Technical Description

Product name: Asix.Evo

Version: 8

1. Intended Use

Asix.Evo is a universal software package for the design and development of IT systems for enterprises, processes, industrial lines, machines and equipment, smart buildings and other installations. In terms of its fundamental functionality, Asix.Evo is a SCADA (*Supervisory Control And Data Acquisition*) system. It can also be used as an HMI (*Human-Machine Interface*) type system or in the MES (*Manufacturing Execution System*) type applications.

Asix.Evo provides tools for visualization of the state of the plant and equipment being controlled. It enables the user to collect and archive analogue and binary data, control the process and be alerted in case of emergency. It contains tools necessary to analyze the gathered data and generate production reports.

The range of applications spans from single workstations (also directly integrated with the machine) to complex multi-station networked configurations.

2. Features

2.1 State Visualization and Plant Operation Control

The object state visualization module allows creating any complex process diagrams for various applications. The basic feature of the module is that it maintains full scalability of the diagrams with the option to automatically adjust to the screen size, in the window (desktop) or browser (Internet) presentation. In both presentation modes, an identical application project is started - no conversion is required.

An application is designed with the use of fully interactive tools which are accessible in all Asix license types. The application can be modified on-line with a real-time view of values of all measurements.

The application module has the following characteristics:

- The application consists of a set of windows divided into sections. Each section contains a process diagram. The diagram in particular window section can be selected independently of other panels.
- A process diagram is a basic element of the application and is used to present process data and control the plant operation.
- The application consists of a set of visualization objects of various types.
  Depending on its purpose, each object has its own specific data visualization method and modes of interacting with the user. The range of available objects includes simple graphic objects (geometric shapes), specialized visual elements (e.g. gauge meters) and complicated objects dedicated to support information of a specific type (e.g. alarm support tables).
- Each of the visualization objects (even the most basic) may, through adequate parametrization, dynamically react to a change of the process state or be used to control it.
- The object state may depend on the current values of process variables, archived values and their aggregates, the status and history of alarm signals. In case of current data, it is possible to control the state of single bits or any bit groups.
- A current value, practically of each property of the object, can be calculated by means of universal arithmetic expressions which take into account values of many process variables and other application elements. There are more than 100 specialized functions of access to various application elements available.
- In the object property value calculation process, you can use conditional arithmetic expressions or the object multi-state abilities (defining alternative configurations of objects for various logic conditions).
- It is possible to create parametrized diagrams (panels) – this allows reducing the number of diagrams being created in case when similar application elements are visualized (e.g. drive panels, meter panels).
- The objects can be grouped into the so-called templates. A template consists of several pre-parametrized objects which are responsible for proper visualization of an application element. The template so prepared is repeatedly embedded in diagrams while the embedding parameters are provided. The later changes in the definition of a template are automatically transferred to all places where it is embedded.

- The process diagrams are fully scalable using the vector graphic. The diagram can be fit into the section in which it’s displayed. You can also use sections with scrollbars where only a part of a large diagram is displayed. There is a functionality available, designed to hide selected objects if their size becomes too small after rescaling of the diagram.
- Creating multi-screen applications is fully supported. Particular screens display separate windows (also these showing the same diagram). There are actions available which allow easy selection of the process diagrams on a proper screen.
- The operations to change the location and size of objects are available.
- Partial transparency effects can be used for the objects.
- The following types of graphic files are supported: BMP, GIF, Animated GIF, PNG, TIFF, SVG. Objects are scalable and can undergo many transformations: rotations, mirroring reflections, color modifications.
- You can use the *Browser* object on the diagram to display the content of any website and also documents, the types of which are supported by the browser.
- The designer can create one's own context menu and use it in selected parts of the application.
- The global properties feature allows you to centrally define the selected parameters (e.g. overflow display colors) and use them in visualization objects.
- The application is controlled by means of operator actions (context menus and keyboard short cuts attached to the objects). There are several dozens of operator actions accessible. The action content may be dynamically built depending on the status of the application/object.
- The program supports keyboard, all mouse buttons and touchscreens.
- Diagrams can be switched in the next/previous mode, based on the history of the operator's earlier actions.

2.1.1 Description of In-built Visualization Objects

Event log

The object enables displaying a table with latest system messages on the diagram.

Flap

The object is used to graphically present the slope of rotating elements such as dampers, blinds, screens etc. For each object you can define an element slope start and end angles, define the element type (single- or double-armed) and the shape of the ends.

Keyboard

The object enables displaying a keyboard on the diagram allowing entering text and numeric values using the mouse or touch screens. The Keyboard may operate in auto launch mode – appears on text entering attempt. It can also be positioned automatically.

Messages

The object displays text descriptions of the bit states of process variables. The definition of the states and the texts corresponding to them are stored in the message file.

Vector image controller

A non-visual object allows you to control the appearance of an image placed on the diagram. It can be an image defined as a diagram background or an image displayed by the Picture type object. In order to the object to be operational, the controlled image must be of the SVG vector type. You can control such attributes of selected elements of the image as: visibility, color, text (for text elements).

Chart controller

The object is used to display a toolbar which allows controlling the Graph objects. It is possible to simultaneously control the operation of a group of the Graph objects in a synchronized manner.

Shape

The object allows displaying four basic figures: rectangle, square, circle and ellipse with many graphic effects. Corners can be rounded and circle / ellipse sections can be displayed.

Line

The object allows creating single line segments or sequences of segments. It is possible to define the shape of the line ends and use dashed lines. It is also possible to smoothly curved segments in the mode of Bezier curves.

Tree List

The object allows creating hierarchic dynamic tree lists. Any operation action can be associated with each tree node. The List structure can be read out from the XML type file (also modified during application run).

Meter

The object is used to display numerical values of the measurements in the form of an analogue clock. The following properties may be defined for each gauge: value range, the number of major and minor divisions of the scale, start and end angles of the scale displayed. It is possible to display the limits (LL, L, H, HH) with colors individually set.

