

# RT-edge

Micronet.Co,

マイクロネット  
Micronet

INDUSTRIAL REALTIME EDGE COMPUTERS

# RT-edge




IOSample プログラム取扱説明書

 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。

## 目次

<b>1. はじめに</b>	<b>3</b>
1.1. 概要.....	3
1.2. 用語解説.....	3
1.3. 関連資料.....	5
<b>2. 処理内容</b>	<b>6</b>
2.1. プログラム構成.....	6
2.2. アクセスするタグ .....	7
2.3. IOSample サービスコンフィグファイル .....	7
<b>3. 実行</b>	<b>8</b>
3.1. 起動方法 .....	8
3.2. 画面説明 .....	9
3.3. 確認方法 .....	9
■ 読み込み値の確認 .....	10
■ 書き込み値の確認 .....	10
3.4. 終了方法 .....	11
<b>4. アクセスタグの変更方法</b>	<b>12</b>
4.1. 変更場所 .....	12
4.2. 変更方法 .....	12

## 1. はじめに

### 1.1. 概要

本ドキュメントでは、製品付属サンプル IOSample について解説します。IOSample では簡単なタグアクセス処理を C#プログラムで作成しています。

具体的には 1 つのタグを周期的に読み込むサービス処理と、画面から任意のタイミングで値を書き込む処理が定義されています。

本サンプルを確認することで、所定のタグへの簡単なアクセス処理の実装例と、その動作を確認できます。また、ソースコード内のタグ定義を変更することでアクセスするタグを変更できるので、他のサービスが生成したタグへの読み込み/書き込み処理の確認もできます。

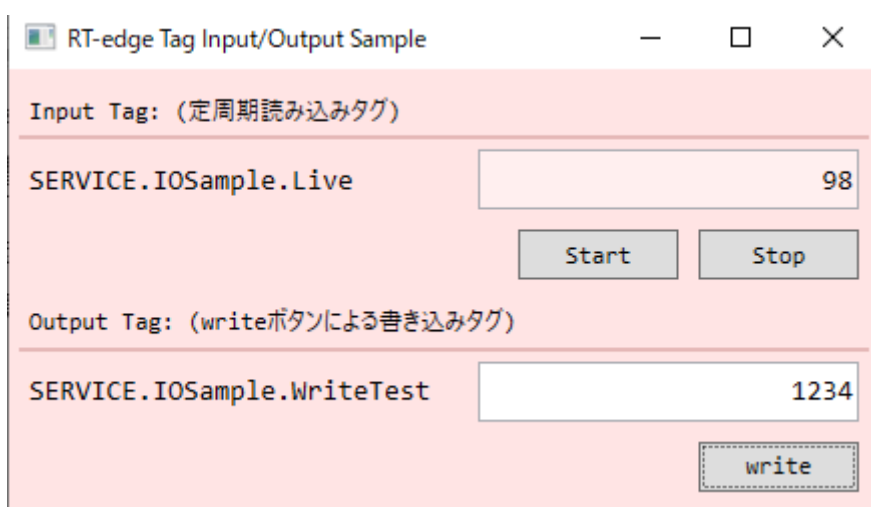


図 1. IOSample 画面

### 1.2. 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
<b>INtime</b>	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。 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。
<b>RTA</b>	<b>RealTime Application</b> : リアルタイムアプリケーションの略称。INtime 上で動作するローダブルプロセスの拡張子です。INtime 上で動作するローダブルアプリケーションは、RTA という拡張子を持ちます。
<b>RSL</b>	<b>Realtime Shared Library</b> : 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。
<b>API</b>	<b>Application Programming Interface</b> : 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。

用語	説明
<b>NTX</b>	INtime's Windows NT extension API: INtime 用 Windows NT 拡張 API の略称。NTX 関数は Windows プログラムが INtime リアルタイム環境上で実行するリアルタイムプログラムと通信を可能とする関数セットです。
<b>サービス/EgService</b>	RT-edge システムを構成する機能プロセス(rta/exe)です。
<b>タグ/EgTag</b>	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。
<b>リンクタグ</b>	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。
<b>データセット/EgDataset</b>	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。
<b>コレクタ/EgCollector</b>	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。
<b>メールボックス/EgMailBox</b>	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。
<b>メッセージ</b>	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。
<b>フレームワーク</b>	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定義 XML の記述により自動処理させることができます。
<b>サービスコンフィグファイル</b>	RT-edge フレームワークが、タグやメールボックスなどの RT-edge オブジェクトの生成やサービスの起動を自動処理するための定義を記述した XML 形式のファイルです。
<b>入力</b>	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。
<b>出力</b>	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。
<b>RTCD</b>	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。 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において利用される機能です。 RTCD は Windows アプリケーション間、INtime®アプリケーション間、Windows-INtime®アプリケーション間いずれの場合も利用可能です。
<b>RSI-ECAT-Master</b>	INtime 向け EtherCAT Master Stack です。 (本書では RSI-ECAT と省略する場合があります。)

