

光造形法の高分解能化と
そのマイクロマシン製作への応用に関する研究

高木 太郎


目次

学位論文

光造形法の高分解能化と
そのマイクロマシン製作への応用に
関する研究

1～212ページ 完

平成6年12月20日 提出

指導教官 中島尚正教授 

27053 高木太郎

目次

本論文の構成	4
第1章 はじめに	5
1-1. 本研究の目的	
1-2. 研究の背景	
1-3. 研究成果の概要	
第2章 光造形法を利用したマイクロマシン製作	10
2-1. 微細加工法の現状と問題点	
2-2. マイクロマシンと先導設計	
2-3. 微細加工法としての光造形法の特徴	
第3章 光造形法	23
3-1. 光造形法の現状	
3-2. 光硬化性樹脂	
3-3. 光造形法の課題	
第4章 加工分解能と固化セルの考え方	41
4-1. 固化セルと加工精度・加工分解能	
4-2. 固化セルの寸法・形状の予測	
4-3. 加工分解能を高めるための指針	
第5章 マイクロマシン製作用光造形システムの試作	51
5-1. 加工装置の設計	
5-2. 加工シーケンスの設計	
5-3. ソフトウェア系の設計	

第6章 加工分解能を高めるための指針の検証	74
6-1. 固化セルの観察	
6-2. 加工分解能の測定	
6-3. 可視光による露光	
6-4. 結果と考察	
第7章 可動部を持つ微細な機構の製作	97
7-1. 手動機構の作製	
7-2. 自動機構の作製	
7-3. アクチュエータの作製	
7-4. 結果と考察	
第8章 考察	120
8-1. 光造形法を利用したマイクロマシン生産についての考察	
8-2. 今後の課題とその解決方法についての考察	
第9章 まとめ	135
謝辞	137
付録	138
用語について	
光造形システム技術マニュアル	
製造元・販売元・工場一覧	
機材仕様・材料仕様一覧	
研究発表一覧	

本論文の構成

第1章では、マイクロマシン技術・光造形法の技術の両者について、それらの利用と研究の現状を概観しながら、光造形法を利用したマイクロマシン製作法の研究の目的と意義を解説する。また本研究の成果を簡単に紹介する。

第2章では、マイクロマシンの製作に用いられる微細加工法を紹介し、既存の微細加工法の問題点を解説する。またマイクロマシンの設計・製作に適した設計手法である「先導設計」の考え方にに基づき、光造形法を利用したマイクロマシン製作法の特徴と、最初に解決しなければならない課題を整理する。

第3章では、光造形法・光造形加工システム・光硬化性樹脂に関する一般的な知識を解説する。またRP/Mに用いられている光造形法の問題点を考えながら、微細加工法の一つとして、光造形法をマイクロマシン製作に応用する際に必要となる仕様を整理する。

第4章では、光造形法で作られた加工物の構成単位であると考えられる「固化セル」の概念を導入し、加工精度・加工分解能と固化セルの形状・寸法の関係を議論する。またシミュレーション・数式解析によって加工条件の違いによる固化セルの変化を予測し、加工分解能を高めるための指針を得る。

第5章では、第6章・第7章で述べる実験のために試作した、新しいマイクロマシン製作用の光造形システムについて、RP/M用光造形システムとの相違を整理しながら、その設計上の特徴を解説する。

第6章では、光造形法の加工分解能に関して行った、単セル露光実験・分解能評価実験・可視露光実験について、実験の方法と結果を報告する。

第7章では、材料の機械的特性に関して行った、機構作製実験・自動機構作製実験・アクチュエータ作製実験について、実験の方法と結果を報告する。

第8章では、これまでの理論検討・実験の結果を踏まえ、光造形法を利用したマイクロマシン製作技術を実用化するための工学的な指針をまとめる。また今後解決しなければならない課題を整理し、その解決方法を提案する。

第9章では、本研究の成果をまとめて示す。

第1章 はじめに

1-1. 本研究の目的

「光造形法」は、「光硬化性樹脂」と呼ばれる合成樹脂を利用した付加加工法の一種である。光硬化性樹脂は初めは粘潤な液状であるが、紫外線などの光が照射されると、光化学反応を起こして固体に変化する。そこで特別な光学系を備えた加工装置を使い、この樹脂を選択的に固化させていくことで、さまざまな形状の加工物を作り出すことができる。これが光造形法の基本概念であり、初の特許は1980年に日本国内で出願された⁽¹⁾。その後CAD設計が普及してくるにつれ、CADで設計された立体形状モデルから、短時間で実体模型（モックアップ）や型を作る手法、いわゆる「ラピッドプロトタイプング・マニユファクチャリング（RP/M）」を実現する手段として積極的に活用されるようになってきた。

一方「マイクロマシン」は、従来の加工技術では製作困難な、きわめて微小な機械、あるいは精密な機械の総称である。実用的なマイクロマシンを作る技術が確立すれば、医療・産業・民生などの応用分野で新たな可能性が拓かれることが期待されている。特に日本国内では平成3年度から通商産業省工業技術院の大型プロジェクトに指定されており、現在は官民の研究機関で基礎と応用の両面から活発な研究がなされている⁽²⁾。

本研究の目的は、光造形法を利用した新たなマイクロマシン製作技術を提案し、この技術を実用化するための工学的な指針を示すことである。マイクロマシンの研究においては、既存の微細加工法の「加工の自由度」の小ささが大きな問題となっている。そこで本研究では「加工の自由度」の大きい加工法として知られている光造形法をマイクロマシン製作に応用し、この問題を解決することを考えた。

1-2. 研究の背景

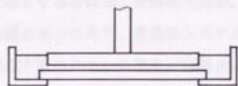
マイクロマシンを作るには、「微細加工法」と呼ばれる特殊な加工法が必要である。これまでに考案された微細加工法の多くは現在も発展途上にあり、ケイ素単結晶のエッチング異方性を利用した「シリコンプロセス」だけが商業ベースで利用されているにすぎない。シリコンプロセスはICの製造法として開発されてきた技術をマイクロマシン製作に転用したもので、加工精度・分解能や生産性の

点でかなり完成された技術となっている。この方法を使うと、機械構造と同時に駆動・制御用の電子回路を構成することができ、加速度センサ・圧力センサなど、構造の機械的特性を利用した電子部品を作ることができる。これらの電子・機械複合デバイスは欧米で「MEMS」と呼ばれており、微細加工技術の主要な応用分野となっている。

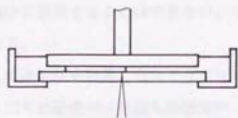
ところが日本の大型プロジェクトが想定している「マイクロマシン」の概念は、MEMSの概念と必ずしも同一ではない。MEMSが単なる「部品」であるのに対し、マイクロマシンはさらに独立性の高い「機械システム」であり、その応用分野もMEMSよりも多岐にわたっている。このためマイクロマシンを作るには、シリコンプロセスよりも自由な加工のできる製作技術が不可欠となる。シリコンプロセスは、シリコンウエハ表面に部分的に化学処理を施すことによって構造を作る「表面加工」の一種であり、製作可能な構造が平面的なものに限られるので、複雑な立体形状を持つ機械構造を製作することは一般に難しい。

立体的な機械構造を作るためにLIGAプロセス・マイグロ放電加工・レーザCVDなど、これまで多くの微細加工法が考案され、その応用が試みられてきた。しかしながら現状では、まだ目的に合致する実用的な機械システムを自由に実現できるようになっていない。また微細な機構部品を作る方法は多くの研究者によって研究されているが、微細な部品を機械システムとして組み立てることは非常に困難で、その方法について研究された例も非常に少ない。そのため組み立て工程を省き、複雑な機械製品や機械モジュールを一度に作ること（ブリアセンブリ）のできる、新たな微細加工技術の確立が望まれている。

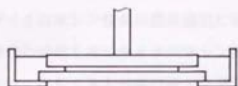
これに対し、光造形法は非常に複雑な形状を持つ加工物を作ることができる。RP/M用光造形システムの代表的な方式である「ステレオリソグラフィ法」の加工シーケンスを図1-1.1に示す。通常的光造形システムでは、多数の薄い層を重ねていくことで加工物を作る、「積層造形法」と呼ばれる加工原理が採用されている。積層造形法は一種の付加工法であり、除去加工法に比べて一般に加工物の形状や構造の制約が少なく、いわゆる「加工の自由度」が大きいという特徴を持つ。これがRP/Mに光造形法が多用されてきた理由の一つであるが、マイクロマシン製作におけるブリアセンブリを実現する上でも、この点は非常に有利である。



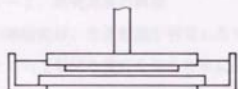
1. Position the elevator near the window.



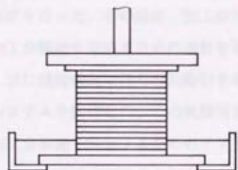
2. Scan the beam along the first layer.



3. Finish the first layer.



4. Raise the elevator for thickness of one layer.



5. Repeat these operations to form the object shape.

図 1-1.1 光造形法の加工シーケンス (ステレオリソグラフィ法)

光造形法を利用したマイクロマシン製作法を実用化することを考えると、まず問題となるのは加工分解能である。加工分解能の改善はRP/Mでも特に重要な課題となっており、光造形システムの各メーカーでも積極的な研究を行っているが、現在市販されているRP/M用光造形システムでは約 $40\mu\text{m}$ の加工分解能しか得られていない⁽¹⁾。これに対してマイクロマシンの製作には $10\mu\text{m}$ 以下の加工分解能が必要とされており⁽¹⁾、市販の光造形システムをそのままマイクロマシン製作に流用することはできない。したがって何らかの指針に基づく改善が必要となる。

光造形法を利用したマイクロマシン製作法の研究例としては、本研究のほかに山口らの研究⁽²⁾・生田らの研究⁽³⁾・鈴木らの研究⁽⁴⁾・単らの研究⁽⁵⁾などがあり、さらに日本国内の数社で研究がなされている。これらの研究に対し、本研究ではマイクロマシン製品の開発過程における微細加工法の位置付けを考慮しており、特定の機能を持つマイクロマシン製品だけでなく、各種のマイクロマシン製品の製作に応用できる基礎技術を開発することを目指していることが特徴である。

1-3. 研究成果の概要

本研究は、光造形法を利用したマイクロマシン製作法の実用化を目指し、そのための工学的な指針を得るために、以下のようなアプローチで進められた。

まず光造形法の加工分解能を高めることを考えて、樹脂が固化する際に起こる物理現象を考慮し、「固化セル」と呼ぶ概念を導入してシミュレーション・数式解析を行った。その結果、加工条件と加工分解能の関係を把握することができ、加工分解能を高めるための指針を導くことができた。

次に理論検討で得られた指針を踏まえ、マイクロマシン製作用の実験用光造形システムを試作した。この実験用光造形システムを用い、加工条件を変えながら加工分解能を評価するためのテストモデルを作製した。その結果、加工分解能を約 $5\mu\text{m}$ にまで高めることができ、これらの指針の妥当性を示すことができた。

さらに実験用光造形システムを用い、各種の動作原理の機構をテストモデルとして作製した。また微細な機構を駆動する各種の方法を提案し、テストモデルに応用した。その結果、設計したとおりにモデルが動くことを確認することができ、可動部を持つ機械構造を作る方法としての光造形法の有用性を示すことができた。

最後に光造形法を利用したマイクロマシン製作法の実用化に向け、そのための工学的な指針を示した。また本研究の遂行によって新たに見つけ出された課題を整理し、今後の解決方法を提案した。

文献

- (1) 小玉秀男, 特開 昭56-144478 (1980).
- (2) 通商産業省工業技術院研究開発官室, 大型プロジェクト 1991年, 日本産業技術振興協会パンフレット (1991).
- (3) Watanabe, T., Okawa, M., Ukachi, T., Kurihara, F. and Harimaya, H., Development of High Precision Solid Creation System, Conference Proceedings of RadTech Asia '93 (1993), pp. 462-467.
- (4) 機械システム振興協会, マイクロロボット開発に関する調査研究報告書 (1991).
- (5) 山口勝美, 中本剛, Abbay, P. A., 壬生周, 紫外線感光性樹脂を使ったマイクロストラクチャの製造 (第1報), 機論 Vol. 57 No. 542 C (1991), pp. 288-294.
- (6) 生田幸士, 広渡憲, 光創製3次元マイクロファブリケーション, 機講論 No. 920-33 B (1992), pp. 545-546.
- (7) 鈴森康一ら, 光造形法によるフレキシブルマイクロアクチュエータの試作, 第11回日本ロボット学会講演会論文集 (1993), p. 569.
- (8) 単学伝, 横浜正毅, 柳沢一向, 光造形法を応用したセラミック/金属微小構造体の製法 (第1報), 1994年度精密工学会春期講演会論文集 (1994), pp. 579-580.

第2章 光造形法を利用したマイクロマシン製作

2-1. 微細加工

「マイクロマシン (micromachine)」は、「微細加工法 (micromachining process)」と呼ばれる加工法で作られる、きわめて微小な機械、あるいは精密な機械の総称である。「微細加工法」という言葉は、「従来の機械加工法よりも高い加工分解能を持つ加工法」を指し、切削や研削といった「従来の機械加工法」に相対する意味を持つものとして使われている。その加工分解能は一般に $10\ \mu\text{m}$ 以下である。マイクロマシンの定義方法については、動作原理による定義や構成要素の代表寸法による定義など多くの考え方があり、いまだに統一されていない。しかし近年では、「新しい加工法で作られる微細な機械システム」とする定義が最も一般的な定義として定着しつつある。

微細加工法を実現するための技術は「微細加工技術 (microtechnology)」と呼ばれている。当初の微細加工技術は、MEMS (micro electro mechanical systems) と呼ばれる、構造の機械的特性を利用した電子・機械複合デバイスを作るための技術であった。ICチップ上に電子回路とともに機械的に動く部分を構成することができれば、加速度センサ・圧力センサなどの機能デバイスを実現することができる。そこでICの製作技術の延長として、機械構造をチップ上に作る微細加工法の研究が始められた。

ICなどの半導体電子部品の製作に用いられるケイ素単結晶の表面加工技術は、総称して「シリコンプロセス (silicon processes)」と呼ばれている。「シリコンプロセス」という言葉が指しているのは単一の加工法ではなく、コーティング・真空蒸着・スパッタリング・CVDなどの薄膜形成法、溶剤洗浄・エッチング・アッシングなどの材料除去法、フォトリソグラフィ・イオン注入・熱酸化などの材料改質法、すべてを含む加工技術体系の全体である。すなわち複数の加工法を組み合わせることで、初めて複雑な構造を持つ電子回路をケイ素基板 (ウエハ) 表面に構成することが可能となるのである。

このシリコンプロセスを使い、1980年ごろからカンチレバーの弾性変形を利用した加速度センサや、ダイヤフラムの弾性変形を利用した圧力センサなどの力学的センサが作られるようになった。さらに1987年ごろには、静電気力を

利用したマイクロアクチュエータやマイクロモータが作られた⁽¹⁾⁽²⁾。またリンク機構やスライダ機構など、各種の機構が試作されるようになった⁽³⁾。

一方日本国内では、1991年度からの大型プロジェクトに「マイクロマシン技術の研究開発」が選定された。これを受けて、1992年に財団法人マイクロマシンセンターが設立され、大型プロジェクトの受託機関として関連事業を行うようになった。大型プロジェクトでは、2-2節で詳述する「ニーズ先行型」のアプローチが採用されている。すなわちマイクロマシンの用途・需要（ニーズ：needs）を医療・産業・民生などの各方面で調査し、具体的な実現目標としてのマイクロマシンシステムを実体化して、その仕様を抽出した後、要求仕様を実現することのできる技術の確立を目指したのである。

大型プロジェクトが開始されるとともに、国内では実用的なマイクロマシンを実現するための基礎技術（シーズ：seeds）がきわめて活発に研究されるようになった。また日本からの国際発表件数も増加し、独立した機械システムとしてのマイクロマシンの概念は国際的に認識されるようになってきた。

マイクロマシンの応用分野の拡大に伴い、単なる部品としてのMEMSよりも複雑な機械構造を作ることのできる、「加工の自由度」の大きい微細加工技術が求められるようになってきた。微細加工技術が発展すると、これをシーズとするマイクロマシンの応用分野が新たに提案される。しかしその後、その応用研究が進んでいくと、微細加工技術に対してさらに厳しい仕様が要求されるようになる。マイクロマシンの応用分野が広く考えられるようになった現在では、さまざまなマイクロマシンを実現するための要求仕様として、大きい加工の自由度が必要となっている。

加工の自由度の問題は、実はMEMSの開発においても以前から重要な問題となっていた。そこでシリコンプロセスの加工の自由度の小ささを補い、シリコンプロセスよりも自由に立体的な構造を製作するための方法として、「LIGA」・「マイクロ放電加工」・「レーザーCVD」などの微細加工法が考え出された。

LIGA (lithographic, galvanofornung, abfornung) は、リソグラフィと電鍍を組み合わせた微細加工技術で、1982年に発表された⁽⁴⁾。この方法ではまず基板の上に塗られたPMMA (polymethylmetacrylate) 樹脂レジストの厚膜に強いX線で露光した後、現像してレジストを除去する。こうして部分的に

基板を露出させてから、電鍍を行ってニッケルなどの金属で構造を作る。最後に金属構造をレジストのメス型から引き抜いて製品とする。この方法ではメス型が光の直進性を利用して作られるので、非常にアスペクト比の大きい柱状の構造を作製することができる。

マイクロ放電加工は従来の放電加工と原理的には同じだが、加工精度を高めるための改善を加え、微細加工に応用されたのは1985年である⁽³⁾。加工電極の摩耗の影響を取り除くため、電極をワイヤとして連続的に供給しながら加工する方法は「ワイヤ放電研削加工 (WEDG: wire electrodischarge grinding)」として知られている。

レーザーCVD (laser assisted chemical vapor deposition) は、気体の熱分解反応を利用した加工法で、1990年に発表された⁽⁴⁾。この方法では熱分解性の反応気体中に置かれた加工物にレーザー光を照射し、加工物の一部を選択的に加熱する。こうして反応気体を分解させ、加工物表面に固体材料を析出させることで目的の形状を作っていく。この方法は光造形法と同じく付加加工であり、大きい加工の自由度を持つ。

ここではいくつかの微細加工法を紹介したが、これらの他にも多くの加工法が提案されており、またそれぞれの加工技術を改良するための研究もなされている。しかしこれらの努力にもかかわらず、大型プロジェクトが想定するような広義のマイクロマシンを自由に製作することのできる、決定的な加工法はいまだに開発されていない。現在マイクロマシンの研究分野では、加工の自由度の大きい微細加工技術を実用化することが急務となっている。

2-2. マイクロマシンと先導設計

マイクロマシン技術を実用的な技術として利用できるようにするには、目的の機能を持つマイクロマシンを自由に設計し、製作することを可能にする、新しい方法論を確立しなければならない。そこでこの方法論についての議論を行う前に、通常の寸法の機械製品とマイクロマシン製品の開発過程を比較しておく。

まず通常の寸法の機械について、その開発過程を分析する。多くの場合に設計段階と製作段階とは明確に分離され、それぞれの段階を専任の設計者・製作者が担当している。両者の間の情報伝達は図面や指示書によってなされているため、

設計者は図面や指示書の作成に専念でき、製作者は図面や指示書を読んで製品を製作することに専念できる。設計者は製作段階の詳細をすべて知っている必要はなく、標準化されている図面や指示書の書き方を知り、製作不可能な構造を判別できればよい。訓練を積んだ設計者によって作成された図面や指示書に示されている対象物は、熟練した製作者が現在の機械加工技術を使用して製作することができる。つまり通常の寸法の機械製品の開発では「設計者の設計した機械構造を、製作者は必ず製作することができる」ということが保証されており、このことが設計者と製作者の分業を成立させるための前提になっている。

ところがマイクロマシン製品の開発においては、以上のような前提は必ずしも成立しない。図2-2.1に機械製品の一般的な開発過程、すなわち要求仕様の設定から製品の製造までの開発段階の流れを示す。通常の寸法の機械の場合には、それぞれの開発段階はすでに十分に確立されており、全体では技術的に連続した手順の流れを構成している。すなわちこの流れに沿って製品開発を進めることで、目的の機能を持つ機械製品を比較的容易に開発することができる。これに対してマイクロマシンの場合には、定式化された開発過程が確立されていない。

図2-2.1をマイクロマシン製品について詳しく書き直すと、図2-2.2のようになる。それぞれの開発段階が独立しておらず、後段の開発段階に依存していることが、現状でのマイクロマシン製品の開発過程の特徴となっている。この依存関係を矢印で図2-2.2に示した。

図2-2.1に示したように、機械製品の開発過程には「設計のらせん (helix of design)」と呼ばれる繰り返し部分が存在する。すなわち製品開発を短時間で行うには、この繰り返し部分に要する時間を短縮することが重要になる。しかしマイクロマシンの場合には、繰り返し部分の内部にある動作原理から機械構造を決定する過程が、その後段にある加工段階に大きく依存しているので、きわめて非能率的なものとなっている。このことが、マイクロマシン製品の設計には特に熟練を要し、要求仕様を満足するマイクロマシン製品を短時間で開発することが難しい理由となっている。

MEMSの研究開発においては、まず基礎研究によってシーズを蓄積していき、それらのシーズから新たなニーズが生み出されることを狙う、いわゆる「シーズ先行型」のアプローチが採られてきた。しかしこのやり方では、図2-2.2に

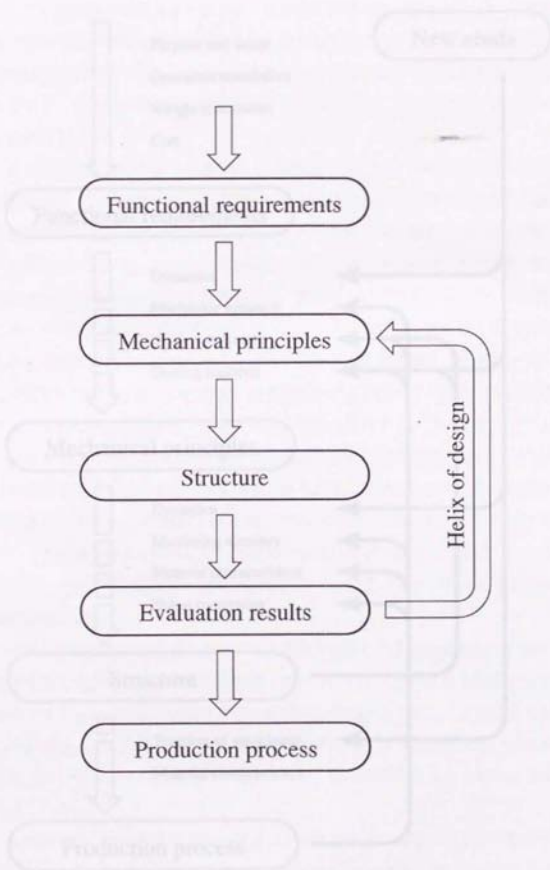


図 2-2.1 機械製品の開発過程

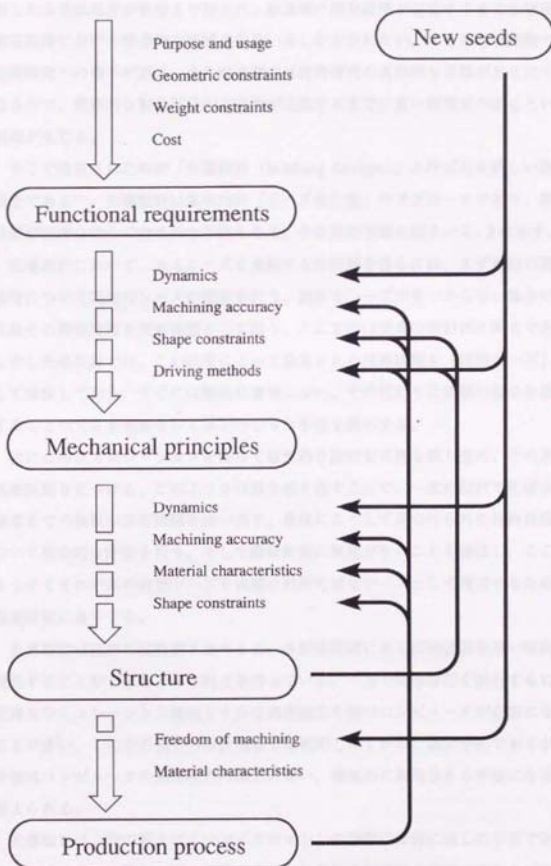


図 2-2.2 マイクロマシン製品の開発過程

示した未完成部分が最後まで残され、製造側の開発段階が完成するまで仕様側の開発段階における製造側の問題点の洗い出しがなされない。すなわち仕様側では技術研究への着手が遅れ、また製造側では技術研究の具体的な目標が立てにくくなるので、最終的な製品開発の方法論が完成するまでに長い期間がかかるという問題が生じる。

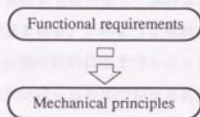
そこで提案されたのが「先導設計 (leading design)」と呼ばれる新しい設計概念である¹⁾。先導設計は基本的に「ニーズ先行型」のアプローチであり、開発目標が明確な場合に効果的な手法となる。その実行手順を図2-2.3に示す。

先導設計において、あるニーズを実現する設計解を得るには、まず最初の開発段階について既知のシーズの探索を行う。適当なシーズが見つからない場合には、以後その開発段階を技術課題として扱う。ここまでは従来の設計法と同じである。しかし先導設計では、この探索によって発見された技術課題を「仮想シーズ」として保留しておき、すぐには解決に着手しない。その代わりに課題の部分置換することのできる適切なシミュレーション手法を提示する。

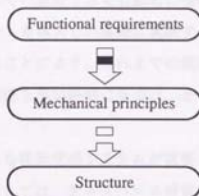
次にこのシミュレーションを使って仮想的な設計を可能な限り進め、その次の技術課題を見つける。このような作業を繰り返すことで、一度の設計で仕様から製造までの複数の技術課題を洗い出す。最後にこうして見つけられた技術課題について総合的な評価を行う。そして開発計画に無理がないことを確認し、ここできちんとそれぞれの仮想シーズを実際に利用可能なシーズとして確立するための技術研究に着手する。

先導設計は従来の設計法と比べると、各開発段階にある技術課題を早い時期に発見することができるという利点を持っている。一方で先導設計を実行するには、正確なシミュレーション技法と十分な処理能力を持つコンピュータが必要になることが多い。このため過去の設計環境では利用しにくかった設計手法であるが、今後はコンピュータの処理能力の向上に伴い、積極的に利用される手法になると考えられる。

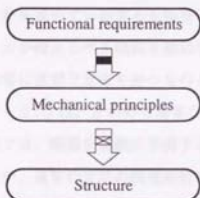
先導設計は、先の見えにくいマイクロマシンの設計には特に適した手法である。マイクロマシンの設計では、寸法によって対象物の物理的な挙動が変わる「寸法効果 (scale effect)」のため、従来からの直観的な設計知識の適用は設計ミスの原因となることがある。そこで人工知能を利用した提案的なシミュレータを設計



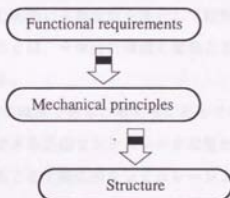
1. Start design and find difficulty.



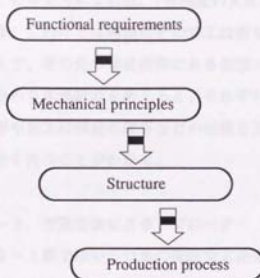
2. Replace difficulty with simulation.



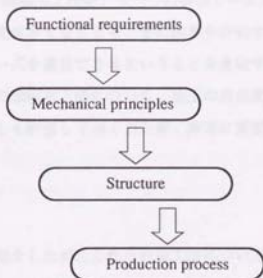
3. Step to next and find impossibility.



4. Go as traditional design method.



5. Arrive at the design goal.



6. Solve all sub-problems.

図 2-2.3 先導設計の実行手順

作業と並行して走らせて、設計者が発見しにくい設計ミスを自動的に指摘させることが望まれる。このような設計支援システムを導入し、実際に実験することのできない先の設計段階までシミュレートすることにより、これまでの設計法では実験結果が得られるまで発見されなかった問題を早い時期に見つけ、それに対処することができる。

理想的な先導設計の実現には、以上のような設計支援システムが重要な役割を果たす。このようなシステムに関する研究としては、データベース検索によって物理現象を連鎖的に追跡するシステムを試作した例がある¹⁴⁾。実用的な設計支援システムが完成するまでの当面は、手作業によってシミュレータを起動しながら先導設計を進めていかざるを得ない。しかし実際に先導設計を実行して設計支援システムが備えるべき機能を抽出していくことは、今後設計環境を整備していく上で非常に重要な意味を持つものと思われる。

図2-2.2は、きわめて簡単な先導設計の結果であると考えられることもできる。この例では、問題を詳細に予測することのできる正確なシミュレータは使われていないが、通常の寸法の機械設計と対比することで疑似的なシミュレーションが行われたものと見なすことができる。つまり図2-2.2に示した未完成部分は、この簡単な先導設計によって発見された仮想サイズである。

この考え方によれば、「自由度の大きい微細加工技術」も一つの仮想サイズである。このことは実際にその加工技術を完成させなくとも、その限界を評価することで、その先の開発段階にある仮想サイズを発見できるということの意味する。すなわち先導設計を考えると、それぞれの微細加工法について、加工の自由度の限界や加工分解能の限界などの仕様を正しく評価しておくことが、非常に重要な意味を持つことがわかる。

2-3. 光造形法によるアプローチ

2-1節ではいくつかの微細加工法を紹介したが、これらの加工法についてはいずれも加工の自由度が問題となっており、加工物形状の制約を緩和することが大きな課題となっている。

ところで通常の寸法の機械製品を開発する場合、加工の自由度の問題はあまり深刻な問題にならない。これは従来の加工法が、既存の微細加工法よりも大きい

加工の自由度を持っているためではなく、一つの加工物を製作するために複数の加工法を組み合わせてことができ、それぞれの加工法の持っている難点を互いに補うことができるためである。

しかしマイクロマシン製品を開発する場合には、加工の自由度が深刻な問題となる。マイクロマシンは微細なので、複数の加工法を組み合わせて一つ一つの加工物を作ることが難しく、一般にはただ一つの加工法に依存せざるを得ない。すなわちそれぞれの加工法の持つ特徴は、そのまま設計段階に反映されることになる。現在多くの微細加工法が十分に大きい加工の自由度を持っていないという事実は、マイクロマシンの設計の自由度を失わせ、各応用分野において決定的な障害をもたらしている。

そこで本研究では、立体的な加工物を自由に製作することのできる光造形法に着目し、この加工法をマイクロマシン製法として利用することを考えた⁹⁾。

光造形法を利用したマイクロマシン製法には、次のような利点がある。

1. 設計してから実物を得るまでの期間が短い。CADで設計したマイクロマシンは、光造形法を使うと数時間で完成品とすることができる。他の加工法では、実物を得るまで数ヶ月の期間を要することも多い。
2. 機械構造を自由に設計できる。光造形法は加工の自由度が大きく、他の加工法では作れない複雑な構造を作ることができる。そのため設計時に構造上の制限を考慮する必要がなくなり、能率的な設計作業が行える。
3. 加工がすべて自動的に行われる。人手が必要なのは、基板の据え付けと完成品の取り外し、および洗浄だけである。しかも作業手順は加工物の形状によらないので、不慣れによる人的トラブルが避けられる。
4. 設備が経済的である。基本的な機能を持つ光造形システムは500万円以下で構成することができる。この金額はシリコンプロセスやLIGAプロセスの設備費の1割に満たない。
5. 大がかりな空気浄化設備が不要である。光造形法では、加工物が粘濁な樹脂の内部で作られ、空気中の塵埃は液面にトラップされて加工物まで届かない。そのためクリーンルームなどの設備はきわめて簡便ですむ。

光造形法を利用したマイクロマシン製法の特徴を、既存の微細加工法と比較したものを表2-3.1に示す。

表2-3.1 各種の微細加工法の比較

	加工分解能	加工物形状	加工材料	問題点など
シリコンプロセス	0.2 μm	主として 平面的な形状	半導体・金属・ 酸化物など	複数の加工装置を用いるので、空気 浄化設備が必要である
LIGA	1 μm	主として 柱状の形状	金属	加工装置がきわめて高価であり、製品 製造コストも高い
マイクロ放電加工	50 μm	外部から加工 できる形状	金属	複雑な加工物を作る場合には、専用の 加工装置が必要である
レーザCVD	20 μm	ほぼ任意の 形状	ケイ素・ホウ素・ 金属など	熱を利用しているので、現在よりも 加工分解能を高めることは難しい
光造形法	2 μm	ほぼ任意の 形状	樹脂	

以上のように、光造形法を利用したマイクロマシン製作法には多くの画期的な利点が期待できる。しかしこの方法を実用化するには、解決しなければならない課題もいくつか残されている。

特に重要な課題として、加工分解能を高めることが挙げられる。加工分解能の改善はR P/Mにおいてもすでに大きな課題となっており、解決方法が積極的に議論されている。しかしマイクロマシンを製作する微細加工法には、R P/Mにおける目標値の10分の1以下に相当する、約1 μ mの加工分解能が求められる。このような高い加工分解能を実現するには、R P/Mの場合とは異なる何らかの技術的な指針を導くことが必要となる。

さらに別の課題として、樹脂材料の機械的特性や機械構造の設計指針に関する知識を蓄えることが挙げられる。光造形法によって実際に力や動きを伝える機械構造を製作した事例はわずかなので、実際に試作品を作り、それを評価しながらデータを蓄積していく必要があると考えられる。

文献

- (1) Tang, W. C., Nguyen, T. H. and Howe, R. T., Laterally Driven Polysilicon Resonant Microstructures, Conference Proceedings of MEMS '89 (1989), pp. 53-59.
- (2) Tai, Y., Fan, L. and Muller, R. S., IC-processed Micro-motors: Design, Technology, and Testing, Conference Proceedings of MEMS '89 (1989), pp. 1-6.
- (3) Muller, R. S., From ICs to Microstructures: Materials and Technologies, Conference Proceedings of MEMS '87 (1987).
- (4) Ehrfeld, W., Bley, P., Götz, F., Hagmann, P., Maner, A., Mohr, J., Moser, H. O., Münchmeyer, D., Schelb, W., Schmidt, D. and Becker, E. W., Fabrication of Microstructures Using the LIGA Process, Conference Proceedings of MEMS '87 (1987).
- (5) Masaki, T., Kawata, K. and Masuzawa, T., Micro Electro-discharge Machining and Its Applications, Conference Proceedings of MEMS '90 (1990), pp 21-26.

- (6) Boman, M., Westberg, H., Johansson, S. and Schweiz, J., Helical Microstructures Grown by Laser Assisted Chemical Vapour Deposition, Conference Proceedings of MEMS '92 (1992), pp. 162-167.
- (7) Nakajima, N., Micromachines as Intelligent Artifacts, Conference Proceedings of 1st International Symposium on Research into Artifacts (1993), pp. 48-51.
- (8) Kiriya, T., A Design Environment for Micromechanics, Conference Proceedings of 1st International Symposium on Research into Artifacts (1993), pp. 69-72.
- (9) 高木太郎, 中島尚正, 光造形法による微細加工の可能性, 機論 Vol. 59 No. 560 C (1993), pp. 287-292.

第3章 光造形法

3-1. 光造形法の現状

「光造形法 (photopolymer fabrication, photocuring)」は、光硬化性樹脂 (light curable resin) の光硬化性を利用した立体造形法の総称である。光硬化性樹脂は合成樹脂の一種で、一般に粘潤な液状物として市販されているものである。この樹脂は、表面にある種の光を照射すると、樹脂内部で光化学反応を起こして固形の樹脂に変化 (固化: solidification) する。この性質を利用し、光硬化性樹脂は以前から塗料やコーティング材料として用いられていたが、光造形法ではこの樹脂を選択的な露光によって部分的に固化させることで、基板や基材の上に目的の立体形状を持つ加工物を作り出していく。

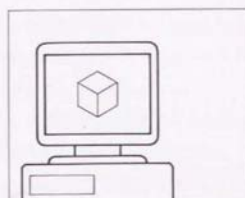
この技術は1981年に、三次元情報の表示方法として考案された⁽¹⁾。その後CAD設計の普及に伴い、ラビッドプロトタイプング・マニファクチャリング (RP/M: rapid prototyping / manufacturing) の重要性が認識されてくると、立体形状モデルを短時間で実体模型 (モックアップ: mock-up) 化する「立体プロッタ」を実現する手段の一つとして位置付けられた。現在日本国内・国外の数社のメーカーが光造形システムを提供しているが、これらはすべてRP/M用のものである。市販されているRP/M用光造形システムの一覧を表3-1.1に示す。またこれらの光造形システムを用いて実体模型を製作する一般的な手順を図3-1.1に示す。

加工技術としての光造形法のとらえられ方は、最近まで日本と欧米で明らかに異なっていた。日本では一般的な加工法の一つとして理解されていたのに対し、欧米では単なるRP/Mの実現手段として理解されていたにすぎなかった。このことが、日本国内ではすでに普通に使われていた「光造形法」に該当する言葉が欧米に存在せず、今日までの長い間にわたって商品名で呼ばれていたこと、また光造形法をRP/M以外の分野に応用した例が欧米では非常に少なかったことの原因となっている。

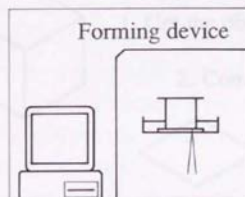
RP/M用光造形システムの多くは「積層造形法」と呼ばれる加工原理を応用している。積層造形法の一般的な加工手順を図3-1.2に示す。この方法では、まず対象モデルを多数の水平面で切った時に現れる断面形状を順次求め、一連の

表3-1.1 市販のRP/M用光造形システム

システム名 (製造元・販売元)	加工方式	走査分解能	加工分解能
SLA (3D Systems)	ステレオリソグラフィ法 自由液面式・走査露光式	水平: 5 μm 垂直: 100 μm	—
SOUP (シーメット)	ステレオリソグラフィ法 自由液面式・走査露光式	水平: 50 μm 垂直: 50 μm	—
SCS (ディーメック)	ステレオリソグラフィ法 自由液面式・走査露光式	水平: 40 μm 垂直: 10 μm	水平: 40 μm 垂直: 400 μm
Solider (Cubital)	固体下地法 一括露光式	水平: 100 μm 垂直: 100 μm	—
COLAMM (三井造船)	ステレオリソグラフィ法 規制液面式・走査露光式	水平: 40 μm 垂直: 10 μm	—
Stereos (EOS)	ステレオリソグラフィ法 自由液面式・走査露光式	水平: 20 μm 垂直: 50 μm	—
SLP (デンケンエンジニアリング)	ステレオリソグラフィ法 規制液面式・走査露光式	水平: 40 μm 垂直: 40 μm	—
Soliform (帝人製機)	ステレオリソグラフィ法 自由液面式・走査露光式	水平: 40 μm 垂直: 40 μm	—

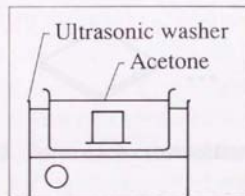


1. Modeling



Forming device

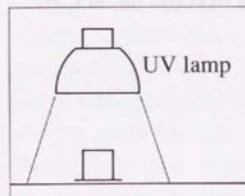
2. Forming



Ultrasonic washer

Acetone

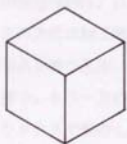
3. Washing



UV lamp

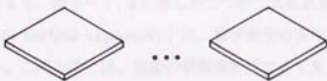
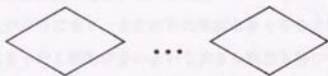
4. Postcure

図3-1.1 光造形システムを用いて実体模型を作る手順



1. Get the object solid model.

2. Compute to make slicing data.



3. Form each crosssectional layer.

4. Pile up layers to build the work.

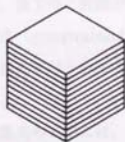


図3-1.2 積層造形法の加工手順

断面形状データ（スライスデータ）を作る。次にそれぞれの断面形状に合わせて薄い層または板を加工する。最後に各層を順に重ね、目的の加工物を得る。この方法の最大の特徴は、スライスデータさえ得られれば、ほぼあらゆる立体形状を実体化できることである。

こうした積層造形法を実現するため、図3-1.3に示した二つの加工方式の光造形システムが実用化されている。一方のステレオリソグラフィ法（stereolithography）は、樹脂槽に満たした液状樹脂の中で加工物を作る方法である。この方法は加工装置の構成や保守が簡単であり、また材料の無駄も少ないという利点を持つ反面、加工物のオーバハング部を加工することが難しいという欠点も持つ。もう一方の固体下地法（SGC法：solid ground curing）は、加工物をワックスで保持しながら作る方法である。この方法はステレオリソグラフィ法と比較すると加工装置が大がかりになり、また材料の無駄も多くなるという欠点を持つが、加工物形状にはまったく制約がないという大きな利点も持つ⁽¹⁾。

ステレオリソグラフィ法の加工装置は、露光がなされる樹脂表面の位置を制御する方法により、図3-1.4に示した二つの方式に区別される。一方の「自由液面式（free surface exposure）」は、液状樹脂の自由液面に対して露光を行う方法である。この方法では、後者の規制液面式でよく見られる、加工中に樹脂が窓板に固着するトラブルが起こらない。もう一方の「規制液面式（constrained surface exposure）」は、「窓板」と呼ばれる透明な板を通して露光を行う方法である。この方法では垂直方向の加工精度が容易に保証される⁽²⁾。

加工物の断面形状に合わせて各層を加工するために行う描画露光の方法には、図3-1.5に示した二つの方式がある。一方の「一括露光式（マスク露光式：photomask patterning）」は、光学マスクを使って断面を一度に露光する方法である。この方法では描画工程と露光工程が独立しているため、両方の工程を並行して行うことで加工時間を短縮できる。もう一方の「逐次露光式（走査露光式：scan patterning）」は、露光ビームを走査して断面形状を描きながら露光を行う方法である。この方法は加工装置の構成や保守が簡単で、実際の設計現場で利用しやすい⁽³⁾。

これらの加工装置は、一般に「STL」と呼ばれるインターフェースを備えている。対象モデルはあらかじめ「STL形式」と呼ばれるデータ形式に変換され、

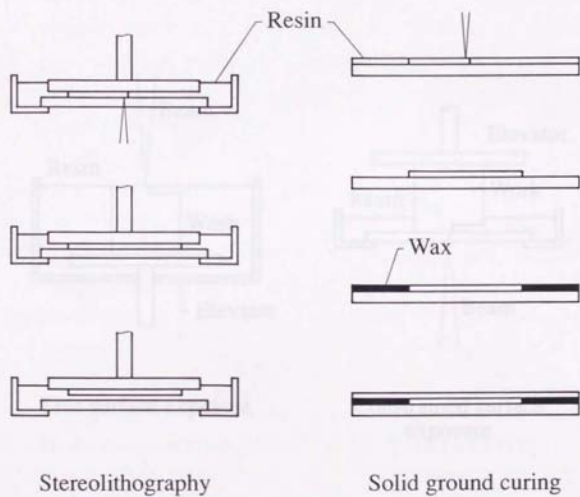


図3-1.3 ステレオリソグラフィ法と固体下地法

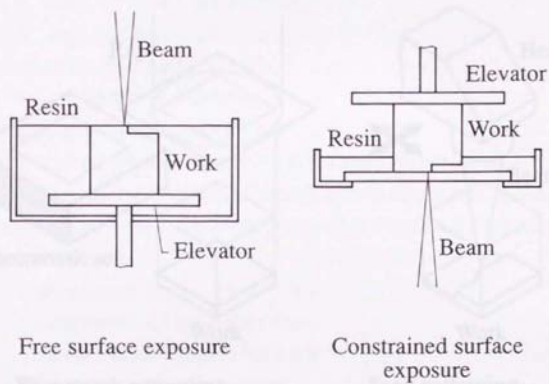


图 3-1.4 自由液面式と規制液面式

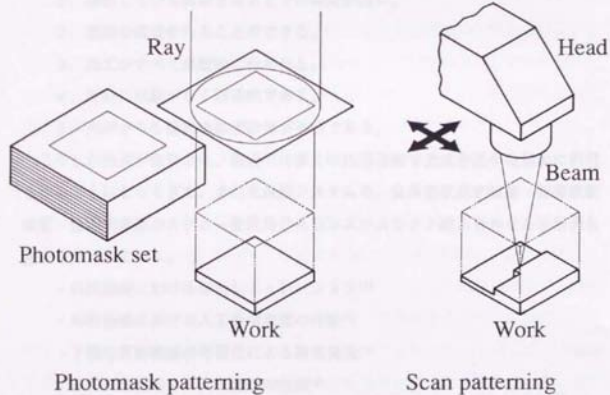


図3-1.5 一括露光式と逐次露光式

加工装置に渡される。STLではモデル表面を三角形の平面パッチ群で置換し、モデル全体を近似的に多面体として扱う。したがって加工装置を制御するための計算処理は簡単であるが、特に高い加工精度が必要な場合には、近似による寸法誤差が問題となることもある⁽⁴⁾。

