

# CHOICES

## オートメーションコントロールの 未来

白書-2

### コントローラ

「コントローラのあらゆる選択肢が揃っているとすれば、その情報を生かして意思決定するには、表向きではなく、その中心機能をアプリケーションに当てはめてどのような答えが出るかを考えることが求められます。」

## 目次

- 1 はじめに
  - オートメーションのトレンドは、コントローラに影響を及ぼす
  - 新技術の導入
  - アーキテクチャの細分化
  - オープンシステム
  - アプリケーションに特化したソリューション
  - 工場作業員の役割
  
- 4 これまでの制御の定義
  - リレーによる制御
  - シングル・ボード・コントローラ
  - プログラマブルコントローラ
  - PC使用の制御
  - 分散制御システム
  - コントロールの新しい定義問題
  
- 12 あらゆるレベルの選択肢
  - ハードウェアプラットフォーム/パッケージング
  - オペレーティングシステムの選択肢
  - プログラミング法
  - ロジックエンジン
  - マン・マシン・インターフェイス
  
- 19 情報を生かして意思決定するための重要な検討事項
  - オープン度
  - 速度
  - 信頼性/耐環境性の高い運用
  - アプリケーション：不連続/プロセスの連続体
  - 据付け/統合
  - 通信性能
  - 安全性/データの整合性
  - コスト
  
- 27 結論

## はじめに

将来の様々なビジョンや人間の進む方向、そして姿を現しつつあるテクノロジーを考えると、すべてに共通するテーマとして「選択肢」・CHOICES<sup>1</sup>があることがわかります。絶対、というものはありません。その人に有効なものでも、他の同僚には、有効性がないかも知れません。あるアプリケーションには正しいソリューションでも、アプリケーションが変わると、そのソリューションではまったく間違った結果になることもあります。だからこそ、どのレベルの製造事業にも興奮を覚えるほどの様々な選択肢が揃っています。

コントローラの場合がまさにそれであり、ロックウェル・オートメーション CHOICES<sup>1</sup>白書シリーズの本号の主題です。白書<sup>1</sup>では、製造業の将来を決める要因(ドライバ)の概要を紹介しました。次にこうした主カドライバを取り上げ、こうした主カドライバがどのようにトレンドに発展して制御システムに影響を及ぼすかについて結論を導き出しました。まだ白書<sup>1</sup>をお読みでなければ、ぜひご一読をお勧めします。これは後続号のよい基礎となります。

本号では、特にコントローラで利用可能な選択肢にこうしたトレンドを結び付け、特有の必要条件を満たすソリューションを模索する際の検討要素を説明します。このシリーズの他の号では、マン・マシン・インターフェイス、ネットワーク、I/O、センサとアクチュエータ、パワーデバイスその他の中心的な自動化技術を扱います。

本書を読み進まれる中で、コントローラテクノロジーとアプリケーションが向かっている行く先について、当社の考えがよくおわかりいただけることを願っています。ただ一つのソリューションを無理に押しつけるつもりはありませんし、このパズルのピースの中には、当社でご提供しなかったり、できないものもあることを認めます。しかし、それでもお客さまが情報を生かして意思決定できるよう、選択肢を見分けるお手伝いをいたします。

ご意見をお待ちしています。内容にご賛同であればそれを、またお気に召さなければそれをお聞かせください。何かアイデアがあればお知らせください。当社は、お客さまの声に耳を傾け、それに応えてまいります。

オートメーションのトレンドは、コントローラに影響を及ぼす CHOICESシリーズの白書<sup>1</sup>では、オートメーションシステムに影響を及ぼす主なトレンドとして、新技術採用が早いこと、制御システムの細分化の進行、「オープン」システムの役割が明らかになってきたこと、アプリケーションに大きく特化したソリューション、工場作業員の役割の変化の5つを概説しました。驚くべきことではありませんが、これらの主要トレンドはコントローラの進化に常に影響を及ぼします。

**新技術の導入** 新技術はたいてい政府や民間セクタで開発されますが、驚くべき速さで産業用制御システムへ浸透してゆきます。集積回路が発表されてからプログラマブルコントローラが市場に登場するまでの期間は数十年でした。ファジー理論技術が誕生したのはわずか数年前のことですが、今では、プラスチックを注入する成形用コントローラで均一性を保つために役立っています。本来自動車用に開発されたCAN(コントローラ・エリア・ネットワーク)技術は、今では優れたプラント・フロア・デバイス間ネットワークを実現するコントローラに採用されています。わずか数年前に発表された Pentium<sup>®</sup>級のチップも、今ではパーソナルコンピュータを土台とする制御システムの標準となっています。

このような新技術にはリスクが伴います。かつては採用のサイクルが長かったため、コントローラに適用した新技術が機能し、信頼性があることを証明するのにその時間が役立っていました。現在では、新技術が制御システムに浸透するのがあまりに速いため、購入者であるユーザは、甘んじて受け入れられる相対リスクの評価に余計な労力を払わされています。

たとえば、PCMCIA (Personal Computer Memory Card International Association) 技術を取り上げてみましょう。PCMCIAカードは、1991年にパーソナルコンピュータ用として開発され、今では数多くの制御システムで共通の接続用機能となっています。このカードは便利で使いやすく、どこでも手に入りますが、技術やハードウェアの信頼性と耐久性にまだ問題があります。これらを採用するべきでないという意味ではなく、産業環境での使用にあたってはある程度のリスクが伴うため、PCMCIAの柔軟性という長所と引き換えに、信頼性の面での大きなリスクを進んで受け入れる必要があるということです。

**アーキテクチャの細分化** ある意味でコントローラ技術はアーキテクチャによって完結しました。リレーは、ほぼ1対1のI/Oの割合を実現しましたが、制御の必要条件や機能に後押しされ、業界は集中化の進んだプログラマブルコントローラ(PLC)や分散制御システム(DCS)に移行しました。そして、コントロールの主要モデルとして、複数の機能を中央の大型コンピュータや制御システム1台で制御するというピラミッド構造が出現しました。

現在、時代の振り子は逆戻りを始めています。製造者がプロセスの特定要素を厳格に制御する必要を感じ始め、テクノロジーの進歩で費用効率のよい分散が可能になるにつれて、集中制御モデルは論理的な細分化コンポーネントに分かれました。かつて1台の大型プログラマブルコントローラが製造セルの全機能を負担していたところでは、今や複数の小型コントローラや「マイクロ」コントローラが常駐し、ネットワークで結ばれています。こうした複数のシステムは、モーション・コントロール・システム、パーソナルコンピュータ、シングル・ループ・コントローラ、ミニチュアDCSシステムなどといった別の種類のコントローラとも頻繁にリンクされています。

**オープンシステム** 究極の到達点とか特定の終着点としてオープンシステムを捉えるのではなく、連続体として、つまりそのまま複次元のものとして捉える必要があると考えています。オープン度は、プログラミング法、オペレーティングシステム、オペレータインターフェイス(OI)手法、ハードウェアプラットフォーム、ネットワーク、I/Oといったコントローラのほぼあらゆる面で見られます。つまり、制御システムを購入する人は、低コスト、高品質、交換の速さ、高信頼性、使いやすさなどの具体的な目標を達成して、それをもとにコントローラのオープン度のレベルや度合いを評価しなければならぬということになります。「オープン」それ自体は意味をなさない言葉であり、オープン度のためにオープン度を選定するというのは、コストがかかり、時間の浪費につながる発想です。

**アプリケーションに特化したソリューション** 同時に、コントローラは根本的にアプリケーションへの専門化が進んだものになり、「プロセスを制御する」仕事から情報をコントロールするものへと移行してきました。1980年代には、基本的に6種類のコントローラを見分けることができました。ブラックボックス(埋め込み型)、リレー、シングル・ボード・コンピュータ(SBC)、プログラマブルコントローラ、コンピュータ数値制御コントローラ(CNC)およびDCSです。現在では、特注のアプリケーションソフトウェアやファームウェアを用いることによって、これら6種類の基本システムは「変異」し、注入成形コントローラ、ループコントローラ、バッチコントローラ、モーションコントローラなど数多くの広範なアプリケーション専門化製品が生まれています。そして顧客の多くは、これらのおおむねすべてを1つの設備の中に置いています。

