

SEP

Smart Embedded Platform

ハードウェア、OS、通信の違いを意識することなく、世界中のMachineとMachine (M2M)をどこでも誰とでもシームレスにつなげる世界初の次世代志向スマート制御プラットフォーム。それがSEPです。

「世界を動かす真ん中に」

SEPは今後、電気で動く世界中のMachineとMachineをつなぐことになります。

そこでロゴは、Machineの「m」をモチーフに、コードとコードを接続した部分をイメージしたカタチとしてデザインしました。

色は青。世界中をつないでいる「空」と「海」の色に、SEPの未来を重ねあわせています。

カタログはそんなSEPがつなぐ世界をイメージし、コードがどこまでも伸びていくデザインとなっています。

SEPは、機器同士が直接会話する真のm2mを実現します。

近年では、各機器(Machine)どうしをクラウド経由で通信するというクラウド型M2Mや、各機器の情報をクラウドシステムへ収集し、その収集されたデータ(Big Data)を分析・解析することで様々な業務効率化を行うといったIoT/IoEソリューションの話をよく耳にするようになりましたが、ILCはM2Mの原点に立ち戻り、機器と機器をクラウドシステムを介さず、直接・簡単につなぐことでM2Mの新たな付加価値を創造します。ILCが提案するM2Mとは、各機器どうしが直接会話する世界の創造であり、機器と機器、機器と人など複数をつなげた新たなシステムを作ることで、機器単体や個人ではできなかった新しいシステム(即ち、スマート社会システム)を創造することだと考えます。SEPは、つなぎ方、制御する相手、実行するプラットフォームなど、つなぎの違いや機器の違いといった個々の違いを越えて機器どうしが制御(連携)できる仕組みを提供します。(クラウド型M2Mとは違うM2Mという思いを込めて、ILCが提案するこのM2Mは **m2m** と表記します。)

商品の特長	3
適用例	4
SEP Basic Class Library 一覧表	8
商品構成	10
SEPcore (SEP組込プラットフォーム)	10
SEP Platform Builder (SEP組込プラットフォーム構築ツール)	12
SEP ポーテイングサポートパッケージ	14
動作環境	15
SEPアプリケーション開発フロー	16
SEPメーカーカスタマイズ機能	17
SEP Ver.2.10の新機能	17
関連用語集	19

<SEP (Smart Embedded Platform)とは>

SEPは、多数の機器やコンピュータで構成された分散型組込システムにおいて、ソフトウェア開発者が機器間の通信処理を意識することなく、あたかも1台の機器やコンピュータ上のソフトウェア開発と同じ感覚で開発することができる仕組みを提供します。SEPが提供するスマート組込プラットフォームとは、個々のハードウェアやOSの違い、機器間の通信を意識することなくシームレスにつなぐことを可能とし、組込システムのm2m/IoT/IoE化開発を強力に支援するソフトウェアプラットフォームです。

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

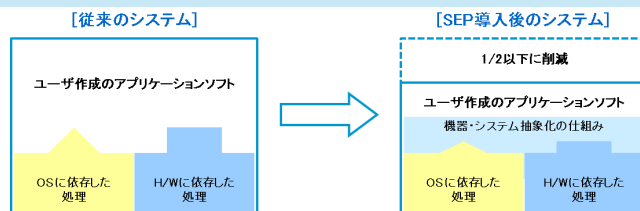
POINT 1 Safety feature and Secure communication (安全性強化の仕組みとセキュア通信機能)

組込システムにおいても、機器間のセキュアな通信機能は最重要課題の一つになっています。SEPでは、CPUパワーの限られた非力なプラットフォームから、強力なCPUを搭載したハイエンドプラットフォームにいたるまでのいずれにおいても、SEP組込機器間をつなぐためのSEP通信プロトコルの上で搬送されるデータを保護するため、簡易なBlowfish暗号から強力なAES (256bit)暗号を標準でサポートしています。(ユーザ独自の暗号アルゴリズムの組込も対応可能な仕組みを提供します。) さらに、チャレンジレスポンスによる動的鍵交換や暗号強度のエスカレーション機能の提供、セキュリティ喪失の検出と対策機能など、セキュアな通信機能を提供します。



POINT 2 Compatible and Scalable (プラットフォームに依存しない互換性と可搬性)

SEPは、ハードウェアやOSなどプラットフォームに依存する部分を分離し、プラットフォーム変更時のアプリケーション層への影響を最小限に抑える仕組みを提供するため、システムリプレイス時の開発コストを大幅に削減します。



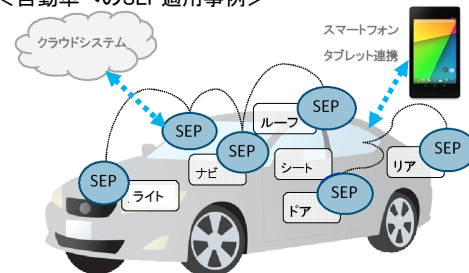
POINT 3 Speedy & Easy control (制御システム開発に必要な機能がAll in One.)

SEPが提供する豊富なクラス(ソフトウェア部品集)を活用して制御システムを簡単に開発することができます。クラウドや様々な機器との通信処理を提供するクラス、デバイスドライバ処理を提供するクラス、機器の制御に必要な処理を提供するクラス、マルチメディア処理を提供するクラスなど、アプリケーション開発に必要な様々なクラスを用意しています。

POINT 4 Next generation integrated control system (機器と機器、システムとシステムを一つに統合)

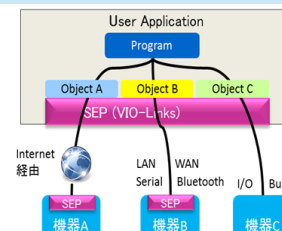
SEPの通信仮想化機構であるVIO-Links (Virtual Object Linking System)により、制御システム内のマイコン間通信、CPUコア間通信はもちろん、スマートフォン・タブレットなどのスマートデバイスとの連携や、クラウドシステムとのデータ共有などインターネットまで広がるシームレスな世界を実現します。VIO-Linksは通信形態の違いを吸収し、レガシーな機器も含んだ複数の機器、複数のシステムを名前指定だけで簡単に接続することができるため、機器間やシステム間の通信プログラム開発が不要となります。

<自動車へのSEP適用事例>



POINT 5 Microcomputer to Microcomputer (分散型組込システムをシームレスに連携)

VIO-Linksで機器間をつなぐと、通信形態や機器構成を意識することなく、接続先機器のクラスメソッドを、あたかも自分の機器上に存在するように呼び出すことができますようになります。この仕組み(Remote Agent Callと呼びます)を利用することで、通信形態や機器構成をカプセル化し、分散型組込システムのアプリケーションをシンプルに構築できるようになります。また、システムの機器構成の変更や、通信形態の変更にも瞬時に対応できるようになります。

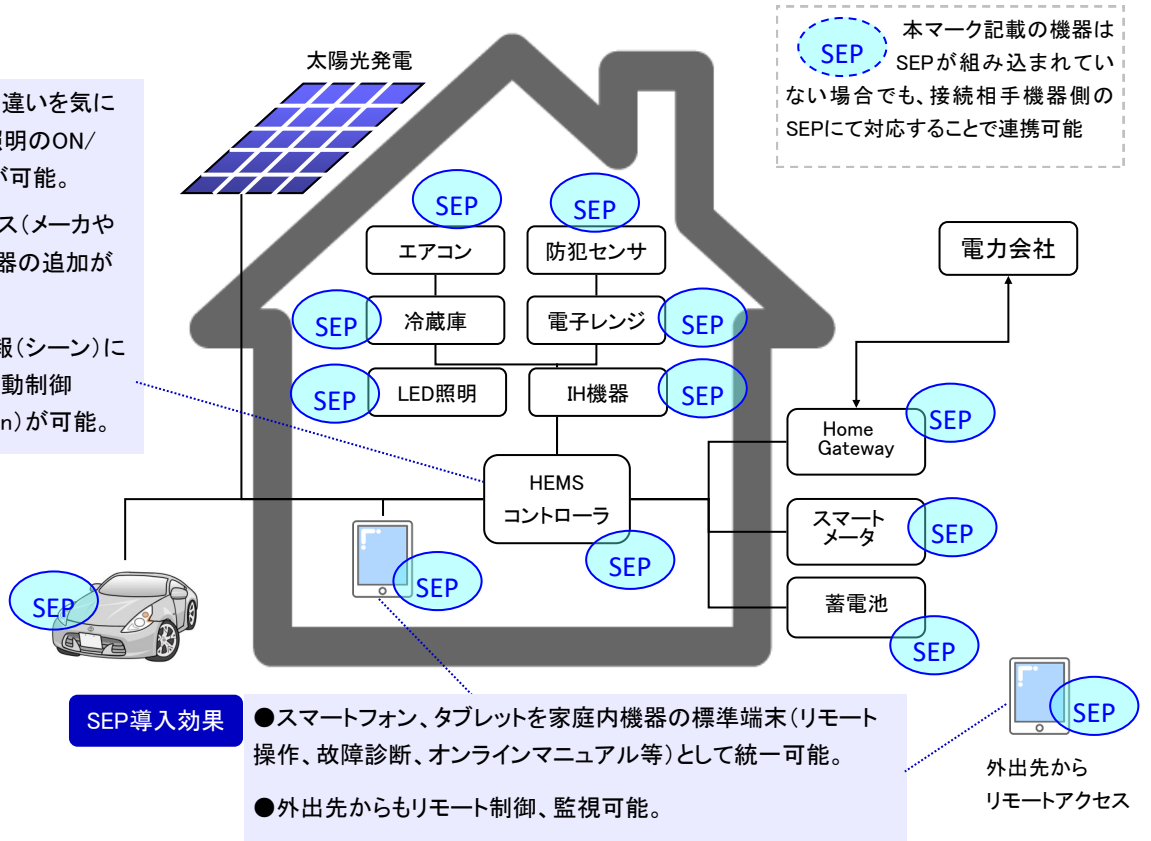


POINT 6 Low cost and High quality (飛躍的な工数削減効果による開発コスト削減と高品質の実現)

ILCが品質を保証するSEPのソフトウェア部品集(SEP Basic Class Library)と開発ツール(SEP Platform Builder)を活用することで、制御システムの開発工数、テスト工数、保守工数を大幅に削減する(当社比較にて導入前の1/5にまで削減)ことができます。

