

解説 組込みシステム開発の現状

5. FA 制御機器のシステム開発について —組込み機器としての数値制御装置を中心に—

System Development of Numerical Controller as Embedded Component by Shigeki NANKAKU (Numerical Control System Dept, Mitsubishi Electric Corp.) and Hidemasa IIDA (Asic Design Center).

南 角 茂 樹¹ 飯 田 秀 正²

1 三菱電機(株)名古屋製作所 NC システム部

2 三菱電機(株)名古屋製作所 ASIC 開発センター

1. はじめに

FA 制御機器の組込みソフトウェアとは機器内部の制御手順や情報処理のために、製品に組み込んだマイクロプロセッサなどによるソフトウェアのことであり、ハードウェアと一体になった製品である。FA 制御機器の組込みソフトウェアの特徴としては次のような点があげられる。

- (1) 産業用機械の制御に使われるため不正動作は人命にかかわり、正しく動作できない状況の場合は必ず安全側で動作する、フェイルセーフが重要である。
- (2) ハードウェア自体の開発も行うため、正しく動く保証がないハードウェアへソフトウェアを組み込むところから開発を始めなければならない。たとえばワークステーションやパソコンのソフトウェアであれば、開発したソフトウェアが正しく動作しない時は、ほとんどの場合ソフトウェア自体が不具合を含んでいる、しかしハードウェアも新規開発した場合は、メモリへ書いた情報とメモリから読んだ情報が異なっている場合さえある。
- (3) ソフトウェアを ROM 化して機器に実装することが多く、ROM 化可能な実行コードの生成 (パソコンソフトウェアのように実行コード自体を書き換えるようなソフトウェアを作成することができない) を考慮した設計が必要である。
- (4) 製品によってはリアルタイム OS をも含めた設計が重要となる。
- (5) リアルタイムの応答性、コスト、実装面

積、さらには使用電力をも含めて、ハードウェアとソフトウェアの間で総合的なトレードオフ技術が必要となる。

本稿では組込みソフトウェア製品としての FA 制御機器全体を簡単にサーベイした後、その1つである、数値制御装置 (Numerical Controller-以後 NC) におけるシステム開発の現状および開発手順と問題点、さらに今後の展開について紹介する。

2. FA 制御機器分野における組込みシステム

FA 制御機器は次のように分類できる。

- 数値制御装置 (NC)
- プログラムロジックコントローラ (以後 PLC)
- 汎用サーボアンプ
- インバータ
- 汎用主軸アンプ

NC に関しては次の章で詳しく述べる。

PLC とは、その主な制御機能は信号処理で、リレーを組み合わせて、工場のラインなどを制御していたものをソフトウェアで制御できるように置き換えたものである。ユーザがパソコンなどのラダーツールなどで作成した信号処理のプログラムを PLC は処理する。なお、ラダーとは信号処理のプログラムを作成するための専用のプログラム言語である。現在では PLC 処理をすべてパソコンで行うものもあるが、PLC の主流はまだ専用の PLC ハードウェアで制御を行う機器である。

ユーザは、たとえば1つの材料を複数の NC で順番に加工したり、加工したものをロボットに

表-1

規格	推進団体
DEviceNet	ODVA
JPCN-1	日本電機工業界
CC-Link	三菱電機
LONWorks	エシエロン社
Fieldbus	Fieldbus 協会
PROFIBUS	シーメンス社

渡したりなどの全体のラインの調停/制御などにも PLC を使用する。

PLC の通信ネットワークは接点信号の通信が主でそのネットワークをフィールドバス（またはデバイスネット）と呼んでいるが、たとえば表-1 に示すような規格が現在提唱されている。

組込み制御装置としてみると、PLC はビット演算の機能を強化した 32 ビットカスタム CPU を使用して、コードサイズ 4~8 K バイトのハードウェアを使用して、ユーザが入力したプログラムを繰り返し実行している。そのためプログラムの長さによって繰り返し周期は異なる。ラダープログラムの 1 ステップ命令の処理時間は平均 $0.5\mu\text{s}$ 以下である。

