

(解説)

# 効率的な制御システム構築ツール「Kobenicle<sup>®</sup>」

Efficient Programming Environment “ Kobenicle<sup>TM</sup> ” for Controlling Systems



松本泰治\*

Yasuharu MATSUMOTO



森本光孝\*

Mitsutaka MORIMOTO



掛林康典\*\*

Yasunori KAKEBAYASHI



佐藤 隆\*\*\* (工博)

Dr. Takashi SATO

We have developed a new PC system “ Kobenicle ” for industrial machine control systems. Since it uses OLE for process control(OPC) any programmable logical controller (PLC) can be applied to this system. Furthermore it is based on “.NET ” application that gives us various useful standard parts to construct user interface.

まえがき = 生産システムの基本要素としての産業機械の制御システムに対する要求は次第に高度になってきている。対象装置単体の制御に限らず、複数の装置による協調制御や、分散制御 (DCS: Distributed Control System) 系システムフレームの中でライン連携制御の要求もある。これらの要求に短期間で対応し、かつ信頼性の高いシステムを構築するため、制御システムを効率的に開発することが急務となってきている。

そこで当社は、Supervisory Control and Data Acquisition (以下、SCADA という)と同様、Human Machine Interface (以下、HMI という)構築機能を持ち、さらにエンジニアリング効率を向上するシステムプラットフォーム Kobenicle<sup>®</sup>を開発した。Kobenicle は、複雑化するシステム要求に応じた高度な制御機能を組込むことができ、産業機械装置全般に適用可能である。

## 1. 従来の産業機械制御システム

産業機械の制御システムとそのHMIを作成する場合、機械を制御するコントローラにProgrammable Logic Controller (以下、PLC という)を用い、機械操作やデータ入力、機械状態表示を行うグラフィックディスプレイにタッチセンサを内蔵したタッチパネル(以下、TP という)を利用することが多い。

PLCには、ユーザスペックに応じて国内外各社の製品が採用されている。一方、TPは、機種ごとにHMIのプログラミング手法が異なるため、多様なPLCとTPの組み合わせに対してそれぞれ個別のプログラム開発が必要となり、全体としての開発効率が低下する。TPが保持できるデータ量などのハードウェア上の制約もあり、複雑な制御や大容量のデータ採取・処理が必要な場合は対応が困難である。

そこで最近では、柔軟かつ高度な処理を行うために、

TPに替えてPersonal Computer (以下、PC という)を採用する場合が増えてきている。

## 2. PCシステムの課題

PCをユーザインタフェースとするシステムは通常、専門のプログラミング技術を必要とするうえに、システム全体が複雑になるため、品質管理項目が多くなる。また、汎用PLCとの通信ソフトウェアの作成にあたっては、それぞれのPLCメーカー固有の通信ドライバに対応させる必要があるなど、高度なプログラミング技術が求められる。設計・作業負荷も大きくなる。これらの課題を解決し、PCによる高機能な制御と汎用TP並みの低い設計負荷で作成できるシステムを実現するため、標準的なシステムプラットフォームとしてKobenicleを開発した。図1にそのPCシステムの構成を示す。

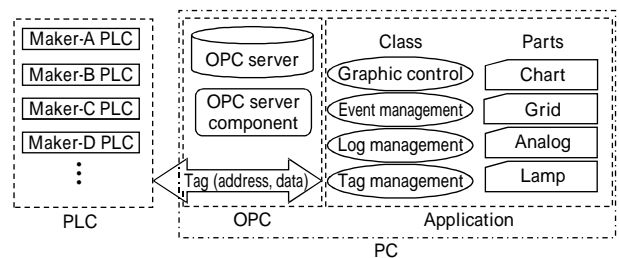


図1 PCシステムの構成図  
Fig. 1 Configuration of PC system

## 3. PCシステムの標準化

### 3.1 OPCとアプリケーション開発環境

PCで構成した制御システムと汎用PLCとを通信接続する手段として、マイクロソフト社が提唱しているOLE for Process Control(以下、OPC という)規格を採用した。

