

リアルタイム制御とデータ収集機能を1台で構成

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。

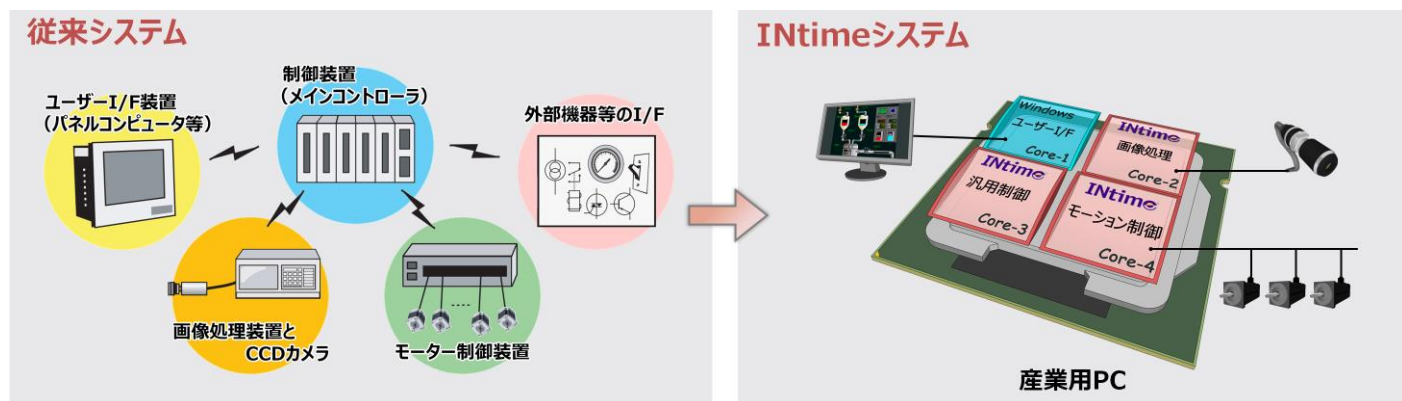


図1 従来システムとINtimeシステム

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

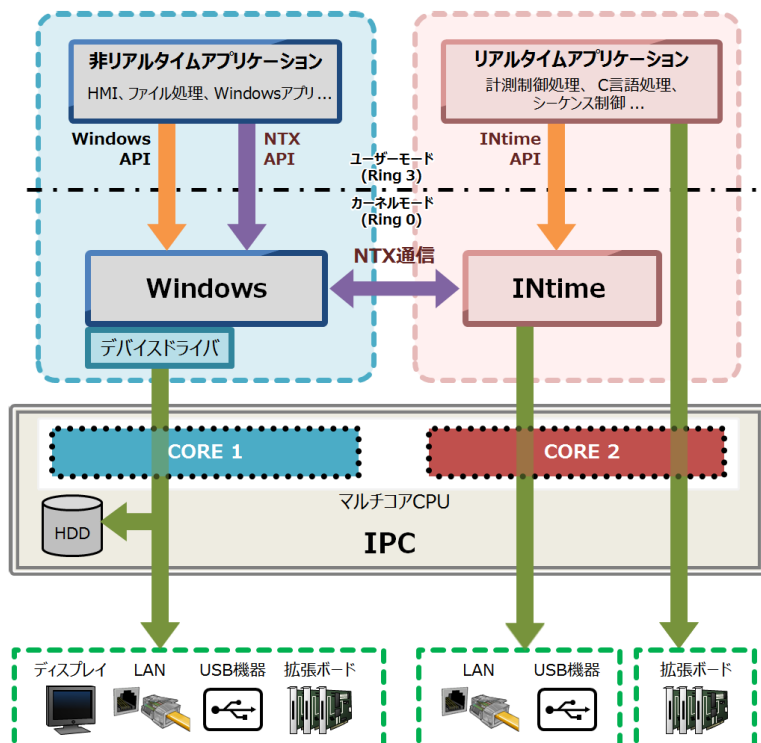


図2 INtimeシステム構成

- 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グを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。

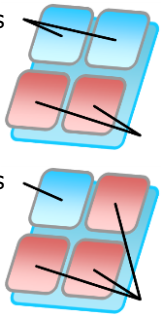
マルチカーネルモード:

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	Windows  INtime
マルチカーネルモード Multi Kernel Mode	Windows  INtime

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、シリアル経由デバッグとして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。

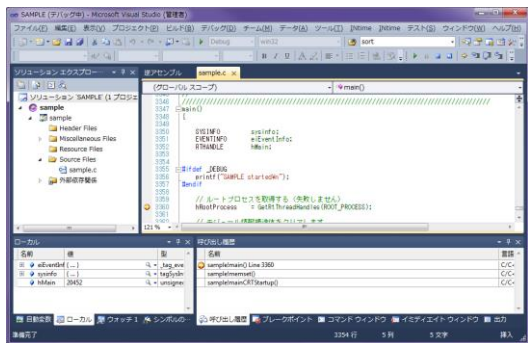


図3 Microsoft Visual Studio

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。

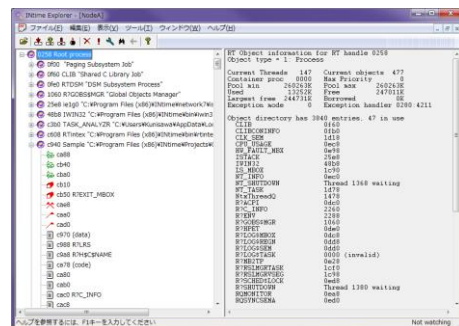


図4 INtime Explorer

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、バッドポイント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。

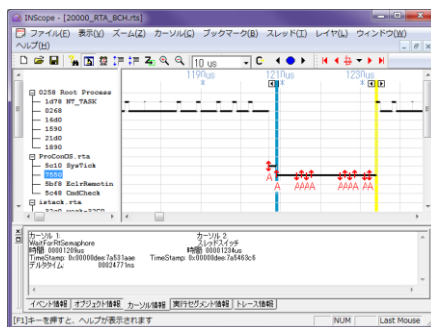


図5 INscope

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。

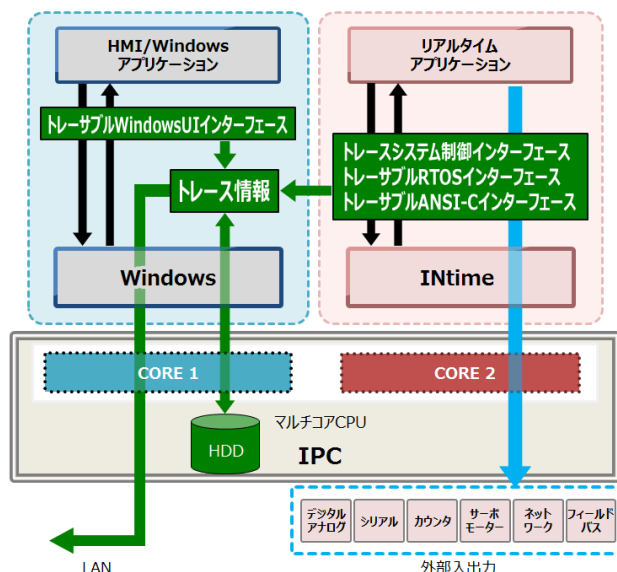


図6 トレーサブルコントローラ機能ブロック

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 = atoi(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, tcQuarantyTrace, 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。

※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。