### 1.3. 関連資料

名称	ファイル名	内容
ユーザーズマニュアル	DOCRTEEDGEUSR.pdf	RT-edge システム全般的な内容の説明が記載されています。
サービス作成マニュアル	DOCRTEEDGE_SRV.pdf	サービスの作成方法、及びサービス開発環境構築手順の説明が記載されています。

## 2. 処理内容

### 2.1. プログラム構成

IOSampe.exe は特定のタグから定周期で値を読み取るユーザーサービスです。  
以下のプログラム構成となっています。

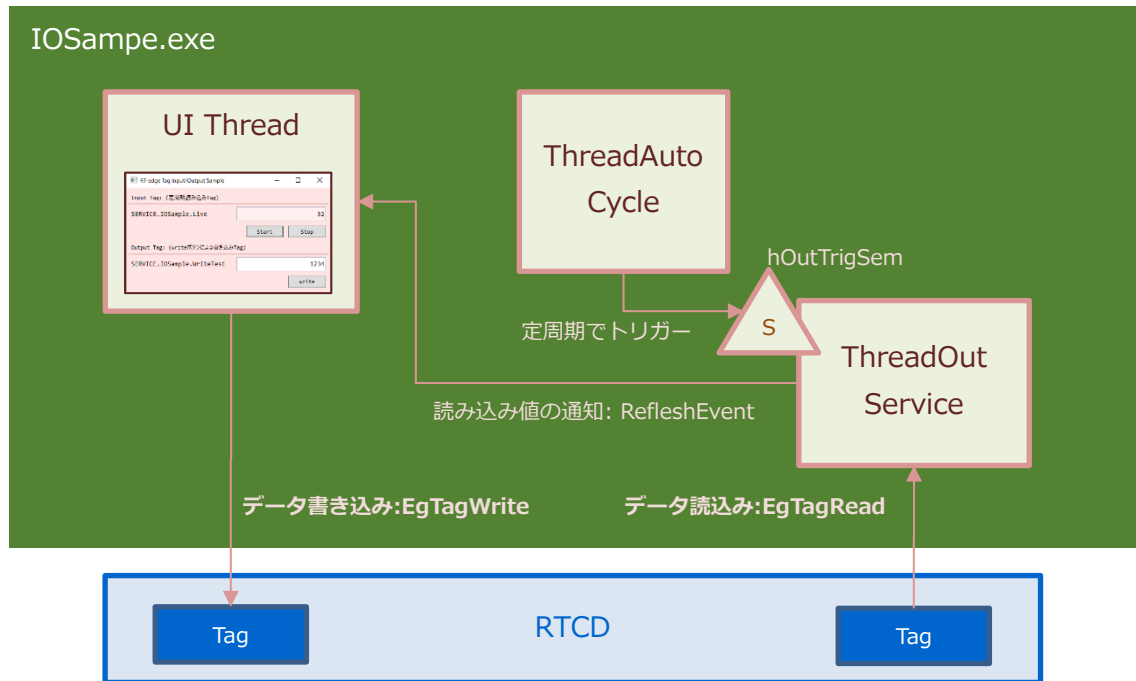


図 2. プログラム構成図

表 2. プログラム構成オブジェクト解説

名称	定義ファイル	内容
UI Thread	MainWindw.xaml/ MainWindow.xaml.cs	画面表示処理/ボタンクリックなどのイベントハンドリングを行います。 プログラムの最初に MainWindow クラスコンストラクタが実行され、各画面オブジェクトの初期化处理、RT-edge を使うための初期化处理を行っています。 また、ThreadOutService からの値更新イベント (RefreshEvent) を購読し、受信したときには、画面上に表示します。
ThreadAutoCycle	edgeCode.cs	ThreadOutService スレッドへ、処理実行のトリガーをかけるスレッドです。 SERVICE.IOSample.Cycle で設定された周期にてトリガーをかけます。
ThreadOutService	edgeCode.cs	RTCD からのデータ読み込み処理を実行します。 読込んだデータは値更新イベント (RefreshEvent) で UI Thread に通知します。
hOutTrigSem	edgeCode.cs	ThreadOutService スレッドが待機するセマフォです。 ThreadAutoCycle スレッドはこのセマフォを使って ThreadOutService スレッドに動作タイミングを通知します。

