OpenSCADA 0.9 LTS
Savochenko R. O.
OpenSCADA Team <oscada@oscada.org>

The announce of the new LTS version 0.9 of the project OpenSCADA — an open Supervisory Control and Data Acquisition system. The new stable version is the result of six years of development, implementation of various solutions, stabilization and exploitation of OpenSCADA in its Work branch. But all this time, updates for 0.8.0 LTS continued to be released and what would happen for the announced 0.9 LTS.

1 Introduction

The release of OpenSCADA, an open SCADA (Supervisory Control and Data Acquisition) system, version 0.9 is a stable industrial release of the long-term support (LTS).

The main purpose of the release is to provide an updated and stable platform for building integrated automation systems and other adjacent solutions for the community of users and developers of the free software. Also, the release is intended to become an actual and solid foundation for commercial solutions building.

The release is the next release of the stable branch for which, over a long period of time, there provided the technical support from developers and releasing of updates in the form of public builds for the main and stable environments of Linux, as well as operational builds for holders of packages of the technical support. The life cycle of the previous release 0.8.0 LTS stops from its latest update, right before the first builds of 0.9 LTS packages.

It was at this version that the final transition to the Work/LTS development scheme took place, that is, the development is carried out within the working version and on its basis periodically released the stable releases, which, in turn and in parallel, are updated by the back-porting compatible changes from the working version. The initiating transition took place in 2013, when the current stable version was recognized as 0.8.0 LTS and the working as 0.9 Work. Currently, 0.9 LTS is announced as a stable version, and 1 Work as a working version.

In general, the new stable version is the result of six years of development, implementation to various solutions, stabilization and exploitation of OpenSCADA in its work branch, which is currently being released as a stable one. All this time updates for 0.8.0 LTS continued to be released and what will be done for the announced 0.9 LTS.

1.1 Overall information

The OpenSCADA project was founded by Roman Savochenko in 2003 as a

free implementation of the SCADA system or the Human Machine Interface (HMI), based on its thorough projecting during 2002 and the experience of using and developing a commercial SCADA system to this.

SCADA or HMI systems are generally designed and used to carry out human operational control over the work of complex and responsible technological equipment and processes of various manufacturing enterprises.

At the time of announcement of 0.9 LTS, OpenSCADA is a developed SCADA/HMI system, which is quite widely used as in the direct destination and, due to its flexibility, in many related industries and it can definitely be called more general — a dynamic system of working with data in the real time.

The number of the stable version of OpenSCADA is less than one only because its purpose was claimed multiplatform, to achieve which in the planned volume is scheduled for the next stable release number 1. In general, the objectives of OpenSCADA are:

- **openness** — mostly GPLv2;
- **scalability, flexibility, extensibility** — modularity, multithreading and internal dynamism;
- **executive redundancy** — reservation;
- **accessibility** — opened source texts; multilingual; dynamic multilingual; automatic build for archives, packages, live disks, ...; builds and executes on wide age Linux environments, from 2002 (2009-ALTLinux 6, LP8x81, Fedora 12; 2012-Debian 7) to modern ones;
- **reliability** — practical application, quick problems solving;
- **security** — permissions distribution, SSL;
- **multiplatform** — x86_32, x86_64, ARM, Web, Android, QNX (preadapted), MS Windows (scheduled);
- **united, unified, user-friendly, dynamic and advanced user interface** — Qt, Web, transparent and multi-level remote dynamic control;
- **wide range of the data sources** — unified ones, DAQ boards and low-level buses, created into the environment of OpenSCADA.

1.2 Application

For actual implementations, where at least one implementation is known and with a short list of which from the project participants you can read through this link, belong:

- ACS TP (SCADA/HMI) or telemechanics systems — the main direction and there are many implementations;
- dynamic models, imitators and training apparatus of technological pro-

cesses into the real-time;

- machine tools and industrial robots;
- agricultural dispatching and control systems, poultry-yards;
- embedded and mobile systems — environments of execution Programming Logical Controllers (PLC), robots, ...;
- server equipment monitoring;
- smart houses and home automation.

