お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ（http://www.renesas.com）

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社（http://www.renesas.com）

【問い合わせ先】http://japan.renesas.com/inquiry

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

アプリケーション・ノート

RENESAS

μPD17103

資料番号　IEA-675（第1版）
発行年月　October 1996 N

アプリケーション・ノート

NEC

μPD17103

本製品が外国為替および外国貿易管理法の規定による戦略物資等（または役務）に該当するか否かは，ユーザ（仕様を決定した者）が判定してください。

本資料に掲載の応用回路および回路定数は，例示的に示したものであり，量産設計を対象とするものではありません。

○本資料の内容は，後日変更する場合があります。
○文書による当社の承諾なしに本資料の転載複製を禁じます。
○本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して，当社は当社もしくは第三者の知的所有権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合，当社はその責を負うものではありませんのでご了承ください。
○当社は品質，信頼性の向上に努めていますが，半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として，人身事故，火災事故，社会的な損害等を生じさせない冗長設計，延焼対策設計，誤動作防止設計等安全設計に十分ご注意願います。
○当社は，当社製品の品質水準を「標準水準」，「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また，各品質水準は以下に示す用途に製品が使われることを意図しておりますので，当社製品の品質水準をご確認の上ご使用願います。

標準水準：コンピュータ，OA機器，通信機器，計測機器，AV機器，家電，工作機械，パーソナル機器，産業用ロボット

特別水準：輸送機器（自動車，列車，船舶等），交通用信号機器，防災／防犯装置，各種安全装置，生命維持を直接の目的としない医療機器

特定水準：航空機器，航空宇宙機器，海底中継機器，原子力制御システム，生命維持のための医療機器，生命維持のための装置またはシステム等

当社製品のデータ・シート／データ・ブック等の資料で，特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は，必ず事前に当社販売窓口までご相談頂きますようお願い致します。

○この製品は耐放射線設計をしておりません。

巻末にアンケート・コーナを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

目　　次

第1章　概　　説

μPD17103は，ROM容量1Kバイト（512×16ビット），RAM容量16ワード（16×4ビット）とI/Oポート11本で構成されている非常にシンプルなCMOS 4ビット・シングルチップ・マイクロコンピュータです。

CPUとして，アキュームレータがなく，直接データ・メモリを操作することのできるμPD17000アーキテクチャを採用しているため大変効率のよいプログラミングが可能です。またすべての命令は16ビット長1語で構成されています。

μPD17103により，家電製品やTOYなどの電子制御化や，汎用ロジックICで構成されている回路のシングルチップ化が簡単にしかも低価格で実現できます。

μPD17103のシステム開発用として使いやすいインサーキット・エミュレータやアセンブラが用意されています。

なお，タイニ・マイクロコントローラ・シリーズとしてはμPD17103のほかに，ポートが16本のμPD17104，また，外付け抵抗発振回路が抵抗1本でOKのμPD17107とμPD17108があります。

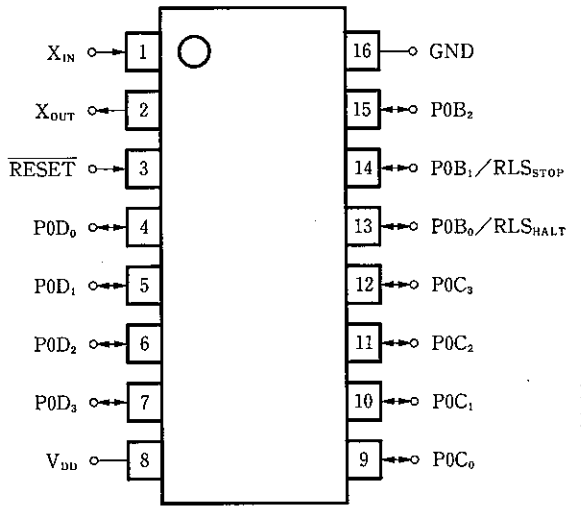
特　　徴

○プログラム・メモリ（ROM）：1Kバイト（512×16ビット）
○データ・メモリ（RAM）：16ワード（16×4ビット）
○スタック・レベル：1
○32種の分かりやすいインストラクション・セット
○10進／16進加減算可能
○命令実行時間：2μs（8MHz X' tal発振子／セラミック発振子接続）
○I/Oポート：11本（Nチャネル・オープン・ドレイン出力3本）
○スタンバイ機能保有（HALT／STOP命令）
○動作電源電圧：2.7～6.0V（2MHz 動作時）
　　4.5～6.0V（8MHz 動作時）
○CMOS低消費電力

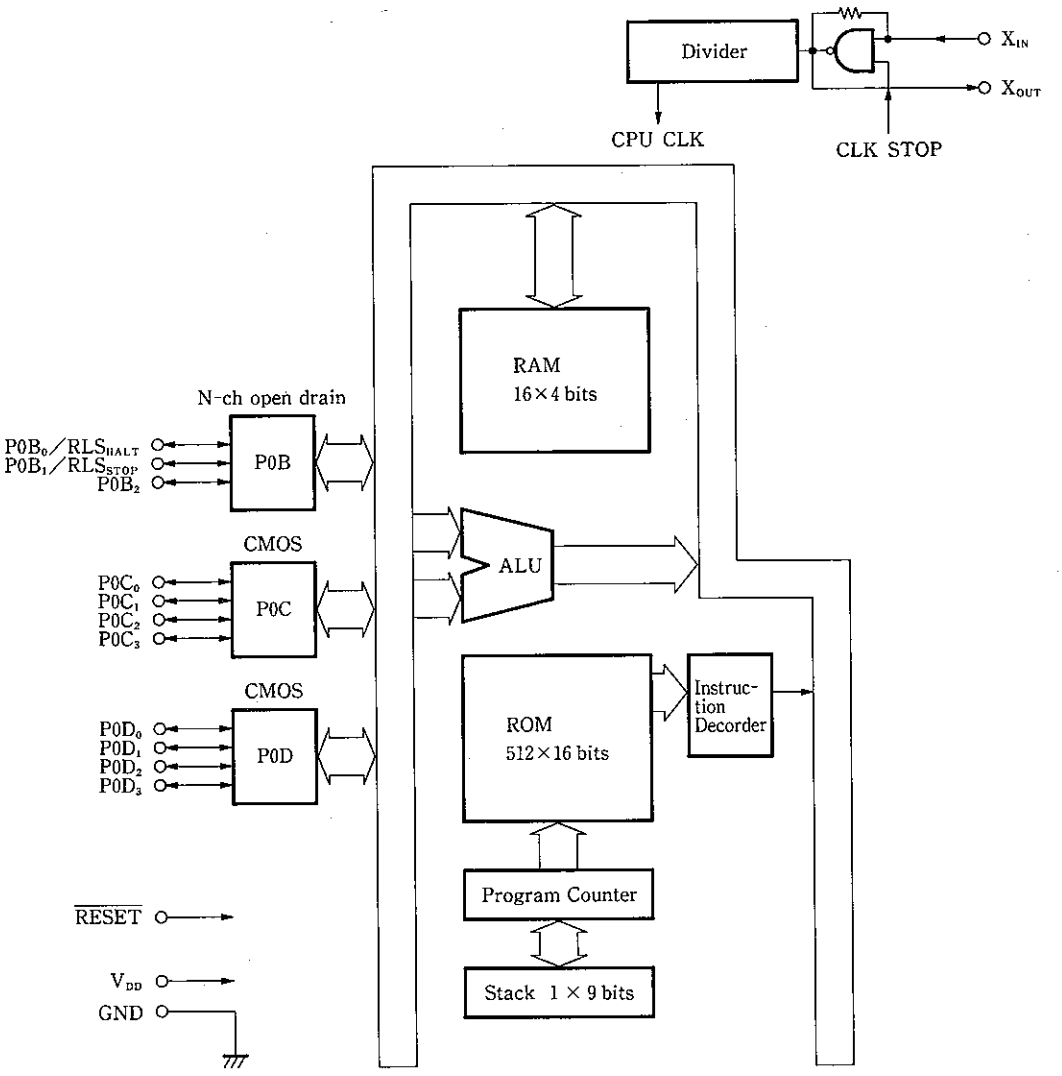
オーダ情報

オーダ名称	パッケージ
μPD17103CX-×××	16ピン・プラスチックDIP（300 mil）
μPD17103GS-×××	16ピン・プラスチックSOP（300 mil）

μPD17103端子接続図（Top View）



μPD17103ブロック図



端子機能一覧

○ポート端子

端子名称	入出力	機能	リセット時
$P0B_0$／RLS_{HALT}	入出力	HALTモード解除用	・オープン・ドレイン時 ハイ・インピーダンス （入力モード） ・プルアップ抵抗内蔵時 ハイ・レベル （入力モード）
$P0B_1$／RLS_{STOP}		STOPモード解除用	
$P0B_2$		・N-chオープン・ドレイン4ビット入出力ポート（Port 0 B） ・ビット単位でプルアップ抵抗内蔵可能(マスク・オプション) ・オープン・ドレイン時　9V耐圧	
$P0C_0$～$P0C_3$	入出力	CMOS（プッシュプル）4ビット入出力ポート（Port 0 C）	ハイ・インピーダンス （入力モード）
$P0D_0$～$P0D_3$	入出力	CMOS（プッシュプル）4ビット入出力ポート（Port 0 D）	ハイ・インピーダンス （入力モード）

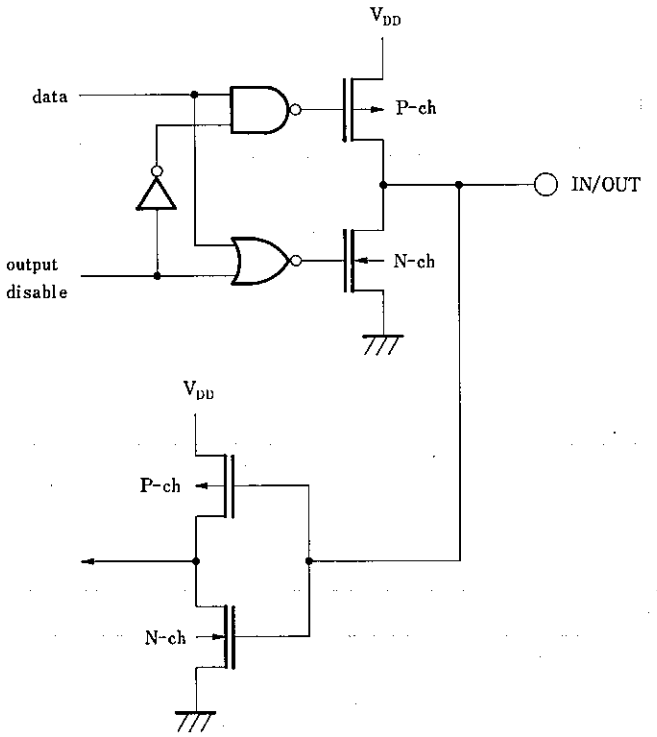
○ポート端子以外

端子名称	入出力	機能	リセット時
$\overline{RESET}$	入　力	・システム・リセット入力端子 ・プルアップ抵抗内蔵可能（マスク・オプション）	
V_{DD}		・正電源端子	
GND		・GND電位端子	
X_{IN}，X_{OUT}		・システム・クロック発振用発振子接続端子	

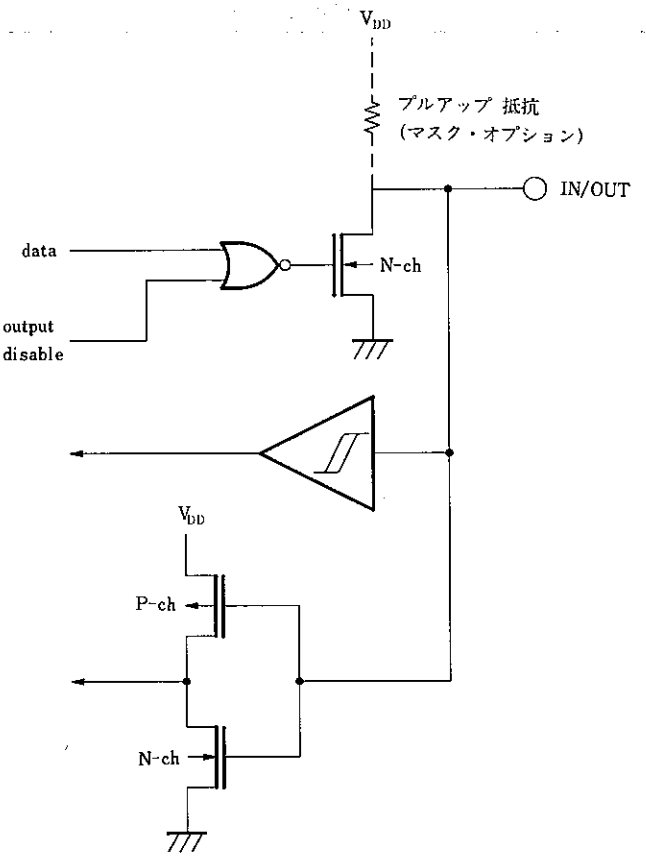
端子の入出力回路

μPD17103の各端子の入出力回路を一部簡略化した形式を用いて示します。

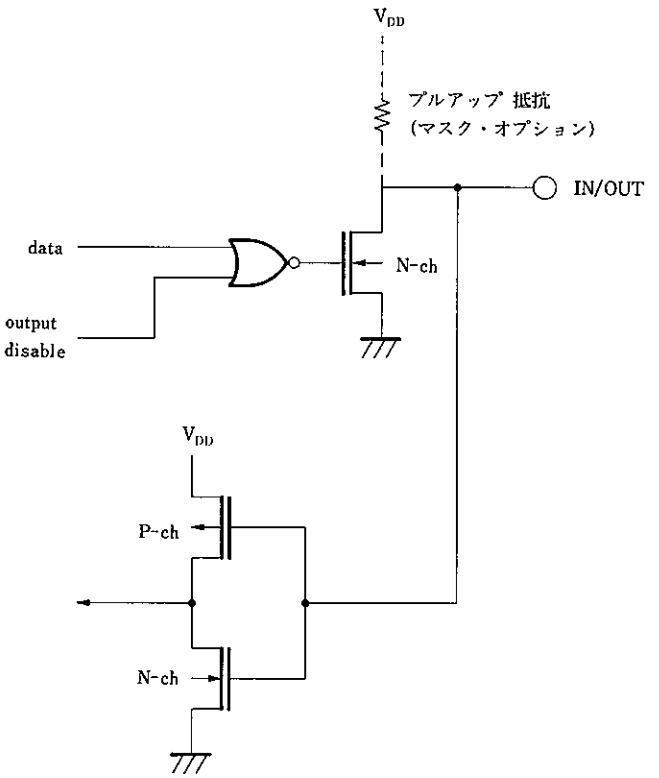
(1) P0C, P0D



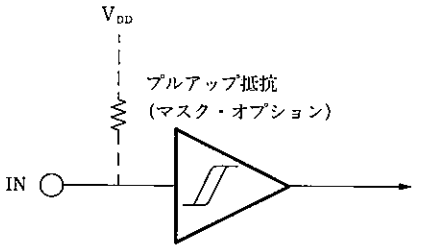
(2) $P0B_0$, $P0B_1$



(3) $P0B_2$



(4) $\overline{RESET}$



第２章　リモコン受信プログラム

２．１　プログラム説明

本アプリケーション・ノートに記述するプログラム内容について示します。

２．１．１　プログラム概説

μPD17103を使用してリモコン受信，２つの７セグメントLEDのダイナミック表示，実行ステップによるタイマ制御，タイマのカウント動作状態を示すLED表示を行います。

タイマの制御は，受信したリモコンの特定のコードにより行い，７セグメントLEDには，タイマ・カウンタに設定されている値を表示します。また，LEDは，タイマがカウント動作中のとき点灯します。

２．１．２　リモコン受信

μPD17103のP0B_0端子にプリアンプ（μPC1491）を接続してリモコンの受信を行います。受信したリモコンのコードにより処理が以下のように異なります。

有効カスタム・コードは04Hとします。04H以外のカスタム・コードを受信したときは，リモコン・送信器の不一致としてデータ・コードの受信は行いません。

04Hのカスタム・コードを受信したときは，データ・コードの受信を行います。有効データ・コードは，00H，01H，02Hとします。00H，01H，02H以外のデータ・コードを受信したときは，ノイズとしてタイマの制御を行いません。

01H，02Hのデータ・コードのリピート・コードは有効とします。01H，02H以外のデータ・コードのリピート・コードを受信してもデータ・コードの不一致として無効とします。

２．１．３　タイマ制御

タイマ制御は，受信したリモコンのコードにより行います。タイマのカウンタに設定できる値は，分刻みで１分から100分までの範囲です。タイマ・モードにはタイマ動作モードとタイマ設定モードの２種類のモードがあり，タイマ設定モードにおいてのみ，タイマ・カウンタに新たにデータを設定することができます。

(1)　タイマ・モード

タイマ・モードには，以下に示す２種類のモードがあり，データ・コード00Hを受信するごとにモードが切り替わり，タイマ・カウント開始またはカウント値の設定のどちらかを選択することができます。

(a)　**タイマ動作モード**

タイマのカウント動作を行うモードです。

この動作モードにおいては，タイマ・カウンタに新たにカウント値を設定することはできません（データ・コード01H，02Hを受信しても無効になります）。

(b)　**タイマ設定モード**

タイマ・カウンタに新たにカウント値を設定するモードです。このモードにおいて，データ・コード01H，02Hを受信するとタイマ・カウンタに新たにデータを設定できます。

タイマがカウント動作中であれば，カウント動作を継続して行います。ただし，データ・コード01H，02Hを受信してタイマ・カウンタに新たにデータを設定するとタイマのカウント動作は停止します。

また，タイマ設定モードにおいて，タイマのカウントが終了するとモードはタイマ動作モードに切り替わります。

(2)　**タイマ・カウンタのデータ設定**

タイマ・カウンタはリモコンの特定のデータ・コード（01H，02H）により制御します。タイマ・カウンタの制御を行うモードは，タイマ設定モードです。タイマ設定モードにおいてデータ・コード（01H，02H）およびそのリピート・コードを受信すると次のように処理を行います。

(a)　**データ・コード01Hを受信したときの処理**

タイマ動作モードからタイマ設定モードに切り替えて，最初に受信したデータ・コードが01Hのときは，タイマ・カウンタに1分を設定します。このときタイマのカウント動作は停止します。

タイマ設定モードにおいて一度，01Hまたは02Hのデータ・コードを受信すると以後は，01Hのデータ・コードを受信するごとにタイマ・カウンタの値をインクリメントします。たとえばタイマ・カウンタに設定されている値が25分のときに，01Hのデータ・コードを受信するとタイマ・カウンタには26分が設定されます。

表2－1　01Hのデータ・コードを受信したときのタイマ・カウンタの値

受信前のタイマ・カウンタの値(分)	受信後のタイマ・カウンタの値(分)
1から99まで	設定されている値をインクリメントします
100	1

注）　ただし，タイマ動作モードからタイマ設定モードに切り替えて最初に01Hのデータ・コードを受信したときは，タイマ・カウンタには1分を設定します。

(b)　データ・コード02Hを受信したときの処理

タイマ動作モードからタイマ設定モードに切り替えて，最初に受信したデータ・コードが02Hのときは，タイマ・カウンタに100分を設定します。このときタイマのカウント動作は停止します。

タイマ設定モードにおいて一度，01Hまたは02Hのデータ・コードを受信すると以後は，02Hのデータ・コードを受信するごとにタイマ・カウンタの値をデクリメントします。たとえばタイマ・カウンタに設定されている値が25分のときに，02Hのデータ・コードを受信するとタイマ・カウンタには24分が設定されます。

表２－２　02Hのデータ・コードを受信したときのタイマ・カウンタの値

受信前のタイマ・カウンタの値(分)	受信後のタイマ・カウンタの値(分)
２から100まで	設定されている値をデクリメントします
1	100

注）　ただし，タイマ動作モードからタイマ設定モードに切り替えて最初に02Hのデータ・コードを受信したときは，タイマ・カウンタには，100分を設定します。

(c)　リピート・コードを受信したときの処理

タイマ設定モードにおいて01Hまたは02Hのデータ・コードのリピート・コードを受信すると以下のようにタイマ・カウンタの値を設定します。

(i)　01Hのデータ・コードのリピート・コードを受信したときの処理

01Hのデータ・コード受信後，そのコードに続いてリピート・コードを８個受信するとタイマ・カウンタの値をインクリメントします。８個のリピート・コード受信後は，そのリピート・コードに続いてリピート・コードを２個受信するごとにタイマ・カウンタの値をインクリメントします。

(ii)　02Hのデータ・コードのリピート・コードを受信したときの処理

02Hのデータ・コード受信後，そのコードに続いてリピート・コードを８個受信するとタイマ・カウンタの値をデクリメントします。８個のリピート・コード受信後は，そのリピート・コードに続いてリピート・コードを２個受信するごとにタイマ・カウンタの値をデクリメントします。

(3)　タイマのカウント開始

タイマ・カウンタに新たにデータを設定したときは，タイマはカウント停止状態になっています。この状態で再びタイマのカウントを開始するには，モードをタイマ設定モードからタイマ動作モードに切り替えることにより行います。つまり，00Hのデータ・コードを受信することによりタイマは，カウントを開始します。

2.1.4　7セグメントLEDダイナミック表示

7セグメントLEDの表示は，タイマのモード，タイマ・カウンタの状態により以下のように異なります。

(1)　タイマ動作モードにおける表示

タイマ動作モードにおいては，タイマがカウント動作中のときとカウント停止中のときでは表示が以下のように異なります。

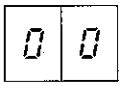
(a)　タイマがカウント動作中のときの表示

タイマがカウント動作中のとき，7セグメントLEDには，タイマ・カウンタの値を表示します。タイマ・カウンタの値により以下のように表示は異なります。

(i)　タイマ・カウンタの値が99分以上のときの表示

タイマ・カウンタの値が99分以上のときは，7セグメントLEDには，“ 00 ”を表示します。

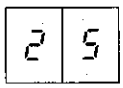
図2－1　タイマ・カウンタの値が99分以上のときの9セグメントLED表示



(ii)　タイマ・カウンタの値が9分以上99分未満のときの表示

タイマ・カウンタの値が9分以上99分未満のときは，タイマ・カウンタの分の値に1を加えた値を2桁表示します。たとえば，タイマ・カウンタの値が24分15秒のときは以下のようになります。

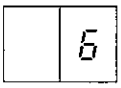
図2－2　タイマ・カウンタの値が24分15秒のときの7セグメントLED表示



(iii)　タイマ・カウンタの値が1分以上9分未満のときの表示

タイマ・カウンタの値が1分以上9分未満のときは，タイマ・カウンタの分の値に1を加えた値を1桁表示します。たとえば，タイマ・カウンタの値が5分34秒のときは以下のようになります。

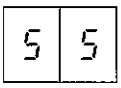
図２－３　タイマ・カウンタの値が５分34秒のときの７セグメントLED表示



(iv)　タイマ・カウンタの値が９秒以上60秒未満のときの表示

タイマ・カウンタの値が９秒以上60秒未満のときは，タイマ・カウンタの秒の値に１を加えた値を２桁表示します。たとえば，タイマ・カウンタの値が54秒のときは以下のようになります。

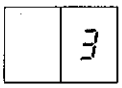
図２－４　タイマ・カウンタの値が54秒のときの７セグメントLED表示



(v)　タイマ・カウンタの値が０秒より大きく９秒未満のときの表示（ただし０秒は含みません）

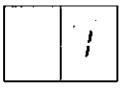
タイマ・カウンタの値が０秒より大きく９秒未満のときは，タイマ・カウンタの秒の値に１を加えた値を１桁表示します。たとえば，タイマ・カウンタの値が２秒のときは以下のようになります。

図２－５　タイマ・カウンタの値が２秒のときの７セグメントLED表示



また，タイマ・カウンタの値が500 msのときは以下のようになります。

図２－６　タイマ・カウンタの値が500 msのときの７セグメントLED表示



(b)　**タイマがカウント停止中のときの表示**

タイマがカウント停止中のとき，７セグメントLEDには“ 00 ”を250 ms間隔で点滅表示します。

(2)　タイマ設定モードにおける表示

タイマ設定モードにおいては，タイマ・カウンタに新たにデータを設定したときと設定していないときでは，表示が以下のように異なります。

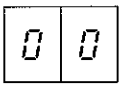
(a)　新たにタイマ・カウンタにデータを設定したときの表示

新たにタイマ・カウンタにデータを設定したとき，7セグメントLEDには，タイマ・カウンタの分の値を表示します。タイマ・カウンタの分の値により以下のように表示は異なります。

(i)　タイマ・カウンタの値が100分のときの表示

タイマ・カウンタの値が100分のときは，7セグメントLEDには，“ 00 ”を表示します。

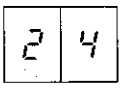
図2-7　タイマ・カウンタの値が100分のときの7セグメントLED表示



(ii)　タイマ・カウンタの値が10分以上99分以下のときの表示

タイマ・カウンタの値が10分以上99分以下のときは，タイマ・カウンタの分の値を2桁表示します。たとえば，タイマ・カウンタの値が24分のときは以下のようになります。

図2-8　タイマ・カウンタの値が24分のときの7セグメントLED表示



(iii)　タイマ・カウンタの値が1分以上9分以下のときの表示

タイマ・カウンタの値が1分以上9分以下のときは，タイマ・カウンタの分の値を1桁表示します。たとえば，タイマ・カウンタの値が5分のときは以下のようになります。

図2-9　タイマ・カウンタの値が5分のときの7セグメントLED表示

(b)　タイマ・カウンタに新たにデータを設定していないときの表示

タイマ・カウンタに新たにデータを設定していないとき，7セグメントLEDには，“－－”を表示します。ただし，タイマがカウント動作中で，カウントを終了したときは，7セグメントLEDには，“ 00 ”を250 ms間隔で点滅表示し，モードをタイマ動作モードに切り替えます。

(3)　コモン端子制御

7セグメントLEDのコモン端子はそれぞれμPD17103のP0B$_1$端子，P0B$_2$端子に接続しており，P0B$_1$端子，P0B$_2$端子の出力により制御を行います。7セグメントLEDは，それぞれP0B$_1$端子，P0B$_2$端子の出力がロウ・レベルのとき点灯します。

P0B$_1$端子，P0B$_2$端子の出力は，表示パターンにより以下のようになります。

図2－10　“ 00 ”点滅表示のときのP0B$_1$ 端子，P0B$_2$ 端子の出力波形

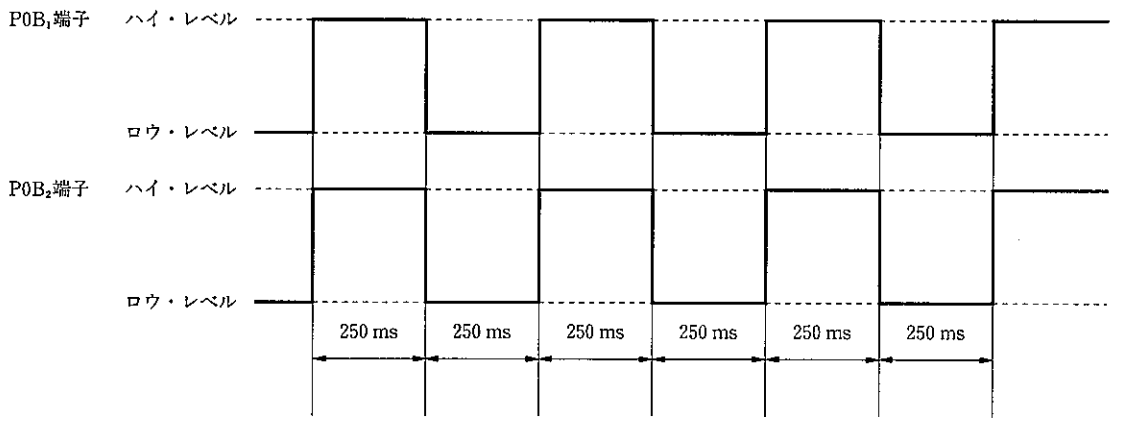
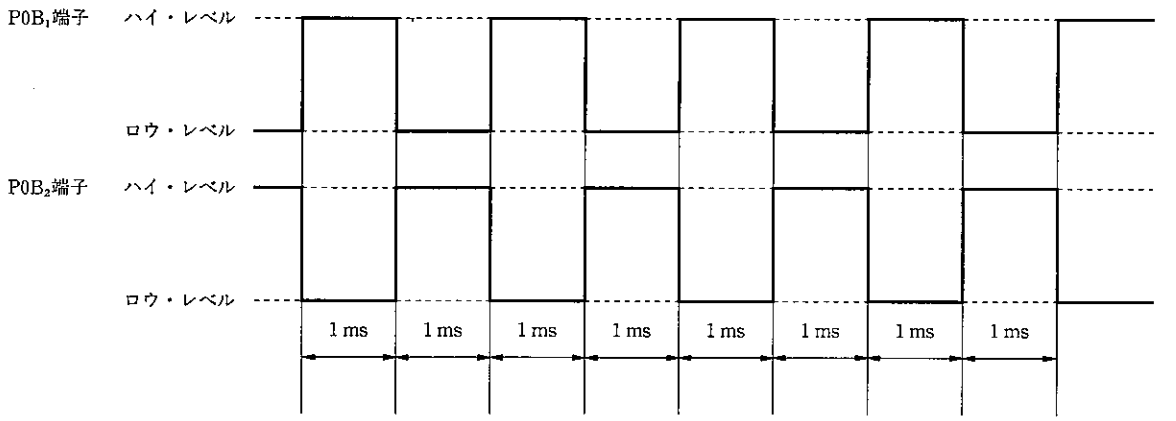


図2－11　“ 00 ”点滅表示以外のときのP0B$_1$ 端子，P0B$_2$ 端子の出力波形



(a)　7セグメントLEDに“ 00 ”の点滅表示を行うときのP0B$_1$端子，P0B$_2$端子の出力波形は，図2－10のようになります。P0B$_1$端子の出力波形とP0B$_2$端子の出力波形は一致し250 ms間隔で端子の出力は反転します。

(b)　7セグメントLEDの表示が(a)以外のときは，P0B$_1$端子，P0B$_2$端子の出力波形は，図2－11のようになります。P0B$_1$端子，P0B$_2$端子の出力波形は反対となり，1 ms間隔で端子の出力は反転します。