加工法としての光造形法は多くの特徴を持っている。2-3節では光造形法をマイクロマシン製作に応用した場合の利点を述べたが、一般的に光造形法は次のような利点を持つ加工法である。

1. 設計してから実物を得るまでの時間が短い。
2. 複雑な構造を作ることができる。
3. 加工がすべて自動的に行われる。
4. 材料に無駄がなく経済的である。
5. 大がかりな廃棄物処理設備が不要である。

こうした利点が注目され、最近では多くの応用分野で光造形法が積極的に利用されるようになってきた。また光造形システムを、立体形状測定装置・断層撮影装置・仮想現実感システム・意匠用CADシステムなどと組み合わせた応用例も発表されてきている。

- ・外科治療における術前シミュレーション⁽⁵⁾
- ・外科治療における人工骨用原型の作製⁽⁶⁾
- ・下顎の運動軌跡の可視化による異常発見⁽⁷⁾
- ・CADを利用した人工歯冠の作製⁽⁸⁾
- ・成型用金型や鋳造用鋳型の作製⁽⁹⁾
- ・CADを利用した宝飾工芸品の作製¹⁰⁾
- ・立体地図を利用した大型建造物の建設計画
- ・立体芸術品や街路標識などの制作^{11) 12)}

このように新たな応用分野が開拓されるにつれ、光造形法に求められる仕様はますます厳しいものになってきた。光造形法は、加工装置を開発する機械技術・樹脂材料を開発する化学合成技術・モデル設計環境や加工シーケンスを開発するソフトウェア技術の複合技術であり、各技術分野ごとに、こうした新しい要求に対する取り組みがなされているのが現状である。

3-2. 光硬化性樹脂

光硬化性樹脂は合成樹脂の一種で、通常は粘潤な液状物として市販されており、光造形法では加工物を作る材料として使用される。RP/M用光造形システムで使われる光硬化性樹脂は、一般に次のような成分構成となっている⁹⁹⁾。

光重合性オリゴマ (光重合性プレポリマ)	40～80重量%
反応性希釈剤 (光重合性モノマ)	10～50重量%
光重合開始剤・増感剤・その他	～10重量%

光重合性オリゴマ (光重合性プレポリマ: photoreactive oligomer / photoreactive prepolymer) は、分子量が数千の比較的重合度の低い反応性の高分子化合物である。光固化反応が起こると、これらのオリゴマ分子は化学的に結合し、立体的な網目構造を作って樹脂固形物の主成分となる。したがって樹脂固形物の物理的性質は、この光重合性オリゴマによってほぼ決められる。

反応性希釈剤 (reactive diluent) は、樹脂の粘性を下げるために加えられる反応性の低分子化合物である。多くの場合、光重合性モノマ (photoreactive monomer) や、きわめて重合度の低い光重合性オリゴマが用いられる。光固化反応が起こると、これらの分子も化学的に結合して樹脂固形物の一部となる。

光重合開始剤 (photoinitiator) は、光を吸収して光化学的な連鎖反応を開始させる低分子化合物である。光重合開始剤分子に光子が当たると、分子は光子の持っていたエネルギーを吸収して活性化し、遊離ラジカル (free radical) や遊離カチオン (遊離陽イオン: free cation) を生成して化学反応を引き起こす。またこの時に増感剤 (sensitizer) が存在すると、ラジカル生成反応やカチオン生成反応の触媒として作用し、これらの反応の量子的な効率が高められる。

光造形法に使われる樹脂材料には、通常次のような条件が求められる⁹⁹⁾。

1. 光感度が十分に大きいこと
2. 固化時の体積収縮が小さいこと
3. 液状物の粘度が低いこと
4. 引火性・毒性が低く、安全であること
5. 固形物の機械的特性が優れていること
6. 液状物・固形物とも保存性が良いこと

このような条件を満たし、現在光造形法に使用されているのは、主に二種類の

組成の樹脂である。一方はアクリル系 (acrylic formulation) と呼ばれるもので、ラジカル重合 (photo radical polymerization) と呼ばれる光化学反応によって固化する樹脂である。もう一方はエポキシ系 (epoxy formulation) と呼ばれるもので、こちらはカチオン重合 (photo cationic polymerization) によって固化する。RP/Mで多く使われているのはアクリル系樹脂であるが、これは一般にエポキシ系樹脂よりも光感度 (photosensitivity) が大きいことによる。しかしアクリル系樹脂には、エポキシ系樹脂に比べて固化時の体積収縮が大きいという短所もあるので、両者は目的に応じて使い分けられている。

図3-2.1に、ラジカル重合の反応機構を示す。光硬化性樹脂に露光光線を照射すると、樹脂に含まれている光重合開始剤分子が、露光光線の光子の持っているエネルギーを吸収し、安定な基底状態 (ground state) から不安定な励起状態 (excited state) に移る。この現象は「活性化 (activation)」と呼ばれている。ただし光子のエネルギーが基底状態と励起状態のエネルギー準位差 ΔE よりも小さい場合、つまり入射光の波長が限界波長 λ_{crit} よりも長い場合には、光重合開始剤分子は活性化されず、以後の化学反応も起こらない。なお ΔE と λ_{crit} の間には、式3-2.1のような関係がある。

$$\lambda_{crit} = \frac{hc}{\Delta E} \quad 3-2.1$$

ただし h はプランク定数、 c は光速である。

光を吸収して励起状態となった光重合開始剤分子は一般に非常に不安定であり、すみやかに解離して、きわめて反応性に富む遊離ラジカルを放出する。

こうして生成した遊離ラジカルは樹脂中を拡散していき、光重合性オリゴマに含まれる官能基 (functional group) や反応性希釈剤分子を攻撃 (attack) して縮合反応を起こす。これらの官能基や分子はラジカルと結合すると活性化され、新たに攻撃性を持つようになる。こうして連鎖的に縮合反応が起こり、オリゴマ分子やモノマ分子が長く結合してポリマとなる重合反応 (polymerization) や、ポリマ分子の間に橋かけ結合 (bridge) が作られる架橋反応 (crosslinking) が進んでいく。

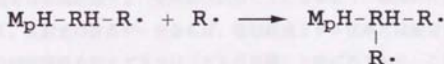
以上の化学反応によって、樹脂内部には立体的な網目構造が作られ、流動性が失われて樹脂は固形物に変化する。しかしこの状態では、樹脂は未反応のモノマ



1. Photoenergy dissolves photoinitiator molecules to radicals.



2. Radicals attack oligomer and monomer molecules.



3. Polymerization occurs to make molecular networks.

分子や官能基・揮発性成分を含んでおり、まだ完全な定常状態には至っていない。この状態は「グリーン (green) 固化状態」と呼ばれている。グリーン固化物を放置すると、外部から入射する光によって徐々に光化学反応が進み、また揮発性成分が加工物表面から揮散して、徐々に安定な完全固化状態へと変化していく。実際の光造形加工では「ポストキュア (後露光: postcure)」と呼ばれる処理により、この過程を人工的に行っている。

樹脂が固化する際には、樹脂に含まれる分子の間の平均距離が減少し、一般に収縮現象が見られる。その収縮量は、アクリル系の樹脂の場合には体積収縮率で約6%、すなわち線収縮率で約2%である。エポキシ系の樹脂の場合には、体積収縮率が約3%で、アクリル系の樹脂よりも小さい値を示す。この差異は両者の分子構造の違いから来るものである¹⁰⁾。

光硬化性樹脂の固化反応は、単位質量の樹脂が吸収する光エネルギーの量 (吸収線量: absorbed dose) に対して進行していく。吸収線量に関係のある量として「露光量 (dose of exposure)」と呼ばれるものがあるが、これは単位面積に入射する光エネルギー量のことである。樹脂の密度、および4-2節で詳述する「吸収係数」が一定ならば、吸収線量と露光量とは完全に比例対応する。よって第4章以後の議論では、主に露光量について考えることにする。

光硬化性樹脂に光が照射されると、光化学反応の進行に伴って樹脂に含まれるモノマ分子や官能基の含有量が減少していく。よって、これらの含有量で樹脂の固化の程度を表すことができる。これは「ゲル化率 (gel fraction)」と呼ばれており、通常は初期状態で0、完全固化状態で1となるように線形的に変換されて表される。入射光の波長が一定ならば、吸収線量とゲル化率とは完全に対応する。これらの対応関係を示すグラフは「ゲル化曲線」と呼ばれており、これは樹脂に固有の特性である。図3-2.2にゲル化曲線の一例を示す。

実際の光造形加工では、加工物はグリーン固化状態で取り出され、溶剤で洗浄された後、ポストキュアによって完全固化する。この溶剤洗浄は、加工物に付着している未固化の樹脂を取り除くために行うものであるが、この際にゲル化率が一定のしきい値に満たない部分は溶剤に溶け、未固化の樹脂とともに除去される。そこで、樹脂が溶剤に溶解なくなり、最終的な加工物の一部として残るようになる吸収線量を「限界吸収線量」と呼ぶことにする。そして限界吸収線量に対応

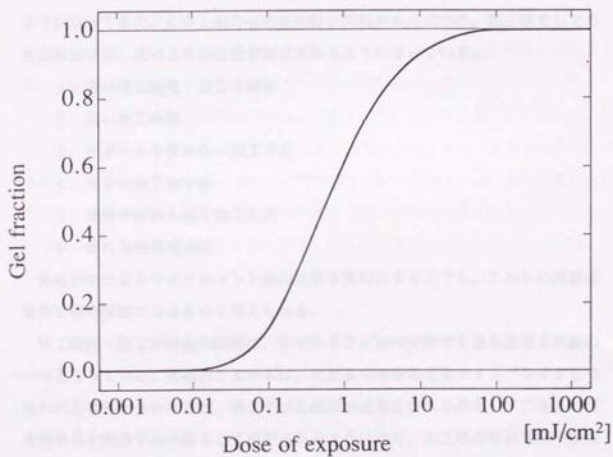


図3-2.2 ゲル化曲線の例

する露光量を「限界露光量」と呼ぶことにする。限界露光量は、事実上光硬化性樹脂が固化する露光量であると考えることができる。

第4章以降では、以上のような光硬化性樹脂の特性や固化原理を考慮しながら具体的な光造形法の改善指針を考える。

3-3. 光造形法の課題

3-1節でも述べたように、現在では光造形法は多くの応用分野で利用されるようになってきた。しかし新たな応用分野が開拓されるにつれ、加工法としての光造形法には、次のような仕様が要求されるようになっている。

1. 高い加工精度・加工分解能
2. 短い加工時間
3. サポートを使わない加工方法
4. 大きい加工物寸法
5. 複数の材料を使う加工方法
6. 優れた後処理方法

光造形法によるマイクロマシン製作技術を実用化する上でも、これらの問題は重要な解決課題になるものと考えられる。

加工精度・加工分解能の問題は、すでにRP/Mの分野でも最も重要な問題の一つとなっている。光造形システムは、これまで単なる立体ディスプレイとして使われる事例が多かったが、最近では生産用の金型を作る手段として、あるいは直接製品を製造する手段として利用されるようになり、加工物の形状精度や表面粗さがそのまま製品に反映されるようになったからである。また医療の分野でも、高い形状精度や表面粗さが要求されることが多くなってきた。もちろんマイクロマシン製作の分野では、加工精度・加工分解能は最も重要な要求仕様となる。

加工時間の問題は、加工精度・加工分解能の問題と同じく、RP/Mの分野でも最も重要な課題の一つとなっている。実際の設計現場では設計物の形状が頻繁に変更されることも多く、たくさんの実体模型が作られる。よって加工時間を短縮することは、そのまま設計時間の短縮につながる。また医療の分野では非常に短い加工時間が要求されることも多い。マイクロマシン製作の分野では、マイクロマシンの生産性を考える場合に加工時間が大きな問題となる。マイクロ

マシン製品には使い捨て製品も多いと考えられるので、その製法は大量生産に対応できるものでなければならない。したがって実際のマイクロマシンの生産に光造形法を利用するには、その加工時間が十分に短いことが重要である。

サポートを使わない加工方法も、マイクロマシン製作など多くの分野で実現が望まれている課題である。ステレオリソグラフィ法の光造形システムでは、樹脂液状物の中で加工物が作られる。しかし樹脂液状物は加工中の加工物を支持することができないので、加工物に力が加わると変形や破損が生じる可能性がある。こうしたトラブルを避けるため、「サポート (support)」と呼ばれる支持部材を目的の立体形状モデルに挿入する方法が使われる。多くの光造形システムでは、変形や破損の恐れがある立体形状モデルについて、あらかじめコンピュータ上で修正を行い、適切な位置に細いサポートを挿入してから実際に加工を開始する。こうして加工物とともに作られるサポートは、通常ポストキュアが行われた後で人手によって切り取られる。しかし微細で複雑なマイクロマシンを製作する際にサポートを使うと、これを加工後に取り外すことはきわめて難しいと考えられる。すでに固体下地法の光造形システムでは、サポートを使わない加工が実現されている。しかし固体下地法の加工装置は一般に大がかりであり、加工精度の維持が難しいのが難点である。

加工物寸法の問題は、加工時間の問題と同じく、マイクロマシンの大量生産を考える場合に重要な問題となる。大きい基板の上に多数のマイクロマシン製品を一度に作ることによって、一個あたりの加工時間を短縮することができるからである。また高さ方向の寸法の大きい加工物を作製することができれば、他の微細加工法では作れないような、大規模なマイクロマシンを製作することができる。

複数の材料を使う加工方法は、実体模型を製作すると同時に着色を行う方法として、RP/M・医療などの分野でも実現が望まれている課題である。マイクロマシン製作の分野では、微細な機械構造と同時に電子回路を構成する方法として期待されている。一つの加工物の内部で導電材料と絶縁材料を同時に作り分けることができれば、マイクロマシン内部に制御用の電子回路を組み込むことができ、きわめて高度な機能を持つマイクロマシンが実現できる。また、ある種の溶剤に溶ける材料と溶けない材料を作り分けることで、サポートを使わない加工を実現することができる。

後処理方法の問題も、多くの分野で解決が望まれている課題である。最近では加工精度の高い加工物が作られることが多くなってきており、溶剤洗浄・ポストキュアの際に加工物の変形や破損が起りにくい後処理方法が必要となっている。強さの小さいマイクロマシンを作る場合、この問題は特に深刻なものとなる。

文献

- (1) 小玉秀男, 3次元情報の表示法としての立体形状自動作成法, 電気通信情報学会論文誌 Vol. J64-C No. 4 (1981), pp. 237-241.
- (2) Jacobs, P. F., 高速3次元成型の基礎 (1993), 日経BP出版センター.
- (3) 丸谷洋二, 光硬化性樹脂による造型法の原理, 合成樹脂 Vol. 36 No. 10 (1990), pp. 7-10.
- (4) 岸浪建史, 田中文基, 光造形法におけるソフトウェア処理のモジュール化とインターフェース, 第3回光造形システムシンポジウム資料 (1992), pp. 22-26.
- (5) 小野一郎, 群司裕則, 金子史男, 片倉俊彦, 木村和衛, 八子恵子, 大河内幸男, 相川通, Helical volume CTを用いた顎顔面骨の三次元表示法とその顎面骨骨折への臨床応用, 形成外科 Vol. 35 No. 9 (1992), pp. 939-950.
- (6) 藤野豊美, シミュレーション外科における光造形法の利用, 第3回光造形システムシンポジウム資料 (1992), pp. 45-48.
- (7) 小松孝雪, 齋藤華織, 増木英郎, 久恒泰宏, 木村朋義, 上田康夫, 内山洋一, 山口泰彦, 光造形システムを用いた下顎運動範囲の立体モデル化。～切歯点における下顎限界運動～, 第90回日本補綴歯科学会学術大会講演論文集 (1993).
- (8) 上田康夫, 疋田一洋, 齊藤比登志, 内山洋一, 光造形システムを用いた歯冠形態の復元 (CAM) に関する研究, 第19回日本歯科理工学会学術講演会講演論文集 (1992).
- (9) 孟陽, 今村正人, 魏杰, 中川威雄, 3次元光造形モデルのロストワックス精密鋳造への適用, 1992年度精密工学会春期講演大会講演概要集 (1992), p. 319.
- (10) 清水誠司, 阿部正人, 萩原茂, 中島俊, 古川進, 伊藤誠, 宝飾工芸品の

CAD/CAMシステム, 第6回NICOGRAPH論文コンテスト
論文集(1990), pp. 298-307.

- (11) 大谷利勝, 鋳造用模型に用いられる光造形技術, 合成樹脂 Vol. 36 No. 10 (1990), pp. 15-19.
- (12) (たとえば) デザインの現場 1991年12月号(1991), p. 52.
- (13) 伊藤忠夫, 光造型に用いられる光硬化樹脂, 合成樹脂 Vol. 36 No. 10 (1990), pp. 20-25.
- (14) 斎藤勝政, 光造型法による模型の製作技術, 素形材 1989年10月号 (1989), pp. 6-12.
- (15) Chikaoka, S. and Ohkawa, K., Characterization of Laser Photopolymerization - Linear Shrinkage on Curing, Conference Proceedings of RadTech Asia '93 (1993), pp. 468-472.

第4章 加工分解能と固化セルの考え方

4-1. 固化セルと加工精度・加工分解能

光造形法の加工分解能を高める方法について議論するに当たり、最初に「加工精度 (machining accuracy, forming accuracy)」と「加工分解能 (machining resolution, forming resolution)」の違いを明確にしておきたい。

まず「加工精度」は、設計寸法と加工物寸法の差のことである。これに対して「加工分解能」は、独立に加工できる2点間の最小距離のことである。すなわち加工精度と加工分解能は別の概念であり、加工分解能が低く加工精度が高い例、加工分解能が高く加工精度が低い例は、どちらもあり得ないものではない。

通常の寸法の機械の場合には、加工分解能は一般に表面粗さという形で表され、加工精度は寸法精度という形で表される。加工分解能と加工精度はどちらも機械製品の性能に影響を与える要素であるが、加工精度は加工物の代表寸法に対する相対値で議論されることが多いのに対し、加工分解能は絶対値で議論されるのが普通である。よってマイクロマシンのように加工物の代表寸法が小さい場合には、加工分解能が加工精度を決定することが多い。そのため微細加工法の仕様を評価する際には、もっぱら加工分解能について議論される。

加工物寸法が比較的大きいRP/Mでは、加工分解能よりも加工精度が問題となることが多い。特にステレオリソグラフィ法の光造形システムでは「余剰固化 (overcuring)」と呼ばれる現象が見られ、垂直方向に非常に大きい寸法偏差が発生する。この問題に対し、あらかじめ対象モデルを補正しておき、余剰固化による寸法偏差を取り除く方法が考案されている⁽¹⁾。また加工シーケンスを工夫し、積層による段差を平滑化する方法も考案されている⁽²⁾。さらに樹脂の収縮による加工物の変形量を有限要素法によって予測し、あらかじめ対象モデルを補正しておく方法も研究されている⁽³⁾⁽⁴⁾。

一方で加工分解能を高めることはRP/Mにおいても重要な課題となっており、すでに約40 μ mの分解能を持つ光造形システムも作られている。しかし各種のマイクロマシンの製作するには、10 μ m以下の加工分解能が必要とされている。したがって市販のRP/M用光造形システムを、そのままマイクロマシン製作に利用することは困難であり、何らかの指針に基づく改善が求められる。

この指針を導くため、本研究では「固化セル (solidified cell)」という概念を導入した。これは「モデルが三次元ビットマップによって記述されており、その1セルに対応する露光が、一定の露光条件で光硬化性樹脂の表面に行われる時に、樹脂が固化する空間領域」と定義されるものである。なお「三次元ビットマップ (three-dimensional bit map)」という言葉は、モデルを記述するデータ形式の一種を指す。このデータ形式では、モデルを含む空間に三次元の格子点を設定し、各格子点を三次元配列の配列要素に対応させる。配列要素は1ビットの大きさを持つデータで、対応する格子点がモデルの内部、または外部のどちらにあるかを表す。格子間隔を十分に小さく設定しておけば、この三次元配列全体でモデルを記述することができる。この三次元配列を構成する1ビットの配列要素は「セル (cell, voxel)」と呼ばれている。

固化セルは、近似的には光造形法で作られる加工物の構成単位であると考えることができる。しかし実際の加工では加工物に強度を与え、あるいは加工時間を短縮するために、固化セルが互いに重なり合うようにするのが普通である。また2-2節でも述べたように、光硬化性樹脂はある吸収線量に達すると急峻に固化するものではなく、一般に樹脂のゲル化率は吸収線量に対して連続的に変化する。したがって固化セルは、各点での吸収光量によって決まるゲル化率の空間分布で表されるべきものであり、明確な空間領域として定義することはできないので、厳密には加工物の構成単位には当たらないものである。

しかし光造形法の加工分解能が固化セルの寸法・形状に大きく依存しており、高い加工分解能を実現するために固化セルの寸法を小さくする必要があることは明らかなことである。また加工物の表面では固化セルが露出しているのも、その形状・寸法が加工分解能や加工物の表面粗さに大きく影響する。したがって加工分解能を改善するには、寸法が十分に小さく、水平方向・垂直方向の寸法がほぼ等しい、いわゆる等方的な (isotropic) 形状を持つ固化セルを作ることが特に重要である。

そこで本研究では、良好な固化セルが得られる加工条件を見つけるため、加工条件の違いによる固化セルの形状・寸法の変化を予測できる実用的な方法を開発することにした。

樹脂に露光を行う際の露光量を小さくしていくと、最終的には固化セルが消滅

する。よって十分に小さい寸法を持つ固化セルを作成することは原理的に可能であり、そのような加工条件が存在することも保証されている。しかし良好な固化セルが得られる加工条件の範囲は一般に狭く、振動などの外乱の影響を受けやすいので、従来の光造形システムでは実現しにくかったものと考えられる。したがって加工条件を求める際には、外乱に対する余裕について考慮することが重要である。

4-2. 固化セルの寸法・形状の予測

固化セルの寸法を最小とする加工条件を求めるには、与えられた条件に対して固化セルの寸法・形状を求める手段が必要である。本研究では、樹脂中での光の挙動を調べるコンピュータシミュレーションを考案し、これによって固化セルの寸法・形状を求めることにした。このシミュレーションは、一定の条件で多数の試行を行うことで統計的に近似解を求める、いわゆるモンテカルロ法 (Monte Carlo method) の一種である。シミュレーションモデルを図4-2.1に示す。

このシミュレーションでは、光の透過 (transmission)・散乱 (scattering)・吸収 (absorption) の三つの物理現象を考えている。まず図4-2.1に示した樹脂表面の入射点Aから、一定の運動ベクトルを持つ光子を樹脂中に入射させるものとする。すると光子は樹脂中で透過と散乱を繰り返し、最終的に吸収点Bで吸収されて、その点にある光反応開始剤分子を励起させる。そこで多数の光子の軌跡を追跡し、その吸収点の空間分布を調べることによって、入射光がどの点にどれだけの光固化反応を引き起こすかについて定量的に知ることができる。なお光子を追跡する計算のフローチャートを図4-2.2に示す。

この計算を行うには、二つのパラメータが必要である。一つは樹脂の吸収係数 (absorption coefficient) で、これは樹脂が光を吸収する程度を表す値である。媒質中を進む光の光量が透過距離 x について $1/r$ に減衰する場合、吸収係数 α は式4-2.1のように定義される。

$$\alpha = \frac{\ln(r)}{x} \quad 4-2.1$$

もう一つのパラメータ σ は「散乱比」と呼ぶことにする。これは樹脂が光を散乱する程度を表す散乱係数の、吸収係数に対する比である。光が媒質中を進む時の減衰は、吸収によるものと散乱によるものとに分けることができる。そこで

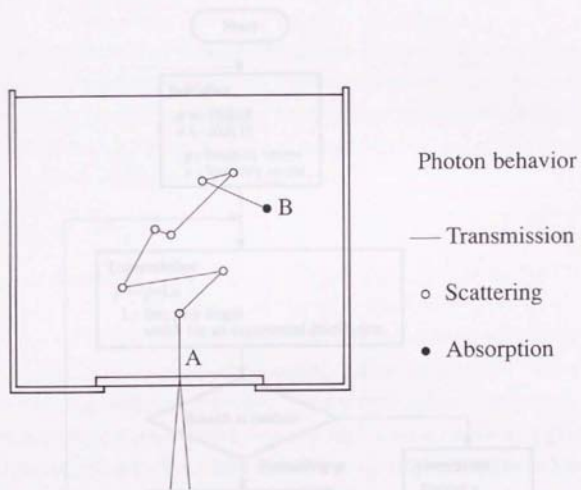


図4-2.1 シミュレーションモデル

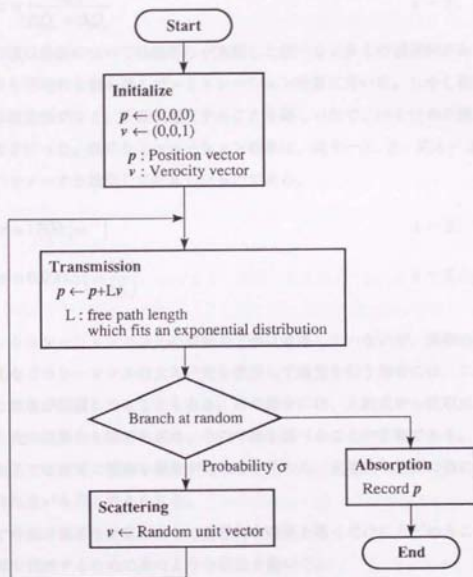


図4-2.2 シミュレーション計算のフローチャート

無限に小さい直径と十分に大きい長さを持つ円柱形の媒質の中を、その中心軸に沿って光が進む場合を考える。この時に光量 ΔQ_0 が媒質に吸収されて熱となり、光量 ΔQ_s が媒質に散乱されて円柱側面から外部に放出される場合、散乱比 σ は式4-2.2のように定義される。

$$\sigma = \frac{\Delta Q_s}{\Delta Q_a + \Delta Q_s} \quad 4-2.2$$

樹脂の吸収係数については植原らが実測した値⁽¹⁾など多くの測定例があるので、その中から平均的な値を選んでシミュレーション計算に用いた。しかし散乱比については測定例がなく、正確に測定することも難しいので、いくつかの値を代入して計算を行った。後のシミュレーション結果は、式4-2.3・式4-2.4のようにパラメータを設定して計算したものである。

$$\alpha = 1200 [m^{-1}] \quad 4-2.3$$

$$\sigma = 0.9375 \left(= \frac{15}{16} \right) \quad 4-2.4$$

このシミュレーションでは光の回折や干渉は考慮していないが、実際の加工でレーザー光などコヒーレンスの大きい光を使用して露光を行う場合には、これらの光学的な現象が問題となることもある。その場合には、入射点から吸収点までの経路長と光の波長から位相を求め、その干渉を調べることが必要である。しかし実際の加工では非常に複雑な現象が起こると思われ、定量的な結果は特に大きな意味を持たないものと考えられる。

そこで今回は簡単な計算によって定性的な結果を導くだけにとどめることにし、計算時間を短縮するために次のような仮定を置いた。

1. 光固化反応に伴う、樹脂の屈折率・吸収係数・散乱比の変化は無視することができる。樹脂の吸収係数と散乱比は、樹脂に含まれている物質の電子状態によって決まる。しかし樹脂の主成分である光重合性ポリマの分子内の化学結合のうち、光固化反応の際に変化するのはごくわずかな部分だけである。よって樹脂のゲル化率は、吸収係数と散乱比に大きく影響しないものと考えられる。また樹脂の屈折率は、固化の際の収縮に伴って変化する。しかし固化セルが等方的な形状を持つ場合、屈折率の

変化による吸収点の変動はあまり大きくならないものと考えられる。

2. 樹脂の固化が起こる限界露光量は、洗浄方法や加工物の部位に関係なく一定の値を持つ。実際の限界露光量は、未固化の樹脂を除くための溶剤洗浄の条件に関係するものである。加工物の寸法が大きい場合、加工物表面の樹脂は、加工物内部の樹脂よりも多くの溶剤に接触しているので、溶解されやすく、そのため加工物表面では限界露光量が多い。しかし加工物が微細な場合には、溶剤が十分に加工物内部にまで浸透するので、このような影響は小さいものと考えられる。
3. 樹脂は均一であり、また光学的に等方性である。市販の光硬化性樹脂は単一の液相からなる液体であり、この条件を満たすものと考えられる。また樹脂に固体粉末などが混入されている場合にも、粉末の平均粒径や粒子間の平均距離が固化セルの寸法に対して十分に小さければ、やはりこの条件を満たすものと考えられる。
4. 窓板と樹脂の界面における光の散乱・基板表面や加工中の加工物による光の反射や散乱は無視することができる。加工分解能を改善することを考えると、固化セルが独立に作られるような加工条件を設定しなければならない。その場合に固化セルの形状は、周囲の影響を受けないものと見なすことができる。
5. 光固化反応の進行は露光時間に関係なく、露光量だけで決まる。感光性物質に露光線が照射される場合には、感光性物質内部の流れや拡散を無視できるものとする、光化学反応は一般に単位質量当たりの物質が吸収する露光量に対して進むことが知られている。

このようにして得られたシミュレーション結果を図4-2.3に示す。図中のパラメータ p は、光子1個の入射に対し、各点の近傍にある単位体積当たりの空間によって吸収される光子の個数を表している。また曲線は、 p の等値面の断面を示している。今回のシミュレーションでは、655万3600個の光子の軌跡を追跡した。

露光ビームの幅が非常に小さい場合には、図4-2.3の曲線はそのまま固化セルの成長を表している。この場合、 p の逆数と露光量は比例対応する。一方露光ビームが有限の幅を持つ場合には、露光ビームを露光面で切った断面（露光

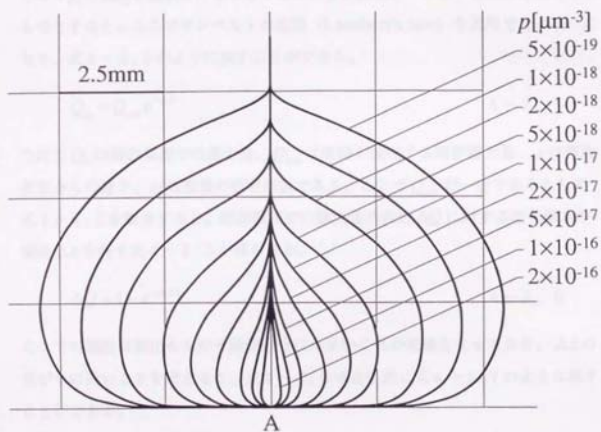


図4-2.3 シミュレーション結果

スポット：exposure spot) について、 p のたたみ込み積分 (convolution) を計算することで固化セルの成長を知ることができる。

このシミュレーション結果から、固化セルの寸法を小さくしていくと、セルの形状が細長くなり、また散乱比の影響が小さくなることがわかった。これは光が樹脂中を進む距離が短くなるため、散乱粒子に出会う割合が低くなることによる。したがってこのような条件では、樹脂中での光の散乱を無視することができる。そこで光の透過と吸収だけを考えることにし、露光ビームを平行光線と見なせるものとする、ここでランベルトの法則 (Lambert's law) を適用できるようになり、式 4-2.5 のように表すことができる。

$$Q_{in} = Q_{crit} e^{-\alpha x} \quad 4-2.5$$

ただし Q_{in} は樹脂表面での露光量、 Q_{crit} は樹脂が固化する限界露光量、 x は樹脂表面からの深さ、 α は樹脂の吸収係数である。ここで Q_{crit} が一定であるとして式 4-2.5 を微分すると、樹脂表面での露光量の変化 ΔQ に対する固化深さの変化 Δx を表す式 4-2.6 が得られる。

$$\Delta Q = 1 - e^{-\alpha \Delta x} \quad 4-2.6$$

ここでの議論は固化セルの寸法が十分に小さいことが前提となっており、 Δx の値が 0 に近いことを考えると、式 4-2.6 は近似的に式 4-2.7 のように表すことができる。

$$\Delta Q = \alpha \cdot \Delta x \quad 4-2.7$$

式 4-2.7 から、樹脂の固化深さは樹脂表面での露光量に対して線形的に変化することがわかる。すなわち、樹脂表面での露光量が限界露光量に達するまでは樹脂の固化は起こらず、その後は露光量に比例して固化深さが増大していく。

4-3. 加工分解能を高めるための指針

シミュレーションと数式解析の結果を整理すると、高い加工分解能が得られる加工条件では、次のような現象が起こることが予想される。

1. 光が樹脂中の散乱粒子に出会う確率が小さくなり、樹脂が透明になった

ものとして考えることができる。

2. 光の散乱の影響が少なくなり、水平面内では露光ビームが直接入射する部分だけで樹脂の固化が起こる。
3. 露光量が一定量に達するまで樹脂の固化は起こらず、その後は露光量に比例して固化深さが増大する。
4. 固化深さの相対誤差を一定に保つ場合、露光量に許される誤差は一層の厚さに比例して減少する。

ここで加工分解能について考えると、上記の2.が水平方向、4.が垂直方向の加工分解能を決める主要因となっていることがわかる。これから加工分解能を高めるための指針を導くと、次のようになる。

1. 露光ビームの放射量を安定化する。
2. 露光スポットの直径を安定化する。
3. 樹脂の吸収係数を大きくする。

これらの改善指針を実際の加工に適用する具体的な方法については、第5章で詳述することにする。

文献

- (1) 後藤典雄, 小澤雅彦, 森谷真寿美, 遠藤敏朗, 大久保賢勉, 光造形における寸法偏差と高精度化技術, 第5回光造形システムシンポジウム資料 (1993), pp. 36-39.
- (2) 榎原弘之, 齋藤勝政, 生体形状のリバースエンジニアリングのための光造形法, 第7回ラピッド・プロトタイプングシンポジウム資料 (1994), pp. 34-42.
- (3) 田中文基, 岸浪達史, 光造形における収縮現象のシミュレーション, 第5回光造形システムシンポジウム資料 (1993), pp. 44-47.
- (4) 井関日出男, 簡易FEMシミュレーションによる反り変形解析の試み, 第7回ラピッド・プロトタイプングシンポジウム資料 (1994), pp. 43-49.
- (5) 榎原弘之, 3次元立体造形のための樹脂硬化特性, 合成樹脂 Vol. 36 No. 10 (1990), pp. 11-14.

第5章 マイクロマシン製作用光造形システムの試作

5-1. 加工装置の設計

第4章では加工分解能を高めるための指針を得るために理論検討を行ったが、この結果の妥当性を実験的に検証し、さらに光造形法を利用して機械構造を作製できることを確認するために、マイクロマシン製作用の実験用光造形システムを試作することを考えた。

実験用光造形システムの仕様としては、加工条件を広い範囲で変えられること、露光条件をきわめて正確に制御できることなどが求められる。そこで本研究では、市販のRP/M用光造形システムを流用することをやめ、自作することにした。

本研究で製作した加工装置は、次のような特徴を持っている。

1. 露光ビームの遮断に高速ビームシャッターを採用しており、各セルに対応する放射量を正確に制御することができる。
2. 露光ビームの伝達に光ファイバを用いず、平面鏡の組み合わせによってこれを実現している。そのため露光スポットを十分に小さくすることができ、露光ビームの集光条件を正確に設定することができる。
3. 制御装置にはパーソナルコンピュータを用い、造形装置ハードウェアと制御用ソフトウェアを独立させている。そのため加工シーケンスの比較検討を容易に行うことができる。

加工装置を自作するに当たっては、まず光造形法を実現する加工方式を決める必要がある。市販の光造形システムに採用されている各種の加工方式については、3-2節で紹介したとおりである。本研究では、造形装置の構成が比較的簡単なステレオリソグラフィ式を採用した。また第4章で得られた加工分解能を高めるための指針を考慮し、各セルに対応する放射量を独立に変えることのできる走査露光式を採用した。さらに樹脂表面の制御方法としては、垂直方向の加工精度を容易に確保することができる規制液面式を採用した。

加工装置は、実際に造形加工を行う「造形装置 (forming device)」と、造形装置を制御する「制御装置 (controller)」から構成されている。制御装置にはパーソナルコンピュータを使用し、これにTTLレベルのパラレル入出力を行う拡張ボードを通して造形装置の電子回路を接続した。電子回路の回路図・仕様・

接続方法などの詳細については付録に示す。

造形装置は、光源・ビーム調整器・ビームシャッター・光量計・スキャナ・露光ヘッド・エレベータ・樹脂槽・電子回路などから構成されており、全体は振動を避けるために防振器を備えた光学定盤の上に固定されている。造形装置の構成を図5-1.1に、仕様を表5-1.1に示す。

光源 (light source) には2種類のレーザを用いた。一つは紫外光を放射するヘリウム・カドミウムレーザ (helium cadmium laser/He-Cdレーザ: He-Cd laser) で、もう一つは可視光を放射するアルゴンイオンレーザ (argon ion laser/Ar⁺レーザ: Ar⁺ laser) である。紫外光は波長が短いため、各種の樹脂を高い効率で硬化させることができる。しかし有機化合物の紫外光に対する吸収係数には上限があるため、特に大きい吸収係数が必要な場合には、可視光を用いて露光を行う必要がある。またAr⁺レーザは光出力のノイズが少ないので、容易に露光ビームの放射量を安定化することができる。なお可視光による露光については6-3節で詳述する。

ビーム調整器 (beam regulator) は、露光ビームの放射束や直径を調整するものである。放射量を制御するにはビームシャッターの開放時間を調節する方法もあるが、放射量が特に小さい場合には、開放時間を正確に調節することが難しい。そこでビーム調整器に取り付けられたフィルタで露光ビームの放射束を小さくし、露光時間を大きくすることで放射量の制御を容易にする。またレーザ光のようなコヒーレンスの大きい光で露光を行う場合には、露光ビームを集光レンズで集光すると顕著な干渉じまが現れ、水平方向の加工分解能が低下する。そこでビームエキスパンダ (beam expander) を用いて露光ビームの直径を大きくし、露光ビームを集光する際の開口数 (numerical aperture) を大きくして、干渉じまの発生を抑える。

ビームシャッター (beam shutter) は、露光ビームを高速で開閉することにより、各セルに対する正確な放射量の露光を実現するものである。加工分解能が比較的低いRP/M用光造形システムでは、スキャナの移動速度を変えることによって露光量を調節することが多い。しかしマイクロマシンを製作する場合には、特に正確な露光量の制御が要求されるので、スキャナの移動速度を増減するだけでは十分に対応することができない。そこで露光ビームの光路の途中に高速のビーム

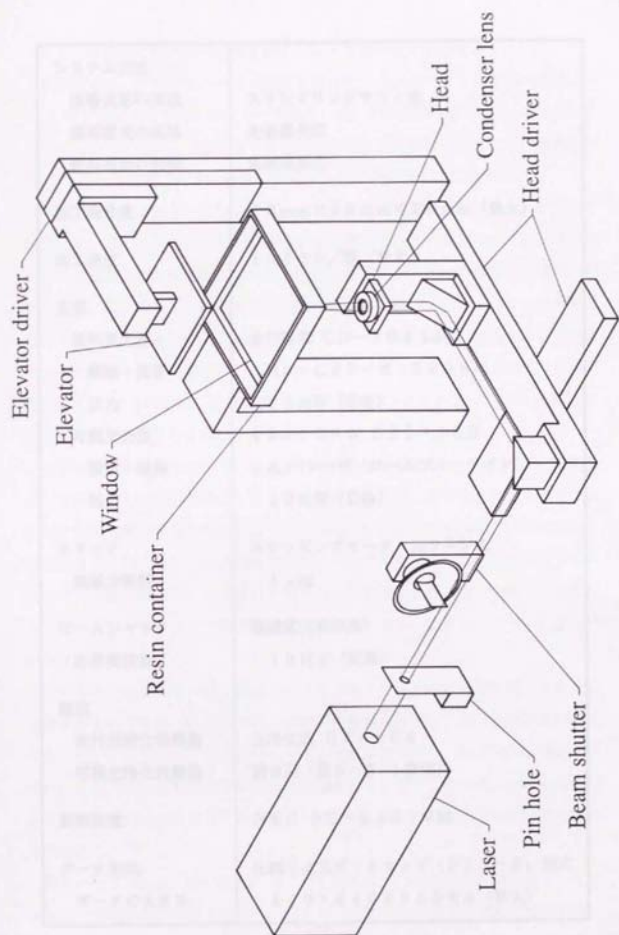


図5-1.1 実験用の造形装置の構成

システム方式	
積層造形の実現	ステレオリソグラフィ法
描画露光の実現	走査露光式
樹脂表面の制御	規制液面式
加工物寸法	20mm×20mm×20mm (最大)
加工速度	100セル/秒 (最大)
光源	
紫外光光源	金門電気 CD-3041R
種類・波長	He-Cdレーザー・325nm
出力	10mW (公称)
可視光光源	オムニクローム 532-10B
種類・波長	Ar+レーザー・オールブルーライン
出力	10mW (公称)
スキャナ	ステッピングモータ・送りネジ式
機械分解能	1 μ m
ビームシャッター	電磁式 (自作品)
応答周波数	1kHz (実測)
樹脂	
紫外光硬化性樹脂	三洋化成 UV-854
可視光硬化性樹脂	調合品 (表6-3.1参照)
制御装置	NEC PC-9801VM
データ形式	圧縮三次元ビットマップ (PFデータ) 形式
データの大きさ	640×640×960セル (最大)

表5-1.1 実験用光造形システムの仕様

シャッタを置き、その開閉時間を変えることで放射量を調節する。

本研究では、市販のスピーカ（直径80mm・インピーダンス 8Ω ・定格入力0.5W）をもとにビームシャッタを製作した。その構造を図5-1.2に示す。このビームシャッタに+5Vと-5Vの電圧を交互に印加したところ、遮光板が約1.5mmの振幅で動き、露光ビームを十分に開閉することができた。

そこで後述する光量計を用い、ビームシャッタを通過する露光ビームの波形を観察して、ビームシャッタの応答周波数を実測した。ビームシャッタに送られるパルスの幅と実際の放射量の関係は一般に非線形的であり、放射量を正確に制御する場合には、あらかじめ両者に対応させるための校正を行っておく必要がある。そこでアナログ光量計の出力波形をオシロスコープで観察し、波形を時間で積分することによって放射量を求めた。パルス幅と放射量の関係を表す校正グラフの例を図5-1.3に示す。この実測により、ビームシャッタが500Hz以上の応答周波数を持つことを確認することができた。

光量計（photometer）は、露光ビームの放射束を測定するためのものである。レーザの光出力は運転を開始してからの時間に依存し、また経年的にも変化する。そのため放射量を正確に制御するには、露光の前に光出力を測定し、露光時間を変えるなどの補正を行う必要がある。そこでリアルタイムで放射束を測定できるアナログ光量計を製作し、市販の光量計によって校正を行った。光量計の出力はA/Dコンバータを通して制御装置に接続されており、制御プログラムによって参照することができる。

スキャナ（scanner）は、露光ヘッドやエレベータを駆動して三次元の走査を実現するものである。本研究では、市販の位置決めステージを3台組み合わせる製作した。このうちの2台は露光ヘッドを駆動するもので、加工物の断面形状を描き、それぞれの層を作る水平方向の走査を行う。残る1台はエレベータを駆動するもので、それぞれの層を積み重ね、立体的な加工物を作るための垂直方向の走査を行う。

露光ヘッド（exposure head）は、平行光線として送られてくる露光ビームを集光して樹脂表面に照射しながら、水平面内を走査して加工物の断面形状を描くものである。露光ヘッドには集光レンズ（condenser lens）と、数枚の反射鏡が取り付けられており、露光ヘッドの位置に関係なく、露光ビームが集光レンズに

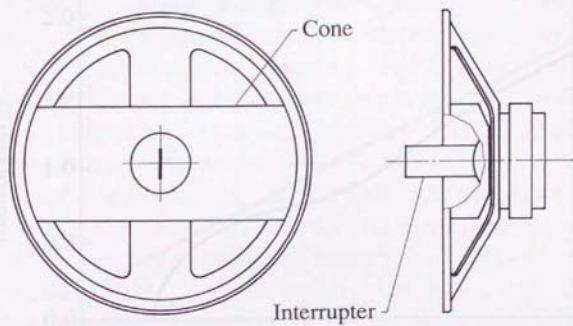


図 5-1.2 ビームシャッタの構造

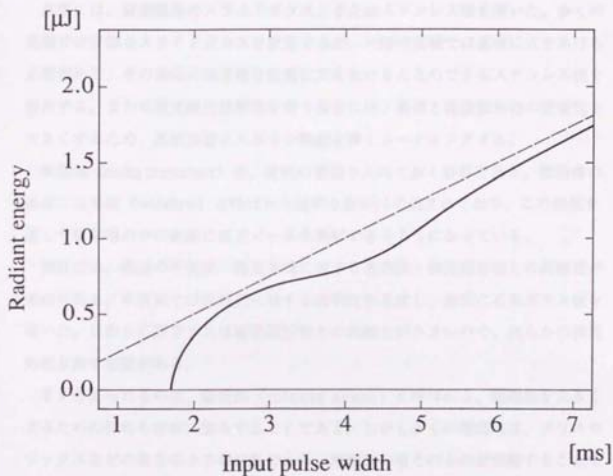


図5-1.3 ビームシャッタ制御出力のパルス幅と放射量の関係

正しく導かれるようになっている。集光レンズには、紫外光に対する透過損失が小さい、石英ガラス製のものを用いた。

エレベータ (elevator) は、一層の加工が終わった加工物を窓板から引き離し、加工物の下に未固化の樹脂の新しい層を作るものである。エレベータの底部にはスライドガラスなどの基板を保持できるようになっており、基板の下に加工物が下向きに成長していく。