Image

The object is designed to display graphic images of the following type: BMP, JPG, GIF, PNG, TIFF, WMF, EMF, ICO, SVG. Also the animated GIF pictures are supported. The displayed picture may be proportionally or non-proportionally scaled. The picture may also be transformed in many ways: rotations, mirror, color, brightness and saturation change.

Authorization field

The object is used to perform operations of logging in, logging out and changing users' passwords. The object allows inserting user management functions directly on the diagram. An alternative solution is to use standard log-in windows accessible in the Control Panel.

Browser

The object is used to display the content of the selected web page or any document supported by the browser (such as PDF, certain Office documents, movies, etc.) on the diagram. One of the applications of this object may be displaying pictures from web or industrial cameras if the source of the picture is available as a service or HTML page. The object also allows displaying of the contents specified directly in the object definition.

Button

The object is designed to visualize the buttons used for digital control. Various types and styles of buttons are available - colors and visual effects may be freely configured. The button function may be indicated by a text or/and picture.
The button can be used to directly set up a variable value or carry any operator's action (e.g. switching over a diagram). The buttons may operate in the "repeat" mode, in which the control (an action) is performed cyclically when the button is kept held down.
It is possible to define a keyboard shortcut which may be used to activate the button function. In addition, the buttons can be grouped under the so-called control group, which enables synchronous control of individual bits of the same variable.

Operating Point

The object is designed to visualize the location of an operating point calculated as a function of two values responsible for the coordinates X and Y. The location of the operating point can be shown against the background of an image which is a part of the object or against the background of other objects. You can display the value scale within the object.
The location of the operating point can be displayed by means of one of the pre-defined cursors or use any image of your own choice. It is possible to draw the trace of the last changes of the operating point location.

Pipeline

The object is used to present the state of the pipelines. The shape of the object is defined by a coordinate series for each vertex of the pipeline. By an appropriate selection of the shape of the pipe ends it is possible to combine several objects into a single flow diagram. The object allows you to simulate the flow effect.

Bar

The object is used to both display the value measurements in the form of a scale bar and to control the value of the connected variable using a slider. The following properties may be defined for each bar: value range, the number of major and minor divisions of the scale, the location of scale display and the style of hatching the bar. It is possible to display the limits (LL, L, H, HH) with colors individually set.

Active alarm table

The object is the basic diagram used to display and support active alarms. It contains user interfaces which allow confirming alarms, excluding them or setting up the filtering condition.

Historical alarms table

The object enables you to display the registered historical alarms in a tabular form. It has actions of interactive selection of the set of the presented events.

AsBase table

The object displays the data from the AsBase program database in a tabular form. There are interactive operations available to select the presented data and to provide support for the data edition.

Database Table

The object displays in tabular form the data from any MS SQL server database. The object also enables you to edit the data.

Table of variables

The object allows displaying in a tabular form any of variable sets defined during editing or automatically loaded based on the selected values of variable attributes. Each table cell can contain any expression value created on the basis of the variable attributes and its value, fixed text, a hyperlink associated with an operator action, variable bit values or texts and descriptions of variable states.
In the Application Run Mode, the table allows the user to insert and delete rows corresponding to the selected variables, to replace the variables in the rows and to save the variable set modified that way.

Conveyor

The object is used to present the state of the conveyors. The shape of the object is defined by a coordinate series of individual belt conveyor joints. While defining the conveyor, its width, type (scraper or standard) and a style of the begin and end roller may be defined. It is also possible to simulate the belt motion in any direction and with any speed.

Text

The object is used to display information in a text form. The displayed text may be either static or may be linked to the process variables values. The object has built-in support for performing control operations. Two Input Modes for new values are available: direct text edition or selection from a set of predefined states.

Polygon

The object allows creating flat closed geometric figures of any shape. The shape of the object is defined by coordinate series for each vertex of the shape. The inside area color, thickness and color of the outline can be specified for each figure. It is also possible to create figures with smoothly curved edges in the mode of Bezier curves.

Chart

The object enables drawing variable values waveforms. In a typical case, the chart displays the values loaded from the archives of the historical process variable values (raw or aggregated). It is also possible to draw curves for values of current variables, reference curves, lines or bands of limits.

It is possible to draw multiple trends (series) on the same chart. It is possible to define separately the following for each series: variable, graph style (line or bar), color of correct values, color of uncertain values, line thickness, style of markers and whether it works in the so-called percentage scaling mode.

The object has its own actions for chart management, such as the scale modification, shifting along the time axis. A synchronous control of Chart object groups is also possible.

Power guard graph

The object is designed to display a graph for a selected media meter of the Power Guard module which allows you to monitor the intervals and the average value forecast in the current cycle. Furthermore, the graph displays the vertical line of the current time, horizontal lines of the limit and the horizontal line showing the value which is safe for the other intervals - so as to ensure that the total average for a cycle does not exceed the limit. The graph also enables you to review historical cycles.

Tank

The object allows displaying the shapes of tanks. The size and style of each tank top (right) and bottom (left) inlet/outlet may be defined.

2.2 Database of Definitions of Process Variables

The basic element of each Asix.Evo application is the database of definitions of process variables. It stores information about all variables of the application, both from the readings of physical devices and virtual variables (calculated in the application). The definitions database contains information about the way of reading and archiving variable values and write permissions. It also contains a set of attributes which describe the measurements, e.g. the unit of measurement and the measurement ranges. The set of attributes can be expanded by the application designer. The content of the variables definition database determines the way in which the application works. The variable attribute values can be displayed or included in the logic of animation of visualization objects.

The variable definitions database is provided with a multidimensional system of grouping the variables – it allows you to easily search for variables even in case of vary large databases.

The variable definitions database is saved in an MDB type database or, in case of large systems, in an SQL type central database. There are tools available to edit the variables definitions database.
In case of large databases, it is possible and recommended to generate the variable definitions database based on data sourced from the Excel program spreadsheets. In case of some

communication drivers (e.g. OPC DA, Network Variables of the WAGO controllers), the database of definitions of variables can be automatically generated based on the information read from the data source.

