

INDUSTRIAL  
EDGE  
SOLUTION  
WITH  
HARD REALTIME  
CAPABILITIES

**RT-edge**

Micronet.Co,

**マイクロネット**  
**Micronet**

INDUSTRIAL REALTIME EDGE COMPUTERS

# I-I/O Container

RT-edge Software I-I/O Service Container  
ユーザースマニュアル






株式会社マイクロネット

<http://www.mnc.co.jp>

TEL: +81(0)299-90-1733

FAX: +81(0)299-90-8557

本書で使用するマークについて

	ノート: 操作方法や手順等の補足情報や注釈を説明しています。
	情報: 製品を利用する上で有効な豆知識となる説明をしています。
	警告: 製品仕様上注意が必要な事象について説明しています。

Windows、Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

INtime は、米国 TenAsys Corporation の登録商標です。

TenAsys®, INtime®, eVM® and iRMX® are registered trademarks in USA of the TenAsys Corporation.

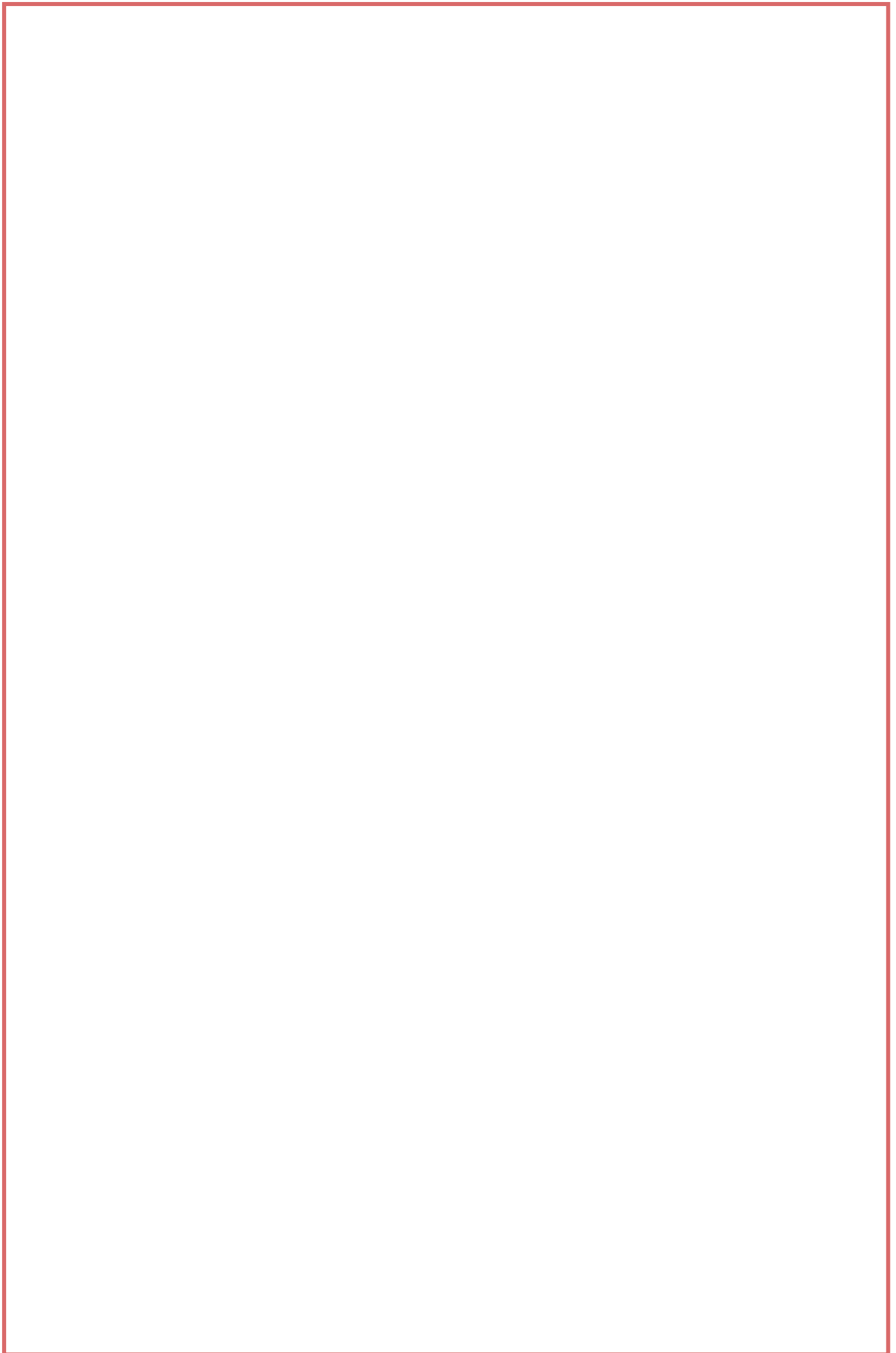
その他、本書に記載されている会社名、商品名は、各社の商標または登録商標です。

本書の内容を無断で転載することは禁止されています。

本書の内容に関しては、予告なしに変更することがあります。あらかじめご了承ください。

## 目次

用語解説	4
関連資料	6
1. 概要	7
1.1. RT-edge とサービスコンテナ .....	7
1.2. I-I/O コンテナ .....	9
2. 仕様	11
2.1. 動作環境 .....	11
2.2. 使用可能タグ数 .....	11
2.3. 対応プロトコルと使用ドライバ .....	11
3. コンテナ導入フロー	12
4. インストール	13
4.1. ファイル .....	13
4.2. ファイルのインストール .....	13
4.3. 起動設定 .....	14
4.4. 動作確認 .....	15
4.5. 終了設定 .....	16
5. 環境設定	17
5.1. Edge コントローラ設定 .....	17
5.2. 接続機器設定 .....	17
6. 設定概要	18
6.1. ECI 設定 .....	18
7. I-I/O サービスコンテナ設定	21
7.1. RTCD 設定概要 .....	21
7.2. ECI(コンテナ設定情報ファイル)の編集 .....	22
7.3. コンテナプロパティ設定 .....	25
8. 設定サンプル	26
9. 動作確認	27
9.1. 動作確認手順 .....	27
9.2. サービスインジケータタグ .....	28
9.3. RT-edge タグデータの妥当性について .....	28
10. 付録	29
10.1. サービスプロパティタグ .....	29
10.2. サービスインジケータタグ .....	30
10.3. サービスメッセージ .....	30
10.4. ECI Address 定義 .....	31
10.5. 環境設定 その他仕様制限事項 .....	33
10.6. トラブルシューティング .....	34



## 用語解説

本ドキュメントにおいて使用される用語・略称について説明します:

表 1. 用語集

用語	説明
RT-edge	エッジコンピューティングを軸とする IT の情報処理と、FA における装置・機器の制御を融合し、密度の高い高頻度データ利用を可能とするソフトウェアプラットフォームです。 FA で要求されるハードリアルタイム制御を組み込むことで、情報処理と機器・装置制御を可能とするエッジコントローラを構成することができます。
RT-edge 基本ソフトウェア	RT-edge 機能の核となる機能・ライブラリを実装するパッケージソフトウェア製品です。
IoT ゲートウェイ	IoT において、端末とインターネットを介した遠隔サーバー(クラウド)がデータのやりとりをする際、中継する役割を担う機能。サーバーや送信経路であるインターネット負荷の軽減をします。
IT システム	オンプレミスもしくはクラウドを活用した業務システムやアプリケーション。
Intime	INtime for Windows: Windows と協調動作可能なリアルタイムカーネル拡張ソフトウェアです(RTOS ソフトウェア)。 INtime Distributed RTOS(dRTOS): Windows OS を必要とせず、スタンドアロンで動作するリアルタイム OS です。
RTA	RealTime Application: リアルタイムアプリケーションの略称。INtime 上で動作するロードダブルプロセスの拡張子です。INtime 上で動作するロードダブルアプリケーションは、RTA という拡張子を持ちます。
RSL	Realtime Shared Library: リアルタイム共有ライブラリの略称。INtime 上でアプリケーションがロード可能なライブラリです。Windows 上で使用される DLL(Dynamic Link Library)のようなものです。RTA から使用されるライブラリインタフェース等は、こちらを使用して作成することができます。
API	Application Programming Interface: アプリケーションプログラミングインタフェースの略称。RT-edge ではデバイスへのアクセスインタフェースとして API ライブラリを提供しています。
エッジアプリケーション	RT-edge 内コンテナにより集積されたデータ(RTCD)を活用、処理実行するソフトウェアです。
エッジコンピューティング	RT-edge 内で稼働する制御コンテナソフトウェアにより装置・機器から収集した高密度なデータをリアルタイムに収集、分析、フィードバックします。IT システムとの情報連携。
オンプレミス	サーバーやソフトウェア等の情報システム、アプリケーション等のソフトウェアを管理する施設内に設置して運用すること。
クラウド	サーバーやストレージ等のインフラやソフトウェアを必要とせず、必要な IT リソースが、インターネットを通じてオンデマンドで得られる形態、サービス。
産業用 PC	高信頼性、耐環境性、長期供給等の特徴をもつ産業用途の PC。
データ収集	診断、分析を行う対象となるデータを集積する処理。
データ加工	集積されたデータを利用しやすい形に変更する処理。
産業機器通信インターフェース	各種フィールドバス経由で機器、装置との通信、もしくは直接入出力デバイスの制御を行うインターフェースです。本インターフェースを介し、センサー値の参照やアクチュエータ制御が可能です。
サービス/EgService	RT-edge システムを構成する機能プロセス(rta/exe)です。
タグ/EgTag	瞬時値データ値 1 つを示すオブジェクトです。ユニーク名とグローバルなスコープを持ち、全ての EgService から読み書きが許されたオブジェクトです。タグは生成時にデータ型が確定され変更はできません。
リンクタグ	同一名称のタグを重複生成した場合に自動的に別名称で生成されるタグを指します。 通常のタグと同様、グローバルなスコープを持ち、全ての EgService から読み書きが許されたオブジェクトです。一つのタグに対し、異なるプロパティ情報を定義したい場合に使用します。
セグメントタグ/Segment	タグの型の一つで、ECI 指定のサイズでメモリブロックを作成し、このメモリブロックに対