## 2.2. アクセスするタグ

デフォルトでは以下となっています。ソースコードを変更しビルドすることでアクセスするタグを変更できます。

表 3. デフォルトでアクセスするタグ一覧

名称	内容
SERVICE.IOSample.WriteTest	画面上で指定した値がかかれるタグです。 IOSample 起動時に生成します。
SERVICE.IOSample.Live	周期的に読み込むタグです。 IOSample 起動時に生成します。

## 2.3. IOSample サービスコンフィグファイル

IOSample.exe 用のサービスコンフィグファイル IOSampe.xml です。

表 4. IOSample サービスコンフィグファイル定義一覧

名称	デフォルト	内容
SERVICE.IOSample.Cycle	500	タグ読み込み周期です。 具体的には、ThreadAutoCycle の動作周期です。 単位は msec です。(Windows で生成する周期の為、リアルタイム性はありません)
SERVICE.IOSample.AutoRun	0	サービス起動時にタグリフレッシュ処理を自動開始するかの設定です。1 に設定すると IOSample 起動時にタグ読み込みが自動で開始されます。0 に設定すると、画面上の Start ボタンによって開始します。 1:自動開始 / 0:手動開始



IOSample を終了しても、IOSample が生成したタグ情報は残ります。そのため手動で再起動 (IOSample.exe ダブルクリックで起動) した場合、前回起動時のタグ値を読み取ります。

これは、例えば IOSample を終了後 Cycle プロパティを変更して再度手動で起動しても、Cycle プロパティへの変更は反映されず、前回の設定値になることを意味します。変更を反映させるには、RT-edge システム (EgBoot の再起動) の再起動してください。



## 3. 実行

### 3.1. 起動方法

- 1) IOSample プロジェクトの Release フォルダにある「IOSample.exe/IOSample.xml」を RT-edge インストールパスへコピーします。

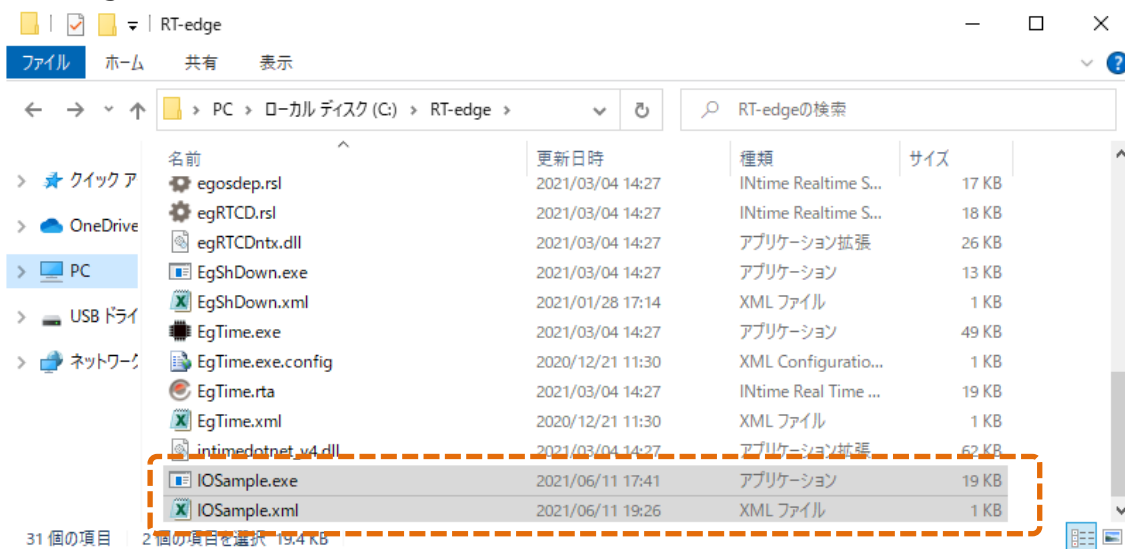


図 3. IOSample の配置

- 2) EgBoot.xml(サービスコンフィグファイル)の Services エlement に IOSample.exe を定義します。定義方法の詳細については、別紙ユーザーズマニュアルを参照ください。

```
<Services>
  <Service Name="EgBoot">
  </Service>
  <Service Name="EgLog" Path="EgLog.exe" Argument="DispNumMax=500">
  </Service>
  <Service Name="EgTime" Path="EgTime.exe">
  </Service>
  <Service Name="IOSample" Path="IOSample.exe">
  </Service>
</Services>
```