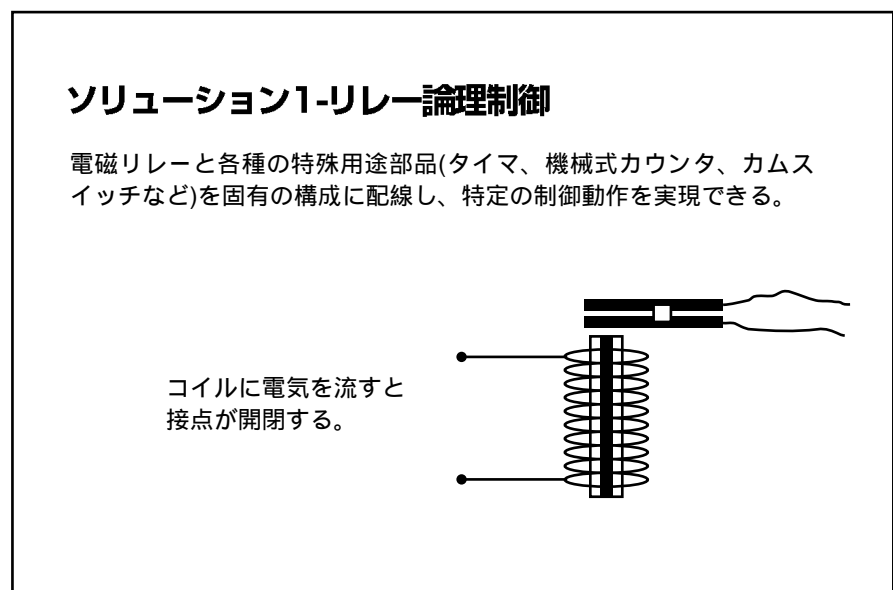
**工場作業員の役割** 最後に、工場作業員の役割の変化もコントローラの進化に影響を及ぼしています。プラントフロア作業員は、かつてないほどの技能を持ち、製造作業の維持と向上に参画しています。参画させるためには、重要度の高い意思決定情報が求められます。コントローラは、もはやI/Oを処理するだけでなく、重要なデータをプラント・フロア・ワークステーションとディスプレイを通じてオペレータに送ったり、製造実行システム(MES)を通じて企業内の別の部分へ送ったり、さらにはインターネット、イントラネット、エクストラネットを通じて工場の外へまでも送ったりします。

## これまでの制御の定義

コントローラという言葉は、問いかける人によって数多くの意味を持ちます。ガレージドアのOEMメーカーにとってのコントローラは小型のリレーバンクでしょうし、注入成形装置メーカーにとっては特注の「ブラックボックス」コントローラとなるでしょう。ポテトチップのメーカーにとってはプログラマブルコントローラとなり、配送倉庫ではパーソナルコンピュータであり、精油所ではDCSとなるでしょう。

今日まで制御は、それを実行する「箱」で定義されてきました。確認のため、少し時間を割いてコントローラのいくつかの基本形-リレー、SBC、プログラマブルコントローラ、モーションコントローラ、DCSを振り返ってみます。

**リレーによる制御** 電子制御によるソリューションが開発される以前は、電磁リレー(および空気式と油圧式の類似品)が標準的な制御手段でした。リレーは、押しボタン、セレクトスイッチ、光電センサのような入力デバイスの切替え、タイミング、複合化機構として使われます(図A参照)。



図A

リレーはきわめて直感的ではありますが、機械式デバイスであるがゆえに、現代のプログラマブル・コントロール・システムに見られるプログラミングやトラブルシューティングの柔軟性はありません。また、リレーは非常に場所をとる、長い配線が必要、定期メンテナンスが必要といったことでも有名です。それゆえ、リレーはソリッドステートの制御形態の登場で消滅すると最初は思われましたが、反対に驚くような復活を見せています。

## 必要な改善点

- 設置コストが割高
  - 部品点数が多い、人件費がかかる
- トラブルシューティングが困難
  - 制御システムやプロセスの診断機能がない
- 改造や拡張が困難
  - 配線量が膨大
- オペレータインターフェイスが限られる
  - ランプや模擬パネルが考えられる
- 機能が限られる
  - アナログか、計算か、通信か
- 寿命が限られる

### 図B

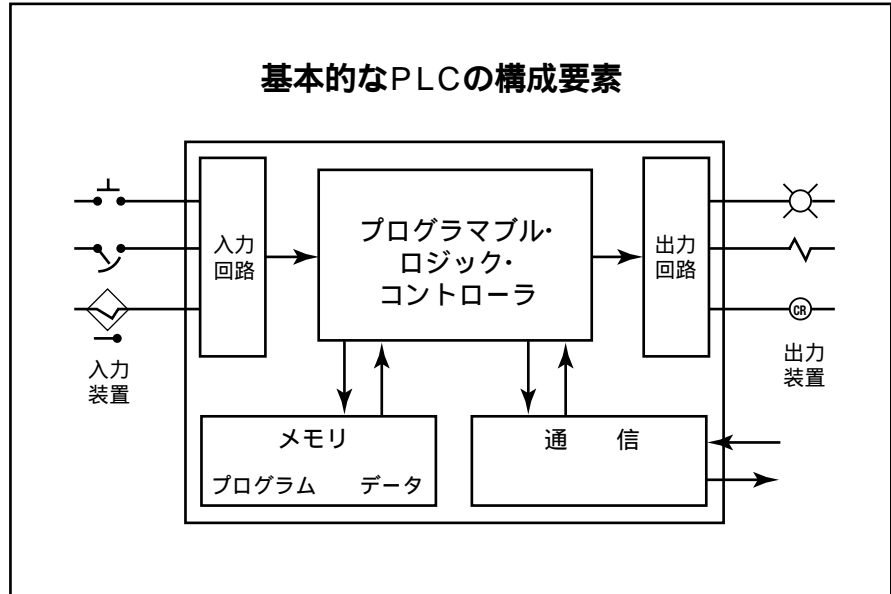
数多くの工場では20年前と同じリレーを今も使っていて、その性能にまったく不自由を感じていません。さらに、数多くの制御アプリケーションでは、分離やその他の専用電気機能用として、さらに洗練された制御形態と一緒にリレーを使用しています。CHOICESシリーズの将来号では、電気機械式デバイスの選択肢をさらに詳しく説明します。

**シングル・ボード・コントローラ** 最初のプリント基板を使った電子制御、つまりSBCは、1960年代初頭に登場しました。このような「論理モジュール」は、トランジスタ、コンデンサ、抵抗といった個別部品を基板上に配して作られていました。こうしたソリッドステートシステムは、消耗する可動部品がないため、根本的に高い信頼性がありました。一方、SBCの設置、操作、保守には、ハードウェアの初期費用を越える多くのコストが伴います。一般に、既製品として入手できないため、SBCのコストには、電気エンジニアがボードを設計して実用可能性をテストするサービス取得費が含まれています。

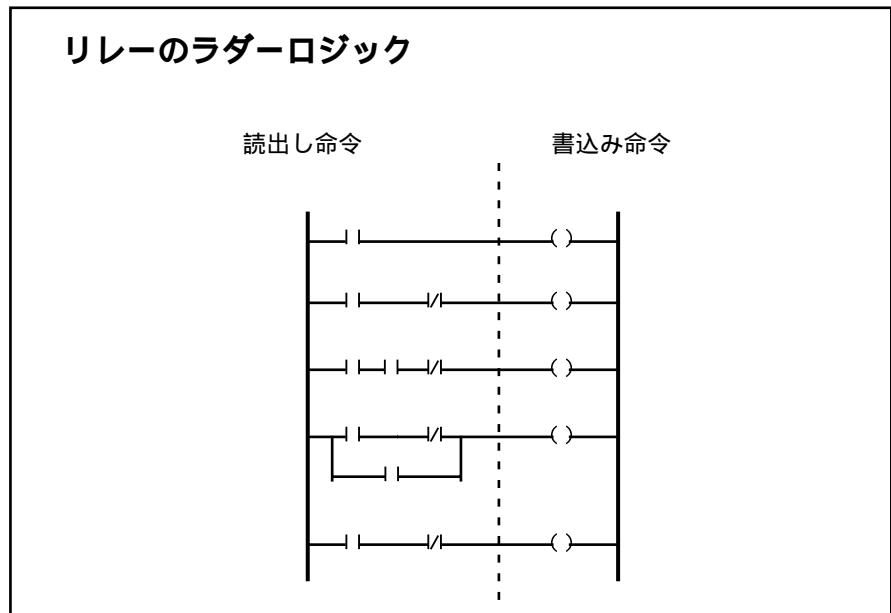
それでも今日でさえ、オリジナル機器メーカーの多くは、自社独自の装置アプリケーション用として、シングル・ボード・コントローラを設計開発する方法を選んでいきます。SBCは一般に特定の装置専用であることがきわめて多く、初期費用は割安にできます。装置に特定の能力が必要で、標準的な既製の制御製品を使っても実現できない場合には、申し分なく適しています。

