OpenSCADA 0.7.1

(Release notes)

Table of contents

Introduction

This release is a planned one of a working version of OpenSCADA, which are usually made at intervals of 3-4 months. The main purpose of this release is the implementation of commercial service on the OpenSCADA basis and adaptation to the ARM hardware. In addition to the main purposes within the limits of this release a great work for stabilization, cleaning the source code, as well as for the full empowerment is made.

This document is the processing (compilation) of <ChangeLog> document of the OpenSCADA system of 0.7.1 version, which is designed for briefly and clearly covering the new features and changes 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: http://oscada.org/websvn/filedetails.php?repname=OpenSCADA&path=%2Ftrunk%2FOpenSCADA %2FChangeLog .

The key features of this version are:
- Implementation of planned tasks.
- Optimization, increasing of stability, sustainability and system performance.
- Improvement and stabilization of the graphics subsystem.
- Formation and stabilization of user programming API.
- System-wide expansions.

New and updated modules are:
- *Archive.FSArch (1.4.5)* - Significant stabilization.
- *DB.MySQL (1.7.0)* - Stabilization. The ability to specify connection timeouts is added.
- *DB.SQLite (1.6.3)* - Stabilization.
- *DB.FireBird (0.9.6)* - Stabilization.
- *DB.PostgreSQL (0.9.1)* - Stabilization.
- *DAQ.DiamondBoards (1.2.2)* - Stabilization.
- *DAQ.System (1.7.3)* - Stabilization. Dependence on the library "Libsensors" is made optional.
- *DAQ.BlockCalc (1.4.1)* - Stabilization. The syntax highlighting is added.
- *DAQ.JavaLikeCalc (1.9.0)* - Stabilization. The syntax highlighting is added. The adaptation to the ARM architecture hardware is made. The implementation of regular expressions and a number of extensions of user API is added. The ability to write characters in a string using octal "\041" and hex "\x21" numbers is added.
- *DAQ.LogicLev (1.1.3)* - Stabilization. Certain extension.
- *DAQ.SNMP (0.6.0)* - Stabilization. Implementation of the full support of the client-side of protocol. The support for all types of values is added. The support for writing values is added. Errors handling is improved. Additional connection configuration options are added.
- *DAQ.Siemens (1.2.4)* - Stabilization.
- *DAQ.ModBus (1.1.2)* - Stabilization. The syntax highlighting of the attributes' list is added. Some improvements.
- *DAQ.DCON (0.5.0)* - The module is completely rewritten for support of mixed modules by selecting the query command.
- *DAQ.ICP_DAS (0.7.3)* - Stabilization.
- *DAQ.DAQGate (0.9.2)* - Stabilization. The parameters' cache for their initialization during the autonomous operation is added.
- *DAQ.SoundCard (0.6.2)* - Stabilization.
- *DAQ.OPC_UA (0.6.1)* - Stabilization. he syntax highlighting of the attributes' list is added.
- *DAQ.BFN (0.5.0)* - New module.
- *Transport.Sockets (1.4.6)* - Stabilization.
- *Transport.SSL (0.9.6)* - Stabilization.
- *Transport.Serial (0.7.2)* - Stabilization. During initialization of the port the previous settings are used and an opportunity to omit some of the settings is implemented.
- *Protocol.ModBus (0.6.2)* - Stabilization. Some improvements.
- *Protocol.OPC_UA (0.6.1)* - Stabilization.
- *Protocol.HTTP (1.5.1)* - Stabilization. The support for HTTP variable "Transfer-Encoding =