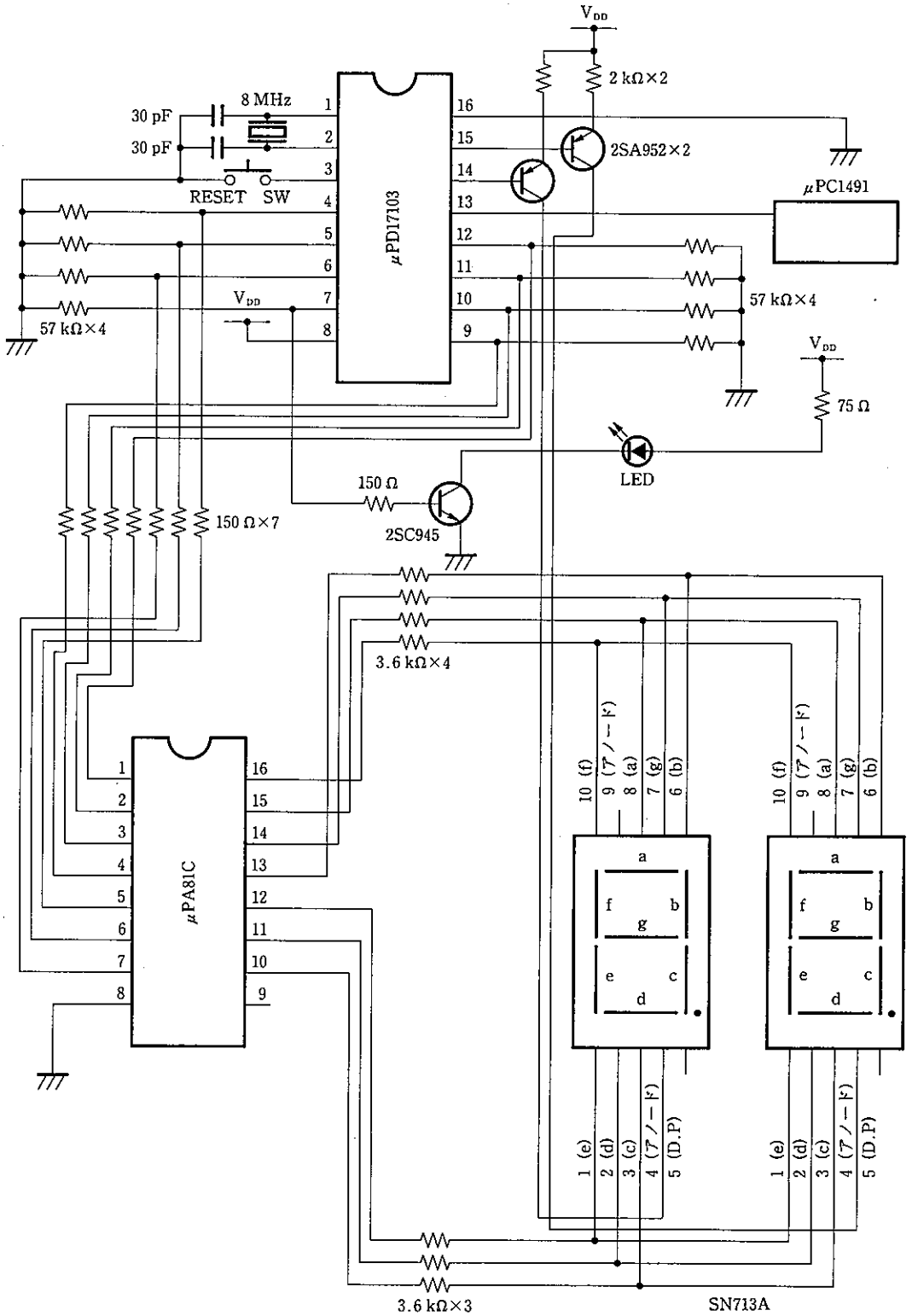
2.1.5　タイマ・カウント動作中におけるLEDの点灯

タイマがカウント動作中のときは，LEDを点灯します。

2.2　ハードウエアの構成

2.2.1　回 路 例

図2－12　回 路 図



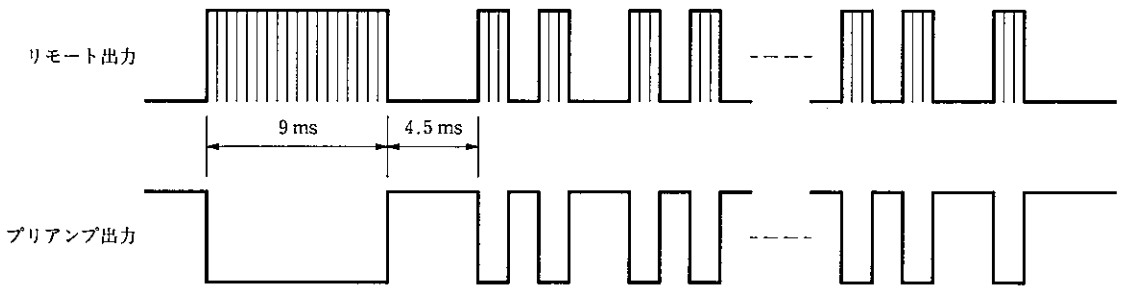
2.2.2　端子説明

端子番号	記　　号	端子名称	機　　能	出力形式
1 2	X_{IN} X_{OUT}	水晶発振子	水晶発振子の接続端子です。 8 MHzの水晶発振子を接続します。	―――
3	$\overline{RESET}$	リセット	システム・リセット用の入力端子です。マスク・オプションで内蔵プルアップ抵抗を選択します。	入力
4 ～ 6 9 ～ 12	SEG1 ～SEG3 ($P0D_0$ ～$P0D_2$) SEG4 ～SEG7 ($P0C_0$ ～$P0C_3$)	7セグメント LED セグメント 出力	7セグメントLEDの表示を制御する端子です。 7セグメントLEDのセグメント端子に接続します。 電源投入時およびリセット時は，ハイ・インピーダンス状態になりますので，プルダウン抵抗を接続します。	CMOS プッシュプル
7	$\overline{LED}$ ($P0D_3$)	LED制御	タイマ・カウント動作中に点灯させるLEDを制御する端子です。 電源投入時およびリセット時は，ハイ・インピーダンス状態になりますので，プルダウン抵抗を接続します。	CMOS プッシュプル
8	V_{DD}	電源入力	デバイスの電源入力端子です。 デバイスの動作時には，5 V±10%の電圧を供給します。	―――
13	$\overline{RECEPT}$ ($P0B_0$)	リモコン受信	リモコンの送信データを受信する端子です。 プリアンプ（μPC1491）の出力端子と接続します。 マスク・オプションでプルアップ抵抗内蔵入出力を選択してください。	N-ch オープン ・ドレイン ＋ プルアップ 抵抗内蔵 入出力
14 15	$\overline{COM1}$ ($P0B_1$) $\overline{COM2}$ ($P0B_2$)	7セグメント LED コモン出力	7セグメントLEDの表示を制御する端子です。 7セグメントLEDのコモン端子に接続します。 マスク・オプションでプルアップ抵抗内蔵入出力を選択してください。	N-ch オープン ・ドレイン ＋ プルアップ 抵抗内蔵 入出力
16	GND	グランド	グランド端子です。 （－）側電源に接続します。	―――

2.3　リモート出力波形，受信用プリアンプ出力波形

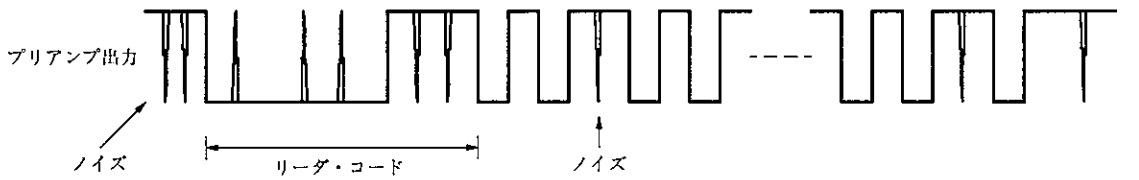
リモート出力波形とそのリモート波形を受信したときのリモコン・プリアンプ（アクティブ・ロウ）からの出力波形は，図2-13のようになります。

図2－13



しかし，実際には，蛍光灯などの外来光により，多数のノイズが現れます（図2-14参照）。

図2－14



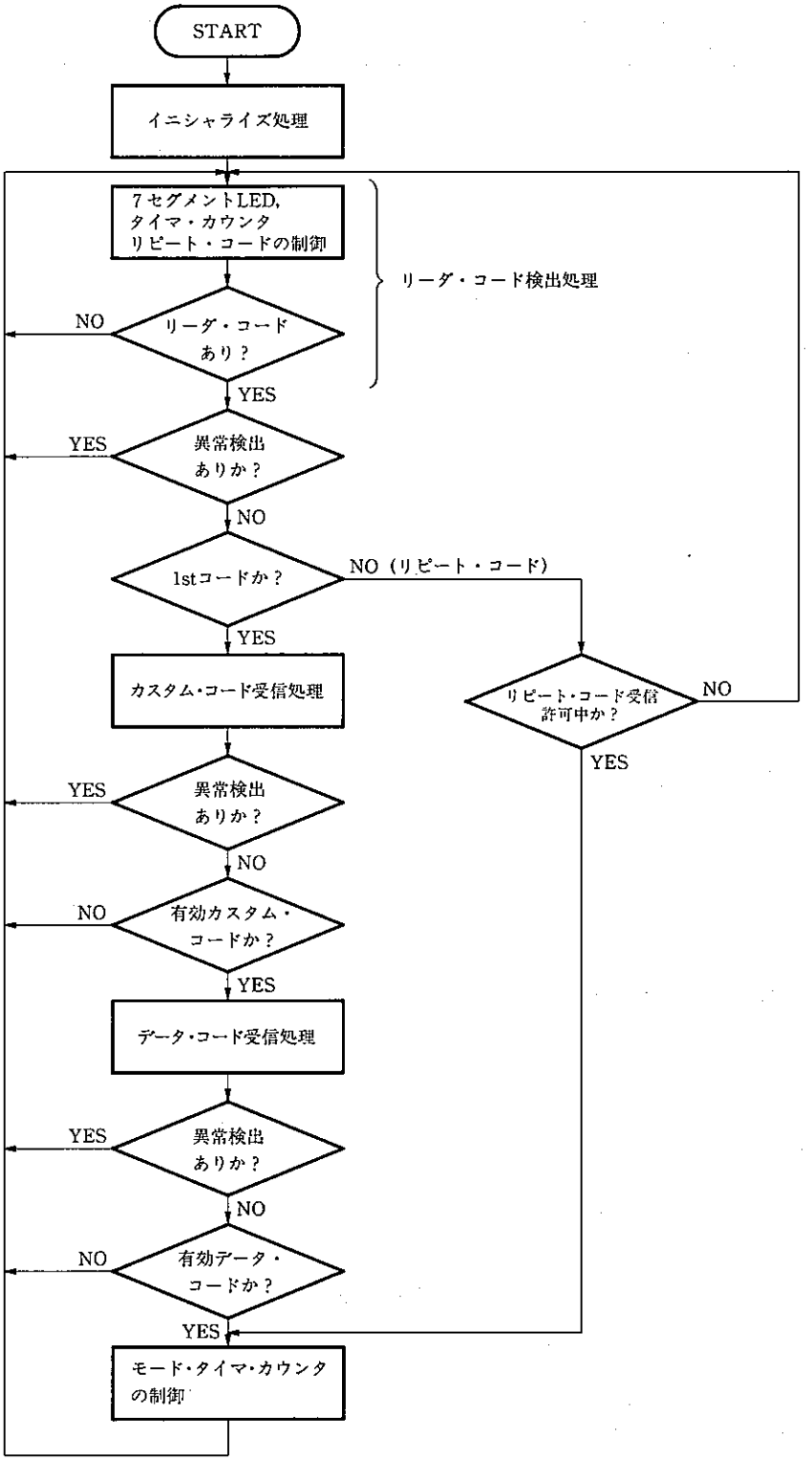
このノイズの中からコードを判別するために，たとえばリーダ・コードの9 ms間，ロウ・レベルが出力されるところを，6 ms以上ロウ・レベルが連続すればリーダ・コードとみなしたり，立ち下がったあと2 msごとにチェックしてロウ・レベルが3回連続していたら，そのあとは立ち上がりエッジを検出するのみとし，その立ち下がりから立ち上がりまでの時間をチェックすることで判断をするなどプログラミング上配慮が必要です。

2.4　ソフトウエア構成

2.4.1　ジェネラル・フロー・チャート

本プログラムのジェネラル・フロー・チャートを図2-15に示します。

図2－15　ジェネラル・フロー・チャート



注）　リーダ・コード検出処理中以外は，100 μsごとに7セグメントLED，タイマ・カウンタ，リピート・コードの制御を行います。
また異状検出とは，NECフォーマット以外のリモコンの送信パターンを受信したことを示します。

2.4.2　データ・メモリの割り付け

データ・メモリは，次に示すように割り付けます。

記　号	番　地	説　　明
WORK	00H	ワーク・エリアとして使用します。
CNT100	01H	100 μs単位のカウンタです。
MCNT1	02H	10分単位のタイマ・カウンタです。
MCNT2	03H	1分単位のタイマ・カウンタです。
SCNT1	04H	10秒単位のタイマ・カウンタです。
SCNT2	05H	1秒単位のタイマ・カウンタです。
MUSCNT1 MUSCNT2 MUSCNT3	06H 07H 08H.3 08H.2	ms単位のタイマ・カウンタです。 （なお，MUSCNT3の下位2ビットは，タイマ・カウンタには使用しません。）
COUNT_F	08H.1	タイマを制御するフラグとして使用します。 “1”：タイマがカウント中であることを示します。 “0”：タイマが停止中であることを示します。
MODE_F	08H.0	モードを示すフラグとして使用します。 “1”：タイマ動作モードであることを示します。 “0”：タイマ設定モードであることを示します。
INCNT	09H	二通りに使用します。 (1)　カスタム・コード，データ・コード受信時は，受信ビット数を制御するカウンタです。 (2)　リピート・コード受信時は，リピート・コードの受信数をカウントするカウンタです。
DATA1	0AH	二通りに使用します。 (1)　カスタム・コード，データ・コード受信時は，カスタム・コード，データ・コードの最初の4ビット目までの状態を格納するエリアです。 (2)　リピート・コード受信許可状態では，216 msをカウントするカウンタです。
DATA2	0BH	二通りに使用します。 (1)　カスタム・コード，データ・コード受信時は，カスタム・コード，データ・コードの5ビット目から8ビット目までの状態を格納するエリアです。 (2)　リピート・コード受信許可状態では，216 msをカウントするカウンタです。
CNT1 CNT2	0CH 0DH	受信コードの状態を判定する100 μs単位のカウンタです。
FLAG1	0EH.3 0EH.2 0EH.1	216 ms間に受信したリピート・コード数をカウントするカウンタです。 （なお，下位1ビットはカウンタとして使用しません。）
SEL_F	0EH.0	1 ms単位の制御に使用するフラグで，1 msごとに状態が反転します。

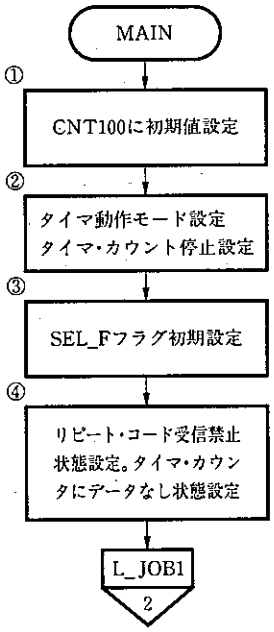
記　号	番　地	説　　明
CD_F	0FH.3	受信コードを示すフラグとして使用します。 “1”：データ・コードを受信中であることを示します。 “0”：カスタム・コードを受信中であることを示します。
ROK_F	0FH.2	リピート・コードの有効，無効を示すフラグとして使用します。 “1”：受信したリピート・コードが有効であることを示します。 “0”：受信したリピート・コードが無効であることを示します。
UP_F	0FH.1	三通りに使用します。 (1)　データ・コード，カスタム・コード受信時は，受信した１ビットの状態を示すフラグです。 “1”：コード“1”を受信したことを示します。 “0”：コード“0”を受信したことを示します。 (2)　データ・コード，カスタム・コード比較時は，比較結果を示すフラグです。 “1”：比較結果が一致したことを示します。 “0”：比較結果が不一致であることを示します。 (3)　受信データ・コードの種類を示すフラグです。 “1”：01Hのデータ・コードを受信したことを示します。 “0”：02Hのデータ・コードを受信したことを示します。
SET_F	0FH.0	タイマを制御するフラグとして使用します。 “1”：新たにタイマ・カウンタにデータを設定した状態を示します。 “0”：タイマ・カウンタのデータの変更を行っていない状態を示します。

2.4.3　ディテール・フロー・チャート

(1)　初 期 化

電源立ち上げ時およびリセット時は，μPD17103のデータ・メモリの内容は不定となっています。そのために電源立ち上げ時およびリセット時には，データ・メモリの内容を新たに設定する必要があります。

フロー・チャート 1



注1)　← 飛び先のレーベル
← 飛び先のフロー・チャート番号

注2)　( )　( ) 内の数字はフロー・チャート番号

```
MAIN :
        MOV   CNT100, #0 ──────────①  リーダ・コードのロウ・レベルの状態をカウントするCNT100を0に初期設定します。
        CLR2  MODE_F, COUNT_F ─────②  MODE_Fフラグ，COUNT_Fフラグをそれぞれ"0"に設定することにより，モードをタイマ動作モードのカウント停止状態にします。
        CLR1  SEL_F ───────────────③  SEL_Fフラグに "0" をセットします。
        CLR2  ROK_F, SET_F ────────④  ROK_Fフラグ，SET_Fフラグにそれぞれ"0"をセットして，リピート・コード受信禁止状態に，タイマ・カウンタに新たにデータの設定を行っていない状態にします。
```

(2)　リーダ・コード検出処理

リーダ・コードの検出は，2.0 msごとに，P0B$_0$端子をチェックすることによって行い，ロウ・レベルが3回連続すればリーダ・コードと判断します。

このとき，P0B$_0$端子が途中でハイ・レベルになればノイズとみなして，リーダ・コードの検出処理を最初から行います。

リーダ・コードと判断したら，次に0.1 msごとに立ち上がりの検出を行い，立ち上がりを検出したら，ハイ・レベルが連続するのを確認し，次に立ち下がりの検出を行います。この立ち上がりから立ち下がりまでの時間により，表2-3のようにコードの判定をします。

図2-16

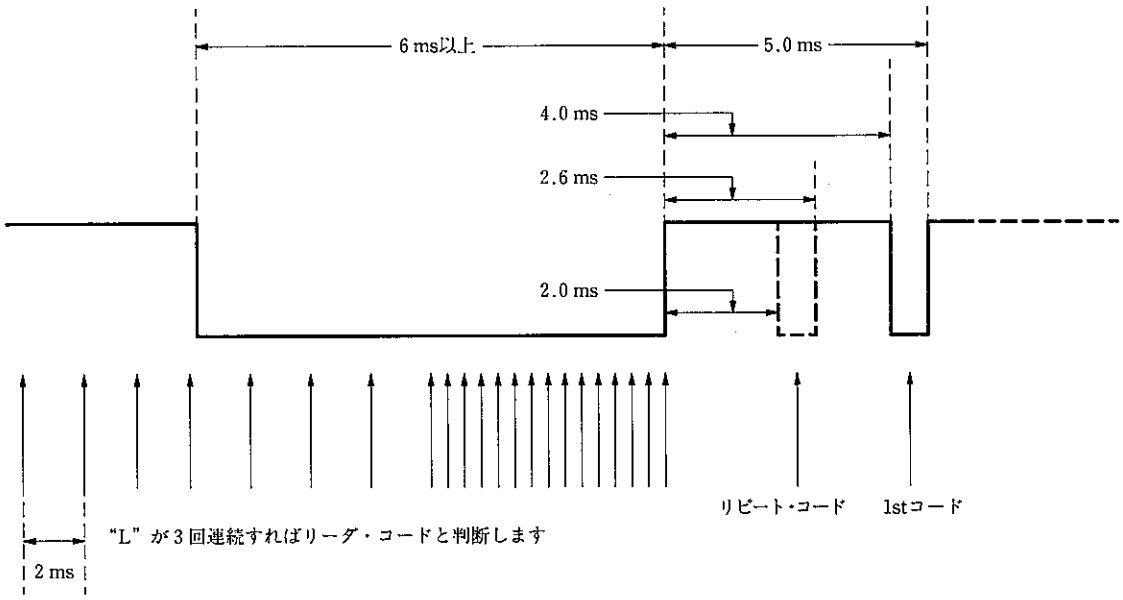
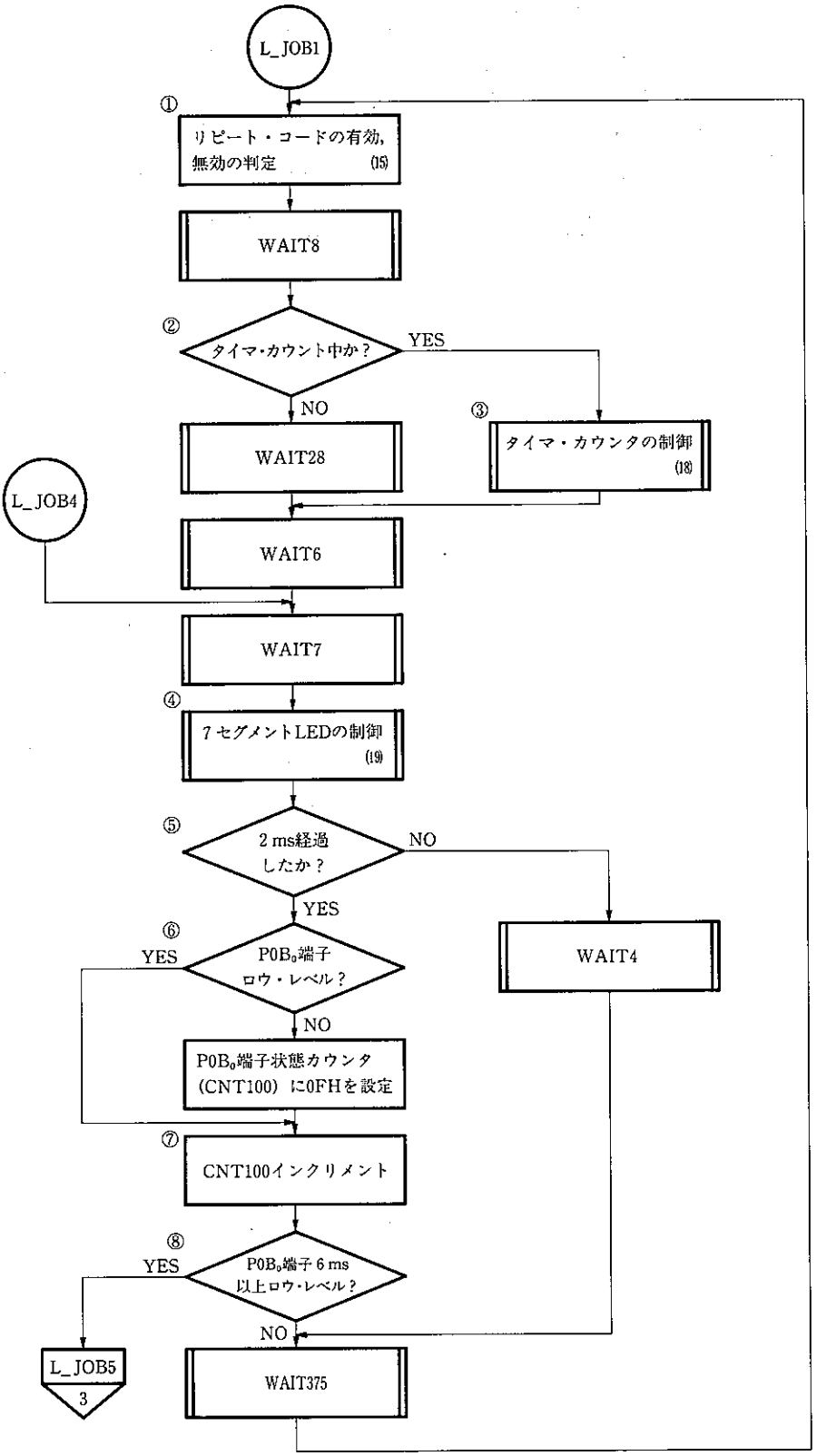


表2-3　受信コードの判定

立ち上がりから立ち下がりまでの時間	判　　定
2.0 ms未満	異常検出
2.0 ms以上2.6 ms未満	リピート・コード
2.6 ms以上4.0 ms未満	異常検出
4.0 ms以上5.0 ms未満	カスタム・コード
5.0 ms以上	異常検出

フロー・チャート 2



注)　リーダ・コード検出処理中は，図2－15に示すルーチンをループして処理を行います。L_JOB1から処理を行い，再びL_JOB1に戻るまでの実行ステップは500ステップ(1 ms)になるように設定してあります。

```
L_JOB1：
        CALL    RECNT ──────────────①
        CALL    WAIT8
        SKT1    COUNT_F }──────────②
        BR      L_JOB2  }
        CALL    COUNT1 ─────────────③
        BR      L_JOB3
L_JOB2：
        CALL    WAIT10
        CALL    WAIT10
        CALL    WAIT8
L_JOB3：
        CALL    WAIT6
L_JOB4：
        CALL    WAIT7
        CALL    SEGMENT_JOB ────────④
        SKT1    SEL_F   }──────────⑤
        BR      L_CHECK }
        CALL    WAIT4
        BR      WAIT_375
L_CHECK：
        SKF1    P0B0    }──────────⑥
        MOV     CNT100,#0FH }
        ADD     CNT100,#1 ──────────⑦
        SKNE    CNT100,#4 }────────⑧
        BR      L_JOB5    }
WAIT_375：
        MOV     WORK,#0DH
L_LOOP：
        CALL    WAIT10
        CALL    WAIT10
        CALL    WAIT5
        SUB     WORK,#1
        SKT1    Z
        BR      L_LOOP
        CALL    WAIT9
        BR      L_JOB1
```

① リピート・コードが受け付け許可状態なのかどうか，受信許可状態であればリピート・コードの有効，無効を判定します。

② タイマがカウント中かを判定します。タイマがカウント中でない（COUNT_F＝0）ときは，L_JOB2にとびます。

③ タイマ・カウンタの制御を行います。

④ 7セグメントLEDの制御を行います。

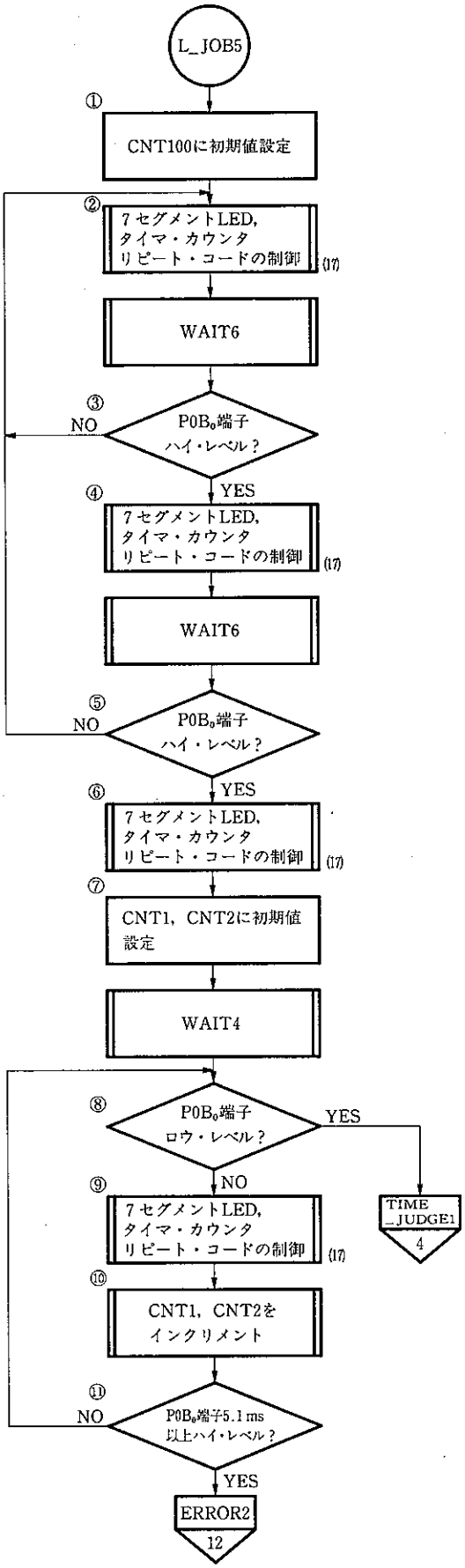
⑤ 2 msごとにL_CHECKにとびP0B$_0$端子の状態をチェックします。

⑥ P0B$_0$端子の状態をチェックします。P0B$_0$端子がハイ・レベルのとき，ロウ・レベル状態が何回続いたかをカウントするCNT100を0FHにします。

⑦ CNT100をインクリメントします。

⑧ P0B$_0$端子の状態が6 ms以上ロウ・レベルか判定します。6 ms以上ロウ・レベルのときはリーダ・コードを受信したとしてL_JOB5にとびます。

フロー・チャート3


L_JOB5
①
CNT100に初期値設定
②
7セグメントLED,
タイマ・カウンタ
リピート・コードの制御
(17)
WAIT6
③
NO
P0B0端子
ハイ・レベル？
YES
④
7セグメントLED,
タイマ・カウンタ
リピート・コードの制御
(17)
WAIT6
⑤
NO
P0B0端子
ハイ・レベル？
YES
⑥
7セグメントLED,
タイマ・カウンタ
リピート・コードの制御
(17)
⑦
CNT1，CNT2に初期値
設定
WAIT4
⑧
P0B0端子
ロウ・レベル？
YES
TIME
_JUDGE1
4
NO
⑨
7セグメントLED,
タイマ・カウンタ
リピート・コードの制御
(17)
⑩
CNT1，CNT2を
インクリメント
⑪
NO
P0B0端子5.1 ms
以上ハイ・レベル？
YES
ERROR2
12

```
L_JOB5：
        MOV     CNT100,#0 ──────────①
L_JOB6：
        CALL    TIME_CTL4 ──────────②
        CALL    WAIT6
        SKT1    P0B0      }
        BR      L_JOB6    } ────────③
        CALL    TIME_CTL4 ──────────④
        CALL    WAIT6
        SKT1    P0B0      }
        BR      L_JOB6    } ────────⑤
        CALL    TIME_CTL4 ──────────⑥
        MOV     CNT2,#0EH }
        MOV     CNT1,#0CH } ────────⑦
        CALL    WAIT4
L_JOB7：
        SKT1    P0B0          }
        BR      TIME_JUDGE1   } ────⑧
        CALL    TIME_CTL4 ──────────⑨
        CALL    CNT_UP ─────────────⑩
        BR      L_JOB7    }
        BR      ERROR2    } ────────⑪
```

①　7セグメントLED，タイマ・カウンタ，リピート・コードの制御を行うCNT100の値を0に設定します。

②　7セグメントLED，タイマ・カウンタ，リピート・コードの制御を行います。

③　$P0B_0$端子の状態をチェックします。

④　7セグメントLED，タイマ・カウンタ，リピート・コードの制御を行います。

⑤　$P0B_0$端子の状態をチェックします。ハイ・レベルを検出すると2回続けてハイ・レベルを検出したとして立ち下がりの検出に処理を移します。

⑥　7セグメントLED，タイマ・カウンタ，リピート・コードの制御を行います。

⑦　リーダ・コードの立ち上がりから次のコードの立ち下がりの時間をカウントするためのCNT1，CNT2に初期値を設定します。

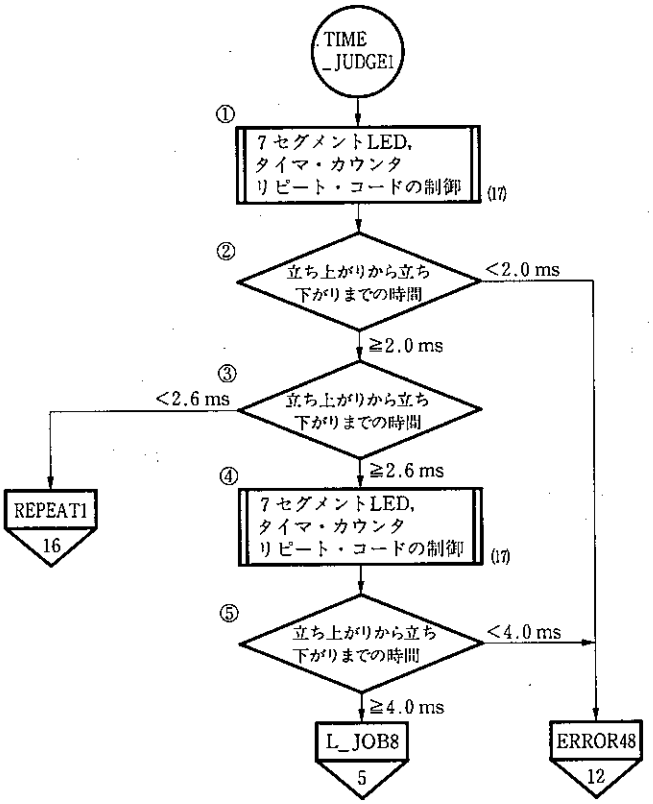
⑧　$P0B_0$端子の状態をチェックします。ロウ・レベルを検出すると，TIME_JUDGE1にとびます。

⑨　7セグメントLED，タイマ・カウンタ，リピート・コードの制御を行います。

⑩　100 msごとにCNT1，CNT2の制御を行います。

⑪　$P0B_0$端子が5.1 ms以上ハイ・レベルのときは，異常検出としてERROR2にとびます。

フロー・チャート4


TIME
_JUDGE1
①
7セグメントLED,
タイマ・カウンタ
リピート・コードの制御
(17)
②
立ち上がりから立ち
下がりまでの時間
<2.0 ms
≧2.0 ms
③
<2.6 ms
立ち上がりから立ち
下がりまでの時間
≧2.6 ms
④
7セグメントLED,
タイマ・カウンタ
リピート・コードの制御
(17)
⑤
立ち上がりから立ち
下がりまでの時間
<4.0 ms
≧4.0 ms
REPEAT1
16
L_JOB8
5
ERROR48
12

```
TIME_JUDGE1 :
        CALL    TIME_CTL4 ──────────①  ７セグメントLED，タイマ・カウンタ，リピート・
                                       コードの制御を行います。
        SUB     CNT2,#1   }
        SUBC    CNT1,#0EH  ├──────②  立ち上がりから立ち下がりまでの時間を判定します。
        SKF1    CY         │          2.0 ms未満のときは，異常検出としてERROR48にと
        BR      ERROR48   ┘           びます。
        SUB     CNT2,#6   }
        SUBC    CNT1,#0    ├──────③  立ち上がりから立ち下がりの時間を判定します。2.6
        SKF1    CY         │          ms未満のときは，リピート・コードとしてREPEAT1
        BR      REPEAT1   ┘           にとびます。
        CALL    TIME_CTL4 ──────────④  ７セグメントLED，タイマ・カウンタ，リピート・
                                       コードの制御を行います。
        SUB     CNT2,#0EH }
        SUBC    CNT1,#0    ├──────⑤  立ち上がりから立ち下がりの時間を判定します。4.0
        SKF1    CY         │          ms未満のときは，異常検出としてERROR48にとび
        BR      ERROR48   ┘           ます。
```

(3)　データ受信処理

データの受信処理は，以下のようにして行います。立ち下がりを検出したら立ち上がりを検出し，再び立ち下がりを検出します。前の立ち下がりから，この立ち下がりの時間により表2－4のようにコードを判定します。