**プログラマブルコントローラ** 最初のプログラマブルコントローラは1970年に発表されましたが、これは、コンピュータの柔軟性のあるソリッドステートシステムで、しかも工場のエンジニアや技術者がプログラムしたり、メンテナンスのできるものが欲しいというゼネラルモータズの要望に応じて開発されました。これら初期のプログラマブルコントローラは、それに置き換えられるリレー、カウンタ、タイマやその他の制御部品に比べて場所をとらず、プログラミングし直すことができる点で大変に柔軟性がありました(図C参照)。初期のプログラミング言語は、一般に電気技師が使用していたラダー

図と電気記号を用いたもので、プログラマブルコントローラが業界に受け入れられるうえで重要でした(図D参照)。



図C



図D

プログラマブルコントローラは、リレーのラダーロジックでプログラムできるので、電気図面をプログラマブルコントローラのプログラムに変換するのは比較的簡単です。変換プロセスには、各制御点の動作規則の定義、この規則のラダーロジックへの変換、出力の識別とラベル付け(アドレス指定)が含まれます。今日の職場戦力は、様々なエンジニアの混成であり、多少経験があつてラダーロジックをよく知っている人もいれば、コンピュータ中心でのプログラミングと制御のほうが快適な新しいエンジニアもいます。そのため、ユーザの背景知識とアプリケーションのニーズに基づいて、適用できるプログラミング技術の混成が求められます。

プログラマブルコントローラには、主に固定型とモジュラー型の2種類があります。固定型プログラマブルコントローラは、プロセッサ、電源、あらかじめ決まった数のディスクリットおよびアナログの入力と出力のある自己完結型ユニットとして納品されます。固定型プログラマブルコントローラには、拡張用の独立した相互接続式コンポーネントを伴うこともあり、小型、安価で取付けの容易なものです。一方、モジュラー型コントローラは柔軟性が高く、I/O能力、プロセッサのメモリサイズ、入力電圧、通信の種類と数量のオプションが用意されています。

本来、プログラマブルコントローラは、主にI/Oがデジタルの制御アプリケーションで使用されていました。本質的に、連続制御よりも、シーケンシャルや不連続アプリケーションに理想的でした。時間を経てアナログ能力と処理能力が追加され、プログラマブルコントローラはバッチ制御およびプロセス制御アプリケーションの有効なソリューションとなりました。

プログラマブルコントローラの発展により、コントローラは、それまで代替技術に依存してきた制御システムにとって魅力的な選択肢になりました。1980年代にマイクロPLCが導入されるまで、簡単な機械と単純なプロセスの自動化を進めるもっとも共通の手段はリレーとSBCでした。従来のプログラマブルコントローラの機能はアプリケーションにしばしば恩恵をもたらしましたが、コストは必ずしも見合いませんでした。コストが問題にならない場合でもサイズが頻繁に問題になり、小型のプログラマブルコントローラでさえ単純に大きすぎて、電子制御に割り当てた空間に収まらないことがありました。

リレーやSBCよりも効率のよいものとして、プログラマブルコントローラが小型装置需要を経済的に満たせるようになったのは、マイクロPLCが導入されてからでした。こうした固定I/Oコントローラは、一般に300US\$未満のパッケージで10~32のI/Oを処理できるように設計され、非常に小さいリレー盤に対しても有効な代品となりました。さらにこの低コストなコントローラが選択肢となったことで、多くの小型装置OEMメーカーに門戸が開かれ、過去には不可能だったところに自動制御を適用できるようになりました。たとえば、あるメーカーでは、抽選券の枚数カウント装置のコントローラとしてマイクロPLCを使用しています。

モジュラー型コントローラは、機能の点では固定型と同等ですが、コントローラからI/Oを物理的に分離しました。これによりI/Oラックを分散させてアプリケーションの近くに置くようになり、この場所から産業用ネットワークを通じてコントローラと通信しています。モジュラー型コントローラによって、ユーザは特定のアプリケーションのニーズにコントローラのコンポーネントを細かく一致させることができます。

リレー、SBC、マイクロPLC、フルサイズのプログラマブルコントローラのどれがアプリケーションに適しているかを判断する前に検討の必要がある要素は数多いですが、次の表は、この4種類がアプリケーションの基本条件をどう満たしているかを示しています(図E参照)。

アプリケーションの特徴	リレー	マイクロPLC	SBC	フルサイズのプログラマブルコントローラ
入出力	最大1~12系統	32系統まで	可変	32~1,000系統台
タイマ	可	可	可	可
加算/減算カウンタ	可	可	可	可
高速化機能	不可	可	可	可
データ計算	不可	可	可	可
データ取得	不可	可	可	可
通信	不可	制限付き	可	可
オペレータインターフェイス	原始的	可変	可変	可変
メモリサイズ	該当せず	1~10K	1~100K	2Mまで

**図E アプリケーションに対する制御方法の評価**

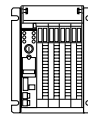
**PC使用の制御** プログラマブルコントローラのようなテクノロジーは、(無数のアップグレード、テクノロジーの進歩や発展形が登場してきたにもかかわらず)25年以上も市販され続けているので、「それに代わるものは何か」という当然の疑問が浮かびます。25年前にはリレーについて同じ疑問を投げかけましたが、リレーはまだ健在で精彩があります。となれば、もっと適切な問いは「ほかに何が必要か」ということでしょう。

PC使用の制御とは、通常ハードウェアやソフトウェアに埋め込まれている制御機能を制御プラットフォームへ適用しようという概念です。これには、制御エンジンだけでなく、プログラミング、オペレータインターフェイス(OI)、オペレーティングシステム、通信アプリケーション・プログラミング・インターフェイス(API)、ネットワーキングやI/Oまでも含む制御システムのあらゆる面が包含されています。ソフト制御とは、従来からのコントローラをソフトウェアに置き換えて、パーソナルコンピュータ上でプログラマブルコントローラの機能を実行できるようにしようという取り組みです。ソフト制御は、情報処理量の多い制御アプリケーションを扱う人には重要ですが、ソフト制御は、マイクロPLCとは別物としてではなく、従来のプログラマブルコントローラの実現形と捉えることが大切です。そうでなくてもソフト制御は「PC使用の制御」という巨大なトレンドの一部に過ぎないのです。

## 主要な属性：PLCとPC

### PLCの優位点

- メンテナンスコストが低い
  - 既存のインフラストラクチャ
- 制御に最適化されたハードウェアとOS
- ブート時間が短い
- 丈夫
- MTBF, MTTR
- 動作中の修理可能/ホットインサート
- I/Oバス統合デザイン
- ウォッチドッグの統合



### PCの優位点

- ハイエンドシステムでも購入コストが低い
- ビデオの統合
- 大容量のメモリとストレージ
- 豊富な市販ソフトウェア
- オープンネットワーキング
- SQLデータベースの統合
- 豊富な開発ツール
- 充実したオペレーティングシステム
- 大規模な産業R&D



図F PCとPLCの主要な属性

**分散制御システム** 分散制御システム(DCS)は、プロセス管理産業で発展した製品です。DCSは、1970年代中期にシングルループのデジタルとアナログのコントローラ、および中央コンピュータシステムの置換えとして開発されました。一般にDCSは、多重ループを処理できるユニットコントローラ、大量のI/Oを処理できるマルチプレクサユニット、オペレータやエンジニアリング用のインターフェイスワークステーション、履歴管理、通信用ゲートウェイ、先進の制御機能を1台の中央コンピュータにまとめたものからなります。これらはすべて完全に統合され、通常は通信ネットワークを通じて接続されています。

