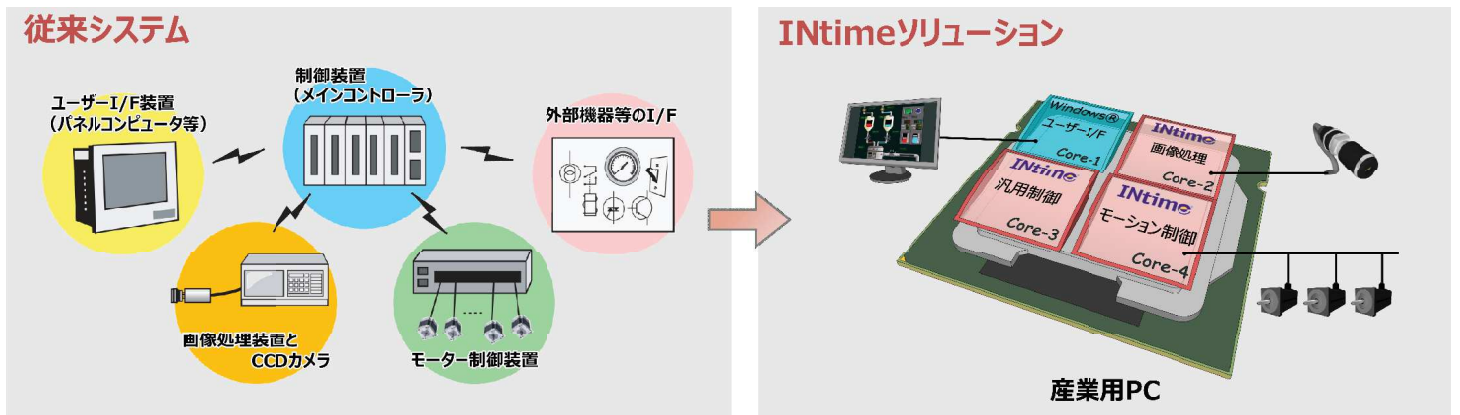


# INtime®

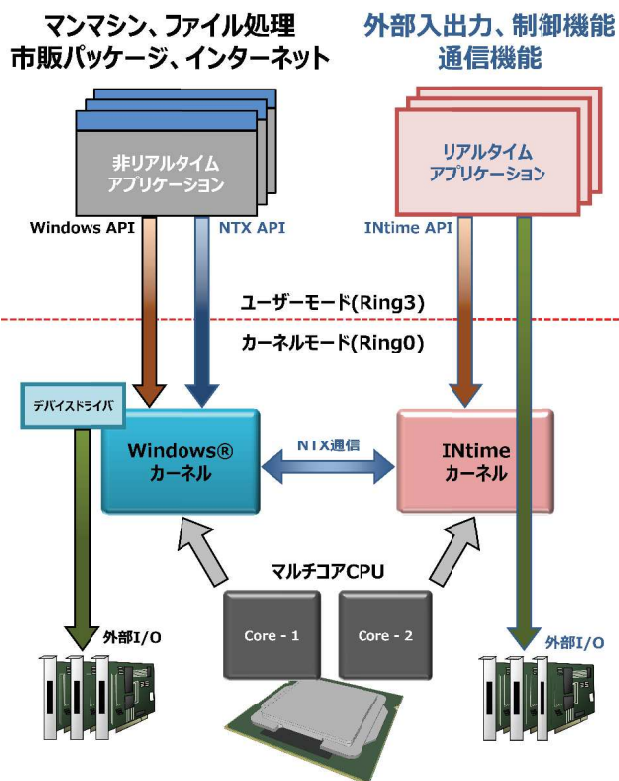
近年、産業用機器の制御装置に工業用PCが広く使われています。PC技術はグローバルスタンダードな技術で、これを用いた製品の製造やシステム導入に伴う人・物・予算・時間等の総合的なコスト削減が可能です。さらに製品およびサービスの品質安定が期待でき、技術的進歩にも容易に対応できます。

INtimeはPCを産業用制御に利用するためにリアルタイム性と信頼性を提供するリアルタイムOSです。Windows® operating systemと共存/協調して動作するので、ソフトウェア・ハードウェア両面で使いやすく、また将来的にも安心してお使いいただけます。

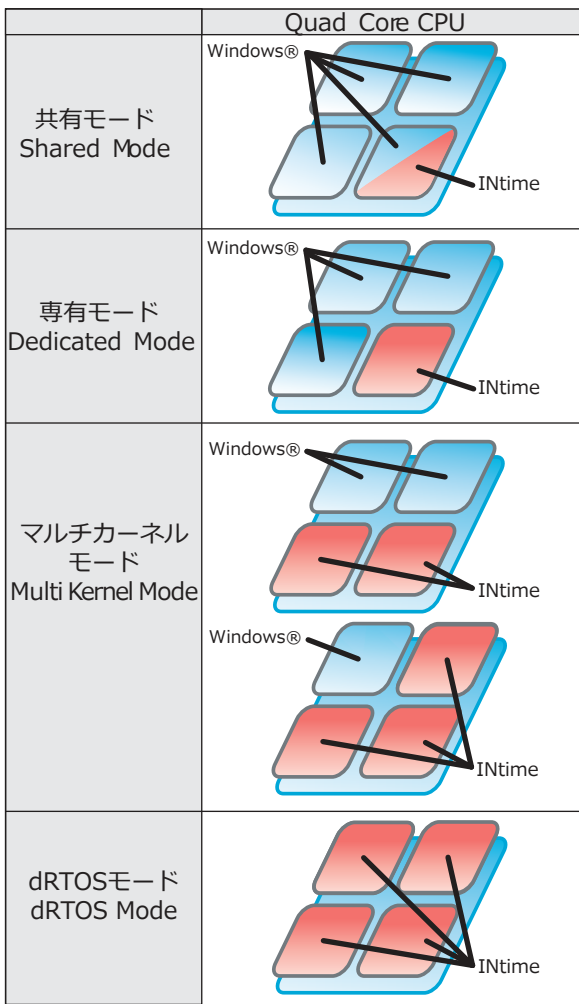


**モーション制御・機器制御・画像処理・Windows®の機能が、一台のPCハードウェア上で実現します。**

## INtimeの特長



- Windows®と協調動作
- Windows® Embedded対応
- 豊富なリアルタイムスケジューリング
  - ・256段階のプライオリティレベル
  - ・アプリケーションプログラムとしての割り込みハンドラー
  - ・IRQ / MSI対応
  - ・マルチプロセス、マルチスレッド機能
  - ・多彩なシステムオブジェクト (メールボックス、セマフォ、アラーム、リージョン、共有メモリ、メッセージキュー)
- マルチコアCPUでは1つのシステム上で複数のリアルタイムカーネル並列動作を実現
- Windows®なしでも動作可能
- Microsoft® Visual Studioの開発環境と統合 (Visual C/C++)
- ユーザーモードによる完全メモリ保護、アドレス分離
- ソースレベルのダイナミックデバッガをMicrosoft® Visual Studioに統合
  - ・OSを止めずにデバッグ可能
  - ・Windows® とINtimeのアプリケーションを同期デバッグ
- リアルタイムTCP/IPドライバ/ライブラリ標準装備



## INtimeの動作モード

INtimeシステムでは、PCプラットフォーム上に、ユーザーのニーズにあわせ様々な動作モードを実現することが可能です。

### 共有モード：

1つのCPU上でWindows®とINtimeリアルタイムカーネルを同時に動作させるモードです。Intel CPUのハードウェアマルチタスク機能を利用してWindows®とINtimeの切り替えを行なっているため、シングルコアCPUのPCにおいてもWindows®とINtimeの共存が実現可能です。

### 専有モード：

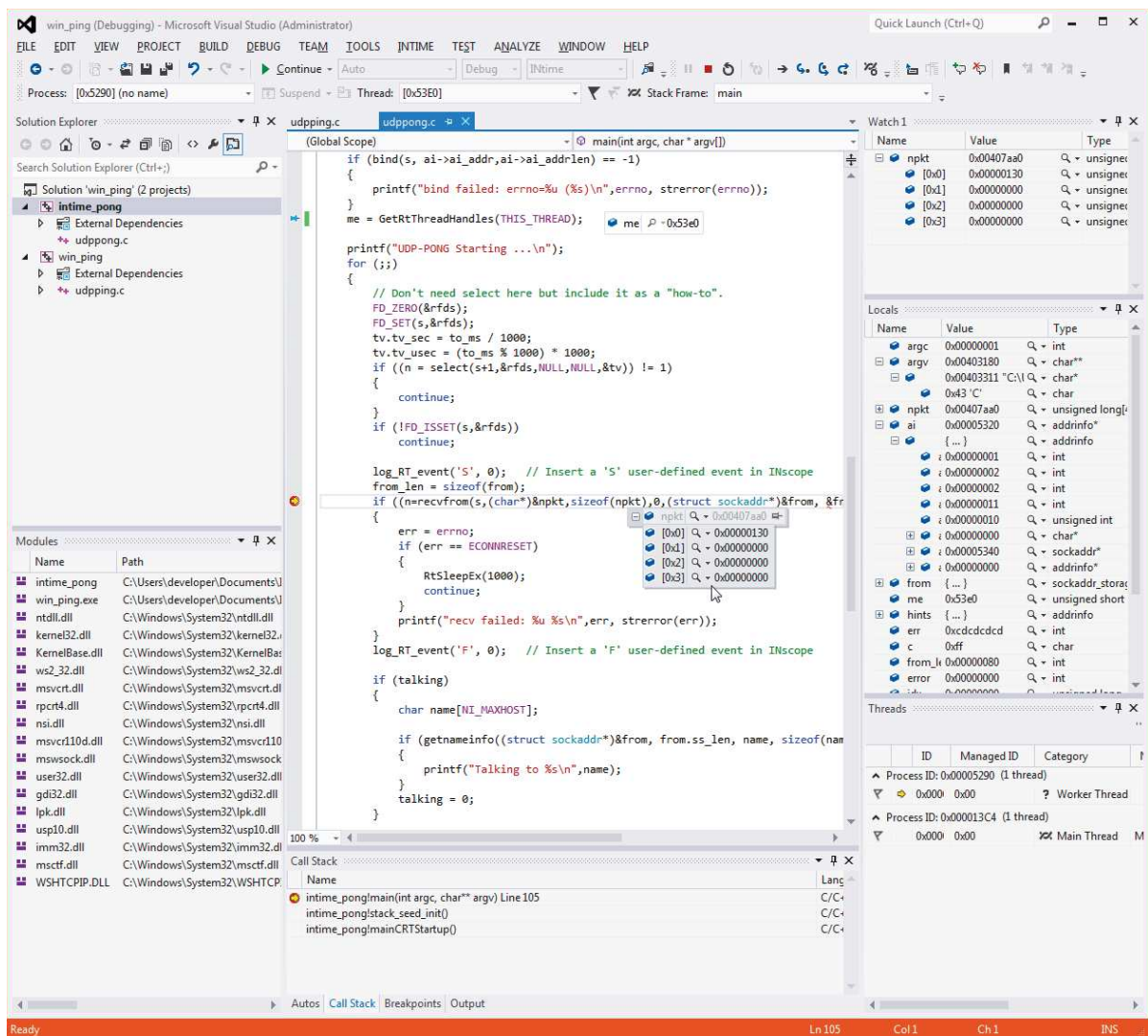
物理コアを2つ以上実装しているCPUでは、コア一つを丸ごとINtimeリアルタイムカーネルに割り当てる専有モードを実現することが可能です。共有モードと比較するとOS切り替えが発生しないため、応答性能が向上します。

