

生産システムにおけるロボットのペトリネットに基づくフレキシブル制御

安田元一*・野崎宏之**

Petri Net Based Flexible Control of Industrial Robots for Intelligent Manufacturing Systems

YASUDA Gen'ichi and NOZAKI Hiroyuki

This paper presents a Petri net based control scheme for flexible control of robotic manufacturing systems. The experimental system is composed of one five-degree industrial robot and several conveyors. The conveyor system transports workpieces sequentially into specified working places, and the robot handles them to build up a product synchronously with the conveyor system. All of the concurrent processes in the system are defined and represented using Petri nets. The control software for discrete event robotic manufacturing processes is implemented in Visual Basic language. Experimental results of system operations are presented to show the effectiveness of the proposed control scheme.

1. 緒言

1970年代から自動車などの生産ラインに産業用ロボットが導入され、溶接、組立てや検査など、さまざまな作業の省力化、知能化に向けた開発が続けられてきた。工場の生産システムでは1980年代、消費者ニーズに迅速に対応するための多品種少量生産に適した生産技術としてFMS（フレキシブル生産システム）に代表される柔軟なシステム技術が提案され、生産ラインのさまざまな条件の変更がハードウェアの変更なしにソフトウェアによって対応可能になった。さらに近年では、FMSやCIM（コンピュータ統合生産）に代表される生産ラインの情報化、システム化の進行とともに、大規模ソフトウェアシステムとしての問題点が顕在化し、生産システムにおける分散知能や協調制

御の考え方に基づくIMS（知的生産システム）¹⁾が次の段階の生産システムとして期待されるようになってきている。

以上のようなロボットと生産システムの知能化への進化を背景として、本研究は生産システムを構成するロボット、コンベアなど複数の機器をPC（パソコン）により統一的に状態監視し、機器間の連携がとれたシステム全体の統合制御を遂行するフレキシブルな生産ライン制御システムを構築することを目的とする。

2. 実験システムの構成

生産ラインを模擬するために、アームロボット1台とその周囲を1周するパレット搬送コンベアを使用した。アームロボットとコンベアは連動して部品を移送し、コンベアの定位置で作業を実施

* 情報学部 知能情報学科 教授

**不動技研工業(株)

2008年3月28日受付

する仕様となっている。図1に実験システムの外観を示す。

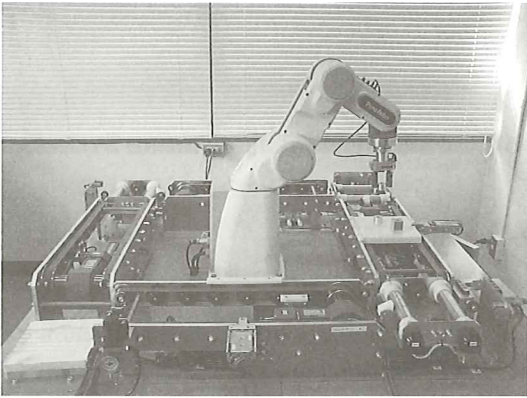


図1 実験システムの外観

2.1 アームロボット

アームロボットとして5自由度垂直多関節ロボット（松下電器製 PanaRobo KS-V20）を使用した。ロボットコントローラ（図2）の操作モードとして、①パネルモード、②ティーチングボックスモード、③コンソールモード、④シーケンスモードがある²⁾。①はコントローラ前面パネルの操作スイッチを使用して登録済みのプログラムの運転、停止などの操作を行う。②はティーチングボックス（図3）を接続して、位置（ポーズ）データ登録、プログラム No. 選択、各軸のキーによる手動操作などを行う。③はコンピュータと接続し、コンピュータで作成した制御プログラムの転送や、ダイレクトコマンドによる位置データ登録・プログラム No. 選択・運転/停止などの操作を行う。④はシーケンサ（シーケンスコントローラ、プログラマブルコントローラ）と接続し、シーケンサとの入出力信号によるシーケンス制御を行う。今回は搬送コンベアとの運動を行うためシーケンスモードを使用する。

アームの手先には電動ハンド（図4）を取り付け、ハンドコントローラ（図5）が位置検出用近接スイッチ信号を用いた開閉動作の制御を行う。なお、ロボットを運転するには専用のロボット言



図2 ロボットコントローラ



図3 ティーチングボックス

語でかかれたプログラムがロボットコントローラに転送されていることと、プログラム内で使用している位置データがティーチングされていることが必要である。

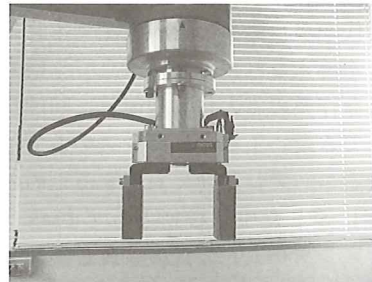


図4 電動ハンド



図5 ハンドコントローラ

2.2 パレット搬送コンベア

アームロボットと組み合わせ小規模の生産システムとして設置されたもので、メインコンベア、コーナーコンベア、ターンテーブル、パレットストッパ、パレット位置決めユニットの動作部に分けられる。それらはコンベア制御用シーケンサによって制御されている。コンベアの運転モードとして自動/手動モード、オンライン/オフラインモードがあり、手動でオフラインモードのときコンベア操作ボックスから各動作部を操作できる。自動でオフラインモードのときパレット搬送コンベア単体として一連の動作を実行する。ただしパレット位置決めユニットは動作せずパレットは停止せずに通過する。自動でオンラインモードのときシーケンサを用いた制御によりアームロボットと連動して作業を実行する。

(1) メインコンベア

実験システムには同じサイズの主コンベアが4台あり、実験ではそのうち3台を使用する。メインコンベア1台(図6)のサイズは長さ610mm幅160mmで、ACギヤ付モータでベルトコンベアを一定方向に駆動し、パレットを搬送する。パレットストッパ(図7)は空気圧駆動により先端部が上下に動いてパレットの停止と排出を行う。パレットストッパはシーケンス制御開始と同時に上限となり、メインコンベア上のパレットで定位位置センサ(図8)を感知すると下限となる。アクリル製パレット一辺の窪みに空気圧駆動された位



図7 パレットストッパ

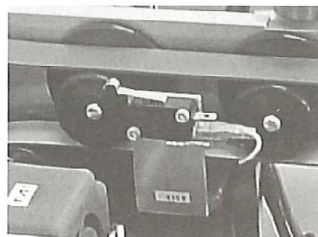


図8 定位位置センサ (マイクロスイッチ)