用語	説明
	して、値の入出力を行うことができます。構造体など大きいサイズのデータを扱う場合使用します。
文字列タグ/String	タグの型の一つで、ECI 指定のサイズで文字列を格納するメモリブロックを作成し、文字列を格納、または取り出すことができます。
データセット/EgDataset	タグ 1 つ以上の組み合わせでデータ並び順(データ構造)を定義する名前付きオブジェクトです。
コレクタ/EgCollector	データセットに定義されたデータ構造に従って、同時刻のバイナリデータ列で生成し、データレコードとしてメールボックスに送信するオブジェクト (スレッド) です。
メールボックス/EgMailBox	時系列なデータセット、または時系列メッセージを FIFO で蓄えることができ、また受信イベントとして処理できるオブジェクトです。
タグ参照/TagRef	タグの参照として使用するオブジェクトです。タグの名前を保持し値は保持しません。サービスコンフィグファイルでデータセットの収集用タグとして定義することや、サービス内のオブジェクトとして定義することでサービスのメンバ変数として使用することができます。
コレクタ参照/CollectorRef	コレクタの参照として使用するオブジェクトです。コレクタの名前を保持しそれ以外のオブジェクトは保持しません。サービスコンフィグファイルでサービス内のオブジェクトとして定義することでサービス内のメンバ変数として使用することができます。
メッセージ	メールボックスで扱われる 1 レコード分のデータ、またはサービス間のコマンド、応答の電文です。
フレームワーク	フレームワークは、アプリケーションが API を組み合わせて実装するよくある処理についてマクロ化、自動化したものでサービスコンフィグファイルの記述により自動処理させることができます。
RT-edge コンテナ設定情報 (ECI)	RT-edge コンテナが RTCD に展開する入出力データ定義の他、RT-edge コンテナフレームワークが、オブジェクト生成やコンテナサービス等自動処理するための定義設定情報(XML 型式)。
入力	RT-edge システムを中心に見た場合、外部の情報を RT-edge システムへ取り込む方向性のデータの流れを意味します。
出力	RT-edge システムを中心に見た場合、RT-edge システムが持つデータを外部に書き出す方向性のデータの流れを意味します。
RTCD	Realtime Common Data の略称。RT-edge システム上で最もベースとなる共有データ構造機能です。RT-edge サービスコンテナの生成する入出力データタグのコレクション。 例えば、センサーや装置から収集したデータをアプリケーション間で受け渡しを行う場合、またはアプリケーション間でメッセージのやり取りを行う場合等、アプリケーション間でデータの受け渡しを行うケースで利用されます。 RTCD には入出力データの瞬時値が格納されるだけでなく、入出力データを処理した加工データや、アプリケーションや各サービスコンテナが独自に提供したデータも格納されます。
通信プロトコル	異なるデバイスやコンピュータシステム、ソフトウェアなどが互いに通信するために制定された規格です。I-I/O コンテナでは装置、デバイス、センサーなどの機器との通信で使用する通信プロトコルを利用します。
Edge I/O ポート	通信対象の入出力データエリアを、連続した一つのアドレス空間として定義した RT-edge 独自のアドレス表現です。
EtherCAT	「EtherCAT®」(Ethernet for Control Automation Technology)は、Ethernet と互換性のあるオープンなフィールドネットワーク技術で、効率の高いデータ転送により、高速かつ高精度の制御が可能な、マシンコントロールに最適化された高速なフィールドネットワークです。
RSI-ECAT-Master	INtime 向け EtherCAT Master Stack です。 (本書では RSI-ECAT と省略する場合もあります。)
NIC	ネットワークインターフェースカード / ネットワークインターフェースコントローラです。
ENI	EtherCAT Network Information ファイルです。スレーブ機器の構成情報が記載されたファイルです。
入力ステータス	入力ステータスは、ON/OFF 出力である DI(Discrete Input)やスレーブデバイスの状態入力として用いられます。
入力レジスタ	入力レジスタは、AI(Analog Input)やスレーブ内の情報として用いられます。16 ビット長のデータで参照のみで変更はできません。

用語	説明
保持レジスタ	保持レジスタは、AO(Analog Output)やスレーブ内の設定情報として用いられます。16 ビット長のデータで参照・変更が可能です。
CAN	Controller Area Network: 特に車載用途で使用するシリアルバス規格です。耐ノイズ性が高い伝送方式や、静的に決定可能なフレーム優先度制御(ID 部を使用)などを持ち、堅牢な制御ネットワークを構築することができます。最大速度は 1Mbps 程度です。
EtherCAT-CAN ゲートウェイ	EtherCAT スレーブおよび CAN ノードとして機能するデバイスで、EtherCAT と CAN の双方に接続し、CAN フレームの送受信を EtherCAT 経由で行うことができます。
RSI-CAN-E	EtherCAT-CAN ゲートウェイの制御を行う CAN 通信ドライバです。本ドライバは INtime 上で動作します。

## 関連資料

### RT-edge 製品に含まれる資料

表 2 .RT-edge 関連資料

名称	ファイル名	内容
RT-edge ユーザーズマニュアル	DOCRTEGEUSER.pdf	RT-edge システムの全般的な説明が記載されています。
RT-edge API リファレンス	DOCRTEGEAPI.pdf	RT-edge API の使用方法が記載されています。
RT-edge コンテナ作成マニュアル	DOCRTEGDSRV.pdf	RT-edge コンテナの構造、サンプルプロジェクトを利用した作成方法等について記載されています。

表 3 .ドライバ製品関連資料

名称	ファイル名	内容
RSI-ECAT ユーザーズマニュアル	usersmanual.pdf	RSI-ECAT (INtime 用 EtherCAT Master ドライバ) の全般的な説明が記載されています。
RSI-CAN-E セットアップマニュアル	DOCRSICANEDRV.pdf	RSI-CAN-E (INtime 用 CAN 通信ドライバ) を使用するための説明が記載されています。

# 1. 概要

## 1.1. RT-edge とサービスコンテナ

RT-edge とは、エッジコンピューティングを軸とする IT の情報処理と、FA における装置・機器の制御を融合し、密度の高い高頻度データ利用を可能とするソフトウェアプラットフォームです。

RT-edge の利用により、装置やセンサーからの高密度なデータ収集、分析だけでなく、提供される開発ライブラリキットを使用し、タグデータをレジスタとした機器制御を行うハードリアルタイムエッジアプリケーションの開発が可能です。

### サービスコンテナ

RT-edge の処理ターゲットは、エッジコンピューティングを軸とした IT 情報処理(IT-Process)と、ミリ秒精度のハードリアルタイム性を要求される FA 制御(FA-Control)に分類され、ターゲットの機能に特化した専門処理サービスをコンテナ(サービスコンテナ)と呼びます。

IT 情報処理ターゲットは上位層にあり、主に外部システムからの要求指示の受付や、外部システムへのデータ公開、通信等を担う要素となります。IT 情報処理サービスコンテナは、制御システムのコンソール画面や外部システムから WEB ブラウザ経由でのアクセス機能、制御データ情報を外部クラウドストレージに保存する機能等、上位システムとの接続・インターフェースを提供します。

一方、FA 制御ターゲットは下位層に位置し、主に通信やハードウェアへの直接 I/O 入出力等により装置・機器制御を担う要素です。FA 制御サービスコンテナは、産業用フィールドバスやコントローラ通信プロトコルによるロボット制御、計測機器からのデータロギング、デジタルパルス出力等、装置・機器へのアクセスを提供します。

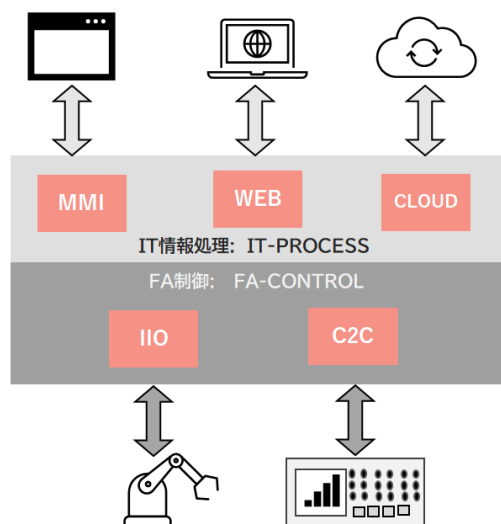


図 1. ターゲットとコンテナ

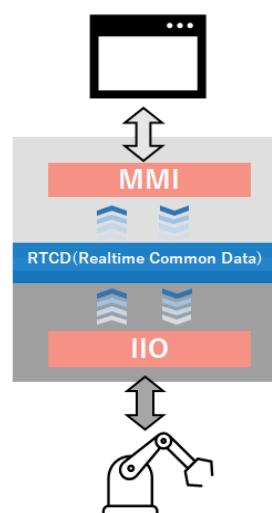


図 2. コンテナの役割





#### サービスコンテナ例

WEB: IT 情報処理ターゲット内には、IIS(Internet Information Service)を介し、制御情報をインターネット上に公開する WEB サービスコンテナ

IIO: ロボットアーム制御に特化した産業 I/O サービスコンテナ

サービスコンテナはターゲットに特化した入出力データをシステム内でグローバルにアクセス可能なタグ情報としてリンクし、このタグ情報のコレクションを RTCD(Realtime Common Data)とよばれる共有データ構造に展開します。

サービスコンテナは、タグ情報コレクションや、動作・挙動を決定するパラメータ設定と、ターゲット処理に特化した一つ以上の実行処理の集合体です:

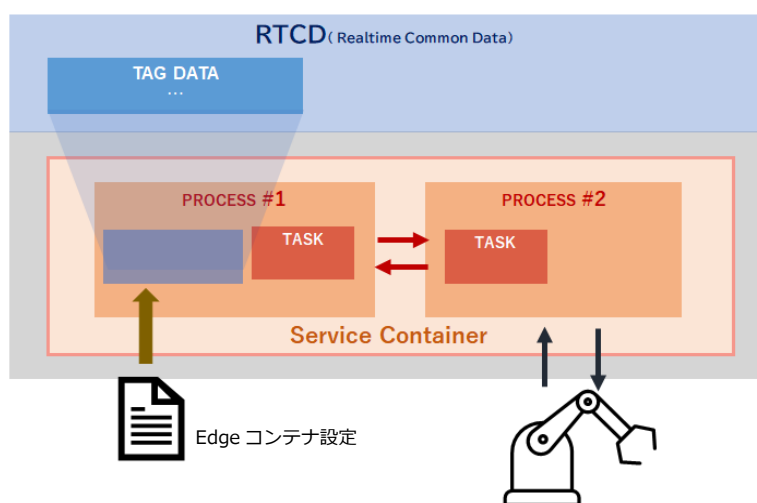


図 3. サービスコンテナの構造

#### サービスコンテナ例

##### PROCESS #1

RTCD へのアクセシビリティを提供する処理プロセス(サービスハンドラ)

##### PROCESS #2

ターゲット特化処理プロセス

##### Edge コンテナ設定情報

タグデータコレクション等コンテナ設定

## 1.2. I-I/O コンテナ

I-I/O コンテナ(Industrial Input/Output Interface コンテナ)は、ターゲットとなる装置、デバイス、センサーなどの機器に対して、各種フィールドバス通信で制御を行うサーバー(マスター)機能です。

RT-edge (I-I/O コンテナ)を導入することで、装置、デバイス、センサーなどの機器にアクセス、タグを介した入出力が可能になります。

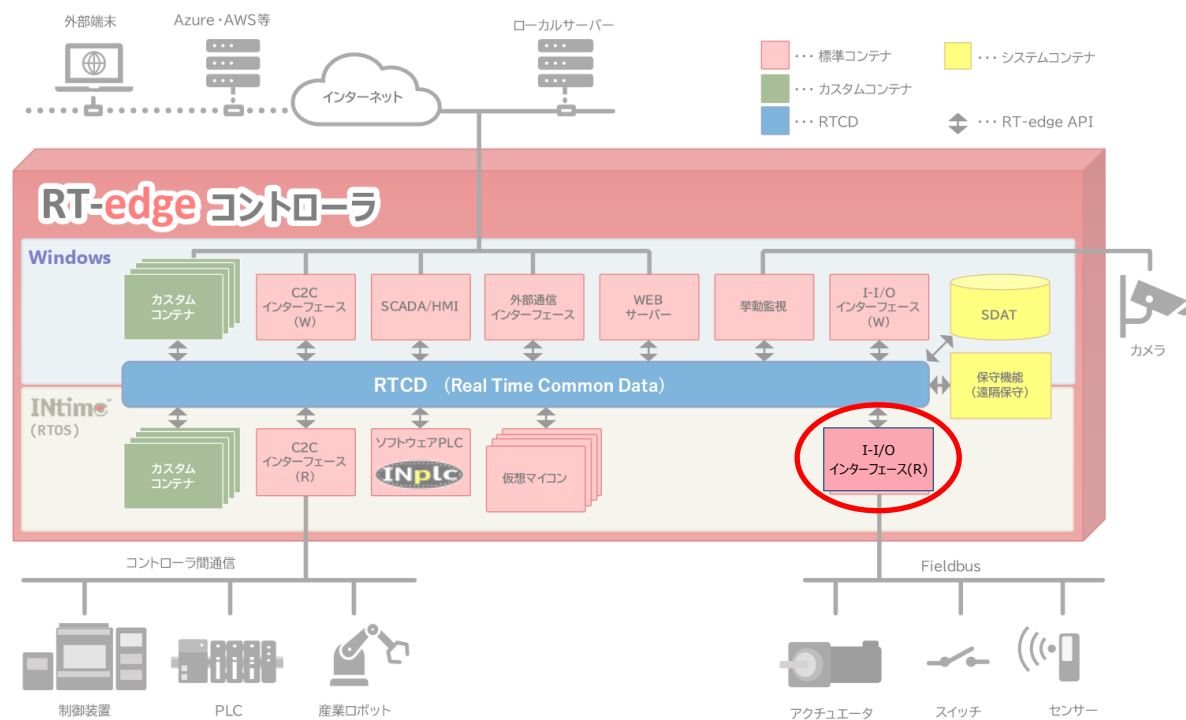


図 4.RT-edge 相關図

## 構成要素

I-I/O コンテナは以下コンポーネントから構成されます：

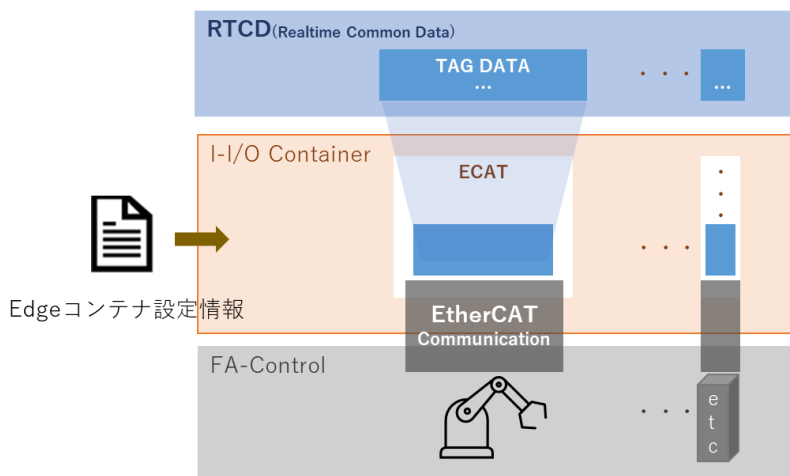


図 5. I-I/O コンテナ構成

表 4. I-I/O コンテナ構成要素

コンポーネント	内容
EgIIO.rta	RT-edge I-I/O コンテナに含まれるソフトウェアコンポーネントです。 ECI(RT-edge コンテナ設定情報:XML 形式ファイル)を読み込み、各通信プロトコルを利用し、センサーや測定器といった状況監視機器、サーボモータといった制御機器などの I/O メモリ領域へのアクセスビリティを RTCD に展開します。
eWrapECAT.rsl	RT-edge I-I/O コンテナに含まれるソフトウェアコンポーネントです。 EtherCAT 通信ドライバと RTCD とを繋ぐライブラリです。EtherCAT 通信プロトコルを用いる場合に使用します。
eWrapCAN.rsl	RSI-CAN-E(別売品)に含まれるソフトウェアコンポーネントです。 EtherCAT 接続の CAN ゲートウェイを使用し、CAN 通信を行うライブラリです。 EtherCAT 通信ライブラリ(eWrapECAT.rsl)と併用して使用します。
RT-edge コンテナ設定情報 (ECI)	センサーや測定器といった状況監視機器、サーボモータといった制御機器などの各種入出力領域、RT-edge 上の RTCD 空間のマッピング情報や各ソフトウェアの動作パラメータを含む設定情報です。



本ドキュメントでは、主に I-I/O コンテナの利用方法について説明します。RT-edge 基本ソフトウェア、他サービスコンテナ、および PLC については各々のマニュアルを参照ください。

## 2. 仕様

### 2.1. 動作環境

- RT-edge 基本ソフトウェアバージョン 3.8.1 以降
- Windows OS x64
- .NET Framework 4.6

### 2.2. 使用可能タグ数

使用可能なタグ数は、RT-edge 基本ソフトウェア 仕様に依存します。

### 2.3. 対応プロトコルと使用ドライバ

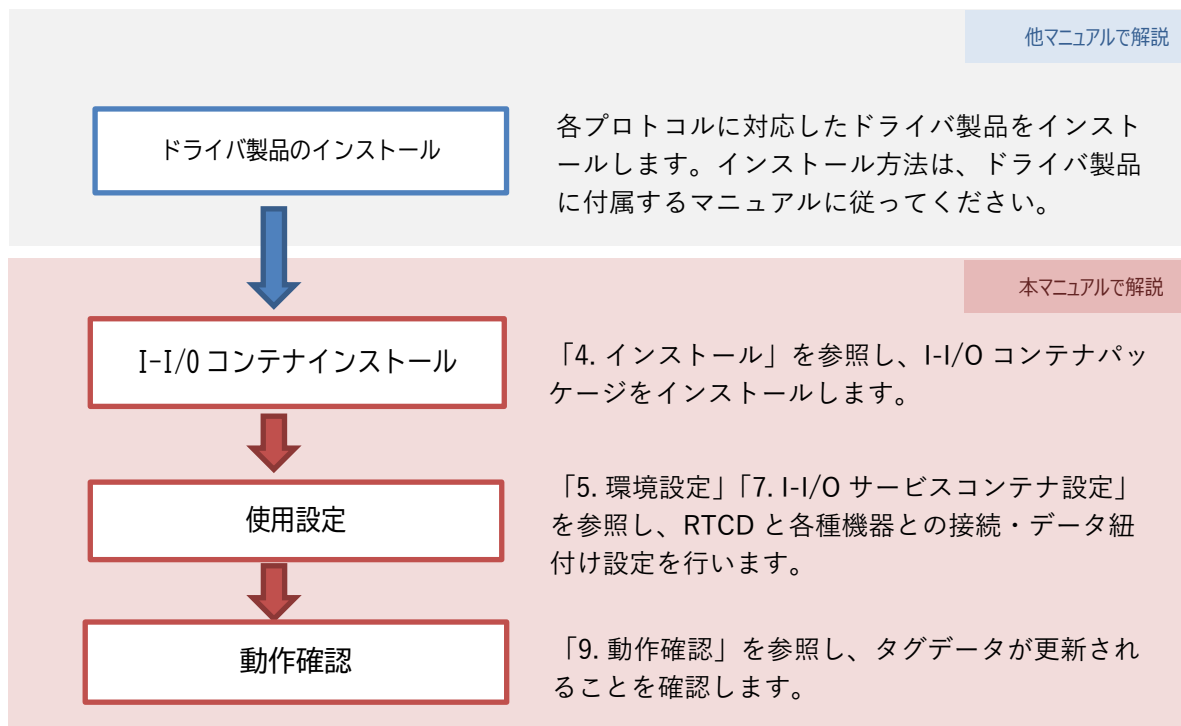
I-I/O コンテナでは以下のプロトコルをサポートしています。また使用するドライバ製品とバージョン、対応する通信プロトコル名を記載します。