\*機械エンジニアリングカンパニー 圧縮機事業部 回転機技術部 \*\*機械エンジニアリングカンパニー 開発センター \*\*\*機械エンジニアリングカンパニー 開発センター 開発部

OPC は、PC と制御装置などを接続するための汎用的な通信仕様であり、データの受渡し経路にタグというデータセットを使う。タグの定義情報には PLC のアドレスやデータ型など、制御システム固有の属性を持たせることができる。このタグ定義情報を用いることにより、PLC の機種の違いを OPC で吸収することができる。

PC システムの開発プラットフォームには、マイクロソフト社の .NET Framework ベースアプリケーション開発環境を採用した。これにより、同環境上で開発された市販あるいは内製のソフトウェア部品を使用することができ、これらを画面に配置することで画面設計や機能設計が容易になった。例えば、機械の状態を表示するランプや数値・グラフ表示器、機械への動作を指示するボタンや数値入力ボックス欄などを、PLC のデータと連携する機能を持った標準部品としてあらかじめ準備しておき、これらの部品の属性にタグ情報を割当てることにより、上述した OPC を利用した PLC とのデータ連携が自動的に行われる。このような仕組みを採用することにより、汎用の TP と同様に部品を配置し、属性を設定するだけでユーザインタフェース設計を行うことが可能となり、設計効率が向上した。

### 3.2 ソフトウェアの再利用

Kobenicle においては、ソフトウェアの画面制御、データロギング管理、アラームイベント管理などの各種設定を図 2 に例示する eXtensible Markup Language (XML) で表記したファイルで定義しており、ユーザ間あるいは機種間の差異にこの定義ファイルを変更することで容易に対応することができる。この構成方法により、複雑な制御や大容量のデータ採取など、様々な機能を記述したプログラムを部品化することが可能となった。また、プログラムを変更する必要がなくなったことから、再利用率が高まり、高い生産性と信頼性を同時に達成させることができた。

市販の SCADA などのソフトウェア製品を使用する場合、個々の製品仕様の差異を把握した上でソフトウェア品質を確保する必要があり、統一的手法では対応できな

```
<?xml version="1.0" encoding="shift_jis" ?>
<ArrayOfDataLogDef xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<DataLogDef>
  <UniqID>1</UniqID>
  <Cycle>true</Cycle>
  <CycleMinutes>1</CycleMinutes>
  <PeriodTime>
    <string>12:00</string>
    <string>18:00</string>
    <string>0:00</string>
  </PeriodTime>
  <GridFormName>frmDataLog1</GridFormName>
  <GridName>KccDataLogGrid1</GridName>
  <OutputDir>c:\kobecocontroller\Log\PLC1\DataLog1\</OutputDir>
  <MaxFileSize>100000</MaxFileSize>
  <MaxGridRec>100</MaxGridRec>
  <KeepDays>1095</KeepDays>
</DataLogDefRecord>
<DataLogDefRecord>
  <DataLogDefRec>
    <OPCTagID>U01.F01.D100</OPCTagID>
    <TagNO>PI-101</TagNO>
    <TagName>圧縮機吸込圧力</TagName>
    <TagUnit>MPaG</TagUnit>
    <TagFormat>0.00</TagFormat>
  </DataLogDefRec>

