

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ひ申し上げます。

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のですが、誤りが無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、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と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。

μ PD17201A

4ビット・シングルチップ・マイクロコントローラ

μ PD17201A

4ビット・シングルチップ・マイクロコントローラ

- 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。
- この製品を使用したことにより、第三者の工業所有権等にかかわる問題が発生した場合、当社製品の構造製法に直接かかわるもの以外につきましては、当社はその責を負いませんのでご了承ください。
- 当社は、航空宇宙機器、海底中継器、原子力制御システム、生命維持のための医療用機器など極めて高い信頼性が要求される「特定」用途に推奨できる製品を標準的には用意しておりません。当社製品を「特定」用途にご使用をお考えのお客様、および、「標準」品質水準品を当社が意図した用途以外にご使用をお考えのお客様は、事前に販売窓口までご連絡頂きますようお願い致します。

当社推奨の用途例

- 標準：電算機、事務器、通信機器(端末、移動体)、計測機器、AV機器、家電、産業用ロボット等
- 特別：自動車電装、列車制御、通信機器(幹線)、交通信号制御、燃焼制御、防災・防犯装置等
- この製品は耐放射線設計をしておりません。

目 次

第1章 概 説	1
第2章 キー・スキャン	3
2.1 プログラム説明	3
第3章 LCD表示	49
3.1 LCD仕様	49
3.2 4時分割表示	51
3.2.1 表示データ	51
3.2.2 4時分割LCDパネル	53
第4章 タイマ	93
4.1 プログラム説明	93
4.1.1 時計カウント処理	93
4.1.2 オン／オフ・タイマ処理	93
第5章 リモコン応用	133
5.1 受 信	133
5.1.1 プログラム説明	135
5.2 送 信	193
5.2.1 プログラム説明	193
第6章 A/D変換	237
6.1 プログラム説明	237
第7章 シリアルI/O	283
7.1 プログラム説明	283
第8章 スタンバイ機能および システム・クロックの切り替え	327
8.1 プログラム説明	327
8.1.1 HALTモード	327
8.1.2 STOPモード	327
8.2 システム・クロック	328
8.3 ウォッチドッグ・タイマ動作タイミング	328

第1章 概 説

μ PD17201AはLCDコントローラ／ドライバ、A/Dコンバータ、リモコン用キャリア発生回路を1チップに納めた赤外線リモート・コントローラ用4ビット・シングルチップ・マイクロコントローラです。 μ PD17201Aは、CPUとしてアキュムレータがなく、1命令でデータ・メモリ間あるいはデータ・メモリと周辺回路との間の転送や演算が可能です。また、すべての命令が16ビットの1語で構成されています。 μ PD17201Aのシステム開発用として、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。

このアプリケーション・ノートの各章内の「ドキュメント」、「RAMマップ」、「RAM使用説明」、「クロスレファレンス」の項目については、17KシリーズのアセンブラであるAS17Kのドキュメント機能を使用して作成しています。

なお、赤外線リモコン送信機用マイクロコントローラ・シリーズとしては、 μ PD17201Aのほかに、 μ PD17202A、 μ PD17207、また、学習リモコン用として、 μ PD17203A、 μ PD17204があります。



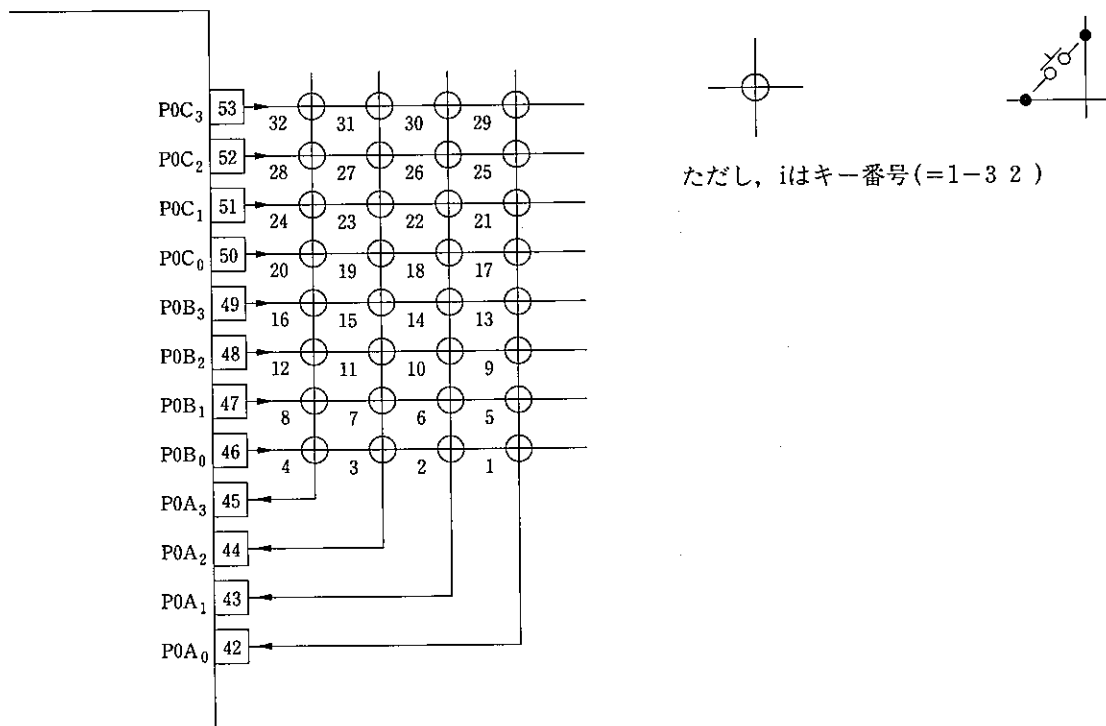
第2章 キー・スキャン

ここでは、モメンタリ・キーの取り込みを、ポートを使用して行うキー・スキャン方法について説明します。

P0A₀-P0A₃を入力とし、P0B₀-P0B₃, P0C₀-P0C₃を出力とした32 (4 × 8) キーのキー入力を行います。

図2-1にキー・マトリクスの構成図を示します。

図2-1 キー・マトリクスの構成図



2.1 プログラム説明

本プログラムでは、入力されたキー・データは1キーを8ビットで表し、RAMの8ビット (KEYDATA) に格納します。

キー・データの割り付けは、表2-1のようになります。

また、キー入力の状態 (キーが押されているか、無効なキーか、押されているキーが変わったか) をステータス (K_EXIST, K_ERROR, K_CHANGE) によって表します。

表2-1 キー・データ割り付け

キー番号	キー・データ(HEX)	キー番号	キー・データ(HEX)
1	00	17	10
2	01	18	11
3	02	19	12
4	03	20	13
5	04	21	14
6	05	22	15
7	06	23	16
8	07	24	17
9	08	25	18
10	09	26	19
11	0A	27	1A
12	0B	28	1B
13	0C	29	1C
14	0D	30	1D
15	0E	31	1E
16	0F	32	1F

本プログラムでは、二重押しおよび多重押しは無効（キー・エラー）とし、キー・データはFF（HEX）としています。

二重押しによる動作を行わせる場合は、このプログラムの最初で有効な二重押しかどうかの判断処理を行うことにより有効です。

〈使用するハードウェア〉

ポート：P0A, P0B, P0C

〈初期設定〉

ポート・モード：P0A←入力モード

P0B, P0C←出力モード

プルアップ抵抗：P0A←プルアップ抵抗内蔵

キー・データ : FF (HEX)

〈起動方法〉

本キー入力サブルーチンは、3回連続して同じキー入力状態のとき、有効なキー入力と判断します。したがってチャタリング吸収時間を仮に15msにするには、約5msごとにコールすればよいことになります。

ドキュメント

AS17K Z1.10 D1 << E17201A DOCUMENT >> 10:36:20 02/16/91

TABLE OF CONTENTS

	PAGE
第2章 キー・スキャン	1
<プログラム説明>	1
<<RAMの使用説明>	2
<<キー入力状態>	3
<<キーの数字>	4
<<キー・リターンの安定待ち時間>	5
<<キー押しおび多量押し>	6
<<キー・リターン設定>	6
<<オン／オフ・チャタリング・チェック>	8

第 2 章 キー・スキャン

PUBLIC (FLG) : OFFC_F, ONC_F
 PUBLIC (LAB) : KEYSKAN
 ADDR RANGE : 0000H - 00A9H

<プログラム説明>

このプログラムは、モメンタリ・キーの取り込み方を、ポートを使用したキー・スキャン方法について説明します。

ENTRANCES :-
 MEMORIES CHANGED :-
 MEMORIES REFERRED :-
 MEMORIES MANIPULATED :-
 FLAGS CHANGED :-
 FLAGS REFERRED :-
 DATA REFERRED :-
 BRANCH TO :-
 SUBROUTINES CALLED :-
 LABELS MANIPULATED :-
 SYSTEM CALL :-

<RAM使用説明>

WORK1 0. 0 0 H 汎用レジスタです。
 WORK2 0. 0 1 H 汎用レジスタです。
 SCANDAT_H 0. 1 0 H キー・ソースを1本ずつ出すためのスキャン・データ用の
 SCANDAT_L エリアです。
 CHAT_CNT 0. 1 2 H チャタリング回数設定用のエリアです。
 RETURN 0. 1 3 H キー・リターン・データを格納するためのエリアです。
 SOURCE_H 0. 1 4 H キー・ソース・データ用のエリアです。
 SOURCE_L
 KEYDATA_H 0. 1 8 H キー・データ（キー番号）用のエリアです。
 KEYDATA_L
 B_K_DAT1_H 0. 1 A H 前のキー・データを格納しておくためのエリアです。
 B_K_DAT1_L
 B_K_DAT2_H 0. 1 C H 前のキー・データを格納しておくためのエリアです。
 B_K_DAT2_L
 K_SCAN_C 0. 2 0 H キー・ソースを1本ずつ出す際、8本分カウントする
 ためのカウンタです。


```

TM_C ..... 0, 2 1 H ..... 3 8 μs 待ち用のカウンタです。
KEY_RET ..... 0, 7 0 H ..... 入力ポート (P O A) 用のエリアです。
SOURCE1 ..... 0, 7 1 H ..... 出力ポート (P O B) 用のエリアです。
SOURCE2 ..... 0, 7 2 H ..... 出力ポート (P O C) 用のエリアです。
KEY_FLG ..... 0, 1 E H ..... キーの入力状態を知るためのフラグ用のエリアです。
K_EXIST ..... # 3 ..... キーが押されている場合セットされます。
K_CHANGE ..... # 2 ..... 押されているキーが変更した場合セットされます。
K_ERROR ..... # 1 ..... 二重押しおよび多重押し(無効)の場合セットされます。

KEY1_FLG ..... 0, 1 F H ..... キーのチャタリング状態を知るためのフラグ用のエリアです。
ONC_F ..... # 3 ..... オン・チャタリング中の場合セットされます。
OFFC_F ..... # 2 ..... オフ・チャタリング中の場合セットされます。
CHECK_F ..... # 1 ..... 二重押しおよび多重押し(無効)以外の場合セットされます。
    
```

```

ENTRANCES      :-
MEMORIES CHANGED :-
MEMORIES REFERRED :-
MEMORIES MANIPULATED :-
FLAGS CHANGED  :-
FLAGS REFERRED :-
DATA REFERRED  :-
BRANCH TO      :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL    :-
    
```

<キーの入力状態>

キーの入力状態をステータス (K_EXIST, K_ERROR, K_CHANGE) であらわします。
 キーが押されている場合、K_EXISTをセットし、多重押しは無効で、K_ERROR がセットされます。
 押されているキーが変更した場合、K_CHANGEがセットされます。

```

ENTRANCES      :KEYSCAN
MEMORIES CHANGED :B_K_DAT2_H   B_K_DAT2_L
                  K_SCAN_C     SCANDAT_H
                  SCANDAT_L     WORK1
MEMORIES REFERRED :KEYDATA_H   KEYDATA_L
                  WORK1        WORK2
MEMORIES MANIPULATED :-
FLAGS CHANGED    :CHECK_F     K_CHANGE
                  K_ERROR
FLAGS REFERRED   :OFFC_F     ONC_F
DATA REFERRED    :-
BRANCH TO       :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL     :-
    
```

<キー数>

P0A0~P0A3をキー・リターン, P0B0~P0B3, P0C0~P0C3をキー・ソースとした32(4×8)キーのキー入力を行います。

ENTRANCES	:SOUR_SET	
MEMORIES CHANGED	:SOURCE1	SOURCE2
	WORK1	WORK2
MEMORIES REFERRED	:SCANDAT_H	SCANDAT_L
	WORK1	WORK2
MEMORIES MANIPULATED	:-	
FLAGS CHANGED	:-	
FLAGS REFERRED	:-	
DATA REFERRED	:-	
BRANCH TO	:-	
SUBROUTINES CALLED	:-	
LABELS MANIPULATED	:-	
SYSTEM CALL	:-	

<キー・リターンの安定待ち時間>

ここでは、キー・ソースを出したあとキー・リターンより、安定したデータが得られるまでの時間待ちをします。
また、TM_Cの値を変えることによりこの待ち時間を変えることができます。

TM_C(H)	待ち時間(μs)	
0 F	1 2	(M I N)
0 E	2 4	
0 D	3 8	
0 C	4 8	
0 B	6 0	
0 A	7 2	
9	8 4	
8	9 8	
7	1 0 8	
6	1 2 0	
5	1 3 2	
4	1 4 4	
3	1 5 6	
2	1 6 8	
1	1 7 0	
0	1 8 2	(M A X)

```

ENTRANCES          :-
MEMORIES CHANGED   :-TM_C      WORK1
MEMORIES REFERRED  :-KEY_RET
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-WAIT_TM
BRANCH TO          :-RET_CHK   ;単押し
                   SCAN       ;どのキーも押されていなければキー・スキャン・チェックへ
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

<二重押しおよび多重押し>

二重押しおよび多重押しは、無効（キー・エラー）とし、キー・データをFF（H）としていますが、二重押しで何かの動作をさせたい場合は、このプログラムの最初に有効な二重押しかどうかの判断処理を行い有効であれば、その二重押しの処理を行うことが可能です。

ENTRANCES	:RET_CHK	
MEMORIES CHANGED	:KEYDATA_H	KEYDATA_L
	RETURN	SOURCE_H
	SOURCE_L	WORK1
	WORK2	
MEMORIES REFERRED	:KEY_RET	SCANDAT_H
	SCANDAT_L	WORK1
	WORK2	
MEMORIES MANIPULATED	:-	
FLAGS CHANGED	:CHECK_F	K_CHANGE
	K_ERROR	K_EXIST
	OPFC_F	ONC_F
FLAGS REFERRED	:CHECK_F	
DATA REFERRED	:-	
BRANCH TO	:-	
SUBROUTINES CALLED	:-	
LABELS MANIPULATED	:-	
SYSTEM CALL	:-	

<キー・データ設定>

入力されたキー・データは、1キーを8ビットであらわし、RAMの8ビット(B_K_DAT1 ... 前キー・データ)に格納します。

```

ENTRANCES          :SCAN
MEMORIES CHANGED   :B_K_DAT1_H   B_K_DAT1_L
                   :K_SCAN_C     SCANDAT_H
                   :SCANDAT_L     WORK1
                   :WORK2
MEMORIES REFERRED   :RETURN       SCANDAT_H
                   :SCANDAT_L     SOURCE_H
                   :SOURCE_L      WORK1
                   :WORK2
MEMORIES MANIPULATED :--
FLAGS CHANGED      :--
FLAGS REFERRED     :--
DATA REFERRED      :--
BRANCH TO          :SCAN_END      ;
                   :SOUR_SET      ;
SUBROUTINES CALLED :--
LABELS MANIPULATED :--
SYSTEM CALL        :--

```

<オン/オフ・チャタリング・チェック>

このプログラムでは、オン・チャタリングおよびオフ・チャタリングを、それぞれ3回ずつチェックします。3回ともキー入力状態が一致した場合そのキー・データ(KEYDATA)が確定します。

```

ENTRANCES          :SCAN_END
MEMORIES CHANGED   :B_K_DAT2_H   B_K_DAT2_L
                   :CHAT_CNT     KEYDATA_H
                   :KEYDATA_L     WORK1
                   :WORK2
MEMORIES REFERRED   :B_K_DAT1_H   B_K_DAT1_L
                   :B_K_DAT2_H   B_K_DAT2_L
                   :KEYDATA_H     KEYDATA_L
                   :KEYDATA_H     WORK1
                   :WORK2
MEMORIES MANIPULATED :--
FLAGS CHANGED      :K_CHANGE     K_ERROR
                   :K_EXIST      OFFC_F
                   :ONC_F
FLAGS REFERRED     :CHECK_F      K_EXIST
                   :OFFC_F      ONC_F
DATA REFERRED      :--
BRANCH TO          :--
SUBROUTINES CALLED :--
LABELS MANIPULATED :--
SYSTEM CALL        :--

```


RAMマップ

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 10:36:04 02/15/91 PAGE 02-001

PROG =

SOURCE = KEY.ASM

(PUBLIC/LOCAL MAP)

BANK = 0

		ROW								
		7	6	5	4	3	2	1	0	
			!LCDD32	!LCDD16	!LCDD0					0
!KEY_RET							!K_SCAN_C	!SCANDAT_H	!WORK1	
			!LCDD33	!LCDD17	!LCDD1					1
!SOURCE1							!TM_C	!SCANDAT_L	!WORK2	
			!LCDD34	!LCDD18	!LCDD2					2
!SOURCE2								!CHAT_CNT		COL
			!LCDD35	!LCDD19	!LCDD3					3
								!RETURN		
!AR3				!LCDD20	!LCDD4					4
								!SOURCE_H		
!AR2				!LCDD21	!LCDD5					5
								!SOURCE_L		
!AR1				!LCDD22	!LCDD6					6
!AR0				!LCDD23	!LCDD7					7
!WR				!LCDD24	!LCDD8					8
								!KEYDATA_H		
!BANK				!LCDD25	!LCDD9					9
								!KEYDATA_L		
!IXH				!LCDD26	!LCDD10					A
!MPH										
								!B_K_DAT1_H		
!IXM				!LCDD27	!LCDD11					B
!MPL										
								!B_K_DAT1_L		

AS17K Z1.10 D1 << B17201A DATA MEMORY MAP >> 10:36:04 02/15/91 PAGE 02-002

PROG -

SOURCE - KEY.ASM

(PUBLIC/LOCAL MAP)

BANK - 0

ROW

	7	6	6	4	3	2	1	0	
IXL			LCDD28	LCDD12				DBF9	C
							IB_K_DAT2_H		
IRPH			LCDD29	LCDD13				DBF2	D
							IB_K_DAT2_L		
IRPL			LCDD30	LCDD14				DBF1	R
							KEY_FLG		
IPSW			LCDD31	LCDD15				DBF0	F
							KEY1_FLG		

AS17K Z1.10 D1 << E17201A FLG MAP >> 10:36:04 02/15/91 PAGE 02-003

PROG -

SOURCE - KEY.ASM

(FLAG MAP) BANK = 0

MSB	3	2	1	0	LSB
1E					
	IK_EXIST	IK_CHANGE	IK_ERROR		
1F					
	IONC_F	IOFFC_F	ICHECK_F		
70					
	IPOA3	IPOA2	IPOA1	IPOA0	
71					
	IPOB3	IPOB2	IPOB1	IPOB0	
72					
	IPOC3	IPOC2	IPOC1	IPOC0	
73					
	IPOD3	IPOD2	IPOD1	IPOD0	
7A					
	IMPE				
7E					
				IBCD	
7F					
	ICMF	ICY	IZ	IIXE	
82					
			ISYCK	IXEN	
83					
	INTRES	INTWMD	INTWRES		
87					
	IVDDDET1	IVDDDET0			
8F					
				INT	
91					
				INRZBF	
92					
				INRZ	
A0					
				ADCCMP	
A1					
	IVPREFEN	IADCBN	IADCCHI	IADCCHO	

AS17K Z1.10 D1 << B17201A FLG MAP >> 10:36:04 02/15/91 PAGE 02-004

PROG =

SOURCE = KEY.ASM

(FLAG MAP) BANK = 0

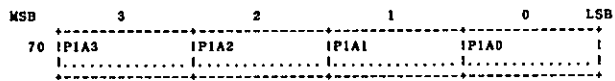
MSB	3	2	1	0	LSB
A2	ISLOTS	ISIOHIZ	SILOCK1	SILOCK0	
A3		INRZEN	ITMOE	ISIOEN	
A7	IP0DB103	IP0DB102	IP0DB101	IP0DB100	
AP	IPSI0	IPWTM	IP	IPTM	
B1	LCDEN	LCCK2	LCCK1	LCCK0	
B2	LCMD3	LCMD2	LCMD1	LCMD0	
B3	TMEN	TMRES	TMCK1	TMCK0	
B7	PIAGIO	POCGIO	POBGIO	POAGIO	
BB	IRQSI0				
BC		IRQWTM			
BD			IRQ		
BE				IRQTM	

AS17K Z1.10 D1 << E17201A FLG MAP >> 10:36:04 02/15/81 PAGE 02-005

PROG *

SOURCE - KEY.ASM

(FLAG MAP) BANK - 1



RAM使用説明

AS17K Z1.10 D1 << R17201A DATA MEMORY MAP >> 10:38:04 02/15/91 PAGE 02-006

PROG =

SOURCE = KEY.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
ADCC00	FLG	0	A1	...1	
ADCC01	FLG	0	A1	...1	
ADCC0P	FLG	0	A0	...1	
ADCC0N	FLG	0	A1	...1	
AR0	MEM	0	77		
AR1	MEM	0	76		
AR2	MEM	0	75		
AR3	MEM	0	74		
B_K_DAT1_H	MEM	0	1A		前のキー・データを格納しておくためのエリアです。
B_K_DAT1_L	MEM	0	1B		
B_K_DAT2_H	MEM	0	1C		前のキー・データを格納しておくためのエリアです。
B_K_DAT2_L	MEM	0	1D		
BANK	MEM	0	79		
BCD	FLG	0	7E	...1	
CHAT_CNT	MEM	0	12		チャタリング回数設定用のエリアです。
CHECK_F	FLG	0	1F	...1	二重押しおよび多重押し(無効)以外の場合セットされます。
CMP	FLG	0	7F	...1	
CY	FLG	0	7F	...1	
DBF0	MEM	0	F		
DBF1	MEM	0	E		
DBF2	MEM	0	D		
DBF3	MEM	0	C		
INT	FLG	0	8F	...1	
IP	FLG	0	AF	...1	
IPSI0	FLG	0	AF	...1	
IPTM	FLG	0	AF	...1	
IPWTM	FLG	0	AF	...1	
IRQ	FLG	0	BD	...1	
IRQSIO	FLG	0	BD	...1	
IRQTM	FLG	0	BE	...1	
IRQWTM	FLG	0	BC	...1	
IXE	FLG	0	7F	...1	
IXH	MEM	0	7A		
IXL	MEM	0	7C		
IXM	MEM	0	7B		
K_CHANGE	FLG	0	1E	...1	押されているキーが変更された場合セットされます。
K_ERROR	FLG	0	1E	...1	二重押しおよび多重押し(無効)の場合セットされます。
K_EXIT	FLG	0	1E	...1	キーが押されている場合セットされます。
K_SCAN_C	MEM	0	20		キー・ソースを1本ずつ出す際のカウンタ用のエリアです。
KEY1_FLG	MEM	0	1F		キーのチャタリング状態を知るためのフラグ用のエリアです。
KEY_FLG	MEM	0	1E		キーの入力状態を知るためのフラグ用のエリアです。
KEY_RET	MEM	0	70		キー入力ポート(F0A)用のエリアです。
KEYDATA_H	MEM	0	18		キー・データ(キー番号)用のエリアです。
KEYDATA_L	MEM	0	19		
LCDC00	FLG	0	B1	...1	
LCDC01	FLG	0	B1	...1	
LCDC02	FLG	0	B1	...1	
LCDD0	MEM	0	40		
LCDD1	MEM	0	41		
LCDD10	MEM	0	4A		
LCDD11	MEM	0	4B		
LCDD12	MEM	0	4C		

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 10:38:04 02/15/91 PAGE 02-007

PROG *

SOURCE = KEY.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCDD13	MEM	0	4D		
LCDD14	MEM	0	4E		
LCDD15	MEM	0	4F		
LCDD16	MEM	0	50		
LCDD17	MEM	0	51		
LCDD18	MEM	0	52		
LCDD19	MEM	0	53		
LCDD2	MEM	0	42		
LCDD20	MEM	0	54		
LCDD21	MEM	0	55		
LCDD22	MEM	0	56		
LCDD23	MEM	0	57		
LCDD24	MEM	0	58		
LCDD25	MEM	0	59		
LCDD26	MEM	0	5A		
LCDD27	MEM	0	5B		
LCDD28	MEM	0	5C		
LCDD29	MEM	0	5D		
LCDD3	MEM	0	43		
LCDD30	MEM	0	5E		
LCDD31	MEM	0	5F		
LCDD32	MEM	0	60		
LCDD33	MEM	0	61		
LCDD34	MEM	0	62		
LCDD35	MEM	0	63		
LCDD4	MEM	0	44		
LCDD5	MEM	0	45		
LCDD6	MEM	0	46		
LCDD7	MEM	0	47		
LCDD8	MEM	0	48		
LCDD9	MEM	0	49		
LCDEN	FLG	0	B1	1...	
LCDM00	FLG	0	B2	...1	
LCDM01	FLG	0	B2	..1.	
LCDM02	FLG	0	B2	.1..	
LCDM03	FLG	0	B2	1...	
MPF	FLG	0	7A	1...	
MPH	MEM	0	7A		
MPL	MEM	0	7B		
NRZ	FLG	0	92	...1	
NRZBF	FLG	0	91	...1	
NRZEN	FLG	0	A3	1..	
OFFC_F	FLG	0	1F	1..	オフ・チャタリング中の場合セットされます。
ONC_F	FLG	0	1F	1..	オン・チャタリング中の場合セットされます。
POA0	FLG	0	70	...1	
POA1	FLG	0	70	..1.	
POA2	FLG	0	70	.1..	
POA3	FLG	0	70	1...	
POAGIO	FLG	0	B7	...1	
POB0	FLG	0	71	...1	
POB1	FLG	0	71	..1.	
POB2	FLG	0	71	1..	

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 10:36:04 02/15/91 PAGE 02-008

PROG -

SOURCE - KEY.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
POB3	FLG	0	71	1...	
POBGIO	FLG	0	B7	...1	
POC0	FLG	0	72	...1	
POC1	FLG	0	72	...1	
POC2	FLG	0	72	...1	
POC3	FLG	0	72	1...	
POCGIO	FLG	0	B7	...1	
POD0	FLG	0	73	...1	
POD1	FLG	0	73	...1	
POD2	FLG	0	73	...1	
POD3	FLG	0	73	1...	
PODBIO0	FLG	0	A7	...1	
PODBIO1	FLG	0	A7	...1	
PODBIO2	FLG	0	A7	...1	
PODBIO3	FLG	0	A7	1...	
PIA0	FLG	1	70	...1	
PIA1	FLG	1	70	...1	
PIA2	FLG	1	70	...1	
PIA3	FLG	1	70	1...	
PIAGIO	FLG	0	B7	1...	
PSW	MEM	0	7F		
RETURN	MEM	0	13		キー・リターン・データを格納するためのエリアです。
RPH	MEM	0	7D		
RPL	MEM	0	7E		
SCANDAT_H	MEM	0	10		キー・ソースを1本ずつ出すためのスキャン・データ用の
SCANDAT_L	MEM	0	11		エリアです。
SI0CK0	FLG	0	A2	...1	
SI0CK1	FLG	0	A2	...1	
SI0EN	FLG	0	A3	...1	
SI0HIZ	FLG	0	A2	...1	
SI0TS	FLG	0	A2	1...	
SOURCE1	MEM	0	71		出力ポート (POB) 用のエリアです。
SOURCE2	MEM	0	72		出力ポート (POC) 用のエリアです。
SOURCE_H	MEM	0	14		キー・ソース・データ用のエリアです。
SOURCE_L	MEM	0	15		
SYSC	FLG	0	82	...1	
TM_C	MEM	0	21		30μs待ち用のカウンタです。
TMCK0	FLG	0	B3	...1	
TMCK1	FLG	0	B3	...1	
TMEN	FLG	0	B3	...1	
TMOE	FLG	0	A3	...1	
TMRES	FLG	0	B3	...1	
VDDDET0	FLG	0	87	...1	
VDDDET1	FLG	0	87	1...	
VREFEN	FLG	0	A1	1...	
WDTRES	FLG	0	83	1...	
WORK1	MEM	0	0		汎用レジスタです。
WORK2	MEM	0	1		汎用レジスタです。
WR	MEM	0	78		
WTMMD	FLG	0	83	...1	
WTMRES	FLG	0	83	...1	
XEN	FLG	0	82	...1	

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 10:36:04 02/15/91 PAGE 02-009

PROG -

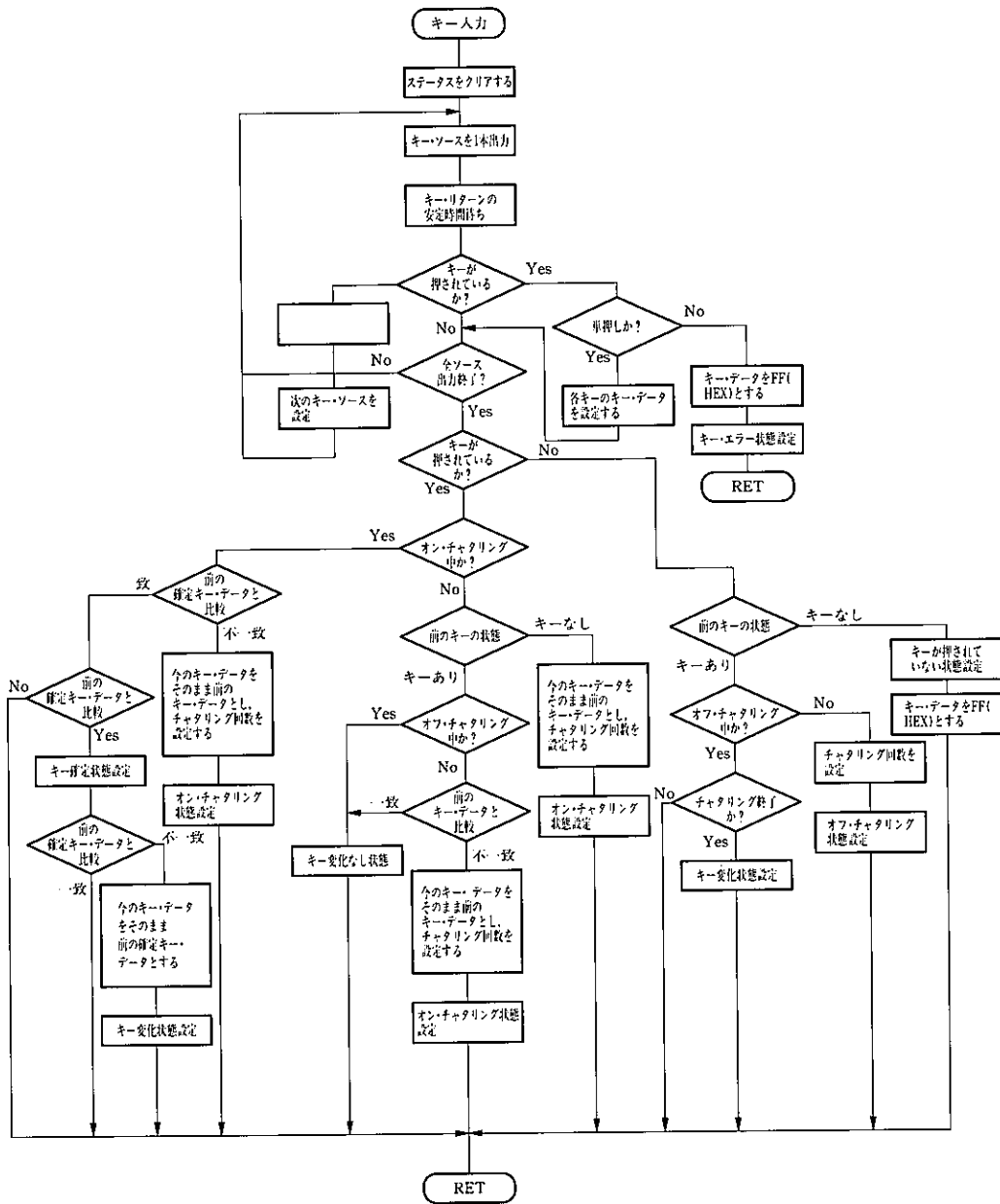
SOURCE - KEY.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
Z	PLG	0	7F	..1.	

フロー・チャート

フロー・チャート



リスト


```
INCLUDE 'KEY.PUB'
SUMMARY      '%%', '第2章 キー・スキャン'
%%
SUMMARY      '%%', '<プログラム説明>'
```

このプログラムは、モメンタリ・キーの取り込み方を、ポートを使用したキー・スキャン方法について説明します。

```
%%
;*****
;*** 使用メモリ、レジスタ、フラグ ***
;*****
```

WORK1	MEM	0.00H	;汎用レジスタです。
WORK2	MEM	0.01H	;汎用レジスタです。
SCANDAT_H	MEM	0.10H	;キー・ソースを1本ずつ出すためのスキャン・データ用の
SCANDAT_L	MEM	0.11H	;エリアです。
CHAT_CNT	MEM	0.12H	;チャタリング回数設定用のエリアです。
RETURN	MEM	0.13H	;キー・リターン・データを格納するためのエリアです。
SOURCE_H	MEM	0.14H	;キー・ソース・データ用のエリアです。
SOURCE_L	MEM	0.15H	
KEYDATA_H	MEM	0.18H	;キー・データ(キー番号)用のエリアです。
KEYDATA_L	MEM	0.19H	
B_K_DAT1_H	MEM	0.1AH	;前のキー・データを格納しておくためのエリアです。
B_K_DAT1_L	MEM	0.1BH	
B_K_DAT2_H	MEM	0.1CH	;前のキー・データを格納しておくためのエリアです。
B_K_DAT2_L	MEM	0.1DH	
K_SCAN_C	MEM	0.20H	;キー・ソースを1本ずつ出す際、8本分カウントするためのカウンタです。
TM_C	MEM	0.21H	;36μs待ち用のカウンタです。
KEY_RET	MEM	0.70H	;入力ポート(P0A)用のエリアです。
SOURCE1	MEM	0.71H	;出力ポート(P0B)用のエリアです。
SOURCE2	MEM	0.72H	;出力ポート(P0C)用のエリアです。
EJECT			
KEY_FLG	MEM	0.1EH	;キーの入力状態を知るためのフラグ用のエリアです。
K_EXIST	FLG	KEY_FLG.3	;キーが押されている場合セットされます。
K_CHANGE	FLG	KEY_FLG.2	;押されているキーが変更された場合セットされます。
K_ERROR	FLG	KEY_FLG.1	;二重押しおよび多重押し(無効)の場合セットされます。
KEY1_FLG	MEM	0.1FH	;キーのチャタリング状態を知るためのフラグ用のエリアです。
ONC_F	PLG	KEY1_FLG.3	;オン・チャタリング中の場合セットされます。
OFFC_F	PLG	KEY1_FLG.2	;オフ・チャタリング中の場合セットされます。
CHECK_F	PLG	KEY1_FLG.1	;二重押しおよび多重押し(無効)以外の場合セットされます。
WAIT_TM	DAT	0DH	;キー入力用ウエイト時間(36μs)

```
SUMMARY      '%%', '<RAM使用説明>'
```

WORK1	0.00H	汎用レジスタです。
WORK2	0.01H	汎用レジスタです。
SCANDAT_H	0.10H	キー・ソースを1本ずつ出すためのスキャン・データ用の
SCANDAT_L		エリアです。
CHAT_CNT	0.12H	チャタリング回数設定用のエリアです。
RETURN	0.13H	キー・リターン・データを格納するためのエリアです。

```

SOURCE_H ..... 0, 14 H ..... キー・ソース・データ用のエリアです。
SOURCE_L

KEYDATA_H ..... 0, 18 H ..... キー・データ (キー番号) 用のエリアです。
KEYDATA_L

B_K_DAT1_H ..... 0, 1A H ..... 前のキー・データを格納しておくためのエリアです。
B_K_DAT1_L

B_K_DAT2_H ..... 0, 1C H ..... 前のキー・データを格納しておくためのエリアです。
B_K_DAT2_L

K_SCAN_C ..... 0, 20 H ..... キー・ソースを1本ずつ出す際、8本分カウントする
                           ためのカウンタです。
.EJECT

TM_C ..... 0, 21 H ..... 38μs待ち用のカウンタです。

KEY_RET ..... 0, 70 H ..... 入力ポート (P0A) 用のエリアです。

SOURCE1 ..... 0, 71 H ..... 出力ポート (P0B) 用のエリアです。

SOURCE2 ..... 0, 72 H ..... 出力ポート (P0C) 用のエリアです。

KEY_PLG ..... 0, 1E H ..... キーの入力状態を知るためのフラグ用のエリアです。
K_EXIST ..... #3 ..... キーが押されている場合セットされます。
K_CHANGE ..... #2 ..... 押されているキーが変更した場合セットされます。
K_ERROR ..... #1 ..... 二重押しおよび多重押し (無効) の場合セットされます。

KEY1_PLG ..... 0, 1FH ..... キーのチャタリング状態を知るためのフラグ用のエリアです。
ONC_F ..... #3 ..... オン・チャタリング中の場合セットされます。
OFFC_F ..... #2 ..... オフ・チャタリング中の場合セットされます。
CHECK_F ..... #1 ..... 二重押しおよび多重押し (無効) 以外の場合セットされます。

```

%%

```

;.....
;*** キー・スキャン ***
;.....
SUMMARY '%%', '<キーの入力状態>'

```

キーの入力状態をステータス (K_EXIST, K_ERROR, K_CHANGE) であらわします。
 キーが押されている場合、K_EXISTをセットし、多重押しは無効で、K_ERRORがセットされます。
 押されているキーが変更した場合、K_CHANGEがセットされます。

.EJ-1

%%

KEYSCAN:

```

SKF2  ONC_F, OFFC_F
BR     STAT_CLS
LD     WORK1, KEYDATA_H
LD     WORK2, KEYDATA_L
ST     B_K_DAT2_H, WORK1
ST     B_K_DAT2_L, WORK2

```

STAT_CLS:

```

CLR3  K_CHANGE, K_ERROR, CHECK_F      ;ステータスをクリアする
MOV   K_SCAN_C, #8                    ;キー・スキャン用のカウンタを設定する
MOV   SCANDAT_L, #111101111B AND 0FH  ;スキャン・データの初期化を行う
MOV   SCANDAT_H, #111101111B SHR 4) AND 0FH

```

SUMMARY

'%%', '<キー数>'

P0A0~P0A3をキー・リターン、P0B0~P0B3, P0C0~P0C3をキー・ソースとした32(4×8)キーのキー入力を行います。

.EJ-1

```

%%
SOUR_SET:                                ;キー・ソースを1本ずつ出力する
LD    WORK1, SCANDAT_H                    ;スキャン・データの設定
LD    WORK2, SCANDAT_L
ST    SOURCE1, WORK1
ST    SOURCE2, WORK2
    
```

SUMMARY '%%', '<キー・リターンの安定待ち時間>'

ここでは、キー・ソースを出したあとキー・リターンより、安定したデータが得られるまでの時間待ちをします。
また、TM_Cの値を変えることによりこの待ち時間を変えることができます。

TM_C(H)	待ち時間 (μs)	
0 F	1 2	(M I N)
0 E	2 4	
0 D	3 8	
0 C	4 8	
0 B	8 0	
0 A	7 2	
9	8 4	
8	9 8	
7	1 0 8	
6	1 2 0	
5	1 3 2	
4	1 4 4	
3	1 5 6	
2	1 6 8	
1	1 7 0	
0	1 8 2	(M A X)

```

.EJ-1
%%
MOV    TM_C, #WAIT_TM                    ;38 μs (キー・リターンの安定時間) 待ち用のカウンタを設定
WAIT:
ADD    TM_C, #1
SKT1  CY
BR     WAIT
LD     WORK1, KBY_RET                    ;キーが押されているか?
SKNE  WORK1, #1111B                      ;どのキーも押されていないければキー・スキャン・チェックへ
BR     SCAN
;同一ソース上の多重押しチェック
SKE   WORK1, #1110B                       ;単押しか?
SKNE  WORK1, #1101B
    
```

```

BR      RET CK      ;単押し
SKE     WORK1, #0111B
SKNE    WORK1, #0111B
BR      RET CK      ;単押し

SUMMARY  '%%', '<二重押しおよび多重押し>'

二重押しおよび多重押しは、無効(キー・エラー)とし、キー・データをFF(H)としていますが、二重押し
で何かの動作をさせたい場合は、このプログラムの最初に有効な二重押しかどうかの判断処理を行い有効であれば、
その二重押しの処理を行うことが可能です。

.EJ-1
%%
ERROR:    ;多重押しエラー

MOV      KEYDATA_L, #0FH      ;キー・データをFF(H)とする
MOV      KEYDATA_H, #0FH
CLR3     ONC_F, OFFC_F, K_CHANGE ;キー・エラー状態設定
SET2     K_EXIST, K_ERROR
RET

RET CK:
SKF1     CHECK_F      ;前回のスキャン・データで、キーが押されていないと判断したか?
BR       ERROR        ;多重押しエラー
SET1     CHECK_F

;各キーのキー・データを設定
;-----
;      キー・リターンをストア
;-----
LD       WORK1, KEY_RET
ST       RETURN, WORK1
;-----
;      キー・ソースをストア
;-----
LD       WORK1, SCANDAT_H
LD       WORK2, SCANDAT_L
ST       SOURCE_H, WORK1
ST       SOURCE_L, WORK2

SUMMARY  '%%', '<キー・データ設定>'

入力されたキー・データは、1キーを8ビットであらわし、RAMの8ビット(B_K_DAT1 ... 前キー・データ)
に格納します。
%%
;キー・データ設定 (キー・データ割付 SOURCE1: 00-0F, SOURCE2: 10-1F)

MOV      B_K_DAT1_H, #0H
LD       WORK1, SOURCE_L
SKE     SOURCE_H, #0FH
BR      S_P0B
MOV      B_K_DAT1_H, #1H
BR      CK_0

EJECT
S_P0B:   LD       WORK1, SOURCE_H

CK_0:    ;ビット0のチェック

SKE     WORK1, #1110B
BR      CK_1
MOV      B_K_DAT1_L, #0H
BR      RYSET

CK_1:    ;ビット1のチェック

```

```

SKE   WORK1, #1101B
BR    CK_2
MOV   B_K_DAT1_L, #4H
BR    KYBSET

CK_2: ;ビット2のチェック
SKE   WORK1, #1011B
BR    CK_3
MOV   B_K_DAT1_L, #8H
BR    KYBSET
CK_3: ;ビット3のチェック
MOV   B_K_DAT1_L, #0CH

KYBSET:
;B_K_DAT1の下位4ビットの設定
;キーリターンチェック
LD    WORK1, RETURN
CLR1  CY

SHIFT:
RORC  WORK1
SKT1  CY
BR    SCAN
ADD   B_K_DAT1_L, #1
BR    SHIFT

SCAN: ;キー・スキャン・チェック
      ;スキャン・データ(キー・ソース8本分)のカウンタ
      ;全ソース出力終了か?
ADD   K_SCAN_C, #1
SKF1  CY
BR    SCAN_END

EJECT ;スキャン・データ・シフト
LD    WORK1, SCANDAT_H ;次のキー・ソース(スキャン・データ)を設定
LD    WORK2, SCANDAT_L
ADD   WORK1, WORK1
OR    WORK1, #0001B
ADDC  WORK2, WORK2
ST    SCANDAT_H, WORK1
ST    SCANDAT_L, WORK2
BR    SOUR_SET

SUMMARY
'%%', '<オン/オフ・チャタリング・チェック>'

このプログラムでは、オン・チャタリングおよびオフ・チャタリングを、それぞれ3回ずつチェックします。
3回ともキー入力状態が一致した場合そのキー・データ(KEYDATA)が確定します。

%%
SCAN_END: ;キー・スキャン終了
SKF1  CHECK_F ;キーが離されているか?
BR    ON_CHAT ;キーが押されている場合

;現在キーが離されている場合
SKF1  K_EXIST ;前のキーの状態チェック
BR    OFF_CHA1 ;キーが押されていた場合

;前のキーの状態チェックでキーが離されていた場合
CLR4  OFFFC_F, ONC_P, K_ERROR, K_CHANGE ;キーが押されていない状態設定
RET   ;OFF→OFF (離されたままの状態)
;

ON_CHAT: ;オン・チャタリング中か?
SKT1  ONC_F
BR    CHAT
LD    WORK1, B_K_DAT1_H
LD    WORK2, B_K_DAT1_L

```

```

SET2  CMP, Z           ;前のキー・データと比較
SUB   WORK1, B_K_DAT2_H
SUB   WORK2, B_K_DAT2_L
SKT1  Z
BR    ON_CHAT1       ;不一致
;一致

ADD   CHAT_CNT, #1H
SKT1  CY             ;オン・チャタリング終了?
RET   ;ON→ON
;
;オン・チャタリング終了
SET1  K_EXIST       ;キー確定状態設定
CLR1  ONC_F
LD    WORK1, B_K_DAT1_H
LD    WORK2, B_K_DAT1_L
SET2  CMP, Z           ;前のキー・データと比較
SUB   WORK1, KEYDATA_H
SUB   WORK2, KEYDATA_L
SKT1  Z
BR    CHANGE        ;不一致
CLR2  K_ERROR, K_CHANGE ;一致
RET   ;
EJRCT ;
CHANGE: ;キーが変化
LD    WORK1, B_K_DAT1_H ;今のキー・データをそのまま前の確定キー・データとする
LD    WORK2, B_K_DAT1_L
ST    KEYDATA_H, WORK1
ST    KEYDATA_L, WORK2
SET1  K_CHANGE      ;キーの変化状態設定
CLR1  K_ERROR
RET   ;
ON_CHAT1: ;ON →有効 ON
LD    WORK1, B_K_DAT1_H ;今のキー・データを、そのまま前のキー・データとして設定
LD    WORK2, B_K_DAT1_L
ST    B_K_DAT2_H, WORK1
ST    B_K_DAT2_L, WORK2
MOV   CHAT_CNT, #0DH ;チャタリング回数の設定
SET1  ONC_F         ;オン・チャタリング状態設定
CLR2  OFFC_F, K_CHANGE
RET   ;
CHAT:
SKT1  K_EXIST       ;前のキーの状態チェック
BR    CHAT1        ;キーが離されていた場合
;キーが押されていた場合
SKF1  OFFC_F        ;オフ・チャタリング中か?
BR    OFF_CHAT     ;オフ・チャタリング中
LD    WORK1, B_K_DAT1_H
LD    WORK2, B_K_DAT1_L
SET2  CMP, Z           ;前のキー・データと比較
SUB   WORK1, B_K_DAT2_H
SUBC  WORK2, B_K_DAT2_L
SKT1  Z
BR    DIFFER       ;不一致
;一致

```



