

電子情報工学専攻

Advanced Course of Electronic and Information Engineering

平成6年度 専攻科特別研究論文

題 目

S_nO₂系半導体ガスセンサを用いた
匂いセンシングシステムの開発

The development of Smell sensing system
using S_nO₂ semiconductor gas detectors

指導教官名 成田 紘一 教授

論文提出者名 桐島 俊之

国立奈良工業高等専門学校・専攻科

Faculty of Advanced Engineering at Nara National College of Technology

S.O₂系半導体ガスセンサを用いた 匂いセンシングシステムの開発

指導教官名 成田 紘一 氏名 相島 俊之

1. はじめに

近年、多種多様なガス検出素子が開発・改良され、それらを使った工学的匂い識別に関する研究が行われるようになってきた。現在最も普及している半導体ガスセンサは、多くの種類のガスに反応することから匂いセンシングシステムのガスセンサとして有望視されている。匂いセンシングシステムとは、匂いの測定・学習・識別をコンピュータを利用して行うものである。半導体ガスセンサのガス検出感度は、ガス吸着の行われる半導体表面の温度に大きく依存する。本研究では、素子内部のヒータをON/OFF制御することにより得られる匂い測定パターンを、相互作用可変法とニューラルネットワークにより識別する匂いセンシングシステムを開発した。

2. 測定方法

半導体ガスセンサとしてアンモニアガスセンサ(TGS824)・硫化水素ガスセンサ(TGS825)・メタンガスセンサ(TGS813)・アルコールガスセンサ(TGS823)の4種類のそれぞれ特性が異なるガスセンサを使用し、匂い測定装置を製作した。匂いサンプルとして、清酒(2種類)・吟醸酒(2種類)・焼酎(泡盛)・VSOP(ブランド)・エタノール・ハッカ油・ポカリスウェット・マスカットジュース・コーヒー・ウーロン茶の12種類を選定した。これらの匂いサンプルを、製作した匂い測定装置とパーソナルコンピュータ(PC9801DS)を使用して各匂いサンプルにつき10個ずつ(訓練サンプル5個・テストサンプル5個)の匂い測定パターンを得た。

3. 測定結果

図1に多変量解析の指標であるF比(分散比)をアルコールガスセンサについて計算した結果を示す。図1に示すように通常の測定方法(行程1)よりも匂い測定パターン間の分離が良い測定点がヒータ電源のON/OFF制御により得られることが分かった。

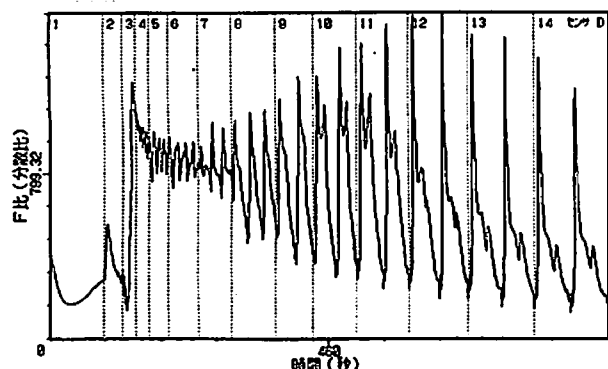


図1. アルコールガスセンサの分散比分布

4. 識別方法および識別結果

次に考案した相互作用可変法の処理手順を示す。

- 1) 匂い測定パターンの分散比分布を求めて、最も分散比が大きくなる測定点より測定電圧値の集合(匂い応答パターン)を得る。
- 2) 匂い応答パターンを構成する各要素の分散が正規分布で表現できると仮定して各要素に重みと影響曲線の勾配を与える。
- 3) 与えた分布により匂い応答パターンの各要素が正しく識別されない場合、最も大きな影響を及ぼしている要素の学習要素を調整することで各要素が正しい匂いカテゴリに属するように学習する。
- 4) 最終的に得られる匂い識別用標準パターンを利用して匂いを識別する。

図2に匂い標準パターンよりアルコールガスセンサの匂い識別領域を求めた結果を示す。

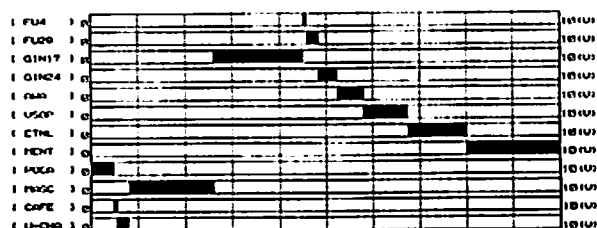


図2. アルコールガスセンサの匂い識別領域

図3に相互作用可変法とニューラルネットワークを組み合わせ、テストサンプルを識別した結果を示す。

	FLU	FLU2	GIN17	GIN24	FWA	USOP	ETML	FENT	POCA	PRSC	CAFE	U-CHA
1	●	●	●	●	●	●	●	●	●	●	●	●
	0.124	0.056	0.857	0.352	0.434	0.676	0.782	0.837	0.945	0.813	0.654	0.837
2	●	●	●	●	●	●	●	●	●	●	●	●
	0.200	0.189	0.999	0.201	0.759	0.708	0.931	0.615	0.825	0.812	0.834	0.859
3	●	●	●	●	●	●	●	●	●	●	●	●
	0.219	0.231	0.523	0.292	0.855	0.539	0.888	0.888	0.839	0.893	0.893	0.825
4	●	●	●	●	●	●	●	●	●	●	●	●
	0.128	0.497	0.854	0.153	0.868	0.851	0.808	0.989	0.949	0.895	0.882	0.852
5	●	●	●	●	●	●	●	●	●	●	●	●
	0.143	0.238	0.810	0.233	0.738	0.798	0.997	0.991	0.798	0.824	0.945	0.858

注) 円の大きさは識別確度、×印は誤識別を表す

図3. テストサンプルの識別結果

5. むすび

本研究では、市販の半導体ガスセンサを利用した匂いセンシングシステムを構築することで、選定した12種類の匂いサンプルの識別を行った。今後は、匂い分子の重量的性質に関する情報が得られる水晶振動子アレイ等を本システムで採用することで、微妙な匂いの違いをより確実に識別できるように改善されることが望まれる。

SnO_2 系半導体ガスセンサを用いた 匂いセンシングシステムの開発

The development of Smell Sensing System
using SnO_2 semiconductor gas detectors

桐島 俊之

Toshiyuki KIRISHIMA

奈良工業高等専門学校・電子情報工学専攻・大和郡山市矢田町22 (〒639-11)
Faculty of Advanced Engineering at Nara National College of Technology,
22 Yata-cho, Yamatokoriyama-shi, Nara 639-11, Japan

ABSTRACT : Recent diversification and improvements of gas detecting devices have led to the increasing number of research activities in smell sensing technology. Among others, semiconductor gas sensors whose use is not limited to the detection of specified gases are expected to play a major role in constructing a smell sensing system. Smell sensing system was proposed by MORIIZUMI et al. and refers to a computer system that carries out the measurement, learning and discrimination of various kinds of odors. Newly-developed smell sensing system employs ON/OFF switching control of the heaters. As for the discrimination of odors, Variable Interaction Method, an original method which is based upon Bayesian discriminant algorithms was proposed. In the process of this research, it was found that proposed method and neural network share an interdependent nature. By taking advantage of the nature of both methods the smell discrimination was performed. In addition, suitability of gas detectors for selected odor samples was estimated as well as the selection of odor samples for gas sensors. The effectiveness of the method in this research is demonstrated by the results of experiments.

【 目 次 】

第 1 章	まえがき	1
第 2 章	生体嗅覚系について	3
2. 1	匂いの種類とその受容機構に関する種々の仮説	3
2. 2	匂い受容機構への工学的アプローチ	4
第 3 章	半導体ガスセンサについて	5
3. 1	半導体ガスセンサの検出原理	5
3. 2	SnO_2 系半導体ガスセンサの利用方法	7
第 4 章	匂いセンシングシステム	8
4. 1	匂い測定装置の製作	8
4. 2	匂いセンシングシステムの各種機能	10
第 5 章	匂い測定手順	11
第 6 章	匂い識別処理	12
6. 1	相互作用可変法	13
6. 2	相互作用可変法における評価値の計算方法	15
6. 3	相互作用可変法における学習方法	16
6. 4	相互作用可変法により匂い識別領域を求める方法	16
6. 5	匂い識別領域から匂いごとに適したガスセンサを選択する方法	17
6. 6	本研究で使用するニューラルネットワークについて	18
6. 7	相互作用可変法とニューラルネットワークを組み合わせる方法	19

第7章 実験結果	20
7.1 12種類の訓練用匂いサンプルを測定した結果	20
7.2 匂い測定パターンから各ガスセンサのF比特性を計算した結果	32
7.3 匂い測定パターンから匂い応答パターンを求めた結果	33
7.4 各ガスセンサの匂い標準パターンを求めた結果	44
7.5 各ガスセンサの匂い識別領域を求めた結果	45
7.6 相互作用可変法とニューラルネットワークの学習曲線を求めた結果	46
7.7 テストサンプルの識別結果	48
7.8 訓練サンプルの識別結果	50
7.9 相互作用可変法とニューラルネットワークを組み合わせた際の識別結果	52
7.10 各種匂いサンプルに適したガスセンサの選択結果	53
第8章 考察及び検討	54
8.1 匂い測定装置に関する考察・検討	54
8.2 半導体ガスセンサに関する考察・検討	55
8.3 相互作用可変法による匂いの学習・識別に関する考察・検討	57
8.4 ニューラルネットワークによる匂いの学習・識別に関する考察・検討	58
8.5 相互作用可変法とN.N.を組み合わせた場合の識別結果の考察・検討	59
第9章 むすび	60
【 謝辞 】	
【 参考文献 】	
【 業績 】	
【 付録 】	

第 1 章 まえがき

人間はその生活を営んでいく上でいわゆる五感（視覚・聴覚・触覚・味覚・嗅覚）を利用している。これらの感覚の中で視覚・聴覚・触覚は物理量を対象としている一方で味覚・嗅覚は化学量を対象としている。生命体において化学量を対象とする感覚は極めて重要な役割を果たしている。とりわけ、食物・環境等における危険を検知する嗅覚の役割は大きい。匂い識別の自動化および匂いの定性的・定量的評価は、防災・環境計測・食品管理など我々の日常生活と直接・間接的に関わる分野において、その実現が望まれている。これまでガス分析には、一般にガスクロマトグラフィー法が用いられているが、この方法では数時間もの測定時間を要する上に、得られる莫大な数の分子種のピーク群は簡便な匂いの識別には適さない。

1937年に細谷・吉田が世界に先駆けて電気生理学の立場から嗅覚の研究を始め、その際に犬の嗅粘膜を摘出して嗅電図（Electro-OlfactoGram : EOG）を記録した。このような具体的な生体実験を通じて嗅覚機構の解明を試みる電気生理学に立脚した研究により、一つの嗅細胞は複数の匂いに応答することが分かった。これは生体における匂いの識別は複数種の選択性の低い嗅細胞からの出力を脳がパターン認識しているという仮説を支持するものである。このように多数のセンサからの出力をもとに化学物質を識別する技術はケモメトリックスと呼ばれ、化学センサを実現する際の情報認識法として有効である³⁾。

最初にパターン認識型匂いセンサを考案・試作したのは、池上等¹⁷⁾である。彼等は、厚膜作製技術を用いて6種類の半導体ガスセンサをアルミナ基板上に集積化し、その応答パターンにより有機ガスの種類を識別した。しかし、複合臭に関しては十分な識別能力が得られなかった。つまり微妙な匂いの違いを識別するには、さらに高度なアルゴリズム等に基づく匂い識別系が必要となる。森泉等は匂い検出部に水晶振動子、匂い学習・匂い識別部にニューラルネットワークを使用した匂いセンシングシステムを開発し、ケモメトリックな手法による匂い識別が複合臭に対しても有効であることを示した¹⁾。石淵等は複数の半導体ガスセンサからの応答パターンを判別分析法によりガス種の分類を行った²⁾。さらに小西等はファジー推論を用いてガス種の識別を行っている¹⁸⁾。このように匂いの識別アルゴリズムに関する研究は、センサ素子の開発と並行して増加する傾向にある。

半導体ガスセンサは、1968年にフィガロ技研により実用化されて以来、家庭用ガス警報器などに利用され、現在最も普及しているガスセンサである。半導体ガスセンサは感度が高い反面、ガス選択性が悪く、さらにセンサの信頼性において重要なファクタである

再現性、環境依存性等に関して解決すべき問題点がある。しかし、半導体ガスセンサのガス選択能力の低さを利用すればケモメトリックな手法により複合臭などの匂いガスを識別ができる可能性が高く、実現すれば低コストで保守をあまり必要としない匂いセンシングシステムが実現できる。

そこで本研究では、現在市販されている半導体ガスセンサ¹⁶⁾を用いた匂いセンシングシステムを開発することにした。開発した匂いセンシングシステムは、センサ部に4種類の S_nO_2 系半導体ガスセンサを用いて、その各電圧応答パターンをコンピュータに取り込み、そのパターンから匂い識別処理を行う、いわゆるパターン認識型匂いセンシングシステムである。本研究では限定サンプル・限定環境における匂いの識別を対象とし、独自の匂い測定装置を開発した。また、 S_nO_2 系半導体ガスセンサの検出感度が素子表面の温度に依存する特徴を利用して、センサ内部のヒータ電源のON/OFF制御を行うことにより、選定した12種類のサンプルについて従来の測定方法よりも匂い応答パターンの分離の良い測定点が得られるのかどうかを調べた。匂いの学習・識別方法として相互作用可変法が考案され、その結果を客観的に比較するためにニューラルネットワークによる匂いの学習・識別も行った。さらに、相互作用可変法により得られた識別用標準パターンから求めた各ガスセンサの識別領域から、選定した12種類の匂いサンプルに適したガスセンサの選択と、各ガスセンサが識別を得意としている匂いサンプルの選択を行った。

本論文では、第2章で生体嗅覚系について必要な範囲内の説明を行い、第3章で半導体ガスセンサに関連する事項について述べ、匂いセンシングシステムの機能などについては第4章で説明する。第5章と第6章で匂いの測定・学習・識別方法について述べて、各種の実験結果については第7章に示す。第8章と第9章では匂いセンシングシステムの各要素について考察・検討を加えて今後の課題等について述べる。

第 2 章 生体嗅覚系について

2. 1 匂いの種類とその受容機構に関する種々の仮説

匂いの実体は、「揮発性物質が分子の熱運動によって気化したもの」であり、すなわち化学物質である。匂いを識別することは、化学物質を識別することを意味している。つまり分子レベルでの匂い分子の性質（電氣的・重量的・化学的性質）の識別が必要となる。分子の違いが匂いの違いであると考えた場合、我々の周囲には膨大な種類の匂いが存在していることになる。

これまでに行われてきている匂いの種類に関する研究により、その数は数十万種ともいわれる。匂いに基本臭があると仮定して、樟脳臭・ジャ香臭・花臭・ハッカ臭・エーテル臭・刺激臭・腐敗臭の 7 種類が基本臭であるとする報告⁵⁾ もあれば 47 種以上もの基本臭があるとする報告もある⁴⁾。嗅覚は個人差が大きいので客観的にある種の基本臭の設定を行うことは現在でも困難である。表 1 に生体における匂い受容機構の諸説を示す。

表 1. 生体における匂い受容機構の諸説

諸説名	概要
振動説	匂いは、光や音と同じように振動として伝わり、嗅粘膜を刺激するという説
化学説	匂い分子が嗅細胞の受容膜との間で化学反応を起こして刺激するという説
酵素説	受容器の表面にある酵素系に匂い分子が与える影響によって匂いの違いが生じるという説
立体構造説	マンクリーフとアモーレによる。匂い分子と受容器は鍵と鍵穴の関係にあるという「鍵と鍵穴」説で匂いの違いは匂い分子の外形とそのディメンジョンによって決まるという説

以上の諸説は実験的根拠に乏しく不完全なものであることから、これらの説により匂い受容機構を厳密に説明することは現在でも困難である。現在では電気生理学の研究から、受容器細胞内で生じる脱分極の波が神経インパルスとなることで匂いの伝達が行われていると考えられている。図 1 に生体嗅覚の伝達経路を示す¹⁾。

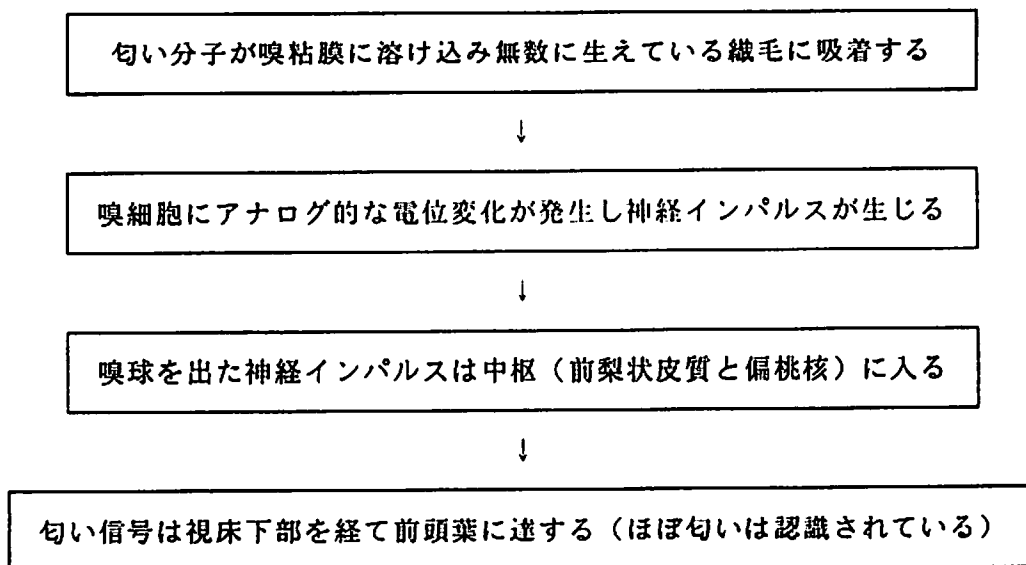


図 1. 生体嗅覚の伝達経路

2.2 匂い受容機構への工学的アプローチ

動作原理・センサ素子の価格・実験設備等を考慮して本研究では半導体ガスセンサを使用することにした。半導体ガスセンサを使用する場合の匂い識別方式と生体における匂い受容・認識過程を対比した模式図を図 2 に示す。

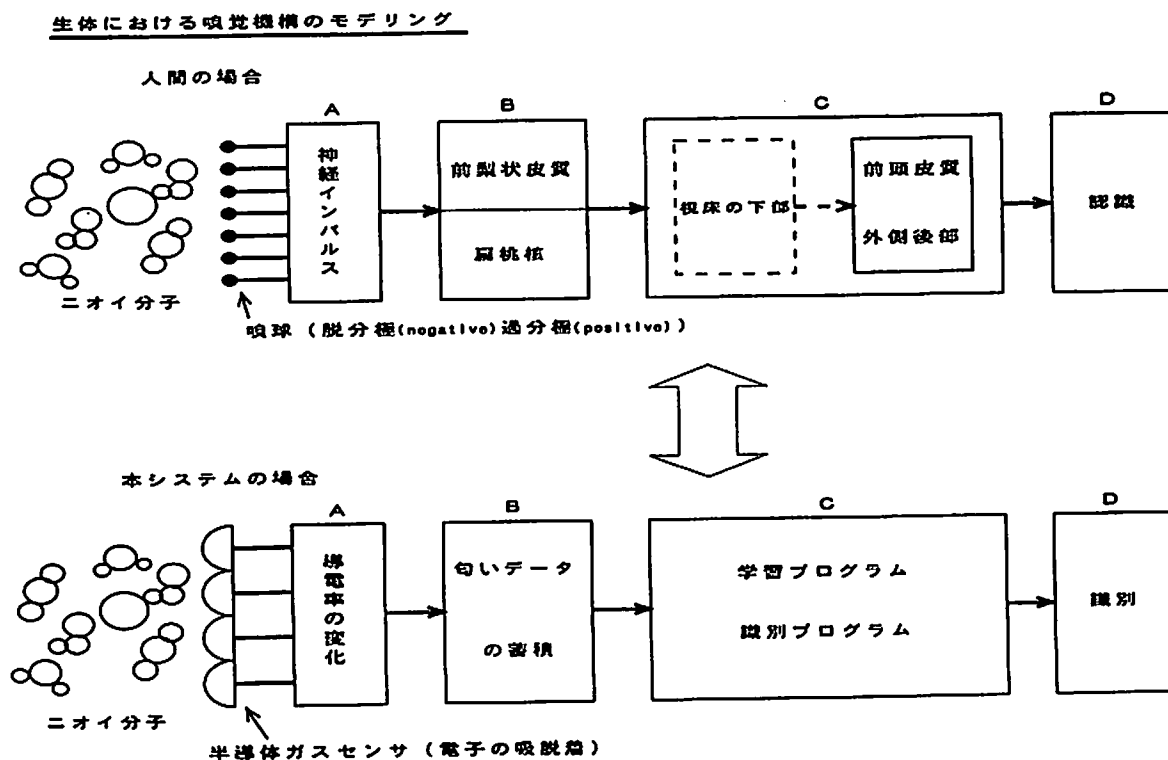


図 2. 匂い受容・認識過程の対比

第3章 半導体ガスセンサについて

3.1 半導体ガスセンサの検出原理¹¹⁾

SnO_2 半導体ガスセンサは、 SnO_2 微結晶の結晶粒界における電位障壁がガス等の雰囲気により変化してセンサの導電率が変化する現象を利用する^{12, 14)}。図3に半導体ガスセンサの検出機構を示す。まずセンサ部の素子が加熱されて定常状態になった後、素子表面へガスが接して吸着すると、まず吸着分子は表面を自由に拡散し（物理吸着）、その運動エネルギーを失う。この間に一部の分子は蒸発し、残りの分子は熱解離して吸着席へ局在（化学吸着）する。

素子の仕事関数が吸着分子の電子親和力より小さければ、吸着分子は素子から電子を奪って陰イオン吸着する。この陰イオン吸着の傾向を持つ（酸化形ガスまたは電子受容性ガスと呼ばれる）ものには、 O_2 、 NO_x などがある。素子の仕事関数が吸着分子のイオン化エネルギーよりも大きければ、吸着分子は素子へ電子を与え陽イオン吸着する。この陽イオン吸着の傾向を持つ（還元形ガスあるいは電子供与性ガスと呼ばれる）ものには、 H_2 、 CO 、炭化水素、アルコール類等がある。

酸化形ガスがn形半導体へ、還元形ガスがp形半導体へ吸着するとキャリアを減少させて電気抵抗を増加させる。一方、還元形ガスがn形、酸化形ガスがp形半導体へ吸着するとキャリアを増加させて電気抵抗を減少させる。図4に SnO_2 半導体ガスセンサへ還元形ガスが吸着した場合の内部結晶粒界における電位障壁の変化を図示する。また、n形半導体へガスが接した場合に起こる素子抵抗値変化のグラフを図5に示す。

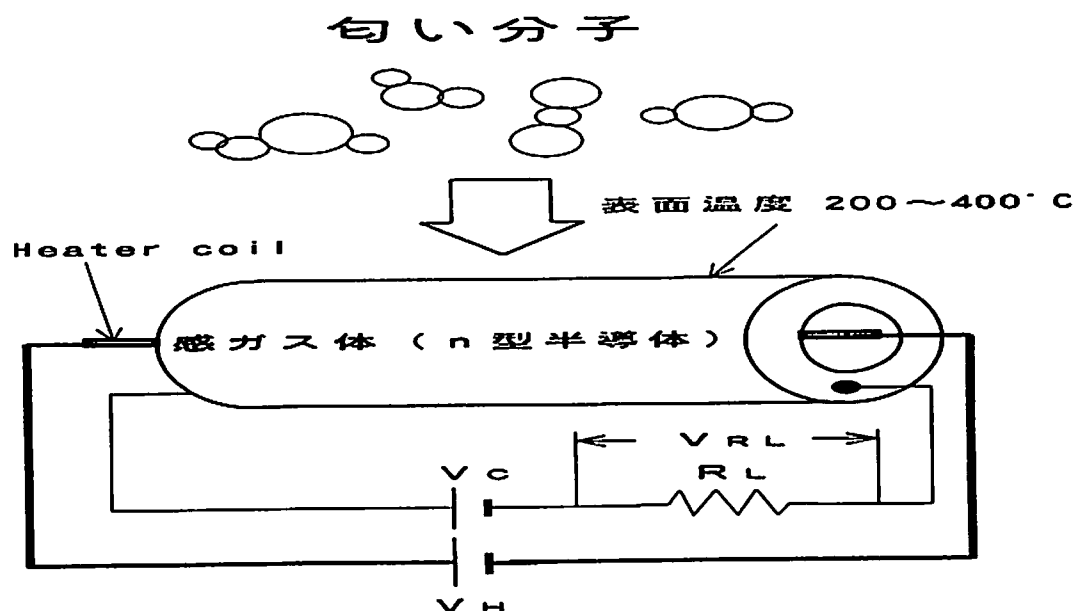


図3. 半導体ガスセンサのガス検出機構

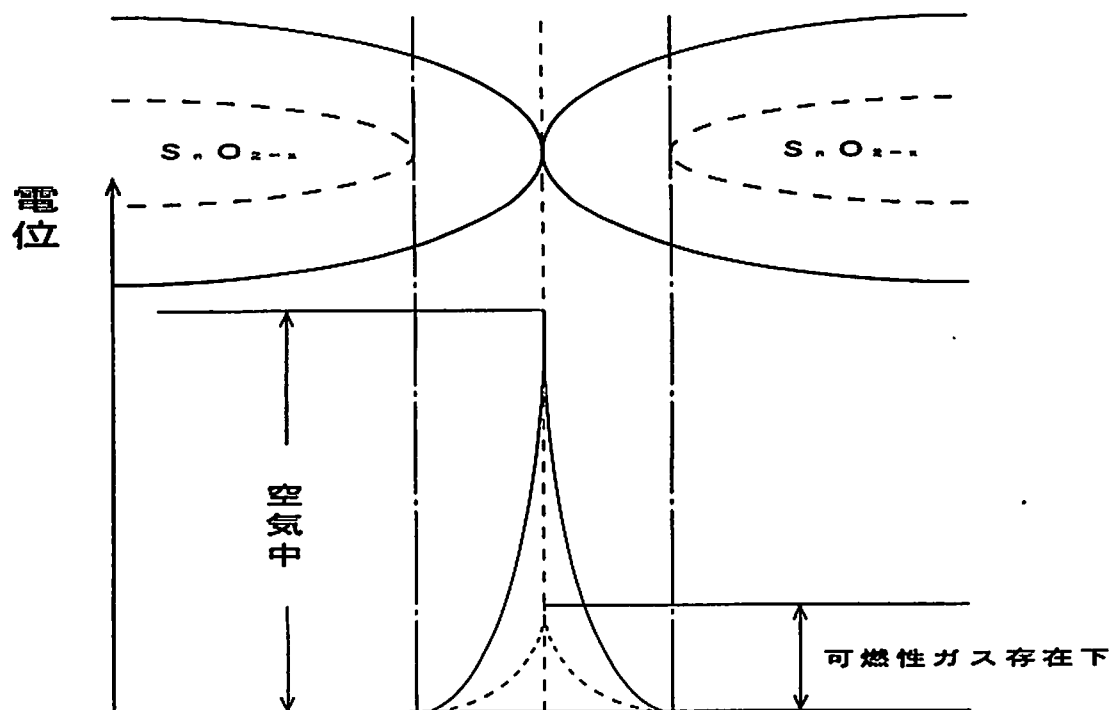


図4. 素子内部結晶粒界における電位障壁の様子

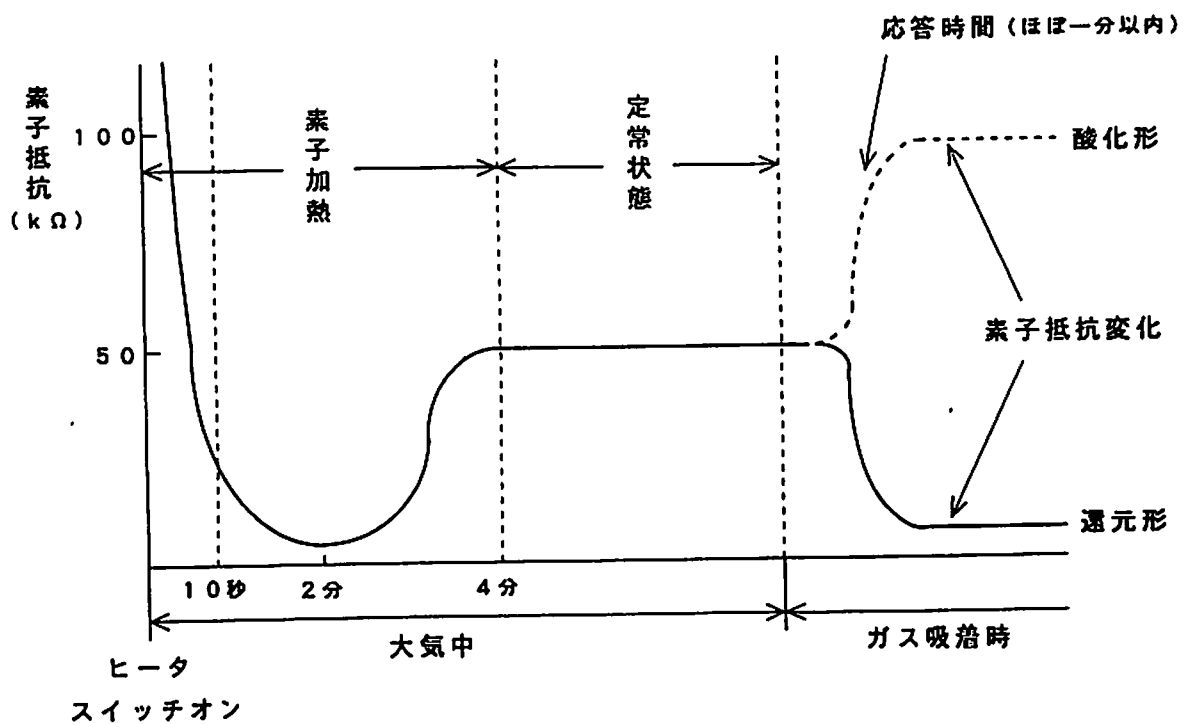


図5. n形半導体ガスセンサにガスが吸着したときの素子抵抗の変化

3.2 SnO_2 系半導体ガスセンサの利用方法

SnO_2 半導体ガスセンサの場合、図6に示すようにいずれの分子種に対しても検出感度が最大となる温度(T_m)が存在する。つまり、センサの動作温度を適当に組み合わせれば、ガス選択性の賦与がある程度可能であることを示している。また図6の特性はセンサ素子を製造する際に加える添加剤によって大きく異なることが判明している¹⁰⁾。

本研究では、ガスセンサの素子表面の温度を変化させることにより従来測定方法よりも、多くの種類の匂い分子の検出が可能であると仮定して、半導体ガスセンサ内部のヒータを任意の時間間隔でON/OFF制御して匂いの測定を行うことにした。

一方、半導体ガスセンサはガス選択能力が低いためにケモメトリクス(多数のセンサからの出力をもとに化学物質を識別する技術)による匂い識別に適しているので、複数のガス検出特性が異なる半導体ガスセンサを使用することにした。

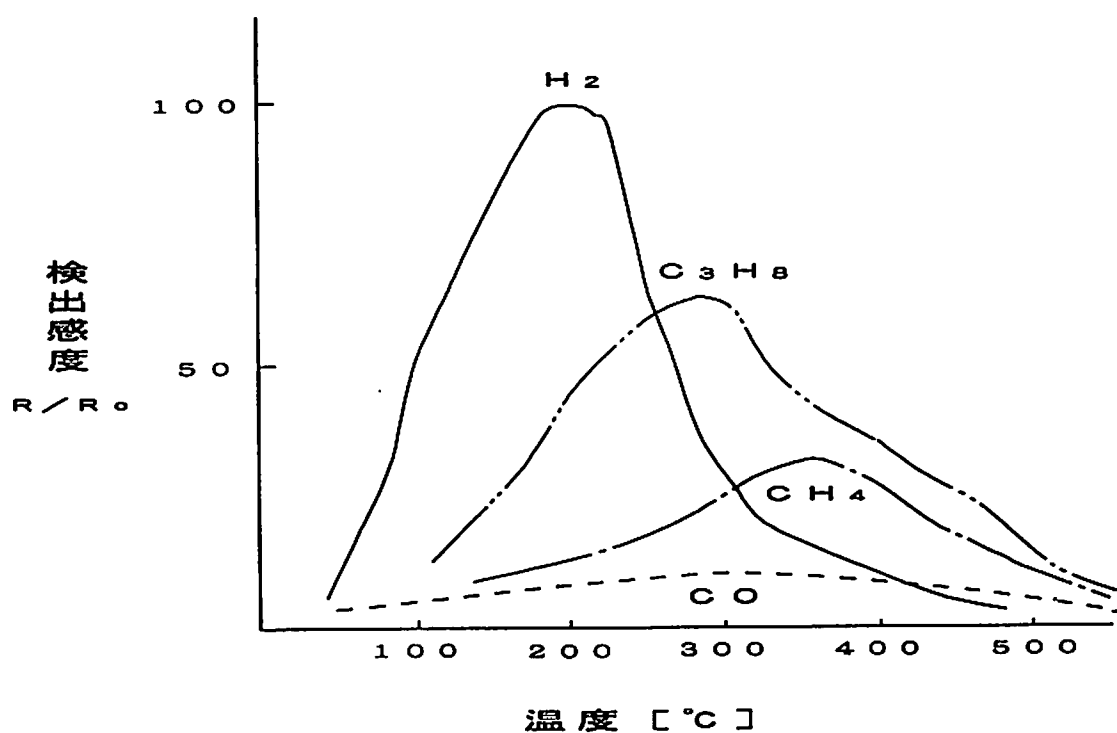


図6. 半導体ガスセンサの動作温度とガス選択性の関係

第 4 章 匂いセンシングシステム^{3, 6)}

本研究では匂い検出素子として特性がそれぞれ異なる 4 種類の半導体ガスセンサを使用した。使用した半導体ガスセンサの名称と特徴を表 2 に示す。また、図 7 に開発した匂いセンシングシステムの外観を示す。

表 2. 採用するセンサ名とその特徴

センサ名	特徴
アルコールガスセンサ	アルコールなどの有機溶剤ガスに対する感度が優れる
アンモニアガスセンサ	アンモニアガスを 3 0 p p m から検出可能
硫化水素ガスセンサ	硫化水素ガスを 5 p p m から検出可能
メタンガスセンサ	メタンガス・イソブタンガスに対する感度が優れる



図 7. 匂いセンシングシステムの外観

4 . 1 匂い測定装置の製作

半導体ガスセンサを使用する際には以下に示す点について考慮する必要がある。

- ・半導体ガスセンサはガス濃度に対して非常に高感度である。
- ・半導体ガスセンサは雰囲気中の気温・湿度の影響を受ける。

そこで、匂い測定装置は以下に示す点を考慮して製作した。

- ・ 被測定ガスを発生させるまでは、半導体ガスセンサを $30 \pm 2^{\circ}\text{C}$ の気温の空気中に入れておく。
- ・ マイクロシリンジを用いて一定の量の被測定サンプルをヒータによって気化させるようにする。
- ・ 気化したばかりの高温の状態の被測定ガスを半導体ガスセンサにかけないようにする。
- ・ 被測定ガスは実験装置の中で一様に分布するように循環用のファンを用いて十分に拡散されるようにする。

製作した匂い測定装置の見取り図を図 8 に示す。

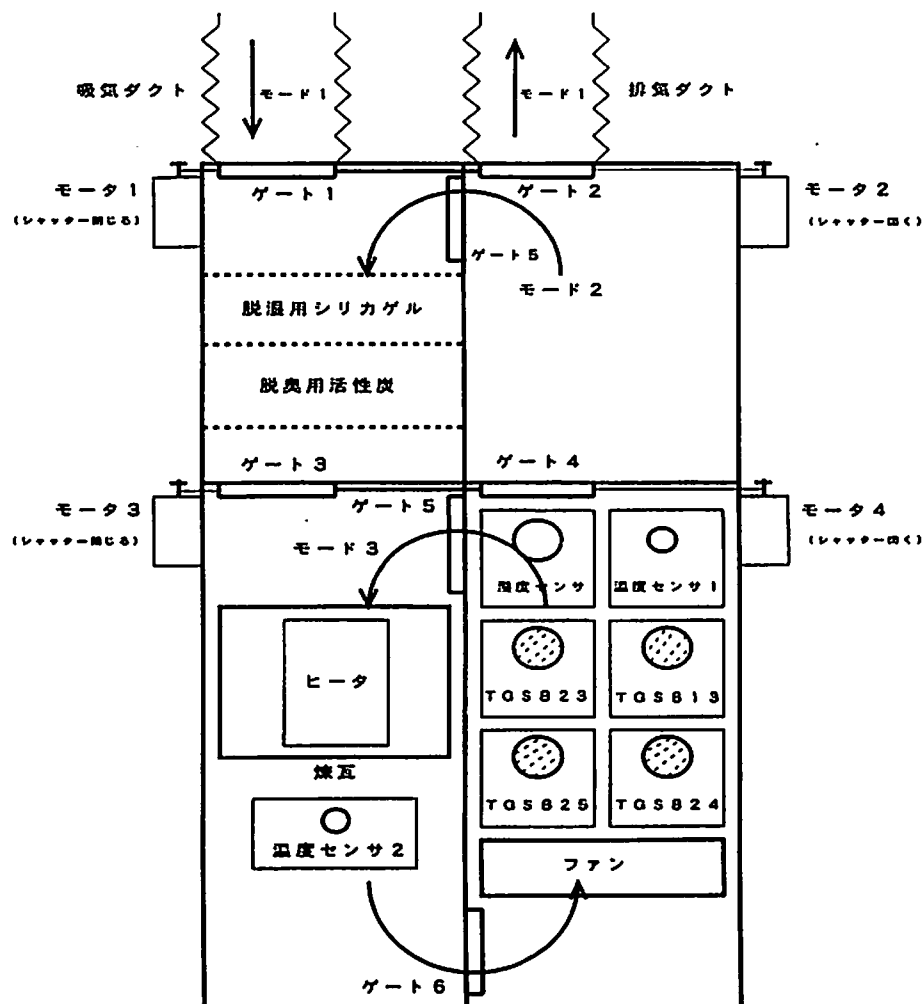


図 8. 匂い測定装置の見取り図

匂い測定装置は主に測定チャンバ、サンプル気化チャンバ、標準空気生成チャンバの3つのチャンバから構成され、換気兼拡散用ファンと4つのモータにより制御されるシャッターにより測定装置内の空気の流れは以下の3つのモードで循環する仕組みになっている。

モード1：測定装置内の換気（残留ガスを排出）

モード2：標準空気の生成（気温・湿度を調整）

モード3：被測定ガスの循環・拡散

4.2 匂いセンシングシステムの各種機能

開発した匂いセンシングシステムは、匂いの測定・記録・学習・識別さらに種々の分析を全てパーソナルコンピュータを利用して行う。匂いセンシングシステムの機能構成を図9に示す。

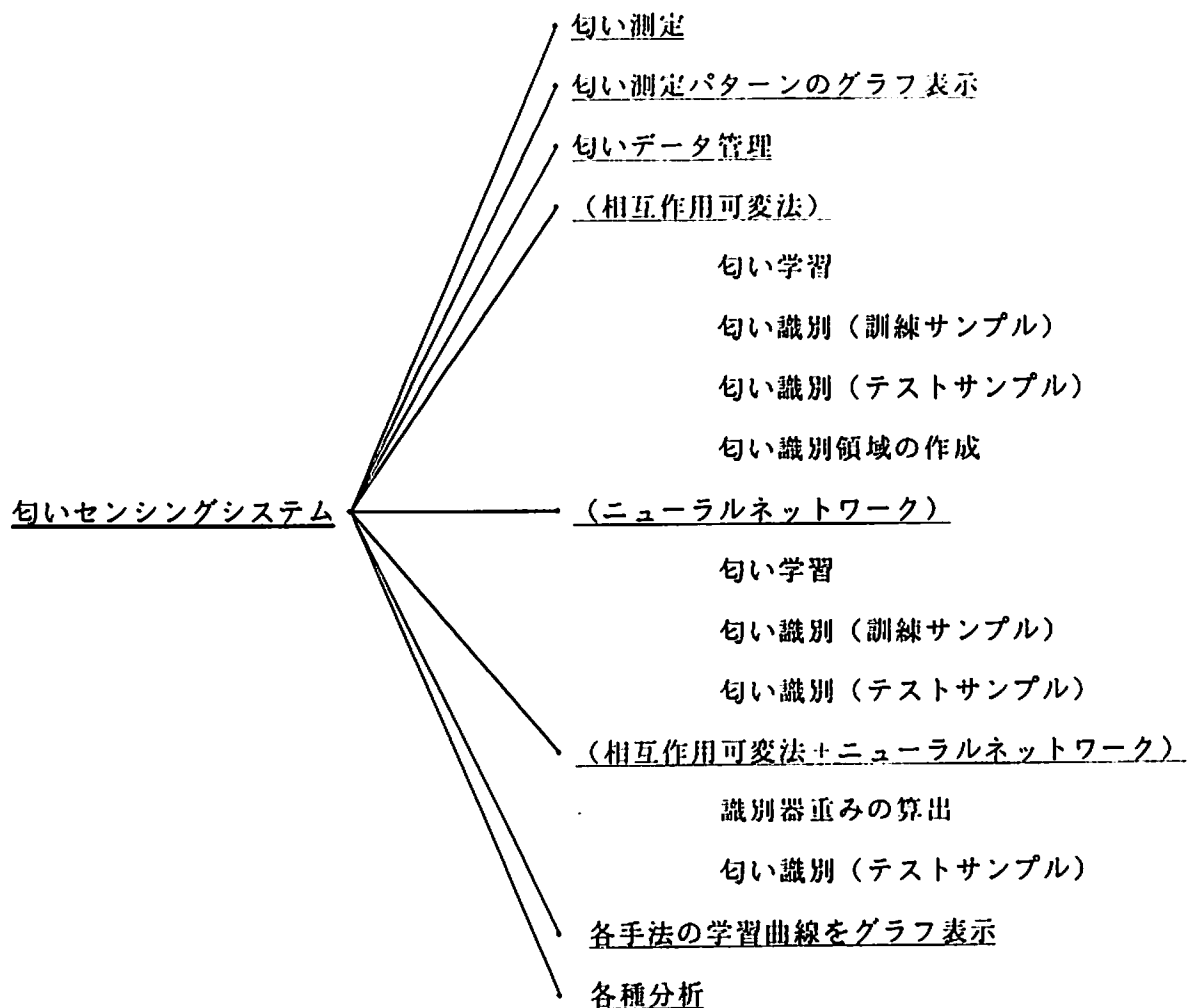


図9. 匂いセンシングシステムの機能構成

第 5 章 匂い測定手順

前章に示した匂い測定装置を用いて、表 3 に示す 1 2 種類のサンプルについて各ガスセンサからの匂い測定パターンを測定した（測定は全て自動制御方式）。

表 3．本研究で選定した 1 2 種類の匂いサンプル

サンプル名	省略名	分類
普通酒	FU4	アルコール飲料
	FU28	
吟醸酒	GIN17	
	GIN24	
泡盛	AWA	
V S O P	VSOP	
エタノール	ETNL	原臭
ハッカ油	MENT	
ポカリスエット	POCA	ジュース
マスカットジュース	MASC	
コーヒー	CAFE	コーヒー
ウーロン茶	U-CHA	お茶

匂い測定パターンは、図 1 0 に示す手順で 1 2 種類のサンプルについてそれぞれ 10 回測定した。測定の際には表 4 に示す測定手順で匂い測定パターンをパソコンに入力し記録した。半導体ガスセンサの抵抗変化は負荷抵抗の端子電圧の変化として現れるので、負荷抵抗の端子電圧を A / D コンバータにより取り込み、プログラム内でアナログ電圧値に変換した後にハードディスクに記録した。温度センサ・湿度センサからの電圧応答についても同様の処理を行った。

表 4．測定手順（ヒータ ON / OFF 間隔）

手順	ON / OFF 間隔（秒）	ON / OFF 回数
1	90	1
2	30	1
3	1	20
4	2	10
5	4	8
6	6	8
7	9	6
8	12	6
9	15	4
10	18	4
11	21	4
12	24	4
13	27	4
14	30	4

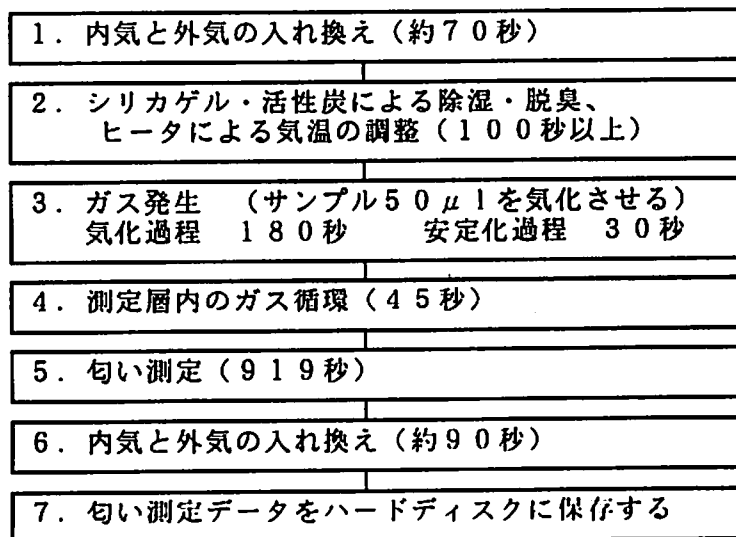


図10. 匂い実験の手順

第6章 匂い識別処理

本研究では匂い識別処理（匂いの学習及び識別）を行うために相互作用可変法を考案している。また、相互作用可変法による匂い識別処理の結果を客観的に比較するためにニューラルネットワークによる匂い識別処理も行っている。図11に2つの手法により匂い識別処理を行う際のデータ処理行程を示す。図11において「匂い識別領域の生成」と「ガスセンサごとの評価」に関しては相互作用可変法のみが処理する。

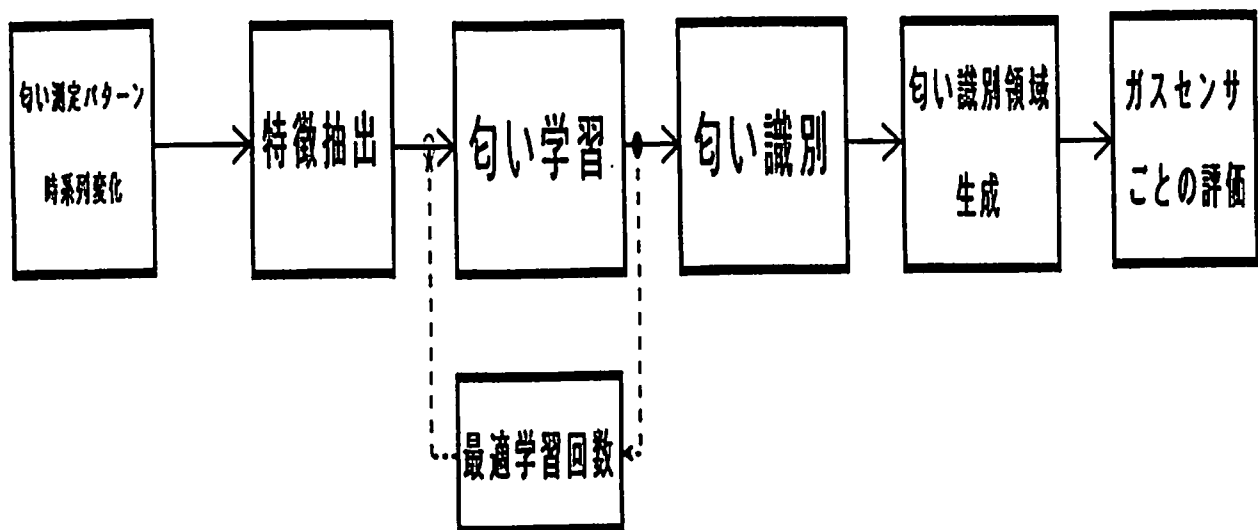


図11. 匂い識別処理におけるデータ処理行程

6.1 相互作用可変法

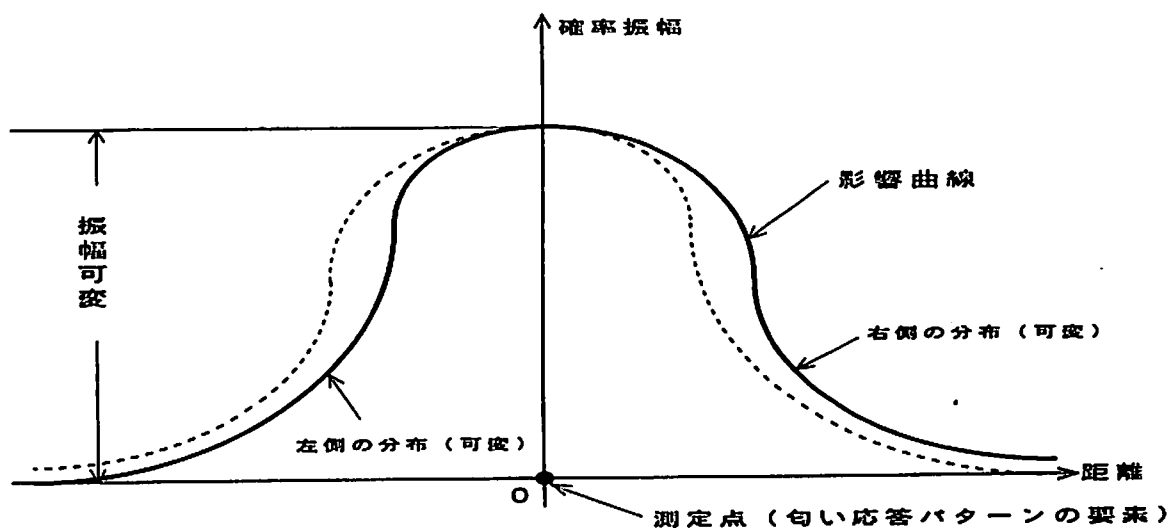
半導体ガスセンサから得られる匂い測定パターンは数V程度のバラツキを有する。類似臭の場合、匂い測定パターンが近接し、その結果、ある匂い測定パターンがそれと類似した匂い測定パターンと重なってしまう場合がある。ここで、特徴抽出した後に使用する匂い測定パターンを匂い応答パターンと呼ぶことにする。匂い応答パターンには次のような性質がある。

- 1) ガス濃度・湿度・気温等の変動が原因で予測不可能なバラツキが含まれる。
- 2) ガスの種類によって匂い応答パターンの分散が異なる傾向がある。

また匂い識別において識別器に求められる性質として

- 1) 区間性を有する。
- 2) 連続な確率分布が得られる。

これらの条件を満たすために相互作用可変法では、匂い応答パターンにおける各要素のバラツキが正規分布で表現できると仮定する。このため相互作用可変法はベイズ識別法⁶⁾に基づき、その学習方法はベイズ推定学習法における一手法と考えられる。ベイズ推定学習法では、カテゴリ条件確率（識別関数を求める際に必要になる項）におけるパラメータ θ （平均値と分散）を確率変数とする。 θ は学習パターン集合に条件付けられる。訓練サンプルに関する推定の不確かさが減少するようにこのパラメータ θ （初期値）を学習することで尤度の大きな θ' を求める。この θ' を求めるのが相互作用可変法である。図12に学習要素を示す。



1 個の測定点において左右に 2 個の正規分布を与える

図 12. 相互作用可変法における学習要素

相互作用可変法では次に示す特徴がある。図 1 3 に相互作用可変法の処理手順を示す。

- 1) 1つの学習要素に関して2つの正規分布型の関数を割り当てる。
- 2) 相関パラメータ（分散比が大きく匂い応答パターンの分散が小さければ小さいほど重みを大きくする）を導入する。

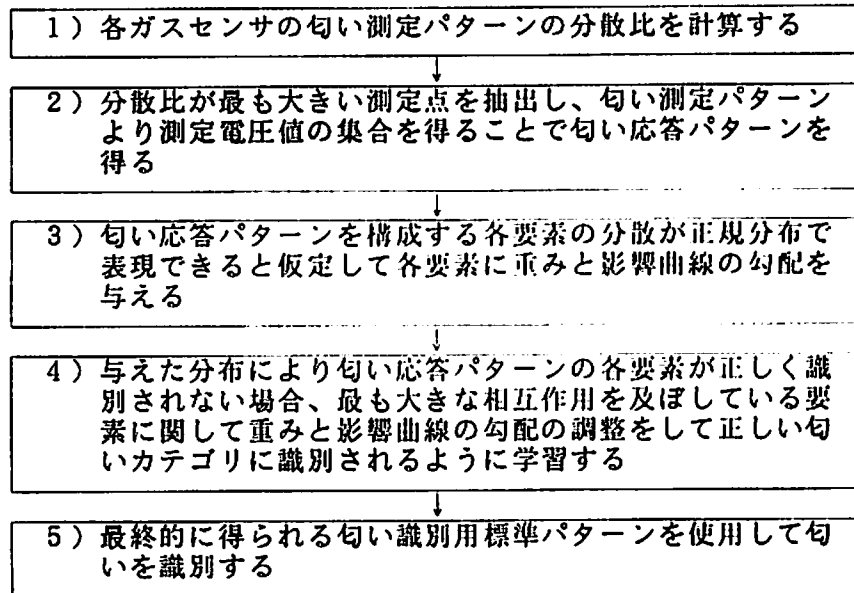


図 1 3. 相互作用可変法の処理手順

図 1 4 に図 1 3 での処理手順を図解した図を示す。

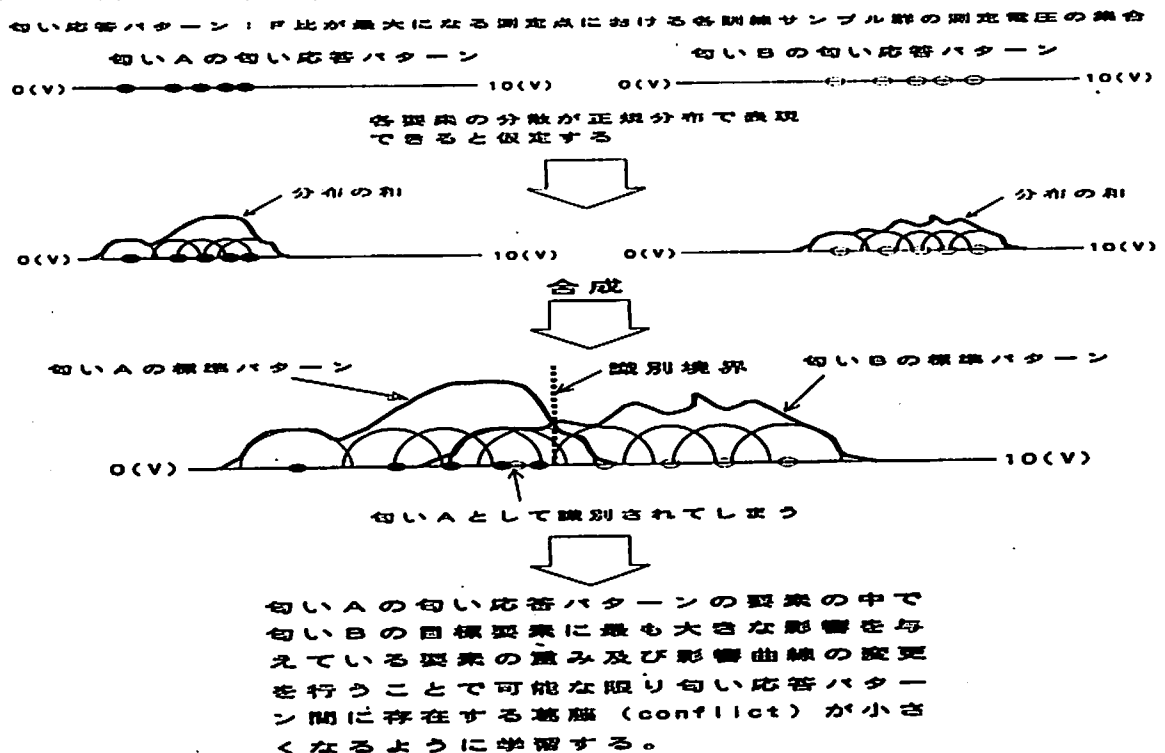


図 1 4. 相互作用可変法による標準パターン生成処理の手順

6.2 相互作用可変法における評価値の計算方法

評価対象電圧値を V_o として、カテゴリ番号 c 、訓練サンプル番号 s のサンプルの電圧値 $V_{s,c}$ と V_o 間の相互作用の大きさ $W_{o,s,c}$ を次式により計算する。

$$W_{o,s,c} = S_{c,F} (1 - S_{c,s,d}) D_{h,c} e^{(-D_r (V_o - V_{s,c}) (V_o - V_{s,c})^{-1})} \quad V_{s,c} < V_o \text{ の場合} \quad (1)$$

$$= S_{c,F} (1 - S_{c,s,d}) D_{h,c} e^{(-D_l (V_{s,c} - V_o) (V_{s,c} - V_o)^{-1})} \quad V_{s,c} > V_o \text{ の場合} \quad (2)$$

ここで

カテゴリ番号 : c ($1 \leq c \leq n$)

訓練サンプル番号 : s ($1 \leq s \leq N$)

測定電圧値 : $V_{s,c}$ ($0 \leq V_{s,c} \leq 10(V)$)

D_h : 測定電圧値に持たせる重み (初期値 : 1)

D_r : ボケ変化右側の傾き (初期値 : 2.0)

D_l : ボケ変化左側の傾き (初期値 : 2.0)