PLC 自体のソフトウェアは通常 OS は使用せず、その開発にはアセンブラを使用してリモートソースレベルデバッグでデバックを行う。

汎用サーボアンプとはサーボモータの位置、速度などを制御する装置である。とくに汎用と書いたのはアンプのインタフェースが公開されていて、たとえばパソコンに挿入するタイプのサーボアンプのインタフェースボードはパソコンからボードにデータを送ることにより、サーボモータの制御が可能だからである。

16 から 32 ビットの 1 チップマイコンや汎用 CPU、DSP などを使用して、コードサイズは 32 K バイト~128 K バイトで構成する。

そのソフトウェアはサーボを制御する部分はアセンブラで、それ以外の通信、表示などは C や C++ などの高級言語で記述することが多い。通常 OS は使用せず、デバックには ICE を使用することが多い。

制御装置の世界ではインバータとは標準の 3 相モータの速度制御を行う制御装置であり、汎用の

32 ビット CPU を用いたコードサイズ 128 K バイトほどの装置で、OS は使用しない場合が多い。

主軸アンプは主軸モータの制御を行う装置であり、機能的にはちょうどサーボアンプとインバータの間の装置である。その機能として主軸モータの位置制御が必要か、速度制御だけでよいかによりサーボアンプとインバータのどちらに近いかが決まる。

なお上であげたのはあくまでも一例であり、装置によっては仕様が大きく異なる種類のものも存在する。これらの制御機器は各々独立しても使用されるし、組み合わせて使われることも多い。

また NC や PLC を群管理するためのセルコントローラや、工作機械で 3 次元加工を行うためのプログラムを生成する CAM 装置なども FA 制御装置の一種といえるがそれらについては今回は省略する。

3. NC 装置とそのソフトウェアの概要

NC 装置とは、概念的には各種工作機械やロボット、レーザ、放電加工機などの動きを制御するための工業用のコンピュータである。また NC 装置はサーボアンプや主軸アンプ、PLC と一体になった装置であるので、とくにこれらもいっしょに開発して、新しい補間方式の組込みによる高速、高精度などを実現するために NURBS 補間など独自の機能を付加することが多い。サーボアンプ、主軸アンプも独自に開発することにより、NC とサーボアンプ、主軸アンプの機能の分担を最適化したり、汎用サーボのインタフェース以外の情報をやりとりすることができるからである。

一般に使われる汎用サーボアンプ、主軸アンプに対して特定の NC 装置に専用に開発されたものを専用サーボアンプ、主軸アンプと呼ぶ。とくに高性能 NC 装置は専用サーボアンプ、主軸アンプとしか接続できない。普及機能の NC 装置には汎用サーボアンプを使用できるものもある。PLC に関しても同様である。以後 NC 装置と呼ぶ場合はとくに断らないかぎりサーボ、主軸、PLC すべてを含んだものことである。

人間に代わって機械を操作して、均質な製品をより早く、より安く作り出す装置として生み出され、発展してきたのが NC 装置である。JIS 数値制御工作機械用語によると NC の定義は“数値

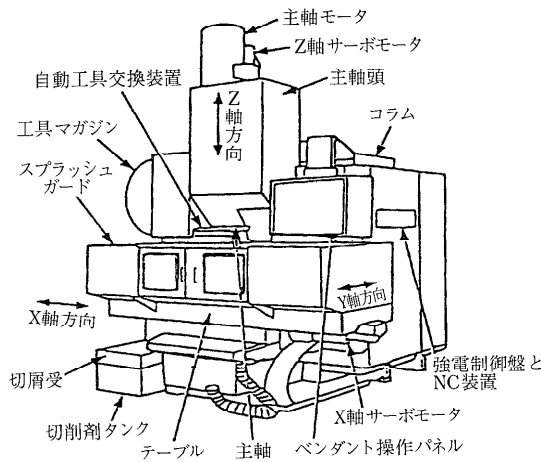


図-1 縦形マシニングセンター外観

制御工作機械において、工作物に対する工具の位置を、それに対応する数値情報で指令する制御”となっている。

NC装置の入力は、与えられた加工物の図面に対して、その加工に必要な工具経路や加工条件などすべての情報を記述できる、NC動作を記述する言語で書かれた、NCプログラムである。