```

OFF_CHAT:      ;オフ・チャタリング中
CLR3          OFFFC_F,K_CHANGE,K_ERROR      ;キー変化なし状態
RET
;
DIFFER:        ;無効 ON → 有効 ON
;
ST            B_K_DAT2_H,WORK1              ;今のキー・データを、そのまま前のキー・データとして設定
ST            B_K_DAT2_L,WORK2
MOV           CHAT_CNT,#0DH                 ;チャタリング回数の設定
SET1         ONC_F                          ;オン・チャタリング状態設定
CLR2         K_EX1ST,K_CHANGE
RET
;
EJECT         ;OFF→ON
CHAT1:        ;OFF→ON
;
LD            WORK1,B_K_DAT1_H              ;今のキー・データをそのまま前のキー・データとして設定
LD            WORK2,B_K_DAT1_L
ST            B_K_DAT2_H,WORK1
ST            B_K_DAT2_L,WORK2
MOV           CHAT_CNT,#0DH                 ;チャタリング回数の設定
SET1         ONC_F                          ;オン・チャタリング状態設定
CLR1         K_CHANGE
RET
;
OFF_CHA1:     ;オフ・チャタリング中か?
SKT1         OFFFC_F
BR            OFF_CHA2
ADD          CHAT_CNT,#1H
SKT1         CY                             ;オフ・チャタリング終了か?
RET
;
MOV           KEYDATA_H,#0FH                ;キー・オフ・データ設定
MOV           KEYDATA_L,#0FH
CLR4         ONC_F,OFFFC_F,K_EX1ST,K_ERROR ;オフ・チャタリング終了
SET1         K_CHANGE                       ;キー変化状態設定
RET
;
OFF_CHA2:     ;ON→OFF      無効 ON→OFF
;
MOV           CHAT_CNT,#0DH                 ;チャタリング回数の設定
SET1         OFFFC_F
CLR1         K_CHANGE                       ;オフ・チャタリング状態設定
RET
;
END

```


クロスレファレンス

AS17K Z1.10 D1 << E17201A XREF LIST >>

10:36:03 02/15/91 PAGE 02-001

PROG -

SOURCE = KEY.ASM

SYMBOL	TYPE	A	VALUE	/REF(#DEF)
B_K_DAT1_H	MEM	L	0.1A	/# 28 , 246 , 250 , 329 , 346 , 359 , 369 , 387 , 413
B_K_DAT1_L	MEM	L	0.1B	/# 29 , 260 , 267 , 273 , 277 , 291 , 330 , 347 , 360
B_K_DAT2_H	MEM	L	0.1C	/# 31 , 118 , 332 , 371 , 390 , 403 , 415
B_K_DAT2_L	MEM	L	0.1D	/# 32 , 119 , 333 , 372 , 391 , 404 , 416
CHANGE	LAB	L	70	/ 352 , # 357
CHAT	LAB	L	80	/ 328 , # 378
CHAT1	LAB	L	94	/ 380 , # 411
CHAT_CNT	WRM	L	0.12	/# 18 , 339 , 373 , 405 , 417 , 426 , 438
CHECK_F	FLG	L	0.1F.1	/# 50 , 121 , 121-2 , 121-2 , 220 , 220-1 , 220-1 , 222 , 222-1
CK_0	LAB	L	33	/ 251 , # 256
CK_1	LAB	L	37	/ 259 , # 263
CK_2	LAB	L	3B	/ 266 , # 270
CK_3	LAB	L	3F	/ 272 , # 275
DIFPER	LAB	L	8E	/ 393 , # 401
ERROR	LAB	L	1D	/# 211 , 221
K_CHANGE	FLG	L	0.1E.2	/# 44 , 121 , 121-1 , 121-1 , 215 , 215-2 , 215-2 , 323 , 323-2
				353 , 353-1 , 383 , 383-1 , 363-1 , 375 , 375-2 , 375-2 , 398
				398-2 , 398-2 , 407 , 407-1 , 419 , 419-1 , 419-1 , 433 , 433-1
				433-1 , 440 , 440-1 , 440-1
K_ERROR	FLG	L	0.1E.1	/# 45 , 121 , 121-1 , 215 , 216-1 , 323 , 323-2 , 323-2 , 353
				353-1 , 353-1 , 364 , 384-1 , 384-1 , 398 , 398-2 , 432 , 432-2
K_EXIST	FLG	L	0.1E.3	/# 48 , 215 , 216-1 , 216-1 , 318 , 318-1 , 318-1 , 344 , 344-1
				344-1 , 379 , 379-1 , 379-1 , 407 , 407-1 , 407-1 , 432 , 432-2
K_SCAN_C	MEM	L	0.20	/# 34 , 123 , 294
KEY1_FLG	MEM	L	0.1F	/# 47 , 48 , 49 , 50
KEY_FLG	MEM	L	0.1E	/# 42 , 43 , 44 , 45
KEY_RET	MEM	L	0.79	/# 38 , 191 , 229
KEYDATA_H	MEM	L	0.18	/# 25 , 116 , 214 , 349 , 361 , 430
KEYDATA_L	MEM	L	0.19	/# 26 , 117 , 213 , 350 , 362 , 431
KEYSCAN	LAB	P	1	/ 1-4 , # 113
KYBSET	LAB	L	40	/ 261 , 268 , 274 , # 279
OFF_CHA1	LAB	L	9C	/ 319 , # 423
OFF_CHA2	LAB	L	A7	/ 425 , # 436
OFF_CHAT	LAB	L	8B	/ 385 , # 397
OFF_F	FLG	P	0.1F.2	/ 1-6 , # 49 , 114 , 114-1 , 215 , 215-1 , 323 , 323-1 , 323-1
				375 , 375-1 , 375-1 , 384 , 384-1 , 384-1 , 398 , 398-1 , 398-1
				424 , 424-1 , 424-1 , 432 , 432-1 , 439 , 439-1 , 439-1
ON_CHAT	LAB	L	59	/ 314 , # 326
ON_CHAT1	LAD	L	77	/ 335 , # 367
ONC_F	FLG	P	0.1F.3	/ 1-5 , # 48 , 114 , 114-1 , 114-1 , 215 , 215-1 , 215-1 , 323
				323-1 , 327 , 327-1 , 327-1 , 345 , 345-1 , 345-1 , 374 , 374-1
				374-1 , 406 , 406-1 , 406-1 , 418 , 418-1 , 418-1 , 492 , 432-1
RET_CK	LAB	L	23	/ 199 , # 219
RETURN	MEM	L	0.13	/# 20 , 230 , 285
S_FOB	LAB	L	32	/ 249 , # 253
SCAN	LAB	L	47	/ 193 , # 290 , # 293
SCAN_END	LAB	L	52	/ 296 , # 312
SCANDAT_H	MEM	L	0.10	/# 15 , 126 , 136 , 234 , 299 , 304
SCANDAT_L	MEM	L	0.11	/# 15 , 125 , 137 , 235 , 300 , 306
SHIFT	LAB	L	42	/# 287 , 292

AS17K Z1.10 D1 << E17201A XREF LIST >> 10:36:03 02/15/91 PAGE 02-002

PROG -

SOURCE - KEY.ASM

SYMBOL TYPE A VALUE /REP (#DEF)

SOUR_SET	LAB	L	C	/#	134	.	306												
SOURCE1	MEM	L	0.71	/#	39	.	138												
SOURCE2	MEM	L	0.72	/#	40	.	139												
SOURCE_H	MEM	L	0.14	/#	22	.	236	,	248	,	254								
SOURCE_L	MEM	L	0.15	/#	23	.	237	,	247										
STAT_CLS	LAB	L	7	/	115	.	# 120												
TM_C	MEM	L	0.21	/#	36	.	185	,	188										
WAIT	LAB	L	11	/#	186	.	190												
WAIT_TM	DAT	L	D	/#	52	.	185												
WORK1	MEM	L	0.00	/#	12	.	116	,	118	,	136	,	138	,	191	,	192	,	197
						.	200	,	201	,	229	,	230	,	234	,	236	,	247
						.	265	,	271	,	285	,	288	,	299	,	301	,	301
						.	329	,	332	,	345	,	349	,	359	,	361	,	369
						.	390	,	403	,	413	,	415						
WORK2	MEM	L	0.01	/#	13	.	117	,	119	,	137	,	139	,	235	,	237	,	300
						.	303	,	305	,	330	,	333	,	347	,	350	,	360
						.	372	,	388	,	391	,	404	,	414	,	416		

TOTAL SYMBOLS - 52

END OF XREF LIST

第3章 LCD表示

LCDコントローラ／ドライバは、表示モード・レジスタ、144ビット（36×4ビット）の表示データ・メモリ（データ・メモリのバンク0：40H-5FHを使用）、タイミング・コントローラ、マルチプレクサ、基準電圧発生回路、LCD駆動電圧コントロール回路、セグメント・ドライバ、コモン・ドライバで構成されています。

表示モード・レジスタ（LCDコントローラ／ドライバの表示モードの選択、フレーム周波数の選択およびLCD表示出力のオン／オフ）を設定することにより、表示データ・メモリの内容に従ってLCDパネルを直接駆動します。

3.1 LCD仕様

ここで使用するLCDパネルの仕様は以下のとおりです。

☆ LCD表示仕様

〈使用LCD表示素子〉

LD-B8003： 4時分割

フレーム周波数 TYP=140 Hz

14セグメント表示

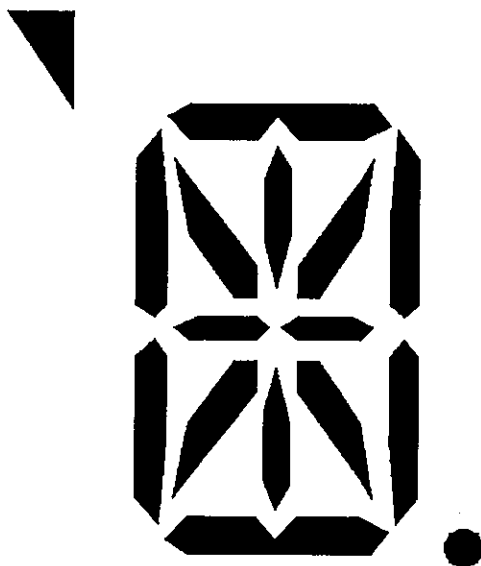


図 3-1 LCDパネル

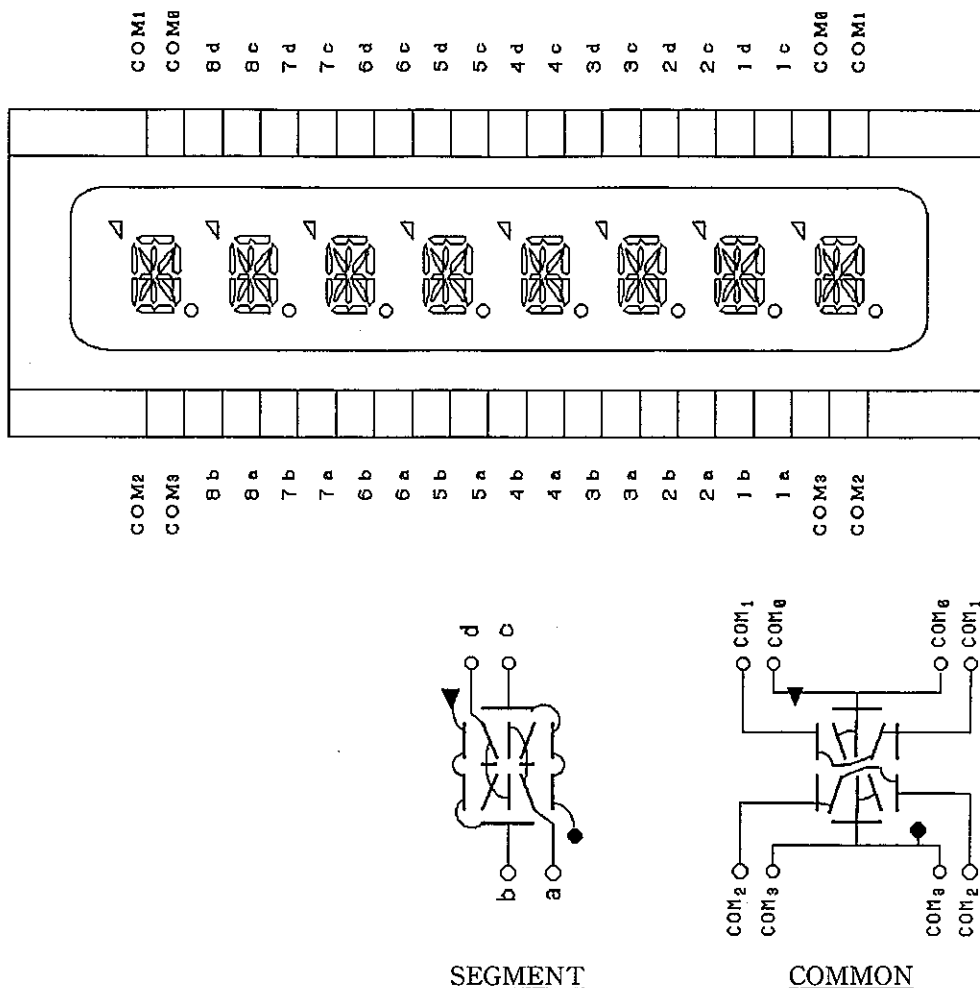


表 3-1 結線表 (LD-B88003- μ PD17201A)

パネル端子名	μ PD17201A端子名	パネル端子名	μ PD17201A端子名	パネル端子名	μ PD17201A端子名	パネル端子名	μ PD17201A端子名
1a	LCD ₀	2a	LCD ₄	3a	LCD ₈	4a	LCD ₁₂
1b	LCD ₃	2b	LCD ₇	3b	LCD ₁₁	4b	LCD ₁₅
1c	LCD ₂	2c	LCD ₆	3c	LCD ₁₀	4c	LCD ₁₄
1d	LCD ₁	2d	LCD ₅	3d	LCD ₉	4d	LCD ₁₃
パネル端子名	μ PD17201A端子名	パネル端子名	μ PD17201A端子名	パネル端子名	μ PD17201A端子名	パネル端子名	μ PD17201A端子名
5a	LCD ₁₆	6a	LCD ₂₀	7a	LCD ₂₄	8a	LCD ₂₈
5b	LCD ₁₉	6b	LCD ₂₃	7b	LCD ₂₇	8b	LCD ₃₁
5c	LCD ₁₈	6c	LCD ₂₂	7c	LCD ₂₆	8c	LCD ₃₀
5d	LCD ₁₇	6d	LCD ₂₁	7d	LCD ₂₅	8d	LCD ₂₉

注) COM₂とLCD₃₅, COM₃とLCD₃₄は同一端子です。

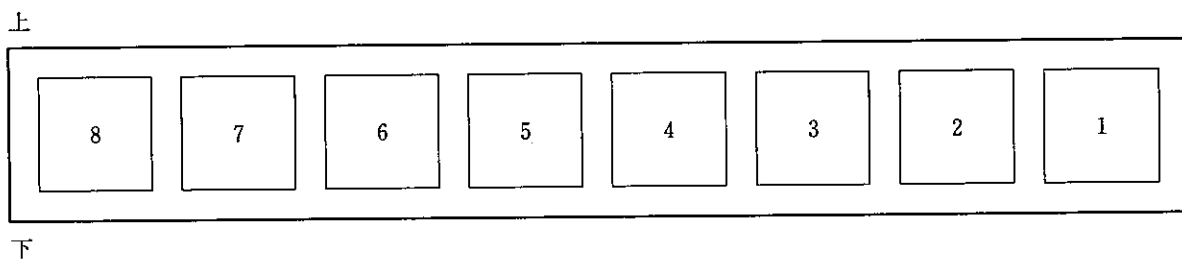
3.2 4時分割表示

4時分割表示LCDパネルの場合のプログラムを紹介します。

3.2.1 表示データ

LCDパネル上の表示位置を図3-2に示します。

図3-2 LCD表示位置



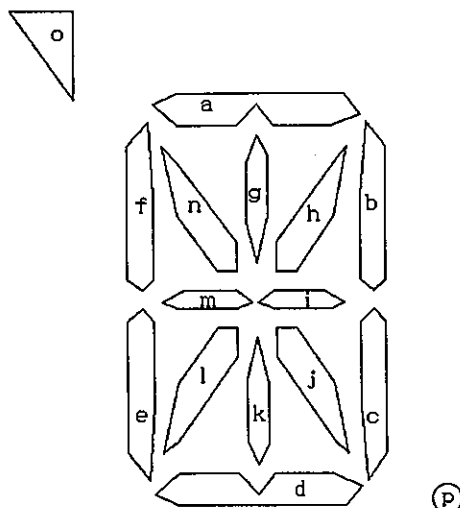
表示データ・メモリの割り付けを表3-2に示します。

表3-2 表示データ・メモリ (バンク0)

表示位置	表示データ・メモリ	表示位置	表示データ・メモリ
1	40H, 41H, 42H, 43H	5	50H, 51H, 52H, 53H
2	44H, 45H, 46H, 47H	6	54H, 55H, 56H, 57H
3	48H, 49H, 4AH, 4BH	7	58H, 59H, 5AH, 5BH
4	4CH, 4DH, 4EH, 4FH	8	5CH, 5DH, 5EH, 5FH

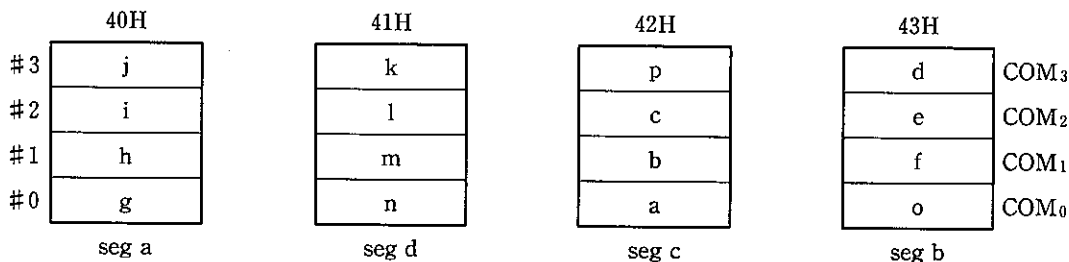
各セグメントを図3-3に示します。

図3-3 セグメント



前頁で示した各セグメントは以下のように設定します。

LCD表示位置1の場合



注) 他の表示位置については、対応データ・メモリ・アドレスが変わるだけで、ビット設定は変化しません。

このときの表示データを表3-3に示します。

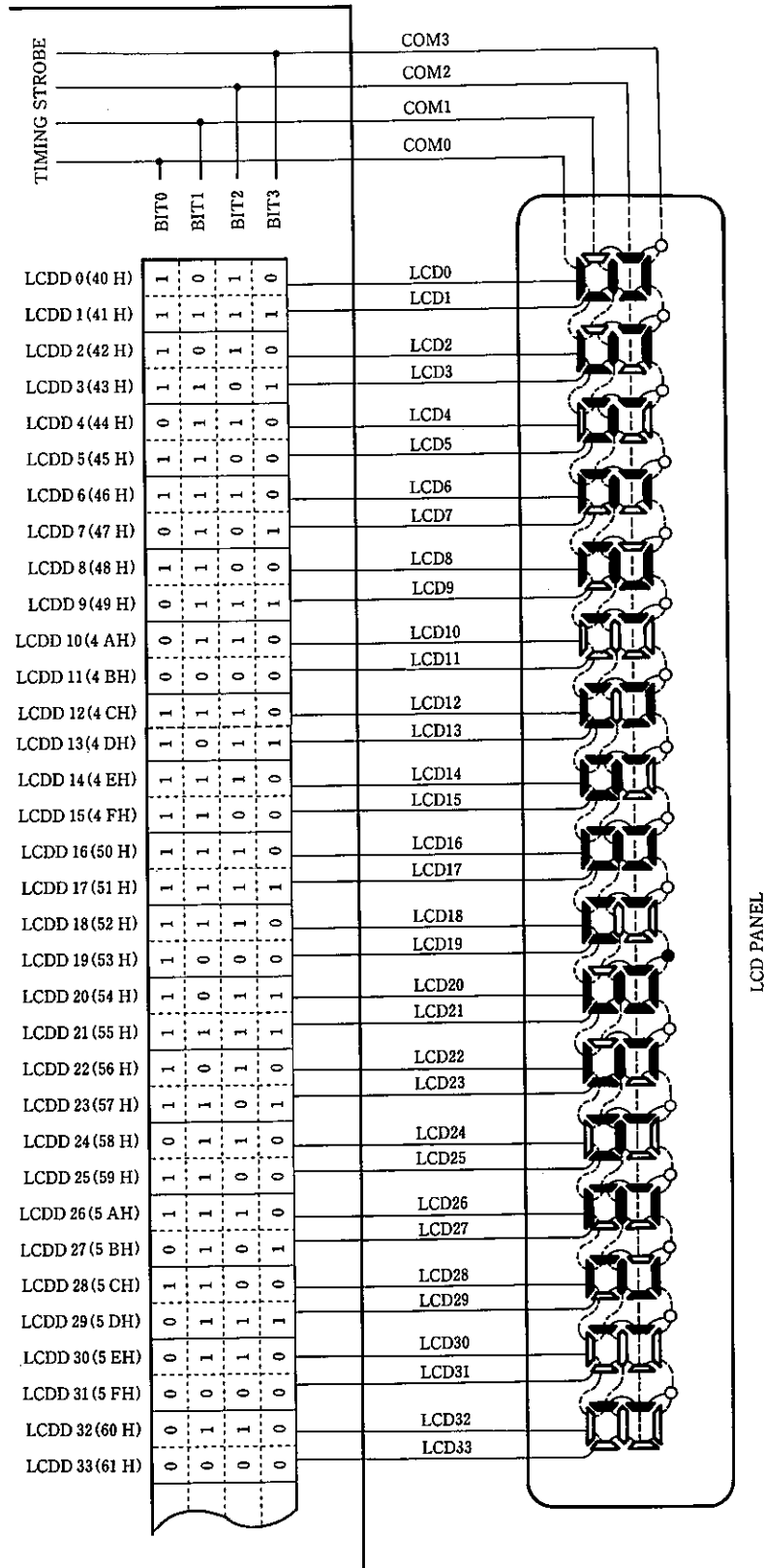
表3-3 表示データ

COM SEG Disp	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	Hex Data
	j	i	h	g	k	l	m	n	p	c	b	a	d	e	f	o	
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	007E
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0060
2	0	1	0	0	0	0	1	0	0	0	1	1	1	1	0	0	423C
3	0	1	0	0	0	0	1	0	0	1	1	1	1	0	0	0	4278
4	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	4262
5	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	425A
6	0	1	0	0	0	0	1	0	0	1	0	1	1	1	1	0	425E
7	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0072
8	0	1	0	0	0	0	1	0	0	1	1	1	1	1	1	0	427E
9	0	1	0	0	0	0	1	0	0	1	1	1	1	0	1	0	427A
A	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	0	4276
B	0	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	5878
C	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	001E
D	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1878
E	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	421E
F	0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	0	4216

3.2.2 4時分割LCDパネル

4時分割LCDパネルの結線例を図3-4に示します。

図3-4 4時分割LCDパネルの結線例



<使用するハードウェア>

番号	端子名	使用説明
76	COM ₀	LCDドライバのコモン信号とセグメント信号の出力です。 セグメント信号出力とコモン信号出力の切り替えは、レジスタ・ファイルのLCDMD3-LCDMD0によって行います。
77	COM ₁	
78	LCD ₃₅ /COM ₂	
79	LCD ₃₄ /COM ₃	
80	LCD ₃₃	・COM ₀ -COM ₃ : LCDドライバのコモン信号出力
1	LCD ₃₂	・LCD ₃₅ -LCD ₀ : LCDドライバのセグメント信号出力
32	LCD ₁	
34	LCD ₀	

<初期設定>

表示モード : 3 (LCDMD3=0, LCDMD2=0, LCDMD1=1, LCDMD0=1)

LCD表示 : オン (LCDEN=1)

フレーム周波数 : 64 Hz (LCDCK2=0, LCDCK1=1, LCDCK0=1)

<起動方法>

表示データ・エリア (LCDDAT1-LCDDAT7) に表示する値を設定し、LCDDISPをCALLすることによりセグメント・データが表示データ・メモリに書き込まれます。ただし、実際にLCDパネルを駆動するためには、表示モード・レジスタ (LCDMD3-LCDMD0) と、LCD表示のオン/オフ (LCDEN)、フレーム周波数 (LCDCK2-LCDCK0) を設定しておく必要があります。

ドキュメント

AS17K Z1.10 D1 << E17201A DOCUMENT >> 13:30:34 02/19/91

TABLE OF CONTENTS

	PAGE
第 3 章 LCD表示	1
<プログラム説明>	1
<RAM使用説明>	3
<キーボード設定>	3
<表示位置と表示データ・メモリ (バンク0の40H-5FH) の関係>	3

第 3 章 LCD表示

```
PUBLIC (MEM) : LCDDAT1, LCDDAT2, LCDDAT3, LCDDAT4, LCDDAT5, LCDDAT6, LCDDAT7, LCDDAT8, WORK1
PUBLIC (LAB) : LCD表示
ADDR RANGE : 0000H - 005AH
```

<プログラム説明>
表示モード・レジスタを設定し、表示データを表示データ・メモリ（バンク0の40H-5FH）に設定することにより、LCD表示することができます。表示するプログラム例では、表示データ・メモリに設定する表示データをテーブル・データにし、4時分割表示LCDパネルの場合のプログラムを紹介します。

```
ENTRANCES :-
MEMORIES CHANGED :-
MEMORIES REFERRED :-
MEMORIES MANIPULATED :-
FLAGS CHANGED :-
FLAGS REFERRED :-
DATA REFERRED :-
BRANCH TO :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL :-
```

<RAM使用説明>

```
WORK1 ..... 0, 00H ..... 汎用レジスタです。
WORK2 ..... 0, 01H ..... 汎用レジスタです。
WORK3 ..... 0, 02H ..... 汎用レジスタです。
WORK4 ..... 0, 03H ..... 汎用レジスタです。
WORK5 ..... 0, 04H ..... 汎用レジスタです。
WORK6 ..... 0, 05H ..... 汎用レジスタです。
WORK7 ..... 0, 06H ..... 汎用レジスタです。
WORK8 ..... 0, 07H ..... 汎用レジスタです。
LCDDAT1 ..... 0, 10H ..... 書きたい数文字用のエリアです。
LCDDAT2 ..... 0, 11H ..... 書きたい数文字用のエリアです。
LCDDAT3 ..... 0, 12H ..... 書きたい数文字用のエリアです。
LCDDAT4 ..... 0, 13H ..... 書きたい数文字用のエリアです。
LCDDAT5 ..... 0, 14H ..... 書きたい数文字用のエリアです。
LCDDAT6 ..... 0, 15H ..... 書きたい数文字用のエリアです。
LCDDAT7 ..... 0, 16H ..... 書きたい数文字用のエリアです。
LCDDAT8 ..... 0, 17H ..... 書きたい数文字用のエリアです。
```


LCD0	0	4 0 H	1	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD1	0	4 1 H	1	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD2	0	4 2 H	1	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD3	0	4 3 H	1	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD4	0	4 4 H	2	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD5	0	4 5 H	2	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD6	0	4 6 H	2	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD7	0	4 7 H	2	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD8	0	4 8 H	3	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD9	0	4 9 H	3	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD10	0	4 A H	3	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD11	0	4 B H	3	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD12	0	4 C H	4	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD13	0	4 D H	4	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD14	0	4 E H	4	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD15	0	4 F H	4	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD16	0	5 0 H	5	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD17	0	5 1 H	5	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD18	0	5 2 H	5	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD19	0	5 3 H	5	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD20	0	5 4 H	6	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD21	0	5 5 H	6	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD22	0	5 6 H	6	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD23	0	5 7 H	6	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD24	0	5 8 H	7	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD25	0	5 9 H	7	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD26	0	5 A H	7	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD27	0	5 B H	7	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD28	0	5 C H	8	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD29	0	5 D H	8	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD30	0	5 E H	8	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す
LCD31	0	5 F H	8	の	置	表	示	せ	た	め	表	デ	タ	メ	モ	リ	で	す

ENTRANCES :-
 MEMORIES CHANGED :-
 MEMORIES REFERRED :-
 MEMORIES MANIPULATED :-
 FLAGS CHANGED :-
 FLAGS REFERRED :-
 DATA REFERRED :-
 BRANCH TO :-
 SUBROUTINES CALLED :-
 LABELS MANIPULATED :-
 SYSTEM CALL :-

<テーブル・データ>

0-Fを表示させるデータ(16ビット)を、10H番地から順番に設定します。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

<表示データ設定>

はじめに、アドレス・レジスタを10H番地に設定しておきます。
次に、表示させたい数文字(LCDDAT1-LCDDAT8)をいったんWORK1-WORK8に格納します。
その値を、アドレス・レジスタに計算した値が、書きたい数文字の表示データの位置(アドレス)を示します。
その表示データをDBFを介して表示データ・メモリにいきます。
こうして、LCDパネルに表示することができます。

```

ENTRANCES          :LCD表示
MEMORIES CHANGED   :WORK1          WORK2
                   :WORK3          WORK4
                   :WORK5          WORK5
                   :WORK7          WORK8
MEMORIES REFERRED  :LCDDAT1        LCDDAT2
                   :LCDDAT3        LCDDAT4
                   :LCDDAT5        LCDDAT5
                   :LCDDAT7        LCDDAT8
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

<表示位置と表示データ・メモリ(バンク0の40H-5FH)の関係>

- 1の位置に表示する場合.....表示データ・メモリは40H-43H
- 2の位置に表示する場合.....表示データ・メモリは44H-47H
- 3の位置に表示する場合.....表示データ・メモリは48H-4BH
- 4の位置に表示する場合.....表示データ・メモリは4CH-4FH
- 5の位置に表示する場合.....表示データ・メモリは50H-53H
- 6の位置に表示する場合.....表示データ・メモリは54H-57H
- 7の位置に表示する場合.....表示データ・メモリは58H-5BH
- 8の位置に表示する場合.....表示データ・メモリは5CH-5FH

```

ENTRANCES          :-
MEMORIES CHANGED  :LCD0          LCD1
                  :LCD10         LCD11
                  :LCD12         LCD13
                  :LCD14         LCD15
                  :LCD16         LCD17
                  :LCD18         LCD19
                  :LCD2          LCD20
                  :LCD21         LCD22
                  :LCD23         LCD24
                  :LCD25         LCD26
                  :LCD27         LCD28
                  :LCD29         LCD3
                  :LCD30         LCD31
                  :LCD4          LCD5
                  :LCD6          LCD7
                  :LCD8          LCD9
MEMORIES REFERRED  :WORK1         WORK2
                  :WORK3         WORK4
                  :WORK5         WORK6
                  :WORK7         WORK8
MEMORIES MANIPULATED :-
FLAGS CHANGED     :-
FLAGS REFERRED    :-
DATA REFERRED     :-
BRANCH TO         :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL       :-
    
```


RAMマップ

第3章 LCD表示

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 13:30:23 02/19/91 PAGE 02-001

PROG -

SOURCE - LCD.ASM

(PUBLIC/LOCAL MAP)

BANK - 0

		ROW							
		7	6	5	4	3	2	1	0
I	LCDD32	LCDD16	LCDD0				LCDDAT1	WORK1	0
		LCDD16	LCDD0						
I	LCDD33	LCDD17	LCDD1				LCDDAT2	WORK2	1
		LCDD17	LCDD1					WORK2	
I	LCDD34	LCDD18	LCDD2				LCDDAT3	WORK3	2
		LCDD18	LCDD2					WORK3	
I	LCDD35	LCDD19	LCDD3				LCDDAT4	WORK4	3
		LCDD19	LCDD3					WORK4	
IAR3		LCDD20	LCDD4				LCDDAT5	WORK5	4
		LCDD20	LCDD4					WORK5	
IAR2		LCDD21	LCDD5				LCDDAT6	WORK6	5
		LCDD21	LCDD5					WORK6	
IAR1		LCDD22	LCDD6				LCDDAT7	WORK7	6
		LCDD22	LCDD6					WORK7	
IAR0		LCDD23	LCDD7				LCDDAT8	WORK8	7
		LCDD23	LCDD7					WORK8	
IWR		LCDD24	LCDD8						8
		LCDD24	LCDD8						
IBANK		LCDD25	LCDD9						9
		LCDD25	LCDD9						
IIXH IMPH		LCDD26	LCDD10						A
		LCDD26	LCDD10						
IIXM IMPL		LCDD27	LCDD11						B
		LCDD27	LCDD11						

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 13:30:23 02/19/91 PAGE 02-002

PROG *

SOURCE = LCD.ASM

(PUBLIC/LOCAL MAP)

BANK = 0

ROW

	7	6	5	4	3	2	1	0	
IXL			LCDD28	LCDD12				DBF3	C
			LCDD28	LCDD12					
IRPH			LCDD29	LCDD13				DBF2	D
			LCDD29	LCDD13					
IRPL			LCDD30	LCDD14				DBF1	E
			LCDD30	LCDD14					
IFSW			LCDD31	LCDD15				DBF0	F
			LCDD31	LCDD15					

AS17K 21.10 D1 << E17201A FLG MAP >> 19:30:23 02/19/91 PAGE 02-003

PROG -

SOURCE - LCD.ASM

(FLAG MAP) BANK = 0

MSB	3	2	1	0	LSB
70	IP0A3	IP0A2	IP0A1	IP0A0	
71	IP0B3	IP0B2	IP0B1	IP0B0	
72	IP0C3	IP0C2	IP0C1	IP0C0	
73	IP0D3	IP0D2	IP0D1	IP0D0	
7A	IP0E				
7E				IBCD	
7F	ICMP	ICY	IZ	IXE	
82			ISYSCK	IXEN	
83	IWDRES	IWTMD	IWMRES		
87	IVDDDET1	IVDDDET0			
8F				IINT	
91				INRZBF	
92				INRZ	
A0				IADCCMP	
A1	IVREFEN	IADCEN	IADCCH1	IADCCH0	
A2	ISLOTS	ISIOHIZ	SILOCK1	SILOCK0	
A3		INRZEN	ITMOE	ISIOEN	

AS17K Z1.10 D1 << E17201A FLG MAP >> 13:30:23 02/19/91 PAGE 02-004

PROG -

SOURCE = LCD.ASM

(FLAG MAP) BANK = 0

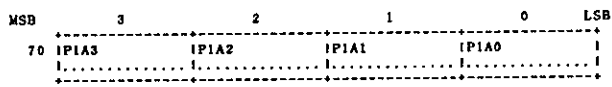
MSB	3	2	1	0	LSB
A7	!PODBIO3	!PODBIO2	!PODBIO1	!PODBIO0	
AF	!IPSIO	!IPWTM	!IP	!IPTM	
B1	!LCDEN	!LCDCK2	!LCDCK1	!LCDCK0	
B2	!LCDMD3	!LCDMD2	!LCDMD1	!LCDMD0	
B3	!TMEN	!TMRES	!TMCK1	!TMCK0	
B7	!PIAGIO	!POCGIO	!POBGIO	!POAGIO	
BB	!IRQSIO				
BC		!IRQTM			
BD			!IRQ		
BE				!IRQTM	

AS17K Z1.10 D1 << E17201A FLG MAP >> 13:30:23 02/19/91 PAGE 02-005

PROG -

SOURCE - LCD.ASM

(FLAG MAP) BANK - 1





RAM使用説明

AS17K Z1.10 DI << E17201A DATA MEMORY MAP >> 13:30:23 02/19/91 PAGE 02-007

PROG -

SOURCE - LCD.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCD3	MEM	0	43		1の位に置き換
LCD30	MEM	0	5E		8の位に置き換
LCD31	MEM	0	5F		8の位に置き換
LCD4	MEM	0	44		2の位に置き換
LCD5	MEM	0	45		2の位に置き換
LCD6	MEM	0	46		2の位に置き換
LCD7	MEM	0	47		2の位に置き換
LCD8	MEM	0	48		3の位に置き換
LCD9	MEM	0	49		3の位に置き換
LCDDK0	FLG	0	B1	...1	
LCDDK1	FLG	0	B1	..1.	
LCDDK2	FLG	0	B1	.1..	
LCDD0	MEM	0	40		
LCDD1	MEM	0	41		
LCDD10	MEM	0	4A		
LCDD11	MEM	0	4B		
LCDD12	MEM	0	4C		
LCDD13	MEM	0	4D		
LCDD14	MEM	0	4E		
LCDD15	MEM	0	4F		
LCDD16	MEM	0	50		
LCDD17	MEM	0	51		
LCDD18	MEM	0	52		
LCDD19	MEM	0	53		
LCDD2	MEM	0	42		
LCDD20	MEM	0	54		
LCDD21	MEM	0	55		
LCDD22	MEM	0	56		
LCDD23	MEM	0	57		
LCDD24	MEM	0	58		
LCDD25	MEM	0	59		
LCDD26	MEM	0	5A		
LCDD27	MEM	0	5B		
LCDD28	MEM	0	5C		
LCDD29	MEM	0	5D		
LCDD3	MEM	0	43		
LCDD30	MEM	0	5E		
LCDD31	MEM	0	5F		
LCDD32	MEM	0	60		
LCDD33	MEM	0	61		
LCDD34	MEM	0	62		
LCDD35	MEM	0	63		
LCDD4	MEM	0	44		
LCDD5	MEM	0	45		
LCDD6	MEM	0	46		
LCDD7	MEM	0	47		
LCDD8	MEM	0	48		
LCDD9	MEM	0	49		
LCDDAT1	MEM	0	10		書きたい数字用のエリアです。
LCDDAT2	MEM	0	11		書きたい数字用のエリアです。
LCDDAT3	MEM	0	12		書きたい数字用のエリアです。
LCDDAT4	MEM	0	13		書きたい数字用のエリアです。

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 13:30:23 02/19/91 PAGE 02-008

PROG -

SOURCE - LCD.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCDDAT5	MEM	0	14		書きたい数文字用のエリアです。
LCDDAT6	MEM	0	15		書きたい数文字用のエリアです。
LCDDAT7	MEM	0	16		書きたい数文字用のエリアです。
LCDDAT8	MEM	0	17		書きたい数文字用のエリアです。
LCDEN	FLG	0	B1	1...	
LCDM00	FLG	0	B2	...1	
LCDM01	FLG	0	B2	...1	
LCDM02	FLG	0	B2	...1	
LCDM03	FLG	0	B2	...1	
MPE	FLG	0	7A	1...	
MPH	MEM	0	7A		
MPL	MEM	0	7B		
NRZ	FLG	0	92	...1	
NRZBF	FLG	0	91	...1	
NRZEN	FLG	0	A3	...1	
POA0	FLG	0	70	...1	
POA1	FLG	0	70	...1	
POA2	FLG	0	70	...1	
POA3	FLG	0	70	...1	
POAGIO	FLG	0	B7	...1	
POB0	FLG	0	71	...1	
POB1	FLG	0	71	...1	
POB2	FLG	0	71	...1	
POB3	FLG	0	71	...1	
POBGIO	FLG	0	B7	...1	
POC0	FLG	0	72	...1	
POC1	FLG	0	72	...1	
POC2	FLG	0	72	...1	
POC3	FLG	0	72	...1	
POCGIO	FLG	0	B7	...1	
POD0	FLG	0	73	...1	
POD1	FLG	0	73	...1	
POD2	FLG	0	73	...1	
POD3	FLG	0	73	...1	
PODBIO0	FLG	0	A7	...1	
PODBIO1	FLG	0	A7	...1	
PODBIO2	FLG	0	A7	...1	
PODBIO3	FLG	0	A7	...1	
PIA0	FLG	1	70	...1	
PIA1	FLG	1	70	...1	
PIA2	FLG	1	70	...1	
PIA3	FLG	1	70	...1	
PIAGIO	FLG	0	B7	...1	
PSW	MEM	0	7F		
RPH	MEM	0	7D		
RPL	MEM	0	7E		
SIOCK0	FLG	0	A2	...1	
SIOCK1	FLG	0	A2	...1	
SIOEN	FLG	0	A3	...1	
SIOHIZ	FLG	0	A2	...1	
SLOTS	FLG	0	A2	...1	
SYSCK	FLG	0	82	...1	

AS17K Z1.10 DI << E17201A DATA MEMORY MAP >> 13:30:23 02/19/91 PAGE 02-009

PROG -

SOURCE - LCD.ASM

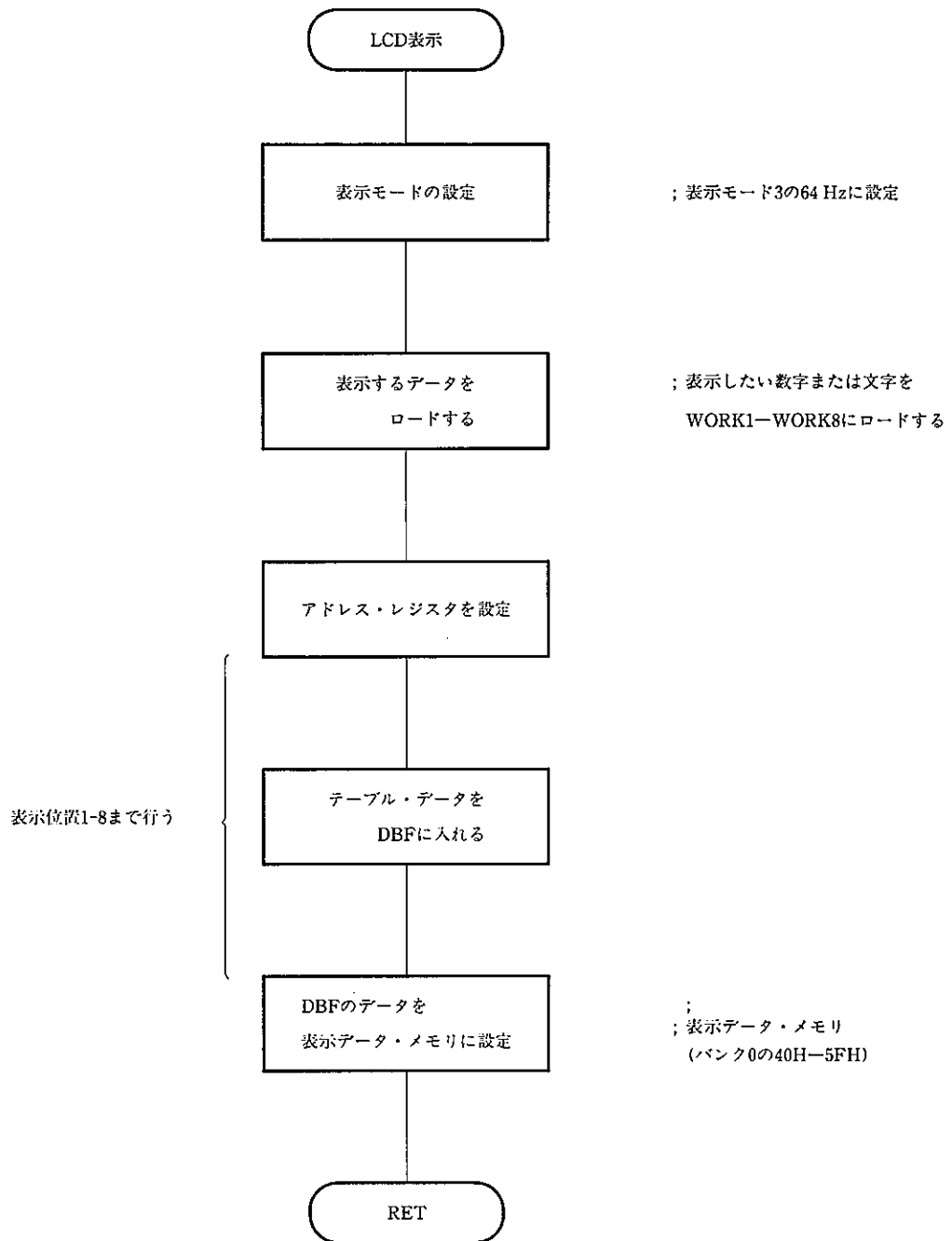
(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
TMCK0	FLG	0	B3	...1	
TMCK1	FLG	0	B3	...1	
TMBN	FLG	0	B3	1...	
TMOE	FLG	0	A3	...1	
TMRES	FLG	0	B3	...1	
VDDDET0	FLG	0	B7	...1	
VDDDET1	FLG	0	B7	...1	
VREFEN	FLG	0	A1	1...	
WDTRES	FLG	0	B3	1...	
WORK1	MEM	0	0		汎用レジスタです。
WORK2	MEM	0	1		汎用レジスタです。
WORK3	MEM	0	2		汎用レジスタです。
WORK4	MEM	0	3		汎用レジスタです。
WORK5	MEM	0	4		汎用レジスタです。
WORK6	MEM	0	5		汎用レジスタです。
WORK7	MEM	0	6		汎用レジスタです。
WORK8	MEM	0	7		汎用レジスタです。
WR	MEM	0	7B		
WTMMD	FLG	0	B3	...1	
WTMRES	FLG	0	B3	...1	
XEN	FLG	0	B2	...1	
Z	FLG	0	7F	...1	

フロー・チャート



フロー・チャート



リスト

INCLUDE 'LCD.PUB'
SUMMARY '%', '第 3 章 LCD表示'

%%
SUMMARY '%', '<プログラム説明>
表示モード・レジスタを指定し、表示データを表示データ・メモリ（バンク0の40H-5FH）に設定することにより、LCD
を表示するプログラムを、表示データ・メモリに設定する表示データをテーブル・データにし、4時分表示LCDパネルの場合の
プログラムを紹介します。

.....
:*** レジスタ ***
.....

WORK1	MEM	0.00H	:汎用レジスタです。
WORK2	MEM	0.01H	:汎用レジスタです。
WORK3	MEM	0.02H	:汎用レジスタです。
WORK4	MEM	0.03H	:汎用レジスタです。
WORK5	MEM	0.04H	:汎用レジスタです。
WORK6	MEM	0.05H	:汎用レジスタです。
WORK7	MEM	0.06H	:汎用レジスタです。
WORK8	MEM	0.07H	:汎用レジスタです。

.....
:*** LCDの表示データ・メモリ ***
.....

LCD0	MEM	0.40H	:1の位置に表示されるための表示データ・メモリです。
LCD1	MEM	0.41H	:1の位置に表示されるための表示データ・メモリです。
LCD2	MEM	0.42H	:1の位置に表示されるための表示データ・メモリです。
LCD3	MEM	0.43H	:1の位置に表示されるための表示データ・メモリです。
LCD4	MEM	0.44H	:2の位置に表示されるための表示データ・メモリです。
LCD5	MEM	0.45H	:2の位置に表示されるための表示データ・メモリです。
LCD6	MEM	0.46H	:2の位置に表示されるための表示データ・メモリです。
LCD7	MEM	0.47H	:2の位置に表示されるための表示データ・メモリです。
LCD8	MEM	0.48H	:3の位置に表示されるための表示データ・メモリです。
LCD9	MEM	0.49H	:3の位置に表示されるための表示データ・メモリです。
LCD10	MEM	0.4AH	:3の位置に表示されるための表示データ・メモリです。
LCD11	MEM	0.4BH	:3の位置に表示されるための表示データ・メモリです。
LCD12	MEM	0.4CH	:4の位置に表示されるための表示データ・メモリです。
LCD13	MEM	0.4DH	:4の位置に表示されるための表示データ・メモリです。
LCD14	MEM	0.4EH	:4の位置に表示されるための表示データ・メモリです。
LCD15	MEM	0.4FH	:4の位置に表示されるための表示データ・メモリです。
LCD16	MEM	0.50H	:5の位置に表示されるための表示データ・メモリです。
LCD17	MEM	0.51H	:5の位置に表示されるための表示データ・メモリです。
LCD18	MEM	0.52H	:5の位置に表示されるための表示データ・メモリです。
LCD19	MEM	0.53H	:5の位置に表示されるための表示データ・メモリです。
LCD20	MEM	0.54H	:6の位置に表示されるための表示データ・メモリです。
LCD21	MEM	0.55H	:6の位置に表示されるための表示データ・メモリです。
LCD22	MEM	0.56H	:6の位置に表示されるための表示データ・メモリです。
LCD23	MEM	0.57H	:6の位置に表示されるための表示データ・メモリです。
LCD24	MEM	0.58H	:7の位置に表示されるための表示データ・メモリです。
LCD25	MEM	0.59H	:7の位置に表示されるための表示データ・メモリです。
LCD26	MEM	0.5AH	:7の位置に表示されるための表示データ・メモリです。
LCD27	MEM	0.5BH	:7の位置に表示されるための表示データ・メモリです。
LCD28	MEM	0.5CH	:8の位置に表示されるための表示データ・メモリです。
LCD29	MEM	0.5DH	:8の位置に表示されるための表示データ・メモリです。
LCD30	MEM	0.5EH	:8の位置に表示されるための表示データ・メモリです。
LCD31	MEM	0.5FH	:8の位置に表示されるための表示データ・メモリです。

LCDDAT1 MEM	0. 10H	書きたい数字用
LCDDAT2 MEM	0. 11H	書きたい数字用
LCDDAT3 MEM	0. 12H	書きたい数字用
LCDDAT4 MEM	0. 13H	書きたい数字用
LCDDAT5 MEM	0. 14H	書きたい数字用
LCDDAT6 MEM	0. 15H	書きたい数字用
LCDDAT7 MEM	0. 16H	書きたい数字用
LCDDAT8 MEM	0. 17H	書きたい数字用

SUMMARY '%%'. <RAM使用説明>

WORK1	0. 00H	汎用レジスタです。
WORK2	0. 01H	汎用レジスタです。
WORK3	0. 02H	汎用レジスタです。
WORK4	0. 03H	汎用レジスタです。
WORK5	0. 04H	汎用レジスタです。
WORK6	0. 05H	汎用レジスタです。
WORK7	0. 06H	汎用レジスタです。
WORK8	0. 07H	汎用レジスタです。
LCDDAT1	0. 10H	書きたい数字用のエリアです。
LCDDAT2	0. 11H	書きたい数字用のエリアです。
LCDDAT3	0. 12H	書きたい数字用のエリアです。
LCDDAT4	0. 13H	書きたい数字用のエリアです。
LCDDAT5	0. 14H	書きたい数字用のエリアです。
LCDDAT6	0. 15H	書きたい数字用のエリアです。
LCDDAT7	0. 16H	書きたい数字用のエリアです。
LCDDAT8	0. 17H	書きたい数字用のエリアです。
LCD0	0. 40H	1の位置に表示させるための表示データメモリです。
LCD1	0. 41H	1の位置に表示させるための表示データメモリです。
LCD2	0. 42H	1の位置に表示させるための表示データメモリです。
LCD3	0. 43H	1の位置に表示させるための表示データメモリです。
LCD4	0. 44H	2の位置に表示させるための表示データメモリです。
LCD5	0. 45H	2の位置に表示させるための表示データメモリです。
LCD6	0. 46H	2の位置に表示させるための表示データメモリです。
LCD7	0. 47H	2の位置に表示させるための表示データメモリです。
LCD8	0. 48H	3の位置に表示させるための表示データメモリです。
LCD9	0. 49H	3の位置に表示させるための表示データメモリです。
LCD10	0. 4AH	3の位置に表示させるための表示データメモリです。
LCD11	0. 4BH	3の位置に表示させるための表示データメモリです。
LCD12	0. 4CH	4の位置に表示させるための表示データメモリです。
LCD13	0. 4DH	4の位置に表示させるための表示データメモリです。
LCD14	0. 4EH	4の位置に表示させるための表示データメモリです。
LCD15	0. 4FH	4の位置に表示させるための表示データメモリです。
LCD16	0. 50H	5の位置に表示させるための表示データメモリです。
LCD17	0. 51H	5の位置に表示させるための表示データメモリです。

LCD18	0	5 2 H	5	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD19	0	5 3 H	5	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD20	0	5 4 H	6	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD21	0	5 5 H	6	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD22	0	5 8 H	6	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD23	0	5 7 H	6	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD24	0	5 8 H	7	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD25	0	5 9 H	7	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD26	0	5 A H	7	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD27	0	5 B H	7	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD28	0	5 C H	8	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD29	0	5 D H	8	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD30	0	5 E H	8	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。
LCD31	0	5 F H	8	の位置に	表示させ	るため	の表示デ	ータ・メ	モリで	です。

.EJ-1

```

%%
;.....
;...   LCD表示   ...
;.....
;...   テーブル・データ   ...
;.....

```

SUMMARY '%%', '<テーブル・データ>'
0-Fを表示させるデータ(10ビット)を、10H番地から順番に設定します。

%%

ORG 10H

```

DW 00000000111110B ;0
DW 00000000110000B ;1
DW 01000100011100B ;2
DW 01000100111100B ;3
DW 01000100110001B ;4
DW 01000100101101B ;5
DW 01000100101110B ;6
DW 00000000110010B ;7
DW 01000100111110B ;8
DW 01000100111010B ;9
DW 01000100111010B ;A
DW 01011000111000B ;B
DW 00000000001111B ;C
DW 00011000111000B ;D
DW 01000100001111B ;E
DW 01000100001011B ;F

```

```

;.....
;...   LCD表示:   ...
;.....