図 4. IOSample.exe の定義

- 3) EgBoot.exe を起動します。(すでに起動している場合は、一旦終了させたのち起動させます)
- 4) IOSample.exe が自動的に起動します。

### 3.2. 画面説明

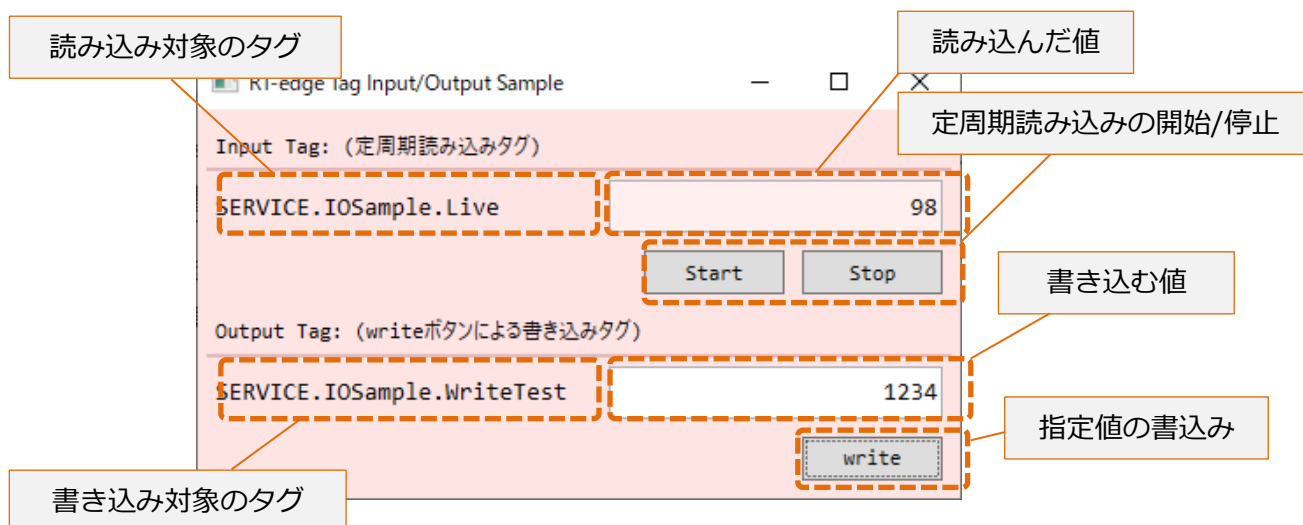


図 5. IOSample 画面説明

### 3.3. 確認方法

読み込み値、書き込み結果が正しいかを確認するには、EgBrow.exe を使います。EgBrow.exe は RT-edge インストールパスに配置されています。

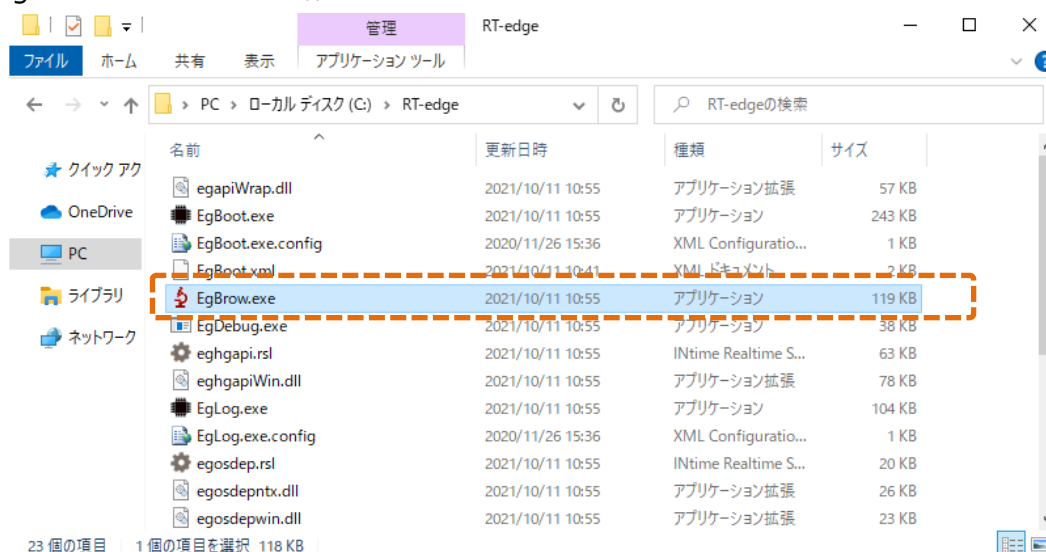


図 6. EgBrow.exe 配置場所

EgBrow.exe では RT-edge システムで生成されているタグの一覧をモニタリングできます。これを使い、IOSample でアクセスするタグの現在値を確認し、IOSample 上の表示と一致するかを確認します。