### マルチカーネルモード：

クワッドコア以上のCPUの場合は、1つのシステム上に2つ以上のリアルタイムカーネルを動作させることができます。それぞれのリアルタイムカーネルは、非同期に動作します。このモードを利用することで、処理負荷の高いアプリケーションを独立したカーネルへと移すなどの負荷分散を実現できます。

### dRTOSモード：

お客様のニーズに合わせて、Windows®が無い状態のスタンドアロンでの動作環境も構築できます。これによりリアルタイム制御だけを行うLowEndシステムも構成可能です。また、Windows®の有無を問わず、ソフトウェアの資産を共通してご利用いただくことができますので、ソフトウェアの開発効率が向上します。



## ライブラリサポート

### CとC++のフルサポート

INtimeのC99互換CライブラリとANSI C++ライブラリはSTLのためのフルサポートを含んでいます。INtime RtClassライブラリは標準リアルタイム・オブジェクトにC++インターフェースを供給します。

### Shared real-time libraries(RSL)

INtime Real-time Shared Libraries(RSL)は、WindowsのDLLのような存在となります。RSLは、複数のリアルタイムプロセスに一つのリアルタイムコードライブラリを共有する手段を提供します。

### SSE Library

Intel® Integrated Performance Primitive Static Libraries (Intel® IPP)を利用することで、開発者はINtimeのリアルタイムプラットフォーム上での演算スピードを向上できます。

## I/Oサポート

INtimeアプリケーションのために、I/Oハードウェアに対してのダイレクトなアクセス手段が用意されています。I/Oハードウェアアクセスのための困難なカーネルデバッグ技術を習得する必要も固有のデバイスドライバを作成する必要もありません。

### PCI&PCI Express

INtime APIには、PCIまたはPCI Expressデバイスを検出して初期化を行う関数実装されています。またINtimeアプリケーションがハードウェアにアクセスを行うサンプルコードおよびテンプレートも用意されています。

### Real-time TCP/IPネットワークアクセス

INtimeはWindowsに依存しない、INtime上で駆動するネットワークドライバが付属しています。INtimeアプリケーションはソケットAPIを使用することで、このネットワークドライバを利用して、外部機器との通信を行うことができます。INtimeのTCP/IPドライバは、リアルタイムスレッドとして稼動するため、WindowsのTCP/IP (WinSock)通信と比較して、性能の安定化が望め、信頼性面でも向上が見込めます。

## USBアクセスサポート

USB3.0ホストコントローラーインターフェースのReal-time stackサポートが実装されています。このスタックを利用することで、INtimeアプリケーションはUSB機器に対してダイレクトアクセスを行います。

## 各種フィールドバス対応

INtimeでは各種フィールドバスのドライバソフトウェアを幅広く対応しています。

EtherCAT、CC-Link、CC-Link IE、FL-net、MECHATROLINKなど多数のドライバが、マイクロネットやサードパーティによって提供されます。

## デバッグおよびランタイムサポート

### 充実のデバッグ機能

INtimeはMicrosoft® Visual Studio開発環境と完全統合され、コーディング、ビルドに加えて、INtimeアプリケーションのデバッグ実行が可能です。

また、シリアル経由のデバッガとしてSDM (System Debugging Monitor) を標準装備し、外部シリアル端末にハードウェア例外トラップ情報を出力して、現在のシステム状態を参照したり、端末からの入力によりデバッグ作業を進めることも可能です。

この他、INtimeオブジェクトの状態監視オブジェクトブラウザINtime Explorer、カーネル上で実行されるスレッドのタイムチャートを収集するINscopeなどのデバッグツールが標準実装されております。

### 例外ハンドリングと障害管理

INtimeはデバッグ効率を改善すると共にランタイムに信頼性を提供するため、幾つかのレベルの例外処理および障害管理機能を実装しています。例外がアプリケーションによって構造化した例外処理あるいはC++例外処理の使用によって扱われない場合、設定されたデバッグ方法で処理されます。

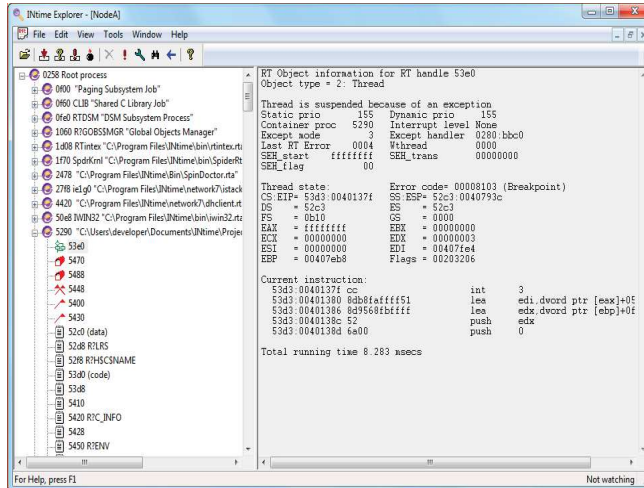
もっとも低レベルの場合は、例外が発生したスレッドを単純にサスペンドし、他のシステムの実行の継続を確保します。

### 統合開発環境

Microsoft® Visual Studio  
INtimeアプリケーションのための統合開発環境はMicrosoft® Visual Studioを利用します。

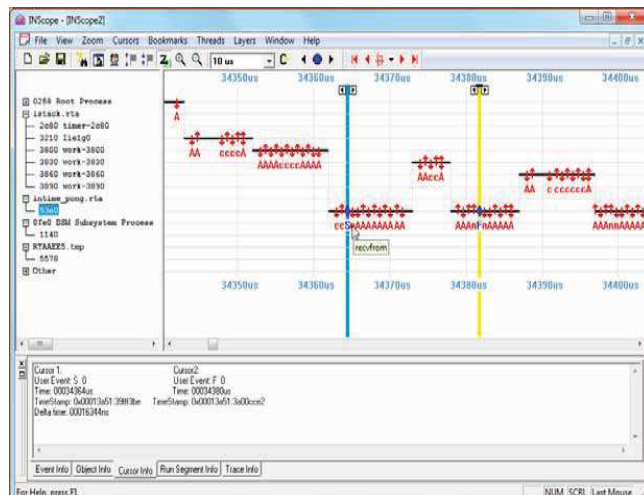
Microsoft® Visual Studioのパワフルな開発機能は勿論、デバッグ機能もそのまま利用することができるため、Real-timeプロセスや変数のモニタリングは勿論、Breakpointを用いたステップ実行も可能です。

さらに、ページ例外、スタック例外、バッドポインタアクセスやゼロ除算などのリアルタイムプロセスで発生したエラーの位置をMicrosoft® Visual Studio上で指し示しますので、デバッグ作業を非常に効率的に行うことが可能です。



### ダイナミックオブジェクトブラウザ INtime Explorer

INtime Explorerツールは、INtimeの管理するプロセス、スレッド、メールボックス、セマフォ、メモリ他の全てのReal-timeオブジェクトの情報をユーザーに提供します。ユーザーはこれらの情報をダイナミックに参照し、管理することができます。また、アプリケーションエラーのレポート機能も持っています。



### Real-timeシステムアナライザ INscope

INscopeは、INtimeのReal-timeパフォーマンスのアナライザツールです。Real-timeスレッドのシーケンスデータを正確な時間で計測します。このツールを使用することで、Real-timeスレッドの処理時間、スレッド切り替え状況、APIコール履歴などを分析することができます。