```

SUMMARY '%%', '<表示データ設定>'
はじめに、アドレス・レジスタを10H番地に設定しておきます。
次に、表示させたい数文字(LCDDAT1-LCDDAT8)をいったんWORK1-WORK8に格納します。
その値を、アドレス・レジスタに加算した値が、書きたい数文字の表示データの位置(アドレス)を示します。
その表示データをDBFを介して表示データ・メモリにいれます。
こうして、LCDパネルに表示することができます。

%%

```

MOV AR0,#0
MOV AR1,#1
LD WORK1,LCDDAT1 ;0-F(HEX)
LD WORK2,LCDDAT2 ;0-F(HEX)

```

```
LD   WORK3, LCDDAT3 ;0-F (HEX)
LD   WORK4, LCDDAT4 ;0-F (HEX)
LD   WORK5, LCDDAT5 ;0-F (HEX)
LD   WORK6, LCDDAT6 ;0-F (HEX)
LD   WORK7, LCDDAT7 ;0-F (HEX)
LD   WORK8, LCDDAT8 ;0-F (HEX)
```

SUMMARY '%%', <表示位置と表示データ・メモリ(バンク0の40H-5FH)の関係>

- 1の位置に表示する場合・・・表示データ・メモリは40H-43H
- 2の位置に表示する場合・・・表示データ・メモリは44H-47H
- 3の位置に表示する場合・・・表示データ・メモリは48H-4BH
- 4の位置に表示する場合・・・表示データ・メモリは4CH-4FH
- 5の位置に表示する場合・・・表示データ・メモリは50H-53H
- 6の位置に表示する場合・・・表示データ・メモリは54H-57H
- 7の位置に表示する場合・・・表示データ・メモリは58H-5BH
- 8の位置に表示する場合・・・表示データ・メモリは5CH-5FH

%%

```
ST   AR0, WORK1
MOVT DBF, @AR

ST   LCD0, DBF3 ;1の位置に表示する
ST   LCD1, DBF2
ST   LCD2, DBF1
ST   LCD3, DBF0

ST   AR0, WORK2
MOVT DBF, @AR

ST   LCD4, DBF3 ;2の位置に表示する
ST   LCD5, DBF2
ST   LCD6, DBF1
ST   LCD7, DBF0

ST   AR0, WORK3
MOVT DBF, @AR

ST   LCD8, DBF3 ;3の位置に表示する
ST   LCD9, DBF2
ST   LCD10, DBF1
ST   LCD11, DBF0

ST   AR0, WORK4
MOVT DBF, @AR

ST   LCD12, DBF3 ;4の位置に表示する
ST   LCD13, DBF2
ST   LCD14, DBF1
ST   LCD15, DBF0

ST   AR0, WORK5
MOVT DBF, @AR

ST   LCD16, DBF3 ;5の位置に表示する
ST   LCD17, DBF2
ST   LCD18, DBF1
ST   LCD19, DBF0
```

```

ST      ARO, WORK6
MOVT    DBF, @AR

ST      LCD20, DBF3      ; 6 の位置に表示する
ST      LCD21, DBF2
ST      LCD22, DBF1
ST      LCD23, DBF0

ST      ARO, WORK7
MOVT    DBF, @AR

ST      LCD24, DBF3      ; 7 の位置に表示する
ST      LCD25, DBF2
ST      LCD26, DBF1
ST      LCD27, DBF0

ST      ARO, WORK8
MOVT    DBF, @AR

ST      LCD28, DBF3      ; 8 の位置に表示する
ST      LCD29, DBF2
ST      LCD30, DBF1
ST      LCD31, DBF0

RET
:
END

```


クロスレファレンス

AS17K Z1.10 D1 << R17201A XREF LIST >> 13:30:22 02/19/91 PAGE 02-001

PROG -

SOURCE - LCD.ASM

SYMBOL	TYPE	A	VALUE	/REF	(#DEF)	
LCD0	MEM	L	0.40	/#	28	, 228
LCD1	MEM	L	0.41	/#	29	, 229
LCD10	MEM	L	0.4A	/#	40	, 245
LCD11	MEM	L	0.4B	/#	41	, 247
LCD12	MEM	L	0.4C	/#	43	, 252
LCD13	MEM	L	0.4D	/#	44	, 253
LCD14	MEM	L	0.4E	/#	45	, 254
LCD15	MEM	L	0.4F	/#	46	, 255
LCD16	MEM	L	0.50	/#	48	, 260
LCD17	MEM	L	0.51	/#	49	, 261
LCD18	MEM	L	0.52	/#	50	, 262
LCD19	MEM	L	0.53	/#	51	, 263
LCD2	MEM	L	0.42	/#	30	, 230
LCD20	MEM	L	0.54	/#	53	, 268
LCD21	MEM	L	0.55	/#	54	, 269
LCD22	MEM	L	0.56	/#	55	, 270
LCD23	MEM	L	0.57	/#	56	, 271
LCD24	MEM	L	0.58	/#	58	, 275
LCD25	MEM	L	0.59	/#	59	, 277
LCD26	MEM	L	0.5A	/#	60	, 278
LCD27	MEM	L	0.5B	/#	61	, 279
LCD28	MEM	L	0.5C	/#	63	, 284
LCD29	MEM	L	0.5D	/#	64	, 285
LCD3	MEM	L	0.43	/#	31	, 231
LCD30	MEM	L	0.5E	/#	65	, 286
LCD31	MEM	L	0.5F	/#	66	, 287
LCD4	MEM	L	0.44	/#	33	, 236
LCD5	MEM	L	0.45	/#	34	, 237
LCD6	MEM	L	0.46	/#	35	, 238
LCD7	MEM	L	0.47	/#	36	, 239
LCD8	MEM	L	0.48	/#	38	, 244
LCD9	MEM	L	0.49	/#	39	, 245
LCDDAT1	MEM	P	0.10	/	1-12, #	68 , 197
LCDDAT2	MEM	P	0.11	/	1-11, #	69 , 198
LCDDAT3	MEM	P	0.12	/	1-10, #	70 , 199
LCDDAT4	MEM	P	0.13	/	1-9, #	71 , 200
LCDDAT5	MEM	P	0.14	/	1-8, #	72 , 201
LCDDAT6	MEM	P	0.15	/	1-7, #	73 , 202
LCDDAT7	MEM	P	0.16	/	1-6, #	74 , 203
LCDDAT8	MEM	P	0.17	/	1-5, #	75 , 204
LCD表示	LAD	P	0.20	/	1-4, #	184
WORK1	MEM	P	0.00	/	1-13, #	15 , 197 , 225
WORK2	MEM	L	0.01	/#	16	, 198 , 233
WORK3	MEM	L	0.02	/#	17	, 199 , 241
WORK4	MEM	L	0.03	/#	18	, 200 , 249
WORK5	MEM	L	0.04	/#	19	, 201 , 257
WORK6	MEM	L	0.05	/#	20	, 202 , 265
WORK7	MEM	L	0.06	/#	21	, 203 , 273
WORK8	MEM	L	0.07	/#	22	, 204 , 281

TOTAL SYMBOLS - 49

END OF XREF LIST



第4章 タイマ

μPD17201Aには、以下のような2系統のタイマがあります。

① 8ビット・タイマ/カウンタ

リモコン信号のリーダ・パルスの作成や、コード出力などに使用。

② 時計用タイマ

時計用の割り込み信号発生と、ウォッチドッグ・タイマのリセット信号発生などに使用。

ここでは、特に時計用タイマを利用したプログラムを紹介します。8ビット・タイマについては、第5章リモコン応用で紹介しています。

4.1 プログラム説明

4.1.1 時計カウント処理

時計用タイマを使った時計カウントのプログラムを紹介します。

このプログラムは、時計用タイマのソース・クロックを4 Hzに設定し、IRQWTMがセットされるごとにカウントし、4回カウントすると1秒とします。

4.1.2 オン/オフ・タイマ処理

ここでは、現在の時間がオン/オフ・タイマに設定した時間と一致した場合、ポート出力を行います。

- ・オン・タイマ設定時間になった場合 P0D₀をハイ・レベル出力します。
- ・オフ・タイマ設定時間になった場合 P0D₀をロウ・レベル出力します。

ただし、オン/オフ・タイマは、スタンバイ状態の場合のみ作動します。

また、オン・タイマの設定時間とオフ・タイマの設定時間が同一の場合は、オン/オフ・タイマは作動しません (P0D₀はロウ・レベル)。

スタンバイ状態 (STB_F=1) ……………オン/オフ・タイマ有効

スタンバイ状態解除 (STB_F=0) ……オン/オフ・タイマ無効

〈使用するハードウェア〉

- ・タイマ：時計用タイマ (ソース・クロック：4 Hz)
- ・ポート：P0D₀ (オン/オフ・タイマのポート出力)

〈初期設定〉

時間データ：時間データ (オン/オフ・タイマ時刻も含む) はすべてAM12:00に設定します。

〈起動方法〉

時計カウント処理で1分カウントごとにオン/オフ・タイマ処理をコールし、オン・タイマ設定時間およびオフ・タイマ設定時間と現在の時間を比較します。

ドキュメント



AS17K Z1.10 D1 << E17201A DOCUMENT >> 21:39:22 02/11/91

TABLE OF CONTENTS

	PAGE
第 4 章 タイマ	1
< プログラム説明 >	1
<< RAM の使用 >>	1
<<< 時計 / タイマ >>>	3
<<<< オフ / タイマ >>>>	3
<<<<< 初期設定 >>>>>	4

第 4 章 タイマ

PUBLIC (MEM) : COUNTER_H, COUNTER_L, HOUR_H, HOUR_L, OFFHR_H, OFFHR_L, ON_HR_H, ON_HR_L, SEC_H, SEC_L, T_1SEC_CNT
 PUBLIC (FLG) : AMPM_F, CLOCK, CLOCK_SET_F, DFF_AM_F, ON_AM_F, ON_OFF_SET_F, STB_F
 PUBLIC (LAB) : EI_RET11, TIMER_CNT
 ADDR RANGE : 00CSH - 011CH

<プログラム説明>

このプログラムは、時計用タイマを用いた12時間制時計およびオン/オフ・タイマ設定を行います。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

<RAM使用説明>

```

SEC_L ..... 0. 2 2 H ..... 12時間制時計の時間データ(秒の一の位)用のエリアです。
SEC_H ..... 0. 2 8 H ..... 12時間制時計の時間データ(秒の十の位)用のエリアです。
MIN_L ..... 0. 2 4 H ..... 12時間制時計の時間データ(分の一の位)用のエリアです。
MIN_H ..... 0. 2 5 H ..... 12時間制時計の時間データ(分の十の位)用のエリアです。
HOUR_L ..... 0. 2 6 H ..... 12時間制時計の時間データ(時の一の位)用のエリアです。
HOUR_H ..... 0. 2 7 H ..... 12時間制時計の時間データ(時の十の位)用のエリアです。
AMPM ..... 0. 2 8 H ..... 12時間制時計のAM/PM切り替えのフラグ用のエリアです。
    AMPM_F ..... # 3 ..... AMのときにセットしています。
    
```


ON_MIN_L 0. 3 4 H オン・タイマ時間データ（分の一の位）用のエリアです。
ON_MIN_H 0. 3 5 H オン・タイマ時間データ（分の十の位）用のエリアです。
ON_HR_L 0. 3 8 H オン・タイマ時間データ（時の一の位）用のエリアです。
ON_HR_H 0. 3 7 H オン・タイマ時間データ（時の十の位）用のエリアです。
ON_AM_P # 3 AMのときにセットしています。
OFFMIN_L 0. 6 4 H オフ・タイマ時間データ（分の一の位）用のエリアです。
OFFMIN_H 0. 6 5 H オフ・タイマ時間データ（分の十の位）用のエリアです。
OFFHR_L 0. 6 6 H オフ・タイマ時間データ（時の一の位）用のエリアです。
OFFHR_H 0. 6 7 H オフ・タイマ時間データ（時の十の位）用のエリアです。
OFFAMP 0. 8 8 H オン・タイマ時間のAM/P切り替えのフラグ用のエリアです。
OFF_AM_P # 3 AMのときにセットしています。
COUNTER_H 0. 2 9 H 持ち時間設定用のカウンタです。
(0. 2 A H)
T_1SEC_CNT 0. 2 B H 時計カウント（1秒）用のカウンタです。
FLAG 0. 3 1 H オン/オフ・タイマ設定中のフラグ用のエリアです。
ON_OFF_SET_P # 3 オン/オフ・タイマ設定中の場合セットします。
C_S_TM 0. 3 0 H 時計調整およびスタンバイ状態のフラグ用のエリアです。
CLOCK_SET_P # 3 時計調整中の場合セットします。
STB_F # 2 スタンバイ状態の場合セットします。
TIMER_F # 1 現在時刻とオン・タイマ時間が一致したときセットします。
CLOCK # 0 時計調整キーが押された時セットします。
WR_1 0. 6 9 H スタックの退避用のエリアです。
PSW_1 0. 6 A H スタックの退避用のエリアです。

ENTRANCES :-
MEMORIES CHANGED :-
MEMORIES REFERRED :-
MEMORIES MANIPULATED :-
FLAGS CHANGED :-
FLAGS REFERRED :-
DATA REFERRED :-
BRANCH TO :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL :-

<時計カウント>

ここでは、時計用タイマのソース・クロックを4Hzに設定し、IRQTMがセットされる(割り込み発生する)たびにカウントし、4回のカウントで1秒とします。

```

ENTRANCES          :BI_RETII      TIMER_CNT
MEMORIES CHANGED   :COUNTER_H     COUNTER_L
                   :  HOUR_H       HOUR_L
                   :  MIN_H        MIN_L
                   :  PSW_I        SEC_H
                   :  SEC_L        T_1SEC_CNT
                   :  WR_I
MEMORIES REFERRED   :PSW_I         WR_I
MEMORIES MANIPULATED : -
FLAGS CHANGED      :AMPW_F
FLAGS REFERRED     :CLOCK_SET_F   ON_OFF_SET_F
                   :STB_F
DATA REFERRED      : -
BRANCH TO          : -
SUBROUTINES CALLED :ON_OFF_TIMER
LABELS MANIPULATED : -
SYSTEM CALL        : -
    
```

<オン/オフ・タイマ>

ここでは、オンまたはオフ・タイマに設定した時間が現在の時間と一致するかどうか調べます。スタンバイ状態のときのみ、このサブ・ルーチンが呼び出されます。また、オン・タイマとオフ・タイマで設定した時間が同じ場合、オン/オフ・タイマは作動しません。

現在の時刻が、オン・タイマ時間と一致した場合、PODの#0でハイ・レベル出力します。
現在の時刻が、オフ・タイマ時間と一致した場合、PODの#0でロウ・レベル出力します。

```

ENTRANCES          :ON_OFF_TIMER
MEMORIES CHANGED   :HOUR_H         HOUR_L
                   :MIN_H          MIN_L
MEMORIES REFERRED   :OFFHR_H       OFFHR_L
                   :OFFMIN_H      OFFMIN_L
                   :ON_HR_H        ON_HR_L
                   :ON_MIN_H       ON_MIN_L
MEMORIES MANIPULATED : -
FLAGS CHANGED      :TIMER_F
FLAGS REFERRED     :AMPW_F         OFF_AM_F
                   :ON_AM_F       TIMER_F
DATA REFERRED      : -
BRANCH TO          : -
SUBROUTINES CALLED : -
LABELS MANIPULATED : -
SYSTEM CALL        : -
    
```

<初期設定>

時計およびオン/オフ・タイマの初期設定は、すべてAM 12:00です

```
PUBLIC (LAB) : MAIN_INIT
EXTRN (MEM) : HOUR_H, HOUR_L, KEYDATA_L, OFFHR_H, OFFHR_L, ON_HR_H, ON_HR_L
EXTRN (PLG) : AMPM_F, K_CHANGE, K_ERROR, K_EXIST, OFF_AM_F, ON_AM_F, STB_F
EXTRN (LAB) : DISPLAY, KEYCHECK, RAMCLEAR, TIME_SET
ADDR RANGE : 0262H - 02ACH
```



RAMマップ

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 21:38:40 02/11/91 PAGE 03-001

PROG =

SOURCE = TIMER.ASM

(PUBLIC/LOCAL MAP)

BANK = 0

ROW

	7	6	5	4	3	2	1	0	
1		1LCDD32	1LCDD16	1LCDD0					0
1					1C_S_TM				
1		1LCDD33	1LCDD17	1LCDD1					1
1					1FLAG				
1		1LCDD34	1LCDD18	1LCDD2		1SEC_L			2
1									
1		1LCDD35	1LCDD19	1LCDD3		1SEC_H			3
1									
1	1ARB		1LCDD20	1LCDD4					4
1		1OFFMIN_L			1ON_MIN_L	1MIN_L			
1									
1	1AR2		1LCDD21	1LCDD5					5
1		1OFFMIN_H			1ON_MIN_H	1MIN_H			
1									
1	1AR1	1OFFHR_L	1LCDD22	1LCDD6	1ON_HR_L	1HOUR_L			6
1									
1	1AR0	1OFFHR_H	1LCDD23	1LCDD7	1ON_HR_H	1HOUR_H			7
1									
1	1WR		1LCDD24	1LCDD8					8
1									
1	1BANK		1LCDD25	1LCDD9		1COUNTER_H			9
1		1WR_L							
1									
1	1IXH		1LCDD26	1LCDD10		1COUNTER_L			A
1	1MPH								
1		1PSW_L							
1									
1	1IXM		1LCDD27	1LCDD11		1T_1SEC_CNT			B
1	1MPL								
1									
1	1IXL		1LCDD28	1LCDD12			1DBF3		C
1									
1	1RPH		1LCDD29	1LCDD13			1DBF2		D
1									

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 21:38:40 02/11/91 PAGE 03-002

PROG =

SOURCE = TIMER.ASM

(PUBLIC/LOCAL MAP)

BANK = 0

ROW								
7	6	5	4	3	2	1	0	
IRPL		LCDD30	LCDD14				DBF1	R
IPSW		LCDD31	LCDD15				DBF0	F

AS17K Z1.10 D1 << R17201A FLG MAP >> 21:38:40 02/11/91 PAGR 03-003

PROG -

SOURCE - TIMBR.ASM

(FLAG MAP) BANK - 0

MSB	3	2	1	0	LSB
27	IANPM_F				
30	ICLOCK_SET_F	ISTB_F		ICLOCK	
			TIMER_F		
31	ION_OFF_SET_F				
37	ION_AM_F				
67	IOFF_AM_F				
70	IPOA3	IPOA2	IPOA1	IPOA0	
71	IPOB3	IPOB2	IPOB1	IPOB0	
72	IPOC3	IPOC2	IPOC1	IPOC0	
73	IPOD3	IPOD2	IPOD1	IPOD0	
7A	IMFE				
7E				IBCD	
7F	ICMP	ICY	IZ	IXE	
82			ISYCK	IXEN	
83	IWTRES	IWTWMD	IWTMRES		
87	IVDDDET1	IVDDDET0			
8F				INT	
91				INRZBP	

AS17K Z1.10 D1 << B17201A FLG MAP >> 21:38:40 02/11/91 PAGE 03-004

PROC -

SOURCE = TIWER.ASM

(FLAG MAP) BANK = 0

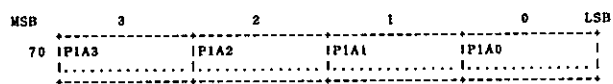
MSB	3	2	1	0	LSB
B2				INRZ	
A0				ADCCMP	
A1	VREPEN	ADCRN	ADCCH1	ADCCH0	
A2	SIOTS	SIOHIZ	SIOCK1	SIOCK0	
A3		INRZEN	TMOE	SIOEN	
A7	P0DBIO3	P0DBIO2	P0DBIO1	P0DBIO0	
AP	IPSIO	IPWTM	IP	IPTM	
B1	LCDEN	LCDCK2	LCDCK1	LCDCK0	
B2	LCDMD3	LCDMD2	LCDMD1	LCDMD0	
B3	TMEN	TMRSS	TMCK1	TMCK0	
B7	PIAGIO	P0CGIO	P0BGIO	P0AGIO	
BB	IRQSIO				
BC		IRQTM			
BD			IRQ		
BE				IRQTM	

AS17K Z1.10 D1 << E17201A FLG MAP >> 21:38:40 02/11/91 PAGE 03-005

PROG -

SOURCE - TIMER.ASM

(FLAG MAP) BANK - 1



RAM使用説明

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 21:38:40 02/11/91 PAGE 03-005

PROG -

SOURCE - TIMER.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
ADCC0	FLG	0	A1	...1	
ADCC1	FLG	0	A1	..1.	
ADCCMP	FLG	0	A0	...1	
ADCN	FLG	0	A1	...1	
AMPM_F	FLG	0	27	1...	12時間制時計のAM/PM切り替えのフラグ用のエリアです。
AR0	MEM	0	77		
AR1	MEM	0	76		
AR2	MEM	0	75		
AR3	MEM	0	74		
BANK	MEM	0	79		
BCD	FLG	0	7B	...1	
C_S_TM	MEM	0	30		時計調整およびスタンバイ状態のフラグ用のエリアです。
CLOCK	FLG	0	30	...1	時計調整キーが押された時セットします。
CLOCK_SET_F	FLG	0	30	1...	時計調整中の場合セットします。
CMP	FLG	0	7F	1...	
COUNTER_H	MEM	0	29		待ち時間設定用のカウンタです。
COUNTER_L	MEM	0	2A		待ち時間設定用のカウンタです。
CF	FLG	0	7F	...1	
DBF0	MEM	0	F		
DBF1	MEM	0	E		
DBF2	MEM	0	D		
DBF3	MEM	0	C		
FLAG	MEM	0	31		オン/オフ・タイマ設定中のフラグ用のエリアです。
HOUR_H	MEM	0	27		12時間制時計の時間データ(時の十の位)用のエリアです。
HOUR_L	MEM	0	26		12時間制時計の時間データ(時の一の位)用のエリアです。
INT	FLG	0	8F	...1	
IP	FLG	0	AF	...1	
IPSIO	FLG	0	AF	1...	
IPTM	FLG	0	AF	...1	
IPWTM	FLG	0	AF	...1	
IRQ	FLG	0	DD	...1	
IRQSIO	FLG	0	DB	1...	
IRQTM	FLG	0	BE	...1	
IRQWTM	FLG	0	BC	...1	
IXE	FLG	0	7F	...1	
IXH	MEM	0	7A		
IXL	MEM	0	7C		
IXM	MEM	0	7B		
LCDC0	FLG	0	B1	...1	
LCDC1	FLG	0	B1	..1.	
LCDC2	FLG	0	B1	...1	
LCDD0	MEM	0	40		
LCDD1	MEM	0	41		
LCDD10	MEM	0	4A		
LCDD11	MEM	0	4B		
LCDD12	MEM	0	4C		
LCDD13	MEM	0	4D		
LCDD14	MEM	0	4E		
LCDD15	MEM	0	4F		
LCDD16	MEM	0	50		
LCDD17	MEM	0	51		
LCDD18	MEM	0	52		

AS17K Z1.10 D1 << B17201A DATA MEMORY MAP >> 21:38:40 02/11/91 PAGE 03-007

PROG -

SOURCE - TIMER.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCDD19	MEM	0	53		
LCDD2	MEM	0	42		
LCDD20	MEM	0	54		
LCDD21	MEM	0	55		
LCDD22	MEM	0	56		
LCDD23	MEM	0	57		
LCDD24	MEM	0	58		
LCDD25	MEM	0	59		
LCDD26	MEM	0	5A		
LCDD27	MEM	0	5B		
LCDD28	MEM	0	5C		
LCDD29	MEM	0	5D		
LCDD3	MEM	0	43		
LCDD30	MEM	0	5E		
LCDD31	MEM	0	5F		
LCDD32	MEM	0	60		
LCDD33	MEM	0	61		
LCDD34	MEM	0	62		
LCDD35	MEM	0	63		
LCDD4	MEM	0	44		
LCDD5	MEM	0	45		
LCDD6	MEM	0	46		
LCDD7	MEM	0	47		
LCDD8	MEM	0	48		
LCDD9	MEM	0	49		
LCDEN	FLG	0	B1	1...	
LCDM0	FLG	0	B2	...1	
LCDM1	FLG	0	B2	...1	
LCDM2	FLG	0	B2	...1	
LCDM3	FLG	0	B2	...1	
MIN_H	MEM	0	25		1 2時間制時計の時間データ(分の十の位)用のエリアです。
MIN_L	MEM	0	24		1 2時間制時計の時間データ(分の一の位)用のエリアです。
NPE	FLG	0	7A	1...	
NPH	MEM	0	7A		
NPL	MEM	0	7B		
NRZ	FLG	0	92	...1	
NRZBF	FLG	0	91	...1	
NRZEN	FLG	0	A3	1...	
OFF_AM_P	FLG	0	67	1...	オフ
OFFHR_H	MEM	0	67		タイマ時間データのデ
OFFHR_L	MEM	0	66		タイマ時間データのデ
OFFMIN_H	MEM	0	66		タイマ時間データのデ
OFFMIN_L	MEM	0	64		タイマ時間データのデ
ON_AM_P	FLG	0	37	1...	オン
ONHR_H	MEM	0	37		タイマ時間データのデ
ONHR_L	MEM	0	36		タイマ時間データのデ
ONMIN_H	MEM	0	35		タイマ時間データのデ
ONMIN_L	MEM	0	34		タイマ時間データのデ
ON_OFF_SET_F	FLG	0	31	1...	オン
POA0	FLG	0	70	...1	オフ
POA1	FLG	0	70	...1	タイマ時間データのデ
POA2	FLG	0	70	...1	タイマ時間データのデ

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 21:38:40 02/11/01 PAGE 03-008

PROG -

SOURCE = TIMER.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
POA3	FLG	0	70	1...	
POAGIO	FLG	0	B7	...1	
POB0	FLG	0	71	...1	
POB1	FLG	0	71	...1	
POB2	FLG	0	71	...1	
POB3	FLG	0	71	...1	
POBGIO	FLG	0	B7	...1	
POC0	FLG	0	72	...1	
POC1	FLG	0	72	...1	
POC2	FLG	0	72	...1	
POC3	FLG	0	72	...1	
POCGIO	FLG	0	B7	...1	
POD0	FLG	0	73	...1	
POD1	FLG	0	73	...1	
POD2	FLG	0	73	...1	
POD3	FLG	0	73	...1	
PODBIO0	FLG	0	A7	...1	
PODBIO1	FLG	0	A7	...1	
PODBIO2	FLG	0	A7	...1	
PODBIO3	FLG	0	A7	...1	
PIA0	FLG	1	70	...1	
PIA1	FLG	1	70	...1	
PIA2	FLG	1	70	...1	
PIA3	FLG	1	70	...1	
PIAGIO	FLG	0	B7	...1	
PSW	MEM	0	7F		
PSW_I	MEM	0	6A		スタックの退避用のエリアです。
RPH	MEM	0	7D		
RPL	MEM	0	7E		
SEC_H	MEM	0	23		1 2 時間制時計の時間データ (秒の十の位) 用のエリアです。
SEC_L	MEM	0	22		1 2 時間制時計の時間データ (秒の一の位) 用のエリアです。
SIOCK0	FLG	0	A2	...1	
SIOCK1	FLG	0	A2	...1	
SIOEN	FLG	0	A3	...1	
SIOHIZ	FLG	0	A2	...1	
SIOIS	FLG	0	A2	...1	
STB_F	FLG	0	30	...1	スタンバイ状態の場合セットします。
SYSC	FLG	0	82	...1	
T_1SEC_CNT	MEM	0	2B		時計カウント (1秒) 用のカウンタです。
TIMER_F	FLG	0	30	...1	現在時刻とオン・タイマ時間が一致したときセットします。
TMCK0	FLG	0	B3	...1	
TMCK1	FLG	0	B3	...1	
TMBN	FLG	0	B3	...1	
TMOE	FLG	0	A3	...1	
WTRES	FLG	0	B3	...1	
VDDDET0	FLG	0	87	...1	
VDDDET1	FLG	0	87	...1	
VRRFEN	FLG	0	A1	...1	
WDRES	FLG	0	83	...1	
WR	MEM	0	78		
WR_I	MEM	0	69		スタックの退避用のエリアです。
WTMND	FLG	0	83	...1	

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 21:38:40 02/11/91 PAGE 03-009

PROG -

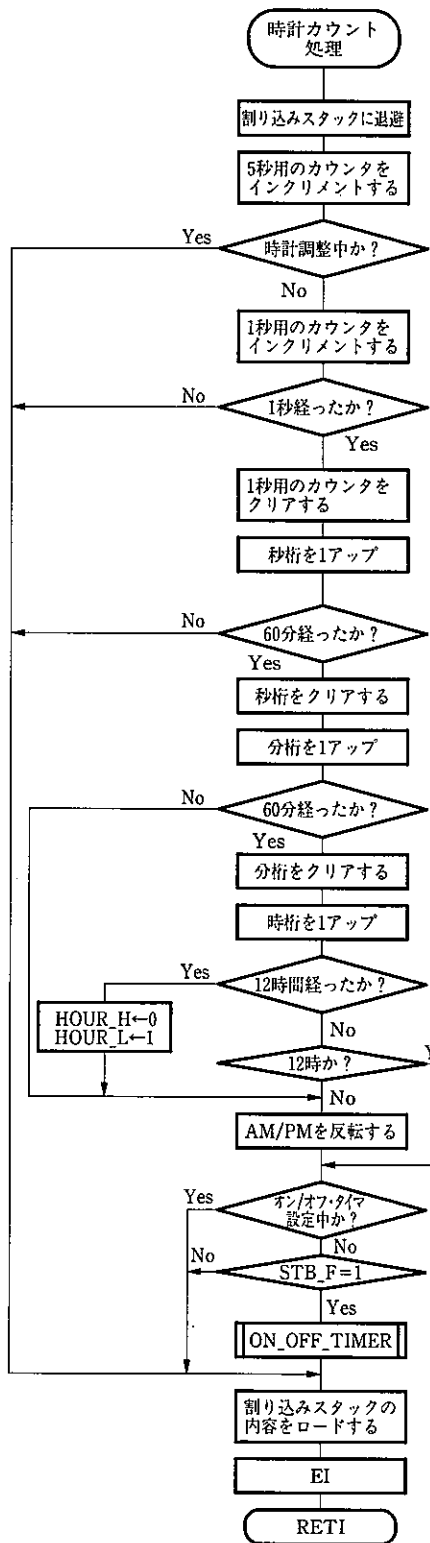
SOURCE - TIMER.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	DIT	INFORMATION
WTMRES	FLG	0	83	..1.	
XEN	FLG	0	82	...1	
Z	FLG	0	7F	..1.	

フロー・チャート

フロー・チャート



他の処理の動作用タイマ・カウンタ

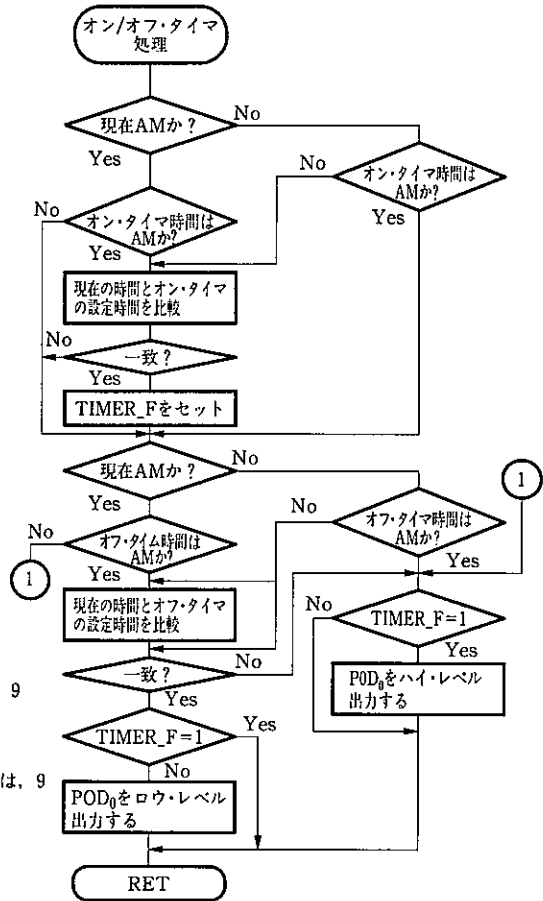
時計調整中は時計カウントしない

HOUR_H=1または、9
HOUR_L=3

HOUR_H=1または、9
HOUR_L=2

スタンバイ状態か?

オン/オフ・タイマ処理ルーチン



リスト


```
INCLUDE 'TIMER.PUB'
SUMMARY '%%', '第4章 タイマ'
%%
SUMMARY '%%', '<プログラム説明>'
```

このプログラムは、時計用タイマを用いた12時間制時計およびオン/オフ・タイマ設定を行います。

```
%%
:.....
:*** 使用メモリ、レジスタ、フラグ ***
:.....
```

SEC_L	MEM	0.22H	; 12時間制時計の時間データ(秒の一の位)用のエリアです。
SEC_H	MEM	0.23H	; 12時間制時計の時間データ(秒の十の位)用のエリアです。
MIN_L	MEM	0.24H	; 12時間制時計の時間データ(分の一の位)用のエリアです。
MIN_H	MEM	0.25H	; 12時間制時計の時間データ(分の十の位)用のエリアです。
HOUR_L	MEM	0.26H	; 12時間制時計の時間データ(時の一の位)用のエリアです。
HOUR_H	MEM	0.27H	; 12時間制時計の時間データ(時の十の位)用のエリアです。
AMPM_F	FLG	HOUR_H.3	; 12時間制時計のAM/PM切り替えのフラグ用のエリアです。 ; AMのときにセットしています。
COUNTER_H	MEM	0.29H	; 待ち時間設定用のカウンタです。
COUNTER_L	MEM	0.2AH	; 待ち時間設定用のカウンタです。
T_1SEC_CNT	MEM	0.2BH	; 時計カウント(1秒)用のカウンタです。
ON_MIN_L	MEM	0.34H	; オン・タイマ時間データ(分の一の位)用のエリアです。
ON_MIN_H	MEM	0.35H	; オン・タイマ時間データ(分の十の位)用のエリアです。
ON_HR_L	MEM	0.36H	; オン・タイマ時間データ(時の一の位)用のエリアです。
ON_HR_H	MEM	0.37H	; オン・タイマ時間データ(時の十の位)用のエリアです。
ON_AM_F	FLG	ON_HR_H.3	; オン・タイマ時間のAM/PM切り替えのフラグ用のエリアです。 ; AMのときにセットしています。
OFFMIN_L	MEM	0.64H	; オフ・タイマ時間データ(分の一の位)用のエリアです。
OFFMIN_H	MEM	0.65H	; オフ・タイマ時間データ(分の十の位)用のエリアです。
OFFHR_L	MEM	0.66H	; オフ・タイマ時間データ(時の一の位)用のエリアです。
OFFHR_H	MEM	0.67H	; オフ・タイマ時間データ(時の十の位)用のエリアです。
OFF_AM_F	FLG	OFFHR_H.3	; オフ・タイマ時間のAM/PM切り替えのフラグ用のエリアです。 ; AMのときにセットしています。
FLAG	MEM	0.31H	; オン/オフ・タイマ設定中のフラグ用のエリアです。
ON_OFF_SET_F	FLG	FLAG.3	; オン/オフ・タイマ設定中の場合セットします。
C_S_TM	MEM	0.30H	; 時計調整およびスタンバイ状態のフラグ用のエリアです。
CLOCK_SET_F	FLG	C_S_TM.3	; 時計調整中の場合セットします。
STB_F	FLG	C_S_TM.2	; スタンバイ状態の場合セットします。
TIMER_F	FLG	C_S_TM.1	; 現在時刻とオン・タイマ時間が一致したときセットします。
CLOCK	FLG	C_S_TM.0	; 時計調整キーが押された時セットします。
WR_1	MEM	0.69H	; スタックの退避用のエリアです。
PSW_1	MEM	0.6AH	; スタックの退避用のエリアです。

SUMMARY '%', '<RAM使用説明>'

SEC_L	0. 2 2 H	1 2 時間制時計の時間データ (秒の一の位) 用のエリアです。
SEC_H	0. 2 3 H	1 2 時間制時計の時間データ (秒の十の位) 用のエリアです。
MIN_L	0. 2 4 H	1 2 時間制時計の時間データ (分の一の位) 用のエリアです。
MIN_H	0. 2 5 H	1 2 時間制時計の時間データ (分の十の位) 用のエリアです。
HOURL	0. 2 6 H	1 2 時間制時計の時間データ (時の一の位) 用のエリアです。
HOURL	0. 2 7 H	1 2 時間制時計の時間データ (時の十の位) 用のエリアです。
AMPM	0. 2 8 H	1 2 時間制時計の AM / PM 切り替えのフラグ用のエリアです。
AMPM_P	# 3	AM のときにセットしています。
EJECT		
ON_MIN_L	0. 3 4 H	オン・タイマ時間データ (分の一の位) 用のエリアです。
ON_MIN_H	0. 3 5 H	オン・タイマ時間データ (分の十の位) 用のエリアです。
ON_HR_L	0. 3 6 H	オン・タイマ時間データ (時の一の位) 用のエリアです。
ON_HR_H	0. 3 7 H	オン・タイマ時間データ (時の十の位) 用のエリアです。
ON_AM_P	# 3	AM のときにセットしています。
OFFMIN_L	0. 6 4 H	オフ・タイマ時間データ (分の一の位) 用のエリアです。
OFFMIN_H	0. 6 5 H	オフ・タイマ時間データ (分の十の位) 用のエリアです。
OFFHR_L	0. 6 6 H	オフ・タイマ時間データ (時の一の位) 用のエリアです。
OFFHR_H	0. 6 7 H	オフ・タイマ時間データ (時の十の位) 用のエリアです。
OFFAMPM	0. 6 8 H	オン・タイマ時間の AM / PM 切り替えのフラグ用のエリアです。
OFF_AM_P	# 3	AM のときにセットしています。
COUNTER_H	0. 2 9 H (0. 2 A H)	待ち時間設定用のカウンタです。
T_1SEC_CNT	0. 2 B H	時計カウント (1 秒) 用のカウンタです。
FLAG	0. 3 1 H	オン / オフ・タイマ設定中のフラグ用のエリアです。
ON_OFF_SET_F	# 3	オン / オフ・タイマ設定中の場合セットします。
C_S_TM	0. 3 0 H	時計調整およびスタンバイ状態のフラグ用のエリアです。
CLOCK_SET_F	# 3	時計調整中の場合セットします。
STB_F	# 2	スタンバイ状態の場合セットします。
TIMER_F	# 1	現在時刻とオン・タイマ時間が一致したときセットします。
CLOCK	# 0	時計調整キーが押された時セットします。
WR_I	0. 6 9 H	スタックの送還用のエリアです。
PSW_I	0. 6 A H	スタックの送還用のエリアです。

EJ-1

%%

SUMMARY '%', '<時計カウント>'

ここでは、時計用タイマのソース・クロックを 4 H π に設定し、IRQNTM がセットされる (割り込み発生する) たびにカウントし、4 回のカウントで 1 秒とします。

```

%%
TIMER_CNT:
;.....
;*** SAVE SYSTEM REGISTOR ***
;.....
    POKE    WR_1,WR          ;割り込みスタックに送達
    PEEK    WR,PSW
    POKE    PSW_1,WR
    MOV     PSW,#0
;.....
;*** 12時間制時計 ***
;.....
    ADD     COUNTER_L,#1     ;5秒カウント用のカウンタをインクリメントする
    ADDC    COUNTER_H,#0     ;(オン/オフ・タイマ時間設定後、5秒で通常の時計表示に戻すため。)
    SKPF1   CLOCK_SET_F     ;時計調整中の場合時計カウントしない
    BR      EI_RET1
    ADD     T_ISEC_CNT,#1    ;時計用タイマの割り込み(250ms毎)が入ったら1カウント・アップ
    SKE     T_ISEC_CNT,#4    ;1秒経ったか?
    BR      EI_RET1
    MOV     T_ISEC_CNT,#0    ;カウンタをクリア
;.....
;*** SEC COUNT ***
;.....
    SETI    BCD              ;以下、10進数で計算する
    ADD     SEC_L,#1         ;秒桁カウント
    ADDC    SEC_H,#0
    SKE     SEC_H,#6        ;60秒経ったか?
    BR      TIMER2
    MOV     SEC_L,#0        ;秒桁クリア
    MOV     SEC_H,#0
;.....
;*** MIN COUNT ***
;.....
    ADD     MIN_L,#1        ;分桁カウント
    ADDC    MIN_H,#0
    SKE     MIN_H,#6       ;60分経ったか?
    BR      TIMER1
    MOV     MIN_L,#0       ;分桁クリア
    MOV     MIN_H,#0
;.....
;*** HOUR COUNT ***
;.....
    ADD     HOUR_L,#1       ;時桁カウント
    ADDC    HOUR_H,#0
    SKE     HOUR_H,#1      ;12時間経ったか?(PMの場合)
    SKNE    HOUR_H,#9      ;(AMの場合)
    SKE     HOUR_L,#3
    BR      TIMER0
    MOV     HOUR_L,#1
    AND     HOUR_H,#1000B   ;AMの場合、最上位ビットが1になっているため下位3ビットのみクリアする
    BR      TIMER1
;
TIMER0:
    SKE     HOUR_H,#1      ;12時になったか?(PMの場合)
    SKNE    HOUR_H,#9     ;(AMの場合)
    SKE     HOUR_L,#2
    BR      TIMER1
NOT1     AMPM_P           ;AM→PM OR PM→AM
TIMER1:
    SKPF1   ON_OFF_SET_F   ;オン/オフ・タイマ設定中か?
    BR      TIMER2
    SKPF1   STB_F          ;オン/オフ・タイマ設定中の場合、オン/オフ・タイマ処理は行いません。
    CALL    ON_OFF_TIMER   ;スタンバイ状態か?
    CALL    ON_OFF_TIMER   ;オン/オフ・タイマ時間と現在時刻と比較する
TIMER2:
    CLR1    BCD            ;16進数計算に戻す

```

```

:*****
:*** LOAD SYSTEM REGISTER ***
:*****
EI_RET1:
        PEEK    WR, PSW_1      ;割り込みスタックの内容をロードする
        POKE    PSW, WR
        PEEK    WR, WR_1
EI_RET11:
        EI
        RETI
:*****
:*** オン/オフ・タイマと現在時刻との比較 ***
:*****
SUMMARY    '%%. ' <オン/オフ・タイマ>'

ここでは、オンまたはオフ・タイマに設定した時間が現在の時間と一致するかどうか調べます。
スタンバイ状態のときのみ、このサブ・ルーチンが呼びべれます。
また、オン・タイマとオフ・タイマで設定した時間が同じ場合、オン/オフ・タイマは作動しません。

現在の時刻が、オン・タイマ時間と一致した場合、POD の #0 でハイ・レベル出力します。
現在の時刻が、オフ・タイマ時間と一致した場合、POD の #0 でロウ・レベル出力します。
%%

ON_OFF_TIMER:
        MOV     RPL, #4
        CLR1    TIMER_F
        SKT1    AMPM_F          ;現在 AM か？
        BR      ON_T0
        SKF1    ON_AM_F         ;オン・タイマの設定時間は AM か？
        BR      ON_T0
        BR      OFF_T

ON_T0:
        SKF1    ON_AM_F         ;オン・タイマの設定時間は AM か？
        BR      OFF_T

ON_T1:
        SET2    CMP, Z          ;現在の時間とオン・タイマの設定時間を比較
        SUB     MIN_L, ON_MIN_L
        SUB     MIN_H, ON_MIN_H
        SUB     HOUR_L, ON_HR_L
        SUB     HOUR_H, ON_HR_H
        SKF1    Z
        SET1    TIMER_F        ;一致
                                ;不一致
                                ;現在 AM か？

OFF_T:
        SKT1    AMPM_F
        BR      OFF_T0
        SKF1    OFF_AM_F       ;オフ・タイマの設定時間は AM か？
        BR      OFF_T1
        BR      RET_1

OFF_T0:
        SKF1    OFF_AM_F       ;オフ・タイマの設定時間は AM か？
        BR      RET_1

OFF_T1:
        SET2    CMP, Z          ;現在の時間とオフ・タイマの設定時間を比較
        SUB     MIN_L, OFFMIN_L
        SUB     MIN_H, OFFMIN_H
        SUB     HOUR_L, OFFHR_L
        SUB     HOUR_H, OFFHR_H
        SKT1    Z
        BR      RET_1          ;不一致
        BR      RET_2          ;一致

RET_1:
        SKT1    TIMER_F
        BR      RET_3          ;オン/オフ・タイマの両方の時間も現在の時刻と一致しない場合
        SET1    POD0
        BR      RET_3          ;現在時刻とオン・タイマ時間が一致した場合 POD の #0 よりハイ・レベル出力

```

```
RET_2:          SKT1   TIMER_F      ;オン・タイマ時間とオフ・タイマ時間が一致した場合
                CLR1   P0D0        ;現在時刻とオフ・タイマ時間が一致した場合 P0D の #0 よりロウ・レベル出力
RET_3:          MOV    RPL, #0
                RET
END
```


クロスレファレンス

第4章 タイマ

AS17K Z1.10 D1 << E17201A XREF LIST >>

21:38:39 02/11/91 PAGE 03-001

PROG -

SOURCE = TIMER.ASM

SYMBOL	TYPE	A	VALUE	/REP	(#DEF)													
AMPM_F	FLG	P	0.27.3	/	1-4	# 25												
					243-1	# 243-1												
C_S_TM	MEM	L	0.30	/#	59	# 60												
CLOCK	FLG	P	0.30.0	/	1-5	# 63												
CLOCK_SET_F	FLG	P	0.30.3	/	1-8	# 60												
COUNTER_H	MEM	P	0.29	/	1-6	# 28												
COUNTER_L	MEM	P	0.2A	/	1-7	# 30												
E1_RET1	LAB	L	F1	/	149	# 152												
E1_RET11	LAB	P	F4	/	1-8	# 206												
FLAG	MEM	L	0.31	/#	56	# 57												
HOUR_H	MEM	P	0.27	/	1-11	# 23												
					239	# 256												
HOUR_L	MEM	P	0.28	/	1-10	# 21												
MIN_H	MEM	L	0.25	/#	19	# 168												
MIN_L	MEM	L	0.24	/#	17	# 167												
OFF_AM_F	FLG	P	0.67.3	/	1-13	# 53												
OFF_T	LAB	L	106	/	230	# 233												
OFF_TO	LAB	L	10B	/	244	# 248												
OFF_T1	LAB	L	10D	/	246	# 251												
OFFHR_H	MEM	P	0.67	/	1-14	# 51												
OFFHR_L	MEM	P	0.66	/	1-16	# 49												
OFFMIN_H	MEM	L	0.65	/#	47	# 254												
OFFMIN_L	MEM	L	0.64	/#	45	# 253												
ON_AM_F	FLG	P	0.37.3	/	1-15	# 42												
ON_HR_H	MEM	P	0.37	/	1-12	# 40												
ON_HR_L	MEM	P	0.36	/	1-17	# 38												
ON_MIN_H	MEM	L	0.35	/#	36	# 237												
ON_MIN_L	MEM	L	0.34	/#	34	# 236												
ON_OFF_SET_F	FLG	P	0.31.3	/	1-18	# 57												
ON_OFF_TIMER	LAB	L	F6	/	196	# 223												
ON_TO	LAB	L	FD	/	227	# 231												
ON_T1	LAB	L	FF	/	229	# 234												
PSW_I	MEM	L	0.6A	/#	66	# 141												
RET_1	LAB	L	116	/	247	# 250												
RET_2	LAB	L	119	/	259	# 265												
RET_3	LAB	L	11B	/	262	# 264												
SEC_H	MEM	P	0.23	/	1-19	# 15												
SEC_L	MEM	P	0.22	/	1-20	# 13												
STD_F	FLG	P	0.39.2	/	1-21	# 61												
T_1SEC_CNT	MEM	P	0.2B	/	1-22	# 32												
TIMER0	LAB	L	E7	/	181	# 186												
TIMER1	LAB	L	EC	/	170	# 184												
TIMER2	LAB	L	FO	/	161	# 194												
TIMER_CNT	LAB	P	CS	/	1-23	# 135												
TIMER_F	FLG	L	0.30.1	/#	62	# 225												
					261-1	# 266												
WR_I	MEM	L	0.69	/#	65	# 139												

TOTAL SYMBOLS = 44

END OF XREF LIST

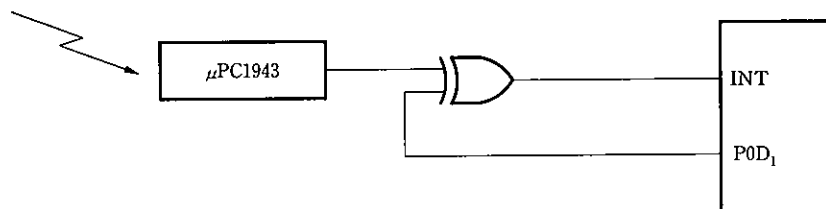
第5章 リモコン応用

5.1 受信

リモコン受信例として、リモート・コントロール送信用デバイス μ PD6121Gからの送信データを、リモート・コントロール用プリアンプ μ PC1943を通して、INT端子（41ピン）から入力します。

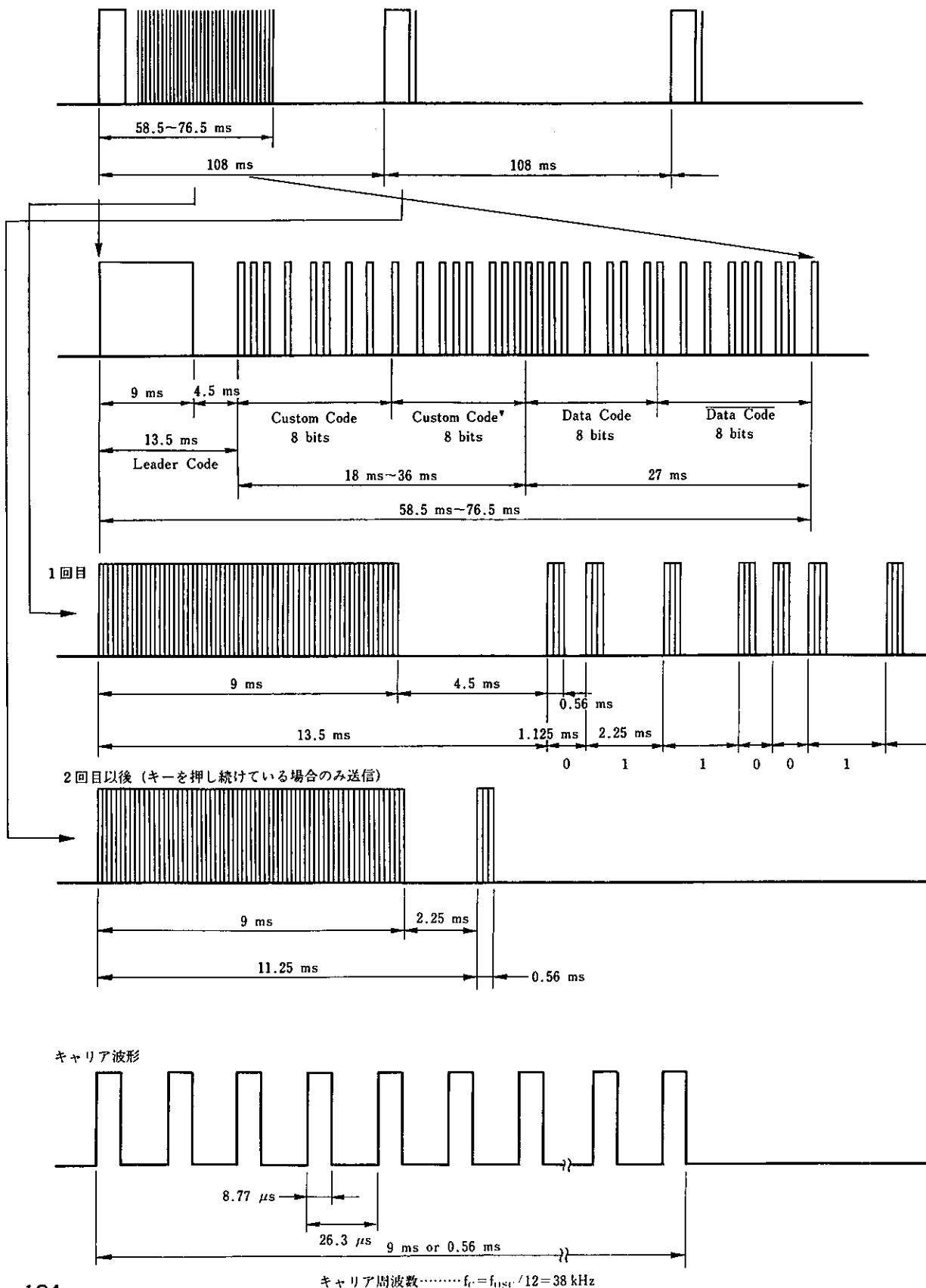
このとき、INT端子の入カレベルによるINT端子の割り込み動作を両エッジで可能にするために、割り込みエッジ選択用出力ポートとしてP0D₁端子を使用し、プリアンプとINT端子の間にEx-OR回路を設けます。このときの回路を図5-1に示します。

図5-1 リモコン受信回路例



μPD6121Gからの送信信号を図5-2に示します。

図5-2 リモコン送信用IC出力信号



5.1.1 プログラム説明

INT端子よりリモコン信号を入力し、カスタム・コードおよびデータ・コードを設定します。

〈使用するハードウェア〉

番号	端子名	使用説明																					
41	INT	リモコン送信用ICからの出力信号をP0D ₁ 端子のレベルとEx-ORしたレベルが入力されます。																					
55	P0D ₁	<p>INT端子に入力されるリモコン送信用ICからの出力信号を制御するための出力端子です。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>リモコンIC出力</th> <th>P0D₁</th> <th>INT入力</th> </tr> </thead> <tbody> <tr> <td>ロウ</td> <td>ロウ</td> <td>ロウ</td> </tr> <tr> <td>ロウ</td> <td>ハイ</td> <td>ハイ</td> </tr> <tr> <td>ハイ</td> <td>ロウ</td> <td>ハイ</td> </tr> <tr> <td>ハイ</td> <td>ハイ</td> <td>ロウ</td> </tr> </tbody> </table> <p>すなわち、INT端子の割り込みは、立ち上がりエッジ（ロウ・レベルからハイ・レベルに変化したとき）のみ発生するので、P0D₁の出力によりINT端子に入力するレベルを切り替えます。 P0D₁端子の出力レベルとINT端子による割り込み発生関係を以下に示します。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>P0D₁レベル</th> <th>INT端子の割り込み発生条件</th> </tr> </thead> <tbody> <tr> <td>ハイ</td> <td>リモコンIC出力の立ち上がりエッジ</td> </tr> <tr> <td>ロウ</td> <td>リモコンIC出力の立ち上がりエッジ</td> </tr> </tbody> </table>	リモコンIC出力	P0D ₁	INT入力	ロウ	ロウ	ロウ	ロウ	ハイ	ハイ	ハイ	ロウ	ハイ	ハイ	ハイ	ロウ	P0D ₁ レベル	INT端子の割り込み発生条件	ハイ	リモコンIC出力の立ち上がりエッジ	ロウ	リモコンIC出力の立ち上がりエッジ
リモコンIC出力	P0D ₁	INT入力																					
ロウ	ロウ	ロウ																					
ロウ	ハイ	ハイ																					
ハイ	ロウ	ハイ																					
ハイ	ハイ	ロウ																					
P0D ₁ レベル	INT端子の割り込み発生条件																						
ハイ	リモコンIC出力の立ち上がりエッジ																						
ロウ	リモコンIC出力の立ち上がりエッジ																						

・タイマ：8ビット・タイマ

〈初期設定〉

シーケンス・カウンタ値：SEQ_CNTH=0, SEQ_CNTL=1

ステータス・データをクリア

8ビット・タイマにリーダのハイ・レベル時間の判定最大時間を設定し、割り込みを許可

INT端子の割り込みを許可

〈起動方法〉

INT端子の割り込みが発生すると、本プログラムが呼び出されます。

ドキュメント

AS17K Z1.10 D1 << E17201A DOCUMENT >> 12:00:23 02/26/91

TABLE OF CONTENTS

	PAGE
第 5 章 リモコン応用 (受信)	1
<プログラム説明>	1
1. INT 端子割り込みルーチン	2
2. タイマ割り込みルーチン	11

第 5 章 リモコン応用 (受信)

```

PUBLIC (MEM) : CUSTOM1, CUSTOM2, CUSTOM3, CUSTOM4, DATA1, DATA2, SEQ_H, SEQ_L
PUBLIC (FLG) : CODE_IN, INT_EG, REP_CODE, REP_IN
PUBLIC (LAB) : E1_RET1, REMCON_INT, TIMER_INT
ADDR RANGE : 0005H - 00C7H
    
```

<プログラム説明>

本プログラムは、μPD17201Aを使った、赤外線リモコンの受信用プログラムです。
以下に、本プログラムでのリモコンの受信方法を説明します。

(1) 端子

使用する端子は、INT端子とPOD1端子の2端子です。

・INT端子

リモコン波形の入力用端子です。プリアンプからの出力をPOD1端子の出力とをEORした出力レベルが入力されます。
INT端子は、立ち上がりエッジでのみ割り込みを発生することができます。

・POD1端子

INT端子に入力する入力レベルを制御するための出力です。
この出力を制御することにより、INT端子の割り込みエッジを制御することが可能です。

(2) タイマ

タイマでは、INT端子の割り込みに同期して、リモコンのハイ・レベル時間およびロウ・レベル時間を測定します。

(3) 割り込み

使用する割り込みは、INT端子の割り込みとタイマの割り込みの2つです。

・INT割り込み

INT割り込みは、INT端子がロウ・レベルからハイ・レベルに変化したとき(立ち上がりエッジ検出時)に発生します。
本プログラムでは、POD1端子の出力レベルを変えることで、プリアンプ出力の立ち上がりおよび立ち下がりエッジでINT割り込みを発生することができます。

・タイマ割り込み

タイマ割り込みは、タイマのモジュロ・レジスタの値と、タイマ値が一致したときに発生します。
本プログラムでは、タイマのモジュロ・レジスタに、受信時の最大測定値を設定しておきタイマの割り込みが発生したとき
に、エラー処理などを行います。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

1. INT端子割り込みルーチン

本ルーチンは、INT端子の割り込みにより受信したリモコン・コードのハイ・レベルおよびロウ・レベル区間の時間を測定し、受信したコードがN E Cコードであるかを判定するルーチンです。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