With some restrictions and improvements, largely in the user's internal environment, OpenSCADA can be used in the following areas:

- enterprise resources management (ERP);
- Geo-location and location tracking;
- trading systems;
- medical diagnostic systems;
- accounting and bookkeeping;
- billing systems.

2 Results of the previous release 0.8.0 LTS

The release 0.8.0 LTS was released in April 2012 and during these six years it received 20 updates, which, in total, corrected over 500 errors and added many improvements that do not violate the compatibility of library databases and configurations.

The configuration and library databases of 0.8.0 LTS were generally frozen due to large incompatible changes at 0.9 Work, for OpenSCADA launch methods, and the imperfection of distributing the library databases at the release date of 0.8.0 LTS. Therefore, the upgrade and transition to 0.9 LTS will be non-trivial, although 0.9 LTS provides everything possible for simplification this procedure. Updating 0.9 LTS to the planned future 1 LTS will no longer be so complex and in fact can only turn into a formal change in the version of the working branch, since now it is planned to upgrade all, including the library databases.

3 Planned tasks of the release

The development of OpenSCADA, after the previous LTS version and within the Work branch, was mainly through deep stabilization and through the practical adaptation with elements of the expansion of the existing functional, aimed to provide a stable and reliable environment for the industrial automation and related tasks, and therefore, there was no clearly defined plan. But three years before this release, such a plan appeared and made the following tasks, as seen from the general development plan:

- Full revising the main documentation and preparing the release announce.

- Adaptation for work on the software platform "Android".
- OpenSCADA knowledge and documenting WIKI-resource moving to a new engine with the structure unification for multi-languages into the priority: English, Ukrainian, Russian; and the off-line documentation generation at it changes.
- Main Web-modules revision, actualization and some expanding.
- OpenSCADA expanding and adapting for direct working with the low-level buses and their devices like to 1Wire, I2C. Implementation for "the Smart House" in my own apartments.
- Automation Linux distributive of the project OpenSCADA formal creation and documenting.
- Creating the Automatic building system of the OpenSCADA packages.
- Moving the OpenSCADA server infrastructure to its own equipment and Internet channel.
- DAQ.OPC_UA: Simplification, features rising and protocol's code moving to a separate library on LGPL v3.

These tasks are done and some details about them are given below.

4 System-wide properties

The new stable version of OpenSCADA has gained significant system-wide extensions, increased stability in work and productivity, and also received significant improvements to the graphical user interface and its environment, such as: advanced configuration, documentation, accessible directly from the program (offline and online), and which significantly revised and updated.

The common part of documentation of the project has been moved to a new Wiki, based on MediaWiki, and a significant amount of this part has been revised and translated into three languages — English, Ukrainian, Russian. For the pages transferring a converting procedure from the WackoWiki engine dialect to MediaWiki was created which was completely written in the internal programming language of OpenSCADA, and is used to transfer large volume of the old Wiki. Format the offline documentation has been changed from static PDF-files, that were not updated after the previous LTS version, to HTML-files which dynamically generated from the actual knowledge base (Wiki) of the project, and have actual cross-links between pages and links on the online-documentation, for exclusively external materials. The offline-documentation is also generated by a specially-written procedure in the internal language of OpenSCADA, which, along with the procedure of the Wiki-dialects converting and complex testing the release OpenSCADA, is a bright sign of power and the current level of development of the internal language of OpenSCADA.

Within the old Wiki, after the release 0.8.0 LTS, some documents were added

and updated, and with the transferring to the new Wiki, in addition to revision and translation of the main part of the documentation, the overall unification of its structure was made for reasons of: multilingualism (English as a primary language), the logic of the organization, the convenience of translation and the ability to remove duplicate articles from the official site, which at the moment are simply used with the Wiki. We will separately note the significantly expanded main documents:

- Quick start;
- Program manual;
- User API of OpenSCADA;
- FAQ and How to ...;
- Creating OpenSCADA module.

