parameter's attributes and same parameters moved to generic function TDAQ::ctrListPrmAttr() and it's used everywhere.
- DAQ.OPC-UA,{Lib,Protocol}.OPC-UA: The OPC-UA protocol's part for client and server moved to LGPL v3 library to simplify this using by other projects. The controller's object simplified to selection for End Points and transports; You can simple set an End Point and a transport will be created from it. For an enabled controller's object you get list of presented End Points. TokenId and SessionId adapted to the complex type NodeId and checked on Ignition OPC-UA server. OPC-UA server fix to processing of merged input messages. The main node attributes processing significant improved and expanded.
- TVal: The archive stopping added at the attribute's object disabling to unlock and finish the actions.
- Subsystem “Archives”: Archives enabling from DAQ parameters and values archive crash fixed, archives processing significant improved. Need periodicity request added to allow access to the value archives of high dimension, by the pointed frame size and periodicity. Archive files destruction prevented into Archive.FSArch through memory buffer of indexes overlap, big messages placing into the flat text archive fixed, at the buffer limit achievement.
- Subsystem “User Interfaces”:
  - UI.{VCAEngine,Vision,WebVision}: Upper threshold of parameters number into primitive “Diagram” increased to 20, splitting scales of the different parameters implemented, trends forming from direct pointed data added through prefixes "data:" and "line:". Indexes definition into “types.h” used for primitives "FormEl", "ElFigure", "Media", "Box".
  - UI.{Vision,WebVision}: Overfull by current values and lost old visible ones fixed for data buffer of primitive “Diagram”, at achievement to the buffer limit. Loading of configuration elements of primitive "ElFigure" fixed considering to demand of all points presence. Function attr{s}Set() of primitive "ElFigure", and "FormEl" into UI.Vision, fully reviewed and made optimal.
  - UI.{Vision,VCAEngine}: The elements view variant “Tree” added to primitive "FormEl". Single-line resources and files navigation added to image/media attributes. Table of "Mime" containers allowed for direct loading but not only for the selected items and in that case "mime" takes from its extension; The "mime" also uses at the filed uploading. Limit to loaded files size set.
  - UI.VCAEngine: Loading/enabling of library widgets and projects by demand added. Limit of archives of primitive “Document” fixed for extension up to 1000000. The exporting symbols moved to file types.h and it's appointed to enable by the visualizers. Parent's address of the included widgets partial changing added at the library widget's or page's parent change, at enabling.
  - UI.WebVision: The periodicity time calculation rewrote and updated to provide more sensibility and to adapt for network channel real properties. Try for load an external JS file "WebVisionVCA.js" added from the current folder. Changes into the JS code: significant cleaning and fixing; checking to full transparency added for the background disable; try of the

asynchronous requests using in the main cycle; primitives "ElFigure", "Media", "FormEl" significant adapted to prevent full context changing on dynamic changes of single properties; images holding in FireFox disabled. The session's object ID renamed to PATH and nodeName() set to string by a separator by path. The option of PNG compression level added for some control to productivity of images forming and the value by default set to "1".

- UI.Vision: Checking to real visibility of widget's part (by the pixel transparently). UI.Vision: Function WdgView::attrsSet() added and used to set group of attributes into single request, to prevent lost of some linked requests in it processing and also to increase performance of working into network. Changes and fixes of primitive "ElFigure": values range definition added and used; enumeration ShapeItem::Type for figures "Line", "Arc" and "Bessie" added; copy of indexes of "Widths", "Styles" and "Colors" added at the figures copy; 5 points splitting of several arcs fixed at it moving by keyboard and mouse; mouse cursor moving together keyboard actions added; arc contour tear at moving from keyboard added; primitive "ElFigure" crash on press to common points 1,2 of arc and point 1 of line fixed; significant code cleaned. Vector index\_array reforming for points 1 and 2 fixed to place only common figures once (exclude the Arc for points 1 and 2 merging accident). Crashed on Lines and Arcs fixed (for previous wrong fix). The primitive "ElFigure" fixed to scale and unscale of widths by moving that action direct to figures imaging.
- UI.WebCfg: Fields "Date+Time" imaging fixed into tables and other places.
- Subsystem "Special": Spec.ModSched: Field "ModAllow" and "ModDeny" opened to edit and to save into the configuration file.

Though more then one year went from the release of the Long Time Support 0.8.0 and there was not any one release into the work branch the works on version 0.8.1 is active and some significant changes included to it, which incompatible for it porting to the LTS version. In general we must to note the work branch also fell to soul of stabilization and the development going in way to improve and to extend old modules and functions, to create new modules and assemblings of the work branch uses for creation new responsible solutions.