基板には、顕微鏡用のスライドガラス、またはステンレス板を用いた。多くの実験では安価なスライドガラスを使用するが、一部の実験では基板に穴をあける必要があり、その場合には正確な位置に穴をあけることのできるステンレス板を使用する。また可視光硬化性樹脂を使う場合には、基板と樹脂固形物の接着性を大きくするため、基板表面にエポキシ樹脂を薄くコーティングする。

樹脂槽 (resin container) は、材料の樹脂を入れておく容器である。樹脂槽の底部には窓板 (window) と呼ばれる透明な板がはめ込まれており、この窓板を通して樹脂槽の中の樹脂に露光ビームを照射できるようになっている。

窓板には、表面の平面度・露光光線に対する透明性・樹脂固形物との剥離性が求められる。本研究では紫外光に対する透明性を考慮し、窓板に石英ガラス板を用いた。しかし石英ガラスは樹脂固形物との剥離性が小さいので、何らかの表面処理を施す必要がある。

まず考えられるのは、離型剤 (release agent) と呼ばれる、剥離性を大きくするための材料を窓板に塗布することである。しかし多くの離型剤は、グリスマワックスなどの強さの小さい材料であり、離型剤の層そのものが分離することで剥離性を実現するものである。そのためこのような離型剤を用いると、加工物の表面が離型剤で汚染されて次の層が加工物に接着しなかったり、離型剤が徐々に消費されて剥離性が継続しなかったりするなどのトラブルが生じ、良好な結果が得られない。したがって窓板の表面処理には、ある程度の強さを持つ固体材料を使わなければならない。

また窓板に融点の異なる物質を塗っておき、露光による樹脂の温度上昇を測定する実験を行ったところ、露光スポット付近の温度は瞬間的に100℃を超えることが確認された。このことを考えると、窓板の表面処理には、透明性・強さに加えて耐熱性も必要となる。

多くの物質に対して大きい剥離性を持つ固体としては、グラファイト・二硫化モリブデン・パラフィン・ふっ素樹脂・シリコン樹脂などがよく知られている。本研究では透明性・強さ・耐熱性を考慮し、ふっ素樹脂とシリコン樹脂を使った以下の表面処理方法について、剥離性を評価する実験を行った。

1. テフロン[®] PTFE樹脂の粘着テープを貼り付ける方法
2. テフロン PFA[®]樹脂の粘着テープを貼り付ける方法
3. テフロン FEP[®]樹脂を焼き付けコーティングする方法
4. テフロン PFA[®]樹脂フィルムを熱融着する方法
5. シリコン樹脂を焼き付けコーティングする方法
6. シリコンゴムを塗布し、シリコンオイルを含浸する方法

この実験では、ふっ素樹脂を使った処理方法の方が、全体的にシリコン樹脂を使った処理方法よりも良い成績を示し、粘着テープ・コーティング・フィルムのいずれの場合にも良好な剥離性が得られた。中でも粘着テープ・フィルムによる処理方法は、コーティングによる処理方法を上回る優れた成績を示した。これは表面処理層の表面粗さが剥離性に影響しているものと思われる。なお表面処理の寿命についてはコーティングによる処理方法が優れた成績を示し、粘着テープ・フィルムの場合の3倍以上の耐久性が得られた。

窓板と樹脂固形物の剥離性を大きくするため、圧電アクチュエータを使用して窓板を振動させ、樹脂固形物の剥離を促進させる方法についても考えた。しかし実際に実験を行ったところ、加工中の加工物が振動を受けて破損するトラブルが多く発生した。また窓板の表面処理層の磨耗が早くなり、表面処理の寿命も短くなった。加工物が樹脂液状物の中で作られるステレオリソグラフィ法では、この方法を利用することは難しいと考えられる。

評価実験の結果を踏まえ、本研究では最終的に窓板表面に厚さ約200 μm のテフロン PFA[®]樹脂の粘着テープを貼り付けて使うことにした。このテープは耐久性の点ではコーティングに劣るものの、表面が平滑で非常に優れた剥離性を示すものである。

以上の加工装置の全体、および各部の外観を写真5-1.1~写真5-1.6に示す。

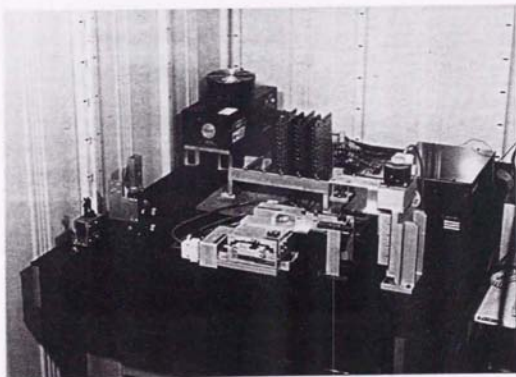


写真5-1.1 加工装置の外観(全体)

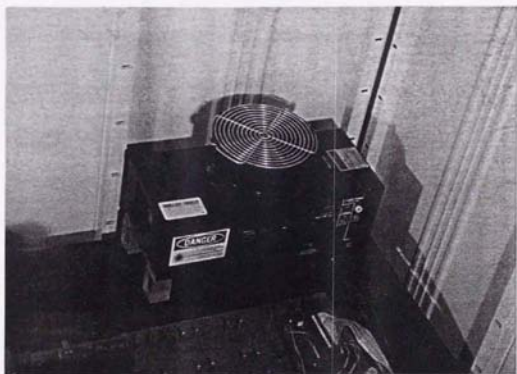


写真 5-1.2 加工装置の外観 (レーザー)



写真5-1.3 加工装置の外観（ビーム調整器・ビームシャット）

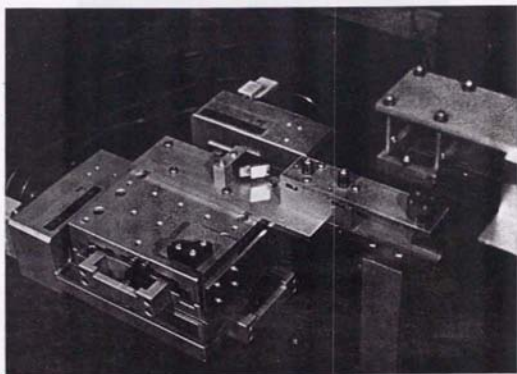


写真5-1.4 加工装置の外観（露光ヘッド）

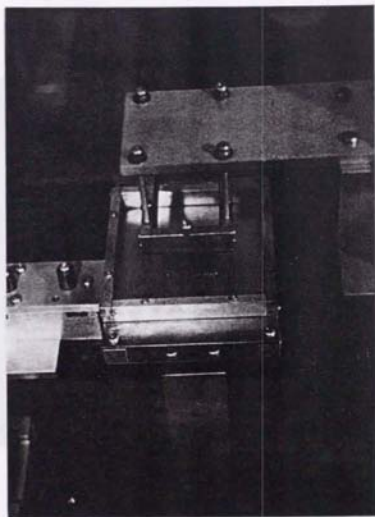


写真5-1.5 加工装置の外観 (エレベータ)

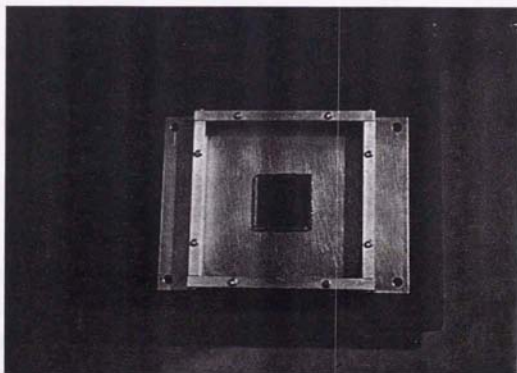


写真5-1.6 加工装置の外観（樹脂槽）

5-2. 加工シーケンスの設計

第4章で得られた改善指針に基づく加工条件を実現するため、本研究では加工装置制御プログラム「PFMAKE」を作成した。PFMAKEは、フロッピーディスクに記録されているモデル記述データを読み取り、そのモデルに基づいて造形装置を制御するための加工シーケンスを生成する。このプログラムによって生成される加工シーケンスは、次のような特徴を持っている。

1. 加工物内部を均一に露光し、後処理の際に起こる加工物の変形や破損を抑えるため、すべてのセルについての露光条件を描いている。そのため走査アルゴリズムとしてベクトル走査の代わりにラスタ走査を採用し、また連続露光の代わりにパルス露光を採用している。
2. スキャナのバックラッシュの影響を取り除くため、走査アルゴリズムとしてラスタ走査を採用している。さらに走査経路を工夫し、スキャナが目的の位置に到着する直前の移動方向を描いている。
3. 樹脂液状物の粘性・固形物の弾性や強さの影響を考慮している。加工物寸法が小さい場合には、加工物の破損や変形を防ぐため、エレベータを制御する際に寸法効果について配慮することが重要である。

PFMAKEは加工を行う前に、操作者に対して加工原点の指定を求めてくる。ここで操作者は露光ヘッドとエレベータを移動させ、加工原点を指定する。この操作で、エレベータは窓板のわずかに上の位置、露光ヘッドは基板の原点位置に合わせられる。このようにして指定された原点は、以後の制御の基準点となる。

こうして原点が決まると、PFMAKEは窓板と基板のすき間に残されている樹脂全体に対し、大きい露光量で露光を行って全体を固化させる。規制液面式の加工装置では、窓板と基板のすき間が固化セルの垂直方向の寸法よりも大きいと、加工物を窓板表面から引き離すことができなくなり、加工を続けられなくなる。このトラブルを避けるには、窓板と基板を平行に保持し、両者を十分に近づけておかなければならない。しかし固化セルの垂直方向の寸法が数 μm であるようなマイクロマシン製作の場合には、基板の正確な保持はかなり困難である。そこですき間に残されている樹脂全体を固化させることで、すき間をなくすようにする。なお、この加工工程によって樹脂が固化する部分を「基礎部」と呼ぶことにする。

こうして基礎部を作ると、加工物は基礎部によって基板に固定される。そこで

基礎部の水平方向の寸法を加工物の寸法よりも大きくして、樹脂固形物と基板の接触面積を大きくすることにより、加工物を強く基板に固定することができる。

基礎部を加工する際には、非常に大きい放射束を持つ露光ビームを用いるので、集光された露光ビームが窓板の表面処理層や樹脂に吸収されて大きい熱が発生し、表面処理層や樹脂が変質したり焼損したりする恐れがある。そこで集光レンズの焦点を外し、発生する熱を分散させて温度上昇を抑える。このため造形装置は、加工中に集光レンズの焦点位置を変えることができるように作られている。

基礎部は窓板と接触している面積が大きいので、基礎部全体を固化させてから引き離すには特に大きい力が必要になる。この力が窓板に加わると、窓板の表面処理層を破損する恐れがある。そこで基礎部と窓板の接触面積が一定の大きさに達したら、エレベータを一度上昇させて接触部分を引き離し、再び元の位置まで下降させるようにすることで、接触面積が過大になることを避ける。

加工中に起こるトラブルは、基礎部を作製する加工工程に集中している。特に加工物が窓板表面に固着してしまい、窓板表面から引き離すことができなくなるトラブルがよく見られるが、そのほとんどは基礎部を作製している時に起こる。よって基礎部の作製中には、加工状態を頻繁に確認することが必要である。PF MAKEでは、操作者が割り込みボタンを押すことで加工シーケンスを中断し、加工状態を確かめられるようになっている。

基礎部の加工が完了すると、PF MAKEは与えられたモデルの製作を始める。樹脂が固化してできる加工物のうち、モデルとして定義されている部分のことを、基礎部と区別するために「形状部」と呼ぶ。形状部を作製するには、各層の断面形状に沿って露光ビームを走査させ、樹脂を固化させて層を作ってから、一層の厚さだけエレベータを上昇させるシーケンスを層の枚数だけ繰り返す。形状部の加工では、基礎部を加工する時よりも露光スポットの直径を小さくし、また露光ビームの放射束を小さくして高い加工分解能を実現する。さらに加工物内部での露光条件を均一にして、後処理の際の変形を単純な相似変形となるようにする。

断面形状を描く走査アルゴリズムとしては、RP/M用光造形システムに多く用いられるベクトル走査 (vector scanning) の代わりに、ラスタ走査 (raster scanning) を採用した。この方法ではすべてのセルについて、対応する位置にスキャナが到着する直前の移動方向を描えることができるので、スキャナのガタ

(バックラッシュ)の影響を取り除くことができる。

ラスタ走査による露光では、加工時間がほぼ走査面の面積に比例する。そこで露光を行う前に加工物の断面形状の内部だけを走査するような走査経路を計画し、加工時間を短縮する。この際にスキヤナのバックラッシュの影響を取り除くため、図5-2.1に示すような走査経路を生成する。

またシャッタを開放したままスキヤナを移動させて露光を行う「連続露光」の代わりに、スキヤナのステッピング動作が終わってスキヤナが静止している間に、一定時間だけシャッタを開いて露光を行う「パルス露光」を採用した。この露光方法では、露光時間の精度がスキヤナのステッピング時間の精度によらないので、露光ビームの放射量を正確に制御することができる。

こうして一層の露光が終わると、基板が取り付けられたエレベータを上昇させ、加工物を窓板表面から引き離す。この時にエレベータの移動距離を十分に大きくすることで、加工物の弾性変形のために加工物が窓板に付着したままになるのを防ぐ。その後エレベータを元の位置よりも一層の厚さだけ上の位置まで下降させ、加工物の下に未固化の樹脂の層を作る。

エレベータを上昇させたり下降させたりする時には、樹脂液状物の持つ大きい粘度がトラブルの原因となることがある。エレベータが下降する時には、樹脂が基板と窓板のすき間に挟まれるので、絞り膜効果 (squeeze film effect) による大きい圧力が発生する。その結果、制御信号が送られてからエレベータが所定の位置に停止するまでに数秒の待ち時間が必要になる。この待ち時間の間に露光を再開して樹脂を固化させると、さらにエレベータが下降した時に固化した樹脂を窓板表面に押し付けることになり、窓板の表面処理層を傷める原因になる。

これと同じ理由から、エレベータを上昇させる速度が大きい場合には、窓板と基板のすき間に大きい負圧が生じる。その結果、すき間に急激に流れ込む樹脂のために加工物に大きい力がかかり、加工物の破損が起こる原因となる。そのため加工物を窓板から引き離す際には、エレベータの上昇速度をきわめて小さくすることが必要である。

以上がPFMAKEによる加工シーケンスである。このシーケンスが終わると加工物が固定されている基板を取り出し、超音波洗浄器を用いてアセトンで洗浄した後、ポストキュアを行って完全に固化させる。

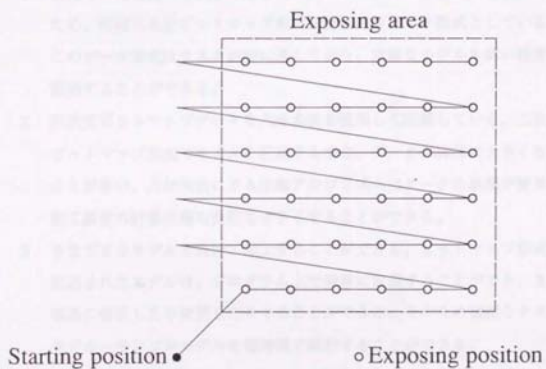


図5-2.1 PFMAKEの生成する走査経路

5-3. ソフトウェア系の設計

本研究では、光造形法によるマイクロマシン製作をサポートするソフトウェア体系（PFシステム）を作成し、テストモデル・サンプルモデルの設計や製作に使用した。PFシステムの全体的な構成を図5-3.1に示す。PFシステムを構成する個々のプログラムについて、TurboPASCALTM言語で書かれたソースリストを付録に示す。

このソフトウェア体系は、次のような特徴を持っている。

1. 固化セルの寸法・形状を考慮したモデルのローカルな補正を容易にするため、圧縮三次元ビットマップ形式を標準的なデータ形式としている。このデータ形式はラスタ走査に適しており、複雑なモデルを高い精度で記述することができる。
2. 三次元ビットマップデータを八分木法を使用して圧縮している。三次元ビットマップ形式でモデルを記述する場合、データが非常に大きくなることが多い。八分木法による圧縮アルゴリズムはデータの展開が簡単で、加工装置の計算処理の負荷を小さくすることができる。
3. さまざまなモデルを設計・加工することができる。ビットマップ形式で記述されたモデルは、プログラム上で簡単に定義することができ、また容易に修正したり変更したりすることができる。そのため複雑なテストモデル・サンプルモデルを短時間で設計することができる。
4. 最適な加工条件を実現するため、いろいろな加工条件を簡単に設定することができる。さらに一度設定した加工条件を記録したり、参照したりすることができるので、プログラムを起動するたびに毎回複雑な設定を行う必要がない。
5. 加工装置制御プログラムは無人運転を想定しており、加工シーケンスを自動的に生成して造形装置を制御する。加工条件を自動的に変えながら複数のテストモデルを製作することもできるので、短時間で最適な加工条件を見つけることができる。

PFシステムでは、モデルを記述する標準的なデータ形式として、圧縮三次元ビットマップ形式（PFデータ形式）を採用した。RP/M用光造形システムで標準となっているSTL形式は、ベクトル走査を前提とするデータ形式であり、

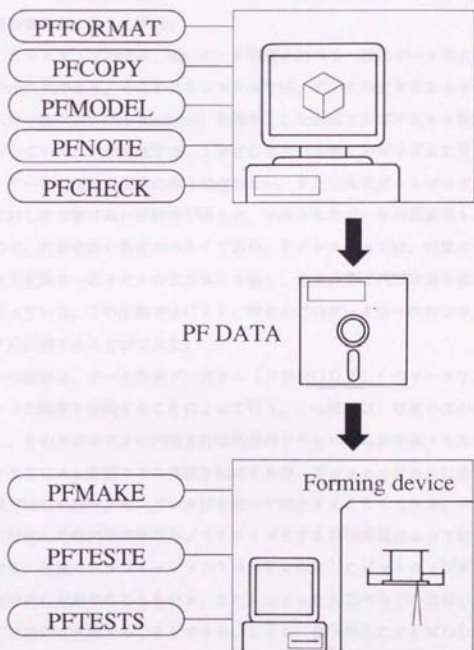


図5-3.1 PFシステムソフトウェアの構成

ラスタ走査には必ずしも適していない。ビットマップ形式では、加工装置に渡すデータの大きさは大きくなるが、加工装置での計算処理の負荷は小さくて済み、加工に要する時間を短縮することができる。さらに固化セルの寸法・形状を考慮したローカルな補正を行う場合にも、ビットマップ形式の方がSTL形式よりも計算処理の負荷が小さくて済む。

しかしビットマップ形式は、他のデータ形式と比べて一般にデータの大きさが大きいのが欠点である。そこでPFシステムでは、データの大きさを小さくするため、八分木法(oct-tree method)を基本とした圧縮アルゴリズムを使用して圧縮を行っている。八分木法では、JPEGなどの圧縮アルゴリズムに見られるような、データの圧縮・伸長に伴う劣化がない。また三次元ビットマップ形式のデータに対しては特に高い圧縮率が得られ、圧縮されたデータの再展開も比較的容易なので、計算処理の負荷が小さくて済む。PFシステムでは、対象モデルを含む三次元空間を一辺4セルの立方体に分割し、各立方体に八分木法を適用して圧縮を行っている。この圧縮方法により、ほとんどのデータは一枚のフロッピーディスクに記録することができた。

モデルの設計は、データ作成プログラム「PFMODEL」のソースリストの中で、一つの関数を定義することによって行う。この関数は、任意の点の座標を引数とし、その点がモデルの内部または外部のどちらにあるかを返すものである。通常はPASCAL言語でこの関数を記述するが、TurboPASCALTMの言語仕様ではC言語やアセンブラ言語を使って記述することもできる。モデルは代数的に記述した幾何学的形状をプリミティブとする形状演算によって定義することもでき、各種のグラフィックソフトウェアで作成したビットマップデータを用いて能率的に定義することもでき、またAutoCADTMなどの立体CADを利用して対話的に定義することもできる。こうして書き換えたPFMODELのソースリストをTurboPASCALTMコンパイラでコンパイルし、作られたプログラムを実行すると、目的のモデルを記述したPFデータが得られる。

PFデータは、特殊なフォーマットを行った一枚のフロッピーディスクに記録されている。これはディスクのアクセス時間を短縮するため、データの守秘性を高めるため、ディスク容量を大きくするための措置である。PFデータは正規のDOSファイルのフォーマットではないが、パーソナルコンピュータに付属する

基本ソフトウェアであるBIOSを使うことにより、PC-9801⁽¹⁾シリーズ各機種、およびIBM-PC⁽²⁾・互換機各機種で、読み出し・書き込みができる。

① PC-9801シリーズ

製品名が違っても、このソフトは、どの機種でも使えます。ただし、このソフトは、PC-9801シリーズの機種でしか使えません。

② IBM-PCシリーズ
このソフトは、IBM-PCシリーズの機種でも使えます。ただし、このソフトは、IBM-PCシリーズの機種でしか使えません。また、このソフトは、IBM-PCシリーズの機種でも使えません。

③ 互換機シリーズ
このソフトは、互換機シリーズの機種でも使えます。ただし、このソフトは、互換機シリーズの機種でしか使えません。また、このソフトは、互換機シリーズの機種でも使えません。また、このソフトは、互換機シリーズの機種でも使えません。

④ その他
このソフトは、その他シリーズの機種でも使えます。ただし、このソフトは、その他シリーズの機種でしか使えません。また、このソフトは、その他シリーズの機種でも使えません。

⑤ 注意事項
このソフトは、注意事項を守って使ってください。また、このソフトは、注意事項を守って使ってください。また、このソフトは、注意事項を守って使ってください。また、このソフトは、注意事項を守って使ってください。

⑥ 問い合わせ先
このソフトは、問い合わせ先を守って使ってください。また、このソフトは、問い合わせ先を守って使ってください。また、このソフトは、問い合わせ先を守って使ってください。

第6章 加工分解能を高めるための指針の検証

6-1. 固化セルの観察

第5章で述べた実験用加工システムを用いて実際に加工を行い、加工分解能を評価する実験を行った。

最初に、固化セルを実際につけて観察する実験（単セル露光実験）を行った。この実験の目的は、露光条件の違いによる固化セルの寸法・形状の変化を調べ、第4章で示した加工分解能を高めるための指針について、その妥当性を実験的に検証することである。

この実験では、あらかじめ造形装置のエレベータを窓板表面から十分に離し、露光ヘッドを決められた位置に固定しておく。この状態で樹脂表面に露光を行い、窓板表面に独立した固化セルを作る。この際に露光ビームの焦点位置と放射量を変え、露光条件の違いによる固化セルの変化を調べる。窓板には剥離性を大きくするための表面処理を施していないものを用い、作られた固化セルが窓板に固着するようにしておく。こうして露光を行った後に窓板を取り外して洗浄し、その表面に作られた固化セルを顕微鏡で観察する。

第4章では、加工分解能を高めるための指針の一つとして、「露光スポットの直径を安定化すること」が挙げられた。この指針を具体化するため、造形装置の集光レンズの焦点位置を樹脂表面に正しく合わせるのではなく、一定の焦点外し(defocusing)を持たせて露光を行う方法を考えた。

集光された露光ビームの焦点は、模式的には放射照度が無限大となる特異点である。よって焦点位置がわずかにでも動くことがあると、露光スポット内の放射照度は非常に大きく変化するので、固化セルの垂直方向の寸法を制御することができなくなり、垂直方向の加工分解能は低下することになる。これに対して露光ビームの焦点を樹脂表面から外し、露光スポットに有限の直径を持たせるようにすることで、集光レンズの焦点位置の変動の影響を受けにくくすることができる。これによって水平方向の加工分解能はやや低下するが、垂直方向の加工分解能は大きく向上するので、全体としては加工分解能を高めることができる。

露光スポットの直径を安定化する方法としては、集光レンズの焦点距離を長くすることも考えられる。しかしレーザー光のようなコヒーレンスの大きい光を焦点

距離の長いレンズで集光すると、光の回折による顕著な干渉じまが発生し、水平方向の加工分解能が著しく低下する。したがって集光レンズには、なるべく焦点距離の短いものを用いなければならない。

露光スポットの直径の調節は、窓板と集光レンズの距離を変えることで行う。実験用光造形システムでは、焦点距離 8 mm の集光レンズで、直径 $300 \mu\text{m}$ の露光ビームを集光している。よって窓板がないとした場合の仮想的な焦点位置と樹脂表面との距離を d とし、露光スポットの直径を D とすると、両者の間には式 6-1.1 のような関係が成り立つ。

$$D = 0.0375d$$

$$6-1.1$$

この実験では、 d の値を $0.0 \text{ mm} \cdot 0.8 \text{ mm} \cdot 1.6 \text{ mm}$ としたので、 D の値は $0 \mu\text{m} \cdot 30 \mu\text{m} \cdot 60 \mu\text{m}$ となる。この実験の加工条件を表 6-1.1 にまとめて示す。

こうして作られた固化セルを写真 6-1.1・写真 6-1.2 に示す。これらのうちで最大の固化セルを作る際の放射量は、1セル当たり約 $18 \mu\text{J}$ であった。

写真 6-1.1 から、集光条件の違いによって固化セルの寸法・形状が大きく変化することがわかる。 d が 0.0 mm の場合、固化セルの形状は非常に細長いものとなり、露光ビームの放射量を小さくしても、垂直方向と水平方向の寸法の比は大きいままである。さらに放射量を小さくすると、垂直方向の寸法が小さくなる前に固化セルは消滅する。一方 d が 1.6 mm の場合には、比較的等方的な形状を持つ固化セルが作られ、また放射量を小さくすると等方的な形状を保ったままで全体の寸法が小さくなる様子が観察される。固化セルの水平方向の寸法が露光スポットの直径よりも小さくなるのは、露光ビームの断面で露光光線の放射照度が分布しており、露光ビームの放射量が小さい場合には露光スポットの中心付近だけで樹脂の固化が起こるためと考えられる。

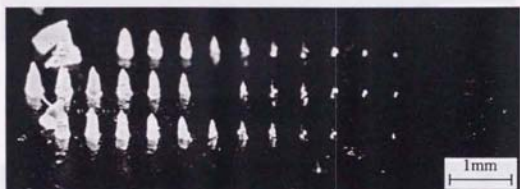
これらの固化セルの寸法・形状について、実際の加工を考慮しながら定量的に評価することを考えた。実際の加工では、最初にスキヤナの走査分解能を決めてモデルを設計する。そこで水平方向の走査分解能を決めておき、それぞれの集光条件で作られた固化セルの中から、水平方向の寸法がこの分解能に等しいものを選んで、その垂直方向の寸法を測定する。この値は垂直方向の加工分解能を示す

露光条件	
光源	He-Cdレーザー
露光スポットの直径	0 μm ・25 μm ・50 μm (目標値)
露光ビームの放射量	0 μJ ~18 μJ /セル
温度	30℃
走査条件	
X軸方向の走査分解能	-
Y軸方向の走査分解能	-
Z軸方向の走査分解能	-
樹脂	UV-854
洗浄方法	アセトン中で振盪

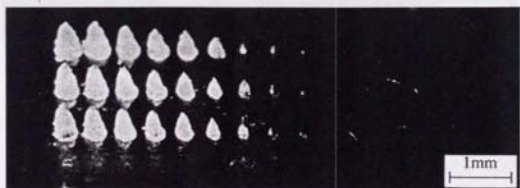
表6-1.1 単セル露光実験の加工条件



$d=0.0\text{mm}$



$d=0.8\text{mm}$



$d=1.6\text{mm}$

写真6-1.1 固化セルの作製例



$d=0.0\text{mm}$



$d=1.6\text{mm}$

写真6-1.2 固化セルの作製例(拡大)

ものであり、この値が小さいほど全体としての加工分解能は高いと考えられる。

そこで水平方向の走査分解能を $40 \mu\text{m}$ として、固化セルの垂直方向の寸法を測定した。すると d が 0.0 mm の場合の寸法は約 $500 \mu\text{m}$ であったのに対し、 1.6 mm の場合には約 $50 \mu\text{m}$ であり、 0.0 mm の場合よりも等方的な形状を持つ良好な固化セルが形成されることがわかった。

単セル露光実験の結果は第4章の理論検討で得られた結果とほぼ一致しており、この実験によって改善指針「露光スポットの直径を安定化すること」の妥当性を確かめることができた。

6-2. 加工分解能の測定

次に、テストモデルを作って加工分解能を評価する実験（分解能評価実験）を行った。この実験の目的は、光造形法を利用して実際に微細な構造を作製できることを確認すること、および最も高い加工分解能が得られる露光条件を把握して今後の実験に役立てることである。

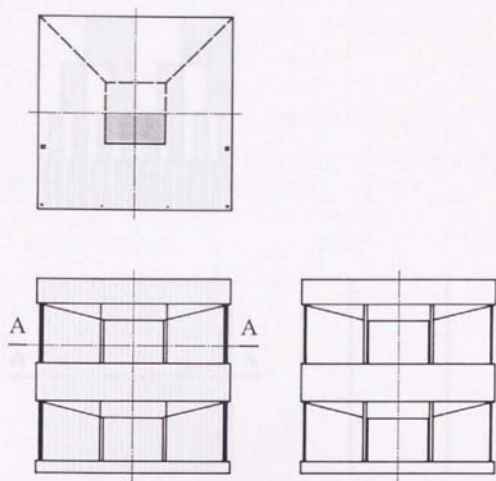
この実験に用いた2種類のテストモデル、「弦のモデル」・「くしのモデル」をそれぞれ図6-2.1・図6-2.2に示す。またこれらのモデルを作る際の加工条件を表6-2.1にまとめて示す。

「弦のモデル」は、最も高い加工分解能が得られる露光条件を見つけるためのテストモデルである。このモデルは、垂直に張られた細い糸状の部分（弦部）と水平に支持された板状の部分（屋根部）を持っており、これらの加工状態を観察することによって露光条件を評価することができる。

弦部は、水平方向の加工分解能を評価するために設けられている部分である。露光ビームの放射量を小さくしていくと弦部は細くなっていき、最終的には消滅する。そこで弦部が正しく作られる最小の放射量を求めれば、この値が最も高い加工分解能を得ることのできる放射量であると考えることができ、またその時の弦部の太さから、水平方向の加工分解能を推定することができる。

一方屋根部は、垂直方向の加工分解能を評価するために設けられている部分である。オーバハングになっている屋根部の下部には余剰固化物が生成するので、これを観察することによって垂直方向の加工分解能を推定することができる。

「弦のモデル」・「くしのモデル」を用いた実験では、一度に複数の露光条件に

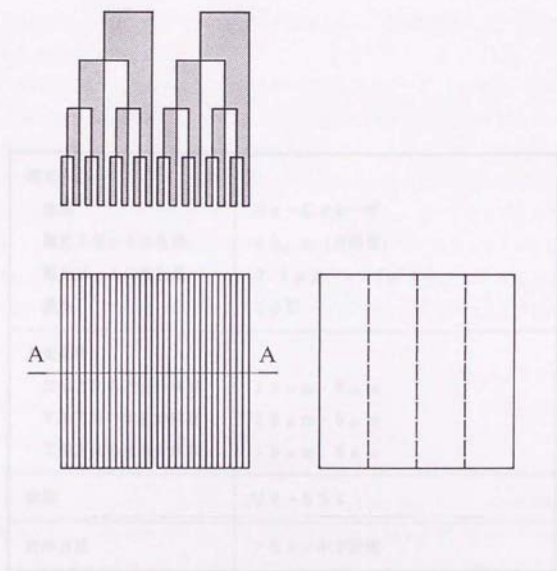


平面図/A-A断面図

立面図

側面図

図6-2.1 「弦のモデル」の構造



A-A断面図

立断面

側断面

図6-2.2 「くしのモデル」の構造

霧光条件	
光源	He-Cdレーザ
霧光スポットの直径	60 μm (目標値)
霧光ビームの放射量	0.1 μJ ~ 1 μJ /セル
温度	30 $^{\circ}\text{C}$
走査条件	
X軸方向の走査分解能	16 μm · 8 μm
Y軸方向の走査分解能	16 μm · 8 μm
Z軸方向の走査分解能	16 μm · 8 μm
樹脂	UV-854
洗浄方法	アセトン中で振盪

表6-2.1 分解能評価実験の加工条件

ついて評価を行うため、モデルの各部分ごとに露光ビームの放射量を変えながら加工を行った。

こうして作られたテストモデルの一例を写真6-2.1に示す。露光ビームの放射量が小さすぎた部分では、屋根部が破損したり、弦部が消失したりする加工不良が観察される。

またこれまでで得られたうちで最も細い弦部を写真6-2.2に示す。各層が4個の固化セルで構成されており、これらの層が積み重なって弦部を作っている様子が観察される。この弦部の代表径は約 $25\mu\text{m}$ であるが、個々の固化セルの水平方向の寸法は約 $16\mu\text{m}$ で、水平方向の加工分解能もこれに近い値であると考えられる。しかしこの分解能はスキャナの走査分解能に近いことを考慮すると、約 $16\mu\text{m}$ の加工分解能が光造形法の本質的な加工分解能であると考えられるよりも、むしろスキャナの走査分解能によって加工分解能が制限されていると考える方が妥当である。すなわち走査分解能を高めることにより、水平方向の加工分解能はさらに高めることができるものと考えられる。

しかしながら屋根部の下部には、写真6-2.3に示すようにつらら状の余剰固化物が観察され、その寸法は $200\mu\text{m}$ にも及ぶことがあった。余剰固化物は露光スポット内の放射照度に各セルごとのバラツキがあるためにできるもので、放射量を小さくすることによって本質的に解決することができない。すなわち放射量を小さくすれば、露光量の不足による加工不良が起こる。この問題に対し、露光スポットの直径を大きくして焦点位置の変動の影響を除き、露光時間を長くして露光ビームの光ノイズを平滑化すれば、余剰固化物の生成を抑制することができる。しかしこの方法では加工時間が非常に長くなり、根本的な解決方法にはならない。

「くしのモデル」は、水平方向の加工分解能を定量的に測定するためのテストモデルである。このモデルはくし形のテストパターンで、ピッチの異なる複数のパターンを持ち、その加工状態を観察することによって水平方向の加工分解能を正確に測定することができるようになっている。

作製されたテストモデルの一つを写真6-2.4に示す。この例ではピッチが $32\mu\text{m}$ のくし形のパターンが作られており、これから水平方向の加工分解能が $16\mu\text{m}$ 以下であることがわかる。

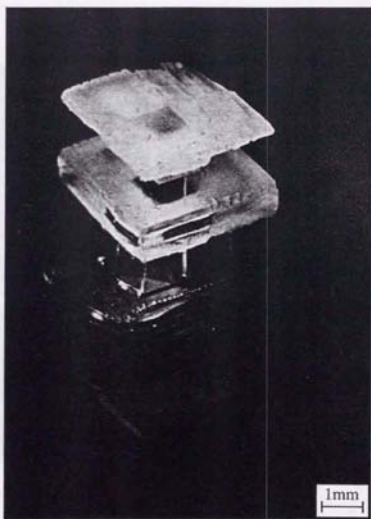


写真6-2.1 「弦のモデル」の作製例

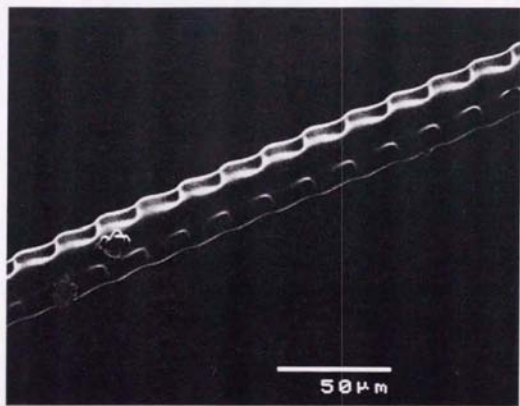


写真6-2.2 最小の代表径を持つ弦部

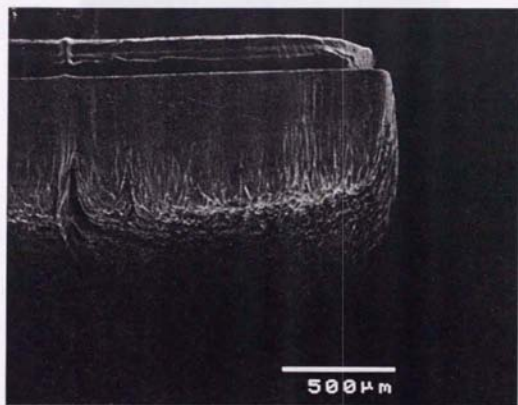


写真6-2.3 余剰固化物の生成

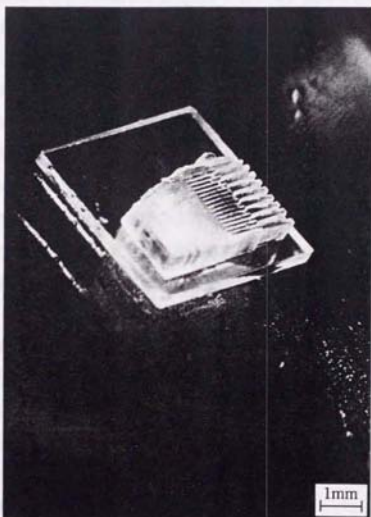


写真6-2.4 「くしのモデル」の作製例

「くしのモデル」を用いた実験では、水平方向の加工分解能だけしか評価することができない。そこで最初に「弦のモデル」を用いて、水平方向と垂直方向の加工分解能が等しくなるような露光条件を求めておき、次に「くしのモデル」を用いて水平方向の加工分解能を測定すれば、全方向の加工分解能を求めることができる。こうして測定した加工分解能は約 $30 \mu\text{m}$ であった。

このように露光スポットの直径を安定化することで、従来の RP/M 用光造形システムよりも高い加工分解能を実現することができ、第 4 章で示した改善指針「露光スポットの直径を安定化すること」の妥当性を確かめることができた。

この二つのテストモデルを加工するには、それぞれ 20 時間以上の加工時間を要した。これは光源の光出力が小さすぎることによるものであり、より光出力の大きい光源を用いることで短縮できると考えられる。

以上のように第 4 章で示した改善指針に基づいて加工条件を見直すことにより、水平方向だけでは約 $10 \mu\text{m}$ にまで、全方向では約 $30 \mu\text{m}$ にまで加工分解能を高めることができた。これよりも高い加工分解能を実現するには、光源と樹脂を見直す必要がある。これについては 6-3 節で詳述する。

6-3. 可視光による露光

さらに、可視光光源・可視光硬化性樹脂を使ってテストモデルを製作する実験（可視光露光実験）を行った。この実験の目的は、第 4 章で示した加工分解能を高めるための三つの指針をすべて具体化することで、きわめて高い加工分解能を実現して指針の妥当性を検証するとともに、光造形法を利用したマイクロマシン製作法の意義を示すことである。

第 4 章では、加工分解能を高めるための指針として、「露光ビームの放射量を安定化すること」・「露光スポットの直径を安定化すること」・「樹脂の吸収係数を大きくすること」の三つが挙げられた。このうち「露光スポットの直径を安定化すること」については、単セル露光実験・分解能評価実験により、その妥当性を検証することができた。そこで残された二つの改善指針の妥当性を検証するため、従来の光源・樹脂の代わりに、光ノイズの小さい光源と吸収係数の大きい樹脂を使用して実験を行うことを考えた。

多くの RP/M 用光造形システムの光源には、He-Cd レーザが用いられて

いる。しかしHe-Cdレーザの光ノイズは10%ppにも及び、また熱的な制御に起因する低周波数成分が多く含まれている。露光ビームの放射量を正確に制御することを考えると、これは非常に不利な特性である。これに対してAr⁺レーザ・高圧水銀燈・水銀アーク燈の光ノイズは1%以下であり、きわめて高い加工分解能を実現するには、これらの光源を使うことが望ましいと考えられる。

一方樹脂の吸収係数を大きくする方法としては、樹脂に光吸収材料を添加することが考えられる。しかし紫外光領域の光に対する物質の吸収係数は、主としてその物質を構成する原子の電子状態に依存している。そのため光吸収材料として有機物質を加えても、大きな改善を期待することは難しい。紫外光に対する吸収係数を大きくするには、電子が広い範囲のエネルギー準位を持つことのできる分子構造を持っている物質を光吸収材料として加える必要がある。

このような光吸収材料としては、自由電子 (free electron) や π 電子 (pi-electron) など、エネルギー準位の遷移が起こりやすい電子を分子内に含んでいる金属・グラファイト・半導体・芳香族化合物・共役ポリエン化合物などの物質が考えられる。しかし無機物である金属・グラファイト・半導体は樹脂にほとんど溶解せず、微粉末として樹脂中に分散させることも難しい。また芳香族化合物・共役ポリエン化合物には揮発性の物質が多く、これらを添加した樹脂は加工中にその組成が変化する恐れがある。また反応性の大きい物質も多く、樹脂に多量に添加すると光化学反応を阻害する恐れもある。このように紫外光に対する樹脂の吸収係数を大きくすることは、一般にかなり困難である。

しかし可視光を放射するAr⁺レーザを用いる場合には、樹脂にある種の有機物質を加えることで、容易に大きな吸収係数を持たせることができる。可視光に対する物質の吸収係数は、主としてその物質を構成する分子の原子間結合の振動モードに関係しており、電子の状態には大きく依存していない。よって紫外光を用いる場合のように特殊な光吸収材料を使う必要はなく、最適な材料を選択することができる。光吸収材料として染料を使えば、特定の波長の光に対して大きい吸収係数を与えることもできる。

そこで可視光露光実験では青色の光を放射するAr⁺レーザを用い、樹脂には青色の補色に当たる黄色の有機染料を加えて、青色光に対する樹脂の吸収係数を大きくした。通常の光硬化性樹脂は可視光で硬化しないように作られているので、

可視光で固化する光硬化性接着剤を混合して可視光硬化性樹脂を作り、光硬化性接着剤に含まれている光反応開始剤の作用によって樹脂全体を固化させることを考えた。この実験に使用した樹脂の配合を表6-3.1に示す。

こうしてA rレーザと調合した可視光硬化性樹脂を使用し、テストモデルを作製した。可視光露光実験は7-2節で述べる自動機構製作実験を兼ねて行った実験であり、7-2節で述べる「マイクログリッパのモデル」をテストモデルとして用いた。このモデルを作る際の加工条件を表6-3.2にまとめて示す。

こうして作られたテストモデル全体を写真6-3.1に、またテストモデルの一部を拡大したものを写真6-3.2・写真6-3.3に示す。オーバハング部の下部に生成する余剰固化物の寸法が非常に小さくなり、垂直方向の加工分解能が著しく向上したことがわかる。余剰固化物の寸法を測定した結果、垂直方向には $10\mu\text{m}$ 以下の加工分解能が得られることがわかり、またモデルの稜線の曲率を測定した結果、水平方向には約 $2\mu\text{m}$ の加工分解能が得られることがわかった。全方向では $5\mu\text{m}\sim 10\mu\text{m}$ の加工分解能が得られた。

この実験により、第4章で示した改善指針「露光ビームの放射量を安定化すること」・「樹脂の吸収係数を大きくすること」の妥当性を確かめることができた。また既存の微細加工法の持つ加工分解能に近い加工分解能を実現することができ、マイクロマシン製法法としての光造形法の意義を示すことができた。

このテストモデルを加工するには100時間以上の加工時間を要した。これは光源の光出力の不足によるものではなく、窓板の表面処理層や樹脂の温度上昇を避けるために露光ビームの放射束を小さくしていること、および可視光によって光固化反応を開始させる光反応開始剤の光感度が小さいことによる。したがって光源の光出力を大きくするだけでは、加工時間を短縮することができない。この問題に対しては、さらに光感度の大きい樹脂を使用する、あるいは一括露光式を採用するなど、樹脂や露光方法を見直す必要がある。

6-4. 結果と考察

以上のように、第4章で示した改善指針に基づいて加工条件を見直して実験を行った結果、加工分解能が大幅に高められることを確認し、指針の妥当性を検証することができた。

ディーメック SCR-500	4.9%w
東亜合成化学工業 VL-2302	4.9%w
藍熊染料 メタノール フラスト エロー	2%w

表6-3.1 可視光硬化性樹脂の配合

露光条件	
光源	Ar+レーザー
露光スポットの直径	40 μm (目標値)
露光ビームの放射量	0.5 μJ ~ 5 μJ /セル
温度	30 $^{\circ}\text{C}$
走査条件	
X軸方向の走査分解能	8 μm
Y軸方向の走査分解能	8 μm
Z軸方向の走査分解能	8 μm
樹脂	可視光硬化性樹脂 (表6-3.1参照)
洗浄方法	アセトン中で振盪