The original language (English) was completely revised in original messages of the program and mainly for the main articles of the documentation-Wiki OpenSCADA, which is currently making OpenSCADA an adequately favorable for an audience that does not understand the original language — Ukrainian or Russian. And, along with the full realization of the dynamic translation mechanism, it is possible to build on the basis of OpenSCADA the dynamically multilingual user interfaces that you can see, in particular, on the publicly available Web-interface of the dynamic simulators of TP: AGLKS, Boiler.

The Work version, based on this stable, for the first time defined the OpenSCADA projects conception and implemented a command line script for launching and creating OpenSCADA projects. Immediately before the release of this version, the concept of the OpenSCADA project was finally assigned to the folder with the data of the separate project and the configuration file of OpenSCADA, and the implementation of the project manager was integrated directly into OpenSCADA. Consequently, this LTS version has a developed concept for the project manager that allows to flexibly work with them and eliminate the dangerous possibility of multiple launches with the general data of one project.

The Work version also introduced modification of the modules versions at modification in the code of the module and just before they are uploaded to the source repository, and therefore, the versions of the OpenSCADA modules of this release clearly reflect the overall level of development and stability.

In the process of implementing works on the Work branch, based on this stable, there were introduced the repositories of Linux distribution packages with the OpenSCADA builds, which until now were only provided as separate packages. That significantly simplified the deployment of OpenSCADA and keeping it up to date. Then automatic builder of these packages was created, which currently has up to 100 targets, and which greatly simplified the release of updates for both branches, that is the Work and this stable. Therefore, this LTS version provides

packages collections for the main Linux environments and the entire history of public updates will be saved.

To the OpenSCADA packages collections there also provide builds of the live disks of the quick acquaintance and deployment of OpenSCADA together with the system environment. Currently, they have received a formal background in the form of the Linux Automation Distribution of OpenSCADA.

OpenSCADA builds and packages are accompanied by a number of open and free materials of the internal environment of the program, that is, the development of the data acquisition and processing layer, graphic representation elements and whole-complex projects of the TP simulators. These materials are provided as SQLite database files and include:

- Functions libraries (OscadaLibs), "LibsDB/OscadaLibs.db" — contains all the development of the OpenSCADA project in the data acquisition and processing layer, including data source elements of the user protocol.
- VCA: Main libraries (vcaBase), "LibsDB/vcaBase.db" — contains main elements of the graphic representation and elements of the mnemonic schemes of the OpenSCADA project.
- VCA: Tests (vcaTest), "LibsDB/vcaTest.db" — contains test elements of the graphic representation primitives.
- VCA: Library of electrical-elements of the mnemonic schemes of the user interface (vcaElectroEls), "LibsDB/vcaElectroEls.db" — contains elements of the graphical representation of the electrical schemes.

Hosting of the project in general and materials of 0.9 LTS in particular were transferred to own project server, where additionally were deployed: the demonstration Web-interfaces of OpenSCADA simulators, the project of server monitoring and smart home based on OpenSCADA and builder of the packages of the OpenSCADA repositories.

4.1 Internal

The resolution of internal data of the integer type of the OpenSCADA environment is increased to 64 bits. In general, the internal data of OpenSCADA, with error value reservation (EVAL) for each, is unified by common types: logical, integer, real, string, and object. Which primarily concerns the data sources.

To the OpenSCADA core, its own protocol and all nodes that work with remote stations OpenSCADA; added the ability to "raise" the nodes of OpenSCADA that are located behind other nodes and, as a rule, in another network. What generally allows you to centrally manage the OpenSCADA network at any level of the hierarchy.

4.2 Improvements and adaptions to the different platforms

This version of OpenSCADA has gained in-depth support and ability to adapt to different platforms. This was mainly due to the adaptation to work on the Android software platform and the restoration of the building and work with ucLibC, and that it is planned to be used for further adaptation to work on QNX and MS Windows software platforms.

Working on the Single-board PCs was expanded by the Raspberry Pi and Orange Pi boards.

Support of the Linux smartphones of Nokia was appended by the last one based on MeeGo 1.2, that is Nokia N9. Or it was the renewing of support for the Nokia N950.