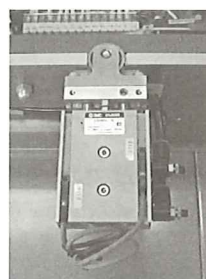


図9 パレット位置決めユニット

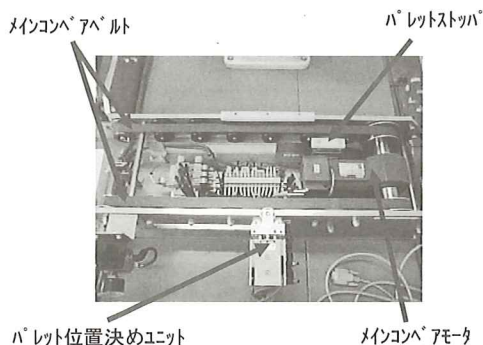


図6 メインコンベア

置決めユニット(図9)の先端部が入り込むことで、所定位置でのパレットの位置決めを行う。アームロボットからの“作業完了”信号を受信するとパレット位置決めユニットは元に戻る。

(2) コーナーコンベアとターンテーブル

メインコンベアから搬送されたパレットをモータ駆動のコーナーコンベアが正転動作で受け入れ、ターンテーブル上から少し通り過ぎたところでパレット付属の金属片を近接センサ(KEYENCE製EV-118M、高周波発振型)で感知し、逆転動作に

一定時間切り換え、止め具によりパレットをターンテーブル上に固定する。次にエア駆動によるターンテーブルの90度方向転換完了をセンサで感知した後、コーナーコンベアが正転動作でパレットを次のメインコンベアに搬出する。メインコンベア上でパレット位置決め完了後、ターンテーブルは元に戻る。

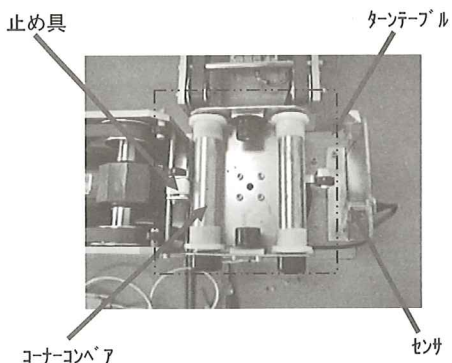


図10 ターンテーブル

上記のコンベアとは別に搬入コンベア2台、搬出コンベア1台を考え、これらは既存のシステムの状態とは無関係に搬入、搬出の作業を行うものとする。搬入コンベア2台はNo.2コンベア定位置付近に配置し、No.2コンベア上での作業対象

となる部品 A, B を搬入する。搬出コンベア1台はNo.3コンベア定位置付近に配置し、No.3コンベア上にて作業完了した部品 C を搬出する。搬入コンベア、搬出コンベアともにコンベア上で流すことのできる部品は1個のみとし、これらの部品は模擬信号で表す。また、No.1コンベアおよびNo.3コンベア上での作業対象部品をO1, O3としてそれぞれの保管場所を設定する。図11に実験対象とする生産ライン全体の構成を示す。

2.3 シーケンス制御システム

シーケンスモード時、ロボットコントローラは外部入出力信号により外部機器と連携したシーケンス動作を行う。ロボットと外部機器との連携のためのシーケンサとして、OMRON 製 SYSMAC C60H (図12) を使用した³⁾。

シーケンサからロボットコントローラへの入力信号は、シーケンサ操作ボックスで登録されたプログラム No. を指定するための“ロボット番号指定”信号、アームロボットがコンベア定位置で作業開始をする際に必要な“パレット位置決定完了”信号、シーケンサ側からの運転開始/停止指令およびエラーの解除のための“START”, “STOP”, “RESET” 信号があり、ロボットコントローラ

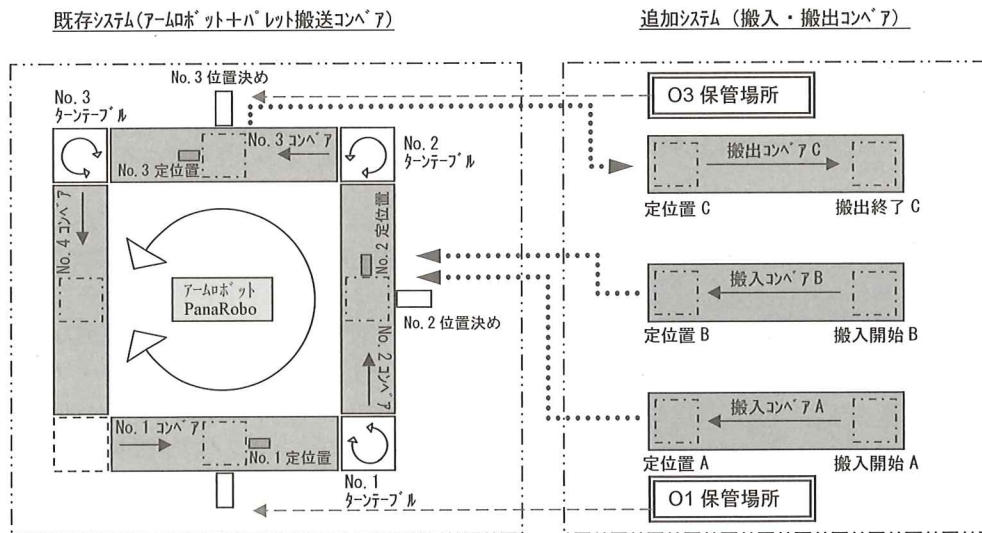


図11 実験対象生産ライン全体の構成

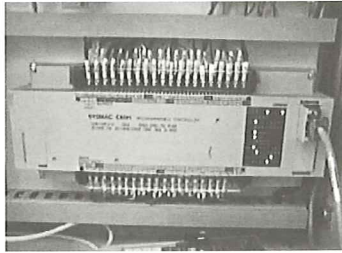


図12 シーケンサ
(OMRON SYSMAC C60H)

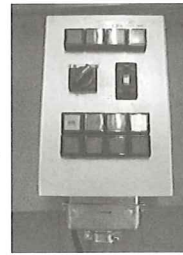


図13 シーケンサ操作ボックス

からの出力信号には、アームロボットの制御プログラムから電動ハンドを制御するための“ハンドCLOSE/OPEN”信号、コンベアが定位置からパレットを搬送する際に必要な“作業完了”信号、

シーケンス制御開始前の条件であるアームロボットの原点検出後の“ORIGIN”信号，シーケンス制御の条件であるアームロボット運転中を示す“RUN”信号，アームロボット側でエラーが発生した際にシーケンサ制御を中断するための