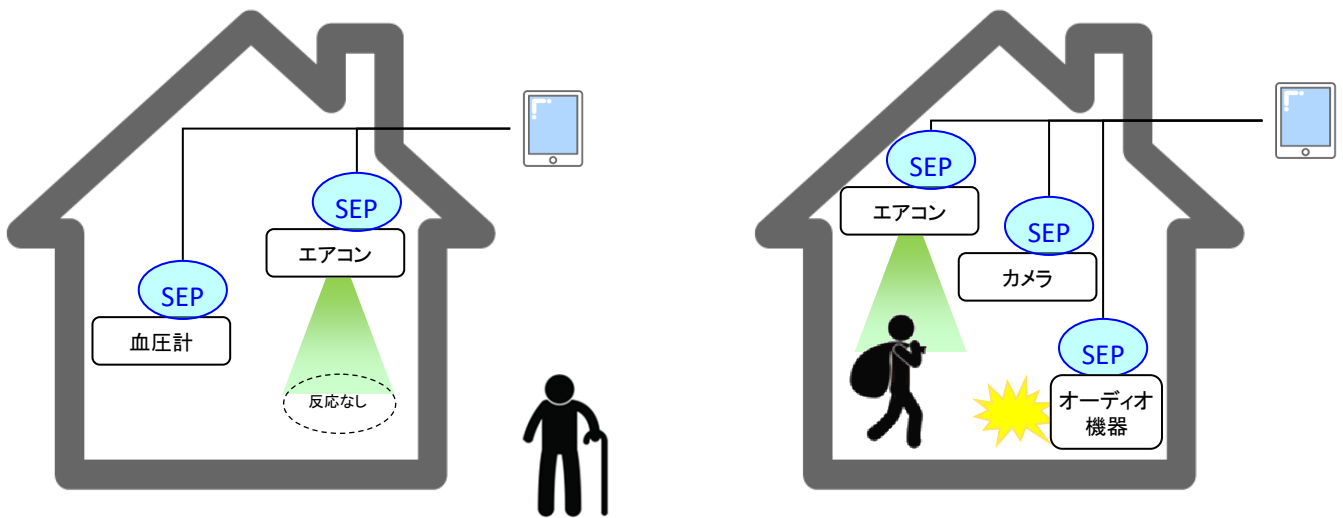
* 適用例1 スマートホーム(Home Automationシステム)への適用

- メーカーや型式の違いを気にせず、家電や照明のON/OFFや設定変更が可能。
- 機器のリプレイス(メーカーや型式の切替)、機器の追加が簡単。
- 家の内外の情報(シーン)に基づき、住宅内自動制御(Home Automation)が可能。



HEMS: Home Energy Management System

SEPが搭載された機器を組み合わせた動作を、ユーザ(住人)が設定することも可能。



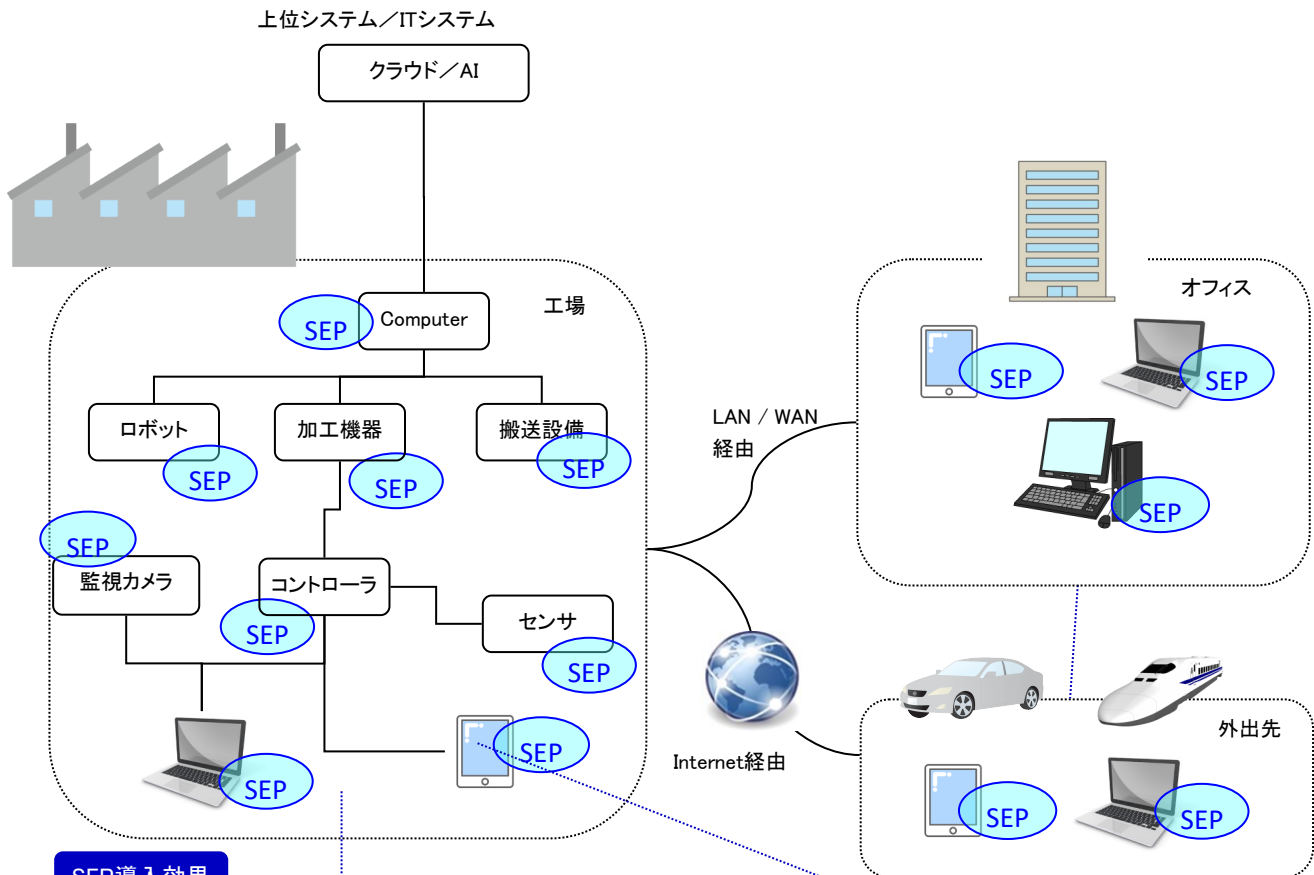
動作例: お年寄りの見守り

- ・血圧計の情報をスマートホンに送信
- ・エアコンの人感センサの反応が長時間なくなれば、徘徊の可能性をスマートホンに通知

動作例: 留守中のセキュリティー

- ・エアコンの人感センサが反応するとカメラで撮影し、画像をスマートホンに通知
- ・オーディオ機器から音を出し、不審者に警告

* 適用例2 スマートインダストリ (Factory Automationシステム)への適用



SEP導入効果

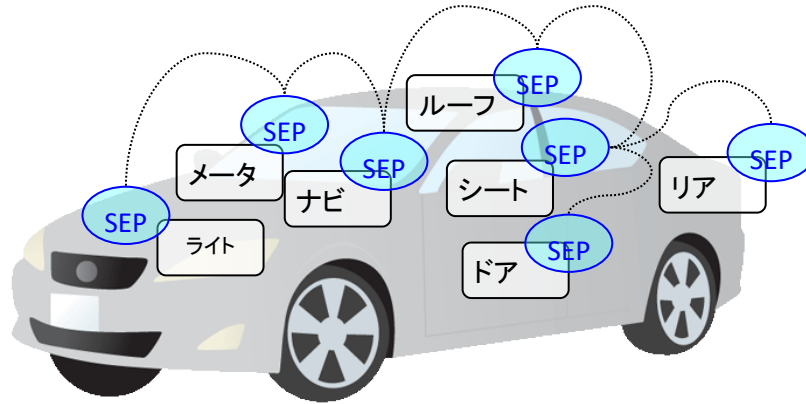
- メーカーや型式の違いを気にせずに、工場内の機械や装置を一元的に情報収集、異常監視、制御可能。
- 各機器間をセキュアに接続可能。
- 機器のリプレイス(メーカーや型式の切替)、機器の追加が簡単。
- REST/MQTTプロトコルによるクラウド/AIシステムへのアクセスが可能。

SEP導入効果

- スマートフォン、タブレットを工場内の全ての機器の標準端末(リモート操作、故障診断、オンラインマニュアル等)として統一可能。
- オフィスや外(世界中)からでも監視&情報収集可能。
- 工場内と外部のスマートフォン、タブレットとセキュアな通信が可能。
- スマートフォン、タブレット上で動作するアプリケーションを簡単に快活可能。

* 適用例3 スマートカー(自動車内マイコンシステム)への適用

自動車内で使用されている各マイコン間を連携 (車載マイコン間ネットワークを意識しないスマートカー制御システム開発)



① Head UnitとRear Unitとの接続

自動車の後部座席にあるRear Unitは、運転席側にあるHead Unit(カーナビなど)に組み込まれている音楽再生、動画再生、ナビ、電話機能等を使うことができる装置です。

Rear Unitのソフトは、Head Unitに組み込まれている音楽再生などの各種機能をUnit間通信を介して利用する必要がありますが、SEPを導入することで、従来のようにUnit間接続方式や通信プロトコル設計、通信処理プログラムを開発することなく、言い換えれば、Unit間通信を意識することなく、Head Unitの各機能呼び出すことができるようになります。

② スマートフォン・タブレットとの接続

最近の自動車では、スマートフォン・タブレットをWiFi、Bluetoothで車載システムと接続して、Head Unitの音楽再生機能や動画再生機能のリモート操作ができます。

このようなシステムを開発する場合、SEPを導入することで、スマートフォン・タブレット側のソフトやHead Unit側のソフト開発において、従来のように通信プロトコル設計や通信処理プログラムを開発することなく、両機器間の機能連携ソフト開発できるようになります。

通信プログラム開発不要とは、例えば、スマートフォン上のソフト開発において、無線でつながっているHead Unit内の音楽再生機能を通信プログラムを作らないで実行できることを意味します。

