SEP

Smart Embedded Platform

SEP is the world-first next-generation-oriented smart control platform – Realizing seamless M2M connection with anyone anywhere around the world with no awareness of differences in hardware, OS and communication.

INTERNATIONAL LABORATORY CORPORATION

SEP

Smart Embedded Platform

"New driver around the globe"

SEP will constitute future connection of electric-powered Machine to Machine all over the world.

Logo represents the image of cord-to-cord connection with the motif of "m" in the Machine.

The blue color symbolizes "Sky" and "Ocean" connecting the world, fusing with SEP in the future.

Design images the SEP-integrated world, and limitlessly-expanding-code represents its infinite possibilities

＜What is SEP（Smart Embedded Platform）?＞

In a distributed embedded system constituted by a number of devices and computers, SEP provides a mechanism that enables SW exploiter to conduct development in the same way as that on a single device or computer without awareness of the communication process between devices. The SEP-provided **smart embedded platform** connects seamlessly without concerning the differences in individual hardware and OS, and the communications between devices so that the development for upgrading embedded systems to m2m/IoT/IoE can be strongly supported.

（m2m：machine to machine　IoT：Internet of Things　IoE：Internet of Everything）

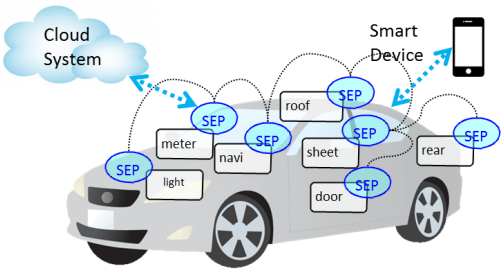
POINT 1　Speedy & Easy control （All-in-One functionality required for control system development ）

The plenty of SEP classes (software component collection) makes it easy to develop the control systems.

＜Control Class example＞				
	SepDiag	Diagnostic function class	SepController	Access I/F with Controller class
	SepVio	I/O access processing class	SepDataCollection	Data log processing class
	SepTimer	Timer processing class	SepDataDelivery	Recipe switching class
	SepVsensor	Sensor function class	SepAlarm	Alarm monitoring class

POINT 2　Next generation integrated control system （Machine to Machine, System to System Integration）

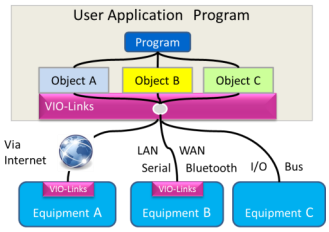
VIO-Links (Virtual Object Linking System), the communication virtualization system of SEP, realizes a seamless world including inter-microcomputer communication in the control system, inter-CPU core communication, as well as integration with smart devices such as smartphones / tablets, and up to widen approach like data sharing with cloud system on the Internet. VIO-Links absorb the difference in communication mode so that multiple systems and devices including legacy equipment can be easily connected only with the name specified. Communication program development among systems and devices is not required.



<SEP provision case to automobiles>

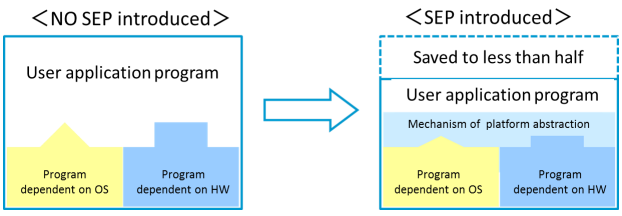
POINT 3　Microcomputer to Microcomputer （Seamless integration with distributed embedded systems ）

When connecting between devices with VIO-Links, the class method on the destination device can be called as if it were on your device without concerns of communication mode and device configuration. Using this mechanism (so-called Remote Agent Call), it enables to encapsulate communication mode and device configuration so that you can simply build applications for distributed embedded systems. Furthermore, you can instantaneously cope with modification in system device configuration and communication mode.



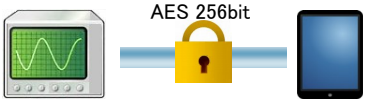
POINT 4　Compatible and Scalable (Platform-independent compatibility and portability)

SEP provides the mechanism which separates the platform-dependent parts such as hardware and OS, and minimizes the impact to the application layer at the time of platform change. It results in significant cost savings in the event of a system replacement.



POINT 5　Safety feature and Secure communication (Mechanism of strengthening safety and secure communication function)

SEP provides the safety enhancing mechanism including interlock mechanism or multiplexed processing upon error, as well as secure communication function which strengthens security by dynamic conversion of encryption key and encryption scheme in the inter-microcomputer / inter-device communication.



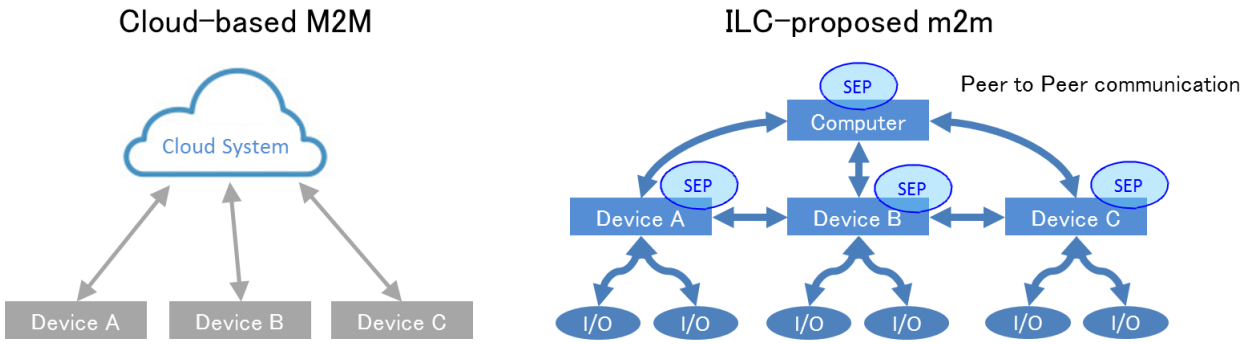
POINT 6　Low cost and High quality (Development cost reduction and high-quality implementation by the effects of significantly saved man-hour cost)

With use of ILC-guaranteed SEP software component collection (SEP Basic Class Library) and development tool (SEP Platform Builder), you can significantly reduce man-hours in the control system development, test and maintenance (reduced to one-fifth cost of pre- introduction according to in-house comparison).

What is SEP?

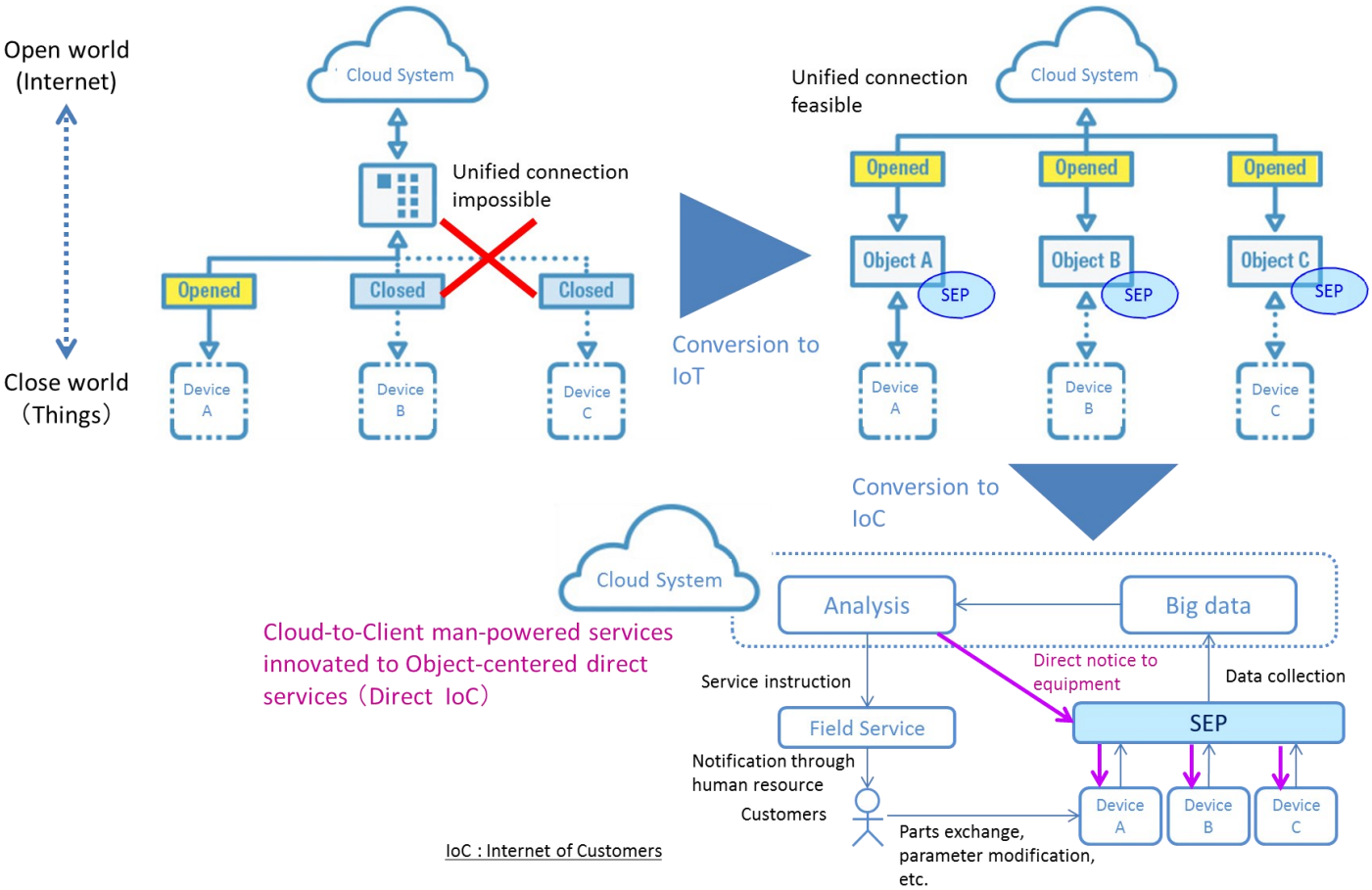
1: SEP realizes real m2m direct communication.