・システム・レジスタの退避

本ルーチンで使用されるシステム・レジスタの内容を退避領域に退避します。それと同時に、退避したシステム・レジスタを初期化します。

μ P D 1 7 2 0 1 Aでは、バンク・レジスタ (BANK) とプログラム・ステータス・ワード (PSW) の内容は、割り込みが発生すると自動的に割り込みスタックに退避します。このとき、バンク・レジスタの内容は、自動的に " 0 " となります。本ルーチンでは、上記の内容のほかに、ウィンドウ・レジスタとレジスタ・ポインタの内容を変更しますので退避して置く必要があります。そして、レジスタ・ポインタをバンク0の1番目のロウ・アドレスに設定します。

```

ENTRANCES          :REMCON_INT
MEMORIES CHANGED   :D_RPL          D_WR
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

・タイマ値の入力

タイマ値を読み込み、タイマを初期化し、再スタートさせます。この時点で、以前INT割り込みがあったときから、今回の割り込みまでの時間がデータ・バッファ (D B F) に設定されます。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

・受信処理

この処理で、実際の受信を行います。受信を行うときの状態を表すのに、シーケンス・カウンタ (SEQ_H, SEQ_L) を使います。シーケンス・カウンタは、バンク0のメモリの10Hおよび11H番地に設定されています。シーケンス・カウンタの値による動作状態を以下に示します。

SEQ_H, SEQ_L	動作説明
0 1	リーダのロウ・レベル部を受信を行います。データ・コードかを判断します。
0 2-1 1	データ・コードのカスタム部を受信します。 偶数カウンタ：カスタム部のハイ・レベル部。 奇数カウンタ：カスタム部のロウ・レベル部。
1 2-2 1	データ・コードのカスタム'部を受信します。 偶数カウンタ：カスタム'部のハイ・レベル部。 奇数カウンタ：カスタム'部のロウ・レベル部。
2 2-3 1	データ・コードのデータ部を受信します。 偶数カウンタ：データ部のハイ・レベル部。 奇数カウンタ：データ部のロウ・レベル部。
3 2-4 1	データ・コードの反転データ部を受信します。 偶数カウンタ：反転データ部のハイ・レベル部。 奇数カウンタ：反転データ部のロウ・レベル部。
4 2	ストップ・ビットまたは、リピート・ビットを受信します。
4 3	コード受信終了後のロウ・レベル時間を判断します。 ここでは、3ms間ロウ・レベルであればコード受信終了となります。
4 4-D F	未使用カウンタ値。
E 0-E F	受信エラー後の受信状態を示します。 3ms間ロウ・レベルとなる状態を探しカウンタ値をF0に設定します。
F 0-F F	コード受信終了を判定するための状態を示します。 150ms間ロウ・レベルであれば、コード受信を終了しカウンタ値を0に設定します。150msの間にコード入力を確認すれば、コード受信を継続します。

- ENTRANCES :-
- MEMORIES CHANGED :-
- MEMORIES REFERRED :-
- MEMORIES MANIPULATED :-
- FLAGS CHANGED :-
- FLAGS REFERRED :-
- DATA REFERRED :-
- BRANCH TO :-
- SUBROUTINES CALLED :-
- LABELS MANIPULATED :-
- SYSTEM CALL :-

- ・受信エラー処理
受信時にエラーとなれば、その時点で受信コードの判定を止め、3ms間ロウ・レベルと判定するまで読み飛ばします。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :_INT_EG      ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

- ・再受信処理
コード受信終了判定中にコードの入力が確認されるとコードの連続受信となります。しかし、リビート・コード受信後の70ms以内にコード入力が確認された場合には、受信エラーとなります。
コードの連続受信と判定されると、タイマにリーダーのハイ・レベル最大時間を設定します。

```

ENTRANCES          :-
MEMORIES CHANGED   :-SEQ_H      SEQ_L
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-REP_IN
DATA REFERRED      :-LEAD_MAX
BRANCH TO          :_INT_EG      ;
                   :CODE_ERR     ;ロウ・レベル区間が75ms以内であればエラー
                   :REM_IN       ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

・リモコン・コード受信
リモコンを受信し、シーケンス・カウンタの値に従って、各ルーチンの処理を行うために、分岐します。

```

ENTRANCES          :REM_IN
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :CODE_DAT      ;コード部分の受信
                   :CODE_REP      ;リーダ部分のロウ・レベルを受信
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
    
```

(1) リーダ受信処理
リーダ部分の受信判定を行います。

(a) リーダのハイ・レベル部の受信処理
リーダのハイ・レベル部は、通常 9ms ありますが、電池の消耗や反射信号などにより短くなる場合があります。従って、本ルーチンでは、受信範囲を 8ms から 9.9ms に設定しています。本ルーチンでは、最大値の判定は、タイマ割り込みで行います。よって、ここでは、最小値のみの判定を行います。もし、最小値よりリーダのハイ・レベルが短ければ、受信エラーとなります。

受信範囲	6.0ms - 9.9ms
------	---------------

範囲内であれば、タイマにリポート時のロウ・レベル最大時間を設定します。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :REP_CODE
FLAGS REFERRED      :INT_EG
DATA REFERRED       :LEAD_MIN      REP_MAX
BRANCH TO          :_INT_EG        ;リーダ部の立ち下がり検出に設定
                   :CODE_ERR
                   :SEQ_INC
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
    
```

(b) リーダーのロウ・レベル部の受信処理は、リピート受信時とデータ受信時とは異なります。リピート時の最大時間をタイマに設定し、タイマの割り込みが発生したら、タイマにデータの最大時間を設定するようリピートします。データ受信時とデータ受信時とは異なります。リピート時の最大時間をタイマに設定し、タイマの割り込みが発生したら、タイマにデータの最大時間を設定するようリピートします。データ受信時とデータ受信時とは異なります。リピート時の最大時間をタイマに設定し、タイマの割り込みが発生したら、タイマにデータの最大時間を設定するようリピートします。

REP_CODE	タイマ設定状態
0	リピート・コード入力状態
1	データ・コード入力状態

受信状態	受信範囲
リピート受信	1.0ms - 2.8ms
データ受信	3.5ms - 5.4ms

受信範囲外であれば、受信エラーとなります。受信範囲内であれば、ストップ・ビットの最大時間をタイマに設定します。リピート時であれば、カウンタを42Hに設定し、リピート・コードの最大時間をタイマに設定します。

```

ENTRANCESS          :CODE_REP
MEMORIES CHANGED    :_CODE_H      _CODE_L
                    :CODE_H      CODE_L
MEMORIES REFERRED    :-
MEMORIES MANIPULATED :-
FLAGS CHANGED        :-
FLAGS REFERRED       :REP_CODE
DATA REFERRED        :DAT_MIN
BRANCH TO            :BIT_SET      ;
                    :CODE_ERR      ;
                    :REPEAT        ;リピート時間を受信
SUBROUTINES CALLED   :-
LABELS MANIPULATED  :-
SYSTEM CALL          :-
    
```

(2) リピート・コードの受信処理
 リピート・コード受信を判定します。
 タイマ設定時間中に、INT端子の割り込みが発生すると、リピート・コードと判断します。
 リピート・コードと判断すると、シーケンス・カウンタを41Hに設定し、次のコード待ちとなります。

```

ENTRANCES          :REPEAT
MEMORIES CHANGED   :SEQ_H          SEQ_L
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED      :-
DATA REFERRED       :REP_MIN
BRANCH TO           :BIT_SET      ;
                   :CODE_ERR      ;

SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
    
```


- (3) データ・コードの受信処理
 NECフォーマットのデータ・コードは、カスタム部、カスタム'部、データ部、反転データ部と4部分に分かれています。
 4部ともそれぞれ8ビットで構成されています。
 0/1の判定は、0.56msのハイ・レベルに続くロウ・レベルの時間で決まります。以下に、0/1判定を行うロウ・レベルの受信範囲を示します。

ビット	受信範囲
0	0.2ms - 1.5ms
1	1.5ms - 2.4ms

ハイ・レベル受信範囲	0.2ms - 1.1ms
------------	---------------

範囲外であれば、受信エラーとなります。
 カスタム部、およびカスタム'部を受信後、決められたカスタム・データと比較します。正しければ、続くデータ部を受信します。
 データ部、および反転データ部を受信後、反転データを反転し、データと比較します。同じデータであれば、ストップ・ビットの受信を行います。

```

ENTRANCES          :CODE_DAT
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :BIT_P
FLAGS REFERRED      :-
DATA REFERRED       :BITO_MAX      BIT_MIN
                    LOW_TIME
BRANCH TO           :BIT_CHK      ;
                    CODE_ERR      ;
                    SEQ_INC       ;
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
    
```

- (4) エラー処理
 リモコン受信中に受信範囲以外のコードや範囲外のビット数を受信したときに、エラー処理を行います。
 エラーと判断すると、それ以外のコードは無視され、一定時間のロウ・レベル区間のみの検出処理を行います。
 一定時間のロウ・レベル区間が見えされると、次のリーダー・コード待ちとなります。

```

ENTRANCES          :CODE_ERR
MEMORIES CHANGED   :SEQ_H
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :LOW_TIME
BRANCH TO          :_INT_EG      ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

- (5) ビット・データの設定処理
 データ部の受信が終了すると、ビット・データをメモリ(CODE_H, CODE_L, _CODE_H, _CODE_L)に設定し、16ビット設定が完了するとデータとデータの反転データとを比較し、一致するとデータの設定処理を行います。

```

ENTRANCES          :BIT_CHK
MEMORIES CHANGED   :_CODE_H      _CODE_L
                   :CODE_H      CODE_L
                   :CUSTOM1     CUSTOM2
                   :CUSTOM3     CUSTOM4
                   :DATA1       DATA2
MEMORIES REFERRED   :_CODE_H      _CODE_L
                   :CODE_H      CODE_L
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :BIT_F
DATA REFERRED      :BITO_MIN
BRANCH TO          :BIT_SET      ;
                   :CODE_ERR     ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

- (6) ビット・データのキャリア部の時間設定
 ビット・データおよびストップ・ビットのキャリア部の最大時間をタイマに設定します。

```

ENTRANCES          :BIT_SET
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :BIT_MAX
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

- ・ 割り込み時の後処理
各割り込み処理が終了するごとに、カウンタの値をインクリメントし、割り込みエッジの反転を行います。
最後に、退避したシステム・レジスタの内容を復帰し、割り込みポイントに復帰します。

```

ENTRANCES          :_INT_EG      EI_RET1
                   :RETURN      SEQ_INC
MEMORIES CHANGED   :SEQ_H        SEQ_L
MEMORIES REFERRED  :D_RPL        D_WR
MEMORIES MANIPULATED :-
FLAGS CHANGED      :INT_EG
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

2. タイマ割り込みルーチン
 タイマ割り込みルーチンは、リモコン受信時には、エラー発生時の割り込み処理として使用されます。
 主に、リモコン波形の受信最大幅超過を越えたときに処理を行います。
 そのほかには、データ・コード受信とリピート・コード受信の判定、ビット0とビット1データの判定を行う処理も含まれて
 います。

```

ENTRANCES          :TIMER_INT
MEMORIES CHANGED   :D_RPL          D_WR
MEMORIES REFERRED  :--
MEMORIES MANIPULATED :--
FLAGS CHANGED      :--
FLAGS REFERRED     :--
DATA REFERRED      :--
BRANCH TO          :--
SUBROUTINES CALLED :--
LABELS MANIPULATED :--
SYSTEM CALL        :--
    
```

(1) エラー処理
 リモコン受信時にエラーとなったときの、一定時間(3ms)ロウ・レベルになったかを判定するために処理です。

```

ENTRANCES          :SEQ_OP
MEMORIES CHANGED   :SEQ_H          SEQ_L
MEMORIES REFERRED  :--
MEMORIES MANIPULATED :--
FLAGS CHANGED      :--
FLAGS REFERRED     :INT_EG
DATA REFERRED      :LOW_BASE
BRANCH TO          :REM_OK          ;
                   :RETURN          ;
SUBROUTINES CALLED :--
LABELS MANIPULATED :--
SYSTEM CALL        :--
    
```

(2) リモコン受信の終了
 リモコン受信が終了したかを判定するための処理です。
 受信が終了するとデータ・コードならCODE_INフラグがセットされ、リピート・コードならREP_INフラグがセットされます。

```

ENTRANCES          :REM_OK
MEMORIES CHANGED   :--
MEMORIES REFERRED  :--
MEMORIES MANIPULATED :--
FLAGS CHANGED      :CODE_IN        REP_IN
FLAGS REFERRED     :REP_CODE
DATA REFERRED      :--
BRANCH TO          :NEXT_IN        ;
                   :SEQ_OP        ;
SUBROUTINES CALLED :--
LABELS MANIPULATED :--
SYSTEM CALL        :--
    
```

(3) リモコン受信の終了判定
 データ・コードまたはリピート・コードの受信後、150ms間リモコンの受信が確認できない場合には、受信終了と判定し
 タイマ動作を停止します。

```

ENTRANCES          :NEXT_IN
MEMORIES CHANGED   :SEQ_H          SEQ_L
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :INT_EG          REP_IN
FLAGS REFERRED      :-
DATA REFERRED       :LEAD_MAX
BRANCH TO           :CODE_CHK        ;コード受信中
                   :RETURN          ;
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
    
```

(4) コード受信時の処理
 コード受信中に、タイマの割り込みが発生するには、以下の状態があります。

1. リーダ・コード受信時に、リーダ・コードの受信最大時間を越えた場合
2. データ・コード/リピート・コード判断時に、リピート・コード判定時間の最大時間を越えたとき、およびデータ・コード判断時にデータ・コード判定の最大時間を越えた場合
3. ビットのキャリア部の時間測定時に、受信最大時間を越えた場合
4. ビットのロウ・レベルの時間測定時に、ビット0の最大判断時間を越えたとき、ビット1の最大判断時間を越えた場合

```

ENTRANCES          :CODE_CHK
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :BIT_F          REP_CODE
FLAGS REFERRED      :BIT_F          REP_CODE
DATA REFERRED       :BITI_MAX        DAT_MAX
BRANCH TO           :CODE_ERR        ;
                   :RETURN          ;
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
    
```



RAMマップ



AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 11:04:15 02/25/91 PAGE 02-001

PROG -

SOURCE - REM.ASM

(PUBLIC/LOCAL MAP)

BANK - 0

ROW

	7	6	5	4	3	2	1	0	
		LCDD32	LCDD16	LCDD0			SEQ_H	CUSTOM1	0
		ID_WR				REM_FLG			
		LCDD33	LCDD17	LCDD1			SEQ_L	CUSTOM2	1
		ID_RPL							
		LCDD34	LCDD18	LCDD2				CUSTOM3	2
							CODE_H		C
		LCDD35	LCDD19	LCDD3				CUSTOM4	3
							CODE_L		
AR3			LCDD20	LCDD4				DATA1	4
							CODE_H		
AR2			LCDD21	LCDD5				DATA2	5
							CODE_L		
AR1			LCDD22	LCDD6					6
AR0			LCDD23	LCDD7					7
WR			LCDD24	LCDD8					8
BANK			LCDD25	LCDD9					9
IXH			LCDD26	LCDD10					A
IMPH									
IXM			LCDD27	LCDD11					B
IMPL									
IXL			LCDD28	LCDD12				DBP3	C
IRPH			LCDD29	LCDD13				DBP2	D

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 11:04:15 02/25/91 PAGE 02-002

PROG -

SOURCE - REM. ASM

(PUBLIC/LOCAL MAP) BANK - 0

		ROM								
		7	6	5	4	3	2	1	0	
IRPL				LCDD30	LCDD14				IDBF1	E
IPSW				LCDD31	LCDD15				IDBF0	F

AS17K Z1.10 D1 << E17201A FLG MAP >> 11:04:15 02/25/91 PAGE 02-003

PROG -

SOURCE - REM.ASM

(FLAG MAP) BANK - 0

MSB	3	2	1	0	LSB
20	I REP_IN	I CODE_IN	I BIT_F	I REP_CODE	
70	I POA3	I POA2	I POA1	I POA0	
71	I POB3	I POB2	I POB1	I POB0	
72	I POC3	I POC2	I POC1	I POC0	
73	I POD3	I POD2	I INT_EG I POD1	I POD0	
7A	I MPR				
7E				I BCD	
7F	I CMP	I CY	I Z	I IXB	
82			I SYSCK	I XEN	
83	I WDTRES	I WTMMD	I WTMRES		
87	I VDDET1	I VDDET0			
8F				I INT	
91				I NRZBF	
92				I NRZ	
A0				I ADCCMP	
A1	I VREPEN	I ADCEN	I ADCCH1	I ADCCH0	
A2	I SIOTS	I SIOHIZ	I SIOCK1	I SIOCK0	

AS17K Z1.10 D1 << E17201A PLG MAP >> 11:04:15 02/25/91 PAGE 02-004

PROG -

SOURCE - REM.ASM

(FLAG MAP) BANK - 0

MSB 3 2 1 0 LSB
----->

AS17K Z1.10 D1 << B17201A FLG MAP >> 11:04:15 02/25/91 PAGE 02-005

PROG =

SOURCE = REM.ASM

(FLAG MAP) BANK = 0

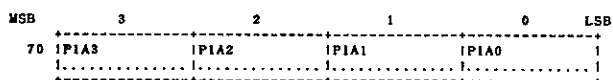
MSB	3	2	1	0	LSB
A3		INRZEN	TMOE	SIOEN	
A7	P0DBIO3	P0DBIO2	P0DBIO1	P0DBIO0	
AF	IPSIO	IPWTM	IP	IPTM	
B1	LCDEN	LCDCK2	LCDCK1	LCDCK0	
B2	LCDMD3	LCDMD2	LCDMD1	LCDMD0	
B3	TMEN	TMRES	TMCK1	TMCK0	
B7	PIAGIO	P0CGIO	P0BGIO	P0AGIO	
BB	IRQSIO				
BC		IRQTM			
BD			IRQ		
BE				IRQTM	

AS17K Z1.10 D1 << E17201A PLG MAP >> 11:04:15 02/25/91 PAGE 02-006

PROG -

SOURCE = REM.ASM

(PLG MAP) BANK = 1



RAM使用説明



AS17K Z1.10 D1 << R17201A DATA MEMORY MAP >> 11:04:15 02/25/01 PAGE 02-007

PROG -

SOURCE - REM.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
_CODE_H	MEM	0	14		反転受信コードの設定用メモリ(上位)
_CODE_L	MEM	0	15		反転受信コードの設定用メモリ(下位)
ADCCH0	FLG	0	A1	...1	
ADCCH1	FLG	0	A1	..1.	
ADCCMP	FLG	0	A0	...1	
ADCBN	FLG	0	A1	..1.	
AR0	MEM	0	77		
AR1	MEM	0	76		
AR2	MEM	0	75		
AR3	MEM	0	74		
BANK	MEM	0	79		
BDC	FLG	0	78	...1	
BIT_F	FLG	0	20	..1.	ビット判定用フラグ
CMF	FLG	0	7F	1...	
CODE_H	MEM	0	12		受信コードの設定用メモリ(上位)
CODE_IN	FLG	0	20	..1.	リモコンのデータ・コード入力完了
CODE_L	MEM	0	13		受信コードの設定用メモリ(下位)
CUSTOM1	MEM	0	0		カスタム・コード設定メモリ
CUSTOM2	MEM	0	1		カスタム・コード設定メモリ
CUSTOM3	MEM	0	2		カスタム・コード設定メモリ
CUSTOM4	MEM	0	3		カスタム・コード設定メモリ
CY	FLG	0	7F	..1.	
D_RPL	MEM	0	81		割り込み時のRPL退避用メモリ
D_WR	MEM	0	80		割り込み時のWR退避用メモリ
DATA1	MEM	0	4		データ・コード設定メモリ
DATA2	MEM	0	5		データ・コード設定メモリ
DBF0	MEM	0	F		
DBF1	MEM	0	E		
DBF2	MEM	0	D		
DBF3	MEM	0	C		
INT	FLG	0	8F	...1	
INT_EG	FLG	0	73	..1.	リモコンの割り込みエッジ決定用ポート出力
IP	FLG	0	AF	..1.	
IPS10	FLG	0	AF	1...	
IPTM	FLG	0	AF	...1	
IPWTM	FLG	0	AF	..1.	
IRQ	FLG	0	BD	..1.	
IRQS10	FLG	0	BB	1...	
IRQTM	FLG	0	BE	...1	
IRQWTM	FLG	0	BC	..1.	
IXE	FLG	0	7F	...1	
IXH	MEM	0	7A		
JXL	MEM	0	7C		
JXM	MEM	0	7B		
LCDCCK0	FLG	0	B1	...1	
LCDCCK1	FLG	0	B1	..1.	
LCDCCK2	FLG	0	B1	..1.	
LCDD0	MEM	0	40		
LCDD1	MEM	0	41		
LCDD10	MEM	0	4A		
LCDD11	MEM	0	4B		
LCDD12	MEM	0	4C		

AS17K Z1.10 DJ << E17201A DATA MEMORY MAP >> 11:04:16 02/25/91 PAGE 02-008

PROG =

SOURCE = REM.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCDD13	MEM	0	4D		
LCDD14	MEM	0	4E		
LCDD15	MEM	0	4F		
LCDD16	MEM	0	50		
LCDD17	MEM	0	51		
LCDD18	MEM	0	52		
LCDD19	MEM	0	53		
LCDD2	MEM	0	42		
LCDD20	MEM	0	54		
LCDD21	MEM	0	55		
LCDD22	MEM	0	56		
LCDD23	MEM	0	57		
LCDD24	MEM	0	58		
LCDD25	MEM	0	59		
LCDD26	MEM	0	5A		
LCDD27	MEM	0	5B		
LCDD28	MEM	0	5C		
LCDD29	MEM	0	5D		
LCDD3	MEM	0	43		
LCDD30	MEM	0	5E		
LCDD31	MEM	0	5F		
LCDD32	MEM	0	60		
LCDD33	MEM	0	61		
LCDD34	MEM	0	62		
LCDD35	MEM	0	63		
LCDD4	MEM	0	44		
LCDD5	MEM	0	45		
LCDD6	MEM	0	46		
LCDD7	MEM	0	47		
LCDD8	MEM	0	48		
LCDD9	MEM	0	49		
LCDEN	FLG	0	B1	...	
LCDDM0	FLG	0	B2	...	
LCDDM1	FLG	0	B2	...	
LCDDM2	FLG	0	B2	...	
LCDDM3	FLG	0	B2	...	
MPE	FLG	0	7A	...	
MPH	MEM	0	7A		
MPL	MEM	0	7B		
NRZ	FLG	0	92	...	
NRZBF	FLG	0	91	...	
NRZEN	FLG	0	A3	...	
POA0	FLG	0	70	...	
POA1	FLG	0	70	...	
POA2	FLG	0	70	...	
POA3	FLG	0	70	...	
POAGIO	FLG	0	B7	...	
POB0	FLG	0	71	...	
POB1	FLG	0	71	...	
POB2	FLG	0	71	...	
POB3	FLG	0	71	...	
POBGIO	FLG	0	B7	...	

AS17K Z1.10 D1 << B17201A DATA MEMORY MAP >> 11:04:15 02/25/91 PAGE 02-009

PROG =

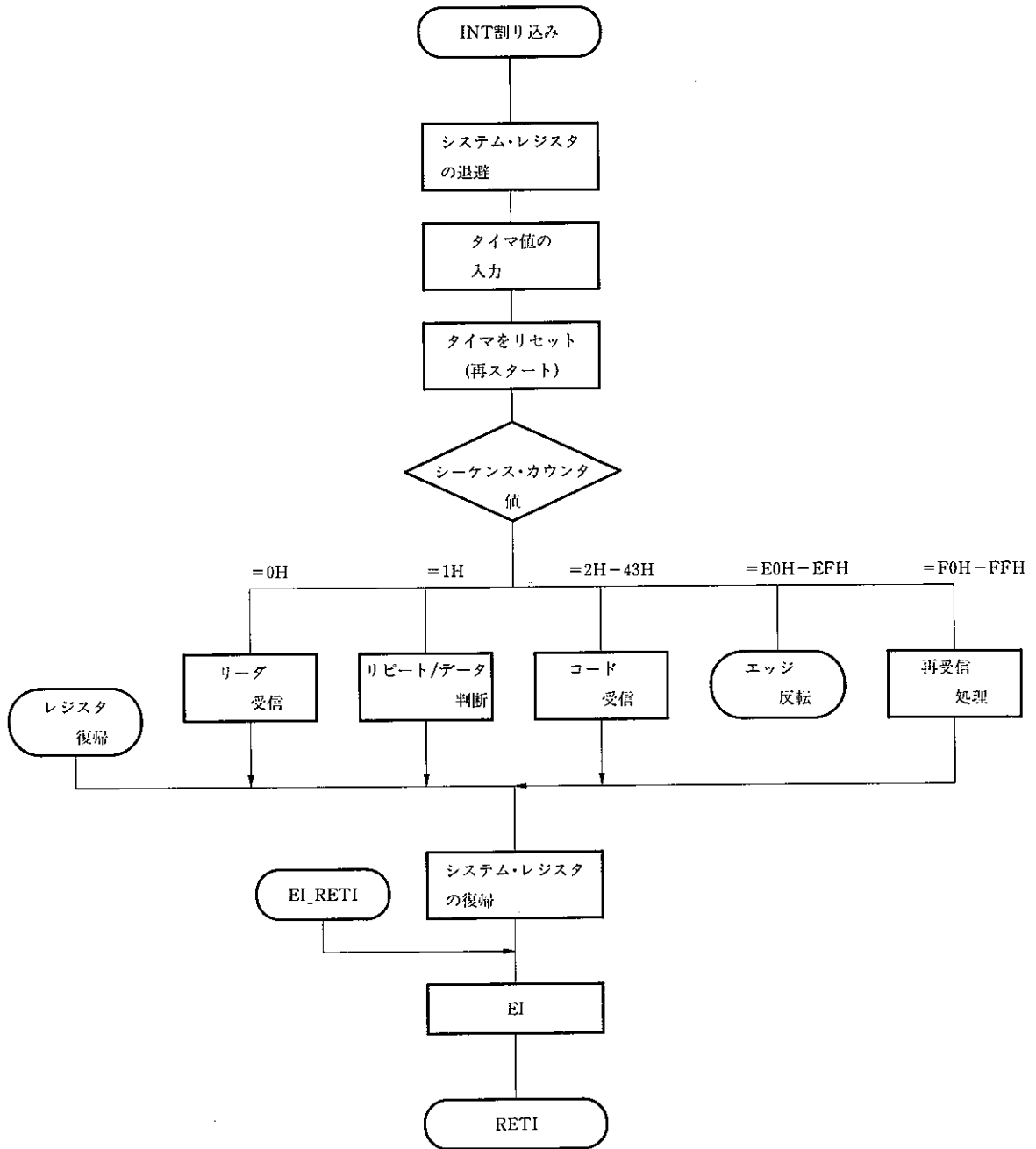
SOURCE = REM.ASM

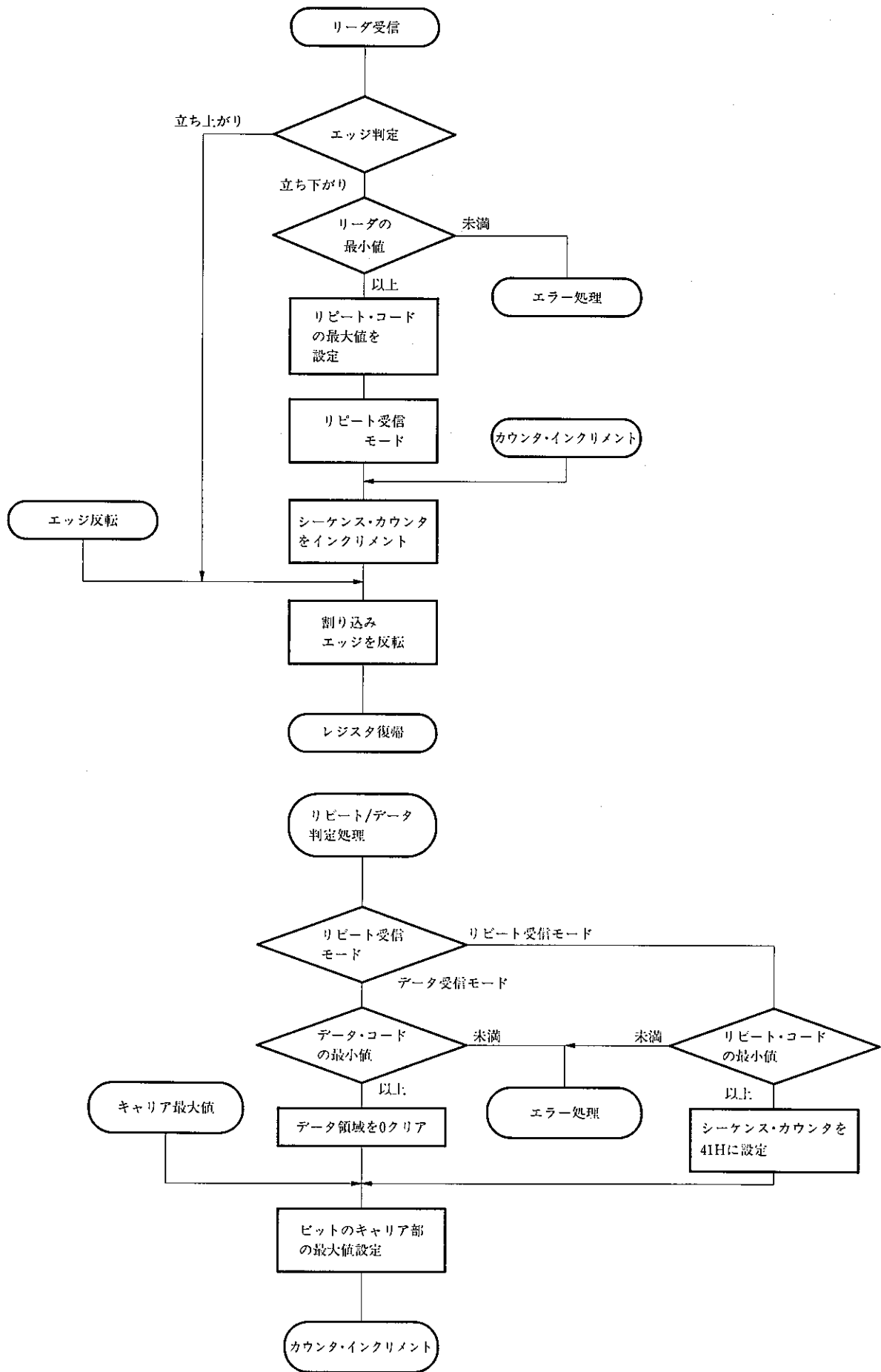
(DATA AREA INFORMATION)

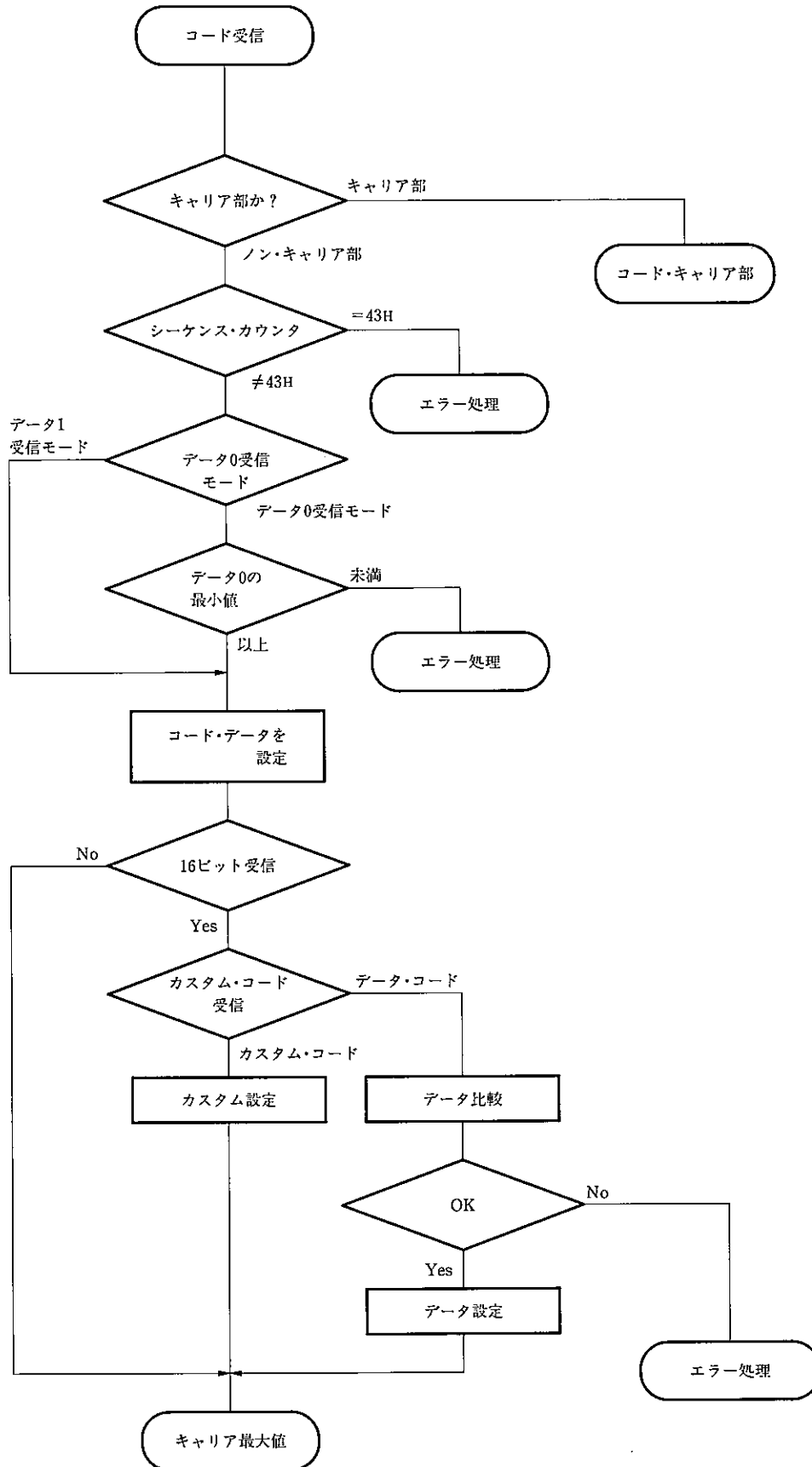
SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
P0C0	FLG	0	72	...1	
P0C1	FLG	0	72	..1.	
P0C2	FLG	0	72	.1..	
P0C3	FLG	0	72	1...	
P0CGIO	FLG	0	B7	.1..	
P0D0	FLG	0	73	...1	
P0D1	FLG	0	73	..1.	
P0D2	FLG	0	73	.1..	
P0D3	FLG	0	73	1...	
P0DBIO0	FLG	0	A7	...1	
P0DBIO1	FLG	0	A7	..1.	
P0DBIO2	FLG	0	A7	.1..	
P0DBIO3	FLG	0	A7	1...	
P1A0	FLG	1	70	...1	
P1A1	FLG	1	70	..1.	
P1A2	FLG	1	70	.1..	
P1A3	FLG	1	70	1...	
P1AGIO	FLG	0	B7	1...	
PSW	MEM	0	7F		
REM_FLG	MEM	0	20		リモコン受信用フラグ設定メモリ
REP_CODE	FLG	0	20	...1	リビート・コード受信状態設定用フラグ
REP_IN	FLG	0	20	1...	リモコン・コード入力中
RPH	MEM	0	7D		
RPL	MEM	0	7E		
SEQ_H	MEM	0	10		リモコン受信用の動作カウンタ(上位)
SEQ_L	MEM	0	11		リモコン受信用の動作カウンタ(下位)
SIOCK0	FLG	0	A2	...1	
SIOCK1	FLG	0	A2	..1.	
SIOEN	FLG	0	A3	...1	
SIOHIZ	FLG	0	A2	.1..	
SIOIS	FLG	0	A2	1...	
SYSCK	FLG	0	82	...1	
TMCK0	FLG	0	B3	...1	
TMCK1	FLG	0	B3	..1.	
TMEN	FLG	0	B3	1...	
TMOE	FLG	0	A3	..1.	
TMR5	FLG	0	B3	.1..	
VDDDET0	FLG	0	87	.1..	
VDDDET1	FLG	0	87	1...	
VREFEN	FLG	0	A1	1...	
WDTRES	FLG	0	83	1...	
WR	MEM	0	78		
WTMMD	FLG	0	83	.1..	
WTMRES	FLG	0	83	1...	
XEN	FLG	0	82	...1	
Z	FLG	0	7F	..1.	

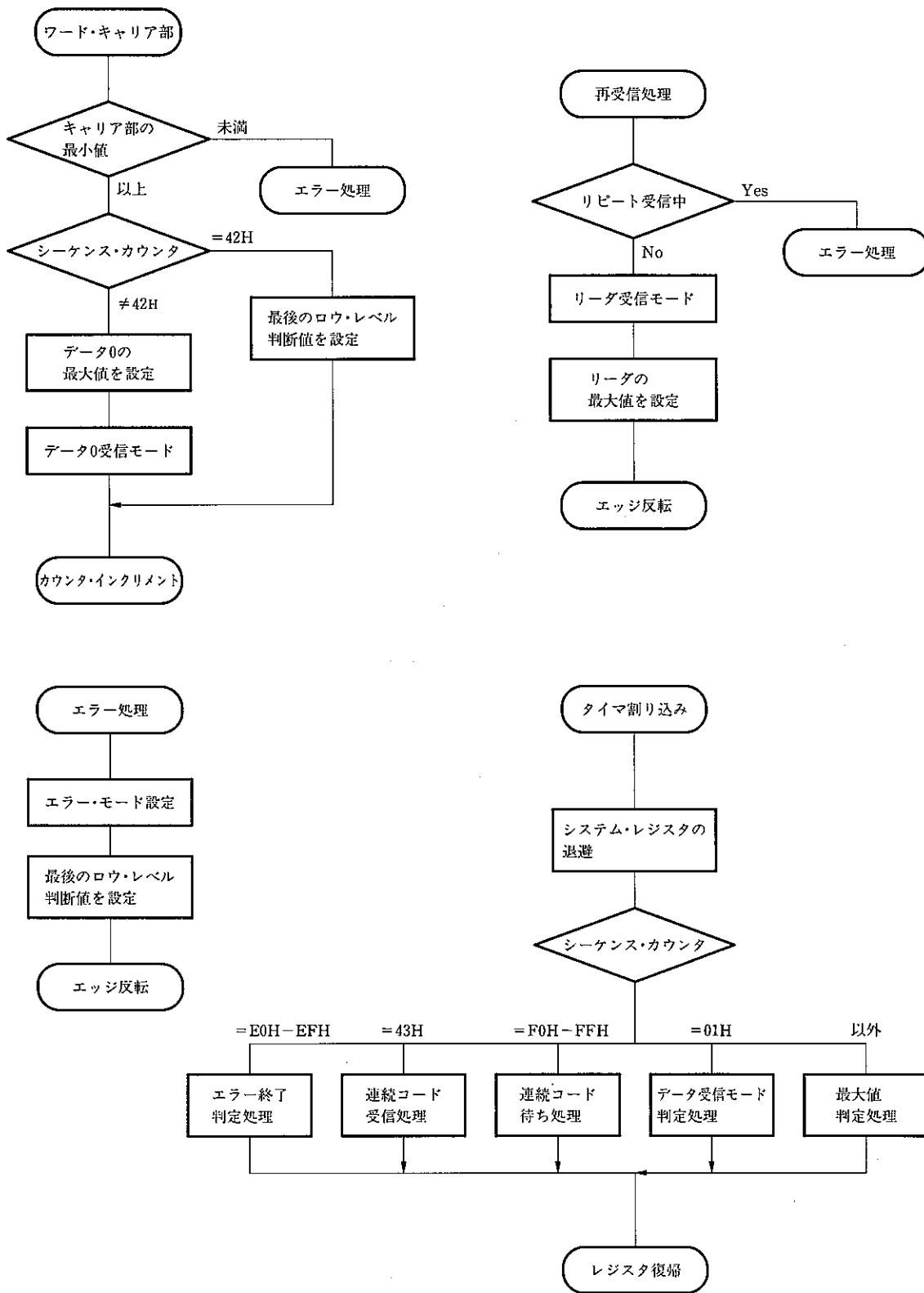
フロー・チャート

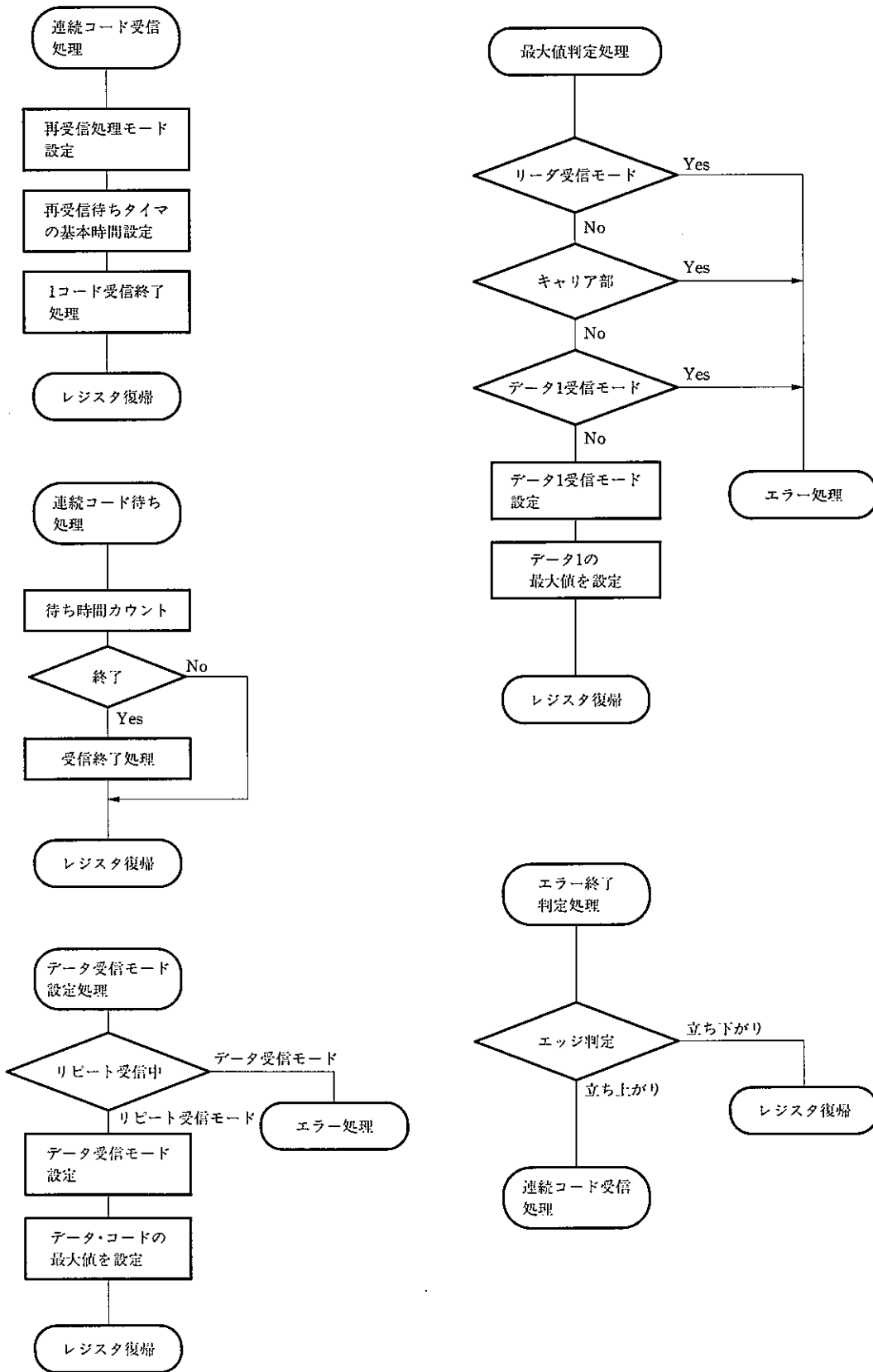
フロー・チャート











リスト


```

;.....
;*** リモコン受信プログラム
;.....
SUMMARY '%%', '第 5 章 リモコン応用 (受信)'
%%
SUMMARY '%%', '<プログラム説明>'
本プログラムは、μPD17201Aを使った、赤外線リモコンの受信プログラムです。
以下に、本プログラムでのリモコンの受信方法を説明します。

(1) 端子
使用する端子は、INT端子とPODI端子の2端子です。

・INT端子
リモコン波形の入力用端子です。プリアンプからの出力をPODI端子の出力とをEORした出力レベルが入力されます。
INT端子は、立ち上がりエッジでのみ割り込みを発生することができます。

・PODI端子
INT端子に入力する入力レベルを制御するための出力です。
この出力を制御することにより、INT端子の割り込みエッジを制御することが可能です。

(2) タイマ
タイマでは、INT端子の割り込みに同期して、リモコンのハイ・レベル時間およびロウ・レベル時間を測定します。

(3) 割り込み
使用する割り込みは、INT端子の割り込みとタイマの割り込みの2つです。

・INT割り込み
INT割り込みは、INT端子がロウ・レベルからハイ・レベルに変化したとき(立ち上がりエッジ検出時)に発生します。
本プログラムでは、PODI端子の出力レベルを変えることで、プリアンプ出力の立ち上がりおよび立ち下がりがエッジでINT割り
込みを発生することができます。

・タイマ割り込み
タイマ割り込みは、タイマのモジュロ・レジスタの値と、タイマ値が一致したときに発生します。
本プログラムでは、タイマのモジュロ・レジスタに、受信時の最大測定値を設定しておきタイマの割り込みが発生したとき
に、エラー処理などを行います。

```

```

.EJ-1
%%
;-----
;          P U B L I C 重 音
;-----
;--- LABEL ---
PUBLIC REMCON_INT
PUBLIC TIMER_INT
PUBLIC EI_RET

;--- FLAG ---
PUBLIC REP_CODE
PUBLIC CODE_IN
PUBLIC REP_IN
PUBLIC INT_EG

;--- MEMORY ---
PUBLIC SEQ_H, SEQ_L
PUBLIC CUSTOM1, CUSTOM2, CUSTOM3, CUSTOM4, DATA1, DATA2

EJECT
;.....
;          割 り 込 み ル ー チ ン
;.....
SUMMARY '%%', '1. INT端子割り込みルーチン'
本ルーチンは、INT端子の割り込みにより受信したリモコン・コードのハイ・レベルおよびロウ・レベル区間の時間を測定し、受信し
たコードがNECコードであるかを判定するルーチンです。
%%
;.....
;          I N T 端 子 割 り 込 み
;.....

```

```

:.....
:   メモリオよびデータ設定
:.....
://////
:   メモリ定義
://////
CUSTOM1    MEM    0.00H    ; カスタム・ユーザ設定メモリ
CUSTOM2    MEM    0.01H    ; カスタム・ユーザ設定メモリ
CUSTOM3    MEM    0.02H    ; カスタム・ユーザ設定メモリ
CUSTOM4    MEM    0.03H    ; カスタム・ユーザ設定メモリ
DATA1      MEM    0.04H    ; データ・ユーザ設定メモリ
DATA2      MEM    0.05H    ; データ・ユーザ設定メモリ

SEQ_H      MEM    0.10H    ; リモコン受信用の動作カウンタ (上位)
SEQ_L      MEM    0.11H    ; リモコン受信用の動作カウンタ (下位)

CODE_H     MEM    0.12H    ; 受信コードの設定用メモリ (上位)
CODE_L     MEM    0.13H    ; 受信コードの設定用メモリ (下位)
_CODE_H    MEM    0.14H    ; 反転受信コードの設定用メモリ (上位)
_CODE_L    MEM    0.15H    ; 反転受信コードの設定用メモリ (下位)

REM_FLG    MEM    0.20H    ; リモコン受信用フラグ設定メモリ
RSP_CODE   FLG    REM_FLG.0 ; リビートコード受信状態設定用フラグ
BIT_P      FLG    REM_FLG.1 ; ビット判定用フラグ
CODE_IN    FLG    REM_FLG.2 ; リモコンのデータ・コード入力完了
RSP_IN     FLG    REM_FLG.3 ; リモコン・コード入力中

INT_EG     FLG    PDD1    ; リモコンの割り込みエッジ決定用ポート出力

D_WR       MEM    0.60H    ; 割り込み時のWR退避用メモリ
D_RPL      MEM    0.61H    ; 割り込み時のRPL退避用メモリ

://////
:   データ設定
://////
LEAD_MAX   DAT    9BH-1    ; リーダ・コードの最大値 (9.92ms)
LEAD_MIN   DAT    65H-1    ; リーダ・コードの最小値 (6.464ms)

REP_MAX    DAT    2BH-1    ; リビート・コード検出時の最大値 (2.752ms)
REP_MIN    DAT    1CH-1    ; リビート・コード検出時の最小値 (1.792ms)

DAT_MAX    DAT    55H-1    ; データ・コード検出時の最大値 (5.44ms)
DAT_MIN    DAT    38H-1    ; データ・コード検出時の最小値 (3.584ms)

BIT_MAX    DAT    0FH-1    ; ビットのキャリア部の最大値 (0.96ms)
BIT_MIN    DAT    04H-1    ; ビットのキャリア部の最小値 (0.256ms)

BIT1_MAX   DAT    26H-1    ; ビット・データ1時の最大値 (2.432ms)

BIT0_MAX   DAT    14H-1    ; ビット・データ0時の最大値 (1.28ms)
BIT0_MIN   DAT    04H-1    ; ビット・データ0時の最小値 (0.256ms)

LOW_TIME   DAT    2PH-1    ; ビットなし判定時間 (3.008ms)
LOW_BASE   DAT    0EBH-1   ; リモコン終了判定基本時間 (15.04ms)

```

EJECT

```

:.....
:   割り込みルーチン
:.....

```

REMCON_INT:

SUMMARY "RM".

・システム・レジスタの退避

本ルーチンで使用するシステム・レジスタの内容を退避領域に退避します。それと同時に、退避したシステム・レジスタを初期化します。
 #PDD17201Aでは、バンク・レジスタ (BANK) とプログラム・ステータス・ワード (PSW) の内容は、割り込みが発生すると自動的に割り込みスタックに退避します。このとき、バンク・レジスタの内容は、自動的に"0"となります。

本ルーチンでは、上記の内容のほかに、ウィンドウ・レジスタとレジスタ・ポインタの内容を変更しますので退避して置く必要があります。そして、レジスタ・ポインタをバンク0の1番目のロウ・アドレスに設定します。

```

%%
:-----
: 割り込み時のレジスタの退避
:-----
POKE  D_WR, WR
PEEK  WR, RPL
POKE  D_RPL, WR
MOV   PSW, #D
MOV   RPL, #0010B
    
```

SUMMARY '%%', ''
 ・タイマ値の入力
 タイマ値を読み込み、タイマを初期化し、再スタートさせます。
 この時点で、以前INT割り込みがあったときから、今回の割り込みまでの時間がデータ・バッファ (DBF) に設定されます。

```

EJ-1
%%
:-----
: タイマ値の読み込み
:-----
GET   DBF, TMC

:-----
: タイマの再スタート
:-----
SET2  TMRES, TMEN
    
```

SUMMARY '%%', ''
 ・受信処理
 この処理で、実際の受信を行います。受信を行うときの状態を表すのに、シーケンス・カウンタ (SEQ_H, SEQ_L) を使います。シーケンス・カウンタは、バンク0のメモリの10Hおよび11H番地に設定されています。シーケンス・カウンタの値による動作状態を以下に示します。

SEQ_H, SEQ_L	動作説明
0 1	リーダーのロウ・レベル部の受信を行います。次の入力のリピート・コードか、データ・コードかを判断します。
0 2 - 1 1	データ・コードのカスタム部を受信します。 偶数カウンタ: カスタム部のハイ・レベル部。 奇数カウンタ: カスタム部のロウ・レベル部。
1 2 - 2 1	データ・コードのカスタム'部を受信します。 偶数カウンタ: カスタム'部のハイ・レベル部。 奇数カウンタ: カスタム'部のロウ・レベル部。
2 2 - 3 1	データ・コードのデータ部を受信します。 偶数カウンタ: データ部のハイ・レベル部。 奇数カウンタ: データ部のロウ・レベル部。
3 2 - 4 1	データ・コードの反転データ部を受信します。 偶数カウンタ: 反転データ部のハイ・レベル部。 奇数カウンタ: 反転データ部のロウ・レベル部。
4 2	ストップ・ビットまたは、リピート・ビットを受信します。
4 3	コード受信終了後のロウ・レベル時間を判断します。ここでは、3ms間ロウ・レベルであればコード受信終了となります。
4 4 - DF	未使用カウンタ値。
E 0 - EF	受信エラー後の受信状態を示します。3ms間ロウ・レベルとなる状態を探しカウンタ値をF0に設定します。

F O - F F	コード受信終了を判定するための状態を示します。 150ms間ロウ・レベルであれば、コード受信を終了しカウンタ値を0に 設定します。150msの間にコード入力を確認すれば、コード受信を継続 します。
-----------	---