表6-3.2 可視光露光実験の加工条件



写真6-3.1 「マイクログリッパのモデル」の作製例

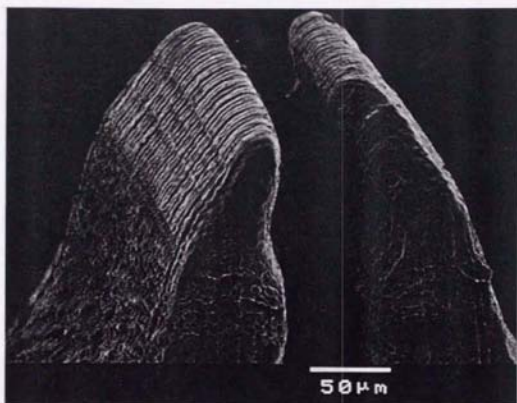


写真6-3.2 マイクログリッパのアーム部先端の拡大

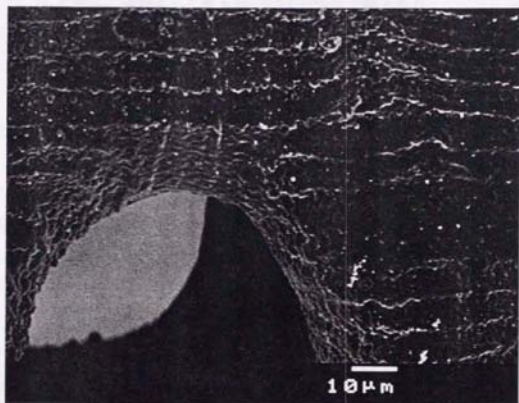


写真6-3.3 マイクログリッパのオーバハング部の拡大

これらの実験を行うことで得られた知見を次に述べる。

1. 第4章で示した三つの改善指針を具体化することによって加工分解能を大幅に高めることができ、水平方向と垂直方向で同時に $10\mu\text{m}$ 以下の分解能を実現することができる。
2. 露光ビームに一定の焦点外しを与え、露光スポットの直径を一定の値に保つことで、良好な形状・寸法を持つ固化セルが作られる。スキヤナの走査分解能が $10\mu\text{m}$ の場合には、露光スポットの直径を約 $60\mu\text{m}$ とした時に、全体として最も高い加工分解能が得られた。
3. 加工分解能を高めるために樹脂の吸収係数を大きくすると、樹脂に照射される露光光線のエネルギーの多くが熱として消費されるようになるので、エネルギー効率は低下する。また発生した熱は、樹脂の焼損や変質などの原因となる。このトラブルを避けるには、露光ビームの放射束を小さくする方法も考えられるが、加工時間は長くなる。

こうして得られた知見を踏まえ、光造形法を利用したマイクロマシン製作法を実用化するための指針について、第8章で議論することにする。

第7章 可動部を持つ微細な機構の作製

7-1. 手動機構の作製

光造形法を利用してマイクロマシンを作ることを考えると、単に複雑な形状を持つ加工物を作るだけでなく、力や動きを伝達することのできる機械構造を実現しなければならない。そこで本研究では実際に動かすことのできる機構を試作し、その機械的特性を評価して、可動部を持つ機械構造を設計する上での指針を得ることを考えた。

最初に、手動で動かすことのできる機構を作る実験（機構作製実験）を行った。この実験の目的は、樹脂固形物が機械構造を作る材料として十分な機械的特性を持っていることを確認し、また機械構造の代表寸法が小さくなることによる寸法効果の影響を調べて、光造形法によって実際に動く機構を作製できることを実証することである。

動く機械構造を設計する際には、可動部を実現する方法が大きな問題となる。ステレオリソグラフィ法では、一度の加工で原理的に一つの部材しか作ることができないので、複数の部材から構成される複雑な機構を作ることは一般に難しい。また作製できる加工物形状には制約があるので、ある種の機構を作製する際には3-3節で述べたサポートを挿入しなければならない。この場合にはサポートを挿入する位置や方向について、各部材の動きを考慮した特別な検討が必要となる。

さらにマイクロマシンを設計する際には、2-2節で述べた寸法効果や、加工精度の影響についても考慮する必要がある。機械構造の代表寸法が小さくなると機械構造を構成する部材は、寸法効果のために剛性があたかも小さくなったかのような挙動を示す。このことを積極的に利用すると、部材の弾性変形を応用した可動部を実現することができるが、一方で変形させたくない固定部については、剛性を大きくするための対策が特に必要となることもある。また慣性力に対する摩擦の影響が無視できなくなり、加工物の表面粗さが代表寸法に対して大きくなるので、慣性を応用した機構原理や、部材と部材の接触や滑動を応用した機構原理は、一般には利用できなくなる。

以上のことを考慮しながら、外部からの手動操作によって動かすことのできるいくつかの機構をテストモデルとして設計し、実験用加工システムを用いて作製

した。次のテストモデルを作る際の加工条件を表7-1.1にまとめて示す。

「てこのモデル」(構造：図7-1.1/作製例：写真7-1.1)

接続ロッドで結合された3本のてこが運動するように設計された機構。

A部を下方に押すとB部が上方に動く。てこの支点には薄板の曲がりを利用し、摩擦が生じないようにてこを支持している。

「タービンのモデル」(構造：図7-1.2/作製例：写真7-1.2)

「竹とんぼのモデル」(構造：図7-1.3/作製例：写真7-1.3)

ビボット軸受で支えられた回転子が回転するように設計された機構。

軸の上下に2個の軸受を持ち、加工後に軸受中心のサポートをねじ切ることで、回転子が自由に回転できるようにする。

このようにして作られたテストモデルを、針などを用いて手で動かしてみたところ、ほぼ設計したとおりに動作することが観察できた。

「てこのモデル」では、A部を下方に1mm押すと、B部が上方に約0.5mm動いた。てこ部を支持する支点部は、てこ部を約30度傾斜させても破損せず、この傾斜の動作を10回以上繰り返すことができた。接続ロッド部は、てこ部を約10度傾斜させても破損したり座屈したりすることはなかった。しかし10度以上傾斜させると、張力が加わる方の接続ロッド部が破断することがあった。

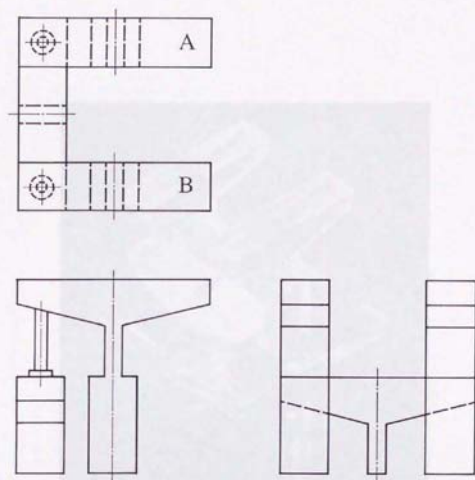
「タービンのモデル」・「竹とんぼのモデル」では、加工後に回転子部を回して軸受中心のサポートをねじ切る。この時には強い力が必要だが、一度サポートが切られると回転子部は自由に回転するようになり、回転子部を回すモーメントをかけると、軸受から外れるまでは無限に回転させることができた。

この実験により、樹脂固形物が機械構造を作る材料として十分な機械的特性を持っていることを実証し、既存の微細加工法と同様、光造形法を利用することで微細な機械構造が作製できることを確認することができた。

これらのテストモデルを動かすと、構造部材の剛性の不足が問題となりやすいことがわかる。これは寸法効果に関する次元解析から理論的に示されることでもあるが、ヤング率の小さい樹脂で作られた機械構造では特に顕著な現象となって観察される。たとえば「てこのモデル」ではてこ部が湾曲して変位を正しく伝達しなくなり、また「タービンのモデル」・「竹とんぼのモデル」ではフレーム部が変形して回転子が軸受から外れたりする。モデルの設計段階では、部材の剛性に

露光条件	
光源	He-Cdレーザー
露光スポットの直径	60 μm (目標値)
露光ビームの放射量	0.1 μJ ~ 1 μJ /セル
温度	30 $^{\circ}\text{C}$
走査条件	
X軸方向の走査分解能	16 μm \cdot 8 μm
Y軸方向の走査分解能	16 μm \cdot 8 μm
Z軸方向の走査分解能	16 μm \cdot 8 μm
樹脂	UV-854
洗浄方法	アセトン中で振盪

表7-1.1 機構作製実験の加工条件



平面図

立面図

側面図

図7-1.1 「てこのモデル」の構造

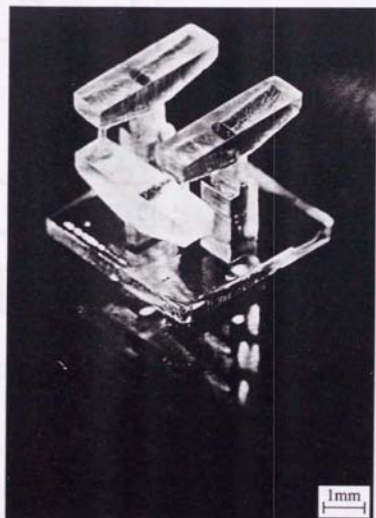
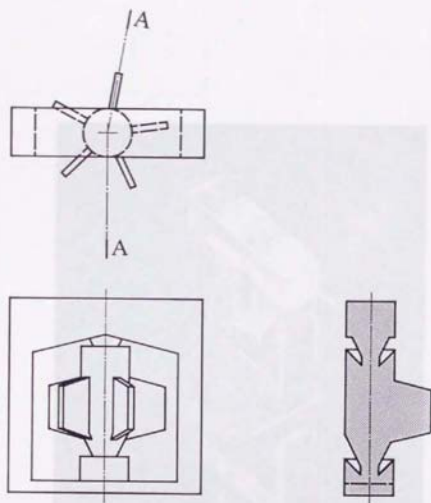


写真7-1.1 「てこのモデル」の作製例



平面図

立面図

A-A断面図

図7-1.2 「タービンのモデル」の構造

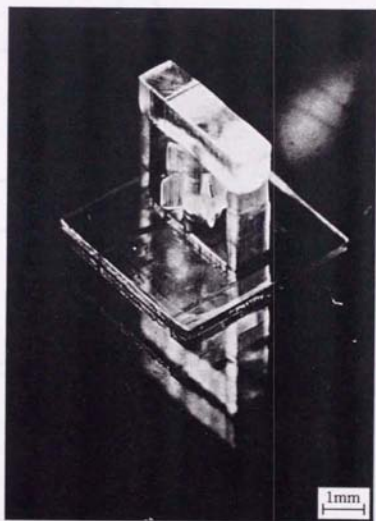
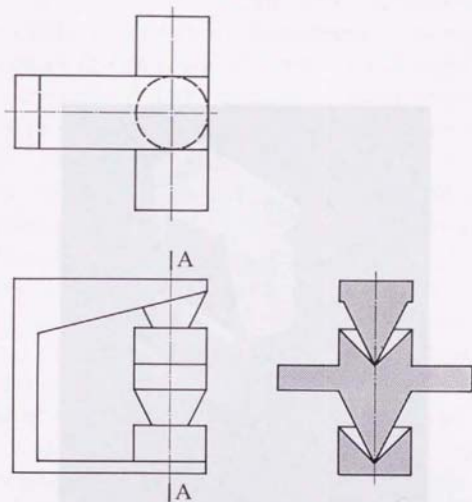


写真7-1.2 「タービンのモデル」の作製例



平面図

立面図

A-A断面図

図7-1.3 「竹とんぼのモデル」の構造

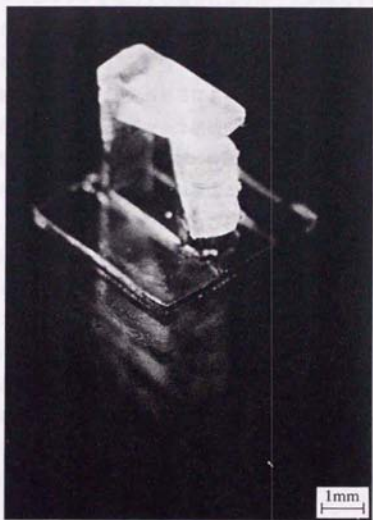


写真7-1.3 「竹とんぼのモデル」の作製例

ついてFEMなどの手法による解析を行い、変形を定量的に評価しておくことが重要である。

これらのテストモデルには、それぞれ積層することのできる方向があり、他の方向からは作製することができない。これは光造形法で作ることのできる加工物形状に制約があるためである。RP/Mではサポートを挿入して対処することができるが、マイクロマシンを製作する場合にはサポートを挿入することが難しく、積層方向が大きな問題となる。これについても、モデルの設計段階で十分に検討しておくことが重要である。

このような制約を考えると、光造形法によって作られる微細な機構は、通常の機械に用いられている機構とは異なった構造のものになると考えられる。微細な機械構造を設計する際には、材料の機械的特性を積極的に利用していかなければならない。したがって樹脂固形物が機械構造として使用できる材料であることを検証するには、その機械的特性を評価することが重要であり、特に破壊について調べることが必要である。そこでテストモデルを繰り返し変形させ、構造部材の変化を観察する材料試験を行った。

実験用加工システムを用いて作製した試験片は、その一部が長さ約0.4mm、幅約0.5mm、厚さ約0.2mmの長方形平板となっているもので、この長方形平板部に繰り返し一定の曲げ変形を生じさせるような変位を試験片端部に与える。この試験に使用した試験片の形状・寸法、および試験方法を図7-1.4に示す。変位の繰り返し周波数は、試験片の共振周波数に対して十分に小さくなるように10Hz以下に設定した。また変位の大きさは、試験片の破壊位置における最大ひずみが2%・4%・6%となるように設定した。

この試験の結果を表7-1.2に示す。この試験結果から、繰り返し動作するマイクロマシンを作る場合、構造部材の最大ひずみは2%まで許容されることがわかった。この最大ひずみは、従来の微細加工法で用いられるケイ素単結晶や金属など、では利用することが難しい値であり、材料のコンプライアンスを利用した動作原理を応用することを考えると、光造形法を利用することが特に有利であることがわかる。

このように光造形法を利用したマイクロマシン製作法を用いることで、多くの種類の機構を実現することができる。しかし通常の機械とマイクロマシンでは、

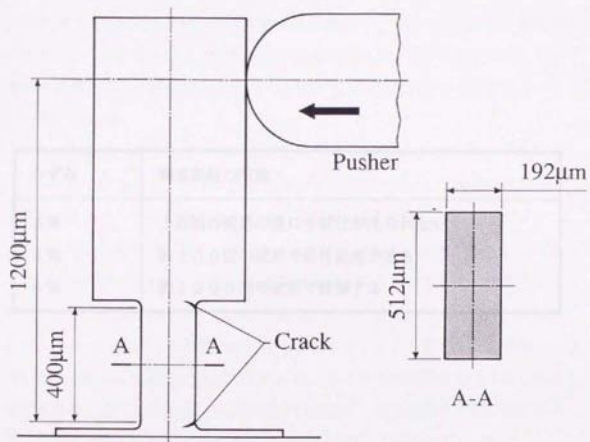


図7-1.4 材料試験の方法

それぞれに用いられる機構の動作原理が異なったものになると考えられる。寸法効果や加工の仕様を考慮しながら、利用できる機構原理を模索してライブラリ化していくことは、今後の重要な課題の一つである。

7-2. 自動機構の作製

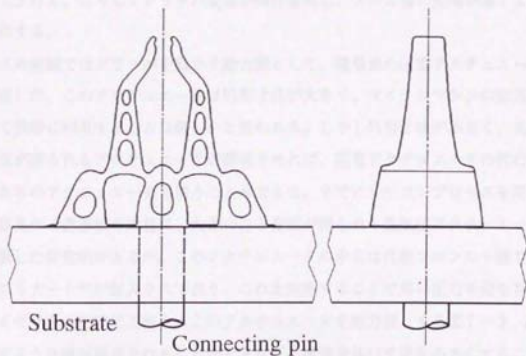
次に、基板の下に置かれた圧電アクチュエータに電圧を印加することによって動作する自動機構を作る実験（自動機構作製実験）を行った。この実験の目的は、光造形法で作られた機械構造に力や動きを供給する方法の一つとして、基板内にアクチュエータを組み込んで機械内部の目的の部分まで力や動きを伝達していく方法を提案し、この駆動方法を実現する手段として光造形法を利用できることを示すことである。

この実験に用いたテストモデルを図7-2.1に示す。このモデル「マイクログリッパのモデル」は、小さい物体を操作するグリッパとして設計されており、圧電アクチュエータなどのアクチュエータを用いて駆動することができる。

このモデルを作る際には、いくつかの小穴が明けられた基板を使う。小穴にはグリッパを動かすための「グリッパ操作用」のもと、加工原点の位置合わせを行うための「位置合わせ用」のものがある。小穴の直径はどちらも0.5mmである。グリッパ操作用の小穴には、あらかじめ短いピアノ線のできた接続ピンをさし込んでおく。位置合わせ用の小穴には、あらかじめ透明な樹脂を流し込んで固めておき、小穴に露光ビームが照射された時に、露光光線の一部が基板の上に出てくるようにしておく。その後基板表面を研磨して平滑にする。

基板を加工装置に取り付けた後、グリッパ操作用の小穴の上にグリッパ本体が作られるように、加工原点の位置を合わせる。この時には、エレベータを窓のわずかに上の位置に置き、非常に小さい放射束の露光ビームを照射しながら露光ヘッドを動かして、位置合わせ用の小穴を通して基板の上に露光光線が出てくる位置を見つける。このようにして位置合わせ用の小穴の座標を確認し、これからグリッパ操作用の小穴の座標・加工原点の座標を算出して、原点を指定する。

このようにしてグリッパ本体を作製した後、グリッパの動きを観察するための評価用のホルダに基板を取り付ける。基板の下には圧電アクチュエータを置き、接続ピンとアクチュエータをつないで、アクチュエータの動きがグリッパ本体に



立断面図

側断面図

図7-2.1 マイクログリッパの構造

伝わるようにする。

この圧電アクチュエータは、約140Vの電圧を印加することで、垂直方向に約20 μ m伸長する。基板の下に置かれている圧電アクチュエータに電圧を印加すると、接続ピンが上に押し上げられ、基礎部を貫通してグリッパ本体の一部を押し上げる。こうしてグリッパ全体が弾性変形し、アーム部の先端が開くように動作する。

この実験ではグリッパを動かす動力源として、積層型の圧電アクチュエータを使用した。このアクチュエータは外形寸法が大きく、マイクロマシンの動力源として実際に利用することは難しいと思われる。しかし外形寸法が小さく、大きい変位が得られるアクチュエータが開発されれば、圧電アクチュエータの代わりにそれらのアクチュエータを使うこともできる。すでにシリコンプロセスを用いて、単結晶ケイ素基板の内部に、大きい力と変位が得られる蒸気圧アクチュエータを作製した研究例がある⁽¹⁾。このアクチュエータの中には代替フロン⁽²⁾の一種であるフロリナート⁽³⁾が封入されており、これを加熱することで高い圧力を発生させてダイヤフラムを変位させる。このアクチュエータを動力源とする図7-2.2に示すような機械構造を作ることができれば、構造全体の寸法を小さくすることができると思われる。

グリッパの最適な形状は対象物によって変わるが、光造形法は加工の自由度が高く、形状を簡単に変更することができる。また、この構造では小さい対象物をつかむことも難しいが、これについてもグリッパの形状を変えることで解決することができる。このようにそれぞれの用途に応じて製品の仕様変更が求められるマイクログリッパは、光造形法の特徴を生かした代表的な応用例である。

この実験では光源と樹脂を見直し、グリッパの加工精度を高めることを考えた。自動機構作製実験は6-3節で述べた可視光露光実験を兼ねて行った実験なので、6-3節では「マイクログリッパのモデル」を用いて加工分解能の評価を行った。このモデルを作る際の加工条件は表6-3.2にまとめて示した。また作られたテストモデルは写真6-3.1に示した。

このグリッパの動作を調べるため、接続ピンを一定の変位で繰り返し押し上げ、その時のグリッパの動きを顕微鏡で観察した。その結果、接続ピンを約30 μ m押し上げた時に、アーム部の先端が約60 μ m開くことがわかった。この様子を

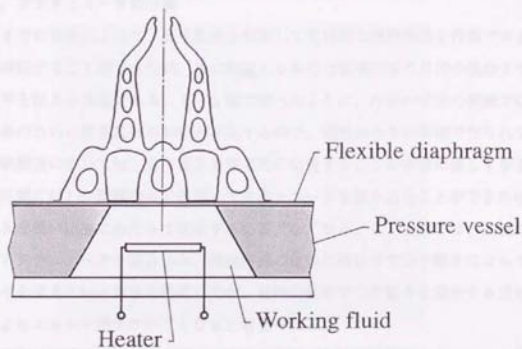


図7-2.2 蒸気圧アクチュエータによるマイクログリッパの駆動

写真7-2.1に示す。この実験により、基板内にアクチュエータを組み込んで機械内部の目的の部分まで力や動きを伝達していく方法について、その可能性を確かめることができ、この駆動方法を実現することのできる微細加工法としての光造形法の有効性を示すことができた。

しかしアクチュエータの変位に対するアーム部の先端の変位は、当初設計した値の約30%にとどまった。これは構造部材の剛性が不足していること、および接続ピンとグリッパ本体の間にすき間ができることによると考えられる。

7-3. アクチュエータの作製

これまでの実験によって、光造形法を利用して立体的な機械構造を作製できることを確認することができたが、次に問題となるのは機械内部の目的の部分までエネルギーを伝える方法である。7-1節で述べたように、小さい寸法の機械では寸法効果のために構造部材の剛性が不足するので、剛性の小さい樹脂で作られている機械構造においては、力や動きを効率的に伝達することが非常に難しくなる。

この問題に対し、機械構造の内部にアクチュエータを組み込むことができれば、力や動きを長い距離にわたって伝達する必要がなくなる。7-2節では、基板の内部にアクチュエータを組み込み、機械内部の目的の部分まで力や動きによってエネルギーを伝えていく方法を提案したが、目的の部分で力や動きを発生する方が伝達によるエネルギー損失が小さくとなると考えられる。

そこで機械構造の内部に組み込んで使うことのできる流体圧アクチュエータを考案し、これを試作する実験（アクチュエータ作製実験）を行った。この実験の目的は、機械構造の内部に組み込むことのできるアクチュエータを提案し、このアクチュエータを作る手段として光造形法を利用できることを示すことである。

流体圧アクチュエータの動作原理を図7-3.1に示す。これは剛性の小さい一種の圧力容器で、内部に圧力をかけることで弾性変形が起こるように作られているものである。このアクチュエータを動作させるには、基板に明けられた小穴（ポート）を通してアクチュエータ内部に高圧の液体を送り込んだり、あるいはアクチュエータ内部に封入されている揮発性の液体を加熱し、高い蒸気圧を発生させたりすることで、アクチュエータを変形させる。

このアクチュエータの特徴としては、寸法・構造を自由に設計することができ、

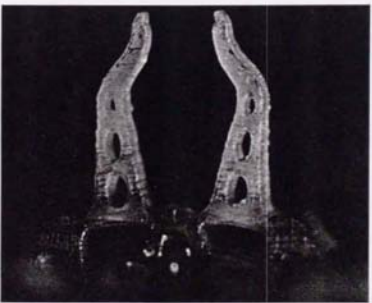
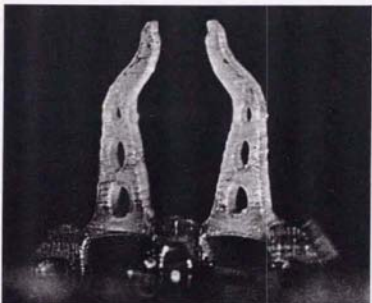
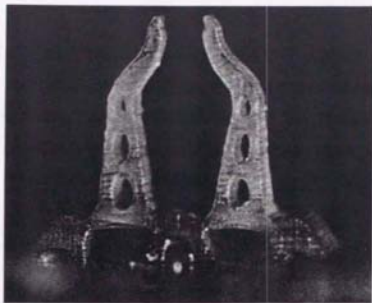


写真7-2.1 マイクログリッパの動作

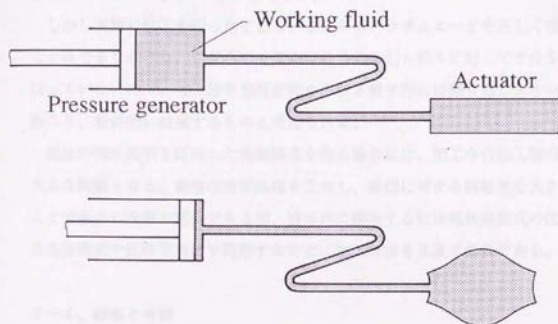


図7-3.1 流体圧アクチュエータの動作原理

さまざまな動作を実現できること、構造がきわめて簡単であり、高い加工精度を必要としないので各種の加工法で作製できること、光造形法を利用すれば容易に機械構造の内部に組み込めること、動作原理が簡単なので動作特性を簡単に評価できることなどが挙げられる。

アクチュエータ作製実験では、動作の異なる4種類の流体圧アクチュエータを設計した。これらは圧力をかけると、それぞれ中心軸方向の収縮・中心軸方向の伸長・中心軸に対する屈曲・中心軸回りの回転を行う。これらのテストモデルのうち、中心軸方向に収縮するものの構造を図7-3.2に示す。またこの実験の加工条件を表7-3.1にまとめて示す。

しかし実際に加工を行ったところ、これらのアクチュエータを正しく作製することはできなかった。これらのモデルは軸方向の引っ張りに対して十分な剛性を持っていないので、加工物を窓板表面から引き離す時には繰り返し大きい変形が起り、最終的に破損するものと考えられる。

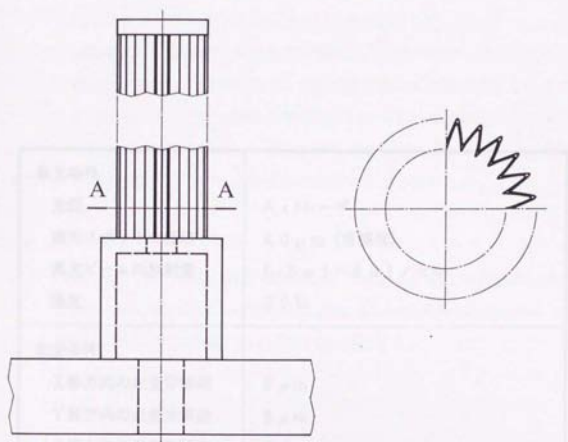
部材の弾性変形を応用した機械構造を作る場合には、加工中の加工物の変形が大きな問題となる。窓板の表面処理を工夫し、樹脂に対する剥離性を大きくすることで多少の改善が期待できるが、根本的に解決するには規制液面式の代わりに自由液面式や固体下地法を採用するなど、加工方法を見直す必要がある。

7-4. 結果と考察

以上のように、各種の動作原理で動く微細な機構を試作する実験を行った結果、光造形法で作られた構造が力や動きを伝える機械構造として機能することを確認することができた。

これらの実験を行うことで得られた知見を次に述べる。

1. 樹脂成形物を材料として、実際に動かすことのできる微細な機構を作製することができる。
2. 微細な機構を駆動するため、基板内にアクチュエータを組み込んで機械内部の目的の部分まで力や動きを伝達していく方法が利用できる。
3. 光造形法で作製できる加工物形状の制約は、サポートを挿入することの難しい機械構造を設計する際に大きな問題となる。より大きな剥離性を持つ窓板表面の処理方法を確立したり、自由液面式や固体下地法を採用



立视图

A-A断面图 (2:1)

图 7-3.2 流体压アクチュエータの構造

露光条件	
光源	Ar+レーザー
露光スポットの直径	40 μm (目標値)
露光ビームの放射量	0.5 μJ ~ 5 μJ /セル
温度	30 $^{\circ}\text{C}$
走査条件	
X軸方向の走査分解能	8 μm
Y軸方向の走査分解能	8 μm
Z軸方向の走査分解能	8 μm
樹脂	可視光硬化性樹脂 (表6-3.1参照)
洗浄方法	アセトン中で振盪

表7-3.1 アクチュエータ作製実験の加工条件

したりするなどの対処が必要である。

4. 規制液面式では、構造部材の弾性変形を利用した機構を作製することが難しい。自由液面式や固体下地法を採用し、加工物を窓板表面から引き離す工程をなくすなどの対処が必要である。
5. 光造形法によって作られる微細な機構は、通常の機械に用いられている機構とは異なった構造を持つ。寸法効果や加工の仕様を考慮しながら、利用することのできる機構原理を模索していくことが必要である。

こうして得られた知見を踏まえ、光造形法を利用したマイクロマシン製作法を実用化するための指針について、第8章で議論することにする。

文献

- (1) Mizoguchi, H., Ando, M., Mizuno, T., Takagi, T. and Nakajima, N., Design and Fabrication of Light Driven Micropump, Conference Proceedings of MEMS '92 (1992), pp 31-36.

第8章 考察

8-1. 光造形法を利用したマイクロマシン生産についての考察

本研究では光造形法を利用したマイクロマシン製作法について、以上のように研究を進めてきた。その結果、光造形法を微細加工法の一つとして用いる場合に必要になると考えられる加工法の特徴、すなわち「光造形法を利用したマイクロマシン製作法の仕様」を確認することができた。本研究によって明らかになった二つの仕様を次に示す。

- ・ $10\ \mu\text{m}$ 以下の高い加工分解能を実現することができる

第4章で示した加工分解能を高めるための指針に基づいて露光条件を見直すことで、水平方向と垂直方向で同時に約 $30\ \mu\text{m}$ の加工分解能を実現することができた。さらに露光条件・光源・樹脂を見直すことで、 $10\ \mu\text{m}$ 以下の加工分解能を実現することができた。

- ・ 1万回以上の動作に耐える機械構造を作ることができる

光造形法で作られた微細な機械構造を駆動する方法を示し、基板内に圧電アクチュエータを組み込んでおく方法で、実際に機構を動作させることができた。また構造部材を繰り返し変形させる試験を行ったところ、最大ひずみが2%以下の場合には、1万回の変形の後にも部材に変化が見られなかった。

以上のような光造形法の仕様は、既存の微細加工法の仕様の範囲にほぼ含まれ、光造形法が実用的なマイクロマシン製作法の一つとなり得ることを示している。そこで光造形法を利用したマイクロマシン製作法を新たに導入することによってマイクロマシン製品の生産にどのような効果をもたらされるか、またこの技術を実用化するためには今後どのようなシーズが必要となるかについて、ここで考察することにした。

第2章では、現状でのマイクロマシン製品の開発過程について、その問題点を述べた。既存の微細加工法を用いたマイクロマシン製品の開発においては、設計段階で製作段階における制約を考えなければならないことが特に大きな問題点となっている。すなわち設計者は既存のあらゆる微細加工法に精通し、それぞれの加工法の仕様を把握していなければならない。そして対象とする加工物について、

いずれの微細加工法を用いれば製作できるか、またはいかなる加工法を用いても製作できないことがないかどうか、適切に判断しなければならない。この判断を行うには、マイクロマシン製品を加工する際、あるいは使用する際に起こり得る物理現象を予測し、設計問題を短時間のうちに正確に抽出することが必要である。しかしマイクロマシン製品の場合、通常の寸法の機械製品と異なり、その加工や使用に関係する物理現象は、寸法効果のためにきわめて非直感的なものとなる。そのため設計者が、自身の経験によって自然に身に付けてきた直感に惑わされることなく正確な判断を行うことは、一般には容易なことではない。

この問題を解決するには、コンピュータによる設計支援が非常に重要な役割を果たすと考えられる。通常の寸法の機械製品の開発では、設計者が必要であると判断した時に、適宜シミュレータを起動して物理現象を予測することがなされる。ところがマイクロマシン製品の場合には、問題を起こしやすい物理現象について、シミュレーションの必要性そのものを設計者が見逃す恐れが多分にある。そこで問題を起こしやすい物理現象については、設計者の意識に関係なく設計作業中に絶え間なく評価を行っており、起こり得るトラブルを自動的に発見して設計者に注意を促すような、知的な設計支援システムの利用が効果的であると考えられる。

こうした知的設計支援システムの研究開発はまだ始まったばかりであり、その実現は今後の課題である。しかし将来、マイクロマシン製品の設計環境において知的設計支援システムが非常に重要な位置を占め、マイクロマシン製品の開発について議論を行う際には無視できないものになることが予想される。したがってある設計にどのような設計支援システムが必要となるかを考えることは、同時にその設計方法の難易の評価を行うことでもある。すなわち実現が困難な知的設計支援システムを必要とする設計は実用化しにくいものであり、実際に利用される可能性も小さいと思われる。

そこでマイクロマシン生産に光造形法を導入することにより、要求される知的設計支援システムの仕様がどのように変わるかについて考察する。それにはまず知的設計支援システムが、製品開発におけるどのような問題・トラブルに対して有効なものであるかを把握し、また光造形法を導入することで、これらの問題・トラブルがどのように変わるかを知らなければならない。

機械製品の機能だけを考える場合、製品開発における問題・トラブルは一般に

次の2種類に分類することができる。

1. 「目的の製品が作れない」という形で現れる問題（製品生産時の問題）

目的の製品仕様を満たす製品を製作する手段が見つからない

2. 「製品がこわれやすい」という形で現れる問題（製品使用時の問題）

製品仕様が許容範囲から逸脱しやすいために実用的でない

このうち1.は製品の出荷前に問題となるのに対し、2.の多くは出荷後に問題となって現れ、社会的に大きな影響を及ぼすことも多い。そこで普通メーカでは、製品仕様よりも合格基準を厳しく設定した製品試験を行い、使用時の問題として現れる恐れのある要因を、製品の出荷前に生産時の問題として発現させることで対処する。このような対処方法は、マイクロマシン製品の生産においても通常の寸法の機械製品と変わらないと考えられる。

しかしながら、この方法では本質的に生産時の問題・トラブルを避けることができないので、これについては製品試験と別の方法で対処しなければならない。通常の寸法の機械製品の場合には、設計の際に正確なシミュレーションを行って生産時の問題を予測することで対処するのが普通であり、これ以外には系統的な対処方法が見つからない。特にマイクロマシン製品など製品がこわれやすい場合には、試験の合格基準を本来の製品仕様よりもかなり厳しくする必要があり、製品試験に合格する製品を開発すること自体が難しくなる。したがってマイクロマシン製品の設計段階では、製作段階における問題・トラブルを正確に予測し、試作品が製品試験に合格する確率を大きくしていくことが重要であり、それには正確なシミュレータを備えた知的設計支援システムの利用が特に効果的であると考えられる。

マイクロマシン製品の生産時の問題・トラブルの中でも代表的なものとして、微細加工法の自由度の小ささからくる加工物形状の制約・加工材料の制約など、製作段階における各種の制約が挙げられる。よってマイクロマシン製品の設計・開発にコンピュータを用いた知的設計支援システムを利用するには、製作段階における制約をそれぞれ評価項目としてコンピュータが解釈できるように記述し、あらかじめデータベースの形で用意しておかなければならない。ところが既存の微細加工法を用いる場合、この制約は複雑になることが多く、データベース化も容易ではなかった。すでにシリコンプロセスを用いたマイクロマシン開発を支援

するシステムを作成した研究例⁽¹⁾が発表されているが、製作段階における制約を完全にデータベース化するには至っておらず、実用的なデータベースを作成した研究例はいまだに見当たらないのが現状である。

ところが光造形法を利用したマイクロマシン製法を導入すると、製作段階における制約は非常に単純なものとなり、設計段階で予測しなければならない評価項目の数も少なくなる。そのため、比較的容易に評価項目をデータベース化することができるようになる。現在考えられるすべての評価項目、およびそれぞれの評価に必要なデータベース・解析手法をまとめたものを表8-1.1に示す。

これらの評価項目の抽出に当たっては、本研究で採用したものと同ジステレオリソグラフィ法・規制液面式・走査露光式の加工方法を想定したが、8-2節で述べる加工方法の見直しを行うことで、いくつかの項目は除外することができる。たとえば自由液面式の加工方法を採用する場合には、4.の評価項目を評価する必要がなくなる。

これらの評価項目は、そのまま知的設計支援システムに求められる評価機能を表している。マイクロマシン製品の生産に光造形法を導入することにより、他の多くの微細加工法を利用する場合よりも評価項目の数が少なくなることがわかる。このことは、マイクロマシン製品の生産に光造形法を導入することにより、設計段階で製品評価を行うために必要となるデータベースと解析手法の整備が容易になることを表す。特に中心的な解析手法となる異方性粘弾性FEMについては、その理論的な枠組みがほぼ確立されており、これを知的設計支援システムで利用することができるように新たにプログラム化することはそれほど難しいことではないと考えられる。また樹脂の固化現象を正確にシミュレートすることのできるモデルが確立され、材料試験などの実験を行うことなく加工条件から解析条件を求める方法が確立されれば、きわめて能率的な製品開発が実現されるものと期待される。

製作段階における制約が少ないことは、シミュレーションによって問題を予測しながら設計を進める先導設計を実現する上でも非常に重要な意味を持つ。先導設計では、製品の開発過程で未開発のシーズが発見されると、そのシーズが実現された場合に製品の性能がどのように変わるかをシミュレーションによって予測する。したがって正確なモデル化が容易で、再現性の大きいシミュレーションが

できるような製品を開発する場合には、先導設計が特に効果的な設計手法となる。

マイクロマシン製作に光造形法を導入すると、設計段階での評価項目の多くが非常にモデル化しやすく、解析的に評価できるものとなる。そのため設計段階で正確なシミュレーションを行い、その先の開発段階にある技術課題を予測できるようになる。すなわち先導設計を活用する上でも、光造形法を利用したマイクロマシン製法は有利である。

マイクロマシン製品の生産時の問題・トラブルとしては、以上で述べたような製作段階における制約によるものだけでなく、物理現象が非直感的であるために基本設計が難しいことも挙げられる。通常の寸法の機械の場合、すでに設計者はさまざまな機構原理を直感的に理解しており、それらの機構原理を組み合わせることで目的の動作を実現する。それに対してマイクロマシンの場合、機構原理が通常の寸法の機械に用いられるものとは異なり、きわめて非直感的なものとなる。そのため目的の動作を実現できる機構原理の組み合わせを抽出し、それを最適化することは非常に難しくなる。

このことを考えると、知的設計支援システムには機構原理のライブラリを探索する機能も求められる。マイクロマシン製品の設計では、構造部材の変形を利用した機構原理を多く用いることになると考えられる。そこで機械構造を構成する部材について、単に変形を解析するだけでなく、力や動きを伝達する機能を持つ機構としてパラメトリックに設計することが必要となる。そこでさまざまな機構原理をデータベース化したライブラリを用意し、要求される伝達特性を実現する機構原理の組み合わせを探索することのできる知的設計支援システムを開発することが望まれる。

8-2. 今後の課題とその解決方法についての考察

本研究では、マイクロマシン製法としての光造形法の仕様を確かめることができただけでなく、光造形法を利用したマイクロマシン生産を実用化するために解決しなければならない課題や、複数の課題の間にある関連性を抽出することができた。そこでそれらの課題とその解決方法を整理し、さらに本研究で得られた知見を踏まえて今後の改良の指針を示しておきたい。

マイクロマシンの応用分野が広がるにつれ、マイクロマシンを作るための微細加工法には、一般に次のような課題の解決が求められるようになって考えられる。

1. 垂直方向（樹脂液面に垂直な方向）の加工分解能の向上
2. 水平方向（樹脂液面に平行な方向）の加工分解能の向上
3. 加工時間の短縮
4. 生産性の向上
5. より複雑な機械を作る方法の確立

本研究では光造形法に要求される最も重要な仕様として加工分解能を取り上げ、それを高めることを最初の課題として議論してきた。しかし光造形法には各種の加工方法があり、方法によってこれらの課題のうちのいくつかがトレードオフの関係となる場合もある。したがって実際の生産に应用するには、すべての課題を考慮して総合的な評価を行い、最適な加工方法・加工条件を選択する必要がある。そこで各課題に対する具体的な対処、およびそれらによって派生する副作用的な問題について検討を行った。それぞれの課題が独立しておらず、互いに関連している様子を表8-2.1に示す。

これらの課題・対処・問題を詳細に追跡することで、それぞれの課題に対する解決方法を提案することができた。この検討結果をまとめたものを、関連研究とあわせて次に示す。

1. 垂直方向（樹脂液面に垂直な方向）の加工分解能の向上

光造形法をマイクロマシン製作に应用する場合には、水平方向の加工分解能よりも垂直方向の分解能を高めることが難しくなる。したがって全体としての加工分解能を高めるには、特に垂直方向の分解能を高める必要がある。第4章で示した改善指針もすべて垂直方向の加工分解能を高めるためのものであり、これらの指針を具体化することが基本となる。

第4章で示した三つの改善指針の中でも、「樹脂の吸収係数を大きくすること」は特に本質的な指針であると考えられる。本研究では樹脂に有機染料を添加して実験を行ったが、さらに吸収係数を大きくするにはカーボンブラックや金属粉など、無機物質の微粉末を光吸収材料として添加することが必要となる。この場合には光吸収材料を樹脂中に安定に分散させる方法が問題となるが、これについては印刷インキ製造などに

課題1. 垂直方向（樹脂液面に垂直な方向）の加工分解能の向上

- 対処10. 露光スポットの直径を安定化する
- 対処12. 露光ビームの放射量を安定化する
- 対処22. 樹脂の吸収係数を大きくする

課題2. 水平方向（樹脂液面に平行な方向）の加工分解能の向上

- 対処11. 露光スポットの直径を小さくする

課題3. 加工時間の短縮

- 対処7. 樹脂の光硬化反応に必要な時間を短くする
- 対処20. 加工物断面の歪みに要する時間を短くする
- 対処29. 樹脂液面の位置の制御に必要な時間を短くする

課題4. 生産性の向上

- 課題3. 加工時間の短縮
- 対処44. 試作から設計へのフィードバック時間を短くする
- 対処47. 一度に作る試作品・製品の数を多くする

課題5. より複雑な機械を作る方法の確立

- 対処30. 自由曲面式・準自由曲面式の加工方法を採用する

対処1. 白熱灯・放電灯・固体発光素子などを光源に用いる

- 問題7. 樹脂液面にできる光源の像が大きくなり、水平方向の加工分解能が低下する
- 問題11. 樹脂の光硬化反応に必要な時間が長くなり、加工時間が長くなる

対処2. ガスレーザを光源に用いる

- 問題8. 露光ビームの回折の影響が大きくなり、水平方向の加工分解能が低下する

対処3. 半導体レーザを光源に用いる

- 問題8. 露光ビームの回折の影響が大きくなり、水平方向の加工分解能が低下する

対処4. 露光ビームの放射束を大きくする

- 問題2. 温度上昇によって樹脂や基板の表面処理が変質し、加工不可能になる

対処5. 露光ビームの放射束を小さくする

- 問題11. 樹脂の光硬化反応に必要な時間が長くなり、加工時間が長くなる

対処6. 露光ビームの光ノイズを小さくする

- 対処1. 白熱灯・放電灯・固体発光素子などを光源に用いる
- 対処3. 半導体レーザを光源に用いる

対処7. 露光ビームの放長域を狭くする

- 対処2. ガスレーザを光源に用いる
- 対処3. 半導体レーザを光源に用いる

対処8. 露光ビームの放長を短くする

- 問題4. 樹脂の吸収係数を大きくすることができなくなり、垂直方向の加工分解能が低下する

対処9. 基礎部の加工中に露光ビームの放長を長くする

- 対処34. 光硬化剤を添加した可視光硬化性樹脂を用いる
- 問題15. 加工システムが大規模・複雑・高価になり、実用化しにくくなる

対処10. 露光スポットの直径を安定化する

- 対処13. 露光ビームの焦点を樹脂液面から外す
- 対処18. 露光レンズの開口数を小さくする
- 対処20. 光学系の振動を低く

表8-2.1 課題に対する対処と派生する問題の関連（その1）

- 対処11. 露光スポットの直径を小さくする
 →対処14. 露光ビームの焦点を樹脂表面に合わせる
 →対処15. 露光ビームの回折の影響を小さくする
 →対処16. 集光レンズの収差の影響を小さくする
 →問題12. 加工物断面の公差に要する時間が長くなり、加工時間が長くなる
- 対処12. 露光ビームの放射量を安定化する
 →対処5. 露光ビームの光ノイズを小さくする
 →対処21. 露光ビームの光ノイズを平均化する
- 対処13. 露光ビームの焦点を樹脂表面から外す
 →問題5. 露光スポットの径が大きくなり、水平方向の加工分解能が低下する
- 対処14. 露光ビームの焦点を樹脂表面に合わせる
 →問題8. 露光ビームの回折の影響が大きくなり、水平方向の加工分解能が低下する
 →問題10. 光学系の振動に対し、垂直方向・水平方向の加工分解能が低下しやすくなる
- 対処15. 露光ビームの回折の影響を小さくする
 →対処1. 白熱灯・放電灯・固体発光素子などを光源に用いる
 →対処17. 集光レンズの開口数を大きくする
- 対処16. 集光レンズの収差の影響を小さくする
 →対処7. 露光ビームの直径域を狭くする
 →対処18. 集光レンズの開口数を小さくする
 →対処19. 非球面・色消しの集光レンズを用いる
- 対処17. 集光レンズの開口数を大きくする
 →問題6. 集光レンズの収差の影響が大きくなり、水平方向の加工分解能が低下する
 →問題10. 光学系の振動に対し、垂直方向・水平方向の加工分解能が低下しやすくなる
- 対処18. 集光レンズの開口数を小さくする
 →問題8. 露光ビームの回折の影響が大きくなり、水平方向の加工分解能が低下する
 →問題11. 樹脂の光固化反応に要する時間が長くなり、加工時間が長くなる
- 対処19. 非球面・色消しの集光レンズを用いる
- 対処20. 光学系の振動を除く
- 対処21. 露光ビームの光ノイズを平均化する
 →対処5. 露光ビームの放射量を小さくする
- 対処22. 各セルに対応する露光を複数回に分けて行う
 →問題12. 加工物断面の公差に要する時間が長くなり、加工時間が長くなる
- 対処23. 複数のセルに対応する露光を同時に行う
 →対処24. 露光ビームの本数を多くする
 →対処25. 液晶マスクを用いた一括露光式の加工方法を採用する
 →対処26. 電子写真を用いた一括露光式の加工方法を採用する
- 対処24. 露光ビームの本数を多くする
 →問題15. 加工システムが大規模・複雑・高価になり、実用化しにくくなる
- 対処25. 液晶マスクを用いた一括露光式の加工方法を採用する
 →問題3. 液晶マスクが紫外光によって劣化し、加工不可能になる
- 対処26. 電子写真を用いた一括露光式の加工方法を採用する
 →問題15. 加工システムが大規模・複雑・高価になり、実用化しにくくなる
- 対処27. 樹脂の光固化反応に要する時間を短くする
 →対処4. 露光ビームの放射量を大きくする
 →対処35. 樹脂の光感度を大きくする