③ スイッチ・センサの接続

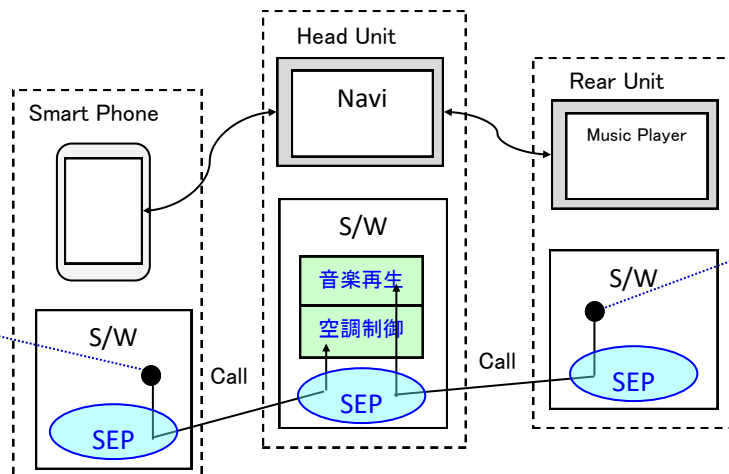
自動車内に組み込まれている各種スイッチやセンサは、今まではLINやCANなどの専用ネットワークで接続されていましたが、安全性への関与が低いスイッチやセンサに対しては、配線が不要なBluetoothやWiFiなどのワイヤレス接続も検討され始めています。

CANからWiFiへの変更といった通信接続の方法を変更する場合においても、SEPを導入することで、アプリケーションソフトのプログラム変更(手直し)を最小限に抑えることができます。

現状の課題	SEP導入効果
類似機能の重複開発により開発工数が増大 (例: RearUnitの音楽再生機能開発)	他Unitに搭載されている機能を簡単に利用できるため、 類似機能の重複開発が不要。
スマートデバイスの活用のために追加開発が必要	スマートデバイスから各Unitの搭載機能を簡単に利用できるため、 開発が簡単。
通信方式変更の度に開発が必要 (例: CANからWi-Fiへの変更)	通信方式を変更してもアプリケーションは 変更不要。

SEP導入効果

- スマートフォン上に空調制御機能は非搭載だが、Head Unit上の空調制御機能を簡単に利用可能。
- Rear Unitが搭載されない車種ではスマートフォンで同等の機能を提供可能。

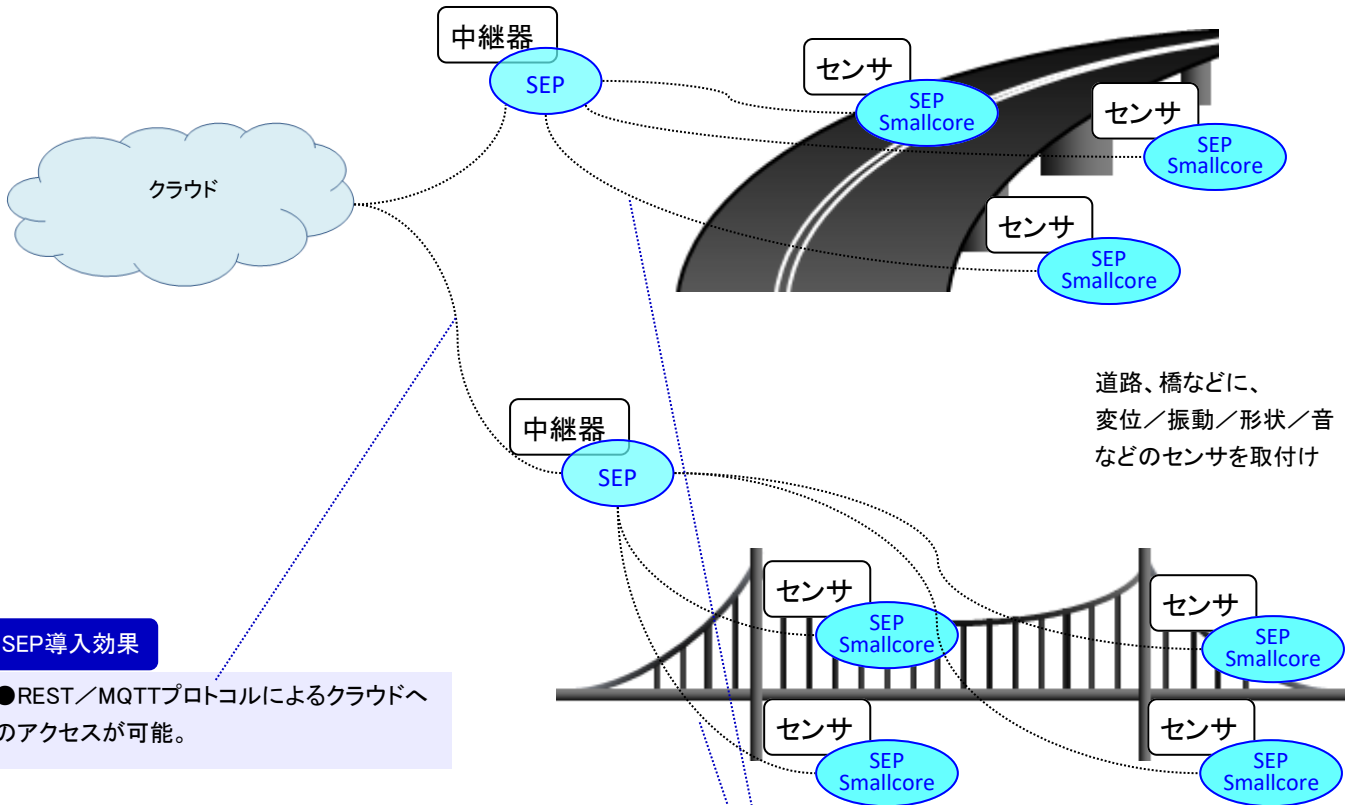


SEP導入効果

- Rear Unit上に音楽再生機能は非搭載だが、Head Unit上の音楽再生機能を簡単に利用可能。(動画再生、空調制御など他の機能も同様に利用可能。)

* 適用例4 社会インフラ への適用

社会インフラ(道路、橋など)に取り付けられた大量のセンサの情報をクラウドで収集し、劣化などを診断。



SEP導入効果

●REST/MQTTプロトコルによるクラウドへのアクセスが可能。

SEP導入効果

- センサと中継器の間をセキュアに接続
 - メーカー・機種異なるセンサであっても、中継器に接続可能。
 - センサのようなリソースが限られた環境であっても、データ送信アプリケーションを搭載可能。
- それによりセンサの消費電力が抑えられ、センサが長期間動作可能。

SEP Basic Class一覧表 (その1)

NO	分類	Class名	機能概要
1	基本的な処理を提供するクラス	SepObjectBase	SEPのすべてのクラスの基底となるクラス
2		SepObject	SEPの全てのクラスの基底となるクラス、全てのクラスは、SepObjectを継承する。SepObjectを継承したクラスは、公開範囲(スコープ)を決めて、メソッドを公開することができる。
3		SepEventSource	イベントを通知するためのクラス。基本的に、イベントは、Callbackで通知する。通常、通知される側は、SepEventを継承する必要はない。イベントは、特定のメソッドを登録して、メソッドを実行した結果ないしは状態変化等のトリガ条件に対して通知を受けるためのものであるが、ブロードキャストされるPublicイベント受信に用いることも可能で、その場合は、Publicイベントを受ける側もSepEventSourceクラスを継承する必要がある。
4		SepMedia	動画・静止画・音声などのマルチメディア機能の基底クラス。このクラス自体が単独で用いられることはない。
5		SepVsensor	センサー機能を抽象化したクラス
6		SepController	PLC/Motion Controller/CNCなどFA分野制御コントローラの基底クラス。このクラス自体が単独で用いられることはなく、このクラスを継承した各社製コントローラ対応のクラスを用いてアプリケーションを開発する必要がある。
7		SepApplication	アプリケーションとして、メソッドを動的に公開する場合に継承が必要となるクラス。このクラスを継承しなくても、他の公開メソッドを使用することはできるが、メソッドを公開する場合には、このメソッドを継承しないしは実行する必要がある。
8		SepSystem	SEPのシステム情報を取得する、システム設定を変更するためのクラス。Keyを指定して個別情報を取得・設定するほか、一括して取得・設定することもできる。
9		SepSecurity	暗号などのセキュリティ情報を扱うクラス。ブロック暗号や連続したストリーム暗号にも対応する。独自暗号もCPL(Cal Porting Layer)で実装することができる。
10		SepNetwork	物理・論理媒体に依らない通信を提供するクラス。IPを用いた通信だけでなく、Bluetooth、RS232C、USBや共有メモリを使用する通信についても抽象化を行い双方向通信する機能を提供する。相手のIPが不明な場合でも、Nameを指定することで直接通信することもできる。
11		SepNameResolver	Name解決処理を提供するクラス。抽象的な表現であるNameを実体に変換する。検索先の範囲(スコープ)を設定して検索を実行できる。
12		SepMail	電子メールを送受信する機能を提供するクラス
13		SepFileUtil	ファイルアクセスするためのクラス。ファイル処理は、言語・OSが提供する機能を利用することもできるが、このクラスを用いることで、環境依存性を無くすだけでなく、遠隔ファイルにアクセスすることができる。
14		SepFileUtilAsync	SepFileUtilクラスの各メソッドを非同期化したクラス
15		SepUtil	ユーティリティ機能を提供するクラス。バージョンの取得だけでなく、言語・OS・環境に依存する処理を汎用化(ラップ)して提供する。
16		SepExec	処理を実行するクラス。他アプリケーションの起動などを行う。
17		SepScaleTransform	スケール変換処理を行うクラス。変換式を用いて任意の数値・単位への変換をおこなうことができる。
18		SepStatistics	統計処理用クラス
19		SepScript	論理式や演算式を処理するクラス
20		SepCalendar	カレンダー機能を提供するクラス
21		SepClock	時計機能を提供するクラス
22		SepJsonparser	Jsonのパースを提供するクラス
23	クラウド連携処理を提供するクラス	SepCommRest	REST-APIによる通信を行うためのクラス
24		SepCommMqtt	MQTTによる通信を行うためのクラス
25	データ共有の仕組みを提供するクラス	SepDatabase	データベースを扱うためのクラス。データベース自体は、ネットワークで接続されたサーバに存在することが前提で、このクラスでは、Thin Clientを用いて機能を実現するため、クラス自体にデータベース機能は持っていない。
26		SepDataSaver	データ共有(簡易データベース)処理を提供するクラス。
27	マルチメディア処理をサポートするクラス	SepStreamer	ストリームデータを扱うためのクラス。ファイル処理で扱うことができない不定長のストリームデータを処理するためのクラス。オブジェクト間で不定長のデータをやりとりすることもできる。
28		SepCamera	カメラ機能を抽象化したクラス。このクラスは、カメラを制御する機能を持つが、静止画のEncode/Decode機能は持たない。
29		SepVideo	Videoストリームを抽象化したクラス。このクラスは、Videoストリームを制御する機能を持つが、動画再生機能は持たない。