$S_{c,v}$: 訓練サンプルの平均値

$S_{c,s,d}$: 訓練サンプルの標準偏差

$S_{c,F}$: 訓練サンプルのF比 (分散比)

$S_{c,F} (1 - S_{c,s,d})$: 相関パラメータ

(注) F比とは、フィッシャー評価関数を指す。フィッシャー評価関数は、クラス内分散に対するクラス間分散の比として定義される⁷⁾。

カテゴリ番号 c における評価値 W_c は、

$$W_c = \sum_{i=1}^N W_{o,i,c} \quad (3)$$

で与えられる。また、

$$W_{c,max} = \max_{1 \leq i \leq n} W_i \quad (4)$$

を満たすカテゴリ番号 c_{max} が評価対象電圧値 V_o の属する匂いカテゴリの番号となる。

(注) \max は最大値を求める関数である。

6.3 相互作用可変法における学習方法

学習対象となる匂いデータがカテゴリ数 n 、訓練サンプル数 N であるとする。今、

$c_r: 1 \sim n$ の一様乱数 (整数値)

$s_r: 1 \sim N$ の一様乱数 (整数値)

として、カテゴリ番号 c_r 、サンプル番号 s_r におけるサンプルの評価対象電圧を V_r とし、式(1)(2)に基づいて全訓練サンプルに対する評価値を計算する。最も大きな評価値を与えたカテゴリ番号を c_{max} 、その評価値を $W_{c_{max}}$ とする。次に、訓練サンプルの中でカテゴリ c_r の識別に最も大きな影響を与えているデータ (カテゴリ番号を c_{max2} 、評価値を $W_{c_{max2}}$ 、電圧値を $V_{c_{max2}}$ とする) に関して、次に示すパラメータの重みおよび勾配の変更を行う。

$$\delta_r = W_{c_{max2}} / W_{c_{max}} \quad ; \delta_r \text{ は勾配変位係数} \quad (5)$$

$V_r < V_{c_{max2}}$ のとき

$$D_i = D_i + \delta_r \quad (D_i < l_{max}) : \text{傾き調節 (相互作用抑制項)} \quad (6)$$

$$D_h = D_h \times 0.96 \quad (D_i \geq l_{max}) : \text{重み調節 (抑制項)} \quad (7)$$

$V_r > V_{c_{max2}}$ のとき

$$D_i = D_i + \delta_r \quad (D_i < r_{max}) : \text{傾き調節 (相互作用抑制項)} \quad (8)$$

$$D_h = D_h \times 0.96 \quad (D_i \geq r_{max}) : \text{重み調節 (抑制項)} \quad (9)$$

訓練サンプルに対してある所定回数 (本研究では600回) だけ相互作用可変法を適用し、その際に正答が得られた全訓練サンプルの正答率の和を求めることで学習曲線を求める。最終的に得られた学習曲線の最大値を与える回数を学習回数とする。

6.4 相互作用可変法により匂い識別領域を求める方法

相互作用可変法により得られる各センサの標準パターンを利用して匂い識別領域を求める。まず、各センサの標準パターンの幅 (0~10(V)) を 1000 分割する。次にそれぞれの区間における匂い識別を行う。匂い識別領域は、以上に述べた 1000 個の区間での識別結果から構成される。図 15 に匂い識別領域の例を示す。

匂い識別領域：各センサの標準パターンの幅(0～10(V))を1000分割し、それぞれの区間において匂い識別を行った結果から構成される。

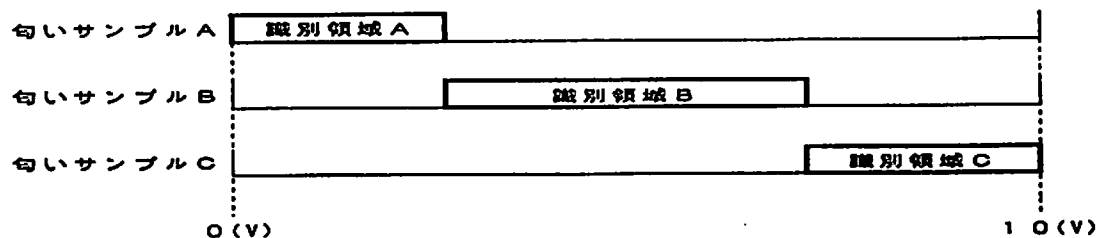


図 15．匂い識別領域の例

6.5 匂い識別領域から匂いごとに適したガスセンサを選択する方法

次に、匂いごとに適したガスセンサの選択手順を示す。

1. 各ガスセンサの各匂いサンプルに対する識別領域の広さを求める。
2. 最も識別領域の広いガスセンサの順に重みを4, 3, 2, 1の順で割り当てる。
 (注) 各種の匂いにおいて重みが最も大きいセンサに適したガスセンサとする。
 また、ガスセンサごとに、最も重みが大きくなる匂いを選出することで、
 各ガスセンサに適した匂いを選択する。
3. テストサンプルに対する識別結果から表5の基準により、ガスセンサの組み合わせに関する評価(識別可能性に関する評価)を行う。

表 5．識別余裕度に対する評価設定

識別余裕度	評価
0.331～1.000	確実
0.231～0.330	やや確実
0.121～0.230	可能
0.081～0.120	やや可能
0.066～0.080	やや難しい
0.041～0.065	難しい
0.000～0.040	困難

注) 識別余裕度とは、テストサンプルの識別を行った際に、正解したサンプルの正答率と誤答率との差である(しきい値は実験的に設定する)。

6.6 本研究で使用するニューラルネットワークについて

ニューラルネットワークは生体の神経回路網を工学的にモデル化したもので、優れたパターン認識の能力を持っている。図16に使用する多層ニューラルネットワークの概略図を示す。本研究では、表6に示す6種類の間層素子構造を持つニューラルネットワークを使用して匂いを識別する。しきい値 h の代わりに、入力層と中間層に -1 のユニットを入れる。結合重みを修正する際には教師信号が与えられる。本研究ではRumelhartが提唱した誤差逆伝搬学習（Back Propagation）⁹⁾が使用され、このネットワークの入力層に、特徴抽出部より得られた匂い応答パターンを入力し、出力層で得られる出力パターンに基づいて学習・識別が行われる。

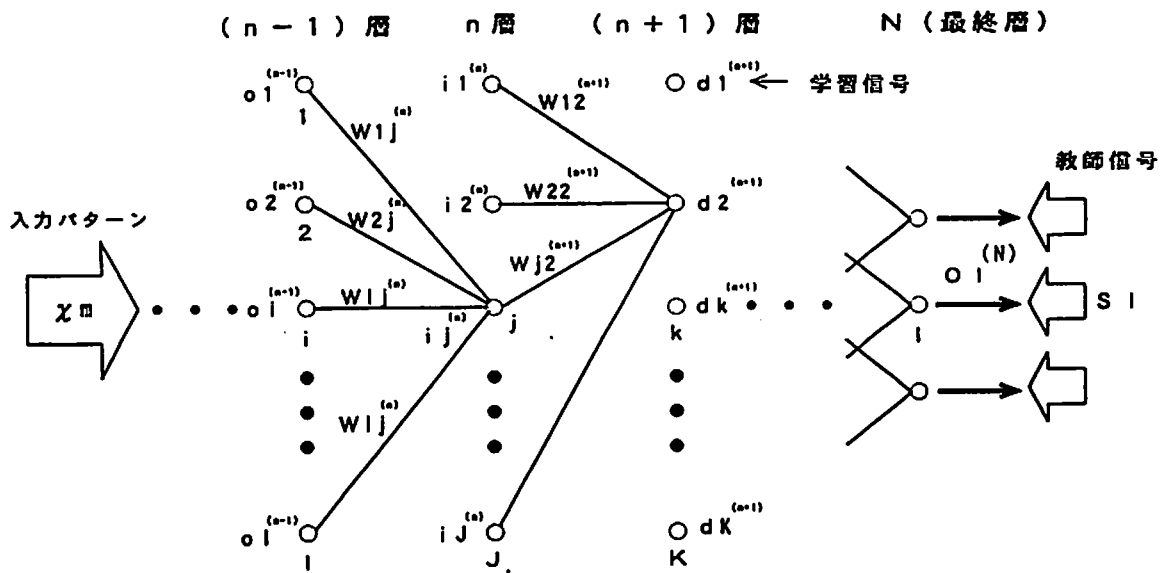


図16. 多層ニューラルネットワークの概略図

表6. 採用する中間素子構造

ネットワーク	入力層	中間層1	中間層2	中間層3	出力層
A	5	6	.	.	1 2
B	5	9	.	.	1 2
C	5	1 3	.	.	1 2
D	5	6	6	.	1 2
E	5	6	1 2	.	1 2
F	5	6	6	6	1 2

6.7 相互作用可変法とニューラルネットワークを組み合わせる方法

相互作用可変法とニューラルネットワークの訓練サンプルに対する識別結果から、各匂いサンプルを識別する際の識別器重みを求める。この識別器重みに基づいてテストサンプルを識別する。次に両手法を組み合わせる場合の識別手順を示す。

- 1) 訓練サンプルを相互作用可変法とニューラルネットワークにより識別する。
- 2) 各手法においてそれぞれの匂いサンプルに関して平均識別余裕度を求める。相互作用可変法により得られる平均識別余裕度を $V_{s,1} \sim V_{s,12}$ とし、ニューラルネットワークにより得られる平均識別余裕度を $N_{s,1} \sim N_{s,12}$ とする。
- 3) 各識別器の重みを次式により計算する。

$$\text{相互作用可変法の識別器重み } V_i = \frac{V_{s,i}}{V_{s,i} + N_{s,i}} \quad (i=1, 2, \dots, 12) \quad (10)$$

$$\text{ニューラルネット識別器重み } N_i = \frac{N_{s,i}}{V_{s,i} + N_{s,i}} \quad (i=1, 2, \dots, 12) \quad (11)$$

- 4) テストサンプルを相互作用可変法とニューラルネットワークにより識別する。相互作用可変法により得られる識別余裕度を $V_{r,i}$ 、ニューラルネットワークにより得られる識別余裕度を $N_{r,i}$ とする。
- 5) 両手法による合成識別余裕度 $C_{r,i}$ を次式により求める。

$$C_{r,i} = V_{r,i} \times V_i + N_{r,i} \times N_i \quad (i=1, 2, \dots, 12) \quad (12)$$