表8-2.1 課題に対する対処と派生する問題の関連(その2)

- 対処28. 加工物断面の走査に要する時間を短くする
 →対処23. 複数のセルに対応する露光を同時に行う
 →対処42. 露光ヘッドの動作を高速化する
- 対処29. 樹脂表面の位置の制御に要する時間を短くする
 →対処37. 樹脂の粘度を小さくする
- 対処30. 自由液面式・準自由液面式の加工方法を採用する
- 対処31. 基礎部の厚さを薄くする
 →問題1. 樹脂の圧力によって基板が破損し、加工不可能になる
- 対処32. 樹脂の吸収係数を大きくする
 →対処33. 吸収係数の大きい樹脂を新規開発する
 →対処34. 光吸収材を添加した可視光硬化性樹脂を用いる
 →問題11. 樹脂の光固化反応に要する時間が長くなり、加工時間が長くなる
 →問題14. 基礎部の加工に要する時間が著しく長くなり、加工時間が長くなる
- 対処33. 吸収係数の大きい樹脂を新規開発する
- 対処34. 光吸収材を添加した可視光硬化性樹脂を用いる
 →問題11. 樹脂の光固化反応に要する時間が長くなり、加工時間が長くなる
- 対処35. 樹脂の光感度を大きくする
 →対処36. 光感度の大きい樹脂を新規開発する
- 対処36. 光感度の大きい樹脂を新規開発する
- 対処37. 樹脂の粘度を小さくする
 →対処38. 粘度の小さい樹脂を新規開発する
 →対処39. 加工中に樹脂の温度を上げる
- 対処38. 粘度の小さい樹脂を新規開発する
- 対処39. 加工中に樹脂の温度を上げる
- 対処40. 基板の面積を大きくする
 →問題1. 樹脂の圧力によって基板が破損し、加工不可能になる
- 対処41. 基板の表面処理の耐熱性を上げる
- 対処42. 露光ヘッドの動作を高速化する
 →問題9. スキャナ位置決の精度が低下し、水平方向の加工分解能が低下する
- 対処43. エレベータの動作を低速化する
 →問題12. 加工物断面の走査に要する時間が長くなり、加工時間が長くなる
- 対処44. 試作から設計へのフィードバック時間を短くする
 →対処45. 専用の設計支援システムを開発する
 →対処47. 一度に作る試作品・製品の数を多くする
- 対処45. 専用の設計支援システムを開発する
- 対処46. 加工装置の台数を多くする
 →問題15. 加工システムが大形・複雑・高価になり、実用化しにくくなる
- 対処47. 一度に作る試作品・製品の数を多くする
 →対処40. 基板の面積を大きくする
 →対処46. 加工装置の台数を多くする

表8-2.1 課題に対する対処と派生する問題の関連 (その3)

- 問題1. 樹脂の圧力によって意板が破損し、加工不可能になる
 →対策37. 樹脂の粘度を小さくする
 →対策43. エレベータの動作を低速化する
- 問題2. 温度上昇によって樹脂や意板の表面処理が変質し、加工不可能になる
 →対策5. 露光ビームの照射量を小さくする
 →対策22. セールに対応する露光を複数回に分けて行う
 →対策25. 露光マスクを用いた一括露光式の加工方法を採用する
 →対策26. 電子写真を用いた一括露光式の加工方法を採用する
 →対策41. 意板の表面処理の耐熱性を上げる
- 問題3. 液晶マスクが紫外光によって劣化し、加工不可能になる
 →対策34. 光吸収材を添加した可視光硬化性樹脂を用いる
- 問題4. 樹脂の収収係数を大きくすることができなくなり、垂直方向の加工分解能が低下する
 →対策10. 露光スポットの直径を安定化する
 →対策12. 露光ビームの照射量を安定化する
- 問題5. 露光スポットの径が大きくなり、水平方向の加工分解能が低下する
- 問題6. 露光レンズの収差の影響が大きくなり、水平方向の加工分解能が低下する
 →対策7. 露光ビームの後戻りを防ぐ
 →対策15. 露光レンズの開口数を小さくする
 →対策19. 非球面・色選しの露光レンズを用いる
- 問題7. 樹脂表面にできる光源の像が大きくなり、水平方向の加工分解能が低下する
 →対策2. ボスレーザを光源に用いる
 →対策3. 半導体レーザを光源に用いる
 →対策17. 露光レンズの開口数を大きくする
- 問題8. 露光ビームの回折の影響が大きくなり、水平方向の加工分解能が低下する
 →対策1. 白熱灯・放電灯・固体発光素子などを光源に用いる
 →対策17. 露光レンズの開口数を大きくする
- 問題9. スキャナの位置決め精度が低下し、水平方向の加工分解能が低下する
 →対策25. 露光マスクを用いた一括露光式の加工方法を採用する
 →対策26. 電子写真を用いた一括露光式の加工方法を採用する
- 問題10. 光学系の振動に対し、垂直方向・水平方向の加工分解能が低下しやすくなる
 →対策20. 光学系の振動を除く
- 問題11. 樹脂の光固化反応に要する時間が長くなり、加工時間が長くなる
 →対策4. 露光ビームの照射量を大きくする
 →対策35. 樹脂の光感度を大きくする
- 問題12. 加工物断面の走査に要する時間が長くなり、加工時間が長くなる
 →対策23. 複数のセールに対応する露光を同時に行う
 →対策42. 露光ヘッドの動作を高速化する
- 問題13. 樹脂表面の位置の制御に要する時間が長くなり、加工時間が長くなる
 →対策37. 樹脂の粘度を小さくする
- 問題14. 基礎部の加工に要する時間が著しく長くなり、加工時間が長くなる
 →対策4. 露光ビームの照射量を大きくする
 →対策9. 基礎部の加工中に露光ビームの径を長くする
 →対策31. 基礎部の厚さを薄くする
- 問題15. 加工システムが大型・複雑・高価になり、実用化しにくくなる
 →対策1. 白熱灯・放電灯・固体発光素子などを光源に用いる
 →対策3. 半導体レーザを光源に用いる

用いられているコロイド技術を参考にすることができると考えられる。

多くの物質が電子ビームに対して大きい吸収係数を持っていることを利用し、電子ビームを用いて露光を行う方法も考えられる。しかしこの方法では、樹脂が真空中に置かれることになるので、樹脂の変質を防ぐための工夫が必要となる。樹脂の電子線固化はすでに実用化されている技術であるが、一般には加工時間が短く、樹脂の変質が問題とならない場合が多い。

光造形法の加工精度は、未固化の樹脂を除くための溶剤洗浄の条件に大きく関係している。そこで露光量が限界露光量に達するまでは固化がほとんど起こらず、限界露光量を超えると急峻に固化する「自己触媒的 (self-catalytic)」な光感度特性を持つ樹脂が望まれる。

走査露光式では露光スポットの直径の変動を避けるため、造形装置の主要部分に大きい剛性を持たせることが重要である。また走査露光式の代わりに、露光ビームの光ノイズや露光スポットの直径の変動の影響が小さい、一括露光式を採用する方法も考えられる。

2. 水平方向 (樹脂液面に平行な方向) の加工分解能の向上

水平方向の加工分解能は、主に露光スポットの直径によって決まる。そこで電子ビームを用いて露光を行い、露光ビームの回折限界を小さくする方法が考えられる。これについては IC 製造などに用いられている描画露光 (パターニング) 技術を参考にすることができると考えられる。

垂直方向・水平方向の最終的な加工分解能は、樹脂中での反応物質の拡散距離によって決まる。通常の樹脂では $10 \text{ nm} \sim 100 \text{ nm}$ の加工分解能が実現できるものと考えられる。

3. 加工時間の短縮

マイクロマシン製品の大量生産を考えると、加工時間の短縮も重要な課題となる。RP/M の場合には露光ビームの放射束を大きくして対処することが多いが、マイクロマシン製作の場合には窓板の表面処理層や樹脂の温度上昇によってトラブルが起こるので、露光ビームの放射束を大きくすることが難しい。この問題に対し、露光を複数回に分けて行うマルチビーム露光式、あるいはモデルの断面全体を長い時間にわたって

露光する一括露光式の加工方法が研究されている⁽¹⁾。

また光硬化性樹脂の光感度も加工時間に大きな影響を及ぼすことから、光感度の大きい樹脂も強く望まれる。

4. 生産性の向上

加工物の断面積を大きくすることで、一度の加工で多数の製品を作る方法が考えられる。しかし規制液面式の加工装置に用いられる窓板には、その寸法が大きくなると絞り膜効果による大きい曲げ力がかかるようになるので、曲げに対する窓板の強さが問題となる。このことは、一層の厚さの薄いマイクロマシン製作においては特に大きな問題となる。この問題に対しては、自由液面式を採用して対処する方法が考えられている。

またモデルを能率的に記述することのできるデータ形式が必要となる。STL形式やピットマップ形式では、加工物形状が複雑になるにつれてデータの大きさが大きくなり、多数の製品を一度に作るような場合には非常に大きいデータを扱うことになる。複製・グループ化などの高級な記述をサポートした、PostScript^(R)のような形状記述言語の開発と規格化が望まれる。

5. より複雑な機械を作る方法の確立

サポートを挿入せずに破損しやすい加工物を作る方法としては、まず固体下地法を採用する方法が考えられる。また樹脂を冷却して一時的に固化させておき、露光を行って光固化反応を起こさせてから、加温して加工物を得る方法も研究されている⁽²⁾。

光硬化性樹脂はマイクロマシンの材料として用いる場合、寸法効果のために非常に破損しにくい材料となる。しかし剛性・耐熱性などの材料特性を考慮すると、光硬化性樹脂だけでは不十分な場合も考えられる。そこで光造形法によって型を作製し、これから電鍍や鋳造によって金属製品を作る方法が研究されている⁽¹⁾⁽³⁾。

樹脂に金属やセラミックの微粉末を混合し、吸収係数を大きくすると同時に、機械特性の改善を図る方法も研究されている⁽⁴⁾。また微粉末を混合した樹脂で作られた加工物をそのまま製品とするだけでなく、焼結することで、金属やセラミックの製品を作る方法も研究されている⁽⁵⁾。

樹脂に多量の粉末を加えると、樹脂の見かけの粘性が非常に大きくなり、樹脂表面の制御がきわめて難しくなることから、自由液面式を採用して対処する方法も考えられている。

あらかじめ他の微細加工法で作られた、複数の半製品を機械的に結合する「結合機構 (glue mechanism)」を作ることができれば、複雑な構造を持つマイクロマシン製品を組み立てることができる。また一つの加工物の内部で複数の材料を使い分けることができれば、複雑な機能を持つマイクロマシン製品を作ることでもできる。たとえば導電材料と絶縁材料を組み合わせるにより、機械構造と同時に電気回路を構成することができる。

以上を整理すると、今後光造形法を実際のマイクロマシン製作に応用していく上では、次のようなアプローチで光造形法、およびその関連技術を整備していくことが必要になると考えられる。

1. 粘度が小さく、光感度の大きい可視光硬化性樹脂の開発
2. 自由液面式、または準自由液面式の採用
3. 露光ビームの放射量の安定化
4. 可視非コヒーレント光源と液晶パネルによる一括露光式の採用
5. 設計・製作を通して利用できる専用設計支援システムの開発

このように光造形法をマイクロマシン製法として考える場合には、これまで気付かれることの少なかった多くの可能性が発見されることがわかった。今後は光造形法を利用したマイクロマシン製法を導入することによって、どのようなマイクロマシンを製作できるようになるか、その可能性の限界を見極めることが重要な課題になると思われる。

文献

- (1) Gilbert, J. R., Osterberg, P. M., Harris, R. M., Ouma, D. O., Cai, X., Pfajfer, A., White, J. and Senturia, S. D., Implementation of a MEM-CAD System for Electrostatic and Mechanical Analysis of Complex Structures from Mask Descriptions, Conference Proceedings of MEMS '93 (1993), pp. 207-212.

- (2) 矢野崇史, 中島尚正, マイクロマシン用光造形におけるマスク露光方式に関する研究, 平成6年度東京大学工学部卒業論文 (予定)
- (3) 中屋裕一郎, 中島尚正, マイクロマシン用光造形における凍結切削式液面規制に関する研究, 平成6年度東京大学工学部卒業論文 (予定)
- (4) 山口勝美, 中本剛, Abbay, P., A., 壬生周, 紫外線感光性樹脂を使ったマイクロストラクチャの製造 (第1報), 機論 Vol. 57 No. 542 C (1991), pp. 288-294.
- (5) 生田幸士, 広渡憲, 光創製3次元マイクロファブリケーション, 機論 No. 920-33 B (1992), pp. 545-546.
- (6) 萩原茂, 日原政彦, 八代浩二, 佐野正明, 粉末混入樹脂による光造形に関する研究 (第1報), 1993年度精密工学会秋期講演会論文集 (1993), pp. 507-508.
- (7) 单学伝, 横浜正毅, 柳沢一向, 光造形法を応用したセラミック/金属微小構造体の製法 (第1報), 1994年度精密工学会春期講演会論文集 (1994), pp. 579-580.

第9章 まとめ

以上のように、光造形法を利用したマイクロマシン製作法の実用化を目指して研究を行った。最後に本研究のアプローチを振り返り、本研究の成果をまとめておくことにする。

まず光造形法の加工分解能を高めることを考えて、樹脂が固化する際に起こる物理現象を考慮し、「固化セル」と呼ぶ概念を導入してシミュレーション・数式解析を行った。その結果、加工条件と加工分解能の関係を把握することができ、加工分解能を高めるための指針を導くことができた。(→第4章)

第4章の結論をここで再掲する。

1. 露光ビームの放射量を安定化する。
2. 露光スポットの直径を安定化する。
3. 樹脂の吸収係数を大きくする。

次に理論検討で得られた指針を踏まえ、マイクロマシン製作用の実験用光造形システムを試作した。この実験用光造形システムを用い、加工条件を変えながら加工分解能を評価するためのテストモデルを作製した。その結果、加工分解能を約 $5\mu\text{m}$ にまで高めることができ、これらの指針の妥当性を示すことができた。

(→第5章・第6章)

第5章・第6章の結論をここで再掲する。

1. 第4章で示した三つの改善指針を具体化することによって加工分解能を大幅に高めることができ、水平方向と垂直方向で同時に $10\mu\text{m}$ 以下の分解能を実現することができる。
2. 露光ビームに一定の焦点外しを与え、露光スポットの直径を一定の値に保つことで、良好な形状・寸法を持つ固化セルが作られる。スキャナの走査分解能が $10\mu\text{m}$ の場合には、露光スポットの直径を約 $60\mu\text{m}$ とした時に、全体として最も高い加工分解能が得られた。
3. 加工分解能を高めるために樹脂の吸収係数を大きくすると、樹脂に照射される露光光線のエネルギーの多くが熱として消費されるようになるので、エネルギー効率は低下する。また発生した熱は、樹脂の焼損や変質などの原因となる。このトラブルを避けるには、露光ビームの放射束を小さく

する方法も考えられるが、加工時間は長くなる。

さらに実験用光造形システムを用い、各種の動作原理の機構をテストモデルとして作製した。また微細な機構を駆動する各種の方法を提案し、テストモデルに応用した。その結果、設計したとおりにモデルが動くことを確認することができ、可動部を持つ機械構造を作る方法としての光造形法の有用性を示すことができた。

(→第7章)

第7章の結論をここで再掲する。

1. 樹脂固形物を材料として、実際に動かすことのできる微細な機構を作製することができる。
2. 微細な機構を駆動するため、基板内にアクチュエータを組み込んで機械内部の目的の部分まで力や動きを伝達していく方法が利用できる。
3. 光造形法で作製できる加工物形状の制約は、サポートを挿入することの難しい機械構造を設計する際に大きな問題となる。より大きな剥離性を持つ窓板表面の処理方法を確立したり、自由液面式や固体下地法を採用したりするなどの対処が必要である。
4. 規制液面式では、構造部材の弾性変形を利用した機構を作製することが難しい。自由液面式や固体下地法を採用し、加工物を窓板表面から引き離す工程をなくすなどの対処が必要である。
5. 光造形法によって作られる微細な機構は、通常の機械に用いられている機構とは異なった構造を持つ。寸法効果や加工の仕様を考慮しながら、利用することのできる機構原理を模索していくことが必要である。

最後に光造形法を利用したマイクロマシン製作法の実用化に向け、そのための工学的な指針を示した。また本研究の遂行によって新たに見つけ出された課題を整理し、今後の解決方法を提案した。(→第8章)

「光造形法」と「マイクロマシン」はどちらも若い技術であり、研究の余地も多く残されているのが現状である。光造形法を利用したマイクロマシン製作法は、RP/M用として開発されてきた光造形法の単なる改良に過ぎないという側面を持っている。しかしマイクロマシンの研究分野では、これまで予想されなかった新しい考え方が有効となる可能性もある。従来常識にとらわれない今後の技術開発に期待しつつ、本論文の結びとしたい。

謝辞

本研究を遂行するに当たり、親切かつ多大なるご指導をくださった、東京大学工学部産業機械工学科の中島尚正教授にはとりわけ深い感謝の意を申し上げます。また本論文をまとめる上で貴重なご助言をくださった、東京大学工学部機械情報工学科の三浦宏文教授、東京大学工学部産業機械工学科の畑村洋太郎教授、東京大学先端科学技術研究センターの佐藤知正教授には深い感謝の意を申し上げます。そして5年の間たゆまず本研究を支援してくださった、東京大学工学部産業機械工学科の村上存助教授・及川和広技術官には深く感謝いたします。また本研究で使用した実験材料を提供してくださった、三井造船(株)の永森茂様と三洋化成工業(株)の虎谷純行様、(株)ディーメックの田上英二郎様・栗原文夫様には深く感謝いたします。さらに光造形法に関する幅広い知識を提供してくださった、(株)シーメットの早野誠治様、(株)デンケンエンジニアリングの日浦昭二様をはじめとする光造形シンポジウム出席者の皆さま、またマイクロマシンに関するたくさんの示唆をくださった、計量研究所の梅田章様をはじめとする「マイクロマシンの世界」分科会・「マイクロマシンからナノマシンへ」分科会・マイクロマシン研究会出席者の皆さま、そしてこれらの研究会でお世話になった事務局の皆さまには深く感謝いたします。また共同研究者とともに問題に取り組んでくださった、単学伝様をはじめとするオリンパス光学工業(株)の皆さま、安藤充宏様・水野智公様をはじめとする(株)アイシニコスモス研究所の皆さまには深く感謝いたします。また本研究と筆者を温かく見守ってくださった、東京大学工学部産業機械工学科設計工学研究室の皆さまには心からのお礼を申し上げます。最後に本研究遂行中の1994年3月5日にご逝去されました、東京大学工学部産業機械工学科の故林紀子様には、本研究への献身的なご協力に深く感謝するとともに、ご冥福を心よりお祈り申し上げます。

なお本研究は、平成5年度・6年度科学研究費補助金(試験研究B)、および日本学術振興会特別研究員に交付する平成6年度科学研究費補助金を受けて遂行されたものです。

付録

次の資料を付録として添付する。

- ・用語について
- ・光造形システム技術マニュアル
 - ・加工装置の構成
 - ・I/Oポート割り当て
 - ・電子回路回路図・接続図
 - ・プログラム説明書・リスト
 - ・使用機材・材料一覧
- ・製造元・販売元・工場一覧
 - ・機材製造元・販売元
 - ・材料製造元・販売元
 - ・加工工場
- ・機材仕様・材料特性一覧
 - ・CD-3041R (He-Cdレーザー) の仕様
 - ・532-10B (Ar⁺レーザー) の仕様
 - ・UV-854 (光硬化性樹脂) の特性
 - ・SCR-500 (光硬化性樹脂) の特性
 - ・VL-2302 (光硬化性接着剤) の特性
- ・研究発表一覧

付録—用語について

マイクロマシンや光造形法の研究は、開始されてからの歴史が浅く、その関連分野も光学・機械工学・化学・情報工学など多岐にわたっており、特殊な用語が使われることが多い。本論文にもこのような用語が用いられているが、定義や説明を本文中に書くとは混乱を招く恐れがあるので、ここに一覧を置くことにした。必要に応じて適宜参照されたい。

光造形法 photopolymer fabrication, photocuring

光硬化性樹脂を用いた立体製作法の総称。英語の呼称とされている"stereo-lithography"は商品名であり、一般的には使われない。

固化 solidification

液状物が変化して固形物となること。(→硬化)

硬化 hardening

固形物の硬さが増大すること。「光硬化性樹脂」などの語句に使われている「硬化」(curing)は化学用語であり、一般的には使われない。(→固化)

露光光線 exposure ray

露光に用いられる光線のこと。

露光ビーム exposure beam

有限の幅を持つ露光光線のこと。

放射量 radiant energy

電磁波によって伝わるエネルギー量のこと。単位はジュール (J)。

放射束 radiant flux

単位時間当たりの放射量のこと。単位はワット (W)。

放射照度 irradiance

単位面積当たりの被照射面に照射される放射束のこと。単位はワット毎平方メートル (W/m^2)。

吸収線量 absorbed doze

単位質量当たりの物質によって吸収される放射量のこと。単位はジュール毎キログラム (J/kg)。(→3-2節)

露光量 doze of exposure

単位面積当たりの被照射面に照射される放射量のこと。あるいは露光光線の放射照度を時間で積分したもの。単位はジュール毎平方メートル (J/m^2)。光硬化性樹脂の固化は、一定の露光量で起こるものと考えることができる。(→3-2節)

露光スポット exposure spot

露光ビームを被照射面で切った断面のこと。

微細 fineness

対象とする構造の代表寸法が小さいこと。

微小 tininess

対象とする加工物の外形寸法が小さいこと。

精密 precision

対象とする加工物の加工精度が高いこと。

加工分解能 machining resolution, forming resolution

独立に加工することのできる2点の間の最小距離のこと。この距離が小さいほど、加工分解能が高いという。(→加工精度・4-1節)

加工精度 machining accuracy, forming accuracy

設計寸法と加工物寸法の差のこと。この差が小さいほど、加工精度が高いという。(→加工分解能・4-1節)

加工の自由度 freedom of machining, freedom of forming

ある加工物形状の集合を規定した時、その集合要素のうちで加工することのできる要素が占める割合のこと。

モデル model

立体形状を記述するコンピュータデータのこと。または、そのコンピュータデータから作られる加工物のこと。

セル cell, voxel

三次元ビットマップ形式のモデルの構成単位のこと。(→4-1節)

剥離性 separability

固化物を固体表面から引き離すのに要する力を表す比較値の一つ。この力が小さいほど、剥離性が大きいという。(→5-1節)

付録——光造形システム技術マニュアル

加工装置の構成

P Fシステムの加工装置は、制御装置と造形装置から構成されている。

制御装置にはパーソナルコンピュータを使用する。パーソナルコンピュータの拡張スロットにはT T Lレベルのパラレル入出力を行う拡張ボードを挿入する。拡張ボードに接続する信号線については、「I/Oポート割り当て」の項で詳述する。

拡張ボードのベースポートアドレスは00D0Hに設定する。この設定は拡張ボードの上にあるDIPスイッチによって行う。(→コンテック P I O-48W (98) 取扱説明書)

制御装置は、拡張ボードに付属しているフラットケーブルを通して造形装置の電子回路に接続する。この電子回路については、「電子回路回路図・接続図」の項で詳述する。

造形装置全体は振動を避けるため、防振器を備えた光学定盤の上に固定する。

加工装置は専用の実験室建て屋の中に置き、クラス3 Bレーザ管理区域として安全基準に基づく安全管理を行う必要がある。

I/Oポート割り当て

拡張ボードの入出力端子に割り当てられる信号の名称と機能を以下に示す。

A00	EXPOSE	B00	BUTTON	C00
A01	$\overline{\text{EXPOSE}}$	B01	CLOCK	C01
A02	LASER	B02	$\overline{\text{LCW}}$	C02
A03	$\overline{\text{SEPARATOR}}$	B03	$\overline{\text{LCCW}}$	C03
A04	$\overline{\text{TEST}}$	B04		C04
A05	$\overline{\text{CALIBRATE}}$	B05		C05
A06	$\overline{\text{BASE}}$	B06		C06
A07	CLEAR PHOTOMETER	B07		C07
A08	MT0DIR	B08	D0	C08
A09	MT0TRIG	B09	D1	C09
A10	MT1DIR	B10	D2	C10
A11	MT1TRIG	B11	D3	C11
A12	MT2DIR	B12	D4	C12
A13	MT2TRIG	B13	D5	C13
A14	MT3DIR	B14	D6	C14
A15	MT3TRIG	B15	D7	C15

EXPOSE・**EXPOSE**：ビームシャット制御出力

ビームシャットを開放している時に、EXPOSEがHレベル、**EXPOSE**がLレベルとなる。

LASER：レーザ制御出力

P F M A K EのパラメータLSDがYESの場合、加工が終了した時にLレベルとなる。

SEPARATOR：セパレータ制御出力

エレベータが上昇し、加工物を基板表面から引き離している時にLレベルとなる。

TEST：割り込み検査モード出力

加工シーケンスが割り込み検査モードに入っている時にLレベルとなる。

CALIBRATE：光出力校正モード出力

P F M A K EのパラメータLPBがYESの場合、加工シーケンスが光出力校正モードに入っている時にLレベルとなる。

BASE：基礎部加工モード出力

加工シーケンスが基礎部加工モードに入っている時にLレベルとなる。

CLEAR PHOTOMETER：光量計クリア出力

積算光量計の積算値をクリアする時にHレベルとなる。

MT*DIR：モータ回転方向出力

スキヤナを駆動するステッピングモータの回転方向を制御する信号である。

MT*TRIG：モータ回転トリガ出力

スキヤナを駆動するステッピングモータを1ステップ回転させる信号である。

BUTTON：割り込みボタン入力

割り込みボタンが押された時にLレベルとなるように接続する。

CLOCK：外部クロック入力

光量計からのデータ入力を行う際に、データの同期を取る信号である。

LCW・**LCCW**：時計回り移動限界入力・反時計回り移動限界入力

いずれかのリミットスイッチが、スキヤナの位置が限界位置に達したことを検出した時に、対応するLCW・LCCWがLレベルとなるように接続する。

D*：データ入力

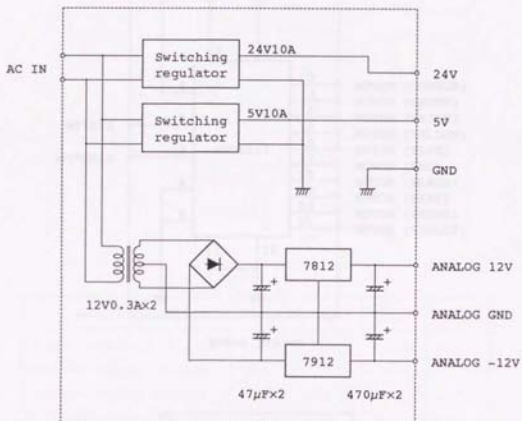
光量計からのバイナリデータを入力する信号である。

電子回路回路図・接続図

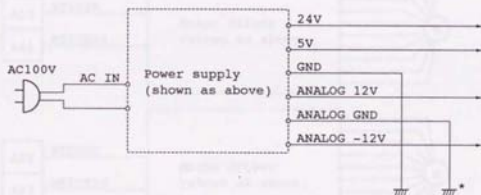
造形装置の電子回路について、その回路図と設計仕様、制御装置との接続図を次ページ以下に示す。

- ・電源回路
- ・スキャナ駆動回路
- ・ビームシャッタ駆動回路
- ・光量計回路
- ・その他の電子回路

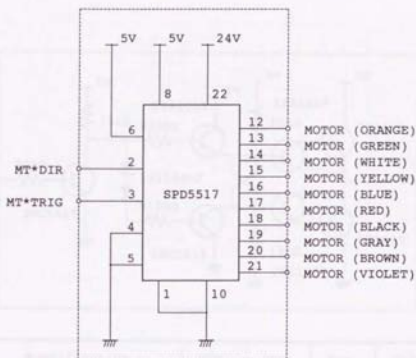




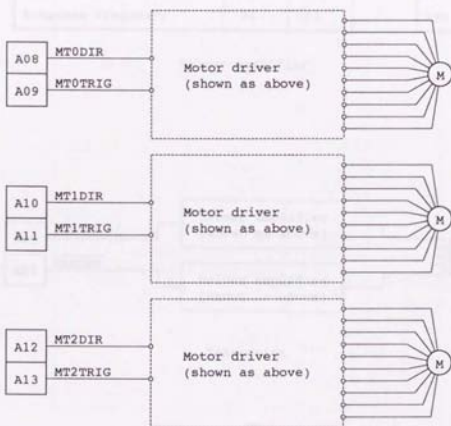
Power supply circuit



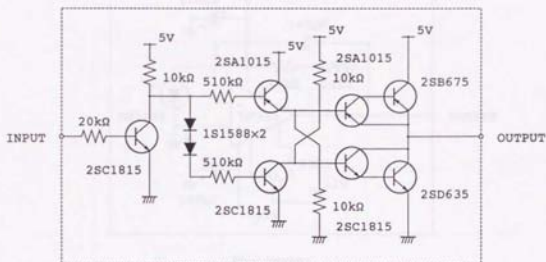
Connection



Motor driver

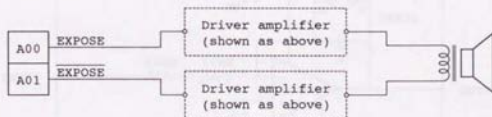


Connection

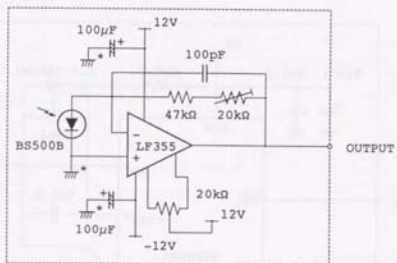


Specification	Min.	Typ.	Max.	Unit
Input threshold voltage	0.6		1.0	V
Maximum supply voltage	20	30		V
Maximum output current	4	7		A
Response frequency	20	25		kHz

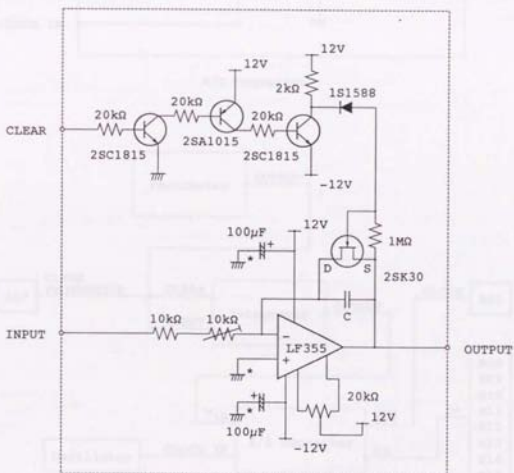
Driver amplifier



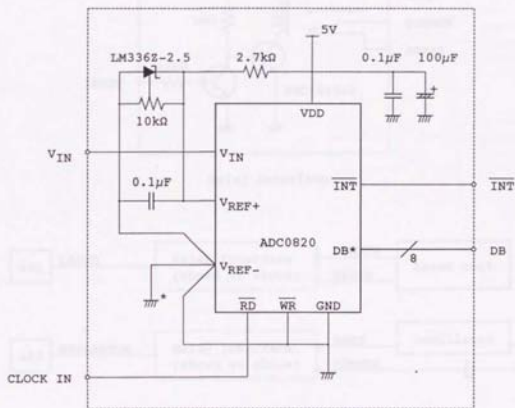
Connection



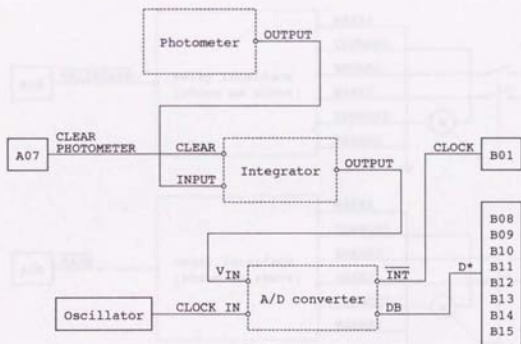
Photometer



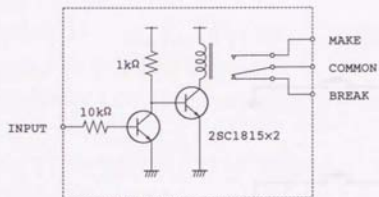
Integrator



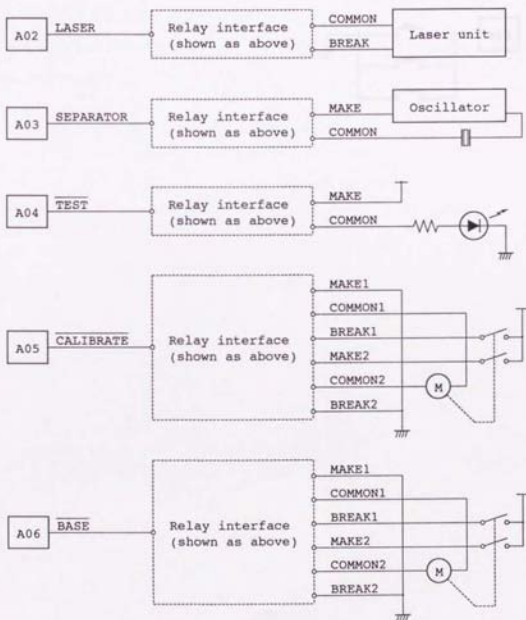
A/D converter



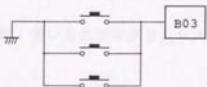
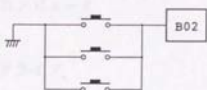
Connection



Relay interface



Connection



プログラム説明書・リスト

PFシステムのソフトウェアは、TurboPASCAL^(R)言語（バージョン6.0）で書かれたプログラムで構成されている。

PFシステムを動作させるには、次のハードウェア・ソフトウェアを用意する必要がある。

- ・PC-9801^(R)シリーズのパーソナルコンピュータ
- ・640Kバイトのメインメモリ
- ・2台以上の2HDフロッピーディスクドライブ
- ・PC-9801^(R)シリーズ専用高解像度カラーディスプレイ
- ・TurboPASCAL^(R)コンパイラ

PFシステムのプログラムについて、その機能・使い方とプログラムリストを次ページ以下に示す。

- ・ PFFORMAT データディスクフォーマットプログラム
 機能：新しいフロッピーディスクをPFシステムで使用できるようにフォーマットする。
 使い方：PFFORMAT <disk drive>
 <disk drive> = {A: B: C: D: E: F: G: H:}
- ・ PFCOPY データディスクコピープログラム
 機能：PFデータディスクを別のフロッピーディスクにコピーする。
 使い方：PFCOPY <master disk drive> <new disk drive>
 <master disk drive> = {A: B: C: D: E: F: G: H:}
 <new disk drive> = {A: B: C: D: E: F: G: H:}
- ・ PFMODEL データ作成プログラムフレームワーク
 機能：PFアークを作成する。
 使い方：PFMODEL <disk drive>
 <disk drive> = {A: B: C: D: E: F: G: H:}
- ・ PFNOTE データ見出し書き換えプログラム
 機能：PFデータディスクの見出し文字列を書き換える。
 使い方：PFNOTE <disk drive>
 <disk drive> = {A: B: C: D: E: F: G: H:}
- ・ PFCHECK データチェックプログラム
 機能：PFデータディスクの内容を見る。
 使い方：PFCHECK <disk drive>
 <disk drive> = {A: B: C: D: E: F: G: H:}
- ・ PFMAKE 加工装置制御プログラム
 機能：加工装置を制御する。
 使い方：PFMAKE [<configuration> ...]
 <configuration> = {configuration strings}
- ・ PFTESTE 加工装置露光系調整用プログラム
 機能：加工装置の露光系を調整するための信号を発生する。
 使い方：PFTESTE <cycle time> <exposure time> [<total time>]
 <cycle time> = {0000000001 .. 2147483647}
 <exposure time> = {0000000001 .. 2147483647}
 <total time> = {0000000001 .. 2147483647}
- ・ PFTESTS 加工装置走査系調整用プログラム
 機能：加工装置の走査系を調整するための信号を発生する。
 使い方：PFTESTS <axis / dir> <number of steps>
 <relaxation time> <step time>
 <axis / dir> = {X+ X- Y+ Y- Z+ Z-}
 <number of steps> = {001 .. 255}
 <relaxation time> = {0000000001 .. 2147483647}
 <step time> = {0000000001 .. 2147483647}

STDLIB.PAS 2317bytes 1992-12-05 2:00:00
 STDLIB.TPU 1376bytes 1992-12-05 2:00:00

Printed on 1994-09-22

```

unit $DLIB;
(* 標準ライブラリ Version 2.00 1992/12/05 *)
($D+,L-)
interface
function BitMaskP(Value:LongInt):Word;
function BitMaskX(Value:LongInt):Word;
procedure Delay(Time:LongInt);
function UpCaseStr(S:istring):istring;
function J18ToJIS(LCharCode:Word):Word;
function J21ToJIS(LCharCode:Word):Word;
implementation
function BitMaskP:
const BitMaskTable:array[0..255] of Word=
($001,$0002,$0004,$0008,$0010,$0020,$0040,$0080,
$100,$200,$400,$800,$1000,$2000,$4000,$8000);
var ValueByte:Byte absolute Value;
begin
  BitMaskP:=BitMaskTable[ValueByte and 255];
end;
function BitMaskX:
const BitMaskTable:array[0..255] of Word=
($FFF,$FFF0,$FFF8,$FFF7,$FFF3,$FFF2,$FFF1,$FFF0,
$FFF,$FFF,$FFF,$FFF,$FFF,$FFF,$FFF,$FFF);
var ValueByte:Byte absolute Value;
begin
  BitMaskX:=BitMaskTable[ValueByte and 255];
end;
procedure Delay:
var Count:LongInt;
begin
  for Count:=1 to Time do;
  end;
function UpCaseStr(S:istring):istring;
var P:Byte;
    AttrCode:array[0..255] of Byte;
begin
  P:=0;
  repeat
    Inc(P);
    if ((Ord(S[P])>=65) and (Ord(S[P])<=90)) or
        ((Ord(S[P])>=97) and (Ord(S[P])<=122)) then
      begin
        AttrCode[P]:=65;
        Inc(P);
        AttrCode[P]:=65;
      end
    else
      AttrCode[P]:=97;
  until P=Length(S);
  UpCaseStr[0]:=0;
  for P:=1 to Length(S) do
    if (AttrCode[P]=65) and (S[P]='a') and (S[P]<<'A') then
      UpCaseStr[P]:=Chr(Ord(S[P])-32)
    else
      UpCaseStr[P]:= S[P];
  end;
function J18ToJIS(LCharCode:Word):Word;
begin
  case
    MOV AX,LCharCode
    ROR AX,1
    JC 001
    ADD AX,$7070E
    JMP 001
    001:
    ADD AX,$711F0
    INC AL
    JE 002
    DEC AX
    002:
    CMP AX,$0FF0
    JBE 003
    ADD AX,$4000
    003:
    MOV @Result,AX
  end;
function J21ToJIS(LCharCode:Word):Word;
begin
  case
    MOV AX,LCharCode
    ROL AX,1
    ADC AX,$70E
    ROR AL,$008
    ADC AL,$410
    JNB 001
    ADD AX,$00A20
    001:
    ADD AX,$10FF0
    MOV @Result,AX
  end;
end;

```

```

-----
KEYLIB.PAS      1336bytes  1992-12-05  2:00:00
KEYLIB.TPU      816bytes   1992-12-05  2:00:00
-----

```

Printed on 1994-09-02

```

unit KeyLib;
(* キーボードライブラリ Version 1.00 1992/12/05 *)
($D+,L-)

interface

type KeyTableType=array[000..099] of byte;
procedure GetKeyCode(var KeyCode:word);
procedure GetShiftKeyCode(var ShiftKeyCode:byte);
procedure GetKeyTable(var KeyTable:KeyTableType);
function KeyPressed:boolean;
procedure ResetKey;

implementation

procedure GetKeyCode;
begin
  asm
    MOV  AX,0000
    INT  0100
    LBB  BX,KeyCode
    MOV  SP,[BX],AX
  end;
end;

procedure GetShiftKeyCode;
begin
  asm
    MOV  AX,0020
    INT  0100
    LBB  BX,ShiftKeyCode
    MOV  SP,[BX],AL
  end;
end;

procedure GetKeyTable;
begin
  asm
    LBB  BX,KeyTable
    MOV  CL,0100
    $01:
    MOV  AX,0040
    MOV  AL,CL
    STB  AL,0100
    INB  AL
    INT  0100
    MOV  SP,[BX],AX
    INC  BX
    LOOP $01
  end;
end;

function KeyPressed;
begin
  asm
    MOV  AX,0010
    INT  0100
    MOV  AL,0000
    CMP  BX,0000
    JE   $01
    MOV  AL,0010
    $01:
    MOV  $Result,AL
  end;
end;

procedure ResetKey;
begin
  asm
    MOV  AX,00C000
    INT  0210
  end;
end;
end.

```

 CRTLIB.PAS
 CRTLIB.TPU

1344bytes 1992-12-05 2:00:00
 1760bytes 1992-12-05 2:00:00

Printed on 1994-09-02

```

unit CRTLib;
(* CRTライブラリ Version 2.00 1993/12/05 *)
($D-,L-)

interface

type KeyTableType=array[100..255] of Byte;

procedure AssignStdIO;
procedure MoveCursorL(Step:Byte);
procedure MoveCursorR(Step:Byte);
procedure MoveCursorU(Step:Byte);
procedure MoveCursorD(Step:Byte);
procedure ClrEOL;
procedure ClrScr;
procedure CursorOn;
procedure CursorOff;

implementation

procedure AssignStdIO;
begin
  Assign(InPut, '');
  Reset(InPut);
  Assign(OutPut, '');
  Rewrite(OutPut);
end;

procedure MoveCursorL;
var Count:Byte;
begin
  for Count:=1 to Step do
    Write($5B);
end;

procedure MoveCursorR;
var Count:Byte;
begin
  for Count:=1 to Step do
    Write($5C);
end;

procedure MoveCursorU;
var Count:Byte;
begin
  for Count:=1 to Step do
    Write($5A);
end;

procedure MoveCursorD;
var Count:Byte;
begin
  for Count:=1 to Step do
    Write($5B);
end;

procedure ClrEOL;
begin
  Write($0D);
end;

procedure ClrScr;
begin
  Write($1B, 'OK');
end;

procedure CursorOn;
begin
  mov     ah, 01h
  int     010h
end;

procedure CursorOff;
begin
  mov     ah, 01h
  int     010h
end;
end.