それをNC装置の情報処理装置が逐次解釈(インタープリット)して、サーボ機構への指令パルスに変換する。

サーボ機構ではこの指令パルスを入力として機械を制御して、機械はNCプログラムに対応した動作を行って目的の加工を実現する。

図-1に示すのはNC装置付きの工作機械の外観(縦形マシニングセンタ)である。

NCに要求される機能/性能を次にあげる。

(1) リアルタイム性

- サーボ：0.2 ms以下の応答性，周期性
- 人がNCに与える指令：1 ms以下の応答性

(2) 高度なサーボ制御技術

- 0.1 μ m以下の位置制御
- 振動を起こさない高速加減速制御
- 機械的誤差(バックラッシュ，ねじれ，たわみ，伸縮)に対する補正制御
- 同期，重畳制御(主軸によって回転中の加工物を回転させたまま別の主軸で掴みかえる，動いている加工物を追いかけて加工する)

(3) 高度なマンマシンインタフェース(MMI)

- CAM機能，加工条件の自動決定などのエキス

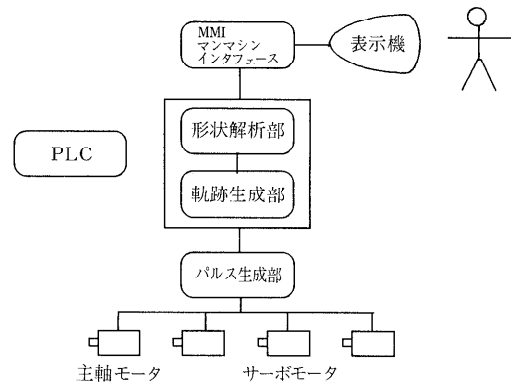


図-2 NCアーキテクチャのブロック図

パート機能などNCの加工プログラムを作成するための対話型プログラミング

- 被加工物の加工状況を示す3次元グラフィックシミュレーション
 - 機械の周辺動作(PLCの信号など)のリアルタイムモニタリング
- (4) 豊富なI/O機能
- シリアル通信やイーサネット通信，サーボ通信，PLCネットワークなど各種通信プロトコルのサポート
 - フロッピーディスク，ハードディスクなど大容量外部記憶デバイスのサポート

NC装置のソフトウェアアーキテクチャのブロックを図-2に示す。NC装置におけるPLCは外部のPLCとの通信のほかにNC工作機械自体の機械制御(たとえば切削油を出したり止めたり，照明のオン/オフなど)も行い，独立したプロセッサを備えることも多い。

この図ではPLCを除くと，下に書いたブロックほど時間的制約が厳しい。とくにパルス生成部からサーボ/主軸への通信制御はハードデッドライン制御が必要である。

一定周期時間ごとのサーボ/主軸へのデータ送信が止まると，加工に失敗して(たとえば旋盤でネジの加工中に刃物を動かしているサーボへの指令が止まるとネジではなく溝になってしまう)，NC装置としては成り立たなくなる。

一方MMIはたとえば現在の刃先の座標表示が多少遅れても加工には影響しないなど，それほどリアルタイム性は必要ではない。

さらにNC装置のソフトウェアの構造は次のような特徴も備えている。

機能的に複数のレイヤに切り分けることができ、レイヤ間の情報の受け渡しはバッファリング、通信などを利用することにより各レイヤの動作時間、周期時間の違いを吸収することが可能である。

サーボからの周期的なフィードバック情報の受け取りや、非同期に発生する外部からのイベント(各種ドライバからの割込み、非常停止など)など、割込み処理が多い。

レイヤ間の情報の流れは一方方向ではない。たとえばサーボモータとの間にはフィードバックループが必要である。

外部機器とのデータ交換(シリアル通信、ネットワーク通信、ハードディスクなど)が必須であるため各種ドライバを備える。

以上の特徴をもつため、NC装置にはOSとしてはリアルタイムOSを利用することが適している。そのためNC装置では従来からリアルタイムOSを使用してきた。ハードウェア的には32~64ビットCPU、DSPを1~複数個使用して、メモリは4~8Mバイト、開発言語はとくに速度が要求される部分はアセンブラ、そのほかの部分はCやC++などの高級言語で記述している。

4. NCシステムの開発事例

図-2には現れないOSなどの部分も加えて、NC装置はソフトウェア的には大きく次の5つのブロックに分けることができる。・OS、ドライバ部 ・マンマシンインタフェース部 ・PLC部 ・形状解析軌跡生成部 ・サーボ/主軸パルス生成部