```

.EJ-1
%%
SUMMARY '%%', ''
    ・受信エラー処理
    受信時にエラーとなれば、その時点で受信コードの判定を止め、3ms間ロウ・レベルと判定するまで読み飛ばします。
%%
-----
;
;   受信エラー時
;
SKNB  SEQ_H, #0EH          ;SEQ_Hの値が0EHであればエラー処理中。
BR     _INT_EG

SUMMARY '%%', ''
    ・再受信処理
    コード受信終了判定中にコードの入力が確認されるとコードの連続受信となります。しかし、リピート・コード受信後の70ms以
    内にコード入力が確認された場合には、受信エラーとなります。
    コードの連続受信と判定されると、タイマにリダのハイ・レベル最大時間を設定します。
.EJ-1
%%
-----
;
;   リダ・コードの再受信判定
;
SKE   SEQ_H, #0FH          ;SEQ_Hの値が0FHであれば、コードの再受信待ち
BR     REM_IN
SKT1  REP_IN              ;リピート時の再受信待ちか?
BR     RESTART
SKGB  SEQ_L, #5           ;ロウ・レベル区間の最低時間を測定
BR     CODE_ERR           ;ロウ・レベル区間が75ms以内であればエラー

RESTART:
MOV   DBF1, #LEAD_MAX SHR 4) AND 0FH ;リダ受信時の初期値をタイマに設定
MOV   DBF0, #LEAD_MAX AND 0FH
PUT   TMM, DBF
MOV   SEQ_H, #0           ;シーケンス・カウンタをリダ受信に設定
MOV   SEQ_L, #0
BR     _INT_EG
;

EJRCT
SUMMARY '%%', ''
    ・リモコン・コード受信
    リモコンを受信し、シーケンス・カウンタの値に従って、各ルーチンの処理を行うために、分岐します。
%%
-----
;
;   リモコン・コード判定
;
REM_IN:
SKNB  SEQ_H, #0           ;コード部分かリダ部分かを判定
SKLT  SEQ_L, #2
BR     CODE_DAT          ;コード部分の受信
SKE   SEQ_L, #0          ;リダ部分のハイ・レベルかロウ・レベルかを判定
BR     CODE_REP         ;リダ部分のロウ・レベルを受信

SUMMARY '%%', ''
    (1) リダ受信処理
    リダ部分の受信判定を行います。

    (a) リダのハイ・レベル部の受信処理
    リダのハイ・レベル部は、通常8msありますが、電池の消耗や反射信号などにより短くなる場合があります。
    従って、本ルーチンでは、受信範囲を8msから9.8msに設定しています。
    本ルーチンでは、最大値の判定は、タイマ割り込みで行います。よって、ここでは、最小値のみの判定を行います。
    もし、最小値よりリダのハイ・レベルが短ければ、受信エラーとなります。
    
```

受信範囲	0.0ms - 9.9ms
------	---------------

範囲内であれば、タイマにリピート時のロウ・レベル最大時間を設定します。

```

.EJ-1
%%
://////////
: リーダ・コード判定
://////////
SKF1 INT_EG ;リーダ部の立ち上がりを検出
BR _INT_EG ;リーダ部の立ち下がり検出に設定

SET1 CMP ;リーダ部の最小判定時間と比較
SUB DBF0,#LEAD_MIN AND 0FH
SUBC DBF1,#(LEAD_MIN SHR 4) AND 0FH
SKF1 CY
BR CODE_ERR
MOV DBF1,#(REP_MAX SHR 4) AND 0FH ;リーダのロウ・レベル時間の検出にタイマ値を設定
MOV DBF0,#REP_MAX AND 0FH
PUT TMM,DBP
SET1 REP_CODE ;タイマ値がリピート時間設定であることを示します
BR SEQ_INC
;
    
```

SUMMARY '%%',''

(b) リーダのロウ・レベル部の受信処理
 リーダのロウ・レベル部は、リピート受信時とデータ受信時とは異なります。
 ここで、まずリピート時の最大時間をタイマに設定し、タイマの割り込みが発生したら、タイマにデータの最大時間を設定するようにします。
 どちらのタイマ値デコード入力があったかを判断するために、REP_CODEフラグを使います。

REP_CODE	タイマ設定状態
0	リピート・コード入力状態
1	データ・コード入力状態

受信状態	受信範囲
リピート受信	1.9ms - 2.8ms
データ受信	3.5ms - 5.4ms

受信範囲外であれば、受信エラーとなります。
 範囲内であれば、ストップ・ビットの最大時間をタイマに設定します。リピート時であれば、カウンタを42Hに設定し、リピート・コードの最大時間をタイマに設定します。

```

.EJ-1
%%
://////////
: リピート/データ判定
://////////
CODE_REP:
SKF1 REP_CODE ;リピート時間かを判定
BR REPEAT ;リピート時間を受信
:*****
: データ判定
:*****
SET1 CMP
SUB DBF0,#(DAT_MIN-REP_MAX) AND 0FH ;データ時のリーダ・ロウ部分の最小判定時間と比較
SUBC DBF1,#((DAT_MIN-REP_MAX) SHR 4) AND 0FH
SKF1 CY
    
```

```

BR      CODE_ERR
MOV     CODE_H,#0           ;データ受信用のデータ・エリアをクリア
MOV     CODE_L,#0
MOV     _CODE_H,#0
MOV     _CODE_L,#0
BR      BIT_SET
;
SUMMARY '%',...
(2) リピート・コードの受信処理
      リピート・コード受信を判定します。
      タイマ設定時間中に、INT端子の断り込みが発生すると、リピート・コードと判断します。
      リピート・コードと判断すると、シーケンス・カウンタを41Hに設定し、次のコード待ちとなります。

```

```

.EJ-1
%%
;*****
;      リピート判定
;*****
REPEAT:
SETI    CMP
SUB     DBF0,#REP_MIN AND 0FH           ;リピート時のリーダー・ロウ部の最小判定時間と比較
SUBC   DBF1,#(REP_MIN SHR 4) AND 0FH
SKF1   CY
BR      CODE_ERR
MOV     SEQ_H,#4           ;シーケンス・カウンタをリピート・コード受信に設定
MOV     SEQ_L,#1
BR      BIT_SET
;

```

```

EJECT
SUMMARY '%',...
(3) データ・コードの受信処理
      NECフォーマットのデータ・コードは、カスタム部、カスタム'部、データ部、反転データ部と4部分に分かれています。
      4部ともそれぞれ8ビットで構成されています。
      0/1の判定は、0.56msのハイ・レベルに続くロウ・レベルの時間で決まります。以下に、0/1判定を行うロウ・レベル
      の受信範囲を示します。

```

ビット	受信範囲
0	0.2 ms - 1.5 ms
1	1.5 ms - 2.4 ms

ハイ・レベル受信範囲	0.2 ms - 1.1 ms
------------	-----------------

範囲外であれば、受信エラーとなります。
 カスタム部、およびカスタム'部を受信後、決められたカスタム・データと比較します。正しければ、続くデータ部を受信します。
 データ部、および反転データ部を受信後、反転データを反転し、データと比較します。同じデータであれば、ストップ・ビットの受信を行います。

```

.EJ-1
%%
;////////////////////////////////////
;      データ・コード受信
;////////////////////////////////////
CODE_DAT:
SKP     SEQ_L,#0001B           ;キャリア部受信か、ロウ・レベル部受信かを判定
BR      BIT_LOW                ;偶数値であれば、キャリア部受信、奇数値であれば、ロウ・レベル部受信
;*****
;      ビット・データのキャリア部受信
;*****
SETI    CMP                     ;受信範囲チェック

```

```

SUB    DBF0,#BIT_MIN AND 0FH
SUBC   DBF1,#(BIT_MIN SHR 4) AND 0FH
SKP1   CV
BR     CODE_ERR
;-----
;      ストップ・ビット判定
;-----
SKNR   SEQ_H,#4           ;ストップ・ビットを受信中かをチェック
SKE    SEQ_L,#2
BR     BIT0_SET
MOV    DBF1,#(LOW_TIME SHR 4) AND 0FH ;ストップ・ビットであれば、タイマにロウ・レベル時間を設定
MOV    DBF0,#LOW_TIME AND 0FH
PUT    TMM,DBF
BR     SEQ_INC
;
;-----
;      ビット0の判定用時間設定
;-----
BIT0_SET:
MOV    DBF1,#(BIT0_MAX SHR 4) AND 0FH ;ビット0判定用の最大時間をタイマに設定
MOV    DBF0,#BIT0_MAX AND 0FH
PUT    TMM,DBF
CLR1   BIT_F
BR     SEQ_INC
;
;-----
;      ビット・データ判定
;-----
BIT_LOW:
;-----
;      ビット・オーバー判定
;-----
SKNB   SEQ_H,#4           ;ビット数をチェック
SKE    SEQ_L,#3
BR     BIT_CHK
SUMMARY '%', ''
(4) エラー処理
リモコン受信中に受信範囲以外のコードや範囲外のビット数を受信したときに、エラー処理を行います。
エラーと判断すると、それ以降のコードは無視され、一定時間のロウ・レベル区間のみの検出処理を行います。
一定時間のロウ・レベル区間が発見されると、次のリーダー・コード待ちとなります。
%%
;-----
;      コード受信エラー
;-----
CODE_ERR:
MOV    SEQ_H,#0EH         ;シーケンス・カウンタをエラー時に設定
MOV    DBF1,#(LOW_TIME SHR 4) AND 0FH ;エラー時間終了判定時間にタイマ値を設定
MOV    DBF0,#LOW_TIME AND 0FH
PUT    TMM,DBF
BR     _INT_EG
;
EJECT
SUMMARY '%', ''
(5) ビット・データの設定処理
データ部の受信が終了すると、ビット・データをメモリ(CODE_H, CODE_L, _CODE_H, _CODE_L)に設定し、18ビット設定が完了
するとデータとデータの反転データとを比較し、一致するとデータの設定処理を行います。
%%
;-----
;      ビットの値判定
;-----
BIT_CHK:
SKP1   BIT_F             ;ビットが「0」か「1」かをチェック

```

```

BR      BIT_OK

;*****
; ビット 0 の最小値判定
;*****
SET1    CMP                      ;ビット 0 時の受信最小範囲をチェック
SUB     DBF0,#BIT0_MIN AND 0FH
SUBC    DBF1,#(BIT0_MIN SHR 4) AND 0FH
SKF1    CY
BR      CODE_ERR

EJECT
-----
; ビット・データ設定
;-----
BIT_OK:
CLR1    CY                      ;設定ビット・データをCYフラグに設定
SKF1    BIT_F                   ;設定ビット=1の時 CY=1
SET1    CY                      ;設定ビット=0の時 CY=0
RORC    _CODE_H                 ;_CODE_Hの最上位ビットよりビット・データを設定
RORC    _CODE_L                 ;これにより受信データの最後のビットが_CODE_Hの最上位ビットとなる
RORC    CODE_H
RORC    CODE_L

;-----
; 16ビット受信判定
;-----
SKE     SEQ_L,#1                ;16ビット分受信したかをチェック
BR      BIT_SET

;*****
; 受信コード判定
;*****
SKNR    SEQ_H,#4                ;データ部の受信かカスタム・コード部の受信かをチェック
BR      BIT_DATA
SKE     SEQ_H,#2
BR      BIT_SET

;*****
; カスタム・コード設定
;*****
ST       CUSTOM1,_CODE_H        ;カスタム・コードをCUSTOM1,CUSTOM2,CUSTOM3,CUSTOM4に設定
ST       CUSTOM2,_CODE_L
ST       CUSTOM3,_CODE_H
ST       CUSTOM4,_CODE_L
BR      BIT_SET
;

BIT_DATA:
;*****
; データ・コード設定
;*****
XOR     _CODE_H,#1111B          ;コードの反転データを反転
XOR     _CODE_L,#1111B
SET2    CMP,Z                   ;受信コードのチェック
SUB     CODE_L,_CODE_L
SUBC    CODE_H,_CODE_H
SKT1    Z
BR      CODE_ERR

ST       DATA1,_CODE_H        ;データ・コードをDATA1,DATA2に設定
ST       DATA2,_CODE_L

SUMMARY "%",...
(8) ビット・データのキャリア部の時間設定
      ビット・データおよびストップ・ビットのキャリア部の最大時間をタイマに設定します。
.EJ-1
%
```

```

////////////////////////////////////
:      コードのキャリア部の時間設定
////////////////////////////////////
BIT_SET:
MOV    DBF1,#(BIT_MAX SHR 4) AND 0FH ;ビット受信時の最大判定時間タイマに設定
MOV    DBF0,#BIT_MAX AND 0FH
PUT    TMM,DBF

EJECT
SUMMARY '%',''
・割り込み時の復旧処理
各割り込み処理が終了するごとに、カウンタの値をインクリメントし、割り込みエッジの反転を行います。
最後に、退避したシステム・レジスタの内容を復旧し、割り込みポイントに復帰します。
.EJ-1
%%
-----
:      受信カウンタの更新
-----
SEQ_INC:
ADD    SEQ_L,#1 ;シーケンス・カウンタをインクリメント
ADDC   SEQ_H,#0

-----
:      リモコンの割り込みエッジの反転
-----
INT_EG:
NOTI   INT_EG ;受信時の割り込み発生エッジを反転

-----
:      割り込み時のレジスタの復旧
-----
RETURN:
PEEK   WR,D_RPL ;退避レジスタ値を復旧
POKE   RPL,WR
PEEK   WR,D_WR

-----
:      割り込みルーチン終了
-----
EI_RET:
EI     ;割り込みを許可し、割り込み処理を終了
RETI

EJECT
SUMMARY '%',''
2. タイマ割り込みルーチン
タイマ割り込みルーチンは、リモコン受信時には、エラー発生時の割り込み処理として使用されます。
主に、リモコン波形の受信最大範囲を超えたときに処理を行います。
そのほかには、データ・コード受信とビット・コード受信の判定、ビット0とビット1データの判定を行う処理も含まれて
います。
%%
-----
:      タイマの割り込みルーチン
-----
TIMER_INT:
-----
:      割り込み時のレジスタの退避
-----
POKE   D_WR,WR ;割り込み時のレジスタ値退避
PEEK   WR,RPL
POKE   D_RPL,WR
MOV    PSW,#0
MOV    RPL,#0010B

SUMMARY '%',''
(1) エラー処理
リモコン受信時にエラーとなったときの、一定時間(3ms)ロウ・レベルになったかを判定するために処理です。

```

```

%%
:-----
:      エラー処理?
:-----
SKE      SEQ_H,#0EH      ;エラー時のシーケンス・カウンタ値を判定
BR       REM_OK

://////////
:      エラー終了判定
://////////
SKT1     INT_EG          ;ロウ・レベル時の割り込みかを判定
BR       RETURN         ;ハイ・レベルであればエラー未終了

SEQ_0F:  MOV      SEQ_H,#0FH      ;エラーを終了し、次のコード待ち状態にシーケンス・カウンタを設定
        MOV      SEQ_L,#0

:*****
:      受信終了用基本時間設定
:*****
MOV      DBF1,#(LOW_BASE SHR 4) AND 0FH ;受信終了判定時間の基本時間をタイマに設定
MOV      DBF0,#LOW_BASE AND 0FH
PUT      TMM,DBF
BR       RETURN
:

EJECT
SUMMARY '%',''
(2) リモコン受信の終了
    リモコン受信が終了したかを判定するための処理です。
    受信が終了するとデータ・コードならCODE_INフラグがセットされ、リピート・コードならREP_INフラグがセットされます。
%%
.EJ-1
%%
:-----
:      受信OK?
:-----
REM_OK:  SKNR     SEQ_H,#4      ;受信終了判定
        SKE     SEQ_L,#3
        BR     NEXT_IN

://////////
:      受信OK
://////////
SET1     REP_IN          ;リピート・コード受信チェック
SKP1     REP_CODE       ;リピート・コード受信状態
BR       SEQ_0F         ;データ・コード受信状態
CLR1     REP_IN
SET1     CODE_IN
BR       SEQ_0F
:

SUMMARY '%',''
(3) リモコン受信の終了判定
    データ・コードまたはリピート・コードの受信後、150ms間リモコンの受信が確認できない場合には、受信終了と判定し
    タイマ動作を停止します。
%%
:-----
:      終了時間?
:-----
NEXT_IN: SKE      SEQ_H,#0FH      ;受信終了判定中か?
        BR     CODE_CHK      ;コード受信済

://////////
:      受信終了時間の測定
://////////
ADD      SEQ_L,#1        ;時間のカウント

```



```

SKGB SEQ_L,#0AH
BR RETURN
;//////////
; 受信終了
;//////////
SETJ INT_EG ;タイマおよび割り込みエッジを初期化
CLR1 TMEN ;タイマをストップ
SETJ TMRES ;タイマを初期化
MOV SEQ_H,#0 ;シーケンス・カウンタをリーダ受信状態に設定
MOV SEQ_L,#0
MOV DBF1,#(LEAD_MAX SHR 4) AND 0FH ;タイマにリーダ受信時の最大判定時間を設定
MOV DBF0,#LEAD_MAX AND 0FH
PUT TMM,DBF
CLR1 REP_IN ;リピート・コード受信状態をクリア
BR RETURN
;

```

EJECT SUMMARY '%',''

(4) コード受信時の処理
コード受信中に、タイマの割り込みが発生するには、以下の状態があります。

1. リーダ・コード受信時に、リーダ・コードの受信最大時間を越えた場合
2. データ・コード/リピート・コード判断時に、リピート・コード判定時間の最大時間を越えたとき、およびデータ・コード判断時にデータ・コード判定時間の最大時間を越えた場合
3. ビットのキャリア部の時間測定時に、受信最大時間を越えた場合
4. ビットのロウ・レベル時間測定時に、ビット0の最大判断時間を越えたとき、ビット1の最大判断時間を越えた場合

%%

; コード受信中

CODE_CHK:

```

SKNE SEQ_H,#0 ;ビット・データ受信中のチェック
SKLT SEQ_L,#2
BR BIT_OVR
SKNE SEQ_L,#0 ;リーダ・コード受信中であればエラー
BR CODE_ERR

```

//////////

; データ・コード受信判定

//////////

```

SKT1 REP_CODE ;リピート・コード受信中チェック
BR CODE_ERR
CLR1 REP_CODE ;リピート・コード受信中であれば、データ・コード時の最大時間をタイマに設定
MOV DBF1,#((DAT_MAX-REP_MAX) SHR 4) AND 0FH
MOV DBF0,#(DAT_MAX-REP_MAX) AND 0FH
PUT TMM,DBF
BR RETURN
;

```

//////////

; ビット・データ受信判定

//////////

BIT_OVR:

```

SKP SEQ_L,#0001B ;キャリア部の受信中チェック
SKP1 BIT_F ;ビット1データ受信中のチェック
BR CODE_ERR
SET1 BIT_F ;ビット1の最大時間をタイマに設定
MOV DBF1,#((BIT1_MAX-BIT0_MAX) SHR 4) AND 0FH
MOV DBF0,#(BIT1_MAX-BIT0_MAX) AND 0FH
PUT TMM,DBF
BR RETURN
;

```

END



クロスレファレンス



AS17K Z1.10 D1 << E17201A XREF LIST >>

12:00:05 02/28/91 PAGE 02-001

PROG =

SOURCE = REM.ASM

SYMBOL	TYPE	A	VALUE	/REF(#DEF)
_CODE_H	MEM	L	0.14 /# 85	, 334 , 481 , 504 , 513 , 517
_CODE_L	MEM	L	0.15 /# 86	, 335 , 482 , 505 , 514 , 516
_INT_EG	LAB	L	83 / 214	, 208 , 276 , 449 , # 554
BITO_MAX	DAT	L	13 /# 116	, 418 , 419 , 718 , 719
BITO_MIN	DAT	L	3 /# 117	, 469 , 470
BITO_SET	LAB	L	4F / 407	, # 417
BITI_MAX	DAT	L	25 /# 114	, 718 , 719
BIT_CHK	LAB	L	5C / 434	, # 461
BIT_DATA	LAB	L	75 / 496	, # 509
BIT_F	FLG	L	0.20.1 /# 90	, 421 , 421-1 , 421-1 , 462 , 462-1 , 462-1 , 479 , 479-1
			479-1	, 716 , 715-1 , 715-1 , 717 , 717-1 , 717-1
BIT_LOW	LAB	L	54 / 391	, # 428
BIT_MAX	DAT	L	E /# 111	, 533 , 534
BIT_MIN	DAT	L	3 /# 112	, 397 , 398
BIT_OK	LAB	L	63 / 483	, # 477
BIT_OVR	LAB	L	C0 / 895	, # 713
BIT_SET	LAD	L	7E / 336	, 356 , 490 , 498 , 506 , # 532
CODE_CHK	LAD	L	B4 / 855	, # 692
CODE_DAT	LAB	L	41 / 251	, # 389
CODE_ERR	LAB	L	57 / 231	, 282 , 331 , 353 , 400 , # 444 , 472 , 519 , 697
			702	, 716
CODE_H	MEM	L	0.12 /# 83	, 332 , 483 , 502 , 517 , 521
CODE_IN	FLG	P	0.20.2 / 48	, # 91 , 641 , 641-1 , 641-1
CODE_L	MEM	L	0.13 /# 84	, 333 , 484 , 503 , 516 , 522
CODE_REP	LAB	L	2D / 253	, # 321
CUSTOM1	MEM	P	0.00 / 54	, # 73 , 502
CUSTOM2	MEM	P	0.01 / 54	, # 74 , 503
CUSTOM3	MEM	P	0.02 / 54	, # 75 , 504
CUSTOM4	MEM	P	0.03 / 54	, # 76 , 505
D_RPL	MEM	L	0.61 /# 97	, 141 , 561 , 588
D_WR	MEM	L	0.60 /# 96	, 139 , 563 , 586
DAT_MAX	DAT	L	54 /# 108	, 704 , 705
DAT_MIN	DAT	L	37 /# 109	, 328 , 329
DATA1	MEM	P	0.04 / 54	, # 77 , 521
DATA2	MEM	P	0.05 / 54	, # 78 , 522
EI_RET	LAB	P	87 / 44	, # 568
INT_EG	FLG	P	0.73.1 / 50	, # 94 , 275 , 275-1 , 275-1 , 565 , 565-1 , 565-1 , 606
			605-1	, 605-1 , 666 , 666-1 , 666-1
LEAD_MAX	DAT	L	9A /# 102	, 233 , 234 , 671 , 672
LEAD_MIN	DAT	L	64 /# 103	, 279 , 280
LOW_BASE	DAT	L	EA /# 120	, 813 , 814
LOW_TIME	DAT	L	2E /# 119	, 408 , 409 , 446 , 447
NEXT_IN	LAB	L	A1 / 532	, # 653
REM_FLG	MEM	L	0.20 /# 88	, 89 , 90 , 91 , 92
REM_IN	LAB	L	1C / 227	, # 248
REM_OK	LAB	L	98 / 600	, # 629
REMCON_INT	LAB	P	5 / 42	, # 125
REP_CODE	FLG	P	0.20.0 / 47	, # 89 , 286 , 286-1 , 286-1 , 322 , 322-1 , 322-1 , 638
			638-1	, 701 , 701-1 , 701-1 , 703 , 703-1 , 703-1
REP_IN	FLG	P	0.20.3 / 49	, # 92 , 228 , 228-1 , 228-1 , 637 , 637-1 , 637-1 , 640
			640-1	, 640-1 , 674 , 674-1 , 674-1 , 704 , 705
REP_MAX	DAT	L	2A /# 105	, 283 , 284 , 328 , 329 , 704 , 705
REP_MIN	DAT	L	1B /# 106	, 350 , 351
REPEAT	LAB	L	39 / 323	, # 348

AS17K Z1.10 D1 << E17201A XREP LIST >> 12:00:05 02/26/91 PAGE 02-002

PROG -

SOURCE - REM.ASM

SYMBOL TYPE A VALUE /REF (#DEF)

RESTART	LAB L	16 /	229	,#	232														
RETURN	LAB L	84 /#	560	,	606	,	616	,	662	,	675	,	707	,	721				
SEQ_OF	LAB L	92 /#	607	,	539	,	642												
SEQ_H	MEM P	0.10 /	53	,#	80	,	213	,	226	,	236	,	249	,	354	,	405	,	432
			445	,	485	,	497	,	549	,	598	,	608	,	630	,	654	,	688
			693																
SEQ_INC	LAB L	81 /	287	,	411	,	422	,#	547										
SEQ_L	MEM P	0.11 /	53	,#	81	,	230	,	237	,	250	,	252	,	355	,	390	,	406
			433	,	489	,	548	,	609	,	631	,	660	,	661	,	670	,	694
			696	,	714														
TIMER_INT	LAB P	89 /	43	,#	582														

TOTAL SYMBOLS - 66

END OF XREP LIST

5.2 送 信

μPD17201Aは、リモコン・キャリア発生回路を内蔵しており、ハイ・レベル期間とロウ・レベル期間を、DBFを介してNRZHTMMおよびNRZLTMMに設定し、キャリア・デューティ比およびキャリア周波数を決定します。

キャリアを出力するREM端子の出力制御は、レジスタ・ファイル上のNRZ、NRZBFおよび8ビット・タイマで行います。

NRZの内容が"1"の期間はリモコン・キャリア発生回路で生成したキャリア信号をREM端子に出力し、NRZの内容が"0"の期間は、REM端子の出力は、ロウ・レベルになります。

また、8ビット・タイマによる割り込みが発生すると、自動的にNRZBFの内容がNRZに転送されます。

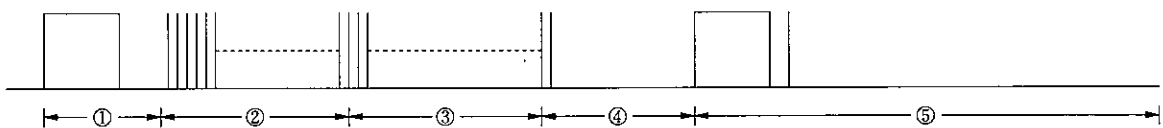
よって、割り込みが発生する前にあらかじめNRZBFにデータを設定しておくと、8ビット・タイマのカウント動作に同期して、REM端子の状態が変化します。

5.2.1 プログラム説明

ここで紹介するプログラムでは、送信波形用のシーケンス・カウンタ (S_CNT) とビット・カウンタ (B_CNTH, B_CNTL) を使用しています。これらのカウンタを用いることにより現在何を送信しているかが、明確になります。

各波形に対応するシーケンス・カウンタを図5-3に示します。

図5-3 波形とシーケンス・カウンタの関係



- | | |
|-------------------------------|----------|
| ① リード・コード | S_CNT= 1 |
| ② カスタム・コード8ビットおよびカスタム・コード8ビット | S_CNT= 2 |
| ③ データ・コード8ビットおよびデータ・コード8ビット | S_CNT= 3 |
| ④ ストップ・ビットおよびフレーム・スペース | S_CNT= 4 |
| ⑤ リピート・コード | S_CNT= 5 |

シーケンス・カウンタおよびビット・カウンタ、NRZBFの関係を表5-1に示します。

表5-1 シーケンス・カウンタおよびビット・カウンタ, NRZBFの関係

波 形	①			②/③					④				
	設定時間 (ms)	-	3	4.5	0.56	0.56/1.69		*	0.56	0.56/1.69		0.56	6.7×5
シーケンス・カウンタ	1			2/3					4				
ビット・カウンタ	1	2	3	1	2		1F	20		1	2-6	7
NRZBF	1	0	1	0	1		0	1		0	0	1

波 形	⑤					
	設定時間 (ms)	6	3	2.25	0.56	6.7×14
シーケンス・カウンタ	5					
ビット・カウンタ	1	2	3	4	5-12	13
NRZBF	1	0	1	0	0	1

* : ビット・カウンタが奇数の場合はハイの時間 (0.56 ms), 偶数の場合はロウの時間 (データ'0'の場合0.56 ms, データ'1'の場合1.69 ms) となります。

例) カスタム・コードの場合

カスタム・コード : 04H

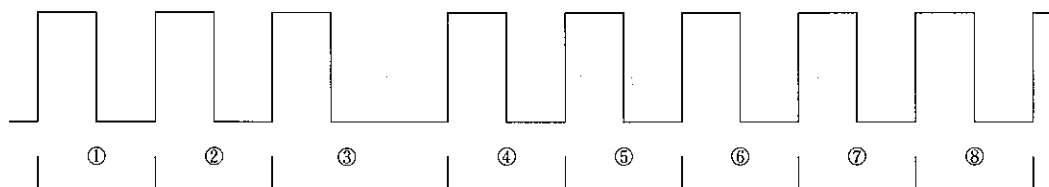


表5-2 ビット・データとビット・カウンタおよびNRZBFの関係

波 形	①		②		③		④		⑤		⑥	
	ビット・データ	0		0		1		0		0		0
設定時間 (ms)	0.56	0.56	0.56	0.56	0.56	1.69	0.56	0.56	0.56	0.56	0.56	0.56
シーケンス・カウンタ	2											
ビット・カウンタ (HEX)	1	2	3	4	5	6	7	8	8	A	B	C
NRZBF	0	1	0	1	0	1	0	1	0	1	0	1

波 形	⑦		⑧	
	ビット・データ	0		0
設定時間 (ms)	0.56	0.56	0.56	0.56
シーケンス・カウンタ	2			
ビット・カウンタ (HEX)	D	E	F	10
NRZBF	0	1	0	1

〈使用するハードウェア〉

- ・タイマ：8ビット・タイマ
- ・REM端子（キャリア周波数=37.74 kHz デューティ比=1/3 ハイ・レベル期間=17, ロウ・レベル期間=36)
- ・LED出力端子

〈初期設定〉

フラグ：KCHK_OK（フレーム・スペース送信中用のフラグ）をクリアします。

カウンタ：S_CNT=1, B_CNTL=0, B_CNTH=0

8ビット・タイマ：リーダー・コードのハイの時間9msのうち6msを設定します。

割り込み：8ビット・タイマ割り込みを許可します。

〈起動方法〉

最初にキーの入力状態をチェックし、有効キーが押されている場合、送信ルーチン（SOUSIN）をコールしてください。どのキーも押されていない場合、スタンバイ状態にします。



ドキュメント



AS17K Z1.10 D1 << E17201A DOCUMENT >> 15:20:13 02/19/91

TABLE OF CONTENTS

	PAGE
第5章 リモコン応用 (送信)	1
5.1 プログラム使用説明	1
5.2 シーケンサから送信機へのデータ送信	2
5.3 シーケンサから送信機へのデータ送信 (カウンタのチェック)	3
5.4 シーケンサから送信機へのデータ送信 (ループ)	4

第 5 章 リモコン応用 (送信)

```

PUBLIC (LAB) : EI_RET11, SOUSIN, TIME_IR
EXTRN (MRM) : S_DAT1, S_DAT2, S_DAT3, S_DAT4, S_DAT5, S_DAT6, S_DAT7, S_DAT8
EXTRN (FLG) : K_CHANGE, K_ERROR, K_EXIST, OPFC_F, ONC_F
EXTRN (LAB) : CODESET, KEYSKAN
ADDR RANGE : 00B7H - 0194H
    
```

<プログラム説明>

リモコン・キャリア発生回路は、6ビット・カウンタ、ハイ・レベル期間設定用モジュロ・レジスタ(NRZ_LTM)、ロウ・レベル期間設定用モジュロ・レジスタとコンパレータで構成されており、ハイ・レベル期間とロウ・レベル期間をそれぞれに対応するモジュロ・レジスタに設定することで、キャリア・デューティおよびキャリア周波数をキャリアを出力するREM端子の出力制御は、レジスタ・ファイル上のNRZ、NRZBFおよび8ビット・タイマで行います。このプログラム例では、各キーに対応するの送信コードをテーブル・データより呼び出し、キャリア周波数: 37.74kHz、デューティ比: 1/3、ハイ・レベル期間: 17、ロウ・レベル期間: 36のリモコン信号の送信を行います。

```

ENTRANCES      :-
MEMORIES CHANGED :-
MEMORIES REFERRED :-
MEMORIES MANIPULATED :-
FLAGS CHANGED  :-
FLAGS REFERRED :-
DATA REFERRED  :-
BRANCH TO      :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL    :-
    
```

<RAM使用説明>

```

WORK1 ..... 0. 0 0 H ..... 汎用レジスタです。
WORK2 ..... 0. 0 1 H ..... 汎用レジスタです。
WORK3 ..... 0. 0 2 H ..... 汎用レジスタです。
WORK4 ..... 0. 0 3 H ..... 汎用レジスタです。

S_DAT1 ..... 0. 3 0 H ..... 送信データ格納用のエリアです。
S_DAT2 ..... 0. 3 1 H ..... 送信データ格納用のエリアです。
S_DAT3 ..... 0. 3 2 H ..... 送信データ格納用のエリアです。
S_DAT4 ..... 0. 3 3 H ..... 送信データ格納用のエリアです。
S_DAT5 ..... 0. 3 4 H ..... 送信データ格納用のエリアです。
S_DAT6 ..... 0. 3 5 H ..... 送信データ格納用のエリアです。
S_DAT7 ..... 0. 3 6 H ..... 送信データ格納用のエリアです。
S_DAT8 ..... 0. 3 7 H ..... 送信データ格納用のエリアです。

WR_1 ..... 0. 3 A H ..... WR用のスタック・メモリです。
PSW_1 ..... 0. 3 B H ..... PSW用のスタック・メモリです。
RPL_1 ..... 0. 3 C H ..... RPL用のスタック・メモリです。

S_CNT ..... 0. 2 2 H ..... 送信用のシーケンス・カウンタです。
B_CNTH ..... 0. 2 3 H ..... 送信波形の時間データ設定用のカウンタです。
B_CNTH ..... 0. 2 4 H ..... 送信波形の時間データ設定用のカウンタです。

CHATARING ..... 0. 3 8 H ..... チャタリングのフラグ用のエリアです。
RCHK_OK ..... # 3 ..... 送信中にキーのチェックが可能な場合セットされます。
    
```

```

ENTRANCES          :-
MEMORIES CHANGED  :-
MEMORIES REFERRED :-
MEMORIES MANIPULATED :-
FLAGS CHANGED     :-
FLAGS REFERRED    :-
DATA REFERRED     :-
BRANCH TO         :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL       :-
    
```

<送信コードの設定>

各キーに対応する送信コード・データをテーブル・データより呼び出し、メモリ S_DAT1 - S_DAT8 に格納します。
 S_DAT1 - S_DAT4 カスタム・コード

S_DAT6 - S_DAT8 データ・コード

```

ENTRANCES          :TIMR_IR
MEMORIES CHANGED   :B_CNTH          B_CNTH
                   :D_CNTH          RFL_I
                   :FSM_I           WR_I
                   :S_CNT
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :KCHK_OK
FLAGS REFERRED     :K_CHANGE        K_ERROR
                   :K_EXIST        KCHK_OK
                   :OPPC_P         ONC_F
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :CODESET         KEYSKAN
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

<送信用シーケンス・カウンタのチェック>

送信用シーケンス・カウンタ S_CNT の値と役割

- S_CNT = 1 の場合 リーダ・コードの時間データを設定します。
- S_CNT = 2 の場合 カスタム・コードの時間データを設定します。
- S_CNT = 3 の場合 データ・コードの時間データを設定します。
- S_CNT = 4 の場合 ストップ・ビットおよびフレーム・スペースの時間データを設定します。
- S_CNT = 5 の場合 リピート・リーダー・コードの時間データを設定します。

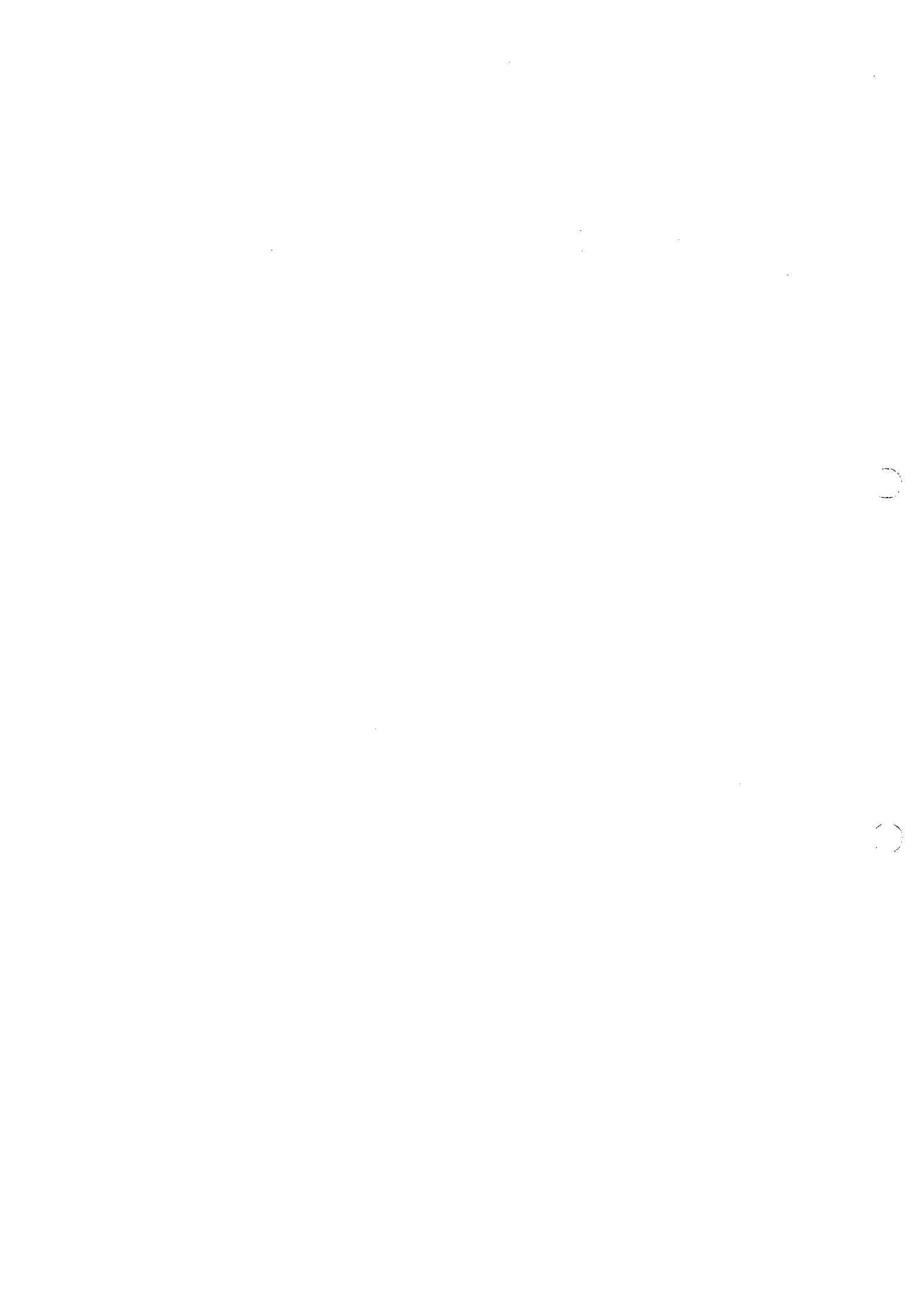
```

ENTRANCES          :EI_RET11
MEMORIES CHANGED   :B_CNTH          B_CNTH
                   :S_CNT          WORK1
                   :WORK2          WORK3
                   :WORK4
MEMORIES REFERRED   :PSW_1          RPL_1
                   :S_DAT1         S_DAT2
                   :S_DAT3         S_DAT4
                   :S_DAT5         S_DAT6
                   :S_DAT7         S_DAT8
                   :WORK1          WORK2
                   :WORK3          WORK4
                   :WR_1
MEMORIES MANIPULATED :-
FLAGS CHANGED      :KCHK_OK
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```


<キー・データから送信データへの変換ルーチン>

各キーに対する送信コード・データをテーブルから呼び出し、4ビットずつ S_DAT1 - S_DAT4 に格納します。

```
PUBLIC (MEM) : S_DAT1, S_DAT2, S_DAT3, S_DAT4, S_DAT5, S_DAT6, S_DAT7, S_DAT8
PUBLIC (LAB) : CODESET
EXTRN (MEM) : KEYDATA_H, KEYDATA_L
ADDR RANGE : 0195H - 01A8H
```



RAMマップ

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 15:19:55 02/19/91 PAGE 03-001

PROG -

SOURCE - SOUSIN.ASM

(PUBLIC/LOCAL MAP)

BANK - 0

		ROW								
		7	6	5	4	3	2	1	0	
			LCDD32	LCDD16	LCDD0					0
									WORK1	
			LCDD33	LCDD17	LCDD1					1
									WORK2	
			LCDD34	LCDD18	LCDD2					2
							S_CNT		WORK3	C
			LCDD35	LCDD19	LCDD3					3
							B_CNTL		WORK4	
IAR3				LCDD20	LCDD4					4
							B_CNTH			
IAR2				LCDD21	LCDD5					5
IAR1				LCDD22	LCDD6					6
IAR0				LCDD23	LCDD7					7
IWR				LCDD24	LCDD8					8
							CHATARING			
IBANK				LCDD25	LCDD9					9
			IWR_I							
IIXH				LCDD26	LCDD10					A
IMPH										
			IPSW_I							
IIXM				LCDD27	LCDD11					B
IMPL										
			IRPL_I							
IIXL				LCDD28	LCDD12				IDBF3	C

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 15:19:55 02/19/91 PAGE 03-002

PROG -

SOURCE - SOUSIN.ASM

(PUBLIC/LOCAL MAP)

BANK - 0

ROW

7	6	5	4	3	2	1	0	
IRPH		LCDD29	LCDD13				DBF2	D
IRPL		LCDD30	LCDD14				DBF1	E
IPSW		LCDD31	LCDD15				DBF0	F

AS17K Z1.10 D1 << B17201A FLG MAP >> 15:19:55 02/19/91 PAGE 03-003

PROG =

SOURCE = SOUSIN.ASM

(FLAG MAP) BANK = 0

MSB	3	2	1	0	LSB
38					
	IKCHK_OK				
70	IPOA3	IPOA2	IPOA1	IPOA0	
71	IPOB3	IPOB2	IPOB1	IPOB0	
72	IPOC3	IPOC2	IPOC1	IPOC0	
73	IPOD3	IPOD2	IPOD1	IPOD0	
7A	IMPE				
7E				IBCD	
7F	ICMF	ICY	IZ	IXE	
82			ISYCK	IXEN	
83	IWTRES	IWTMD	IWTRES		
87	IVDDDET1	IVDDDET0			
8F				INT	
91				INRZBP	
92				INRZ	
A0				ADCCMP	
A1	IVPFEN	IADCEN	IADCCH1	IADCCH0	
A2	ISLOTS	ISIOHIZ	ISIOCK1	ISIOCK0	

AS17K 21.10 D1 << E17201A FLAG MAP >> 15:19:55 02/19/91 PAGE 03-004

PROG -

SOURCE - SOUSIN.ASM

(FLAG MAP) BANK = 0

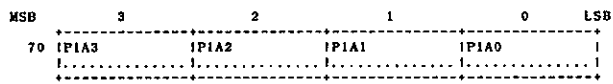
MSB	3	2	1	0	LSB
A3		NRZEN	TMOE	SIOEN	
A7	PODBIO3	PODBIO2	PODBIO1	PODBIO0	
AF	IPSIO	PWTM	IP	PTM	
B1	LCDEN	LCDCK2	LCDCK1	LCDCK0	
B2	LCDMD3	LCDMD2	LCDMD1	LCDMD0	
B3	TMEN	TMRES	TMCK1	TMCK0	
B7	PIAGIO	PCGGIO	POBGIO	POAGIO	
DB	IRQSIO				
BC		IRQTM			
BD			IRQ		
DE				IRQTM	

AS17K Z1.10 D1 << B17201A FLG MAP >> 15:19:55 02/18/91 PAGE 03-005

PROG =

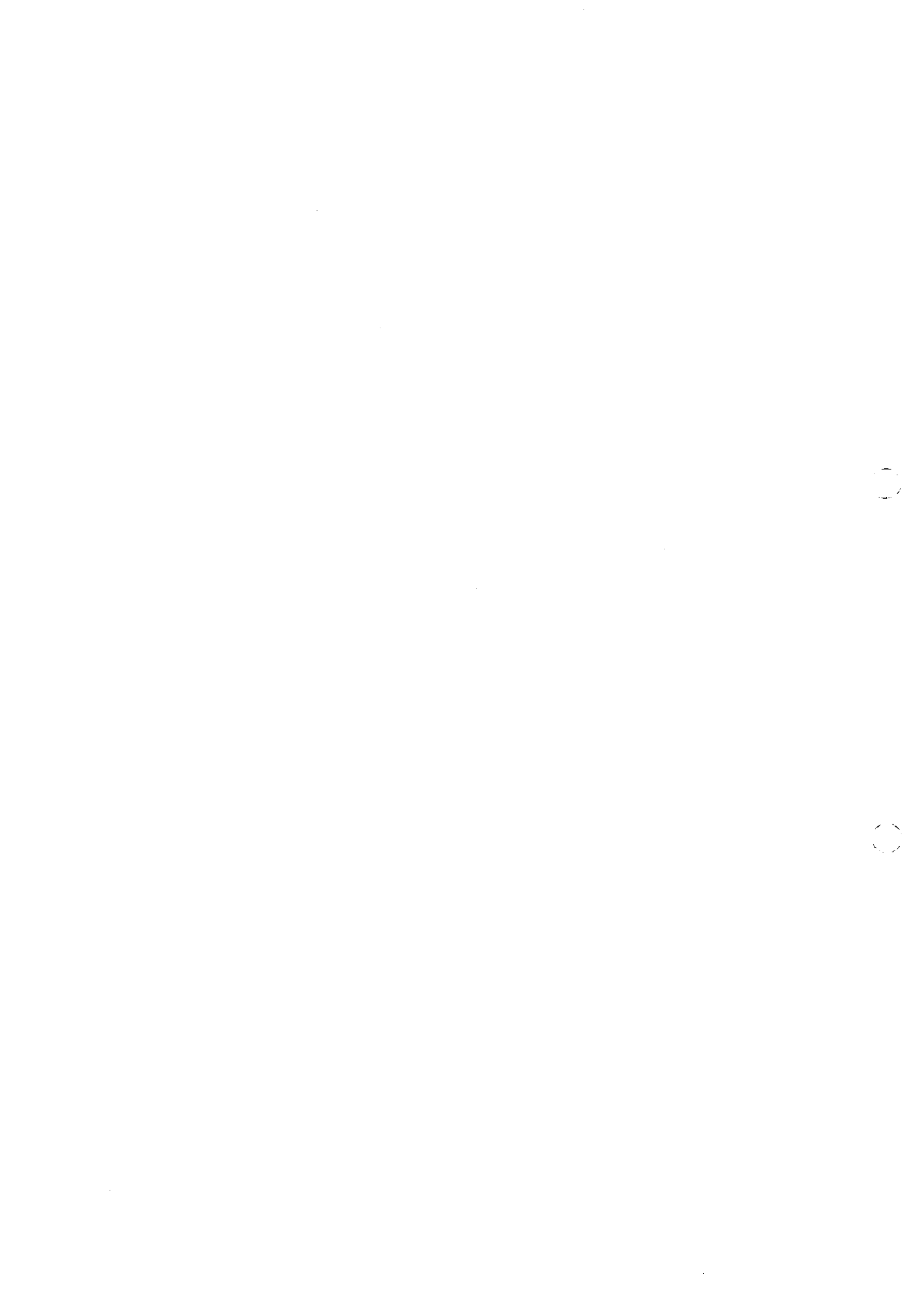
SOURCE = SOUSIN.ASM

(FLAG MAP) BANK = 1





RAM使用説明



AS17K Z1.10 D1 << B17201A DATA MEMORY MAP >> 15:19:55 02/19/91 PAGE 03-008

PROG -

SOURCE - SOUSIN.ASK

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
ADCC0	FLG	0	A1	...1	
ADCC1	FLG	0	A1	...1	
ADCCMP	FLG	0	A0	...1	
ADCCN	FLG	0	A1	...1	
AR0	MEM	0	77		
AR1	MEM	0	76		
AR2	MEM	0	75		
AR3	MEM	0	74		
B_CNTH	MEM	0	24		
B_CNTHL	MEM	0	23		
BANK	MEM	0	79		
BCD	FLG	0	7E	...1	
CHATARING	FLG	0	38		
CMF	FLG	0	7F	1...	
CY	FLG	0	7F	...1	
DBF0	MEM	0	F		
DBF1	MEM	0	E		
DBF2	MEM	0	D		
DBF3	MEM	0	C		
INT	FLG	0	8F	...1	
IP	FLG	0	AF	...1	
IPSI0	FLG	0	AF	1...	
IPTM	FLG	0	AF	...1	
IPWTM	FLG	0	AF	...1	
IRQ	FLG	0	BD	...1	
IRQS10	FLG	0	BB	1...	
IRQTM	FLG	0	BE	...1	
IRQWTM	FLG	0	BC	...1	
IXE	FLG	0	7F	...1	
IXH	MEM	0	7A		
IXL	MEM	0	7C		
IXM	MEM	0	7B		
KCHK_OK	FLG	0	38	1...	送信中にキーのチェックが可能な場合セットされます。
LCCK0	FLG	0	B1	...1	
LCCK1	FLG	0	B1	...1	
LCCK2	FLG	0	B1	...1	
LCDD0	MEM	0	40		
LCDD1	MEM	0	41		
LCDD10	MEM	0	4A		
LCDD11	MEM	0	4B		
LCDD12	MEM	0	4C		
LCDD13	MEM	0	4D		
LCDD14	MEM	0	4E		
LCDD15	MEM	0	4F		
LCDD16	MEM	0	50		
LCDD17	MEM	0	51		
LCDD18	MEM	0	52		
LCDD19	MEM	0	53		
LCDD2	MEM	0	42		
LCDD20	MEM	0	54		
LCDD21	MEM	0	55		
LCDD22	MEM	0	56		

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 15:19:55 02/19/91 PAGE 03-007

PROG -

SOURCE - SOUSIN.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCDD23	MEM	0	57		
LCDD24	MEM	0	58		
LCDD25	MEM	0	59		
LCDD26	MEM	0	5A		
LCDD27	MEM	0	5B		
LCDD28	MEM	0	5C		
LCDD29	MEM	0	5D		
LCDD9	MEM	0	43		
LCDD30	MEM	0	5E		
LCDD31	MEM	0	5F		
LCDD32	MEM	0	60		
LCDD33	MEM	0	61		
LCDD34	MEM	0	62		
LCDD35	MEM	0	63		
LCDD4	MEM	0	44		
LCDD5	MEM	0	45		
LCDD6	MEM	0	46		
LCDD7	MEM	0	47		
LCDD8	MEM	0	48		
LCDD9	MEM	0	49		
LCDEN	FLG	0	B1	1...	
LCDM0	FLG	0	B2	...1	
LCDM1	FLG	0	B2	...1	
LCDM2	FLG	0	B2	...1	
LCDM3	FLG	0	B2	1...	
WPE	FLG	0	7A	1...	
WPH	MEM	0	7A		
MPL	MEM	0	7B		
NRZ	FLG	0	92	...1	
NRZBP	FLG	0	91	...1	
NRZEN	FLG	0	A3	1...	
POA0	FLG	0	70	...1	
POA1	FLG	0	70	...1	
POA2	FLG	0	70	...1	
POA3	FLG	0	70	1...	
POAGIO	FLG	0	B7	...1	
POB0	FLG	0	71	...1	
POB1	FLG	0	71	...1	
POB2	FLG	0	71	1...	
POB3	FLG	0	71	1...	
POBGIO	FLG	0	B7	...1	
POC0	FLG	0	72	...1	
POC1	FLG	0	72	...1	
POC2	FLG	0	72	1...	
POC3	FLG	0	72	1...	
POCCIO	FLG	0	B7	1...	
PDD0	FLG	0	73	...1	
PDD1	FLG	0	73	...1	
PDD2	FLG	0	73	1...	
PDD3	FLG	0	73	1...	
PODBIO0	FLG	0	A7	...1	
PODBIO1	FLG	0	A7	...1	

AS17K 21.10 D1 << E17201A DATA MEMORY MAP >> 15:19:55 02/19/91 PAGE 03-008

PROG -

SOURCE - SOUSIN.ASM

(DATA AREA INFORMATION)

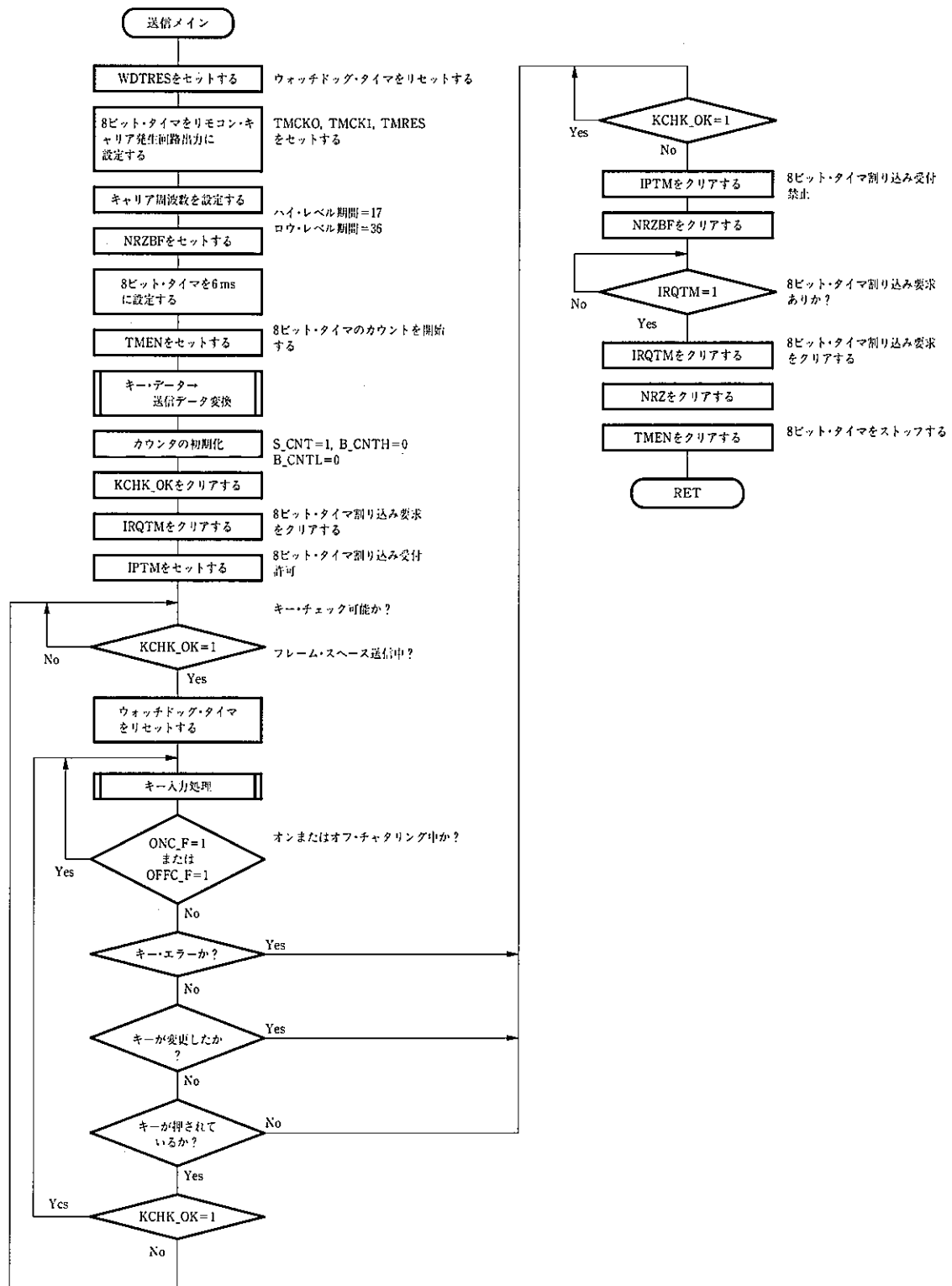
SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
P0DBIO2	FLG	0	A7	.1..	
P0DBIO3	FLG	0	A7	1...	
PIA0	FLG	1	70	...1	
PIA1	FLG	1	70	..1.	
PIA2	FLG	1	70	.1..	
PIA3	FLG	1	70	1...	
PIAGIO	FLG	0	B7	1...	
PSW	MEM	0	7F		
PSW_1	MEM	0	6A		スタックの過剰用のエリアです。
RPH	MEM	0	7D		
RPL	MEM	0	7E		
RPL_1	MEM	0	9B		スタックの過剰用のエリアです。
S_CNT	MEM	0	22		送信シーケンス・カウンタです。
SIOCK0	FLG	0	A2	...1	
SIOCK1	FLG	0	A2	..1.	
SIOEN	FLG	0	A3	...1	
SIOHIZ	FLG	0	A2	.1..	
SLOTS	FLG	0	A2	1...	
SYSCK	FLG	0	82	..1.	
TMCK0	FLG	0	B3	...1	
TMCK1	FLG	0	B3	..1.	
TMEN	FLG	0	B3	1...	
TMOE	FLG	0	A3	..1.	
TMRES	FLG	0	B3	.1..	
VDDDET0	FLG	0	87	..1..	
VDDDET1	FLG	0	87	1...	
VREFEN	FLG	0	A1	1...	
WDTRES	FLG	0	83	1...	
WORK1	MEM	0	0		汎用レジスタです。
WORK2	MEM	0	1		汎用レジスタです。
WORK3	MEM	0	2		汎用レジスタです。
WORK4	MEM	0	3		汎用レジスタです。
WR	MEM	0	76		
WR_1	MEM	0	69		スタックの過剰用のエリアです。
WTRND	FLG	0	83	.1..	
WTRRES	FLG	0	83	..1.	
XEN	FLG	0	82	...1	
Z	FLG	0	7F	..1.	

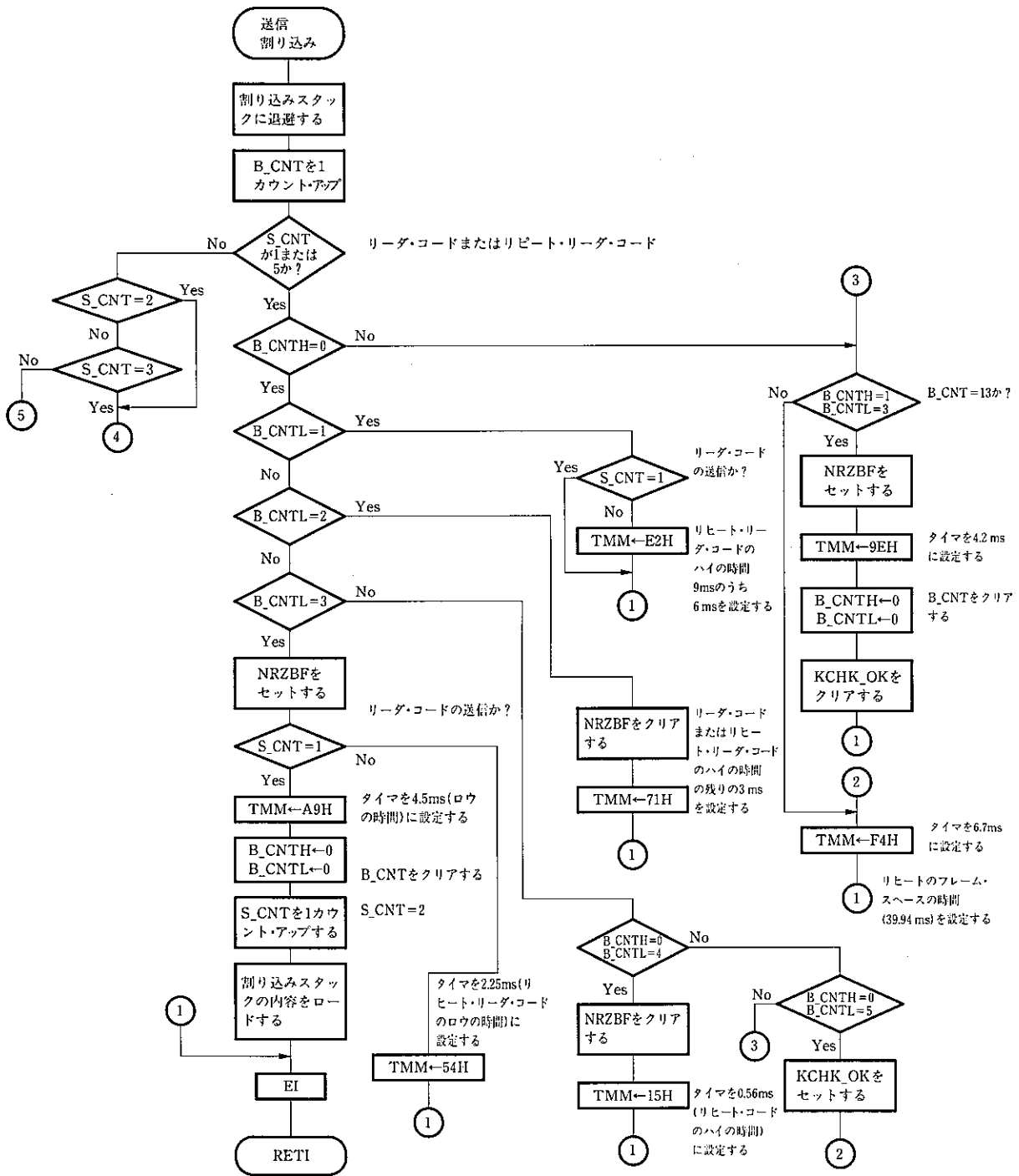


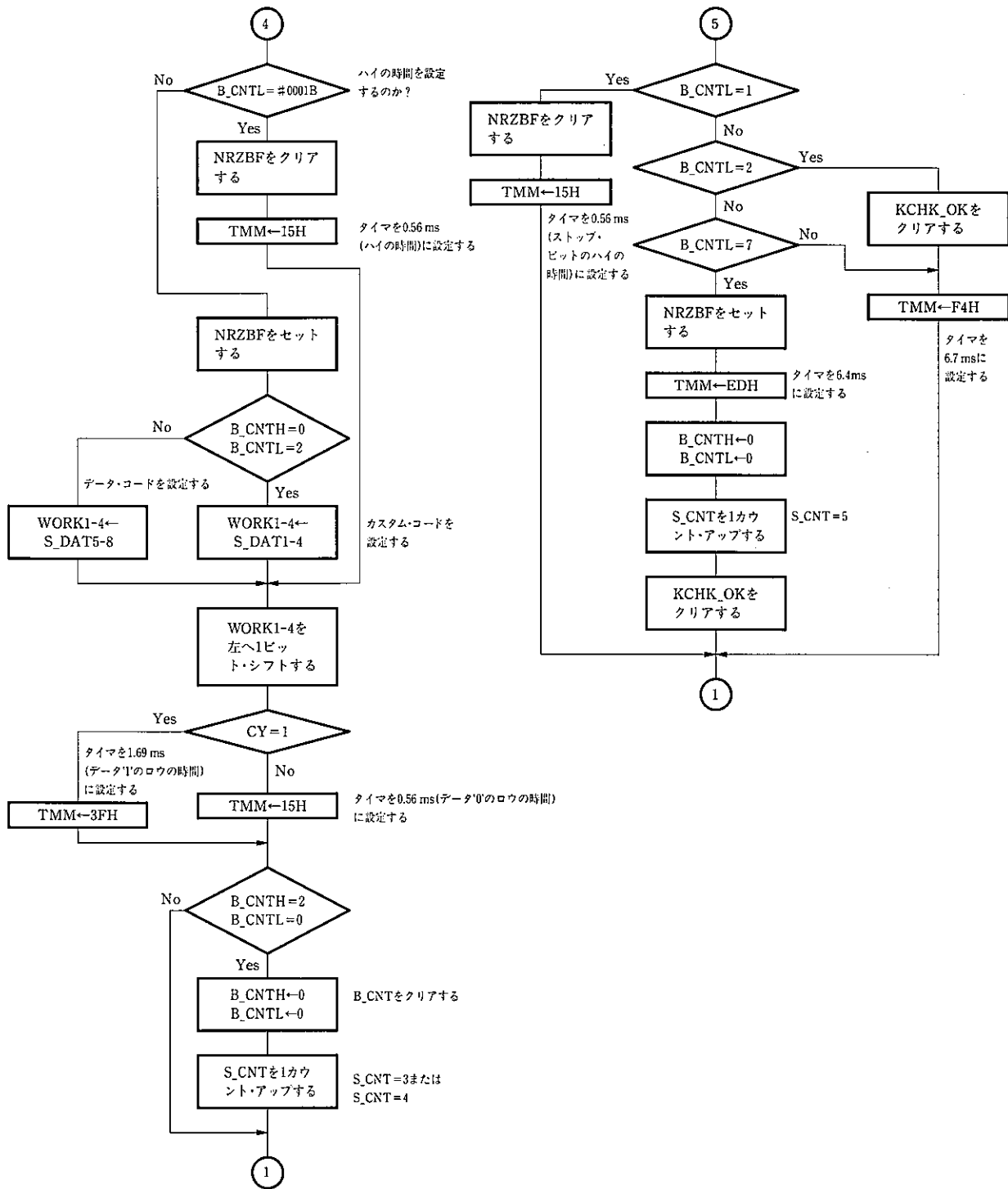
フロー・チャート



フロー・チャート







リスト