図2－17　データ・コード波形

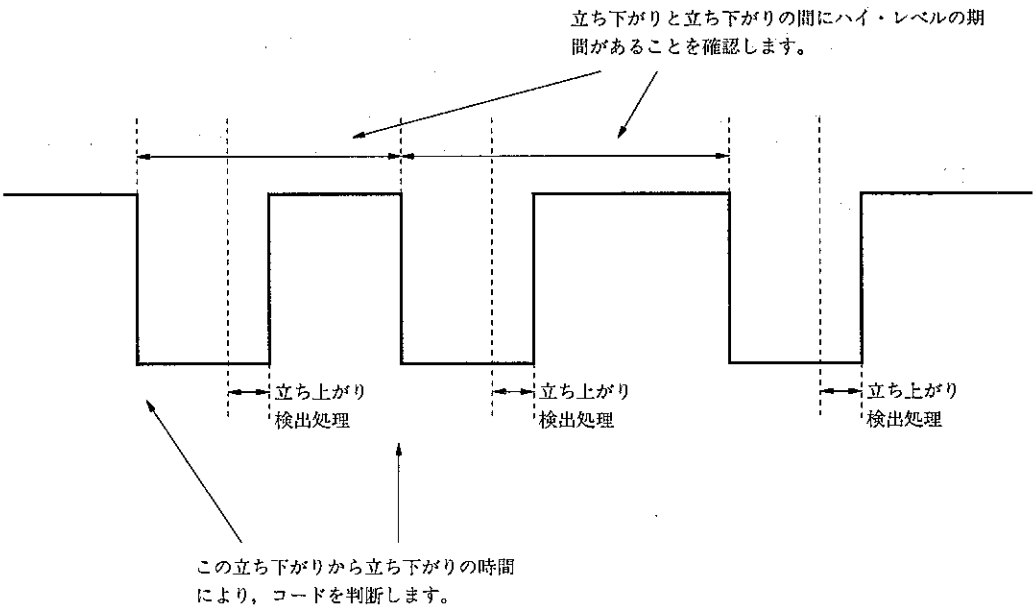
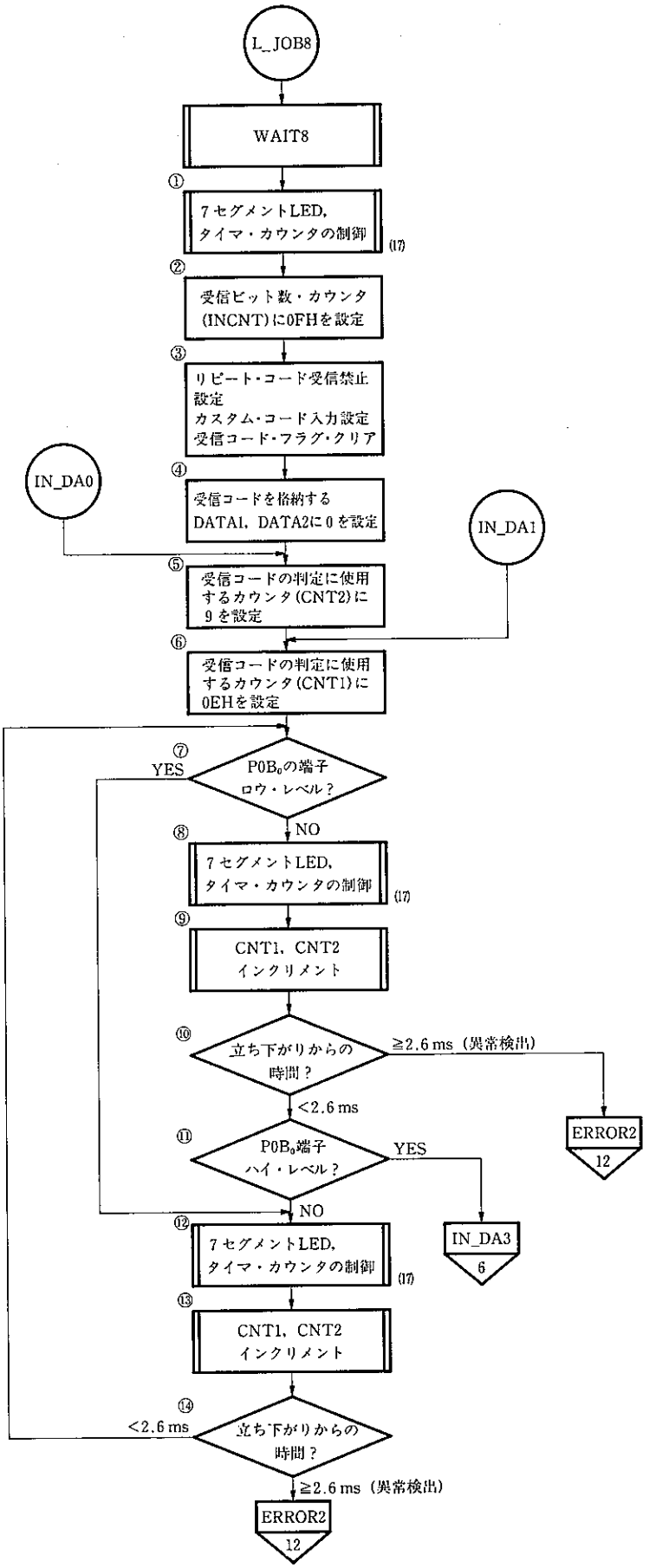


表2－4　受信コードの判定

立ち下がりから立ち下がりまでの時間	判　　定
1.0 ms未満	異常検出
1.0 ms以上1.4 ms未満	受信データ“0”
1.4 ms以上2.0 ms未満	異常検出
2.0 ms以上2.6 ms未満	受信データ“1”
2.6 ms以上	異常検出

立ち下がりから立ち下がりまでの時間が1.0 ms以上1.4 ms未満のときは，データ・コード“0”を受信したと判断します。また，2.0 ms以上2.6 ms未満のときは，データ・コード“1”を受信したと判断します。それ以外のときは，異常検出として再びリーダ・コードの検出処理を行います。

フロー・チャート 5


L_JOB8
WAIT8
①
7セグメントLED,
タイマ・カウンタの制御
(17)
②
受信ビット数・カウンタ
(INCNT)に0FHを設定
③
リピート・コード受信禁止
設定
カスタム・コード入力設定
受信コード・フラグ・クリア
IN_DA0
④
受信コードを格納する
DATA1, DATA2に0を設定
IN_DA1
⑤
受信コードの判定に使用
するカウンタ(CNT2)に
9を設定
⑥
受信コードの判定に使用
するカウンタ(CNT1)に
0EHを設定
⑦
YES
P0B0の端子
ロウ・レベル?
NO
⑧
7セグメントLED,
タイマ・カウンタの制御
(17)
⑨
CNT1, CNT2
インクリメント
⑩
立ち下がりからの
時間?
≧2.6 ms (異常検出)
ERROR2
12
<2.6 ms
⑪
P0B0端子
ハイ・レベル?
YES
IN_DA3
6
NO
⑫
7セグメントLED,
タイマ・カウンタの制御
(17)
⑬
CNT1, CNT2
インクリメント
⑭
<2.6 ms
立ち下がりからの
時間?
≧2.6 ms (異常検出)
ERROR2
12

```
L_JOB8：
        CALL    WAIT8
        CALL    TIME_CTL ――――――――――――①
        MOV     INCNT,＃0FH ――――――――――②
        CLR3    ROK_F,CD_F,UP_F ――――――――③
        MOV     DATA1,＃0 }――――――――――④
        MOV     DATA2,＃0 }
        MOV     CNT2,＃9 ――――――――――――⑤
IN_DA0：
        MOV     CNT1,＃0EH ――――――――――⑥
IN_DA1：
        SKT1    P0B0 }――――――――――――⑦
        BR      IN_DA2 }
        CALL    TIME_CTL4 ――――――――――――⑧
        CALL    CNT_UP ――――――――――――⑨
        BR      CHA確認 }――――――――――⑩
        BR      ERROR2 }
CHA確認：
        SKF1    P0B0 }――――――――――――⑪
        BR      IN_DA3 }
IN_DA2：
        CALL    TIME_CTL4 ――――――――――――⑫
        CALL    CNT_UP ――――――――――――⑬
        BR      IN_DA1 }――――――――――⑭
        BR      ERROR2 }
```

① 7セグメントLED，タイマ・カウンタの制御を行います。

② カスタム・コード，データ・コードの受信ビット数を示すINCNTに0FHを設定します。

③ ROK_F，CD_F，UP_Fに“0”をセットすることにより，リピート・コード受信禁止状態に，カスタム・コード受信状態に，また受信コード状態を示すUP_Fをリセット状態に設定します。

④ 受信コードを格納するDATA1，DATA2に0を設定します。

⑤ 立ち下がりから次の立ち下がりまでの時間をカウントするCNT2に初期値を設定します。

⑥ 立ち下がりから次の立ち下がりまでの時間をカウントするCNT1に初期値を設定します。

⑦ $P0B_0$端子の状態をチェックします。ロウ・レベルのときは，IN_DA2にとびます。

⑧ 7セグメントLED，タイマ・カウンタの制御を行います。

⑨ 100 msごとにCNT1，CTN2の制御を行います。

⑩ $P0B_0$端子の状態が2.6 ms以内に変化しないとき異常検出としてERROR2にとびます。

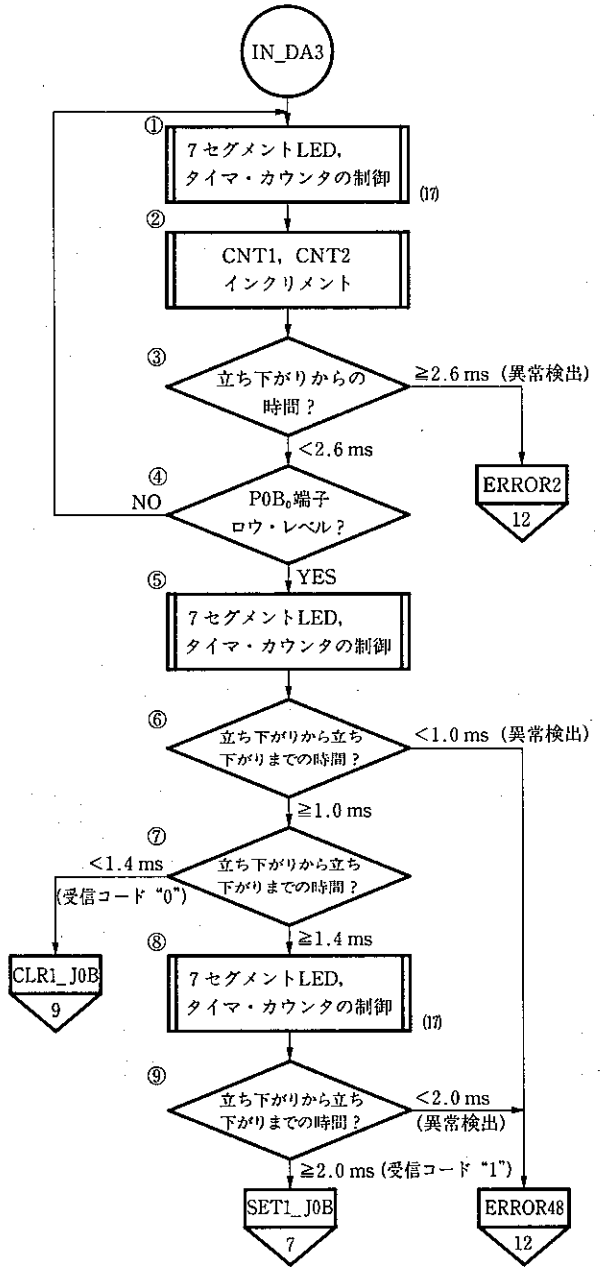
⑪ $P0B_0$端子の状態をチェックします。ハイ・レベルのとき，2回続けてハイ・レベルを検出したとして，立ち下がりの検出に処理を移します。

⑫ 7セグメントLED，タイマ・カウンタの制御を行います。

⑬ 100 msごとにCNT1，CNT2の制御を行います。

⑭ $P0B_0$端子の状態が2.6 ms以内にロウ・レベルに変化しないときは，異常検出としてERROR2にとびます。

フロー・チャート 6

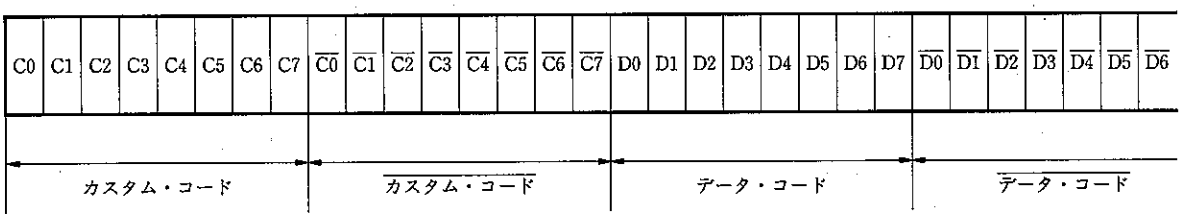

IN_DA3
①
7セグメントLED,
タイマ・カウンタの制御
(17)
②
CNT1, CNT2
インクリメント
③
立ち下がりからの
時間？
≧2.6 ms（異常検出）
<2.6 ms
④
NO
P0B0端子
ロウ・レベル？
ERROR2
12
YES
⑤
7セグメントLED,
タイマ・カウンタの制御
⑥
立ち下がりから立ち
下がりまでの時間？
<1.0 ms（異常検出）
≧1.0 ms
⑦
<1.4 ms
(受信コード "0")
立ち下がりから立ち
下がりまでの時間？
CLR1_J0B
9
≧1.4 ms
⑧
7セグメントLED,
タイマ・カウンタの制御
(17)
⑨
立ち下がりから立ち
下がりまでの時間？
<2.0 ms
(異常検出)
≧2.0 ms (受信コード "1")
SET1_J0B
7
ERROR48
12

```
IN_DA3：
        CALL    TIME_CTL4 ──①   7セグメントLED，タイマ・カウンタの制御を行います。
        CALL    CNT_UP    ──②   100 msごとにCNT1，CNT2の制御を行います。
        BR      IN_DA4    }──③   P0B0端子の状態が2.6 ms以内にロウ・レベルに変化
        BR      ERROR2    }      しないときは，異常検出としてERROR2にとびます。
IN_DA4：
        SKF1    P0B0      }──④   P0B0端子の状態をチェックします。ハイ・レベルの
        BR      IN_DA3    }      ときは，IN_DA3にとびます。
        CALL    TIME_CTL4 ──⑤   7セグメントLED，タイマ・カウンタの制御を行います。
        SUB     CNT2,#0   }──⑥   立ち下がりから立ち下がりまでの時間を判定します。
        SUBC    CNT1,#0FH }      1.0 ms未満のときは異常検出として，ERROR48にとびます。
        SKF1    CY        }
        BR      ERROR48   }
        SUB     CNT2,#3   }──⑦   立ち下がりから立ち下がりまでの時間を判定します。
        SUBC    CNT1,#0   }      1.4 ms未満のときはコード"0"を受信したとしてCLR1 JOBにとびます。
        SKF1    CY        }
        BR      CLR1_JOB  }
        CALL    TIME_CTL4 ──⑧   7セグメントLED，タイマ・カウンタの制御を行います。
        SUB     CNT2,#7   }──⑨   立ち下がりから立ち下がりまでの時間を判定します。
        SUBC    CNT1,#0   }      2.0 ms未満のときは異常検出として，ERROR48にとびます。
        SKF1    CY        }
        BR      ERROR48   }
```

(4)　データ格納，比較処理

カスタム・コードおよびデータ・コードは，図2-18のように8ビットのコードとその逆コードにより構成されます。

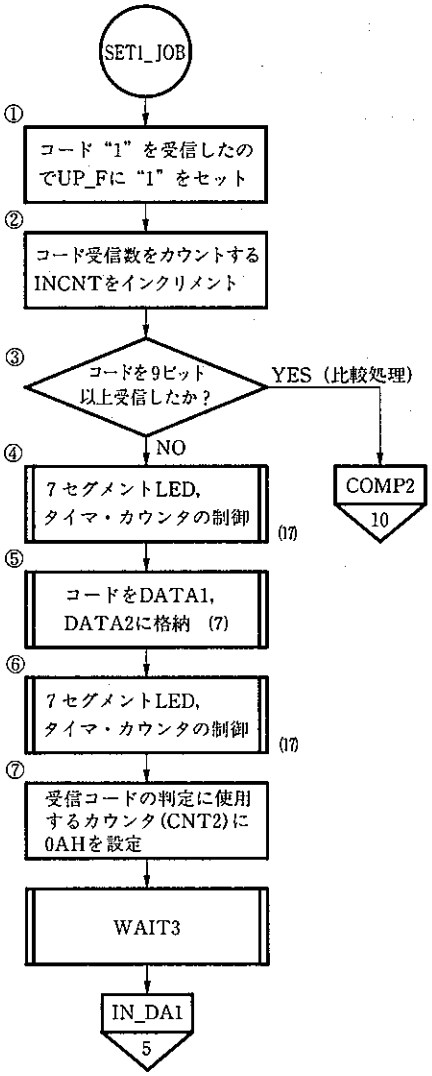
図2-18　カスタム・コード，データ・コード構成



まず，８ビットのカスタム・コードをデータ・メモリDATA1，DATA2に格納します。８ビットのカスタム・コードを格納後，続いて受信するカスタム・コードを１ビット受信するごとにすでに受信したカスタム・コードの値と確実に反転をしているか検証します。つまり，カスタム・コードのある１ビットとそれに相当するカスタム・コードのある１ビットを比較検証を行った結果が不一致のときは，正常にカスタム・コードが受信されているとして，次のカスタム・コードの１ビットを受信し再び比較検証を行います。比較を行った結果が一致のときは，正しくカスタム・コードが受信できなかったとして異常検出として処理し再びリーダ・コードの検出処理を行います。

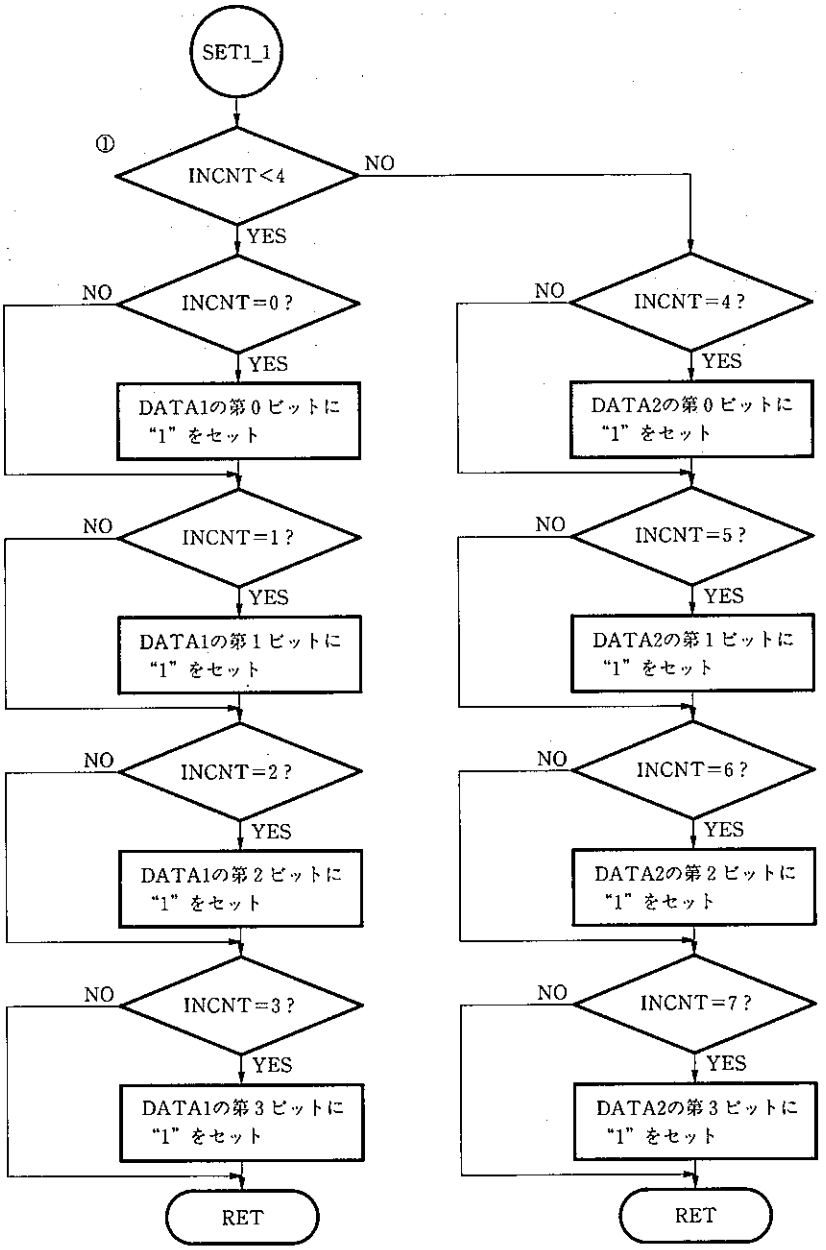
さらに８ビットの逆コードを受信を完了すると，受信されたカスタム・コードが有効なカスタム・コードであるかの判定を行います。本プログラムにおいては，有効なカスタム・コードは04Hと考えていますので，受信したカスタム・コードが04H以外のときは，受信したカスタム・コードを無効として再びリーダ・コードの検出処理を行います。04Hのカスタム・コードを受信したときは，データ・コードをカスタム・コードと同じ方法で受信します。

フロー・チャート7


SET1_JOB
①
コード“1”を受信したのでUP_Fに“1”をセット
②
コード受信数をカウントするINCNTをインクリメント
③
コードを9ビット以上受信したか？
YES（比較処理）
NO
COMP2
10
④
7セグメントLED，タイマ・カウンタの制御
(17)
⑤
コードをDATA1，DATA2に格納 (7)
⑥
7セグメントLED，タイマ・カウンタの制御
(17)
⑦
受信コードの判定に使用するカウンタ(CNT2)に0AHを設定
WAIT3
IN_DA1
5

```
SET1_JOB :
        SET1    UP_F ──────────────────①   コード“1”を受信したとして，UP_Fに“1”をセットします。
        ADD     INCNT,#1 ──────────────②   コード受信数を示すINCNTをインクリメントします。
        SKLT    INCNT,#8 }─────────────③   コードを９ビット以上受信したか判断します。
        BR      COMP2    }                 ９ビット以上受信したときは，COMP2にとびます。
        CALL    TIME_CTL4 ─────────────④   ７セグメントLED，タイマ・カウンタの制御を行います。
        CALL    SET1_1 ────────────────⑤   コード“1”を受信したとしてDATA1，DATA2のある１ビットに“1”をセットします。
        CALL    TIME_CTL ──────────────⑥   ７セグメントLED，タイマ・カウンタの制御を行います。
        MOV     CNT2,#0AH ─────────────⑦   立ち下がりから立ち下がりまでの時間をカウントするCNT2に初期値を設定します。
        CALL    WAIT3
        BR      IN_DA0
```

フロー・チャート８


SET1_1
①
INCNT＜4
NO
YES
INCNT＝0？
DATA1の第０ビットに“1”をセット
INCNT＝1？
DATA1の第１ビットに“1”をセット
INCNT＝2？
DATA1の第２ビットに“1”をセット
INCNT＝3？
DATA1の第３ビットに“1”をセット
RET
INCNT＝4？
DATA2の第０ビットに“1”をセット
INCNT＝5？
DATA2の第１ビットに“1”をセット
INCNT＝6？
DATA2の第２ビットに“1”をセット
INCNT＝7？
DATA2の第３ビットに“1”をセット
RET

```
SET1_1:
        SKLT    INCNT,#4
        BR      SET1_2
        SKNE    INCNT,#0
        OR      DATA1,#0001B
        SKNE    INCNT,#1
        OR      DATA1,#0010B
        SKNE    INCNT,#2
        OR      DATA1,#0100B
        SKNE    INCNT,#3
        OR      DATA1,#1000B
        RET
SET1_2:
        SKNE    INCNT,#4
        OR      DATA2,#0001B
        SKNE    INCNT,#5
        OR      DATA2,#0010B
        SKNE    INCNT,#6
        OR      DATA2,#0100B
        SKNE    INCNT,#7
        OR      DATA2,#1000B
        RET
```

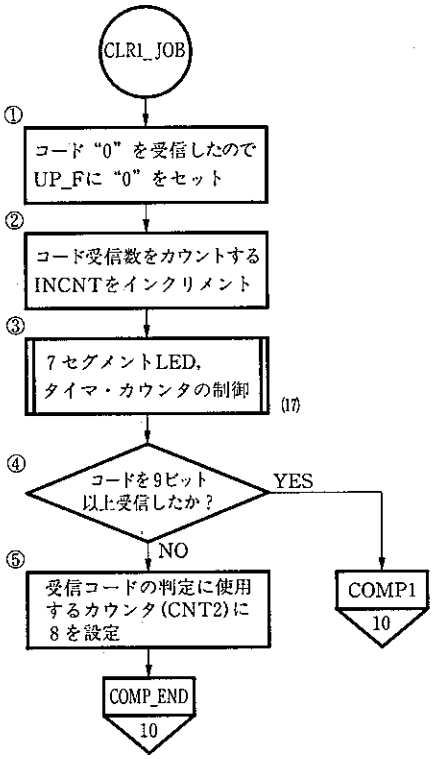
① INCNTの値によりDATA1または，DATA2のある1ビットに“1”をセットします。INCNTの値と“1”をセットする位置の関係は，表2－5に示すようになります。

表2－5　INCNTと“1”をセットする位置

INCNT	DATA2				DATA1			
	#3	#2	#1	#0	#3	#2	#1	#0
0								*
1							*	
2						*		
3					*			
4				*				
5			*					
6		*						
7	*							

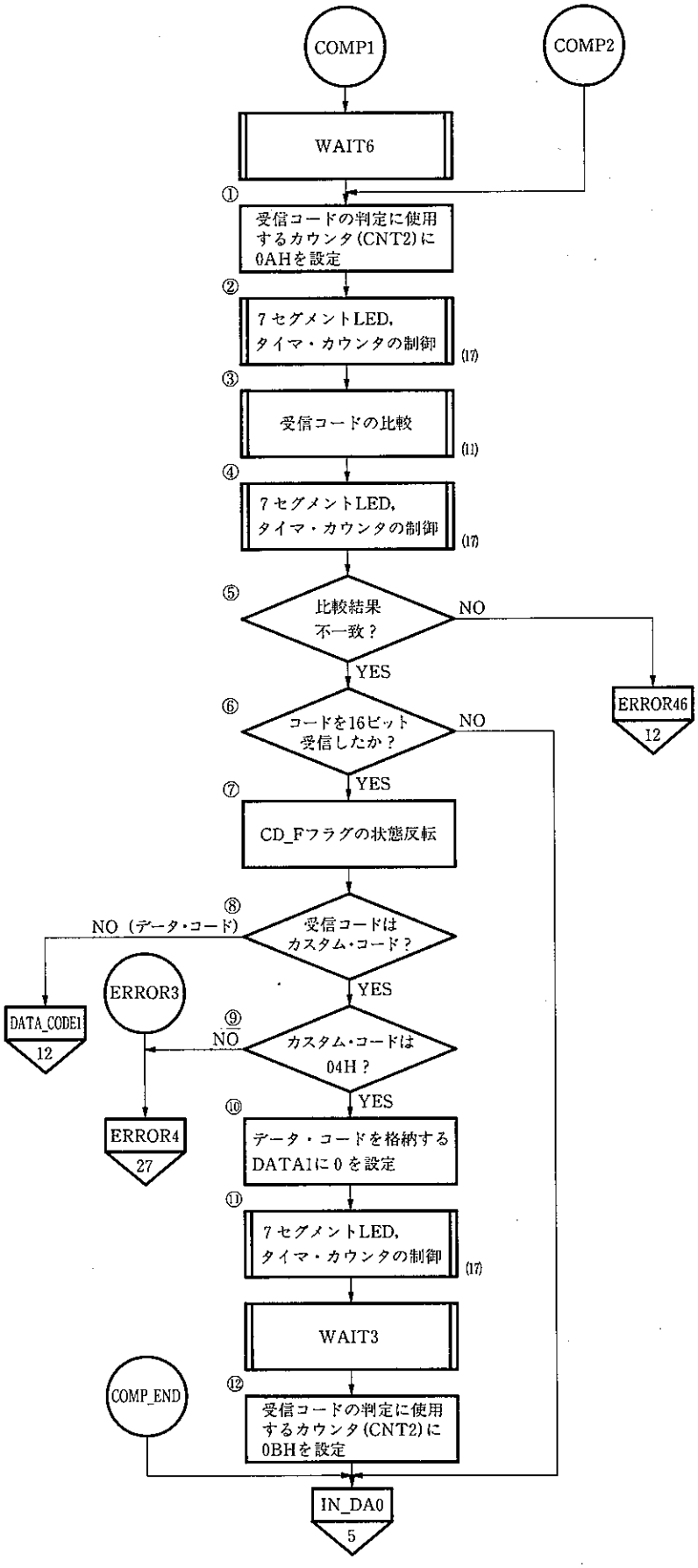
注）　＊は，“1”をセットする位置を示します。

フロー・チャート 9


CLR1_JOB
①
コード“0”を受信したので
UP_Fに“0”をセット
②
コード受信数をカウントする
INCNTをインクリメント
③
7セグメントLED,
タイマ・カウンタの制御
(17)
④
コードを9ビット
以上受信したか？
YES
NO
⑤
受信コードの判定に使用
するカウンタ(CNT2)に
8を設定
COMP1
10
COMP_END
10

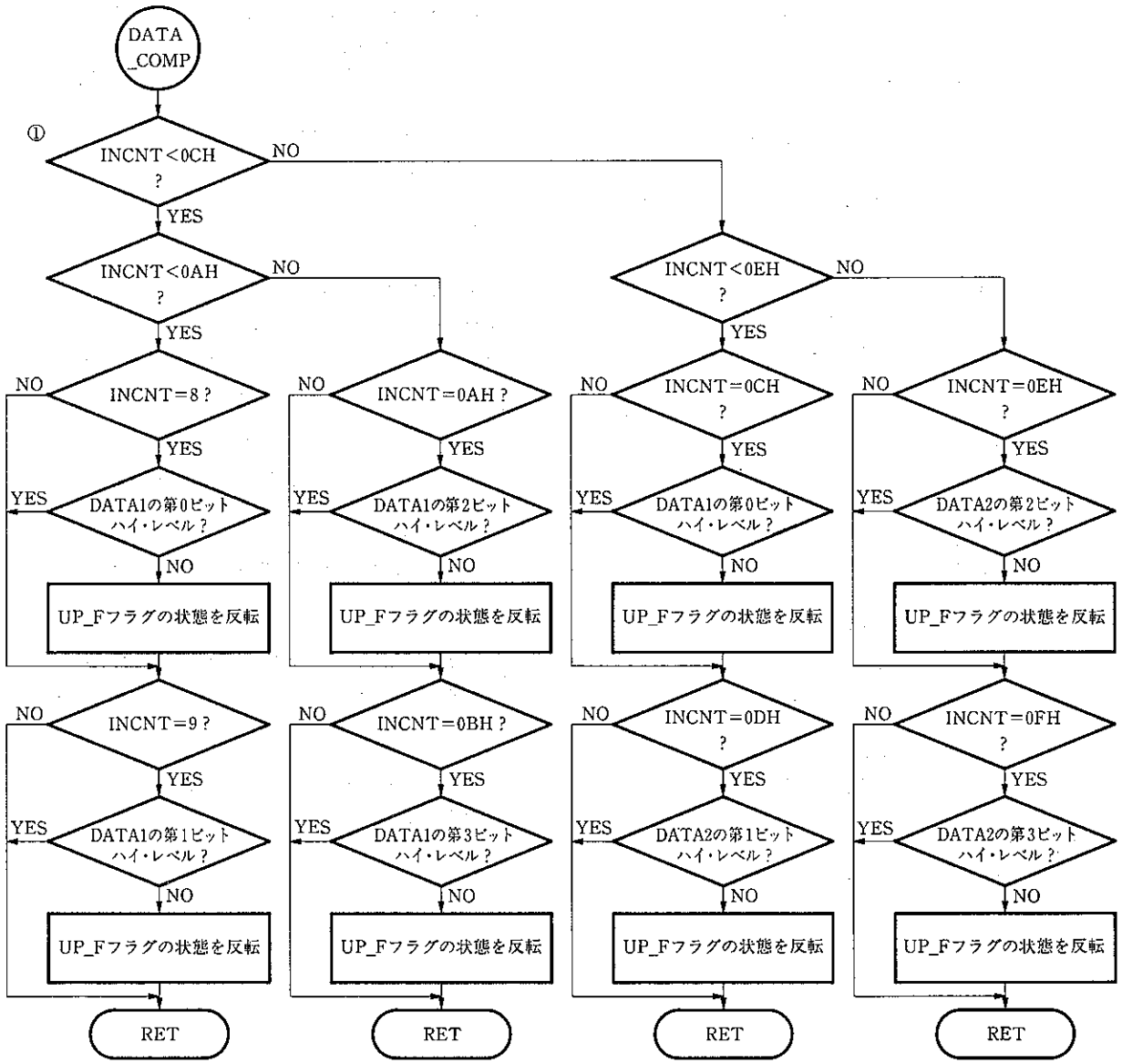
```
CLR1_JOB :
        CLR1    UP_F ──────────────────①  コード“0”を受信したとして，UP_Fフラグに“0”
                                          をセットします。
        ADD     INCNT,#1 ──────────────②  コード受信数を示すINCNTをインクリメントしま
                                          す。
        CALL    TIME_CTL2 ─────────────③  7セグメントLED，タイマ・カウンタの制御を行い
                                          ます。
        SKLT    INCNT,#8 }─────────────④  コードを9ビット以上受信したか判断します。
        BR      COMP1    }                9ビット以上受信したときは，COMP1にとびます。
        MOV     CNT2,#8 ───────────────⑤  立ち下がりから立ち下がりまでの時間をカウントす
                                          るCNT2に初期値を設定します。

        BR      COMP_END
```

