

産業用マイクロコントローラ及びそのソフトウェア “PL/H-C”

Microcontroller for Industrial Use and Its Software “PL/H-C”

最近、マイクロコンピュータの産業用制御装置への応用は急激に増加している。従来8ビットマイクロコンピュータシステム“HMCS 6800”及びソフトウェア高級言語“PL/H”が開発されていたが、今回日立製作所は、モジュール構造で産業用制御装置に適したマイクロコントローラを開発したのでここに紹介する。

ハードウェアは、耐環境性、RAS機能に優れた取り扱いやすいものとなっており、ソフトウェアは、産業用の制御に広い用途をもつシーケンス制御を記述しやすくした“PL/H-C”(“PL/H”に制御用のソフトウェアを追加したもの)をもっている。

進士三千男* *Shinji Michio*
 小川靖一郎* *Ogawa Seiichirô*
 吉村一馬** *Yoshimura Kazuma*
 加藤正道** *Katô Masamichi*
 秦清治*** *Hata Seiji*
 布目勝**** *Nunome Masaru*

1 緒言

最近、産業の各分野で生産性の向上が要請され、加工、組立、検査の工程で省力化が推進されている。これに対して、この効果をより増強させるために、従来から精緻な制御及び情報管理が求められている。

マイクロコンピュータの出現は、その経済性とあいまってこれらの潜在ニーズを一挙に実現するものと受け取られ、急速に浸透し始めてきている。

今回日立製作所では、これらのニーズを満足させるため、産業用として広く機能を発揮できる産業用マイクロコントローラを実現したのでここに紹介する。

その特長はハードウェアとしては、

- (1) 日立マイクロコンピュータシステム“HMCS6800”を中核に使用している。
- (2) 耐環境性、耐ノイズ性に優れた装置である。
- (3) 産業用としての使用に十分なRAS (Reliability Availability Serviceability) 機能をもっている。
- (4) 実装に便利な構造をとっている。
- (5) 豊富な入・出力装置をもっている。

ソフトウェアとしては、

- (1) 言語処理用としてマクロアセンブラ、“PL/H”、“PL/H-C”などがある。
- (2) 言語処理とデバッグ管理用としてEMS (Executive Monitor System) と、制御プログラム管理用としてRMS (Real-time Monitor System) がある。
- (3) プログラムライブラリが豊富である。

2 ハードウェア

ハードウェアのシステム構成図を図1に示す。装置全体はモジュール構造となっており、実装上の簡便さを大いに増している。中央演算処理ユニットは、CPU(中央処理装置)、メモリモジュール及び周辺装置インタフェースモジュールを収納している。図2にCPUのブロック図を示す。同図にあるように、“HMCS 6800”を中心としてまとめられており、タイマ部、異常検出部など産業用に不可欠の機能を持ち、更

にメモリ[ROM(Read Only Memory)及びRAM(Random Access Memory)]をも内蔵している。またプログラマ、データタイプライタはこのCPUから直接接続できるようになっている。CPUは図3のようなモジュール構造となっている。