```

図 2 XML 定義ファイル  
Fig. 2 XML definition file

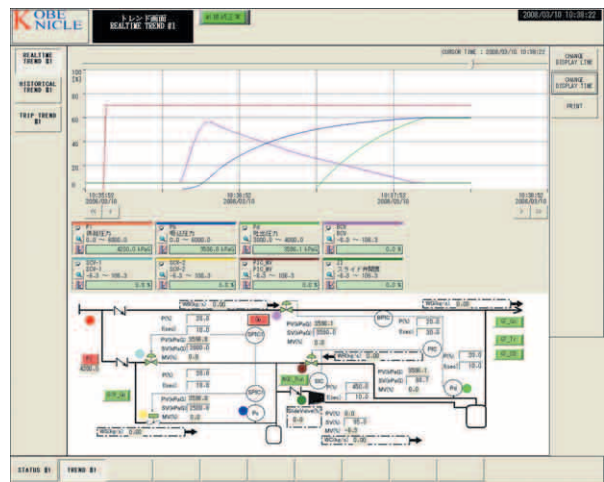


図 3 プロセスのシミュレーション  
Fig. 3 Process simulation

い。また、バージョンアップを使用側では管理することはできない。Kobenicle を用いることでプログラムの標準化が可能になり、必要なプログラムを XML 記述で容易に構成することができる。また、個々のモジュールごとにバージョン管理を行うことにより、設計時間の短縮や設計効率の向上が可能になった。

### 3.3 OPC とシミュレーション環境

前述したように OPC は統一的な通信インタフェースを定義しており、接続する装置間の通信ソフトウェアの開発負担を軽減することができる。PC と装置の間でデータ授受ができる特徴を利用した機能拡張が可能である。一つの応用として、効率的なシミュレーション環境の構築がある。プロセスや現場インタフェース、各種電動機盤などをソフトウェアによって作成し、PC 内にシミュレーション環境を構築することができる。これにより、実機による試運転を待たずに、制御システム全体の検証を設計段階で行うことが可能となる。

プロセスのシミュレーション例を図 3 に示す。

## 4. 適用事例

### 4.1 スクリュー式ガス圧縮機での適用事例

制御・監視システム用に Kobenicle が適用された事例として、天然ガス発電所に納入したスクリー式ガス圧縮機の制御システムを紹介する。

圧縮機個々の制御用 (3 台) に加え、全体ユーティリティ装置の制御用 (1 台) と合わせて 4 台の PLC を使用している。これらを一括集中管理したいというユーザからの要求に基づき、Kobenicle を用いた制御システム (図 4) を構築した。

### 4.2 各種機能の紹介

#### 4.2.1 ステータス表示機能

Kobenicle では、スイッチ、ランプ、バーグラフ、トレンドグラフ、グリッド表示など約 20 種類のオリジナルの標準ユーザインタフェース部品を準備している。このため、HMI プログラミングは、機能設計後に画面構成を XML で記述し、グラフィック要素を配置するだけで完了する。また、Kobenicle では .NET Framework の開発環境を使用できるため、柔軟なプログラミングが可能で



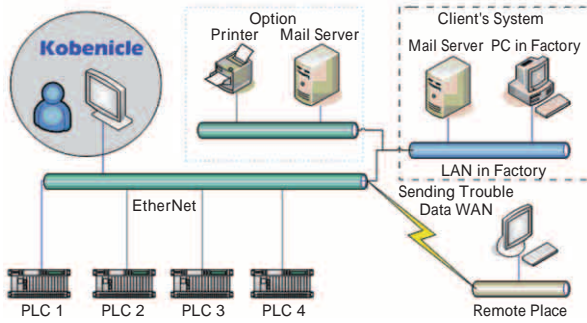


図 4 Kobenicle システム  
Fig. 4 Kobenicle system

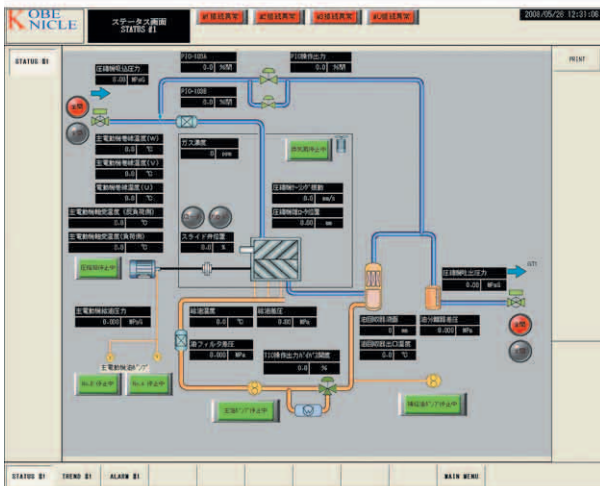
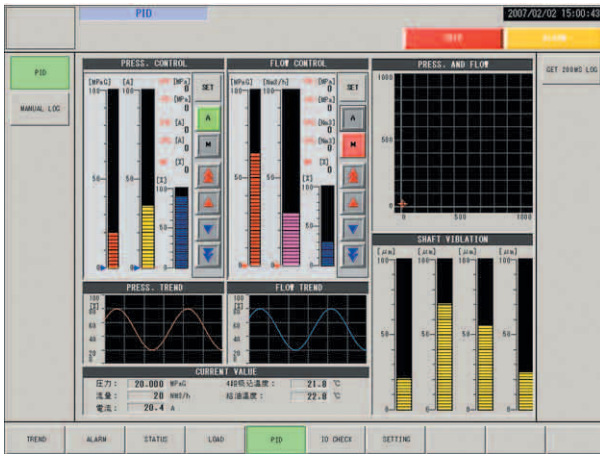