```
INCLUDE 'SOUSIN.PUB'
SUMMARY '%', '第5章 リモコン応用(送信)'
%%
```

```
SUMMARY '%', '<プログラム説明>'
```

リモコン・キャリア発生回路は、8ビット・カウンタ、ハイ・レベル期間設定用モジュロ・レジスタ(NRZ L T M)、ロウ・レベル期間設定用モジュロ・レジスタとコンパレータで構成されており、ハイ・レベル期間とロウ・レベル期間をそれぞれに対応するモジュロ・レジスタに設定することで、キャリア・デューティ比およびキャリア周波数を決定します。キャリア出力するREM端子の出力制御は、レジスタ・ファイル上のNRZ、NRZBFおよび8ビット・タイマで行います。プログラム例では、各キーに対応するの送信コードをテーブル・データより呼び出し、キャリア周波数：37.7kHz、デューティ比：1/3、ハイ・レベル期間：17、ロウ・レベル期間：38のリモコン信号の送信を行います。

```
%%
SUMMARY '%', '<RAM使用説明>'
```

```
WORK1 ..... 0.00H ..... 汎用レジスタです。
WORK2 ..... 0.01H ..... 汎用レジスタです。
WORK3 ..... 0.02H ..... 汎用レジスタです。
WORK4 ..... 0.03H ..... 汎用レジスタです。

S_DAT1 ..... 0.30H ..... 送信データ格納用のエリアです。
S_DAT2 ..... 0.31H ..... 送信データ格納用のエリアです。
S_DAT3 ..... 0.32H ..... 送信データ格納用のエリアです。
S_DAT4 ..... 0.33H ..... 送信データ格納用のエリアです。
S_DAT5 ..... 0.34H ..... 送信データ格納用のエリアです。
S_DAT6 ..... 0.35H ..... 送信データ格納用のエリアです。
S_DAT7 ..... 0.36H ..... 送信データ格納用のエリアです。
S_DAT8 ..... 0.37H ..... 送信データ格納用のエリアです。

WR_I ..... 0.3AH ..... WR用のスタック・メモリです。
PSW_I ..... 0.3BH ..... PSW用のスタック・メモリです。
RPL_I ..... 0.3CH ..... RPL用のスタック・メモリです。

S_CNT ..... 0.22H ..... 送信シーケンス・カウンタです。
B_CNTL ..... 0.23H ..... 送信波形の時間定数のカウンタです。
B_CNTH ..... 0.24H ..... 送信波形の時間定数のカウンタです。

CHATARING ..... 0.38H ..... チャタリングのフラグ用のエリアです。
KCHK_OK ..... #3 ..... 送信中にキーのチェックが可能な場合セットされます。
```

```
%%
EJECT
```

```
*****
: 使用メモリ、レジスタ、フラグ
*****
```

```
WORK1      MEM      0.00H      : 汎用レジスタです。
WORK2      MEM      0.01H      : 汎用レジスタです。
WORK3      MEM      0.02H      : 汎用レジスタです。
WORK4      MEM      0.03H      : 汎用レジスタです。

WR_I       MEM      0.3AH      : スタックの逃避用のエリアです。
PSW_I      MEM      0.3BH      : スタックの逃避用のエリアです。
RPL_I      MEM      0.3CH      : スタックの逃避用のエリアです。

S_CNT      MEM      0.22H      : 送信シーケンス・カウンタです。
B_CNTL     MEM      0.23H
B_CNTH     MEM      0.24H

CHATARING  MEM      0.38H
KCHK_OK    PLG      CHATARING.3 : 送信中にキーのチェックが可能な場合セットされます。
```

```
EJECT
```

```
*****
: 送信メイン
*****
```

```

;*****
SOUSIN:
SET1  WDTRES          ;ウォッチドッグ・タイマをリセット
SET3  TMCK0, TMCK1, TMRES ;8ビット・タイマをリモコン・キャリア発生回路出力に設定

;キャリア周波数の設定   キャリア周波数 = 37.74kHz   チューニング比 = 1/3
MOV   DBF1, #01H
MOV   DBF0, #02H
PUT   NRZHTMM, DBP      ;ハイ・レベル期間 = 1 7
MOV   DBF1, #02H
MOV   DBF0, #04H
PUT   NRZLTMM, DBF      ;ロウ・レベル期間 = 3 6
SET1  NRZBF

MOV   DBF1, #0EH        ;タイマを 6 m s に設定
MOV   DBF0, #02H
PUT   TMM, DBF
SET1  TMEN              ;8ビット・タイマ スタート
SUMMARY
      '%%', ' <送信コードの設定>'

各キーに対応する送信コード・データをテーブル・データより呼び出し、メモリ S_DAT1 - S_DAT8 に格納します。
S_DAT1 - S_DAT4 ..... カスタム・コード
S_DAT5 - S_DAT8 ..... データ・コード
,EJ-1
%%

      CALL  CODESET      ;送信コード・データの読み込み
      MOV   S_CNT, #1     ;各カウンタの初期設定
      MOV   B_CNTL, #0
      MOV   B_CNTH, #0
      CLR1  KCHK_OK
      CLR1  IRQTM
      SET1  IPTM          ;8ビット・タイマ割り込み要求クリア
                          ;8ビット・タイマ割り込み受付許可
EJECT
CHK_BIT:
SKT1   KCHK_OK          ;最終ビットか?
BR     CHK_BIT
SET1   WDTRES          ;ウォッチドッグ・タイマをリセット
KEY:
      CALL  KEYSKAN      ;キー入力処理
      SKT1  ONC_F
      SKF1  OFFC_F
      BR   KEY
      SKT1  K_ERROR
      SKF1  K_CHANGE
      BR   T_OFF
      SKT1  K_EXIST
      BR   T_OFF
      SKF1  KCHK_OK
      BR   KEY
      BR   CHK_BIT
T_OFF:
SKF1   KCHK_OK
BR     T_OFF
CLR1   IPTM
CLR1   NRZBF          ;8ビット・タイマによる割り込み受付禁止
T_WAIT:
SKT1   IRQTM          ;8ビット・タイマ割り込みありか?
BR     T_WAIT
CLR1   IRQTM
CLR1   NRZ
CLR1   TMEN          ;8ビット・タイマをストップする
RET
;

```

```

:.....
: 送信割り込み
:.....
TIME_IR: ; 割り込みスタックに返避する
        POKB   WR,WR
        PEEK   WR,PSW
        POKB   PSW,WR
        PEEK   WR,RPL
        POKB   RPL,WR
        MOV    PSW,#0
        MOV    RPL,#0
        ADD    B_CNTL,#1
        ADDC   B_CNTH,#0

SUMMARY  '%', '<送信シーケンス・カウンタのチェック>'

送信シーケンス・カウンタ S_CNT の値と役割

S_CNT = 1   の場合 リーダ・コードの時間データを設定します。
S_CNT = 2   の場合 カスタム・コードの時間データを設定します。
S_CNT = 3   の場合 データ・コードの時間データを設定します。
S_CNT = 4   の場合 ストップ・ビットおよびフレーム・スペースの時間データを設定します。
S_CNT = 5   の場合 リピート・リーダー・コードの時間データを設定します。

%%
SKE     S_CNT,#1
SKNB    S_CNT,#5
BR      READER ; 1 または 5 ならば、リーダー・コード または、ビット・コード の設定
SKE     S_CNT,#2
SKNB    S_CNT,#3 ; 2 または 3 ならば、カスタム・コード または、データ・コード の設定
BR      CT_DAT  ; 4 ならば、ストップ・ビット と フレーム・スペース の設定
;
READER:
SKE     B_CNTH,#0
BR      REPT4
SKNB    B_CNTL,#1 ; リーダー・コード のハイの時間 (9ms のうち 6ms ) を設定
BR      CHK_B
SKNB    B_CNTL,#2
BR      SET_3ms ; 残り 3ms の設定
SKE     B_CNTL,#3
BR      REPT1
SETI    NRZDF ; リーダー・コード のロウの時間 (4.5ms または 2.25ms) を設定
SKE     S_CNT,#1
BR      REPT0
MOV     DBF1,#0AH ; タイマを 4.5ms (ロウの時間) に設定
MOV     DBF0,#09H
PUT     TMM,DBF
MOV     B_CNTL,#0
MOV     B_CNTH,#0
ADD     S_CNT,#1 ; S_CNT を 2 (カスタム・コード 設定) に カウント・アップ

EI_RETI:
PEEK   WR,PSW_1 ; 割り込みスタックの内容をロード
POKE   PSW,WR
PEEK   WR,RPL_1
POKE   RPL,WR
PEEK   WR,WR_1

EI_RETI1:
EI
RETI
;
CHK_B:

```

```

SKNE S_CNT, #1
BR EI_RET1
MOV DBF1, #0EH ; リレト・リレト・コート のハイの時間 (9msのうち6ms) 設定
MOV DBF0, #02H
PUT TMM, DBF
BR EI_RET1
SET_3ms:
CLR1 NRZBF
MOV DBF1, #07H ; タイマを 3ms (リレト・リレト・コート のハイの時間の残り 3ms) に設定
MOV DBF0, #01H
PUT TMM, DBF
BR EI_RET1
;
REPT0:
MOV DBF1, #05H ; タイマを 2.25ms (リレト・リレト・コート のロウの時間) に設定
MOV DBF0, #04H
PUT TMM, DBF
BR EI_RET1
;
REPT1:
SKNE B_CNTH, #0
SKE B_CNTH, #4
BR REPT2
CLR1 NRZBF
MOV DBF1, #01H ; タイマを 0.56ms (リレト・リレト・コート のハイの時間) に設定
MOV DBF0, #05H
PUT TMM, DBF
BR EI_RET1
;
REPT2:
SKNE B_CNTH, #0
SKE B_CNTH, #6
BR REPT4
SET1 KCHK_OK
EJECT
REPT3:
MOV DBF1, #0FH ; タイマを 6.7ms に設定
MOV DBF0, #04H
PUT TMM, DBF
BR EI_RET1
;
REPT4:
SKNE B_CNTH, #1
SKE B_CNTH, #3
BR REPT3
SET1 NRZBF
MOV DBF1, #09H ; タイマを 4.2ms に設定
MOV DBF0, #0BH
PUT TMM, DBF
MOV B_CNTH, #0
MOV B_CNTH, #0
CLR1 KCHK_OK
BR EI_RET1
;
CT_DAT:
SKT B_CNTH, #0001B ; ハイの時間を設定するか?
BR DAT_LOW
CLR1 NRZBF
MOV DBF1, #01H ; タイマを 0.56ms (ハイの時間) に設定
MOV DBF0, #05H
PUT TMM, DBF
BR EI_RET1
;
DAT_LOW:
SET1 NRZBF ; ロウの時間を設定
SKNE B_CNTH, #0

```

```

SKE    B_CNTL,#2
BR     SHIFT
SKE    S_CNT,#2
BR     DAT_CODE
LD     WORK1,S_DAT1 ; カスタム・コードを設定
LD     WORK2,S_DAT2
LD     WORK3,S_DAT3
LD     WORK4,S_DAT4
BR     SHIFT
DAT_CODE:
LD     WORK1,S_DAT5 ; データ・コードを設定
LD     WORK2,S_DAT6
LD     WORK3,S_DAT7
LD     WORK4,S_DAT8
EJECT
SHIFT:
ADD    WORK4,WORK4 ; 送信データを左へ1ビット・シフト
ADDC   WORK3,WORK3
ADDC   WORK2,WORK2
ADDC   WORK1,WORK1
SKT1   CY
BR     DAT_0
MOV    DBF1,#03H ; タイマを 1.69ms (7*2^1' のロウの時間) に設定
MOV    DBF0,#0FH
PUT    TMM,DBF
BR     CT_DAT1
DAT_0:
MOV    DBF1,#01H ; タイマを 0.56ms (7*2^0' のロウの時間) に設定
MOV    DBF0,#05H
PUT    TMM,DBF
CT_DAT1:
SKNR   B_CNTH,#2
SKE    B_CNTL,#0
BR     EI_RET1
MOV    B_CNTH,#0
MOV    B_CNTL,#0
ADD    S_CNT,#1 ; S_CNT を 2 (7*2^1' コード 設定) または 3 (ストップ・ビット
BR     EI_RET1 ; 設定) にカウントアップ
;
SP_FLAME:
SKE    B_CNTL,#1
BR     FLAME
CLR1   NRZBF
MOV    DBF1,#01H ; タイマを 0.56ms (ストップ・ビット のハイの時間) に設定
MOV    DBF0,#05H
PUT    TMM,DBF
BR     EI_RET1
;
FLAME:
SKE    B_CNTL,#2
BR     FLAME2
SET1   KCHK_OK
FLAME1:
MOV    DBF1,#0FH ; タイマを 6.7ms に設定
MOV    DBF0,#04H
PUT    TMM,DBF
BR     EI_RET1
;
FLAME2:
SKE    B_CNTL,#7
BR     FLAME1
SET1   NRZBF
MOV    DBF1,#0EH ; タイマを 5.4ms に設定
MOV    DBF0,#0DH
PUT    TMM,DBF
MOV    B_CNTL,#0

```

```
MOV    B_CNTH,#0
ADD    S_CNT,#1      ; S_CNT を 5 (リモコンコード設定) にカウントアップ
CLR    KCHK_OK
BR     EI_RBT1
;
```

END

クロスレファレンス

AS17K Z1.10 D1 << E17201A XREF LIST >> 15:19:54 02/19/91 PAGE 03-002

PROG -

SOURCE - SOUSIN.ASM

SYMBOL	TYPE	A	VALUE	/REP	(#DEF)
TIME_IR	LAB	P	F8 /	1-8	,# 138
WORK1	MEM	L	0.00 /#	48	, 269 , 275 , 284 , 284
WORK2	MEM	L	0.01 /#	49	, 270 , 276 , 283 , 283
WORK3	MEM	L	0.02 /#	50	, 271 , 277 , 282 , 282
WORK4	MEM	L	0.03 /#	51	, 272 , 278 , 281 , 281
WR_1	MEM	L	0.69 /#	53	, 137 , 193

TOTAL SYMBOLS - 53

END OF XREF LIST

第6章 A/D変換

μPD17201Aは、8ビットの分解能で、4チャンネルのアナログ入力（ADC₀–ADC₃）を持つ逐次比較型A/Dコンバータを内蔵しています。

6.1 プログラム説明

温度センサにサーミスタ（6 kΩ/0℃）を使用して–20～+50℃（表示分解能1℃）の温度を計測するプログラムの例を紹介します。

サーミスタの温度に対する抵抗の変化は、次のような式で表すことができます。

$$R = R_0 \exp \left\{ B \left(\frac{1}{T} - \frac{1}{T_0} \right) \right\} \quad \dots\dots\dots \text{式 6-1}$$

R : 任意の温度T [°K] における抵抗値

T : 任意の温度 [°K]

R₀ : 基準となる温度T₀ [°K] における抵抗値

T₀ : 基準となる温度 [°K]

B : 基準となる温度T₀ [°K] とΔT₀ [°K] において求められる定数

(一般にB定数と呼ぶ)

ただし、B定数は一定ではなく温度により変化します。B定数は式6-1を変形して次のような式で求めることができます。

$$B = \frac{1}{\frac{1}{T} - \frac{1}{T_0}} \ln \frac{R}{R_0} \quad \dots\dots\dots \text{式 6-2}$$

また本プログラムでは図6-1に示すような回路により–20℃で0V、+50℃で5Vが入力されます。図中のR₁は+15℃のときの出力電圧が+50℃のときの出力電圧の1/2になるように設定されているものとしています。

図6-1 入力回路

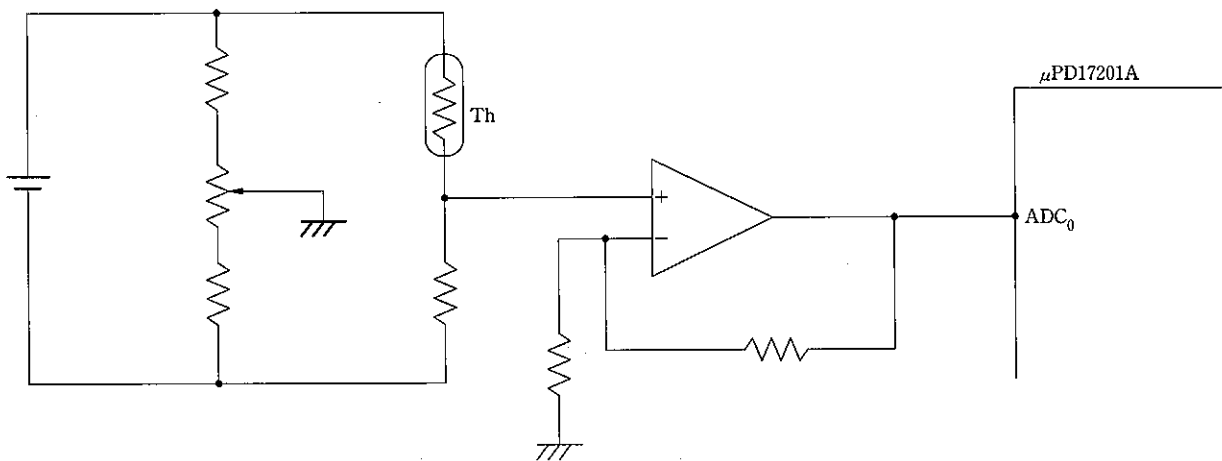


図6-2は、図6-1に示した回路の特性（計算値）を、5℃ごとにプロットしたグラフです。

この図を見ると分かるようにサーミスタの変化特性が直線でないため、A/D変換値を計算で温度に変換しようとする補正が必要になり、処理が複雑になります。本プログラムでは、表6-1に示すようなデータ・テーブルを持ち、これとA/D変換値を比較して温度を求めることにより処理を単純にしています。

図6-2 温度-出力特性

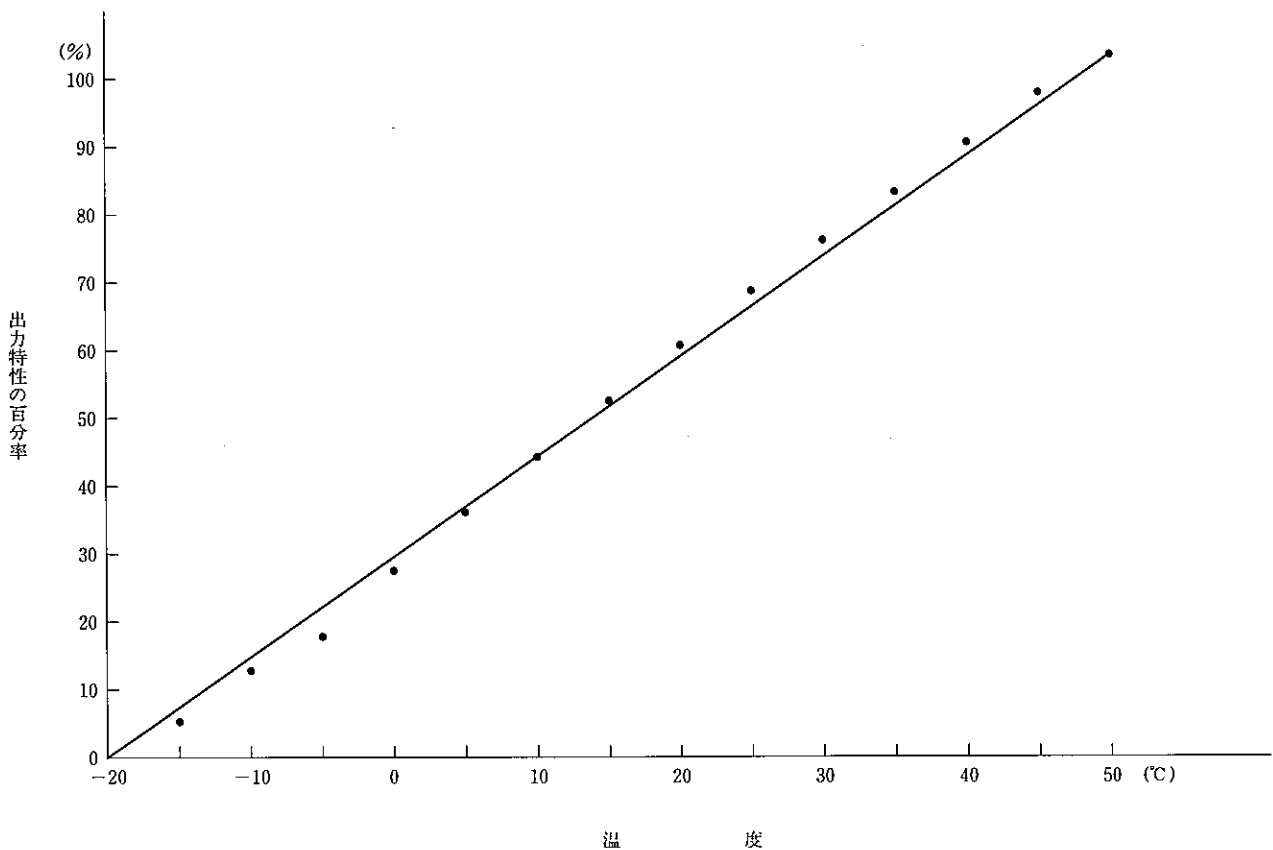


表 6-1 温度とA/D変換値の関係

温度[°C]	変換値[H]	温度[°C]	変換値[H]	温度[°C]	変換値[H]	温度[°C]	変換値[H]
-20	0	-2	38	16	86	34	CE
-19	1	-1	3C	17	8B	35	D2
-18	4	0	40	18	8F	36	D6
-17	7	1	48	19	93	37	D9
-16	A	2	4C	20	97	38	DC
-15	C	3	50	21	9B	39	E0
-14	F	4	54	22	9F	40	E3
-13	12	5	58	23	A3	41	E7
-12	16	6	5C	24	A8	42	EA
-11	19	7	60	25	AC	43	ED
-10	1C	8	64	26	B0	44	F0
-9	1F	9	69	27	B4	45	F3
-8	23	10	6D	28	B7	46	F6
-7	26	11	71	29	BB	47	F9
-6	2A	12	75	30	BF	48	FC
-5	2D	13	7A	31	C3	49	FE
-4	31	14	7E	32	C7	50	FF
-3	35	15	82	33	CB		

〈使用するハードウェア〉

番 号	端 子 名	使 用 説 明
36	ADC ₀	サーミスタからのアナログ値が入力されます。 その他の入力 (ADC ₁ –ADC ₃) には、何も接続しません。

- ・ A/Dコンバータ
- ・ 時計用タイマ

〈初期設定〉

A/D検索回数 (ADCNT) を 0 クリア

A/Dのステータスをクリア

A/Dコンバータのモード設定

VREFEN = 1

ADCEN = 1

ADCCH1 = 1, ADCCH0 = 0

〈起動方法〉

本プログラム例では、AD_CHECKルーチンをコールすることで、A/Dの比較を行うかどうかを、自動的に判断している。

ドキュメント

AS17K Z1.10 D1 << E17201A DOCUMENT >> 11:03:22 02/25/91

TABLE OF CONTENTS

	PAGE
第 6 章 A/D変換	1
1. A/D値検索ルーチン	2
2. 表示用A/D値決定ルーチン	5

第 6 章 A/D変換

```
PUBLIC (MEM) : AD_NOWH, AD_NOWL
PUBLIC (PLG) : AD_CHK, AD_OK
PUBLIC (LAB) : AD_CHECK
ADDR RANGE : 0005H - 0089H
```

<プログラム説明>

本プログラムは、 μ PD17201AのA/Dコンバータを使ったプログラムです。
以下に、本プログラムでのA/Dコンバータの使用方法を説明します。

- (1) 端子
使用する端子は、ADC0端子です。
 - ・ADC0端子
ADC0端子には、外部からアナログ値が入力され、内部のA/Dコンバータにより、本プログラムで設定した値と比較されます。そのときの結果が、レジスタ・ファイル内のADCCMPフラグに設定されます。
- (2) タイマ
本プログラムでは、時計用タイマを使用していますが、A/Dコンバータを使用するにあたっては、まったく関係がありません。本プログラムでは、A/D値の入力を1秒間隔で行っています。この1秒を作るのに時計用タイマを使用しています。
- (3) 割り込み
本プログラムでは、時計用タイマの割り込みのみ使用しています。プログラム中ときどきEI/DI命令が使われていますが、これは割り込みの発生をソフト的に制御しているからです。割り込みを使用しないときには使う必要はありません。

```
ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REPERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REPERRED      :-
DATA REPERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
```

1. A/D値検索ルーチン
本ルーチンでは、ADC0端子より入力されたアナログ値を判定し、そのときの設定デジタル値を得するためのルーチンです。
このルーチンでは、入力アナログ値がどのような値であるかを、バイナリ・サーチ法を使って検索しています。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

・初期化

A/D値を決定するときの各メモリに、初期値を設定します。
ここでは、検索するA/D値の範囲を最大範囲に設定します。
最大範囲とは、
A/D値の上限 = FFH
A/D値の下限 = 00H

```

ENTRANCES          :-AD_COMP
MEMORIES CHANGED   :-AD_LOWL   AD_LOWL
                   :-AD_UPH     AD_UPL
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

・検索ルーチン

ここから、バイナリ・サーチ法を使って、入力アナログ値を検索します。
A/D時のバイナリ・サーチ法は、以下の順に行います。

- (1) 上限値と下限値を比較し、等しければ検索を終了します。
- (2) 上限値と下限値の中間点である中間値をD/Aコンバータに設定します。
- (3) 中間値と入力アナログ値を比較します。
- (4) 中間値 \leq 入力アナログ値であれば、中間値を新たな下限値とします。
- (5) 中間値 $>$ 入力アナログ値であれば、中間値を新たな上限値とします。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

- (1) 検索の終了判定
 上限値と下限値を比較し、等しければ、検索を終了しWORK0, WORK1に下限値の値を設定します。

```

ENTRANCES          :AD_LOOP
MEMORIES CHANGED   :WORK0          WORK1
MEMORIES REFERRED  :AD_LOWH        AD_LOWL
                   :AD_UPH          AD_UPL
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :NEXT_AD        ;結果が0でなければ検索を継続
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

- (2) 中間値をD/Aコンバータに設定
 中間値を得るために上限値から下限値を減算し、その結果1ビット右にシフトし、さらにその結果を下限値に加えています。

```

ENTRANCES          :NEXT_AD
MEMORIES CHANGED   :-
MEMORIES REFERRED  :AD_LOWH        AD_LOWL
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

- (3) 比較
 D/A値に中間値を設定した後に、中間値のアナログ値と入力アナログ値を比較し、結果をADCCMPに設定するまでにシステム・クロックがメイン・クロックとなっている場合には4ステップ分だけ待ち時間が必要です。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :AD_LOWER        ;中間値 > 入力アナログ値
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

(4) 中間値 \leq 入力アナログ値
 ここでは、通常は、中間値を下限値に設定しますが、通常でない場合が考えられます。

- ・ 中間値 = 下限値のとき
 このときには、上限値と下限値との差が1のときなので、上限値をD/Aコンバータに設定して、そのときの結果が定めます。
- ・ 上限値 \leq 入力アナログ値のときは、上限値を下限値に設定します。
- ・ 上限値 $>$ 入力アナログ値のときは、下限値を上限値に設定します。

```

ENTRANCES          :-
MEMORIES CHANGED   :AD_LOWH      AD_LOWL
                   :AD_UPH       AD_UPL
MEMORIES REFERRED  :AD_LOWH      AD_LOWL
                   :AD_UPH       AD_UPL
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :AD_LOOP      ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
  
```

(5) 中間値 $>$ 入力アナログ値
 このときには、中間値を上限値に設定します。

```

ENTRANCES          :AD_LOWER
MEMORIES CHANGED   :AD_UPH      AD_UPL
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :AD_LOOP      ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
  
```

2. 表示用 A/D 値決定ルーチン
 本ルーチンでは、入力したアナログ値にあった表示を行うための表示用データ決定ルーチンです。
 ここでは、A/D 値検索ルーチンより得た A/D 値を 1 秒ごとに 3 回読み込み、3 回読み込んだときの中間値を表示用データとしてい
 ます。
 そのときの最終データは、AD_NOWH, AD_NOWL のメモリに設定されます。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

・ A/D 値検索ルーチンの呼び出し
 ここでは、A/D 値検索ルーチンを呼び出すか否かを AD_CHK フラグの値により判断します。AD_CHK フラグは、時計用タイマにより
 1 秒ごとにセットされます。

```

ENTRANCES          :AD_CHECK
MEMORIES CHANGED   :AD_CNT
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :AD_CHK
FLAGS REFERRED      :AD_CHK
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED :AD_COMP
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

- A/D 値を 1 回検索したとき
1 回目のときは, AD_RD1H, AD_RD1L のメモリに, A/D 値を設定します。

```
ENTRANCES          :-  
MEMORIES CHANGED   :AD_RD1H      AD_RD1L  
MEMORIES REFERRED  :WORK0        WORK1  
MEMORIES MANIPULATED :-  
FLAGS CHANGED      :-  
FLAGS REFERRED     :-  
DATA REFERRED      :-  
BRANCH TO          :AD_CHK_2      ;  
SUBROUTINES CALLED :-  
LABELS MANIPULATED :-  
SYSTEM CALL        :-
```

- A/D 値を 2 回検索したとき
2 回目のときは, AD_RD2H, AD_RD2L のメモリに, A/D 値を設定します。

```
ENTRANCES          :AD_CHK_2  
MEMORIES CHANGED   :AD_RD2H      AD_RD2L  
MEMORIES REFERRED  :WORK0        WORK1  
MEMORIES MANIPULATED :-  
FLAGS CHANGED      :-  
FLAGS REFERRED     :-  
DATA REFERRED      :-  
BRANCH TO          :AD_CHK_3      ;  
SUBROUTINES CALLED :-  
LABELS MANIPULATED :-  
SYSTEM CALL        :-
```

- ・ A/D 値を 3 回検出したとき
 3 回目のときは、AD_RD3H, AD_RD3L のメモリに、A/D 値を設定します。
 次に、1 回目、2 回目、3 回目の A/D 値をソート用メモリに設定します。
 ここで、それぞれのデータを昇順にソートし、中間値を表示用データとします。
 また、次からは、1 秒ごとに最終データの設定を行うように、A/D カウンタの値を 2 回目に設定します。

```

ENTRANCES          :AD_CG          AD_CHK_3
MEMORIES CHANGED   :AD_1H          AD_1L
                   :AD_2H          AD_2L
                   :AD_3H          AD_3L
                   :AD_CHT          AD_RD3H
                   :AD_RD3L        WORK0
                   :WORK1          WORK2
                   :WORK3
MEMORIES REFERRED   :AD_RD1H        AD_RD1L
                   :AD_RD2H        AD_RD2L
                   :WORK0          WORK1
                   :WORK2          WORK3
MEMORIES MANIPULATED :-
FLAGS CHANGED      :AD_CHANG
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :CG_END          ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