表 5. 対応プロトコル一覧

No	通信プロトコル名		使用ドライバ	ドライババージョン
1	ECAT	(EtherCAT)	micronet 社製 RSI-ECAT Driver	4.1.0 以上
2	ECAT	(EtherCAT + CAN 通信)	micronet 社製 RSI-ECAT Driver micronet 社製 RSI-CAN-E	4.1.0 以上 1.0.0 以上

通信プロトコルごとの仕様・制限事項は 10.5. 環境設定 その他仕様制限事項 を参照ください。

## 3. コンテナ導入フロー



## 4. インストール

予め RT-edge を「C:\RTedge」にインストールした状態を前提としています。

### 4.1. ファイル

I-I/O コンテナパッケージには以下のファイルコンポーネントが含まれています:

表 6.I-I/O コンテナパッケージコンポーネント (全コンポーネント)

配置先 フォルダ階層	ファイル名	説明
RTedge\bin	EgIIIO.rta	各通信プロトコルを用いて、制御機器の I/O メモリ領域に対するアクセシビリティを RTCD に展開するサービスプロセスです。
RTedge\bin	eWrapECAT.rs1	EtherCAT 通信ドライバと RTCD とを繋ぐライブラリです。EtherCAT 通信プロトコルを用いる場合に使用します。
RTedge\bin	EgIIIO.xml	I-I/O コンテナ用 RT-edge 設定情報です。本設定ファイルの構成により I-I/O コンテナが RTCD に生成するタグの設定を行います。
RTedge\doc	DOCRTEGEGSRC_IIIO.pdf	本ドキュメントです。



各通信プロトコルに対応したライブラリは、必要に応じてインストールを行います。

### 4.2. ファイルのインストール

I-I/O コンテナ用のモジュールインストール方法については、別紙インストール手順書\_IIIO を参照ください。

## 4.3. 起動設定

RT-edge におけるサービスコンテナ、および関連サービス・アプリケーションの設定は、RT-edge ブートストラッパー設定により行います。I-I/O コンテナの起動設定も同様、RT-edge ブートストラッパー設定に準拠します:

- 1) C:\RT-edge\bin\EgBoot.xml をテキストエディタで開きます。
- 2) RTedge エlement内の Services Element内に、I-I/O サービスコンテナ用のElement (Service Element)を追加します。サービス名には使用する通信機能に対応する通信プロトコル名を指定します。また、Argument 属性には通信機能毎に用意されているライブラリの指定を行います。

Name="通信プロトコル名"

Argument="-RSL RSL 名"

通信プロトコル名、及び RSL 名には以下の名前を入力します。

使用する通信機能	通信プロトコル名	RSL 名
EtherCAT	ECAT	eWrapECAT.rsl
EtherCAT + CAN	ECAT	eWrapCAN.rsl

以下では、EtherCAT 通信プロトコルを用いた I-I/O サービスコンテナの起動設定の内容です。

```
<?xml version="1.0" encoding="utf-8"?>
<RTedge xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  :
  <Services>
    <Service Name="EgBoot" Argument="RTCD=NodeA;TagMaxNum=10000" >
    </Service>
    <Service Name="EgLog" Path="EgLog.exe" Argument="DispNumMax=500" >
    </Service>
    <Service Name="EgTime" Path="EgTime.exe" >
    </Service>
    <Service Name="ECAT" Path="EgIIO.rta" ESI="EgIIO.xml" Argument="-RSL
eWrapECAT.rsl" >
    </Service>
  </Services>
</RTedge>
```

図 6.EgIIO 起動登録

## 4.4. 動作確認

RT-edge ソフトウェアを起動し、I-I/O サービスコンテナが正常に起動することを確認します。ここでは「図 6.EgIIO 起動登録」で記載の EtherCAT 通信プロトコルを使用した例を用いて説明します:

- 1) RT-Edge ソフトウェア(C:\RT-edge\EgBoot.exe)を開始します。
- 2) RT-Edge オブジェクトブラウザ(C:\RT-edge\EgBrow.exe)を起動します。



RT-edge 起動直後に RT-edge オブジェクトブラウザを起動すると初期化中の為、想定されるタグが表示されない場合があります。一度 RT-edge オブジェクトブラウザを終了し、再度起動させてください。

- 3) サービスインジケータタグから、正常状態であることを確認します(サービスインジケータタグについては「9.2. サービスインジケータタグ」を参照ください)。

### 正常状態

- SERVICE.ECAT.Status が 01(1) であること
- SERVICE.ECAT.Run が true(1) であること
- SERVICE.ECAT.Error が False(0) であること
- SERVICE.ECAT.Live が 増加していくこと

RT-edge Object Browser

ファイル			
Tags Containers Collectors Datasets Mailboxes			
Name	Current Value	Type	Source
SERVICE.FLNET.Status	01 (1)	byte	
SERVICE.FLNET.Run	True (1)	bool	
SERVICE.FLNET.Live	0000b736 (46902)	UInt32	
SERVICE.FLNET.Error	False (0)	bool	

図 7. RT-edge オブジェクトブラウザ起動時の様子

上記の状態になっていない場合には、以下のトラブルシューティングをご参照ください。

- 「サービスインジケータの.Error が true になっています。」
- 「サービスインジケータの.Run が false になっています。」



## 4.5. 終了設定

RT-edge におけるサービスコンテナ、および関連サービス・アプリケーションの終了は、RT-edge 終了サービス「EgShDown」により行います。I-I/O サービスコンテナの終了設定も同様、RT-edge 終了サービス設定に準拠します:

- 1) C:\RTedge\bin\EgShDown.xml をテキストエディタで開きます。
- 2) ArrayOfAnyType エlement 内に、I-I/O コンテナの使用プロトコル名を追加します。

```
<anyType xsi:type="xsd:string">ECAT</anyType>
```

- 3) 編集を保存し、ファイルを閉じます。
- 4) 追加結果は以下のようになります。

```
...:
<ArrayOfAnyType ...>
  <anyType xsi:type="xsd:string">EgLog</anyType>
  <anyType xsi:type="xsd:string">EgTime</anyType>
  <anyType xsi:type="xsd:string">EgBoot</anyType>
  <anyType xsi:type="xsd:string">ECAT</anyType>1
</ArrayOfAnyType>
```

図 8.I-I/O コンテナ終了登録

RT-edge ソフトウェアの終了時は、C:\RTedge\bin\EgShDown.exe を実行します。

## 5. 環境設定

### 5.1. Edge コントローラ設定

I-I/O コンテナでは、通信プロトコルごとの設定が必要です。設定する内容、通信プロトコルごとの仕様については、10.5. 環境設定 その他仕様制限事項をご参照ください。

### 5.2. 接続機器設定

接続機器側で行う設定については、各製品のマニュアルをご参照ください。

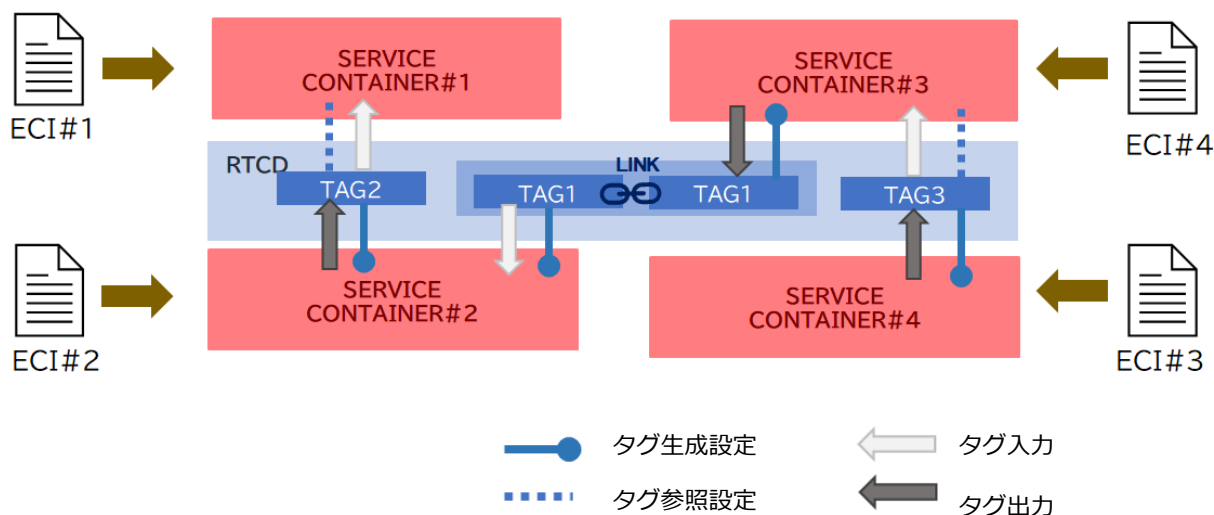
## 6. 設定概要

サービスコンテナ設定により担当するターゲットのデータと RTCD との接続が可能となります。  
サービスコンテナにおける基本設定は、タグ、データセットの定義を主とした RTCD 設定と、サービスコンテナの入出力周期やプライオリティ設定等、コンテナプロパティ設定に分類されます:

設定項目	説明
RTCD 設定	<p><u>タグ設定</u></p> <p>ローカルタグ生成設定・リンクタグ生成設定 タグ参照設定</p> <p><u>データセット設定</u></p> <p>タグ・コレクション定義 周期・プライオリティ設定</p>
コンテナプロパティ設定	<p>データ更新方式(オンデマンド・サイクリック(周期設定) プライオリティ設定等</p> <p>※コンテナプロパティ値は各サービスコンテナにより実装が異なります。</p>

### 6.1. ECI 設定

RTCD 設定、プロパティ設定は、コンテナ毎に定義する設定情報(ECI: RT-edge コンテナ設定情報)に基づきます。ECI ファイルは XML 形式のテキストファイルとして生成されています:



各サービスコンテナは、タグ・リンクタグ生成設定を行います。  
タグに対する入出力方向設定・参照設定を行います。

## RTCD 設定

RTCD 設定では、ECI ファイル内で編集する XML タグの編集要素は以下のように定義されています:

設定項目	設定手順												
タグ設定	ローカルタグ <div> &lt;Tags&gt;エレメント内に、&lt;Tag&gt;を生成します。  サービスコンテナ独自の名称(一意名)で&lt;Tag Name=の設定を行います。 </div>												
生成	リンクタグ <div> &lt;Tages&gt;エレメント内に、&lt;Tag&gt;を生成します。  他サービスコンテナの提供するタグと同名で&lt;Tag Name=の設定を行います。 </div>												
	<div> <div>Tag</div> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>公開するタグ名を設定します。</td></tr> <tr> <td>Type</td><td>RT-edge データ型に関連する型定義値を設定します。 ※参照: データ型</td></tr> <tr> <td>Size</td><td>タグデータサイズを指定します。 ※参照: データ型</td></tr> <tr> <td>Address</td><td>サービスコンテナにおけるデータ取得元、宛先となるアドレス情報を指定します。本アドレス書式は、サービスコンテナ毎に異なります。</td></tr> <tr> <td>Comment</td><td>タグに対するコメントを設定します。</td></tr> </table> </div>	キーワード	説明	Name	公開するタグ名を設定します。	Type	RT-edge データ型に関連する型定義値を設定します。 ※参照: データ型	Size	タグデータサイズを指定します。 ※参照: データ型	Address	サービスコンテナにおけるデータ取得元、宛先となるアドレス情報を指定します。本アドレス書式は、サービスコンテナ毎に異なります。	Comment	タグに対するコメントを設定します。
キーワード	説明												
Name	公開するタグ名を設定します。												
Type	RT-edge データ型に関連する型定義値を設定します。 ※参照: データ型												
Size	タグデータサイズを指定します。 ※参照: データ型												
Address	サービスコンテナにおけるデータ取得元、宛先となるアドレス情報を指定します。本アドレス書式は、サービスコンテナ毎に異なります。												
Comment	タグに対するコメントを設定します。												
参照	<div> 入力参照: &lt;TagRefs_IN&gt; エレメント内  出力参照: &lt;TagRefs_OUT&gt; エレメント内  &lt;TagRef Name=に参照するタグを指定します。 </div>												
	<div> <div>TagRef</div> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>参照するタグを指定します。</td></tr> </table> </div>	キーワード	説明	Name	参照するタグを指定します。								
キーワード	説明												
Name	参照するタグを指定します。												
データセット設定	<div> &lt;Datasets&gt;エレメント内に、&lt;Dataset&gt;を作成し  &lt;Dataset Name=&gt;にデータセット名を指定します。  Dataset エレメント内に、&lt;TagRefs&gt;エレメントを作成します。  &lt;TagRef Name=に参照するタグを指定します。 </div>												
タグ・コレクション定義	<div> <div>TagRef</div> <table> <tr> <th>キーワード</th><th>説明</th></tr> <tr> <td>Name</td><td>参照するタグを指定します。</td></tr> </table> <div>※Dataset 内に TagRef オブジェクトを列挙します。</div> </div>	キーワード	説明	Name	参照するタグを指定します。								
キーワード	説明												
Name	参照するタグを指定します。												

設定項目	設定手順
周期・プライオリティ 設定	<p>&lt;Collectors&gt;エレメント内に、&lt;Collector&gt;を作成し &lt;Collector Name=&gt;に名称を設定します (Dataset を収集する機能名)</p>
<b>Collector</b>	
キーワード	説明
Name	Dataset 収集機能名を指定します
Interval	収集周期を指定します (1ms 単位)
Priority	プライオリティを設定します
DatasetName	収集するデータセット名を指定します。

## コンテナプロパティ設定

サービスコンテナプロパティ設定値は、Tag として登録されており、サービスコンテナ実装毎に数や種類は異なります。規定値プロパティは、SERVICE.キーワードをプリフィックスとしたタグ名で登録されています:

設定項目	設定手順
コンテナプロパティ値	<p>&lt;Tags&gt;エレメント内の&lt;Tag Name=&gt;において、SERVICE.キーワードをプリフィックスとしたタグは、コンテナプロパティタグです:</p>
<b>Tag</b>	
キーワード	説明
Name	SERVICE. キーワードをプリフィックスとした名称で設定されています。
Type	RT-edge データ型に関連する型定義値を設定します。
Size	タグデータサイズを指定します。
Value	設定値
Comment	タグに対するコメントを設定します。



コンテナプロパティ設定については、各サービスコンテナに付与するユーザズマニュアルを参照してください。

## 7. I-I/O サービスコンテナ設定

I-I/O コンテナにおける RTCD 設定には、制御対象機器の入出力情報と RTCD との接続設定が含まれます。プロパティ設定ではネットワーク番号やノード番号など、サービスコンテナに特化した設定が含まれます。

### 7.1. RTCD 設定概要

I-I/O サービスコンテナではタグ定義設定を行います:

制御対象機器の入出力情報 を RTCD に公開	ローカルタグ生成 + タグ参照	制御対象機器の入出力情報を RTCD に定義するタグのデータ入出力先として割り当てる。
----------------------------	-----------------------	---

#### ■ ローカルタグ生成

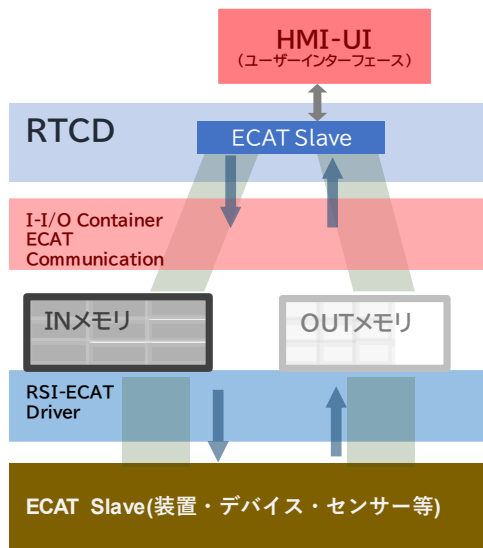
RT-edge システム内で一意となるタグを定義し、データ取得元、データ宛先の設定を行います。

#### データ取得元とデータ宛先設定

- ECI(Edge コンテナ設定情報:EgIIIO.xml)の編集

ローカルタグとして定義したタグのデータ取得元/宛先を設定します。I-I/O サービスコンテナ用 ECI(Edge コンテナ設定情報:EgIIIO.xml)に、タグを定義し、データ取得元、データ宛先となる要素として、制御対象機器の入出力情報を割り当てます。

ここでは、EtherCAT プロトコルを利用して接続スレーブ情報を割り当てた例を以下に記載します。



ローカルタグを生成し、タグのデータ取得元、もしくはデータ宛先を EtherCAT スレーブ(装置やデバイス、センサーなどに接続)の値を公開しているドライバの I/O メモリに設定します。

図 10. EtherCAT 通信を用いたデバイスアクセス

## ■ タグ参照設定

I-I/O サービスコンテナでは、生成したローカルタグに対する内部参照設定が必要です。本設定により生成したタグに対する入出力方向を定義し、EtherCAT 通信などで装置やデバイス、センサーなどの値を公開しているメモリとタグデータの更新が行われます。

## 7.2. ECI(コンテナ設定情報ファイル)の編集

ECI の編集を行います。下記では EtherCAT 通信プロトコルを利用した例を記載します。

### 1) タグ生成

- 1) ECI コンテナ設定情報(EgIIO.xml)ファイルを開きます。
- 2) RTedge > Tags 内に、Tag エlementを追加します。

<code>&lt;Tag Name="CJ2M.CH[0]" Type="3" Address="ECAT/16#0000" Comment="チャンネルI/0"/&gt;</code>	①
<code>&lt;Tag Name="CJ2M.A[1]" Type="3" Address="ECAT/9" Comment="A レジスタ"/&gt;</code>	②
<code>&lt;Tag Name="CJ2M.A[0]W" Type="5" Address="ECAT/16#00011090" Comment="A レジスタ(word)"/&gt;</code>	③

ECAT VIOS\_IN 領域(0x00000000~0x0000FFFF) / ECAT VIOS\_OUT 領域(0x00010000~0x0001FFFF)

- ① ECAT VIOS\_IN 領域 オフセット 0 を BYTE 型(3)のタグに紐づけます。
- ② ECAT VIOS\_IN 領域 オフセット 9 を BYTE 型(3)のタグに紐づけます。
- ③ ECAT VIOS\_OUT 領域 オフセット 0x1090(4240)を WORD 型(5)のタグに紐づけます。