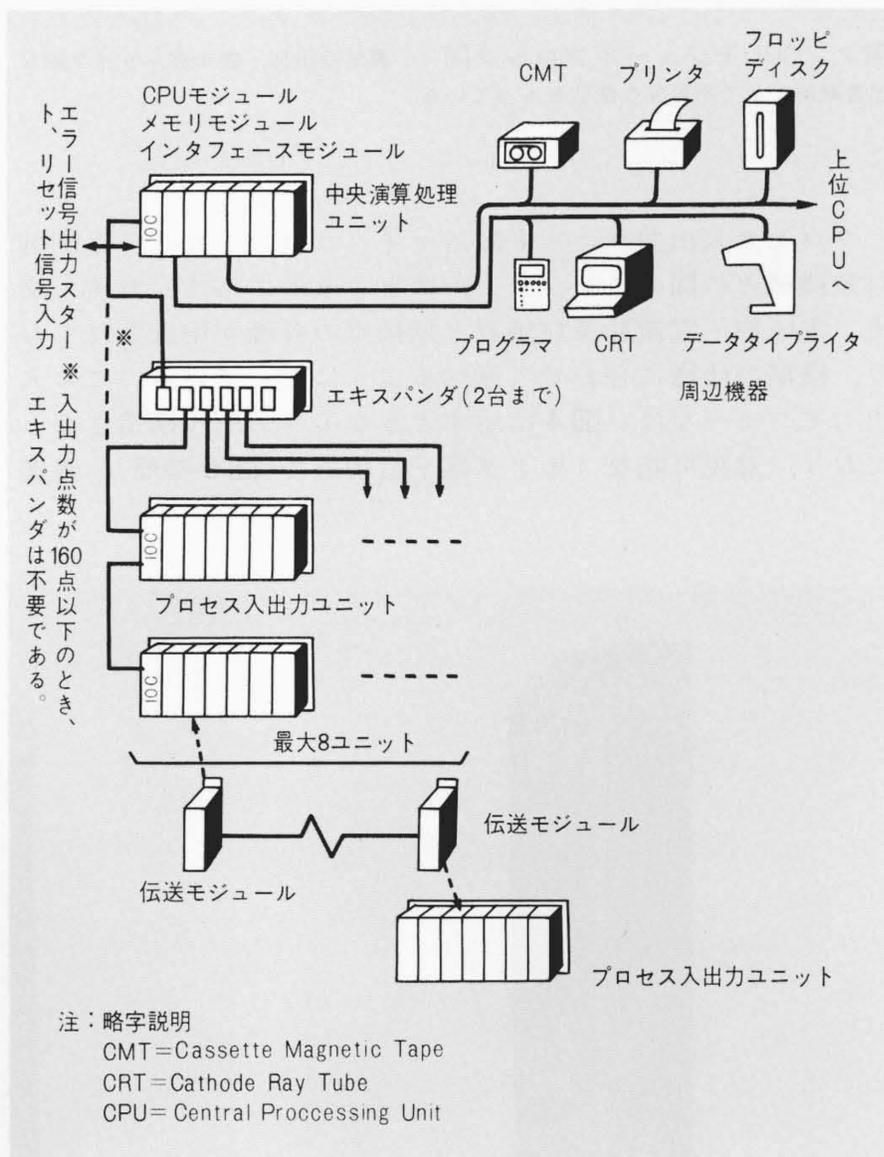


図1 N-7000システム構成図 CPUユニット、プロセス入出力ユニットは、各モジュール構造となっている。また各種の周辺装置も用意されている。

* 日立製作所習志野工場 ** 日立製作所システム開発研究所 *** 日立製作所生産技術研究所 **** 日立エンジニアリング株式会社

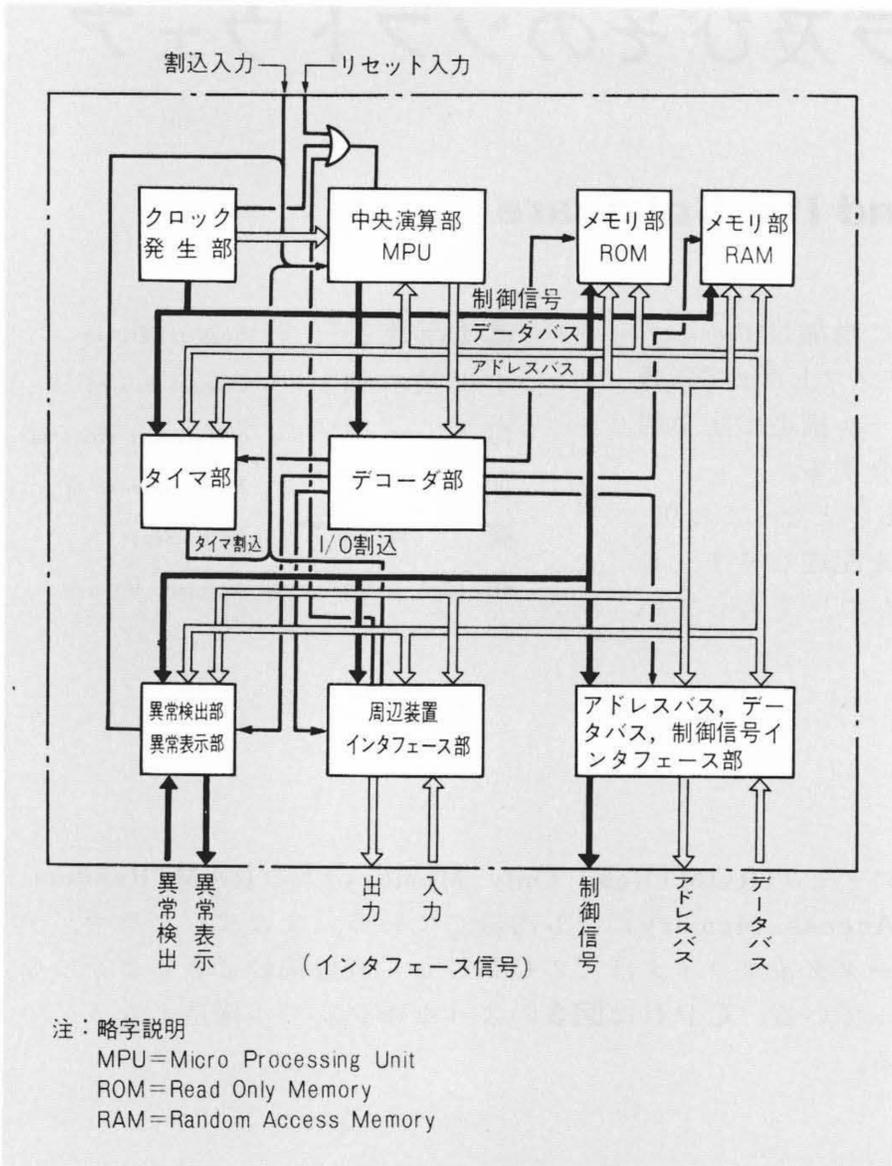


図2 CPUモジュールブロック図 異常検出部、表示部、タイマ部など産業用として不可欠な機能をもっている。

プロセス入出力ユニットは、マイクロコントローラと制御対象機械との間のインタフェースとなるところで、直流・交流、電圧形・電流形及び接点・無接点の各種が用意されており、機械の仕様に合わせて選べるようになっている。この入出力モジュールは、図4に示すようなモジュール構造となっており、着脱可能なコネクタ端子台構造で(図5参照)、実装

表1 ハードウェア仕様 ハードウェア全体の仕様を示す。PI/Oは豊富に用意されている。

区分	項目	
一般仕様	周囲温度	0~55°C
	電源	100V/110V
	耐ノイズ	NEMA規格準拠
	寸法	幅・高さ・奥行=570・350・277(mm)
各部仕様	CPU	"HMCS6800"
	メモリ	バッテリーバックアップ CMOSメモリ EPROMなど
	周辺	ポータブルプログラマ・データタイプライタ CRT・CMT・プリンタ フロッピディスク
	プロセス入出力 (PI/O)	エキスパンダ PI/O増設用 PI/Oモジュール 入力：5V, 24V, 48V DC 100V, 200V AC 出力：5V, 24V, 48V, 100V DC 100V, 200V AC (接点, 無接点)

注：EPROM=Erased & Electrically Programmable ROM

に便利な方式である。

表1にハードウェア仕様を示す。

3 ソフトウェア

アプリケーションソフトウェアの開発コストのうち、マイクロコントローラ体系開発コストが占める割合は非常に大きい。N-7000マイクロコントローラのソフトウェア体系は、図6に示すとおりである。ソフトウェアの生産性向上が急務であり、プログラムライブラリの充実、高級言語の導入、標準モジュール化など、種々の工夫と努力が進められつつある。