## INtime仕様

プライオリティスケジューリング	(高) 0~255 (低) 256段階
カーネルティック	50us, 100us, 125us, 200us, 250us, 500us, 1ms, 2ms, 5ms, 10ms
ラウンドロビンスケジューリング	適用プライオリティ閾値変動可 20ms ~ 100ms (10ms単位)
メールボックス	FIFO式/優先度式
セマフォ	FIFO式/優先度式
利用可能メモリサイズ	システムトータルで2GB
割り込み	ハンドラ/スレッド構造 IRQ/MSI対応
開発言語	C/C++
リアルタイムデバッグ	Visual Studio Integrated Debugger Spider Debugger SDM(System Debug Monitor)
標準I/O	TCP/IP, UDP/IP, USB, RS232C
最大オブジェクト数	8190個 (システムを含む)

## INtime API (主要抜粋)

<b>割り込み管理</b>
SignalEndOfRtInterrupt, SetRtInterruptHandler, ResetRtInterruptHandler, SignalRtInterruptThread, WaitForRtInterrupt, DisableRtInterrupt, EnableRtInterrupt
<b>メールボックス管理</b>
CreateRtMailbox *, DeleteRtMailbox *, SendRtHandle *, ReceiveRtHandle *, SendRtData *, ReceiveRtData *
<b>メモリ管理</b>
AllocateRtMemory, FreeRtMemory, CreateRtMemoryHandle, DeleteRtMemoryHandle, MapRtShareMemory *, MapRtPhysicalMemory, GetRtPhysicalAddress, GetRtSize *
<b>オブジェクト管理</b>
CatalogRtHandle *, LookupRtHandle *, UncatalogRtHandle *, InspectRtProcessDirectory, GetRtHandleType *, GetRtHandleTypeEx
<b>プロセス管理</b>
CreateRtProcess *, ExitRtProcess *, RtNotifyEvent, RegisterRtDependency, UnregisterRtDependency, RegisterRtSponsor, UnregisterRtSponsor
<b>スケジューラ管理</b>
knRtSleep, knStartRtScheduler, knStopRtScheduler
<b>セマフォ管理</b>
CreateRtSemaphore *, DeleteRtSemaphore *, WaitForRtSemaphore *, ReleaseRtSemaphore *
<b>ステータス管理</b>
GetLastRtError *, SetLastRtError, CopyRtSystemInfo, ReportRtEvent
<b>スレッド管理</b>
CreateRtThread, DeleteRtThread, GetRtThreadPriority, GetRtThreadHandles, SetRtThreadPriority, SetRtProcessMaxPriority, RtSleep, getRtThreadAccounting, SuspendRtThread, ResumeRtThread, GetRtThreadInfo
<b>システムデータ管理</b>
ntxGetLocationByName, ntxGetFirstLocotion, ntxGetNextLocation, ntxGetNameOfLocation

(\* ) 対応するNTX APIがあります。

## 充実のサポート体制

### INtime製品メンテナンスサービス

INtime製品の開発および保守をサポートするメンテナンス製品です。INtime-DK購入後1年間付属しており、本サービス期間中は、INtime-DKのバージョンアップを自由に行うことのできる「製品更新サービス」と弊社技術スタッフによる「技術サポートサービス」を受けることができます。

### オンサイトサポートサービス

通常の技術サポートの他、お客様のニーズに応じてマイクロネット技術者が直接お客様の元へとサポートに伺うオンサイトサービスもご用意しております。日本国内に留まらず、海外への派遣にも対応しております。

### INtimeセミナーサービス

INtime導入後のお客様向けに、INtimeセミナーサービスをご用意しております。弊社のINtime技術者が直接お客様の下で行う出張セミナーから、高度ポリテクセンターにて実施される定期セミナーなど、ニーズに合わせてご利用いただけます。

### プリインストールPC販売

マイクロネットでは工業用として実績のあるPCにINtime (およびWindows®) をプリインストールして販売しています。マイクロネットがINtimeの適合性を動作確認しています。

### 受託開発サービス

お客様の開発業務の一部を、受託開発としてお引き受けするサービスを行っています。INtime用のドライバ開発からシステムそのものの開発まで、幅広くお引き受けいたします。

## 動作適合ハードウェア

プラットフォーム	PC/AT互換機
CPU	Intel x86/x64 PCアーキテクチャ
メモリ	64MB以上
HDD	< 必要容量 > SDK 約 200MB 程度 RT 約 90MB 程度 < フォーマット > FAT32/NTFS
適合Windows®	Windows® Server 2003, R2 SP2以降 Windows® Vista SP1以降 Windows® Server 2008、およびServer 2008 R2 Windows® 7、Windows® 8、Windows® 8.1 Windows® 10、Windows® Embedded対応

## 製品体系

### 開発キット

INtime-DK INtime開発ライセンス及び開発キット

### 組込みライセンス

INtime-RT INtime組込みライセンス  
INtime-MCRT INtime組込みライセンス(マルチカーネル対応)  
RTOS-RT INtime組込みライセンス(dRTOS)  
RTOS-MCRT INtime組込みライセンス(dRTOSマルチカーネル対応)

### 製品メンテナンスサービス

INtime-MAINT 技術サポート及びバージョンアップ

INtimeはTenAsys社の登録商標です。  
その他に記載されているすべての製品名は、各社の商標または登録商標です。

このカタログ内の仕様は予告なしに変更することがあります。

### 【開発元】

**tenAsys**<sup>®</sup>  
Real-time Virtualization Experts  
http://www.tenasys.com

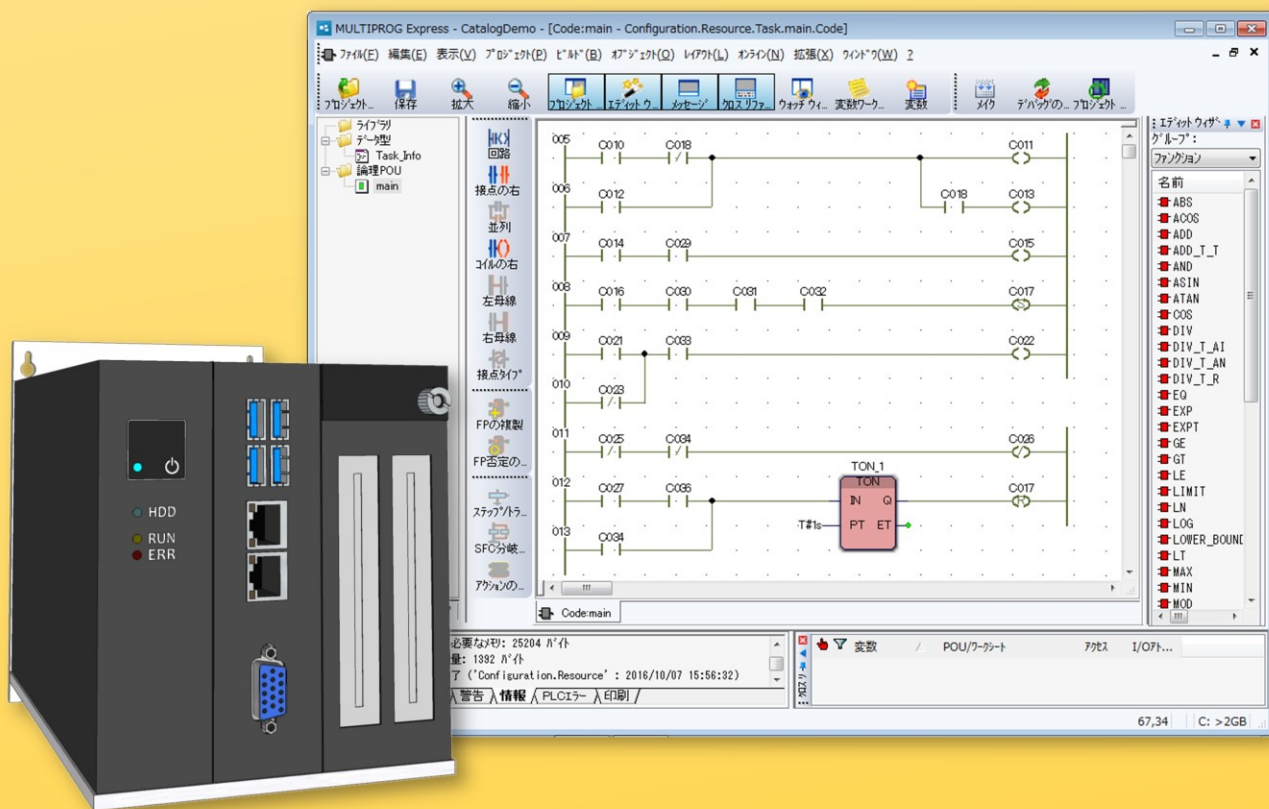
### 【代理店】

**マイクロネット**  
✉ cde@mnc.co.jp  
☎ 03-6909-3371  
URL : http://www.mnc.co.jp  
〒104-0032 東京都中央区八丁堀3-17-13 TMYビル 9F

## リアルタイムソフトウェアPLC



### Windows PCで高性能PLC機能を実現



- ✓ 制御周期は最速 **0.1ミリ秒**
- ✓ 1台のPC上でWindowsとPLCが **並列動作**
- ✓ EtherCAT®他多数の **フィールドバス** に対応
- ✓ 制御と同時にデータ採取も行う **IoT対応 PLC**