フロー・チャート 10


COMP1
COMP2
WAIT6
①
受信コードの判定に使用するカウンタ(CNT2)に0AHを設定
②
7セグメントLED,タイマ・カウンタの制御
(17)
③
受信コードの比較
(11)
④
7セグメントLED,タイマ・カウンタの制御
(17)
⑤
比較結果不一致？
NO
YES
ERROR46
12
⑥
コードを16ビット受信したか？
NO
YES
⑦
CD_Fフラグの状態反転
⑧
受信コードはカスタム・コード？
NO（データ・コード）
YES
DATA_CODE1
12
ERROR3
⑨
カスタム・コードは04H？
NO
YES
ERROR4
27
⑩
データ・コードを格納するDATA1に0を設定
⑪
7セグメントLED,タイマ・カウンタの制御
(17)
WAIT3
COMP_END
⑫
受信コードの判定に使用するカウンタ(CNT2)に0BHを設定
IN_DA0
5

```
COMP1：
        CALL    WAIT6
COMP2：
        MOV     CNT2,＃0AH ―――――――――――①  立ち下がりから立ち下がりまでの時間をカウントするCNT2に初期値を設定します。
        CALL    TIME_CTL3 ―――――――――――②  7セグメントLED，タイマ・カウンタの制御を行います。
        CALL    DATA_COMP ―――――――――――③  受信したコードと８ビット前に受信したコードの比較を行います。
        CALL    TIME_CTL ―――――――――――④  7セグメントLED，タイマ・カウンタの制御を行います。
        SKF1    UP_F      ―――――――――――⑤  比較結果を判定します。比較が一致したときは，異常検出としてERROR46にとびます。
        BR      ERROR46
        SKE     INCNT,＃0FH ―――――――――⑥  コードを16ビット受信したか判断します。
        BR      COMP_END                    16ビット受信していないときは，COMP_ENDにとびます。
        NOT1    CD_F ―――――――――――⑦  CD_Fフラグの状態を反転します。
        SKT1    CD_F      ―――――――――――⑧  カスタム・コードの受信を終了したか判断します。
        BR      DATA_CODE1                  データ・コードの受信を終了した（CD_F＝0）ときは，DATA_CODE1にとびます。
        SKNE    DATA1,＃4 ―――――――――⑨  カスタム・コードが04Hか判定します。04H以外のコードを受信したときは，ERROR4にとびます。
        SKE     DATA2,＃0
ERROR3：
        BR      ERROR4
        MOV     DATA1,＃0 ―――――――――⑩  受信コードを格納するDATA1に０を設定します。
        CALL    TIME_CTL1 ―――――――――⑪  7セグメントLED，タイマ・カウンタの制御を行います。
        CALL    WAIT3
        MOV     CNT2,＃0BH ―――――――――⑫  立ち下がりから立ち下がりまでの時間をカウントするCNT2に初期値を設定します。
COMP_END：
        BR      IN_DA0
```

フロー・チャート 11


DATA
_COMP
①
INCNT＜0CH
?
NO
YES
INCNT＜0AH
?
NO
YES
INCNT＜0EH
?
NO
YES
NO
INCNT＝8？
YES
YES
DATA1の第0ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
NO
INCNT＝0AH？
YES
YES
DATA1の第2ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
NO
INCNT＝0CH
?
YES
YES
DATA1の第0ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
NO
INCNT＝0EH
?
YES
YES
DATA2の第2ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
NO
INCNT＝9？
YES
YES
DATA1の第1ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
NO
INCNT＝0BH？
YES
YES
DATA1の第3ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
NO
INCNT＝0DH
?
YES
YES
DATA2の第1ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
NO
INCNT＝0FH
?
YES
YES
DATA2の第3ビット
ハイ・レベル？
NO
UP_Fフラグの状態を反転
RET
RET
RET
RET

```
DATA_COMP :
        SKLT    INCNT , # 0CH
        BR      DATA_COMP11
        SKLT    INCNT,#0AH
        BR      DATA_COMP3
        SKE     INCNT,#8
        BR      DATA_COMP1
        SKT     DATA1,#0001B
        NOT1    UP_F
DATA_COMP1 :
        SKE     INCNT,#9
        BR      DATA_COMP2
        SKT     DATA1,#0010B
        NOT1    UP_F
DATA_COMP2 :
        RET
DATA_COMP3 :
        SKE     INCNT,#0AH
        BR      DATA_COMP4
        SKT     DATA1,#0100B
        NOT1    UP_F
DATA_COMP4 :
        SKE     INCNT,#0BH
        BR      DATA_COMP5
        SKT     DATA1,#0010B
        NOT1    UP_F
DATA_COMP5 :
        RET
DATA_COMP11 :
        SKLT    INCNT,#0EH
        BR      DATA_COMP6
        SKE     INCNT,#0CH
        BR      DATA_COMP7
        SKT     DATA2,#0001B
        NOT1    UP_F
DATA_COMP7 :
        SKE     INCNT,#0DH
        BR      DATA_COMP8
        SKT     DATA2,#0010B
        NOT1    UP_F
DATA_COMP8 :
        RET
DATA_COMP6 :
        SKE     INCNT,#0EH
        BR      DATA_COMP9
        SKT     DATA2,#0100B
        NOT1    UP_F
DATA_COMP9 :
        SKE     INCNT,#0FH
        BR      DATA_COMP10
        SKT     DATA2,#1000B
        NOT1    UP_F
DATA_COMP10 :
        RET
```

① INCNTの値によりDATA1または，DATA2のある1ビットにセットした値（8ビット前に受信した値）と今，受信したコードの比較を行います。今，受信したコードの状態は，UP_Fフラグにセットされてあり表2－6に示すようになります。また比較結果もUP Fフラグにセットし，表2－7に示すようになります。INCNTと比較を行うビットの関係は表2－8に示すようになります。

表2－6　UP_Fフラグと入力データの関係

UP_F	受信コード
“1”	1
“0”	0

表2－7　UP_Fフラグと比較結果の関係

UP_F	比較結果
“1”	一致
“0”	不一致

表2－8　INCNTと比較を行う位置

INCNT	DATA2				DATA1			
	#3	#2	#1	#0	#3	#2	#1	#0
8								*
9							*	
A						*		
B					*			
C				*				
D			*					
E		*						
F	*							

注）　＊は，比較を行う位置を示します。

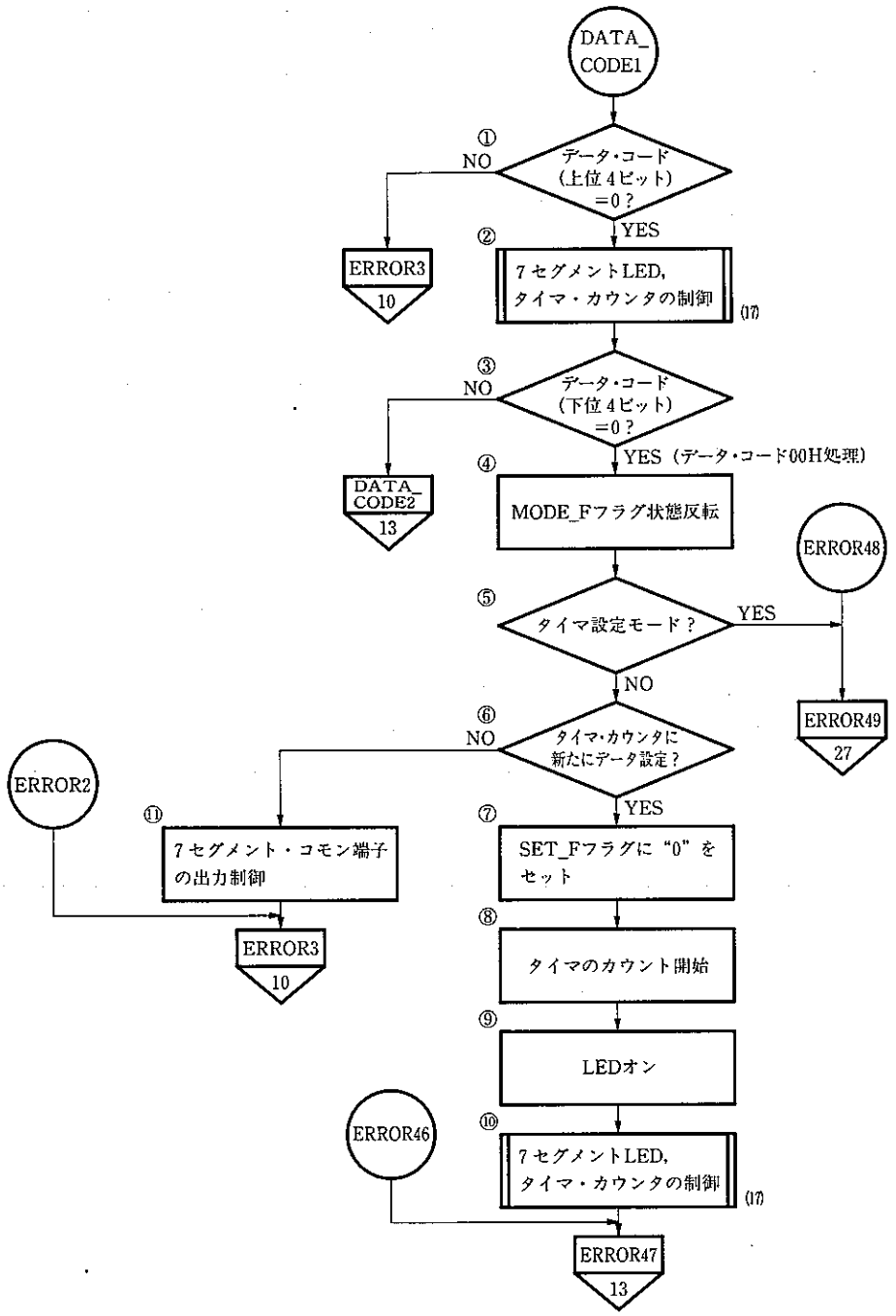
(5) データ・コードによるタイマ・カウンタ，モード制御

受信したデータ・コードにより，表2-9に示すように処理を行います。

表2-9　受信データ・コードによる処理

データ・コード	処　　理
00H	タイマ設定モードとタイマ動作モードを00Hを受信するごとに切り替えます。 タイマ設定モードでカウンタに新たにデータを設定した状態で00Hを受信すると，タイマ動作モードに切り替えてタイマのカウントを開始します。
01H	タイマ設定モードにおいてのみ有効です。 タイマ設定モードにおいて01Hを受信すると，カウンタに新たにデータを設定します。01Hを受信したときの状態により，データの設定の方法が異なります。 (a)　最初に受信した有効データ・コードが01Hのときカウンタに1分を設定し，タイマのカウントを停止します。 (b)　(a)以外のとき，カウンタの分の値をインクリメントします。カウンタの設定値が100分のときに01Hを受信すると，カウンタの設定値を1分にします。
02H	タイマ設定モードにおいてのみ有効です。 タイマ設定モードにおいて02Hを受信すると，カウンタに新たにデータを設定します。02Hを受信したときの状態により，データの設定の方法が異なります。 (a)　最初に受信したデータ・コードが02Hのときカウンタに100分を設定し，タイマのカウントを停止します。 (b)　(a)以外のとき，カウンタの分の値をデクリメントします。カウンタの設定値が1分のときに02Hを受信すると，カウンタの設定値を100分にします。
00H,01H,02H以外のコード	無効データ・コードとして再びリーダ・コード検出処理を行います。

フロー・チャート 12


DATA_
CODE1
①
NO
データ・コード
(上位4ビット)
=0？
YES
ERROR3
10
②
7セグメントLED,
タイマ・カウンタの制御
(17)
③
NO
データ・コード
(下位4ビット)
=0？
YES（データ・コード00H処理）
DATA_
CODE2
13
④
MODE_Fフラグ状態反転
ERROR48
⑤
タイマ設定モード？
YES
NO
ERROR49
27
⑥
NO
タイマ・カウンタに
新たにデータ設定？
YES
ERROR2
⑪
7セグメント・コモン端子
の出力制御
⑦
SET_Fフラグに“0”を
セット
ERROR3
10
⑧
タイマのカウント開始
⑨
LEDオン
ERROR46
⑩
7セグメントLED,
タイマ・カウンタの制御
(17)
ERROR47
13

```
DATA_CODE1：
        SKE     DATA2,#0 }
        BR      ERROR3   }
        CALL    TIME_CTL3
        SKE     DATA1,#0 }
        BR      DATA_CODE2 }
        NOT1    MODE_F
        SKF1    MODE_F   }
ERROR48：                |
        BR      ERROR49  }
        SKT1    SET_F    }
        BR      MODE_TURN }
        CLR1    SET_F
        SET1    COUNT_F
        SET1    P0D3
        CALL    TIME_CTL2
ERROR46：
        BR      ERROR47
MODE_TURN：
        MOV     P0B,#1
ERROR2：
        BR      ERROR3
```

① データ・コード下位4ビットがすべて“0”か判定します。“0”でないときはERROR3にとびます。

② 7セグメントLED，タイマ・カウンタの制御を行います。

③ データ・コード00Hを受信したか判定します。00H以外のデータ・コードを受信したときは，DATA_CODE2にとびます。

④ モードの状態を示すMODE_Fフラグの状態を反転します。

⑤ タイマ設定モード(MODE_F＝1)のときは，ERROR49にとびます。

⑥ タイマ・カウンタに新たにデータを設定したかを判断します。新たにデータを設定していない(SET_F＝0) ときは，MODE_TURNにとびます。

⑦ SET_Fフラグに“0”をセットします。

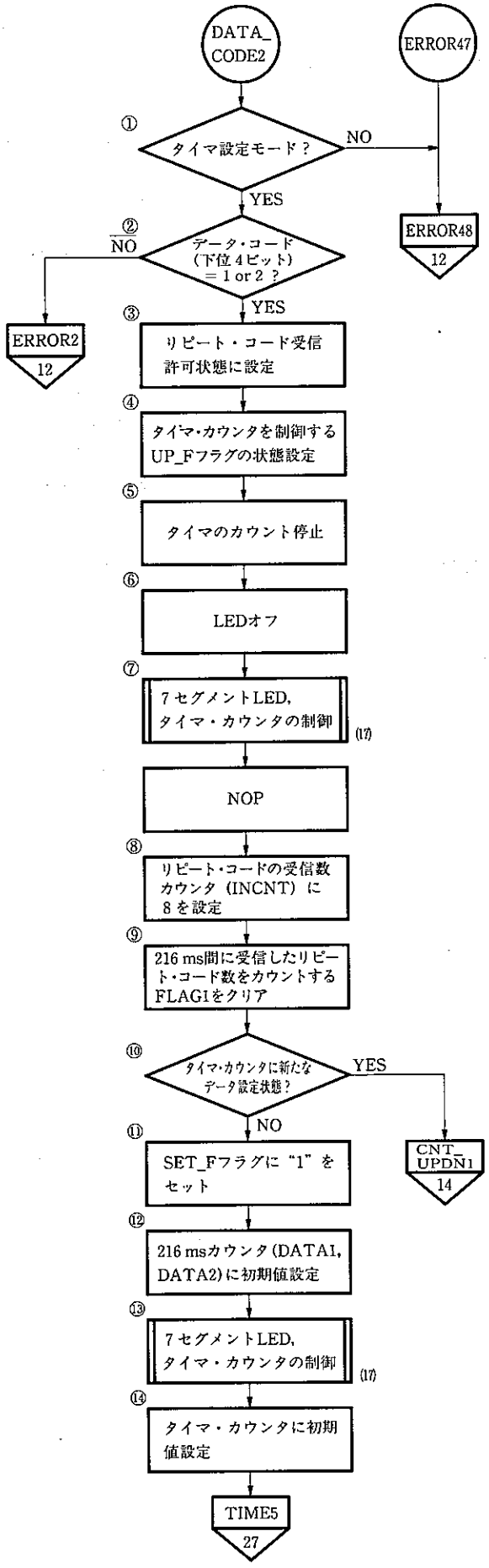
⑧ COUNT_Fフラグに“1”をセットしてタイマのカウントを開始します。

⑨ タイマがカウント状態であることを示すLEDをオンします。

⑩ 7セグメントLED，タイマ・カウンタの制御を行います。

⑪ 7セグメントLEDのコモン端子出力の制御を行います。

フロー・チャート 13


DATA_
CODE2
ERROR47
①
タイマ設定モード？
NO
YES
②
NO
データ・コード
(下位4ビット)
＝1 or 2 ？
ERROR48
12
YES
ERROR2
12
③
リピート・コード受信
許可状態に設定
④
タイマ・カウンタを制御する
UP_Fフラグの状態設定
⑤
タイマのカウント停止
⑥
LEDオフ
⑦
7セグメントLED,
タイマ・カウンタの制御
(17)
NOP
⑧
リピート・コードの受信数
カウンタ（INCNT）に
8を設定
⑨
216 ms間に受信したリピー
ト・コード数をカウントする
FLAG1をクリア
⑩
タイマ・カウンタに新たな
データ設定状態？
YES
NO
CNT_
UPDN1
14
⑪
SET_Fフラグに“1”を
セット
⑫
216 msカウンタ(DATA1,
DATA2)に初期値設定
⑬
7セグメントLED,
タイマ・カウンタの制御
(17)
⑭
タイマ・カウンタに初期
値設定
TIME5
27

```
DATA_CODE2：
        SKT1    MODE_F          ①
ERROR47：
        BR      ERROR48
        SKE     DATA1,#1        ②
        SKNE    DATA1,#2
        BR      DATA_CODE3
        BR      ERROR2
DATA_CODE3：
        SET1    ROK_F           ③
        SKNE    DATA1,#1        ④
        SET1    UP_F
        CLR1    COUNT_F         ⑤
        CLR1    P0D3            ⑥
        CALL    TIME_CTL        ⑦
        NOP
        MOV     INCNT,#8        ⑧
        AND     FLAG1,#0001B    ⑨
        SKF1    SET_F           ⑩
        BR      CNT_UPDN1
        SET1    SET_F           ⑪
        MOV     DATA1,#8        ⑫
        MOV     DATA2,#2
        CALL    TIME_CTL4       ⑬
        MOV     MUSCNT3,#5      ⑭
        MOV     MUSCNT2,#8
        MOV     MUSCNT1,#OFH
        MOV     SCNT2,#0
        MOV     SCNT1,#6
        MOV     MCNT2,#0
        MOV     MCNT1,#0
        SKF1    UP_F
        MOV     MCNT2,#1
        BR      TIME5
```

① タイマ動作モード(MODE_F＝1)のとき，ERROR48にとびます。

② データ・コードの上位４ビットが１か２かを判定します。データ・コードの上位４ビットが１または２でないときは，ERROR2にとびます。

③ リピート・コードの受信（ROK_F＝1）を許可します。

④ タイマ・カウンタの値を制御するUP_Fフラグの状態を設定します。

⑤ COUNT_Fに“0”をセットして，タイマのカウントを停止します。

⑥ タイマがカウント動作状態であることを示すLEDをオフします。

⑦ ７セグメントLED，タイマ・カウンタの制御を行います。

⑧ リピート・コードの受信数をカウントするINCNTに８を設定します。

⑨ 216 ms間に受信したリピート・コード数をカウントするFLAG1の上位３ビットをクリアします。

⑩ タイマ・カウンタに新たにデータを設定したか判断します。新たにデータを設定していない（SET_F＝1）ときは，CNT_UPDN1にとびます。

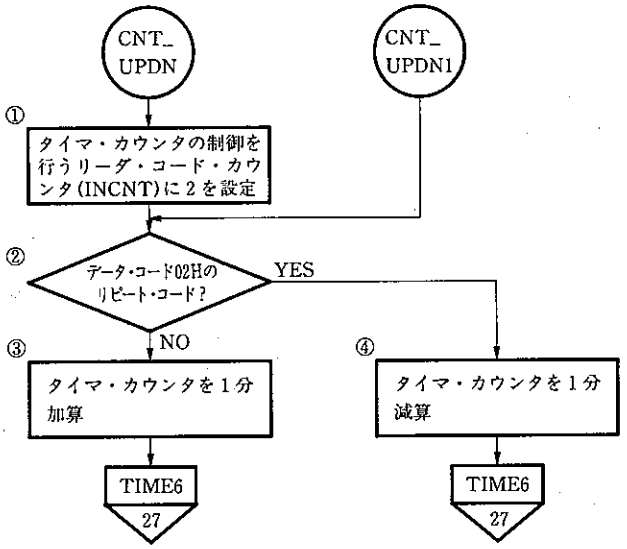
⑪ タイマ・カウンタに新たにデータを設定した状態を示すSET_Fフラグに“1”をセットします。

⑫ 216 msをカウントするDATA1, DATA2に初期値を設定します。

⑬ ７セグメントLED，タイマ・カウンタの制御を行います。

⑭ タイマ・カウンタに初期値を設定します。データ・コード01Hを受信したときは１分を，02Hを受信したときは100分を設定します。

フロー・チャート 14


CNT_
UPDN
CNT_
UPDN1
①
タイマ・カウンタの制御を
行うリーダ・コード・カウ
ンタ(INCNT)に２を設定
②
データ・コード02Hの
リピート・コード?
YES
NO
③
タイマ・カウンタを１分
加算
④
タイマ・カウンタを１分
減算
TIME6
27
TIME6
27

```
CNT_UPDN：
        MOV     INCNT,#0EH ──────────①
CNT_UPDN1：
        SET1    BCD
        SKT1    UP_F    }──────────②
        BR      DOWN1   }
        ADD     MCNT2,#1}──────────③
        ADDC    MCNT1,#0}
        CLR1    BCD     }
        BR      TIME6   }
DOWN1：
        SUB     MCNT2,#1}──────────④
        SUBC    MCNT1,#0}
        CLR1    BCD     }
        BR      TIME6   }
```

① リピート・コードを2つ受信するごとにタイマ・カウンタの制御を行うようにINCNTに0EHを設定します。

② タイマ・カウンタを1分，加算するのか減算するのかの判断をします。02Hのデータ・コードのリピート・コードを受信した（UP_F=0）ときは，DOWN1にとびます。

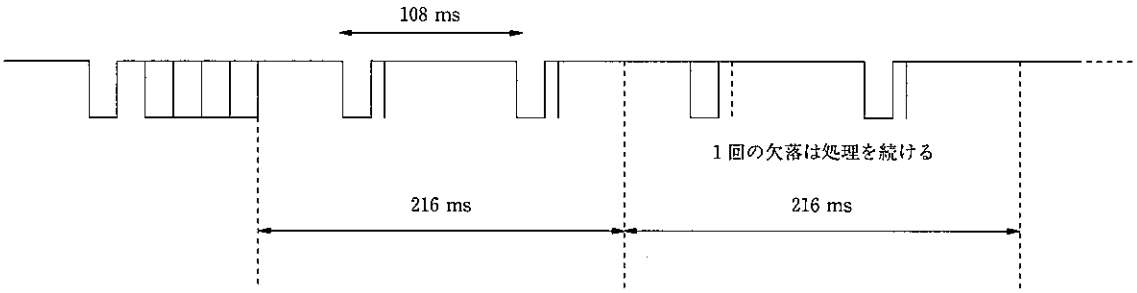
③ タイマ・カウンタを1分，加算します。

④ タイマ・カウンタを1分，減算します。

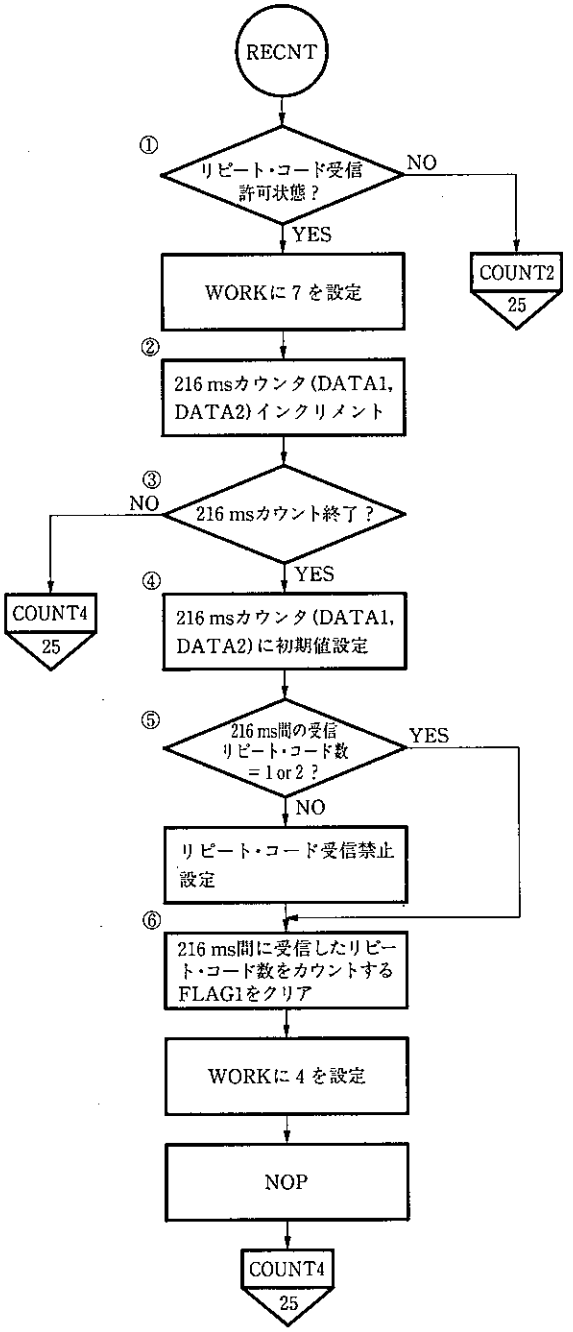
(6)　リピート・コード有効，無効判定

リモコン送信側でキーを押し続けている場合は，108 msごとに2回目以後はリピート・コードを送信しています。本プログラムでは1回目のデータ・コード受信終了後，216 ms間に受信したリピート・コードをカウントすることにより，リピート・コードの有効，無効の判定を行います。216 ms間に受信したリピート・コードの数が1つまたは2つのときは，そのリピート・コードを有効とし，それ以外のときは，リピート・コードを無効としてリピート・コードの受信処理を終了します。

図2－19



フロー・チャート 15


RECNT
①
リピート・コード受信
許可状態？
NO
YES
COUNT2
25
WORKに 7 を設定
②
216 msカウンタ (DATA1,
DATA2) インクリメント
③
NO
216 msカウント終了？
YES
COUNT4
25
④
216 msカウンタ (DATA1,
DATA2) に初期値設定
⑤
216 ms間の受信
リピート・コード数
＝1 or 2 ？
YES
NO
リピート・コード受信禁止
設定
⑥
216 ms間に受信したリピー
ト・コード数をカウントする
FLAG1をクリア
WORKに 4 を設定
NOP
COUNT4
25

```
RECNT :
        SKT1    ROK_F      }———————①
        BR      COUNT2     }
        MOV     WORK, #7
        ADD     DATA1, #1  }———————②
        ADDC    DATA2, #0  }
        SKT1    CY         }———————③
        BR      COUNT4     }
        MOV     DATA1, #8  }———————④
        MOV     DATA2, #2  }
        SKGE    FLAG1, #6  }———————⑤
        SKGE    FLAG1, #2  }
        CLR1    ROK_F      }
        AND     FLAG1, #0001B——————⑥
        MOV     WORK, #4
        NOP
        BR      COUNT4
```

① リピート・コードを受信許可状態かどうか判断します。受信許可状態でない（ROK_F=0）ときは，実行ステップ数を調整するためCOUNT2にとび，28ステップ実行してCALL先へ帰ります。また，リピート・コード受信許可状態であればリピート・コードを数える処理に移行します。

② 216 msをカウントするDATA1, DATA2をインクリメントします。

③ 216 msをカウントしたか判断します。216 msをカウントしていない(CY=0)ときは，P0B端子の状態をチェックするため，COUNT4にとび実行ステップを調整した後CALL先へ戻ります。

④ 216 msをカウントするDATA1, DATA2に初期値を設定します。

⑤ リピート・コードの有効，無効を判定します。216 ms間に受信したリピート・コードの数が，1，2以外のときは，無効としてリピート・コードの受信を禁止状態（ROK_F=0）にします。

⑥ リピート・コードの受信数をカウントするFLAG1の上位3ビットを0にします。

(7)　リピート・コードによるタイマ・カウンタ制御

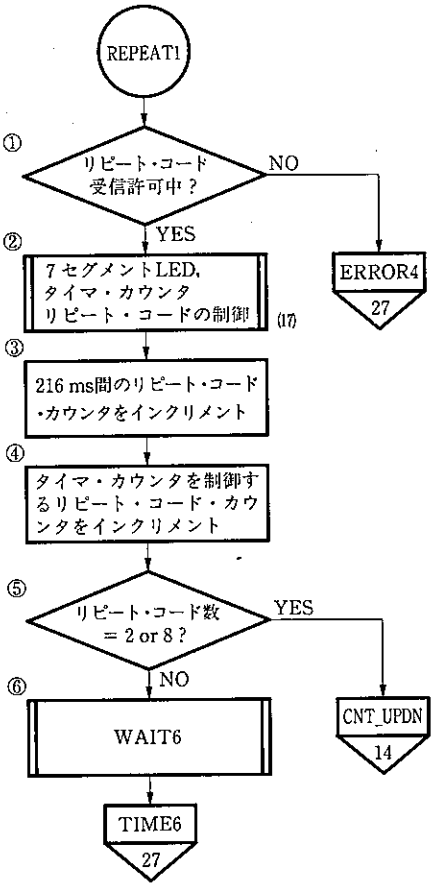
リーダ・コードを検出後，リード・コードの立ち上がりから次のコードの立ち下がりまでの時間が，2.0 ms以上2.6 ms未満のときは，そのコードをリピート・コードと判定します。

タイマ・カウンタの制御は，有効なリピート・コードを連続して2つまたは8つ受信したときに行います。有効なリピート・コードとは，以下に示す条件を満たしたものです。

(1)　01Hまたは02Hのデータ・コードのリピート・コードである。

(2)　216 ms間に受信したリピート・コード数が1つまたは2つである。

フロー・チャート 16



```
REPEAT1：
        SKT1    ROK_F  }
        BR      ERROR4 }
        CALL    TIME_CTL2
        ADD     FLAG1,#2
        ADD     INCNT,#1
        SKF1    CY     }
        BR      CNT_UPDN}
        CALL    WAIT6
        BR      TIME6
```

① リピート・コード受信許可状態かを判定します。リピート・コード受信許可状態でない（ROK_F=0）ときは，受信したリピート・コードは無効としてERROR4にとびます。

② 7セグメントLED，タイマ・カウンタ，リピート・コードの制御を行います。

③ 216 ms間に受信したリピート・コード数をカウントするFLAG1のカウンタをインクリメントします。

④ リピート・コードの受信数をカウントするINCNTをインクリメントします。

⑤ タイマ・カウンタの値の制御を行うか判定します。CY=1のときは，リピート・コードを連続して2つまたは8つ受信したとしてCNT_UPDNにとびます。

(8)　7セグメントLED，タイマ・カウンタ制御

(a)　7セグメントLEDの制御

7セグメントLEDの制御は，TIME_CTLルーチン内のSEGMENT_JOBルーチンで行います。μPD17103にはタイマを内蔵していないために，本プログラムにおいては命令を500ステップ（1 ms）実行するごとに，SEGMENT_JOBルーチンに処理を移し7セグメントLEDの制御を行います。なお，SEGMENT_JOBルーチンに処理を移す方法はリーダ・コード検出処理中とそれ以外のときは異なり以下のようになります。

(i)　リーダ・コード検出処理中は，命令を500ステップ（1 ms）実行するごとにSEGMENT JOBルーチンに処理を移し，7セグメントLEDの制御を行います。

(ii)　リーダ・コード検出処理以外のときは，命令を50ステップ(100 μs)実行するごとにTIME CTLルーチンに処理を移し，そのときCNT100が0AHの場合はSEGMENT_JOBルーチンに処理を移して7セグメントLEDの制御を行います。

(b)　タイマ・カウンタ制御

μPD17103はタイマを内蔵していないため，実行ステップ数により時間の制御を行います。本プログラムにおいては，リーダ・コード検出処理以外のときは，100 μsごとにプリアンプ(P0B$_0$端子）の状態を確認しますので，タイマのカウンタに使用するメモリ（RAM）は，100 μs以上をカウントできるように以下に示すように設定します。

表2－10　タイマのカウンタに使用するRAM

メモリ名	番地（HEX）	機　　能
MCNT1	02	カウンタの10分の桁の制御
MCNT2	03	カウンタの1分の桁の制御
SCNT1	04	カウンタの10秒の桁の制御
SCNT2	05	カウンタの1秒の桁の制御
MUSCNT1	06	この3つのメモリによりカウンタのms単位の3桁の制御
MUSCNT2	07	
MUSCNT3	08	
CNT100	01	カウンタの100 μs単位の制御

注）MUSCNT3の下位2ビットは，タイマのカウンタには使用していません。

タイマのカウンタに設定できる値は1分から100分までです。たとえば，59分を設定したときのカウンタの値は，以下のようになります。

MCNT1　　=5
MCNT2　　=9
SCNT1　　=6
SCNT2　　=0
MUSCNT1　=0FH
MUSCNT2　=8
MUSCNT3　=01XXB（Xは影響を与えないことを示します。）

また，カウント終了時のカウンタの値は，以下のようになります。

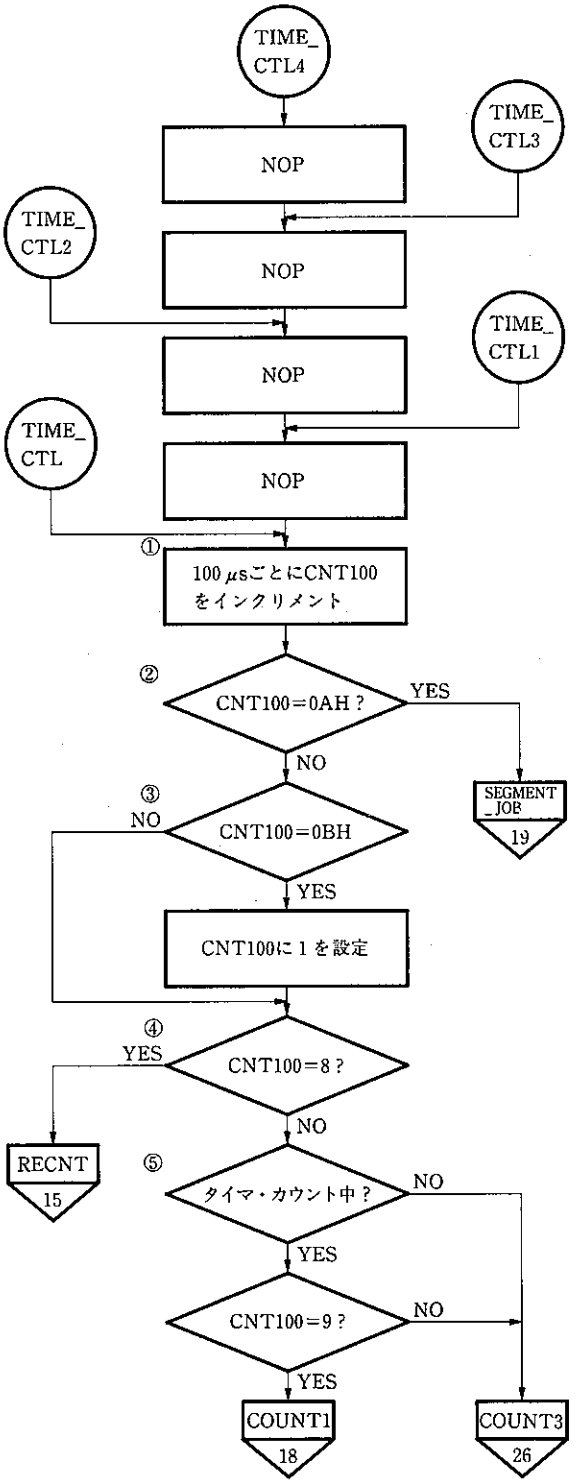
MCNT1　　=0
MCNT2　　=1
SCNT1　　=0
SCNT2　　=0
MUSCNT1　=0FH
MUSCNT2　=0FH
MUSCNT3　=11XXB（Xは影響を与えないことを示します。）