図 5 表示画面サンプル  
Fig. 5 Monitoring display

あり、視認性の高いグラフィカルなステータス表示など（図 5）を容易に実現することができる。

#### 4.2.2 トレンドデータロギング機能

本事例では 32 点のアナログデータを 1 秒間隔でロギングするが、HDD の容量次第で数年間のロギングが可能である。

画面上で任意の時刻を指定してこれらの保存データを呼出すことにより、トレンド表示することができる（図 6）。また、CSV 形式のテキストファイルで保存されているため、詳細なデータ解析などを行うことができる。

#### 4.2.3 解析用データロギング機能

前述のトレンドデータとは別に、圧縮機トリップ発生時には、トリップ発生前 15 分間、およびトリップ発生後 1 分間のデータを 0.2 秒間隔で保存することができる。

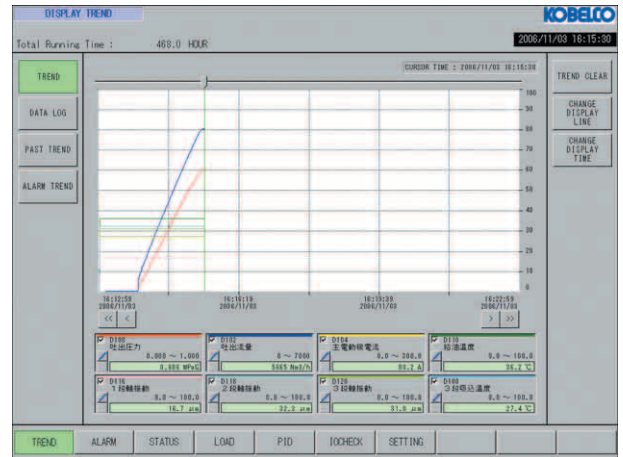


図 6 トレンドデータ  
Fig. 6 Trend data

この解析用データは、トレンドデータ同様に任意の時刻データを画面上にグラフ表示することができ、かつファイル出力も可能であることから、現場での解析ツールとして利用できる。

#### 4.2.4 リモート通報機能

本システムは、圧縮機トリップ発生時には、指定した宛先に電子メールによって通報する機能を持っている。また、解析用データログを電子メールに添付して通報することもできる。

産業機械制御用 PC のネットワークは、セキュリティの観点からユーザのメールサーバに接続することが難しいため、オプションで工場内の端末専用のメールサーバを用意し、上記異常通報を発信できるようにした。図 7 のような携帯電話通信網のメールサーバサービスを利用したシステムを活用することもでき、解析用データログをあらかじめ圧縮してデータバケットを極小化し、実効通信速度の向上を図っている。このリモート通報機能により、解析用データを指定した相手に送信することができ、故障原因の迅速な解析と早期復旧に貢献している。

#### 4.2.5 イベントログ機能

トリップ、アラーム、およびイベントのログは、図 8 のようなフォーマットで表示し、管理レベルに応じて色分け表示することができる。また、イベント発生と同時にそのイベントに関する対処内容を表示する仕組みとし、操作マニュアルレス化を図っている。