In recent years, Cloud type M2M which communicates via cloud between each machine and IoT/IoE solution in which each machine information is collected to cloud system and analysis of the aggregated data (Big Data) is conducted for business efficiency are getting popular, whilst ILC has returned back to the origin of M2M, and intends to create new additional value of M2M by connecting machine and machine directly, not via cloud.
ILC-proposed M2M is a new concept of Machine to Machine direct communication. The purpose of M2M, IoT (Internet of Things) and IoE (Internet of Everything) is not connection to the Internet or Cloud system, but creating new system (so-called smart social system) with use of machine-to-machine, machine-to-user connection, which has never realized by single machine or individual person. SEP provides a transparent mechanism which allows inter-machine control regardless of gaps in connection method, control target, execution platform etc. ILC proposes m2m which description means its difference from cloud-based M2M.



2: New IoT connection in Close-Open worlds feasible by SEP

IoT requires the connection between various kinds of objects (machine, equipment, system, etc.) and information system (cloud), while the reality resides in the Close world of objects with unique access method or specs depending on manufacturer, model, location on NW, etc.
IoT of the Close world needs the mechanism which connects uniformly with information system, what is called the Open world regardless of considering the gap from different constraints among objects.
Introducing the SEP-provided object oriented technology (method for generalization of object), accessing to SEP object equals to connecting with the object. Indeed, using SEP object allows us to realize the Close-Open world connection (IoT). In addition, SEP is expected to be useful in IoC, Internet, business connection with Customer being further than IoT .



What is SEP?

3: SEP integrates devices with secure cryptographic communication.

In the embedded system, inter-machine secure communication function is a major issue. From a powerless platform with limited CPU power, up to a high-end platform with powerful CPU, SEP supports encryption method from simple Blowfish to AES (256bit) as standard to secure data transmitted on SEP communication protocol for connecting SEP embedded machines. (User specific encryption algorithm for embedding can be also provided.) In addition, secure communication functions including dynamic key exchange by challenge response, escalation function of encryption strength, security loss detection and preventive function are also provided.

4: SEP seeks memory saving and compactification.

SEP embedded platform works in a minimum structure (VIO-Links function) with 30KBytes of ROM and 8KBytes of work RAM memory. It can also work on a compact MCU such as 16-bit microcomputer and µITRON etc. or RTOS (Real Time OS). It supports Bluetooth, SIO, CAN, NFC in addition to TCP/IP and UDP in the transport layer of the communication.

<SEP providing robust support for m2m/IoT introduction in embedded systems >

Since embedded system development generally needs to cope with various constraints (small memory size, power-saving CPU, strict real-time, 24 hour- continuous operation etc.), standardization of the platform (such as H/W or OS), communication (protocol, route, H/W etc.) and I/F with device (access method and others) are troublesome. Compared to the information system development with various standardization preceded, the issue of poor development efficiency has become more serious. As a solution, SEP provides a built-in platform and development support tool that have the following advantageous functions and structures.

* Classes (software component) with mechanism for abstracting the individual difference, and is not affected by replacement of platform, communication and devices.
* Classes with aggregated know-how of control system development (alarm monitoring, data log, event log, change recipe, system diagnostics etc.)
* Mechanism without needs of protocol design and communication program development (Remote Agent Call function)
* Mechanism to absorb the difference in platform such as hardware and OS for simplifying the system replacement (multi-platform function)
* Mechanism and secure communication function to enhance the functional safety

	Case	Conventional method (SEP unintroduced)	SEP introduced
1	Control system development with embedded device	General-purpose software widely used in personal computers are not applicable, and it must have all of the functions developed from basic functions to application-dependent part.	Using SEP-provided plentiful classes needn't basic function development. Development is required for application-dependent part only.
2	Development of system with connection of inter-microcomputer, inter-device, inter-system for functional integration	Protocol design, communication program development and secure communication function development are required.	No need of protocol design, communication program development and secure communication function development. (SEP provides inter-device communication function)
3	When add or change the communication specification such as path, communication method etc.	Application software modification development and re-testing are required.	Only the development for change in SEP communication-dependent layer and fixing communication settings are required. No need of any change in the application software.
4	When replacing the computer platform (H/W, OS etc.)	Modification development of application software and re-testing are required. (Entire modification needed in most cases)	Modification of platform-dependent layer in SEP only. No need of application software change.
5	When changing the devices in the system (manufacturers and series etc.)	Expanding impact range due to change of application software. Re-testing range also tends to be widened.	Replacement of inheritance-origin class that is target of specification change, and the difference development of modified part are required. Impact range due to the change can be minimized.
6	System building with use of general-purpose terminal (Windows / Android / iOS)	Communication program development between embedded devices and general-purpose terminal is required. Application development for each OS is also required.	Development for communication program between embedded devices and general-purpose terminal is unnecessary. Using separately sold INTAWORLD (see P.18) allows the development of application which can operate on each terminal transparently with different OS.
7	Development of multiple inter-computer integration function (Construction of the distributed control system)	Inter-computer communication processing program and integration function are required.	Software on several computers can be developed as a singular transparent system.
8	Integration with existing system	Communication / integration processing between existing system and newly built system are required.	With SEP Add-On to existing system, no need of development for communication / integration processing with new system.
9	Ensuring the functional safety	Additional development for ensuring functional safety is required.	SEP-provided functional safety function is available. No need of development.
10	Development assisting tool	User needs tool development or needs to purchase tools .	SEP-provided development assisting tool is available.

Table of SEP Basic Class (Part-1)

Note: A part of Class will be supported at SEP Ver.1.4 or later.

Legend : Foundation class | Basic class | Control class | Media class | FA class | Administration shell class | Cloud class

NO	Class Name	Derivation-origin Class	Description of Function
1	SepObject	Foundation class	Class for originating all the SEP classes. Each class inherits SepObject.
2	SepEventSource	SepObject	Class for notifying events. Basically, events are informed with Callback. The notified side needn't inherit SepEvent. Event is to receive the notification responding to method execution results or trigger-condition such as status change based on the specific method registered in advance. It is also utilized for receiving Public event broadcasted, in that case, the side receiving Public event is also required to inherit SepEventSource class.
3	SepApplication	SepObject	Class required to be inherited in the case of dynamic disclosure of method as an application. Other disclosed method can be used without this class inherited. When opening a method, this method is required to be inherited or executed.
4	SepSystem	SepObject	Class for obtaining SEP system information and changing system configuration. It specifies Key for obtaining / setting an individual information. It is also possible to obtain / set all required information at once.
5	SepLog	SepObject	Class providing user / log function for debug. Log is output by the method "write" included in this Class. It also allows the scope of output with specified levele. Log output for entire Object (application) is controllable.
6	SepFileUtil	SepEventSource	Class for file accessing. For file accessing, the language/OS-provided function can be used. Using this class can enables environment dependency dismissal and remote file accessing.
7	SepFileUtilAsync	SepFileUtil	Class desynchronizing each method of SepFileUtil class
8	SepStreamer	SepEventSource	Class for stream data handling. It processes unfixed-length stream data which cannot be handled by file processing. Transferring unfixed-length data between objects is also possible.
9	SepDatabase	SepEventSource	Class for database handling. Database itself should be on server connected to the network. This class realizes the function using Thin Client, which means no database function in the class itself.
10	SepSecurity	SepEventSource	Class for handling security information such as encryption. It copes with block encryption and consecutive stream encryption. A specific encryption method can be also implemented with CPL（Cal Porting Layer）
11	SepNetwork	SepEventSource	Class for providing physical/logical-media-independent communication. Abstraction not only for IP communication, but also for the communication using Bluetooth, RS232C, USB and shared memory provides bidirectional communication function. Even if the target's IP is unknown, you can directly communicate with specifying the Name.
12	SepDataSaver	SepEventSource	Class for data sharing (simple database) processing. Since this data is basically located on a shared memory, it may be disappeared when power discontinuity if no data is retrieved in application side.
13	SepNameResolver	SepObject	Class for Name solution processing. It converts Name, which is abstract concept, into entity. Searching is possible with setting the scope of search destination.
14	SepUtil	SepEventSource	Class for providing utility function. Not only obtaining version, it also provides language/OS/environment-dependent-processing as generalized (wrapped).
15	SepExec	SepEventSource	Class for process execution. It executes other application starting etc.
16	SepScaleTransform	SepObject	Class for scale converting process. With use of conversion equation, it can convert into arbitrary variables / units.
17	SepStatistics	SepObject	Class for statistical processing
18	SepEventLog	SepObject	Class for event log handling
19	SepCalendar	SepEventSource	Class for providing calendar function
20	SepClock	SepEventSource	Class for providing clock function
21	SepMail	SepEventSource	Class for e-mail send / receive function
22	SepPrinter	SepVio	Class for printing
23	SepImageProcessing	SepEventSource	Class for image processing. Preprocessing for specific image data (banalization, filtering etc.), measurement (edge detection, pattern matching etc.), determination processing are available.
24	SepPowerMng	SepEventSource	Class for powering
25	SepConfig	SepEventSource	Class for retrieval / reconstruction of each Object settings. It retrieves / reconstructs the setting, but the processing with use of the setting is implemented to Object.
26	SepDiag	SepEventSource	Class for handling diagnostic function. Each diagnostic function should be implemented in the Porting Layer when SEPcore porting by user.
27	SepDiagControl	SepDiag	Class of diagnostic processing for control devices. Diagnostic processing for a specific object's On / Off times, cumulative time, tolerance value excessive time.
28	SepSimpleVio	SepEventSource	Class for I/O access processing. It provides the transparent mechanism for I/O accessing regardless of the difference in device access method. For various I/O board or other devices, it defines the specific device class which inherits this class. Port setting / read and write etc. is possible for one specific port or for multiple ports at one time.