タイマの制御は，TIME_CTLルーチン内のCOUNT1ルーチンで行います。命令を500ステップ（1 ms）実行するごとに，COUNT1ルーチンに処理を移してタイマの制御を行います。なお，COUNT1ルーチンに処理を移す方法はリーダ・コード検出処理中とそれ以外では異なり以下に示すようになります。

(i)　リーダ・コード検出処理中は，命令を500ステップ（1 ms）実行するごとにCOUNT1ルーチンに処理を移し，タイマ・カウンタの制御を行います。

(ii)　リーダ・コード検出処理以外のときは，命令を50ステップ（100 μs）実行するごとにTIME CTLルーチンに処理を移し，そのときCNT100が9の場合はCOUNT1ルーチンに処理を移し，タイマ・カウンタの制御を行います。

フロー・チャート 17


TIME_
CTL4
TIME_
CTL3
NOP
TIME_
CTL2
NOP
TIME_
CTL1
NOP
TIME_
CTL
NOP
①
100 μsごとにCNT100
をインクリメント
②
CNT100＝0AH？
YES
NO
SEGMENT
_JOB
19
③
NO
CNT100＝0BH
YES
CNT100に１を設定
④
YES
CNT100＝8？
NO
RECNT
15
⑤
タイマ・カウント中？
NO
YES
CNT100＝9？
NO
YES
COUNT1
18
COUNT3
26

```
TIME_CTL4：
        NOP
TIME_CTL3：
        NOP
TIME_CTL2：
        NOP
TIME_CTL1：
        NOP
TIME_CTL：
        ADD     CNT100,#1 ―――――――――――――――①
        SKNE    CNT100,#0AH }――――――――――――②
        BR      SEGMENT_JOB }
        SKNE    CNT100,#0BH }――――――――――――③
        MOV     CNT100,#1   }
        SKNE    CNT100,#8 }――――――――――――――④
        BR      RECNT     }
        SKF1    COUNT_F   }――――――――――――――⑤
        SKE     CNT100,#9 }
        BR      COUNT3    }
```

① 100 μsごとにCNT100をインクリメントします。

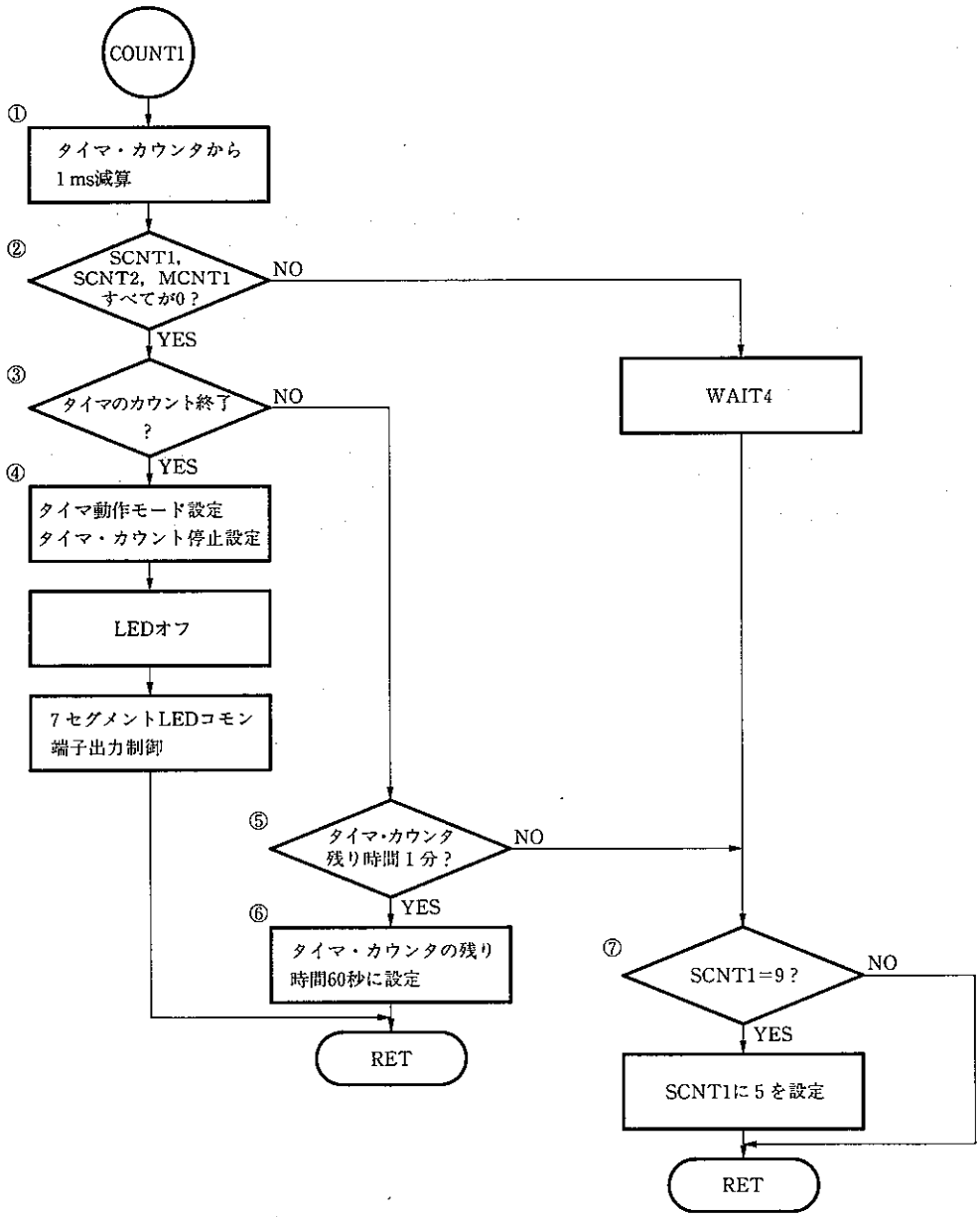
② CNT100＝0AHのとき，7セグメントLEDの制御を行うためにSEGMENT_JOBにとびます。

③ CNT100＝0BHのとき，CNT100を1にします。

④ CNT100＝8のとき，リピート・コードの有効，無効の判定を行うためにRECNTにとびます。

⑤ タイマがカウント停止状態（COUNT_F＝0），またはCNT100≠9のとき，実行ステップ数を調整するためにCOUNT3にとびます。

フロー・チャート 18


COUNT1
①
タイマ・カウンタから
1 ms減算
②
SCNT1,
SCNT2, MCNT1
すべてが0？
NO
YES
WAIT4
③
タイマのカウント終了
？
NO
YES
④
タイマ動作モード設定
タイマ・カウント停止設定
LEDオフ
7セグメントLEDコモン
端子出力制御
⑤
タイマ・カウンタ
残り時間1分？
NO
YES
⑥
タイマ・カウンタの残り
時間60秒に設定
RET
⑦
SCNT1＝9？
NO
YES
SCNT1に5を設定
RET

```
COUNT1：
        SUB     MUSCNT3,#4      }
        SUBC    MUSCNT2,#0      |
        SUBC    MUSCNT1,#0      |
        SKF1    CY              |
        AND     MUSCNT3,#0111B  |
        SKF1    CY              |
        MOV     MUSCNT2,#8      |
        SET1    BCD             |
        SUBC    SCNT2,#0        |
        SUBC    SCNT1,#0        |
        SUBC    MCNT2,#0        |
        SUBC    MCNT1,#0        |
        CLR1    BCD             }
        NOP
        SET2    CMP,Z           }
        SUB     SCNT2,#0        |
        SUB     SCNT1,#0        |
        SUB     MCNT1,#0        |
        SKT1    Z               }
        BR      CNTX
        SKE     MCNT2,#1        }
        BR      CNTXX           }
        CLR2    MODE_F,COUNT_F  }
        CLR1    POD3            |
        MOV     POB,#1          }
        BR      RET_JOB
CNTXX：
        SKE     MCNT2,#2        }
        BR      CNTXXX          }
        MOV     SCNT1,#6        }
        MOV     MCNT2,#1        }
RET_JOB：
        RET
CNTX：
        NOP
        NOP
        NOP
        NOP
CNTXXX：
        SKNE    SCNT1,#9        }
        MOV     SCNT1,#5        }
        RET
```

① タイマ・カウンタに設定された値から1 msを減算します。

② SCNT2＝0，SCNT1＝0，MCNT1＝0かの判断を行います。

③ タイマのカウントの終了を判断します。
MCNT2＝1以外のときは，カウントは終了していないとしてCNTXXにとびます。

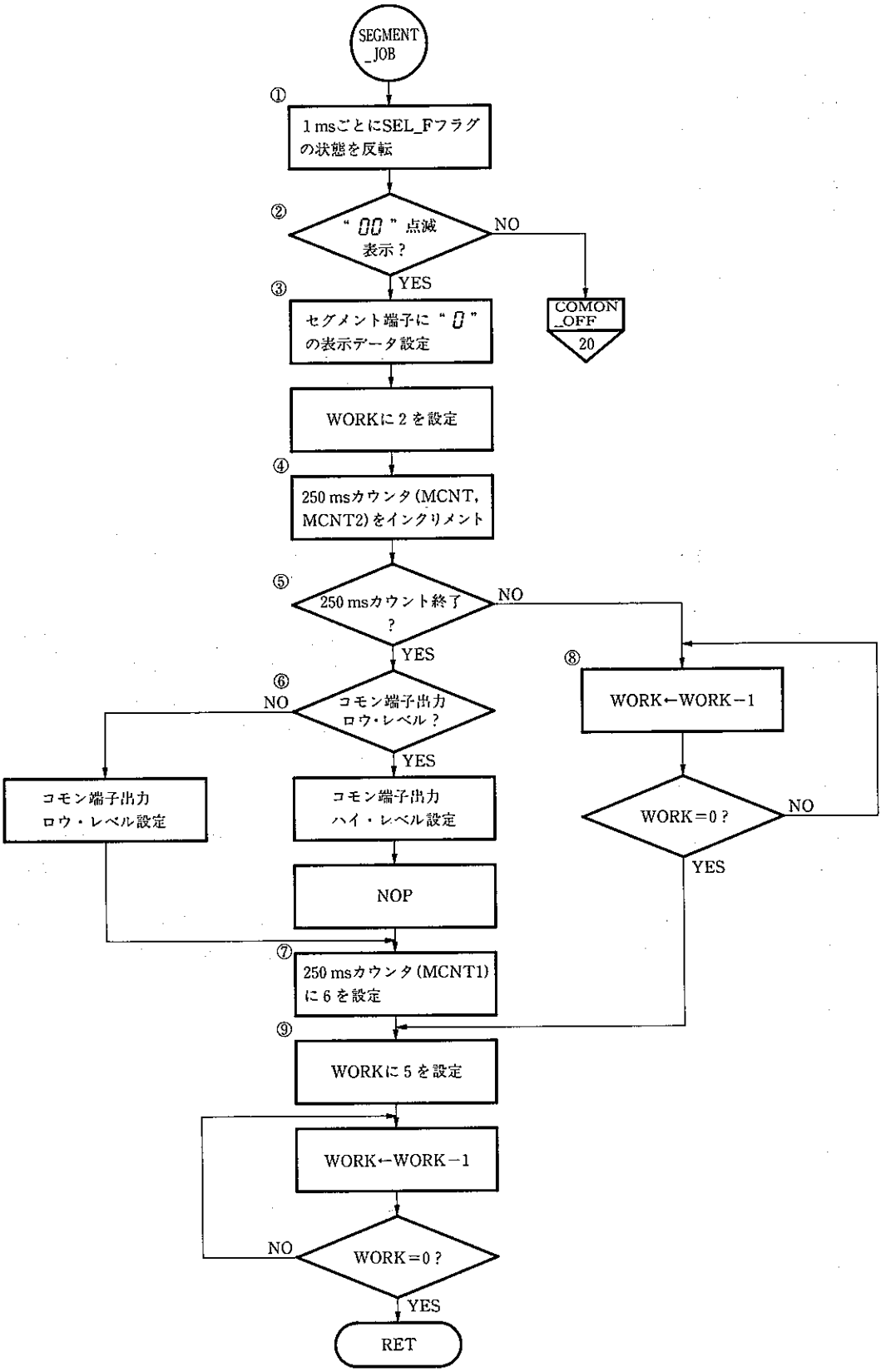
④ タイマのカウント終了したとしてCOUNT_Fに“0”を（カウント停止状態を示す），MODE_Fに“0”を（タイマ動作モードを示す），$P0D_3$端子の出力をロウ・レベル（LEDオフ）に設定します。また，7セグメントLEDのコモン端子の出力をロウ・レベルに設定します。

⑤ タイマ・カウンタの残り時間が1分であるかを判断します。MCNT2＝2以外のときは，残り時間が1分でないとしてCNTXXXにとびます。

⑥ 7セグメントLEDに“60”を表示するためにSCNT1に6をMCNT2に1を設定します。

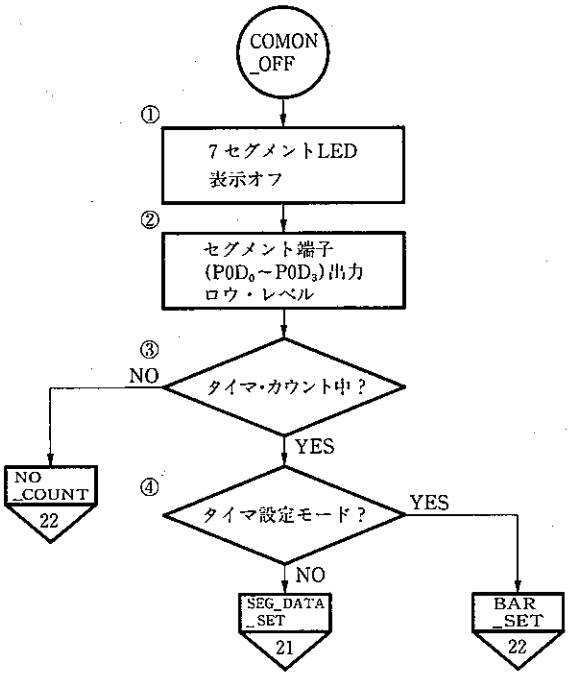
⑦ SCNT1＝9のときはSCNT1を5に設定します。

フロー・チャート 19


SEGMENT_JOB
①
1 msごとにSEL_Fフラグの状態を反転
②
“ 00 ” 点滅表示？
NO
YES
③
セグメント端子に “ 0 ” の表示データ設定
COMON_OFF
20
WORKに 2 を設定
④
250 msカウンタ (MCNT, MCNT2) をインクリメント
⑤
250 msカウント終了？
NO
YES
⑥
コモン端子出力ロウ・レベル？
NO
YES
コモン端子出力ロウ・レベル設定
コモン端子出力ハイ・レベル設定
NOP
⑦
250 msカウンタ (MCNT1) に 6 を設定
⑧
WORK←WORK−1
WORK＝0？
NO
YES
⑨
WORKに 5 を設定
WORK←WORK−1
NO
WORK＝0？
YES
RET

```
SEGMENT_JOB：
        NOT1    SEL_F                    ①  1 ms (500ステップ) ごとにSEL_Fフラグの状態を反転します。
        SKT1    COUNT_F   }              ②  7セグメントLEDに“ 00 ”を点滅表示するかを判断します。“ 00 ”を点滅表示しない (COUNT_F＝1または, MODE_F＝0)* のときは, COMON_OFFへとびます。
        SKT1    MODE_F    |
        BR      COMON_OFF }
                                            ＊COUNT_F＝1：タイマ，カウント中
                                              MODE_F＝0：タイマ設定モード
        MOV     P0C, #0DH }              ③  7セグメントLEDに“ 00 ”を表示するようにセグメント端子にデータを設定します。
        OR      P0D, #0111B }
        MOV     WORK, #2
        ADD     MCNT1, #1 }              ④  点滅は，250 ms間隔で行います。250 msをカウントするMCNT1，MCNT2をインクリメントします。
        ADDC    MCNT2, #0 }
        SKT1    CY        }              ⑤  250 msをカウントしたか判断します。250 msをカウントしていない（CY＝0）ときは，実行ステップ数を調整するために，FLASH_LOOP1にとびます。
        BR      FLASH_LOOP1 }
        SKT1    P0B2      }              ⑥  コモン端子の出力を反転します。
        BR      ALL_ON    |
        MOV     P0B, #1   |
        BR      FLASH_JOB2 }
ALL_ON：
        MOV     P0B, #7
        NOP
FLASH_JOB2：
        MOV     MCNT1, #6                ⑦  250 msをカウントするMCNT1に初期値を設定します。
        BR      FLASH_JOB3
FLASH_LOOP1：
        SUB     WORK, #1  }              ⑧  実行ステップ数を調整します。
        SKT1    Z         |
        BR      FLASH_LOOP1 }
FLASH_JOB3：
        MOV     WORK, #5  }              ⑨  実行ステップ数を調整します。
FLASH_LOOP2：
        SUB     WORK#1    |
        SKT1    Z         |
        BR      FLASH_LOOP2 }
        RET
```

フロー・チャート 20


COMON
_OFF
①
7セグメントLED
表示オフ
②
セグメント端子
(P0D0～P0D3)出力
ロウ・レベル
③
NO
タイマ・カウント中？
YES
NO
_COUNT
22
④
タイマ設定モード？
YES
NO
SEG_DATA
_SET
21
BAR
_SET
22

```
COMON_OFF :
        MOV     P0B,#7 ──────────①
        AND     P0D,#1000B ──────②
        SKT1    COUNT_F  }
        BR      NO_COUNT }──────③
        SKT1    MODE_F   }
        BR      BAR_SET  }──────④
```

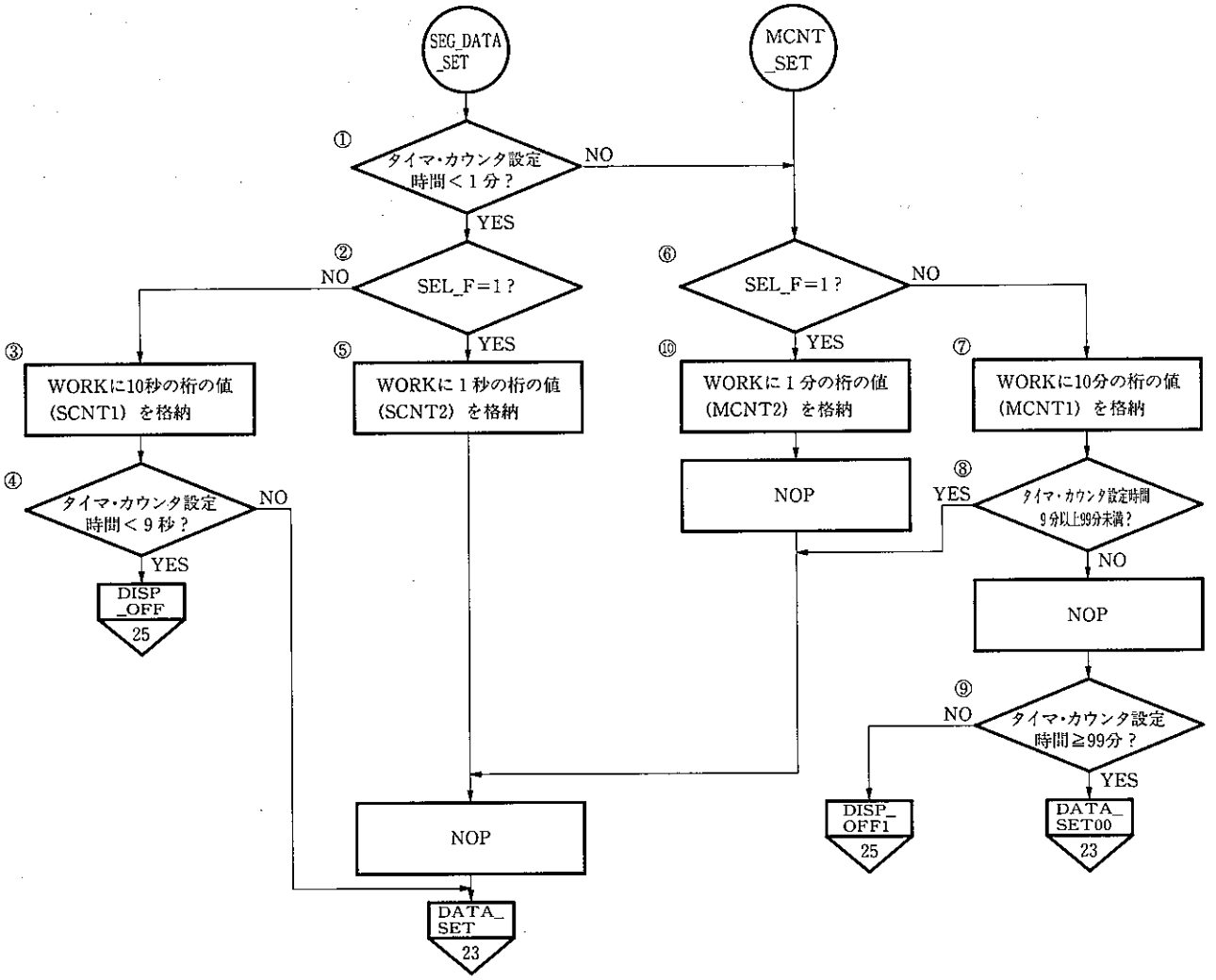
① コモン端子（$P0B_1$，$P0B_2$端子）の出力をハイ・レベルにして，7セグメントLEDの表示をオフします。

② セグメント端子（$P0D_0$～$P0D_3$端子）の出力をロウ・レベルにします。

③ タイマがカウント中かを判定します。カウント中でない（COUNT_F=0）ときは，NO_COUNTにとびます。

④ タイマ設定モードかを判定します。タイマ設定モードのとき（MODE_F=1）は，BAR_SETにとびます。

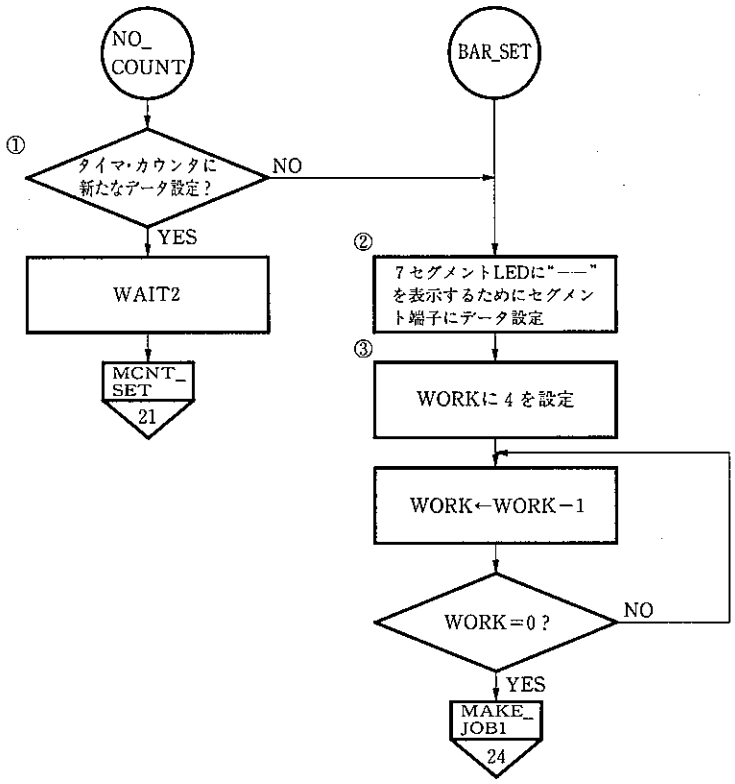
フロー・チャート 21


SEG_DATA_SET
MCNT_SET
①
タイマ・カウンタ設定時間＜１分？
NO
YES
②
NO
SEL_F=1？
YES
⑥
SEL_F=1？
NO
YES
③
WORKに10秒の桁の値(SCNT1) を格納
⑤
WORKに１秒の桁の値(SCNT2) を格納
⑩
WORKに１分の桁の値(MCNT2) を格納
⑦
WORKに10分の桁の値(MCNT1) を格納
④
タイマ・カウンタ設定時間＜９秒？
NO
YES
DISP_OFF 25
NOP
⑧
YES
タイマ・カウンタ設定時間 9分以上99分未満？
NO
NOP
⑨
NO
タイマ・カウンタ設定時間≧99分？
YES
DISP_OFF1 25
DATA_SET00 23
NOP
DATA_SET 23

```
SEG_DATA_SET：
        SKNE    MCNT1,#0  }
        SKE     MCNT2,#1  |
        BR      MCNT_SET  }          ①  タイマ・カウンタの設定時間が1分未満か判定します。1分以上のときは，MCNT_SETにとびます。
        SKF1    SEL_F     }
        BR      SCNT2_SET }          ②  1 msごとに状態が反転するSEL_Fフラグの状態を判定します。SEL_Fフラグが“1”のときは，SCNT2_SETにとびます。
        LD      WORK,SCNT1           ③  WORKに10秒の桁の値（SCNT1）を格納します。
        SKE     WORK,#0   }          ④  タイマ・カウンタの設定時間が9秒未満か判定します。9秒未満のときは，DISP_OFFにとびます。
        BR      DATA_SET  |
        BR      DISP_OFF  }
SCNT2_SET：
        LD      WORK,SCNT2           ⑤  WORKに1秒の桁の値（SCNT2）を格納します。
        BR      DATA_SET11
MCNT_SET：
        SKF1    SEL_F     }
        BR      MCNT2_SET }          ⑥  1 msごとに状態が反転するSEL_Fフラグの状態を判定します。SEL_Fフラグが“1”のときは，MCNT2_SETにとびます。
        LD      WORK,MCNT1           ⑦  WORKに10分の桁の値（MCNT1）を格納します。
        SKE     WORK,#0   }          ⑧  タイマ・カウンタの設定時間が9分以上99分未満か判定します。9分以上99分未満のときは，DATA_SETにとびます。
        BR      DATA_SET  }

        NOP
        SKE     MCNT2,#0  }          ⑨  タイマ・カウンタの設定時間が99分以上か判定します。99分以上のときはDATA_SET00にとびます。
        BR      DISP_OFF1 |
        BR      DATA_SET00}
MCNT2_SET：
        LD      WORK,MCNT2           ⑩  WORKに1分の桁の値（MCNT2）を格納します。
        NOP
DATA_SET11：
        NOP
```

注）　このルーチンでは，7セグメントLEDに表示する値をWORKに格納します。

フロー・チャート 22


NO_
COUNT
BAR_SET
①
タイマ・カウンタに
新たなデータ設定？
NO
YES
WAIT2
MCNT_
SET
21
②
7セグメントLEDに"——"
を表示するためにセグメン
ト端子にデータ設定
③
WORKに4を設定
WORK←WORK−1
WORK＝0？
NO
YES
MAKE_
JOB1
24

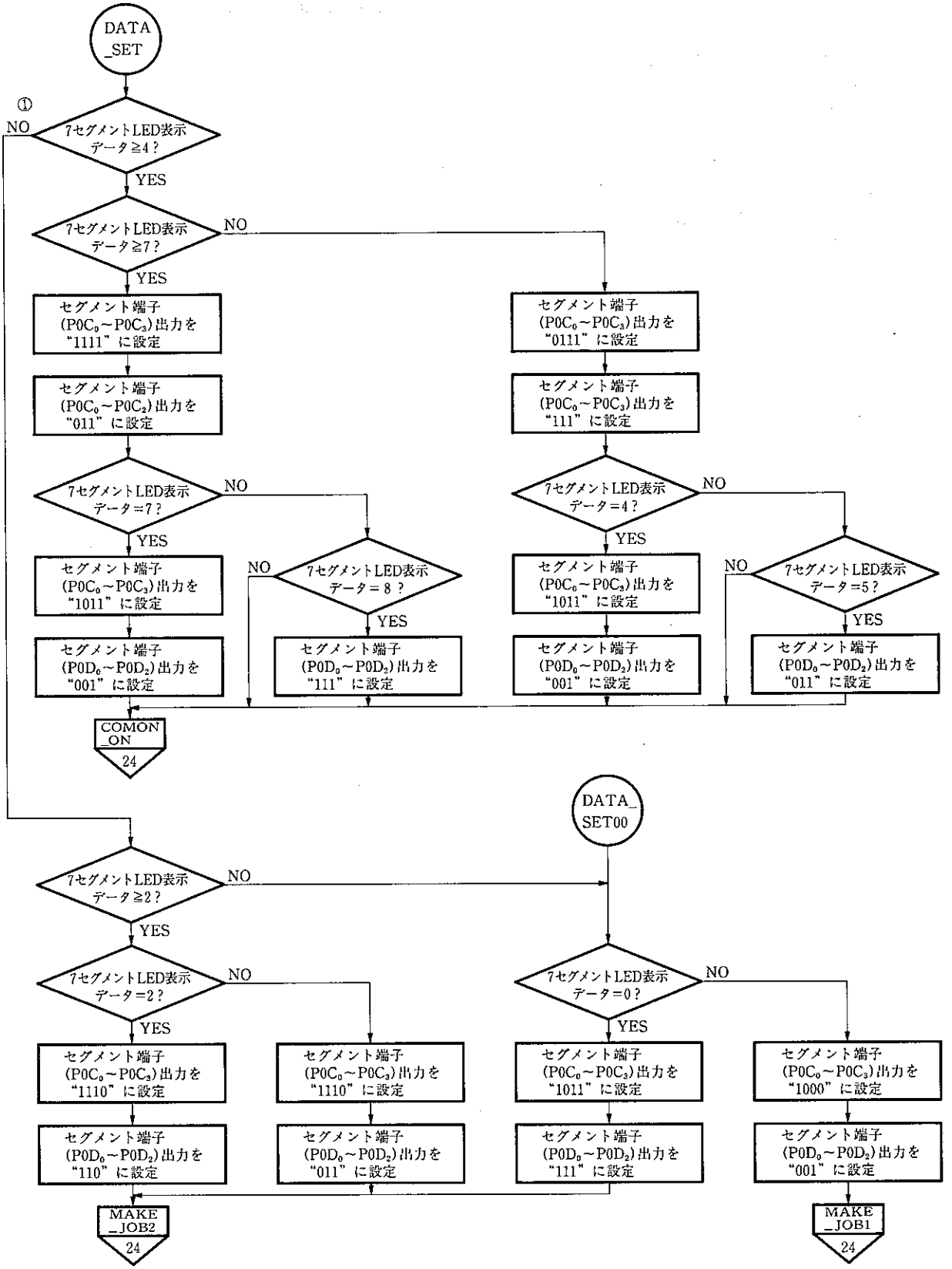
```
NO_COUNT :
        SKT1    SET_F
        BR      BARSET
        NOP
        NOP
        BR      MCNT_SET
BAR_SET :
        MOV     P0C,#2
        OR      P0D,#0
        MOV     WORK,#4
BAR_LOOP :
        SUB     WROK,#1
        SKT1    Z
        BR      BAR_LOOP
        BR      MAKE_JOB1
```

① タイマ・カウンタに新たにデータを設定したかを判断します。新たにデータを設定しない(SET_F=0)ときはBAR_SETにとびます。

② 7セグメントLEDに“—”を表示するためにセグメント端子にデータを設定します。

③ 実行ステップ数を調整します。

フロー・チャート 23


DATA_SET
①
NO
7セグメントLED表示データ≧4？
YES
7セグメントLED表示データ≧7？
NO
YES
セグメント端子(P0C0～P0C3)出力を"1111"に設定
セグメント端子(P0C0～P0C2)出力を"011"に設定
7セグメントLED表示データ=7？
NO
YES
セグメント端子(P0C0～P0C3)出力を"1011"に設定
セグメント端子(P0D0～P0D2)出力を"001"に設定
NO
7セグメントLED表示データ＝8？
YES
セグメント端子(P0D0～P0D2)出力を"111"に設定
セグメント端子(P0C0～P0C3)出力を"0111"に設定
セグメント端子(P0C0～P0C3)出力を"111"に設定
7セグメントLED表示データ=4？
NO
YES
セグメント端子(P0C0～P0C3)出力を"1011"に設定
セグメント端子(P0D0～P0D2)出力を"001"に設定
NO
7セグメントLED表示データ=5？
YES
セグメント端子(P0D0～P0D2)出力を"011"に設定
COMON_ON
24
DATA_SET00
7セグメントLED表示データ≧2？
NO
YES
7セグメントLED表示データ=2？
NO
YES
セグメント端子(P0C0～P0C3)出力を"1110"に設定
セグメント端子(P0D0～P0D2)出力を"110"に設定
セグメント端子(P0C0～P0C3)出力を"1110"に設定
セグメント端子(P0D0～P0D2)出力を"011"に設定
7セグメントLED表示データ=0？
NO
YES
セグメント端子(P0C0～P0C3)出力を"1011"に設定
セグメント端子(P0D0～P0D2)出力を"111"に設定
セグメント端子(P0C0～P0C3)出力を"1000"に設定
セグメント端子(P0D0～P0D2)出力を"001"に設定
MAKE_JOB2
24
MAKE_JOB1
24