### ■ 読み込み値の確認

読み込んでいるタグの値が同じであるか確認します。

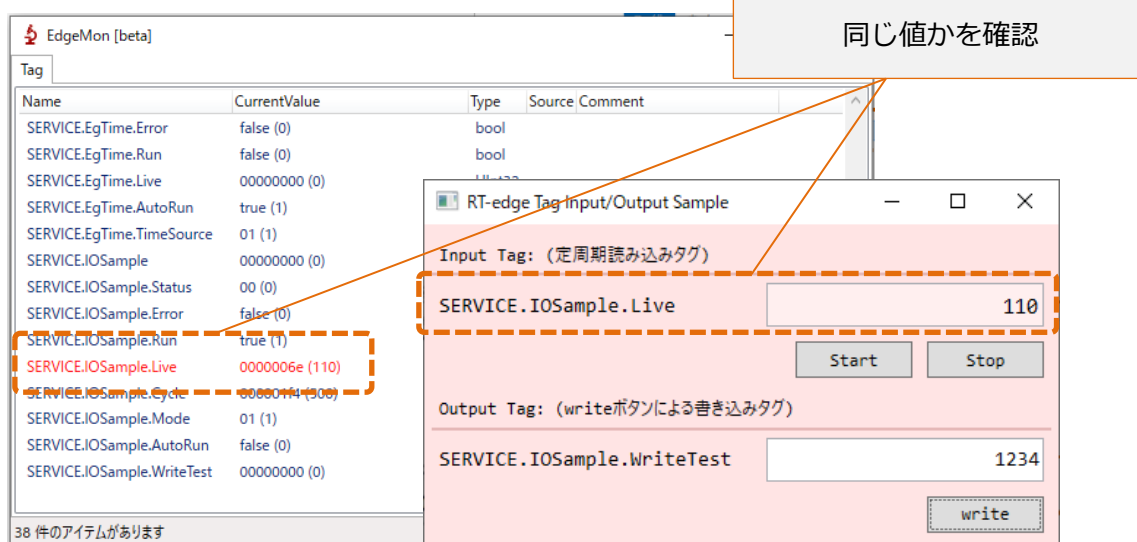


図 7. 読み込み値の確認

上記例では、IOSample は「SERVICE.IOSample.Live」を定周期で読み込んでいます。ある瞬間に値が 110 であったとき、EgBrow でも確認したとき同じく 110 となっています。問題なく読み込めていることが確認できます。