この方法では訓練サンプルの識別結果がテストサンプルの識別結果にもある程度反映されることを前提としている。

第 7 章 実験結果

7. 1 12 種類の訓練用匂いサンプルを測定した結果

図 17～図 20 に普通酒 4 を測定した際に各ガスセンサから得られた匂い測定パターンを示す。（注：グラフ内の番号は測定手順番号を示す）

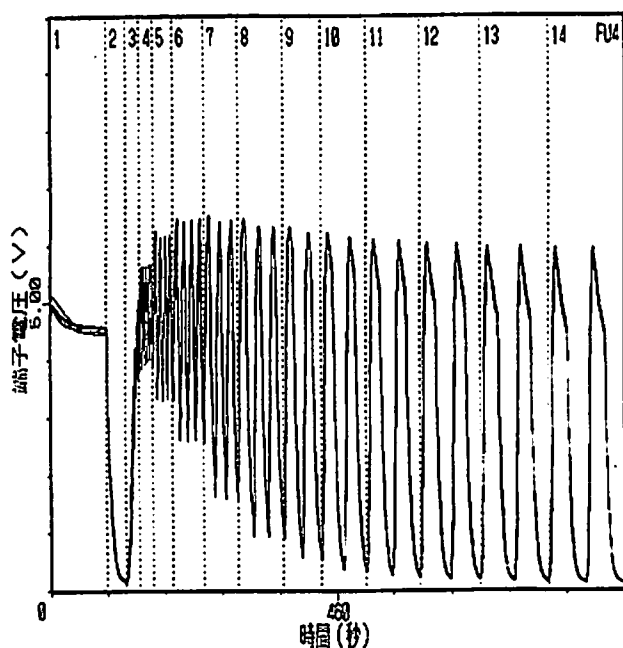


図 17. アンモニアガスセンサの応答

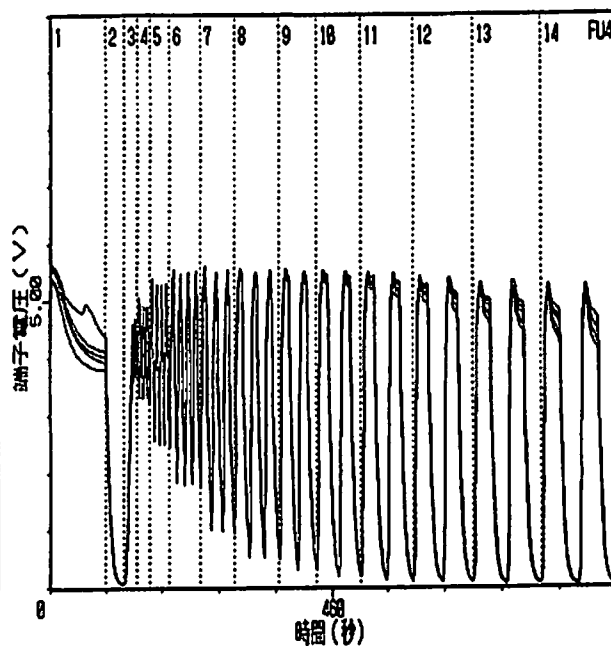


図 18. 硫化水素ガスセンサの応答

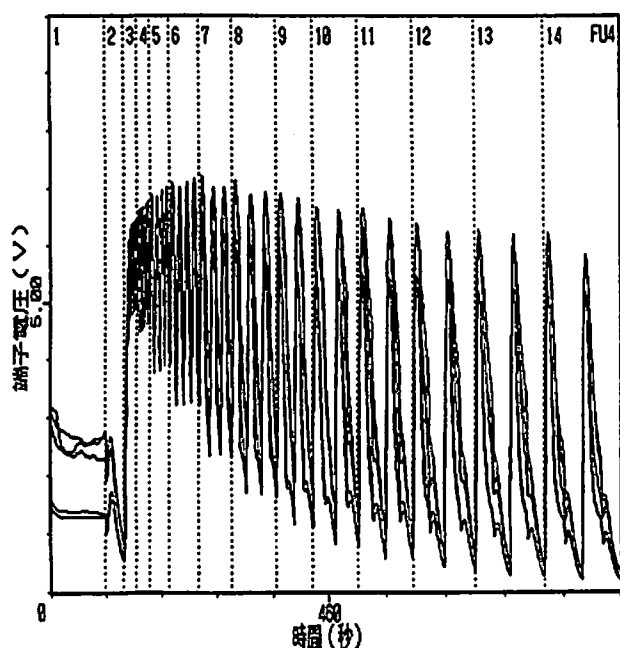


図 19. メタンガスセンサの応答

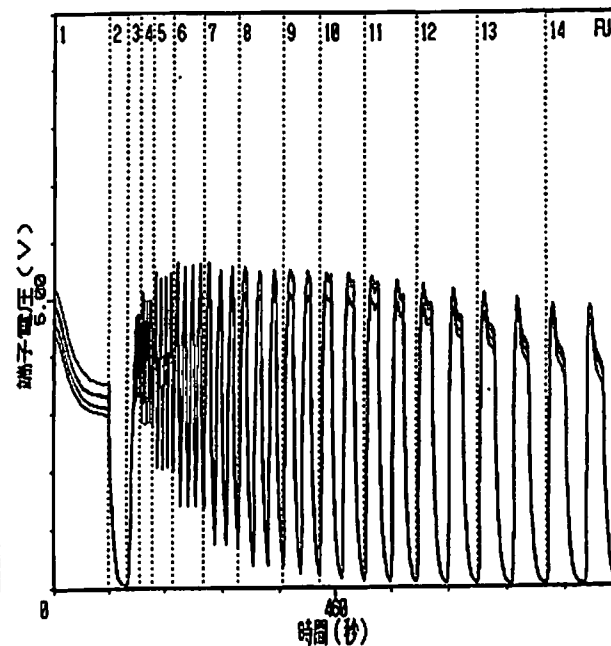


図 20. アルコールガスセンサの応答

図 2 1 ～図 2 4 に普通酒 2 8 を測定した際に各ガスセンサから得られた匂い測定パターンを示す。

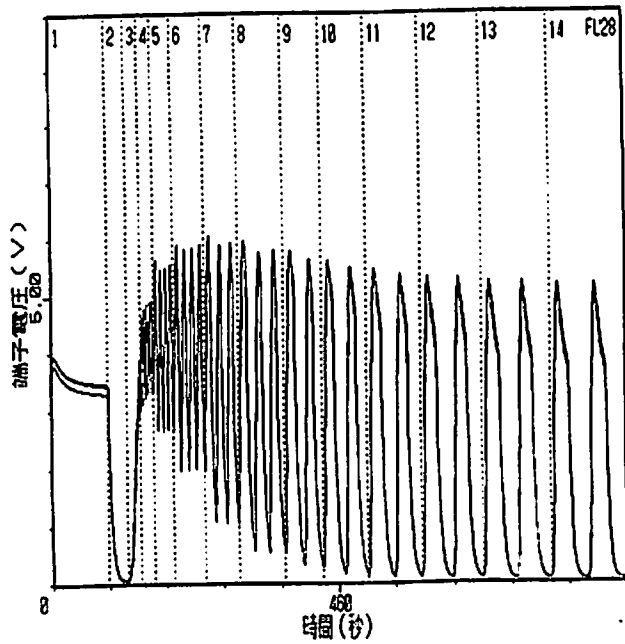


図 2 1 . アンモニアガスセンサの応答

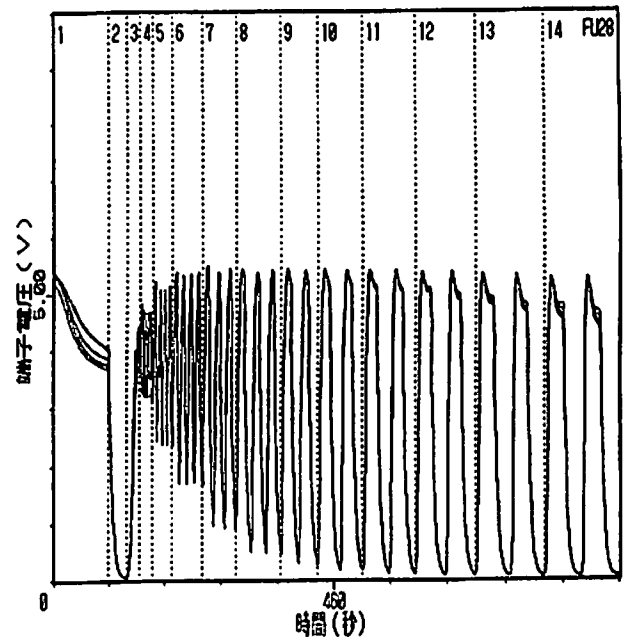


図 2 2 . 硫化水素ガスセンサの応答

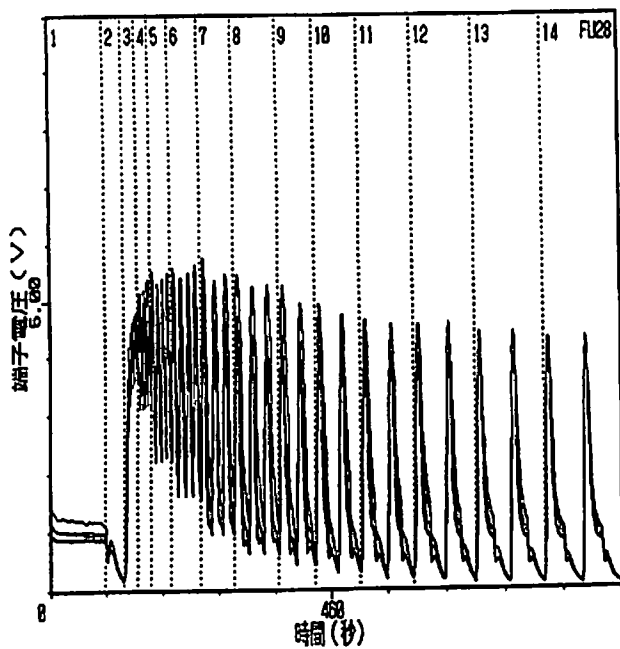


図 2 3 . メタンガスセンサの応答

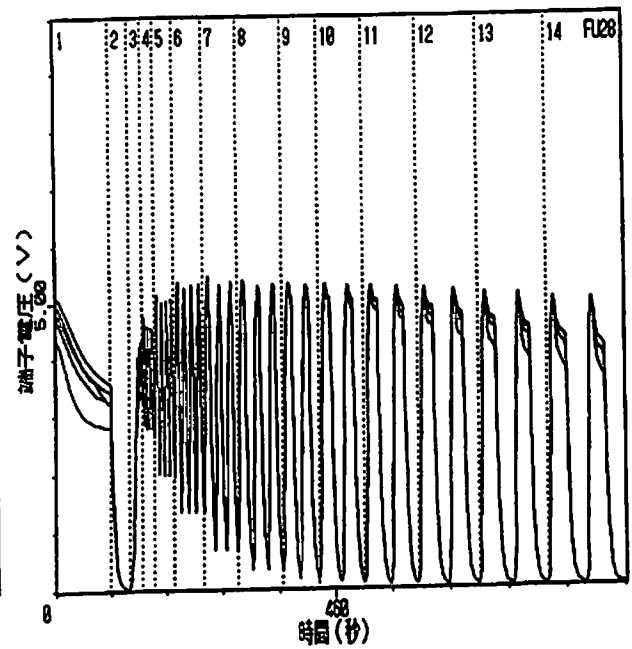


図 2 4 . アルコールガスセンサの応答

図 2 5 ～ 図 2 8 に吟醸酒 1 7 を測定した際に各ガスセンサから得られた匂い測定パターンを示す。

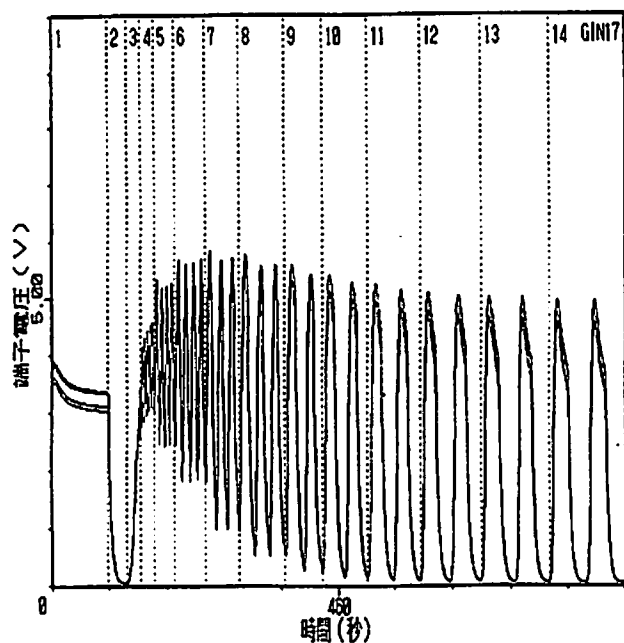


図 2 5. アンモニアガスセンサの応答

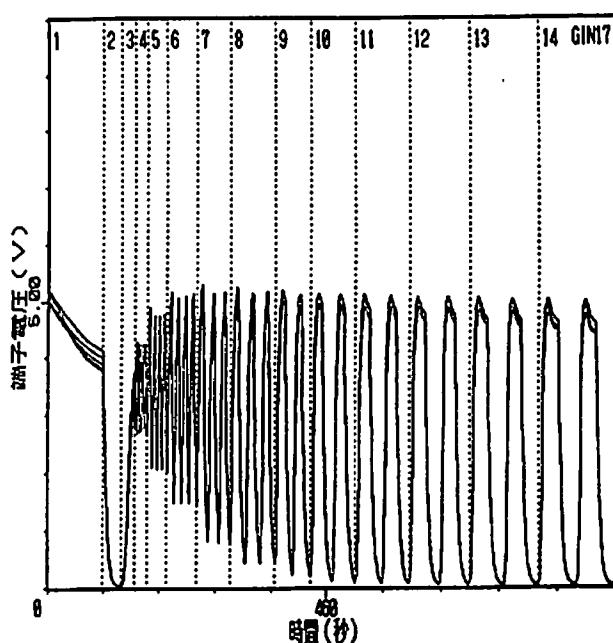


図 2 6. 硫化水素ガスセンサの応答

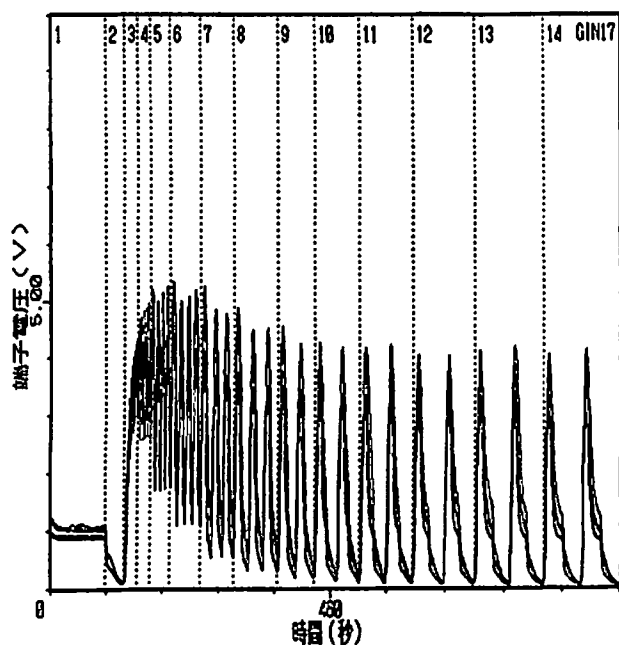


図 2 7. メタンガスセンサの応答

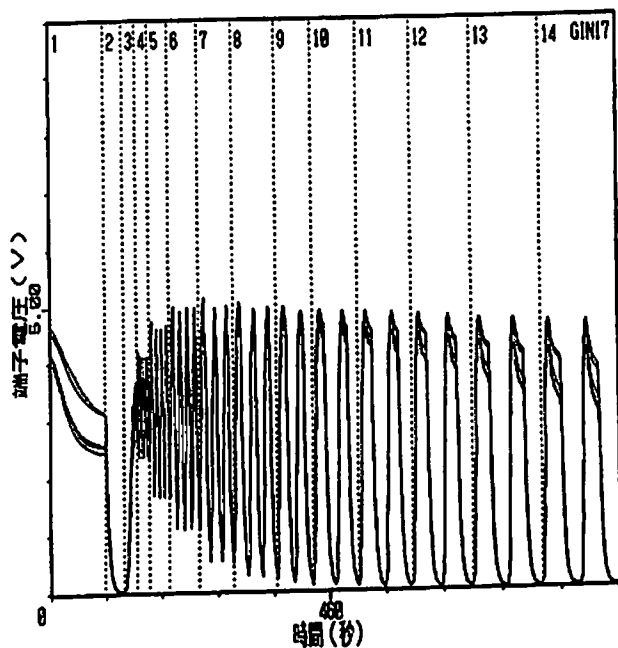


図 2 8. アルコールガスセンサの応答

図 2 9 ～ 図 3 2 に吟醸酒 2 4 を測定した際に各ガスセンサから得られた匂い測定パターンを示す。

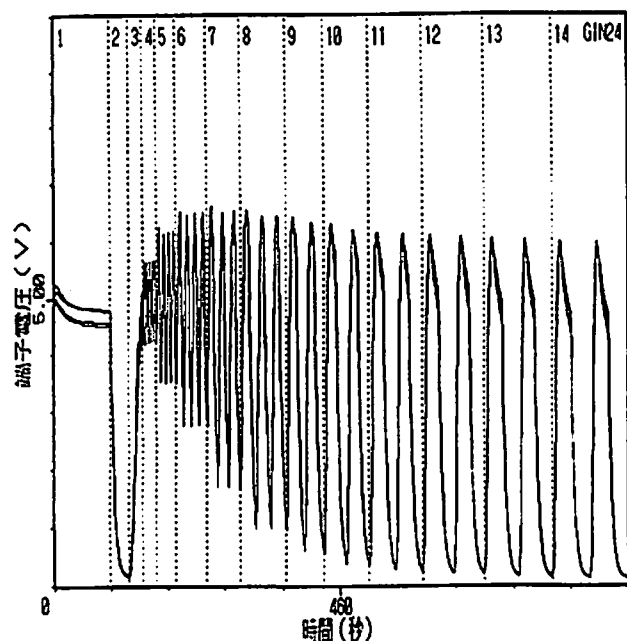


図 2 9 . アンモニアガスセンサの応答

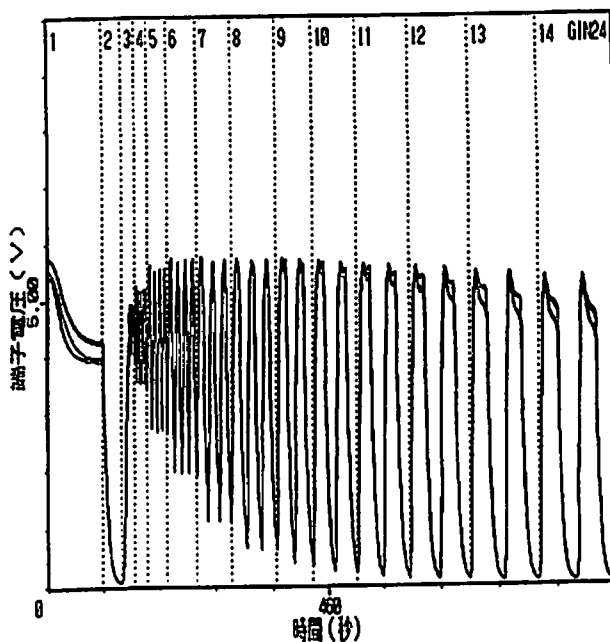


図 3 0 . 硫化水素ガスセンサの応答

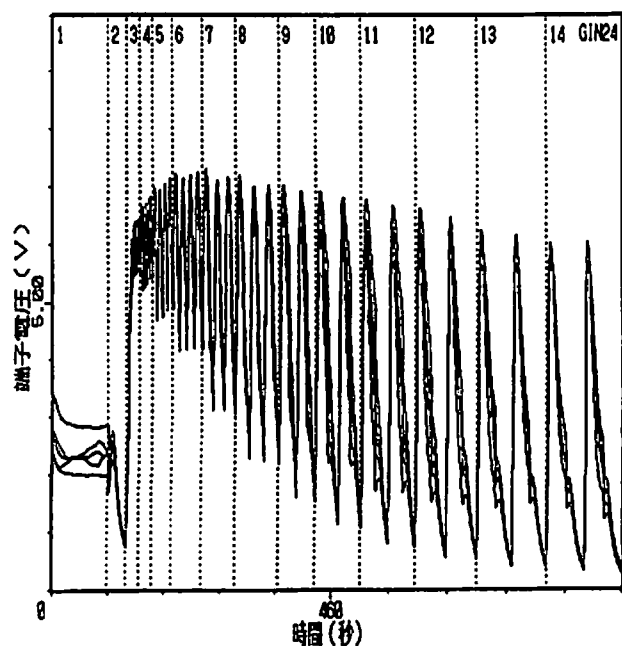


図 3 1 . メタンガスセンサの応答

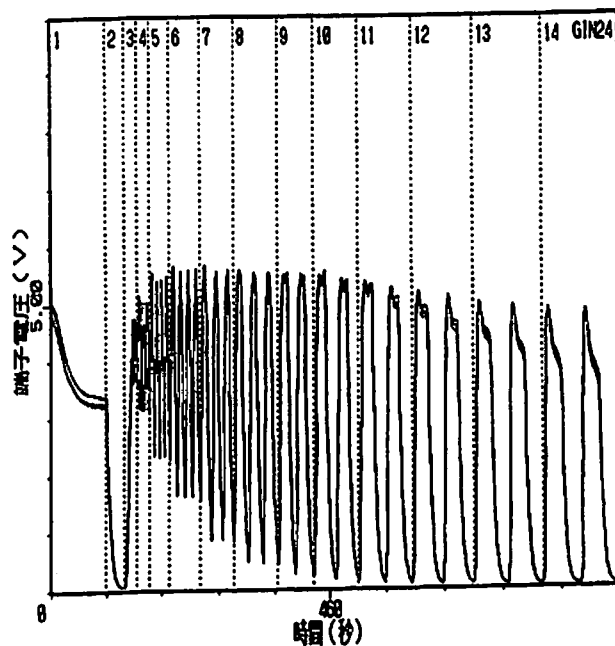


図 3 2 . アルコールガスセンサの応答

図 3 3 ～図 3 6 に泡盛を測定した際に各ガスセンサから得られた匂い測定パターンを示す。

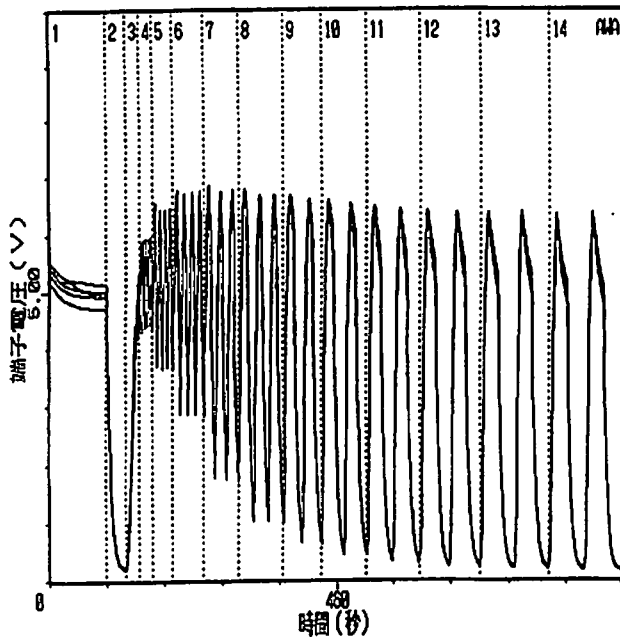


図 3 3 . アンモニアガスセンサの応答

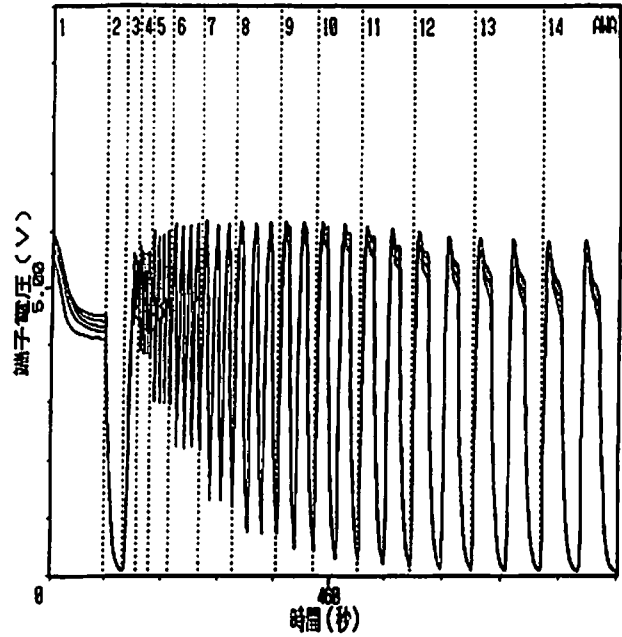


図 3 4 . 硫化水素ガスセンサの応答

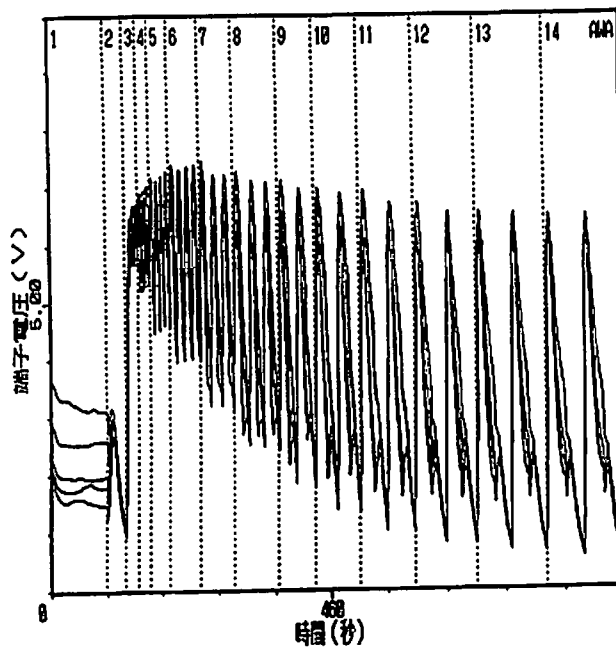


図 3 5 . メタンガスセンサの応答

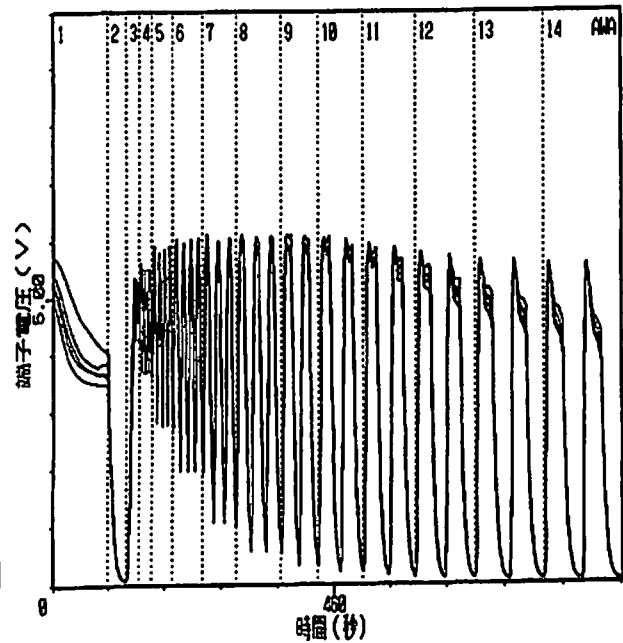


図 3 6 . アルコールガスセンサの応答

図37～図40にV S O Pを測定した際に各ガスセンサから得られた匂い測定パターンを示す。

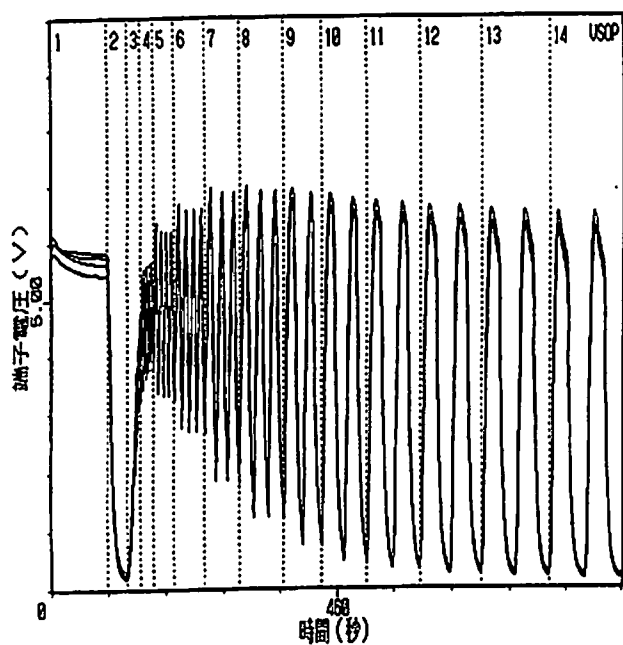


図37. アンモニアガスセンサの応答

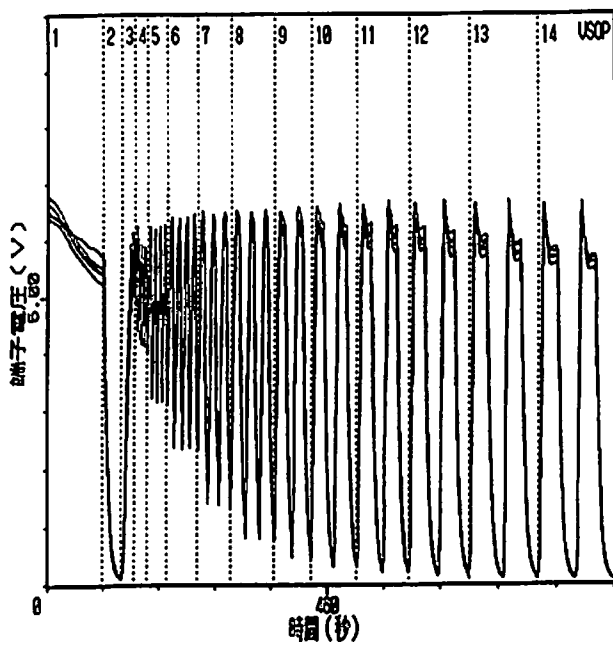


図38. 硫化水素ガスセンサの応答

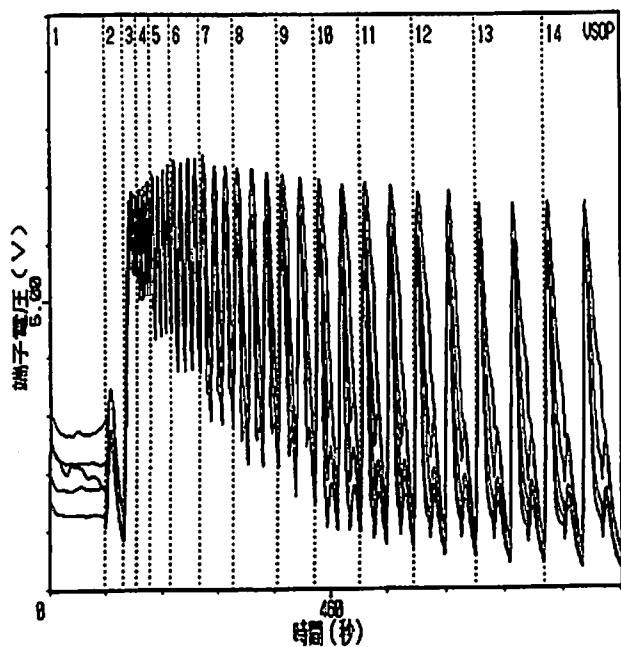


図39. メタンガスセンサの応答

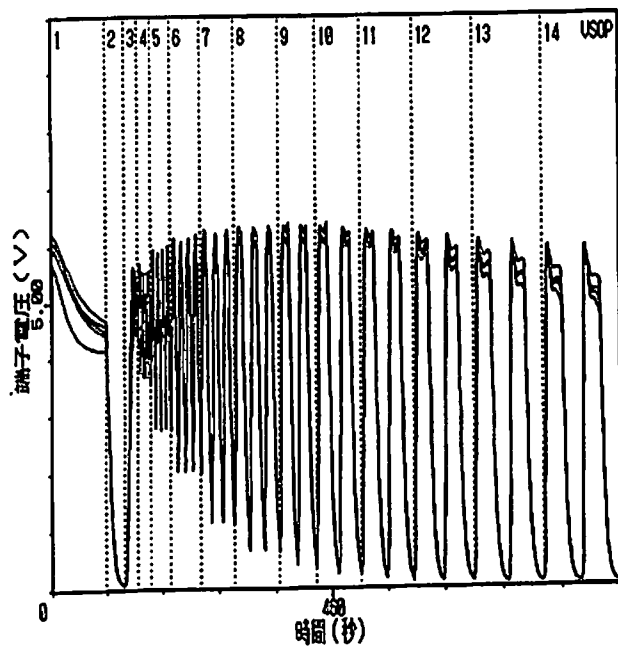


図40. アルコールガスセンサの応答

図 4 1 ～ 図 4 4 にエタノールを測定した際に各ガスセンサから得られた匂い測定パターンを示す。

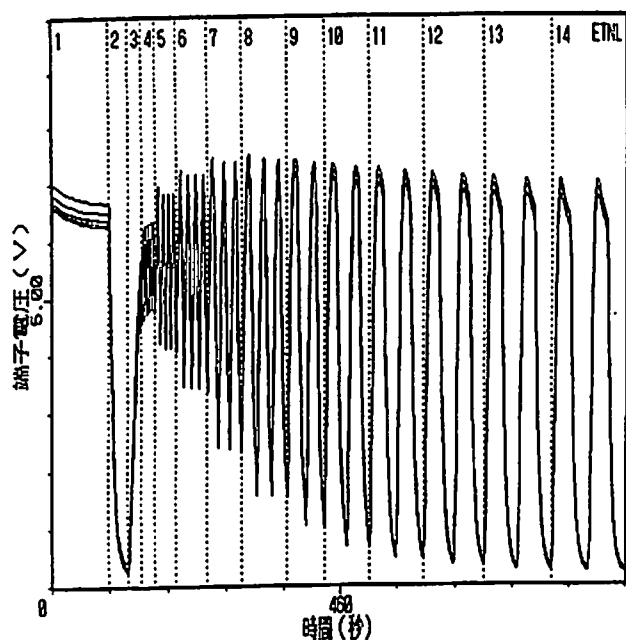


図 4 1 . アンモニアガスセンサの応答

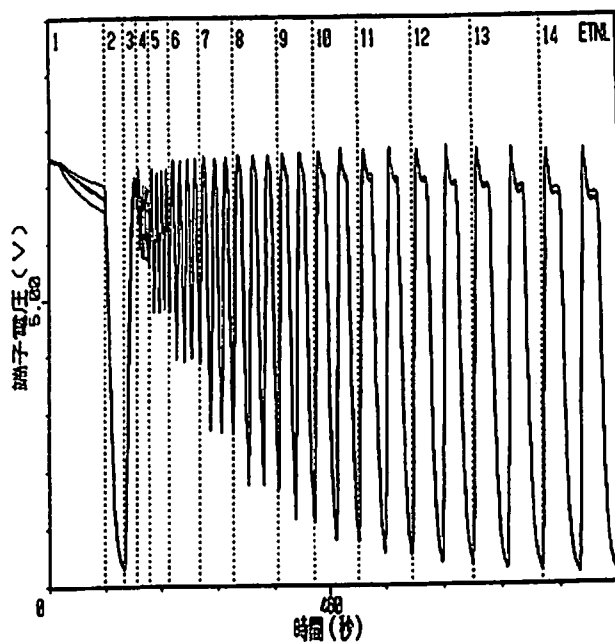


図 4 2 . 硫化水素ガスセンサの応答

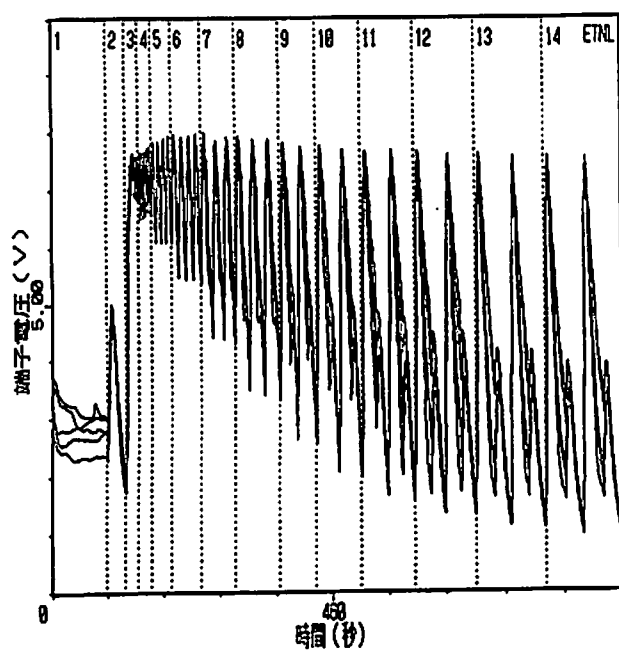


図 4 3 . メタンガスセンサの応答

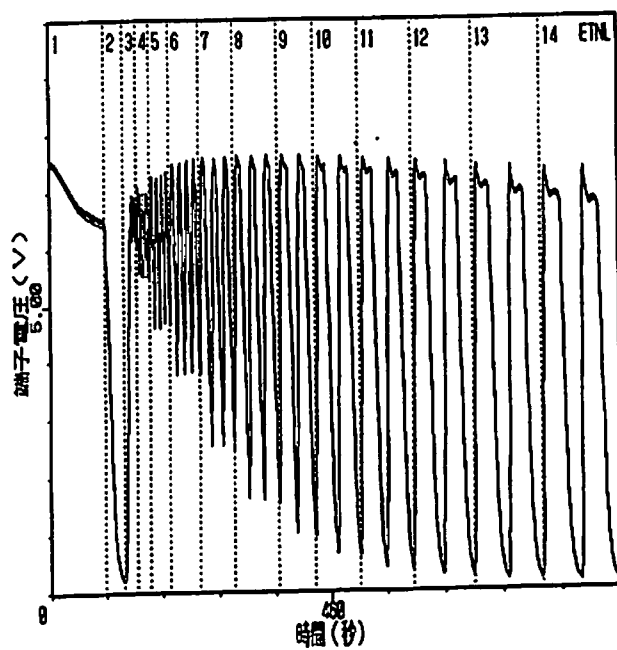


図 4 4 . アルコールガスセンサの応答

図45～図48にハッカ油を測定した際に各ガスセンサから得られた匂い測定パターンを示す。

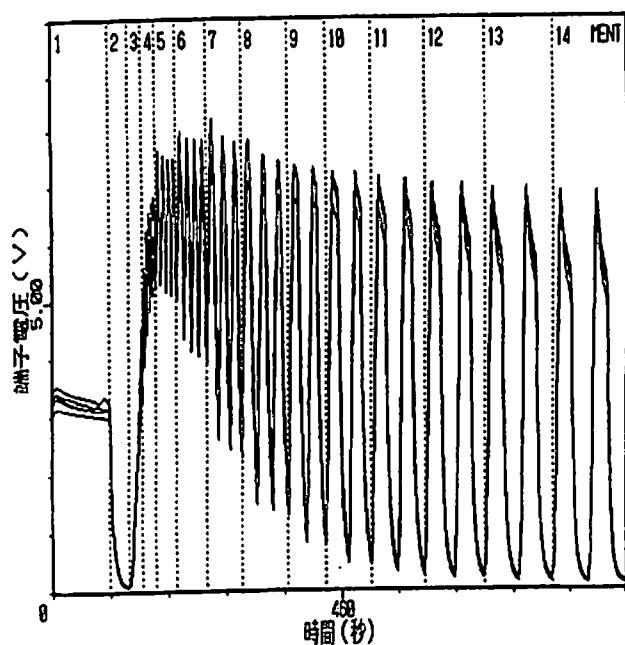


図45. アンモニアガスセンサの応答

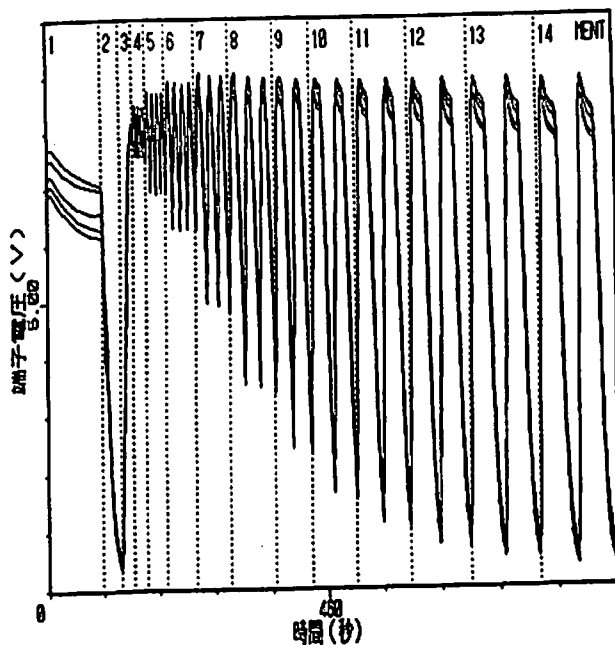


図46. 硫化水素ガスセンサの応答

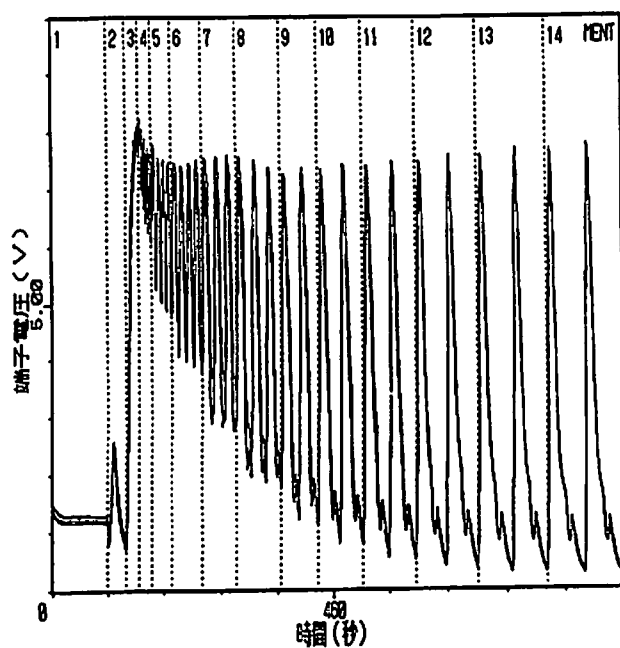


図47. メタンガスセンサの応答

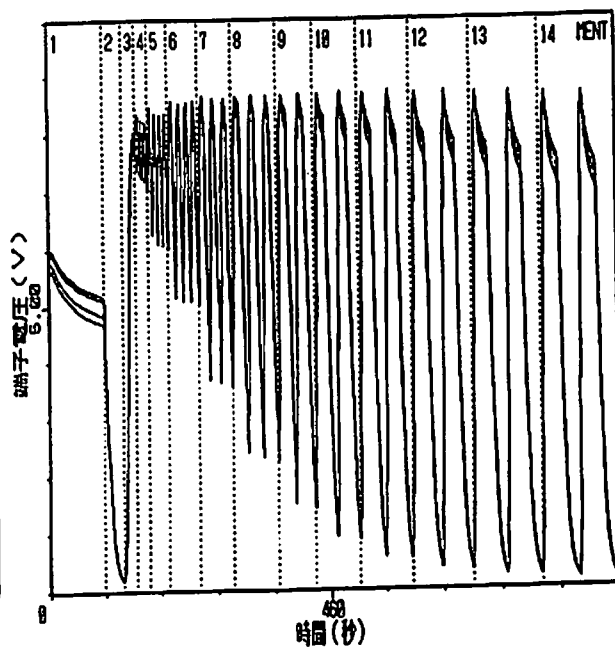


図48. アルコールガスセンサの応答

図49～図52にポカリスウェットを測定した際に各ガスセンサから得られた匂い測定パターンを示す。

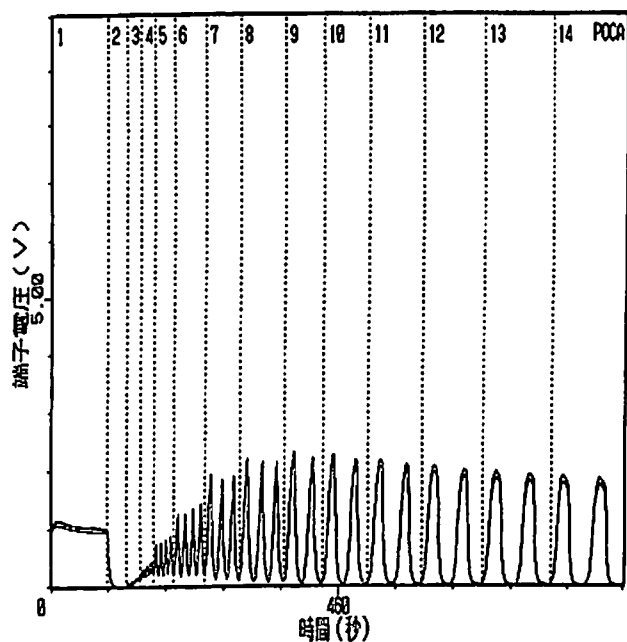


図49．アンモニアガスセンサの応答

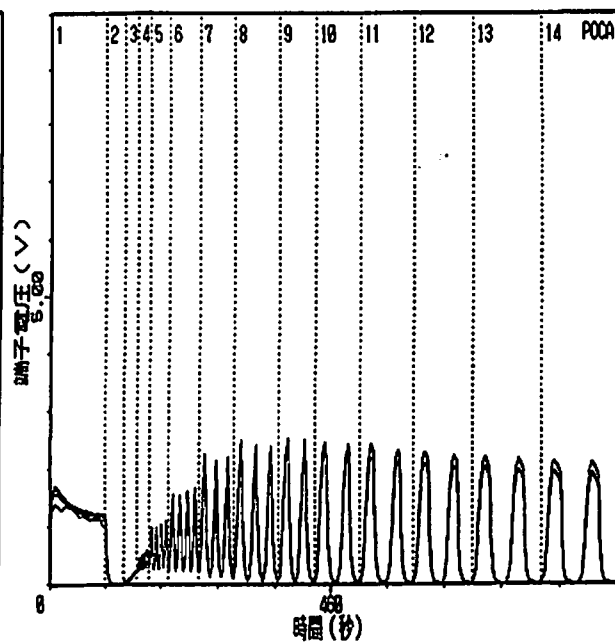


図50．硫化水素ガスセンサの応答

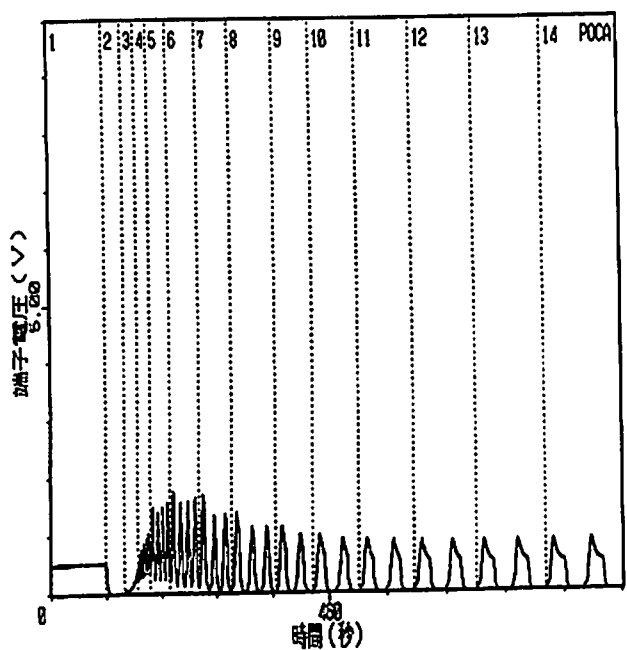


図51．メタンガスセンサの応答

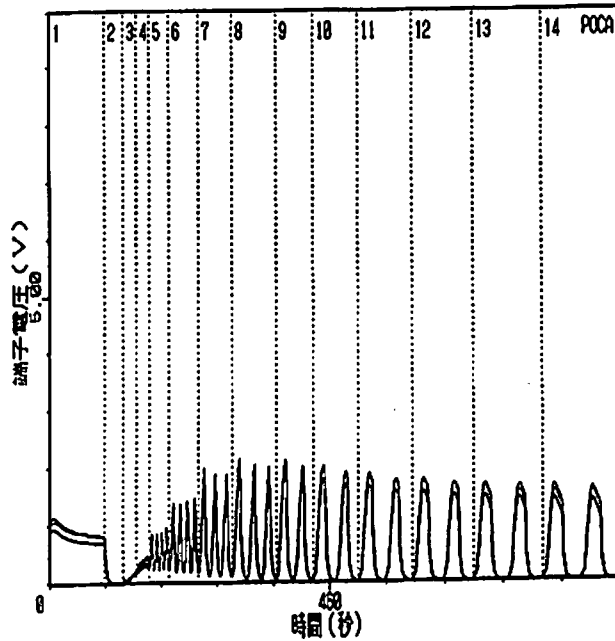


図52．アルコールガスセンサの応答

図53～図56にマスカットジュースを測定した際に各ガスセンサから得られた匂い測定パターンを示す。

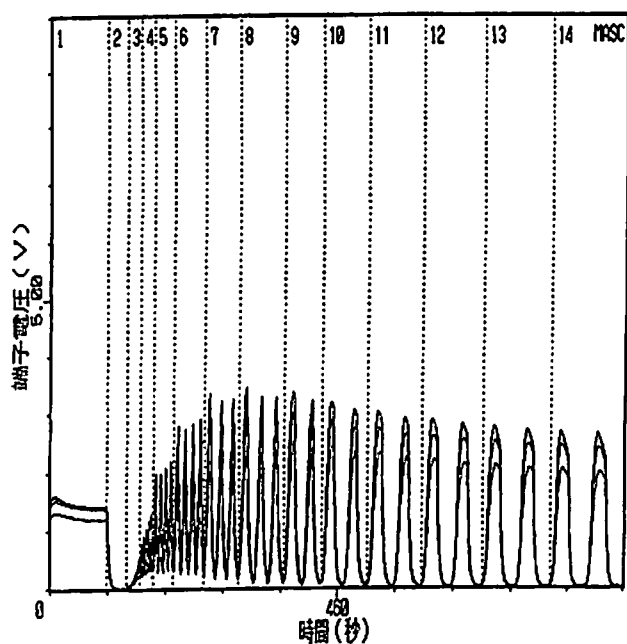


図53. アンモニアガスセンサの応答

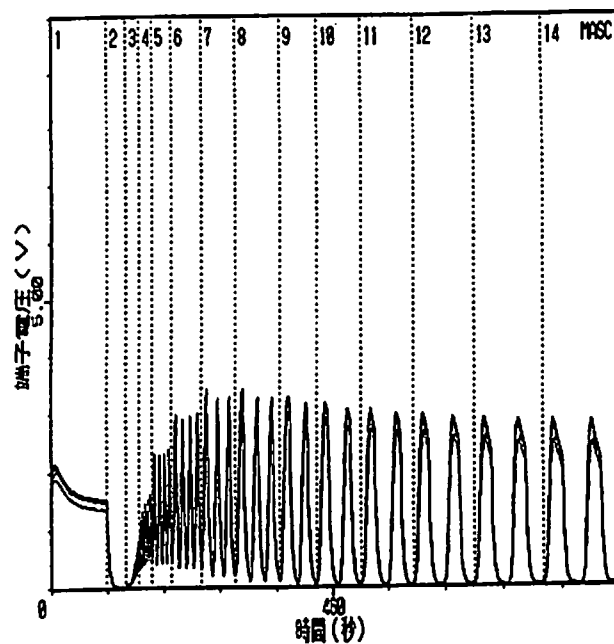


図54. 硫化水素ガスセンサの応答

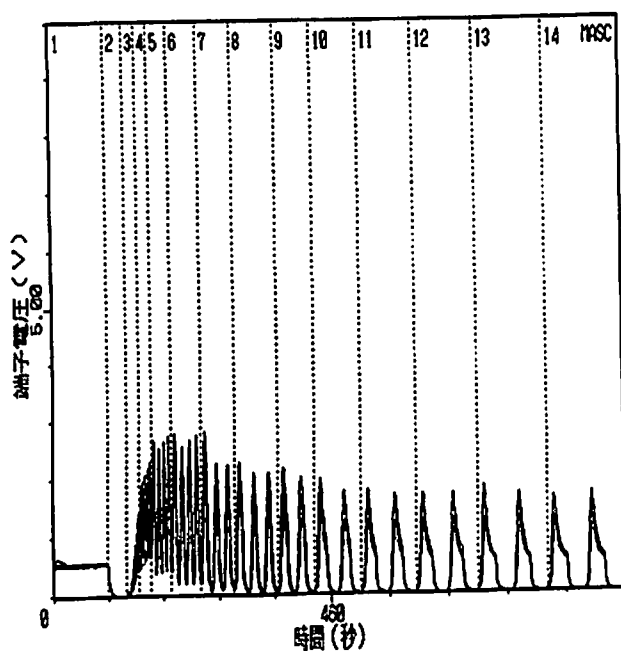


図55. メタンガスセンサの応答

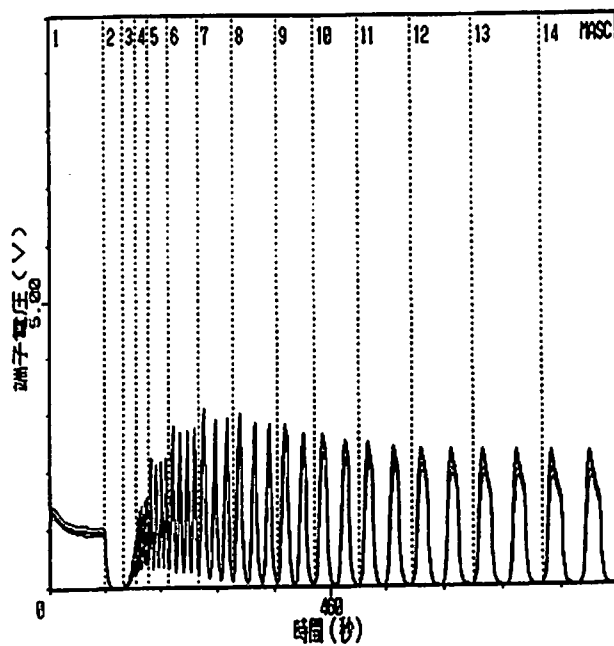


図56. アルコールガスセンサの応答

図57～図60にコーヒーを測定した際に各ガスセンサから得られた匂い測定パターンを示す。

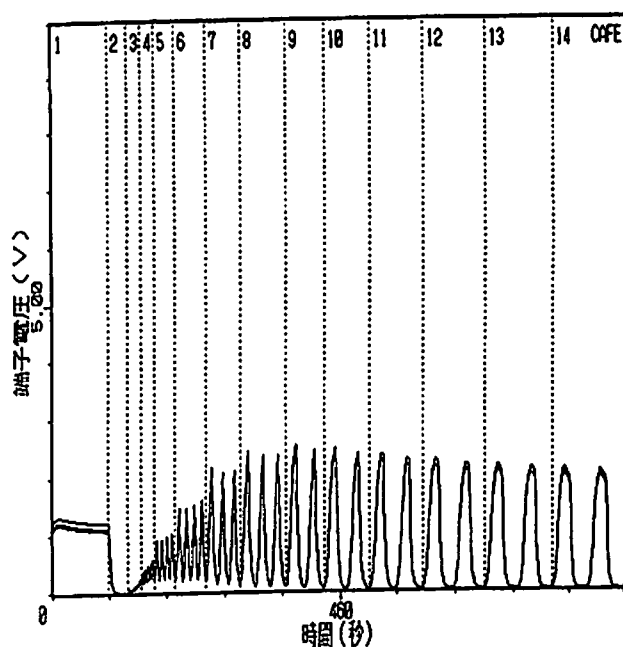


図57. アンモニアガスセンサの応答

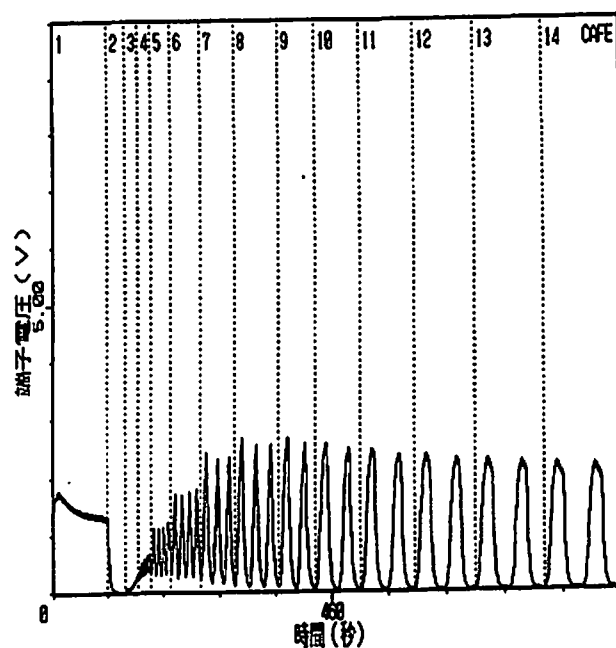


図58. 硫化水素ガスセンサの応答

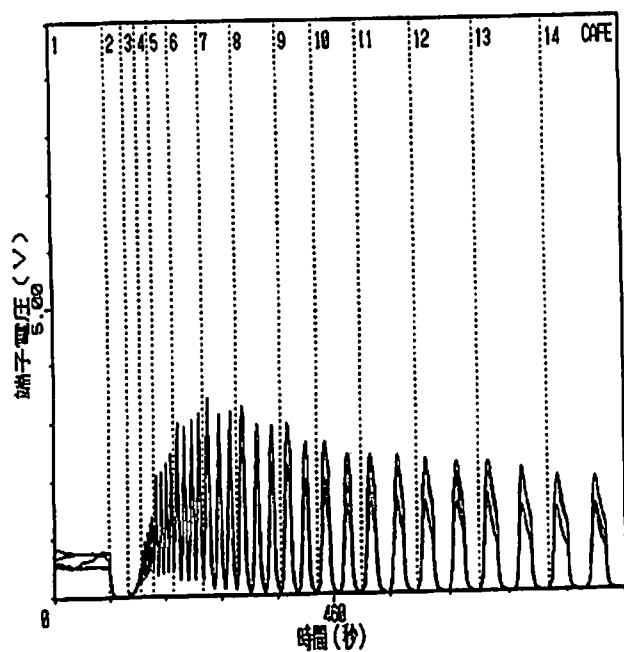


図59. メタンガスセンサの応答

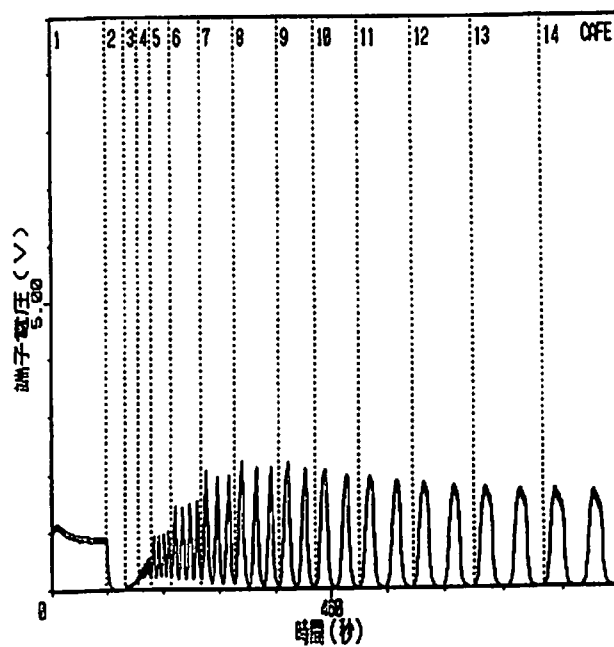


図60. アルコールガスセンサの応答

図 6 1 ～図 6 4 にウーロン茶を測定した際に各ガスセンサから得られた匂い測定パターンを示す。

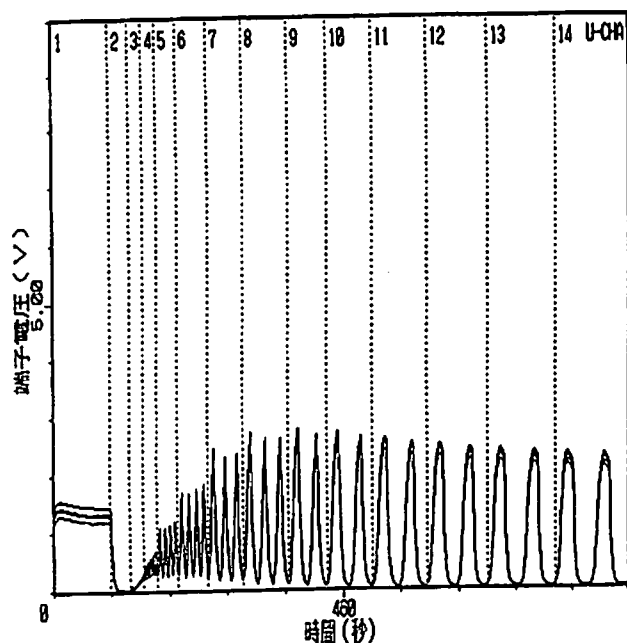


図 6 1. アンモニアガスセンサの応答

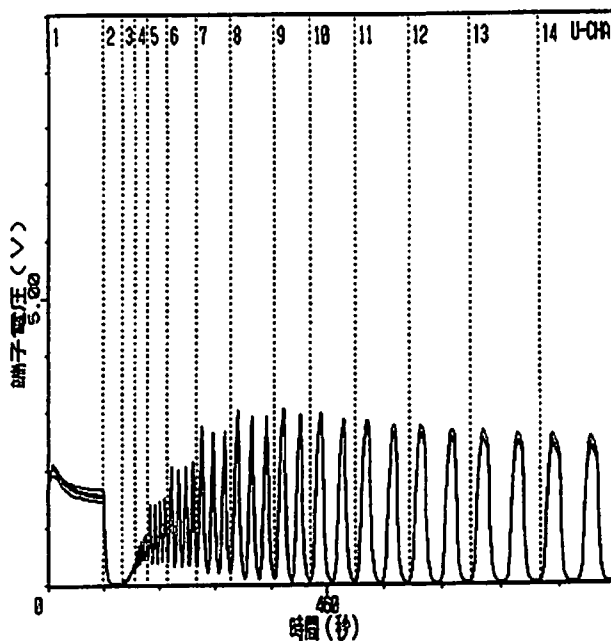


図 6 2. 硫化水素ガスセンサの応答

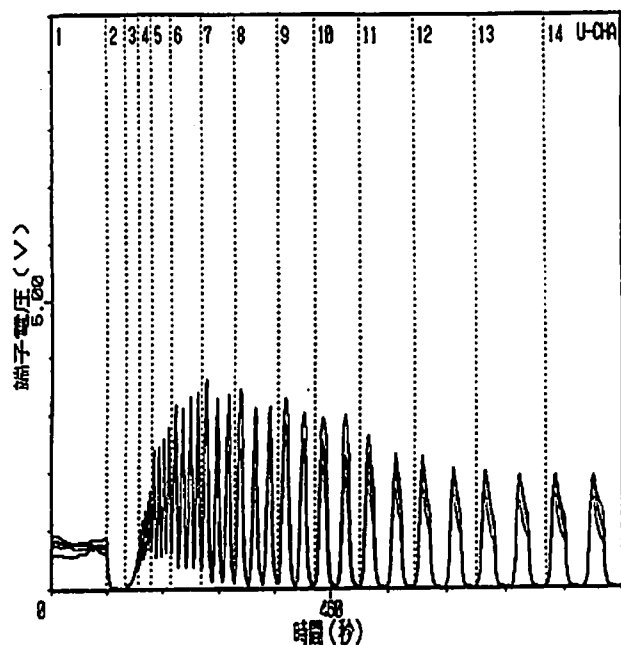


図 6 3. メタンガスセンサの応答

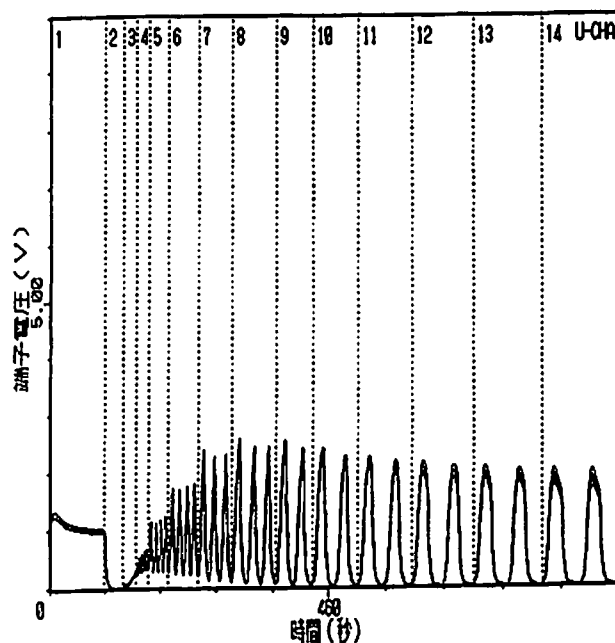


図 6 4. アルコールガスセンサの応答

7. 2 匂い測定パターンから各ガスセンサのF比特性を計算した結果

アンモニアガスセンサのF比特性を図65に、硫化水素ガスセンサのF比特性を図66に、メタンガスセンサのF比特性を図67に、アルコールガスセンサのF比特性を図68に示す。

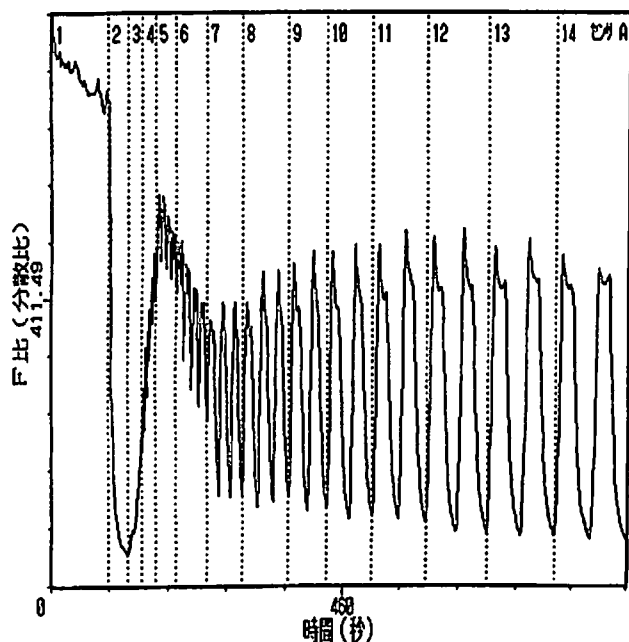


図65. アンモニアガスセンサのF比特性

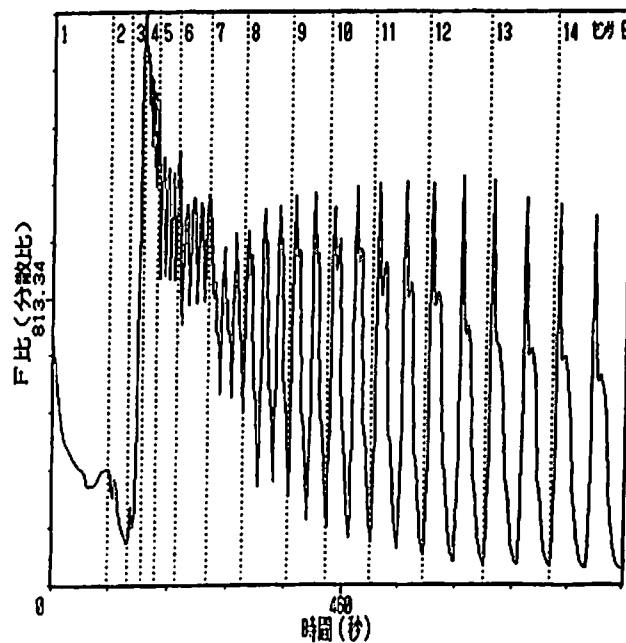


図66. 硫化水素ガスセンサのF比特性

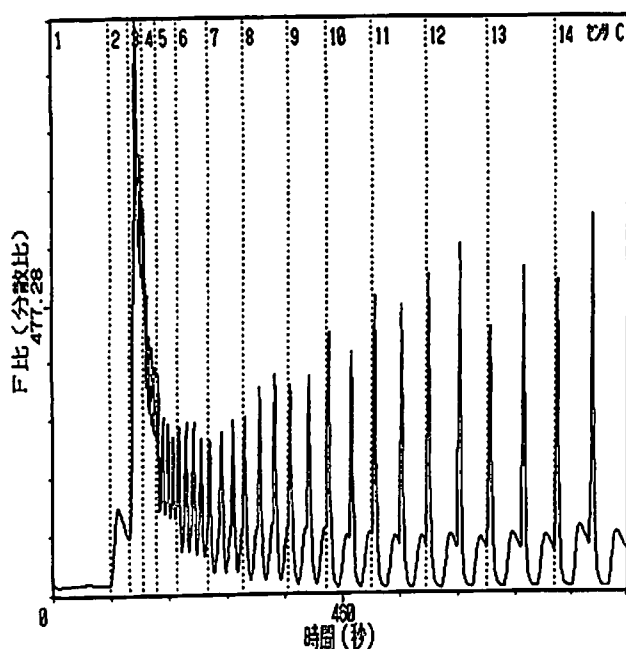


図67. メタンガスセンサのF比特性

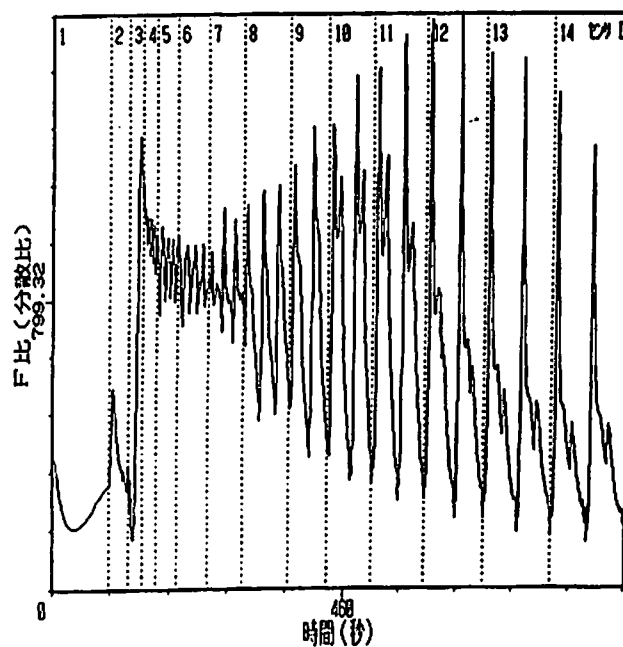


図68. アルコールガスセンサのF比特性

7.3 匂い測定パターンから匂い応答パターンを求めた結果

図65～図68で示した各ガスセンサのF比特性において最も分散比が大きい測定点に関して、匂い測定パターンから測定電圧値の集合（匂い応答パターン）を求めた。その結果を訓練サンプルについて表7～表18に示す。また、テストサンプルデータを表19～表30に示す。ただし訓練サンプルの結果については学習結果を併記して示す。表7～表30においてセンサA～センサDの対応は次に示す通りである。

- ・センサA：アンモニアガスセンサ（TGS824）
- ・センサB：硫化水素ガスセンサ（TGS825）
- ・センサC：メタンガスセンサ（TGS813）
- ・センサD：アルコールガスセンサ（TGS823）

表7. 普通酒4の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	5.018	4.760	4.800	4.910	4.870
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	39	20	20	403	36
	左勾配	20	20	20	20	210
B	測定値(V)	3.620	3.690	3.890	3.820	3.830
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	43	194	201	40	55
	左勾配	184	141	20	20	20
C	測定値(V)	5.550	5.690	6.180	6.200	6.280
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	47	154	74
	左勾配	20	20	83	53	106
D	測定値(V)	4.680	4.610	4.910	4.950	4.950
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	114	53	20	20
	左勾配	114	31	87	20	20

表 8. 普通酒 28 の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	3.870	3.870	3.710	3.870	3.880
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	274	20	20
	左勾配	84	84	64	98	75
B	測定値(V)	3.600	3.690	3.550	3.690	3.660
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	206	63	20	68	44
	左勾配	21	20	26	20	41
C	測定値(V)	4.380	4.430	4.100	4.250	4.240
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	20	20	22	20	20
D	測定値(V)	4.830	4.900	4.650	4.830	4.980
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	75	115	219	167	95
	左勾配	20	20	210	20	123

表 9. 吟醸酒 17 の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	3.810	3.490	3.770	3.570	3.750
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	248	20	86	20	68
	左勾配	27	121	81	40	144
B	測定値(V)	3.160	2.810	3.110	2.840	3.120
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	20	20	20	20	20
C	測定値(V)	3.540	3.150	3.620	3.270	3.640
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	22
	左勾配	20	20	20	20	20
D	測定値(V)	4.280	3.980	4.310	3.940	4.170
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	20	20	20	20	20

表 1 0 . 吟醸酒 2 4 の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	4.920	5.170	5.050	4.860	4.920
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	149	209	160	20
	左勾配	99	20	174	165	99
B	測定値(V)	3.950	4.060	4.080	3.930	4.030
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	22	51	59	20	38
	左勾配	102	24	22	229	34
C	測定値(V)	6.050	6.210	6.220	6.250	6.110
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	34	51	243	99
	左勾配	29	302	112	42	21
D	測定値(V)	4.840	4.970	5.010	4.960	4.920
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	36	207	21	23	20
	左勾配	370	20	23	216	342

表 1 1 . 泡盛の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	5.280	5.210	5.050	5.410	5.290
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	28	31	20
	左勾配	33	196	123	20	26
B	測定値(V)	4.510	4.400	4.270	4.620	4.300
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	21	20	20	44	20
	左勾配	20	20	50	20	39
C	測定値(V)	6.490	6.410	6.390	6.570	6.530
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	190	38	201	35
	左勾配	25	35	61	20	20
D	測定値(V)	5.640	5.500	5.560	5.710	5.480
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	30	20	20	51	20
	左勾配	20	20	20	20	20

表 1 2. V S O P の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	5.740	6.070	5.920	5.930	5.750
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	22	20	20	20
	左勾配	25	20	20	20	25
B	測定値(V)	4.850	4.980	5.280	5.170	4.980
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	47	20	20	20	20
C	測定値(V)	6.170	6.600	6.890	6.310	6.430
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	66	20	94	46	84
	左勾配	88	155	106	38	146
D	測定値(V)	5.920	6.040	6.020	6.130	5.980
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	54	22	25	20	31

表 1 3. エタノールの匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	6.520	6.540	6.730	6.920	6.590
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	23	20	20	20	20
B	測定値(V)	6.430	6.450	6.390	6.470	6.570
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	20	20	20	20	20
C	測定値(V)	7.410	7.320	7.550	7.620	7.480
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	35	188	20	20	23
D	測定値(V)	7.430	7.460	7.500	7.450	7.480
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	20	20	20	20	20

表 1 4 . ハッカ油の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	3.300	3.420	3.220	3.020	3.300
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	25	191	20	20	28
	左勾配	20	20	20	20	20
B	測定値(V)	7.970	7.900	7.790	7.900	7.750
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	20	20	20	20	20
C	測定値(V)	7.100	6.900	7.260	6.850	7.120
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	24	20	131	20	29
	左勾配	20	212	20	20	20
D	測定値(V)	8.690	8.660	8.480	8.600	8.570
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	20	20	20	20	20

表 1 5 . ポカリスウェットの匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	1.020	0.950	1.000	0.940	0.960
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	232	40	122	38	48
	左勾配	20	20	20	20	20
B	測定値(V)	0.220	0.190	0.210	0.180	0.210
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	274	31	121	24	68
	左勾配	20	20	20	20	20
C	測定値(V)	0.050	0.040	0.060	0.040	0.040
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	53	37	68	35	35
	左勾配	20	20	20	20	20
D	測定値(V)	0.540	0.460	0.550	0.470	0.460
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	74	70	178	115	70
	左勾配	402	20	171	20	20

表 1 6 . マスカットジュースの匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	1.170	1.480	1.450	1.450	1.380
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	111	20	20	20	215
	左勾配	86	29	40	40	105
B	測定値(V)	0.290	0.630	0.620	0.550	0.420
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	191	20	20	20	20
	左勾配	69	20	20	20	20
C	測定値(V)	0.110	0.370	0.390	0.300	0.240
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	37	20	20	20	20
D	測定値(V)	0.700	1.240	1.270	1.170	0.910
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	20	20	20	20	20
	左勾配	135	20	20	20	20

表 1 7 . コーヒーの匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	1.080	1.180	1.090	1.050	1.090
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	31	152	34	24	36
	左勾配	96	264	82	267	76
B	測定値(V)	0.210	0.220	0.220	0.210	0.220
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	86	89	112	86	72
	左勾配	168	66	52	169	87
C	測定値(V)	0.040	0.070	0.070	0.040	0.070
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	111	216	122	111	216
	左勾配	222	30	67	222	30
D	測定値(V)	0.500	0.530	0.560	0.520	0.514
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	39	216	161	64	46
	左勾配	249	90	177	119	166

表 1 8. ウーロン茶の匂い応答パターン

センサNo.	要素番号	1	2	3	4	5
A	測定値(V)	1.280	1.280	1.260	1.160	1.390
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	41	39	34	84	266
	左勾配	20	20	23	136	180
B	測定値(V)	0.290	0.310	0.260	0.270	0.310
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	60	90	43	45	90
	左勾配	20	20	20	20	20
C	測定値(V)	0.070	0.100	0.080	0.070	0.090
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	204	239	55	204	60
	左勾配	91	20	51	92	23
D	測定値(V)	0.700	0.670	0.620	0.620	0.640
	重み	1.000	1.000	1.000	1.000	1.000
	右勾配	214	47	32	32	37
	左勾配	20	20	20	20	20

表 1 9. 普通酒 4 のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	4.710	4.842	4.784	4.832	4.832
B	測定値(V)	3.761	3.815	3.796	3.869	3.752
C	測定値(V)	5.879	6.026	5.796	5.879	5.928
D	測定値(V)	4.754	4.788	4.759	4.813	4.735

表 2 0 . 普通酒 2 8 のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	4.011	3.923	3.854	3.820	4.079
B	測定値(V)	3.771	3.727	3.678	3.536	3.678
C	測定値(V)	4.466	4.294	4.353	4.216	4.441
D	測定値(V)	4.930	4.872	4.852	4.637	4.769

表 2 1 . 吟醸酒 1 7 のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	3.663	3.629	3.888	3.693	3.727
B	測定値(V)	2.842	2.910	3.218	3.077	3.209
C	測定値(V)	3.336	3.282	3.712	3.468	3.624
D	測定値(V)	3.903	3.991	4.133	3.947	4.182

表 2 2 . 吟醸酒 2 4 のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	5.057	4.999	4.901	4.925	4.823
B	測定値(V)	4.089	3.874	4.050	3.854	3.923
C	測定値(V)	6.231	6.119	6.241	5.977	6.114
D	測定値(V)	5.126	4.916	4.965	4.862	5.023

表 2 3 . 泡盛のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	5.596	5.150	5.317	5.302	5.160
B	測定値(V)	4.368	4.412	4.471	4.387	4.436
C	測定値(V)	6.222	6.447	6.437	6.491	6.393
D	測定値(V)	5.468	5.610	5.556	5.659	5.527

表 2 4 . V S O P のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	5.860	5.659	5.820	6.045	6.001
B	測定値(V)	5.033	4.974	5.141	5.209	5.234
C	測定値(V)	6.931	6.862	6.466	6.735	6.784
D	測定値(V)	5.923	5.833	6.168	6.109	5.918

表 2 5 . エタノールのテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	6.535	6.642	6.628	6.637	6.760
B	測定値(V)	6.495	6.432	6.481	6.349	6.476
C	測定値(V)	7.210	7.342	7.420	7.425	7.503
D	測定値(V)	7.498	7.459	7.449	7.420	7.430

表 2 6 . ハッカ油のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	3.302	3.008	3.253	3.277	3.238
B	測定値(V)	7.831	7.787	7.899	7.718	7.801
C	測定値(V)	7.268	6.539	7.244	7.151	7.087
D	測定値(V)	8.550	8.579	8.608	8.452	8.584

表 2 7 . ポカリスウェットのテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	0.998	1.008	0.973	0.973	0.998
B	測定値(V)	0.205	0.201	0.186	0.191	0.196
C	測定値(V)	0.054	0.054	0.049	0.049	0.049
D	測定値(V)	0.533	0.514	0.474	0.499	0.494

表 2 8 . マスカットジュースのテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	1.379	1.345	1.409	1.404	1.360
B	測定値(V)	0.460	0.470	0.538	0.562	0.479
C	測定値(V)	0.289	0.274	0.386	0.396	0.269
D	測定値(V)	0.968	1.052	1.096	1.145	1.061

表 2 9 . コーヒーのテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	1.052	1.071	1.110	1.061	1.066
B	測定値(V)	0.215	0.240	0.205	0.220	0.215
C	測定値(V)	0.059	0.059	0.064	0.054	0.059
D	測定値(V)	0.553	0.567	0.523	0.558	0.553

表 3 0 . ウーロン茶のテストサンプルデータ

センサNo.	データ番号	1	2	3	4	5
A	測定値(V)	1.198	1.203	1.208	1.198	1.208
B	測定値(V)	0.269	0.284	0.269	0.274	0.274
C	測定値(V)	0.073	0.073	0.073	0.059	0.078
D	測定値(V)	0.611	0.636	0.607	0.651	0.616

ア . 4 各ガスセンサの匂い標準パターンを求めた結果

相互作用可変法を用いて 1 2 種類の匂いサンプルの各 5 個の訓練サンプルを学習し、各ガスセンサの匂い標準パターンを求めた。アンモニアガスセンサの匂い標準パターンを図 6 9 に、硫化水素ガスセンサの匂い標準パターンを図 7 0 に、メタンガスセンサの匂い標準パターンを図 7 1 に、アルコールガスセンサの匂い標準パターンを図 7 2 に示す。

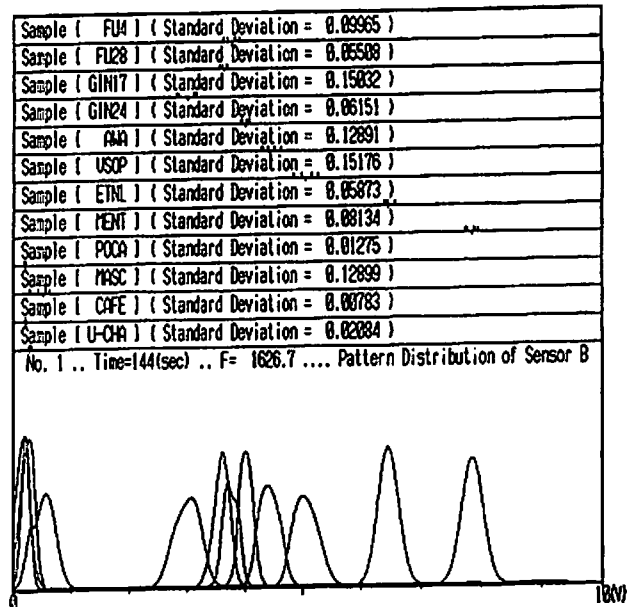
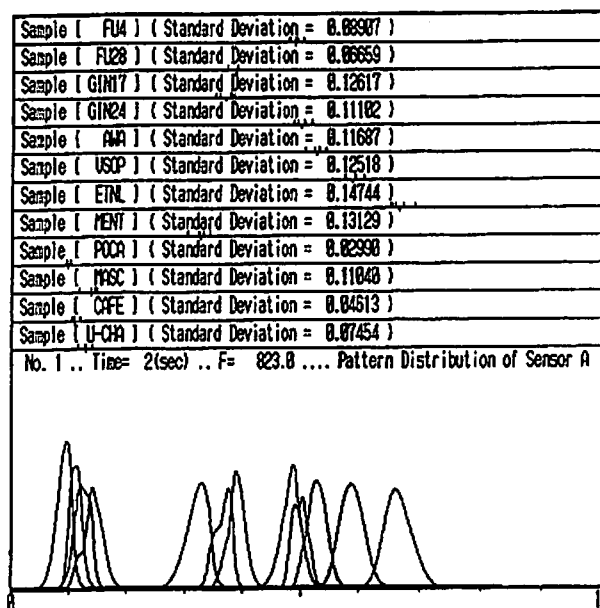


図 6 9 . アンモニアガスセンサ標準パターン 図 7 0 . 硫化水素ガスセンサ標準パターン

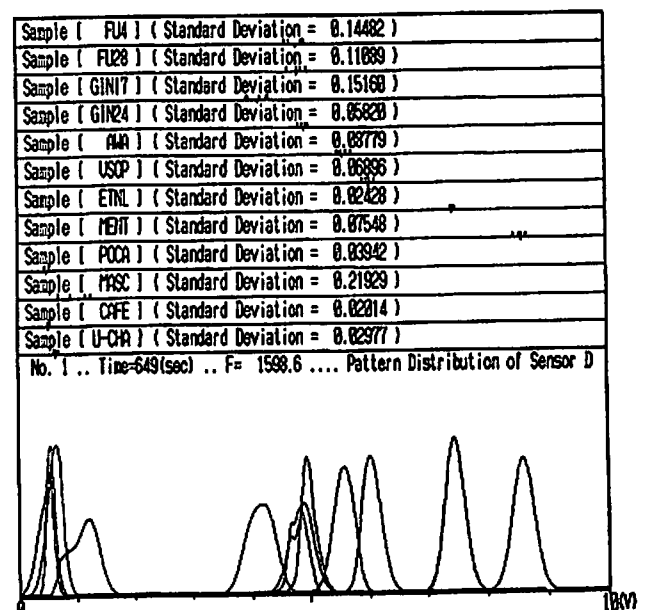
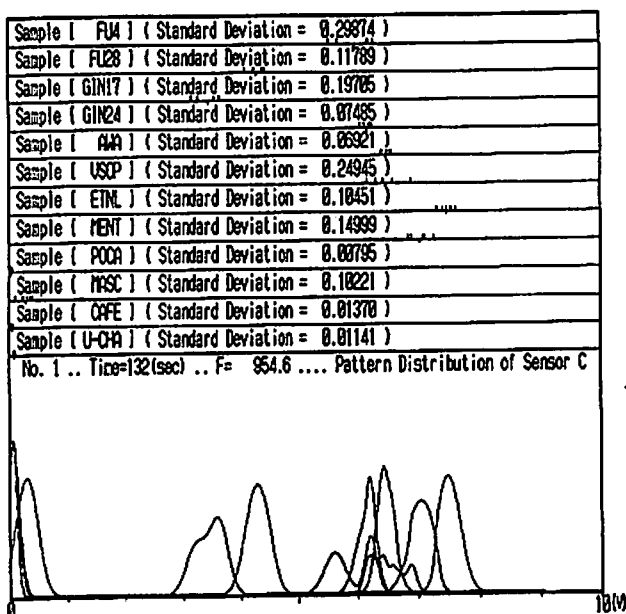


図 7 1 . メタンガスセンサ標準パターン 図 7 2 . アルコールガスセンサ標準パターン

7.5 各ガスセンサの匂い識別領域を求めた結果

各ガスセンサの匂い標準パターンから匂い識別領域を求めた。アンモニアガスセンサの匂い識別領域を図73に、硫化水素ガスセンサの匂い識別領域を図74に、メタンガスセンサの匂い識別領域を図75に、アルコールガスセンサの匂い識別領域を図76に示す。

(6.4参照)

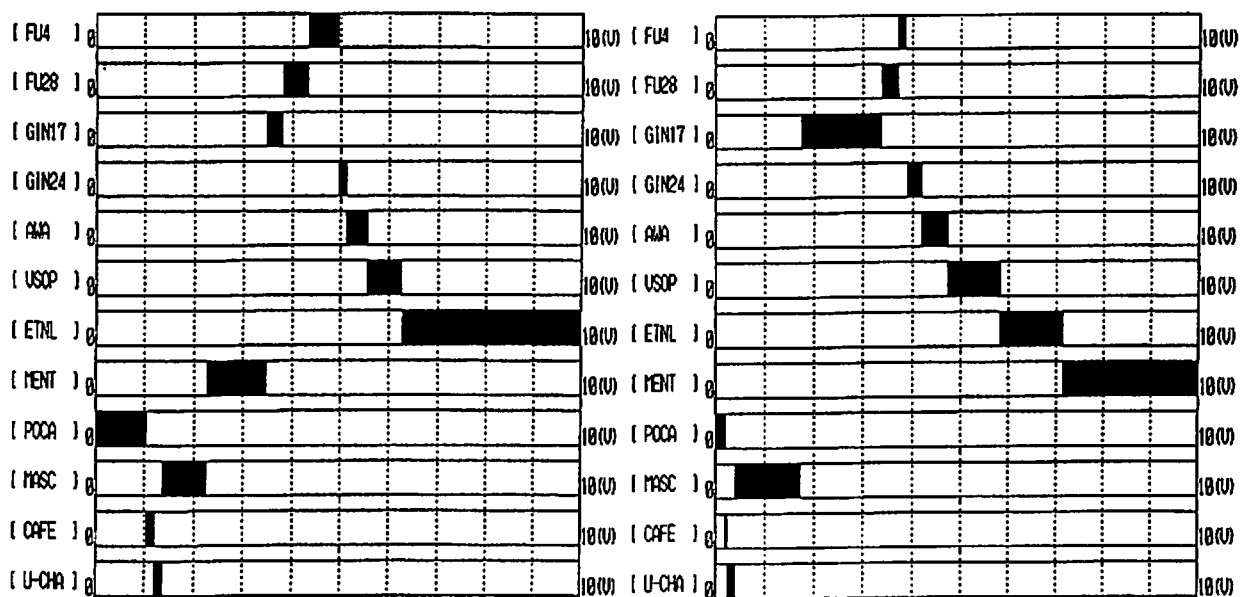


図73. アンモニアガスセンサ匂い識別領域 図74. 硫化水素ガスセンサ匂い識別領域

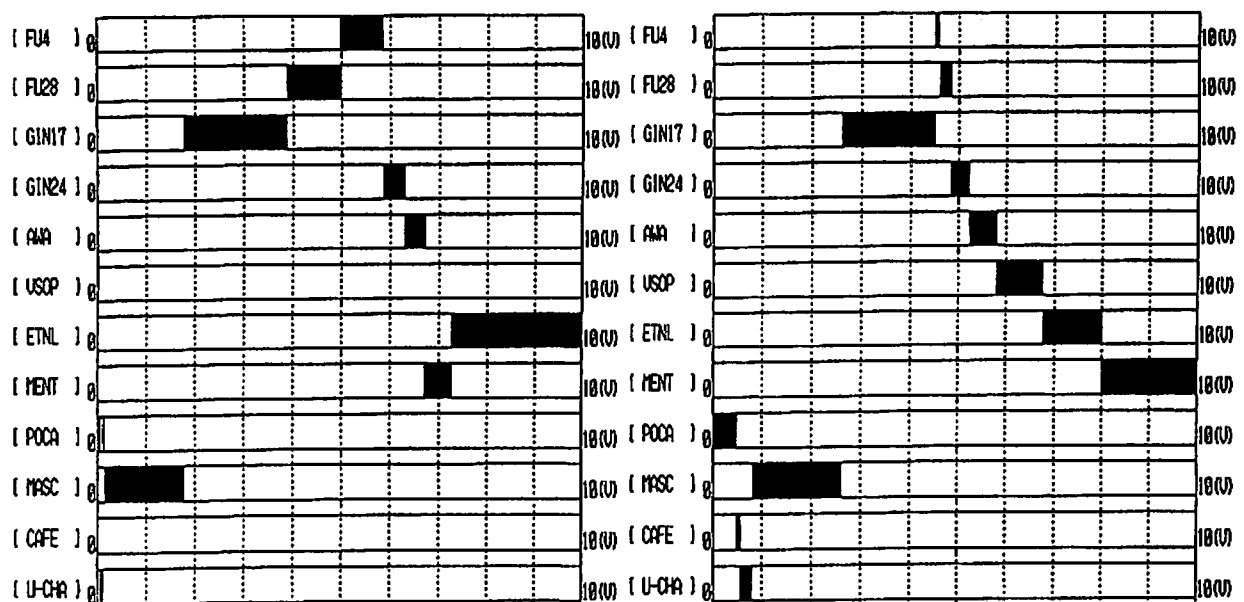


図75. メタンガスセンサ匂い識別領域 図76. アルコールガスセンサ匂い識別領域

7.6 相互作用可変法とニューラルネットワークの学習曲線を求めた結果

12種類の匂いサンプルの各5個の訓練サンプルについて、相互作用可変法を用いて匂いを学習したときの学習曲線を図77に、ニューラルネットワーク（TypeA～F）を用いて匂いを学習したときの学習曲線を図78～図83に示す。

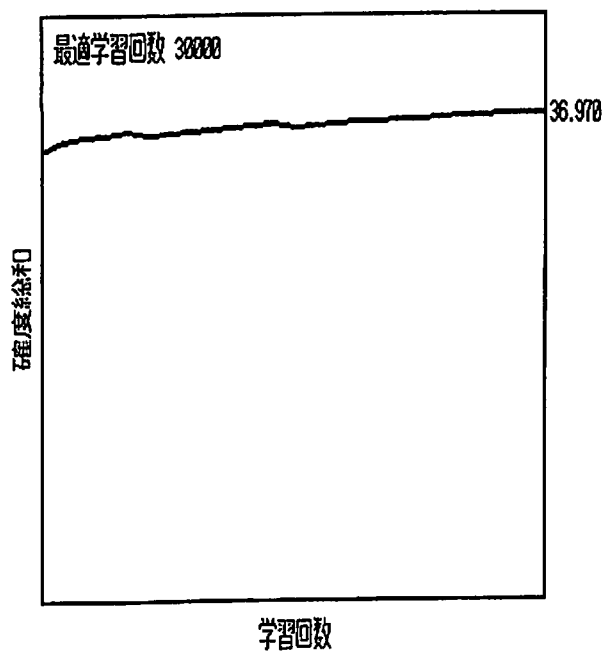


図77. 相互作用可変法の学習曲線

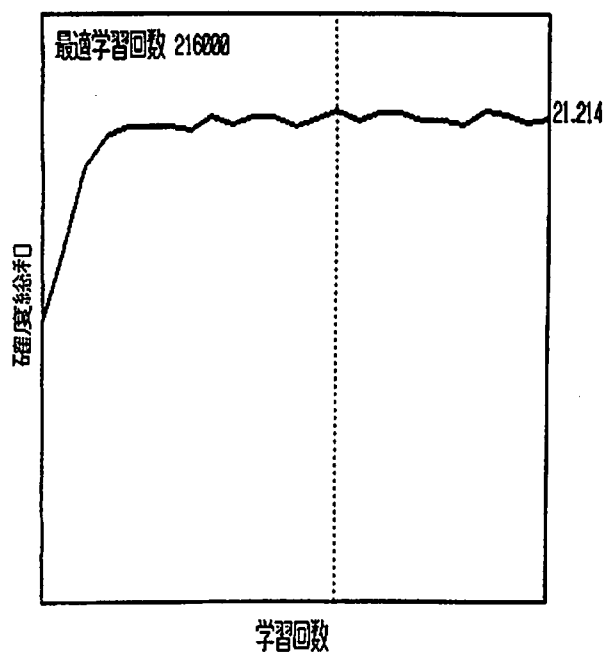


図78. N.N. (Type A) の学習曲線

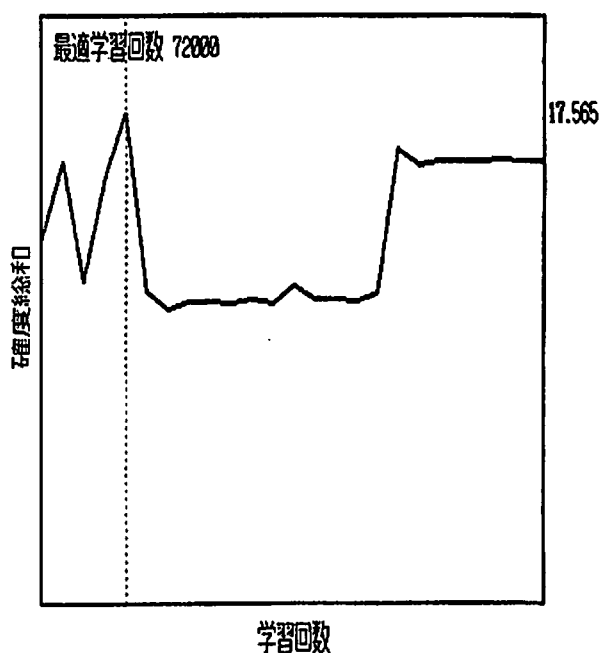


図79. N.N. (Type B) の学習曲線

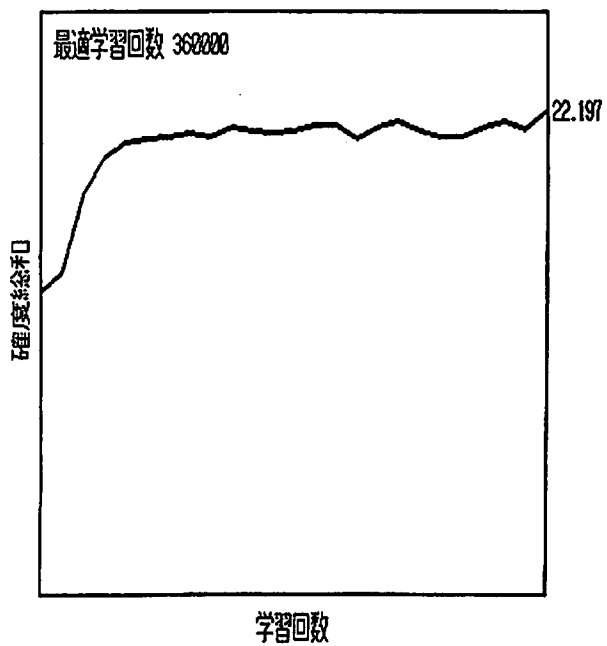


図 8 0 . N . N . (Type C) の学習曲線

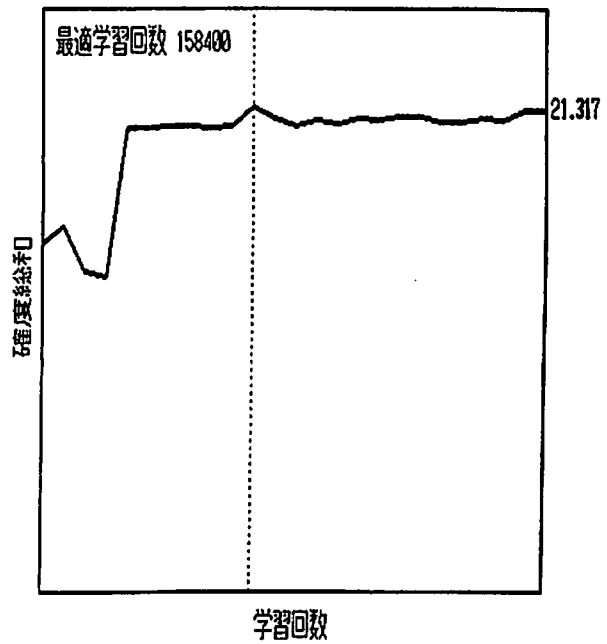


図 8 1 . N . N . (Type D) の学習曲線

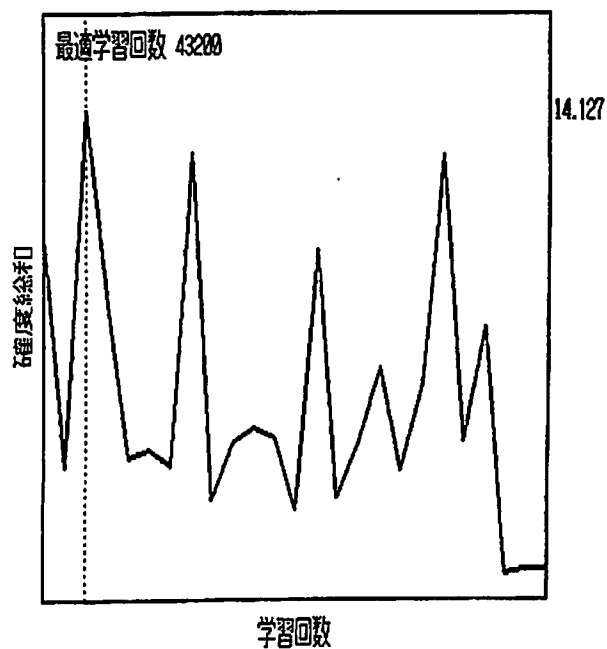


図 8 2 . N . N . (Type E) の学習曲線

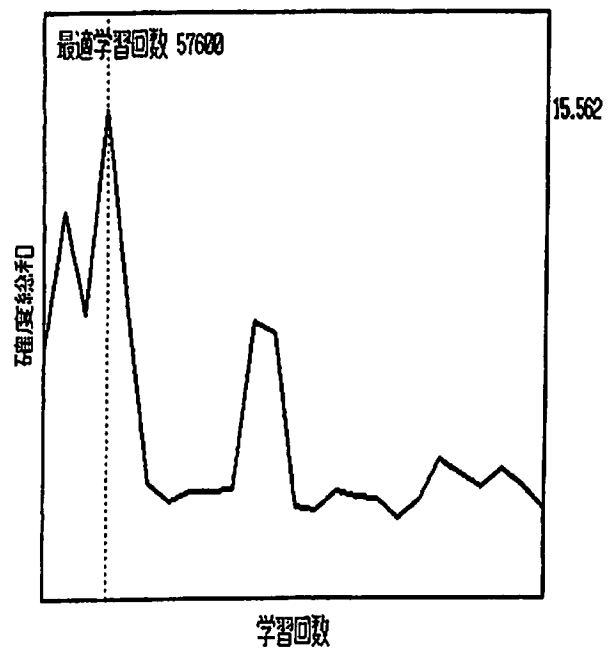


図 8 3 . N . N . (Type F) の学習曲線

ア . ア テストサンプルの識別結果

12種類の匂いサンプルの各5個ずつのテストサンプルを、相互作用可変法を用いて識別した際の結果を図84に、ニューラルネットワーク（TypeA～D）を用いて識別した際の結果を図85～図90に示す。外円の大きさは、識別ターゲットサンプルの識別が正解であったときの確実さを示し（円の下に0～1の値で表示）、内円の大きさは間違っ

	FL4	FL28	GIN17	GIN24	ALA	USOP	ETNL	MENT	POCA	MYSC	CAFE	U-CHA
1												
	0.505	0.400	0.926	0.615	0.616	0.823	0.898	0.917	0.343	0.721	0.344	0.379
2												
	0.478	0.506	0.944	0.461	0.836	0.832	0.965	0.784	0.348	0.718	0.338	0.405
3												
	0.539	0.528	0.761	0.567	0.913	0.762	0.993	0.943	0.374	0.858	0.358	0.377
4												
	0.461	0.684	0.925	0.379	0.918	0.907	0.993	0.983	0.364	0.864	0.344	0.402
5												
	0.492	0.589	0.904	0.534	0.818	0.871	0.999	0.994	0.357	0.738	0.346	0.387

図84．相互作用可変法の識別結果

	FL4	FL28	GIN17	GIN24	ALA	USOP	ETNL	MENT	POCA	MYSC	CAFE	U-CHA
1												
	0.063	0.094	0.081	0.163	0.186	0.230	0.123	0.059	0.998	0.988	0.918	0.972
2												
	0.063	0.094	0.081	0.163	0.186	0.230	0.123	0.059	0.946	0.975	0.998	0.988
3												
	0.063	0.094	0.081	0.163	0.186	0.230	0.123	0.059	0.998	0.974	0.997	0.967
4												
	0.063	0.094	0.081	0.163	0.186	0.230	0.123	0.059	0.998	0.973	0.984	0.976
5												
	0.063	0.094	0.081	0.163	0.186	0.230	0.123	0.059	0.941	0.975	0.993	0.978

図85．N.N.（Type A）の識別結果

	FL4	FL28	GIN17	GIN24	ALA	USOP	ETNL	MENT	POCA	MYSC	CAFE	U-CHA
1												
	0.149	0.064	0.187	0.157	0.202	0.101	0.136	0.082	1.00	0.968	0.888	0.991
2												
	0.149	0.064	0.187	0.157	0.202	0.101	0.136	0.082	1.00	0.968	0.888	0.993
3												
	0.149	0.064	0.187	0.157	0.202	0.101	0.136	0.082	1.00	0.968	0.888	0.990
4												
	0.149	0.064	0.187	0.157	0.202	0.101	0.136	0.082	1.00	0.968	0.888	0.990
5												
	0.149	0.064	0.187	0.157	0.202	0.101	0.136	0.082	1.00	0.968	0.888	0.996