その他、運転チャート表示、設定値入力など、圧縮機に必要な機能に特化したオリジナルライブラリを整備している。

#### 4.3 今後の展開

Kobenicle には市販の SCADA ソフトで提供しているいわゆるプログラムレスの HMI 構築機能が組込まれており、産業機械用の標準的な HMI を容易に実現することができる。一方で、汎用開発環境を採用していることからソフトウェアによる機能追加が容易であり、機械の特性に適合した特殊な制御機能、監視機能などを柔軟に拡充することができる。

この利点を活かして、次のような展開を行う予定である。

- 1) ターボ圧縮機の特長アンチサージ制御機能の追加

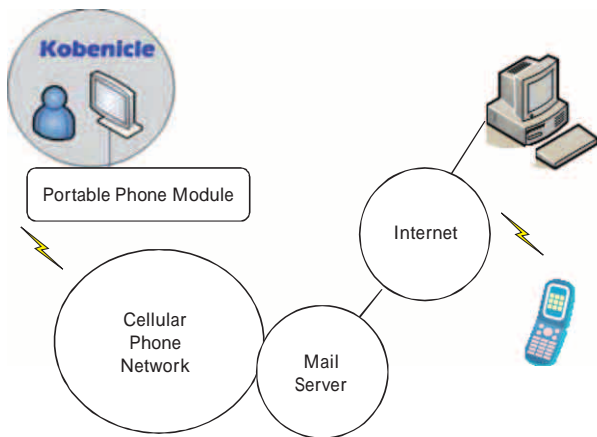


図7 携帯電話通信メールシステム  
Fig. 7 Cellular phone mail system

発生日時	タイプ	タグ名	メッセージ	ステータス
2000/05/28 13:58:12	異常検知	PF-105L	圧縮機停止作動中	発生
2000/05/28 13:58:13	異常検知	PF-105R	圧縮機停止作動中	発生
2000/05/28 13:58:14	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:15	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:16	イベント	10230	シフト開始	発生
2000/05/28 13:58:17	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:18	異常検知	PF-105L	圧縮機停止作動中	発生
2000/05/28 13:58:19	異常検知	PF-105R	圧縮機停止作動中	発生
2000/05/28 13:58:20	イベント	10230	シフト開始	発生
2000/05/28 13:58:21	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:22	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:23	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:24	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:25	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:26	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:27	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:28	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:29	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:30	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:31	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:32	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:33	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:34	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:35	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:36	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:37	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:38	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:39	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:40	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:41	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:42	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:43	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:44	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:45	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:46	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:47	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:48	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:49	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:50	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:51	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:52	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:53	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:54	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:55	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:56	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:57	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:58:58	異常検知	EXG2	異常停止 (C/C/C)	発生
2000/05/28 13:58:59	異常検知	EXG1	異常停止 (C/C/C)	発生
2000/05/28 13:59:00	異常検知	EXG2	異常停止 (C/C/C)	発生

図8 イベントログ  
Fig. 8 Event log

- 2) 圧縮機専用の簡易振動解析・故障診断システムの追加
- 3) 運転シミュレーションシステムの構築
- 4) 遠隔監視システムの構築
- 5) 上位システムとの接続

これらの機能は個々のニーズに沿って開発を進めるが、完了の後、標準機能として整備し蓄積していく予定である。

むすび=産業機械の制御、操作に汎用的に使用することを目的として、HMI構築用ソフトウェア Kobenicle を開発し、実機に適用した。OPC インタフェースを導入することで多様な PLC に接続するためのプログラム負荷を軽減し、産業機械に求められる HMI を実現するだけでなく、PC のプログラム開発環境を利用することによって、機械固有の制御機能などを組込むことを可能とした。XML による記述方式の採用により、全体の操作フローを容易にカスタマイズすることを可能とした。さらに、OPC の特徴を生かしたシミュレーションシステムなど、多様な機能の組込みを実現することを可能とした。加えて、内製化により、バージョンアップなども主体的に実施することができ、広く産業機械全般を対象としたソフトウェアへの展開により、さらに有用性を増すと考えられる。今後とも機能拡張と適用範囲拡大を図って行きたい。