In addition to direct work (natively) on different platforms, the Web-interface has been significantly expanded, which currently implements all the general features of the concept of the Visualization Control Area (VCA).

4.3 Optimization, stabilization and performance

Significant stabilization of the OpenSCADA core, and the overall program, has been achieved through the unification of internal resources control and the expansion of capabilities of the user debugging. In general, the user diagnostic and debugging expanded:

- general enabling-disabling of the debugging and target control of the debugging nodes;
- special debugging of the controller objects of the data sources;
- special debugging of execution of the VCA projects;
- in-depth statistics formation for execution of the dynamic objects OpenSCADA, such as: controller objects of the data sources and their parameters, incoming and outgoing traffic, VCA sessions with detail up to widgets;
- logging of ingoing and outgoing traffic of the transports.

Almost all of the OpenSCADA modules have been subjected to deep and comprehensive stabilization and many have been optimized, of which especially should be noted:

- All DB modules — added processing and verification of database or DBMS errors, and issuance of messages about these errors in the case of user uploading and recording; increased productivity (up to ten times) DBs that support SQL, through the implementation of the pre-loading mode of the scan query;
- Calculator on the Java-like language (DAQ.JavaLikeCalc) — increased

productivity by: saving the context of the function execution, constants pre-loading and direct access to string.

- Archiver to DB (Archive.DBArch) — significantly optimized for recording and reading of the database, that is: group recording of multiple archives to one table and block reading (multiples of ten) in one query.
- All transports — increased for the overall productivity.
- Operation user interface (WEB) (UI.WebVision) — sensitivity of the interface increased by using asynchronous queries in the overall updating cycle.

Reliability of the redundant station on an integrated solution scale, more precisely the preservation of the history data, was enhanced by the extension of the redundancy mechanism, which potentially involves reserving any subsystem and currently implements the subsystems "Data acquisition" and "Archives-History".

And, for the program as a whole, a series of formal comprehensive tests was executed, that was pre-expanded in the internal integrated testing procedure of the OpenSCADA release. On the basis of these tests, several bugs were detected which were fixed.

4.4 Data acquisition

Given the key role of the data acquisition in this type of software, this feature, in the person of the subsystem "Data acquisition" and its modules, has received significant improvements, of which particular attention should be paid: shifting emphasis on the extending supported data sources from implementation of individual modules of the subsystem "Data acquisition" in the system language "C/C++", to their implementation in the OpenSCADA environment and in its internal language — logical level of OpenSCADA. That is, at the logical level of OpenSCADA, can be and implements everything that: uses the network to access the data, does not require to use specific libraries and functions, and is not very complicated. Currently there implemented in this way:

- Sending SMS (SMS) and Email (SMTP).
- Uninterruptible Power Supply (UPS), as a data object with attributes of values.
- Simple sensors:
  - Елемер TM510x;
  - EDWARDS TURBOMOLECULAR PUMPS (SCU750);
  - Sycon Multi Drop Protocol (SMDP);
  - Power supply of the turbo-molecular pump (TMP-xx03);
  - Temperature measurement IT-3 (IT3);

    - IVE-452HS-02;
    - OPTRIS CT/CTL;
    - CTR 100, 101.
- Computer of the heat-counter VKT7.
- IEC-60870-104.
- Test implementations and examples: DCON, OWEN.
- Bus "One Wire" in help of {DS9097,DS9097U} (1W_{DS9097,DS9097}) for chips: DS1820, DS1820/DS18S20/DS1920, DS1822, DS2413, DS2408, DS2450, DS2438.
- Bus I2C: PCF8591, PCF8574, BMP180, DS3231, AT24C{32|64}.
- Generic ports IO (GPIO): DHT11,22 (AM23XX). GPIO|I2C: 1602A(HD44780).

Given the increased role of the logical level of OpenSCADA, there has been an increase in the requirements to the flexibility of the data model of the data source, which was satisfied by:

- completion of the coverage of the internal data model by all the functions that are specific to the areas of application;
- adding hierarchy to the parameters of controller objects of the data sources;
- giving the possibility of an arbitrary and dynamic formation of the data model — a set of attributes of parameters.