INPUT	端子番号	リレー内容
0ch	0	IN/O 自動スイッチ ON
	1	RST/1 運転スイッチ ON
	2	002 停止スイッチ ON
	3	003 リセットスイッチ ON
	4	004 ロボット定位置 ON
	5	005 ハンド ON(手動モード時)
	6	006
	7	007
	8	008 強制完了 ON(手動モード時)
	9	009 ディップスイッチ(プログラム番号選択)
	10	010 ディップスイッチ(プログラム番号選択)
	11	011 ディップスイッチ(プログラム番号選択)
	12	012 ディップスイッチ(プログラム番号選択)
	13	013 ハンド CLOSE
	14	014 ハンド SLOW
15	015 ハンド STOP	
1ch	0	100 1:ハンド CLOSE/0:ハンド OPEN [OUT(1)]
	1	101 1:作業完了 [OUT(2)]
	2	102 1ビットデータ-3 [OUT(3)]
	3	103 1ビットデータ-4 [OUT(4)]
	4	104 1ビットデータ-5 [OUT(5)]
	5	105 ORGIN(原点検出後 ON)
	6	106 TROUBLE(エラー発生時 ON)
	7	107 PEND 信号 [OUT(16)]
	8	108 RUN(プログラム実行中)
	9	109
	10	110
	11	111 パレット位置決完-3
	12	112 コンベア ON LINE 入力
	13	113 パレット位置決完-1
	14	114 コンベア異常入力
15	115 パレット位置決完-2	

OUTPUT	端子番号	リレー内容
2ch	0	200 運転ランプ ON
	1	201 停止ランプ ON
	2	202 リセットランプ ON
	3	203 ロボット定位置ランプ ON
	4	204 ロボットトラップランプ ON
	5	205
	6	206 コンベア運転ランプ ON
	7	207 コンベア ONLINE ランプ ON
	8	208 ハンド CLOSE
	9	209 ハンド SLOW
	10	210 ハンド STOP
	11	211
	12	
	13	
	14	
15		
3ch	0	300 プログラム番号指定(1ビットデータ)-1 [INP(1)]
	1	301 プログラム番号指定(1ビットデータ)-2 [INP(2)]
	2	302 プログラム番号指定(1ビットデータ)-3 [INP(3)]
	3	303 プログラム番号指定(1ビットデータ)-4 [INP(4)]
	4	304 コンベア位置決出-1 [INP(7)]
	5	305 コンベア位置決出-2 [INP(8)]
	6	306 コンベア位置決出-3 [INP(9)]
	7	307
	8	308
	9	309 RESET(エラー解除/プログラム中断解除)⇒初期状態
	10	310 START(指定プログラム起動)
	11	311 MEND 信号 [INP(16)]
	12	312 STOP(実行中の命令行中断,原点復帰中断)
	13	313 コンベア自動起動
	14	314 ロボット作業完了-1
15	315 コンベア非常停止	

※ 塗潰しは from/toの相手を示す。

: シーケンサ操作ボックス
 : ハンドコントローラ
 : トラップコントローラ
 : コンベアシーケンサコントローラ

図14 シーケンサ入出力リレー内容

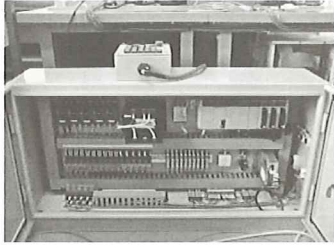


図15 コンベア制御盤

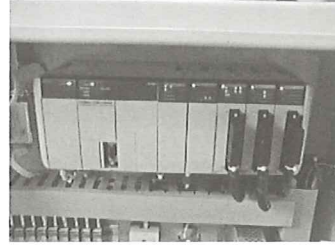


図16 コンベア制御用シーケンサ (OMRON SYSMAC CQM1)

“TROUBLE” 信号がある。

シーケンサ操作ボックス (図13) はシーケンサに対して、自動モード時に指令 (運転/停止/リセット) 入力やディップスイッチによるロボットプログラムの選択を行い、シーケンサの状態信号を出力する。シーケンサはハンドコントローラとの間で、指令信号と電動ハンド状態信号の入出力も行う。手動モード時には電動ハンドの単独閉閉

信号の出力が可能である。入出力リレー信号の詳細を図14に示す。

4台のコンベアの制御はコンベア制御盤 (図15) のシーケンサ (OMRON 製 SYSMAC CQM1, 図16) によって実施した。ロボットとコンベアの連携用のシーケンサはコンベア制御盤に対してアームロボットとのインターロック的な役割を持

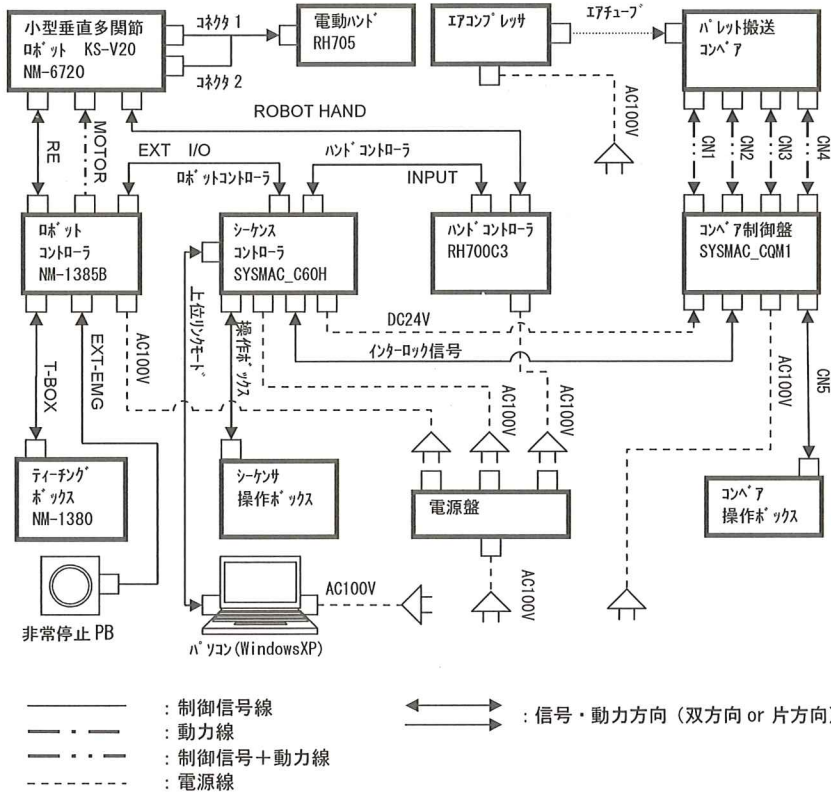


図17 実験システムの機器構成

つ制御信号を入出力する。シーケンサへの入力信号としては、アームロボットがコンベア定位置で作業開始する条件となる“パレット位置決定完了”信号，コンベアがシーケンス動作のモードであることを示す“ON-LINE モード”信号，シーケンス動作を停止させるための“コンベア異常”信号，シーケンサからの出力信号としては，運転開始指令をコンベアに伝える“コンベア自動起動”信号，コンベアが定位置からパレットを搬送する際に必要な“ロボット作業完了”信号，シーケンサおよびアームロボット側の何らかの原因でシーケンス動作を停止するための“コンベア非常停止”信号がある。実験で使用する機器全体の構成を図17に示す。