### ■ 書き込み値の確認

画面から指定した値が EgBrow 上の表示と同じであることを確認します。

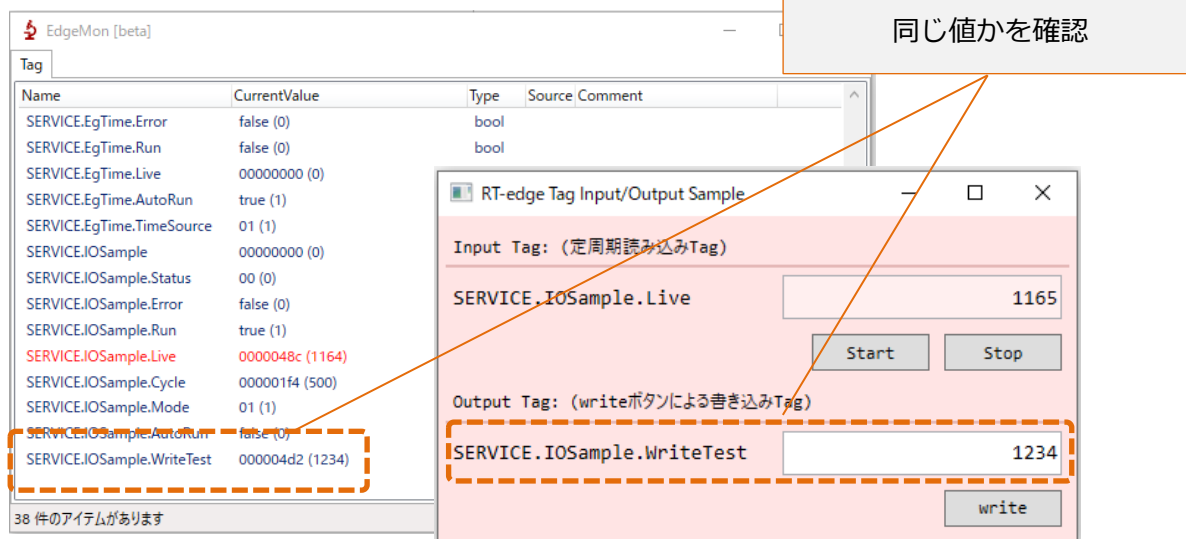


図 8. 書き込み値の確認

上記例では、IOSample は「SERVICE.IOSample.WriteTest」へ値 1234 を書いています。Write ボタンを押した後に、EgBrow 上での値が確認すると、1234 となっています。問題なく書き込めていることが確認できます。

### 3.4. 終了方法

画面右上の×ボタンで終了します。処理としては Closing イベントハンドラにてサービス終了処理を実行します。

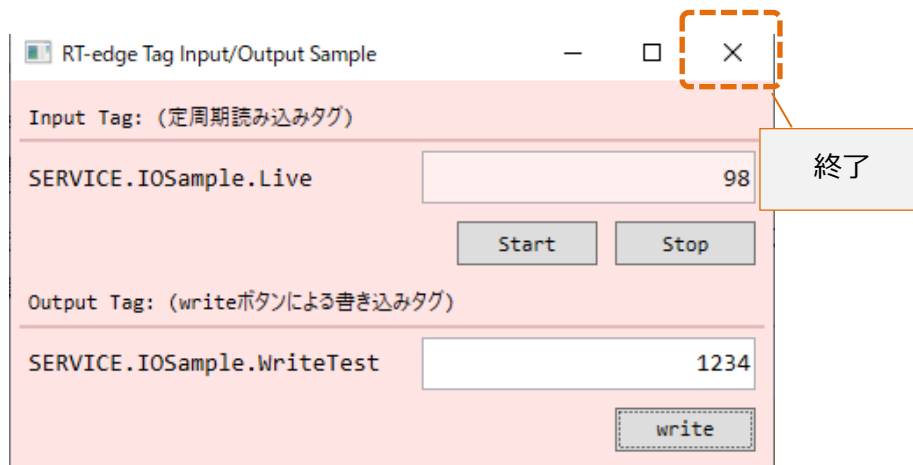


図 9. IOSample 終了



IOSample を終了しても、IOSample が生成したタグ情報は残ります。そのため手動で再起動（IOSample.exe ダブルクリックで起動）した場合、前回起動時のタグ値を読み取ります。

これは、例えば IOSample を終了後 Cycle プロパティを変更して再度手動で起動しても、Cycle プロパティへの変更は反映されず、前回の設定値になることを意味します。変更を反映させるには、RT-edge システム（EgBoot の再起動）の再起動してください。

## 4. アクセスタグの変更方法

### 4.1. 変更場所

アクセスするタグを変更するには MainWindow.xaml.cs の定義を変更します。

EgBrow で表示されるタグは全て定義できます。

表 5. アクセスタグ定義

ファイル名	行数	変数名	デフォルト値
MainWindow.xaml.cs	40 行目	TAG_INPUT_VALUE	SERVICE.IOSample.Live
	45 行目	TAG_OUTPUT_VALUE	SERVICE.IOSample.WriteTest

```

36
37
38
39
40
41
42
43
44
45

```

```

/// <summary>
/// 周期的に読み込むTag
/// </summary>
private static string TAG_INPUT_VALUE = "SERVICE.IOSample.Live";

/// <summary>
/// Writeボタンで書き込むTag
/// </summary>
private static string TAG_OUTPUT_VALUE = "SERVICE.IOSample.WriteTest";

```

図 10. アクセスタグ定義位置

### 4.2. 変更方法

変更には IOSample.exe のリビルドが必要になります。そのため事前に、開発環境設定が行われているか確認してください。具体的な設定はサービス作成マニュアル内の開発環境設定の章を参照ください。

- 1) EgBrow を起動し、生成済みのタグの一覧を表示させます。
- 2) IOSample からアクセスさせるタグを、一覧から探し名称を控えておきます。(この後の手順で IOSample.exe ソースコード内に定義します。)