従来よりDCSのサプライヤは、テクノロジーとアプリケーションの専門技術を融合させて特定の課題を解決するという手法をとります。ファンクション・ブロック・プログラミングという非常に一般的なプログラミング法でさえ、開発者はこれを使って実際のプロセスやデータフローを模倣してシステムをプログラムすることができます。DCSを利用すると、プロセス内で信頼性の高い通信と制御が実現しますが、これはDCSが階層手法で制御を行ない、情報処理機能の大半を中央の産業用コンピュータに収めていることによります。この大規模なシステムは、比較的高価なコントロールの選択肢でもあります。

DCSに似たものといえば、メインフレームコンピュータとデスクトップコンピュータが好例です。最近まで、社内の情報処理をメインフレームコンピュータ以外で行なっている会社など聞いたことがなかったと思います。近年、ハードウェアとソフトウェアの能力が爆発的に高まったことで、多くの会社は現在、ネットワーク化した数多くの強力なデスクトップコンピュータで社内の情報処理をまかなっています。往年のメインフレームよりも強力であるうえに、柔軟でユーザフレンドリなネットワーク環境が、相当するメインフレームのコストの数割で実現します。

一般に、DCSはアナログとデジタルの比率が60:40よりも高く、実行する制御機能が複雑なアプリケーションに利用されています。DCSはプロセスが連続制御の産業に理想的です。面積的に広い領域へ大量のアナログ情報が分散されることが多く、主な支出はダウンタイムに関連するものです。この例は、パルプや製紙産業、公益事業、製油や化学産業に大変多く見られます。

DCSは、荷分けのような特定のアプリケーションを重視しているエンドユーザにとってはモデルソリューションですが、荷分けアプリケーションに加えて、プロセスに供給している50基の補助タンクがある場合には、エンドユーザはプログラマブルコントローラとDCSのどちらを利用するかを判断する必要があります。荷分けアプリケーションはDCSで制御するほうが優れていますが、50基のタンクはPLCで制御するほうが有利です。エンドユーザには両方を使用するという選択肢もありますが、そうするとプロセスのコストは大幅に増加し、エンドユーザは2種類の異なるコントロールを熟知し、メンテナンスすることを余儀なくされます。図Gは、両システムの経験による主な基本適性比較です。これは絶対的なものを表しているわけではなく、また、両者の境界線は、制御モデルの大半で内包されるため、あいまいです。今日、ハイブリッドコントローラの出現を目の当たりにしていますが、これは、小型から中規模のシステムの場合、従来のPLCとDCSの優位点の多くが得られます。

## DCSの主な適性

プロセス産業アプリケーション  
 シングルループ制御の置換え  
 複雑なアナログI/O  
 I/O  
 分解能の高いI/O  
 ファンクション・ブロック・プログラミング  
 集中制御  
 統合型OI  
 Windows NTまたはUNIX  
 オペレーティングシステム  
 制限のあるサードパーティ製デバイスのサポート  
 場合に依じた設定ツール  
 ベンダアプリケーション専門技術専用

ループ周回(>100ms)

## PLCの主な適性

不連続製造制御  
 リレーの置換え  
 デジタルおよび適度なアナログI/O  
 分解能が中程度のI/O  
 ラダー・ロジック・プログラミング  
 非集中および集中制御  
 周辺機器/アドオンとしてのOI  
 専用オペレーティングシステム

豊富なサードパーティ製デバイスのサポート  
 場合に依じた設定ツール  
 汎用およびベンダアプリケーション専門技術専用  
 ループ周回(<100ms)

## 図G DCSとPLCの適性

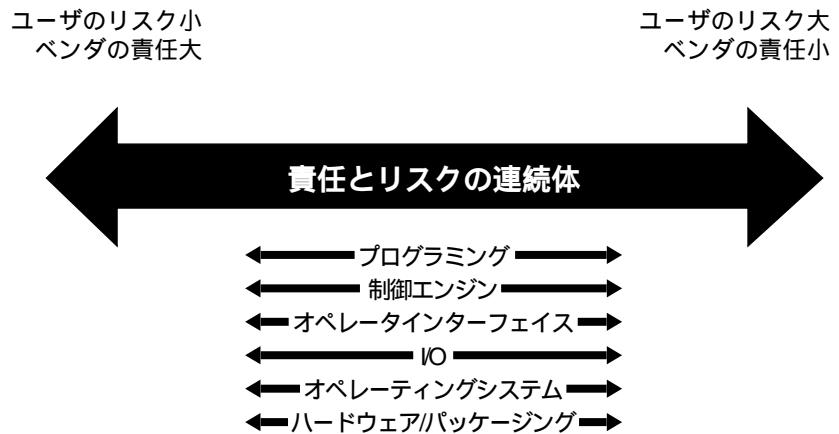
**コントロールの新しい定義問題** 前述のように、コントロールを定義する場合には、その機能ではなく、機能を実行する「箱」で定義する傾向があります。かつてはその方法が十分便利でしたが、今後も同じ方法でコントロールを定義し続けるとなれば、混乱が混乱を呼ぶこととなります。

プリント基板を電子リレーで埋めるとすると、これはリレーによる制御ですか、それともシングルボード制御ですか。プログラマブルコントローラはプリント基板を一例で使用するので、これらはプログラマブルコントローラですか、それともシングルボード制御ですか。ソフトコントローラは、従来のプログラマブルコントローラと同じ設定ツールとI/Oインターフェイスを使用することができますが、これはPC使用の制御ですか、それとも単にPLCの発展形ですか。現在のPLCはファンクションブロックでプログラムでき、アナログI/Oを処理する能力があります。となれば、これをパッチ制御のようなプロセスアプリケーションに使用している場合、これはプログラマブルコントローラですか、それとも単にDCSの発展形ですか。同様に、今ではファンクション・ブロック・プログラミングやアナログI/Oを市販のパーソナルコンピュータで実行できます。これはPCですか、それとも今やDCSであるということですか。DCSサプライヤは、UNIXオペレーティングシステムを離れ、次期製品をMicrosoft Windows NTで構築していますが、これは各ベンダがDCS事業をやめてPC使用のプロセス制御に移行しているということですか。



連続体の右側は、複数のベンダオプションを表す選択肢なので、結果としてユーザが大きいリスク負担を求められます。これのすべてが悪いわけではありません。特に、ユーザが自分の経験やスキルに自信を持っており、社内やサードパーティを通じて統合、カスタマイズ、サポート責務の大半を実行できる場合、リスクを進んで受け入れることと引き換えに複数ベンダ環境の柔軟性が得られます。

責任とリスクの連続体は、プログラミング、制御エンジン、OI, I/O通信インターフェイス、オペレーティングシステム、ハードウェア/パッケージングを含むコントローラの全機能に適用できます。ちなみにこの連続体は、サポートの種類、サプライヤ窓口のレベル、「オープン度」の程度、新技術に関する情報の受け取り方法などといったあまり具体的でないその他の属性にも適用できます(図J参照)。

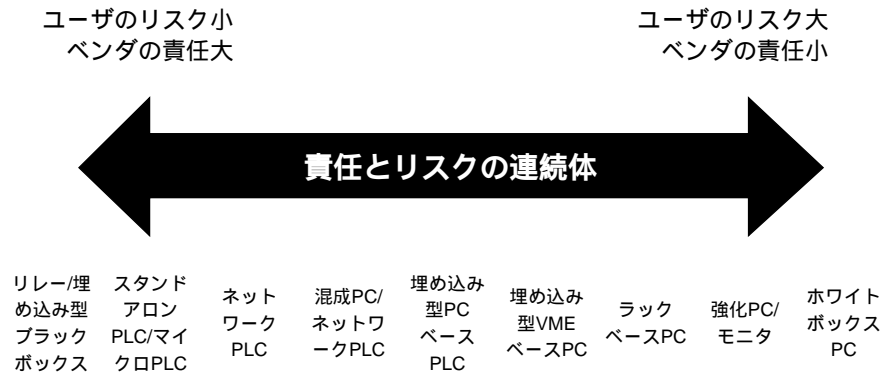


図J 責任とリスクの連続体の拡大

ハードウェアプラットフォーム/パッケージング これ以外の要素のほとんどは、それが動作するハードウェアというコンテキストの範疇で説明できるので、これが合理的な出発点です。ところが現在、ハードウェアではなく、ソフトウェアがコントローラの選定動機となる時代へと急速に移行しています。この状況は、市販のコンピュータ業界と同じように比較できます。IBM互換のコンピュータかマッキントッシュのどちらを買うかを決めてから、互換性のあるソフトウェアを買うのが常識でした。今では、ソフトウェアはハードウェアと同じくらい全体の意思決定で重要になっています。