図86．N.N.（Type B）の識別結果

	FU4	FU28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MSC	CAFE	U-CHA
1	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.994	0.995	0.835	0.983
2	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.930	0.991	0.979	0.991
3	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.989	0.990	0.996	0.977
4	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.995	0.999	0.963	0.988
5	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.916	0.991	0.995	0.989

図 8 7. N. N. (Type C) の識別結果

	FU4	FU28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MSC	CAFE	U-CHA
1	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.999	0.999	0.944	0.982
2	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.995	0.999	0.975	0.993
3	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.997	0.999	0.975	0.991
4	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.999	0.999	0.968	0.994
5	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.993	0.999	0.981	0.992

図 8 8. N. N. (Type D) の識別結果

	FU4	FU28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MSC	CAFE	U-CHA
1	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.000
2	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.000
3	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.000
4	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.000
5	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.000

図 8 9. N. N. (Type E) の識別結果

	FU4	FU28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MSC	CAFE	U-CHA
1	●	X	X	X	X	X	X	X	X	●	X	X
	0.159	0.183	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
2	●	X	X	X	X	X	X	X	X	●	X	X
	0.159	0.183	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
3	●	X	X	X	X	X	X	X	X	●	X	X
	0.159	0.183	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
4	●	X	X	X	X	X	X	X	X	●	X	X
	0.159	0.183	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
5	●	X	X	X	X	X	X	X	X	●	X	X
	0.159	0.183	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213

図 9 0. N. N. (Type F) の識別結果

7. 8 訓練サンプルの識別結果

12種類の匂いサンプルの各5個ずつの訓練サンプルを、相互作用可変法を用いて識別した際の結果を図91に、ニューラルネットワーク（TypeA～F）を用いて識別した際の結果を図92～図97に示す。外円の大きさは、識別ターゲットサンプルの識別が正解であったときの確実さを示し（円の下に0～1の値で表示）、内円の大きさは間違っ

	FLA	FLB	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1												
	0.396	0.687	0.846	0.493	0.914	0.682	0.989	0.991	0.333	0.156	0.358	0.432
2												
	0.548	0.478	0.899	0.575	0.859	0.834	0.953	0.915	0.383	0.958	0.343	0.454
3												
	0.385	0.628	0.892	0.599	0.784	0.847	1.00	0.922	0.333	0.921	0.343	0.398
4												
	0.481	0.545	0.950	0.587	0.862	0.723	1.00	0.949	0.385	0.894	0.347	0.376
5												
	0.397	0.478	0.918	0.570	0.906	0.767	0.998	0.989	0.377	0.593	0.347	0.431

図91. 相互作用可変法の識別結果

	FLA	FLB	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1												
	0.063	0.094	0.081	0.163	0.186	0.238	0.123	0.059	0.979	0.974	0.998	0.985
2												
	0.063	0.094	0.081	0.163	0.186	0.238	0.123	0.059	0.997	0.973	0.982	0.984
3												
	0.063	0.094	0.081	0.163	0.186	0.238	0.123	0.059	0.998	0.973	0.996	0.998
4												
	0.063	0.094	0.081	0.163	0.186	0.238	0.123	0.059	0.998	0.974	0.998	0.974
5												
	0.063	0.094	0.081	0.163	0.186	0.238	0.123	0.059	0.994	0.997	0.998	0.996

図92. N.N. (Type A) の識別結果

	FLA	FLB	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1												
	0.149	0.064	0.187	0.157	0.282	0.181	0.136	0.082	1.00	0.968	0.088	0.987
2												
	0.149	0.064	0.187	0.157	0.282	0.181	0.136	0.082	1.00	0.969	0.088	0.989
3												
	0.149	0.064	0.187	0.157	0.282	0.181	0.136	0.082	1.00	0.968	0.088	0.987
4												
	0.149	0.064	0.187	0.157	0.282	0.181	0.136	0.082	1.00	0.968	0.088	0.998
5												
	0.149	0.064	0.187	0.157	0.282	0.181	0.136	0.082	1.00	0.968	0.088	0.998

図93. N.N. (Type B) の識別結果

	FL4	FL28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.972	0.983	0.997	0.993
2	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.996	0.989	0.983	0.993
3	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.992	0.988	0.995	0.988
4	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.996	0.991	0.980	0.988
5	X	X	X	●	X	X	X	X	●	●	●	●
	0.130	0.185	0.038	0.479	0.002	0.002	0.158	0.002	0.994	0.996	0.996	0.991

図 9 4 . N . N . (Type C) の識別結果

	FL4	FL28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.993	0.987	0.976	0.994
2	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.998	0.989	0.974	0.993
3	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.999	0.989	0.979	0.992
4	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.999	0.989	0.980	0.993
5	X	X	X	X	X	X	X	●	●	●	●	●
	0.076	0.105	0.114	0.119	0.128	0.094	0.100	0.259	0.997	0.989	0.975	0.991

図 9 5 . N . N . (Type D) の識別結果

	FL4	FL28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.080
2	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.080
3	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.080
4	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.080
5	X	X	X	X	X	X	X	●	X	X	X	X
	0.061	0.083	0.104	0.090	0.094	0.081	0.088	0.161	0.065	0.093	0.080	0.080

図 9 6 . N . N . (Type E) の識別結果

	FL4	FL28	GIN17	GIN24	AAA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1	●	X	X	X	X	X	X	X	●	X	X	X
	0.159	0.103	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
2	●	X	X	X	X	X	X	X	●	X	X	X
	0.159	0.103	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
3	●	X	X	X	X	X	X	X	●	X	X	X
	0.159	0.103	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
4	●	X	X	X	X	X	X	X	●	X	X	X
	0.159	0.103	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213
5	●	X	X	X	X	X	X	X	●	X	X	X
	0.159	0.103	0.098	0.131	0.114	0.098	0.121	0.153	0.196	0.400	0.183	0.213

図 9 7 . N . N . (Type F) の識別結果

7.9 相互作用可変法とニューラルネットワークを組み合わせた場合の識別結果

12種類の匂いサンプルの各5個の訓練サンプルに対するニューラルネットワークおよび相互作用可変法の識別結果から、両手法を組み合わせて各種匂いサンプルを識別する際の識別器重みを求めた（6.7参照）。求めた識別器重みを表31に示す。求めた識別器重みを用いて相互作用可変法とニューラルネットワークの両手法を組み合わせて、12種類の匂いサンプルの各5個のテストサンプルを識別した際の結果を図98に示す。ただし、ニューラルネットワークは、最も識別率の良いタイプCのネットワークを使用した。

表31. 識別器重みを求めた結果

	相互作用可変法	N.N. (type C)
普通酒 4	1.000	0.000
普通酒 2 8	1.000	0.000
吟醸酒 1 7	1.000	0.000
吟醸酒 2 4	0.481	0.519
泡盛	1.000	0.000
V S O P	1.000	0.000
エタノール	1.000	0.000
ハッカ油	1.000	0.000
ポカリスウェット	0.047	0.953
マスカットジュース	0.368	0.632
コーヒー	0.032	0.968
ウーロン茶	0.148	0.852

	FU4	FU28	GIN17	GIN24	AWA	USOP	ETNL	MENT	POCA	MASC	CAFE	U-CHA
1	●	●	●	●	●	●	●	●	●	●	●	●
	0.124	0.066	0.857	0.352	0.434	0.676	0.780	0.837	0.946	0.813	0.654	0.837
2	●	●	●	●	●	●	●	●	●	●	●	●
	0.200	0.189	0.899	0.201	0.759	0.708	0.931	0.615	0.825	0.812	0.934	0.859
3	●	●	●	●	●	●	●	●	●	●	●	●
	0.219	0.231	0.523	0.292	0.855	0.539	0.986	0.888	0.939	0.893	0.963	0.825
4	●	●	●	●	●	●	●	●	●	●	●	●
	0.128	0.497	0.854	0.153	0.860	0.851	0.986	0.969	0.949	0.895	0.902	0.852
5	●	●	●	●	●	●	●	●	●	●	●	●
	0.148	0.238	0.810	0.233	0.730	0.798	0.997	0.991	0.798	0.824	0.945	0.850

図98. 相互作用可変法とニューラルネットワークを組み合わせた場合の識別結果

7. 1 O 各種匂いサンプルに適したガスセンサの選択結果

各種匂いサンプルの匂い識別領域における区間の広さを各ガスセンサごとに求めることで、各種匂いサンプルに適したガスセンサの選択を行った。その結果を図99に示す。

Sample [FU4]	Suitable Sensor=C	(Reliability 0.1639	評価[やや難しい]
Sample [FU28]	Suitable Sensor=C	(Reliability 0.2443	評価[やや可能]
Sample [GIN17]	Suitable Sensor=C	(Reliability 0.7884	評価[確実]
Sample [GIN24]	Suitable Sensor=C	(Reliability 0.2461	評価[やや可能]
Sample [AWA]	Suitable Sensor=D	(Reliability 0.7276	評価[確実]
Sample [VSOP]	Suitable Sensor=B	(Reliability 0.7144	評価[確実]
Sample [ETNL]	Suitable Sensor=A	(Reliability 0.9360	評価[確実]
Sample [MENT]	Suitable Sensor=B	(Reliability 0.8599	評価[確実]
Sample [POCA]	Suitable Sensor=A	(Reliability 0.8916	評価[確実]
Sample [MASC]	Suitable Sensor=D	(Reliability 0.8474	評価[確実]
Sample [CAFE]	Suitable Sensor=A	(Reliability 0.8795	評価[確実]
Sample [U-CHA]	Suitable Sensor=D	(Reliability 0.8444	評価[確実]

図99. 各種匂いサンプルに適したガスセンサの選択結果

以上の各種匂いサンプルに適したガスセンサの選定結果から各ガスセンサに適した匂いを選択した。その結果を図100に示す。

>>> Role of Sensor A (アンモニアガスセンサ) :

ETNL(原 臭) POCA(アイソトニック飲料) CAFE(コーヒー)

Reliability for all allotted samples : 0.3353 評価[可能]

>>> Role of Sensor B (硫化水素ガスセンサ) :

VSOP(ブランデー) MENT(原 臭)

Reliability for all allotted samples : 0.7872 評価[確実]

>>> Role of Sensor C (メタンガスセンサ) :

FU4(清 酒) FU28(清 酒) GIN17(吟 醸 酒) GIN24(吟 醸 酒)

Reliability for all allotted samples : 0.3408 評価[可能]

>>> Role of Sensor D (アルコールガスセンサ) :

AWA(焼 酎) MASC(ジュース) U-CHA(お 茶)

Reliability for all allotted samples : 0.4436 評価[やや確実]

図100. 各ガスセンサに適した匂いの選択結果

第 8 章 考察及び検討

8.1 匂い測定装置に関する考察・検討

本研究では現在までに 2 台の匂い測定装置を製作した。まず始めに 1 台目の匂い測定装置について考察する。図 101 に 1 台目の匂い測定装置の見取り図を示す。

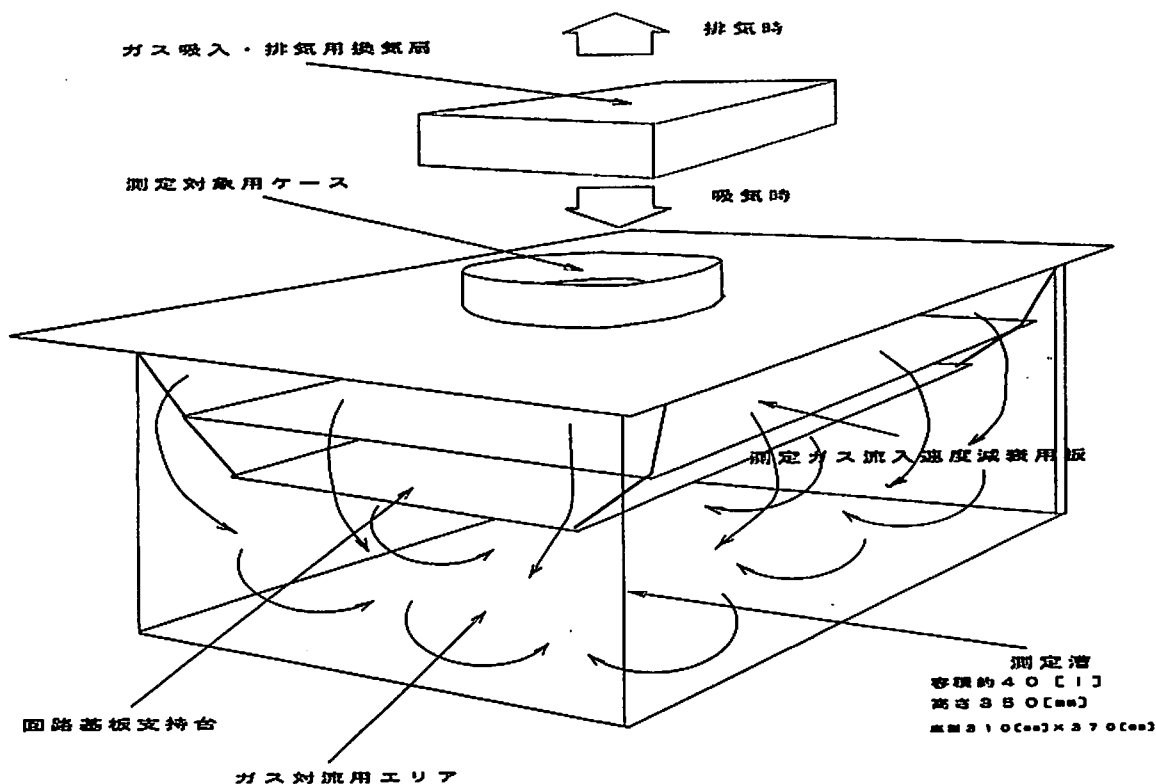


図 101. 1 台目の匂い測定装置の見取り図

次に 1 台目の匂い測定装置の問題点を示す。

- ・ 吸気口および排気口が独立していないために吸排気を十分に行えない。
- ・ ファンにより外部から吸気してパレット上の匂いサンプルを気化させる方式であるので、外部の雑ガスの影響を受ける上に、気化する匂いサンプルの量が一定でない。

以上の問題点を改善しなかったために、匂い応答パターン内の分散は平均して 1.8(V) となった（分散は各ガスセンサごとに異なる）。

半導体ガスセンサは素子表面における匂い分子の吸着に伴う電気的变化を検出するために匂いサンプルの「濃度の違い」と「種類の違い」を明確に区別できない。このため現状では半導体ガスセンサを用いて匂いを識別するためには、「ガス濃度を一定にする」あるいは「ガス種を 1 つに限定する」必要がある。1 台目の測定装置では、この点を十分に考

慮していないため、得られた匂い測定パターンによる匂いの種類の識別は困難であった¹⁶⁾。
2 台目の匂い測定装置は、1 台目の問題点を十分に考慮して設計・製作した（吸排気、サンプル気化量の問題点について）。

次に 2 台目の匂い測定装置について考察を行う（図 8 参照）。まず、2 台目の測定装置の問題点を次に示す。

- ・標準空気を生成するためにシリカゲルと活性炭を使用しているが、循環扇の出力が低いため、シリカゲルと活性炭の設置密度を小さくしているので十分な効果が得られていない。
- ・匂いサンプルがマイクロシリンジによりマニュアルで導入されているので、導入時刻のずれが生じてしまう。

以上の問題点を解決するために、

- ・シリカゲルと活性炭設置部に空気循環用のファン（12(W)以上の大きめの出力）を新たに導入する。
- ・匂いサンプルの導入を自動化することで測定を完全に自動化する。

といった改善策をとることが必要である。匂いを測定する際の再現性は学習サンプル数の設定に大きな影響を与えるので、匂いセンシングシステムにおいては非常に重要な要素である。匂い測定装置のさらなる改善が望まれる。

8.2 半導体ガスセンサに関する考察・検討

本研究では、半導体ガスセンサのヒータ電源の ON/OFF 制御を任意の時間間隔で行うことで、通常の測定方法よりも、選定した 12 種類の匂いサンプルの測定パターン間の分離が良くなる測定点を得られることが判明した（各ガスセンサの F 比分布より明らか）。ここでは、半導体ガスセンサの動作温度について考察することにする。

一般に半導体ガスセンサには、〇〇用ガスセンサ等と名前が付けられている。例外なく使用した 4 つの半導体ガスセンサにも型番以外の名前が付けられている。

- ・アンモニアガスセンサ（TGS824）
- ・硫化水素ガスセンサ（TGS825）
- ・メタンガスセンサ（TGS813）
- ・アルコールガスセンサ（TGS823）

センサ素子製造の際の添加物が違うこと以上に重要なことは、それぞれの素子の動作温

度が異なるという事実である。センサ素子の検出対象となる分子種は素子の動作温度に依存する。従って市販のガスセンサに付けられている名前は、センサ素子の動作温度（明確に言えばヒーターの電源電圧）を変えてしまうとほとんど意味を持たなくなる。

一方、F比分布（図65～図68）を考察することで次のことが分かる。

- ・アンモニアガスセンサに関しては、選定した12種類の匂いサンプルを測定する際にヒータ電源のON/OFF制御する必要はない。

（理由）通常の測定方法（行程番号1）で得られるF比より大きなF比を行程2以降で得られていない。

- ・硫化水素ガスセンサ・メタンガスセンサ・アルコールガスセンサに関しては、選定した12種類の匂いサンプルを測定する際に、ヒータ電源のON/OFF制御を行うことで、通常の測定方法よりも匂い応答パターン間の分離がかなり良い測定点が得られる。

ここで「なぜ、アンモニアガスセンサだけがヒータ電源のON/OFF制御の意味を失ってしまうのだろうか？」という疑問が沸く。この疑問は、センサ素子の動作温度との関連で容易に解消される。ヒータ電源のON/OFF制御は「センサ素子にあらかじめ設定された動作温度よりも低い温度でガスの吸着を行う」ことを意味している。アンモニアガスセンサの素子温度が下がっても12種類の匂いサンプルに含まれている分子種に対して有意な情報が得られない理由は、アンモニアガスセンサの素子の動作温度以上の領域に12種類の匂いサンプルの主な分子種が存在すると考えるのが自然である。従って選定した12種類の匂いサンプルに関してアンモニアガスセンサのヒータ電源のON/OFF制御を行う必要はない。また、以上からアンモニアガスセンサの動作温度は4つのガスセンサ中で最も低いと推定することもできる。

以上から、複数種の半導体ガスセンサを使用してさらにヒータ電源のON/OFF制御を行うことは、「分子の電気的性質に関するスペクトルにおいて、すだれ状にペンキを塗るようにして匂い分子の情報を得る」ことを意味するのである。今後は、ヒータ電圧をいわば「掃引」制御（5(V)から0(V)までヒータ電圧を少しずつ下げる）することで、個々のセンサにおける「分子の電気的性質に関するスペクトル」を求めることにより、匂い分子に関する情報がより定性的に得られるようになることが期待される。

8.3 相互作用可変法による匂いの学習・識別に関する考察・検討

特徴抽出部より得られる匂い応答パターンの学習を行う前に、訓練サンプル数について検討しておく必要がある。訓練サンプル数は一般に多いことが望ましいが、ガスセンサの場合、センサ素子表面で発生する突然の結晶成長による特性変化やガス濃度の不均一性に影響を受けるため、匂い応答パターンの分布関数を想定できないという困難さがある。

本研究では、匂い領域の区間性を重視して最も分散の大きいデータを訓練サンプルとして使用し、訓練サンプル数は5つという非常に少ない数とした。これらの訓練サンプルを相互作用可変法により学習し、それぞれ5つのテストサンプルを識別した結果、訓練サンプル数が5つという少なさにもかかわらず、選定した12種類の匂いサンプルの各5個のテストサンプル合計60個のうち59個のサンプルを正しく識別することができた。

しかし相互作用可変法では、ポカリスウェット・コーヒー・ウーロン茶に関して十分な大きさの識別余裕度が得られなかった。相互作用可変法により生成した匂い標準パターンを調べた結果、これらの匂いサンプルの標準パターンは互いに近接していることが分かった。これは匂い標準パターンが連続な分布である以上、近接パターンに関してさらに識別余裕度を大きくすることは容易でない。この状況を改善する方法は、

- ・ 識別対象サンプルの匂い識別領域が広いガスセンサと取り替える。
- ・ 訓練サンプル数を増やす。

の2つが考えられる。しかし、これらの改善策には

- ・ 市販半導体ガスセンサの種類は限られている。
- ・ 必ずしも訓練サンプルの増加＝識別余裕度の改善にならない（分布を決定する要因(ガス濃度・湿度・気温・センサ使用経歴)が複合的であるため）。

という問題点がある。

一方で、相互作用可変法では各ガスセンサごとに独立して識別対象電圧の各サンプルに対する相互作用を求めるので、ガスセンサごとの識別性能を調べるのが容易である。さらに学習により得られた匂い標準パターンから匂い識別領域を求めることで、匂いサンプルに適したガスセンサの選択やガスセンサに適した匂いサンプルの選択を容易に行える。これらの点は相互作用可変法の大きな利点であると考えられる。また、相互作用可変法による匂いの学習では、必要な学習回数がニューラルネットワークより少なく、学習曲線がニューラルネットワークよりも安定していることは注目すべき性質である。

8.4 ニューラルネットワークによる匂いの学習・識別に関する考察・検討

当初、客観的に相互作用可変法を評価するために特徴抽出部から得た匂い応答パターンをニューラルネットワークに入力した。ニューラルネットワークは、中間層の構造によりその内部情報表現形態が変わる。3層ニューラルネットワークは、連続写像を表す関数のフーリエ変換表現の形態で写像を実現していることは、既に証明されている⁸⁾。しかしながら4層・5層・・・となるとまだ不明な点が多い。本研究では、6種類の間層構造を持つ多層ニューラルネットワークを用いて匂いの識別を試みた。結果的には、ネットワーク構造を複雑にするほど識別結果は悪化した。原因としては、不必要に複雑な構造をもつニューラルネットワークを使用したために過剰学習が発生したということが考えられる（各学習曲線より明か）。ニューラルネットによる匂いの識別では、ポカリスウェット・コーヒー・ウーロン茶等の相互作用可変法が識別を苦手としている匂いサンプルに関する識別結果が著しく良かった。しかし、相互作用可変法が識別を得意としている匂いサンプルの識別に関しては著しく悪かった。両手法の識別結果を重ね合わせることで相互作用可変法とニューラルネットワークは相補的な性格を持っていることが判明した。識別結果と各匂いサンプルの匂い応答パターンを調べた結果、ニューラルネットワークは訓練サンプル内の分散が小さい離散的なパターンの識別に適していることが分かった。このことは、3層ニューラルネットワークが連続写像を表す関数をフーリエ変換表現の形態で写像を実現していることの証明を裏付ける結果となったと思われ興味深い。また、少数の訓練サンプル内の分散が大きく各匂い応答パターンが互いに重なり合うような場合の識別（連続的な分布を想定するのが有利な場合）を苦手としていることも識別結果より推定できたのも興味深い。ニューラルネットワークを使用する場合の問題点を以下に示す。

- ・いつ何時過剰学習（学習の対象としない性質までも細部まで学習してしまう現象）となるのか明確ではない。
- ・局所最適解に収束するのを避けるためにSA（Simulated Annealing）法などを使うことも考えられるが、その場合増大する計算量が問題となる。
- ・相互作用可変法のように匂い識別領域を生成できないので、ガスセンサごとに評価するのが困難である。
- ・ニューラルネットワークのみによる学習では多くの訓練サンプルを使用するのが望ましいが、匂い応答パターンの分布関数は収束する保証がないので、訓練サンプルを増やしても識別率が改善しない恐れがある。

8.5 相互作用可変法とN.N.を組み合わせた場合の識別結果の考察・検討

本研究では、相互作用可変法とニューラルネットワークは相補的な性格を持っていることが判明したので両方の手法を組み合わせた識別器を構成した。具体的には、訓練サンプルに対する識別結果をもとに識別器重みを求めてから、テストサンプルをそれぞれの識別器重みで識別を行った。その結果、12種類の匂いサンプルの内9種類の匂いサンプルについて十分な識別余裕度を得ることができた。これにより事実上少ない訓練サンプルで選定した12種類の匂いサンプルの識別が可能となった。

匂い分子は電氣的・重量的性質を持ち合わせている。半導体ガスセンサでは、匂い分子の電氣的性質に基づいた識別しかできない。近年、水晶振動子等の分子の吸着に伴う重量変化を共振周波数の変化 ΔF として検出し、その変化パターンより匂いの識別を行う方式の匂いセンシングシステムが盛んに研究されている³⁾。半導体ガスセンサにより匂い分子の電氣的性質に関する情報を、水晶振動子により匂い分子の重量的性質に関する情報を獲得し、本研究で考案した識別器を複合的に構成することにより、より多種多様な匂いを識別できる匂いセンシングシステムの開発も可能であると考える。図102に半導体ガスセンサと水晶振動子を組み合わせた場合の匂いセンシングシステムの構成図を示す。

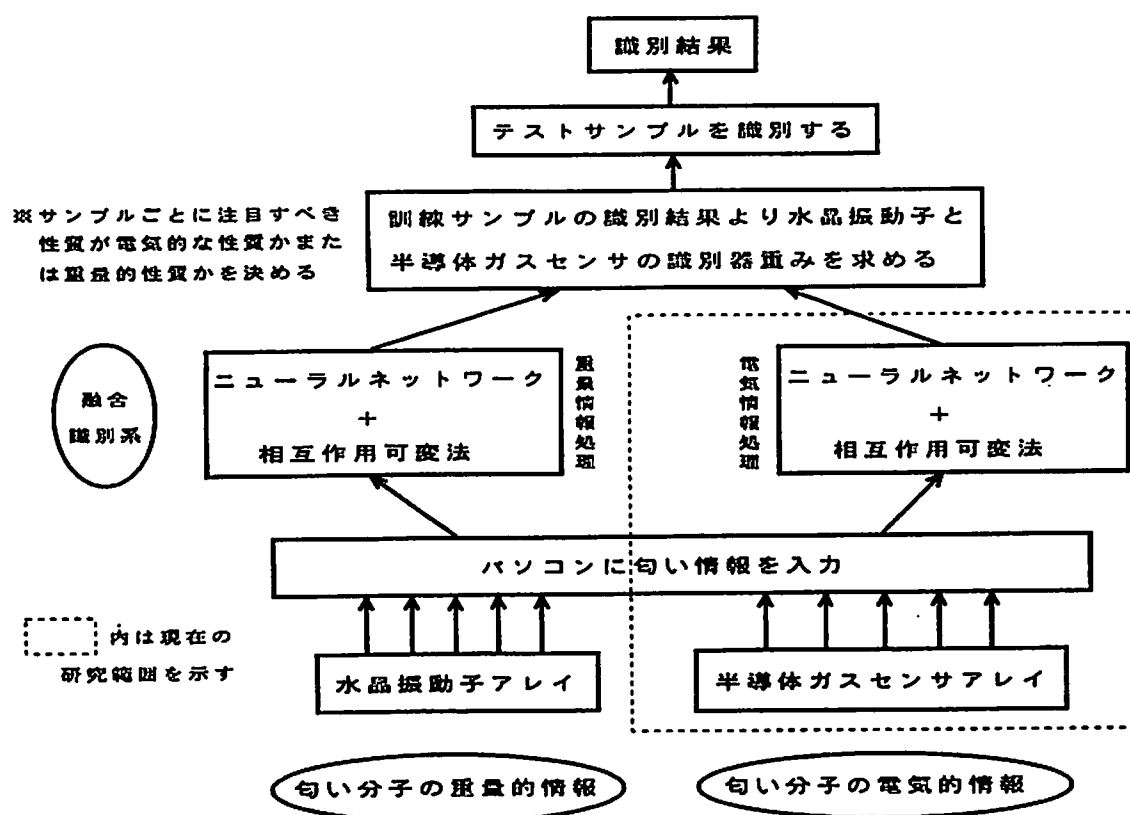


図102. 半導体ガスセンサ・水晶振動子による匂いセンシングシステムの構成

第 9 章 むすび

本研究では、4 種類のそれぞれ特性の異なる半導体ガスセンサを匂い検出素子として使用し、匂いセンシングシステムの構築を行った。匂いセンシングシステムは、匂いの測定・学習・識別をコンピュータを利用して行うものである。

匂いの測定はパーソナルコンピュータにより自動化した。測定の際には、半導体ガスセンサ内部のヒータ電源の ON/OFF 制御を行うことで、通常の測定方法よりも選定した 12 種類の匂いサンプルに対して有意な匂い情報を得られることが判明した。

さらに相互作用可変法を考案し、相互作用可変法とニューラルネットワークを組み合わせた識別器を構成することで、選定した 12 種類の匂いサンプルの学習および識別を行った。その結果、60 個のテストサンプル（各匂いサンプルにつき 5 個のテストサンプル）を全て正しく識別できた。特に、相互作用可変法とニューラルネットワークを組み合わせた場合の識別系では、12 種類の匂いサンプルの内 9 種類について十分な識別余裕度を得ることができた。

また相互作用可変法による匂いの学習の結果得られる匂い標準パターン群より匂い識別領域を求めることで、選定した 12 種類の匂いサンプルそれぞれの測定に適した半導体ガスセンサの選定、各ガスセンサが識別を得意とする匂いサンプルの選定を行うことができた。以上の結果、構築した匂いセンシングシステムにより、限定サンプル・限定環境における匂いの識別が可能となった。

しかしながら、現在のシステムでは生体で行われている匂いセンシングには全く及ばない（匂いの名前などを忘れない点では優れているかも知れないが）。なぜなら生体嗅覚機構は、匂い分子の電氣的＋重量的＋化学的性質に基づいて匂いを認識していると考えられ、さらに、匂いを様々な方法で嗅ぐことにより多面的かつ柔軟に学習しているからである。このような生体での匂い認識能力に近づくには、匂い分子の多面的な性質を検出できるセンサの開発と、コンピュータの処理速度のさらなる向上と記憶容量の大規模化が不可欠である。

本システムの今後の改善策については第 8 章で述べたが特に次の点が挙げられる。

- 1) ヒータ電源の ON/OFF 制御に変わって任意のヒータ電源電圧における測定を行い、より定性的に匂い分子の情報を得られる測定系にする。
- 2) 匂いサンプルの導入を自動化することで匂いの測定を完全に自動化する。
- 3) 匂い分子の重量的性質に関する情報を得るために水晶振動子を新たに導入し、

半導体ガスセンサと水晶振動子から得られる匂い情報を複合的に識別する識別系を設計する。

以上の改善策により限定サンプル・限定環境における本システムでの匂い識別能力はかなり向上することが予想される。しかし、ロボットなどの嗅覚系に応用する場合、「限定環境」という条件は望ましくない。本システムでの「限定環境」とは、言い替えれば「一定のガス濃度における測定環境」を意味する。ガス濃度に識別系が対応するためには次に示す方策が考えられる。

- 1) 識別対象サンプルの各濃度における匂い識別領域を生成する（ガス濃度に関する知識データとして記憶しておく）。
- 2) テストサンプルを識別器に入力して、識別結果が最も整合する匂い識別領域を使用するのが最も有意義であると想定する。

しかしながら、この方法では整合ピークにおける識別確度がどの程度あれば良いのかを設定する必要が生じてしまう問題点がある。ガス濃度が変われば匂いが違うと錯覚するように、生体においてもガス濃度に関しては対応できない面がある。このように複数種の匂い分子に対して応答するようなガスセンサ（匂い受容器）を使用する場合は、ガス濃度により識別の確かさが影響を受けてしまう。もしも、特定の分子の識別だけを行う場合は、個別にガスセンサを開発すればよいが、ロボットなどの嗅覚系を構成するためには、ガス濃度に対応できる匂いセンシング技術の確立が必要になる。

以上に本研究で開発した匂いセンシングシステムに関して述べてきた。これまで、家庭用ガス警報器・電子レンジ・エアコン・ストーブ・自動車エンジン制御用酸素ガスセンサ等¹³⁾において、個別にガスセンサは工夫して使用されてきた。より一般的な用途に対応できる可能性を秘めている匂いセンシングシステムは、匂いセンシングと関連する多くの分野の技術と融合しあうことにより、そう遠くはない未来に匂いセンシングの分野において Breakthrough をもたらすのではないかと期待している。

【 謝 辞 】

本研究を遂行するにあたり常にご指導くださいました成田紘一教授、中村善一助教授に深く感謝いたします。また、実験装置の開発等において多くの助言、貴重な情報を提供して頂きました電気工学科 京兼純教授、山本善啓技官に深く感謝いたします。

最後に電気工学科、専攻科 電子情報工学専攻での7年間に渡り多方面でお世話になりました電気工学科・情報工学科の先生方を始め、一般教科の先生方、研究の際に多くの支援、数々の助言を頂きました専攻科 電子情報工学専攻1年生の岡崎哲三君、電気工学科5年生の栗野篤史君、成田・中村研究室の皆様感謝の意を表したいと思います。

【 参考文献 】

- 1) 中本高道・森泉豊栄：ニューラルネットワークを用いた匂いセンサー
応用物理 第58巻 第7号 1989
- 2) 石淵久生・田中英夫・岩本則子：多次元区間データの判別分析と匂いセンサへの応用
計測自動制御学会論文集 vol.25 No.6 1989 pp.630-637
- 3) 森泉豊栄：匂いセンシングシステム
電気化学学会論文誌 No.3 1994
- 4) 高木貞敬・渋谷達明：匂いの科学
朝倉書店 1989
- 5) 長谷川香料(株)：においの化学
裳華房 1988
- 6) 都甲潔・山藤馨：美味しさを測る
ブルーバックス 講談社
- 7) 大村平：多変量解析のはなし
日科技連
- 8) 舟久保登：パターン認識
共立出版株式会社
- 9) 甘利俊一・向殿政男：ニューロとファジィ
培風館

- 10) 工業調査会編集部編 工業調査会：センサ活用技術
中谷吉彦：ガスセンサ pp.276-277
山添・三浦：半導体ガスセンサ pp.396-401
- 11) 自動化技術編集部編 工業調査会：やさしいセンサ技術
ガスセンサ利用上の要点 pp.58-60
ガスセンサ：pp.238-250
- 12) 清山哲郎：ガスセンサー気体成分の電気化学的識別
電気化学学会論文誌 解説 No.1 1982
- 13) 五百蔵弘典・沖野一雄：金属酸化物半導体ガスセンサの応用
センサ技術 vol.3 No.7 7月号 1983
- 14) 外村正一郎・松岡継文・山本直登・坪村宏：金属－半導体接合型ガスセンサーとその作動機構
日本化学学会誌 No.10 1980 pp.1585-1590
- 15) 磯部満夫：ガスセンサの諸特性と利用のポイント
メカトロニクス増刊号 1月号 1981
- 16) 島田鉄也：ニオイセンサに関する研究
平成5年度電気工学科卒業論文 1994
- 17) Akira IKEGAMI, Hideo ARIMA, Shoichi IWANAGA AND Masayoshi KANEYASU
: THICK FILM SENSORS AND THEIR INTEGRATION
4th European Hybrid Microelectronics Conference 1983, Copenhagen
- 18) Ryosuke KONISHI, Osamu TAKETOSHI, Kenji KOBAYASHI and Hiroshi SASAKURA
: DISCRIMINATION OF GASES BY USING FUZZY REASONING
Proceedings of the international Conference on Fuzzy Logic &
Neural Networks pp.81-86

【業績】

- [1] “ SnO_2 系半導体ガスセンサを用いたパターン認識型匂いセンシングシステムの開発”, 桐島, 平成6年電気関係学会関西支部連合大会 G10-27, 1994年11月

【 付 録 】

1. 半導体ガスセンサ関連資料.....	付-1
(1) アルコールガスセンサ (T G S 8 2 3)	付-1
(2) アンモニアガスセンサ (T G S 8 2 4)	付-3
(3) 硫化水素ガスセンサ (T G S 8 2 5)	付-5
(4) メタンガスセンサ (T G S 8 1 3)	付-7
2. 半導体ガスセンサ基本測定回路図.....	付-8
3. システムブロック図.....	付-9
4. 匂いセンシングシステムプログラムリスト	
4. 1 匂い実験関連プログラムリスト	
(1) DEF. H : 構造体宣言・マクロ宣言.....	付-10
(2) QUO. H : グローバル変数宣言.....	付-12
(3) SAMPLE. H : A/D変換・リレー制御用ヘッダ.....	付-13
(4) MAIN. C : メインプログラム.....	付-18
(5) WINDOW. C : ウィンドウ処理用関数群.....	付-21
(6) AD. C : A/D変換用関数群.....	付-35
(7) RELAY. C : リレー制御用関数群.....	付-40
(8) GRAPH. C : グラフ描画用関数群.....	付-43
(9) FILE. C : ファイル処理用関数群.....	付-56
(10) GAS. C : 匂い測定用関数群.....	付-59
4. 2 匂い識別関連プログラムリスト	
(1) DEF. H : 構造体宣言・マクロ宣言.....	付-72
(2) QUO. H : グローバル変数宣言.....	付-73
(3) MAIN. C : メインプログラム.....	付-74
(4) FUNC. C : 識別関連処理用(相互作用可変法を含む)関数群.....	付-85
(5) NN. C : 多層ニューラルネットワーク処理用関数群.....	付-96
(6) FILE. C : ファイル処理用関数群.....	付-103
(7) MEM. C : メモリ割り当て用ファイル.....	付-108
(8) MEM2. C : メモリ割り当て用ファイル.....	付-109

3. 回路条件

表1にTGS822の安全動作領域を示します。実用回路においては、 V_c 、 V_m および P_s の値はこの表に示された範囲を越えてはいけません。この範囲内であれば V_c 、 R_L の値は自由に定めることができます。

表1. TGS822の安全動作領域

項目	記号	規定値	備考
ヒータ電圧	V_m	$5.0 \pm 0.2V$	AC or DC
回路電圧	V_c	MAX. 24V	AC or DC $P_s \leq 15mW$ の範囲内
負荷抵抗	R_L	可変	$P_s \leq 15mW$ の範囲内
センサ消費電力	P_s	MAX. 15mW	$P_s = \frac{V_c \cdot R_s}{(R_s + R_L)}$

4. 試験条件および特性仕様

TGS822の特性は表2の標準試験条件において各製品が表3に示す特性仕様を満足しています。

表2. 標準試験条件

雰囲気条件	$20 \pm 2^\circ C$ $65 \pm 5\% RH$ の清浄大気中
回路条件	$V_c = 10.0 \pm 0.1V$ (AC or DC) $V_m = 5.00 \pm 0.05V$ (AC or DC) $R_L = 3.9k\Omega \pm 1\%$
試験前エージング	7日以上(上記回路条件にて)

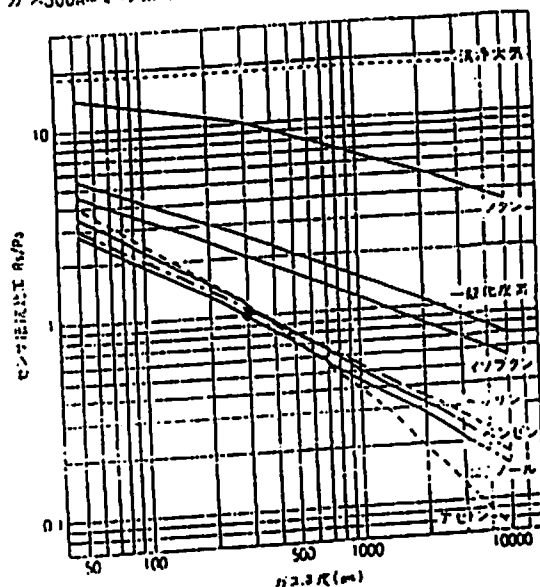
表3. TGS822の特性仕様

項目	条件	特性仕様
1. 感ガス特性		
1) センサ抵抗比(R_s)	エタノール 300ppm中	$1.0 - 10.0 \times 10$
2) ガス中 センサ抵抗比	$\frac{R_s(\text{エタノール} 300ppm)}{R_s(\text{エタノール} 50ppm)}$	0.4 ± 0.1
2. 電力的特性		
1) ヒータ抵抗値(R_h)		$38 \pm 3\Omega$
2) ヒータ消費電力(P_m)		$\approx 660mW$
3. 機械的強度特性		
1) 耐振性	振動試験条件 周波数: 1000cpm 振幅: 4mm(垂直方向) 時間: 1時間	上記1.2に 変化のないこと
2) 耐衝撃性	衝撃試験条件 加速度: 100G テスト回数: 5回	上記1.2に 変化のないこと

5. 標準的特性データ

5-1. 各種ガス感度特性

図5はTGS822の各種ガスに対する標準的な感度特性を示すものです。この図は各ガスの各濃度におけるセンサ抵抗値と、エタノールガス300ppmでのセンサ抵抗値との比であらわされています。

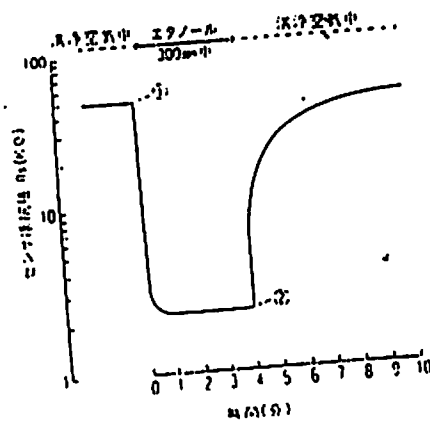


■図5. ガス感度特性

(注) R_s : エタノール300ppmを含む空気中でのセンサ抵抗値
 R_s : 各ガス、各濃度でのセンサ抵抗値

5-2. レスポンス特性

図6はTGS822を清浄大気中からエタノールガス300ppm中へ入れた時(図中①)、逆に同ガス中より清浄大気中へ戻した時(図中②)の標準的なセンサのレスポンス特性を示しています。エタノールガスに対する良好なレスポンス特性が得られています。



■図6. レスポンス特性

アンモニアガス検出用 FIGARO GAS SENSOR

TGS 824

TGS 824は、アンモニアの漏洩検出に適したセンサとして開発された、酸化スズ (SnO_2) 半導体ガスセンサです。このセンサは、雰囲気中の検知ガス濃度の増加により、その電気伝導度が大きく増加しますので、簡単な回路構成で、検知ガスの存在を検出することができます。また、ハウジングのベースにアルミナセラミックを用いており高温雰囲気中での使用が可能です。

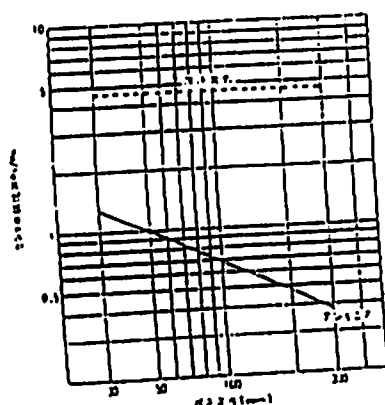
特 徴

- ・アンモニアガスを 30ppm から検出可能
- ・再現性、安定性に優れる
- ・検知ガスによる電導度変化が大きく、簡単な回路構成でガス検出が可能
- ・セラミックベースの採用により、高温雰囲気での使用が可能



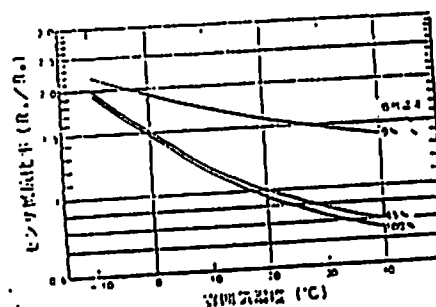
感度特性の代表例

- ・測定は標準試験条件(仕様の項参照)による
- R_s : 各ガス、各濃度でのセンサ抵抗値
- R_a : アンモニア 50ppm を含む空気中でのセンサ抵抗値

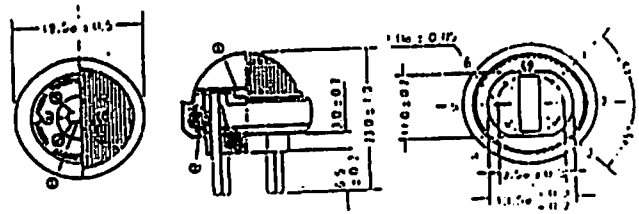


温度依存性

- R_s : アンモニア 50ppm を含む各温度の空気中におけるセンサ抵抗値
- R_a : アンモニア 50ppm を含む 20°C 65% の空気中におけるセンサ抵抗値

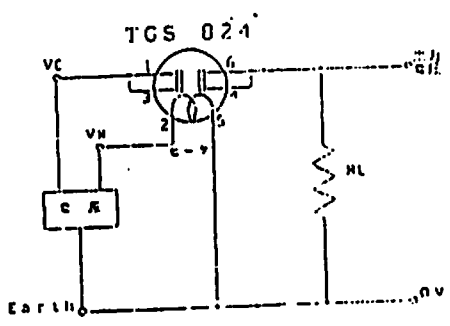


構造及び寸法



- ① 感ガス体：SnO₂セラミック半導体
ヒータを内蔵したアルミナセラミック
チューブの表面に形成されています。
- ② ベース：アルミナセラミック製
- ③ 防塵金網：100メッシュ SUS316 二重

基本測定回路



動作条件

TGS 824 の特性を安定に保って御使用願うため、次の回路条件をお守りください。

項目	記号	規格値	備考
ヒータ電圧	V _H	5.0 ± 0.2V	A Cor D C
回路電圧	V _c	MAX. 24V	A Cor D C * P _s ≤ 15mW の 範囲内
負荷抵抗	R _L	可変	* P _s ≤ 15mW の 範囲内

センサ記号に添付した数字は上記構造図中に示されているピンNoを示します。

センサ抵抗値(R_s)は、出力電圧(V_{out})から次の式により算出されます。

$$R_s = \left(\frac{V_c}{V_{out}} - 1 \right) \times R_L$$

* P_s は、感ガス体を流れる電流により消費される電力で、次の式により算出されます。

$$P_s = \frac{V_c^2 R_s}{(R_s + R_L)^2}$$

電気特性

項目	条件	規格値
センサ抵抗値 (R _s)	アンモニア 50ppm/空気中	5K ~ 20K Ω
ガス濃度-センサ抵抗変化比	R _s (7ppm) / R _s (50ppm) R _s (7ppm) / R _s (50ppm)	0.5 ± 0.05
ヒータ抵抗値	室温	59.0 ± 4.0 Ω
ヒータ消費電力	V _H = 5.0V	425 ± 30mW

※ 以下の値は標準条件により得られる値です。
 ベースガス：20 ± 2℃ 65 ± 5% R.H. の空気中
 試験条件：V_c = 10.0 ± 0.1V (AC or DC)
 R_L = 3.9K ± 1%

※ 以下の値は標準条件により得られる値です。
 試験条件：V_c = 10.0 ± 0.1V (AC or DC)
 R_L = 3.9K ± 1%

— 硫化水素ガス検出用 —
FIGARO GAS SENSOR

TGS 825

TGS 825は、硫化水素の漏洩検出に適したセンサとして開発された、酸化スズ (SnO_2) 半導体ガスセンサです。このセンサは、雰囲気中の検知ガス濃度の増加により、その電気伝導度が大きく増加しますので、簡単な回路構成で、検知ガスの存在を検出することができます。また、ハウジングのベースにアルミナセラミックを用いており高温雰囲気中での使用が可能です。

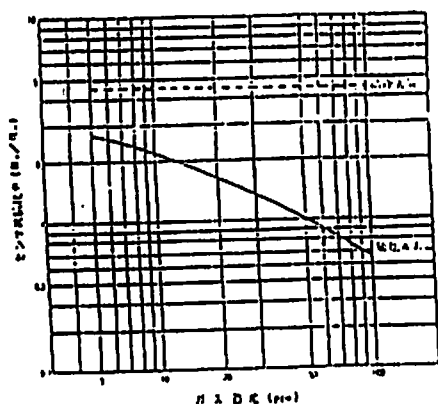
特 徴

- ・ 硫化水素ガスを 5 ppm から検出可能
- ・ 再現性、安定性に優れる
- ・ 検知ガスによる電導度変化が大きく、簡単な回路構成でガス検出が可能
- ・ セラミックベースの採用により、高温雰囲気中での使用が可能



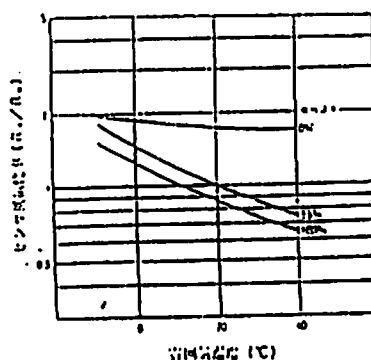
感度特性の代表例

- ・ 測定は標準試験条件(仕様書の参照)による
- R_s : 各ガス、各濃度でのセンサ抵抗値
- R_o : 硫化水素 50 ppm を含む空気中でのセンサ抵抗値

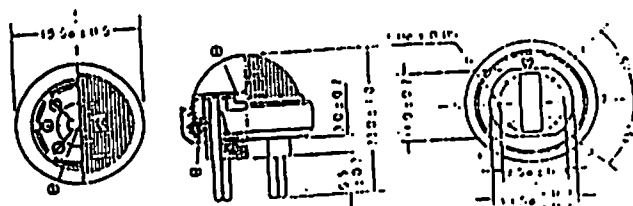


温湿度依存性

- R_s : 硫化水素 50 ppm を含む各温湿度の空気中におけるセンサ抵抗値
- R_o : 硫化水素 50 ppm を含む 20°C 65% の空気中におけるセンサ抵抗値

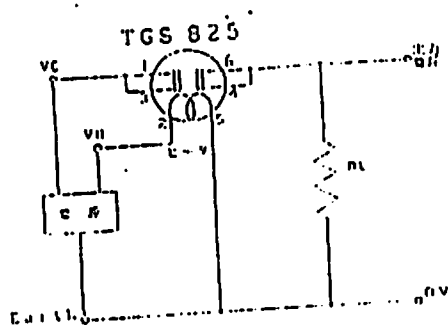


構造及び寸法



- ①感ガス体: SnO₂セラミック半導体
ヒータを内蔵したアルミナセラミック
チューブの表面に形成されています。
- ②ベース: アルミナセラミック製
- ③防塵金網: 100メッシュ SUS 316 二重

基本測定回路



センサ記号に添付した数字は上記構造図中に示されているピンNoを示します。

センサ抵抗値(R_s)は、出力電圧(V_{out})から次の式により算出されます。

$$R_s = \left(\frac{V_c}{V_{out}} - 1 \right) \times R_L$$

動作条件

TGS 825 の特性を安定に保って御使用頂くため、次の回路条件をお守りください。

項目	記号	規格値	備考
ヒータ電圧	V_H	$5.0 \pm 0.2V$	A Cor D C
回路電圧	V_c	MAX. 24V	A Cor D C * $P_s \leq 15mW$ の範囲内
負荷抵抗	R_L	可変	* $P_s \leq 15mW$ の範囲内

* P_s は、感ガス体を流れる電流により消費される電力で、次の式により算出されます。

$$P_s = \frac{V_c^2 R_s}{(R_s + R_L)^2}$$

電気特性

項目	条件	規格値
センサ抵抗値 (R_s)	硫化水素 50ppm/空気中	3K ~ 30K Ω
ガス濃度-センサ抵抗変化比	R_s (硫化水素 50ppm/空気中) R_s (硫化水素 10ppm/空気中)	0.45 ± 0.15
ヒータ抵抗値	室温	$38.0 \pm 3.0 \Omega$
ヒータ消費電力	$V_H = 5.00V$	$660 \pm 55mW$

①(2)は次の条件により算出されています。

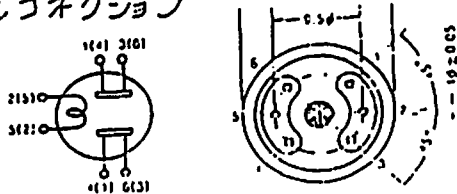
ベースガス: $20 \pm 2^\circ C$ $65 \pm 5\% R.H.$ の空気
 動作電圧: $V_c = 10.0 \pm 0.1V$ (AC or DC)
 $R_L = 3.9 \pm 0.1\%$

②(4)は、 $V_H = 5.00 \pm 0.05V$ (AC or DC)
 $V_c = 5.00 \pm 0.05V$ (AC or DC)

フイカロガスセンサー #812及び#813

1. フイカロガスセンサー TGS #812及び#813は 回路電圧24V以下、ヒーター電圧5Vで使用可能なセンサーです。水素ガス、一酸化炭素ガス、アルコールガス、メタンガス、ブタンガス、プロパンガス、LPG天然ガス、都市ガス、有機溶剤蒸気、第一級可燃性ガスを検知することができます。

2. ピンコネクション



3. 特徴

- (1) #812、#813共に汎用センサーとして一般可燃性ガスの検出に用いることができます。
- (2) #812は#813に比べてアルコールガス、一酸化炭素ガス、メタンガス、ブタンガス、プロパンガス等の検知感度が向上しています。
- (3) #813は#812に比べてメタンガス、ブタンガス、プロパンガス等の検知感度が向上しています。
- (4) #813は#812に比べてメタンガス、ブタンガス、プロパンガス等の検知感度が向上しています。
- (5) #812、#813共に温度湿度依存性はきわめて低く耐湿です。

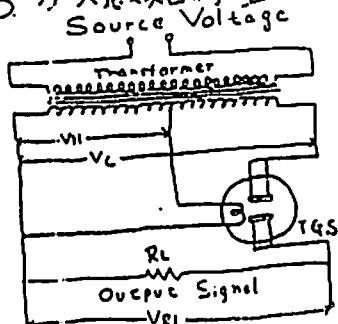
4. 用途

大気汚染検出器、ガス検知警報器、毒性ガス検知器、アルコール検知器、自動換気扇、火災報知器、ブタンガス検知器、工業用各種ガス検知器、一酸化炭素検出器、その他応用に応用できます。

5. 電気的仕様

	#812	#813
ヒーター電圧	5V ± 0.2V (AC or DC)	
回路電圧	24V max	
ヒーター消費電力	49650-W	49830-W
ヒーター抵抗	50Ω ± 3Ω	50Ω ± 3Ω
ヒーター抵抗	1k~10kΩ	5k~15kΩ
ヒーター抵抗	1000ppm/Air	1000ppm/Air

6. ガス感度特性

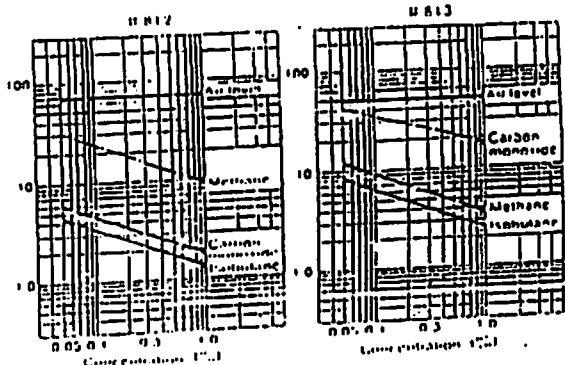


VR: ヒーター電圧 5.0V
VC: 回路電圧 10V
RL: 負荷抵抗 4kΩ

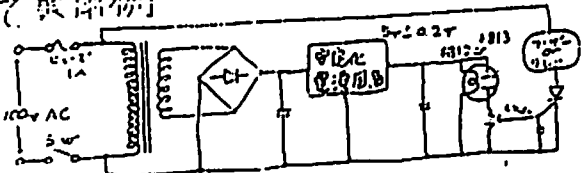
TGS感度特性はVRにて調整し
TGSの抵抗(RL)は
式で変換します。

$$R_s = R_L \left(\frac{V_C}{V_{RL}} - 1 \right)$$

7. 測定回路及びガス感度特性



7. 実回路例



この図は最もシンプルなマイコン回路例です。但しヒーターはヒーター電圧のよいものを使用下さい。感度はVRで調節します。尚、多量生産用でロバストな回路例もあり得るが試作には適しませんので割愛します。

8.