Table of SEP Basic Class (Part-2)

NO	Class Name	Derivation-origin Class	Description of Function
29	SepVio	SepSimpleVio	Class of security control function such as exclusive lock and interlock based on SepSimpleVio class.
30	SepTimer	SepEventSource	Class of timer processing which creates event at designated time. It allows one-time call or periodic call after an appointed time. Timer can be set in the order of milliseconds, while timer precision in the case of loaded onto non-RTOS such as Windows and Android depends on a platform or application.
31	SepVsensor	SepVio	Class of abstracted sensor function
32	SepBarcode	SepVsensor	Class of abstracted QR code reader
33	SepNfc	SepVsensor	Class of abstracted NFC reader / writer function
34	SepDataCollection	SepVio	Class providing the function which obtains a current variable of a specific registered Name List data with a designated condition (periodical, at the time of event). It is used for data log processing.
35	SepDataDelivery	SepVio	Class providing the function which modifies (rewrites) a specific registered Name List data into a specific variable with a designated condition (periodical, at the time of event). It is used for system
36	SepInterval	SepVio	Class for ON/OFF switching a specific Name data with a designated time interval.
37	SepAlarm	SepVio	Class for alarming
38	SepAlarmHistory	SepAlarm	Class for history management of alarm object
39	SepAlarmList	SepAlarm	Class of list management for alarm history
40	SepMedia	SepVio	Foundation class of multimedia function including movie, picture, voice etc. This class is never for single use.
41	SepCamera	SepMedia	Class of abstracted camera function. This class has the function for camera control, but no Encode/Decode function for picture.
42	SepVideo	SepMedia	Class of abstracted Video stream. This class has the function for Video stream control, but no movie playing function.
43	SepController	SepVio	Foundation class for FA area controller including PLC/Motion Controller/CNC. This class is never for single use. Class for each manufacturer's controller which inherits this class is necessary for applica-
44	SepAdministration	SepApplication	Class for each object management. Management shell for Industry4.0 can be created with derivation of this class.
45	SepAdministrationList	SepAdministration	Class for SepAdministration Objects management. This class manages Objects with hierarchy structure. Management shell hierarchy structure for Industry4.0 can be created with derivation of this class.
46	SepCloudSender	SepEventSource	Class with abstracted function to receive order or information sent from Cloud
47	SepCloudReceiver	SepEventSource	Class with abstracted function to send sensor information, etc. to Cloud
48	SepLanguageClassifier	SepEventSource	Class with abstracted function to extract the meaning of text data which describes natural language
49	SepTextToSpeech	SepEventSource	Class with abstracted function to generate voice from text data
50	SepSpeechToText	SepMedia	Class with abstracted function to analyze voice data and convert it to text data
51	SepImageRecogniton	SepMedia	Class with abstracted function to analyze image data and detect object

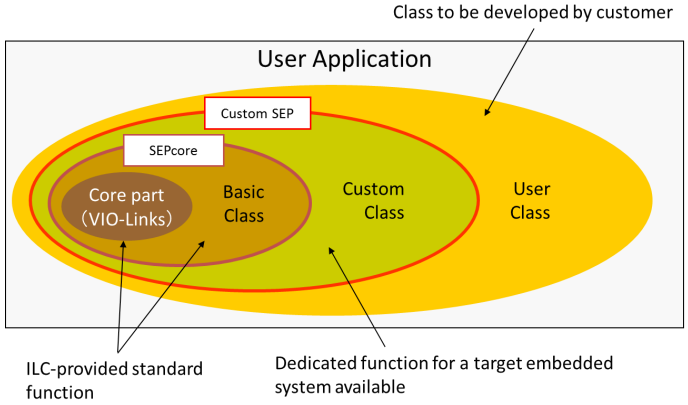
SEP manufacturer customization function

For development of SEP embedded application software, user have to port SEP to the target device (Please refer to p.14 for details).

This Manufacturer customization function eliminates the necessity of porting so that user can start development of SEP application immediately after purchasing the target device.

With the control target I/O, Driver for accessing various device and Class Library (so-called SEP Custom Class Library) provided by the embedded device manufacturer, user can develop SEP application without porting.

SEP Custom Class Library is developed with use of SEP development tool, SEP Platform Builder（Class Builder） and SEP Basic Class Library. (For every control target, define the Custom Class inheriting SEP Basic Class, and then implement the processing program of each Method in Custom Class.)



SEP application development flow

■ SEP application development flow

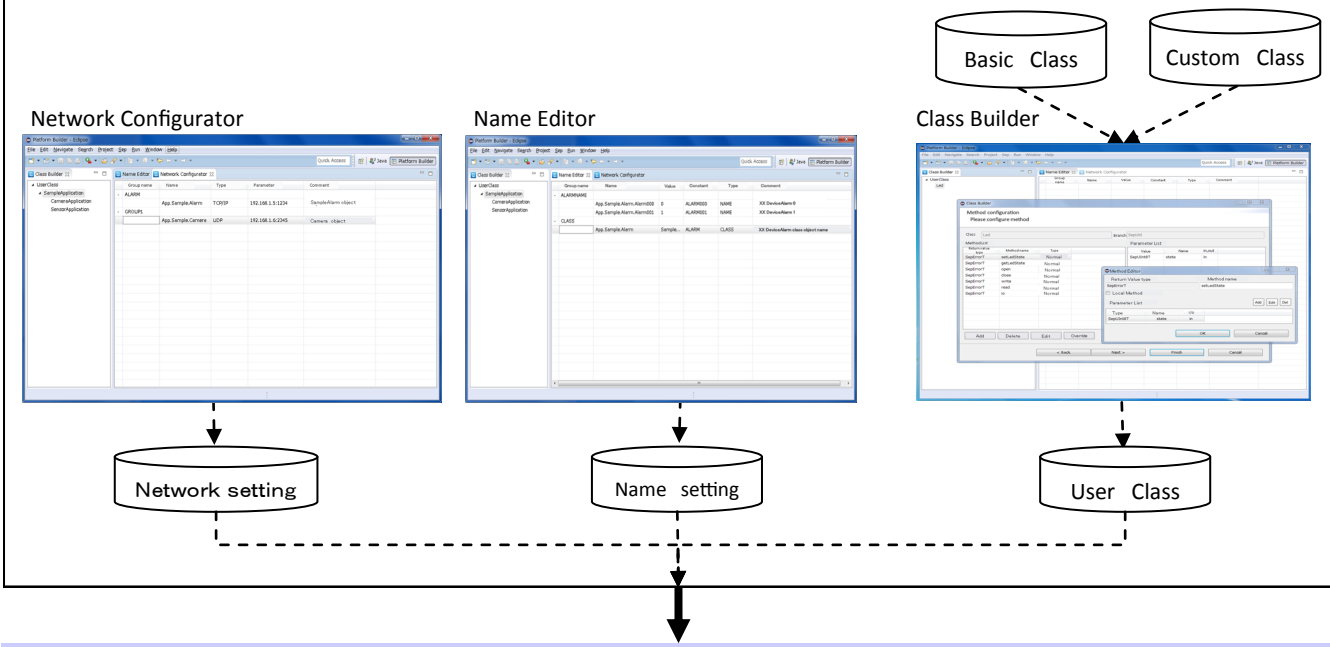
Followings are the work flow in the development of SEP embedded application.

(1) Porting an embedded platform (SEPcore) to the target device for embedding

<Required development tool>
SEP porting support package, C/C++ IDE for the target device
<Work items>
* Porting a platform dependent part (PAL)
Implementing the OS multiple tasking function (timer, semaphore, task relative function etc.) process corresponding to SEP I/F. And, implementing the process for accessing with H/W conforming with SEP I/F.
* Porting a communication dependent part (CAL)
Implementing the process for communicating with devices to be connected in accordance with SEP I/F.

(2) Work for preparation of SEP application software development

<Required development tool>
SEP Platform Builder
<Work items>
* Network setting (Network Configurator)
Setting the parameters (IP address, communication speed etc.) for communication with devices to be connected.
* Name setting (Name Editor)
Setting the Name such as device name and data name in device used in the system
* Class design (Class Builder)
Defining the Class of device and system with use of class library (SEP Basic Class Library) providing system basic function. C program code is automatically generated from data of each tool setting. This code is embedded for SEP application development.