SEP Basic Class一覧表 (その2)

NO	Class名	派生元	機能概要
30	制御に必要な処理を提供するクラス	SepSimpleVio	I/Oアクセス処理クラス。機器のアクセス手段の違いを意識することなくI/Oアクセスする仕組みを提供する。各種I/Oボードなどは、このクラスから派生させて固有の機器クラスを定義する。Portへの設定・読み書き等は、1ポートを指定することも可能だが複数ポートに対して一括で実施することもできる。
31		SepVio	SepSimpleVioクラスに排他ロックやインタロックなどの安全制御機能を追加したクラス
32		SepTimer	指定時間満了後にイベントを発生するタイマ処理クラス。指定時間後に一回だけ呼び出すか周期的に呼び出すかは指定可能。タイマの指定単位は1ms単位で指定可能であるが、WindowsやAndroidのような非RTOS上に搭載された場合のタイマ精度はプラットフォームやアプリケーションに応じて変化する。
33		SepPowerMng	電源を扱うためのクラス
34		SepConfig	各Objectの設定等を保存・復元するためのクラス。設定の保存・復元は行うが、設定を用いた処理はObject側に実装する。
35		SepDiag	診断機能のハンドリングを提供するクラス。個別の診断機能は、ユーザがSEPcore移植時にPorting Layerで実装しなければならない。
36		SepDataCollection	登録した指定のName Listのデータの現在値を、指定の条件(定周期、イベント発生時)で取得する機能を提供するクラス。データログ処理に使用する。
37		SepDataDelivery	登録した指定のName Listのデータの値を、指定の条件(定周期、イベント発生時)でそれぞれ指定の値に変更する(書き換える)機能を提供するクラス。システムの段取り替え処理に使用する。
38		SepDiagControl	制御機器向けの診断処理を提供するクラス。指定のオブジェクトのON/OFF回数、累積時間や許容値オーバー回数などの診断処理を行うことができる。
39		SepInterval	指定の時間間隔で指定のNameの値をON/OFFするクラス
40		SepEventLogger	イベントログを扱うクラス
41		SepAlarm	アラーム監視用クラス
42		SepAlarmHistory	アラームオブジェクトの履歴管理クラス
43		SepAlarmList	アラーム履歴のリスト管理クラス
44		様々な機器へのアクセスをサポートするクラス	SepPrinter
45	SepPlc		PLC(Programmable Logic Controller)とのI/F機能を抽象化したクラス
46	SepMotion		Motion ControllerとのI/F機能を抽象化したクラス
47	SepInvertor		インバータとのI/F機能を抽象化したクラス
48	SepCnc		CNCとのI/F機能を抽象化したクラス
49	SepBarcode		バー(QR)コードリーダ機能を抽象化したクラス
50	SepNfc		NFCリーダライタ機能を抽象化したクラス
51	SepCommModbusTcp		Modbus(TCP)による通信を行うためのクラス。通信パケットはユーザが定義でき、その内容に基づき通信を行う。
52	SepCommModbusRtu		Modbus(RTU)による通信を行うためのクラス。通信パケットはユーザが定義でき、その内容に基づき通信を行う。
53	標準の通信形式をサポートするクラス		SepCommGateway
54		SepCommGatewayRoutin	物理層が異なるネットワーク間の通信を行うための設定を行うクラス。
55		SepCommOpcuaClient	OPC-UAクライアントクラス。通信パケットはユーザが定義でき、その内容に基づき通信を行う。
56	デバイスドライバを提供するクラス	SepDriverCAN	CAN通信のドライバを提供するクラス
57		SepDriverEther	Ethernet通信のドライバを提供するクラス
58		SepDriverLIN	LIN通信のドライバを提供するクラス
59		SepDriverFRay	FlexRay通信のドライバを提供するクラス
60		SepDriverSpi	SPI通信のドライバを提供するクラス
61		SepDriverTimer	タイマへのアクセスを提供するクラス
62		SepDriverWatchDog	ウォッチドッグタイマへのアクセスを提供するクラス
63		SepDriverInputCapter	キャプチャしたデータへのアクセスを提供するクラス
64		SepDriverPwm	パルス幅変調器へのアクセスを提供するクラス
65		SepDriverAdc	A/D変換器へのアクセスを提供するクラス

■ 商品構成



1-1 : SEPcoreの全体概要

SEPcoreとは、SEPが提供する機能を活用するシステム(アプリケーションソフト)に組み込まれる機器組込プラットフォームです。

SEPcoreは、大きく分けると3つの機能層で構成されています。(図1-1-1参照)

- ① SEP Basic Class Library
- ② VIO-Links
- ③ 移植層インターフェイス

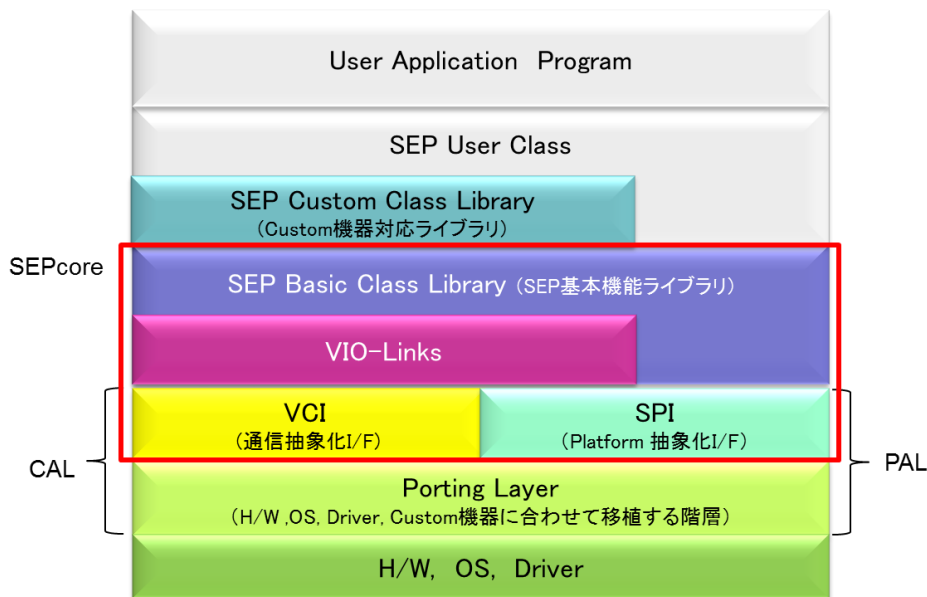


図1-1-1 SEPアプリケーションのソフトウェア構成図

・CAL (Communication Abstraction Layer)

VCIとVCIIIに基づくターゲット機器との通信処理の実装部を合わせた部分をCALと呼びます。SEPでは、CALの標準機能としてTCP/IP、UDP、Bluetooth、仮想シリアル通信の実装部も含めて提供します。

・PAL (Platform Abstraction Layer)

SPIとSPIIIに基づくターゲット機器のH/WやOSなどプラットフォームに依存した処理の実装部を合わせた部分をPALと呼びます。SEPでは、PALの標準機能として、Windows、Linux、Android対応の実装部も含めて提供します。

1-1-1: SEP Basic Class Library

SEPが提供する機能を活用したシステム(アプリケーションソフト)を開発するために必要な基本機能を提供します。

各メーカーの機器・装置の独自機能を用いたシステムを開発するためには、本Class Libraryを継承し、各機器や装置とのインターフェイス処理を実装したSEP Custom Class Libraryを作ります。

例えば、カメラ機能を提供するSepCameraでは、「撮る」「見る」といったカメラの基本機能を提供するのでカメラとして必要最小限の機能を提供しますが、メーカー独自の機能であるスマイルシャッターなどの機能に対応するためには、本Class Libraryを継承して、スマイルシャッター機能を持つSEP Custom Class を作ります。

SEPアプリケーション開発では、SEPの開発ツールSEP Platform Builderに含まれるClass Builderを使用してSEP Basic Class やSEP Custom Classから派生させたユーザ定義クラス(SEP User Class)を利用します。(図1-1-2を参照)

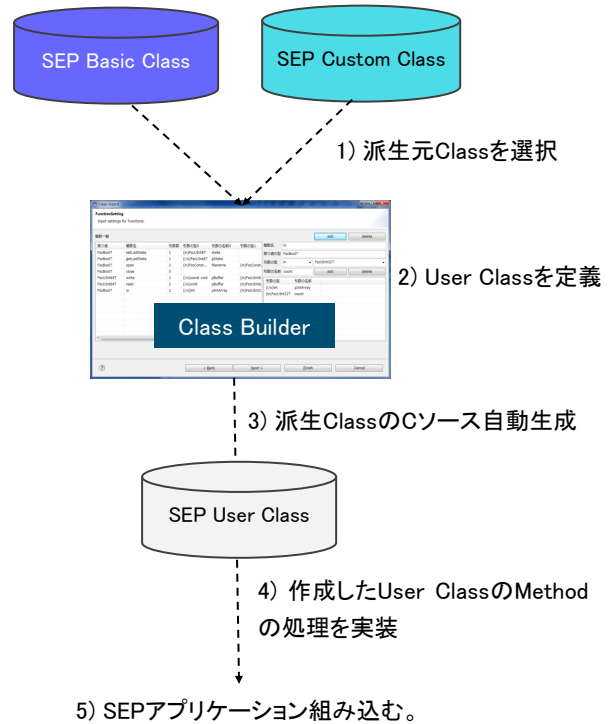


図1-1-2 SEP Basic Class Libraryの利用フロー

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

VIO-Linksは、「機能を使いたい側」と「機能を提供する側」に分離し、双方が離れていても、その間の通信を仮想化して、あたかも同じ装置の中の機能を使っているように見える仕組みです。VIO-Links自体は、SEPcoreの中で、SEP Basic Class Library、SEP User Class/Custom Classと、移植層インターフェイスの間に位置し(図1-1-1参照)、アプリケーションソフトからアクセスするオブジェクト(機器やシステム)が、ローカルにある(アプリケーションソフトと同一コンピュータからアクセスする)のか、あるいはリモートにある(ネットワークでつながった他のコンピュータ経由でアクセスする)のかを意識せず、更にはオブジェクトとの通信手段やプラットフォーム(H/WやOS等)の違いも意識することなく、シームレスにアクセス可能とする位置透過性を提供しています(図1-1-3参照)。

この仕組みにより、User Application Programでは実体(図1-1-3の場合では機器A/B/C)にアクセスするための詳細な手段を意識することなく(知らなくても)プログラム開発が可能となります。また、Object変更(図1-1-3の場合では機器A/B/Cの切替)の場合には、User Application Programを変更することなく、Network Configuration(通信設定)、Name(アクセス先の定義)、User Class(Object定義)の派生元Classの変更だけで対応できます。