Accent more significant improvements into the work branch, which incompatible with the LTS version and which aren't included to it:

- *DB*: Libraries of procedures and visual elements of OpenSCADA significant expanded and appended.
- *SYS*: The script of starting projects of OpenSCADA work stations rewrote — "openscada\_start" to store all the project data into a different folder; to possibility of starting the project pointed into the environment variable "OSCADA\_ProjName"; to possibility of starting the project pointed into the symbolic link name "openscada\_{ProjName}"; to possibility of selection and creation a new project into dialog from "dialog", "kdialog", "zenity" та "Xdialog".
- *SYS*: The build system expanded for install all need program's data (direct through "make install"). Path of installing data changed to "/usr/share/openscada". All paths removed from the configuration file but pointed ones at configuration and building set as by default direct into the core library OpenSCADA. Left configuration variables set into the configuration file through "\*.in" replacing. Targets of creation of the distributive archives unified. The mechanism of modules including to the program core simplified and unified.
- *SYS*: Debug messages forming mechanism changed. The debug messages enable by set the messages level into the system — "Debug (0)" and select need subsystems of the messages generation into tab "Debug" of the main page. The station identifier removed from the generic path of local object OpenSCADA.
- *SYS*: A new mechanism for termination of system tasks execution implemented which based on the signal SIGUSR1 and checking for function TSYS::taskEndRun().
- *DAQ*: The function "Diagnostic" added, which allows into proper tab of the data source select a diagnostic level and enable the debug mode of requests and it reactions by the level "Debug (0)" set.
- *DAQ.JavaLikeCalc*: Productivity of internal executions increased by saving the execution context

and by moving the algorithm of checking for the safe timeout to comparing with an external time variable.

- *DAQ.ICP\_DAS*: In fact fully rewrote to support more modules of series I8k and I87k, and also boards on the bus ISA.
- *DAQ.Comedy*: The new module of supporting wide DAQ-boards on buses ISA, PCI, USB by the library <http://comedy.org>.
- *DAQ.DiamondBoards*: Fully rewrote to support mostly DAQ boards of firm “Diamond Systems”, and also all the three mechanisms synchronous, asynchronous and packed implemented.
- *DAQ.OneWire*: The new module of firstly support of sensors on the bus 1-Wire.
- *DAQ.SMH2Gi*: The new, specialized module of implementing of direct acquisition of data sources of controller SMH2Gi from Segnetics.
- *DAQ.DAQGate*: The possibility of messages of the processed data sources to gate was added.
- *UI*: The functions of getting of icons and resources of graphical interfaces `icoPresent()`, `icoGet()` and `icoPath()` merged to single function `icoGet()` and adapted to process several folders. The folder of icons set to "icons/;@oscd\_datadir\_full@/icons/".
- *UI.{VCAEngine,Vision}*: Number of allowed parameters of the primitive “Diagram” increased to 100. The view “Button” of the primitive «Form elements» expanded by modes: “Standard”, “Selection”, “Menu”, “Load”, “Save”.
- *UI.VCAEngine*: Documents generation into the primitive “Document” moved to a different task to prevent of main one blocking at process. On the document processing the result updates for each 5 seconds. The attribute "process" added to control the process end.
- *UI.Vision*: The possibility of using WebKit for the primitive “Document” rendering.

## **General stabilization of the execution environment — Linux distributive with OpenSCADA**

OpenSCADA — is an environment of working with dynamic data into realtime, itself works into OS Linux environment and also into a graphical stack of a desktop environment with using here different free libraries. Then stability of the complex end solution depends from coordinated and stable work of the program's parts in whole. By that reason the OpenSCADA team has significant attention to stability of the OpenSCADA execution environment in way to creation a stable packages repository based on ALTLinux T6 distributive and to participate into stabilization of some parts of the environment.

Into the introduction we mentioned about actuality of the problem of stability and long term support of the Linux free software environment especially into the responsible tasks. But there observed positive trends of switching to the LTS versions releasing scheme for some programs and it distributives in whole:

- “RedHat Linux” (Centos, Scientific Linux) — releases stable LTS distributives targeted to using in servers.
- “Linux kernel» — clearly defined splitting of the LTS releases and its long term support.
- “Ubuntu” — for a list of the products the stable distributives released and supported.
- “Trinity DE” — support and development of KDE3 code base, it has a clearly defined LTS branch.
- “KDE4” — passing to the LTS version 4.11 planed, but there also observes the version 4.12.
- Web-browsers and number of some other applications.

But into the OS Linux environment yet stayed number of components the development policy of with doesn't mean LTS and that result is problems with hardware integration. The problem stays especially for XOrg.

The OpenSCADA team is direct participating into free projects of the SCADA-system environment:

- “Linux kernel” — for a number of tasks of the programming logical controllers (PLC) performs a thin adapting and building of realtime kernels in base of kernel 2.6.29 and PREEMPT\_RT patches.
- “TDE” — to provide the adjusted, undemanded and developed desktop environment TDE there it was reanimated, fixed for critical errors and ongoing support, the LTS releases building provide.

## **Conclusion**

The project OpenSCADA developing mostly directed now to target stabilization and extending, in which reason the long term plans significant reviewed. The date of next long term support release 0.9.0 LTS moved to Mai 2014, and life time of the current long term support release 0.8.0 LTS longed to the term.

Besides to priority of stability the development also continued. It is proved by significant extensions and simplifications into the project's work branch, that isn't allowed to move to the stable branch but it is significant improves the program in general. The development process was targeted and for need tasks prepared a pool for the project's contributors and for preparing a developing perspective in general. Some resume of the developing planed by releasing of the first work branch 0.8.1 up to the year end.

After the work branch 0.8.1 releasing planed a significant update of the representation (logotype, design, ...), opening a program of the targeting funding to the project's attract more developers which for now isn't too more but the project's increasing interest allow tell us about perspectives and significant potential of that measures into the stability increasing, improvement and adaption to different platforms. After the 0.8.1 release there also planed to start of the planed OpenSCADA adaption and buildings for alternative program platforms.

The OpenSCADA team thanks you for your attention to the project and invite you to participation and cooperation into tasks of various and overall automation!