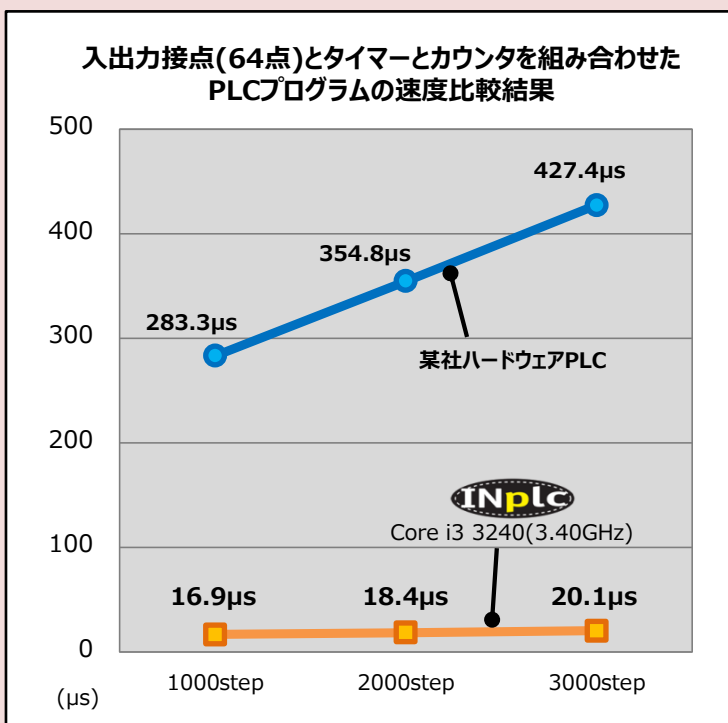
## ソフトウェアPLCによる装置の高速・高精度化

### ● 最速0.1ミリ秒の制御周期を実現

リアルタイムソフトウェアPLC「INplc」はリアルタイムOS「INtime」上で動作します。これにより従来のPLCを超える0.1ミリ秒の制御周期を実現しました。

### ● PC用CPUによる高速パフォーマンス

高性能のPC用CPUによって、一般的なハードウェアPLCよりも高速パフォーマンスを発揮します。



## Windowsアプリケーションとの連携

### ● WindowsとPLC制御の並列動作

INplcは1台のPC上でWindowsと並列動作します。PLC制御で使用するプログラム領域・メモリ領域・ハードウェアはWindows管理下から独立しているため、Windows制御の影響を受けません。

高負荷のWindowsアプリケーションが実行されたとしても、0.1ミリ秒の制御周期を確実に実現します。

### ● Windowsがフリーズしても動作

ブルースクリーンなどによりWindowsが停止した場合においても、PLC機能はその影響を受けることなく処理を継続します。

### ● PLC機能とWindowsのデータ連携

INplcとWindowsアプリケーションのデータ通信をサポートするWindowsコンポーネント「INpMac」を提供します。データの受け渡しは共有メモリを利用したメモリ間通信のため高速です。

また、OPC通信もサポートしています。(OPCサーバーをオプション提供)

## 豊富な外部インターフェース

PLC機能を活用するための外部インターフェースに広く対応しています。

表1 外部インターフェース

種別	対応状況
PC標準IO	RS232C USB Ethernet*
PCI / PCI Express 拡張ボード	デジタル入出力 アナログ入出力 RS232C / 422A / 485
ネットワーク	TCP/IP UDP/IP FL-net
フィールドバス	EtherCAT® CC-Link CC-Link IE Field MECHATROLINK-III PROFINET EtherNet/IP
モーション制御	CiA402対応 RT-Motion (マイクロネット製品)

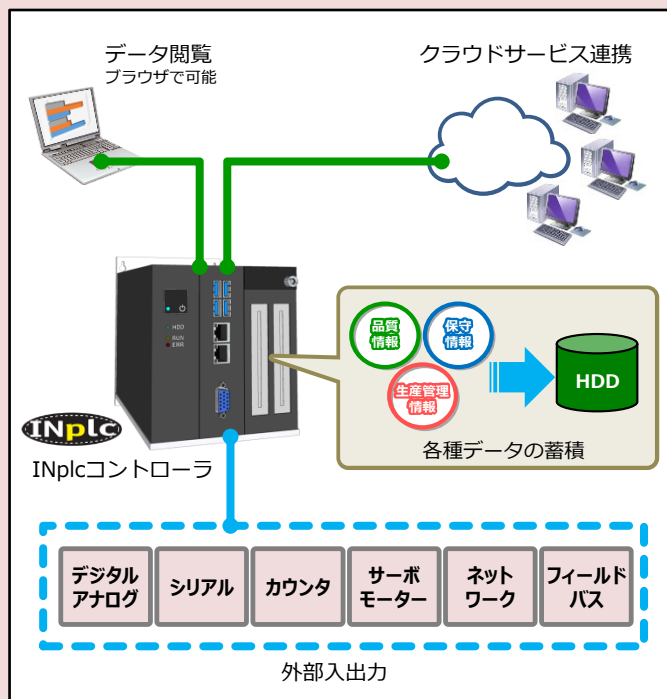
※ 対応するネットワークコントローラは以下の通りです。

- ・ Intel Gigabit ethernet controllerシリーズ, Intel Pro/100
- ・ Realtek Gigabit ethernet controllerシリーズ, Realtek 100Mbps controller

## トレーサブル機能によるIoT対応

INplcでは標準機能として制御データのトレース機能が実装され、I/O情報やファンクションの実行状況などの制御データがPC上にロギングされます。

ロギングされたデータは、Webサーバー機能によって他のPCでブラウザ閲覧したり、クラウドサービスと連携してデータ分析に活用することができます。



# 1台のPC上に機能統合

## 現状

機能毎に専用の制御装置を用意

Windows処理  
マイコン  
モーション制御  
PLC制御

PCやPLC、マイコンなどの機器毎に機器独自の技術やツールが必要...

機器(機能)間の通信に時間がかかり、タクトタイムを短縮できない...

機能毎に専用ハードウェアが必要で、ハードウェアコストが多くかかる...

## 導入後

### 各機能をINplcコントローラに集約

Windows処理  
C言語処理  
モーション制御  
PLC処理 INplc

INplcは国際標準の技術・ツールによって開発およびメンテナンスを行うため、技術の標準化を実現！

共有メモリを利用したメモリ間通信によって機能間の通信速度が格段に向上し、タクトタイムを大幅に短縮！

ハードウェアを1台のPCに統合することで、ハードウェアコストを大幅に削減！

# データトレース機能詳細

INplcはファンクションブロックコールの挙動やI/Oのインターフェース情報、システムイベント情報、タスクの状態情報などを制御データとして蓄積する「データトレース機能」を実装しています。蓄積されたデータは「生産管理情報や製造ロット毎の品質情報、アプリケーションの保守情報などとして自由なデータフォーマットで活用できます。



## ● トレーサブル機能 (主要抜粋)

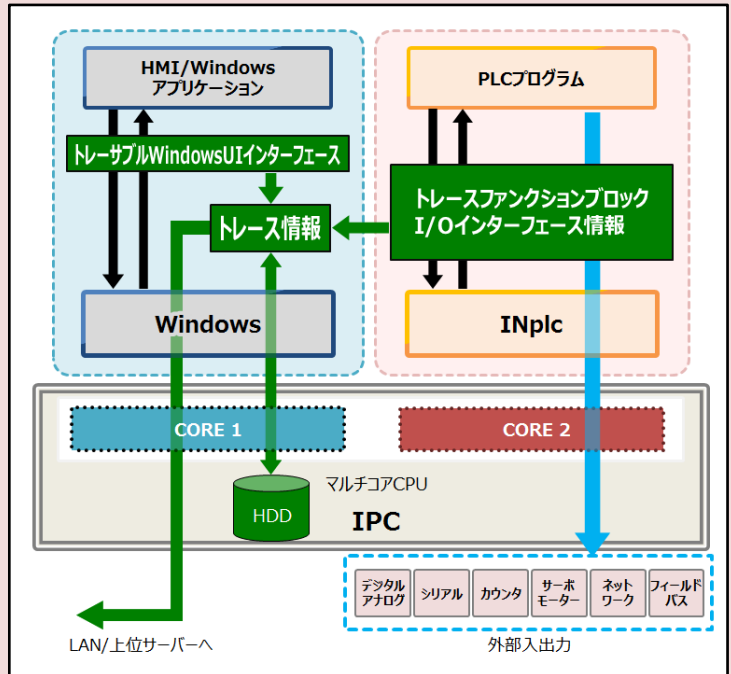
I/Oインターフェース情報	デジタル入出力、アナログ入出力、シリアル通信、制御ネットワーク、フィールドバス
システムイベント情報	PLCプログラムスタート、PLCプログラム停止、ウォッチドッグ発生、ゼロ除算発生、スタックオーバーフロー発生、I/Oドライバエラー、浮動小数点エラー、CPUオーバーロード発生、配列・構造体の範囲外、プログラミング内部エラー、ブルースクリーン
タスク状態	最大処理時間、最小処理時間、タスク実行回数、システム イベント発生時
その他ファンクション	EtherCATファンクション、シリアル通信ファンクション、RTOSファンクション、Windowsファンクション、データベースファンクション

## ● トレース情報の例

```

13:10:10.238,200,3ee0,3ee8,ECAT_FindSlave(i),0x3eec,100
13:10:10.339,200,3ee0,3ee8,RS_SEND(i),COM1,msg="GET",len=3
13:10:10.340,200,3ee0,3ee8,RS_SEND(o),E_OK
13:10:10.341,200,3ee0,3ee8,RS_RECV(i),COM1,0x410050,len=10
13:10:10.346,200,3ee0,3ee8,RS_RECV(o),msg="132.5",len=5,E_OK
13:10:10.348,200,3ee0,3ee8,SendSemaphore(i),3f00,unit=1
13:10:10.349,200,3ee0,3ee8,SendSemaphore(o),ret=TRUE,E_OK
13:10:10.350,200,3ee0,3ee8,Marking,"温度異常障害"
    
```