- ・ 比較を終了
 比較終了時に、比較をしていたときに値の入れ替えがあったかをチェックします。もし、入れ替えがあった場合には、再び比較を行います。
 また、次の入力のために、A/D 値の移動を行っています。

```

ENTRANCES          :CG_END
MEMORIES CHANGED   :AD_NOWH        AD_NOWL
                   :AD_RD1H        AD_RD1L
                   :AD_RD2H        AD_RD2L
                   :WORK0          WORK1
MEMORIES REFERRED   :AD_2H          AD_2L
                   :AD_RD2H        AD_RD2L
                   :AD_RD3H        AD_RD3L
                   :WORK0          WORK1
MEMORIES MANIPULATED :-
FLAGS CHANGED      :AD_OK
FLAGS REFERRED     :AD_CHANG
DATA REFERRED      :-
BRANCH TO          :AD_CG          ;再び比較
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```


RAMマップ



AS17K Z1.10 D1 << B17201A DATA MEMORY MAP >> 11:02:59 02/25/91 PAGE 02-001

PROG =

SOURCE = AD.ASM

(PUBLIC/LOCAL MAP)

BANK = 0

ROW

	7	6	5	4	3	2	1	0	
		!LCDD32	!LCDD16	!LCDD0					0
								!WORK0	
		!LCDD33	!LCDD17	!LCDD1					1
								!WORK1	
		!LCDD34	!LCDD18	!LCDD2					2
								!WORK2	
		!LCDD35	!LCDD19	!LCDD3					3
							!AD_CNT	!WORK3	
!AR3			!LCDD20	!LCDD4					4
						!AD_1H	!AD_RD1H		
			!LCDD21	!LCDD5					5
						!AD_1L	!AD_RD1L		
!AR1			!LCDD22	!LCDD6					6
						!AD_2H	!AD_RD2H		
!AR0			!LCDD23	!LCDD7					7
						!AD_2L	!AD_RD2L		
!WR			!LCDD24	!LCDD8					8
						!AD_3H	!AD_RD3H		
!BANK			!LCDD25	!LCDD9					9
						!AD_3L	!AD_RD3L		
!IXH			!LCDD26	!LCDD10					A
!MPH						!AD_FLAG	!AD_UPH		
!IXM			!LCDD27	!LCDD11					B
!MPL							!AD_UPL		

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 11:02:59 02/25/91 PAGE 02-002

PROG =

SOURCE = AD.ASM

(PUBLIC/LOCAL MAP)

BANK = 0

ROW

7	6	5	4	3	2	1	0	
IXL		LCDD28	LCDD12				IDBF3	C
						AD_LOWH		
IRPH		LCDD29	LCDD13				IDBF2	D
						AD_LOWL		
IRPL		LCDD30	LCDD14			AD_NOWH	IDBF1	E
IPSW		LCDD31	LCDD15			AD_NOWL	IDBF0	F

AS17K Z1.10 D1 << B17201A FLG MAP >> 11:02:59 02/25/91 PAGE 02-003

PROG -

SOURCE - AD.ASM

(FLAG MAP) BANK - 0

MSB	3	2	1	0	LSB
2A			IAD_CHK	IAD_OK	
		IAD_CHANG			
70	IPOA3	IPOA2	IPOA1	IPOA0	
71	IPOB3	IPOB2	IPOB1	IPOB0	
72	IPOC3	IPOC2	IPOC1	IPOC0	
73	IP0D3	IP0D2	IP0D1	IP0D0	
7A	IMPR				
7E				IBCD	
7F	ICMP	ICY	IZ	IXE	
82			ISYSCK	IXEN	
83	IWTRES	IWTMD	IWTRES		
87	IVDDDET1	IVDDDET0			
8P				INT	
91				INRZBP	
92				INRZ	
A0				IADCCMP	
A1	IVREFEN	IADCBN	IADCC1	IADCC0	
A2	SIOTS	SIOHIZ	SIOCK1	SIOCK0	

AS17K Z1.10 D1 << B17201A FLG MAP >> 11:02:59 02/25/91 PAGE 02-004

PROG -

SOURCE - AD.ASM

(FLAG MAP) BANK - 0

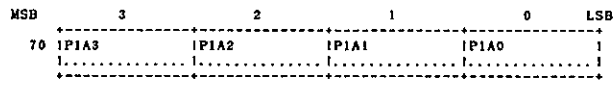
MSB	3	2	1	0	LSB
A3		INRZEN	TMOE	ISIOEN	
A7	IP0DBIO3	IP0DBIO2	IP0DBIO1	IP0DBIO0	
AP	IPSIO	IPWTM	IP	IPTM	
B1	LCDEN	LCDCK2	LCDCK1	LCDCK0	
B2	LCDMD3	LCDMD2	LCDMD1	LCDMD0	
B3	TMEN	TMRES	TMCK1	TMCK0	
B7	PIAGIO	PCG10	PBG10	PAG10	
BB	IRQSIO				
BC		IRQTM			
BD			IRQ		
BE				IRQTM	

AS17K Z1.10 D1 << E17201A FLG MAP >> 11:02:59 02/25/91 PAGE 02-005

PROG -

SOURCE - AD.ASM

(FLAG MAP) BANK - 1



RAM使用説明

AS17K Z1.10 D1 << R17201A DATA MEMORY MAP >> 11:02:59 02/25/91 PAGE 02-006

PROG -

SOURCE - AD.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
AD_1H	MEM	0	24		ソート時の最中小A/D
AD_1L	MEM	0	25		ート時の最中小A/D
AD_2H	MEM	0	26		ソート時の最中小A/D
AD_2L	MEM	0	27		ソート時の最中小A/D
AD_3H	MEM	0	28		ソート時の最中小A/D
AD_3L	MEM	0	29		ソート時の最中小A/D
AD_CHANG	FLG	0	2A	..1.	ソート時の最中小A/D
AD_CHK	FLG	0	2A	..1.	ソート時の最中小A/D
AD_CNT	MEM	0	13		A/D変換時の入力値
AD_FLAG	MEM	0	2A		A/D変換時の入力値
AD_LOWH	MEM	0	1C		A/D変換時の入力値
AD_LOWL	MEM	0	1D		A/D変換時の入力値
AD_NOWH	MEM	0	1E		最終値
AD_NOWL	MEM	0	1F		最終値
AD_OK	FLG	0	2A	...1	最終値
AD_RD1H	MEM	0	14		1回目の入力値
AD_RD1L	MEM	0	15		1回目の入力値
AD_RD2H	MEM	0	16		2回目の入力値
AD_RD2L	MEM	0	17		2回目の入力値
AD_RD3H	MEM	0	18		3回目の入力値
AD_RD3L	MEM	0	19		3回目の入力値
AD_UPH	MEM	0	1A		A/D変換時の入力値
AD_UPL	MEM	0	1B		A/D変換時の入力値
ADCCH0	FLG	0	A1	...1	
ADCCH1	FLG	0	A1	..1.	
ADCCH2	FLG	0	A0	...1	
ADCEN	FLG	0	A1	..1.	
AR0	MEM	0	77		
AR1	MEM	0	76		
AR2	MEM	0	75		
AR3	MEM	0	74		
BANK	MEM	0	79		
BCD	FLG	0	7E	...1	
CMP	FLG	0	7F	1...	
CY	FLG	0	7F	..1.	
DBF0	MEM	0	F		
DBF1	MEM	0	E		
DBF2	MEM	0	D		
DBF3	MEM	0	C		
INT	FLG	0	BF	...1	
IP	FLG	0	AF	..1.	
IFSIO	FLG	0	AF	1...	
IFTM	FLG	0	AF	...1	
IFWTM	FLG	0	AF	..1.	
IRQ	FLG	0	BD	..1.	
IRQSIO	FLG	0	BB	1...	
IRQTM	FLG	0	BE	...1	
IRQWTM	FLG	0	BC	..1.	
IXE	FLG	0	7F	...1	
IXH	MEM	0	7A		
IXL	MEM	0	7C		
IXM	MEM	0	7B		

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 11:02:59 02/25/91 PAGE 02-007

PROC -

SOURCE - AD.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCDCCK0	FLG	0	B1	...	1
LCDCCK1	FLG	0	B1	...	1
LCDCCK2	FLG	0	B1	...	1
LCDD0	MEM	0	40		
LCDD1	MEM	0	41		
LCDD10	MEM	0	4A		
LCDD11	MEM	0	4B		
LCDD12	MEM	0	4C		
LCDD13	MEM	0	4D		
LCDD14	MEM	0	4E		
LCDD15	MEM	0	4F		
LCDD16	MEM	0	50		
LCDD17	MEM	0	51		
LCDD18	MEM	0	52		
LCDD19	MEM	0	53		
LCDD2	MEM	0	42		
LCDD20	MEM	0	54		
LCDD21	MEM	0	55		
LCDD22	MEM	0	56		
LCDD23	MEM	0	57		
LCDD24	MEM	0	58		
LCDD25	MEM	0	59		
LCDD26	MEM	0	5A		
LCDD27	MEM	0	5B		
LCDD28	MEM	0	5C		
LCDD29	MEM	0	5D		
LCDD3	MEM	0	43		
LCDD30	MEM	0	5E		
LCDD31	MEM	0	5F		
LCDD32	MEM	0	60		
LCDD33	MEM	0	61		
LCDD34	MEM	0	62		
LCDD35	MEM	0	63		
LCDD4	MEM	0	44		
LCDD5	MEM	0	45		
LCDD6	MEM	0	46		
LCDD7	MEM	0	47		
LCDD8	MEM	0	48		
LCDD9	MEM	0	49		
LCDDEN	FLG	0	B1	...	1
LCDDM0	FLG	0	B2	...	1
LCDDM1	FLG	0	B2	...	1
LCDDM2	FLG	0	B2	...	1
LCDDM3	FLG	0	B2	...	1
MPE	FLG	0	7A	...	1
MPH	MEM	0	7A		
MPL	MEM	0	7B		
NRZ	FLG	0	92	...	1
NRZBF	FLG	0	91	...	1
NRZEN	FLG	0	A3	...	1
POA0	FLG	0	70	...	1
POA1	FLG	0	70	...	1

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 11:02:59 02/25/91 PAGE 02-008

PRG -

SOURCE - AD.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
P0A2	FLG	0	70	.1..	
P0A3	FLG	0	70	1...	
P0AG10	FLG	0	87	...1	
P0B0	FLG	0	71	...1	
P0B1	FLG	0	71	.1..	
P0B2	FLG	0	71	.1..	
P0B3	FLG	0	71	1...	
P0BG10	FLG	0	87	...1	
P0C0	FLG	0	72	...1	
P0C1	FLG	0	72	.1..	
P0C2	FLG	0	72	.1..	
P0C3	FLG	0	72	1...	
P0CG10	FLG	0	87	.1..	
P0D0	FLG	0	73	...1	
P0D1	FLG	0	73	.1..	
P0D2	FLG	0	73	.1..	
P0D3	FLG	0	73	1...	
P0DB100	FLG	0	A7	...1	
P0DB101	FLG	0	A7	.1..	
P0DB102	FLG	0	A7	.1..	
P0DB103	FLG	0	A7	1...	
P1A0	FLG	1	70	...1	
P1A1	FLG	1	70	.1..	
P1A2	FLG	1	70	.1..	
P1A3	FLG	1	70	1...	
P1AG10	FLG	0	87	1...	
PSW	MEM	0	7F		
RPH	MEM	0	7D		
RPL	MEM	0	7E		
SI0CK0	FLG	0	A2	...1	
SI0CK1	FLG	0	A2	.1..	
SI0EN	FLG	0	A3	...1	
SI0HIZ	FLG	0	A2	.1..	
SI0TS	FLG	0	A2	1...	
SY0CK	FLG	0	82	...1	
TM0CK0	FLG	0	B3	...1	
TM0CK1	FLG	0	B3	.1..	
TM0EN	FLG	0	B3	1...	
TM0E	FLG	0	A3	.1..	
TMRES	FLG	0	B3	.1..	
VDDDET0	FLG	0	87	.1..	
VDDDET1	FLG	0	87	1...	
VREFEN	FLG	0	A1	1...	
WDTRES	FLG	0	83	1...	
WORK0	MEM	0	0		汎用レジスタ (A/D判定ルーチンからの戻り値の上位)
WORK1	MEM	0	1		汎用レジスタ (A/D判定ルーチンからの戻り値の下部)
WORK2	MEM	0	2		汎用レジスタ
WORK3	MEM	0	3		汎用レジスタ
WR	MEM	0	78		
WTM0D	FLG	0	83	.1..	
WTMRES	FLG	0	83	...1	
XEN	FLG	0	82	...1	

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 11:02:59 02/26/91 PAGE 02-009

PROG =

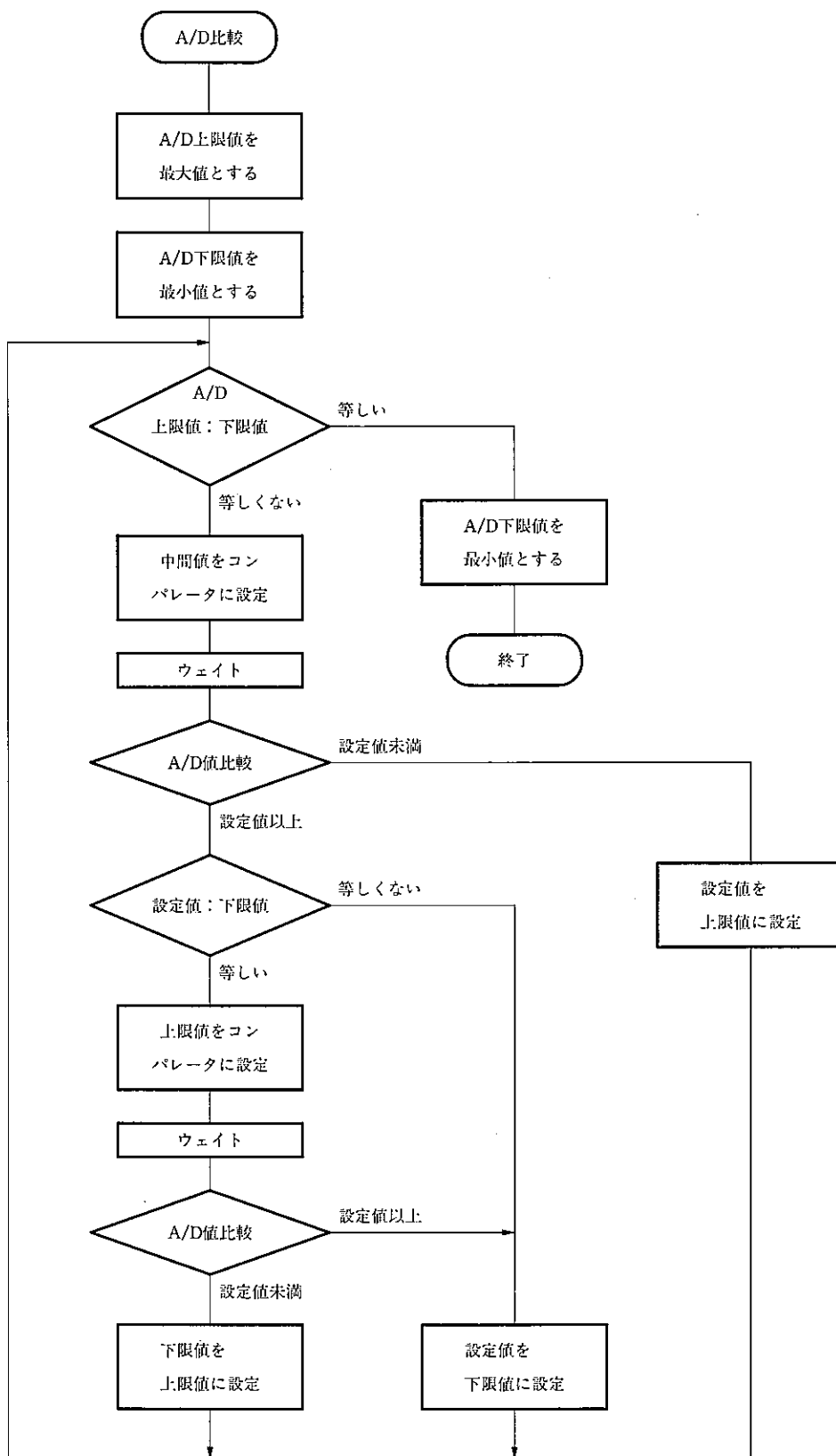
SOURCE = AD.ASM

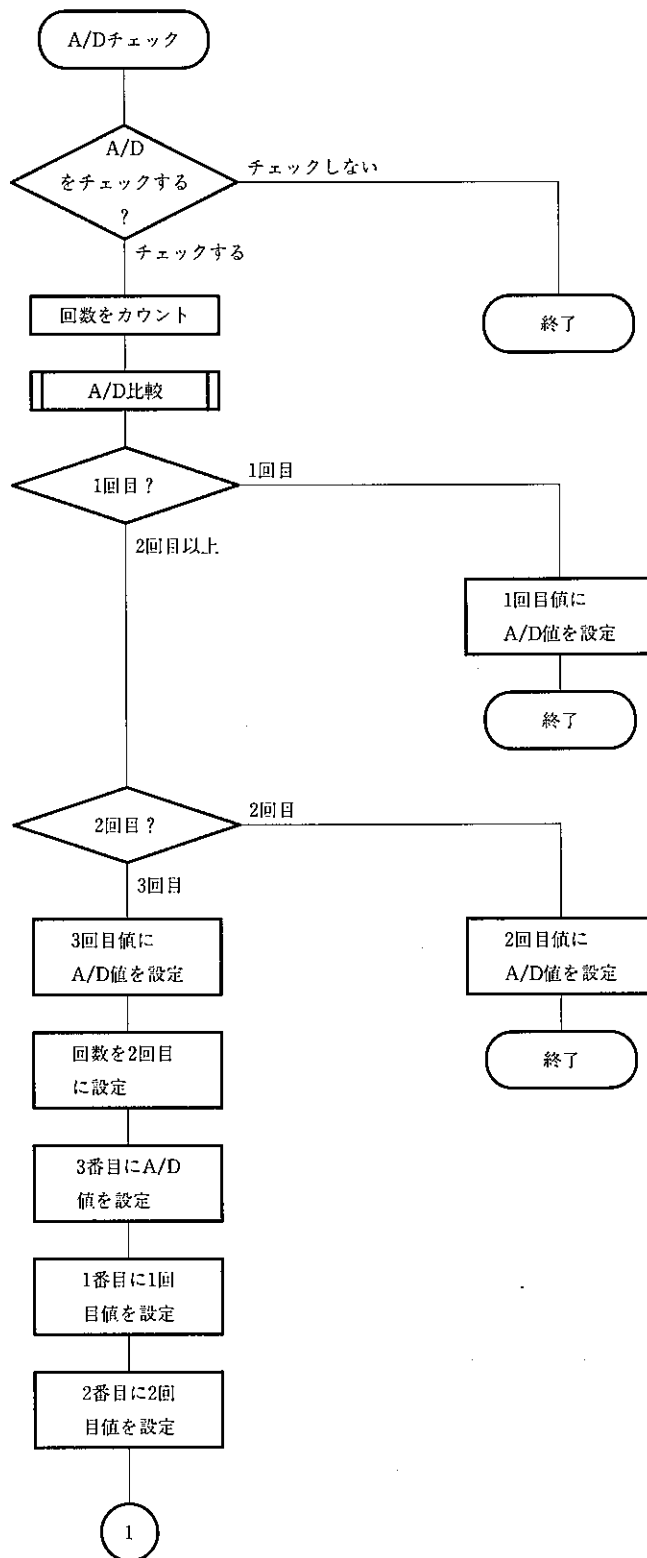
(DATA AREA INFORMATION)

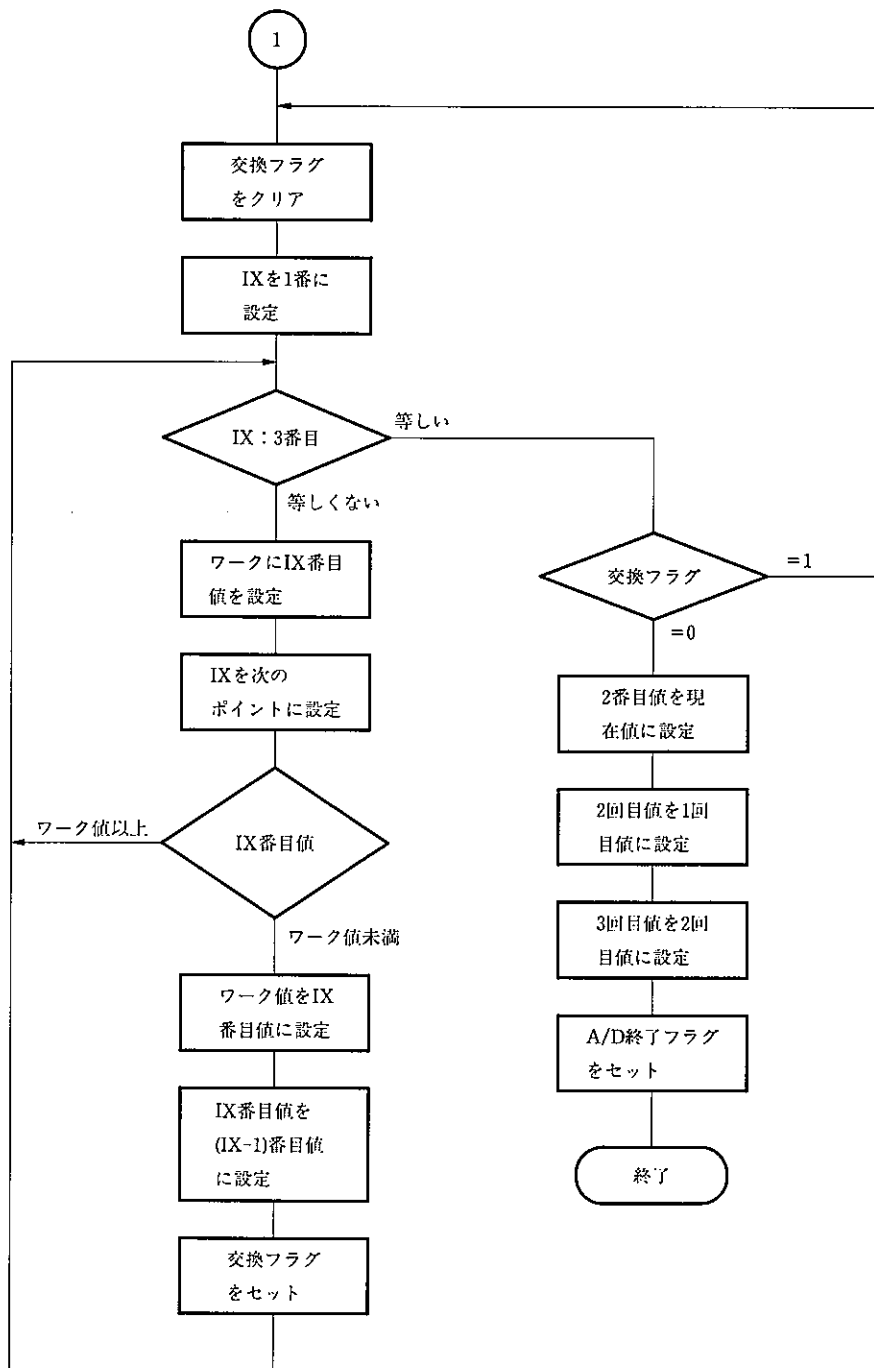
SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
Z	FLG	0	7F	..1.	

フロー・チャート

フロー・チャート







リスト


```

;*****
;***  A/Dプログラム  ***
;*****
SUMMARY '%%','第 6 章 A/D変換'
%%
;-----
; PUBLIC宣言
;-----
;--- LABEL ---
PUBLIC AD_CHECK

;--- MEMORY ---
PUBLIC AD_NOWH, AD_NOWL

;--- FLAG ---
PUBLIC AD_OK, AD_CHK

SUMMARY '%%',''
<プログラム説明>
本プログラムは、μPD17201AのA/Dコンバータを使ったプログラムです。
以下に、本プログラムでのA/Dコンバータの使用法を説明します。

(1) 端子
使用する端子は、ADC0端子です。
・ADC0端子
ADC0端子には、外部からアナログ値が入力され、内部のA/Dコンバータにより、本プログラムで設定した値と比較されます。そのときの結果が、レジスタ・ファイル内のADCCMPフラグに設定されます。

(2) タイマ
本プログラムでは、時計用タイマを使用していますが、A/Dコンバータを使用するにあたっては、まったく関係がありません。本プログラムでは、A/D値の入力を1秒間隔で行っています。この1秒を作るのに時計用タイマを使用しています。

(3) 割り込み
本プログラムでは、時計用タイマの割り込みのみ使用しています。本プログラム中とまどきEI/DI命令が使われていますが、これは割り込みの発生をソフト的に制御しているからです。割り込みを使用しないときには使う必要はありません。

.EJ-1
%%
SUMMARY '%%',' 1. A/D値検索ルーチン'
本ルーチンでは、ADC0端子より入力されたアナログ値を判定し、そのときの設定デジタル値を導すためのルーチンです。このルーチンでは、入力アナログ値がどのような値であるかを、バイナリ・サーチ法を使って検索しています。
%%
;-----
; メモリ定義
;-----
WORK0 MEM 0.00H ;汎用レジスタ (A/D判定ルーチンからの戻り値の上位)
WORK1 MEM 0.01H ;汎用レジスタ (A/D判定ルーチンからの戻り値の下位)
WORK2 MEM 0.02H ;汎用レジスタ
WORK3 MEM 0.03H ;汎用レジスタ

AD_CNT MEM 0.13H ;A/D値の取り込み回数カウンタ用メモリ
AD_RD1H MEM 0.14H ;1回目の入力A/D値の上位
AD_RD1L MEM 0.15H ;1回目の入力A/D値の下位
AD_RD2H MEM 0.16H ;2回目の入力A/D値の上位
AD_RD2L MEM 0.17H ;2回目の入力A/D値の下位
AD_RD3H MEM 0.18H ;3回目の入力A/D値の上位
AD_RD3L MEM 0.19H ;3回目の入力A/D値の下位
AD_UPH MEM 0.1AH ;A/D値検索中の上位
AD_UPL MEM 0.1BH ;A/D値検索中の下位
AD_LOWH MEM 0.1CH ;A/D値検索中の上位
AD_LOWL MEM 0.1DH ;A/D値検索中の下位
AD_NOWH MEM 0.1EH ;最終A/D値の上位
AD_NOWL MEM 0.1FH ;最終A/D値の下位

```

```

AD_1H      MEM      0.24H      ;ソート時の最小A/D値の上位
AD_1L      MEM      0.25H      ;ソート時の最小A/D値の下部
AD_2H      MEM      0.26H      ;ソート時の中間A/D値の上位
AD_2L      MEM      0.27H      ;ソート時の中間A/D値の下部
AD_3H      MEM      0.28H      ;ソート時の最大A/D値の上位
AD_3L      MEM      0.29H      ;ソート時の最大A/D値の下部

AD_FLAG    MEM      0.2AH      ;A/D変換時の状態フラグ用メモリ
AD_OK      FLG      AD_FLAG.0   ;最終A/D値が決定したとき(-1)
AD_CHK     FLG      AD_FLAG.1   ;A/D値の入力を行うとき(-1)
AD_CHANG   FLG      AD_FLAG.2   ;ソート用のフラグ ソートが終了したとき(-0)
    
```

```

EJECT
;*****
;      A/D比較
;*****
AD_COMP:
    
```

SUMMARY '%', ''

・初期化

A/D値を決定するときの各メモリに、初期値を設定します。
 ここでは、検索するA/D値の範囲を最大範囲に設定します。
 最大範囲とは、
 A/D値の上限 = FFH
 A/D値の下限 = 00H

```

%%
MOV      AD_UFH, #0FH      ;上限値の初期化
MOV      AD_UFL, #0FH
MOV      AD_LOWH, #0
MOV      AD_LOWL, #0
    
```

SUMMARY '%', ''

・検索ルーチン

ここから、バイナリ・サーチ法を使って、入力アナログ値を検索します。
 A/D時のバイナリ・サーチ法は、以下の順に行います。

- (1) 上限値と下限値を比較し、等しければ検索を終了します。
- (2) 上限値と下限値の中間点である中間値をD/Aコンバータに設定します。
- (3) 中間値と入力アナログ値を比較します。
- (4) 中間値 ≤ 入力アナログ値であれば、中間値を新たな下限値とします。
- (5) 中間値 > 入力アナログ値であれば、中間値を新たな上限値とします。

```

%%
AD_LOOP:
    
```

SUMMARY '%', ''

- (1) 検索の終了判定
 上限値と下限値を比較し、等しければ、検索を終了しWORK0, WORK1に下限値の値を設定します。

```

%%
LD      DBF1, AD_UFH      ;上限値から下限値を減算
LD      DBF0, AD_UFL
SUB     DBF0, AD_LOWL
SUBC   DBF1, AD_LOWH
SKNE   DBF1, #0          ;減算結果を判定
SKE    DBF0, #0
BR     NEXT_AD          ;結果が0でなければ検索を継続
LD      WORK0, AD_LOWH   ;結果が0であればWORK0, WORK1に検索結果を設定し、ルーチンを終了
LD      WORK1, AD_LOWL
RET
    
```

SUMMARY '%', ''

- (2) 中間値をD/Aコンバータに設定
 中間値を得るために上限値から下限値を減算し、その結果1ビット右にシフトし、さらにその結果を下限値に加えています。

```

%%
NEXT_AD:
CLR1   CY              ;減算結果を1ビット右にシフト(+2)
    
```



```

RORC  DBF1
RORC  DBF0
ADD   DBF0, AD_LOWL           ;右にシフトした値を下限值に加算し中間値を算出
ADDC  DBF1, AD_LOWH          ;D/Aコンバータに中間値を設定
PUT   ADCR, DBF

SUMMARY '%', ''
(3) 比較
      D/A値に中間値を設定した後に、中間値のアナログ値と入力アナログ値を比較し、結果をADCCMPに設定するまでにシステム・クロックがメイン・クロックとなっている場合には4ステップ分だけ待ち時間が必要です。
.EJ-1
%%
NOP
NOP                                     ;比較結果がADCCMPに設定されるまでウェイト(4ステップ)
NOP
NOP
NOP
NOP
SKT1  ADCCMP                       ;比較結果の判定
BR    AD_LOWER                     ;中間値 > 入力アナログ値

SUMMARY '%', ''
(4) 中間値 ≤ 入力アナログ値
      ここでは、通常は、中間値を下限值に設定しますが、通常でない場合が考えられます。
      ・中間値 = 下限値のとき
        このときには、上限値と下限値との差が1のときなので、上限値をD/Aコンバータに設定して、そのときの結果から判定します。
      ・上限値 ≤ 入力アナログ値のときは、上限値を下限值に設定します。
      ・上限値 > 入力アナログ値のときは、下限値を上限値に設定します。
%%
DI
SET2  CMP, Z                       ;中間値と下限値を比較
SUB   DBF1, AD_LOWH
SUBC  DBF0, AD_LOWL
SKT1  Z
BR    AD_LOW                       ;中間値 ≠ 下限値
EI                                         ;中間値 = 下限値
LD    DBF1, AD_UPH                 ;上限値をD/Aコンバータに設定
LD    DBF0, AD_UPL
PUT   ADCR, DBF
NOP
NOP                                     ;結果待ち
NOP
NOP
NOP
NOP
SKP1  ADCCMP
BR    AD_LOW                       ;上限値 ≤ 入力アナログ値
LD    DBF1, AD_LOWH               ;上限値 > 入力アナログ値
LD    DBF0, AD_LOWL               ;下限値を上限値に設定
ST    AD_UPH, DBF1
ST    AD_UPL, DBF0
BR    AD_LOOP
;
AD_LOWER:
EI
ST    AD_LOWH, DBF1               ;上限値を下限值に設定
ST    AD_LOWL, DBF0
BR    AD_LOOP
;

SUMMARY '%', ''
(5) 中間値 > 入力アナログ値
      このときには、中間値を上限値に設定します。
.EJ-1
%%
AD_LOWER:
ST    AD_UPH, DBF1               ;中間値を上限値に設定

```

```

ST      AD_DPL,DBF0
BR      AD_LOOP
;
EJECT
SUMMARY '%%','
    2. 表示用 A/D 値決定ルーチン'
本ルーチンでは、入力したアナログ値にあった表示を行うための表示用データ決定ルーチンです。
ここでは、A/D 値検索ルーチンより得た A/D 値を 1 秒ごとに 3 回読み込み、3 回読み込んだときの中間値を表示用データとしてい
ます。
そのときの最終データは、AD_NOWH,AD_NOWL のメモリに設定されます。
%%
;.....
;      A/D 値判断
;.....
AD_CHECK:

SUMMARY '%%','
    A/D 値検索ルーチンの呼び出し
    ここでは、A/D 値検索ルーチンを呼び出すか否かを AD_CHK フラグの値により判断します。AD_CHK フラグは、時計用タイマにより
    1 秒ごとにセットされます。
.EJ-1
%%
SKT1    AD_CHK          ; A/D 値検索ルーチンの呼び出し判断
RET     ; 検索しない
CLR1    AD_CHK          ; 検索時は、AD_CHK フラグをクリア
ADD     AD_CNT,#1       ; 検索回数をインクリメント
CALL   AD_COMP         ; A/D 値検索ルーチンの呼び出し

SUMMARY '%%','
    A/D 値を 1 回検索したとき
    1 回目のときは、AD_RD1H,AD_RD1L のメモリに、A/D 値を設定します。
%%
SKE     AD_CNT,#1       ; 1 回目を判定
BR      AD_CHK_2
ST      AD_RD1H,WORK0   ; A/D 値を設定
ST      AD_RD1L,WORK1
RET
;

SUMMARY '%%','
    A/D 値を 2 回検索したとき
    2 回目のときは、AD_RD2H,AD_RD2L のメモリに、A/D 値を設定します。
.EJ-1
%%
AD_CHK_2:
SKE     AD_CNT,#2       ; 2 回目を判定
BR      AD_CHK_3
ST      AD_RD2H,WORK0   ; A/D 値を設定
ST      AD_RD2L,WORK1
RBT
;

SUMMARY '%%','
    A/D 値を 3 回検索したとき
    3 回目のときは、AD_RD3H,AD_RD3L のメモリに、A/D 値を設定します。
次に、1 回目、2 回目、3 回目の A/D 値をソート用メモリに設定します。
ここで、それぞれのデータを昇順にソートし、中間値を表示用データとします。
また、次からは、1 秒ごとに最終データの設定を行うように、A/D カウンタの値を 2 回目に設定します。
%%
AD_CHK_3:
ST      AD_RD3H,WORK0   ; 3 回目の A/D 値を設定
ST      AD_RD3L,WORK1
MOV     AD_CNT,#2       ; 検索回数を 2 回目に設定
ST      AD_3H,WORK0     ; 3 番目のソート用メモリに 3 回目の A/D 値を設定
ST      AD_3L,WORK1
LD      WORK0,AD_RD1H   ; 1 番目のソート用メモリに 1 回目の A/D 値を設定
LD      WORK1,AD_RD1L

```

```

ST      AD_1H, WORK0
ST      AD_1L, WORK1
LD      WORK0, AD_RD2H      ; 2番目のソート用メモリに2回目のA/D値を設定
LD      WORK1, AD_RD2L
ST      AD_2H, WORK0
ST      AD_2L, WORK1

AD_CG:  CLR1  AD_CHANG      ; ソート用フラグをソート終了状態に設定
        MOV  IXL, #2      ; インデックスを1番目に設定
        MOV  IXL, #4

AD_CG_LP:
EI
CLR1    IXE
SKNS    IXL, #8          ; 比較終了判定
BR      CG_END
SET1    IXE
LD      WORK0, WORK0     ; IX番地の値をWORK0, WORK1に設定
LD      WORK1, WORK1
INC     IX               ; IXを次の番地に設定
INC     IX
DI
SET2    CMP, Z          ; IX番地の値と(IX-2)番地の値を比較
SUB     WORK0, WORK0
SUBC    WORK1, WORK1
SKT1    Z
SKT1    CY
BR      AD_CG_LP        ; (IX-2)番地の値 ≤ IX番地の値

```

: 比較した値同士を交換
: -----

```

EI
LD      WORK2, WORK0     ; IX番地の値をWORK2, WORK3に設定
LD      WORK3, WORK1
ST      WORK0, WORK0     ; (IX-2)番地の値をIX番地に設定
ST      WORK1, WORK1
CLR1    IXE
SET1    AD_CHANG        ; ソート終了フラグをセット(終了しない状態)
SUB     IXL, #2         ; IXを前の番地に設定
SET1    IXE
ST      WORK0, WORK2     ; (IX+2)番地の値をIX番地に設定
ST      WORK1, WORK3
INC     IX               ; IXを次の番地に設定
INC     IX
BR      AD_CG_LP

```

SUMMARY '%s', ''

・ 比較を終了

比較終了時に、比較をしていたときに値の入れ替えがあったかをチェックします。もし、入れ替えがあった場合には、再び比較を行います。
また、次の入力のために、A/D値の移動を行っています。

%%

```

CG_END: SKF1  AD_CHANG      ; 入れ替えがあったか判定
        BR   AD_CG        ; 再び比較
        LD  WORK0, AD_2H   ; 中間値を最終データに設定
        LD  WORK1, AD_2L
        ST  AD_NOWH, WORK0
        ST  AD_NOWL, WORK1
        LD  WORK0, AD_RD2H ; 2回目のA/D値を1回目に設定
        LD  WORK1, AD_RD2L
        ST  AD_RD1H, WORK0
        ST  AD_RD1L, WORK1
        LD  WORK0, AD_RD3H ; 3回目のA/D値を2回目に設定
        LD  WORK1, AD_RD3L

```

```
ST   AD_RD2H, WORK0  
ST   AD_RD2L, WORK1  
SETI AD_OK  
RET  
;  
END
```

: A / D 判定終了

クロスレファレンス



AS17K Z1.10 D1 << E17201A XREF LIST >> 11:02:58 02/25/91 PAGE 02-001

PROG -

SOURCE - AD.ASM

SYMBOL	TYPE	A	VALUE	/REP	(#DEP)																
AD_1H	MEM	L	0.24	/#	67	.	265														
AD_1L	MEM	L	0.25	/#	68	.	266														
AD_2H	MEM	L	0.26	/#	69	.	269	.	321												
AD_2L	MEM	L	0.27	/#	70	.	270	.	322												
AD_3H	MEM	L	0.28	/#	71	.	261														
AD_3L	MEM	L	0.29	/#	72	.	262														
AD_CG	LAB	L	59	/#	271	.	320														
AD_CG_LP	LAB	L	5C	/#	275	.	291	.	309												
AD_CHANG	FLG	L	0.2A.2	/#	77	.	272	.	272-1	.	272-1	.	302	.	302-1	.	302-1	.	319	.	319-1
AD_CHECK	LAB	P	3D	/	10	.	# 212														
AD_CHK	FLG	P	0.2A.1	/	16	.	# 76	.	220	.	220-1	.	220-1	.	222	.	222-1	.	222-1		
AD_CHK_2	LAB	L	47	/	231	.	# 242														
AD_CHK_3	LAB	L	4C	/	244	.	# 257														
AD_CNT	MEM	L	0.13	/#	53	.	223	.	230	.	243	.	260								
AD_COMP	LAB	L	5	/#	83	.	224	.	76	.	77										
AD_FLAG	MEM	L	0.2A	/#	74	.	75	.	77	.	200										
AD_LOOP	LAB	L	9	/#	109	.	183	.	189	.	200										
AD_LOF	LAB	L	36	/	168	.	178	.	# 185												
AD_LOWER	LAB	L	3A	/	150	.	# 197														
AD_LOWH	MEM	L	0.1C	/#	82	.	95	.	118	.	122	.	135	.	165	.	179	.	187		
AD_LOWL	MEM	L	0.1D	/#	83	.	98	.	117	.	123	.	135	.	166	.	180	.	188		
AD_NOWH	MEM	P	0.1E	/	13	.	# 64	.	323												
AD_NOWL	MEM	P	0.1F	/	13	.	# 65	.	324												
AD_OK	FLG	P	0.2A.0	/	16	.	# 75	.	333	.	333-1	.	333-1								
AD_RD1H	MEM	L	0.14	/#	54	.	232	.	283	.	327	.	328	.	331	.	332	.	332		
AD_RD1L	MEM	L	0.15	/#	55	.	233	.	284	.	328	.	325	.	331	.	332	.	332		
AD_RD2H	MEM	L	0.16	/#	56	.	245	.	267	.	325	.	331	.	332	.	332	.	332		
AD_RD2L	MEM	L	0.17	/#	57	.	246	.	268	.	326	.	332	.	332	.	332	.	332		
AD_RD3H	MEM	L	0.18	/#	58	.	258	.	329	.	330	.	330	.	330	.	330	.	330		
AD_RD3L	MEM	L	0.19	/#	59	.	259	.	330	.	330	.	330	.	330	.	330	.	330		
AD_UPH	MEM	L	0.1A	/#	60	.	93	.	115	.	170	.	181	.	198	.	199	.	199		
AD_UPL	MEM	L	0.1B	/#	61	.	94	.	116	.	171	.	182	.	199	.	199	.	199		
CG_END	LAB	L	7A	/	279	.	# 318														
NEXT_AD	LAB	L	13	/	121	.	# 131														
WORK0	MEM	L	0.00	/#	48	.	122	.	232	.	245	.	258	.	261	.	283	.	265	.	287
						.	259	.	281	.	281	.	287	.	287	.	297	.	299	.	305
						.	321	.	323	.	325	.	327	.	329	.	331	.	331	.	331
WORK1	MEM	L	0.01	/#	49	.	123	.	233	.	246	.	259	.	262	.	264	.	266	.	268
						.	270	.	282	.	282	.	288	.	288	.	298	.	300	.	306
						.	322	.	324	.	326	.	328	.	330	.	332	.	332	.	332
WORK2	MEM	L	0.02	/#	50	.	297	.	305	.	305	.	305	.	305	.	305	.	305	.	305
WORK3	MEM	L	0.03	/#	51	.	298	.	306	.	306	.	306	.	306	.	306	.	306	.	306

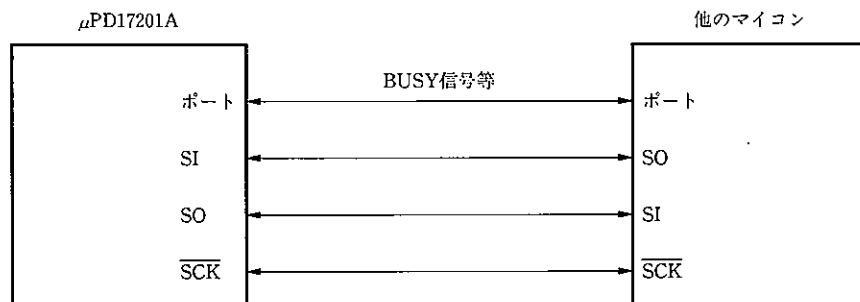
TOTAL SYMBOLS - 38

END OF XREF LIST

第7章 シリアルI/O

μ PD17201Aは、3線式シリアル・インタフェースを内蔵しております。

シリアル通信には、シリアル・クロック (SCK)、シリアル入力 (SI)シリアル出力 (SO) を使用し、スレーブからマスタへの現在の状態を知らせる信号としてBUSYを使用します。以下にシステム構成を示します。

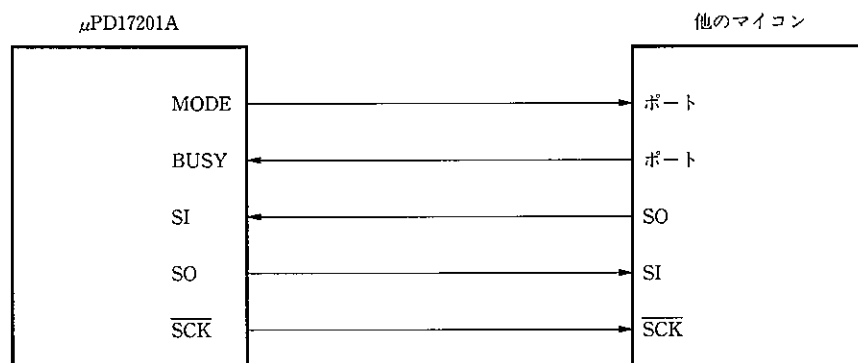


シリアル・クロックについては、マスタ・マイコンかスレーブ・マイコンどちらかが管理することになります。

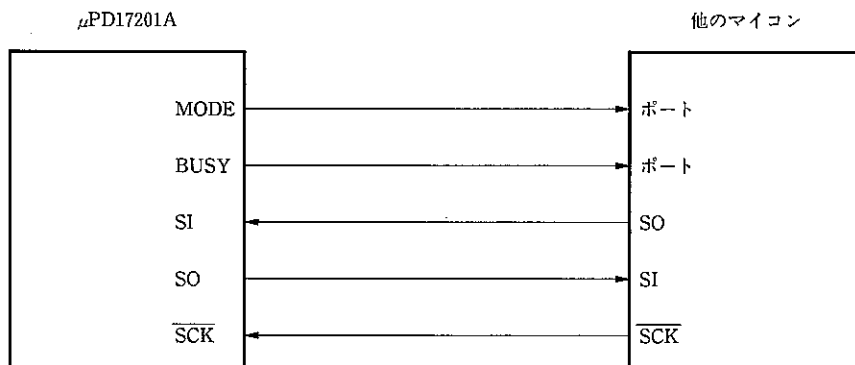
7.1 プログラム説明

プログラム例では、 μ PD17201AよりのMODE出力レベルにより、他のマイコンより、データの受信や送信を行います。送受信時のシステム構成を以下に示します。

送信時



受信時



送受信時のちがいは、送信時は、BUSY端子が入力、 $\overline{\text{SCK}}$ 端子は出力となり、受信時は、BUSY端子が出力、 $\overline{\text{SCK}}$ 端子は入力となります。

モード	BUSY端子	$\overline{\text{SCK}}$ 端子
送信	入力	出力
受信	出力	入力

$\mu\text{PD17201A}$ で設定可能なシリアル・クロックは、250 kHz, 312.5 kHz, 3.9 kHz (システム・クロック : 4 MHz時) です。

データ入力時に入力可能なシリアル・クロックは最大500 kHzで、データ出力時には、最大100 kHzとなっています。(V_{DD}=4.5~5.5 V時)

<使用するハードウェア>

番号	端子名	使用説明						
57	P0D ₃	<p>μPD17201Aより他のマイコンへ、シリアル通信の通信モードを知らせるための出力端子です。</p> <table border="1"> <thead> <tr> <th>出力レベル</th> <th>通信モード</th> </tr> </thead> <tbody> <tr> <td>ハイ</td> <td>受信モード</td> </tr> <tr> <td>ロウ</td> <td>送信モード</td> </tr> </tbody> </table>	出力レベル	通信モード	ハイ	受信モード	ロウ	送信モード
出力レベル	通信モード							
ハイ	受信モード							
ロウ	送信モード							
56	P0D ₂	<p>BUSY信号入出力用の端子です。 通信モードにより、以下のように動作が異なります。</p> <table border="1"> <thead> <tr> <th>通信モード</th> <th>B U S Y 動 作</th> </tr> </thead> <tbody> <tr> <td>受信モード</td> <td>BUSY出力用端子です。 受信可能状態となったときに、BUSY出力をハイ・レベルからロウ・レベルにします。</td> </tr> <tr> <td>送信モード</td> <td>BUSY入力用端子です。 BUSY入力がハイ・レベルからロウ・レベルになった後、データ送信を開始します。</td> </tr> </tbody> </table> <p>受信を行うとき以外は、入力端子となっています。</p>	通信モード	B U S Y 動 作	受信モード	BUSY出力用端子です。 受信可能状態となったときに、BUSY出力をハイ・レベルからロウ・レベルにします。	送信モード	BUSY入力用端子です。 BUSY入力がハイ・レベルからロウ・レベルになった後、データ送信を開始します。
通信モード	B U S Y 動 作							
受信モード	BUSY出力用端子です。 受信可能状態となったときに、BUSY出力をハイ・レベルからロウ・レベルにします。							
送信モード	BUSY入力用端子です。 BUSY入力がハイ・レベルからロウ・レベルになった後、データ送信を開始します。							

・シリアル・インタフェース

通信モード	使用説明
受信モード	<p>SI端子 : 入力 SO端子 : ハイ・インピーダンス SCK端子 : 外部クロック入力</p>
送信モード	<p>SI端子 : 入力 SO端子 : 出力 SCK端子 : 3.9 kHz出力</p>

・タイマ：時計用タイマ

通常時のタイム・アウト時間 (500 ms) の測定。



ドキュメント



AS17K Z1.10 D1 << E17201A DOCUMENT >> 19:39:45 02/26/91

TABLE OF CONTENTS

	PAGE
第 7 章 シリアル I / O	1
・ シリアル送信	2
・ シリアル受信	5
・ 割り込みルーチン	7

第 7 章 シリアル I/O

```

PUBLIC (MEM) : I_DATA_H, I_DATA_L, O_DATA_H, O_DATA_L
PUBLIC (FLG) : BUSY, MODE, SI_OK, SIO_ERR, SO_OK
PUBLIC (LAB) : CLK_INT, EI_RET, SI_PRO, SIO_INT, SO_PRO
ADDR RANGE : 0005H - 0075H

```

<プログラム説明>

このプログラムは、シリアル・インタフェースを使って、シリアル通信を行うプログラムです。
このプログラムでは、シリアル・インタフェース用の端子(SI, SO, SCK)の他に、BUSY状態入出力用の端子を使用しています。

端子説明

・BUSY端子

この端子は、データ送信側(メイン)がデータ受信側(スレーブ)に対し、データの送信開始を確認するための端子です。
通常は、ハイ・レベルとなっており、データの送信を開始するとき、受信側(スレーブ)が端子をロウ・レベルに設定し
送信側(メイン)は、ロウ・レベルになったことを確認して、データの送信を行います。
このプログラムでは、データの送信時に、BUSY端子の状態を確認していますが、状態の確認開始から1秒以内にBUSY端子が
ロウ・レベルにならなければ、通信エラーとしています。

・SCK端子

この端子は、通信時のシリアル・クロックを出力するための端子です。
この端子の入出力状態は、送信時と受信時で異なり、送信時はクロックを出力し、受信時はクロックを入力します。
送信時のクロックは、3.9kHzとなっています。

割り込み

このプログラムでは、割り込みとして、時計用タイマ割り込みとシリアル割り込みを使用しています。

・時計用タイマ割り込み

ここでは、シリアル通信時の通信異常検出用としてのタイム・アウト・カウント用として使用しています。
割り込みが発生すると、COUNTをインクリメントします。

・シリアル割り込み

ここでは、シリアル通信時の8ビット通信の完了検出用として使用しています。
割り込みが発生すると、SIO_ENDをセットします。

通信異常

シリアル通信時に、発生する異常には以下の点が考えられます。

1. BUSY状態からBUSY解除状態にならない。
2. シリアル・クロック入力時に、クロックが入力されない。

上記の状態について、このプログラムでは簡単な対策を行っています。

・対策例

1. BUSY端子チェック時に、チェック時間を設けています。ここでは、1秒間の間だけチェックを行い、1秒経過してもBUSY状態が解除されなければ、通信に異常が発生したと判断し通信エラーとしています。
2. シリアル・クロック入力時に、クロック入力時間を設定しています。ここでは、0.5秒間を設定し、0.5秒以内に通信を終了しなければ、通信異常と判断し通信エラーとしています。

```

ENTRANCES          :-
MEMORIES CHANGED  :-
MEMORIES REFERRED :-
MEMORIES MANIPULATED :-
FLAGS CHANGED     :-
FLAGS REFERRED    :-
DATA REFERRED     :-
BRANCH TO         :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL       :-

```


・シリアル送信

シリアル送信を行うためのルーチンです。
 ここでは、スレーブ側のBUSY状態をチェックし、BUSYが解除されたら、送信用データを設定し、シリアル・データをレシーブ側に送信します。
 このとき、通信中に異常が発生したときには、SIO_ERRフラグをセットしてこのルーチンを抜けます。異常もなしに送信を終了したときには、SO_OKフラグをセットしてこのルーチンを抜けます。

```

ENTRANCES          :SO_PRO
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :SI_OK          SIO_END
                   :SIO_ERR        SO_OK
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

(1) BUSY状態のチェック

スレーブ側の状態を1秒間チェックします。
 1秒以内にBUSY状態が解除されれば、そのままデータの送信を行い、1秒経過してもBUSY状態が解除されなければ、通信エラーとします。

```

ENTRANCES          :-
MEMORIES CHANGED   :COUNT
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :SIO_ERR
FLAGS REFERRED      :BUSY
DATA REFERRED       :-
BRANCH TO          :S_OUT          ; 送信処理開始へ
                   :SIO_OFF        ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

(2) 送信準備

ここでは、シリアル・インタフェースを使用して、送信を行うための準備を行います。
 シリアル・インタフェースをシリアル送信時のモードを設定し、SI, SO, SCK端子をシリアル・モードに設定します。
 シリアル送信時のモード設定には、SIOHIZ, SIOCKO, SIOCKIを設定します。
 シリアル・モードの設定に、SIOENを設定します。
 送信データは、送信を開始する前に予め設定しておく必要があります

```

ENTRANCES          :S_OUT
MEMORIES CHANGED   :-
MEMORIES REFERRED  :O_DATA_H      O_DATA_L
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

(3) 送信開始

送信を開始する前に、タイム・アウト用のカウンタをクリアします。
 カウンタ・クリア後、送信を開始します。
 送信の開始は、SIOTSフラグをセットすることで可能です。

```

ENTRANCES          :-
MEMORIES CHANGED   :COUNT
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

(4) 送信終了判断

送信の終了判断は、SIO_ENDフラグをチェックします。
 SIO_ENDフラグは、シリアル送信にて8ビット・データを送信し終することで、シリアル割り込みが発生し、その割り込み処理内でセットされます。
 また、ここでは、タイム・アウト用のカウンタの値も同時にチェックし、0.5秒経過すると送信エラーとします。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :SIO_ERR
FLAGS REFERRED     :SIO_END
DATA REFERRED      :-
BRANCH TO          :SIO_OFF      ;
                   SO_END        ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

(5) 送信終了
送信終了時には、SO_OKフラグをセットします。

```
ENTRANCES          :SO_BND
MEMORIES CHANGED  :-
MEMORIES REFERRED :-
MEMORIES MANIPULATED :-
FLAGS CHANGED     :SO_OK
FLAGS REFERRED    :-
DATA REFERRED     :-
BRANCH TO         :SIO_OFF      ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL       :-
```

・シリアル受信

シリアル受信を行うためのルーチンです。
 ここでは、外部からのシリアル・クロックに同期して、スレーブ側からのデータを受信します。
 このとき、通信中に異常が発生したときには、SIO_ERRフラグをセットしてこのルーチンを抜けます。異常もなしに送信を終了したときには、SI_OKフラグをセットしてこのルーチンを抜けます。

```

ENTRANCES          :SI_PRO
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :SI_OK          SIO_END
                   :SIO_ERR        SO_OK
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO           :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
  
```

(1) 受信準備

ここでは、シリアル・インタフェースを使用して、受信を行うための準備を行います。
 シリアル・インタフェースをシリアル受信時のモードを設定し、SI, SO, SCK端子をシリアル・モードに設定します。
 シリアル受信時のモード設定には、SIOHIZ, SIOCK0, SIOCK1を設定します。
 シリアル・モードの設定には、SIOENを設定します。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO           :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
  
```

(2) 受信開始

受信を開始する前に、タイム・アウト用のカウンタをクリアします。
 カウンタ・クリア後、受信を開始します。
 受信の開始は、SLOTSフラグをセットすることで可能です。受信を開始したら、BUSY状態を解除します。

```

ENTRANCES          :-
MEMORIES CHANGED   :COUNT
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :BUSY
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO           :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-
  
```

(3) 受信終了判断

受信の終了判断は、SIO_ENDフラグをチェックします。
SIO_ENDフラグは、シリアル受信にて8ビット・データを受信し終了すると、シリアル割り込みが発生し、その割り込み処理内でセットされます。
また、ここでは、タイム・アウト用のカウンタの値も同時にチェックし、0.5秒経過すると受信エラーとします。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-SIO_ERR
FLAGS REFERRED      :-SIO_END
DATA REFERRED       :-
BRANCH TO          :-SIO_END      ;
                   :-SIO_OFF     ;
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-

```

(4) 受信終了

受信終了時には、SI_OKフラグをセットし、シフト・レジスタより、受信データを入力します。

```

ENTRANCES          :-SI_END
MEMORIES CHANGED   :-I_DATA_H    I_DATA_L
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-SI_OK
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-

```

(5) シリアル受信終了処理

シリアル受信終了時には、シリアル受信モードを解除します。

```

ENTRANCES          :-SIO_OFF
MEMORIES CHANGED   :-
MEMORIES REFERRED   :-
MEMORIES MANIPULATED :-
FLAGS CHANGED       :-BUSY      MODE
FLAGS REFERRED      :-
DATA REFERRED       :-
BRANCH TO          :-
SUBROUTINES CALLED  :-
LABELS MANIPULATED :-
SYSTEM CALL         :-

```

・割り込みルーチン
このプログラムにて使用した、割り込みルーチンは、シリアル割り込みと、時計用割り込みです。

- (1) シリアル割り込みルーチン
この割り込みは、シリアル通信時に8ビットのデータを送信または受信したときに発生します。
ここでは、通信の終了を示すために、SIO_ENDフラグをセットしています。

```

ENTRANCES          :SIO_INT
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :SIO_END
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :EI_RET1 ;
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

- (2) 時計用割り込み
この割り込みは、時計用タイマに同期して割り込みを発生します。割り込み発生周期は、250msです。
ここでは、250msごとにCOUNTをインクリメントしています。

```

ENTRANCES          :CLK_INT      EI_RET1
MEMORIES CHANGED   :COUNT
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-

```

RAMマップ

AS17K Z1.10 DI << E17201A DATA MEMORY MAP >> 19:39:19 02/26/91 PAGE 02-001

PROG =

SOURCE = SIO.ASM

(PUBLIC/LOCAL MAP)

BANK = 0

		ROW								
		7	6	5	4	3	2	1	0	
I			LCDD32	LCDD16	LCDD0					0
I			LCDD33	LCDD17	LCDD1					1
I			LCDD34	LCDD18	LCDD2				IO_DATA_H	2
I					IFLAG					3
I			LCDD35	LCDD19	LCDD3				IO_DATA_L	4
I					ICOUNT					5
I	IAR3		LCDD20	LCDD4					II_DATA_H	6
I	IAR2		LCDD21	LCDD5					II_DATA_L	7
I	IAR1		LCDD22	LCDD6						8
I	IAR0		LCDD23	LCDD7						9
I	IWR		LCDD24	LCDD8						A
I	IBANK		LCDD25	LCDD9						B
I	IIXH		LCDD26	LCDD10						C
I	IMPH									D
I	IIXM		LCDD27	LCDD11						E
I	IMPL									
I	IIXL		LCDD28	LCDD12					IDBP3	
I	IRPH		LCDD29	LCDD13					IDBP2	
I	IRPL		LCDD30	LCDD14					IDBP1	

AS17K Z1.10 D1 << B17201A DATA MEMORY MAP >> 19:39:19 02/26/01 PAGE 02-002

PROG =

SOURCE = SIO.ASM

(PUBLIC/LOCAL MAP) BANK = 0

ROW							
7	6	5	4	3	2	1	0
IPSW		LCDD31	LCDD15				DBP0

AS17K Z1.10 D1 << E17201A FLG MAP >> 19:39:19 02/26/91 PAGE 02-003

PROG -

SOURCE = S10.ASM

(FLAG MAP) BANK = 0

MSB	3	2	1	0	LSB
32	ISI_OK	ISO_OK	ISIO_ERR		
				ISIO_END	
70	IPOA3	IPOA2	IPOA1	IPOA0	
71	IPOB3	IPOB2	IPOB1	IPOB0	
72	IPOC3	IPOC2	IPOC1	IPOC0	
73	IMODE	IBUSY	IPOD1	IPOD0	
	IPOD3	IPOD2			
7A	IMPE				
7B				IBCD	
7F	ICMP	ICY	IZ	IXR	
82			ISYSCK	IXEN	
83	IWDRES	IWKMD	IWMRES		
87	IVDDDET1	IVDDDET0			
8F				IINT	
91				INRZBF	
92				INRZ	
A0				IADCCNP	
A1	IVRBFEN	IADCEN	IADCCH1	IADCCH0	
A2	ISIOTS	SIOSHIZ	SILOCK1	SILOCK0	

AS17K Z1.10 D1 << E17201A FLG MAP >> 19:39:19 02/26/91 PAGE 02-004

PROG =

SOURCE = SIO.ASM

(FLAG MAP) BANK = 0

MSB 3 2 1 0 LSB
+-----+-----+-----+-----+-----+

AS17K Z1.10 D1 << E17201A FLG MAP >> 19:39:19 02/26/91 PAGE 02-005

PROG -

SOURCE - SIO.ASM

(FLAG MAP) BANK = 0

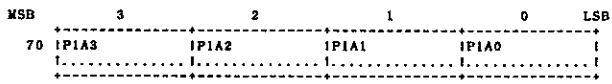
MSB	3	2	1	0	LSB
A3		NRZEN	TMOE	SIOEN	
A7	PODBIO3	PODBIO2	PODBIO1	PODBIO0	
AF	IPSIO	IPWTM	IP	IPTM	
B1	LCDBN	LCDCK2	LCDCK1	LCDCK0	
B2	LCDMD3	LCDMD2	LCDMD1	LCDMD0	
B3	TMEN	TMRES	TMCK1	TMCK0	
B7	PIAGIO	POCGIO	POBGIO	POAGIO	
BB	IRQSIO				
BC		IRQTM			
BD			IRQ		
BE				IRQTM	