(3) SEP application software development

<Required development tool>
SEP Platform Builder, C/C++ IDE for target devices
<Work items>
* SEP application software development
With use of network setting, Name and Class prepared in (2) , conduct the SEP application development.
* SEP application debugging
Debugging SEP application using LogViewer and Object Caller.

Notes: Separately sold “INTAWORLD” allows us to develop SEP application using visual design method with no necessity of C programming.

Example of SEP application field

■ Example of SEP application field

SEP can be utilized in a wide range of embedded control system development. Followings are some examples of SEP application field.

(1) Smart vehicle (Intelligent car)

Automotive	Automotive control system In-vehicle HMI system construction machinery etc.

(2) Smart home

Home Automation	Intercom system HEMS system Home security system etc.
Energy conservation	Air conditioning system Solar power generation system etc.

(3) Smart industry

Logistics	Automated warehouse system Logistics sorting system etc.
Monitoring	Power monitoring system Security surveillance system etc.
Building management	Building management system In-building security system In-building disaster preventing system etc.
Device manufacturer	Semiconductor production machine Liquid crystal production machine Construction machine, Processing equipment etc.
Control	Various controller (PLC, motion, inverter, CNC, robot etc.)
Traffic related	Road surveillance system etc.
Inspection device	Various inspection device Various measuring equipment etc.
Terminal for business	Electric automobile charging terminal Accounting machine at parking lot Various vending machine etc.

(4) Smart health care

Health care related device	Sphygmomanometer Scales Pedometer etc.
Medical	Control system in operating room In-hospital system etc.

(5) Utilization of smart device

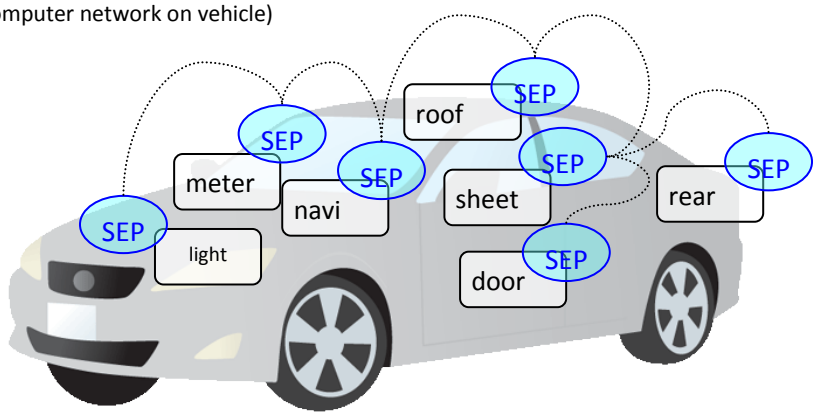
With SEP installed in smart devices like smart phone and tablet, the smart device can be utilized as remote terminal for a control system.

* As remote operation terminal
* As a monitor terminal displaying device operation status graphically
* As a security terminal for device and system (working with manufacturer-provided e-manual and user-created e-data)

Internet function between SEP-equipped devices enables remote operation.

Application example 1 Application to smart vehicle (Microcomputer systems in vehicle)

Linkage between microcomputers used in vehicle (Development of transparent smart vehicle control system with no concerns of inter-microcomputer network on vehicle)



(1) Connection between Head Unit and Rear Unit

Rear Unit in the back seat of the vehicle can utilize music/ video playback, navigation, phone function etc. embedded in the Head Unit (Automotive navigation system) in the driver's seat.

Rear Unit software needs to use Head Unit embedded various functions like music reproduction via inter-Unit communication. Once SEP introduced, without design of inter-Unit connection method and communication protocol as well as communication processing program development, in other words, regardless of inter-Unit communication, Head Unit's each function can be called.

(2) Connection with smartphone/tablet

In recent automobiles, remote operation of music/movie reproduction function in Head Unit is enabled by the connection between vehicle system and smartphone/tablet via WiFi and Bluetooth.

In the development of such a system, with use of SEP installation, inter-device function integration software for smartphone/tablet and Head Unit can be developed with no need of communication protocol design and communication processing program development.

No need of communication program development means that for example, in the development of software on smartphone, wireless-connected music replay function in Head Unit can be executed without communication program developed.

(3) Connection of switch/sensor

Various switches and sensors embedded in a vehicle are connected with the dedicated network like LIN and CAN up to the present. However, for switches and sensors of little relevance to the safety, no-wiring Bluetooth and wireless connection like WiFi are being considered.

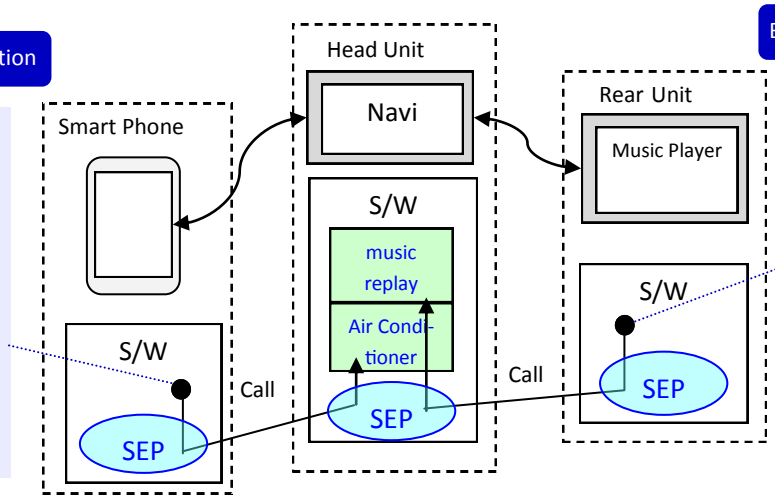
In the case of changing the communication connection method from CAN to WiFi, introducing SEP can minimize the revision of application software program.

Current issues	Benefits of SEP introduction
Development cost increased due to duplicative development of similar function (e.g. Music replay function development of RearUnit)	Due to easily reusing the function already equipped on other Unit, duplicative development of similar function is unnecessary.
Additional development is required for utilizing smart devices	Due to equipped function of each Unit available from smart device, development is simple.
Development is required each time of changing communication method (e.g. change from CAN to Wi-Fi)	Even if communication method changed, no application change is necessary.

Benefits of SEP introduction

* Air conditioning function is not equipped on a smartphone, but the same function on Head Unit is easy to use.

* In vehicles with no Rear Unit equipped, smartphone can provide the same function.



Benefits of SEP introduction

* Air conditioning function is not equipped on a smartphone, but the same function on Head Unit is easy to use.

* In vehicles with no Rear Unit equipped, smartphone can provide the same function.

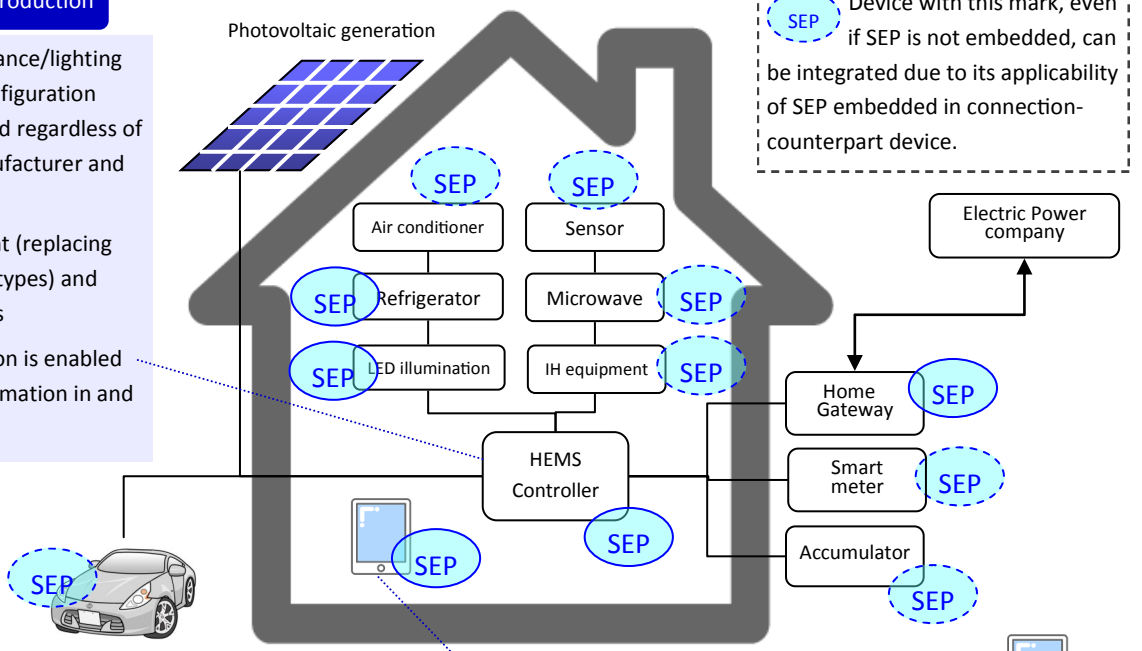
Application Example

Application example 2　Application to the Smart home (Home Automation system)

Benefits of SEP introduction

* ON/OFF of appliance/lighting apparatus and configuration change are enabled regardless of difference in manufacturer and type.
* Easy replacement (replacing manufacturers or types) and addition of devices
* Home Automation is enabled based on the information in and out of the house.

SEP Device with this mark, even if SEP is not embedded, can be integrated due to its applicability of SEP embedded in connection-counterpart device.