## ● トレースデータフロー



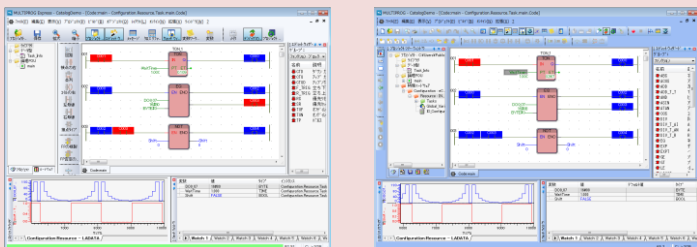
# PLCの国際標準プログラミング

## ● IEC 61131-3準拠の開発環境

INplc-SDKはPLCプログラムの開発キットです。

国際標準規格IEC61131-3に準拠した5言語でのプログラム開発が可能です。プログラム開発後は、コントローラと接続してダウンロードしてプログラムの配置・デバッグを行います。プログラム開発をサポートするロジックアナライザ機能やウォッチウィンドウ機能、レシピ機能などを実装しています。

また国際標準の開発環境のため、日本語はもちろん、英語、中国語、ドイツ語、フランス語など多言語にも対応しています。



※ 通常の開発キットの他に、機能拡張した開発キットも用意しています。

### 国際標準規格IEC 61131-3とは？

国際電気標準会議(IEC)が1993年に発行したPLC用のプログラム言語を定義した標準規格です。欧米やアジアで利用されるPLCコントローラの多くに採用されています。

## ● 豊富な制御用命令

IEC 61131-3準拠ファンクションの他、ファイル入出力や通信などを行う多くの制御用命令を用意しています。

表2 制御用命令 (一部抜粋)

No	グループ	内容
1	型変換ファンクション	変数のデータ型を変換する
2	算術ファンクション	加算・減算・積算・除算・べき乗・補数など
3	ビット文字列ファンクション	左右ローテーション・左右シフトなど
4	文字列ファンクション	文字列の連結・検索・比較・置き換えなど
5	比較ファンクション	=、>=、<=、>、<、<>
6	数値ファンクション	絶対値・sin・cos・tan・平方根など
7	ビット操作ファンクション	AND・NOT・OR・XORなど
8	エッジ検出ファンクションブロック	立ち上り、立ち下り検出
9	カウンタファンクションブロック	アップカウンタ、ダウンカウンタなど
10	タイマファンクションブロック	パルスカウンタ、ディレイタイマなど
11	PID処理	比例 + 積分 + 微分ループ機能
12	ファイル操作	ファイルの読み書き
13	時間操作	日付や時間取得
14	PLC操作	PLCの開始や停止、リテン書き込みなど
15	その他	国内大手メーカーからの移植に便利な機能

## ● 充実したサポート体制

開発・導入方法はもちろん、システム構築やハードウェア選定まで、お気軽にご相談ください。また、開発をお手伝いする受託開発も行います。

## INplcの主な仕様

プログラム言語 (IEC61131-3準拠)	ラダーダイアグラム(LD)、ファンクションブロックダイアグラム(FBD)、インストラクションリスト(IL)、ストラクチャードテキスト(ST)、シーケンシャルファンクションチャート(SFC)	
タスク 実行タスク数	5タスク(16タスク※1)	
	制御周期	最速0.1ms
	タスクの種類	サイクリック、デフォルト、イベント、システム※1
PLCプログラム 最大ステップ数	約300,000ステップ (約500,000ステップ※1)	
データ領域サイズ	I/O点数: 合計128KB (入力16MB、出力16MB※1) 共有メモリ: 32MB	
接点・コイル点数	1,500点 (16,000点※1)	
リテン機能	あり	
ファイル入出力	外部記憶装置(HDD/SSD/USBメモリ)へ読み書き可能	
外部I/O	デジタル、アナログ、カウンタ、サーボモータ、RS-232C/422/485、TCP/UDP、FL-net、CC-Link、CC-Link IE Field、EtherCAT、PROFIBUS、EtherNet/IP、MECHATROLINK-III等	

※1 … この機能を使用いただくにはINplc-SDK(Pro+)が必要です。

## 製品体系

INplcコントローラ	INplcはすぐ利用可能な形のプリインストールのコントローラ形式で提供します。「省スペースモデル」や「多スロットモデル」など、お客様のニーズに合わせた各種モデルを準備しております。また、ご要望に応じたカスタムPCの選定も対応しています。
INplc-SDK(Express)	INplcコントローラ付属の開発キットです。INplcのプログラム開発およびデバッグができます。
INplc-SDK(Pro+)	開発キットの機能拡張製品です。 ・タスク数 : 16タスク ・タスク種別 : システムタスク利用可能 ・ステップ数 : 約500,000ステップ ・I/O点数 : 入力16MB、出力16MB ・接点・コイル数 : 16,000点 でプログラム開発ができます。
INplc-OPCserver	INplc向けOPCサーバーソフトウェアです。

その他OEM販売も対応可能ですので、弊社営業までお気軽にご相談ください。

● EtherCAT®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。 ● その他、本カタログに記載されている商品名・会社名は、各社の登録商標または商標です。

✉ [cde@mnc.co.jp](mailto:cde@mnc.co.jp) (24時間対応)

☎ 03-6909-3371 (月～金※祝日除く 9:30～18:30)

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

株式会社マイクロネット 営業部

〒104-0032 東京都中央区八丁堀3丁目17-13 TMYビル 9F

※記載された内容および製品の仕様については、予告なく変更する場合がございます。あらかじめご了承ください。



## トレサブルコントローラ for INtime

実績あるリアルタイムOS INtimeと互換。  
多様な生産現場でのIoT対応に最適。

近年、産業用機器の制御装置に工業用PCが広く使われています。PC技術はグローバルスタンダードな技術で、これを用いた製品の製造やシステム導入に伴う人・物・予算・時間などの総合的なコスト削減が可能です。さらに製品およびサービスの品質安定が期待でき、技術的進歩にも容易に対応できます。

トレサブルコントローラ for INtimeは、PC用リアルタイムOS「INtime」によってPCに100μ秒のリアルタイム制御機能と制御情報の収集機能を実装した産業用機器制御のIoT対応に最適な製品です。Windows OSと共存/協調して動作するので、ソフトウェア・ハードウェア両面で使いやすく、将来的にも安心してお使いいただけます。

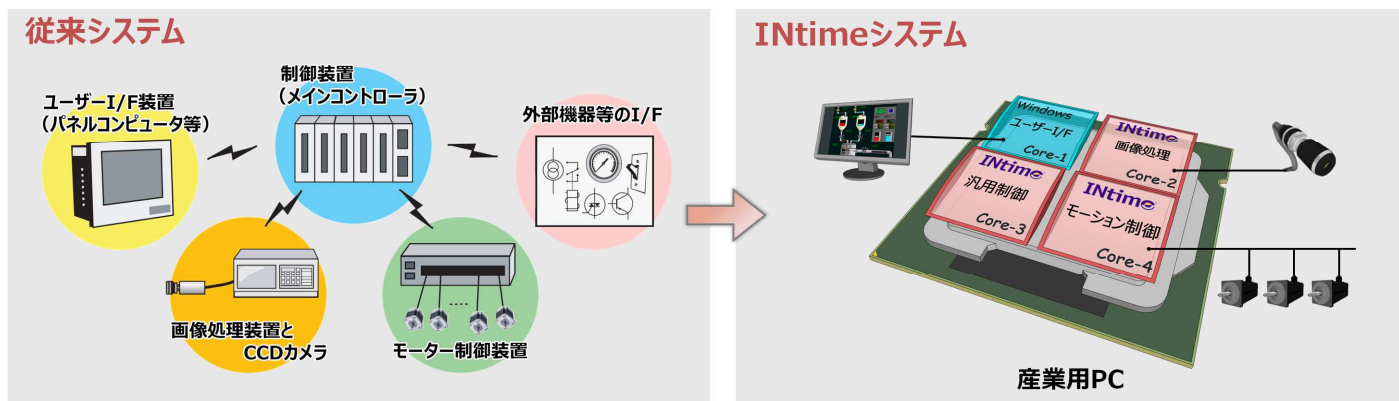


図1 従来システムとINtimeシステム

モーション制御・機器制御・画像処理・Windowsの機能が、一台のPCハードウェア上で実現します。

### INtimeの特徴

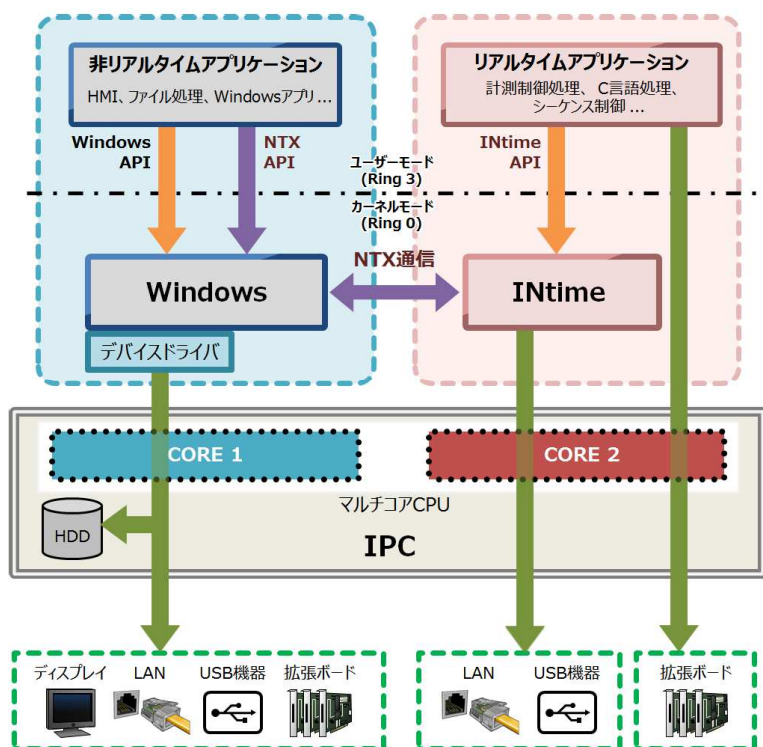


図2 INtimeシステム構成

- PC上でWindowsと協調動作するリアルタイムOS
- Windows Embeddedにも対応
- 豊富なリアルタイムスケジューリング
  - 256段階のプライオリティレベル
  - アプリケーションプログラムとしての割り込みハンドラ(IRQ/MSI対応)
- 多彩なシステムオブジェクト
  - メールボックス、セマフォ、アラーム、リージョン、共有メモリ、メッセージキュー
- マルチコアCPUでは1つのシステム上で複数のリアルタイムカーネル並列動作を実現
- Windows無しのリアルタイムOS単体動作も可能
- Visual Studioの開発環境と統合(Visual C/C++)
- ユーザーモードによる完全メモリ保護、アドレス分離
- ソースレベルのダイナミックデバッガをVisual Studioに統合
  - OSを止めずにデバッグ可能
  - WindowsとINtimeのアプリケーションを同期デバッグ
- リアルタイムTCP/IPドライバおよびライブラリを標準装備