VIO-Links Liteは限られたリソース環境向けのVIO-Linksで、ROMサイズ:30KB、RAMサイズ:10KB、OSなしの環境でも動作してします。「Smart Embedded Platform Lite」として提供します。なお、VIO-Links LiteをSEP Smallcoreとも呼びます。

1-1-3: SEP移植層インターフェイス (Porting Layer Interface)

SEPcoreを搭載する機器のプラットフォーム(H/WやOSなど)に合わせて移植するためのインターフェイス層で、大きく分けると次の2つのインターフェイスが提供されます。

- 通信に依存した機能を実装するためのVCI (Vio-links Communication abstraction layer Interface)
- Platform(H/WやOS)に依存した機能を実装するためのSPI (SEPcore Platform abstraction layer Interface)

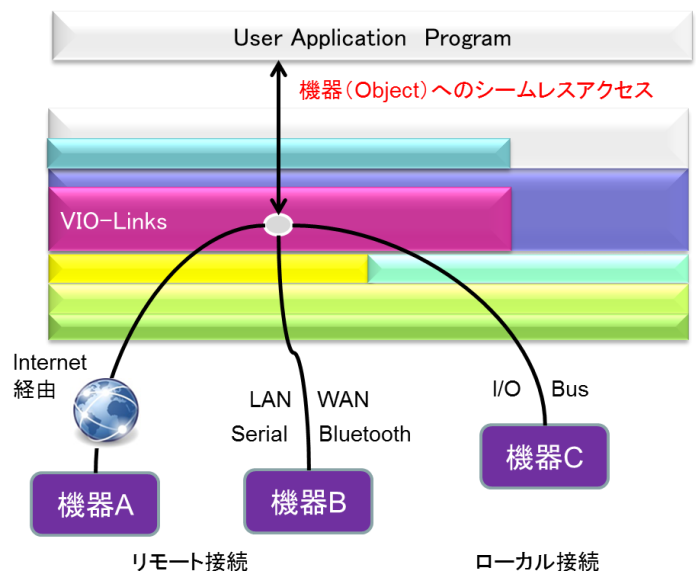


図1-1-3 VIO-Linksの機能説明図

1-2: SEP Platform Builderの全体概要

SEP Platform Builderは、SEPを組み込んだアプリケーションソフト開発を支援するための開発支援環境です。SEPを使用したアプリケーション開発の工程に対応して下記の機能を提供します。

① SEPをお客様のターゲット機器に組み込む開発工程

SEPポーティングサポートパッケージ(3参照)を用いて、ターゲット機器のプラットフォーム(H/W、OS、通信対象機器等)に合わせて移植開発を行います。但し、SEP Custom Class Library付きのターゲット機器をご使用の場合は、すでにSEPは組込済みの状態で提供されますので、お客様による本開発工程(移植開発)は必要ありません。

② SEPを使用したアプリケーション開発の前準備工程

アプリケーションを開発する前の準備工程では、ネットワーク設定機能(Network Configurator)、Name編集機能(Name Editor)、User Class編集機能(Class Builder)を使用して、システムを構築するために必要な設定ファイル、Cソースファイルなどを準備(各機能からデータを自動生成)します。

③ SEPを使用したアプリケーション開発、デバッグ、保守工程

工程②にて生成されたデータに基づいてアプリケーションプログラムを開発したのち、生成された実行モジュールをターゲット機器に組み込んでデバッグする工程やシステム稼働後の保守工程では、アプリケーションソフトやSEPシステムが生成するログデータを表示する機能(LogViewer)を使用します。

1-2-1: Network Configurator(ネットワーク設定機能)

SEPを使用したアプリケーションを開発する前に、本機能を使用してシステムで使用するObjectのネットワークパラメータ設定を行います。本設定データはSEPのネットワーク設定ファイルとして生成され、アプリケーションの実行モジュールと一緒にターゲット機器に組み込みます。本設定はSEPを組み込むターゲット機器毎に設定して組み込む必要があります。

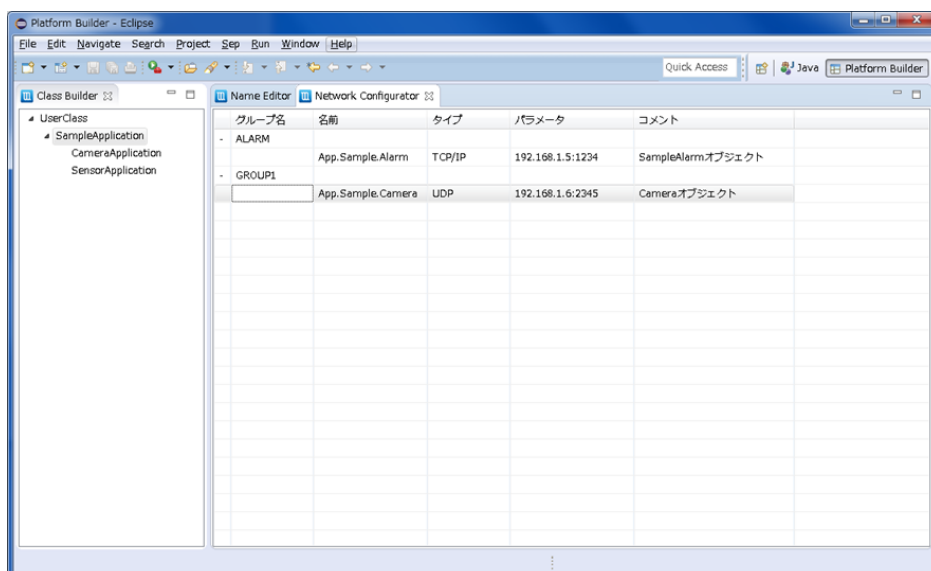


図2-1-1 Network Configuratorの画面例

1-2-2 :Name Editor(Name編集機能)

SEPを使用したアプリケーションを開発する前に、本機能を使用してアプリケーションプログラム内で使用する様々な定数、アクセス対象機器内のデータを指定するための固有の名称(例:PLC内のアドレス名等)に対して、汎用的な名称(Name)と実体(定数値や固有の名称など)を割り付けます。本設定データはCプログラムのインクルードファイルとして自動生成され、アプリケーションプログラムにインクルードして使用します。

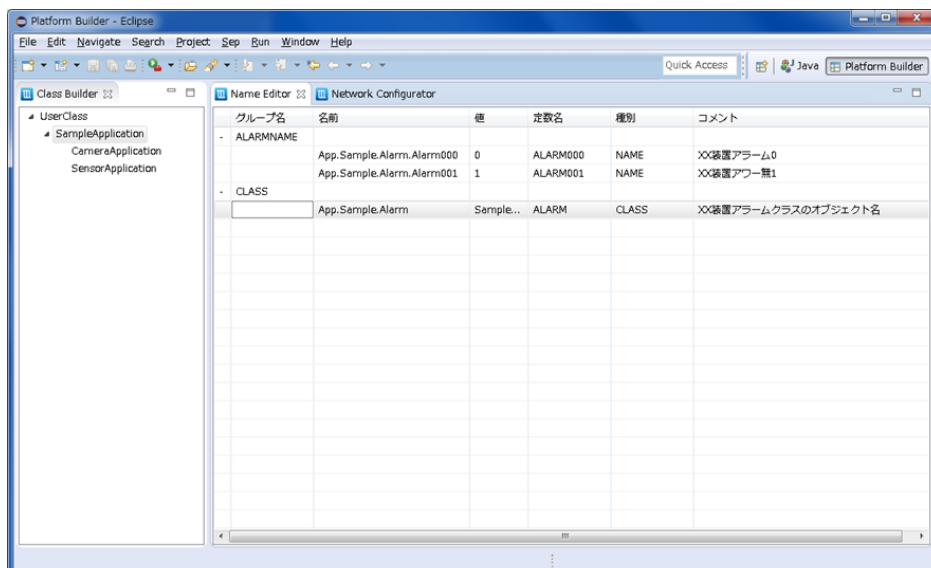


図2-1-2 Name Editorの画面例

1-2-3 :Class Builder(User Class編集機能)

SEPを使用したアプリケーションを開発する前に、本機能を使用してシステムで使用するObjectを定義するためのUser Classの設計を行います。アプリケーションプログラム開発前のUser Classの設計手順は以下の通りです。

① アプリケーションプログラムのアクセス対象となる装置、機器のクラス(User Class)を定義

複数の装置、機器にアクセスする必要がある場合は、その数分のクラス定義を行います。クラスを定義するためには、本機能にてSEPが提供するSEP Basic Classの中から、使用する機器のベースとなるクラス(例えばPLCベースの装置ならばSepPlcクラス)を選択、もしくは機器メーカーが提供する専用クラス(例えば xxxPLC クラス等)を選択して、新しいクラス(例えば MachineABC クラス)を作成します。

② アプリケーションプログラムのクラスを定義

SEPを使用したアプリケーションプログラムでは、SEP Basic Classの中のSepApplication Classから派生してアプリケーションクラスを作成し、SEPアプリケーション自体が一つのオブジェクトとして、他のアプリケーションに対してMethodやメンバ

変数を公開(利用可能と)することができます。(全てのI/Fを非公開としたり、パスワードやライセンスキーなどを用いて認可した特定のユーザに対してのみ公開することもできます。)

③ Cソースコードを自動生成

①②にて定義されたクラスは、本機能によりCソースコード(User Classの定義ファイル)が自動生成されます。

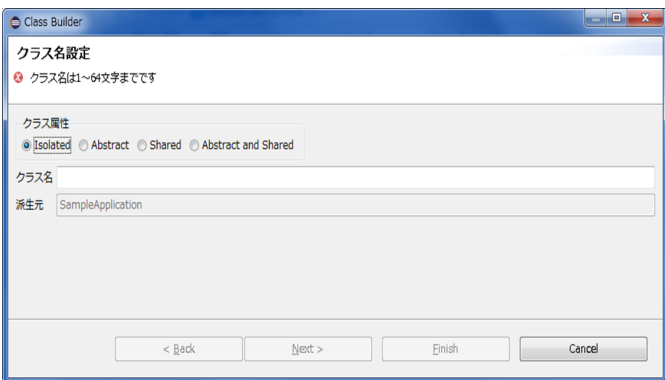
④ 新しく定義したクラスのプログラム作成

③で自動生成された各クラス(例えば MachineABC)の定義ファイルに対して、初期化処理(Constructor)、終了処理(Destructor)、独自に定義したMethod(処理)の実体となるプログラムコードを追加します。