2.3 Process Data Communication

The basis for the operation of the Asix.Evo communication module is the database of definitions of process variables and the configuration of communication channels. Each channel is associated with a driver dedicated to the device from which the data is accessed. Asix offers a rich set of more than 100 drivers for a wide range of controllers, regulators and metering devices. Communication can be implemented simultaneously through a number of different physical channels – Ethernet, field bus, serial ports. There are drivers available which support industrial communication standards such as: Modbus RTU and TCP, OPC DA, OPC UA, Profibus, SNMP. Apart from universal drivers, the drivers dedicated to specific devices from many manufacturers are also available. There is also a group of special drivers which are not directly associated with the communication. They allow access to data from such sources as databases or text files and creating virtual variables (calculated within the application).

The list of communication drivers supplied with the Asix package is continuously expanded with new items. The Sales Information leaflet contains an up-to-date list. There is also a possibility of writing additional drivers upon an individual request.

The minimum period of reading the data from the devices is 1 second. The selected drivers can perform faster sampling at a millisecond rate.

The communication system supports all types of data from 8-bit to 64-bit ones. In case of binary data, it is not necessary to separate each signal for a separate variable. Binary signals can be received in groups in multi-bit type data – visualization objects are ready to work on the selected bits.

All data coming from communication channels can be initially recalculated – this enables conversion from measurement units (e.g. 0-2048) to physical units already at the data read stage. An identical reverse conversion is performed while transmitting controls / setpoints.

It is also possible to create virtual variables, the values of which are calculated based on values of many other variables from physical channels.

2.4 Data Archiving

The process variable values coming from physical channels and virtual channels can be archived by an internal Historian module named Aspad. Historian is an integral part of the Asix.Evo system, present in every package. Data is registered in the form of continuous waveforms with the minimum write period of 1 second.

The collected data is saved in the archived resources. It is up to the application designer to which resource the variable is assigned and what will be its archiving parameters.

There are two types of archived resources:

Standard Archives

The standard archives are stored in binary files. They are used to archive a great number of variables of a quuick sampling period. Variable values are compressed at very high ratio, which allows storing multiple years archives on the hard disk. The compression used takes the trend of the value changes into account. Particularly, fixed values are not registered. The designer can improve the compression rate, declaring registration accuracy for every variable. The Historian speed of data write and read is counted in hundreds of thousands of samples per second.

In multi-station configurations, standard archives have the following possibilities:

- In a system of redundant workstations, at the moment of start-up the historian updates the data corresponding to the shut-down period based on the data sourced from another workstation.
- It is possible to build archives by continuously replicating data from a master historian (server) . This ensures having identical data, regardless of whether the workstation operates continuously and whether the network communication works correctly all the time.

SQL Archives

The values of the SQL type archives are stored in the Microsoft SQL Server database. The data is not compressed and therefore the SQL archives are suitable only for storing variables with relatively slow sampling periods (e.g. minute-long ones). However, the SQL type archives have specific characteristics:

- Data synchronization in redundant systems – archives at many workstations are on-line synchronized : missing data are mutually refilled and measurements of incorrect state are replaced with the correct ones.
- The data can be updated backwards if the communication driver allows sending historical data.

The archived data is accessible in the same way regardless of the type of the resource in which they have been registered.

Other features of the archiving module are:

- Conditional archiving which allows you to limit the size of an archive – data is registered when a specific condition is fulfilled.
- Calculating and archiving a number of pre-defined aggregates which facilitate and significantly speed up advanced data analysis. By querying the aggregated data, you can significantly reduce the amount of data transmitted between the Historian and the application objects as well as the data analysis programs.. The aggregates are calculated for any chosen intervals of aggregation. Standard aggregates are: minimum values, maximum values, average, integral, increment.

2.5 Alarm system

The Asix.Evo alarm system has been designed to follow, to maximum extent, the guidelines of EEMUA (*The Engineering Equipment and Materials Users Association*), defined in the publication number 191.

The alarm system is divided into any number of the so-called alarm domains. A domain is practically an independent system of alarms. Standard, small applications use a single domain.
In larger ones, it may be favorable to divide alarms into many domains, e.g. for various parts of a plant or for various types of events being monitored.

The main components of the alarm system domain configuration is the alarm definition base and a set of alarm detection strategies. The following items are defined in the alarm definition base for each alarm:

- Identifier (ID)
- Priority
  Specifies the alarm importance. Message, Warning, Alarm, Urgent, Critical
- Category
  The available alarms include "ordinary" alarms for which initial and final events of the alarm duration are detected, and "warning" type alarms for which an occurrence of the alarm is recognized but there is no duration time.
- Detection strategy and parameters
  They determine the way in which the alarm status changes are detected
- Alarm text
  Separate texts can be defined for the beginning and for the end of the alarm. You can place additional data of the event in the alarm content.
- Event additional data support
  As an alarm is notified, the detection strategies can provide any set of additional data related to the alarm. Interpretation of these data is specific for the strategy employed. Furthermore, for every alarm the application designer can define any calculated expression, the value of which will be added to the event data at the moment when the alarm is prompted.
- Operator action
  You can define a suggested operator action for each alarm. The user will be able to perform this operation in the table of alarms.
- Exclusion signaling parameters
  The application operator can perform the so-called exclusion operation (blocking of events). This fact can be saved in the process variable in order to inform the PLC controller, that the alarm is excluded (e.g. to block sound signal emitted by the controller).
- Sound signaling
  An alarm occurrence can be signaled by sound. It is possible to read the alarm text or play the sound file. Each alarm can have a different tone. The tone can be repeated until it is faded out by the operator.
- Note entry flag
  The operator can enter a text note for every alarm confirmation.
- Mode of sending SMS/Email alerts
  The alarm occurrence note can be sent by SMS or electronic mail (by means of the optional AsAlert module). The addressee of an alert is individually declared for every alarm.