## INtimeの動作モード

INtimeシステムでは、PCプラットフォーム上に、ユーザーのニーズにあわせた動作モードを実現することが可能です。

### シングルカーネルモード:

INtimeをインストールしたPC上では、マルチコアCPUのコアの1つがINtimeに専有して割り当てられます。

Windowsとはコアが分離しているため、Windows側の負荷がINtimeの動作に影響することはありません。

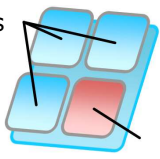
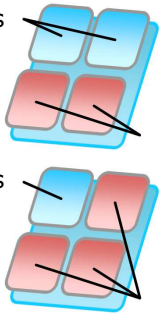
### マルチカーネルモード:

クアッドコア以上のCPUの場合は、1つのシステム上に2つ以上のリアルタイムカーネルを動作させることができます。

それぞれのリアルタイムカーネルは、非同期に動作します。

このモードを利用することで、処理負荷の高いアプリケーションを独立したカーネルへと移すなどの負荷分散を実現できます。

表1 カーネルモード

	Quad Core CPU
シングルカーネルモード Single Kernel Mode	
マルチカーネルモード Multi Kernel Mode	

## I/Oサポート

INtimeアプリケーションのために、I/Oハードウェアに対してのダイレクトなアクセス手順を用意しています。

I/Oハードウェアアクセスのための困難なカーネルデバッグ技術を習得する必要も固有のデバイスドライバを作成する必要もありません。

### PCI & PCI Express

INtime APIには、アナログ/デジタル入出力ボード等のPCIおよびPCI Expressデバイスを検出して初期化を行う関数が実装されています。

また、INtimeアプリケーションがデバイスにアクセスするサンプルコードおよびテンプレートも用意しています。

### Real-time TCP/IPネットワークアクセス

INtimeはWindowsに依存しない、INtime上で駆動するネットワークドライバが付属しています。

INtimeアプリケーションはソケットAPIを使用することで、このネットワークドライバを利用して、外部機器との通信を行うことができます。

INtimeのTCP/IPドライバはリアルタイムスレッドとして稼動するため、Windows TCP/IP(Winsock)通信と比較して性能の安定化が望め、信頼性面でも向上が見込めます。

### USBアクセスサポート

USB3.0ホストコントローラーインターフェースのReal-time stackサポートが実装されています。

このスタックを利用することで、INtimeアプリケーションはUSB機器に対してダイレクトアクセスを行います。

### 各種フィールドバス対応

INtimeでは各種フィールドバスのドライバソフトウェアを幅広く対応しています。

EtherCAT、CC-Link、CC-Link IE、FL-net、MECHATROLINKなど多数のドライバが、マイクロネットやサードパーティによって提供されます。

## ライブラリ

### CとC++のフルサポート

INtimeのC99互換CライブラリとANSI C++ライブラリはSTLのためのフルサポートを含んでいます。

INtime RtClassライブラリは標準リアルタイム・オブジェクトにC++インターフェースを供給します。

### Real-time Shared Libraries (RSL)

INtime Real-time Shared Library (RSL) は、WindowsのDLLのような存在となります。

RSLは、複数のリアルタイムプロセスに1つのリアルタイムコードライブラリを共有する手段を提供します。

### SSE Library

Intel Integrated Performance Primitive Static libraries (Intel IPP) を利用することで、開発者はINtimeのリアルタイムプラットフォーム上での演算スピードを向上できます。

## デバッグおよびランタイム

### 充実のデバッグ機能

INtimeはMicrosoft Visual Studio開発環境と完全統合され、コーディング、ビルドに加えて、INtimeアプリケーションのデバッグ実行が可能です。

また、シリアル経由でデバッガとしてSDM(System Debugging Monitor)を標準装備し、外部シリアル端末にハードウェア例外とラップ情報を出力して、現在のシステム状況を参照したり、端末からの入力によりデバッグ作業を進めることも可能です。

この他、INtimeオブジェクトの状態監視オブジェクトブラウザINtime Explorerや、カーネル上で実行されるスレッドのタイムチャートを収集するINscopeなどのデバッグツールが標準実装されています。

### 例外ハンドリングと障害管理

INtimeはデバッグ効率を改善すると共にランタイムに信頼性を提供するために、いくつかのレベルの例外処理および障害管理機能を実装しています。

例外がアプリケーションによって構造化した例外処理あるいはC++例外処理の使用によって行われなかった場合、設定されたデバッグ方法で処理されます。

もっとも低レベルの場合は、例外が発生したスレッドを単純にサスペンドし、他システムの実行の継続を確保します。

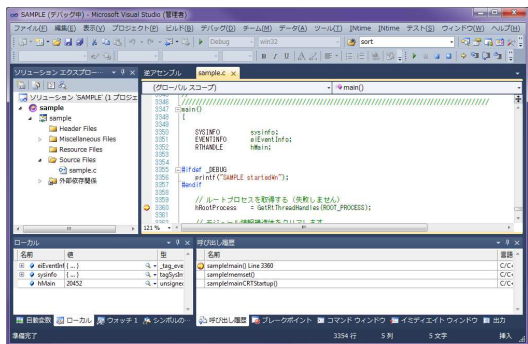


図3 Microsoft Visual Studio

## ダイナミックオブジェクトブラウザ 【INtime Explorer】

INtime Explorerツールは、INtimeの管理するプロセス、スレッド、メールボックス、セマフォ、メモリ他すべてのReal-timeオブジェクトの情報をユーザーに提供します。ユーザーはこれらの情報をダイナミックに参照し、管理することができます。また、アプリケーションエラーのレポート機能も持っています。

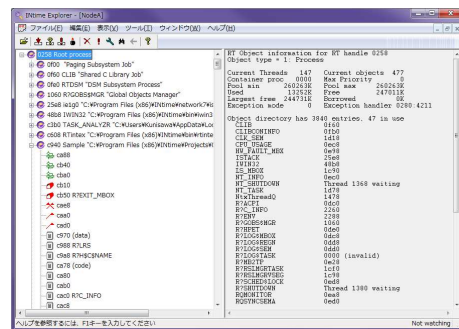


図4 INtime Explorer

## 統合開発環境 【Microsoft Visual Studio】

INtimeアプリケーションのための統合開発環境には、Visual Studioを利用します。

Visual Studioの強力な開発機能は勿論、デバッグ機能もそのまま利用できるため、Real-timeプロセスや変数のモニタリングやブレークポイントを用いたステップ実行も可能です。さらに、ページ違反やスタック例外、バッドポインタアクセスやゼロ除算などのリアルタイムプロセスで発生したエラーの位置をVisual Studio上で指し示しますので、デバッグ作業を非常に効率的に行うことが可能です。

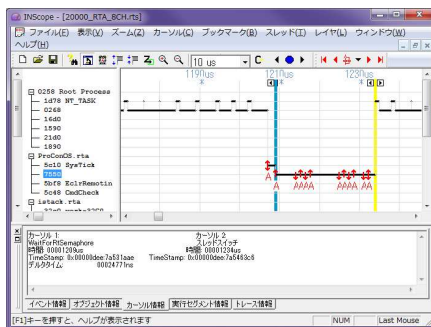


図5 INscope

## Real-timeシステムアナライザ 【INscope】

INscopeは、INtimeのReal-timeパフォーマンスのアナライザツールです。Real-timeスレッドのシーケンスデータの正確な時間で計測します。このツールを使用することで、Real-timeスレッドの処理時間、スレッド切り替え状況、APIコール履歴などを分析できます。

## 制御データトレース機能

### 1. 生産管理情報、品質情報、保守情報をトレース

トレーサブルコントローラ for INtimeでは、標準機能としてトレーサブルAPIが提供されます。トレーサブルAPIは、アプリケーションプログラムに組み込んで利用でき、生産管理情報や製造ロット毎の品質情報、アプリケーションの保守情報などに利用することができます。(※主要なAPIは、裏面に記載)

### 2. 制御データトレース機能の仕組み

制御データのトレースする仕組みは、「トレーサブルRTOSインターフェース」APIや「トレーサブルWindowsUIインターフェース」APIなどをアプリケーション作成時に利用するだけで組み込まれます。組み込まれたAPIは、フラグを立てることによって、トレース情報をWindowsから確認できるファイルシステムで保存されます。