#### (ア) Name 設定

RT-edge のタグ名を指定します。

#### (イ) Type 設定

以下の表を参考に、入出力を行うデータ型(Type)を指定します。

表 7. データ型に対応する Type 一覧

PLC データ型	Type(数値)	Size	用途	RT-edge データ型(egTag)
BOOL	Type="1"	Size="1"	BOOL 値	Boolean
INT	Type="4"	Size="2"	符号付き 16bit 整数	Int16
DINT	Type="6"	Size="4"	符号付き 32bit 整数	Int32
LINT	Type="8"	Size="8"	符号付き 64bit 整数	Int64
UINT	Type="5"	Size="2"	符号なし 16bit 整数	UInt16
UDINT	Type="7"	Size="4"	符号なし 32bit 整数	UInt32
ULINT	Type="9"	Size="8"	符号なし 64bit 整数	UInt64
REAL	Type="10"	Size="4"	単精度実数(32bit)	Float
LREAL	Type="11"	Size="8"	倍精度実数(64bit)	Double
WORD	Type="5"	Size="2"	符号なし 16bit 整数	UInt16
DWORD	Type="7"	Size="4"	符号なし 32bit 整数	UInt32
LWORD	Type="9"	Size="8"	符号なし 64bit 整数	UInt64
-	Type="2"	Size="1"	符号付き 8bit 整数	SByte
-	Type="3"	Size="1"	符号なし 8bit 整数	Byte

(ウ) Address 設定

I-I/O コンテナにおける Address 設定は以下フォーマットに沿って設定します:

Address = "ECAT/16#00ff"

①

②

表 8. Address フォーマット

名称	設定値	説明
① 通信プロトコル名	※説明欄参照	通信プロトコル名を指定します。 指定可能な通信プロトコル名は付録 10.4. ECI Address 定義を参照ください。
② Edge I/O ポート オフセット値	※説明欄参照	Edge I/O ポートのオフセット値を指定します。 オフセット値は、10 進数表記、16 進数表記に対応しています。16 進数表記の場合は先頭に"16#"を付けることで、16 進数と解釈します。  10 進数表記：255 16 進数表記：16#00ff  Edge I/O ポートのアドレス定義は通信プロトコルごとに異なります。詳細は付録 10.4. ECI Address 定義を参照ください。

## ◆ セグメントタグ使用

上記で説明した指定の方法は、チャネルごとやレジスタごとにタグを割り当てる方法でしたが、IN メモリ空間や OUT メモリ空間をそれぞれ一つの Tag(セグメントタグ)に割り当てることもできます。

- 1) ECI コンテナ設定情報(EgIIO.xml)ファイルを開きます。
- 2) RTedge > Tags 内に、Tag エlementを追加します。
- 3) RTedge > Tags 内に、Tag エlementを追加します。

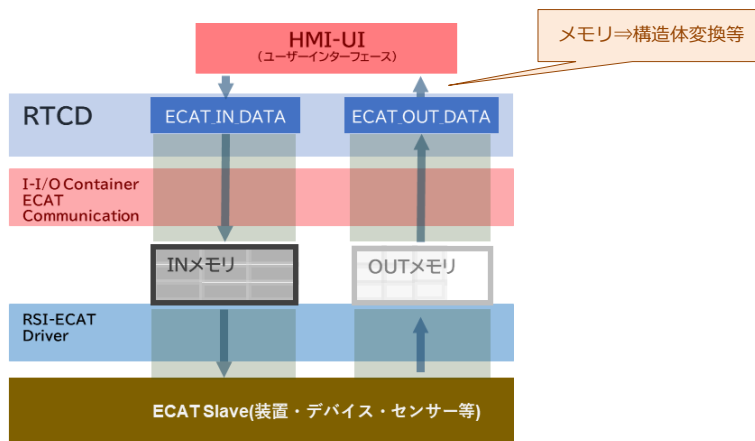
```
<Tag Name="ECAT_IN_DATA" Type="13" Size="1024" Address="ECAT/16#0000" Comment="ECAT IN データ"/>
<Tag Name="ECAT_OUT_DATA" Type="13" Size="4096" Address="ECAT/16#10000" Comment="ECAT OUT データ"/>
```

①

②

ECAT VIOS\_IN 領域(0x00000000~0x0000FFFF) / ECAT VIOS\_OUT 領域(0x00010000~0x0001FFFF)

- ① ECAT VIOS\_IN 領域 オフセット 0、サイズ 1024byte を Segment 型(13)のタグに紐づけます。
- ② ECAT VIOS\_OUT 領域 オフセット 0、サイズ 4096byte を Segment(13)のタグに紐づけます。







割り当てる位置はセグメントタグのオフセット 0 固定です。また上記の例での IN メモリから取得するデータサイズは、ECI で指定したタグの Size 値です。Size に取得対象のメモリサイズより大きいサイズは指定しないでください。

## 2) 入力タグ参照

EtherCAT 通信で扱うメモリ領域（I/O メモリ領域）等から、入力を行うタグを設定します。デバイスや外部機器からの入力が不要な場合、本設定は行いません。

- 1) RTEdge > Services > Service(Name 属性="プロトコル名") > TagRefs > TagRefs\_IN エレメントを追加します。
- 2) TagRefs\_IN エレメント内に、TagRef エレメントを追加します。
- 3) TagRef の Name 属性には、使用するローカルタグを設定します。

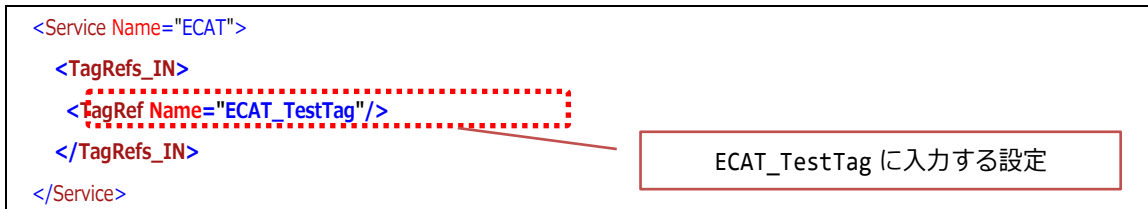


図 11. TagRefs\_IN の設定



存在しないタグを指定した場合は無視されます。

## 3) 出力タグ参照

EtherCAT 通信で扱うメモリ領域（I/O メモリ領域）等に対して出力を行うタグを設定します。デバイスへの出力が不要な場合、本設定は行いません。

- 1) RTEdge > Services > Service(Name 属性="プロトコル名") > TagRefs > TagRefs\_OUT エレメントを追加します。
- 2) TagRefs\_OUT エレメント内に、TagRef エレメントを追加します。
- 3) TagRef の Name 属性には、使用するローカルタグを設定します。

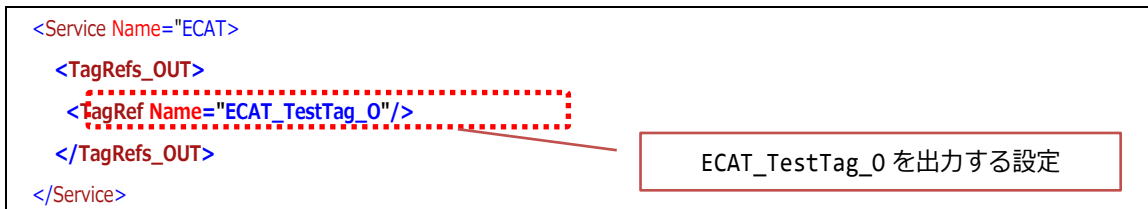


図 12. TagRefs\_OUT の設定



存在しないタグを指定した場合は無視されます。

## 7.3. コンテナプロパティ設定

I-I/O コンテナには、以下の主要なプロパティ設定があります。サービスコンテナのプロパティ設定はタグの一部として構成されています。以下にコンテナで共通なコンテナプロパティ定義を記載します。

表 9.コンテナプロパティ設定

設定	Tag 名	デフォルト設定	内容
サービス 起動設定	<b>SERVICE.&lt;通信プロトコル名&gt;.AutoRun</b>	1 (タグデータ更新開始)	サービス起動時のタグデータ更新自動開始を指定します。
タグ更新周期	<b>SERVICE.&lt;通信プロトコル名&gt;.Cycle</b>	1000	タグデータの入出力周期を設定します。(ms)



コンテナプロパティタグはあらかじめ定義されており、名称は、SERVICE.から開始しています。



IP アドレスやポート番号といったプロパティ設定は通信プロトコルごとに異なります。通信プロトコルごとに設定が必要なプロパティタグの一覧は付録 10.1. サービスプロパティタグを参照ください。

## 8. 設定サンプル

ここでは I-I/O コンテナ(R)ECAT の設定を例に説明します。

### 1) EtherCAT スレーブの準備・設定(RT-edge インストール PC)

RSI-ECAT ドライバのインストールを行い、EtherCAT スレーブの準備・設置を行ってください。またドライバの設定や、NIC の設定、ENI ファイルの作成から起動まで、「RSI-ECAT ユーザーズマニュアル」を参考に、設定を行ってください。

### 2) タグ設定

「7.1. RTCD 設定概要」を参考に、タグの生成・入出力設定を行います。各タグの Address に Edge I/O ポートのアドレス値を設定します。

```
<Services>
  <Service Name="ECAT" >
    <Tags>
      ...
      <Tag Name="CJ2M.CH[0]" Type="3" Address="ECAT/16#0000" Comment="チャンネル I/O"/>
      <Tag Name="CJ2M.A[0]" Type="3" Address="ECAT/9" Comment="A レジスタ"/>
      <Tag Name="CJ2M.A[10]WORD" Type="5" Address="ECAT/16#0001000A" Comment="A レジスタ(word)"/>
      <Tag Name="CJ2M.CH[3]" Type="3" Address="ECAT/3" Comment="チャンネル I/O"/>
      <Tag Name="CJ2M.CH[5]" Type="3" Address="ECAT/16#00010005" Comment="チャンネル I/O"/>
    </Tags>
    <TagRefs_IN>
      <TagRef Name="CJ2M.CH[0]"/>
      <TagRef Name="CJ2M.A[0]"/>
      <TagRef Name="CJ2M.CH[3]"/>
    </TagRefs_IN>
    <TagRefs_OUT>
      <TagRef Name="CJ2M.A[10]WORD"/>
      <TagRef Name="CJ2M.CH[5]"/>
    </TagRefs_OUT>
  </Service>
</Services>
```