chunked" in the outgoing transport is added.
- *Protocol.SelfSystem (0.9.4)* - Stabilization.
- *Protocol.UserProtocol (0.6.1)* - The syntax highlighting is added.
- *Special.SystemTests (1.5.1)* - Some improvements.
- *UI.QTStarter (1.6.1)* - Stabilization. Adaptation for the ARM.
- *UI.QTCfg (2.0.0)* - Stabilization. The group request to the control interface is used to optimize the remote access to OpenSCADA stations. Adaptation for the ARM. The function for the manual change the height of the elements of the text and tables (grabbed the bottom edge and dragging) is added. The multi-level syntax highlighting is added.
- *UI.WebCfg (1.5.5)* - Stabilization.
- *UI.WebCfgD (0.6.6)* - Stabilization.
- *UI.VCAEngine (1.1.0)* - Stabilization. The tooltip for the attributes of widget's primitives is added. The syntax highlighting to the text attributes of widget's primitives is added.
- *UI.Vision (1.1.0)* - Significant stabilization. Notable improvements. Adapted for the ARM. The display and restoration of lost communication are added during the remote execution of the session. The separate full-text editor for text attributes is added. The tooltip for the attributes of widget's primitives is added. The syntax highlighting for text attributes of the widgets primitives is added. The dialog of the set properties adjustment of separately selected figures is added.
- *UI.WebVision (0.9.1)* - Significant stabilization. Small improvements.
- *UI.WebUser (0.6.1)* - Stabilization. The syntax highlighting of the procedures is added.

1 Implementation of planned tasks

In accordance with the plane of the release there were made the following tasks:

- *Formation of the provision of commercial services based on OpenSCADA.* - The concept is built and the mechanisms are created to provide commercial services by the developers based on OpenSCADA (http://oscada.org/en/services).
- *Adaptation of OpenSCADA to work on the ARM hardware platform.* - The building, adaptation and full testing of OpenSCADA on the ARM architecture is made on the N800 Internet tablet of the Nokia company (http://wiki.oscada.org/Works/Tests/ARM (RU)). As well as private activities for adaptation:
    - *SYS*: Functions' prototypes TMess::put(), TMess::get(), TCntrNode::grpSize(), TCntrNode::grpId(), TCntrNode::grpAt() are changed to correct the usage of an unsigned type "char" on the ARM as a signed.
    - *SYS.TVariant, DAQ.AMRDevs, DAQ.OPC_UA*: The not aligned access to data is fixed.
    - *SYS.TCntrNode*: The usage of an unsigned type "char" on the ARM as a signed is fixed.
    - *Archive*: Functions' prototypes TArchiveS::messPut() and TArchiveS::messGet() are changed to correct variable of the message level as a signed number.
    - *DAQ.JavaLikeCalc*: Virtual Machine is rewritten for direct using the structures of commands to avoid alignment problems, as well as for some increasing of productivity.
    - *DAQ.Siemens*: The problem of the NoDave library for a global packing structures command calling "#pragma pack(1)" is fixed.
    - *UI.QTCfg*: The short time sleep after closing all windows in the switch off function is added. In order to provide the time for the object of all the windows to be finally destroyed and to prevent crashing.

2 Optimization, stability, sustainability and system performance increasing

During the work on this version, as well as its practical adaptation, it was discovered and corrected in a total of about 90 errors. In general, work has been done on cleaning code from the compiler's warnings with the "-Wall" flag:

- unused variables removing;
- adding the return in the all functions;
- initialization of variables with the possibility of their undefined using;
- exception of the incomplete handling of the enumeration types in the select operation;
- comparison between signed with unsigned integer.