一般産業への応用はその範囲が広いため、アプリケーションプログラムの開発はモジュール化と問題向き高級言語によるのが、トータル的に得策と思われる。

"HMCS6800"用高級言語"PL/H"(インテル社PL/Mと同等)が開発され、これによりプログラムの開発効率は向上した。

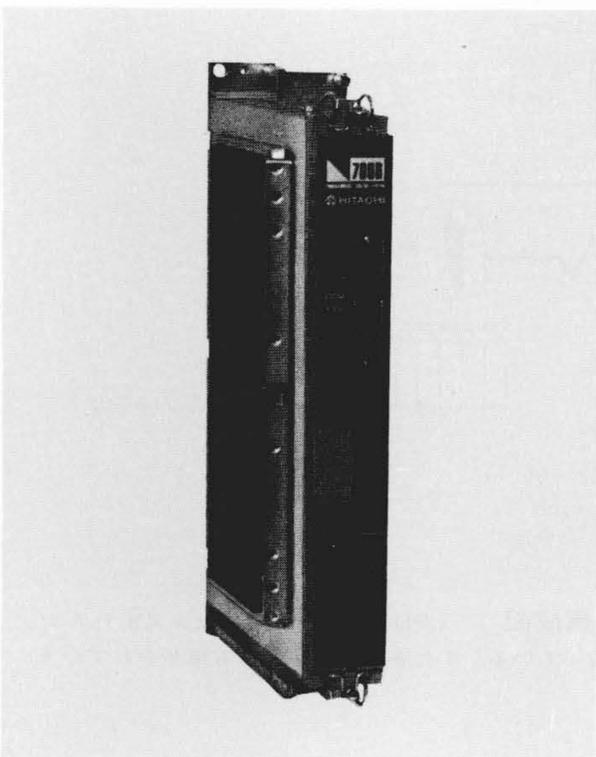


図3 CPU(中央処理装置) CPU概観図を示すもので、モジュール構造となっている。

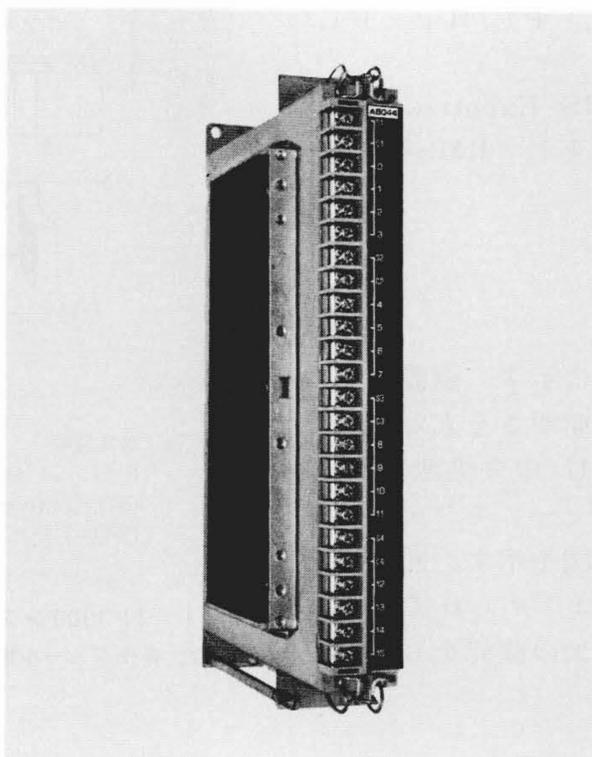


図4 入出力モジュール 自然空冷方式のモジュール構造である。

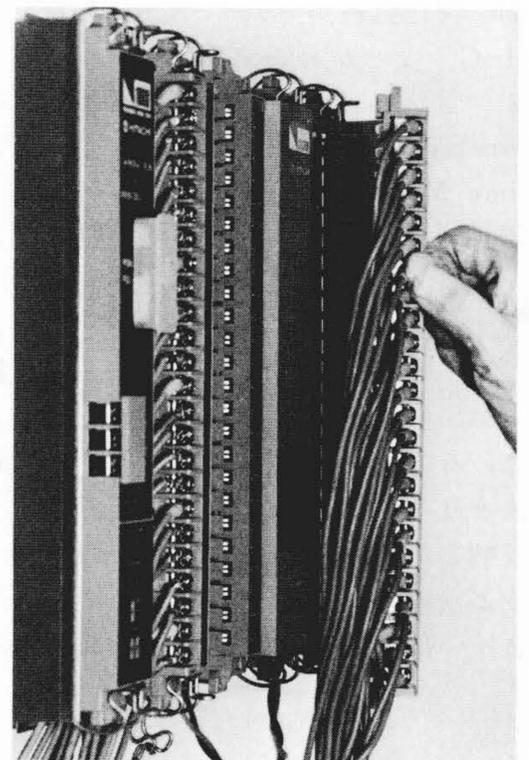


図5 入出力モジュール 着脱可能なコネクタ式の端子台構造となっている。

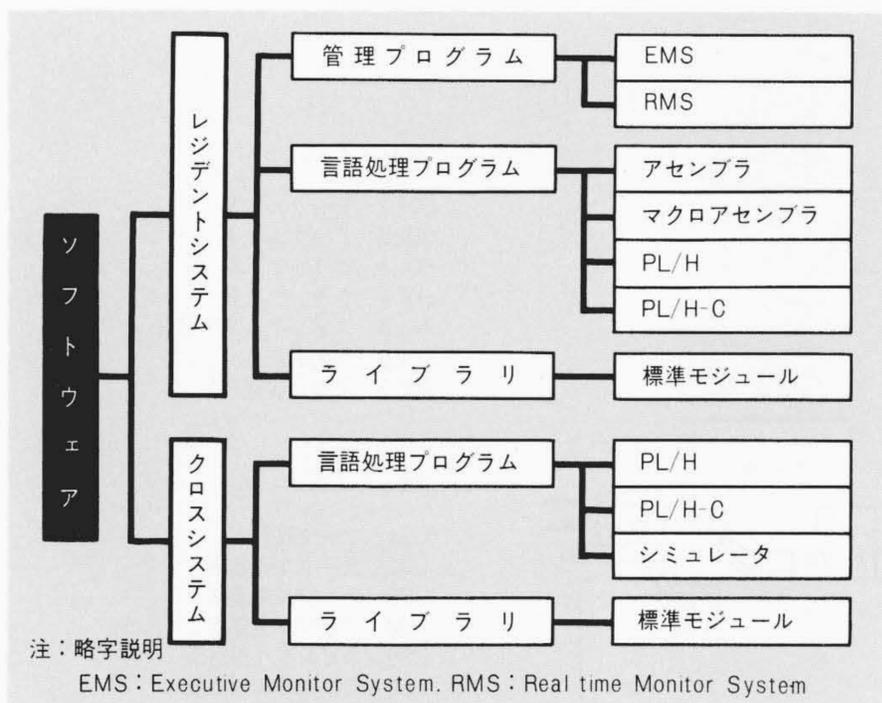


図6 ソフトウェア構成図 “HMCS6800”システムの豊富なソフトウェア“PL/H”、“PL/H-C”、標準モジュールなどに特長をもっている。

ここに紹介する“PL/H-C”は、“PL/H”に制御用機能を付け加えて制御用プログラムを書きやすくしたものであり、産業用マイクロコントローラ・システムのソフトウェア開発の生産性を一段と向上させた。

3.1 “PL/H-C”言語と処理の特徴

“PL/H-C”は、産業設備の制御によく使われるシーケンス制御の記述を容易にすること、オブジェクトの制御効率を良くすることを主目的として、“PL/H”に機能を追加し開発したものである。

3.2 “PL/H-C”言語の特徴

“PL/H-C”の制御プログラムの記述方法には、大別してある条件が成立するのを待って次のステップへ進む順序制御方式と、条件によって対応する処理が異なる条件制御方式とがある。条件制御方式は、従来からシーケンス制御に用いられているリレー回路と同じ考え方である。

これらの二通りの制御方式について、“PL/H-C”言語を用いてのプログラム記述をもう少し詳しく述べる。