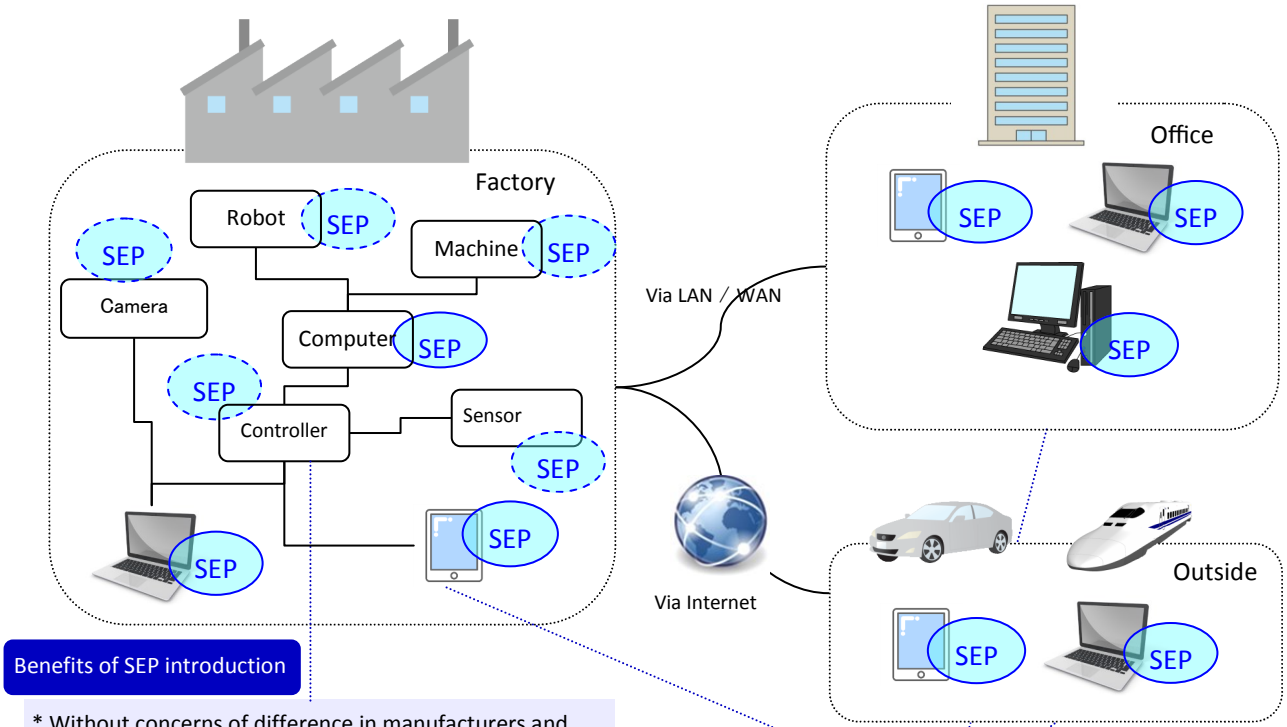


Benefits of SEP introduction

* Smartphone and tablet can substitute home appliance standard terminal (remote operation, error diagnosis, online manual etc.)
* Remote control and monitoring from the outside.

Remote access from outside

Application example 3　Application to smart industry (Factory Automation system)



Benefits of SEP introduction

* Without concerns of difference in manufacturers and types, centralized information collecting, error monitoring and control of device/equipment in a factory are allowed.
* Easy replacement (replacing manufacturers or types) and addition of devices.

Benefits of SEP introduction

* Smartphone and tablet can substitute standard terminal for all devices in a factory (remote operation, error diagnosis, online manual etc.)
* Remote monitoring and information collecting allowed from an office and the outside (worldwide).

SEP Merchandise line

Merchandise line

- 1: SEPcore (SEP embedded platform)
  - 1-1 : SEP Basic Class Library — Class library (Software component group)
  - 1-2 : VIO-Links — Communication virtualization mechanism
  - 1-3 : SEPPorting layer interface — Porting layer interface
- 2: SEP Platform Builder (SEP embedded platform building tool)
  - 2-1 : Network Configurator — Network setting function
  - 2-2 : Name Editor — Name editing function
  - 2-3 : Class Builder — User Class editing function
  - 2-4 : LogViewer — Log data display function
  - 2-5 : Object Caller — Debug function
  - 2-6 : Key Generator — Key code generation function
- 3: SEP Porting support package (SEP porting-dedicated development kit)

SEPcore

1: SEPcore Overview

SEPcore is the machine-embedded platform workable in the system (application) which uses SEP-provided function.

SEPcore consists of the following three function layers. (See FIG. 1-1-1)

1) SEP Basic Class Library

2) VIO-Links

3) Porting layer interface

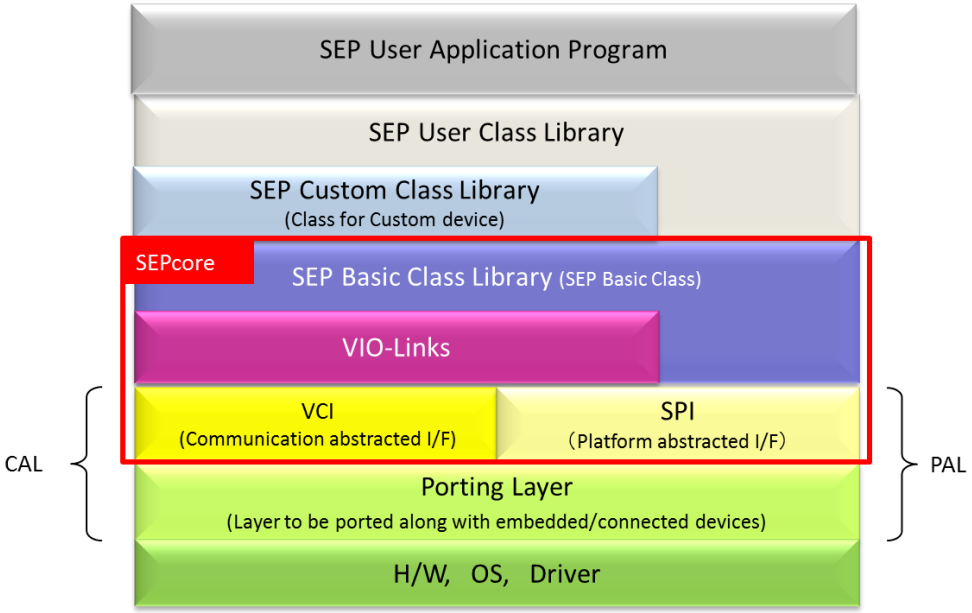


Figure. 1-1-1 Software structure of SEP application

・CAL（Communication Abstraction Layer)
Combination of VCI and the process (function) for communication processing with target device based on VCI is called CAL. SEP provides the process (function) for TCP/IP, UDP, Bluetooth and virtual serial communication as CAL standard function.

・PAL（Platform Abstraction Layer)
Combination of SPI and the process (function) dependent on the platform such as H/W, OS etc. of the target device based on SPI is called PAL. SEP provides the process (function) for Windows, Linux, Android as PAL standard function.

1-1: SEP Basic Class Library

It provides the basic function required to develop a system (application software) that utilizes SEP function.

In order to develop a system using the manufacturer's specific function of equipment and devices, inheritance of the Class Library is required, and creating SEP Custom Class Library that implements the interface processing between each equipment and device is also needed.

For example, in SepCamera providing camera function, it provides the necessary minimum basic camera function such as "take photo" and "look at photo". In order to respond to manufacturer specific function such as Smile shutter, it inherits this Class Library, and create SEP Custom Class with Smile Shutter function.

In SEP application development, with the Class Builder included in SEP development tool, so-called SEP Platform Builder, the SEP Basic Class and the user-defined class (SEP User Class) derived from SEP Custom Class are used. (See FIG. 1-1-2)

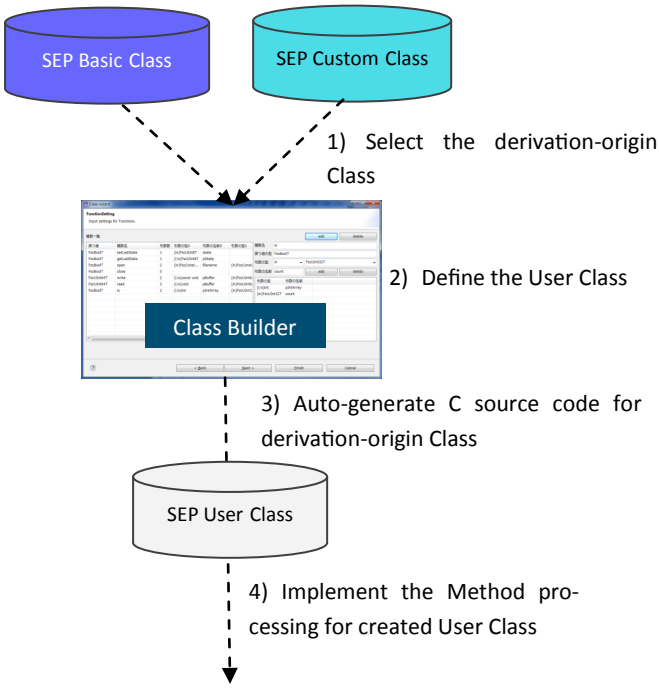


FIG.1-1-2 Flow for using SEP Basic Class Library

1-2: VIO-Links® (VIrtual Object LINKing System)

VIO-Links is separated into two sides, one is "Want to use function" side, the other is "Provide function" side. Even though both sides exist far from each other, the VIO-Links mechanism virtualizes the communications between them as if they use a function in the same device. VIO-Links itself in SEPcore is deployed between 1) SEP Basic Class Library, SEP User Class / Custom Class and 3) Porting layer interface (See FIG.1-1-1). Regardless of whether an object (device or system) to be accessed from an application software exists in Local (access from the same computer as application software) or Remote (access from other computer connected via network), and also without concerns of gap in communication method with the object or difference in platform (H/W, OS etc.), it provides transparency that enable us seamless access. (See FIG.1-1-3) This mechanism allows the program development in the User Application Program without being aware of the detail method for accessing the entities (Device A/B/C in the case of FIG. 1-1-3). Furthermore, in the case of changing the Object (switching the device A/B/C in FIG. 1-1-3), without modifying the User Application Program, what you have to do is just to change the derivation-origin Class of Network Configuration (communication setting), Name (definition of access destination) and User Class (Object definition).