- Grouping attributes
  You can use a multidimensional grouping of alarms in the alarm definition base. This allows you to divide alarms in any way required. At the visualization stage, it is possible to present an aggregated status of any generated subgroup of alarms.

The so-called alarm strategies are responsible for detecting the alarm events. The basic types of the strategies are:

- bit strategy, detecting changes of process variable state bit values;
- condition-based strategy which allows monitoring the status of any logical conditions (e.g. controlling whether the variable values overrun alarm limits);
- OPC strategy - client receiving alarms from the OPC AE (*Alarm and Events*) server;
- buffer strategy which allows reporting a precise sequence of events
  with a millisecond resolution but requires special software on side of the PLC controller.

Apart from the above, there are also ancillary strategies used to receive events from internal Asix.Exo modules concerning e.g. errors of communication channels, overruns of the warning limits of the AsService counters, alarms prompted by application scripts.

An entire set of information describing a detected alarm event is registered for every such event. These are:

- alarm start and stop times (assigned by the detection strategy);
- alarm start and stop detection times (moments when the event is recognized);
- time, place and a user confirming the alarm;
- additional parameters of the occurrence and the alarm-related expression value;
- confirmation note.

The history of alarm events is stored in the local alarm base of each Asix.Evo workstation used to support alarms on an on-going basis. To support reporting in the *Reporting Services* or statistical analyses, it is possible to create an additional central archive in an SQL type base.

The emergence of a new alarm event may optionally be signalled using the pop-up information window. Also, the status of each individual alarm or any subgroup of alarms can be shown on a process diagram by means of standard visualization objects. Apart from that, you can use two special-purpose objects. The active *Alarms table* object is designed to show all currently active alarms in a tabular form. Depending on the configuration, it can also show events which have ended recently and/or have not been confirmed yet. The *Historical alarms table* object is designed to review the history of alarm events.

In the case of both objects the operator has at its disposal a number of tools allowing fast searching through the alarm list using various search parameters such as time, text, status, type, identifier. It is also possible to print alarms while maintaining the pre-set criteria. The active *Alarms table* is also an alarm support center. It provides interfaces for:

- confirming alarms and adding notes;
- excluding alarms - blocking, permanently or on a one-time basis, of alarms recognized as incorrectly detected (e.g. due to a failure of sensors or a partial shut-down of the system);

- filtering alarms - establishing time-based filters limiting the number of alarms (e.g. eliminating sequences of repeating events for the so-called bouncing contacts).

In multi-station configurations, alarms are supported in a fully synchronized way. Regardless of where the alarm was detected, it is propagated onto all workstations. All actions, such as alarm confirmation or exclusion, are immediately visible on the remaining stations. Similarly, when a workstation is restarted, the entire history of alarms is updated with the information about the current status of the exclusions and the filters.

2.6 Users' Permissions Control

The Asix.Evo authorization system is based on the user role system. Each Application user is assigned to one or several roles. The specific user privileges results from the fact of the assignment to a particular role.

There exists a set of in-built pre-defined authorities of predetermined meaning. The program automatically controls the assignment of these authorities and adequately blocks selected actions. An application designer can also add his/her own authorities of any meaning.
The process diagrams of an application offer functions which allow the designer to control whether a logged-in user has a proper role or authority (predefined or custom) and to block some of the operations.

Optionally, being assigned to the role may also determine the right to use particular files of an application, e.g. process diagrams or context menus.

As far as the process control is concerned, a three-level authority procedure is employed:

1. Checking whether a workstation from which an operation is initiated has a permission to perform control in the communication channel to which the controlled variable belongs.
2. Checking whether a logged-in user has a general authority to perform controls.
3. It is optionally verified if the logged-in user has a authority to perform control for a specific process variable.

In case of small single-workstation applications, the configuration of the access rights is saved in a protected local file. In larger multi-workstation applications it is recommended to use an SQL type central access rights base. Using a central base additionally expands the functionality of the system of authorities, particularly the operations related to the support of validated applications.

The authority system contains automatic user log-out feature. Logging out may be performed in various scenarios: after a certain period of time following the last logging in, at a specific time or after a specified period of the user being idle.

It is also possible to work in a one-time authorization mode. In case of particularly important operations, the user may be prompted for authorization while performing this operation (particularly, a user other the currently logged-in may be authorized, but it is a valid only during this operation).

The Asix.Evo application access system may be integrated with the *Active Directory* system services (the Windows domain). In this option, the correctness of the user log-in attempt is verified by a query to the domain (and not checking the password in the Asix.Evo configuration). You can also

associate the affiliation to the Windows groups with Asix.Evo application roles. In such option, the user administration may be entirely transferred to a domain.
Integration with a domain also imposes all domain-related restrictions with regard to the principles of defining and regularly changing passwords.

2.7 Support for Validated Systems

The Asix.Evo allows you to create applications which fulfil requirements established in validation procedures for systems designed to work in pharmacy, food and automotive industry, in accordance with the standards GAMP5 and FDA 21 CFR Part 11.

In this context, there are functionalities available which constitute a practical implementation of the requirements established for computer systems by the above mentioned regulations:

- The user authority control system with a central register of all log-in operations and other important events in the entire network of the application workstations. The integration with the *Active Directory*, services and the automatic log-out feature additionally increase the level of security. The access is controlled on the level of individual operations, files and process variables.
- Registration of control operations performed by the system operators, including the information on who, when and where performed a control operation, the old value and the new value of the changed parameter. The logical interpretation of the changes allows describing the performed control operation not only by means of numbers but also text descriptions (e.g. *On* , *Manual Mode*) .
- Application integrity control consisting of checking compliance of all application components, including database of process variable definitions and process diagram definition files with reference state allowed for use on the basis of validation procedures and on that basis preventing attempts to put the unauthorized application components into operation;
- Registration of the operator's actions (at selected workstations) – a list of opened diagrams and trends.
- A system of the user's notes, making it possible to comment on control operations, confirm alarms and any other events.

In order to use control registration, integrity control and operator activity registration functions, you need to have an additional AsAudit license.