The OpenSCADA internal programming language has made significant improvements and, at the moment, satisfies all the requirements of the areas of application OpenSCADA. Many of these enhancements came in the previous version of 0.8.0 LTS, and some could not get there because of the backward compatibility or lack of stability at that time. Of these significant improvements, it should be noted again: increasing the resolution of the data of the OpenSCADA internal environment of the integer type up to 64 bits and preserving the context of execution of the internal procedures.

4.5 Graphical environment

In general, the graphical environment has been significantly developed, and the main one has been the extension of the module of starting the Qt-interface, as the basis for the rest local interface modules. First of all, this module and the core of OpenSCADA are adapted to allow the Qt-library to run in the main thread of the program, eliminating many of the problems associated with execution in the non-main thread, and also provides work with version 5 of this library. Secondly, this module took on the role of the selecting interface of the OpenSCADA projects

when it launches and switches, as well as creating new ones. And thirdly, because of its primacy to launching Qt, it received the function of controlling the appearance of the program regardless of the graphical environment and the possibility of its launch-closing in the system tray. In general, it has made of the possibility to customize OpenSCADA to personal user requirements and to adapt it to very specific environments such as Android.

Notable improvements have been made to the Qt and Web configuration modules, which are particularly noteworthy:

- Configurator-Qt (UI.QTCfg): querying remote stations in a separate thread from the Qt-thread made it more convenient and more predictable and even, along with one update, got into the previous stable versions of 0.8.0 LTS.
- Configurator-WEB (UI.WebCfgD): in general, has been completely updated to the interface, which is now more dynamic, user-friendly and can be extended by themes.

Notable improvements were also made to the Visual Control Area (VCA), consisting of all its modules:

- VCA Engine and the visualizer modules at all (UI.VCAEngine):
  - allows to carry out a full-hot development, that is — editing the VCA project at the time of its execution by sessions;
  - the original interface messages are fully corrected, grammatically and spelling correct for the English language;
  - provided a number of additional types of the primitive "Form elements", "Diagram" and extended the primitive "HTML" by "HTML" displaying;
  - the mechanism of the widget specific attributes to the visualizer was provided, which allowed to reveal and use their individual properties;
  - new-flexible mechanism of user notification about extraordinary events in the process controlled by the system, which provides the possibility of free formation of custom announcers with the necessary properties, such as: mono-tone signal, synthesis of speech.
- Visualizer-Qt (UI.Vision):
  - noticeably improved performance of execution the remote interfaces and it is ensured for queries to the remote station in a separate thread from the Qt one, which facilitated remote development with parallel execution of the project;
  - forming the primitive "Document" can be performed at help WebKit.
- Visualizer-WEB (UI.WebVision):

- unified, optimized and expanded by using CSS3;
- provides the implementation for all elements-primitives of the unified interface;
- performs the scaling to allowed space of the browser window;
- increased for the working productivity and the interface sensibility, through using only the asynchronous mechanism at updating.

All Web modules, in general, have the opportunity to change the view by topics, through the system-wide interfaces of the protocol HTTP and its mechanisms for query processing and response forming. Also, they all have the support of dynamic translation of the interface, which is especially relevant for multi-user Web-interfaces, and what you can see on the demo Web-interfaces of the OpenSCADA models: AGLKS, Boiler. There is also a mechanism for the distribution of access to the pages, which, in particular, allows to differentiate access to the Web-modules in general.

5 New and significantly updated modules

The new version has added new modules and significantly updated a number of present ones:

- New modules added:
  - MMS(IEC-9506) (DAQ.MMS) — a module of supporting for data exchanging at the protocol "Manufacturing Message Specification (MMS, IEC-9506)".
  - Comedi (DAQ.Comedi) — a module of supporting the data sources of the real time (library "Comedi"), which are based on the data acquisition boards of different manufacturers and are installed on the buses: ISA, PCI, PCMCIA and USB.
  - SMH2Gi (DAQ.SMH2Gi) — a module of implementation of access to hardware modules of the data sources of PLC Segnetics SMH2Gi,SMH4 for "MC", "MR", and also for interaction with the original environment "SMLogix".
  - Fastwel IO (DAQ.Fastwel) — a module of data exchanging with the hardware modules of Fastwel IO.
  - FT3 (АПСТМ) (DAQ.FT3) — a module of data exchanging with the PLC АПСТМ, АСДКУ, СУАП.
  - GPIO (DAQ.GPIO) — a module of accessing to GPIO of the single-board PC like to Raspberry Pi, Orange Pi and other.
- Calculator on the Java-like language (DAQ.JavaLikeCalc) expanded for: support for internal functions, dynamic translation of messages and many other functions of the user programming interface in general.

- Data sources gate (DAQ.DAQGate) expanded by reflecting the messages associated with the selected data source.
- Data acquisition of OS (DAQ.System) expanded for the data sources: "File System", UPS, QSensor and the ability to separate slow sources from fast ones.
- ModBus ({DAQ,Protocol}.ModBus) expanded for support string, as a sequence of values of the registers.
- Client DCON (DAQ.DCON) significantly expanded for specific modules support.
- OPC-UA (DAQ.OPC_UA) significantly expanded for Publishes and to support for "Chunks" into client part of the service, the specific protocol code is separated from the library.
- Equipment of ICP_DAS (DAQ.ICP_DAS) significantly expanded, and in fact completely rewritten, for support all available ICP_DAS data acquisition boards for the series I8k, I-87k, on the ISA bus and unified types for standard modules of the I7k series.
- Siemens S7 PLC (DAQ.Siemens) significantly expanded its own implementation of ISO-TSAP.
- Diamond data acquisition boards (DAQ.DiamondBoards) significantly expanded, but in fact completely rewritten, to support all existing data acquisition boards from Diamond Systems.
- AMR devices ACKO (DAQ.AMRDevs) appended of support the counter Kontar (MZTA).
- All DB modules significantly expanded by the dynamic translation.
- Archiver to DB (Archive.DBArch) expanded by archiving several archives into one table and restoring the list of archives from information in the database.
- Archiver to FS (Archive.FSArch) added for supported the intermediate types "Int16", "Int32", "Int64", "Float", "Double", and appended by an absolute limit to the size of the archive on the disk.
- All transport modules extended for the pooling mode of the input transports and protocols.
- Sockets (DAQ.Sockets) expanded by the RAWCAN bus support and initiated connections of the input transports.
- Serial interfaces (Transport.Serial) expanded for the low-level bus I2C support; special user functions of the serial interface: "sendbreak", "TS", "DR", "DCD", "RI"; advanced control RTS for RS-485.

- Security Sockets Layer (Transport.SSL) expanded by support: TLSv1.1, TLSv1.2, DTLSv1.
- Own protocol of the program (Protocol.SelfSystem) expanded for hierarchical and multi-level targeting of requests to external hosts.
- HTTP (Protocol.HTTP) expanded for adapting to the user interfaces in the system dialogs and providing a generic API for building HTTP interfaces for both the user and the modules behind it.
- Program configurator (Qt) (UI.QTCfg) OpenSCADA control interface requests are moved to a different thread and improved for implementation of the control elements.
- Program configurator (Dynamic WEB) (UI.WebCfgD) expanded with a new design using CSS3 and the features of modern WEB browsers.
- Conception and the Visual Control Area (VCA) (UI.VCAEngine) expanded by: support for specific visualizer attributes of the widgets, implementation of the new-flexible mechanism of the user's notification, background (in a separate thread) execution of the document forming task, increase of the number of trends in one frame to 100 and support of the logarithmic scale for the primitive "Diagram".
- Operation user interface (Qt) (UI.Vision) expanded: to work on the network through the visualizer server; implementing a group of attributes specific to this visualizer; realization of the views "Tree", "Table" and extension of the type "Button", of the primitive "Form elements"; implementation of the view "XY" of the primitive "Diagram"; significant refactoring code of the primitive "Elementary figure".
- Operation user interface (WEB) (UI.WebVision) expanded for using CSS3 and the capabilities of modern browsers, in particular: added zoom to the available window browser space, implemented all the primitives of the unified interface.
- Functions library of the system API of the user programming environment (Special.FLibSYS) expanded by the functions and objects: "floatExtract", "md5", "tmSleep", the object "IO"; significantly expanded for the actual functions and objects: "dbReqSQL", "FFT", "strParse", "strDec4Bin".