現在生産されているものはTGSには#812、#813及び#1109があります。これらのTGSはどれも通電後1~2分のウォームアップタイムが必要です。ですから上のような回路例の回路で通電すると約1~2分間はTGSが導通状態になりアースから取り出しに感度リレーが入ったままの状態になります。しかしそのまま放置しておくと1~2分後には正常状態に入りますのでアースと取り出しをオフにします。ウォームアップタイムを短縮する必要がある場合やウォームアップタイムをオフ状態に保ちたい場合は適当な回路があります。多少複雑になりますが、その場合は場合や大量生産用には御使用の場合、もしくは他社的にTGSを御使用の場合は、用途、会社名、所属、明記の上、別途資料を御請求下さい。

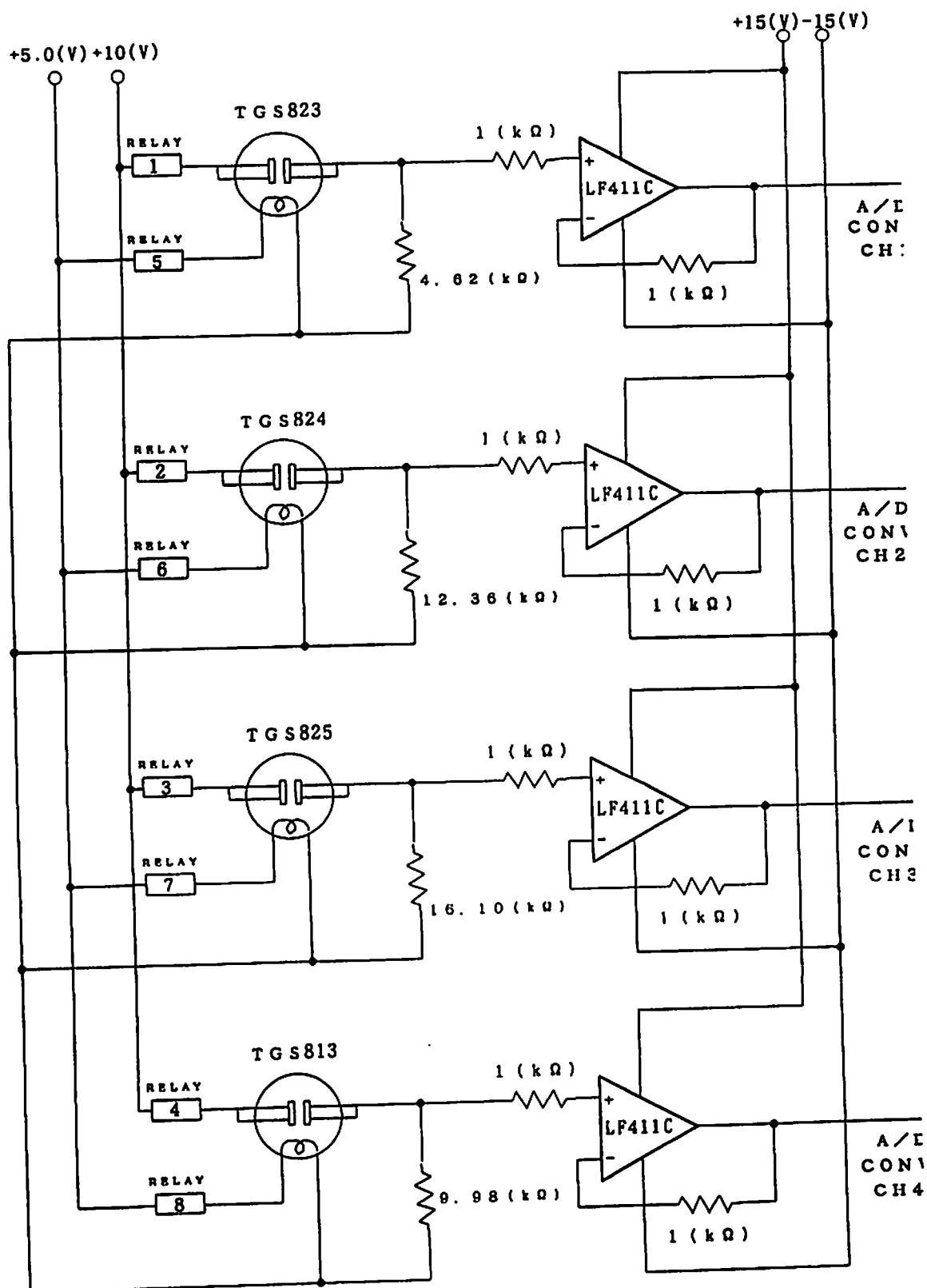


図103. 半導体ガスセンサ基本測定回路図

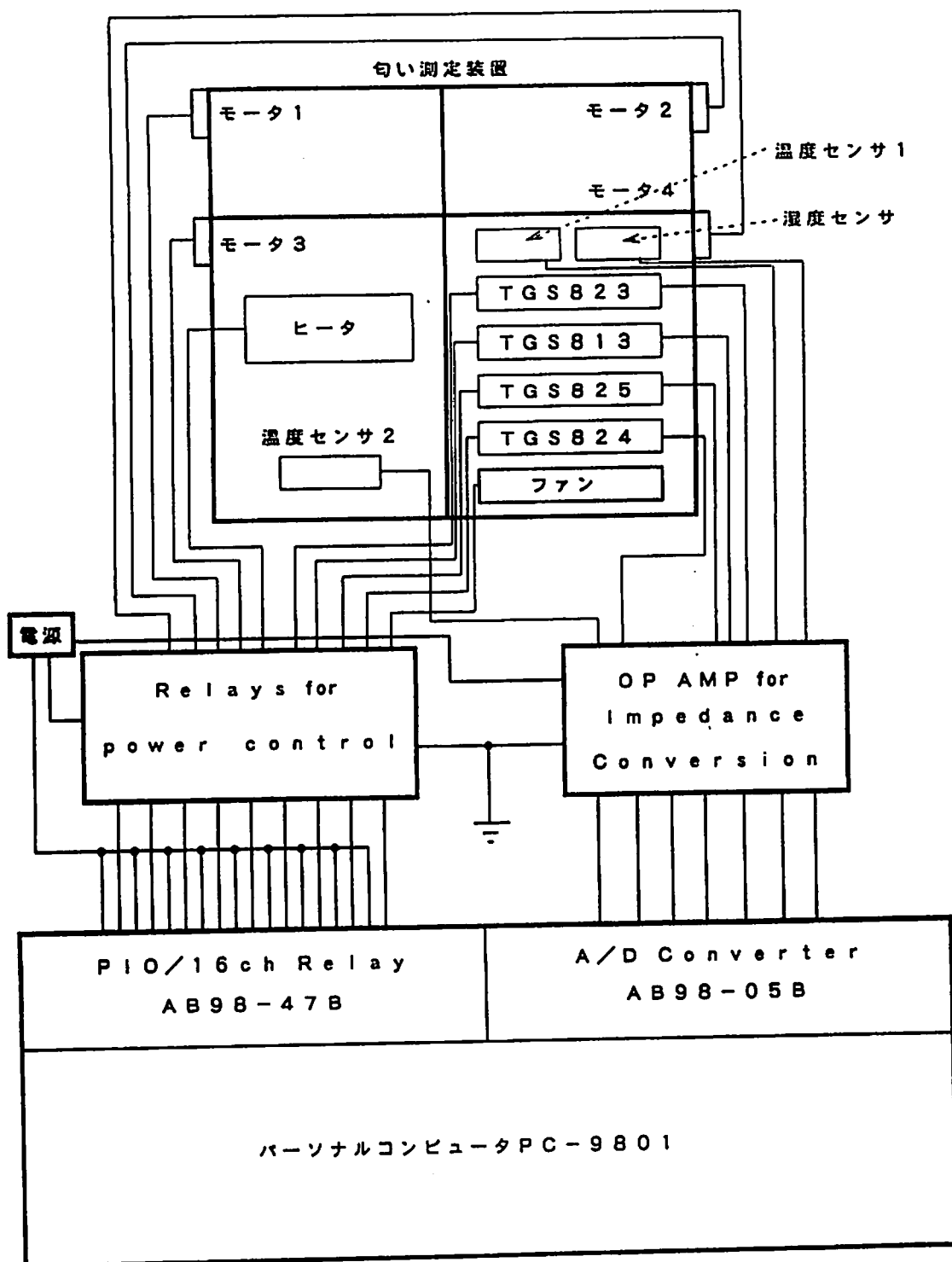


図104. システムブロック図

```

#define DATA_WD 76 /* 入力データ最大幅 */

#define MAX_PAGE 80 /* オープン出来るウィンドウの枚数 */
#define MERGIN 7 /* データNO表示用 */
#define NUM_MERGIN 5 /* データNOを表示しないとき */
#define WD_OPEN 0 /* ウィンドウ関数使用時のdefault値 */
#define WD_CLOSE 0,1,0,0
#define WD_CLOSEV 1
#define WD_REMAIN 2
#define FW_OPEN 3
#define FW_REMAIN 4

#define FN_MAX 200 /* 最大ファイル数(変更時let's.envに注意!) */
#define WINDOW_MAX 18 /* 登録ウィンドウの数 */

#define FN_SIZE sizeof( struct EXP_DATA_FNAME )
#define WINDOW_SIZE sizeof( WINDOW )

struct EXP_DATA_FNAME {
    char fname[DATA_WD]; /* ファイル名 */
    int temp,humid; /* 温度・湿度 */
    char st[80],ed[80]; /* 開始終了日時 */
    double t1,t2,t3,t4; /* 32byte Spare */
};

typedef struct WD {
    int x; /* ウィンドウ表示座標 */
    int y;
    int wx; /* ウィンドウの大きさ */
    int wy;
    int col1; /* ウィンドウ フレーム表示色 */
    int col2; /* ウィンドウ タイトルの色 */
    int col3; /* ウィンドウ 表示文字列の色 */
    int l; /* ウィンドウ カーソル座標(変数) (from 0) */
    char stitle; /* ウィンドウ タイトル */
    int inp_node; /* 0..内容更新可能タイプ
                  /* 1..表示のみのウィンドウ
                  /* 2..内容更新不能タイプ(選択のみ)
    int cmx; /* カーソル上限(この値には、ならない)(from 0)
    int cmi; /* カーソル下限(この値もで、なれる) (from 0)
    int button; /* 0..ESC機能をサポート(ボタンあり)
                /* 1..上下左右ESC機能をサポート
                /* 2..メッセージのみ表示する(ボタンないがESC可能)
    int cursor_on; /* カーソルを表示する? 1..YES 0..NO
    int lineno_on; /* 0..ラインNOを表示しない
                  /* 1..ラインNOを表示する
                  /* ディレクトリ表示の際は変数 ul
                  /* 用の退避変数
    int inp_line; /* インプット行の指定(from 0)
                  /* -1 の時全ての行で入力可能
                  /* ディレクトリ表示の際は横方向に
                  /* 表示するファイル名の数
} WINDOW;

struct TGS_INFO {
    char tname;
    double r1;
    unsigned int col; /* グラフ上の色
    char cn[3]; /* 色の漢字名
};

#define OPENING 0 /* used
#define FUNC_MENU 1 /* for function menu
#define FILE_DISP 2 /* file menu display
#define HD_FILE 3 /* used
#define MAKESURE 4 /* for yes / no selector
#define MESDISP 5 /* for message display function
#define GET_FNAME 6 /* used
#define AD_DISP 7 /* for displaying A/D input
#define RELAY_CON 8 /* for relay control
#define ST_DISP 9 /* for control state table
#define GP_MENU 10 /* for graph menu
#define GP_TYPE 11 /* for output type menu
#define GP_STEP 12 /* select drawing range
#define GP_DISP 13 /* select drawing data
#define JG_MENU 14 /* for Snell Judge menu
#define JG_DISP 15 /* display judgement result
#define OMOMI 16 /* 置み設定用ウィンドウ

```

DEF.H

```

#define SCR_DISP      17 /* スクロールメッセージウィンドウ */

#define AD_START      0X00D2 /* ADコンバータスタートI/Oアドレス */
#define AD_SELECT     0X00D0 /* ADコンバータセレクトI/Oアドレス */
#define AD_HIGH       0X00D2 /* ADコンバータ上位I/Oアドレス */
#define AD_LOW        0X00D0 /* ADコンバータ下位I/Oアドレス */

#define RELAY_0        0X00D4 /* リレー入出力I/Oアドレスポート0 */
#define RELAY_1        0X00D5 /* リレー入出力I/Oアドレスポート1 */
#define RELAY         0X00D4 /* リレー入出力I/O先頭アドレス */

/*
/*      測定手順格納用構造体
/*
struct PROC {
    int time; /* 所要時間(sec) */
    int exiv; /* 測定間隔(sec) */
    int sb; /* 安定化ルーチンフラグ */
    int vh; /* ヒータ電圧ON/OFF */
    int vf; /* 換気扇 ON/OFF */
    int iv; /* 換気扇ドライブ間隔(sec) */
    int sw; /* ガスセンサヒータON/OFF */
    int hc; /* ヒータコントロールフラグ */
    char *title; /* プロセス名 */
    char *mes1; /* 画面表示メッセージ 1(wait) */
    char *mes2; /* 画面表示メッセージ 2 */
    char *mes3;

};

#define MOSAIC_X 1 /* モザイク法X幅 */
#define MOSAIC_Y 1120 /* モザイク法Y幅 */
#define LGTP 6 /* お酒の種類の数 */

struct MOSAIC_TABLE {
    /* モザイク法用データ構造体 */
    /* 判別テーブル S Width Y Width X */
    unsigned int tbl[4][MOSAIC_Y][MOSAIC_X];
};

struct POINT {
    float ratio;
    float h;
    float Upara; /* 上方向傾斜係数 */
    float Dpara; /* 下方向傾斜係数 */
    float Atn; /* 減衰係数 */
    float Rst; /* 抑制係数 */
};

/*
/*      モザイク深さ-->30(max) 測定手順の数-->10
/*      モザイク深さについては、ディスク領域を記録媒体にするので特に気にしない
/*      測定手順に変更があった場合、それに対応する場所の変更が必要
/*
struct SMELL {
    int interval,n;
};

```



```
/* */
/* ボードアドレス設定値を格納する変数 */
/* */
extern unsigned int select_prt;
extern unsigned int low_data_prt;
extern unsigned int start_prt;
extern unsigned int high_data_prt;

extern unsigned int prt_0;
extern unsigned int prt_1;

/* ウィンドウ関係 */
extern unsigned int _stklen;
extern struct EXP_DATA_FNAME fn(FN_MAX);
extern WINDOW WD_info(16);
extern char DIA();
extern int cfn,MS_RB,MS_LB,MS_X,MS_Y;
extern int sblink,txsense,nxsense,nysense;

/* 測定データ */
extern double V[7][1000],PD[7][1000];
extern struct TGS_INFO tgs[7];
extern struct PROC Proc(22);
extern struct MOSAIC_TABLE Mt( LOTP );
extern struct SMELL SM[14];
```

SAMPLE.H

```

/* ..... */
/* ..... */
/* :: A B 9 8 B シリーズ : サンプルプログラム用 :: */
/* :: ヘッダーファイル :: */
/* ..... */
/* ..... */

```

```

/* ..... */
/* hardware interrupt vector number define */
/* ..... */
#define TIMER_INT 0x08 /* timer */
#define KB_INT 0x09 /* キーボード */
#define CRTV_INT 0x0a /* V-SYNC */
#define INT_EXT_BUS0 0x0b /* 拡張バス INT0 */
#define RS232C_INT 0x0c /* RS232C */
#define INT_EXT_BUS1 0x0d /* 拡張バス INT1 */
#define INT_EXT_BUS2 0x0e /* 拡張バス INT2 */
#define PRN_INT 0x10 /* セントロニクス printer */
#define INT_EXT_BUS3 0x11 /* HD */
#define INT_EXT_BUS41 0x12 /* 640k byte FD */
#define INT_EXT_BUS42 0x13 /* 1M byte FD */
#define INT_EXT_BUS5 0x14 /* 拡張バス INT5 */
#define INT_EXT_BUS6 0x15 /* 拡張バス INT6 */
#define CO_PROBER_INT 0x16 /* 8087 */

```

```

/* flag control code define */
#define ON 1
#define OFF NULL
#define CR 0x0d
#define LF 0x0a
#define INPUT_MAX_LENGTH 128
#define CTRL_C_DATA 3
#define function_key_status 0x1b

```

```

/* console character attribute code */
#define normal 0
#define high_light 1
#define b_line 2
#define under_score 4
#define brink 5
#define rvs 7
#define secret 8

```

```

#define C_BLACK 0
#define C_BLUE 1
#define C_RED 2
#define C_PURPLE 3
#define C_GREEN 4
#define C_MIZUIRO 5
#define C_YELLOW 6
#define C_WHITE 7

```

```

/* color code */
#define BLUE 34
#define PURPLE 35
#define GREEN 32
#define YELLOW 33
#define RED 31
#define MIZUIRO 36
#define WHITE 37
#define BLACK 30

```

```

/* reverse color code */
#define BLUE_RVS 44
#define PURPLE_RVS 45
#define GREEN_RVS 42
#define YELLOW_RVS 43
#define RED_RVS 41
#define MIZUIRO_RVS 46
#define WHITE_RVS 47
#define BLACK_RVS 40

```

SAMPLE.H

```

/*          */
/* CRT control function define */
/*          */
#define locate_set(x,y)    printf("Y033(%d;%dH",y,x)
#define cls_set( )        printf("Y033(2J")
#define color_set(x)       printf("Y033(%dm",x)
#define cursor_clr( )      printf("Y033(>5h")
#define cursor_dsp( )      printf("Y033(>5l")
#define cursor_dwn( )      printf("Y033(1D")
#define one_line_clr( )    printf("Y033(0k")

/*          */
/* error & message code data define */
/*          */
enum    err_status {
    input_data_error = 1, file_not_found,
    same_file_found, file_length_err, file_fmt_err, file_read_err,
    file_write_err, no_open_file, memory_allocate_err,
    memory_read_err, disk_full, disk_acs_err, dos_error, file_again_input,
    timer_loop_calcu, timer_loop_result, wait_msg, initial_set_error,
    start_ok, env_take_ng, usag_msg, parameter_err, verify_err,
    file_size_over, abort_end, ignore_err, sample_program_end,
    ctrl_c_end
};

enum    bord_select_no {
    ab9802b = ctrl_c_end + 1, ab9804b = ab9802b + 2, ab9805b, ab9806b,
    ab9808b = ab9806b + 2, ab9809b, ab9810b, ab9816b = ab9810b + 6,
    ab9819b = ab9816b + 3, ab9821b = ab9819b + 2, ab9822b, ab9823b,
    ab9839b = ab9823b + 16
};

#define    xalloc(t,n) ((t*))halloc((long)n,(unsigned)sizeof(t))
#define    MAX( A, B ) (( A ) > ( B ) ? ( A ) : ( B ))
#define    MIN( A, B ) (( A ) < ( B ) ? ( A ) : ( B ))

/*          */
/* input data attribute area define */
/*          */
struct    input_item {
    unsigned long    input_min; /* 許容入力最小値 */
    unsigned long    input_max; /* 許容入力最大値 */
    int zero_f; /* 0入力許容ステータス */
    int special_char_item; /* 記号入力許容数 (最大16個) */
    char    special_strings[16]; /* 許容入力記号データ */
    int input_length; /* 許容入力長 */
    int radix; /* 入力基数 */
    int msg_dsp_x; /* メッセージ表示カラム */
    int msg_dsp_y; /* メッセージ表示行 */
    int erase_mode; /* 0 = なにもしない
                    /* 1 = 表示行以降を消去
    };

/*          */
/* r/w length data display information area define */
/*          */
struct    rw_dsp_prm {
    int dsp_x_init;
    int dsp_y_init;
    int dsp_x;
    int dsp_y;
    unsigned long    load_length;
    int dsp_cnt;
    int load_cnt;
};

/*          */
/* unsigned long byte data access */

```

SAMPLE.H

```

/* use structure(union) data area define */
/*
*/
struct byte_tag {
    unsigned char byte_0; /* LSB */
    unsigned char byte_1;
    unsigned char byte_2;
    unsigned char byte_3; /* MSB */
};

struct word_tag {
    unsigned int lower;
    unsigned int higher;
};

union byte_4_acs {
    struct byte_tag b_acs;
    struct word_tag w_acs;
    unsigned char byte_array[3];
    unsigned long long_byte;
};

/*
*/
/* byte data access use structure define */
/*
*/
struct byte_acs_0 {
    unsigned char byte_0;
    unsigned char byte_1;
    unsigned char byte_2;
    unsigned char byte_3;
};

struct byte_acs_1 {
    unsigned char byte_4;
    unsigned char byte_5;
    unsigned char byte_6;
    unsigned char byte_7;
};

struct byte_acs {
    struct byte_acs_0 long_byte_0;
    struct byte_acs_0 long_byte_1;
};

/*
*/
/* double byte access utility define */
/*
*/
union double_byte_acs {
    struct byte_acs byte_0_acs;
    unsigned char byte_array[7];
    unsigned int wrd_acs_0;
    unsigned int wrd_acs_1;
    unsigned int wrd_acs_2;
    unsigned int wrd_acs_3;
    unsigned long long_acs_0;
    unsigned long long_acs_1;
};

/*
*/
/* ON/OFF出力操作を指示するためのパラメータ */
/*
*/
struct puls_out_infm {
    unsigned int out_data; /* 出力データ */
    unsigned int out_count; /* 出力回数 */
    unsigned int on_time; /* 指定した信号(ビット)をONさせる時間 */
    unsigned int off_time; /* 指定した信号(ビット)をOFFさせる時間 */
};

/*
*/
/* 8255の設定状態を格納する構造体 */
/*
*/

```

SAMPLE.H

```

struct port_direction_set {
    int a_port_direction;
    int b_port_direction;
    int c_hport_direction;
    int c_lport_direction;
};

/* A/D変換指定データ格納用構造体 */
struct ad_infm {
    int chanel_no;
    unsigned int read_cnt;
    double scale_max;
    double scale_min;
};

/* D/A変換指定データ格納用構造体 */
struct da_infm {
    int chanel_no;
    unsigned int da_digital_data;
    double da_analog_data;
    double scale_data;
    double scale_max;
    double scale_min;
};

/*
 */
/* SMPL02B.C function prototype define */
/*
 */
void write_after_wait(void);
void init_rs232c(unsigned int,int,int);
int one_byte_rcv(unsigned int,int *);
int one_byte_snd(unsigned int,unsigned char);
unsigned int status_ptr_get(int);
int chanel_one_byte_rcv(int,int *);
int chanel_one_byte_snd(int,unsigned char);
int status_ptr_read(int);
int chanel_comand_status_read(int);
void chanel_comand_status_write(int,int);
void chanel_cmd_data_ptr_data_take(int,int *,int *);
void cmd_data_out_and_save(int,int,int);
void rts_set(int,int);
void dtr_set(int,int);
void txen_set(int,int);
void rxen_set(int,int);
void er_reset(int);
void rs_reset(int);

/*
 */
/* SMPL04B.C function prototype define */
/*
 */
unsigned int port_adres_get(int,unsigned int *);
unsigned int ppi_data_input(int);
int ppi_data_out(int,int);
int ppi_bit_set(int,int,int);
unsigned int ppi_bit_check(int,int);
int ppi_byte_puls_out(int,struct puls_out_infm *,int);
int ppi_bit_puls_out(int,struct puls_out_infm *,int);

/*
 */
/* SMPL05B.C function prototype define */
/*
 */
int ad_read(int,unsigned int *);
int ad_count_read(int,unsigned int,unsigned int *);
double analog_scale_convert(int,unsigned int,double,double);
double voltage_convert(int,unsigned int);

/*
 */
/* SMPL06B.C function prototype define */
/*
 */
int da_out(int,unsigned int);
double deg_out_convert(int,unsigned int);
unsigned int analog_convert_out(int,double);
unsigned int digital_scale_convert(int,double,double,double);

```

SAMPLE.H

```

/*
/*
/* SMPL00B.C function prototype define */
/*
/*
unsigned long bit_status_set(int,unsigned long,int);
int chanel_output(int,int);
int chanel_byte_out(int,int);
void chanel_all_out(unsigned long);
int chanel_byte_puls_out(int,struct puls_out_infm *,int);
int chanel_bit_puls_out(int,struct puls_out_infm *,int);

/*
/*
/* SMPL09B.C function prototype define */
/*
/*
int bit_status_check(unsigned long,int);
int chanel_input_check(int);
int chanel_byte_input(int,int *);
unsigned long chanel_all_input(void);

/*
/*
/* SMPL10B.C function prototype define */
/*
/*
int fet_data_out(int,unsigned int);
int fet_bit_set(int,int,int);
unsigned int phot_data_input(void);
int phot_bit_check(int);
int fet_byte_puls_out(int,struct puls_out_infm *,int);
int fet_bit_puls_out(int,struct puls_out_infm *,int);

/*
/*
/* SMPL16B.C function prototype define */
/*
/*
unsigned int ppi96_port_adres_get(int,unsigned int *);
int ppi96_cw_port_acses(int);
int ppi96_cport_bit_check(int,int);
unsigned int ppi96_data_input(int);
int ppi96_data_out(int,int);
int ppi96_bit_set(int,int,int);
unsigned int ppi96_bit_check(int,int);
int ppi96_byte_puls_out(int,struct puls_out_infm *,int);
int ppi96_bit_puls_out(int,struct puls_out_infm *,int);

/*
/*
/* SMPL21B.C function prototype define */
/*
/*
int dcl6_data_out(int,unsigned int);
int dcl6_bit_set(int,int,int);
int dcl6_data_input(int,unsigned int *);
int dcl6_bit_check(int,int);
int dcl6_byte_puls_out(int,struct puls_out_infm *,int);
int dcl6_bit_puls_out(int,struct puls_out_infm *,int);

/*
/*
/* SMPL22B.C function prototype define */
/*
/*
int bit32_status_check(unsigned long,int);
int chanel32_input_check(int);
int chanel32_byte_input(int,int *);
unsigned long chanel32_all_input_pros(void);

/*
/*
/* SMPL23B.C function prototype define */
/*
/*
unsigned long bit32_status_set(int,unsigned long,int);
int chanel32_output(int,int);
int chanel32_byte_out(int,int);
void chanel32_all_out(unsigned long);
int chanel32_byte_puls_out(int,struct puls_out_infm *,int);
int chanel32_bit_puls_out(int,struct puls_out_infm *,int);

```

MAIN.C

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
#include <dir.h>
#include "def.h"
#include "sample.h"

unsigned int _stklen=1024*60; /* スタック領域の確保 */
struct EXP_DATA_FNAME fn[FN_MAX]; /* ファイル情報格納用構造体 */

/***** ウィンドウ関係各種定義 *****/

/* ウィンドウデータがないときのためのデータ群 */

WINDOW WD_info[ WINDOW_MAX ]={
{ 19,5,42,12,T_CYAN,T_WHITE,T_WHITE,0,"",1,0,0,2,0,0,0 },
{ 4,1,32,9,T_CYAN,T_YELLOW,T_WHITE,0,"",2,7,0,0,1,1,0 },

{ 4,10,70,12,T_GREEN,T_CYAN,T_WHITE,0,"",2,FN_MAX+1,0,0,1,0,0 },
{ 36,2,34,6,T_CYAN,T_YELLOW,T_WHITE,0,"ファイル処理",2,4,0,0,1,1,0 },

{ 14,8,50,5,T_RED,T_CYAN,T_WHITE,1,"",2,3,1,0,1,0,0 },
{ 2,17,76,3,T_CYAN,T_YELLOW,T_YELLOW,0,"メッセージ",2,1,0,2,0,0,0 },
{ 2,10,70,3,T_YELLOW,T_RED,T_WHITE,0,"",0,1,0,0,1,0,0 },
{ 9,14,60,10,T_RED,T_YELLOW,T_WHITE,0,"AD_DISP",1,1,0,2,0,0,0 },

{ 20,5,40,18,T_CYAN,T_YELLOW,T_WHITE,0,"RELAY_CON",2,16,0,0,1,0,0 },
{ 15,1,50,13,T_MAGENTA,T_YELLOW,T_WHITE,0,"コントロールテーブル",2,0,0,2,0,0,0 },

{ 6,3,20,9,T_YELLOW,T_CYAN,T_WHITE,0,"グラフメニュー",2,7,0,0,1,0,0 },
{ 8,7,20,8,T_YELLOW,T_CYAN,T_WHITE,0,"グラフ表示タイプメニュー",2,6,0,0,1,0,0 },
{ 10,12,40,13,T_YELLOW,T_CYAN,T_WHITE,0,"グラフ区間選択",2,14,0,0,1,0,0 },
{ 20,10,40,7,T_YELLOW,T_CYAN,T_WHITE,0,"描画データ選択",2,5,0,0,1,0,0 },

{ 16,3,30,6,T_BLUE,T_CYAN,T_WHITE,0,"判別メニュー",2,4,0,0,1,0,0 },
{ 2,1,70,23,T_CYAN,T_YELLOW,T_WHITE,0,"判別結果表示",1,0,0,1,0,0,0 },
{ 9,8,52,6,T_CYAN,T_YELLOW,T_WHITE,0,"センサ置みづけ",2,4,0,0,1,0,0 },
{ 2,15,76,10,T_CYAN,T_YELLOW,T_WHITE,0,"スクロールウィンドウ",1,0,0,1,0,0,0 }
};

char DIR[80]; /* カレントディレクトリ保存用 */
int cfn,MS_LB=0,MS_RB=0,MS_X=320,MS_Y=200;
int sbblink=30,txsense=120,nxsense=60,nysense=70;

int func_menu( void ) /*===== 各種機能選択メニュー */
{
    static char data[][DATA_WD]={
        "各チャンネル出力表示",
        "ソレー制御(16Ch)",
        "匂い測定開始",
        "匂い応答グラフ表示",
        "匂い鑑別",
        "匂いファイルの管理",
        "MS-DOSコマンドの実行"
    };
    static int c,flg=WD_OPEN;

    WD_info[ FUNC_MENU ].title="匂いセンシングシステム";
    c=twindow( 1,flg,&WD_info[ FUNC_MENU ],data );
    flg=WD_REMAIN;

    return c;
}

void init( void ) /*===== ファイル名の読み込み */
{
    FILE *fp;
    char dirtmp[80];
    int gd=DETECT,gn;

    setmem( fn,FN_SIZE*FN_MAX,0 );
    sprintf( dirtmp,"%s\\data\\%smell.env",DIR );
    if( (fp=fopen(dirtmp,"rb"))!=NULL )
    {
        fread( fn,FN_SIZE,FN_MAX,fp ); /* FN_MAX変更時ファイル内容に注意 */
        if ( feof( fp ) == NULL ) /* IBバージョンとの互換 */
        {
            fscanf( fp,"%d %d %d %d",&sblink,&nxsense,&nysense,&txsense );

```

```

        getch( fp );

        fread( WD_info, WINDOW_SIZE, WINDOW_MAX, fp );
    }
}
fclose( fp );
mouseinit();
mouseareaset();
mouseposset( 320, 200 );
setnewdriver( "PC98", detect98 );
initgraph( &gd, &gm, "B:¥¥tc" );
return;
}

void quit( void ) /*===== ファイル名の保存 及び 終了処理 */
{
    FILE *fp;
    char dirtmp[80];
    int wflg=0;

    twindow( WD_CLOSE );
    clrscr(); /* カーソルの表示 */
    textcursor( DISP_CURSOR );

    sprintf( dirtmp, "¥s¥¥data¥¥smell.env", DIR );
    if ( ( fp=fopen(dirtmp, "wb") )!=NULL )
    {
        fwrite( fn, FN_SIZE, FN_MAX, fp );
        fprintf( fp, "¥3d ¥3d ¥3d ¥3d¥n", sblink, nxsense, nysense, txsense );
        fwrite( WD_info, WINDOW_SIZE, WINDOW_MAX, fp );
    }
    fclose( fp );
    textcursor( DISP_CURSOR );
    if ( makesure( "現在のリレー状態の維持" )==2 )
    {
        port_data_output( 0, 0 );
        port_data_output( 1, 0 );
    }
    closegraph();
    exit(0);
}

void main( void ) /*===== メイン・プログラム */
{
    int i, i2, i3;
    ,C;
    char *tmp, tmpdir[80];

    getcwd( tmpdir, 80 ); /* 環境変数の読み出し */
    tmp=getenv( "DATA" );
    if ( tmp != NULL ) strcpy( DIR, tmp ); else strcpy( DIR, tmpdir );

    clrscr(); /* カーソルの表示 */
    init(); /* 初期化 */
    adrs_set( RELAY, 0 ); /* リレーの初期化 */
    port_data_output( 0, 0xf0 ); /* ガスセンサのヒータをON */
    port_data_output( 1, 0x00 );

    while( 1 )
    {
        switch( func_menu() )
        {
            case -1: if ( makesure("プログラムの終了")==1 ) quit();
                     break;
            case 0: graph_bar(); /* A/Dコンバータより入力表示 gas.c */
                     break;
            case 1: relay_control(); /* リレーの制御 gas.c */
                     break;
            case 2: experiment(); /* 実験開始・測定・記録 gas.c */
                     break;
            case 3: draw_graph(); /* グラフ表示 graph.c */
                     break;
            case 4: /$data_judge(); /* 匂い識別 Judge.c */
                     break;
            case 5: hd_file(); /* ファイル管理 file.c */
                     break;
            case 6: msdos();
                     break;
        }
    }
}

```


MAIN.C

default:break;

```

/*****
 *
 *          ウィンドウ関数 (twindow)
 *
 *   Wed. April 28th. 1993 MADE BY Toshiyuki Kirishima
 *
 *****/

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <jstring.h>
#include <jctype.h>
#include <mem.h>
#include <dos.h>
#include <bios98.h>
#include <time.h>
#include <graphics.h>
#include <alloc.h>
#include <stdlib.h>
#include <math.h>
#include "def.h"
#include "quo.h"

#define BS 0x08    /* Back Space */
#define LF 0x0A    /* Line Feed */
#define VT 0x0B    /* Vertical Tab */
#define FF 0x0C    /* Home Clear */
#define CR 0x0D    /* Carriage Return */
#define ESC 0x1B   /* Escape */
#define UP 0x0b
#define LEFT 0x08
#define RIGHT 0x0c
#define DOWN 0x0a
#define WUP '8'
#define WLEFT '4'
#define WRIGHT '6'
#define WDOWN '2'

#define REC 1 /* recover reverse string */
#define REV 0 /* print reverse string */

static struct FN_POS {          /* ファイル名位置記憶用構造体 */
    int x;
    int y;
} Pos[FN_MAX];

struct WD_PARA {
    int x1,y1;
    int x2,y2;
    void *adr;
};

static int mx,my;

int key_in( int sw,int sw2 );

char input( int rl,WINDOW *wd,char wddata[][DATA_WD] );
void getmen( struct WD_PARA wd_para[],int i,int x1,int y1,int x2,int y2 );
void flame( int mode,int mxpg,int uspg,WINDOW *wd );
void print( int mode,int pl,WINDOW *wd,char wddata[][DATA_WD] );
void dspdata( int ul,int c,WINDOW *wd,char wddata[][DATA_WD] );
void fprintf( int mode,int pl,WINDOW *wd,char wddata[][DATA_WD] );
void fdspdata( int ul,WINDOW *wd,char wddata[][DATA_WD] );
void allclose( void );

/***** ウィンドウサブルーチン *****/
int get_fname( char *,char * );
int makesure( char * );
void mesdisp( int,char * );

/***** マウス関係 *****/
int mouseinit();          /* マウスの初期化 */
void mouseense( int mx,int my); /* マウス感度の設定 */
void mouseon();           /* マウスカーソル表示 */
void mouseoff();          /* マウスカーソル消去 */
void mouseposset( int x,int y); /* マウスの位置セット */
void mouseinput();        /* 座標とキー入力 */
int mousediff();
void mouseput();

```

```

void mouseclr();
void trinput();

/*****
 *
 *      ウィンドウ関数 (twindow)
 *
 *****/
int twindow( int MAXPAGE, int mode, WINDOW wd[], char wddata[][DATA_WD] )
{
    static struct WD_PARA wd_para[ MAX_PAGE ]; /* ウィンドウ表示位置の記憶 */
    static int UseNum=-1; /* バッファ使用中の最新番号 */
    int Sec =-1; /* セクション指標用の変数 */
    int secst[ MAX_PAGE ]; /* セクション開始バッファ番号 */
    int secen[ MAX_PAGE ]; /* セクション終了バッファ番号 */
    int ul=0; /* データ指標位置 */
    int r1=0; /* データの絶対位置 */
    int o=0,rc=0; /* コマンド用変数 */
    int i4,i5,i,i2; /* 雑用変数 */
    int tm,dtm,x,y,vx,vy; /* 雑用変数 */
    int UsePage; /* 表示しているウィンドウ番号 */

    static int otb[] = { T_BLUE, T_GREEN, T_CYAN, T_RED, T_MAGENTA, T_YELLOW,
                        T_WHITE, T_BLUE }; /* カラーテーブル */

    textcursor( NODISP_CURSOR );

    if ( mode==WD_CLOSEV ) /* ウィンドウのクローズ */
    {
        if ( Sec==-1 ) return (-1); /* これ以上クローズできない */
        for ( i=secen[Sec]; i>=secst[Sec]; --i ) /* エリアの復元 */
        {
            puttext( wd_para[i].x1,wd_para[i].y1,
                    wd_para[i].x2,wd_para[i].y2,wd_para[i].adr );
            free( wd_para[i].adr );
            UseNum--;
        }
        Sec--;
        return ( 0 );
    }

    setmem( Pos, sizeof( struct FN_POS ) * 100, 0 );

    if ( mode==WD_OPEN :: mode==FW_OPEN ) /* ウィンドウのオープン */
    {
        Sec++; /* 次のセクションへ */
        UseNum++; /* 次のバッファ領域へ */
        if ( UseNum+MAXPAGE > MAX_PAGE ) return( 0 ); /* これ以上開けない */
        secst[Sec]=UseNum;
        if ( secen[Sec] < secst[Sec] :: MAXPAGE==1 ) secen[Sec]=secst[Sec];
        tm= secen[Sec]-secst[Sec];
        for ( i=0; i <= tm; ++i ) /* カレントページへと自動的に開く */
        {
            getmem( wd_para, UseNum, wd[i].x-1, wd[i].y /* バッファ領域の獲得 */
                    , wd[i].x+wd[i].wx-1+1, wd[i].y+wd[i].wy-1 );
            gettext( wd_para[UseNum].x1, wd_para[UseNum].y1, /* エリアの保存 */
                    wd_para[UseNum].x2, wd_para[UseNum].y2, wd_para[UseNum].adr );
            ul=( MAXPAGE > 1 ? wd[i].cmin : 0 );
            flame( mode, MAXPAGE, i, &wd[i] );

            if ( mode==FW_OPEN )
            {
                ul=wd[i].lineno_on;
                fdspdata( ul, &wd[i], wddata );
            }
            else dspdata( ul, 0, &wd[i], wddata );

            UseNum++;
        }
        UseNum--;
    }

    UsePage= secen[Sec]-secst[Sec]; /* 現在開いているページの設定 */
    if ( mode == WD_REMAIN ) {
        if ( Sec==-1 ) return(-1);
        ul=( MAXPAGE > 1 ? wd[UsePage].cmin : 0 );
        if ( wd[UsePage].button != 2 )
            flame( -1, MAXPAGE, UsePage, &wd[UsePage] );
        dspdata( ul, 0, &wd[UsePage], wddata );
    }
}

```

```

if ( mode == FW_REMAIN ) (
    ul=wd(UsePage).lineno_on;
    if ( wd(UsePage).button != 2 )
        flame( -1,MAXPAGE,UsePage,&wd(UsePage) );
    fdspdata( ul,&wd(UsePage),wddata );
)
if ( mode==FW_OPEN :: mode==FW_REMAIN )
{
    ul=wd(UsePage).lineno_on;
    rl=wd(UsePage).l;
    /*
        FW window と WD window との処理の違いについて一言。
        FW window では ul と rl を独立して扱っている。
        WD window では ul と rl は組み合わせて使っている。
    */
}
if ( wd(UsePage).inp_mode==1 ) return 0; /* 早くとリターン (表示のみの場合) */

rewind( stdin );

x=wd(UsePage).x;
y=wd(UsePage).y;
mx=320;
my=200;
mouseposset( mx,my );

while( 1 )
{
    if ( mode != FW_OPEN && mode != FW_REMAIN ) rl=ul+wd(UsePage).l;
    if ( (c=(rc==0 ? key_in( 0,0 ) : rc ))==0 )
    {
        if ( strlen( wddata[rl] )==0 && wd(UsePage).inp_mode==0
            && (wd(UsePage).inp_line==rl :: wd(UsePage).inp_line==--1) ) c=LEFT;
        else if ( rc==0 && wd(UsePage).cursor_on==1 )
        {
            if ( mode==FW_OPEN :: mode==FW_REMAIN )
            {
                fprintf( REV,rl,&wd(UsePage),wddata );
            } else print( REV,rl,&wd(UsePage),wddata );
        }
    }
    rc=0;

    switch( c )
    {
        case 'e' :/*sputchar( 7 );exit(0);*/ break;
        case RIGHT:if ( mode==FW_OPEN :: mode==FW_REMAIN )
            {
                fprintf( REC,rl,&wd(UsePage),wddata );
                if ( rl < ul+wd(UsePage).inp_line*(wd(UsePage).wy-2)-1 ) rl++;
                else if ( rl+1 < wd(UsePage).cmax-1 ){
                    ul+=wd(UsePage).inp_line;
                    rl++;
                    fdspdata( ul,&wd(UsePage),wddata );
                }
                fprintf( REV,rl,&wd(UsePage),wddata );
                break;
            }
        if ( UsePage==MAXPAGE-1 :: MAXPAGE==1 ) break;
        print( REC,rl,&wd(UsePage),wddata );
        UsePage++;
        UseNum++;
        secen{Sec}++;
        getnem( wd_para,UseNum,wd(UsePage).x-1
            ,wd(UsePage).y
            ,wd(UsePage).x+wd(UsePage).wx-1+1
            ,wd(UsePage).y+wd(UsePage).wy-1 );
        gettext( wd_para[UseNum].x1,wd_para[UseNum].y1,
            wd_para[UseNum].x2,wd_para[UseNum].y2,
            wd_para[UseNum].adr );
        flame( mode,MAXPAGE,UsePage,&wd(UsePage) );
        ul=( MAXPAGE > 1 ? wd(UsePage).cmin : 0 );
        dspdata( ul,0,&wd(UsePage),wddata );
        break;
        case DOWN:if ( mode == FW_OPEN :: mode == FW_REMAIN )
            {
                fprintf( REC,rl,&wd(UsePage),wddata );
                if ( rl+wd(UsePage).inp_line <=ul+wd(UsePage).inp_line*(wd(UsePage).wy-2)-1 ) rl+=wd(UsePage
            ).inp_line;
            } else {

```

```

        if ( r1+wd[UsePage].inp_line< wd[UsePage].cmax-1 ){
            ul+=wd[UsePage].inp_line;
            r1+=wd[UsePage].inp_line;
            fdspdata( ul,&wd[UsePage],wddata );
        }
        fprintf( REV,r1,&wd[UsePage],wddata );
        break;
    }
    if ( wd[UsePage].cmax <= 1 ) break;
    dtm=0;
    if ( wd[UsePage].l==wd[UsePage].wy-3
        && ul < wd[UsePage].cmax-(wd[UsePage].wy-2) )
    { /* スクロール処理 */
        ul++;
        print( REC,r1,&wd[UsePage],wddata );
        dspdata( ul,c,&wd[UsePage],wddata );
        dtm=20;
    } else if ( wd[UsePage].l < wd[UsePage].wy-3
        && wd[UsePage].l < wd[UsePage].cmax-1 )
    { /* スクロール無し */
        print( REC,r1,&wd[UsePage],wddata );
        wd[UsePage].l++;
        dtm=50;
    }
}
if ( dtm==0 && MAXPAGE==1 )
{
    print( REC,r1,&wd[UsePage],wddata );
    wd[UsePage].l=wd[UsePage].cmin;
    if ( ul > wd[UsePage].cmin )
    {
        ul=wd[UsePage].cmin;
        wd[UsePage].l=0;
        dspdata( ul,-1,&wd[UsePage],wddata ); /*!!!!!!!!!!!!!!*/
    }
    /**/
}
if ( wd[UsePage].inp_line != -1 ) print( REV,ul+wd[UsePage].l,&wd[UsePage],wddata );
delay( dtm );
break;
case UP : if ( mode == FW_OPEN :: mode == FW_REMAIN )
{
    fprintf( REC,r1,&wd[UsePage],wddata );
    if ( r1-wd[UsePage].inp_line >= ul ) r1-=wd[UsePage].inp_line;
    else {
        if ( ul>0 ){
            ul-=wd[UsePage].inp_line;
            r1-=wd[UsePage].inp_line;
            fdspdata( ul,&wd[UsePage],wddata );
        }
        fprintf( REV,r1,&wd[UsePage],wddata );
        break;
    }
}
if ( wd[UsePage].cmax <= 1 ) break;
dtm=0;
if ( wd[UsePage].l==0 && ul > wd[UsePage].cmin )
{ /* スクロール処理 */
    ul--;
    print( REC,r1,&wd[UsePage],wddata );
    dspdata( ul,c,&wd[UsePage],wddata );
    dtm=20;
} else if ( ul+wd[UsePage].l > wd[UsePage].cmin )
{ /* スクロール無し */
    print( REC,r1,&wd[UsePage],wddata );
    wd[UsePage].l--;
    dtm=50;
}
}
if ( dtm==0 && MAXPAGE==1 )
{
    print( REC,r1,&wd[UsePage],wddata );
    wd[UsePage].l=wd[UsePage].cmax-1;
    if ( wd[UsePage].l > wd[UsePage].wy-3 )
    {
        ul=wd[UsePage].cmax-1 -(wd[UsePage].wy-3);
        wd[UsePage].l=wd[UsePage].wy-3;
        dspdata( ul,-1,&wd[UsePage],wddata ); /*!!!!!!!!!!!!!!*/
    }
    /**/
}
}

```

```

if ( wd[UsePage].inp_line != -1 ) print( REV,ul+wd[UsePage].l,&wd[UsePage].wddata );
delay( dtm );
break;
case LEFT :if ( mode==FW_OPEN :: mode==FW_REMAIN )
{
    fprintf( REC,r1,&wd[UsePage].wddata );
    if ( r1 > ul ) r1--;
    else if ( ul>0 ){
        ul=wd[UsePage].inp_line;
        r1--;
        fdspdata( ul,&wd[UsePage].wddata );
    }
    fprintf( REV,r1,&wd[UsePage].wddata );
    break;
}

if ( wd[UsePage].inp_mode==0 &&
    (r1==wd[UsePage].inp_line :: wd[UsePage].inp_line== -1 ) )
{
    /* 内容修正 */
    if ( wd[UsePage].inp_line== -1 ) /* どの行でも入力モード */
    {
        wd[UsePage].inp_line=r1;
        rc=(int)input( r1,&wd[UsePage].wddata );
        wd[UsePage].inp_line= -1;
    } else rc=(int)input( r1,&wd[UsePage].wddata );
    print( REC,r1,&wd[UsePage].wddata );
}

if ( UsePage==0 :: MAXPAGE==1 ) break; /* クローズ処理 */
puttext( wd_para[UseNum].x1,wd_para[UseNum].y1,
          wd_para[UseNum].x2,wd_para[UseNum].y2,
          wd_para[UseNum].adr );
free( wd_para[UseNum].adr );
UseNum--;
UsePage--;
secon[Sec]--;
ul=( MAXPAGE > 1 ? wd[UsePage].cmin : 0 );
break;
case ' ' :
case CR : if ( mode==FW_OPEN :: mode==FW_REMAIN )
{
    wd[UsePage].l=r1;
    wd[UsePage].lineno_on=ul;
}
flame( WD_CLOSEV,MAXPAGE,UsePage,&wd[UsePage] );
return( r1 );
case ESC : delay( 100 ); /* ウィンドウ開数脱出 */
if ( mode==FW_OPEN :: mode==FW_REMAIN )
{
    wd[UsePage].l=r1;
    wd[UsePage].lineno_on=ul;
    fprintf( REC,r1,&wd[UsePage].wddata );
    /* これこそエラーの原因であった! fprintf と printf の違い! */
} else print( REC,r1,&wd[UsePage].wddata );
flame( WD_CLOSEV,MAXPAGE,UsePage,&wd[UsePage] );
return( -1 );

case 'P' : /* ウィンドウの表示位置をプリント */
gotoxy( 1,22 );cprintf( "x=%3d y=%3d",wd[UsePage].x,wd[UsePage].y);
getch();
gotoxy( 1,22 );cprintf( " ");
c=0;
break;
case 'F' : /* ウィンドウのフレームカラーを変更 */
for( i=0;i<7;++i ) if ( wd[UsePage].coll==ctb[i] ) break;
if ( i<7 ) wd[UsePage].coll=ctb[i+1];

flame( mode,MAXPAGE,UsePage,&wd[UsePage] );
if ( mode == FW_OPEN :: mode == FW_REMAIN )
{
    fdspdata( ul,&wd[UsePage].wddata );
} else {
    /*
    * ul=( MAXPAGE > 1 ? wd[UsePage].cmin : 0 );
    */
    dspdata( ul,0,&wd[UsePage].wddata );
}
c=0;
break;
case 'T' : /* ウィンドウのタイトルカラーを変更 */

```

```

for( i=0;i<7;++i ) if ( wd[UsePage].col2==ctb[i] ) break;
if ( i<7 ) wd[UsePage].col2=ctb[i+1];

flame( mode,MAXPAGE,UsePage,&wd[UsePage] );
if ( mode == FW_OPEN || mode == FW_REMAIN )
{
    fdspdata( ul,&wd[UsePage],wddata );
} else {
    /*
    * ul=( MAXPAGE > 1 ? wd[UsePage].cmin : 0 );
    */
    dspdata( ul,0,&wd[UsePage],wddata );
}
c=0;
break;
case 'S' : /* ウィンドウの文字列カラーを変更 */
for( i=0;i<7;++i ) if ( wd[UsePage].col3==ctb[i] ) break;
if ( i<7 ) wd[UsePage].col3=ctb[i+1];

flame( mode,MAXPAGE,UsePage,&wd[UsePage] );
if ( mode == FW_OPEN || mode == FW_REMAIN )
{
    fdspdata( ul,&wd[UsePage],wddata );
} else {
    /*
    * ul=( MAXPAGE > 1 ? wd[UsePage].cmin : 0 );
    */
    dspdata( ul,0,&wd[UsePage],wddata );
}
c=0;
break;
case 'H' : /* ウィンドウのタイプの変更 */
if ( wd[UsePage].button==0 ) wd[UsePage].button=1;
else wd[UsePage].button=0;
flame( mode,MAXPAGE,UsePage,&wd[UsePage] );
if ( mode == FW_OPEN || mode == FW_REMAIN )
{
    fdspdata( ul,&wd[UsePage],wddata );
} else {
    /*
    * ul=( MAXPAGE > 1 ? wd[UsePage].cmin : 0 );
    */
    dspdata( ul,0,&wd[UsePage],wddata );
}
c=0;
break;

default : break;
}
vx=0;vy=0;
switch( c )
{
    case '1' : vx=-1;vy= 1;break;
    case WDOWN : vx= 0;vy= 1;break;
    case '3' : vx= 1;vy= 1;break;
    case WLEFT : vx=-1;vy= 0;break;
    case WRIGHT : vx= 1;vy= 0;break;
    case '7' : vx=-1;vy=-1;break;
    case WUP : vx= 0;vy=-1;break;
    case '9' : vx= 1;vy=-1;break;
    default:break;
}
if ( vx==0 && vy==0 ) continue;
if ( x+vx > 2 && x+vx+wd[UsePage].wx < 80 ) x+=vx;
if ( y+vy > 0 && y+vy+wd[UsePage].wy < 26 ) y+=vy;
puttext( wd_para[UseNum].x1,wd_para[UseNum].y1,
          wd_para[UseNum].x2,wd_para[UseNum].y2,
          wd_para[UseNum].adr );
wd[UsePage].x=x;
wd[UsePage].y=y;
wd_para[UseNum].x1=wd[UsePage].x-1;
wd_para[UseNum].y1=wd[UsePage].y;
wd_para[UseNum].x2=wd[UsePage].x+wd[UsePage].wx-1+1;
wd_para[UseNum].y2=wd[UsePage].y+wd[UsePage].wy-1;
gettext( wd_para[UseNum].x1,wd_para[UseNum].y1,
          wd_para[UseNum].x2,wd_para[UseNum].y2,
          wd_para[UseNum].adr );
flame( mode,MAXPAGE,UsePage,&wd[UsePage] );
if ( mode == FW_OPEN || mode == FW_REMAIN )
{

```

```

        fdspdata( ul,&wd[UsePage],wddata );
    } else {
        /*
         * ul=( MAXPAGE > 1 ? wd[UsePage].cmin : 0 );
         */
        dspdata( ul,0,&wd[UsePage],wddata );
    }
} /* while */
}

/*****
 * 飛火野用ウィンドウ関数群
 *****/
void allclose( void )
{
    while( twindow( WD_CLOSE ) != -1 );
}
void getmen( struct WD_PARA wd_para[],int i,int x1,int y1,int x2,int y2 )
{
    unsigned int len;
    wd_para[i].x1=x1;
    wd_para[i].y1=y1;
    wd_para[i].x2=x2;
    wd_para[i].y2=y2;
    len= ( x2 - x1 +1 )*( y2 - y1 +1 )*4;
    if ( ( wd_para[i].adr=malloc(sizeof(char)*len ) )==NULL ) {
        clrscr();
        printf("malloc によるメモリーの獲得に失敗しました。\\n");
        exit(1);
    }
    setmen( wd_para[i].adr,len,0 );
}

char input( int r1,WINDOW *wd,char wddata[][DATA_WD] )
{
    static char *ptr,oldstr[DATA_WD],str[DATA_WD];
    int c,al,na,i,l2,numtmp,tmp,x,y;

    tmp=(wd->lineno_on==0 ? NUM_MERGIN-1 : 0 );
    numtmp=( wd->lineno_on==1 ? NUM_MERGIN-1 : 0 );
    strcpy( str,wddata[r1] );
    strcpy( oldstr,str );
    rewind( stdin );
    while( 1 )
    {
        textcursor( NODISP_CURSOR );
        l2=strlen( str );
        for(i=12;i<((wd->wx-7+tmp-1))+1) str[i]=' ';
        gotoxy( wd->x+MERGIN-NUM_MERGIN+numtmp,wd->y+1+wd->l/8*inp_lines/ );cprintf(str);
        for(i=12;i<((wd->wx-7+tmp-1))+1) str[i]='\\0';
        gotoxy( wd->x+MERGIN-NUM_MERGIN+numtmp+strlen(str),wd->y+1+wd->l/8*inp_lines/ );

        if ( strlen(str)>((wd->wx-7+tmp-1)-1 ) textcursor( NODISP_CURSOR );
            else textcursor( DISP_CURSOR );

        do{
            c=key_in( 0,1 );
        } while( c==0 );
        if ( c==DOWN :: c==UP :: c==ESC :: c==CR )
        {
            if ( c==CR ) c=DOWN;
            jstrncpy(wddata[r1],str,jstrlen(str));
            break;
        }

        switch( c )
        {
            case BS:al=jstrlen(str);na= strlen(str);
                if ( na>0 ){
                    ptr=jstradv(str,al-1);
                    *ptr='\\0';
                    if( na-strlen(str) == 2 ) *(ptr+1)='\\0';
                }
                break;
            case CR:jstrncpy( wddata[r1],str,jstrlen(str) );
                goto out;
            case l2:break;
            default: if ( strlen(str)<((wd->wx-7+tmp-1) )
                {
                    al=jstrlen(str);
                    ptr=jstradv(str,al);
                    *ptr=c;
                }
        }
    }
}

```



```

        if ( al == jstrlen(str) ){
            if ( strlen(str)==(wd->wx-7+tmp-1) ){
                putchar( 7 );
                *ptr='%0';
                rewind( stdin );
            }

            else ( c=getch();
                    *ptr+=c;
                  )
        }
    } else rewind( stdin );
    break;
}

out:
    textcursor( NODISP_CURSOR );
    return c;
}

void flame( int mode,int mxpg,int uspg,WINDOW wdw )
{
    int i,i2,x=wd->x+(wd->wx)/2-strlen(wd->title)/2;

    if ( mode==WD_CLOSEV /+:: mode== -1+/ )
    {
        if ( mxpg==1 && wd->button!=2 ){
            textreverse( REVERSE );
            textcolor( wd->coll );
            gotoxy( wd->x,wd->y );
            if ( mode== -1 ){
                textcolor( T_BLUE );
                cprintf("■");
            } else cprintf(" ");
            textreverse( NOREVERSE );
            textcolor( T_WHITE );
        }
        return;
    }

    window( wd->x,wd->y,wd->x+wd->wx-1,wd->y+wd->wy-1 );
    if ( mode != -1 ) clrscr();
    window( 1,1,80,25 );

    textcolor( wd->coll );
    if ( wd->button == 2 )
    {
        textreverse( NOREVERSE );
        gotoxy( wd->x,wd->y );
        cprintf("P");
    } else {
        textreverse( REVERSE );
        textcolor( T_BLUE );
        gotoxy( wd->x,wd->y );
        cprintf("■");
        textcolor( wd->coll );
        if ( wd->button==0 ) textreverse( NOREVERSE );
    }

    gotoxy( wd->x+wd->wx-1 -1 , wd->y );
    if ( wd->button == 1 ) cprintf(" "); else cprintf(" ");
    gotoxy( wd->x , wd->y+wd->wy-1 );
    if ( wd->button == 1 ) cprintf(" "); else cprintf("L");
    gotoxy( wd->x+wd->wx-1 -1 , wd->y+wd->wy-1 );
    if ( wd->button == 1 ) cprintf(" "); else cprintf("J");
    for( i=wd->x+2;i<wd->x+wd->wx-1-1;i+=2 )
    {
        if ( wd->button !=1 ){
            if ( i<x:: i>x+strlen(wd->title)-2 )
            {
                gotoxy( i,wd->y );cprintf("—");
            }
        }
        gotoxy( i,wd->y+wd->wy-1 );cprintf("—");
    }
    for( i=wd->y+1;i<wd->y+wd->wy-1;i++ )
    {
        gotoxy( wd->x,i );
        if ( wd->button == 1 ) cprintf(" "); else cprintf(" |");
        gotoxy( wd->x+wd->wx-1-1,i );cprintf(" |");
    }
    if ( wd->button==1 ){
        gotoxy( wd->x+wd->wx-2,wd->y+1 );cprintf("▲");
    }
}

```