## 9. 動作確認

### 9.1. 動作確認手順

ここでは I-I/O コンテナ(R) ECAT の設定を例に説明します。

RT-edge ソフトウェアを起動し、I-I/O コンテナ(R) ECAT が正常に起動することを確認します：

- 1) Edge コントローラと接続機器の接続設定を行います。例ではデジタル入出力スレーブを用いています。
- 2) Edge コントローラとデジタル入出力スレーブを LAN ケーブルで接続します。

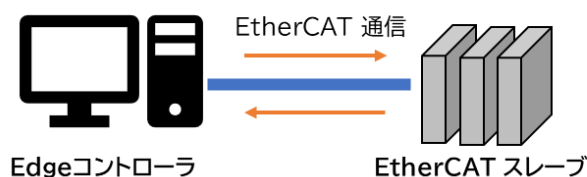


図 13. 接続構成図

- 3) RT-Edge ソフトウェア(C:\¥RTedge¥EgBoot.exe)を開始します。
- 4) RT-edge オブジェクトブラウザ(C:\¥RTedge¥EgBrow.exe)を起動します。



RT-edge 起動直後に RT-edge オブジェクトブラウザを起動すると初期化中の為、想定されるタグが表示されない場合があります。一度 RT-edge オブジェクトブラウザを終了し、再度起動させてください。

- 5) RT-edge コンテナ設定情報(ECI)で定義したタグが表示されていることを確認します。

RT-edge Object Browser のウィンドウのスクリーンショット。タブは「Tags」が選択されています。以下の表が表示されています。

Name	Current Value	Type	Source	Comment
SERVICE.ECAT.Live	00042f24 (274212)	UInt32		
SERVICE.ECAT.MboxNameValue	00000023 (35)	UInt32		
SERVICE.ECAT.MboxColSize	00000800 (2048)	UInt32		
SERVICE.ECAT.MboxRowSize	00000400 (1024)	UInt32		
SERVICE.ECAT.Mode	01 (1)	byte		1=入出力Auto
SERVICE.ECAT.AutoRun	True (1)	bool		1=起動時に開始
SERVICE.ECAT.Cycle	00000001 (1)	UInt32		Refresh Cycletime in ms
SERVICE.ECAT.Timeout	00001770 (6000)	UInt32		Driver API Timeout time in
SERVICE.ECAT.ConnectChkCycle	000003e8 (1000)	UInt32		Connection Check Cycletim
SERVICE.ECAT.InThreadNum	01 (1)	byte		Number of input threads g
SERVICE.ECAT.OutThreadNum	01 (1)	byte		Number of output threads
SERVICE.ECAT.EdgeIOPortEnabled	True (1)	bool		Edge I/O Port Use Flag
SERVICE.ECAT.NodeName	00 (0)	byte	NodeA	INtime Node Name
SERVICE.ECAT.ReConnectCount	00000003 (3)	UInt32		Connection Check Count
CJ2M.CH[0]	01 (1)	byte	ECAT/16#000	チャンネル/O
CJ2M.A[0]	00 (0)	byte	ECAT/9	レジスタ
CJ2M.A[10]WORD	0000 (0)	UInt16	ECAT/16#000	レジスタ(word)
CJ2M.CH[3]	00 (0)	byte	ECAT/3	チャンネル/O
CJ2M.CH[5]	00 (0)	byte	ECAT/16#000	チャンネル/O

下部には「CSV出力」「一覽再取得」ボタンと「76 items available」の表示があります。

図 14. コンテナ生成タグイメージ

## 9.2. サービスインジケータタグ

動作確認で必要となるサービスインジケータタグについて説明します:

表 10. サービスインジケータ共通タグ一覧

サービスインジケータ Tag 名	備考
SERVICE.<通信プロトコル名>.Status	現在のサービス起動状態を示します
SERVICE.<通信プロトコル名>.Error	現在のサービスエラー状態を示します
SERVICE.<通信プロトコル名>.Run	現在のデータタグ参照・更新動作の状態を示します
SERVICE.<通信プロトコル名>.Live	サービスが健全であることを示すカウンタ

上記 Tag の詳細はユーザーズマニュアルを参照ください。

## 9.3. RT-edge タグデータの妥当性について

タグリンクされた RT-edge タグのデータは、サービスインジケータが以下の状態になっている時、妥当であると判断出来ます。

1. SERVICE.<通信プロトコル名>.Error = FALSE (エラーが発生していない)
2. SERVICE.<通信プロトコル名>.Run = TRUE (タグ参照・更新が実行されている)

上記の状態になっていない場合には、タグデータの入出力に何らかの問題が発生しています。詳細は、以下のトラブルシューティングをご参照ください。

- 「サービスインジケータの.Error が true になっています。」
- 「サービスインジケータの.Run が false になっています。」

# 10. 付録

## 10.1. サービスプロパティタグ

I-I/O コンテナでは以下のサービスプロパティタグを使用します。

### 共通

表 11. サービスプロパティタグ一覧(共通)

サービスプロパティ Tag 名	Type	初期値	概要
SERVICE.<通信プロトコル名>.Cycle	UInt32	1000	動作サイクルタイムを指定します。 1~65535(ms) (1ms の指定は可能ですが、使用するタグ数が多いほど Read/Write 処理に時間を要します。タグ数に応じて Cycle 時間を増やしてください。)
SERVICE.<通信プロトコル名>.ExecuteReConnect	Boolean	0	エラー時、再接続するか否かを指定します。1=再接続処理を実施。0=再接続処理を実施しない(デフォルト)
SERVICE.<通信プロトコル名>.UseWait	Boolean	1	プロトコルのイベント応答後、1 入出力処理を行います。イベントを待つ場合は True を。イベントを待たずにサイクリックで処理する場合は False を指定します。 EtherCAT の場合は、EtherCAT Master のサイクル処理開始を待機し、応答時 1 入出力処理を行います。
SERVICE.<通信プロトコル名>.WaitTimeout	UInt32	0xFFFFFFFF (WAIT_FOREVER)	プロトコルのイベント待ちタイムアウト時間(ms)を指定します。デフォルトは、WAIT_FOREVER(待ち続ける)です。
SERVICE.<通信プロトコル名>.Mode	Byte	1	動作モードを指定します。1=入出力 Auto (システムで使用する為、変更不可)
SERVICE.<通信プロトコル名>.AutoRun	Boolean	1	動作自動スタートの有効無効を指定します。1=起動時に開始 (システムで使用する為、変更不可)
SERVICE.<通信プロトコル名>.InThreadNum	Byte	1	入力動作スレッド生成数を指定します。(システムで使用する為、変更不可)
SERVICE.<通信プロトコル名>.OutThreadNum	Byte	1	出力動作スレッド生成数を指定します。(システムで使用する為、変更不可)
SERVICE.<通信プロトコル名>.edgeIOPortEnabled	Boolean	1	Edge I/O ポート使用有無を指定します。(システムで使用する為、変更不可)
SERVICE.<通信プロトコル名>.Timeout	Byte	10000	ドライバの通信タイムアウト時間を指定します。

背景がグレーの Tag はシステムで利用しているため、値の変更はできません。

### ECAT

EtherCAT 通信プロトコルを利用時のサービスプロパティタグは以下の通りです。

表 12. サービスプロパティタグ一覧(ECAT)

サービスプロパティ Tag 名	Type	初期値	概要
SERVICE.ECAT.NodeName	Byte	0	ノード名を指定します。 Address 部分にノード名称を指定してください。 デフォルト：NodeA
SERVICE.ECAT.WAIT_ESM	Boolean	0	通信回線オープン時の動作を指定します。 0=マスタースタートを Operational に自分で変更する 1=マスタースタートが Operational になるまで待機する

## ECAT(CAN 通信併用時)

EtherCAT + CAN 通信利用時は、ECAT 使用時のサービスプロパティタグに加えて使用可能なタグがあります。詳細については、「RSI-CAN-E セットアップマニュアル」を参照ください。

## 10.2. サービスインジケータタグ

### ECAT(CAN 通信併用時)

EtherCAT + CAN 通信利用時は、通常のサービスインジケータタグに加えて使用可能なタグがあります。詳細については、「RSI-CAN-E API リファレンスマニュアル」を参照ください。

## 10.3. サービスメッセージ

I-I/O コンテナでは以下のメッセージに対する処理が実装されています:

### 共通メッセージ

表 13.共通メッセージ一覧

メッセージ名	番号	説明
EM_SERVICE_STOP	101	サービスを終了させます。
EM_SERVICE_RUN	102	データ更新処理を開始します。
EM_SERVICE_PAUSE	103	データ更新処理を一時停止します。再開するためには、EM_SERVICE_RUN を送信してください。

各メッセージ送信後のサービスインジケータタグは以下の状態に遷移します:

表 14.メッセージ送信後のサービスインジケータタグの状態一覧

サービスインジケータタグ	初期化完了 (.AutoRun=True)	初期化完了 (.AutoRun=False)	EM_SERVICE STOP	EM_SERVICE RUN	EM_SERVICE PAUSE
SERVICE.<通信プロトコル名>.Status	1	1	0	1	1
SERVICE.<通信プロトコル名>.Run	TRUE	FALSE	FALSE	TRUE	TRUE
SERVICE.<通信プロトコル名>.Live	増加	停止	停止	増加	増加
SERVICE.<通信プロトコル名>.Error	FALSE	FALSE	FALSE	FALSE	FALSE

## 10.4. ECI Address 定義

### ECAT

通信プロトコル名: ECAT

### Edge ECAT I/O ポート アドレス構造

表 15.I/O ポートアドレス構造表

領域 No	領域名称	仮想レジスタ空間 アドレス範囲	説明
0	VIOS IN 領域	0x00000000 ~ 0x0000FFFF	ベースアドレス 0x00000000 から、情報領域にある VIOS IN 領域のサイズ分のみ有効なデータになります。
1	VIOS OUT 領域	0x00010000 ~ 0x0001FFFF	ベースアドレス 0x40000000 から、情報領域にある VIOS OUT 領域のサイズ分のみ有効なデータになります。
2	情報領域	0x00020000 ~ 0x0002FFFF	情報領域については、以下の情報領域を参照してください。
3	情報領域(統計情報)	0x00030000 ~ 0x0003001C	情報領域(統計情報)については、以下の情報領域(統計情報)を参照してください。
4	情報領域 (CyclicAnalyzer)	0x00040000 ~ 0x00040144	情報領域(CyclicAnalyzer)については、以下の情報領域(CyclicAnalyzer)を参照してください。

ECI で指定する IO address は以下の計算式で求めることができます。

$$\text{IO address} = \text{領域 No} \times 0x10000 + \text{オフセット}$$

(VIOS\_IN : 領域 No=0 VIOS\_OUT : 領域 No=1 ...)