2.8 Process Data Trends

The integral part of the Asix system is a program named AsTrend which allows an advanced analysis of process data in the form of graphs. The AsTrend program can be used as a part of the Asix.Evo visualization system or a standalone program (also to analyze data not coming from the Asix.Evo application). AsTrend is available as a standard window-based program or it can be run in a web browser.

AsTrend allows you to analyze data coming from many sources. These are:

- Trends of analogue data coming from process data archives, also for aggregated values.
- Trends of binary values sourced from process data archives. It is possible to plot the state of any bit.
- The value waveforms which are calculated based on other measurements from archives.
- Information on the timing of alarm status changes.
- Data sourced from the AsBase event archiving module database. The data can be displayed in a uniform way (without taking into account the data write time if only the values of consecutive measurements are of importance and not their time sequence).
- Fast signal waveforms registered by the AsLogger module.
- Data read from the CSV and XLS type files.
- Data coming from the SQL databases accessible through the OLE DB interface.
- Data sourced from the OPC HDA standard historical data servers.

The AsTrend program offers the following functional possibilities:

- It allows creating any custom data sets and displaying their waveforms in a common window.
- Many graph panels
  You can open up to four independent chart panels in the program. Each panel can display a chart of any selected waveforms from the list of all waveforms present in the trend. Each panel can contain a different axis type or even a different graph type.
- Individual scaling of graphs
  The multiple OY axis mode makes it easier to analyze and compare curves representing different parameters on the same chart. The curves can be scaled independently allowing them to be moved away from each other which facilitates assessment of their interdependence.
- Arbitrary arrangement of graphs from different periods
  The function allows comparing the variable value waveforms in different time periods. It is possible to achieve this effect by the use of an additional time axis added for one of the variables whose time curve needs to be compared with that of another variable (it is also possible to compare time based curve of the same variable in two different time periods).
- Two-dimensional X-Y graphs
  AsTrend also offers a possibility of presenting and analyzing the relationship between variables using two-dimensional X-Y graphs, which are essentially graphs of many variables (so called dependent variables) as a function of a single independent variable. Furthermore, by adding a minimum and a maximum, the range of allowable variable changes can be declared (also as a function of independent variable), which allows placing the graph and analysis within the correct operating range of the controlled system.
- Creator of trends for easy creation of charts
- Value reading lines
- Legend, the layout of which is freely configurable by the user
- Operations of smoothly zooming in / zooming out the selected part of the waveform
- Displaying value waveforms in the form of a table
- Exporting value waveforms to PDF, BMP and TXT files or an MS Excel spreadsheet
- Displaying descriptions of values of variables as text labels of the axis of values

- Generating ad-hoc simple tabular reports for various periods of time and selected aggregates of variables

2.9 Event Archiving and Recipe Formulation Module

The event archiving and formulation module named AsBase complements the possibilities offered by the Asix.Evo application with an automatic, event-controlled archiving of selected structural data and a system designed to manage sets of setpoints (recipes).

Both functionalities are based on the *Microsoft SQL Server* database.
In multi-station configurations, one central database is used as a standard. AsBase offers mechanisms which enable momentary operation without connection to the database server. If necessary, the data is locally buffered.

Event Archiving

The AsBase module archiving allows registering freely definable data records which contain selected parameters of the process. When combined with a unique identifier, these data can be associated with a specific product or its batch. This allows production flow tracking and generating production documentation for a single product or product batch. Contrary to the archiving of process variables of the Aspad historian, which is oriented towards saving continuous measurement series of variables, the AsBase registers data in the form of records while the recording time is only an ancillary value.

The data registration moment is determined by the fulfilment of a logical condition – the possible conditions may be based on the monitoring of values of process variables and time conditions. The manual registration mode is also available.

The registered data come from sets of process variables. It is possible to register in one table data coming from several sets – it allows registering production from several lines in one location. Some data can be complemented manually.

The AsBase module also allows registering the production data in many interrelated tables.

In multi-station configurations, there is available a mechanism of automatic negotiation of the workstation to record the data.

Recipe Formulation

As far as the recipe formulation is concerned, the AsBase provides tools for editing, storing and applying sets of values recognized as a recipe or a set of setpoints. Each recipe is identified by its name. The recipe structure is freely definable.

The recipe selected by the user is sent to the set of process variables controlling the production. There can be many sets of variables – this allows controlling the operation of many similar devices from one set of recipes. It is also possible to use the automatic recipe loading mode in response to the fulfilment of a specific logical condition.

There are also available so-called percentage recipes. In this mode, some ingredients are given as a percentage value – the batch amount shall be specified at the moment of loading and all percentage fields will be appropriately recalculated.

All recipe loading operations are separately registered. The full content of the procedure is recorded, including the recalculated values of the percentage fields. It is possible to establish relations with the event archiving tables, which makes it possible to combine the production data with the recipes which were used to develop the product.

The AsBase data can be reviewed and edited in the AsBase program window or on standard visualization diagrams of the Asix.Evo application. By using the AsTrend application, it is also possible to display data in the form of charts from event logging system tables. The diagrams can take the time scale into account or distribute points in a uniform manner (if the time of events is not important and only the consecutive registered measurements are valid). The registered data can also be used to search for archived data. The search engine in the AsTrend module enables you to select a required record, find its associated record in the Aspad archive and draw the curves on the screen.

2.10 Reporting

The Asix.Evo system offers several universal methods for creating reports. They differ from one another in terms of the extent of knowledge necessary to create a report and the type of data which can be used in a report. Some of the system modules have also their own specialized subsystems for generating documents, for example: printing the AsBase module tables, the AsTrend program charts or making screenshots in the visualization module.

Features of the universal report generation modules are described below.

AsRaport (*Reporting Services*)

The report creation method known as AsRaport is a tool for designing and generating reports based on the *Microsoft SQL Server Reporting Services*. The reports are generated by means of free tools provided by Microsoft, e.g. *Report Builder*. They can be generated by every user with proper skills. The reports can be generated based on the Aspad historian archived data (raw and aggregated) and all SQL databases of the Asix.Evo application (the databases of AsBase, AsService, historical alarms). The AsRaport facilitates creating SQL queries which are the basis for generating reports. The Asix.Evo also provides methods to start calculation and viewing the reports.