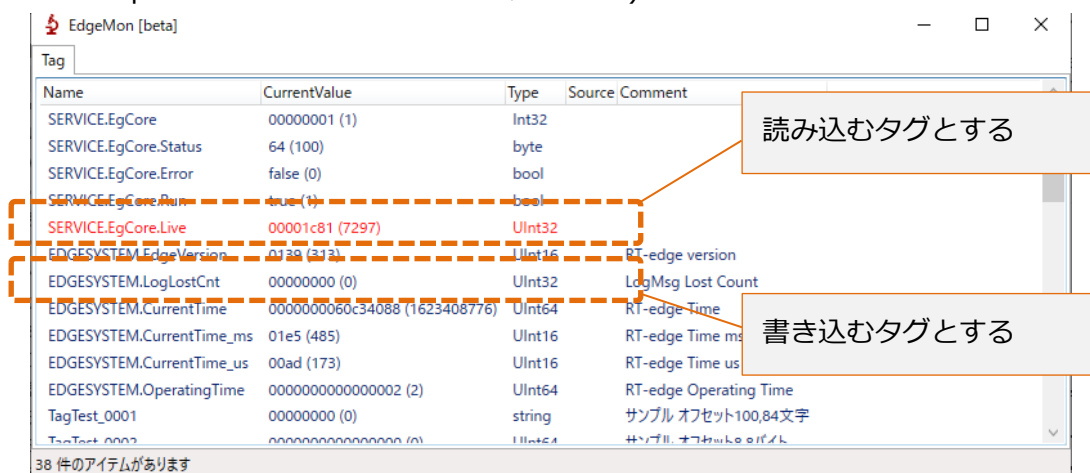


図 11. 変更するタグの選択

例) 読み込みタグ: SERVICE.EgBoot.Live / 書き込みタグ: EDGESYSTEM.LogLostCnt

- 3) IOSample プロジェクトを Visual Studio で開きます。
- 4) ソリューションエクスプローラー > 参照設定 を開き、egapiWrap.dll を参照設定に追加します。



IOSample プロジェクトの参照設定は、デフォルトで RT-edge が C ドライブ直下にインストールされているとして参照設定がされています。

そのため、C ドライブ直下にインストールしている場合は、egapiWrap.dll への参照は不要となります（設定済みなので）

- 5) MainWindow.xaml.cs を開きます。
- 6) TAG\_INPUT\_VALUE に定周期で読み込みたいタグを定義します。
- 7) TAG\_OUTPUT\_VALUE に書き込みたいタグを定義します。