AS17K Z1.10 D1 << E17201A FLG MAP >> 19:39:19 02/26/91 PAGE 02-006

PROG =

SOURCE = SIO.ASM

(FLAG MAP) BANK = 1



RAM使用説明

AS17K Z1.10 Di << E17201A DATA MEMORY MAP >> 19:39:19 02/26/91 PAGE 02-007

PROG -

SOURCE - SIO.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
ADCCH0	FLG	0	A1	...1	
ADCCH1	FLG	0	A1	...1	
ADCCMP	FLG	0	A0	...1	
ADCEN	FLG	0	A1	...1	
AR0	MEM	0	77		
AR1	MEM	0	76		
AR2	MEM	0	75		
AR3	MEM	0	74		
BANK	MEM	0	79		
BCD	FLG	0	7E	...1	
BUSY	FLG	0	73	...1	BUSY状態入出力端子 (1:BUSY状態)
CBP	FLG	0	7F	...1	
COUNT	MEM	0	33		タイム・アウト時間測定用メモリ (基準時間:250ms)
CY	FLG	0	7F	...1	
DBF0	MEM	0	F		
DBF1	MEM	0	E		
DBF2	MEM	0	D		
DBF3	MEM	0	C		
FLAG	MEM	0	32		シリアル通信状態用フラグ
I_DATA_H	MEM	0	4		シリアル受信データ
I_DATA_L	MEM	0	5		
INT	FLG	0	8P	...1	
IP	FLG	0	AF	...1	
IPS10	FLG	0	AF	...1	
IPTM	FLG	0	AF	...1	
IPWTM	FLG	0	AF	...1	
IRQ	FLG	0	BD	...1	
IRQS10	FLG	0	BB	...1	
IRQTM	FLG	0	BE	...1	
IRQWTM	FLG	0	BC	...1	
IXE	FLG	0	7F	...1	
IXH	MEM	0	7A		
IXL	MEM	0	7C		
IXM	MEM	0	7B		
LCDC0	FLG	0	B1	...1	
LCDC1	FLG	0	B1	...1	
LCDC2	FLG	0	B1	...1	
LCDD0	MEM	0	40		
LCDD1	MEM	0	41		
LCDD10	MEM	0	4A		
LCDD11	MEM	0	4B		
LCDD12	MEM	0	4C		
LCDD13	MEM	0	4D		
LCDD14	MEM	0	4E		
LCDD15	MEM	0	4F		
LCDD16	MEM	0	50		
LCDD17	MEM	0	51		
LCDD18	MEM	0	52		
LCDD19	MEM	0	53		
LCDD2	MEM	0	42		
LCDD20	MEM	0	54		
LCDD21	MEM	0	55		

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 19:39:19 02/26/91 PAGE 02-008

PROG -

SOURCE = SIO.ASM

(DATA AREA INFORMATION)

SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
LCDD22	MEM	0	56		
LCDD23	MEM	0	57		
LCDD24	MEM	0	58		
LCDD25	MEM	0	59		
LCDD26	MEM	0	5A		
LCDD27	MEM	0	5B		
LCDD28	MEM	0	5C		
LCDD29	MEM	0	5D		
LCDD3	MEM	0	43		
LCDD30	MEM	0	5E		
LCDD31	MEM	0	5F		
LCDD32	MEM	0	60		
LCDD33	MEM	0	61		
LCDD34	MEM	0	62		
LCDD35	MEM	0	63		
LCDD4	MEM	0	44		
LCDD5	MEM	0	45		
LCDD6	MEM	0	46		
LCDD7	MEM	0	47		
LCDD8	MEM	0	48		
LCDD9	MEM	0	49		
LCDEN	FLG	0	B1	1...	
LCDM0	FLG	0	B2	...1	
LCDM1	FLG	0	B2	...1	
LCDM2	FLG	0	B2	...1	
LCDM3	FLG	0	B2	1...	
MODE	FLG	0	73	1...	シリアル・モード設定 (1:送信モード 0:受信モード)
MPE	FLG	0	7A	1...	
MPH	MEM	0	7A		
MPL	MEM	0	7B		
NRZ	FLG	0	92	...1	
NRZBF	FLG	0	91	...1	
NRZEN	FLG	0	A3	...1	
O_DATA_H	MEM	0	2		シリアル送信データ
O_DATA_L	MEM	0	3		
POA0	FLG	0	70	...1	
POA1	FLG	0	70	...1	
POA2	FLG	0	70	...1	
POA3	FLG	0	70	1...	
POAGIO	FLG	0	B7	...1	
POB0	FLG	0	71	...1	
POB1	FLG	0	71	...1	
POB2	FLG	0	71	...1	
POB3	FLG	0	71	1...	
POBGIO	FLG	0	B7	...1	
POC0	FLG	0	72	...1	
POC1	FLG	0	72	...1	
POC2	FLG	0	72	...1	
POC3	FLG	0	72	1...	
POCGIO	FLG	0	B7	...1	
POD0	FLG	0	73	...1	
POD1	FLG	0	73	...1	

AS17K Z1.10 D1 << E17201A DATA MEMORY MAP >> 19:39:19 02/26/91 PAGE 02-009

PROG -

SOURCE - SIO.ASM

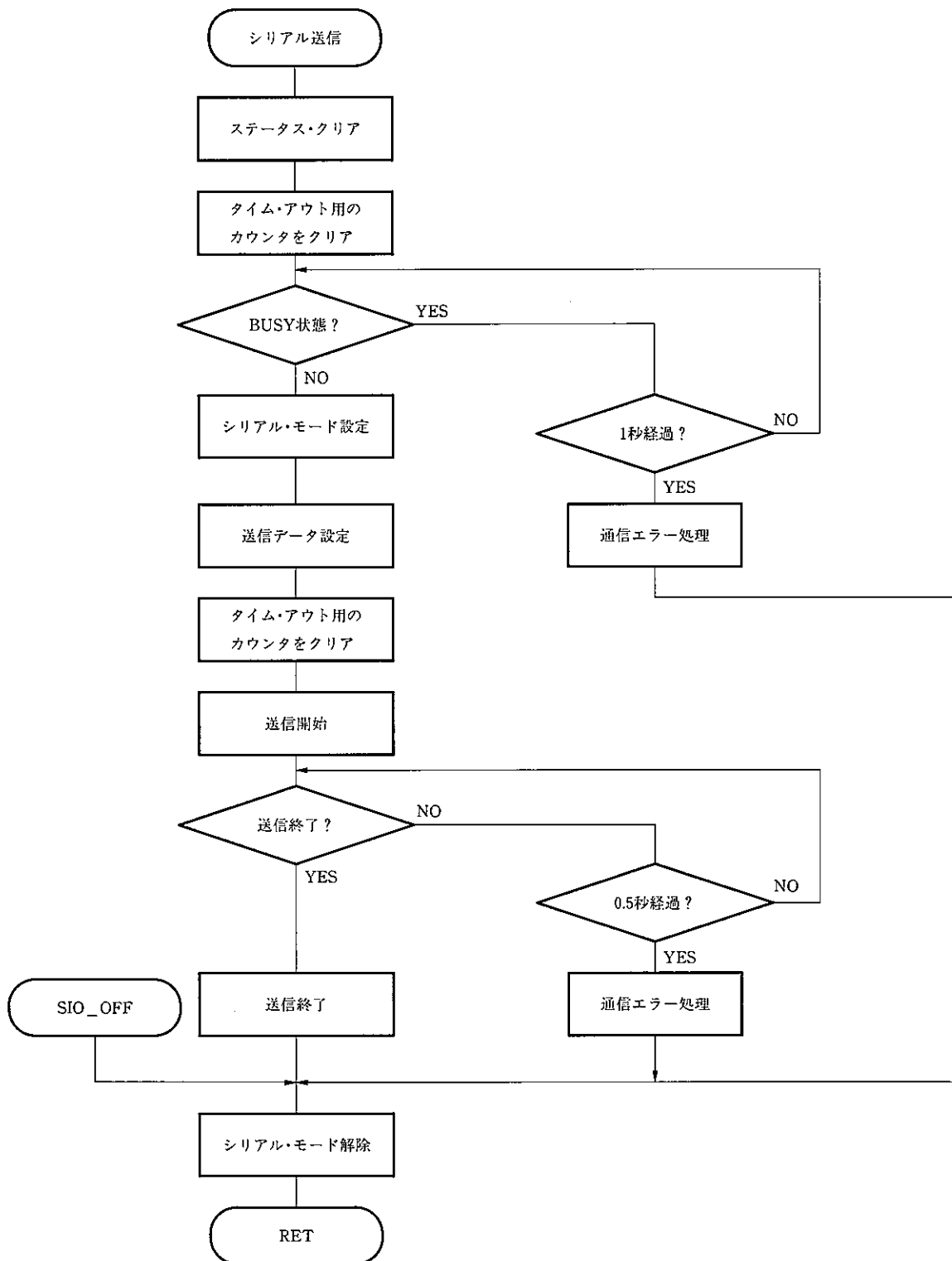
(DATA AREA INFORMATION)

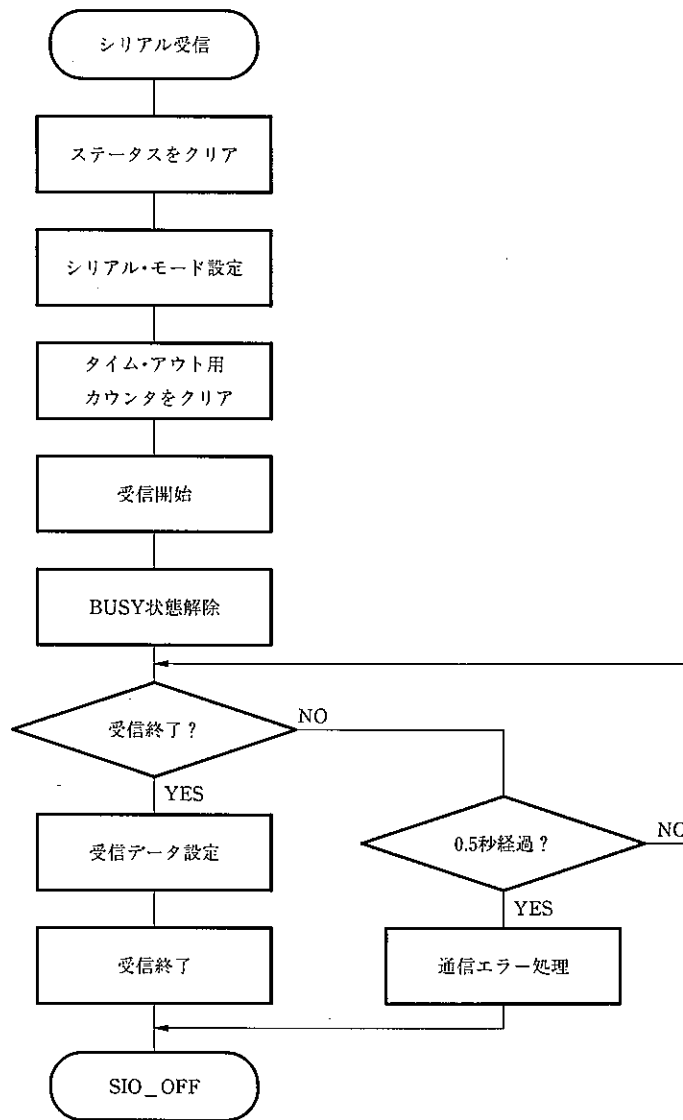
SYMBOL	TYPE	BNK	LOC	BIT	INFORMATION
POD2	FLG	0	73	.1..	
POD3	FLG	0	73	1..	
PODB100	FLG	0	A7	...1	
PODB101	FLG	0	A7	...1	
PODB102	FLG	0	A7	...1	
PODB103	FLG	0	A7	1..	
PIA0	FLG	1	70	...1	
PIA1	FLG	1	70	...1	
PIA2	FLG	1	70	...1	
PIA3	FLG	1	70	1..	
PIAG10	FLG	0	B7	1..	
PSW	MEM	0	7F		
RPH	MEM	0	7D		
RPL	MEM	0	7E		
SI_OK	FLG	0	32	1..	シリアル受信終了フラグ (1:終了)
SIO_END	FLG	0	32	...1	8ビット・データ通信終了フラグ (1:終了)
SIO_ERR	FLG	0	32	...1	シリアル通信異常終了フラグ (1:異常終了)
SIOCK0	FLG	0	A2	...1	
SIOCK1	FLG	0	A2	...1	
SIOEN	FLG	0	A3	...1	
SIOHIZ	FLG	0	A2	...1	
SLOTS	FLG	0	A2	1..	
SO_OK	FLG	0	32	...1	シリアル送信終了フラグ (1:終了)
SYSCK	FLG	0	82	...1	
TMCK0	FLG	0	B3	...1	
TMCK1	FLG	0	B3	...1	
TMEN	FLG	0	B3	1..	
TMOS	FLG	0	A3	...1	
TMRES	FLG	0	B3	...1	
VDDDET0	FLG	0	87	...1	
VDDDET1	FLG	0	87	...1	
VREFEN	FLG	0	A1	...1	
WDTRES	FLG	0	83	...1	
WR	MEM	0	78		
WTMMD	FLG	0	83	...1	
WTMRES	FLG	0	83	...1	
XEN	FLG	0	82	...1	
Z	FLG	0	7F	...1	

フロー・チャート



フロー・チャート





リスト



```

:*****
:*** シリアル送受信プログラム
:*****
SUMMARY '%s', '第 7 章 シリアルI/O'
%
```

```

SUMMARY '%s', '
<プログラム説明>'

```

このプログラムは、シリアル・インタフェースを使って、シリアル通信を行うプログラムです。
このプログラムでは、シリアル・インタフェース用の端子(SI, SO, SCK)の他に、BUSY状態入出力用の端子を使用しています。

端子説明

・BUSY端子

この端子は、データ送信側(メイン)がデータ受信側(スレーブ)に対し、データの送信開始を確認するための端子です。送信側は、ハイ・レベルとなっており、データを送信を開始すると、受信側(スレーブ)が端子をロー・レベルに設定し送信側(メイン)は、ロー・レベルになったことを確認して、データの送信を行います。このプログラムでは、データの送信時に、BUSY端子の状態を確認していますが、状態の確認開始から1秒以内にBUSY端子がロー・レベルにならなければ、通信エラーとしています。

・SCK端子

この端子は、通信時のシリアル・クロックを入出力するための端子です。この端子の入出力状態は、送信時と受信時で異なり、送信時はクロックを出力し、受信時はクロックを入力します。送信時のクロックは、3.0kHzと なっています。

割り込み

このプログラムでは、割り込みとして、時計用タイマ割り込みとシリアル割り込みを使用しています。

- ・時計用タイマ割り込み
ここでは、シリアル通信時の通信異常検出用としてのタイム・アウト・カウント用として使用しています。割り込みが発生すると、COUNTをインクリメントします。
- ・シリアル割り込み
ここでは、シリアル通信時の8ビット通信の完了検出用として使用しています。割り込みが発生すると、SIO_ENDをセットします。

通信異常

シリアル通信時に、発生する異常には以下の点が考えられます。

1. BUSY状態からBUSY解除状態にならない。
2. シリアル・クロック入力時に、クロックが入力されない。

上記の状態について、このプログラムでは簡単な対策を行っています。

・対策例

1. BUSY端子チェック時に、チェック時間を設けています。ここでは、1秒間の間だけチェックを行い、1秒経過してもBUSY状態が解除されなければ、通信に異常が発生したと判断し通信エラーとしています。
2. シリアル・クロック入力時に、クロック入力時間を設定しています。ここでは、0.5秒間を設定し、0.5秒以内に通信を終了しなければ、通信異常と判断し通信エラーとしています。

```

.EJ-1
%
-----
PUBLIC宣言
-----
:--- LABEL ---
PUBLIC SO_PRO, SI_PRO
PUBLIC SIO_INT, CLK_INT
PUBLIC EI_RET1

:--- FLAG ---
PUBLIC SO_OK, SI_OK, SIO_ERR
PUBLIC MODE, BUSY

:--- MEMORY ---
PUBLIC O_DATA_H, O_DATA_L, I_DATA_H, I_DATA_L

:*****
: シリアル通信ルーチン
:*****
:////////////////////
: メモリ定義
:
```

```

//////////
O_DATA_H      MEM      0.02H      : シリアル送信データ
O_DATA_L      MEM      0.03H
I_DATA_H      MEM      0.04H      : シリアル受信データ
I_DATA_L      MEM      0.06H

FLAG          MEM      0.32H      : シリアル通信状態用フラグ
SIO_ERR       PLG      FLAG.0     : 8ビット・データ通信終了フラグ (1:終了)
SIO_ERR       PLG      FLAG.1     : シリアル通信異常終了フラグ (1:異常終了)
SO_OK         PLG      FLAG.2     : シリアル送信終了フラグ (1:終了)
SI_OK         PLG      FLAG.3     : シリアル受信終了フラグ (1:終了)

COUNT        MEM      0.33H      : タイム・アウト時間測定用メモリ (基準時間:250ms)

BUSY          PLG      POD2       : BUSY状態入出力端子 (1:BUSY状態)
MODE          PLG      POD3       : シリアル・モード設定 (1:送信モード 0:受信モード)
    
```

SUMMARY '%%', ' : シリアル送信'
 シリアル送信を行うためのルーチンです。
 ここでは、スレーブ側のBUSY状態をチェックし、BUSYが解除されたら、送信用データを設定し、シリアル・データをレシーブ側に送信します。
 このとき、通信中に異常が発生したときには、SIO_ERRフラグをセットしてこのルーチンを抜けます。異常もなしに送信を終了したときには、SO_OKフラグをセットしてこのルーチンを抜けます。

```

%%
:.....
: シリアル送信ルーチン
:.....
SO_PRO:
:-----
: ステータス・クリア
:-----
CLR4  SIO_BND,SIO_ERR,SO_OK,SI_OK  : 各種状態フラグを初期化
    
```

SUMMARY '%%', ' (1) BUSY状態のチェック'
 スレーブ側の状態を1秒間チェックします。
 1秒以内にBUSY状態が解除されれば、そのままデータの送信を行い、1秒経過してもBUSY状態が解除されなければ、通信エラーとします。

```

.EJ-1
%%
:-----
: レシーブ側の状態チェック
:-----
MOV    COUNT,#0          : タイム・アウト用カウンタ・クリア
SET1   WTMRES            : 時計用タイマ・リセット
CLR1   IRQWTM            : 時計用タイマの割り込み要求クリア
SET1   IPWTM             : 時計用タイマの割り込み許可
EI      : 割り込み許可

BUSY_CHK:
SET1   WDIRBS            : ウォッチ・ドッグ・タイマ・リセット
SKT1   BUSY              : BUSY状態検出
BR     S_OUT              : 送信処理開始へ
SKGB   COUNT,#4         : 1秒チェック
BR     BUSY_CHK           : BUSY状態をチェック

:-----
: タイム・アウト
:-----
SET1   SIO_ERR           : 1秒タイム・アウト
BR     SIO_OFF
    
```

SUMMARY '%%', ' (2) 送信準備'
 ここでは、シリアル・インタフェースを使用して、送信を行うための準備を行います。
 シリアル・インタフェースをシリアル送信時のモードを設定し、SI,SO,SCK端子をシリアル・モードに設定します。
 シリアル送信時のモード設定には、SIOHIZ,SIOCK0,SIOCK1を設定します。
 シリアル・モードの設定には、SIOENを設定します。
 送信データは、送信を開始する前に予め設定しておく必要があります。

```

%%
:-----
: 送信処理
:-----
S_OUT:
: DI
:-----
: シリアル送信状態に設定
:-----
INITPLG NOT SIOTS, NOT SIOHIZ, SIOCK1, SIOCK0 ; シリアル・モードを送信モードに設定
; SO端子=シリアル・アウト状態
; SCK端子=3.9kHz
SET1 SIOEN ; シリアル送信状態設定
SET1 IPSIO ; シリアル送信時の割り込み許可
:-----
: 送信データを設定
:-----
LD DBF1, O_DATA_H ; 送信データ設定
LD DBF0, O_DATA_L
PUT SIOSPR, DBF

```

SUMMARY '%%', ''
 (3) 送信開始
 送信を開始する前に、タイム・アウト用のカウンタをクリアします。
 カウンタ・クリア後、送信を開始します。
 送信の開始は、SIOTSフラグをセットすることで可能です。

```

%%
:-----
: タイム・アウト時間の設定
:-----
MOV COUNT, #0 ; タイム・アウト用カウンタ・クリア
SET1 WTMRES ; 時計用タイマ・リセット
CLR1 IRQWTM ; 時計用タイマの割り込み要求クリア
SET1 IPWTM ; 時計用タイマの割り込み許可
:-----
: 送信スタート
:-----
SET1 SIOTS ; シリアル送信開始
EI

```

SUMMARY '%%', ''
 (4) 送信終了判断
 送信の終了判断は、SIO_ENDフラグをチェックします。
 SIO_ENDフラグは、シリアル通信にて8ビット・データを送信し終することで、シリアル割り込みが発生し、その割り込み処
 理内でセットされます。
 また、ここでは、タイム・アウト用のカウンタの値も同時にチェックし、0.5秒経過すると通信エラーとします。

```

.EJ-1
%%
:-----
: シリアル送信中
:-----
LOOP_SO:
SET1 WDTRES ; 8ビット送信終了判断
SKF1 SIO_END ; 500msタイム・アウト判断
BR SO_END
SKGE COUNT, #2
BR LOOP_SO
:-----
: タイム・アウト
:-----
SET1 SIO_ERR ; 500msタイム・アウト
BR SIO_OFF
;

```

SUMMARY '%%', ''
 (5) 送信終了

```

                送信終了時には、SO_OKフラグをセットします。
.EJ-1
%%
-----
:
:   送信終了
:
-----
SO_END:
    SETI   SO_OK           ; 送信終了
    BR     SIO_OFF
:
EJECT
SUMMARY '%%', 'シリアル受信'
    シリアル受信を行うためのルーチンです。
    シリアル受信を行うためのルーチンです。
    ここでは、外部からのシリアル・クロックに同期して、スレーブ側からのデータを受信します。
    このとき、通信中に異常が発生したときには、SIO_ERRフラグをセットしてこのルーチンを抜けます。異常もなしに送信を終了したと
    きには、SI_OKフラグをセットしてこのルーチンを抜けます。
%%
-----
:
:   シリアル受信ルーチン
:
-----
SI_PRO:
-----
:
:   ステータス・クリア
:
-----
CLR4     SIO_END, SIO_ERR, SO_OK, SI_OK   ; シリアル通信状態クリア
SUMMARY '%%', '
(1) 受信準備
    ここでは、シリアル・インタフェースを使用して、受信を行うための準備を行います。
    シリアル・インタフェースをシリアル受信時のモードを設定し、SI, SO, SCK端子をシリアル・モードに設定します。
    シリアル受信時のモード設定には、SIOHIZ, SIOCK0, SIOCK1を設定します。
    シリアル・モードの設定には、SIOENを設定します。
%%
-----
:
:   シリアル受信状態に設定
:
-----
INITPLG NOT SIOTS, SIOHIZ, NOT SIOCK1, NOT SIOCK0 ; シリアル・モード設定
                                                ; SO端子-ハイ・インピーダンス状態
                                                ; SCK端子-外部入力
SETI     SIOEN                               ; シリアル通信設定
SETI     IFSIO                              ; シリアル通信時の割り込み許可
%%
SUMMARY '%%', '
(2) 受信開始
    受信を開始する前に、タイム・アウト用のカウンタをクリアします。
    カウンタ・クリア後、受信を開始します。
    受信の開始は、SIOTSフラグをセットすることで可能です。受信を開始したら、BUSY状態を解除します。
.EJ-1
%%
-----
:
:   タイム・アウト時間の設定
:
-----
MOV     COUNT, #0                          ; タイム・アウト用カウンタ・クリア
SETI   WTMRES                              ; 時計用タイマ・リセット
CLR1   IRQWTM                              ; 時計用タイマの割り込み要求クリア
SETI   IPWTM                               ; 時計用タイマの割り込み許可
:
:   受信スタート
:
-----
SETI   SIOTS                               ; シリアル通信開始
CLR1   BUSY                                ; BUSY状態解除
EI
SUMMARY '%%', '
(3) 受信終了判断
    受信の終了判断は、SIO_ENDフラグをチェックします。
    SIO_ENDフラグは、シリアル通信にて8ビット・データを受信し終了すると、シリアル割り込みが発生し、その割り込み処

```

理内でセットされます。
また、ここでは、タイム・アウト用のカウンタの値も同時にチェックし、0.5秒経過すると通信エラーとします。

```

%%
:-----
:      シリアル受信
:-----
LOOP_SI:
  SET1  WDTRES
  SKP1  SIO_END           ; 8ビット通信判断
  BR    SIO_END
  SKGE  COUNT.#2         ; タイム・アウト判断
  BR    LOOP_SI
:-----
:      タイム・アウト
:-----
  SET1  SIO_ERR           ; タイム・アウト
  BR    SIO_OFF
:

```

SUMMARY '%%','' (4) 受信終了
受信終了時には、SI_OKフラグをセットし、シフト・レジスタより、受信データを入力します。

```

%%
:-----
:      受信終了
:-----
SI_END:
  SET1  SI_OK             ; シリアル受信終了
  GET   DBF,SIOSPR        ; 受信データの入力
  ST    I_DATA_H,DBP1     ; 受信データ設定
  ST    I_DATA_L,DBP0

```

SUMMARY '%%','' (5) シリアル通信終了処理
シリアル通信終了時には、シリアル通信モードを解除します。

```

.EJ-1
%%
:-----
:      シリアル通信状態の解除
:-----
SIO_OFF:
  DI
  SET1  MODE              ; シリアル受信モード設定
  SET1  POBB102           ; BUSY端子=出力
  SRT1  BUSY              ; BUSY端子をBUSY状態
  SET1  SIOHIZ            ; SO端子=ハイ・インピーダンス状態
  CLR1  SIOEN             ; シリアル通信モード解除
  CLR2  IPFTM,IPSIO      ; 時計用タイマとシリアル通信用割り込みを禁止
  RET
:

```

BJBCT
SUMMARY '%%','' ・割り込みルーチン
このプログラムにて使用した、割り込みルーチンは、シリアル割り込みと、時計用割り込みです。
(1) シリアル割り込みルーチン
この割り込みは、シリアル通信時に8ビットのデータを送信または受信したときに発生します。
ここでは、通信の終了を示すために、SIO_ENDフラグをセットしています。

```

%%
:-----
:      割り込みルーチン
:-----
:      シリアル通信割り込み
:-----
SIO_INT:

```

```

SETI   SIO_END           ; 8ビット送信を終了
BR     EI_RET1
;

```

SUMMARY '%', ''
 (2) 時計用割り込み
 この割り込みは、時計用タイマに同期して割り込みを発生します。割り込み発生周期は、250msです。
 ここでは、250msごとにCOUNTをインクリメントしています。

```

%%
:.....
:      時計カウンタ割り込み
:.....
CLK_INT:
ADD    COUNT, #1        ; タイムアウト用カウンタをインクリメント
EI_RET1:
EI     RETI              ; 割り込み処理から復帰
;
END

```


クロスレファレンス

第8章 スタンバイ機能およびシステム・クロックの切り替え

μPD17201Aは、スタンバイ機能としてHALTモードとSTOPモードがあります。

スタンバイ機能を使用することにより消費電流を低減することができます。

また、μPD17201Aは、メイン・クロック (X) とサブ・クロック (XT) の2系統の発振回路を内蔵しており、どちらもシステム・クロックとして使用することができます。

8.1 プログラム説明

8.1.1 HALTモード

HALTモードは、メイン・クロックを発振させた状態でプログラムの実行を一時停止させ、消費電流を抑えるときに使用します。

ここでは、HALT命令の使用例およびシステム・クロックの切り替え方を紹介します。

最初に電源立ち上げ時のサブ・クロックの安定待ち時間 (約10秒) を考慮しなければなりません。

このプログラムでは、8ビット・タイマを使用して約10秒待っています。

次にシステム・クロックをメイン・クロックからサブ・クロックへ切り替え、メイン・クロックの発振を停止します。ここで使用するHALT命令のオペランドの値は1000Bで、解除条件は以下のとおりです。

- ① 割り込み許可フラグ (IPTM, IPWTM, IPSIO, IP) がセットされている割り込みに対し、割り込み要求 (IRQTM, IRQWTM, IRQSIO, IRQ) が発生したとき
- ② P0A₀–P0A₃のいずれかの端子がロウ・レベルになったとき
- ③ P0B₀–P0B₃のいずれかの端子がハイ・レベルになったとき

HALT命令のオペランドにはこの他、0010B, 1010Bがあります。

それぞれの解除条件はデータ・シートを参照してください。

また、メイン・クロックを発振停止状態から発振開始状態に変更したあと、システム・クロックをサブ・クロックからメイン・クロックに切り替える場合、プログラムで10 ms程度の発振安定時間を考慮しなければなりません。

8.1.2 STOPモード

STOPモードは、メイン・クロックの発振を停止させた状態でプログラムの実行を一時停止させ、消費電流をきわめて低くするときに使用します。

ここでは、STOP命令の使用例を紹介します。

STOP命令のオペランドの値は、1000Bのみで解除条件は以下のとおりです。

- ① 割り込み許可フラグ (IPTM, IPWTM, IPSIO, IP) がセットされている割り込みに対し、割り込み要求 (IRQTM, IRQWTM, IRQSIO, IRQ) が発生したとき

- ② P0A₀–P0A₃のいずれかの端子がロウ・レベルになったとき
- ③ P0B₀–P0B₃のいずれかの端子がハイ・レベルになったとき

STOPモード解除後、次の命令を実行するまでの時間は、8ビット・タイマのモジュロ・レジスタに設定した値で決まります。モジュロ・レジスタの値をTMMとした場合この時間は式8-1であらわされます。

$$(TMM + 1) \times 1024 / fsys \quad [\text{秒}] \quad \dots\dots\dots \text{式 8-1}$$

fsys : システム・クロックの周波数

よって、システム・クロックとして4 MHzを使用し、STOPモード解除から次の命令を実行するまでの時間を10 msとすると、モジュロ・レジスタに設定する値TMMは、以下のとおりになります。

$$\begin{aligned} 10 \text{ [ms]} &= 10000 \text{ [}\mu\text{s]} \\ &= (TMM + 1) \times 1024 / 4 \\ &= (TMM + 1) \times 256 \text{ [}\mu\text{s]} \\ TMM &= 10000 / 256 - 1 \\ &= 38.0625 \\ &= 26 \text{ [HEX]} \end{aligned}$$

8.2 システム・クロック

システム・クロックとして、メイン・クロック (X)、サブ・クロック (XT) のどちらを使用するかを選択は、レジスタ・ファイル上のSYSCKフラグで行います。

また、メイン・クロックは消費電流を低減するためにレジスタ・ファイル上のXENフラグをリセットすることにより発振を停止させることができます。

サブ・クロックを使用しない場合は、XT_{IN}端子をGNDに接続し、XT_{OUT}端子をV_{REG}端子に接続してください。

このとき、ウォッチドッグ・タイマのカウント・クロックはメイン・クロックとなります。

サブ・クロックを使用した場合、ウォッチドッグ・タイマのカウント・クロックはサブ・クロックとなります。

8.3 ウォッチドッグ・タイマ動作タイミング

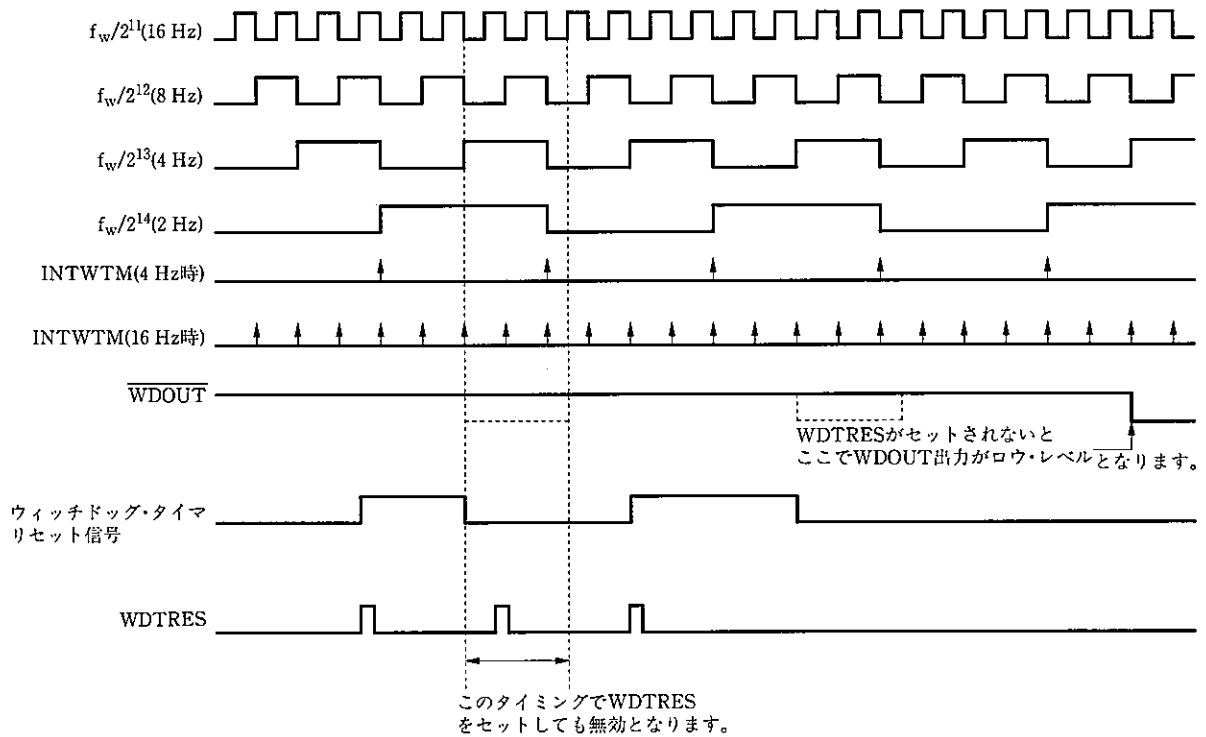
ウォッチドッグ・タイマのリセットは、下記のタイミングにおいて2 Hz信号と4 Hz信号がともにハイ・レベルになる前にレジスタ・ファイル (03H番地、ビット3) のWDTRESをセット (WDTRES=1) することで行います。

2 Hz信号と4 Hz信号がともにハイ・レベルになっている期間は、ウォッチドッグ・タイマのリセットがで

きません。

したがって、ウォッチドッグ・タイマによる暴走検出を禁止するためには約340 ms以下の周期でWDTRESをセットするプログラムにしてください。

図8-1 ウォッチドッグ・タイマの動作タイミング



<使用するハードウェア>

番号	端子名	使用説明
67	WDOUT	<p>暴走検出用の出力です。</p> <p>ウォッチドッグ・タイマやスタック・オーバーなどが発生した場合に、ロウ・レベルとなります。</p> <p>ウォッチドッグ・タイマを使用する場合は、外部でRESET端子（65番）に接続してください。</p> <p>ウォッチドッグ・タイマを使用しない場合は、何も接続しません。</p>

ドキュメント



AS17K Z1.10 D1 << E17201A DOCUMENT >> 11:28:42 02/25/91

TABLE OF CONTENTS

	PAGE
第 8 章 スタンバイ・モードおよびシステム・クロック	1
<<プログラム説明>>	1
<<サブ・クロックの安定時間待ち>>	1
<<システム・クロックの切り替え (メイン→サブ)>>	2
<<システム・クロックの切り替え (サブ→メイン)>>	2

第 8 章 スタンバイ・モードおよびシステム・クロック

PUBLIC (LAB) : RAMC_END
 ADDR RANGE : 0250H - 0295H

<プログラム説明>

1. H A L Tモード

H A L Tモードはメイン・クロックを発振させた状態でプログラムの実行を一時停止させ、消費電流を抑えるときに使用します。
 ここでは、H A L T命令の使用例およびシステム・クロックの切り替え方を紹介します。

```

ENTRANCES          :-
MEMORIES CHANGED   :-
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

<サブ・クロックの安定時間待ち>

電源立ちあげ時の安定時間として約10秒待ちます。
 ここでは、10秒のカウントを8ビット・タイマで行っています。

```

ENTRANCES          :RAMC_END
MEMORIES CHANGED   :CNT10_H   CNT10_L
                   :CNT10_M
MEMORIES REFERRED  :-
MEMORIES MANIPULATED :-
FLAGS CHANGED      :-
FLAGS REFERRED     :-
DATA REFERRED      :-
BRANCH TO          :-
SUBROUTINES CALLED :-
LABELS MANIPULATED :-
SYSTEM CALL        :-
    
```

<システム・クロックの切り替え（メイン→サブ）>

システム・クロックをメイン・クロックからサブ・クロックに切り替え、メイン・クロックの発振を停止します。

```
EXTRN (MEM) : SOURCE1, SOURCE2
EXTRN (LAB) : DISPLAY, TIMER
ADDR RANGE : 02B1H - 02E4H
```

<システム・クロックの切り替え（サブ→メイン）>

メイン・クロックを発振停止状態から発振開始状態に変更した後、システム・クロックをサブ・クロックからメイン・クロックに切り替える場合、プログラムで10ms程度の発振安定時間を考慮しなければなりません。

```
ENTRANCES           :MAIN_CLOCK
MEMORIES CHANGED    :CNT_10ms
MEMORIES REFERRED    :-
MEMORIES MANIPULATED :-
FLAGS CHANGED        :-
FLAGS REFERRED       :-
DATA REFERRED        :-
BRANCH TO            :-
SUBROUTINES CALLED   :-
LABELS MANIPULATED   :-
SYSTEM CALL          :-
```

AS17K Z1.10 D1 << E17201A DOCUMENT >> 11:52:04 02/26/91

TABLE OF CONTENTS

< S T O P モード >

PAGE
1

<STOPモード>

2. STOPモード

STOPモードは、メイン・クロックの発振を停止させた状態でプログラムの実行を一時停止させ、消費電流をきわめて低くするときに使用します。
ここでは、STOP命令の使用例を紹介します。

STOPモード解除後、次の命令を実行するまでの時間は、8ビット・タイマのモジュロ・レジスタに設定した値によって決まります。

システム・クロックとして4MHzを使用した場合、STOPモード解除から次の命令を実行するまでの時間は、

$$(TMM+1) \times 256 [\mu s]$$

となります。

ここでは、この時間を約10msとした場合の例をあげます。

$$(TMM+1) \times 256 = 10000 [\mu s] = 10 [ms]$$

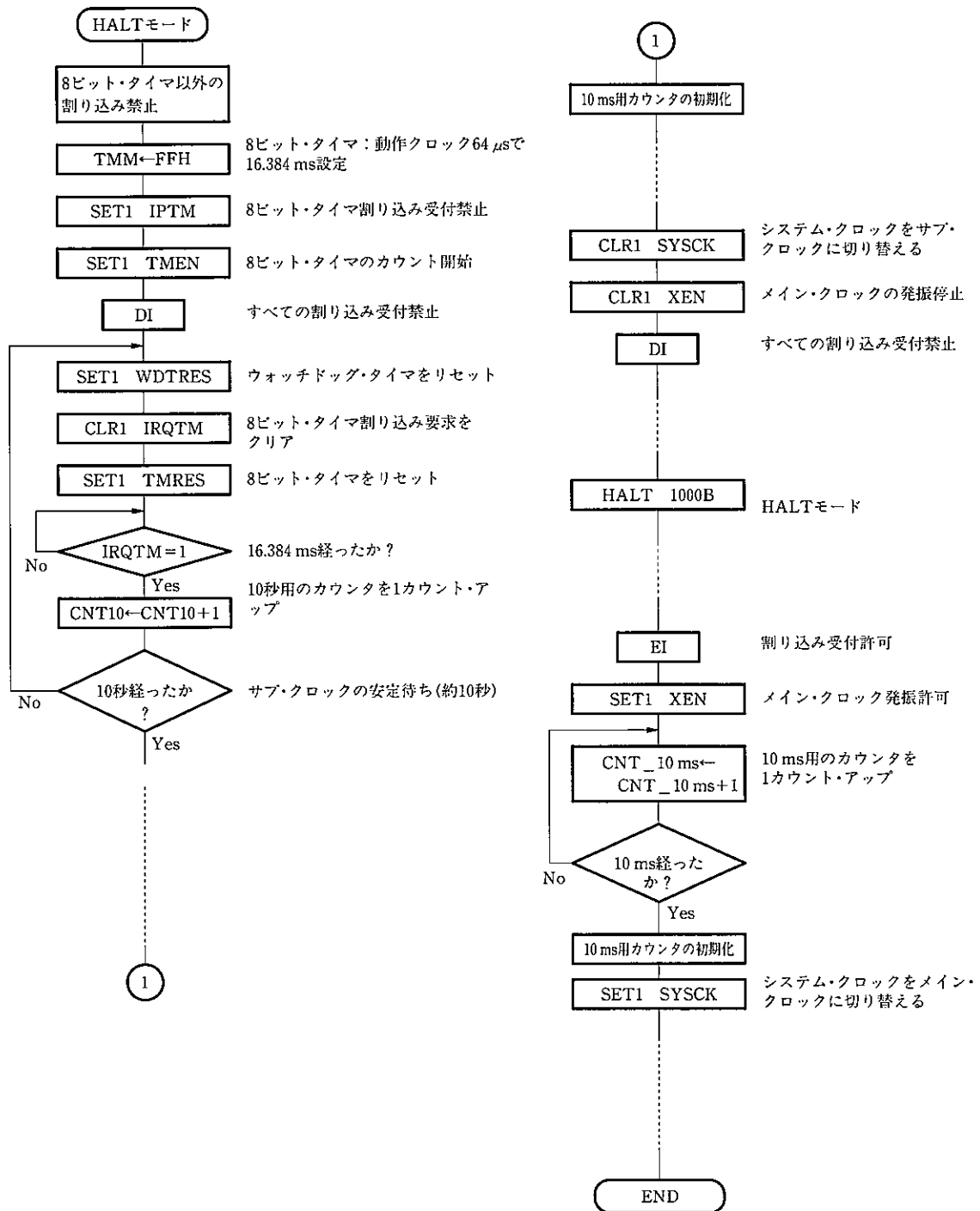
$$TMM = 26 (\text{HEX})$$

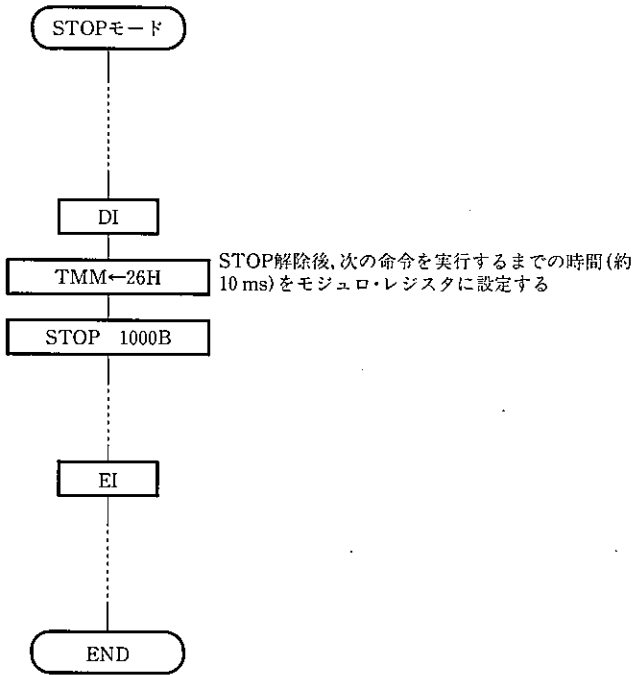
ADDR RANGE : 01F9H - 01PDH

フロー・チャート



フロー・チャート





リスト


```

INCLUDE 'HALT2.PUB'
SUMMARY '%%', ' 第 8 章 スタンバイ・モードおよびシステム・クロック
%%
CNT10_H      MEM      0.39H      ;サブ・クロックの安定時間(約10秒)待ち用のカウンタです。
CNT10_M      MEM      0.3AH
CNT10_L      MEM      0.3BH

SUMMARY '%%', '<プログラム説明>'

1. HALTモード

HALTモードはメイン・クロックを発振させた状態でプログラムの実行を一時停止させ、消費電流を抑える
ときに使用します。
ここでは、HALT命令の使用例およびシステム・クロックの切り替え方を紹介します。
%%
SUMMARY '%%', '<サブ・クロックの安定時間待ち>'

電源立ち上げ時の安定時間として約10秒待ちます。
ここでは、10秒のカウントを8ビット・タイマで行っています。
%%

RAMC_END:
;.....
;... サブ・クロックの安定時間(約10秒)待ち ...
;.....
;... INTERRUPT ...
;.....
INITPLG NOT IRQSIO,NOT IRQWTM,NOT IRQ,NOT IRQTM
INITPLG NOT IPSIO,NOT IPWTM,NOT IP,IPTM      ;8ビット・タイマ以外の割り込み以外は禁止
;.....
;... 8ビット・タイマ ...
;.....
INITPLG      NOT TMEN,NOT TMRES,TMCK1,NOT TMCK0      ;64 us TIMER
;.....
;... 10秒用カウンタ初期化 ...
;.....
MOV      CNT10_L,#0
MOV      CNT10_M,#0
MOV      CNT10_H,#0
;.....
;... 8ビット・タイマ設定 ...
;.....
MOV      DBF1,#0FH      ;8ビット・タイマを16、384msに設定
MOV      DBF2,#0FH
PUT      TMM,DBF
SET1     IPTM            ;8ビット・タイマ割り込み受付許可
SET1     TMEN           ;8ビット・タイマのカウント開始

DI      ;すべての割り込み禁止

SYS_CLOCK:
SET1     WDTRES         ;ウォッチドック・タイマをリセット
CLR1     IRQTM          ;8ビット・タイマ割り込み要求をクリア
SET1     TMRES         ;8ビット・タイマをリセット

WAIT_10a:
SKT1     IRQTM
BR       WAIT_10a
ADD      CNT10_L,#1
ADDC    CNT10_M,#0
ADDC    CNT10_H,#0
SET2    CMP,Z
SUB     CNT10_L,#3
SUB     CNT10_M,#6
SUB     CNT10_H,#2
SKT1    Z
BR      SYS_CLOCK
    
```

```
CLR1  TMEN  
CLR1  IRQM  
CLR1  IPTM  
EI
```

:割り込み許可

END


```

INCLUDE 'HALT4.PUB'
;.....
;*** H A L T モード ***
;.....
CNT_10ms      MEM      0.16H          ;メイン・クロックの発振安定時間(約10ms)待ち用のカウンタです。
;.....
;*** 10ms用カウンタの初期化 ***
;.....
MOV          CNT_10ms,#9
;.....
;*** DISPLAY CLOCK ***
;.....
CALL         DISPLAY          ;表示処理
SET1        IPWTM            ;時計タイマ用割り込み受付許可
CLR1        IPTM             ;8ビット・タイマ割り込み受付禁止
;#####
;### ENVELOPE ###
;#####
CLR1        NRZ              ;NRZ
CLR1        NRZBF           ;NRZBF
SUMMARY     '%', '<システム・クロックの切り替え(メイン→サブ)>'
;システム・クロックをメイン・クロックからサブ・クロックに切り替え、メイン・クロックの発振を停止します。
%%
;.....
;*** システム・クロックの切り替え ***
;.....
CLR1        SYSCK           ;システム・クロックをサブ・クロックに切り替え
CLR1        XEN             ;メイン・クロックの発振を停止します。
DI          ;すべての割り込み禁止
SUB_CLOCK:
;#####
;### HALT ###
;#####
CALL        DISPLAY          ;時計表示処理
HALT        1000B
SET2        P0BGIO,P0CGIO
CLR2        P0AGIO,P1AGIO
MOV         SOURCE1,#0
MOV         SOURCE2,#0
SET1        WDTRES
SRT1        IRQWTM           ;時計タイマ用割り込み要求有りか?
BR          MAIN_CLOCK
CLR1        IRQWTM
CALL        TIMER            ;時計カウント処理
BR          SUB_CLOCK
;
SUMMARY     '%', '<システム・クロックの切り替え(サブ→メイン)>'
;メイン・クロックを発振停止状態から発振開始状態に変更した後、システム・クロックをサブ・クロックからメイン・クロックに切り替える場合、プログラムで10ms程度の発振安定時間を考慮しなければなりません。
%%
MAIN_CLOCK:
;時計カウント以外の処理は、メイン・クロックで行う
EI          ;割り込み許可
SET1        XEN             ;メイン・クロック発振許可
;.....
;*** メイン・クロックの発振安定時間(約10ms)待ち ***
;.....
W_10ms:
ADD         CNT_10ms,#1

```

```
      SBT1  CY
      BR   W_10ms
      MOV  CNT_10ms, #9

OPTION SET1  SYSCK      ;システム・クロックをメイン・クロックに切り替え
OPTRES OPEN
OPTCK  USEX, USEXT
ENDOP
END
```

SUMMARY '%%', '<STOPモード>'

2. STOPモード

STOPモードは、メイン・クロックの発振を停止させた状態でプログラムの実行を一時停止させ、消費電流をきわめて低くするときに使用します。ここでは、STOP命令の使用例を紹介します。

STOPモード解除後、次の命令を実行するまでの時間は、8ビット・タイマのモジュロ・レジスタに設定した値によって決まります。

システム・クロックとして4MHzを使用した場合、STOPモード解除から次の命令を実行するまでの時間は、

$$(TMM+1) \times 256 [\mu s]$$

となります。

ここでは、この時間を約10msとした場合の例をあげます。

$$(TMM+1) \times 256 = 10000 [\mu s] = 10 [ms]$$

$$TMM = 26 (\text{HEX})$$

```

%%
;.....
;... STOPモード ...
;.....

DI
MOV   DBP1,#2      ;STOPモード解除後、次の命令を実行するまでの時間(約10ms)を
MOV   DBP0,#6      ;モジュロ・レジスタに設定する
PUT   TMM,DBF

;%%%%%%%%%%%%%%%%%%%%%%%%%%
;%%          STOP          %%
;%%%%%%%%%%%%%%%%%%%%%%%%%%
STOP  1000B        ;STOPモード

RND
    
```


クロスレファレンス

AS17K Z1.10 D1 << E17201A XREF LIST >> 19:33:50 02/19/91 PAGE 02-001

PROG -

SOURCE - HALT2.ASM

SYMBOL	TYPE	A	VALUE	/REF	(#DEF)
CNT10_H	MEM	L	0.39	/#	5 , 43 , 63 , 67
CNT10_L	MEM	L	0.3B	/#	7 , 41 , 61 , 65
CNT10_M	MEM	L	0.3A	/#	6 , 42 , 62 , 66
RAMC_END	LAB	P	260	/	1-4 , # 25
SYS_CLOCK	LAB	L	279	/#	54 , 69
WAIT_10#	LAB	L	281	/#	58 , 60

TOTAL SYMBOLS - 8

END OF XREF LIST

AS17K Z1.10 D1 << E17201A XREF LIST >> 11:28:32 02/26/91 PAGE 04-001

PROG -

SOURCE - HALT4.ASM

SYMBOL	TYPE	A	VALUE	/REF	(#DEF)				
CNT_10ms	MEM	L	0.16	/#	5	,	10	,	66 , 68
DISPLAY	LAB	E	32	/	1-4	,	14	,	39
MAIN_CLOCK	LAB	L	2D8	/	47	,	# 59		
OPTCK	MAC	L	*****	/	74				
OPTRES	MAC	L	*****	/	73				
SOURCE1	MEM	E	0.71	/	1-6	,	43		
SOURCE2	MEM	E	0.72	/	1-5	,	44		
SUB_CLOCK	LAB	L	2C4	/#	35	,	50		
TMSR	LAB	E	11	/	1-7	,	49		
W_10ms	LAB	L	2DC	/#	65	,	68		

TOTAL SYMBOLS - 10

END OF XREF LIST

AS17K Z1.10 D1 << E17201A XREF LIST >> 11:51:54 02/28/91 PAGE 02-001

PROG -

SOURCE - STOP2.ASM

SYMBOL TYPE A VALUE /REF (#DEP)

TOTAL SYMBOLS - 0

END OF XREF LIST

— NEC 日本電気株式会社 —

本社 〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル)
半導体第一、第二販売事業部 〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル) 東京(03)3454-1111
関西支社半導体販売部 〒540 大阪市中央区城見一丁目4番24号(日本電気関西ビル) 大阪(06)945-3178
 大阪(06)945-3200
中部支社半導体販売部 〒460 名古屋市中区栄四丁目14番5号(松下中日ビル) 名古屋(052)242-2755

北海道支社	札幌(011)231-0161	立川支社	立川(0425)26-0911
東北支社	仙台(022)261-5511	静岡支社	静岡(0472)27-5441
関東支社	東京(0196)51-4344	岡崎支社	岡崎(054)255-2211
中部支社	名古屋(0236)23-5511	津支社	津(0559)53-4455
近畿支社	大阪(0249)23-5511	奈良支社	奈良(0534)52-2711
中国支社	岡山(0246)21-5511	和歌山支社	和歌山(0762)23-1621
四国支社	高松(0258)36-2155	徳島支社	徳島(0776)22-1866
九州支社	福岡(0292)26-1717	香川支社	高松(075)221-8511
		愛媛支社	松山(078)332-3311
		高松支社	高松(082)242-5504
		鳥取支社	鳥取(0857)27-5311
		島根支社	松江市(0862)25-4455
		岡山支社	岡山市(0878)36-1200
		広島支社	広島(0897)32-5001
		山口支社	山口(0899)45-4111
		徳島支社	徳島(092)271-7700
		香川支社	高松(093)541-2887
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
		香川支社	
		愛媛支社	
		高松支社	
		鳥取支社	
		島根支社	
		岡山支社	
		広島支社	
		山口支社	
		徳島支社	
	</		