```

gotoxy( wd->x+wd->wx-2,wd->y+wd->wy-1-1 );cprintf("▼");
if ( mxpg>1 && uspg>0 :: mode==FW_OPEN :: mode==FW_REMAIN ){
    textcolor( T_RED );
    gotoxy( wd->x+2 ,wd->y+wd->wy-1 );
    cprintf("《");
    textcolor( wd->col1 );
}
if ( mxpg>1 && uspg<mxpg-1 :: mode==FW_OPEN :: mode==FW_REMAIN ){
    textcolor( T_RED );
    gotoxy( wd->x+wd->wx-4,wd->y+wd->wy-1 );
    cprintf("》");
    textcolor( wd->col1 );
}
textcolor( wd->col2 );
for(i=wd->x+2;i<wd->x+wd->wx-2;++i){
    gotoxy( i,wd->y );cprintf(" ");
}
}
textcolor( wd->col2 );
gotoxy( x,wd->y );
textreverse( REVERSE );
cprintf(wd->title);
textcolor( T_WHITE );
textreverse( NOREVERSE );
return;
}

void dspdata( int ul,int c,WINDOW *wd,char wddata[][DATA_WD] )
{
    int i,i2=0;
    char tmp[DATA_WD];
    window( wd->x+2,wd->y+1,
            wd->x+wd->wx-3,wd->y+wd->wy-2 );
    if ( c==UP ) { gotoxy( 1,1 );insline(); }
    if ( c==DOWN ) { gotoxy( 1,1 );delline(); }
    if ( c==1 ) clrscr();
    window( 1,1,80,25 );
    for(i=ul;i<ul+wd->wy-2;++i)
    {
        if ( c==UP && i-ul >0 :: c==DOWN && i-ul < wd->wy-3 ) continue;
        gotoxy( wd->x+2,wd->y+1+i-ul );
        textcolor( wd->col1 );
        if ( wd->lineno_on==1 ) { i2=5;cprintf("X3d |",i+1 ); }
        textcolor( wd->col3 );
        setmem( tmp,DATA_WD,' ' );
        sprintf(tmp,"X-s",wddata[ i ]);
        tmp[strlen(wddata[i])]=' ';
        tmp[wd->wx-4-i2]=0;
        cprintf("X-s",tmp);
        textcolor( T_WHITE );
    }
}

void print( int mode,int pl,WINDOW *wd,char wddata[][DATA_WD] )
{
    int i,len,tmp;
    tmp=(wd->lineno_on==0 ? NUM_MERGIN : 0 );
    len=strlen( wddata[pl] );
    for ( i=len;i<wd->wx-7+tmp-1-1;++i) wddata[pl][i]=' ';
    wddata[pl][i]=0;
    textcolor( wd->col1 );
    if ( mode==REV ) textreverse( REVERSE );
    gotoxy( wd->x+2,wd->y+wd->1+1 );
    if ( wd->lineno_on==1 ) cprintf("X3d |",pl+1);
    textcolor( wd->col3 );
    cprintf("X-s",wddata[pl] );
    textreverse( NOREVERSE );
    textcolor( T_WHITE );
    for ( i=len;i<wd->wx-7+tmp-1-1;++i) wddata[pl][i]=' \0';
}

void fdspdata( int ul,WINDOW *wd,char wddata[][DATA_WD] )
{
    int i,i2,dx,dy;
    char prbuf[80];
    window( wd->x+2,wd->y+1,
            wd->x+wd->wx-3,wd->y+wd->wy-2 );
    clrscr();
    window( 1,1,80,25 );
    dx=dy=0;

```

```

for( i=ul;i<ul+wd->inp_line*(wd->wy-2);++i )
{
    Pos[i].x=wd->x+2+dx*(wd->wx-4)/wd->inp_line;
    Pos[i].y=wd->y+1+dy;
    gotoxy( Pos[i].x,Pos[i].y );
    setmem( prbuf,80,0 );
    for( i2=0 ; i2<(wd->wx-4)/wd->inp_line-1 ; i2++ )
        prbuf[i2]=( wddata[i][i2] == '¥0' ? ' ' : wddata[i][i2] );
    textcolor( wd->col3 );
    cprintf("X-s",prbuf);
    dx++;
    if ( dx==wd->inp_line ) { dx=0;dy++; }
}
textcolor( T_WHITE );
}

void fprint( int mode,int pl,WINDOW *wd,char wddata[][DATA_WD] )
{
    int i;
    char prbuf[80];
    setmem( prbuf,80,0 );
    for( i=0 ; i<(wd->wx-4)/wd->inp_line-1 ; i++ )
        prbuf[i]=( wddata[p1][i] == '¥0' ? ' ' : wddata[p1][i] );
    if ( mode==REV ) textreverse( REVERSE );
    gotoxy( Pos[p1].x,Pos[p1].y );
    textcolor( wd->col3 );
    cprintf("X-s",prbuf );
    textreverse( NOREVERSE );
    textcolor( T_WHITE );
}

void trig_wait( int lb,int rb )
{
    while ( lb==MS_LB && rb==MS_RB ) triginput();
    MS_LB=lb;
    MS_RB=rb;
    return;
}

int key_in( int sw,int sw2 )
{
    int c=0;
    do{
        if ( (c=( bdos(0x06,0xff,0) & 0x00ff ))!=0 ) return c;
        /*
        mouseinput();
        if ( MS_Y != my )
        {
            if ( MS_Y > my+nysense ) c=10;
            else if ( MS_Y < my-nysense ) c=11;
        }
        if ( MS_X != mx && c==0 )
        {
            if ( MS_X > mx+nxsense+txsense*sw2 ) c=12;
            else if ( MS_X < mx-nxsense-txsense*sw2 ) c=8;
        }
        if ( MS_LB != 0 )
        {
            putchar( 7 );
            trig_wait( MS_LB,MS_RB );
            c=13;
        } else if ( MS_RB != 0 )
        {
            putchar( 7 );
            trig_wait( MS_LB,MS_RB );
            c=27;
        }
        if ( MS_X < 20 :: MS_X > 620 :: MS_Y < 20 :: MS_Y > 380 )
        {
            mouseposset( 320,200 );
            mx=MS_X;
            my=MS_Y;
        }
        if ( c )
        {
            sw=0;
            mx=MS_X;
            my=MS_Y;
        }
    }
    */
}

```

```

    } while( sw != 0 );
    return c;
}

/*****
 *
 *          twindow を用いた各種ユーティリティ群
 *
 *****/

/*
WINDOW WD_info( MAKESURE )=( 14,8,50,5,T_RED,T_CYAN,T_WHITE,1,
                             "",2,3,1,0,1,0,0 );
*/
int makesure( char *title )
{
    char data[3](DATA_WD);
    int c;
    char tmp[80];

    WD_info( MAKESURE ).title="";
    setmem( data,DATA_WD,3,0 );
    setmem( data[0],DATA_WD-1,' ' );
    sprintf( &data[0],abs( (46-12-strlen(title))/2 ),"%sをしますか?",title);
    sprintf( data[1],"          はい" );
    sprintf( data[2],"          いいえ" );
    c=twindow( 1,WD_OPEN,&WD_info( MAKESURE ),data );
    twindow( WD_CLOSE );
    return c;
}

/*
WINDOW WD_info( MESDISP )=( 2,17,76,3,T_CYAN,T_YELLOW,T_YELLOW,0,
                             "メッセージ",2,1,0,2,0,0,0 );
*/
void mesdisp( int col,char *title )
{
    char data[1](DATA_WD);

    WD_info( MESDISP ).title="メッセージ";
    WD_info( MESDISP ).inp_mode=2;
    setmem( data,DATA_WD,0 );
    sprintf( data,"%s",title );
    WD_info( MESDISP ).col3=col;
    twindow( 1,WD_OPEN,&WD_info( MESDISP ),data );
    twindow( WD_CLOSE );
    return;
}

void scrdisp( int flg,char *title,char *str )
{
    static char data[8](DATA_WD),pt;
    int i;
    char tmp[80];

    if ( flg==WD_OPEN )
    {
        setmem( data,DATA_WD,8,0 );/* スクロールウィンドウクリア */
        pt=0;
    }
    WD_info( SCR_DISP ).title=title;
    WD_info( SCR_DISP ).x=2;
    WD_info( SCR_DISP ).y=15;
    WD_info( SCR_DISP ).wx=76;
    WD_info( SCR_DISP ).wy=10;
    WD_info( SCR_DISP ).col1=T_CYAN;
    WD_info( SCR_DISP ).col2=T_YELLOW;
    WD_info( SCR_DISP ).col3=T_WHITE;
    WD_info( SCR_DISP ).l=0;
    WD_info( SCR_DISP ).inp_mode=1;
    WD_info( SCR_DISP ).cmax=0;
    WD_info( SCR_DISP ).cmin=0;
    WD_info( SCR_DISP ).button=1;
    WD_info( SCR_DISP ).cursor_on=0;
    WD_info( SCR_DISP ).lineno_on=0;
    WD_info( SCR_DISP ).inp_line=0;

    sprintf( tmp,"%s",str );

```

WINDOW.C

```

strcpy( data[pt],tmp );
twindow( 1,fl,&WD_info[ SCR_DISP ],data );
if ( pt==7 )
{
    for( i=0; i<7 ;++i ) strcpy( data[i],data[i+1] );
    setmem( data[7],DATA_WD,0 );
} else pt++;
return;
}

void nowaitdisp( int col,char *title )
{
    char data[11][DATA_WD];

    WD_info[ MESDISP ].title="おしらせ";
    WD_info[ MESDISP ].inp_mode=1;
    setmem( data,DATA_WD,0 );
    sprintf( data,"%s",title );
    WD_info[ MESDISP ].col3=col;
    twindow( 1,WD_OPEN,&WD_info[ MESDISP ],data );
    return;
}

/*
WINDOW WD_info[ GET_FNAME ]={ 2,10,70,3,T_YELLOW,T_RED,T_WHITE,0,
                                "",0,1,0,0,1,0,0 };
*/
int get_fname( char *mes,char *fname ) /*===== ファイル名の取得 */
{
    int c,i,fl;
    char tmp[11][DATA_WD],message[DATA_WD];

    sprintf( message,"%s",mes );
    WD_info[ GET_FNAME ].title=message;
    while( 1 )
    {
        setmem( tmp,DATA_WD,0 );
        c=twindow( 1,WD_OPEN,&WD_info[ GET_FNAME ],tmp );
        twindow( WD_CLOSE );
        if ( c==-1 ) return -1;
        fl=0;
        for( i=0;i<FN_MAX;++i )
        {
            if ( strcmp( tmp[i],fn[i].fname )==0 && strlen( tmp[i] ) )
            {
                mesdisp( T_YELLOW,"入力内容が不適当です。違う文字列を入力してください。");
                fl=1;
            }
        }
        if ( fl==1 ) continue;
        if ( strlen( tmp ) > 8 )
        {
            mesdisp(T_RED,"入力した文字列が長すぎます！8文字までです。");
            continue;
        }
        break;
    }
    strcpy( fname,tmp );
    return 0;
}

void msdos( void )
{
    static char dos[DATA_WD];
    int s;

    if ( (s=makesure("コマンドラインからの実行"))==1 ) return;
    if ( s==1 )
    {
        mesdisp(T_WHITE,"戻るときは、EXIT と入力してください。");
        clrscr();
        system("");
        clrscr();
        return;
    }
    if ( get_fname("MS-DOSコマンドの実行",dos)!=-1 )
    {
        clrscr();
        if ( system(dos)!=0 )
        {

```

WINDOW.C

```

        mesdisp( T_WHITE, "コマンドが不適切です。");
    } else mesdisp(T_WHITE, "リターンキーを押してください。");
    clrscr();
}
return;
}

/*
*****
*      マウスハンドラー      *
*****
*/
/*
-----
マウスの初期化
-----
*/
int mouseinit( void )
{
    union REGS  regs;
    struct SREGS segregs;

    regs.x.ax=0;
    int86(0x33,&regs,&regss);
    if(regs.x.ax==0) return -1;
    mouseclr();
    mousesense(1,1);
    regs.x.ax=0x09;
    regs.x.bx=0;
    regs.x.cx=0x01;
    int86(0x33,&regs,&regss);
}
/*
-----
マウスの感度設定
-----
*/
void mousesense(int mx,int my) /* マウス感度の設定 */
{
    union REGS  regs;

    regs.x.ax=15;
    regs.x.cx=mx;
    regs.x.dx=my;
    int86(0x33,&regs,&regss);
}
/*
-----
マウスカーソルのクリア
-----
*/
void mouseclr()
{
    union REGS  regs;

    regs.x.ax=2;
    int86(0x33,&regs,&regss);
}
/*
-----
マウスカーソルの位置設定
-----
*/
void mouseposset(int x,int y)
{
    union REGS  regs;

    regs.x.ax=4;
    regs.x.cx=x;
    MS_X=x;
    regs.x.dx=y;
    MS_Y=y;
    int86(0x33,&regs,&regss);
}
/*
-----
マウスカーソルの領域設定
-----
*/
void mouseareaset( void )

```

```

{
    union REGS    regs;

    regs.x.ax=0x10;          /* DIRECTION: HORIZON */
    regs.x.cx=0;
    regs.x.dx=640;
    int86(0x33,&regs,&regs);
    regs.x.ax=0x11;          /* DIRECTION: VERTICAL */
    regs.x.cx=0;
    regs.x.dx=400;
    int86(0x33,&regs,&regs);
}
/*
-----
マウスの各種データの取得
-----
*/
void mouseinput( void )
{
    union REGS    regs;

    regs.x.ax=3;
    int86(0x33,&regs,&regs);
    MS_LB = regs.x.ax; /* 左ボタン */
    MS_RB = regs.x.bx; /* 右ボタン */
    MS_X = regs.x.cx; /* X座標 */
    MS_Y = regs.x.dx; /* Y座標 */
    return;
}
/*
-----
マウスのボタンデータの取得
-----
*/
void triginput( void )
{
    union REGS    regs;

    regs.x.ax=3;
    int86(0x33,&regs,&regs);
    MS_LB = regs.x.ax; /* 左ボタン */
    MS_RB = regs.x.bx; /* 右ボタン */
    return;
}

```

```

/* ..... */
/* ..... */
/* :: A B 9 8 - 0 5 B用サンプルプログラム :: */
/* :: Copyright ADTEK SYSTEM SCIENCE Co.,ltd 1991 :: */
/* ..... */
/* ..... */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <alloc.h>
#include <process.h>
#include <math.h>
#include <io.h>
#include <dos.h>
#include <fcntl.h>
#include <dir.h>
#include <limits.h>
#include <time.h>
#include <mem.h>
#include <conio.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/timeb.h>
#include "sample.h"
#include "def.h"
#include "quo.h"

/* ----- */
/* 関数名 : ad_read */
/* ----- */
/* ..... */
/* ..... */
/* ■ 要約 ■ */
/* ..... */
/* int ad_read(chanel_no,ad_data) */
/* ..... */
/* int chanel_no : A/D変換操作対象となるチャンネルNo.を指定する. */
/* ..... */
/* 1から8までのデータ */
/* ..... */
/* unsigned int *ad_data : 変換データを格納するポインタ */
/* ..... */
/* ■ 概要 ■ */
/* ..... */
/* A/D変換操作対象となるチャンネルNo.からデジタル値を読み込みます. */
/* チャンネルNo.の誤った指定は変換入力をしません. */
/* ..... */
/* ■ 戻り値 ■ */
/* ..... */
/* 0 : 正常変換終了( *ad_data に変換値が入る ) */
/* 1 : チャンネルNo.指定エラー */
/* ..... */
/* ----- */
int ad_read(chanel_no,ad_data)
int chanel_no;
unsigned int *ad_data;
{
    unsigned int i,j,k;
    unsigned int ad_start_data;
    unsigned int ad_read_data;

    /* チャンネルNo. OK? */
    if((chanel_no == 0) :: (chanel_no > 8) :: (chanel_no < 0)){
        /* N.G */
        return(1);
    }

    /* サンプル指定 */
    ad_start_data = (chanel_no - 1) & 0x07;
    outp(select_prt,ad_start_data);

    /* 電圧が完全に取り込まれるのを待つ(ウェイト) */
    for(i = 0;i < 64;++i){
        ;
    }

    /* ホールド指定 */
    ad_start_data := 0x08;
    outp(select_prt,ad_start_data);

```



```

/* A/D変換スタート */
outp(start_prt,1);

/* A/D変換終了待ち */
while(1){
    i = inp(select_prt);
    /* EOC ON ? */
    if(i & 0x01){
        break;
    }
}

/* A/D変換値の読み込み & データ作成 */

/* 下位4ビットデータを得る */
i = inp(low_data_prt) & 0xf0;
ad_read_data = (i >> 4) & 0x0f;
/* 上位8ビットデータを得る */
j = inp(high_data_prt);
/* 最上位4ビットデータを得る */
i = ((j & 0xf0) >> 4) & 0x0f;
/* 真ん中の4ビットデータを得る */
k = ((j & 0x0f) << 4) & 0xf0;
/* 12ビットデータ作成 */
ad_read_data |= k;
ad_read_data := ((i << 8) & 0xf000);

*ad_data = ad_read_data;

return(NULL);
}

/* ----- */
/* 関数名 : ad_count_read */
/* ----- */
/*
/* ■ 要約 ■
/*
/* int ad_count_read(chanel_no, read_cnt, ad_data)
/*
/* int chanel_no : A/D変換操作対象となるチャンネルNo.を指定する。
/*
/*      1から8までのデータ
/*
/* unsigned int read_cnt : 入力する回数 ( 1から65535まで )
/*      指定された回数分入力した後, 平均化したデータを
/*      算出し結果とします。
/*
/* unsigned int *ad_data : 変換データを格納するポインター
/*
/* ■ 概要 ■
/*
/* A/D変換操作対象となるチャンネルNo.からデジタル値を読み込みます。
/* 各パラメータの指定値が誤ったものを指定した場合は変換入力をしません。
/*
/* ■ 戻り値 ■
/*
/* 0 : 正常変換終了( *ad_data に変換値が入る )
/* 1 : チャンネルNo.指定エラー
/* 2 : 入力回数指定エラー
/*
/* ----- */
int ad_count_read(chanel_no, read_cnt, ad_data)
int chanel_no;
unsigned int read_cnt;
unsigned int *ad_data;
{
    int loop_cnt;
    unsigned int i, j, k;
    unsigned int ad_start_data;
    unsigned int ad_read_data;
    unsigned long ad_add_data;

    *ad_data = NULL;
    ad_add_data = (long)NULL;
    loop_cnt = read_cnt;

```

```

/* 入力回数はO.K? */
if(read_cnt == NULL){
    /* N.G */
    return(2);
} else {
    /* チャンネルNo.O.K? */
    if((chanel_no == 0) || (chanel_no > 8) || (chanel_no < 0)){
        /* N.G */
        return(1);
    }
}

while(loop_cnt){
    ad_read(chanel_no,&ad_read_data);
    /* 指定回数分データを加える */
    ad_add_data += (unsigned long)ad_read_data;
    loop_cnt--;
}

/* 平均化 & 結果セット */
sad_data = (unsigned int)(ad_add_data / (unsigned long)read_cnt);

return(NULL);
}

/* ----- */
/* 関数名 : analog_scale_convert */
/* ----- */
/*
/* ■ 要約 ■
/*
/* double analog_scale_convert(voltage_spec,ad_data,scale_max,scale_min)
/*
/* int voltage_spec : ボードの電圧設定を指定する( 0から4までのデータ )
/*
/*
/* 0 : ±2.5 v
/* 1 : ±5 v
/* 2 : ±10 v
/* 3 : 0-5 v
/* 4 : 0-10 v
/*
/* unsigned int ad_data : A/D変換したデータ( 生データ )
/*
/* double scale_max : 電圧値最大( 0FFFF )の時のスケール値
/*
/* double scale_min : 電圧値最小( 0 )の時のスケール値
/*
/* ■ 概要 ■
/*
/* A/D変換操作により得られた値をボードの入力電圧仕様によって指定される
/* スケール値に変換するプログラムです。
/*
/* ■ 戻り値 ■
/*
/* スケール値に合わせた値を返す。
/* (電圧設定の指定が誤っている場合は強制的に±2.5 v仕様として変換します。)
/*
/* ----- */
double analog_scale_convert(voltage_spec,ad_data,scale_max,scale_min)
int voltage_spec;
unsigned int ad_data;
double scale_max;
double scale_min;
{
    double calcu_data;
    double max_min_def;

    max_min_def = scale_max - scale_min;

    ad_data &= 0xffff;

    switch(voltage_spec){
        case 0: /* ±2.5 v */
        case 1: /* ±5 v */
        case 2: /* ±10 v */
            calcu_data = (((double)ad_data * max_min_def) / 4096.0) + scale_min;
            break;

```

```

case 3: /* 0-5 V */
case 4: /* 0-10 V */
    if(scale_max <= 0.0){
        calcu_data = ((double)ad_data * scale_min) / 4089. /*4096.0*/;
    } else {
        calcu_data = ((double)ad_data * scale_max) / 4089. /*4096.0*/;
    }
    break;

default:
    /* ±2.5 V */
    calcu_data = (((double)ad_data * max_min_def) / 4096.0) + scale_min;
}

return(calcu_data);
}

```

```

/* ----- */
/* 関数名 : voltage_convert */
/* ----- */
/*
/* ■ 要約 ■
/*
/* double voltage_convert(voltage_spec, ad_data)
/*
/* int voltage_spec : ボードの電圧設定を指定する( 0から4までのデータ )
/*
/*
/* 0 : ±2.5 V
/* 1 : ±5 V
/* 2 : ±10 V
/* 3 : 0-5 V
/* 4 : 0-10 V
/*
/* unsigned int ad_data : A/D変換したデータ( 生データ )
/*
/* ■ 概要 ■
/*
/* A/D変換操作により得られた値をボードの入力電圧仕様によって電圧値に変換
/* するプログラムです。
/*
/* ■ 戻り値 ■
/*
/* 電圧値に変換した値を返す。
/* (電圧設定の指定が誤っている場合は強制的に±2.5 V仕様として変換します。)
/*
/* ----- */
double voltage_convert(voltage_spec, ad_data)
int voltage_spec;
unsigned int ad_data;
{
    double calcu_data;

    ad_data &= 0xffff;

    switch(voltage_spec){
        /* ±2.5 V */
        case 0:
            calcu_data = (double)((double)ad_data * 5.0 / 4096.0) - 2.5;
            break;

        case 1:
            /* ±5 V */
            calcu_data = (double)((double)ad_data * 10.0 / 4096.0) - 5.0;
            break;

        case 2:
            /* ±10 V */
            calcu_data = (double)((double)ad_data * 20.0 / 4096.0) - 10.0;
            break;

        case 3:
            /* 0-5 V */
            calcu_data = (double)((double)ad_data * 5.0 / 4096.0);
            break;

        case 4:

```



```

/* ----- */
/* ----- */
/*   A B 9 8 - 4 7 B 用 サンプル プログラム   [ E X 4 7 . C ]   */
/*   Copyright ADTEK SYSTEM SCIENCE Co.,ltd 1992               */
/* ----- */
/* ----- */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <process.h>
#include <io.h>
#include <dos.h>
#include <fcntl.h>
#include <dir.h>
#include <limits.h>
#include <mem.h>
#include <conio.h>
#include <signal.h>
#include <sys\types.h>
#include <sys\timeb.h>
#include <math.h>
#include <time.h>
#include <alloc.h>
#include "def.h"
#include "quo.h"

/* ----- */
/* 関数名 : adrs_set(port, adrs_cond) */
/* ----- */
/*
/*   ■ 要約 ■
/*
/*   void adrs_set(port)
/*       unsigned int port      先頭のポートアドレス
/*       unsigned int adrs_cond adrs_cond = 0 連続アドレス
/*       adrs_cond != 0 偶数または奇数アドレス
/*
/*   ■ 概要 ■
/*
/*   ボードのアドレスを設定し、E X 4 7 . C の変数に各ポートのアドレス値
/*   をセットします。
/*   E X 4 7 . C を使用する際に、最初に 1 回設定する。
/*
/*   ■ 戻り値 ■
/*
/*   戻り値はありません
/*
/* {} ----- {}
/* アドレスの設定 A B 9 8 4 7 . E X E を実行する際に内部で使用
/* {} ----- {}
void adrs_set(port, adrs_cond)
unsigned int port;
unsigned int adrs_cond;
{
    if(adrs_cond == 0){
        prt_0 = port; /* 連続アドレスの場合 */
        prt_1 = port+1;
    }
    else {
        prt_0 = port; /* 偶数または奇数アドレスの場合 */
        prt_1 = port+2;
    }
}

/* ----- */
/* 関数名 : port_adres_get */
/* ----- */
/*
/*   ■ 要約 ■
/*
/*   int port_adres_get(port_no, adres_data)
/*       int port_no;      ポートナンバー
/*       unsigned int *adres_data; ポートアドレスをセットするポインタ
/*
/*       port_no = 0 : ポート 0
/*       port_no = 1 : ポート 1
/*   ■ 概要 ■
/*

```

```

/* ポートナンバーにより、ポートアドレスを求める */
/*
/* ■ 戻り値 ■
/*
/* 0 : ポートアドレス取得完了 ( 正常終了 )
/* -1 : ポート No. エラー
/*
/* ----- */
unsigned int port_adres_get(port_no, adres_data)
int port_no;
unsigned int *adres_data;
{
    unsigned int ad_dat;

    switch(port_no)
    {
        case 0:
            /* 出力ポート 0 */
            ad_dat = prt_0;
            break;
        case 1:
            /* 出力ポート 1 */
            ad_dat = prt_1;
            break;
        default:
            return(-1);
            break;
    }
    *adres_data = ad_dat;
    return(0);
}

/* ----- */
/* 関数名 : port_data_output */
/*
/* ----- */
/*
/* ■ 要約 ■
/*
/*
/* int port_data_output(port_no, out_data)
/*
/* int port_no : 出力操作対象となるポートを指定する.
/*
/* port_no = 0 : ポート 0
/* port_no = 1 : ポート 1
/*
/* int out_data : 出力するデータ
/*
/* ■ 概要 ■
/*
/* 出力操作対象となるポートに 1 バイトデータを出力するプログラムです.
/* 出力操作対象とするポートがコントロールポート設定により出力ポートに設定されて
/* いることを前提とします.
/* ポート No. は 0 から 1 ままでそれ以外の値を設定した時は何も出力しません.
/*
/* ■ 戻り値 ■
/*
/* 0 : 出力終了 ( 正常終了 )
/* -1 : ポート No. エラー
/*
/* ----- */
int port_data_output(port_no, out_data)
int port_no;
int out_data;
{
    unsigned int address_data;
    unsigned int output_data;

    /* ポート No. によりアドレス値を求める */
    if(0 != port_adres_get(port_no, &address_data)){
        return(-1);
    }

    output_data = out_data;

    if(port_no > 1 || port_no < 0)
        return(-1);

    output_data &= 0x0ff;
    outp(address_data, output_data);
}

```

```

    return(0);
}

/* ----- */
/* 関数名 : port_data_input */
/* ----- */
/*
/* ■ 要約 ■
/*
/* int port_data_input(port_no)
/*
/* int port_no : 入力操作対象となるポートを指定する。
/*
/* port_no = 0 : ポート0
/* port_no = 1 : ポート1
/*
/* ■ 概要 ■
/*
/* 入力操作対象となるポートから1バイトデータを入力するプログラムです。
/* ポートNo. は0, 1まででそれ以外の値を設定した時はFFFFHを
/* 返します。(入力操作は実施しない)
/*
/* ■ 戻り値 ■
/*
/* 指定されたポートの入力した値を返します。
/*
/* ----- */
port_data_input(port_no)
int port_no;
{
    unsigned int address_data;
    int input_data;

    /* 対象外のポートを入力しようとしたか? */
    if((port_no > 1) || (port_no < 0)){
        /* ポート指定エラー */
        return(-1);
    }

    /* ポートNo.によりアドレス値を求める */
    if(0 != port_adres_get(port_no, &address_data)){
        return(-1);
    }

    /* 指定されたポートから入力 */
    input_data = (inp(address_data) & 0x00ff);

    return(input_data);
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <math.h>
#include <mem.h>
#include <dos.h>
#include <graphics.h>
#include <io.h>
#include "def.h"
#include "quo.h"

extern char txt[8000];

int kekka=0, /* 判別結果のグラフ表示 */
    keika=0, /* 判別中の経過を表示 */
    hosei=0; /* 補正曲線の表示 */

#define TGS1 0 /* 0..alcohol */
#define TGS2 1 /* 1..ammonia */
#define TGS3 3 /* 3..methene */

double PD[7][1000]; /* プロットデータ格納用配列 */
static int ratio[ 7 ]=( 100,100,100,100,100,100,100 ); /* 重み記憶用配列 */
static int stpcol[11]=( BLUE,RED,CYAN,GREEN,YELLOW, /* ステップ色 */
    BROWN,MAGENTA,BLUE,BLUE,BLUE );
struct TGS_INFO tgs[7]={ /* ガスセンサ情報用構造体 */
    ( "アンモニアガス",4700.,YELLOW,"黄" ),
    ( "酸化水素ガス", 4700.,CYAN,"水" ),
    ( "メタンガス", 4700.,MAGENTA,"紫" ),
    ( "アルコールガス",4700.,WHITE,"白" ),
    ( "温度1", 200.,BLUE,"青" ),
    ( "温度2", 200.,RED,"赤" ),
    ( "温度", 200.,GREEN,"緑" )
};
static struct GRAPH_INFO { /* グラフスケール情報構造体 */
    int x1,y1,x2,y2;
    char stitle, sxnes, syaes;
} g[3]={
    ( 55, 0,370,300," (電圧-時間)", "[Sec]","[V]" ),
    ( 55, 0,370,300," (抵抗-時間)", "[Sec]","[Ω]" ),
    ( 55, 0,370,300," (消費電力-時間)", "[Sec]","[mW]" ),
};
int angle /* 3D表示の際の角度 (y軸) */
,clrflg /* グラフ重ね合わせフラグ */
,crair; /* 清浄大気基準フラグ */

void V_rectify();

void draw_scale( int s1,int s2,int s3 ) /* グラフ用スケールの描写 */
{
    int i,i2,tp,tp2
    ,px= 380,py= 22 /* Valuables for Phase Chart */
    ,tx= 380,ty=207 /* Valuables for Sensor Type */
    ,dx,dy;
    static char tmp[80];

    /* フェーズメニュー表示 */
    setcolor( YELLOW );
    outtextxy( px,py-18," Experimental Procedure");
    setcolor( RED );

    line( px+40,py-2,px+27*8,py-2 );
    setcolor( LIGHTCYAN );
    for( i=0;i<14/* 注意 */;++i )
    {
        sprintf( tmp,"No%2d Switching Interval %3d(Sec)",
            i+1,SH[i].interval );
        outtextxy( px,py+i*13,tmp );
    }

    tp=( s3==14 ? 14 : s3 );
    tp2=strlen( tmp );
    if ( s3<14 ) /* 全ステップ表示の場合 */
    {
        setcolor( YELLOW ); /* 選択ステップを枠で囲む */
        rectangle( px-2,py+tp*13+2,px+tp2*8,py+tp*13+14 );
    }

    /*

```



```

settextstyle( DEFAULT_FONT,HORIZ_DIR,1 );
*/

/**** 測定日時を表示 ****/
setcolor( LIGHTGRAY );          /* 測定日時を表示 */
sprintf( tmp, " (測定開始日時) %s", fn(cfn).st );
outtextxy( 0,360,tmp );
outtextxy( 0,380,tmp );
sprintf( tmp, " (測定終了日時) %s (気温) %3d (湿度) %3d",
          fn(cfn).ed,fn(cfn).temp,fn(cfn).humid );
outtextxy( 0,380,tmp );

/**** グラフカラーとセンサの対応表示 ****/
setcolor( YELLOW );
outtextxy( tx,ty, " Line color - Sensor type" );
setcolor( RED );
line( tx+24,ty+17,tx+27*8,ty+17 );
setcolor( LIGHTGRAY );
for( i=0 ; i<4 ; ++i )
{
    sprintf( tmp, "[%s]-%sセンサ出力", tgs[i].cn, tgs[i].name );
    outtextxy( tx,ty+20+i*16,tmp );
}
/**** グラフスケールの描写 ****/
switch( s2 )
{
    case 0: /**** 通常グラフ表示の場合 ****/
        rectangle( g[s1].x1,g[s1].y1,g[s1].x2,g[s1].y2 );
        setcolor( CYAN );
        outtextxy( g[s1].x2+4 ,g[s1].y2-10,g[s1].xmes );
        setcolor( YELLOW );
        outtextxy( g[s1].x1+(g[s1].x2-g[s1].x1-strlen(g[s1].title)*9)/2,
                    g[s1].y2+22, g[s1].title );
        setcolor( WHITE );
        outtextxy( g[s1].x1-8,g[s1].y2,"0" );
        break;
    case 1: /**** 3次元表示の場合 ****/
        angle=45;
        line( g[s1].x1,g[s1].y1+70,g[s1].x1 ,g[s1].y2 );
        line( g[s1].x1,g[s1].y2 ,g[s1].x2-40,g[s1].y2 );
        dx=80.*cos(angle*M.PI/180.);
        dy=80.*sin(angle*M.PI/180.);
        line( g[s1].x1,g[s1].y2,g[s1].x1+dx,g[s1].y2-dy );
        setcolor( tgs[TGS1].col );
        outtextxy( g[s1].x2+6-40 ,g[s1].y2-10 , "TGSX1" );
        setcolor( tgs[TGS2].col );
        outtextxy( g[s1].x1+dx ,g[s1].y2-dy-24,"TGSX2" );
        setcolor( tgs[TGS3].col );
        outtextxy( g[s1].x1-40 ,g[s1].y1+70-24,"TGSX3" );
        setcolor( YELLOW );
        outtextxy( g[s1].x1+(g[s1].x2-g[s1].x1-18*8)/2,
                    g[s1].y2+32,"センサ応答3D表示" );
        setcolor( WHITE );
        outtextxy( g[s1].x1-8,g[s1].y2,"0" );
        break;
    case 2: /**** 多角形表示の場合 ****/
        setcolor( CYAN );
        dx= (g[s1].x2-g[s1].x1)/2.+g[s1].x1 -20;
        dy= (g[s1].y2-g[s1].y1)/2.+g[s1].y1 +20;
        line( g[s1].x1 -20,dy,g[s1].x2 -20,dy );
        line( dx,g[s1].y1 +20,dx,g[s1].y2 +20 );
        setcolor( YELLOW );
        outtextxy( g[s1].x1+(g[s1].x2-g[s1].x1-20*8)/2-20,
                    g[s1].y2+32,"センサ応答多角形表示" );
        setcolor( WHITE );
        outtextxy( dx -10,dy +2,"0" );
        break;
    case 3: /**** 出力比率表示の場合 ****/
        rectangle( g[s1].x1,g[s1].y1+30,g[s1].x2,g[s1].y2 );
        setcolor( CYAN );
        outtextxy( g[s1].x2+4 ,g[s1].y2-10,g[s1].xmes );
        setcolor( YELLOW );
        outtextxy( g[s1].x1+(g[s1].x2-g[s1].x1-12*8)/2,
                    g[s1].y2+32,"出力比率表示" );
        setcolor( WHITE );
        outtextxy( g[s1].x1-8,g[s1].y2,"0" );
        default:break;
}
return;
}

```

```

double unit( int s1,double value,char un ) /***** 単位文字列処理 *****/
{
    double div;
    div=1;
    switch( s1 )    /* 単位 */
    {
        case 0: strcpy( un,"[V]" );
                    break;
        case 1: strcpy( un,"[Ω]" );
                    if ( value >1e3 )
                    {
                        strcpy( un,"[kΩ]" );
                        div=1e3;
                    }
                    if ( value >1e6 )
                    {
                        strcpy( un,"[MΩ]" );
                        div=1e6;
                    }
                    break;
        case 2: strcpy( un,"[mW]" );
                    break;
    }
    if ( cralf==1 ) strcpy( un,"[倍]" );
    return div;
}

double disp_value( int s1,int s2,int st,int ed,int p ) /* 最大・最小・平均 */
{
    int i,i2,x,
        vx=430,vy=303;          /* Valuables for displaying values */
    char tmp[80],un[6];
    static double vmax=0,vmin=1e10,vave=0,sum=0,div;

    if ( clrflg==1 && vmax !=0 ) return vmax;    /* リターン? */
    vmax=vave=sum=div=0;
    vmin=1e10;
    /**** 最大値を求める *****/
    for( i2=st ; i2<=ed ; ++i2 )
    {
        for( i=0;i<p;++i ) if ( PD[i2][i]>vmax ) vmax=PD[i2][i];
    }
    setcolor( CYAN );
    div=unit( s1,vmax,un );          /* 単位取得 */
    if ( s2==0 ) outtextxy( g[s1].x1-44, g[s1].y1,un );    /* 単位表示 */
    setcolor( YELLOW );
    sprintf( tmp,"最大値 : %5.2f %s",vmax/div,un );
    outtextxy( vx,vy,tmp );
    line(vx,vy+17,vx+strlen(tmp)+8,vy+17 );

    /**** 最小値を求める *****/
    for( i2=st ; i2<=ed ; ++i2 )
    {
        for( i=0;i<p;++i ) if ( PD[i2][i]>=0 && PD[i2][i]<vmin ) vmin=PD[i2][i];
    }
    div=unit( s1,vmin,un );          /* 単位取得 */
    sprintf( tmp,"最小値 : %5.2f %s",vmin/div,un );
    setcolor( GREEN );
    outtextxy( vx,vy+20,tmp );
    line(vx,vy+20+17,vx+strlen(tmp)+8,vy+20+17 );

    /**** 平均値を求める *****/
    x=0;
    for( i2=st ; i2<=ed ; ++i2 )
    {
        for( i=0;i<p;++i ) if ( PD[i2][i]>=0 ) sum+=PD[i2][i]; else x++;
    }
    vave = sum/(p*((ed-st)+1)-x);    /* 平均値 */
    div=unit( s1,vave,un );          /* 単位取得 */
    sprintf( tmp,"平均値 : %5.2f %s",vave/div,un );
    setcolor( CYAN );
    outtextxy( vx,vy+40,tmp );
    line(vx,vy+40+17,vx+strlen(tmp)+8,vy+40+17 );

    setcolor( BLUE );
    for( i=0 ; i<2 ; i++ ) rectangle vx-2-i,vy-2-i,vx+2+2+1,vy+40+17+1 );
    setcolor( WHITE );
    return vmax;
}

```

```

void value_order(int s1,int i,double vmax,int px,int py,int n) /* オータ調整 */
{
    char tmp[80];
    double vtmp;
    if ( s1 == 1 ) /* 1の単位調整 */
    {
        if ( vmax > 1000 ) vtmp= vmax/1000.;
        if ( vmax > 1e6 ) vtmp= vmax/1e6;
    }
    else vtmp=vmax;
    sprintf(tmp,"%5.2f",vtmp/n+1);
    outtextxy( px,py,tmp );
}

void graph( int s1,int s2,int s3,int p ) /* グラフ描画処理 */
{
    static char data[8][DATA_WD];
    int i,i2,i3,st,ed,
        xsm,ox,oy,cx,cy,dx2,dy2,
        px[7],py[7],dx,dy,
        s4,flg1=WD_OPEN,fg=1,x,flg;
    double vmin,vtmp,vsum;
    static double vmax=0;
    char tmp[80];

    gettext( 1,1,80,25,tmp ); /* テキスト画面保存 */
    while( fg )
    {
        setactivepage( 0 );
        setvisualpage( 0 );
        clrflg=1;
        if ( makesure("グラフのクリア")==1 )
        {
            cleardevice();
            clrflg=0;
        }
        clrscr();
        draw_scale( s1,s2,s3 ); /* スケールを描く */
        switch( s2 )
        {
            case 0: /* 通常タイプのグラフの描画 */
                for( i=0;i<7;++i ) sprintf( data[i]," %sセンサ",tgs[i].name );
                sprintf( data[7]," @ 全センサを対象とする" );
                WD_infol( GP_DISP ).title="プロットするセンサの選択";
                WD_infol( GP_DISP ).cmax=8;
                WD_infol( GP_DISP ).wy=10;
                s4=st=ed=twindow( 1,flg1,&WD_infol( GP_DISP ),data );
                if ( s4==-1 ) /* 描画終了する */
                {
                    twindow( WD_CLOSE );
                    fg=0;
                    break;
                }
                flg1=WD_REMAIN;
                clrscr();
                if ( s4==7 ) /* 全部表示するのか */
                {
                    st=0; /* うん */
                    ed=3;
                }
                vmax=disp_value( s1,s2,st,ed,p ); /* 最大・最小・平均を表示 */
                i2=0; /* 縦の目盛りを入れる */
                for( i=g[s1].y2; i>= g[s1].y1+20 ; i-= (g[s1].y2-g[s1].y1-20.)/10. )
                {
                    if ( i2++ ==0 ) continue;
                    line( g[s1].x1,i,g[s1].x1+4,i );
                    value_order( s1,i2-1,vmax,2,i-8,10 );
                }
                x=i2=0; /* 横の目盛りを入れる */
                for( i=0 ; i<p ; ++i ) if ( PD[st][i]<0 ) x++;
                x--;
                for( i=g[s1].x1; i<= g[s1].x2 ; i+= (g[s1].x2-g[s1].x1)/5. )
                {
                    if ( i2++ ==0 ) continue;
                    line( i,g[s1].y2,i,g[s1].y2-4 );
                    sprintf(tmp,"%d",((int)((p-x)/5.+(i2-1)));
                    outtextxy( i-strlen(tmp)+8/2,g[s1].y2+8,tmp );
                }
                for( i=st ; i<=ed ; ++i ) /* グラフを描く */
                {

```

GRAPH.C

```

xsm=g[s1].x2-g[s1].x1; /* グラフ画面最大幅 */
flg=0;
setcolor( tgs[1].col );
for( i2=0;i2<p;++i2 ) /* グラフのプロット */
{
    if ( PD[i1][i2] <= 0 )
    {
        if ( PD[i1][i2]<-1 && PD[i1][i2]>-15 && i==st ) /* セバレータの処理 ステップ変更
        {
            setcolor( stpcol[ -PD[i1][i2]-1 ] );
            sprintf( tmp,"Xd", (int)(-PD[i1][i2]) );
            outtextxy( g[s1].x1+(xsm/(double)p)*i2+3
                ,g[s1].y1+3,tmp );
            setcolor( CYAN );
            line( g[s1].x1+(xsm/(double)p)*i2+1,g[s1].y1+1
                ,g[s1].x1+(xsm/(double)p)*i2+1,g[s1].y2-1 );
            setcolor( tgs[1].col );
        }
        flg=0;
        continue;
    }
    dx= g[s1].x1+((double)xsm/(p-1))*i2 +1;
    dy= g[s1].y2-((g[s1].y2-g[s1].y1-30.)/vmax)*PD[i1][i2];
    if ( flg==0 ) { flg=1; moveto( dx,dy ); }
    lineto( dx,dy );
}
setcolor( WHITE );
getch();
break;

case 1: /* 3D表示の場合 */
vmax=disp_value( s1,s2,0,3,p ); /* 最大・最小・平均値表示 */
i2=0; /* Z軸の目盛りを入れる */
for( i=g[s1].y2; i>= g[s1].y1+70 ; i-= (g[s1].y2-g[s1].y1-70.)/5. )
{
    if ( i2++ ==0 ) continue;
    line( g[s1].x1,i,g[s1].x1+4,i );
    value_order( s1,i2-1,vmax,2,i-8,5 );
}
i2=0; /* X軸の目盛りを入れる */
for( i=g[s1].x1; i<= g[s1].x2-40 ; i+= (g[s1].x2-g[s1].x1-40.)/5. )
{
    if ( i2++ ==0 ) continue;
    line( i,g[s1].y2,i,g[s1].y2-4 );
    value_order( s1,i2-1,vmax,i-20,g[s1].y2+4,5 );
}
i2=0; /* y軸の目盛りを入れる */
ox=80.*cos(angle*M_PI/180.);
oy=80.*sin(angle*M_PI/180.);
for( i=0 ; i<= 80 ; i+= 80./3. )
{
    if ( i2++ ==0 ) continue;
    i2--;
    line( (int)(g[s1].x1+ox/3.*i2),(int)(g[s1].y2-oy/3.*i2),
        (int)(g[s1].x1+ox/3.*i2+4),(int)(g[s1].y2-oy/3.*i2));
    value_order( s1,i2,vmax,g[s1].x1+ox/3.*i2+6
        ,g[s1].y2-oy/3.*i2-8,3 );
    i2++;
}
/*==== グラフを描く =====*/
flg=0;
setcolor( stpcol[s2] );
for( i=0;i<p;++i ) /* グラフのプロット */
{
    if ( PD[0][i]<0 )
    {
        if ( PD[0][i]<-1 && PD[0][i]>-8 )
        {
            setcolor( stpcol[ -PD[0][i]-1 ] );
        }
        flg=0;
        continue;
    }
    dx= g[s1].x1+((double)(g[s1].x2-g[s1].x1-40.)/vmax)*PD[TGS1][i]
        +(80./vmax)*PD[TGS2][i]*cos(angle*M_PI/180.);
    dy= g[s1].y2-((double)(g[s1].y2-g[s1].y1-70.)/vmax)*PD[TGS3][i]
        -(80./vmax)*PD[TGS2][i]*sin(angle*M_PI/180.);
    if ( dx==g[s1].x1 && dy==g[s1].y2 ) continue;
}

```

```

        if ( flg==0 ) ( flg=1; moveto( dx,dy ); )
        lineto( dx,dy );
    )
    setcolor( WHITE );
    getch();
    fg=0;
    break;
case 2: /*多角形表示の場合 */
    vmax=disp_value( s1,s2,0,3,p ); /* 最大・最小・平均値を表示 */
    px[0]= g[s1].x2-20+2; /* 0 */
    py[0]= g[s1].y1+(g[s1].y2-g[s1].y1)/2.+20;
    px[1]= g[s1].x1+(g[s1].x2-g[s1].x1)/2.-20; /* 90 */
    py[1]= g[s1].y1+20;
    px[2]= g[s1].x1-20-2; /* 180 */
    py[2]= g[s1].y1+(g[s1].y2-g[s1].y1)/2.+20;
    px[3]= g[s1].x1+(g[s1].x2-g[s1].x1)/2.-20; /* 270 */
    py[3]= g[s1].y2+20;
    oy= g[s1].y1+(g[s1].y2-g[s1].y1)/2.+20; /* center */
    ox= g[s1].x1+(g[s1].x2-g[s1].x1)/2.-20;

    setcolor( tgs[0].col );
    outtextxy( px[0]-16,py[0]-20,"TGS1" );
    setcolor( tgs[1].col );
    outtextxy( px[1]-40,g[s1].y1+10,"TGS2" );
    setcolor( tgs[2].col );
    outtextxy( px[2]-16,py[2]-20,"TGS3" );
    setcolor( tgs[3].col );
    outtextxy( px[3]-40,g[s1].y2+10,"TGS4" );
    setcolor( WHITE );
    i2=0;
    for( i=ox; i<=px[0] ; i+= (px[0]-ox)/3. )
    {
        if ( i2++ ==0 ) continue;
        line( i,py[0]-2,i,py[0]+2 );
        value_order( s1,i2-1,vmax,i-20,py[0]+4,3 );
    }
    i2=0; /* 180 */
    for( i=ox; i>=px[2] ; i-= (ox-px[2])/3. )
    {
        if ( i2++ ==0 ) continue;
        line( i,py[2]-2,i,py[2]+2 );
        value_order( s1,i2-1,vmax,i-20,py[2]+4,3 );
    }
    i2=0; /* 90 */
    for( i=oy; i>=py[1] ; i-= (oy-py[1])/3. )
    {
        if ( i2++ ==0 ) continue;
        line( px[1]-2,i,px[1]+2,i );
        value_order( s1,i2-1,vmax,px[1]+2,i-8,3 );
    }
    i2=0; /* 270 */
    for( i=oy; i<=py[3] ; i+= (py[3]-oy)/3. )
    {
        if ( i2++ ==0 ) continue;
        line( px[3]-2,i,px[3]+2,i );
        value_order( s1,i2-1,vmax,px[3]+2,i-8,3 );
    }
    /* グラフを描く */
    setcolor( stpcol[s3] );
    for( i=0;i<p;+i )
    {
        if ( PD[0][i]<0 ) /* 節目 */
        {
            setcolor( stpcol[ -PD[0][i]-1 ] );
            continue;
        }
        flg=0;
        for( i2=0;i2<7/4;+i2 )
        {
            dx= ox+(px[i2]-ox)/vmax*PD[i2][i];
            dy= oy+(py[i2]-oy)/vmax*PD[i2][i];
            if ( flg==0 )
            {
                flg=1;
                moveto( dx,dy );
                cx=dx;
                cy=dy;
            }
            lineto( dx,dy );
        }
    }
}

```

```

        lineto( cx,cy ); /* 線の接続 */
        if ( kbhit() ) break;
    }
    getch();
    fg=0;
    break;
case 3: /* 出力比表示の場合 */
    setcolor( CVAN );
    outtextxy( g[s1].x1-44, g[s1].y1+4, "[Y]" );
    setcolor( WHITE );
    if ( clrflg==0 :: vmax==0 )
    {
        if ( PD[0][0] < 0 ) flg=0; else flg=1;
        vmax=0;
        for( i=0 ; i<p ; ++i ) /* 割合の最大値を求める */
        {
            if ( PD[0][i]<0 )
            {
                if( PD[0][i]== -5 ) flg=1; else flg=0;
            }
            vsum=0;
            for( i2=0 ; i2<4 ; ++i2 ) vsum+= PD[i2][i]; /* 局部分計 */
            for( i2=0 ; i2<4 ; ++i2 )
            {
                if ( vsum > 0 && flg )
                {
                    if ( PD[i2][i]/vsum>vmax && PD[i2][i]/vsum<=1. )
                    {
                        vmax= PD[i2][i]/vsum;
                    }
                }
            }
        }
    }
    i2=0; /* 縦の目盛りを入れる */
    for( i=g[s1].y2; i>= g[s1].y1+30 ; i-= (g[s1].y2-g[s1].y1-30.)/10. )
    {
        if ( i2++ ==0 ) continue;
        line( g[s1].x1,i,g[s1].x1-2,i );
        value_order( -1,i2-1,vmax*100,2,i-8,10 );
    }
    x=i2=0; /* 横の目盛りを入れる */
    for( i=0 ; i<p ; ++i ) if ( PD[0][i]<0 ) x++;
    x--;
    for( i=g[s1].x1; i<= g[s1].x2 ; i+= (g[s1].x2-g[s1].x1)/5. )
    {
        if ( i2++ ==0 ) continue;
        line( i,g[s1].y2,i,g[s1].y2+2 );
        sprintf( tmp,"%d",(int)((p-x)/5.*(i2-1)));
        outtextxy( i-strlen(tmp)+8/2,g[s1].y2+8,tmp );
    }
    /* 出力比グラフを描く */
    for( i=0 ; i<4 ; i++ )
    {
        flg=0;
        for( i3=0;i3<p;++i3 )
        {
            vsum=0;
            for( i2=0;i2<4;++i2 ) vsum+=PD[i2][i3]; /* 合計 */
            if ( PD[0][i3]<-1 && PD[0][i3]>-15 && i==0 ) /* ステップ表示 */
            {
                setcolor( stpcol[ -PD[0][i3]-1 ] );
                sprintf( tmp,"%d",(int)(-PD[0][i3]) );
                outtextxy( dx+3,g[s1].y1+8,tmp );
            }
            dx= g[s1].x1+(double)(g[s1].x2-g[s1].x1)/(p-1)*i3+1;
            if ( ( PD[i][i3]<0 && i==0 ):: vsum==0 )
            {
                setcolor( LIGHTGRAY );
                line( dx,g[s1].y1+31,dx,g[s1].y2-1 );
                flg=0;
                continue;
            }
            if ( PD[i][i3]<0 ) continue;
            dy= g[s1].y2-(double)(g[s1].y2-g[s1].y1-30)*(PD[i][i3]/vsum)/vmax;
            if ( flg==0 )
            {
                moveto( dx,dy );
                flg=1;
            }
        }
    }

```

```

        setcolor( tgs[i].col );
        lineto( dx,dy );
    )
    }
    getch();
    fg=0;
    break;
    default: fg=0; break;
}
)
puttext( 1,1,80,25,tst );
setvisualpage( 1 );
setactivepage( 1 );
return;
}

void data_set( char data[][DATA_WD] )
{
    int i;
    for( i=0; i<4; ++i )
    {
        sprintf( data[i], " X-14sセンサの重み(0-100%%): %3d[%%]", tgs[i].name, ratio[i] );
    }
    return;
}

void value_set( int num, WINDOW *wd, char data[][DATA_WD] )
{
    int i, i2, s, c;

    while( 1 )
    {
        wd->cursor_on=0;
        wd->inp_mode=1;
        data_set( data );
        data[num][1]='s';
        twindow( 1, WD_REMAIN, wd, data );
        switch( c=key_in( 1, 0 ) )
        {
            case 8: if ( ratio[ num ] > 9 ) ratio[ num ]-=5;
                    break;
            case 12: if ( ratio[ num ] < 96 ) ratio[ num ]+=5;
                    break;

            case 13:
            case 27: data[num][1]=' ';
                    wd->cursor_on=1;
                    wd->inp_mode=2;
                    return;
            default: break;
        }
    }
}

/*
WINDOW WD_info( OMOMI )=( 9,8,52,6,T_CYAN,T_YELLOW,T_WHITE,0,
                           "センサ重みづけ",2,4,0,0,1,0,0 );
*/
void onomi( void )
{
    static char data[6][DATA_WD];
    int i,s,c,flg;
    char tmp[DATA_WD],tmp2[80];

    WD_info( OMOMI ).title="センサ重みづけ設定";
    WD_info( OMOMI ).wx=52;
    setmem( data, DATA_WD*6, 0 );
    data_set( data );
    flg=WD_OPEN;
    while( 1 )
    {
        s=twindow( 1, flg, &WD_info( OMOMI ), data );
        flg=WD_REMAIN;
        switch( s )
        {
            case -1: twindow( WD_CLOSE );
                    return;
            case 0: value_set( 0, &WD_info( OMOMI ), data );
                    break;
            case 1: value_set( 1, &WD_info( OMOMI ), data );

```

```

        break;
    case 2: value_set( 2,&WD_info[ OMOMI ],data );
        break;
    case 3: value_set( 3,&WD_info[ OMOMI ],data );
        break;
    default: break;
    }
}

void draw_graph( void )      /**** 測定データよりグラフを描画する ****/
{
    static char data1[7][DATA_WD]={
        " 電圧-時間",
        " 抵抗-時間",
        " 消費電力-時間",
        " オプション",
        " オプション",
        " オプション",
        " オプション"
    },
    data2[6][DATA_WD]={
        " 通常グラフ表示",
        " 3次元表示",
        " 多角形表示",
        " 出力比表示",
        " オプション",
        " オプション"
    },
    data3[15][DATA_WD];

    FILE *fp;
    int i,i2,i3,flg1,flg2,flg3,s1,s2,s3,st,ed,ufg1,ufg2,ufg3,p;
    char dirtmp[80];
    double rs,v0[7],vmin;

    for( i=0;i<14;++i ) /* フェーズ選択メニュー用データ設定 */
    {
        sprintf( data3[i],"Step %2d: ヒータ ON/OFF 間隔 %3d(Sec)",
            i+1,SM[i].interval );
    }
    sprintf( data3[14],"◎ 全ステップをグラフ表示する" );
    while( 1 )
    {
        files("グラフ出力用データの選択");      /* データファイルの選択 */
        twindow( WD_CLOSE );
        if ( cfn == -1 ) return;
        if ( !strlen( fn(cfn).fname ) ) continue;
        sprintf( dirtmp,"%s\\VRL\\%s.VRL",DIR,fn(cfn).fname);
        if ( (fp=fopen( dirtmp,"rb" )) == NULL ) /* ファイル読み込めず? */
        {
            mesdisp( T_RED,"データを読み込めません！");
            continue;
        }
        fread( V, sizeof( double )*7000 , 1, fp );      /* 電圧データ読み込み */
        fclose( fp );
        if ( makesure("プロットデータの補正")==1 )
        {
            putchar( '?' );
            V_rectify( cfn );
        }
        setmem( PD,sizeof( double )*7000,0 );
        flg1=WD_OPEN;
        while( 1 )
        {
            /**** グラフメニュー ****/
            WD_info[ GP_MENU ].title="パラメータ";
            if ( (s1=twindow( 1,flg1,&WD_info[ GP_MENU ],data1 ))!=-1 )
            {
                twindow( WD_CLOSE );
                break;
            }
            flg1=WD_REMAIN;
            if ( s1 > 2 ) /* サポートしていないパラメータを選択 */
            {
                mesdisp( T_RED,"表示するグラフのパラメータ形式をまだサポートしていません。");
                continue;
            }
            if ( makesure("センサの重みづけ設定")==1 ) omomi(); /* ガスセンサ重みづけ */

            flg2=WD_OPEN;
            while( 1 )
            {
                /**** グラフの種類選択 ****/

```



```

WD_info[ GP_TYPE ].title="グラフの種類";
if ( (s2=twindow( 1,flg2,&WD_info[ GP_TYPE ],data2 ))== -1 )
{
    twindow( WD_CLOSE );
    break;
}
flg2=WD_REMAIN;
if ( s2 > 3 ) /* サポートしていないグラフを選択 */
{
    nesdisp( T_RED,"表示するグラフの表示形式をまだサポートしていません。" );
    continue;
}
flg3=WD_OPEN;
while( 1 )
{
    /*==== グラフの処理範囲 ====*/
    WD_info[ GP_STEP ].title="表示範囲の選択";
    WD_info[ GP_STEP ].cmax=15;
    if ( (s3=twindow( 1,flg3,&WD_info[ GP_STEP ],data3 ))== -1 )
    {
        twindow( WD_CLOSE );
        break;
    }
    flg3=WD_REMAIN;

    /*==== グラフプロット用データ範囲の決定 ====*/
    st=ed=0;
    ufg1=0;
    for( i=0;i<1000;++i )
    {
        if ( V[0][i]== -s3-1 ) /* 該当するステップはあるか */
        {
            ufg1=1;
            st=i+1; /* データスタート */
        }
        else if ( V[0][i] < 0 && ufg1 ) /* データ終了か? */
        {
            ed=i-1;
            break;
        }
        if ( s3==14 /*10*/ && V[0][i]== -999 ) /* 全データの場合 */
        {
            ed=i; /* 測定終了フラグ発見 */
            break;
        }
    }
    if ( st==0 && ed==0 ) ed=st<2 ) /* 該当するデータがなかったとき */
    {
        nesdisp( T_RED,"該当するステップのデータはありませんでした。" );
        continue;
    }