Apart from the services provided by the Asix.Evo system, you can use all possibilities offered by the *Reporting Services*, especially the possibility to publish the reports on browser-based websites and save ready reports in PDF, DOC, XLS and HTML files.

Excel program spreadsheets

The services designed to provide access to the archived and current data through the Automation interface allow each user of the Excel program to easily import selected data directly into the spreadsheet. This process is additionally facilitated by the set of prepared VBA macros and templates.

AsTrend reports

Apart from its basic graph display functionality, the AsTrend program has been equipped with the function of creating ad-hoc tabular reports. The report editing process is fully interactive and does not require the user to have any special skills.
In the standard use, it is enough to select the process variables and aggregates and to provide the period of time and the aggregation range. It is also possible to generate a report for variables calculated based on the archived waveforms. For each report, you can freely define the report header and the summary (max. five lines long) and the ready reports can be printed and exported to HTML and XLS files.

Script reports

The Asix.Evo application script module contains specialized objects which make it easier to create reports. Practically, these reports can use data from any sources. The data can come from elements of the application which are not a part of the Asix.Evo system. Making reports requires basic programming skills and therefore they are to be handled by the application designers. Script reports are generated in the form of PDF format files. The script report module is complemented by the report manager window which allows the user to calculate and review reports. The reports can be automatically calculated, too.

2.11 Schedules and Timetable

Asix.Evo uses two mechanisms designed to automatically perform actions which determine the state of the application and the controlled object.

The schedules module is intended for the application operators. It has an interface which allows you to plan actions in the application execution mode. It offers rich possibilities to precisely plan the moments when selected operations are to be performed. An action can be performed at a specific time or cyclically. While defining cycles, you can take the calendar (working days, holidays etc.) into account.

The timetable module is entirely configured by the system designer. It is used to define the permanent behaviors of the application. It cannot be changed by the operator while the application is being run. The timetable cycles are fixed. The timetable enables to synchronize action execution with the occurrence of the conditions other than time based (such as a predefined value of a process variable, operator's exchange).

2.12 Media Monitoring

Asix.Evo provides a special-purpose module named Power Guard designed to control the consumption of any utilities (e.g. electric energy, gas) on an on-going basis. The Guard allows you to monitor the on-going consumption and use a forecast to generate an advance alarm about the risk of exceeding the acceptable limit (e.g. contracted power limit, which when exceeded results in severe extra fees). Monitoring is performed in freely declared (e.g. 15-minute-long) settlement periods. The actual consumption level and its forecast are visualized on a specialized graph or by means of standard objects. The limit overrun risk may be signalled by a standard alarm.

2.13 Reference Curves

The mechanism of reference curves allows creating ideal characteristics of changes of the variable values that are used to control the actual process. The reference curves can be applied on the current waveform in the *Graph* object. This allows you to visually control the state of the process. The reference curve can also be transferred to the controller in order to precisely carry out the controlled value change pre-set program.

There is an interactive editor of the reference curves available for the system operators to edit and manage the reference curves.

2.14 Multi-Workstation Configurations and Redundancy

Asix.Evo allows creating applications consisting of a single workstation as well as complex systems which include many workstations with various functions. In such systems you can distinguish the servers which have direct access to the PLC controllers and perform data archiving. The separate group are the operator's terminals (window-based and browser-based) which source all data by means of networked interfaces from the servers . Any mixed configurations are also possible – the servers can source some data through network interfaces, too.

Asix.Evo has been designed specifically for process control systems. This is why a great emphasis has been put on system stability, internal control of correct operation of individual modules and ensuring redundancy to help minimize the effects of hardware failure. In particular, the Asix is prepared for creating a redundant structure called "hot spare", when two or more server workstations are connected to the site through separate communication channels conducting independent acquisition and archiving of process variables and have their own copy of event and alarms log which is updated in real-time.

Such a system of redundant Asix workstations when connected through Ethernet becomes resistant to failures including:

- Damage of the physical channel of communication with a plant
  The workstation experiencing such a problem automatically switches over to source data from a redundant workstation through a network connection while maintaining its full functionality. After the connection with the controller is repaired, the physical connection is restored.
- Server failure or planned shut-down
  The terminal workstations automatically switch over to the other server. After a successful start-up of the server, all archives are updated from the redundant one.

High reliability operation mode is supplied by standard with every operator server license.

2.15 Asix Mobile Module

Asix Mobile module is an extension of a set of tools and programs available in the Asix system with dedicated functionality for all kinds of mobile devices - a smartphone or tablet type.

Asix Mobile enables creating a fully functional interface for visualization of inspected facility state, as well as control of its operation. Design and use of the graphical interface has been fully adapted for devices with small screens.

Asix Mobile enables you to view currend and historical values of measurements, control the facility work, check alarm status including acknowledgment of the alarm reading, make notes transferred to fixed stations.

2.16 AsAlarm Module

Optional AsAlarm module is an application providing tools for in-depth analysis of alarms generated by the monitored site and of other data relating to alarm system operation. The application meets the EEMUA (*The Engineering Equipment and Materials Users Association*) guidelines No 191. The module allows two-level alarm analysis:

- static assessment of the design of alarm system structure for a specific application;
- dynamic analysis of alarms registered on site;

The dynamic analysis of the history of events can be performed on any data subset. The analyses results can be viewed by means of:

- historical event table,
- waveform charts of selected alarm events,
- analytical section for calculating various statistics.

The AsAlarm program can calculate and visualize the following statistical data:

- event occurrence distribution (number, percentage),
- event duration,
- average time to acknowledge,
- number of terminated alarms,
- number of acknowledged alarms,
- identification of most frequently occurring events,
- identification of the longest events.

The AsAlarm module is available as a standard window-based program or a browser-based application.