順序制御は、例えば「リミットスイッチLS1がONになるのを待って、コイル(Coil1)をONにして、電動機1を起動させ、その5秒後にコイル(Coil2)をOFFにして、電動機2を停止させる」という制御を行なう場合が一例として挙げられるが、これをPL/H-C言語を用いて記述すると、図7に示すとおりとなる。同図について説明すると、各々“;”で区切られたものを文といい、処理の最小単位を表わす。最初の文は、WAITというキーワードで始まるのでWAIT文といい、WAITの後ろに記述された条件が成立する(例の場合、リミットスイッチLS1がONになる)のを待ち、条件が成立すると、次の文の実行に移ることを指示する。第2の文は、ON文で、ONの後ろに記述された信号〔例の場合コイル(Coil)〕をONにすることを指示する。第3の文は、DELAY文で、DELAYの後ろに記述された時間(例の場合、5秒)待ってから、次の文の実行に移ることを指示する。第4の文は、OFF文で、OFFの後ろに記述された信号〔例の場合、コイル(Coil2)〕をOFFにすることを指示する。以上で同図のプログラムの処理が、前に述べた例の処理と一致していることが分かるであろう。同図の例からも分かるように、PL/H-Cのプログラムでは、シーケンス制御などの入出力信号(あ

るいはプロセス入出力)の各点に対して名前を付けることができ、プログラム中では、その名前で参照、あるいは入出力の指定ができる。順序制御は、同図に述べたような一連のある順序に従って制御を行なうものの記述のほかに、ある順序処理と別の順序処理を同時に進行させ、それらの二つ以上の順序処理がすべて完了してから、その次の順序処理を行なうような場合の並列処理と、その同期処理についても簡明に記述できる機能がある。

次に、条件制御について例により説明する。例えば、「タンクの水位がHレベルに達すると放流用のバルブVALVE1をあげ、Lレベルに達するとバルブVALVE1を閉じる」という制御の“PL/H-C”プログラム記述は、図8に示すとおりである。同図について説明すると、COND文は、CONDの後ろの条件(例の場合、HあるいはLがONであるか)が成立すると、THEN以後の文(例の場合、ON文あるいはOFF文)を実行し、成立しなければ次のCOND文に移ることを指示する。以上で、同図のプログラムは、前に述べた例の処理を行なうことが分かる。原則として、条件制御は同図に示したように、COND文を並べて記述する。

3.3 “PL/H-C”処理の特徴

“PL/H-C”プログラムは、前にも述べたように順序制御と条件制御とから成るが、その処理内容には、WAIT文の待ち条件、DELAY文の指定時間遅延、タイマ関数の時間計測など、いわゆるリアルタイムモニタの助けがあれば実現できるものがある。ただし、リアルタイムモニタの機能を使