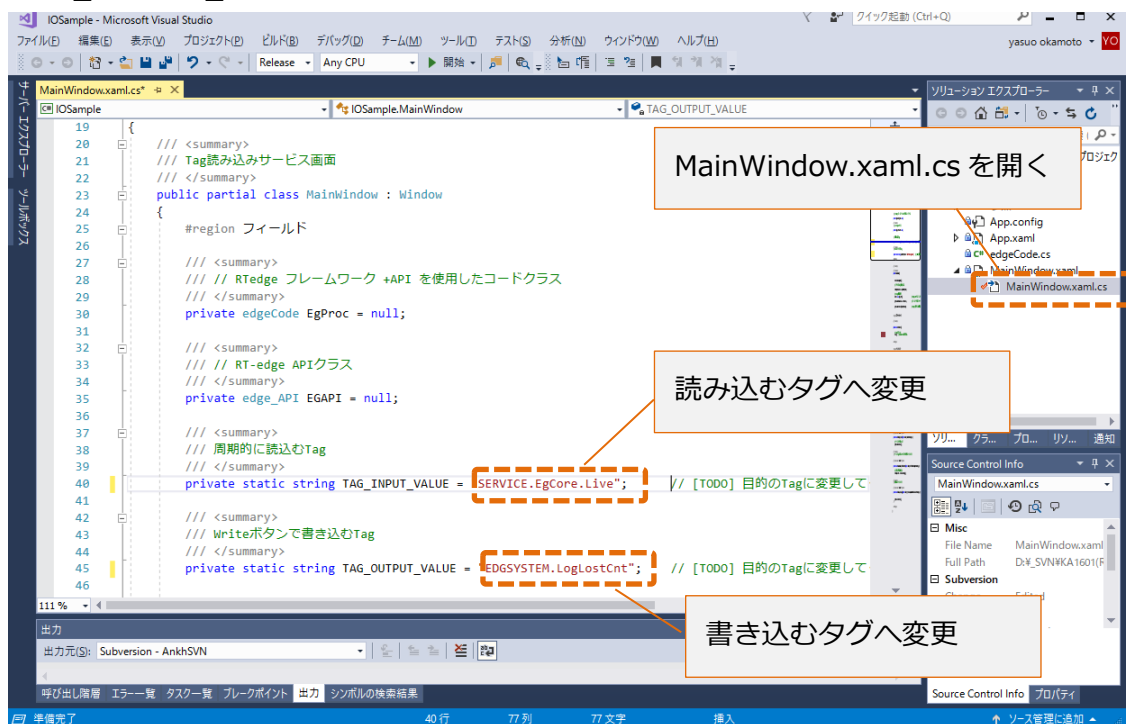


図 12. アクセスするタグへ変更

- 8) IOSample プロジェクトをビルドします。
- 9) IOSampe.exe が起動している場合は終了させます。同時に EgBoot.exe も終了させます。
- 10) 生成された IOSample.exe を RT-edge インストールパスへコピーします。
- 11) EgBoot.exe を起動します。
- 12) 画面を操作し、目的のタグにアクセスできるかを確認します。

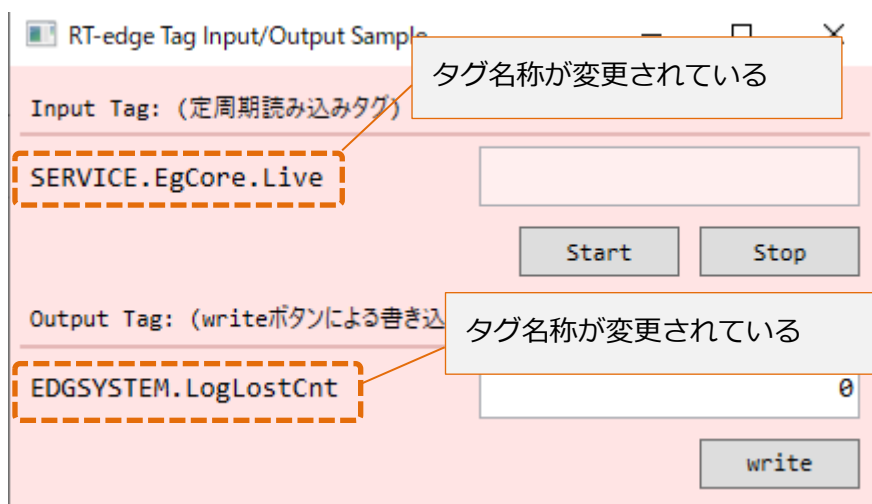


図 13. 変更後の画面



VisualStudio からデバッグ実行する場合は、Debug フォルダに以下のファイルを配置してください。配置していない場合、**EgInit()コールで失敗します。**これらのファイルは RT-edge インストールフォルダ内に格納されています。

**eghgapiWin.dll / egosdepntx.dll / egRTCDntx.dll**

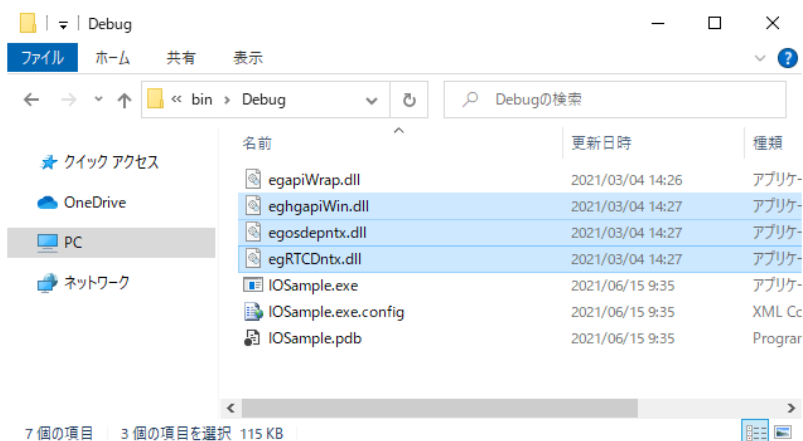


図 14. デバッグ実行時の DLL 配置





## 更新履歴

版	日付	更新説明
1	2021.6	初回版

REALTIME SERVICE for Windows

# RT-edge IOSample プログラム取扱説明書

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

- ・本書の著作権は、マイクロネットに帰属します。
- ・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が一ご不審な点や記載もれなどお気づきな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。
- ・EtherCAT<sup>®</sup>は、ドイツ Beckhoff Automation GmbH の登録商標であり、特許で保護されている技術です。
- ・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。