3. ペトリネットによるシステムの動作表現

今回実験対象とした生産ラインにおいて，アームロボットと搬送コンベアに関しては指定されたシーケンス動作を行うものとし，追加したコンベア上での作業対象物の有無を条件としてシーケンス動作が運転可能かどうかを判定するものとする。アームロボットと搬送コンベアの連携動作は入力リレー信号を用いたシーケンサの制御プログラムの記述により可能である。

システムの動作仕様は以下のとおりとする。まず，運転開始前はアームロボットはホームポジションにあり，パレット搬送コンベアではパレットがNo.1コンベアの定位置より手前にセットされ，さらに保管場所には部品O1が置かれているものとする。シーケンサ操作ボックスの運転スイッチONで，アームロボットは指定されたプログラムNo.の動作を実行する。同時にパレット搬送コンベアはNo.1コンベア上でパレット搬送を開始する。コンベア定位置までパレットが搬送されると，パレットは位置決めユニットにより固定される。アームロボットはNo.1コンベア定位置までO1を運び，そこでO1に対して作業を開始する。ロボットから“作業完了”信号を入力する

と位置決めユニットはパレットを解放し，搬送を継続する。アームロボットは，搬入コンベアの定位置にA, Bのどちらかが有る場合，それをNo.2コンベア定位置に運び作業を行う。A, Bがともに有る場合はAを優先し，どちらも定位置にない場合はNo.1コンベア定位置での作業完了の状態待機し，AかBが搬入コンベアの定位置上で検知された時点で運転を再開する。No.2コンベア定位置での作業完了のタイミングで部品O3保管場所でのO3の有無を判定し，O3が有る場合にアームロボットはNo.3コンベア定位置に運び作業を行う。O3が保管場所になれば待機し，保管場所で検知された時点で運転を再開する。No.3コンベア定位置での作業完了の時点で，作業完了した部品Cを搬出コンベア定位置に運ぶ。前回の作業完了部品Cが搬出コンベア定位置に有るときは作業完了の状態待機し，搬出が検知された時点で運転を再開する。その後，アームロボットは運転前の状態に戻り，パレット搬送コンベアがNo.4コンベアの定位置にパレットが到着した時点でシーケンス動作は終了となる。

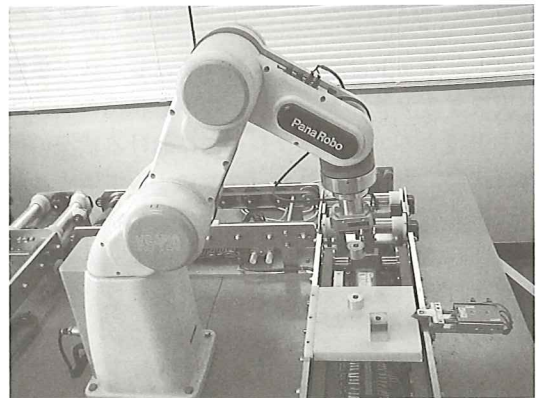


図18 No.3コンベア定位置で作業中のロボット

以上の実験システムの動作シーケンスを，ペトリネットモデルを用いて表現する。ペトリネット⁴⁾は非同期性，順序性，並列同時進行性，競合などの性質を持つシステムの表現に用いられ，事象と状況からなるグラフでモデル化し，このモデルが動くことが制御系の動作に対応する。