2.17 AsAlert Module

The optional AsAlert module provides services which allow sending notifications about the occurrence of important events in a plant controlled by the Asix.Evo application.

Messages can reach recipients in a variety of ways:

- standard electronic mail over the Internet using SMTP protocol;
- text messages over GSM network;
- messages addressed to terminal devices of the Ascom industrial call and telephone system.

In the standard manner of use, the user declares in the application alarm system configuration who should be informed about the alarm events and which events are they. It is also possible to send the alerts from the user scripts. In this case, the condition of sending and the message content may be of any kind.

Each alert can be send to many addressees at the same time. Sending through the schedules is also possible – the alert will go to various addressees, depending on the time of sending it.

2.18 AsLogger Module

The optional AsLogger module is a Fast Measurement Series Recorder application for recording and analyzing measurement series where measurement samples are time-stamped with accuracy of up to 1µs. Contrary to the standard archiving of process data which is done in a continuous manner with the maximum resolution of 1 second, the AsLogger archive is oriented towards short-lasting measurement series registered at high frequency. Each measurement series is individually registered and identified.

The measurement-series data is taken from a recording device, pre-processed and saved in an SQL database. The AsLogger handles the retrieval of data from an OPC DA server, thus supporting any device which meets the requirements of data exchange open standards. It also supports control and measurement cards and USB measurement equipment by Advantech, where measurement data may be read in the *'Fast AI Transferring'* mode (*Interrupt Transferring or DMA Transferring*), using the ADSAPI library by Advantech. The AsLogger also supports switching board protection controllers MUPASZ 2001G, MUPASZ 07 and MUPASZ710 produced by Instytut Tele- i Radiotechniczny and MultiMuz produced by JM-Tronik. A measurement series can also be written by means of the user scripts – this allows registering series coming from any data sources.

The registered measurement series of the AsLogger module can be analyzed by means of graphs and tables of the AsTrend program.

2.19 AsService Module

The optional AsService module is a program dedicated to the maintenance management and control of production assets. It allows registering the time of operation and the number of starts of the devices by means of counters. The counter values are calculated based on the data sourced from the Asix application. This makes it possible to alert the user to carry out the necessary maintenance and repair operations, report any missed maintenance and log the related technical data for each monitored device.

For each device, it is possible to define a set of electronic documents related to its operations, including the maintenance manuals, operation manuals, test reports, process diagrams and others.

The counter values and their configuration data can be shown the Asix.Evo application process diagrams. Warning threshold overruns of operating time counters and the number of starts can be signaled to the operators by alarm events.

2.20 Openness of the System

If necessary, the functionality of the Asix.Evo application can be expanded by means of the designer's additional custom-made program modules. The following possibilities are available:

Application scripts

The Asix.Evo applications can be written in the C# and VB.Net languages. To serve the scripts, there is a program interface provided which, among other functions, gives access to process variables (current and archived), control the state of application alarms and control the work of the application. Apart from this interface, the scripts can use all possibilities offered by the .NET framework. It is possible to access the system files, support any databases and create interactive graphic interfaces using the model of the WinForms objects.

External visualization objects

The designer can create one's own visualization objects extending the set of built-in objects. The external objects are created in the .NET framework programming languages. The external objects may be graphic-oriented (their form is directly displayed on the screen) or object-oriented (they used standard WinForms controls to create interacting elements). The external objects' code has full access to the process variables of the application.

2.21 Providing Access to Process Data and Alarms History

The data possessed by the Asix.Evo application can be made accessible for the third companies' software. The type of access depends on the kind of data. The following methods are available:

Current process data

- Automation

It is designed for use in programs and scripts written in languages which support the Automation mechanism, particularly in the Excel program macros.

- OPC DA 2.0

Implementation of the standard industrial OPC DA protocol for transmitting current data.

- OPC UA

Implementation of industry standard protocol OPC UA for current data sending.

- .NET server

It is designed for use in programs written in the .NET framework code.

- Web Service

Providing access to the data in the Internet network according to the Web Service standard.

- DDE

Making data accessible by means of the DDE mechanism from Microsoft.

Historical process data

- Automation

It is designed for use in programs and scripts written in languages which support the Automation mechanism, particularly in the Excel program macros.

- OLE DB

It provides access to the data to clients who can communicate directly with the server, using the OLE DB protocol, or indirectly through the ADO interface.

- SQL

Mechanism designed to fulfil the requirements of the *Reporting Services* service. It allows data querying through SQL queries. It can return values of many variables in a tabular form.

- .NET server

It is designed for use in programs written in the .NET framework code.

- Web Service

Providing access to the data in the Internet network according to the Web Service standard.

Each of the archived data read method is available regardless of the type of the archive where the data is stored (binary, SQL). It is possible to read raw and aggregated data.

Historic alarms

The archived alarm events are read directly from the SQL type alarm archive database.

Making the data accessible within the same computer does not require any additional licenses. Making them accessible through another computer requires the *AsixConnect* license to be installed on that computer or, if the Web Service is used, the *@asix4internet* license to be installed on the computer which provides the access to the data.

2.22 Asix Excel Add-In

Asix Excel add-in alows a typical Excel user to use current and historical data of the Asix.Evo application. Add-in is installed automatically during the installation of Asix package. MS Excel interface is then extended with tools supporting the use of Asix system data: collection of specialized functions and Asix Table object. Add-in also makes access Asix data easy, using scripts based on a specially designed object of Asix.Evo class.

2.23 Operation Mode

The Asix.Evo applications can be run in the following modes:

Standard window application

The basic operation mode as an independent window-based application of the MS Windows system. The software must be installed on every workstation where the application is to be started. The application works as an operator's workstation. It can also perform server roles – provides data to other workstations.

Operation in the Windows system service mode

The application runs in the background - it does not require a Windows user to be logged in. It only works as a server providing data - it does not have its own graphic interface. The software must be installed on every computer.