### 3. 大容量ストレージを利用して、大量データ蓄積

PCベースコントローラの強みを活用して、大容量のストレージを活用することで、大量の制御データを蓄積することができます。2TBのトレーサブル容量を用意した場合、2Byteデータ1000項目を10ms毎に蓄積したとしても、約3年分のデータを保存することができます。

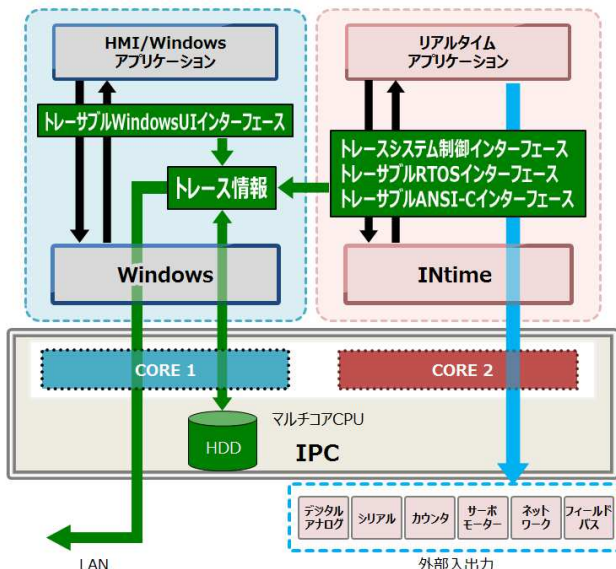


図6 トレーサブルコントローラ機能ブロック

## トレーサブルAPIの使用例

```
// 温度センサを監視して異常判定するスレッド
void ThreadTemperatureWarning(void)
{
    RTHANDLE hIntval; // 処理サイクル
    BYTE buf[128]; // 受信バッファ
    TCHANDLE hT = TC_ON; // トレースをアクティブ
    hIntval = tcCreateRtAlarm(hT, 10); // サイクル生成
    while( 1 )
    {
        // 処理サイクル待ち
        hIntval=tcWaitForRtAlarm(hT,hIntval,100MS);
        // 温度センサユニットとCOM1で通信
        tc_rs_transmit(hT,COM1,"GET",3);
        tc_rs_receive(hT,COM1,buf,10);
        // 設計温度上限判定
        fTemp1 = atol(buf);
        if(fTemp1 > TempUpLim)
        {
            // 温度異常発生イベントを発行
            tcReleaseRtSemaphore(hT,hWarning,1);
            tcMarking(hT,"温度異常障害");
        }
    }
}
```

図7 トレーサブルAPI 使用例

## トレース情報の例

```
13:10:10.238,200,3ee0,3ee8,tcWaitForRtAlarm(i),0x3eec,100
13:10:10.338,200,3ee0,3ee8,tcWaitForRtAlarm(o),ret=TRUE,E_OK
13:10:10.339,200,3ee0,3ee8,tc_rs_transmit(i),COM1,msg="GET",len=3
13:10:10.340,200,3ee0,3ee8,tc_rs_transmit(o),E_OK
13:10:10.341,200,3ee0,3ee8,tc_rs_receive(i),COM1,0x410050,len=10
13:10:10.346,200,3ee0,3ee8,tc_rs_receive(o),msg="132.5",len=5,E_OK
13:10:10.348,200,3ee0,3ee8,tcReleaseRtSemaphore(i),3f00,unit=1
13:10:10.349,200,3ee0,3ee8,tcReleaseRtSemaphore(o),ret=TRUE,E_OK
13:10:10.350,200,3ee0,3ee8,tcMarking,"温度異常障害"
```

図8 トレース情報の例

## オプションなコントローラ通信、フィールドバスI/F

マイクロネットが用意している各種外部インターフェースドライバもトレーサブルの仕組みが組み込まれています。

表2 各種インターフェースドライバ

RSI-001	RS-232C/422/485通信
RSI-008i	高速多CHシリアル通信
RSI-040	FL-netバス通信
RSI-CCIE	CC-Link IEバス通信
RSI-ECAT	EtherCATマスター
RSI-CAN	CANバス通信

## リアルタイム計測制御エンジンの仕様

プライオリティ スケジューリング	(高) 0 ~ 255 (低) 256段階
カーネルティック	50µs, 100µs, 125µs, 200µs, 250µs, 500µs, 1ms, 2ms, 5ms, 10ms
ラウンドロビン スケジューリング	適用プライオリティ閾値変動可 20ms ~ 100ms (10ms単位)
メールボックス	FIFO式 / 優先度式
セマフォ	FIFO式 / 優先度式
利用可能メモリサイズ	システムトータルで2GB
割り込み	ハンドラ / スレッド構造 IRQ/MSI対応
開発言語	C/C++
リアルタイムデバガ	Visual Studio Integrated Debugger, Spider Debugger, SDM (System Debug Monitor)
標準I/O	TCP/IP, UDP/IP, USB, RS232C
最大オブジェクト数	8190個 (システムを含む)

## トレーサブルAPI (主要抜粋)

### 1. トレースシステム制御インターフェース

トレースシステム全体、あるいは一部の働きをカスタマイズできます。

```
tcCreateHandle, tcChangeFilter, tcStartTrace, tcStopTrace,  
tcProductionTrace, tcQuarityTrace, tcMarking
```

### 2. トレーサブルRTOSインターフェース

リアルタイムOS INtimeのAPIトレース付きバリエーション。主要APIを網羅。

メモリ管理	tcAllocateRtMemory, tcFreeRtMemory, tcCreateRtMemoryHandle, tcDeleteRtMemoryHandle, tcMapRtSharedMemory
リアルタイム プロセス管理	tcCreateRtProcess, tcExitRtProcess, tcWaitForRtProcess, tcDeleteRtProcess, tcTerminateRtProcess, tcSetRtProcessMaxPriority
オブジェクト名 管理	tcCatalogRtHandle, tcLookupRtHandle, tcUncatalogRtHandle
リアルタイム スレッド管理	tcCreateRtThread, tcDeleteRtThread, tcGetRtThreadHandles, tcSetRtThreadPriority, tcRtSleep, tcRtSleepEx, tcSuspendRtThread, tcResumeRtThread, tcKnRtSleep
セマフォ管理	tcCreateRtSemaphore, tcDeleteRtSemaphore, tcWaitForRtSemaphore, tcReleaseRtSemaphore
メールボックス 管理	tcCreateRtMailbox, tcDeleteRtMailbox, tcSendRtHandle, tcReceiveRtHandle, tcSendRtData, tcReceiveRtData
リージョン 管理	tcCreateRtRegion, tcDeleteRtRegion, tcAcceptRtControl, tcWaitForRtControl, tcReleaseRtControl
ハードウェア 割り込み管理	tcSetRtInterruptHandlerEx, tcResetRtInterruptHandler, tcSignalRtInterruptThread, tcWaitForRtInterrupt
アラーム管理	tcCreateRtAlarm, tcWaitForRtAlarm, tcResetRtAlarm, tcDeleteRtAlarm
グローバル オブジェクト	tcGetRtNodeStatus, tcGetRemoteRootRtProcess, tcGetRtNodeLocationByName, tcGetRtNodeInfo
補助	tcCopyRtSystemInfo, tcWaitForRtIoService, tcGetLastRtError

### 3. トレーサブルWindowsUIインターフェース

リアルタイムアプリケーションのWindows UI開発に利用可能。

### 4. トレーサブルANSI-Cインターフェース

主にファイルシステムインターフェースをトレースできるC関数ラッパー。

## データベースアクセスライブラリ (主要抜粋)

.netによるデータベース分析アプリケーションの開発に使用できます。

データアクセス制御	RTTlib::open, RTTlib::close, RTTlib::seek
データレコード制御	RTTlib::next, RTTlib::filter
データパーサー	RTTParser::Parse

## 充実のサポート体制

### 製品メンテナンスサービス

システムの開発および保守をサポートするメンテナンス製品です。購入後1年間付属しており、本サービス期間中はバージョンアップを自由に行うことのできる「製品更新サービス」と弊社技術スタッフによる「技術サポートサービス」を提供します。

### オンサイトサポートサービス

通常の技術サポートの他、お客様のニーズに応じて弊社の技術者が直接お客様の元へとサポートに伺うオンサイトサービスもご用意しております。日本国内に留まらず、海外への派遣にも対応しております。

### セミナーサービス

製品導入後のお客様向けに、セミナーサービスをご用意しております。弊社の技術者が直接お客様の元で行う出張セミナーから、高度ポリテクセンターにて実施される定期セミナーなど、ニーズに合わせてご利用いただけます。

### プリインストールPC販売

マイクロネットでは工業用として実績のあるPCに製品をプリインストールして販売しています。マイクロネットがリアルタイム性能の適合性を動作確認しています。

### 受託開発サービス

お客様の開発業務の一部をお受けする受託開発サービスを行っています。INtime用のドライバ開発からシステム全体の開発まで、幅広くお受けしています。

## 実行環境適合ハードウェア

プラットフォーム	PC/AT互換パソコン
CPU	Intel / Intel互換CPU
メモリ	64MB以上
ストレージ	<必要容量> 約 90MB <標準トレーサブル容量> 32GB以上 <フォーマット> FAT32 / NTFS
適合Windows	Windows Server 2008およびServer 2008 R2 Windows 7 Windows 8 および 8.1 Windows 10 Windows Embedded
適合開発環境	Visual Studio 2008, 2010, 2012, 2013, 2015

## 製品体系

開発キット	
TraCon-SDK	トレーサブルコントローラ for INtimeの開発キット。 印刷物の日本語マニュアルが付属。
ランタイムライセンス	
TraCon-RT	トレーサブルコントローラ for INtime 組み込みライセンス
TraCon-MCRT	トレーサブルコントローラ for INtime 組み込みライセンス (マルチコア)