    /*==== 清浄大気を基準にして表示するか ====*/
    crair=0;
    for( i=0;i<7/4*4*4;++i ) v0[i]=1.;
    if ( makesure("清浄大気基準表示")==1 )
    {
        for( i=0;i<7/4*4*4;i++ )
        {
            ufg1=0;
            for( i2=0;i2<1000;++i2 )
            {
                if ( V[i][i2]== -3 && ufg1==0 ) ufg1=i2;
                if ( V[i][i2]== -4 && ufg1 )
                {
                    /* 清浄気の最小値 */
                    vmin=100;
                    for( i3=ufg1+1 ; i3<i2 ; ++i3 )
                    {
                        if ( V[i][i3] < vmin && V[i][i2]>=0 ) vmin=V[i][i3];
                    }
                    v0[i]=vmin;
                    break;
                }
            }
        }
        crair=1;
    }
}
/*==== プロット用データの作成 ====*/
p=0;

```

```

for( i2=st ; i2<ed ; ++i2 )
{
    for( i=0 ; i<7/4*7 ; ++i )
    {
        if ( V[i][i2]>0 ) rs= ( 10./(V[i][i2]*(ratio[i]/100.)/v0[i]) -1 )*tgs[i].rl;
        else {
            if ( V[i][i2] !=0 )
            {
                /* セバレータ発見 */
                for( i3=0; i3<7/4*7; ++i3 ) PD[i3][p]=V[i3][i2];
                p++;
                break;
            }
            rs=0;
        }
        switch( s1 )
        {
            case 0 : /* 電圧-時間 */
                PD[i][p]= V[i][i2]*(ratio[i]/100.)/v0[i];
                break;

            case 1 : /* 抵抗-時間 */
                PD[i][p]= rs;
                break;

            case 2 : /* 消費電力(mW)-時間 */
                PD[i][p]= 1000.* 10*10*rs/((rs+tgs[i].rl)*(rs+tgs[i].rl));
                break;
            default: break;
        }
    }
    p++;
}
graph( s1,s2,s3,p ); /* グラフの描画 */
} /* GP_STEP */
} /* GP_TYPE */
} /* GP_MENU */
} /* while */
return;
}

```

```

void date_set( char *nichizi ) /* 日付・日時データの獲得 */
{
    struct date *dt=NULL;
    struct time *tm=NULL;
    char tmp[50],year[10],mon[10],day[10],min[10],hour[10],sec[10];
    getdate( dt );
    itoa( (int)dt->da_year,year,10 );
    itoa( (int)dt->da_mon,mon,10 );
    itoa( (int)dt->da_day,day,10 );
    setmen( tmp,50,0 );
    gettime( tm );
    itoa( (int)tm->ti_hour,hour,10 );
    itoa( (int)tm->ti_min,min,10 );
    itoa( (int)tm->ti_sec,sec,10 );
    sprintf( tmp,"%4s年 %2s月 %2s日 %2s時 %2s分 %2s秒"
            ,year,mon,day,hour,min,sec );
    strcpy( nichizi,tmp );
    return;
}

```

```

void V_rectify( int cn ) /* 温度、湿度補正を行う */
/* あらかじめPDは設定されているものとして処理 */
/* cn means file number */
{
    struct RECTIFY {
        double a1,a2;
        double b1,b2;
    };
    static struct RECTIFY R[3]=
    {
        { -0.003846,1.035 , -0.008462,2.296 },
        { 0.002 ,0.4 , -0.004 ,1.9 },
        { 0.0045 ,0.15 , -0.006 ,1.85 }
    };
    static double vpu[6]={ 0.8, 0.8, 1.0, 1.0, 0.9, 0.9 }
    ,vpd[6]={ 1.2, 1.2, 1.0, 1.0, 1.1, 1.1 };
    double standard_a,standard_b,standard_r,a,b,r,Rx,Rv,Rst2;
    int i,i2,i3,flg,dx,dy,dx2,dy2;
    char tmp[40];
}

```

```

if ( hose1 )
{
    for( i=0; i<3 ;++i )
    {
        line( 100,0,100,400 );          /* スケールを描く */
        line( 100,200,500,200 );
        outtextxy( 70,191,"1.0" );
        outtextxy( 510,191,"Temperature('C)" );
        sprintf( tmp,"%14sセンサ温度特性",tgs[i].name );
        outtextxy( 260,3,tmp );
        line( 100,200-log( 0.5 )+200,105,200-log( 0.5 )+200 );
        outtextxy( 70,192-log( 0.5 )+200,"0.5" );
        line( 100,200-log( 1.5 )+200,105,200-log( 1.5 )+200 );
        outtextxy( 70,192-log( 1.5 )+200,"1.5" );
        line( 100,200-log( 2.0 )+200,105,200-log( 2.0 )+200 );
        outtextxy( 70,192-log( 2.0 )+200,"2.0" );
        settxtstyle( DEFAULT_FONT,VERT_DIR,1 );
        outtextxy( 60,152,"log( Rs/Ro )" );
        settxtstyle( DEFAULT_FONT,HORIZ_DIR,1 );

        for( i2=0; i2<100 ;i2+=10 )
        {
            a= R[i].a1+i2+R[i].a2;
            b= R[i].b1+i2+R[i].b2;
            flg=0;
            for( i3=0; i3<30 ;++i3 )
            {
                r= -(a/25.)*i3+b;
                dx=100+i3*400/29;
                dy=200-log(r)*200;

                if( i3%2 ==0 )
                {
                    line( dx,200,dx,205 ); /* 温度メモリを入れる */
                    sprintf( tmp,"%2d",i3 );
                    outtextxy( dx-8,200,tmp );
                }
                if ( flg==0 )
                {
                    dx2=dx;
                    dy2=dy;
                    flg=1;
                }
                line( dx,dy,dx2,dy2 );
                dx2=dx;
                dy2=dy;
            }
        }
        getch();
        cleardevice();
    }
}
for( i=0; i<3 ;++i )
{
    standard_a= R[i].a1*fn[cn].humid+R[i].a2;
    standard_b= R[i].b1*fn[cn].humid+R[i].b2;
    standard_r= -(standard_a/25.)*20.+standard_b;

    a= R[i].a1*fn[cn].humid+R[i].a2;
    b= R[i].b1*fn[cn].humid+R[i].b2;
    r= -(a/25.)*fn[cn].temp+b;

    /*
    r *= fn[cn].temp/20.;
    */

    Rx= standard_r/r;          /* 修正比率(to 20C,65%) */

    for( i2=0; i2<1000 ;++i2 )
    {
        if ( V[i][i2]>0 )
        {
            Rst2=( 10./V[i][i2]-1. )*tgs[i].r1;
            Rv= ( Rst2+tgs[i].r1 )/( Rst2+Rv+tgs[i].r1 );
            V[i][i2] *= Rv;
        }
    }
    gotoxy( 1,1+i );cprintf( "%f",Rv);
}

```

FILE.C

```

(
    sprintf( dirtmp, "%s\\%v\\%v\\%s.vr1", DIR, fn[s1].fname );
    if ( (fp=fopen(dirtmp, "rb")) != NULL )
    {
        fread( V, sizeof( double ) * 7000, 1, fp );
        sprintf( dirtmp, "%s\\%v\\%v\\%s.vr1", DIR, fn[s2].fname );
        if ( (fp2=fopen(dirtmp, "wb")) != NULL )
        {
            fwrite( V, sizeof( double ) * 7000, 1, fp2 );
            strcpy( cptmp, fn[s2].fname ); /* ファイル名の待選 */
            memcpy( &fn[s2], &fn[s1], FN_SIZE ); /* パラメータのコピー */
            strcpy( fn[s2].fname, cptmp ); /* ファイル名設定 */
            fclose( fp2 );
        } else mesdisp( T_RED, "複写先のファイルが作成できません！" );
        fclose( fp );
    } else mesdisp( T_RED, "複写元のファイルを読み込めません！" );
}
twindow( WD_CLOSE ); /* files window close */
twindow( WD_CLOSE );
return;
)

void rename_file( void ) /**** ファイル名の変更 ****/
(
    static char data[DATA_WD];
    int i, s, s2;
    char old[80], new[80];

    s = files( "名前を変更するデータファイルの選択" );
    if ( s == -1 )
    {
        twindow( WD_CLOSE );
        return;
    }
    setmem( data, DATA_WD, 0 );
    if ( get_fname( "新しいファイル名の設定", data ) == -1 )
    {
        twindow( WD_CLOSE );
        return;
    }
    if ( makesure( "データファイル名変更" ) == 1 )
    {
        sprintf( old, "%s\\%v\\%v\\%s.vr1", DIR, fn[s].fname );
        sprintf( new, "%s\\%v\\%v\\%s.vr1", DIR, data );
        if ( rename( old, new ) != 0 ) mesdisp( T_YELLOW, "ファイル名の変更ができません！" );
        else strcpy( fn[s].fname, data );
    }
    twindow( WD_CLOSE );
    return;
)

void move_file( void ) /**** ファイル名の移動 ****/
(
    int s, d;
    char tmp[DATA_WD];

    s = files( "移動するファイル名の選択" );
    twindow( WD_CLOSE );
    if ( s == -1 ) return;

    d = files( "移動先の選択" );
    if ( d == -1 )
    {
        twindow( WD_CLOSE );
        return;
    }
    if ( makesure( "データファイル名の移動" ) == 1 && s != d )
    {
        memcpy( &fn[d], &fn[s], FN_SIZE ); /* パラメータのコピー */
        setmem( &fn[s], FN_SIZE, 0 ); /* 移動もとのクリア */
        mesdisp( T_WHITE, "データファイル名の移動を行いました。" );
    }
    twindow( WD_CLOSE );
    return;
)

/*
WINDOW WD_info[ HD_FILE ] = ( 36, 2, 34, 6, T_CYAN, T_YELLOW, T_WHITE, 0, "ファイル処理",
                               2, 4, 0, 0, 1, 1, 0 );
*/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
#include <dir.h>
#include <time.h>
#include <bios98.h>
#include "def.h"
#include "quo.h"
#include "sample.h"

extern char txt(8000);

/* Define constants for 16ch Relay */

#define HEATER_TGS1 0
#define HEATER_TGS2 1
#define HEATER_TGS3 2
#define HEATER_TGS4 3
#define HEATER      4
#define MOTOR1      5
#define MOTOR2      6
#define FAN          7

/* Define constants for 8ch A/D converter */

#define TGS1         1
#define TGS2         2
#define TGS3         3
#define TGS4         4
#define TEMP1        5
#define TEMP2        6
#define HYGRO        7

/*
/* ボードアドレス設定値を格納する変数 */
/*
/*
unsigned int select_prt=AD_SELECT;
unsigned int low_data_prt=AD_LOW;
unsigned int start_prt=AD_START;
unsigned int high_data_prt=AD_HIGH;

unsigned int prt_0=RELAY_0;
unsigned int prt_1=RELAY_1;

unsigned int rly_state; /* リレーの状態記憶変数 */
int dc,dc2;             /* data counter */

double V(7)(1000);      /* 測定データ保存用配列 */

#define bar_x 4
#define bar_y 180
#define gr_x 180
#define gr_y 180

#define env_x 46*8
#define env_y 1*8

/**** 関数群プロトタイプ ****/
void save();
void HEATER_TGS_voltage();
void circuit_voltage();
void fan_voltage();
void dspvalue();
void flame_bar();
void flame_clr();

static unsigned int rd[16]={ 0,0,0,0,1,1,1,1, 0,0,0,0,0,0,0,0 };
/* リレー状態格納用配列 */

void mes_print( int col,char *string ) /* メッセージの表示 */
{
    textcolor( col );
    gotoxy( 1,1 );
    cprintf("
    gotoxy( 1,1 );
    cprintf( string );
    textcolor( T_WHITE );

```



```

void flame_clr( void )
{
    int i;

    setfillstyle( EMPTY_FILL, BLACK );
    cleardevice();
    /*
    bar( bar_x, bar_y , bar_x+96, bar_y+205 );
    bar( gr_x, gr_y , gr_x+450, gr_y+205 );
    bar( gr_x-30, gr_y-8, gr_x, gr_y+205 );
    bar( 320, env_y , env_x+260, env_y+160 );
    */
    for( i=0; i<4 ; ++i )
    {
        gotoxy( 13, i*4+12 );
        cprintf( "      " );
    }
    for( i=0; i<3 ; ++i )
    {
        gotoxy( 36, (i+1)*3 );
        printf( "      " );
    }
    gotoxy( 1, 1 );
    cprintf( "      " );

    return;
}

void flame_bar( void ) /* 棒グラフ表示用のフレームを描く */
{
    int i, c, n;
    char tmp[10];
    setcolor( CYAN );
    rectangle( bar_x, bar_y , bar_x+96, bar_y+205 );
    rectangle( gr_x, gr_y , gr_x+450, gr_y+205 );
    rectangle( env_x-1, env_y+16 , env_x+261, env_y+160-16 );

    n=0;
    for( i=env_x-1 ; i<env_x+262 ; i+= (int)262/4. )
    {
        sprintf( tmp, "X2d", n );
        n+= 10;
        outtextxy( i-8, env_y-3, tmp );
        line( i, env_y+13, i, env_y+16 );
    }
    n=0;
    for( i=env_x-1 ; i<env_x+262 ; i+= (int)262/10. )
    {
        sprintf( tmp, "X3d", n );
        n+= 10;
        outtextxy( i-16, env_y+160-12, tmp );
        line( i, env_y+160-13, i, env_y+160-16 );
    }
    c=100;
    for( i=0; i<200 ; i+= 200/10. ) /* 割合表示 */
    {
        setcolor( GREEN );
        line( bar_x+1, bar_y+5+i, bar_x+96-1, bar_y+5+i );
        line( gr_x+1, gr_y+5+i, gr_x+450-1, gr_y+5+i );
        sprintf( tmp, "X3d", c );
        settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
        setcolor( CYAN );
        outtextxy( gr_x-30, gr_y+i-8, tmp );
        line( gr_x, gr_y+i , gr_x-2, gr_y+i );
        c-=10;
    }
    settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
    return;
}

/* 各センサ出力比を棒グラフにして表示 */
void draw_bar( int t, unsigned int v[] )
{
    static int x2, y2[4];
    unsigned int sum=0;
    int i, i2, x, y;
    double r[4];

    /* 棒グラフを描く */
    for( i=0 ; i<3 ; i++ )

```

```

(
    setfillstyle( EMPTY_FILL, BLACK );
    bar( env_x, env_y+1+48+26, env_x+260, env_y+1+48+26+6 );

    switch( i )
    {
        case 0:
        case 1: setfillstyle( SOLID_FILL, YELLOW );
                bar( env_x, env_y+1+48+26+4
                    , env_x+(int)(get_temp(i+1)*260./400.), env_y+1+48+26+6 );
                gotoxy( 35, 3+(i==1)*3 );
                cprintf( "気温 %d %4.1f°C", i+1, get_temp(i+1)/10. );
                break;

        case 2: setfillstyle( SOLID_FILL, RED );
                bar( env_x, env_y+1+48+26+4
                    , env_x+(int)(get_humid()*260./100.), env_y+1+48+26+6 );
                gotoxy( 35, 9 );
                cprintf( "湿度 %4.1f%", get_humid() );
                break;

        default: break;
    }
}
setcolor( DARKGRAY );
x=gr_x+(int)(t % 450)+1;
line( x, gr_y+1, x, gr_y+204 );

for( i=0; i<4 ; ++i )
{
    r[i]= (double)v[i]/4089.*10.;

    /* 棒グラフを消す */
    setfillstyle( EMPTY_FILL, BLACK );
    bar( bar_x+1+24+4, bar_y+1, bar_x+1+24+16, bar_y+204 );

    /* 棒グラフを描く */
    setfillstyle( SOLID_FILL, tgs[i].col );
    bar( bar_x+1+24+4, bar_y+204-(int)(r[i]*20)
        , bar_x+1+24+16, bar_y+204 );
    gotoxy( 13, i+4+12 );
    cprintf( "X5.2fV", r[i] );

    /* データプロットしてゆく */
    x=gr_x+(int)(t % 450)+1;
    y=gr_y+204-(int)(r[i]*20);
    setcolor( tgs[i].col );
    if ( (t % 450)==0 )
    {
        y2[i]=y;
        x2=x;
    }
    line( x, y, x2, y2[i] );
    setcolor( WHITE );
    y2[i]=y;
}
x2=x;
return;
}

void graph_bar() /* called from main() */
{
    char c;
    int t;
    unsigned int r[7], sum;

    flame_bar();
    t=0;
    while( 1 )
    {
        Ginp( r );
        draw_bar( t, r );
        t++;
        if ( t > 450 ) t=0;
        if ( kbhit() )
        {
            c=getch();
            if ( c=='h' )
            {
                HEATER_TGS_voltage( ON ); /* ヒータ電圧ON/OFF */
            }
        }
    }
}

```


GAS.C

```

        { 24 , 4 },      /* 12 */
        { 27 , 4 },      /* 13 */
        { 30 , 4 } );    /* 14 */

void smell( void ) /* 匂い測定ルーチン */
{
    int i,i2,i3,l,dl;
    unsigned int tmp;
    long btime,ptime,ptime2;

    dc=dc2=0; /* data counter */
    dl=0;

    memset( V,sizeof( double )*7000, 0 ); /* 測定データ用配列クリア */

    rly_state &= 0xff0f;
    port_data_output( 0,rly_state ); /* TGS Heater voltage OFF */

    flame_bar();

    for( i=0 ; i<14 ; ++i )
    {
        putchar( 7 );
        putchar( 7 );
        putchar( 7 );
        for( l=0 ; l<7 ; ++l ) V[ l ][ dc ] = -(i+1); /* 仕切りフラグ */
        gotoxy( 1,3 );
        textcolor( T_CYAN );
        cprintf(">> Procedure number %2d/14",i+1 );
        gotoxy( 1,9 );
        cprintf(">> Switching Interval %3d",SM[ i ].interval );
        dc++;

        for( i2=0 ; i2 < SM[ i ].n ; ++i2 )
        {
            rly_state ^= 0x00f0; /* Reverse */
            port_data_output( 0,rly_state );

            secinit();
            time( &btime );

            do {
                secinit();
                time( &ptime2 );
                if ( kbhit()!=0 ) /* Interrupt */
                {
                    getch();
                    putchar( 7 );
                    goto RET;
                }
                read_data();
                dspvalue( dl++ );
                while( time( &ptime )==ptime2 );

            } while( (long)(ptime - btime) < (long)SM[ i ].interval );
        }
    } /* 906 seconds */

RET:
    for( i=0;i<7;++i ) V[i][dc]= -999; /* 測定終了フラグを立てる */
    flame_clr();
    rly_state |= 0x00f0;
    port_data_output( 0,rly_state ); /* TGS Heater voltage ON */

    return;
}

/* ===== */

void experiment( void ) /* 匂い測定手順手続きルーチン */
{
    int i,i2,step;
    char date1[80],date2[80];

    long btime,ptime;

    gettext( 1,1,80,25,tst ); /* Save Screen of Current State */
    clrscr();

    HEATER_TGS_voltage( ON ); /* 半導体ガスセンサのヒータ ON */

```

```

date_set( datel );                                /* 実験開始日時の獲得 */

for( step = 1; step < 8 ; ++step )
{
    putchar( 7 );
    if ( kbhit()!=0 )          /* Interrupt */
    {
        getch();
        putchar( 7 );
        break;
    }
    switch( step )
    {
        case 1: /* 内気と外気との入れ替え */
            mes_print(T_WHITE, "・ただ今、内気と外気の入替えを行っています");
            gate_control1( OFF );      /* Open gate No.1 */
            gate_control2( OFF );      /* Open gate No.2 */
            FAN_voltage( ON );         /* Drive Ventilator */

            flame_bar();
            for( i=0 ; i<450 ; ++i )
            {
                if ( kbhit()!=0 )      /* Interrupt */
                {
                    getch();
                    putchar( 7 );
                    break;
                }
                dspvalue( i );
                delay( 150 );
            }
            flame_clr();

            FAN_voltage( OFF );
            break;

        case 2: /* シリカゲル、活性炭による除湿、脱臭、気温・湿度などの調整 */
            mes_print(T_YELLOW, "・ただ今、内気の調整作業を行っています。");
            gate_control1( ON );        /* Close gate No.1 */
            FAN_voltage( ON );         /* Drive Ventilator */

            flame_bar();
            dc=0;
            time( &btime );
            do {
                if ( kbhit()!=0 )      /* Interrupt */
                {
                    getch();
                    putchar( 7 );
                    flame_clr();
                    break;
                }
                dspvalue( dc++ );
                heater_control();
                time( &ptime );
            } while ( get_humid() > 35 :: (ptime-btime)< 100 );

            FAN_voltage( OFF );
            HEATER_voltage( OFF );

            mes_print(T_YELLOW, "・ただ今、ファンの回転が収まるのを待っています。");
            sleep( 20 );               /* Waiting the fan to stop */
            gate_control2( ON );       /* Close gate No.2 */
            break;

        case 3: /* ガス発生 */
            gotoxy( 1,2 );
            cprintf(" ");

            gas_vaporizer();           /* vaporize sample */
            gas_stabilizer();          /* coordinate temperature */
            break;

        case 4: /* 内気循環 */
            mes_print( T_GREEN, "・ただ今、測定槽内のガスの循環を行っています");
            flame_bar();
            dc=0;
            for( i=1 ; i<12 ; ++i )    /* stir the gas inside */
            {
                FAN_voltage( ON );
                for( i2=0 ; i2<20 ; ++i2 )
            }

```

```

        dspvalue( dc++ );
        delay( 100 );
    }
    FAN_voltage( OFF );
    for( i2=0 ; i2<17 ; ++i2 )
    {
        dspvalue( dc++ );
        delay( 100 );
    }
    }
    flame_clr();

    break;
case 5: /* 測定 */
    mes_print(T_WHITE, "・ただ今、匂い測定中です。");
    smell(); /* call smell() routine */
    break;
case 6: /* 内気と外気との入れ替え */
    mes_print(T_WHITE, "・ただ今、内気と外気の入替えを行っています");
    gate_control1( OFF ); /* Open gate No.1 */
    gate_control2( OFF ); /* Open gate No.2 */
    FAN_voltage( ON ); /* Drive Ventilator */
    HEATER_voltage( OFF );

    flame_bar();
    for( i=0 ; i<450 ; ++i )
    {
        dspvalue( i );
        delay( 200 );
    }
    flame_clr();

    FAN_voltage( OFF );
    break;
case 7: /* ファイル名の保存 */
    mes_print(T_WHITE, "・すべての測定行程が終了しました");
    while( kbhit()==0 ) putchar( 7 );
    getch();

    date_set( date2 ); /* 実験終了日時の獲得 */
    save( date1, date2 ); /* 測定データの保存 */
    break;
default: break;
}

} /* step */

puttext( 1,1,80,25,txt ); /* Recover Screen of Previous State */
return;
}

void save( char *dt1, char *dt2 ) /* ===== 測定データの保存 */
{
    FILE *fp;
    int c,num;
    char dirtmp[80],tmp[DATA_WD];

    if ( (num=files("測定データ保存ファイルの選択"))== -1 )
    {
        twindow( WD_CLOSE );
        return;
    }
    setmen( tmp, DATA_WD, 0 );
    if ( strlen( fn[num].fname )==0 ){
        if ( get_fname("ファイル名の設定",tmp)==-1 )
        {
            twindow( WD_CLOSE );
            return;
        }
    }
    else strcpy( tmp, fn[num].fname );

    sprintf( dirtmp, "%s\\%VRL%Y%$s.vr1", DIR, tmp );
    if ( ( fp=fopen( dirtmp, "wb" ) )!=NULL )
    {
        fwrite( V, sizeof(double)*7000, 1, fp );
        fclose( fp );
    }
    setmen( &fn[num], FN_SIZE, 0 ); /* パラメータのクリア */
    strcpy( fn[num].fname, tmp ); /* ファイル名の格納 */
}

```


GAS.C

```
fn[num].temp= (int)get_temp( 1 )/10.;
fn[num].humid=(int)get_humid();
/* 測定開始・終了時刻の記録 */
strcpy( fn[num].st , dt1 );      /* 開始時刻の記録 */
strcpy( fn[num].ed , dt2 );      /* 終了時刻の記録 */

twindow( WD_CLOSE );

return;
}
```

```

/* File Write Protect */
#define UP_DATE 1 /* 0-> not write 1-> write */

/* Data Index */
#define START -1
#define END -999

/* Target Sensor */

#define SS 0 /* start 0-3 */
#define SE 3+1 /* end 0-3 +1 */

#define G_X 10 /* X axis for distribution graph */
#define G_Y 300 /* Y axis for distribution graph */

#define n 12 /* number of category */
#define N 5 /* number of each samples */
#define DN 5 /* number of discriminating samples */

#define DEPTH 1
#define S_LEARN 25

struct SMELL_DATA {
    char name[10];
};

struct FRAME_INFO {
    int x1,y1,x2,y2;
    char X_unit[20],Y_unit[20];
};

struct DATA {
    double v,h,r,l;
};

struct STATISTICS {
    double ave,std,i0,F;
};

```

QUO.H

```

extern char DS[ 4 ][ n ][ 1000 ]; /* Smell distribution */
extern double psum[ S_LEARN ];

extern int col[ 12 ];
extern char *Ln[ 12 ];

extern double V[7][1000],F[1000];
extern double W[ 4 /* sensor type */ ][ DEPTH ][ n ]; /* holds Evaluation results */

extern char Rmap[ 50 /*Ns*/ ][ n ]; /* which represents whether it is Right answer or not */
extern double RmapS[ 50 /*Ns*/ ][ n ]; /* contents of the right answer */
extern double RmapS2[ 50 /*Ns*/ ][ n ]; /* contents of the right answer */

extern struct SMELL_DATA SD[N][n],SD2[ DN ][ n ],SDTMP[ DN+N ][ n ];
extern struct FRAME_INFO FI[4],FI2[4];

extern struct DATA DI[ DEPTH ][ N ][ n ];
extern struct STATISTICS SI[ DEPTH ][ n ];

```

MAIN.C

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <mem.h>
#include <math.h>
#include <graphics.h>

#include "def.h"

struct SMELL_DATA SD[N][n]={
    /* 訓練サンプルファイル名 */
    "FU4-7", "FU28-1", "GIN17-1", "GIN24-1", "AWA1", "VSOP1", "ETNL1", "MENTHA3", "POCALI3", "MASC1", "C
OFFEE6", "CHA1",
    "FU4-8", "FU28-2", "GIN17-2", "GIN24-2", "AWA2", "VSOP2", "ETNL2", "MENTHA4", "POCALI4", "MASC2", "C
OFFEE7", "CHA3",
    "FU4-9", "FU28-3", "GIN17-3", "GIN24-3", "AWA3", "VSOP3", "ETNL3", "MENTHA5", "POCALI5", "MASC3", "C
OFFEE8", "CHA4",
    "FU4-10", "FU28-4", "GIN17-4", "GIN24-4", "AWA4", "VSOP4", "ETNL4", "MENTHA6", "POCALI6", "MASC5", "C
OFFEE9", "CHA6",
    "FU4-11", "FU28-5", "GIN17-5", "GIN24-5", "AWA5", "VSOP5", "ETNL5", "MENTHA7", "POCALI7", "MASC6", "C
OFFEE10", "CHA7"
},SD2[ DN ][ n ]={
    /* テストサンプルファイル名 */
    "FU4-12", "FU28-8", "GIN17-6", "GIN24-6", "AWA6", "VSOP6", "ETNL6", "MENTHA8", "POCALI8", "MASC7", "C
OFFEE11", "CHA8",
    "FU4-14", "FU28-9", "GIN17-7", "GIN24-7", "AWA7", "VSOP7", "ETNL7", "MENTHA9", "POCALI9", "MASC8", "C
OFFEE12", "CHA9",
    "FU4-15", "FU28-10", "GIN17-8", "GIN24-8", "AWA8", "VSOP8", "ETNL8", "MENTHA10", "POCALI10", "MASC9", "C
OFFEE13", "CHA10",
    "FU4-16", "FU28-11", "GIN17-9", "GIN24-9", "AWA9", "VSOP9", "ETNL9", "MENTHA11", "POCALI11", "MASC10", "C
OFFEE14", "CHA11",
    "FU4-17", "FU28-12", "GIN17-10", "GIN24-10", "AWA10", "VSOP10", "ETNL10", "MENTHA12", "POCALI12", "MASC11", "C
OFFEE15", "CHA12"
},SDTMP[ DN+N ][ n ];
/* (説明)ここでは、訓練サンプルとテストサンプルを分けて登録しているが、
   実際には、file_sort() 関数を使って、分散の大きい順番で訓練サンプル
   を選定している。
*/

int col[ 12 ]={ WHITE,RED,YELLOW,MAGENTA,BLUE,GREEN,WHITE,CYAN,BROWN,YELLOW,GREEN,MAGENTA };
/* グラフ表示の際に線の色を変えるためのデータを格納してある配列 */

char *Ln[ 12 ]={ "FU4", "FU28", "GIN17", "GIN24", "AWA", "VSOP", "ETNL", "MENT", "POCA", "MASC", "
CAFE", "U-CHA" };
/* 12種類のサンプルの省略名を格納しておくための配列 */

char *Category[ 12 ]={ "清酒", "清酒", "吟醸酒", "吟醸酒", "焼酎", "ブランデー", "原典", "原典", "アイソトニック飲料",
"-ジュース", "コーヒー", "お茶" };
/* 12種類のサンプルの属性を格納しておくための配列 */

int catocol[12]={ WHITE,WHITE,WHITE,WHITE,WHITE,WHITE,WHITE,WHITE,BLUE,BLUE,MAGENTA
,MAGENTA,MAGENTA,MAGENTA };
/* サンプルに適した匂いの選定を行った際の、センサの組み合わせに関して、相関比
   を計算してその結果をグラフィック表示の際の線の色を格納しておくための配列 */

struct FRAME_INFO FI[5]=
{
    ( 40, 1,638/*199*/ ,359,"時間(秒)", "端子電圧(V)" ),
    ( 40/*200*/ , 1,638/*359*/ ,359,"時間(秒)", "端子電圧(V)" ),
    ( 40/*360*/ , 1,638/*519*/ ,359,"時間(秒)", "端子電圧(V)" ),
    ( 40/*520*/ , 1,638,359,"時間(秒)", "端子電圧(V)" ),
    ( 40/*520*/ , 1,638,359,"時間(秒)", "F比(分散比)" )
},
FI2[4]={ ( 1, 1,638, 95,"", "" ),
        ( 1,101,638,195,"", "" ),
        ( 1,201,638,295,"", "" ),
        ( 1,301,638,395,"", "" )
};
/* 測定結果などを表示する際に必要な各種パラメータを格納するための構造体 */

extern char DSI 4/* sensor number */ ][ n ][ 1000 ]; /* Smell distribution */
/* 匂い識別領域を求めた結果を格納するための配列 */

struct DATA DI[ DEPTH ][ N ][ n ];
/* 測定電圧、各勾配の傾き、重みなどを格納するための構造体 */

struct STATISTICS SI[ DEPTH ][ n ];
/* 測定点時間、F比などを格納するための構造体 */

extern double V[7][1000],F[1000];

```

```

/* 匂い測定パターン、F比分布を格納するための配列 */
double Wl[ 4 /* sensor type */ ][ DEPTH ][ n ]; /* holds Evaluation results */
/* 相互作用計算用関数 Estimate で使用される配列 */

char Rmap[ 50 /*Ns*/ ][ n ]; /* which represents whether it is Right answer or not */
double RmapS[ 50 /*Ns*/ ][ n ],RmapS2[ 50 /*Ns*/ ][ n ]; /* contents of the right answer */
char Mmap[ 50 /*Ns*/ ][ n ]; /* which represents whether it is Right answer or not */
double MmapS[ 50 /*Ns*/ ][ n ],MmapS2[ 50 /*Ns*/ ][ n ]; /* contents of the right answer */
/* 匂い識別結果を格納するための配列 */

double psum[ 8_LEARN ]; /* which represents learning effectiveness curve */
/* 学習曲線を記憶しておくための配列 */

int read_F(),save_F(),read_S(),save_S();
double Correlation();

void init_graph( void ) /* initialize graphics screen */
{
    int gd=DETECT,gm;
    setnewdriver( "PC98",detect98 );
    initgraph( &gd,&gm,"a:\etc" );
    return;
}

void main( int argc,char *argv[] )
{
    int i0,i,i2,i3,i4,mode,tx,ty,dx,dy,wd,cs,ce,px,py,px2,py2,pc,flg;
    int S_S=atoi(argv[2]), /* 0-3 */
        L_S=atoi(argv[3]),L_E=atoi(argv[4]), /* 0-(n-1) */
        F_F=atoi(argv[5]),F_S,F_E; /* 1..F ratio */

    static int AS[ n ][ 4 /* sensor type */ ]; /* Assign Sensor */
    static double Correlation_tbl[ n ][ 4 /* sensor number */ ],Correlation_total[ n ];
    static char Used_Flag[ 4 ][ n ]; /* Useful sensor check flag */

    long ltm;

    double sum,tmp,tmp2,v,xmax,xmin,ymax,ymin,
        Reliability[ n ],
        Average_Margin[ 4 ][ n/* category number */ ],
        Omoni[ 2 ][ n ];

    static struct S_RANK {
        int sn; /* smell number */
        double v; /* sum of margin */
    } S_rank[ n ],SRtmp;

    static struct ESTIMATE {
        double threshold;
        char *estimation;
    } ESTI[ 7 ]={
        { 0.05, "困難" },
        { 0.10, "難しい" },
        { 0.20, "やや難しい" },
        { 0.30, "やや可能" },
        { 0.40, "可能" },
        { 0.50, "やや確実" },
        { 1.0, "確実" }
    };

    char fname[40],stmp[ 80 ];
    FILE *fp;

    if ( atoi(argv[1])==1 && argc<6 )
    {
        printf("main 1 S_S(0-3) L_S,L_E(0-(n-1)) F_F(1->F比)");
        exit(0);
    }
    if ( F_F==1 ) F_S=F_E=5;
    else {
        F_S=0;
        F_E=4;
    }

    setmem( DS,sizeof( char )*4*n*1000,0 );
    setmem( Rmap,sizeof( char )*N*n,0 );
    setmem( RmapS,sizeof( double )*N*n,0 );
    setmem( RmapS2,sizeof( double )*N*n,0 );

    init_graph();

```

```

cleardevice();

mode = atoi( argv[ 1 ] );
if ( mode < 1 || mode > 21 || argc == 1 )
{
    printf("Error : argument should follow function number below.\n\n");
    printf("    1. 電圧応答パターン表示およびF比計算\n\n");
    printf("    2. 匂い学習 (相互作用可変法) \n\n");
    printf("    3. 匂い識別 (テストサンプル) \n\n");
    printf("    4. 匂い識別 (訓練 サンプル) \n\n");
    printf("    5. 匂い識別領域の作成\n\n");
    printf("   10. 匂い学習 (Neural Network) \n\n");
    printf("   11. 匂い識別 (テストサンプル) \n\n");
    printf("   12. 匂い識別 (訓練 サンプル) \n\n");
    printf("   15. N e u r o 相互作用可変法による匂い識別\n\n");
    printf("   16. N e u r o 相互作用可変法での匂い識別結果\n\n");
    printf("   20. 学習曲線をグラフ表示\n\n");
    printf("   21. 各種分析\n\n");
    exit(0);
}

if ( read_SD() == -1 )
{
    file_sort();
    save_SD();
}

clrscr();
randomize();
switch( mode )
{
    /* Z */
    case 1: /* 1. 電圧応答パターン表示およびF比計算 */
        for( i2= S_S ; i2< S_S+1 ; ++i2 ) /* sensor type */
        {
            for( i0= L_S; i0< L_E+1 ; ++i0 ) /* Liquor type */
            {
                for( i= F_S ; i< F_E+1 ; ++i ) /* file sequence */
                {
                    /* draw graph on the screen */
                    setcolor( col[ i0 ] );
                    mapping( (F_S==5 && F_E==5 ? n-1 : i0 ), i2, i, &xmax, &ymax ); /* lqtype sensory
pe file */

                    frame( FI( ( F_S==5 && F_E==5 ? 4 : i2 ) ), xmax, ymax );
                    setcolor( WHITE );
                    if ( F_S != 5 ) /* print out sample/sensor name */
                    {
                        sprintf( tmp, "%5s", ln[ i0 ] );
                        outtextxy( FI( i2 ).x2 - 44, 4, tmp );
                    } else {
                        sprintf( tmp, "%7c", (char)('A'+S_S) );
                        outtextxy( FI( 4 ).x2 - 44, 4, tmp );
                    }
                }
            }
        }
        break;
    case 2: /* 2. 匂い学習 (相互作用可変法) */
        /* 実行するときは必ずSS=0 SE=4と再定義してください */

        /* 各測定データを初期化する */
        setmem( D, sizeof( struct DATA ) * DEPTH * N * n, 0 );
        /* 学習曲線データ格納領域を初期化する */
        setmem( psum, sizeof( double ) * S_LEARN, 0 );

        /* 学習回数を決定する */
        flg=0;
        if ( read_Psum( 0 ) == -1 )
        {
            ltm=1200; /* 単位学習回数 1200 */
            flg=1;
            gotoxy( 1, 1 ); printf( "単位学習回数 = %ld", ltm );
        } else {
            /* 既にある学習曲線のピークにおける学習回数を設定する */
            tmp=0;
            for( i=0 ; i<S_LEARN ; ++i )
            {
                if ( psum[ i ] > tmp )
                {
                    tmp=psum[ i ];
                }
            }
        }
    }
}

```

MAIN.C

```

        ltm= (long)(i+1)*1200;
    }
    gotoxy( 1,1 );printf( "目標学習回数 = %ld",ltm );
}

for( pc=0 ; pc<( flg==1 ? S_LEARN : 1 ) ; ++pc )
{
    gotoxy( 1,24 );cprintf( "pc=%3d",pc+1 );

    for( i2=0 ; i2<4 ;++i2 ) /* i2 : Sensor type */
    {
        if ( read_F( i2 )!=-1 ) break;
        for( i3=0 ; i3 < DEPTH ;++i3 ) /* i3 : depth */
        {
            /* F比のピークを検出 */
            tmp=i0=0;
            for( i=0; i<1000 ;++i )
            {
                if ( F[ i ] > tmp )
                {
                    tmp=F[ i ];
                    i0=i;
                }
            }
            /* 相互作用可変法を適用する
            ltm : 学習回数
            i3 : 深さ
            i2 : センサ番号
            i0 : F比のピーク位置
            F[i0]/100. : F比 */

            VI_Method( ltm,i3,i2,i0,F[i0]/100. );
            F[ i0 ]=0;

            if ( kbhit() ) { getch();return; }
            if ( UP_DATE==0 ) getch();
        }
        /* save generated data */
        save_D( i2 );
        save_S( i2 );
    }

    if ( flg ) /* 学習曲線を作成するために訓練サンプルを識別 */
    {
        setmem( Rmap,sizeof( char )*N*n,0 );
        setmem( RmapS,sizeof( double )*N*n,0 );
        setmem( RmapS2,sizeof( double )*N*n,0 );

        for( tx=0 ; tx<n ;++tx ) /* Target Liquor type */
        {
            for( ty=0; ty<N ;++ty ) /* Target Data sample number */
            {
                sprintf( fname,"a:V%rP%Vvr1l%V%$s.vr1",SD[ ty ][ tx ].name );
                if ( (fp=fopen( fname,"rb" ))==NULL )
                {
                    fprintf( stderr,"MAIN.C -> case 2 の if( flg ) 内でエラー発生。" );
                    exit( 1 );
                }
                fread( V ,sizeof( double )*7000,1,fp );
                fclose( fp );
                V_rectify();

                Estimate(); /* Set W( Presumes the existence of V data ) */
                Results( tx,ty,SD[ ty ][ tx ].name ); /* Display results */
                if ( kbhit() ) { getch();return; }
            } /* ty */
        } /* tx */

        /* 正解サンプルの確度の和を求める */
        sum=0;
        for( tx=0 ; tx<n ;++tx ) /* Target Liquor type */
        {
            for( ty=0; ty<N ;++ty ) /* Target Data sample number */
            {
                if( Rmap[ ty ][ tx ]==1 ) sum+= RmapS[ ty ][ tx ];
            }
        }
        psum[ pc ]=sum;
    } /* if ( flg ) */
}

```

```

    ) /* for */
    if ( flg ) save_Psum( 0 ); /* 作成した学習曲線を保存 */
    break;
case 3: /* 3. 匂い識別 (テストサンプル) Rmap( 0 ) */
    if ( read_Rmap( 0 ) == -1 )
    {
        for( tx=0 ; tx<n ; ++tx ) /* Target Liquor type */
        {
            for( ty=0; ty<DN ; ++ty ) /* Target Data sample number */
            {
                /* Read target data -> V */

                sprintf( fname,"a:%Yrp%Yvrl%YXs.vrl",SD2[ ty ][ tx ].name );
                if ( (fp=fopen( fname,"rb" ))==NULL ) exit(0); /* continue */
                fread( V ,sizeof( double )*7000,1,fp );
                fclose( fp );
                V_rectify();

                Estimate(); /* Set W( Presumes the existence of V data) */
                Results( tx,ty,SD2[ ty ][ tx ].name ); /* Display results */
                if ( kbhit() ) ( getch();return; )
            } /* ty */
            /*
            if ( getch()==27 ) break;
            */
        } /* tx */
        save_Rmap( 0 );
    }

    cleardevice();
    for( i=0; i<DN ; ++i ) /* i : sample number */
    {
        result_graph( i,RmapS[ i ],RmapS2[ i ] );
    }

    break;
case 4: /* 4. 匂い識別 (訓練 サンプル) Rmap( 1 ) */
    if ( read_Rmap( 1 ) == -1 )
    {
        for( tx=0 ; tx<n ; ++tx ) /* Target Liquor type */
        {
            for( ty=0; ty<N ; ++ty ) /* Target Data sample number */
            {
                /* Read target data -> V */

                sprintf( fname,"a:%Yrp%Yvrl%YXs.vrl",SD[ ty ][ tx ].name );
                if ( (fp=fopen( fname,"rb" ))==NULL ) exit(0); /* continue */
                fread( V ,sizeof( double )*7000,1,fp );
                fclose( fp );
                V_rectify();

                Estimate(); /* Set W( Presumes the existence of V data) */
                Results( tx,ty,SD[ ty ][ tx ].name ); /* Display results */
                if ( kbhit() ) ( getch();return; )
            } /* ty */
            /*
            if ( getch()==27 ) break;
            */
        } /* tx */
        save_Rmap( 1 );
    }

    cleardevice();
    for( i=0; i<DN ; ++i ) /* i : sample number */
    {
        result_graph( i,RmapS[ i ],RmapS2[ i ] );
    }

    break;
case 5: /* 5. 匂い識別領域の作成 */
    setmem( DS,sizeof( char )*4*n*1000,0 );
    if( read_DS()== -1 ) flg=1; else flg=0;

    for( i=0 ; i<4 ; ++i ) /* i : sensor type */
    {
        /* Obtain snell map */
        if ( flg )

```



```

{
    /* 匂い識別領域を作成する */
    for( v=0 ; v < 10.0 ; v+= 0.01 ) /* voltage */
    {
        Estimate2( v,i );
        if ( max_no( Wl( i )( 0 ),-1 ) >= 0 )
        {
            DS( i )( max_no( Wl( i )( 0 ),-1 ) )( (int)(v*100) )=1;
        }
    }
}
setcolor( BLUE );
setlinestyle( DOTTED_LINE , 0 , NORM_WIDTH );
for( i2=0; i2<=10 ; ++i2 )
{
    if ( i2==0 ; i2==10 ) setlinestyle( SOLID_LINE , 0 , NORM_WIDTH );
    else setlinestyle( DOTTED_LINE , 0 , NORM_WIDTH );
    line( (int)(88.+i2*503./10.),4,
          (int)(88.+i2*503./10.),380 );
}
for( i2=0; i2<12 ; ++i2 ) /* i2 : category number */
{
    dy= 32*i2+4;
    /* サンプル名 */
    sprintf( tmp,"[ X-5s ]",Ln[i2] );
    /* 目盛りなど */
    setcolor( YELLOW );
    outtextxy( 0,dy+4,tmp );
    outtextxy( 78,dy+8,"0" );
    outtextxy( 592,dy+8,"10(V)" );
    setcolor( BLUE );
    line( 88,dy,591,dy );
    line( 88,dy+22,591,dy+22 );
    line( (int)(88.+5*503./10.),dy+22,
          (int)(88.+5*503./10.),dy+27 );
    /* 領域の書き込み */
    setcolor( CYAN );
    for( i3=0; i3<1000 ; ++i3 ) /* i3 : array no */
    {
        if ( DS( i )( i2 )( i3 )!=1 ) continue;
        dx= (int)( 88.+i3*503./1000. );
        line( dx,dy,dx,dy+22 );
    }
    setcolor( WHITE );
}
putchar( 7 );
getch();
cleardevice();
}
if ( flg ) save_DS(); /* save smell distribution */
break;
case 10: /* 10. 匂い学習 (Neural Network) */
    setnem( psum,sizeof( double )*S_LEARN,0 );

    /* 学習回数を決定 */
    if ( read_Psum( 1 )== -1 )
    {
        ltm=1200*12;
        flg=1;
        gotoxy( 1,1 );printf( "単位学習回数 = %ld",ltm );
    } else {
        tmp=0;
        flg=0;
        for( i=0 ; i<S_LEARN ; ++i )
        {
            if ( psum[ i ] > tmp )
            {
                tmp=psum[ i ];
                ltm= (long)(i+1)*1200*12;
            }
        }
        gotoxy( 1,1 );printf( "目標学習回数 = %ld",ltm );
    }

    set_X(); /* ニューラルネット学習用データを読み込む */

    for( pc=0 ; pc<( flg==1 ? S_LEARN : 1 ) ; ++pc )
    {
        gotoxy( 1,24 );cprintf( "pc=%3d",pc+1 );
    }
}

```

```

NN_learn( ltn ); /* X をニューラルネットで学習する */

/* 学習曲線用のデータを作成 */
setmem( Rmap,sizeof( char )*N*n,0 );
setmem( RmapS,sizeof( double )*N*n,0 );
setmem( RmapS2,sizeof( double )*N*n,0 );

NN_judge();

if ( flg ) /* flg : 1-単位学習 0-目標学習 */
{
    /* calculate sum of correct answers */
    sum=0;
    for( tx=0 ; tx<n ; ++tx ) /* Target Liquor type */
    {
        for( ty=0; ty<DN ; ++ty ) /* Target Data sample number */
        {
            if( Rmap[ ty ][ tx ]==1 ) sum+= RmapS[ ty ][ tx ];
        }
    }
    psum[ pc ]=sum;
} /* if ( flg ) */
} /* for */
if ( flg ) save_Psum( 1 );
break;
case 11: /* 1.1. 匂い鑑別 (テストサンプル) Rmap( 2 ) */
    if ( read_Rmap( 2 )== -1 )
    {
        setmem( Rmap,sizeof( char )*N*n,0 );
        setmem( RmapS,sizeof( double )*N*n,0 );
        setmem( RmapS2,sizeof( double )*N*n,0 );
        set_X2(); /* テストサンプルの匂い応答パターンを読み込む */
        NN_judge();
        save_Rmap( 2 );
    }
    cleardevice();
    for( i=0; i<DN ; ++i ) /* i : sample number */
    {
        result_graph( i,RmapS[ i ],RmapS2[ i ] );
    }
    break;
case 12: /* 1.2. 匂い鑑別 ( 訓練 サンプル) Rmap( 3 ) */
    if ( read_Rmap( 3 )== -1 )
    {
        setmem( Rmap,sizeof( char )*N*n,0 );
        setmem( RmapS,sizeof( double )*N*n,0 );
        setmem( RmapS2,sizeof( double )*N*n,0 );
        set_X(); /* 訓練サンプルの匂い応答パターンを読み込む */
        NN_judge();
        save_Rmap( 3 );
    }
    cleardevice();
    for( i=0; i<N ; ++i ) /* i : sample number */
    {
        result_graph( i,RmapS[ i ],RmapS2[ i ] );
    }
    break;
case 15: /* 1.5. Neuro 相互作用可変法による匂い鑑別 Rmap( 4 ) */
    if ( read_Rmap( 4 )!= -1 ) break;

    /* 各鑑別結果の平均鑑別余裕度を求める */
    setmem( Average_Margin,sizeof( double )*4*n,0 );
    setmem( Onomi,sizeof( double )*2*n,0 );
    /* 注) Onomi[0] : 相互作用可変法の結果にかかる重み
       Onomi[1] : ニューラルネットワークの結果にかかる重み */
    for( i=0; i<4 ; ++i ) /* Result file number */
    {
        if ( read_Rmap( i )== -1 )
        {
            fprintf( stderr,"鑑別結果ファイル Rmap( %c ) がありません。", 'A'+i );
            exit( 1 );
        }
        for( i2=0; i2<n ; ++i2 ) /* カテゴリ番号 */
        {
            for( i3=0; i3<N ; ++i3 ) /* 訓練サンプル番号 */
            {
                if ( Rmap[ i3 ][ i2 ] == 1 )
                {
                    Average_Margin[ i ][ i2 ]
                    += (RmapS[ i3 ][ i2 ]-RmapS2[ i3 ][ i2 ]);
                }
            }
        }
    }

```

MAIN.C

```

    )
    Average_Margin[ i ][ i2 ] /= N; /* 平均 */
}
}

/* 合成重みを作成する */
for ( i=0; i<2 ; ++i ) /* 識別器の切り替え */
{
    for( i2=0; i2<n ;++i2 )
    {
        sum= Average_Margin[ 1 ][ i2 ] + Average_Margin[ 3 ][ i2 ];
        if ( sum==0 ) Omoni[ i ][ i2 ]=0;
        else Omoni[ i ][ i2 ]= Average_Margin[ i==0 ? 1 : 3 ][ i2 ] /sum;

        /* 注) Average_Margin[ 0 ] : VIM test sample
           Average_Margin[ 1 ] : VIM training sample
           Average_Margin[ 2 ] : NM test sample
           Average_Margin[ 3 ] : NM training sample */
    }
}

/* 識別結果を Rmap( 4 ) に作成する */
setmem( Mmap,sizeof( char )*N*n,0 );
setmem( MmapS,sizeof( double )*N*n,0 );
setmem( MmapS2,sizeof( double )*N*n,0 );
/* 合成重みにより各識別結果より最終識別結果を得る */
for( i=0; i<2 ;++i ) /* 識別器切り替え */
{
    if ( read_Rmap( i==0 ? 0 : 2 )== -1 )
    {
        fprintf( stderr,"識別結果ファイル Rmap( %c ) がありません。", 'A'+(i==0 ? 0 : 2) );

        exit( 1 );
    }
    for( i2=0; i2<n ;++i2 ) /* カテゴリ番号 */
    {
        for( i3=0; i3<DN ;++i3 ) /* テストサンプル番号 */
        {
            if ( Rmap[ i3 ][ i2 ]==1 )
            {
                MmapS[ i3 ][ i2 ]
                += Omoni[ i ][ i2 ]
                *(RmapS[ i3 ][ i2 ]-RmapS2[ i3 ][ i2 ]);
                MmapS2[ i3 ][ i2 ]=0; /* 誤識別率0に設定 */
                Mmap[ i3 ][ i2 ]=( MmapS[ i3 ][ i2 ] > 0 ? 1 : 0 );
            }
        }
    }
}

/* Transfer Mmap to Rmap and save */
memcpy( Rmap,Mmap,sizeof( char )*N*n );
memcpy( RmapS,MmapS,sizeof( double )*N*n );
memcpy( RmapS2,MmapS2,sizeof( double )*N*n );
save_Rmap( 4 );
break;
case 16: /* 16. Neuro相互作用可変法での勾配識別結果(TEST SAMPLE) */
    if ( read_Rmap( 4 )== -1 ) break;
    cleardevice();
    for( i=0; i<DN ;++i ) /* i : sample number */
    {
        result_graph( i,RmapS[ i ],RmapS2[ i ] );
    }
    break;
case 20: /* 20. 学習曲線をグラフ表示 */
    for( i2=0; i2<2 ;++i2 )
    {
        if ( read_Psum( i2 )== -1 ) continue;

        cleardevice();
        clrscr();

        /* obtain psum's max value -> tmp */
        tmp=0;
        for( i=0 ; i<S_LEARN ;++i )
        {
            if ( psum[ i ] > tmp )
            {
                tmp=psum[ i ];
                ltm= (long)(i+1);
            }
        }
    }
}

```

MAIN.C

```

)
sprintf( tmp, "最適学習回数 %ld", ltm+1200*(i2==0 ? 1 : 12) );
outtextxy( 110, 50, tmp );
outtextxy( 100+(220-32), 360, "学習回数" );
settextstyle( DEFAULT_FONT, VERT_DIR, 1 );
outtextxy( 90, 50+(150-32), "総度総和" );
settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );

/* draw graph */

setcolor( CYAN );
rectangle( 100, 50, 540, 350 );
rectangle( 99, 49, 541, 351 );
sprintf( tmp, "%6.3f", tmp );
outtextxy( 544, 100-8, tmp );

setcolor( YELLOW );
setlinestyle( DOTTED_LINE, 0, NORM_WIDTH );
px= 100 + (ltm-1)*(540.-100.)/(S_LEARN-1);
line( px, 51, px, 349 );

setlinestyle( SOLID_LINE, 0, THICK_WIDTH );
setcolor( WHITE );
px2=py2=0;
for( pc=0; pc<S_LEARN; ++pc )
{
    /* plot out learning effectiveness on the screen */
    px= 100 + pc*(540.-100.)/(S_LEARN-1);
    py= 350 - 250*(psum( pc )/tmp);
    if ( px2==0 && py2==0 )
    {
        px2= px ;
        py2= py ;
    }
    line( px2, py2, px, py );
    px2=px;
    py2=py;
}
setlinestyle( SOLID_LINE, 0, NORM_WIDTH );
getch();
}
break;
case 21 : /* 21. 各種分析 */
    setmem( Correlation_tbl, sizeof( double ) * n * 4, 0 );
    setmem( AS, sizeof( int ) * n * 4, 0 );
    setmem( DS, sizeof( char ) * 4 * n * 1000, 0 );
    if( read_DS() == -1 ) break;

    for( i=0; i<n; ++i ) /* i : category number */
    {
        for( i2=0; i2<4; ++i2 ) /* i2 : sensor number */
        {
            pc=0;
            for( i3=0; i3<1000; ++i3 )
            {
                if ( DS[ i2 ][ i ][ i3 ] == 1 ) pc++;
            }
            AS[ i ][ i2 ] = pc ;
        }
    }

    /* Display The Best Sensor For Each Smells */

    clrscr();
    cleardevice();
    gotoxy( 1, 14 ); /* print index */
    for( i=0; i<n; ++i ) /* i : category number */
    {
        /* get maximum value */
        pc= -1;
        tmp=0;
        for( i2=0; i2<4; i2++ )
        {
            if ( AS[ i ][ i2 ] > tmp )
            {
                tmp= AS[ i ][ i2 ];
                pc= i2;
            }
        }
    }

```

MAIN.C

```

/* draw frame */
setcolor( CYAN );
rectangle( i*53+3,0 , i*53+3 +49 ,200 );

/* bar graph */
for( i2=0; i2<4 ;i2++ )
{
    px= i*53+3 + i2*50./4 ;
    py= 200- (int)((198.*AS[ i ][ i2 ]/(double)tmp) );
    setcolor( col[ i2 ] );
    rectangle( px+1,py , px+ (int)(50./4)-1,200 );
    if ( AS[ i ][ i2 ] > 0 )
    {
        setfillstyle( LTSLASH_FILL , col[ i2 ] );
        floodfill( px+3,py+1,col[ i2 ] );
    }
}
setcolor( WHITE );
outtextxy( i*53+3+(int)(( 50-strlen(Ln[i])*9)/2. ) , 210 ,Ln[ i ] );

/* sort useful sensor in Top down order */
tmp2= tmp;
for( i3=0; i3<4 ;++i3 )
{
    pc= -1;
    tmp=0;
    for( i2=0; i2<4 ;i2++ )
    {
        if ( AS[ i ][ i2 ] >= tmp )
        {
            tmp= AS[ i ][ i2 ];
            pc= i2;
        }
    }
    if ( pc != -1 )
    {
        sprintf( stmp,"%c",'A'+pc );
        outtextxy( i*53+3+(int)((50-strlen(stmp)*8)/2. ) , 228+18*i3 ,stmp );
        Correlation_tbl[ i ][ i3 ]=(double)pc+1. /AS[i][pc]/tmp2*/;
        AS[ i ][ pc ]= 0 ;
    }
}
} /* for */

getch();
clrscr();
cleardevice();

/*== Estimate Sensor Performance ==*/
if ( read_Rmap( 4 ) != -1 ) /* Neuro+VIM for test samples */
{
    setmem( Used_Flag,sizeof( char )*4*n,0 );
    for( i=0; i<n ;++i ) /* i : category number */
    {
        sum=0;
        for( i2=0; i2<DN ;++i2 ) /* test sample number */
        {
            if ( Rmap[ i2 ][ i ]==1 )
            {
                sum+= RmapS[ i2 ][ i ]-RmapS2[ i2 ][ i ];
            }
        }
        sum /= DN ; /* get average */
        Reliability[ i ]= sum ;

        /* Estimate */
        i0=0;
        for( i2=0; i2<7 ;++i2 )
        {
            if ( EST[i2].threshold >= sum )
            {
                i0=i2;
                break;
            }
        }

        printf(" Sample [ %5s ] Suitable Sensor=%c ( Reliability %6.4f 評価[ %10s ] )%n"
            ,Ln[ i ],(int)(Correlation_tbl[ i ][ 0 ]-1+'A'),sum,EST[ i0 ].estimation );
    }
}