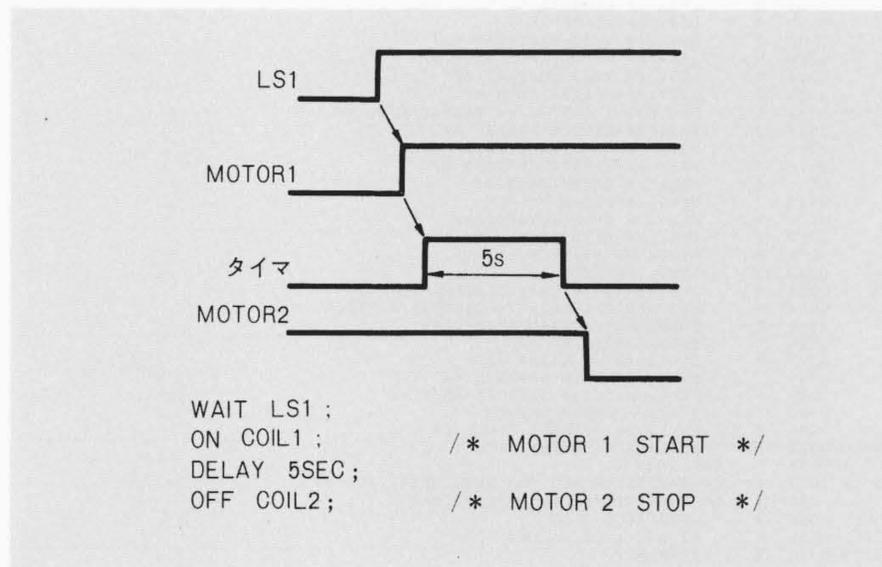


図7 順序制御のPL/H-Cプログラムの例 物の動きに一致したプログラム記述を示す。

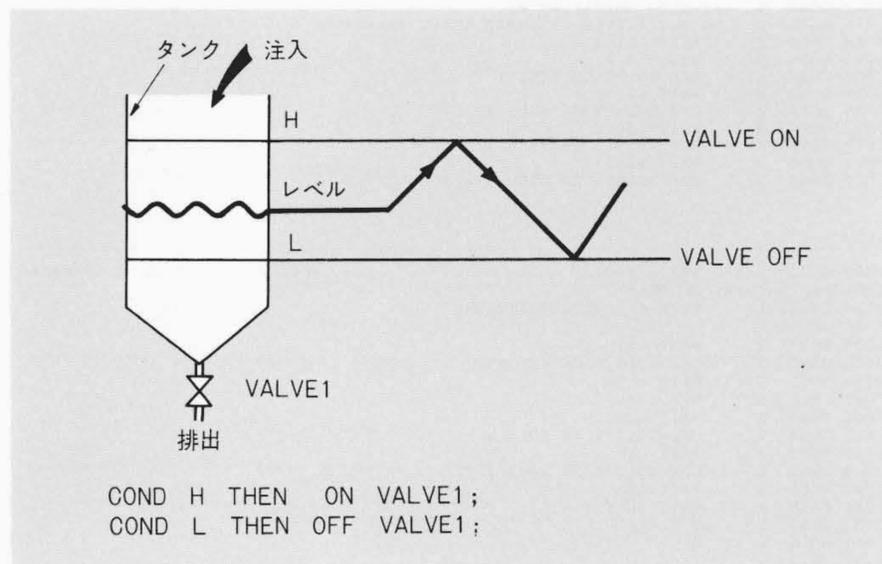


図8 条件制御の“PL/H-C”プログラム例 物の動きに一致したプログラム記述を示す。

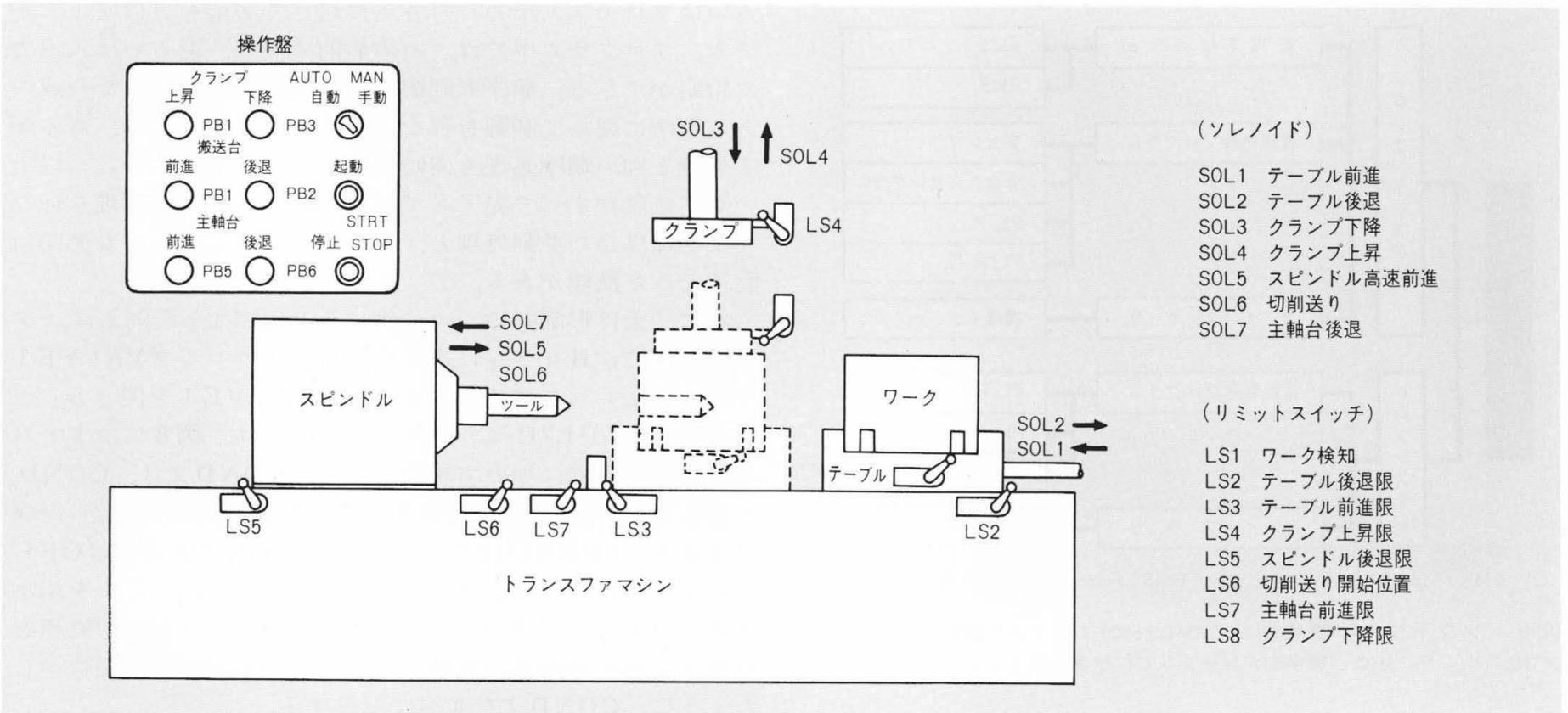


図9 トランスファマシン 簡単な動きをする例題を示す(Allen-Brdley社の次のマニュアルより引用: INSTRUCTION MANUAL, BULLETIN 1750, PMC)。

```

HMCS-6800 PL/H-C VER : 1.0          79/03/15  PAGE 1 *****
TRANSFER MACHINE PROGRAM
STNO.LINO. L      PL/H-C SOURCE STATEMENT

0001 0001 0 /* TRANSFER MACHINE */
0002 0002 0 TRANSMACHINE: DO ;
0003 0003 1  TRAN:SPECIFICATIONS PUBLIC;
0004 0004 2  DEFINE DDI # UNIT 1 AT (#000H) ;
0005 0005 2  PIO DDI 0 (LS1, /* WORK DETECT */
0006 0006 2  LS2, /* CARRIER BACKWARD */
0007 0007 2  LS3, /* CARRIER FORWARD */
0008 0008 2  LS4, /* CLAMP UP */
0009 0009 2  LS5, /* HEAD BACKWARD */
0010 0010 2  LS6, /* MIDDLE POSITION */
0011 0011 2  LS7, /* HEAD FORWARD */
0012 0012 2  LS8); /* CLAMP DOWN */
0013 0013 2  PIO DDI 1 (AUTO, /* SELECT AUTO */
0014 0014 2  MAN, /* SELECT MANUAL */
0015 0015 2  PB1, /* CARRIER ADVANCE */
0016 0016 2  PB2, /* CARRIER RETURN */
0017 0017 2  PB3, /* CLAMP DOWN */
0018 0018 2  PB4, /* CLAMP UP */
0019 0019 2  PB5, /* HEAD ADVANCE */
0020 0020 2  PB6, /* HEAD RETURN */
0021 0021 2  STRT, /* READY */
0022 0022 2  STP); /* STOP */