```

```

FDDLIB.PAS      2348bytes 1992-12-05 2:00:00
FDDLIB.TPU      1376bytes 1992-12-05 2:00:00

```

Printed on 1994-09-02

```

unit FDDLib;
(* FDDライブラリ Version 2.00 1992/12/05 *)
($D-,L-)

interface

procedure ReadDiskData (DrvCode,WCyl,WOHead,WOWer,LenSecCode:Byte;
  var Buffer; var ErrorCode:Byte);

procedure WriteDiskData (DrvCode,WCyl,WOHead,WOWer,LenSecCode:Byte;
  var Buffer; var ErrorCode:Byte);

procedure Recalibrate (DrvCode:Byte; var ErrorCode:Byte);

procedure SenseDrive (DrvCode:Byte; var ErrorCode:Byte);

procedure FormatTrack (DrvCode,WCyl,WOHead,LenSecCode,FillByte:Byte;
  var ErrorCode:Byte);

implementation

uses DOS;

procedure ReadDiskData;
var LenSec:Word;
    Regs:Register;
begin
  case LenSecCode of
    $0:LenSec:=0128;
    $1:LenSec:=0256;
    $2:LenSec:=0512;
    $3:LenSec:=1024;
  end;
  Regs.A0:=E76;
  Regs.A1:=DrvCode;
  Regs.D0:=LenSec;
  Regs.CH:=LenSecCode;
  Regs.CL:=WCyl;
  Regs.DH:=WOHead;
  Regs.DL:=WOWer;
  Regs.BF:=Reg(Buffer);
  Regs.BF:=Cof(Buffer);
  Intr($1B,Regs);
  ErrorCode:=Regs.AH;
end;

procedure WriteDiskData;
var LenSec:Word;
    Regs:Register;
begin
  case LenSecCode of
    $0:LenSec:=0128;
    $1:LenSec:=0256;
    $2:LenSec:=0512;
    $3:LenSec:=1024;
  end;
  Regs.A0:=E75;
  Regs.A1:=DrvCode;
  Regs.D0:=LenSec;
  Regs.CH:=LenSecCode;
  Regs.CL:=WCyl;
  Regs.DH:=WOHead;
  Regs.DL:=WOWer;
  Regs.BF:=Reg(Buffer);
  Regs.BF:=Cof(Buffer);
  Intr($1B,Regs);
  ErrorCode:=Regs.AH;
end;

procedure Recalibrate;
var Regs:Register;
begin
  Regs.A0:=E07;
  Regs.A1:=DrvCode;
  Intr($1B,Regs);
  ErrorCode:=Regs.AH;
end;

procedure SenseDrive;
var Regs:Register;
begin
  Regs.A0:=E04;
  Regs.A1:=DrvCode;
  Intr($1B,Regs);
  ErrorCode:=Regs.AH;
end;

procedure FormatTrack;
var WOWer:Byte;
    IDBuffer:array[000..255] of Byte;
    Regs:Register;
begin
  for WOWer:=01 to 64 do
  begin
    IDBuffer[(WOWer-1)*4] :=WCyl;
    IDBuffer[(WOWer-1)*4+1] :=WOHead;
    IDBuffer[(WOWer-1)*4+2] :=WOWer;
    IDBuffer[(WOWer-1)*4+3] :=LenSecCode;
  end;
  Regs.A0:=E7D;
  Regs.A1:=DrvCode;
  Regs.BF:=256;
  Regs.CH:=LenSecCode;
  Regs.CL:=WCyl;
  Regs.DH:=WOHead;
  Regs.DL:=FillByte;
  Regs.BF:=Reg(IDBuffer);
  Regs.BF:=Cof(IDBuffer);
  Intr($1B,Regs);
  ErrorCode:=Regs.AH;
end;
end.

```

```

-----
IOLIB.PAS          16168bytes 1992-12-05 2:00:00
IOLIB.TPU          11296bytes 1992-12-05 2:00:00
-----
Printed on 1994-09-02

```

```

unit IOLib;
(* I/Oライブラリ Version 2.00 1992/12/05 *)
($D-,L-)

interface

type ICharType=array[0..1] of Char;
function Hex (Value:LongInt):Char;
function HexLChar(Value:LongInt):string;
procedure WaitKey;
function KeyToByte:
function KeyToDecimal:Byte;
function HexLChar(MemChar:Char):string;
procedure StrIntegerLChar (Value:LongInt;
var S:string);
procedure StrIntegerCharW(Value:LongInt; Width:Byte;
var S:string);
procedure StrHexLChar (Value:Extended;
var S:string);
procedure StrHexLCharW (Value:Extended; Width:Byte;
var S:string);
procedure StrHexLCharWD (Value:Extended; Width,Decimals:Byte;
var S:string);
procedure WriteIntegerLChar (Value:LongInt );
procedure WriteIntegerCharW(Value:LongInt; Width:Byte );
procedure WriteHexLChar (Value:Extended );
procedure WriteHexLCharW (Value:Extended; Width:Byte );
procedure WriteHexLCharWD (Value:Extended; Width,Decimals:Byte);
function InputIntegerLChar (Value:LongInt; Width:Byte;
var ErrorCode:Byte):LongInt;
function InputIntegerCharW(Value:LongInt; Width:Byte;
var ErrorCode:Byte):LongInt;
function InputIntegerCharW(Value:LongInt; Width:Byte;
var ErrorCode:Byte):LongInt;
function InputHexLChar (Value:Extended; Width:Byte;
var ErrorCode:Byte):Extended;
function InputHexLCharW (Value:Extended; Width:Byte;
var ErrorCode:Byte):Extended;
function InputHexLCharWD (Value:Extended; Width,Decimals:Byte;
var ErrorCode:Byte):Extended;
procedure InputString(var S:string);

implementation
uses CRTLib,KeyLib;

function Hex;
const HexTable:array[000..999] of Char='0123456789ABCDEF';
var ValueByte:Byte absolute Value;
begin
Hex:=HexTable[ValueByte and 99];
end;

function HexLChar;
type ICharType=array[0..1] of Char;
const HexCharTable:array[000..999] of ICharType=
([0,1,2,3,4,5,6,7,8,9,
'A','B','C','D','E','F']);
var ValueByte:Byte absolute Value;
begin
HexLChar:=HexCharTable[ValueByte and 99];
end;

```



```

Function InputIntegerCharW;
const NumCharTable:=array[00..12] of Char=' --0123456789';
var F,StatusCode:Byte;
    KeyCode:Word;
    ValCode:Integer;
    S:String;
begin
  Str(ValCode:Width,S);
  WriteIntegerCharW(ValCode,Width);
  P:=Width-1;
  StatusCode:=001;
  CursorOff;
  repeat
    GetKeyCode(KeyCode);
    if (KeyCode=01800) and (P<1) then
      begin
        MoveCursor(2);
        ClrSOL;
        Dec(P);
        Delete(S,P,Length(S)-P+1);
        StatusCode:=001;
      end;
    if (KeyCode=01800) and (P=1) then
      begin
        MoveCursor(2);
        Dec(P);
      end;
    if (KeyCode=03000) and (P=Length(S)) then
      begin
        MoveCursor(2);
        Inc(P);
      end;
    if (Pos(Chr(Lo(KeyCode)),NumCharTable)<0) and (P=Width) then
      begin
        Write(NumChar(Chr(Lo(KeyCode))));
        if P=Length(S) then
          S:=S+' ';
          S[P]:=Chr(Lo(KeyCode));
        Inc(P);
        StatusCode:=001;
      end;
  until KeyCode=01000;
  WriteLn;
  if StatusCode=000 then
    begin
      while S[Length(S)]=' ' do
        Delete(S,Length(S),1);
      Val(S,Value,ValCode);
      if ValCode=0000 then
        begin
          InputIntegerCharW:=Value;
          ErrorCode:=000;
        end
      else
        begin
          InputIntegerCharW:=0;
          ErrorCode:=001;
        end;
    end
  else
    begin
      InputIntegerCharW:=Value;
      ErrorCode:=000;
    end;
end;

```

```

Function InputRealChar;
const NumCharTable:=array[00..15] of Char=' --0123456789e.';
var F,StatusCode:Byte;
    KeyCode:Word;
    ValCode:Integer;
    S:String;
begin
  S:='';
  P:=1;
  StatusCode:=001;
  CursorOff;
  repeat
    GetKeyCode(KeyCode);
    if (KeyCode=02800) and (P<1) then
      begin
        MoveCursor(2);
        ClrSOL;
        Dec(P);
        Delete(S,P,Length(S)-P+1);
        StatusCode:=001;
      end;
    if (KeyCode=02800) and (P=1) then
      begin
        MoveCursor(2);
        Dec(P);
      end;
    if (KeyCode=03000) and (P=Length(S)) then
      begin
        MoveCursor(2);
        Inc(P);
      end;
    if (Pos(Chr(Lo(KeyCode)),NumCharTable)<0) and (P=Width) then
      begin
        Write(NumChar(Chr(Lo(KeyCode))));
        if P=Length(S) then
          S:=S+' ';
          S[P]:=Chr(Lo(KeyCode));
        Inc(P);
        StatusCode:=001;
      end;
  until KeyCode=01000;
  WriteLn;
  if StatusCode=000 then
    begin
      while S[Length(S)]=' ' do
        Delete(S,Length(S),1);
      Val(S,Value,ValCode);
      if ValCode=0000 then
        begin
          InputRealChar:=Value;
          ErrorCode:=000;
        end
      else
        begin
          InputRealChar:=0;
          ErrorCode:=001;
        end;
    end
  else
    begin
      InputRealChar:=Value;
      ErrorCode:=000;
    end;
end;

```

```

function InputRealCharV;
const WmCharTable=array[00..15] of Char* +=-0123456789AB;
var P,StatusCode:Byte;
    KeyCode:Word;
    ValCode:Integer;
    S:String;
begin
  Str(Value,S);
  WriteRealChar(Value);
  Writeln(S);
  P:=Length(S)+1;
  StatusCode:=001;
  CursorOn;
  repeat
    GetKeyCode(KeyCode);
    if (KeyCode=0B08) and (P=1) then
      begin
        MoveCurrent(2);
        ClrEOL;
        Dec(P);
        Delete(S,P,Length(S)-P+1);
        StatusCode:=001;
      end;
    if (KeyCode=0B0D) and (P=1) then
      begin
        MoveCurrent(2);
      end;
    if (KeyCode=0C0D) and (P=Length(S)) then
      begin
        MoveCurrent(2);
      end;
    if (Pos(Chr(Lo(KeyCode)),WmCharTable)=0) and (P=Width) then
      begin
        Write(WmChar(Chr(Lo(KeyCode))));
        if P=Length(S) then
          S:=S+' ';
          S[P]:=Chr(Lo(KeyCode));
          Inc(P);
          StatusCode:=001;
        end;
      until KeyCode=1C0D;
      Writeln;
      if StatusCode=00 then
        begin
          while S[Length(S)]=' ' do
            Delete(S,Length(S),1);
          Val(S.Value,ValCode);
          if ValCode=0000 then
            begin
              InputRealCharV:=Value;
              ErrorCode:=000;
            end
          else
            begin
              InputRealCharV:=0;
              ErrorCode:=001;
            end;
          end;
        else
          begin
            InputRealCharV:=Value;
            ErrorCode:=000;
          end;
        end;
      end;
end;

```

```

function InputRealCharW;
const WmCharTable=array[00..15] of Char* +=-0123456789;
var P,StatusCode:Byte;
    KeyCode:Word;
    ValCode:Integer;
    S:String;
begin
  Str(Value,Width,S);
  WriteRealCharW(Value,Width);
  P:=Width+1;
  StatusCode:=001;
  CursorOn;
  repeat
    GetKeyCode(KeyCode);
    if (KeyCode=0B08) and (P=1) then
      begin
        MoveCurrent(2);
        ClrEOL;
        Dec(P);
        Delete(S,P,Length(S)-P+1);
        StatusCode:=001;
      end;
    if (KeyCode=0B0D) and (P=1) then
      begin
        MoveCurrent(2);
      end;
    if (KeyCode=0C0D) and (P=Length(S)) then
      begin
        MoveCurrent(2);
      end;
    if (Pos(Chr(Lo(KeyCode)),WmCharTable)=0) and (P=Width) then
      begin
        Write(WmChar(Chr(Lo(KeyCode))));
        if P=Length(S) then
          S:=S+' ';
          S[P]:=Chr(Lo(KeyCode));
          Inc(P);
          StatusCode:=001;
        end;
      until KeyCode=1C0D;
      Writeln;
      if StatusCode=00 then
        begin
          while S[Length(S)]=' ' do
            Delete(S,Length(S),1);
          Val(S.Value,ValCode);
          if ValCode=0000 then
            begin
              InputRealCharV:=Value;
              ErrorCode:=000;
            end
          else
            begin
              InputRealCharV:=0;
              ErrorCode:=001;
            end;
          end;
        else
          begin
            InputRealCharV:=Value;
            ErrorCode:=000;
          end;
        end;
      end;
end;

```

```

function InputRealCharWD;
const NumCharTable:=array[0..13] of Char=' +-0123456789';
var P,StatusCode:Byte;
    KeyCode:Word;
    ValCode:Integer;
    S:string;
begin
  Str(Value,Width,Decimal,S);
  WriteRealCharWD(Value,Width,Decimal);
  P:=Width-1;
  StatusCode:=0;
  CursorOn;
  repeat
    GetKeyCode(KeyCode);
    if (KeyCode=#000) and (P<1) then
      begin
        MoveCurrent(2);
        ClrEOL;
        Set(P);
        Delete(S,P,Length(S)-P+1);
        StatusCode:=0;
      end;
    if (KeyCode=#100) and (P<1) then
      begin
        MoveCurrent(2);
        Set(P);
      end;
    if (KeyCode=#1000) and (P<=length(S)) then
      begin
        MoveCurrent(3);
        Set(P);
      end;
    if (Pos(Chr(Lo(KeyCode)),NumCharTable)>0) and (P<=Width) then
      begin
        Write(NumChar(Chr(Lo(KeyCode))));
        if P=length(S) then
          S:=S+' ';
          S[P]:=Chr(Lo(KeyCode));
        Set(P);
        StatusCode:=0;
      end;
  until KeyCode=#1000;
  WriteLn;
  if StatusCode=#00 then
    begin
      while S[length(S)]=' ' do
        Delete(S,length(S),1);
      Val(S,Value,ValCode);
      if ValCode=#0000 then
        begin
          InputRealCharWD:=Value;
          ErrorCode:=0;
        end
      else
        begin
          InputRealCharWD:=0;
          ErrorCode:=0;
        end;
    end
  else
    begin
      InputRealCharWD:=Value;
      ErrorCode:=0;
    end;
end;

```

```

procedure InputString;
var P:Byte;
    KeyCode:Word;
    AttrCode:array[001..255] of Byte;
begin
  P:=0;
  repeat
    Inc(P);
    if ((Ord(H[P])<=80) and (Ord(H[P])<=8FF)) or
       ((Ord(H[P])<=80) and (Ord(H[P])<=8FF)) then
      begin
        AttrCode[P]:=801;
        Inc(P);
        AttrCode[P]:=802;
      end
    else
      AttrCode[P]:=800;
  until P=length(S);
  Write(S);
  MoveCursor(L,length(S));
  P:=1;
  CursorOn;
  repeat
    GetKeyCode(KeyCode);
    case KeyCode of
      $3800:if P=1 then
        IF AttrCode[P-1]=802 then
          begin
            MoveCursor(2);
            Del(P,2);
          end
        else
          begin
            MoveCursor(1);
            Del(P);
          end
        $3000:if P=length(S) then
          if AttrCode[P]=801 then
            begin
              MoveCursor(2);
              Del(P,2);
            end
          else
            begin
              MoveCursor(1);
              Del(P);
            end
          else
            IF Lo(KeyCode)=800 then
              begin
                IF (not (((Lo(KeyCode)=802) and (Lo(KeyCode)=8FF)) or
                       ((Lo(KeyCode)=800) and (Lo(KeyCode)=8FF)))) and
                   (P=length(S))) then
                  begin
                    S[P]:=Chr(Lo(KeyCode));
                    AttrCode[P]:=800;
                    Write(S[P]);
                    Inc(P);
                    IF AttrCode[P]=802 then
                      begin
                        S[P]:=' ';
                        AttrCode[P]:=800;
                        Write(S[P]);
                        MoveCursor(L);
                      end;
                    end;
                    IF ((Lo(KeyCode)=800) and (Lo(KeyCode)=8FF)) or
                       ((Lo(KeyCode)=800) and (Lo(KeyCode)=8FF))) and
                       (P=length(S)-1)) then
                      begin
                        S[P]:=Chr(Lo(KeyCode));
                        AttrCode[P]:=801;
                        Write(S[P]);
                        Inc(P);
                        GetKeyCode(KeyCode);
                        S[P]:=Chr(Lo(KeyCode));
                        AttrCode[P]:=802;
                        Write(S[P]);
                        Inc(P);
                        IF AttrCode[P]=802 then
                          begin
                            S[P]:=' ';
                            AttrCode[P]:=800;
                            Write(S[P]);
                            MoveCursor(L);
                          end;
                        end;
                      end;
                    end;
                    until KeyCode=8100;
                    WriteLn;
                    end;
  end;

```

MGRAPH.PAS
MGRAPH.TPU

2977bytes 1993-05-20 2:00:00
1632bytes 1993-05-20 2:00:00

Printed on 1994-09-02

```

unit MGraph;
(* ミニグラフィックスライブラリ Version 2.00 1993/05/20 *)
($D-L-)

interface

procedure InitGraph;
procedure GraphOn;
procedure GraphOff;
procedure PutPixel(X,Y:Word; MC:Color;Byte);
procedure GetPixel(X,Y:Word; var MC:Color;Byte);
procedure SetPalette(ColorCode0,ColorCode1,ColorCode2,ColorCode3,
ColorCode4,ColorCode5,ColorCode6,ColorCode7:Byte);
procedure ClearGraph;

implementation

uses StdLib;

procedure InitGraph;
begin
asm
mov ax,0410
mov cx,0C0B
int 010
end;
end;

procedure GraphOn;
begin
asm
mov ax,0410
int 010
end;
end;

procedure GraphOff;
begin
asm
mov ax,0410
int 010
end;
end;

procedure PutPixel;
var VYMMAddr,VYMMMask,VYMMMaskN:Word;
var VYMM:array[0000..03FF] of Word absolute $A00:0000;
VYMM:array[0000..03FF] of Word absolute $B00:0000;
VYMM:array[0000..03FF] of Word absolute $C00:0000;
begin
VYMMAddr:=(X shr 4)+(Y shl 5)+(Y shl 3);
VYMMMask:=neg(BIYMMASK[not X]);
VYMMMaskN:=neg(BIYMMASK[not X]);
if (MC:Color and $01)=$00 then VYMM(VYMMAddr):=VYMMMask of VYMM(VYMMAddr);
else VYMM(VYMMAddr):=VYMMMaskN and VYMM(VYMMAddr);
if (MC:Color and $02)=$00 then VYMM(VYMMAddr):=VYMMMask of VYMM(VYMMAddr);
else VYMM(VYMMAddr):=VYMMMaskN and VYMM(VYMMAddr);
if (MC:Color and $04)=$00 then VYMM(VYMMAddr):=VYMMMask of VYMM(VYMMAddr);
else VYMM(VYMMAddr):=VYMMMaskN and VYMM(VYMMAddr);
end;

procedure GetPixel;
var VYMMAddr,VYMMMask:Word;
var VYMM:array[0000..03FF] of Word absolute $A00:0000;
VYMM:array[0000..03FF] of Word absolute $B00:0000;
VYMM:array[0000..03FF] of Word absolute $C00:0000;
begin
VYMMAddr:=(X shr 4)+(Y shl 5)+(Y shl 3);
VYMMMask:=neg(BIYMMASK[not X]);
if (VYMMMask and VYMM(VYMMAddr))=$00 then MC:Color:=MC:Color or $01
else MC:Color:=MC:Color and $FF;
if (VYMMMask and VYMM(VYMMAddr))=$00 then MC:Color:=MC:Color or $02
else MC:Color:=MC:Color and $FD;
if (VYMMMask and VYMM(VYMMAddr))=$00 then MC:Color:=MC:Color or $04
else MC:Color:=MC:Color and $F4;
MC:Color:=MC:Color and $07;
end;

procedure SetPalette;
begin
for [IAR]:=((ColorCode7 and $07) to
((ColorCode0 and $07) shl 4);
for [IAA]:=((ColorCode4 and $07) to
((ColorCode1 and $07) shl 4);
for [IAC]:=((ColorCode4 and $07) to
((ColorCode2 and $07) shl 4);
for [IAR]:=((ColorCode4 and $07) to
((ColorCode4 and $07) shl 4);
end;

procedure ClearGraph;
var VYMMFtr:Byte absolute $A00:0000;
VYMMFtr:Byte absolute $B00:0000;
VYMMFtr:Byte absolute $C00:0000;
begin
FillChar(VYMMFtr,$0000,$00);
FillChar(VYMMFtr,$0000,$00);
FillChar(VYMMFtr,$0000,$00);
end;
end.

```

```
-----
PPDATLIB.PAS      12848bytes 1993-08-24 2:00:00
PPDATLIB.TPU      7936bytes  1993-08-24 2:00:00
-----
```

Printed on 1994-09-02

```
unit PPDatLib;
(* PFSシステムソフトウェア
 * データ操作ルーチンライブラリ Version 2.00 1993/08/24 *)
($D-L)

interface

Type CubeType=array[0..3,0..3,0..3] of Byte;
LayerType=array[00..39,000..639] of Word;
RecBufType=array[0000..1023] of Byte;

var DrvCode:Byte;
    PFDatStatusCode:Byte;

procedure ReadRec (MRec:Word; var ResultCode:Byte);
procedure WriteRec (MRec:Word; var ResultCode:Byte);
procedure ReadBit (BitAddr:LongInt; var DataByte:Byte; var ResultCode:Byte);
procedure WriteBit (BitAddr:LongInt; DataByte:Byte; var ResultCode:Byte);
procedure OpenDisk (var ResultCode:Byte);
procedure CloseDisk (var ResultCode:Byte);
procedure ReadDiskCode (var DiskCode:Word; var ResultCode:Byte);
procedure WriteDiskCode (DiskCode:Word; var ResultCode:Byte);
procedure ReadDiskK (var DiskWord; var ResultCode:Byte);
procedure WriteDiskK (DiskWord; var ResultCode:Byte);
procedure ReadDiskT (var DiskWord; var ResultCode:Byte);
procedure WriteDiskT (DiskWord; var ResultCode:Byte);
procedure ReadDiskI (var DiskWord; var ResultCode:Byte);
procedure WriteDiskI (DiskWord; var ResultCode:Byte);
procedure ReadDiskE (var Note:string; var ResultCode:Byte);
procedure WriteDiskE (Note:string; var ResultCode:Byte);
procedure ReadContents (BaseI:Word; BitAddr, BitLength:LongInt;
    var ResultCode:Byte);
procedure WriteContents (BaseI:Word; BitAddr, BitLength:LongInt;
    var ResultCode:Byte);
procedure DiskToCube (var BitAddr:LongInt; var Cube:CubeType;
    var ResultCode:Byte);
procedure CubeToDisk (var BitAddr:LongInt; Cube:CubeType;
    var ResultCode:Byte);
procedure DiskToLayer (I:DiskK, SizeT:Word; var Layer:LayerType;
    var ResultCode:Byte);

implementation

uses StdLib, PFDLib;

var MRec:Word;
    RecBuf:RecBufType absolute $FFC0:0000;

function MOCylMRec (MRec:Word):Word;
begin
    MOCylMRec:=MRec div 16;
end;

function M0HeadMRec (MRec:Word):Word;
begin
    M0HeadMRec:=(MRec div 8) mod 2;
end;

function M0SecMRec (MRec:Word):Word;
begin
    M0SecMRec:=(MRec mod 8)-1;
end;
```

```

procedure ReadRec;
begin
  WriteDataData (DirCode, MOY1(MORec), MOY2(MORec), MOY3(MORec),
    ... 503, RecBuf, ResultCode);
  if ResultCode=00 then
  begin
    FFDatStatusCode:=FFDatStatusCode and 5FC;
    MORecC:=MORec
  end
  else
  begin
    FFDatStatusCode:=FFDatStatusCode or 503;
  end;
end;

```

```

procedure WriteRec;
begin
  WriteDataData (DirCode, MOY1(MORec), MOY2(MORec), MOY3(MORec),
    ... 503, RecBuf, ResultCode);
  if ResultCode=00 then
  begin
    FFDatStatusCode:=FFDatStatusCode and 5FA;
    MORecC:=MORec
  end
  else
  begin
    FFDatStatusCode:=FFDatStatusCode or 503;
  end;
end;

```

```

procedure CallRec(MORec:Word; var ResultCode:Byte);

```

```

begin
  if MORecC=MORec then
  begin
    if (FFDatStatusCode and 501)=01 then
      WriteRec(MORecC, ResultCode);
    if ResultCode=00 then
      Exit;
    ReadRec(MORec, ResultCode);
  end
  else
    ResultCode:=501;
end;

```

```

procedure ReadBit;

```

```

  var MORec:Word;
begin
  MORec:=(BIAddr div 8192)+00004;
  if MORec=1278 then
  begin
    ResultCode:=500;
    Exit;
  end;
  CallRec(MORec, ResultCode);
  if ResultCode=00 then
    Exit;
  if (RecBuf[(BIAddr mod 8192) div 8] and BitMaskP[(BIAddr mod 8)])=000 then
    DataByte:=500
  else
    DataByte:=501;
  ResultCode:=500;
end;

```

```

procedure WriteBit;

```

```

  var MORec:Word;
begin
  MORec:=(BIAddr div 8192)+00004;
  if MORec=1278 then
  begin
    ResultCode:=500;
    Exit;
  end;
  CallRec(MORec, ResultCode);
  if ResultCode=00 then
    Exit;
  if Odd(DataByte) then
    RecBuf[(BIAddr mod 8192) div 8]:=RecBuf[(BIAddr mod 8192) div 8] or
      BitMaskP[(BIAddr mod 8)]
  else
    RecBuf[(BIAddr mod 8192) div 8]:=RecBuf[(BIAddr mod 8192) div 8] and
      BitMaskN[(BIAddr mod 8)];
  FFDatStatusCode:=FFDatStatusCode or 501;
  ResultCode:=500;
end;

```

```

procedure OpenDisk;

```

```

begin
  FFDatStatusCode:=503;
  ReadRec(01278, ResultCode);
end;

```

```

procedure CloseDisk;

```

```

begin
  if (FFDatStatusCode and 501)=01 then
    WriteRec(MORecC, ResultCode);
end;

```



```

procedure ReadDlakCode;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  Mem[Seg(DlakCode)+Ofs(DlakCode) ]:=MemBuf[0014];
  Mem[Seg(DlakCode)+Ofs(DlakCode)-1]:=MemBuf[0017];
  ResultCode:=000;
end;

procedure WriteDlakCode;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  MemBuf[0014]:=Mem[Seg(DlakCode)+Ofs(DlakCode) ];
  MemBuf[0017]:=Mem[Seg(DlakCode)+Ofs(DlakCode)-1];
  PFDatStatusCode:=PFDatStatusCode of $01;
  ResultCode:=000;
end;

procedure ReadSiswK;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  Mem[Seg(SiswK)+Ofs(SiswK) ]:=MemBuf[0018];
  Mem[Seg(SiswK)+Ofs(SiswK)-1]:=MemBuf[0019];
  ResultCode:=000;
end;

procedure WriteSiswK;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  MemBuf[0018]:=Lo(SiswK);
  MemBuf[0019]:=Hi(SiswK);
  PFDatStatusCode:=PFDatStatusCode of $01;
  ResultCode:=000;
end;

procedure ReadSiswY;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  Mem[Seg(SiswY)+Ofs(SiswY) ]:=MemBuf[0020];
  Mem[Seg(SiswY)+Ofs(SiswY)-1]:=MemBuf[0021];
  ResultCode:=000;
end;

procedure WriteSiswY;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  MemBuf[0020]:=Lo(SiswY);
  MemBuf[0021]:=Hi(SiswY);
  ResultCode:=000;
end;

procedure ReadSiswZ;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  Mem[Seg(SiswZ)+Ofs(SiswZ) ]:=MemBuf[0022];
  Mem[Seg(SiswZ)+Ofs(SiswZ)-1]:=MemBuf[0023];
  ResultCode:=000;
end;

procedure WriteSiswZ;
begin
  CallProc(D1278, ResultCode);
  if ResultCode=000 then
    Exit;
  MemBuf[0022]:=Lo(SiswZ);
  MemBuf[0023]:=Hi(SiswZ);
  PFDatStatusCode:=PFDatStatusCode of $01;
  ResultCode:=000;
end;

```

```

procedure ReadNote;
var HChar:Word;
begin
  CallNcr(01278,ResultCode);
  if ResultCode=000 then
    exit;
  for HChar:=01 to 72 do
    Note[HChar]:=Chr(ResultCode[HChar+001]);
  Note[00]:=Chr(72);
  ResultCode:=000;
end;

procedure WriteNote;
var HChar:Word;
begin
  CallNcr(01278,ResultCode);
  if ResultCode=000 then
    exit;
  for HChar:=01 to 72 do
    if HChar<<Length(Note) then Result[HChar+001]:=Ord(Note[HChar])
    else Result[HChar+001]:=020;
  ResultCode:=000;
end;

procedure ReadContents;
var HChar:Word;
begin
  HChar:=(Base+16) div 128-01278;
  CallNcr(HChar,ResultCode);
  if ResultCode=000 then
    exit;
  Mem[Seg(BIAddr) +Ofs(BIAddr) ] :=Result[(Base+16) mod 128]*8 ;
  Mem[Seg(BIAddr) +Ofs(BIAddr) +1] :=Result[(Base+16) mod 128]*8+0001;
  Mem[Seg(BIAddr) +Ofs(BIAddr) +2] :=Result[(Base+16) mod 128]*8+0002;
  Mem[Seg(BIAddr) +Ofs(BIAddr) +3] :=Result[(Base+16) mod 128]*8+0003;
  Mem[Seg(BILength)+Ofs(BILength) ] :=Result[(Base+16) mod 128]*8+0004;
  Mem[Seg(BILength)+Ofs(BILength)+1] :=Result[(Base+16) mod 128]*8+0005;
  Mem[Seg(BILength)+Ofs(BILength)+2] :=Result[(Base+16) mod 128]*8+0006;
  Mem[Seg(BILength)+Ofs(BILength)+3] :=Result[(Base+16) mod 128]*8+0007;
  ResultCode:=000;
end;

procedure WriteContents;
var HChar:Word;
begin
  HChar:=(Base+16) div 128-01278;
  CallNcr(HChar,ResultCode);
  if ResultCode=000 then
    exit;
  Result[(Base+16) mod 128]*8 :=Mem[Seg(BIAddr) +Ofs(BIAddr) ];
  Result[(Base+16) mod 128]*8+0001 :=Mem[Seg(BIAddr) +Ofs(BIAddr) +1];
  Result[(Base+16) mod 128]*8+0002 :=Mem[Seg(BIAddr) +Ofs(BIAddr) +2];
  Result[(Base+16) mod 128]*8+0003 :=Mem[Seg(BIAddr) +Ofs(BIAddr) +3];
  Result[(Base+16) mod 128]*8+0004 :=Mem[Seg(BILength) +Ofs(BILength) ];
  Result[(Base+16) mod 128]*8+0005 :=Mem[Seg(BILength) +Ofs(BILength)+1];
  Result[(Base+16) mod 128]*8+0006 :=Mem[Seg(BILength) +Ofs(BILength)+2];
  Result[(Base+16) mod 128]*8+0007 :=Mem[Seg(BILength) +Ofs(BILength)+3];
  ResultCode:=000;
end;

```

```

procedure DiaktOCube;
var BitData, CubeByte: byte;
    PivotX, PivotY, PivotZ, OffX, OffY, OffZ: byte;
begin
  ReadBit(BitAddr, BitData, ResultCode);
  if ResultCode=00 then
    Exit;
  Inc(BitAddr);
  case BitData of
    $00:begin
      ReadBit(BitAddr, BitData, ResultCode);
      if ResultCode=00 then
        Exit;
      Inc(BitAddr);
      case BitData of
        $00:CubeByte:=001;
        $01:CubeByte:=002;
      end;
      for OffX:=0 to 3 do
        for OffY:=0 to 3 do
          for OffZ:=0 to 3 do
            Cube[OffX, OffY, OffZ]:=CubeByte;
          end;
        end;
      end;
    end;
    $01:begin
      for PivotZ:=0 to 1 do
        for PivotY:=0 to 1 do
          for PivotX:=0 to 1 do
            begin
              ReadBit(BitAddr, BitData, ResultCode);
              if ResultCode=00 then
                Exit;
              Inc(BitAddr);
              case BitData of
                $00:CubeByte:=001;
                $01:CubeByte:=002;
              end;
              for OffX:=0 to 1 do
                for OffY:=0 to 1 do
                  for OffZ:=0 to 1 do
                    Cube[PivotX*2+OffX, PivotY*2+OffY, PivotZ*2+OffZ]:=
                      CubeByte;
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
    $02:begin
      for OffX:=0 to 1 do
        for OffY:=0 to 1 do
          for OffZ:=0 to 1 do
            begin
              ReadBit(BitAddr, BitData, ResultCode);
              if ResultCode=00 then
                Exit;
              Inc(BitAddr);
              case BitData of
                $00:CubeByte:=001;
                $01:CubeByte:=002;
              end;
              Cube[PivotX*2+OffX, PivotY*2+OffY, PivotZ*2+OffZ]:=
                CubeByte;
            end;
          end;
        end;
      end;
    end;
  end;
end;
end;
end;
end;

```

```

procedure CubeToDisk;
var BitData, CubeByte: byte;
    PivotX, PivotY, PivotZ, OffX, OffY, OffZ: byte;
begin
  CubeByte:=00;
  for OffZ:=0 to 3 do
  for OffY:=0 to 3 do
  for OffX:=0 to 3 do
    CubeByte:=CubeByte or Cube[OffX,OffY,OffZ];
  if CubeByte=00 then
  begin
    WriteLn(BitAddr, $0, ResultCode);
    if ResultCode=00 then
      Exit;
    Inc(BitAddr);
    case CubeByte of
      $01:BitData:=00;
      $02:BitData:=01;
    end;
    WriteLn(BitAddr, BitData, ResultCode);
    if ResultCode=00 then
      Exit;
    Inc(BitAddr);
  end
  else
  begin
    WriteLn(BitAddr, $01, ResultCode);
    if ResultCode=00 then
      Exit;
    Inc(BitAddr);
    for PivotX:=0 to 1 do
    for PivotY:=0 to 1 do
    for PivotZ:=0 to 1 do
      begin
        CubeByte:=00;
        for OffZ:=PivotZ*2 to PivotZ*2+1 do
        for OffY:=PivotY*2 to PivotY*2+1 do
        for OffX:=PivotX*2 to PivotX*2+1 do
          CubeByte:=CubeByte or Cube[OffX,OffY,OffZ];
        if CubeByte=00 then
          begin
            WriteLn(BitAddr, $00, ResultCode);
            if ResultCode=00 then
              Exit;
            Inc(BitAddr);
            case CubeByte of
              $01:BitData:=00;
              $02:BitData:=01;
            end;
            WriteLn(BitAddr, BitData, ResultCode);
            if ResultCode=00 then
              Exit;
            Inc(BitAddr);
          end
        else
          begin
            WriteLn(BitAddr, $01, ResultCode);
            if ResultCode=00 then
              Exit;
            Inc(BitAddr);
            for OffZ:=PivotZ*2 to PivotZ*2+1 do
            for OffY:=PivotY*2 to PivotY*2+1 do
            for OffX:=PivotX*2 to PivotX*2+1 do
              begin
                case Cube[OffX,OffY,OffZ] of
                  $01:BitData:=00;
                  $02:BitData:=01;
                end;
                WriteLn(BitAddr, BitData, ResultCode);
                if ResultCode=00 then
                  Exit;
                Inc(BitAddr);
              end
            end;
          end;
        end;
      end;
    end;
  end;
end;
end;
end;

```

```

procedure BitToLayer;
var BaseL, BaseY, BaseZ, OffX, OffY, OffZ, X, Y, Word;
    BitAddr, BitLength, Length;
    Cube: cubeType;
begin
  BaseL:=2 div 4;
  BaseYContent:=BaseL, BitAddr, BitLength, ResultCode);
  if ResultCode=00 then
    Exit;
  for BaseY:=00 to (SizeY div 4)-1 do
  for BaseZ:=00 to (SizeZ div 4)-1 do
    begin
      BitAddr:=Cube[BitAddr, Cube, ResultCode);
      if ResultCode=00 then
        Exit;
      OffZ:=mod 4;
      for OffY:=0 to 3 do
      for OffX:=0 to 3 do
        begin
          X:=BaseX*OffX;
          Y:=BaseY*OffY;
          case Cube[OffX,OffY,OffZ] of
            $01:Layer[X div 16, Y]:=Layer[X div 16, Y] and BitMask(X);
            $02:Layer[X div 16, Y]:=Layer[X div 16, Y] or BitMask(X);
          end;
        end;
      end;
    end;
  end;
  begin
    HeapOut:=Pir($P00, 60000);
  end;

```

```

-----
PFDEVLIB.FAS      3791bytes  1993-08-24  2:00:00
PFDEVLIB.TPU      3104bytes  1993-08-24  2:00:00
-----

```

Printed on 1994-09-02

```

unit PFDevLib;
(* P.Fシステムソフトウェア
 * 加工装置操作ルーチンライブラリ Version 2.00 1993/08/24 *)
($D-L)

interface

procedure InitPort;

procedure ExposureOn;
procedure ExposureOff;
procedure Exposure(ST:LongInt);
procedure LaserOn;
procedure LaserOff;
procedure SeparatorOn;
procedure SeparatorOff;
procedure SetTestSign;
procedure ResetTestSign;
procedure SetCellIdSign;
procedure ResetCellIdSign;
procedure SetBaseIdSign;
procedure ResetBaseIdSign;
procedure MoveInFX(var FX:Word; FMin,FMax:Word; ST:LongInt);
procedure MoveOutFX(var FX:Word; FMin,FMax:Word; ST:LongInt);
procedure MoveInFY(var FY:Word; FMin,FMax:Word; ST:LongInt);
procedure MoveOutFY(var FY:Word; FMin,FMax:Word; ST:LongInt);
procedure MoveInFZ(var FZ:Word; FMin,FMax:Word; ST:LongInt);
procedure MoveOutFZ(var FZ:Word; FMin,FMax:Word; ST:LongInt);
function Button.Byte;

implementation
use $D.Lib;
const PortAddr:Word=$00D0;

procedure InitPort;
begin
  PortW[PortAddr+4]:=0000;
  PortW[PortAddr+5]:=FFFF;
end;

procedure ExposureOn;
begin
  PortW[PortAddr]:=PortW[PortAddr] or $0001;
  PortW[PortAddr]:=PortW[PortAddr] and $FFF;
end;

procedure ExposureOff;
begin
  PortW[PortAddr]:=PortW[PortAddr] or $0001;
  PortW[PortAddr]:=PortW[PortAddr] and $FFF;
end;

procedure Exposure;
begin
  ExposureOn;
  Delay(ST);
  ExposureOff;
end;

procedure LaserOn;
begin
  PortW[PortAddr]:=PortW[PortAddr] or $0004;
end;

procedure LaserOff;
begin
  PortW[PortAddr]:=PortW[PortAddr] and $FFF;
end;

procedure SeparatorOn;
begin
  PortW[PortAddr]:=PortW[PortAddr] and $FFF;
end;

procedure SeparatorOff;
begin
  PortW[PortAddr]:=PortW[PortAddr] or $0008;
end;

procedure SetTestSign;
begin
  PortW[PortAddr]:=PortW[PortAddr] and $FFF;
end;

procedure ResetTestSign;
begin
  PortW[PortAddr]:=PortW[PortAddr] or $0010;
end;

```

```

procedure SetCallMign;
begin
  PortW[PortAddr]:=PortW[PortAddr] and $FFF;
end;

procedure ResetCallMign;
begin
  PortW[PortAddr]:=PortW[PortAddr] or $0020;
end;

procedure SetBaseSign;
begin
  PortW[PortAddr]:=PortW[PortAddr] and $FFF;
end;

procedure ResetBaseSign;
begin
  PortW[PortAddr]:=PortW[PortAddr] or $0040;
end;

procedure Move(var F:Word; FMin,FMax:Word; DirCode:Byte; DT:LongInt;
  B1Mask:DirT, B2Mask:DirT, B3Mask:PulseW, B4Mask:PulseW; Word);
begin
  if FMin=0000 then FMin:=F;
  if FMax=0000 then FMax:=F;
  if ((DirCode=00) and (F=FMax)) or
    ((DirCode=01) and (F=FMin)) then
    Bit;
  PortW[PortAddr]:=PortW[PortAddr] or B1Mask:DirT;
  PortW[PortAddr]:=PortW[PortAddr] and B2Mask:DirT;
  PortW[PortAddr]:=PortW[PortAddr] or B3Mask:PulseW;
  Delay(DT);
  PortW[PortAddr]:=PortW[PortAddr] and B4Mask:PulseW;
  Delay(DT);
  if DirCode=00 then Inc(F);
  if DirCode=01 then Dec(F);
end;

procedure MoveIncFX;
begin
  Move(FX, FMinX, FMaxX, 00, DT, 0000, $FFFF, 0200, $DFFF);
end;

procedure MoveDecFX;
begin
  Move(FX, FMinX, FMaxX, 01, DT, 0000, $FFFF, 0200, $DFFF);
end;

procedure MoveIncFY;
begin
  Move(FY, FMinY, FMaxY, 00, DT, 0400, $FFFF, 0400, $7FFF);
end;

procedure MoveDecFY;
begin
  Move(FY, FMinY, FMaxY, 01, DT, 0000, $8FFF, 0400, $7FFF);
end;

procedure MoveIncFZ;
begin
  Move(FZ, FMinZ, FMaxZ, 00, DT, 8100, $FFFF, 8200, $DFFF);
end;

procedure MoveDecFZ;
begin
  Move(FZ, FMinZ, FMaxZ, 01, DT, 0000, $8FFF, 8200, $DFFF);
end;

function Button;
begin
  if (PortW[PortAddr-2] and $0001)=$0001 then Button:=000
  else Button:=001;
end;
end.