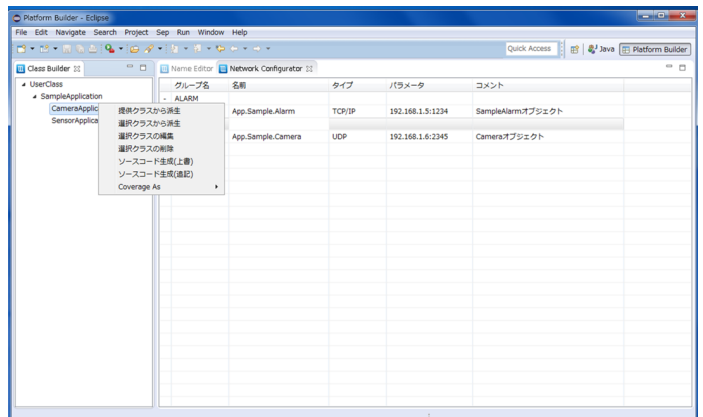
⑤ アプリケーションプログラム開発

アプリケーションプログラム開発では、①から④で作成したクラスに基づくオブジェクトを生成し、生成したオブジェクト(装置、機器)にアクセス(Method実行やメンバ変数値の取得、変更等)するプログラムを開発します。

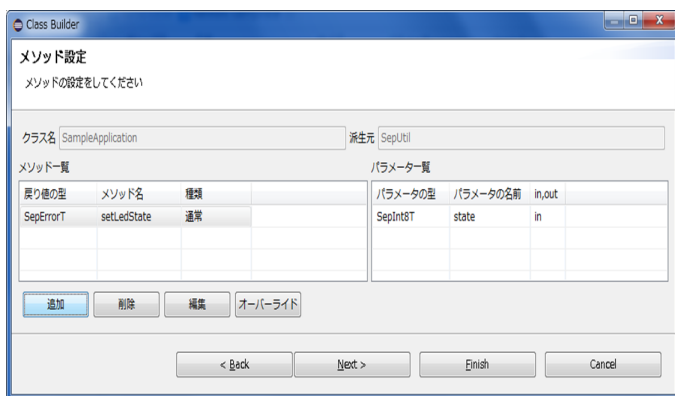
派生元Classを指定して新しいClassを新規作成



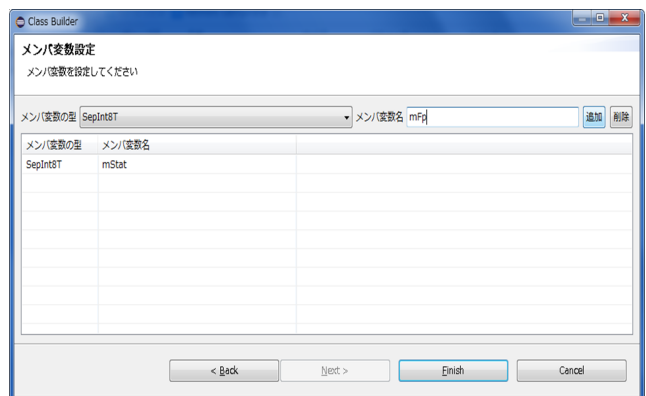
User Classを選択して修正開始



Methodを追加・削除・修正



メンバ変数を追加・削除・修正



Cソースコード自動生成

図2-1-3 Class Builderを用いた設計手順

1-2-4:Log Viewer(ログ表示機能)

本機能は、SEPを使用したアプリケーションをデバッグする時に有効となる機能で、SEP Platform Builderが搭載されたコンピュータ(WindowsPC等)とSEPアプリケーションを組み込んだターゲット機器との間をEthernetで接続した状態で使用します。SEPアプリケーションでは、プログラム内の各所に挿入されたログプログラム(SEP Basic Classに含まれるSepLog Classを使用して記述する)に基づいてデバッグ情報などのログデータがメモリ上に蓄えられています。本機能ではそのログ情報を活用するために以下の2つの機能を提供します。

① リアルタイムログ表示機能

- フィルタ機能付きログ表示
ログ情報にはレベルと種類(エラー、警告、情報等)の付属情報を割り付けることができ、本機能ではそのレベルや種類を指定したフィルタ表示をすることができます。ログ情報の中から必要な情報を絞り込んで表示できます。

1-2-5:Object Caller(デバッグ機能)

本機能は、SEPを使用したアプリケーション間で呼び出されるRemote Agent Callをデバッグするための機能です。

- ネットワーク上のSEP Objectの一覧表示

1-2-6:Key Generator(キーコード生成機能)

本機能は、SEPを使用したアプリケーションを複数台の機器に搭載し、搭載機器間をWAN/Internetを介して相互通信する際のSEP間通信に対する暗号化のためのキーコードを生成する機能です。(暗号化通信が不要な場合は、本機能を使用する必要はありません。また、本キーコードを生成するためには、SEP Network Server PackageとKeyライセンスの購入が必要となります。)

- 複数機器内ログ情報の時間同期表示
ログ情報にはログ時刻情報がついており、SEPアプリケーションが組み込まれた複数の機器からのログ情報をマージした後、ログ時刻順に並べ替えて表示することができます。複数機器上のプログラムを時系列的にトレースしながらデバッグすることができます。
- ログ情報のファイル出力
ターゲット機器から読み出したログ情報をファイルに保存することができます。CSV形式のファイル出力も可能なので、ログ情報を市販の表計算ソフトに読み込むこともできます。

② ログファイル表示機能

①のファイル出力機能により保存したログ情報ファイルに基づき、①と同様のフィルタ機能付きログ表示、複数機器内ログ情報の時間同期表示を行うことができます。それ以外に、下記の機能を提供します。

- キーワード等の条件を指定したログ情報検索
- ログ情報の印刷機能

- 選択したSEP Objectの情報表示
- パラメータを指定してRemote Agent Callを実行
- 実行結果の表示

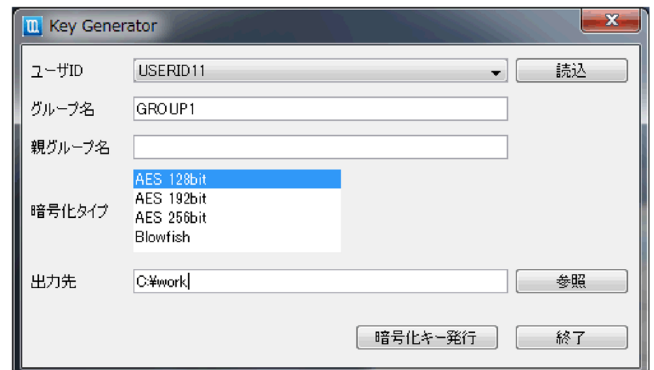


図2-1-4 Key Generatorの画面例

2: SEP ポータリングサポートパッケージ

本パッケージは、SEPアプリケーションを組み込むターゲット機器のプラットフォーム(H/W、OS、通信対象機器等)に合わせて移植開発を行うために必要となります。但し、SEP Custom Class Library付きのターゲット機器をご使用の場合は、すでにSEPは組込済みの状態で提供されますので、お客様が本パッケージを使用して移植開発を行う必要はありません。本パッケージでは以下の内容物が提供されます。

① SEPCoreポータリングマニュアル

- 移植手順概要説明
- 移植前準備事項説明
(移植前に決定しておく必要があるプラットフォーム仕様項目の一覧)
- プラットフォーム依存層のI/F関数API
(本APIに合わせて、ターゲット機器のH/W、OSやアクセス機器の通信仕様に対応したプログラムを実装ください。)

② SEP Basic Class Libraryのソースプログラム

③ プラットフォーム依存層のサンプルプログラム

- CPL (通信依存層) サンプル
Windows PC間でのLAN接続用サンプルプログラムのソースコード一式
- PPL (H/W、OS依存層) サンプル
Windows PC用サンプルプログラムのソースコード一式

注: SEPCoreの中に含まれるVIO-Linksをお客様独自のプラットフォームに合わせて移植するためには、VIO-Linksバイナリ提供サービスをご利用いただく必要があります。詳しくは弊社窓口までお問い合わせください。

■ SEP Platform Builder (SEP組込プラットフォーム構築ツール)

<Windows版>

項目	名称・仕様
CPU	1.0GHz以上のIntel PentiumⅢ相当以上のプロセッサ
OS	Windows7 Professional 32bit/64bit、Windows8.1 Pro 32bit/64bit
言語	日本語・英語
メモリ	Windows7/8.1 32bit 2GB以上、Windows7/8.1 64bit 4GB以上
モニタ	1024x768、32ビットカラー以上を推奨

<Linux版>

項目	名称・仕様
CPU	1.0GHz以上のIntel PentiumⅢ相当以上のプロセッサ
OS	Linux kernel 2.6.3x以上
言語	日本語・英語
メモリ	1GB以上
モニタ	1024x768、32ビットカラー以上を推奨

<MacOS版>

項目	名称・仕様
CPU	1.0GHz以上のIntel PentiumⅢ相当以上のプロセッサ
OS	Mac OS X 10.8
言語	日本語・英語
メモリ	2GB以上
モニタ	1024x768、32ビットカラー以上を推奨

■ SEPCore (SEP組込プラットフォーム)

項目	名称・仕様
CPU	ARMコアのMCU/SOC、ATOM、PowerPCなど組込機器向けCPU、WindowsPCのCPU
OS	SEPCore(Server & Client機能付きの標準版) Linux kernel 2.6.3x以上 Windows7 Professional 32bit/64bit、Windows8.1 Pro 32bit/64bit、 Windows Embedded Compact7(※1) 各種RTOS(μITRON(※2)、TOPPERS、RTX、VxWorks等)(※1) SEP Client (Client機能のみのバージョン) Android4.1以上、iOS7以上
メモリ	最小構成(VIO-Links機能)で30KBytesのROMと8KBytesのワークRAMメモリで動作可能
言語	言語非依存

※1 パッケージに付属するライブラリはWindows7/8、Linux、Androidになります(Ver.1.1)。

付属していないライブラリについては個別にポーティングが必要となります。

※2 実績はμITRON仕様ベースのTOPPERS/ATK2となります。

16ビットマイコンやμITRONなどのコンパクトなMCUやRTOS (Real Time OS)上でも動作可能で、通信のトランスポート層においては、TCP/IP、UDP以外にEthernet、Bluetooth、SIO、CAN、NFCなどの動作実績があります。

■ VIO-Links Lite (SEP Smallcore)