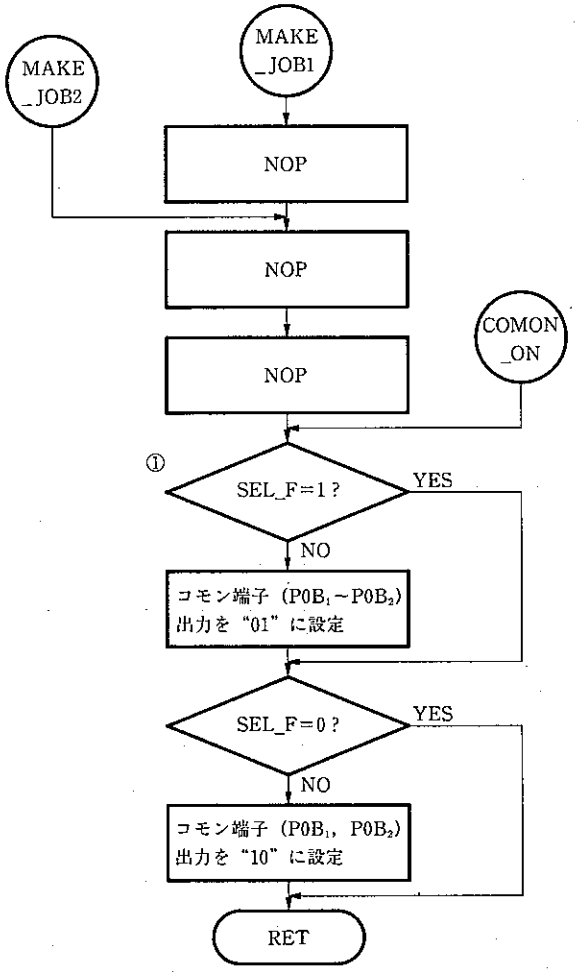
```
DATA_SET：
        SKGE    WORK,#4
        BR      DATA_SET0
        SKGE    WORK,#7
        BR      DATA_SET4
        MOV     P0C,#0FH
        OR      P0D,#0110B
        SKE     WORK,#7
        BR      DATA_SET8
        MOV     P0C,#5
        AND     P0D,#1100B
        BR      COMON_ON
DATA_SET8：
        SKNE    WORK,#8H
        OR      P0D,#0110B
        BR      COMON_ON
DATA_SET4：
        MOV     P0C,#0EH
        OR      P0D,#0111B
        SKE     WORK,#4
        BR      DATA_SET5
        MOV     P0C,#0BH
        AND     P0D,#1100B
        BR      COMON_ON
DATA_SET5：
        SKNE    WORK,#5
        AND     P0D,#1110B
        BR      COMON_ON
DATA_SET0：
        SKGE    WORK,#2
        BR      DATA_SET00
        SKE     WORK,#2
        BR      DATA_SET3
        MOV     P0C,#7
        OR      P0D,#0011B
        BR      MAKE_JOB2
DATA_SET3
        MOV     P0C,#7
        OR      P0D,#0110B
        BR      MAKE_JOB2
DATA_SET00：
        SKE     WORK,#0
        BR      DATA_SET1
        MOV     P0C,#0DH
        OR      P0D,#0111B
        BR      MAKE_JOB2
DATA_SET1：
        MOV     P0C,#1
        OR      P0D,#0100B
```

① WORKの値により表2-11に示すようにセグメント端子（$P0C_0$～$P0C_3$，$P0D_0$～$P0D_2$）にデータを設定します。

表2－11　WORKの値によるセグメント端子出力設定

WORKの値	$P0C_0$	$P0C_1$	$P0C_2$	$P0C_3$	$P0D_0$	$P0D_1$	$P0D_2$
0	1	1	0	1	1	1	1
1	0	0	0	1	1	0	0
2	0	1	1	1	0	1	1
3	0	1	1	1	1	1	0
4	1	0	1	1	1	0	0
5	1	1	1	0	1	1	0
6	1	1	1	0	1	1	1
7	0	1	0	1	1	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	1	1	0

フロー・チャート 24


MAKE
_JOB1
MAKE
_JOB2
NOP
NOP
NOP
COMON
_ON
①
SEL_F=1？
YES
NO
コモン端子（P0B1～P0B2）
出力を“01”に設定
SEL_F=0？
YES
NO
コモン端子（P0B1，P0B2）
出力を“10”に設定
RET

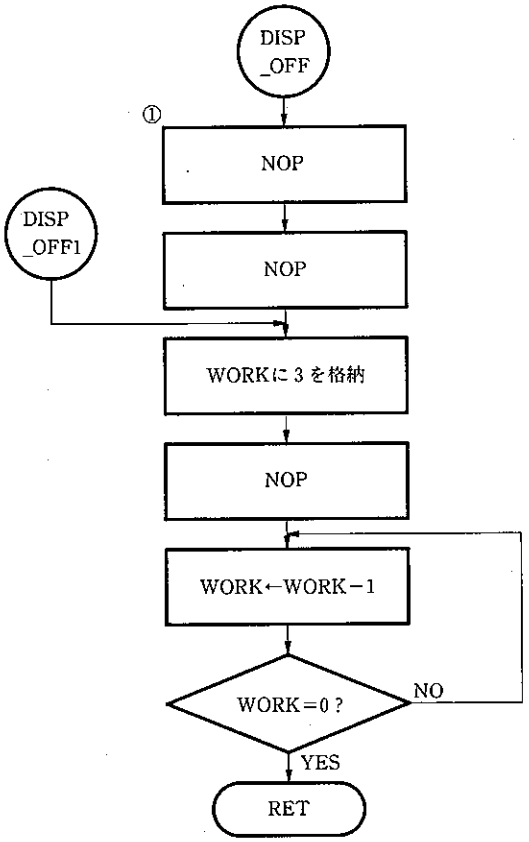
```
MAKE_JOB1 :
        NOP
MAKE_JOB2 :
        NOP
        NOP
COMON_ON :
        SKT1    SEL_F
        MOV     P0B, #5
        SKF1    SEL_F
        MOV     P0B, #3
        RET
```

① SEL_Fフラグの状態でコモン端子($P0B_1$, $P0B_2$端子)の出力を表2-12に示すように設定します。

表2-12　SEL_Fフラグとコモン端子出力

SEL_F	$P0B_1$端子	$P0B_2$端子
"1"	ハイ・レベル	ロウ・レベル
"0"	ロウ・レベル	ハイ・レベル

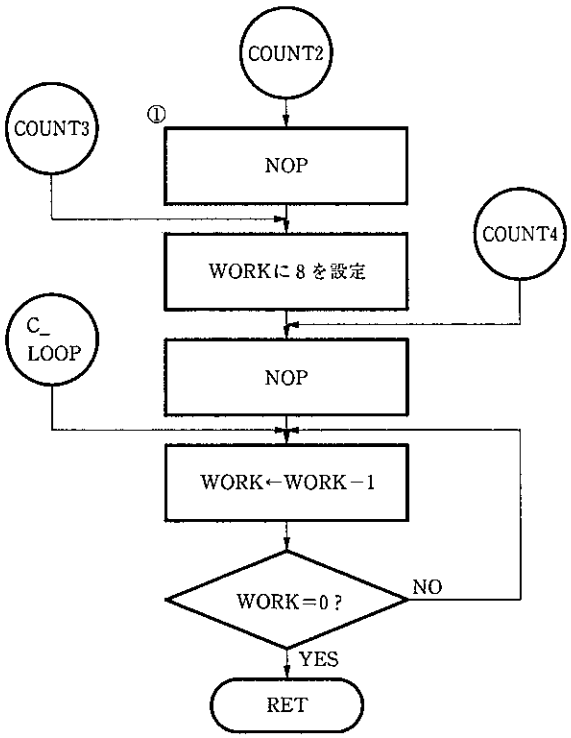
フロー・チャート 25


DISP
_OFF
①
NOP
DISP
_OFF1
NOP
WORKに 3 を格納
NOP
WORK←WORK−1
WORK＝0 ?
NO
YES
RET

```
DISP_OFF :
        NOP                 }
        NOP                 |
DISP_OFF1 :                 |
        MOV     WORK, #3    |
        NOP                 |
DISP_LOOP :                 |
        SUB     WORK, #1    |
        SKF1    Z           |
        RET                 |
        BR      DISP_LOOP   }
        ;
```

① 実行ステップ数を調整します。

フロー・チャート 26


COUNT2
COUNT3
①
NOP
COUNT4
WORKに 8 を設定
C_
LOOP
NOP
WORK←WORK－1
WORK＝0 ?
NO
YES
RET

```
COUNT2 :
        NOP
COUNT3 :
        MOV     WORK, #8
COUNT4 :
        NOP
C_LOOP :
        SUB     WORK, #1
        SKT1    Z
        BR      C_LOOP
        RET
```

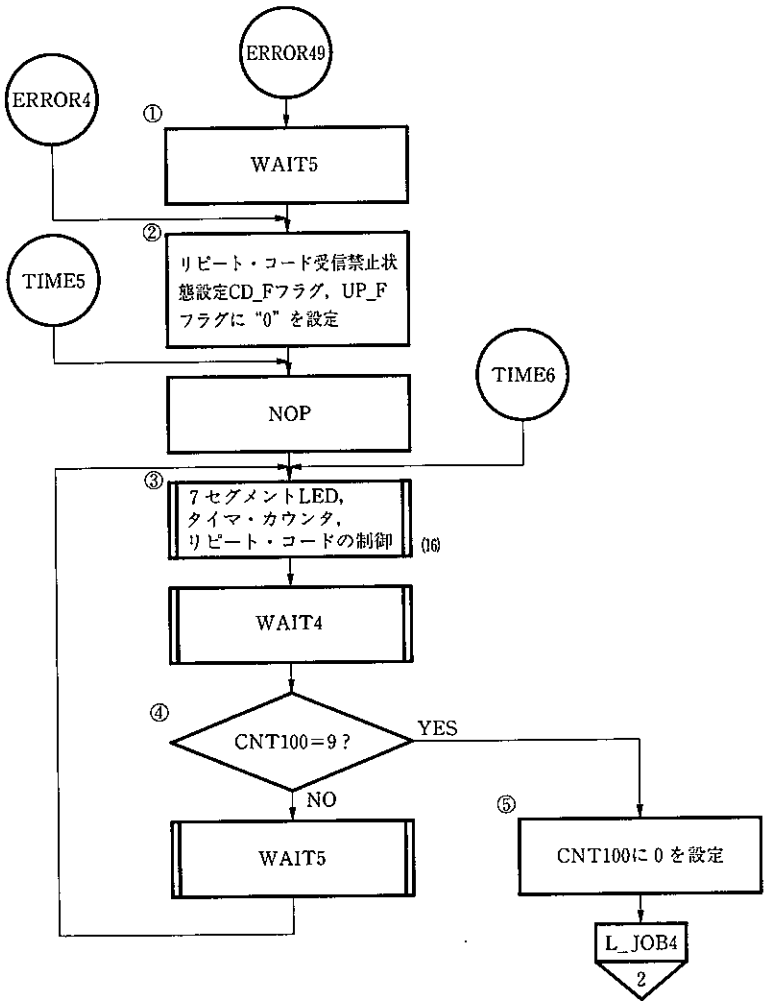
① 実行ステップ数を調整します。

(9)　エラー処理および実行ステップ数調整処理

リーダ・コード検出後，データ受信中に異常を検出するとエラー処理を行います。エラー処理では，リピート・コードの受信を禁止状態にします。

また，リーダ・コード検出後，データ受信処理を行い，再びリーダ・コードの検出処理を行うとき，実行ステップ数の調整を行います。データ受信中は，CNT100の値により実行ステップ数の調整を行っていますので，CNT100＝9のときにリーダ･コード検出処理の７セグメントLEDの制御を行うルーチンにとび，実行ステップ数の調整を行います。

フロー・チャート 27



```
ERROR49：
        CALL    WAIT5 ──────────────────①  実行ステップ数を調整します。
ERROR4：
        CLR3    CD_F,ROK_F,UP_F ────────②  異常検出として，リピート・コード受信禁止状態(ROK
                                           F=0)にします。また，CD_Fフラグ，UP_Fフラグ
                                           に“0”セットします。
TIME5：
        NOP
TIME6：
        CALL    TIME_CTL ──────────────③  7セグメントLED，タイマ・カウンタ，リピート・
                                           コードの制御を行います。
        CALL    WAIT4
        SKNE    CNT100,#9 }
        BR      L_JOB9    } ──────────④  実行ステップ数の調整を行います。CNT100＝9のと
                                           き，L_JOB9にとびます。
        CALL    WAIT5
        BR      TIME6
L_JOB9：
        MOV     CNT100,#0 ──────────⑤  CNT100の値を0に設定します。
        BR      L_JOB4
```

付録　プログラム・リスト

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-001

PROG =

COMMAND LINE OPTION =

SEQ FILE =


< MODULE OPTION >

  NOOBJECT/OBJECT          : OBJECT=¥ASMFILE¥10377ﾌﾘ.OBJ
  NOHEX/HEX                : HEX=¥ASMFILE¥10377ﾌﾘ.HEX
  NOPROM/PROM              : NOPROM
  NOLIST/LIST              : LIST=¥ASMFILE¥10377ﾌﾘ.PRN
  NOXREF/XREF              : XREF=¥ASMFILE¥10377ﾌﾘ.PRN
  NOERROR/ERROR            : NOERROR
  ROW                      : ROW=66
  NOGEN/GEN                : GEN
  COLUMN                   : COLUMN=130
  NOAUTOLOAD/AUTOLOAD      : NOAUTOLOAD
  NOCOND/COND              : COND
  NOSEQ/SEQ                : SEQ
  NOMAP/MAP                : NOMAP
  NODOCUMENT/DOCUMENT      : NODOCUMENT
  NOBRANCH/BRANCH          : NOBRANCH
  PROG                     : PROG=
  NOTAB/TAB                : NOTAB
  NOFORM/FORM              : NOFORM
  NOREPORT/REPORT          : NOREPORT
  NOPUBXREF/PUBXREF        : NOPUBXREF
  SUMMARY                  : SUMMARY=
  WORK                     : WORK=
  ZZZ0                     : ZZZ0=0
  ZZZ1                     : ZZZ1=0
  ZZZ2                     : ZZZ2=0
  ZZZ3                     : ZZZ3=0
  ZZZ4                     : ZZZ4=0
  ZZZ5                     : ZZZ5=0
  ZZZ6                     : ZZZ6=0
  ZZZ7                     : ZZZ7=0
  ZZZ8                     : ZZZ8=0
  ZZZ9                     : ZZZ9=0
  NOHOST/HOST              : NOHOST
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-002

PROG =

SOURCE = ¥ASMFILE¥10377ﾃﾞﾓ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
     1                     ;*****************************************************************
     2                     ;*                                                               *
     3                     ;*              uPD17103 ｱﾌﾟﾘｹｰｼｮﾝﾉｰﾄ用   ﾌﾟﾛｸﾞﾗﾑ                 *
     4                     ;*                                                               *
     5                     ;*****************************************************************
     6       0000          WORK     MEM      0.00H
     7       0001          CNT100   MEM      0.01H
     8       0002          MCNT1    MEM      0.02H
     9       0003          MCNT2    MEM      0.03H
    10       0004          SCNT1    MEM      0.04H
    11       0005          SCNT2    MEM      0.05H
    12       0006          MUSCNT1  MEM      0.06H
    13       0007          MUSCNT2  MEM      0.07H
    14       0008          MUSCNT3  MEM      0.08H
    15       0082          COUNT_F  FLG      0.08H.1
    16       0081          MODE_F   FLG      0.08H.0
    17                     ;
    18       0009          INCNT    MEM      0.09H
    19       000A          DATA1    MEM      0.0AH
    20       000B          DATA2    MEM      0.0BH
    21       000C          CNT1     MEM      0.0CH
    22       000D          CNT2     MEM      0.0DH
    23                     ;
    24       000E          FLAG1    MEM      0.0EH
    25       00E1                   SEL_F    FLG      0.0EH.0
    26                     ;
    27       000F          FLAG2    MEM      0.0FH
    28       00F8                   CD_F     FLG      0.0FH.3
    29       00F4                   ROK_F    FLG      0.0FH.2
    30       00F2                   UP_F     FLG      0.0FH.1
    31       00F1                   SET_F    FLG      0.0FH.0
    32                     ;
    33       0071          POB      MEM      0.71H
    34       0072          POC      MEM      0.72H
    35       0073          POD      MEM      0.73H
    36                     ;
    37                EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-003
PROG =
SOURCE = ¥ASMFILE¥10377ｿﾌﾄ.ASM
E STNO LOC. OBJ.    M  I SOURCE STATEMENT
     38 0000                    ORG      0000H
     39 0000 0C115              BR       MAIN
     40                  EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-004

PROG =

SOURCE = ¥ASMFILE¥10377~9.ASM

E STNO LOC. OBJ.   M  I SOURCE STATEMENT
    41                      ;*********************************************************************
    42                      ;* TIME CONTROL JOB                                                  *
    43                      ;*       リモコンを受信中は,100usごとに,この処理を行います。        *
    44                      ;*       CNT100の値により以下のように処理が分かれます。              *
    45                      ;*       CNT100          JOB                                         *
    46                      ;*         1-7           NO JOB                                      *
    47                      ;*          8            REPEAT JOB                                  *
    48                      ;*          9            CNT 1ms UP                                  *
    49                      ;*          A            7SEGMENT DATA CHANGE                        *
    50                      ;*********************************************************************
    51                TIME_CTL4:
    52 0001 074F0           NOP
    53                TIME_CTL3:
    54 0002 074F0           NOP
    55                TIME_CTL2:
    56 0003 074F0           NOP
    57                TIME_CTL1:
    58 0004 074F0           NOP
    59                TIME_CTL:
    60 0005 10011           ADD     CNT100,#1                       ;CNT100 ← CNT100 + 1 (100 us COUNTER UP)
    61 0006 0B01A           SKNE    CNT100,#0AH                     ;CNT100 ≠ 0AH ?
    62 0007 0C04C           BR      SEGMENT_JOB
    63 0008 0B01B           SKNE    CNT100,#0BH                     ;CNT100 ≠ 0BH ?
    64 0009 1D011           MOV     CNT100,#1                       ;CNT100 ← 1
    65 000A 0B018           SKNE    CNT100,#8                       ;CNT100 ≠ 8 ?
    66 000B 0C035           BR      RECNT
    67            1         SKF1    COUNT_F                         ;COUNT_F = 0 (NO COUNTING) ?
+    1 000C 1F082  1         SKF     .MF.COUNT_F SHR 4,#.DF.COUNT_F AND 0FH
    68 000D 09019           SKE     CNT100,#9                       ;CNT100 = 9 ?
    69 000E 0C046           BR      COUNT3
    70                EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-005

PROG =

SOURCE = ¥ASMFILE¥10377ﾟﾘ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
    71                     ;*************************************************
    72                     ;* COUNT1                                        *
    73                     ;*      タイマ・カウンタの制御を行います。       *
    74                     ;*************************************************
    75              COUNT1:
    76 000F 11084          SUB     MUSCNT3,#4                    ;MUSCNT3 ← MUSCNT3 - 4
    77 0010 13070          SUBC    MUSCNT2,#0                    ;MUSCNT2 ← MUSCNT2 - CY
    78 0011 13060          SUBC    MUSCNT1,#0                    ;MUSCNT1 ← MUSCNT1 - CY
    79              1      SKF1    CY                            ;CY= 0 ?
 +   1 0012 1F7F4  1       SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
    80 0013 14087          AND     MUSCNT3,#0111B                ;MUSCNT3 ← 000XB
    81              1      SKF1    CY                            ;CY = 0 ?
 +   1 0014 1F7F4  1       SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
    82 0015 1D078          MOV     MUSCNT2,#8                    ;MUSCNT2 ← 8
    83              1      SET1    BCD                           ;BCD ← 1 (10進数演算モード)
 +   1 0016 167E1  1       OR      .MF.BCD SHR 4,#.DF.BCD AND 0FH
    84 0017 13050          SUBC    SCNT2,#0                      ;SCNT2 ← SCNT2 - CY
    85 0018 13040          SUBC    SCNT1,#0                      ;SCNT1 ← SCNT1 - CY
    86 0019 13030          SUBC    MCNT2,#0                      ;MCNT2 ← MCNT2 - CY
    87 001A 13020          SUBC    MCNT1,#0                      ;MCNT2 ← MCNT1 - CY
    88              1      CLR1    BCD                           ;BCD ← 0 (16進数演算モード)
 +   1 001B 147EE  1       AND     .MF.BCD SHR 4,#.DF.(NOT BCD AND 0FH)
    89 001C 074F0          NOP
    90              1      SET2    CMP,Z                         ;CMP ← 1, Z ← 1
 +   1 001D 167FA  1       OR      .MF.CMP SHR 4,#.DF.(CMP OR Z) AND 0FH
    91 001E 11050          SUB     SCNT2,#0
    92 001F 11040          SUB     SCNT1,#0
    93 0020 11020          SUB     MCNT1,#0
    94              1      SKT1    Z                             ;Z = 0 (SCNT2,SCNT1,MCNT1 ALL 0) ?
 +   1 0021 1E7F2  1       SKT     .MF.Z SHR 4,#.DF.Z AND 0FH
    95 0022 0C02E          BR      CNTX
    96 0023 09031          SKE     MCNT2,#1                      ;MCNT2 = 1 (NO REMAIN TIME) ?
    97 0024 0C029          BR      CNTXX
    98              1      CLR2    MODE_F,COUNT_F                ;MODE_F ← 0 (タイマ動作モード)
 +   1 0025 1408C  1       AND     .MF.MODE_F SHR 4,#.DF.(NOT (MODE_F OR COUNT_F) AND 0FH)
    99                                                           ;COUNT_F ← 0 (NO COUNTING)
   100              1      CLR1    P0D3                          ;P0D3 ← Low LEVEL OUTPUT SET (LED OFF)
 +   1 0026 14737  1       AND     .MF.P0D3 SHR 4,#.DF.(NOT P0D3 AND 0FH)
   101 0027 1D711          MOV     P0B,#1
   102 0028 0C02D          BR      RET_JOB
   103                     ;
   104              CNTXX:
   105 0029 09032          SKE     MCNT2,#2                      ;MCNT2 = 2 (REMAIN TIME 1 MINUTE) ?
   106 002A 0C032          BR      CNTXXX
   107 002B 1D046          MOV     SCNT1,#6                      ;SCMT1 ← 6
   108 002C 1D031          MOV     MCNT2,#1                      ;MCNT2 ← 1
   109              RET_JOB:
   110 002D 070E0          RET
   111                     ;
   112              CNTX:
   113 002E 074F0          NOP
   114 002F 074F0          NOP
   115 0030 074F0          NOP
   116 0031 074F0          NOP
   117              CNTXXX:
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-006

PROG =

SOURCE = ¥ASMFILE¥10377'9.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   118 0032 0B049              SKNE    SCNT1,#9                    ;SCNTI = 9 ?
   119 0033 1D045              MOV     SCNT1,#5                    ;SCNTI ← 5
   120 0034 070E0              RET
   121                         ;
   122                 EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-007

PROG =

SOURCE = ¥ASMFILE¥10377`ﾘ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   123                      ;*************************************************************
   124                      ;* RECNT                                                     *
   125                      ;*      216ms 間に受信したリピート・コード数により           *
   126                      ;*      リピート・コードの有効  無効を判定します。           *
   127                      ;*************************************************************
   128                  RECNT:
   129              1           SKT1    ROK_F                                   ;ROK_F = 1 (REPEAT CODE RECEIVE OK) ?
 +   1 0035 1E0F4   1           SKT     .MF.ROK_F SHR 4,#.DF.ROK_F AND 0FH
   130 0036 0C045               BR      COUNT2
   131 0037 1D007               MOV     WORK,#7                                 ;WORK ← 7
   132 0038 100A1               ADD     DATA1,#1                                ;DATA1 ← DATA1 + 1(108ms COUNTER UP)
   133 0039 120B0               ADDC    DATA2,#0
   134              1           SKT1    CY                                      ;CY = 1 (108ms COUNT) ?
 +   1 003A 1E7F4   1           SKT     .MF.CY SHR 4,#.DF.CY AND 0FH
   135 003B 0C047               BR      COUNT4
   136 003C 1D0A8               MOV     DATA1,#8                                ;DATA1 ← 8 (108ms COUNTER INITIAL)
   137 003D 1D0B2               MOV     DATA2,#2                                ;DATA2 ← 2 (108ms COUNTER INITIAL)
   138 003E 190E6               SKGE    FLAG1,#6                                ;FLAG1 > 6 (REPEAT CODE COUNTER1 >=3) ?
   139 003F 190E2               SKGE    FLAG1,#2                                ;FLAG1 > 2 (REPEAT CODE COUNTER1 >=1) ?
   140              1           CLR1    ROK_F                                   ;ROK_F ← 0 (REPEAT CODE RECEIVE STOP)
 +   1 0040 140FB   1           AND     .MF.ROK_F SHR 4,#.DF.(NOT ROK_F AND 0FH)
   141 0041 140E1               AND     FLAG1,#0001B                            ;FLAG1 ← 000XB (REPEAT CODE COUNTER1 INITIAL)
   142 0042 1D004               MOV     WORK,#4                                 ;WORK ← 4
   143 0043 074F0               NOP
   144 0044 0C047               BR      COUNT4
   145                          ;
   146                  COUNT2:
   147 0045 074F0               NOP
   148                  COUNT3:
   149 0046 1D008               MOV     WORK,#8                                 ;WORK ← 8
   150                  COUNT4:
   151 0047 074F0               NOP
   152                  C_LOOP:
   153 0048 11001               SUB     WORK,#1                                 ;WORK ← WORK - 1
   154              1           SKT1    Z                                       ;Z = 1 (WORK = 0) ?
 +   1 0049 1E7F2   1           SKT     .MF.Z SHR 4,#.DF.Z AND 0FH
   155 004A 0C048               BR      C_LOOP
   156 004B 070E0               RET
   157                          ;
   158                  EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-008

PROG =

SOURCE = ¥ASMFILE¥10377ﾃﾞﾓ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   158                      ;*********************************************************************
   160                      ;* 7SEGMENT LED CONTROL                                              *
   161                      ;*                                                                   *
   162                      ;*        以下に示すフラグの状態によって，POC ,PODに                 *
   163                      ;*      7SEGMEND LEDに表示するパターンを設定します。                 *
   164                      ;*      なお,このルーチンには,1ｍｓごとに飛んできます。             *
   165                      ;*                                                                   *
   166                      ;*      COUNT=1                                                      *
   167                      ;*              MODE_F=0                        DATA 表示            *
   168                      ;*              MODE_F=1                        -- 表示              *
   169                      ;*      COUNT=0                                                      *
   170                      ;*              MODE_F=0                        00 点滅              *
   171                      ;*              MODE_F=1                                             *
   172                      ;*                              SET_F=1         DATA 表示            *
   173                      ;*                              SET_F=0         -- 表示              *
   174                      ;*********************************************************************
   175              SEGMENT_JOB:
   176           1          NOT1    SEL_F                                   ;SEL_F CHANGE
 +   1 004C 150E1  1          XOR     .MF.SEL_F SHR 4,#.DF.SEL_F AND 0FH
   177           1          SKT1    COUNT_F                                 ;COUNT_F = 1 (COUNTING) ?
 +   1 004D 1E082  1          SKT     .MF.COUNT_F SHR 4,#.DF.COUNT_F AND 0FH
   178           1          SKF1    MODE_F                                  ;MODE_F = 0 (タイマ設定モード) ?
 +   1 004E 1F081  1          SKF     .MF.MODE_F SHR 4,#.DF.MODE_F AND 0FH
   179 004F 0C067           BR      COMON_OFF
   180 0050 1D72D           MOV     POC,#0DH                                ;SEGMENT OUTPUT CONTROL
   181 0051 16737           OR      POD,#0111B
   182 0052 1D002           MOV     WORK,#2                                 ;WORK ← 2
   183 0053 10021           ADD     MCNT1,#1                                ;MCNT1 ← MCNT1 + 1 (250ms COUNTER UP)
   184 0054 12030           ADDC    MCNT2,#0                                ;MCNT2 ← MCNT2 + CY (250ms COUNTER UP)
   185           1          SKT1    CY                                      ;CY=1 (250ms COUNT) ?
 +   1 0055 1E7F4  1          SKT     .MF.CY SHR 4,#.DF.CY AND 0FH
   186 0056 0C05F           BR      FLASH_LOOP1
   187           1          SKT1    POB2                                    ;POB2 = 0 ?
 +   1 0057 1E714  1          SKT     .MF.POB2 SHR 4,#.DF.POB2 AND 0FH
   188 0058 0C05B           BR      ALL_ON
   189 0059 1D711           MOV     POB,#1                                  ;COMON OUTPUT CONTROL
   190 005A 0C05D           BR      FLASH_JOB2
   191                      ;
   192              ALL_ON:
   193 005B 1D717           MOV     POB,#7
   194 005C 074F0           NOP
   195              FLASH_JOB2:
   196 005D 1D026           MOV     MCNT1,#6                                ;MCNT1 ← 6 (250ms COUNTER INITIAL)
   197 005E 0C062           BR      FLASH_JOB3
   198                      ;
   199              FLASH_LOOP1:
   200 005F 11001           SUB     WORK,#1                                 ;WORK ← WORK - 1
   201           1          SKT1    Z                                       ;Z = 1 (WORK = 0) ?
 +   1 0060 1E7F2  1          SKT     .MF.Z SHR 4,#.DF.Z AND 0FH
   202 0061 0C05F           BR      FLASH_LOOP1
   203              FLASH_JOB3:
   204 0062 1D005           MOV     WORK,#5                                 ;WORK ← 5
   205              FLASH_LOOP2:
   206 0063 11001           SUB     WORK,#1                                 ;WORK ← WORK - 1
   207           1          SKT1    Z                                       ;Z = 1 (WORK = 0) ?
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-009

PROG =

SOURCE = ¥ASMFILE¥10377`9.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
  +    1 0064 1E7F2  1             SKT      .MF.Z SHR 4,#.DF.Z AND OFH
     208 0065 0C063               BR       FLASH_LOOP2
     209 0066 070E0               RET
     210                          ;
     211                EJECT
```

```
PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾘ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   212                          ;*****************************************
   213                          ;* COMON_OFF                             *
   214                          ;*      COMON 端子の出力をオフします。   *
   215                          ;*****************************************
   216                  COMON_OFF:
   217 0067 1D717               MOV     POB,#7                                  ;COMON OUTPUT ALL OFF
   218 0068 14738               AND     POD,#1000B                              ;SEGMENT OUTPUT OFF
   219              1           SKT1    COUNT_F                                 ;COUNT_F = 1 (COUNTING) ?
 +   1 0069 1E082   1           SKT     .MF.COUNT_F SHR 4,#.DF.COUNT_F AND 0FH
   220 006A 0C0BD               BR      NO_COUNT
   221              1           SKF1    MODE_F                                  ;MODE_F = 0 (タイマ設定モード) ?
 +   1 006B 1F081   1           SKF     .MF.MODE_F SHR 4,#.DF.MODE_F AND 0FH
   222 006C 0C0C2               BR      BAR_SET
   223                  EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-011

PROG =

SOURCE = ¥ASMFILE¥10377ﾞ9.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   224                      ;*********************************************************************
   225                      ;*       7 SGMENT DATA SET                                           *
   226                      ;*               POC ,POD に表示パターンを設定します。               *
   227                      ;*               ･MCNT1=0 AND MCNT2=0                                *
   228                      ;*                       DISPNT=1XXXB          ¥ORK=MCNT2          *
   229                      ;*                       DISPNT=0XXXB          ¥ORK=MCNT1          *
   230                      ;*               ･MCNT1 NOT=0 OR MCNT2 NOT=0                         *
   231                      ;*                       DISPNT=1XXXB          ¥ORK=SCNT2          *
   232                      ;*                       DISPNT=0XXXB          ¥ORK=SCNT1          *
   233                      ;*********************************************************************
   234 008D 0B020           SKNE    MCNT1,#0                              ;MCNT1 = 0 (TIMER COUNTER < 1 MINUTE) ?
   235 008E 09031           SKE     MCNT2,#1                              ;MCNT2 = 1 (TIMER COUNTER < 1 MINUTE) ?
   236 008F 0C078           BR      MCNT_SET
   237              1       SKF1    SEL_F                                 ;SEL_F = 0 (TIMER COUNTER SETING) ?
 +   1 0070 1F0E1   1       SKF     .MF.SEL_F SHR 4,#.DF.SEL_F AND 0FH
   238 0071 0C076           BR      SCNT2_SET
   239 0072 08040           LD      ¥ORK,SCNT1                            ;¥ORK ← SCNT1
   240 0073 09000           SKE     ¥ORK,#0                               ;¥OKR = 0 (TIMER COUNTER < 9 SECONDS) ?
   241 0074 0C084           BR      DATA_SET
   242 0075 0C085           BR      DISP_OFF
   243                  SCNT2_SET:
   244 0076 08050           LD      ¥ORK,SCNT2                            ;¥ORK ← SCNT2
   245 0077 0C083           BR      DATA_SET11
   246                  MCNT_SET:
   247              1       SKF1    SEL_F                                 ;SEL_F = 0 ?
 +   1 0078 1F0E1   1       SKF     .MF.SEL_F SHR 4,#.DF.SEL_F AND 0FH
   248 0079 0C081           BR      MCNT2_SET
   249 007A 08020           LD      ¥ORK,MCNT1                            ;¥ORK ← MCNT1
   250 007B 09000           SKE     ¥ORK,#0                               ;¥ORK = 0 (TIMER COUNTER < 9 MINUTES) ?
   251 007C 0C084           BR      DATA_SET
   252 007D 074F0           NOP
   253 007E 09030           SKE     MCNT2,#0                              ;MCNT2 = 0 ?
   254 007F 0C0B7           BR      DISP_OFF1
   255 0080 0C0A6           BR      DATA_SET00
   256                  MCNT2_SET:
   257 0081 08030           LD      ¥ORK,MCNT2                            ;¥ORK ← MCNT2
   258 0082 074F0           NOP
   259                  DATA_SET11:
   260 0083 074F0           NOP
   261                  DATA_SET:
   262 0084 19004           SKGE    ¥ORK,#4                               ;¥ORK >= 4 ?
   263 0085 0C09C           BR      DATA_SET0
   264 0086 19007           SKGE    ¥ORK,#7                               ;¥ORK >= 7 ?
   265 0087 0C092           BR      DATA_SET4
   266 0088 1D72F           MOV     POC,#0FH                              ;POC ← 0FH
   267 0089 16736           OR      POD,#0110B                            ;POD ← X11XB
   268 008A 09007           SKE     ¥ORK,#7H                              ;¥ORK = 7 ?
   269 008B 0C08F           BR      DATA_SET8
   270 008C 1D725           MOV     POC,#5                                ;POC ← 5
   271 008D 1473C           AND     POD,#1100B                            ;POD ← XX00B
   272 008E 0C0B0           BR      COMON_ON
   273                  DATA_SET8:
   274 008F 0B008           SKNE    ¥ORK,#8                               ;¥ORK = 8 ?
   275 0090 16737           OR      POD,#0111B                            ;POD ← X111B
   276 0091 0C0B0           BR      COMON_ON
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-012

PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾟﾛｸﾞﾗﾑ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   277                   DATA_SET4:
   278 0092 1D72E              MOV     POC,#0EH                    ;POC ← 0EH
   279 0093 16737              OR      POD,#0111B                  ;POD ← X111B
   280 0094 09004              SKE     WORK,#4                     ;WORK = 4 ?
   281 0095 0C099              BR      DATA_SET5
   282 0096 1D72B              MOV     POC,#0BH                    ;POC ← 0BH
   283 0097 1473C              AND     POD,#1100B                  ;POD ← XX00B
   284 0098 0C0B0              BR      COMON_ON
   285                   DATA_SET5:
   286 0099 0B005              SKNE    WORK,#5                     ;WORK = 5 ?
   287 009A 1473E              AND     POD,#1110B                  ;POD ← XXX0B
   288 009B 0C0B0              BR      COMON_ON
   289                   EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-013

PROG =

SOURCE = ¥ASMFILE¥10377ﾀﾞ9.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   290                   DATA_SET0:
   291 009C 19002                SKGE    WORK,#2                          ;WORK >= 2 ?
   292 009D 0C0A6                BR      DATA_SET00
   293 009E 09002                SKE     WORK,#2                          ;WORK = 2 ?
   294 009F 0C0A3                BR      DATA_SET3
   295 00A0 1D727                MOV     POC,#7                           ;POC ← 7
   296 00A1 16733                OR      POD,#0011B                       ;POD ← XX11B
   297 00A2 0C0AE                BR      MAKE_JOB2
   298                   DATA_SET3:
   299 00A3 1D727                MOV     POC,#7                           ;POC ← 7
   300 00A4 16736                OR      POD,#0110B                       ;POD ← X11XB
   301 00A5 0C0AE                BR      MAKE_JOB2
   302                   DATA_SET00:
   303 00A6 09000                SKE     WORK,#0                          ;WORK = 0 ?
   304 00A7 0C0AB                BR      DATA_SET1
   305 00A8 1D72D                MOV     POC,#0DH                         ;POC ← 0DH
   306 00A9 16737                OR      POD,#0111B                       ;POD ← X111B
   307 00AA 0C0AE                BR      MAKE_JOB2
   308                   DATA_SET1:
   309 00AB 1D721                MOV     POC,#1                           ;POC ← 1
   310 00AC 16734                OR      POD,#0100B                       ;POD ← X1XXB
   311                   MAKE_JOB1:
   312 00AD 074F0                NOP
   313                   MAKE_JOB2:
   314 00AE 074F0                NOP
   315 00AF 074F0                NOP
   316                           ;*****************************************
   317                           ;* COMON_OFF                             *
   318                           ;*      COMON 端子の出力をオンします。   *
   319                           ;*****************************************
   320                   COMON_ON:
   321                1          SKT1    SEL_F                            ;SEL_F = 1 ?
 +   1 00B0 1E0E1     1          SKT     .MF.SEL_F SHR 4,#.DF.SEL_F AND 0FH
   322 00B1 1D715                MOV     POB,#5                           ;POB1 Low OUTPUT SET
   323                1          SKF1    SEL_F                            ;SEL_F = 0 ?
 +   1 00B2 1F0E1     1          SKF     .MF.SEL_F SHR 4,#.DF.SEL_F AND 0FH
   324 00B3 1D713                MOV     POB,#3                           ;POB2 Low OUTPUT SET
   325 00B4 070E0                RET
   326                           ;
   327                   EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-014

PROG =

SOURCE = ¥ASMFILE¥10377゛ﾘ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   328                   DISP_OFF:
   329 00B5 074F0                NOP
   330 00B6 074F0                NOP
   331                   DISP_OFF1:
   332 00B7 1D003                MOV     WORK,#3                                     ;WORK ← 3
   333 00B8 074F0                NOP
   334                   DISP_LOOP:
   335 00B9 11001                SUB     WORK,#1                                     ;WORK ← WORK - 1
   336                 1         SKF1    Z                                           ;Z = 0 (WORK ≠ 0 ) ?
 +   1 00BA 1F7F2      1         SKF     .MF.Z SHR 4,#.DF.Z AND 0FH
   337 00BB 070E0                RET
   338 00BC 0C0B9                BR      DISP_LOOP
   339                           ;
   340                   EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-015

PROG =

SOURCE = ¥ASMFILE¥10377ﾃﾞﾘ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   341                      ;*********************************************************************
   342                      ;*       7 SGMENT -- SET                                             *
   343                      ;*              POC,PODに "--"表示のパターンを設定します。           *
   344                      ;*********************************************************************
   345                  NO_COUNT:
   346              1           SKT1    SET_F                           ;SET_F = 1 (TIMER COUNTER SETING) ?
 +   1 00BD 1E0F1   1           SKT     .MF.SET_F SHR 4,#.DF.SET_F AND 0FH
   347 00BE 0C0C2               BR      BAR_SET
   348 00BF 074F0               NOP
   349 00C0 074F0               NOP
   350 00C1 0C078               BR      MCNT_SET
   351                          ;
   352                  BAR_SET:
   353 00C2 1D722               MOV     POC,#2                          ;POC1 High OUTPUT SET
   354 00C3 16730               OR      POD,#0
   355 00C4 1D004               MOV     WORK,#4                         ;WORK ← 4
   356                  BAR_LOOP:
   357 00C5 11001               SUB     WORK,#1                         ;WORK ← WORK - 1
   358              1           SKT1    Z                               ;Z = 1 (WORK = 0) ?
 +   1 00C6 1E7F2   1           SKT     .MF.Z SHR 4,#.DF.Z AND 0FH
   359 00C7 0C0C5               BR      BAR_LOOP
   360 00C8 0C0AD               BR      MAKE_JOB1
   361                          ;
   362                  EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-016

PROG =

SOURCE = ¥ASMFILE¥10377ﾀﾞ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   363                      ;*****************************************************
   364                      ;* CNT UP JOB                                        *
   365                      ;*       CNT1,CNT2の値をインクリメントします。       *
   366                      ;*****************************************************
   367                CNT_UP:
   368 00C9 100D1           ADD      CNT2,#1                                    ;CNT2 ← CNT2 + 1
   369 00CA 120C0           ADDC     CNT1,#0                                    ;CNT1 ← CNT1 + CY
   370              1       SKT1     CY                                         ;CY = 1 (COUNT OVER) ?
 +   1 00CB 1E7F4   1       SKT      .MF.CY SHR 4,#.DF.CY AND 0FH
   371 00CC 070E0           RET
   372 00CD 071E0           RETSK                                               ;ERROR
   373                      ;
   374                      ;*************************************************************
   375                      ;* WAIT処理                                                  *
   376                      ;*       NOP命令により実行ステップの調整を行います。        *
   377                      ;*************************************************************
   378                WAIT10:
   379 00CE 074F0           NOP
   380                WAIT9:
   381 00CF 074F0           NOP
   382                WAIT8:
   383 00D0 074F0           NOP
   384                WAIT7:
   385 00D1 074F0           NOP
   386                WAIT6:
   387 00D2 074F0           NOP
   388                WAIT5:
   389 00D3 074F0           NOP
   390                WAIT4:
   391 00D4 074F0           NOP
   392                WAIT3:
   393 00D5 074F0           NOP
   394                WAIT2:
   395 00D6 070E0           RET
   396                      ;
   397                EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-017

PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾘ.ASM

E STNO LOC. OBJ.   M  I SOURCE STATEMENT
   398                      ;*************************************************************
   399                      ;* SET1_1                                                    *
   400                      ;*      INCNTの値によってDATA1,またはDATA2                   *
   401                      ;*      の１ビットをセットします。                           *
   402                      ;*              INCNT           DATA1 SET                    *
   403                      ;*                0             XXX1                         *
   404                      ;*                1             XX1X                         *
   405                      ;*                2             X1XX                         *
   406                      ;*                3             1XXX                         *
   407                      ;*              INCNT           DATA2 SET                    *
   408                      ;*                4             XXX1                         *
   409                      ;*                5             XX1X                         *
   410                      ;*                6             X1XX                         *
   411                      ;*                7             1XXX                         *
   412                      ;*************************************************************
   413                  SET1_1:
   414 00D7 1B094           SKLT    INCNT,#4                        ;INCNT < 4 ?
   415 00D8 0C0E2           BR      SET1_2
   416 00D9 0B090           SKNE    INCNT,#0                        ;INCNT = 0 ?
   417 00DA 160A1           OR      DATA1,#0001B                    ;DATA1 ← XXX1B
   418 00DB 0B091           SKNE    INCNT,#1                        ;INCNT = 1 ?
   419 00DC 160A2           OR      DATA1,#0010B                    ;DATA1 ← XX1XB
   420 00DD 0B092           SKNE    INCNT,#2                        ;INCNT = 2 ?
   421 00DE 160A4           OR      DATA1,#0100B                    ;DATA1 ← X1XXB
   422 00DF 0B093           SKNE    INCNT,#3                        ;INCNT = 3 ?
   423 00E0 160A8           OR      DATA1,#1000B                    ;DATA1 ← 1XXXB
   424 00E1 070E0           RET
   425                  SET1_2:
   426 00E2 0B094           SKNE    INCNT,#4                        ;INCNT = 4 ?
   427 00E3 160B1           OR      DATA2,#0001B                    ;DATA2 ← XXX1B
   428 00E4 0B095           SKNE    INCNT,#5                        ;INCNT = 5 ?
   429 00E5 160B2           OR      DATA2,#0010B                    ;DATA2 ← XX1XB
   430 00E6 0B096           SKNE    INCNT,#6                        ;INCNT = 6 ?
   431 00E7 160B4           OR      DATA2,#0100B                    ;DATA2 ← X1XXB
   432 00E8 0B097           SKNE    INCNT,#7                        ;INCNT = 7 ?
   433 00E9 160B8           OR      DATA2,#1000B                    ;DATA2 ← 1XXXB
   434 00EA 070E0           RET
   435                  EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-018

PROG =

SOURCE = ¥ASMFILE¥10377'9.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   436                      ;*************************************************************
   437                      ;* DATA_COMPARE                                              *
   438                      ;*       INCNTの値によってDATA1,またはDATA2                  *
   439                      ;*       の1ビットと今,入力したデータの比較                  *
   440                      ;*       を行います。                                        *
   441                      ;*                INCNT           DATA1 SET                *
   442                      ;*                  8             XXX1                     *
   443                      ;*                  9             XX1X                     *
   444                      ;*                  A             X1XX                     *
   445                      ;*                  B             1XXX                     *
   446                      ;*                INCNT           DATA2 SET                *
   447                      ;*                  C             XXX1                     *
   448                      ;*                  D             XX1X                     *
   449                      ;*                  E             X1XX                     *
   450                      ;*                  F             1XXX                     *
   451                      ;*                IN                                     *
   452                      ;*                UP_F=1          入力  1                  *
   453                      ;*                UP_F=0          入力  0                  *
   454                      ;*                OUT                                    *
   455                      ;*                UP_F=1          ERROR                    *
   456                      ;*                UP_F=0          NO ERROR                 *
   457                      ;*************************************************************
   458                DATA_COMP:
   459 00EB 1B09C           SKLT    INCNT,#0CH                              ;INCNT < 0CH ?
   460 00EC 0C101           BR      DATA_COMP11
   461 00ED 1B09A           SKLT    INCNT,#0AH                              ;INCNT < 0AH ?
   462 00EE 0C0F8           BR      DATA_COMP3
   463 00EF 09098           SKE     INCNT,#8                                ;INCNT = 8 ?
   464 00F0 0C0F3           BR      DATA_COMP1
   465 00F1 1E0A1           SKT     DATA1,#0001B                            ;DATA1 = XXX1B ?
   466              1       NOT1    UP_F                                    ;UP_F CHANGE
 +   1 00F2 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   467                DATA_COMP1:
   468 00F3 09099           SKE     INCNT,#9                                ;INCNT = 9 ?
   469 00F4 0C0F7           BR      DATA_COMP2
   470 00F5 1E0A2           SKT     DATA1,#0010B                            ;DATA1 = XX1XB ?
   471              1       NOT1    UP_F
 +   1 00F6 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   472                DATA_COMP2:
   473 00F7 070E0           RET
   474                      ;
   475                DATA_COMP3:
   476 00F8 0909A           SKE     INCNT,#0AH                              ;INCNT = 0AH ?
   477 00F9 0C0FC           BR      DATA_COMP4
   478 00FA 1E0A4           SKT     DATA1,#0100B                            ;DATA1 = X1XXB ?
   479              1       NOT1    UP_F                                    ;UP_F CHANGE
 +   1 00FB 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   480                DATA_COMP4:
   481 00FC 0909B           SKE     INCNT,#0BH                              ;INCNT = 0BH ?
   482 00FD 0C100           BR      DATA_COMP5
   483 00FE 1E0A8           SKT     DATA1,#1000B                            ;INCNT = 1XXXB ?
   484              1       NOT1    UP_F                                    ;UP_F CHANGE
 +   1 00FF 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   485                DATA_COMP5:
   486 0100 070E0           RET
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-019

PROG =

SOURCE = ¥ASMFILE¥10377`9.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   487                      ;
   488                  DATA_COMP11:
   489 0101 1B09E           SKLT    INCNT,#0EH                          ;INCNT < 0EH ?
   490 0102 0C10C           BR      DATA_COMP6
   491 0103 0909C           SKE     INCNT,#0CH                          ;INCNT = 0CH ?
   492 0104 0C107           BR      DATA_COMP7
   493 0105 1E0B1           SKT     DATA2,#0001B                        ;DATA2 = XXX1B ?
   494              1       NOT1    UP_F                                ;UP_F CHANGE
 +   1 0106 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   495                  DATA_COMP7:
   496 0107 0909D           SKE     INCNT,#0DH                          ;INCNT = 0DH ?
   497 0108 0C10B           BR      DATA_COMP8
   498 0109 1E0B2           SKT     DATA2,#0010B                        ;DATA2 = XX1XB ?
   499              1       NOT1    UP_F                                ;UP_F CHANGE
 +   1 010A 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   500                  DATA_COMP8:
   501 010B 070E0           RET
   502                      ;
   503                  DATA_COMP6:
   504 010C 0909E           SKE     INCNT,#0EH                          ;INCNT = 0EH ?
   505 010D 0C110           BR      DATA_COMP9
   506 010E 1E0B4           SKT     DATA2,#0100B                        ;DATA2 = X1XXB ?
   507              1       NOT1    UP_F                                ;UP_F CHANGE
 +   1 010F 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   508                  DATA_COMP9:
   509 0110 0909F           SKE     INCNT,#0FH                          ;INCNT = 0FH ?
   510 0111 0C114           BR      DATA_COMP10
   511 0112 1E0B8           SKT     DATA2,#1000B                        ;DATA2 = 1XXXB ?
   512              1       NOT1    UP_F                                ;UP_F CHANGE
 +   1 0113 150F2   1       XOR     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   513                  DATA_COMP10:
   514 0114 070E0           RET
   515                      ;
   516                  EJECT
```

```
PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾟﾛｸﾞﾗﾑ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   517                       ;*********************************
   518                       ;* MAIN                          *
   519                       ;*      初期設定を行います。     *
   520                       ;*********************************
   521                 MAIN:
   522 0115 1D010            MOV     CNT100,#0                                ;CNT100 ← 0
   523              1        CLR2    COUNT_F,MODE_F                           ;COUNT_F ← 0 (NOT COUNTING)
 +   1 0116 1408C  1        AND     .MF.COUNT_F SHR 4,#.DF.(NOT (COUNT_F OR MODE_F) AND OFH)
   524                                                                         ;MODE_F ← 0 (タイマ動作モード)
   525              1        CLR1    SEL_F                                    ;SEL_F ← 0
 +   1 0117 140EE  1        AND     .MF.SEL_F SHR 4,#.DF.(NOT SEL_F AND OFH)
   526              1        CLR2    ROK_F,SET_F                              ;ROK_F ← 0 (REPEAT CODE RECEIVE STOP)
 +   1 0118 140FA  1        AND     .MF.ROK_F SHR 4,#.DF.(NOT (ROK_F OR SET_F) AND OFH)
   527                                                                         ;SET_F ← 0 (TIMER COUNTER NO SETTING)
   528                 EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-021

PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾟﾛ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   529                      ;**************************************************
   530                      ;* L_JOB1                                         *
   531                      ;*      リーダ・コードの検出処理を行います。      *
   532                      ;**************************************************
   533                L_JOB1:
   534 0119 1C035           CALL    RECNT                                   ;REPEAT JOB
   535 011A 1C0D0           CALL    WAIT8
   536              1       SKT1    COUNT_F                                 ;COUNT_F = 1 (COUNTING) ?
 +   1 011B 1E082   1       SKT     .MF.COUNT_F SHR 4,#.DF.COUNT_F AND 0FH
   537 011C 0C11F           BR      L_JOB2
   538 011D 1C00F           CALL    COUNT1                                  ;TIMER COUNTER CONTROL
   539 011E 0C122           BR      L_JOB3
   540                L_JOB2:
   541 011F 1C0CE           CALL    WAIT10
   542 0120 1C0CE           CALL    WAIT10
   543 0121 1C0D0           CALL    WAIT8
   544                L_JOB3:
   545 0122 1C0D2           CALL    WAIT6
   546                L_JOB4:
   547 0123 1C0D1           CALL    WAIT7
   548 0124 1C04C           CALL    SEGMENT_JOB                             ;7 SEGMENT CONTROL
   549              1       SKT1    SEL_F                                   ;SEL_F = 1 ?
 +   1 0125 1E0E1   1       SKT     .MF.SEL_F SHR 4,#.DF.SEL_F AND 0FH
   550 0126 0C129           BR      L_CHECK
   551 0127 1C0D4           CALL    WAIT4
   552 0128 0C12E           BR      WAIT_375
   553                L_CHECK:
   554              1       SKF1    POB0                                    ;LEADER CODE CHECK
 +   1 0129 1F711   1       SKF     .MF.POB0 SHR 4,#.DF.POB0 AND 0FH
   555 012A 1D01F           MOV     CNT100,#0FH                             ;Low LEVEL COUNTER (CNT100) INITIAL
   556 012B 10011           ADD     CNT100,#1                               ;CNT100 ← CNT100 + 1
   557 012C 0B014           SKNE    CNT100,#4                               ;CNT100 = 4 (6 ms CONTINUING Low LEVEL) ?
   558 012D 0C137           BR      L_JOB5
   559                WAIT_375:
   560 012E 1D00D           MOV     WORK,#0DH                               ;WORK = 0DH ?
   561                L_LOOP:
   562 012F 1C0CE           CALL    WAIT10
   563 0130 1C0CE           CALL    WAIT10
   564 0131 1C0D3           CALL    WAIT5
   565 0132 11001           SUB     WORK,#1                                 ;WORK ← WORK - 1
   566              1       SKT1    Z                                       ;Z = 1 (WORK = 0) ?
 +   1 0133 1E7F2   1       SKT     .MF.Z SHR 4,#.DF.Z AND 0FH
   567 0134 0C12F           BR      L_LOOP
   568 0135 1C0CF           CALL    WAIT9
   569 0136 0C119           BR      L_JOB1
   570                EJECT
```

PROG =

SOURCE = ¥ASMFILE¥10377ﾘ.ASM

```
E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   571                          ;*****************************************************************************
   572                          ;* L_JOB5                                                                    *
   573                          ;*       リーダ・コードの立ち上がりから次のコードの立ち下がりまでの時間      *
   574                          ;*       受信コードを以下のように判定します。                                *
   575                          ;*                2ms未満                異常検出                           *
   576                          ;*                2ms以上2.6ms未満       リピート・コード                   *
   577                          ;*                2.6ms以上4.0ms未満     異常検出                           *
   578                          ;*                4.0ms以上5.0ms未満     カスタム・コード                   *
   579                          ;*                5ms以上                異常検出                           *
   580                          ;*****************************************************************************
   581                   L_JOB5:
   582 0137 1D010                MOV     CNT100,#0                       ;CNT100 ← 0 (100 us COUNTER INITIAL)
   583                   L_JOB6:
   584 0138 1C001                CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER, REPEAT CODE CONTROL
   585 0139 1C0D2                CALL    WAIT6
   586              1            SKT1    POBO                            ;POBO = High LEVEL (UP EDGE DETECT) ?
 +   1 013A 1E711   1            SKT     .MF.POBO SHR 4,#.DF.POBO AND 0FH
   587 013B 0C138                BR      L_JOB6
   588 013C 1C001                CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER, REPEAT CODE CONTROL
   589 013D 1C0D2                CALL    WAIT6
   590              1            SKT1    POBO                            ;POBO = High LEVEL (CHATRING) ?
 +   1 013E 1E711   1            SKT     .MF.POBO SHR 4,#.DF.POBO AND 0FH
   591 013F 0C138                BR      L_JOB6
   592 0140 1C001                CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER, REPEAT CODE CONTROL
   593 0141 1D0DE                MOV     CNT2,#0EH                       ;CNT2 ← 0EH (CODE JUDGE CONTER INITIAL)
   594 0142 1D0CC                MOV     CNT1,#0CH                       ;CNT1 ← 0CH (CODE JUDGE CONTER INITIAL)
   595 0143 1C0D4                CALL    WAIT4
   596                   L_JOB7:
   597              1            SKT1    POBO                            ;POBO = High LEVEL (DOWN EDGE DETECT) ?
 +   1 0144 1E711   1            SKT     .MF.POBO SHR 4,#.DF.POBO AND 0FH
   598 0145 0C14A                BR      TIME_JUDGE1
   599 0146 1C001                CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER, REPEAT CODE CONTROL
   600 0147 1C0C9                CALL    CNT_UP                          ;CODE JUDGE COUNTER UP
   601 0148 0C144                BR      L_JOB7
   602 0149 0C1B5                BR      ERROR2
   603                   TIME_JUDGE1:
   604 014A 1C001                CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER, REPEAT CODE CONTROL
   605 014B 110D1                SUB     CNT2,#1
   606 014C 130CE                SUBC    CNT1,#0EH
   607              1            SKF1    CY                              ;CY = 0 (CODE JUDGE COUNTER >= 2ms) ?
 +   1 014D 1F7F4   1            SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
   608 014E 0C1AC                BR      ERROR48
   609 014F 110D6                SUB     CNT2,#6
   610 0150 130C0                SUBC    CNT1,#0
   611              1            SKF1    CY                              ;CY = 0 (CODE JUDGE COUNTER >= 4.0ms) ?
 +   1 0151 1F7F4   1            SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
   612 0152 0C1E0                BR      REPEAT1
   613 0153 1C001                CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER, REPEAT CODE CONTROL
   614 0154 110DE                SUB     CNT2,#0EH
   615 0155 130C0                SUBC    CNT1,#0
   616              1            SKF1    CY                              ;CY = 0 (CODE JUDGE COUNTER >= 2.6ms) ?
 +   1 0156 1F7F4   1            SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
   617 0157 0C1AC                BR      ERROR48
   618                   EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-023

PROG =

SOURCE = ¥ASMFILE¥10377ﾘ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   619                      ;*****************************************************************************
   620                      ;* カスタム・コード、データ・コード受信処理                                    *
   621                      ;*      カスタム・コードまたはデータ・コードを受信します。                     *
   622                      ;*      カスタム・コードまたはデータ・コードの上位４ビットはDATA1 に、         *
   623                      ;*      下位４ビットはDATA2 に格納します。                                     *
   624                      ;*      なお、INCNT はカスタム・コードまたはデータ・コードの受信ビット        *
   625                      ;*      数を示します。また、CC_Fの状態によりカスタム・コードを受信中か        *
   626                      ;*      データ・コード受信中かを区別します。                                   *
   627                      ;*            CD_F=0        カスタム・コード受信中                             *
   628                      ;*            CD_F=1        データ・コード受信中                               *
   629                      ;*****************************************************************************
   630                L_JOB8:
   631 0158 1C0D0           CALL    WAIT8
   632 0159 1C005           CALL    TIME_CTL                        ;7 SEGMENT LED, TIMER COUNTER CONTROL
   633 015A 1D09F           MOV     INCNT,#0FH                      ;INCNT ← 0FH (CODE RECEIVE COUNTER INITIAL)
   634              1       CLR3    ROK_F,CD_F,UP_F                 ;ROK_F ← 0 (REPEAT CODE RECEIVE STOP)
 +   1 015B 140F1   1       AND     .MF.ROK_F SHR 4,#.DF.(NOT (ROK_F OR CD_F OR UP_F) AND 0FH)
   635                                                              ;CD_F ← 0 (CUSTOM CODE RECIEVE)
   636                                                              ;UP_F ← 0 (BIT STORE FLAG INITIAL)
   637 015C 1D0A0           MOV     DATA1,#0                        ;DATA1 ← 0 (CODE STORE MEMORY INITIAL)
   638 015D 1D0B0           MOV     DATA2,#0                        ;DATA2 ← 0 (CODE STORE MEMORY INITIAL)
   639 015E 1D0D9           MOV     CNT2,#9                         ;CNT2 ← 9 (CODE JUDGE COUNTER INITIAL)
   640                IN_DA0:
   641 015F 1D0CE           MOV     CNT1,#0EH                       ;CNT1 ← 0EH (CODE JUDGE COUNTER INITIAL)
   642                IN_DA1:
   643              1       SKT1    P0B0                            ;P0B0 = High LEVEL (UP EDGE DETECT) ?
 +   1 0160 1E711   1       SKT     .MF.P0B0 SHR 4,#.DF.P0B0 AND 0FH
   644 0161 0C168           BR      IN_DA2
   645 0162 1C001           CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER CONTROL
   646 0163 1C0C9           CALL    CNT_UP                          ;CODE JUDGE COUNTER UP
   647 0164 0C166           BR      CHA確認
   648 0165 0C1B5           BR      ERROR2
   649                CHA確認:
   650              1       SKF1    P0B0                            ;P0B0 = Low LEVEL (CHATARING) ?
 +   1 0166 1F711   1       SKF     .MF.P0B0 SHR 4,#.DF.P0B0 AND 0FH
   651 0167 0C16C           BR      IN_DA3
   652                IN_DA2:
   653 0168 1C001           CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER CONTROL
   654 0169 1C0C9           CALL    CNT_UP                          ;CODE JUDGE COUNTER UP
   655 016A 0C160           BR      IN_DA1
   656 016B 0C1B5           BR      ERROR2
   657                IN_DA3:
   658 016C 1C001           CALL    TIME_CTL4                       ;7 SEGMENT LED, TIMER COUNTER CONTROL
   659 016D 1C0C9           CALL    CNT_UP                          ;CODE JUDGE COUNTER UP
   660 016E 0C170           BR      IN_DA4
   661 016F 0C1B5           BR      ERROR2
   662                IN_DA4:
   663              1       SKF1    P0B0                            ;P0B0 = Low LEVEL (DOWN EDGE DETECT) ?
 +   1 0170 1F711   1       SKF     .MF.P0B0 SHR 4,#.DF.P0B0 AND 0FH
   664 0171 0C16C           BR      IN_DA3
   665                      ;*****************************************************************************
   666                      ;* 受信コード判定                                                            *
   667                      ;*      カスタム・コード（またはデータ・コード）の立ち下がりから              *
   668                      ;*      次のカスタム・コード（またはデータ・コード）の立ち下がり              *
   669                      ;*      までの時間により以下のように受信データを判定します。                  *
```

PROG =

SOURCE = ¥ASMFILE¥10377ﾞﾘ.ASM

```
E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   670                     ;*                1ms未満                異常検出                        *
   671                     ;*                1ms以上1.4ms未満       受信データ "0"                  *
   672                     ;*                1.4ms以上2.0ms未満     異常検出                        *
   673                     ;*                2.0ms以上2.6ms未満     受信データ "1"                  *
   674                     ;*                2.6ms以上              異常検出                        *
   675                     ;*****************************************************************************
   676 0172 1C001          CALL    TIME_CTL4                                ;7 SEGMENT LED, TIMER COUNTER CONTROL
   677 0173 110D0          SUB     CNT2,#0
   678 0174 130CF          SUBC    CNT1,#0FH
   679              1      SKF1    CY                                       ;CY = 0 (CODE JUDGE COUNTER >= 1.0ms) ?
 +   1 0175 1F7F4  1      SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
   680 0176 0C1AC          BR      ERROR48
   681 0177 110D3          SUB     CNT2,#3
   682 0178 130C0          SUBC    CNT1,#0
   683              1      SKF1    CY                                       ;CY = 0 (CODE JUDGE COUNTER >= 1.4ms) ?
 +   1 0179 1F7F4  1      SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
   684 017A 0C18A          BR      CLR1_JOB
   685 017B 1C001          CALL    TIME_CTL4                                ;7 SEGMENT LED, TIMER COUNTER CONTROL
   686 017C 110D7          SUB     CNT2,#7
   687 017D 130C0          SUBC    CNT1,#0
   688              1      SKF1    CY                                       ;CY = 0 (CODE JUDGE COUNTER >= 2.0ms) ?
 +   1 017E 1F7F4  1      SKF     .MF.CY SHR 4,#.DF.CY AND 0FH
   689 017F 0C1AC          BR      ERROR48
   690                  SET1_JOB:
   691              1      SET1    UP_F                                     ;UP_F ← 1 (RECEIVE CODE "1")
 +   1 0180 160F2  1      OR      .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   692 0181 10091          ADD     INCNT,#1                                 ;INCNT ← INCNT + 1 (CODE RECEIVE COUNTER UP)
   693 0182 1B098          SKLT    INCNT,#8                                 ;INCNT < 8 ?
   694 0183 0C192          BR      COMP2
   695 0184 1C001          CALL    TIME_CTL4                                ;7 SEGMENT LED, TIMER COUNTER CONTROL
   696 0185 1C0D7          CALL    SET1_1                                   ;STORE "1"
   697 0186 1C005          CALL    TIME_CTL                                 ;7 SEGMENT LED, TIMER COUNTER CONTROL
   698 0187 1D0DA          MOV     CNT2,#0AH                                ;CNT2 ← 0AH (CODE JUDGE COUNTER INITIAL)
   699 0188 1C0D5          CALL    WAIT3
   700 0189 0C15F          BR      IN_DA0
   701                  CLR1_JOB:
   702              1      CLR1    UP_F                                     ;UP_F ← 0 (RECEIVE CODE "0")
 +   1 018A 140FD  1      AND     .MF.UP_F SHR 4,#.DF.(NOT UP_F AND 0FH)
   703 018B 10091          ADD     INCNT,#1                                 ;INCNT ← INCNT + 1 (CODE RECEIVE COUNTER UP)
   704 018C 1C003          CALL    TIME_CTL2                                ;7 SEGMENT LED, TIMER COUNTER CONTROL
   705 018D 1B098          SKLT    INCNT,#8                                 ;INCNT < 8 ?
   706 018E 0C191          BR      COMP1
   707 018F 1D0D8          MOV     CNT2,#8                                  ;CNT2 ← 8 (CODE JUDGE COUNTER INITIAL)
   708 0190 0C1A4          BR      COMP_END
   709                  COMP1:
   710 0191 1C0D2          CALL    WAIT6
   711                  COMP2:
   712 0192 1D0DA          MOV     CNT2,#0AH                                ;CNT2 ← 0AH (CODE JUDGE COUNTER INITIAL)
   713 0193 1C002          CALL    TIME_CTL3                                ;7 SEGMENT LED, TIMER COUNTER CONTROL
   714 0194 1C0EB          CALL    DATA_COMP                                ;RECEIVE CODE COMPARE
   715 0195 1C005          CALL    TIME_CTL                                 ;7 SEGMENT LED, TIMER COUNTER CONTROL
   716              1      SKF1    UP_F                                     ;UP_F = 0 (NOT SAME) ?
 +   1 0196 1F0F2  1      SKF     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   717 0197 0C1B3          BR      ERROR46
   718 0198 0909F          SKE     INCNT,#0FH                               ;INCNT = 0FH (RECEIVE CODE COUNTER = 16 BIT) ?
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-025

PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾟﾛｸﾞﾗﾑ.ASM

E STNO LOC. OBJ.    M  I SOURCE STATEMENT
   719 0199 0C1A4                 BR      COMP_END
   720                1           NOT1    CD_F                          ;CD_F CHANGE
 +   1 019A 150F8     1           XOR     .MF.CD_F SHR 4,#.DF.CD_F AND 0FH
   721                1           SKT1    CD_F                          ;CD_F = 1 (DATA CODE RECEIVE) ?
 +   1 019B 1E0F8     1           SKT     .MF.CD_F SHR 4,#.DF.CD_F AND 0FH
   722 019C 0C1A5                 BR      DATA_CODE1
   723 019D 0B0A4                 SKNE    DATA1,#4                      ;DATA1 = 4 (CUSTOM CODE High 4BIT = 4) ?
   724 019E 090B0                 SKE     DATA2,#0                      ;DATA2 = 0 (CUSTOM CODE Low 4BIT = 4) ?
   725                        ERROR3:
   726 019F 0C1D6                 BR      ERROR4
   727 01A0 1D0A0                 MOV     DATA1,#0                      ;DATA1 ← 0 (CODE STORE MEMORY INITIAL)
   728 01A1 1C004                 CALL    TIME_CTL1                     ;7 SEGMENT LED, TIMER COUNTER CONTROL
   729 01A2 1C0D5                 CALL    ¥AIT3
   730 01A3 1B0DB                 MOV     CNT2,#0BH                     ;CNT2 ← 0BH (CODE JUDGE COUNTER INITIAL)
   731                        COMP_END:
   732 01A4 0C15F                 BR      IN_DA0
   733                        EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-026

PROG =

SOURCE = ¥ASMFILE¥10377ﾀﾞ.ASM

E STNO LOC. OBJ.   M  I SOURCE STATEMENT
   734                      ;*****************************************************************************
   735                      ;* DATA_CODE1                                                                *
   736                      ;*       受信したカスタム・コード、データ・コードにより以下のように         *
   737                      ;*       処理を行います。                                                    *
   738                      ;*                                                                           *
   739                      ;*       カスタム・コード                                                    *
   740                      ;*             04H      :データ・コードの受信を行います。                    *
   741                      ;*             04H以外  :リーダ・コード検出処理を行います。                  *
   742                      ;*       データ・コード                                                      *
   743                      ;*             00H      :モードを反転します。                                *
   744                      ;*             01H      :タイマ・カウンタの値をインクリメントします。        *
   745                      ;*                       ただし、タイマ設定モードだけ有効です。              *
   746                      ;*             02H      :タイマ・カウンタの値をデクリメントします。          *
   747                      ;*                       ただし、タイマ設定モードだけ有効です。              *
   748                      ;*****************************************************************************
   749                DATA_CODE1:
   750 01A5 090B0           SKE      DATA2,#0                                ;DATA2 = 0 (DATA CODE Low 4BIT = 0) ?
   751 01A6 0C19F           BR       ERROR3
   752 01A7 1C002           CALL     TIME_CTL3                               ;7 SEGMENT LED, TIMER COUNTER CONTROL
   753 01A8 090A0           SKE      DATA1,#0                                ;DATA1 = 0 (DATA CODE High 4BIT = 0) ?
   754 01A9 0C1B6           BR       DATA_CODE2
   755                   1  NOT1     MODE_F                                  ;MODE_F CHANGE
 +   1 01AA 15081    1     XOR      .MF.MODE_F SHR 4,#.DF.MODE_F AND 0FH
   756                   1  SKF1     MODE_F                                  ;MODE_F = 0 (タイマ設定モード) ?
 +   1 01AB 1F081    1     SKF      .MF.MODE_F SHR 4,#.DF.MODE_F AND 0FH
   757                DATA_CODE48:
   758 01AC 0C1D5           BR       ERROR49
   759                   1  SKT1     SET_F                                   ;SET_F = 1 (TIMER COUNTER SETTING) ?
 +   1 01AD 1E0F1    1     SKT      .MF.SET_F SHR 4,#.DF.SET_F AND 0FH
   760 01AE 0C1B4           BR       MODE_TURN
   761                   1  CLR1     SET_F                                   ;SET_F ← 0 (TIMER COUNTER NO SETTING)
 +   1 01AF 140FE    1     AND      .MF.SET_F SHR 4,#.DF.(NOT SET_F AND 0FH)
   762                   1  SET1     COUNT_F                                 ;COUNT_F ← 1 (COUNTING)
 +   1 01B0 16082    1     OR       .MF.COUNT_F SHR 4,#.DF.COUNT_F AND 0FH
   763                   1  SET1     POD3                                    ;POD3 ← High LEVEL OUTPUT SET (LED ON)
 +   1 01B1 16738    1     OR       .MF.POD3 SHR 4,#.DF.POD3 AND 0FH
   764 01B2 1C003           CALL     TIME_CTL2                               ;7 SEGMENT LED, TIMER COUNTER CONTROL
   765                DATA_CODE46:
   766 01B3 0C1B7           BR       ERROR47
   767                      ;
   768                MODE_TURN:
   769 01B4 1D711           MOV      P0B,#1
   770                ERROR2:
   771 01B5 0C19F           BR       ERROR3
   772                      ;
   773                DATA_CODE2:
   774                   1  SKT1     MODE_F                                  ;MODE_F =1 (タイマ動作モード) ?
 +   1 01B6 1E081    1     SKT      .MF.MODE_F SHR 4,#.DF.MODE_F AND 0FH
   775                ERROR47:
   776 01B7 0C1AC           BR       ERROR48
   777                      ;
   778 01B8 090A1           SKE      DATA1,#1                                ;DATA1 = 1 (DATA CODE High 4BIT = 1) ?
   779 01B9 0B0A0           SKNE     DATA1,#0                                ;DATA1 ≠ 0 (DATA CODE High 4BIT ≠ 0) ?
   780 01BA 0C1BC           BR       DATA_CODE3
   781 01BB 0C1B5           BR       ERROR2
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-027

PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾟﾛｸﾞﾗﾑ.ASM

E STNO LOC. OBJ.    M   I SOURCE STATEMENT
   782                      ;
   783                  DATA_CODE3:
   784                  1           SET1    ROK_F                                   ;ROK_F ← 0 (REPEAT CODE RECEIVE OK)
 +   1 01BC 160F4   1               OR      .MF.ROK_F SHR 4,#.DF.ROK_F AND 0FH
   785 01BD 0B0A1                   SKNE    DATA1,#1                                ;DATA1 ≠ 1 (DATA CODE High 4BIT ≠ 1) ?
   786                  1           SET1    UP_F                                    ;UP_F ← 1 (TIMER COUNTER UP SET)
 +   1 01BE 160F2   1               OR      .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   787                  1           CLR1    COUNT_F                                 ;COUNT_F ← 0 (NOV COUNTING)
 +   1 01BF 1408D   1               AND     .MF.COUNT_F SHR 4,#.DF.(NOT COUNT_F AND 0FH)
   788                  1           CLR1    POD3                                    ;POD3 ← Low LEVEL OUTPUT SET (LED OFF)
 +   1 01C0 14737   1               AND     .MF.POD3 SHR 4,#.DF.(NOT POD3 AND 0FH)
   789 01C1 1C005                   CALL    TIME_CTL                                ;7 SEGMENT LED, TIMER COUNTER CONTROL
   790 01C2 074F0                   NOP
   791 01C3 1D098                   MOV     INCNT,#8                                ;INCNT ← 8 (REPEAT CODE COUNTER INITIAL)
   792 01C4 140E1                   AND     FLAG1,#0001B                            ;REPEAT CODE COUNTER1 INITIAL
   793                  1           SKF1    SET_F                                   ;SET_F = 0 (TIMER COUNTER NO SETTING) ?
 +   1 01C5 1F0F1   1               SKF     .MF.SET_F SHR 4,#.DF.SET_F AND 0FH
   794 01C6 0C1EA                   BR      CNT_UPDN1
   795                  1           SET1    SET_F                                   ;SET_F ← 1 (TIMER COUNTER SETTEING)
 +   1 01C7 160F1   1               OR      .MF.SET_F SHR 4,#.DF.SET_F AND 0FH
   796 01C8 1D0A8                   MOV     DATA1,#8                                ;DATA1 ← 8 (108ms COUNTER INITIAL)
   797 01C9 1D0B2                   MOV     DATA2,#2                                ;DATA2 ← 2 (108ms COUNTER INITIAL)
   798 01CA 1C001                   CALL    TIME_CTL4                               ;7 SEGMENT LED, TIMER COUNTER CONTROL
   799 01CB 1D085                   MOV     MUSCNT3,#5                              ;MUSCNT3 ← 5 (TIMER COUNTER 100 MINUTES SET)
   800 01CC 1D078                   MOV     MUSCNT2,#8                              ;MUSCNT2 ← 8 (TIMER COUNTER 100 MINUTES SET)
   801 01CD 1D06F                   MOV     MUSCNT1,#0FH                            ;MUSCNT1 ← 0FH (TIMER COUNTER 100 MINUTES SET)
   802 01CE 1D050                   MOV     SCNT2,#0                                ;SCNT2 ← 0 (TIMER COUNTER 100 MINUTES SET)
   803 01CF 1D046                   MOV     SCNT1,#6                                ;SCNT1 ← 6 (TIMER COUNTER 100 MINUTES SET)
   804 01D0 1D030                   MOV     MCNT2,#0                                ;MCNT2 ← 0 (TIMER COUNTER 100 MINUTES SET)
   805 01D1 1D020                   MOV     MCNT1,#0                                ;MCNT1 ← 0 (TIMER COUNTER 100 MINUTES SET)
   806                  1           SKF1    UP_F                                    ;UP_F = 0 (TIMER COUNTER DOWN SET) ?
 +   1 01D2 1F0F2   1               SKF     .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   807 01D3 1D030                   MOV     MCNT2,#0                                ;MCNT2 ← 1 (TIMER COUNTER 1 MINUTES SET)
   808 01D4 0C1D7                   BR      TIME5
   809                              ;
   810                  EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-028

PROG =

SOURCE = ¥ASMFILE¥10377ﾌﾟﾛｸﾞﾗﾑ.ASM

E STNO LOC. OBJ.    M   I SOURCE STATEMENT
   811                    ERROR49:
   812 01D5 1C0D3                 CALL     WAIT5
   813                    ERROR4:
   814                1           CLR3     ROK_F,CD_F,UP_F                              ;ROK_F ← 0 (REPEAT CODE RECEIVE STOP)
 +   1 01D6 140F1     1           AND      .MF.ROK_F SHR 4,#.DF.(NOT (ROK_F OR CD_F OR UP_F) AND 0FH)
   815                                                                                  ;CD_F ← 0 (CUSTOM CODE RECIEVE)
   816                                                                                  ;UP_F ← 0 (BIT STORE FLAG INITIAL)
   817                    TIME5:
   818 01D7 074F0                 NOP
   819                    TIME6:
   820 01D8 1C005                 CALL     TIME_CTL                                    ;7 SEGMENT LED, TIMER COUNTER CONTROL
   821 01D9 1C0D4                 CALL     WAIT4
   822 01DA 0B019                 SKNE     CNT100,#9                                   ;CNT100 = 9 (900 us COUNT) ?
   823 01DB 0C1DE                 BR       L_JOB9
   824 01DC 1C0D3                 CALL     WAIT5
   825 01DD 0C1D8                 BR       TIME6
   826                            ;
   827                    L_JOB9:
   828 01DE 1D010                 MOV      CNT100,#0                                   ;CNT100 ← 0 (Low LEVEL COUNTER INITIAL)
   829 01DF 0C123                 BR       L_JOB4
   830                            ;
   831                    EJECT
```

```
AS17K V1.04 V3 <<   D17103 ASSEMBLE LIST >>   14:53:35 01/17/90 PAGE 01-029

PROG =

SOURCE = ¥ASMFILE¥10377ﾘ.ASM

E STNO LOC. OBJ.    M   I SOURCE STATEMENT
   832                      REPEAT1:
   833                 1            SKT1     ROK_F                                  ;ROK_F = 1 (REPEAT CODE RECEIVE OK) ?
 +   1 01E0 1E0F4   1               SKT      .MF.ROK_F SHR 4,#.DF.ROK_F AND 0FH
   834 01E1 0C1D6                   BR       ERROR4
   835 01E2 1C003                   CALL     TIME_CTL2                              ;7 SEGMENT LED, TIMER COUNTER CONTROL
   836 01E3 100E2                   ADD      FLAG1,#2                               ;FLAG1 ←  FLAG1 + 2 (REPEAT CODE COUNTER1 UP)
   837 01E4 10091                   ADD      INCNT,#1                               ;INCNT ←  INCNT + 1 (REPEAT CODE COUNTER UP)
   838                 1            SKF1     CY                                     ;CY = 1
 +   1 01E5 1F7F4   1               SKF      .MF.CY SHR 4,#.DF.CY AND 0FH
   839 01E6 0C1E9                   BR       CNT_UPDN
   840 01E7 1C0D2                   CALL     WAIT6
   841 01E8 0C1D8                   BR       TIME6
   842                              ;
   843                      CNT_UPDN:
   844 01E9 1D09E                   MOV      INCNT,#0EH                             ;INCNT ←  0EH (REPEAT CODE COUNTER INITIAL)
   845                      CNT_UPDN1:
   846                 1            SET1     BCD                                    ;BCD←1 (10進数演算モード)
 +   1 01EA 167E1   1               OR       .MF.BCD SHR 4,#.DF.BCD AND 0FH
   847                 1            SKT1     UP_F                                   ;UP_F = 1 (TIMER COUNTER UP SET) ?
 +   1 01EB 1E0F2   1               SKT      .MF.UP_F SHR 4,#.DF.UP_F AND 0FH
   848 01EC 0C1F1                   BR       DOWN1
   849 01ED 10031                   ADD      MCNT2,#1                               ;MCNT2 ←  MCNT2 + 1 (TIMER COUNTER 1 MINUTES UP)
   850 01EE 12020                   ADDC     MCNT1,#0                               ;MCNT1 ←  MCNT1 + CY (TIMER COUNTER 1 MINUTES UP)
   851                 1            CLR1     BCD                                    ;BCD←0 (16進数演算モード)
 +   1 01EF 147EE   1               AND      .MF.BCD SHR 4,#.DF.(NOT BCD AND 0FH)
   852 01F0 0C1D8                   BR       TIME6
   853                              ;
   854                      DOWN1:
   855 01F1 11031                   SUB      MCNT2,#1                               ;MCNT2 ←  MCNT2 - 1 (TIMER COUNTER 1 MINUTES DOWN)
   856 01F2 13020                   SUBC     MCNT1,#0                               ;MCNT1 ←  MCNT1 - CY (TIMER COUNTER 1 MINUTES DOWN
   857                 1            CLR1     BCD                                    ;BCD←0 (16進数演算モード)
 +   1 01F3 147EE   1               AND      .MF.BCD SHR 4,#.DF.(NOT BCD AND 0FH)
   858 01F4 0C1D8                   BR       TIME6
   859                              ;
   860                              OPTION
   861                 1            OPTPOB           POBPLUP,POBPLUP,OPEN
 +  11       0006   1
   862                 1            OPTRES           RESPLUP
 +   5       0001   1
   863                              ENDOP
   864                              END


TOTAL ERRORS   = 0
TOTAL WARNINGS = 4

END OF LIST
```

```
AS17K V1.04 V3 << D17103 XREF LIST >>        14:53:49 01/17/90 PAGE 01-001

PROG =

SOURCE = ¥ASMFILE¥10377'ｿ.ASM

SYMBOL            TYPE A VALUE   /REF(#DEF)

ALL_ON            LAB  L      5B /   188     ,# 192
BAR_LOOP          LAB  L      C5 /# 356     ,   359
BAR_SET           LAB  L      C2 /   222     ,   347     ,# 352
C_LOOP            LAB  L      48 /# 152     ,   155
CD_F              FLG  L 0.0F.3 /#  28     ,   634     ,   634-1   ,   720     ,   720-1   ,   720-1   ,   721     ,   721-1   ,   721-1
                                        814     ,   814-1
CHA確認           LAB  L     186 /   647     ,# 649
CLR1_JOB          LAB  L     18A /   684     ,# 701
CNT1              MEM  L   0.0C /#  21     ,   369     ,   594     ,   606     ,   610     ,   615     ,   641     ,   678     ,   682
                                        687
CNT100            MEM  L    0.01 /#   7     ,    60     ,    61     ,    63     ,    64     ,    65     ,    68     ,   522     ,   555
                                        556     ,   557     ,   582     ,   822     ,   828
CNT2              MEM  L   0.0D /#  22     ,   368     ,   593     ,   605     ,   609     ,   614     ,   639     ,   677     ,   681
                                        686     ,   698     ,   707     ,   712     ,   730
CNT_UP            LAB  L      C9 /# 367     ,   600     ,   646     ,   654     ,   659
CNT_UPDN          LAB  L     1E9 /   839     ,# 843
CNT_UPDN1         LAB  L     1EA /   794     ,# 845
CNTX              LAB  L      2E /    95     ,# 112
CNTXX             LAB  L      29 /    97     ,# 104
CNTXXX            LAB  L      32 /   106     ,# 117
COMON_OFF         LAB  L      67 /   179     ,# 216
COMON_ON          LAB  L      B0 /   272     ,   276     ,   284     ,   288     ,# 320
COMP1             LAB  L     191 /   706     ,# 709
COMP2             LAB  L     192 /   694     ,# 711
COMP_END          LAB  L     1A4 /   708     ,   719     ,# 731
COUNT1            LAB  L       F /#  75     ,   538
COUNT2            LAB  L      45 /   130     ,# 146
COUNT3            LAB  L      46 /    69     ,# 148
COUNT4            LAB  L      47 /   135     ,   144     ,# 150
COUNT_F           FLG  L 0.08.1 /#  15     ,    67     ,    67-1   ,    67-1   ,    98     ,    98-1   ,   177     ,   177-1   ,   177-1
                                        219     ,   219-1   ,   219-1   ,   523     ,   523-1   ,   523-1   ,   536     ,   536-1   ,   536-1
                                        762     ,   762-1   ,   762-1   ,   787     ,   787-1   ,   787-1
DATA1             MEM  L   0.0A /#  19     ,   132     ,   136     ,   417     ,   419     ,   421     ,   423     ,   465     ,   470
                                        478     ,   483     ,   637     ,   723     ,   727     ,   753     ,   778     ,   779     ,   785
                                        796
DATA2             MEM  L   0.0B /#  20     ,   133     ,   137     ,   427     ,   429     ,   431     ,   433     ,   493     ,   498
                                        506     ,   511     ,   638     ,   724     ,   750     ,   797
DATA_CODE1        LAB  L     1A5 /   722     ,# 749
DATA_CODE2        LAB  L     1B6 /   754     ,# 773
DATA_CODE3        LAB  L     1BC /   780     ,# 783
DATA_COMP         LAB  L      EB /# 458     ,   714
DATA_COMP1        LAB  L      F3 /   464     ,# 467
DATA_COMP10       LAB  L     114 /   510     ,# 513
DATA_COMP11       LAB  L     101 /   460     ,# 488
DATA_COMP2        LAB  L      F7 /   469     ,# 472
DATA_COMP3        LAB  L      F8 /   462     ,# 475
DATA_COMP4        LAB  L      FC /   477     ,# 480
DATA_COMP5        LAB  L     100 /   482     ,# 485
DATA_COMP6        LAB  L     10C /   490     ,# 503
DATA_COMP7        LAB  L     107 /   492     ,# 495
DATA_COMP8        LAB  L     10B /   497     ,# 500
DATA_COMP9        LAB  L     110 /   505     ,# 508
DATA_SET          LAB  L      84 /   241     ,   251     ,# 261
DATA_SET0         LAB  L      9C /   263     ,# 290
```

```
AS17K V1.04 V3 << D17103 XREF LIST >>        14:53:49 01/17/90 PAGE 01-002

PROG =

SOURCE = ¥ASMFILE¥10377'ﾌ.ASM

SYMBOL            TYPE A VALUE   /REF(#DEF)

DATA_SET00        LAB  L      A6 /  255     ,  292     ,# 302
DATA_SET1         LAB  L      AB /  304     ,# 308
DATA_SET11        LAB  L      83 /  245     ,# 259
DATA_SET3         LAB  L      A3 /  294     ,# 298
DATA_SET4         LAB  L      92 /  265     ,# 277
DATA_SET5         LAB  L      99 /  281     ,# 285
DATA_SET8         LAB  L      8F /  269     ,# 273
DISP_LOOP         LAB  L      B9 /# 334     ,  338
DISP_OFF          LAB  L      B5 /  242     ,# 328
DISP_OFF1         LAB  L      B7 /  254     ,# 331
DOWN1             LAB  L     1F1 /  848     ,# 854
ERROR2            LAB  L     1B5 /  602     ,  648     ,  656     ,  661     ,# 770     ,  781
ERROR3            LAB  L     19F /# 725     ,  751     ,  771
ERROR4            LAB  L     1D6 /  726     ,# 813     ,  834
ERROR46           LAB  L     1B3 /  717     ,# 765
ERROR47           LAB  L     1B7 /  766     ,# 775
ERROR48           LAB  L     1AC /  608     ,  617     ,  680     ,  689     ,# 757     ,  776
ERROR49           LAB  L     1D5 /  758     ,# 811
FLAG1             MEM  L    0.0E /#  24     ,  138     ,  139     ,  141     ,  792     ,  836
FLAG2             MEM  L    0.0F /#  27
FLASH_JOB2        LAB  L      5D /  190     ,# 195
FLASH_JOB3        LAB  L      62 /  197     ,# 203
FLASH_LOOP1       LAB  L      5F /  186     ,# 199     ,  202
FLASH_LOOP2       LAB  L      63 /# 205     ,  208
IN_DA0            LAB  L     15F /# 640     ,  700     ,  732
IN_DA1            LAB  L     160 /# 642     ,  655
IN_DA2            LAB  L     168 /  644     ,# 652
IN_DA3            LAB  L     16C /  651     ,# 657     ,  664
IN_DA4            LAB  L     170 /  660     ,# 662
INCNT             MEM  L    0.09 /#  18     ,  414     ,  416     ,  418     ,  420     ,  422     ,  426     ,  428     ,  430
                                    432     ,  459     ,  461     ,  463     ,  468     ,  476     ,  481     ,  489     ,  491
                                    496     ,  504     ,  509     ,  633     ,  692     ,  693     ,  703     ,  705     ,  718
                                    791     ,  837     ,  844
L_CHECK           LAB  L      129 /  550     ,# 553
L_JOB1            LAB  L      119 /# 533     ,  569
L_JOB2            LAB  L      11F /  537     ,# 540
L_JOB3            LAB  L      122 /  539     ,# 544
L_JOB4            LAB  L      123 /# 546     ,  829
L_JOB5            LAB  L      137 /  558     ,# 581
L_JOB6            LAB  L      138 /# 583     ,  587     ,  591
L_JOB7            LAB  L      144 /# 596     ,  601
L_JOB8            LAB  L      158 /# 630
L_JOB9            LAB  L      1DE /  823     ,# 827
L_LOOP            LAB  L      12F /# 561     ,  567
MAIN              LAB  L      115 /  39      ,# 521
MAKE_JOB1         LAB  L       AD /# 311     ,  360
MAKE_JOB2         LAB  L       AE /  297     ,  301     ,  307     ,# 313
MCNT1             MEM  L    0.02 /#   8     ,   87     ,   93     ,  183     ,  196     ,  234     ,  249     ,  805     ,  850
                                    856
MCNT2             MEM  L    0.03 /#   9     ,   86     ,   96     ,  105     ,  108     ,  184     ,  235     ,  253     ,  257
                                    804     ,  807     ,  849     ,  855
MCNT2_SET         LAB  L      81 /  248     ,# 256
MCNT_SET          LAB  L      78 /  236     ,# 246     ,  350
MODE_F            FLG  L 0.08.0 /#  16     ,   98     ,   98-1   ,   98-1   ,  178     ,  178-1   ,  178-1   ,  221     ,  221-1
```

```
AS17K V1.04 V3 << D17103 XREF LIST >>        14:53:49 01/17/90 PAGE 01-003

PROG =

SOURCE = ¥ASMFILE¥10377'ｿ.ASM

SYMBOL          TYPE A VALUE  /REF(#DEF)

                                221-1 ,  523   ,  523-1 ,  755   ,  755-1 ,  755-1 ,  758   ,  756-1 ,  756-1
                                774   ,  774-1 ,  774-1
MODE_TURN       LAB  L    1B4 /  760   ,# 768
MUSCNT1         MEM  L   0.06 /#  12   ,   78   ,  801
MUSCNT2         MEM  L   0.07 /#  13   ,   77   ,   82   ,  800
MUSCNT3         MEM  L   0.08 /#  14   ,   76   ,   80   ,  799
NO_COUNT        LAB  L     BD /  220   ,# 345
OPTPOB          MAC  L ****** /  861
OPTRES          MAC  L ****** /  862
POB             MEM  L   0.71 /#  33   ,  101   ,  189   ,  193   ,  217   ,  322   ,  324   ,  769
POC             MEM  L   0.72 /#  34   ,  180   ,  266   ,  270   ,  278   ,  282   ,  295   ,  299   ,  305
                                309   ,  353
POD             MEM  L   0.73 /#  35   ,  181   ,  218   ,  267   ,  271   ,  275   ,  279   ,  283   ,  287
                                296   ,  300   ,  306   ,  310   ,  354
RECNT           LAB  L     35 /   66   ,# 128   ,  534
REPEAT1         LAB  L    1E0 /  612   ,# 832
RET_JOB         LAB  L     2D /  102   ,# 109
ROK_F           FLG  L 0.0F.2 /#  29   ,  129   ,  129-1 ,  129-1 ,  140   ,  140-1 ,  140-1 ,  526   ,  526-1
                                526-1 ,  634   ,  634-1 ,  634-1 ,  784   ,  784-1 ,  784-1 ,  814   ,  814-1
                                814-1 ,  833   ,  833-1 ,  833-1
SCNT1           MEM  L   0.04 /#  10   ,   85   ,   92   ,  107   ,  118   ,  119   ,  239   ,  803
SCNT2           MEM  L   0.05 /#  11   ,   84   ,   91   ,  244   ,  802
SCNT2_SET       LAB  L     76 /  238   ,# 243
SEGMENT_JOB     LAB  L     4C /   62   ,# 175   ,  548
SEL_F           FLG  L 0.0E.0 /#  25   ,  176   ,  176-1 ,  176-1 ,  237   ,  237-1 ,  237-1 ,  247   ,  247-1
                                247-1 ,  321   ,  321-1 ,  321-1 ,  323   ,  323-1 ,  323-1 ,  525   ,  525-1
                                525-1 ,  549   ,  549-1 ,  549-1
SET1_1          LAB  L     D7 /# 413   ,  696
SET1_2          LAB  L     E2 /  415   ,# 425
SET1_JOB        LAB  L    180 /# 690
SET_F           FLG  L 0.0F.0 /#  31   ,  346   ,  346-1 ,  346-1 ,  526   ,  526-1 ,  759   ,  759-1 ,  759-1
                                761   ,  761-1 ,  761-1 ,  793   ,  793-1 ,  793-1 ,  795   ,  795-1 ,  795-1
TIME5           LAB  L    1D7 /  808   ,# 817
TIME6           LAB  L    1D8 /# 819   ,  825   ,  841   ,  852   ,  858
TIME_CTL        LAB  L      5 /#  59   ,  632   ,  697   ,  715   ,  789   ,  820
TIME_CTL1       LAB  L      4 /#  57   ,  728
TIME_CTL2       LAB  L      3 /#  55   ,  704   ,  764   ,  835
TIME_CTL3       LAB  L      2 /#  53   ,  713   ,  752
TIME_CTL4       LAB  L      1 /#  51   ,  584   ,  588   ,  592   ,  599   ,  604   ,  613   ,  645   ,  653
                                658   ,  676   ,  685   ,  695   ,  798
TIME_JUDGE1     LAB  L    14A /  598   ,# 603
UP_F            FLG  L 0.0F.1 /#  30   ,  466   ,  466-1 ,  466-1 ,  471   ,  471-1 ,  471-1 ,  479   ,  479-1
                                479-1 ,  484   ,  484-1 ,  484-1 ,  494   ,  494-1 ,  494-1 ,  499   ,  499-1
                                499-1 ,  507   ,  507-1 ,  507-1 ,  512   ,  512-1 ,  512-1 ,  634   ,  634-1
                                691   ,  691-1 ,  691-1 ,  702   ,  702-1 ,  702-1 ,  716   ,  716-1 ,  716-1
                                786   ,  786-1 ,  786-1 ,  806   ,  806-1 ,  806-1 ,  814   ,  814-1 ,  847
                                847-1 ,  847-1
WAIT10          LAB  L     CE /# 378   ,  541   ,  542   ,  562   ,  563
WAIT2           LAB  L     D6 /# 394
WAIT3           LAB  L     D5 /# 392   ,  699   ,  729
WAIT4           LAB  L     D4 /# 390   ,  551   ,  595   ,  821
WAIT5           LAB  L     D3 /# 388   ,  564   ,  812   ,  824
WAIT6           LAB  L     D2 /# 386   ,  545   ,  585   ,  589   ,  710   ,  840
WAIT7           LAB  L     D1 /# 384   ,  547
```

```
AS17K V1.04 V3 << D17103 XREF LIST >>          14:53:49 01/17/90 PAGE 01-004

PROG =

SOURCE = ¥ASMFILE¥10377ｶｲﾛ.ASM

SYMBOL              TYPE A VALUE   /REF(#DEF)

WAIT8               LAB  L      D0 /# 382     ,  535     ,  543     ,  631
WAIT9               LAB  L      CF /# 380     ,  568
WAIT_375            LAB  L     12E /  552     ,# 559
WORK                MEM  L    0.00 /#   6     ,  131     ,  142     ,  149     ,  153     ,  182     ,  200     ,  204     ,  206
                                      239     ,  240     ,  244     ,  249     ,  250     ,  257     ,  262     ,  264     ,  268
                                      274     ,  280     ,  286     ,  291     ,  293     ,  303     ,  332     ,  335     ,  355
                                      357     ,  560     ,  585

TOTAL SYMBOLS = 137

END OF XREF LIST
```

アンケート記入のお願い

お手数ですが，このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

［ドキュメント名］ μPD17103 アプリケーション・ノート

（IEA-675（第1版））

［お名前など］（さしつかえのない範囲で）

御社名（学校名，その他） （ ）
ご住所 （ ）
お電話番号 （ ）
お仕事の内容 （ ）
お名前 （ ）

1．ご評価（各欄に○をご記入ください）

項　　目	大変良い	良　い	普　通	悪　い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン，字の大きさなど					
そ の 他（ ）					
（ ）					

2．わかりやすい所（第　章，第　章，第　章，第　章，その他 ）

理由 [ ]

3．わかりにくい所（第　章，第　章，第　章，第　章，その他 ）

理由 [ ]

4．ご意見，ご要望

5．このドキュメントをお届けしたのは

NEC 販売員，特約店販売員，NEC 半導体ソリューション技術本部員，
その他（ ）

ご協力ありがとうございました。
下記あてにFAXで送信いただくか，最寄りの販売員にコピーをお渡しください。

NEC 半導体インフォメーションセンター
FAX：(044)548-7900

キリトリ

お問い合わせは，最寄りのNECへ

【営業関係お問い合わせ先】

半導体第一販売事業部 半導体第二販売事業部 半導体第三販売事業部	〒108-01 東京都港区芝五丁目7番1号（NEC本社ビル）	東京	(03)3454-1111 (大代表)
中部支社 半導体第一販売部 半導体第二販売部	〒460 名古屋市中区錦一丁目17番1号（NEC中部ビル）	名古屋 名古屋	(052)222-2170 (052)222-2190
関西支社 半導体第一販売部 半導体第二販売部 半導体第三販売部	〒540 大阪市中央区城見一丁目4番24号（NEC関西ビル）	大阪 大阪 大阪	(06) 945-3178 (06) 945-3200 (06) 945-3208

北海道支社 札幌	(011)231-0161	太田支店 太田	(0276)46-4011	富山支店 富山	(0764)31-8461
東北支社 仙台	(022)267-8740	宇都宮支店 宇都宮	(028)621-2281	三重支店 津	(0592)25-7341
岩手支店 盛岡	(0196)51-4344	小山支店 小山	(0285)24-5011	京都支社 京都	(075)344-7824
山形支店 山形	(0236)23-5511	長野支社 松本	(0263)35-1662	神戸支社 神戸	(078)333-3854
郡山支店 郡山	(0249)23-5511	甲府支店 甲府	(0552)24-4141	中国支社 広島	(082)242-5504
いわき支店 いわき	(0246)21-5511	埼玉支社 大宮	(048)641-1411	鳥取支店 鳥取	(0857)27-5311
長岡支店 長岡	(0258)36-2155	立川支社 立川	(0425)26-5981	岡山支店 岡山	(086)225-4455
土浦支店 土浦	(0298)23-6161	千葉支社 千葉	(043)238-8116	四国支社 高松	(0878)36-1200
水戸支店 水戸	(029)226-1717	静岡支社 静岡	(054)255-2211	新居浜支店 新居浜	(0897)32-5001
神奈川支社 横浜	(045)324-5524	北陸支社 金沢	(0762)23-1621	松山支店 松山	(089)945-4149
群馬支店 高崎	(0273)26-1255	福井支店 福井	(0776)22-1866	九州支社 福岡	(092)271-7700

【本資料に関する技術お問い合わせ先】

半導体ソリューション技術本部 マイクロコンピュータ技術部	〒210 川崎市幸区塚越三丁目484番地	川崎	(044)548-7923
半導体販売技術本部 東日本販売技術部	〒108-01 東京都港区芝五丁目7番1号（NEC本社ビル）	東京	(03)3798-9619
半導体販売技術本部 中部販売技術部	〒460 名古屋市中区錦一丁目17番1号（NEC中部ビル）	名古屋	(052)222-2125
半導体販売技術本部 西日本販売技術部	〒540 大阪市中央区城見一丁目4番24号（NEC関西ビル）	大阪	(06) 945-3383

半導体
インフォメーションセンター
FAX(044)548-7900
(FAXにてお願い致します)

C96.8