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

- *System-wide:*
  - The command of the building system "$ make dist" is corrected and adapted to correct creation of the OpenSCADA distributions.
  - The TSYS::cron() function to call on the schedule is repeatedly corrected.
  - The variable type to store the result of the find() function, the string, is everywhere changed to the correct one "size_t".
  - *Function*: The possibility of a memory leak when attempting to set the object to a non-object function's parameter is excluded.
- *Data bases:*
  - *DB.PostgreSQL*: The handling of database address to determine the empty fields is fixed.
- *Transports and protocols:*
  - *Transport*: The loading of transports from the database is fixed, if there are records about the transport the module of which is missing.
  - *Transport.Serial*: O_NONBLOCK flag is set to open() function of a serial device to prevent the blocking of port's opening on some USB-> RS485 converters.
  - *Transport.Sockets, Transport.SSL*: The clearing of the input buffer is added to prevent re-use on high latency channels after timeout.
  - *Transport.Sockets*: The closing of the client connection by the time out of the lifetime for connections in the case of the data absence is fixed.
  - *Transport.Sockets*: Unix-socket is set to nonblocking mode by the O_NONBLOCK flag. The generating of the error messages for outgoing transports is fixed.
  - *Protocol.HTTP, UI.WebCfg, UI.WebCfgD, UI.WebVision, UI.WebUser*: The parsing of POST-request elements is fixed by the for parsing all the attributes of each element.
- *Data acquisition and archives:*
  - *Archive*: The using of the buffer with a soft grid with low-and high-resolution time in the archives of values is fixed.
  - *Archive.FSArch*: The creation of messages archive's files when you place a set of old messages is fixed. The parameter to prevent duplicates is added.
  - *Archive.FSArch*: The result of the fgetc() function is recorded in an integer variable instead of a character one that has corrected the discovery of "EOF".
  - *DAQ.System*: The crash when exiting to the "Hddtemp" source is fixed.
  - *DAQ.ModBus*: The setting of values of the parameter's attributes mode is changed to active and the error handling is improved.
  - *DAQ.JavaLikeCalc*: The check of the index when accessing the character of the string on the index is added, in order to prevent the out of range situation.
- *Visualization Interface:*
  - *UI.VCAEngine*: The blocking of the session for dynamically active projects with dynamic access to their own widget's attributes is fixed.
  - *UI.Vision*: The crash of the session for projects with an execution period of more than 500 ms is fixed.
  - *UI.Vision*: The fine-tuning of the coordinates of the "Elementary figure" primitive

shapes is implemented to the remainder of the widget's position.
  • *UI.Vision*: Scaling the text primitive "Text" is changed to change the font size instead of scaling the final image, which increased the overall quality of the display font.
  • *UI.Vision, UI.WebVision*: The resolution of the coordinates is increased of all the widgets to three decimal places, in order to ensure proper positioning on a large scales.
  • *UI.WebVision, UI.Vision*: The redraw of the primitive "Elementary figure" is fixed when the scale is changed.
  • *UI.WebVision*: The validation of figures' points for their getting to the display area of the primitive "Elementary figure" is removed. This function is shifted to the GD library.
  • *UI.WebVision*: The possibility of crashing for primitives "Diagram" or "ElFigure" is excluded in the case of transmission of incorrect values of attributes of the image "xSc" or "ySc" from a Web-browser.

Optimization and performance improvement:
  • *UI.WebCfgD*: The resizing of icons on the server side is added to reduce traffic.
  • *UI.QTCfg*: The group request to the control interface is used for performance improvement on slow and high latency connections.
  • *UI.QTCfg*: The multiple query of the elements of selectable table columns is excluded.

3 Improvement and stabilization of the graphics subsystem

The notable work was made within the graphics system, namely in the VCA engine modules UI.VCAEngine, visualizers UI.Vision and UI.WebVision, as well as in the configurators. The changes were provided for stabilization, the optimization of memory consumption and improvement of user's properties.

Improvements of the graphics subsystem:
  • *SYS, DAQ.JavaLikeCalc, UI.QTCfg, UI.VCAEngine, UI.Vision*: The API of syntax highlighting is added. On the basis of this API the syntax highlighting is added to:
    • *DAQ.BlockCalc*: syntax highlighting in description of the parameters' attributes.
    • *DAQ.JavaLikeCalc*: syntax highlighting for the compiler of the JavaLikeCalc.JavaScript language, as well as for all text fields using the language inside the module.
    • *DAQ.ModBus, DAQ.OPC_UA*: syntax highlighting of the attributes' list.
    • *Protocol.UserProtocol*: syntax highlighting to the text of the protocol's procedures.
    • *UI.QTCfg*: multilevel syntax highlighting displaying, transmitted from the data model in the form of the rules of regular expressions.
    • *UI.VCAEngine*: syntax highlighting for text attributes of the widget's primitives for which this is necessary.
    • *UI.Vision*: syntax highlighting for text attributes of the widget's primitives, as well as for the texts of the widgets' programs.
    • *UI.WebUser*: syntax highlighting in the procedures for constructing the Web-based interface.
  • *UI.QTCfg*:
    • The ability to manual change the height of the text and table fields. Change is made by grasping and dragging the bottom edge of the field.
    • Group request "CntrReqs" is used to query the page's context.
  • *UI.VCAEngine*: The specific properties of the attributes of primitives are added, such as help and syntax highlighting.
  • *UI.Vision*:
    • The display of help for the widget's attributes is added.
    • The display of message about loss of connection with the visualization server and its recovery is added for remote visualization.
    • For the "Elementary figure" primitive the dialog to set properties for the separately selected figures is added.
    • The separate full-text editor for text attributes is added.
  • *UI:WebVision*: The support for gif and jpg images in the elements of the "Elementary figure" primitive is added.