NC装置のソフトウェア開発は通常この5つのブロックを同時並行で開発する。本来はハードウェアのデバッグも並行して行うため、ICEを使用するのが効率的であるが、性能/価格の面から最新のCPUを使うことが多く、ICEが利用できなくなっている。なぜなら最近のCPUでは命令やデータなどがCPUの内部のキャッシュで処理され、その場合はCPU外部にはアドレス信号なども出ないためICEの開発が難しく、必要なICEが発売されるのはNC装置の開発が終わった1年以上後であることが多いからである。

以下にこの開発ステップの一例をOS、ドライ

バ部の開発をもとに示す。OS、ドライバはとりあえずホスト計算機上でソフトウェアは作成しておくが、本当の作業はハードウェアのプロトタイプが完成してから始まる。通常は、ターゲットハードウェアが通常備える不揮発性メモリ(EEPROM、フラッシュメモリ)にモニタを搭載することが最初の作業である。

このモニタのデバックはかなり泥臭い作業である。この段階ではメモリも信頼できず、どこまで実行したかのログをメモリに書き込んでおいてもそれをみる手段が存在しない。

通常はたとえばモニタソフトウェアの最初にLEDを点灯させるコードを入れておく。通常のソフトウェアからは、電源投入によりプログラムを走らせた場合にLEDが点灯するのが当たり前のことに思えるが、これが実行されるということは、コードを書き込んだメモリの内容を正しく読むことができた、リセットペクタが正しく設定できた、ホストでソフトウェアが正しく生成できた、ROM切りも正しくできた、バイトオーダを切り換えることができるCPUの場合はエンディアンの設定も正しくできたなど多くのことがこれにより検証できる。

次いで各種メモリのチェックを行う。ハードウェアが備える各種メモリ(DRAM, SRAM, フラッシュメモリなど)を順次チェックしておく。キャッシュを備えるCPUの場合はキャッシュのチェックも行う。次がI/Oである。たとえばシリアルI/O機能を動かしてRS-232C端末と通信できるようにすると、デバック作業は飛躍的に進む。これがあればメモリ内容のダンプ機能や、CPUレジスタの表示機能、ブレークポイントの設定機能などをモニタに組み込んでいくことが可能になる。

場合によっては逆アセンブラなどもモニタに組み込む場合もある。

このモニタ自身がOSの場合もあるが、このモニタがさらに本当のOSオーダウンロードする場合もある。そのダウンロードする機能もモニタには必要である。

リアルタイムOSとしては通常のタスクスケジューリング機能(ディスパッチャー)、割込み制御、ブレークポイントの設定、メモリやレジスタ類のリード/ライト、シンボリックデバッグの機

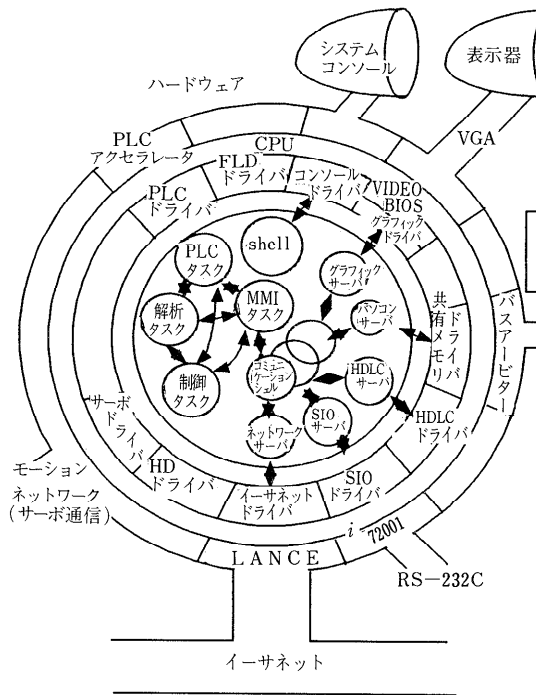


図-3 NCのソフトウェア構造

能のほかに、最近ではホストとの連携によるリモートソースレベルデバッグの機能も必要である。とくにRISCチップを使用した場合の、命令の再配置、最適化、ディレイドブランチなどによってソースレベルデバッグの必要性はますます増大している。