Browser-based application

The application is launched through a web browser. The application works only as an operator's workstation. Its functionality is the same as the window-based version. The application uses the same configuration data (process diagrams) as the window-based version. No conversion of the application is required. Running the application does not require any software to be installed on the client's computer – all necessary program modules and the application definition are downloaded during connection established with the Web server.
As the Web server, only the *Internet Information Services* server, being a part of the MS Windows system, can be used. The browser-based Asix.Evo applications use the XBAP and ActiveX type application technology.

Operation through the RDS remote service desktop.

The application is run as a remote session on the Windows server with RDS services installed. The application is accessed by means of any remote desktop software compatible with the RDP protocol, including one working under an operating system which is different than Windows. In the RDS mode, the application works only as an operator's workstation. The Asix software is installed only on the server's computer.

3. Licensing

The basic product licensing is based on the number of the process variables used and the role played by the PC station. For the licensing system important are the variables accessed by communication drivers and the virtual variables (calculated in the application) which are archived.
Licenses for the following types of workstations are available:

- Engineering workstation
  Workstation dedicated to the development of applications. It allows access to the data from communication channels and local archiving of the data. The operation time in the application running mode is limited to two hours.
- Operator's workstation
  Workstation dedicated for autonomous operation. It allows visualization and control of the process, access to data from communication channels and archiving data locally. It does not provide access to its data to any other workstations.
- Operator's panel
  Workstation with the functionality compatible with the operator's workstation (with a limit of the archived data period) and designed to be used on the HMI panel type devices with the Embedded class Windows system.
- Operator's server
  PC station designed to work in networked multi-workstation configurations. It allows visualization and control of the process, access to data from communication channels and archiving data locally. Current data, archived data and alarm signals can be transmitted to other computer workstations on the network. It can work in redundant configurations.
- Operator's terminal
  This stations allows you to visualize and control the process. All current and archived data is sourced from the operator server type stations.

All the above workstation types have an built-in tools necessary to design applications.

Apart from the basic licenses, there are also additional licensees available which expand the capabilities of the basic licenses. Some of these licenses may also be supplied as standalone licenses.

- @asix4internet licenses
  The licenses allowing you to run the application in the browser-based mode. The licenses are sold in the CAL model - a license for a definite number of simultaneous users. There are full type CAL licenses and the Lite type CAL licenses. The Lite licenses are only intended for viewing the state of the process and come without any process control functionality.
- RDS type terminal licenses
  The licences with the functionality of the operator terminals, but intended only for work in the RDS remote access environment. The licenses are sold in the CAL model - a license for a definite number of simultaneous users.
- OPC/DDE/OLE/.NET server (AsixConnect)
  The licence designed for remote access through third companies' software to the data made accessible by the Asix server stations.
- AsAudit
  Extension of the basic licences by special functions intended for running validated applications.

- AsAlarm
  The licences for the program of statistics analysis of the alarm signals archive.
- AsAlert
  The licences for the server used to send alarm alerts by means of SMS messages and electronic mail.
- AsService
  The licences for the program registering the time of operation and the number of switching-on of the devices.
- AsLogger
  The licences for the module registering fast measurement series.

The type of the license is saved in the provided USB hardware dongle .

The Sales Information document contains a detailed description of all license types .

4. Language Versions

There are two language versions of the Asix.Evo program available: Polish and English.

The user applications can be developed in any language supported by the Unicode encoding system. It is also possible to develop the so-called multilingual applications, including the function which allows changing the language on-line during run of the application.

5. System Requirements

To operate correctly, the Asix.Evo requires one of the following operating systems: Microsoft Windows 7 SP1, Microsoft Windows 8/8.1, Microsoft Windows Server 2008R2 SP1, Microsoft Windows Server 2012. Using any of the earlier versions of the Windows system does not guarantee achieving the full product functionality.

The operation performed with the use of the RDS terminals requires using the server versions of the system: Microsoft Windows Server 2008R2 SP1 or Microsoft Windows Server 2012. You need to purchase an adequate number of the RDS Microsoft access licenses.

To work in the operator's panel mode, you need an Embedded system: Windows Embedded Standard 2009, Windows Embedded Standard 7, Windows Embedded 8.

Running the application in the browser mode requires the Internet Explorer browser, version 8 or newer, or the browsers: Firefox, Opera, Chrome with IE Tab.

In case of using the ASIX package functions which require using the SQL type data base (the SQL type archiving of data, an SQL type database of variables, authority system database, AsBase and AsService modules databases etc.), you need to install the following server: Microsoft SQL Server 2008 R2, Microsoft SQL Server 2012 or Microsoft SQL Server 2014.
In case of all applications, except for the AsService module databases, it is sufficient to use the Express version server. The AsService module requires at least the Standard version server.

6. Hardware Requirements

To ensure correct performance of the Asix.Evo program, it is sufficient if the hardware requirements recommended for the actually used operating system are fulfilled. In case of large applications (supporting several dozens of thousands of process variables) it is recommended to use 4GB of the RAM memory to be used by the Asix.Evo program process.

The optimal capacity of the disk space depends on the amount of the archived data, the selected archiving methods and the expected data storage duration.

One USB connection must be available for connecting the hardware license dongle.

7. Contact

ASKOM Sp. z o.o.
ul. Józefa Sowińskiego 13
44-100 Gliwice, POLAND
tel. +48 32 30 18 100
fax. +48 32 30 18 101
biuro@askom.com.pl
office@askom.com.pl
www.asix.com.pl

SALES INQUIRIES		
Krystyna Boroń	Krystyna.Boron@askom.com.pl	tel. +48 32 30 18 126
Renata Michałek	Renata.Michalek@askom.com.pl	tel. +48 32 30 18 128
TECHNICAL INQUIRIES		
Wacław Bylina	support@askom.com.pl	tel. +48 32 30 18 141
Marian Strzałkowski	asix@askom.com.pl	tel. +48 32 30 18 152

You are welcome to test the Asix SCADA system by installing its free 90-day trial version:

http://www.asix.com.pl/en/about_asix/trial_package/

For Sales Information, please visit:

http://www.asix.com.pl/en/commercial_inf/