```

```

/* set Used_Flag */
if ( Correlation_tbl[ i ][ 0 ] > 0 )
{
    Used_Flag[ Correlation_tbl[ i ][ 0 ]-1 ][ i ]=1;
}
} /* for i */

printf("\n" );
getch();

/* display which sensor is suitable for what */
for( i=0; i<4 ;++i ) /* i : Sensor number */
{
    printf("\n>>> The role of Sensor %c :%n", 'A'+i );
    tmp=0;
    sum=0;
    for( i2=0; i2<n ;++i2 ) /* i2 : category number */
    {
        if ( Used_Flag[ i ][ i2 ]==1 )
        {
            printf(" %5s( %s )",Ln[ i2 ],Category[ i2 ]);
            sum+= Reliability[ i2 ];
            tmp++;
        }
    }
    /* Estimate */
    i0=0;
    for( i2=0; i2<7 ;++i2 )
    {
        if ( EST[i2].threshold >= sum/tmp )
        {
            i0=i2;
            break;
        }
    }

    if ( tmp==0 ) printf( " %n\n In this experiment, Sensor %c is USELESS.%n", 'A'+i );
    else {
        printf("\n\n Reliability for all allotted samples : %6.4f 評価( %10s )%n"
            ,sum/tmp,EST[ i0 ].estimation );
    }
}

} /* if i */
break;

default:break;
}
putchar( ' ' );
getch();
clrscr();
putchar( ' ' );
getch();
closegraph();
return;
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <mem.h>
#include <math.h>
#include <graphics.h>

#include "def.h"
#include "quo.h"

void result_graph( int sample_no, double sum_v[], double sum_v2[] )
/* 正規化せずに結果を表示 */
{
    static int num=0;
    double sum=0;
    int i, i2, dx, xmax=55+12*48;
    char tmp[40];

    settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
    if ( num==n )
    {
        num=0;
        cleardevice();
    }
    if ( num==0 )
    {
        rectangle( 0, 0, xmax, 20 ); /* 上の2本線 */
        line( 0, 18, xmax, 18 );
        for( i=55; i<=xmax; i+=48 ) line( i, 0, i, 20 );
        line( 57, 0, 57, 20 );
        for( i=0; i<n; i++ )
        {
            sprintf( tmp, "%s", Ln[i] );
            outtextxy( 55+i*48+(int)( abs( 48-strlen(Ln[i])*8 )/2. ), i, tmp );
        }
        dx=num*60+20;
        setcolor( WHITE );
        rectangle( 0, dx, xmax, dx+60 ); /* 枠 */
        for( i=55; i<=xmax; i+=48 ) line( i, dx, i, dx+60 ); /* 仕切線 */
        line( 57, dx, 57, dx+60 ); /* 2重仕切線 */
        line( 57, dx+40, xmax, dx+40 ); /* 数字欄 */

        sprintf( tmp, "%2d", num+1 );
        outtextxy( 15, dx+20, tmp );

        for( i=0; i<n; i++ ) sum+=(double)sum_v[i];
        sum=1;

        for( i=0; i<n; i++ )
        {
            if ( Rmap[ sample_no ][ i ] )
            {
                setcolor( WHITE );
                circle( 55+(i+1)*48-24, dx+20, (int)( 19*((double)sum_v[i]/sum) ) );
                setfillstyle( SOLID_FILL, YELLOW );
                floodfill( 55+(i+1)*48-24, dx+20, WHITE );
                if ( sum_v2[ i ] )
                {
                    setcolor( CYAN );
                    circle( 55+(i+1)*48-24, dx+20, (int)( 19*((double)sum_v2[i]/sum) ) );
                    setfillstyle( LTSLASH_FILL, GREEN );
                    floodfill( 55+(i+1)*48-24, dx+20, CYAN );
                }
                setcolor( WHITE );
            }
            else {
                line( 55+i*48, dx, 55+(i+1)*48-1, dx+39 );
                line( 55+(i+1)*48-1, dx, 55+i*48, dx+39 );
            }
            sprintf( tmp, "%4.3f", (double)sum_v[i]/sum );
            outtextxy( 55+i*48+5, dx+42, tmp );
        }
        num++;
    }
    return;
}

void frame( struct FRAME_INFO fi, double xmax, double ymax )
/* グラフ描画用のグラフを作成 */

```

```

(
    int i2,i3,x,y,x2,y2,tmp;
    char stmp[ 80 ];
    float i;

    setcolor( WHITE );
    rectangle( fi.x1 , fi.y1 , fi.x2 , fi.y2 );
    rectangle( fi.x1-1 , fi.y1-1 , fi.x2+1 , fi.y2+1 );

    if ( strlen( fi.X_unit )==0 && strlen( fi.Y_unit )==0 ) return;
    /* X direction */
    x = ( fi.x2 - fi.x1 - strlen(fi.X_unit)*8 )/2 + fi.x1;
    y = fi.y2+20;
    outtextxy( x,y,fi.X_unit );
    i2=0;
    for( i = fi.x1 ; i<= fi.x2 ; i+=( fi.x2-fi.x1 )/10.)
    {
        line( (int)i,fi.y2,(int)i,fi.y2+( i2==5 ? 6 : 3 ) );
        if ( i2==5 ) /* middle value */
        {
            sprintf( stmp,"X3d",(int)(xmax/2.+0.5) );
            tmp=0;
            for( i3=0; i3<strlen( stmp ) ;++i3 ) if ( stmp[ i3 ]!=' ' ) tmp++;
            outtextxy( i-tmp*4,fi.y2+5,stmp );
        }
        i2++;
    }

    /* Y direction */
    x = fi.x1-24;
    y = ( fi.y2 - fi.y1 - strlen(fi.Y_unit)*8 )/2 + fi.y1;
    settextstyle( DEFAULT_FONT,VERT_DIR,1 );
    outtextxy( x,y,fi.Y_unit );
    i2=0;
    for( i = fi.y2 ; i>= fi.y1 ; i-=( fi.y2-fi.y1 )/10.)
    {
        line( fi.x1,(int)i , fi.x1-( i2==5 ? 9 : 4 ),(int)i );
        if ( i2==5 ) /* middle value */
        {
            sprintf( stmp,"X7.2f",ymax/2. );
            tmp=0;
            for( i3=0; i3<strlen( stmp ) ;++i3 ) if ( stmp[ i3 ]!=' ' ) tmp++;
            outtextxy( fi.x1-8, i-tmp*4,stmp );
        }
        i2++;
    }
    settextstyle( DEFAULT_FONT,HORIZ_DIR,1 );
    outtextxy( fi.x1-15,fi.y2+4,"0" );

    return;
}

void mapping( int lq,int st,int d,double *xmax,double *ymax )
/* 任意のセンサのグラフを描く */
{
    int i0,i,i2,i3,x,y,x2,y2,dn,ds,c;
    char fname[ 20 ],stmp[40];

    double Xibar,Xbar,tmp,numerator,denominator
        ,Xl[ n ] /* Xibar の記憶用 */
        ,Vl[ N ][ n ]; /* 使用電圧データ */

    FILE *fp;

    *ymax=10.;

    if ( d<N ) /* 各匂いサンプルの応答パターンをグラフ表示 */
    {
        gotoxy( 4,2 );cprintf( "%14s",SD[d][ lq ].name );
        sprintf( fname,"a:¥¥rp¥¥vrll¥¥¥s.vr",SD[d][ lq ].name );
        if ( (fp=fopen( fname,"rb" ))!=NULL )
        {
            fread( V ,sizeof( double )*7000,1,fp );
            fclose( fp );
            V_rectify();

            /* check data number */
            for( i=0;i<1000;++i )
            {
                if ( V[0][i]== START ) ds=i;
            }
        }
    }
}

```


FUNC.C

```

        if ( V[0][i]== END )
        {
            dn=i;
            break;
        }
    }
    for( i2=0; i2<4 ;++i2 )
    {
        for( i=ds+1; i<dn ;++i ) V[i2][i-(ds+1)]=V[i2][i];
    }
    dn -= (ds+1);
}
}
if ( d==N && lq==n-1 ) /* Calculate F ratio as sensor type st */
{
    /* Set calculation range --> ds <--> dn */

    sprintf( fname,"a:VYrpVYvr11VYXs.vr1",SDI[0][ lq ].name );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( V ,sizeof( double )*7000,1,fp );
        fclose( fp );
        V_rectify();

        for( i=0;i<1000;++i )
        {
            if ( V[st][i]== START ) ds=i;
            if ( V[st][i]== END )
            {
                dn=i;
                break;
            }
        }
    }

    if ( read_F( st ) == -1 ) /* Need to generate F file */
    {
        for( i0=ds+1 ; i0<dn ;++i0 )
        {
            /* calculate Xbar */

            Xbar=0;

            for( i2=0; i2<n ;++i2 ) /* i2 : category */
            {
                /* calculate Xibar->tmp */
                tmp=0;
                for( i3=0; i3<N ;++i3 ) /* i3 : data sequence */
                {
                    sprintf( fname,"a:VYrpVYvr11VYXs.vr1",SDI[ i3 ][ i2 ].name );
                    if ( (fp=fopen( fname,"rb" ))!=NULL )
                    {
                        fread( V ,sizeof( double )*7000,1,fp );
                        fclose( fp );
                        if ( V[ st ][ i0 ]<0 ) continue;
                        V_rectify();

                        tmp+= V[ st ][ i0 ];
                        VT[ i3 ][ i2 ]=V[ st ][ i0 ];
                    }
                }
                tmp /= (double)N;
                Xi[ i2 ]=tmp;
                Xbar += tmp;
            }
            Xbar /= n;

            numerator=0;
            denominator=0;
            for( i=0; i<n ;++i ) /* i : category */
            {
                Xibar= Xi[ i ];
                numerator += ( Xibar-Xbar )*( Xibar-Xbar );

                for( i2=0; i2<N ;++i2 ) /* i2 : data sequence */
                {
                    denominator += ( VT[ i2 ][ i ]-Xi[ i ] )*(
                        VT[ i2 ][ i ]-Xi[ i ] );
                }
            }
        }
    }
}

```

```

        denominator /= n;

        F[ i0 ]= numerator / denominator;

        gotoxy( 2,4 );
        cprintf("F = %6.3f( %3d )",numerator/denominator,dn-(i0+1) );
        if ( kbhit() )
        {
            getch();
            break;
        }
        save_F( st );          /* save F data */
    } /* F file */

    tmp=0;
    for( i=ds+1; i<dn ;++i )
    {
        if ( F[i]>tmp ) tmp=F[i];
    }
    ymax= tmp;
    for( i=ds+1; i<dn ;++i ) V[st][i-(ds+1)]=10.*F[i]/tmp;
    dn -= (ds+1);

} /* if */

/* グラフの描画 */
x2=y2=c=0;
xmax=dn;
for( i2=0 ; i2<dn ;++i2 )
{
    x= (int)(F[st].x1+i2*(double)(F[st].x2-F[st].x1)/(double)dn);
    if ( V[st][i2]<0 :: i2==0 :: ( d==N && V[st][i2]==0 ) )
    {
        setlinestyle( DOTTED_LINE,0,NORM_WIDTH );
        line( x,F[st].y1,x,F[st].y2 );
        setlinestyle( SOLID_LINE,0,NORM_WIDTH );
        c++;
        sprintf( stmp,"%d",c );
        outtextxy( x+4,F[st].y1+4,stmp );
        continue;
    }
    y= (int)(F[st].y2-V[st][i2]*(F[st].y2-F[st].y1)/10.);
    if ( x2==0 && y2==0 )
    {
        x2=x;
        y2=y;
    }
    line( x2,y2,x,y );
    x2=x;
    y2=y;
}

return;
}

int max_no( double w[],int category )
/* 最大値を求めて、カテゴリ番号を返す */
{
    int i,i2,i3,num;
    double max=0;

    /* Find out the maximum value */
    num= -1;
    for( i=0; i<n ;++i ) /* i : Liquor type */
    {
        if ( w[i] > max )
        {
            max=w[i];
            num= i;
        }
    }
    if ( category == -1 :: num == -1 ) return ( num );
    return ( w[ num ]==w[ category ] ? category : num );
}

void VI_Method( long ltm,int dep,int st,int i0,double F )
/* 学習及び学習結果表示 */

```

```

/* ltm : (単位または目標) 学習回数 */
(
static double W0[ DEPTH ][ M ][ n ], wsum[ n ];

unsigned int i, i2, i3;
int x, y, x2, y2, itmp, rx, ry, cx, cy;
char fname[ 20 ], stmp[80];
double ac, tmp, tmp2, tmp3, sum, ave[ n ], std[ n ], inc, zflg1, zflg2;
long double sumx;

FILE *fp;
if ( read_D( st ) == -1 ) /* 匂い応答パターンが D になければ作成する */
{
    /* Set struct DATA */
    for( i=0; i<n ;++i ) /* i : category */
    {
        for( i2=0; i2<N ;++i2 ) /* i2 : data sequence */
        {
            sprintf( fname, "a:VVrpVVvrl1VVXs.vrl", SD[ i ][ i ].name );
            if ( (fp=fopen( fname, "rb" )) != NULL )
            {
                fread( V, sizeof( double ) * 7000, 1, fp );
                fclose( fp );
                if ( V[ st ][ i0 ] < 0 ) continue;
                V_rectify();
                D[ dep ][ i2 ][ i ].v = V[ st ][ i0 ]; /* 測定電圧 */
                D[ dep ][ i2 ][ i ].h = 1.0; /* 振幅 */
                D[ dep ][ i2 ][ i ].r = 20.0; /* ボケ右肩 */
                D[ dep ][ i2 ][ i ].l = 20.0; /* ボケ左肩 */
            }
        }
    }
}

if ( UP_DATE == 0 ) /* 匂い応答パターンの各要素 */
{
    for( i=0; i<n ;++i ) /* i : category */
    {
        for( i2=0; i2<N ;++i2 ) /* i2 : data sequence */
        {
            printf( "==== Sample %6s.%d [深さ %d] =====\n", Ln[i], i2, dep );
            printf( "v=%f\n", D[ dep ][ i2 ][ i ].v ); /* 測定電圧 */
            printf( "h=%f\n", D[ dep ][ i2 ][ i ].h ); /* 振幅 */
            printf( "r=%f\n", D[ dep ][ i2 ][ i ].r ); /* ボケ右肩 */
            printf( "l=%f\n", D[ dep ][ i2 ][ i ].l ); /* ボケ左肩 */
        }
        putchar( '\n' );
    }
}

if( read_S( st ) == -1 ) /* 訓練サンプルの統計量を求める */
{
    /* calculate standard deviation */
    for( i2=0 ; i2<n ;++i2 )
    {
        sum=0;
        for( i=0; i<N ;++i )
        {
            sum += D[ dep ][ i ][ i2 ].v;
        }
        ave[ i2 ] = sum / (double)N;
        sum=0;
        for( i=0; i<N ;++i )
        {
            sum += ( D[ dep ][ i ][ i2 ].v - ave[ i2 ] ) * ( D[ dep ][ i ][ i2 ].v - ave[ i2 ] );
        }
        std[ i2 ] = sqrt( sum / (double)N );
    }
    /* Initialize struct STATISTICS */
    for( i=0; i<n ;++i ) /* i : category */
    {
        S[ dep ][ i ].ave = ave[ i ]; /* 平均値 */
        S[ dep ][ i ].std = std[ i ]; /* 標準偏差 */
        S[ dep ][ i ].i0 = i0; /* 測定点 */
        S[ dep ][ i ].F = F; /* 分散比 */
    }
}

/* フレーム作成と訓練サンプルの分布のグラフ表示を行う */

```

FUNC.C

```

cleardevice();
setcolor( WHITE );
rectangle( 10,10,630,380 );
rectangle( 9,9,631,381 );
sprintf( stmp,"No.X2d .. Time=%3d(sec) .. F=%8.1f .... Pattern Distribution of Sensor %c",dep+1,i0,F*100
,'A'+st );
outtextxy( 8*3,226,stmp );
for( i=0; i<11;++i ) /* draw scale */
{
    itmp= 1*(630.-G_X)/10.+G_X;
    line( itmp,380,itmp,383+( i==5 )*2 );
}
outtextxy( 622,380,"10" );
outtextxy( 6,380,"0" );
for( i=0; i<n ;++i ) line( 10,29+i*12 , 630,29+i*12 );
for( i=0; i<n ;++i )
{
    setcolor( col[i] );
    sprintf( stmp,"Sample [ %5s ] ( Standard Deviation = %8.5f )",Ln[ i ],G[ dep ][ i ].std );
    outtextxy( 18,10+i*18,stmp );
    setcolor( WHITE );
}
gotoxy( 17,18 );cprintf("Now, eliminating conflicts within Smell data");

/* 学習開始 (相互作用可変法) */
if( UP_DATE==1 )
{
    for( i=0 ; i<(int)ltn ;++i ) /* i : iteration( number of learning ) */
    {
        if ( kbhit() ){ getch();break; }

        setmem( W0,sizeof( double )*DEPTH*N*n,0 );

        /* set target data */
        rx= random( n ); /* rx : liquor type */
        ry= random( N ); /* ry : data sample number */

        /* effects estimation */
        for( i2=0 ; i2<n ;++i2 ) /* i2 : category */
        {
            for( i3=0 ; i3<N ;++i3 ) /* i3 : data sample number */
            {
                tmp= D[ dep ][i3][i2].v-D[ dep ][ry][rx].v;
                tmp1= tmp*tmp*D[ dep ][i3][i2].l;
                tmp2= tmp*tmp*D[ dep ][i3][i2].r;
                zflgl=zflgr=1;
                if ( tmp1 > 500 ) tmp1=zflgl=0;
                if ( tmp2 > 500 ) tmp2=zflgr=0;

                if ( D[ dep ][ry][rx].v < D[ dep ][i3][i2].v )
                {
                    W0[ dep ][i3][i2]=zflgl*(1.-S[ dep ][ i2 ].std)* /*<< deleted >>*/
                    D[ dep ][i3][i2].h*exp( -tmp1 );
                }
                else {
                    W0[ dep ][i3][i2]=zflgr*(1.-S[ dep ][ i2 ].std)* /*<< deleted >>*/
                    D[ dep ][i3][i2].h*exp( -tmp2 );
                }
            }
        }

        /* 各カテゴリーごとのsample(rx,ry)に対する相互作用の和を求める */
        setmem( wsum,sizeof( double )*n,0 );
        for( i2=0 ; i2<n ;++i2 ) /* i2 : category */
        {
            for( i3=0 ; i3<N ;++i3 ) /* i3 : data sample number */
            {
                wsum[ i2 ] += W0[ dep ][i3][i2];
            }
        }

        /* 最も大きい影響を及ぼしているサンプルを探す */
        tmp=0;
        cx=cy= -1; /* position flags cleared */
        for( i2=0; i2<n ;++i2 ) /* <<<<< i2 : category */
        {
            if ( rx==i2 ) continue; /* exclude category rx */
            for( i3=0 ; i3<N ;++i3 ) /* i3 : data sample number */
            {
                if ( W0[ dep ][i3][i2] > tmp )

```

FUNC.C

```

        {
            cx=i2;
            cy=i3;
            tmp= W0[ dep ][i2][i3];
        }
    }
    if ( cx== -1 || cy== -1 ) continue;
    /* 影響曲線の勾配、重みを調整する */

    sumx=0;
    for( i2=0 ; i2<n ;++i2 ) if( wsum[ i2 ] > sumx ) sumx=wsum[ i2 ];

    inc= wsum[ cx ]/sumx;

    if ( D[ dep ][ry][rx].v < D[ dep ][cy][cx].v )
    {
        if ( D[ dep ][cy][cx].l < 500. )
        {
            D[ dep ][cy][cx].l += 0.4*inc;
        }
        else {
            D[ dep ][ cy ][ cx ].h += 0.96;
            /*if ( D[ dep ][
cy][cx].h >= 0.1 ) D[ dep ][cy][cx].h -= 0.1;*/
        }
    } else if ( D[ dep ][ry][rx].v > D[ dep ][cy][cx].v )
    {
        if ( D[ dep ][cy][cx].r < 500. )
        {
            D[ dep ][cy][cx].r += 0.4*inc;
        }
        else {
            D[ dep ][ cy ][ cx ].h += 0.96;
        }
    }
    } /* for */
} /* if ( UPDATE==1 ) */

/* 学習結果をグラフで表示 */
gotoxy( 15,18 );printf("
for( i2=0; i2<n ;++i2 ) /* i2 : liquor type */
{
    /* display voltage variance */
    setcolor( WHITE );
    for( i=0; i<N ;++i )
    {
        x= G_X+(int)(620.*D[ dep ][i][i2].v/10.);
        y= 10+i2*18+15;
        bar( x,y,x,y+3 );
    }
    /* display average voltage */
    x= G_X+(int)(620.*S[ dep ][ i2 ].ave/10.);
    y= 10+i2*18+18;
    bar( x,y,x,y+2 );

    setcolor( col[ i2 ] );
    x2=y2=0;
    for( i=0; i<1000 ;i++ ) /* i : display point */
    {
        ac=0; /* variable for accumulation */
        for( i3=0; i3<N ;++i3 ) /* i3 : data sequence */
        {
            if ( 1/100. < D[ dep ][i3][i2].v )
            {
                ac += (1.-S[ dep ][ i2 ].std)*D[ dep ][i3][i2].h
                    *exp( -(D[ dep ][i3][i2].v-(double)i/100.)
                        *(D[ dep ][i3][i2].v-(double)i/100.)
                        *D[ dep ][i3][i2].l );
            } else ac += (1.-S[ dep ][ i2 ].std)*D[ dep ][i3][i2].h
                *exp( -(D[ dep ][i3][i2].v-(double)i/100.)
                    *(D[ dep ][i3][i2].v-(double)i/100.)
                    *D[ dep ][i3][i2].r );
        }
        x= G_X+(int)(620.*i/1000.);
        y= G_Y-ac*20;
        if ( x2==0 && y2==0 )
        {
            x2=x;

```

```

        y2=y;
    }
    line( x2,y2,x,y );
    x2=x;
    y2=y;
}
}
return;
}

void Estimate( void ) /* Set W1 */
/* V(smell pattern)の評価を行う */
{
    int i0,i,12,13,14;
    double tmp,tmp2,tmp3;

    setmem( W1,sizeof( double )*4*DEPTH*n,0 ); /* W1 clear */

    for( i=SS; i<SE ;++i ) /* i : Sensor number */
    {
        /* DファイルとSファイルがなければ実行せず */
        if ( read_D( i )==-1 || read_S( i )==-1 ) continue;

        for( i2=0 ; i2<DEPTH ;++i2 ) /* i2 : depth */
        {
            for( i3=0; i3<n ;++i3 ) /* i3 : Liquor Type */
            {
                tmp2=0;
                for( i4=0; i4<N ;++i4 ) /* i4 : data sequence */
                {
                    tmp= V( i )( ( S( i2 )( i3 ).f ) ); /* tmp : V of target data */
                    tmp3= D( i2 )( i4 )( i3 ).v - tmp; /* tmp3: Distance */
                    if ( tmp < D( i2 )( i4 )( i3 ).v )
                    {
                        tmp2 += ( S( i2 )( i3 ).F )*(1.-S( i2 )( i3 ).std)
                               *D( i2 )( i4 )( i3 ).h*exp( -tmp3*tmp3*D( i2 )( i4 )( i3 ).l );
                    } else {
                        tmp2 += ( S( i2 )( i3 ).F )*(1.-S( i2 )( i3 ).std)
                               *D( i2 )( i4 )( i3 ).h*exp( -tmp3*tmp3*D( i2 )( i4 )( i3 ).r );
                    }
                }
                W1[ i ][ i2 ][ i3 ]= tmp2 ;
            }
        } /* depth */
    } /* Sensor number */

    return;
}

void Estimate2( double v,int s ) /* Set W1 for making Smell map */
{
    int i0,i,12,13,14;
    double tmp,tmp2,tmp3,sum1[DEPTH][n],sum2[n],sumx;

    setmem( W1,sizeof( double )*4*DEPTH*n,0 ); /* W1 clear */

    for( i=s ; i<s+1 ;++i ) /* i : Sensor number */
    {
        /* DファイルとSファイルがなければ実行せず */
        if ( read_D( i )==-1 || read_S( i )==-1 ) break;

        setmem( sum1,sizeof( double )*DEPTH*n,0 );
        setmem( sum2,sizeof( double )*n ,0 );

        for( i3=0; i3<n ;++i3 ) /* i3 : Liquor type */
        {
            tmp2=0;
            for( i2=0 ; i2<DEPTH ;++i2 ) /* i2 : depth */
            {
                for( i4=0; i4<N ;++i4 ) /* i4 : data sequence */
                {
                    tmp= v; /* tmp : arbitrary Voltage */
                    tmp3= D( i2 )( i4 )( i3 ).v - tmp; /* tmp3: Distance */
                    if ( tmp < D( i2 )( i4 )( i3 ).v )
                    {
                        tmp2 += ( S( i2 )( i3 ).F )*(1.-S( i2 )( i3 ).std)
                               *D( i2 )( i4 )( i3 ).h*exp( -tmp3*tmp3*D( i2 )( i4 )( i3 ).l );
                    } else {
                        tmp2 += ( S( i2 )( i3 ).F )*(1.-S( i2 )( i3 ).std)

```

```

        *D[ i2 ][ i4 ][ i3 ].h*exp( -tmp3*tmp3*D[ i2 ][ i4 ][ i3 ].r );
    }
    } /* data sequence */
    sum1[ i2 ][ i3 ]= tmp2;
    } /* depth */
} /* Liquor Type */

/* 深さごとの割合を求める */
for( i3=0; i3<DEPTH ;++i3 ) /* i3 : depth */
{
    sumx=0;
    for( i2=0; i2<n ;++i2 ) sumx+= sum1[ i3 ][ i2 ];
    for( i2=0; i2<n ;++i2 ) /* i2 : category */
    {
        if( sumx==0 ) sum1[ i3 ][ i2 ]=0;
        else sum1[ i3 ][ i2 ]/= sumx;
    }
}
/* 深さごとの割合の和を求める */
for( i2=0; i2<n ;++i2 ) /* i2 : category */
{
    for( i3=0; i3<DEPTH ;++i3 ) /* i3 : depth */
    {
        W1[ i ][ 0/*i3*/ ][ i2 ]+= sum1[ i3 ][ i2 ];
    }
}
} /* Sensor number */

return;
}

void Normalizing_print( double w[] )
{
    int i,i2,i3;
    double max,sum;

    return;
    /* get sum value */
    sum=0;
    for( i=0; i<n ;++i ) sum+= w[i];

    /* Print out the normalized value */
    for( i=0; i<n ;++i ) /* i : liquor type */
    {
        printf("X5.3f ",w[i]/sum /*max*/ );
    }
    printf(" : %s",Ln[ max_no( w,-1 ) ] );
    printf("\n");
    return;
}

void Results( int tx,int ty,char *name ) /* Display results */
{
    int i,i2,i3,mxn;
    double sum[ n ],sumx;

    clrscr();

    /* print out the results */

    printf("\n Discrimination result of < %s >\n\n",name);

    for( i=0; i<DEPTH ;++i ) /* i : depth */
    {
        /* print out sum */
        setmem( sum,sizeof( double )*n,0 );

        for( i2=0; i2<n ;++i2 ) /* i2 : liquor number */
        {
            for( i3=0; i3<4 ;++i3 ) /* i3 : sensor number */
            {
                sum[ i2 ]+= W1[ i3 ][ i ][ i2 ];
            }
        }
        Normalizing_print( sum );
    }

    /* print out total sum */
    for( i3=0; i3<4 ;++i3 ) /* i3 : sensor number */

```

```

{
    setmem( sum,sizeof( double )*n,0 );
    for( i2=0; i2<n ;++i2 ) /* i2 : liquor number */
    {
        for( i=0; i<DEPTH ;++i ) /* i : depth */
        {
            sum[ i2 ]+= W1[ i3 ][ i ][ i2 ];
        }
    }
    Normalizing_print( sum );
}

setmem( sum,sizeof( double )*n,0 );
for( i=0; i<n ;++i ) /* i : liquor number */
{
    for( i2=0; i2<DEPTH ;++i2 ) /* i2 : depth */
    {
        for( i3=0; i3<4 ;++i3 ) /* i3 : sensor number */
        {
            sum[ i ]+= W1[ i3 ][ i2 ][ i ];
        }
    }
}
Normalizing_print( sum );

mxn= max_no( sum,tx );
printf("===== %7s =====\n",
        tx==mxn ? "Correct" : "Wrong!" );

Rmap [ ty ][ tx ]=( tx==mxn ? 1 : 0 );

/* get sum value */
sumx=0;
for( i=0; i<n ;++i ) sumx+= sum[i];
RmapS[ ty ][ tx ]= sum[ tx ]/sumx;

/* set RmapS second largest number */
if ( mxn != -1 ) sum[ mxn ]=0;
mxn= max_no( sum,tx );
RmapS2[ ty ][ tx ]= ( mxn==-1 ? 0 : sum[ mxn ]/sumx );

return;
}

double Correlation( double s1[ 4 ],double s2[ 4 ] ) /* 任意の数列間の相関係数を求める */
{
    int i,i2,i3;
    double av_x,av_y,numerator,denominator1,denominator2,tmp,r;

    /* obtain av_x(credibility) */
    av_x=0;
    for( i=0 ; i<4 ;++i ) /* i : sensor number */
    {
        av_x += s1[ i ];
    }
    av_x /= 4;

    /* obtain av_y(density) */
    av_y=0;
    for( i=0 ; i<4 ;++i ) /* i : sensor number */
    {
        av_y += s2[ i ];
    }
    av_y /= 4;

    /* obtain numerator */
    numerator=0;
    for( i=0; i<4 ;++i ) /* i : sensor number */
    {
        numerator += (s1[ i ]-av_x)*(s2[ i ]-av_y);
    }

    /* obtain denominator1 */
    denominator1=0;
    for( i=0; i<4 ;++i ) /* i : sensor number */
    {
        denominator1 += (s1[ i ]-av_x)*(s1[ i ]-av_x);
    }
    denominator1 = sqrt( denominator1 );

```


FUNC.C

```
/* obtain denominator2 */
denominator2=0;
for( i=0; i<4 ;++i ) /* i : sensor number */
{
    denominator2 += (s2[ i ]-av_y)*(s2[ i ]-av_y);
}
denominator2 = sqrt( denominator2 );
return ( numerator / ( denominator1*denominator2 ) );
}
```

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <graphics.h>

#include "def.h"
#include "quo.h"

#define MAX_LAYER 3
#define MAX_ELEMENT (12 + 1)

double output[ MAX_LAYER ][ MAX_ELEMENT ];
double input[ MAX_LAYER ][ MAX_ELEMENT ];
double weight[ MAX_LAYER ][ MAX_ELEMENT ][ MAX_ELEMENT ];
double dweight[ MAX_LAYER ][ MAX_ELEMENT ][ MAX_ELEMENT ];
double back_output[ MAX_LAYER ][ MAX_ELEMENT ];
int layer_structure[ MAX_LAYER ]={ 4,9,12 }; /* 各構成 < MAX_ELEMENT */

void V_rectify();

struct INPUT {
    double sf5 /* sensor type+1 */;
} X[ N+DN ][ n ]; /* NN input data */

int save_X(),read_X(),save_WEIGHT(),read_WEIGHT();
void set_X(),set_X2();

double calc_f( double x )
{
    return 1./( 1.+exp( -x ) );
}

void Neuron_weight_init( void )
{
    int i,i2,i3;

    for( i=0 ; i<MAX_LAYER ;++i )
    {
        for( i2=0; i2<MAX_ELEMENT ;++i2 ) /* from layer n */
        {
            for( i3=0; i3<MAX_ELEMENT ;++i3 ) /* to layer n+1 */
            {
                weight[ i ][ i2 ][ i3 ]= (double)( random(1000)/1000. );
            }
        }
    }
    return;
}

void Forward_propagation( int tx,int ty,int layer_no )
/* layer_no must be above 1 */
{
    int i,i2,i3;
    double sum;

    output[ layer_no-1 ][ layer_structure[ layer_no-1 ] ]= -1; /* 閾値 */

    /* 出力値の計算 */
    for( i=0; i < layer_structure[ layer_no ] ;++i )
    {
        sum=0;
        for( i2=0; i2<= layer_structure[ layer_no-1 ] ; ++i2 )
        {
            sum+= weight[ layer_no ][ i2 ][ i ]*output[ layer_no-1 ][ i2 ];
        }
        input[ layer_no ][ i ]= sum;
        output[ layer_no ][ i ]= calc_f( sum );
    }
    return ;
}

void Back_propagation( int tx, int ty, int layer_no )
{
    int i,i2,i3,i1,j;
    double sum,sum2;

    if ( layer_no== MAX_LAYER-1 )
    /* 終端出力での誤差逆伝搬 */
    {

```

```

for( j=0 ; j < layer_structure[ layer_no ] ;++j )
{
    back_output[ layer_no ][ j ]=
        ( ( j==tx ? 1. : 0 ) - output[ layer_no ][ j ] )
        *calc_f( input[layer_no][j] )*( 1.-calc_f( input[layer_no][j] ) );
    for( ii=0; ii <= layer_structure[ layer_no-1 ] ;++ii )
    {
        dweight[ layer_no ][ ii ][ j ]=
            0.25*back_output[ layer_no ][ j ]*output[ layer_no-1 ][ ii ]
            +0.9*dweight[ layer_no ][ ii ][ j ];
        weight[ layer_no ][ ii ][ j ] += dweight[ layer_no ][ ii ][ j ];
    }
}
}
else
{
    /* 中間出力での誤差逆伝搬 */
    for( j=0 ; j < layer_structure[ layer_no ] ;++j )
    {
        sum=0;
        for( i=0; i<layer_structure[ layer_no+1 ] ;++i )
        {
            sum+= weight[ layer_no+1 ][ j ][ i ]*back_output[ layer_no+1 ][ i ];
        }
        back_output[ layer_no ][ j ]= sum*
            calc_f( input[layer_no][j] )*( 1.-calc_f( input[layer_no][j] ) );

        for( ii=0; ii <= layer_structure[ layer_no-1 ] ;++ii )
        {
            dweight[ layer_no ][ ii ][ j ]=
                0.25*back_output[ layer_no ][ j ]*output[ layer_no-1 ][ ii ]
                +0.9*dweight[ layer_no ][ ii ][ j ];
            weight[ layer_no ][ ii ][ j ] += dweight[ layer_no ][ ii ][ j ];
        }
    }
}
return;
}

void NN_learn( long ltm ) /* ニューラルネットワークによる匂い学習 */
{
    int i,i2,i3,j,k,tx,ty,wflg;
    long l,c=0;

    if ( read_WEIGHT( 0 )== -1 ) /* 注) 0 は学習用の初期の重みです */
    {
        /* ここは1度だけしか実行されません */
        Neuron_weight_init();
        save_WEIGHT( 0 ); /* 学習用の初期の重みを保存します */
    }

    read_WEIGHT( 1 ); /* 最新の重みを読み込む */

    /* learning reiteration */

    for( l=0; l<ltm ;++l ) /* learning iteration */
    {
        /* random sample pick up */
        ty= random( N ); /* Target Sample number */
        tx= random( n ); /* Target Liquor type */
        for( i=0; i<layer_structure[ 0 ] ;++i )
        {
            output[ 0 ][ i ]= X[ ty ][ tx ].s[ i ];
        }

        for( i=1; i<MAX_LAYER ;++i )
        {
            Forward_propagation( tx,ty,i );
        }
        /*
        c++;
        gotoxy( 1,1 );printf( "Learning iteration : %5ld  target : %2d",c,tx+1 );
        if ( kbhit() )
        {
            if( getch()==27 ) break;
            for( i=0; i<14 /* MAX_LAYER */ ;++i )
            {
                for( i2=0; i2<MAX_LAYER ;++i2 )
                {
                    gotoxy( 3+i2*10,5+i );
                    printf( "(%2d) %5.3lf\n",i+1,output[ i2 ][ i ] );
                }
            }
        }
        */
    }
}

```

```

        )
    }
    getch();
    putchar( 7 );
}
*/
for( i=MAX_LAYER-1 ; i>0 ; --i )
{
    Back_propagation( tx,ty,i );
}

) /* 1 */

save_WEIGHT( 1 ); /* 1は最新の重みです */

return;
}

void NN_judge( void ) /* ニューラルネットワーク識別 */
{
    int i,i2,i3,j,k,tx,ty,l;
    double i12,i13,tap,sumx;

    if ( read_WEIGHT( 1 ) == -1 )
    {
        fprintf( stderr, "Can't read weight file!" );
        exit( 1 );
    }

    for( ty=0; ty<DN ; ++ty ) /* Target Data sample number */
    {
        for( tx=0 ; tx<n ; ++tx ) /* Target Liquor type */
        {
            for( i=0; i<layer_structure[ 0 ] ; ++i )
            {
                output[ 0 ][ i ] = X[ ty ][ tx ].s[ i ];
            }
            for( i=1; i<MAX_LAYER ; ++i )
            {
                Forward_propagation( tx,ty,i );
            }
            /*
            gotoxy( 1,1 ); cprintf( "target : %2d", tx+1 );
            for( i=0; i<14 ; ++i )
            {
                for( i2=0; i2<MAX_LAYER ; ++i2 )
                {
                    gotoxy( 3+i2*10,5+i );
                    cprintf( "(%2d) %5.3lf\n", i+1, output[ i2 ][ i ] );
                }
            }
            getch();
            */
            Rmap[ ty ][ tx ] = ( max_no( output[ MAX_LAYER-1 ], tx ) == tx ? 1 : 0 );
            sumx=0;
            for( i=0; i<n ; ++i ) sumx += output[ MAX_LAYER-1 ][ i ];
            if ( sumx==0 ) continue;
            RmapS[ ty ][ tx ] = output[ MAX_LAYER-1 ][ tx ] / sumx;

            /* set the second largest */
            if ( max_no( output[ MAX_LAYER-1 ], tx ) != -1 )
                output[ MAX_LAYER-1 ][ max_no( output[ MAX_LAYER-1 ], tx ) ] = 0;
            RmapS2[ ty ][ tx ] = ( max_no( output[ MAX_LAYER-1 ], tx ) == -1
                ? 0 : output[ MAX_LAYER-1 ][
                    max_no( output[ MAX_LAYER-1 ], tx ) ] / sumx );

        } /* tx */
        putchar( '\n' );
    } /* ty */

    return;
}

int save_X( char *fn ) /* save X values */
{
    FILE *fp;
    char fname[40];

```

```

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:VYrpVYdata1VYXs",fn );
    if ( (fp=fopen( fname,"wb" ))==NULL ) return -1;
    fwrite( X,sizeof( struct INPUT )*(N+DN)*n,1,fp );
    fclose( fp );
    return 0;
}

int read_X( char *fn ) /* read X values */
{
    FILE *fp;
    char fname[40];

    sprintf( fname,"a:VYrpVYdata1VYXs",fn );
    if ( (fp=fopen( fname,"rb" ))==NULL ) return -1;
    fread( X,sizeof( struct INPUT )*(N+DN)*n,1,fp );
    fclose( fp );
    return 0;
}

void set_X() /* Set X */
{
    int i,12,13,tx,ty,
        tmp,i0,
        F_max_point[ 4 /* sensor type */];
    char fname[40];
    FILE *fp;

    if ( read_X("X")!= -1 ) return;

    /* Initialize struct DATA */
    setmem( X,sizeof( struct INPUT )*(N+DN)*n,0 );
    setmem( F_max_point,sizeof( int )*4,0 );

    for( i2=0 ; i2<4 ;++i2 ) /* i2 : Sensor type */
    {
        if ( read_F( i2 )==-1 )
        {
            fprintf( stderr,"Can't read F data!" );
            exit( 1 );
        }
        /* find out the peek of F value */
        tmp=i0=0;
        for( i=0; i<1000 ;++i )
        {
            if ( F[ i ] > tmp )
            {
                tmp=F[ i ];
                i0=i;
            }
        }
        F_max_point[ i2 ]= i0 ;
    }
    for( tx=0 ; tx<n ;++tx ) /* Target Liquor type */
    {
        for( ty=0; ty<N ;++ty ) /* Target Data sample number */
        {
            /* Read target data -> V */
            sprintf( fname,"a:VYrpVYvr11VYXs.vr1",SD( ty )[ tx ].name );
            if ( (fp=fopen( fname,"rb" ))==NULL )
            {
                fprintf( stderr,"Can't read smell data!" );
                exit( 1 );
            }
            fread( V,sizeof( double )*7000,1,fp );
            fclose( fp );
            V_rectify();
            for( i=0; i<4 ;++i )
            {
                X[ ty ][ tx ].s[ i ]= V[ i ][ F_max_point[ i ] ];
            }
            X[ ty ][ tx ].s[ 4 ]= -1.;
        } /* ty */
    } /* tx */
    save_X("X"); /* save sample data for learning */
    return;
}

```

```

void set_X2() /* Set X */
{
    int i, i2, i3, tx, ty,
        tmp, i0,
        F_max_point[ 4 /* sensor type */];
    char fname[40];
    FILE *fp;

    if ( read_X("X2") != -1 ) return;

    /* Initialize struct DATA */
    setmem( X, sizeof( struct INPUT )*(N+DN)*n, 0 );
    setmem( F_max_point, sizeof( int )*4, 0 );

    for( i2=0 ; i2<4 ; ++i2 ) /* i2 : Sensor type */
    {
        if ( read_F( i2 ) == -1 )
        {
            fprintf( stderr, "Can't read F data!" );
            exit( 1 );
        }
        /* find out the peak of F value */
        tmp=i0=0;
        for( i=0; i<1000 ; ++i )
        {
            if ( F[ i ] > tmp )
            {
                tmp=F[ i ];
                i0=i;
            }
        }
        F_max_point[ i2 ] = i0 ;
    }
    for( tx=0 ; tx<n ; ++tx ) /* Target Liquor type */
    {
        for( ty=0; ty<DN ; ++ty ) /* Target Data sample number */
        {
            /* Read target data -> V */
            sprintf( fname, "a:%04p%04v%04s.vr1", SD2( ty )( tx ).name );
            if ( (fp=fopen( fname, "rb" ))==NULL )
            {
                fprintf( stderr, "Can't read smell data!" );
                exit( 1 );
            }
            fread( V, sizeof( double )*7000, 1, fp );
            fclose( fp );
            V_rectify();
            for( i=0; i<4 ; ++i )
            {
                X[ ty ][ tx ].s[ i ] = V[ i ][ F_max_point[ i ] ];
            }
            X[ ty ][ tx ].s[ 4 ] = -1.;
        } /* ty */
    } /* tx */
    save_X("X2"); /* save unknown sample data */
    return;
}

int save_WEIGHT( int flg ) /* save weight parameters */
{
    FILE *fp;
    char fname[40];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname, "a:%04p%04vdata%04weight%c", 'A'+flg );
    if ( (fp=fopen( fname, "wb" ))==NULL ) return -1;
    fwrite( output, sizeof( double )*MAX_LAYER*MAX_ELEMENT, 1, fp );
    fwrite( input, sizeof( double )*MAX_LAYER*MAX_ELEMENT, 1, fp );
    fwrite( weight, sizeof( double )*MAX_LAYER*MAX_ELEMENT*MAX_ELEMENT, 1, fp );
    fwrite( dweight, sizeof( double )*MAX_LAYER*MAX_ELEMENT*MAX_ELEMENT, 1, fp );
    fwrite( back_output, sizeof( double )*MAX_LAYER*MAX_ELEMENT, 1, fp );
    fclose( fp );

    return 0;
}

int read_WEIGHT( int flg ) /* read weight parameters */
{
    FILE *fp;

```

```

char fname[40];

sprintf( fname,"a:%Yrp%Ydata%Yweight%c",'A'+flg );
if ( (fp=fopen( fname,"rb" ) )==NULL ) return -1;
fread( output,sizeof( double )*MAX_LAYER*MAX_ELEMENT,1,fp );
fread( input,sizeof( double )*MAX_LAYER*MAX_ELEMENT,1,fp );
fread( weight,sizeof( double )*MAX_LAYER*MAX_ELEMENT*MAX_ELEMENT,1,fp );
fread( dweight,sizeof( double )*MAX_LAYER*MAX_ELEMENT*MAX_ELEMENT,1,fp );
fread( back_output,sizeof( double )*MAX_LAYER*MAX_ELEMENT,1,fp );
fclose( fp );

return 0;
}

double get_temp( double v ) /* 温度センサ1・2から入力 */
(
    return v*100.; /* (mV) */
)

double get_humid( double v,double temp ) /* 湿度センサから入力 */
(
    double tmp,g,humidity;

    tmp= temp/10.; /* temperature l */

    g= v/( 0.1251-0.001*( 0.74554*tmp-11.5214 ) ); /* absolute humidity */
    humidity= g*100./52. ;

    return humidity;
)

int hosei=0; /* 補正曲線の表示 */

void V_rectify( void ) /****** 温度、湿度補正を行う */
/* あらかじめPDは設定されているものとして処理 */
/* cn means file number */
(
    struct RECTIFY (
        double a1,a2;
        double b1,b2;
    );
    static struct RECTIFY R[3]=
    (
        ( -0.003846,1.035 , -0.000462,2.296 ),
        ( 0.002 ,0.4 , -0.004 ,1.9 ),
        ( 0.0045 ,0.15 , -0.006 ,1.85 )
    );
    /*
    static double vpu[6]={ 0.8, 0.8, 1.0, 1.0, 0.9, 0.9 }
    ,vpd[6]={ 1.2, 1.2, 1.0, 1.0, 1.1, 1.1 };
    */
    double standard_a,standard_b,standard_r,a,b,r,Rx,Rv,Rst2;
    int i,i2,i3,flg,dx,dy,dx2,dy2;
    char tmp[40];

    return;
    if ( hosei )
    (
        for( i=0; i<3 ;++i )
        (
            line( 100,0,100,400 ); /* スケールを描く */
            line( 100,200,500,200 );
            outtextxy( 70,191,"1.0" );
            outtextxy( 510,191,"Temperature(°C)" );
            sprintf( tmp,"センサ %c 温湿度特性",'A'+i );
            outtextxy( 260,3,tmp );
            line( 100,200-log( 0.5 )*200,105,200-log( 0.5 )*200 );
            outtextxy( 70,192-log( 0.5 )*200,"0.5" );
            line( 100,200-log( 1.5 )*200,105,200-log( 1.5 )*200 );
            outtextxy( 70,192-log( 1.5 )*200,"1.5" );
            line( 100,200-log( 2.0 )*200,105,200-log( 2.0 )*200 );
            outtextxy( 70,192-log( 2.0 )*200,"2.0" );
            settextstyle( DEFAULT_FONT,VERT_DIR,1 );
            outtextxy( 60,152,"log( Rs/Ro )" );
            settextstyle( DEFAULT_FONT,HORIZ_DIR,1 );

            for( i2=0; i2<=100 ;i2+=10 )
            (
                a= R[i].a1*i2+R[i].a2;
                b= R[i].b1*i2+R[i].b2;
                flg=0;
            )
        )
    )
)

```

```

for( i3=0; i3<30 ;++i3 )
{
    r= -(a/25.)*i3+b;
    dx=100+i3*400/29;
    dy=200-log(r)*200;

    if( i3%2 ==0 )
    {
        line( dx,200,dx,205 );    /* 温度メモリを入れる */
        sprintf( tmp,"%2d",i3 );
        outtextxy( dx-8,208,tmp );
    }
    if ( flg==0 )
    {
        dx2=dx;
        dy2=dy;
        flg=1;
    }
    line( dx,dy,dx2,dy2 );
    dx2=dx;
    dy2=dy;
}
}
getch();
cleardevice();
}
for( i=0; i<3 ;++i )
{
    i2=128;
    if ( V[0][i2] <= 0 ) continue;
    standard_a= R[i].a1*65.+R[i].a2;
    standard_b= R[i].b1*65.+R[i].b2;
    standard_r= -(standard_a/25.)*30.+standard_b;

    a= R[i].a1*get_humid(V[6][i2],get_temp(V[4][i2]))+R[i].a2;
    b= R[i].b1*get_humid(V[6][i2],get_temp(V[4][i2]))+R[i].b2;
    r= -(a/25.)*get_temp(V[4][i2])+b;

    /*
    printf("humidity %f : temperature %f\n",get_humid(V[6][i2],get_temp(V[4][i2])),get_temp(V[4][i2]
));
    r *= fn[cn].temp/20.;
    */

    Rx= standard_r/r;          /* 修正比率(to 20C,65%) */

    for( i2=0; i2<1000 ;++i2 )
    {
        if ( V[i][i2]>0 )
        {
            Rst2=( 10./V[i][i2]-1. )*4100.;
            Rv= ( Rst2+4100. )/( Rst2+Rx+4100. );
            V[i][i2] *= Rv;
            /*
            gotoxy( 1,i+1 );cprintf("%f",Rv);
            */
        }
    } /* i2 */
} /* i */
return;
}

```



```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <mem.h>
#include <math.h>
#include <graphics.h>

#include "def.h"
#include "quo.h"

int read_Rmap( int flg )    /* read Rmap data */
(
    FILE *fp;
    char fname[ 40 ];

    sprintf( fname,"a:%YrpYYdata1YYR_MAP%d",flg );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( Rmap,sizeof( char )*50*n,1,fp );
        fread( RmapS,sizeof( double )*50*n,1,fp );
        fread( RmapS2,sizeof( double )*50*n,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
)

int save_Rmap( int flg )    /* save Rmap data */
(
    FILE *fp;
    char fname[ 40 ];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:%YrpYYdata1YYR_MAP%d",flg );
    if ( (fp=fopen( fname,"wb" ))!=NULL )
    {
        fwrite( Rmap,sizeof( char )*50*n,1,fp );
        fwrite( RmapS,sizeof( double )*50*n,1,fp );
        fwrite( RmapS2,sizeof( double )*50*n,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
)

int read_DS( void )    /* read Smell map data */
(
    FILE *fp;
    char fname[ 40 ];

    sprintf( fname,"a:%YrpYYdata1YYS_MAP" );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( DS,sizeof( char )*4*n*1000,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
)

int save_DS( void )    /* save Smell map data */
(
    FILE *fp;
    char fname[ 40 ];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:%YrpYYdata1YYS_MAP" );
    if ( (fp=fopen( fname,"wb" ))!=NULL )
    {
        fwrite( DS,sizeof( char )*4*n*1000,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
)

```

```

int read_Psum( int flg )      /* read Psum data */
{
    FILE *fp;
    char fname[ 40 ];

    sprintf( fname,"a:%Yrp%Ydata1%YL_CURVE%c",'A'+flg );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( psum,sizeof( double )*S_LEARN,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int save_Psum( int flg )      /* save Psum data */
{
    FILE *fp;
    char fname[ 40 ];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:%Yrp%Ydata1%YL_CURVE%c",'A'+flg );
    if ( (fp=fopen( fname,"wb" ))!=NULL )
    {
        fwrite( psum,sizeof( double )*S_LEARN,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int read_F( int st )         /* read F data */
{
    FILE *fp;
    char fname[ 40 ];

    sprintf( fname,"a:%Yrp%Ydata1%Y%c",'A'+st );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( F,sizeof( double )*1000,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int save_F( int st )         /* save F data */
{
    FILE *fp;
    char fname[ 40 ];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:%Yrp%Ydata1%Y%c",'A'+st );
    if ( (fp=fopen( fname,"wb" ))!=NULL )
    {
        fwrite( F,sizeof( double )*1000,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int read_D( int st )         /* read D data */
{
    FILE *fp;
    char fname[ 40 ];

    sprintf( fname,"a:%Yrp%Ydata1%YcD",'A'+st );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( D,sizeof( struct DATA )*DEPTH*N*n,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

```

```

)

int save_D( int st )      /* save D data */
{
    FILE *fp;
    char fname[ 40 ];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:%%rp%%data\\%%xcD", 'A'+st );
    if ( (fp=fopen( fname,"wb" ))!=NULL )
    {
        fwrite( D,sizeof( struct DATA ):DEPTH*n,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int read_S( int st )      /* read S data */
{
    FILE *fp;
    char fname[ 40 ];

    sprintf( fname,"a:%%rp%%data\\%%xcS", 'A'+st );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( S,sizeof( struct STATISTICS ):DEPTH*n,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int save_S( int st )      /* save S data */
{
    FILE *fp;
    char fname[ 40 ];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:%%rp%%data\\%%xcS", 'A'+st );
    if ( (fp=fopen( fname,"wb" ))!=NULL )
    {
        fwrite( S,sizeof( struct STATISTICS ):DEPTH*n,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int read_SD( void )      /* read SD data */
{
    FILE *fp;
    char fname[ 40 ];

    sprintf( fname,"a:%%rp%%data\\%%xcSD" );
    if ( (fp=fopen( fname,"rb" ))!=NULL )
    {
        fread( SD ,sizeof( struct SMELL_DATA ):N*n,1,fp );
        fread( SD2,sizeof( struct SMELL_DATA ):DN*n,1,fp );
        fclose( fp );
        return 0;
    }
    return -1;
}

int save_SD( void )      /* save SD data */
{
    FILE *fp;
    char fname[ 40 ];

    if ( UP_DATE==0 ) return (-1);

    sprintf( fname,"a:%%rp%%data\\%%xcSD" );
    if ( (fp=fopen( fname,"wb" ))!=NULL )
    {
        fwrite( SD ,sizeof( struct SMELL_DATA ):N*n,1,fp );
        fwrite( SD2,sizeof( struct SMELL_DATA ):DN*n,1,fp );
        fclose( fp );
    }
}

```

```

        return 0;
    }
    return -1;
}

void file_sort( void )
{
    int i, j, k, l, m, n, c, flag;
    double tmp1, tmp2, average[ n ], sum;
    char fname[ 40 ];

    static struct AVERAGE {
        double av;
        char name[ 10 ];
    } Av[ DN+N ][ n ], Av_tmp;

    FILE *fp;

    textcursor( MODISP_CURSOR );
    textcolor( T_YELLOW );
    textblink( BLINK );
    cprintf( "Now, checking the sample data and sorting them. Wait a moment please.\n" );
    textblink( NOBLINK );
    textcolor( T_WHITE );

    /* copy SD , SD2 --> SDTMP */
    setmem( SDTMP, sizeof( struct SMELL_DATA )*(DN+N)*n, 0 );
    for( i=0; i<n ;++i ) /* i : Liquor Type */
    {
        for( j=0; j<N ;++j ) /* j : sample number */
        {
            strcpy( SDTMP[ i ][ j ].name, SD[ i ][ j ].name );
        }
        for( j=0; j<DN ;++j ) /* j : sample number */
        {
            strcpy( SDTMP[ i+N ][ j ].name, SD2[ i ][ j ].name );
        }
    }

    /* set each sample's average value */
    setmem( Av, sizeof( struct AVERAGE )*(DN+N)*n, 0 );
    for( i=0; i<n ;++i ) /* i : Liquor Type */
    {
        for( j=0; j<DN+N ;++j ) /* j : sample number */
        {
            /* get average value */

            sprintf( fname, "a:%03d%03d%03d%03d.vrl", SDTMP[ i ][ j ].name );
            if ( (fp=fopen( fname, "rb" ))==NULL )
            {
                fprintf( stderr, "Can't read file %s.", SDTMP[ i ][ j ].name );
                exit( 1 );
            }
            fread( V, sizeof( double )*7000, 1, fp );
            fclose( fp );
            V_rectify();

            sum=c=0;
            for( k=0; k<4 ;++k ) /* k : sensor number */
            {
                flag=0;
                for( l=0; l<1000 ;++l ) /* l : measuring point */
                {
                    if ( V[ i ][ k ][ l ] == -2 ) flag=1;
                    if ( V[ i ][ k ][ l ] >= 0 && flag )
                    {
                        sum += V[ i ][ k ][ l ];
                        c++;
                    }
                }
            }
            Av[ i ][ j ].av = sum / (double)c ;
            strcpy( Av[ i ][ j ].name , SDTMP[ i ][ j ].name );
        }
    }

    /* set average value for each category */
    for( i=0; i<n ;++i ) /* i : category number */
    {
        sum=0;

```

FILE.C

```

    for( i2=0; i2<DN+N ;++i2 ) sum += Av[ i2 ][ i ].av;
    average[ i ]= sum /(double)( DN+N );
}
for( i=0; i<n ;++i ) /* i : category number */
{
    for( i2=0; i2<DN+N ;++i2 ) /* i2 : sample number */
    {
        Av[ i2 ][ i ].av = fabs(Av[ i2 ][ i ].av - average[ i ]);
    }
}

/* sort_file for each category */
for( i=0; i<n ;++i ) /* i : category number */
{
    /* bubble sort */
    for( i2=0; i2<DN+N ;++i2 )
    {
        for( i3=0; i3<DN+N-1 ;++i3 )
        {
            if ( Av[ i3 ][ i ].av < Av[ i3+1 ][ i ].av )
            {
                /* swap */
                Av_tmp=Av[ i3 ][ i ];
                Av[ i3 ][ i ] = Av[ i3+1 ][ i ];
                Av[ i3+1 ][ i ]=Av_tmp;
            }
        }
    }
}

/* Assign results for SD and SD2 */
for( i=0; i<n ;++i ) /* i : category number */
{
    textcolor( T_WHITE );
    printf( " samples for learning ( category %d )\n",i+1 );
    textcolor( T_YELLOW );
    for( i2=0; i2<N ;++i2 ) /* i2 : sample number */
    {
        strcpy( SD[ i2 ][ i ].name,Av[ i2 ][ i ].name );
        printf( "%s ",SD[ i2 ][ i ].name );
    }
    putchar('\n');
    textcolor( T_WHITE );
    printf( " samples for Discrimination ( category %d )\n",i+1 );
    textcolor( T_CYAN );
    for( i2=0; i2<DN ;++i2 ) /* i2 : sample number */
    {
        strcpy( SD2[ i2 ][ i ].name,Av[ i2+N ][ i ].name );
        printf( "%s ",SD2[ i2 ][ i ].name );
    }
    putchar('\n');
    textcolor( T_CYAN );
    printf( "Finished!\n");
    textcolor( T_WHITE );
    return;
}

```

何-108

MEM.C

```
#include "def.h"
```

```
double V[7][1000],F[1000];
```

付-109

MEM2.C

```
#include "def.h"
```

```
char DS[ 4 /* sensor number */ ][ n ][ 1000 ]; /* smell distribution */
```