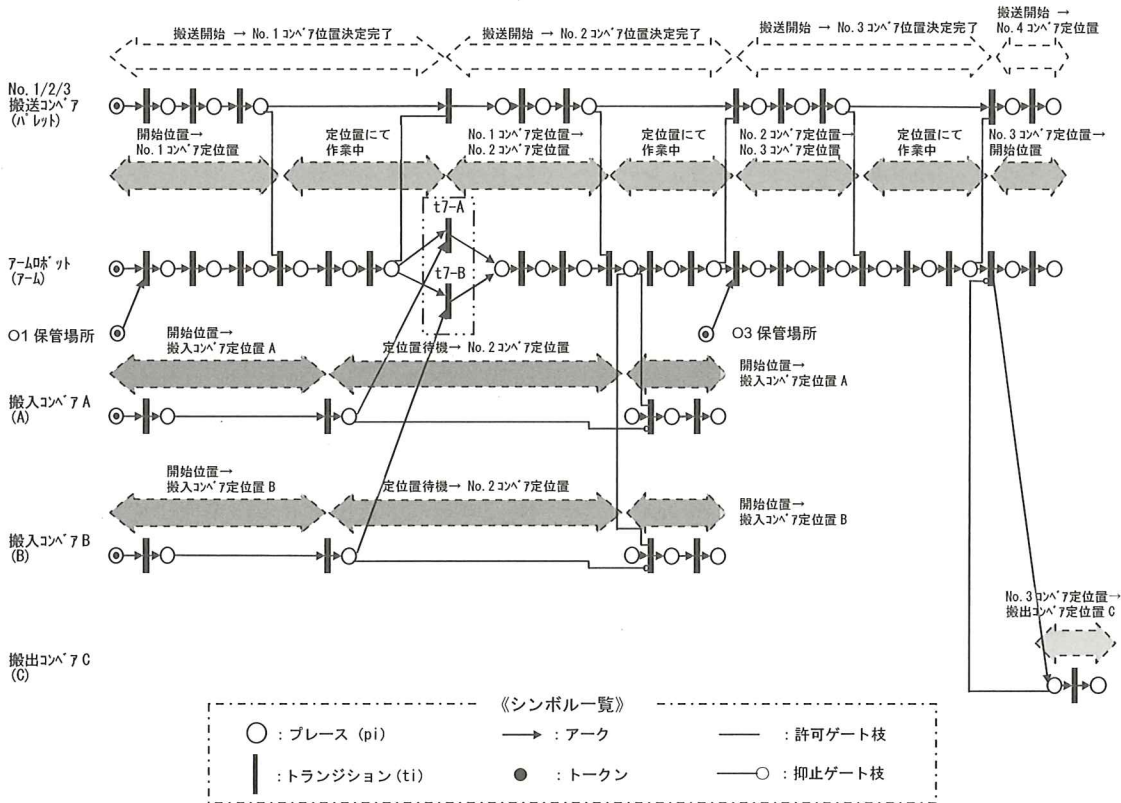


図19 実験システムのペトリネット表現

ペトリネットは状況を表現するプレース、事象を表現するトランジション、事象と状況の前後関係を表現するアーク、プレースにおける状況の保持(状態)を表現するトークンによって構成され、システムの挙動は、事象の発生による状態の遷移で示される。本研究ではプレースはアームロボットや各コンベアの状況、トランジションは動作開始/完了の事象、アークはアームロボットや各コンベアの動作順序、トークンはロボットアーム、パレット、作業対象となる部品 O1, O3, A, B, C の存在を表すものとする。シーケンス制御系特有の外部機器との信号入出力機能のために、許可ゲート枝と抑止ゲート枝をトランジションの発火条件として加えた⁵⁾。許可ゲート枝は信号が“1”の場合に、接続されているトランジションの発火を許可し、抑止ゲート枝は信号が“1”の場合に、接続されているトランジションの発火を抑止する

働きをする。事象の発生に対応するトランジションの発火則は、そのすべての入力プレースがトークンを持ち、すべての許可ゲート枝が“1”で、すべての抑止ゲート枝“0”の場合であり、発火すると、入力トークンは消滅しその出力プレースにトークンが発生する。図19に実験システムのペトリネット表現を示す。図20にロボットと各コンベアの動作シーケンスの説明を示す。

4. 統合制御ソフトウェアの実現

コンピュータを用いて生産ラインの状態監視を行い、システム全体の統合制御を行う方法を検討し制御プログラムを作成した。システム統合のための上位コンピュータとしてパソコン(HP Compaq nx6120, CPU Pentium M 2.0GHz, OS Windows XP)を使用し、開発ソフトは Visual Basic 2005を

アームロボット(アーム)		No.1/2/3コンベア(パレット)		搬入コンベアA(A)		搬入コンベアB(B)		搬出コンベアC(C)				
0	ロボット待機場所	p0	パレット待機場所	p100	搬入開始位置 A	p200	搬入開始位置 B	p300				
1	O1 保管場所へ移動	t1	待機場所からパレット搬送	t101	開始位置から A 搬送	t201	開始位置から B 搬送	t301				
2	O1 保管場所	p1	パレット搬送中	p101	A 搬送中	p201	B 搬送中	p301				
3	O1 を確保	t2	コンベア 1 定位置にパレット到着	t102	定位置に A 到着	t202	定位置に B 到着	t302				
4	O1 ポイント	p2	パレット定位置	p102	A 定位置	p202	B 定位置	p302				
5	O1 をコンベア 1 定位置へ移動	t3										
6	O1 コンベア 1 定位置	p3	コンベア 1 でパレット位置定	t103								
7	O1 を解放	t4	コンベア 1 でパレット位置決定完了	p103								
8	O1 パレット上-1	p4										
9	O1 作業開始	t5										
10	O1 作業中	p5										
11	O1 作業完了	t6										
12	O1 パレット上-2	p6										
13	搬入コンベア A(B) 定位置へ移動	t7	コンベア 1 定位置からパレット搬送	t104								
14	搬入コンベア A(B) 定位置	p7	パレット搬送中	p104								
15	A(B) を確保	t8	コンベア 2 定位置にパレット到着	t105								
16	A(B) ポイント	p8	パレット定位置	p105	A 定位置	p203	搬入開始位置 B	p303				
17	A(B) をコンベア 2 定位置へ移動	t9										
18	A(B) コンベア 2 定位置	p9	コンベア 2 でパレット位置決定	t106					搬入開始位置 A	p203	搬入開始位置 B	p303
19	A(B) を解放	t10	コンベア 2 でパレット位置決定完了	p106					開始位置から A 搬送	t204	開始位置から B 搬送	t304
20	A(B) パレット上-1	p10							A 搬送中	p204	B 搬送中	p304
21	A(B) 作業開始	t11							定位置に A 到着	t205	定位置に B 到着	t305
22	A(B) 作業中	p11										
23	A(B) 作業完了	t12										
24	A(B) パレット上-2	p12										
25	O3 保管場所へ移動	t13	コンベア 2 定位置からパレット搬送	t107					A 定位置	p205	B 定位置	p305
26	O3 保管場所	p13	パレット搬送中	p107								
27	O3 を確保	t14	コンベア 3 定位置にパレット到着	t108								
28	O3 ポイント	p14	パレット定位置	p108								
29	O3 をコンベア 3 定位置へ移動	t15										
30	O3 コンベア 3 定位置	p15	コンベア 3 でパレット位置決定	t109								
31	O3 作業開始	t16	コンベア 3 でパレット位置決定完了	p109								
32	O3 作業中	p16										
33	C 作業完了	t17										
34	C パレット上-2	p17										
35	搬出コンベア C 定位置へ移動	t18	コンベア 3 定位置からパレット搬送	t110								
36	搬出コンベア C 定位置	p18	パレット搬送中	p110	※定位置 A から No.2コンベアに移動後、次の A を搬入可能とする	※定位置 B から No.2コンベアに移動後、次の B を搬入可能とする		C 定位置	p400			
37	待機場所へ移動	t19	コンベア 4 定位置にパレット到着	t111					C 搬送	t401		
38	待機場所	p19	パレット定位置	p111					搬出終了位置 C	p401		

図20 実験システムの動作シーケンス

用いた。シーケンサにはRS-232C インタフェースが内蔵してあり、使用モードとして ①上位リンクモード、②ユーザメモリダウンロード/アップロードモード、③ASCII 入出力モードがある。①の上位リンクモードでコンピュータと接続することにより、運転状態や入出力リレーの動作状態の監視を行う。

4.1 シーケンサとの通信

コンピュータとシーケンサとのデータの受け渡しは、コンピュータがコマンドを発行し、それに対してシーケンサが応答を返す方式となっている。コマンドブロックとその応答のレスポンスブロックは、“号機 No.”(シーケンサが持っている番号

で『00』～『31』で指定)，“ヘッダ”(アルファベット 2 文字の指令コード)，“テキスト”(指令または応答の内容)，“FCS”(伝送上の誤りチェックのために、1つのフレームの最初からテキストの最終文字までの排他的論理和をとった 8 ビットデータを ASCII の 2 文字に変換したもの)，“ターミネータ”(CR キャリッジリターン) で構成されている。また、送受信の単位である 1 フレームは 131 文字以下のブロックで構成され、それ以上は複数フレームとなり、1 フレーム送出ごとにターミネータとデリミタによって送信権を移行する。応答速度は伝送速度、データ容量、シーケンサのサイクルタイムにより異なる。また、RS-232 C インタフェースのサービス時間を設定すること