例: VIOS\_OUT 領域 オフセット 8 (byte 単位で指定)

$$\text{IO address} = 16\#10008 (= 1 \times 0x10000 + 8)$$



## 情報領域

情報領域には、EtherCAT の通信に関する情報が含まれます。

以下の表のオフセットは、情報領域のベースアドレス 0x00020000 を加算してアクセスしてください。書き込みはできません。

表 16.情報領域 データ構造表

オフセット (単位:byte)	サイズ (単位:byte)	説明
0	2	ENI 内のスレーブ定義数
4	2	オンライン上で検出されたスレーブ数
128	4	VIOS IN 領域のサイズ(単位:byte)
132	4	VIOS OUT 領域のサイズ(単位:byte)
136	4	情報領域のサイズ(単位:byte)
1024	4	EtherCAT Master 診断情報
1028	4 x 8192	EtherCAT Slave 診断情報(1 台あたり 4byte、接続順、最大 8192 台まで)

## 情報領域(統計情報)

情報領域(統計情報)には、RSI-ECAT HI API の統計情報が含まれます。

以下の表のオフセットは、情報領域(統計情報)のベースアドレス 0x00030000 を加算してアクセスしてください。書き込みはできません。

表 17.情報領域(統計情報) データ構造表

オフセット (単位:byte)	サイズ (単位:byte)	説明
0	4	HI API 遷移状態(0:オープン済み, 1:未オープン, 2:未初期化)
4	4	Init API コール回数
8	4	Open API コール回数
12	4	Close API コール回数
16	4	Read API コール回数
20	4	Write API コール回数
24	4	最終エラーコード

### 情報領域(CyclicAnalyzer)

情報領域(CyclicAnalyzer)には、RSI ECAT Master のツール「CyclicAnalyzer」に表示されているサイクル周期情報が含まれます。

以下の表のオフセットは、情報領域(CyclicAnalyzer)のベースアドレス 0x00040000 を加算してアクセスしてください。書き込みはできません。

表 18.情報領域(CyclicAnalyzer) データ構造表

オフセット (単位:byte)	サイズ (単位:byte)	説明
0	4	マスターステータス (1:Init, 2:Pre-Operational, 4:Safe-Operational, 8:Operational)
4	4	リクエストステータス (1:Init, 2:Pre-Operational, 4:Safe-Operational, 8:Operational)
8	4	トータルサイクルカウント数
16	4	最大周期時間 (usec) ※float 型
20	4	最小周期時間 (usec) ※float 型
24	4	平均周期時間 (usec) ※float 型
28	4	カーネルティック設定値 (usec)
32	4	サイクリック時間 (usec)
304	4	現在のサイクリック時間 (usec) ※float 型

## 10.5. 環境設定 その他仕様制限事項

### ECAT

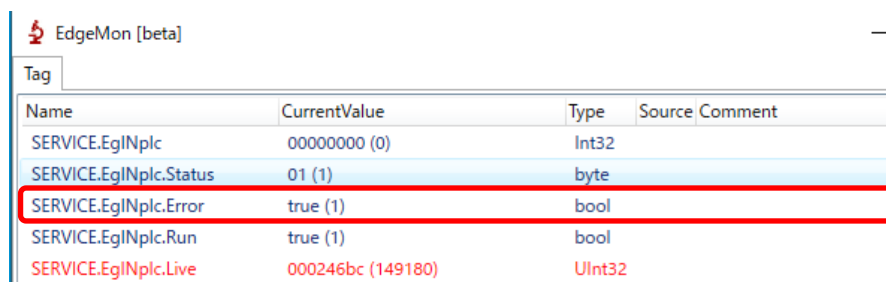
- 1) 弊社ドライバ製品「RSI-ECAT」をインストールしている環境で動作します。

### EtherCAT + CAN 通信

- 1) 弊社ドライバ製品「RSI-ECAT」および「RSI-CAN-E」をインストールしている環境で動作します。

## 10.6. トラブルシューティング

サービスインジケータの.Error が true になっています。



Name	CurrentValue	Type	Source	Comment
SERVICE.EglNplc	00000000 (0)	Int32		
SERVICE.EglNplc.Status	01 (1)	byte		
SERVICE.EglNplc.Error	true (1)	bool		
SERVICE.EglNplc.Run	true (1)	bool		
SERVICE.EglNplc.Live	000246bc (149180)	UInt32		

図 15. Error インジケータが true

「.Error」タグが True になる場合、何れかの通信機器との接続に問題がある状態となります。

通信機器と RT-edge コンテナ設定情報(ECI)ファイルについて、以下をご確認ください:

- 1) ネットワーク設定 (IP アドレス・ポート番号) は正しいか?
- 2) 通信設定 (ネットワークアドレス・ノードアドレス・ユニット No) 等は正しいか?
- 3) 物理的に接続されているか (ケーブル断線・通信機器の電源が入っていないなど) ?

サービスインジケータの.Run が false になっています。

- 原因: サービス開始要求を受け付けられていません。

サービス開始要求が無い、初期化処理が終わらずサービス開始要求を受け付けられていません。

- 対応: サービス開始要求を行います。

アプリケーションから「EM\_SERVICE\_RUN」メッセージを送る、または EgFINS.xml にて「.AutoRun」プロパティの Value を「1」とします。

RT-edge オブジェクトブラウザにサービスインジケータが表示されません。

- 原因: I-I/O コンテナ初期化処理中です。

I-I/O コンテナ初期化処理中では、タグが生成されていません。

- 対応: 終了させ、10 秒程度待ったのち再度 RT-edge オブジェクトブラウザを起動させます。

I-I/O コンテナ初期化処理完了後に再度起動させます。

タグデータが更新されません。

- 原因: タグ参照・更新設定がされていません。

サービスインジケータ「.Run」が TRUE にも関わらずタグデータが更新されない場合、タグ参照・更新設定がされていない可能性があります。

- 対応：タグ参照・更新設定を行います。

設定方法については「7.2. ECI(コンテナ設定情報ファイル)の編集」を参照ください。

### 起動時にコンソールが表示され、「wrong number of arguments (given 0, expected 1+)」が出力されます

- 原因：RSL のロード設定がされていません。

EglIO.rta の起動設定において、通信プロトコルごとの RSL のロード設定がされていない可能性があります。

- 対応：タグ参照・更新設定を行います。

Argument 属性を設定する必要があります。設定方法については「4.3. 起動設定」を参照ください。

### 各エラーコードの原因・対応方法

何らかの問題が発生しサービス異常状態ステータス「.Error」タグが TRUE になっている場合、エラーコードから理由を確認可能です。最後に発生したエラーコードは「.○○○LastErr」タグで確認可能です。各エラーコードの内容・対応方法を以下に記載します。

(○○○にはスレッド名、処理名が入ります)

表 19.エラーコード

エラーコード	内容	対策
0x00000000	正常	
0xFFFF0002	フレーム送受信中のエラー	
0xFFFF0004	受信フレーム処理中のエラー	
0xFFFF0008	NIC アダプタとスレーブ間の接続ダウン	
0xFFFF0010	コンフィグ構成異常	接続状態・順序、またはスレーブ側でエラーになっていないか確認してください。
0xFFFF0020	スレーブ To スレーブ通信でタイムアウト発生	
0xFFFF0040	パケット送信/受信エラーが多数発生したため、デフォルトデータが使用された	
0xFFFF0080	ウォッチドッグタイムアウトが発生	

上記以外のエラーコードは、ドライバ API マニュアルに記載のエラーコードリストを参照ください。



更新履歴

版	日付	更新説明
1	2022.12	初回版
2	2023.04	サービスプロパティタグに以下を追加 SERVICE.<通信プロトコル名>.ExecuteReConnect
3	2024.01	入出力対象に Segment のタグを追加
4	2024.09	サービスプロパティタグに以下を追加 SERVICE.<通信プロトコル名>.UseWait SERVICE.<通信プロトコル名>.WaitTimeout
5	2025.04	EtherCAT + CAN 通信機能に対応 図 6.EgIIIO 起動登録 の記載を修正 その他軽微な修正

INDUSTRIAL REALTIME EDGE COMPUTERS

I-I/O Container ユーザーズマニュアル

発行元:株式会社マイクロネット

TEL: +81(0)299-90-1733

FAX:+81(0)299-90-8557

- ・ 本書の内容、及び付属のソフトウェアの全部または一部を無断で転載することは禁止しております。
- ・ 本製品の内容については、将来予告なしに変更することがあります。
- ・ 本製品の内容について万一ご不審な点や記載もれなどお気づきの点がございましたら、お手数ですが、当社までご連絡ください。
- ・ Windows XP、Windows 7、Windows 8、Windows 10 等、Windows は、米国 Microsoft Corporation における登録商標です。
- ・ Visual Studio、Visual C++等は、米国、およびその他の国における Microsoft Corporation の登録商標です。
- ・ INtime は米国 TenAsys における登録商標です。
- ・ その他、記載されている会社名、製品名は、各社の商標又は登録商標です