1-3: SEP Porting Layer Interface

An interface layer for porting along with the platform (H/W OS, etc.) for SEPcore loaded device consists of the following two interfaces.

* VCI (VIO-links Communication abstraction layer Interface) for implementation of communication dependent function

* SPI (SEPcore Platform abstraction layer Interface) for implementation of platform (H/W, OS) dependent function

Porting development is required in accordance with this interface specification before application software development.

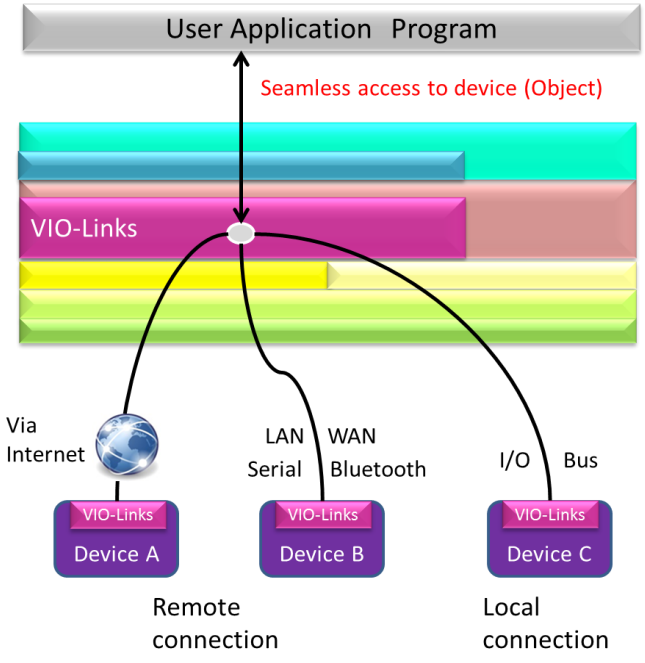


FIG.1-1-3 VIO-Links function diagram

2: Overview of SEP Platform Builder

SEP Platform Builder is the development environment to support development of SEP embedded application software. It provides the following functions along with application software development process using SEP.

(1) Development process for embedding SEP into a customer's target devices

With use of SEP Porting support package (see 3), port development proceeds along with target device platform (H/W, OS, communication target device etc.). Note that in the case of using a target device with SEP Custom Class Library, SEP is already embedded, that means no need of port development by the customer.

(2) Preparation process for application development using SEP

In the preparation process before application development, with use of network configuration function (Network Configurator), Name editing function (Name Editor) and User Class editing function (Class Builder), it prepares (automatically create data from each function) configuration file required for system construction, C source files etc.

(3) Process of application development using SEP, debug and maintenance

After application development based on data created in the process (2) , in the debug process of target devices with created execution module embedded and maintenance process after system launched, the function (LogViewer) for displaying log data generated by application software and SEP system is used.

2-1: Network Configurator (Network configuration function)

Before application development using SEP, it needs to configure network parameters used in the system with this function. This configuration data is created as network configuration file, and embedded into target devices with application execution module. This configuration needs to be set for each target device and then embedded.

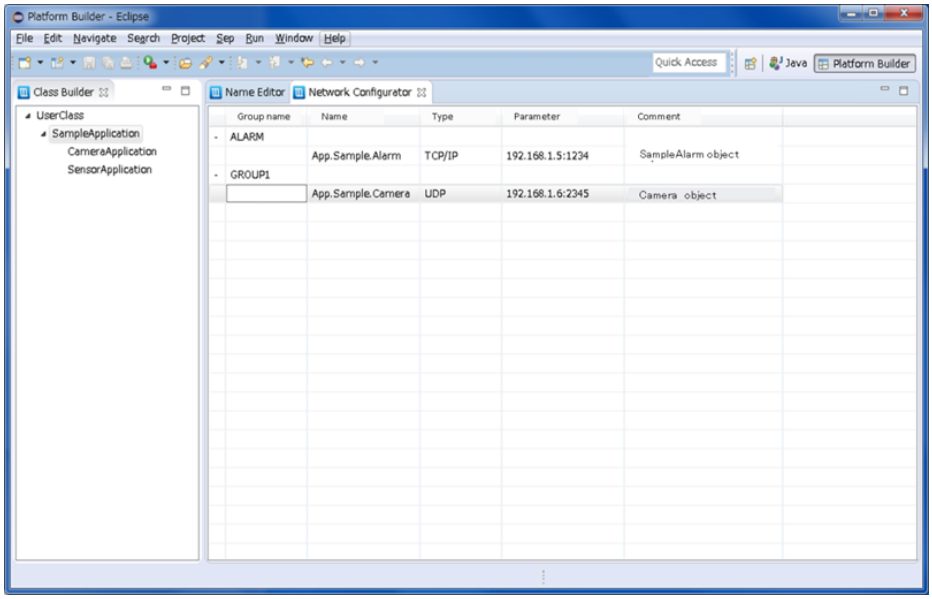


FIG. 2-1-1 Example of Network Configurator screen

2-2 :Name Editor (Name editing function)

In advance of application development using SEP, with use of this function, general name (Name) and entity (constant variable, fixed name etc.) are allocated correspondent to various kinds of constant variables used in application program and fixed name (e.g.: address name in PLC etc.) for specifying data in access target devices. This configuration data is automatically created as include-file of C program. It's included to application program for use.

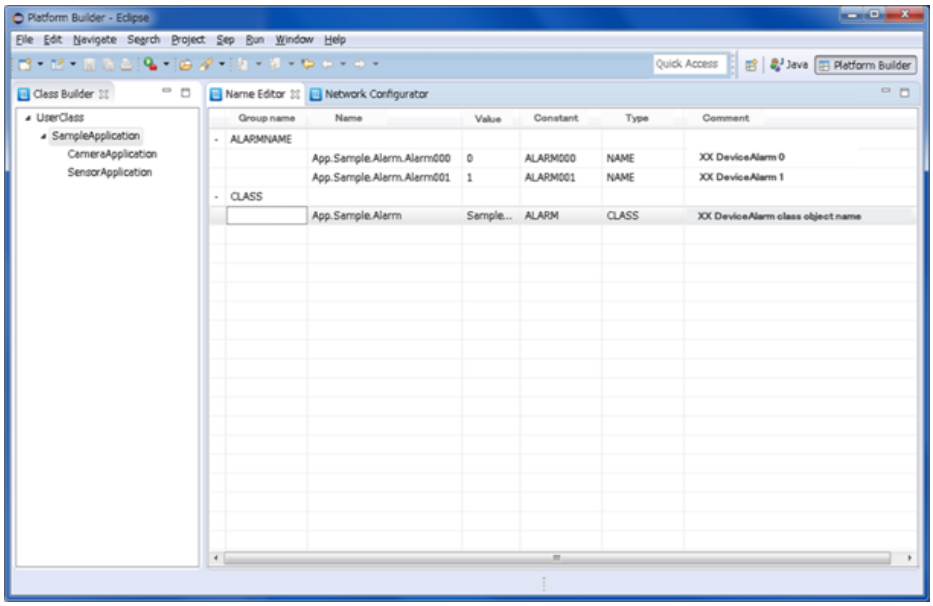


FIG. 2-1-2 Example of Name Editor screen

2-3 :Class Builder (User Class editing function)

In advance of application development using SEP, this function can design the User Class for definition of Object used in the system. Procedure of User Class design before application program development is as follows.

(1) Define the class (User Class) for device and machine being access target of application program

In need of access to multiple device and machine, the necessary number of classes are defined. For class definition, with use of this function, among SEP-provided SEP Basic Class, select the base class for devices (e.g. SepPlc class for PLC based devices). Otherwise, select a device-manufacturer-supplied specific class (e.g. xxPLC class, etc.) to create new class (e.g. MachineABC class).

(2) Define the class of application program

In the application program using SEP, you can create the application class inherited from SepApplication Class in SEP Basic Class. With SEP application itself being as a single object, Method and member variables can be exposed (enabled to be used) to other application . (All I/F can be set as non-exposure, or can be exposed only to the specific users with use of password and license key etc.)

(3) C source code automatic generation

For the classes defined at process (1) and (2), with use of this function, C source code (Definition file of User Class) is automatically generated.

(4) Programming of newly defined class

In accordance with a definition file of each class (e.g. MachineABC) created in (3), the program code being the entity of initiation process (Constructor), termination process (Destructor) and specifically defined Method (process) are added.