図Kは、この連続体をコントローラに当てはめたものです。責任とリスクの連続体の一番左端はリレーと「ブラックボックス」です。どちらの場合も信頼性、メンテナンス性、統合に対してほぼ全責任をベンダが負います(結局、リレーよりも統合度を上げることはできません)。どちらの製品も、アプリケーションに特に合わせて動作させるのに作業はほとんど、あるいはまったく必要なく、単に据え付けて、電源を入れて使うだけです。これらのシステムから少し右へ行くと、各種のプログラマブルコントローラがあります。ど

れもすべてベンダ固有のパッケージングを採用し、一般に長年にわたる研究と経験に基づくカスタムASICとファームウェアを搭載しています。ここでもベンダがコントローラ設計の責任のほとんどを負っていますが、納入先施設でのプログラミングとアプリケーションの面からユーザは柔軟性を獲得しています。



**図K ハードウェアの責任とリスクの連続体**

右端から多少遠ざかると、オープンシステムと専用システムの両方の長所がある混成手法があります。この種のソリューションでは、ネットワークに組み込まれたPLCがプロセスの直接制御を担い、産業用PCがHMIやPLCプログラミング、データロギング、通信機能を担います。このソリューションは、オペレータが膨大なプロセス生産用変数を入力し、大量のデータを記録することが必要なアプリケーションに理想的です。純粋なオープンシステムでは、産業用コンピュータがシステムの頭脳の役割を果たし、HMIタスク、データ処理、通信を処理するだけでなく、実際のアプリケーションも制御します。製造プロセスが頻繁に変わるために最大限の柔軟性を必要とするアプリケーションにとっては、オープン制御システムは理想的であり、容易にアップグレードが可能です。たとえば、エンドユーザは新しい制御ソフトウェアにアップグレードでき、PCのメモリを増設したり、拡張シャーシの追加によってPCをアップグレードして追加の通信用オプションカードに対応させることができます。

少し右へ行くとラックベースのコントローラハードウェアがあり、ベンダ固有のハードウェアではなく、市販のASICとマイクロプロセッサを使用しています。しかし、これらのシステムは、一般にプログラマブルコントローラと同じパッケージングを使用しており、厳しい製造環境に対応した丈夫なI/Oとデザインを利用しています。好例はアレン・ブラドリーのオープンコントローラです。一見プログラマブルコントローラですが、今日のパーソナルコンピュータの大半に見られるものと同じプロセッサ技術を搭載しています。ユーザがこの種のシステムを採用する場合、ベンダはハードウェアの設計、パッケージング、信頼性に責任を負っていますが、ユーザは、プログラミングだけでなく、オペレーティングシステムとアプリケーションソフトウェアの選定にも責任を負っています。複数のベンダがモジュールを提供する

オープン・バス・システムでは、ユーザの選択肢が増える一方、ベンダの責任が少なくなります。この例には、VME, Multibus, PCI, PC-104があります。

責任とリスクの連続体の一番右端は、「ホワイトボックス」のパーソナルコンピュータです。これらはもちろん市販のコンポーネント(Intel® Pentium®など)を使用し、同様にほぼ間違いなくMicrosoft® Windows® NTやWindows 95を搭載しています。このコントロールのシナリオでは、エンドユーザは様々なベンダのコンポーネントをミックス・アンド・マッチできます。しかし、この選択肢の場合は、エンドユーザが優秀なシステムインテグレータのどちらかが制御システム全体の責任を負うこととなります。コントローラとして市販のコンピュータを採用することは、ユーザが既製コンポーネントを使用する道を選んだのであり、社内で統合とカスタマイズを行なうことにほかならないので、全体のシステムと信頼性に対して負担するユーザの責任の度合いが大きくなります。

ホワイトボックスPCの初期投資費用は、産業用PCに比べて非常にわずかで済むことがあります。ホワイトボックスPCの長期メンテナンスおよびサポートは、ダウンタイムと置換えの面から非常に高額になることがあります。さらに、環境コントロール型のエンクロージャのような追加品は、購入費が高額で維持費もかかります。PCのライフサイクルを通してみると、こうした隠れた追加コストは、ホワイトボックスPC購入時の節約分を差し引いても余りあることがあります。また、アプリケーションにうまく適合しないPC使用コントロールソリューションの失敗は、財務問題だけで終わらない恐れがあります。ある種の危険を伴う制御アプリケーションでは、安全性の問題にもなりかねません。一方ベンダは、強化PCプラットフォームのように、提供するテクノロジーをあらかじめパッケージングしてユーザのリスクをできるだけ減らすことができます(図K参照)。

*オペレーティングシステムの選択肢* ハードウェア/パッケージングの選定と密接に結びついているものにオペレーティングシステムがあります。かつての「事務所におけるWindows対マッキントッシュ」論争に似た論争は、製造事業のコントローラに関しても巻き起こっています。オペレーティングシステムとは、簡単に言えば、プロセッサチップ、メモリ、周辺装置間の情報の流れ方を定義するソフトウェアです。プログラマブルコントローラや分散型コンピュータシステム用のオペレーティングシステムは、一般にベンダが開発し、会社特有の品揃えやアプリケーションに合わせて最適化します。こうした場合、オペレーティングシステムはファームウェアに埋め込まれ、そのベンダ専用となっています。

今日のプログラマブルコントローラのオペレーティングシステムは、ハードウェアプラットフォームと同様に25年以上の発展の成果であり、プラットフォームで必要とされる判断材料、産業界で鍛えられた設計、再現性と信頼性を提供します。かつて、このような目的を達成することといえば、ほかでもなくベンダ固有のオペレーティングシステムを選定すること、またそのベンダのコントロールソリューション全体をも選定することでした。これはリスクを回避できる場合には長所ですが、社内で大幅なカスタマイズや統合が必要な場合には障害になることがあります(図L参照)。



図L オペレーティングシステムの共通利用度

連続体の右側に該当するハードウェアプラットフォームを選ぶほど、オペレーティングシステム選択の幅が広がります。さらに、市販のPentium級チップでは、QNXやVertexのようなリアルタイム・オペレーティング・システム(RTOS)などと呼ばれるものと、Microsoft Windows NTのような市販のオペレーティングシステムを選ぶことができます。

RTOSは、製造産業や電気通信産業に見られるような特に高信頼性を求められるアプリケーション向けに開発されたオペレーティングシステムです。これは、ユーザの関心がシステムの信頼性にあり、犠牲となる柔軟性や市販のハードウェアで得られるコスト面の利点にはそれほど関心がなかった状況では大人気でした。リアルタイム・オペレーティング・システムには、一般にそのベンダの製品専用のプログラミングツール基本セット、および専用の記述またはアドオンとしての購入が必要な、サードパーティ製周辺機器との通信ドライバが付属しています。

ベンダの中には、RTOSを採用し、アドオンの設定ツールとドライバのセットを開発し、特定産業向けに最適化した開発環境として提供することで成功を収めたところもあります。これは電気通信産業に納品しているサプライヤの間ではきわめて一般的であり、ControlwareやQNXのような企業がオートメーションコントロール業界でも実施しています。こうした製品の代表的な購買層は、市販のハードウェアプラットフォーム上でコントロールを実行したいけれども、高い信頼性とセキュリティが求められる制御システムの特定の専門技術があるOEMメーカーです。プログラマブルコントローラのパッケージングで市販のコントローラハードウェアを購入することを決めた人が、

業界向けに最適化されたRTOSを選定してその上で実行させるのは珍しいことではありません。基本的な必要条件は同じなのです。

Microsoft Windows NTのような市販のオペレーティングシステムは、Intelベースのハードウェアプラットフォーム上で動作するコントロールの有効な選択肢として、瞬く間にシーンに登場しました。Windowsの初期バージョンは、制御アプリケーションに使えるほど安定しているとは思われませんでした。しかし、Windows NTの登場によって、情報集約的な制御アプリケーションのユーザにとって有効な選択肢であると、この市販オペレーティングシステムを擁護するコントロールベンダが増えています。加えて、業界標準のオペレーティングシステムでは、ユーザは複数のベンダの豊富な開発ツールを入手でき、すべて共通の環境で動作させることができます。