項目	名称・仕様
メモリ	ROM: 30KBytes、RAM: 10KBytes
OS	なし
言語	言語非依存

■ SEPアプリケーション開発フロー

SEPを組み込んだアプリケーションを開発するための、開発フローについて説明します。

① 組込プラットフォーム(SEPcore)をターゲット機器組み込み用に移植

<必要な開発ツール>

SEP ポーティングサポートパッケージ, ターゲット機器用C/C++ IDE

<実施項目>

●プラットフォーム依存部(PAL)の移植

OSのマルチタスク機能(タイマ、セマフォ、タスク関連等)処理をSEPのI/Fに合わせて実装。接続するH/Wとのアクセス処理をSEPのI/Fに合わせて実装。

●通信依存部(CAL)の移植

接続する機器との通信処理をSEPのI/Fに合わせて実装。

② SEPアプリケーションソフト開発の前準備作業

<必要な開発ツール>

SEP Platform Builder

<実施項目>

●ネットワーク設定(Network Configurator)

接続する機器との通信パラメータ(IPアドレス、通信速度等)を設定

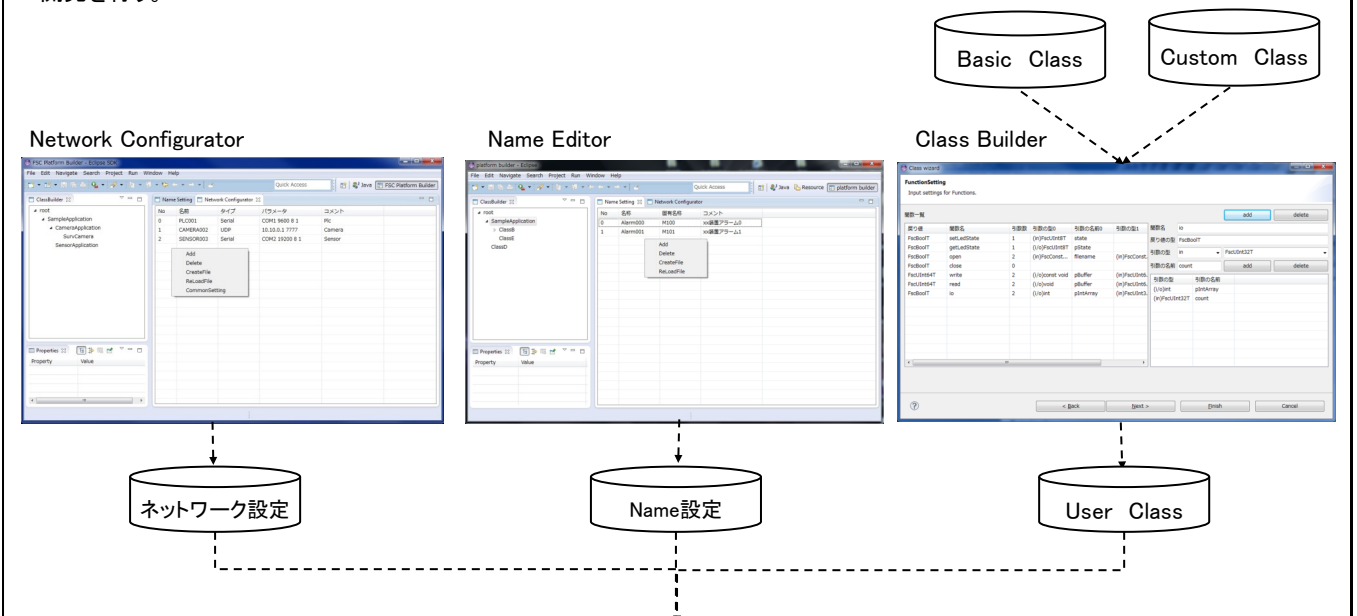
●Name設定(Name Editor)

各機器の名称や機器内部のデータ名称などシステムで使用するNameの設定

●Class設計(Class Builder)

システムの基本機能を提供するクラスライブラリ(SEP Basic Class Library)を用いて、装置やシステムのClass定義を行う。

各ツール設定したデータからはCプログラムコードが自動生成されるので、SEPアプリケーション開発時に当該ソースを組み込んで開発を行う。



③ SEPアプリケーションソフト開発

<必要な開発ツール>

SEP Platform Builder, ターゲット機器用C/C++ IDE

<実施項目>

●SEPアプリケーションソフトの開発

②で準備したネットワーク設定、Name、Classを用いてSEPアプリケーションを開発

●SEPアプリケーションのデバッグ

LogViewer、Object Callerを活用してSEPアプリケーションのデバッグを行う。

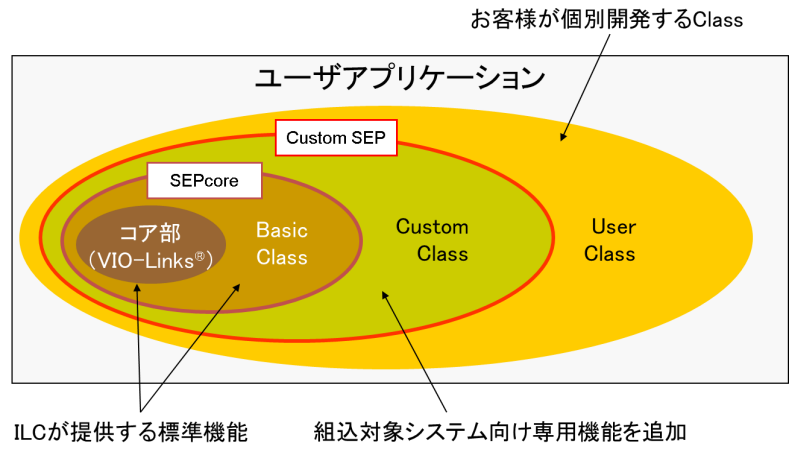
注: 別売の"INTAWORLD"を使用することで、ビジュアルな設計手法を用いてCプログラムコードを開発することなくプログラミングレスにSEPアプリケーションを開発可能。(P18参照)

SEPメーカーカスタマイズ機能

SEPを組み込んだアプリケーションソフトを開発するためには、ユーザは予めSEPをターゲット機器へ移植しておく必要があります(詳しくはP8を参照)が、この移植作業を不要にし、ユーザがターゲット機器を購入後すぐにSEPアプリケーションの開発ができるようにするための仕組みがメーカーカスタマイズ機能です。

組込機器製品のメーカーが、制御対象となるI/Oや各種デバイスアクセス用のDriverとClass Library(これをSEP Custom Class Libraryと呼びます。)を提供することで、ユーザは移植作業無しにSEPアプリケーションを開発できるようになります。

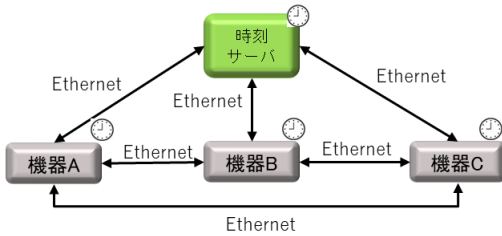
SEP Custom Class Libraryは、SEPの開発ツールであるSEP Platform Builder(Class Builder)とSEP Basic Class Libraryを使用して開発(制御対象毎に、SEP Basic Classを継承したCustom Classを定義し、Custom Classの各Methodの処理プログラムを実装)します。



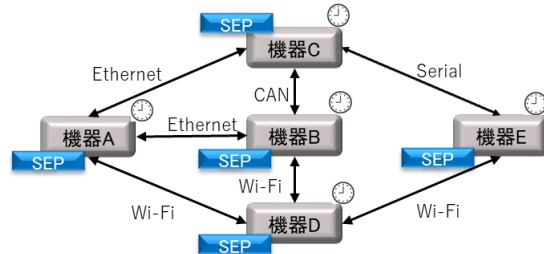
<時刻同期機能>

- ・本プラットフォーム(SEP)が組込まれた機器どうしがつながれたシステムで異なるクロックを持つ各機器の時刻を同一に合わせる機能を提供します。
- ・時刻サーバ不要(機器間の時刻の計算はTSNと同じです。)
- ・機器間の通信方式が異なるシステムにも対応可能です。

TSN = Time-Sensitive Networking



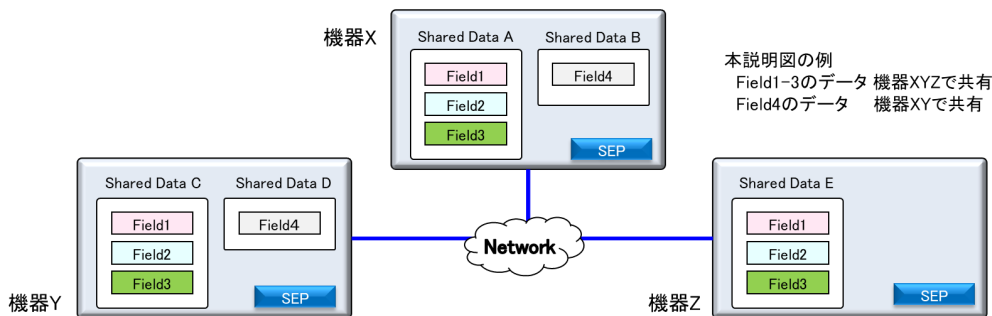
従来の時刻同期システム



調和型時刻同期システム

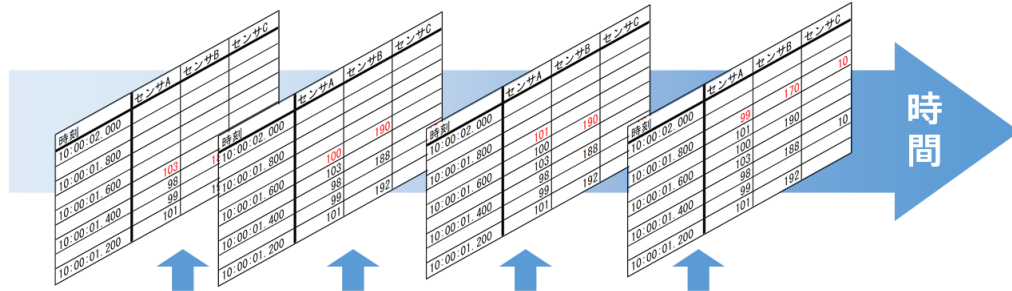
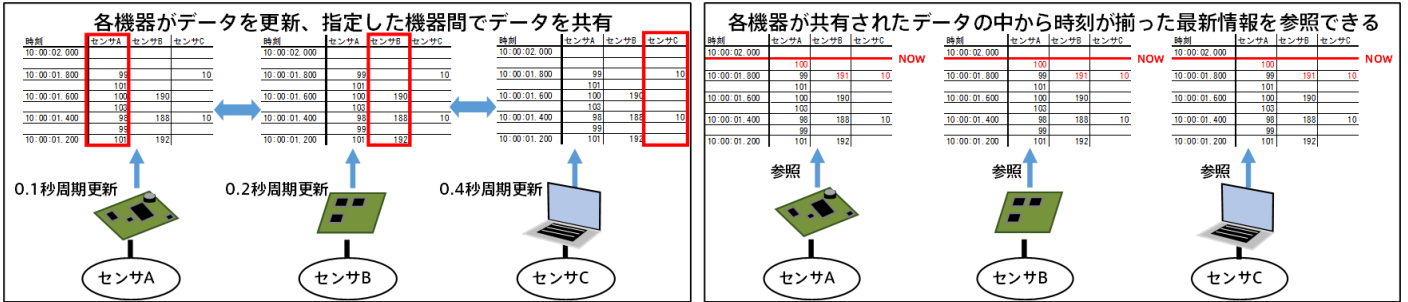
<時刻同期型データ共有機能>