(5) Application program development

In application program development, the objects based on the classes generated in (1) to (4) are created, and the program accessing (Method execution, member variables obtainment / fixing etc.) to the created object (Device and machine) is created.

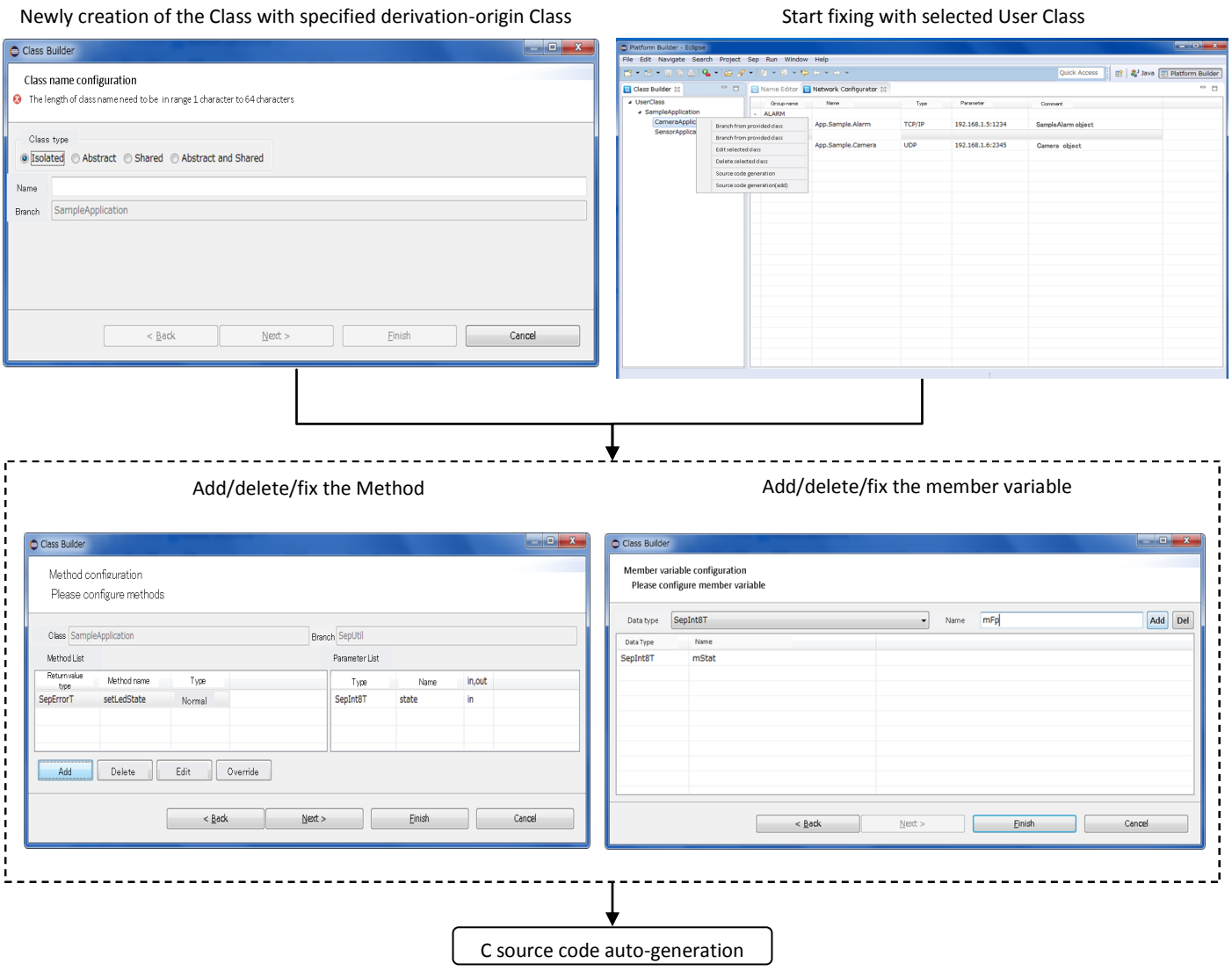


FIG.2-1-3 Design procedure with use of Class Builder

2-4:Log Viewer (Log displaying function)

This function is for debug of application using SEP. It's used with Ethernet connection between computer (WindowsPC, etc.) with SEP Platform Builder loaded and the target device with SEP application embedded. In SEP application, based on the log program deployed in various places in program (Describing with use of SepLog Class in Sep Basic Class), log data such as debug information is aggregated on a memory. The Log Viewer provides the following two functions to utilize the log information.

(1) Real time log displaying function

* Log display with filter function

Attachment information including level and type (error, alarm, information etc.) can be allocated to log information. This function enables filter display with specified level and type. Among the log information you can refine the required information to display.

* Times synchronization display of multiple device log information

Log information has logging time information. After merging the log information coming from multiple devices with embedded SEP application, it can display in the order of logging time. You can debug programs on multiple device tracing in time series.

* File output of log information

Log information read from target device can be saved. CSV file output is also available so that log information can be read into spreadsheet calculating software.

(2) Log file displaying function

Based on the log information retrieved with file output function (1), log display with the same filtering function as (1) and time synchronized display of multiple device log information are available. Other functions are provided as follows.

* Log information search with specified condition such as key word, etc.

* Log information printing function

2-5:Object Caller (Debugging function)

This function is used for debugging Remote Agent Call which is called among applications using SEP.

* Displaying table of SEP Object on network

* Displaying information of selected SEP Object

* Execution of Remote Agent Cal with specified parameters

* Displaying execution results

2-6:Key Generator (key code generation function)

This function generates encryption key codes for SEP intercommunication via WAN/Internet between devices with loaded applications using SEP. (If no need of encrypted communication, this function needn't. For key codes generation, purchasing the SEP Network Server Package and Key license is required.)

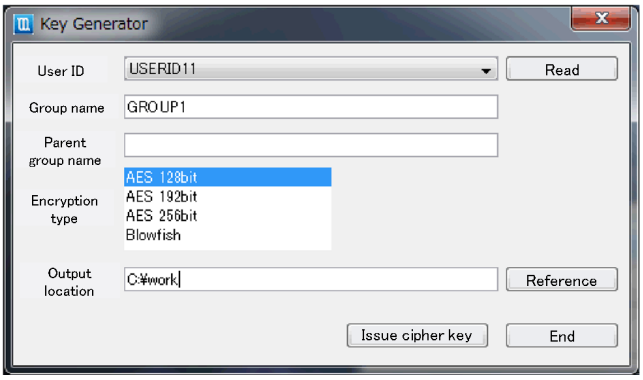


FIG. 2-1-4 Example of Key Generator screen

3: SEP Porting support package

This package is necessary for port development correspondent to platform (H/W, OS, communication target device etc.) of target devices to which SEP application is embedded. Note that when using target devices with SEP Custom Class Library, SEP is already embedded at the time of provision, port development using this package isn't required for customers. This package provides the following contents.

(1) SEPcore porting manual

* Porting procedure overview

* Explanation of pre-porting process
(Table of platform specification items to be determined in advance of porting)

* I/F function API for platform dependent layer
(Please implement the program correspondent to communication specification for H/W, OS of target devices and access devices along with this API.)

(2) Source program of SEP Basic Class Library

(3) Sample program of platform dependent layer

* Sample of CPL (Communication dependent layer)
A set of source codes of sample program for LAN connection between Windows PC

* Sample of PPL (H/W、OS dependent layer)
A set of source codes of sample program for Windows PC

Note: For porting VIO-Links included in SEPcore in accordance with the customer specific platform, customers are required to use VIO-Links binary providing service. For more details, please contact your sales or customer service representatives.

SEP operating environment

SEP Platform Builder (SEP embedded platform construction tool)

<Windows version>

Item	Name / Specification
CPU	Processor of 1.0GHz or over and Intel Pentium Ⅲ or greater
OS	Windows7 Professional 32bit/64bit 、Windows8.1 Pro 32bit/64bit
Language	Japanese/English
Memory	Windows7/8.1 32bit 2GB or greater, Windows7/8.1 64bit 4GB or greater
Monitor	1024x768, 32 bit color or greater recommended

<Linux version〉

Item	Name / Specification
CPU	Processor of 1.0GHz or over and Intel Pentium Ⅲ or greater
OS	Linux kernel 2.6.3x or greater
Language	Japanese/English
Memory	1GB or greater
Monitor	1024x768, 32 bit color or greater recommended

<MAC OS version>

Item	Name / Specification
CPU	Processor of 1.0GHz or over and Intel Pentium Ⅲ or greater
OS	Mac OS X 10.8
Language	Japanese/English
Memory	2GB or greater
Monitor	1024x768, 32 bit color or greater recommended

SEPcore (SEP embedded platform〉

Item	Name / Specification
CPU	ARM core MCU/SOC, CPU for embedded devices such as ATOM, PowerPC, CPU of WindowsPC
OS	SEPcore (Standard version with Server & Client function) Linux kernel 2.6.3x or greater Windows7 Professional 32bit/64bit, Windows8.1 Pro 32bit/64bit, Windows Embedded Compact7(*1) Various RTOS(µITRON(*2),TOPPERS,RTX,VxWorks etc.) (*1) SEP Client (Version with client function only)
Memory	Operable minimum structure (VIO-Links function) with 30Kbytes of ROM and 8Kbytes of work RAM memory.
Language	Language independent

(*1) Library which comes with the package is for Windows7/8.1, Linux and Android (Ver.1.3).
For excluded library, separately porting is necessary.