それでもNTの安定性とリアルタイム制御に対する適性については議論があります。こうした議論の中で、リアルタイムという言葉はしばしば相対的なものです。数社がNT用リアルタイムエクステンション(RTX)というものを開発し、NTのリアルタイム能力が改善されると主張しています。こうしたエクステンションによって改善される性能の程度、また、NTのデータ処理および統合能力に与える影響の程度は、まだ議論的のです。ロックウェル・オートメーションの調査によれば、無修正のWindows NTは、広範な制御アプリケーションに向けた信頼性あるオペレーティングシステムであることがわかりました。NTのコントロールでの使用法について詳細は、ロックウェル・オートメーション白書『Windows NT for Soft Real-Time Control - Separating the Fact From the Fiction』を参照してください。発注方法については、28ページを参照してください。

**プログラミング法** かつて、プログラミング法は、プログラムする「箱」で直接決まりました。リレーは「ソフトウェア」によるプログラミングを必要とせず、ロジックを直接配線します。SBCには一般にエンドユーザによる外部プログラミング手段がないので、使いやすくなっています。プログラマブルコントローラには、特定ベンダのプログラマブルコントローラ用に設計されたラダーロジックによるプログラミングパッケージが常に使用されていました。言い換えれば、アレン・ブラドリーのプログラミングパッケージをGEやFanucのコントローラで使用することはできませんでした(その逆もできません)。DCSシステムには、ベンダ固有のファンクション・ブロック・プログラミングが使用され、一方、パーソナルコンピュータには、一般にMicrosoft BasicやPerlのような上位言語が使用され、今日ではCやC++, Java, Visual Basicが使用されています。

現在、多数の相互補完が見られます。ラダーロジックに加え、プログラマブルコントローラの多くは、フローチャート(SFC)、ファンクションブロック、構造化テキストプログラミング法を使用して設定することができます。DCSは今でもほとんどの部分をファンクション・ブロック・プログラミングで設定しますが、一部のDCSはラダープログラミングのオプションも用意しています。DCSベンダの多くは、Windows NTや95の普及に乗じて、使い慣れた業界標準環境でのファンクション・ブロック・プログラミングを可能にしています。結局、特にプログラミングに精通しているのであれば、ユーザはVisual Basicのような標準的な市販開発パッケージを使って自分のプログラムを設計する道を選ぶこともできます。

コントローラの中には、方法を組み合わせてプログラムできるものさえあります。たとえば、アプリケーションが主に不連続である場合、ユーザがアプリケーションをラダーロジックでプログラムできるよいチャンスがあります。一方、プロセス指向のアプリケーションの細部については、ユーザは適当な個所にファンクションブロックを埋め込むことができます。実際、現在数社より、各種のオープンおよび専用のプラットフォームをプログラムできる各種エディタ(ラダー、SFC、ファンクションブロックなど)が提供されています。

では、どのように選べばいいでしょうか。繰り返しますが、それはアプリケーションによって異なります。しかし、責任とリスクの連続体でプログラミング法を見ると、選択肢が明らかになります。選択したプログラミング法が特定のコントローラに最適化されていることが最大の関心事であれば、連続体の左側の手法を選択しようとするのもよいでしょう。一方、それよりも「一般的」なプログラミング法を探しており、状況に応じてそれを修正できる自分の能力に十分安心できる場合には、連続体の右側にある手法を選ぶことができます(図M参照)。

**ロジックエンジン** ロジックエンジンそれ自体が基本的な選択肢で、「コントローラかDCSかPCか」だけではなく、まさに「オープンか専用か」の選択となります。ユーザは、ロジックエンジンによって残る制御システムがどのように統合されるかを判断する必要があります。

ユーザはオンラインエディタが必要ですか、それともオフラインですか。必要なのは固定機能とデータ構造ですか、それともユーザ定義による機能と構造を持つエンジンが好ましいですか。

ロジックエンジンの利用可能な機能によって、ほかの制御システムコンポーネントとの統合度が決まり、ついにはそれが動作するオープン度が決まります。高い統合度と大量のデータアクセスを要するユーザは、オープン度の高いロジックエンジンの選択肢を選ぼうと考えることでしょう。



図M プログラミングの責任とリスクの連続体

マン・マシン・インターフェイス マン・マシン・インターフェイスは非常に広範な話題なので専用の紙面が必要なため、本紙ではその選択肢についてあまり時間を割かないことにします。きわめて単純なコントローラでさえ、単純な押しボタンパネルであっても、産業用またはホワイトボックスのパーソナルコンピュータ上で動作する複雑なソフトウェアパッケージであっても、ある種のオペレータインターフェイス装置が必要であると言うにとどめておきます。その中間には豊富な選択肢があり、ユーザが進んで受け入れる責任とリスクの程度、およびオペレータの能力とトレーニングに基づいて一つ一つ評価できます。

## 情報を生かして意思決定するための重要な検討事項

さて、コントローラだけでなく、コントローラ内部の機能にも選択肢が揃っていることを確認しました。どの選択肢が自分に適しているかを判断する場合には、検討する要素が数多くあります。

オープン度 CHOICES白書の最初で説明したように、制御システム内にはオープン度の違いがあります。IEC 1131.3に定義されているように、ベンダ固有のコントローラにはオープンプログラミング法があります。アプリケーションを市販のハードウェア上で動作させながらRTOSを使用することができます。Windows NTの動作するホワイトボックス(マイクロ：広範なホワイトボックスPC)のパーソナルコンピュータまたは産業用コンピュータを使用しながら、次にベンダの専用制御ソフトウェアとI/Oシステムを選ぶことができます。

「オープン」なコントローラを選ぶことが目標ではありません。そうではなく、共通利用度、信頼性、アプリケーションの柔軟性、サポートとコストの必要条件を満たすコントローラを選びます。PC使用のエンジンを検討する場合には、「制御を除くその他のソフトウェアをこの装置で実行したいか」と問いかけてみます。オープン度は、それ自体で終始するのではなく、こうした基準の関係を見て評価します。

必要条件の基礎である速度、スケール、パッケージング、信頼性、周辺機器は、これらを必ず満たしてから制御システムのテクノロジーを割り当てる必要

があります。これらの必要条件の中には、産業セグメント固有のものもあり（プロセス、モーション、不連続性）、一方、産業用コントロールのあらゆる面に広く適用できるものもあります。

オペレーティングシステムは、あらゆるオープン・アーキテクチャ・ソリューションの中心にあります。そして、オペレーティングシステムについて語ると、感情的な反応を招くことがあります。オペレーティングシステムとプログラム開発の必要条件の大半は、既存の市販オペレーティングシステム製品で十分満足されており、市場に幅広く認められています。Windowsベースのソリューションが近い将来、こうした機能に望ましいものとなり、認められるのもまた予測できる結末です。

**速度** 機械制御やプロセス制御のプラットフォームとして汎用の市販製品が適用できるかについても論議は少なくありません。制御システム市場では、「リアルタイム」という言葉が非常に事由に使われていますが、リアルタイムとは、トランザクションやイベントをファイルに保存して後からバッチ処理するのではなく、できるだけ迅速かつ予測的に処理することです。したがって、リアルタイム処理がどれほど迅速になるかはアプリケーションによって決まります。

たとえば、ATM(Automated Teller Machine)のリアルタイムは秒で計測できますが、モーション・コントロール・システムのユーザは、ミリ秒の分数でのリアルタイム応答を必要とします。この範囲の広い性能条件は、動作環境にあるコントローラについて語る場合、第一に検討を要する領域です。どれほどの速度がコントローラに必須でしょうか。その再現性はどれほどでなければならぬのでしょうか。

コントローラやDCSを解剖すると、これらもまた、ハードウェアプラットフォームとオペレーティングシステムを中心に構築されていることがわかります。産業用制御プラットフォームのオペレーティングシステムは、パーソナルコンピュータのハードウェアとソフトウェアよりも実質的に専用性が高いので、これらもまた、サポートしている市販オペレーティングシステムよりもきわめて注目度が高くなっています。多くの点で市販製品と似ていますが、コントローラ用オペレーティングシステムにも非常にはっきり異なる特徴があります。