サーボ生成部のソフトウェアも以上のOSドライバの開発と似ている、なぜならこれもサーボハードウェアの開発もともなうからである。

そのほかマンマシンや軌跡生成部のソフトウェアの開発は通常のパソコンやEWSのソフトウェアの開発に似ている。ただしほかの機能ブロックとのインタフェースデータへの依存が大きいいためスタブや仮のデータなどを作成してホスト上でソースレベルデバッグを行う。あるいはホスト上にターゲットをシミュレートできる環境を構築しておきそこで実行する。

以上の各機能ブロックが完成した時点ですべてのモジュールを結合して試験を行う。

このようにしてできたシステム全体のソフトウェアモデルを図-3に示す。

5. 開発における問題点

NCソフトウェアの開発は以上のように行っているため次のような問題点がある。

- ハードウェアができるまでソフトウェアの本格的な開発が開始できない。
 - ハードウェアとソフトウェアで最適な役割分担ができていないかどうか不明である。
 - ドライバなど割込みハンドラのデバッグに効率的な手段、手法がない。
 - ほかのレイヤの生成するデータに依存する部分が多いため単体でのデバッグがやりにくい
 - 複数のCPUを使用する場合、どの機能をどのCPUに割り当てるかを定める方法が確立されたものが存在しない。
- そのため、ヒューリスティックに割り当てを決めている。

- タスクスケジューリングがプライオリティベースであるため、最適なプライオリティを決めるのが困難である。
- 最近のキャッシュを多く備えたCPUではICEが簡単にはできない。
- 現在のNCソフトウェアは機能、処理的にはブロック化、階層化ができてはいるが、データの流れるとしては必ずしも洗練された構造とはなっていない。

以上の問題点を解決するためには

- ハードウェアとソフトウェアを同時に開発する技術であるコンカレントエンジニアリング技術の実際の適用
- ソフトウェアでハードウェアの信号レベルまでもシミュレートできるシミュレータの開発または採用
- デッドライン制御が可能なりアルタイムOSの開発または採用
- 現在の手続き中心型からデータ中心型へのNCソフトウェア構造の見直しなどの対策が必要である。

6. FA制御機器の将来

ここ数年来のオープン化の流れはFA制御機器においても大きな流れとなっている。NC装置においてもパソコン技術との融合が最近の動向である。現状はWindowsなどのパソコンOSのリア

リアルタイム性や割込み応答性の不足などから、それらの必要性が比較的低い MMI 部にパソコンが使われることが多いが、スレッドや仮想デバイスドライバ (VxD)、カーネルモードデバイスドライバなどの使用により、パソコン OS 上でもリアルタイム OS 上と遜色ない NC システムが構築可能であると考えられる。

そのほか、NC をもっと大きく捕らえて、CIM (Computer Integrated Manufacturing) の機器の 1 つとして考えた場合に必要な機能として複数の NC で協調分散環境を構築する場合に現在のソケットを使用した通信のようにネットワークを意識したやり方ではなく、DSM (Distributed Shared Memory) 技術を利用して単なる共有メモリを介した情報伝達のようにみせてプログラムを簡易化したり、複数の CPU を使用する場合に現在のように特定の機能をあらかじめ特定の CPU に割り当てておくのではなく、機能オーガニクに CPU に割り当てる CPU プールモデルの採用など今後解決すべき課題は多い。

(平成 9 年 7 月 28 日受付)



南角 茂樹

1957 年生. 1982 年慶應義塾大学工学部数理工学科卒業. 同年三菱電機(株)に入社. 以来同社名古屋製作所において数値制御装置のリアルタイム OS, システムアプリケーション, CAM システム, 数値制御装置の周辺装置およびソフトウェア開発環境の開発に従事. 専門は OS, とくにリアルタイム OS, 現在同所 NC システム部主幹.



飯田 秀正

1945 年生. 1968 年九州工業大学制御工学科卒業. 同年三菱電機(株)に入社. 同年より鎌倉製作所において数値制御装置の開発に従事. 1978 年には数値制御装置の名古屋製作所への機種移管にともない名古屋製作所に移り, 数値制御装置の開発を統括, 1989 年三菱電機メカトロニクスソフトウェア(株)に出向し FA 全般組込システム開発を統括. 1996 年より名古屋製作所 ASIC 開発センターにて ASIC 開発を統括, 現在に至る. 日本機械学会会員.