```

```

FFSCLIB.FAS      5275bytes 1993-08-24  2:00:00
FFSCLIB.TPU      3248bytes 1993-08-24  2:00:00

```

Printed on 1994-09-03

```

unit FFSCLIB;
(* P Fシステムソフトウェア *)
(* 画面表示ルーチンライブラリ Version 2.00 1993/08/24 *)

($D..L-)

interface
  procedure PutCell(var LayerPtr, DisplayStylePtr: S.T.Word;
                  ColorCodeOut, ColorCodeIn:Byte);
  procedure ShowLayer(var LayerPtr, DisplayStylePtr: S.T.Word;
                    SizeX, SizeY:Word);
  procedure RepaintLayer;
  procedure EraseLayer;
implementation
uses StdLib, WinGraph;
procedure PutCell;
var LayerType:=array[00..39,000..639] of Word;
    var StatusCode:Byte;
        XMin, XMax, YMin, YMax, SC, YC:Word;
        SS, ST:Integer;
        DisplayStyle:=string[1] absolute DisplayStylePtr;
        Layer:=LayerType absolute LayerPtr;
begin
  if DisplayStyle='01' then
  begin
    SC:= 384*X;
    ST:= 347-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='02' then
  begin
    SC:= 192*X;
    ST:= 347-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='03' then
  begin
    SC:=  X;
    ST:= 347-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='04' then
  begin
    SC:= 384*X;
    ST:= 519-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='05' then
  begin
    SC:= 192*X;
    ST:= 519-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='06' then
  begin
    SC:=  X;
    ST:= 519-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='07' then
  begin
    SC:= 384*X;
    ST:= 751-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='08' then
  begin
    SC:= 192*X;
    ST:= 751-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
  if DisplayStyle='09' then
  begin
    SC:=  X;
    ST:= 751-Y;
    XMin:=X;
    XMax:=X;
    YMin:=Y;
    YMax:=Y;
  end else
end
end

```

```

IF DisplayStyle='K1' then
begin
  SX:=384*(X div 2);
  SY:=387*(Y div 2);
  SMX:=(X div 2)*2;
  SMY:=(Y div 2)*2;
  SMXa:=(X div 2)*2;
  SMYa:=(Y div 2)*2;
end else
IF DisplayStyle='K3' then
begin
  SX:=320*(X div 2);
  SY:=367*(Y div 2);
  SMX:=(X div 2)*2;
  SMY:=(Y div 2)*2;
  SMXa:=(X div 2)*2;
  SMYa:=(Y div 2)*2;
end else
IF DisplayStyle='K9' then
begin
  SX:=320*(X div 2);
  SY:=411*(Y div 2);
  SMX:=(X div 2)*2;
  SMY:=(Y div 2)*2;
  SMXa:=(X div 2)*2;
  SMYa:=(Y div 2)*2;
end else
IF DisplayStyle='H1' then
begin
  SX:=384*(X div 2);
  SY:=431*(Y div 2);
  SMX:=(X div 2)*2;
  SMY:=(Y div 2)*2;
  SMXa:=(X div 2)*2;
  SMYa:=(Y div 2)*2;
end else
IF DisplayStyle='Q' then
begin
  SX:=432*(X div 4);
  SY:=319*(Y div 4);
  SMX:=(X div 4)*4;
  SMY:=(Y div 4)*4;
  SMXa:=(X div 4)*4;
  SMYa:=(Y div 4)*4;
end else
IF DisplayStyle='P' then
begin
  case (X mod 8) of
    0,1,2 :begin
      SX:=392*(X div 8)*3;
      SMX:=(X div 8)*8;
      SMXa:=(X div 8)*8*2;
      end;
    3,4 :begin
      SX:=393*(X div 8)*3;
      SMX:=(X div 8)*8*3;
      SMXa:=(X div 8)*8*4;
      end;
    5,6,7 :begin
      SX:=394*(X div 8)*3;
      SMX:=(X div 8)*8*5;
      SMXa:=(X div 8)*8*7;
      end;
  end;
  case (Y mod 8) of
    0,1,2 :begin
      SY:=360*(Y div 8)*3;
      SMY:=(Y div 8)*8;
      SMYa:=(Y div 8)*8*2;
      end;
    3,4 :begin
      SY:=359*(Y div 8)*3;
      SMY:=(Y div 8)*8*3;
      SMYa:=(Y div 8)*8*4;
      end;
    5,6,7 :begin
      SY:=358*(Y div 8)*3;
      SMY:=(Y div 8)*8*5;
      SMYa:=(Y div 8)*8*7;
      end;
  end;
end;
IF (SX=384) and (SX=640) and (SY=112) and (SY=368) then
begin
  StatusCode:=800;
  for NC=SMXa to SMXa do
  for NY=SMYa to SMYa do
  IF ((Layer[NC] div 16, (NY)) and BitMask[NC])=80000) then
  IF StatusCode=801;
  PutPixel[IX,SY,ColorCodeOut];
  else
  PutPixel[IX,SY,ColorCodeIn];
end;
end;

```

```

procedure ShowLayer;

```

```

var X,Y:Word;
    SX,SY:Integer;
begin
  SetPalette(0,0,2,3,0,0,0,7);
  for SX=384 to 630 do
  for SY=367 down to 112 do
  PutPixel[IX,SY,801];
  SetPalette(0,1,2,3,1,1,6,7);
  for Y=0 to X div 16 do
  for X=0 to X div 16-1 do
  PutCall(LayerPtr,DisplayStylePtr,X,Y,800,804);
  SetPalette(0,1,2,3,4,5,6,7);
  end;
end;

```

```

procedure RepaintLayer;

```

```

var ColorCode:Byte;
    SX,SY:Integer;
begin
  SetPalette(0,1,2,3,4,5,6,7);
  for SX=387 down to 112 do
  for SX=384 to 630 do
  begin
    SetPixel[IX,SY,ColorCode];
    if ColorCode=804 then
    ColorCode:=804;
    PutPixel[IX,SY,ColorCode];
  end;
  SetPalette(0,1,2,3,4,5,6,7);
  end;
end;

```

```

procedure EraseLayer;

```

```

var SX,SY:Integer;
begin
  SetPalette(0,0,2,3,0,0,0,7);
  for SY=387 down to 112 do
  for SX=384 to 630 do
  PutPixel[IX,SY,800];
  SetPalette(0,1,2,3,4,5,6,7);
  end;
end;
end;

```



```

=====
FFFORMAT.PAS      1825bytes  1993-08-24  2:00:00
FFFORMAT.EXE     4464bytes  1993-08-24  2:00:00
=====

```

```
Printed on 1994-09-02
```

```

program FFFormat;
(* P Fシステムソフトウェア
 * データディスクフォーマットプログラム Version 2.00 1993/08/24 *)
($D-L)

uses $UCLIB,$FSLIB;

type RecBufType=array[0000..1023] of Byte;

var DrvCode,HCyl,MHHead,MHSec,ReturnCode:Byte;
    RecBuf:RecBufType;
    FF:file of RecBufType;

begin
  if UpCaseStr(ParamStr(1))='A:' then DrvCode:=HL(Hsew[0060:$2C80]) else
  if UpCaseStr(ParamStr(1))='B:' then DrvCode:=HL(Hsew[0060:$2C80]) else
  if UpCaseStr(ParamStr(1))='C:' then DrvCode:=HL(Hsew[0060:$2C8A]) else
  if UpCaseStr(ParamStr(1))='D:' then DrvCode:=HL(Hsew[0060:$2C8C]) else
  if UpCaseStr(ParamStr(1))='E:' then DrvCode:=HL(Hsew[0060:$2C8E]) else
  if UpCaseStr(ParamStr(1))='F:' then DrvCode:=HL(Hsew[0060:$2C90]) else
  if UpCaseStr(ParamStr(1))='G:' then DrvCode:=HL(Hsew[0060:$2C92]) else
  if UpCaseStr(ParamStr(1))='H:' then DrvCode:=HL(Hsew[0060:$2C94]) else
  Halt(00);
  HsewDrive(DrvCode,ReturnCode);
  if (ReturnCode=00) or
  (ReturnCode=10) or
  (ReturnCode=11) then Halt(00) else
  if ReturnCode=00 then Halt(00);
  for HCyl:=00 to 79 do
  for MHHead:=0 to 1 do
  begin
    FormatTrack(DrvCode,HCyl,MHHead,00,00,ReturnCode);
    if ReturnCode=00 then
      Halt(00);
  end;
  Assign(FF,'FFFORMAT.DAT');
  ($I-)
  Reset(FF);
  ($I+)
  if IOHw[1c:0000] then
  Halt(00);
  for MHSec:=01 to 06 do
  begin
    Read(FF,RecBuf);
    Write(LAData[DrvCode,00,0,MHSec,00],RecBuf,ReturnCode);
    if ReturnCode=00 then
      Halt(00);
  end;
  for MHSec:=07 to 08 do
  begin
    Read(FF,RecBuf);
    Write(LAData[DrvCode,79,1,MHSec,00],RecBuf,ReturnCode);
    if ReturnCode=00 then
      Halt(00);
  end;
  Close(FF);
end.

```

```

000000 09 03 00 50 46 20 44 01 54 41 20 00 04 01 01 00
000010 02 03 08 08 08 08 08 08 08 08 08 08 08 08 08 08
000020 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08
000030 08 00 75 38 08 09 01 08 00 0A 00 18 04 00 0D 18
000040 84 14 02 00 00 00 33 00 08 00 00 00 00 24 00 04
000050 01 05 08 74 03 04 00 80 00 00 0F 40 01 01 0A 0A 00
000060 74 04 0A 07 07 00 00 00 37 00 00 00 00 02 02
000070 00 00 08 83 07 00 01 07 14 00 28 00 08 00 14 0A 0A
000080 84 05 24 00 00 02 07 06 02 03 05 01 0A 0A 06 00
000090 07 04 04 02 04 02 04 07 04 02 04 02 04 02 04 02
0000A0 70 08 09 75 13 3C 80 74 18 07 04 03 02 04 01 0A
0000B0 08 00 07 04 06 02 00 00 12 0A 00 75 13 0A 28 07 04
0000C0 00 00 00 00 00 00 00 01 07 06 04 03 02 03 07 04 06
0000D0 00 00 18 09 07 18 0A 01 07 04 02 03 02 04 01 07 06
0000E0 54 02 00 00 14 07 04 06 02 03 2B 03 01 00 00 0A 01
0000F0 00 0A 04 05 08 07 00 01 08 10 1F 51 09 00 00 08 08
000100 01 00 00 04 26 3E 48 08 08 74 03 01 07 20 0F 04 4E
000110 74 08 59 1F 08 0F 00 00 00 00 00 00 00 00 00 00
000120 01 08 18 04 02 00 16 03 03 00 04 19 0A 08 05 00 40
000130 05 49 28 06 02 02 33 0F 0A 04 05 08 00 00 00 00
000140 17 00 74 00 00 00 00 00 00 31 00 74 05 08 07 00 08
000150 8A 0A 00 00 00 00 00 00 00 0A 06 00 00 00 0A 06 00
000160 18 73 0C 0F 08 00 0A 02 07 04 08 00 00 00 00 00
000170 03 58 58 08 0F 01 89 08 0F 08 01 00 00 00 00 00
000180 18 08 58 00 02 26 8A 46 00 45 08 05 07 75 00 3C
000190 00 04 05 02 60 42 10 34 08 18 00 03 04 08 75 03 06
0001A0 2F 58 58 00 00 00 00 00 00 00 00 00 00 00 00 00
0001B0 0A 84 75 02 08 28 0F 06 03 54 48 08 73 20 0A 73
0001C0 30 74 48 45 20 44 81 74 81 20 44 49 73 48 00 48
0001D0 4F 20 73 78 73 74 45 40 20 44 48 4C 05 73 00 49
0001E0 2F 4F 20 65 72 72 6F 72 00 43 4F 20 20 20 20 20
0001F0 30 55 58 53 55 88 0C 0A 22 00 88 0A 02 07 44 0A
000200 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00
000210 00 00 06 01 00 00 00 00 00 00 00 00 00 00 00 00
000220 08 84 83 00 00 00 00 00 00 00 00 00 00 00 00 00
000230 30 84 18 00 00 00 10 50 08 0A 83 04 04 78 0A 78
000240 0F 75 03 08 0A 00 0A 03 10 50 08 0A 04 03 04 02
000250 00 02 18 00 00 00 00 00 00 00 00 00 00 00 00 00
000260 19 50 08 02 19 50 08 01 44 83 04 04 0A 00 02 00
000270 28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000280 08 49 46 0A 08 00 02 74 0C 30 01 00 74 51 50 0F 00
000290 74 55 08 0A 00 44 80 50 80 44 08 00 50 46 00 00
0002A0 84 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0002C0 18 50 08 45 84 83 04 02 80 0A 80 86 07 05 00 44
0002D0 0F 08 17 8D 81 81 08 0F 81 80 74 0A 08 0A 09 10 80
0002E0 28 8A 83 04 02 80 0F 88 3D 00 08 00 74 03 09 8C
0002F0 0F 08 03 00 4A 80 50 60 00 06 28 50 80 4A 00 88
000300 37 0F 83 04 04 00 44 80 50 0F 74 08 0F 78 0C 88
000310 43 00 83 04 06 07 44 0A 0F 83 78 0A 0F 70 03
000320 08 4E 08 08 85 5D 03 55 88 0C 08 02 00 88 8A 0A
000330 07 48 0F 00 00 08 04 19 50 08 0A 83 03 04 02 08
000340 04 18 50 08 00 00 18 00 00 0A 83 04 0A 08 18 88
000350 02 00 50 28 00 50 08 02 50 50 0A 14 50 0A 00
000360 08 43 04 00 00 00 44 00 81 81 00 00 00 74 81 88 08
000370 08 08 85 00 03 55 88 0C 00 00 08 0A 12 0A 14 00
000380 4A 0C 00 00 07 44 8A 00 00 07 44 86 00 00 07 44
000390 38 00 00 07 44 0A 00 00 07 44 86 00 00 07 44 80
0003A0 00 00 07 44 0A 00 00 07 44 82 00 00 08 05 0D 40
0003B0 89 44 28 08 45 04 29 44 08 70 12 03 04 28 44
0003C0 28 78 0A 88 44 04 29 44 08 08 05 07 44 08 00 06
0003D0 07 4A 8A 00 00 88 7A 8A 04 4C 00 00 00 4A 8A 83
0003E0 78 8A 0C 00 00 00 00 00 00 00 00 00 00 00 00 00
0003F0 0F 17 83 04 0A 08 44 18 50 0A 05 13 50 0A 00
000400 0F 0F 0F 7F 0F 0F 7F 0F 0F 7F 0F 0F 7F 0F 0F
000410 7F 0F 7F 7F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000420 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000430 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000440 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000450 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000460 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000470 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000480 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000490 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0004A0 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0004B0 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0004C0 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0004D0 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0004E0 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0004F0 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000500 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000510 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000520 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000540 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000550 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000560 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000580 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000590 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0005A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0005B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0005C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0005D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0005E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0005F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000600 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000610 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000620 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000630 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000640 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000660 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000670 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000680 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000690 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000700 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```

000600 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000610 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000620 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000630 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000640 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000650 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000660 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000670 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000680 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000690 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0006A0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0006B0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0006C0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0006D0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0006E0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0006F0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000700 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000710 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000720 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000730 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000740 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000750 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000760 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000770 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000780 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000790 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0007A0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0007B0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0007C0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0007D0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0007E0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0007F0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000800 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000810 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000820 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000830 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000840 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000850 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000860 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000870 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000880 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000890 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0008A0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0008B0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0008C0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0008D0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0008E0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0008F0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000900 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000910 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000920 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000930 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000940 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000950 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000960 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000970 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000980 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000990 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0009A0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0009B0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0009C0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0009D0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0009E0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
0009F0 0F 0F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F 7F
000900 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000910 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000920 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000930 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000940 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000950 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000970 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000980 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000990 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000900 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000910 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000920 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000930 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000940 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000950 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000970 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000980 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000990 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0009F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000900 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000910 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000920 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000930 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000940 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000950 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000970 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000980 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000990 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```



```

-----
FFCOPY.FAS          2446bytes  1993-08-24  2:00:00
FFCOPY.EXE          4512bytes  1993-08-24  2:00:00
-----
Printed on 1994-09-02
-----

```

```

program FFCopy;
(*  D.F.システムソフトウェア
   *  データディスクコピープログラム Version 2.00 1993/08/24  *)
($D-L)

uses StdLib,FDLib;

type RecBufType=array[0000..1023] of Byte;

var DrvCodeA,DrvCodeB,WCyl,WHHead,WHSec,ReturnCode:Byte;
    RecBuf:array[0..1,01..08] of RecBufType;

begin
  if UpCaseStr(ParamStr(1))='A:' then DrvCodeA:=HI[MemF[00040:$2C84]] else
  if UpCaseStr(ParamStr(1))='B:' then DrvCodeA:=HI[MemF[00040:$2C83]] else
  if UpCaseStr(ParamStr(1))='C:' then DrvCodeA:=HI[MemF[00040:$2C82]] else
  if UpCaseStr(ParamStr(1))='D:' then DrvCodeA:=HI[MemF[00040:$2C81]] else
  if UpCaseStr(ParamStr(1))='E:' then DrvCodeA:=HI[MemF[00040:$2C80]] else
  if UpCaseStr(ParamStr(1))='F:' then DrvCodeA:=HI[MemF[00040:$2C91]] else
  if UpCaseStr(ParamStr(1))='G:' then DrvCodeA:=HI[MemF[00040:$2C92]] else
  if UpCaseStr(ParamStr(1))='H:' then DrvCodeA:=HI[MemF[00040:$2C94]] else
    Halt(101);
  if UpCaseStr(ParamStr(2))='A:' then DrvCodeB:=HI[MemF[00040:$2C84]] else
  if UpCaseStr(ParamStr(2))='B:' then DrvCodeB:=HI[MemF[00040:$2C83]] else
  if UpCaseStr(ParamStr(2))='C:' then DrvCodeB:=HI[MemF[00040:$2C82]] else
  if UpCaseStr(ParamStr(2))='D:' then DrvCodeB:=HI[MemF[00040:$2C81]] else
  if UpCaseStr(ParamStr(2))='E:' then DrvCodeB:=HI[MemF[00040:$2C80]] else
  if UpCaseStr(ParamStr(2))='F:' then DrvCodeB:=HI[MemF[00040:$2C91]] else
  if UpCaseStr(ParamStr(2))='G:' then DrvCodeB:=HI[MemF[00040:$2C92]] else
  if UpCaseStr(ParamStr(2))='H:' then DrvCodeB:=HI[MemF[00040:$2C94]] else
    Halt(101);
  if DrvCodeA=DrvCodeB then
    Halt(101);
  ReadDrive(DrvCodeA,ReturnCode);
  if (ReturnCode=00) or
  (ReturnCode=01) then Halt(102) else
  if (ReturnCode=04) then Halt(103);
  ReadDrive(DrvCodeB,ReturnCode);
  if (ReturnCode=00) or
  (ReturnCode=01) or
  (ReturnCode=04) then Halt(102) else
  if (ReturnCode=04) then Halt(103);
  for WCyl:=00 to 78 do
  begin
    for WHHead:=0 to 1 do
    for WHSec:=01 to 08 do
      begin
        ReadDiskData (DrvCodeA,WCyl,WHHead,WHSec,001,
                     RecBuf[0]Head,WHSec,ReturnCode);
        if ReturnCode=00 then
          Halt(103);
      end;
        for WHSec:=0 to 1 do
        for WHSec:=01 to 08 do
          begin
            WriteDiskData(DrvCodeA,WCyl,WHHead,WHSec,001,
                          RecBuf[0]Head,WHSec,ReturnCode);
            if ReturnCode=00 then
              Halt(103);
          end;
        end;
      end;
    end;
  end;
end;
end;
end;

```

PFMODEL.PAS

3448bytes 1993-08-24 2:00:00

Printed on 1994-08-02

```

program PFMModel;
(* PFDシステムソフトウェア
 * データ作成プログラムフレームワーク Version 2.00 1993/08/24 *)
uses SysLib, IOLib, PFDLib, PFMDefLib;

function CellCode(X,Y,Z:LongInt):Byte;
begin
  if (Z=0) then
    CellCode:=00;
  else
    CellCode:=01;
  end;
end;

procedure SetParameter(var SizeX,SizeY,SizeZ:Word; var Note:string);
begin
  SizeX:=0;
  SizeY:=0;
  SizeZ:=0;
  Note:='';
end;

var ReturnCode:Byte;
    SizeX,SizeY,SizeZ,BaseX,BaseY,BaseZ,OffX,OffY,OffZ:Word;
    X,Y,Z:LongInt;
    BitAddr,BitAddrEnd,BitLength:LongInt;
    Note:string;
    Cube:CubeType;
begin
  if (OpCode=Paramtr[1])='A' then BrvCode:=SI (Name[0040:02C6]) else
  if (OpCode=Paramtr[1])='C' then BrvCode:=SI (Name[0040:02C8]) else
  if (OpCode=Paramtr[1])='D' then BrvCode:=SI (Name[0040:02CA]) else
  if (OpCode=Paramtr[1])='E' then BrvCode:=SI (Name[0040:02CC]) else
  if (OpCode=Paramtr[1])='F' then BrvCode:=SI (Name[0040:02CE]) else
  if (OpCode=Paramtr[1])='G' then BrvCode:=SI (Name[0040:02D0]) else
  if (OpCode=Paramtr[1])='H' then BrvCode:=SI (Name[0040:02D2]) else
  if (OpCode=Paramtr[1])='I' then BrvCode:=SI (Name[0040:02D4]) else
  Halt(00);
  BaseX:=BrvCode.ReturnCode;
  if (ReturnCode=00) or
  (ReturnCode=10) or
  (ReturnCode=11) then Halt(00) else
  if (ReturnCode=00) then Halt(00);
  SetParameter(SizeX,SizeY,SizeZ,Note);
  if ((SizeX mod 16)=0) or (SizeX=640) or
  ((SizeY mod 4)=0) or (SizeY=640) then
  (SizeZ mod 4)=0) or (SizeZ=640) then
    Halt(00);
  OpenIAB(ReturnCode);
  WriteIABCode($000,ReturnCode);
  if (ReturnCode=00) then
    Halt(00);
  WriteLn('PFMODEL, 定義された形状をPFDデータに変換します。');
  BitAddr:=0000000;
  for BaseX:=000 to SizeX div 4-1 do
  begin
    BitAddrEnd:=BitAddr;
    for BaseY:=000 to SizeY div 4-1 do
    for BaseZ:=000 to SizeZ div 4-1 do
    begin
      for OffX:=0 to 3 do
      for OffY:=0 to 3 do
      for OffZ:=0 to 3 do
      begin
        X:=BaseX*4+OffX;
        Y:=BaseY*4+OffY;
        Z:=BaseZ*4+OffZ;
        Cube(OffX,OffY,OffZ):=CellCode(X,Y,Z);
      end;
      CubeYOLok(BitAddrEnd,Cube,ReturnCode);
    end;
    BitLength:=BitAddrEnd-BitAddr;
    WriteContent(BaseZ,BitAddr,BitLength,ReturnCode);
    if (ReturnCode=00) then
      Halt(00);
    BitAddr:=BitAddrEnd;
    Write('PFMODEL, 層');
    WriteIntegerChar((BaseX*4) div 100) mod 10;
    WriteIntegerChar((BaseX*4) div 10) mod 10;
    WriteIntegerChar((BaseX*4) mod 10);
    Write(' ');
    WriteIntegerChar((BaseY*4) div 100) mod 10;
    WriteIntegerChar((BaseY*4) div 10) mod 10;
    WriteIntegerChar((BaseY*4) mod 10);
    Write(' ');
    WriteIntegerChar((BaseZ*4) div 10) mod 10;
    WriteIntegerChar((BaseZ*4) mod 10);
    WriteLn('を定義しました。');
  end;
  WriteLn('PFMODEL, PFDデータが完成しました。');
end;

```

```
-----
FFNOTE.DOS      1699bytes 1993-08-24 2:00:00
FFNOTE.EXE     18240bytes 1993-08-24 2:00:00
Printed on 1994-09-02
-----
```

```
program FFNote;
(* P Fシステムソフトウェア
 * データ見出し書き換えプログラム Version 3.00 1993/08/24 *)
uses StdLib, CRTLib, FFDLib, IOLib, FFDataLib;

var ReturnCode:Byte;
    DiskCode:Word;
    Note:string;

begin
  IF OpCharacter(ParamStr(1))='A' then DirCode:=HI(Mask($0040:$2080)) else
  IF OpCharacter(ParamStr(1))='R' then DirCode:=HI(Mask($0040:$2080)) else
  IF OpCharacter(ParamStr(1))='C' then DirCode:=HI(Mask($0040:$2080)) else
  IF OpCharacter(ParamStr(1))='D' then DirCode:=HI(Mask($0040:$2080)) else
  IF OpCharacter(ParamStr(1))='R' then DirCode:=HI(Mask($0040:$2080)) else
  IF OpCharacter(ParamStr(1))='P' then DirCode:=HI(Mask($0040:$2080)) else
  IF OpCharacter(ParamStr(1))='Q' then DirCode:=HI(Mask($0040:$2080)) else
  IF OpCharacter(ParamStr(1))='B' then DirCode:=HI(Mask($0040:$2080)) else
  Halt($03);
  SetDrive(DirCode,ReturnCode);
  IF (ReturnCode=$00) or
  (ReturnCode=$10) or
  (ReturnCode=$13) then Halt($02) else
  IF ReturnCode=$09 then Halt($03);
  OpenDisk(ReturnCode);
  IF ReturnCode=$00 then
    Halt($03);
  ReadDiskCode(DiskCode,ReturnCode);
  IF ReturnCode=$00 then
    Halt($03);
  IF DiskCode=$FFFF then
    begin
      WriteLn('FFNOTE : このP Fデータは使えません。');
      Halt($04);
    end;
  ReadData(Note,ReturnCode);
  IF ReturnCode=$00 then
    Halt($03);
  Write(' [          ] ');
  MoveCursor(74);
  InputString(Note);
  WriteLn(Note,ReturnCode);
  IF ReturnCode=$00 then
    Halt($03);
  CloseDisk(ReturnCode);
  IF ReturnCode=$00 then
    Halt($03);
end;
```

```

-----
PFHECK.PAS          7406bytes 1993-08-24 2:00:00
PFHECK.EXE         27680bytes 1993-08-24 2:00:00
-----

```

```
Printed on 1994-08-02
```

```

program PFCheck;
(*  DOSシステムソフトウェア          *)
(*  データチェックプログラム  Verion 3.00  1993/08/24  *)
uses Winapi, StdLib, CRTLib, KeyLib, FPU0Lib, IOLib, FPU0TLib, PPF0TLib;
var DisplayType:string[3];
procedure InputDisplayType(var DisplayType&str);
var StatusCode:Byte;
    KeyCode:Word;
    DisplayType:string[3] absolute DisplayType&str;
begin
    StatusCode:=000;
    DisplayType:='';
    repeat
        GetKeyCode(KeyCode);
        if (DisplayType='') then
            begin
                if 01(KeyCode)=010 then
                    begin
                        DisplayType:=DisplayType+'0';
                        Write('0');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=020 then
                    begin
                        DisplayType:=DisplayType+'P';
                        Write('P');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=030 then
                    begin
                        DisplayType:=DisplayType+'B';
                        Write('B');
                    end;
                if 01(KeyCode)=040 then
                    begin
                        DisplayType:=DisplayType+'2';
                        Write('2');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=050 then
                    begin
                        DisplayType:=DisplayType+'4';
                        Write('4');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=060 then
                    begin
                        DisplayType:=DisplayType+'5';
                        Write('5');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=070 then
                    begin
                        DisplayType:=DisplayType+'6';
                        Write('6');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=080 then
                    begin
                        DisplayType:=DisplayType+'8';
                        Write('8');
                        StatusCode:=001;
                    end;
                if (DisplayType='B') or
                    (DisplayType='2') or
                    (DisplayType='4') or
                    (DisplayType='5') or
                    (DisplayType='6') or
                    (DisplayType='8') then
                    begin
                        DisplayType:=DisplayType+'1';
                        Write('1');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=090 then
                    begin
                        DisplayType:=DisplayType+'3';
                        Write('3');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=100 then
                    begin
                        DisplayType:=DisplayType+'7';
                        Write('7');
                        StatusCode:=001;
                    end;
                if 01(KeyCode)=110 then
                    begin
                        DisplayType:=DisplayType+'9';
                        Write('9');
                        StatusCode:=001;
                    end;
            end;
        until (Length(DisplayType)=3) or (StatusCode=001) or (01(KeyCode)=000);
        if StatusCode=001 then
            DisplayType:='';
        end;
    end;
end;

```



```

VAR ReturnCode,ReturnCodeByte;
DIAGCode,HexA,HexW,HexD,HexE,HexF,HexG,HexH,HexI,HexJ,HexK,HexL,HexM,HexN,HexO,HexP,HexQ,HexR,HexS,HexT,HexU,HexV,HexW,HexX,HexY,HexZ,HexAA,HexAB,HexAC,HexAD,HexAE,HexAF,HexBA,HexBB,HexBC,HexBD,HexBE,HexBF,HexCA,HexCB,HexCC,HexCD,HexCE,HexCF,HexDA,HexDB,HexDC,HexDD,HexDE,HexDF,HexEA,HexEB,HexEC,HexED,HexEE,HexEF,HexFA,HexFB,HexFC,HexFD,HexFE,HexFF;
Note:string;
Layer:LayerType;

begin
  InitGraph;
  CursorOff;
  IF UpCaseStr(ParamStr(1))='A' THEN DirCode:=HI (HexW[0040:02C6]); ELSE
  IF UpCaseStr(ParamStr(2))='B' THEN DirCode:=HI (HexW[0040:02C8]); ELSE
  IF UpCaseStr(ParamStr(1))='C' THEN DirCode:=HI (HexW[0040:02CA]); ELSE
  IF UpCaseStr(ParamStr(1))='D' THEN DirCode:=HI (HexW[0040:02CC]); ELSE
  IF UpCaseStr(ParamStr(1))='E' THEN DirCode:=HI (HexW[0040:02CE]); ELSE
  IF UpCaseStr(ParamStr(1))='F' THEN DirCode:=HI (HexW[0040:02D0]); ELSE
  IF UpCaseStr(ParamStr(1))='G' THEN DirCode:=HI (HexW[0040:02D2]); ELSE
  IF UpCaseStr(ParamStr(1))='H' THEN DirCode:=HI (HexW[0040:02D4]); ELSE
  begin
    CursorOn;
    Ball(001);
  end;
  HexaDrive(DirCode,ReturnCode);
  IF ReturnCode=00 THEN
  (ReturnCode=10) then
  begin
    CursorOn;
    Ball(002);
  end; ELSE
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  OpenDisk(ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  ReadDIAGCode(DIAGCode,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  IF DIAGCode=FFFF THEN
  begin
    WriteLn('FFCHECK, このPFデータは壊れません。');
    CursorOn;
    Ball(004);
  end;
  ReadHexA(HexA,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  ReadHexW(HexW,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  ReadHexD(HexD,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  ReadHexE(HexE,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  ReadHexG(HexG,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  ReadHexI(HexI,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  ReadHexM(HexM,ReturnCode);
  IF ReturnCode=00 THEN
  begin
    CursorOn;
    Ball(003);
  end;
  WriteLn('FFCHECK, このディスクには次のPFデータが入っています。');
  WriteLn('  ',Note,');');
  WriteLn;
  WriteLn('  X方向の寸法=');
  WriteIntegerChar((HexW div 100) mod 10);
  WriteIntegerChar((HexW div 10) mod 10);
  WriteIntegerChar((HexW ) mod 10);
  WriteLn;
  WriteLn('  Y方向の寸法=');
  WriteIntegerChar((HexW div 100) mod 10);
  WriteIntegerChar((HexW div 10) mod 10);
  WriteIntegerChar((HexW ) mod 10);
  WriteLn;
  WriteLn('  Z方向の寸法=');
  WriteIntegerChar((HexW div 100) mod 10);
  WriteIntegerChar((HexW div 10) mod 10);
  WriteIntegerChar((HexW ) mod 10);
  WriteLn;
  WriteLn('FFCHECK, リターンキーを押してください。');
  ReadLn;

```

```

SC:=Size2;
repeat
  C:=0;
  repeat
    Write('FFCHECK : 表示する層=_____');
    MoveCursor(1);
    CursorOn;
    E:=KeyDecimal;
    WriteIntegerChar(E);
    E:=E*10+KeyDecimal;
    WriteIntegerChar(E mod 10);
    E:=E*10+KeyDecimal;
    WriteIntegerChar(E mod 10);
    WriteLn;
    CursorOff;
  until E=Size2;
  IF E<=0 then
  begin
    StatusCode:=StatusCode and 8FF;
    WriteLn('FFCHECK : データをロードしますか?');
    WriteLn('      □ - する ..... [Y] 是--');
    WriteLn('      □ - しない ..... [N] 非--');
    IF KeyNo=0 then
      StatusCode:=StatusCode or 801
    else
      IF SC=Size2 then
      begin
        WriteLn('FFCHECK : データがロードされていません。');
        WriteLn('      プログラムを終りますか?');
        WriteLn('      続ける ..... [Y] 是--');
        WriteLn('      終了する ..... [N] 非--');
        IF KeyNo=0 then
          begin
            ClearGraph;
            CursorOn;
            Halt(800);
          end
          else
            StatusCode:=StatusCode or 801;
        end
      else
      begin
        Write('FFCHECK : 表示する層=');
        WriteIntegerChar((SC div 100) mod 10);
        WriteIntegerChar((SC div 10) mod 10);
        WriteIntegerChar(SC mod 10);
        WriteLn;
      end;
      IF (StatusCode and 801)=801 then
      begin
        Move:=E div 4;
        ShowLayer(L, SizeX, SizeY, Layer.ReturnCode);
        IF ReturnCode=800 then
          begin
            CursorOn;
            Halt(800);
          end;
          SC:=E;
        end;
      end;
    repeat
      Write('FFCHECK : 表示の形式=');
      InputDisplayStyle(DisplayStyle);
      WriteLn;
      ClearGraph;
      IF DisplayStyle='' then
        ShowLayer(Layer, DisplayStyle, SizeX, SizeY)
      else
      begin
        WriteLn('FFCHECK : データを表示することができません。');
        WriteLn('      プログラムを終りますか?');
        WriteLn('      続ける ..... [Y] 是--');
        WriteLn('      終了する ..... [N] 非--');
        IF KeyNo=0 then
          begin
            ClearGraph;
            CursorOn;
            Halt(800);
          end;
        end;
      end;
    until DisplayStyle='';
  end.
end.

```



```

procedure SetParams;
procedure SplitDef(StrDef:string; var StrID,StrVal:string; var ResultCode:Byte);
var PosBreak:Byte;
begin
  PosBreak:=Pos('-',StrDef);
  if PosBreak=0 then
    begin
      ResultCode:=001;
      exit;
    end
  else
    begin
      StrID:=Copy(StrDef,1,PosBreak-1);
      StrVal:=Copy(StrDef,PosBreak+1,Length(StrDef)-PosBreak);
      while (StrID[1]<>'@') or (StrID[1]<>'#') do
        StrID:=Copy(StrID,2,Length(StrID)-1);
      while (StrID[Length(StrID)]<>'@') or (StrID[Length(StrID)]<>'#') do
        StrID:=Copy(StrID,1,Length(StrID)-1);
      while (StrVal[1]<>'@') or (StrVal[1]<>'#') do
        StrVal:=Copy(StrVal,2,Length(StrVal)-1);
      while (StrVal[Length(StrVal)]<>'@') or (StrVal[Length(StrVal)]<>'#') do
        StrVal:=Copy(StrVal,1,Length(StrVal)-1);
      ResultCode:=000;
    end;
end;

procedure CheckDef(StrDef:string; var ResultCode:Byte);
var ReturnCode:Byte;
    ValCode:Integer;
    StrID,StrVal:string;
begin
  with Params do
    begin
      ResultCode:=000;
      if StrDef='' then
        begin
          SplitDef(StrDef,StrID,StrVal,ReturnCode);
          if ReturnCode=000 then
            ResultCode:=001
          else
            begin
              StrID:=UpCaseStr(StrID);
              if StrID='CF' then
                begin
                  CF:=UpCaseStr(StrVal);
                end
              else
                if StrID='MA' then
                  begin
                    Val(StrVal,MA,ValCode);
                    if (ValCode=0000) or (MA=00000) then
                      ResultCode:=001;
                    end
                  else
                    if StrID='ATT' then
                      begin
                        if UpCaseStr(StrVal)='YES' then ATT:=001 else
                          if UpCaseStr(StrVal)='NO' then ATT:=000 else
                            ResultCode:=001;
                        end
                      else
                        if StrID='LSD' then
                          begin
                            if UpCaseStr(StrVal)='YES' then LSD:=001 else
                              if UpCaseStr(StrVal)='NO' then LSD:=000 else
                                ResultCode:=001;
                            end
                          else
                            if StrID='FR' then
                              begin
                                if UpCaseStr(StrVal)='YES' then FR:=001 else
                                  if UpCaseStr(StrVal)='NO' then FR:=000 else
                                    ResultCode:=001;
                                end
                              else
                                if StrID='PT' then
                                  begin
                                    if UpCaseStr(StrVal)='YES' then PT:=001 else
                                      if UpCaseStr(StrVal)='NO' then PT:=000 else
                                        ResultCode:=001;
                                    end
                                  else
                                    if StrID='SD' then
                                      begin
                                        if UpCaseStr(StrVal)='A') or
                                           (UpCaseStr(StrVal)='B') or
                                           (UpCaseStr(StrVal)='C') or
                                           (UpCaseStr(StrVal)='D') or
                                           (UpCaseStr(StrVal)='E') or
                                           (UpCaseStr(StrVal)='F') or
                                           (UpCaseStr(StrVal)='G') or
                                           (UpCaseStr(StrVal)='H') then
                                          SD:=UpCaseStr(StrVal)
                                        else
                                          ResultCode:=001;
                                        end
                                      end
                                    end
                                  end
                                end
                              end
                            end
                          end
                        end
                      end
                    end
                  end
                end
              end
            end
          end
        end
      end
    end
  end;
end;

```



```

IF StrID='FMCI' then
begin
Val(StrVal.FMCI.ValCode);
IF (ValCode<=0000) or (FMCI=0000) then
ResultCode:=001;
end else

```

```

IF StrID='FMMA' then
begin
Val(StrVal.FMMA.ValCode);
IF (ValCode<=0000) or (FMMA=0000) then
ResultCode:=001;
end else

```

```

IF StrID='FMNI' then
begin
Val(StrVal.FMNI.ValCode);
IF (ValCode<=0000) or (FMNI=0000) then
ResultCode:=001;
end else

```

```

IF StrID='FMNA' then
begin
Val(StrVal.FMNA.ValCode);
IF (ValCode<=0000) or (FMNA=0000) then
ResultCode:=001;
end else

```

```

IF StrID='FMIN' then
begin
Val(StrVal.FMIN.ValCode);
IF (ValCode<=0000) or (FMIN=0000) then
ResultCode:=001;
end else

```

```

IF StrID='FMMA' then
begin
Val(StrVal.FMMA.ValCode);
IF (ValCode<=0000) or (FMMA=0000) then
ResultCode:=001;
end else

```

```

IF StrID='FM' then
begin
Val(StrVal.FM.ValCode);
IF (ValCode<=0000) or (FM=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMCI' then
begin
Val(StrVal.EMCI.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMNI' then
begin
Val(StrVal.EMNI.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMCI' then
begin
Val(StrVal.EMCI.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMNI' then
begin
Val(StrVal.EMNI.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMMA' then
begin
Val(StrVal.EMMA.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMNA' then
begin
Val(StrVal.EMNA.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMNI' then
begin
Val(StrVal.EMNI.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='EMMA' then
begin
Val(StrVal.EMMA.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='PT' then
begin
Val(StrVal.PT.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='PTT' then
begin
Val(StrVal.PTT.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='PTT' then
begin
Val(StrVal.PTT.ValCode);
IF (ValCode<=0000) then
ResultCode:=001;
end else

```

```

IF StrID='PT' then
begin
Val(StrVal.PT.ValCode);
IF (ValCode<=0000) or (PT=000000) then
ResultCode:=001;
end else

```

```

IF StrID='PTT' then
begin
Val(StrVal.PTT.ValCode);
IF (ValCode<=0000) or (PTT=000000) then
ResultCode:=001;
end else

```

```

IF StrID='PTT' then
begin
Val(StrVal.PTT.ValCode);
IF (ValCode<=0000) or (PTT=000000) then
ResultCode:=001;
end else

```

```

IF StrID='WTI' then
begin
Val(StrVal.WTI.ValCode);
IF (ValCode<=0000) or (WTI=000000) then
ResultCode:=001;
end else

```

```

IF StrID='WTR' then
begin
Val(StrVal.WTR.ValCode);
IF (ValCode<=0000) or (WTR=000000) then
ResultCode:=001;
end else

```

```

IF StrID='WTC' then
begin
Val(StrVal.WTC.ValCode);
IF (ValCode<=0000) or (WTC=000000) then
ResultCode:=001;
end;

```

```

IF StrID='/' then
WriteLn(' ', StrVal);
end;
end;
end;
end;

```

```

procedure InputParam(StrID:string);
var ReturnCode:Byte;
    StrVal:string;
begin
  repeat
    ReadLn(StrVal);
    CheckNull;
    CheckDef(StrID:=''+StrVal.ReturnCode);
    if ReturnCode=0 then
      Write('          * ');
  until ReturnCode=0;
end;

var WParam, WParam, StatusCode, ReturnCode:Byte;
    StrDef:string;
    FPCConfig:Text;
begin
  WParam:=ParamInfo;
  WParam:=ParamCount;
  for WParam:=1 to WParam do
    begin
      StrDef:=ParamStr(WParam);
      CheckDef(StrDef, ReturnCode);
    end;
  Assign(FPCConfig, 'FPNAXX.CPU');
  (E1-);
  Repeat(FPCConfig);
  (E1-);
  if TimeOut<=0000 then
    begin
      WriteLn('FPNAXX : 設定ファイル FPNAXX.CPU から動作パラメータを読み込みます。');
    repeat
      ReadLn(FPCConfig, StrDef);
      CheckDef(StrDef, ReturnCode);
      if ReturnCode=0 then
        begin
          WriteLn('FPNAXX : 次の設定文に誤りがあります。');
          WriteLn('          ', ' '*StrDef);
        end;
    until EOF(FPCConfig);
    Close(FPCConfig);
  end
  else
    if Param.CF='' then
      begin
        WriteLn('FPNAXX : 設定ファイル FPNAXX.CPU が見つかりません。');
        CloseOut;
        Halt(00);
      end;
    if Param.CF='' then
      begin
        StatusCode:=00;
        repeat
          Assign(FPCConfig, Param.CF);
          (E1-);
          Repeat(FPCConfig);
          (E1-);
          if TimeOut<=0000 then
            begin
              WriteLn('FPNAXX : 設定ファイル '+Param.CF+' から動作パラメータを読み込みます。');
            repeat
              ReadLn(FPCConfig, StrDef);
              CheckDef(StrDef, ReturnCode);
              if ReturnCode=0 then
                begin
                  WriteLn('FPNAXX : 次の設定文に誤りがあります。');
                  WriteLn('          ', ' '*StrDef);
                end;
            until EOF(FPCConfig);
            Close(FPCConfig);
            StatusCode:=01;
          end
          else
            begin
              WriteLn('FPNAXX : 設定ファイル '+Param.CF+' が見つかりません。');
              WriteLn('          プログラムを終りますか?');
              WriteLn('          続ける ..... [Y] キー');
              WriteLn('          中止する ..... [N] キー');
              RepeatKey;
              if Key=Y then
                begin
                  CloseOut;
                  Halt(00);
                end;
              StatusCode:=00;
              WriteLn('FPNAXX : もう一度設定ファイルを選びますか?');
              WriteLn('          設定ファイルを選ぶ ..... ファイル名 + リターンキー');
              WriteLn('          このまま続ける ..... リターンキー');
              Write('          ');
              InputParam('CF');
            end;
          until (StatusCode=01) or (Param.CF='') ;
        end;
      for WParam:=1 to WParam do
        begin
          StrDef:=ParamStr(WParam);
          CheckDef(StrDef, ReturnCode);
          if ReturnCode=0 then
            begin
              WriteLn('FPNAXX : 次の設定文に誤りがあります。');
              WriteLn('          ', ' '*StrDef);
            end;
          end;
        end;
      end;

```

```

with Parama do
  IF (SD ='' ) or
  (SD ='' ) or
  (SA ='' ) or
  ((ST0 =00000) and
  (ST1 =00000) and
  (ST2 =00000) and
  (ST3 =00000)) or
  (ST4 =00000) or
  (ST5 =00000) or
  (ST6 =00000) or
  (ST7 =00000) or
  (ST8 =00000) then
begin
  WriteLn('PPMARK : 未設定の動作パラメータがあります。 適切に入力しますか?');
  WriteLn('          入力する ..... [Y] 是~/');
  WriteLn('          プログラムも中止する ..... [N] キー~/');
  RepeatKey;
  IF Key[0] = 'Y' then
  begin
    ClearScr;
    Halt(000);
  end;

  while SD='' do
  begin
    Write('          , パラメータ SD = ');
    InputParam('SD');
  end;

  while SA='' do
  begin
    Write('          , パラメータ SA = ');
    InputParam('SA');
  end;

  while ST0='' do
  begin
    Write('          , パラメータ ST0 = ');
    InputParam('ST0');
  end;

  while ST1='' do
  begin
    Write('          , パラメータ ST1 = ');
    InputParam('ST1');
  end;

  while ST2='' do
  begin
    Write('          , パラメータ ST2 = ');
    InputParam('ST2');
  end;

  while ST3='' do
  begin
    Write('          , パラメータ ST3 = ');
    InputParam('ST3');
  end;

  while ST4='' do
  begin
    Write('          , パラメータ ST4 = ');
    InputParam('ST4');
  end;

  while ST5='' do
  begin
    Write('          , パラメータ ST5 = ');
    InputParam('ST5');
  end;

  while ST6='' do
  begin
    Write('          , パラメータ ST6 = ');
    InputParam('ST6');
  end;

  while ST7='' do
  begin
    Write('          , パラメータ ST7 = ');
    InputParam('ST7');
  end;

  while ST8='' do
  begin
    Write('          , パラメータ ST8 = ');
    InputParam('ST8');
  end;

  WriteLn('PPMARK : 設定値以上で終わります。');
end;
end;

```



```

procedure Locate(FX, FY, FZ, Word; ST: LongInt);
begin
  IF FX=0000 THEN FX:=FARMAX.FX;
  IF FY=0000 THEN FY:=FARMAX.FY;
  IF FZ=0000 THEN FZ:=FARMAX.FZ;
  while FY<-FARMAX.FY do
  begin
    IF FZ<FARMAX.FZ THEN MoveDecFZ(FARMAX.FZ, FARMAX.FZMin, FARMAX.FZMax, ST);
    IF FZ>FARMAX.FZ THEN MoveIncFZ(FARMAX.FZ, FARMAX.FZMin, FARMAX.FZMax, ST);
    end;
    while FY<-FARMAX.FY do
    begin
      IF FY<FARMAX.FY THEN MoveDecFY(FARMAX.FY, FARMAX.FYMin, FARMAX.FYMax, ST);
      IF FY>FARMAX.FY THEN MoveIncFY(FARMAX.FY, FARMAX.FYMin, FARMAX.FYMax, ST);
      end;
    while FX<-FARMAX.FX do
    begin
      IF FX<FARMAX.FX THEN MoveDecFX(FARMAX.FX, FARMAX.FXMin, FARMAX.FXMax, ST);
      IF FX>FARMAX.FX THEN MoveIncFX(FARMAX.FX, FARMAX.FXMin, FARMAX.FXMax, ST);
      end;
    end;
  end;

procedure Clear(var ResultCode: MyInt);
var MMove, MMove_KeyCode: Word;
begin
  MMove:=1224;
  ResultCode:=00;
  RepeatKey;
  repeat
    GetKeyCode(KeyCode);
    RepeatKey;
    IF (KeyCode=1800) AND (MMove=1224) THEN
      MMove:=MMove*2;
    IF (KeyCode=1800) AND (MMove=0001) THEN
      MMove:=MMove div 2;
    for MMove:=0001 to MMove do
      case KeyCode of
        $1C00: MoveIncFX(FARMAX.FX, FARMAX.FXMin, FARMAX.FXMax, FARMAX.ST);
        $1B00: MoveDecFX(FARMAX.FX, FARMAX.FXMin, FARMAX.FXMax, FARMAX.ST);
        $1A00: MoveIncFY(FARMAX.FY, FARMAX.FYMin, FARMAX.FYMax, FARMAX.ST);
        $1900: MoveDecFY(FARMAX.FY, FARMAX.FYMin, FARMAX.FYMax, FARMAX.ST);
        $1800: MoveIncFZ(FARMAX.FZ, FARMAX.FZMin, FARMAX.FZMax, FARMAX.ST);
        $1700: MoveDecFZ(FARMAX.FZ, FARMAX.FZMin, FARMAX.FZMax, FARMAX.ST);
      end;
    IF KeyCode=0019 THEN
      ResultCode:=001;
      WriteLn('PRGMSG : 関数呼び出し');
      WriteLn(FARMAX.FX, 5);
      WriteLn(' ');
      WriteLn(FARMAX.FY, 5);
      WriteLn(' ');
      WriteLn(FARMAX.FZ, 5);
      WriteLn(' ');
      WriteLn(' ');
      MoveCursor(1);
    until (GetKeyCode=000) or (ResultCode=00);
    RepeatKey;
  end;

procedure PullUp;
begin
  Locate(FARMAX.FX, FARMAX.FY, FARMAX.FZ, FARMAX.FZ, FARMAX.ST);
end;

procedure Load(PZ: Word);
begin
  Locate(FARMAX.FX, FARMAX.FY, PZ, FARMAX.ST);
  Delay(FARMAX.WL);
end;

```