0023 0023 2  DEFINE DDO # UNIT 2 AT(5006H);
0024 0024 2  PIO DDO 0 (SOL1, /* CARRIER ADVANCE */
0025 0025 2  SOL2, /* CARRIER RETURN */
0026 0026 2  SOL3, /* CLAMP */
0027 0027 2  SOL4, /* UN CLAMP */
0028 0028 2  SOL5, /* HEAD ADVANCE */
0029 0029 2  SOL6, /* HEAD SLOW ADVANCE */
0030 0030 2  SOL7, /* HEAD RETURN */
0031 0031 2  SPM); /* SPINDLE MOTOR */
0032 0032 2  DECLARE RUNSMODE BYTE ;
0033 0033 1  END TRAN ;
0034 0034 1  /* CONDITION SET FOR AUTO OPERATION */
0035 0035 1  DO CONDITIONAL AUTO RESET MAN ;
0036 0036 2  COND STRT THEN
0037 0037 2  IF LS2 & LS4 & LS5
0038 0038 2  THEN DO:
0039 0039 3  RUNSMODE = 1 ;
0040 0040 3  ON SOL2,SOL4,SOL7 ;
0041 0041 2  END ;
0042 0042 2  COND STP THEN
0043 0043 2  DO ;
0044 0044 3  RUNSMODE = 0 ;
0045 0045 3  OFF SOL1,SOL2,SOL3,SOL4,SOL5,SOL6,SOL7,SPM ;
0046 0046 2  END ;
0047 0047 1  END ;
0048 0048 1  /* AUTO OPERATION */
0049 0049 1  DO SEQUENTIAL RUNSMODE RESET #RUNSMODE ;
0050 0050 2  WAIT LS1 ;
0051 0051 2  DELAY 5 ;
0052 0052 2  OFF SOL2; ON SOL1 ;
0053 0053 2  WAIT LS3 ;
0054 0054 2  OFF SOL2 ; ON SOL3 ;
0055 0055 2  WAIT LS8 ;
0056 0056 2  OFF SOL7 ; ON SOL5 ;
0057 0057 2  WAIT LS6 ;
0058 0058 2  OFF SOL5 ; ON SOL6,SPM ;

HMCS-6800 PL/H-C VER : 1.0          79/03/15  PAGE 2 *****