6 Conclusion

On the way to the new release 0.9 LTS of industrial use, much work has been done to stabilize, expand functionality, and expand adaptability to work on alternative platforms. All this in general has further expanded the scope of full use of OpenSCADA at all levels of industrial automation systems and related areas of automation and automatic.

LTS versions of OpenSCADA are not blank, they are really supported all the time until the next LTS version, and support for this version will be further expanded with the service updates. Also, the emphasis and implementation policy of the LTS version, which was previously recommended to upgrade configurations with frozen library databases, will be shifted to the using priority on a wide range of new solutions.

In the emergence of the new industrial version of OpenSCADA 0.9 LTS took part:

- Roman Savochenko: the main volume of work on the development, testing, building, documenting and translation of the program and documentation in three languages.
- Maxim Kochetkov: implementing the modules DAQ.Fastwel, DAQ.FT3; expanding the module Transport.Sockets by support RAWCAN and the module Transport.Serial by extended control RTS for RS-485.
- Arsen Zakojan: implementing for support the electricity counters "Mercury 200" and "Mercury 230".
- Ruslan Yarmoliuk: implementing for support the electricity counter NIK 2303.
- Almaz Karimov: expanding for the module of the protocol DCON implementation.
- Arcadiy Kisel: implementing for support the I2C temperature, barometric pressure and humidity sensor BME280.
- Constantine (IrmIngeneer): support of the OpenSCADA build on the Linux distribution Gentoo.
- Sergij Doroshka: previous adaptation of OpenSCADA to work on QNX.
- The organizations which caused to the most significant OpenSCADA improvement through it integration to own control systems:
  - Proviron Holding NV: purchasing the technical support packages for the general support, fixing and improvement the Siemens ISO-TSAP(ProfiNet) protocol implementation, some workouts with Raspberry Pi, 2014, 2016-2018.
  - Laboratory of the vacuum technologies: "Vacuum technological facility", 2011-2018, and financing equipment for the OpenSCADA server, 2014
  - Optima: "Automation System of the Metropolitan", 2015-2016.
  - Vector: purchasing the technical support packages for fixing and improvement the OPC-UA implementation to work with different OPC-UA clients and servers, 2015-2016.

    - Kramatorskteploenergo+DIYA: "ACS of the ball drum mills BDM 287/410 of the boiler #8 of BKZ 160–100 PT", 2015, and "ACS of Phosphating, Amination and Hydrazine of boiler BKZ 160–100 PT", 2014. For PLC there used ICP-DAS LP-8781 with OpenSCADA in a role of environment of execution of PLC.
    - Hartron: "Reactivity Monitoring System (RMS) of the Subcritical Nuclear Facility (SNF)", 2013-2015.
- Many other organizations and individual users who wished to remain anonymous, through the purchase of technical support and services, constructive feedback on implementation and exploiting.

Further development efforts will focus on:

- Completion of revisions of the basic documentation, mainly for modules and libraries.
- Clear definition and update of the policy of providing commercial services around the OpenSCADA project:
    - view and update the terms of package of the technical support;
    - policy and organization of building updates to the stable branch: preserving the history of all public updates and organizing service ones;
    - distribution of exclusively commercial builds with public demonstration on an example package for Android — development and formation of a mechanism for monitoring the term of the package of technical support and demonstration mode.
- Mastering and adapting to Enterprise Resource Planning (ERP), focusing on:
    - create a resource management interface on the project server;
    - creating a common interface for tasks control and their funding, with the organization of developers to engage in the implementation of these tasks and technical support.
- Expand the application functions in the areas: "Smart House", "Home Automation" and "Custom Robots".
- Adaptation to work in the environment of the operating systems QNX and MS Windows.