```

procedure AdjJustOrig;
var ReturnCode:Byte;
    FMin,FMax,FMinL,FMaxL,FMinR,FMaxR:Word;
begin
  WriteLn('FPMAXX : スキップを移動させて原点を指定してください。');
  WriteLn('          : 原点を指定する ..... [Y]  ←---');
  WriteLn('          : 加工を中止する ..... ESCキー---');
  FMinL:=Paramm.FMinL;
  FMaxL:=Paramm.FMaxL;
  FMinR:=Paramm.FMinR;
  FMaxR:=Paramm.FMaxR;
  FMin:=Paramm.FMin;
  FMax:=Paramm.FMax;
  FMinL:=Paramm.FMinL;
  FMaxL:=Paramm.FMaxL;
  FMinR:=Paramm.FMinR;
  FMaxR:=Paramm.FMaxR;
  Paramm.FMin:=+0000;
  Paramm.FMax:=+0510;
  Paramm.FMinL:=+0000;
  Paramm.FMaxL:=+0510;
  Paramm.FMinR:=+0000;
  Paramm.FMaxR:=+0510;
  Paramm.FMin:=+0000;
  Paramm.FMax:=+0510;
  Paramm.FMinL:=+0000;
  Paramm.FMaxL:=+0510;
  Paramm.FMinR:=+0000;
  Paramm.FMaxR:=+0510;
  Show(Paramm.ReturnCode);
  if ReturnCode=001 then
  begin
    WriteLn('FPMAXX : 本当に中止していいのですか?');
    WriteLn('          : 中止する ..... [Y]  ←---');
    WriteLn('          : 加工を続ける ..... [N]  ←---');
    ReadKey;
    if Key=#001 then
    begin
      ClearGraph;
      OutWindow;
      Ball(100);
    end;
    WriteLn('FPMAXX : 現在の位置を原点にします。');
    Paramm.FMin:=FMinL;
    Paramm.FMax:=FMaxL;
    Paramm.FMinL:=FMinL;
    Paramm.FMaxL:=FMaxL;
    Paramm.FMinR:=FMinR;
    Paramm.FMaxR:=FMaxR;
    Paramm.FMin:=FMinL;
    Paramm.FMax:=FMaxL;
    Paramm.FMinL:=FMinL;
    Paramm.FMaxL:=FMaxL;
    Paramm.FMinR:=FMinR;
    Paramm.FMaxR:=FMaxR;
    Paramm.FMin:=+0000;
    Paramm.FMax:=+0510;
    Paramm.FMinL:=+0000;
    Paramm.FMaxL:=+0510;
    Paramm.FMinR:=+0000;
    Paramm.FMaxR:=+0510;
  end;
end;

procedure Test;
var ReturnCode:Byte;
    FX,FXL,FXR:Word;
begin
  WriteLn('FPMAXX : 加工を一時的に停止します。');
  FX:=Paramm.FX;
  FXL:=Paramm.FXL;
  FXR:=Paramm.FXR;
  if Paramm.ATT=001 then
  begin
    LockIn(Paramm.FXT,Paramm.FVT,Paramm.FPT,Paramm.FRT);
    WriteLn('FPMAXX : 途中検査をすることができます。');
    WriteLn('          : 加工を中止する ..... ESCキー---');
    Show(Paramm.ReturnCode);
    if ReturnCode=001 then
    begin
      WriteLn('FPMAXX : 本当に中止していいのですか?');
      WriteLn('          : 中止する ..... [Y]  ←---');
      WriteLn('          : 加工を続ける ..... [N]  ←---');
      ReadKey;
      if Key=#001 then
      begin
        ClearGraph;
        OutWindow;
        Ball(100);
      end;
    end;
  else
    WriteLn('FPMAXX : 加工を再開します。');
    ReadTestSign;
    Paramm.FXT:=Paramm.FX;
    Paramm.FVT:=Paramm.FV;
    Paramm.FPT:=Paramm.FP;
    Paramm.FRT:=Paramm.FR;
    LockIn(FX,FXL,FXR,Paramm.FPT);
  end;
end;

procedure TestFinal;
var ReturnCode:Byte;
begin
  WriteLn('FPMAXX : 加工が終わりました。');
  if Paramm.ATT=001 then
  begin
    LockIn(Paramm.FXT,Paramm.FVT,Paramm.FPT,Paramm.FRT);
  end;
  ReadTestSign;
  WriteLn('FPMAXX : 最終検査をすることができます。');
  WriteLn('          : 加工を終了する ..... ESCキー / リターンキー---');
  Show(Paramm.ReturnCode);
  ReadTestSign;
end;

```

```

procedure MakeBase;
var StatusCode:Byte;
    FF,FT,Word:
    S1MSL,S1MSH,S2MSL,S2MSH;
    IZ,I1MSL,I1MSH,I2,I1MSL,I2MSL;
begin
  WriteLn('PPMAX : 基礎形を加工します。');
  SetBaseSign;
  if Parama.FB=01 then
  begin
    WriteLn('PPMAX : 光学系を「基礎形加工モード」にして、どれかキーを押してください。');
    RepeatKey;
    WaitKey;
    RepeatKey;
    WriteLn('PPMAX : 加工を開始します。');
  end
  else
  begin
    Delay(Parama.MTI);
    S1MSL:=Parama.S1MSL div 2;
    S1MSH:=Parama.S1MSH div 2;
    S2MSL:=Parama.S2MSL div 2;
    S2MSH:=Parama.S2MSH div 2;
    if Parama.FB=0100 then
      Parama.FB:=Parama.FB;
    if Parama.FB=0000 then
      Parama.FB:=Parama.FB;
    if (Parama.SA='01') or
      (Parama.SA='02') or
      (Parama.SA='03') then
      begin
        IZMSL:=((( S1MSL +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        IZMSH:=((( S1MSH +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSL:=((( S1MSL +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSH:=((( S1MSH +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
      end
      else
      if Parama.SA='00L' then
      begin
        IZMSL:=((( S1MSL +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        IZMSH:=((( S1MSH +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSL:=(((MSH)*S1MSL+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSH:=(((MSH)*S1MSH+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
      end
      else
      if Parama.SA='00R' then
      begin
        IZMSL:=(((MSH)*S1MSL+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        IZMSH:=(((MSH)*S1MSH+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSL:=(((MSH)*S1MSL+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSH:=(((MSH)*S1MSH+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
      end
      else
      if Parama.SA='00M' then
      begin
        IZMSL:=(((MSH)*S1MSL+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        IZMSH:=(((MSH)*S1MSH+MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSL:=((( S1MSL +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
        I1MSH:=((( S1MSL +MSH)*Parama.FB-Parama.FB-1) div Parama.FB) *
      end;
      for IZ:=IZMSL to IZMSH do
      begin
        StatusCode:=00;
        FF:=IZ*Parama.FB-IZ*IZ;
        FT:=((IZMSL-Parama.MSL)*Parama.FB-IZ*IZ);
        locate(FF,FT,Parama.FE,Parama.FT);
        Load(Parama.FBMSL);
        for IZ:=IZMSL to IZMSH do
        begin
          FF:=IZ*Parama.FB-IZ*IZ;
          locate(FF,FT,Parama.FE,Parama.FT);
          Execute(Parama.MTI);
          if Notcon=01 then
            StatusCode:=01;
          end;
        end;
        if Notcon=01 then
          StatusCode:=01;
        WriteLn('PPMAX : 基礎形/欄');
        WriteLn(IZ-IZMSL,4);
        WriteLn('を完了しました。');
        if StatusCode=01 then
          Test;
        end;
      end;
      WriteLn('PPMAX : 基礎形が完成しました。 形状形の加工に移ります。');
      RepeatBaseSign;
      if Parama.FB=01 then
      begin
        WriteLn('PPMAX : 光学系を「形状形加工モード」にして、どれかキーを押してください。');
        RepeatKey;
        WaitKey;
        RepeatKey;
        if Notcon=01 then
          Test;
        else
          WriteLn('PPMAX : 加工を開始します。');
        end
      end
      else
        Delay(Parama.MTI);
    end;
  end;
end;

```

```

procedure MakeForm;
  type RunType=record
    Detal,DetAM,Data:Byte;
  end;
  var StatusCode:Byte;
      WMin:Array[0..63] of Word;
      RunListPtr:Array[0..63] of pointer;
procedure XLToRun(XL:Word; var Run:RunType);
begin
  Run.Detal:=(Run.Detal and $F0) or ((Lo(XL) and $0F) or (Hi(XL) shr 4) and $0F);
  Run.DetAM:=(Run.DetAM and $F0) or ((Lo(XL) shr 4) and $0F);
  Run.Data:=(Run.Data and $F0) or ((Lo(XL) and $0F) or (Hi(XL) shr 4) and $0F);
end;

procedure X0ToRun(X0:Word; var Run:RunType);
begin
  Run.Detal:=(Run.Detal and $0F) or ((Lo(X0) shl 4) and $0F);
  Run.DetAM:=(Run.DetAM and $0F) or ((Lo(X0) and $0F) or (Hi(X0) shr 4) and $0F);
  Run.Data:=(Run.Data and $0F) or ((Hi(X0) shl 4) and $0F);
end;

procedure RunToXL(Run:RunType; var XL:Word);
begin
  XL:=(Word(Run.Detal and $0F) or (Word(Run.DetAM and $0F) shl 4) or (Word(Run.Data and $0F) shl 8));
end;

procedure RunToX0(Run:RunType; var X0:Word);
begin
  X0:=(Word(Run.Detal and $0F) shr 4) or (Word(Run.DetAM and $0F) or (Word(Run.Data and $0F) shl 4));
end;

procedure RunToRunList(Run:RunType; M0Run:Word; var RunListPtr:pointer);
begin
  Mem[Seg(RunListPtr)+Ofs(RunListPtr)+M0Run*3] :=Run.Detal;
  Mem[Seg(RunListPtr)+Ofs(RunListPtr)+M0Run*3+1]:=Run.DetAM;
  Mem[Seg(RunListPtr)+Ofs(RunListPtr)+M0Run*3+2]:=Run.Data;
end;

procedure RunListToRun(RunListPtr:pointer; M0Run:Word; var Run:RunType);
begin
  Run.Detal:=Mem[Seg(RunListPtr)+Ofs(RunListPtr)+M0Run*3];
  Run.DetAM:=Mem[Seg(RunListPtr)+Ofs(RunListPtr)+M0Run*3+1];
  Run.Data:=Mem[Seg(RunListPtr)+Ofs(RunListPtr)+M0Run*3+2];
end;

procedure MakeRunListTable(HLineX,LineY:Word; var Layer:LayerType);
var X,Y,XMax,YMax,M0Run,M0RunR:Word;
    Run:RunType;
    LineBuffer:Array[0..63] of Byte;
    XL:array[0..63] of Word;
begin
  XMax:=HLineX-1;
  YMax:=LineY-1;
  for Y:=0 to YMax do
  begin
    for X:=0 to XMax do
      if (Layer[X div 16,Y] and BitMask(X)=$000 then LineBuffer[X]:=0;
      else LineBuffer[X]:=0;
    M0Run:=0;
    for X:=0 to XMax do
      begin
        if ((X=0) or ((X=0) and (LineBuffer[X-1]=0))) and
          (LineBuffer[X]=0) then
          begin
            XL[M0Run]:=X;
          end;
        if ((X=XMax) or ((X=XMax) and (LineBuffer[X-1]=0))) and
          (LineBuffer[X]=0) then
          begin
            M0Run:=X;
            Inc(M0RunR);
          end;
        end;
      M0Run[Y]:=M0RunR;
      M0Run:=0;
      M0RunR:=0;
      while M0Run<M0Run[Y] do
        begin
          XL[M0RunR]:=XL[M0Run];
          while (M0Run<M0Run[Y]-1) and ((X[M0Run]+Parase-M0)=XL[M0Run-1]) do
            Inc(M0Run);
          XL[M0RunR]:=M0[M0Run];
          Inc(M0RunR);
          Inc(M0Run);
        end;
      M0Run[Y]:=M0RunR;
      if MemAll((M0Run[Y]*8) then
        begin
          ClearGraph;
          CursorOn;
          Ball(17);
        end;
        GetMem(RunListPtr[Y],M0Run[Y]*3);
        M0Run:=0;
        while M0Run<M0Run[Y] do
          begin
            XLToRun(XL[M0Run],Run);
            X0ToRun(X0[M0Run],Run);
            RunToRunList(Run,M0Run,RunListPtr[Y]);
            Inc(M0Run);
          end;
        end;
end;

```



```

procedure MakeUnit(FSDiv, FSDivig, FSWord, WT:LongInt; var Layer:LayerType);
var
  StatusCode:Byte;
  X,Y,Start,YStart,XL,YS,FX,FY,WRNum,WRNumStart,WRM:Word;
  RowNR, RowNRInStage;
  MCall, MCallI, RTI:LongInt;
  Run:RunType;
begin
  if Param.DR='M0' then
    ShowLayer(Layer, Param.DR, Param.RowNR, Param.RowNR);
  if M0ton=0 then
    StatusCode:=StatusCode or 001;
  Param.FSDiv:=Param.FS;
  if Param.FD=Param.FMID then
    Param.FD:=Param.FMID;
  WT:=WT div Param.MEI;
  RowNR:=Param.RowNR div 2;
  RowNRIn:=Param.RowNR div 2;
  if Param.DR='M0' then
    begin
      GetMCall(MCall);
      MCallI:=00000;
      for WRM:=00001 to Param.MEI do
        begin
          StatusCode:=StatusCode and 0FD;
          YStart:=000;
          WRNumStart:=000;
          repeat
            SearchStartRun(Param.RowNR, YStart, WRNumStart, ReturnCode);
            if ReturnCode=00 then
              begin
                Y:=YStart;
                WRNum:=WRNumStart;
                FX:=(Y-RowNR)*Param.FD+FSDivig;
                MultiTable(MultiFtr(Y), WRNum, Run);
                RunToKL(Run, KL);
                RunToR(Run, R);
                Locate((X-RowNR)*Param.FD+FSDivig-Param.MEI, FX, Param.MEI, Param.FD, Param.FD, Param.FD);
                for X:=KL to R do
                  begin
                    FX:=(X-RowNR)*Param.FD+FSDivig;
                    Locate(FX, FY, Param.FD, Param.FD);
                    if M0ton=0 then
                      StatusCode:=StatusCode or 001;
                    Response(RTI);
                    Inc(MCallI);
                  end;
                Write('PPNAR : ユニットの発光が');
                WriteLn(Trim((MCallI*100)/(MCall*Param.MEI):3));
                WriteLn(' : 発光が完了しました。');
                MoveCursor(1);
              while not IsEndRun(Param.RowNR, Y, WRNum) do
                begin
                  SearchStartRun(Param.RowNR, Y, WRNum, ReturnCode);
                  if ReturnCode=00 then
                    StatusCode:=StatusCode or 002;
                    FX:=(Y-RowNR)*Param.FD+FSDivig;
                    MultiTable(MultiFtr(Y), WRNum, Run);
                    RunToKL(Run, KL);
                    RunToR(Run, R);
                    Locate((X-RowNR)*Param.FD+FSDivig-Param.MEI, FX, Param.FD, Param.FD, Param.FD);
                    for X:=KL to R do
                      begin
                        FX:=(X-RowNR)*Param.FD+FSDivig;
                        Locate(FX, FY, Param.FD, Param.FD);
                        if M0ton=0 then
                          StatusCode:=StatusCode or 001;
                        Response(RTI);
                        Inc(MCallI);
                      end;
                    Write('PPNAR : ユニットの発光が');
                    WriteLn(Trim((MCallI*100)/(MCall*Param.MEI):3));
                    WriteLn(' : 発光が完了しました。');
                    MoveCursor(1);
                  end;
                  Inc(WRNumStart);
                end;
              while
                begin
                  StatusCode:=StatusCode or 002;
                  if M0ton=0 then
                    StatusCode:=StatusCode or 001;
                  end;
                until (StatusCode and 002)=002;
                end;
                WriteLn('PPNAR : ユニットの発光が100%完了しました。');
                MoveCursor(1);
              end;

```

```

else
begin
for MWEI=0001 to Parama.MEI do
begin
if MWEI=Parama.MEI then
begin;layer;
StatusCode=StatusCode and $FD;
YSTART=000;
MWSUBSTART=000;
repeat
SearchStartRun(Parama.SizY,YStart,MWSUBSTART,ReturnCode);
if ReturnCode=00 then
begin
Y=YSTART;
MWSUB=MWSUBSTART;
PT=(Y-SizMWEI)*Parama.PT-PFDOrig;
MWSUBListToRun(MWSUBListPT(Y),MWSUB,Run);
MWSUBListToRun(XL);
MWSUBListToRun(XL);
Locate=(XL-SizMWEI)*Parama.PX-PFDOrig-Parama.MWL-PT-Parama.MWL-Parama.PI-Parama.PT);
for X=XL to RG do
begin
PX=(X-SizMWEI)*Parama.PX-PFDOrig;
Locate=(PX-PT-Parama.PI-Parama.PT);
if MWSUB=00 then
StatusCode=StatusCode or $01;
Response=RT);
PutCell(Layer,Parama.DR,X,Y,$05,$04);
end;
while not IsEndRun(Parama.SizY,Y,MWSUB) do
begin
SearchStartRun(Parama.SizY,Y,MWSUB,ReturnCode);
if ReturnCode=00 then
StatusCode=StatusCode or $02;
PT=(Y-SizMWEI)*Parama.PT-PFDOrig;
MWSUBListToRun(MWSUBListPT(Y),MWSUB,Run);
MWSUBListToRun(XL);
MWSUBListToRun(XL);
Locate=(XL-SizMWEI)*Parama.PX-PFDOrig-Parama.MWL-PT-Parama.PI-Parama.PT);
for X=XL to RG do
begin
PX=(X-SizMWEI)*Parama.PX-PFDOrig;
Locate=(PX-PT-Parama.PI-Parama.PT);
if MWSUB=00 then
StatusCode=StatusCode or $01;
Response=RT);
PutCell(Layer,Parama.DR,X,Y,$05,$04);
end;
end;
Inc(MWSUBSTART);
end
end;
else
begin
StatusCode=StatusCode or $02;
if MWSUB=00 then
StatusCode=StatusCode or $01;
end;
until (StatusCode and $02)=$02;
end;
end;
end;
end;

Function Inhibit0(I,C,Word):Boolean;
begin
if (I=Parama.EM0) and (I=Parama.EMax) and
Parama.PT=00000) and
(I=C-Parama.EM0)) then Inhibit0:=True
else Inhibit0:=False;
end;

Function Inhibit1(I,C,Word):Boolean;
begin
if (I=Parama.EM1) and (I=Parama.EMax) and
Parama.PT=00000) and
(I=C-Parama.EM1)) then Inhibit1:=True
else Inhibit1:=False;
end;

Function Inhibit2(I,C,Word):Boolean;
begin
if (I=Parama.EM2) and (I=Parama.EMax) and
Parama.PT=00000) and
(I=C-Parama.EM2)) then Inhibit2:=True
else Inhibit2:=False;
end;

Function Inhibit3(I,C,Word):Boolean;
begin
if (I=Parama.EM3) and (I=Parama.EMax) and
Parama.PT=00000) and
(I=C-Parama.EM3)) then Inhibit3:=True
else Inhibit3:=False;
end;

```

```

var ReturnCode:Byte;
I,SC,FD,Word;
SizeX,SizeY,MW,MH:Integer;
MCall:longint;
Layer:LayerType;

begin
  WriteLn('PPMAX : 形状加工しました。');
  SizeX:=Parame.SizeX div 2;
  SizeY:=Parame.SizeY div 2;
  MW :=Parame.MW div 2;
  MH :=Parame.MH div 2;
  SC:=000;
  repeat
    FI:=Parame.FM[0]-Parame.FI;
    for I:=000 to Parame.SizeX-1 do
      begin
        if InStr(10,SC) or
           InStr(11,SC) or
           InStr(12,SC) or
           InStr(13,SC) then
          begin
            StatusCode:=StatusCode and SF0;
            DrawImage(I,Parame.SizeX,Parame.SizeY,Layer,ReturnCode);
            if ReturnCode=000 then
              begin
                ClearGraph;
                OutScreen;
                Exit(000);
              end;
            MakeHashTable(Parame.SizeX,Parame.SizeY,Layer);
            GetMCall(MCall);
            StatusCode:=StatusCode and SF0;
            Load(FI);
            if NotIn=001 then
              StatusCode:=StatusCode or 001;
            if (InStr(11,SC)) and (MCall=000000) then
              begin
                StatusCode:=StatusCode and SF0;
                if Parame.OA='00' then
                  MakeUnit(32768
                    ,FI,Parame.ST0,Layer) else
                  if Parame.OA='00L' then
                    MakeUnit(32768
                    ,MW*(MH*3-SizeY*3)*Parame.FI,FI,Parame.ST0,Layer) else
                  if Parame.OA='00R' then
                    MakeUnit(32768*(MH*3-SizeY*3)
                    ,MW*(MH*3-SizeY*3)*Parame.FI,
                    ,FI,Parame.ST0,Layer) else
                  if Parame.OA='00W' then
                    MakeUnit(32768*(MH*3-SizeY*3)*Parame.FI
                    ,FI,Parame.ST0,Layer);
                if (StatusCode and 001)=001 then
                  begin
                    Write('PPMAX : 形状加工済');
                    WriteLn(SC,3);
                    WriteLn('ノユニット0の加工が完了しました。');
                  end;
                end;
              if Parame.FD=001 then
                begin
                  PullUp;
                  if NotIn=001 then
                    StatusCode:=StatusCode or 001;
                  if (StatusCode and 001)=001 then
                    WriteLn('');
                    StatusCode:=StatusCode and SF0;
                    Load(FI);
                  if NotIn=001 then
                    StatusCode:=StatusCode or 001;
                end;
              if (InStr(11,SC)) and (MCall=000000) then
                begin
                  StatusCode:=StatusCode and SF0;
                  if Parame.OA='01' then
                    MakeUnit(32768
                      ,FI,Parame.ST1,Layer) else
                  if Parame.OA='01L' then
                    MakeUnit(32768
                      ,MW*(MH*3-SizeY*3)*Parame.FI,FI,Parame.ST1,Layer) else
                  if Parame.OA='01R' then
                    MakeUnit(32768*(MH*3-SizeY*3)*Parame.FI
                      ,MW*(MH*3-SizeY*3)*Parame.FI,
                      ,FI,Parame.ST1,Layer) else
                  if Parame.OA='01W' then
                    MakeUnit(32768*(MH*3-SizeY*3)*Parame.FI
                      ,FI,Parame.ST1,Layer);
                  if (StatusCode and 001)=001 then
                    begin
                      Write('PPMAX : 形状加工済');
                      WriteLn(SC,3);
                      WriteLn('ノユニット1の加工が完了しました。');
                    end;
                end;
              if Parame.FD=001 then
                begin
                  PullUp;
                  if NotIn=001 then
                    StatusCode:=StatusCode or 001;
                  if (StatusCode and 001)=001 then
                    WriteLn('');
                    StatusCode:=StatusCode and SF0;
                    Load(FI);
                  if NotIn=001 then
                    StatusCode:=StatusCode or 001;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```



```

IF (InCh13(16,0C) AND (Wc11=000000)) THEN
  BEGIN
    STATUSCODE=STATUSCODE AND 8FD;
    IF PARAMS-UA="02" THEN
      MAKEWH1(32768
        32768 ,FS.PARAMS-ST3,Layer) ALSO
    IF PARAMS-UA="05" THEN
      MAKEWH1(32768
        32768*(M000 +S1aW7) *PARAMS-PT,PARAMS-ST3,Layer) ALSO
    IF PARAMS-UA="08" THEN
      MAKEWH1(32768*(M000 +S1aW7) *PARAMS-PT
        32768*(M000 +S1aW7) *PARAMS-PT,PARAMS-ST3,Layer) ALSO
    IF PARAMS-UA="09" THEN
      MAKEWH1(32768*(M000 +S1aW7) *PARAMS-PT
        32768 ,FS.PARAMS-ST3,Layer);
    IF (STATUSCODE AND 802)=802 THEN
      BEGIN
        WRITE('PPMARK : 形状部ノ番');
        WRITE(IN(0,3));
        WRITE(LAB('ユニット2の加工が完了しました。'));
      END;
    END;
    IF PARAMS-PW=801 THEN
      BEGIN
        PULLDP;
        IF BOTTOM=801 THEN
          STATUSCODE=STATUSCODE OR 801;
        IF (STATUSCODE AND 801)=801 THEN
          TEST;
          STATUSCODE=STATUSCODE AND 8FF;
          LAB(PP);
        IF BOTTOM=801 THEN
          STATUSCODE=STATUSCODE OR 801;
        END;
      END;
    IF (InCh13(16,0C) AND (Wc11=000000)) THEN
      BEGIN
        STATUSCODE=STATUSCODE AND 8FD;
        IF PARAMS-UA="02" THEN
          MAKEWH1(32768
            32768 ,FS.PARAMS-ST3,Layer) ALSO
        IF PARAMS-UA="05" THEN
          MAKEWH1(32768
            32768*(M000 +S1aW7) *PARAMS-PT,PARAMS-ST3,Layer) ALSO
        IF PARAMS-UA="08" THEN
          MAKEWH1(32768*(M000 +S1aW7) *PARAMS-PT
            32768*(M000 +S1aW7) *PARAMS-PT,PARAMS-ST3,Layer) ALSO
        IF PARAMS-UA="09" THEN
          MAKEWH1(32768*(M000 +S1aW7) *PARAMS-PT
            32768 ,FS.PARAMS-ST3,Layer);
        IF (STATUSCODE AND 802)=802 THEN
          BEGIN
            WRITE('PPMARK : 形状部ノ番');
            WRITE(IN(0,3));
            WRITE(LAB('ユニット3の加工が完了しました。'));
          END;
        END;
        PULLDP;
        IF BOTTOM=801 THEN
          STATUSCODE=STATUSCODE OR 801;
        IF (STATUSCODE AND 801)=801 THEN
          TEST;
          STATUSCODE=STATUSCODE AND 8FF;
          MAKEWH1(8TABLE);
        END;
      END;
      WRITE('PPMARK : 形状部ノ番');
      WRITE(IN(0,3));
      WRITE(LAB('の加工が完了しました。'));
      IN(0C);
      UNTIL 0C=PARAMS-S1aW7;
    END;
  END;

```

```

var ReturnCode:Byte;
DiskCode:Word;
Addr:Word;LongInt;

begin
  InIPort;
  CursorOff;
  InIPeeph;
  BeginOn;
  ClearGraph;
  SetParams;
  if MaxWall=Params.NA then
  begin
    Addr:Word:=LongInt [(Seg[SegBuf]*16
      LongInt [(Addr[SegBuf]])];
    Addr:Word:=Addr[SegBuf]+MaxWall*Params.NA;
    SegBuf:=Ptr(Addr[SegBuf] div 16,Addr[SegBuf] mod 16);
  end
  else
  begin
    CursorOn;
    Ball(807);
  end;
  if Params.DD='A' then DrvCode:=SI (Max[1000,1208]) else
  if Params.DD='B' then DrvCode:=SI (Max[1000,1208]) else
  if Params.DD='C' then DrvCode:=SI (Max[1000,1208]) else
  if Params.DD='D' then DrvCode:=SI (Max[1000,1208]) else
  if Params.DD='E' then DrvCode:=SI (Max[1000,1208]) else
  if Params.DD='F' then DrvCode:=SI (Max[1000,1208]) else
  if Params.DD='G' then DrvCode:=SI (Max[1000,1208]) else
  if Params.DD='H' then DrvCode:=SI (Max[1000,1208]);
  ReadDirive(DrvCode,ReturnCode);
  if (ReturnCode=0) or
  (ReturnCode=10) then
  begin
    CursorOn;
    Ball(80);
  end
  else
  if ReturnCode=40 then
  begin
    CursorOn;
    Ball(80);
  end;
  OpenDirA(ReturnCode);
  if ReturnCode=00 then
  begin
    CursorOn;
    Ball(80);
  end;
  ReadDirive(DiskCode,ReturnCode);
  if ReturnCode=00 then
  begin
    CursorOn;
    Ball(80);
  end;
  if DiskCode=FFFF then
  begin
    CursorOn;
    Ball(804);
  end;
  ReadDirive(Params.Slow,ReturnCode);
  if ReturnCode=00 then
  begin
    CursorOn;
    Ball(80);
  end;
  ReadDirive(Params.SlowF,ReturnCode);
  if ReturnCode=00 then
  begin
    CursorOn;
    Ball(80);
  end;
  ReadDirive(Params.Slow,ReturnCode);
  if ReturnCode=00 then
  begin
    CursorOn;
    Ball(80);
  end;
  ReadDirive(Params.Slow,ReturnCode);
  if ReturnCode=00 then
  begin
    CursorOn;
    Ball(80);
  end;
  Clear;
  AdjustOrigin;
  MakeLine;
  MakeForm;
  TextFinal;
  if Params.LBW=01 then
  LowOff;
  ClearGraph;
  CursorOn;
  end.

```

```

-----
FFTESTE.PAS      1080bytes  1993-08-24  2:00:00
FFTESTE.EXE      2752bytes  1993-08-24  2:00:00
-----
Printed on 1994-09-02
-----

```

```

program FFTestE;
(* P F システムソフトウェア
 * 加工検査機光沢調整用プログラム Version 2.00 1993/08/24 *)
($D-.L)

uses $DCLib,FFDevLib;

var ValCode:Integer;
    CycleTime,ExposureTime,TotalTime:LongInt;

begin
  InitPort;
  if (ParamCount<=2) and (ParamCount<=3) then
    Halt($G1);
  Val(ParamStr(1),CycleTime,ValCode);
  if (ValCode<=$0000) or (CycleTime<=0000000) then
    Halt($G1);
  Val(ParamStr(2),ExposureTime,ValCode);
  if (ValCode<=$0000) or (ExposureTime<=0000000) or
    (ExposureTime<=CycleTime) then
    Halt($G1);
  if ParamCount=3 then
    begin
      Val(ParamStr(3),TotalTime,ValCode);
      if (ValCode<=$0000) or (TotalTime<=0000000) then
        Halt($G1);
    end
  else
    TotalTime:=0000000;
  repeat
    ExposureOn;
    Delay(ExposureTime);
    ExposureOff;
    Delay(CycleTime-ExposureTime);
    if TotalTime<=0000000 then
      begin
        TotalTime:=TotalTime-CycleTime;
        if TotalTime<=0000000 then
          Halt($G1);
        end;
      end;
  until NotOn=$G1;
end.

```

```

-----
PFTTESTS.PAS      2033bytes  1993-08-24  2:00:00
PFTTESTS.EXE     2484bytes  1993-08-24  2:00:00
-----

```

Printed on 1994-09-22

```

program PFTtests;
(* P F システムソフトウェア
 * 加工検査生産系調程用プログラム Version 2.00 1993/08/24 *)
($D-,L-)

uses StdLib,PFDevLib;

var NStep,MSStep,CtrlCode:Byte;
    FE,FF,FS,FMMin,FMMax,FDMin,FDMax,FDMinL,FDMaxL:Word;
    ValCode:Integer;
    StepTime,RelaxationTime:LongInt;

begin
  IndStep;
  if ParamCount<4 then
    Halt(001);
  if UCaseStr(ParamStr(1))='E' then CtrlCode:=000 else
  if UCaseStr(ParamStr(1))='S' then CtrlCode:=001 else
  if UCaseStr(ParamStr(1))='F' then CtrlCode:=002 else
  if UCaseStr(ParamStr(1))='P' then CtrlCode:=003 else
  if UCaseStr(ParamStr(1))='L' then CtrlCode:=004 else
  if UCaseStr(ParamStr(1))='E' then CtrlCode:=005 else
    Halt(001);
  Val(ParamStr(2),NStep,ValCode);
  if (ValCode<>0000) or (MSStep=00) then
    Halt(001);
  Val(ParamStr(3),RelaxationTime,ValCode);
  if (ValCode<>0000) or (RelaxationTime<>00000) then
    Halt(001);
  Val(ParamStr(4),StepTime,ValCode);
  if (ValCode<>0000) or (StepTime<>000000) then
    Halt(001);
  FE :=32768;
  FF :=32768;
  FS :=00001;
  FMMin:=00001;
  FMMax:=65535;
  FDMin:=65535;
  FDMax:=65535;
  repeat
    for MSStep:=1 to NStep do
      case CtrlCode of
        000 (MoveDcFF(FE,FMMin,FMMax,StepTime);
          001 (MoveDcFF(FS,FMMin,FMMax,StepTime);
          002 (MoveDcFF(FF,FMMin,FMMax,StepTime);
          003 (MoveDcFF(FE,FMMin,FMMax,StepTime);
          004 (MoveDcFF(FS,FMMin,FMMax,StepTime);
          005 (MoveDcFF(FF,FMMin,FMMax,StepTime);
        end;
      for NStep:=1 to MSStep do
        case CtrlCode of
          000 (MoveDcFE(FE,FMMin,FMMax,StepTime);
          001 (MoveDcFF(FS,FMMin,FMMax,StepTime);
          002 (MoveDcFF(FF,FMMin,FMMax,StepTime);
          003 (MoveDcFF(FE,FMMin,FMMax,StepTime);
          004 (MoveDcFF(FS,FMMin,FMMax,StepTime);
          005 (MoveDcFF(FF,FMMin,FMMax,StepTime);
        end;
      Delay(RelaxationTime);
    until NotDone=001;
  end;
end.

```

使用機材・材料一覧

実験用加工装置に用いた機材や実験に用いた材料について、その型番・品番を以下に示す。

- ・パーソナルコンピュータ
 - 日本電気 パーソナルコンピュータ PC-9801VM
- ・拡張I/Oボード
 - コンテック TTLレベル双方向デジタル入出力ボード
PIO-48W (98)
- ・He-Cdレーザ
 - 金門電気 He-Cdレーザ CD3041R
He-Cdレーザ用電源 CD1601A
- ・Ar⁺レーザ
 - Omnichrome Ar⁺レーザ 532-10B
Ar⁺レーザ用電源 150
- ・光学定盤
 - シグマ光機 光学実験台 Σ-11AHT-(4)L
- ・位置決めステージ
 - シグマ光機 精密型X軸ステージ STM50-XA
精密型Z軸ステージ STM50-ZA
- ・集光レンズ
 - シグマ光機 球面平凸レンズ-石英ガラス SLSQ-05-08P
- ・窓板
 - シグマ光機 平行平面基板-合成石英 OPSQ-25S2.4-1-5
- ・反射鏡
 - シグマ光機 アルミ平面ミラー TFA-1015R05-10
- ・その他の光学部品
 - シグマ光機 小型ミラーホルダー Σ-50-(1)F
X軸ネジ送りステージ Σ-203-(2)
XZ軸ネジ送りステージ Σ-603-(2)

・ステッピングモータドライバ

オリエンタルモーター 5相ステッピングモータードライバ

SPD5517

・フッ素樹脂粘着テープ

住友スリーエム Scotchテフロンテープ #5490

・光硬化性樹脂

三洋化成工業 UV-854

ディーメック SCR500

・光硬化性接着剤

東亜合成化学工業 アロンタイト VL-2302

・染料

藍熊染料 メタノール フワスト エロー

付録——製造元・販売元・工場一覧

機材製造元・販売元

- ・日本電気株式会社 (パーソナルコンピュータ)
〒108-01 東京都港区芝5-7-1
(パソコンインフォメーションセンター)
電話 03-3452-8000
- ・株式会社コンテック (I/Oボード)
〒105 東京都港区芝2-12-10 (東京営業所)
電話 03-3769-1061
- ・金門電気株式会社 (He-Cdレーザ)
〒173 東京都板橋区板橋1-53-2
電話 03-5248-4811
- ・Omnichrome Inc., Ltd. (Ar⁺レーザ)
13580 Fifth Street, Chino, CA 91710, U.S.A.
電話 +01-909-627-1594
- ・株式会社オーク製作所 (高圧水銀ランプ)
〒182 東京都調布市調布ヶ丘3-34-1
電話 0424-87-0161
- ・シグマ光機株式会社 (光学台・位置決めステージ・光学部品)
〒175 東京都板橋区成増1-30-13 (東京営業所)
電話 03-3975-0875
- ・オリエンタルモーター株式会社 (ステッピングモータドライバ)
〒111 東京都台東区小島2-21-11 (東京支店)
電話 03-3864-0020

材料製造元・販売元

- ・三洋化成工業株式会社 (光硬化性樹脂)
〒103 東京都中央区日本橋本町1-5-6 (東京支社)
電話 03-3279-3031

・株式会社ディーメック (光硬化性樹脂)

〒104 東京都中央区築地2-11-24

電話 03-5565-6661

・東亜合成化学工業株式会社 (光硬化性接着剤)

〒105 東京都港区西新橋1-14-1

電話 03-3597-7257

・藍館染料株式会社 (染料)

〒111 東京都台東区雷門1-5-1

電話 03-3841-5760

加工工場

・株式会社吉田SKT (フッ素樹脂コーティング)

〒116 東京都荒川区町屋5-4-6

電話 03-3895-0351

付録—機材仕様・材料特性—覧

CD-3041R (He-Cdレーザー) の仕様

Specification	Min.	Typ./Nom.	Max.	Unit
Output		10		mW
Wave length		325		nm
Output noise			6	% p-p
Beam diameter		1.0		mm
Beam divergence		0.6		mrad

532-10B (Ar⁺レーザー) の仕様

Specification	Min.	Typ./Nom.	Max.	Unit
Output		10		mW
Wave length		All blue lines*		nm
Output noise			0.5	% RMS
Beam diameter		0.61		mm
Beam divergence		1.0		mrad

* 455, 458, 466, 473, 477, 488, 497 nm

UV-854 (光硬化性樹脂) の特性

Characteristic		Typ.	Unit
Liquid state	Density	1140	kg/m ³
	Viscosity	1115	cP
Solid state	Density	1210	kg/m ³
	Young's modulus	5.5×10 ⁴	MPa
	Tensile strength	46	MPa
	Tensile elongation	-	%
	Bending modulus	-	MPa
	Bending strength	-	MPa
	Hardness	Hs84	
Solidification	Volume shrinkage	6	%

SCR-500 (光硬化性樹脂) の特性

Characteristic		Typ.	Unit
Liquid state	Density	1110	kg/m ³
	Viscosity	850	cP
Solid state	Density	-	kg/m ³
	Young's modulus	1570	MPa
	Tensile strength	60	MPa
	Tensile elongation	10	%
	Bending modulus	2600	MPa
	Bending strength	101	MPa
	Hardness	Hr115	
Solidification	Volume shrinkage	-	%

VL-2302 (光硬化性接着剤) の特性

Characteristic		Typ.	Unit
Liquid state	Density	1150	kg/m ³
	Viscosity	2000	cP
Solid state	Density	1222	kg/m ³
	Young's modulus	13.9	MPa
	Tensile strength	-	MPa
	Tensile elongation	-	%
	Bending modulus	-	MPa
	Bending strength	-	MPa
	Hardness	Hs81	
Solidification	Volume shrinkage	9.85	%

付録——研究発表一覧

雑誌発表

- (1) 高木太郎, 中島尚正, 光造形法による微細加工の可能性, 機論 Vol. 59 No. 560 C (1993), pp. 287-292.
- (2) T., Takagi, T. and Nakajima, N., Photoforming Applied to Fine Forming, JSME International Journal (in progress).

講演発表 (投稿)

- (3) Mizoguchi, H., Ando, M., Mizuno, T., Takagi, T. and Nakajima, N., Design and Fabrication of Light Driven Micropump, Conference Proceedings of MEMS '92 (1992), pp 31-36.
- (4) 高木太郎, 中島尚正, 光造形法による微細加工の可能性に関する研究, 第4回マイクロマシン・シンポジウム資料 (1992), pp. 9-12.
- (5) 高木太郎, 中島尚正, 光造形法による微細加工, 日本機械学会第2回設計工学・システム部門講演会講演論文集 (1992), pp. 263-266.
- (6) T., Takagi, T. and Nakajima, N., Photoforming Applied to Fine Machining, Conference Proceedings of MEMS '93 (1993), pp 173-178.
- (7) T., Takagi, T. and Nakajima, N., Architecture Combination by Micro Photoforming Process, Conference Proceedings of MEMS '94 (1994), pp 211-216.

講演発表 (依頼)

- (8) 高木太郎, 中島尚正, 光造形法による微細加工, 第3回光造形システムシンポジウム資料 (1992), pp. 37-41.
- (9) T., Takagi, T. and Nakajima, N., Micromachine Fabrication by Photoforming Process, Conference Proceedings of RadTech Asia '93 (1993), pp 451-456.
- (10) 高木太郎, 光造形法によるマイクロマシンの製作 —現状と展望—, 第39回ラドテック研究会講演会資料 (1994), pp. 17-27.

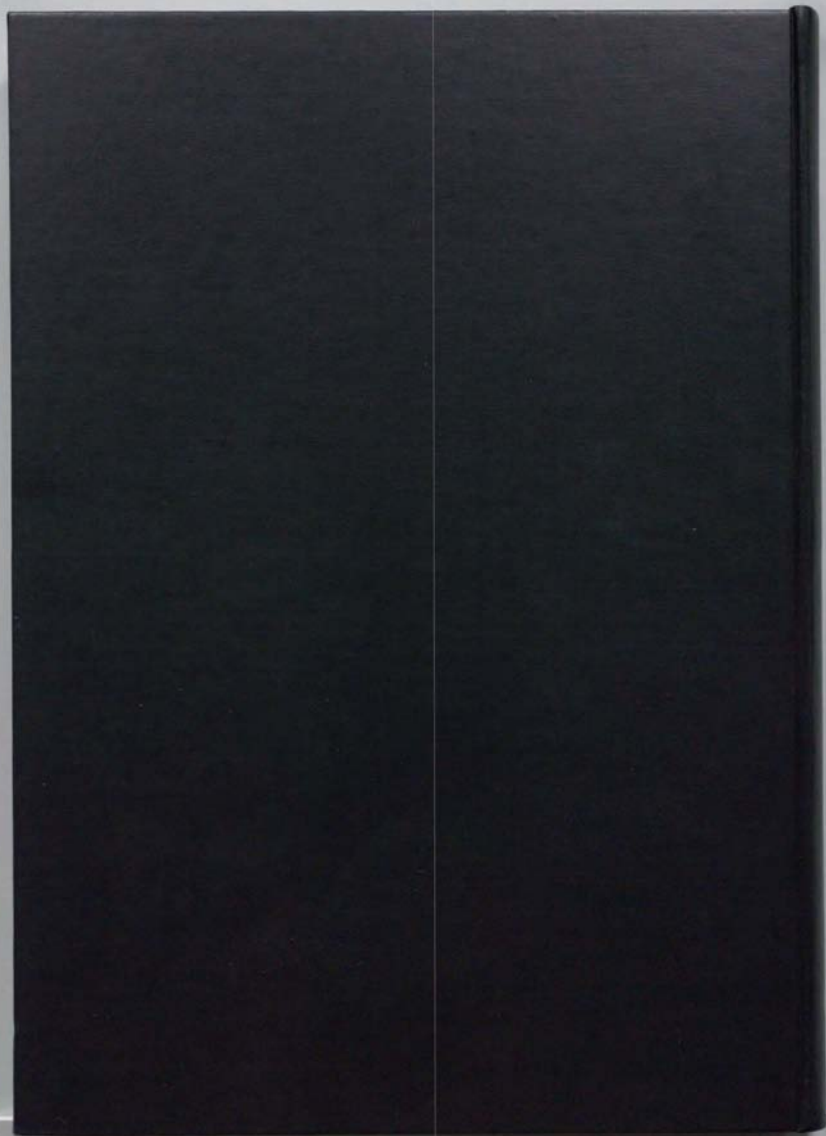
以上

1～212ページ 完

学位論文

平成6年12月20日 提出

27053 高木太郎





Kodak Color Control Patches

Blue Cyan Green Yellow Red Magenta White 3/Color Black

Kodak Gray Scale

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19

C Y M

© Kodak, 2007. TM Kodak