自分の制御アプリケーションの範疇で速度をどう捉えるべきか決めることが大切であり、仕様のために仕様を評価する行為に陥らないようにすることです。

**信頼性/耐環境性の高い運用** 産業用オートメーションシステムは、従来より実質的にライフサイクルが長く、システム性能の向上やソフトウェアの拡張を継続的に行なうことは求められていません。多くの場合、システムの寿命が続く間にわたってスペアパーツを交換することが唯一の必要条件となり

ます。それに対して市販品では、新開発やアップグレードによって、少なくとも24ヶ月ごとに移行を強いられます。

ここから数多くのハードウェア問題が発生します。選定したハードウェアは、自分の製造環境の過酷さに耐えられますか。あるいは、6ヶ月ごとにコンピュータをシャットダウンしてアップグレードしなければならないことを予定していますか。環境が清浄でダストがなく、騒音が少ない場合には、まさに過酷な製造作業向けに設計された製品を購入することは有意義ですか。

多くのベンダは、通常のオフィス用、つまり「ホワイトボックス」PCをPC使用の制御システムに使用できるという考えを支持し、エンドユーザにすべてのPCから同じ「サービス」が得られると信じさせていますが、この説明は必ずしも正しくありません。つまり、産業界のプラントフロア環境を扱えるだけの強さを備えた「ホワイトボックス」PCは非常に少ないのです。プラントフロア用PCに課せられた要求は、信頼性、柔軟性、軽快性、システムアップグレードの容易さ、強靱な運用といった標準的な動作条件を上回ることを求めています。

産業用コンピュータは、すべてプラントフロアの要素となりうる空気中の汚染物質、振動、衝撃、過酷な温度の攻撃に耐えられるように設計されています。産業用PCの多くはTFTフラット・パネル・ディスプレイ、ショック搭載のハードディスクドライブ、動作部品のないソリッドステートのフラッシュ・ハード・ドライブ(メモリ)、フィルタ付きボールベアリング式冷却ファン、内部温度センサ、無停電電源、冗長ハードドライブやシステム・コントロール・モニタ・カードを採用しています。さらに、CRTディスプレイ内蔵の産業用PCの多くは、電気障害でディスプレイ問題が発生しないように、EMI(電磁波障害)シールドを強化して設計されています。こうした一連の改善点は、PLCベースの専用制御システムの強さをPC使用の制御システムに展開できるよう設計されています。

一部のアプリケーションは産業用の強化コンピュータを必要としない、と主張できる場合もあります。たとえば、プラントフロアから遠く離れたコントロールブースで動作するコンピュータや、プラントフロアでも環境コントロール型のエンクロージャに入っているものは、強靱である必要はほとんどないように思われます。オフィスグレードのPCは産業プロセスを処理できますか。多くの場合で答えはイエスです。では、これは常に適切なソリューションですか。信頼性と強さの点からはおそらくノーです。今はハードウェアとパッケージングの選択肢が揃っているので、ソリューションをアプリケーションに細かく一致させることが可能といえます。

ただし、信頼性と耐環境性の高い運用は、実配線されたものだけの問題と捉えないことです。オペレーティングシステムの点について言えば、数多くの実践面で疑問があり、評価が必要です。市販オペレーティングシステムの場合は、18ヶ月ごとにアップグレードすることが慣例になっています。また、次のリリースまで修正できないことがきわめて一般的です。制御ソリューションはこの程度のサポートで間に合いますか。それでよければ使用してもよいでしょう。そうでない場合には、おそらく別の方法が適しています。

これは、ハードウェアとソフトウェアのベンダが共同で分かち合う必要条件と捉え、オープン・アーキテクチャ・システムのライフサイクルにわたって互換性を維持する必要があります。産業界として18~24ヶ月ごとのシステムおよびソフトウェアのアップグレードを進んで受け入れられる場合には、市販製品を十分採用することができます。

これはオペレーティングシステム決定の際の中心です。すべての制御アプリケーションに市販のオペレーティングシステムを利用することが理想とは思いますが、それを動作させなければならないアプリケーションの重要な必要条件を忘れることはできません。制御用オペレーティングシステムは、オペレータとのやりとりの範囲に限られることはなく、電気的および機械的条件に連続的またはリアルタイムな応答を処理できる必要があります。

*アプリケーション：不連続/プロセスの連続体* ベンダの多くは、プロセス制御は不連続制御に比べて必ずしも難しいものではなく、異なるだけであることに気づきました。最初のプログラマブルコントローラは、不連続製造産業(特に自動車)で誕生したので、初期の製品が不連続製造者のニーズに合わせられていたのはごく自然なことでした。しかし、コントローラやI/Oモジュールの能力が高まると(メモリ容量増大、一致能力の向上、命令セットの拡充)、コントローラの利便性は不連続産業を越え、プロセスアプリケーションの長い列に広まりました。加えて、専用I/Oモジュールが登場すると、コントローラは、温度制御、重量計の荷重セル、プラスチック注入成形、高速アナログといったプロセス機能を扱うことができます。

コントローラ互換のプロセス機能ライブラリ、拡張I/O機能(ハードウェアとソフトウェアの両方)の出現と高機能ネットワークの登場によっても、プロセスアプリケーションでのコントローラの卓越性が生まれました。

梱包や材料処理のようにプログラマブルコントローラが成熟に達したアプリケーションも多少ありますが、その他の多くでは成長傾向を示しています。たとえば、マイクロPLCは、小型であることと低コストがOEMやエンドユーザーのアプリケーションに理想的なので、市販品市場と産業アプリケーションへ急速に拡大しています。

プログラマブルコントローラもまた、一般に、広範なアプリケーションに採用されていますが、特に食品や薬品処理、水/排水、石油やガスのパイプラインといったプロセス制御およびSCADA遠隔測定型のアプリケーションで顕著です。こうした移行は、現在のコントローラにプロセスツールのキット一式と遠隔測定アプリケーションに求められる通信機能が備わっているために実現しました。

ハイエンドとローエンドのコントローラの発展はまったく異なります。SCADAアプリケーション(水/排水、石油抽出など)の製造者を中心とするプロセス製造者の多くは、コスト、信頼性、使いやすさの面から、専用のリモート・ターミナル・ユニット(RTU-小型で比較的低価格のSBC)を小型のプログラマブルコントローラに置き換えることを考えています。これは、こうしたシステムのネットワーク能力が向上した近年可能になったばかりです。

ハイエンドでは、それまでDCSがシステムの選択肢だった小規模バッチ作業のような場所で、大容量のメモリとI/O機能を持つ大型PCとプログラマブルコントローラが普及しつつあります。一方、DCSシステムもスケラビリティの向上やI/Oコストの低減によって発展しました。さらに、一つの制御システムに異なる機能を生産的に組み合わせる機会も増えました。その結果、DCSとプログラマブルコントローラの境界線はあいまいになりました。

*据付け/統合* コントローラの設置面積は、ここ数年の間に数百パーセントの割合で減少しました。顧客の用意できるパネル用空間は少なくなり、さらに重要なことに、余分な機能に費用をかけなくなりました。コントローラである決められた問題を解決したい場合、その問題の解決に最適化されたコントローラを求め、それ以外のアプリケーションの主だった問題は除きたいと考えています。言い換えれば、四則演算だけをしたいのに科学計算機は買いたくないというわけです。

コントローラのサイズに影響を与えたもう一つの重要な要因が、分散化と「細分化」の進んだI/Oソリューションの出現です。ある特定の機能の物理的パッケージは変化しており、統合プロセスを簡素化するうえで役立っています。たとえば、かつてのI/Oシャーシは、エンクロージャに収め、端子片に配線し、さらに装置に設置してから相互接続を行ないました。現在は、I/Oが産業規格化された、装置に直接ボルト止めするだけのあらかじめパッケージされたソリューションがユーザの多くに好まれています。

さらに分散化が進み、費用効率のよい制御ソリューションも出現しました。このトレンドに後押しされて、コントロールのメーカーは、I/Oをアプリケーション現場の近くに配置できるモジュラー化されたアーキテクチャを提供するようになりました。このアーキテクチャは配線が少なく、制御システムに継続して最大限の使いやすさを追加できる柔軟性を備え、I/Oがパネル空間を占領しないので、エンドユーザはこれに向かいました。