4 Formation and stabilization of the user programming API

It was continued the development of object API of the user programming, which provides the integration of user's functions in the object's tree of OpenSCADA. In addition, a number of changes was made to the existing library of the user's API functions.

In particular the following changes were made:
- *SYS*: User's API for the TArrayObj object is extended for the possibility of using object properties if the properties are not the numbers.
- *SYS*: In the object tree of OpenSCADA of the user's API the SYS.Security.access() function is added to control user access to OpenSCADA resources with the necessary privileges.
- *SYS, DAQ.JavaLikeCalc*: The implementetion of the support for regular expressions of user's API, based on PCRE, through the TRegExp object is added. The interface of functions RegExp.exec() and RegExp.test() is implemented in the standard for the JavaScript language. The "new RegExp()" initialization of the TRegExp object is implemented for the DAQ.JavaLikeCalc. The additional functions using RegExp: int search(); Array match(); Array split(RegExp pat, int limit); string replace(string substr, string str); string replace(RegExp pat, string str) are implemented for DAQ.JavaLikeCalc.
- *TFunction, DAQ.JavaLikeCalc*: The user's API object "TFuncArgObj" is added for function arguments get.
- *DB*: The following functions of the user's API are added: fieldStruct(), fieldSeek(), fieldGet(), fieldSet() and fieldDel() for thetable object to provide the user access to the database through the database interface OpenSCADA.
- *DAQ.JavaLikeCalc*: The built-in function typeof() to check the value's type is added.

5 System-wide expansions

To the system-wide API of the OpenSCADA system there have been made the significant changes and extensions with the purpose of the general stabilization and in view of the general measures for cleaning the source code and reorganizing the sources' tree:
- *SYS*:
  - The new structure of the sources tree is implemented.
  - All the resources of the project are moved to the separate archive.
  - To create sources' and resources' archives you can use the command "$ make dist-lzma" and "$ make dist-reslzma".
  - Documentation is made the static for modules and is divided by languages. For each documentation language it is formed the separate package. In the documentation of each language it is created an index file for the entire list of local and network documentation. The total list of local documentation is greatly expanded with new documents, as well as with the supported network documentation.
  - The packages of librarian databases are added: *-LibDB.Main and *-LibDB.VCA, they are placed to data/LibsDB.
  - Instead of the DemoDB package the packages of the "AGLKS" and "Boiler" models are added and placed to the data/ModelsDB.
  - Configuration files to build RPM and DEB packages have been updated.
  - Dependence on the library PCRECPP is globally replaced by PCRE.
  - The new function TSYS::time2str() is added to convert time intervals into a string like "1hour 23min 10sec". This function is used everywhere.
  - The implementation of own XML-parser is added, instead of Expat. The own XML-parser solves the problem of supporting different character sets other than UTF, as well as the problem of safe handling the situation of the characters presence that are not valid for the current character encoding.
- *SYS.TMess*: Common control interface messages are moved into one TMess::lab* function.
- *Transport, Transport.**: The function TTransportS::traf2str() is added to convert the value of the traffic's counter to the string like "12.5KiB".

Conclusion

The OpenSCADA system gets the push for the implementation of real hardware multiplatform, whereby the project's code has been largely cleaned up and made multiplatform. These developments will enable the project to continue moving toward the hardware multiplatform, as well as the implementation of the program multiplatform that is scheduled after the next LTS release 0.8.0.

Work also is continued on improving the overall functionality, simplicity and ease of using with the general improvement of the quality of service based on the OpenSCADA system.