(*2) Already in use for µITRON specification based TOPPERS/ATK2.

Making possible operation on compact MCU like 16 bit microcomputer andµITRON or RTOS (Real Time OS), it has been already in use for Ethernet, Bluetooth, SIO, CAN, NFC as well as TCP/IP,UDP in communication transport layer.

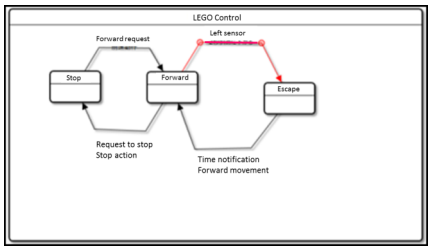
■ SEP-compliant application development environment (INTAWORLD)

Application using SEP is generally developed with C language program. Using the separately sold development environment, INTAWORLD (INTAstudio and INTAcore in the following diagram) in combination allows the development of application software without requesting the development (programing) from system designer to programmer (i.e. in a programmer-less way).

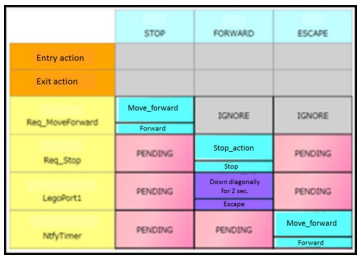
＜Control application designing function＞

(1) Design of state transition diagram and state transition table

Design the control logic with use of state transition diagram and state transition table. The result of editing them will be interactively reflected.



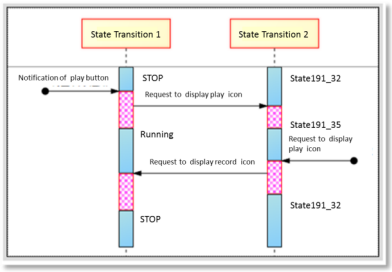
State Transition Diagram



State Transition Table

(2) Design of state sequence diagram

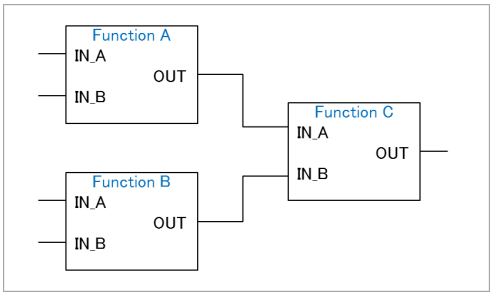
Design the sequence action between multiple state transition specified in (1). Consistency checking between sequence diagram and state transition diagram / table is available.



State Sequence Diagram

(3) Design of Function Block Diagram (FBD)

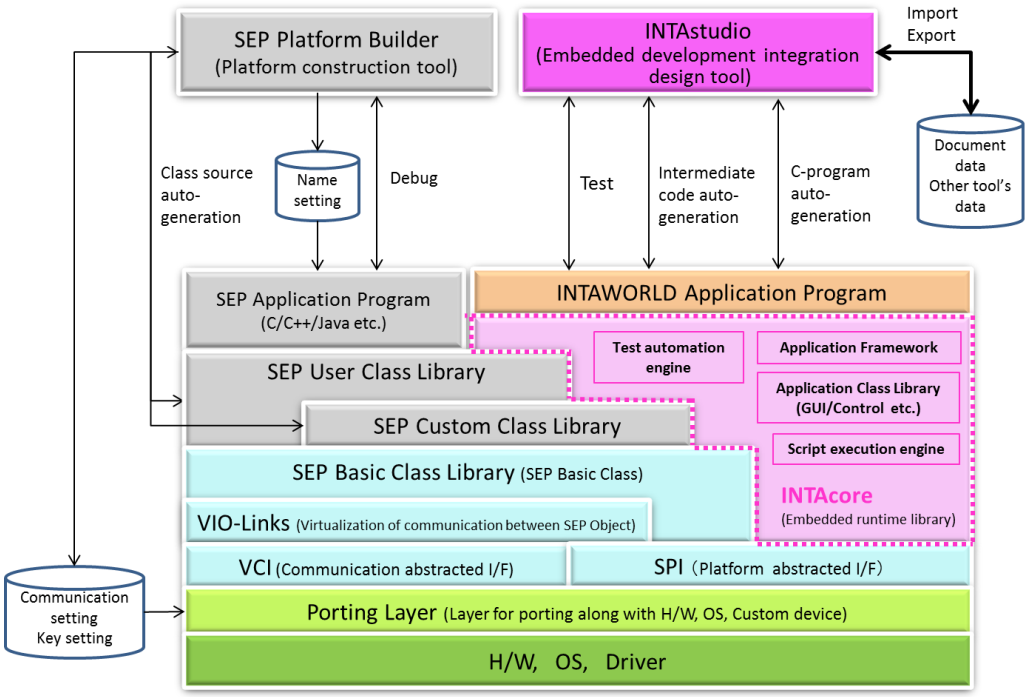
Design the control program by connecting input / output of Function Block.



Function Block Diagram

(4) Automatic generation of the control program

Function to auto-generation of the control program based on the design data and action scenario created in (1)(2)(3).
In the case of middle-code generation, the auto-generated program is transferred to actual machine for prompt execution.
(IDE such as compiler not required.)



Glossary

NO.	Terminology	Definition
1	SEP	Abbreviation of Smart Embedded Platform. General term of this merchandise (product).
2	SEPcore	Refers to The Embedded Software Platform in SEP.
3	SEP Platform Builder	Porting development of SEPcore to target devices, and development environment for preparation of the application software development.
4	SEP Basic Class Library	Class library providing the basic function required for user's application software development.
5	SEP User Class	User-created Class which inherits the Class defined in SEP Basic Class.
6	VIO-Links®	Deployed between the processing part for dedicated access to individual device and user application software, the mechanism allowing the access from the user application software regardless of the difference in device specification.
7	CAL	Abbreviation of Communication Abstraction Layer.
8	VCI	Abbreviation of VIO-Links Cal Interface.
9	CPL	Abbreviation of Cal Porting Layer
10	PAL	Abbreviation of Platform Abstraction Layer
11	SPI	Abbreviation of SEPcore Pal Interface
12	PPL	Abbreviation of Pal Porting Layer
13	SEP protocol	The communication protocol between Application Object with embedded VIO-Links®
14	SEP world	General term of the world which consists of Object and Application Object connected via VIO-Links®.
15	Network Configurator	The function which sets the Name of access target and Network parameter required for accessing.
16	Name Editor	The function defines Name for various use in application program, such as association of Name specified in the application software and the fixed name of connection target device,
17	Class Builder	EDITOR function which edits the user-defined Class inheriting SEP-provided SEP Basic Class.
18	Object	One of the feature of the object oriented programing. Physical / abstract entity subject to the S/W operation in the real world is modelled as a set of attribute (data) and operation (method), and reproduced on a computer.
19	SEP Application Object	With VIO-Links® mechanism, application discloses I/F accessible from the outside, and it enables application software itself to behave as a single Object. This type of application software is called SEP Application Object.
20	encapsulation	One of the concept that constitutes the object-oriented. Concealing the data in-Object, behavior of the Objet, the actual type of the Object etc.
21	Abstraction	Creating high level concept which to group together various kinds.
22	Seamless	Consistent operation (e.g. data taking) in the application software is allowed.
23	Platform	Base foundation of a computer and system, which refers to hardware and OS in general.
24	Multiple platform	Being correspondent to multiple OS and hardware.
25	Class	Template of an object which consists of data and method (operational procedure) defined in the ob-
26	tunneling	When the protocol for ordinary use is not available due to the constraint such as firewall, wrapping the original packet with other protocol's packet for communication.
27	Encryption of communication	Message contents in the communication packet is encrypted with Key which is known only by transmission source and destination for ensuring safety without being intercepted or altered in transit.
28	Remote Agent Call®	The mechanism utilizing the function of distant Object such as network, memory, serial communication etc. with use of VIO-Links®.
29	Secure communication	The communication securing confidentiality with encrypted communication protocol for enhancing the security. SEP provides secure communication such as challenge-response, AES etc. for one-on-one device correspondence.

URL of SEP http://www.ilc.co.jp/commodity/sep/

International Laboratory Corporation

Headquarters Yusen Building 7F, 2-3-2 Marunouchi, Chiyoda-ku,
Tokyo-to, 100-0005, JAPAN
Phone: +81-3-3287-7700 Fax: +81-3-3287-3999

Design Center Hiroshima-matoba Building 9F, 1-3-6 Matoba-cho,
Minami-ku,Hiroshima-city, 732-0924,JAPAN
Phone: +81-82-262-7700 Fax: +81-82-263-4411

For technical inquiries Phone: +81-82-262-7799 Fax: +81-82-263-4411
FATCICS Technical Center E-mail : fa@ilc.co.jp
(Office hours) Monday — Friday (except public holidays) 9:00-17:00

★ Please acknowledge that the description of this catalog might change without a previous notice.
★ Please read the operation manual carefully and use this product correctly.
★ The products provided on this catalog are intended the Japanese market. When using the product in a country other than Japan, please contact us separately.
★ FATICS, INTAWORLD, INTAstudio and INTAcore are registered trademarks of International Laboratory Corporation.
★ Other company's product name (software / hardware) described on this catalog is trademark or registered trademark of respective company.

Precautions

Before using the product in this catalog, please read the supplied manual and all related manuals carefully in order to use the product properly with the enough payment of attention to safety and the correct handling.

BF932-038J-E 201804