メーカーは、コントロールソリューションの観点から全体的な必要条件を理解するため、顧客と密接に協力しています。一般に、プログラマブルコントローラのメーカーは、全体システムを構成する製品すべてを供給してはいません。システムは、単なる制御ハードウェアに収まらず、統合したコンポーネントでもあります。自動車工場のロボットコントローラと溶接制御システム、食品や薬品産業のプロセスアプリケーションの重量計、遠隔測定アプリケーションのモデム、あらゆるアプリケーションに用いられるソフトウェア。これらはすべて既存のプラント・フロア・コントロールやコントローラとシームレスで統合される必要があります。これはたいていの場合、ネットワークインターフェイスを開発し、その製品をコントローラと直接通信できるようにしたサードパーティによって実現されています。

顧客は、周辺装置との統合度にも、従来プロセッサとI/O間で達成されていたのと同程度を望み、同程度の使いやすさとセキュリティも求めます。たとえば、ソフト制御は、現在手元にあるプログラマブル・コントロール・モデルをPCハードウェアに採用することに関心のあるプログラマブルコントローラのユーザには適したソリューションです。

メーカーは、コントローラの補完製品を提供するため、サードパーティと密接に協力しています。制御アーキテクチャに合わせて十分定義された一連のインターフェイスを開発すれば、顧客は、システム内の統合度を高める道筋が得られます。

**通信性能** ほとんどの制御システムには、情報、制御、デバイスの3つのレイヤが存在します。情報レイヤとは、コントロールと組織管理システム(MES, MIS)とのリンクです。一般に、この情報レイヤにはイーサネットネットワークが用いられ、プログラムのメンテナンス、生産レポート、システム全体のデータ収集といった機能を担っています。

制御レイヤでは、一般にコントローラがI/Oラックや、可変速ドライブや専用マン・マシン・インターフェイス・システムなどのI/Oデバイスを監視しています。この領域では、リアルタイム応答とデータを理路整然と送る機能が重要なドライバです。このレイヤは、あとの2レイヤを結ぶ接着剤です。

デバイスレベルでは、既存の実配線に対するサイズ、コスト、拡張された機能がドライバです。デバイスネットワークでは、配線が減少し、先進の診断機能とその他の生産性が得られます。デバイスネットワークは、診断機能が埋め込まれたセンサ、I/O、プロセッサ、ソフトウェアのような製品から送られる診断メッセージを捕捉します。ネットワークは3つのレイヤすべてに必要であり、それぞれが高い性能を提供するだけでなく、ほかのレイヤからネットワークで統合された場合には、統合度の高い高性能システムを創出します。制御システム、つまりコントローラは、この3つのレイヤで表されるネットワーク機能を提供する必要があります。制御手法では、スループットを入力点から出力点までの時間を逆算して測定する必要があるように、システムレベルのスループットは、デバイスネットワーク上のノードから情報ネットワークのノードへデータを送るまでの時間と考える必要があります。

エンドユーザは、コントロールと情報システムを費用効率のよい方法で組み合わせるための理想的な方法が欲しいとメーカーに依頼しています。在庫管理、メンテナンス、その他の工場製造作業を制御するシステムのネットワーク化は、リアルタイム情報の需要が増加するにつれて発展しています。製造作業およびサポート業務の実施方法と各種アプリケーションでのデータ処理方法の性質の違いのため、ほとんどの施設で複数のネットワークが必要です。

**安全性/データの整合性** 製品レベルおよびシステムレベルでの診断は重要な必要条件になりつつあり、それによってオペレータは多くの情報を利用できるようになります。「リミットスイッチ1が故障」というようなメッセージによって、故障箇所を探すのに要する時間を節約できます。診断機能の一部は、ファジー理論を埋め込んだセンサのような、コントローラと併用する周辺製品から送られます。しかし、一部の診断機能は、I/Oラック・タイプ・コントローラ、さらにはソフトウェアまでも含めて根本的に共有させ、利点をすべて享受できるようにする必要があります。工場内でのデバイスネットワークは、デバイスとコントローラとの情報共有をすばやく実現します。

エンジニアは、効率のよいプロセス設計ツールを使って、技術開発の生産性を高めています。シーケンシャル・ファンクション・チャートや構造化テキストのような時間節約につながるツールが、従来のリレーのラダーロジック設計ツールと併用されています。デバイスレベルのネットワークは、診断機能の向上、ライフ・サイクル・コスト、配線コスト削減といった利益を顧客に直接もたらすので、市場で重要なトレンドとなることでしょう。

**コスト** コントローラメーカーは、従来のコントロールソリューションで設定された価格点を継承するかそれを打ち破る必要があります。たとえば、SCADAアプリケーションの会社の大半は、専用RTUを使用していました。

プログラマブルコントローラへの移行を促すためには、メーカーはRTUで設定された価格点に一致させなければなりません。

しかしながら、顧客は購入価格を越えてライフ・サイクル・コスト全体に注目する必要があります。かつて自動化投資は1回きりの購入と考えられていましたが、今では、メーカーは自動化投資ライフサイクルの各種の段階を検討しており、各段階に付加価値を与える複雑な方法を見出しています。メーカーと顧客は、初期購入価格を越え、オートメーションシステムの正当性評価、購入、据付け、操作、メンテナンス、アップグレードに伴うコストを低減する方法を把握しようと努める必要があります。

エンドユーザのプロセスが、時間をあまり重視しない情報の収集と管理だけに関わる場合には、PC使用の制御は費用が少なく済みますが、プロセスに時間を重視する情報やプラントフロアの死活に関わるような変数も含まれている場合には、PC使用の制御も従来のプログラマブルコントローラとあまり変わらない金額になることがあります。たとえば、プラントフロアに高周波や電磁波による障害、温度変動やその他の環境要因がある場合、エンドユーザがプロセスで要求されるあらゆるコンポーネントやシステムレベルの強化でPCを保護し終わる頃には、プログラマブルコントローラのほうが費用が少なくなっていることもあります。

標準のプロセス・ルーチン・ツールがあると、トレーニングコストが削減でき、オペレータがPC使用の制御を早く受け入れられるようになります。こうしたパッケージのいくつかにオートチューニングを組み込むと、ループメンテナンスが簡素化されます。さらに、Visual Basic, C++, Active Xコンポーネントなどのような標準的なツールで開発されたシステムでは、開発者は、非常に価格競争の激しいIPC市場で使用されている比較的安価なツールを利用できます。

DCS, パーソナルコンピュータ、プログラマブルコントローラのどれを使用したシステムを購入するかを判断する場合、日ごとに能力が伯仲しつつあるなかではコストが大きな要素です。I/Oからユーザインターフェイスまで価格差は千差万別です。一般的なDCSのユーザインターフェイス(専用オペレーティングシステム)は50,000ドルかかることがあります。既製のWindows NTオペレーティングシステム上で動作する既製のOIソフトウェアを実行する一般的な産業用コンピュータのユーザインターフェイスは、7,000ドル未満で購入できます。一方、マイクロPLCはわずか300ドルではありません。

## 結論

オートメーションコントローラには、今日あらゆる選択肢が揃っているのに、その情報を生かして意思決定するには、「箱」の奥を見て、その中心機能をアプリケーションに当てはめてどのような答えが出るかを考えることが求められます。製造プロセスを決定付けるトレンド、事業の目標、今日の技術革新を牽引するテクノロジー、そして最終的に自分と会社が責任とリスクの連続体のどこに落ち着くかを仔細に検討してください。あらゆるアプリケーションで正解となる唯一のコントローラなどありませんが、計画的に体系立てられた手法を用いてコントロールの選択肢を検討すれば、製造システムのライフサイクル全体にわたって付加価値の得られるシステムが見つかることでしょう。

© 1998 Rockwell International Corporation  
All trademarks, company names and brand names referred to throughout this publication are used for identification purposes only and remain the property of their respective companies. All rights reserved.



**Allen-Bradley**

Allen-Bradley, a Rockwell automation business, has been helping its customers improve productivity and quality for more than 90 years. We design, manufacture and support a broad range of automation products worldwide. They include logic processors, power and motion control devices, operator interfaces, sensors and a variety of software. Rockwell is one of the world's leading technology companies.



**Worldwide representation.**

Argentina • Australia • Austria • Bahrain • Belgium • Brazil • Bulgaria • Canada • Chile • China, PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Ecuador • Egypt • El Salvador • Finland • France • Germany • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Ireland • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Malaysia • Mexico • Netherlands • New Zealand • Norway • Pakistan • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia-CIS • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela • Yugoslavia

Allen-Bradley Headquarters, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444