TRANSFER MACHINE PROGRAM
STNO.LINO. L      PL/H-C SOURCE STATEMENT

0059 0059 2  WAIT LS7 ;
0060 0060 2  OFF SOL6 ; ON SOL7 ;
0061 0061 2  WAIT #LS6 ;
0062 0062 2  OFF SPM ;
0063 0063 2  WAIT LS5 ;
0064 0064 2  OFF SOL3 ; ON SOL4 ;
0065 0065 2  WAIT LS4 ;
0066 0066 2  OFF SOL1 ; ON SOL2 ;
0067 0067 2  WAIT LS2 ;
0068 0068 2  WAIT #LS1 ;
0069 0069 1  END ;
0070 0070 0 END ;
0071 0071 0
    
```

図10 自動運転プログラム “PL/H-C”の条件形制御と順序形制御のプログラム例を示す。

うと、オーバヘッドがかかり、実行時間の低下を招き、機器の制御に必要な応答時間に間に合わないことがある。そこで、“PL/H-C”では、そのオブジェクトプログラムは、リアルタイムモニタのもとで動くが、個々のWAIT処理などは、リアルタイムモニタの機能を使わずに実現する方法をとることにした。

それは、スキヤニング方式といわれるもので、オブジェクトプログラムは、リアルタイムモニタから一定周期ごとに起動され、実行すべき内容を調べながら、実行を進めていく方式である。

3.4 シーケンス制御のコーディング例

一つの例として、図9のトランスファマシンのシーケンスプログラムをコーディングした例の部分を図10に示す。プログラムは、入出力とデータの定義、自動運転及び手動運転と三つの部分に分かれている。図10は自動運転部分のプログラムである。

1サイクルの作業時間を縮める必要がある場合には、順序形制御処理の中に並列処理用に開発された、DO PARALLELを使うと可能になる。

3.5 “PL/H-C”コンパイラの性能

レジデントシステムのコンパイラの性能は、コンパイル速度:40~45ステートメント/分、メモリ量と実行スピードのアセンブラ比:1.4~1.6倍程度、1回にコンパイルできるプログラムの大きさ:約700~800ステートメントである。

4 結 言

マイクロコンピュータ“HMCS 6800”を使用した産業用のコントローラとして、RAS機能、取扱い性に優れたハードウェア及び従来ネックになりがちであったソフトウェアでも、産業用の制御に適した高級言語“PL/H-C”が開発された。この産業用マイクロコントローラは、拡大しつつある需要に十分こたえるものと期待される。

ハードウェアでは、入出力関係の拡充、ソフトウェアではアプリケーションの標準パッケージ化を行ない、ソフトウェア生産性の向上と応用範囲の拡大に努めていく考えである。