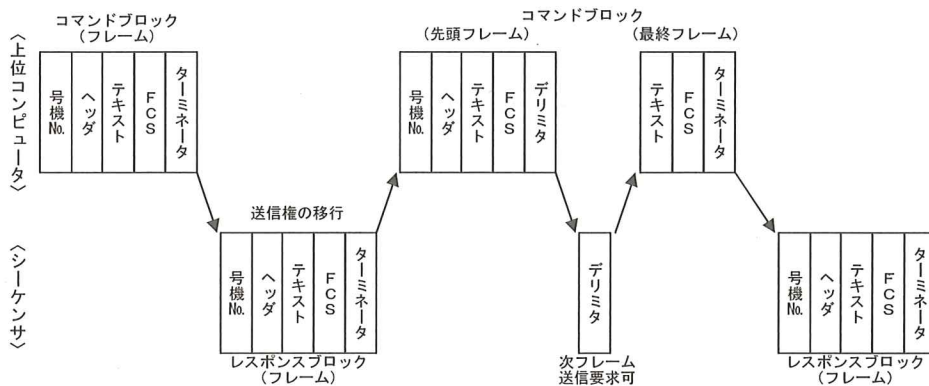


図21 コマンド送信手順例

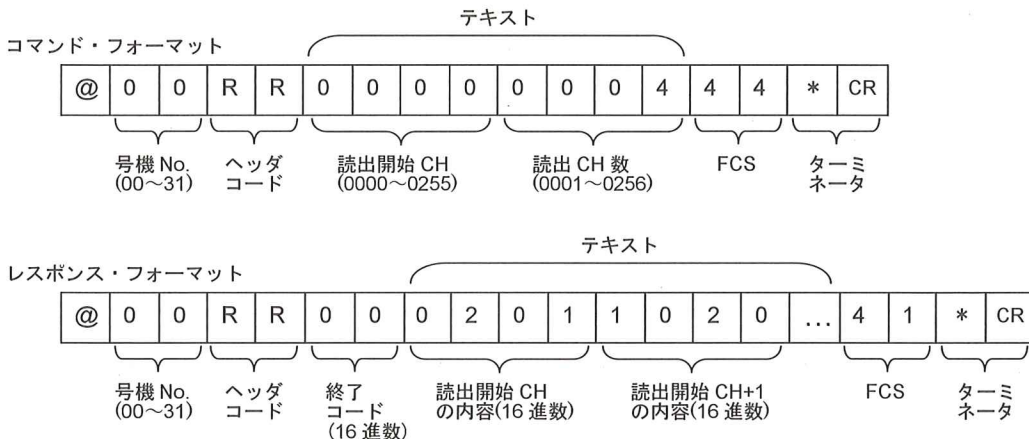


図22 入出力リレー状態読出しのブロックフォーマット

```

' フォームのLordイベントハンドラ
Private Sub RS232CForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

    With ComSerialPort
        .PortName = "COM1"           ' 通信ポートはCOM1
        .BaudRate = 9600             ' 通信速度はbps
        .DataBits = 7               ' データ長はビット
        .Parity = 10.Parity.Even    ' パリティチェック偶数
        .StopBits = 10.Parity.Two  ' ストップビットは
        ' ハードウェアによるハンドシェイク
        .Handshake = 10.Parity.Handshake.RequestToSend
        .RtsEnable = True           ' RTSラインを有効にする
    End With
End Sub

' 通信開始ButtonのClickイベントハンドラ
Private Sub ConnectButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ConnectButton.Click
    Try
        ' 通信ポートが開いているかどうかの判定
        If ComSerialPort.IsOpen = False Then
            ' 通信ポートを開く場合の処理
            ComSerialPort.Open()      ' 通信ポートを開く
            ConnectButton.Text = "通信停止"
        Else
            ' 通信ポートを切断する場合の処理
            ComSerialPort.Close()     ' 通信ポートを切断
            ConnectButton.Text = "通信開始"
        End If
    Catch ex As Exception
        ' 通信ポートの接続・切断時のエラー処理
        MessageBox.Show(ex.Message, "接続エラー", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
    End Try
End Sub

```

図23 シリアル通信の開始・停止プログラムの例

ができ、これを大きく設定すると応答時間も早くなる。コマンド送信手順の例を図21に示す。実際に使用する入出力リレー状態の読出しについて、コマンドとレスポンスのブロックフォーマットを図22に示す。

4.2 制御プログラム

制御プログラムでは、まずシーケンサの入出力リレー状態を読み出す。次に、読み取った内容からアームロボットとコンベアの状態を把握し、次の実行ステップへの条件判定を行う。さらにその判定に従い、アームロボットとコンベアへの動作指令を出す。また、シーケンス動作の開始・停止の指令もコンピュータから出すこととする。

(1) シリアル通信開始・停止

シーケンサとのシリアル通信の設定をシーケンサに合わせて記述した。また、通信の開始および停止は通信ポートの状態を調べてから実施するようにし、動作が完了しなかった場合の例外処理も

記述した。プログラム例を図23に示す。

(2) シーケンサの入出力リレー状態の読出し

シーケンサ入出力リレー読出開始/停止ボタンを設定し、通信ポートが開いているかどうかを判定して読出開始/停止を実行するようにした。またコマンド送信が完了しなかった場合の例外処理も記述した。入出力リレー読出しコマンドをシーケンサのチャンネル (CH) を指定して送信することにより、シーケンサから応答としてリレー状態の内容が CH 毎に16進数で表されて返ってくる。リレー状態はシーケンス動作とともに変化していくため、いつでも条件判定が可能なようにコマンドを繰り返し送信する必要がある。今回は Visual Basic の "Timer" 要素を使用し20ms間隔でコマンドを送信することで安定して入出力リレーの読出しを行うことができた。なお、4 CH 分の入出力リレー読出しでシーケンサのサイクルタイムは6 msec, RS-232C インタフェースのサービス時間は5%に設定している。プログラム例を図24に示す。


```

' 読出開始ButtonのClickイベントハンドラ
Private Sub IORelayButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles IORelayButton.Click

    ' 通信ポートが開いているかどうかの判定
    If ComSerialPort.IsOpen Then

        If IORelayButton.Text = "読出開始" Then
            IORelayButton.Text = "読出停止"
            IORelayTimer.Enabled = True
        Else
            IORelayButton.Text = "読出開始"
            IORelayTimer.Enabled = False
        End If
    Else
        MessageBox.Show("通信が開始されていません", "送信エラー",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
End Sub

' 読出開始ButtonのTimer_Tickイベントハンドラ
Private Sub IORelayTimer_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles IORelayTimer.Tick

    Dim Command As String = "@00RR0000000444*"

    Try
        SendDataTextBox.Text = ""

        ' データの送信
        ComSerialPort.WriteLine(Command & vbCrLf)
        SendDataTextBox.Text &= Command

    Catch ex As Exception
        ' 送信エラー時の処理
        MessageBox.Show(ex.Message, "送信エラー", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
    End Try
End Sub