- ・機器間で同期した時刻付きデータ共有が可能です。



本説明図の例
Field1-3のデータ 機器XYZで共有
Field4のデータ 機器XYで共有

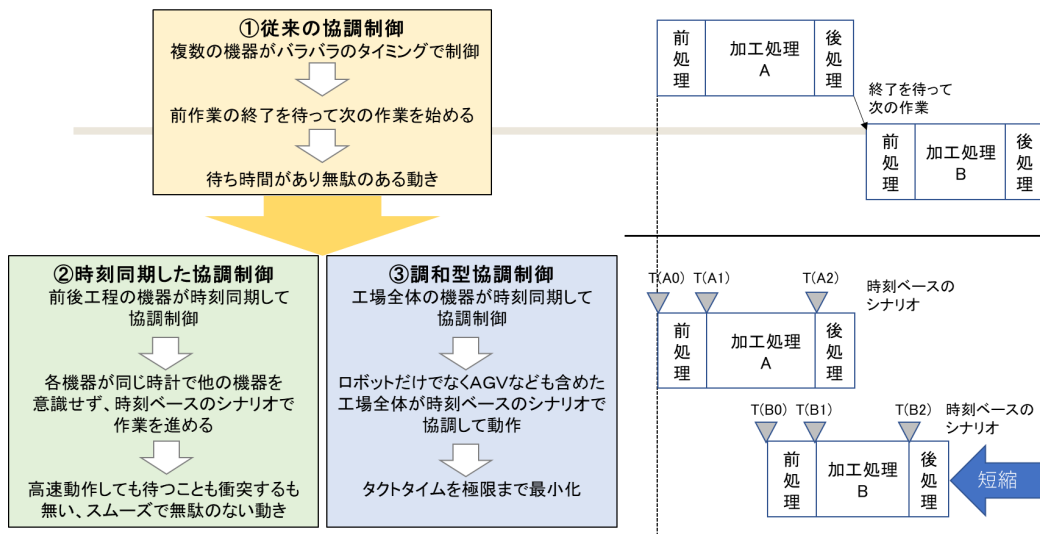
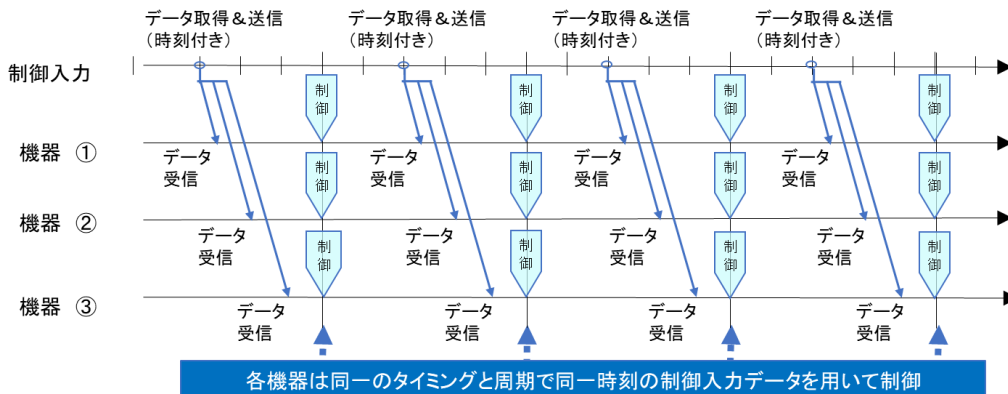
<時刻同期型データ共有機能>



アプリケーションの任意のタイミングでローカルの情報参照するだけで常に時刻が揃った各機器の最新データを参照できる

<調和型時刻同期機能>

- システム全体のタイミングが一致した協調制御が可能です。
(従来の協調制御では実現できなかったこの協調制御のことを「調和型協調制御」と呼びます。)



NO.	単語	定義
1	SEP	Smart Embedded Platformの略称。本商品(製品)の総称を指す。
2	SEPCore	SEPの中の組込ソフトウェアプラットフォームを指す。
3	SEP Platform Builder	SEPCoreのターゲット機器への移植開発およびアプリケーションソフト開発の前準備のための開発支援環境。
4	SEP Basic Class Library	ユーザがアプリケーションソフトを開発するために必要となる基本的な機能を提供するCLASS ライブラリ。
5	SEP User Class	SEP Basic Class で定義されたClassから派生してユーザが作成するClassのこと。
6	VIO-Links [®]	各機器への固有のアクセス処理部とユーザアプリケーションソフトとの間に位置し、ユーザアプリケーションソフトから各機器固有の違いを意識することなくアクセスができるようにするための仕組み。
7	CAL	Communication Abstraction Layerの略。
8	VCI	Vio-links Cal Interfaceの略称。
9	CPL	Cal Porting Layerの略。
10	PAL	Platform Abstraction Layerの略。
11	SPI	SEPCore Pal Interfaceの略
12	PPL	Pal Porting Layerの略。
13	SEP protocol	VIO-Links [®] を組み込んだApplication Object間での通信プロトコル。
14	SEP world	VIO-Links [®] を介してObjectやApplication Objectがつながって構成される世界の総称
15	Network Configurator	アクセス対象の名前(Name)とアクセスに必要なNetworkパラメータ設定を行う機能。
16	Name Editor	アプリケーションソフトで指定する変数名(Name)と接続対象機器での固有名称の割り付けなど、アプリケーションプログラムで使用する様々なNameの定義を作成する機能。SEP Platform Builderに含まれる。
17	Class Builder	SEPが提供するSEP Basic Classから派生させてユーザ独自のクラスを編集するEDITOR機能。
18	Object (オブジェクト)	オブジェクト指向プログラミングが持つ特徴の一つ。S/Wが扱おうとしている現実世界に存在する物理的あるいは抽象的な実体を属性(データ)と操作(メソッド)の集合としてモデル化し、コンピュータ上に再現したもの。
19	SEP Application Object	VIO-Links [®] の仕組みによってアプリケーションソフトが外部からのアクセスが可能なI/Fを公開し、アプリケーションソフト自体を一つのObjectとして振る舞うことができるようになる。このようなアプリケーションソフトのことをSEP Application Objectと呼ぶ。
20	カプセル化	オブジェクト指向を構成する概念の一つ。オブジェクト内部のデータを隠蔽したり、オブジェクトの振る舞いを隠蔽したり、オブジェクトの実際の型を隠蔽したりすることをいう。
21	抽象化	異なる種類をひとまとめにする上位概念をつくること
22	シームレス	アプリケーションソフトの中で、一貫して操作(例えばデータ取得)することができる状態のこと。
23	プラットフォーム	コンピュータやシステムの基礎部分となるもの。通常、ハードウェアおよびOSを指す。
24	マルチプラットフォーム	複数のOSやハードウェアに対応していること。
25	Class (クラス)	オブジェクト指向プログラミングにおいて、データとその操作手順であるメソッドをまとめたオブジェクトの雛型を定義したもの。
26	トンネリング	本来通信を行いたいプロトコルが、ファイアウォールなどの制限で使用できない場合に、本来のパケットを、別のプロトコルのパケットで包んで通信を行うこと。
27	通信の暗号化	通信パケットの伝文内容を送信元と送信先のみが知るKeyで暗号化し、伝送途中で伝文をのぞかれたり、改ざんされたりすることなく、安全に送受信を行うこと。
28	Remote Agent Call [®]	VIO-Links [®] を使用して、ネットワークやメモリ、シリアル通信など離れているObjectの機能を利用する仕組みのこと。
29	セキュアな通信	通信プロトコルを暗号化してセキュリティを高め、秘匿性を保証した通信のこと。SEPでは、装置どうしが一対一で、チャレンジレスポンスやAESなどセキュアな通信を提供している。
30	VIO-Links Lite	リソースが限られた環境で動作するVIO-Linksの省メモリ版。Small SEPCoreとも呼ぶ。

当社ホームページ

<http://www.ilc.co.jp/commodity/sep/>



株式会社アイ・エル・シー

本社	〒100-0005 東京都千代田区丸の内3丁目4-1 新国際ビル4F TEL: 03-3287-7700 FAX: 03-3287-3999
デザインセンター	〒732-0824 広島市南区的場町1丁目3番6号 広島的場ビル9F TEL: 082-262-7700 FAX: 082-263-4411
名古屋オフィス	〒453-0801 名古屋市中村区太閤3丁目1番18号 名古屋KSビル6F TEL: 052-452-7700 FAX: 052-453-4400

技術的なお問い合わせ

インタコネクション テクニカルセンター

電話受付時間：月曜日～金曜日（祝日を除く）9:00～17:00

TEL：082-262-7799 / FAX：082-263-4411

Eメール：fa@ilc.co.jp

ホームページ：http://www.ilc.co.jp/

★このカタログの記載内容は、予告なく変更することがあります。

★このカタログに記載されている商品は日本国内仕様です。海外でご利用になりたい場合には、別途お問い合わせください。

★INTAconnection、FACTICS、Smart Embedded Platform、VIO-Links、Remote Agent Callは株式会社アイ・エル・シーの登録商標です。

★このカタログに記載されている他社製品名(ソフトウェア・ハードウェア)は、各社の商標または登録商標です。



安全に関するご注意

製品のご使用に際しては、製品に付属しているマニュアルおよび関連マニュアルを良くお読みいただくとともに、安全に対して十分に注意を払って、正しい取り扱いをしていただくようお願いいたします。