● EtherCAT®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。 ● その他、本カタログに記載されている商品名・会社名は、各社の登録商標または商標です。

✉ [cde@mnc.co.jp](mailto:cde@mnc.co.jp) (24時間対応)

☎ **03-6909-3371** (月~金※祝日除く 9:30~18:30)

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

株式会社マイクロネット 営業部

〒104-0032 東京都中央区八丁堀3丁目17-13 TMYビル 9F

※記載された内容および製品の仕様については、予告なく変更する場合がございます。あらかじめご了承ください

# EtherCAT<sup>®</sup> Master RSI-ECAT

INtime<sup>®</sup>  
for Windows<sup>®</sup>

EtherCAT<sup>®</sup>は、産業用 PC(IPC)の LAN ポートを直接制御することにより、拡張ボードの増設無しにソフトウェアで多軸、多ノード、大容量での高速通信を実現する世界最高クラスのマシンコントロールネットワークです。  
RSI-ECAT は、リアルタイム OS "INtime" 上でこの EtherCAT<sup>®</sup>を利用するためのマスタースタックソフトウェアです。

## ◆ EtherCAT とは

### ● 超高速リアルタイムフィールドバス

マスタースレーブ間においてデータフレームを環状鉄道状に流す送受信方式。  
高速なデータ循環で遅延が最小限となり、100Mbps Ethernet の通信環境を実現することができます。  
IPC の強力な CPU パワーを利用することでさらに高速なリアルタイムフィールドバスを実現することができます。

### ● 最大延長距離と最大ノード数

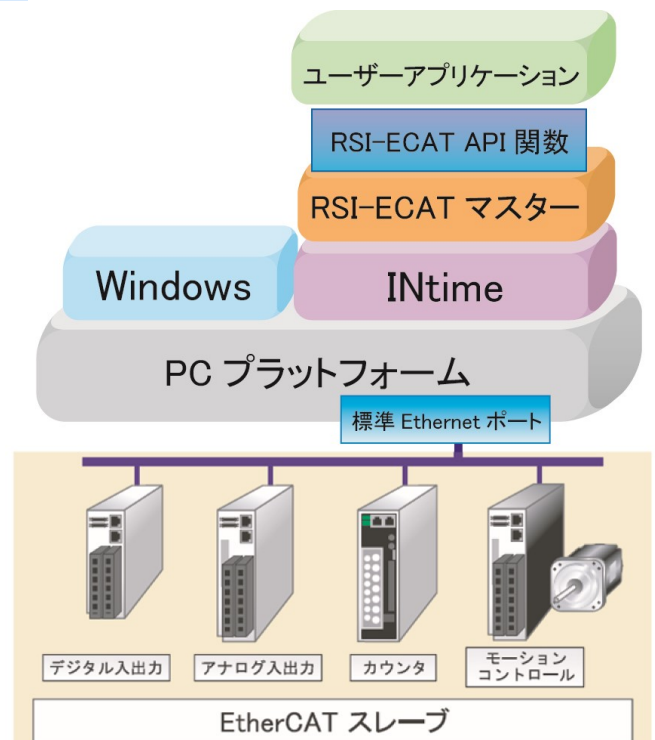
Ethernet と同様に、100Base-TX の最大延長距離である 100m を実現。  
また最大ノード数は規格上 65535 と、非常に自由度の高い配線形態をとることが可能です。

### ● 拡張ボードの増設なしに、すぐにフィールドバスが実現可能

産業用 PC に標準実装されている Ethernet インターフェースを利用してハードウェアを増設することなく利用できます。  
さらにネットワークケーブルも汎用の Ethernet ケーブルが利用可能。  
ハードウェア設備の面で大きくコストカットを図ることが可能です。

## ◆ 主な特長

- RSI-ECAT は INtime RTOS 用 EtherCAT マスターソフトウェアです。
- PC プラットフォームの CPU パワーにより、1 サイクル 100  $\mu$  秒から設定可能
- アプリケーションの開発は Microsoft Visual Studio C/C++。
- 未経験者にも使い易い専用 API が用意されており、簡単に EtherCAT アプリケーション構築が可能。
- 各種 EtherCAT スレーブデバイスに接続可能。
- EtherCAT スレーブのコンフィグレーション作成には RSI-ECAT-Studio を使用。



## API 一覧(一部抜粋)

初期化	EhOpen()	EtherCAT 通信開始
	EhClose()	EtherCAT 通信終了
マスター制御	EhRqState()	マスターステート変更
	EhGetState()	マスターステート取得
	EhWaitForCyclic()	マスターサイクリック処理の待機
スレーブ検索/管理	EhFindSlave()	スレーブを検索
	EhGetSlaveStatus()	スレーブステータス取得
	EhGetOnlineSlaveCount()	接続スレーブ数取得
ProcessData アクセス	EhReadByte()	VIOS IN 領域から 8 ビット読取
	EhWriteByte()	VIOS OUT 領域へ 8 ビット書込
	EhReadWord()	VIOS IN 領域から 16 ビット読取
	EhWriteWord()	VIOS OUT 領域へ 16 ビット書込
	EhReadDWord()	VIOS IN 領域から 32 ビット読取
	EhWriteDWord()	VIOS OUT 領域へ 32 ビット書込

## 開発環境

Visual Studio	対応 Ver: 2008, 2010, 2012, 2013, 2015 対応開発言語: C, C++
---------------	--

※ 開発言語環境はお客様でご用意ください。

## RSI-ECAT の制御プログラミング

RSI-ECAT は、多数の EtherCAT スレーブ機器をあたかも I/O ポートに直結されたデバイスのように制御できるよう配慮されています。複雑な EtherCAT マスターの初期化手続きは使い易く配慮された API 数個のコールで完了し、また I/O 関数自体もシンプルにデザインされています。EtherCAT の専門知識が無くても I/O デバイスの取扱い経験さえあればすぐに制御プログラミングに集中できるのが、RSI-ECAT-Master の特長です。

### RSI-ECAT での基本的な制御手順

RSI-ECAT は、基本的に以下の手順によってスレーブ機器の制御を実施します。

- ① マスターの初期化**  
アプリケーションは、まず RSI-ECAT-Master との接続を初期化します。
- ② マスターを OPERATIONAL 状態に移行**  
EtherCAT のサイクリック通信を開始します。
- ③ 制御対象スレーブの検索**  
EtherCAT スレーブには不揮発な ID (VendorID と ProductCode) が書き込まれています。これをキーにして制御対象のスレーブの存在と位置を検索取得します。
- ④ マスターサイクル周期待ち**  
マスターサイクルを迎えるまで待機します。
- ⑤ 入出力制御**  
RSI-ECAT によって、EtherCAT スレーブ機器は連続する仮想のアドレス空間 (VIOS) に再配置されています。I/O アドレス空間を扱うように EtherCAT スレーブ機器に対する入出力を制御し、I/O スレーブからモーションスレーブまで様々な EtherCAT スレーブ機器を制御します。

## RSI-ECAT-Master に含まれるもの

RSI-ECAT-Master インストール CD	■ EtherCAT マスターソフトウェア
	■ API ライブラリ
	■ 取扱説明書
	■ サンプルプログラム API コールによって EtherCAT® の機能を利用する例を示したサンプルコードです。(Microsoft Visual Studio 2008 用プロジェクトファイル)

## 動作環境

OS	INtime 4.2 以上
CPU	Intel/Intel 互換 CPU(Core2Duo 以上推奨)
代表的な 対応 Ethernet コントローラ	Intel PCIe Gigabit ethernet controller シリーズ (intel チップセット内蔵型) Intel I210 Gigabit Network Connection シリーズ Intel I350 Gigabit Network Connection シリーズ Realtek PCIe Gigabit ethernet controller シリーズ Intel PCI Pro/100 シリーズ

## 製品体系

RSI-ECAT-Master	INtime 向け EtherCAT マスターソフトウェア
RSI-ECAT-Master/DC	DC 機能(DistributedClock: 時刻同期)をサポートしたタイプ。DC 機能を必須とする EtherCAT スレーブを制御する場合に必須となります。
RSI-ECAT-Master/RED	DC 機能に加え、EtherCAT1 回線に対して Ethernet ポートを 2 系統使用し、通信の冗長性を図る CableRedundancy 機能、およびケーブルホットプラグをサポートした INtime 向け EtherCAT マスターソフトウェア

※INtime 組み込みライセンスとセットで販売します。

## 別売製品

EtherCAT 構成ツール RSI-ECAT-Studio	EtherCAT スレーブの構成情報を自動的に検出・認識するコンフィガレーションツールです。RSW-ECAT-Master の利用に必要な構成情報を作成するため、必ず 1 ライセンス導入していただく必要があります。
(株)ハイパーテック HLS-ECMC/RSI INtime Motion Master Pack	(株)ハイパーテックの開発した INtime 用 EtherCAT モーション制御ソフトウェア。国際標準規格に準拠した API 関数でモーション制御を実現。
(株)テクノ MotionLibrary	(株)テクノの開発した INtime 用 EtherCAT モーション制御ソフトウェア。独自の制御 API でモーション制御を実現。

※ EtherCAT®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。  
※ その他、本カタログに記載されている商品名、会社名は、各社の登録商標または商標です。

製品に関するお問い合わせはコチラまで

✉ [cde@mnc.co.jp](mailto:cde@mnc.co.jp)

☎ 03-6909-3371

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

株式会社マイクロネット 営業部

〒104-0032 東京都中央区八丁堀 3-17-13 TMYビル 9F

※機能仕様および本体仕様については、予告なく変更する場合がございます。あらかじめご了承ください。

様式 11-1005-70907 (2017/09/500)