```

図24 入出力リレー状態読出しプログラムの例

(3) 条件判定

条件判定のために、まずコマンドに対する応答の処理が必要となる。Visual BasicにおいてSystem.IO.Ports.SerialPortクラスでのデータ受信のためDataReceivedイベントでコマンド送信に対する応答データを取得した後、DelegateステートメントとInvokeメソッドを使用してコントロールを作成したスレッドを呼び出し、フォーム上のテキストボックスに応答の終了コードや返信内容（入出力リレー状態など）のテキストデータを表示させた。

条件判定プログラムでは、シーケンサを付属の操作ボックスから操作し、No.1~3コンペア定位置上での作業完了となる条件をシーケンサ本体の入出力リレー表示ランプからあらかじめ読み取り、それをCH毎に16進数に変換して条件とした。入出力リレー読出しの内容が条件に一致し、さらに

追加コンペア上でのA, B, Cの有無が整ったときに条件成立とし、判定表示枠に“OK!!”と表示するようにした。また、不成立時には“bad...”と表示した。応答については最新と履歴を別々に表示枠に表示した。プログラム例を図25に示す。

(4) 運転開始・停止指令および条件判定後の動作指令

シーケンス動作の運転開始・停止については、入出力リレー書込みコマンドまたは多点強制セット/リセットコマンドが使用可能である。運転開始の場合、アームロボット側は“START”信号のOFF→ON検出後、50msec以内にプログラムNo.指定入力データを読み取り、そのNo.のプログラムの実行を開始し“RUN”信号をONにする。入出力リレー書込みコマンドの場合、シーケンサのサイクル時間の設定により、書き込まれる時間が50msecより極端により短いとき、プログラムNo.

```

' デリゲート型の宣言
Delegate Sub RecieveDataDelegate(ByVal RecieveData As String)
Private RecieveData As String

' デリゲート型のプロシージャを宣言
Private Sub SetRecieveData(ByVal DataString As String)
    If Microsoft.VisualBasic.Left(DataString, 5) = "@00RR" Then
        RecieveDataTextBox.Text &= DataString & vbCrLf
        RecieveNewDataTextBox.Text = DataString & vbCrLf
    ElseIf Microsoft.VisualBasic.Left(DataString, 5) = "@00FK" Then
        RunStopRcvTextBox.Text &= DataString & vbCrLf
        RunStopNewRcvTextBox.Text = DataString & vbCrLf
    Else
        OtherDataTextBox.Text &= DataString & vbCrLf
    End If
End Sub

' デリゲート型の宣言
Delegate Sub CondljudgDelegate(ByVal Condljudg As String)
Private Condljudg As String

' デリゲート型のプロシージャを宣言
Private Sub SetCondljudg(ByVal CondljudgString As String)
    If CondljudgLabel.InvokeRequired Then
        Dim d1 As New CondljudgDelegate(AddressOf SetCondljudg)
        CondljudgLabel.Invoke(d1, New Object() {CondljudgString})
    Else
        CondljudgLabel.Text = CondljudgString
    End If
End Sub

        .
        . (条件 2~4 も同様)
        .

```

```

' SerialPortコントロールのDataReceivedイベントハンドラ
Private Sub ComSerialPort_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles ComSerialPort.DataReceived

    ' デリゲート型の変数の宣言と生成
    Dim recieve As New RecieveDataDelegate(AddressOf SetRecieveData)

    ' 呼び出しの末尾を解釈するための値を設定
    ComSerialPort.NewLine = vbCrLf

    Try
        ' データの受信
        RecieveData = ComSerialPort.ReadLine

    Catch ex As Exception
        ' 受信エラー時の処理
        RecieveData = ex.Message
    End Try

    ' デリゲート型のプロシージャを呼び出す
    Invoke(recieve, RecieveData)

    ' 次の動作開始の条件判定
    ' デリゲート型の変数の宣言と生成
    Dim condl As New CondljudgDelegate(AddressOf SetCondljudg)

    Condljudg = ""
    If RecieveData = CondlLabel.Text And O1KeepCheckBox.Checked Then
        Condljudg = "OK!!"
    Else
        Condljudg = "bad..."
    End If

    ' デリゲート型のプロシージャを呼び出す
    Invoke(condl, Condljudg)

        .
        . (条件2~4も同様)
        .

End Sub

```

図25 条件判定プログラムの例

```

' 運転開始Button_Clickイベントハンドラ
Private Sub SeqRunButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SeqRunButton.Click

    Dim RunCommand As String = "@00FKC10 000000000000000000502D*"
    Dim RunResetCommand As String = "@00FKC10 0000000000000000008020*"
    Dim StopCommand As String = "@00FKC10 000000000000000000502D*"
    Dim StopResetCommand As String = "@00FKC10 0000000000000000008020*"
    Dim n As Integer
    Dim RT As Integer = 1000000

    ' 通信ポートが開いているかどうかの判定
    If ComSerialPort.IsOpen Then

        Try
            RunDataTextBox.Text = ""
            StopDataTextBox.Text = ""

            ' データの送信
            If SeqRunButton.Text = "運転開始" Then

                ComSerialPort.WriteLine(RunCommand & vbCrLf)
                RunDataTextBox.Text &= RunCommand

                ' 解除コマンド送信用タイマ(運転コマンドのレスポンス受信後に送信する)
                n = 0
                Do While n <= RT
                    n = n + 1
                    If n > RT - 1 Then
                        ComSerialPort.WriteLine(RunResetCommand & vbCrLf)
                        SeqRunButton.Text = "運転停止"
                    End If
                Loop
            Else
                ComSerialPort.WriteLine(StopCommand & vbCrLf)
                StopDataTextBox.Text &= StopCommand

                ' 解除コマンド送信用タイマ(停止コマンドのレスポンス受信後に送信する)
                n = 0
                Do While n <= RT
                    n = n + 1
                    If n > RT - 1 Then
                        ComSerialPort.WriteLine(StopResetCommand & vbCrLf)
                        SeqRunButton.Text = "運転開始"
                    End If
                Loop
            End If

            Catch ex As Exception
                ' 送信エラー時の処理
                MessageBox.Show(ex.Message, "送信エラー", MessageBoxButtons.OK,
                MessageBoxIcon.Error)
            End Try

            Else
                MessageBox.Show("通信が開始されていません", "送信エラー",
                MessageBoxButtons.OK, MessageBoxIcon.Error)
            End If
        End Sub

```

図26 運転開始/停止ボタンによる動作指令プログラムの例


```

'条件判定後の動作
If RecieveData <> "@00RR00020110200088000141*" Then
    :
    (運転開始/停止 PB による動作指令プログラム)
    :
ElseIf Condi judLabel.Text = "OK!!" Then
    ComSerialPort.WriteLine(RunCommand & vbCr)
    RunDataTextBox.Text &= RunCommand

'解除コマンド送信用タイマ (運転コマンドのレスポンス受信後に送信する)
n = 0
Do While n <= RT
    n = n + 1
    If n > RT - 1 Then
        ComSerialPort.WriteLine(RunResetCommand & vbCr)
        SeqRunButton.Text = "運転停止"
    End If
Loop
End If
    
```

図27 条件判定後の動作指令プログラムの例

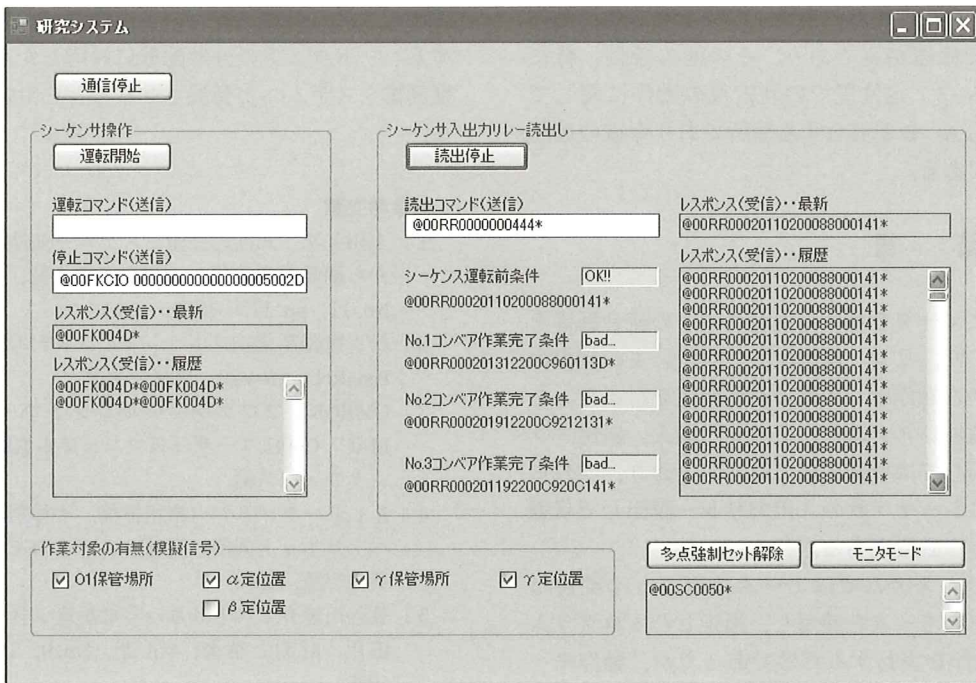


図28 Visual Basic による操作画面

を読み取ることができない。そこで、多点強制セット/リセットコマンドを使用してリレーのセットコードを送信後、セット解除コードを送信することで対応した。ここでも通信における送信権の移行時間を考慮してプログラム内にタイマを設け、コマンド送信による開始・停止指令の動作確認を行った。

条件判定後の動作指令については、判定のタイミングはコンベア定位置で作業完了した時点とし、条件次第で運転または停止の指令を送信するようにした。例えば、運転中に条件2が不成立であったときには、No.1コンベア定位置上で停止指令を受けて停止する。条件成立するまでそこで待機し、成立したときに運転指令を送信して運転再開する。Visual Basic のプログラムでは、ボタンを押すごとに運転開始/停止のコマンドを送信することで動作指令とした。プログラム例を図26, 27に示す。Visual Basic による操作画面例を図28に示す。

現在、運転開始時の条件1による判定後の動作は実機で確認済みであり、その他の条件、特にNo.1コンベア定位置での判定後の動作に関しては、部品A, Bが競合する箇所であり今後の確認が必要である。

5. 結 言

コンピュータを用いた生産ラインの統合制御を目的として、リレー状態の条件判定による制御のプログラムを作成し実機で動作確認を行った。ただし、追加システムとして考えた搬入、搬出コンベアに関しては模擬信号で対応しており、これらも同様にシーケンサの入出力リレー読出しで状態監視を行う必要がある。

今回のシステムではアームロボットの制御はシーケンスモードを使用し、指定したプログラムNo.の動作を実行するだけであったが、動作モードとしてコンソールモードを用いてダイレクトコマンドを送信することでさらにフレキシブルな制御が可能になる。コンソールモードのときのコン

ベアとの連動は、シーケンサの運転指令時に別途アームロボット用にプログラムNo.指定で運転指令を送信することで問題なく行えることは実機で確認済みである。

今回はパソコン1台での制御を行っているが、今後はリアルタイムマルチタスクOS（組み込みLinux, μ ITRON など）を使用し並行プロセス間での情報のやり取りを行うことで、ベトリネットによる進行状況表示やネットワークからの操作などを可能にし、さらに実用性を高めていくことや、自立移動台車や画像認識システムなど、様々な機器から構成される生産システムに対して適用可能な汎用的な制御ソフトウェアとすることが課題である。

以上を基本システムとして、さらに大規模かつ複雑な生産システムに適用可能とするため、生産ライン全体の統合制御、ベトリネットに基づくロボットなど各機械の制御、および各機械の専用コントローラによる制御から構成される階層分散型制御システムの構築を行い、生産システムを構成するハードウェアの分散配置に対応した自律分散型制御システムへと発展させることを計画している。

参考文献

- 1) 上田完次：知能化生産システム—知識型知能化から創発型知能化へ、精密工学会誌, Vol. 59, No. 11, pp. 1755-1760, 1993
- 2) 松下電器産業(株)ロボット・FAシステム事業部：PanaRobo KS-V20 取扱説明書
- 3) OMRON：プログラマブルコントローラSYS-MAC C60H ユーザーズマニュアル RS-232C インタフェース編
- 4) J. L.ピーターソン（市川惇信, 小林重信訳）：ベトリネット入門—情報システムのモデル化, 共立出版, 1984
- 5) 長谷川健介：ベトリネットの生産システムへの応用, 計測と制御, Vol. 28, No. 9, 183-190, 1988
- 6) 金藤仁：自動計測システムのためのVisual Basic 2005入門, 技術評論社, 2007